Software Architecture

Berta Pauline
Berrada El Ghali
Conceicao Nunes Joao
Skiker Hicham

REPORT

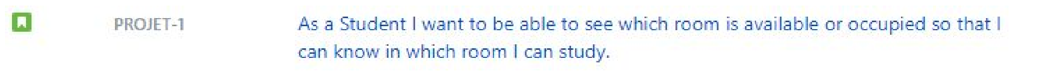
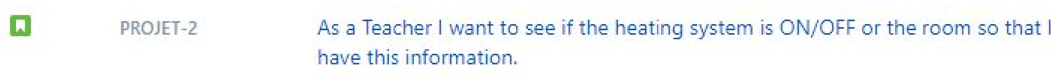
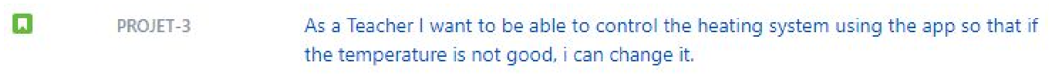
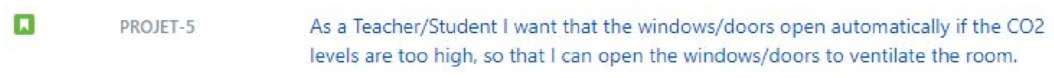
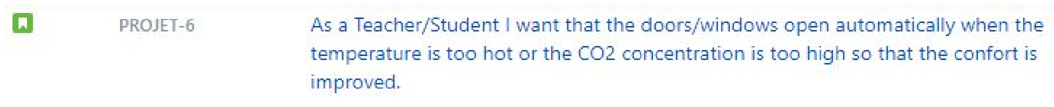
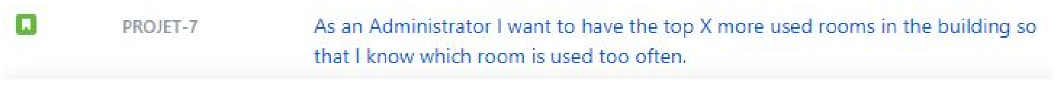
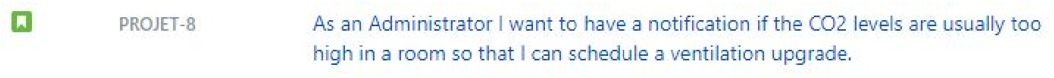
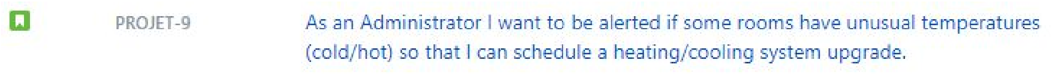
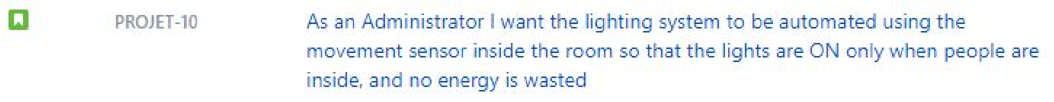
Automation of the INSA's rooms management

I. Subject :

You are 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.

II. Project Organization with Jira :

User Stories List :

Tickets	
N° 1 :	 🟩 PROJET-1 As a Student I want to be able to see which room is available or occupied so that I can know in which room I can study.
N° 2 :	 🟩 PROJET-2 As a Teacher I want to see if the heating system is ON/OFF or the room so that I have this information.
N° 3 :	 🟩 PROJET-3 As a Teacher I want to be able to control the heating system using the app so that if the temperature is not good, i can change it.
N° 4 :	 🟩 PROJET-5 As a Teacher/Student I want that the windows/doors open automatically if the CO2 levels are too high, so that I can open the windows/doors to ventilate the room.
N° 5 :	 🟩 PROJET-6 As a Teacher/Student I want that the doors/windows open automatically when the temperature is too hot or the CO2 concentration is too high so that the confort is improved.
N° 6 :	 🟩 PROJET-7 As an Administrator I want to have the top X more used rooms in the building so that I know which room is used too often.
N° 7 :	 🟩 PROJET-8 As an Administrator I want to have a notification if the CO2 levels are usually too high in a room so that I can schedule a ventilation upgrade.
N° 8 :	 🟩 PROJET-9 As an Administrator I want to be alerted if some rooms have unusual temperatures (cold/hot) so that I can schedule a heating/cooling system upgrade.
N° 9 :	 🟩 PROJET-10 As an Administrator I want the lighting system to be automated using the movement sensor inside the room so that the lights are ON only when people are inside, and no energy is wasted

N° 10 :	🟩	PROJET-11	As an Administrator I want the heating system to be automated so that on weekends it isn't ON for example.
N° 11 :	🟩	PROJET-12	As an Administrator I want to be alerted if movement is detected on a specific time period or on the week-ends so that the security of the building is ensured.

Tasks List :

Tasks			
N° 1 :	✓	PROJET-13	Design Database
N° 2 :	✓	PROJET-14	Communication with Sensors
N° 3 :	✓	PROJET-15	Communication between Administrator and System
N° 4 :	✓	PROJET-16	Developing Automatization Code
N° 5 :	✓	PROJET-17	Developing Threshold Code
N° 6 :	✓	PROJET-18	Developing Data Analysis Code

III. Project Description :

Our project is a simulation of an Insa's room where we put some sensors and actuators to allow the Administrator, the Teachers and the Students to interact with the IoT system. The users can interact with the system using the web interface we developed.
The aim is, for example, to be able to regulate the temperature of the room, to control the rate of CO2 in the air and to switch the light on or off depending on whether there is a presence in the room or not.

We consider four sensors(Port number) :

- Mouvement (8081), *Project Name* : CaptMouv
- Temperature (8082), *Project Name* : CaptTemp
- Rate of CO2 (8083), *Project Name* : CaptCO2
- Humidity (8084), *Project Name* : CaptHum

Four actuators:

- Automatic Door (8085), *Project Name* : AutomPorteApplication
- Automatic Window (8086), *Project Name* : AutomFenetreApplication
- Automatic Light (8087), *Project Name* : AutomLumiereApplication
- Automatic Heating (8088), *Project Name* : AutomChauffageApplication

A controller :

- Controller (8091), *Project Name* : Controller

A local host (Address) :

- Local Host (127.0.0.1)

IV. Demonstration example :

In this part, we want to realize a demonstration of several functionalities which compose our project, and show how some of the functionalities of our web interface work.
We are going to introduce the lighting system that we set up in response to the request of ticket number 9.

Lighting system :

First of all, we want to check if our Light Control is activated or not.

127.0.0.1:8091/controller/isControlLumiereAutomatiqueActivated/ `false`

Then, we activate it and check if the system has been updated.

127.0.0.1:8091/controller/setControlLumiereAutomatique/?oN=true/
127.0.0.1:8091/controller/isControlLumiereAutomatiqueActivated/ `true`

We check if the light is ON or not.

127.0.0.1:8087/AutomLumiereApplication/isON/ `false`

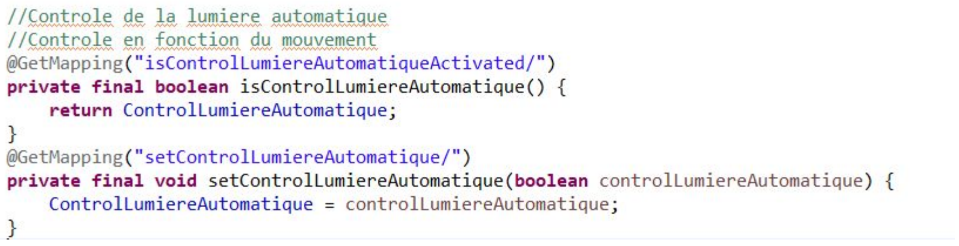
We simulate a movement detection in the room and check, with the Controller, if the light is ON now.

127.0.0.1:8087/AutomLumiereApplication/isON/ `true`

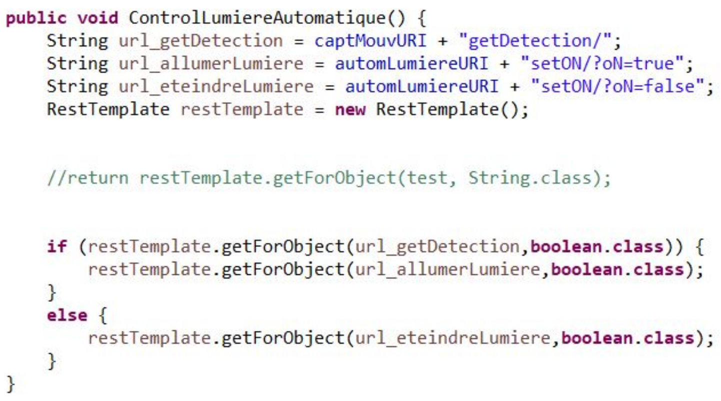
On the other side, we check if the light is OFF when the movement stops.

127.0.0.1:8087/AutomLumiereApplication/isON/ `false`

Corresponding lines of code (Controller) :



```java
//Controle de la lumiere automatique
//Controle en fonction du mouvement
@GetMapping("isControlLumiereAutomatiqueActivated/")
private final boolean isControlLumiereAutomatique() {
    return ControlLumiereAutomatique;
}
@GetMapping("setControlLumiereAutomatique/")
private final void setControlLumiereAutomatique(boolean controlLumiereAutomatique) {
    ControlLumiereAutomatique = controlLumiereAutomatique;
}
```

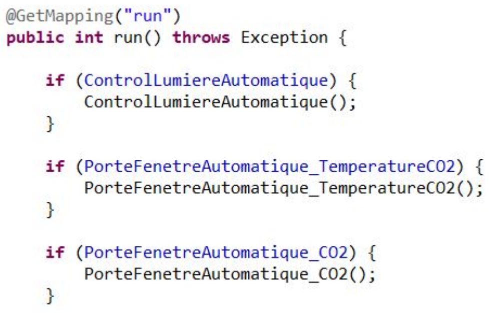


```java
public void ControlLumiereAutomatique() {
    String url_getDetection = captMouvURI + "getDetection/";
    String url_allumerLumiere = automLumiereURI + "setON/?oN=true";
    String url_eteindreLumiere = automLumiereURI + "setON/?oN=false";
    RestTemplate restTemplate = new RestTemplate();


    //return restTemplate.getForObject(test, String.class);


    if (restTemplate.getForObject(url_getDetection,boolean.class)) {
        restTemplate.getForObject(url_allumerLumiere,boolean.class);
    }
    else {
        restTemplate.getForObject(url_eteindreLumiere,boolean.class);
    }
}
```

This is part of the run function that controls everything. The automatic lighting system (ticket n°9) but also, in the same way, the automatic window/door (ticket n°4 & n°5).



```java
@GetMapping("run")
public int run() throws Exception {

    if (ControlLumiereAutomatique) {
        ControlLumiereAutomatique();
    }

    if (PorteFenetreAutomatique_TemperatureCO2) {
        PorteFenetreAutomatique_TemperatureCO2();
    }

    if (PorteFenetreAutomatique_CO2) {
        PorteFenetreAutomatique_CO2();
    }
```

The second example concerns the Temperature System, that is the ticket number 8 with the use of the OM2M technology.

Temperature System :

First of all, we want to check if our Temperature Alarm is activated or not.

127.0.0.1:8091/controller/getTemperatureAlarm/ `false`

Then, we activate it and check if the system has been updated.

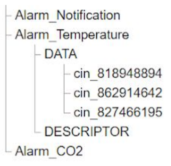
127.0.0.1:8091/controller/setTemperatureAlarm/?active=true/

127.0.0.1:8091/controller/getTemperature Alarm/ `true`

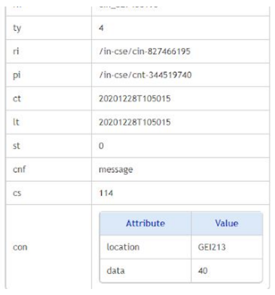
We simulate a temperature variation in the room (with temperature = 40°C) and check if we receive an alert message on OM2M.

127.0.0.1:8082/CaptTemp/getTemp/ `40.0`

127.0.0.1:8080/webpage/



```
Alarm_Notification
Alarm_Temperature
    DATA
        cin_818948894
        cin_862914642
        cin_827466195
    DESCRIPTOR
Alarm_CO2
```

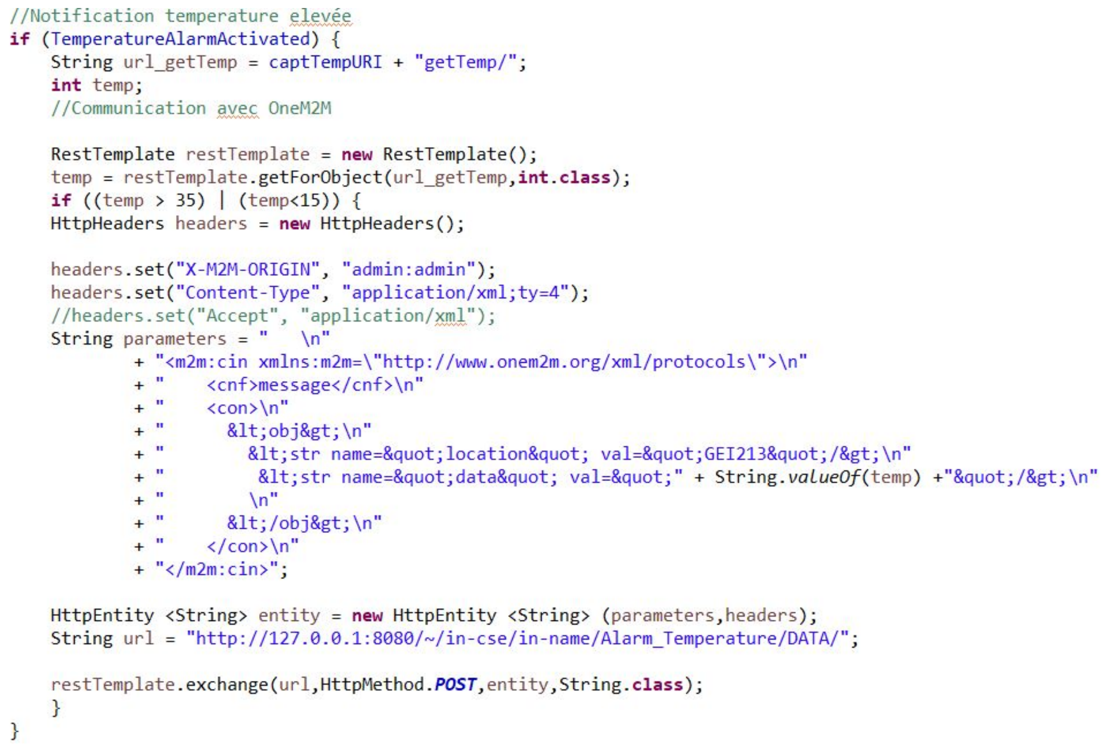


ty	4
ri	/in-cse/cin-827466195
pi	/in-cse/cnt-344519740
ct	20201228T105015
lt	20201228T105015
st	0
cnf	message
cs	114
con	

Attribute	Value
location	GEI213
data	40

Corresponding lines of code (Controller) :

This is part of the run function that controls this temperature system (ticket n°8) but also, in the same way, the CO2 detection system (ticket n°7).



```java
//Notification temperature elevée
if (TemperatureAlarmActivated) {
    String url_getTemp = captTempURI + "getTemp/";
    int temp;
    //Communication avec OneM2M

    RestTemplate restTemplate = new RestTemplate();
    temp = restTemplate.getForObject(url_getTemp,int.class);
    if ((temp > 35) | (temp<15)) {
    HttpHeaders headers = new HttpHeaders();

    headers.set("X-M2M-ORIGIN", "admin:admin");
    headers.set("Content-Type", "application/xml;ty=4");
    //headers.set("Accept", "application/xml");
    String parameters = "    \n"
            + "<m2m:cin xmlns:m2m=\"http://www.onem2m.org/xml/protocols\">\n"
            + "     <cnf>message</cnf>\n"
            + "     <con>\n"
            + "       &lt;obj&gt;\n"
            + "         &lt;str name=&quot;location&quot; val=&quot;GEI213&quot;/&gt;\n"
            + "          &lt;str name=&quot;data&quot; val=&quot;" + String.valueOf(temp) +"&quot;/&gt;\n"
            + "         \n"
            + "       &lt;/obj&gt;\n"
            + "     </con>\n"
            + "</m2m:cin>";

    HttpEntity <String> entity = new HttpEntity <String> (parameters,headers);
    String url = "http://127.0.0.1:8080/~/in-cse/in-name/Alarm_Temperature/DATA/";

    restTemplate.exchange(url,HttpMethod.POST,entity,String.class);
    }
}
```

V. Testing our entire Project :

If you want to test our entire project to see all the functionalities that we have create in response to all our tickets, this is the GIT link :

https://git.etud.insa-toulouse.fr/conceica/ProjetSOA-PaulineBERTA-JoaoNUNEs-GhaliBERRADA-HichamSKIKER.git

On this link you can find a document which is named "Commands List" with all the commands that you can test, representing all the functionalities that the project can execute, and our Eclipse Project with all the Java code. Do not forget to launch each one of the different projects before testing.