INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

2023 - 2024

INSA rooms management

Mohammed Amine Hatibi
Abdelmajid Anka Soubaai

Département Génie Électrique et Informatique
135, Avenue de Rangueil
31077 Toulouse Cedex 4

Dépôt Git : `https://github.com/Abdel211/Insa_room.git`

Table of contents

1 Introduction

As part of the comprehensive service architecture initiative, we were tasked with conceiving and implementing a web application aimed at efficiently overseeing the management of rooms within the INSA environment. This undertaking necessitated a seamless integration of software services, sensors, and actuators into the application's framework. The sensors play a crucial role by collecting data, and the subsequent analysis of this data informs decision-making processes to determine the optimal actions to be executed through the actuators. This dynamic integration of technology ensures a sophisticated and responsive system that enhances the overall management and functionality of the INSA rooms.

2 Scenarios for Application Development

To conceptualize the development of the application, we have envisioned distinct scenarios, presented herein as user stories:

User	User Story
User 1	As a student or teacher, I need to know which room is used or not in order to be able to use it for studying or giving extra classes.
User 2	As an administrator, I want the lights to turn on or off automatically if movement is detected or not in order to save energy if no one is in the room.
User 3	As an administrator, I want the alarm to trigger if movement is detected late at night when no one is studying or having class in order to ensure safety.

Table 1: User Stories

3 Microservice Description

The project is built upon a microservices architecture implemented with Spring Boot, fostering independent collaboration among its components. Operating in a local environment (localhost), each microservice is assigned a unique port, facilitating streamlined testing on the same host. The application comprises four distinct microservices: a sensor, two actuators, and a controller designed to facilitate seamless interaction.

4 Command list

For more details on each command, please refer to the other report.

5 Demonstration

This section provides a demonstration of the microservices implemented in the project, focusing on the functionalities of automatic light control and alarm control during movement detection.

User 1: Room Usage

This functionality checks if a room is in use using movement detection.

- **Simulate movement:**

  1. `http://localhost:8082/MovementDetection/setDetection/?detection=true`
  2. `http://localhost:8080/Controller/getRoomUsed/`

- **If no movement is detected:**

  1. `http://localhost:8082/MovementDetection/setDetection/?detection=false`
  2. `http://localhost:8080/Controller/getRoomUsed/`

User 2: Light Control

Before checking the light status, we need to activate the automatic light control system in the controller.

1. `http://localhost:8080/Controller/setAutoLightControl/?autoLightControl=true`
2. `http://localhost:8080/Controller/isAutoLightControlActivated/`

Check if the light is on (no movement detected yet):

- `http://localhost:8081/LightControl/isON/`

Simulate movement in the room and check with the controller (after activation) if the light is on:

1. `http://localhost:8082/MovementDetection/setDetection/?detection=true`
2. `http://localhost:8080/Controller/runAuto`
3. `http://localhost:8081/LightControl/isON/`

Check if the light is off when the movement stops:

1. `http://localhost:8082/MovementDetection/setDetection/?detection=false`
2. `http://localhost:8081/LightControl/isON/`

User 3: Alarm Control

The alarm should trigger if movement is detected in a room between 10 pm and 6 am.

1. `http://localhost:8080/Controller/setAutoAlarmControl/?autoAlarmControl=true`
2. `http://localhost:8080/Controller/isAutoAlarmControlActivated/`

Check if the alarm is on (no movement detected yet):

- `http://localhost:8083/AlarmControl/status`

Simulate movement in the room and check with the controller (after activation) if the alarm is on:

1. `http://localhost:8082/MovementDetection/setDetection/?detection=true`
2. `http://localhost:8080/Controller/runAuto`
3. `http://localhost:8083/AlarmControl/status`

Simulate movement at 11 pm (the alarm should trigger). Repeat the steps and check if the alarm is on with the controller:

1. `http://localhost:8082/MovementDetection/setDetection/?detection=true`
2. `http://localhost:8080/Controller/runAuto`
3. `http://localhost:8083/AlarmControl/status`

6 Agile Method

6.1 Room Management Project:

In implementing the Room Management Project, we advocate for the adoption of Agile methodologies, specifically SCRUM. This approach aligns seamlessly with the iterative cycle of conception, programming, and testing. Conducting iterations every two weeks allows for the prompt identification and resolution of bugs or implementation of changes, minimizing the risk of deferring actions until the final production stage, as seen in alternative development methods.

Adhering to Agile principles over a two-quarter period would entail completing four iterations, each spanning two weeks. Key elements would include a prioritized list of objectives, a product backlog, and assigning team roles such as SCRUM Master and Product Owner.

The SCRUM Master ensures the smooth execution of sprints, following SCRUM methodology, and addresses impediments for continuous improvement. The Product Owner manages relationships with stakeholders and end-users of the developed application.

6.2 Water Leak Detection Project

For the Water Leak Detection Project, we embraced a Scrum approach, effectively managing tasks using JIRA software. Scrum sprints, tracked with flexibility, ranged from 6 to 15 days based on task complexity. Our Scrum Master, also the project mentor, provided guidance even when not present at every meeting.

Comprehensive reviews after each sprint evaluated progress, identified improvement areas, and refined strategies for subsequent sprints. Within the team, a designated pair assumed the role of the Product Owner, facilitating communication with stakeholders and streamlining workflows for equipment procurement and issue resolution.

Explore the project's progress on JIRA :

`https://vigileak.atlassian.net/jira/software/projects/SCRUM/boards/1/timeline`

To enhance our development and deployment processes, we implemented Continuous Integration (CI) and Continuous Deployment (CD) methodologies, leveraging Jenkins as our chosen platform for automation. This allowed us to automate the building, testing, and deployment phases, ensuring a more efficient and reliable development pipeline. Explore our Jenkins setup by visiting the following link: `https://jenkins.vigileak.obrulez.fr/job/Vigileak/`

7 Conslusion

This project, an application of skills from the Service Architecture course, involved collaborative design and implementation of an architecture automating INSA's room management. Each team member took charge of specific user stories and microservices, encountering challenges that demanded creative problem-solving and collaborative decision-making.

The project not only enhanced technical abilities but also improved teamwork and communication skills. Committed to delivering a solution meeting requirements and reflecting a deep understanding of course principles, we applied theoretical knowledge in a practical setting, reinforcing comprehension of service architecture concepts.