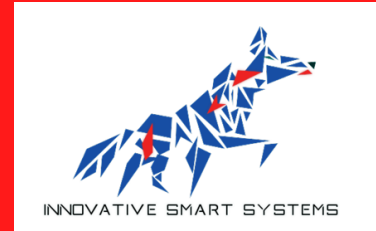


INSA

Institut National des Sciences Appliquées
Toulouse



INSA's rooms management

Amine Benchekroun et Badr Diani

2024/2025

Contents

1	Introduction:	3
2	Description	4
2.1	Microservice:	4
2.2	Scenarios	4
3	Description des Commandes	6
4	Démonstration	8
4.1	Utilisation des salles	8
4.2	Contrôle de l'alarme	9
4.3	Contrôle des lumières	9
4.4	Interaction avec la base de donnée	10
5	Conclusion	11
6	Annexes:	12

Chapter 1

Introduction:

Dans le cadre de la matière Architecture de Service, il nous a été proposé, lors des séances de travaux pratiques, de mettre en œuvre une application web afin de gérer efficacement les salles de l'INSA. Ce projet a pour objectif d'intégrer l'ensemble des services logiciels avec des capteurs et des actionneurs.

Les capteurs installés dans les salles collecteront des données qui seront ensuite analysées pour prendre des décisions sur les actions à effectuer. Ces actions seront exécutées via les actionneurs, et chaque action sera enregistrée dans une base de données, permettant ainsi un suivi précis de tout ce qui s'est passé dans la salle.

Cette intégration dynamique permettra non seulement de mettre en œuvre les actions nécessaires, mais aussi de maintenir un historique complet des événements dans la salle, garantissant ainsi un suivi optimal et une gestion efficace.

Chapter 2

Description

2.1 Microservice:

Le projet repose sur une architecture en microservices, développée avec Spring Boot, garantissant une collaboration indépendante et efficace entre ses différents composants. Hébergée dans un environnement local (localhost), chaque microservice dispose d'un port unique pour simplifier les tests et permettre une exécution simultanée sur une même machine. L'application comprend un capteur, MovementDetection (port : 8082), deux actionneurs, LightControl (port : 8081) et AlarmControl (port : 8083), ainsi qu'un contrôleur central, Controller (port : 8080), tous conçus pour interagir de manière fluide et cohérente.

2.2 Scenarios

Pour conceptualiser le développement de l'application, nous avons défini des scénarios sous forme de *user stories* afin de mieux répondre aux besoins des utilisateurs.

Scénario 1

En tant qu'étudiant, connaître la disponibilité des salles est essentiel, surtout pendant les périodes de préparation. Savoir en temps réel si une salle est occupée ou libre permet de mieux s'organiser pour étudier ou planifier des sessions de révision de manière efficace.

Scénario 2

Pour renforcer la sécurité d'une salle au GEI contenant du matériel informatique coûteux, il est essentiel de s'assurer qu'elle reste fermée et inaccessible aux personnes non autorisées. Une solution pertinente serait d'introduire un système de sécurité basé sur un détecteur de mouvement. Ce système aurait pour objectif de surveiller en temps réel la présence de toute personne entrant dans les locaux. En cas de détection de mouvement, une alarme se déclencherait instantanément pour signaler une intrusion. Cette approche contribuerait à dissuader les accès non autorisés et à renforcer la protection des équipements sensibles, tout en offrant une solution fiable et efficace pour la sécurité des lieux.

Scénario 3

Pour contribuer à la transition écologique, il est crucial de prendre en compte la consommation énergétique des salles, en particulier étant donné leur nombre élevé. L'installation de

détecteurs de mouvement pour contrôler l'éclairage représente une solution efficace. Cela permet de s'assurer que les lumières s'éteignent automatiquement lorsque les salles sont inoccupées, réduisant ainsi la consommation inutile. De plus, cette approche permet de pallier les oublis des étudiants ou des enseignants en éteignant les lumières de manière automatique.

Chapter 3

Description des Commandes

Controller

- `http://localhost:8080/Controller/isAutolightactivated/`
Permet de vérifier l'état du contrôle automatique des lumières pour déterminer s'il est activé ou désactivé.
- `http://localhost:8080/Controller/setAutolightControl/`
Permet d'activer ou de désactiver le contrôle automatique des lumières.
- `http://localhost:8080/Controller/getRoomUsed/`
Permet de vérifier si une pièce est actuellement occupée grâce à la détection de mouvement.
- `http://localhost:8080/Controller/isAutoAlarmActivated/`
Permet de vérifier l'état du contrôle automatique de l'alarme, indiquant s'il est activé ou désactivé.
- `http://localhost:8080/Controller/setAutoAlarmControl/`
Permet d'activer ou de désactiver le contrôle automatique de l'alarme.
- `http://localhost:8080/Controller/runAuto/`
Déclenche l'exécution des actions automatiques, y compris le contrôle automatique des lumières, le contrôle automatique de l'alarme et la détection de mouvement. Fournit un résumé des actions exécutées dans la réponse.

MovementDetection

- `http://localhost:8082/MovementDetection/getDetection/`
Permet de vérifier l'état actuel de la détection de mouvement.
- `http://localhost:8082/MovementDetection/setDetection/`
Permet d'activer ou de désactiver la détection de mouvement.
- `http://localhost:8082/MovementDetection/getNbDetection/`
Permet de récupérer le nombre de détections de mouvement enregistrées.
- `http://localhost:8082/MovementDetection/setNbDetection/`
Permet de définir le nombre de détections de mouvement dans le système.

LightControl

- <http://localhost:8081/LightControl/isON/>
Permet de vérifier l'état actuel de la lumière (allumée ou éteinte).
- <http://localhost:8081/LightControl/setON/>
Permet de modifier l'état de la lumière (allumée ou éteinte).

AlarmControl

- <http://localhost:8083/AlarmControl/status/>
Permet de vérifier l'état actuel du système d'alarme.
- <http://localhost:8083/AlarmControl/setStatus/>
Permet de définir l'état du système d'alarme.

Chapter 4

Démonstration

Cette section présente une démonstration des microservices du projet, illustrant le contrôle automatique de l'éclairage et des alarmes en réponse à des mouvements détectés. Les utilisateurs peuvent exécuter les applications dans Eclipse IDE pour explorer leurs fonctionnalités et comprendre leur interaction au sein du système.

4.1 Utilisation des salles

Cette fonctionnalité permet de vérifier l'état d'occupation d'une salle grâce à la détection de mouvements.

Détection d'un mouvement

Pour simuler un mouvement, on peut accéder à l'URL suivante :

- <http://localhost:8082/MovementDetection/setDetection/?detection=true>

En cliquant sur ce lien, on est dirigé vers une page confirmant que la détection de mouvement a été activée (voir Annexe 6) et sur Eclipse des messages indiquant le déclenchement de la détection de mouvement devraient apparaître (voir Annexe 6) .

Pour vérifier l'état des salles après la simulation d'un mouvement, on peut accéder à l'URL suivante :

- <http://localhost:8080/Controller/getRoomUsed/>

Le résultat indique que la salle est occupée, confirmant ainsi le mouvement simulé(voir Annexe 6).

Absence de mouvement détectée

Pour simuler l'absence de mouvement, on peut accéder à l'URL suivante : <http://localhost:8082/MovementDetection/setDetection/?detection=false>. En cliquant sur ce lien, on est redirigé vers une page confirmant qu'aucun mouvement n'a été détecté(voir Annexe 6).

Pour examiner l'état des salles en l'absence de mouvement simulé, on peut accéder à l'URL suivante : <http://localhost:8080/Controller/getRoomUsed/> Le résultat de cette URL indique que la salle n'est pas utilisée, reflétant l'absence de mouvement simulé(voir Annexe 6).

4.2 Contrôle de l'alarme

Cette fonctionnalité a pour objectif de déclencher l'alarme en cas de détection de mouvements.

Pas de détection de mouvement

Nous veillons à ce que l'alarme ne se déclenche pas en l'absence de détection de présence dans la salle. On commence par activer le système de contrôle d'alarme automatique en accédant aux liens suivants :

- <http://localhost:8080/Controller/setAutoAlarmControl/?autoAlarmControl=true>
- <http://localhost:8080/Controller/isAutoAlarmControlActivated/>

On simule la non détection du mouvement (voir Annexe 6) :

- <http://localhost:8082/MovementDetection/setDetection/?detection=false>

Pour vérifier si l'alarme est activée, il suffit de consulter son état via le lien suivant :

- <http://localhost:8083/AlarmControl/status>

On constate que l'alarme n'est pas activée (voir Annexe 6).

Détection de mouvement

Maintenant, nous veillons à ce que l'alarme se déclenche si on détecte une présence dans la salle. On commence par simuler la détection de mouvement (voir Annexe 6) :

- <http://localhost:8082/MovementDetection/setDetection/?detection=true>

Pour vérifier si l'alarme est activée, il suffit de consulter son état via le lien suivant :

- <http://localhost:8083/AlarmControl/status>

On voit que l'alarme est déclenchée (voir Annexe 6).

4.3 Contrôle des lumières

Cette fonctionnalité a pour but de réduire la consommation énergétique en contrôlant les lumières, en les éteignant ou en les allumant selon les besoins.

Contrôle automatique des lumières

Pour activer le système de contrôle automatique des lumières dans le contrôleur, on peut accéder à l'URL suivante :

- <http://localhost:8080/Controller/isAutoLightControlActivated/>

Le résultat de cette URL indique que le contrôle automatique des lumières est activé (voir Annexe 6).

Statut de la lumière

Pour vérifier l'état initial de la lumière (en supposant qu'aucun mouvement n'a été détecté), on peut accéder à l'URL suivante :

- <http://localhost:8081/LightControl/isON/>

Le résultat de cette URL indique que la lumière est allumée (voir Annexe 6).

Simulation de la detection de mouvement

Pour simuler la détection de mouvement, on peut accéder à l'URL suivante :

- <http://localhost:8082/MovementDetection/setDetection/?detection=true>

Cette action permet de confirmer que la détection de mouvement est activée (voir Annexe 6).

Par la suite, cela entraîne un changement de l'état de la lumière. Pour vérifier que la lumière est allumée, on peut accéder à l'URL suivante :

- <http://localhost:8081/LightControl/isON/>

Le résultat de cette URL confirme que la lumière est allumée (voir Annexe 6).

4.4 Interaction avec la base de donnée

L'objectif principal est de permettre que les actions réalisées soient directement envoyées à la base de données et affichées en temps réel. Cela offrira un suivi immédiat par l'administration ainsi que par les responsables de la sécurité, leur permettant de visualiser les différentes actions effectuées et de détecter d'éventuelles anomalies, notamment en cas d'infraction aux règles établies par la direction.

Vous trouverez dans l'annexe un exemple illustrant une action réalisée dans le premier scénario, où celle-ci a été directement envoyée à la base de données et affichée en temps réel (voir Annexe 6).

Chapter 5

Conclusion

Ce projet de gestion des salles de l'INSA, basé sur une architecture en microservices, a permis d'explorer les interactions entre les capteurs, les actionneurs et les services logiciels pour répondre à des besoins concrets tels que l'optimisation de l'utilisation des salles, la réduction de la consommation énergétique et le renforcement de la sécurité. L'intégration d'un système centralisé capable de collecter, analyser et exécuter des actions en temps réel a démontré son efficacité dans divers scénarios, offrant une solution innovante et adaptable à des environnements similaires.

En outre, ce projet a également mis en évidence l'importance de l'automatisation et de la centralisation des données pour améliorer la gestion des infrastructures. Il constitue une base solide pour des développements futurs, notamment l'ajout de nouvelles fonctionnalités ou l'intégration avec des technologies émergentes pour rendre le système encore plus performant et respectueux de l'environnement.

Enfin, cette expérience a permis d'acquérir des compétences précieuses en conception et déploiement de microservices, en collaboration entre les composants d'un système distribué et en gestion des interactions avec une base de données en temps réel. Ces acquis seront sans aucun doute bénéfiques pour des projets technologiques ultérieurs.

Chapter 6

Annexes:

Annexe 1 : Confirmation de l'activation de la détection de mouvement



Figure 6.1: Page confirmant que la détection de mouvement est activée.

Annexe 2 : Messages dans Eclipse

```
2025-01-14T12:02:41.270+01:00 INFO 46492 --- [nio-8082-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
2025-01-14T12:02:41.303+01:00 INFO 46492 --- [nio-8082-exec-1] f.i.m.M.MovementDetectionApplication : Détection de mouvement activée
```

Figure 6.2: Messages Eclipse montrant le déclenchement de la détection de mouvement.

Annexe 3 : État des salles après la simulation

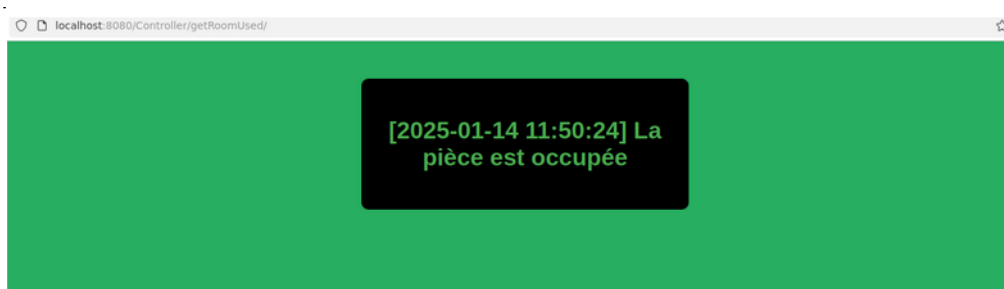


Figure 6.3: Affichage de l'état des salles après la simulation.

Annexe 4 : Confirmation de l'absence de détection de mouvement



Figure 6.4: Page confirmant que la détection de mouvement est désactivée.

Annexe 5 : État des salles en l'absence de mouvement

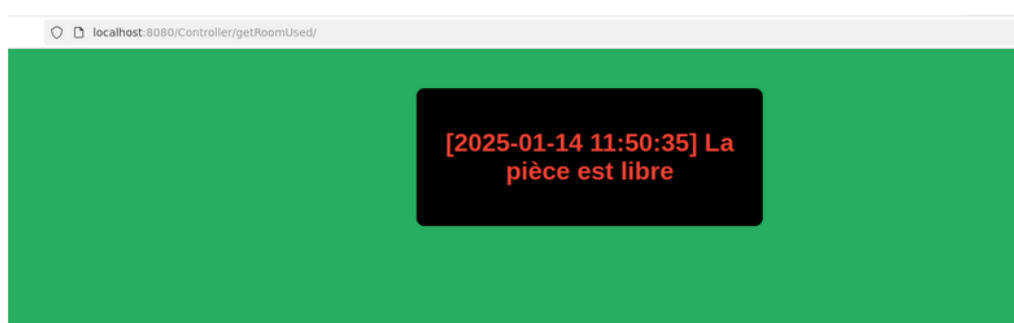


Figure 6.5: Affichage indiquant que la pièce est libre.

Annexe 6 : Simulation de la détection de mouvement désactivée



Figure 6.6: Détection de mouvement désactivée.

Annexe 7 : État de l'alarme sans détection de mouvement



Figure 6.7: L'alarme n'est pas déclenchée en l'absence de détection de mouvement.

Annexe 8 : Simulation de la détection de mouvement activée



Figure 6.8: Détection de mouvement activée.

Annexe 9 : État de l'alarme après détection de mouvement

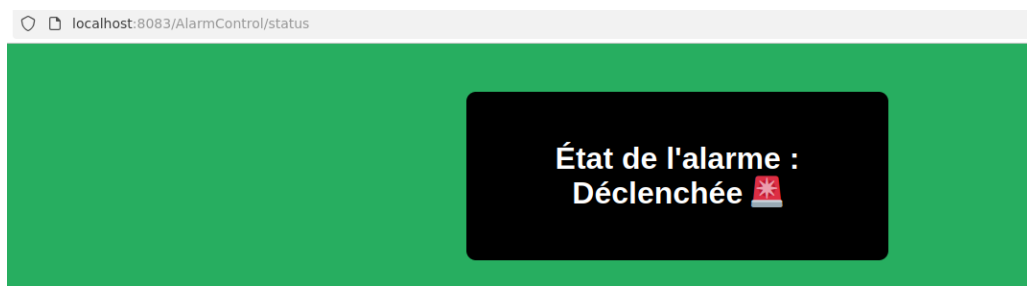


Figure 6.9: L'alarme est déclenchée après la détection de mouvement.

Annexe 10 : Activation du contrôle automatique des lumières

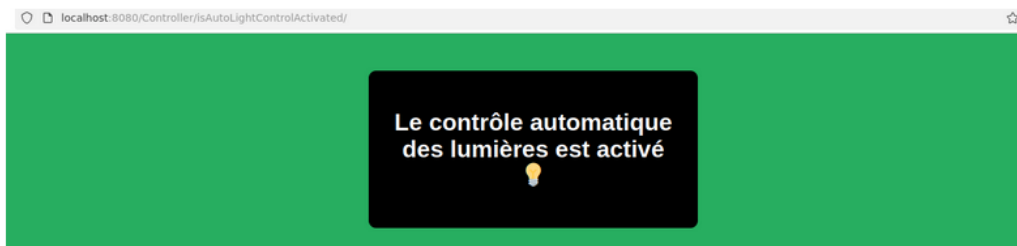


Figure 6.10: Affichage confirmant que le contrôle automatique des lumières est activé.

Annexe 11 : État initial de la lumière

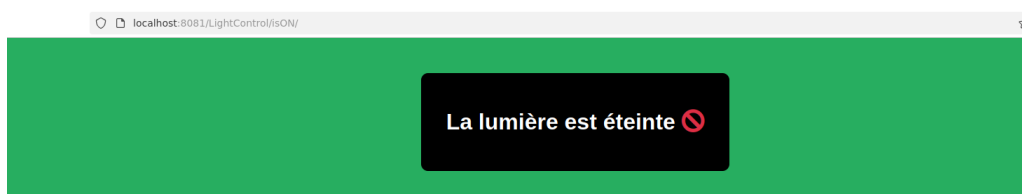


Figure 6.11: Affichage confirmant que la lumière est éteinte.

Annexe 12 : Détection de mouvement activée



Figure 6.12: Affichage confirmant que la détection de mouvement est activée.

Annexe 13 : État de la lumière après la détection de mouvement

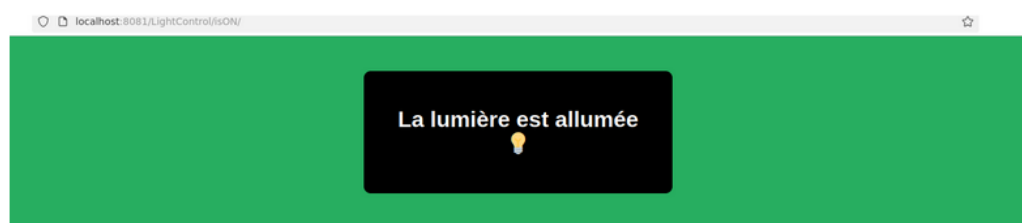


Figure 6.13: Affichage confirmant que la lumière est allumée.

Annexe 14 : Exemple d'action envoyée à la base de données

```
mysql> SELECT * FROM events;
```

id	event_description	event_date
1	La pièce est occupée	2025-01-14 11:50:24

Figure 6.14: Exemple d'une action enregistrée dans la base de données et affichée en temps réel.