



Université Mohammed V  
École nationale supérieure d'informatique et des  
systèmes  
Département de génie logiciel



## Rapport de projet

---

# Développement d'un système automatisé de gestion des absences pour l'ENSIAS

---

Réalisé par :

YASSINE OUHADI

ID LHADJ Ayoub

HOUSNI Badreddine

CHARIFE Mehdi

Encadré par :

M. ELHAMLAOUI Mahmoud

Année universitaire : 2022/2023

” Le succès vient généralement à ceux qui sont trop occupés pour le chercher. ”

Henry David Thoreau

## Dédicace

À Nos chers parents en signe de témoignage d'un grand respect et d'une profonde reconnaissance pour les sacrifices et les efforts qu'ils ont déployés pour notre éducation et notre formation.

À nos chers frères et sœurs.

À tous nos amis et collègues.

À tous nos professeurs.

À toute la famille.

Aucun mot ou expression ne serait valoriser les sacrifices que vous avez consentis pour notre bien-être et notre études.



## Remerciements

Nous tenons à exprimer nos sincères remerciements à notre professeur encadrant, M. ELHAMLAOUI Mahmoud, pour son soutien et son accompagnement tout au long de ce projet. Sa disponibilité, ses conseils et ses encouragements ont été précieux pour mener à bien cette réalisation.

Nous souhaitons également remercier l'ensemble de l'équipe pédagogique du département de Génie Logiciel de l'ENSIAS pour la qualité de leur enseignement, leur expertise et leur disponibilité.

Enfin, nous tenons à exprimer notre gratitude envers toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet, notamment nos camarades de groupe, ainsi que nos amis et notre famille pour leur soutien et leur encouragement.

## **Résumé**

Dans le cadre de notre projet de développement, nous avons proposé une solution innovante pour la gestion des absences à l'ENSIAS en utilisant le système RFID. Nous avons présenté les différentes étapes de la réalisation de ce projet, de la conception à l'implémentation de l'application web de gestion des absences et du système RFID.

# Table des matières

Introduction générale . . . . .	VIII
<b>Chapitre 1: Contexte du projet</b>	<b>1</b>
1.1 Présentation de la mission du projet . . . . .	1
1.1.1 Problématique . . . . .	1
1.1.2 Objectif du projet . . . . .	2
1.1.3 Planification de projet . . . . .	2
1.2 Méthodologie suivie . . . . .	4
1.2.1 La méthodologie Agile . . . . .	4
1.2.2 Principes de l'agilité . . . . .	4
1.3 Conclusion . . . . .	4
<b>Chapitre 2: Mise en place du système IoT</b>	<b>1</b>
2.1 Introduction . . . . .	1
2.2 Introduction au système RFID . . . . .	1
2.3 Choix du matériel . . . . .	1
2.4 Conception du système . . . . .	3
2.5 Intégration du système . . . . .	3
2.6 Conclusion . . . . .	4
<b>Chapitre 3: Analyse et conception</b>	<b>1</b>
3.1 Introduction . . . . .	1
3.2 Cahier des charges . . . . .	1
3.2.1 Besoins fonctionnels . . . . .	1
3.2.2 Besoins non fonctionnels . . . . .	2

---

3.3	Conception . . . . .	3
3.3.1	L'analyse UML . . . . .	3
3.4	Conclusion . . . . .	7
<b>Chapitre 4:</b>	<b>Réalisation</b>	<b>1</b>
4.1	Introduction . . . . .	1
4.2	Environnement du travail . . . . .	1
4.2.1	Choix des outils de développement . . . . .	3
4.3	Architecture du Back-end . . . . .	4
4.4	Interfaces Graphiques . . . . .	5
4.5	Déploiment avec Docker . . . . .	8
4.6	Conclusion . . . . .	9
	Conclusion . . . . .	10

## List of Figures

1.1	Diagramme de Gantt . . . . .	3
2.1	MFRC522 RFID Module . . . . .	2
2.2	Badge RFID ISO14443A . . . . .	2
2.3	Arduino UNO . . . . .	2
2.4	Arduino IDE . . . . .	3
2.5	Carte WiFi . . . . .	3
2.6	PostgreSQL . . . . .	3
3.1	Diagramme des cas d'utilisation d'un user . . . . .	3
3.2	Diagramme des cas d'utilisation d'administration . . . . .	4
3.3	Diagramme des cas d'utilisation de professeur . . . . .	5
3.4	Diagramme des cas d'utilisation d'étudiant . . . . .	6
3.5	Diagramme de classes . . . . .	7
4.1	Edraw Max . . . . .	1
4.2	IntelliJ IDEA . . . . .	1
4.3	PostgreSQL . . . . .	2
4.4	Postman . . . . .	2
4.5	Git . . . . .	2
4.6	Visual Studio Code . . . . .	3
4.7	Angular . . . . .	3
4.8	Spring Boot . . . . .	3
4.9	REST API . . . . .	4



---

4.10 Interface login . . . . .	5
4.11 Interface add studentl . . . . .	5
4.12 Interface add student(suite)l . . . . .	6
4.13 Interface delete studentl . . . . .	6
4.14 Interface add Professorl . . . . .	7
4.15 Interface add seance . . . . .	7
4.16 Interface add seance(suite)l . . . . .	8
4.17 Docker . . . . .	8

## **Introduction générale**

Notre projet de développement s'intéresse à la gestion des absences à l'ENSIAS en utilisant une technologie innovante, le système RFID. Ce projet a été réalisé dans le cadre du module "Plateformes de développement" du semestre 4 de la filière Génie Logiciel à l'ENSIAS. L'objectif de ce projet est de proposer une solution efficace pour la gestion des absences des étudiants et des enseignants dans l'établissement. Dans ce rapport, nous présenterons les différentes étapes de la réalisation de ce projet, de la conception à l'implémentation de l'application web de gestion des absences et du système RFID.

# Chapitre 1

## Contexte du projet

Dans ce chapitre, je commencerai tout d'abord par la présentation de la problématique et le contexte général de mon projet ainsi que la méthodologie suivie pour trouver une solution à la problématique.

### 1.1 Présentation de la mission du projet

Dans le contexte de la gestion des absences dans une école, la gestion manuelle des absences des étudiants peut causer des retards dans la prise de décisions et des erreurs dans les données. De plus, les enseignants et les administrateurs de l'école passent beaucoup de temps à collecter et à saisir les données des absences, ce qui peut entraîner une perte de productivité et des coûts supplémentaires. La solution actuelle pour la gestion des absences est basée sur des feuilles de présence manuelles, ce qui peut rendre la tâche fastidieuse et peu fiable.

#### 1.1.1 Problématique



La problématique du projet est donc de trouver une solution automatisée pour la gestion des absences des étudiants de l'ENSIAS, qui peut permettre de collecter et de saisir les données d'absence de manière plus rapide et plus fiable. Cette solution devrait également être facilement accessible par les enseignants et les administrateurs de l'école. Le projet doit également garantir la sécurité des données sensibles des étudiants et être conforme aux réglementations de confidentialité en vigueur.

### 1.1.2 Objectif du projet

Notre projet vise à développer un système de gestion d'absences pour l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes. L'objectif principal de ce projet est de mettre en place une solution informatique fiable et efficace pour gérer les absences des étudiants et des enseignants, en remplaçant le système manuel actuel qui est source d'erreurs et de pertes de temps. Nous avons identifié les objectifs spécifiques suivants pour atteindre notre objectif principal :

1. Concevoir et développer un système de lecture RFID pour enregistrer la présence des étudiants et des enseignants.
2. Intégrer le système RFID avec une base de données cloud Postgres pour stocker et gérer les données de présence.
3. Développer une application Web pour permettre aux enseignants et aux étudiants de consulter et de gérer leurs absences en temps réel.
4. Améliorer l'expérience utilisateur en offrant une interface conviviale et intuitive pour l'application Web.
5. Assurer la sécurité des données en mettant en place des mécanismes de protection des données et des mesures de sécurité efficaces.

Ces objectifs nous permettront de fournir une solution complète et performante pour la gestion des absences à ENSIAS, tout en améliorant l'efficacité opérationnelle et en offrant une expérience utilisateur optimale pour les enseignants et les étudiants.

Enfin, pour garantir le bon déroulement du projet et respecter les délais, nous avons élaboré une planification globale de conduite du projet, en utilisant un diagramme de Gantt qui nous a permis de hiérarchiser les tâches et de quantifier les délais d'exécution à chaque étape du projet.

### 1.1.3 Planification de projet

La conception et la réalisation d'une solution répondant à l'ensemble des besoins imposés par la problématique constituaient la principale motivation de ce projet. Il s'agissait de concevoir une solution propriétaire, afin de nous donner la liberté de la modifier, l'optimiser et la personnaliser à notre guise.

Comme pour tout projet, une phase de planification s'avère nécessaire pour en assurer le bon déroulement. Cela permet de définir les tâches à réaliser, de maîtriser les risques et de rendre compte de l'état d'avancement du projet. Afin de garantir le respect des délais impartis, nous avons élaboré une planification globale de conduite du projet.

Cette planification a été représentée sous la forme d'un diagramme de Gantt, qui décrit l'ordonnancement prévu des différentes phases du projet.

Nous avons planifié le déroulement des différentes étapes de la manière suivante :

1. Collecte des besoins et analyse : Cette étape nous a permis d'identifier les besoins et les attentes de chaque partie, ainsi que d'analyser les processus existants.
2. Conception de la solution : Sur la base des résultats d'analyse, nous avons conçu une solution complète de gestion des absences qui prend en compte à la fois l'application web et le système IoT de lectures RFID.
3. Développement et test de l'application web : Nous avons commencé le développement de l'application web en utilisant les technologies et les outils les plus adaptés. Cette phase a inclus la programmation, les tests unitaires et les tests d'intégration pour garantir la qualité et la fiabilité de l'application.
4. Développement et test du système IoT de lectures RFID : Nous avons également réalisé des tests unitaires et des tests d'intégration pour assurer la qualité et la fiabilité du système.
5. Intégration de l'application web et du système IoT : Nous avons intégré l'application web et le système IoT de lectures RFID en utilisant une approche de développement agile. Nous avons testé l'ensemble de la solution pour garantir son bon fonctionnement.
6. Formation et déploiement en utilisant Docker : Nous avons également développé un environnement de conteneurisation Docker pour faciliter le déploiement et la gestion de la solution sur différents environnements.
7. Maintenance et support : Après le déploiement, nous avons assuré la maintenance continue de la solution pour corriger les erreurs et les bogues et pour ajouter de nouvelles fonctionnalités si nécessaire.

Le diagramme de Gantt est un outil utilisé en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet.

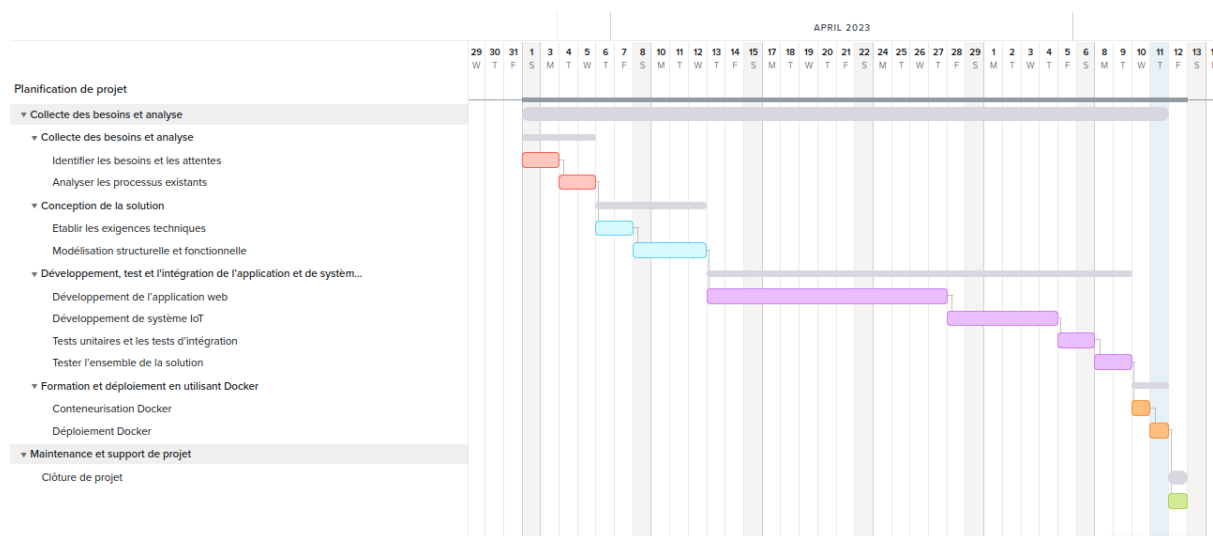


FIGURE 1.1 – Diagramme de Gantt

## 1.2 Méthodologie suivie

Dans cette partie, nous allons aborder la méthodologie de travail mise en place lors du projet.

### 1.2.1 La méthodologie Agile

Pour mener à bien ce projet, nous avons choisi de mettre en place une méthode de travail agile, qui nous permet de travailler de manière itérative, avec une grande flexibilité pour adapter le projet aux changements éventuels. Cette méthode nous a permis de travailler efficacement en équipe et de synchroniser notre travail en utilisant des outils tels que GitHub, Git et JIRA, ce qui nous a permis de collaborer à distance en toute transparence. Grâce à cette approche, nous avons pu répondre rapidement aux besoins du projet et garantir sa réussite. Mon encadrant est impliqué dans le projet du début à la fin ce qui lui donne de la visibilité. Cela permet d'éviter « l'effet tunnel », c'est-à-dire de se lancer dans un projet et arriver à terme avec un produit qui ne correspond pas aux attentes des clients (les encadrants dans mon cas) ou ne pas mener le projet à terme.

### 1.2.2 Principes de l'agilité

Bien que le projet ait été effectué en équipe, plusieurs principes de l'agilité ont été respectés, notamment les suivants :

1. La concentration sur la qualité du développement logiciel, en veillant à ce que chaque fonctionnalité ajoutée soit testée de manière approfondie avant d'être mise en production.
2. L'adaptation au changement en restant ouvert aux retours des utilisateurs et en ajustant les fonctionnalités en conséquence, tout en gardant un plan général du projet.
3. La collaboration étroite entre les membres de l'équipe de développement et les utilisateurs finaux de l'application, pour garantir que les besoins de ces derniers sont pris en compte dans chaque itération du développement.
4. L'utilisation de méthodes de développement Agile telles que Scrum pour planifier et suivre le travail, en veillant à ce que l'équipe reste alignée sur les objectifs globaux du projet.

## 1.3 Conclusion

Après avoir présenté le contexte général du projet, nous allons mettre en exergue l'analyse du cahier de charge et la conception dans le chapitre suivant.

# Chapitre 2

## Mise en place du système IoT

### 2.1 Introduction

Dans ce chapitre, nous allons nous concentrer sur la mise en place du système IoT pour notre projet de gestion des absences. Nous allons expliquer en détail comment nous avons conçu et mis en œuvre le système RFID pour permettre la gestion des absences et des retards. Nous allons également expliquer comment le système IoT fonctionne en relation avec l'application web, et comment il peut améliorer l'efficacité de la gestion des absences et des retards.

### 2.2 Introduction au système RFID

La technologie RFID (Radio Frequency Identification) est une technologie d'identification automatique qui permet de lire à distance des données stockées sur une puce électronique sans contact direct entre la puce et le lecteur. Elle utilise des ondes radio pour échanger des données entre un lecteur et une étiquette ou un badge RFID.

Dans le contexte de notre projet de gestion des absences, nous avons choisi d'utiliser la technologie RFID pour identifier les étudiants et les enseignants à leur arrivée dans les salles de cours. Chaque étudiant et enseignant dispose d'un badge RFID contenant un code unique qui permet d'identifier son propriétaire. Lorsqu'ils passent devant un lecteur RFID placé à l'entrée de la salle de cours, le code unique de leur badge est lu et envoyé au système de gestion des absences.

Le système RFID permet ainsi une identification rapide et fiable des étudiants et des enseignants, et permet de suivre leur présence en temps réel dans les salles de cours.

### 2.3 Choix du matériel

Dans cette partie, nous allons décrire le matériel que nous avons choisi pour mettre en place notre système RFID. Nous avons opté pour les composants suivants :

- Des lecteurs RFID

ces dispositifs sont utilisés pour lire les informations stockées sur les badges RFID. Nous avons choisi d'utiliser des cartes de capture RFID compatibles avec les lecteurs RC522.

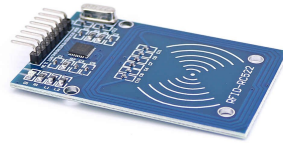


FIGURE 2.1 – MFRC522 RFID Module

- Des badges RFID

chaque étudiant et professeur dispose d'un badge RFID unique qui contient son code d'identification. Nous avons choisi des badges RFID de type ISO14443A



FIGURE 2.2 – Badge RFID ISO14443A

- Des microcontrôleurs

nous avons choisi d'utiliser des cartes Arduino Uno pour programmer les fonctionnalités de notre système RFID.

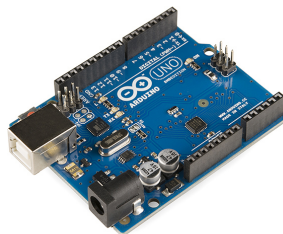


FIGURE 2.3 – Arduino UNO

En plus de ces éléments, nous avons également utilisé d'autres composants tels que des câbles mâles et femelles, une carte WiFi, un écran LCD, des LED rouges et vertes, des résistances et un buzzer pour créer notre système RFID.



## 2.4 Conception du système

La conception du système RFID pour notre projet de gestion des absences a nécessité plusieurs étapes. Tout d'abord, nous avons conçu un schéma de câblage pour connecter les différents composants, notamment les lecteurs RFID, les microcontrôleurs, l'écran LCD, les LED et le buzzer. Nous avons utilisé des câbles mâles et femelles pour faciliter les connexions.



FIGURE 2.4 – Arduino IDE

Ensuite, nous avons choisi les logiciels nécessaires pour notre projet, notamment l'IDE Arduino pour programmer les microcontrôleurs et la bibliothèque MFRC522 pour la gestion des lecteurs RFID. Nous avons également utilisé une carte WiFi pour connecter le système à notre application web de gestion des absences.



FIGURE 2.5 – Carte WiFi

Grâce à cette conception, nous avons pu développer un système RFID efficace et fiable pour la gestion des absences.

## 2.5 Intégration du système

L'intégration du système RFID à l'application de gestion des absences a été un élément clé pour assurer le succès du projet. Pour lier le système RFID à l'application, nous avons utilisé une base de données relationnelle hébergée sur AWS Cloud, à savoir PostgreSQL. Les données collectées par les lecteurs RFID ont été stockées dans base de données en temps réel.



FIGURE 2.6 – PostgreSQL

Nous avons développé une API RESTful pour communiquer avec la base de données PostgreSQL, qui permet à l'application de récupérer les données de présence des étudiants et des enseignants, ainsi que les informations sur les séances de cours programmées.

En utilisant la base de données relationnelle, nous avons pu lier le système RFID à l'application de manière transparente et efficace. Cette approche a permis d'éviter les problèmes de conflit de données et de garantir que toutes les informations pertinentes étaient disponibles en temps réel pour les utilisateurs de l'application.

## **2.6 Conclusion**

Après la mise en place du système RFID nous allons entamer la phase de la réalisation de l'application web de gestion des absences. Il abordera la conception de l'interface utilisateur, la mise en place de la base de données, le développement des fonctionnalités de l'application, ainsi que les tests et la validation de l'application.

# Chapitre 3

## Analyse et conception

### 3.1 Introduction

La conception est une étape primordiale dans le cycle de vie d'une application, elle a pour objectif de faire l'étude des données et des traitements à effectuer. C'est en général dans cette phase que s'appliquent les techniques de modélisation. Je commencerai d'abord par l'analyse du sujet puis je passerai à la conception.

### 3.2 Cahier des charges

Dans cette partie du rapport de notre projet, nous allons décrire en détail les besoins et les attentes, ainsi que les exigences et les fonctionnalités spécifiques du projet.

#### 3.2.1 Besoins fonctionnels

Après l'étude du système, cette partie est réservée à la description des exigences fonctionnelles des utilisateurs : administration, professeurs, chefs de filière et étudiants. Ces besoins se regroupent dans les diagrammes des cas d'utilisation.

##### **Besoins fonctionnels pour le projet**

- Collecter les données de présence des étudiants et des professeurs à partir des scanners RFID.
- Stocker les données de présence collectées dans une base de données en cloud.
- Analyser les données de présence pour générer des statistiques sur l'assiduité des étudiants et des professeurs.
- Permettre aux étudiants et aux professeurs de visualiser leurs propres données de présence.

**Besoins fonctionnels pour le système RFID**

- Chaque étudiant et professeur doit avoir une clé ou un badge contenant son propre code RFID.
- Chaque salle a une boîte de scan utilisant Arduino, serrurée RFID pour scanner les badges à l'entrée.
- Manager l'entrée des étudiants et professeurs, les absences et les retards.

**Besoins fonctionnels pour l'application web**

- Chaque utilisateur possède un compte et doit s'activer afin d'accéder à l'application. Il peut s'inscrire s'il n'a pas de compte.
- Un utilisateur peut récupérer son compte et changer de mot de passe.
- Un utilisateur admin peut gérer les étudiants, professeurs, les branches, les groupes pour chaque année universitaire.
- Un utilisateur admin peut programmer des séances sur une semestre en attribuant un élément, une salle et un professeur et des étudiants.
- Un utilisateur professeur peut valider la présence des étudiants qui ont scanné leurs badges lors de la séance en cours.
- Le système doit valider automatiquement la présence des étudiants qui n'ont pas été validés ou refusés par le professeur.
- L'utilisateur étudiant et professeur peut justifier ses absences et ajouter des supports tels que des certificats médicaux.
- L'utilisateur admin peut accepter les justifications des absences.
- L'admin peut envoyer des alertes à des étudiants et professeurs qui ont plusieurs absences et retards.
- Le chef de filière peut consulter la liste des absences, des statistiques, etc.

**3.2.2 Besoins non fonctionnels**

Les exigences « non fonctionnelles » sont celles qui sont importantes voire critiques aux yeux des utilisateurs et qui assurent le bon fonctionnement du système.

On peut les répartir par catégories, notamment :

- La disponibilité : La plateforme doit être disponible pour être utilisée par n'importe quel utilisateur.
- La sécurité : gestion de la confidentialité, gestion de l'intégrité des données, protection des données personnelles ...
- La convivialité de l'interface graphique : l'application doit fournir une interface conviviale et simple pour l'utilisateur.

- **La fiabilité** : Les données fournies par l'application doivent être fiables. La possibilité de retourner au page d'accueil de la plateforme à partir de n'importe quelle page de celle-ci. Le temps de réponse ne doit pas dépasser cinq secondes.
- **La performance** : Le système doit réagir dans un délai précis, quelle que soit l'action de l'utilisateur.
- **L'évolutivité** : Le système doit être « scalable » avec la possibilité d'ajout de fonctionnalité à tout temps.

## 3.3 Conception

### 3.3.1 L'analyse UML

Nous allons maintenant vous dire quelques mots sur l'analyse UML (Unified Modeling Language) de notre application mais auparavant, nous vous parlerai d'UML. Ce dernier est un langage graphique qui permet de représenter de manière claire et précise, sous forme de modèle objet des applications ou des composants logiciels. En tant que tel, il facilite la création et la compréhension des logiciels actuels.

Dans cette partie et plus exactement pour l'analyse de notre projet, nous présenterons le diagramme de classes, le diagramme d'activités et le diagramme des Uses Cases.

En effet, ce dernier possède un atout majeur puisqu'il présente tous les use cases, autrement dit les grandes lignes du projet. C'est ainsi que nous nous appuierons sur ce diagramme afin de vous illustrer comment nous avons abordé ce projet.

#### 3.3.1.1 Le diagramme des Uses Cases

Le diagramme des cas d'utilisation constitue la première étape de l'analyse UML.

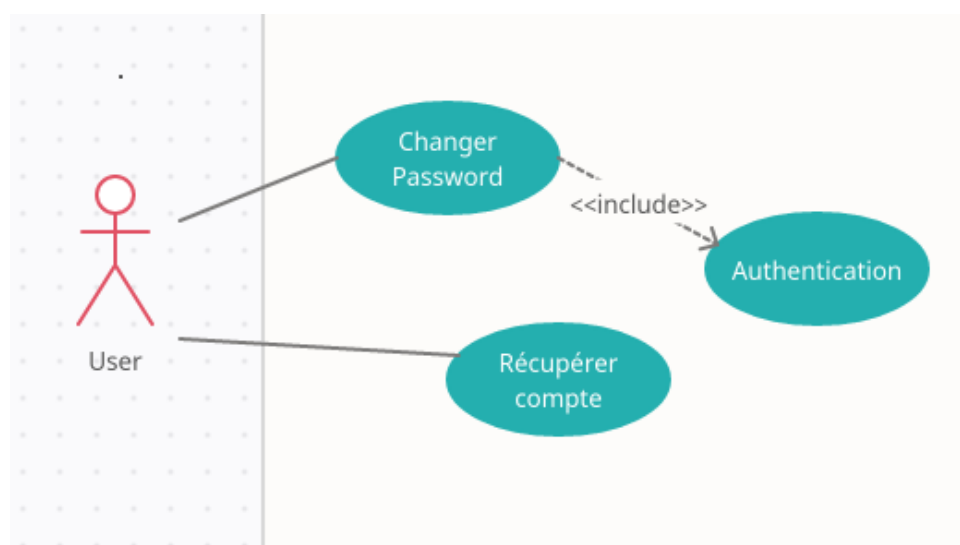


FIGURE 3.1 – Diagramme des cas d'utilisation d'un user

## Diagramme des cas d'utilisation d'administration

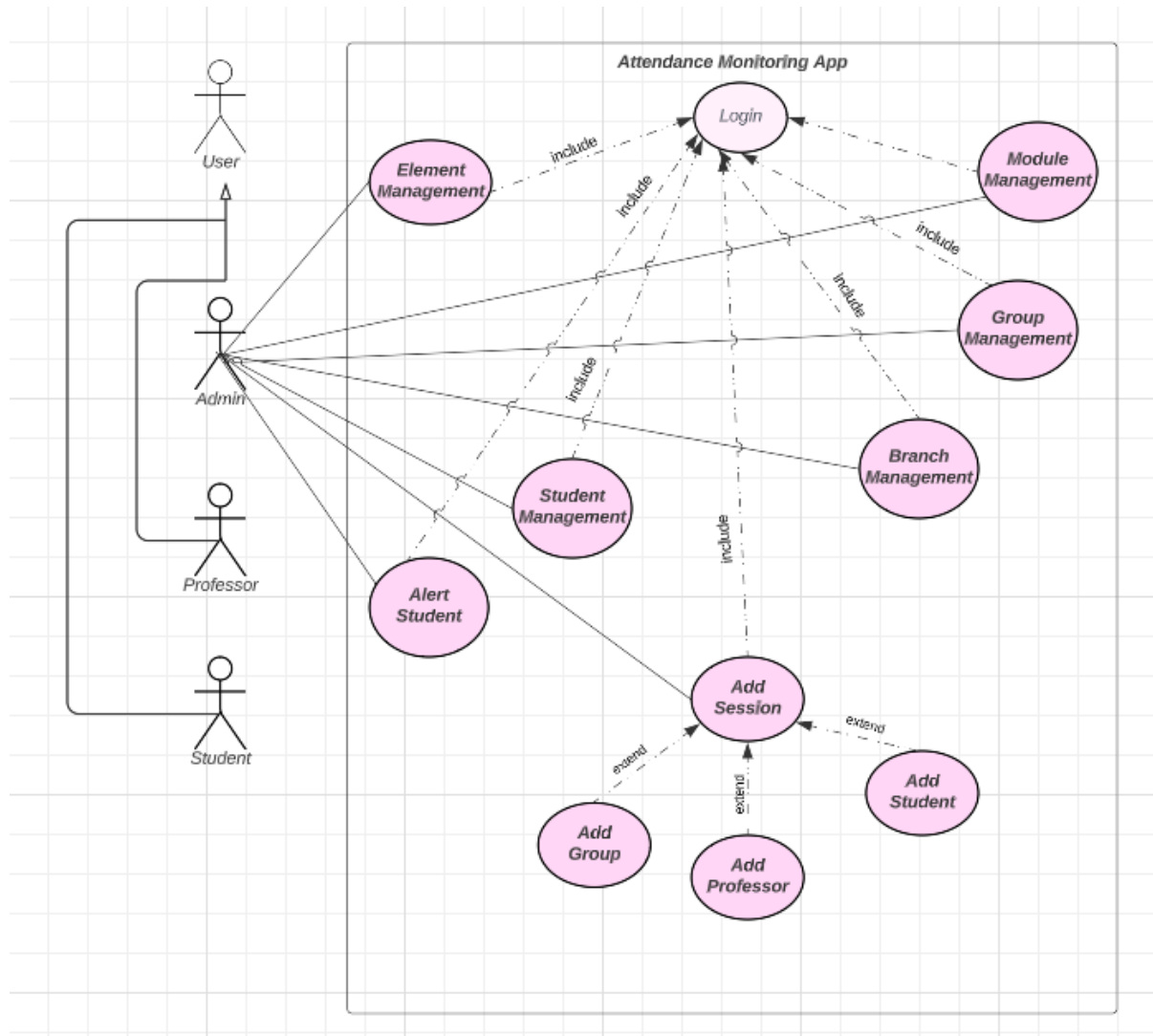


FIGURE 3.2 – Diagramme des cas d'utilisation d'administration

## Diagramme des cas d'utilisation de professeur

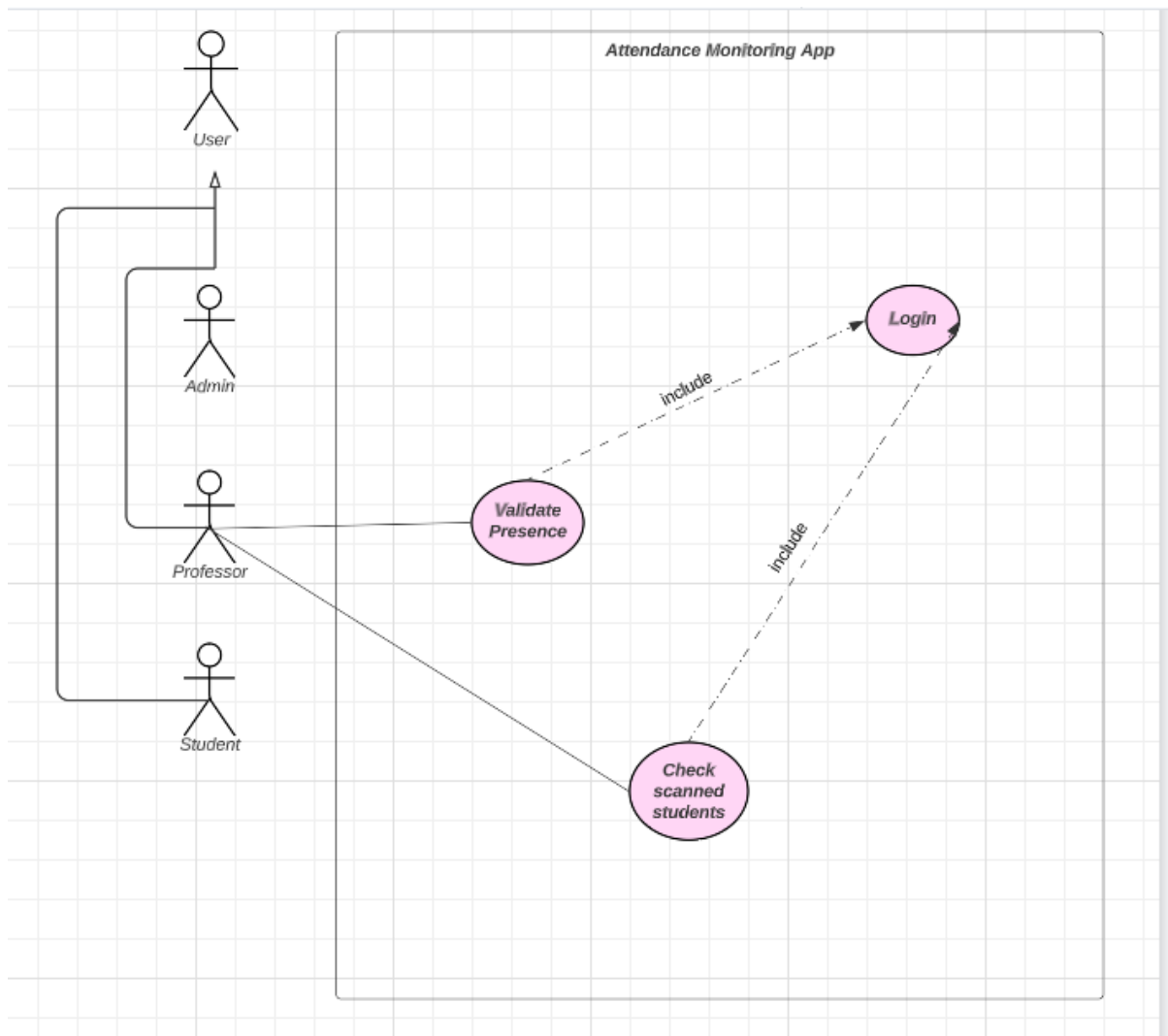


FIGURE 3.3 – Diagramme des cas d'utilisation de professeur

## Diagramme des cas d'utilisation d'étudiant

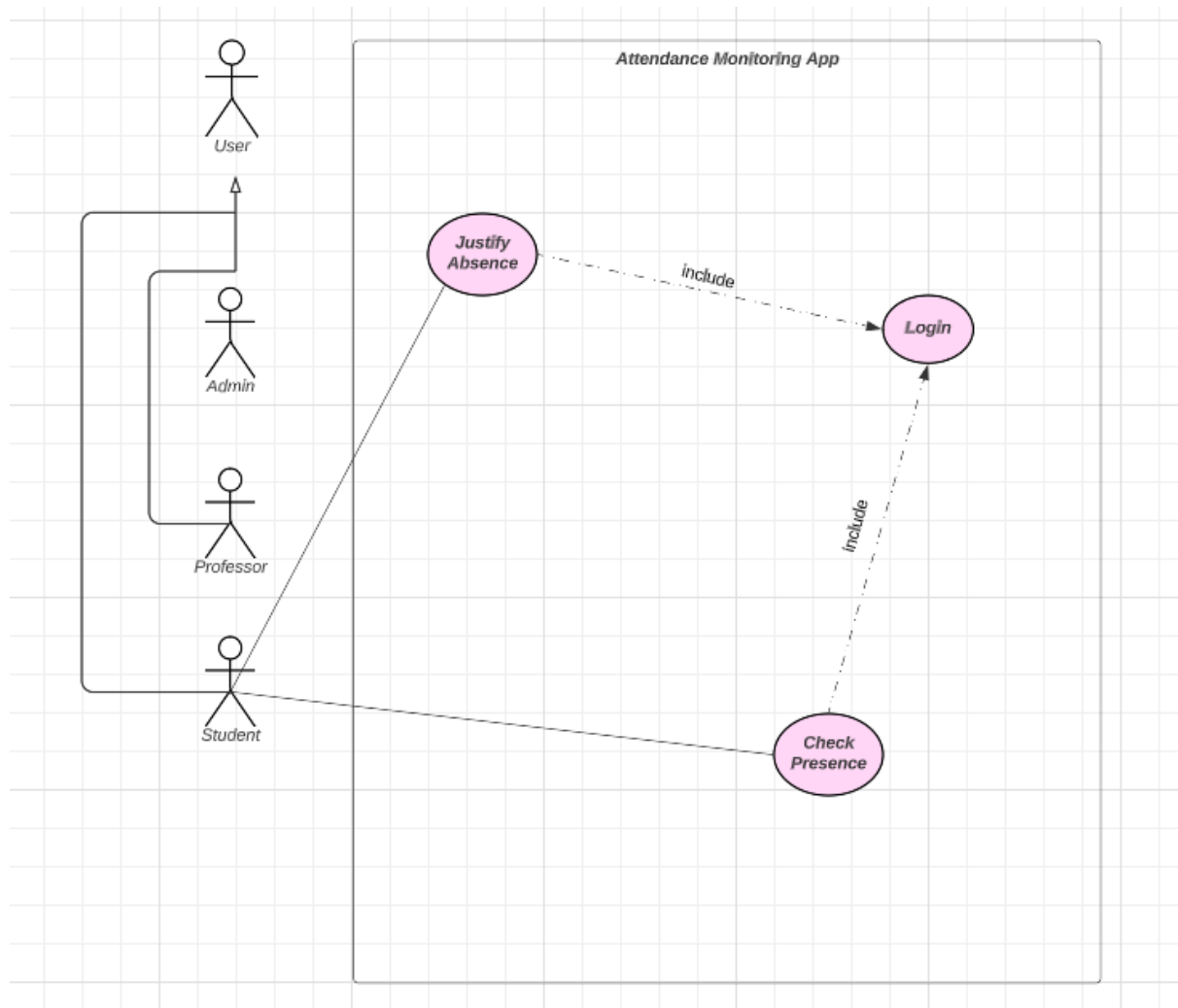


FIGURE 3.4 – Diagramme des cas d'utilisation d'étudiant



### 3.3.1.2 Le diagramme de classes

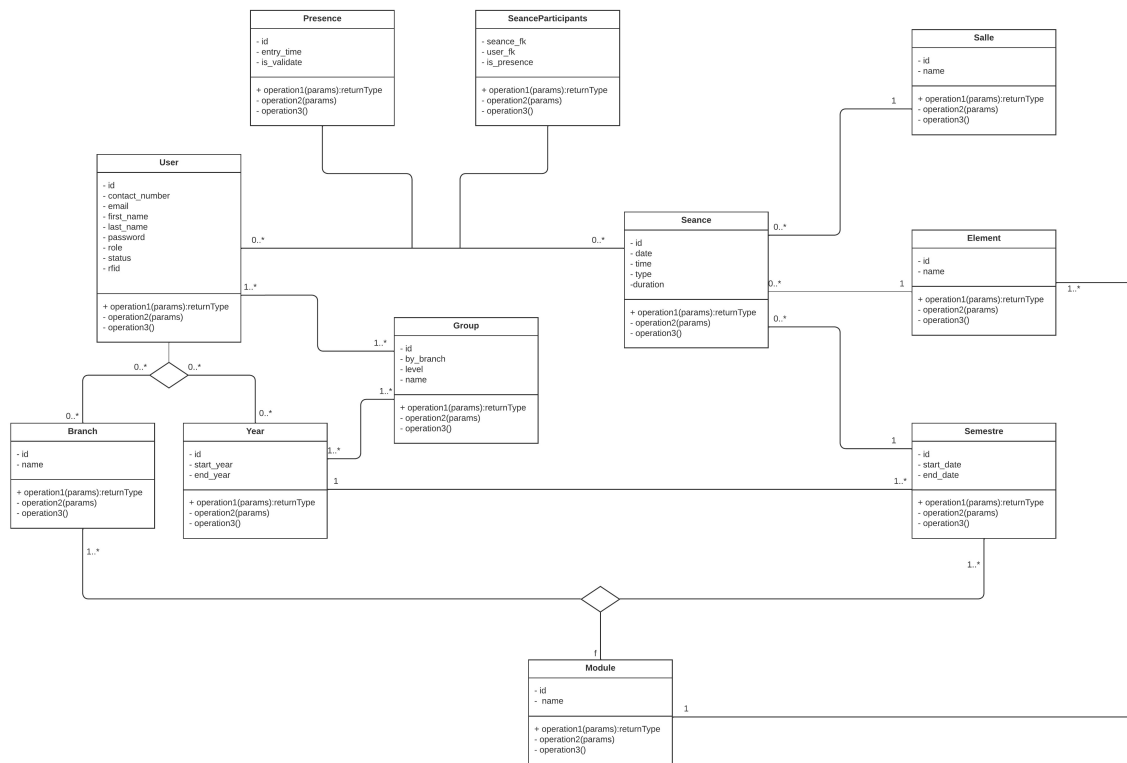


FIGURE 3.5 – Diagramme de classes

## 3.4 Conclusion

Dans ce chapitre, nous allons aborder le cahier des charges, ainsi que la modélisation des données et des traitements en utilisant la l'analyse UML, qui nous permettent de donner une vision bien détaillée sur le fonctionnement du système. À présent, nous allons entamer la phase de réalisation.

# Chapitre 4

## Réalisation

### 4.1 Introduction

Au niveau de cette dernière partie, nous nous intéressons aux outils de développement utilisés pour la réalisation de l'application, ainsi qu'aux principales interfaces de l'application et à quelques scénarios applicatifs.

### 4.2 Environnement du travail

EDRAW MAX



FIGURE 4.1 – Edraw Max

Nous avons utilisé Edraw Max pour créer diagrammes UML pour la conception.

INTELLIJ IDEA



FIGURE 4.2 – IntelliJ IDEA

Nous avons utilisé IntelliJ IDEA pour la création de l'API REST Spring Boot.

## POSTGRESQL



FIGURE 4.3 – PostgreSQL

Nous avons choisi d'utiliser PostgreSQL comme système de gestion de base de données relationnelle pour notre application de gestion des absences, ainsi que pour stocker les données collectées par le système RFID

## POSTMAN



FIGURE 4.4 – Postman

Nous avons utilisé Postman pour tester les API que nous avons créées à l'aide de Spring Boot.

## GIT



FIGURE 4.5 – Git

Nous avez utilisé Git pour synchroniser le travail en équipe.

## VISUAL STUDIO CODE

Nous avons utilisé Visual Studio pour la programmation en Angular.



FIGURE 4.6 – Visual Studio Code

## 4.2.1 Choix des outils de développement

### 4.2.1.1 Front End

ANGULAR



FIGURE 4.7 – Angular

Angular est un framework open-source de développement web pour la création d'applications côté client. Il permet de développer des applications web dynamiques, modulaires et maintenables en utilisant des composants, des services, des directives et des pipes.

### 4.2.1.2 Back End

SPRING BOOT



FIGURE 4.8 – Spring Boot

Spring Boot est un framework open source pour le développement de microservices en Java. Il permet de créer rapidement et facilement des applications Java autonomes et productives. Nous avons utilisé Spring Boot pour développer une API RESTful efficace pour le back-end de notre application. En outre, pour garantir la sécurité de notre application, nous avons

implémenté plusieurs outils tels que Spring Security pour assurer la sécurité, JWT pour l'authentification des utilisateurs et JPA pour la persistance des données. De plus, nous avons implémenté un système de validation automatique des présences à l'aide de Scheduled Jobs.

### 4.3 Architecture du Back-end

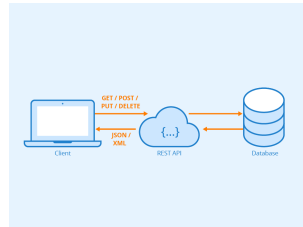


FIGURE 4.9 – REST API

Les services web conformes au style d'architecture REST, établissent une interopérabilité entre les ordinateurs sur Internet. Les services web REST permettent aux systèmes effectuant des requêtes de manipuler des ressources web via leurs représentations textuelles à travers un ensemble d'opérations uniformes et prédéfinies et sans état.

## 4.4 Interfaces Graphiques

Pour notre projet, nous sommes intéressés à fournir des interfaces utilisateur conviviales, attrayantes et simples. Concernant cette dernière partie, nous allons présenter différentes interfaces de notre application.

### INTERFACE LOGIN

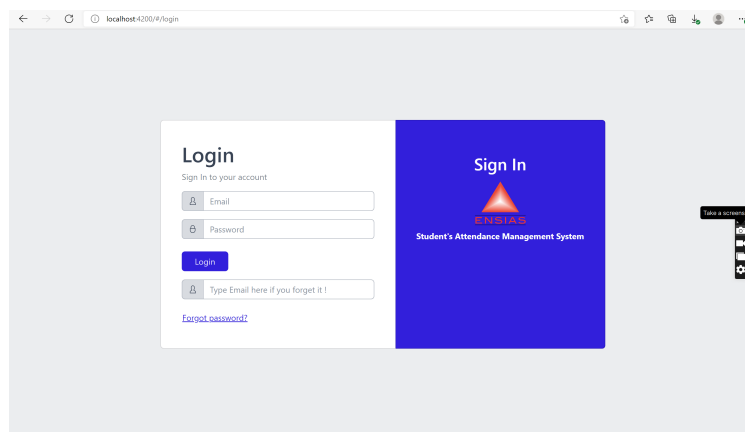


FIGURE 4.10 – Interface login

### INTERFACE ADD STUDENT

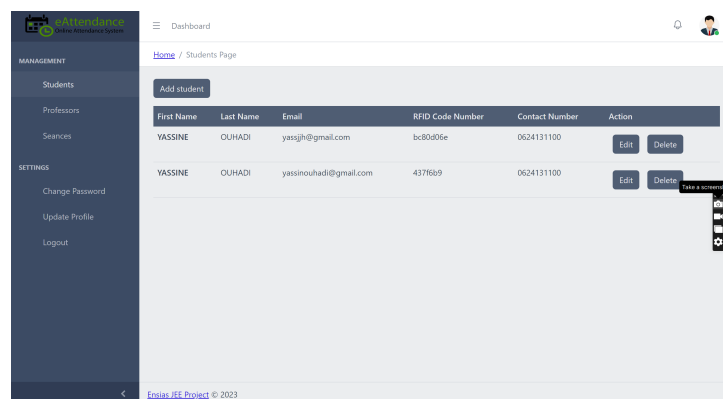


FIGURE 4.11 – Interface add studentl

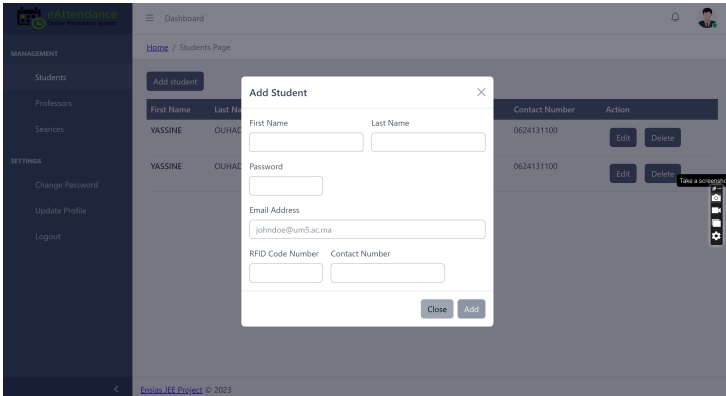


FIGURE 4.12 – Interface add student(suite)l

INTERFACE DELETE STUDENT

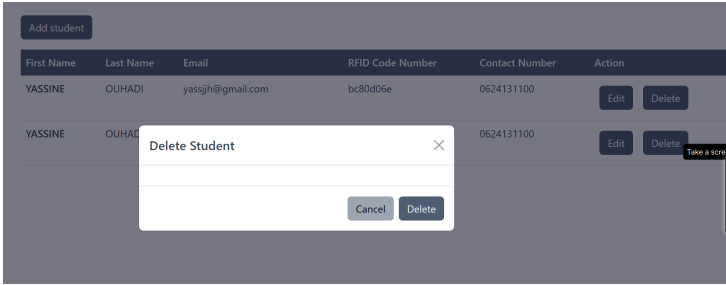


FIGURE 4.13 – Interface delete studentl

INTERFACE ADD PROFESSOR

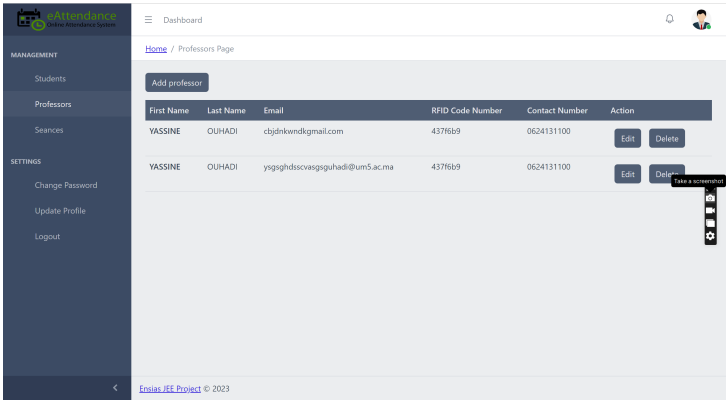


FIGURE 4.14 – Interface add Professorl

INTERFACE ADD SEANCE

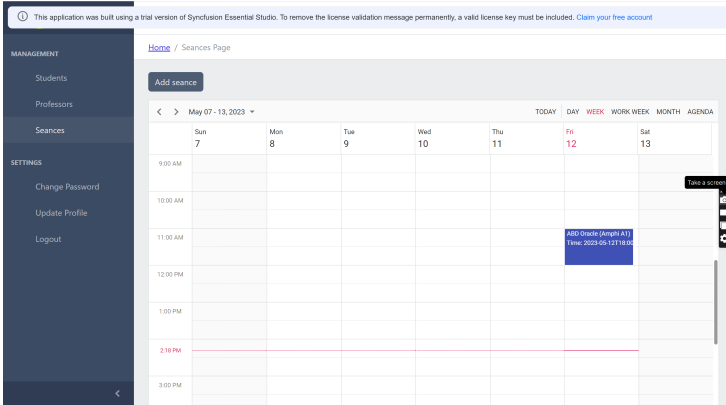


FIGURE 4.15 – Interface add seance



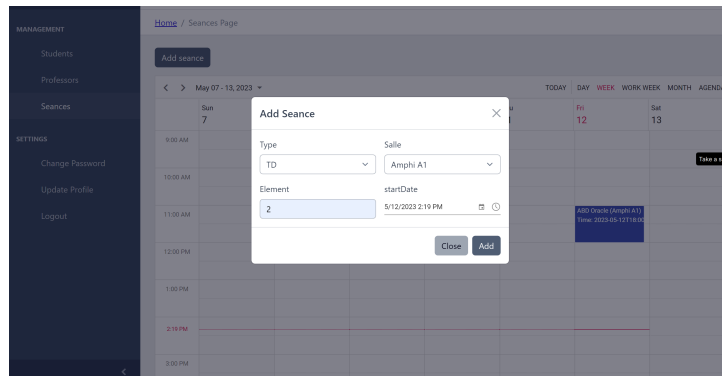


FIGURE 4.16 – Interface add seance(suite)

## 4.5 Déploiment avec Docker

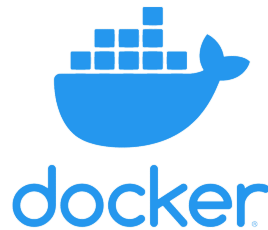


FIGURE 4.17 – Docker

Le déploiement de notre projet a été réalisé avec Docker, une plateforme de conteneurisation qui permet de créer, déployer et exécuter des applications dans des conteneurs. Nous avons créé des images Docker pour notre application Angular et notre application Spring Boot, puis nous les avons regroupées dans un environnement de déploiement en utilisant Docker Compose. Cela nous a permis de simplifier le processus de déploiement, de garantir la portabilité de notre application et de faciliter sa mise à l'échelle en cas de besoin. De plus, en utilisant Docker, nous avons pu isoler notre application et son environnement de production, ce qui a renforcé la sécurité et la stabilité de notre application.

## 4.6 Conclusion

Dans ce chapitre, nous avons montré et détaillé l'architecture et les technologies utilisées pour la réalisation de notre projet, ainsi que l'environnement de travail. Nous avons également présenté les principales interfaces qui permettent de répondre aux besoins de notre application.

## Conclusion

En conclusion, ce projet nous a permis d'appliquer les compétences et connaissances acquises tout au long de notre formation en Génie Logiciel à l'ENSIAS. Nous avons ainsi pu concevoir et réaliser un système automatisé de gestion des absences, combinant une application web de gestion et un système RFID. Ce système offre une solution innovante et efficace pour la gestion des absences des étudiants et enseignants de l'établissement. Nous avons également pu nous familiariser avec les technologies modernes de développement web et de base de données, ainsi qu'avec les processus de gestion de projet. Enfin, ce projet nous a permis de travailler en équipe, de partager nos connaissances et de collaborer efficacement pour atteindre nos objectifs communs. Toutefois, il reste des perspectives d'amélioration de notre solution, notamment en termes d'optimisation des performances et d'ajout de fonctionnalités supplémentaires.

## **Webographie**

*[https ://github.com/YassineOuhadi/Attendance-Rest-API/tree/master](https://github.com/YassineOuhadi/Attendance-Rest-API/tree/master)*

*[https ://github.com/JobeeeAID/Student-s-attendance-management-system-loT](https://github.com/JobeeeAID/Student-s-attendance-management-system-loT)*