

Université Mohammed premier

École Nationale Des Sciences Appliquées d'Oujda

*Génie informatique*

Rapport de projet fin d'année

Soutenu le :

26/06/2024

Sujet :

Conception et Réalisation d'une  
Application de gestion  
d'absence

Réalisé par :

REMLI Oussama  
OMARI Mohamed  
MAKKOUR Mouad  
YACHOU Bilal

Encadré par :

Pr.SABER MOHAMMED

Année universitaire : 2023-2024

# *Remerciements*

Nous tenons à exprimer notre sincère gratitude à notre encadrant, M. Saber Mohammed, pour son accompagnement précieux tout au long de ce projet. Sa disponibilité, son expertise et ses conseils avisés ont été des atouts majeurs qui ont grandement contribué à la réussite de notre travail. Nous sommes particulièrement reconnaissants pour l'attention qu'il a portée à nos interrogations, pour sa capacité à nous orienter vers les bonnes pratiques, et pour l'encouragement constant qu'il nous a prodigué. Ce projet a été une opportunité d'apprentissage exceptionnelle, et c'est en grande partie grâce à son soutien indéfectible que nous avons pu le mener à bien

# Table des matières

<b>Introduction générale.....</b>	<b>9</b>
-----------------------------------	----------

## **Chapitre 1 :**

<b>Contexte Générale Du Projet.....</b>	<b>11</b>
---	-----------

Introduction.....	12
-------------------	----

Contexte Académique.....	12
--------------------------	----

Objectif du projet.....	12
-------------------------	----

Présentation Générale du Projet.....	13
--------------------------------------	----

❖ Problématique.....	13
----------------------	----

❖ Solutions existantes.....	13
-----------------------------	----

❖ Solution Proposée.....	13
--------------------------	----

Méthodologie de Travail.....	14
------------------------------	----

Conclusion.....	14
-----------------	----

## **Chapitre 2 :**

<b>Analyse et conception.....</b>	<b>15</b>
-----------------------------------	-----------

Introduction.....	16
-------------------	----

Besoins Fonctionnels.....	16
---------------------------	----

Besoins Non-Fonctionnels.....	17
-------------------------------	----

Méthodologie d'analyse.....	17
-----------------------------	----

❖ Le langage UML.....	17
-----------------------	----

❖ Le modèle MVC.....	18
----------------------	----

❖ Les acteurs et leurs rôles.....	18
-----------------------------------	----

❖ Diagrammes des cas d'utilisation:.....	20
--	----

➤ administrateur.....	21
-----------------------	----

➤ Professeur.....	22
-------------------	----

➤ Étudiant.....	22
-----------------	----

Diagrammes de Séquence.....	22
-----------------------------	----

➤ Authentification.....	23
-------------------------	----

➤ Ajouter Une Filière.....	24
➤ Importer Liste Professeurs :.....	24
➤ Importer Emploi Du Temps:.....	25
➤ Marquer Absence (Professeur):.....	26
➤ Marquer Présence (Etudiant):.....	27
❖ Diagrammes de classe.....	28
Conclusion.....	29
<b>Chapitre 3:</b>	
<b>Architecture de l'application.....</b>	<b>31</b>
Introduction.....	32
Architecture de l'application.....	32
Technique et Outils de développement.....	33
❖ Backend.....	33
❖ Frontend.....	34
❖ Base de Données.....	34
Environnement logiciel.....	35
Présentation de l'application.....	36
❖ interface d'authentification.....	36
Interface d'accueil.....	36
interface d'authentification.....	37
interface de réinitialisation de mot de passe.....	38
❖ interface administrateur.....	39
interface d'accueil.....	39
Interface de gestion des départements.....	40
interface de suivie d'absence.....	43
interface de paramétrage.....	45
❖ interface professeur.....	46
Interface d'accueil.....	46
Interface de suivi d'absence.....	50
Interface de paramétrage.....	52
❖ interface étudiant.....	54
<b>Conclusion générale.....</b>	<b>56</b>

## Table de figures

Figure 1 : Le modèle MVC .....	18
Figure 2 : Diagrammes des cas d'utilisation d'administrateur .....	21
Figure 3 : Diagrammes des cas d'utilisation de professeur .....	22
Figure 4 : Diagrammes des cas d'utilisation d'étudiant .....	22
Figure 5 : Diagramme de séquence d'authentification .....	23
Figure 6 : diagramme de séquence d'ajout de filière.....	24
Figure 7 : diagramme de séquence d'importation de liste des professeurs.....	25
Figure 8 : diagramme de séquence d'importation d'emploi du temps.....	26
Figure 9 : diagramme de séquence de marquage d'absence .....	27
Figure 10 : diagramme de séquence de marquer la présence .....	28
Figure 11 : diagramme de classe .....	29
Figure 12 : architecture de l'application .....	33
Figure 13 : interface d'accueil de l'application .....	37
Figure 14 : interface d'authentification .....	38
Figure 15 : interface réinitialisation de mot de passe-étape 1 .....	38
Figure 16 : interface réinitialisation de mot de passe-étape 2 .....	39
Figure 17 : interface d'accueil d'administrateur.....	40

Figure 18 : formulaire d'ajout d'une filière .....	40
Figure 19 : liste des modules.....	41
Figure 20 : liste des professeurs.....	41
Figure 21 : formulaire d'ajout des modules .....	42
Figure 22 : formulaire d'ajout des professeurs .....	42
Figure 23: formulaire d'affectation .....	43
Figure 24 : suivi d'absence pour un niveau .....	43
Figure 25 : gestion du niveau .....	44
Figure 26 : Liste des étudiants absents .....	44
Figure 27 : listes des absences d' étudiant.....	44
Figure 28 : menu déroulant .....	45
Figure 29 : interface pour modifier les informations d'administrateur .....	45
Figure 30 : interface d'accueil de professeur .....	46
Figure 31 : composant pour afficher le module actuelle .....	46
Figure 32 : composant pour afficher les niveaux .....	47
Figure 33 : composant pour afficher les étudiants ayant marqué la présence .....	47
Figure 34 : liste des étudiant de séance actuelle .....	47

Figure 35 : liste des boutons pour naviguer entre les niveaux .....	48
Figure 36 : composant pour afficher le code QR de la séance .....	48
Figure 37 : liste des boutons pour choisir la méthode de marquage d'absence .....	49
Figure 38 : Boutons pour marquer l'absence .....	49
Figure 39 : Dialogue pour confirmer le marquage d'absence .....	49
Figure 40 : Barre latérale .....	50
Figure 41 : Liste des étudiants absents .....	51
Figure 42 : listes des absences d' étudiant .....	52
Figure 43 : Composant affichant les informations d'étudiant .....	52
Figure 44 : menu déroulant .....	53
Figure 45 : menu déroulant des séances du professeur .....	53
Figure 46 : interface pour modifier les données du professeur .....	54
Figure 47 : Forme pour remplir l'apogée d'étudiant .....	54
Figure 48 : boîte de dialogue de confirmation des informations de l'étudiant ....	55

## **liste des tableaux**

Tableau 1 : Acteurs et rôles .....	19
------------------------------------	----

## Table des abréviations

MVC	Model View Controller
UML	Unified Modeling language
HTTP	Hyper text Transfert Protocol
CRUD	Create Read update Delete
MUI	Material User Interface
API	Application Programming Interface
RESTful	Representational State transfer
SGBDR	Système de Gestion de Base de Données Relationnelle
QR	Quick Response



# Introduction générale

À l'ère du numérique, il est devenu crucial de numériser les services administratifs pour suivre le rythme d'un monde qui évolue rapidement. Les méthodes traditionnelles, souvent inefficaces et laborieuses, rencontrent des difficultés pour répondre aux exigences d'un monde en constante évolution. La gestion des absences, en particulier, représente un défi majeur dans de nombreuses organisations, où les processus manuels peuvent être pénibles et sources d'erreurs. Face à cette réalité, notre projet de fin d'études se positionne comme une réponse innovante et digitale pour optimiser la gestion des absences.

Notre solution repose sur l'intégration de technologies avancées comme les codes QR et la reconnaissance faciale, visant à optimiser les procédures en offrant une méthode rapide, précise et efficace pour enregistrer les périodes d'absence, tout en assurant une gestion fluide et efficace.

Le présent rapport détaille le contexte général du projet, en mettant en lumière la problématique de la gestion des absences. Nous explorerons également les étapes de conception et modélisation, de développement et d'implémentation de notre solution, en mettant l'accent sur les choix technologiques, et les résultats obtenus.

# Chapitre 1 : Contexte Générale Du Projet

## Introduction

Ce chapitre vise à situer notre projet dans son contexte global en explorant la problématique qui a engendré la création de notre application, en fournissant une description détaillée du projet lui-même et en exposant la solution envisagée.

## Contexte Académique

Ce projet fin d'année s'inscrit dans un cadre académique l'École nationale des sciences appliquées d'Oujda, plus précisément au sein de l'atelier des activités pratiques. Il a pour objectif de donner aux étudiants l'opportunité de mettre en œuvre les compétences acquises et de mettre en pratique les connaissances théoriques assimilées tout au long de la formation génie informatique. Il est conçu pour fournir un environnement stimulant visant à renforcer la compréhension des sujets étudiés et à développer des compétences pratiques essentielles pour une future carrière professionnelle.

## Objectif du projet

Ce projet est conçu pour répondre aux besoins croissants de gestion des absences à l'École nationale des sciences appliquées d'Oujda, où la complexité augmente avec le nombre croissant d'étudiants. Il propose des outils flexibles et conviviaux à destination des professeurs et de l'administration, facilitant ainsi l'enregistrement des absences et la

gestion des divers éléments nécessaires tels que les départements, modules et les emplois du temps.

## Présentation Générale du Projet

### ❖ Problématique

La gestion des absences dans les institutions éducatives est un problème majeur, souvent géré manuellement, ce qui peut entraîner des erreurs et une perte de temps considérable. Le défi est de mettre en place un système efficace, précis et automatisé pour suivre les présences et les absences de manière fiable.

### ❖ Solutions existantes

Les méthodes traditionnelles comme la gestion des présences via des fichiers Excel, des feuilles papier ou des Google Sheets présentent des inconvénients notables, notamment en termes de précision. Avec l'augmentation du nombre d'étudiants et d'employés dans les grandes institutions, ces méthodes manuelles deviennent de plus en plus difficiles à gérer efficacement et sont sujettes à des erreurs.

### ❖ Solution Proposée

Notre projet innove dans la gestion des présences en proposant une application web qui combine l'utilisation de codes QR et la reconnaissance faciale pour automatiser le suivi des présences et des absences. Avant chaque séance, un code QR unique est affiché, permettant aux étudiants de marquer leur présence en scannant ce code à leur arrivée. De plus, un système de reconnaissance faciale enregistre automatiquement l'absence des étudiants en comparant leur visage avec une base de données préenregistrée dès leur entrée dans la salle de cours. facilitant la génération de statistiques détaillées sur les absences. Cette approche simplifie et sécurise la gestion des absences tout en répondant aux besoins suivis de la présence étudiante.

## Méthodologie de Travail

Pour la réalisation de ce projet, nous avons adopté la méthodologie agile Scrum, adaptée à notre équipe d'étudiants et supervisée par notre encadrant. Scrum se compose de cycles de développement appelés sprints, généralement de deux semaines. Chaque sprint commence par une réunion de planification où nous définissons les objectifs et les tâches prioritaires. Nous organisons des réunions quotidiennes pour suivre l'avancement et ajuster les priorités si nécessaire. À la fin de chaque sprint, nous présentons l'avancement du projet à notre encadrant et planifions les tâches à venir en fonction des retours reçus. Enfin, nous réalisons une rétrospective pour identifier les points d'amélioration. Cette approche itérative et collaborative assure une adaptation continue aux besoins des utilisateurs et une livraison rapide de fonctionnalités.

## Conclusion

Ce chapitre présente notre projet et ses objectifs, en contextualisant notre application. Nous avons exposé les défis associés à la gestion des absences dans les institutions éducatives et la solution novatrice que nous proposons..

# Chapitre 2 : Analyse et conception

## Introduction

La phase de spécification des besoins est cruciale pour assurer le succès d'un projet. Dans ce chapitre, nous nous concentrerons sur la définition claire des fonctionnalités attendues de notre application de gestion des absences par code QR et détection faciale. Nous définirons les différents cas d'utilisation et présenterons des scénarios qui illustreront ces cas, afin d'atteindre une vision précise des besoins de notre projet. Ensuite, nous aborderons la phase de conception, essentielle avant la réalisation de tout projet, car elle établit un modèle de référence fidèle à la réalité, permettant de mettre en évidence les aspects clés du projet.

## Besoins Fonctionnels

### Gestion administrative :

L'administrateur doit pouvoir initialiser les données du système, y compris les informations sur les professeurs, les modules et l'emploi du temps. Il doit aussi pouvoir ajouter de nouveaux professeurs et modules, ainsi que les affecter aux plannings appropriés.

### Génération de codes QR :

Avant chaque séance, un code QR unique doit être généré pour chaque utilisateur inscrit à la séance.

### Enregistrement des présences et des absences :

Les données de présence doivent être enregistrées en temps réel dans une base de données centrale. Il doit être possible de justifier les absences des étudiants et de les enregistrer de manière appropriée dans le système.

### Suivi des données I :

Les administrateurs doivent avoir un accès instantané aux informations de présence des étudiants. La consultation des statistiques et rapports sur les absences permet un suivi précis et une gestion proactive.

Intégration d'un système de marquage d'absence utilisant la

### reconnaissance faciale :

Les étudiants sont identifiés par reconnaissance faciale à leur arrivée en

classe. Les absences sont automatiquement enregistrées si un étudiant identifié par reconnaissance faciale n'est pas présent.

## Besoins Non-Fonctionnels

### **Simplicité d'utilisation :**

L'application doit être facile à utiliser sans nécessiter de formation préalable.

### **Temps de réponse rapide :**

L'application doit répondre aux actions des utilisateurs en moins de 2 secondes.

### **Facilité d'optimisation :**

Le code doit être bien structuré pour permettre des améliorations futures.

### **Sécurité des informations utilisateur :**

Les données des utilisateurs doivent être protégées par un chiffrement sécurisé.

### **Interface conviviale :**

L'interface utilisateur doit être intuitive et agréable à utiliser.

### **Confidentialité des données :**

Les données personnelles des utilisateurs doivent être traitées de manière confidentielle et sécurisée.

## Méthodologie d'analyse

### ❖ Le langage UML



UML (Unified Modeling Language) est un langage visuel composé de diagrammes qui fournissent diverses perspectives sur le projet. Il est utilisé pour représenter les aspects fonctionnels, structurels et comportementaux du logiciel à développer. Cette approche visuelle est choisie pour sa capacité à clarifier la complexité du projet et faciliter la communication entre les parties prenantes.



## ❖ Le modèle MVC

L'architecture MVC (Modèle, Vue, Contrôleur) est un concept puissant pour le développement d'applications. Elle sépare les données (modèle), l'affichage (vue) et les actions (contrôleur), assurant ainsi une architecture claire. Cette séparation simplifie la tâche des développeurs, facilitant la maintenance et l'amélioration du projet. Les interactions entre le modèle, la vue et le contrôleur sont illustrées par un schéma explicatif.

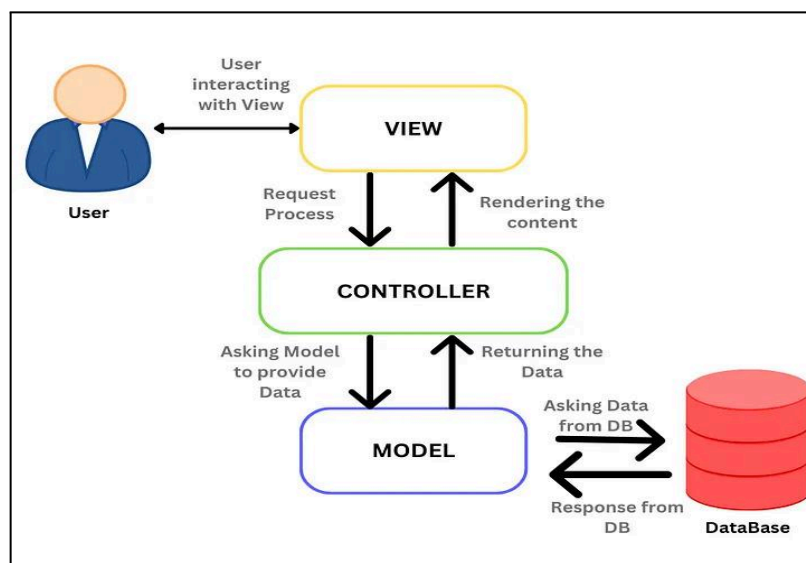


figure 1 : Le modèle MVC

## ❖ Les acteurs et leurs rôles

Un acteur est une entité externe qui interagit avec le système (opérateur, centre distant, autre système...). En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. Les acteurs peuvent être classés (hiérarchie). Pour notre projet les deux acteurs principaux sont l'administrateur du service des absences et les professeurs

Acteur	Rôles
<b>Administrateur</b>	S'Authentifier  Gestion des professeurs Ajouter un/plusieurs professeurs Consulter liste des professeurs par département  Importer liste des étudiants par niveau  Importer emploi du temps  Importer photos des étudiants  Gestion des modules Ajouter un/plusieurs modules Consulter liste des modules par département Affecter modules au professeurs  Gestion des filières Ajouter un nouveau (avec ses niveaux)  Gestion d'Absence Consulter l'absence de chaque niveau Consulter l'absence de chaque module Consulter l'absence de chaque étudiant en de Justifier absence

<b>Professeur</b>	S'Authentifier  Gérer son espace  Marquer Absence Choisir mode de marquage <ul style="list-style-type: none"> <li>o QR Code</li> <li>o Détection faciale</li> </ul> Reporter une séance ultérieurement  Gestion d'Absence <ul style="list-style-type: none"> <li>Consulter l'absence de chaque module</li> <li>Consulter l'absence de chaque étudiant en détail</li> <li>Justifier absence</li> </ul>
Etudiant	Scanner QR Code Entrer Num Apogée Se place devant la caméra

tableau 1: Acteurs et Rôles

### ❖ Diagrammes des cas d'utilisation:

Le diagramme de cas d'utilisation représente les fonctionnalités (ou dit cas d'utilisation) nécessaires aux utilisateurs. On peut faire un diagramme de cas d'utilisation pour le logiciel entier ou pour chaque package.

## ➤ administrateur

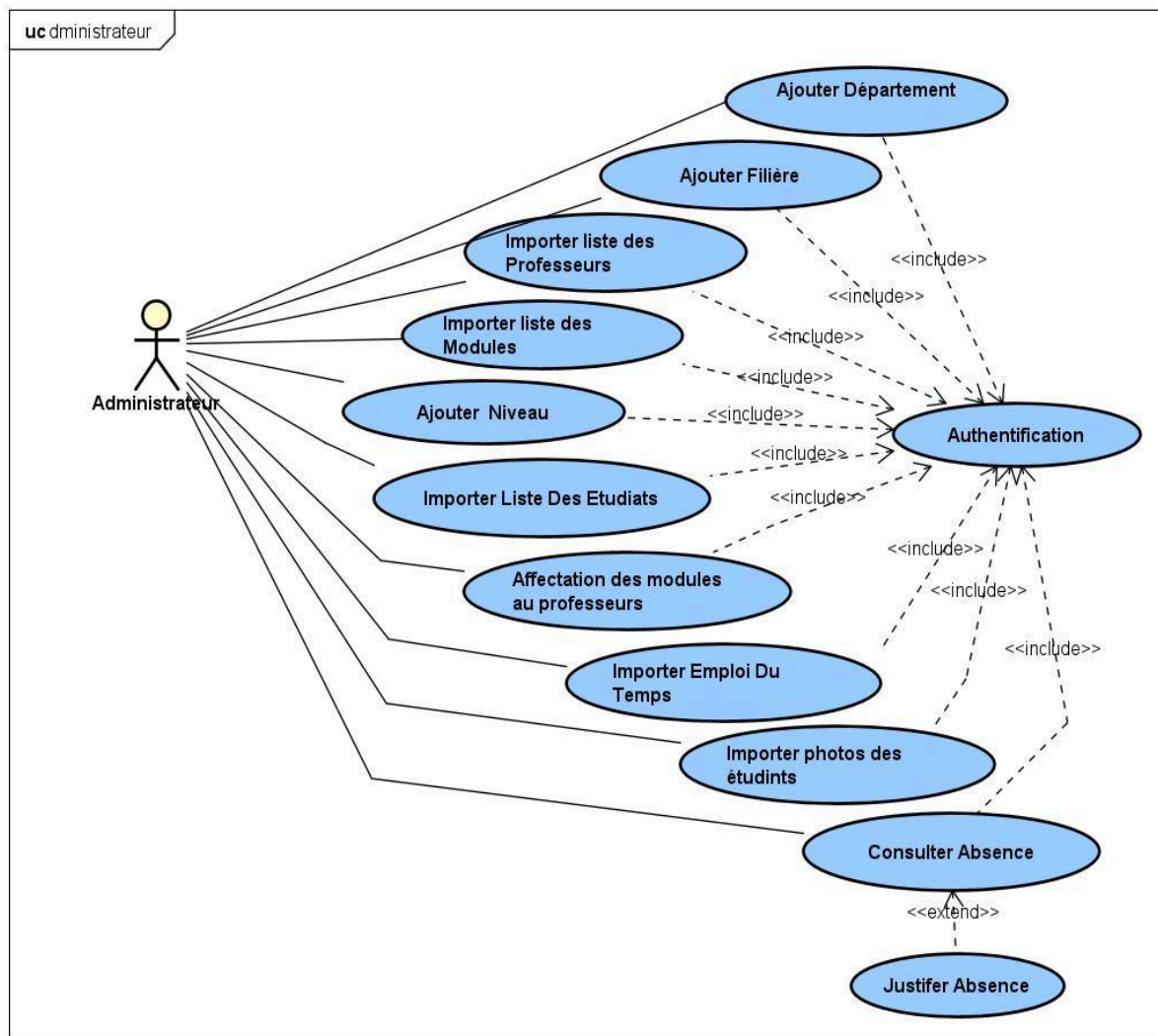


figure 2 : Diagramme des cas d'utilisation d'administrateur

## ➤ Professeur

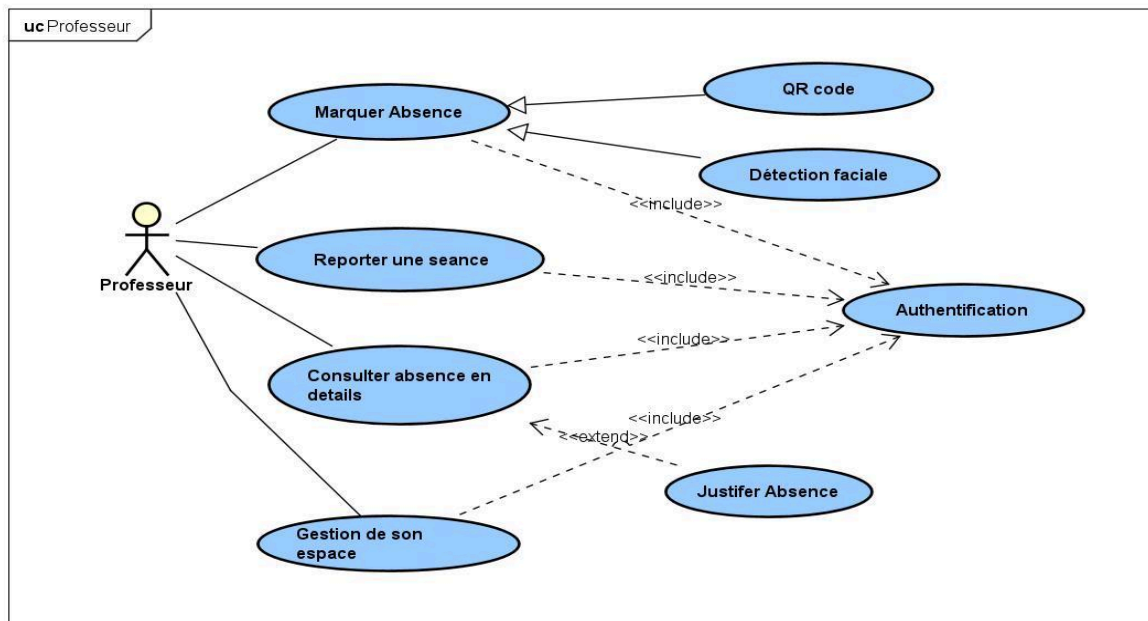


figure 3 : Diagrammes des cas d'utilisation de professeur

## ➤ Étudiant

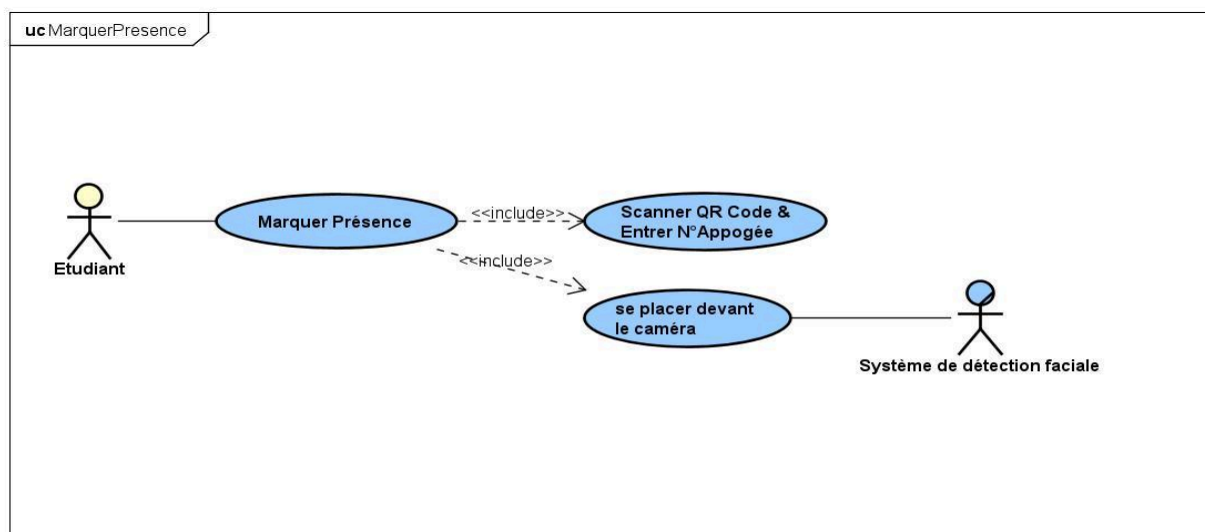


figure 4 : Diagrammes des cas d'utilisation d'étudiant

## Diagrammes de Séquence

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met

l'accent sur la chronologie des envois de messages. En ce qui suit-on présentera quelques diagrammes de séquences relatifs aux cas d'utilisations présentées:

### ➤ Authentification

L'authentification permet à l'utilisateur d'accéder à l'application en saisissant le login et le mot de passe, si tous les champs sont bien remplis et les informations saisies sont correctes elle se redirige vers la page d'accueil, sinon un message d'erreur va être déclenché.

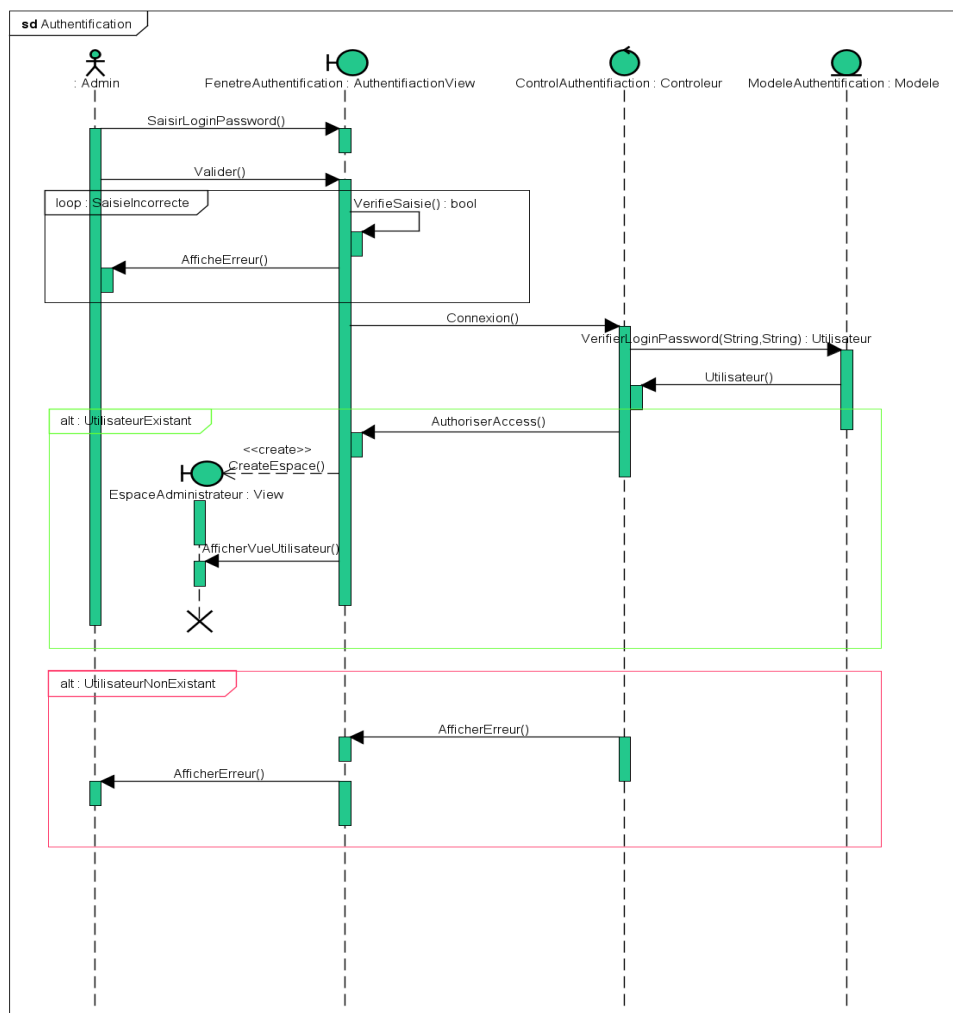


figure 5 : diagramme de séquence d'authentification

## ➤ Ajouter Une Filière

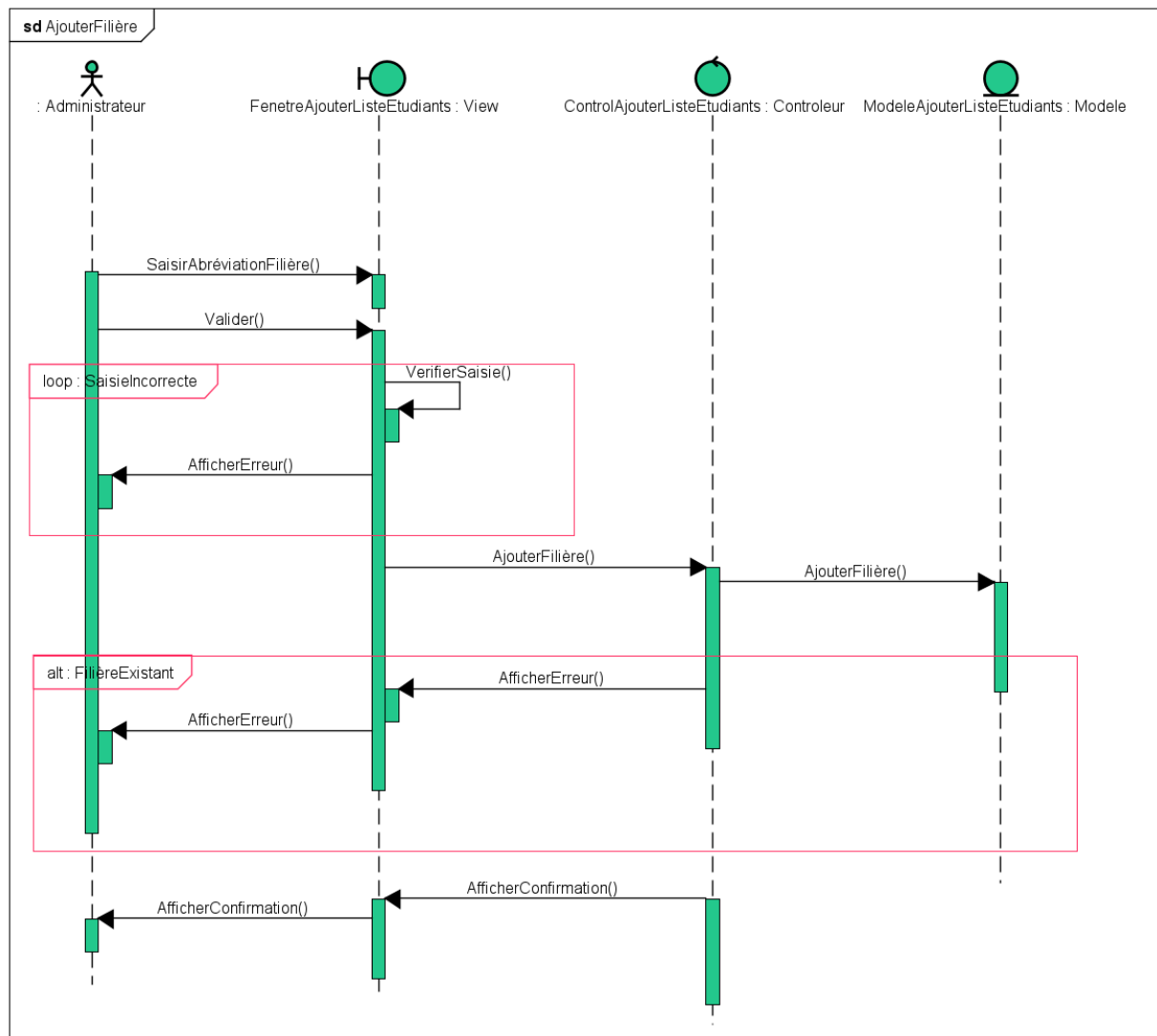


figure 6 : diagramme de séquence d'ajout filière

## ➤ Importer Liste Professeurs :

Une autre fonction de l'administrateur est d'importer la liste des professeurs par département via un fichier excel, ou bien ajouter un professeur individuellement via un formulaire.

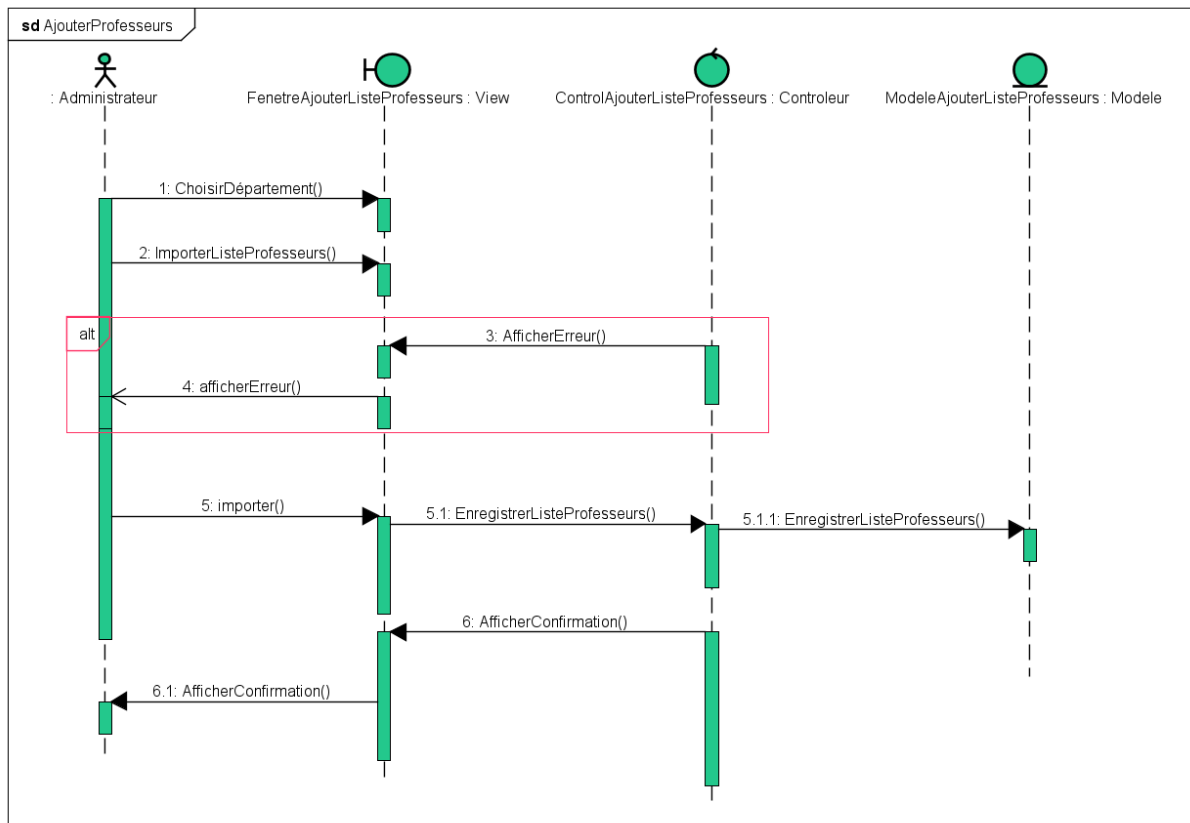


figure 7 : diagramme de séquence d'importation de liste des professeurs

### ➤ Importer Liste Modules :

Une autre fonction de l'administrateur est d'importer la liste des modules par filière via un fichier excel, ou bien ajouter un module individuellement via un formulaire (c'est la même démarche comme l'importation des professeurs).

### ➤ Importer Emploi Du Temps:

L'administrateur peut aussi importer l'emploi du temps pour chaque niveau via un fichier excel.



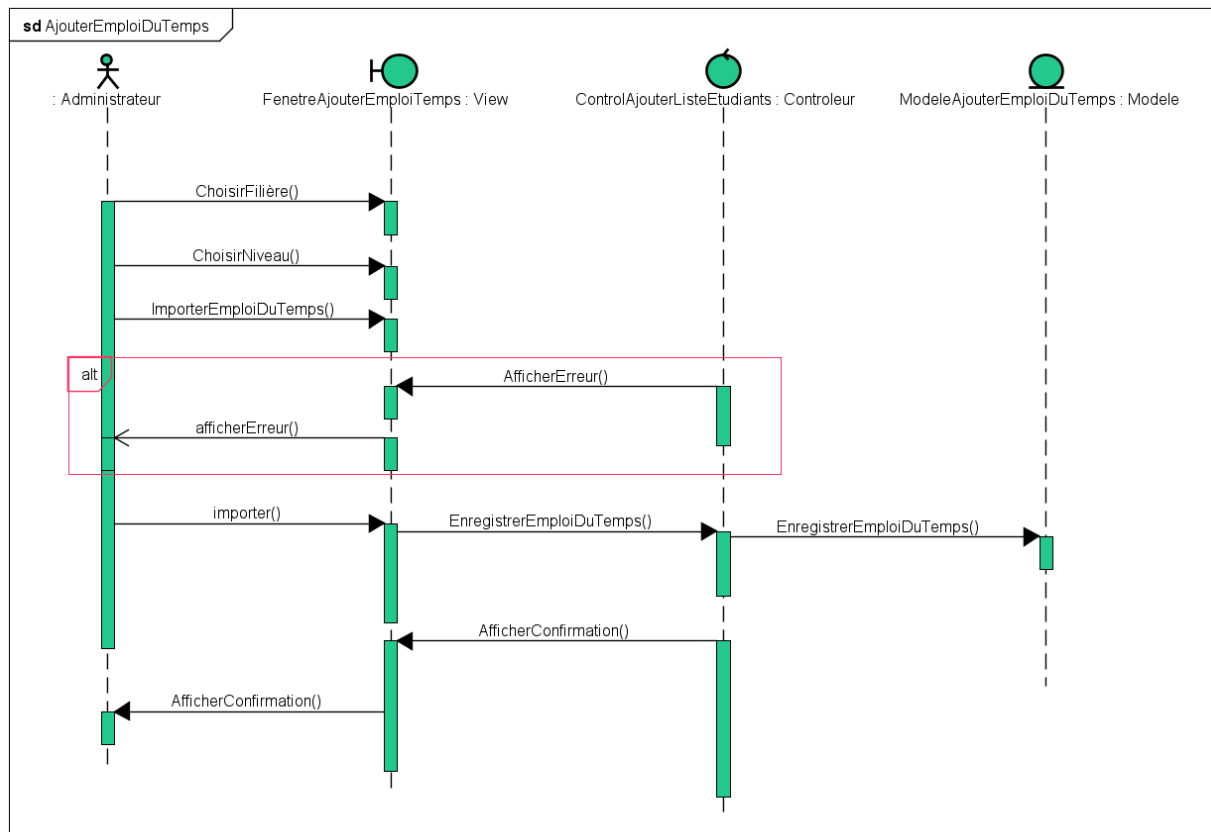


figure 8 : diagramme de séquence d'importation d'emploi du temps

### ➤ Marquer Absence (Professeur):

Le professeur doit marquer l'absence en choisissant soit l'option de QR code ou bien à travers la détection faciale.

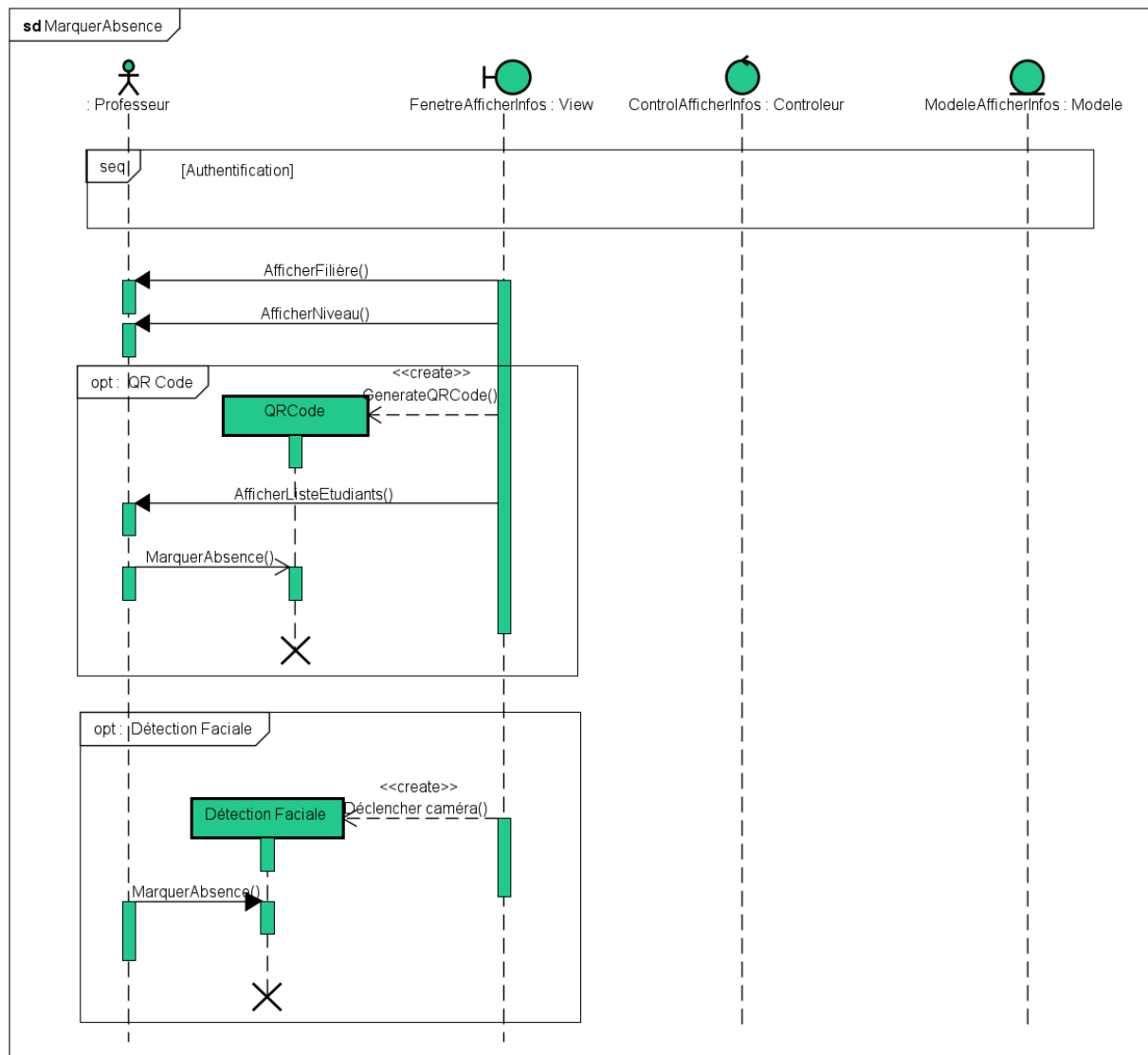


figure 9: diagramme de séquence de marquage d'absence

### ➤ Marquer Présence (Etudiant):

L'étudiant doit marquer sa présence en scannant le QR Code puis valider son numéro apogée dans un champ de saisie ou bien se déplacer devant la caméra.

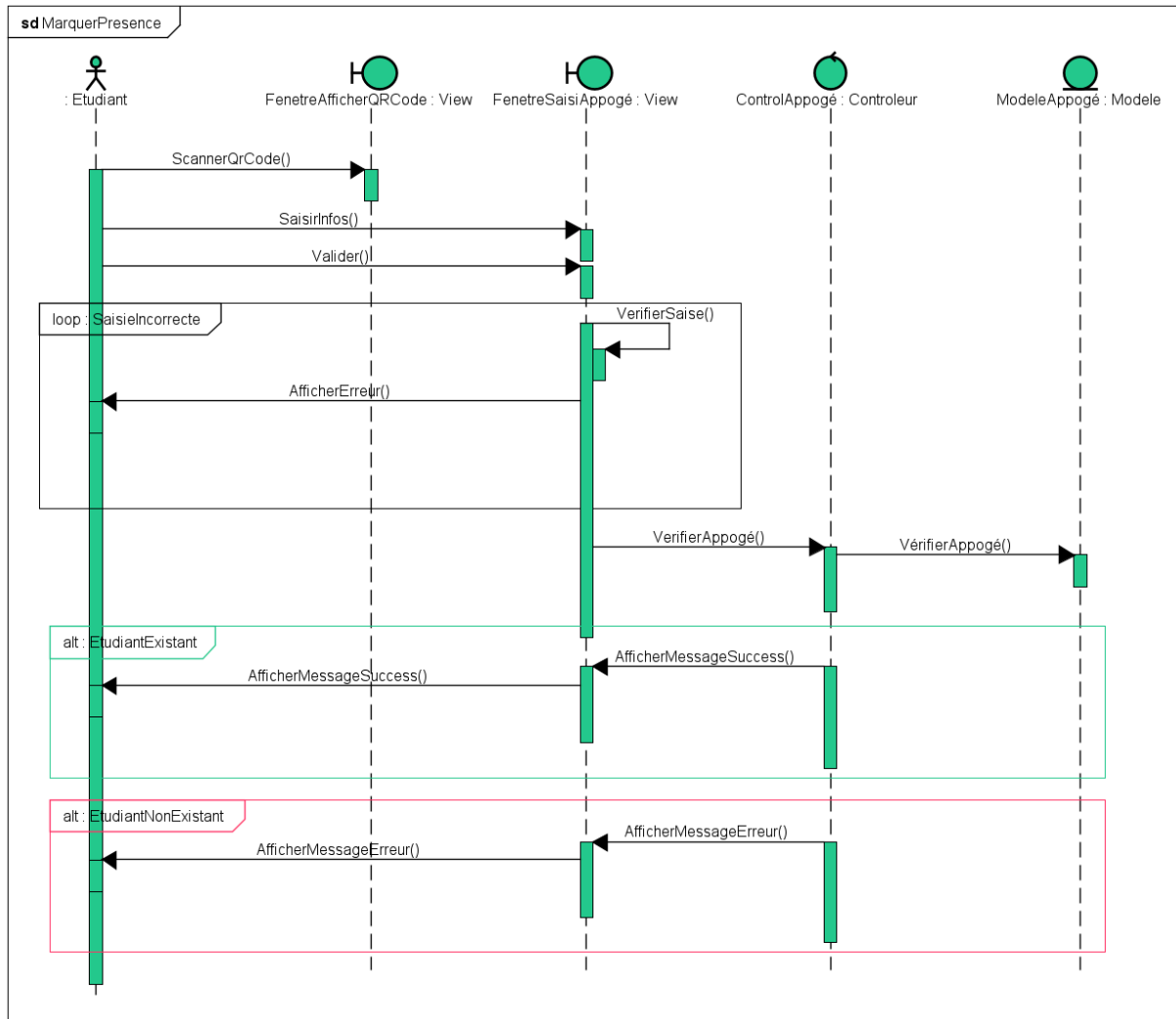


figure 10 : diagramme de séquence de marquer la présence

## ❖ Diagrammes de classe

Dans la phase d'analyse, ce diagramme représente les entités (des informations) manipulées par les utilisateurs. Dans la phase de conception, il représente la structure objet d'un développement orienté objet.

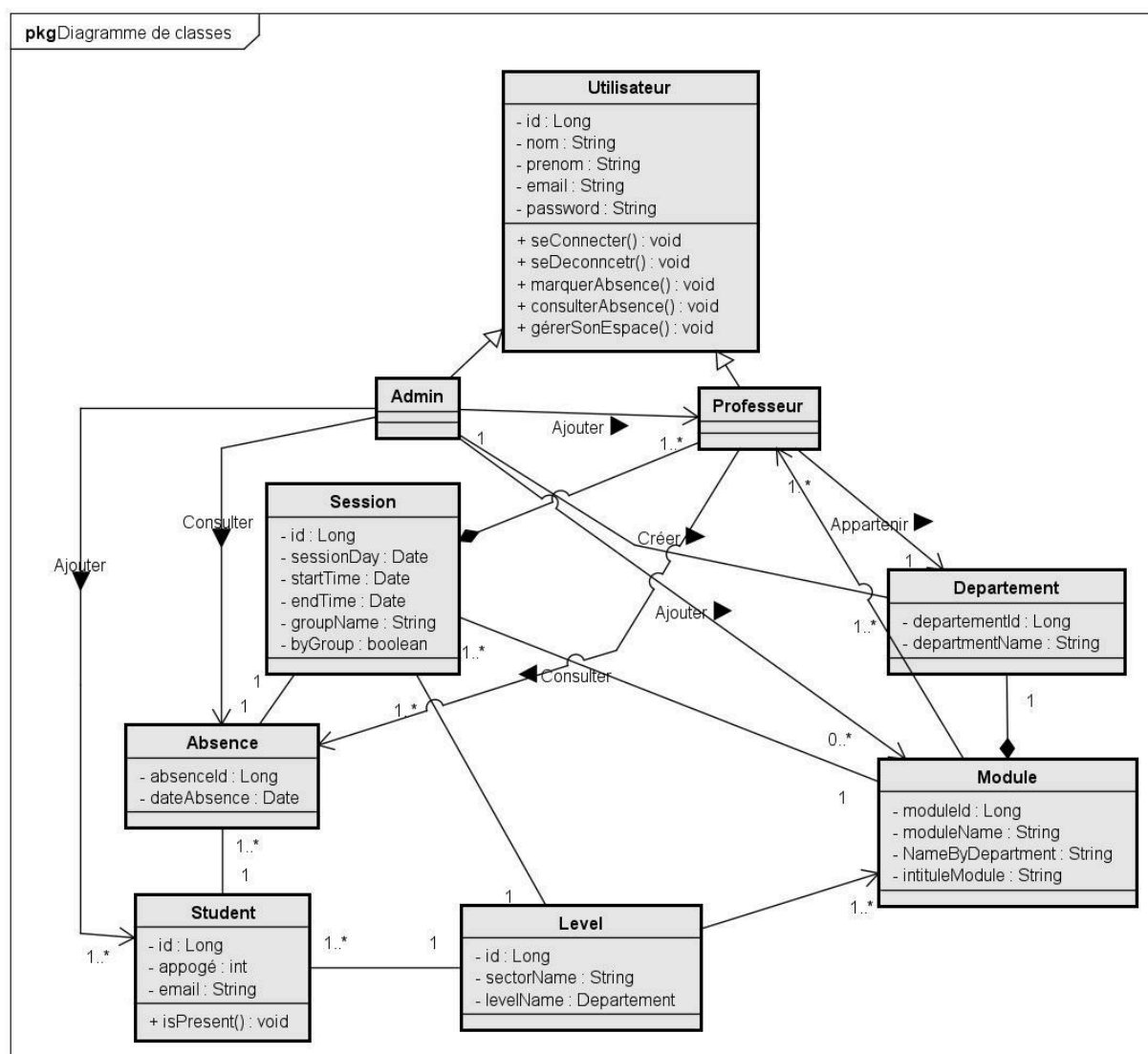


figure 11 : diagramme de classe

## Conclusion

En conclusion, la spécification des besoins de notre application de gestion des absences par code QR et détection faciale a permis de définir clairement les fonctionnalités et exigences nécessaires. Les aspects fonctionnels incluent la gestion administrative, la génération de codes QR, l'enregistrement et le suivi des présences en temps réel, ainsi que l'intégration de la reconnaissance faciale. Les besoins non-fonctionnels mettent l'accent sur la simplicité d'utilisation, la rapidité de réponse, la sécurité des informations, une interface conviviale, et la compatibilité

multiplateforme. La phase de conception nous a permis de modéliser ces besoins en utilisant des méthodologies adaptées comme UML et l'architecture MVC. Ce système efficace permet une gestion précise des présences, adaptée aux besoins spécifiques de chaque utilisateur dans un environnement éducatif structuré.

# Chapitre 3

## Architecture de l'application

## Introduction

Après avoir achevé la phase de conception, cette section met en évidence les étapes essentielles pour la réussite de notre projet. Nous détaillerons l'architecture de l'application ainsi que les outils et langages de développement utilisés. De plus, nous présenterons les différentes interfaces de notre système, notamment celles destinées aux administrateurs, aux professeurs et aux étudiants. Chaque interface est conçue pour répondre aux besoins spécifiques de ses utilisateurs et faciliter leur travail. Enfin, nous aborderons les principales fonctionnalités de ces interfaces et expliquerons comment elles permettent à chacun d'utiliser le système de manière efficace.

## Architecture de l'application

L'architecture de l'application est une structure modulaire adopte une approche à trois niveaux, séparant clairement la couche de présentation, la logique métier et la gestion des données. La couche de présentation se compose d'une UI intuitive et accessible, permettant aux administrateurs de gérer les comptes et les données, aux professeurs de suivre les absences et aux étudiants de marquer la présence. La logique métier est gérée par les serveurs Flask et Spring, où Flask se concentre sur le traitement des images pour l'identification des étudiants via une base de données SQLite locale, tandis que Spring gère la persistance des données relatives aux étudiants, professeurs, modules, séances et absences dans une base de données principale. La gestion des données est assurée par deux bases de données distinctes : SQLite pour les images, permettant un accès rapide et efficace aux données d'identification, et une base de données principale gérée par Spring pour les autres informations, garantissant robustesse, sécurité et scalabilité des données.

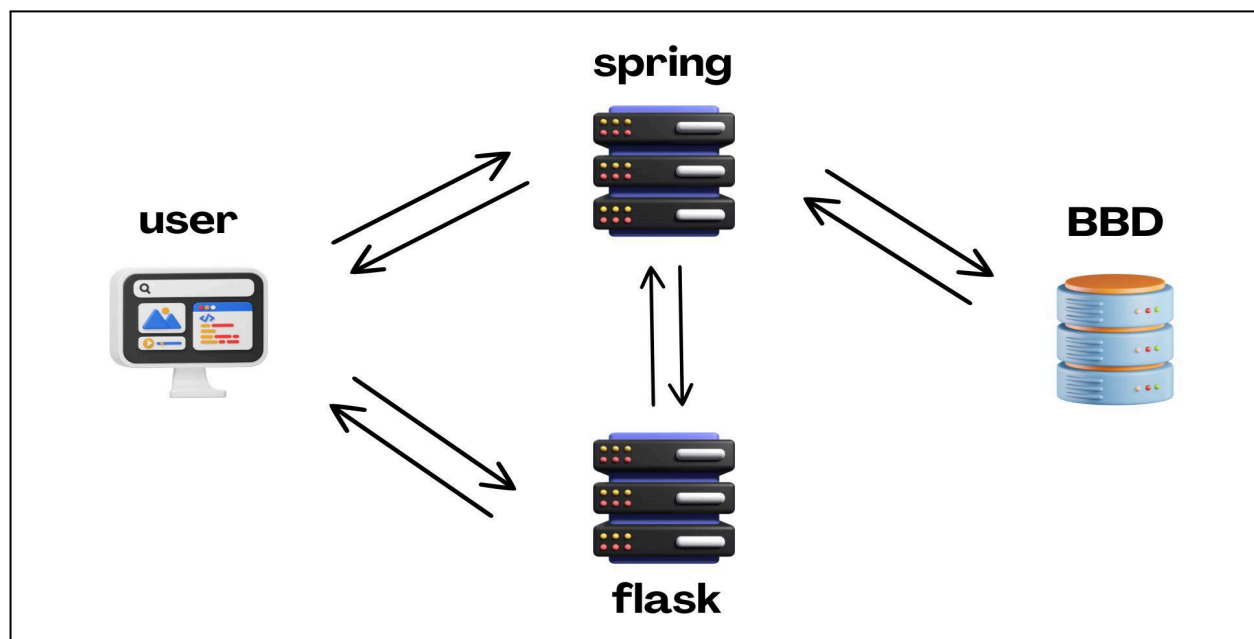


figure 12 : architecture de l'application

## Technique et Outils de développement

### ❖ Backend

#### Spring Boot



Spring Boot a été sélectionné comme framework principal pour le développement du backend, offrant une plateforme robuste pour créer des API RESTful. Les fonctionnalités clés incluent la gestion des requêtes HTTP pour les opérations CRUD sur les données des étudiants et leurs absences, la sécurité des informations sensibles des professeurs et des étudiants, et l'intégration optimisée avec la base de données MySQL grâce à Spring Data JPA, facilitant ainsi l'interaction et la manipulation des données.

#### Flask



**Flask**

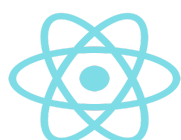
Flask a été sélectionné comme framework pour le micro service de détection faciale dans l'application en raison de sa simplicité, de sa flexibilité permettant l'intégration de la



bibliothèque face-recognition spécialement conçues pour la reconnaissance faciale

## ❖ Frontend

### React JS & MUI Library :



Pour le frontend, nous avons choisi React JS combiné à Material-UI (MUI), permettant de créer une interface utilisateur moderne et réactive. En mettant l'accent sur des composants réutilisables avec React, nous avons facilité la modularité et la maintenance de l'interface. Material-UI a été essentiel pour concevoir des interfaces élégantes grâce à ses composants pré-stylés comme les boutons, les tableaux et les formulaires, assurant ainsi une cohérence esthétique et une facilité d'utilisation.

## ❖ Base de Données

### MySQL



MySQL a été choisi comme SGBDR pour stocker les informations sur les étudiants et leurs absences. La conception du schéma relationnel a été essentielle pour structurer efficacement les tables et leurs relations. Des contraintes telles que les clés primaires et étrangères ont été mises en place pour assurer l'intégrité des données, garantissant ainsi la fiabilité et la cohérence des informations enregistrées dans la base de données.

### SQLite



SQLite est une base de données légère, simple et performante, idéale pour les applications embarquées. Elle

offre une gestion intégrée sans serveur, une grande compatibilité, une faible empreinte mémoire, des performances rapides même avec de gros volumes de données.

## Environnement logiciel

### Visual Studio Code (VS Code)



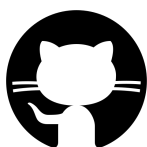
Visual Studio Code est un éditeur de code léger et puissant, populaire pour le développement front-end. Il offre des extensions spécialisées pour JavaScript et React, augmentant la productivité avec des fonctionnalités avancées et une intégration Git intégrée pour un contrôle de version efficace, simplifiant ainsi le flux de travail des développeurs.

### IntelliJ IDEA



IntelliJ IDEA est un IDE réputé pour Java et Spring Boot, offrant une complétion intelligente, un débogage efficace, et une intégration fluide avec Maven et Gradle. Il optimise la productivité des développeurs en simplifiant le développement Java et Spring, avec un support avancé pour la gestion de projets Spring, facilitant la navigation et l'utilisation des annotations.

### Github



GitHub facilite la gestion du code source et la collaboration en équipe grâce à son système de versioning précis. Les branches permettent un développement parallèle efficace, tandis que les pull-requests simplifient les revues de code et l'intégration, assurant une gestion transparente et sécurisée des contributions au projet..

### Gestion de Version : Git



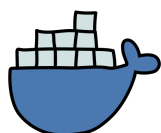
Git est essentiel pour le contrôle de version, assurant le suivi détaillé des modifications du code source. Il facilite la collaboration en permettant aux équipes de travailler simultanément sur des branches distinctes et de fusionner leurs contributions de manière ordonnée. De plus, Git joue un rôle crucial dans la gestion des versions en permettant de gérer efficacement les différentes itérations et releases du logiciel, assurant ainsi une gestion structurée et cohérente du développement.

### postman



Postman a été utilisé pour tester les endpoints de l'API backend, ce qui implique la création de collections de requêtes. Ces collections permettent de structurer les requêtes API afin de faciliter les tests et d'assurer la conformité aux spécifications de l'API.

### Docker



Docker a conteneurisé tous les composants de l'application : backend Spring Boot, frontend React, microservice Flask pour la reconnaissance faciale, et bases de données MySQL. Docker Compose a orchestré ces conteneurs, simplifiant la gestion des dépendances et assurant l'isolation des services. Cette approche a facilité le développement, les tests et le déploiement en continu, avec une configuration réseau sécurisée et une gestion efficace des environnements logiciels.

## **Présentation de l'application**

L'application est organisée en deux interfaces principales, une pour les administrateurs et une pour les éducateurs. Il existe une interface spécifique à l'étudiant pour suivre la présence et une interface d'authentification.

## ❖ interface d'authentification

### Interface d'accueil

L'interface d'accueil permet aux utilisateurs de choisir leur rôle, soit en tant qu'administrateur, soit en tant que professeur, afin d'accéder aux fonctionnalités appropriées. Cette page facilite la navigation en proposant des options claires et visuelles adaptées à chaque type d'utilisateur.

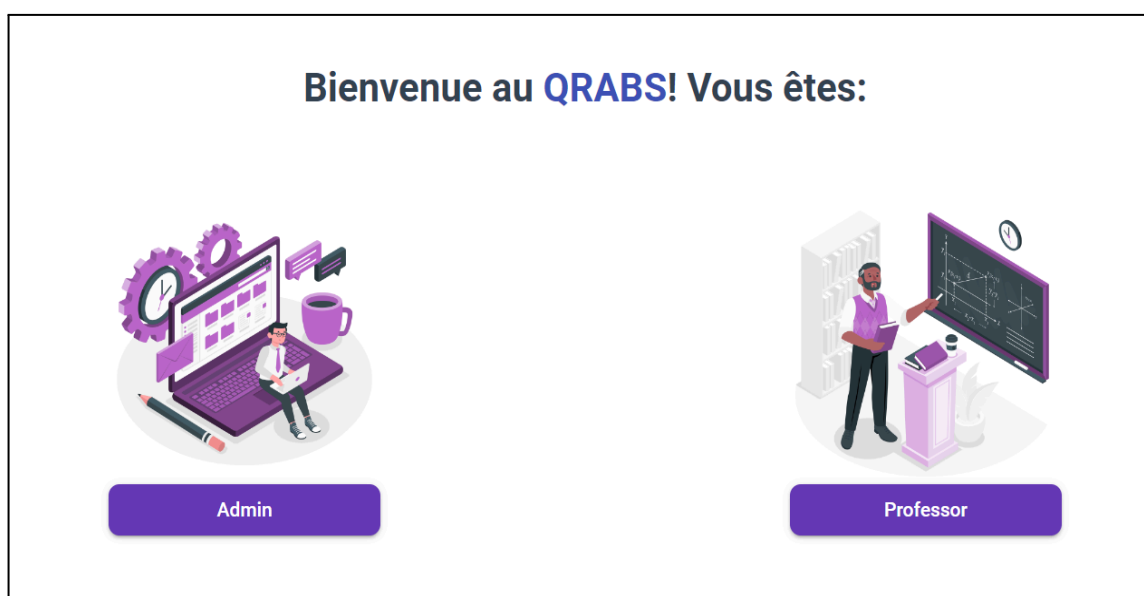
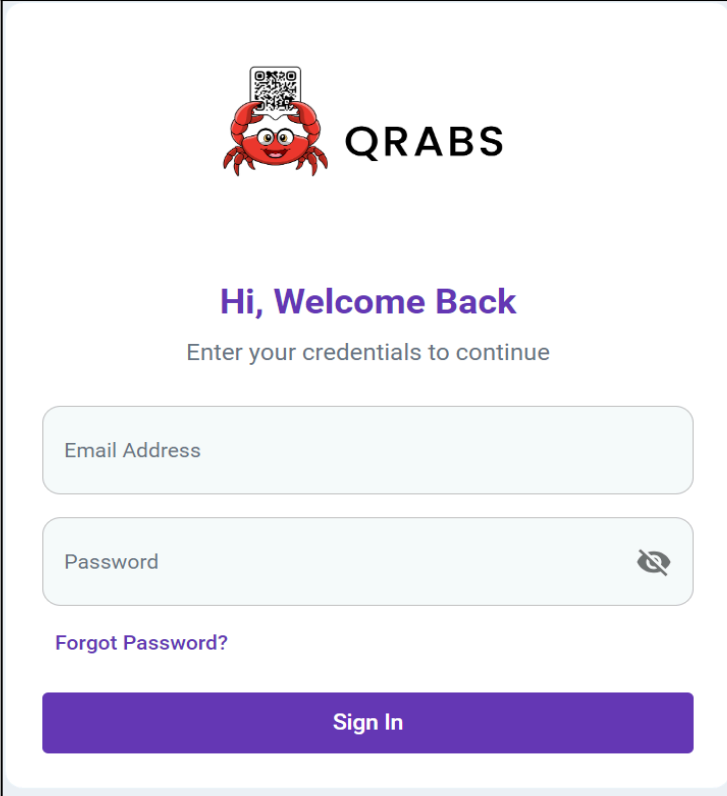


figure 13 : interface d'accueil de l'application

## interface d'authentification

C'est la page qui permet l'authentification des utilisateurs. Si les coordonnées saisies sont correctes, alors l'utilisateur est dirigé vers son espace, sinon il est redirigé une autre fois vers la page d'authentification.

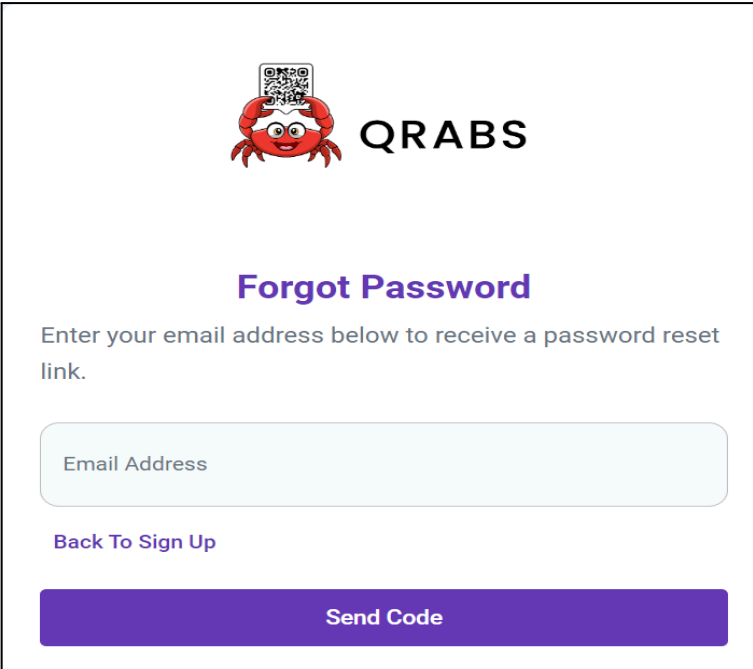


The login interface for QRABS features a red crab logo holding a QR code at the top left, followed by the text "QRABS". Below this is a purple heading "Hi, Welcome Back" and a subtitle "Enter your credentials to continue". The form includes an "Email Address" input field, a "Password" input field with a toggle icon, a "Forgot Password?" link, and a purple "Sign In" button.

figure 14 : interface d'authentification

## interface de réinitialisation de mot de passe


Cette section se divise en deux étapes. Tout d'abord, l'utilisateur saisit son adresse e-mail afin de recevoir un e-mail de vérification.



The forgot password interface for QRABS features the same red crab logo and "QRABS" text at the top. Below is a purple heading "Forgot Password" and a subtitle "Enter your email address below to receive a password reset link.". The form includes an "Email Address" input field, a "Back To Sign Up" link, and a purple "Send Code" button.

figure 15 : interface réinitialisation de mot de passe-étape 1

Après avoir reçu le code de vérification par email, l'utilisateur entre à la fois son nouveau mot de passe et le code de vérification.



**QRABS**

**Verification Code**

A verification code has been sent to your email. Please enter the code below and set a new password.

Verification Code

New Password

Confirm New Password

**Update Password**

figure 16 : interface réinitialisation de mot de passe-étape 2

## ❖ interface administrateur

### interface d'accueil

Après l'authentification, l'administrateur peut accéder à la page d'accueil qui contient un formulaire d'ajout des filières ainsi que des cartes représentant les filières et niveaux d'études, avec le nombre d'absences indiqué. Les cartes sont codées par couleur, les cartes rouges signalant un problème d'absentéisme, si le nombre total d'absence pour le niveau est supérieur à 3 absences, facilitant ainsi la gestion et le suivi des présences des étudiants.

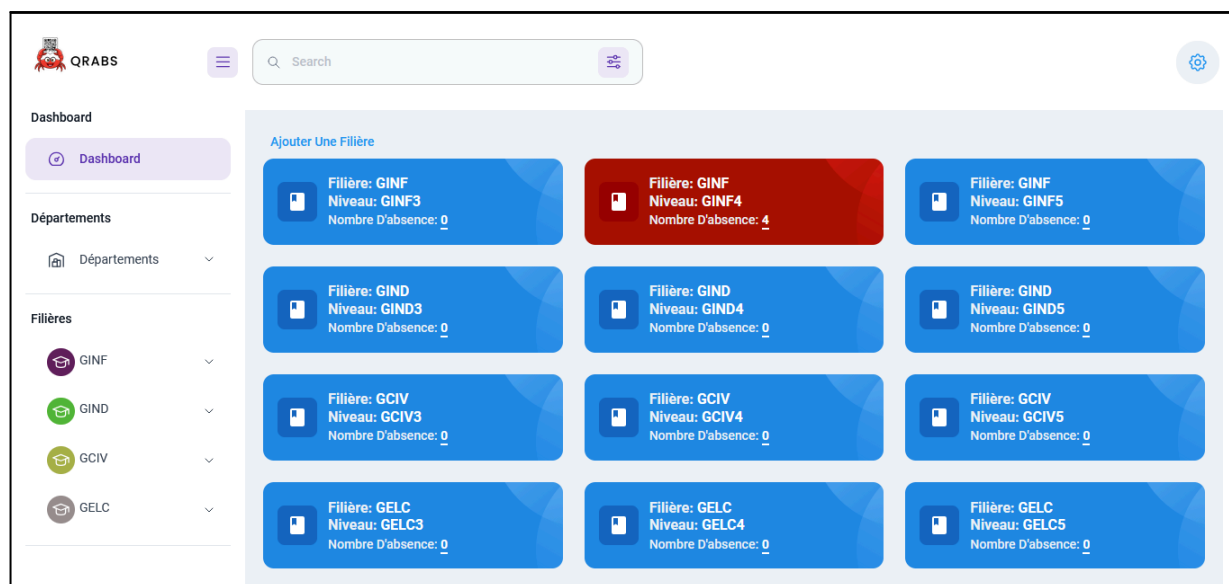


figure 17 : interface d'accueil d'administrateur

Pour ajouter une filière, il suffit de cliquer sur Ajouter une filière et de remplir le formulaire en entrant l'abréviation correspondante.

The screenshot shows the 'Ajouter Une Filière' form. It features a text input field labeled 'Abbréviation du secteur' and a blue 'Ajouter' button below it.

figure 18 : formulaire d'ajout d'une filière

## Interface de gestion des départements

L'Administrateur peut consulter les professeurs et les modules de chaque département, et les ajouter soit par un fichier csv ou bien individuellement, ainsi de l'affectation des modules aux professeurs.

**EIT**  
Département d'Electronique Informatique et Télécommunications

[Voir Modules](#) [Ajouter Module](#) [Voir Professeurs](#) [Ajouter Professeur](#) [Affecter Modules Au Professeur](#)

<input type="checkbox"/>	Module Name	Name By Department	Level
<input type="checkbox"/>	Electrotechnique de base	Electrotechnique de base	GELC3
<input type="checkbox"/>	Système à microprocesseur	Système à microprocesseur	GELC3
<input type="checkbox"/>	Capteurs et Instrumentatio...	Capteurs et Instrumentatio...	GELC3
<input type="checkbox"/>	Electronique de puissance 1	Electronique de puissance 1	GELC3
<input type="checkbox"/>	Automates programmables...	Régulation industrielle et A...	GELC4

figure 19 : liste des modules

[Voir Modules](#) [Ajouter Module](#) [Voir Professeurs](#) [Ajouter Professeur](#) [Affecter Modules Au Professeur](#)

<input type="checkbox"/>	First Name	Last Name	Full name
<input type="checkbox"/>	MOHAMMED	AISSAOUI	MOHAMMED AISSAOUI
<input type="checkbox"/>	RAMDANE	BADDI	RAMDANE BADDI
<input type="checkbox"/>	BENAISSA	BELLACH	BENAISSA BELLACH
<input type="checkbox"/>	AHMED	BENLGHAZI	AHMED BENLGHAZI
<input type="checkbox"/>	TOUMI	BOUCHENTOUF	TOUMI BOUCHENTOUF

figure 20 : liste des professeurs



**Ajouter modules via Excel**

Parcourir... Aucun fichi...lectionné.

Importer

OR

**Ajouter module**

Nom De Module

Intitulé De Module

Nom By Département

Sélectionner le département

Sélectionner le professeur

Sélectionner le niveau

Ajouter

figure 21 : formulaire d'ajout des modules

**Ajouter professeurs via Excel**

Parcourir... Aucun fichi...lectionné.

Importer

OR

**Ajouter Professeur**

Nom

Prénom

Email

Password

Sélectionner le département

Ajouter

figure 22 : formulaire d'ajout des professeurs

figure 23: formulaire d'affectation

## interface de suivi d'absence

L'Administrateur peut consulter les modules de chaque niveau et prend une vue détaillée sur l'absence du module; la carte de module devient rouge si le total d'absence pour ce module a dépassé 3 absences. Il peut ajouter la liste des étudiants, l'emploi du temps et les photos des étudiants de chaque niveau.

GINF4	
Emploi du temps	Liste des étudiants
<b>Atelier Usine Logiciel</b> Pr.TOUMI BOUCHENTOUF Nombre d'absence: 0	<b>Applications et Programmations réparties</b> Pr.ZAKARIAE HAJA Nombre d'absence: 0
<b>Dev APP Av</b> Pr.LEILA KELMOUA Nombre d'absence: 0	<b>Interconnexion et Administration des réseaux</b> Pr.MOHAMMED SABER Nombre d'absence: 3
<b>Communication et Compétences</b> Pr.MOHAMMED AMINE BADDOU Nombre d'absence: 0	<b>Communication en anglais et milieu professionnel</b> Pr.HANAE BELHASSANI Nombre d'absence: 0
<b>Management fondamentale et outils de management de projet</b> Pr.MERYEM BELHASSANI	<b>Marketing fondamental</b> Pr.MERYEM BELHASSANI Nombre d'absence: 0

figure 24 : suivi d'absence pour un niveau

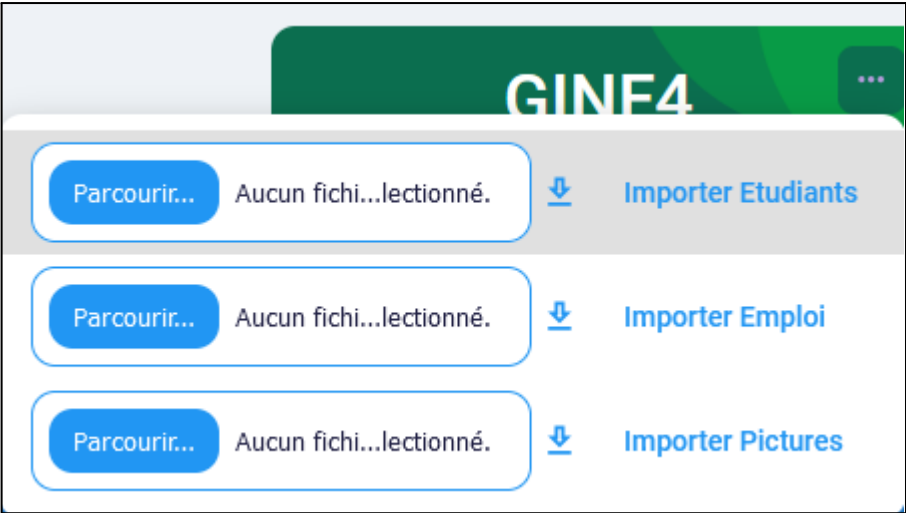


figure 25 : gestion du niveau

<div> <div>Total</div> <div>Cours</div> <div>TD</div> <div>TP</div> </div>				
Avatar	Apogee	Name	Nombre d'absence	
	4571	Yasmin Rachidi	0	
	4584	Saïd Kabbaj	1	
	4577	Samira Bouzid	0	
	4585	Farid Ghanem	1	

figure 26 : Liste des étudiants absents

Date	Type de séance	Justifié	
2024-06-23	Cours	Non	<input type="checkbox"/>
2024-06-23	Cours	Non	<input type="checkbox"/>
2024-06-23	Cours	Non	<input type="checkbox"/>

figure 27 : listes des absences d' étudiant

## interface de paramétrage

Une barre de navigation comprend une icône de paramètres permettant d'accéder aux réglages d'administrateur ou de terminer la session de connexion.

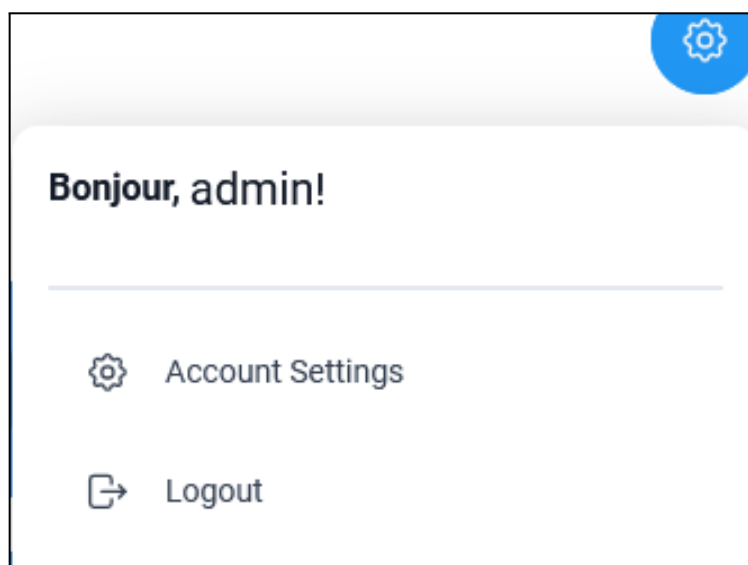


figure 28 : menu déroulant

ainsi que de modifier ses informations, telles que l'email et le mot de passe. Il peut aussi réinitialiser les données en supprimant toutes les absences, les emplois du temps ainsi que les étudiants.

A screenshot of the "Paramètres" (Settings) page for an administrator. The page has a light blue background. At the top, it says "Paramètres" and "Changer l'email". Below this is a text input field containing "admin@gmail.com" and a blue button labeled "Confirmer Le Nouvel Email". The next section is "Changer le mot de passe", featuring two text input fields: "Nouveau mot de passe" and "Confirmer le mot de passe", followed by a blue button labeled "Confirmer Le Nouveau Mot De Passe". The final section is "Réinitialiser les données", with a subtitle "Cette section vous permet de supprimer toutes les absences, emplois du temps ainsi que les étudiants." and a red button labeled "Réinitialiser Les Données".

figure 29 : interface pour modifier les informations d'administrateur

## ❖ interface professeur

### Interface d'accueil

Après l'authentification, le professeur peut accéder à la page d'accueil qui affiche la séance en cours, si celle-ci est active au moment de sa connexion.

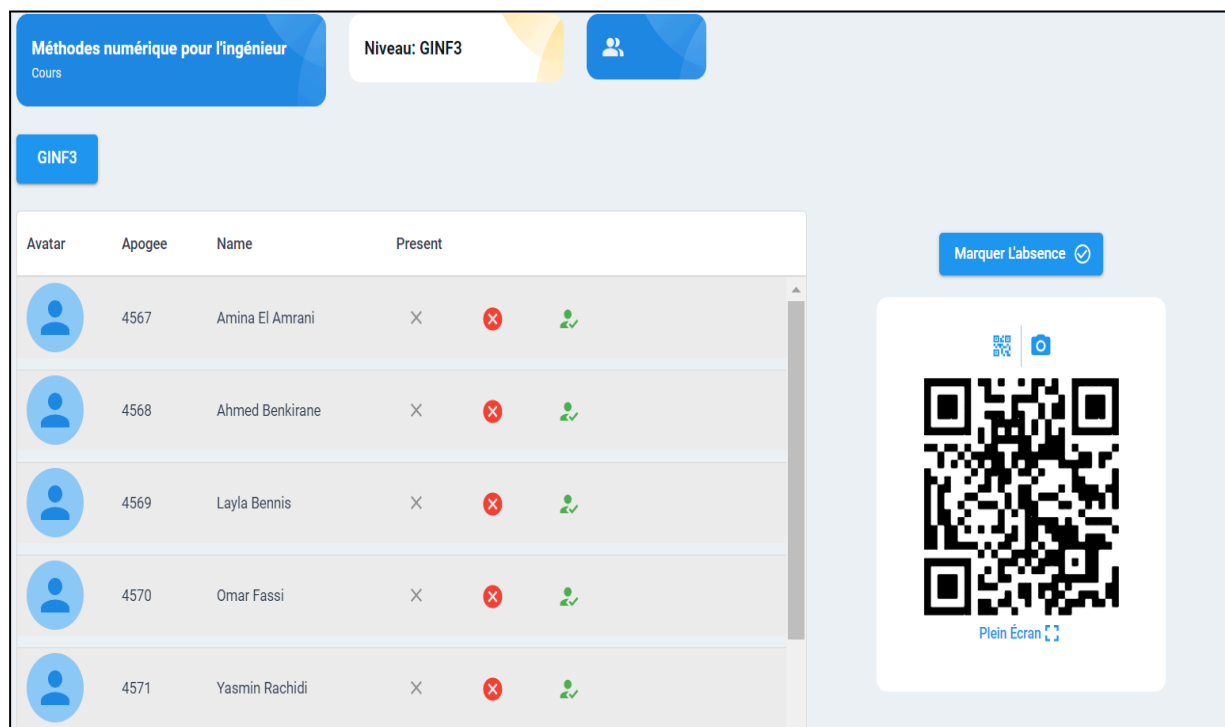


figure 30 : interface d'accueil de professeur

La page d'accueil indique si la séance en cours est de type cours, TP (travaux pratiques) ou TD (travaux dirigés), ainsi que le créneau de la séance

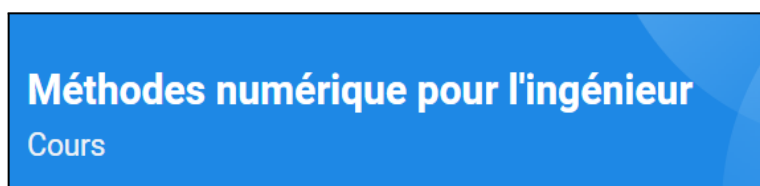


figure 31 : composant pour afficher le module actuelle

Il précise le niveau ou les niveaux concernés lors d'une séance partagée par plusieurs niveaux, ainsi que le groupe lors d'une séance par groupe spécifique.



figure 32 : composant pour afficher les niveaux

Un compteur qui enregistre le nombre d'étudiants ayant marqué leur présence.



figure 33 : composant pour afficher les étudiants ayant marqué la présence

la liste des étudiants du niveau concerné par la séance actuelle, ou les listes des étudiants en cas d'une séance partagée avec plusieurs niveaux.

Avatar	Apogee	Name	Present		
	4567	Amina El Amrani	✓	✗	
	4568	Ahmed Benkirane	✓	✗	
	4569	Layla Bennis	✓	✗	
	4570	Omar Fassi	✓	✗	
	4571	Yasmin Rachidi	✓	✗	

figure 34 : liste des étudiant de séance actuelle

La liste des étudiants indique ceux qui ont marqué leur présence. Elle permet au professeur de confirmer ou d'annuler l'absence dans les cas où l'étudiant n'a pas pu marquer sa présence ou en cas de fraude.

La liste des boutons permet de naviguer entre les niveaux pour accéder aux listes des étudiants et aux codes QR.

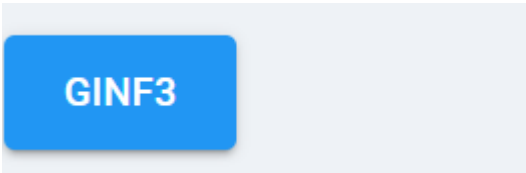


figure 35 : liste des boutons pour naviguer entre les niveaux

Le code QR de la séance est l'outil principal utilisé par défaut pour enregistrer la présence



figure 36 : composant pour afficher le code QR de la séance

Les boutons "Caméra" et "QR Code" permettent de switcher entre les méthodes (QR Code ou reconnaissance faciale).



figure 37 : liste des boutons pour choisir la méthode de marquage d'absence

Lorsque la reconnaissance faciale est sélectionnée, le bouton de la caméra est activé pour utiliser cette méthode de marquage de présence.

Un bouton pour marquer l'absence après avoir vérifié que le nombre d'étudiants présents correspond à celui enregistré dans le compteur.

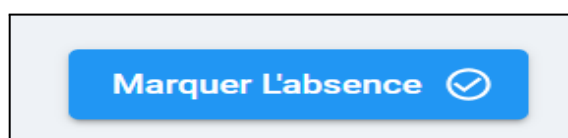


figure 38 : Boutons pour marquer l'absence



Un dialogue de confirmation pour le marquage d'absence, visant à éviter les erreurs de frappe.



figure 39 : Dialogue pour confirmer le marquage d'absence

Une barre latérale affichant la liste des modules enseignés, permettant d'accéder aux statistiques détaillées sur les absences des étudiants.

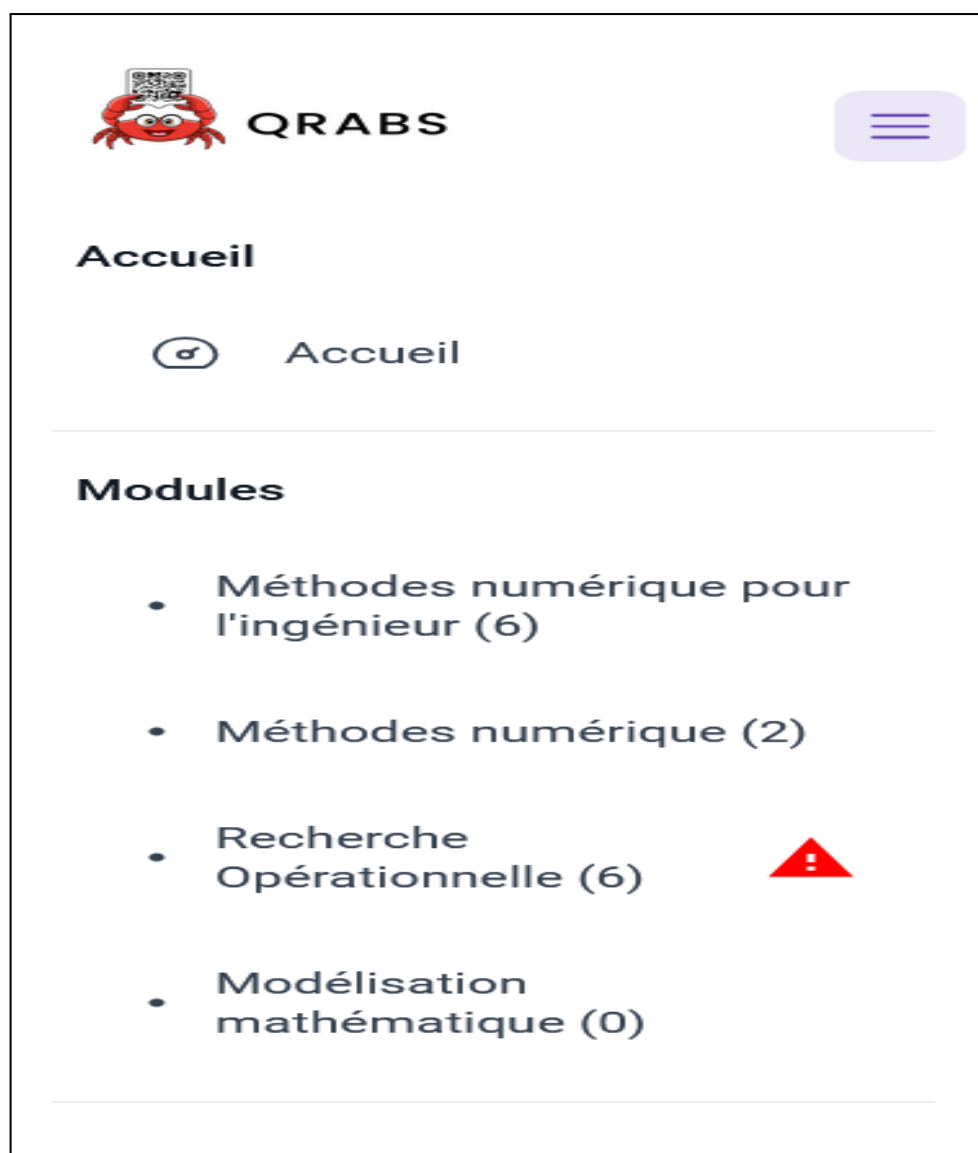


figure 40 : Barre latérale

Chaque module indique le nombre total d'absences pour ce niveau, ainsi qu'une icône d'avertissement signalant qu'au moins un étudiant a dépassé 3 absences non justifiées.

## Interface de suivi d'absence

La page de liste des absences présente la liste des étudiants avec leur nombre total d'absences non justifiées dans le module, ainsi que le décompte des absences lors des séances de cours, des travaux pratiques (TP) et des travaux dirigés (TD).









<div>TotalCoursTDTP</div>				
Avatar	Apogee	Name	Nombre d'absence	
	4571	Yasmin Rachidi	0	
	4584	Saïd Kabbaj	1	
	4577	Samira Bouzid	0	
	4585	Farid Ghanem	1	

figure 41 : Liste des étudiants absents

Le professeur peut consulter les détails des absences de chaque étudiant, incluant la date de l'absence, le type de séance manquée (cours, TP, TD), ainsi que l'état de justification pour déterminer si l'absence est justifiée ou non. Le professeur peut également justifier l'absence.

Date	Type de séance	Justifié	
2024-06-23	Cours	Non	<input type="checkbox"/>
2024-06-23	Cours	Non	<input type="checkbox"/>
2024-06-23	Cours	Non	<input type="checkbox"/>

figure 42 : listes des absences d' étudiant

En plus de cela, le professeur peut accéder à davantage d'informations sur l'étudiant, telles que sa photo de profil, son nom, son niveau d'étude, son email, etc.



figure 43 : Composant affichant les informations d'étudiant

## Interface de paramétrage

Une barre de navigation comprend une icône de paramètres permettant d'accéder aux réglages du professeur ou de terminer la session de connexion.

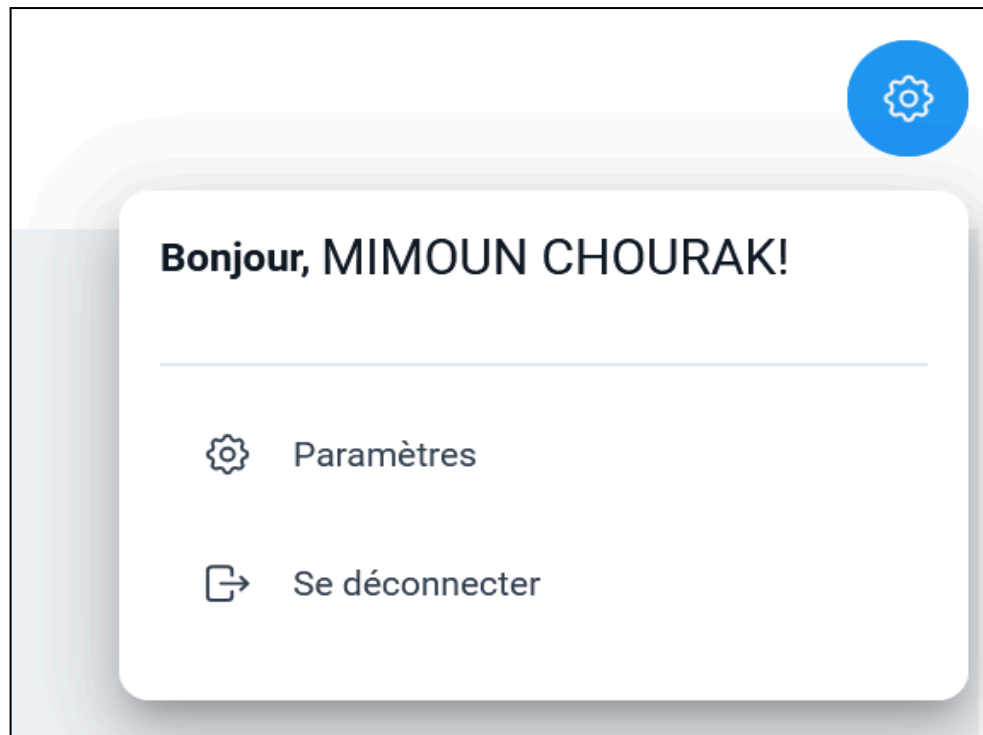


figure 44 : menu déroulant

La page des paramètres permet de gérer les absences d'une séance reportée de son créneau original,

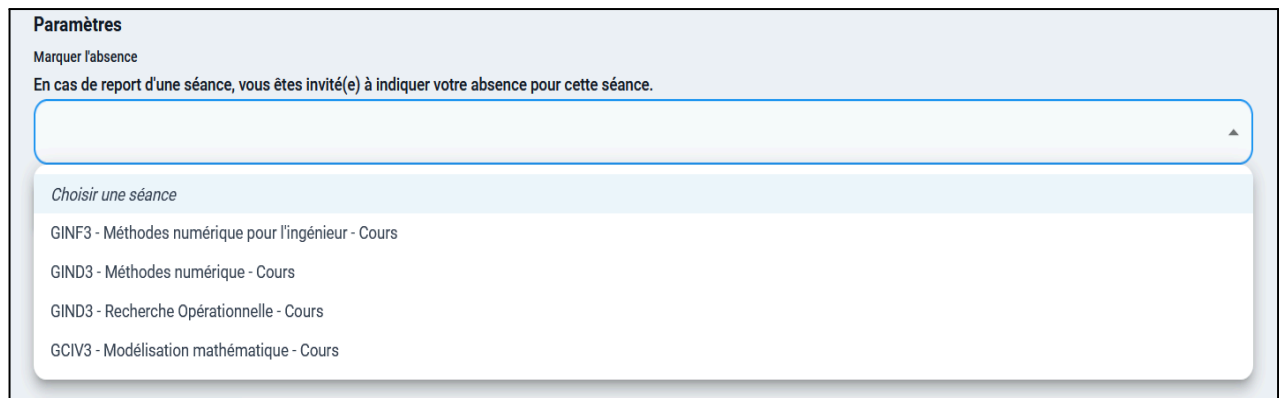


figure 45 : menu déroulant des séances du professeur

ainsi que de modifier les informations de l'utilisateur, telles que l'email et le mot de passe.

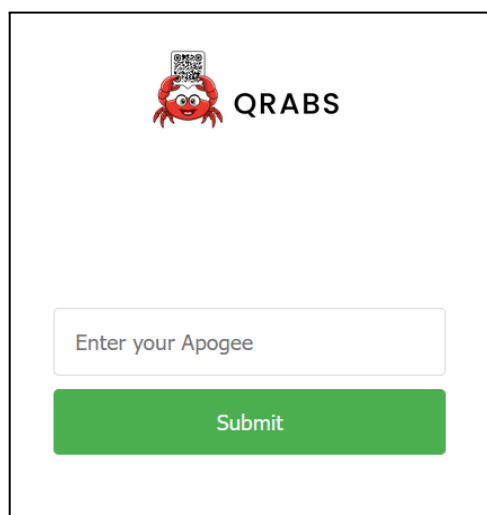


The screenshot shows a web form with two sections. The first section, titled 'Changer l'email', contains a text input field with the placeholder 'Nouvelle adresse e-mail' and the value 'Nomprofesseur@gmail.com', followed by a blue button labeled 'Confirmer Le Nouvel Email'. The second section, titled 'Changer le mot de passe', contains two text input fields: the first with the placeholder 'Nouveau mot de passe' and the second with the placeholder 'Confirmer le mot de passe', followed by a blue button labeled 'Confirmer Le Nouveau Mot De Passe'.

figure 46 : interface pour modifier les données du professeur

### ❖ interface étudiant

Lorsque l'étudiant scanne le code QR, il est redirigé vers une page web pour saisir son numéro d'Apogée.



The screenshot shows a web form with a QR code icon and the text 'QRABS' at the top. Below this is a text input field with the placeholder 'Enter your Apogee' and a green button labeled 'Submit'.

figure 47 : forme pour remplir l'apogée d'étudiant

Une fois le numéro d'Apogée saisi, une boîte de dialogue de confirmation des informations de l'étudiant s'affiche.

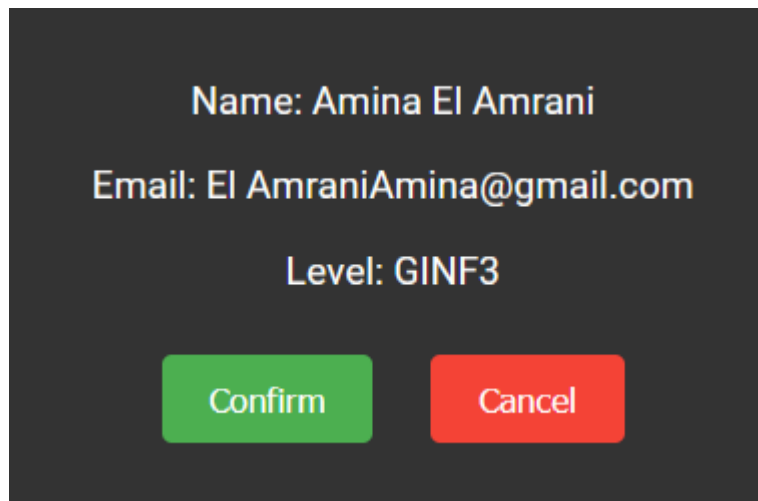


figure 48 : boîte de dialogue de confirmation des informations de l'étudiant

Après confirmation, la présence sera marquée si l'étudiant est enregistré dans la séance. Sinon, le marquage sera refusé. De plus, les données de l'étudiant seront enregistrées dans le navigateur pour simplifier le processus : la présence sera marquée automatiquement sans avoir besoin de saisir à nouveau le numéro d'Apogée.

## Conclusion générale

Dans ce rapport, nous avons présenté notre projet de développement d'une application web de gestion des absences pour l'ENSAO. Cette expérience a été enrichissante, nous permettant d'appliquer nos connaissances en développement logiciel et en gestion de projets.

L'application que nous avons réalisée offre une solution complète et efficace pour la gestion des absences, simplifiant ainsi le processus de suivi des étudiants, la gestion des informations des professeurs, des modules et des emplois du temps. Tout au long du développement, nous avons dû relever divers défis techniques, que nous avons surmontés avec détermination afin de garantir le bon fonctionnement de l'application.

En conclusion, ce projet a été une opportunité pour nous de mettre en pratique nos compétences et d'apporter une solution innovante et utile à notre école. Nous tenons à exprimer notre gratitude à notre encadrant pour son soutien et ses conseils tout au long de ce projet. Nous sommes convaincus que cette application apportera une réelle valeur ajoutée à la gestion des présences et des absences dans les établissements éducatifs et ouvrira la voie à de futures innovations dans ce domaine.