ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

Projet de Fin d'Année

4eme année
Ingénierie Informatique et Réseaux

Etude et developpement d'un systeme de l'association des étudiants

---

Réaliser par :

AIT EL QADI Anas

IZMAR Mohamed Taha

Encadré par :

M. EL HASSANI Mouhcine

SOUTENU LE 30/06/2025 DEVANT LE JURY D'EXAMEN :

M. EL HASSANI Mouhcine

M. ALAMI Mohamed

Année universitaire 2024/2025

Dédicace

À tous ceux qui ont contribué à la concrétisation de ce projet,nous avons ensemble traversé des épreuves et trouvé des inspirations, partagé des rires et des moments de réflexion.

Ce projet est bien plus qu'une simple réalisation ; il est le reflet de notre dévouement, de notre créativité et de notre esprit de collaboration. Chaque membre de notre groupe a apporté une touche unique, illuminant ce projet de manière spéciale et précieuse.

Que cette dédicace témoigne de notre travail acharné, de notre solidarité et de notre passion commune.Puissions-nous nous souvenir de ce projet comme d'une preuve indélébile de ce que nous sommes capables d'accomplir ensemble.

Remerciements

Nous souhaitons exprimer notre profonde gratitude à tous ceux qui ont contribué à la réalisation de ce projet de fin d'études. Tout d'abord, nous remercions sincèrement Monsieur le Directeur de l'EMSI pour la formation de qualité qu'il nous a offerte. Grâce à son leadership et à sa vision, nous avons bénéficié d'un environnement académique exceptionnel, propice à notre épanouissement et à notre réussite.Nos sincères remerciements vont également à nos professeurs, dont l'expertise, la patience et l'engagement ont été essentiels à notre apprentissage. Leurs enseignements et leurs conseils nous ont guidés tout au long de notre parcours académique, nous permettant de développer les compétences nécessaires pour relever les défis de demain.À nos amis, merci pour votre soutien constant, vos encouragements et pour les moments de camaraderie qui ont rendu cette expérience encore plus mémorable. Votre présence à nos côtés a été une source de motivation et de réconfort.Enfin, nous souhaitons exprimer notre infinie reconnaissance à nos parents. Votre amour, votre soutien inconditionnel et vos sacrifices ont été le pilier sur lequel nous avons construit nos rêves. Sans vous, rien de tout cela n'aurait été possible.Ce projet est le fruit de la collaboration, du travail acharné et du soutien de chacun d'entre vous. Merci de nous avoir permis de réaliser cet accomplissement.

Résumé

Ce rapport discute du processus de développement ainsi que des principales fonctionnalités associées à une application Web de l'association des étudiants. L'objectif fondamental de l'application en question est de fournir une interface web permettant les étudiants et les responsables de recrutement d'avoir un espace pour partager leur besoin , et donner ;es étudiantes des opportunités .

L'implémentation de cette application s'est effectuée en quatre étapes : la planification, la construction, le test et le déploiement. Planification : creation dune interface utilisateur guident leurs interactions avec l'application.

Pour réaliser l'application, des technologies web modernes, y compris HTML, CSS, JavaScript pour le frontend , ont été utilisées. Outre, des Frame-work tels que Bootstrap et spring boot pour le côté backend , ont été utilisés pour garantir une expérience utilisateur transparente et réactive. Des tests ont été effectués à chaque étape du cycle de vie de développement pour assurer la qualité et la fiabilité de l'application. Il s'agissait de tests unitaires, d'intégration et de régression pour identifié les erreurs ou les problèmes de performance.

En fin de compte, cette application propose une solution viable et facile a utiliser aux fins de perimetre defini.

Abstract

This report discusses the development process and the key features associated with a student association web application. The fundamental objective of this application is to provide a web interface that allows students and hiring managers to share their needs and provide opportunities for students.

The implementation of this application was carried out in four stages : planning, building, testing, and deployment. Planning : Creating a user interface to guide their interactions with the application.

To create the application, modern web technologies, including HTML, CSS, and JavaScript for the frontend, were used. In addition, frameworks such as Bootstrap and Spring Boot were used for the backend to ensure a seamless and responsive user experience. Testing was conducted at each stage of the development lifecycle to ensure the quality and reliability of the application. These included unit, integration, and regression testing to identify errors or performance issues.

Ultimately, this application provides a viable and easy-to-use solution for the purposes of a defined scope.

Table des matières

Dédicace **2**

Remerciements **3**

Résumé **4**

Abstract **5**

Liste des sigles et acronymes **11**

Introduction générale **12**

1 Contexte général du projet **13**
1.1 Périmètre du projet 14
1.1.1 But 14
1.1.2 Missions 14
1.1.3 Contraintes 15
1.1.4 Tableau des Sprints 15
1.2 Étude de l'existant 16
1.2.1 Concept métier 16
1.2.2 Logiciels similaires 16
1.3 Étude fonctionnelle 17
1.3.1 Objectifs fonctionnels 17
1.3.2 Besoins fonctionnels 18
1.4 Conclusion 21

2 Analyse et conception **22**
2.1 Introduction 23
2.2 Spécification Fonctionnelle Générale 23
2.2.1 Objectif 23
2.2.2 Présentation du système 23
2.2.3 Acteurs du système 23
2.2.4 Modules fonctionnels 24
2.2.5 Relations entre cas d'utilisation 24
2.2.6 Règles de gestion globales 24
2.2.7 Messages standard (globaux) 25
2.2.8 Diagrammes de cas d'utilisation par module 25
2.3 Spécification Fonctionnelle Détaillée 31
2.3.1 Diagramme de classe 31

2.3.2 Cas d'utilisation détaillés . . . 33
2.4 Conclusion . . . 35

3 Etude pratique **36**
3.1 Introduction . . . 37
3.2 Outils . . . 37
3.2.1 Html . . . 37
3.2.2 css . . . 39
3.2.3 bootstrap . . . 41
3.2.4 js . . . 42
3.2.5 staruml . . . 43
3.2.6 Écosystème Spring . . . 45
3.2.7 Architecture spring en Couches . . . 46
3.2.8 Language Java . . . 47
3.2.9 MySQL . . . 48
3.2.10 MAMP . . . 49
3.2.11 git/github . . . 50
3.2.12 Postman API . . . 51
3.2.13 intellij idea ultimate . . . 52
3.2.14 Mailtrap.io . . . 53
3.3 Réalisation du projet . . . 54
3.3.1 Page d'accuei . . . 54
3.3.2 page de connexion . . . 55
3.3.3 Page d'Inscription . . . 56
3.3.4 Page des utilisateurs (pour admin) . . . 57
3.3.5 Page "Tableau de Bord" . . . 58
3.3.6 page de profil . . . 59
3.3.7 Page de modification du Profil . . . 60
3.3.8 Page des Offres . . . 61
3.3.9 Page du Détails de l'Offre . . . 61
3.3.10 Page d'ajoute d'une Offre . . . 62
3.3.11 Page des des Événements . . . 63
3.3.12 Page des Détails de l'Événement . . . 64
3.3.13 Page d'ajout d'un Événement . . . 64
3.3.14 Page des documents . . . 65
3.3.15 Page d'ajout d'un document . . . 66
3.3.16 les "e-mail" automatiquement envoyé . . . 67
3.4 Conclusion . . . 69

Conclusion générale **70**

Perspectives **71**

Webographie **72**

Bibliographie **73**

Table des figures

2.1 diagramme de cas d'utilisation "utilisateurs" 26
2.2 diagramme de cas d'utilisation "Gestion des Étudiants et Inscription" . . 27
2.3 diagramme de cas d'utilisation "Gestion des offres" 28
2.4 diagramme de cas d'utilisation "Gestion des événements" 29
2.5 diagramme de cas d'utilisation "Gestion des documents" 30
2.6 diagramme de cas d'utilisation "Gestion des Statistiques" 31
2.7 diagramme de classe 32
2.8 diagramme de séquence cas 1 33
2.9 diagramme de séquence cas 2 34
2.10 diagramme de séquence cas 3 35

3.1 Html logo 37
3.2 Structure de base d'un document HTML 38
3.3 css logo 39
3.4 Structure de CSS 40
3.5 bootstrap logo 41
3.6 js logo 42
3.7 staruml logo 43
3.8 spring boot 45
3.9 architecture spring boot 46
3.10 java 47
3.11 MySql 48
3.12 MAMP 49
3.13 git/github 50
3.14 Postman API 51
3.15 intellij idea 52
3.16 Mailtrap 53
3.17 page d'accueil 55
3.18 page de connexion 56
3.19 Page d'Inscription 57
3.20 Page des utilisateurs 57
3.21 rapport des utilisateurs 58
3.22 Tableau de Bord 59
3.23 page de profil 59
3.24 Page de modification du Profil 60
3.25 Page des Offres 61
3.26 Page du Détails de l'Offre 62
3.27 Page d'ajoute d'une Offre 63
3.28 Page des des Événements 63

3.29 Page des Détails de l'Événement 64
3.30 Page d'ajout d'un Événement 65
3.31 Page des documents 65
3.32 Page d'ajout d'un document 66
3.33 Mail de confirmation d'inscription 67
3.34 Mail de notification d'un événement 68
3.35 Mail d'annonce d'une offre 69

Liste des tableaux

1 Tableau des sigles et acronymes 11

1.1 Tableau des sprints et fonctionnalités 15
1.2 Règles de gestion – Étudiant 19
1.3 Règles de gestion – Offres et recruteurs 20
1.4 Spécifications des ressources 20
1.5 Spécifications du tableau de bord 21

2.1 Rôles des acteurs 23
2.2 Relations 24
2.3 Relations 25
2.4 Messages 25

Liste des sigles et acronymes

Sigle	**Description**
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
idea	intellij idea
UML	Unified Modeling Language

Table 1 – Tableau des sigles et acronymes

Introduction générale

Avec l'évolution des besoins numériques au sein des établissements académiques, il devient indispensable de doter les associations étudiantes de plateformes modernes, sécurisées et performantes pour gérer efficacement leurs activités. C'est dans ce cadre que s'inscrit notre projet, réalisé dans le contexte du Projet de Fin d'Année (PFA) de la filière 4IIR à l'EMSI. Il porte sur l'étude et le développement d'une application web dédiée à la gestion complète de l'association des étudiants.

Ce système centralisé a pour vocation de faciliter l'administration des utilisateurs, des documents, des événements, des offres d'emploi, ainsi que la génération de statistiques utiles à la prise de décision. Conçue avec une architecture robuste basée sur Java 17, Spring Boot et Spring Security, et une interface dynamique construite avec Thymeleaf, HTML/CSS et MySQL, l'application assure une expérience fluide et sécurisée pour tous les profils d'utilisateurs.

Notre solution propose une palette de fonctionnalités clés : authentification sécurisée, gestion des documents et des offres d'emploi, tableau de bord dynamique, envoi automatique d'e-mails à la création de comptes, d'offres ou d'événements, ainsi qu'une interface responsive accessible depuis tout support.

Ce rapport présente une analyse complète du processus de développement, des choix technologiques effectués, des étapes de conception jusqu'à l'implémentation des modules fonctionnels. Ce projet nous a permis de mettre en pratique nos compétences en développement full stack, en architecture logicielle et en gestion de projet, tout en répondant à un besoin réel et concret de notre environnement universitaire.

Chapitre 1

Contexte général du projet

1.1 Périmètre du projet

1.1.1 But

L'objectif principal de ce projet est l'étude, la conception et la mise en œuvre d'un système d'information destiné à une association d'étudiants. Ce système vise à centraliser et automatiser la gestion des principales activités de l'association, à travers les modules suivants :

Gestion des étudiants et des inscriptions : suivi des dossiers des étudiants, inscriptions aux événements ou aux services proposés.

Gestion des événements : création, planification et gestion de la participation aux événements organisés par l'association.

Gestion des offres d'emploi et de stage : publication, consultation et gestion des candidatures aux opportunités proposées par les partenaires.

Gestion des ressources : administration des ressources matérielles, humaines ou documentaires de l'association.

Gestion du tableau de bord : visualisation des indicateurs clés et pilotage global du système.

Ce système permettra d'améliorer l'efficacité de l'association, de faciliter la communication avec les membres, et d'optimiser la gestion interne.

1.1.2 Missions

Les missions réalisées dans le cadre de ce projet sont réparties selon les étapes suivantes :

Étude technique : identification des technologies adaptées au développement du système (langages, frameworks, base de données, etc.).

Étude fonctionnelle : recueil et analyse des besoins des utilisateurs finaux et définition des cas d'usage.

Conception : modélisation des données, architecture logicielle, conception de l'interface utilisateur.

Réalisation : développement des différents modules conformément aux spécifications.

Déploiement : mise en production du système dans un environnement stable et accessible.

Documentation : rédaction des documents techniques et guides utilisateurs afin d'assurer la pérennité et la maintenabilité du système.

1.1.3 Contraintes

Le projet est soumis à un certain nombre de contraintes qu'il convient de respecter rigoureusement :

Délais : chaque étape du projet doit être réalisée dans un calendrier précis, souvent limité par la durée du stage ou du semestre.

Budget : l'utilisation de ressources (hébergement, licences, outils) doit respecter les limites financières imposées.

Technologie : les choix technologiques doivent tenir compte des compétences disponibles, de la compatibilité entre composants et de l'évolutivité du système.

Lois et règlements : le projet doit se conformer à la législation en vigueur, notamment en matière de protection des données personnelles (RGPD).

1.1.4 Tableau des Sprints

Sprint	**Phase**	**Fonctionnalité**	**Date de livraison**	**Responsables**
0	initiale	- Création de cahier des charges	05/05/2025	Ingénieures IT
1	- conception - réalisation - Test	- Conception du projet - Gestion des étudiants / inscription	19/05/2025	Ingénieures dev
2	- conception - réalisation - Test	- Gestion des événements - Gestion du tableau de bord	02/06/2025	Ingénieures dev
3	- conception - réalisation - Test	- Gestion ressources - Gestion des offres d'emploi/stage	16/06/2025	Ingénieures dev
4	- conception - réalisation - Test	- amélioration - Test final	23/06/2025	Ingénieures dev

Table 1.1 – Tableau des sprints et fonctionnalités

1.2 Étude de l'existant

1.2.1 Concept métier

Le concept métier de l'application repose sur l'amélioration de la gestion administrative, collaborative et événementielle d'une association étudiante. Elle vise à centraliser et automatiser les processus essentiels tout en offrant une interface intuitive aux différents utilisateurs (administrateurs, membres, étudiants, entreprises partenaires).

Les principaux besoins métier sont : Gestion des offres d'emploi et de stage : Permettre aux entreprises et aux étudiants de publier, consulter et postuler à des offres en toute simplicité.

Gestion des ressources : Suivre et administrer les ressources mises à disposition par l'association (documentation, etc...).

Gestion des événements : Organiser et planifier des événements (conférences, ateliers, networking), avec des fonctionnalités d'inscription et de suivi des participants.

Problématiques adressées Ce concept vise à répondre aux défis courants rencontrés par les associations étudiantes, notamment :

La dispersion des informations et des opportunités pour les étudiants.

Les difficultés d'organisation et de communication interne.

Le manque de suivi et d'optimisation des ressources et événements.

L'application offrira ainsi une solution centralisée et efficace pour améliorer la gestion et la coordination des activités au sein de l'association.

1.2.2 Logiciels similaires

Une analyse des plateformes existantes permet d'identifier leurs forces et faiblesses afin d'inspirer le développement de l'application. EMSI Community est une solution pertinente à étudier dans ce contexte.

1. EMSI Community
Description :

Plateforme dédiée aux étudiants de l'EMSI, permettant le partage d'informations, la mise en relation entre étudiants et entreprises, ainsi que la gestion d'événements et de groupes.

Forces :

Interface adaptée aux étudiants de l'EMSI.
Gestion des groupes et des événements académiques.
Plateforme collaborative facilitant les échanges entre étudiants.

Faiblesses :

Fonctionnalités limitées en matière de gestion des ressources et des offres d'emploi/stage.

2. Analyse comparative :

L'application EMSI Community répond à plusieurs besoins étudiants, mais présente certaines limites qui peuvent être optimisées dans le cadre d'une application de gestion d'association étudiante plus générale.

L'application proposée devra :

Intégrer une gestion avancée des offres d'emploi et de stage, en facilitant les interactions entre étudiants et entreprises.

Optimiser la gestion des groupes, en permettant une organisation plus fluide des échanges et des activités internes.

Mettre en place une gestion efficace des ressources, afin d'offrir aux étudiants un accès centralisé aux documents et supports académiques.

Améliorer la gestion des événements, avec des outils interactifs pour planifier, suivre et promouvoir les événements étudiants.

L'objectif est de développer une plateforme qui dépasse les fonctionnalités actuelles d'EMSI Community, en proposant un outil plus flexible et personnalisable, capable de s'adapter aux besoins de différentes associations étudiantes.

1.3 Étude fonctionnelle

1.3.1 Objectifs fonctionnels

Les objectifs fonctionnels définissent les grandes lignes des fonctionnalités à intégrer dans l'application afin de répondre aux problématiques spécifiques de gestion d'une association étudiante :

1. Simplification des processus administratifs

Centraliser la gestion des offres d'emploi et de stage, des groupes étudiants, des événements et des ressources disponibles.
Offrir un espace structuré pour la communication et le partage d'informations entre les membres.

2. Automatisation des tâches courantes

Réduire la charge de gestion grâce à des notifications automatiques pour les événements, les candidatures aux offres et les mises à jour des groupes.
Optimiser l'organisation des activités en intégrant une gestion des inscriptions et des rappels automatisés. **3. Amélioration de l'expérience utilisateur**

Proposer une interface intuitive et fluide, adaptée aux besoins des étudiants et des responsables d'association.
Faciliter l'accès aux différentes fonctionnalités avec une navigation simplifiée et un tableau de bord personnalisé.

4. Accès sécurisé et confidentialité

Mettre en place une gestion des accès selon les rôles des utilisateurs (étudiants, administrateurs, entreprises).
Assurer la protection des données personnelles et des informations partagées sur la plateforme.

5. Génération de statistiques et rapports

Fournir des indicateurs clés sur l'engagement des étudiants, la participation aux événements et le suivi des offres d'emploi/stage.
Permettre aux administrateurs d'analyser l'évolution des activités et d'adapter les services proposés en conséquence.

L'objectif est de créer une plateforme complète et performante qui facilite la gestion d'une association étudiante tout en améliorant la communication et l'interaction entre ses membres.

1.3.2 Besoins fonctionnels

L'application Web de gestion de l'association des étudiants est conçue pour répondre aux besoins spécifiques des différents membres de l'association. Chaque utilisateur dispose de fonctionnalités adaptées à son rôle, afin d'assurer une organisation efficace et une communication fluide au sein de l'association. Voici les fonctionnalités principales par type d'utilisateur :

Bloc fonctionnel : Gestion des étudiants / inscription

1.User stories :

En tant que visiteur, le système me permet de :

Créer un compte.

Se connecter et accéder à son espace personnel.

En tant qu'étudiant, le système me permet de :

Consulter le profile.
Modifier son profil (ajouter CV, compétences, etc.).
Se déconnecter.

En tant qu'un administrateur, le système me permet de :

Lister les profils existants.
Consulter detail d'un profil.
Editer un profil.
Supprimer un profil.
Consulter les statistiques.

2.Règles de gestion :

Référence	Description
RG 01	Structure d'un Étudiant : - Matricule - Nom - Prénom - Mail - adresse - Numéro de téléphone - mot de passe
RG 2	un étudiant peut avoir un seul compte
RG 3	un e-mail de bienvenue doit être envoyé à la création du compte
RG 4	Le mot de passe doit contenir au minimum 8 caractères, incluant au moins une majuscule, une minuscule, un chiffre et un caractère spécial

Table 1.2 – Règles de gestion – Étudiant

Bloc fonctionnel : Gestion des offres d'emploi/stage

1.User stories :

En tant qu'étudiant, le système me permet de :

Consulter les offres d'emploi et de stage.
Postuler aux offres disponibles.

En tant que responsable de recrutement/RH , le système me permet de :

ajouter les offres d'emploi et de stage.

2. Règles de gestion :

Référence	Description
RG 01	Structure d'une offre : - Titre - Nom de la société - Affiche - Type d'emploi - Fonction du Poste - Secteur d'activité - Description - Lien ou mail pour postuler
RG 2	Structure d'un responsable de recrutement : - Matricule - Nom - Prénom - Nom de la société - Photo - Mail - Ville - Numéro de téléphone - Compte LinkedIn
RG 3	une offre doit avoir une date limite
RG 4	à la création d'une offre, un e-mail doit être envoyé à l'utilisateur
RG 5	un étudiant peut postuler une seule fois à chaque offre

Table 1.3 – Règles de gestion – Offres et recruteurs

Bloc fonctionnel : Gestion ressources

1.User stories :

En tant qu'administrateur le système me permet de :

Partager des resources .

En tant qu'étudiant, le système me permet de :

Consulter les resources partager.
Télécharger les resources partager.

2. Règles de gestion :

Référence	Description
RG 01	Structure d'une ressource : - ID - Nom - Description - Documents (PDF, Word, Excel, ...)
RG 2	La taille du document ne doit pas dépasser (3-6) Mo
RG 3	Avant d'enregistrer le document, un test des codes malveillants doit être réalisé (test de sécurité)

Table 1.4 – Spécifications des ressources

Bloc fonctionnel : Gestion ressources

1.User stories :

En tant qu'administrateur , le système me permet de :

Suivre le nombre total de membres inscrits.
Suivre le nombre d'événements créés.
Suivre le nombre de participations aux événements.
Suivre les Activités récentes (nouvelles inscriptions nouvelles publications , ...).

2. Règles de gestion :

Référence	Description
RG 01	Les données doivent être automatiquement mises à jour (temps réel ou périodique)
RG 2	Le tableau de bord doit présenter les informations de manière synthétique et claire
RG 3	Le tableau de bord peut contenir des graphes comme il peut contenir des tableaux
RG 4	L'utilisateur peut choisir le nombre de données affichées par page

Table 1.5 – Spécifications du tableau de bord

1.4 Conclusion

Pour résumer, notre projet sera une réponse adaptée aux besoins grâce à une solution moderne, sécurisée et conviviale. Ce chapitre a posé les fondements de notre approche et des fonctionnalités attendues, établissant ainsi les bases sur lesquelles nous allons développer et mettre en œuvre notre solution innovante.

Chapitre 2

Analyse et conception

2.1 Introduction

La conception joue un rôle essentiel dans la réalisation de solutions innovantes. Que ce soit dans le domaine de l'ingénierie, de l'architecture, du design ou d'autres domaines créatifs, la conception est le processus clé qui permet de transformer une idée abstraite en une réalité tangible. Ce chapitre se concentre sur l'importance de la conception et explore les différentes étapes et approches nécessaires pour concevoir des solutions innovantes. En résumé, ce chapitre de conception nous permettra de plonger dans l'univers de notre application en commençant par l'élaboration du diagramme de cas d'utilisation et terminer par le diagramme de séquence .

2.2 Spécification Fonctionnelle Générale

2.2.1 Objectif

Cette partie a pour objectif de spécifier les fonctionnalités principales de la plateforme de gestion de l'association des étudiants. Il sert de référence pour la conception fonctionnelle, la validation des exigences et la coordination entre les parties prenantes.

2.2.2 Présentation du système

La plateforme permet de gérer une association étudiante avec quatre profils utilisateurs :

- Visiteur : peut s'inscrire ou se connecter pour accéder aux services.
- Utilisateur : participe aux événements , consulte et postuler aux offres et aussi consulte ses documents.
- Administrateur : gère les utilisateurs, événements, offres et supervise l'ensemble du système via un tableau de bord.
- RH : Gère les offres et les événements .

2.2.3 Acteurs du système

Acteur	Description
Visiteur	Non connecté, peut s'inscrire ou s'authentifier
Étudiant	Accède à son espace, consulte/modifie son profil, consulte/postule aux offres, consulte les événements, consulte/ajoute/supprime ses documents
Administrateur	Gère les utilisateurs/offres/événements, consulte les statistiques, consulte/modifie son profil
RH	Gère les offres/événements, consulte/modifie son profil

Table 2.1 – Rôles des acteurs

2.2.4 Modules fonctionnels

Gestion des étudiants / inscription

- Consulter, modifier, supprimer son profil (membre) Ajouter, consulter, modifier, supprimer des adhérents (admin)

Gestion des offres d'emploi/stage

- Accéder et consulter les offres publiés Postuler a l'une ou tous les offres disponibles Créer, modifier, supprimer des offres (admin/RH)

Gestion ressources

- Visualiser la liste des resources qui appartient a l'utilisateur connecté Importer , supprimer des resources (utilisateur)

Gestion du tableau de bord

- Visualisation des statistiques(admin) Suivi des utilisateurs , offres , évènement récentes

Gestion des événements

Accéder et consulter les événements publiés Créer, modifier, supprimer des événements (admin/RH)

2.2.5 Relations entre cas d'utilisation

Cas Principal	**«include» / «extend»**
S'inscrire	Validation des champs
Modifier profil	Vérification des données
Créer événement (admin)	Saisie les données
Créer une offre (admin/RH)	Saisie les données

Table 2.2 – Relations

2.2.6 Règles de gestion globales

RG1	a la créations d'une offre un e-mail doit être envoyé a les utilisateur
RG2	Le mot de passe doit contenir au minimum 8 caractères, incluant au moins une majuscule, une minuscule, un chiffre et un caractère spécial
RG3	un e-mail de bienvenue doit être envoyer a la création du compte

Table 2.3 – Relations

2.2.7 Messages standard (globaux)

Code	Message
MSG 001	Accès refusé
MSG 002	Opération réalisée avec succès
MSG 003	Erreur : données incomplètes

Table 2.4 – Messages

2.2.8 Diagrammes de cas d'utilisation par module

diagramme de cas d'utilisation module : utilisateurs

Utilisateur : Rôle générique pour toute personne interagissant avec le système.

Étudiant : Utilisateur spécifique pouvant consulter des offres, gérer son profil, et postuler.
Administrateur : Gère les offres, les événements, les statistiques, et les profils.
RH (Ressources Humaines) : Ajoute/modifie des offres et gère les événements liés au recrutement.

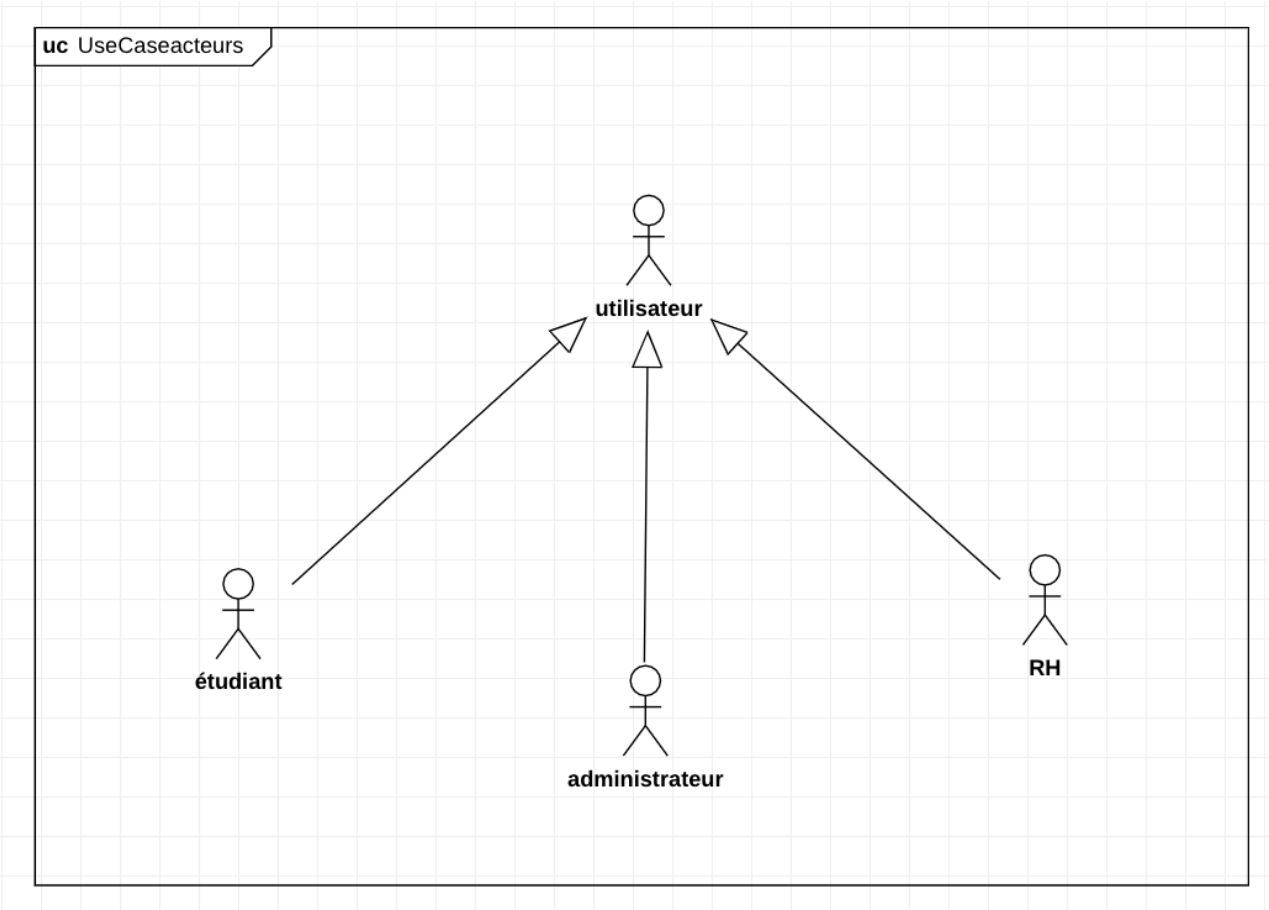

FIGURE 2.1 – diagramme de cas d'utilisation "utilisateurs"

diagramme de cas d'utilisation module : Gestion des Étudiants et Inscription

Fonctionnalités principales :

Inscription/Connexion : Création de compte, connexion, et accès à une page de bienvenue.
Gestion des profils : Consultation, modification, suppression de profils (réservée à l'administrateur).
Statistiques : Tableaux de bord consultables par l'administrateur.

Relations entre cas :

« Extend » pour des extensions optionnelles (ex : se déconnecter après une connexion).
« Include » pour des sous-étapes obligatoires (ex : être connecté pour modifier un profil).

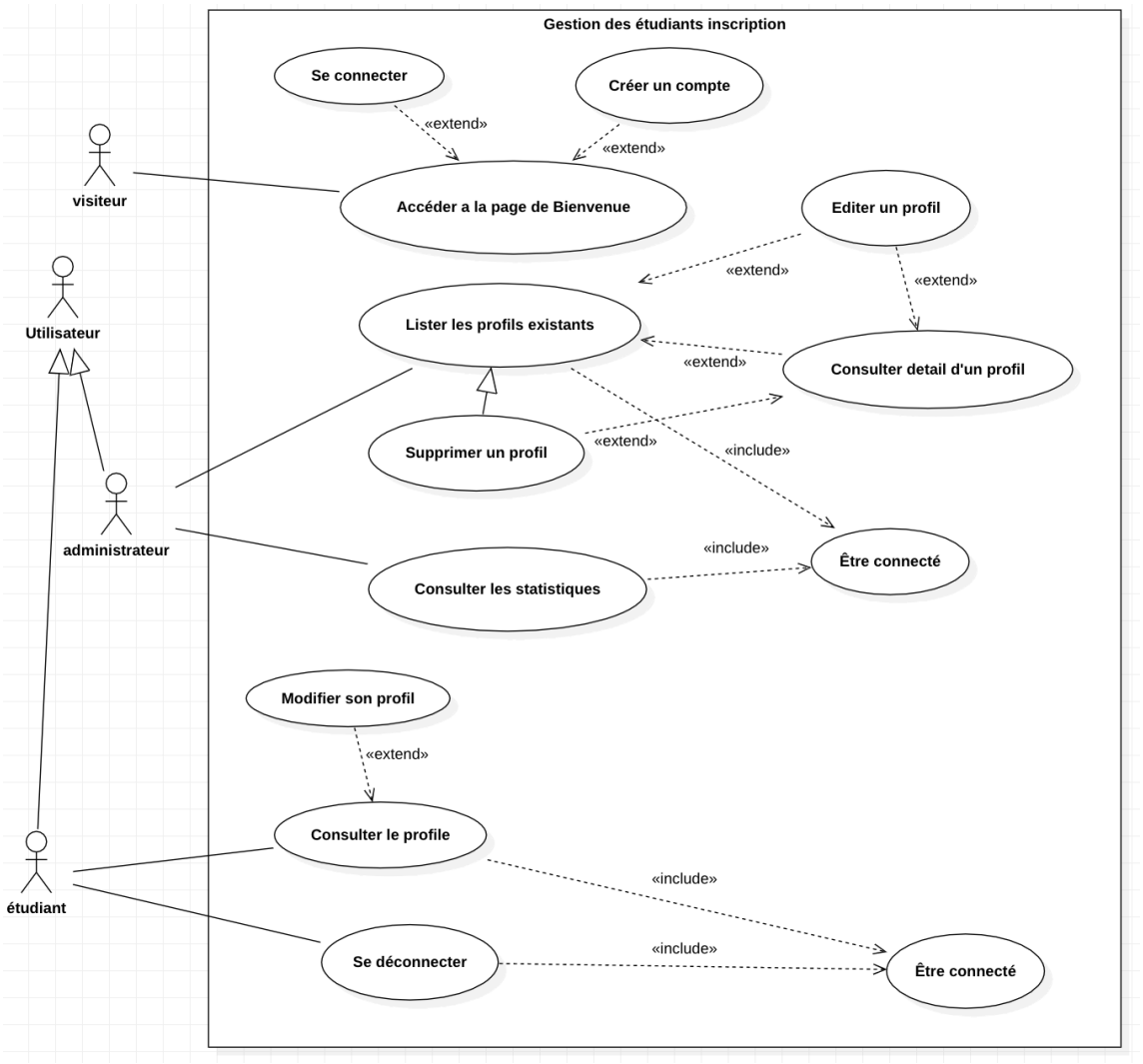

FIGURE 2.2 – diagramme de cas d'utilisation "Gestion des Étudiants et Inscription"

diagramme de cas d'utilisation module : Gestion des offres

Pour les Étudiants :

Consulter la liste des offres et postuler (nécessite d'être connecté).

Pour le RH :

Ajouter/modifier/supprimer des offres.

Pour l'Administrateur :

Gestion complète des offres (validation, suppression).

Termes techniques :

« Externa » indique des interactions externes non détaillées.

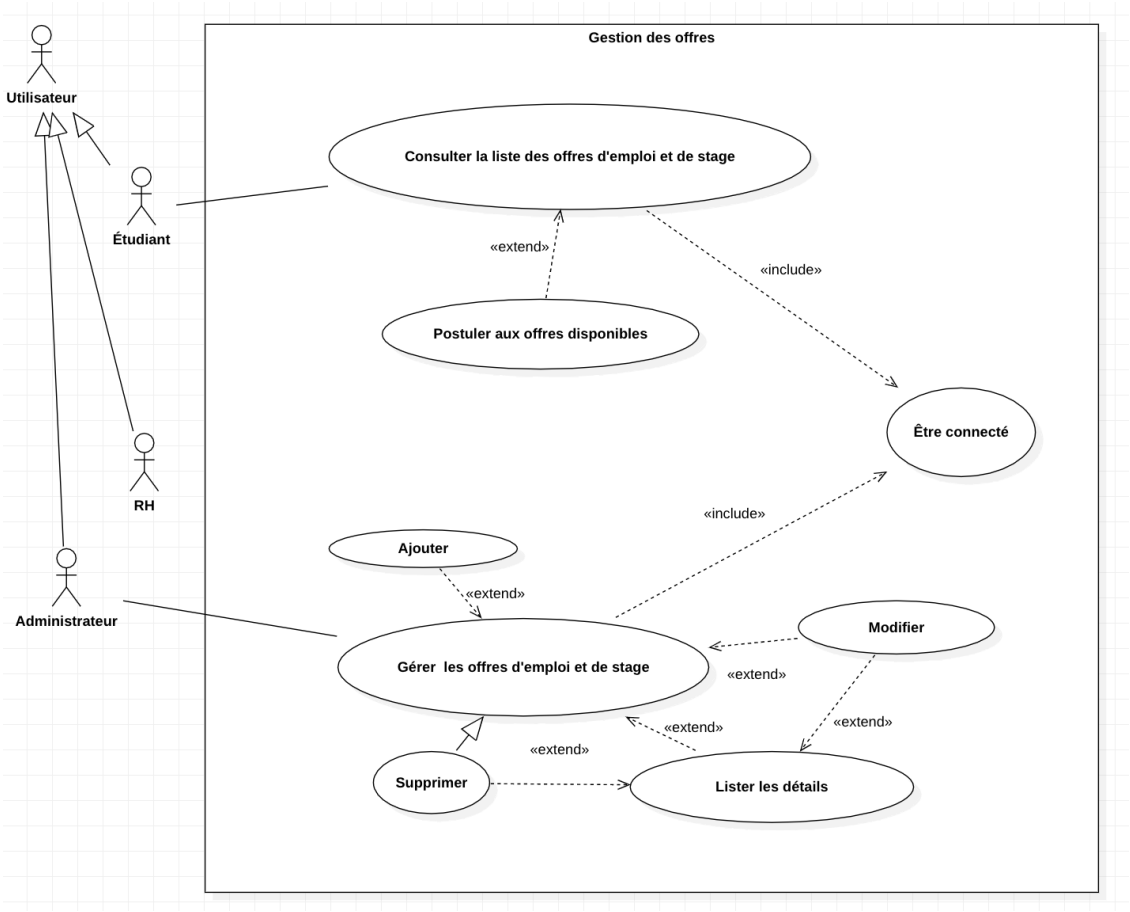

FIGURE 2.3 – diagramme de cas d'utilisation "Gestion des offres"

diagramme de cas d'utilisation module : Gestion des evénements

Participants : Étudiants, RH, Administrateur.

Actions clés :

Ajouter/modifier des événements (RH/Admin).
Consulter les événements partagés (étudiants).
Lister les détails (tous les rôles).

Relations :

« Include » pour des étapes obligatoires (ex : gestion des événements inclut la modification).

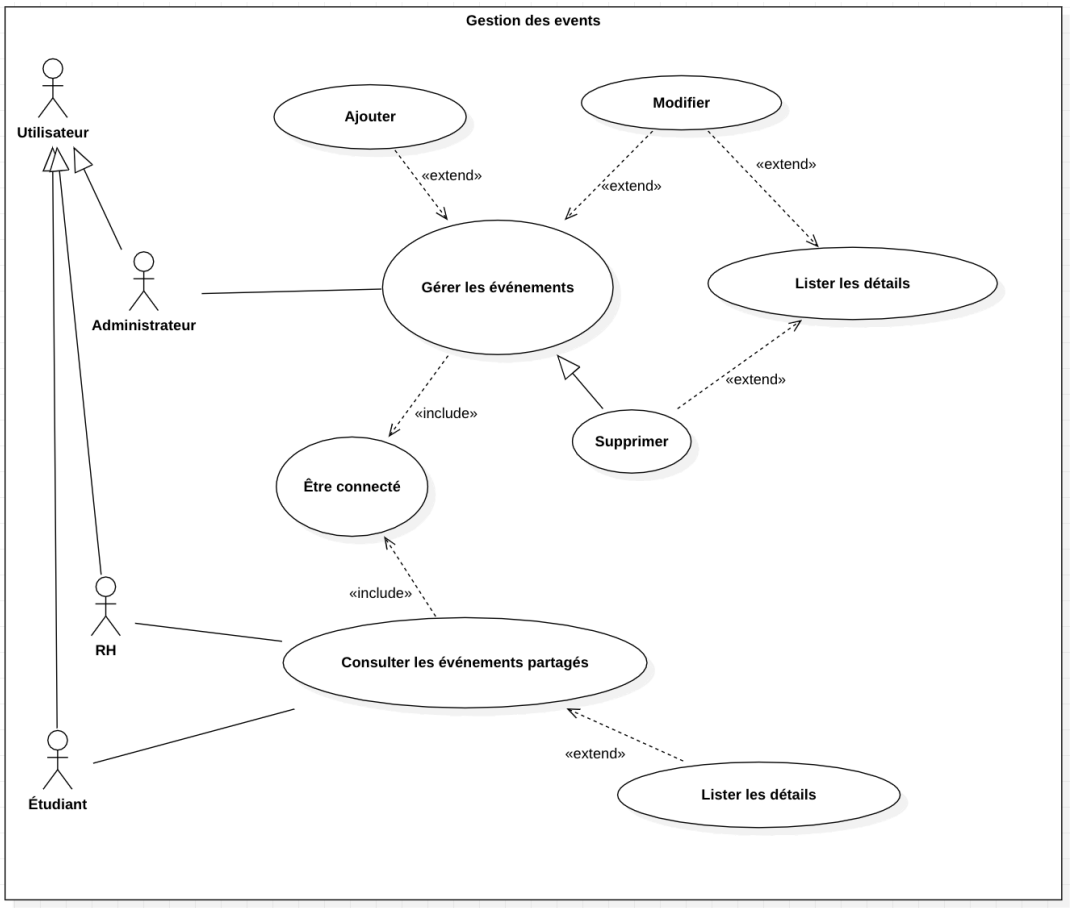

FIGURE 2.4 – diagramme de cas d'utilisation "Gestion des événements"

diagramme de cas d'utilisation module : Gestion des Documents

Fonctionnalités :

Ajout/téléchargement/suppression de documents (étudiants).
Consultation des documents partagés.

Relations :

« Extend » pour des extensions (ex : suppression après consultation).

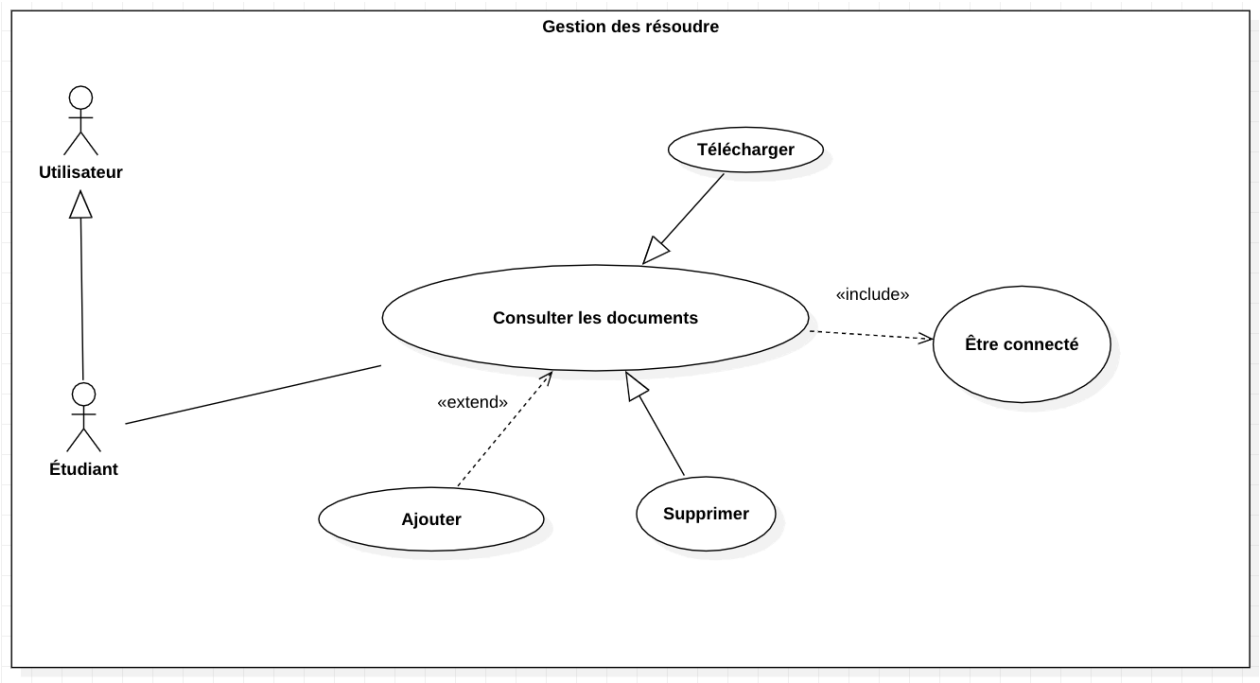

FIGURE 2.5 – diagramme de cas d'utilisation "Gestion des documents"

diagramme de cas d'utilisation module : Gestion des Statistiques

Rôle principal : Administrateur.

Fonctionnalités :

Consultation des tableaux de bord (statistiques sur les inscriptions, offres, etc.).

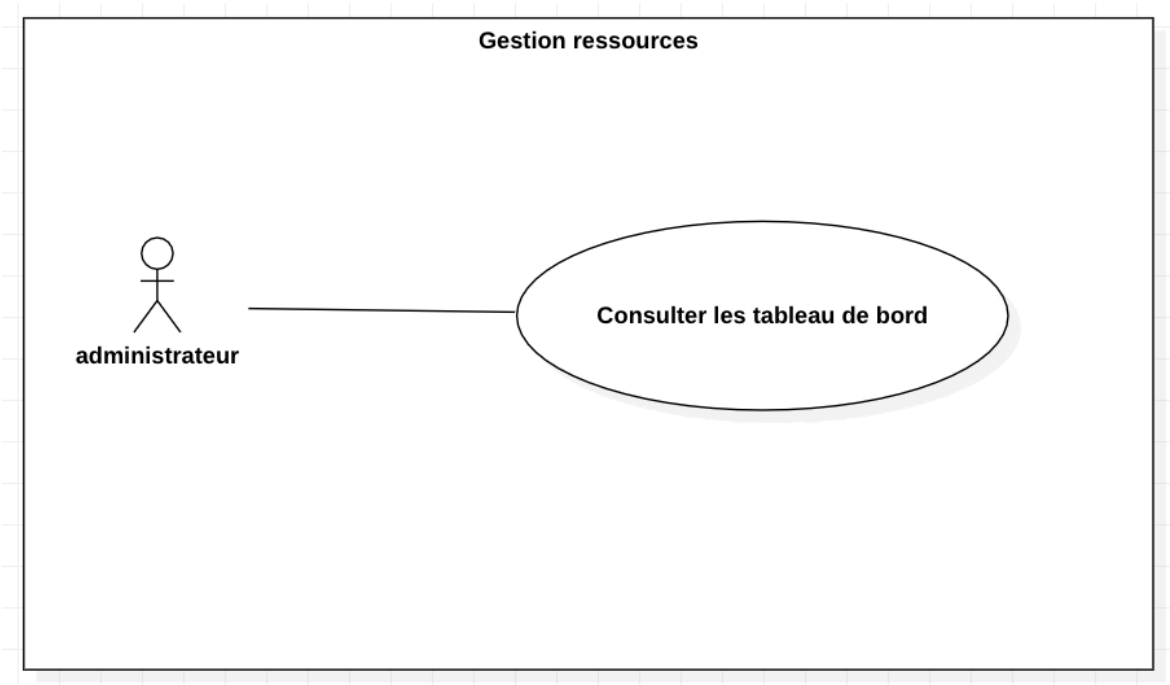

FIGURE 2.6 – diagramme de cas d'utilisation "Gestion des Statistiques"

2.3 Spécification Fonctionnelle Détaillée

2.3.1 Diagramme de classe

Le schéma de base de données de l'application de gestion d'association étudiante comprend quatre tables principales interconnectées. La table "users" sert de socle central avec les informations des utilisateurs (étudiants, RH et administrateurs), stockant leurs identifiants (id, email, mot de passe chiffré), rôles (role), données personnelles (noms, adresse, téléphone) et paramètres de compte (token de confirmation, statut activé/désactivé). Les documents partagés dans le système sont gérés par la table "document" qui enregistre les métadonnées (nom, type) et le contenu binaire des fichiers, tous liés à leur propriétaire via user-id. Les opportunités professionnelles sont consignées dans "offres-emploi" avec les détails des offres (titre, description, société, type de poste) et leur lien de candidature, chaque offre étant associée à l'utilisateur qui l'a publiée. Enfin, la table "event" organise la gestion des événements en conservant leurs caractéristiques (titre, date, lieu, description) et visuels (image binaire), également rattachés à un organisateur. L'architecture relationnelle repose sur des clés étrangères user-id qui créent des liens cohérents entre ces tables,

permettant ainsi des requêtes efficaces pour des fonctionnalités comme la modération de contenu, le suivi des activités utilisateur ou la personnalisation des interfaces selon les rôles. Cette structure normalisée offre une base solide pour étendre le système avec des modules complémentaires comme la messagerie ou les statistiques avancées.

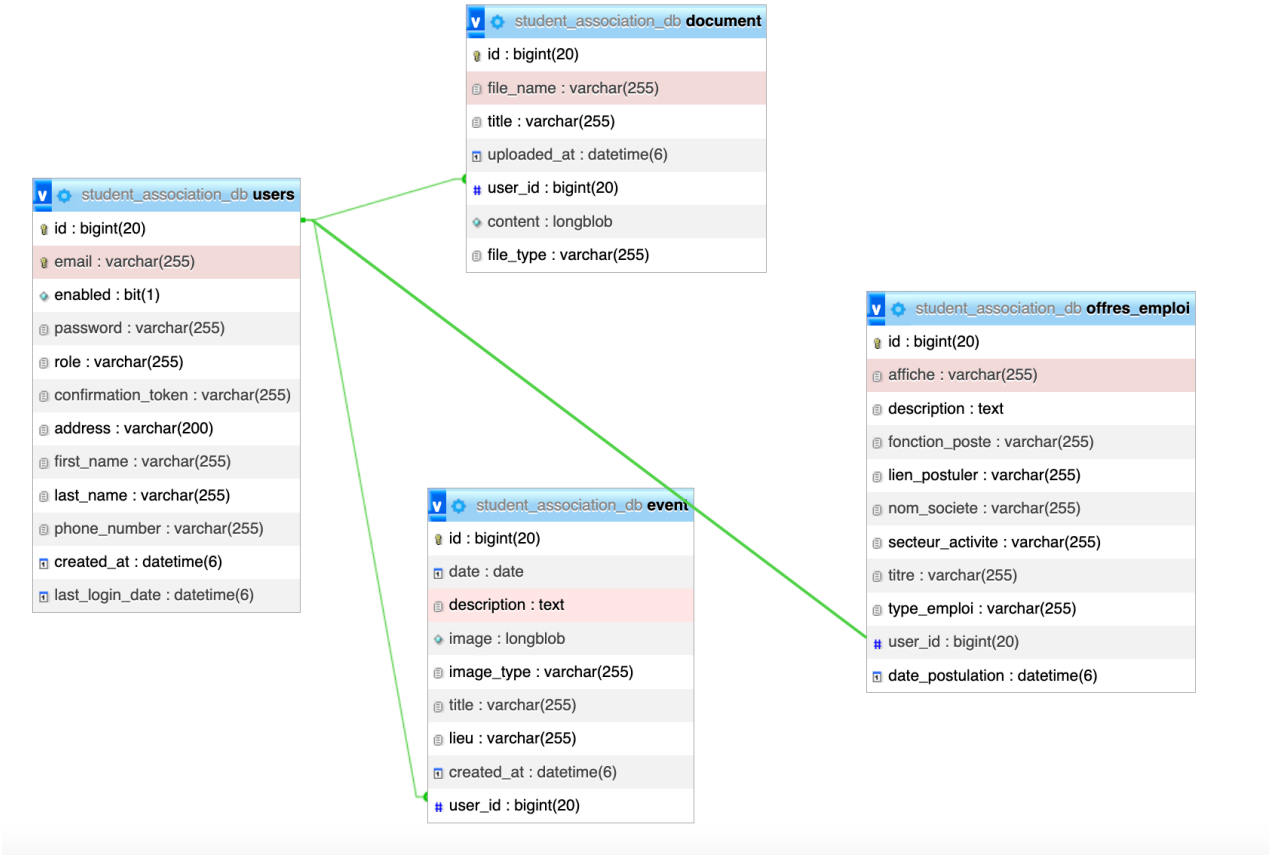

FIGURE 2.7 – diagramme de classe

2.3.2 Cas d'utilisation détaillés

Cas d'utilisation 1 : Inscription :

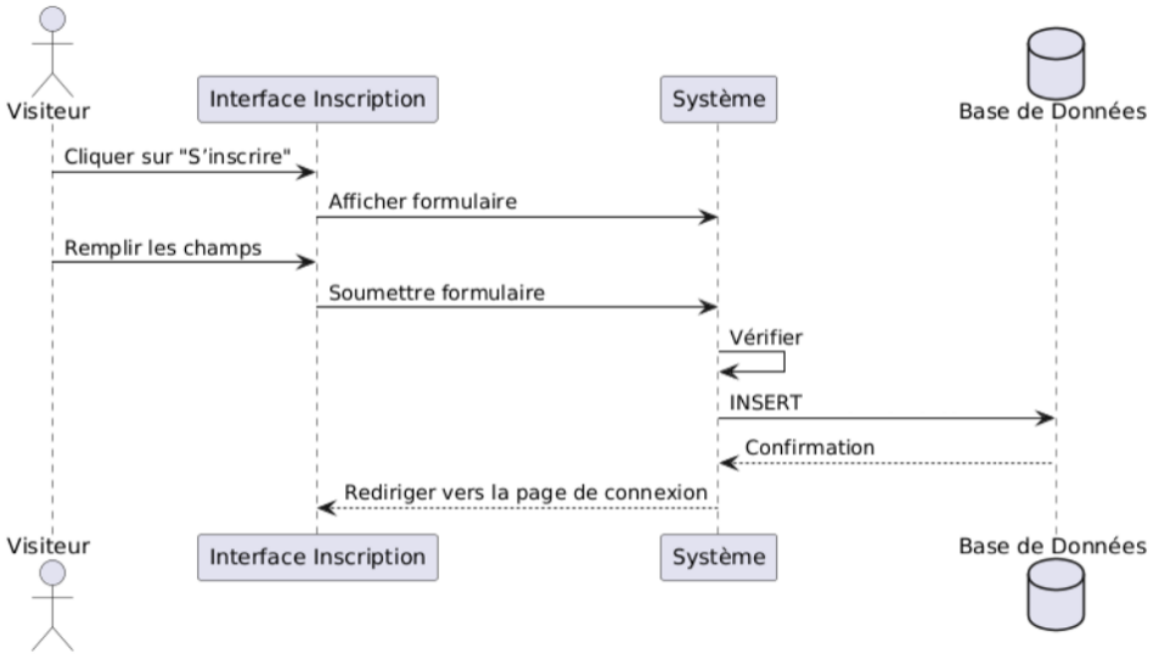

FIGURE 2.8 – diagramme de séquence cas 1

Acteur principal : Visiteur

Description : Permet à un visiteur de créer un compte sur la plateforme.

Préconditions : Le visiteur n'est pas connecté.

Postconditions : Le visiteur est redirigé vers la page de connexion après la création du compte.

Déroulement nominal :

1. Le visiteur clique sur 'S'inscrire'.
2. Le système affiche le formulaire d'inscription.
3. Le visiteur remplit les champs requis.
4. Le système valide les informations.
5. Le compte est créé et e-mail de bienvenue est envoyé .
6. L'utilisateur est redirigé vers la page de connexion.

Cas d'utilisation 2 : Modifier son profil :

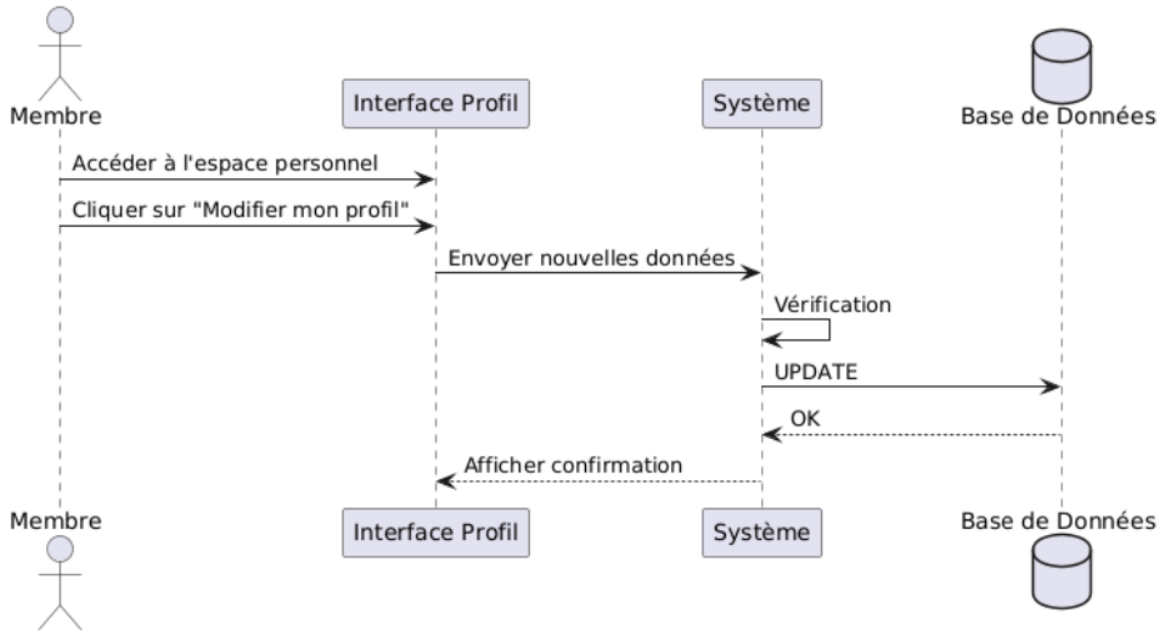

FIGURE 2.9 – diagramme de séquence cas 2

Acteur principal : tout les utilisateurs

Description : Permet au utilisateur de mettre à jour ses informations personnelles.

Préconditions : Le membre est connecté.

Postconditions : Les informations du profil sont mises à jour dans la base de données.

Déroulement nominal :

1. Le membre accède à son espace personnel.
2. Il clique sur 'Modifier mon profil'.
3. Il modifie les champs souhaités.
4. Le système vérifie la validité des données.
5. Les modifications sont enregistrées.

Cas d'utilisation 3 : Créer un événement :

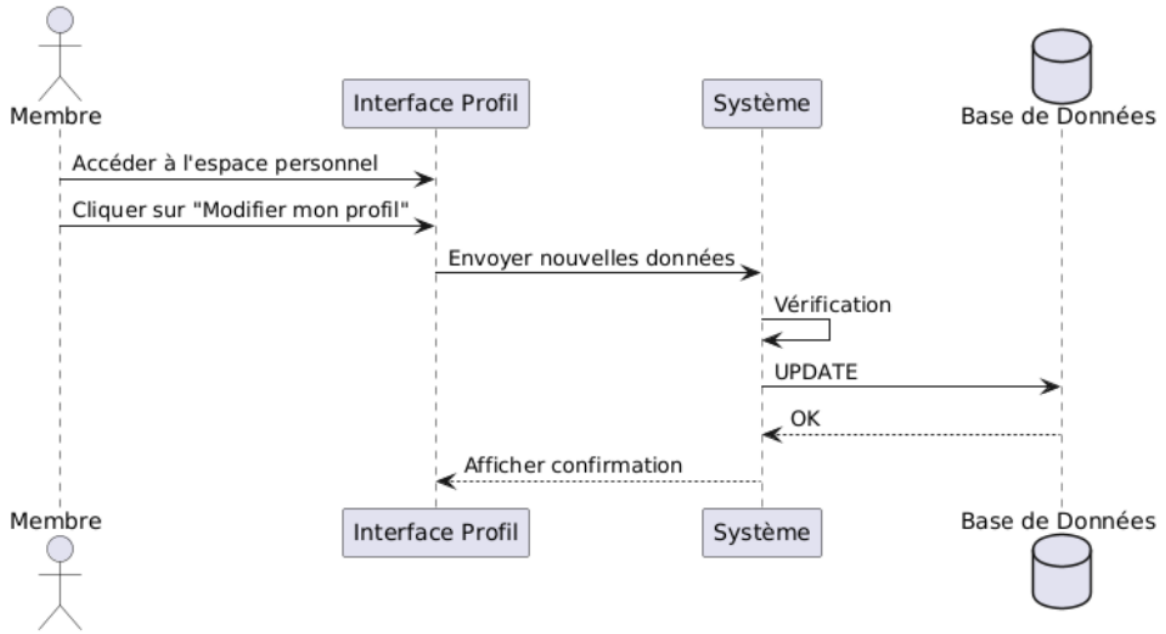

FIGURE 2.10 – diagramme de séquence cas 3

Acteur principal : Administrateur / RH

Description : Permet à un administrateur/RH de ajouter un événement.

Préconditions : L'administrateur/RH est connecté.

Postconditions : L'événement est enregistré dans le système.

Déroulement nominal :

1. L'administrateur/RH accède à l'espace 'Gestion des événements'.
2. Il clique sur 'Créer un événement'.
3. Il saisit les informations .
4. Le système vérifie que tous les champs obligatoires sont remplis.
5. L'événement est enregistré dans la base de données.

2.4 Conclusion

Nous avons structuré notre application autour de compétences architecturales avec une distinction nette, et l'aide fut fournie par des diagrammes de conception pour donner le développement clair et consistant. A travers plusieurs schémas de cahier de charges, séquences de classe et autres diagrammes, nous avons pu tracer des fondations solides pour démarrer et réussir notre projet.

Chapitre 3

Etude pratique

3.1 Introduction

Idée d'amélioration pour faciliter la vérification des composants du système d'information, la création d'une application web dédiée à la vérification et à la gestion des patches de sécurité est une solution efficace. solution, afin de rendre les processus de sécurité automatisés et simplifiés. L'application il est il est possible de la développer en utilisant des technologies informatiques appropriées. Parmi les exemples figurent l'HTML et à CSS (cascading style sheets).

3.2 Outils

3.2.1 Html

FIGURE 3.1 – Html logo

Développé par le W3C (World Wide Web Consortium) et le WHATWG (Web Hypertext Application Technology Working Group), le format ou langage HTML est apparu dans les années 1990. Il a progressivement subi des modifications et propose depuis 2014 une version HTML5 plus aboutie.

L'HTML est ce qui permet à un créateur de sites Web de gérer la manière dont le contenu de ses pages Web va s'afficher sur un écran, via le navigateur. Il repose sur un système de balises permettant de titrer, sous-titrer, mettre en gras, etc., du texte et d'introduire des éléments interactifs comme des images, des liens, des vidéos... L'HTML est plus facilement compris des robots de crawl des moteurs de recherche que le language JavaScript, aussi utilisé pour rendre les pages plus interactives.

Structure de base d'un document HTML

Voici un exemple

```
<!DOCTYPE html>
    <html>

        <head>
            <title> Title here </title>
        </head>

        <body>
            Web page content goes here.
        </body>

    </html>
```

FIGURE 3.2 – Structure de base d'un document HTML

Le rôle du HTML
est donc crucial puisqu'il va être notre langage privilégié pour indiquer aux navigateurs ce quoi est constituée chaque page et ce qu'ils doivent afficher. Grâce au HTML, on va par exemple pourvoir indiquer que tel contenu est un texte qui n'est qu'un paragraphe, que tel autre contenu est un texte qui est un titre de niveau 1 dans notre page, que tel autre contenu est une liste, un lien, etc. En plus de cela, le HTML va également nous permettre d'insérer différents médias (images, vidéos, etc.) dans nos pages web en indiquant au navigateur » à cette place- là dans ma page, je veux que s'affiche cette image « . Notez que dans ce cas précis, pour que le navigateur affiche la bonne image, on va lui fournir l'adresse de l'image dans le code HTML.

3.2.2 CSS

FIGURE 3.3 – css logo

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML puisque leur options stylistiques d'un document sont quelque peu limitées. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

À quoi sert le CSS

Le CSS permet donc de définir l'esthétique d'un site web. Sans CSS, une page web ne serait qu'une succession d'éléments noirs sur fond blanc les uns à la suite des autres.

C'est avec du code CSS que l'on définit par exemple :

la police d'un texte et son aspect (couleur, taille, etc.)

les marges et le rembourrage (padding) entourant un élément

l'apparence d'un menu

la création d'éléments géométriques

la mise en place de grilles d'images

l'ajout de bordure

l'application de différents effets ou animations simples

le changement de présentation d'un site selon l'appareil utilisé par l'utilisateur (téléphone portable, ordinateur, tablette)

la manière dont on navigue dans un site internet

La structure d'une déclaration CSS

Vous pouvez utiliser une déclaration CSS pour déterminer quels éléments de votre document électronique doivent prendre quelles valeurs ou propriétés. Dans sa structure de base, la déclaration se compose d'un sélecteur et d'accolades. Dans ces parenthèses, vous énumérez les déclarations de propriétés, qui sont séparées par des points-virgules. Chaque directive se compose du nom de la propriété, de deux points et d'une valeur spécifique. Après la dernière déclaration de propriété et avant la parenthèse finale, il est possible de placer un autre point-virgule, mais il n'est pas obligatoire. La déclaration CSS dans le diagramme ci-dessous permet de s'assurer que le titre h1 est affiché dans la couleur bleue et la taille de police 12 :

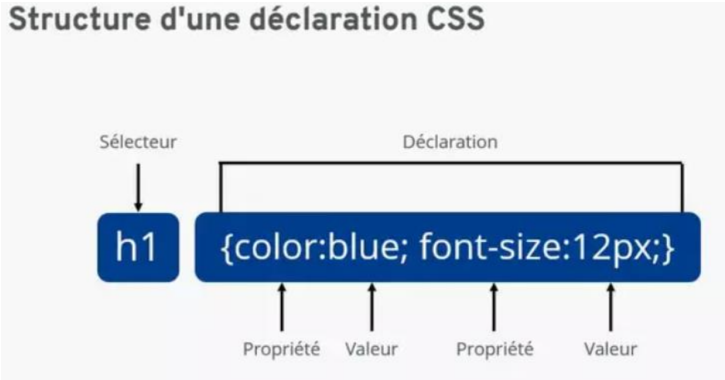

FIGURE 3.4 – Structure de CSS

3.2.3 bootstrap

FIGURE 3.5 – bootstrap logo

Bootstrap est un framework de développement front-end gratuit et open source pour la création de sites Web et d'applications Web. Conçu pour permettre le développement réactif de sites Web destinés aux mobiles, Bootstrap fournit une collection de syntaxes pour la conception de modèles.

En tant que framework, Bootstrap inclut les bases du développement Web réactif, les développeurs n'ont donc qu'à insérer le code dans un système de grille prédéfini. Le framework Bootstrap est construit sur Hypertext Markup Language (HTML), des feuilles de style en cascade (CSS) et JavaScript. Les développeurs Web utilisant Bootstrap peuvent créer des sites Web beaucoup plus rapidement sans perdre de temps à se soucier des commandes et fonctions de base.

3.2.4 js

FIGURE 3.6 – js logo

Caractéristiquesde JS

1. Langage de Script Côté Client : JavaScript est principalement employé côté navigateur pour manipuler le contenu HTML/CSS et interagir avec l'utilisateur, évitant ainsi de recharger la page.

2. Polyvalent et Dynamique : JavaScript permet l'utilisation de différents paradigmes de programmation, tels que la programmation orientée objet, fonctionnelle et impérative. Il convient parfaitement à la création de applications web complexes, jeux vidéo, applications mobiles et bien d'autres.

3. Syntaxe Simple : La syntaxe simple et expressive de JavaScript en fait un langage apprécié, accessible aux débutants mais proposant également des fonctionnalités avancées pour les développeurs expérimentés.

4. Modèle d'Objet Document (DOM) : En JavaScript, il est possible de manipuler le DOM qui représente les éléments HTML d'une page sous forme d'arborescence. Cela permet de faciliter la modification dynamique de l'interface utilisateur ainsi que la gestion des événements utilisateurs.

5. Large Écosystème : JavaScript est doté d'une large gamme de frameworks et de bibliothèques tels que React. js, Vue. js et AngularJS facilitent la création d'applications web modernes et réactives.

6. Exécution Rapide : JavaScript étant un langage interprété, il peut s'exécuter directement dans le navigateur. Cela offre aux utilisateurs des interactions instantanées et des mises à jour en temps réel, sans devoir recharger la page.

Utilisation de JS :

JavaScript est couramment utilisé pour améliorer l'interactivité des pages web en y ajoutant des fonctionnalités comme des animations, des formulaires dynamiques, des effets

visuels et même la création de jeux ou d'applications web en temps réel. Il est également possible de développer des applications mobiles hybrides avec JavaScript. JavaScript est un élément essentiel du développement front-end moderne, se combinant souvent avec HTML et CSS pour former une base solide.

3.2.5 staruml

FIGURE 3.7 – staruml logo

StarUML offre des fonctions de modélisation orientée objet basée, avec un support complet d'UML, conforme aux normes UML 2.x :

Diagrammes d'activité

Diagrammes de séquence

Diagrammes de communication

Diagrammes chronologiques

Diagrammes de transitions d'états

Diagrammes de flux d'informations

Diagrammes de vue d'ensemble d'interactions

Diagrammes de profils

Diagrammes de classes

Diagrammes de structure composite

Diagrammes de composants

Diagrammes d'objets

Diagrammes de paquetages

Diagrammes de cas d'utilisation

StarUML dispose également de capacités de modélisation en SysML :

Diagrammes de besoins

Diagrammes de blocs

Diagrammes internes de blocs

StarUML permet également de modéliser avec des représentations graphiques plus anciennes telles que :

Organigrammes

Diagrammes Entités-Associations (ERD)

Diagrammes de flux de données (DFD)

3.2.6 Écosystème Spring

L'infrastructure technique s'appuie sur l'écosystème Spring, un framework Java modulaire pour les applications d'entreprise. Les composants stratégiques intégrés incluent :

Spring Boot (conteneur léger avec auto-configuration)

Spring Data JPA (abstraction ORM pour la persistance des données)

Spring Security (gestion des authentifications)

Spring Cloud (orchestration des microservices)

Spring REST Docs (génération automatisée de documentation API)

Cette stack technique permet d'implémenter rapidement des fonctionnalités critiques tout en garantissant évolutivité et maintenabilité, avec des temps de développement réduits grâce à l'inversion de contrôle (IoC) et l'injection de dépendances.

FIGURE 3.8 – spring boot

3.2.7 Architecture spring en Couches

L'application adopte une architecture MVC (Modèle-Vue-Contrôleur) structurée en :

Couche Présentation : Contrôleurs REST traitant les requêtes HTTP

Couche Métier : Services implémentant la logique applicative

Couche Persistance : Repositories Spring Data pour l'accès aux données

Le flux typique suit le cycle :
Client → Controller → Service → Repository → Base de données

Points clés :
Dependency Injection pour un couplage faible
Spring Data JPA simplifiant les opérations CRUD
Validation centralisée dans les couches métiers

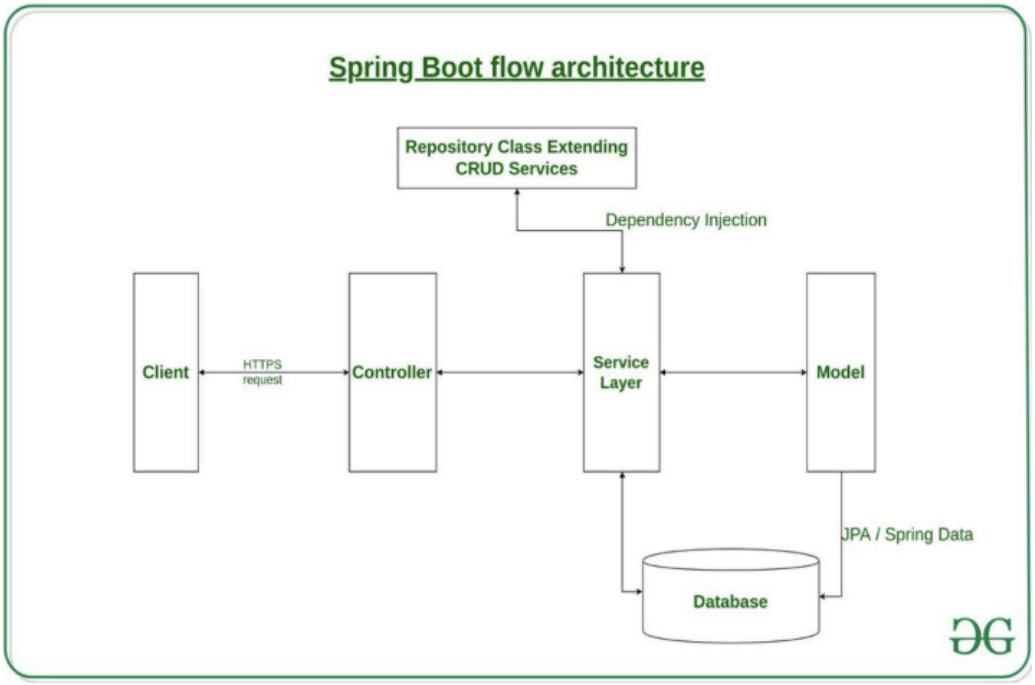

FIGURE 3.9 – architecture spring boot

3.2.8 Language Java

Java est un langage de programmation orienté objet, multiplateforme et sécurisé, fonctionnant sur la JVM (Java Virtual Machine).

Principales caractéristiques :

Syntaxe claire et typage statique

Gestion automatique de la mémoire (ramasse-miettes)

Richesse de l'écosystème (Spring, Hibernate, Maven)

Portabilité ("Write Once, Run Anywhere")

Cette combinaison Spring Boot + Java offre un équilibre optimal entre productivité (conventions over configuration) et performance, tout en restant adaptée aux évolutions futures (migration vers des microservices, intégration de Kafka...).

FIGURE 3.10 – java

3.2.9 MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) open-source.

Rôle dans le projet :

Stockage structuré des données (utilisateurs, offres, documents)

Requêtes SQL optimisées pour les opérations CRUD

Compatibilité avec Spring Data JPA pour l'abstraction ORM

FIGURE 3.11 – MySql

3.2.10 MAMP

MAMP (Macintosh, Apache, MySQL, PHP/Python/Perl) est un environnement de développement local.

Utilisation :

Hébergement du serveur MySQL/MariaDB en local

Test des API via Apache/PHP (bien que le projet utilise Java/Spring)

Alternative : XAMPP (multiplateforme)

Figure 3.12 – MAMP

3.2.11 git/github

Git : Système de contrôle de version décentralisé pour le suivi des modifications du code.

Commandes essentielles : commit, push, pull, branch

GitHub : Plateforme cloud pour l'hébergement de dépôts Git.

Fonctionnalités projet :

Collaboration (pull requests, issues)

CI/CD via GitHub Actions

Gestion des versions (tags, releases)

FIGURE 3.13 – git/github

3.2.12 Postman API

Postman est un outil de test et documentation d'API REST.

Fonctionnalités clés :

Envoi de requêtes HTTP (GET, POST, PUT, DELETE)

Validation des réponses JSON

Automatisation des tests (collections)

Génération de documentation interactive

FIGURE 3.14 – Postman API

3.2.13 intellij idea ultimate

IntelliJ IDEA Ultimate est l'environnement de développement intégré (IDE) principal utilisé pour ce projet, offrant des fonctionnalités avancées spécialement optimisées pour Java et Spring Boot. Cet outil permet un développement backend efficace grâce à sa prise en charge native de Spring (auto-complétion des configurations, reconnaissance des annotations), un débogage puissant, et une intégration transparente avec MySQL pour la gestion des bases de données. Il inclut également des outils intégrés pour Git/GitHub (gestion visuelle des branches) et Postman (tests d'API), tout en boostant la productivité via des fonctionnalités de refactoring intelligent et de génération de code. Son interface intuitive et ses performances optimisées pour les projets complexes en font un choix idéal pour ce type d'application d'entreprise, malgré son coût, justifié par les gains de temps et la réduction des erreurs lors du développement.

FIGURE 3.15 – intellij idea

3.2.14 Mailtrap.io

Mailtrap.io est un service de test d'e-mails qui permet aux développeurs de capturer et d'inspecter les e-mails sortants dans un environnement sandbox sécurisé, sans envoyer de vrais messages aux destinataires. Il sert principalement à :

Tester les fonctionnalités d'envoi d'e-mails (inscriptions, réinitialisations de mot de passe, notifications)

Éviter le spam lors du développement et des tests

Analyser le contenu des e-mails (HTML, texte, pièces jointes)

Simuler différents scénarios (bounces, délivrabilité)

Avantages clés :
Intégration facile avec SMTP ou API (compatible avec Spring Boot, Node.js, etc.)
Environnement isolé pour le staging et le développement
Historique des e-mails et outils de débogage

Utilisation typique :

Configurer Mailtrap comme serveur SMTP dans l'application.

Les e-mails sont redirigés vers un inbox virtuel Mailtrap.

Vérifier le rendu et les métadonnées avant le passage en production.

FIGURE 3.16 – Mailtrap

3.3 Réalisation du projet

La phase de réalisation s'est déroulée de manière itérative, en suivant la logique des sprints définis dans la planification. Chaque fonctionnalité a été développée, testée et intégrée progressivement, ce qui a permis une amélioration continue du produit. Nous avons utilisé Spring Boot pour la partie back-end et Thymeleaf pour le rendu des interfaces, tout en adoptant une architecture bien structurée respectant le principe MVC. L'ensemble du code source ainsi que l'historique de développement, les issues traitées, et les améliorations apportées sont consultables sur le dépôt GitHub suivant :

https://github.com/anasaeQ10/PFA-4IIR-project.git

Ce référentiel centralise toutes les contributions techniques, les captures d'écran de l'application, ainsi que les éventuelles tâches planifiées à travers les sprints.

L'application distingue trois profils d'utilisateurs, chacun disposant de droits et d'interfaces adaptés à son rôle :

Administrateur
Il supervise l'ensemble de la plateforme, gère les utilisateurs, valide les offres d'emploi et événements, visualise les statistiques globales, et administre les documents.

Étudiant
Il peut consulter les offres d'emploi/stage, s'inscrire aux événements, accéder aux documents, modifier son profil et interagir avec les ressources proposées.

Recruteur (RH)
Il a la possibilité de publier des offres d'emploi ou de stage, consulter les candidatures, et interagir avec les étudiants via le système.

3.3.1 Page d'accuei

Cette page d'accueil, intitulée "Trouvez votre emploi idéal", est une interface simple et conviviale conçue pour guider les utilisateurs vers des opportunités professionnelles. Elle met en avant le Career Center d'Honoris United Universities, dont la mission est de connecter les talents aux meilleures offres d'emploi.

La page propose deux actions principales :

Créer un compte : pour les nouveaux utilisateurs.

Se connecter : Option pour les utilisateurs déjà inscrits afin d'accéder à leur espace personnel, candidatures, alertes, etc.

Le design est épuré, avec un message clair et des call-to-action directs, ce qui facilite la navigation et l'engagement des utilisateurs. L'accent est mis sur l'accessibilité et la

simplicité, reflétant probablement une approche centrée sur les besoins des étudiants ou chercheurs d'emploi.

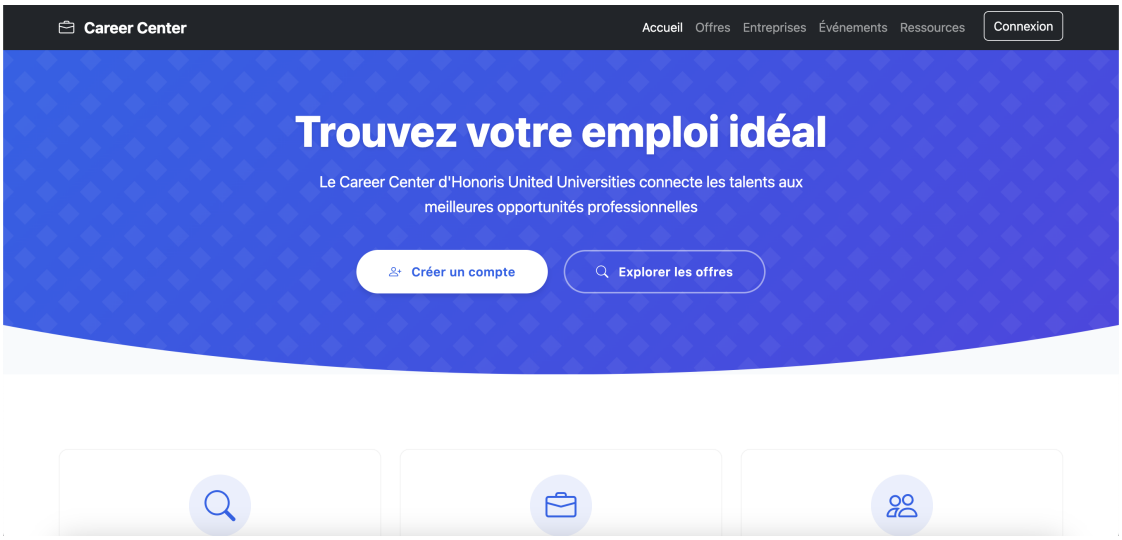

FIGURE 3.17 – page d'accueil

3.3.2 page de connexion

Objectif : Permettre aux utilisateurs existants d'accéder à leur compte.

Éléments clés :

Titre : "Connection" (erreur : "Connexion" en français).
Instruction : "Entrez vos identifiants pour accéder à votre compte".

Champs du formulaire :

Adresse email (vide).

Mot de passe (pré-sélectionné par défaut, ce qui est inhabituel).

Case à cocher "Se souvenir de moi".

Lien "Mot de passe oublié ?".

Bouton : "[+] [ ] Se connecter" (design non standard).

Option alternative : "Pas encore de compte ? S'inscrire".

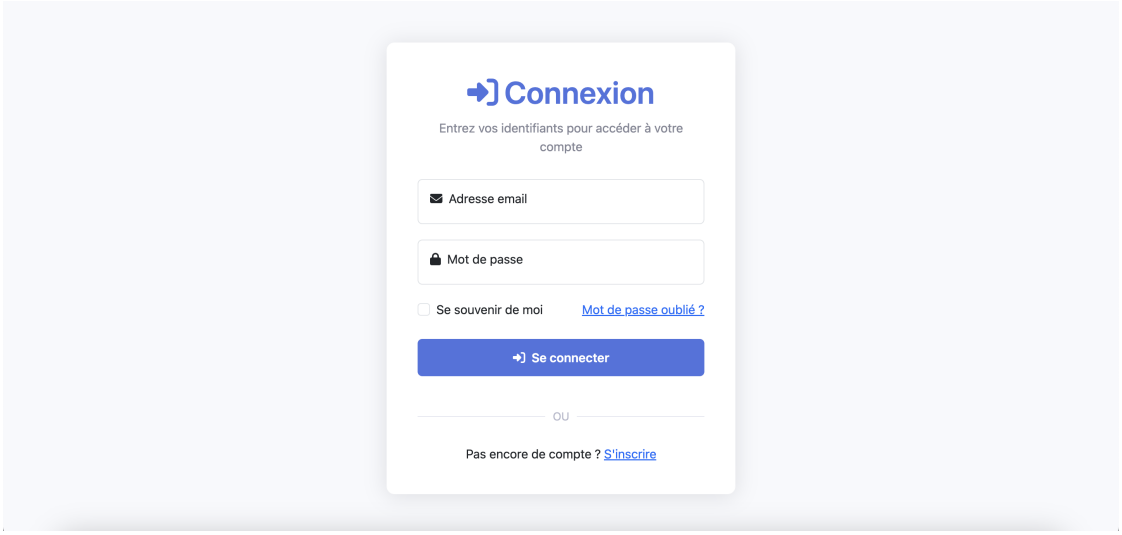

FIGURE 3.18 – page de connexion

3.3.3 Page d'Inscription

Objectif : Permettre aux utilisateurs de créer un compte pour accéder aux services du Career Center.

Éléments clés :

Titre : "Créer un compte" (orthographe correcte attendue : "Créer un compte").

Instruction : "Remplissez le formulaire pour vous inscrire" (erreur de frappe : "Bemplissez").

Champs du formulaire :

Prénom et Nom (champs séparés).

Email et Mot de passe (avec une exigence de sécurité indiquée : "Doit contenir au moins 8 caractères, une majuscule, une minuscule, un chiffre et un caractère spécial").

Confirmation du mot de passe.

Téléphone et Adresse (champs supplémentaires pour les informations utilisateur).

Bouton : "S'inscrire" (soumission du formulaire).

Lien alternatif : "Déjà un compte ? Se connecter" (erreur de formulation : "Se connectez").

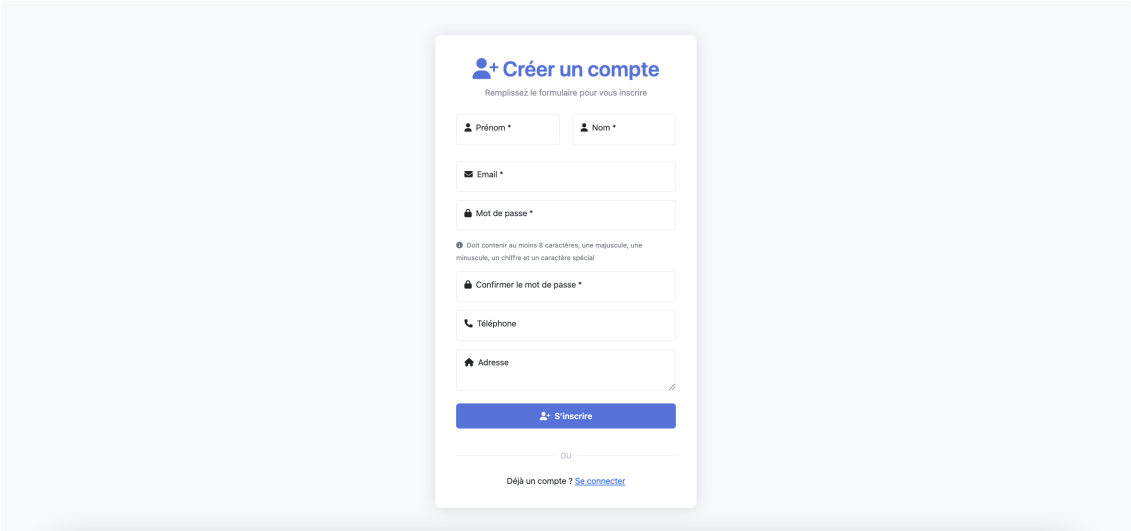

FIGURE 3.19 – Page d'Inscription

3.3.4 Page des utilisateurs (pour admin)

Objectif : Permettre à l'administrateur de gérer les comptes utilisateurs (visualisation, recherche, modifications).

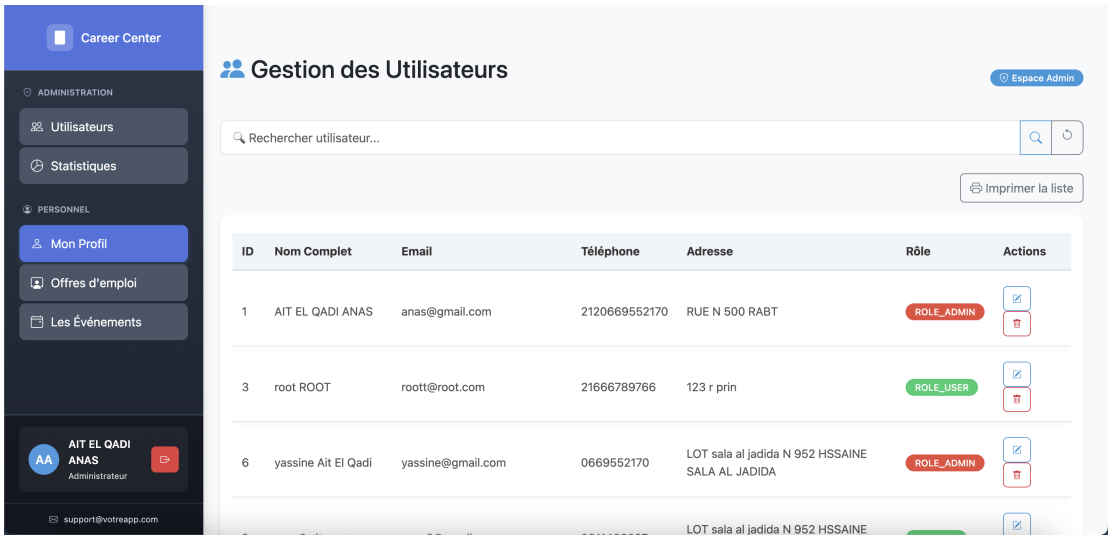

FIGURE 3.20 – Page des utilisateurs

Contenu :

Barre de recherche : Champ "Rechercher utilisateur..." pour filtrer les résultats.

Tableau des utilisateurs avec colonnes :

ID (numéro de ligne).

Nom Complet (ex. "AIT EL QADI ANAS").

Email (ex. anas@gmail.com).

Téléphone (formats variés, ex. *2120669552170*).

Adresse (ex. "RUEN 500 RABT").

Rôle (ROLE ADMIN ou ROLE USER ou ROLE RH).

Actions (manquantes dans l'image, probablement boutons Éditer/Supprimer).

En plus de la possibilité de imprimer des rapports (la liste des utilisateurs).

Liste des Utilisateurs

Imprimé le : samedi 5 juillet 2025 à 19:09:36
Par : AIT EL QADI ANAS

ID	Nom Complet	Email	Téléphone	Adresse	Rôle
1	AIT EL QADI ANAS	anas@gmail.com	2120669552170	RUE N 500 RABT	ROLE_ADMIN
3	root ROOT	root@root.com	21666789766	123 r prin	ROLE_USER
6	yassine Ait El Qadi	yassine@gmail.com	0669552170	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	ROLE_ADMIN
9	anas2 ait	anas2@gmail.com	0611436307	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	ROLE_RH
19	A anas	anasbenrajaaa@gmail.com	0611436307	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	ROLE_USER
20	Anas Ait El Qadi	anasbenrajaa@gmail.com	0669552170	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	ROLE_RH
21	AIT EL QADI Ait El Qadi	Aitelqadi.Anas@emsi-edu.ma	0669552170	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	ROLE_RH
23	Anas Ait El Qadi	Aitelqadi.Anas23@emsi-edu.ma	0669552170	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	ROLE_USER
24	IZMAR MOHAMED TAHA	taha@gmail.com	212639274831	LOT idi alal el bahrawi N 1021	ROLE_USER
27	hamid benani	hamid.benani@gmail.com			ROLE_USER
28	test mail	anasbenraja@gmail.com			ROLE_USER
29	larbi lhadi	larbi.lhadi32@gmail.com			ROLE_USER
30	omar ait el	omer@omar.com	034565643	rabar	ROLE_USER

FIGURE 3.21 – rapport des utilisateurs

3.3.5 Page "Tableau de Bord"

Objectif : Fournir une vision synthétique et visuelle des données clés du système.

Contenu analysé :

Métriques Principales (KPI) :

13 UTILISATEURS | 8 OFFRES D'EMPLOI | 0 NOUVELLES OFFRES (7J) → Indicateurs clairs mais "0 nouvelles offres" pourrait être mis en évidence (couleur, icône).

Graphique des Offres d'Emploi :
Attendu : Histogramme l'évolution des offres sur 30 jours.

Graphique de la répartition des utilisateurs par rôle :
Attendu : graphique de type (chart js doughnut) montre la répartition des utilisateurs

par leurs rôles (ADMIN , étudiant , RH)

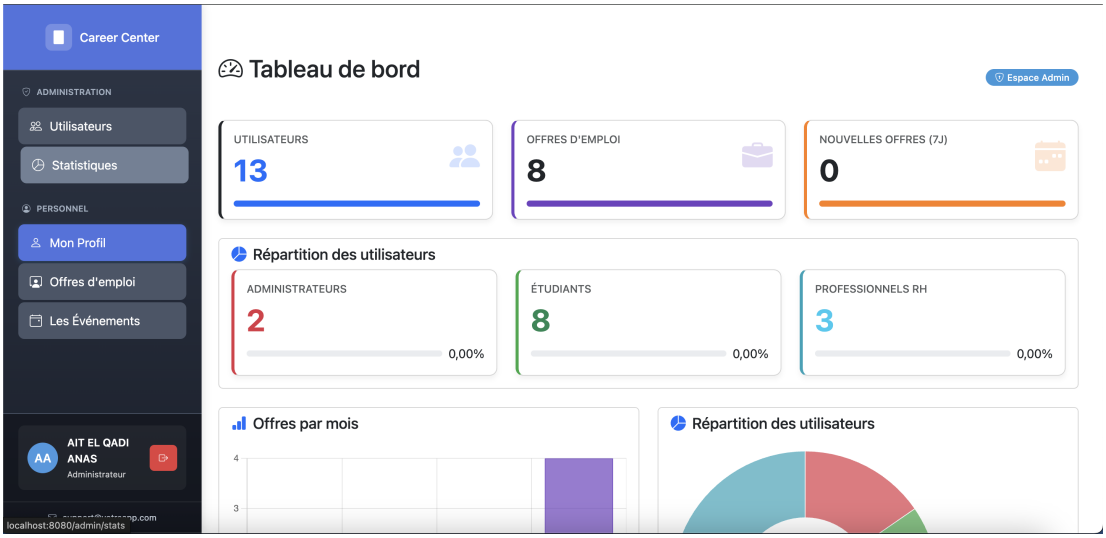

FIGURE 3.22 – Tableau de Bord

3.3.6 page de profil

Objectif : Afficher et permettre la gestion des informations personnelles de l'utilisateur.

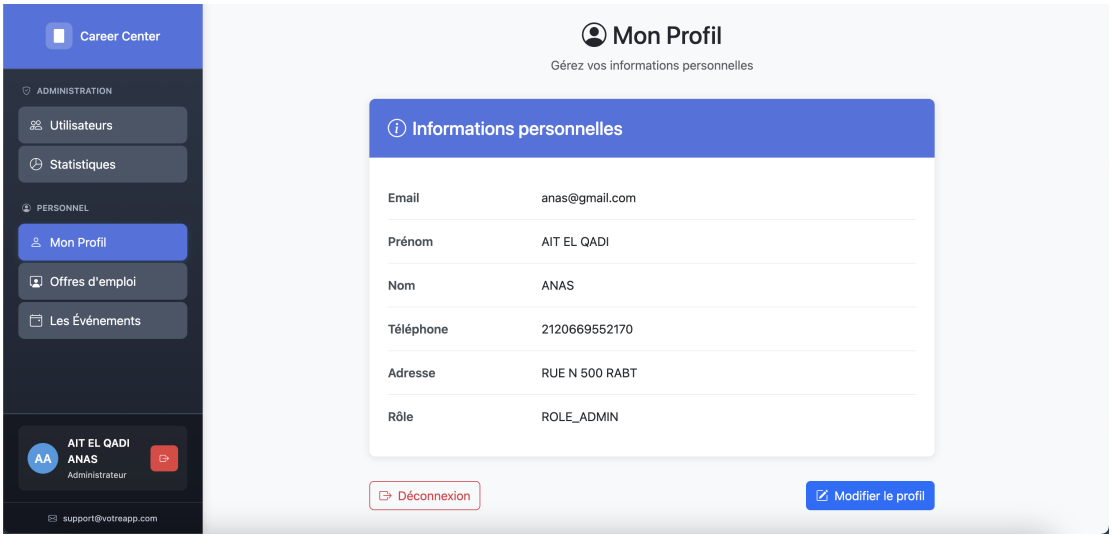

FIGURE 3.23 – page de profil

Contenu :

Section "Informations personnelles" (tableau structuré) :

Email : anas@gmail.com

Prénom/Nom : AIT EL QADI ANAS (format en majuscules)

Téléphone : 2120669552170 (format international non standardisé)

Adresse : RUE N 500 RABT (erreur probable : "RUE" au lieu de "RUEN")

Rôle : ROLE ADMIN

Actions disponibles :

Boutons [Déconnexion] et [Modifier le profil].

3.3.7 Page de modification du Profil

Objectif : Permettre à l'utilisateur de mettre à jour ses informations.

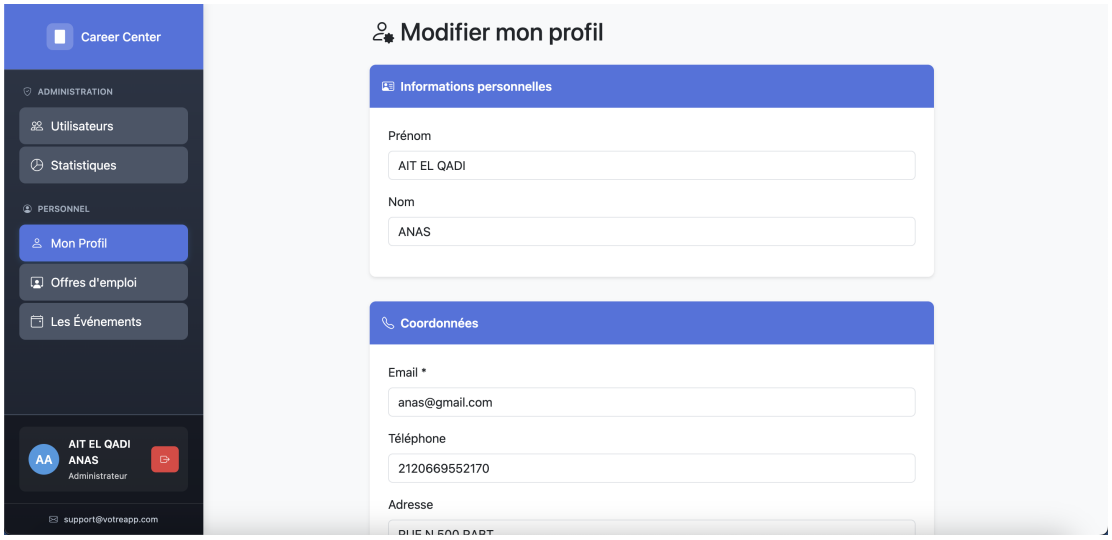

FIGURE 3.24 – Page de modification du Profil

Contenu :

Section "Informations personnelles" :

Champs Prénom (AIT EL QADI) et Nom (ANAS) séparés.

Section "Coordonnées" :

Email (non modifiable, marqué par une astérisque *).

Téléphone : 2120669552170 (même format problématique).

Adresse : "DIE N ERA BART" (incohérente avec la page profil).

Mot de passe : avec une exigence de sécurité indiquée : "Doit contenir au moins 8 caractères, une majuscule, une minuscule, un chiffre et un caractère spécial, et il doit être différent a l'ancien mot de passe .

3.3.8 Page des Offres

Objectif : Afficher et gérer les offres d'emploi publiées.

Contenu :

Menu latéral (incohérent avec le contexte) :
Options comme "Administration", "Statistiques" (non liées aux offres).

Tableau des offres :
Colonnes : Titre, Société, Type, Secteur, Posté par, Actions.
Exemples d'offres :

"offre test " (Microsoft, stage PFE, IT).

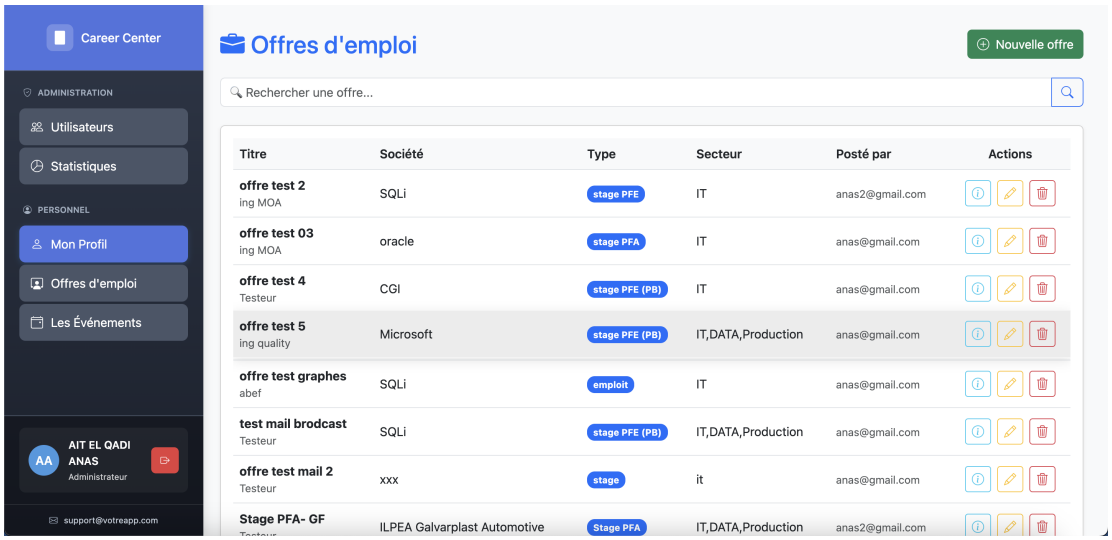

FIGURE 3.25 – Page des Offres

3.3.9 Page du Détails de l'Offre

Objectif : Afficher les détails d'une offre spécifique.

Contenu :

Titre : "Offre test 5".

Description : Texte dense sans mise en forme (ex. "DESCRIPTIF DU POSTE...").

Informations clés :

Type d'emploi : "stage PFA/PFE...".

Secteur : "IT/DATA/Production".

Section "Société" :

Exemple : Nom (Microsoft).

Bouton : "Postuler maintenant".

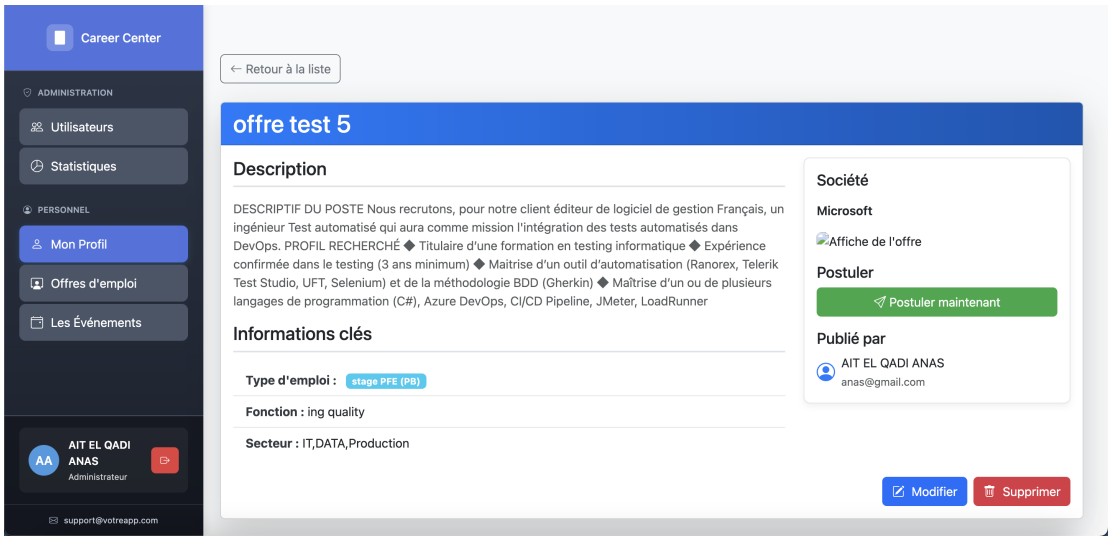

FIGURE 3.26 – Page du Détails de l'Offre

3.3.10 Page d'ajoute d'une Offre

Objectif : Permettre aux admins/recruteurs de publier une nouvelle offre.

Contenu :

Champs obligatoires (*) :

Titre, Société, Secteur, Type d'emploi, Lien pour postuler, Description.

Champ optionnel :

URL de l'affiche.

FIGURE 3.27 – Page d'ajoute d'une Offre

3.3.11 Page des des Événements

Objectif : Afficher les événements à venir (visites d'entreprises, forums, ateliers).

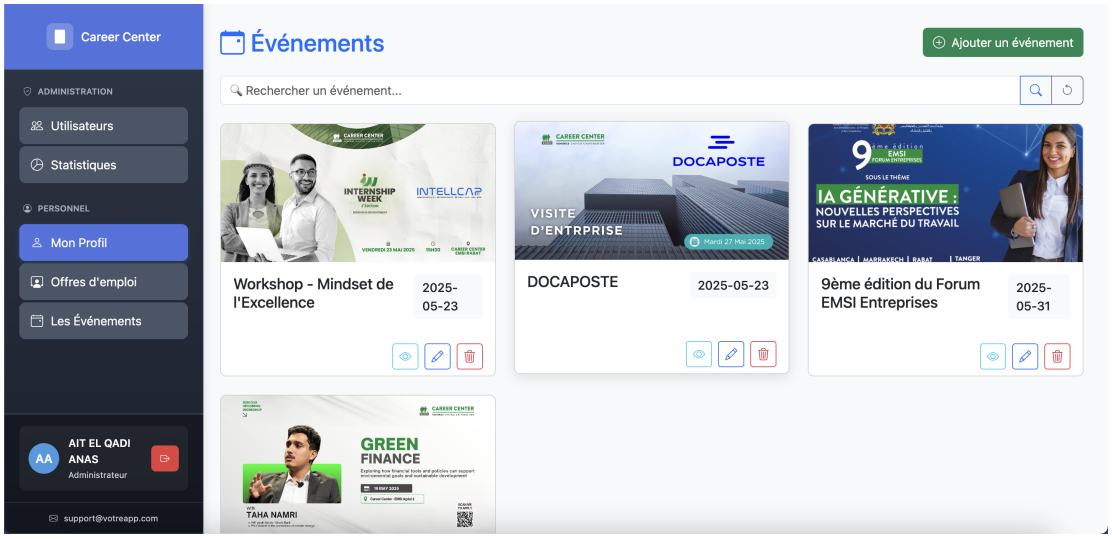

FIGURE 3.28 – Page des des Événements

Contenu :

Barre de recherche : "Redhercher un événement..." (erreur de frappe).

Exemples d'événements :

"VISITE D'ENTREPRISE – DOCAPOSTE" (dates : 23-31/05/2025).

"Forum EMSI Entreprises" (9ème édition, 31/05/2025).

"Workshop - Mindset de l'Excellence" (23/05/2025).

3.3.12 Page des Détails de l'Événement

Objectif : Afficher les détails complets d'un événement.

Contenu :

Menu latéral : Incohérent.

Titre : "Workshop - Mindset de l'Excellence".

Description : Texte mal formaté (ex. "visite de l'entreprise DOCAPOSTE...").

Sections parasites :

"INTERNSHIP WEEK" et "INTELLCAR" .

Informations pratiques :

Date : 23/05/2025 .

Lieu : "RABAT Agdal , emsi 1" .

FIGURE 3.29 – Page des Détails de l'Événement

3.3.13 Page d'ajout d'un Événement

Objectif : Permettre aux admins de créer un événement.

Contenu :

Champs minimaux :

Titre, Description, Date (format dd/mm/yyyy), Lieu.

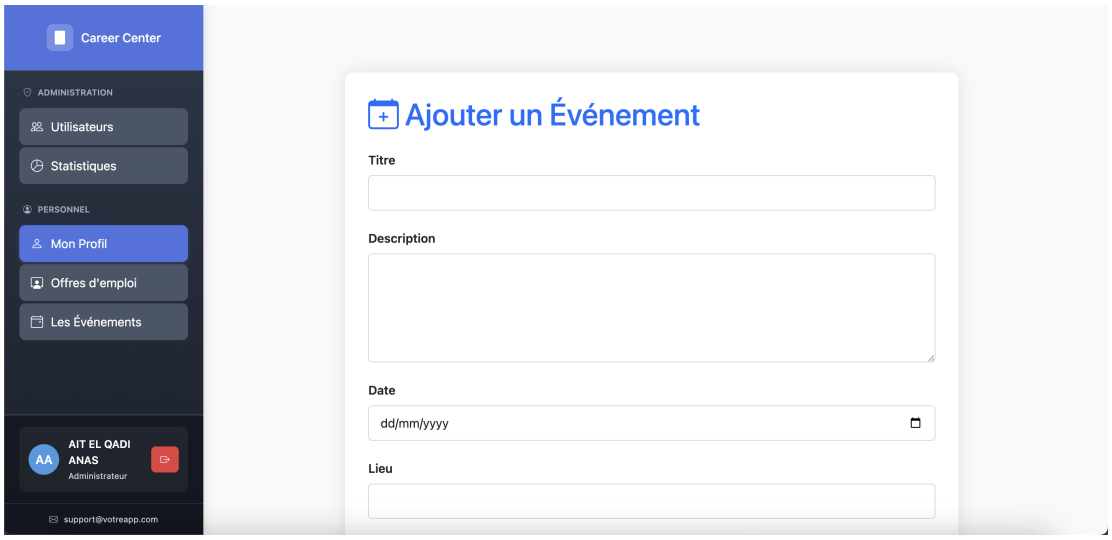

FIGURE 3.30 – Page d'ajout d'un Événement

3.3.14 Page des documents

Objectif : Permettre aux utilisateurs de gérer leurs fichiers uploadés (CV, contrats, etc.).

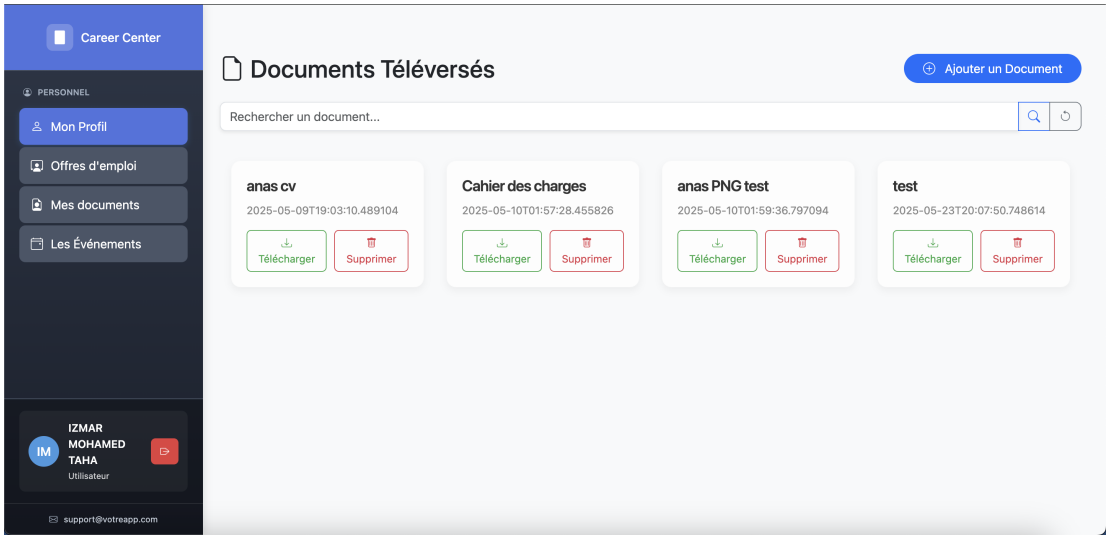

FIGURE 3.31 – Page des documents

Contenu :

Barre de recherche : "Rechercher un document..." (fonctionnelle).

Liste des documents :

Exemples : "anas cv", "Cahier des charges", "test".

Métadonnées : Date/heure de téléversement (ex. *2025-05-09T19 :03 :10.489104*).

Actions : Boutons Télécharger et Supprimer.

3.3.15 Page d'ajout d'un document

Objectif : Téléverser un nouveau fichier.

Contenu :

Champs :

Titre du Document (texte libre).

Choisir un Fichier (importer le document).

Bouton : [Ajouter le Document].

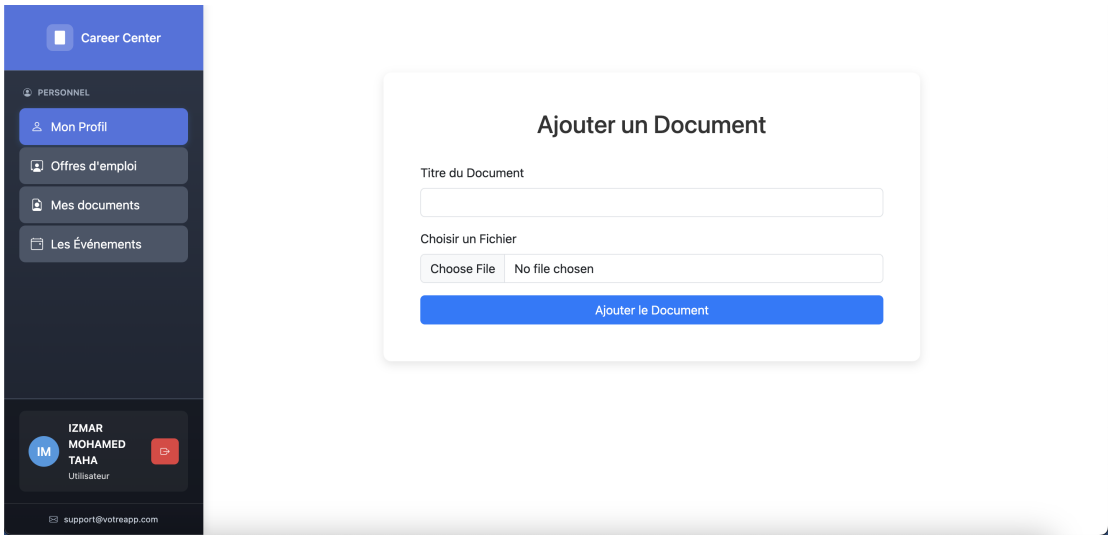

FIGURE 3.32 – Page d'ajout d'un document

3.3.16 les "e-mail" automatiquement envoyé

Mail de confirmation d'inscription

Ce courriel est automatiquement envoyé à l'utilisateur lors de son inscription sur la plateforme. Il a pour objectif de confirmer la création de son compte.

Le design est simple et professionnel, avec les éléments suivants :

Objet du mail : "Bienvenue sur notre plateforme"

Contenu principal : un message de bienvenue qui félicite l'utilisateur pour son inscription.

Call-to-action (CTA) : un bouton cliquable intitulé "Activer mon compte" ou similaire, qui redirige vers une page de vérification du compte.

Design : couleurs cohérentes avec l'identité visuelle de l'application, typographie lisible, et structure bien hiérarchisée.

Ce type de mail est essentiel pour assurer la sécurité et la validation des utilisateurs.

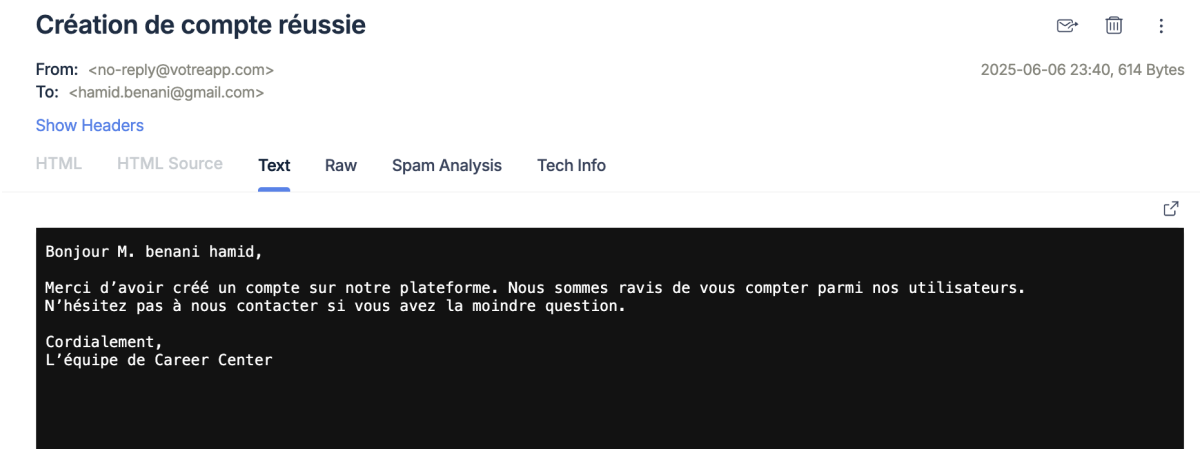

FIGURE 3.33 – Mail de confirmation d'inscription

Mail de notification d'un événement

Ce courriel est utilisé pour informer un utilisateur qu'un nouvel événement a été ajouté ou qu'il a été invité à participer à un événement.

Objet du mail : généralement du type "Nouvel événement disponible" ou "Invitation à un événement"

Contenu :

Titre de l'événement

Description brève

Date et lieu

Lien vers la plateforme pour consulter les détails ou confirmer sa participation.

Design : structuré de manière à faire ressortir les informations clés (titre, date, bouton d'action). Le bouton "Voir l'événement" permet à l'utilisateur de visualiser les détails rapidement.

Ce type d'email améliore l'engagement utilisateur et assure une bonne communication autour des activités proposées.

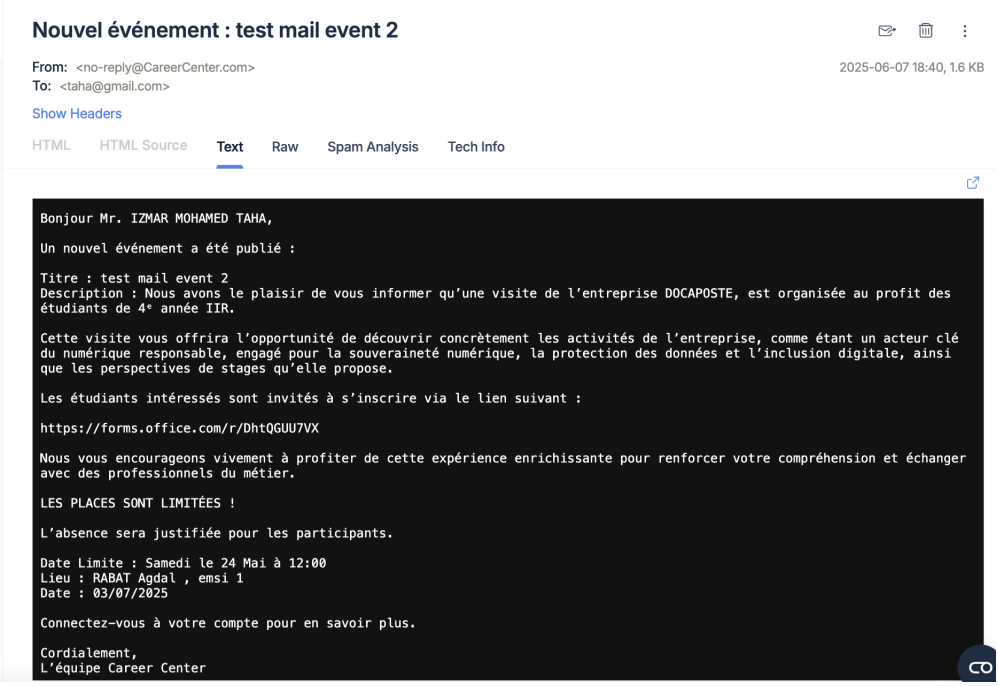

FIGURE 3.34 – Mail de notification d'un événement

Mail d'annonce d'une offre

Ce mail informe l'utilisateur qu'une nouvelle offre (ou annonce) a été publiée sur la plateforme.

Objet du mail : du type "Nouvelle offre disponible sur la plateforme"

Contenu :

Titre de l'offre ou de l'annonce

Description brève

Informations essentielles : lieu, date, type d'offre

Lien pour consulter les détails de l'offre

Design : attractif et clair, avec un bouton d'action "Voir l'offre" ou "Consulter maintenant", permettant à l'utilisateur de naviguer facilement.

Ce type de notification assure que les utilisateurs sont informés en temps réel des nouvelles opportunités.

FIGURE 3.35 – Mail d'annonce d'une offre

3.4 Conclusion

Pour conclure cette section consacrée à l'étude pratique de notre projet, nous avons minutieusement analysé les outils de développement utilisés et la séquence des interfaces. Nos éléments sont essentiels afin de garantir la solidité technique et l'ergonomie de notre application . Le choix méticuleux des outils et la planification minutieuse des interfaces représentent deux étapes cruciales pour réaliser notre vision.

Conclusion générale

En conclusion, notre application web dédiée à la gestion des offres de stage et des candidatures représente une avancée significative dans le domaine de l'accompagnement des étudiants vers l'insertion professionnelle. Nous avons conçu une plateforme intuitive et sécurisée, permettant aux étudiants de consulter et postuler à des offres de stage, et aux recruteurs de publier et gérer efficacement leurs annonces.

Grâce à l'intégration de technologies modernes telles que Spring Boot et Thymeleaf, et à une approche centrée sur l'expérience utilisateur, nous avons pu développer des fonctionnalités essentielles : la gestion des profils utilisateurs (étudiants et recruteurs), la publication d'offres avec des informations complètes, la limitation à une candidature par offre, ainsi que la notification automatique par e-mail lors de la création de compte ou de publication d'une nouvelle offre.

Ce projet nous a permis de relever divers défis techniques et organisationnels, enrichissant notre maîtrise du développement web full stack, de l'architecture logicielle, ainsi que de la gestion de projet. Nous avons également porté une attention particulière à l'ergonomie de l'interface pour assurer une navigation fluide et agréable.

Nous sommes convaincus que cette application apportera une réelle valeur ajoutée tant pour les établissements d'enseignement que pour les entreprises en facilitant la mise en relation entre étudiants et recruteurs. Les acquis techniques et humains de ce projet constituent une base solide pour de futures évolutions, afin d'adapter la solution aux besoins croissants du marché et aux retours des utilisateurs.

Perspectives

Dans l'optique de renforcer l'utilité de notre application de gestion des correctifs de sécurité et d'optimiser l'expérience utilisateur, plusieurs axes d'évolution ont été identifiés.

Tout d'abord, le développement d'une version mobile est prévu. Cette déclinaison, spécialement pensée pour les smartphones et tablettes, offrira une interface ergonomique et responsive, garantissant un accès simplifié et complet aux fonctionnalités de l'application, quel que soit l'endroit où se trouvent les utilisateurs.

Un autre chantier essentiel concerne le renforcement de la sécurité. Des améliorations techniques seront mises en place pour assurer une meilleure protection des données, un contrôle plus rigoureux des accès et une détection proactive des vulnérabilités.

Enfin, la prochaine étape stratégique consistera à développer intégralement la partie backend de l'application. Ce travail fera l'objet du projet de stage de l'année prochaine et visera à concevoir une architecture robuste, modulaire et scalable. L'objectif est d'assurer une meilleure gestion des données, une performance accrue, ainsi qu'une maintenance facilitée.

Nous sommes convaincus que ces évolutions contribueront à faire de notre application un outil encore plus fiable et performant, répondant pleinement aux exigences des professionnels de la sécurité informatique, tout en favorisant une meilleure anticipation et gestion des incidents techniques.

Webographie

1. Pour STARUML : https://staruml.io/user-guide

2. Pour springboot : https://docs.spring.io/spring-boot/index.html

3. Pour bootstrap : https://getbootstrap.com/docs/5.3/getting-started/introduction/

4. Pour l'éditeur intellij idea ultimate : https://www.jetbrains.com/idea/download

5. Pour w3schools (guide por HTML/CSS/JS) : https://www.w3schools.com/

Bibliographie

Pour la réalisation de ce projet, un ensemble de ressources documentaires fiables et reconnues ont été mobilisées afin de garantir une bonne maîtrise des technologies employées, notamment Java et le framework Spring Boot. Concernant Java, les ouvrages de référence comme *Effective Java* de Joshua Bloch (Addison-Wesley, 2018) ont permis de comprendre les meilleures pratiques de programmation orientée objet en Java, notamment en termes de conception, de gestion de la mémoire et d'utilisation des API standards. Le livre *Head First Java* de Kathy Sierra et Bert Bates (O'Reilly, 2008), très didactique, a été utile pour consolider les bases du langage, grâce à son approche pédagogique axée sur la compréhension visuelle. *Thinking in Java* de Bruce Eckel (Prentice Hall, 2006), plus approfondi, a apporté une vision complète de la philosophie Java et de la programmation orientée objet.

En ce qui concerne Spring Boot, l'ouvrage *Spring in Action* de Craig Walls (Manning, 2018) a servi de guide principal pour comprendre l'architecture modulaire de Spring, la configuration automatique, l'injection de dépendance, ainsi que les différents modules intégrés comme Spring Data JPA ou Spring Security. De plus, *Clean Code* de Robert C. Martin (Prentice Hall, 2008) a permis d'appliquer une discipline de code propre et maintenable tout au long du développement, ce qui est essentiel dans tout projet collaboratif ou de longue durée.

Les documentations officielles ont également été une source essentielle d'information. La documentation Java SE d'Oracle (`https://docs.oracle.com/javase/`) a apporté des détails précis sur les classes, interfaces et fonctionnalités du langage. Celle de Spring Boot (`https://docs.spring.io/spring-boot/docs/current/reference/html/`) a été d'une grande utilité pour comprendre les options de configuration, les annotations et les bonnes pratiques de développement avec ce framework.

Pour renforcer les connaissances pratiques, plusieurs sites web spécialisés ont été consultés. Le site Baeldung (`https://www.baeldung.com/`) propose de nombreux articles techniques et cas d'usage concrets autour de Spring, Spring Boot, JPA et la sécurité web. Le portail GeeksforGeeks (`https://www.geeksforgeeks.org/java/`) a complété la formation avec des explications simples et des exemples clairs, tandis qu'OpenClassrooms a proposé un tutoriel pas à pas pour créer une application web avec Spring Boot. Enfin, les conventions de codage Java d'Oracle (`https://www.oracle.com/java/technologies/javase/codeconventions-contents.html`) ont été suivies afin de produire un code cohérent, lisible et conforme aux standards industriels.