



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

# PROJET DE STAGE

## 2022/2023

### Conception et Réalisation d'un système De recrutement intelligent

## Ingénierie Informatique et Réseaux

Réalisé par :

Tirgani Badreddine

Encadré par :

Mohammed AMEKSA

Patrick Baukassa

# DEDICACE

*Je dédie cet événement marquant de ma vie  
À la mémoire de mon père disparu trop tôt.  
J'espère que, du monde qui est sien  
Maintenant, il apprécie cet humble geste  
Comme preuve de reconnaissance de la part  
D'un fils qui a toujours prié pour le salut de  
Son âme. Puisse Dieu, le tout puissant,  
L'avoir en sa sainte miséricorde !*

# REMERCIEMENTS

Au terme de ce stage, je tiens à exprimer mes respects mes connaissances et mes sincères remerciements aux personnels du Openmind Services, le directeur de la société Patrick BAUKASSA, qu'ont contribué à l'élaboration de ce rapport modeste. Ainsi ceux qui m'aient aidées durant cette période pour effectuer mon stage dans les meilleures conditions, et je tiens également à remercier vivement tous ceux qui m'ont fait preuve d'un grand esprit de collaboration et d'initiative,

Tous les personnels de l'équipe technique et qui m'apportent une aide précieuse par leurs suggestions et leurs conseils.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail trouvent ici l'expression de nos sincères grâtes.

# RESUME

Le présent document est le fruit de mon travail dans le cadre de projets de stage de quatrième année. Ce projet avait comme but la réalisation d'une application web de sourcing.

Afin de mener à bien notre mission, il s'est avéré indispensable d'entamer le travail par une étude fonctionnelle, permettant d'identifier la problématique, ainsi que les caractéristiques du système à développer.

Nous avons opté pour une étude conceptuelle afin de mieux définir le comportement de notre application, ceci à travers des diagrammes de cas d'utilisation, de classe et de séquence, ce qui a rendu la phase d'implémentation plus spontanée. Et finalement une phase de développement et réalisation.

Afin de mener notre mission à bon port, nous avons choisi les technologies Python, Laravel (Otwell, 2011) pour le côté back-end, et comme service nous avons utilisée nginx (Sysoev, 2019), ce qui concerne le SGBD (Système de gestion de base de données) nous avons utilisé MySQL pour une gestion fluide des données, pour la partie front-end nous avons utilisé React JS (Platforms, 2013) pour un affichage efficace des composants. En fin pour l'abrégement nous avons utilisé un serveur dédié avec un system d'exploitation Linux (Unix)

Nous vous présenterons donc tout au long de ce rapport, les étapes que nous avons suivies ainsi que les outils que nous avons utilisés pour réaliser notre projet.

**MOTS CLÉS :** Système De recrutement intelligent.

# ABSTRACT

This document is the fruit of my work within the framework of fourth year internship projects. This project aimed to create a sourcing web application.

To carry out our mission, it proved essential to start the work with a functional study, making it possible to identify the problem, as well as the characteristics of the system to be developed.

We opted for a conceptual study to better define the behavior of our application, through use case, class and sequence diagrams, which made the implementation phase more spontaneous. Finally a phase of development and realization.

In order to carry out our mission successfully, we chose Python, Laravel (Framework of PHP) technologies for the back-end side, and as a service, we used nginx, for the MBDS (système de gestion de base de données, 2008) we used MySQL (MySQL AB, 1995) for fluid management data, for the front-end part we used React JS (Platforms, 2013) for an efficient display of the components.

In the end, for hosting, we used a dedicated server with an ubuntu Linux (Unix)operating system

We will therefore present throughout this report, the steps we followed as well as the tools we used to carry out our project.

**KEYWORDS :** Intelligent recruitment system.

.

# Table des matières

<b>DEDICACE</b>	<b>1</b>
<b>REMERCIEMENTS</b>	<b>2</b>
<b>RESUME</b>	<b>3</b>
<b>ABSTRACT</b>	<b>4</b>
<b>TABLE DES MATIERES</b>	<b>V</b>
<b>LISTE DES FIGURES</b>	<b>7</b>
<b>INTRODUCTION GENERALE</b>	<b>1</b>
<b>CADRE GENERALE DU PROJET</b>	<b>1</b>
1. INTRODUCTION	3
2. PRESENTATION DE OPENMIND-SERVICE	3
3. PRESENTATION	3
4. ANALYSE DE L'EXISTANT :	3
5. PROBLEMATIQUE :	4
6. SOLUTIONS PROPOSEE :	5
7. PLANNING PREVISIONNEL	5
8. CYCLE DE VIE ET METHODOLOGIE UTILISEE	5
9. CYCLE DE VIE D'UN LOGICIEL :	6
10. BESOINS FONCTIONNELS	6
11. BESOINS NON FONCTIONNELS	8
12. PROCESSUS DE CALCUL DE LA CORRESPONDANCE ENTRE L'OFFRE ET LA DEMANDE	8
13. CONCLUSION	8
<b>ANALYSE ET CONCEPTION</b>	<b>9</b>
1. INTRODUCTION	10
2. PRESENTATION DU LANGAGE (UML)	10
3. MODELISATION	10
3.1. DIAGRAMME DE CAS D'UTILISATION	10
3.2. DIAGRAMME DE SEQUENCE	13
3.3. DIAGRAMME DE CLASSES	15
4. CONCLUSION :	16
<b>REALISATION DE L'APPLICATION</b>	<b>17</b>
1. INTRODUCTION	18

<b>2.</b>	<b>ENVIRONNEMENT DU TRAVAIL</b>	<b>18</b>
<b>3.</b>	<b>REALISATION</b>	<b>21</b>
<b>4.</b>	<b>CONCLUSION</b>	<b>31</b>

<b>CONCLUSION GENERALE :</b>	<b>32</b>
------------------------------	-----------

<b>REFERENCES</b>	<b>33</b>
-------------------	-----------

# Liste des figures

Figure 2.1 : Digramme de cas d'utilisation de Back Office.....	11
Figure 2. 2 : Diagramme de cas d'utilisation de l'Entreprise .....	12
Figure 2.3 : Diagramme de cas d'utilisation des Candidats.....	13
Figure 2.4 : Diagramme de séquence .....	14
Figure 2.5 : Digramme de classe.....	15
Figure 3.3.1– Page d'accueil sans authentification .....	21
Figure 3.3.1– Page de connexion .....	22
Figure 3.3.1– Page de registration .....	22
Figure 3.3.1– Page d'accueil.....	23
Figure 3.3.2- liste des utilisateurs .....	23
Figure 3.3.3 pour ajouter un utilisateur.....	24
Figure 3.3.4 listes des candidats.....	24
Figure 3.3.5 détail candidats .....	24
Figure 3.3.6 profils Candidats .....	25
Figure 3.3.7 _liste des offres de profil .....	25
Figure 3.3.8_ Ajouter Candidats .....	26
Figure 3.3.9 _ dictionnaire des tags .....	26
Figure 3.3.10 liste des offres .....	27
Figure 3.3.11 liste des candidats de l'offre.....	27
Figure 3.3.12 Fiche d'entretien (profil-offre) .....	28
Figure 3.3.13 liste des fiches d'entretien .....	28
Figure 3.3.14 Détail fiche d'entretien .....	29
Figure 3.3.15 Détail de Fiche d'entretien-les expériences .....	29
Figure 3.3.16 Modifier une expérience.....	30
Figure 3.3.17 offres et service .....	30
Figure 3.3.18 les offre de l'entreprise.....	31
Figure 3.3.19 Listes des offres-profils .....	31



# Introduction Générale

De nos jours, le marché de l'emploi connaît une évolution sans précédent : des métamoraux posés procéduraux, des mutations technologiques, des professions qui apparaissent et d'autres qui disparaissent. Dans ce turn-over de l'emploi, des nouveaux usages s'installent pour le processus de la recherche d'emploi et poussent les candidats ainsi que les entreprises à s'y adapter. Parmi ces nouveaux usages, on cite : le sourcing que l'on définit comme un processus de recherche pour le recrutement. Il a pour objectif d'identifier des profils précis répondant à une liste de critères de sélection pour un poste professionnel donné ainsi que et les sites d'emploi ou bien les jobs bord. Notre application web regroupe ces deux changements. C'est une plateforme pour un cabinet de sourcing qui propose des services au candidat ainsi aux entreprises, elle permet, d'un côté, aux candidats de gagner du temps pour la recherche d'emploi, au lieu de se déplacer au sein des entreprises durant quelques jours, ils peuvent comparer plusieurs offres et choisir celles qui leur semblent intéressantes en juste consacrant quelques minutes pour consulter les offres d'emploi, et de l'autre côté, elle permet aux entreprises de trouver les candidats idéales pour ses offres en juste postulant leurs détails de l'offre et le cabinet se charge d'organiser des entretiens aux candidats inscrits.

# CHAPTRE 1 :

## CADRE GENERALE DU PROJET

## 1. Introduction

Dans ce chapitre nous allons nous intéresser à la problématique de notre projet. Nous allons énumérer les objectifs et présenter le travail demandé.

## 2. Présentation de Openmind-service

Openmind Services est une start-up Française intervenant entre autres sur la formation, le développement de projets ainsi que le recrutement de profils dans les métiers en tension. Leur objectif est de faire correspondre les besoins et aspirations des entreprises clientes ainsi que des candidats sur le long terme.

Ils développent leurs activités HR autour de 4 axes :

1. L'accompagnement dans le recrutement de candidats en CDI en France ou dans l'Union Européenne (UE).
2. L'accompagnement dans le recrutement de candidats en CDI Hors UE et l'accompagnement dans les formalités d'introduction en France ou en UE auprès de l'OFII (prévoir 2 à 3 mois).
3. Le forfait de mise à disposition de profils techniques expérimentés et sélectionnés par vos soins à distance depuis notre plateau au Maroc pour palier à des besoins ponctuels tout en maîtrisant votre budget comme vos délais. Pour cette prestation nous garantissons le management de proximité et la réversibilité. Peut être envisagée comme une période d'essai pré-recrutement.
4. Accompagner la montée en compétence grâce à la formation continue et la certification de candidats grâce à nos partenariats technologiques et méthodologiques de premier plan (Amazon Web Services (Amazon, 2006), SAP (SE, 1972), International Scrum Institute...). Mise en pratique dans le cadre de la réalisation de projets innovants.

## 3. Présentation

Dans le cadre de ce chapitre, en premier temps, nous allons comprendre c'est quoi le sourcing, ses objectifs et ses valeurs ajoutés, Dans un deuxième temps, on va décrire la procédure de recrutement des recruteurs ainsi que les recherches effectuées par les candidats afin de trouver un emploi adéquats à ses compétences et à ses exigences. Ce bilan de l'existant amènera à la problématique du sujet et la méthodologie de travail adoptée.

## 4. Analyse de L'existant :

Le sourcing est un processus de recrutement digital qui vise à identifier les candidats répondant à un profil

précis défini par le recruteur. Cette technique RH de sourcing en recrutement est très populaire, car elle permet de gagner du temps dans le processus de recrutement et de cibler facilement les candidats pertinents et de réduire les coûts éventuels liés au recrutement.

En effet, puisque cette manière de recrutement est une méthode digitale, donc elle est considérée comme une carte gagnante pour les candidats, ce qui est demandé, c'est juste consacrer quelques minutes ou quelques heures pour consulter les offres d'emploi émanant des entreprises. Ainsi, en une journée, tout au plus, ils peuvent comparer plusieurs offres et choisir celles qui leur semblent intéressantes.

## 5. Problématique :

La société open sourcing ouvre la porte pour les entreprises pour postuler ses offres en envoyant des emails qui contiennent tous les détails de l'offre, le recruteur organise toutes ces offres par rapport aux profils demandés à l'aide du logiciel Trello. Ensuite, il cherche dans les sites de recrutement les différents profils et les communique pour avoir un rendez-vous dans lequel le recruteur tente de remplir un formulaire qui va l'envoyer à l'entreprise, après la confirmation du candidat sur l'offre.

Du coup plusieurs rubriques des offres ainsi des candidats sont chevéche, et il devient très dur pour le recruteur de trouver les bonnes offres pour des candidats ainsi de les attendre à confirmer leurs suggestions des offres et donc d'envoyer à la fin ces candidats aux entreprises et de l'attendre lui-même d'accepter un des candidats suggère.

Parmi les problèmes confrontés par le recruteur :

- Le recruteur doit chercher dans plusieurs sites de recrutement ainsi sur LinkedIn pour collecter les contacts et les CV des candidats qui cherchent des offres.
- Le non-ordonnancement du travail surtout quand le nombre des entreprises participées au service devient plus grand.
- Il peut arriver qu'un candidat a deux profils différents donc deux CV différents ce qui prête à confusion entre les profils.
- Le recruteur n'est pas toujours capable de déterminer si une offre est convenable à un candidat ou non car il n'est pas spécialisé dans le domaine en question.
- Le traitement d'une seule offre prend beaucoup de temps ce qui n'est pas favorable pour l'entreprise qui préfère d'avoir la réponse le plus tôt possible.

## 6. Solutions Proposée :

Les solutions proposées pour ces problèmes et qui traité par notre plateforme sont les suivants : au lieu d'avoir des rubriques non organisées et face à la dispersion, la plateforme propose des interfaces pour l'entreprise ainsi pour le back-office et pour les candidats, dans lesquelles :

- Le candidat peut facilement créer un compte et saisir ses informations et son CV.
- Chaque candidat inscrit peut avoir plusieurs CV et ainsi plusieurs profils dans la base de données.
- Une entreprise peut se connecter à son compte dans la plateforme et remplir les détails de son offre, ainsi un processus se lance pour déterminer les tags de la nouvelle offre ajoutée et calculer pour chaque profil le pourcentage de correspondre à cette offre.
- Le même processus pour un nouveau candidat, le système génère la liste des tags de ce candidat et calcule le pourcentage de correspondance de ce nouveau candidat pour chaque offre.
- Après ces calculs une liste des candidats générés pour chaque offre trié en ordre croissant par leurs notes (leur pourcentage de correspondance) comme ça le recruteur n'est pas indispensable aux offres choisies pour un candidat ou bien les candidats correspondent à une entreprise, et donc il n'est pas besoin des connaissances dans chaque domaine ou bien l'aide des autres spécialistes.

## 7. Planning prévisionnel

Notre travail a commencé par une étude théorique, premièrement nous avons défini notre projet par l'édition d'un cahier de charges selon les consignes du prof encadrent, afin de mieux comprendre les différents aspects fonctionnels du sujet qui nous a été proposé. Cette phase a été suivie d'une analyse et d'une spécification des besoins fonctionnels de notre système avant d'aborder la phase de conception pour passer finalement à la réalisation.

## 8. Cycle de vie et méthodologie utilisée

Avant de programmer l'application et se lancer dans l'écriture du code : il faut tout d'abord organiser les idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation. Cette démarche antérieure à l'écriture que l'on appelle modélisation ; son produit est un module.

La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Dans le cadre de notre projet nous avons utilisé l'UML ((OMG), 1997) pour la modélisation des différents diagrammes.

## 9. Cycle de vie d'un logiciel :

Le cycle de vie d'un logiciel (en anglais software life cycle), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

L'origine de ce découpage provient du constat que les erreurs ont un coût d'autant plus élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.

Le cycle de vie du logiciel comprend généralement au minimum les étapes suivantes :

- Définition des objectifs : Cette étape consiste à définir la finalité du projet et son inscription dans une stratégie globale.
- Des besoins et faisabilité : C'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes, puis l'estimation de la faisabilité de ces besoins.
- Spécifications ou conception générale : Il s'agit de l'élaboration des spécifications de l'architecture générale du projet.
- Conception détaillée : Cette étape consiste à définir précisément chaque sous-ensemble du projet.
- Codage (Implémentation ou programmation) : C'est la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- Qualification (ou recette) : C'est-à-dire la vérification de la conformité du projet aux spécifications initiales.
- Documentation : Elle vise à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- Mise en production : C'est le déploiement sur site du logiciel
- Maintenance : Elle comprend toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel

## 10. Besoins fonctionnels

Les besoins fonctionnels sur lesquels on va concentrer dans notre projet :

- Permettre à l'entreprise de :
  - ✓ Créer son compte et postuler ses offres, ainsi de consulter les détails de ses offres et les tags

générés pour son offre, et alors avoir une liste pour chaque offre des candidats triée par ordre croissant selon leur pourcentage de correspondance.

- ✓ Visualiser les détails du profil pour chaque candidat, par exemple son CV, son dossier de compétences, ses informations personnelles, le script de l'audio de son entretien.
- Permettre au Candidat de :
  - ✓ Créer son compte et postuler son CV.
  - ✓ des emails d'après le recruteur pour choisir un créneau pour passer l'entretien, ou bien pour le recruteur lui communiquer les offres qui acceptent sa candidature.
  - ✓ Consulter son profil et son dossier de compétence généré et liste des tags lui affecter.
  - ✓ Visualiser la liste générée des offres en ordre croissant avec sa note de correspondance et consulter le détail de chaque offre.
- Permettre au back office de :
  - ✓ Recevoir des offres des entreprises facilement pour avoir une base de données des offres intéressante.
  - ✓ Attirer l'attention des candidats pour s'inscrire aux systèmes et alors avoir une base de données intéressante des candidats au lieu de chercher dans les sites de recrutements les candidats qui cherchent du travail.
  - ✓ Construire une liste des tags généraux stockés dans la base de données initialement pour que le système puisse générer pour chaque offre et pour chaque candidat inscrit ses tags convenables.
  - ✓ Le processus de calcul des notes est sera automatique et donc le pourcentage de correspondance d'un candidat à une offre ou l'inverse est calculé par le système à l'aide d'une comparaison et pas par un effort de recruteur.
  - ✓ Avoir des modèles des emails prêts pour ne pas taper le même texte pour chaque candidat ou bien chaque entreprise.

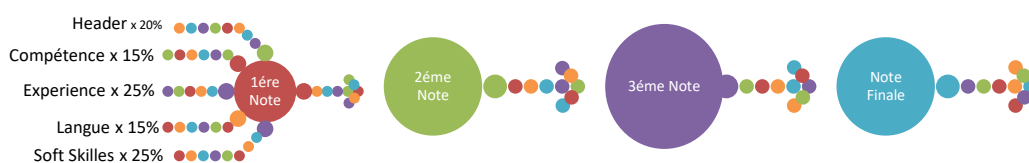
## 11. Besoins non fonctionnels

Les besoins non fonctionnels de l'application sont aussi à ne pas négliger. Il

S'agit des besoins qui caractérisent le système et qui assurent le bon fonctionnement du ce dernier.

- Performance : Le temps de réponse de l'application doit être rapide.
- Convivialité : L'application doit être facile à utiliser en termes de navigation entre les pages.
- Fiabilité : L'application doit être fiable et fonctionner sans erreurs.

## 12. Processus de calcul de la correspondance entre l'offre et la demande



## 13. Conclusion

Ce chapitre a été le point de départ pour l'élaboration du projet, dans la mesure où il décrivait la problématique, l'objectif général à atteindre ainsi que notre planning prévisionnel.



# CHAPTRE 2 :

---

## Analyse et Conception

---

## 1. Introduction

La conception est une étape fondamentale dans la réalisation d'un projet. Elle consiste à modéliser les composants du système à réaliser. Nous allons présenter tout au long de ce chapitre notre conception. Nous allons utiliser UML comme langage de modélisation.

## 2. Présentation du langage (UML)

UML (Unified Modeling Language) : UML en français Langage de Modélisation Unifié :

C'est un langage graphique de modélisation des données et des traitements, il est l'accomplissement de la fusion des précédents langages de modélisation objet : Booch (Booch, 1992), OOSE (Jacobson, 1995), OMT (TV, 1999), il est un standard défini par l'OMG (Object Management Group).

Le secteur informatique a adopté UML comme langage pivot, très complet, orienté objet et indépendant des langages de programmations, il est riche et ouvert. Nous avons adopté le langage UML comme langage de modélisation pour notre système d'arrosage, ce n'est pas un choix arbitraire, mais ceci est dû en fait à plusieurs raisons :

- UML ((OMG), 1997) est basée sur les mécanismes d'abstraction, de hiérarchie et de décomposition
- Elle considère un système comme une totalité organisée dont les éléments solidaires ne peuvent être définis que les uns par rapport aux autres.
- La modélisation orientée objet avec UML facilite la réutilisation du site.

## 3. Modélisation

### 3.1. Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation est une représentation du comportement du système de point de vue de l'utilisateur, c'est une définition des besoins qu'attend un utilisateur du système, il contient tous les cas d'utilisation en liaison directe ou indirecte avec les acteurs.

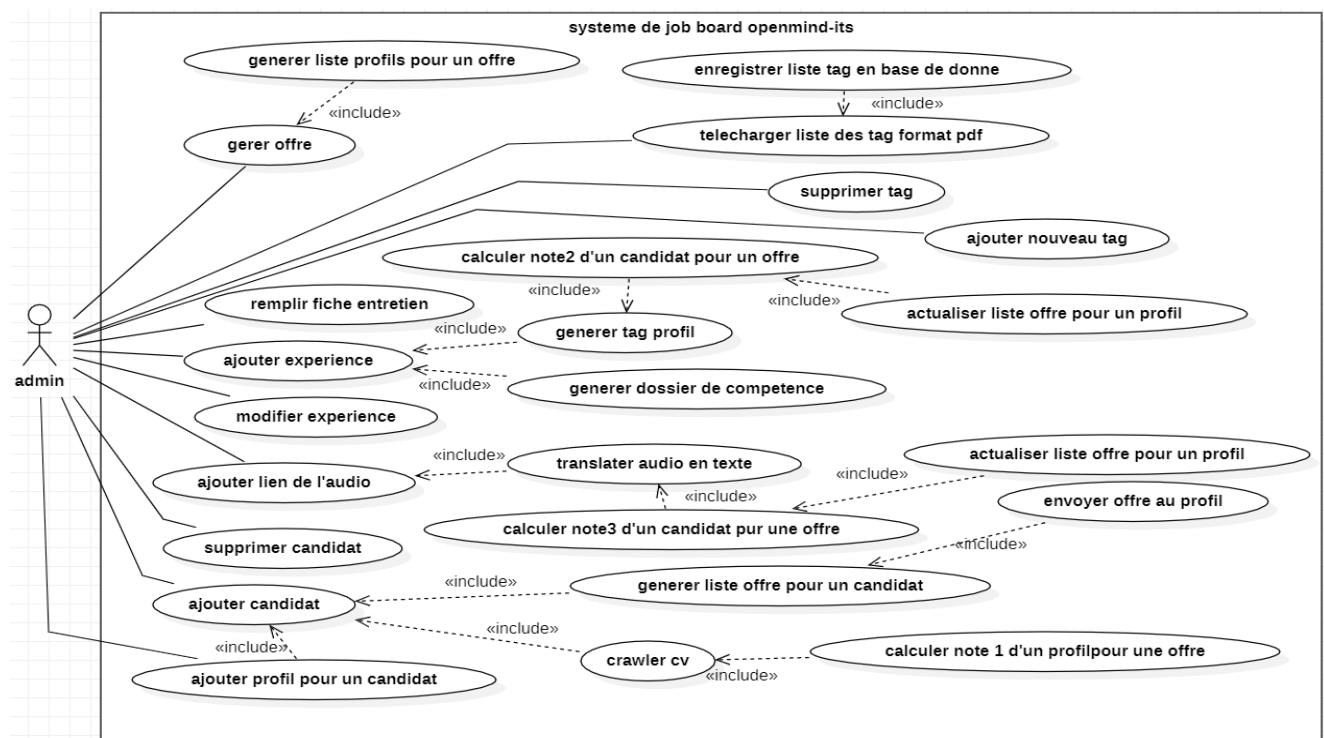
- Les acteurs : les acteurs se représentent sous la forme de petits personnages nommés, un acteur représente le rôle joué par une personne ou une classe qui interagit avec un système.
- Les cas d'utilisation : un cas d'utilisation est un classificateur qui modélise une fonctionnalité d'un système ou d'une classe. L'instanciation d'un cas d'utilisation se traduit par l'échange de message entre le système et ses acteurs. Il définit les activités attendues des différents utilisateurs par rapport à l'application. Le cas d'utilisation décrit ce que fait un système sans préciser comment il le fait : il donne un point de vue externe sur le système.

Nous aborderons trois acteurs principaux :

- Les candidats : cherchent de trouver des offres correspondantes à leurs profils.
- Les entreprises : qui cherchent de trouver des profils convenables à ses offres d'emplois.
- Le back Office : c'est Openmind servies qui vise à répondre aux besoins des entreprises et aux besoins des candidats, sa mission principale consiste à chercher des profils les plus convenables et

dans un temps réduit pour les offres et des offres adéquates pour les candidats postulés.

### 3.1.1. Diagramme de cas d'utilisation du backoffice :



**Figure 2.1 : Digramme de cas d'utilisation de Back Office**

- **Explication de diagramme :**

- Télécharger la liste des tags (les tags comme Java, PHP (Hypertext Preprocessor, 1994), machine Learning,) pour stocker cette liste dans la base de données et la possibilité de suppression de l'un des tags ou ajouter un nouveau tag dans cette liste : C'est la liste global des tags.
- Ajouter une offre : le crawler de l'offre pour obtenir les tags de l'offre à l'aide d'une comparaison avec la liste des tags global stocker dans la base de données. Après une comparaison de entre les tags de cette offre et tous les listes des tags des profils qui existent dans la base de données. Afin de générer à cette offre la liste des profils classés par une note correspondants à elle.
- Après avoir ajouté un candidat et son profil : le CV de ce candidat va être crawlé pour avoir la liste des tags de ce profil qui est obtenu par une comparaison avec la liste des tags global qui existe dans la base de données.

Ensuite le système va faire une comparaison entre cette liste des tags de ce profil et toutes les listes des tags des offres qui existe dans la base de données pour générer une note 1 à

ce profil par rapport à chaque offre et générer à ce profil une liste des offres qui lui correspond. Afin d'actualiser la liste des profils générer à chacun des offres.

- La possibilité de suppression d'un candidat
- Après l'entretien avec le profil pour un offre, le back office va remplir la fiche d'entretien, ajouter des expériences ou modifier d'expérience et il va remplir le dossier de compétence. Après on va calculer une note 2 de ce profile par rapport aux tags de la fiche d'entretien et de dossier de compétence la note de classement des profils par rapport à l'offre devient :  $Note = 0.5 * note1 + 0.5 * note2$
- Afin d'actualiser le classement de ce profil dans la liste des profils de cette offre par rapport à cette nouvelle note.
- Ajouter le lien de l'audio pour faire la transcription à l'audio afin d'obtenir un texte et calculer de la note 3 par rapport aux tags de ce texte. La note de classement des profils devient par rapport à l'offre devient :  $Note = 0.7 * note2 + 0.3 * note3$
- Afin d'actualiser le classement des profils de l'offre par rapport à cette note.
- Gérer l'offre : générer à cette offre la liste des profils que lui correspondent classer par une note. Et cette liste est actualisable à chaque fois qu'on calcule une nouvelle note.

### 3.1.2. Diagramme de cas d'utilisation de l'entreprise :

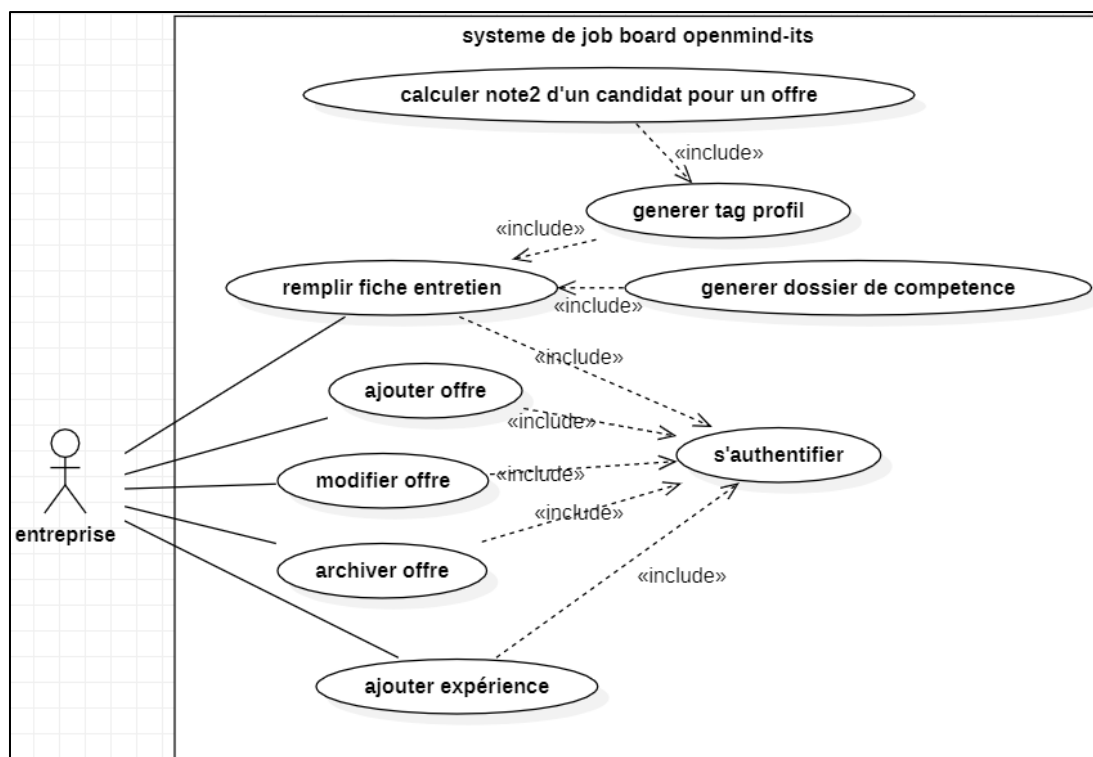


Figure 2. 2 : Diagramme de cas d'utilisation de l'Entreprise

### 3.1.3. Diagramme de cas d'utilisation du candidat :

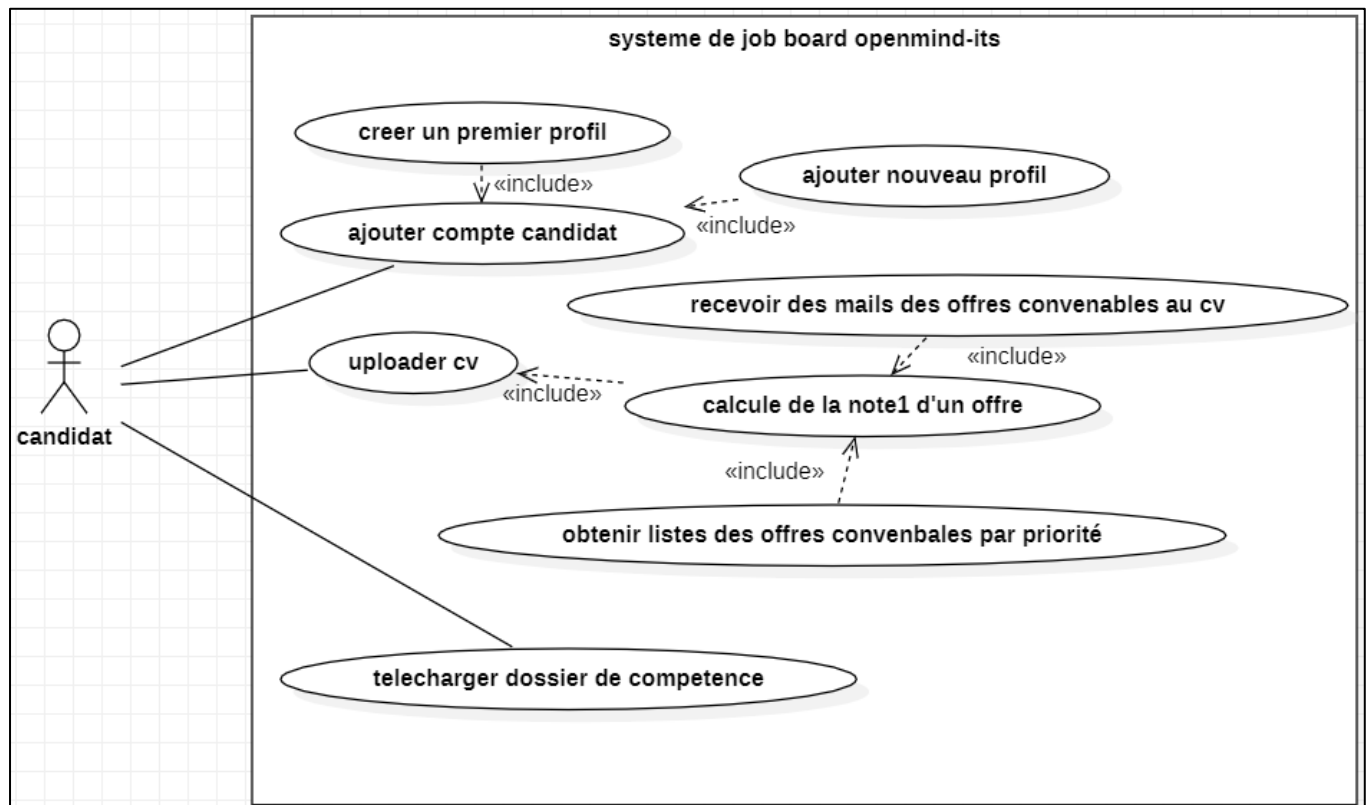


Figure 2.3 : Diagramme de cas d'utilisation des Candidats

### 3.2. Diagramme de Séquence

Un Diagramme de séquence permet de spécifier les interactions qui existent entre un groupe d'objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois des messages. Comme les diagrammes des séquences sont toujours lus du haut vers le bas, ils illustrent l'ordre dans lequel les messages sont envoyés entre les objets. Il peut illustrer un cas d'utilisation.

Les composants d'un diagramme de séquence sont les suivants :

- Les objets : sur un diagramme de séquence, les objets apparaissent toujours dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction
- Les messages : les messages sont représentés par des flèches directionnelles. Au-dessus de ces flèches figure un texte nous informant du message envoyé entre les objets.

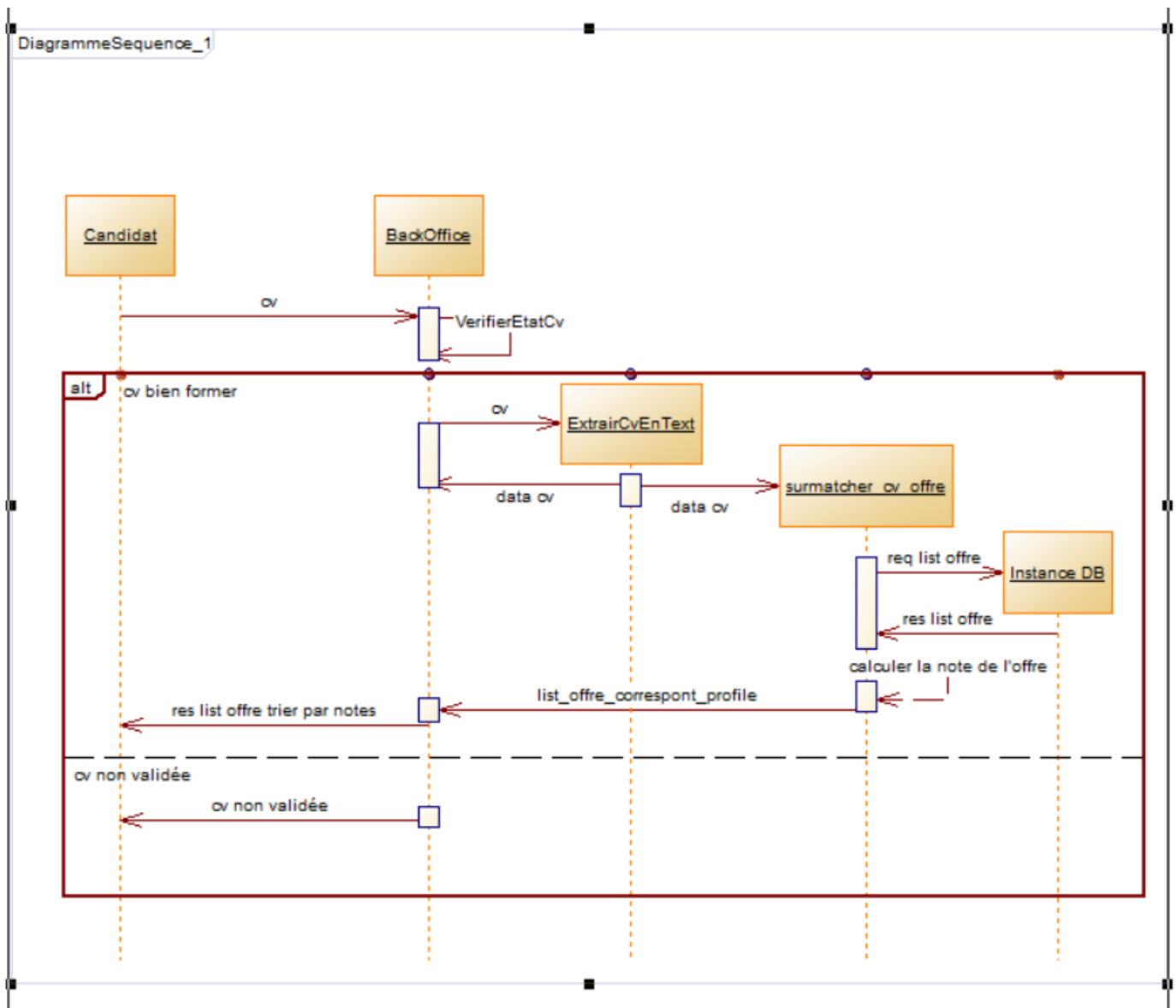


Figure 2.4 : Diagramme de séquence Ajouter profile à une offre

■ Explication du diagramme :

❖ **Le candidat et le back office :**

- Après que le candidat va créer son profil et uploader le cv, le back office va renvoyer à-il la liste des offres qui lui corresponds.
- Après que le profil choisi l'un ou plusieurs offres il va apparaitre dans la liste des profils classé proposer à l'offre de l'entreprise.

❖ **L'entreprise et le back office :** lorsque l'entreprise inséré un offre le back office va lui renvoyer une liste des profils classé par une note corresponds à cette offre. Avec un statut sur les profils qu'à valider cette offre ou non.

❖ **L'offre vers le profil :** Après que l'entreprise choisi l'un de ces profils ont un statut valider,

il va envoyer à-il un email d'acceptation.

### 3.3. Diagramme de Classes

Le diagramme de classes présenter les classes et les interfaces des systèmes ainsi que leurs relations ainsi que leurs relations la classe : décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

La figure suivante représente le diagramme de classe de notre système, dévoilant la manière avec laquelle la base de données est construite.

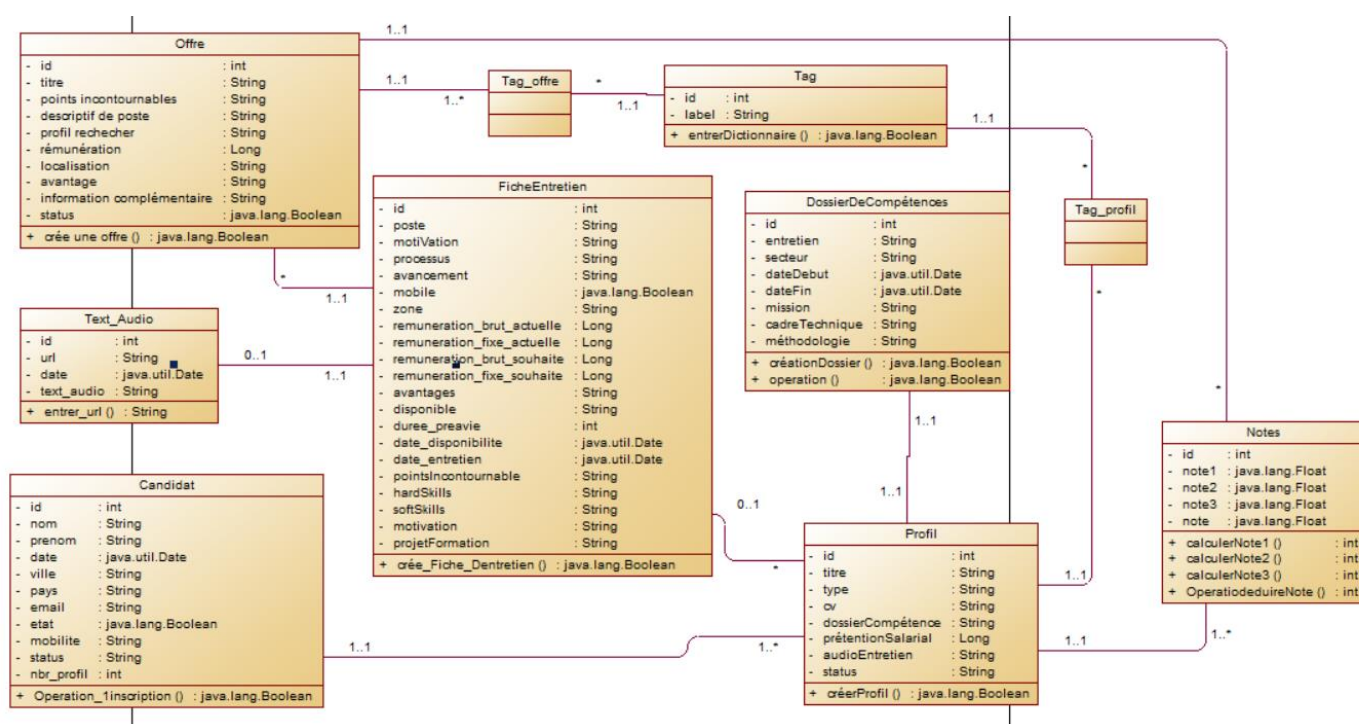


Figure 2.5 : Diagramme de classe

#### ■ Explication de diagramme :

##### ❖ Définitions des tables :

- Offre : contient des informations sur l'offre (titre de l'offre, points incontournable...) créé par l'entreprise. Les opérations possibles sur cette table créer l'offre, supprimer l'offre ou modifier l'offre.
- Candidat : contient les informations d'inscription de candidat sur la plateforme d'openmind (nom, prénom,...).
- Tag : contient tous les mots clé possibles comme compétence (PHP, JAVA (Naughton, 1982), python (Rossum, 1991), machine learning,...). Rempli automatiquement après l'insertion de dictionnaire des tags par le back office.

- Profil : contient les profils des candidatas qui cherchent une offre pour leurs profils.
- Dossier de compétences : contient des informations sur le profil.
- Fiche d'entretien : fiche créer par le back office après l'entretien il contient aussi des informations sur le profil et leurs requis (rémunération brute souhaité,...)
- Texte de l'audio : contient l'url vers l'audio de l'entretien et le texte de cette audio et la date de l'audio.
- Note : contient les notes d'un profil par rapport à une offre.

#### ❖ Les relations entre les tables :

- Offre & tag : chaque offre est liée à un 1 tag au plusieurs et chaque tag sa peut être dans plusieurs ou aucun offre, donc c'est une relation plusieurs à plusieurs, c'est pour ça on créer la table tag offre entre ces deux tables.
- Offre & fiche d'entretien & profil : l'offre est liée avec le profil par la fiche d'entretien dans laquelle enregistré les requises de l'offre satisfait par le profil.
- Offre & note & profil : dans la table note existe la note de profil par rapport à l'offre.
- Profil & tag : chaque profil a un 1 tag au plusieurs et chaque tag sa peut être dans plusieurs ou aucun profil, donc c'est une relation plusieurs à plusieurs, c'est pour ça on créer la table tag profil entre ces deux tables.
- Profil & candidat : le candidat peut avoir plusieurs profils, un profil est lié à un seul candidat.
- Profil & dossier de compétences : un profil est lié à un seul dossier de compétences et un dossier de compétences est lié à un seul profil
- Fiche d'entretien & texte de l'audio : le texte de l'audio contient la transaction de l'audio.

## 4. CONCLUSION :

Dans ce chapitre, nous avons présenté une phase indispensable à l'étude et l'analyse de notre site. Nous avons défini les acteurs de cette application, et modélisé les cas d'utilisation. Ce chapitre représente la phase de conception, c'est la partie la plus importante avant l'implémentation. On a pu modéliser les différentes opérations effectuées sur le site. Nous avons défini le Diagramme de classe et le Diagramme de séquence. Dans le prochain chapitre, on présentera la réalisation du site web, ainsi que les outils de développement utilisés.



## CHAPITRE 3 :

---

### Réalisation de l'application

---

## 1. Introduction

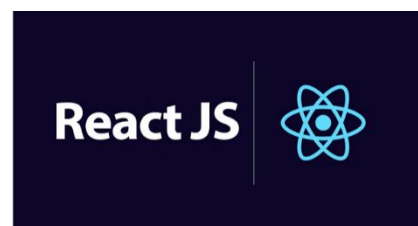
Dans ce chapitre, nous discuterons le cadre technique de notre application, nous décrire la phase de réalisation. Nous allons présenter, en premier lieu, l'environnement du travail utilisé pour le développement de l'application, ensuite, nous allons donner un aperçu sur le travail accompli à travers des captures d'écran.

## 2. Environnement du travail

Pour l'implémentation de l'architecture proposée dans le chapitre précédent on a utilisé principalement les technologies et outils suivants :

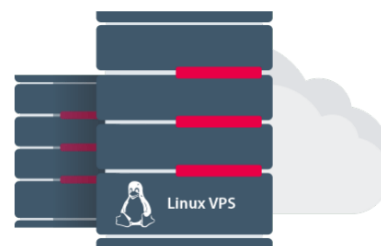
- **React JS :**

Une bibliothèque JavaScript pour créer des interfaces utilisateur. Est facilite la création d'interfaces utilisateur interactives. Concevez des vues simples pour chaque état de votre application, et React mettra à jour et affichera efficacement les bons composants lorsque vos données changent. Les vues déclaratives rendent votre code plus prévisible et plus facile à déboguer. Créez des composants encapsulés qui gèrent leur propre état, puis composez-les pour créer des interfaces utilisateur complexes. Étant donné que la logique des composants est écrite en JavaScript au lieu de modèles, vous pouvez facilement transmettre des données riches via votre application et garder l'état hors du DOM (Document Object Model). Nous ne faisons aucune hypothèse sur le reste de votre pile technologique, vous pouvez donc développer de nouvelles fonctionnalités dans React sans réécrire le code existant. React peut également effectuer un rendu sur le serveur à l'aide de Node et alimenter des applications mobiles à l'aide de React Native.



- **Linux VPS :**

Linux VPS Virtual Server (Un serveur Linux est une variante efficace et puissante du système d'exploitation Linux open source (OS). Les serveurs Linux sont conçus pour répondre aux exigences toujours croissantes des applications d'entreprise telles que l'administration système et réseau, les services Web et la gestion de bases de données.



Les serveurs Linux sont souvent préférés aux autres systèmes d'exploitation serveur en raison de leur réputation de sécurité, de cohérence et de flexibilité. Quelques exemples de systèmes d'exploitation de serveur Linux sont CentOS, Ubuntu Server, Gentoo, Debian, Slackware, etc. Linux est considéré comme la meilleure option pour les serveurs d'exploitation car il n'y a pas besoin d'une interface utilisateur graphique ; toutes les commandes peuvent être exécutées via l'invite de commande. Cela aide les utilisateurs à atteindre des performances système maximales.

- **PHP (Hypertext Preprocessor) :**

PHP (Hypertext Preprocessor) est un langage de script qui est principalement utilisé pour être exécuté par un serveur HTTP, mais il peut fonctionner comme n'importe quel langage interprété en utilisant les scripts et son interpréteur sur un ordinateur. On désigne parfois PHP comme une plate-forme plus qu'un simple langage.



Le langage PHP est un langage extrêmement puissant : il permet de créer des pages web, au travers lesquelles l'utilisateur peut échanger des informations avec le serveur ; c'est ce qu'on appelle des pages web dynamiques.

- **Laravel :**

LARAVEL est un Framework du langage de programmation PHP.

Laravel fonctionne avec son propre système de Template blade. Ainsi, il possède des vues qui ont l'extension blade.php et dans lesquelles il est possible d'afficher le contenu des variables PHP. Par ailleurs, il est



indispensable de mentionner que Laravel adopte une architecture en MVC pour modèle vue contrôleur. Ce framework peut générer des formulaires et créer des layouts. En effet, ces layouts sont des modèles de base pour les pages html. Avec Blade, vous pourrez aisément vous servir de différentes instructions notamment des instructions conditionnelles et itératives. Elles sont d'une grande efficacité dans la gestion du code. Par ailleurs, par son fonctionnement, Laravel offre bien une grande sécurité. Entre autres, les formulaires créés génèrent des tokens qui empêchent les attaques CSRF. En effet, les Cross-Site Request Forgery sont des attaques consistant à contourner l'authentification d'un site afin d'effectuer des actions malveillantes pouvant être envoyées par un formulaire. En outre, le framework Laravel dispose d'un système de routage qui est compatible avec toutes les méthodes http.

- **github :**

GitHub est un service d'hébergement Internet pour le développement de logiciels et le contrôle de version à l'aide de Git. Il fournit le contrôle de version distribué de Git plus le contrôle d'accès, le suivi des bogues, les demandes de fonctionnalités logicielles, la gestion des tâches, l'intégration continue et les wikis pour chaque projet. Basé en Californie, c'est une filiale de Microsoft depuis 2018.



Couramment utilisé pour héberger des projets de développement de logiciels open source. En juin 2022, GitHub a signalé avoir plus de 83 millions de développeurs et plus de 200 millions de référentiels, dont au moins 28 millions de référentiels publics. C'est le plus grand hôte de code source en novembre 2021.

- **MySQL :**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde<sup>3</sup>, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server. Son nom vient du prénom de la fille du cocréateur Michael Widenius, My (sv) (prononcer [my]). SQL fait référence au Structured Query Language, le langage de requête utilisé. MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains<sup>4</sup>. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Data base et MySQL . Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010<sup>5,6</sup>. Depuis mai 2009, son créateur Michael Widenius a créé MariaDB (Maria est le prénom de sa deuxième fille) pour continuer son développement en tant que projet Open Source.



- **Visual Studio Code :**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, la refactorisation du code et Git (Conservancy, 2005) intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. Sa facilité d'utilisation ainsi que ses fonctionnalités diverses justifient son utilisation.



- **MailTrap :**

mailtrap : Lorsque vous développez un site, une application SAAS, vous allez devoir envoyer des emails. Mais étant en phase de développement, vous ne souhaitez pas expédier un email à un réel utilisateur. Nous allons voir comment capturer les emails que vous voulez expédier. Capturer vos emails Le but d'un outil comme Mailtrap.io (Freemium) est de vous permettre de récupérer sur une interface web, le contenu de l'email que vous avez expédié depuis votre site. Ainsi vous ne risquez pas, par erreur, d'expédier un email sur un client, abonné, réel.



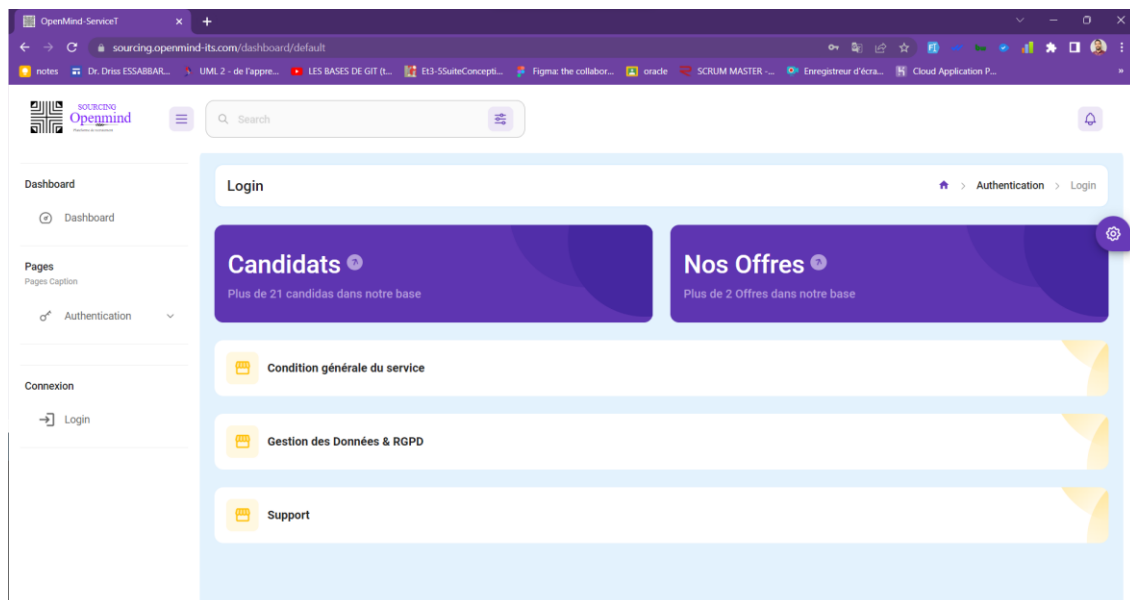
- **SnowflakeSSH :**

SnowflakeSSH (SSH client, 2013) Snowflake est l'une des solutions d'entrepôt de données les plus fiables actuellement sur le marché. Il offre un large éventail de fonctionnalités faciles d'accès et d'utilisation. Principales caractéristiques de Snowflake. Le magasin de métadonnées snowflakes facilite et accélère les choses.' Pour accéder aux données. Par conséquent, les flocons de neige créent un réseau de fournisseurs de données et de consommateurs qui permet de nombreux cas d'utilisation. Pour ceux qui ne possèdent pas de compte, snowflake offre la possibilité de créer un compte de lecteur, un moyen économique qui permet aux consommateurs d'accéder gratuitement aux données partagées

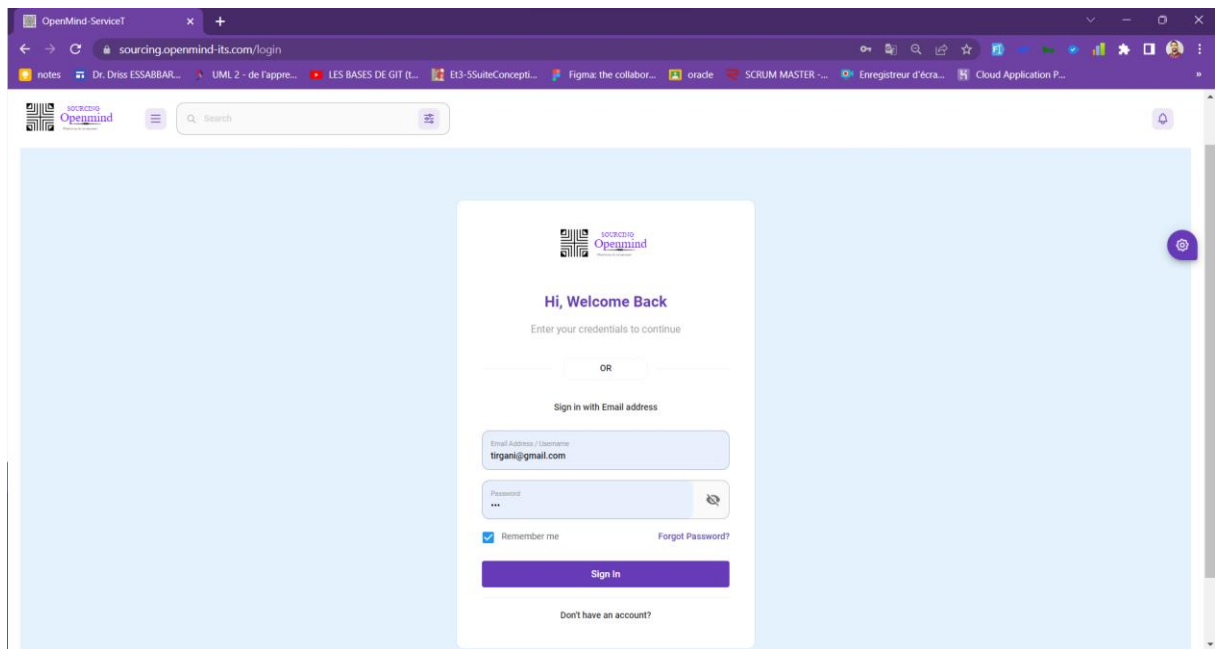


### 3. Réalisation

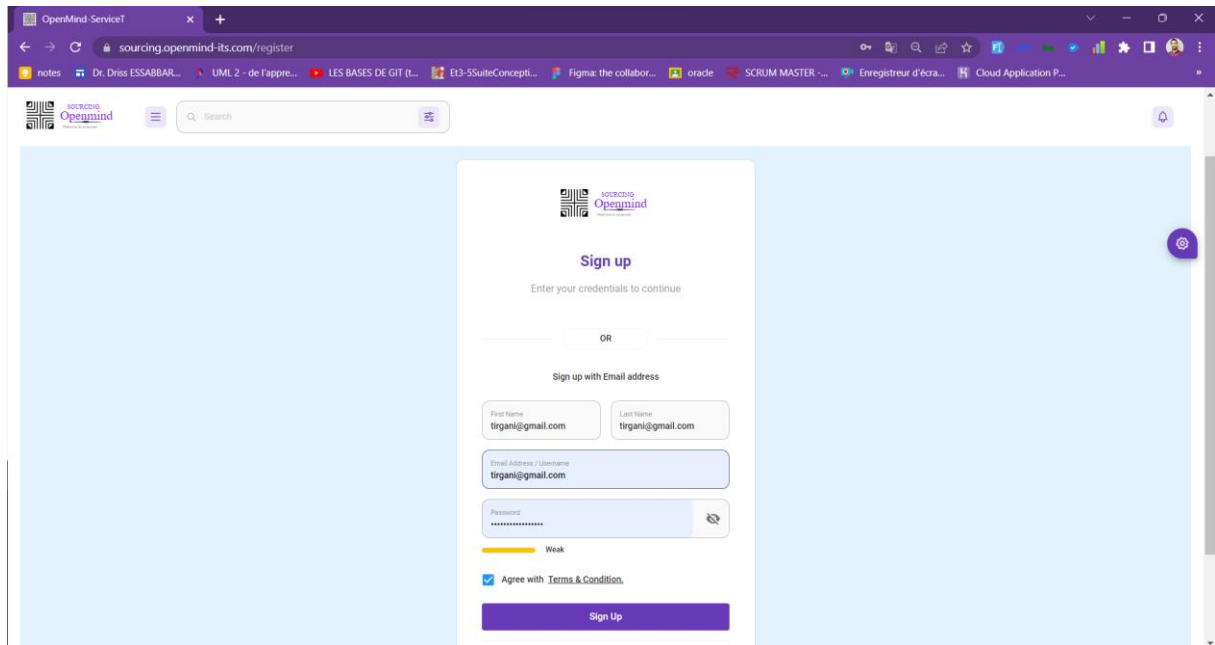
Dans cette partie, nous allons présenter notre application à travers une série de captures d'écran illustrant un scénario d'utilisation.



**Figure 3.3.1– Page d'accueil sans authentification**



**Figure 3.3.1– Page de connexion**



**Figure 3.3.1– Page de registration**

**Accès recruteur après authentification :**

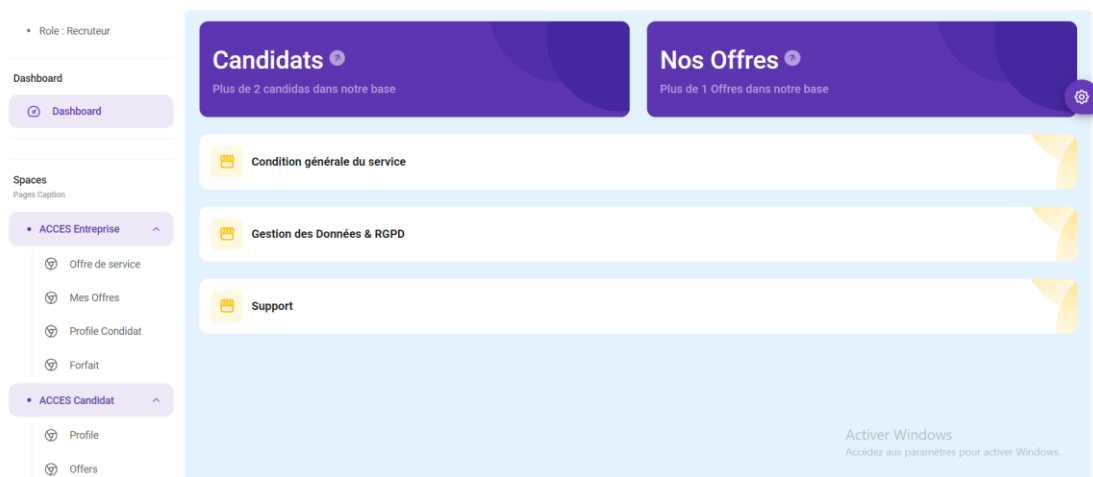


Figure 3.3.1– Page d'accueil

- Liste d'utilisateurs de la plateforme :

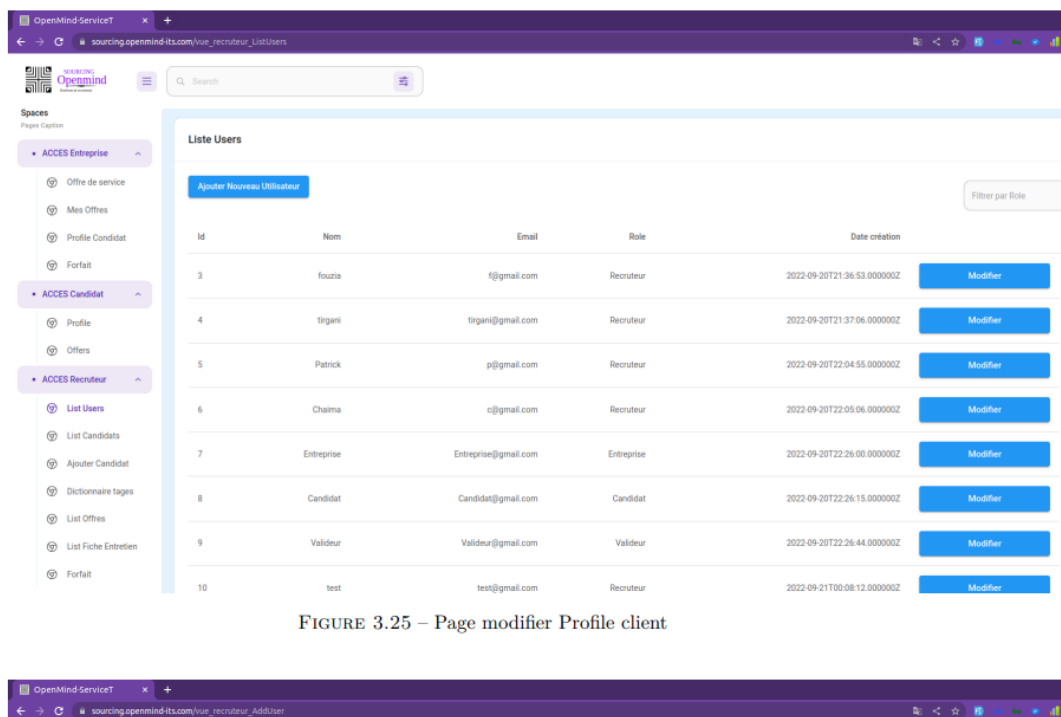


FIGURE 3.25 – Page modifier Profile client

Figure 3.3.2- liste des utilisateurs

- Ajouter un utilisateur :

**Ajouter Utilisateur**

Nom  Email

Password  Role

[+ Ajouter](#)

Activer Windows  
Accédez aux paramètres pour activer Windows.

**Figure 3.3.3 pour ajouter un utilisateur**

- Liste des candidats et pour ajouter un candidat :

**Liste Candidats**

[Ajouter Candidat](#) Filtrer par Status

Id	Nom	Prenom	status	Date de Naissance	
1	BenSadouk	Fouad	active	2022-09-15	<a href="#">Detail</a> <a href="#">Archivé</a>

Activer Windows  
Accédez aux paramètres pour activer Windows.

**Figure 3.3.4 listes des candidats**

- Pour accéder aux détails candidats :

**Détail Candidat**

[Ajouter Profile](#) [Envoyer Email](#)

détails du candidat : BenSadouk Fouad

Nom Complet : -BenSadouk Fouad  
Email : -@gmail.com  
Date de Naissance : -2022-09-15  
Ville : -France  
Pays : -France  
Etat du profile : -active  
Audio Entretien : -  
Mobilité : -mobile  
Status : -active  
Nombre de profile : -2  
[Modifier Candidat](#)

> Profile : Ingénieur  
> Profile : Ingénieur Développeur REACT

Activer Windows  
Accédez aux paramètres pour activer Windows.

**Figure 3.3.5 détail candidats**



- Accès aux profils de candidats, ajouter un profil :

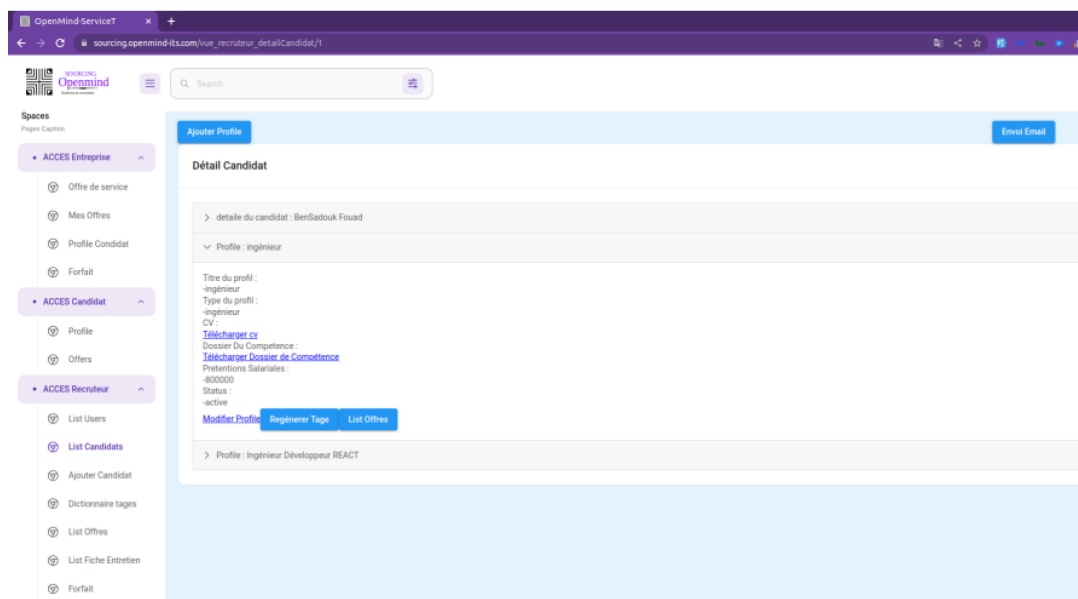


Figure 3.3.6 profils Candidats

- Accès aux listes des offres correspondants à ces profils

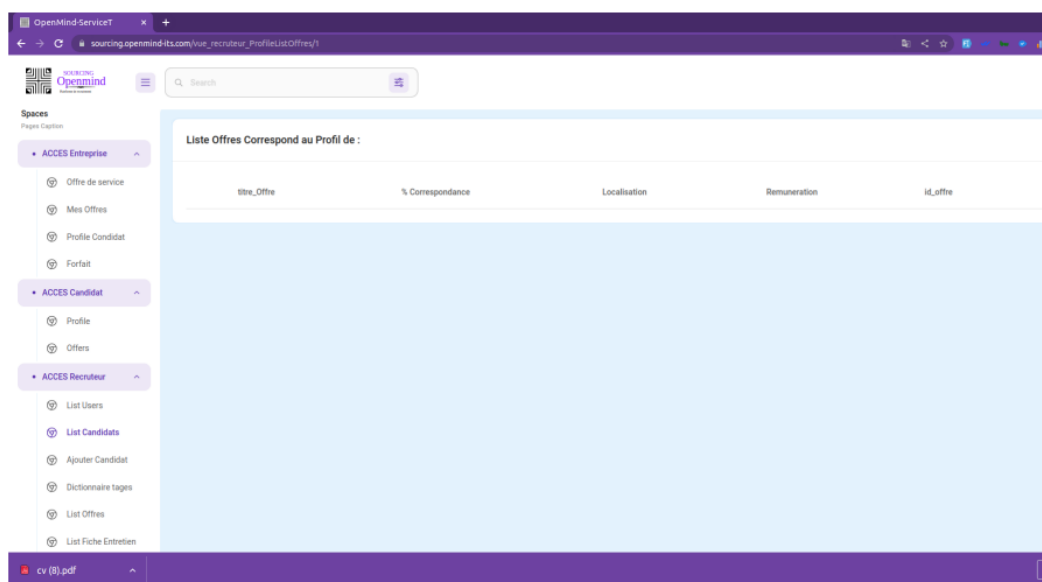


Figure 3.3.7 \_liste des offres de profil

- Ajouter un Candidat :

OpenMind-ServiceT

sourcing.openmind-its.com/vue\_recruteur\_addCandidat

Openmind

Offre de service

Mes Offres

Profile Candidat

Forfait

ACCES Candidat

Profile

Offers

ACCES Recruteur

List Users

List Candidats

Ajouter Candidat

Dictionnaire tags

List Offres

List Fiche Entretien

Forfait

ACCES Valideur

cv (8).pdf

Ajouter Candidat

Nom

Prénom

date Naissance

j/mm/aaaa

Ville

Pays

Email

Choisir un fichier

Aucun fichier choisi

Titre Profile

Status de l'offre

Mobile

Type Profile

Prévention Salarial

Ajouter

**Figure 3.3.8\_ Ajouter Candidats**

- L'accès au Dictionnaire des tags avec la possibilité d'ajouter ou supprimer un tag :

OpenMind-ServiceT

sourcing.openmind-its.com/vue\_recruteur\_dictionnaire

Openmind

Offre de service

Mes Offres

Profile Candidat

Forfait

ACCES Candidat

Profile

Offers

ACCES Recruteur

List Users

List Candidats

Ajouter Candidat

Dictionnaire tags

List Offres

List Fiche Entretien

Forfait

ACCES Valideur

cv (8).pdf

Dictionnaire des tags

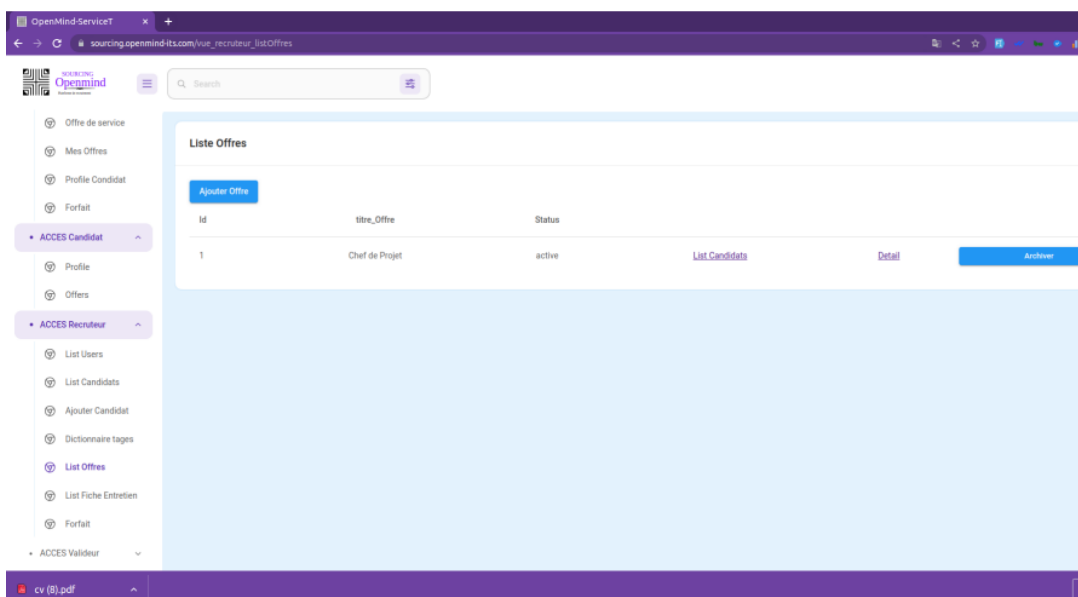
Ajouter Tag

Search by Task

ID	LABEL	
1	développement	Supprimer
2	implémentation de plateformes	Supprimer
3	suit projet	Supprimer
4	développeur backend	Supprimer
5	polyvalent	Supprimer
6	frontend	Supprimer
7	fullstack	Supprimer
8	methodes agiles	Supprimer

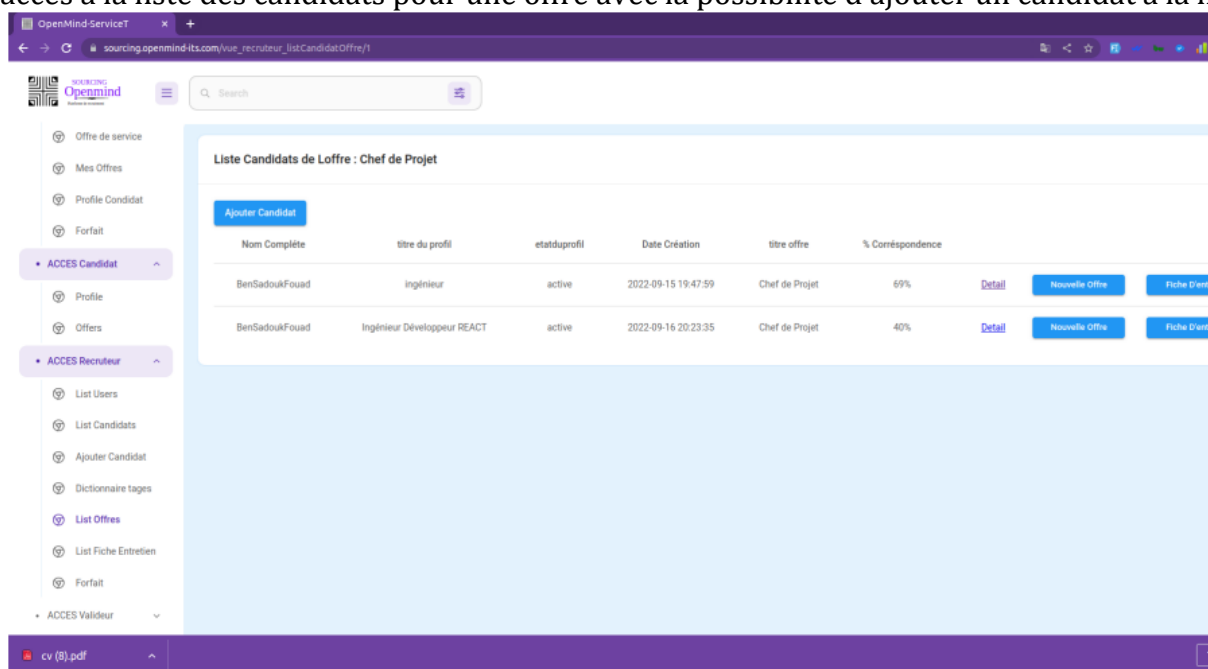
**Figure 3.3.9 \_ dictionnaire des tags**

- L'accès à la liste des offres avec la possibilité d'ajouter ou supprimer une offre :



**Figure 3.3.10 liste des offres**

- L'accès à la liste des candidats pour une offre avec la possibilité d'ajouter un candidat à la liste.



**Figure 3.3.11 liste des candidats de l'offre**

- Accès à une Fiche d'entretien de chaque profil et l'offre :

OpenMindServiceT

sourcing.openmind-its.com/vue\_recuteur\_ficheDentretien/1/1

Openmind

Search

Offre de service

Mes Offres

Profile Candidat

Forfait

• ACCES Candidat

Profile

Offers

• ACCES Recruteur

List Users

List Candidats

Ajouter Candidat

Dictionnaire tags

List Offres

List Fiche Entretien

Forfait

• ACCES Valideur

Merci d'insérer tout les champs !

Fiche d'entretien + Dossier de Compétence

Le candi a-t-il en poste ?

Quelle est sa motivation pour ce changement de poste ? \*

Le candidat a-t-il d'autres processus de recrutement en cours ? \*

Recruteur

Quel est le niveau d'avancement de ces autres processus ? \*

Le candidat est-il mobile si nécessaire ? \*

Si oui précisez la zone de mobilité

Rémunération brute actuelle \*

Dont rémunération fixe:

Rémunération brut souhaitée:

Dont rémunération fixe:

Avantages souhaités:

Le candidat est-il ?

duree\_preavis

Date de disponibilité \*

Le candidat a été vu en entretien visio le \*

Listez les points incontournables du candidat en lien avec ceux définis dans l'offre \*

Compétences (hard skills) \*

cv (9).pdf

Figure 3.3.12 Fiche d'entretien (profil-offre)

- Liste des fiches d'entretien :

Liste Fiche Entretien

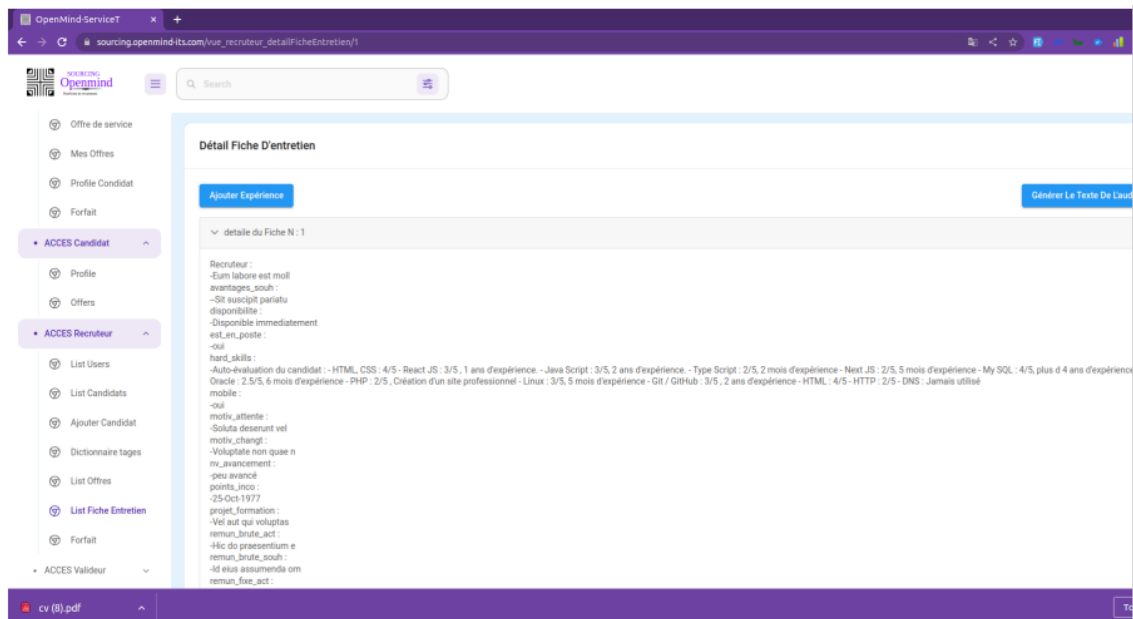
Filtrer par Status

Recruteur	Titre Offre	Nom Complète	Titre du profil	Audio(Oui,Non)	Date Creation	
Eum labore est moll	oui	Voluptate non quae n	Magni minima blandit	peu avancé	2022-09-15T19:52:01.000000Z	<a href="#">Detail</a> <a href="#">Modifier</a>

Activier Windows

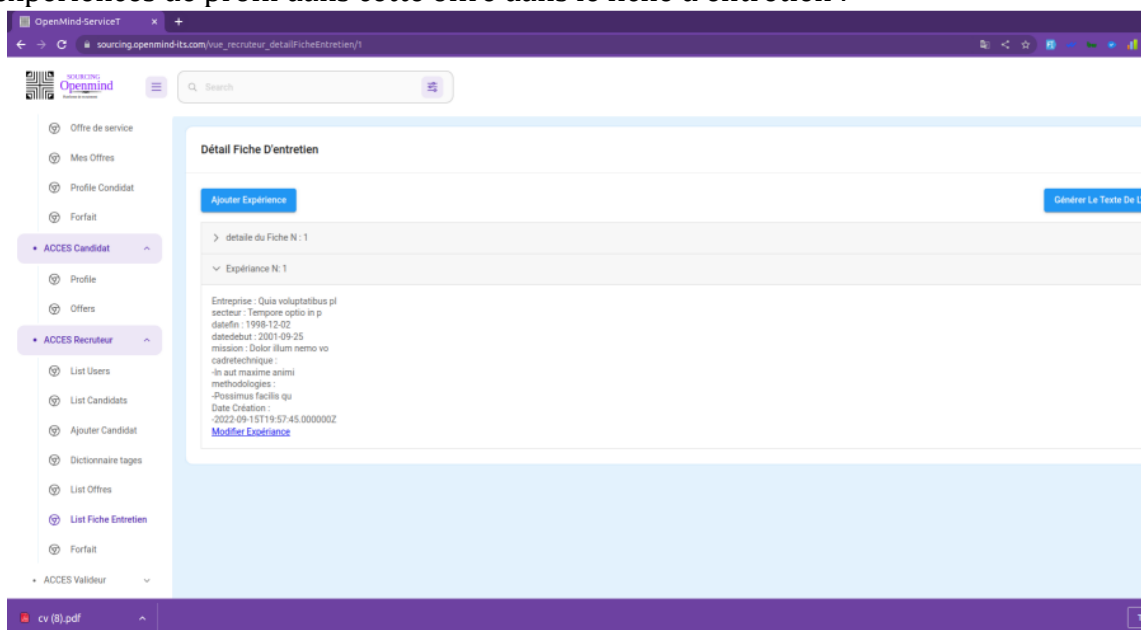
Figure 3.3.13 liste des fiches d'entretien

- Détail de chaque Fiche d'Entretien :



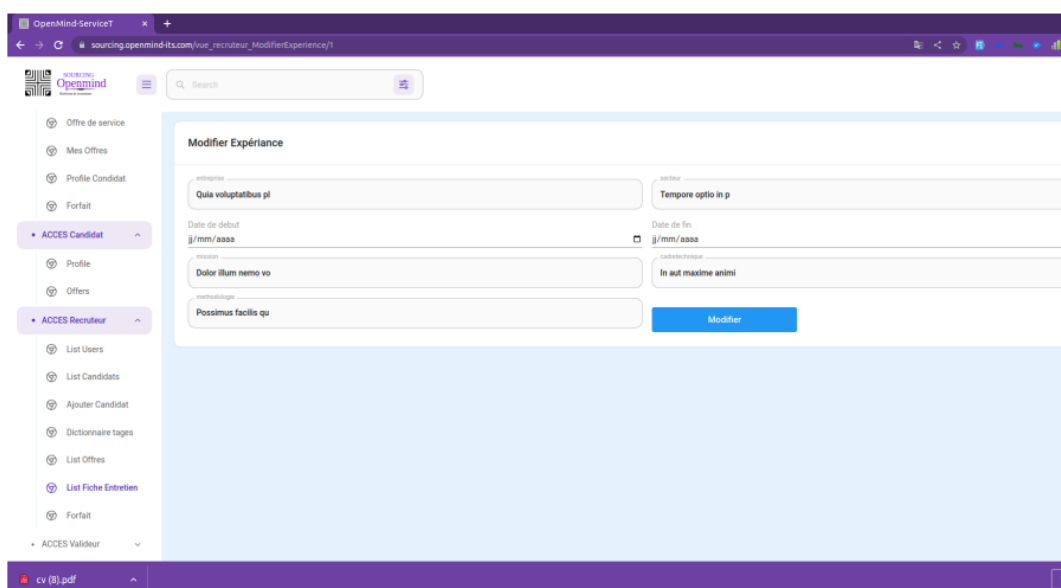
**Figure 3.3.14 Détail fiche d'entretien**

- Les expériences de profil dans cette offre dans le fiche d'entretien :



**Figure 3.3.15 Détail de Fiche d'entretien-les expériences**

- Modifier une expérience :



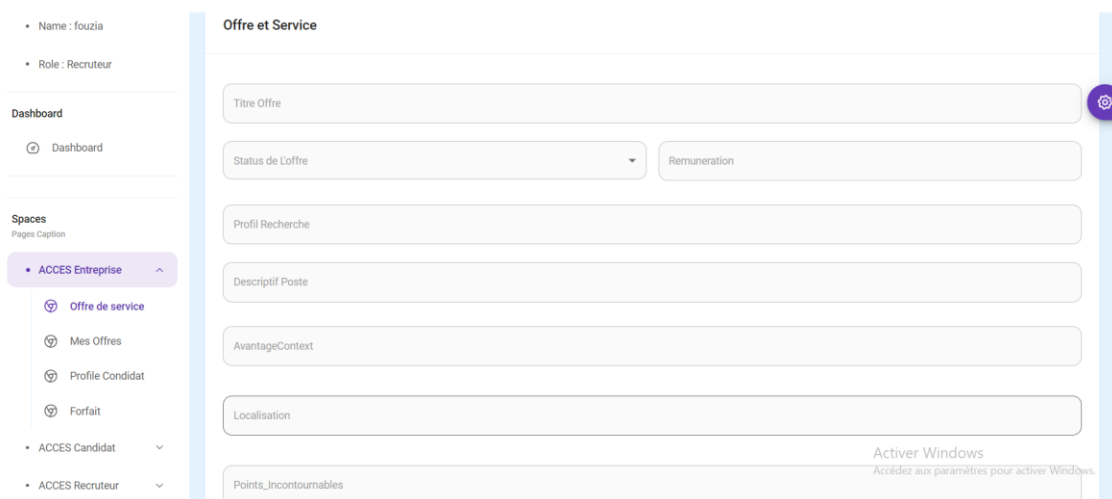
**Figure 3.3.16 Modifier une expérience**

#### 1.1.1.1 Accès Candidat :

- L'accès à ces profils (les profils de ce candidat) : tâche en-cours
- L'accès aux offres qui correspondent à chacun de ces profils : tâche en-cours

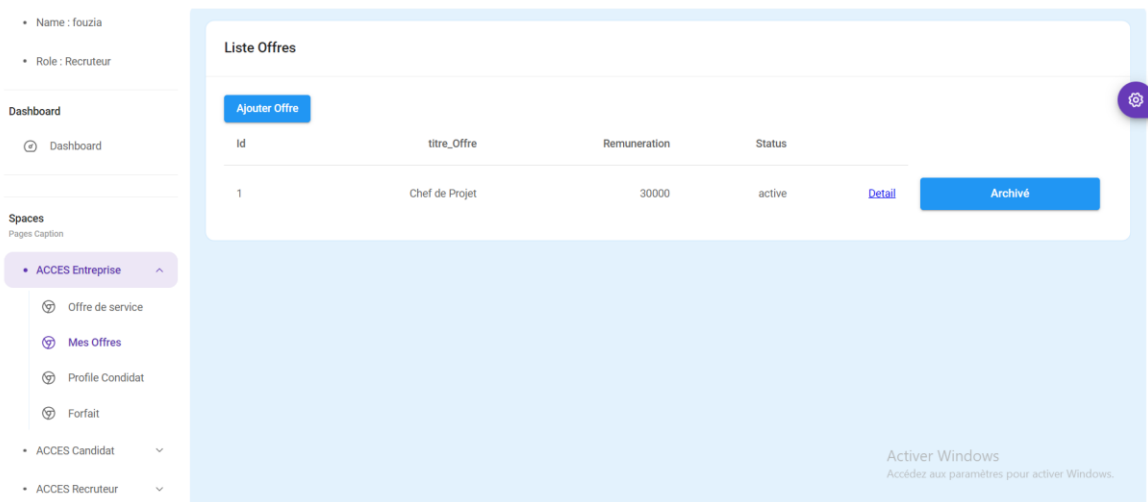
#### 1.1.1.2 Accès Entreprise :

- Offre de service : pour que l'entreprise arrive à crée une offre



**Figure 3.3.17 offres et service**

- Mes offres : qui contiennent les offres offertes par cette entreprise, il peut archiver ou ajouter une offre.



**Figure 3.3.18 les offre de l'entreprise**

- Profil candidat : contient les profils qui correspondent à chacun des offres de l'entreprise. Chaque offre à son nombre des profils avec l'état de cette offre

Offres	Nbr Candidas	Identification	Etat	
Offre 1	15	1	validée	<a href="#">Detail</a>
Offre 2	1	2	validée	<a href="#">Detail</a>
Offre 3	115	3	Non validée	<a href="#">Detail</a>
Offre 4	25	4	validée	<a href="#">Detail</a>
Offre 5	2	5	Non validée	<a href="#">Detail</a>

**Figure 3.3.19 Listes des offres-profils**

- Pour le forfait on n'est pas arrivé à réaliser cette tâche.

#### 4. Conclusion

Ce chapitre représente une récapitulation de tout le travail élaboré pendant ce projet de conception et de développement ainsi qu'une présentation des résultats atteints. En effet, nous avons décrit les outils et environnements sur lesquels nous nous sommes basés pour construire notre application. Enfin, nous avons passé à la présentation de quelques interfaces qui montrent l'ensemble des fonctionnalités de base de notre site Web.

## Conclusion générale :

En guise de conclusion, nous vous avons présenté tout au long de ce rapport la démarche que nous avons suivie pour mettre au point ce site web, en commençant par la présentation de l'entreprise, du sujet et du cahier de charge pour, ensuite, aborder la partie de conception et, enfin, de réalisation finale de site Web.

Ce projet nous a permis d'appliquer les connaissances qui nous ont été facilitées et transmises au cours de cette année depuis l'analyse et la conception d'une application jusqu'à son implémentation. De plus, il nous a permis, entre autres, de développer nos aptitudes de travail en équipe, de communication et de résolution des problèmes qui surgissent lors du déroulement d'un projet. Nous pourrions tout de même développer encore plus ce travail pour le rendre plus interactif et améliorer ses fonctionnalités.



# Références

- (s.d.).
- (OMG), O. M. (1997). *Langage de Modélisation Unifié*.
- Amazon. (2006). *Amazon Web Services (AWS)*.
- Booch, G. (1992). *La méthode Booch*.
- Conservancy, S. F. (2005). *logiciel de gestion de versions décentralisé*.
- (s.d.). *Document Object Model*. 1992.
- (1994). *Hypertext Preprocessor*.
- Jacobson, I. (1995). *Object Oriented Software Engineering*.
- MySQL AB, S. M. (1995). *système de gestion de bases de données relationnelles*.
- Naughton, J. G. (1982). *langage de programmation orienté objet*.
- Otwell, T. (2011). *framework web open-source écrit en PHP*.
- Platforms, M. (2013). *React.js*. Facebook, Instagram et la communauté.
- Rossum, G. v. (1991). *langage de programmation interprété*.
- SE, S. (1972). *logiciel de planification des ressources d'entreprise*.
- (2013). *SSH client*.
- Sysoev, I. (2019). *nginx*.
- (2008). *système de gestion de base de données*.
- (s.d.). *Système de gestion de base de données*.
- Tenhunen., S. (2021). *Expo react native-sovellusten*. kehitysalustana. .
- TV, O. (1999). *chaîne de télévision française*.
- Unix. (s.d.). *système d'exploitation GNU/Linux*.

