



Réf : PFE-2021-09

# Rapport de Projet de Fin d'Études

Pour obtenir le

**Diplôme d'Ingénieur en Génie Informatique**  
**Option : Génie Logiciel et Informatique Décisionnelle**

Présenté et soutenu publiquement le 09 juillet 2021 Par

**Nawras HALOUI**

---

## Conception et développement d'une application de recrutement et d'analyse des Curriculum vitae

---

### Composition du jury

Madame	Yassine Faleh	Président
Madame	Narjes Hachani	Rapporteur
Monsieur	Skandar Bacha	Encadreur Entreprise
Madame	Ines Bayoudh Saadi	Encadreur ENSIT

Année universitaire : 2020-2021

## **Dédicaces**

*Je dédie ce travail :*

*À mon très cher père, à ma tendre mère, à qui je ne pourrais jamais exprimer ma gratitude, Que ce travail soit le couronnement de vos sacrifices, vos efforts et votre dévouement absolu.*

*À mon frère et ma sœur, à vous mes amours, nulle dédicace n'exprime ma gratitude de vous avoir dans ma vie, que Dieu le tout puissant vous préserve et vous accorde bonheur et réussite...*

*Aux membres de ma famille, pour leur soutien et leur encouragement continu.*

*À mes amis, mes camarades et tous ceux qui me sont chers, merci pour les bons moments que nous avons vécus ensemble.*

*J'espère que ce travail sera pour eux une petite compensation pour les sacrifices qu'ils ont toujours consentis à mon égard.*

*Haloui Nawras*

## **Remerciements**

Au terme de ce projet je tiens à remercier tous ceux qui ont participé de près ou de loin à ce travail et j'espère qu'ils trouveront un vrai témoignage de ma sincère reconnaissance.

Je tiens à exprimer mes sincères remerciements à toutes les personnes qui m'ont aidé à passer ce Projet de Fin d'Étude au sein de la société BFI dans des conditions favorables.

Je tiens à remercier Monsieur Bacha Skandar, mon encadrant chez BFI pour la confiance qu'il m'a accordée pour mener à bien ce projet.

Je tiens à remercier Madame Ines Bayoudh Saadi Ines, mon encadrante au sein de l'École Nationale Supérieure d'Ingénieurs de Tunis pour son assistance, son support et son aide précieuse durant toute la période de stage.

Enfin, je remercie tous les enseignants de l'École Nationale Supérieure d'Ingénieurs de Tunis qui m'ont enseigné durant les trois années passées, et qui m'ont donné la clé de la réussite.

# Table des matières

<b>Introduction générale .....</b>	<b>1</b>
<b>Chapitre 1 : Cadre général du projet .....</b>	<b>3</b>
1.    Cadre du stage .....	3
2.    Présentation de l'organisme d'accueil.....	3
2.1.    Le groupe Banque Finance Internationale « BFI » .....	3
2.2.    Organigramme de la société .....	4
3.    Présentation du projet .....	4
3.1.    Étude et critique de l'existant.....	5
3.2.    Problématique.....	5
3.3.    Solution proposée.....	6
4.    Méthodologie de travail.....	6
<b>Chapitre 2 : Phase de planification .....</b>	<b>9</b>
1.    Présentation de l'équipe Scrum .....	9
2.    Analyse des besoins.....	9
2.1.    Identification des acteurs.....	10
2.2.    Backlog du produit .....	10
2.3.    Spécification des besoins fonctionnels.....	13
2.4.    Spécification des besoins non fonctionnels.....	13
3.    Diagramme des cas d'utilisation global.....	14
4.    Spécification architecturale .....	14
4.1.    Architecture physique .....	14
4.2.    Architecture logique .....	15
5.    Environnement de travail.....	16
5.1.    Environnement matériel .....	16

5.2.	Environnement logiciel .....	17
6.	Planification des sprints.....	18
<b>Chapitre 3 : Release 1 « Gestion des utilisateurs, des offres &amp; des candidatures ».....</b>	<b>21</b>	
1.	Planification de la release 1 .....	21
2.	Sprint 1.1 : S'authentifier .....	22
2.1.	Mécanisme de sécurité .....	22
2.2.	Backlog du sprint 1.1 .....	23
2.3.	Analyse des besoins du sprint 1.1 .....	23
2.4.	Conception du sprint 1.1 .....	25
2.5.	Réalisation du sprint 1.1.....	27
3.	Sprint 1.2 : Gérer le profil .....	28
3.1.	Backlog du sprint 1.2 .....	29
3.2.	Analyse des besoins du sprint 1.2 .....	29
3.3.	Conception du sprint 1.2 .....	31
3.4.	Réalisation du sprint 1.2.....	33
4.	Sprint 1.3 : Gérer les offres et les candidatures.....	35
4.1.	Backlog du sprint du sprint 1.3 .....	36
4.2.	Analyse des besoins du sprint 1.3 .....	36
4.3.	Conception du sprint 1.3 .....	38
4.4.	Réalisation du sprint 1.3.....	40
<b>Chapitre 4 : Release 2 « Gestion des responsables, des candidats &amp; des entretiens ».....</b>	<b>43</b>	
1.	Planification de la release 2 .....	43
2.	Sprint 2.1 : Gérer les responsables .....	43
2.1.	Backlog du sprint 2.1 .....	43
2.2.	Analyse du sprint 2.1.....	44

2.3. Conception du sprint 2.1 .....	46
2.4. Réalisation du sprint 2.1.....	47
3. Sprint 2.2 : Gérer les candidats et les tests .....	49
3.1. Backlog du sprint 2.2 .....	49
3.2. Analyse des besoins du sprint 2.2 .....	49
3.3. Conception du sprint 2.2 .....	51
3.4. Réalisation du sprint 2.2.....	52
4. Sprint 2.3 : Gérer les entretiens .....	56
4.1. Backlog du sprint 2.3 .....	56
4.2. Analyse des besoins du sprint 2.3 .....	56
4.3. Conception du sprint 2.3 .....	57
4.4. Réalisation du sprint 2.3.....	59
<b>Chapitre 5 : Release 3 « Analyse des CV ».....</b>	<b>61</b>
1. Planification de la release 3 .....	61
2. Système d'analyse des CV .....	61
2.1. Découpage et segmentation en section d'un CV.....	62
2.2. Conversion et Extraction.....	64
2.3. Évaluation.....	65
2.4. Affichage et Tri .....	66
3. Sprint 3.1 : Analyser les CV par compétence.....	66
3.1. Backlog du sprint 3.1 .....	66
3.2. Analyse des besoins du sprint 3.1 .....	67
3.3. Conception du sprint 3.1 .....	68
3.4. Réalisation du sprint 3.1.....	70
4. Sprint 3.2 : Analyser les CV par niveau de la langue.....	78

4.1.	Backlog du sprint 3.2 .....	78
4.2.	Analyse des besoins du sprint 3.2 .....	79
4.3.	Conception du sprint 3.2 .....	80
4.4.	Réalisation du sprint 3.2.....	82
<b>Conclusion générale.....</b>	<b>88</b>	
<b>Netographie .....</b>	<b>89</b>	

# Table des figures

Figure 1:Clients et filiales de BFI dans le monde [2].....	3
Figure 2: Organigramme de la société .....	4
Figure 3: Scrum agile [4].....	8
Figure 4: Implémentation des sprints dans Taïga [5] .....	12
Figure 5: Analyse des CV avec Taïga .....	12
Figure 6: Diagramme de cas d'utilisation globale.....	14
Figure 7: Architecture Microservice[5].....	15
Figure 8: Mécanisme du JWT [20]. .....	23
Figure 9: Diagramme de cas d'utilisation du sprint 1.1.....	24
Figure 10: Diagramme de classe du sprint 1.1 .....	25
Figure 11: Diagramme de séquence « S'authentifier » .....	26
Figure 12: Interface de connexion.....	27
Figure 13: Les exigences de complexité d'authentification.....	27
Figure 14: Inexistence d'utilisateur pour l'authentification .....	28
Figure 15: Authentification des utilisateurs .....	28
Figure 16: Diagramme de cas d'utilisation « Gérer le profil » .....	29
Figure 17: Diagramme de classe du sprint 1.2 .....	31
Figure 18: Diagramme de séquence du sprint 1.2.....	33
Figure 19: Interface de création d'un compte.....	34
Figure 20: Interface de la création d'un profil.....	34
Figure 21: Dernière étape d'inscription.....	34
Figure 22: Affichage des coordonnées du profil candidat.....	35
Figure 23: Modification du profil candidat.....	35
Figure 24: Diagramme de cas d'utilisation du sprint 1.3.....	37
Figure 25: Diagramme de classe du sprint 1.3 .....	39
Figure 26: Diagramme de séquence du sprint « Gérer les candidatures et les offres » .....	40
Figure 27: Ajout des offres .....	41
Figure 28: Postulation des candidats aux offres.....	42
Figure 29: Diagramme de cas d'utilisation du sprint 2.1 .....	44
Figure 30: Diagramme de classe du sprint 2.1 .....	46
Figure 31: Diagramme de séquence du sprint 2.1.....	47
Figure 32: Liste des responsables tests .....	48

Figure 33: Ajout d'un responsable.....	48
Figure 34: Modification des responsables.....	49
Figure 35: Cas d'utilisation du sprint 2.2 .....	50
Figure 36: Diagramme de classe du sprint 2.2 .....	51
Figure 37: Diagramme de séquence du sprint 2.2.....	52
Figure 38: Suppression des candidats .....	53
Figure 39: Suppression des candidats affectée .....	53
Figure 40: Envoi automatique des mails .....	54
Figure 41: Première interface du test .....	54
Figure 42: Réponses des candidats au test .....	55
Figure 43: Niveau des candidats dans le test .....	55
Figure 44: Diagramme de cas d'utilisation du sprint 2.3.....	56
Figure 45: Diagramme de classe du sprint 2.3 .....	58
Figure 46: Diagramme de séquence du sprint 2.3.....	58
Figure 47: Ajout des dates d'entretien .....	59
Figure 48: Suppression des dates d'entretien .....	59
Figure 49: Modèle d'analyse des CV.....	62
Figure 50: Etape de découpage d'un CV.....	62
Figure 51: Etape de segmentation d'un CV .....	63
Figure 52: Analyse syntaxique des compétences .....	64
Figure 53: Etape de conversion d'un CV.....	64
Figure 54: Étapes du traitement NLP [22] .....	65
Figure 55: Etape d'évaluation d'un CV .....	65
Figure 56: Etape de tri des CV .....	66
Figure 57: Diagramme de cas d'utilisation du sprint 3.1.....	67
Figure 58: Diagramme de classe de sprint 3.1.....	69
Figure 59: Diagramme de séquence d'analyse des CV .....	70
Figure 60: Code d'analyse des CV par compétence (1) .....	71
Figure 61: Code d'analyse des CV par compétence (2) .....	72
Figure 62: Découpage d'un CV en sections .....	73
Figure 63: Résultat de l'exécution de la fonction de découpage .....	74
Figure 64: Code de segmentation d'un CV .....	74
Figure 65: Exécution de la segmentation d'un CV.....	75
Figure 66: Etape de conversion d'un CV.....	76
Figure 67: Etape d'extraction d'un CV .....	76

Figure 68: Exécution d'identification des compétences.....	77
Figure 69: Calcul du score des compétences d'un CV .....	77
Figure 70: Analyse des candidatures par compétences .....	78
Figure 71: Diagramme de cas d'utilisation du sprint 3.1.....	79
Figure 72: Diagramme du sprint 3.1.....	81
Figure 73: Diagramme du sprint 3.2.....	82
Figure 74: Code d'analyser les CV par niveau de la langue (1) .....	83
Figure 75: Code d'analyser les CV par niveau de la langue (2).....	84
Figure 76: Code de la comparaison des fautes.....	85
Figure 77: Exécution du code de comparaison des fautes .....	85
Figure 78: Code de déterminaison des fautes orthographiques .....	86
Figure 79: Liste des fautes orthographiques .....	86
Figure 80: Liste des candidatures analysée par niveau de la langue.....	87

# Liste des tableaux

Tableau 1: Comparaison entre les méthodologies de travail agile XP et Scrum.....	7
Tableau 2: Backlog du produit.....	11
Tableau 3: Les outils de développement.....	17
Tableau 4: Technologies de développement.....	18
Tableau 5: Planification des sprints.....	20
Tableau 6: Planification des sprints du release 1 .....	21
Tableau 7: Backlog du sprint 1.1 .....	23
Tableau 8: Description textuelle du cas d'utilisation : « S'authentifier ».....	24
Tableau 9: Backlog du sprint « Gérer le profil » .....	29
Tableau 10: Description textuelle du cas d'utilisation "Ajouter le profil" .....	30
Tableau 11: Description cas d'utilisation « Modifier le profil ».....	31
Tableau 12: Backlog du Sprint 1.3 .....	36
Tableau 13: Description cas d'utilisation « Ajouter une candidature à une offre » .....	38
Tableau 14: Description cas d'utilisation « Supprimer une offre » .....	38
Tableau 15: Planification du release 2.....	43
Tableau 16: Backlog du sprint 2.1 .....	44
Tableau 17: Description de cas d'utilisation « Ajouter responsables » .....	45
Tableau 18: Backlog du sprint 2.2 .....	49
Tableau 19: Description cas d'utilisation de suppression des candidats.....	50
Tableau 20: Backlog du sprint 2.3 .....	56
Tableau 21: Description cas d'utilisation Ajouter les dates.....	57
Tableau 22: Planification des sprints de release 3 .....	61
Tableau 23: Backlog du sprint 3.1 .....	66
Tableau 24: Description cas d'utilisation : Analyse des CV .....	68
Tableau 25: Backlog du sprint 3.2 .....	78
Tableau 26: Description cas d'utilisation : Analyser les CV par compétence .....	80

## Introduction générale

Il est depuis longtemps moins coûteux pour l'employeur de publier ses offres d'emploi, celui-ci les diffuse plus facilement et n'hésite plus à croiser plusieurs espaces de publication pour une même offre. En parallèle le candidat s'est vu doté, grâce à l'Internet, d'un outil facilitant considérablement son processus de candidature. La recherche d'emploi pouvant maintenant s'effectuer de manière passive : nous remarquons même qu'un grand nombre de demandeurs d'emploi sont déjà en poste.

À la recherche des nouveaux talents, l'entreprise reçoit plusieurs candidatures, de qualités variables. Il s'agit alors de ne pas perdre le cap et de réagir de façon adaptée. Une bonne gestion des candidatures et des tests est donc nécessaire : il s'agit d'un système clair nous permettant de gérer l'organisation des candidatures.

D'autre part, lors d'un processus de recrutement, le Curriculum Vitae « CV » reste l'outil principal pour juger un candidat. Savoir qualifier rapidement et surtout efficacement un CV permettra d'être plus pertinent également en entretien. Cette procédure clarifie les questions sur les informations importantes, celles qui sont écrites dans le CV.

Dans ce cadre s'intègre notre projet de fin d'études qui s'intitule au « Conception et développement d'une application de recrutement et d'analyse des Curriculum Vitae » effectué au sein de la société BFI sous forme de stage de cinq mois, et dont la mission consiste à concevoir et mettre en place une solution web donnant une gestion des candidatures et des tests de recrutement d'une façon complète et avancée.

Cette application a pour but de simplifier les processus de recrutement dans le but d'accélérer le taux d'acquisition de nouveaux adhérents.

Notamment, le présent rapport est la synthèse des étapes de mise en œuvre de cette application. Nous pourrons présenter les détails du ce rapport qui s'intitule en cinq chapitres.

- Le premier chapitre présente la société d'accueil, le contexte général de notre projet, la description de la problématique ainsi le choix de la méthodologie de travail.
- Le second chapitre est consacré pour la phase d'analyse et planification des besoins de notre application.

- Le troisième chapitre introduit la première release intitulé « Gestion des utilisateurs, des offres et des candidatures » qui sera consacrée à la présentation des trois premiers sprints « Gérer les utilisateurs », « Gérer les offres » et « Gérer les candidatures » en passant par les phases de l’analyse, la conception et la réalisation.
- Le quatrième chapitre introduit la deuxième release intitulée « Gérer les responsables et les candidats » qui inclut les trois prochains sprints « Gérer les responsables », « Gérer les candidats et les test » et « Gérer les entretiens » en présentant les différentes phases de mise en place du travail qui comportent l’analyse, la conception détaillée ainsi que les interfaces réalisées.
- Le cinquième chapitre intitulée « Analyse des CV » sera consacré au dernier release de notre incrément final de l’application et portera sur la présentation des deux derniers sprints « Analyser les CV par compétence » et « Analyser les CV par niveau de la langue ». Nous passons également par la description de l’analyse contextuelle, la conception et enfin la partie réalisation.

Nous clôturerons ce rapport par une conclusion générale sur le projet réalisé en mettant l’accent sur les principales perspectives et les éventuelles améliorations.

# Chapitre 1 : Cadre général du projet

## Introduction

Pour commencer, nous exposerons dans ce chapitre le contexte de notre projet. Dans un premier temps, nous présenterons l'environnement du stage. Par la suite, nous décrirons le projet qui nous a été confié.

### 1. Cadre du stage

Ce stage est dans le cadre d'un projet de fin d'études pour obtenir le diplôme d'ingénieur en informatique de l'École Nationale Supérieure d'Ingénieurs de Tunis (ENSIT). Notre stage a été effectué au sein de la société BFI « Banque Finance Internationale ».

Le sujet est intitulé « Conception et développement d'une application de recrutement et d'analyse des Curriculum vitae ».

### 2. Présentation de l'organisme d'accueil

Dans cette partie, nous allons présenter la société dans laquelle nous avons réalisé le projet tout en introduisant ses services, son marché et son organigramme.

#### 2.1. Le groupe Banque Finance Internationale « BFI »

La BFI est une société tunisienne spécialisée dans l'édition et l'intégration de solutions logicielles destinées aux banques et institutions financières créée en 1994.

Cette entreprise se repose essentiellement sur le marché africain de la gestion bancaire dont elle occupe une place importante avec une réputation de très haut niveau en possédant environ 200 institutions financières dans 25 pays à travers le continent africain [1].

La figure 1 illustre les clients et les filiales de BFI dans le monde entier.



Figure 1:Clients et filiales de BFI dans le monde [2]

## 2.2. Organigramme de la société

Notamment, la BFI propose une gamme de solutions couvrant les différents domaines de l'activité bancaire tels que les systèmes front / back-office, la gestion des clients, la gestion des risques, le contrôle des engagements, le crédit, la trésorerie, les systèmes de paiement et la comptabilité comme l'indique la figure 2.

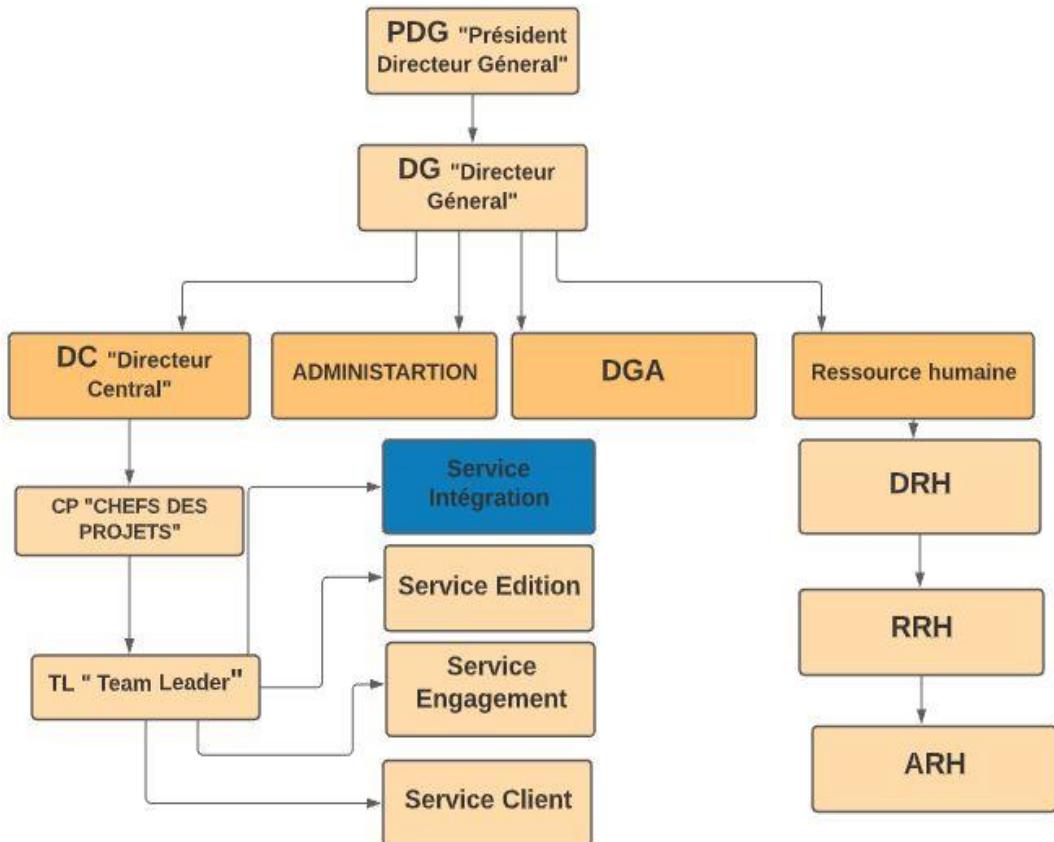


Figure 2: Organigramme de la société

Tout du long ce stage, le déroulement du mon projet est effectué dans le service d'intégration pour réaliser les besoins du département ressource humaine.

## 3. Présentation du projet

Dans le cadre du présent travail, nous nous sommes chargés de développer une application web qui permet d'analyser les CV et gérer les candidatures et de créer des tests d'évaluation pour évaluer les candidatures adéquates aux offres de recrutement publiées. Pour identifier ces tâches, nous commençons par présenter l'étude et critique de l'existant, la problématique et la solution proposée.

### **3.1. Étude et critique de l'existant**

La gestion des candidatures au sein de l'entreprise « BFI » se déroule actuellement de la manière suivante :

- Les données des candidatures sont traitées sous format papier et ou numérique.
- Les données issues de ces dossiers sont saisies dans un tableau Microsoft Excel et un premier avis est donné par le responsable qui examine l'intégralité des dossiers.
- Les meilleurs candidats sont sélectionnés parmi le flux des CV reçus.
- Des entretiens sont planifiés avec les candidats qui ont été choisis et qui vont être informés par un email.

La gestion des ressources humaines consulte éventuellement des agences de recrutement ou à travers les médias traditionnels. Le recrutement à travers cette méthode est loin de satisfaire objectivement les besoins des employeurs et des employés.

En outre, des solutions indépendantes sont utilisées afin de satisfaire les besoins du personnel, mais ces outils ne sont pas toujours efficaces du fait qu'ils sont divisés sur plus qu'un service indépendant et contenant une grande partie traitée manuellement non centralisée et non partagée à l'accès du système informatique.

Par conséquent, les employés qualifiés n'étant pas toujours informés sur les offres disponibles sur le marché du travail.

### **3.2. Problématique**

Notre problématique consiste à proposer et développer un processus de recrutement et d'analyse des CV selon les compétences techniques et linguistiques permettant de trouver les candidats appropriés aux offres proposées.

L'outil développé doit permettre de :

- Centraliser l'information et les données concernant le recrutement.
- Diminuer le temps de saisie des candidatures à travers l'analyse des offres selon le besoin.
- Diminuer le temps de suivi d'une candidature grâce aux sélections des compétences par l'analyse des curriculums vitae.
- Gérer et partager plus facilement l'information des candidatures avec les responsables techniques et tests pour le bien déroulement des tests et des entretiens.

### **3.3. Solution proposée**

En tenant compte de la problématique citée ci-dessus, une solution proposée sera donc de concevoir et développer une application web d'analyse des CV, afin de gérer aussi bien la postulation aux offres des emplois à travers le développement d'un modèle intelligent qui analyse les CV des candidats pour la sélection des meilleures candidatures de chaque offre.

Cette application doit respecter les conditions suivantes :

- Gestion des utilisateurs et des candidatures
- Gestion des responsables et des candidats.
- Gestion des entretiens et des tests.
- Analyse des CV par compétence et par niveau de la langue.

## **4. Méthodologie de travail**

Le choix de la méthodologie du travail est une phase importante pour réussir un projet. Ce choix se diffère d'un projet à un autre suivant des critères tels que la nature et la taille du projet. L'objectif de la gestion de projet est d'augmenter le niveau essentiellement de satisfaction de clients tout en rendant le travail de développement plus facile. Il existe principalement deux méthodes pour la gestion de projet :

- Les méthodes classiques : ces méthodes sont prédictibles, qui se caractérisent par définir les besoins de début, planification rigide et documentation lourde tels que le cycle en V et en Cascade. Ces méthodes se caractérisent par l'absence de communication avec le client et par la difficulté de prendre en considération les nouveaux besoins lorsque le projet est en cours.
- Les méthodes agiles : à l'opposé des méthodes classiques, les méthodes agiles sont adaptatives, qui s'orientent vers les personnes plutôt que les processus et les outils tels que le SCRUM et le XP.

Ces méthodes se caractérisent par la collaboration avec le client, le contrôle des coûts, meilleure visibilité puisque le client sait très bien ses besoins et la priorité des tâches à réaliser et ils assurent l'acceptation de changement plutôt qu'un contrat blindé.

Une étude théorique comparative des plus populaires méthodes en usage aujourd'hui est menée pour choisir celle la plus appropriée pour mener à bien la réalisation du projet. Le tableau 1 illustre cette comparaison.

Méthodologie	Scrum	Extreme Programming (XP)
<b>Planification</b>	Au début de chaque sprint	Le suivi du projet se fait au fur et à mesure de l'avancement des itérations
<b>Conseil</b>	Product backlog, Scrum board, burndown	Priorisation par le client, pratiques XP
<b>Quand choisir ?</b>	Équipe dédiée à 100% au projet	Amélioration de la qualité du logiciel critique, prise en compte immédiate des changements
<b>Caractéristiques principales</b>	Vélocité Equipe Team Leader	<ul style="list-style-type: none"> <li>• Qualité code</li> <li>• Outilage</li> </ul>
<b>Top 3 bénéfices</b>	Productivité Scalabilité Engagement des équipes	<ul style="list-style-type: none"> <li>• Qualité du code plus importante</li> <li>• Réactivité</li> <li>• Niveau d'expertise des équipes</li> </ul>

Tableau 1: Comparaison entre les méthodologies de travail agile XP et Scrum.

Nous avons choisi SCRUM comme méthodologie de travail pour notre projet, vu qu'elle nous permet de :

- S'adapter aux changements dans les sprints grâce à des itérations courtes.
- Avoir un produit final qui correspond au plus près aux besoins du client.

Cette méthodologie consiste tout d'abord à définir les exigences fonctionnelles du produit qui sont les besoins priorisés par le client et qui sont définis dans le Backlog Produit. Ensuite, l'estimation des tâches en temps et en complexité par l'équipe de développement par suite l'estimation de charge.

Les fonctionnalités sont ordonnées et regroupées par complexité dans le Backlog Sprint. Le développement des tâches du Backlog Sprint se fait par itération, chaque itération à une durée fixe de 2 à 4 semaines qui s'appelle sprint. Après chaque sprint, il faut livrer un produit partiel. Cette méthodologie, consiste à faire une mêlée quotidienne de 15 minutes pour suivre la progression des tâches. Une mêlée après chaque sprint « revue de sprint » où l'équipe effectue une démo de livrable de ce Sprint. A la fin de chaque sprint, l'équipe fait une mêlée de rétrospective pour avoir ce qui marche et ce qui doit corriger [3].

La figure 3 résume le principe du scrum agile.

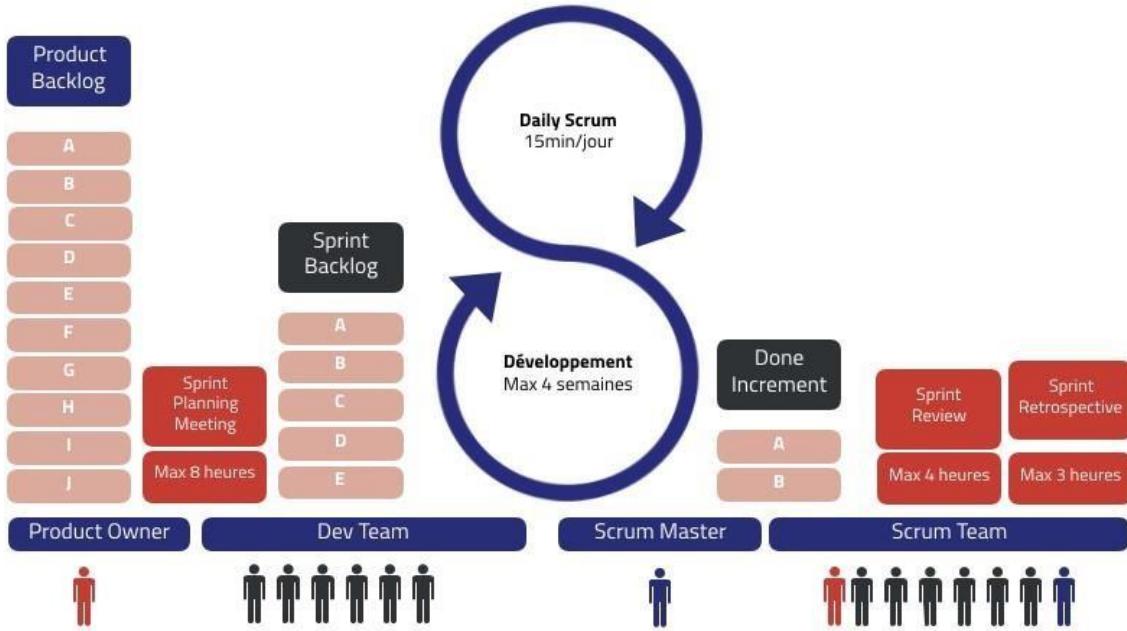


Figure 3: Scrum agile [4]

## Conclusion

Dans ce chapitre, nous avons présenté le contexte général de notre projet. Tout d'abord, nous avons décrit l'étude de l'existant, les limites et la solution envisagée. Ensuite, nous avons passé par la méthodologie de travail à adopter pour mener à bien notre projet.

Et enfin, nous avons exposé l'environnement de travail approuvé tout en spécifiant les technologies utilisées.

Dans le chapitre suivant, nous allons analyser nos besoins et planifier les sprints à réaliser.

# **Chapitre 2 : Phase de planification**

## **Introduction**

Ce chapitre est consacré en premier lieu l'équipe Scrum. Nous exposons en deuxième lieu les besoins fonctionnels et non fonctionnels de notre application. Ensuite nous identifions les acteurs et le cas d'utilisation global du système. Par la suite, nous nous intéressons à la présentation de l'environnement de travail et en introduisant le backlog produit et en planifiant les sprints tout en présentant l'outil de gestion de projet utilisé.

### **1. Présentation de l'équipe Scrum**

Dans le contexte de notre projet, l'équipe Scrum est composée par :

- Le propriétaire du produit : son rôle consiste à présenter et à définir la priorité des fonctionnalités de produit à développer. Il est le responsable de valider la livraison avec l'équipe de développement à la fin de chaque sprint. Dans notre cas, le propriétaire du produit est MM. TEBAI Dorra.
- Le Scrum master : son rôle consiste à organiser les réunions Scrum et à assurer la bonne pratique de la méthodologie Scrum. Il suit la démarche et l'avancement de travail. Dans notre cas, le Scrum master est M. BACHA Skandar.
- L'équipe de développement : peut comporter une ou plusieurs personnes. Le rôle de cette équipe est de livrer régulièrement un produit fonctionnel (livraison). Cette équipe est autoorganisée et elle est censée faire la conception, l'implémentation et le test. Dans notre cas l'équipe est composée par moi-même.

### **2. Analyse des besoins**

Dans tout système, les fonctionnalités doivent être mises en relation avec un ensemble de besoins. Ces besoins définissent les fonctions que les utilisateurs s'attendent à voir par le système. Afin d'avoir une bonne compréhension des besoins attendus de notre système, dans cette partie, nous présentons la spécification et l'analyse des besoins auxquels notre application doit répondre.

D'abord, nous commençons par identifier les acteurs. Ensuite, nous décrivons les besoins fonctionnels et non fonctionnels en présentant le Backlog du produit.

## 2.1. Identification des acteurs

Avant d'entamer la présentation des diagrammes, il faut identifier les acteurs qui sont en interaction avec l'application. Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, il y'a eu dans notre projet d'identifier deux acteurs principaux :

- Les candidats qui sont à la recherche d'un emploi.
- La ressource humaine a pour rôle de recevoir les candidatures, gérer les offres d'emploi, évaluer les candidats, gérer le calendrier pour choisir les candidatures et fixer les entretiens.

## 2.2. Backlog du produit

Nous entreprenons, dans ce paragraphe, à la description des fonctionnalités à développer sous forme des récits utilisateurs (user-stories) dans le backlog du produit.

Le backlog de notre application est illustré par le tableau 2.

ID	User story	Titre	Priorité
1	En tant qu'un candidat, je peux me connecter à l'espace spécifique.	S'authentifier	Élevée
2	En tant qu'un responsable des ressources humaines, je peux me connecter à l'espace RH.		Élevée
3	En tant qu'un candidat, je peux s'inscrire et ajouter un profil.	Gérer le profil	Élevée
4	En tant qu'un candidat, je peux modifier le profil.		Moyenne
5	En tant qu'un responsable des ressources humaines, je peux ajouter des offres.	Gérer les offres et les candidatures	Élevée
6	En tant qu'un utilisateur connecté (RH / candidat), je peux afficher les offres.		Moyenne
7	En tant qu'un candidat, je peux afficher les candidatures d'une offre		Élevée
8	En tant qu'un utilisateur connecté (RH / candidat), je peux afficher les détails de chaque offre.		Élevée

<b>9</b>	En tant qu'un responsable des ressources humaines, je peux supprimer une offre.		Élevée
<b>10</b>	En tant qu'un candidat, je peux ajouter une candidature à une offre.		Élevée
<b>11</b>	En tant qu'un responsable des ressources humaines, je peux afficher les candidats des candidatures.		Élevée
<b>12</b>	En tant qu'un responsable des ressources humaines, je peux ajouter les responsables techniques et de tests.	<b>Gérer les responsables</b>	Moyenne
<b>13</b>	En tant qu'un responsable des ressources humaines, je peux modifier les responsables techniques et de tests.		Moyenne
<b>14</b>	En tant qu'un responsable des ressources humaines, je peux supprimer les responsables techniques et de tests.		Moyenne
<b>15</b>	En tant qu'un responsable des ressources humaines, je peux choisir des candidats et les contacter par mail.	<b>Gérer les candidats et les tests</b>	Élevée
<b>16</b>	En tant qu'un responsable des ressources humaines, je peux supprimer des candidats.		Élevée
<b>17</b>	En tant qu'un candidat, je peux réaliser des tests.		Élevée
<b>18</b>	En tant qu'un responsable des ressources humaines, je peux ajouter des dates d'entretiens dans le calendrier.	<b>Gérer les entretiens</b>	Moyenne
<b>19</b>	En tant qu'un responsable des ressources humaines, je peux supprimer les dates d'entretiens.		Moyenne
<b>20</b>	En tant qu'un responsable des ressources humaines, je peux modifier les dates d'entretiens.		Moyenne
<b>21</b>	En tant qu'un responsable des ressources humaines, je peux analyser les CV par compétence.	<b>Analyser les CV</b>	Élevé
<b>22</b>	En tant qu'un responsable des ressources humaines, je peux analyser les CV par niveau de la langue.		Élevé
<b>23</b>	En tant qu'un responsable des ressources humaines, je peux évaluer les candidatures.		Élevé

Tableau 2: Backlog du produit

Les méthodes agiles vont dans le sens d'un rééquilibrage, en faisant passer l'outil après la maîtrise des pratiques. Notamment dans notre planification, nous avons choisi « Taïga » pour la gestion de notre projet.

La figure 4 illustre les interfaces de l'outil Taïga en présentant les tâches à faire pour chaque sprint.

The screenshot shows the Taïga interface with three visible sprint taskboards:

- SPRINT TASKBOARD** (Top): Gestion des responsables (10 May 2021-12 May 2021). Status: 0 closed / 0 open.
- SPRINT TASKBOARD** (Middle): Analyse des CVs (03 Apr 2021-10 May 2021). Status: 0 closed / 0 open.
- SPRINT TASKBOARD** (Bottom): Gestion des candidatures et des candidats (20 Mar 2021-03 Apr 2021). Status: 0 closed / 0 open.

Below the taskboards is a list of activity logs:

- nawras haloui has updated the attribute "Status" of the task #11 Inscription des employés(Frontend) which belongs to the user story #3 Inscription et gestion d'accès to Closed a month ago
- nawras haloui has updated the attribute "Status" of the task #9 Inscription des candidat(Frontend) which belongs to the user story #3 Inscription et gestion d'accès to Closed a month ago
- nawras haloui has updated the attribute "Status" of the task #10 Inscription des employés(Backend) which belongs to the user story #3 Inscription et gestion d'accès to Closed a month ago
- nawras haloui has updated the attribute "Status" of the task #8 Inscription des candidats(partie backend) which belongs to the user story #3 Inscription et gestion d'accès to Closed a month ago
- nawras haloui has updated the attribute "Status" of the task #8 Inscription des candidats(partie backend) which belongs to the user story #3 Inscription et gestion d'accès to Ready for test a month ago
- nawras haloui has updated the attribute "Status" of the task #11 Inscription des employés(Frontend) which belongs to the user story #3 Inscription et gestion d'accès to Ready for test a month ago
- nawras haloui has updated the attribute "Status" of the task #10 Inscription des employés(Backend) which belongs to the user story #3 Inscription et gestion d'accès to Ready for test a month ago
- nawras haloui has updated the attribute "Status" of the task #9 Inscription des candidat(Frontend) which belongs to the user story #3 Inscription et gestion d'accès to Ready for test a month ago
- nawras haloui has created a new task #11 Inscription des employés(Frontend) in Stage PFE which belongs to the user story #3 Inscription et gestion d'accès a month ago
- nawras haloui has created a new task #10 Inscription des employés(Backend) in Stage PFE which belongs to the user story #3 Inscription et gestion d'accès a month ago

Figure 4: Implémentation des sprints dans Taïga [5]

Tenons compte du nombre important des sprints, nous allons étudier l'implémentation du sprint d'analyse des CV avec Taïga comme il est indiqué dans la figure 5.

USER STORY	NEW	IN PROGRESS	READY FOR TEST	CLOSED
#3 Détection du niveau des compétences des candidats 13 pts	+ NEW			
				#4 Détection les candidats les plus performants par compétences Not assigned #8 Détection des fautes dans la langues des candidats Not assigned

Figure 5: Analyse des CV avec Taïga

## **2.3.Spécification des besoins fonctionnels**

Les besoins fonctionnels sont les besoins spécifiant un comportement d'entrée / sortie du système. En fonction des exigences fonctionnelles, un ingénieur détermine le comportement (sortie) qu'un logiciel est censé présenter dans le cas d'une entrée donnée. Nos acteurs doivent satisfaire leurs besoins fonctionnels qui se résument dans les points suivants.

Le candidat et le responsable des ressources humaines partagent les mêmes fonctionnalités suivantes :

- S'authentifier.
- Gérer les candidatures et les offres.

Le candidat à la possibilité de :

- Gérer son profil.

Le responsable des ressources humaines peut faire les fonctionnalités suivantes :

- Gérer les responsables.
- Gérer les entretiens.
- Analyser les CV.
- Gérer les candidats et les tests.

## **2.4.Spécification des besoins non fonctionnels**

Nous allons déterminer l'ensemble de contraintes d'implémentation à respecter pour garantir le bon fonctionnement de l'application :

- La sécurité : C'est un besoin très important à assurer lorsqu'on parle de banques, pour cela chaque opération qui semble avoir un risque potentiel il doit y avoir une autre validation d'identité par un code secret.
- La rapidité : L'application doit être conçue pour avoir un temps de réponse minimum.
- La fiabilité : L'application doit avoir une bonne qualité de contenu ainsi qu'une bonne adaptabilité aux différentes tailles des écrans des appareils mobiles. D'autre part il faut assurer le bon fonctionnement sans erreur.
- L'ergonomie : L'application doit être simple à maintenir et à comprendre.
- La disponibilité : Le réseau doit être disponible en permanence pour permettre à chaque utilisateur d'accéder aux services disponibles à n'importe quel moment.

### 3. Diagramme des cas d'utilisation global

Le diagramme des cas d'utilisation (Use Case Diagram) constitue la première étape de l'analyse UML en modélisant les besoins des utilisateurs, identifiant les grandes fonctionnalités et les limites du système et représentant les interactions entre le système et ses utilisateurs.

La figure 6 illustre le diagramme des cas d'utilisation global de notre application.

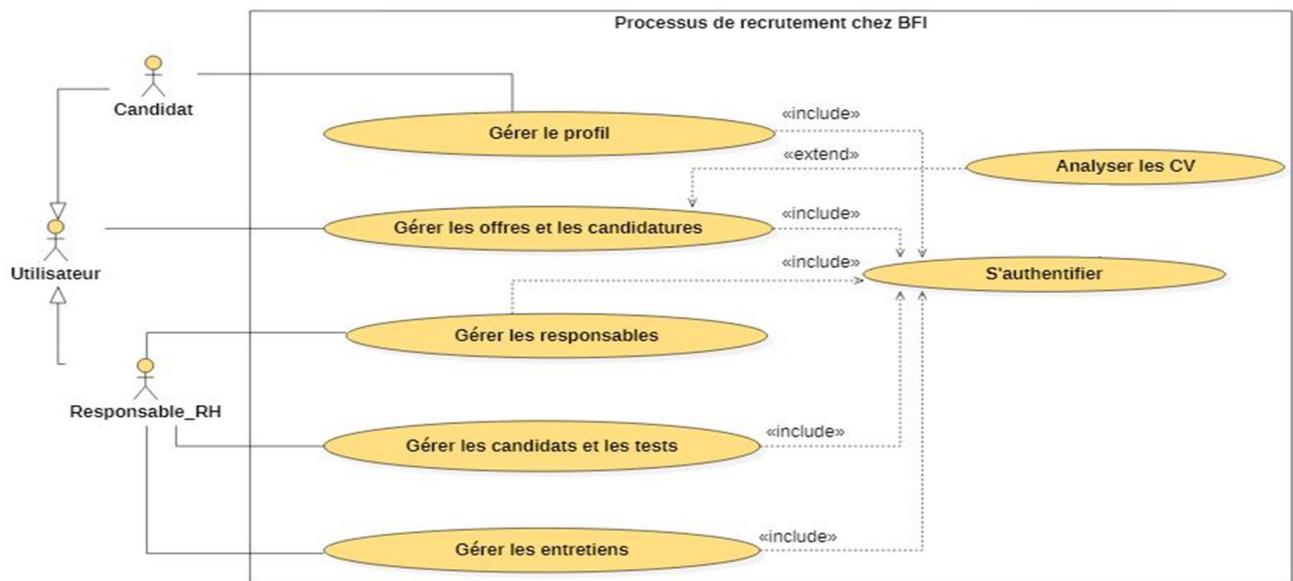


Figure 6: Diagramme de cas d'utilisation globale

### 4. Spécification architecturale

Dans ce stade, nous présentons le modèle architectural de notre solution. Pour ce faire, nous entamons en premier lieu, l'aspect physique à travers l'architecture opérationnelle. En deuxième lieu, nous présentons l'aspect logique à travers l'architecture applicative.

#### 4.1. Architecture physique

La conception architecturale permet d'identifier les sous-systèmes ainsi que les différents nœuds. Notre application sera développée dans une architecture microservice.

Autrement dit, plusieurs services indépendamment déployable communiquent entre eux. Cette architecture est présentée dans la figure 7.

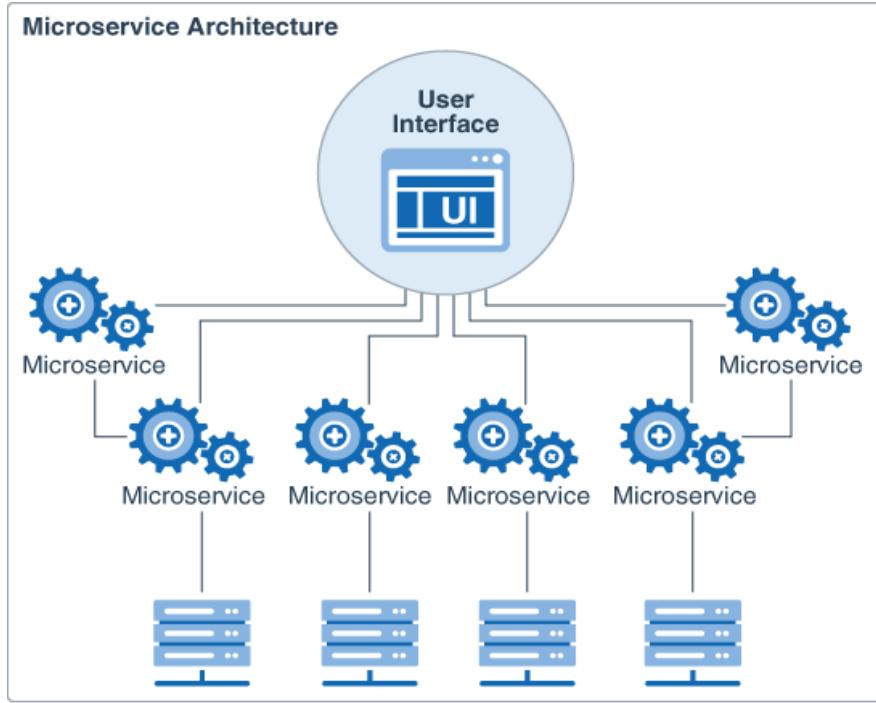


Figure 7: Architecture Microservice[5]

Cette architecture présente plusieurs avantages :

- L'hétérogénéité technologique.
- La résistance contre l'échec.
- La scalabilité sur mesure.
- La facilité de communication entre les services.
- L'alignement organisationnel.
- La réutilisabilité.

## 4.2. Architecture logique

Dans cette partie, nous présentons l'architecture applicative de notre solution. En fait, l'architecture mise en place pour notre application doit présenter un couplage faible entre les différents composants, faciliter l'interaction homme-machine et garantir l'évolutivité et la maintenance. La Figure 9 illustre une représentation de l'architecture logicielle de notre application qui est basée sur le modèle MVC (Model View Controller).

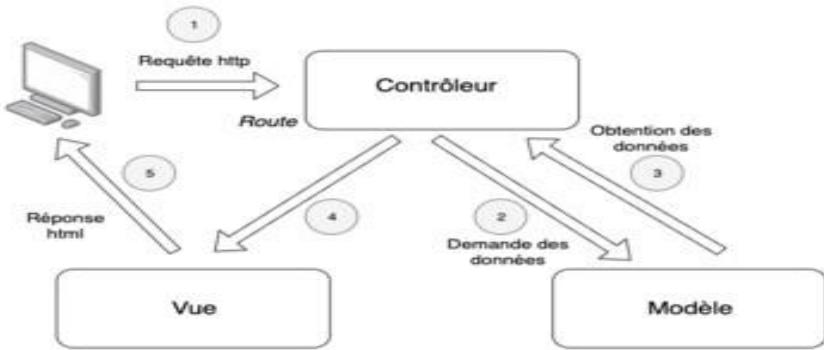


Figure 9: Architecture logique [6]

Le modèle MVC, est une technique de développement devenue un design pattern qui découpe l'application en trois couches principales [7]. La distinction de ces couches :

- La partie Modèle : contient les données de l'application et la logique métier. Par exemple, les comptes des utilisateurs, les produits que vous vendez, un ensemble de photos, etc. La composante modèle n'a aucune connaissance de l'interface graphique.
- La partie Vue : contient tout ce qui est visible à l'écran et qui propose une interaction avec l'utilisateur. Par exemple, les boutons, les images, les zones de saisie, etc.
- La partie Contrôleur : c'est la "colle" entre la vue et le modèle, qui gère également la logique de l'application. Le contrôleur permet de réagir aux interactions de l'utilisateur et de lui présenter les données qu'il demande.

## 5. Environnement de travail

Dans cette section, nous présentons l'ensemble des équipements utilisés ainsi que les technologies adoptées pour mener à bien la réalisation de notre projet.

### 5.1. Environnement matériel

Afin de bien mener notre travail, nous avons utilisé le PC de l'entreprise, dont les caractéristiques sont le suivant :

- Processseur Intel® Core™ i5.

- 12 Go de mémoire vive.
- Disque dur de capacité 1 To.
- Système d’exploitation Microsoft Windows 10.

## 5.2. Environnement logiciel

Pour développer notre application, nous avons utilisé plusieurs outils de développement et de modélisation comme les montre le tableau 3 suivant.

	IntelliJ : est un environnement de développement intégré (IDE). Il offre une rapidité et intelligence en proposant des suggestions adaptées dans tous les contextes telle la saisie automatique de code [8].
	Postman : est un environnement de développement d’API qui aide les utilisateurs à créer, tester, documenter, surveiller et publier la documentation de leurs API [9].
	Oracle SQL Developer est un environnement de développement intégré gratuit qui simplifie le développement et la gestion d’Oracle Database dans les traditionnels et dans le Cloud [10].
	StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux [11].
	Gitlab : est une plateforme qui permet d’optimiser la gestion des codes sources ou encore des dépôts Git. Elle permet également d’héberger l’ensemble de projets web, profiter d’une gestion optimale des versions des codes sources ou encore profiter d’un système performant d’intégration continue [12].

Tableau 3: Les outils de développement

Comme technologies, nous avons adopté des frameworks illustrés dans le tableau 4 suivant.

	Angular : est un Framework client open source basé sur TypeScript. Pour aider à développer un projet plus rapidement, Angular est particulièrement apprécié pour son outil CLI très puissant, qui le rend scalable et permet de générer des squelettes d’applications, des classes, services et composants depuis son propre terminal [13].
---	---

	Spring boot : est un Framework libre et open source pour le développement des applications d'entreprise. Il est fondé pour structurer, améliorer et simplifier la configuration des applications grâce à l'injection de dépendances. Les applications Spring Boot sont une avancée majeure dans le développement Java et l'utilisation du framework Spring [14].
	Hibernate : est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Il permet donc de représenter une base de données en objets Java et vice versa. [15]
	JWT : est une structure de RFC 7519 qui permet de transmettre les données entre les différentes parties de manière sécurisé sous le format JSON. Il est utilisé pour l'authentification ou l'échange des données [16].
	Spring Security : est un framework qui se concentre sur la fourniture à la fois d'authentification et d'autorisation aux applications Java. En fait, il présente un cadre d'authentification et de contrôle d'accès puissant et hautement personnalisable [17].
	Python est un langage de programmation open source utilisé dans le domaine de machine Learning, Big Data... Il s'agit d'un langage de programmation interprété, qui ne nécessite donc pas d'être compilé pour fonctionner [18].

Tableau 4: Technologies de développement

## 6. Planification des sprints

Après la définition du notre backlog, nous passons à la planification des sprints, une phase assez importante dans le déroulement du projet assurant une meilleure visibilité sur son évolution ainsi qu'un respect des échéances.

Selon la priorité des fonctionnalités ainsi que de leurs complexités, nous sommes arrivés à faire la planification décrite ci-dessous dans le tableau 5.

<b>Release 1 : Gestion des utilisateurs, des offres et des candidatures</b>	Sprint 1.1 : S'authentifier	S'authentifier pour des candidats.
		S'authentifier pour les responsables des ressources humaines.
	Sprint 1.2 : Gérer profil	Ajouter le profil.
		Modifier le profil.
		Afficher le profil.
	Sprint 1.3 : Gérer les offres et les candidatures.	Ajouter les offres.
		Afficher les offres.
		Afficher les candidatures d'une offre.
		Afficher les détails d'une offre.
		Supprimer une offre.
		Ajouter une candidature à une offre.
		Afficher les candidats des candidatures.
	Sprint 2.1 : Gérer les responsables	Ajouter les responsables techniques et de tests.
		Supprimer les responsables techniques et de tests.
		Modifier les coordonnées des responsables techniques et de test.
<b>Release2 : Gestion des responsables, des candidats et des entretiens.</b>	Sprint 2.2 : Gérer les candidats et les tests.	Envoyer des mails automatiques pour le processus de recrutement.
		Supprimer des candidats.
		Réaliser des tests.
	Sprint 2.3 : Gérer les entretiens	Ajouter un entretien au calendrier.
		Supprimer une date du calendrier.
		Modifier une date du calendrier.
<b>Release 3 : Analyse des CV</b>	Sprint 3.1 : Analyser des CV par compétences.	Sélectionner une offre.
		Sélectionner une compétence.
		Trier les candidatures triées par compétence.
		Calculer score pour chaque candidature.
		Extraire les données d'un CV.

	Sélectionner une offre.
	Trier les candidatures triées par compétence.
	Calculer score pour chaque candidature.
	Extraire les données d'un CV.

*Tableau 5: Planification des sprints*

## Conclusion

Dans ce chapitre, nous avons présenté l'équipe Scrum, ensuite nous avons passé par les besoins fonctionnels et non fonctionnels de notre application en élaborant le backlog du produit. D'autre part, nous avons exposé l'architecture globale de notre projet en planifiant les différents sprints. Dans le chapitre qui suit, nous allons étudier la première release.

# Chapitre 3 : Release 1 « Gestion des utilisateurs, des offres & des candidatures »

## Introduction

Ce chapitre consiste dans sa première partie à modéliser la première release pour l'espace du candidat en présentant la phase de planification de notre travail réalisé. Dans la deuxième partie, nous déterminons les différents sprints de notre release tout en spécifiant les étapes de chaque sprint : backlog du sprint, l'analyse des besoins, la conception au moyen des différents diagrammes et à la fin la phase de réalisation via les différentes interfaces.

## 1. Planification de la release 1

Notre première release est divisée en trois sprints comme l'illustre le tableau 6. Partant des estimations de chaque user story ainsi que la vitesse de l'équipe de développement, nous attribuons les fonctionnalités à développer dans le backlog du produit de chaque sprint.

Sprint 1.1 : S'authentifier	Sprint 1.2 : Gérer le profil	Sprint 1.3: Gérer les offres et les candidatures
<ul style="list-style-type: none"><li>Authentification des candidats.</li><li>Authentification de l'RH.</li></ul> <p>(De 20/02/2021 à 24/02/2021)</p>	<ul style="list-style-type: none"><li>Ajouter un profil</li><li>Modifier des coordonnées dans le profil.</li><li>Supprimer des coordonnées dans le profil.</li></ul> <p>(De 25/02/2021 à 02/03/2021)</p>	<ul style="list-style-type: none"><li>Ajouter des offres.</li><li>Afficher les offres.</li><li>Afficher les candidatures d'une offre.</li><li>Supprimer une offre.</li><li>Ajouter une candidature à une offre.</li><li>Afficher les candidats des candidatures.</li><li>Afficher les détails d'une offre.</li></ul> <p>(De 03/03/2021 à 16/03/2021)</p>

Tableau 6: Planification des sprints du release 1

## **2. Sprint 1.1 : S'authentifier**

Dans cette partie nous exposons une présentation détaillée du sprint avec ses spécificités lors de ce premier sprint. Nous commençons par le backlog du sprint suivi par l'analyse et le diagramme des cas d'utilisation global. Enfin nous clôturons par la partie réalisation dans laquelle nous présentons l'ensemble des interfaces développées.

### **2.1. Mécanisme de sécurité**

La sécurité d'un système doit être pensée avant même son lancement. Le choix de l'architecture et des moyens pour le sécuriser doivent être tout autant pris en compte que le service qu'offrira l'application.

Néanmoins, nous allons préciser dans cette partie l'outil JWT que nous avons adapté dans notre application.

#### **2.1.1. JSON Web Token (JWT)**

Les « JSON Web Token » ou JWT sont des jetons générés par un serveur lors de l'authentification d'un utilisateur sur une application Web, et qui sont ensuite transmis au client. Ils seront renvoyés avec chaque requête HTTP au serveur, ce qui lui permettra d'identifier l'utilisateur [19].

#### **2.1.2. Mécanisme basé sur deux jetons JWT**

Lorsque l'utilisateur se connecte à l'aide des ses informations d'identification, deux jetons Web JSON (access-token, refresh-token) sont renvoyés et doivent être enregistrés dans le stockage local.

Pour ce faire, les informations contenues dans le jeton sont signées à l'aide d'une clé privée détenue par le serveur. Quand il recevra à nouveau le jeton, le serveur n'aura qu'à comparer la signature envoyée par le client et celle qu'il aura générée avec sa propre clé privée et à comparer les résultats. Si les signatures sont identiques, le jeton est valide [20].

La figure 8 explique les étapes du mécanisme JWT.

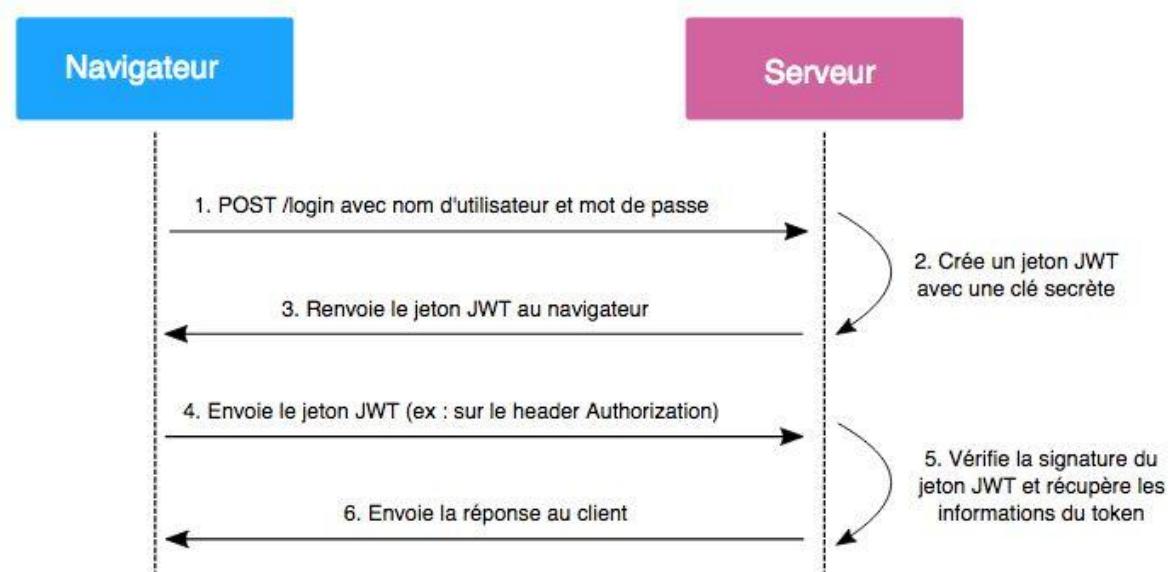


Figure 8: Mécanisme du JWT [20].

## 2.2.Backlog du sprint 1.1

Dans le tableau 7, nous exposons le backlog du sprint 1.1 contenant les user stories. Nous énumérons les différents user stories que nous développons dans la partie suivante.

ID Module	Module	Id user Story	User story
1	S'authentifier	1.1	En tant qu'un candidat, je veux me connecter à l'espace des candidats.
		1.2	En tant qu'un responsable des ressources humaines, je veux me connecter à l'espace des ressources humaines.

Tableau 7: Backlog du sprint 1.1

## 2.3.Analyse des besoins du sprint 1.1

Dans cette partie, nous allons spécifier les différentes spécifications et fonctionnalités des user stories mentionnées précédemment.

### 2.3.1.Diagramme des cas d'utilisation du sprint 1.1

La figure 9 illustre le diagramme des cas d'utilisation du sprint Authentification où nous citons les différentes fonctionnalités pour sprint « S'authentifier ».

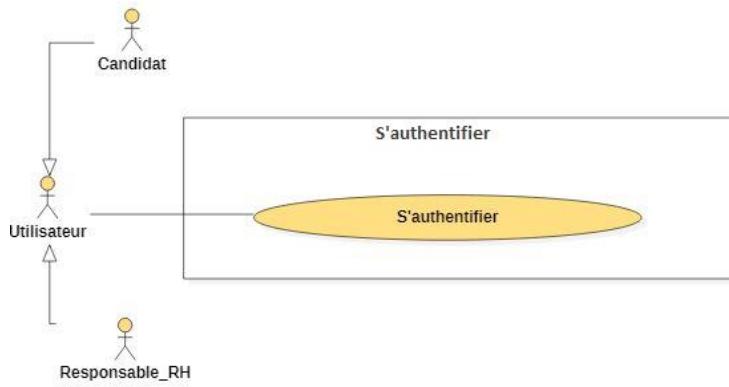


Figure 9: Diagramme de cas d'utilisation du sprint 1.1

### 2.3.2. Raffinements du sprint 1.1

Même si le diagramme des cas d'utilisation donne une représentation simple du système et montre les relations entre les acteurs et les cas d'utilisation, nous avons besoin de diriger vers le raffinement de quelques cas d'utilisation.

#### Description textuelle du cas d'utilisation : « S'authentifier »

Le tableau 8 présente le raffinement de cas d'utilisation « S'authentifier » avec les contrôles nécessaires et nous introduisons le scénario nominal ainsi que les exceptions qui peuvent se produire.

Raffinement de cas d'utilisation	
<b>Titre</b>	S'authentifier
<b>Acteur</b>	Utilisateur (Candidat ou Responsable des ressources humaines).
<b>Objectif</b>	Accéder à l'espace spécifique pour chaque utilisateur
<b>Résumé</b>	L'utilisateur accède à son espace pour pouvoir accomplir des services et fonctionnalités selon le rôle (Candidat ou Responsable des ressources humaines).
Conditions	
<b>Préconditions</b>	<b>Postconditions</b>
L'utilisateur doit avoir un compte.	L'utilisateur se connecte à son compte l'espace spécifique à son rôle.
Scénario	
1- Le système affiche le formulaire d'authentification. 2- L'utilisateur saisit ses informations de login et mot de passe dans le formulaire 3- Le système vérifie les informations insérées 4- Un token sera générer coté back contenant les informations de l'utilisateur et une redirection vers la page d'accueil.	
Exception	
Dans l'étape 2 du scénario nominal, si les informations de connexions (login et/ou mot de passe) sont erronées, l'interface affiche une exception.	

Tableau 8: Description textuelle du cas d'utilisation : « S'authentifier »

## 2.4. Conception du sprint 1.1

Après avoir présenté l'analyse de notre sprint, nous passons à l'étape de la conception. Dans chaque sprint nous avons spécifier les attributs, les relations et les méthodes du ce sprint. Tout d'abord, nous commençons par la présentation des classes et ensuite nous introduisons le diagramme de séquence du sprint 1.1

### 2.4.1. Diagramme des classes de conception

La figure 10 décrit le diagramme de classes utilisées dans le deuxième sprint, leurs attributs, leurs méthodes et les relations entre elles.

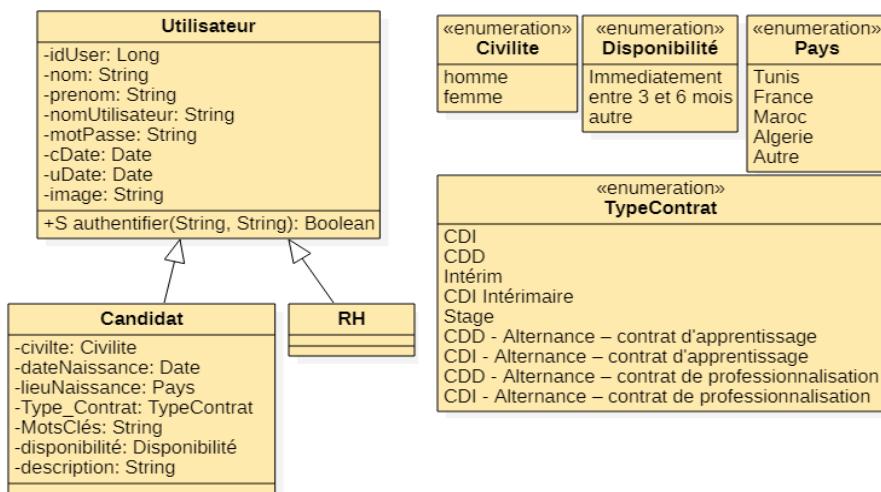


Figure 10: Diagramme de classe du sprint 1.1

Nous avons développé dans le premier sprint 3 classes :

- Classe « *Utilisateur* » : C'est la classe mère de notre système. Elle modélise les différents utilisateurs de l'application à savoir les candidats et le responsable des ressources humaines de la BFI. Elle est définie par les attributs tels que *idUser*, *nom*, *prenom*, *nomUtilisateur*, *motPasse*, *cDate*, *uDate* et *image*. Elle offre la méthode *S'authentifier(String, String)* : Boolean.
- Classe « *RH* » : c'est la classe fille de la classe utilisateur qui représente le responsable des ressources humaines.
- Classe « *Candidat* » : Elle hérite de la classe utilisateur, et elle représente les utilisateurs externes de la BFI qui ont pour but de trouver un emploi chez la BFI. Elle est décrite par l'ensemble des attributs suivants : *civilite*, *dateNaissance*, *lieuNaissance*, *Type\_Contrat*, *MotsClés*, *disponibilite* et *description*.

Dans ce sprint, nous avons utilisés 4 classes énumération :

- Classe « *Civilité* » : C'est une classe d'énumération qui spécifie la civilité des candidats.
- Classe « *Disponibilité* » : C'est une classe d'énumération qui spécifie la disponibilité des candidats.
- Classe « *Pays* » : C'est une classe d'énumération qui spécifie les pays des candidats.
- Classe « *TypeContrat* » : C'est une classe d'énumération qui spécifie le type contrat de recrutement choisi pour postuler à une offre.

#### 2.4.2. Diagramme de séquences du sprint 1.1

Après avoir expliqué le comportement statique du système à travers le diagramme de classes, nous exposons dans la figure 11 le diagramme de séquence du cas d'utilisation "S'authentifier" afin d'expliciter les différentes interactions entre l'acteur du système.

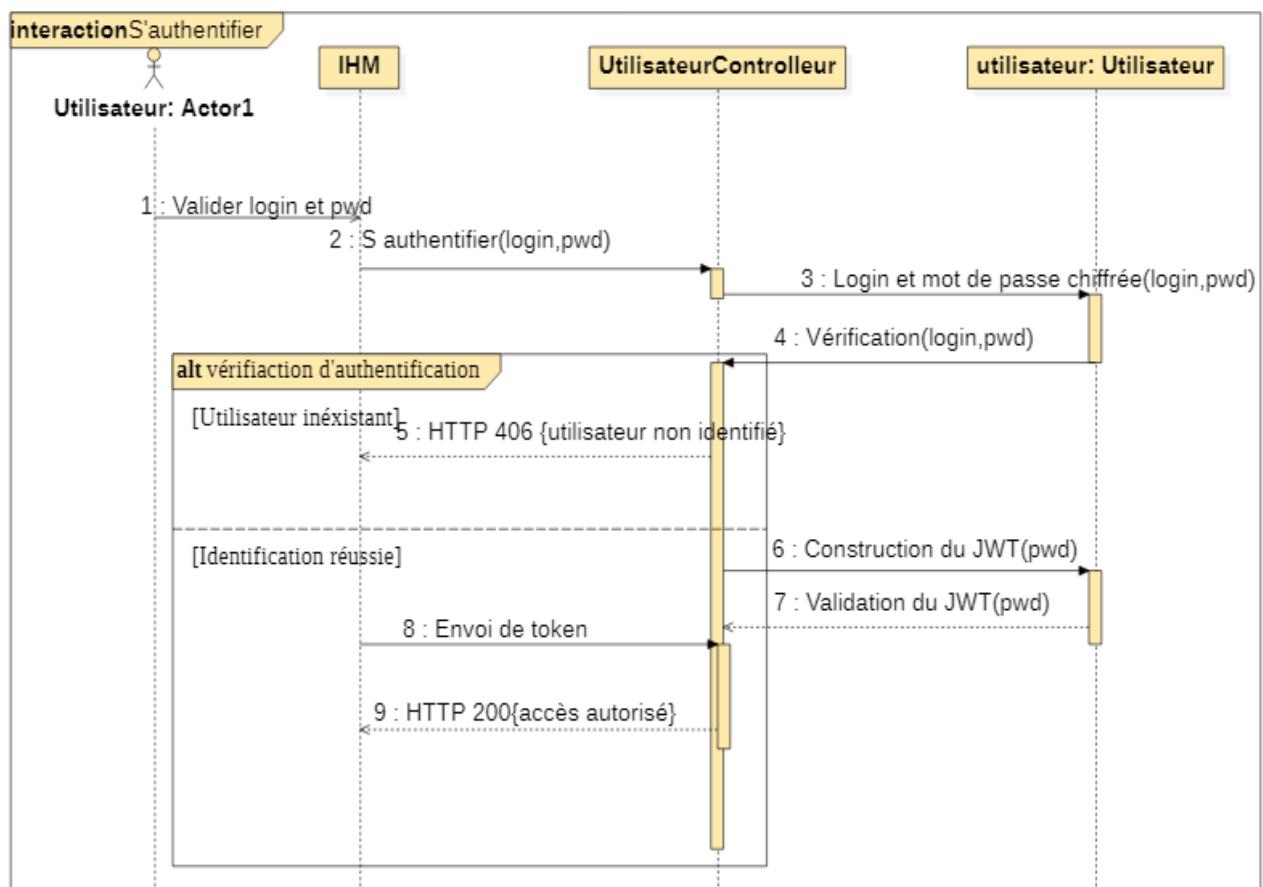


Figure 11: Diagramme de séquence « S'authentifier »

## 2.5. Réalisation du sprint 1.1

Nous décrivons dans cette section l'implémentation et les interfaces du premier sprint.

### 2.5.1. Interface d'accueil

Tout d'abord, pour accéder aux différents espaces spécifiques à chaque utilisateur, il faut passer par l'interface présentée par la figure 12 afin de pouvoir se connecter ou créer un compte par la suite.

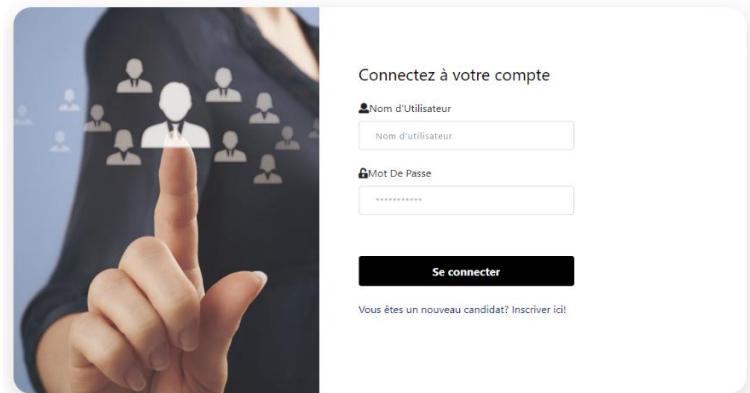


Figure 12: Interface de connexion

### 2.5.2. Interface de connexion

Comme l'indique la figure 13, l'interface d'authentification de notre système oblige l'utilisateur de saisir le nom d'utilisateur et le mot de passe qui doit respecter des exigences de complexité.

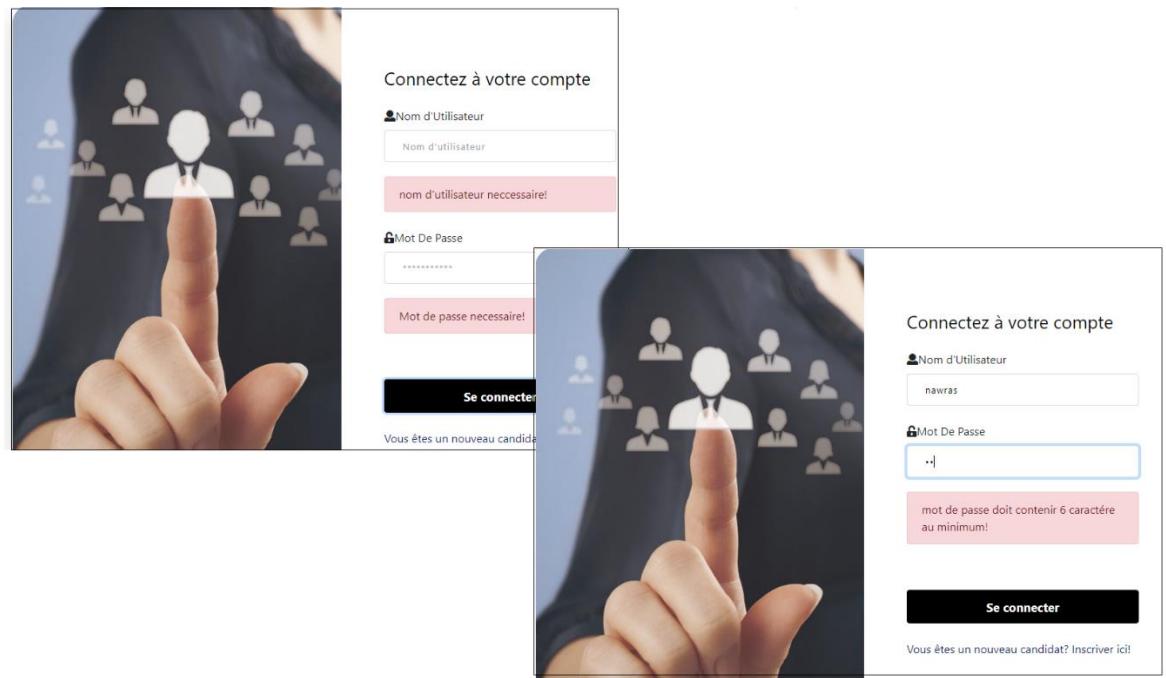
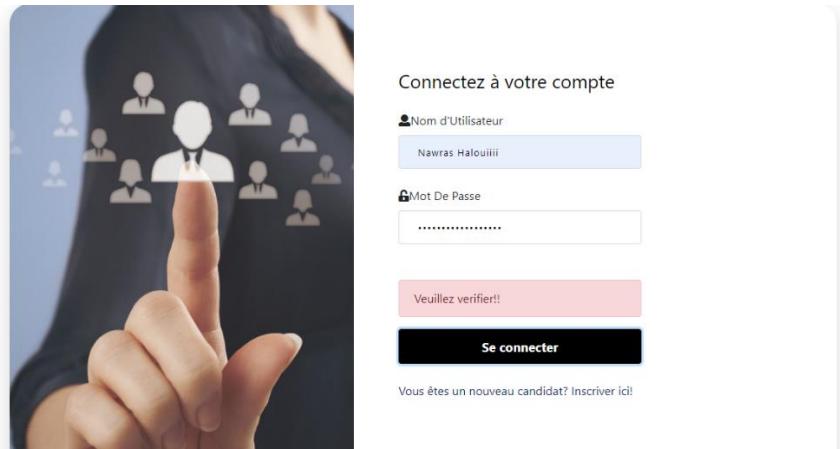


Figure 13: Les exigences de complexité d'authentification

Dans le cas d'un nom d'utilisateur erroné, un message d'erreur va être affiché comme l'indique la figure 17.



*Figure 14: Inexistence d'utilisateur pour l'authentification*

Si l'authentification était effectuée avec succès l'utilisateur serait renvoyé vers son interface suivant son rôle : Espace propre à un candidat ou à la responsable des ressources humaines. La figure 15 illustre les interfaces d'accueil des utilisateurs.

*Figure 15: Authentification des utilisateurs*

### 3. Sprint 1.2 : Gérer le profil

Nous présentons dans cette partie le deuxième sprint intitulé « Gérer le profil ». Nous détaillons les fonctionnalités à mettre en place pour donner une meilleure visibilité sur les tâches réalisées lors de ce sprint. Tout d'abord, nous allons passer par le backlog du sprint

suivi d'un diagramme de cas d'utilisation et d'une analyse contextuelle. Ensuite, nous passons à la conception et nous terminons par les interfaces de la réalisation.

### 3.1.Backlog du sprint 1.2

Le tableau 9 représente le backlog de notre deuxième sprint "Gérer le profil". Nous exposons dans cette partie les user stories liés à ce sprint pour pouvoir les analyser dans la suite de ce chapitre.

ID Module	Module	Id user Story	User story
1	<b>Gérer le profil</b>	1.1	En tant qu'un candidat, je peux ajouter un profil.
		1.2	En tant qu'un candidat, je peux modifier des coordonnées dans le profil.
		1.3	En tant qu'un candidat, je peux afficher le profil.

Tableau 9: Backlog du sprint « Gérer le profil »

### 3.2. Analyse des besoins du sprint 1.2

Après avoir fixé nos exigences dans le backlog du sprint, nous nous intéressons dans cette section à les détailler en présentant le diagramme de cas d'utilisation et la relation entre l'utilisateur et les différentes fonctionnalités dans lesquelles il est impliqué.

#### 3.2.1. Diagramme de cas d'utilisation

Le diagramme 22 représente les fonctionnalités du deuxième sprint " Gérer le profil". Nous présentons dans cette partie l'interaction entre les acteurs et les différentes fonctionnalités.

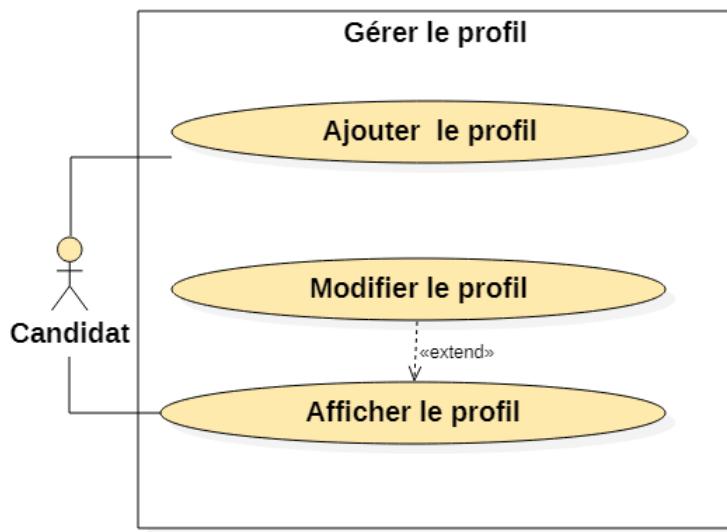


Figure 16: Diagramme de cas d'utilisation « Gérer le profil »

### 3.2.2. Raffinement des cas d'utilisation

#### Description textuelle du cas d'utilisation « Ajouter le profil »

Le tableau 10 explique le cas d'utilisation « Ajouter le profil ». Nous décrivons dans ce tableau l'objectif ainsi que l'acteur principal de la fonctionnalité et nous introduisons le scénario nominal.

Raffinement de cas d'utilisation	
<b>Titre</b>	Ajouter un profil
<b>Acteur</b>	Candidat
<b>Objectif</b>	Ajouter des informations relatives à un candidat donné
<b>Résumé</b>	Profil candidat ajouté
Conditions	
<b>Préconditions</b>	<b>Postconditions</b>
Candidat n'admet pas un compte.	Profil ajouté avec succès.
Scenario	
1- Le candidat se dirige vers l'interface de création d'un compte. 2- Le candidat ajoute la section des informations personnelles. 3- Le candidat ajoute ces expériences. 4- Le candidat ajoute ces certifications et ces réalisations. 5- Le candidat ajoute la section du CV. 6- Le candidat clique sur l'icône créer. 7- Le système enregistre l'ajout du profil et affiche un message d'accéder à son espace.	
Exception	
Dans l'étape 6 du scénario nominal, si l'utilisateur clique sur annuler, retour à étape 5.	

Tableau 10: Description textuelle du cas d'utilisation "Ajouter le profil"

#### Description textuelle du cas d'utilisation « Modifier le profil »

Le tableau 11 illustre un tableau récapitulatif du cas d'utilisation « Modifier le profil ».

Raffinement de cas d'utilisation	
<b>Titre</b>	Modifier le profil
<b>Acteur</b>	Candidat
<b>Objectif</b>	Changer des informations relatives à un candidat donné
<b>Résumé</b>	Profil candidat mis à jour
Conditions	
<b>Préconditions</b>	<b>Postconditions</b>
Le profil du candidat doit être affiché.	Profil mis à jour avec succès.
Scenario	
1- Le candidat se connecte à son espace. 2- Le candidat choisit la section à modifier. 3- Le candidat clique sur l'icône modifier. 4- Le système affiche un formulaire avec les informations du candidat en mode texte. 5- Le candidat change les informations souhaitées et valide. 6- Le système affiche un récapitulatif des changements. 7- Le système enregistre les modifications faites et affiche un message de succès.	

## Exception

Dans l'étape 4 du scénario nominal : - Si l'utilisateur clique sur annuler, retour à étape 3.

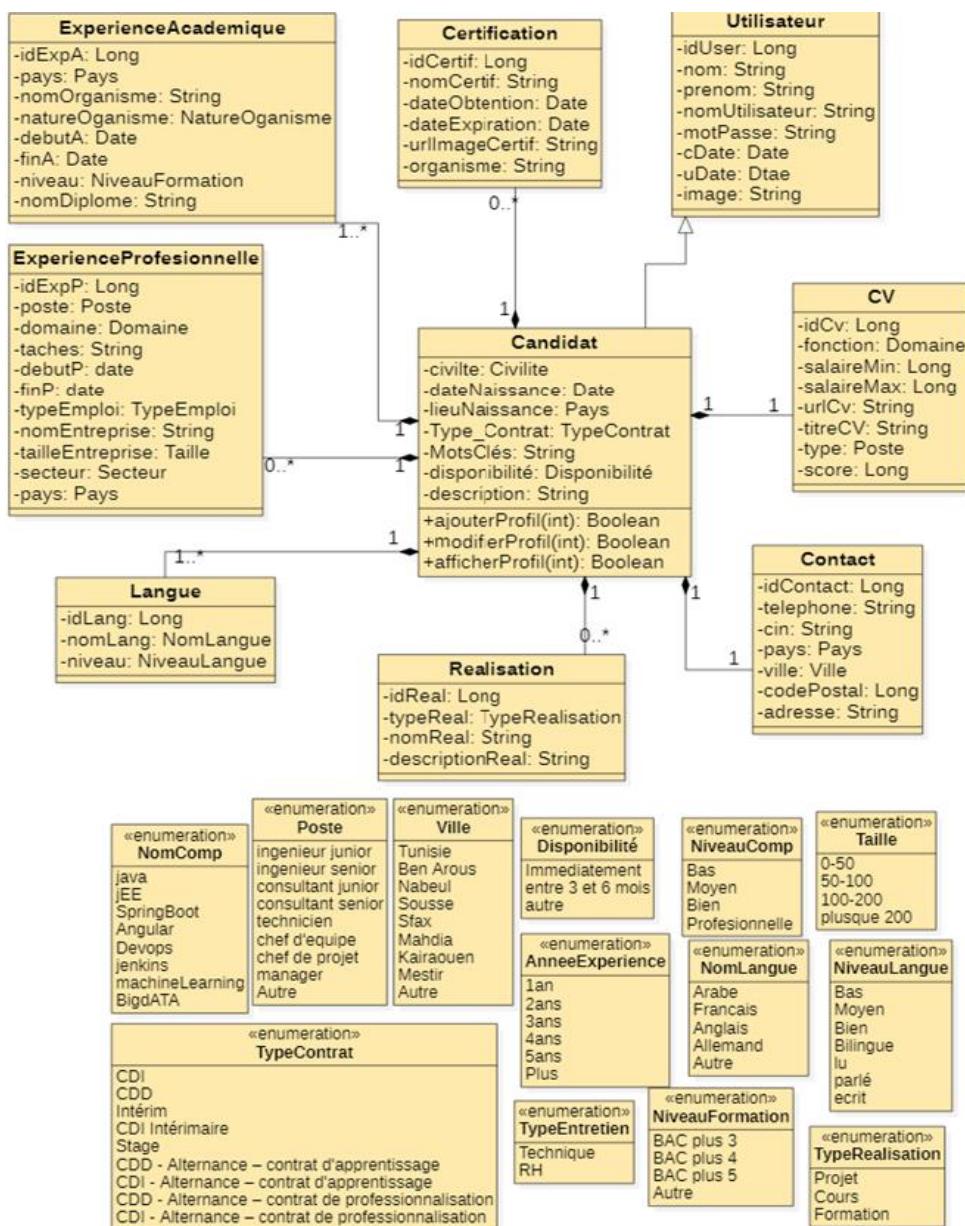
*Tableau 11: Description cas d'utilisation « Modifier le profil »*

### 3.3. Conception du sprint 1.2

Après avoir présenté l'analyse de notre sprint, nous passons à l'étape de la conception. Tout d'abord, nous commençons par la présentation des classes et ensuite nous introduisons le diagramme de séquence du sprint 1.2

#### 3.3.1. Diagramme des classes de conception

La figure 20 présente décrit le système de classes utilisées dans le deuxième sprint, leurs attributs, les méthodes du sprint 1.2 et les relations entre elles.



*Figure 17: Diagramme de classe du sprint 1.2*

Notre diagramme de classe comporte 9 classes que nous allons décrire comme suit :

- Classe « *Candidat* » : Elle hérite de la classe utilisateur, et elle représente les utilisateurs externes de la BFI qui ont pour but de trouver un emploi. Elle est définie par les attributs *civilite*, *dateNaissance*, *lieuNaissance*, *Type\_Contrat*, *MotsClés*, *desponibilité* et *description*. Cette classe offre les méthodes *AjouterProfil(int):Boolean*, *modifierProfil(int):Boolean* et *afficherProfil(int):Boolean*
- Classe « *CV* » : C'est la classe principale de l'application. En effet, elle contribue à la sélection adéquate des candidatures pour chaque offre grâce à l'analyse selon plusieurs critères fixés par le responsable des ressources humaines. Elle est caractérisée par les attributs *idCv*, *fonction*, *salaireMax*, *urlCV*, *titreCv*, *type* et *score*.
- Classe « *Expérience Académique* » : Elle représente les expériences académiques d'un candidat dans sa carrière professionnelle. Elle est définie par les attributs *idExpA*, *pays*, *nomOrganisme*, *natureOrganisme*, *debutA*, *finA*, *niveau* et *nomDiplome*.
- Classe « *Expérience Professionnelle* » : Elle représente les stages, et les anciens postes occupés par un candidat et elle contient les attributs *idExp*, *poste*, *domaine*, *taches*, *debutP*, *finP*, *typeEmploi*, *nomEntreprise*, *tailleEntreprise*, *secteur* et *pays*.
- Classe « *Réalisation* » : Elle représente les projets et les réalisations professionnelles des candidats. Elle est définie par les attributs *idReal*, *typeReal*, *nomReal* et *descriptionReal*.
- Classe « *Certification* » : Elle représente les certifications obtenues par candidat pour renforcer son profil. Elle contient l'ensemble des attributs *idCertif*, *nomCertif*, *dateObtention*, *dateExpiration*, *urlImageCertif* et *organisme*.
- Classe « *Contact* » : Elle représente les informations personnelles des employés et des candidats pour pouvoir les contacter. Elle fournit l'ensemble des attributs *idContact*, *telephone*, *cin*, *pays*, *ville*, *codePostal* et *adresse*.
- Classe « *Langue* » : Elle représente le niveau des langues pour les candidats. Elle est caractérisée par les attributs *idLang*, *nomLang* et *niveau*.

Notamment, nous avons montré les classes énumération qui correspondent aux classes du diagramme de classe du sprint 1.2 :

- Classe « *NomComp* » : C'est une classe d'énumération qui définit les compétences.
- Classe « *NiveauComp* » : C'est une classe d'énumération qui définit les niveaux des compétences des candidats.
- Classe « *Poste* » : C'est une classe d'énumération qui spécifie les différents postes.
- Classe « *Ville* » : C'est une classe d'énumération qui énumère les villes des candidats.
- Classe « *Disponibilité* » : C'est une classe d'énumération qui détermine la disponibilité des candidats (immédiatement, entre 3 et 6 mois, etc)
- Classe « *Taille* » : C'est une classe d'énumération qui définit la taille d'une entreprise.
- Classe « *AnneeExperience* » : C'est une classe d'énumération qui définit le nombre d'années d'expériences.
- Classe « *NomLangue* » : C'est une classe d'énumération qui définit la langue.
- Classe « *TypeRealisation* » : C'est une classe d'énumération qui définit le type de réalisation du candidat.

### 3.3.2. Diagramme de séquences du sprint 1.2

Après avoir expliqué le comportement statique du système à travers le diagramme de classes, nous exposons dans la figure 18 le diagramme de séquence du cas d'utilisation "Gérer le profil" afin d'expliquer les différentes interactions entre l'acteur du système.

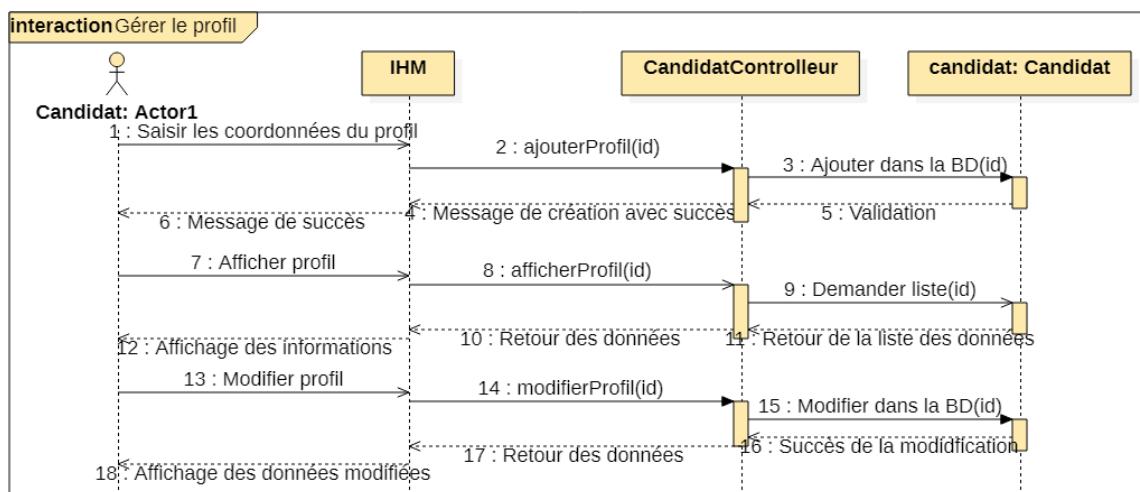


Figure 18: Diagramme de séquence du sprint 1.2

### 3.4. Réalisation du sprint 1.2

Dans cette partie, nous avons étudié les différentes interfaces réalisées tout de long du sprint 1.2.

### 3.4.1. Interface de création de compte

Après avoir sélectionné « créer un compte », l'utilisateur sera redirigé vers l'interface de création d'un compte, un contrôle de saisie sera affecté lors de création d'un compte des candidats lorsque le nom d'utilisateur existe déjà dans ce cas l'inscription ne sera pas être affectée comme il est montré dans la figure 19.

Nom d'utilisateur  
Nawras Halouii

Email  
nawrashalou@gmail.com

Mot de passe  
.....

S'INSCRIRE

Inscription non établie!

Figure 19: Interface de création d'un compte

Si la création d'un compte a été effectuée avec succès, le candidat sera redirigé vers l'interface de création du profil pour compléter les étapes d'inscription comme il est montré dans la figure 20.

Inscription effectuée avec succès!

CRÉER PROFIL

Figure 20: Interface de la création d'un profil

Après avoir passé par toutes les étapes du formulaire, l'interface de la figure 21 sera être affichée.

REmplir vos informations pour l'inscription

Profil

Expériences

Réalisation et Certification

Curriculum Vitae

Inscription établie avec succès

Accéder à votre espace

Figure 21: Dernière étape d'inscription

### 3.4.2. Interfaces de gestion du profil

Chaque candidat possède un écran dans lequel il peut changer les paramètres de son compte tels que ses expériences, son éducation, son CV et ses compétences.

La figure 22 montre l'interface, dans laquelle le candidat affiche les coordonnées du son profil.

The screenshot shows a candidate's profile page. At the top left is a circular profile picture of a woman. Below it, the name "Nawras Haloui" is displayed, followed by the title "Ingenieur en informatique". On the left side, there is a sidebar with three items: "Mon profil", "Mes candidatures", and "Mes quiz et tests techniques". The main content area is divided into sections: "Description" (with a text input field containing "Ma passion pour la technologie m'a conduit à l'ingénierie informatique. J'aimerais travailler avec des groupes polyvalents pour élargir mes connaissances et affiner mes compétences techniques tout en comptant sur ma capacité d'apprentissage."), "Experience" (listing "Ingenieur web" with a note "Conception et développement d'une Application Web de 'Gestion des clubs universitaires'.") and "Education" (listing "Ingenieur en UI / UX").

Figure 22: Affichage des coordonnées du profil candidat

La figure 23 illustre la modification au niveau du profil des candidats.

This figure illustrates the modification of a candidate's profile. It consists of two screenshots. The top screenshot shows a modal dialog titled "Description" with a text input field containing "Mon objectif est de trouver un poste de développement en informatique". The bottom screenshot shows the updated profile page where the same description has been saved. A red arrow points from the modal dialog to the updated description on the profile page.

Figure 23: Modification du profil candidat

## 4. Sprint 1.3 : Gérer les offres et les candidatures

Nous présentons dans cette partie le troisième sprint intitulé « Gérer les candidatures et les offres ». Nous étudions les fonctionnalités à faire lors de ce sprint. En premier lieu,

nous allons passer par le backlog du sprint suivi d'un diagramme de cas d'utilisation et d'une analyse contextuelle et enfin nous passons à la conception et nous terminons par les interfaces de la réalisation.

#### **4.1.Backlog du sprint du sprint 1.3**

Le tableau 12 représente le backlog de notre sprint "Gérer les offres et les candidatures". Nous exposons dans cette partie les user stories liés à ce sprint pour pouvoir les analyser dans la suite de ce chapitre.

<b>ID Module</b>	<b>Module</b>	<b>Id user Story</b>	<b>User story</b>
1	<b>Gérer les offres et les candidatures.</b>	1.1	En tant qu'un responsable des ressources humaines, je peux ajouter les responsables Techniques.
		1.2	En tant qu'un responsable des ressources humaines, je peux modifier les responsables Techniques.
		1.3	En tant qu'un responsable des ressources humaines, je peux supprimer les responsables Techniques.
		1.4	En tant qu'un responsable des ressources humaines, je peux ajouter les responsables Tests.
		1.5	En tant qu'un responsable des ressources humaines, je peux modifier les responsables Tests.
		1.6	En tant qu'un responsable des ressources humaines, je peux supprimer les responsables Tests.
		1.7	En tant qu'un responsable des ressources humaines, je peux ajouter les responsables Techniques.

*Tableau 12: Backlog du Sprint 1.3*

#### **4.2. Analyse des besoins du sprint 1.3**

Dans cette partie, nous nous intéressons à présenter le diagramme de cas d'utilisation et la relation entre l'utilisateur et les différentes fonctionnalités.

##### **4.2.1. Diagramme de cas d'utilisation**

Le diagramme représente les fonctionnalités du troisième sprint " Gérer les offres et les candidatures". Nous présentons dans la figure 24 l'interaction entre les acteurs et les différentes fonctionnalités.

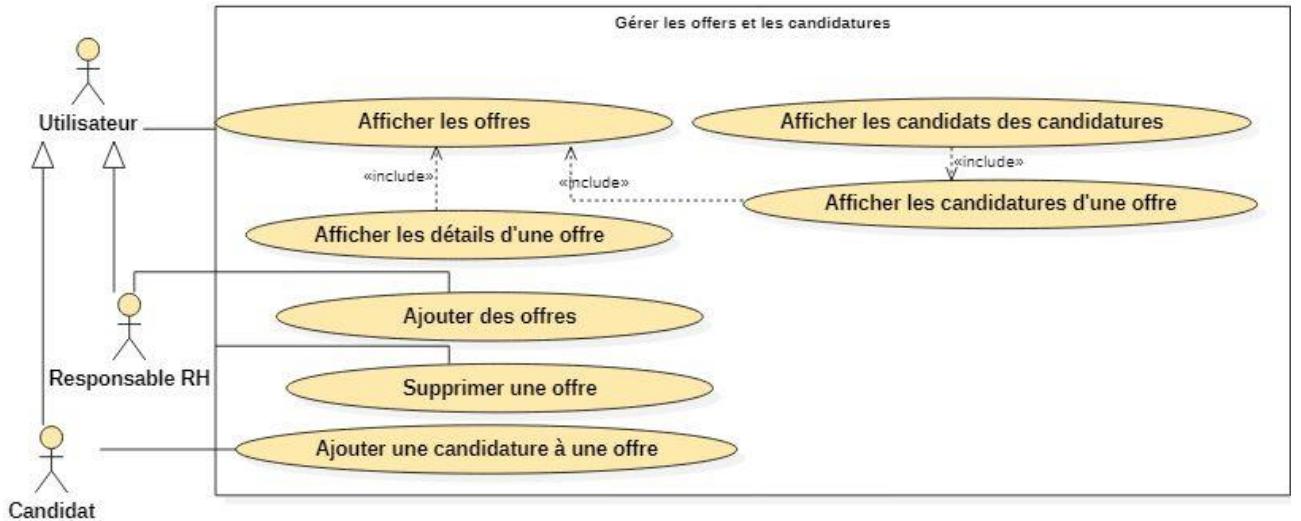


Figure 24: Diagramme de cas d'utilisation du sprint 1.3

#### 4.2.2. Raffinement des cas d'utilisation

Dans cette partie, nous allons étudié les descriptions textuelles du sprint 1.3.

##### Description textuelle du cas d'utilisation « Ajouter une candidature à une offre »

Le tableau 13 illustre une description du cas d'utilisation « Ajouter une candidature à une offre ». Nous spécifions dans ce tableau l'objectif ainsi que l'acteur principal de la fonctionnalité et nous précisons le scénario nominal et les exceptions qui peuvent se produire.

Raffinement de cas d'utilisation	
<b>Titre</b>	Ajouter une candidature à une offre
<b>Acteur</b>	Candidat
<b>Objectif</b>	Postuler à une offre qui correspond à ces compétences et son profil.
<b>Résumé</b>	Le candidat doit postuler à une offre qui contient plusieurs détails concernant ses compétences et notamment son CV.
Conditions	
Préconditions	Postconditions
Le candidat doit être connecté à l'application.	Le candidat ajoute une candidature d'une offre avec succès.
Scénario	
1- Le candidat sera redirigé vers l'interface d'accueil après avoir connecté à l'application. 2- Le système affiche les derniers offres publiés par le responsable des ressources humaines. 3- Le candidat doit vérifier les détails de l'offre. 4- Le système affiche un message de succès.	
Exception	

Dans l'étape 3 du scénario nominal, si le candidat sélectionne le bouton annuler il sera redirigé vers l'étape précédente.

*Tableau 13: Description cas d'utilisation « Ajouter une candidature à une offre »*

### **Description textuelle du cas d'utilisation « Supprimer une offre »**

Le tableau 14 illustre un tableau récapitulatif du cas d'utilisation « Supprimer une offre ». Nous décrivons dans ce tableau l'objectif ainsi que l'acteur principal de la fonctionnalité et nous introduisons le scénario nominal.

<b>Raffinement de cas d'utilisation</b>	
<b>Titre</b>	Supprimer une offre.
<b>Acteur</b>	Responsable des ressources humaines.
<b>Objectif</b>	Supprimer une offre
<b>Résumé</b>	Offre supprimée.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Postconditions</b>
Offre non supprimée.	Offre supprimée avec succès.
<b>Scenario</b>	
1- Le responsable des ressources humaines se connecte à son espace. 2- Le responsable des ressources humaines choisit l'offre à supprimer. 3- Le responsable des ressources humaines supprime les offres souhaitées. 4- Le système enregistre les modifications faites et affiche un message de succès.	
<b>Exception</b>	
Dans l'étape 3 du scénario nominal, si l'utilisateur clique sur annuler, retour à étape 2.	

*Tableau 14: Description cas d'utilisation « Supprimer une offre »*

## **4.3. Conception du sprint 1.3**

Dans cette partie, nous commençons par la présentation des classes et ensuite nous introduisons le diagramme de séquence du sprint 1.3

### **4.3.1. Diagramme des classes de conception du sprint 1.3**

La figure 25 présentée décrit le diagramme de classe utilisé dans le troisième sprint.

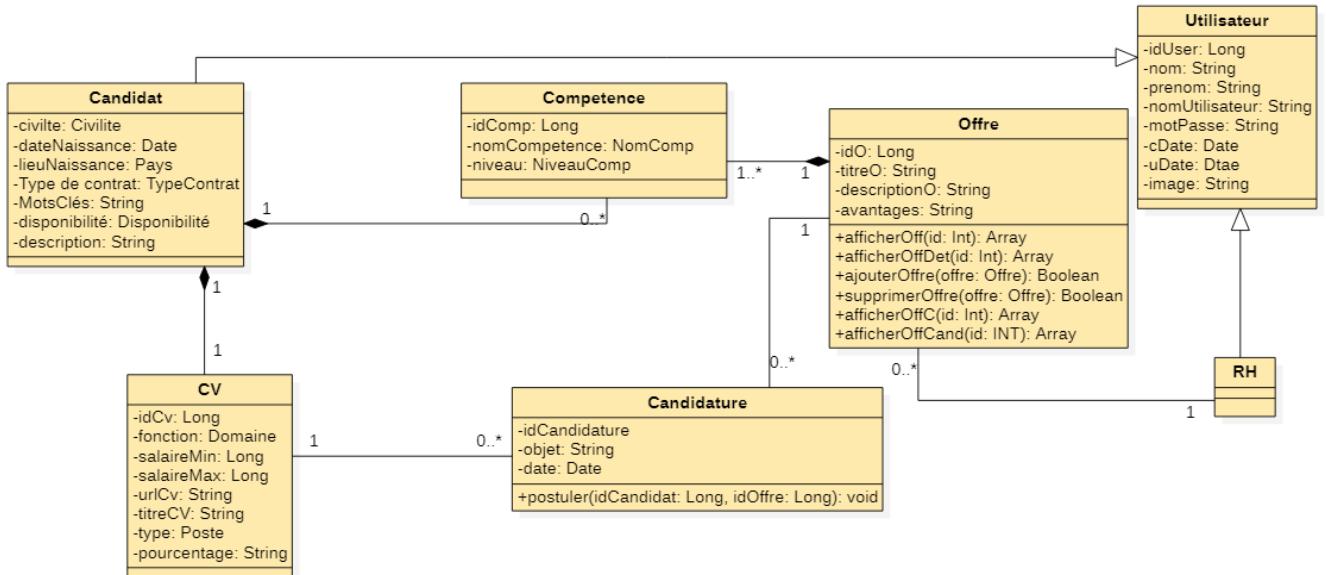


Figure 25: Diagramme de classe du sprint 1.3

Nous avons travaillé dans ce sprint avec les classes :

- Classe « Offre » : elle est définie par les attributs *idO*, *titreO*, *descriptionO*, *avantages* et par les méthodes *afficherOff(int) :Array*, *afficherOffDet(int) :Array*, *ajouterOffre(Offre) :Boolean*, *supprimerOffre(Offre) :Boolean*, *afficherOffCand(id : int) :Array* et *afficherOffC(int) :Array*
- Classe « Candidature » : Elle est caractérisée par les attributs *idCandidature*, *date* et *objet* et elle contient la méthode *postuler(Long,Long) :void*.

Les autres classes « *Candidat* », « *Compétence* », « *Utilisateur* », « *RH* » et « *CV* » nous avons déjà les décrits dans les sprints précédents.

#### 4.3.2. Diagramme de séquences du sprint 1.3

A cette étape, nous allons exposer dans la figure 26 le diagramme de séquence du cas d'utilisation "Gérer les offres et les candidatures" afin d'expliciter les différentes interactions entre l'acteur du système.

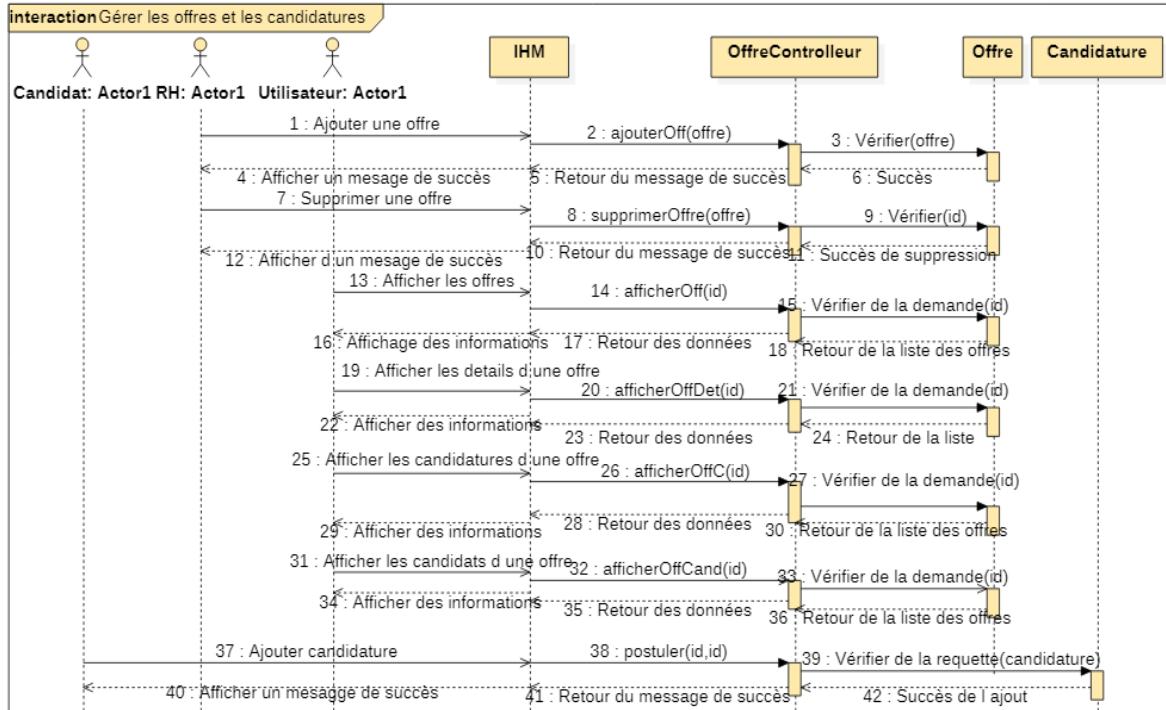


Figure 26: Diagramme de séquence du sprint « Gérer les candidatures et les offres »

## 4.4. Réalisation du sprint 1.3

Dans ce sprint, nous allons étudier les interfaces de l'application de gestion des candidatures et des offres.

### 4.4.1. Interfaces de gestion des candidatures

Chaque candidat et responsable des ressources humaines possède un écran dans lequel il visualise les candidatures de chaque offre et donc la possibilité de postuler à des offres pour les candidats et de supprimer les candidatures de chaque offre pour le responsable des ressources humaines.

La figure 27 montre l'interface, dans laquelle le responsable des ressources humaines permet d'ajouter des offres.

## Ajouter une offre

Poste de l'offre: Ingenieur en informatique

Description: bfi recrute pour le compte d'une banque de renommée : des Ingénieurs en Informatique Si vous êtes intéressé veuillez envoyer votre CV

Avantage: Un poste en CDI  
Des bureaux agréables  
Un environnement innovant  
Un package compétitif composé de : fixe+ bonus individuel  
Et bien plus...

Education: Ecole d'ingénieur, BAC + 5 ou équivalent

Experience: 3ANS

Description de l'Experience: Une expérience dans le domaine de la finance de marché ou idéalement dans le domaine de Data Management serait un plus.  
Vous possédez impérativement des connaissances techniques solides et dev orienté objet.

Competence: PHP  
Java

Responsable technique: Mr BACHA SKANDAR

Offre

Ingenieur en informatique

bfi recrute pour le compte d'une banque de renommée : des Ingénieurs en Informatique Si vous êtes intéressé veuillez envoyer votre CV.

Education: PHP      Experiences: JAVA      Competences: PHP      JAVA

voir candidatures

Date: 2021-05-25

Voir

Figure 27: Ajout des offres

En contrepartie, la figure 28 explique la possibilité du candidat de postuler à ces offres pour trouver un emploi chez BFI, et donc le responsable des ressources humaines visualise la liste des candidatures à cette offre.

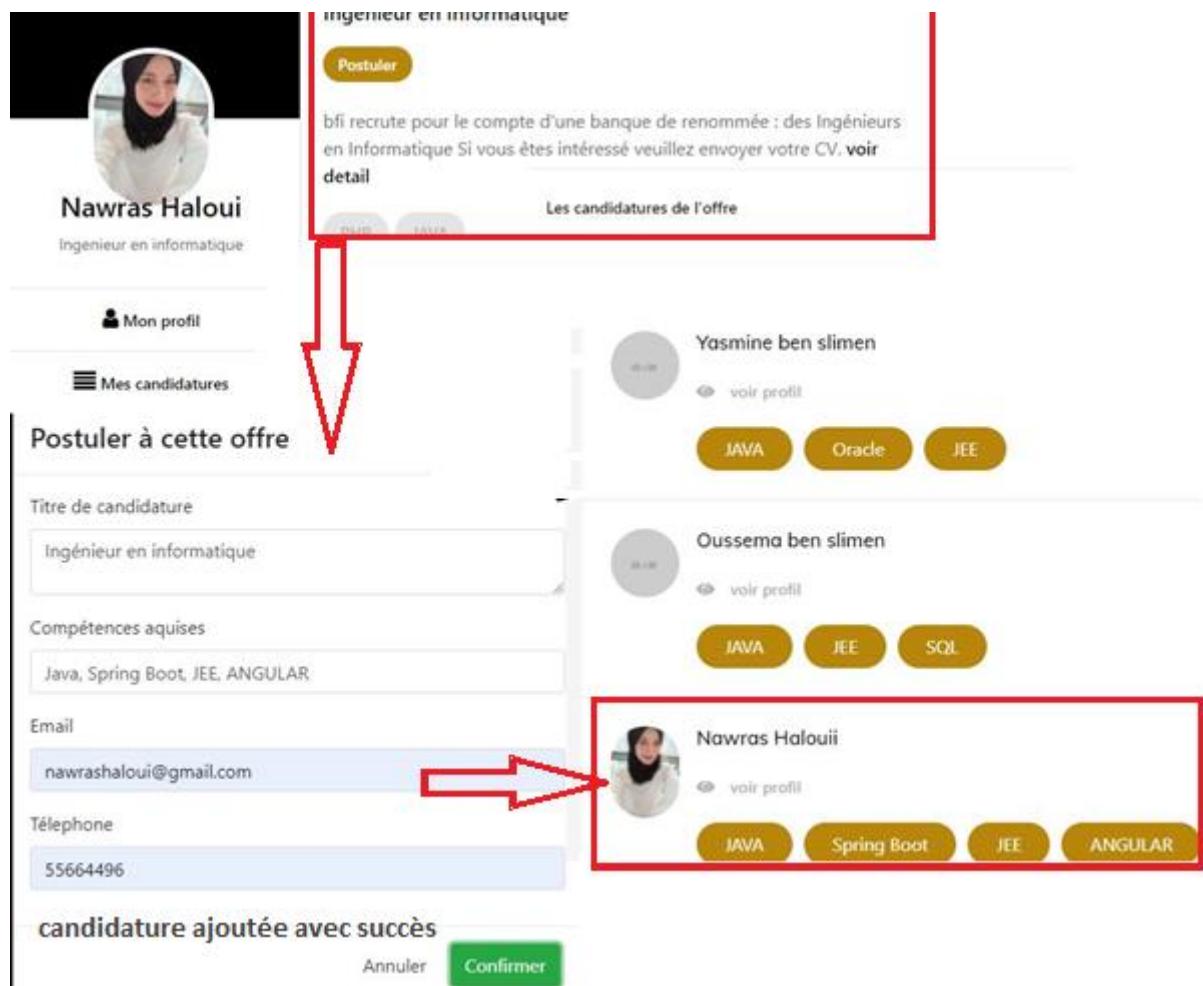


Figure 28: Postulation des candidats aux offres

## Conclusion

À travers ce chapitre, nous avons présenté les trois sprints du premier release. Nous avons décrit pour chaque sprint l'analyse contextuelle en passant par le backlog produit et la spécification fonctionnelle, puis la conception et enfin la réalisation. Le chapitre suivant sera consacré à la présentation du deuxième release.

# Chapitre 4 : Release 2 « Gestion des responsables, des candidats & des entretiens »

## Introduction

Le présent chapitre a pour objectif de présenter les différentes phases conduisant à la mise en œuvre du deuxième release. Nous étudions tous les détails de la phase d'analyse, de conception et de réalisation des sprints. Nous présentons le diagramme de cas d'utilisation, le diagramme des classes ainsi que le diagramme de séquence pour donner une meilleure compréhension des exigences utilisateurs.

## 1. Planification de la release 2

Cette section vise à exposer les différentes phases de mise en œuvre du deuxième release. Nous avons décomposé le release en 3 sprints comme l'illustre le tableau 15 qui nous permettent à la fin de chacun de réaliser un livrable

Sprint 2.1 : Gérer les responsables techniques et de tests.	Sprint 2.2 : Gérer les candidats et les tests.	Sprint 2.3 : Gérer les entretiens
<ul style="list-style-type: none"><li>• Ajouter les responsables.</li><li>• Supprimer les responsables.</li><li>• Modifier les coordonnées des responsables. (De 17/03/2021 à 22/03/2021)</li></ul>	<ul style="list-style-type: none"><li>• Envoyer des mails automatiques pour le processus de recrutement</li><li>• Supprimer des candidats.</li><li>• Réalisation des tests de recrutement (De 23/03/2021 à 10/04/2021)</li></ul>	<ul style="list-style-type: none"><li>• Ajouter un entretien au calendrier.</li><li>• Supprimer une date du calendrier. (De 11/04/2021 à 20/04/2021)</li></ul>

Tableau 15: Planification du release 2

## 2. Sprint 2.1 : Gérer les responsables

A ce niveau, nous allons analyser le sprint 1.2 « Gérer les responsables »

### 2.1. Backlog du sprint 2.1

Le tableau 16 représente le backlog de notre sprint " Gérer les responsables " où nous listons les différentes exigences à mettre en place dans cette section.

<b>ID Module</b>	<b>Module</b>	<b>Id user Story</b>	<b>User story</b>
1	<b>Gérer les responsables techniques et de tests</b>	1.1	En tant qu'un responsable des ressources humaines, je peux ajouter un responsable test ou technique.
		1.2	En tant qu'un responsable des ressources humaines, je peux modifier un responsable test ou technique.
		1.3	En tant qu'un responsable des ressources humaines, je peux supprimer un responsable test ou technique.

Tableau 16: Backlog du sprint 2.1

## 2.2. Analyse du sprint 2.1

Suite à l'étape de déterminaison des exigences dans le deuxième sprint nous passons au diagramme de cas d'utilisation pour plus de détails et nous présentons un diagramme de séquence qui contient les fonctions essentielles.

### 2.2.1. Diagramme de cas d'utilisation du sprint 2.1

La figure 29 illustre le diagramme du cas d'utilisation du sprint "Gestion des responsables techniques et des tests" qui contient toutes les fonctionnalités à réaliser.

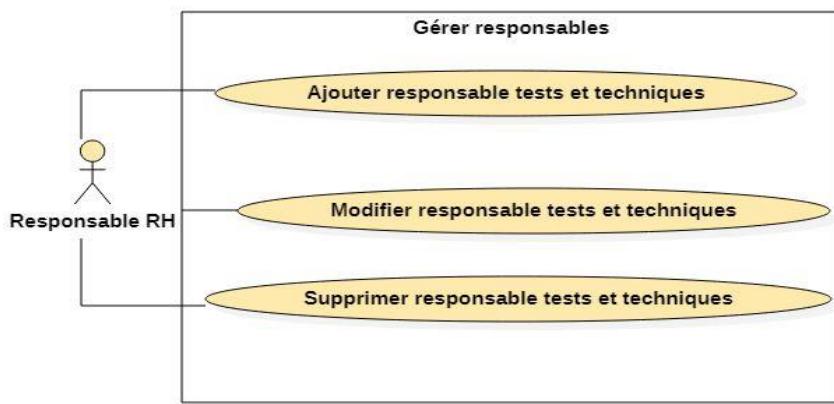


Figure 29: Diagramme de cas d'utilisation du sprint 2.1

### 2.2.2. Raffinements des cas d'utilisation

Dans cette partie, nous allons étudier l'analyse de gestion des responsables techniques et responsables tests.

#### Description textuelle du cas d'utilisation « Ajouter responsable »

Le tableau 17 illustre un tableau récapitulatif du cas d'utilisation « Ajouter responsables ». Nous décrivons dans ce tableau l'objectif, l'acteur principal de la fonctionnalité, le scénario nominal et les exceptions qui peuvent se produire.

<b>Raffinement de cas d'utilisation</b>	
<b>Titre</b>	Ajouter responsables
<b>Acteur</b>	Responsable des ressources humaines
<b>Objectif</b>	Ajouter les responsables techniques et de tests chez BFI responsable du recrutement.
<b>Résumé</b>	Le responsable des ressources humaines doit ajouter les responsables chez BFI.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Postconditions</b>
Le responsable des ressources humaines doit être connecté à l'application.	Le responsable des ressources humaines ajoute les responsables avec succès.
<b>Scénario</b>	
1- Le responsable des ressources humaines sera redirigé vers l'interface gestion des responsables techniques ou des responsables techniques après avoir connecté à l'application. 2- Le système affiche la liste des responsables. 3- Le responsable des ressources humaines ajoute un nouveau responsable. 4- Le responsable des ressources humaines remplit le formulaire d'ajout et clique sur confirmer. 5- Le système affiche un message de succès.	
<b>Exception</b>	
Dans l'étape 3 et 4 du scénario nominal, si un des champs est invalide, le système affiche un message d'erreur	

Tableau 17: Description de cas d'utilisation « Ajouter responsables »

#### Description textuelle du cas d'utilisation « Modifier responsable »

Le tableau 17 illustre un tableau récapitulatif du cas d'utilisation « Modifier responsables ». Nous décrivons dans ce tableau l'objectif, l'acteur principal de la fonctionnalité, le scénario nominal et les exceptions qui peuvent se produire.

<b>Raffinement de cas d'utilisation</b>	
<b>Titre</b>	Modifier responsables
<b>Acteur</b>	Responsable des ressources humaines
<b>Objectif</b>	Modifier les responsables techniques et de tests chez BFI responsable du recrutement.

<b>Résumé</b>	Le responsable des ressources humaines doit ajouter les responsables chez BFI.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Postconditions</b>
Le responsable des ressources humaines doit être connecté à l'application.	Le responsable des ressources humaines modifie les responsables avec succès.
<b>Scénario</b>	
1- Le responsable des ressources humaines sera redirigé vers l'interface gestion des responsables techniques ou des responsables techniques après avoir connecté à l'application. 2- Le système affiche la liste des responsables. 3- Le responsable des ressources humaines modifie un responsable. 4- Le responsable des ressources humaines remplit le formulaire de modification et clique sur confirmer. 5- Le système affiche un message de succès.	
<b>Exception</b>	
Dans l'étape 3 et 4 du scénario nominal, si un des champs est invalide, le système affiche un message d'erreur	

Tableau 18: Description de cas d'utilisation « Modifier responsables »

## 2.3. Conception du sprint 2.1

Après avoir présenté l'analyse contextuelle de notre sprint, nous devons passer à la description de la partie conceptuelle. En premier lieu, nous présentons le diagramme des classes de ce sprint et en second lieu nous montrons le diagramme de séquence « Gérer les responsables ».

### 2.3.1. Diagramme des classes de conception du sprint 2.1

Le diagramme 34 représente le système des classes du premier sprint « Gérer les responsables ». Nous présentons dans cette partie les classes et les relations entre elles.

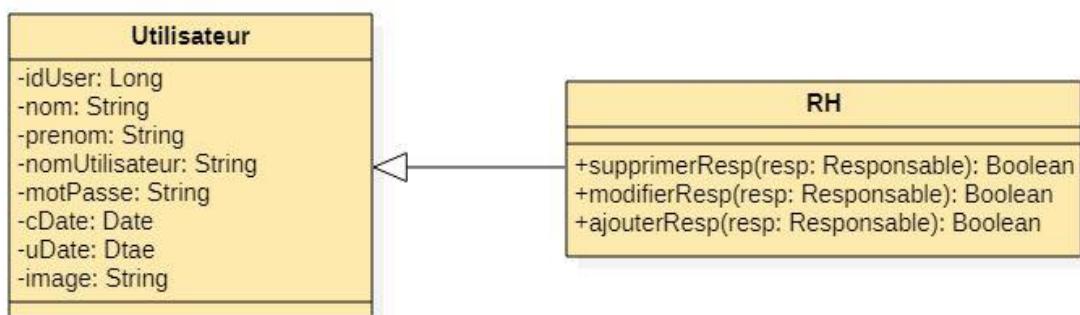


Figure 30: Diagramme de classe du sprint 2.1

Dans ce sprint nous avons défini les deux classes « *Utilisateur* » et « *RH* ».

La classe « RH » présente les méthodes *supprimerResp(Responsable) : Boolean*, *modifierResp(Responsable) : Boolean*, *ajouterResp(Responsable) : Boolean*

### 2.3.2. Diagramme de séquences du sprint 2.1

Après avoir expliqué le comportement statique du système à travers le diagramme de classes, nous exposons dans la figure 31 le diagramme de séquence du cas d'utilisation "Gérer les responsables" afin d'expliciter les différentes interactions entre l'acteur du système

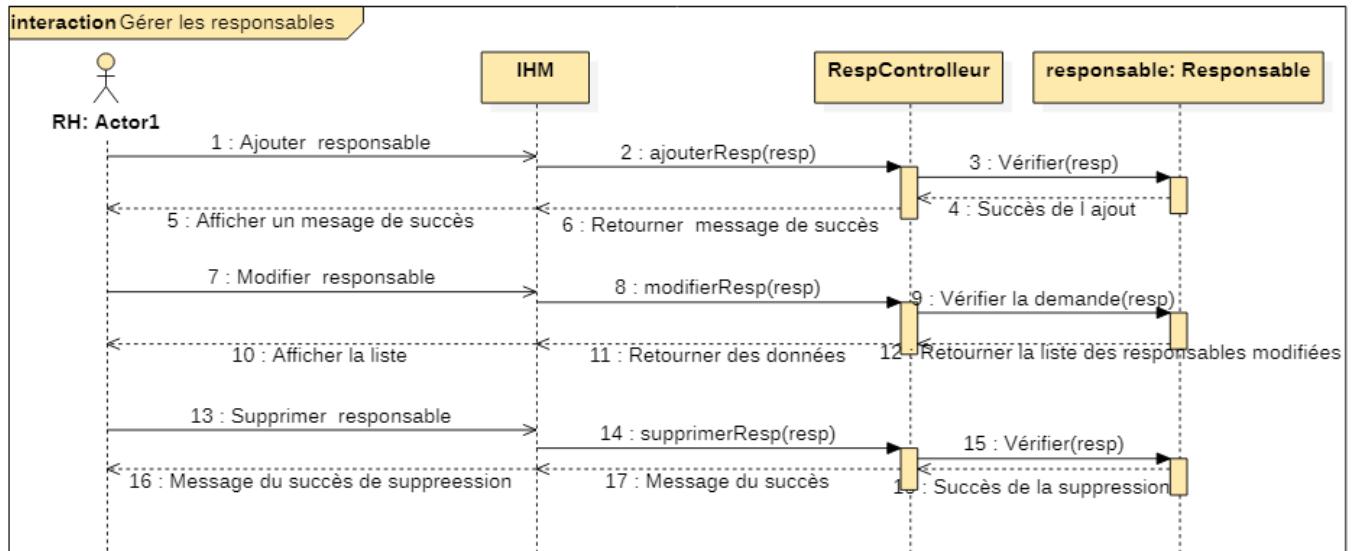


Figure 31: Diagramme de séquence du sprint 2.1

## 2.4. Réalisation du sprint 2.1

Dans cette partie, nous avons étudier l'analyse du sprint 2.1.

### 2.4.1. Interfaces de gestion des responsables techniques

Tenons compte que la gestion des différents responsables s'établit de la même manière, nous allons prendre l'exemple de l'étude des responsables techniques pour montrer sa gestion en détails. Nous présentons dans ce qui suit quelques interfaces représentant le travail élaboré dans ce sprint.

Tout d'abord, voici l'interface qui représente la liste des responsables test.

## Responsables Tests

[Ajouter](#) [Supprimer](#)

<input type="checkbox"/>	Nom complet	Email	Adresse	Telephone	CIN	cDate	uDate	Actions
<input type="checkbox"/>	nesrine ben hriss	nesrine@gmail.com	avenue hbib bourgiba	(216) 58884874	09867292	11/06/2021	12/06/2021	
<input type="checkbox"/>	AHMED	AHMED@mail.com	89 TUNISIA	(216) 55125525	08884846	04/12/2018	05/04/2019	
<input type="checkbox"/>	oUSEMA BEN ASKER	OUSEMA@mail.com	ARIANA 5816	(216) 98615446	08884846	04/12/2018	05/04/2019	
<input type="checkbox"/>	Mariem ddd	mariaanders@mail.com	25, rue bourgiba, tunisie	(216) 74448	08884846	04/12/2018	05/04/2019	

Figure 32: Liste des responsables tests

La figure 33 montre l'ajout d'un responsable test.

Ajouter responsable

Nom complet  
nesrine ben hriss

Email  
nesrine@gmail.com

Address  
avenue hbib bourgiba

Phone  
58884874

## Responsables Tests

[Ajouter](#) [Supprimer](#)

<input type="checkbox"/>	Nom complet	Email	Adresse	Telephone	CIN	cDate	uDate
<input type="checkbox"/>	nesrine ben hriss	nesrine@gmail.com	avenue hbib bourgiba	(216) 58884874	09867292	11/06/2021	12/06/2021
<input type="checkbox"/>	AHMED	AHMED@mail.com	89 TUNISIA	(216) 55125525	08884846	04/12/2018	05/04/2019
<input type="checkbox"/>	oUSEMA BEN ASKER	OUSEMA@mail.com	ARIANA 5816	(216) 98615446	08884846	04/12/2018	05/04/2019
<input type="checkbox"/>	Mariem ddd	mariaanders@mail.com	25, rue bourgiba, tunisie	(216) 74448	08884846	04/12/2018	05/04/2019

CANCEL ADD

Figure 33: Ajout d'un responsable

Pour la modification, si nous prenons la première ligne nous aurons tout d'abord les données de cette ligne remplies par défauts dans les différents champs comme la montre la figure 34.

	Nom complet	Email	Adresse	Téléphone	CIN	cDate	uDate	Actions
<input type="checkbox"/>	nearine ben hrisa	nearine@gmail.com	avenue habib bourgiba	(216) 58884874	09867292	06/2021	2/06/2021	
<input type="checkbox"/>	AHMED	AHMED@mail.com	89 TUNISIA.	(216) 55125525	08884846	04/12/2018	05/04/2019	
<input type="checkbox"/>	OUSEMA BEN ASKER	OUSEMA@mail.com	ARIANA 5016	(216) 98615446	08884846	04/12/2018	05/04/2019	
<input type="checkbox"/>	Mariem ddd	marianders@mail.com	25, rue bourgiba, tunisie	(216) 74448	08884846	04/12/2018	05/04/2019	

Figure 34: Modification des responsables

### 3. Sprint 2.2 : Gérer les candidats et les tests

#### 3.1. Backlog du sprint 2.2

Le tableau 18 représente le backlog de notre sprint « Gérer les candidats et les tests » où nous listons les différentes exigences à mettre en place dans cette section.

ID Module	Module	Id user Story	User story
1	Gérer les candidats et les tests	1.1	En tant qu'un responsable des ressources humaines, je peux choisir des candidats et les contacter par mail.
		1.2	En tant qu'un responsable des ressources humaines, je peux supprimer des candidats.
		1.3	En tant qu'un candidat, je peux réaliser des tests.

Tableau 18: Backlog du sprint 2.2

#### 3.2. Analyse des besoins du sprint 2.2

Cette partie a pour objectif de donner une vue d'analyse sur les exigences du deuxième sprint. Nous décrivons les acteurs ainsi que leurs interactions avec les cas d'utilisations, puis nous passons à la présentation d'un diagramme de séquence système.

##### 3.2.1. Diagramme de cas d'utilisation du sprint 2.2

La figure 38 illustre le diagramme des cas d'utilisation du sprint « Gérer les candidats et les tests » qui décrit les fonctionnalités à envisager dans cette partie.

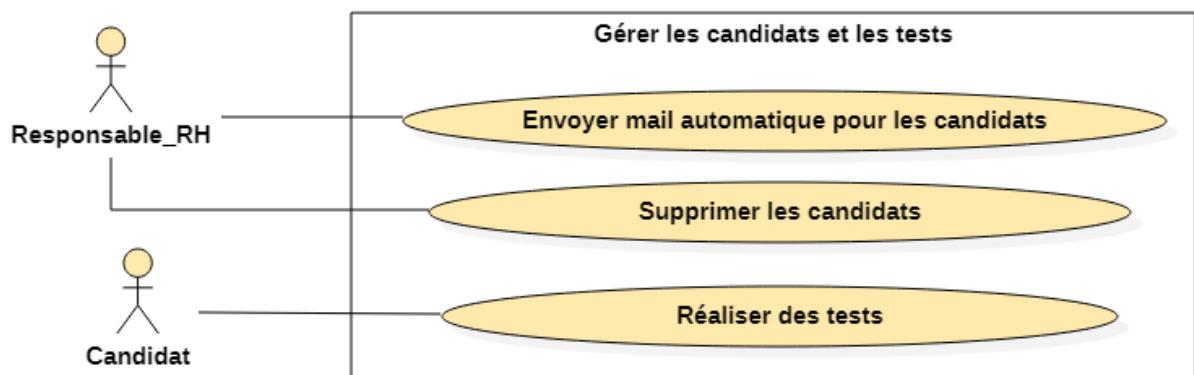


Figure 35: Cas d'utilisation du sprint 2.2

### 3.2.2. Raffinements des cas d'utilisation

#### Description textuelle du cas d'utilisation « Supprimer les candidats »

Le tableau 19 illustre un tableau récapitulatif du cas d'utilisation « Supprimer les candidats ». Nous décrivons dans ce tableau l'objectif, l'acteur principal de la fonctionnalité, le scénario nominal et les exceptions qui peuvent se produire.

Raffinement de cas d'utilisation	
<b>Titre</b>	Supprimer les candidats.
<b>Acteur</b>	Responsable des ressources humaines
<b>Objectif</b>	Supprimer les candidats inscrits à l'application et des tests.
<b>Résumé</b>	Le responsable des ressources humaines doit supprimer un candidat.
Conditions	
<b>Préconditions</b>	<b>Postconditions</b>
Le responsable des ressources humaines doit être connecté à l'application.	Le responsable des ressources humaines supprime les candidats postulés aux offres.
Scenario	
1- Le responsable des ressources humaines sera redirigé vers l'interface gestion des candidats. 2- Le système affiche la liste des candidats. 3- Le responsable des ressources humaines peut supprimer un candidat. 4- Le système affiche un message de succès.	
Exception	
Dans l'étape 3 du scénario nominal, si nous avons cliqué sur annuler, la suppression ne sera pas effectuée.	

Tableau 19: Description cas d'utilisation de suppression des candidats

### 3.3. Conception du sprint 2.2

Dans cette partie nous entamons la description de la partie conceptuelle. En premier lieu nous présentons le diagramme des classes de ce sprint et en second lieu nous montrons le diagramme de séquence objet du scénario "Gérer les candidats et les tests".

#### 3.3.1. Diagramme de classe du sprint 2.2

La figure 36 représente le diagramme de classe du sprint 2.2.

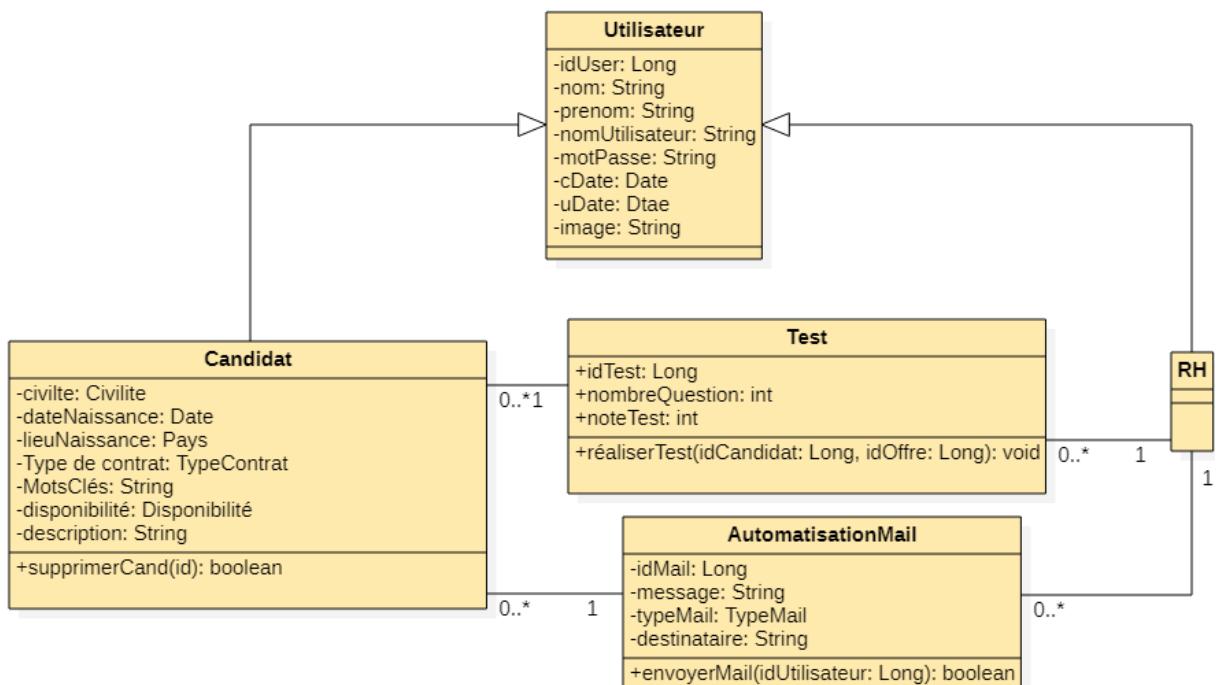


Figure 36: Diagramme de classe du sprint 2.2

Nous allons décrire les classes :

- Classe « *Test* » : Elle représente les tests réalisés par les candidats. Elle est définie par les attributs *idTest*, *nombreQuestion* et *noteTest* et par la méthode *réaliserTest(Long, Long) :void*
- Classe « *AutomatisationMail* » : C'est la classe relative aux responsables, notamment elle est dédiée pour l'envoi automatique des mails et la gestion des mails. Elle contient les attributs *idMail*, *message*, *typeMail*, *destinataire* et elle définit par la méthode *envoyerMail(Long) :Boolean*.
- Classe « *Candidat* » : elle est caractérisée par la méthode *supprimerCand(id) :Boolean*.

### 3.3.2. Diagramme de séquence du sprint 2.2

Le diagramme 40 représente le diagramme de séquence du sprint « Gérer les candidats et les tests ».

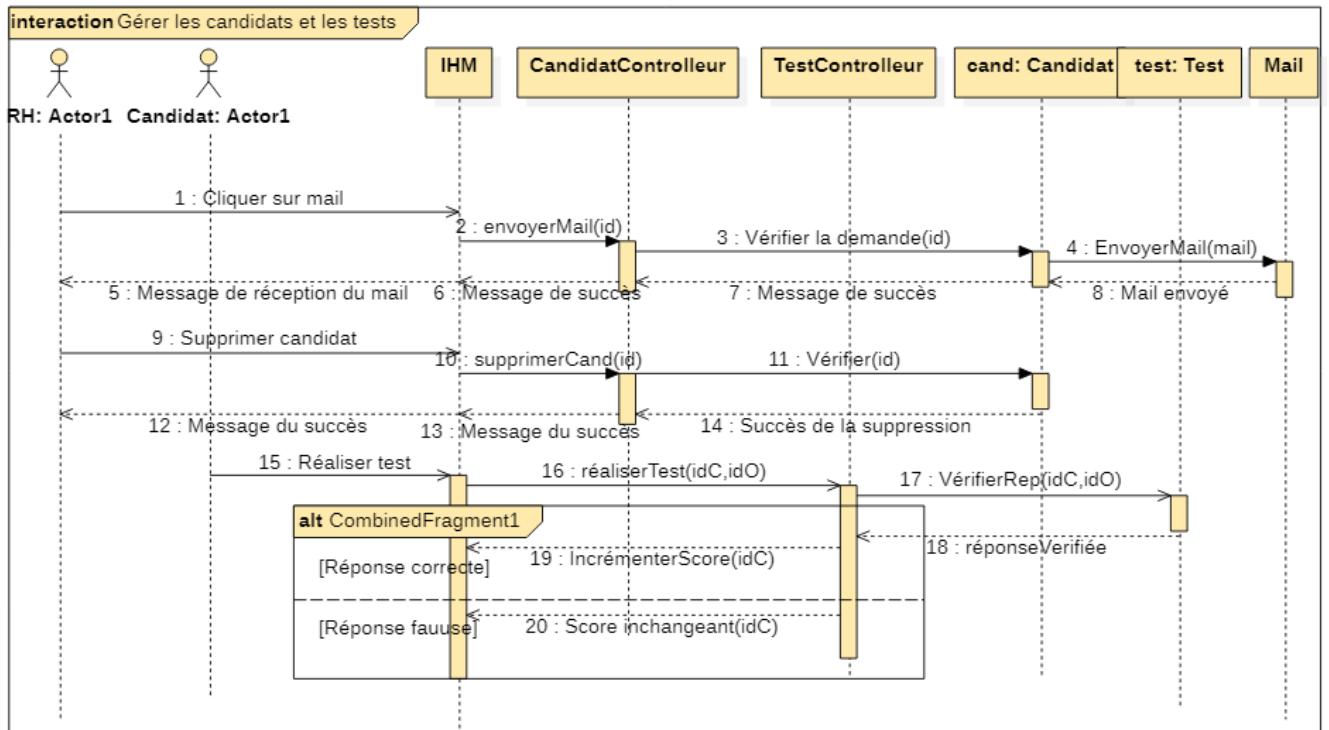


Figure 37: Diagramme de séquence du sprint 2.2

## 3.4. Réalisation du sprint 2.2

### 3.4.1. Interfaces de gestion des candidats et des tests

Dans cette partie, nous avons étudié les différentes interfaces relatives à la gestion des candidats ainsi que la gestion des tests.

En premier lieu, la figure 38 montre la suppression des candidats inscrits à l'application par le des ressources humaines.

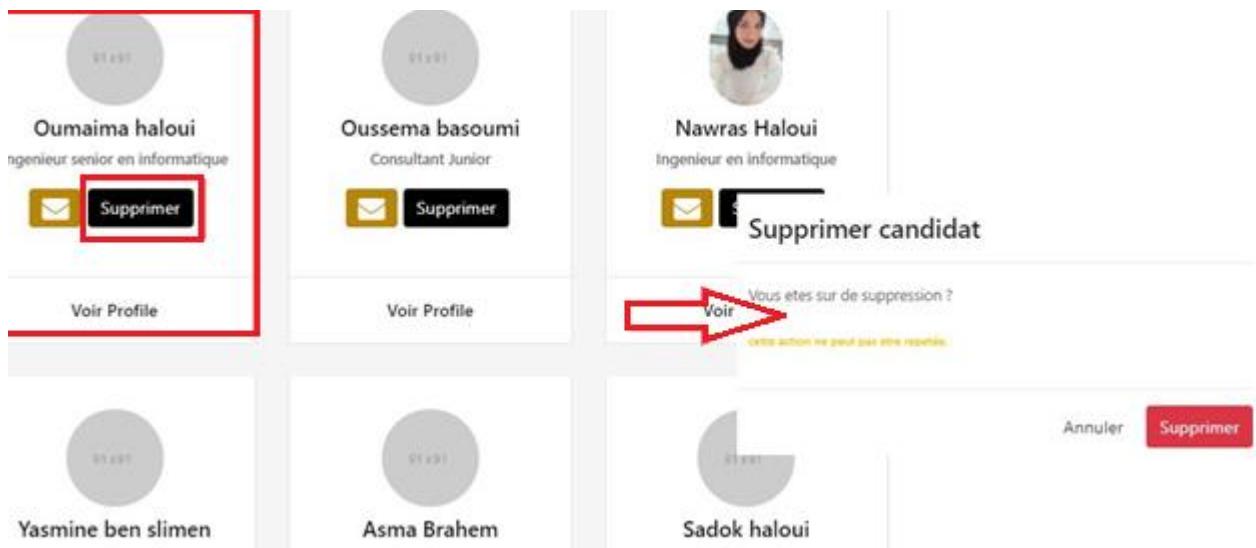


Figure 38: Suppression des candidats

Après la suppression du candidat « Oumaima », la nouvelle liste des candidats sera affichée dans la figure 39.

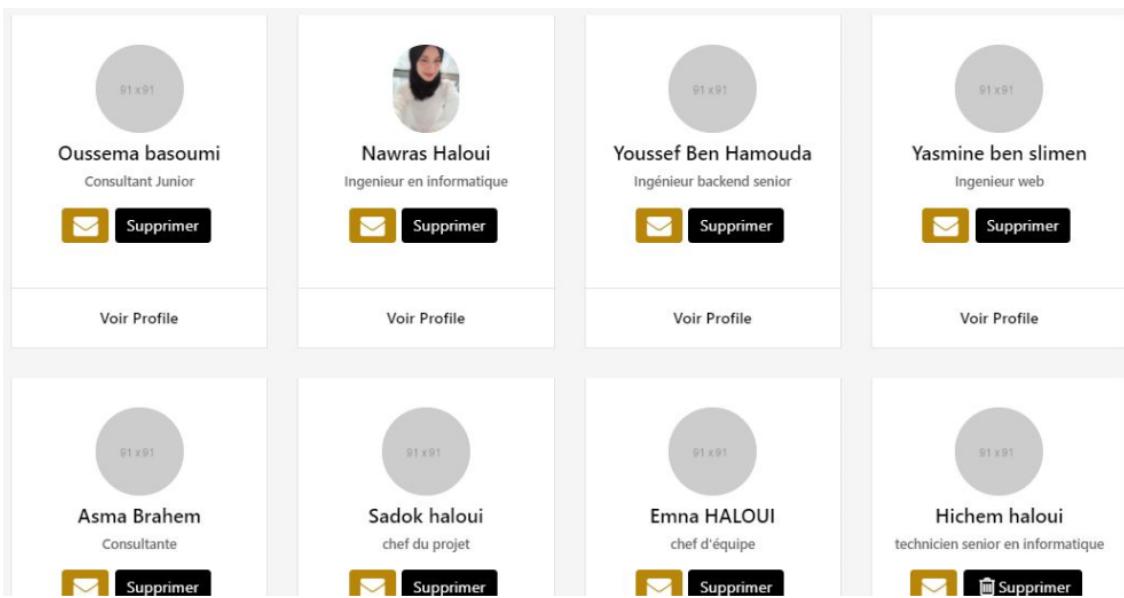


Figure 39: Suppression des candidats affectée

Par la suite, lorsque le responsable des ressources humaines veut visualiser les détails d'un candidat pour puisse envoyer un email de recrutement, il clique sur le bouton d'email et donc un mail sera envoyé automatiquement sur l'email du BFI de recrutement comme la montre la figure 40.

The screenshot shows a recruitment platform interface. At the top, there's a sidebar with contact information: phone (+216 55 66 44 96), email (utilisateur@gmail.com), LinkedIn (utilisateur-linkedin.com), and GitHub (utilisateur-github.com). Below this is a circular placeholder for a profile picture (170x170 pixels) with the text "Generated by HTML". The main area features a candidate profile for "Youssef Ben Hamouda" with a yellow "Test" button and a black "Entretien" button. A yellow bar at the bottom indicates the current step: "Analyse du son CV pour cette offre". To the right, there's a rating of four stars. A red arrow points down from the rating to the email preview below.

**Votre Candidature chez BFI**

**nawras test <testnawras2021@gmail.com>**

À moi : Bonjour .

17:42 (il y a 13 minutes) ★ ↗

Suite à votre candidature , nous souhaiterions vous rencontrer afin d'examiner votre candidature de manière plus complète et surtout plus conviviale.

Nous vous proposons un entretien xxxx à partir de xxH00.

Nous vous prions de confirmer, par e-mail, votre présence.

Cordialement

Figure 40: Envoi automatique des mails

### 3.4.2. Interfaces de gestion des tests

La réalisation des tests est l'une des étapes les plus importantes dans le processus de recrutement chez BFI.

La figure 43 montre la première interface de test pour les candidats.



Figure 41: Première interface du test

Dans figure 44 nous avons montré le cas d'une réponse correcte et d'une réponse incorrecte, sachant que si le chronomètre dépasse le temps limite, cette réponse sera comptabilisée comme fausse et à chaque réponse juste on affecte 1 et on incrémentale le score de 1.

The screenshot shows a web-based test interface. At the top, there's a header with "Test web" and a timer showing "Time Off 00". Below the header, the first question is displayed: "1. HTML quel sa signification?". Four options are listed: "Hyper Text Preprocessor" (light blue background), "Hyper Text Markup Language" (green background, indicating it is the correct answer), "Hyper Text Multiple Language" (light blue background), and "Hyper Tool Multi Language" (light blue background). The user has selected the correct answer. The interface includes navigation buttons "Suivant" (Next) and "Précédent" (Previous) at the bottom.

1 of 5 Questions

Time Left 14

Test web

1. HTML quel sa signification?

Hyper Text Preprocessor

Hyper Text Markup Language

Hyper Text Multiple Language

Hyper Tool Multi Language

Stylish Question Language

Stylesheet Query Language

Statement Question Language

Structured Query Language

Suivant

Figure 42: Réponses des candidats au test

A la fin du test, le résultat sera être affiché au responsable des ressources humaines qui indique le niveau du candidat : faible, moyen ou bien selon le score atteint comme le montre la figure 43.

The screenshot displays three separate feedback messages, each enclosed in a box:

- Feedback 1 (Faible):** "Vous avez terminé le test! Niveau Faible 😞, Vous avez seulement 1 sur 5". Buttons: "repeter le test" (blue) and "Quitter" (white).
- Feedback 2 (Moyen):** "Vous avez terminé le test! Moyen 😌, Vous avez 3 sur 5". Buttons: "repeter le test" (blue) and "Quitter" (white).
- Feedback 3 (Bien):** "Vous avez terminé le test! Congratulation! 🎉, Vous avez 5 sur 5". Buttons: "repeter le test" (blue) and "Quitter" (white).

Figure 43: Niveau des candidats dans le test

## 4. Sprint 2.3 : Gérer les entretiens

### 4.1. Backlog du sprint 2.3

Le tableau 20 représente le backlog de notre sprint « Gérer les entretiens » où nous listons les différentes exigences à mettre en place dans cette section.

ID Module	Module	Id user Story	User story
1	<b>Gérer les entretiens</b>	1.1	En tant qu'un responsable des ressources humaines, je peux ajouter un entretien au calendrier
		1.2	En tant qu'un responsable des ressources humaines, je peux supprimer une date du calendrier.
		1.3	En tant qu'un responsable des ressources humaines, je peux modifier une date.

Tableau 20: Backlog du sprint 2.3

### 4.2. Analyse des besoins du sprint 2.3

Après avoir défini le backlog du sprint, nous nous intéressons à la partie analyse et nous commençons par la présentation du diagramme de cas d'utilisation, ensuite nous générerons un diagramme de séquence système.

#### 4.2.1. Diagramme de cas d'utilisation du sprint 2.3

La figure 47 illustre le diagramme des cas d'utilisation du sprint « Gérer les entretiens » qui décrit les fonctionnalités à envisager dans cette partie.

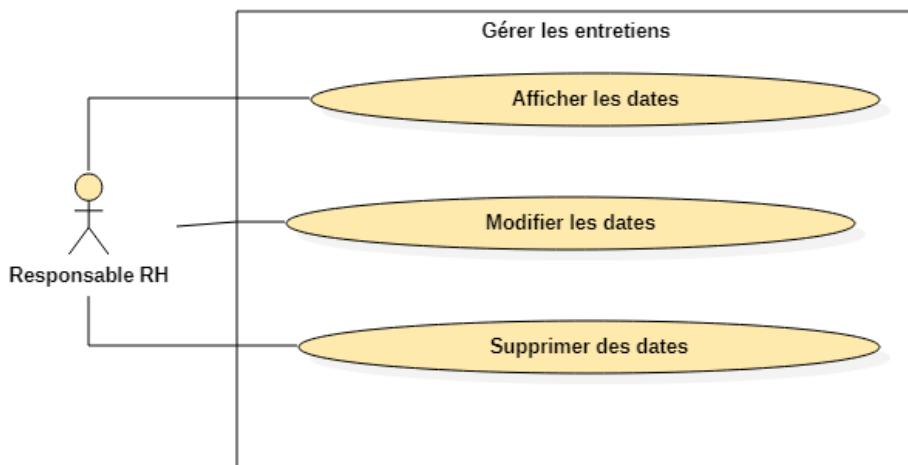


Figure 44: Diagramme de cas d'utilisation du sprint 2.3

## Raffinements des cas d'utilisation

### Description textuelle du cas d'utilisation « Ajouter les dates »

Le tableau 21 illustre une description du cas d'utilisation « Ajouter les dates ». Nous décrivons dans ce tableau l'objectif, l'acteur principal de la fonctionnalité, le scénario nominal et les exceptions qui peuvent se produire.

Raffinement de cas d'utilisation	
<b>Titre</b>	Ajouter les dates
<b>Acteur</b>	Responsable des ressources humaines
<b>Objectif</b>	Gérer les dates des entretiens.
<b>Résumé</b>	Le responsable des ressources humaines doit ajouter les dates d'entretiens.
Conditions	
<b>Préconditions</b>	<b>Postconditions</b>
Le responsable des ressources humaines doit être connecté à l'application.	Le responsable des ressources humaines ajoute les dates d'entretiens avec succès.
Scenario	
1- Le responsable des ressources humaines sera redirigé vers l'interface du calendrier pour fixer les dates d'entretiens après avoir connecté à l'application. 2- Le système affiche la liste des dates déjà mise. 3- Le responsable des ressources humaines ajoute des dates. 4- Le système affiche un message de succès.	
Exception	
Dans l'étape 3 et 4 du scénario nominal, si un domaine ne contient pas un évènement, le système affiche un message qu'il n'y a pas des dates.	

Tableau 21: Description cas d'utilisation « Ajouter les dates »

## 4.3. Conception du sprint 2.3

Dans cette partie nous entamons la description de la partie conceptuelle. En premier lieu nous présentons le diagramme des classes de ce sprint et en second lieu nous montrons le diagramme de séquence objet du scénario "Gérer les entretiens".

### 4.3.1. Diagramme de classe du sprint 2.3

La figure 45 illustre le diagramme de classe de « Gérer les entretiens ».

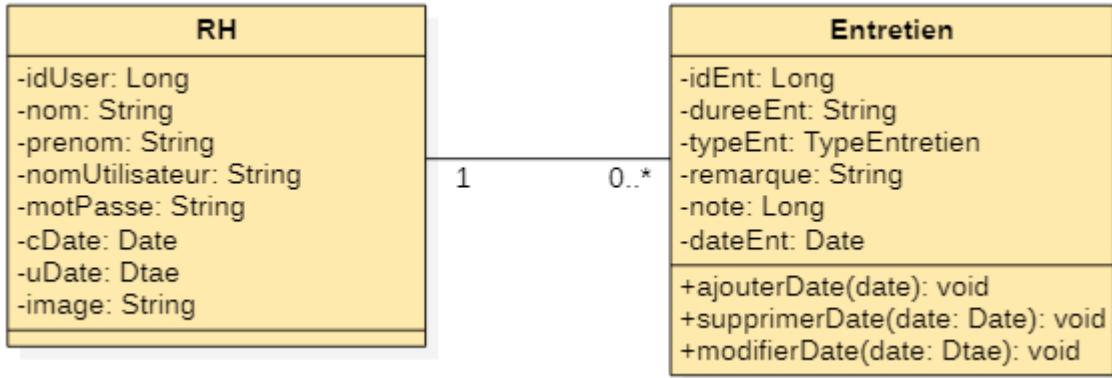


Figure 45: Diagramme de classe du sprint 2.3

- Classe « *Entretien* » : Elle représente le calendrier et les dates des entretiens RH et entretiens techniques. Elle est définie par les attributs *idEnt*, *dureeEnt*, *typeEnt*, *remarque*, *note*, *dateEnt* et par les méthodes *ajouterDtae(date) :void*, *supprimerDtae(date) :void* et *modiferDtae(date) :void*.

#### 4.3.2. Diagramme de séquence du sprint 2.3

Le diagramme 46 représente le diagramme de séquence du sprint « Gérer les entretiens ».

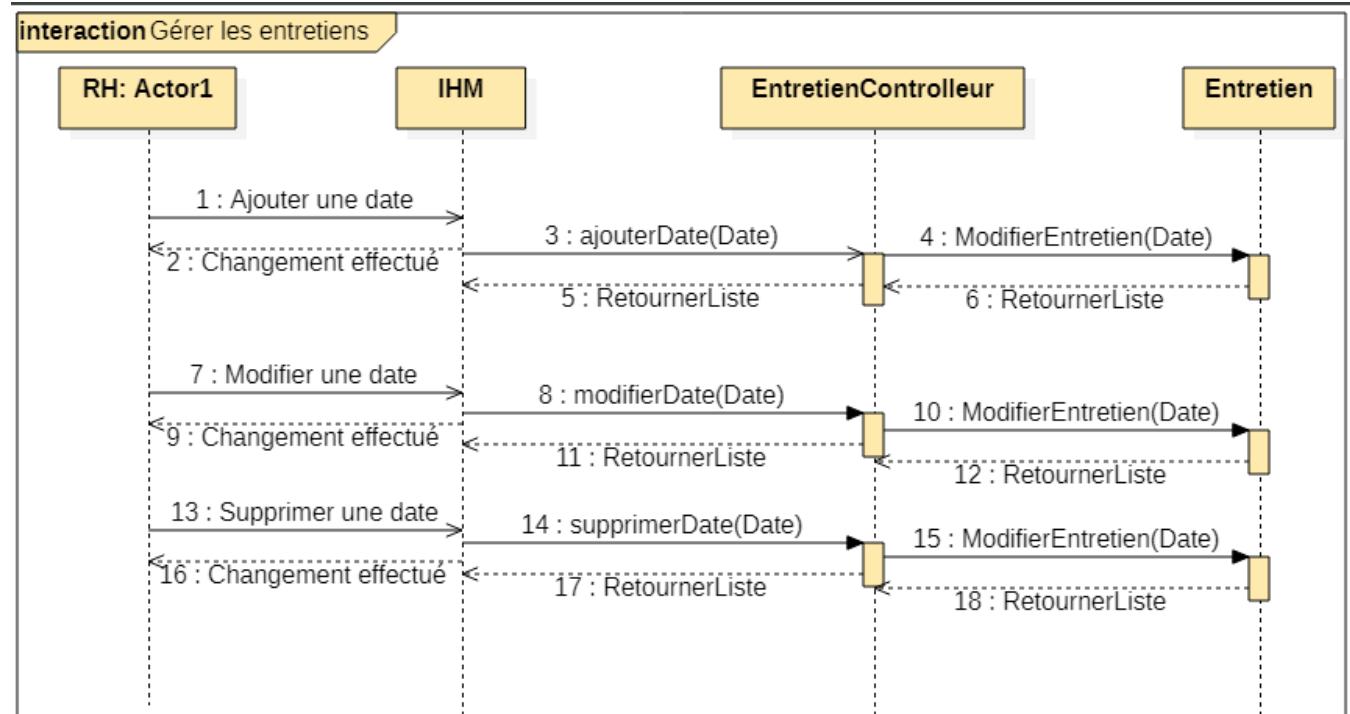


Figure 46: Diagramme de séquence du sprint 2.3

## 4.4. Réalisation du sprint 2.3

Dans cette partie, nous allons étudier les interfaces de gestion des entretiens.

### 4.4.1. Interfaces de gestion des entretiens

Après avoir choisi les candidats adéquats aux offres, il est indispensable pour le responsable des ressources humaines de fixer les dates d'entretiens pour continuer le processus de recrutement chez BFI.

La figure 50 montre l'ajout des dates des entretiens au calendrier.

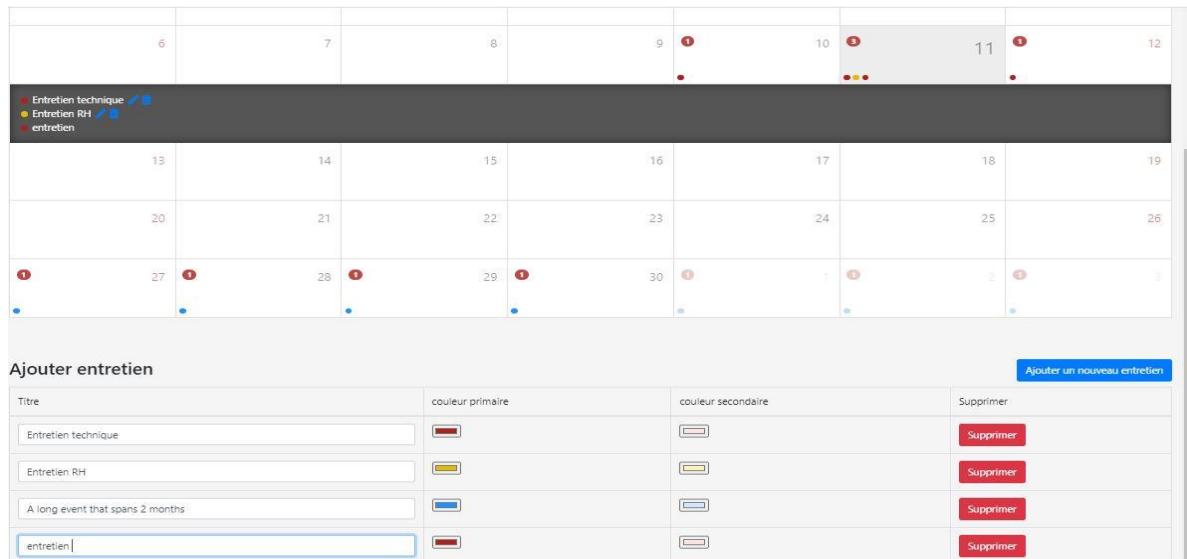


Figure 47: Ajout des dates d'entretien

De même, il peut modifier ou supprimer comme l'indique la figure 48.

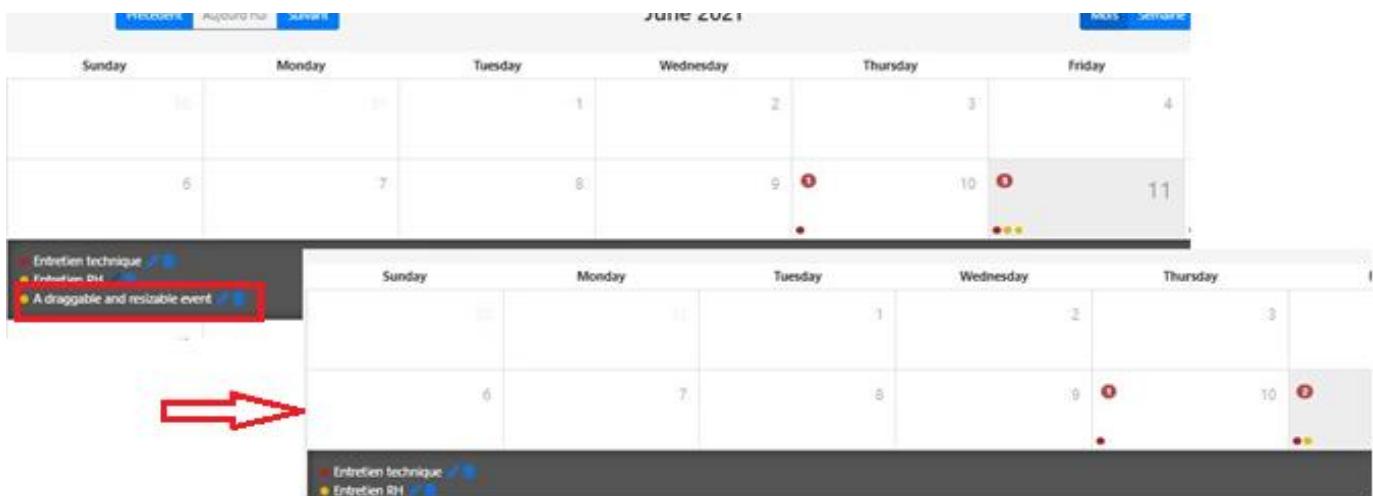


Figure 48: Suppression des dates d'entretien

## **Conclusion**

Au cours de ce chapitre, nous avons détaillé la troisième release tout en partant du backlog du produit et en finissant par le résultat. Les phases intermédiaires ont aussi été présentées de l'analyse à la conception. Le prochain chapitre introduira la dernière itération dans le cadre de cette mission et qui couvrira une phase critique du sujet. Elle est très importante pour la détection des candidats adéquats aux offres et résume la partie de virtualisation et conteneurisation de l'application.

# Chapitre 5 : Release 3 « Analyse des CV »

## Introduction

Le présent chapitre consiste à présenter le travail réalisé durant la release 3. Tout au long de ce chapitre nous aurons deux sprints. Lors de chaque sprint nous commençons par présenter le backlog. Ensuite nous entreprenons la spécification des besoins fonctionnels, leur modélisation via les diagrammes UML et enfin la réalisation.

## 1. Planification de la release 3

Notre dernière release se compose de deux sprints, le tableau 22 décrit leurs planifications.

Sprint 3.1: Analyser les CV par compétences	Sprint 2.3 : Analyser les CV par niveau de la langue
<ul style="list-style-type: none"><li>• Sélectionner les compétences dans les CV des candidatures qui correspond à chaque offre.</li><li>• Filtrer les candidatures des candidats les plus compétents</li><li>• Evaluer les candidatures. (De 21/04/2021 à 19/05/2021)</li></ul>	<ul style="list-style-type: none"><li>• Analyser les CV par niveau de la langue.</li><li>• Evaluer les candidatures. (De 20/05/2021 à 15/06/2021)</li></ul>

Tableau 22: Planification des sprints de release 3

## 2. Système d'analyse des CV

Pour aboutir à exploiter un système qui répond aux besoins du BFI, notamment un système qui sert à analyser les CV, nous avons conçus à développer tout un modèle avec l'intelligence artificielle qui sert à filtrer les candidats les plus adéquats pour chaque offre.

Cependant, le développement du système a été segmenté en des étapes comme l'indique la figure 49.

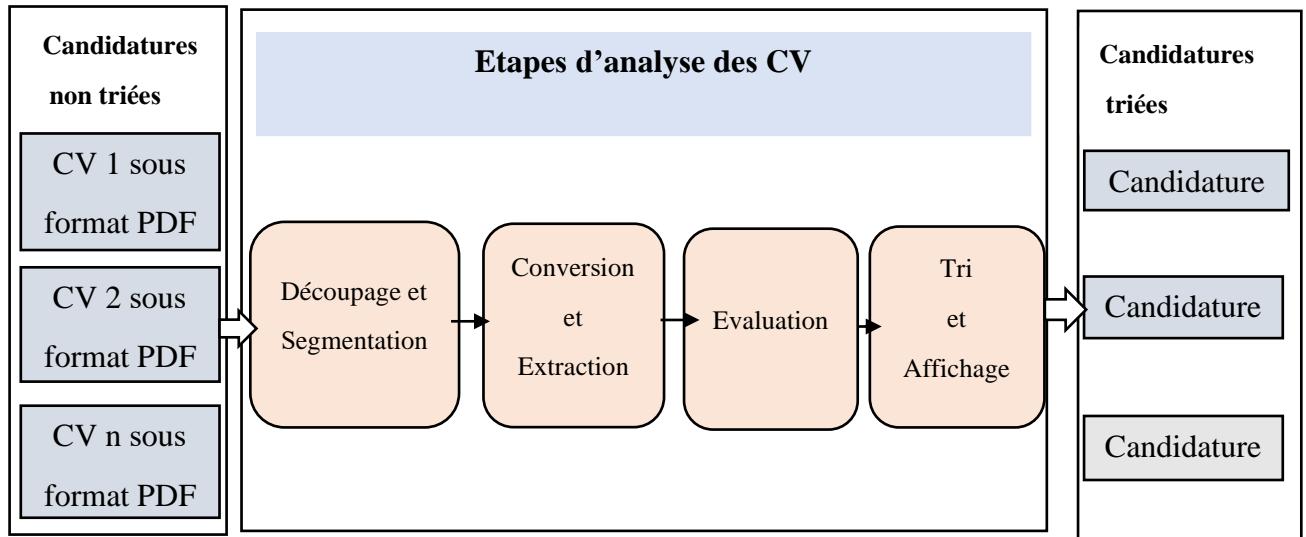


Figure 49: Modèle d'analyse des CV

## 2.1.Découpage et segmentation en section d'un CV

La première tâche consiste à repérer, découper et regrouper les sections pertinentes des CV. Elle comporte deux étapes à savoir le découpage et la segmentation.

### 2.1.1. Découpage

Cette étape de *découpage* consiste à faire un découpage à des tailles fixes. Notre objectif étant de découper les CV de manière fine en se basant la structure du CV : *les titres, les sous-titres* ou les débuts des lignes avec un symbole délimitant une section.

La figure 50 explique cette étape.

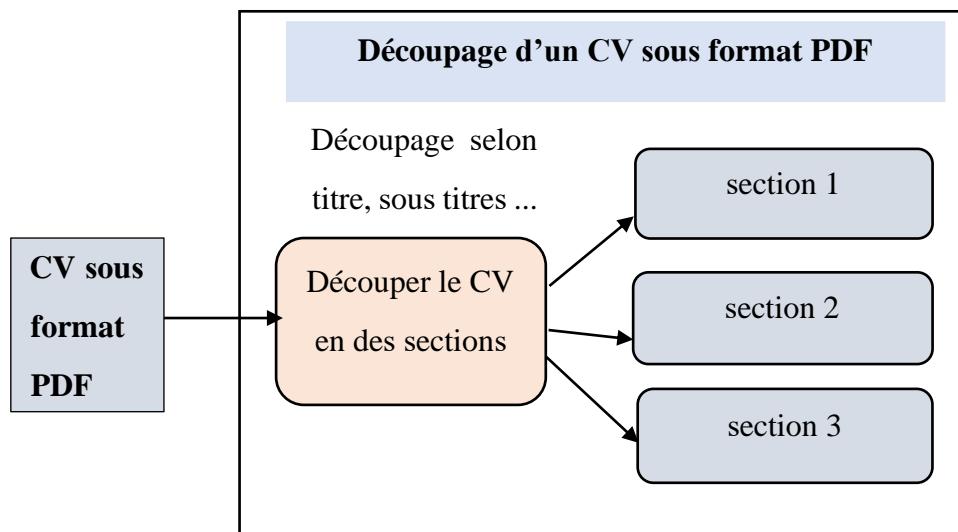


Figure 50: Etape de découpage d'un CV

### 2.1.2. Segmentation

L'étape de *segmentation* essaie d'améliorer le découpage au moyen de mots-clés qui seront recherchés à l'intérieur des sections. Le découpage fait appel à un prétraitement (élimination ou normalisation de symboles et la normalisation d'espaces), puis, les règles de structure sont appliquées. Après ce découpage, nous vérifions la taille des sections trouvées. Par exemple, si un fragment de texte dans la section « *Compétences* » contient « les mots célibataire » ou « situation de famille » ce fragment sera déplacé au segment « *Identité* ».la figure 51 explique cette étape.

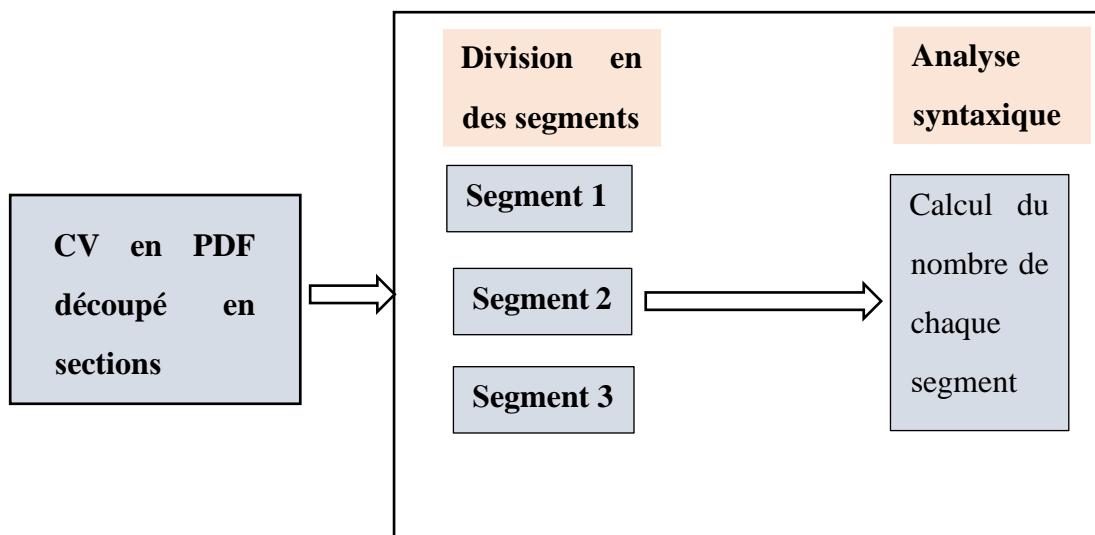


Figure 51: Etape de segmentation d'un CV

L'*analyse syntaxique* du texte permet de calculer le nombre de segments.

A titre d'exemple, prenons le cas des compétences dans un CV, l'analyse permet de récupérer les mots les plus importants colorés en vert par exemple, à travers la partie des compétences découpée du CV et qui figure dans la « Section des compétences », nous avons calculés le nombre des mots qui détermine les compétences tels que « Skills » et « Proficiencies » dans la figure 52.



Figure 52: Analyse syntaxique des compétences

## 2.2. Conversion et Extraction

A ce niveau, nous allons étudier le traitement d'analyse intitulé « Conversion et extraction » qui consiste à extraire les informations nécessaires pour pouvoir évaluer les candidatures par la suite.

### 2.2.1. Conversion

Cette étape consiste à convertir les CV du format PDF en des fichiers .txt en faisant appel à des bibliothèques prédéfinies dans Python comme l'illustre la figure 53.

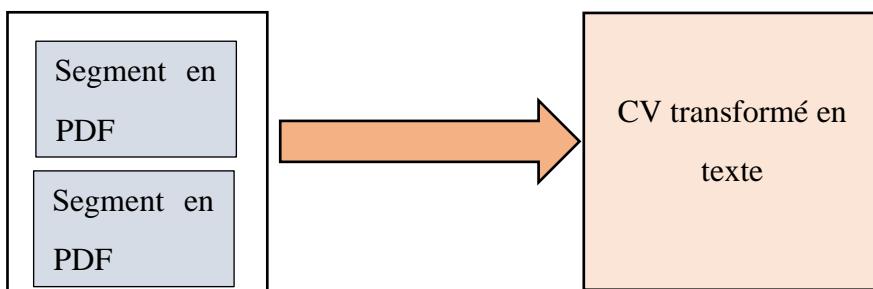


Figure 53: Etape de conversion d'un CV

### 2.2.2. Extraction

Afin d'extraire les données utiles à partir d'un CV, nous avons développé un algorithme qui se base sur le traitement du langage naturel (NLP) et l'analyse de texte, et donc implique l'utilisation d'algorithmes d'apprentissage automatique et d'une intelligence artificielle (IA) « étroite » pour comprendre la signification des CV en documents texte.

Ces documents contiennent du texte à savoir : des commentaires sur les réseaux sociaux, des critiques en ligne, des réponses à des sondages, voire des documents financiers, médicaux, juridiques et réglementaires. Essentiellement, le rôle de l'apprentissage automatique et l'IA dans le traitement du langage naturel et analyse de texte est d'améliorer, d'accélérer et d'automatiser les sous-jacentes analyse de texte fonctions et la PNL dispose ce tour ce texte non structuré en données utilisables et des idées [21].

Le traitement automatique des langues effectue une analyse qui peut se découper en plusieurs étapes. Chacune de ces étapes de traitement prend en entrée le résultat de l'analyse précédente et fournit en sortie un résultat contenant les informations obtenues par la suivante. La figure 54 montre les étapes du traitement NLP.

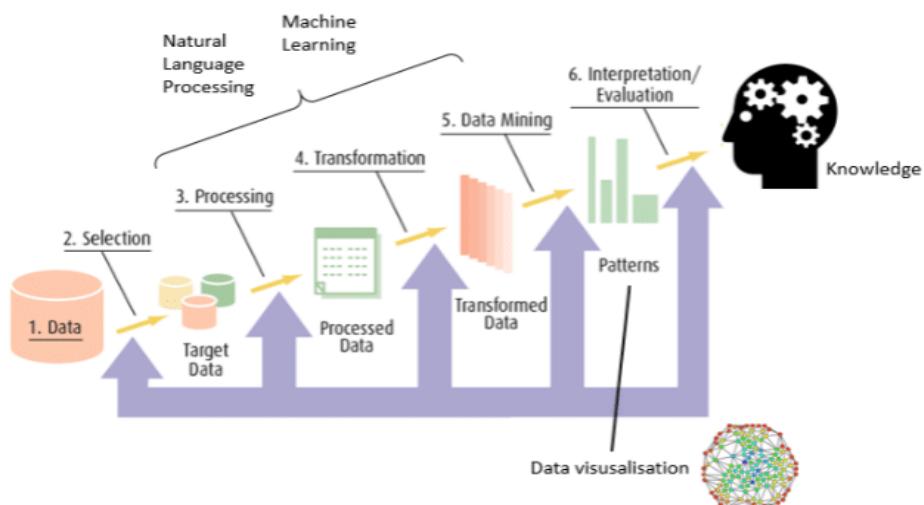


Figure 54: Étapes du traitement NLP [22]

### 2.3. Évaluation

Cette étape consiste à évaluer les données des CV et calculer les scores.

Dans la figure 55, nous trouverons les détails de cette étape.

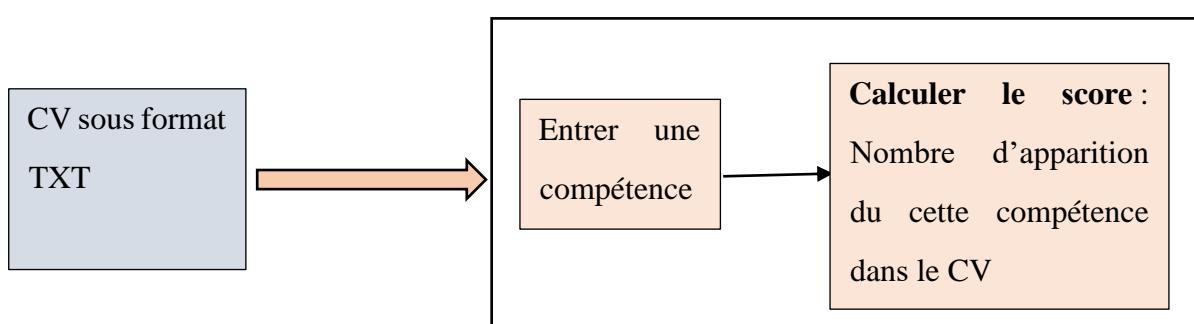


Figure 55: Etape d'évaluation d'un CV

## 2.4.Affichage et Tri

En fonction des scores calculés, il y a lieu de trier les candidatures des candidats de chaque offre selon leurs compétences ou bien selon le niveau de la langue.

Les traitements effectués dans cette étape seront clarifiés dans la figure 56.

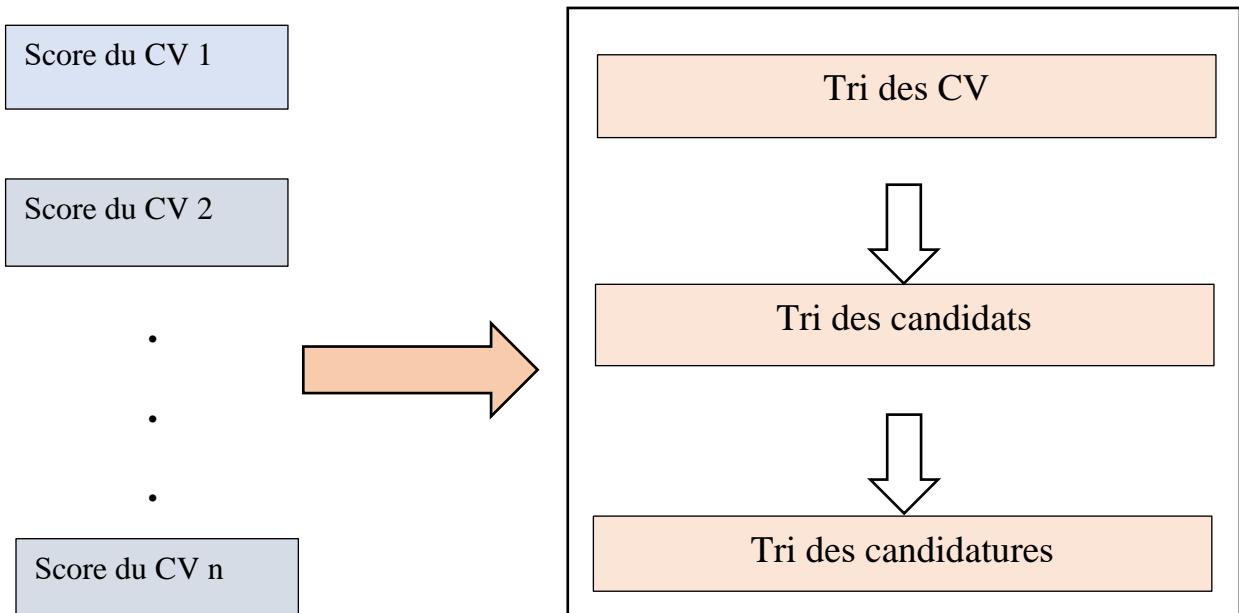


Figure 56: Etape de tri des CV

## 3. Sprint 3.1 : Analyser les CV par compétence

### 3.1. Backlog du sprint 3.1

Le tableau 23 représente le backlog de notre sprint " Analyser des CV par compétence " où nous listons les différentes exigences à mettre en place dans cette section.

ID Module	Module	Id user Story	User story
1	Analyser les CV par compétence	1.1	En tant qu'un responsable des ressources humaines, sélectionner une offre.
		1.2	En tant qu'un responsable des ressources humaines, je peux sélectionner une compétence.
		1.3	En tant qu'un responsable des ressources humaines, je peux afficher les candidatures triées par compétence.
		1.4	Le modèle va calculer le score pour chaque candidature.
		1.5	Le modèle va extraire les données d'un CV.

Tableau 23: Backlog du sprint 3.1

## 3.2. Analyse des besoins du sprint 3.1

Dans cette partie nous allons traiter le diagramme de cas d'utilisation ainsi que le diagramme de séquence qui contient les fonctions essentielles pour ce sprint.

### 3.2.1. Diagramme de cas d'utilisation

La figure 57 illustre le diagramme du cas d'utilisation du sprint " Analyser les CV par compétence " qui contient toutes les fonctionnalités à réaliser.

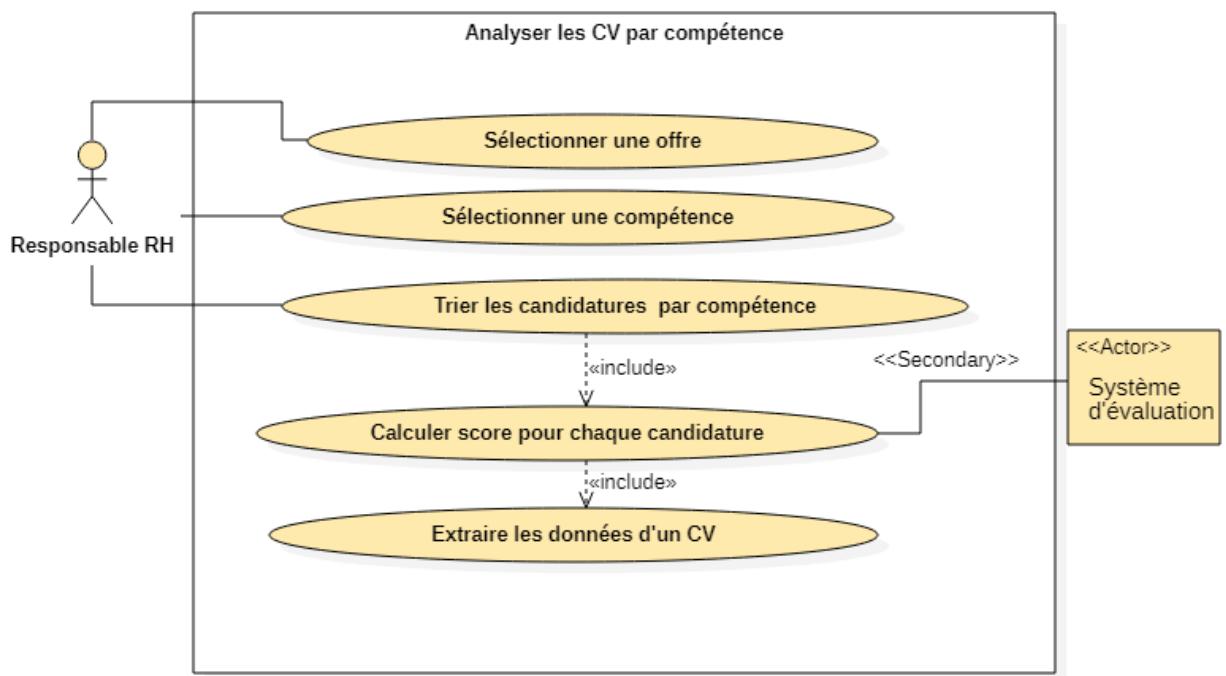


Figure 57: Diagramme de cas d'utilisation du sprint 3.1

### 3.2.2. Raffinements des cas d'utilisation

#### Description textuelle du cas d'utilisation « Analyser les CV par compétence »

Le tableau 24 illustre un tableau récapitulatif du cas d'utilisation « Analyser les CV par compétence ». Nous décrivons dans ce tableau l'objectif, l'acteur principal de la fonctionnalité, le scénario nominal et les exceptions qui peuvent se produire.

Raffinement de cas d'utilisation	
<b>Titre</b>	Analyse les CV par compétence
<b>Acteur</b>	Responsable des ressources humaines
<b>Objectif</b>	Trier les candidatures les plus performants à travers la sélection de la compétence par le responsable des ressources humaines.

<b>Résumé</b>	Le responsable des ressources humaines doit choisir la compétence pour trier les candidatures de chaque offre.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Postconditions</b>
<ul style="list-style-type: none"> <li>- Le responsable des ressources humaines doit être connecté à l'application.</li> <li>- Présence des candidatures relatives à l'offre désirée.</li> </ul>	Le responsable des ressources humaines choisit la compétence et les candidatures des candidats vont être triées.
<b>Scenario</b>	
<ol style="list-style-type: none"> <li>1- Le responsable des ressources humaines sera redirigé vers l'interface gestion des offres après avoir connecté à l'application.</li> <li>2- Le système affiche la liste des offres.</li> <li>3- Le responsable des ressources humaines visualise les candidatures de chaque offre.</li> <li>4- Le responsable des ressources humaines sélectionne une compétence pour trier les candidatures.</li> <li>5- Le système affiche la liste des candidatures triées selon le niveau des candidats dans cette compétence.</li> </ol>	
<b>Exception</b>	
<p>Dans l'étape 3 : - Si une offre ne contient des candidatures, le processus d'analyse des compétences à travers les CV ne se produit pas.</p> <p>Dans l'étape 4 : - Si le responsable des ressources humaines ne sélectionne pas une compétence, le processus d'analyse des compétences ne se produit pas.</p>	

Tableau 24: Description cas d'utilisation : Analyse des CV

### 3.3. Conception du sprint 3.1

Dans cette étape, nous présentons le diagramme des classes de ce sprint et en second lieu nous montrons le diagramme de séquence « Analyser les CV par compétence ».

#### 3.3.1. Diagramme des classes de conception du sprint 3.1

Le diagramme 58 présente le diagramme de classe du premier sprint « Analyser les CV par compétence ». Nous présentons dans cette partie les classes et les relations entre elles.

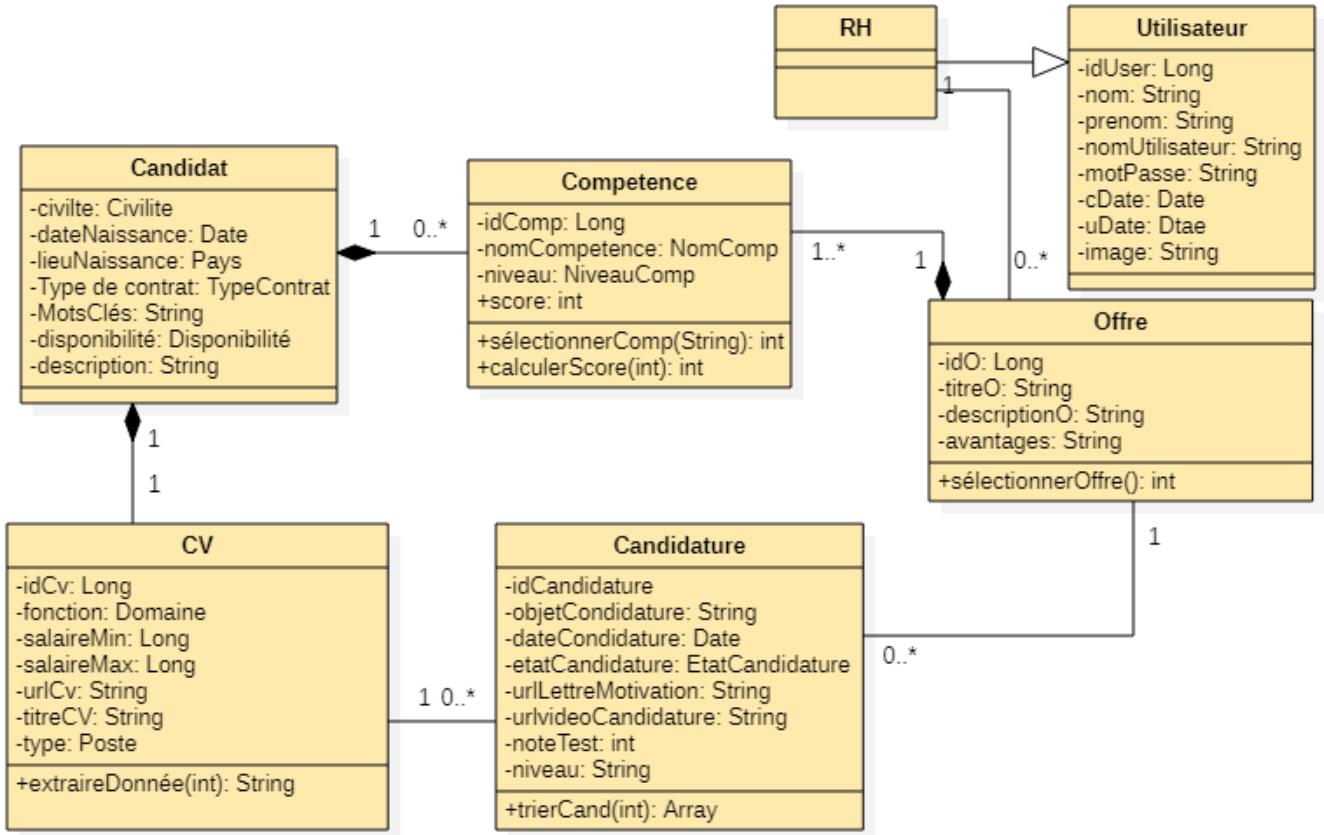


Figure 58: Diagramme de classe de sprint 3.1

Dans ce sprint nous allons décrire les classes suivantes :

- Classe « *CV* » : C'est la classe qui contribue à la sélection adéquate des candidatures pour chaque offre grâce à l'analyse selon plusieurs critères fixés par le responsable des ressources humaines. Elle est caractérisée par les attributs *idCv*, *fonction*, *salaireMax*, *urlCV*, *titreCV*, *type* et *score*. Elle admet la méthode *extraireDonnée(int) :String*.
- Classe « *Compétence* » : Elle est caractérisée par les attributs *idComp*, *nomCompetence*, *niveau* et *score*. Cette classe est définie par les méthodes *sélectionnerComp(steing) :int* et *calculerScore(int) :int*.
- Classe « *Candidature* » : Elle contient les attributs *idCandidature*, *dateCandidature*, *objetCandidature*, *noteTest* et *niveau*, et elle admet la méthode *trierCand(int) :Array*.
- Classe « *Offre* » : elle est définie par les attributs *idO*, *titreO*, *descriptionO*, *avantages* et par la méthode *sélectionnerOffre() :int*.

### 3.3.2. Diagramme de séquence du sprint 3.1

Le diagramme 59 montre le diagramme de séquence de sprint « Analyser les CV par compétence ».

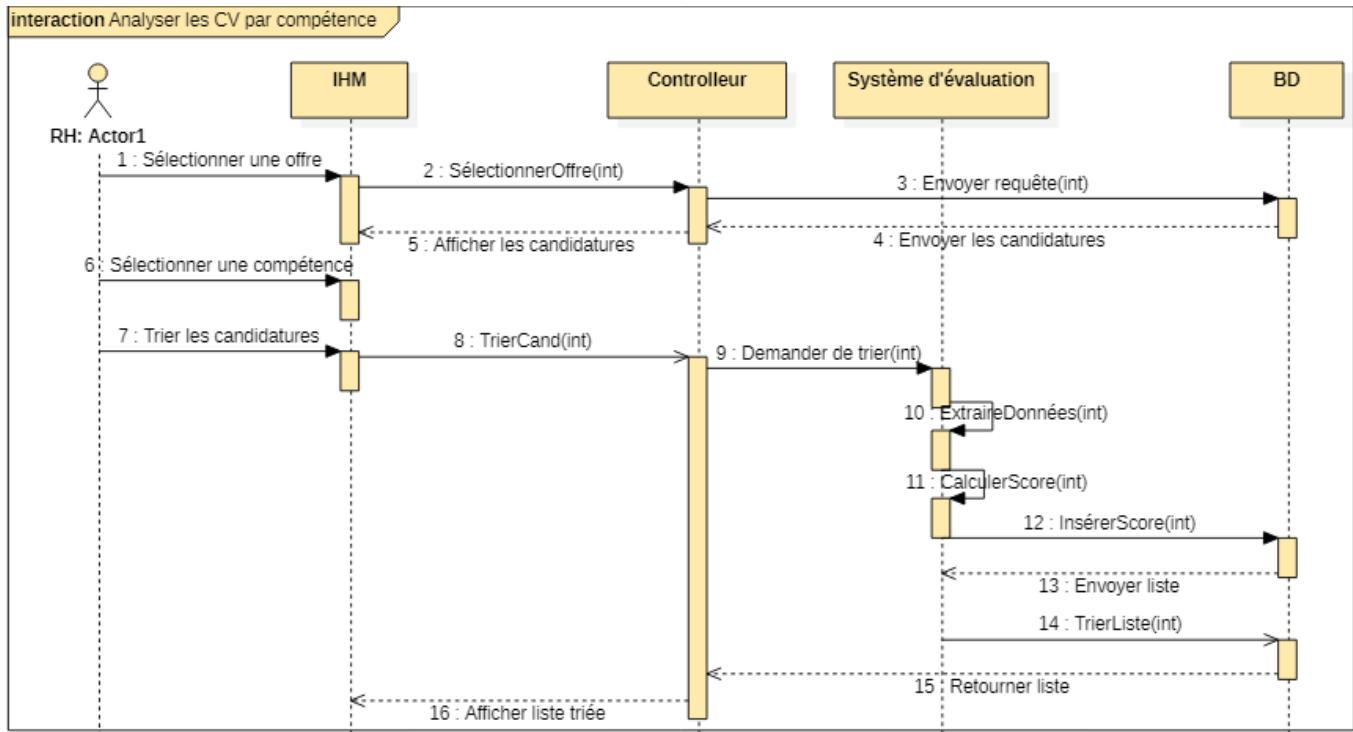


Figure 59: Diagramme de séquence d'analyse des CV

### 3.4. Réalisation du sprint 3.1

Dans cette partie, nous allons détailler les étapes de réalisation de notre Analyse les CV par compétences en passant par les quatre étapes de « *Découpage et segmentation* », « *Conversion et Extraction* », « *Evaluation* » et finalement « *Tri et affichage* ».

Les figures 60 et 61 montrent le code qui traite cette analyse.

```

1.from flask import Flask, jsonify, request
2.from pdf2image import convert_from_path
3.import easyocr
4.import numpy as np
5.import PIL
6.from PIL import ImageDraw
7.import spacy
8.import cx_Oracle
9.from flask_cors import CORS
10.import requests
11.from textblob import TextBlob
12.from spellchecker import SpellChecker
13.from pathlib import Path, PureWindowsPath
14.import pdf2image
15.import re
16.app = Flask(__name__)
17.cors = CORS(app, resources={r"/api/*": {"origins": "*"}})
18.reader = easyocr.Reader(['en'])
19.poppler_path='C:\poppler-0.68.0\bin'
20.from IPython.display import display, Image
21.conn = cx_Oracle.connect("bficv_def", "sa", "localhost:1521/orcl")
22.print(conn.version)
23.cur= conn.cursor()

```

## #Importation des bibliothéques et connexion à la base de données

```

@app.route("/api/analyseComp/<name>/<idapp>", methods=['GET'])
1.def compCV(name,idapp):
connexion à la base
2. curr= conn.cursor()

#Extraire les CV qui ont cette compétence
3.images = curr.execute("select cvv.cv from t_cv cvv,t_application ap,t_candidat cd where ap.candidata_id=cd.id and cd.id=cvv.candidat_id and ap.id_app=:idA", idA=idapp)
6. b = curr.execute("select cvv.cv from t_cv cvv,t_application ap,t_candidat cd where ap.candidata_id=cd.id and cd.id=cvv.candidat_id and ap.id_app=:idA", idA=idapp).fetchall()

```

```

#Découpage et segmentation
9. for bound in bounds:
10.     p0, p1 ,p2, p3 = bound[0]
11.     # draw.line([*p0, *p1, *p2, *p3, *p0], fill='yellow', width=2)
12.     text=''
13.     for i in range(len(bounds)):
14.         text = text +bounds[i][1]

```

## #Conversion

```

15. a=0
16. i=0
17. while i<len(text):
18.     if text[i]==" ":
19.         i= i+1
20.     elif text[i:i+len(name)].upper()==name.upper():
21.         a=a+1
22.         print(text[i:i+len(name)])
23.         i=i+len(name)
24.     else:
25.         i=i+1
26.     try:
27.         cur.execute("update t_skill sk SET sk.cv_comp=:note WHERE (select ap.id_app from t_application ap ,t_candidat cd where ap.candidata_id=cd.id and cd.id=sk.candidat_id )=:idA and sk.skill=:nomcomp ", note=a, idA=idA, nomcomp=name)
31.         conn.commit()
32.         print("score du cv : " +str(a)+ " de la compétence :" + str(name))
33.     except:
34.         print("erreur dans l'insertion dans la table comp")
35.     return ("note en "+str(name)+" est : "+ str(a))

```

## #Fonction d'analyse des CV par une compétence donnée

Figure 60: Code d'analyse des CV par compétence (1)

# Fonction de tri des  
candidatures de chaque  
offre (**Evaluation**)

```

38.@app.route('/api/getallApp/<name>/<idOffer>', methods=['GET'])
39.def api_all(name,idOffer):
40. c=currr.execute("select id_app from t_application where offer_id_offer=:41. idA", idA=idOffer)
..fetchall()
42. x = str(c)
43. y=x[2:len(x)-2]
44. b=y.replace(", \" ")
45. a=b.replace(')',"")
46. list=a.replace(',', "")
47. list=list.replace(" ", "")
48. l=str(list)
49. # return("laaaa"+l)
50. l=y[1:len(l)-1]
51. print("pour cette offre on a "+str(len(list)) +" candidatures" )
52. # return list[0]
53. # l=list[1,len(list)-1]
54. for j in range(len(list)) :
55.     # print(j )
56.     id= list[j]
57.     print("l'id de la candidature est : "+id)
58.     #return int(id)
59.     compCV(name, int(list[j]))
60. b=cur.execute("select app.* from t_application app, t_candidat cd ,t_sk61.ill s  where cd.id=
s.candidat_id and cd.id=app.candidata_id and s.skill =62.: nomcomp and app.offer_id_offer= : id o
rder by s.cv_comp desc", nomcomp663.=name, id=idOffer).fetchall()
64. print("tri de la compétence est effectué avec succès!!")
65. return jsonify(b)

```

Figure 61: Code d'analyse des CV par compétence (2)

### 3.4.1. Découpage et segmentation en section d'un CV

La première étape pour analyser un CV nécessite de découper un CV pour puisse segmenter et l'analyser par la suite.

Dans cette partie nous allons donc étudier l'étape de découpage et segmentation d'un CV.

#### Découpage

La figure 62 représente le code du découpage d'un CV en sections.

```
>pip install poppler-utils
Requirement already satisfied: poppler-utils in c:\users\sadok\anaconda3\lib\site-packages (0.1.0)
Requirement already satisfied: Click>=7.0 in c:\users\sadok\anaconda3\lib\site-packages (from poppler-utils) (7.1.2)
Note: you may need to restart the kernel to use updated packages.

!pip install pdf2image
!pip install easyocr

Requirement already satisfied: pdf2image in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (1.14.0)
Requirement already satisfied: pillow in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from pdf2image) (8.2.0)
Requirement already satisfied: easyocr in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (1.3.0.1)
Requirement already satisfied: scipy in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (1.6.2)
Requirement already satisfied: python-bidi in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (0.4.2)
Requirement already satisfied: numpy in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (1.20.2)
Requirement already satisfied: opencv-python in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (4.5.1.48)
Requirement already satisfied: torchvision>=0.5 in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (0.9.1)
Requirement already satisfied: PyYAML in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (5.4.1)
Requirement already satisfied: Pillow in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (8.2.0)
Requirement already satisfied: scikit-image in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (0.18.1)
Requirement already satisfied: torch in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from easyocr) (1.8.1)
Requirement already satisfied: typing-extensions in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from torch->easyocr) (3.7.4.3)
Requirement already satisfied: six in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from python-bidi->easyocr) (1.12.0)
Requirement already satisfied: networkx>=2.0 in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from scikit-image->e
    ... --snip-->

!pip install poppler-utils
Requirement already satisfied: poppler-utils in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (0.1.0)
Requirement already satisfied: Click>=7.0 in c:\users\sadok\anaconda3\envs\cvextract\lib\site-packages (from poppler-utils) (7.1.2)

from pdf2image import convert_from_path
import easyocr
import numpy as np
import PIL
from PIL import ImageDraw
import spacy

reader = easyocr.Reader(['en'])

CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.

poppler_path='C:\\poppler-0.68.0\\bin'

images = convert_from_path('C:\\\\Users\\\\sadok\\\\Desktop\\\\3GLID\\\\entretien\\\\cv\\\\cv2.pdf')

def draw_boxes(image, bounds, color='yellow', width=2):
    draw = ImageDraw.Draw(image)
    for bound in bounds:
        p0, p1 ,p2, p3 = bound[0]
        draw.line([*p0, *p1, *p2, *p3, *p0], fill=color, width=width)
    return image
draw_boxes(images[0], bounds)
```

Figure 62: Découpage d'un CV en sections

Le résultat de l'exécution du ce code est illustré par la figure 63.

Out[10]:

**Nawras Haloui**  
Engineering student in Software Engineering and Business Intelligence.  
I seek to increase my technical skills through practical projects in order to improve myself and be operational for the benefit of business developments.

**EDUCATION**

Tunisian National School of Engineers  
(ENSIT)  
09/2018 - Présent Tunis

PHYSICAL MATHEMATICS preparatory cycle at the Preparatory Institute for Engineering Studies in Nabeul  
(IPEIN)  
09/2016 - 06/2018 Hammamet

**SKILLS**

Programming : C, C++, Java, Python, C#.

Web and Mobile : JavaScript, Jquery, Bootstrap, PHP5, Ajax, JEE, Spring MVC/Boot, Angular8, Android.

Database : PL-SQL Oracle, MYSQL, SQLITE3, MongoDB Oracle.

Modeling and Design : UML, Merise, Agile, Design Patterns.

BI: Talend, Pentaho.

Big Data: Hadoop, Spark.

Machine Learning: Anaconda, Python.

Operating system : Windows, Linux, Ubuntu, Kali.

Framework : Hibernate, JSP.

Cloud : AWS, RDS, S3, ELB, ASC.

DevOps: Docker, Docker Jenkins, Git.

**CERTIFICATIONS**

- 98-364:MTA: Database Fundamentals
- 98-388:MTA: Introduction To Programming using Java
- 98-382:MTA: Introduction To Programming using JavaScript
- 98-381:MTA: Introduction To Programming using Python
- 98-383:MTA: Introduction to Programming using HTML and CSS

Figure 63: Résultat de l'exécution de la fonction de découpage

## Segmentation

L'étape de segmentation consiste à segmenter les sections déjà découpées et de faire l'analyse syntaxique des sections découpées.

La figure 64 montre le code de cette étape.

```
from pdf2image import convert_from_path
import easyocr
import numpy as np
import PIL
from PIL import ImageDraw
import spacy
reader = easyocr.Reader(['en'])
images = convert_from_path('C:\\\\Users\\\\sadok\\\\Desktop\\\\\\3GLID\\\\\\entretien\\\\cv\\\\cv2.pdf')
bounds = reader.readtext(np.array(images[0]), min_size=0, slope_ths=0.2, ycenter_ths=0.7, height_ths=0.6, width_ths=0.8, decoder='ctc')
bounds
x = str(b)
y=x[2:len(x)-3]
string_path = y
the_path = Path(string_path)
p = PureWindowsPath(y)
y= str(p)
y=y[1:len(y)-1]
#response = requests.get(y)
#images = pdf2image.convert_from_bytes(response.content)
#y=y.replace('/', '\\')
images = convert_from_path(y)

bounds = reader.readtext(np.array(images[0]), min_size=0, slope_ths=0.2, ycenter_ths=0.7, height_ths=0.6, width_ths=0.8, decoder='ctc')
#draw = ImageDraw.Draw(display(images[0]))
```

Figure 64: Code de segmentation d'un CV

A titre d'exemple, le nombre des segments des compétences calculés est données par la figure 65.

Le nombre des segments de la compétence « Python » est indiqué dans la liste encadrée en rouge, notamment le premier indice de la colonne représente la ligne et le deuxième indice représente la ligne de son apparition.

```
([[476, 1239], [786, 1239], [786, 1270], [476, 1270]],  
 'Framework : Hibernate,JSP',  
 0.6254210011620082),  
 ([[83, 1254], [164, 1254], [164, 1285], [83, 1285]],  
 'Python',  
 0.9999548368053072),  
 ([[474, 1291], [806, 1291], [806, 1323], [474, 1323]],  
 'Cloud :AWS,RDS,S3,ELB, ASC.',  
 0.7720314368793177),  
 ([[84, 1320], [348, 1320], [348, 1346], [84, 1346]],  
 '98-383.MTA: Introduction',  
 0.5777396573968945),  
 ([[84, 1347], [314, 1347], [314, 1378], [84, 1378]],  
 'to Programming using',  
 0.9613203699466414),  
 ([[473, 1339], [891, 1339], [891, 1377], [473, 1377]],  
 'DevOps: Docker; Docker Jenkins, Git:',  
 0.852574263466928),
```

Figure 65: Exécution de la segmentation d'un CV

### 3.4.2. Conversion et extraction

Dans cette étape, nous allons préciser les étapes de conversion et de l'extraction.

#### Conversion

Suite à la conversion du CV du format PDF en un fichier texte, l'ensemble des segments seront assembler et présenter dans ce fichier comme le montre la figure 66.

```

Entrée [51]: text='''
for i in range(len(bounds)):
    text = text +bounds[i][1]
print(text)

Naouas HalouiEngineering student in Software Engineering and Business Intelligence:Iseek to increase my technical skills through practical projects in order to improve myself and be operational forthe benefit of business developments_EDUCATIONNaouasHaloui1@gmail.comTunisian National School of Engineers55 66 4496(ENSI)09/2018 - PresentTunisavenue hbib bourgiba,PHYSICAL MATHEMATICS preparatory cycle at the Preparatory InstituteHammanet; Tunisiefor Engineering Studies in Nabeul(IEIN)linkedin.com/in/naouas-naouas-09/2016 - 06/2018Hammanet774a96170githubcom/naouas-halouiSKILLSProgramming : C, C++, Java, Python, C#:CERTIFICATIONSWeb and Mobile : JavaScript, Jquery, Bootstrap, PHP5, Ajax, JEE, Spring MVC/Boot, Angular8,98-364MTA: DatabaseAndroid,Fundamental sDatabase : PL-SQL Oracle; MySQL, SQLITE3, MongoDB Oracle:98-388.MTA: IntroductionModeling and Design : UML, Merise, Agile, Design Patterns Programming using Java@II: Talend, Pentaho,98-382MTA: IntroductionBig Data: Hadoop, Sparkle Programming usingJava ScriptMachine Learning: Anaconda, Python:98-381MTA: IntroductionOperating system : Windows, Linux, Ubuntu; Kali:to Programming usingFramework : Hibernate,JSF,PythonCloud :AWS,RDS,S3,ELB, ASC:98-383.MTA: Introductionto Programming usingDevOps: Docker; Docker Jenkins, Git:HTML and CSSCISCO Cisco CertifiedEXPERIENCESNetwork AssociateTechnician internshipBFI Groupe07/2020 - 09/2020 a TunisLANGUAGESDesign and development ofa "Credit and account management" web application:French: Read, written, spoken:Technologies and methodology: SpringBoot; Angular 8, MicroServices,MongoDB, Scrum:Worker internshipEnglish: Read, written, spoken:Kiwi Softwares06/2019 - 08/2019 a HammanetDesign and development ofa "Stock Management" desktop applicationTechnologies : Java Swing, MySQL_PERSONAL PROFILEPERSONAL PROJECTSPFAZ: Design and development of a "Premises management" websiteCreative, Responsible_Technologies and methodology: Spring boot MVC, Angular8, bootstrap, MySQL, Scrum:serious, organized,motivated.Desktop application o f "JuniorArtus Invoice Management":developed the part of invoicemanagement and printing within the Junior club For the company ArtusActive member of the ClubTechnologies: Java Swing, MySQL:Microsoft(2018/2019).Web application of "Management of universitl clubs".Technologies: PHP Bootstrap, Sqlite:Active member of the ClubJunior (2019/2020)_Web application "Platform for submitt ing student work"_Technologies: Python Bootstrap, Wamp MySQL.

```

Figure 66: Etape de conversion d'un CV

## Extraction

Afin d'identifier les compétences des candidats, nous avons fait appel à la bibliothèque Spacy du modèle NLP qui fournit une API concise pour accéder à ses méthodes et propriétés régies par des modèles d'apprentissage machine (et en profondeur) entraînés [23].

La figure 67 montre le code de cette étape.

```

from pyresparser import ResumeParser

from resume_parser import resumeparse

data = resumeparse.read_file('C:\\\\Users\\\\sadok\\\\Desktop\\\\3GLID\\\\entretien\\\\cv\\\\cv2.pdf')
data

2021-05-18 09:25:04,376 [MainThread ] [INFO ] Retrieving http://search.maven.org/remotecontent?filepath=org/apache/tika/tika-
server/1.24/tika-server-1.24.jar to C:\\Users\\sadok\\AppData\\Local\\Temp\\tika-server.jar.
2021-05-18 09:26:47,209 [MainThread ] [INFO ] Retrieving http://search.maven.org/remotecontent?filepath=org/apache/tika/tika-
server/1.24/tika-server-1.24.jar.mds to C:\\Users\\sadok\\AppData\\Local\\Temp\\tika-server.jar.mds.
2021-05-18 09:26:49,339 [MainThread ] [WARN] Failed to see startup log message; retrying...
2021-05-18 09:26:54,449 [MainThread ] [WARN] Failed to see startup log message; retrying...
2021-05-18 09:26:59,462 [MainThread ] [WARN] Failed to see startup log message; retrying...
2021-05-18 09:27:04,474 [MainThread ] [ERROR] Tika startup log message not received after 3 tries.
2021-05-18 09:27:04,513 [MainThread ] [ERROR] Failed to receive startup confirmation from startServer.
ERROR:root>Error in tika installation:: Unable to start Tika server.
ERROR:root:-----
ERROR:root>Install java for better result

```

Figure 67: Etape d'extraction d'un CV

Le résultat d'identification des compétences de ce code est illustré dans la figure 68.

```

{'email': 'nawrashaloui@gmail.com',
'phone': '',
'name': 'Nawras Haloui',
'total_exp': 0,
'university': [],
'designition': ['student',
'devops',
'network associate',
'associate',
'technician'],
'degree': [],
'skills': ['github.com/nawras-haloui',
'Programming : C',
'C++',
Java',
Python',
C#',
'Design and development of a "Credit and account management" web application.: Python',
'Bootstrap',
'Wamp MYSQL.'],
'Companies worked at': []}

```

Figure 68: Exécution d'identification des compétences

### 3.4.3. Évaluation

Après avoir extrait les compétences, nous allons travailler dans cette partie sur les calculs des scores des compétences.

Dans cet exemple, nous avons étudié le cas de la sélection de la compétence « BI » pour s'avoir son apparition dans un CV qui est égale à quatre dans ce cas (figure 69).

```

a=0
i=0
name="bi"
while i<len(text):
    if text[i]==" ":
        i=i+1

    elif text[i:i+len(name)].upper()==name.upper():
        a=a+1
        print(text[i:i+len(name)])
        i=i+len(name)

    else:
        i=i+1

print(a)

```

```

bi
bi
BI
BI
4

```

Figure 69: Calcul du score des compétences d'un CV

### 3.4.4. Affichage et tri

Suite à la sélection de la compétence du langage « JEE », la liste triée des candidatures sera affichée comme l'indique la figure 70.

The screenshot shows a user interface for managing job applications. At the top, there's a 'MENU' button and a header 'Les candidatures de l'offre'. Below this, three candidate profiles are listed:

- Yasmine ben slimen**: Profile picture, 'voir profil' link, skills: JAVA, Oracle, JEE.
- Oussema ben slimen**: Profile picture, 'voir profil' link, skills: JAVA, JEE, SQL.
- Nawras Halouii**: Profile picture, 'voir profil' link, skills: JAVA, Spring Boot, JEE, ANGULAR.

To the right, a sidebar titled 'Filtrer' contains a section for 'Competence' with checkboxes for Java (unchecked), JEE (checked), Oracle (unchecked), and reseau (unchecked).

Figure 70: Analyse des candidatures par compétences

## 4. Sprint 3.2 : Analyser les CV par niveau de la langue

Dans cette partie, nous allons étudier l'analyse des CV par niveau de langue pour déterminer les fautes orthographiques des candidats dans les CV.

### 4.1. Backlog du sprint 3.2

Le tableau 25 représente le backlog de notre sprint " Analyser des CV par niveau de langue" où nous listons les différentes exigences à mettre en place dans cette section.

ID Module	Module	Id user Story	User story
1	<b>Analyser des CV par niveau de la langue</b>	1.1	En tant qu'un responsable des ressources humaines, sélectionner une offre.
		1.2	En tant qu'un responsable des ressources humaines, je peux afficher les candidatures triées par compétence.
		1.3	Le modèle va calculer score pour chaque candidature.
		1.4	Le modèle va extraire les données d'un CV.

Tableau 25: Backlog du sprint 3.2

## 4.2. Analyse des besoins du sprint 3.2

Dans cette partie nous allons passer au diagramme de cas d'utilisation pour entrer plus en détails tout en présentant un diagramme de séquence qui contient les fonctions essentielles.

### 4.2.1. Diagramme de cas d'utilisation

La figure 71 illustre le diagramme du cas d'utilisation du sprint " Analyser les CV par niveau de la langue" qui contient toutes les fonctionnalités à réaliser.

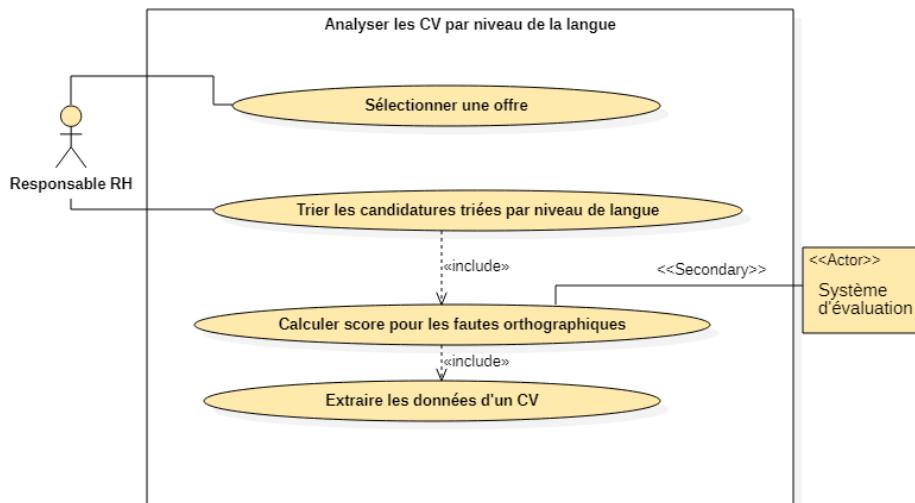


Figure 71: Diagramme de cas d'utilisation du sprint 3.1

### 4.2.2. Raffinements des cas d'utilisation

#### Description textuelle du cas d'utilisation « Analyser les CV par niveau de la langue »

Le tableau 26 illustre un tableau récapitulatif du cas d'utilisation « Analyser des CV par niveau de langue ». Nous décrivons dans ce tableau l'objectif, l'acteur principal de la fonctionnalité, le scénario nominal et les exceptions qui peuvent se produire.

Raffinement de cas d'utilisation	
<b>Titre</b>	Analyse des CV par niveau de la langue
<b>Acteur</b>	Responsable des ressources humaines
<b>Objectif</b>	Trier les candidatures les plus performants à travers la sélection de la compétence par le responsable des ressources humaines.
<b>Résumé</b>	Le responsable des ressources humaines doit trier les candidatures de chaque offre selon le niveau de la langue.
Conditions	
<b>Préconditions</b>	<b>Postconditions</b>

<ul style="list-style-type: none"> <li>- Le responsable des ressources humaines doit être connecté à l'application.</li> <li>- Présence des candidatures relatives à l'offre sélectionnée</li> </ul>	<p>Le responsable des ressources humaines choisit la compétence et les candidatures des candidats vont être triées.</p>
<b>Scenario</b>	
<p>1- Le responsable des ressources humaines sera redirigé vers l'interface gestion des offres après avoir connecté à l'application.</p> <p>2- Le système affiche la liste des offres.</p> <p>3- Le responsable des ressources humaines visualise les candidatures de chaque offre.</p> <p>4- Le responsable des ressources humaines sélectionne une compétence pour trier les candidatures.</p> <p>5- Le système affiche la liste des candidatures triées selon les fautes orthographiques dans la langue des candidats.</p>	
<b>Exception</b>	
<p>Dans l'étape 3 : - Si une offre ne contient des candidatures, le processus d'analyse des compétences à travers les CV ne se produit pas.</p> <p>Dans l'étape 4 : - Si le responsable des ressources humaines ne sélectionne pas une compétence, le processus d'analyse des compétences ne se produit pas.</p>	

Tableau 26: Description cas d'utilisation : Analyser les CV par compétence

### 4.3. Conception du sprint 3.2

Dans cette étape, nous présentons le diagramme des classes de ce sprint et en second lieu nous montrons le diagramme de séquence « Analyser les CV par niveau de la langue ».

#### 4.3.1. Diagramme des classes de conception du sprint 3.2

Le diagramme 72 représente le système des classes du premier sprint « Analyser les CV par niveau de la langue ». Nous présentons dans cette partie les classes et les relations entre elles.

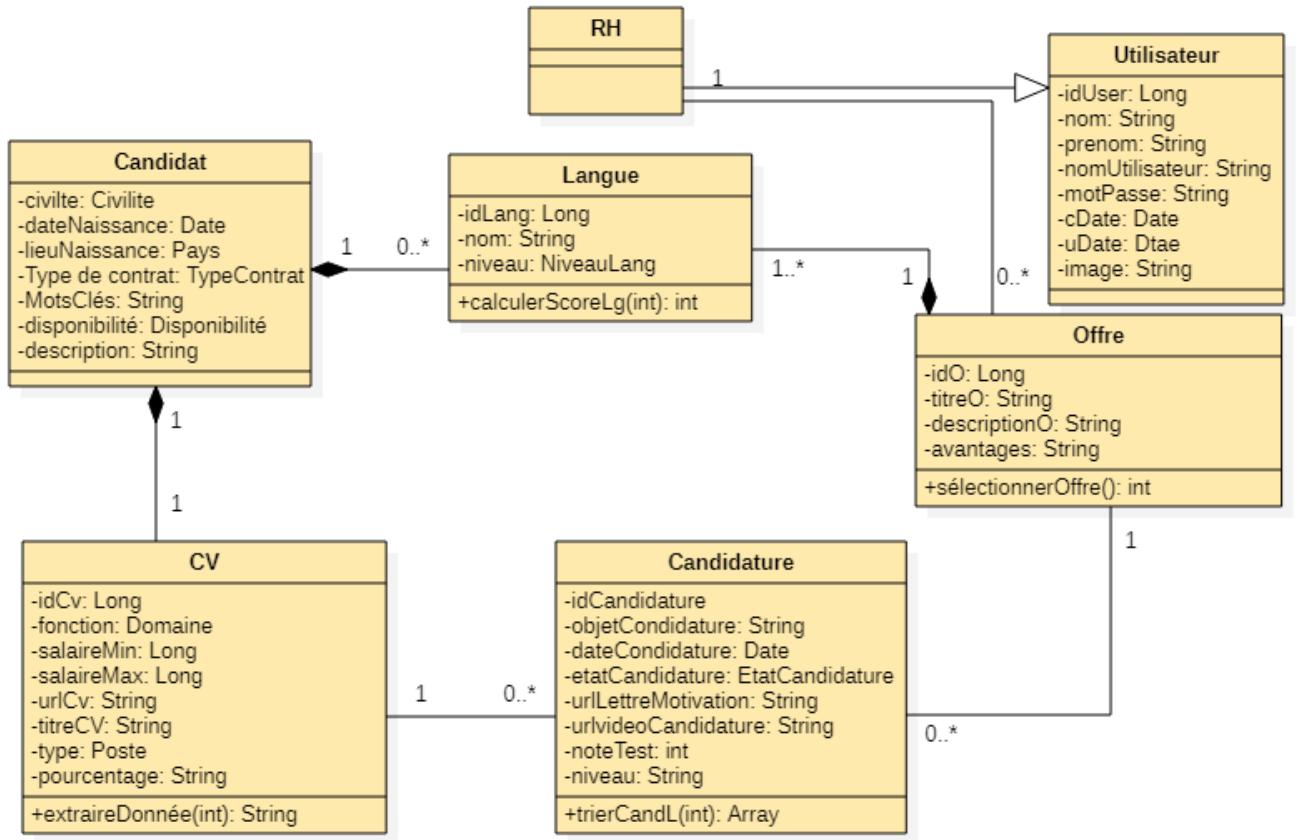


Figure 72: Diagramme du sprint 3.1

Dans ce sprint nous avons défini les classes suivantes :

- Classe « *CV* » : Elle est caractérisée par les attributs *idCv*, *fonction*, *salaireMax*, *urlCV*, *titreCv*, *type* et *score*. Elle admet la méthode *extraireDonnée(int) :String*.
- Classe « *Langue* » : Elle est définie par les attributs *idLang*, *nom* et *niveau*. Cette classe admet la méthode *calculerScoreLg(int) :int*.
- Classe « *Candidature* » : Elle contient les attributs *idCandidature*, *dateCandidature*, *objetCandidature*, *noteTest* et *niveau*, et elle admet la méthode *trierCandL(int) :Array*.
- Classe « *Offre* » : elle est définie par les attributs *idO*, *titreO*, *descriptionO*, *avantages* et par la méthode *sélectionnerOffre() :int*.

#### 4.3.2. Diagramme de séquence du sprint 3.2

Le diagramme 73 montre le diagramme de séquence de sprint « « Analyser les CV par niveau de la langue ».

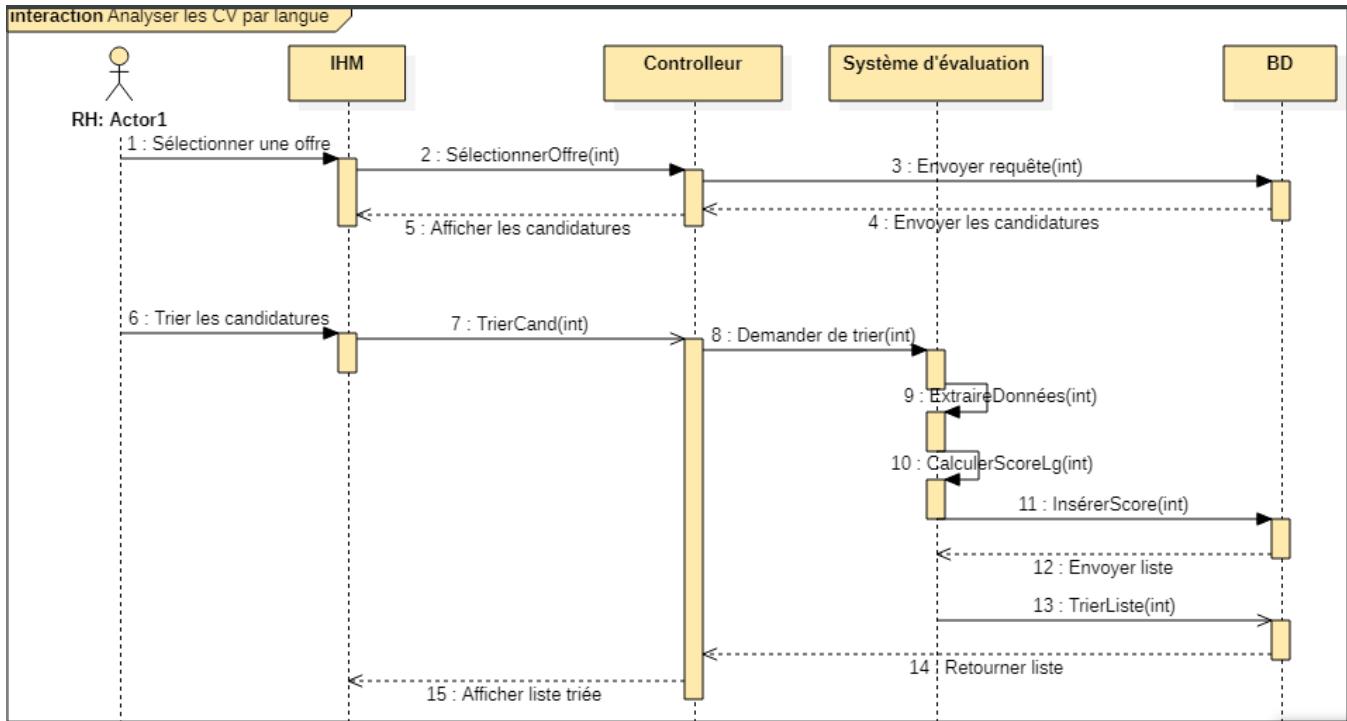


Figure 73: Diagramme du sprint 3.2

#### 4.4.Réalisation du sprint 3.2

Suite aux mêmes étapes précédentes de « Découpage et segmentation », « Conversion et extraction », nous allons étudier la partie d'évaluation d'analyse des CV par niveau de la langue et pour cette raison nous allons chercher, déterminer et compter le nombre des erreurs orthographiques commise par le candidat.

Les figures 74 et 75 présentent le code développé pour ce sprint.

```

1. from flask import Flask, jsonify,
2. from pdf2image import convert_from
3. import easyocr
4. import numpy as np
5. import PIL
6. from PIL import ImageDraw
7. import spacy
8. import cx_Oracle
9. from flask_cors import CORS
10. import requests
11. from textblob import TextBlob
12. from spellchecker import SpellChecker
13. from pathlib import Path, PureWindowsPath
14. import pdf2image
15. import re
16. app = Flask(__name__)
17. cors = CORS(app, resources={r"/api/*": {"origins": "*"}})
18. reader = easyocr.Reader(['en'])
19. poppler_path='C:\poppler-0.68.0\bin'
20. from IPython.display import display, Image
21. conn = cx_Oracle.connect("bficv def", "sa"

```

```

10. cd where ap.candidata_id=cd.id and cd.id=
cvv.candidat_id and ap.id_app= :
11. idA", idA=idapp).fetchall()
12. x = str(b)
13. y=x[2:len(x)-3]
14. string_path =y
15. the_path = Path(string_path)
16. p = PureWindowsPath(y)
17. y=str(p)
18. y=y[1:len(y)-1]
19. #response = requests.get(y)
20. #images = pdf2image.convert_from_bytes(response.content)
21. # y=y.replace('/', '//')
22. file1=y
#Conversion
23. images = convert_from_path(y)
24. bounds = reader.readtext(np.array(images[0]), min_size=0, slope_ths=0.2, 25.ycenter_
ths=0.7, height_ths=0.6, width_ths=0.8, deco_
der='beamsearch', beam26.Width=10 )
27. text=''
28. for i in range(len(bounds)):
29.   text = text +bounds[i][1]
30. file1=text

```

## #Importation des bibliothéques et connexion à la base de données

## #Fonction d'analyse des CV par niveau de la langue

```

1. @app.route('/langue/<idapp>')
2. def langue(idapp):
#Lister des mots clés pour ne pas les considerer des fautes
3. comp=['java','jee','springboot','bi','spark','javascr
ipt']
4. nb=0
5. curr= conn.cursor()
#Sélection
6. images = curr.execute("select cvv.cv from t_cv cvv,t_a
pplication ap,t_7.candidat cd where ap.candidata_id=cd.i
d and cd.id=cvv.candidat_id and ap.id8._app= : idA", idA
=idapp)
9. b = curr.execute("select cvv.cv from t_cv cvv,t_appli
cation ap,t_7.candidat cd where ap.candidata_id=cd.i
d and cd.id=cvv.candidat_id and ap.id8._app= : idA", idA
=idapp)
31. #with open(file1,"r+") as filehandle:
32.   #filecontent=filehandle.read()
33.   print("Original Text:\n", str(file1))
34.   b = TextBlob(file1)
35.   print("Corrected text:\n", str(b.correct()))
36.#remove all punctuations before finding possible mis
spelled words
37. s = re.sub(r'[^\w\s]','',file1)
38. print("Text without punctuations:\n",s)
39.wordlist=s.split()
40. spell = SpellChecker()
41.# find those words that may be misspelled
42.misspelled = list(spell.unknown(wordlist))
43. print("Possible list of misspelled words in the ori
ginal text:\n",misspel44.led)
45. print("la liste des fautes orthographique est : \n"
, len(misspelled))
46.#textblob cannot correct all misspellings , some cor
rections might be meaningless, so its important to f
ind all candidate words
48. for word in misspelled:
49.   # Get the one `most likely` answer
50.   for wordd in comp:
51.     if (word==wordd):
52.       nb+=1

```

Figure 74: Code d'analyser les CV par niveau de la langue (1)

## #Fonction de tri des candidatures (**Evaluation**)

```
74.@app.route('/api/getallAppL/<idOffer>', methods=['GET'])
75.def api_allLang(idOffer):
76. c=currr.execute("select id_app from t_application where offer_id_offer=: 77.idA", idA=idOffer)
    .fetchall()
78. x = str(c)
79.y=x[2:len(x)-2]
80.b=y.replace("(, ")
81.a=b.replace(')', "")
82.list=a.replace(',', "")
83.list=list.replace(" ", "")
84.l=str(list)
85. l=y[1:len(l)-1]
86.print("pour cette offre on a "+str(len(list)) +" candidatures" )
87.for j in range(len(list)) :
88.     id= list[j]
89.     print("l'id de la candidature est : "+id)
90     langue(int(list[j]))
91.b=cur.execute("select app.* from t_application app, t_candidat cd ,t_l
92.ngue s  where cd.id=s.candidat_id and  cd.id=app.candidata_id and app.offer_id_offer= : id
order by s.level_lang desc", id=idOffer).fetchall()
94.()
95.print("tri de la langue est effectué avec succès!!")
96.return jsonify(b)
```

Figure 75: Code d'analyser les CV par niveau de la langue (2)

Tout d'abord, la première tâche dans l'évaluation, c'était de corriger les fautes dans le texte original et de le comparer avec l'originale.

Ce traitement est illustré dans la figure 76.

```
pip install pyspellchecker

Collecting pyspellchecker
  Downloading pyspellchecker-0.6.2-py3-none-any.whl (2.7 MB)
Installing collected packages: pyspellchecker
Successfully installed pyspellchecker-0.6.2
Note: you may need to restart the kernel to use updated packages.

from textblob import TextBlob
from spellchecker import SpellChecker
import re
file1="original.txt"
with open(file1,"r+") as filehandle:
    filecontent=filehandle.read()
    print("Original Text:\n", str(filecontent))
    b = TextBlob(filecontent)
    print("Corrected text:\n", str(b.correct()))
```

Figure 76: Code de la comparaison des fautes

La figure 77 illustre la sortie du code de comparaison.

Original Text:  


Nawras HalouiEngineering student in Software Engineering and Business Intelligence seek to increase my technical skills through practical projects in order to improve myself and be operational for the benefit of business developments\_EDUCATIONnawrashalo ui@gmailcomTunisian National School of Engineers55 66 4496(ENSI)09/2018 - PresentTunisavenue hbib bourgiba,PHYSICAL MATHEMATIC S preparatory cycle at the Preparatory InstituteHammamet; Tunisiefor Engineering Studies in Nabeulin(IEIN)linkedin.com/in/halo ui-nawras-09/2016 - 06/2018Hammamet774a96170githubcom/nawras-halouiSKILLSProgramming : C, C++, Java, Python, C#:CERTIFICATIONS Web and Mobile : JavaScript, JQuery, Bootstrap, PHP, Ajax, JEE, Spring MVC/Foot, Angular8,98-364MTA: DatabaseAndroid.Fundamentals alDatabase : PL-SQL Oracle; MySQL, SQLITE3, MongoDB Oracle:98-388.MTA: IntroductionModeling and Design : UML, Merise, Agile, D esign Patternssto Programming using JavaBI: Talend, Pentaho.98-382MTA: IntroductionBig Data: Hadoop, Sparkto Programming usingJa vaScriptMachine Learning: Anaconda, Python:98-381MTA: IntroductionOperating system : Windows, Linux, Ubuntu; Kallisto: Programmimg usingFramework : Hibernate,JSPPythonCloud :AWS,RDS,S3,ELB, ASC.98-383.MTA: Introductionto Programming usingDevOps: Docker; Do cker Jenkins, Git:HTML and CSSCISCO Cisco CertifiedEXPERIENCESNetwork AssociateTechnician InternshipBFI Groupe07/2020 - 09/2020 a TunisLANGUAGESDesign and development of a "Credit and account management" web application:French: Read, written, spoken:DTechn ologies and methodology: SpringBoot; Angular 8, MicroServices,MongoDB, Scrum:Worker internshipEnglish: Read, written, spoken:Ki wiSoftwares06/2019 - 08/2019 a HammametDesign and development of a "Stock Management" desktop applicationTechnologies : JavaSwin g, MYSQL\_PERSONALPROFILEPERSONAL PROJECTSPFAZ: Design and development of a "Premises management" websiteCreative, Responsible\_T echnologies and methodology: Spring boot MVC, Angular8, bootstrap, MySQL, Scrum:serious, organized,motivated.Desktop applicatio n of "JuniorArtus Invoice Management":developed the part of invoicemanagement and printing within the Junior club For the company ArtusActive member of the ClubTechnologies: Java Swing, MySQL:Microsoft(2018/2019).Web application of "Management of university clubs".Technologies: PHP Bootstrap, Sqlite:Active member of the ClubJunior (2019/2020)\_Web application "Platform for submitting student work"\_Technologies: Python Bootstrap, Wamp MYSQL.  
  
Corretted text:  


Taras HalouiEngineering student in Software Engineering and Business Intelligence Seek to increase my technical skill through practical projects in order to improve myself and be operational forth benefit of business developments\_EDUCATIONnawrashaloui@g mailcomTunisian National School of Engineers55 66 4496(ENSI)09/2018 - PresentTunisavenue his bourgiba,PHYSICAL MATHEMATICS pre paratory cycle at the Preparatory InstituteHammamet; Tunisiefor Engineering Studies in Nabeulin(IEIN)linkedin.com/in/aloud-taras -09/2016 - 06/2018Hammamet774a96170githubcom/taras-halouiSKILLSProgramming : C, C++, Java, Python, C#:CERTIFICATIONS Web and Mo bile : JavaScript, Query, Bootstrap, PHP, Tax, JEE, Spring MVC/Foot, Angular,98-364MTA: DatabaseAndroid.FundamentalsDatabase : of-SQL Oracle; MySQL, SQLITE3, MongoDB Oracle:98-388.MTA: IntroductionModeling and Design : UML, RISE, Agile, Design Patternssto Programming using JavaBI: Talent, Pentaho.98-382MTA: IntroductionBig Data: Hadoop, Sparkto Programming usingJavaScriptMachine L earning: Anaconda, Python:98-381MTA: IntroductionOperating system : Windows, Sinus, Bunt; Ali:to Programming usingFramework : L iberate,JSPPythonCloud :AWS, RDS,Of,ELB, ASC.98-383.MTA: Introductionto Programming usingDevOps: Pocket; Pocket Elkins, It:HTML and CSSCISCO Disco CertifiedEXPERIENCESNetwork AssociateTechnician InternshipBFI Groupe07/2020 - 09/2020 a TunisLANGUAGESDesign and development of "Credit and account management" web application:French: Head, written, spoken:DTechnologies and methodology: SpringBoot; Angular 8, MicroServices,MongoDB, Scrum:Worker internshipEnglish: Head, written, spoken:KiwiSoftwares06/2019 - 08/2 019 a HammametDesign and development of "Stock Management" desktop applicationTechnologies : JavaSwing, MYSQL\_PERSONALPROFILEPE RSONAL PROJECTSPFAZ: Design and development of a "Premises management" websiteCreative, Responsible\_Technologies and methodology: Spring boot MVC, Angular, bootstrap, MySQL, Scrum:serious, organized,motivate.Desktop application of "JuniorArtus Voice Man agement":developed the part of invoicemanagement and printing within the Junior club For the company ArtusActive member of the ClubTechnologies: Cava Owing, MySQL:Microsoft(2018/2019).Web application of "Management of university clubs".Technologies: PHP Bootstrap, Quite:Active member of the ClubJunior (2019/2020)Web application "Platform for submitting student work"\_Technologie s: Python Bootstrap, Camp MYSQL.

Figure 77: Exécution du code de comparaison des fautes

La deuxième étape s'agit d'éliminer les ponctuations pour ne les considèrent pas comme des fautes orthographiques et donc de calculer le nombre des fautes orthographiques et de lister les différentes fautes du texte saisies.

A figure 78 présente le code développé de cette étape.

```
Entrée [16]: #remove all punctuations before finding possible misspelled words
s = re.sub(r'[^\\w\\s]', '', filecontent)
print("Text without punctuations:\n",s)
wordlist=s.split()
spell = SpellChecker()
# find those words that may be misspelled
misspelled = list(spell.unknown(wordlist))
print("Possible list of misspelled words in the original text:\n",misspelled)
print("la liste des fautes orthographique est : \n",len(misspelled))
#textblob cannot correct all misspellings , some corrections might be meaningless, so its important to find all candidate words
for word in misspelled:
    # Get the one 'most likely' answer
    print(" word:",spell.correction(word))
    # Get a list of 'likely' options
    # print("Candidate words:",spell.candidates(word))

#finally overwrite the text file with the corrected text

#with open(file1,"w") as filehandle:
#    filehandle.write(str(b.correct()))
```

Figure 78: Code de déterminaison des fautes orthographiques

La figure 79 montre donc l'exécution de ce code, en effet, dans cet exemple nous avons le nombre « 83 » qui présente le nombre des fautes orthographiques dans ce CV et les mots « word » caractérise les mots fausses dans le CV.

```
la liste des fautes orthographique est :
83
word: applicationfrench
word: javascript
word: usingjavascriptmachine
word: usingframework
word: clubjunior
word: cscisco
word: presenttunisavenue
word: pops
word: mysel
word: pop
word: angular898364mta
word: 20192020_web
word: oracle98388mta
word: nara
word: work_technologies
word: artusactive
word: internshipenglish
```

Figure 79: Liste des fautes orthographiques

Finalement, le responsable des ressources humaines filtre les candidatures par niveau de la langue, et donc les candidatures qui ont des scores des fautes orthographiques les plus faibles seront être affichées comme la montre la figure 80.

## Les candidatures de l'offre

The screenshot displays a user interface for managing job applications. On the left, a list of candidates is shown, each with a profile picture, name, a 'voir profil' link, and a 'JAVA' skill badge. The candidates listed are Nawras Halouii and saknadar. On the right, there are two filtering sections: 'Analyser les CV' (Analyze CVs) which includes a 'Compétence' section with a search bar and a 'Filtrer' button, and a 'Filtrer par langue' (Filter by language) section with its own 'Filtrer' button.

Figure 80: Liste des candidatures analysée par niveau de la langue

## Conclusion

Ce chapitre élabore un résumé du dernier release de notre application qui met en œuvre la conception et le développement des deux sprints : Analyse les CV par compétence et analyse des CV par niveau de la langue.

## Conclusion générale

Le but de ce projet est de mettre en place une application web de recrutement et d'analyse des CV permettant aux candidats chez BFI de faciliter le processus de recrutement, l'analyse des candidatures de l'application. Cette application permet à la société de gagner en termes de temps, coût et ressources.

Nous avons commencé dans un premier lieu à situer notre projet dans son cadre en touchant la problématique qui a déclenché notre travail via une solution à suivre. Dans le second chapitre, nous avons décomposé le travail demandé en des releases et des sprints. Nous avons fait une étude comparative entre les méthodes agiles (XP, scrum) afin de situer le choix de l'entreprise pour le suivi et la planification du projet. Dans le troisième chapitre, nous avons exposé notre première release consacrée à la gestion des utilisateurs et candidatures. Dans le quatrième chapitre, nous avons détaillé la deuxième release consacrée à la gestion des responsables, des tests et des entretiens. Dans le dernier chapitre, nous avons présenté la dernière release dédiée au développement d'un modèle apprentissage automatique qui permet d'analyser les CV.

La réalisation de cette application web nous a été bénéfique sur le plan technique. En effet, nous avons eu une opportunité pour maîtriser les technologies : Java, Angular, Spring boot et apprentissage automatique avec l'utilisation du modèle NLP et les bibliothèques Spacy et du PDF. Ce projet nous a donné l'occasion de mettre à l'épreuve nos connaissances pratiques en matière de conception et réalisation et surtout de nous intégrer dans la vie professionnelle et d'enrichir nos connaissances techniques sur les plans théorique et pratique. Nous estimons que l'application peut être considérablement étendue par l'ajout de nouvelles fonctionnalités telles que les perspectives suivantes :

- Analyser d'autres formats de CV tels que : HTML, CSV, etc.
- Développer une plateforme de test pour les responsables techniques et de test.
- Déployer l'application sur le cloud.
- Ajouter un module de gestion des employés et suivre leurs fonctionnalités

# **Netographie**

## **[1].BFI:**

disponible sur: <http://www.bfigroupe.com/>.(**visité le 01/03/2021**)

## **[2].Scrum**

disponible sur: <https://www.leslivresblancs.fr/dossier/scrum-la-methode-agile-de-la-theorie-la-pratique>.(**visité le 15/03/2021**)

## **[3].Agile**

disponible sur: <https://www.leslivresblancs.fr/dossier/scrum-la-methode-agile-de-la-theorie-la-pratique>.(**visité le 15/03/2021**)

## **[4].Taiga**

disponible sur: <https://www.taiga.io/>.(**visité le 15/03/2021**)

## **[5].Design Pattern**

disponible:<https://www.developpez.net/forums/d1207967/general-developpement/alm/design-patterns/mvc-modification-directe-vue-controleur/>.(**visité le 15/03/2021**)

## **[6].MVC**

disponible:<https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur>.(**visité le 15/03/2021**)

## **[7].Midilleware**

<https://www.redhat.com/fr/topics/middleware/what-is-ide> (**visité le 15/03/2021**)

## **[8].Postman**

<https://riptutorial.com/Download/postman-fr.pdf> (**visité le 15/03/2021**)

## **[9].SQL**

<https://www.logiciels.pro/logiciel-saas/oracle-sql-developer/>.(**visité le 15/03/2021**)

## **[10].StarUML**

<https://inf1410.teluq.ca/teluqDownload.php?file=2014/01/INF1410-PresentationStarUML> (**visité le 15/04/2021**)

## **[11].GITLAB**

<https://junto.fr/blog/gitlab/> (**visité le 15/03/2021**)

## [12].Angular

[https://easypartner.fr/blog/que-choisir-entre-react-ou-angular/#:~:text=Pour%20aider%20%C3%A0%20d%C3%A9velopper%20un,composants%20depuis%20son%20propre%20terminal.\(visité le 15/04/2021\)](https://easypartner.fr/blog/que-choisir-entre-react-ou-angular/#:~:text=Pour%20aider%20%C3%A0%20d%C3%A9velopper%20un,composants%20depuis%20son%20propre%20terminal.(visité le 15/04/2021))

## [13].SpringBoot

[https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5122425-decouvrez-le-framework-spring-boot \(visité le 15/04/2021\)](https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5122425-decouvrez-le-framework-spring-boot (visité le 15/04/2021))

## [14].Hibernate

[https://www.jmdoudoux.fr/java/dej/chap-hibernate.htm#:~:text=Hibernate%20est%20une%20solution%20open,objets%20Java%20et%20vice%20versa. \(visité le 15/04/2021\)](https://www.jmdoudoux.fr/java/dej/chap-hibernate.htm#:~:text=Hibernate%20est%20une%20solution%20open,objets%20Java%20et%20vice%20versa. (visité le 15/04/2021))

## [15].JWT

[https://oa.dnc.global/web/-JSON-Web-Token-JWT-JWS-.html \(visité le 15/04/2021\)](https://oa.dnc.global/web/-JSON-Web-Token-JWT-JWS-.html (visité le 15/04/2021))

## [16].SpringSecurity

[https://www.invivoo.com/securiser-application-spring-boot-spring-security/ \(visité le 10/05/2021\)](https://www.invivoo.com/securiser-application-spring-boot-spring-security/ (visité le 10/05/2021))

## [17].Python

[https://www.lebigdata.fr/python-langage-definition#:~:text=Python%20est%20un%20langage%20de,d%C3%AAtre%20compl%C3%A9mentaire%20pour%20fonctionner.&text=Il%20s'agit%20d'un%20langage%20id%C3%A9al%20pour%20les%20d%C3%A9butants. \(visité le 10/05/2021\)](https://www.lebigdata.fr/python-langage-definition#:~:text=Python%20est%20un%20langage%20de,d%C3%AAtre%20compl%C3%A9mentaire%20pour%20fonctionner.&text=Il%20s'agit%20d'un%20langage%20id%C3%A9al%20pour%20les%20d%C3%A9butants. (visité le 10/05/2021))

## [18].JWT SECURITY

[https://www.vaadata.com/blog/fr/jetons-jwt-et-securite-principes-et-cas-utilisation/#:~:text=Les%20%C2%AB%20JSON%20Web%20Token%20%C2%BB%20ou,d'identifier%20l'utilisateur. \(visité le 10/05/2021\)](https://www.vaadata.com/blog/fr/jetons-jwt-et-securite-principes-et-cas-utilisation/#:~:text=Les%20%C2%AB%20JSON%20Web%20Token%20%C2%BB%20ou,d'identifier%20l'utilisateur. (visité le 10/05/2021))

## [19].machine learning

[https://oa.dnc.global/web/-Gestion-du-jeton-d-acces-.html \(visité le 25/05/2021\)](https://oa.dnc.global/web/-Gestion-du-jeton-d-acces-.html (visité le 25/05/2021))

[https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing#:~:text=Machine%20learning%20for%20NLP%20and,and%20other%20aspects%20of%20text.&text=It%20also%20could%20be%20a,known%20as%20unsupervised%20machine%20learning \(visité le 25/05/2021\)](https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing#:~:text=Machine%20learning%20for%20NLP%20and,and%20other%20aspects%20of%20text.&text=It%20also%20could%20be%20a,known%20as%20unsupervised%20machine%20learning (visité le 25/05/2021))

**[20].NLP**

[https://www.google.com/search?q=etape+de+traitement+nlp+cv&sxsrf=ALeKk00KCYUwAkepNXonnypqdOWSdG\\_NHA:1624258081425&source=lnms&tbs=isch&sa=X&ved=2ahUKEwi8xt3vkKjxAhVSOhoKHWIwDFIQ\\_AUoAXoECAEQAw&biw=1366&bih=657#imgrc=iIRPqJJMSdAdsM](https://www.google.com/search?q=etape+de+traitement+nlp+cv&sxsrf=ALeKk00KCYUwAkepNXonnypqdOWSdG_NHA:1624258081425&source=lnms&tbs=isch&sa=X&ved=2ahUKEwi8xt3vkKjxAhVSOhoKHWIwDFIQ_AUoAXoECAEQAw&biw=1366&bih=657#imgrc=iIRPqJJMSdAdsM) (**visité le 25/05/2021**)

**[21].SPACY**

<https://www.moodle.lesiteduprof.fr/Data/Martelet/NSI/opale/langages/co/00-InstallerPython/co/01-PresentationPython.html> (**visité le 15/06/2021**)

## Résumé

Ce projet réalisé au sein de la BFI, s'inscrit dans le cadre de notre projet de fin d'étude au sein de l'École Nationale Supérieure d'Ingénieurs de Tunis. Il consiste à concevoir et développer une application web permettant aux candidats de postuler pour une offre de recrutement et aux responsables des ressources humaines d'analyser et d'évaluer automatiquement les CV. Pour le conduire, nous avons utilisé la méthodologie Agile Scrum et pour le réaliser nous avons utilisé le modèle NLP pour l'extraction de l'information avec sa bibliothèque Spacy, et Angular, SpringBoot et Flask pour le développement de l'application web.

**Mots clés :** Analyse des CV, offre de recrutement, apprentissage automatique, modèle NLP, SpringBoot, Angular, Flask , Python.

## Abstract

This project, carried out within the BFI, is part of our end-of-study project at the National School of Engineers of Tunis. It consists in designing and developing a web application allowing candidates to apply for a recruitment offer and human resources managers to automatically analyze and evaluate CVs. To conduct it, we used the Agile Scrum methodology and to achieve it, we used the NLP model for information extraction with its Spacy library, and Angular, SpringBoot and Flask for the development of the web application.

**Keywords :** CV analysis, recruitment offer, machine learning, NLP model, SpringBoot, Angular, Flask, Python.

## الملخص

هذا المشروع، الذي تم تنفيذه في BFI، هو جزء من مشروع نهاية الدراسة لدينا في المدرسة الوطنية للمهندسين بتونس. وهو يتالف من تصميم وتطوير تطبيق ويب يسمح للمرشحين بالتقدم بطلب للحصول على عرض توظيف ويسمح أيضاً لمديري الموارد البشرية لتحليل وتقييم السير الذاتية تلقائياً. لقيادتها، استخدمنا منهجية Agile Scrum ولتحقيقها استخدمنا نموذج البرمجة اللغوية العصبية لاستخراج المعلومات مع مكتبة Spacy الخاصة بها ، و Angular و SpringBoot و Flask لتطوير تطبيق الويب.

**الكلمات المفتاحية:** تحليل السيرة الذاتية، عرض التوظيف، التعلم الآلي، نموذج البرمجة اللغوية العصبية، Angular، SpringBoot ،Python، Flask