

# *Dédicaces*

## **À mes parents :**

Je vous remercie profondément pour votre amour, vos sacrifices et votre soutien constant. Vous avez toujours cru en moi et m'avez donné la force d'avancer malgré les difficultés. Votre présence a été essentielle tout au long de mon parcours. Vous êtes les véritables piliers de ma réussite.

## À mes sœurs :

Merci pour votre affection, votre patience et votre écoute dans les moments difficiles.

Votre soutien moral m'a beaucoup aidé durant ces années d'études. Vous avez su m'apporter du réconfort et de la sérénité. Je vous suis sincèrement reconnaissant pour cela.

## À chers amis :

Votre présence a été précieuse tout au long de cette aventure. Merci pour les encouragements, les échanges et les souvenirs partagés. Vous avez su rendre cette période plus agréable et motivante. Votre amitié a été un vrai moteur dans mon quotidien.

## À tous mes professeurs et encadrants :

Je vous remercie pour votre disponibilité, vos conseils et votre patience. Votre accompagnement a grandement contribué à ma formation. Chaque enseignement reçu a enrichi mon parcours personnel et académique. Votre implication a été pour moi une réelle source d'inspiration.

---

# Remerciement

Afin de m'assurer du succès de mon stage et de mon projet, je suis redevable à de nombreuses personnes.

Tout d'abord, je tiens à remercier, en premier lieu, l'ensemble de **l'équipe pédagogique de l'École Polytechnique de Sousse**, ainsi que les intervenants professionnels responsables de la formation du Cycle d'Ingénieur en Génie Informatique, pour leur engagement dans la transmission des connaissances théoriques.

Je souhaite également exprimer ma profonde gratitude à mon encadrant académique, **Monsieur Haythem saoudi**, pour son accompagnement et ses conseils avisés durant ce travail. Leur expertise et leur bienveillance ont été indispensables à l'aboutissement de ce projet.

Je voudrais aussi exprimer mes profonds remerciements à mon encadrante professionnelle, **Madame Sondos Aouachri**, pour son aide précieuse et son soutien tout au long de la réalisation de ce projet.

Enfin, je tiens à remercier chaleureusement les membres du jury pour leur temps et leur expertise dans l'évaluation de mon travail. Je suis honoré qu'ils aient accepté de juger ce projet et j'espère qu'il a été à la hauteur de leurs attentes.

---

# Liste des acronymes

**API** : Application Programming Interface

**CSS** : Cascading Style Sheets

**HTML** : HyperText Markup Language

**HTTP** : HyperText Transfert Protocol

**IT** : Informatique Technologie

**JSON** : JavaScript Object Notation

**JWT** : JSON Web Token

**SQL** : Structured Query Language

**UI** : User Interface

**UML** : Unified Modeling Language

---

# Table des matières

Introduction Générale . . . . .	1
<b>1 Cadre général du projet</b>	<b>2</b>
1 Introduction . . . . .	3
2 Entreprise d'accueil . . . . .	3
2.1 Présentation de Tekboot Solutions . . . . .	3
2.2 Secteurs d'activités . . . . .	4
2.3 Spécialisations . . . . .	4
2.4 Les services de l'entreprise . . . . .	4
2.5 Fiche signalétique . . . . .	4
3 Cadre du projet . . . . .	5
3.1 Contexte Général et problématique . . . . .	5
3.1.1 Présentation du projet . . . . .	5
3.1.2 Problématique . . . . .	5
3.2 Étude de l'existant . . . . .	6
3.2.1 EmploiTunisie.com . . . . .	6
3.2.2 TanitJobs.com . . . . .	8
3.2.3 KeeJob.com . . . . .	9
3.3 Critique des solutions existantes . . . . .	10
4 Solution proposée . . . . .	12
5 Etude de la méthodologie de développement . . . . .	12
5.1 Choix de la méthodologie de développement . . . . .	13
5.2 Avantages de Scrum . . . . .	13
5.3 Mise en pratique de la méthodologie Scrum . . . . .	13
6 Conclusion . . . . .	15

<b>2</b>	<b>Analyse et spécification des besoins</b>	<b>16</b>
1	Introduction . . . . .	17
2	Description fonctionnelle des besoins . . . . .	17
2.1	Identification des acteurs . . . . .	17
2.2	Capture des besoins fonctionnels . . . . .	18
2.3	Capture des besoins non fonctionnels . . . . .	19
3	Pilotage du projet avec SCRUM . . . . .	19
3.1	Équipe et rôles . . . . .	19
3.2	Backlog du produit . . . . .	21
3.3	Diagramme de cas d'utilisation général . . . . .	23
3.4	Diagramme de classes général . . . . .	25
3.5	Planification des sprints . . . . .	26
4	Environnement de travail . . . . .	27
4.1	4.1 Environnement logiciel . . . . .	27
4.2	4.2 Environnement matériel . . . . .	28
4.3	4.3 Technologies utilisées . . . . .	29
5	Spécification architecturale . . . . .	33
5.1	Architecture physique . . . . .	33
5.2	Architecture logique . . . . .	34
5.3	Interaction entre les composants . . . . .	35
6	Conclusion . . . . .	36
<b>3</b>	<b>Etude et réalisation du release 1</b>	<b>37</b>
1	Introduction . . . . .	38
2	Développement du sprint 1 « Sécurité et gestion d'utilisateurs et leurs rôles » . . . . .	38
2.1	Backlog du sprint 1 . . . . .	38
2.2	Spécification fonctionnelle . . . . .	39
2.2.1	Diagramme de cas d'utilisation sprint 1 . . . . .	39
2.2.2	Description textuelle des cas d'utilisation du sprint 1 . . . . .	40
2.2.3	Diagramme de séquence . . . . .	40
2.3	Réalisation . . . . .	41
2.3.1	Interface d'accueil . . . . .	41

2.3.2	Interface de Connexion . . . . .	42
2.3.3	Interface de création d'un compte et interface de gestion utilisateurs . . . . .	42
2.3.4	Interface de création d'un compte et interface de gestion utilisateurs . . . . .	43
2.3.5	Interface de gestion d'utilisateurs . . . . .	43
2.4	Rétrospective . . . . .	44
3	Développement du sprint 2 « Gestion des départements et des projets »	44
3.1	Backlog du sprint 2 . . . . .	44
3.2	Spécification fonctionnelle . . . . .	45
3.2.1	Diagramme de cas d'utilisation sprint 2 . . . . .	45
3.2.2	Diagrammes de séquence . . . . .	46
3.3	Réalisation . . . . .	47
3.3.1	Interface de gestion de départements et des projets . .	47
3.4	Rétrospective . . . . .	48
4	Conclusion . . . . .	49
<b>4</b>	<b>Etude et réalisation du release 2</b>	<b>50</b>
1	Introduction . . . . .	51
2	Développement du sprint 1 « Gestion des tests et validation des projets »	51
2.1	Backlog du sprint 1 . . . . .	51
2.2	Spécification fonctionnelle . . . . .	52
2.2.1	Diagramme de cas d'utilisation sprint 1 . . . . .	52
2.2.2	Diagramme de séquence de créer un test . . . . .	53
2.2.3	Diagramme de séquence de publier un projet . . . . .	54
2.2.4	Description textuelle des cas d'utilisation du sprint 1 .	55
2.3	Réalisation . . . . .	56
2.3.1	Interface d'ajout test . . . . .	57
2.3.2	Interface de la liste de projets . . . . .	57
2.4	Rétrospective . . . . .	58
3	Développement du sprint 2 « Postulation pour un projet et Gestion des rendez-vous RH» . . . . .	58
3.1	Backlog du sprint 2 . . . . .	58

---

3.2	Spécification fonctionnelle . . . . .	59
3.2.1	Diagramme de cas d'utilisation «Postuler pour un projet»	59
3.2.2	Diagramme de séquence de «Postuler pour un projet»	60
3.2.3	Diagramme de séquence de «corriger un projet» . . . .	61
3.2.4	Description textuelle des cas d'utilisation du sprint 2 .	61
3.2.5	Diagramme de classes sprint 2 . . . . .	62
3.3	Réalisation . . . . .	62
3.3.1	Interface de postulation . . . . .	62
3.3.2	Interface du test . . . . .	63
3.3.3	Interface de la liste de résultats des candidats . . . . .	65
3.3.4	Interface de la liste de rendez-Vous RH . . . . .	65
3.4	Rétrospective . . . . .	66
4	Conclusion . . . . .	66
<b>Conclusion générale</b>		<b>67</b>
<b>Webographie</b>		<b>68</b>

---

# Liste des figures

1.1	Logo de la société I@T . . . . .	3
1.2	Page d'accueil du site web <i>EmploiTunisie.com</i> . . . . .	7
1.3	Page d'accueil du site web <i>TanitJobs</i> . . . . .	8
1.4	Page d'accueil du site web <i>KeeJob</i> . . . . .	9
1.5	La méthodologie agile « SCRUM » . . . . .	14
2.1	Diagramme de cas d'utilisation général . . . . .	24
2.2	Diagramme de classes général . . . . .	25
2.3	Planification des sprints . . . . .	26
2.4	Logo Visual Studio . . . . .	27
2.5	Logo Postman . . . . .	27
2.6	Logo Bootstrap . . . . .	28
2.7	Logo StarUML . . . . .	28
2.8	Logo Overleaf . . . . .	28
2.9	Caractéristiques de la machine utilisée pour le développement . . . . .	29
2.10	Logo Java . . . . .	30
2.11	Logo Spring Boot . . . . .	30
2.12	Logo Kafka . . . . .	30
2.13	Logo React . . . . .	31
2.14	Logo GitHub . . . . .	31
2.15	Logo Jira . . . . .	31
2.16	Logo IntelliJ IDEA . . . . .	32
2.17	Logo VS Code . . . . .	32
2.18	Logo Postman . . . . .	32



2.19	Logo Keycloak . . . . .	33
2.20	Logo PostgreSQL . . . . .	33
2.21	Architecture physique de l'application TalentCloud . . . . .	34
3.1	Diagramme de cas d'utilisation sprint 1 . . . . .	39
3.2	Diagramme de séquence « S'authentifier » . . . . .	41
3.3	Interface d'accueil . . . . .	42
3.4	Interface de Connexion . . . . .	42
3.5	Interface de création d'un compte pour un candidat . . . . .	43
3.6	Interface d'ajout utilisateur en y affectant un rôle . . . . .	43
3.7	Interface de création de gestion utilisateur . . . . .	44
3.8	Diagramme de cas d'utilisation sprint 2 . . . . .	45
3.9	Diagramme de séquence d'ajout d'un projet . . . . .	46
3.10	Diagramme de séquence de suppression d'un projet . . . . .	47
3.11	Interface de gestion département . . . . .	48
3.12	Interface de gestion projet . . . . .	48
4.1	Diagramme de use case . . . . .	53
4.2	Diagramme de séquence de créer test . . . . .	54
4.3	Diagramme de séquence de publier projet . . . . .	55
4.4	Interface du test . . . . .	57
4.5	Interface de la liste des projets validés et refusés . . . . .	57
4.6	Diagramme de cas d'utilisation « Postuler pour un projet » . . . . .	59
4.7	Diagramme de séquence « Postuler pour un projet » . . . . .	60
4.8	Diagramme de séquence « corriger un projet » . . . . .	61
4.9	Diagramme de classes sprint 2 . . . . .	62
4.10	Interface de postulation . . . . .	63
4.11	Interface contenant le test . . . . .	64
4.12	Resultat du test . . . . .	65
4.13	Interface de la liste de résultats des candidats. . . . .	65
4.14	Interface de la liste de rendez-Vous RH. . . . .	66

---

# Liste des tableaux

1.1	Fiche signalétique de l'entreprise Tekboot Solutions . . . . .	5
1.2	Tableau comparatif des différents sites . . . . .	11
2.1	Backlog des fonctionnalités – Admin . . . . .	21
2.2	Backlog des fonctionnalités – Client . . . . .	22
2.3	Backlog des fonctionnalités – Candidat . . . . .	23
3.1	Backlog du sprint 1- Release 1 . . . . .	39
3.2	Description textuelle du cas d'utilisation « S'inscrire » . . . . .	40
3.3	Description textuelle du cas d'utilisation « S'authentifier » . . . . .	40
3.4	Backlog du deuxième sprint . . . . .	45
4.1	Backlog su sprint 1- Release 2 . . . . .	52
4.2	Description textuelle du cas d'utilisation « Ajout Test» . . . . .	56
4.3	Description textuelle du cas d'utilisation « Ajout Test» . . . . .	56
4.4	Backlog du sprint 2- Release 2 . . . . .	59
4.5	Description textuelle du cas d'utilisation « Réserver une date pour un rendez-Vous RH» . . . . .	61

## Introduction Générale

Dans un contexte où la transformation numérique redéfinit les modes de gestion au sein des entreprises, la recherche de solutions technologiques efficaces devient primordiale, notamment dans le domaine des ressources humaines. Le recrutement, en particulier, nécessite des outils modernes capables de s'adapter aux nouveaux défis liés à la compétitivité, à l'analyse de données et à l'automatisation des processus.

C'est dans cette optique que nous avons développé **TalentCloud**, une plateforme innovante dédiée à l'optimisation du processus de recrutement. Elle repose sur une architecture avancée basée sur des microservices pour le backend, associée à un frontend monolithique, offrant ainsi flexibilité, évolutivité et facilité de maintenance. La solution permet une analyse approfondie des performances et une prise de décision éclairée grâce à l'exploitation des données.

Le développement de TalentCloud s'appuie sur des méthodologies modernes garantissant un cycle de vie logiciel agile, des déploiements continus, ainsi qu'une haute disponibilité du système. Cette approche assure une expérience utilisateur fluide tout en répondant aux exigences techniques et fonctionnelles du secteur.

Ce rapport présente notre travail structuré autour de quatre chapitres :

- **Contexte général** : introduction au domaine du recrutement numérique, présentation du projet TalentCloud, problématique et méthodologie de développement adoptée.
- **Spécification des besoins** : identification des besoins fonctionnels (gestion des candidatures, tableau de bord analytique, alertes) et non fonctionnels (sécurité, performance, scalabilité), accompagnée de diagrammes d'analyse.
- **Conception** : description de l'architecture modulaire, des choix technologiques, et des modèles de données assurant la cohérence du système.
- **Réalisation** : présentation de l'environnement technique (frameworks, bases de données, services), illustrée par des interfaces clés de la plateforme.

Enfin, une conclusion générale mettra en lumière les résultats obtenus et proposera des perspectives d'évolution pour faire de TalentCloud un acteur majeur du recrutement intelligent.

---

# Chapitre 1

## Cadre général du projet

### Sommaire

1	Introduction . . . . .	17
2	Description fonctionnelle des besoins . . . . .	17
2.1	Identification des acteurs . . . . .	17
2.2	Capture des besoins fonctionnels . . . . .	18
2.3	Capture des besoins non fonctionnels . . . . .	19
3	Pilotage du projet avec SCRUM . . . . .	19
3.1	Équipe et rôles . . . . .	19
3.2	Backlog du produit . . . . .	21
3.3	Diagramme de cas d'utilisation général . . . . .	23
3.4	Diagramme de classes général . . . . .	25
3.5	Planification des sprints . . . . .	26
4	Environnement de travail . . . . .	27
4.1	4.1 Environnement logiciel . . . . .	27
4.2	4.2 Environnement matériel . . . . .	28
4.3	4.3 Technologies utilisées . . . . .	29
5	Spécification architecturale . . . . .	33
5.1	Architecture physique . . . . .	33
5.2	Architecture logique . . . . .	34
5.3	Interaction entre les composants . . . . .	35
6	Conclusion . . . . .	36

# 1 Introduction

Ce chapitre constitue une étude préliminaire du projet. Il débute par une présentation de l'entreprise hôte, suivie d'une analyse de l'existant afin de mieux comprendre le contexte et les contraintes du projet. Ensuite, les différentes méthodologies seront explorées avant de justifier le choix de celle adoptée. Enfin, la phase de conception sera détaillée, mettant en avant l'architecture et les principales composantes de la solution proposée.

## 2 Entreprise d'accueil

### 2.1 Présentation de Tekboot Solutions

Tekboot Solutions est une entreprise technologique basée à Bruxelles, spécialisée dans le développement logiciel, les architectures microservices, le DevOps, et l'intégration de solutions cloud modernes. Créée en 2020, elle accompagne les entreprises dans leur transformation digitale en leur fournissant des solutions innovantes, scalables et sécurisées.

Tekboot agit à la fois comme startup studio (en développant ses propres produits) et ESN (Entreprise de Services du Numérique), en mettant à disposition des talents pour des missions techniques en Belgique et à l'international.

Sa mission est d'autonomiser les entreprises et les particuliers à réaliser leurs aspirations numériques et à prospérer dans un monde technologique en constante évolution.»[1].

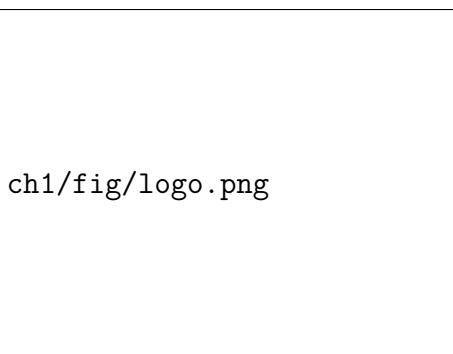


FIGURE 1.1 – Logo de la société I@T

## 2.2 Secteurs d'activités

Nous présentons, ici, différents services de Tekboot Solutions :

## 2.3 Spécialisations

- Front-end development (HTML/CSS/JavaScript).
- Back-end development (Node.js/Express.js).
- Full-stack development.
- Développement mobile (Android/iOS, Flutter).
- Développement de jeux (Unity).
- Développement d'applications web et e-commerce.
- Cloud computing (AWS, Google Cloud).
- DevOps et CI/CD.
- Gestion agile de projet et design UI/UX.

## 2.4 Les services de l'entreprise

- **Développement** : Conception et réalisation d'applications web et mobiles en adoptant des technologies modernes et évolutives.
- **Architecture logicielle** : Élaboration de systèmes robustes, modulaires et maintenables grâce à des architectures logicielles bien pensées.
- **Exploitation des données** : Création de tableaux de bord décisionnels et extraction d'indicateurs clés pour guider les décisions stratégiques.
- **Sécurité** : Mise en place de mécanismes de contrôle d'accès stricts et de pratiques assurant la protection des données.
- **Qualité logicielle** : Intégration de tests automatisés et d'analyses continues du code afin de garantir la fiabilité et la performance des solutions livrées.

## 2.5 Fiche signalétique

Dénomination sociale	Tekboot Solutions
Forme juridique	Société privée
Date de création	2020
Secteur d'activité	Technologies de l'information et du numérique
Activité principale	Développement web, mobile et exploitation des données
Directeur général	Non spécifié
Adresse	Boulevard Bischoffsheim 39, 1000 Bruxelles, Belgique
Téléphone	Non spécifié
Site Web	<a href="https://tekboot.io">https://tekboot.io</a>
Effectif total	11-50 employés

TABLE 1.1 – Fiche signalétique de l'entreprise Tekboot Solutions

## 3 Cadre du projet

### 3.1 Contexte Général et problématique

#### 3.1.1 Présentation du projet

Le projet **TalentCloud** vise à créer une plateforme moderne dédiée à la gestion des carrières et au recrutement. Cette plateforme a pour ambition de faciliter les échanges entre les candidats et les entreprises, tout en accompagnant les différents acteurs tout au long du processus de recrutement. Elle offre un environnement intuitif et adaptable, permettant de mieux gérer les talents, de suivre les candidatures et d'optimiser l'ensemble des étapes liées à l'emploi.

#### 3.1.2 Problématique

Les solutions traditionnelles de recrutement manquent d'automatisation et présentent de nombreuses lacunes, notamment en matière de validation des données, de sécurité et de performance. Elles peinent à s'adapter à la croissance rapide des besoins des entreprises modernes. Cette absence de fiabilité et de flexibilité impacte négativement l'efficacité des processus RH. Il est donc impératif d'introduire une solution technologique innovante capable de pallier ces limites, tout en assurant une expérience utilisateur optimale.

## 3.2 Étude de l'existant

L'étude de l'existant est une phase primordiale pour la réalisation d'un projet. Il s'agit en effet d'observer et d'analyser les solutions existantes puis d'identifier leurs points faibles et leurs points forts afin d'en tenir compte lors de la mise en place de l'application concernée. Il se compose de trois parties : description de l'existant, critique de l'existant et proposition d'une solution.

### 3.2.1 EmploiTunisie.com

*EmploiTunisie.com* est l'un des principaux sites de recrutement en Tunisie. Il propose des offres d'emploi dans divers secteurs et niveaux d'expérience, ainsi que des offres de stages pour les étudiants. Les entreprises peuvent également publier des annonces d'emploi et rechercher des CV en ligne. Le site est convivial et facile à naviguer. La Figure 1.2 présente la page d'accueil du site web *EmploiTunisie* [**emploitunisie**].



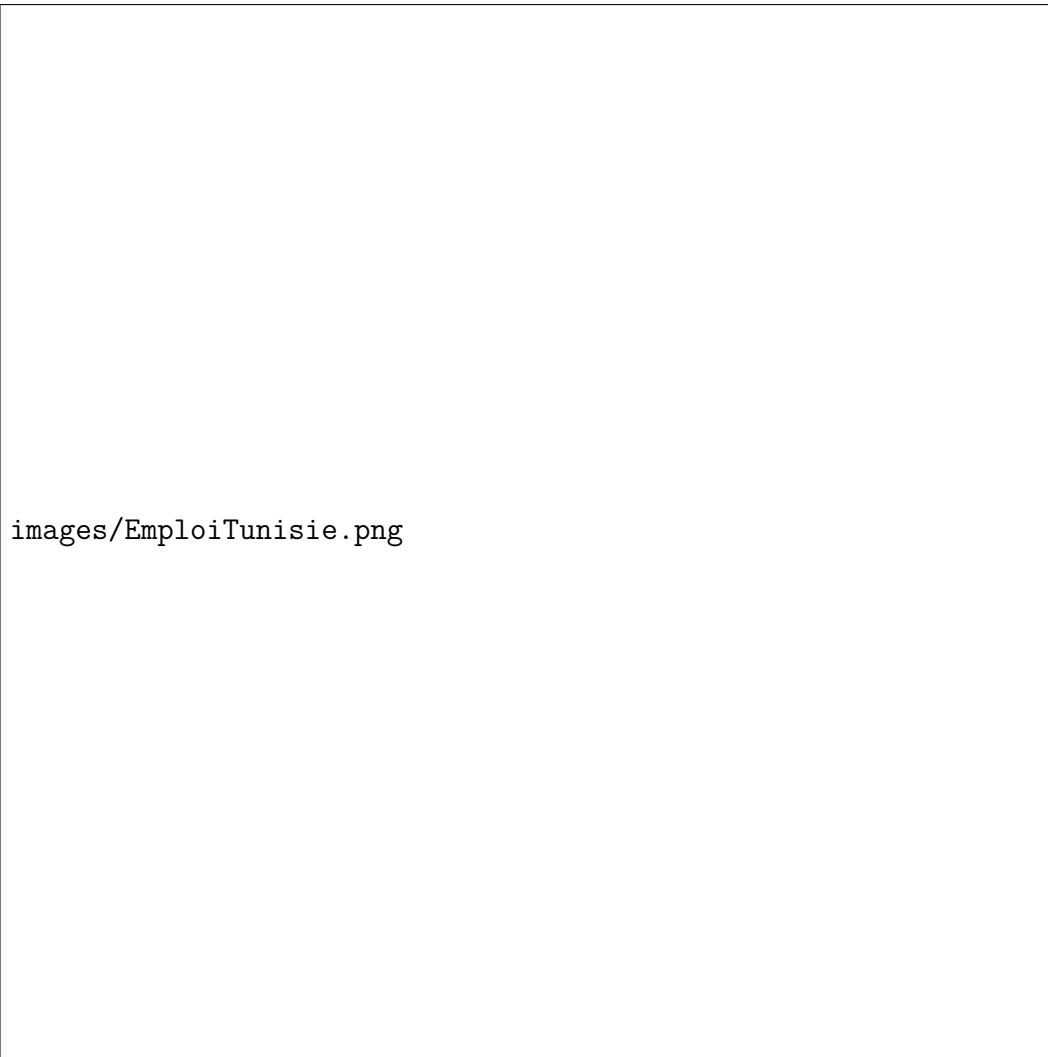


FIGURE 1.2 – Page d’accueil du site web *EmploiTunisie.com*

### **Avantages**

- Un large éventail d’annonces d’offres de stages est disponible.
- Un système de filtrage avancé pour affiner les résultats de recherche.
- Une option de dépôt de CV en ligne pour les candidats.

### **Inconvénients**

- Les fonctionnalités sont limitées pour les recruteurs.
- Les annonces de stages ne sont pas toujours vérifiées.
- Un manque de transparence concernant les informations sur les étudiants et les recruteurs.

### 3.2.2 TanitJobs.com

*TanitJobs.com* C'est également un site de recrutement en Tunisie bien connu, proposant des offres de stages dans divers secteurs et régions. Le site offre également des services tels que la publication de CV et des alertes e-mail pour les offres de stages pertinentes. Tanit-jobs.com est réputé pour son interface utilisateur intuitive et sa grande base de données de CV. La Figure 1.3 présente la page d'accueil du site web *TanitJobs* [TanitJobs].

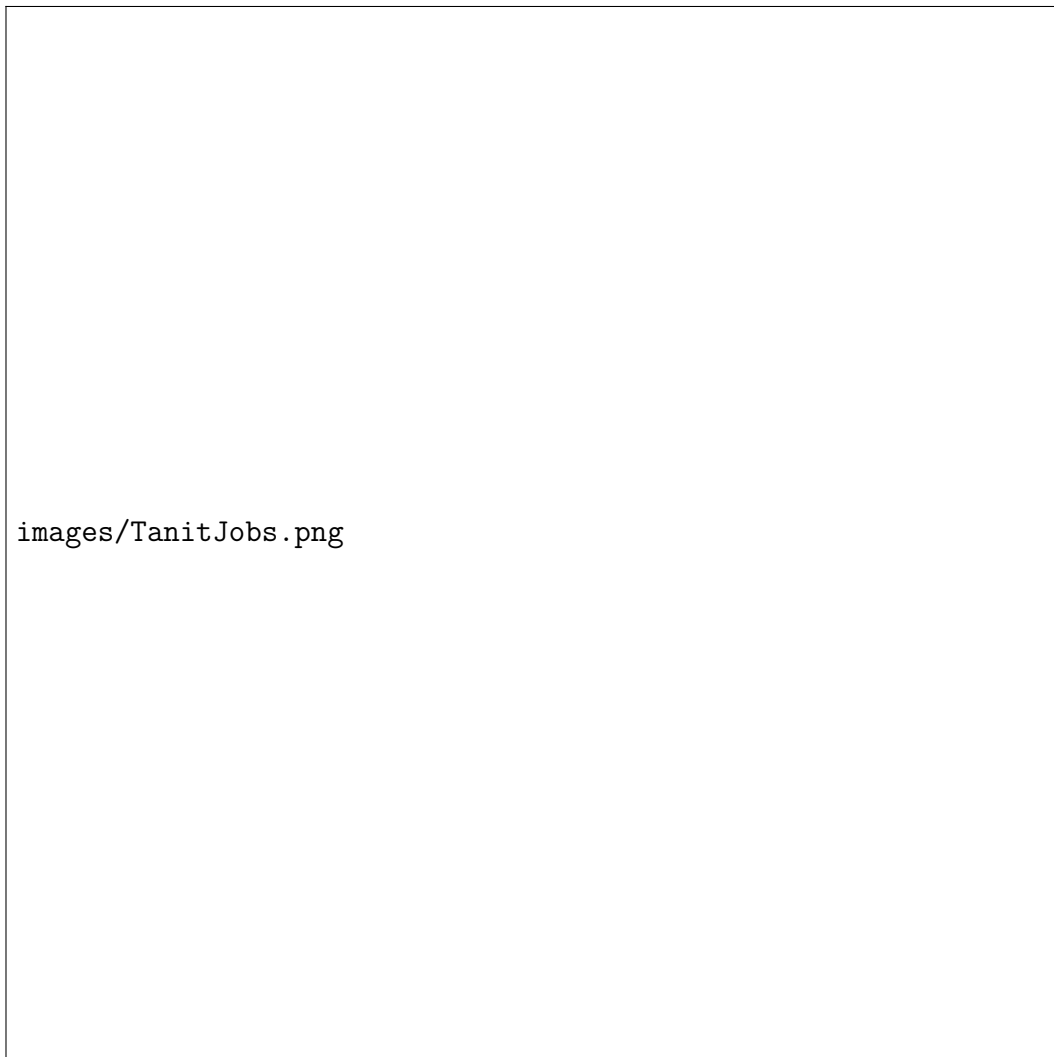


FIGURE 1.3 – Page d'accueil du site web *TanitJobs*

#### Avantages

- Une interface utilisateur simple et facile à utiliser.
- Un large éventail d'annonces d'offres de stages est disponible.
- Une bonne réputation et notoriété sur le marché tunisien.

### Inconvénients

- Le manque de transparence concernant les informations sur les recruteurs.
- Le coût relativement élevé pour les recruteurs pour poster des annonces d'emploi.
- Le manque d'interactivité et d'engagement avec les utilisateurs.

### 3.2.3 KeeJob.com

*KeeJob.com* C'est une autre plateforme de recrutement en ligne populaire en Tunisie. Le site propose des offres de stages dans divers secteurs et niveaux d'expérience, ainsi que des fonctionnalités pour la publication de CV et la recherche d'emploi. Keejob.com se distingue également par son interface utilisateur conviviale et sa compatibilité mobile. La Figure 1.4 présente la page d'accueil du site web *KeeJob* [**KeeJob.com**].

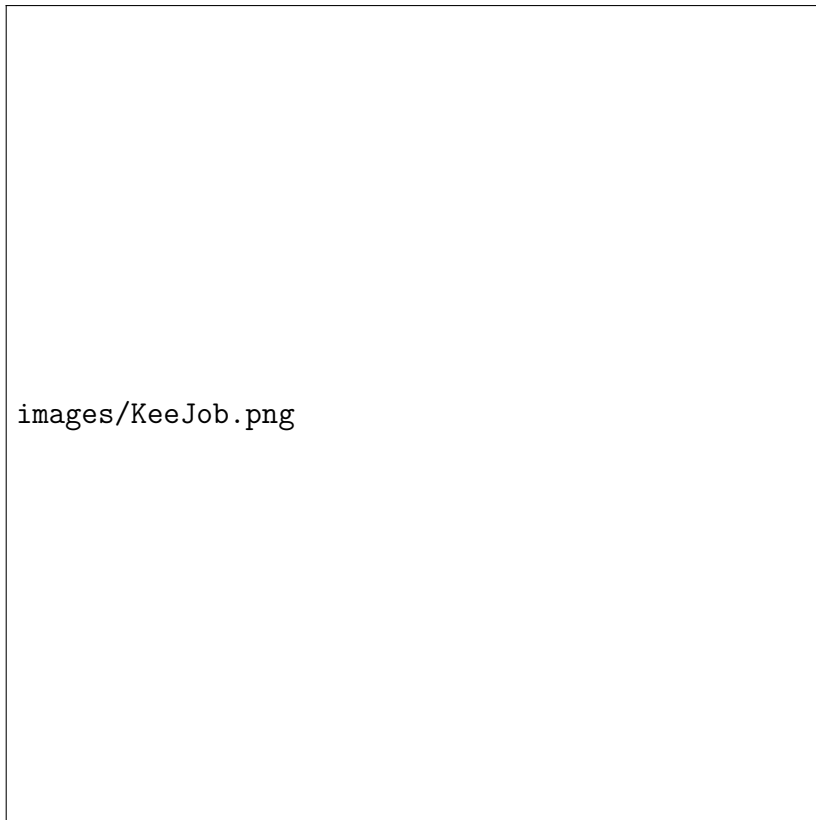


FIGURE 1.4 – Page d'accueil du site web *KeeJob*

### Avantages

- Un système de filtrage avancé pour affiner les résultats de recherche.
- Des annonces d'emploi vérifiées.
- Une option de CV en ligne pour les candidats.

### **Inconvénients**

- Une interface utilisateur un peu désuète.
- Un manque d’interactivité et d’engagement avec les utilisateurs.
- sUn coût élevé pour les recruteurs pour poster des annonces d’emploi.

## **3.3 Critique des solutions existantes**

Dans cette partie, nous allons présenter sous forme d’un tableau comparatif une synthèse résultante des solutions existantes, cette synthèse est faite selon des critères d’évaluation de ces solutions.

Le tableau 1.2 présente une étude comparative des différents sites.

array

Critères	EmploiTunisie	KeeJob	TanitJobs	TalentCloud
C1 - Validation et vérification rigoureuse des données des candidats et des entreprises	X	X	X	✓
C2 - Gestion fine de la visibilité des profils avec contrôle d'accès	X	X	X	✓
C3 - Intégration d'un système d'analyse et de matching intelligent	X	X	X	✓
C4 - Publication et gestion des offres d'emploi	X	X	X	✓
C5 - Interface utilisateur intuitive et responsive	X	X	X	✓
C6 - Système de notification en temps réel (candidatures, réponses)				✓
C7 - Suivi et gestion des entretiens RH				✓
C8 - Outils d'analyse statistique des candidatures				✓

TABLE 1.2 – Tableau comparatif des différents sites

Après cette analyse comparative, il apparaît que les plateformes actuelles présentent

des limites importantes, notamment en matière de fiabilité des données, de gestion de la confidentialité et de pertinence des mises en relation entre candidats et recruteurs. Notre projet TalentCloud se distingue précisément par l'intégration de mécanismes robustes de validation des informations, un contrôle avancé sur la visibilité des profils, ainsi qu'un système intelligent de matching qui optimise les opportunités pour chaque utilisateur. Ces éléments clés visent à offrir une expérience plus sécurisée, transparente et efficace, répondant ainsi aux besoins non satisfaits des solutions existantes.

## 4 Solution proposée

La solution proposée vise à répondre de manière globale et innovante aux défis rencontrés dans le domaine de la gestion des talents et du recrutement. Elle place l'expérience utilisateur au cœur de ses priorités, en offrant un environnement flexible et sécurisé où chaque acteur peut interagir de façon transparente et efficace. Cette plateforme favorise une meilleure adéquation entre les profils des candidats et les attentes des recruteurs grâce à une approche centrée sur la qualité des données et la pertinence des échanges. Par ailleurs, elle encourage la confiance et l'engagement en assurant une gestion rigoureuse des informations personnelles, tout en facilitant le suivi et la progression des candidats tout au long du processus de recrutement. En combinant ces aspects, la solution ambitionne de transformer et d'optimiser durablement les pratiques de recrutement, au bénéfice de l'ensemble des utilisateurs.

## 5 Etude de la méthodologie de développement

La méthodologie de travail est un élément essentiel à la réussite de tout projet. Il existe plusieurs approches, chacune étant adaptée à des contextes et besoins spécifiques. Parmi les plus connues, on peut citer le **modèle en cascade**, qui est particulièrement efficace lorsque les besoins du projet sont clairement définis dès le départ. On trouve également la **méthodologie Scrum**, qui propose une approche itérative et flexible, favorisant l'adaptation continue aux évolutions du projet. Enfin, la **méthodologie 2TUP** répond aux contraintes de changement continu imposées aux systèmes d'information des entreprises, en proposant une démarche centrée sur l'adaptabilité.

## 5.1 Choix de la méthodologie de développement

Après avoir étudié les différentes méthodologies de développement logiciel, nous avons opté pour Scrum en tant que méthodologie de développement agile pour la conception et le développement de notre plateforme.

## 5.2 Avantages de Scrum

La méthodologie **Scrum** s'adapte parfaitement à la décomposition de notre projet. Elle repose sur plusieurs avantages clés qui en font une approche particulièrement efficace pour le développement logiciel :

- **Flexibilité** : Scrum peut être utilisée pour différents types de projets, quelle que soit leur nature ou leur complexité.
- **Collaboration et communication** : Elle favorise un fort niveau d'interaction entre les membres de l'équipe, ce qui contribue à une meilleure qualité du travail fourni.
- **Cycles courts et itératifs** : Grâce à des sprints de courte durée, Scrum permet de livrer rapidement des fonctionnalités, tout en intégrant régulièrement les retours des utilisateurs pour améliorer le produit.
- **Satisfaction du client** : La transparence du processus et sa capacité à s'adapter aux besoins permettent de mieux répondre aux attentes du client.
- **Adaptabilité aux changements** : Scrum offre une grande réactivité face aux imprévus ou aux évolutions des exigences, ce qui est essentiel dans les projets dynamiques.

## 5.3 Mise en pratique de la méthodologie Scrum

Dans une approche Scrum, le projet est structuré autour de cycles de développement appelés **sprints**, dont la durée varie généralement entre deux et quatre semaines. Chaque sprint vise à livrer un incrément fonctionnel du produit.

Le projet commence par un **sprint 0**, qui constitue une phase préparatoire essentielle. Ce sprint est consacré à la mise en place de l'architecture du projet, à la configuration des environnements de développement, ainsi qu'à l'installation des outils nécessaires pour le suivi des tâches et l'intégration continue.

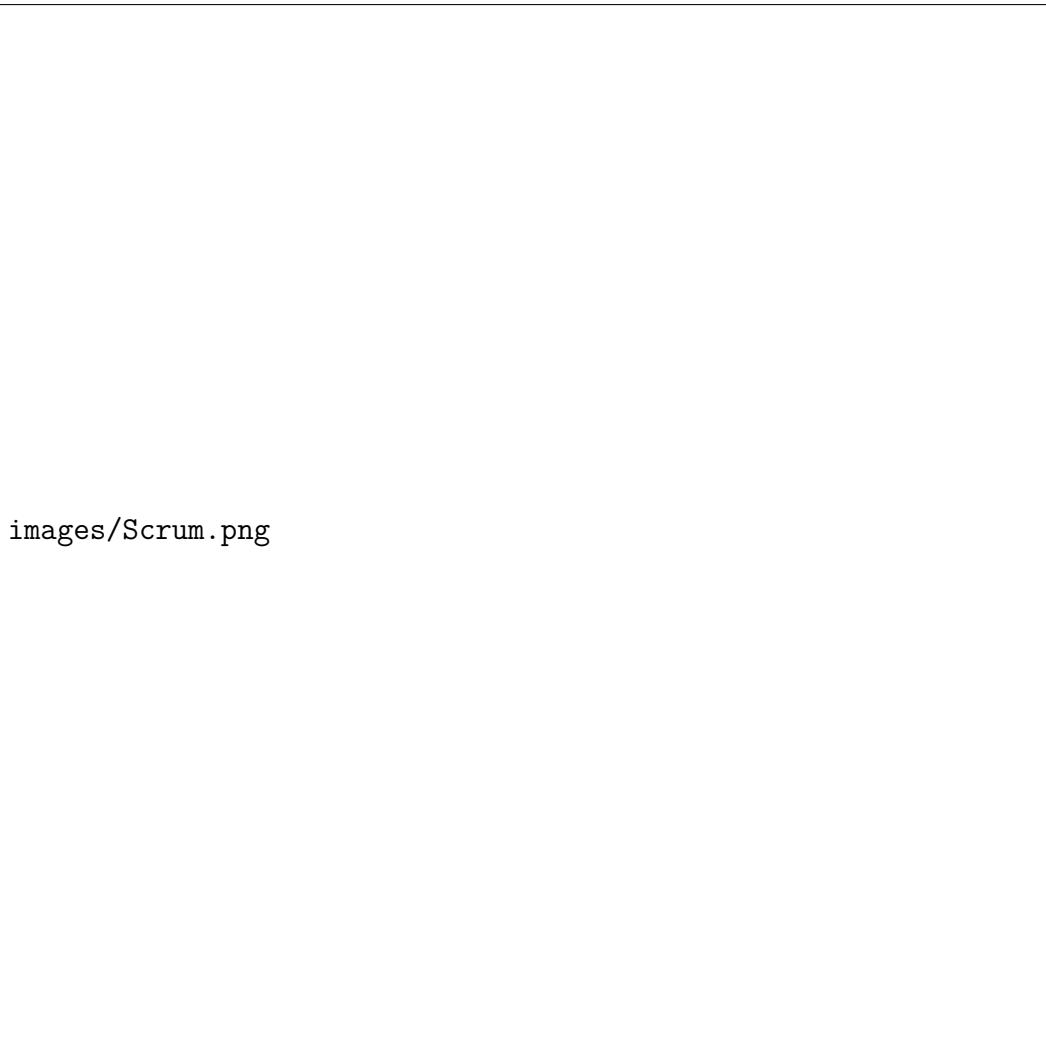


FIGURE 1.5 – La méthodologie agile « SCRUM »

Comme illustré dans la figure 1.5, la méthodologie **Scrum** repose sur trois éléments fondamentaux : les **sprints et leurs événements**, les **trois rôles principaux** (Product Owner, Équipe de Développement, Scrum Master), et le **backlog produit**.

- **Le Product Owner** : il représente les clients et utilisateurs du produit. Il est chargé de gérer le backlog produit, de prioriser les fonctionnalités et de s'assurer que les besoins métier sont bien pris en compte.
- **L'équipe de développement** : elle transforme les besoins exprimés par le Product Owner en fonctionnalités concrètes. Composée de développeurs, testeurs, designers et autres contributeurs, elle est responsable de livrer, à la fin de chaque sprint, les éléments définis comme prioritaires.
- **Le Scrum Master** : garant du bon déroulement de la méthodologie Scrum, il joue un rôle de coach auprès du Product Owner et de l'équipe de développement.



Il veille à la bonne application des principes Scrum et à la résolution des obstacles.

- **Le Sprint** : chaque itération (appelée sprint) a une durée fixe, généralement entre deux et quatre semaines, et reste constante tout au long du projet. Cette régularité permet de maintenir un rythme soutenu et de produire des indicateurs fiables. Le projet débute souvent par un **sprint 0**, consacré aux préparatifs : mise en place de l'architecture, des environnements et des outils.
- **Le Product Backlog** : il s'agit d'une liste priorisée des fonctionnalités et exigences du produit. Ce backlog est évolutif et géré par le Product Owner, qui y classe les éléments selon leur valeur métier, le niveau de risque, l'effort requis, et la maturité technique ou fonctionnelle.
- **La Sprint Review** : réunion organisée à la fin de chaque sprint, elle permet à l'équipe Scrum et aux parties prenantes de passer en revue l'incrément de produit livré. Sa durée maximale est généralement de quatre heures.
- **La Sprint Retrospective** : cette réunion interne à l'équipe Scrum vise à analyser le sprint écoulé pour en tirer des enseignements. Elle permet d'identifier ce qui a bien fonctionné, ce qui peut être amélioré, et d'adapter les processus de développement en conséquence.

## 6 Conclusion

Ce chapitre a posé les bases du projet en précisant son contexte, ses objectifs, et la méthodologie appliquée pour garantir un développement fluide et structuré de l'application TalentCloud. Le chapitre suivant abordera les spécifications techniques et les choix technologiques retenus pour la mise en œuvre de la solution.

---

# Chapitre 2

## Analyse et spécification des besoins

### Sommaire

1	Introduction . . . . .	<b>38</b>
2	Développement du sprint 1 « Sécurité et gestion d'utilisateurs et leurs rôles » . . .	<b>38</b>
2.1	Backlog du sprint 1 . . . . .	38
2.2	Spécification fonctionnelle . . . . .	39
2.3	Réalisation . . . . .	41
2.4	Rétrospective . . . . .	44
3	Développement du sprint 2 « Gestion des départements et des projets » . . . . .	<b>44</b>
3.1	Backlog du sprint 2 . . . . .	44
3.2	Spécification fonctionnelle . . . . .	45
3.3	Réalisation . . . . .	47
3.4	Rétrospective . . . . .	48
4	Conclusion . . . . .	<b>49</b>

## 1 Introduction

Dans le chapitre précédent, nous avons abordé le choix de Scrum comme méthodologie de travail. Dans ce chapitre, nous présentons la première phase, appelée "Sprint Zéro", où nous effectuons la planification et l'organisation du projet. Pendant ce sprint, nous identifions les acteurs, les besoins fonctionnels et non fonctionnels, réalisons le "Product Backlog" et planifions les sprints et les releases. Le Sprint Zéro pose les bases du projet et prépare le terrain pour le développement ultérieur de l'application.

## 2 Description fonctionnelle des besoins

Généralement, les besoins de chaque application sont de trois types : les besoins fonctionnels, les besoins non fonctionnels et les besoins techniques.

### 2.1 Identification des acteurs

L'objectif de cette section est de décrire les différents acteurs impliqués dans le fonctionnement de notre application, ainsi que leurs rôles et fonctionnalités respectifs. Un acteur est défini comme étant un individu, une machine ou un système qui n'est pas directement impliqué dans la solution que nous développons, mais qui interagit avec celle-ci pour permettre son bon fonctionnement global. Les acteurs principaux de cette application sont :

- **Admin** : Cette personne sera responsable de la validation des documents ainsi que des profils des candidats et des clients sur la plateforme. Elle assure également la gestion des utilisateurs avec leurs rôles, garantissant ainsi un contrôle approprié des accès et des permissions au sein du système. .

- **Client** : CLe client est responsable de la création des offres de travail sur la plateforme. Il peut consulter la liste des candidats ayant postulé à ses offres, ainsi qu'accepter ou rejeter les candidatures. De plus, il a la possibilité de planifier des rendez-vous avec les candidats sélectionnés en vue de leur recrutement. .

- **Candidate** : Le candidat est un utilisateur qui crée et complète son profil sur la plateforme afin de présenter ses compétences et expériences. Il peut consulter l'ensemble des offres de travail publiées, sélectionner celles qui correspondent à ses aspirations et

postuler directement aux annonces qui l'intéressent. Cette démarche lui permet de participer activement au processus de recrutement en ligne.

## 2.2 Capture des besoins fonctionnels

Les besoins fonctionnels d'une application sont étroitement liés aux tâches que l'utilisateur souhaite effectuer. Ces besoins sont liés aux fonctionnalités que l'utilisateur attend de l'application pour réaliser ces tâches. Dans le cadre de notre projet, voici les principales fonctionnalités que nous envisageons de proposer :

- **Admin** :

- Ajouter, modifier et supprimer des utilisateurs (candidats, clients).
- Valider ou rejeter les profils des candidats et gérer l'historique des candidatures et des documents associés.
- Configurer les paramètres de sécurité, y compris l'authentification à deux facteurs (2FA).

candidate.

- **candidate** :

- S'inscrire pour créer un compte utilisateur sur la plateforme.
- S'authentifier pour accéder à leur espace personnel sécurisé.
- Créer et gérer leur profile en ajoutant des informations personnelles, compétences, expériences et documents comme les certifications.
- Rechercher des offres d'emploi et postuler directement via la plateforme.
- Suivre le statut de leurs candidatures en temps réel.

- **client** :

- S'inscrire pour créer un compte utilisateur sur la plateforme.
- S'authentifier pour accéder à leur espace personnel sécurisé.
- Créer, publier et gérer des offres d'emploi.

- Examiner les candidatures, analyser les profils des candidats et effectuer un classement.
- Contacter les candidats, négocier les offres et gérer les retours d'expérience.

## 2.3 Capture des besoins non fonctionnels

Les besoins non fonctionnels sont les exigences qui garantissent la qualité et l'efficacité du système. Ils ne sont pas liés directement aux fonctionnalités de l'application, mais ils sont tout aussi importants pour répondre aux attentes des utilisateurs et pour assurer le succès du projet. Ces exigences incluent des critères de performance, d'esthétique et de sécurité qui doivent être respectés pour garantir une expérience utilisateur optimale et la fiabilité du système.

• **La convivialité** : La plateforme doit être facile à utiliser et intuitive pour les utilisateurs. Les interfaces utilisateur doivent être conçues pour permettre une navigation facile et une recherche rapide des informations pertinentes.

• **La performance** : La plateforme doit être capable de gérer de nombreux utilisateurs simultanément et d'effectuer des tâches rapidement pour assurer une expérience utilisateur fluide.

• **La sécurité** : Les utilisateurs doivent être assurés que leurs informations personnelles et leurs données de connexion sont en sécurité. La plateforme doit être conforme aux normes de sécurité et aux règles de protection des données pour assurer la confidentialité, l'intégrité et la disponibilité des données des utilisateurs.

• **La compatibilité de l'application** : L'application doit être compatible avec différents navigateurs web, différents systèmes d'exploitation et différents types de périphériques. Ainsi, les utilisateurs peuvent accéder à l'application à partir de n'importe quel appareil ou navigateur sans rencontrer de problèmes d'affichage ou de fonctionnement.

## 3 Pilotage du projet avec SCRUM

### 3.1 Équipe et rôles

Dans le cadre de notre projet, plusieurs équipes spécialisées interviennent afin d'assurer le bon déroulement du développement, du déploiement et de l'analyse des don-

nées :

**Équipe IA** Dans ce projet, l'équipe IA développe un système de matching qui analyse les profils des candidats et les offres d'emploi pour proposer des suggestions personnalisées sur la page d'accueil. Elle conçoit également un modèle d'analyse comportementale par reconnaissance faciale durant les entretiens, fournissant des statistiques sur le comportement et la confiance du candidat. En outre, l'IA extrait et analyse des données des profils, tests et interactions pour générer des rapports utiles aux recruteurs. Enfin, elle automatise des tâches répétitives comme la présélection, l'évaluation initiale et la gestion des rendez-vous, rendant le processus de recrutement plus efficace et personnalisé.

**Équipe DevOps** L'équipe DevOps est chargée d'automatiser, de superviser et d'optimiser le cycle de vie complet du déploiement de l'application. Elle met en place des pipelines d'intégration et de déploiement continu (CI/CD) pour assurer la livraison rapide et fiable des nouvelles fonctionnalités. Elle gère les environnements de développement, de test et de production, tout en surveillant les performances et la disponibilité du système. En automatisant les processus de build, test et déploiement, l'équipe DevOps garantit la stabilité, la scalabilité et la sécurité de la plateforme tout au long du projet.

**Équipe Business Intelligence (BI)** L'équipe BI est chargée de collecter, traiter et analyser les données générées par l'application. Elle conçoit des tableaux de bord, des rapports statistiques et des indicateurs clés de performance (KPI) qui aident à suivre l'avancement du projet, la satisfaction des utilisateurs et l'efficacité des processus. Ces analyses permettent d'orienter les décisions stratégiques et d'optimiser les fonctionnalités proposées.

### **Équipe SCRUM :**

Pour notre application, les rôles sont répartis comme suit :

Product Owner : M. Khaled Fourati

Scrum Master : Mme Sondos Aouachri

Dev Team : Mr Bacem Ben Akkari

## 3.2 Backlog du produit

Une fois l'analyse des besoins est entamée, nous les transformons en ce qu'on appelle des « User Stories » qui seront détaillées dans le « Product Backlog ». Chaque « User Story » permet de décrire avec suffisamment de précision le contenu d'une fonctionnalité à développer.

À ce niveau, en tant que PO nous devons alors nous demander par quoi commencer ? et pourquoi une user story est plus prioritaire qu'une autre ?

« MoSCoW » [1] est l'une des techniques souvent empruntée sur le terrain par les entreprises pour prioriser le Product Backlog. Elle consiste à juger de l'importance d'un Product Backlog Item (PBI : un constituant du BP) pour le profit du client et l'entreprise de développement.

Nous la retenons pour prioriser notre PB :

- M - Must have : doit être réalisée.
- S - Should have : devrait être réalisée si possible.
- Co - Could have : pourrait être réalisée s'il n'y a pas d'impact sur les tâches courantes.
- W - Would have : ne sera pas réalisée tout de suite mais serait souhaitable pour une version ultérieure.

Le tableau 2.1 présente la liste des scénarios utilisateurs.

Tâches	Description	Priorité
<b>Admin</b>		
S'inscrire	En tant qu'admin, je peux m'inscrire sur la plateforme.	M
S'authentifier	En tant qu'admin, je peux m'authentifier pour accéder à mon espace sécurisé.	M
Gérer les utilisateurs et leurs rôles	En tant qu'admin, je peux attribuer, modifier ou révoquer les rôles des utilisateurs (admin, client, candidat).	M
Valider les documents et profils	En tant qu'admin, je peux valider les documents soumis et les profils des candidats et clients.	M

TABLE 2.1 – Backlog des fonctionnalités – Admin

Tâches	Description	Priorité
<b>Client</b>		
S'inscrire	En tant que client, je peux créer un compte sur la plateforme.	M
S'authentifier	En tant que client, je peux m'authentifier pour accéder à mon espace personnel.	M
Modifier mes informations	En tant que client, je peux modifier mes informations personnelles et les détails de mon entreprise.	M
Créer et gérer les offres d'emploi	En tant que client, je peux créer, modifier et publier des offres d'emploi.	M
Consulter les candidatures	En tant que client, je peux consulter les candidatures reçues pour mes offres.	M
Filtrer et rechercher des candidats	En tant que client, je peux filtrer les candidatures selon des critères spécifiques.	M
Accepter ou rejeter des candidats	En tant que client, je peux accepter ou rejeter un candidat après étude de sa candidature.	M
Recevoir des notifications	En tant que client, je peux recevoir des notifications lors de nouvelles postulations et de validation de mon profile par l'admin.	M
Consulter les statistiques détaillées	En tant que client, je peux voir les statistiques de mes offres et consulter la liste des candidats qui ont postulé.	M

TABLE 2.2 – Backlog des fonctionnalités – Client




Tâches	Description	Priorité
<b>Candidat</b>		
S'inscrire	En tant que candidat, je peux créer un compte sur la plateforme.	M
S'authentifier	En tant que candidat, je peux m'authentifier pour accéder à mon espace personnel.	M
Créer et gérer mon profil	En tant que candidat, je peux compléter et modifier mon profil, y compris ajouter CV et autres documents.	M
Consulter les offres d'emploi	En tant que candidat, je peux consulter les offres d'emploi disponibles.	M
Rechercher et filtrer les offres	En tant que candidat, je peux rechercher des offres selon mes critères (localisation, domaine, salaire).	M
Postuler aux offres	En tant que candidat, je peux postuler aux offres d'emploi en soumettant ma candidature.	M
Suivre mes candidatures	En tant que candidat, je peux consulter le statut de mes candidatures en cours.	M
Recevoir des notifications	En tant que candidat, je peux recevoir des notifications sur l'évolution de mes candidatures.	S

TABLE 2.3 – Backlog des fonctionnalités – Candidat

### 3.3 Diagramme de cas d'utilisation général

Chaque cas d'utilisation représente une fonctionnalité fournie afin de produire un résultat souhaité. Par conséquent, le diagramme de cas d'utilisation décrit l'interaction entre le système et ce qui l'entoure en déterminant les besoins et tout ce que le système doit faire à ces acteurs.

La figure 2.1 présente le diagramme de cas d'utilisation général de notre système.



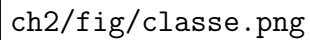
ch2/fig/usegeneral.png

FIGURE 2.1 – Diagramme de cas d'utilisation général

### 3.4 Diagramme de classes général

Le diagramme de classes permet de représenter l'application dans son ensemble. Chaque classe contient des données qui lui sont associées. Elle entre en interaction avec toutes les autres classes tout en décrivant les traitements entre elles.

La figure 2.2 présente le diagramme de classes général de notre application qui présente toutes les classes de notre projet, leurs compositions et les différentes relations entre eux.

The figure is a large rectangular box containing the text 'ch2/fig/classe.png'. This indicates that the actual UML class diagram is not rendered in this view but is located at the specified file path.

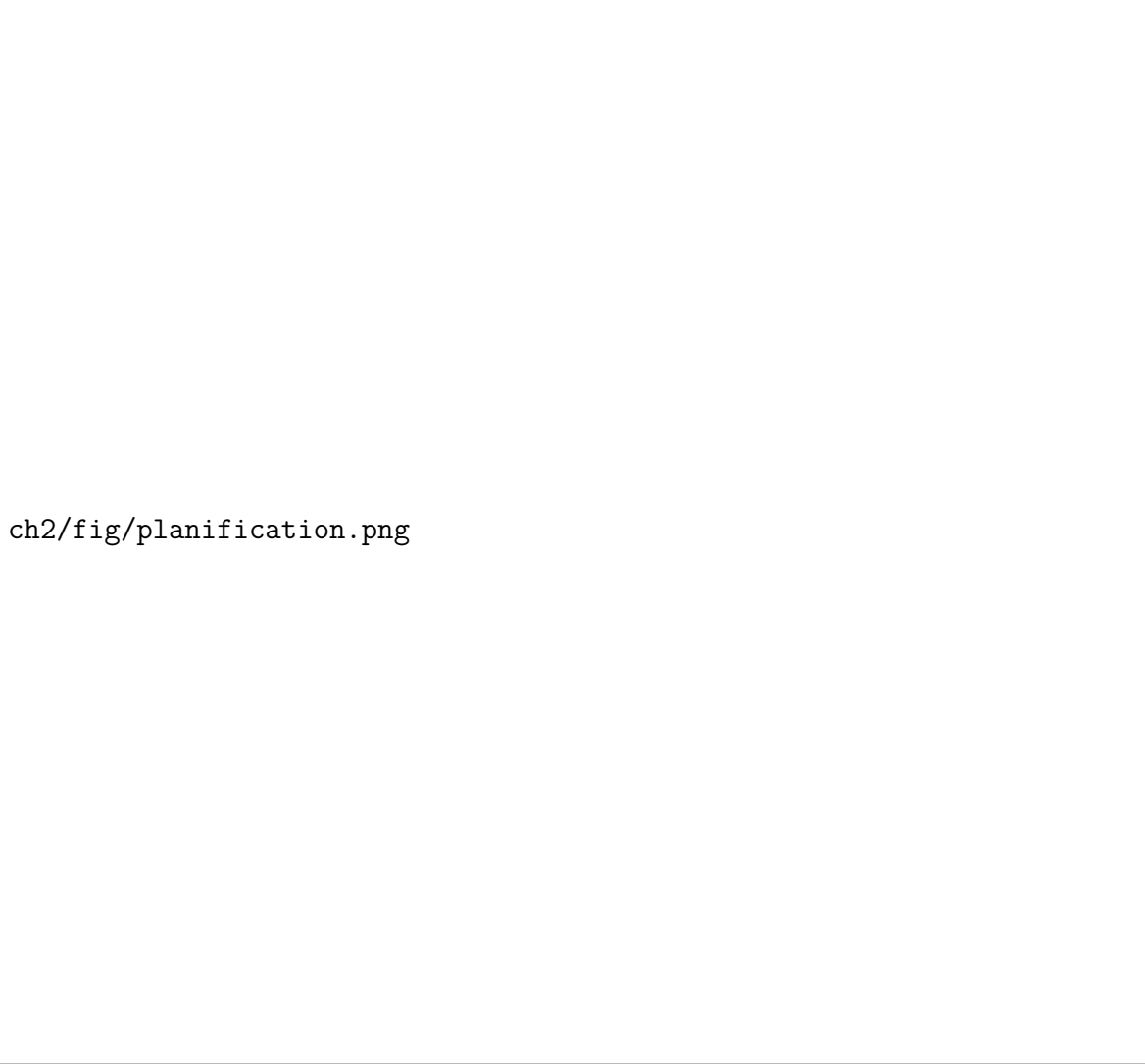
ch2/fig/classe.png

FIGURE 2.2 – Diagramme de classes général

### 3.5 Planification des sprints

Pendant la planification de la diffusion, un plan de haut niveau pour plusieurs sprints est élaboré, ce qui reflète les fonctionnalités attendues et leur échéance. Ce plan sert de guide pour suivre les progrès du projet et permet de livrer des résultats intermédiaires ou une livraison finale à la fin. Pour créer ce plan, il est essentiel d'avoir un backlog de produits Scrum hiérarchisé et estimé, ainsi que la vitesse de l'équipe Scrum et les objectifs de temps, portée et ressources. Dans notre projet, nous avons choisi de le découper en deux releases.

La figure 2.3 présente la planification des sprints.



ch2/fig/planification.png

FIGURE 2.3 – Planification des sprints

## 4 Environnement de travail

### 4.1 4.1 Environnement logiciel

#### ✓Visual Studio

Visual Studio, créé par Microsoft, est un environnement complet où vous pouvez écrire, tester et déboguer du code pour divers types de logiciels, comme les applications web, mobiles et de bureau. Il prend en charge de nombreux langages comme C#, JavaScript, Python, etc. Il permet aussi la gestion de versions, le déploiement, et l'intégration de services cloud.

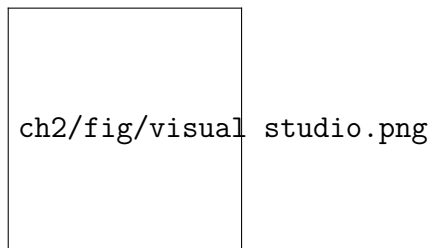


FIGURE 2.4 – Logo Visual Studio

#### ✓Postman

Postman est une plate-forme complète pour le test et la documentation des API. Elle facilite l'envoi de requêtes HTTP, la vérification des réponses et la collaboration autour des interfaces REST.

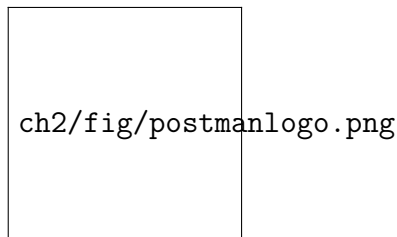
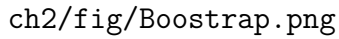


FIGURE 2.5 – Logo Postman

#### ✓Bootstrap

Bootstrap est un framework front-end permettant de créer rapidement des interfaces responsives. Il fournit des composants prédéfinis basés sur HTML, CSS et JavaScript.

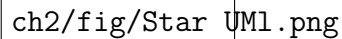


ch2/fig/Bootstrap.png

FIGURE 2.6 – Logo Bootstrap

### ✓StarUML

StarUML est un outil de modélisation UML utilisé pour concevoir visuellement des architectures logicielles à travers divers types de diagrammes UML.



ch2/fig/Star UML.png

FIGURE 2.7 – Logo StarUML

### ✓Overleaf

Overleaf est une plateforme en ligne de rédaction collaborative en LaTeX, utilisée pour produire des rapports, des présentations, et autres documents scientifiques ou techniques.



ch4/fig/overleaf.png

FIGURE 2.8 – Logo Overleaf

## 4.2 Environnement matériel

Le projet a été développé sur un environnement personnel dont les caractéristiques matérielles sont résumées dans la figure ci-dessous.



FIGURE 2.9 – Caractéristiques de la machine utilisée pour le développement

### 4.3 4.3 Technologies utilisées

Cette section décrit les différentes technologies employées dans le cadre du projet TalentCloud, réparties en langages de programmation, plateformes de travail et autres outils spécialisés.

#### Langages et Frameworks de programmation

**Java :** Utilisé côté backend pour les microservices, ce langage orienté objet garantit portabilité, performance et sécurité.



FIGURE 2.10 – Logo Java

**Spring Boot** : Framework basé sur Spring, permettant un développement rapide des services backend Java avec une configuration minimale.

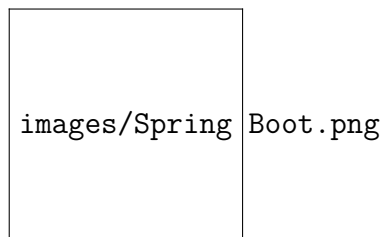


FIGURE 2.11 – Logo Spring Boot

**Kafka** : Plateforme de streaming distribuée pour la transmission d'événements et la mise en place de communications asynchrones.



FIGURE 2.12 – Logo Kafka

**React** : Bibliothèque JavaScript utilisée pour construire des interfaces utilisateur dynamiques en Single Page Application (SPA).



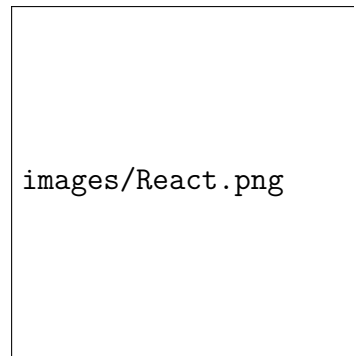


FIGURE 2.13 – Logo React

### Plateformes de développement utilisées

**GitHub** : Plateforme Git pour l'hébergement du code, la gestion des branches, des pull requests et la collaboration continue.

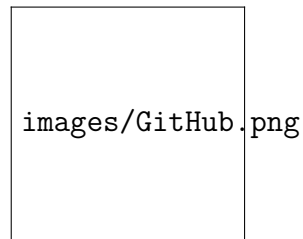


FIGURE 2.14 – Logo GitHub

**Jira** : Outil de gestion Agile utilisé pour suivre les tâches, sprints et releases du projet.

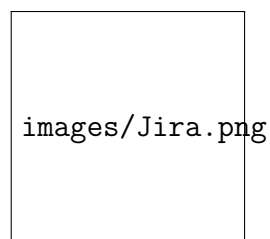


FIGURE 2.15 – Logo Jira

**IntelliJ IDEA** : IDE pour le développement Java, utilisé dans la création des microservices avec Spring Boot.

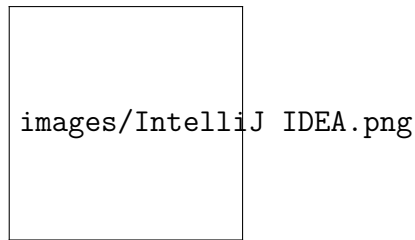


FIGURE 2.16 – Logo IntelliJ IDEA

**Visual Studio Code (VS Code) :** Éditeur de code utilisé pour le frontend React, avec intégration Git et extensions JavaScript/TypeScript.

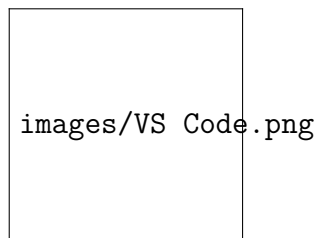


FIGURE 2.17 – Logo VS Code

**Postman :** Outil de test pour les APIs REST exposées par le backend.

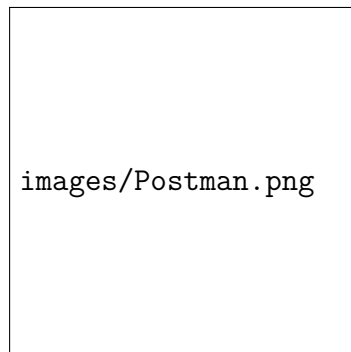


FIGURE 2.18 – Logo Postman

### Autres Technologies

**Keycloak :** Système d'authentification centralisé utilisé pour sécuriser l'accès et gérer les rôles des utilisateurs à travers OAuth2.



FIGURE 2.19 – Logo Keycloak

**PostgreSQL** : Base de données relationnelle utilisée pour la persistance des données métiers du projet.



FIGURE 2.20 – Logo PostgreSQL

## 5 Spécification architecturale

### 5.1 Architecture physique

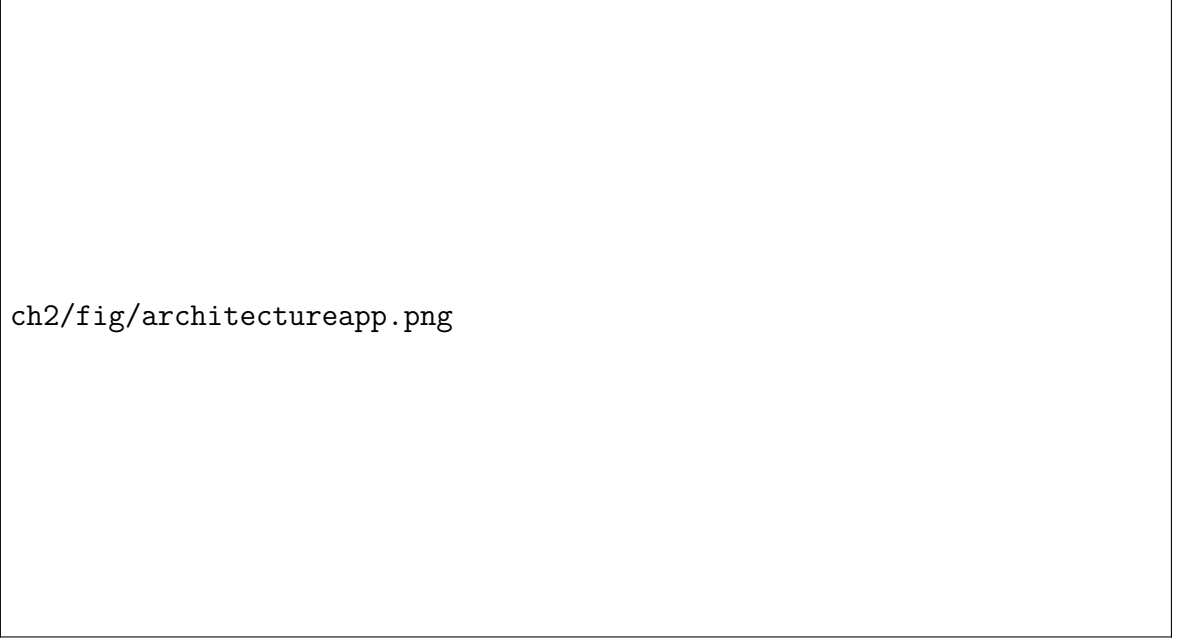
L'architecture physique de la plateforme TalentCloud repose sur une approche distribuée, conteneurisée et orientée microservices. L'ensemble des services est déployé dans des conteneurs Docker et orchestré via Docker Compose (ou tout autre orchestrateur tel que Kubernetes selon les besoins). Cette structure garantit une meilleure gestion des ressources, une évolutivité dynamique et une isolation des composants.

L'infrastructure est organisée comme suit :

- **Frontend monolithique** : L'interface utilisateur est développée sous forme de Single Page Application (SPA) en React et déployée dans un conteneur web.
- **API Gateway** : Sert de point d'entrée unique à toutes les requêtes client. Elle

redirige les appels vers les microservices concernés et gère des fonctionnalités transversales comme la sécurité ou le versionnage.

- **Microservices backend** : Chaque service métier (utilisateurs, offres, candidatures) est déployé dans son propre conteneur. Chaque service est indépendant, stateless et communique via HTTP/REST.
- **Service Discovery – Eureka** : Tous les microservices s’enregistrent auprès d’un registre centralisé (Eureka), permettant une découverte dynamique et automatique entre services.
- **Base de données PostgreSQL** : Stockage relationnel des données, accessible uniquement aux services autorisés via une couche ORM (type Entity Framework ou autre).



ch2/fig/architectureapp.png

FIGURE 2.21 – Architecture physique de l’application TalentCloud

## 5.2 Architecture logique

L’architecture logique illustre l’organisation fonctionnelle des composants logiciels. Le backend suit une architecture microservices, tandis que le frontend est construit sous forme d’application monolithique.

## Backend – Microservices

Chaque microservice traite un domaine fonctionnel bien distinct. Les services développés dans le cadre de ce projet sont :

- **Service Utilisateur** : gestion des comptes, rôles (admin, client, candidat), authentification.
- **Service Offres** : création, publication et consultation des offres d'emploi.
- **Service Candidature** : soumission et suivi des candidatures avec pièces jointes (CV, lettre de motivation).

Ces services sont exposés via des APIs REST et interconnectés via l'API Gateway. Le registre de service Eureka permet une localisation dynamique sans configuration statique.

## Frontend – Application monolithique

L'interface utilisateur est une application SPA (Single Page Application) développée en React avec typage fort (TypeScript). L'architecture orientée composant favorise la réutilisabilité, la modularité et la cohérence visuelle.

- Les composants sont regroupés par fonctionnalités (authentification, offres, candidatures).
- Les appels aux microservices se font via l'API Gateway.
- Le routage côté client évite les rechargements complets, améliorant l'expérience utilisateur.

## 5.3 Interaction entre les composants

L'interaction globale entre le frontend et les microservices backend est orchestrée par l'API Gateway. Toutes les requêtes du frontend transitent par cette passerelle, qui les redirige vers le service concerné.

Le registre de services Eureka permet aux services backend de découvrir dynamiquement leurs homologues. Cela élimine les dépendances rigides et renforce la résilience du système.

**Résumé du flux :**

1. Le frontend envoie une requête HTTP vers l'API Gateway.
2. La Gateway identifie la route et redirige la requête vers le microservice concerné.
3. Le microservice interagit avec la base de données si nécessaire, puis renvoie la réponse.
4. La Gateway renvoie la réponse au frontend.

## 6 Conclusion

Dans ce chapitre, nous avons présenté les besoins fonctionnels et non fonctionnels du projet TalentCloud, décrit l'environnement de développement ainsi que les différentes architectures adoptées, et détaillé les technologies utilisées. Ce socle technique et organisationnel permet d'assurer une base solide pour la conception, la mise en œuvre et l'évolution du système. Le chapitre suivant sera consacré à l'étude détaillée et à la réalisation du premier release du projet.

---

# Chapitre 3

## Etude et réalisation du release 1

### Sommaire

1	Introduction . . . . .	51
2	Développement du sprint 1 « Gestion des tests et validation des projets » . . . . .	51
2.1	Backlog du sprint 1 . . . . .	51
2.2	Spécification fonctionnelle . . . . .	52
2.3	Réalisation . . . . .	56
2.4	Rétrospective . . . . .	58
3	Développement du sprint 2 « Postulation pour un projet et Gestion des rendez-vous RH» . . . . .	58
3.1	Backlog du sprint 2 . . . . .	58
3.2	Spécification fonctionnelle . . . . .	59
3.3	Réalisation . . . . .	62
3.4	Rétrospective . . . . .	66
4	Conclusion . . . . .	66

## 1 Introduction

Dans ce chapitre, nous allons détailler le travail réalisé durant le premier release. En effet, chaque release, qui est l'ensemble d'itérations (sprint), représente une vision distribuée de la période de la production du livrable. Ce premier Release comprend deux sprints :

- Sprint 1 « Sécurité et gestion d'utilisateurs avec ses rôles »
- Sprint 2 « Gestion des départements et des projets »

Le développement de chaque sprint passe par les étapes d'analyse, de conception et de réalisation.

## 2 Développement du sprint 1 « Sécurité et gestion d'utilisateurs et leurs rôles »

Ce premier sprint s'étale sur 30 jours et se décompose en trois items :

- S'inscrire.
- S'authentifier.
- Gérer les utilisateurs ainsi que leurs rôles.

### 2.1 Backlog du sprint 1

Pour bien comprendre les fonctionnalités du premier sprint nous réservons cette partie pour présenter le backlog de ce dernier. Le Sprint Backlog est la liste des fonctionnalités sélectionnées pour ce sprint .

Le tableau [3.1](#) résume le Backlog du premier sprint.



Item	User Story	Description	Priorité
S'inscrire	S'inscrire	En tant qu'utilisateur, je peux créer un compte	1
S'authentifier	S'authentifier	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace	2
Gérer les utilisateurs et les rôles	Gérer les utilisateurs et les rôles	En tant qu'admin, je peux ajouter un ou plusieurs utilisateurs en leur attribuant un rôle (administrateur, gestionnaire, recruteur RH).	3

TABLE 3.1 – Backlog du sprint 1- Release 1

## 2.2 Spécification fonctionnelle

### 2.2.1 Diagramme de cas d'utilisation sprint 1

La figure 3.1 représente le diagramme de cas d'utilisation du premier sprint.

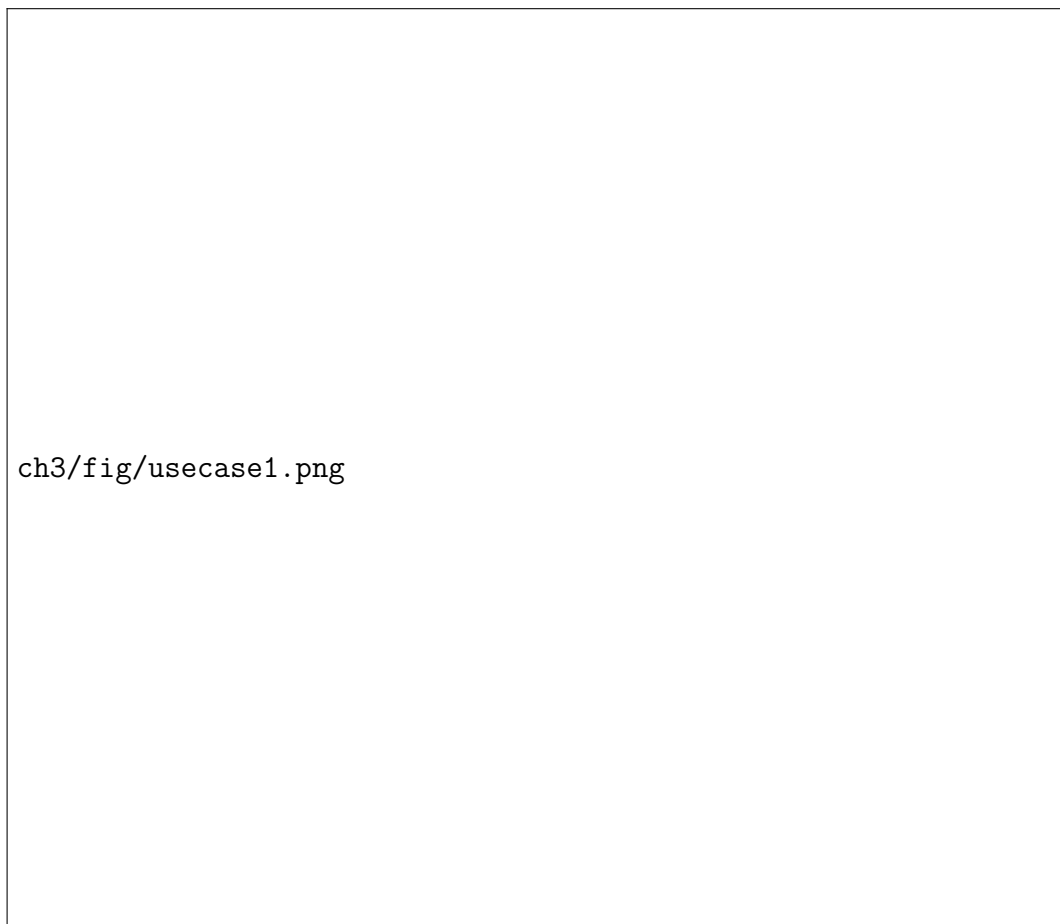


FIGURE 3.1 – Diagramme de cas d'utilisation sprint 1

### 2.2.2 Description textuelle des cas d'utilisation du sprint 1

La description textuelle des cas d'utilisation met en évidence les interactions entre les actions de l'utilisateur et le système. Nous avons choisi d'expliquer ces scénarios à travers un tableau qui montre clairement ce qui est fait par l'utilisateur et ce qui est fait par le système.

Le tableau 3.2 présente une description textuelle du cas d'utilisation « S'inscrire ».

Cas d'utilisation	S'inscrire
Acteur	User
Pré-condition	utilisateur non inscrit
Post Condition	Utilisateur inscrit
Scénario principal	1. L'utilisateur clique sur le bouton « S'inscrire ». 2. L'utilisateur remplit le formulaire d'inscription. 3. L'utilisateur clique sur le bouton « S'inscrire ». 4. Le système enregistre la demande.
Scénario alternatif	Le système affiche un message d'erreur « Veuillez remplir ce champ. ».

TABLE 3.2 – Description textuelle du cas d'utilisation « S'inscrire »

Le tableau 3.3 présente une description textuelle du cas d'utilisation «S'authentifier».

Cas d'utilisation	S'authentifier
Acteur	Utilisateur
Pré-condition	Utilisateur non authentifié
Post Condition	Utilisateur authentifié
Scénario principal	1. L'utilisateur clique sur le bouton « Connexion ». 2. L'utilisateur remplit le formulaire. 3. L'utilisateur clique sur le bouton « Se connecter ». 4. Le système enregistre la demande.
Scénario alternatif	Le système affiche un message d'erreur « Veuillez remplir ce champ. ».

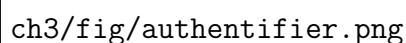
TABLE 3.3 – Description textuelle du cas d'utilisation « S'authentifier »

### 2.2.3 Diagramme de séquence

#### • Diagramme de séquence « S'authentifier »

Pour garantir un accès sécurisé aux ressources du système, un système d'authentification "JWT" a été mis en place, qui permet de stocker les données de session dans un jeton pour vérification et confiance.

La figure 3.2 montre comment un utilisateur peut se connecter à l'application afin d'utiliser ses propres fonctionnalités.



ch3/fig/authentifier.png

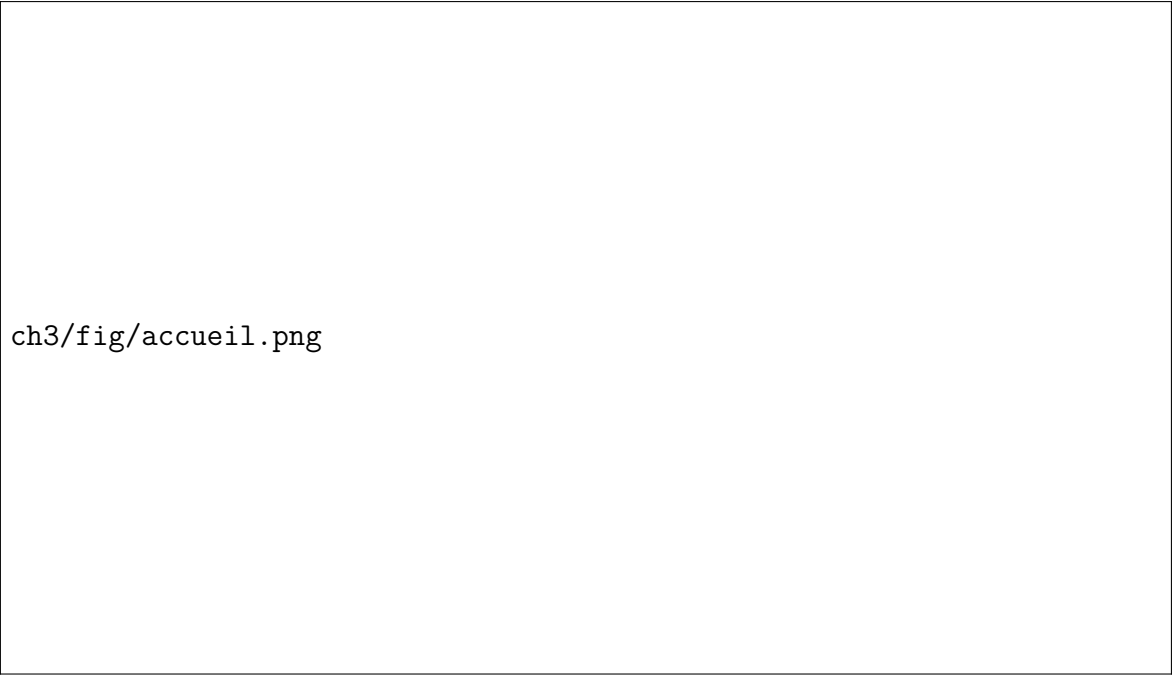
FIGURE 3.2 – Diagramme de séquence « S'authentifier »

## 2.3 Réalisation

Dans cette section, nous présentons les interfaces des fonctionnalités réalisées au cours de ce sprint.

### 2.3.1 Interface d'accueil

La figure 3.3 représente l'interface d'accueil de notre application avant de s'authentifier.




ch3/fig/accueil.png

FIGURE 3.3 – Interface d’accueil

### 2.3.2 Interface de Connexion

La figure 3.4 représente l’interface de connexion de notre application permettant aux utilisateurs de se connecter à leurs espaces respectifs.

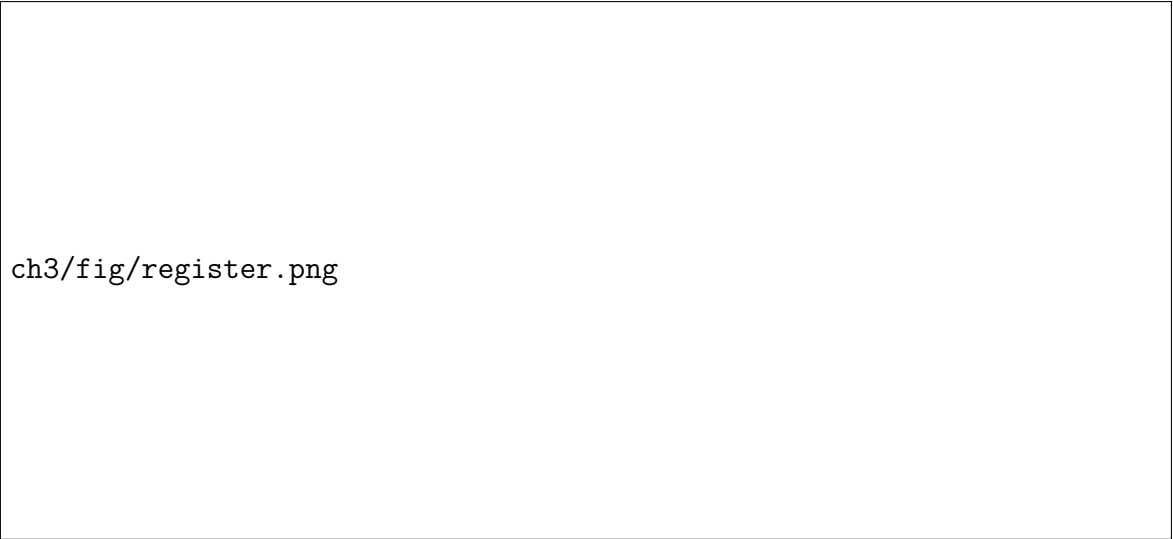


ch3/fig/connexion.png

FIGURE 3.4 – Interface de Connexion

### 2.3.3 Interface de création d’un compte et interface de gestion utilisateurs

La figure 3.5 représente l’interfaces de création d’un compte. Ainsi, la figure 3.6 représente l’inteface de gestion des utilisteurs qui est accessible sauf par l’administrateur.



ch3/fig/register.png

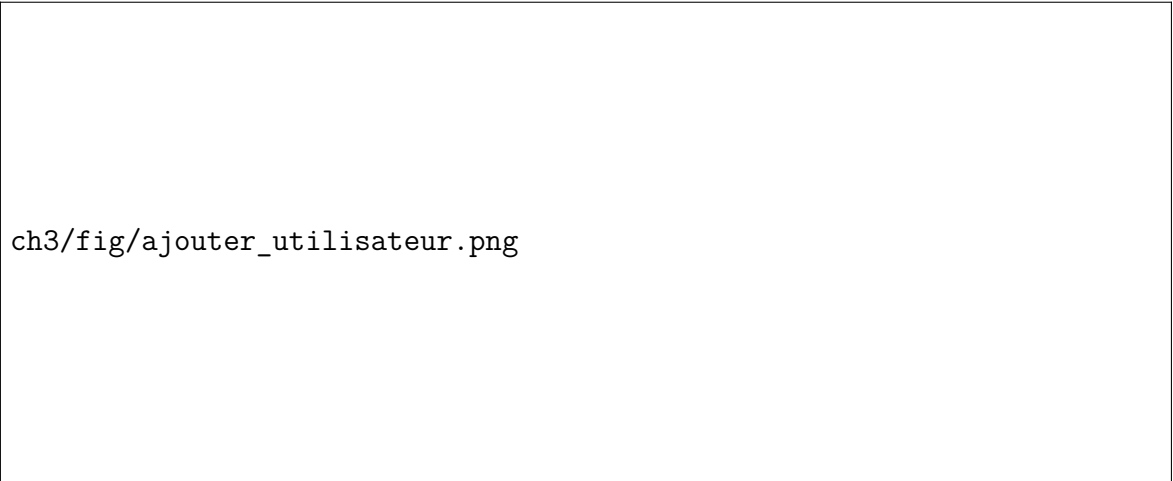
FIGURE 3.5 – Interface de création d'un compte pour un candidat

#### 2.3.4 Interface de création d'un compte et interface de gestion utilisateurs

La figure 3.6 représente l'interface d'ajout utilisateur en y effectant son rôle.

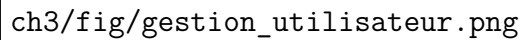
#### 2.3.5 Interface de gestion d'utilisateurs

La figure 3.7 représente l'interface de gestion d'utilisateurs.



ch3/fig/ajouter\_utilisateur.png

FIGURE 3.6 – Interface d'ajout utilisateur en y affectant un rôle



ch3/fig/gestion\_utilisateur.png

FIGURE 3.7 – Interface de création de gestion utilisateur

## 2.4 Rétrospective

Dans cette partie, nous allons citer les différentes rétrospectives dégagées par le Product Owner.

Ce qui a été validé :

- ✓ Authentification.
- ✓ Inscription.
- ✓ Gestion des utilisateurs ainsi que leurs rôles.

Ce qui n'a pas été validé : Rien.

Ce qui doit être amélioré : Rien.

## 3 Développement du sprint 2 « Gestion des départements et des projets »

Ce premier sprint s'étale sur 30 jours et se décompose en deux items :

- Gérer département.
- Gérer projet.

### 3.1 Backlog du sprint 2

Le tableau 3.4 résume le Backlog du deuxième sprint.

Item	User Story	Description	Priorité
Gérer département	Ajouter et consulter département	En tant que manager, je peux ajouter et consulter département	1
	Supprimer département	En tant que manager, je peux supprimer un département.	
	Modifier un département	En tant que manager, je peux modifier un département.	
Gérer Projet.	Gérer projet.	En tant que manager , je peux gérer yn projet et y affecté à un département.	2
	Consulter les projets publiés	En tant qu'utilisateur, je peux consulter les projets publiés sur la plateforme.	

TABLE 3.4 – Backlog du deuxième sprint

## 3.2 Spécification fonctionnelle

### 3.2.1 Diagramme de cas d'utilisation sprint 2

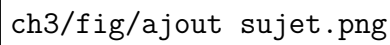
La figure 3.12 représente le diagramme de cas d'utilisation du deuxième sprint.

ch3/fig/diagramegestiondepartement.png

FIGURE 3.8 – Diagramme de cas d'utilisation sprint 2

### 3.2.2 Diagrammes de séquence

La figure 3.9 représente le diagramme de séquence d'ajout d'un projet.



ch3/fig/ajout\_sujet.png

FIGURE 3.9 – Diagramme de séquence d'ajout d'un projet





ch3/fig/DiagSequenceSupp.png

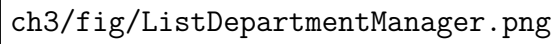
FIGURE 3.10 – Diagramme de séquence de suppression d'un projet

### 3.3 Réalisation

Dans cette section, nous présentons les interfaces des fonctionnalités réalisées au cours de ce sprint.

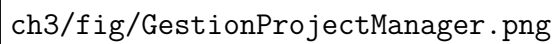
#### 3.3.1 Interface de gestion de départements et des projets

les figures suivantes représente les interfaces de gestion départements et projets



ch3/fig/ListDepartmentManager.png

FIGURE 3.11 – Interface de gestion département



ch3/fig/GestionProjectManager.png

FIGURE 3.12 – Interface de gestion projet

### 3.4 Rétrospective

Dans cette partie, nous allons citer les différentes rétrospectives dégagées par le Product Owner.

Ce qui a été validé :

- ✓ Gestion de départements.
- ✓ Gestion des projets.
- ✓ Affectation de projets à de départements.

Ce qui n'a pas été validé : Rien.

Ce qui doit être amélioré : Rien.

## 4 Conclusion

Ce chapitre a été consacré au premier release là où nous avons spécifié, modélisé, développé et testé l'ensemble des fonctionnalités des deux sprints 1 et 2 précédemment établies dans le Product Backlog. A ce stade, nous avons produit les deux sprints de la première release de notre projet livrables et exploitables. Le prochain chapitre sera dédié au deuxième release.

---

# Chapitre 4

Etude et réalisation du release 2

## 1 Introduction

Dans ce chapitre, nous allons détailler le travail réalisé durant le premier release. En effet, chaque release, qui est l'ensemble d'itérations (sprint), représente une vision distribuée de la période de la production du livrable. Ce premier Release comprend deux sprints :

- Sprint 1 « Gestion des tests et validation des projets »
- Sprint 2 « Postulation pour un projet et gestion des rendez-vous »

Le développement de chaque sprint passe par les étapes d'analyse, de conception et de réalisation.

## 2 Développement du sprint 1 « Gestion des tests et validation des projets »

### 2.1 Backlog du sprint 1

Pour bien comprendre les fonctionnalités du premier sprint nous réservons cette partie pour présenter le backlog de ce dernier. Le Sprint Backlog est la liste des fonctionnalités sélectionnées pour ce sprint Le tableau [4.1](#) résume le Backlog du premier sprint.


Item	User Story	Description	Priorité
Gérer les tests	ajouter un test	En tant qu'un manager, je peux ajouter un test.	1
	Modifier un test	En tant qu'un manager, je peux modifier un test.	
	Affecter un test à un projet	En tant qu'un manager, je peux affecter chaque test au projet cible.	
Valider les projets	Consulter les projects qui sont proposés de Manager	En tant qu'administrateur, je peux consulter les projets.	2
	Valider les projets	En tant qu'administrateur, je peux valider les projects ou les refuser.	
	Publier les projets sur la plateforme	En tant qu'administrateur, je peux publier les projets validés sur la plateforme.	

TABLE 4.1 – Backlog su sprint 1- Release 2

## 2.2 Spécification fonctionnelle

### 2.2.1 Diagramme de cas d'utilisation sprint 1

La figure 4.3 représente le diagramme de cas d'utilisation du premier sprint.

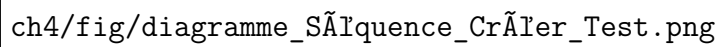


ch4/fig/usecase\_gestion\_test\_validation\_projet.png

FIGURE 4.1 – Diagramme de use case

### 2.2.2 Diagramme de séquence de créer un test

La figure 4.3 représente le diagramme de séquence du création de test.



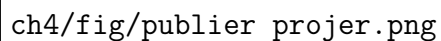
ch4/fig/diagramme\_S quence\_Cr ler\_Test.png

FIGURE 4.2 – Diagramme de s quence de cr er test

### 2.2.3 Diagramme de s quence de publier un projet

La figure 4.3 repr sente le diagramme de s quence du publier projet.





ch4/fig/publier\_projer.png

FIGURE 4.3 – Diagramme de séquence de publier projet

#### 2.2.4 Description textuelle des cas d'utilisation du sprint 1

La description textuelle des cas d'utilisation met en évidence les interactions entre les actions de l'utilisateur et le système. Nous avons choisi d'expliquer ces scénarios à travers un tableau qui montre clairement ce qui est fait par l'utilisateur et ce qui est fait par le système.

Le tableau suivant présente une description textuelle du cas d'utilisation « Gestion Test et validation projet ».

Le tableau 4.2 présente une description textuelle du cas d'utilisation «Ajout Test »

Cas d'utilisation	ajouter un test
Acteur	Manager
Pré-condition	Manager authentifié
Post Condition	Test n'est pas créé
Scénario principal	<ol style="list-style-type: none"> <li>1. Le manager accède à la« Liste des projets ».</li> <li>2.Choisir un projet.</li> <li>3. Affecter le test au projet.</li> <li>4. Le système affiche un message de confirmation.</li> <li>5. le manager appuie sur le bouton « Oui ».</li> <li>6. Le système recharge la page et met à jour la liste des tests et projets.</li> </ol>
Scénario alternatif	<ol style="list-style-type: none"> <li>1. Le manager appuie sur le bouton « Non ».</li> <li>2. Le système retourne à la liste des projets.</li> </ol>

TABLE 4.2 – Description textuelle du cas d'utilisation « Ajout Test»

Le tableau 4.3 présente une description textuelle du cas d'utilisation «Valider Projet ».

Cas d'utilisation	Valider un projet
Acteur	Admin
Pré-condition	Admin authentifié
Post Condition	la validation du projet n'ont pas encore été pris en compte.
Scénario principal	<ol style="list-style-type: none"> <li>1. L'admin accède à la« Liste des projets ».</li> <li>2.Choisir un projet.</li> <li>3. Consulter le détails du projet.</li> <li>4.le manager appuie sur le bouton de validation ou d'annulation .</li> <li>5.Le système affiche un message de confirmation.</li> <li>6. Le système recharge la page et met à jour la liste des projets.</li> </ol>
Scénario alternatif	<ol style="list-style-type: none"> <li>1. Le manager n' appuie sur aucun bouton .</li> <li>2.Le système renvoie un message pour inciter à la validation des projets.</li> </ol>

TABLE 4.3 – Description textuelle du cas d'utilisation « Ajout Test»

## 2.3 Réalisation

Dans cette section nous présentons les interfaces des fonctionnalités réalisées au cours de ce sprint.

### 2.3.1 Interface d'ajout test

La figure ci-dessous montre l'interface d'ajout de test et l'affectation à un projet spécifique .

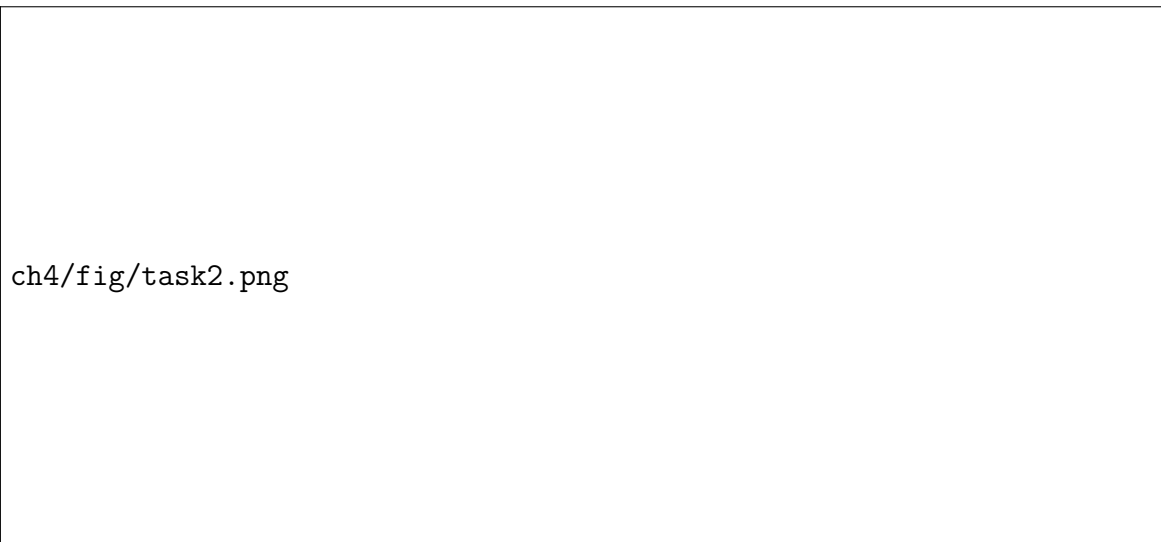


FIGURE 4.4 – Interface du test

### 2.3.2 Interface de la liste de projets

La figure ci-dessous montre la liste des projets qui doivent être valider avant la publication sur la plateforme. Seul l'adminisatrteur de notre plateforme a l'accès à cette interface.

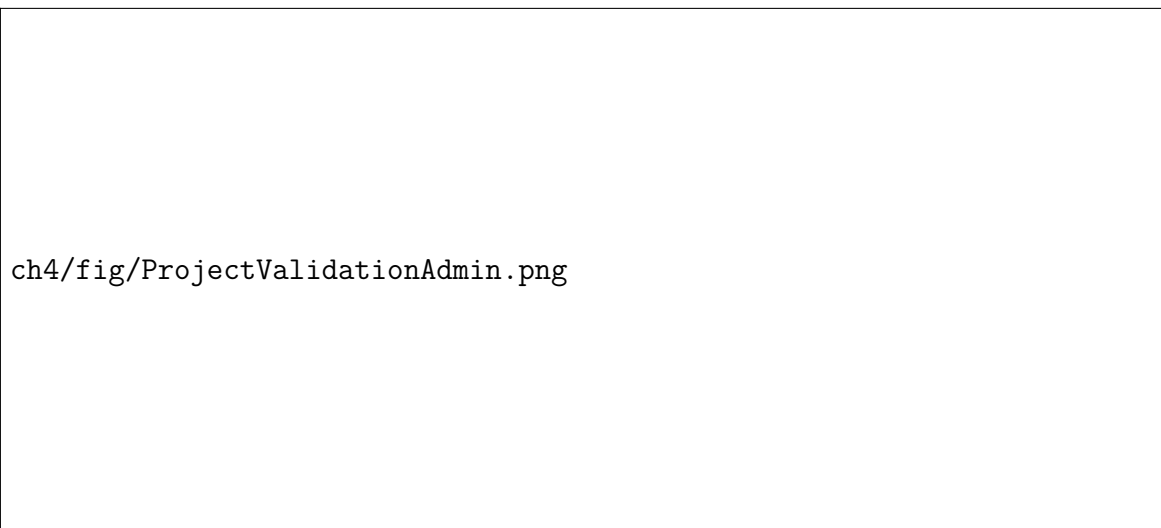


FIGURE 4.5 – Interface de la liste des projets validés et refusés

## 2.4 Rétrospective

Dans cette partie, nous allons citer les différentes rétrospectives dégagées par le Product Owner.

Ce qui a été validé :

- ✓ Gérer les tests.
- ✓ Valider les projets.
- ✓ publier les projets validés sur la plateforme.

Ce qui n'a pas été validé : Rien.

Ce qui doit être amélioré : Rien.

## 3 Développement du sprint 2 « Postulation pour un projet et Gestion des rendez-vous RH »

Ce premier sprint s'étale sur 30 jours et se décompose en trois items :

- Postuler pour un projet.
- Implementer les documents nécessaires.
- Passer le test.
- Corriger le test.
- Recevoir un résultat de test.
- Réserver un rendez-Vous.
- Gérer les rendez-Vous RH .

### 3.1 Backlog du sprint 2

Le tableau ci-dessous [4.4](#) résume le Backlog du deuxième sprint.

Item	User Story	Description	Priorité
Postuler pour un projet.	Postuler à un projet.	En tant qu'un candidat, je peux postuler pour un projet	1
Implémenter les documents.	Implémenter les documents	En tant qu'un candidat, je doit implémenter le Curriculum Vitae et la lettre de motivation .	2
Passer le test	Passer le test	En tant que candidat, je peux passer le test remis en plateforme .	
Corriger le test.	Corriger le test.	le système corrige en calculant le score obtenu.	3
Réserver un rendez-Vous RH.	Réserver un rendez-Vous RH.	En cas de réussite de test le candidat peut choisir une date pour un rendez-vous RH.	4

TABLE 4.4 – Backlog du sprint 2- Release 2

## 3.2 Spécification fonctionnelle

### 3.2.1 Diagramme de cas d'utilisation «Postuler pour un projet»

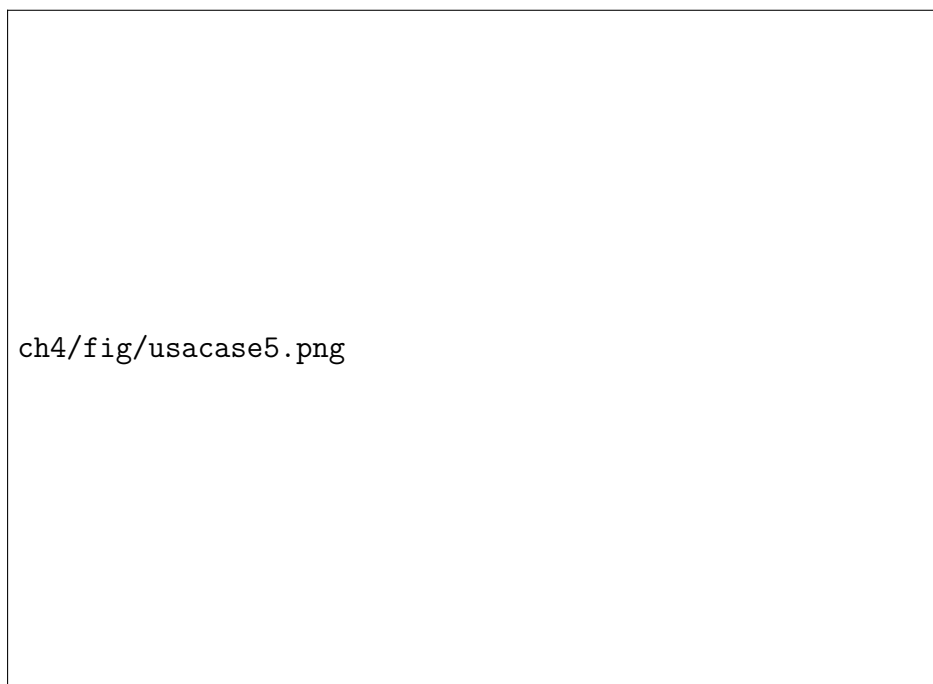


FIGURE 4.6 – Diagramme de cas d'utilisation « Postuler pour un projet »

### 3.2.2 Diagramme de séquence de «Postuler pour un projet»

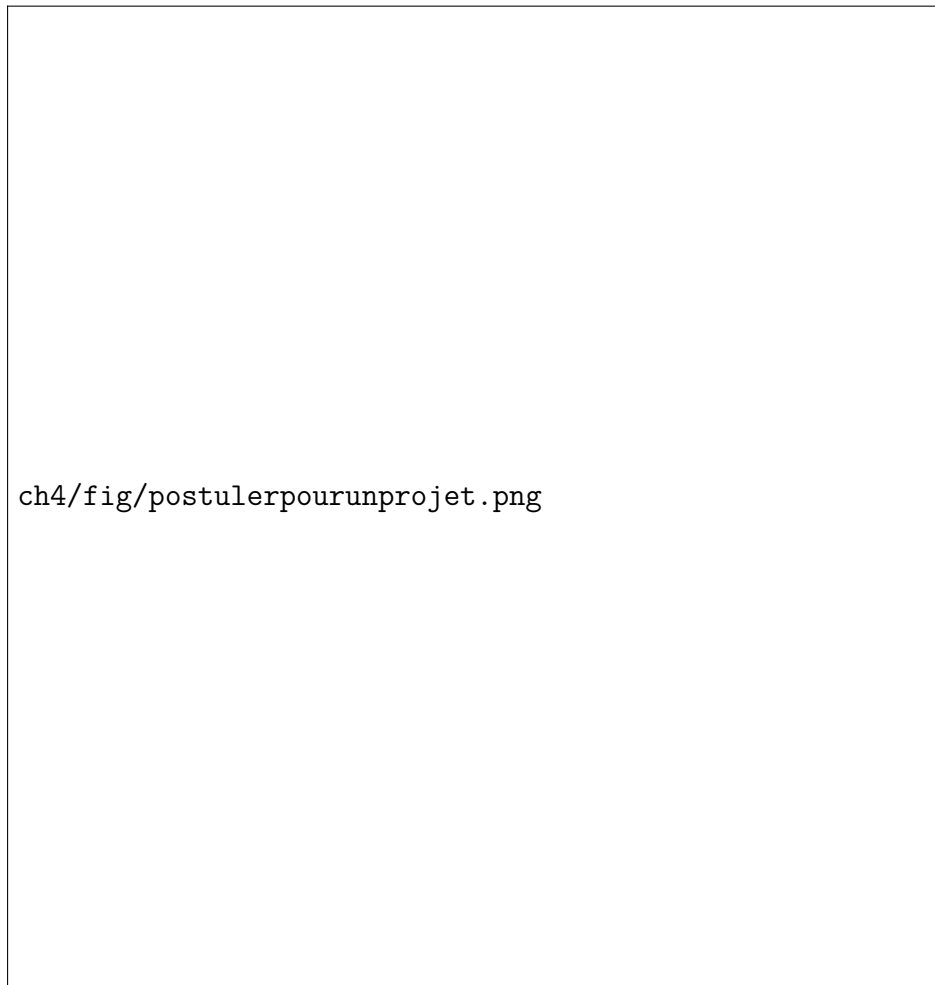


FIGURE 4.7 – Diagramme de séquence « Postuler pour un projet »

### 3.2.3 Diagramme de séquence de «corriger un projet»



FIGURE 4.8 – Diagramme de séquence « corriger un projet»

### 3.2.4 Description textuelle des cas d'utilisation du sprint 2

La description textuelle des cas d'utilisation met en évidence les interactions entre les actions de l'utilisateur et le système. Nous avons choisi d'expliquer ces scénarios à travers un tableau qui montre clairement ce qui est fait par l'utilisateur et ce qui est fait par le système.

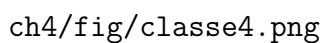
Le tableau 4.5 présente une description textuelle du cas d'utilisation « Choisir un rendez-Vous RH ».

Cas d'utilisation	Choisir un rendez-Vous RH
Acteur	Candidat
Pré-condition	Candidat authentifié
Post Condition	Date est réservée
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le bouton « choisir un date pour un entretien RH ».</li> <li>2. L'utilisateur choisit la Bouton "Postuler" .</li> <li>3. L'utilisateur écrit son message.</li> <li>4. la date sera indisponible dans la liste des rendez-Vous RH.</li> <li>5. le candidat sera enregistré dans la listes de rendez-vous Rh de dashbord HR Recruiter</li> </ol>
Scénario alternatif	Date est indisponible.

TABLE 4.5 – Description textuelle du cas d'utilisation « Réserver une date pour un rendez-Vous RH»

### 3.2.5 Diagramme de classes sprint 2

La figure 4.9 présente le diagramme de classes du deuxième sprint.

The image is a large, empty rectangular box with a thin black border. Inside the box, on the left side, is the text 'ch4/fig/classe4.png' in a monospaced font. This text likely serves as a placeholder for a UML class diagram that is not visible in this version of the document.

ch4/fig/classe4.png

FIGURE 4.9 – Diagramme de classes sprint 2


## 3.3 Réalisation

Dans cette section nous présentons les interfaces des fonctionnalités réalisées au cours de ce sprint.

### 3.3.1 Interface de postulation

La figure ci-dessous 4.10 représente l'interface de postulation pour un projet PFE  
\*.



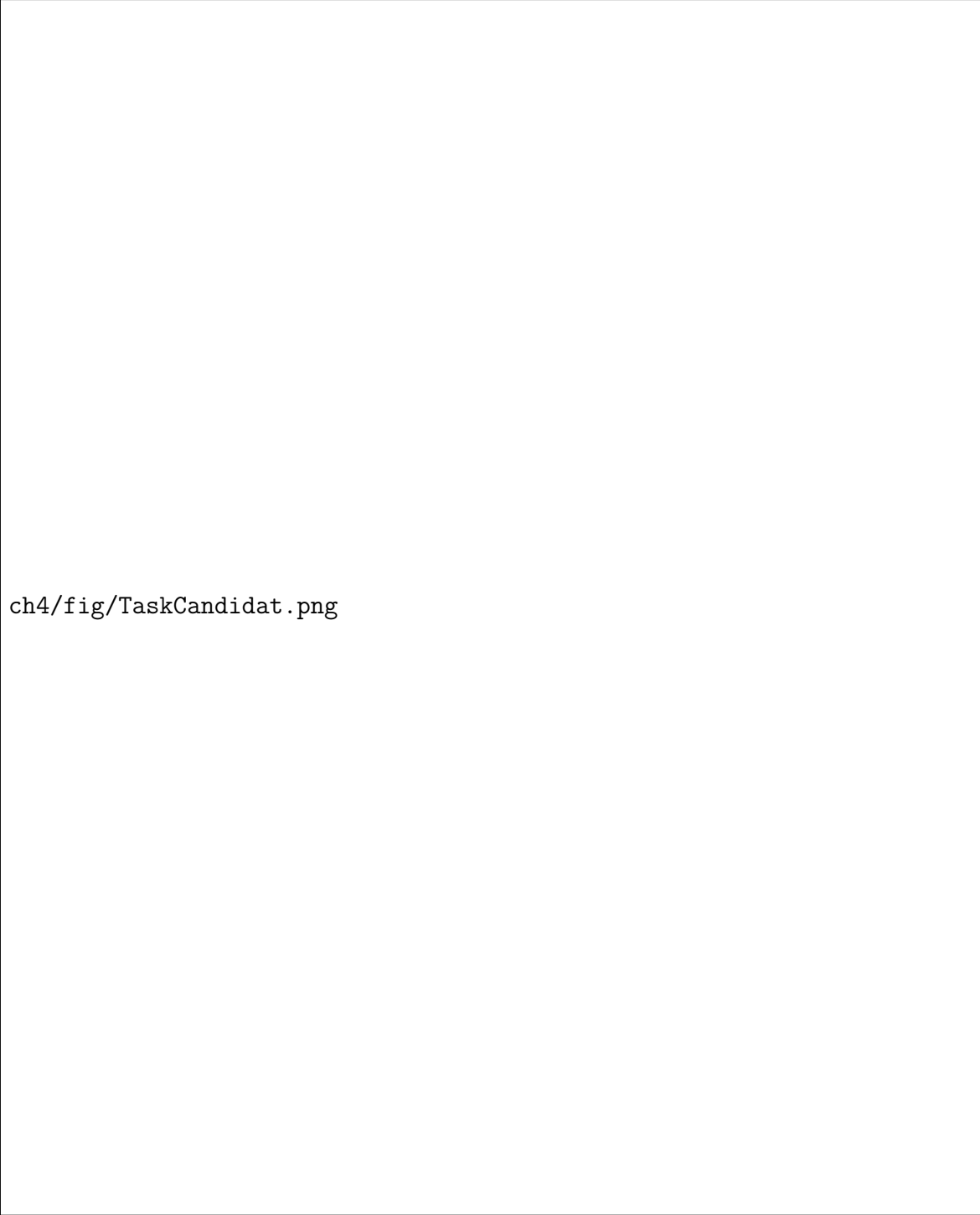


ch4/fig/Candidate file + diag.png

FIGURE 4.10 – Interface de postulation

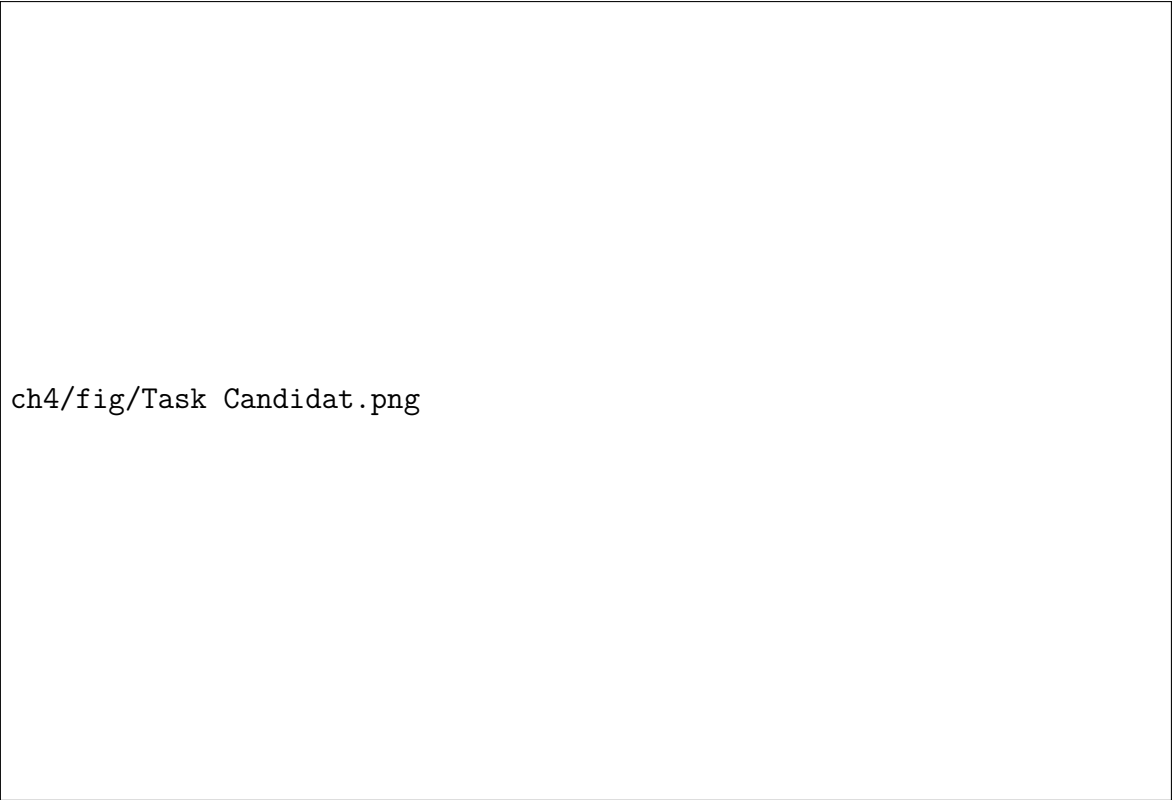
### 3.3.2 Interface du test

La figure 4.11 représente l'interface de test et la figure 4.12 représente le résultat du test.



ch4/fig/TaskCandidat.png

FIGURE 4.11 – Interface contenant le test

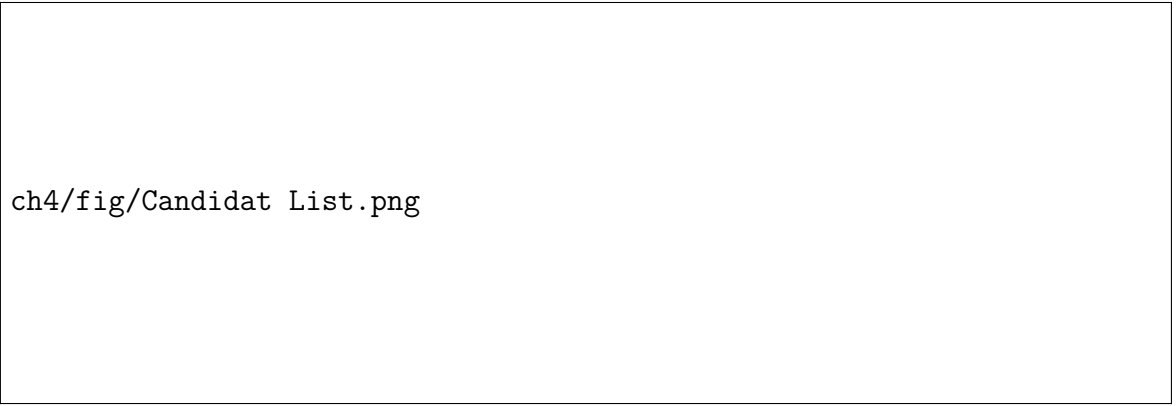


ch4/fig/Task Candidat.png

FIGURE 4.12 – Resultat du test

### 3.3.3 Interface de la liste de résultats des candidats

La figure ci-dessous 4.14 représente l'interface de la liste de résultats des candidats.

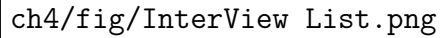


ch4/fig/Candidat List.png

FIGURE 4.13 – Interface de la liste de résultats des candidats.

### 3.3.4 Interface de la liste de rendez-Vous RH

La figure ci-dessous 4.14 représente Interface de la liste de rendez-Vous RH.

The image is a placeholder for a screenshot of the RH appointment list interface. The text 'ch4/fig/InterView List.png' is displayed within a rectangular frame, indicating the location of the original figure file.

ch4/fig/InterView List.png

FIGURE 4.14 – Interface de la liste de rendez-Vous RH.

### 3.4 Rétrospective

Dans cette partie, nous allons citer les différentes rétrospectives dégagées par le Product Owner.

Ce qui a été validé :

- ✓Postuler pour un test.
- ✓Passer un test.
- ✓Choisir un rendez-vous Rh.
- ✓Gérer les rendez-vous RH

Ce qui n'a pas été validé : Rien.

Ce qui doit être amélioré : Rien.

## 4 Conclusion

Dans ce chapitre, nous avons réussi à exhiber le 2ème release de notre projet. Nous avons à cet effet accompli la spécification, la modélisation, le développement de la deuxième release comme précisées dans le Product Backlog.

---

# Conclusion générale

En conclusion, notre projet de développement d'une plateforme dédiée à la recherche de stages représente une réponse significative au défi du chômage et à la nécessité croissante de connecter efficacement les demandeurs de stages aux opportunités disponibles. En partenariat avec l'entreprise "Innov-Alliance-Tech (I@T)", nous avons conçu une solution innovante qui tire parti des avancées technologiques et de la digitalisation pour simplifier et optimiser le processus de recrutement de stagiaires.

Ce rapport met en lumière les différentes phases de développement de la plateforme, de la conceptualisation initiale à la mise en œuvre technique, en passant par l'analyse approfondie des besoins des utilisateurs. Chaque étape a été abordée avec rigueur et souci du détail, dans le but de créer une solution robuste et efficace.

Alors que nous clôturons cette étape du projet, il est important de souligner que notre plateforme n'est pas une fin en soi, mais plutôt le début d'un voyage vers l'amélioration continue. Nous sommes conscients que le paysage du recrutement évolue constamment, et nous restons engagés à faire évoluer notre solution pour répondre aux besoins changeants du marché.

Enfin, nous tenons à exprimer notre gratitude envers toutes les parties prenantes qui ont contribué à la réalisation de ce projet. Leur collaboration et leur soutien ont été essentiels à chaque étape du processus, et nous sommes impatients de poursuivre cette collaboration fructueuse dans le futur.

---

# Webographie

- [1] <http://ia-tech.net/> [Accès le 05/02/2024].
- [2] <https://www.emploiunisie.com/> [Accès le 28/02/2024].
- [3] <https://www.tanitjobs.com/> [Accès le 28/02/2024].
- [4] <https://www.keejob.com/> [Accès le 28/02/2024].
- [5] <https://dotnet.microsoft.com/apps/aspnet>. [Accès le 02/03/2024].
- [6] <https://visualstudio.microsoft.com/docs> [Accès le 05/03/2024].
- [7] <https://www.postman.com/> [Accès le 05/03/2024].
- [8] <https://fr.overleaf.com/learn> [Accès le 08/03/2024].