



Ministère de l'Enseignement
Supérieur
et de la Recherche Scientifique



Rapport de Fin d'Etudes

Préparé en vue de l'obtention du diplôme National d'Ingénieur
en Informatique

Option : Génie Logiciel et Systèmes d'Information

Conception et Mise en Place d'une Plateforme pour la Gestion des Experts

Réalisé par : Sofiene Khannoussi

Organisme d'accueil :
Medianet Consulting



Encadrant pédagogique : M. Hassen LAZRAG

Encadrant professionnel : M. Ali MAATIG

Année Universitaire 2023 - 2024

Table des matières

Introduction générale	1
1 Analyse et Spécifications des besoins	2
1.1 Introduction	2
1.2 Analyse du système actuel	2
1.3 Etude de l'existant	3
1.3.1 Tanitjobs.com	3
1.3.2 Malt.fr	4
1.3.3 Expression du besoin	6
1.4 Planification et conduite du projet	6
1.4.1 Présentation de la méthode agile	6
1.4.2 Présentation de la methode Scrum	7
1.4.3 Processus de développement	10
1.5 Méthodologies Adoptées	12
1.6 Spécification des besoins	13
1.6.1 Identification des acteurs	13
1.6.2 Analyse des besoins globaux	14
1.7 Planification des Sprints	15
1.8 Architecture de l'application	16
1.8.1 Architecture Logique	17
1.8.2 Architecture Physique	19
1.9 Technologies et outils utilisés	20
1.9.1 Environnement matériel	20

1.9.2	Environnement Logiciel	21
1.10	Conclusion	26
2	Conception	27
2.1	Introduction	27
2.2	Diagramme de Classes Global	27
2.3	Diagramme de Cas d'Utilisation	30
2.3.1	Diagramme de cas d'utilisation général	31
2.3.2	Diagramme de cas d'utilisation pour Expert	32
2.3.3	Diagramme de cas d'utilisation pour Manager	40
2.3.4	Diagramme de cas d'utilisation pour Admin	44
2.4	Diagramme de Séquence	48
2.4.1	Diagramme de Séquence : Visualiser mon profil (Gestion de données)	48
2.4.2	Diagramme de Séquence : Modifier mes informations personnelles (Gestion expert)	49
2.4.3	Diagramme de Séquence : Ajouter une compétence (Gestion Com- pétence)	50
2.4.4	Diagramme de Séquence : Modifier une compétence (Gestion Com- pétence)	51
2.4.5	Diagramme de Séquence : Désactiver une compétence (Gestion Com- pétence)	52
2.4.6	Diagramme de Séquence : Ajouter une Mission (Gestion Mission) .	53
2.4.7	Diagramme de Séquence : Modifier une Mission (Gestion Mission) .	54
2.4.8	Diagramme de Séquence : Visualiser les Contrats Validés Pendant une Période (Admin)	55
2.4.9	Diagramme de Séquence : Visualiser les Contrats d'une Société (Ad- min)	56
2.4.10	Diagramme de Séquence : Visualiser les Contrats d'un Expert (Admin)	57
2.5	Diagramme d'Activités	57
2.5.1	Diagramme d'Activités : Visualiser Mon Profil (Gestion Expert) . .	58
2.5.2	Diagramme d'Activités : Modifier Mes Informations Personnelles (Ges- tion Expert)	58

2.5.3	Diagramme d'Activités : Ajouter Une Compétence (Gestion Compétence)	60
2.5.4	Diagramme d'Activités : Modifier Une Compétence (Gestion Compétence)	61
2.6	Conclusion	62
3	Réalisation	63
3.1	Introduction	63
3.2	Création du logo	63
3.3	Release N°1 : Gestion des Comptes Admin, Manager, Expert , et Inscription	64
3.3.1	Backlog du sprint 1	64
3.3.2	Réalisation	64
3.4	Release N°2 : Gestion de compétences et gestion de CV	68
3.4.1	Backlog du sprint 2	68
3.4.2	Réalisation	69
3.5	Release N°3 : Gestion des missions	71
3.5.1	Backlog du sprint 3	71
3.5.2	Réalisation	72
3.6	Release N°4 : Gestion des contrats	75
3.6.1	Backlog du sprint 4	75
3.6.2	Réalisation	76
3.7	Release N°5 : Gestion des tableaux de bord et messagerie interne	77
3.7.1	Backlog du sprint 5	77
3.7.2	Réalisation	78
3.8	Environnement de Développement	78
3.9	Structure du Projet	79
3.9.1	Organisation du Code Source	79
3.9.2	Détails des Modules	79
3.10	Déploiement	79
3.10.1	Préparation du Déploiement	80
3.10.2	Déploiement sur Serveur	80
3.10.3	Étapes de Déploiement	80

3.10.4 Scripts d'Automatisation	81
3.10.5 Surveillance et Maintenance Post-Déploiement	81
3.11 Dockerisation du Projet	81
3.11.1 Avantages de la Dockerisation	82
3.11.2 Création des Fichiers Docker	82
3.11.3 Gestion des Réseaux et Volumes	82
3.12 Déploiement des Conteneurs	83
3.12.1 Automatisation du Déploiement	83
3.13 Conclusion	83
Conclusion générale & perspectives	84
Références	85

Table des figures

1.1	TanitJobs	3
1.2	Malt.fr	5
1.3	La méthodologie agile	7
1.4	La méthode scrum	11
1.5	La méthode scrum	13
1.6	Planning du projet	16
1.7	Architecture du flux de démarrage Spring	18
1.8	Les couches de Spring Boot	18
1.9	Architecture logique de l'application.	19
1.10	Architecture physique du déroulement de l'application	20
1.11	Logo Css	21
1.12	Logo Javascript	22
1.13	Logo Json	22
1.14	Logo JQuery	22
1.15	Logo Spring Boot	23
1.16	Logo Angular	23
1.17	Logo Draw.io	24
1.18	Logo Visuel Studio code	24
1.19	Logo Overleaf	24
1.20	Logo MySQL	25
1.21	Logo GitHub	25
1.22	Logo StarUML	25
1.23	Logo Tesseract	26

2.1	Diagramme de classe global	30
2.2	Diagramme de cas d'utilisation général	31
2.3	Diagramme de cas d'utilisation pour Expert	39
2.4	Diagramme de cas d'utilisation pour Manager	43
2.5	Diagramme de cas d'utilisation pour Admin	47
2.6	Diagramme de Séquence : Visualiser mon profil (Gestion de données	48
2.7	Diagramme de Séquence : Modifier mes informations personnelles (Gestion expert)	49
2.8	Diagramme de Séquence : Ajouter une compétence (Gestion Compétence) .	50
2.9	Diagramme de Séquence : Modifier une compétence (Gestion Compétence)	51
2.10	Diagramme de Séquence : Désactiver une compétence (Gestion Compétence)	52
2.11	Diagramme de Séquence : Ajouter une Mission (Gestion Mission)	53
2.12	Diagramme de Séquence : Modifier une Mission (Gestion Mission)	54
2.13	Diagramme de Séquence : Visualiser les Contrats Validés Pendant une Période (Admin)	55
2.14	Diagramme de Séquence : Visualiser les Contrats d'une Société (Admin) . .	56
2.15	Diagramme de Séquence : Visualiser les Contrats d'un Expert (Admin) . .	57
2.16	Diagramme d'Activités : Visualiser Mon Profil (Gestion Expert)	58
2.17	Diagramme d'Activités : Modifier Mes Informations Personnelles (Gestion Expert)	59
2.18	Diagramme d'Activités : Ajouter Une Compétence (Gestion Compétence) .	60
2.19	Diagramme d'Activités : Modifier Une Compétence (Gestion Compétence)	61
3.1	Logo Expert Consulting	63
3.2	Qui Je suis	65
3.3	Inscription expert	65
3.4	Inscription societe	66
3.5	Connexion	67
3.6	Données personnelles	67
3.7	Profil expert	69
3.8	Profil expert -Ajout des Compétences	69
3.9	Profil expert - Ajout des Diplomes	70

3.10 Profil expert - Ajout des missions réalisées	70
3.11 Création d'une mission	72
3.12 Modification d'une mission	72
3.13 Postuler une candidature	73
3.14 Liste des candidatures	74
3.15 Ajouter un contrat	76
3.16 Dashboard	78

Liste des tableaux

1.1	Scrum vs. Waterfall	8
1.2	Scrum vs. Kanban	9
1.3	Scrum vs. Extreme Programming (XP)	9
1.4	Répartition des releases	15
1.5	Parties du code	17
1.6	Environnement matériel.	21
3.1	Backlog du sprint 1	64
3.2	Backlog du sprint 2	68
3.3	Backlog du sprint 3	71
3.4	Backlog du sprint 4	75
3.5	Backlog du sprint 5	77

Liste des abréviations

API	<i>Application Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
REST	<i>Representational State Transfer</i>
MVC	<i>Model View Controller</i>
CSS	<i>Cascading Style Sheets</i>
JWT	<i>Json Web Token</i>
SQL	<i>Structured Query Language</i>
OCR	<i>Reconnaissance Optique de Caractères</i>
NLP	<i>Natural Language Processing</i>
UML	<i>Unified Modeling Language</i>

Introduction générale

La transformation numérique a radicalement changé la manière dont les organisations gèrent leurs ressources humaines, en particulier dans les environnements multi-sociétés. Ce rapport de Projet de Fin d'Études (PFE) en ingénierie de génie logiciel se concentre sur le développement d'une solution numérique innovante au sein de Medianet Consulting Tunisie.

Medianet Consulting Tunisie, spécialiste en formation informatique, conseil en innovation et intégration de solutions informatiques, accompagne les entreprises dans le développement des compétences informatiques de leurs équipes et la mise en œuvre de solutions technologiques avancées et rentables. Dans ce contexte, la digitalisation est devenue un levier stratégique crucial pour optimiser les processus internes, en particulier ceux liés à la gestion des experts.

Notre projet vise à concevoir et à développer une application web destinée à simplifier et améliorer le processus de recherche et de gestion des experts au sein de Medianet Consulting. L'objectif est de créer une plateforme centralisée qui automatise les étapes de recrutement, de validation et d'intégration des experts, afin d'optimiser l'efficacité et la transparence du processus.

Ce rapport est structuré en plusieurs chapitres :

- Un premier chapitre intitulé analyse et spécification des besoins dans lequel il y aura une étude de l'existant et expression des besoins relatifs au projet.
- Le deuxième chapitre détaille les besoins et attentes en termes de fonctionnalités et de performance, en préparation du Backlog et de l'architecture de l'application.
- Le troisième chapitre sera dédié à la réalisation où nous décrivons en premier lieu, les technologies et les outils utilisés et en second lieu, la structure du projet et les interfaces du système.

En conclusion, ce rapport met en lumière les étapes cruciales du développement d'une solution numérique destinée à transformer la gestion des experts au sein de Medianet Consulting Tunisie. La mise en œuvre de cette solution vise à améliorer la fluidité des processus internes et à renforcer la capacité de l'organisation à répondre aux besoins de ses clients de manière plus efficace et transparente.

Chapitre 1

Analyse et Spécifications des besoins

1.1 Introduction

Ce chapitre présente l'étude du processus actuel adopté par Medianet Consulting. En premier lieu, une analyse sera faite pour faire un état de lieux et cerner exhaustivement les objectifs à atteindre. En second lieu, une définition des besoins fonctionnels et non fonctionnels ainsi que les différents acteurs du système sera élaboré.

1.2 Analyse du système actuel

- Avant la digitalisation, chez Medianet ¹ Consulting, la gestion des candidats se faisait entièrement sur papier. Les CV et lettres de motivation étaient reçus par courrier ou en personne, puis triés et rangés dans des classeurs physiques.
- Chaque dossier candidat était géré manuellement, avec des notes d'entretien et autres documents ajoutés au fur et à mesure. Ce processus demandait beaucoup de temps et rendait la recherche de candidats potentiels difficile. De plus, il était facile de perdre des documents ou de mélanger des informations, ce qui compliquait le suivi des candidatures et la coordination entre les membres de l'équipe de recrutement.
- Avant En résumé, cette méthode manuelle était lente, peu fiable, et nécessitait une attention constante pour éviter les erreurs.

1. <https://www.medianet.tn/>

1.3 Etude de l'existant

La phase d'étude de l'existence est une étape incontournable dans la réalisation de tout projet. Elle implique la compréhension et l'analyse des solutions déjà existantes ainsi que des défis potentiels liés au projet. Cette étape permet de construire un projet répondant aux besoins spécifiques de l'établissement, tout en ayant une vue d'ensemble sur les avantages et les inconvénients des projets existants.

En effectuant une analyse détaillée des projets existants, on peut déterminer les points forts et les points faibles de chaque solution, et ainsi s'appuyer sur ces informations pour concevoir un projet optimal.

Ainsi, en utilisant ces informations, il est possible de déterminer les besoins du projet et de les prendre en compte lors de la conception et de la réalisation. Pour cela, chaque projet est découpé en deux parties distinctes : la description de l'existant et la critique de l'existant.

1.3.1 Tanitjobs.com

Description de l'existant :

Tanit Jobs est une plateforme de recherche d'emploi qui permet aux demandeurs d'emploi de consulter et de postuler à des offres d'emploi publiées par des employeurs. Cette plateforme couvre un large éventail de secteurs et de métiers, offrant ainsi des opportunités d'emploi pour tous les types de profil.

Le portail est destiné aussi bien aux employeurs qu'aux chercheurs d'emploi en Tunisie, mais également à l'international. Les employeurs peuvent ainsi publier des offres d'emploi pour des postes à pourvoir dans leur entreprise, tandis que les demandeurs d'emploi peuvent accéder à ces offres et postuler en ligne, en fonction de leur profil et de leurs compétences.



FIGURE 1.1 – TanitJobs

Critique de l'existant :

Dans cette partie, nous allons énumérer les différents points forts et les limites de Tanit Jobs.

Les points forts

Tanitjobs.com offre plusieurs avantages pour les chercheurs d'emploi et les employeurs :

- Accès à un large éventail d'offres d'emploi.
- Facilité de recherche d'emploi : Tanitjobs.com permet aux chercheurs d'emploi de rechercher des offres d'emploi en fonction de différents critères.
- Processus de recrutement en ligne : Les employeurs peuvent publier des offres d'emploi et recevoir les candidatures en ligne.
- Confidentialité : Tanitjobs.com offre aux employeurs la possibilité de publier des offres d'emploi en toute confidentialité.
- Accessibilité : Tanitjobs.com est accessible en ligne à tout moment.

Limites de Tanit Jobs

- Tanit Jobs joue le rôle d'intermédiaire entre les chercheurs d'emploi et les employeurs. De nombreuses entreprises ont la possibilité de publier des offres d'emploi sur ce site.
- Offres d'emploi limitées : Bien que Tanit Jobs soit un site de recherche d'emploi populaire, il ne propose pas toutes les offres d'emploi disponibles sur le marché.
- Pas d'accompagnement personnalisé : Tanit Jobs est un site de recherche d'emploi en ligne automatisé. Il n'offre pas d'accompagnement personnalisé pour les candidats. Les utilisateurs doivent donc naviguer seuls sur le site, trouver des offres d'emploi, postuler et gérer leur candidature.

1.3.2 Malt.fr

Description de l'existant :

Malt est une plateforme en ligne qui met en relation des freelances et des entreprises. Fondée en 2013, Malt a pour mission de faciliter l'accès aux talents freelances et de simplifier la gestion des missions pour les entreprises. La plateforme se distingue par son approche orientée vers la qualité, la transparence et la confiance, créant un environnement favorable pour les freelances et les recruteurs.

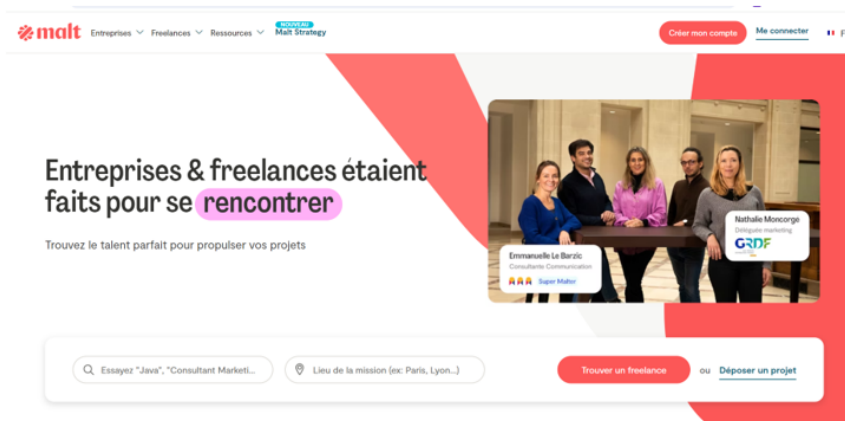


FIGURE 1.2 – Malt.fr

Critique de l'existant : Dans cette partie, nous allons énumérer les différents points forts et les limites de Malt.

Les points forts : Malt offre plusieurs avantages pour les freelances et les employeurs :

- Une grande variété de missions disponibles, permettant aux freelances de trouver des projets adaptés à leurs compétences et à leurs intérêts.
- Outils pour faciliter la gestion des missions, de la proposition initiale à la facturation et au paiement.
- Profils publics et évaluations permettant de se démarquer et de bâtir une réputation solide.
- Une vaste base de données de freelances avec des compétences variées et des profils vérifiés.
- Possibilité de trouver des talents pour des missions ponctuelles ou des projets de longue durée, selon les besoins de l'entreprise.
- Processus simplifié pour la recherche, la sélection et la gestion des freelances, avec un gain de temps et de ressources.

Limites de Malt

- Malt prélève une commission sur les revenus des freelances, ce qui peut réduire leurs gains nets. Les entreprises doivent également prendre en compte ces coûts supplémentaires dans leur budget.
- Avec un nombre croissant de freelances inscrits sur la plateforme, la concurrence peut être intense, rendant difficile pour certains freelances de se démarquer et de décrocher des missions.
- La concurrence peut également exercer une pression sur les tarifs, certains freelances étant prêts à accepter des missions pour

1.3.3 Expression du besoin

Cette partie est consacrée à l'examen de l'existant au niveau de l'entreprise. En effet, la première partie procure une idée sur l'existant et la deuxième présente une critique de ce dernier ce qui permette d'identifier les besoins du projet.

Après avoir étudié les différentes solutions existantes en ligne, Medianet consulting a pris conscience des défis liés à la recherche et l'embauche de nouveaux collaborateurs. Pour répondre à ces besoins, l'entreprise a décidé de développer une application efficace et de mettre en place une extension pour sa plateforme de recrutement. Cette solution digitale facilitera toutes les étapes nécessaires pour embaucher un nouveau collaborateur, depuis la recherche de profil jusqu'à son intégration dans l'entreprise.

1.4 Planification et conduite du projet

Dans le cadre du projet de gestion des experts, nous avons opté pour une décomposition des sprints basée sur les différents modules du système, afin d'assurer un développement structuré et cohérent. Chaque sprint a été dédié à un module spécifique, ce qui a permis de concentrer les efforts de développement sur un ensemble restreint de fonctionnalités à chaque itération.

Le choix de séparer les sprints par module a été fait dans le but de mieux maîtriser le processus de développement, en attribuant à chaque sprint un objectif clair et précis. Par exemple, un sprint entier était consacré au développement du module de **gestion des utilisateurs**, tandis qu'un autre sprint était dédié au module de **gestion des compétences**. De cette manière, on a pu se focaliser sur les aspects fonctionnels d'un module à la fois, réduisant ainsi la complexité des intégrations entre les différents composants.

Chaque sprint a donc représenté un module distinct du projet, ce qui a permis d'optimiser la gestion du temps, d'améliorer la qualité des livrables et de faciliter les tests. Cela a également favorisé la livraison incrémentale de chaque module, garantissant ainsi que les fonctionnalités principales étaient livrées et testées avant de passer aux modules suivants. Grâce à cette approche, le projet a pu avancer de manière fluide, tout en assurant une meilleure visibilité sur l'état de progression de chaque module.

1.4.1 Présentation de la méthode agile

La méthode agile est une approche de gestion de projet qui met l'accent sur la flexibilité, la collaboration et l'adaptabilité aux changements et aux nouvelles exigences, ce qui rend les projets plus réactifs et plus agiles. Contrairement aux méthodes classiques, les méthodes agiles encouragent une approche itérative et incrémentale du développement, où

les exigences et les solutions évoluent grâce à une collaboration continue entre l'équipe de développement et le client. L'approche agile favorise la collaboration étroite entre les membres de l'équipe, la communication fréquente avec le client et la rétroaction continue. L'objectif principal est de fournir rapidement et de manière itérative des fonctionnalités de haute qualité qui répondent aux besoins du client. Les méthodes agiles les plus connues sont SCRUM et Kanban.

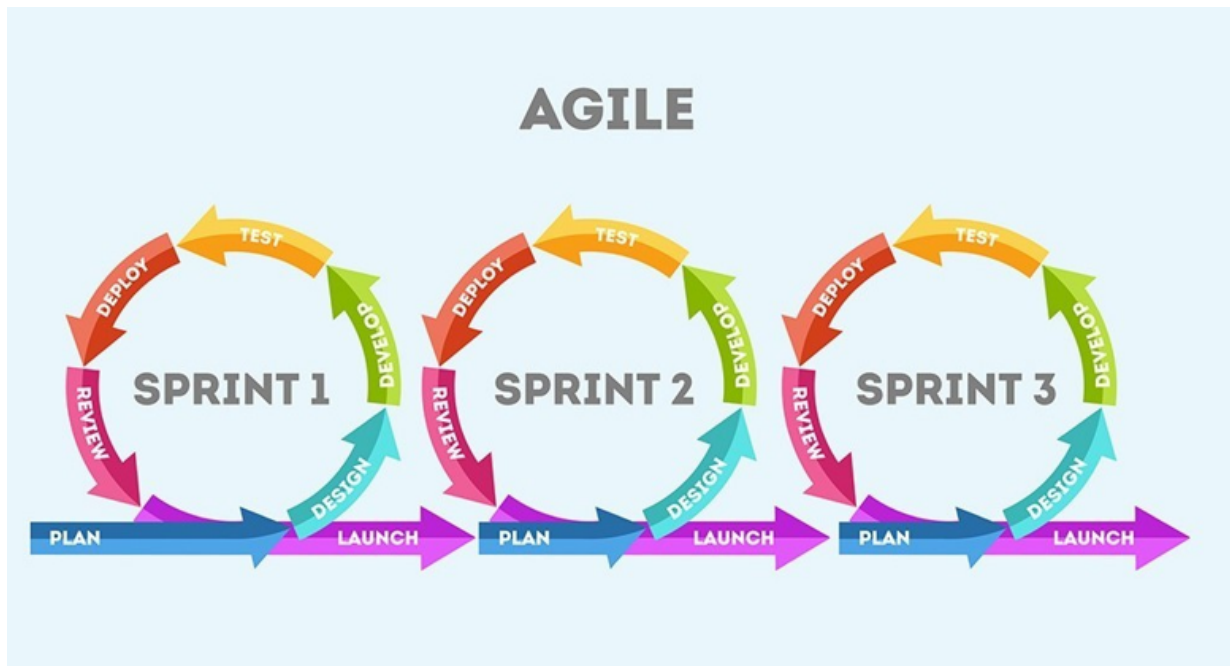


FIGURE 1.3 – La méthodologie agile

Pour la réalisation de notre projet. Dans la prochaine section, nous présenterons en détail la méthodologie Scrum, en mettant l'accent sur ses principes fondamentaux, ses éléments clés et son processus de travail.

1.4.2 Présentation de la methode Scrum

Scrum² est une méthodologie Agile populaire utilisée pour la gestion de projets, en particulier dans le développement de logiciels. Les tableaux ci dessous 1.1 , 1.2 et 1.3 représentent une comparaison entre Scrum et d'autres méthodologies de gestion de projets, comme le modèle en cascade (Waterfall), Kanban, et d'autres approches agiles :

2. <https://fr.wikipedia.org/wiki/Scrum>

- Scrum vs. Waterfall (Modèle en cascade)

	Scrum	Waterfall (Modèle en cascade)
Approche :	Agile et itératif. Les projets sont divisés en cycles de travail appelés sprints, qui durent généralement 2 à 4 semaines. À la fin de chaque sprint, une version potentiellement livrable du produit est présentée.	Approche linéaire et séquentielle. Chaque phase (conception, développement, test, etc.) doit être terminée avant de passer à la suivante. Il n'y a pas de retour en arrière, sauf en redémarrant complètement.
Flexibilité :	Très flexible. Les exigences peuvent évoluer au fur et à mesure du projet, et les équipes s'adaptent aux changements.	Rigide. Les exigences sont définies au début, et les modifications en cours de projet sont coûteuses et difficiles à intégrer.
Livraison :	Livraison continue et incrémentale de fonctionnalités. Chaque sprint peut produire une partie fonctionnelle du produit.	Livraison unique à la fin du projet. Le produit n'est généralement disponible qu'à la fin du cycle complet de développement.

TABLE 1.1 – Scrum vs. Waterfall

- Scrum vs. Kanban

	Scrum	Kanban
Approche :	Basé sur des sprints avec des cycles de travail définis, des rôles bien établis (Scrum Master, Product Owner, équipe de développement), et des cérémonies (sprint planning, rétrospective, etc.).	Une méthode Agile plus visuelle et fluide. Elle se concentre sur la gestion du flux de travail en limitant le nombre de tâches en cours (WIP - Work In Progress). Il n'y a pas de sprints fixes, mais plutôt un flux continu de tâches.

Cadence :	Les tâches sont planifiées et exécutées dans des itérations fixes (sprints). Chaque sprint a un objectif et une durée définie.	Pas de sprints définis. Le travail est géré en flux continu avec des tâches prises en charge dès qu'il y a de la capacité disponible.
Structure :	Très structuré avec des rôles spécifiques et des événements réguliers (réunions quotidiennes, rétrospectives).	Moins structuré, plus flexible. Il n'y a pas de rôles spécifiques ni d'événements obligatoires. L'accent est mis sur la visualisation du flux de travail et la réduction des goulots d'étranglement.

TABLE 1.2 – Scrum vs. Kanban

- Scrum vs. Kanban

	Scrum	Extreme Programming (XP)
Objectifs :	Axé sur la gestion de projet et la livraison de valeur. La gestion du backlog et la collaboration avec le Product Owner sont essentielles.	se concentre sur des pratiques comme le développement piloté par les tests (TDD), la programmation en binôme et la refactorisation, pour améliorer la qualité du code et faciliter la collaboration entre développeurs.
Cadence :	Travaille en sprints avec des objectifs bien définis pour chaque itération.	Utilise aussi des itérations courtes, mais met davantage l'accent sur les pratiques techniques et la qualité du code.

TABLE 1.3 – Scrum vs. Extreme Programming (XP)

Scrum : Aujourd'hui « Scrum est la méthode agile la plus populaire. Ce terme signifie mêlée au rugby. La méthode scrum s'appuie sur des « sprints qui sont des espaces temps assez courts pouvant aller de quelques heures jusqu'à un mois. Généralement et de préférence un sprint s'étend sur deux semaines. À la fin de chaque sprint, l'équipe présente ce qu'elle a ajouté au produit ³.

1.4.3 Processus de développement

La nature du projet et sa forte dépendance aux acteurs du domaine sont les raisons qui expliquent le fait d'être toujours à l'écoute du client et prêt à répondre à ses nouveaux besoins. C'est pour cela, l'équipe du projet a opté pour un cycle de développement agile et plus précisément SCRUM.

Le principe de la méthodologie SCRUM est de développer un logiciel de manière incrémentale en maintenant une liste totalement transparente des demandes d'évolutions ou de corrections à implémenter (backlog).

Avec des livraisons très fréquentes, toutes les 4 semaines en général, le client reçoit un logiciel à chaque itération. Plus nous avançons dans le projet, plus le logiciel est complet et possède de plus en plus de fonctionnalités.

Pour cela, la méthode s'appuie sur des développements itératifs à un rythme constant d'une durée de 2 à 4 semaines comme le montre la figure 1.4

Le schéma illustre un exemple de planification en Scrum : les itérations (sprints) durent en pratique entre 2 et 4 semaines, et possède chacune un but. Le but de chaque sprint une liste d'items du backlog de produit ou de fonctionnalités à réaliser. Ces items sont décomposés par l'équipe en tâches élémentaires de quelques heures.

Comme nous pouvons le remarquer dans cette figure, pour mettre en place la méthode SCRUM, il faut tout d'abord définir les différentes fonctionnalités de notre application qui forment le backlog du produit. Ensuite, vient l'étape de la planification du sprint pour définir le plan détaillé d'une itération.

3. <https://www.tuleap.org>

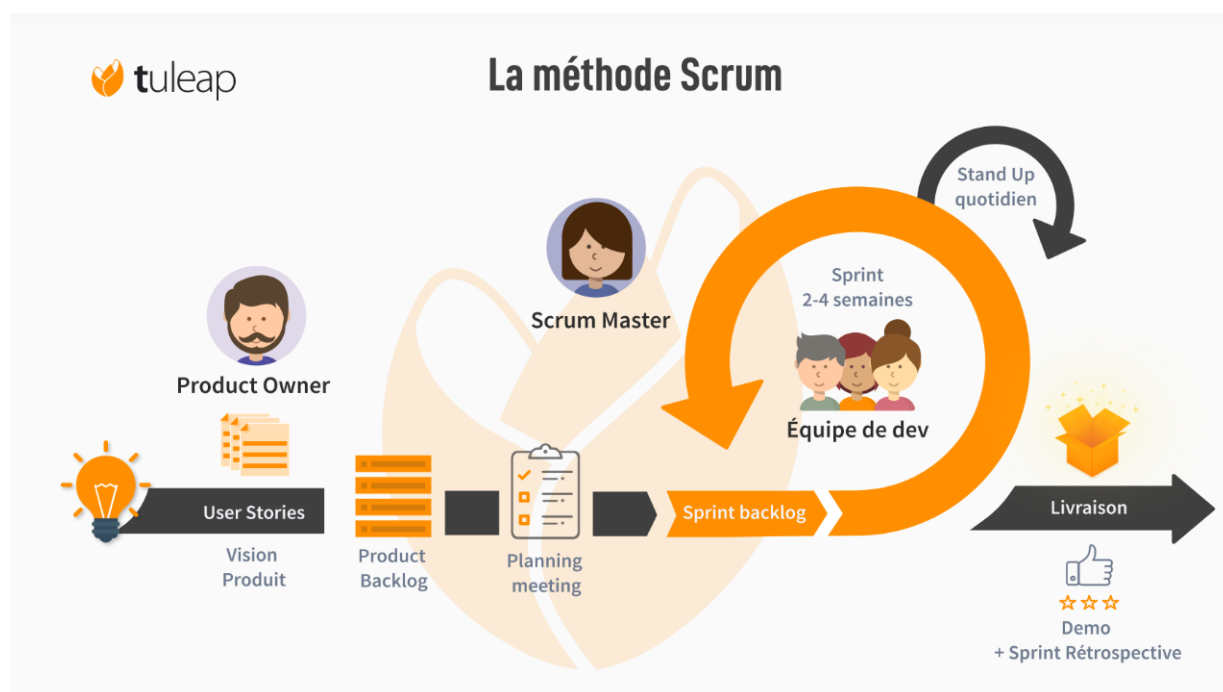


FIGURE 1.4 – La méthode scrum

Durant un sprint, il y a toujours des réunions quotidiennes entre les différents collaborateurs du projet afin de présenter l'état d'avancement des différentes tâches en cours, les difficultés rencontrées ainsi que les tâches restantes à réaliser. Une fois le produit partiel est prêt, nous vérifions la conformité de ce qui a été fait durant le sprint et nous pouvons alors l'améliorer en procédant à l'étape de rétrospective.

La méthode agile est une approche de gestion de projet qui met l'accent sur la flexibilité, la collaboration et l'adaptabilité aux changements et aux nouvelles exigences, ce qui rend les projets plus réactifs et plus agiles. Contrairement aux méthodes classiques, les méthodes agiles encouragent une approche itérative et incrémentale du développement, où les exigences et les solutions évoluent grâce à une collaboration continue entre l'équipe de développement et le client.

L'approche agile favorise la collaboration étroite entre les membres de l'équipe, la communication fréquente avec le client et la rétroaction continue. L'objectif principal est de fournir rapidement et de manière itérative des fonctionnalités de haute qualité qui répondent aux besoins du client. Les méthodes agiles les plus connues sont SCRUM et Kanban.

Après avoir comparé les deux méthodes et discuté avec le responsable du projet, nous avons opté pour l'utilisation de la méthodologie Agile, et plus spécifiquement Scrum, pour la réalisation de notre projet. Dans la prochaine section, nous présenterons en détail la méthodologie Scrum, en mettant l'accent sur ses principes fondamentaux, ses éléments clés et son processus de travail.

Dans notre projet, l'adoption de la méthodologie Agile Scrum s'est révélée bénéfique à plusieurs égards. Tout d'abord, cette approche nous a permis de gérer efficacement les changements et les évolutions des spécifications du projet. Scrum nous a offert la flexibilité nécessaire pour ajuster notre planification et notre travail en fonction des besoins changeants, en mettant l'accent sur la livraison régulière d'incréments fonctionnels.

1.5 Méthodologies Adoptées

Dans notre projet, l'adoption de la méthodologie Agile Scrum s'est révélée bénéfique à plusieurs égards. Tout d'abord, cette approche nous a permis de gérer efficacement les changements et les évolutions des spécifications du projet. Scrum nous a offert la flexibilité nécessaire pour ajuster notre planification et notre travail en fonction des besoins changeants, en mettant l'accent sur la livraison régulière d'incréments fonctionnels. En utilisant des cycles courts appelés Sprints, nous avons pu découper notre travail en itérations gérables. À chaque Sprint, nous avons sélectionné un ensemble de fonctionnalités prioritaires à implémenter, en nous concentrant sur leur réalisation de manière itérative. Cette approche itérative nous a permis de progresser de manière incrémentale, en fournissant des résultats tangibles à la fin de chaque itération. Elle nous a également offert l'opportunité de recueillir des retours fréquents de nos parties prenantes, ce qui nous a aidés à nous ajuster et à améliorer nos résultats au fur et à mesure de l'avancement du projet.

Un élément essentiel de Scrum est son orientation vers l'équipe de travail. Chaque membre de l'équipe joue un rôle clé et est responsable de ses propres tâches. La collaboration et la communication entre les membres de l'équipe sont encouragées, favorisant ainsi l'échange d'idées, le partage des connaissances et la résolution collective des problèmes. Cette dynamique d'équipe a contribué à renforcer notre cohésion et notre engagement envers le projet, tout en maximisant notre productivité.

Pour garantir une gestion efficace du projet, nous adoptons la méthodologie Agile avec Scrum :

- **Scrum** : Assure une vision d'ensemble optimale et permet de gérer les changements grâce à des sprints courts.
- **Rôles clés** :
 - **Product Owner** : Définit et priorise les fonctionnalités (Monsieur Ali Maatig).
 - **Scrum Master** : Garantit l'application de Scrum et la résolution des obstacles (Monsieur Hassen Lazrag).
 - **Équipe de Développement** : Organisée pour accomplir les tâches des sprints (Monsieur Sofiene Khannoussi).

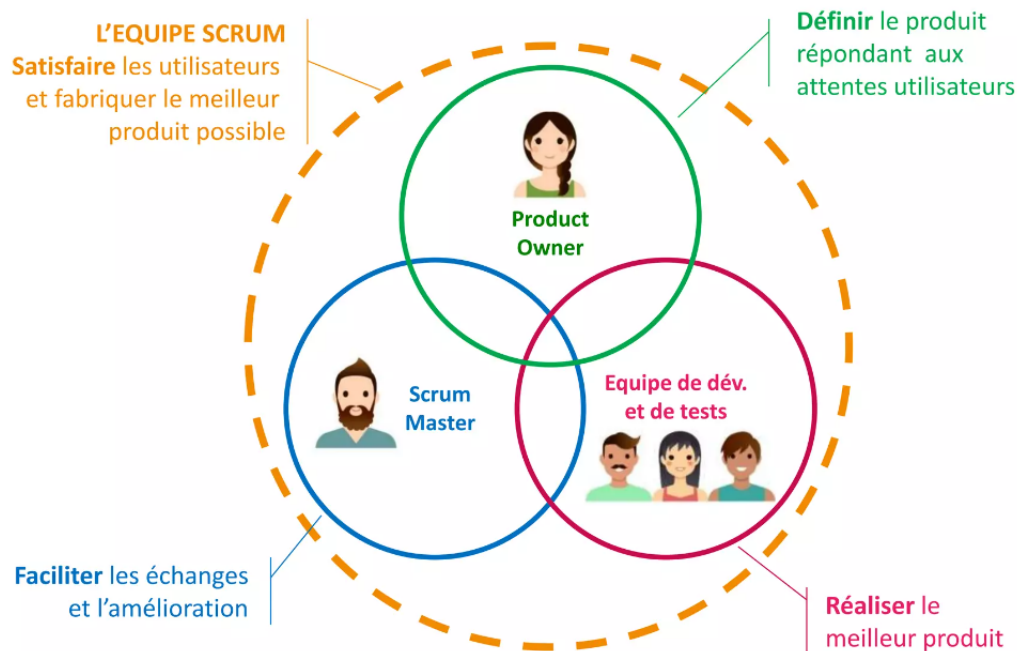


FIGURE 1.5 – La méthode scrum

1.6 Spécification des besoins

Le rôle de cette partie est l'identification des acteurs ainsi que la spécification des besoins fonctionnelles et non fonctionnels.

1.6.1 Identification des acteurs

- **Admin** : L'admin a accès à toutes les fonctionnalités du site : il peut gérer la liste des experts et la liste des sociétés, modérer la liste des missions et gérer les contrats.
- **Manager** : Un manager a pour responsabilité de gérer les demandes et les besoins de son équipe en ce qui concerne de nouveaux profils et postes. Il doit également s'occuper de la gestion des contrats. De plus, il est chargé de recueillir les retours et les commentaires, ainsi que de mener les entretiens annuels, tout en ayant la possibilité de consulter les statistiques.
- **Expert** : Un expert est responsable de renseigner ses compétences et de maintenir son profil à jour. Il peut consulter et postuler aux missions disponibles, ainsi que gérer les contrats qui lui sont attribués. L'expert peut également fournir des retours sur les missions effectuées et consulter les statistiques personnelles liées à ses performances.

1.6.2 Analyse des besoins globaux

Cette partie a pour objectif de présenter les actions que le système doit réaliser et les objectifs du projet, ainsi que de simplifier le processus de développement. Cette partie est essentielle pour orienter les efforts de développement et assurer que le système réponde aux besoins et aux attentes des utilisateurs.

Les besoins fonctionnels

Les besoins fonctionnels font référence aux fonctionnalités et aux actions que le système doit être capable de réaliser pour répondre aux besoins et aux attentes des utilisateurs. Ils décrivent les tâches que les utilisateurs doivent pouvoir accomplir à travers le système. Dans notre cas, notre système doit répondre aux besoins suivants :

- **Gestion des demandes des managers** : offrir un espace digital où les managers peuvent définir leurs besoins, les profils requis ainsi que les postes ouverts.
- **Gestion des compétences et CV** : Offrir une interface où les experts peuvent ajouter des compétences qui seront utilisées pour créer des profils.
- **Gestion des missions** : permettre au manager de gérer les offres de missions.
- **Gestion des candidatures spontanées** : Le manager a le droit de gérer les demandes de candidature de chaque expert. Pour chaque demande, il reçoit le CV téléchargé par l'expert. Lors de la soumission de la candidature, celle-ci est envoyée au manager avec les détails du candidat ainsi que le pourcentage de correspondance. Ce pourcentage est calculé par un outil d'OCR qui compare le contenu du CV avec la description de la mission.
- **Analyse et extraction des informations textuelles via l'OCR** : Les CV téléchargés seront analysés par un processus OCR pour extraire les informations textuelles. Ensuite, les CV seront filtrés en fonction des critères prédéfinis, facilitant ainsi la sélection des experts les plus pertinents pour les offres de mission.
- **Gestion des évaluations post-intégration** : permettre aux managers de gérer les candidatures afin de valider un expert le plus adéquat à la demande d'offre de mission.
- **Gestion des contrats** : Lors de la validation d'un expert, le manager doit créer un contrat avec ce dernier contenant les différentes clauses liées à la mission.
- **Gestion des statistiques** : cette fonctionnalité permet d'afficher des statistiques en fonction de différents critères et de les exporter.

Besoins non fonctionnels

Notre application doit répondre à plusieurs besoins non fonctionnels essentiels pour assurer sa qualité et sa réussite. Parmi ces besoins non fonctionnels, on peut citer :

- **Performance** : Il est crucial que notre application soit rapide, efficace et utilise les ressources disponibles de manière optimale pour répondre aux attentes des utilisateurs et fournir une expérience utilisateur fluide.
- **Fiabilité** : Nous devons nous assurer que notre application est stable, disponible en permanence et capable de tolérer les pannes pour garantir une utilisation fiable et constante.
- **Sécurité** : Il est primordial que notre application soit sécurisée pour protéger les données sensibles des utilisateurs et éviter les vulnérabilités qui pourraient compromettre sa sécurité.
- **Convivialité** : Nous devons garantir que notre application est facile à utiliser, intuitive et ergonomique pour offrir une expérience utilisateur agréable et facilement accessible.
- **Compatibilité** : Nous devons nous assurer que notre application est compatible avec les systèmes d'exploitation, les navigateurs web et les plateformes mobiles pour garantir une expérience utilisateur homogène et sans faille.
- **Maintenabilité** : Il est important que notre application soit facile à maintenir et à mettre à jour pour garantir sa durabilité et sa capacité à évoluer avec le temps.
- **Portabilité** : Nous devons nous assurer que notre application est facilement transportable sur différentes plateformes et appareils pour permettre aux utilisateurs d'accéder à l'application où qu'ils se trouvent.

En répondant à ces besoins non fonctionnels, nous pouvons offrir une application de qualité supérieure qui répond aux attentes des utilisateurs et offre une expérience utilisateur positive et satisfaisante.

1.7 Planification des Sprints

L'élaboration du Backlog et la planification des sprints sont des aspects essentiels dans la gestion d'un projet agile, tels que le développement d'une application ou d'un système informatique. Ces éléments jouent un rôle crucial en garantissant l'efficacité, la transparence et la souplesse du processus de développement.

Release ID	Nom du Sprint
1	Sprint 1 : Gestion des Comptes Admin, Manager, Expert , et Inscription.
2	Sprint 2 : Gestion de compétences et gestion de CV
3	Sprint 3 : Gestion des missions
4	Sprint 4 : Gestion des contrats
5	Sprint 5 : Gestion des tableaux de bord et messagerie interne

TABLE 1.4 – Répartition des releases

La planification des sprints, élément clé de la méthodologie Agile, est le processus par lequel les équipes de développement définissent les objectifs et les tâches pour une itération donnée. Ce processus fournit un cadre itératif et structuré, facilitant la livraison continue de nouvelles fonctionnalités. Dans cette section, nous examinons en détail la planification des sprints, en mettant l'accent sur la sélection stratégique des éléments du Backlog, l'estimation des efforts nécessaires, et l'élaboration d'un plan d'action précis pour chaque itération. Découvrez les éléments essentiels d'une planification efficace, alliant souplesse et résultats concrets.

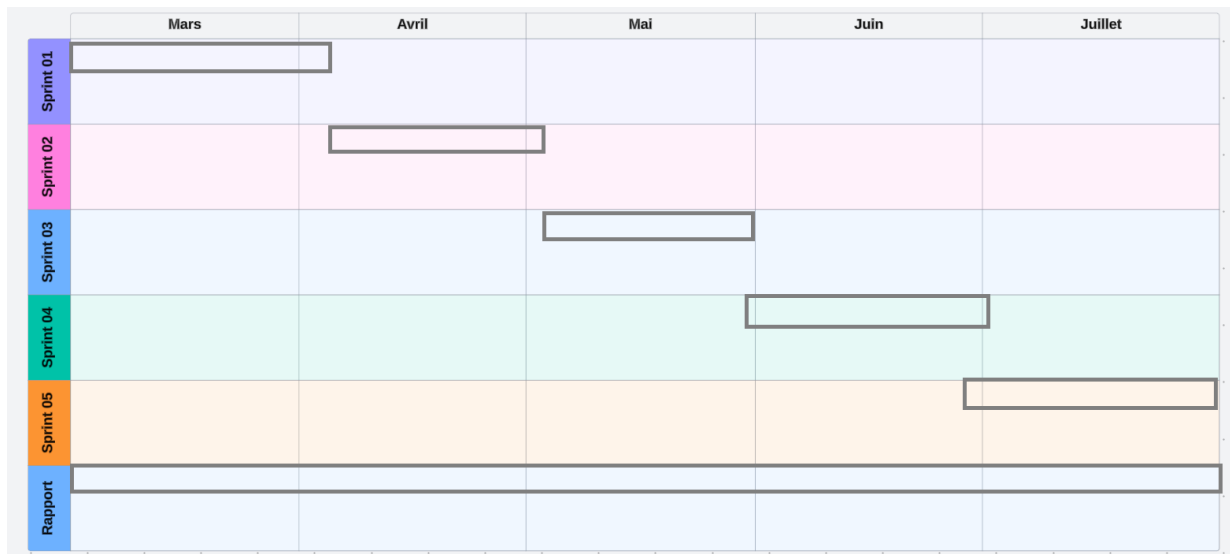


FIGURE 1.6 – Planning du projet

1.8 Architecture de l'application

Cette section consiste à décrire l'architecture générale du système donc elle va nous présenter son architecture physique puis son architecture logique. L'architecture est le point de départ ou la feuille de route pour créer une application. Vous devrez cependant effectuer les choix de mise en œuvre que l'architecture n'inclut pas. Par exemple, le choix du langage de programmation est la première étape pour écrire une application.

1.8.1 Architecture Logique

Notre application repose sur une architecture monolithique, où l'ensemble des fonctionnalités sont intégrées dans une seule base de code. Contrairement à une approche microservices, toutes les parties de l'application sont interconnectées et dépendent les unes des autres dans une structure unifiée. Cette architecture simplifie le déploiement, car une seule application est développée et mise en production. Elle favorise une gestion centralisée des ressources et des services, mais nécessite une coordination rigoureuse lors des évolutions ou des mises à jour pour garantir la cohérence et la stabilité du système global.

Backend	Frontend
Nous avons choisi Spring Boot pour le backend en raison de sa robustesse, de sa flexibilité et de sa facilité de configuration, ce qui nous permet de développer rapidement des applications Java hautement performantes. En utilisant Spring Boot, nous bénéficions également de son écosystème riche en outils et en bibliothèques, ce qui facilite l'intégration avec d'autres technologies et la mise en œuvre de fonctionnalités avancées.	Nous avons opté pour Angular pour le frontend en raison de sa structure modulaire et de sa capacité à créer des interfaces utilisateur interactives et dynamiques. Avec Angular, nous pouvons tirer parti de ses fonctionnalités avancées telles que le binding de données bidirectionnel, la gestion efficace des états de l'application et une architecture orientée composants, facilitant ainsi le développement et la maintenance de notre application web.

TABLE 1.5 – Parties du code

- **Sécurité JWT** : Nous avons implémenté la sécurité JWT (JSON Web Token) pour gérer de manière sécurisée les identités et les accès dans notre application. En utilisant JWT, nous pouvons mettre en place des mécanismes d'authentification et d'autorisation robustes, garantissant ainsi la confidentialité des données et la sécurité de l'ensemble du système. Les tokens JWT permettent de sécuriser les communications entre les différents services de manière décentralisée, tout en offrant une flexibilité accrue pour la gestion des sessions utilisateur.
- **REST API** : Nous avons mis en œuvre une architecture RESTful pour notre API, offrant ainsi une approche flexible et scalable pour la communication entre le frontend et le backend. En utilisant des principes REST, nous favorisons l'interopérabilité, la simplicité et la facilité de maintenance de notre API, ce qui permet une intégration harmonieuse avec d'autres systèmes et une évolutivité efficace de notre application.

Architecture Logique de Spring Boot

- Côté Serveur (Spring Boot) :

Configuration de la sécurité avec Spring Security pour gérer l'authentification et l'autorisation.

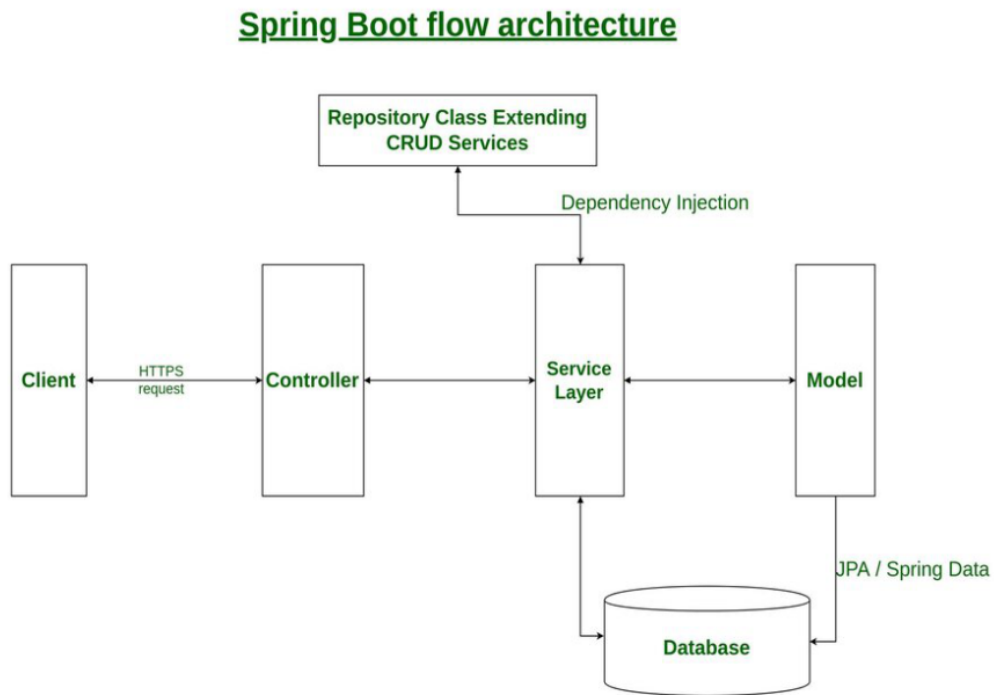


FIGURE 1.7 – Architecture du flux de démarrage Spring

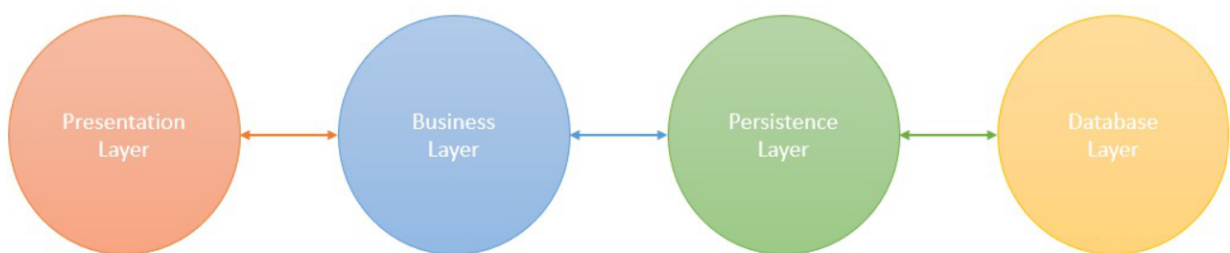


FIGURE 1.8 – Les couches de Spring Boot

Architecture logique d'Angular

Angular est un framework pour créer la partie Front End des applications web en utilisant HTML et JavaScript ou TypeScript compilé en JavaScript. Une application Angular se

compose d'un ou plusieurs modules, dont un est principal. Chaque module peut inclure :

- **Composants web** : La partie visible de l'application Web (IHM).
- **Services pour la logique applicative** : Les composants peuvent utiliser les services via le principe de l'injection des dépendances.
- **Directives** : Un composant peut utiliser des directives.
- **Pipes** : Utilisés pour formater l'affichage des données dans les composants.
- Côté Client (Angular) :

Utilisation de services d'authentification pour gérer les jetons JWT. Gardiens de routes pour protéger les routes et vérifier les permissions de l'utilisateur.

Cette architecture modulaire permet de séparer les préoccupations entre le front-end et le back-end, facilitant ainsi le développement, la maintenance et l'évolution de l'application. En utilisant Angular et Spring Boot, on bénéficie des avantages d'un framework front-end moderne et d'une puissante infrastructure back-end, offrant une base solide pour la création de plateformes de gestion des experts ou d'autres types d'applications web complexes.

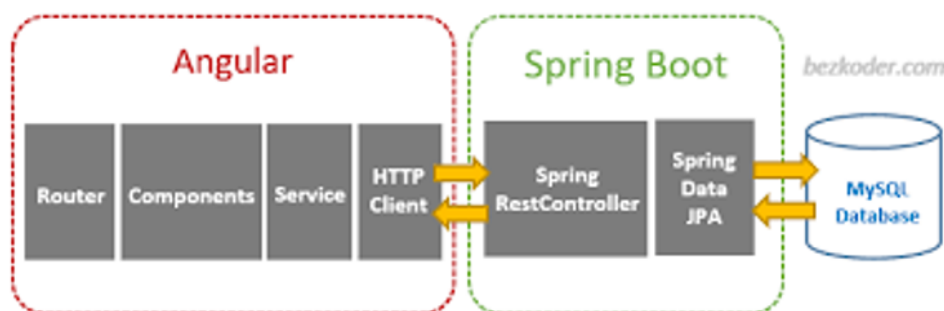


FIGURE 1.9 – Architecture logique de l'application.

1.8.2 Architecture Physique

Notre application adopte une architecture 3-tiers qui permet une séparation claire des responsabilités et une gestion efficace des opérations de données et facilite la maintenance et l'évolution de l'application.

- Le premier tier est le composant de l'application, qui reçoit les requêtes HTTP d'une application cliente externe. Il agit comme une interface de communication.
- Le deuxième tier est la logique métier de l'application. Il traite les requêtes reçues, effectue les opérations nécessaires sur les données et renvoie les résultats.
- Le troisième tier est la couche de données, où toutes les informations sont stockées de manière sécurisée et fiable.

L'application est divisée en trois couches :

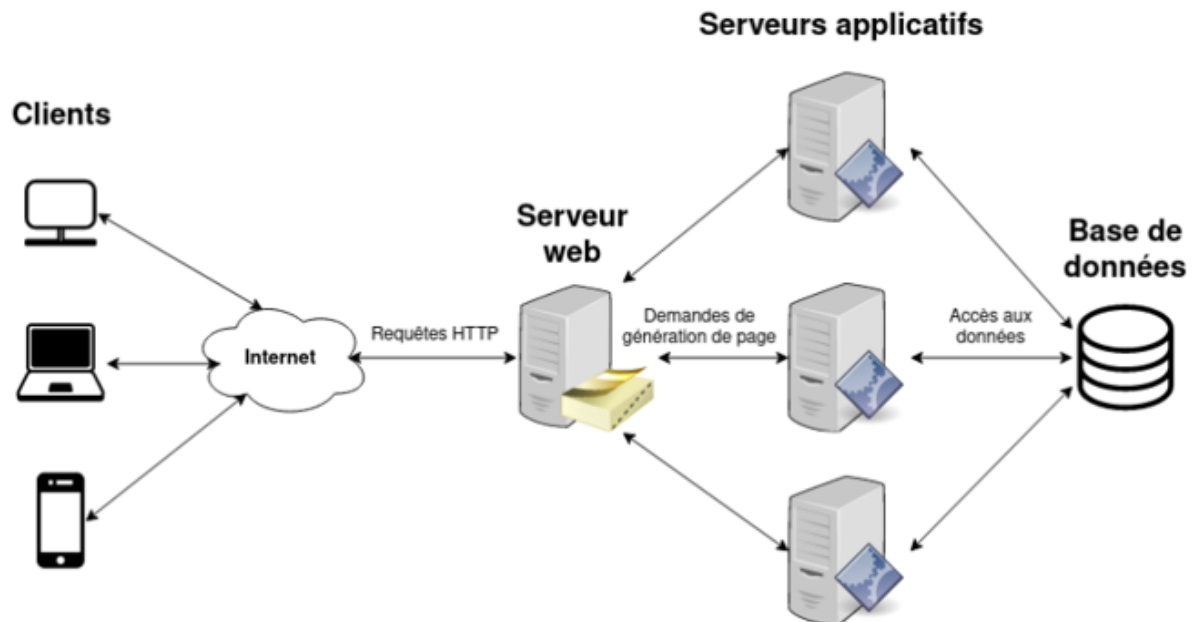


FIGURE 1.10 – Architecture physique du déroulement de l'application

- **Couche de Présentation** : Angular, déployée sur un serveur web avec Docker.
- **Couche Logique** : Spring Boot, conteneurisé avec Docker.
- **Couche de Données** : Bases de données MySQL, conteneurisée avec Docker.

1.9 Technologies et outils utilisés

Nous avons consacré cette partie à la présentation des technologies et des outils de travail utilisés pour la réalisation de notre projet.

1.9.1 Environnement matériel

Pour la réalisation de notre projet, nous avons utilisé un ordinateur portable avec les caractéristiques suivantes, telles qu'indiquées dans le tableau ci-dessous.

Ordinateur	
Propriétaire	Sofiène Khannoussi
Processeur	Intel(R) Core(TM) i5-10310U CPU @ 1.70GHz 2.21 GHz
Mémoire RAM installé	16 Go
Type de système	Système d'exploitation 64 bits
Système d'exploitation	Windows 10 Entreprise

TABLE 1.6 – Environnement matériel.

1.9.2 Environnement Logiciel

Dans cette section, nous vous présentons l'environnement logiciel et les langages que nous avons utilisés lors du développement de notre application.

Présentation des langages de développement

- **CSHTML** : Une extension de fichier utilisée dans le développement web avec ASP.NET. Il s'agit d'un fichier qui combine du code C-sharp avec du code HTML.
- **Css** : (Cascading Style Sheets) permet de créer des pages web à l'apparence soignée. Cet article vous propose de lever le voile en expliquant ce qu'est CSS ; un exemple simple en présentera la syntaxe, puis quelques termes clés du langage seront introduits. CSS est un langage basé sur des règles — on définit des règles de styles destinées à des éléments ou des groupes d'éléments particuliers dans la page.



FIGURE 1.11 – Logo Ccss

- **Javascript** : (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que Node.js, Apache CouchDB voire Adobe Acrobat. Le code JavaScript est interprété ou compilé à la volée (JIT). C'est un langage à objets utilisant le concept de prototype, disposant d'un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orientée objet. JavaScript ne doit pas être confondu avec le langage de programmation Java. Java et JavaScript sont deux marques déposées par Oracle dans de nombreux pays

mais ces deux langages de programmation ont chacun une syntaxe, une sémantique et des usages différents.



FIGURE 1.12 – Logo Javascript

- **Json** : JavaScript Object Notation est un format de données textuel dérivé de la notation des objets du langage JavaScript. Il concurrence XML pour la représentation et la transmission d'information structurée¹.

Créé par Douglas Crockford entre 2002 et 2005², la première norme du JSON est ECMA-4043 d'Ecma International qui a été publiée en octobre 2003⁴. Il est également décrit en 2017 par la RFC 82595 de l'Internet Engineering Task Force qui se veut compatible avec Ecma-404 et ECMA-404.



FIGURE 1.13 – Logo Json

- **Jquery** : est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web². La première version est lancée en janvier 2006 par John Resig.

Le but de la bibliothèque étant le parcours et la modification du DOM (y compris le support des sélecteurs CSS 1 à 3 et un support basique de XPath), elle contient de nombreuses fonctionnalités ; notamment des animations, la manipulation des feuilles de style en cascade (accessibilité des classes et attributs), la gestion des événements, etc. L'utilisation d'Ajax est facilitée et de nombreux plugins sont présents.



FIGURE 1.14 – Logo JQuery

- **SQL** : Un langage utilisé pour travailler avec les bases de données. Il permet de manipuler les données stockées dans une base de données. SQL est utilisé dans de nombreux systèmes de gestion de bases de données relationnelles.

Framework utilisés

Les logiciels utilisés dans notre projet sont les suivants :

- **Spring Boot** : est un Framework Java open source utilisé pour programmer des applications Spring autonomes de qualité production avec un minimum d'effort. Spring Boot est une extension de convention sur configuration pour la plateforme Spring Java destinée à aider à minimiser les problèmes de configuration lors de la création d'applications basées sur Spring. Il est largement utilisé pour créer des micro services , des applications Web et d'autres projets basés sur Java en raison de sa facilité d'utilisation et de sa robustesse.



FIGURE 1.15 – Logo Spring Boot

- **Angular** : Angular est un Framework open source, basé sur TypeScript Il permet la création d'applications Web et plus particulièrement d'applications Web monopages : des applications Web accessibles via une page Web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.



FIGURE 1.16 – Logo Angular

Logiciels utilisés :

Les logiciels utilisés dans notre projet sont les suivants :

- **Draw.io** : Une application web qui permet de créer facilement des diagrammes et des schémas. C'est un outil pratique pour dessiner des organigrammes, des diagrammes de flux, des diagrammes UML et bien d'autres.



FIGURE 1.17 – Logo Draw.io

- **Visuel Studio code** : Un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS³. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code (IntelliSense⁴), les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.



FIGURE 1.18 – Logo Visuel Studio code

- **Overleaf** : Une plateforme en ligne qui facilite la rédaction collaborative de documents LaTeX. Cette plateforme offre une interface conviviale pour créer et éditer des documents scientifiques, tout en simplifiant la gestion des références, des tableaux et des figures. Utiliser Overleaf permet d'assurer une présentation professionnelle du rapport.



FIGURE 1.19 – Logo Overleaf

- **MySQL DB** : Nous avons sélectionné MySQL comme système de gestion de base de données en raison de sa fiabilité, de ses performances et de sa large adoption dans l'industrie. En utilisant MySQL, nous pouvons stocker et gérer efficacement les données de notre application, tout en bénéficiant de fonctionnalités avancées telles que la réplication, la gestion des transactions et la compatibilité avec de nombreux frameworks et outils de développement.



FIGURE 1.20 – Logo MySQL

- **GitHub** : Un système de gestion de versions, ou VCS, suit l'historique des modifications quand des personnes et équipes collaborent sur des projets. Lorsque les développeurs apportent des modifications au projet, toute version antérieure du projet peut être récupérée à tout moment.



FIGURE 1.21 – Logo GitHub

- **Langage de modélisation** : Nous avons choisi d'utiliser UML, qui est un langage de modélisation standard largement utilisé dans le domaine du génie logiciel. UML offre une notation graphique et un ensemble de concepts permettant de représenter et de documenter les différents aspects d'un système logiciel.



FIGURE 1.22 – Logo StarUML

- **Tesseract** : est une bibliothèque open source d'OCR développée par HP et maintenue par Google. Elle convertit des images de texte en texte éditable, supporte plus de 100 langues et utilise des réseaux neuronaux pour une haute précision. Compatible avec divers formats d'image, elle est idéale pour extraire du texte de documents scannés et de photos.



FIGURE 1.23 – Logo Tesseract

1.10 Conclusion

Dans ce chapitre, nous avons analysé l'existant au niveau de l'entreprise. Ensuite, nous avons identifié les besoins fonctionnels et non fonctionnels. Ainsi, nous avons pu définir le périmètre de notre travail, ce qui va nous permettre d'entamer la partie conception de la solution. Cette phase sera décrite dans le chapitre suivant.

Chapitre 2

Conception

2.1 Introduction

Ce chapitre détaille la conception du système de gestion des experts, une étape cruciale dans le cycle de vie du développement logiciel. La conception traduit les exigences fonctionnelles et non fonctionnelles en une architecture technique robuste, évolutive et maintenable. Nous allons explorer divers aspects de la conception, notamment les diagrammes de classes, d'architecture, de séquence, d'activités, et de cas d'utilisation, chacun apportant une vision claire sur les différentes dimensions du système.

2.2 Diagramme de Classes Global

Le diagramme de classes est un outil en génie logiciel permettant de visualiser les classes et interfaces d'un système, ainsi que leurs relations. Faisant partie de la dimension statique de l'UML, il ne se concentre ni sur les aspects temporels ni sur les aspects dynamiques. Une classe représente les responsabilités, les comportements et les types d'un ensemble d'objets. Les objets qui appartiennent à cet ensemble sont appelés les instances de la classe. Une classe regroupe des fonctions et des données (attributs) reliées par un contexte sémantique commun. Utilisée en programmation orientée objet, elle permet de modéliser un programme en décomposant une tâche complexe en plusieurs sous-tâches plus simples. Les classes peuvent être reliées entre elles par l'héritage, un mécanisme qui met en évidence des relations hiérarchiques. D'autres types de relations peuvent exister entre des classes, chacune étant représentée par un type spécifique de lien dans le diagramme de classes. Enfin, les classes sont instanciées pour créer des objets. Une classe peut être vue comme un modèle pour les objets : elle définit leurs caractéristiques, tandis que les objets, lorsqu'ils sont instanciés, possèdent leurs propres valeurs pour chacune caractéristiques.

Description textuelle du Diagramme de Classe Global :

- **Manager :**
 - **Rôle :**
 - Manager est responsable de la gestion des missions et des experts associés.
 - Il peut publier des missions, évaluer des experts, et gérer l'état des missions
 - **Attributs et méthodes :**
 - publier Mission(mission : Mission) : void : Publie une nouvelle mission.
 - évaluer Expert(expert : Expert, feedback : Feedback) : void : Évalue un expert en fonction des retours.
 - gérer Mission(mission : Mission) : void : Gère les missions assignées.
- **Expert :**
 - **Rôle :**
 - Représente une personne ayant des compétences spécifiques et pouvant postuler pour des missions.
 - **Attributs et méthodes :**
 - profil : string : Profil de l'expert.
 - compétences : list[str] : Liste des compétences que l'expert possède
 - postuler Mission (mission : Mission) : bool : Permet à l'expert de postuler pour une mission.
 - consulter Feedback() : listFeedback : Permet à l'expert de consulter les retours sur ses missions.
- **Mission :**
 - **Rôle :**
 - Représente une mission à accomplir, avec des compétences requises et un état de progression.
 - **Attributs et méthodes :**
 - idMission : int : Identifiant unique de la mission.
 - titre : string : Titre de la mission.
 - description : string : Description détaillée de la mission.
 - compétences requises : list : Liste des compétences nécessaires pour la mission.
 - statut : string : Statut actuel de la mission.
 - attribuer Expert(expert : Expert) : void : Attribue un expert à la mission.
 - mettre a Jour Statut(nouveau statut : string) : void : Met à jour le statut de la mission.
- **Compétence :**
 - **Rôle :**
 - Représente une compétence spécifique que les utilisateurs (experts) peuvent posséder.

- **Attributs et méthodes :**
 - compId : int : Identifiant unique de la compétence.
 - titre : string : Titre de la mission.
 - compNom : string : Nom de la compétence.
 - ajouterCompetence() : string : Ajoute une nouvelle compétence.
 - modifierCompetence() : string : Modifie une compétence existante.
 - desactiverCompetence() : string : Désactive une compétence.
 - afficherCompetence() : string : Affiche les détails d'une compétence.
- **Utilisateur :**
 - **Rôle :**
 - Représente un utilisateur du système, avec des informations de connexion et des rôles spécifiques.
 - **Attributs et méthodes :**
 - UserID : int : Identifiant unique de l'utilisateur.
 - UserNom : string : Nom de l'utilisateur.
 - UserPrenom : string : Prénom de l'utilisateur.
 - Email : string : Adresse email de l'utilisateur.
 - MotDePasse : string : Mot de passe pour la connexion.
 - Role : string : Rôle de l'utilisateur (par exemple : expert, manager, etc.).
 - seConnecter() : bool : Permet à l'utilisateur de se connecter au système.
 - seDéconnecter() : void : Permet à l'utilisateur de se déconnecter du système.
- **Administrateur :**
 - **Rôle :**
 - L'administrateur est responsable de la gestion des utilisateurs, des catégories, du contenu, et des statistiques du système.
 - **Attributs et méthodes :**
 - gérerUser(utilisateur : Utilisateur) : void : Gère les utilisateurs du système.
 - gérerCatégories(catégorie : Catégorie) : void : Gère les catégories au sein du système.
 - gérerContenu(contenu : string) : void : Gère le contenu publié sur le système.
 - consulterStatistiques() : dict : Permet de consulter les statistiques du système.
- **Contrat :**
 - **Rôle :**
 - Représente un contrat entre une société et un expert pour une mission spécifique.
 - **Attributs et méthodes :**
 - Description : string : Description du contrat.
 - periode : date : Période pendant laquelle le contrat est valide.
 - offreId : int : Identifiant de l'offre associée au contrat.

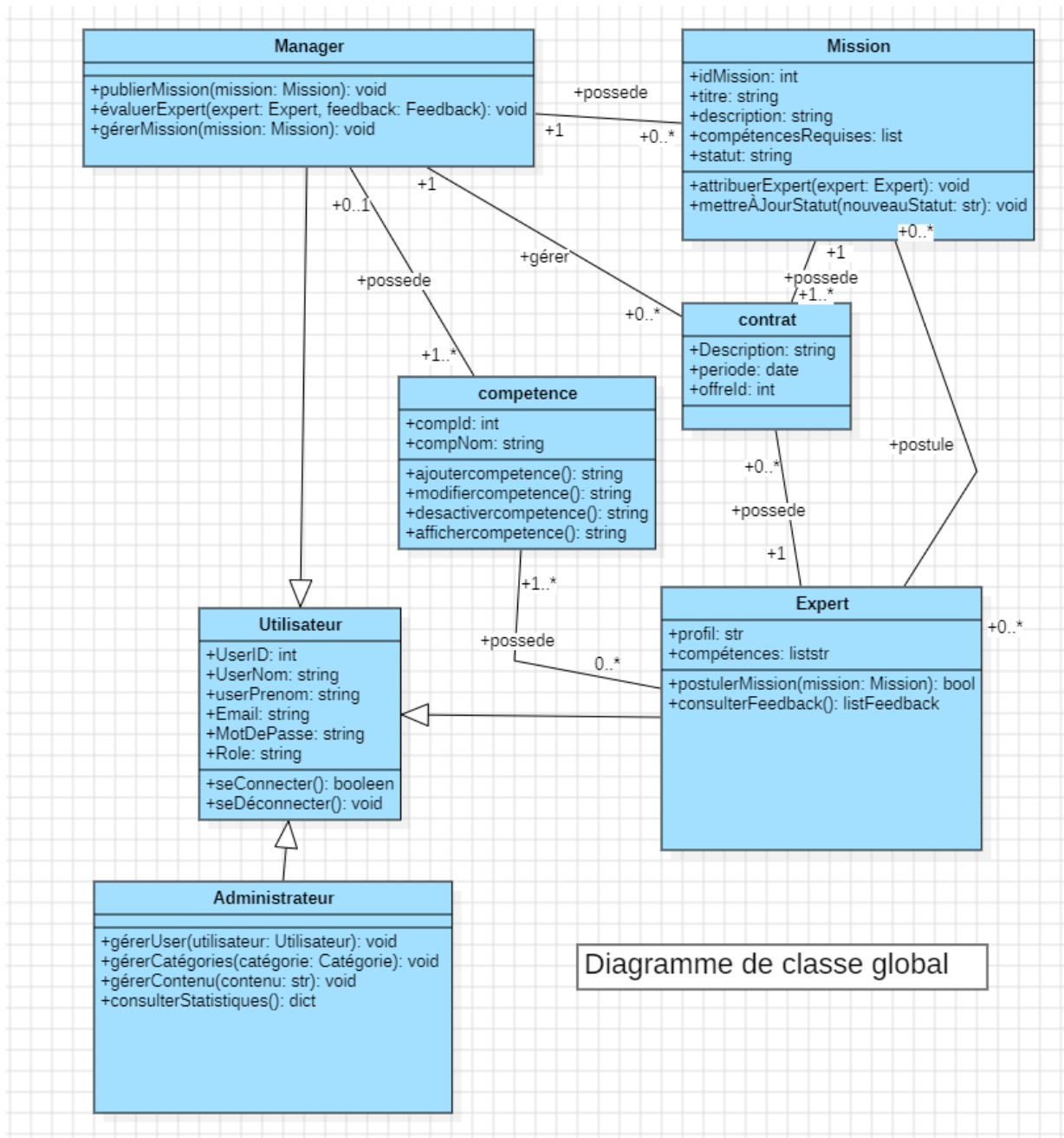


FIGURE 2.1 – Diagramme de classe global

2.3 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation met en évidence les fonctionnalités du système du point de vue des utilisateurs, y compris les interactions entre les acteurs (Experts, Managers, Admin) et le système.

2.3.1 Diagramme de cas d'utilisation général

Ce diagramme illustre le cas d'utilisation générale de notre système. Ces cas d'utilisation seront par la suite expliqués en détail. (voir la figure 2.2) :

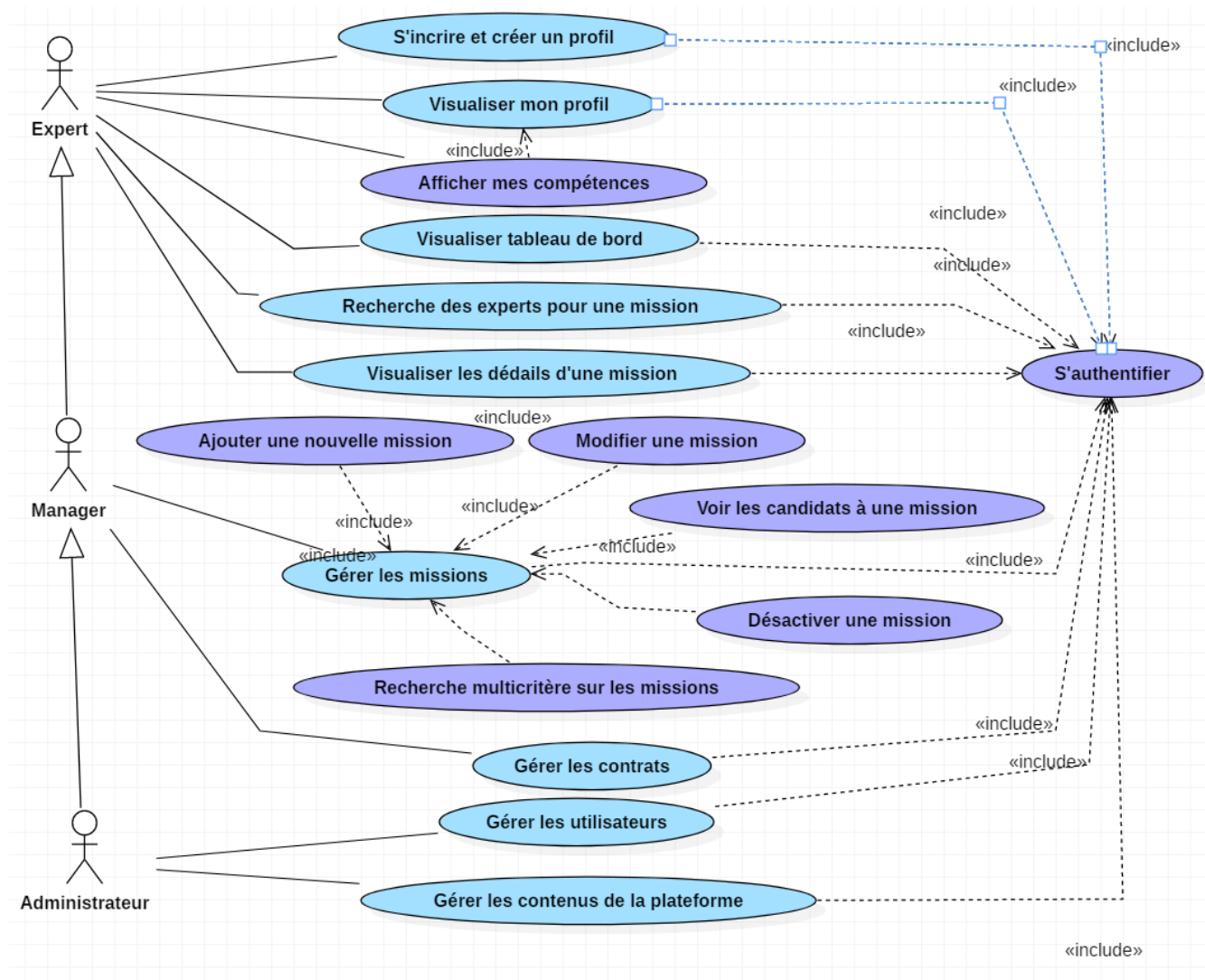


FIGURE 2.2 – Diagramme de cas d'utilisation général

2.3.2 Diagramme de cas d'utilisation pour Expert

Description textuelle du Diagramme de cas d'utilisation pour Expert :

- **Cas d'utilisation : Gérer mon profil :**
 - **Objectif :**
 - Permettre à l'Expert de mettre à jour ses informations personnelles sur la plateforme.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié sur la plateforme.
 - **Post-conditions :**
 - Les informations personnelles de l'Expert sont mises à jour dans le système.
L'Expert est informé de la réussite ou de l'échec de la mise à jour.
 - **Scénario nominal :**
 - Le système affiche les informations actuelles du profil de l'Expert.
 - L'Expert modifie ses informations personnelles (par exemple, nom, email, etc.)
 - Le système vérifie la validité des informations modifiées.
 - Le système enregistre les modifications valides dans la base de données.
 - Le système informe l'Expert que la mise à jour a été effectuée avec succès.
 - **Scénario alternatif :**
 - Erreur de mise à jour : Si certaines informations sont invalides (par exemple, email incorrect), le système informe l'Expert de l'erreur, et l'Expert doit corriger les informations avant de pouvoir soumettre à nouveau.
- **Cas d'utilisation : Gérer mes compétences :**
 - **Objectif :**
 - Permettre à l'Expert d'ajouter, modifier ou supprimer ses compétences sur la plateforme.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié sur la plateforme.
 - **Post-conditions :**
 - Les compétences de l'Expert sont mises à jour.
 - **Scénario nominal :**
 - L'Expert accède à la section des compétences.

- L'Expert ajoute, modifie ou désactive une compétence.
- Le système vérifie les modifications.
- Le Le système enregistre les modifications.
- Le système confirme la mise à jour.
- **Scénario alternatif :**
 - Si une compétence est invalide, le système informe l'Expert et lui demande de corriger l'erreur.
 - L'Expert corrige les informations et réessaie.
- **Cas d'utilisation :Recherche multicritère des compétences :**
 - **Objectif :**
 - Rechercher des compétences en utilisant des filtres avancés
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - La liste des compétences filtrées est affichée.
 - **Scénario nominal :**
 - L'Expert accède à l'outil de recherche multicritère des compétences.
 - L'Expert définit les critères de recherche.
 - Le système affiche la liste des compétences correspondant aux critères.
 - L'Expert consulte les résultats.
 - Le système confirme la mise à jour.
 - **Scénario alternatif :**
 - Si aucun résultat n'est trouvé, le système informe l'Expert et propose de modifier les critères de recherche.
- **Cas d'utilisation :Recherche multicritère des compétences :**
 - **Objectif :**
 - Rechercher des compétences en utilisant des filtres avancés
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - La liste des compétences filtrées est affichée.
 - **Scénario nominal :**
 - L'Expert accède à l'outil de recherche multicritère des compétences.

- L'Expert définit les critères de recherche.
- Le système affiche la liste des compétences correspondant aux critères.
- L'Expert consulte les résultats.
- Le système confirme la mise à jour.
- **Scénario alternatif :**
 - Si aucun résultat n'est trouvé, le système informe l'Expert et propose de modifier les critères de recherche.
- **Cas d'utilisation :Envoyer ma candidature pour une mission :**
 - **Objectif :**
 - Postuler pour une mission disponible
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié et avoir les compétences nécessaires.
 - **Post-conditions :**
 - La candidature de l'Expert est soumise.
 - **Scénario nominal :**
 - L'Expert consulte les missions disponibles.
 - L'Expert sélectionne une mission et envoie sa candidature.
 - Le système vérifie la conformité de la candidature.
 - La candidature est soumise avec succès.
 - Le système confirme la soumission.
 - **Scénario alternatif :**
 - Si la candidature est incomplète, le système informe l'Expert et lui demande de compléter les informations manquantes.
 - L'Expert soumet à nouveau la candidature.
- **Cas d'utilisation :Visualiser les détails d'une mission :**
 - **Objectif :**
 - Consulter les informations détaillées d'une mission.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - Les détails de la mission sont affichés à l'Expert.
 - **Scénario nominal :**
 - L'Expert sélectionne une mission à partir de la liste des missions disponibles.

- Le système affiche les détails de la mission.
- L'Expert consulte les informations.
- **Scénario alternatif :**
 - Si la mission est non disponible, le système informe l'Expert et lui propose de revenir à la liste des missions.
- **Cas d'utilisation :Recherche multicritère sur les missions :**
 - **Objectif :**
 - Rechercher des missions en utilisant des filtres avancés.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - La liste des missions filtrées est affichée.
 - **Scénario nominal :**
 - L'Expert accède à l'outil de recherche multicritère sur les missions.
 - L'Expert définit les critères de recherche.
 - Le système affiche la liste des missions correspondant aux critères.
 - L'Expert consulte les résultats.
 - **Scénario alternatif :**
 - Si aucun résultat n'est trouvé, le système informe l'Expert et propose de modifier les critères de recherche.
- **Cas d'utilisation :Visualiser mes missions passées :**
 - **Objectif :**
 - Consulter l'historique des missions déjà réalisées par l'Expert.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - L'historique des missions passées est affiché.
 - **Scénario nominal :**
 - L'Expert accède à la section des missions passées.
 - Le système affiche l'historique des missions terminées.
 - L'Expert consulte les missions.
 - **Scénario alternatif :**

- Si aucune mission passée n'est trouvée, le système informe l'Expert.
- **Cas d'utilisation :Visualiser mes missions en cours :**
 - **Objectif :**
 - Consulter les missions que l'Expert est en train de réaliser.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - La liste des missions en cours est affichée.
 - **Scénario nominal :**
 - L'Expert accède à la section des missions en cours.
 - Le système affiche la liste des missions en cours.
 - L'Expert consulte les missions.
 - **Scénario alternatif :**
 - Si aucun résultat n'est trouvé, le système informe l'Expert et propose Si aucune mission en cours n'est trouvée, le système informe l'Expert.
- **Cas d'utilisation :Valider un contrat :**
 - **Objectif :**
 - Valider un contrat proposé pour une mission.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - Le contrat est validé par l'Expert.
 - **Scénario nominal :**
 - L'Expert consulte les contrats disponibles.
 - L'Expert valide un contrat pour une mission.
 - Le système enregistre la validation du contrat.
 - Le système confirme la validation.
 - **Scénario alternatif :**
 - Si le contrat est incomplet ou invalide, le système informe l'Expert et lui demande de réviser le contrat.
- **Cas d'utilisation :Visualiser mes contrats :**
 - **Objectif :**

- Consulter tous les contrats associés aux missions de l'Expert.
- **Acteur :**
 - Expert
- **Pré-condition :**
 - L'Expert doit être authentifié.
- **Post-conditions :**
 - La liste des contrats est affichée.
- **Scénario nominal :**
 - L'Expert accède à la section des contrats.
 - Le système affiche la liste des contrats.
 - L'Expert consulte les contrats.
- **Scénario alternatif :**
 - Si aucun contrat n'est trouvé, le système informe l'Expert.
- **Cas d'utilisation :Suivre mes activités dans mes contrats :**
 - **Objectif :**
 - Permettre à l'Expert de suivre l'état et les activités liées à ses contrats.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - Les activités et l'état des contrats sont affichés.
 - **Scénario nominal :**
 - Expert accède à la section de suivi des contrats.
 - Le système affiche l'état des activités et des contrats.
 - Expert consulte les informations.
 - **Scénario alternatif :**
 - Si aucune activité n'est disponible, le système informe l'Expert.
- **Cas d'utilisation :Envoyer et recevoir des messages :**
 - **Objectif :**
 - Permettre à l'Expert de communiquer avec d'autres utilisateurs via des messages.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.

- **Post-conditions :**
 - Les messages sont envoyés ou reçus avec succès.
- **Scénario nominal :**
 - L'Expert accède à la messagerie.
 - L'Expert envoie ou consulte des messages.
 - Le système enregistre les messages envoyés et affiche les messages reçus.
- **Scénario alternatif :**
 - Si l'envoi du message échoue, le système informe l'Expert.
- **Cas d'utilisation :Tableau de bord :**
 - **Objectif :**
 - Fournir à l'Expert un résumé de ses missions, compétences, contrats et activités sur la plateforme.
 - **Acteur :**
 - Expert
 - **Pré-condition :**
 - L'Expert doit être authentifié.
 - **Post-conditions :**
 - Un tableau de bord récapitulatif est affiché.
 - **Scénario nominal :**
 - L'Expert accède à son tableau de bord.
 - Le système affiche un récapitulatif de ses missions, compétences, contrats, et activités.
 - L'Expert consulte le tableau de bord.
 - **Scénario alternatif :**
 - Si le tableau de bord ne peut pas être généré, le système informe l'Expert du problème.

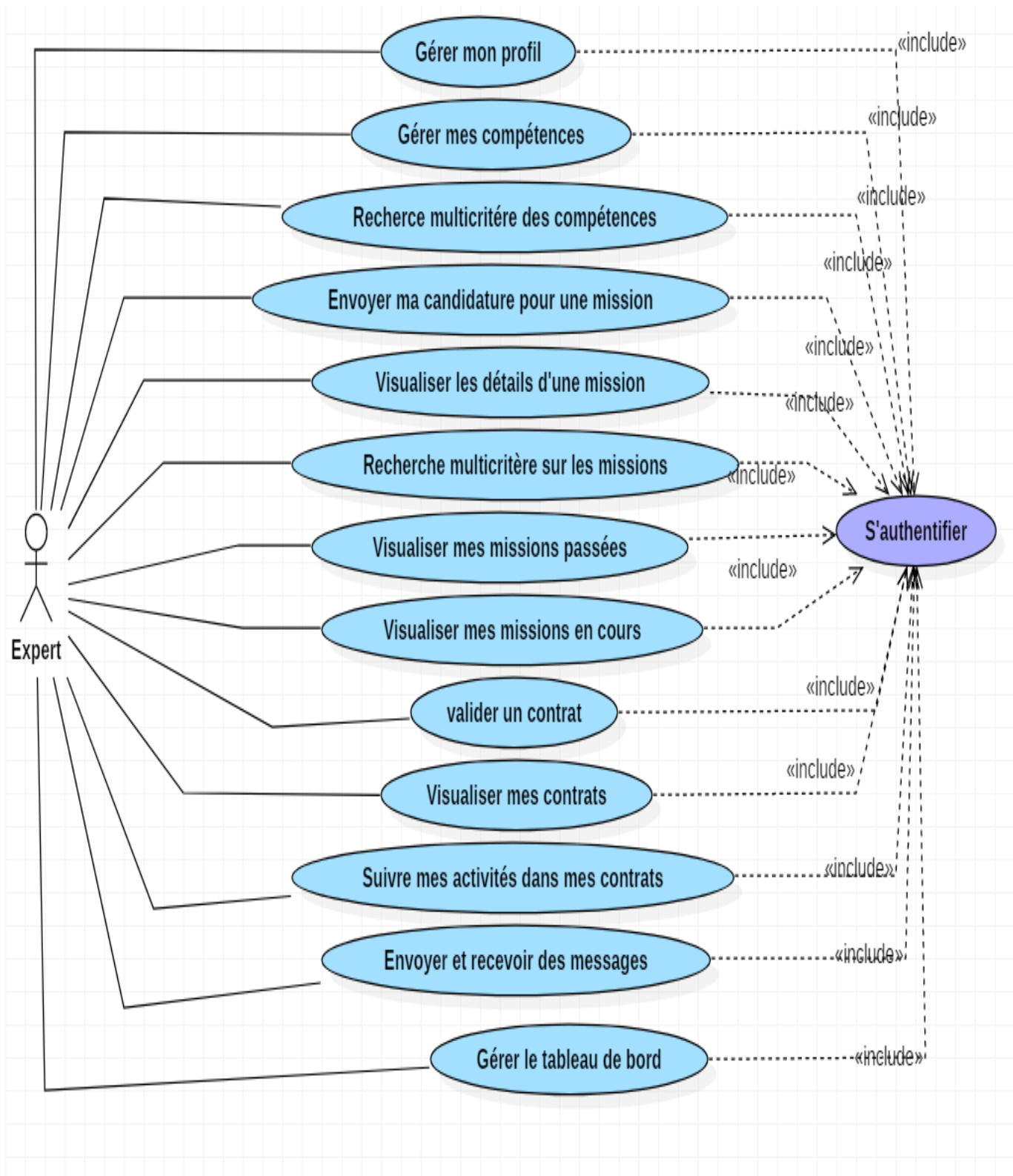


FIGURE 2.3 – Diagramme de cas d'utilisation pour Expert

2.3.3 Diagramme de cas d'utilisation pour Manager

Description textuelle du Diagramme de cas d'utilisation pour Manager :

- **Cas d'utilisation : Gestion mission :**
 - **Objectif :**
 - Gérer les missions, y compris la création, la modification et la désactivation.
 - **Acteur :**
 - Manager
 - **Pré-condition :**
 - Le Manager doit être authentifié.
 - **Post-conditions :**
 - Les missions sont correctement gérées (créées, modifiées, ou désactivées).
 - **Scénario nominal :**
 - Le Manager accède à la section de gestion des missions.
 - Le Manager crée, modifie ou désactive une mission.
 - Le système valide les modifications.
 - Le système enregistre les changements.
 - **Scénario alternatif :**
 - Si les modifications sont invalides, le système informe le Manager et demande des corrections.
- **Cas d'utilisation : Gestion Contrat :**
 - **Objectif :**
 - Gérer les contrats entre les experts et la société, y compris la création, la modification, et la résiliation des contrats.
 - **Acteur :**
 - Manager
 - **Pré-condition :**
 - Le Manager doit être authentifié.
 - **Post-conditions :**
 - Les contrats sont correctement gérés (créés, modifiés ou résiliés).
 - **Scénario nominal :**
 - Le Manager accède à la section de gestion des contrats.
 - Le Manager crée, modifie ou résilie un contrat.
 - Le système valide les modifications.
 - Le système enregistre les changements.
 - **Scénario alternatif :**
 - Si les informations sont invalides, le système informe le Manager et demande

des corrections.

- **Cas d'utilisation : Gestion Compétence :**

- **Objectif :**

- Gérer les compétences des experts, y compris leur ajout, modification ou suppression.

- **Acteur :**

- Manager

- **Pré-condition :**

- Le Manager doit être authentifié.

- **Post-conditions :**

- Les compétences sont correctement gérées.

- **Scénario nominal :**

- Le Manager accède à la section de gestion des compétences.
- Le Manager ajoute, modifie ou supprime une compétence.
- Le système valide les modifications.
- Le système enregistre les changements.

- **Scénario alternatif :**

- Si les informations sont invalides, le système informe le Manager et demande des corrections.

- **Cas d'utilisation : Gestion Messagerie Interne :**

- **Objectif :**

- Permettre au Manager d'envoyer et de recevoir des messages internes sur la plateforme.

- **Acteur :**

- Manager

- **Pré-condition :**

- Le Manager doit être authentifié.

- **Post-conditions :**

- Les messages sont envoyés et reçus avec succès.

- **Scénario nominal :**

- Le Manager accède à la messagerie interne.
- Le Manager envoie ou consulte des messages.
- Le système enregistre les messages envoyés et affiche les messages reçus.

- **Scénario alternatif :**

- Si l'envoi du message échoue, le système informe le Manager.

- **Cas d'utilisation : Tableau de Bord :**

- **Objectif :**
 - Offrir un résumé des missions, contrats, et activités gérés par le Manager.
- **Acteur :**
 - Manager
- **Pré-condition :**
 - Le Manager doit être authentifié.
- **Post-conditions :**
 - Un tableau de bord récapitulatif est affiché.
- **Scénario nominal :**
 - Le Manager accède à son tableau de bord.
 - Le système affiche un récapitulatif des missions, contrats, et activités en cours.
 - Le Manager consulte le tableau de bord.
- **Scénario alternatif :**
 - Si le tableau de bord ne peut pas être généré, le système informe le Manager du problème.

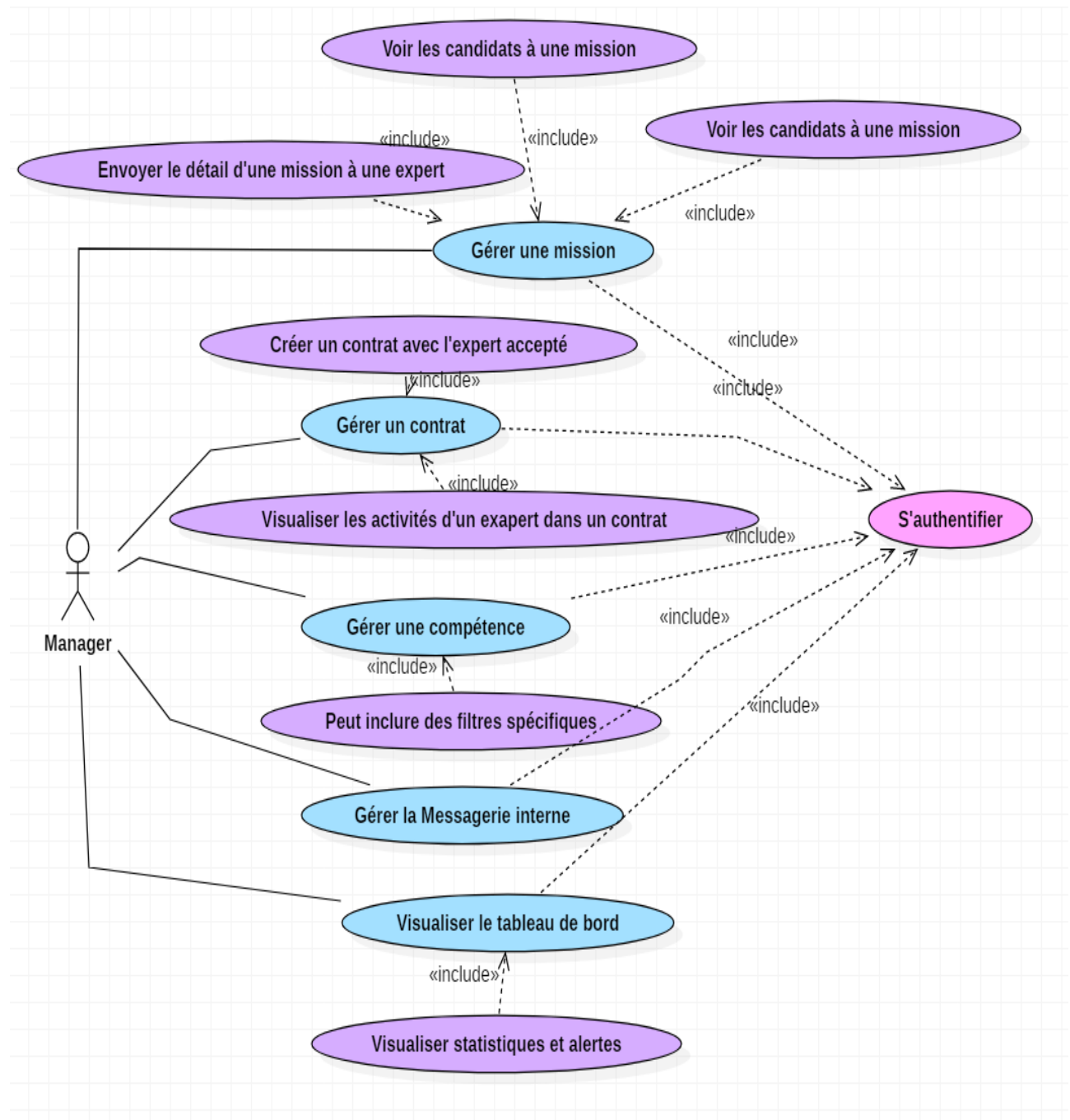


FIGURE 2.4 – Diagramme de cas d'utilisation pour Manager

2.3.4 Diagramme de cas d'utilisation pour Admin

Description textuelle du Diagramme de cas d'utilisation pour Admin :

- **Cas d'utilisation :Gérer les utilisateurs :**
 - **Objectif :**
 - Permettre à l'Administrateur de gérer les utilisateurs de la plateforme (ajouter, modifier, supprimer des utilisateurs).
 - **Acteur :**
 - Administrateur
 - **Pré-condition :**
 - L'Administrateur doit être authentifié.
 - **Post-conditions :**
 - Les informations des utilisateurs sont correctement mises à jour dans le système.
 - **Scénario nominal :**
 - L'Administrateur accède à la gestion des utilisateurs.
 - L'Administrateur ajoute, modifie ou supprime un utilisateur.
 - Le système valide les modifications.
 - Le système enregistre les changements.
 - Le système confirme que les informations des utilisateurs ont été mises à jour avec succès.
 - **Scénario alternatif :**
 - Si les informations sont invalides, le système informe l'Administrateur et demande des corrections.
- **Cas d'utilisation :Visualiser les contrats validés pendant une période :**
 - **Objectif :**
 - Permettre à l'Administrateur de visualiser tous les contrats validés pendant une période spécifiée.
 - **Acteur :**
 - Administrateur
 - **Pré-condition :**
 - L'Administrateur doit être authentifié.
 - **Post-conditions :**
 - Les contrats validés pendant la période spécifiée sont affichés
 - **Scénario nominal :**
 - L'Administrateur accède à la section de visualisation des contrats.

- L'Administrateur spécifie une période.
- Le système affiche les contrats validés pendant cette période.
- L'Administrateur consulte les contrats.
- **Scénario alternatif :**
 - Si aucun contrat n'est trouvé pour la période spécifiée, le système informe l'Administrateur.
- **Cas d'utilisation : Visualiser les contrats d'un expert :**
 - **Objectif :**
 - Permettre à l'Administrateur de consulter tous les contrats associés à un expert spécifique.
 - **Acteur :**
 - Administrateur
 - **Pré-condition :**
 - L'Administrateur doit être authentifié.
 - **Post-conditions :**
 - La liste des contrats associés à l'expert est affichée.
 - **Scénario nominal :**
 - L'Administrateur sélectionne un expert.
 - Le système affiche les contrats associés à cet expert.
 - L'Administrateur consulte les contrats.
 - **Scénario alternatif :**
 - Si aucun contrat n'est associé à l'expert sélectionné, le système informe l'Administrateur.
- **Cas d'utilisation : Visualiser les contrats d'une société :**
 - **Objectif :**
 - Permettre à l'Administrateur de consulter tous les contrats associés à une société spécifique.
 - **Acteur :**
 - Administrateur
 - **Pré-condition :**
 - L'Administrateur doit être authentifié.
 - **Post-conditions :**
 - La liste des contrats associés à la société est affichée.
 - **Scénario nominal :**
 - L'Administrateur sélectionne une société.
 - Le système affiche les contrats associés à cette société.

- L'Administrateur consulte les contrats.
- **Scénario alternatif :**
 - Si aucun contrat n'est associé à la société sélectionnée, le système informe l'Administrateur.
- **Cas d'utilisation : Gestion Messagerie Interne :**
 - **Objectif :**
 - Permettre à l'Administrateur d'envoyer et de recevoir des messages internes sur la plateforme.
 - **Acteur :**
 - Administrateur
 - **Pré-condition :**
 - L'Administrateur doit être authentifié.
 - **Post-conditions :**
 - Les messages sont envoyés et reçus avec succès.
 - **Scénario nominal :**
 - L'Administrateur accède à la messagerie interne.
 - L'Administrateur envoie ou consulte des messages.
 - Le système enregistre les messages envoyés et affiche les messages reçus.
 - **Scénario alternatif :**
 - Si l'envoi du message échoue, le système informe l'Administrateur.
- **Cas d'utilisation : Tableau de Bord :**
 - **Objectif :**
 - Offrir un résumé des contrats, utilisateurs, statistiques, et activités gérés par l'Administrateur.
 - **Acteur :**
 - Administrateur
 - **Pré-condition :**
 - L'Administrateur doit être authentifié.
 - **Post-conditions :**
 - Un tableau de bord récapitulatif est affiché.
 - **Scénario nominal :**
 - L'Administrateur accède à son tableau de bord.
 - Le système affiche un récapitulatif des contrats, utilisateurs, statistiques, et activités en cours.
 - L'Administrateur consulte le tableau de bord.
 - **Scénario alternatif :**

- Si le tableau de bord ne peut pas être généré, le système informe l'Administrateur du problème.

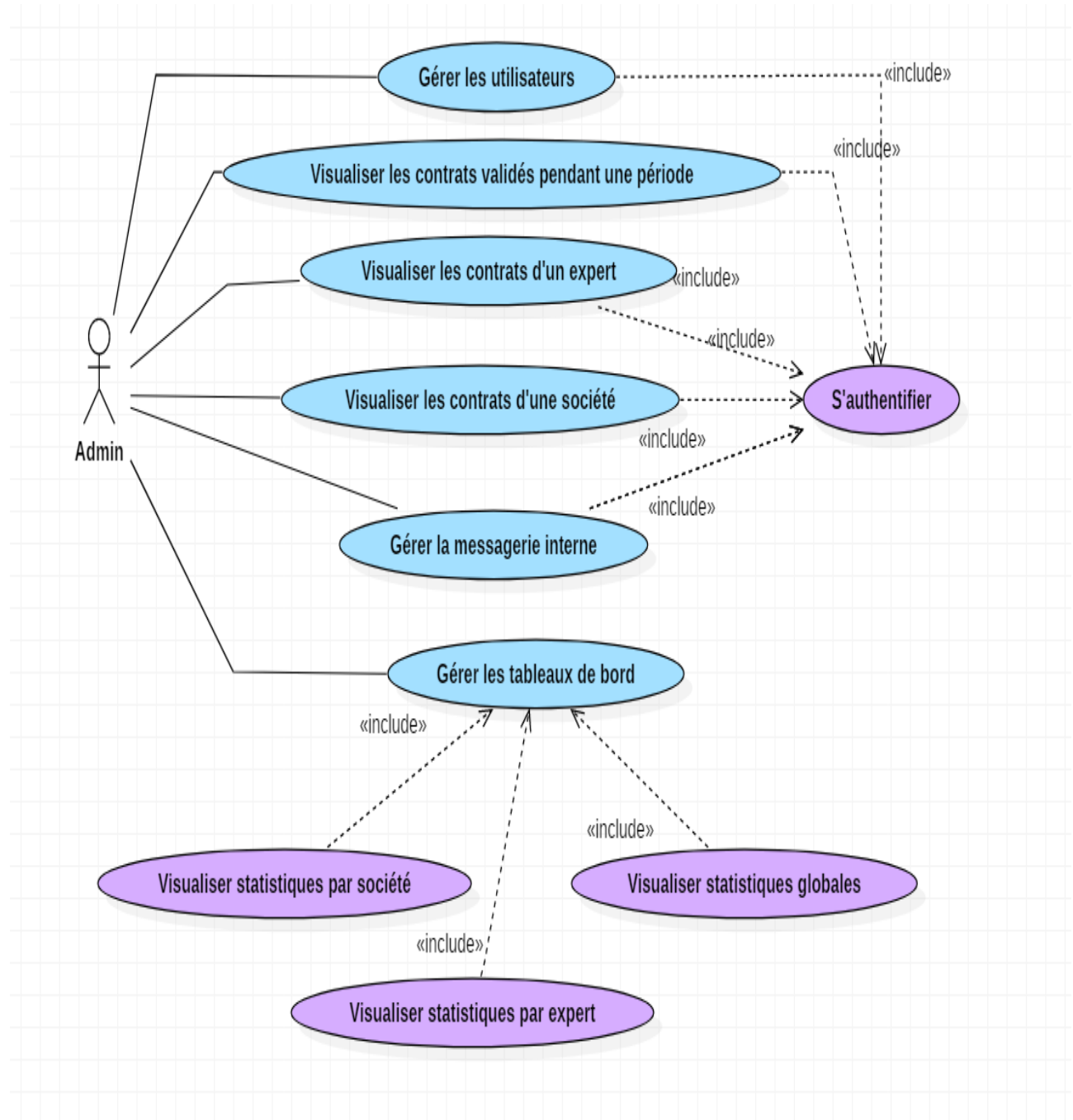


FIGURE 2.5 – Diagramme de cas d'utilisation pour Admin

2.4 Diagramme de Séquence

Le diagramme de séquence montre l'interaction entre les objets dans un scénario spécifique au fil du temps. Il est essentiel pour comprendre la dynamique des opérations du système.

2.4.1 Diagramme de Séquence : Visualiser mon profil (Gestion de données)

Ce diagramme de séquence montre le processus de visualisation du profil utilisateur pour un expert.

- Expert : L'expert demande la visualisation de son profil via l'interface.
- Interface : L'interface transmet cette demande au Service GestionExpert.
- Service GestionExpert : Ce service envoie une requête pour récupérer les informations de profil de l'utilisateur à l'API des profils.
- API Profils : L'API lit les données de profil et les renvoie au Service GestionExpert.
- Service GestionExpert : Le service transfère ensuite les données de profil à l'interface.
- Interface : Les informations de profil sont retournées à l'expert.
- Expert : Le profil utilisateur est affiché pour l'expert.

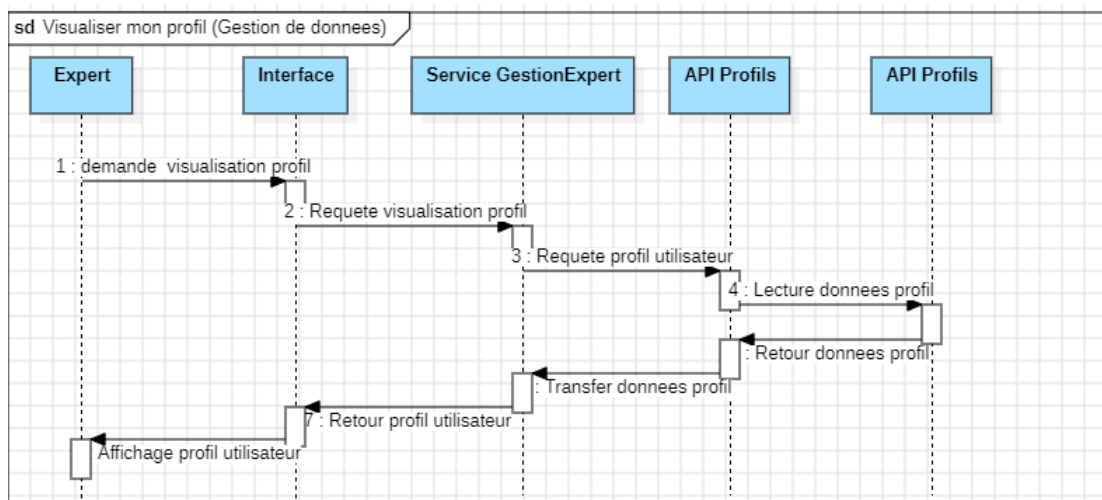


FIGURE 2.6 – Diagramme de Séquence : Visualiser mon profil (Gestion de données)

2.4.2 Diagramme de Séquence : Modifier mes informations personnelles (Gestion expert)

Ce diagramme de séquence décrit le processus de modification des informations personnelles pour un expert.

- Expert : L'expert initie la modification de ses informations personnelles via l'interface.
- Interface : La demande de modification est transmise au Service Gestion Expert.
- Service Gestion Expert : Le service met à jour le profil de l'utilisateur et envoie les nouvelles informations à l'API Profils.
- API Profils : L'API enregistre les nouvelles données dans la base de données.
- Base de données : Confirme la mise à jour réussie des données.
- API Profils : Retourne la confirmation de la mise à jour au Service Gestion Expert.
- Service Gestion Expert : Informe l'interface du succès de la modification.
- Interface : Retourne la confirmation de la modification du profil à l'expert.

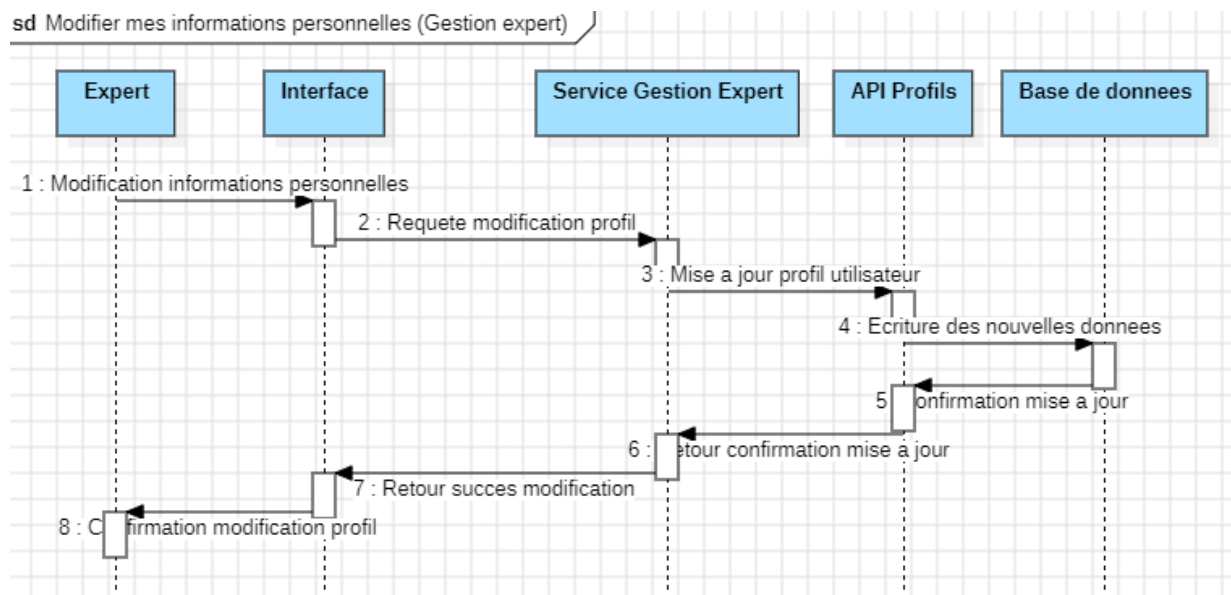


FIGURE 2.7 – Diagramme de Séquence : Modifier mes informations personnelles (Gestion expert)

2.4.3 Diagramme de Séquence : Ajouter une compétence (Gestion Compétence)

Ce diagramme de séquence décrit le processus d'ajout d'une nouvelle compétence pour un expert.

- Expert : L'expert ajoute une nouvelle compétence via l'interface utilisateur.
- Interface : La demande d'ajout de compétence est envoyée au Service Gestion Expert.
- Service Gestion Expert : Le service crée la nouvelle compétence et envoie les informations à l'API Profils.
- API Profils : L'API insère la nouvelle compétence dans la base de données.
- Base de données : Confirme l'insertion réussie de la nouvelle compétence.
- API Profils : Retourne la confirmation de l'ajout au Service Gestion Expert.
- Service Gestion Expert : Informe l'interface que l'ajout de compétence a réussi.
- Interface : Retourne la confirmation de l'ajout de la compétence à l'expert.

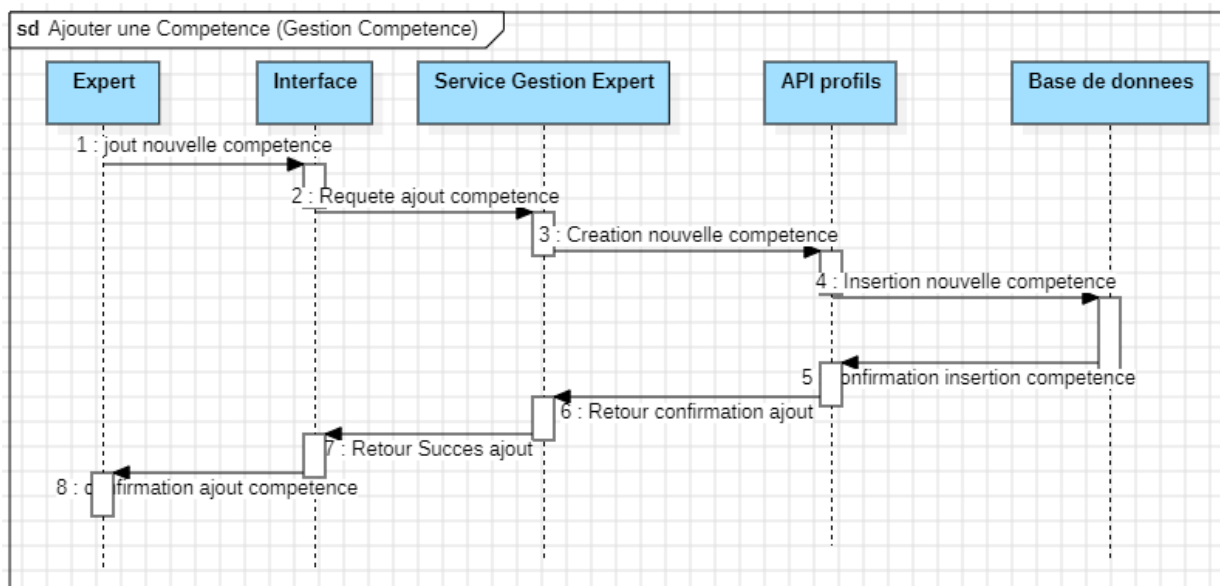


FIGURE 2.8 – Diagramme de Séquence : Ajouter une compétence (Gestion Compétence)

2.4.4 Diagramme de Séquence : Modifier une compétence (Gestion Compétence)

Ce diagramme de séquence décrit le processus de modification d'une compétence existante pour un expert.

- Expert : L'expert modifie une compétence existante via l'interface utilisateur.
- Interface : La demande de modification de compétence est envoyée au Service Gestion Expert.
- Service Gestion Expert : Le service met à jour la compétence et envoie les informations modifiées à l'API Profils.
- API Profils : L'API enregistre les nouvelles informations de la compétence dans la base de données.
- Base de données : Confirme la mise à jour réussie de la compétence.
- API Profils : Retourne la confirmation de la modification au Service Gestion Expert.
- Service Gestion Expert : Informe l'interface que la modification de la compétence a réussi.
- Interface : Retourne la confirmation de la modification de la compétence à l'expert.

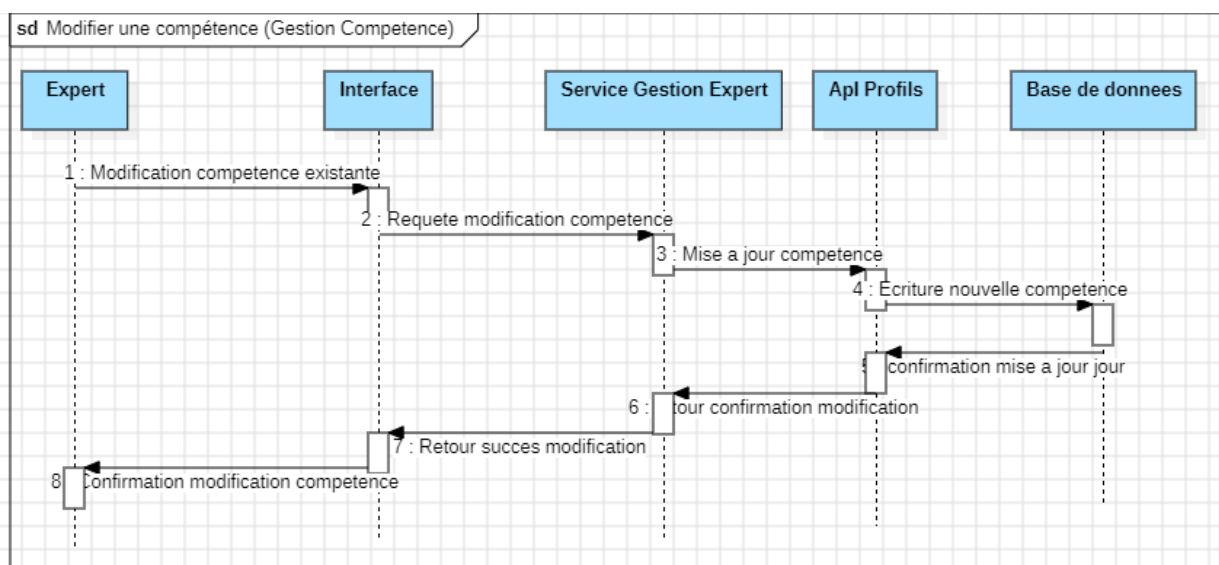


FIGURE 2.9 – Diagramme de Séquence : Modifier une compétence (Gestion Compétence)

2.4.5 Diagramme de Séquence : Désactiver une compétence (Gestion Compétence)

Ce diagramme de séquence décrit le processus de désactivation d'une compétence pour un expert.

- Expert : L'expert désactive une compétence via l'interface utilisateur.
- Interface : La demande de désactivation de la compétence est envoyée au Service Gestion Compétence.
- Service Gestion Compétence : Le service met à jour l'état de la compétence pour la désactiver et envoie cette information à l'API Compétence.
- API Compétence : L'API met à jour l'état de la compétence dans la base de données.
- Base de données : Confirme la désactivation réussie de la compétence.
- API Compétence : Retourne la confirmation de la désactivation au Service Gestion Compétence.
- Service Gestion Compétence : Informe l'interface que la désactivation de la compétence a réussi.
- Interface : Retourne la confirmation de la désactivation de la compétence à l'expert.

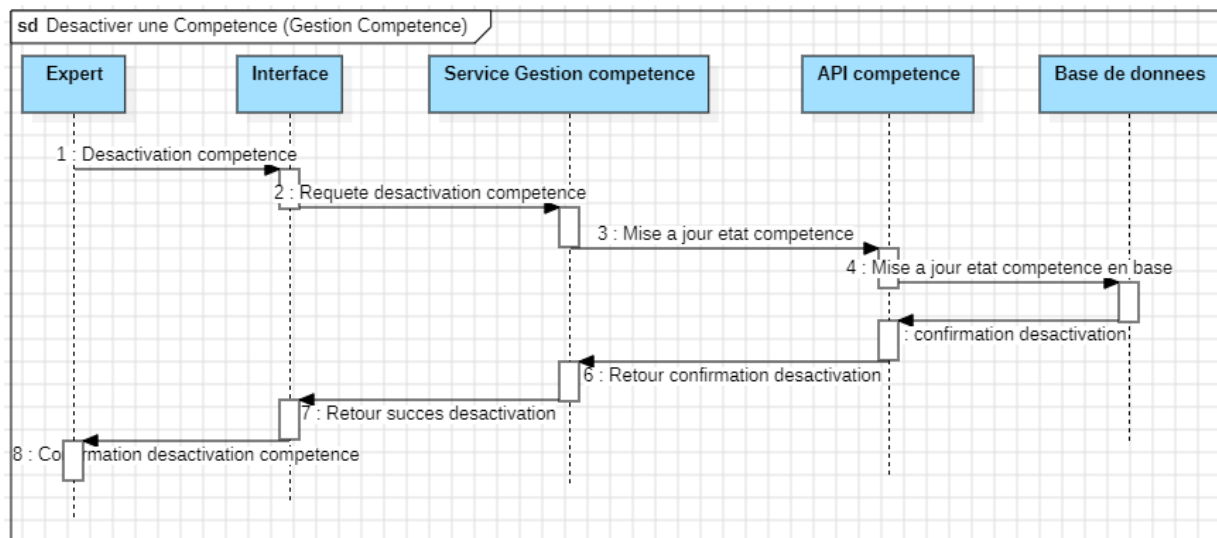


FIGURE 2.10 – Diagramme de Séquence : Désactiver une compétence (Gestion Compétence)

2.4.6 Diagramme de Séquence : Ajouter une Mission (Gestion Mission)

Ce diagramme de séquence décrit le processus d'ajout d'une nouvelle mission par un manager.

- Manager : Le manager ajoute une nouvelle mission via l'interface utilisateur.
- Interface : La demande d'ajout de mission est envoyée au Service Gestion Mission.
- Service Gestion Mission : Le service crée la nouvelle mission et envoie les informations à l'API Missions.
- API Missions : L'API insère la nouvelle mission dans la base de données.
- Base de Données : Confirme l'insertion réussie de la nouvelle mission.
- API Missions : Retourne la confirmation de l'ajout de la mission au Service Gestion Mission.
- Service Gestion Mission : Informe l'interface que l'ajout de la mission a réussi.
- Interface : Retourne la confirmation de l'ajout de la mission au manager.

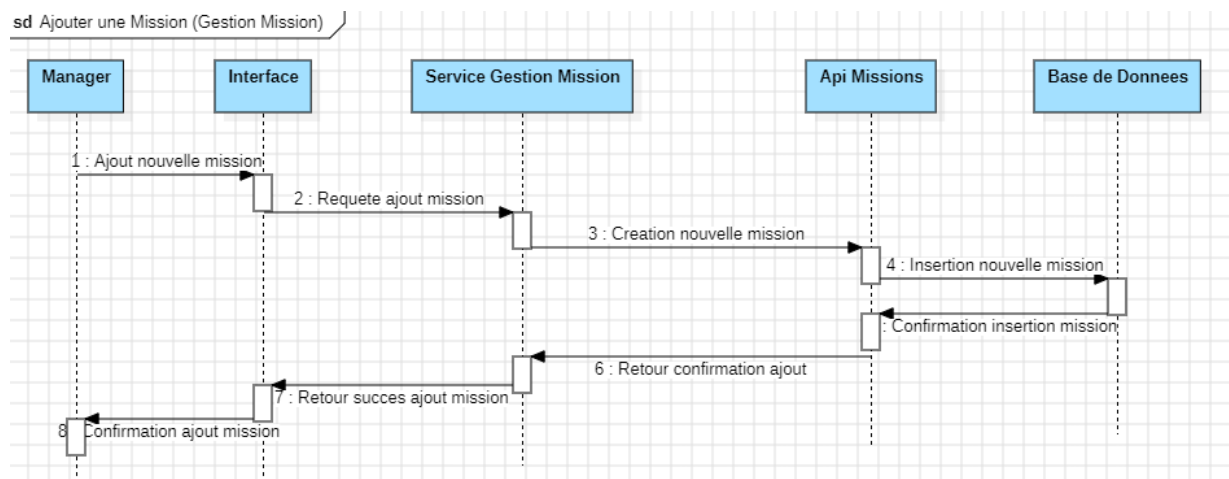


FIGURE 2.11 – Diagramme de Séquence : Ajouter une Mission (Gestion Mission)

2.4.7 Diagramme de Séquence : Modifier une Mission (Gestion Mission)

Ce diagramme de séquence décrit le processus de modification d'une mission existante par un manager.

- Manager : Le manager modifie une mission existante via l'interface utilisateur.
- Interface : La demande de modification de mission est envoyée au Service Gestion Mission.
- Service Gestion Mission : Le service met à jour la mission avec les nouvelles informations et les envoie à l'API Missions.
- API Missions : L'API enregistre les nouvelles informations de la mission dans la base de données.
- Base de Données : Confirme la mise à jour réussie de la mission.
- API Missions : Retourne la confirmation de la modification au Service Gestion Mission.
- Service Gestion Mission : Informe l'interface que la modification de la mission a réussi.
- Interface : Retourne la confirmation de la modification de la mission au manager.

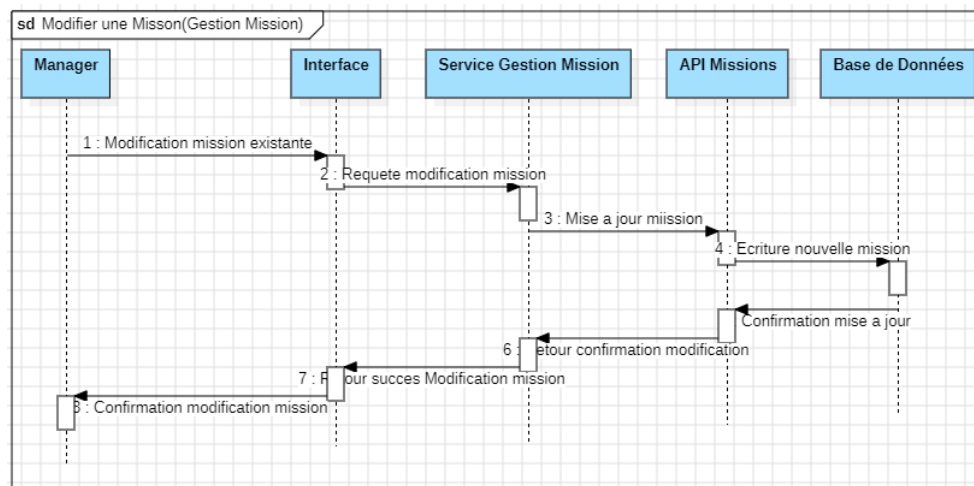


FIGURE 2.12 – Diagramme de Séquence : Modifier une Mission (Gestion Mission)

2.4.8 Diagramme de Séquence : Visualiser les Contrats Validés Pendant une Période (Admin)

Ce diagramme de séquence décrit le processus permettant à un administrateur de visualiser les contrats validés pendant une période spécifique.

- Admin : L'administrateur demande à visualiser les contrats validés pendant une période via l'interface utilisateur.
- Interface : La demande est envoyée au Service Gestion Contrat.
- Service Gestion Contrat : Le service recherche les contrats validés correspondant à la période spécifiée et envoie une requête à l'API Contrats.
- API Contrats : L'API récupère les contrats validés dans la base de données.
- Base de Données : Confirme et renvoie les contrats validés.
- API Contrats : Transfère les données des contrats au Service Gestion Contrat.
- Service Gestion Contrat : Retourne la liste des contrats validés à l'interface.
- Interface : Affiche les contrats validés à l'administrateur.

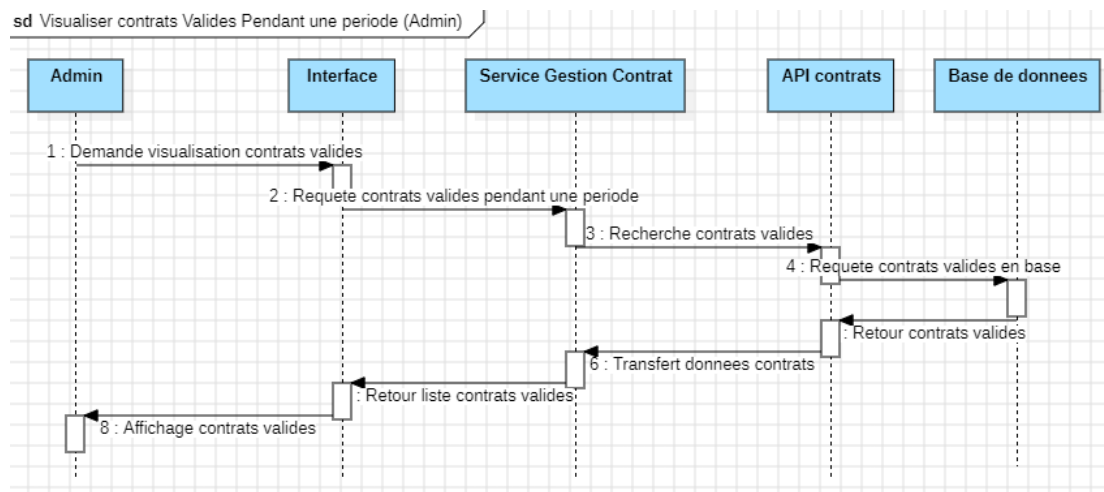


FIGURE 2.13 – Diagramme de Séquence : Visualiser les Contrats Validés Pendant une Période (Admin)

2.4.9 Diagramme de Séquence : Visualiser les Contrats d'une Société (Admin)

Ce diagramme de séquence décrit le processus permettant à un administrateur de visualiser les contrats associés à une société spécifique.

- Admin : L'administrateur demande à visualiser les contrats d'une société via l'interface utilisateur.
- Interface : La demande est envoyée au Service Gestion Contrat.
- Service Gestion Contrat : Le service recherche les contrats liés à la société spécifiée et envoie une requête à l'API Contrats.
- API Contrats : L'API récupère les contrats de la société dans la base de données.
- Base de Données : Confirme et renvoie les contrats associés à la société.
- API Contrats : Transfère les données des contrats au Service Gestion Contrat.
- Service Gestion Contrat : Retourne la liste des contrats à l'interface.
- Interface : Affiche les contrats associés à la société pour l'administrateur.

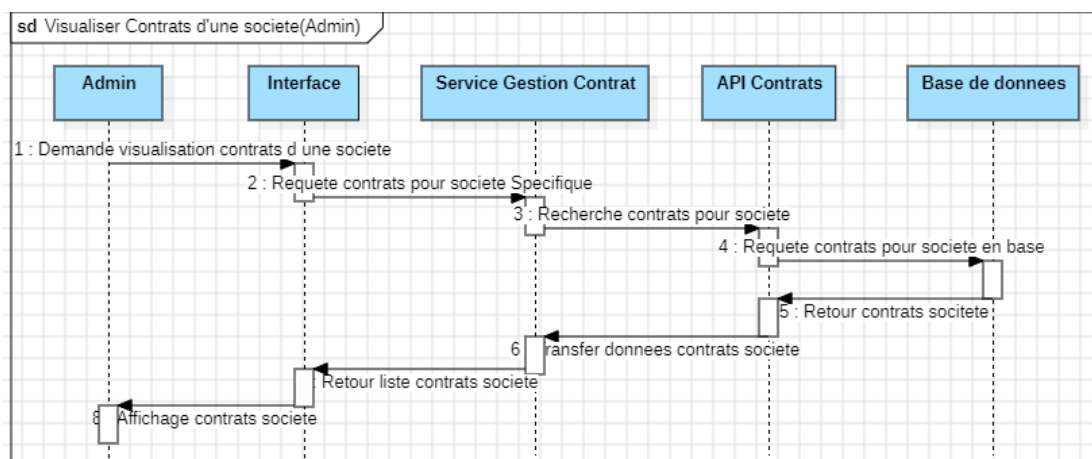


FIGURE 2.14 – Diagramme de Séquence : Visualiser les Contrats d'une Société (Admin)

2.4.10 Diagramme de Séquence : Visualiser les Contrats d'un Expert (Admin)

Ce diagramme de séquence décrit le processus permettant à un administrateur de visualiser les contrats associés à un expert spécifique.

- Admin : L'administrateur initie la demande de visualisation des contrats d'un expert via l'interface utilisateur.
- Interface : La requête pour récupérer les contrats de l'expert est envoyée au Service Gestion Contrat.
- Service Gestion Contrat : Le service recherche les contrats liés à l'expert spécifié et envoie une requête à l'API Contrats.
- API Contrats : L'API récupère les contrats de l'expert dans la base de données.
- Base de Données : Confirme et renvoie les contrats associés à l'expert.
- API Contrats : Transfère les données des contrats au Service Gestion Contrat.
- Service Gestion Contrat : Retourne la liste des contrats à l'interface.
- Interface : Affiche les contrats associés à l'expert pour l'administrateur.

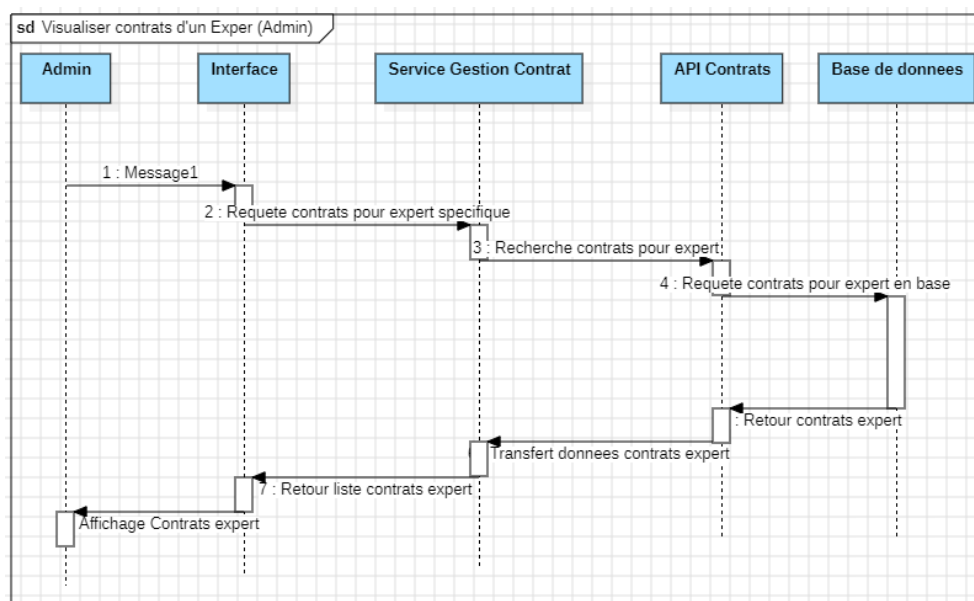


FIGURE 2.15 – Diagramme de Séquence : Visualiser les Contrats d'un Expert (Admin)

2.5 Diagramme d'Activités

Le diagramme d'activités montre le flux de travail dans le système, détaillant les différentes activités et les transitions entre elles.

2.5.1 Diagramme d'Activités : Visualiser Mon Profil (Gestion Expert)

Ce diagramme d'activité présent dans la figure 2.15 représente le processus de visualisation du profil d'un expert. Les étapes incluent l'affichage des informations personnelles, des compétences, des diplômes, des formations, et des missions réalisées. Chaque étape est exécutée séquentiellement pour permettre à l'expert de consulter tous les aspects de son profil.

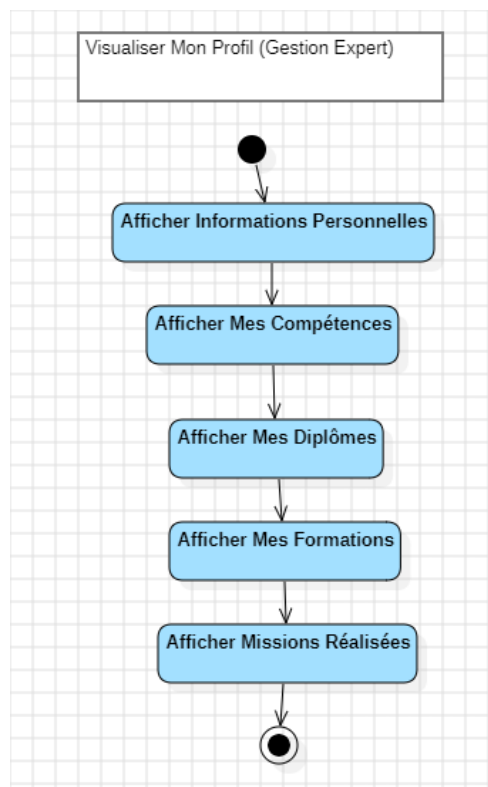


FIGURE 2.16 – Diagramme d'Activités : Visualiser Mon Profil (Gestion Expert)

2.5.2 Diagramme d'Activités : Modifier Mes Informations Personnelles (Gestion Expert)

Ce diagramme d'activité illustre le processus de modification des informations personnelles d'un expert. Les étapes comprennent l'affichage du formulaire de modification, la saisie des nouvelles informations, la validation des modifications, la mise à jour du profil dans la base de données, et la confirmation de la mise à jour réussie. Chaque étape est exécutée de manière séquentielle pour garantir que les nouvelles informations sont correctement enregistrées comme le montre dans la figure 2.17.

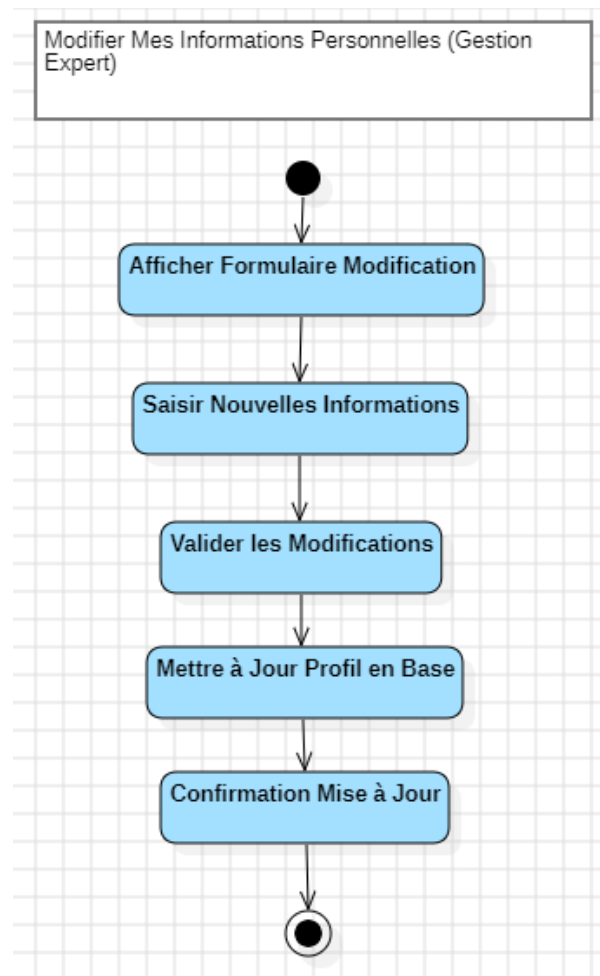


FIGURE 2.17 – Diagramme d'Activités : Modifier Mes Informations Personnelles (Gestion Expert)

2.5.3 Diagramme d'Activités : Ajouter Une Compétence (Gestion Compétence)

Ce diagramme d'activité présent dans la figure 2.18 décrit le processus d'ajout d'une nouvelle compétence pour un expert. Les étapes incluent l'affichage du formulaire d'ajout de compétence, la saisie de la nouvelle compétence, la validation de l'ajout, l'enregistrement de la compétence dans la base de données, et la confirmation de l'ajout réussi. Chaque étape est réalisée séquentiellement pour assurer que la nouvelle compétence est correctement ajoutée au système.

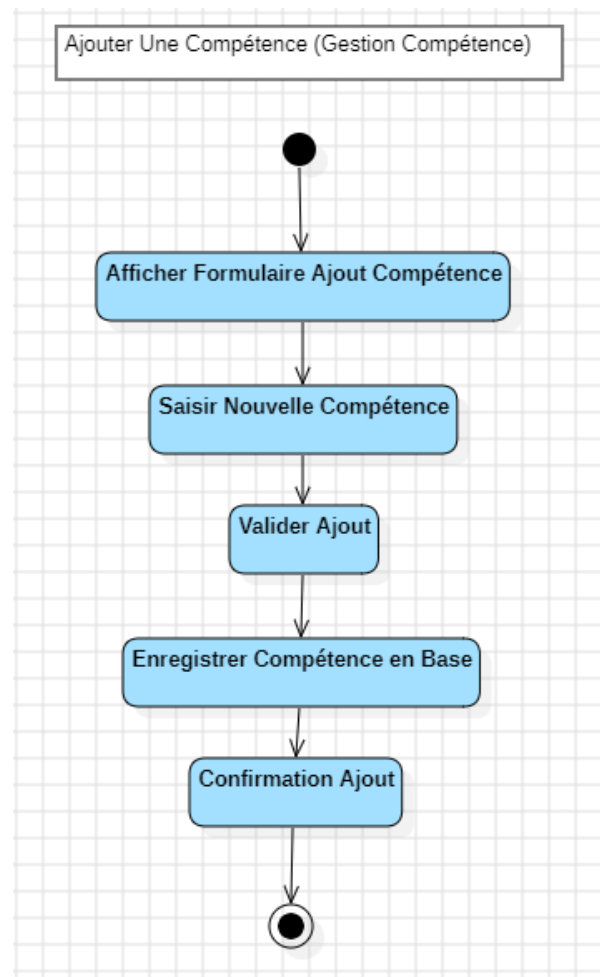


FIGURE 2.18 – Diagramme d'Activités : Ajouter Une Compétence (Gestion Compétence)

2.5.4 Diagramme d'Activités : Modifier Une Compétence (Gestion Compétence)

Ce diagramme d'activité présent dans la figure 2.19 montre le processus de modification d'une compétence existante pour un expert. Les étapes comprennent l'affichage des compétences existantes, la sélection de la compétence à modifier, la modification des détails de la compétence, la validation des modifications, la mise à jour de la compétence dans la base de données, et la confirmation de la modification réussie. Ce processus garantit que les modifications apportées à une compétence sont correctement mises à jour dans le système.

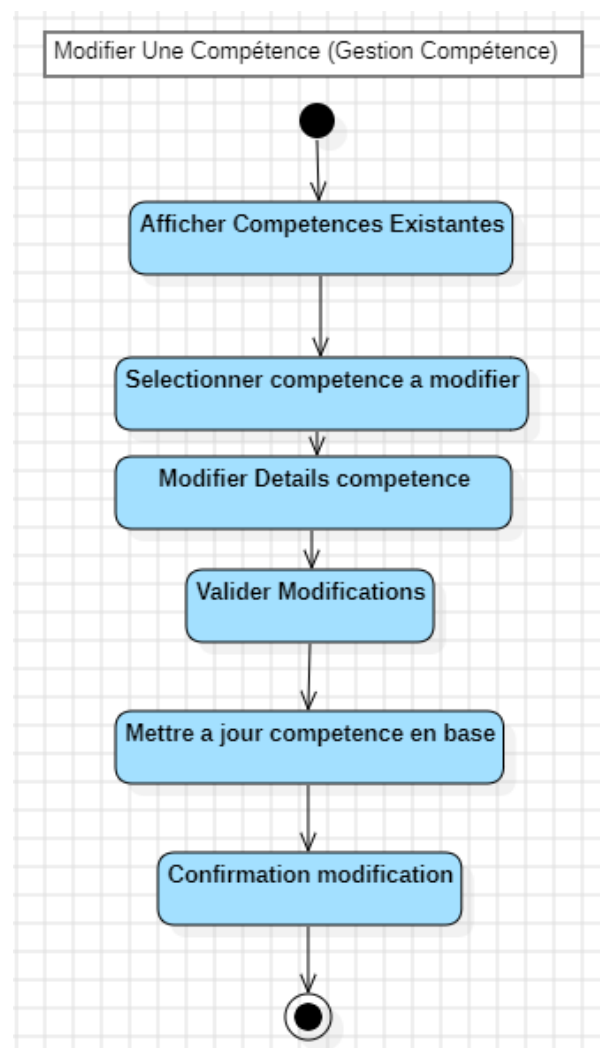


FIGURE 2.19 – Diagramme d'Activités : Modifier Une Compétence (Gestion Compétence)

2.6 Conclusion

Ce chapitre a présenté une vue d'ensemble détaillée de la conception du système de gestion des experts. Les diagrammes de classes, de séquence, de cas d'utilisation, d'architecture, et d'activités ont été utilisés pour illustrer les différentes facettes de la conception. Chaque diagramme et description contribue à une compréhension approfondie de la structure et du fonctionnement du système, facilitant ainsi le développement et la maintenance futurs.

Chapitre 3

Réalisation

3.1 Introduction

Dans ce chapitre, nous décrivons en détail la phase de réalisation du projet de gestion des experts en développement. Cette phase est cruciale car elle traduit les conceptions théoriques en implémentations concrètes. Nous discuterons des outils et technologies utilisés, des défis techniques rencontrés, des méthodes de résolution de problèmes, et des différents tests effectués pour assurer la qualité et la robustesse du système.

3.2 Création du logo

La réalisation du logo de l'application de gestion des experts a été conçue pour refléter à la fois la modernité et la simplicité. En utilisant des formes épurées et des couleurs professionnelles, le logo présent dans la figure 3.1 symbolise la compétence, l'efficacité, et l'expertise, caractéristiques essentielles à cette application.



FIGURE 3.1 – Logo Expert Consulting

3.3 Release N°1 : Gestion des Comptes Admin, Manager, Expert , et Inscription

Dans cette partie, nous abordons un aspect essentiel du développement de notre application : la réalisation du premier sprint, qui se concentre sur la gestion des comptes et des inscriptions.

3.3.1 Backlog du sprint 1

Dans cette partie, nous détaillerons ces fonctionnalités en présentant le Backlog dans le tableau 3.1. Ce Backlog fournira une vue détaillée des fonctionnalités développés, y compris les exigences spécifiques, les priorités, et les tâches associées à chaque fonctionnalité.

Sprint	User Story
Gestion du compte	Création du compte
	Connexion/Déconnexion
	Modification du compte
	Suppression du compte
	Activation/Désactivation du compte
Gestion de profil expert	Visualiser mes informations personnelles
	Visualiser mes compétences
	Visualiser mes diplômes
	Visualiser mes formations
	Visualiser les missions réalisées
Gestion de société	Ajouter une société
	Modifier une société
	Supprimer une société
	Activer/désactiver une société
	Consulter liste des comptes

TABLE 3.1 – Backlog du sprint 1

3.3.2 Réalisation

Dans cette section, nous mettrons en lumière plusieurs interfaces conçues pour le sprint 1. Ces interfaces ont été élaborées dans le but de répondre aux besoins spécifiques des

utilisateurs et de faciliter l'interaction avec le système.

La Figure 3.2 représente l'interface « Qui je suis » dans laquelle l'internaute doit tout d'abord renseigner s'il sera inscrit en tant que Manager ou un Expert.

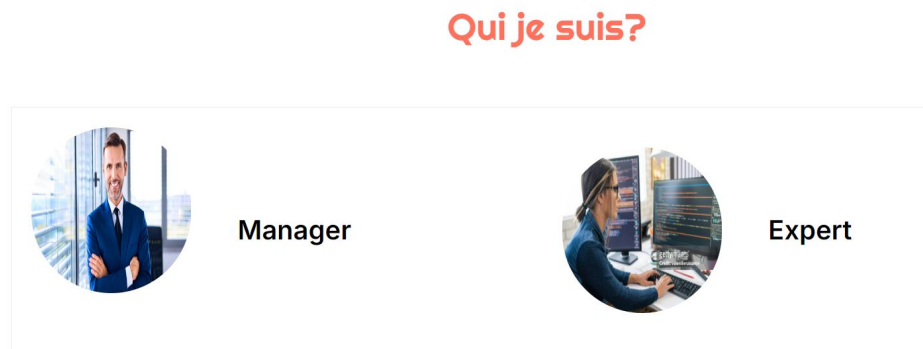


FIGURE 3.2 – Qui Je suis

Si l'internaute est un expert, il va être redirigé vers l'interface « Inscription Société » comme le montre la figure 3.3. Dans cette interface il doit remplir le formulaire pour créer son compte.

Sign up

Full Name

Email

phone

biography

Please enter your CIN number with 8 digits.

cv

Please enter

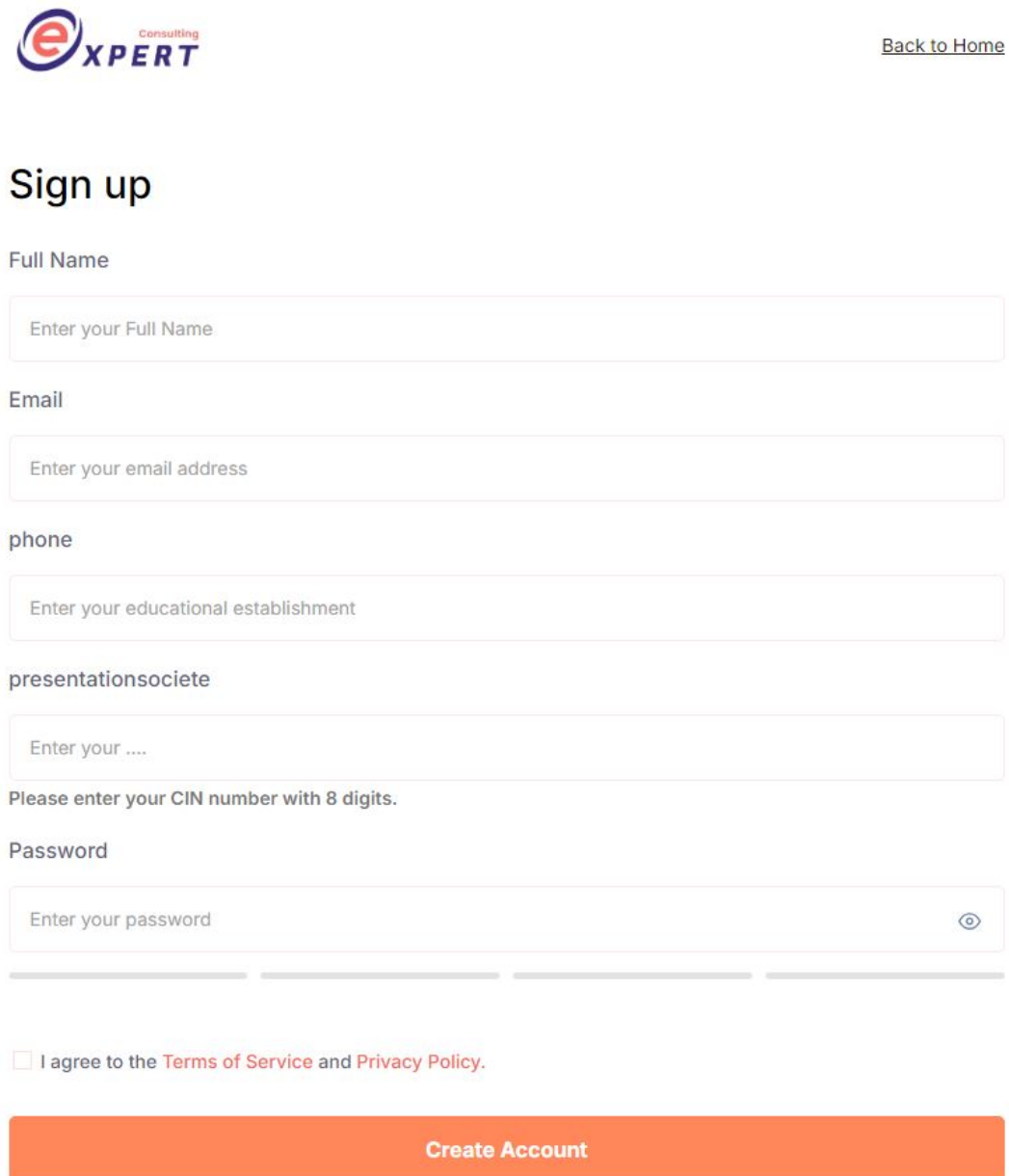
Password


☐ I agree to the [Terms of Service](#) and [Privacy Policy](#).

Create Account

FIGURE 3.3 – Inscription expert

Si l'internaute est un Manager, il va être redirigé vers l'interface « Inscription Société » comme le montre la figure 3.4. Dans cette interface il doit remplir le formulaire pour créer son compte.



 [Back to Home](#)

Sign up

Full Name


Email

phone

presentationsociete

Please enter your CIN number with 8 digits.

Password


 

☐ I agree to the [Terms of Service](#) and [Privacy Policy](#).

Create Account

FIGURE 3.4 – Inscription societe


A l'aide de la sécurité JWT, l'internaute reçoit un mail pour activer son compte à travers un token qui ne sera pas utilisable apres une periode de 60 secondes. Après avoir activer son compte, il peut s'authentifier à travers l'interface « Connexion » comme représente la figure 3.5

[Back to Home](#)

Sign into Your Account

Email

Password


[Forgot Password ?](#)

☐ Remember me


[Sign In](#)


FIGURE 3.5 – Connexion

Une fois on est authentifié en tant qu'expert on se trouve dans la page de détails de l'expert comme représenté dans la figure 3.6



Home ▾ Expert ▾ Pages ▾





Sofiene khannoussi

Expert

[Offer Request](#)

Menu

- [Edit Profile](#)
- [Security](#)
- [Social Profiles](#)
- [Delete Profile](#)

Personal Details

Edit your personal information and address.

Full Name	Email
<input type="text" value="Sofiene khannoussi"/>	<input type="text" value="expert@mail.com"/>
Biography	Adresse
<input type="text" value="Sofiene khannoussi"/>	<input type="text" value="adresse"/>
Phone	
<input type="text" value="198"/>	

[Update Profile](#)

FIGURE 3.6 – Données personnelles

3.4 Release N°2 : Gestion de compétences et gestion de CV

Dans cette seconde release, nous progressons vers une étape cruciale de notre projet en intégrant des fonctionnalités vitales pour notre plateforme. on va s'intéresser sur le profil expert ou il va renseigner tous ses données

3.4.1 Backlog du sprint 2

Dans cette section, nous explorerons en profondeur ces fonctionnalités en décrivant le Backlog associé. Ce Backlog offrira une vue détaillée des fonctionnalités à développer pour le sprint 2, incluant les exigences spécifiques, les priorités. Le Tableau 3.2 détaille les différentes user Stories du sprint 2.

Sprint	User Story
Gestion des compétences	En tant qu'expert : je peux ajouter une de mes compétences.
	En tant qu'expert : je peux modifier mes compétences.
	En tant qu'expert : je peux désactiver une de mes compétences.
	En tant qu'expert : je peux afficher mes compétences.
	En tant qu'expert ou société : Recherche multicritères des compétences.
Gestion des CV	En tant qu'expert : je peux ajouter un CV.
	En tant que société : je peux consulter les CV.

TABLE 3.2 – Backlog du sprint 2

3.4.2 Réalisation

La figure 3.7 représente le profil expert en tant que CV réel. Dans cette page, l'expert doit ajouter ses informations nécessaires telles que ses données personnelles, ses compétences, ses diplômes et ses projets réalisés.

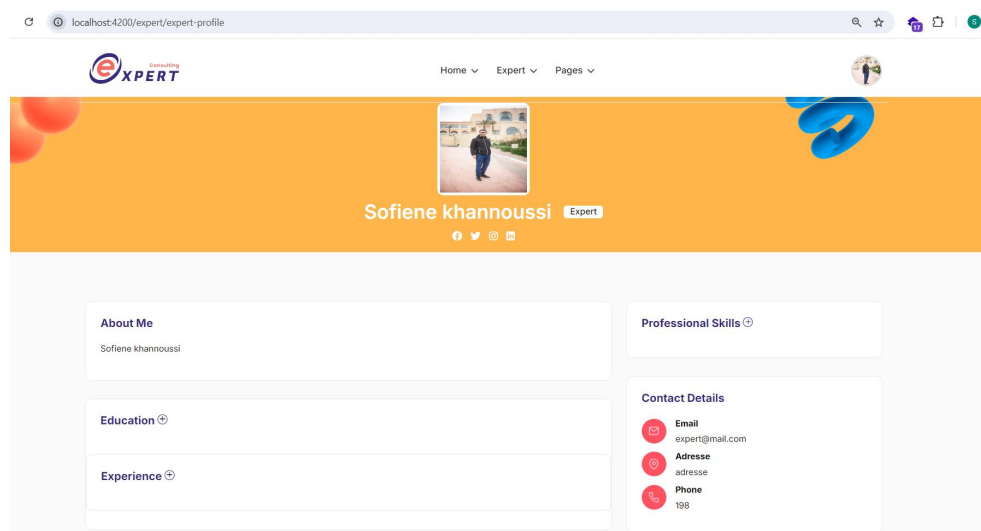


FIGURE 3.7 – Profil expert

La figure 3.8 représente l'ajout d'une nouvelle compétence

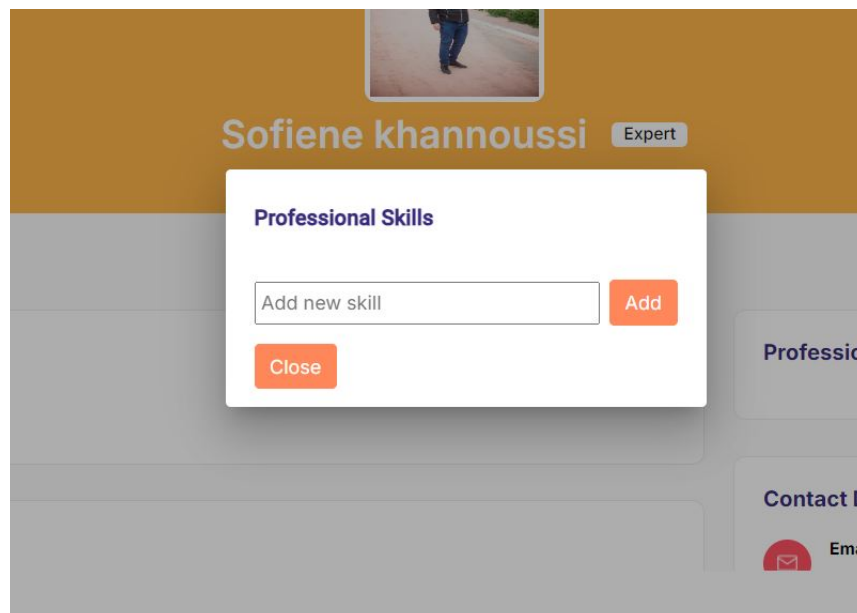
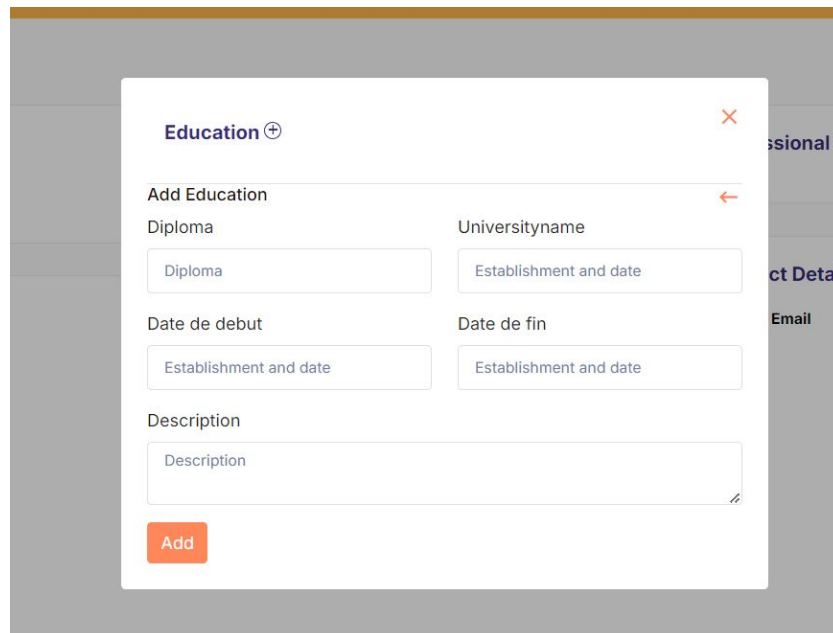


FIGURE 3.8 – Profil expert -Ajout des Compétences

La figure 3.9 représente l'ajout d'un nouveau diplome



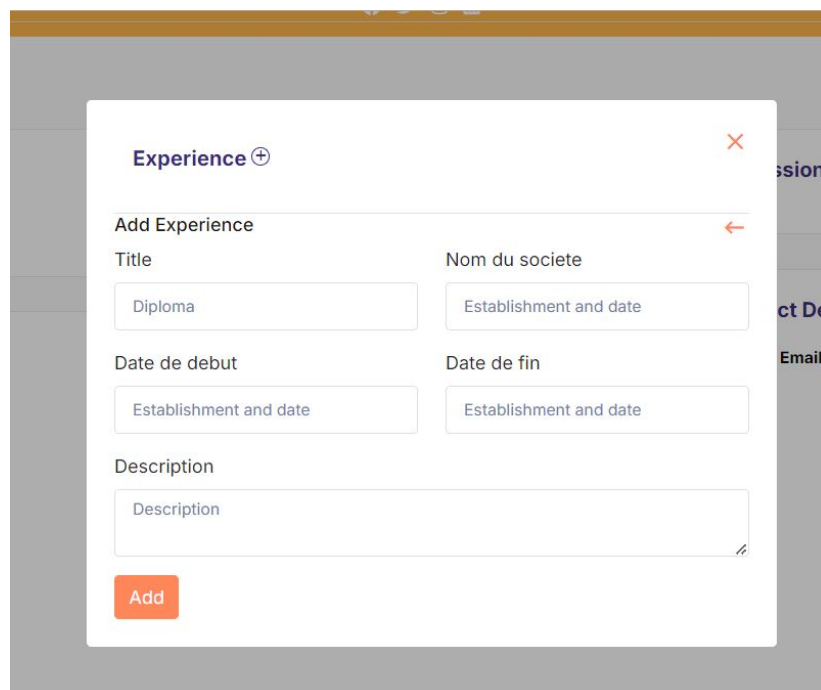
The screenshot shows a modal window titled "Education" with a close button (X) in the top right corner. Below the title is a section "Add Education" with a back arrow. The form contains the following fields:

- Diploma:** A text input field with the placeholder text "Diploma".
- Universityname:** A text input field with the placeholder text "Establishment and date".
- Date de debut:** A text input field with the placeholder text "Establishment and date".
- Date de fin:** A text input field with the placeholder text "Establishment and date".
- Description:** A text area with the placeholder text "Description".

At the bottom left of the form is an orange "Add" button.

FIGURE 3.9 – Profil expert - Ajout des Diplomes

La figure 3.10 représente l'ajout d'une nouvelle expérience



The screenshot shows a modal window titled "Experience" with a close button (X) in the top right corner. Below the title is a section "Add Experience" with a back arrow. The form contains the following fields:

- Title:** A text input field with the placeholder text "Diploma".
- Nom du societe:** A text input field with the placeholder text "Establishment and date".
- Date de debut:** A text input field with the placeholder text "Establishment and date".
- Date de fin:** A text input field with the placeholder text "Establishment and date".
- Description:** A text area with the placeholder text "Description".

At the bottom left of the form is an orange "Add" button.

FIGURE 3.10 – Profil expert - Ajout des missions réalisées

3.5 Release N°3 : Gestion des missions

Le troisième sprint de notre projet est dédié à la gestion des missions, permettant aux managers de créer, suivre et attribuer des missions aux experts. Il vise à faciliter le processus de gestion des offres de missions et leur suivi tout au long de leur cycle de vie. Les objectifs principaux sont l'efficacité, la fluidité de gestion et la centralisation des informations relatives aux missions.

3.5.1 Backlog du sprint 3

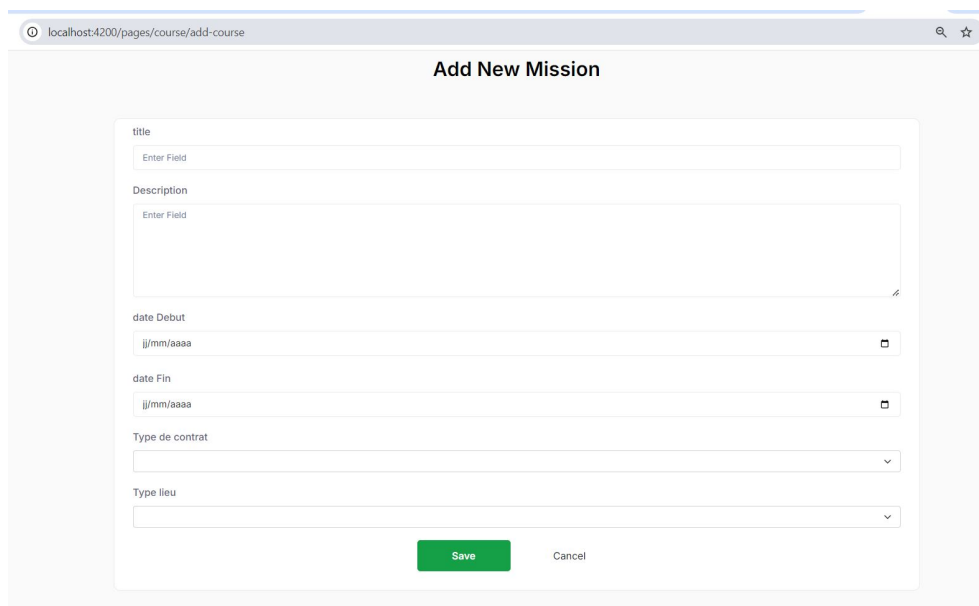
Dans cette section, nous examinerons en détail les fonctionnalités en décrivant le Backlog associé comme le montre le tableau 3.3. Ce Backlog fournira une vue complète des fonctionnalités à développer pour le sprint 3

Sprint	User Story
Gestion des missions (Société)	En tant que Société : je peux ajouter une nouvelle mission.
	En tant que Société : je peux Modifier une mission qui m'appartient.
	En tant que Société : je peux activer une mission qui m'appartient.
	En tant que Société : je peux désactiver une mission qui m'appartient.
	En tant que Société : je peux envoyer le détail de la mission à un expert.
	En tant que Société : je peux voir les candidats à une mission.
	En tant que Société : je peux rechercher les experts correspondants à ma mission.
Gestion des missions (Expert)	En tant qu'expert : je peux envoyer ma candidature pour une mission.
	En tant qu'expert : je peux visualiser les détails d'une mission.
	En tant qu'expert : je peux faire une recherche multicritères sur les missions.
	En tant qu'expert : je peux visualiser mes missions déjà faites.
	En tant qu'expert : je peux visualiser mes missions en cours.

TABLE 3.3 – Backlog du sprint 3

3.5.2 Réalisation

Dans cette partie, on va s'intéresser sur la fonctionnalité principale du Manger ou il peut ajouter une nouvelle mission comme représente la figure 3.11. Par conséquent, il doit renseigner les informations principales de la mission telles que la description, date de debut , date de fin...



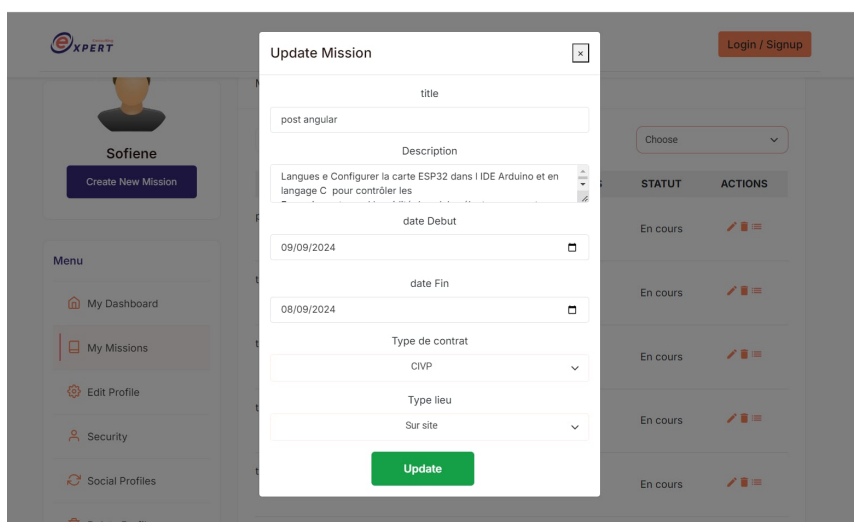
The screenshot shows a web browser window with the URL 'localhost:4200/pages/course/add-course'. The page title is 'Add New Mission'. The form contains the following fields:

- title**: A text input field with the placeholder 'Enter Field'.
- Description**: A text area with the placeholder 'Enter Field'.
- date Debut**: A date input field with the placeholder 'jj/mm/aaaa' and a calendar icon.
- date Fin**: A date input field with the placeholder 'jj/mm/aaaa' and a calendar icon.
- Type de contrat**: A dropdown menu.
- Type lieu**: A dropdown menu.

At the bottom of the form are two buttons: a green 'Save' button and a grey 'Cancel' button.

FIGURE 3.11 – Création d'une mission

Après avoir créer une mission, le manager peut modifier la mission



The screenshot shows a web application interface with a sidebar menu and a main content area. The sidebar menu includes a user profile for 'Sofiene' and a 'Create New Mission' button. The main content area displays a table with columns 'STATUT' and 'ACTIONS'. A modal titled 'Update Mission' is open, showing the following fields:

- title**: A text input field with the value 'post angular'.
- Description**: A text area with the value 'Langues e Configurer la carte ESP32 dans l IDE Arduino et en langage C pour contrôler les'.
- date Debut**: A date input field with the value '09/09/2024' and a calendar icon.
- date Fin**: A date input field with the value '08/09/2024' and a calendar icon.
- Type de contrat**: A dropdown menu with the value 'CIVP'.
- Type lieu**: A dropdown menu with the value 'Sur site'.

At the bottom of the modal is a green 'Update' button.

FIGURE 3.12 – Modification d'une mission

Après avoir créer et publier une mission, elle sera affiché dans la liste des offres . cette interface est dedié à l'expert pour consulter cette liste et pouvoir postuler dans les missions interessées comme le montre dans la figure 3.14

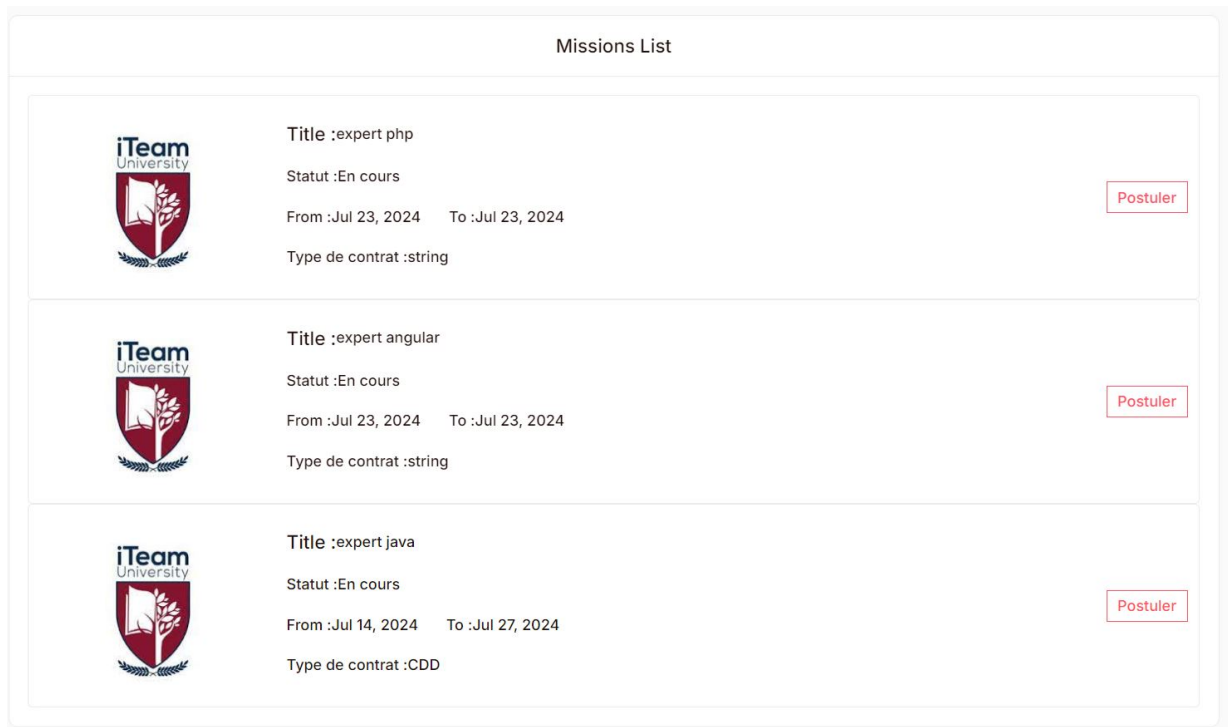


FIGURE 3.13 – Postuler une candidature

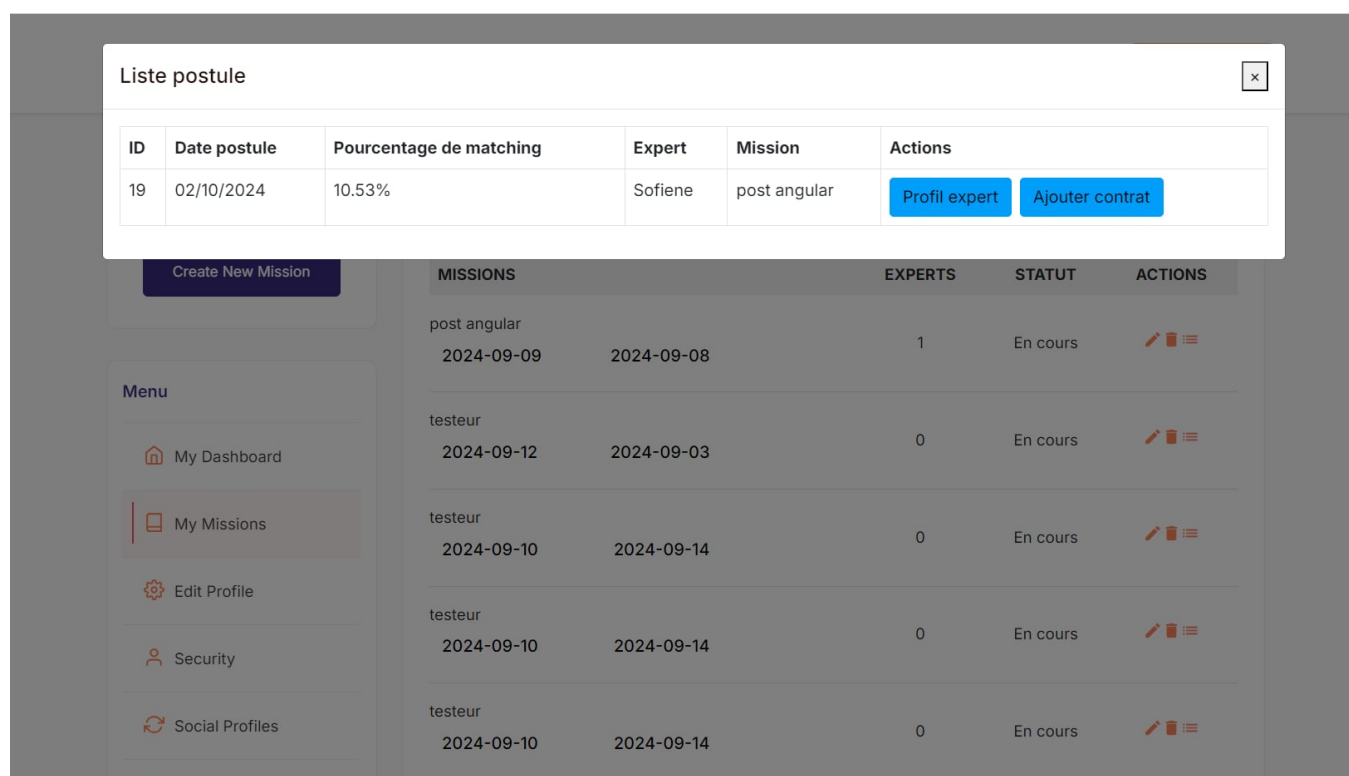
Mise en oeuvre de la technologie OCR

L'OCR est une technologie qui permet de convertir des documents numérisés, des images de texte manuscrit ou imprimé, en texte numérique éditable. Elle utilise des techniques de traitement d'image et d'intelligence artificielle pour identifier et extraire les caractères présents dans une image, permettant ainsi de traiter et de manipuler le texte de manière numérique.

L'OCR est couramment utilisé pour la numérisation de documents, la reconnaissance de texte dans des images, et la conversion de livres imprimés en formats numériques.

Les figures 3.13 et 3.14 illustrent les étapes détaillées du processus de correspondance entre le CV et la description de l'offre, généré par l'OCR. Étape 1 : Conversion du CV en format PDF en fichier texte : Le CV, initialement au format PDF, est converti en texte brut à l'aide de la technologie OCR, qui identifie et extrait les caractères du document numérisé.

La figure 3.15 représente la liste des postule à une mission



The image shows a web application interface. A modal window titled 'Liste postule' is open, displaying a table with one row of data. The background shows a table with columns for 'MISSIONS', 'EXPERTS', 'STATUT', and 'ACTIONS'. A sidebar menu is visible on the left.

ID	Date postule	Pourcentage de matching	Expert	Mission	Actions
19	02/10/2024	10.53%	Sofiene	post angular	Profil expert Ajouter contrat

MISSIONS		EXPERTS	STATUT	ACTIONS
post angular				
2024-09-09	2024-09-08	1	En cours	
testeur				
2024-09-12	2024-09-03	0	En cours	
testeur				
2024-09-10	2024-09-14	0	En cours	
testeur				
2024-09-10	2024-09-14	0	En cours	
testeur				
2024-09-10	2024-09-14	0	En cours	

FIGURE 3.14 – Liste des candidatures

3.6 Release N°4 : Gestion des contrats

Ce release se focalise sur la mise en place des fonctionnalités liées à la gestion des contrats dans notre plateforme de gestion des experts. Les managers peuvent désormais créer, modifier et valider des contrats directement via l'interface. Cela permet de formaliser et suivre les engagements des experts sur leurs missions respectives. Ce module garantit une gestion centralisée et fluide des contrats.

3.6.1 Backlog du sprint 4

Le Tableau 3.4 décrit les user Stories du sprint 4

Sprint	User Story
Gestion des contrats (Société)	En tant que Société : je peux valider le candidat accepté pour une mission.
	En tant que Société : je peux créer un contrat avec l'expert accepté pour une mission.
	En tant que Société : je peux modifier un contrat avec l'expert accepté pour une mission.
	En tant que Société : je peux arrêter un contrat avec l'expert accepté pour une mission.
	En tant que Société : je peux visualiser les contrats en cours.
	En tant que Société : je peux visualiser les activités d'un expert dans un contrat.
	En tant que Société : je peux visualiser les contrats finis.
Gestion des contrats (Expert)	En tant qu'expert : je peux valider un contrat où j'ai été accepté.
	En tant qu'expert : je peux visualiser mes contrats.
	En tant qu'expert : je peux suivre mes activités de mes contrats.
Gestion des contrats (Admin)	En tant que Super-admin : je peux visualiser les contrats validés pendant une période.
	En tant que Super-admin : je peux visualiser les contrats d'une société.
	En tant que Super-admin : je peux visualiser les contrats d'un expert.

TABLE 3.4 – Backlog du sprint 4

3.6.2 Réalisation

Après avoir finir le sprint 3, en validant un expert à l'aide d'OCR, l'expert le plus proche de la mission est celui qui a la grande pourcentage de matching par apport à l'autr .la figure 3.15 représente la création d'un contrat ;

The screenshot shows the 'Ajouter un contrat' (Add Contract) form in the EXPERT application. The interface includes a top navigation bar with 'Home', 'Features', and 'Contact' links, and a 'Login / Signup' button. The user profile 'Sofiene' is visible on the left, with a 'Create New Mission' button. The main form area is titled 'Profile Details' and contains the following fields:

Reference	Type
<input type="text" value="kjkjkj"/>	<input type="text" value="satge"/>

Durée	Date
<input type="text" value="20"/>	<input type="text" value="15/10/2024"/>

Below the date field is a PDF icon and a red 'Ajouter contrat' button.

FIGURE 3.15 – Ajouter un contrat

3.7 Release N°5 : Gestion des tableaux de bord et messagerie interne

Ce dernier sprint se concentre sur la gestion des tableaux de bord, offrant aux utilisateurs la possibilité de visualiser leurs missions et contrats. Chaque utilisateur peut consulter un aperçu global de ses activités. Cette fonctionnalité permet une meilleure gestion et suivi des missions en cours. Ainsi, les managers et les experts peuvent aisément vérifier l'état de leurs engagements respectifs.

3.7.1 Backlog du sprint 5

Dans cette section, nous allons détailler les fonctionnalités en présentant le Backlog correspondant. Ce Backlog offre une vue exhaustive des fonctionnalités développées pour le sprint 5, en mettant en avant les priorités, les objectifs et les tâches clés associées à chaque fonctionnalité. .

Le Tableau 3.5 décrit les user Stories du sprint 5

Sprint	User Story
Tableau de bord (Expert)	Expert : tableau de bord missions, contrats, compétences.
Tableau de bord (Société)	Société : tableau de bord missions, contrats.
Tableau de bord (Super-admin)	Super-admin : tableau de bord global missions, contrats.
	Super-admin : tableau de bord missions, contrats par expert.
	Super-admin : tableau de bord missions, contrats par société.
Messagerie interne	En tant qu'expert : je peux envoyer ou recevoir des messages.

TABLE 3.5 – Backlog du sprint 5

3.7.2 Réalisation

L'interface Dashboard dans la figure 3.13

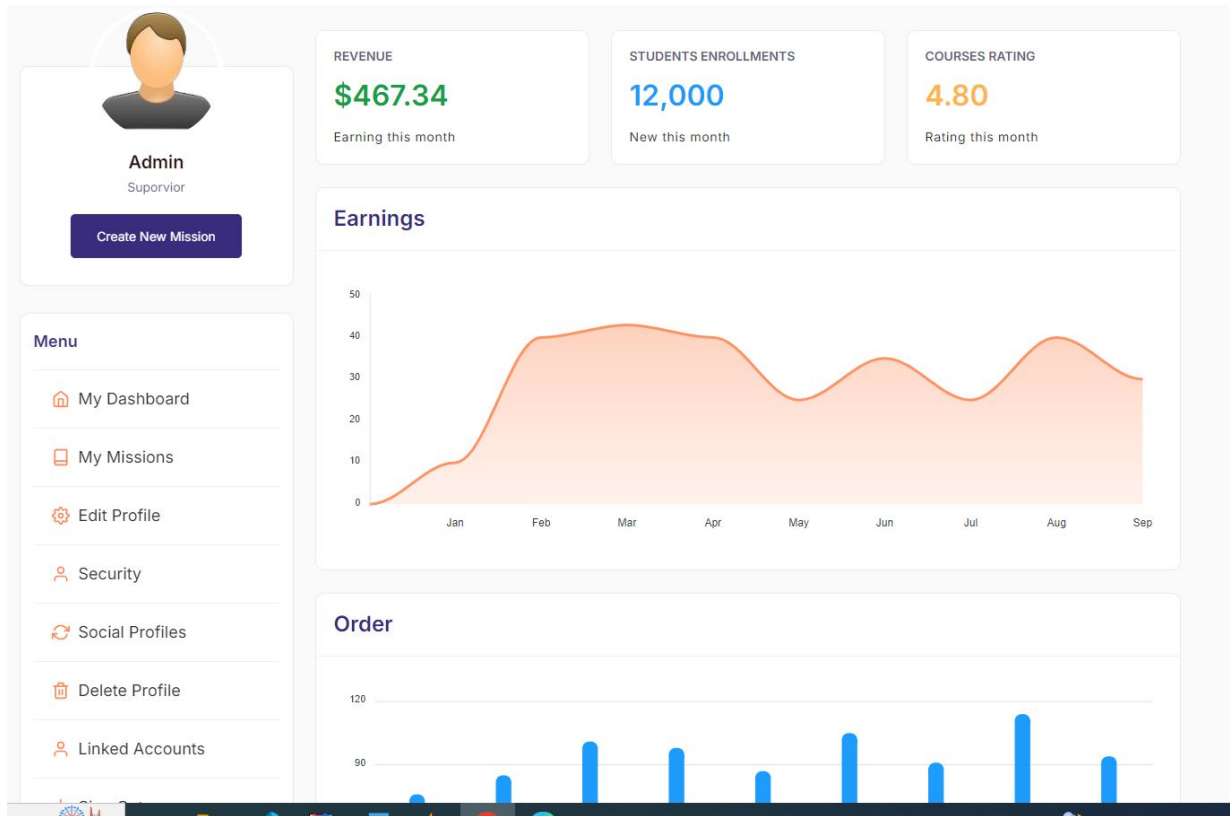


FIGURE 3.16 – Dashboard

3.8 Environnement de Développement

Nous avons sélectionné une combinaison d'outils et de technologies pour assurer une réalisation efficace et robuste du projet. Chaque outil a été choisi en fonction de ses avantages spécifiques pour notre projet.

- **IDE** : Visual Studio Code pour sa polyvalence et ses extensions riches.
- **Système de Gestion de Versions** : Git pour le contrôle de versions, avec GitHub comme plateforme de collaboration.
- **Langage Backend** : Java avec Spring Boot, pour une architecture robuste et extensible.
- **Langage Frontend** : TypeScript avec Angular pour une interface utilisateur réactive.

- **Base de Données** : MySQL, pour sa fiabilité et sa large adoption.
- **Serveur Web** : Nginx, utilisé pour déployer l'application en production.

3.9 Structure du Projet

3.9.1 Organisation du Code Source

Le code source est organisé en plusieurs modules distincts pour améliorer la maintenabilité et la scalabilité du projet. Chaque module se concentre sur une partie spécifique de l'application.

- **Backend** : Contient le code serveur, organisé en services, contrôleurs et entités.
- **Frontend** : Comprend les composants Angular, les services, les modules et les routes.
- **Base de Données** : Inclut les scripts SQL pour la création des schémas et des tables.
- **Tests** : Séparés en tests unitaires et fonctionnels pour les différentes couches de l'application.

3.9.2 Détails des Modules

Le code de notre projet se compose de deux parties essentielles :

Backend :

- *Controllers* : Gère les requêtes HTTP et les routes.
- *Services* : Contient la logique métier, souvent réutilisable par plusieurs contrôleurs.
- *Repositories* : Gestion des accès aux données via JPA.

Frontend :

- *Components* : Représentent les différentes vues et interactions utilisateurs.
- *Services* : Fournissent des données aux composants en interagissant avec le backend.
- *Routing* : Gère la navigation et les différentes pages de l'application.

3.10 Déploiement

La phase de déploiement consiste à mettre en production l'application en transférant les fichiers sur un serveur dédié, configurant les services nécessaires et s'assurant que toutes les fonctionnalités sont opérationnelles pour les utilisateurs finaux.

3.10.1 Préparation du Déploiement

Avant le déploiement, plusieurs étapes cruciales ont été réalisées pour s'assurer que le système est prêt pour la production. Ces étapes incluent :

- **Tests Complets** : Tous les modules de l'application ont été soumis à des tests unitaires, d'intégration et de performance pour s'assurer de leur bon fonctionnement et de la stabilité globale du système.
- **Configuration de l'Environnement** : Un environnement de production a été configuré pour ressembler étroitement à l'environnement de développement, minimisant ainsi les risques de problèmes inattendus lors du déploiement.
- **Sauvegarde des Données** : Une sauvegarde complète des bases de données et des fichiers de configuration a été réalisée pour assurer la récupération rapide en cas de problème lors du déploiement.
- **Validation des Dépendances** : Toutes les dépendances du projet, y compris les bibliothèques et les services tiers, ont été vérifiées et mises à jour si nécessaire pour garantir leur compatibilité et leur performance en production.

3.10.2 Déploiement sur Serveur

L'application a été déployée sur un serveur Linux, en utilisant Nginx comme serveur web. Ce choix offre une performance élevée et une gestion efficace des connexions simultanées, ce qui est essentiel pour le système de gestion des experts.

3.10.3 Étapes de Déploiement

Les étapes de déploiement incluent :

1. **Installation de Nginx** : Nginx a été configuré comme serveur web pour gérer les requêtes HTTP et servir le frontend de l'application.
2. **Déploiement du Backend** : Le backend a été déployé en utilisant un conteneur Docker pour isoler les services et assurer une meilleure gestion des dépendances.
3. **Configuration de la Base de Données** : MySQL a été installé et configuré sur le serveur, avec des scripts de migration exécutés pour préparer les schémas de base de données.
4. **Automatisation du Déploiement** : Des scripts d'automatisation ont été mis en place pour simplifier le déploiement et réduire les erreurs humaines. Ces scripts gèrent les mises à jour du code, la construction du projet et le redémarrage des services nécessaires.

3.10.4 Scripts d'Automatisation

Le processus de déploiement a été automatisé à l'aide de scripts shell, garantissant une mise à jour rapide et fiable des composants de l'application. Voici un exemple de script de déploiement utilisé pour ce projet :

```
[language=bash, caption=Script de déploiement pour le projet]#!/bin/bash  Script de dé-  
ploiement pour le projet de gestion des experts
```

```
Mettre à jour le code source git pull origin main
```

```
Construire le projet mvn clean package
```

```
Redémarrer le serveur sudo systemctl restart nginx
```

Ce script assure une mise à jour fluide du code source en récupérant les dernières modifications depuis le dépôt Git, puis en construisant le projet avec Maven pour s'assurer que toutes les dépendances sont à jour et que le code est compilé correctement. Enfin, le serveur Nginx est redémarré pour appliquer les changements.

3.10.5 Surveillance et Maintenance Post-Déploiement

Après le déploiement, une surveillance continue est mise en place pour s'assurer que le système fonctionne comme prévu. Cela inclut :

- **Surveillance des Performances** : Des outils comme Grafana et Prometheus sont utilisés pour surveiller les performances du serveur et détecter les anomalies.
- **Gestion des Logs** : Les journaux du système sont analysés régulièrement pour identifier les erreurs potentielles et les problèmes de performance.
- **Mises à Jour et Patches** : Les mises à jour de sécurité et les patches sont appliqués régulièrement pour protéger le système contre les vulnérabilités.
- **Support Utilisateur** : Un canal de support est mis en place pour permettre aux utilisateurs de signaler des bugs ou des problèmes, facilitant ainsi la maintenance proactive de l'application.

Cette approche structurée de déploiement assure que le système de gestion des experts est non seulement déployé de manière efficace, mais aussi maintenu en bon état de fonctionnement, répondant aux attentes des utilisateurs finaux et aux exigences de performance.

3.11 Dockerisation du Projet

La dockerisation est une étape essentielle dans la modernisation et la simplification du déploiement des applications. En encapsulant notre application dans des conteneurs Docker, nous assurons une portabilité accrue, une gestion simplifiée des dépendances et un

déploiement cohérent à travers différents environnements. Ce chapitre explore le processus de dockerisation du projet de gestion des experts, incluant la création des fichiers Docker, la configuration des conteneurs, et la mise en place de la gestion des réseaux et des volumes pour une application scalable et maintenable.

3.11.1 Avantages de la Dockerisation

La dockerisation offre plusieurs avantages pour notre projet :

- **Isolation des Environnements** : Chaque microservice ou composant de l'application est isolé dans son propre conteneur, ce qui permet de gérer les dépendances et les configurations de manière indépendante.
- **Portabilité** : Les conteneurs Docker peuvent être exécutés sur n'importe quelle machine qui supporte Docker, ce qui garantit que l'application fonctionnera de la même manière sur tous les environnements (développement, test, production).
- **Déploiement Simplifié** : Avec Docker, les applications peuvent être déployées de manière rapide et efficace à l'aide de commandes simples, réduisant ainsi le temps nécessaire pour configurer et mettre en ligne de nouveaux environnements.
- **Scalabilité** : Docker facilite le scaling horizontal de l'application en permettant de lancer plusieurs instances de conteneurs pour répondre aux besoins croissants de l'utilisateur.

3.11.2 Création des Fichiers Docker

Pour chaque composant de notre application, un fichier **Dockerfile** est créé afin de définir l'image Docker correspondante. Voici un exemple de fichier **Dockerfile** pour le backend de l'application :

[language=Dockerfile, caption=Exemple de Dockerfile pour le backend] Utiliser une image de base officielle Java FROM openjdk :11-jre-slim

Définir le répertoire de travail WORKDIR /app

Copier le fichier JAR de l'application dans le conteneur COPY target/backend.jar .

Exposer le port sur lequel l'application écoute EXPOSE 8080

Commande pour exécuter l'application CMD ["java", "-jar", "backend.jar"]

3.11.3 Gestion des Réseaux et Volumes

Pour permettre la communication entre les différents conteneurs, un réseau Docker dédié est créé. Les volumes Docker sont utilisés pour persister les données, en particulier pour

la base de données, afin que les informations ne soient pas perdues lorsque les conteneurs sont redémarrés.

3.12 Déploiement des Conteneurs

Le déploiement des conteneurs Docker est simplifié grâce à Docker Compose. En exécutant la commande suivante, tous les services définis dans le fichier `docker-compose.yml` sont déployés et démarrés :

[language=bash, caption=Commande pour démarrer les conteneurs] `docker-compose up -d`

Cette commande construit les images, configure les réseaux et démarre les conteneurs en arrière-plan. Les logs et l'état des conteneurs peuvent être consultés via les commandes Docker standards.

3.12.1 Automatisation du Déploiement

Pour faciliter le déploiement continu et l'intégration continue, des pipelines CI/CD peuvent être configurés pour construire et déployer automatiquement les conteneurs Docker à chaque mise à jour du code source. Des outils tels que Jenkins, GitLab CI, ou GitHub Actions peuvent être intégrés pour orchestrer ces processus.

3.13 Conclusion

La phase de réalisation a permis de transformer les conceptions en une application fonctionnelle, prête à être utilisée par les experts et les managers. Chaque étape a été documentée et les choix technologiques ont été justifiés pour assurer la qualité et la maintenabilité du projet.

Conclusion générale & perspectives

Ce présent rapport s'inscrit dans le cadre d'un stage de fin d'étude pour une durée de six mois au sein de l'entreprise Medianet Consulting. Le travail effectué se résume dans la réalisation d'une plateforme de gestion des experts 'Expert Consulting'.

Hormis le côté technique, ce projet m'a permis de m'intégrer dans le monde professionnel et d'appréhender le travail dans une hiérarchie professionnelle. En outre, j'ai eu l'occasion de mettre en pratique mes études universitaires et approfondir mes compétences dans le domaine du développement.

J'ai également appris à appliquer les bonnes pratiques de la réalisation d'un produit de qualité tout en participant au déploiement de ce dernier.

Mis à part les changements de besoins qui ont apparus au cours du développement, les obstacles de l'environnement technique rencontrés ainsi que les difficultés inhérentes comme la répartition du temps et des efforts, je considère que mon travail a parfaitement atteint ses objectifs spécifiés. J'ai aussi réussi à implémenter d'autres fonctionnalités qui n'ont pas été planifiées au début.

Par ailleurs, des fonctionnalités avancées comme l'intégration d'un système de Traitement du Langage Naturel (NLP) pour l'analyse et le filtrage des demandes de candidatures n'ont pas encore été mises en œuvre. Le NLP, contrairement à l'OCR, permettrait de lire et d'interpréter le contexte des documents, offrant ainsi une analyse plus fine et précise des compétences et des expériences des candidats. Un modèle NLP bien entraîné pourrait améliorer la précision et l'efficacité du processus de sélection, assurant des temps de traitement rapides et une amélioration continue grâce à l'apprentissage des nouvelles données. De plus, le NLP garantirait une meilleure confidentialité et sécurité des données des utilisateurs tout en étant capable de s'adapter aux évolutions des critères de sélection.

Finalement, je tiens à souligner quelques perspectives intéressantes de notre projet. Il peut être étendu afin de satisfaire les futurs besoins éventuels. À l'issue de ce projet, nous pouvons après développer d'autres modules intéressants tel que la messagerie interne et les notifications.

Références

- [1] **Medianet** : Organisme d'accueil [**en ligne**]. Disponible sur :
<https://www.medianet.tn/>
- [2] **Tanitjobs** : site de gestion des candidats. Disponible sur :
<https://tanitjobs.com/>
- [3] **Malt** : Plateforme française de gestion des candidats. Disponible sur :
<https://www.malt.fr/>
- [4] **Scrum**. Disponible sur :
<https://fr.wikipedia.org/wiki/Scrum>
- [5] **Tuleap**. Application Lifecycle Management [**en ligne**]. Disponible sur :
<https://www.tuleap.org/fr/>

Résumé — Ce projet de fin d'études a été réalisé au sein de l'entreprise Medianet Consulting, de 1 Mars 2024 à 31 Août 2024. Il s'agit du développement d'une application de gestion des experts en développement informatique, intégrant des fonctionnalités telles que la gestion des compétences, des missions et des candidatures. Un système de matching OCR a été mis en place pour automatiser l'analyse des CVs, facilitant ainsi le processus de sélection. L'application est développée en utilisant Spring Boot pour le backend et Angular pour le frontend. Ce projet suit une méthodologie Scrum et met en œuvre des concepts avancés tels que l'authentification sécurisée avec JWT.

Mots clés : Expert, Manager, Postule Compétences, missions, Rest API, Scrum, Angular, SpringBoot, JWT , OCR.

Abstract — This final year project was carried out at the company Medianet Consulting from March 1, 2024, to August 31, 2024. It involves the development of an application for managing experts in software development, integrating features such as skills management, mission tracking, and candidate applications. A matching OCR system was implemented to automate the analysis of CVs, thus facilitating the selection process. The application is developed using Spring Boot for the backend and Angular for the frontend. This project follows a Scrum methodology and implements advanced concepts such as secure authentication with JWT.

Keywords : Expert, Manager, Skills, Missions, Rest API, Scrum, Angular, Spring Boot, JWT, OCR.
