

# Gestion des projets dans une entreprise

Khalil Smairi –Jihed Kdidi –Baha eddine Fatnassi –Nour el Houda Chouket

September 2024

# Introduction Général

Dans un contexte de transformation numérique et de concurrence accrue, les entreprises doivent s'adapter aux défis imposés par la complexité croissante des projets et la rapidité des évolutions technologiques. La gestion efficace des tâches et des projets joue un rôle clé pour assurer le succès et la pérennité des activités.

Ce rapport s'inscrit dans le cadre d'un projet académique visant à développer une application web de gestion de projet. Cette solution a pour objectif de fournir une plateforme intuitive et personnalisable, permettant aux équipes de planifier, suivre et collaborer efficacement sur leurs projets.

Nous aborderons tout d'abord le cadre général du projet, suivi d'une analyse de l'existant pour identifier les problématiques actuelles. Ensuite, nous présenterons la solution proposée ainsi que les méthodologies de travail, notamment la méthode agile Scrum.

L'objectif final est de démontrer comment cette application répond aux besoins identifiés et améliore la gestion des projets en offrant une solution performante et adaptée.

# Chapitre 1

## Cadre Général du Projet

# Introduction

Nous commencerons ce premier chapitre par une description du cadre général du projet, ainsi qu'une introduction de l'organisme d'accueil et des différentes notions énoncées dans la suite du rapport. Dans la présentation du projet, nous étudierons l'existant, qui nous mènera vers une problématique.

Ensuite, Nous proposerons la solution à ce problème.

À la fin de ce chapitre, Nous aborderons l'étude fonctionnelle et non fonctionnelle du projet, ainsi qu'une introduction aux méthodes de travail utilisées.

## 1.1 Présentation de l'entreprise : TEK UP

TEK-UP est une institution qui vous offre la possibilité de bénéficier d'une formation académique reconnue et appréciée dans le monde entier. L'université décerne le diplôme national en génie informatique ou en télécommunications.

### 1.1.1 Contexte général du projet

Ce projet a été réalisé dans le cadre de la validation de la matière **Projet de Développement** à l'Institut Tekup. Il se déroulera sur une période de trois mois et a pour objectif de développer une application de gestion de projet. Cette application vise à faciliter la gestion de nos futurs projets tout au long de notre parcours académique.

### 1.1.2 problématique

De nos jours, les entreprises sont confrontées à des défis majeurs, tels que la concurrence mondiale, les avancées technologiques rapides et les réorganisations fréquentes de leurs structures. Dans ce contexte, la gestion efficace des tâches est devenue essentielle pour assurer la performance et la réussite des projets.

Une bonne gestion des tâches permet non seulement de faciliter la coordination entre les équipes, mais également de suivre l'évolution des projets et d'accéder rapidement aux informations nécessaires. En fournissant des indicateurs clés, cette gestion aide les responsables à prendre des décisions stratégiques éclairées.

Pour mettre en place un système de gestion des tâches performant et adapté

aux besoins actuels des entreprises, il est crucial d'utiliser un logiciel de gestion de projet de qualité. Ce type d'outil doit non seulement optimiser la planification et le suivi des tâches, mais également favoriser la collaboration entre les membres de l'équipe, contribuant ainsi à une meilleure efficacité opérationnelle.

### 1.1.3 Etude de l'existant

Actuellement, des outils populaires comme **Trello**, **Jira** et **Asana** dominent le marché de la gestion de projets et des tâches. Ces plateformes offrent des fonctionnalités variées :

- **Trello** : Gestion visuelle des tâches à l'aide de tableaux Kanban simples et intuitifs.
- **Jira** : Outil puissant pour la gestion de projets complexes, particulièrement adapté aux méthodologies agiles (Scrum, Kanban).
- **Asana** : Collaboration et planification avancées avec une interface conviviale.

Malgré leurs avantages, ces outils présentent des **limitations significatives** dans certains contextes spécifiques :

- Ils manquent souvent de **personnalisation** pour répondre aux besoins particuliers d'une organisation ou d'un client.
- Les fonctionnalités de suivi et de gestion des rôles restent limitées sans passer par des modules payants ou des intégrations externes.
- L'absence d'une **attribution centralisée des projets aux chefs de projet** et d'un suivi clair des relations entre les projets et les tâches.
- Une gestion limitée des utilisateurs, rendant difficile la personnalisation des permissions selon les rôles.

### 1.1.4 Solution proposée

Après une étude approfondie de l'existant et afin d'apporter une solution cohérente aux problèmes cités précédemment, nous avons décidé de réaliser une application web de gestion de projet. Cette application offre plusieurs services :

- ✓ Elle permet à l'administrateur de gérer les utilisateurs, les projets de campagne et d'assigner un chef de projet.
- ✓ Elle permet aux chefs de projet de gérer des tâches, d'assigner des tâches aux employés et de changer l'état du projet.

✓Elle permet aux employés d'accéder aux informations des projets et de changer l'état des tâches.

✓Elle notifie les utilisateurs par mail lorsqu'ils ont une nouvelle tâche ou projet.

L'objectif de cette application est de faciliter l'échange de données entre différentes applications en utilisant des web services REST, permettant ainsi une intégration fluide avec divers systèmes.

## **1.2 Choix de la méthodologie de développement**

### **1.2.1 Méthodologie agile**

La méthode agile est un processus qui aide l'équipe à gérer un projet en le divisant en plusieurs étapes. Tout au long du projet, les membres de l'équipe collaborent et cherchent constamment à améliorer leurs performances, en prenant en considération les attentes du client a propos du produit final.

Ensuite, l'équipe travaille ensemble pour planifier, réaliser et évaluer le projet, en ajustant éventuellement le produit final pour répondre aux besoins du client. Une collaboration continue est essentielle entre les membres de l'équipe et les parties prenantes du projet pour prendre de décisions éclairées.

### **1.2.2 Méthodologie de conception SCRUM**

Dans cette section, nous présentons la méthode de travail adoptée, notamment Scrum, une approche agile alignée sur la culture de l'entreprise. Scrum fixe des objectifs à court terme, évalue les résultats, puis adapte les étapes suivantes pour atteindre l'objectif final.

Cette méthode repose sur une collaboration étroite de l'équipe et une livraison itérative des fonctionnalités (toutes les 2 à 4 semaines), tout en impliquant le client à chaque étape. La figure 1.1 illustre clairement le processus Scrum qui structure efficacement notre projet

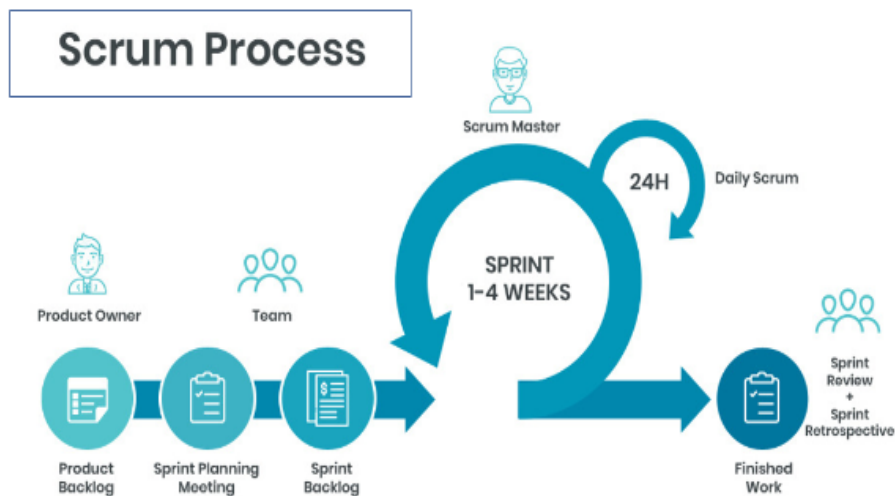


FIGURE 1.1 – Scrum Process

**Sprint** : Période courte et fixe au cours de laquelle des activités sont réalisées, aboutissant à un incrément fonctionnel du produit.

**Backlog Produit** : Liste des exigences et besoins du client, souvent mise à jour en fonction des changements.

**Backlog Sprint** : Sous-ensemble du backlog produit, défini au début d'un sprint, comprenant les fonctionnalités à réaliser pour atteindre l'objectif du sprint.

**Les acteurs en Scrum** :

- **Product Owner** : Gère le backlog produit et veille à sa compréhension par l'équipe.
- **Scrum Master** : Facilite la méthode Scrum et assure la productivité de l'équipe.
- **Équipe** : Développe les fonctionnalités du produit.

### 1.3 Langage de modélisation :

UML (Unified Modeling Language) est un langage de modélisation graphique qui permet de représenter et de visualiser les systèmes logiciels. Il est utilisé pour la conception, la documentation et la communication des systèmes orientés objet. UML est utilisé dans le processus de développement de logiciels pour aider à la conception et à la modélisation de systèmes complexes.

La figure ci-dessous décrit le langage de modélisation graphique. UML constitue un outil précieux pour visualiser et appréhender la structure et le comportement du système modélisé.



FIGURE 1.2 – UML

### ***Conclusion :***

Dans ce chapitre, nous avons donné un aperçu sur le projet en décrivant l'organisme d'accueil, et le contexte. Nous avons présenté aussi notre méthodologie de travail pour réaliser notre projet, comme la méthode Scrum. Le reste du rapport est organisé selon cette méthodologie. Le chapitre suivant sera consacré à l'analyse et à la spécification des besoins de notre projet.



## Chapitre 2

# Analyse et spécification des Besoins

## Introduction

Nous allons, dans ce chapitre, clarifier et organiser les besoins de l'application en identifiant les besoins fonctionnels en premier lieu et les besoins non fonctionnels en second lieu. Nous allons, par la suite, énoncer les éléments du Backlog du produit.

### 2.1 Présentation des acteurs

Dans ce qui suit, nous allons présenter les différents acteurs interagissant avec notre système. En effet, un acteur peut consulter et modifier directement l'état du système, en émettant ou en recevant des messages éventuellement porteurs de données. Les principaux acteurs ainsi que leurs rôles sont présentés dans la table suivant :

Tableau 2.1 – Acteurs et Rôles

Acteur	Rôle
Administrateur	C'est un acteur qui doit pouvoir gérer l'application et son contenu. C'est la seule personne qui doit avoir tous les privilèges d'exploiter toutes les fonctionnalités de l'application et de modifier tout type d'information.
Chef de projet	C'est un acteur qui doit gérer les tâches, assigner des employés pour les tâches et changer le statut du projet.
Employé	C'est un acteur qui a accès à la description des projets et des tâches, et qui peut changer le statut de la tâche sur laquelle il travaille.

### 2.2 Spécification des besoins

Une étape importante du cycle de vie logique consiste en des identifiants entre fonctions et non-fonctions expérimentant des exigences d'utilisateur. Ce livre est suivi d'une modélisation des fonctions du système à l'aide de divers diagrammes et outils de modélisation.

#### 2.2.1 Spécification des besoins fonctionnels

Dans cette partie, nous allons identifier les besoins fonctionnels par acteur, c'est-à-dire l'action que peut effectuer chaque intervenant sur le système et

ceci dans le but de préciser le rôle de chaque acteur au sein du futur système.

- **Authentification** : l'utilisateur doit s'authentifier afin d'assurer un accès sécurisé et bénéficier des fonctionnalités offertes de l'application.

- **Gestion des Utilisateurs** : Le système doit permettre la création, modification, et suppression des utilisateurs. Chaque utilisateur se verra attribuer un rôle spécifique (administrateur, chef de projet, ou membre d'équipe) avec des droits d'accès ajustés en fonction de ce rôle. La gestion des permissions doit assurer que seuls les utilisateurs autorisés puissent accéder aux projets et tâches qui leur sont assignés.

- **Gestion des projets** : Les utilisateurs pourront créer, modifier et supprimer des projets. Chaque projet contiendra des informations telles que le nom, la description, et les dates de début et de fin. Il sera également possible d'assigner des membres de l'équipe aux projets, tout en permettant le suivi de l'avancement du projet (en cours, terminé, annulé).

- **Gestion des tâches** : Le système permettra la création et la gestion des tâches au sein de chaque projet. Ces tâches pourront être assignées à des utilisateurs, avec des délais et des priorités spécifiques. Chaque tâche aura un état d'avancement (non démarrée, en cours, terminée) pour un suivi plus efficace.

- **Mise en place d'un tableau de Bord** : Un tableau de bord présentera une vue d'ensemble des projets et tâches en cours. Les utilisateurs pourront visualiser l'état d'avancement des projets, les tâches assignées et les échéances à venir. Des graphiques aideront à suivre la progression et les priorités seront affichées de manière claire.

- **Système de notifications par email** : Le système enverra des notifications par email aux utilisateurs pour les informer des événements clés, tels que la création ou la modification de tâches et de projets, les échéances approchant, ou les tâches en retard. Les utilisateurs pourront personnaliser leurs préférences de notification pour recevoir des alertes selon leurs besoins.

## 2.2.2 Spécification des Besoins non fonctionnels

Un besoin non fonctionnel est une restriction ou une contrainte qui pèse sur un service du système, tel que les contraintes liées à l'environnement et à

l'implémentation et les exigences en matière de performances, de dépendance de plateforme, facilité de maintenance, extensibilité et fiabilité. Les besoins non fonctionnels du site web réalisé sont :

- **Sécurité** : L'application doit être sécurisée par le système d'authentification.
- **Performance** : Une application doit être avant tout performante, c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usages d'une manière optimale.
- **Ergonomie** : Les interfaces permettant la présentation de l'évolution des données doivent être conviviales et assurer une navigation aisée.
- **Design** : le style des standards de l'interface : un design moderne, des pages claires, des images significatives et des couleurs attirantes.

## 2.3 Diagramme de cas d'utilisations globale

La Figure ci dessous met en évidence le diagramme de cas d'utilisation global, qui joue un rôle essentiel en fournissant une vision complète du comportement fonctionnel de notre système. Ce diagramme permet de visualiser les différentes interactions entre les acteurs et les fonctionnalités du système, En exposant ce diagramme, nous facilitons la compréhension des fonctionnalités clés offertes.

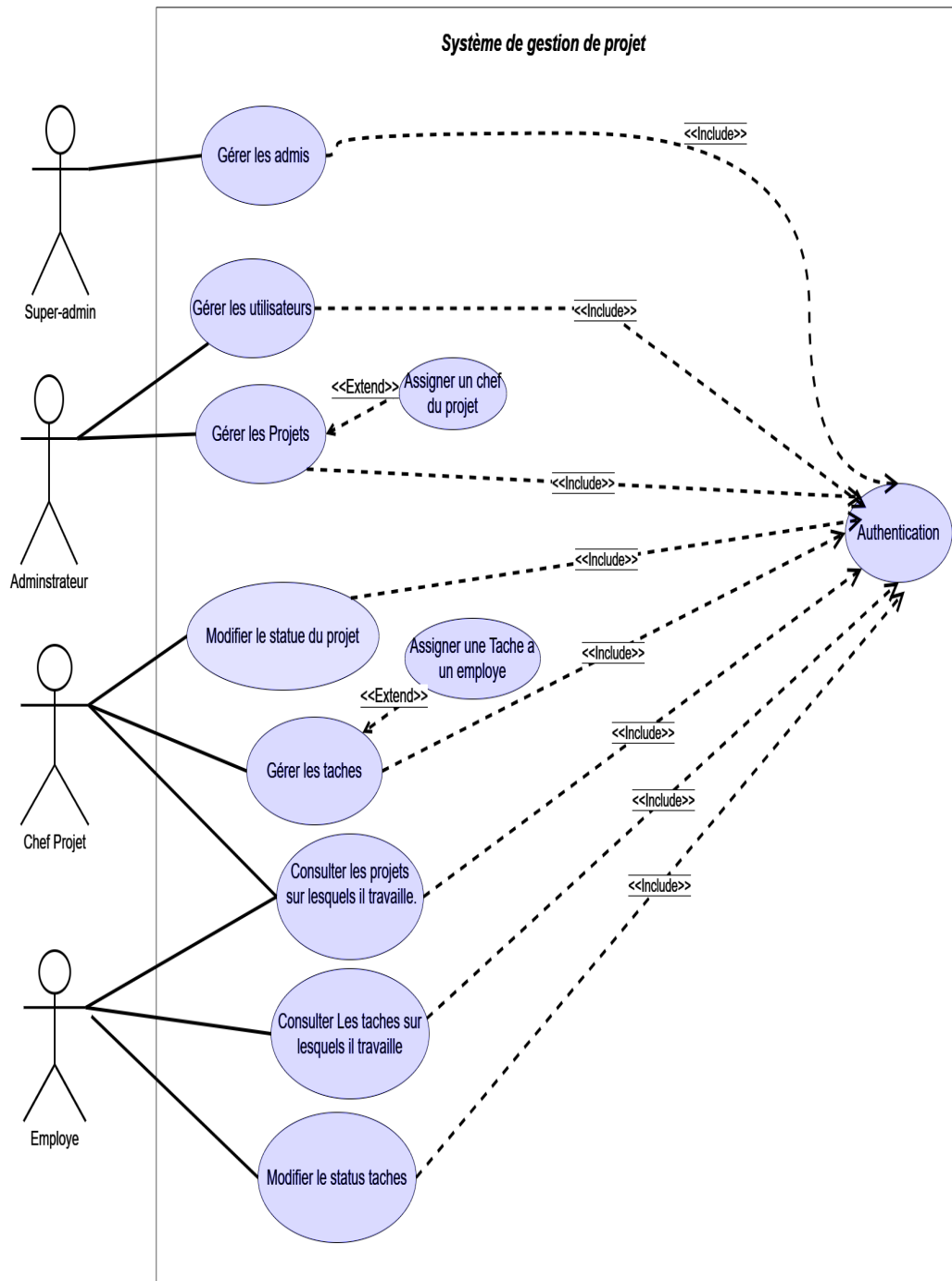


FIGURE 2.1 – diagramme de cas d'utilisation globale

## 2.4 Diagramme de classes

Le figure ci dessous dévoile le schéma de classes (diagramme classe), qui est une représentation graphique des classes, interfaces et de leurs relations au sein d'un système.

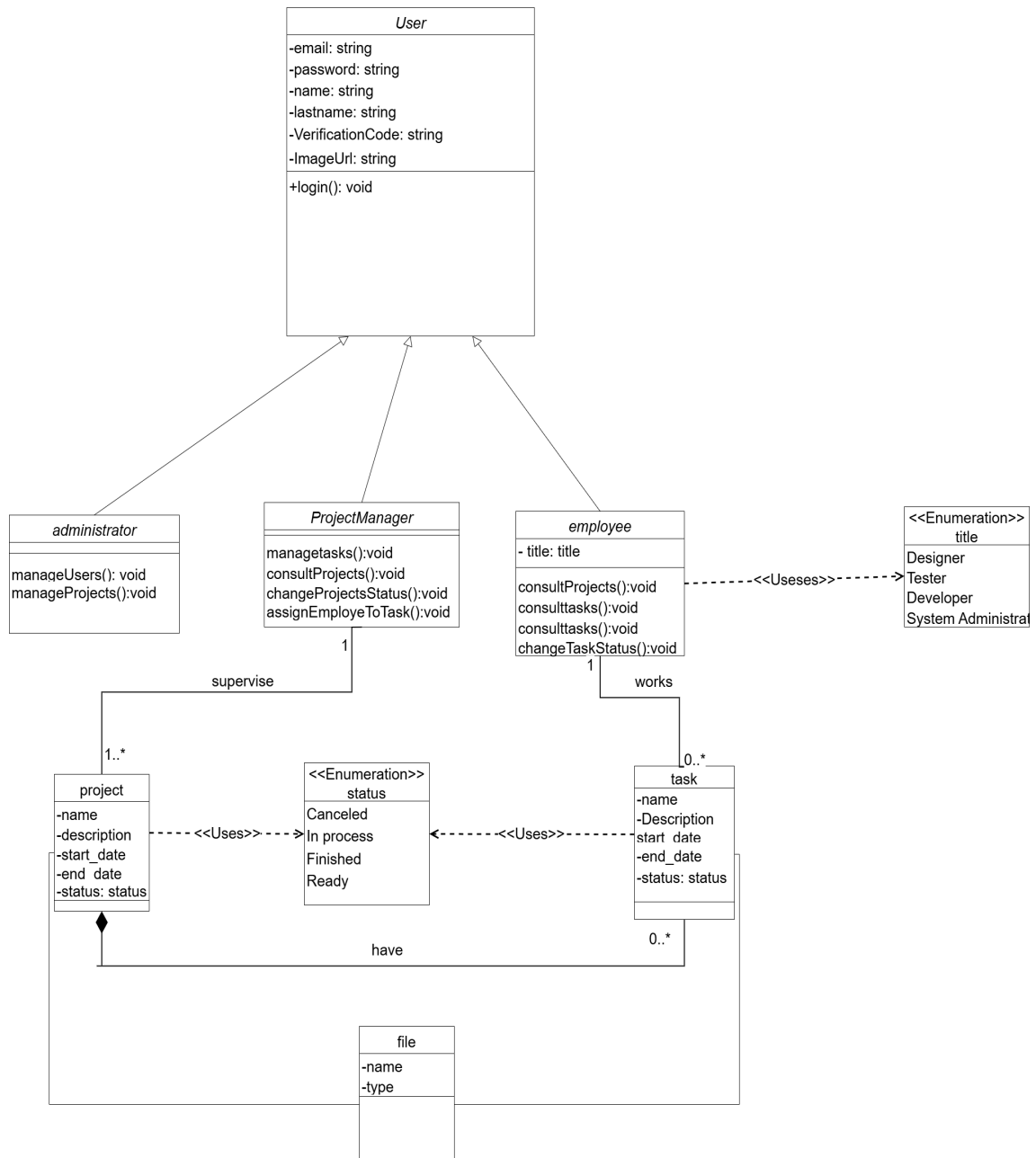


FIGURE 2.2 – diagramme de classes

## 2.5 Pilotage du Projet avec Scrum

La réussite du projet est tributaire de sa bonne planification qui doit être orientée vers l'objectif et établie sur la base du plan d'exécution du projet. La planification doit être réalisée soigneusement, progressivement et conformément aux possibilités et des ressources. La planification n'est pas un processus statique mais plutôt dynamique. L'apparition de nouveaux éléments tout au long des étapes successives peuvent souvent porter à modifier la planification et les choix retenus en début de projet.

Cette partie est réservée à la gestion du notre projet Scrum, en effet nous allons définir l'équipe du projet, le Backlog produit et le découpage en sprint.

### 2.5.1 Équipe du projet Scrum

Pour notre application les rôles sont repartis comme suit :

Tableau 2.2 – Acteurs et Rôles

Rôle Scrum	Personne Affecté
Product Owner	TEK-UP University
Scrum Master	Khaoula Tbarki
Scrum Team	Jihed Kdidi Nour El Houda Chouket Baheddine Fatnassi Khalil Smairi

### 2.5.2 Product backlog

Le backlog centralise toutes les exigences du client que l'équipe de projet doit satisfaire. Il contient la liste des fonctionnalités nécessaires à la création du produit et les éléments nécessitant l'attention de l'équipe. Chaque élément est priorisé **Priority**, indiquant l'ordre de réalisation. Chaque **user story** possède un identifiant unique **ID**, une description concise de la fonctionnalité souhaitée par le client, un résumé bref, ainsi qu'une estimation de l'effort requis pour sa réalisation, exprimée en **story points**, permettant de quantifier la complexité et le temps de travail requis.

Tableau 2.3 – Product Backlog

ID	Feature	User Story	P	SP
1	Authentification	En tant qu'utilisateur, je dois m'authentifier pour accéder aux fonctionnalités	1	5
		En tant qu'utilisateur, je dois entrer un code après le login que je vais recevoir par email.	3	3
2	Gestion des Utilisateurs	En tant qu'administrateur, je peux créer des utilisateurs	1	3
		En tant qu'administrateur, je peux modifier des utilisateurs	1	3
		En tant qu'administrateur, je peux supprimer des utilisateurs	1	3
3	Gestion des Projets	En tant que administrateur, je peux créer des projets	2	4
		En tant que administrateur, je peux modifier des projets	2	4
		En tant que administrateur, je peux supprimer des projets	2	4
		En tant qu'administrateur, je peux assigner un chef de projet au projet.	2	4
		En tant que chef de projet, je peux consulter les projets dont je suis responsable.	2	4
		En tant qu'employé, je peux consulter les projets sur lesquels j'ai des tâches.	2	4
		En tant que chef de projet, je peux changer le statut du projet	2	4
4	Gestion des Tâches	En tant que chef de projet, je peux créer des tâches	2	5
		En tant que chef de projet, je peux modifier des tâches	2	5
		En tant que chef de projet, je peux supprimer des tâches	2	5
		En tant que chef de projet, je peux assigner des tâches aux employés	2	5
		En tant qu'employé, je peux consulter les tâches sur lesquelles je travaille.	2	5
		En tant qu'employé, je peux changer le statut de la tâche	2	5



ID	Feature	User Story	P	SP
5	Mise en place d'un tableau de Bord	En tant qu'administrateur, je peux consulter un tableau de bord pour suivre l'avancement des projets et tâches	3	4
6	Notifications par Email	En tant qu'utilisateur, je reçois des notifications par email pour les événements clés (création ou modification de tâches et projets)	4	3
7	Déploiement et mise en place de l'environnement	En tant qu'ingénieur devops, je souhaite configurer l'ensemble des dépendances de l'application, établir les variables d'environnement, gérer les images Docker	4	3
		En tant qu'ingénieur devops, je veux configurer un cluster Kubernetes et garantir la sécurité l'isolation de l'infrastructure	4	5
		En tant qu'ingénieur d'exploitation, je souhaite tester l'environnement, surveiller les logs en temps réel, et identifier rapidement les éventuels problèmes pour assurer la stabilité et la performance de l'application	4	2

## 2.6 Planification des Sprints

La figure ci dessous c'est une représentation graphique claire et concise de la planification des sprints, mettant en évidence les étapes spécifiques et les durées prévues pour chaque sprint.

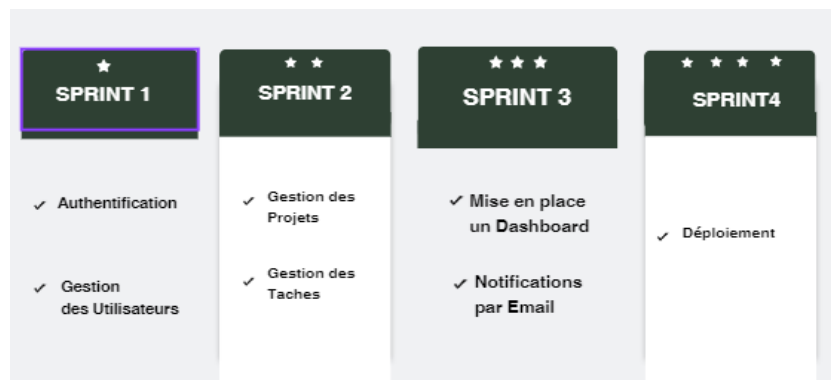


FIGURE 2.3 – Planification des sprints

## Chapitre 3

### Étude et réalisation du release 1

## Introduction

Ce chapitre présente le développement du Release 1 du projet de gestion des projets d'une entreprise.

Le Release 1 a pour objectif d'intégrer des fonctionnalités essentielles pour l'authentification, la gestion des utilisateurs, ainsi que la gestion des projets et des tâches.

Le Release 1 est divisé en deux sprints :

le Sprint 1, qui se concentre sur l'authentification et la gestion des utilisateurs.

le Sprint 2, qui introduit la gestion des projets et des tâches.

### 3.1 Sprint 1

Le Sprint 1 de notre projet se concentre sur l'authentification et la gestion des utilisateurs, deux éléments clés pour assurer la sécurité et l'intégrité de l'application. Cette section explore les mécanismes d'authentification mis en place et la gestion des utilisateurs.

#### 3.1.1 Backlog Sprint1

Le tableau du backlog du Sprint 1 résume les besoins en authentification et gestion des utilisateurs

Tableau 3.1 – Sprint 1 : Authentification et Gestion des Utilisateurs

ID	Fonctionnalité	User Story	Tâches
1	Authentification	En tant qu'Utilisateur, je veux pouvoir m'authentifier.	Créer le modèle utilisateur Créer la vue d'authentification Créer le contrôleur d'authentification Créer la fonction de connexion Créer la route de connexion Tester la fonctionnalité de connexion

ID	Fonctionnalité	User Story	Tâches
2	<b>Gestion des utilisateurs</b>	En tant qu'Administrateur, je veux pouvoir créer de nouveaux utilisateurs.	Créer la vue de création d'utilisateur Créer le contrôleur de création d'utilisateur Créer la fonction d'ajout d'utilisateur Créer la route de création d'utilisateur Tester la fonctionnalité de création
		En tant qu'Administrateur, je veux pouvoir modifier les informations des utilisateurs.	Créer la vue de modification d'utilisateur Créer le contrôleur de modification Créer la fonction de mise à jour d'utilisateur Créer la route de modification Tester la fonctionnalité de modification
		En tant qu'Administrateur, je veux pouvoir supprimer des utilisateurs.	Créer la vue de suppression d'utilisateur Créer le contrôleur de suppression Créer la fonction de suppression d'utilisateur Créer la route de suppression Tester la fonctionnalité de suppression

### 3.1.2 Analyse et spécification fonctionnelle

Dans cette section, nous procédons à l'analyse et à la spécification fonctionnelle de ce sprint

### 3.1.3 Diagramme de cas d'utilisation du sprint1

Il est important de noter que ce diagramme de cas d'utilisation est destiné à être utilisé pour planifier les fonctionnalités à développer pour le sprint 1. En décrivant les fonctionnalités attendues et les actions que les utilisateurs peuvent effectuer, le diagramme de cas d'utilisation aide l'équipe de développement à comprendre les tâches à accomplir et les dépendances entre elles, comme illustré dans la figure

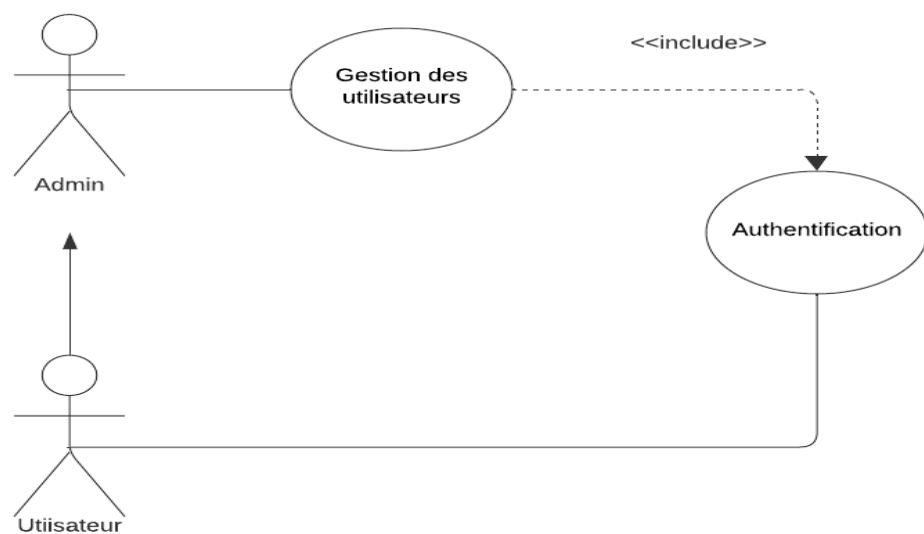


FIGURE 3.1 – Diagramme de cas d'utilisation du sprint1

### 3.1.4 Raffinement des cas d'utilisation Authentification

La section Raffinement des cas d'utilisation "Authentification" se concentre sur les fonctionnalités liées à l'authentification dans notre système, comme illustré dans la figure

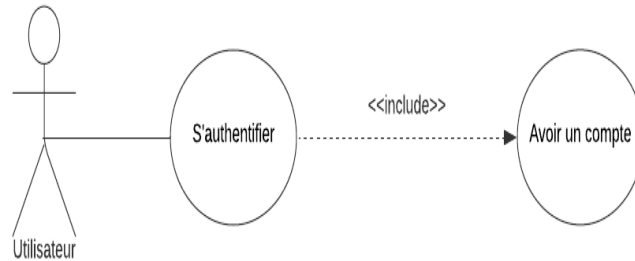


FIGURE 3.2 – : Raffinement CU «Authentification»

### 3.1.5 Description textuelle du CU «Authentification»

Tableau 3.2 – Cas d'utilisation : Authentification

<b>Titre</b>	Ajouter un domaine
<b>Acteurs</b>	Utilisateur
<b>Objectif</b>	Lors de l'accès au site, l'utilisateur doit se connecter pour accéder au site.
<b>Description</b>	Lorsqu'un utilisateur du système veut accéder à l'application, il doit saisir son login et son mot de passe : ensuite le système vérifie s'ils sont corrects ou pas afin d'autoriser ou bien refuser l'accès.
<b>Préconditions</b>	L'utilisateur doit être créé dans la base de données et connaître ses identifiants.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système affiche le formulaire d'identification.</li> <li>2. L'internaute remplit le formulaire avec l'ensemble des informations nécessaires à son identification.</li> <li>3. Le système vérifie les informations saisies par l'utilisateur et renvoie vers la page d'accueil en affichant sa catégorie.</li> </ol>
<b>Scénario d'exception</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur n'a pas saisi les bons identifiants.</li> <li>2. Le système renvoie un message d'erreur et signale à l'utilisateur de recommencer.</li> <li>3. L'utilisateur n'existe pas dans la base de données.</li> </ol>
<b>Post-conditions</b>	Utilisateur authentifié.

### 3.1.6 Diagramme de séquences du CU «Authentification»

Le diagramme de séquence est un outil de modélisation UML (Unified Modeling Language) utilisé pour décrire la façon dont les objets interagissent dans un système ou une application.

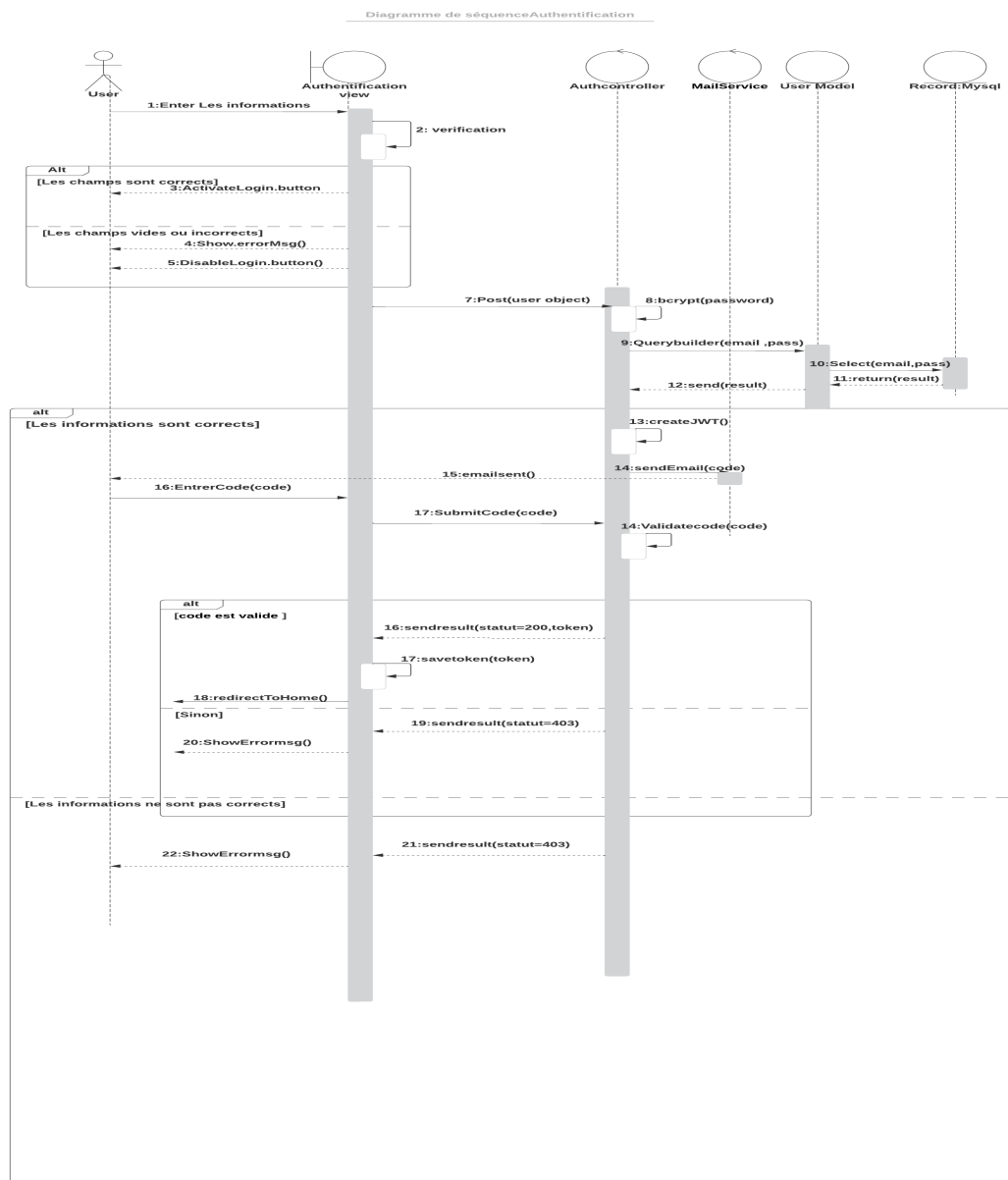


FIGURE 3.3 – : Raffinement CU «Authentification»

### 3.1.7 Raffinement des cas d'utilisation «Gestion des Utilisateurs»

La section Raffinement des cas d'utilisation "Gestion des utilisateurs" se concentre sur les fonctionnalités liées à la gestion des utilisateurs dans notre système, comme illustré dans la figure :

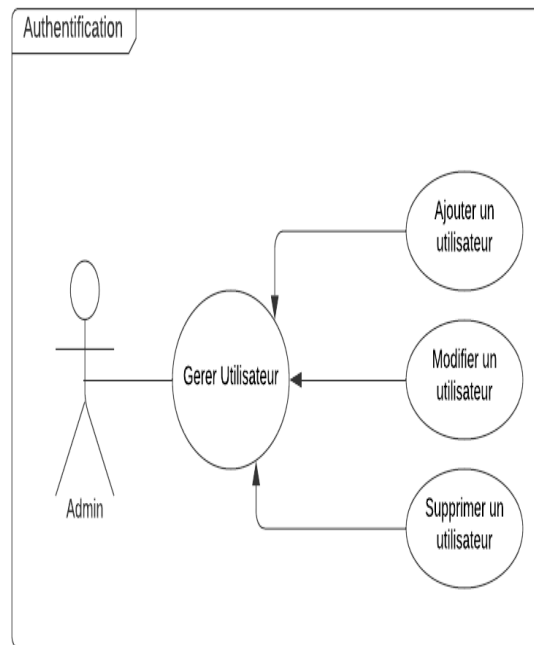


FIGURE 3.4 – : Raffinement CU «Gestion des utilisateurs»



### 3.1.8 Description textuelle du CU «Ajouter Utilisateur»

Tableau 3.3 – Cas d'utilisation : Ajouter un utilisateur

<b>Titre</b>	Ajouter un utilisateur
<b>Acteur</b>	Admin
<b>Objectif</b>	Ajouter un utilisateur selon sa position.
<b>Description</b>	L'administrateur peut ajouter de nouveaux utilisateurs en remplissant un formulaire avec des informations nécessaires comme le nom, l'email, le rôle, etc.
<b>Préconditions</b>	<ul style="list-style-type: none"><li>— L'administrateur est authentifié.</li><li>— L'administrateur a les droits de gestion des utilisateurs.</li></ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"><li>1. L'administrateur accède à la gestion des utilisateurs.</li><li>2. Il clique sur "Ajouter".</li><li>3. Il remplit le formulaire (nom, email, rôle, etc.).</li><li>4. Il soumet le formulaire.</li><li>5. Le système ajoute l'utilisateur et met à jour la liste des utilisateurs.</li></ol>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"><li>— L'email existe déjà dans le système (message d'erreur).</li><li>— Le formulaire est incomplet (message d'erreur demandant de remplir tous les champs obligatoires).</li></ul>
<b>Post-condition</b>	L'utilisateur est ajouté à la base de données et visible dans la liste des utilisateurs.

### 3.1.9 Diagramme de séquences du CU «Ajouter utilisateur»

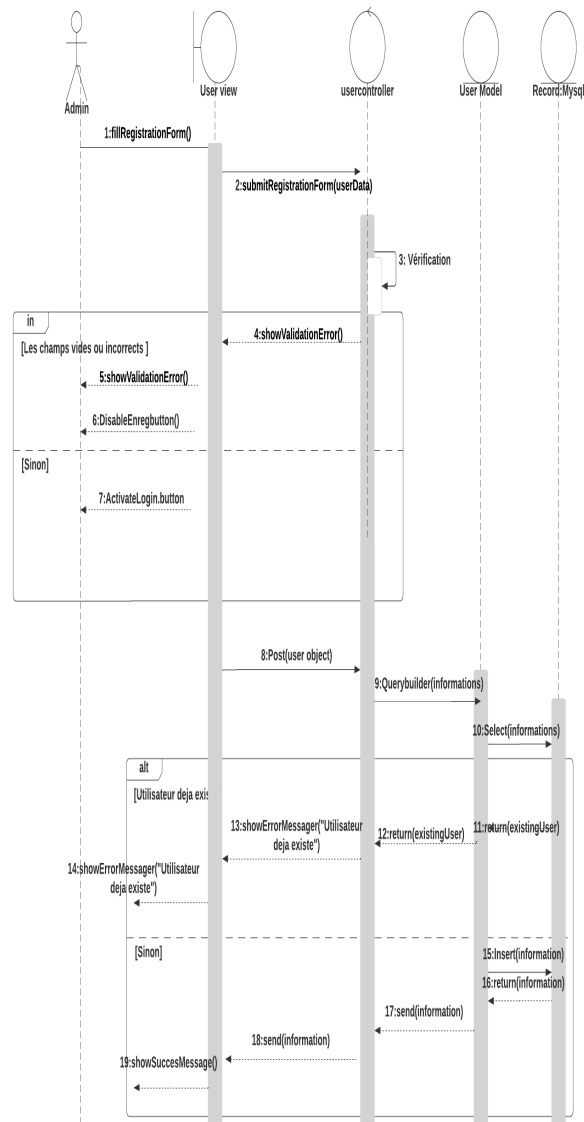


FIGURE 3.5 – : Raffinement CU «Ajouter utilisateur»

### 3.1.10 Description textuelle du CU «Modifier Utilisateur»

Cette partie suit le même principe que la description textuelle du cas d'utilisation «Ajouter Utilisateur» et en reprend les principales similitudes.

### 3.1.11 Diagramme de séquences du CU «Modifier utilisateur»

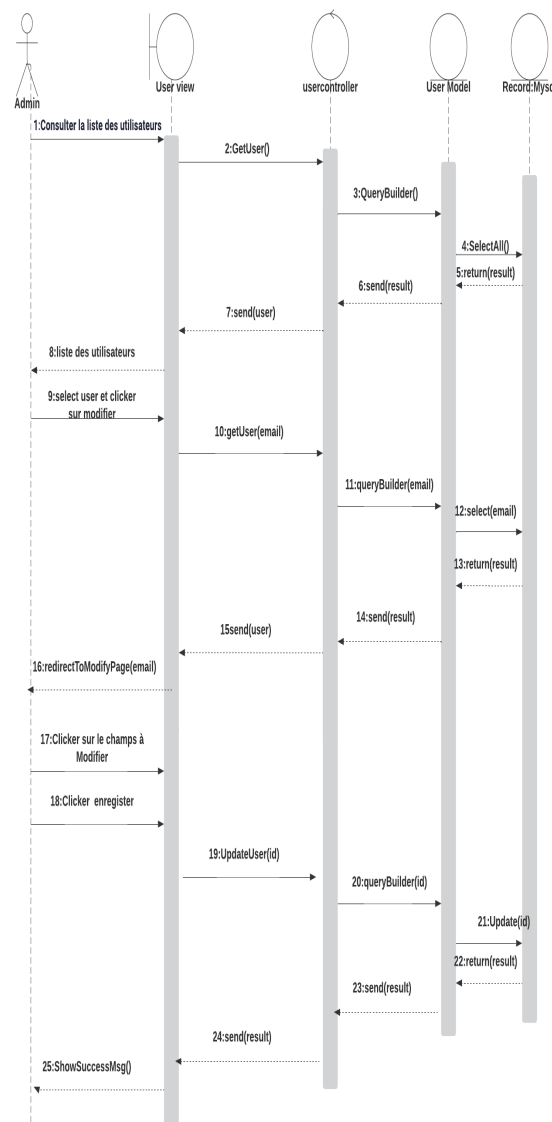


FIGURE 3.6 – : Raffinement CU «Modifier utilisateur»

### 3.1.12 Description textuelle du CU «Supprimer utilisateur»

Cette partie suit le même principe que la description textuelle du cas d'utilisation «Ajouter Utilisateur» et en reprend les principales similitudes.

### 3.1.13 Diagramme de séquences du CU «Supprimer utilisateur»

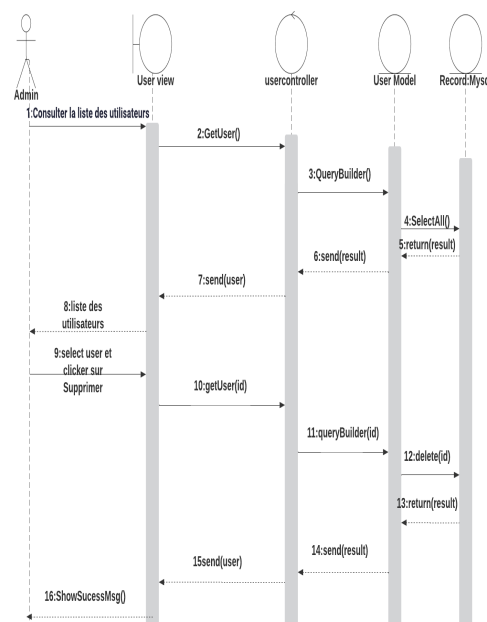


FIGURE 3.7 – : Raffinement CU «Supprimer utilisateur»

### 3.1.14 Diagramme de classes du sprint1

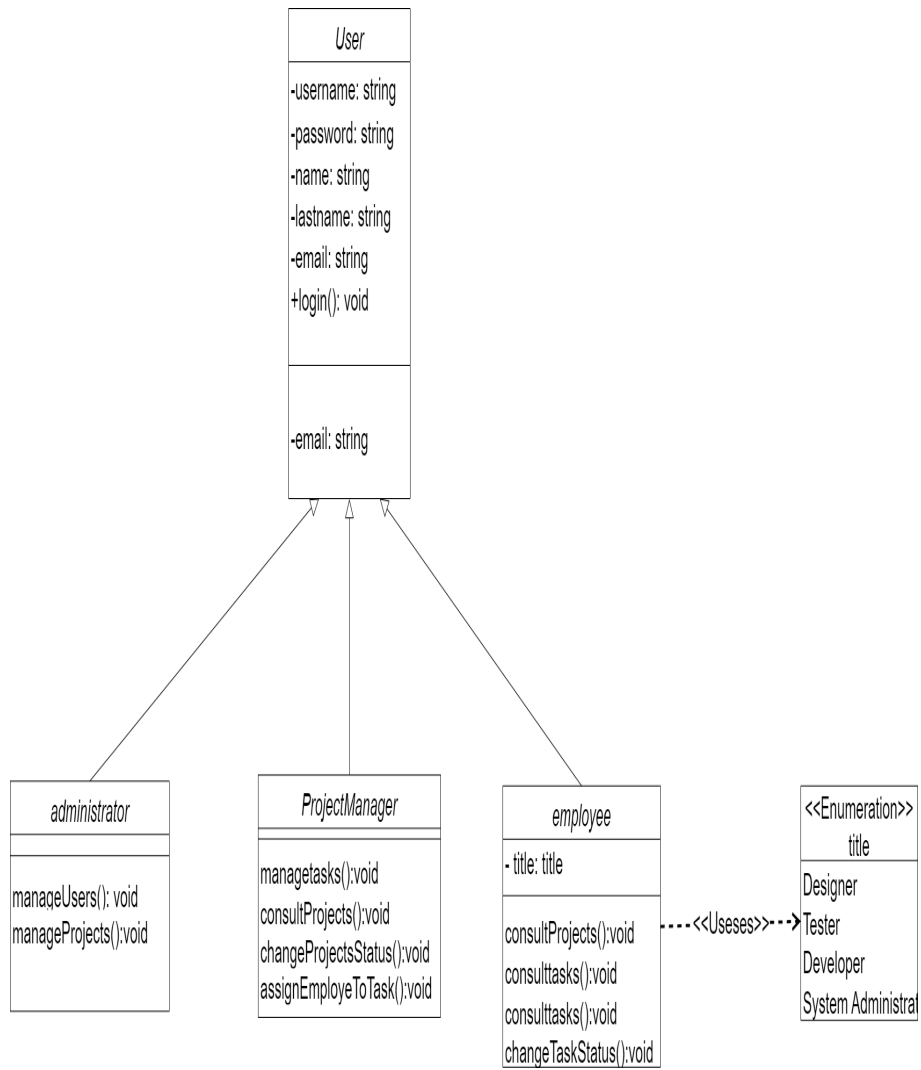


FIGURE 3.8 – : Diagramme de classes du sprint1

### 3.1.15 Réalisation

---

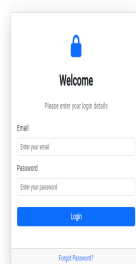


FIGURE 3.9 – : Capture d’écran de l’interface de connexion

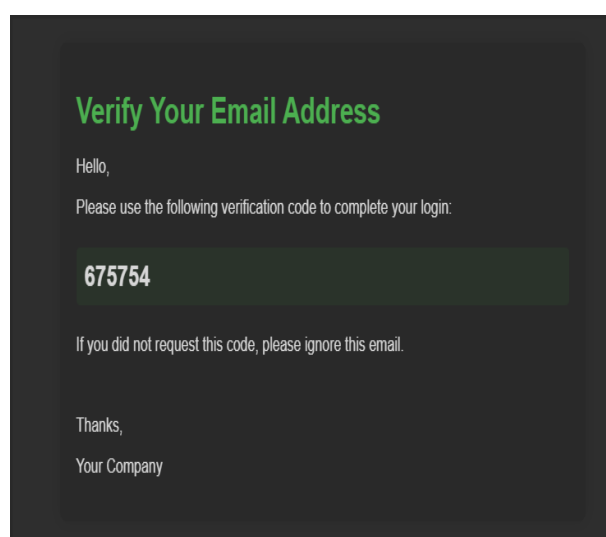


FIGURE 3.10 – : Capture d’écran de l’email de vérification

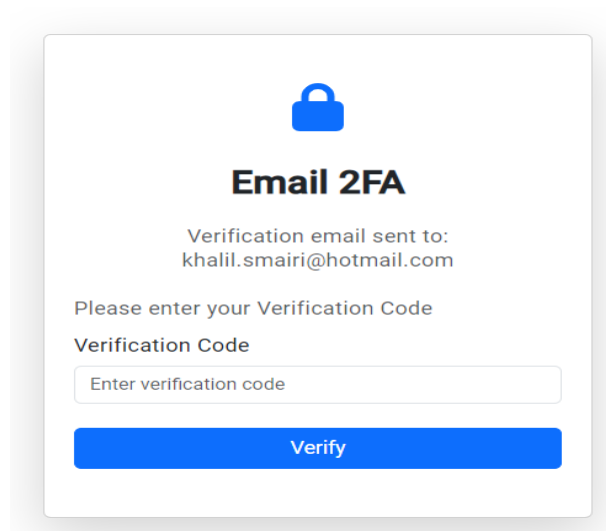


FIGURE 3.11 – : Capture d’écran de l’interface de vérification par mail

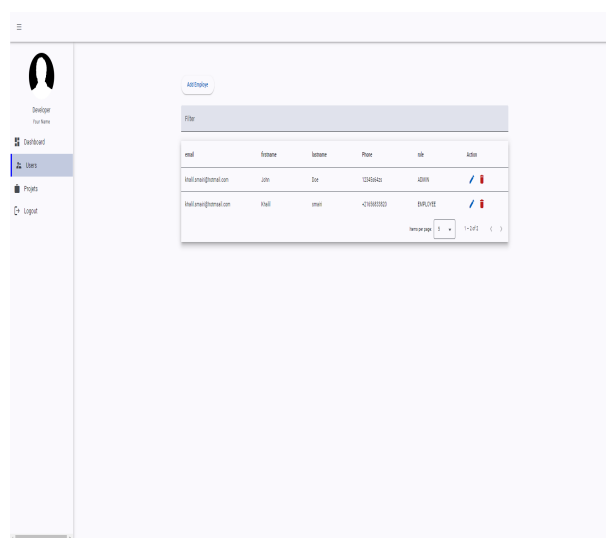


FIGURE 3.12 – : Capture d’écran de la liste des utilisateurs

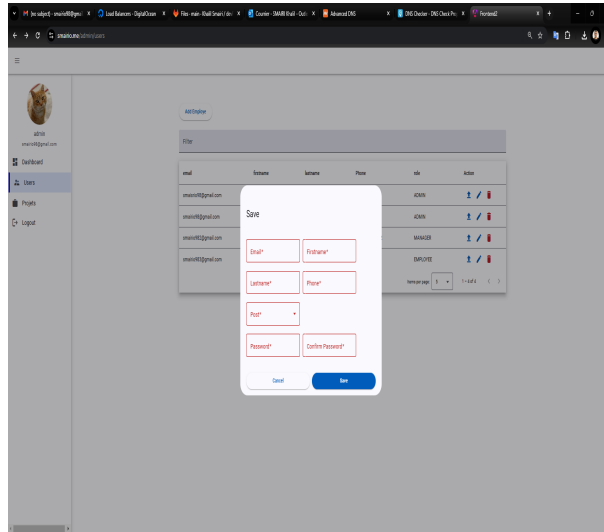


FIGURE 3.13 – : Capture d'écran de l'interface d'ajout d'utilisateur

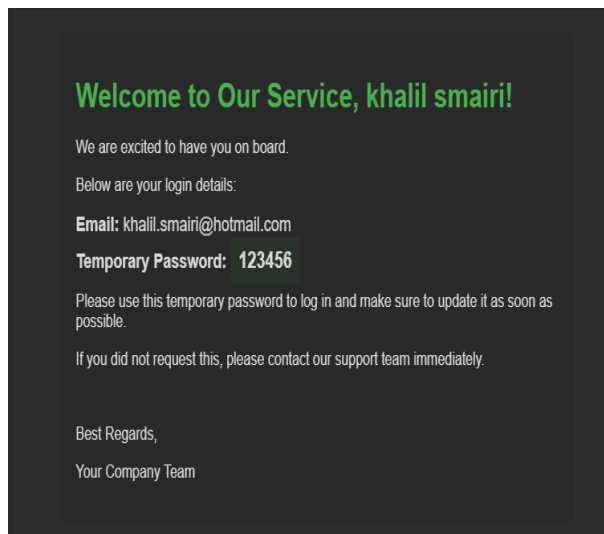


FIGURE 3.14 – : Capture d'écran de l'email de bienvenue



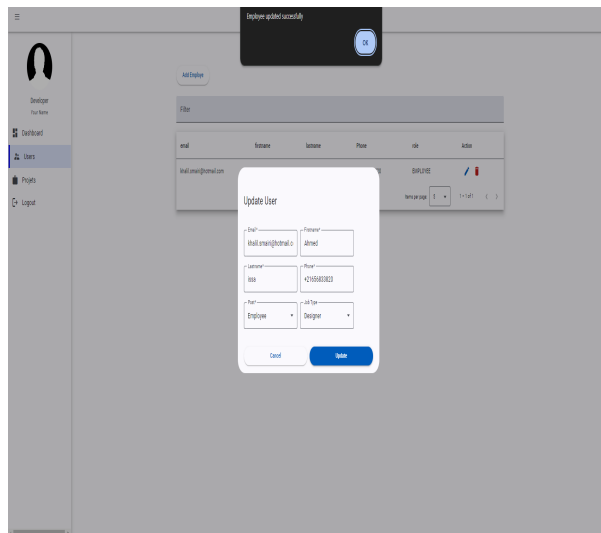


FIGURE 3.15 – : Capture d'écran de l'interface de modification d'utilisateur

## 3.2 Sprint 2

Le Sprint 2 porte sur la gestion des projets et des tâches, incluant leur création, modification, assignation, et suivi, pour assurer une organisation et un avancement optimaux.

### 3.2.1 Backlog Sprint2

Le tableau du backlog du Sprint 2 résume les besoins en gestion de projet et gestion des tâches :

Tableau 3.4 – Sprint 2 : Gestion des Projets et des Tâches

ID	Fonctionnalité	User Story	Tâches
3	Gestion des Projets	En tant qu'administrateur, je peux créer des projets.	Créer la vue de création de projet Créer le contrôleur de création de projet Créer la fonction d'ajout de projet Créer la route de création de projet Tester la fonctionnalité de création de projet
		En tant qu'administrateur, je peux modifier des projets.	Créer la vue de modification de projet Créer le contrôleur de modification Créer la fonction de mise à jour de projet Créer la route de modification de projet Tester la fonctionnalité de modification de projet
		En tant qu'administrateur, je peux supprimer des projets.	Créer la vue de suppression de projet Créer le contrôleur de suppression de projet Créer la fonction de suppression de projet Créer la route de suppression de projet Tester la fonctionnalité de suppression de projet
		En tant qu'administrateur, je peux assigner un chef de projet au projet.	Créer la vue d'assignation d'un chef de projet Créer le contrôleur d'assignation Créer la fonction d'assignation du chef de projet Créer la route d'assignation de chef de projet Tester la fonctionnalité d'assignation

ID	Fonctionnalité	User Story	Tâches
		En tant que chef de projet, je peux consulter les projets dont je suis responsable.	Créer la vue des projets assignés au chef de projet Créer le contrôleur pour consulter les projets Créer la fonction de consultation des projets Créer la route pour consulter les projets Tester la fonctionnalité de consultation des projets
		En tant qu'employé, je peux consulter les projets sur lesquels j'ai des tâches.	Créer la vue des projets assignés à l'employé Créer le contrôleur de consultation des projets pour l'employé Créer la fonction de consultation des projets pour l'employé Créer la route pour consulter les projets de l'employé Tester la fonctionnalité de consultation des projets
		En tant que chef de projet, je peux changer le statut du projet.	Créer la vue pour changer le statut du projet Créer le contrôleur pour modifier le statut Créer la fonction de modification du statut Créer la route de modification du statut Tester la fonctionnalité de changement de statut
4	<b>Gestion des Tâches</b>	En tant que chef de projet, je peux créer des tâches.	Créer la vue de création de tâche Créer le contrôleur de création de tâche Créer la fonction d'ajout de tâche Créer la route de création de tâche Tester la fonctionnalité de création de tâche
		En tant que chef de projet, je peux modifier des tâches.	Créer la vue de modification de tâche Créer le contrôleur de modification de tâche Créer la fonction de mise à jour de tâche Créer la route de modification de tâche Tester la fonctionnalité de modification de tâche

ID	Fonctionnalité	User Story	Tâches
		En tant que chef de projet, je peux supprimer des tâches.	Créer la vue de suppression de tâche Créer le contrôleur de suppression de tâche Créer la fonction de suppression de tâche Créer la route de suppression de tâche Tester la fonctionnalité de suppression de tâche
		En tant que chef de projet, je peux assigner des tâches aux employés.	Créer la vue d'assignation de tâches Créer le contrôleur d'assignation Créer la fonction d'assignation des tâches aux employés Créer la route d'assignation de tâches Tester la fonctionnalité d'assignation de tâches
		En tant qu'employé, je peux consulter les tâches sur lesquelles je travaille.	Créer la vue des tâches assignées à l'employé Créer le contrôleur de consultation des tâches Créer la fonction de consultation des tâches Créer la route pour consulter les tâches de l'employé Tester la fonctionnalité de consultation des tâches
		En tant qu'employé, je peux changer le statut de la tâche.	Créer la vue pour changer le statut de la tâche Créer le contrôleur de modification du statut Créer la fonction de changement de statut de la tâche Créer la route de changement de statut de la tâche Tester la fonctionnalité de changement de statut

### 3.2.2 Analyse et specification fonctionnelle

Dans cette section, nous procédons à l'analyse et à la spécification fonctionnelle de ce sprint

### 3.2.3 Diagramme de cas d'utilisation du sprint2

Ce diagramme de cas d'utilisation pour le sprint 2 aide à planifier les fonctionnalités à développer. Il décrit les actions que les utilisateurs peuvent

effectuer, clarifiant ainsi les tâches à réaliser et les relations entre elles, comme montré dans la figure :

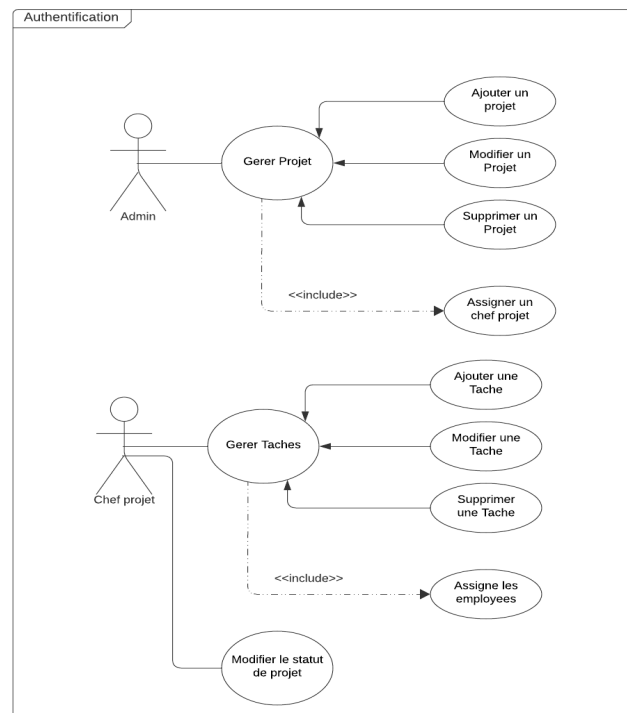


FIGURE 3.16 – Diagramme de cas d utilisation du sprint 2

### 3.2.4 Diagramme de séquence du CU «AjouterProject»

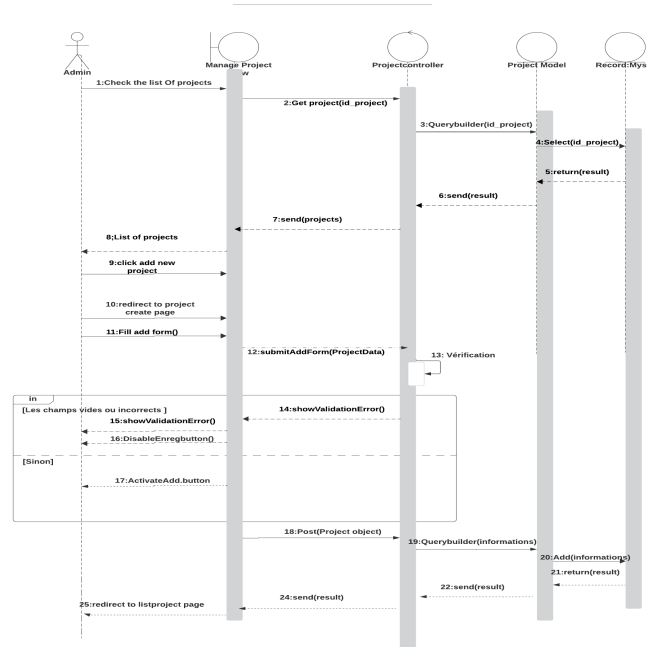


FIGURE 3.17 – : Raffinement CU «AjouterProject»

### 3.2.5 Diagramme de séquence du CU «Modifier Project»

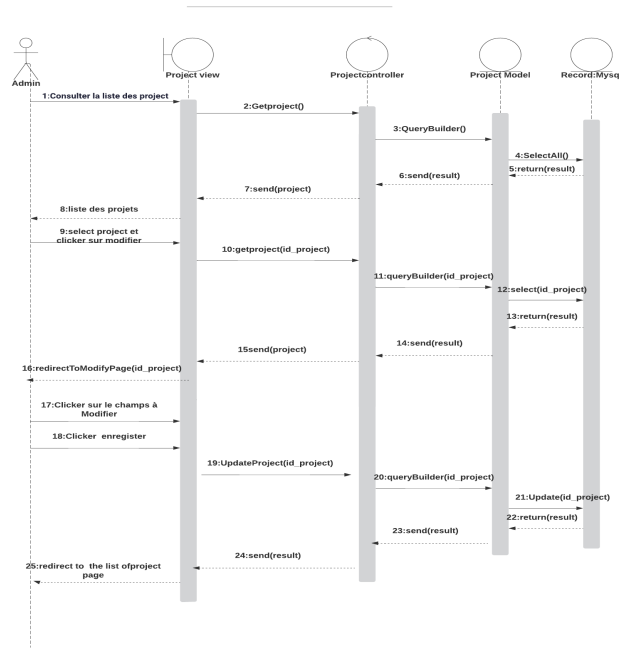


FIGURE 3.18 – : Raffinement CU «ModifierProject»

### 3.2.6 Diagramme de séquence du CU «Supprimer Project»

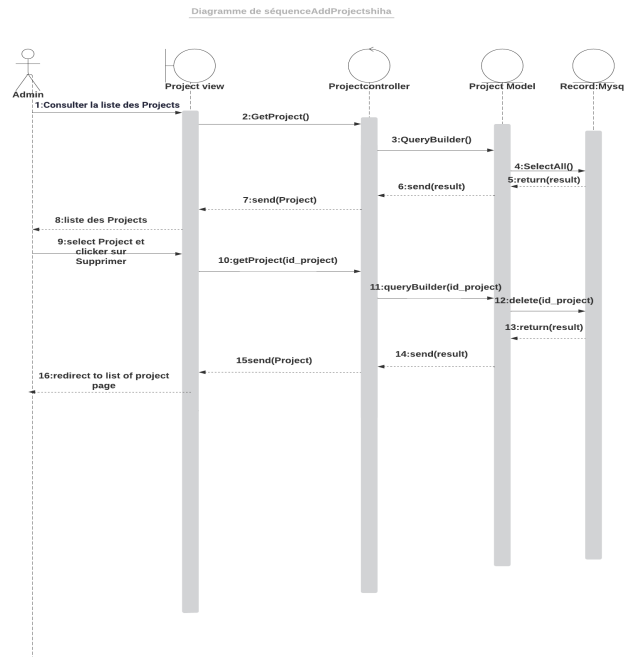


FIGURE 3.19 – : Raffinement CU «SupprimerProject»

### 3.2.7 Les diagramme de CU relation au task

Cette partie suit le même principe que la gestion des projets et en reprend les principales similitudes.



### 3.2.8 Realisation

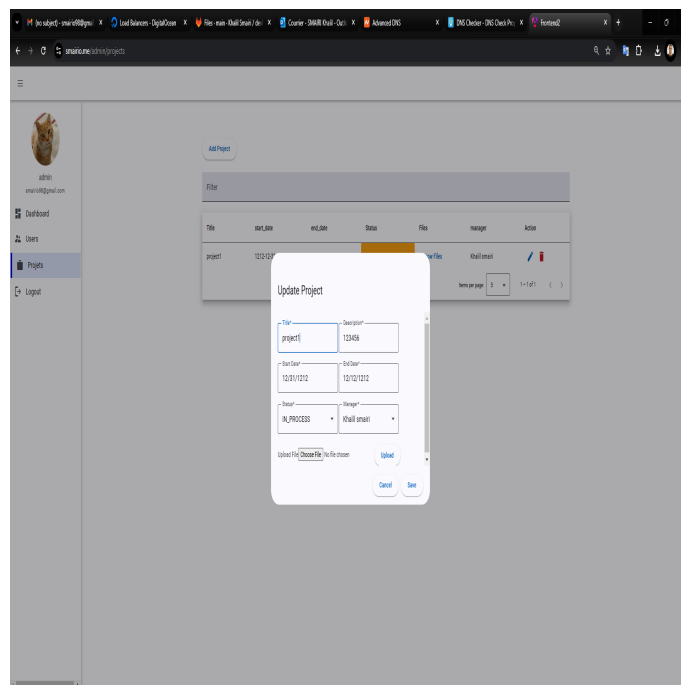


FIGURE 3.20 – «Modifier Project»

## Chapitre 4

### Étude et réalisation du release 2

## 4.1 Introduction

Ce chapitre présente le développement du Release 2 du projet de gestion des projets d'une entreprise, qui comprend :

Sprint 3 : conception d'un tableau de bord dynamique en temps réel.

Sprint 4 : déploiement de l'application sur Kubernetes.

## 4.2 Sprint 3 : Tableau de bord dynamique

Le tableau de bord affiche des indicateurs clés, notamment le nombre total de tâches, les tâches terminées, prêtes, en cours, et annulées, ainsi que des données similaires pour les projets. Chaque utilisateur (employé, administrateur, chef de projet) dispose d'une vue personnalisée adaptée à ses responsabilités.

### 4.2.1 Backlog sprint 3

Tableau 4.1 – Sprint 3 : Tableau de Bord et Notifications par Email

ID	Fonctionnalité	User Story	Tâches
5	Mise en place d'un Tableau de Bord	En tant qu'administrateur, je peux consulter un tableau de bord pour suivre l'avancement des projets et tâches.	Créer la vue du tableau de bord Créer le contrôleur du tableau de bord Récupérer les données d'avancement des projets et tâches Créer des graphiques et indicateurs pour l'avancement Intégrer les données dans le tableau de bord Tester la fonctionnalité du tableau de bord

ID	Fonctionnalité	User Story	Tâches
6	<b>Notifications par Email</b>	En tant qu'utilisateur, je reçois des notifications par email pour les événements clés (création ou modification de tâches et projets).	Créer un service de gestion des notifications Configurer l'envoi d'emails pour les événements clés Créer des templates d'email pour les notifications Implémenter la logique pour l'envoi d'emails sur modification ou création de tâche/projet Tester l'envoi de notifications par email

## 4.2.2 Realisation

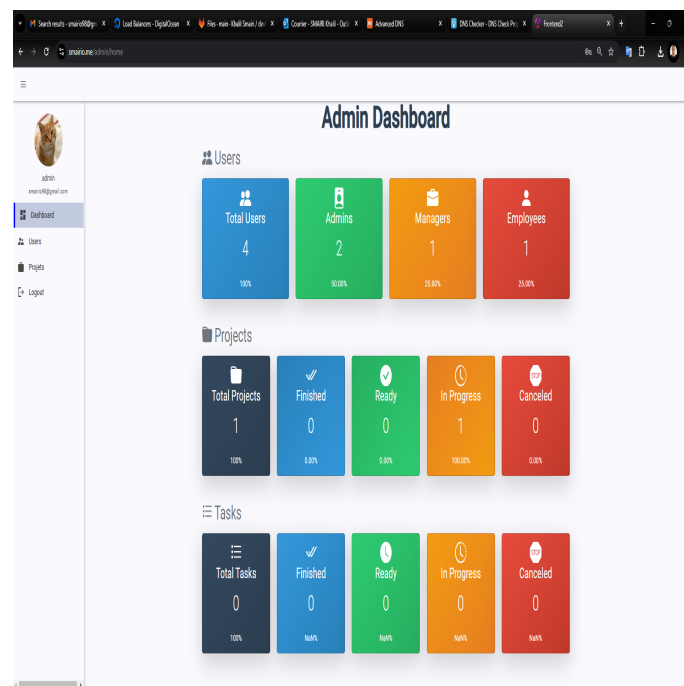


FIGURE 4.1 – Dashboard admin

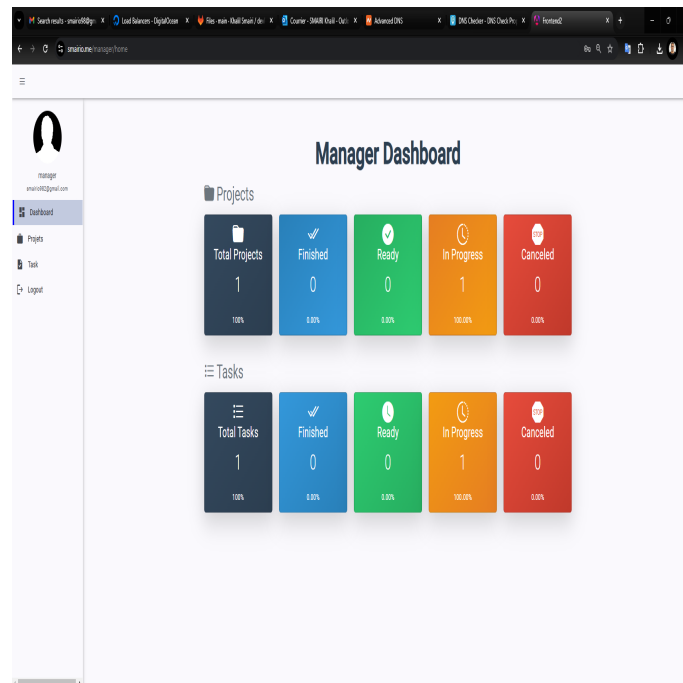


FIGURE 4.2 – Dashboard manager

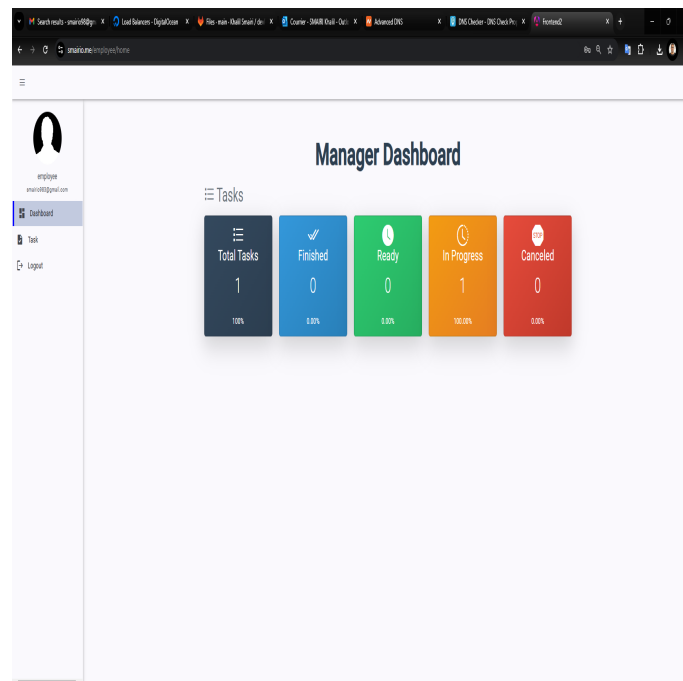


FIGURE 4.3 – Dashboard employee

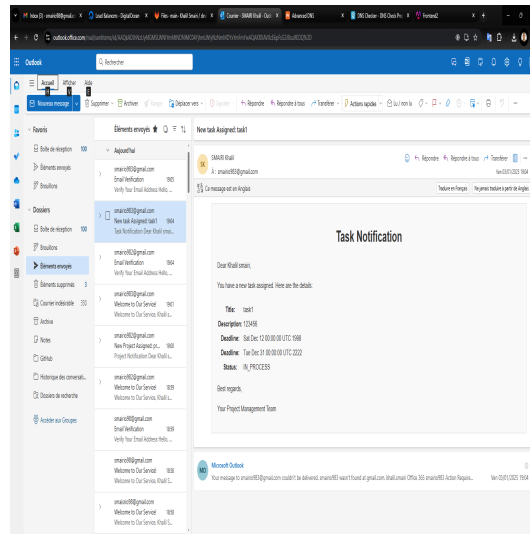


FIGURE 4.4 – La notification par e-mail lors de l’ajout d’une nouvelle tâche.

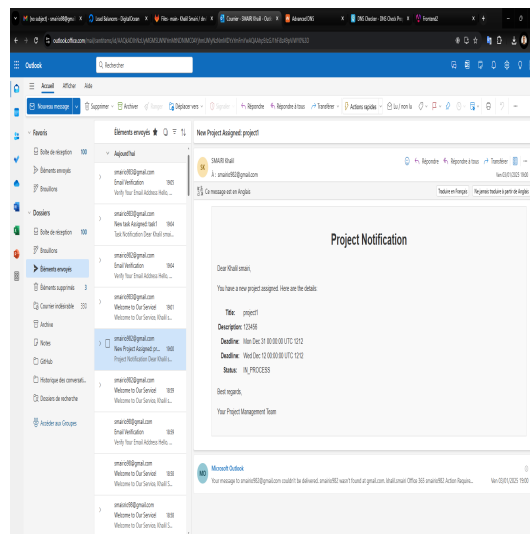


FIGURE 4.5 – La notification par e-mail lors de l’ajout d’un nouveau projet

## 4.3 Sprint 4 : Déploiement

### 4.3.1 Backlog Sprint 4

Tableau 4.2 – Sprint 4 : Déploiement et Mise en Place de l'Environnement

ID	Fonctionnalité	User Story	Tâches
7	<b>Déploiement et Mise en Place de l'Environnement</b>	En tant qu'ingénieur DevOps, je souhaite configurer l'ensemble des dépendances de l'application, établir les variables d'environnement, gérer les images Docker.	Créer et configurer les fichiers de configuration Docker Gérer les dépendances de l'application Définir les variables d'environnement pour l'application Créer et gérer les images Docker nécessaires Tester la configuration des images Docker
		En tant qu'ingénieur DevOps, je veux configurer un cluster Kubernetes et garantir la sécurité et l'isolation de l'infrastructure.	Créer et configurer le cluster Kubernetes Définir les règles de sécurité et d'accès au cluster Implémenter des mécanismes d'isolation entre les différents services Tester la sécurité et l'isolation du cluster Vérifier la scalabilité et la résilience du cluster

ID	Fonctionnalité	User Story	Tâches
		En tant qu'ingénieur d'exploitation, je souhaite tester l'environnement, surveiller les logs en temps réel, et identifier rapidement les éventuels problèmes pour assurer la stabilité et la performance de l'application.	<p>Configurer les outils de surveillance et de collecte de logs</p> <p>Mettre en place la surveillance en temps réel des logs</p> <p>Tester l'environnement et identifier les éventuels problèmes de performance</p> <p>Mettre en place des alertes pour détecter les anomalies</p> <p>Assurer la stabilité et la performance de l'application en continu</p>

### 4.3.2 Realisation

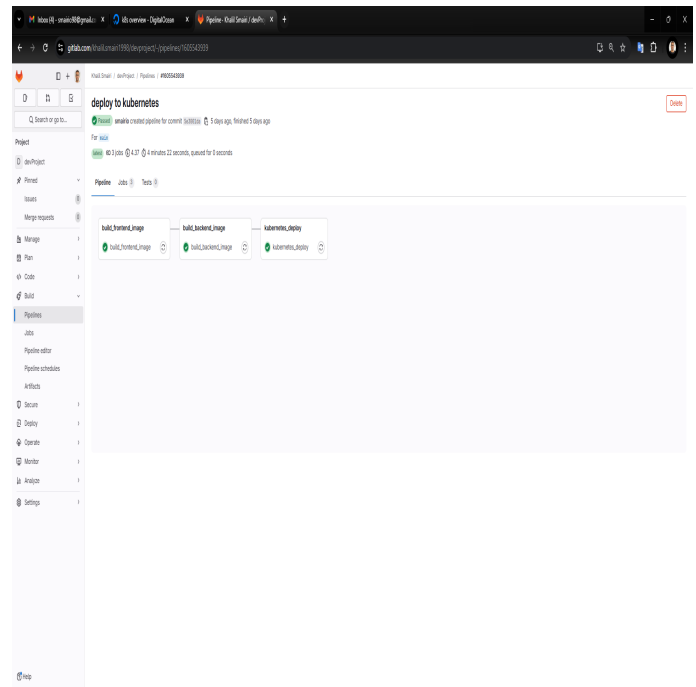


FIGURE 4.6 – pipeline



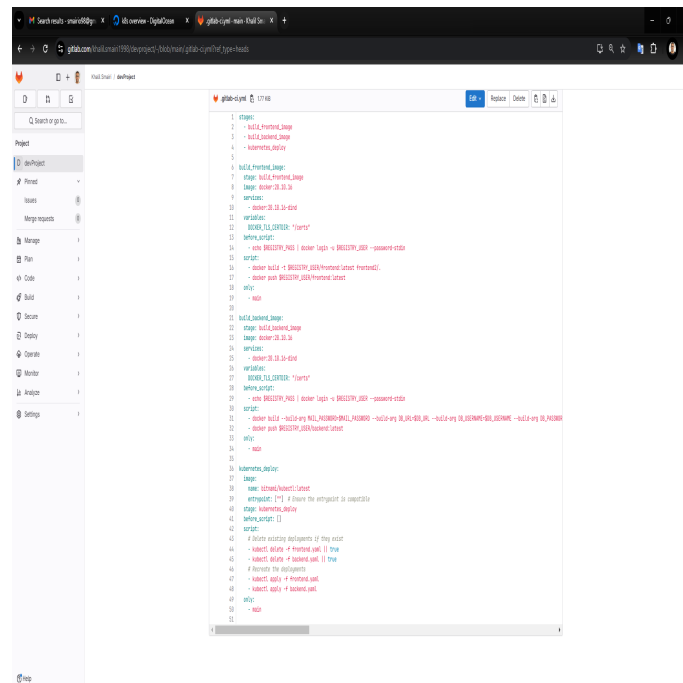


FIGURE 4.7 – Les conteneurs

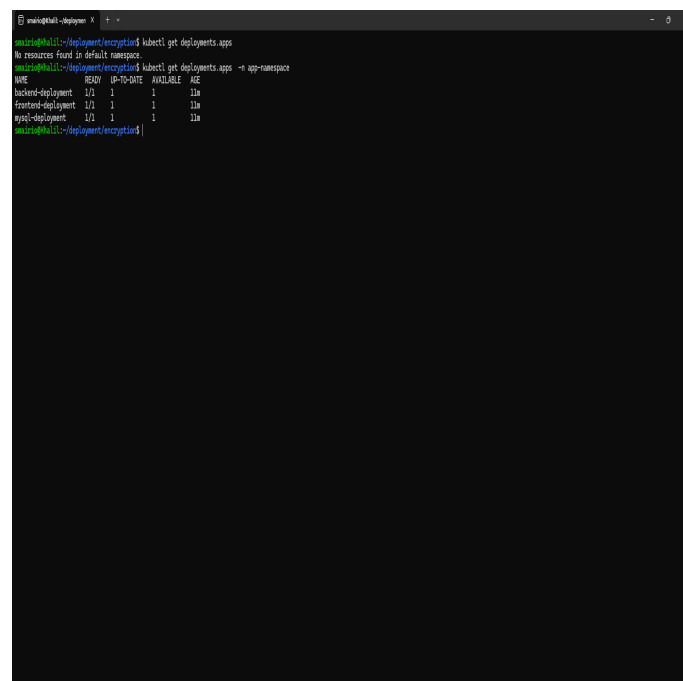


FIGURE 4.8 – Déploiement sur Kubernetes

## 4.4 Conclusion Generale

Conclusion générale sur le projet

Le projet présenté dans ce rapport vise à développer une application de gestion de projets et de tâches, offrant des fonctionnalités essentielles telles que l'authentification, la gestion des utilisateurs, des projets et des tâches, ainsi qu'un tableau de bord pour le suivi de l'avancement. L'intégration de notifications par email permet une communication en temps réel pour les événements clés.

Le projet intègre également des aspects techniques avancés, notamment la configuration d'un environnement DevOps avec Docker et Kubernetes, garantissant une gestion optimale des dépendances, la sécurité et la performance de l'application. La mise en place d'une infrastructure robuste assure la stabilité du système tout au long de son utilisation.

L'approche méthodologique agile, guidée par le Product Backlog, permet de suivre une planification précise et une exécution organisée, avec des priorités clairement définies pour chaque fonctionnalité. Cela assure une progression fluide du projet tout en répondant efficacement aux besoins des utilisateurs. Ce projet est ainsi conçu pour être flexible, évolutif et capable de s'adapter aux exigences futures de l'entreprise.