



---

# DÉDICACES



---

# REMERCIEMENT



---

# TABLE DES MATIÈRES

<b>INTRODUCTION GÉNÉRALE</b>	<b>1</b>
<b>1 Etude préalable</b>	<b>2</b>
1.1 Introduction . . . . .	3
1.2 Présentation de l'organisme d'accueil . . . . .	3
1.2.1 Présentation de « Axia solutions » . . . . .	3
1.2.2 Services de « Axia solutions » . . . . .	4
1.3 Contexte du projet . . . . .	4
1.3.1 Problématique . . . . .	4
1.3.2 Analyse de l'existant . . . . .	5
1.3.2.1 Étude de l'existant . . . . .	5
1.3.2.2 Critique de l'existant . . . . .	6
1.3.2.3 Solution proposée . . . . .	7
1.4 Méthodologie de travail . . . . .	7
1.4.1 Choix de la méthode . . . . .	9
1.4.2 Présentation de la méthodologie SCRUM . . . . .	9
1.5 Conclusion . . . . .	10
<b>2 Spécification et analyse des besoins</b>	<b>11</b>
2.1 Introduction . . . . .	12
2.2 Identification des acteurs et des besoins . . . . .	12
2.2.1 Identification des acteurs du système . . . . .	12
2.2.2 Besoins fonctionnels . . . . .	13
2.2.3 Besoins non fonctionnels . . . . .	14
2.3 Conception . . . . .	14
2.3.1 Diagramme de cas d'utilisation global . . . . .	15
2.3.2 Diagramme de classes global . . . . .	16
2.4 Backlog Produit . . . . .	17
2.5 Architecture . . . . .	19

2.5.1	Architecture Physique . . . . .	19
2.5.2	Patron d'architecture . . . . .	20
2.5.2.1	Partie Backend . . . . .	20
2.5.2.2	Partie Frontend . . . . .	21
2.6	Environnement de Travail . . . . .	22
2.6.1	Environnement Logiciel . . . . .	22
2.7	Maquettes développées . . . . .	26
2.8	Conclusion . . . . .	27
<b>3</b>	<b>Sprint 1 : Fonctionnalités liées à l'utilisateur et projets</b>	<b>28</b>
3.1	Introduction . . . . .	29
3.2	Identification des tâches . . . . .	29
3.2.1	Objectif attendu . . . . .	29
3.2.2	Backlog Sprint 1 . . . . .	29
3.3	Analyse des besoins . . . . .	31
3.3.1	Diagramme de cas d'utilisation détaillé . . . . .	31
3.3.2	Cas d'utilisation « Gérer utilisateurs » . . . . .	32
3.3.3	Diagramme de classe sprint 1 . . . . .	32
3.3.3.1	Description textuelle . . . . .	33
3.3.3.2	Diagramme de cas d'utilisation du scénario « Gérer utilisateurs » . . . . .	34
3.3.4	Cas d'utilisation « S'authentifier » . . . . .	35
3.3.4.1	Description textuelle . . . . .	35
3.3.4.2	Diagramme de séquence du scénario « S'authentifier » . . . . .	35
3.3.5	Cas d'utilisation « Mettre à jour profil » . . . . .	36
3.3.5.1	Description textuelle . . . . .	36
3.3.5.2	Diagramme de séquence du scénario « Mettre à jour profil » . . . . .	37
3.3.6	Cas d'utilisation «Gérer projets» . . . . .	39
3.3.6.1	Description textuelle . . . . .	39
3.3.6.2	Raffinement du cas d'utilisation « Gérer projets » . . . . .	39
3.3.6.3	Diagramme de séquence du scénario « Créer un projet » . . . . .	40
3.4	Réalisation . . . . .	41
3.4.1	Gérer utilisateurs . . . . .	42
3.4.2	S'authentifier . . . . .	42
3.4.3	Mettre à jour profil . . . . .	42
3.4.4	Gérer projets . . . . .	43

3.5	Conclusion . . . . .	43
<b>4</b>	<b>Sprint 2 : Gestion des tâches,des backlogs ,des sprints et des colonnes kanban</b>	<b>44</b>
4.1	Introduction . . . . .	45
4.2	Identification des tâches . . . . .	45
4.2.1	Objectif attendu . . . . .	45
4.2.2	Backlog du Sprint 2 . . . . .	45
4.3	Analyse des besoins . . . . .	47
4.3.1	Diagramme de cas d'utilisation détaillé . . . . .	47
4.3.2	Diagramme de classe sprint 2 . . . . .	48
4.3.3	Cas d'utilisation « Créer une tâche » . . . . .	50
4.3.3.1	Description textuelle . . . . .	50
4.3.3.2	Diagramme de séquence du scénario « Créer une tâche » . . . . .	50
4.3.4	Cas d'utilisation « Déplacer une tâche » . . . . .	51
4.3.4.1	Description textuelle . . . . .	51
4.3.4.2	Diagramme de séquence du cas d'utilisation « Déplacer une tâche » . . . . .	52
4.4	Réalisation . . . . .	53
4.4.1	Créer une tâche . . . . .	53
4.4.2	Consulter tâches . . . . .	54
4.4.3	Créer sprints . . . . .	55
4.5	Conclusion . . . . .	56



---

## LISTE DES FIGURES

1.1	Axia solutions . . . . .	3
1.2	Compétences des consultants Axia solutions . . . . .	4
1.3	Domaines d'activités pour les clients de Axia solutions . . . . .	4
1.4	Cycle de vie de la méthode SCRUM . . . . .	10
2.1	Diagramme de cas d'utilisation global . . . . .	15
2.2	Diagramme de classes global . . . . .	16
2.3	Architecture physique . . . . .	19
2.4	Architecture microservices . . . . .	21
2.5	Architecture Redux . . . . .	21
2.6	.Net . . . . .	23
2.7	.Net web api . . . . .	23
2.8	React Js . . . . .	23
2.9	SQLServer . . . . .	24
2.10	Aws Cloud . . . . .	24
2.11	GitHub . . . . .	25
2.12	Swagger . . . . .	25
2.13	Figma . . . . .	25
2.14	Docker . . . . .	26
2.15	Maquettes Axia-Agile . . . . .	26
3.1	Diagramme de cas d'utilisation Sprint1 . . . . .	31
3.2	Diagramme de classes du sprint 1 . . . . .	32
3.3	Diagramme de cas d'utilisation du scénario « Gérer utilisateurs » . . . . .	34
3.4	Diagramme de séquence du scénario « S'authentifier » . . . . .	36
3.5	Diagramme de séquence du scénario « Mettre à jour profil » . . . . .	38
3.6	Raffinement du cas d'utilisation « Gérer projets » . . . . .	40
3.7	Diagramme de séquence du scénario « Créer un projet » . . . . .	41
3.8	Liste des utilisateurs . . . . .	42

3.9	Formulaire de création d'utilisateur . . . . .	42
3.10	Interface login . . . . .	42
3.11	Login Erreur . . . . .	42
3.12	Interface profile «Consulter» . . . . .	43
3.13	Interface profile «mettre à jour» . . . . .	43
3.14	Interface projets . . . . .	43
3.15	Formulaire pour la création de projet . . . . .	43
4.1	Diagramme de cas d'utilisation Sprint2 . . . . .	48
4.2	Diagramme de classes du sprint 2 . . . . .	49
4.3	Diagramme de séquence du scénario « Créer une tâche » . . . . .	51
4.4	Diagramme de séquence du cas d'utilisation « Déplacer une tâche » . . . . .	53
4.5	Formulaire « Créer une tâche via l'interface Kanban » . . . . .	54
4.6	Formulaire « Créer une tâche » via l'interface backlog . . . . .	54
4.7	Consulter tâches depuis le backlog . . . . .	54
4.8	Consulter tâches depuis le Kanban . . . . .	54
4.9	Filtrer tâches par sprint . . . . .	55
4.10	Filtrer tâches par backlog . . . . .	55
4.11	Filtrer tâches par priorité spécifique . . . . .	55
4.12	Formulaire pour créer un sprint . . . . .	55
4.13	Notification de type document . . . . .	56



---

# LISTE DES TABLEAUX

1.1	Tableau comparatif entre Trello, Jira et Axia Agile . . . . .	6
1.2	Étude comparative 2 approches . . . . .	8
2.1	Product Backlog . . . . .	17
3.1	Backlog Sprint 1 . . . . .	30
3.2	Backlog du Sprint 1 — Partie 2 . . . . .	31
3.3	Description textuelle du cas d'utilisation « créer utilisateurs » . . . . .	33
3.4	Description textuelle du cas d'utilisation « S'authentifier » . . . . .	35
3.5	Description textuelle du cas d'utilisation « Mettre à jour profil » . . . . .	37
3.6	Description textuelle du cas d'utilisation « Gérer projets » . . . . .	39
4.1	Backlog du Sprint 2 — Partie 1 . . . . .	46
4.2	Backlog du Sprint 2 — Partie 2 . . . . .	47
4.3	Description textuelle du cas d'utilisation « Créer une tâche » . . . . .	50
4.4	Description textuelle du cas d'utilisation « Déplacer une tâche » . . . . .	52





---

# LISTE DES ABRÉVIATIONS

**API** = Application Programming Interface

**AWS** = Amazon Web Services

**DTO** = Data Transfer Object

**HTTP** = Hyper Text Transfer Protocol

**JS** = JavaScript

**JWT** = JSON Web Token

**.Net** = Dot Net

**ORM** = Object Relational Mapping

**QR** = Quick Response

**UI** = User Interface

**UML** = Unified Modeling Language

**URL** = Uniform Resource Locator

**UX** = User Experience



---

# INTRODUCTION GÉNÉRALE

---

# **Etude préalable**

## **Plan**

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Présentation de l'organisme d'accueil</b>	<b>3</b>
<b>3</b>	<b>Contexte du projet</b>	<b>5</b>
<b>4</b>	<b>Méthodologie de travail</b>	<b>8</b>
<b>5</b>	<b>Conclusion</b>	<b>13</b>

## 1.1 Introduction

Dans ce chapitre, nous allons présenter l'organisme d'accueil «Axia solutions» et le travail demandé. Ensuite, nous allons effectuer une étude de l'existant pour dégager les problèmes. Enfin, nous expliquerons la méthode de travail adoptée au cours de notre projet.

## 1.2 Présentation de l'organisme d'accueil

Dans cette partie, nous allons présenter l'organisme d'accueil "Axia solutions". nous allons par la suite mentionner ses services et ses domaines d'activités.

### 1.2.1 Présentation de « Axia solutions »

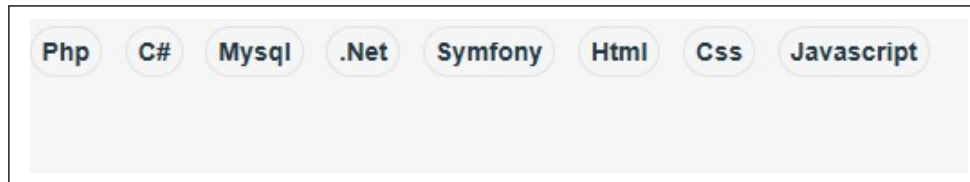
Notre stage de fin d'étude est réalisé au sein de l'entreprise « Axia solutions » (cf. figure 1). En effet, Axia Solutions est un éditeur tunisien spécialisé dans la conception et la mise en œuvre de solutions de gestion de l'innovation digitale. Fondée sur une culture de créativité et d'excellence, l'entreprise s'est imposée comme un partenaire de choix pour les organisations cherchant à relever les défis liés à l'agilité, à la performance et au développement. Axia Solutions place l'innovation au cœur de son ADN, en investissant continuellement dans la recherche et le développement pour offrir à ses clients et partenaires des solutions technologiques de pointe.



FIGURE 1.1 – Axia solutions

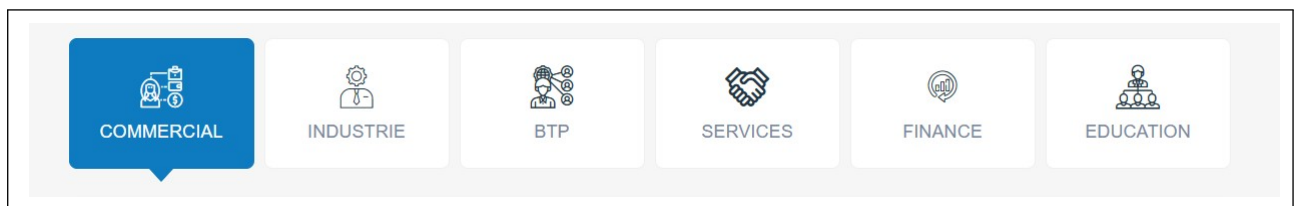
### 1.2.2 Services de « Axia solutions »

Les services techniques et le support d’Axia solutions consistent à fournir des experts qui maîtrisent les technologies présentées ci-dessous.



**FIGURE 1.2 – Compétences des consultants Axia solutions**

L’organisation consacre ses efforts au conseil en services dans les technologies récentes, les plates-formes les plus courantes et les langages les plus efficaces et utilisés. Axia solutions opère dans des domaines d’activités diversifiés . La figure ci-dessous illustre les domaines d’activités où les experts Axia solutions fournissent des solutions sur mesure aidant les clients à surmonter les défis commerciaux.



**FIGURE 1.3 – Domaines d’activités pour les clients de Axia solutions**

## 1.3 Contexte du projet

Dans cette partie, nous allons présenter l’étude de l’existant, la problématique, les objectifs ainsi que les différentes étapes du déroulement du projet.

### 1.3.1 Problématique

Chaque année, des milliers d’entreprises à travers le monde lancent des projets de développement logiciel, mais un nombre important d’entre eux échoue en raison d’une gestion désorganisée et d’un manque de structuration dans la coordination des équipes. Sans l’utilisation d’outils

dédiés à la gestion agile de projets, les entreprises se retrouvent confrontées à des défis majeurs qui entravent leur efficacité et leur capacité à livrer des produits dans les délais impartis. En l'absence d'une application de gestion agile, la communication devient difficile, entraînant pertes d'informations, retards et incompréhensions. Les responsabilités sont mal attribuées, les échéances oubliées et la priorisation inefficace, compromettant la qualité du projet.

### **1.3.2 Analyse de l'existant**

L'analyse de l'existant permet de détecter les défauts des solutions qui existent sur le marché ainsi que les imperfections afin de proposer une solution meilleure.

#### **1.3.2.1 Étude de l'existant**

L'étude de l'existant est une étape importante dans la réalisation des applications informatiques, dans n'importe quel domaine. Il est ainsi primordial, avant de commencer la réalisation du projet, d'effectuer la recherche sur les projets existants et de les analyser afin d'éviter les défauts dans le présent projet. Il y en a plusieurs solutions qui assurent la gestion agile des projets et cela fait partie de notre projet. Parmi ces solutions, nous mentionnons :

- Jira
- Trello

<b>Critères</b>	<b>Trello</b>	<b>Jira</b>	<b>Axia Agile (mon outil)</b>
<b>Complexité</b>	Faible complexité	Complexité élevée	Complexité modérée
<b>Fonctionnalités</b>	Gestion limitée au tableau Kanban	Supporte Kanban et Scrum avec de nombreuses fonctionnalités avancées	Supporte à la fois Kanban et Scrum avec des fonctionnalités personnalisables
<b>Interface</b>	Interface simple mais limitée	Interface complexe et parfois difficile à utiliser	Interface claire, moderne et facile à utiliser
<b>Flexibilité des méthodes agiles</b>	Uniquement Kanban	Kanban et Scrum	Kanban et Scrum , personnalisable selon le besoin de l'équipe

TABLE 1.1 – Tableau comparatif entre Trello, Jira et Axia Agile

### 1.3.2.2 Critique de l'existant

C'est vrai que les solutions mentionnées ci-dessus assurent de nombreux avantages, mais elles présentent également certaines limites. C'est à ce stade qu'Axia Agile, une solution complète pour remédier à ces désavantages. et répondre aux lacunes fonctionnelles des outils antérieurs, notamment grâce à l'incorporation de caractéristiques indispensables et quelques fonctionnalités telles que :

- Interface simple, moderne et facile à utiliser.
- Système de messagerie instantanée.
- Gestion des réunions.

### 1.3.2.3 Solution proposée

Afin de pallier aux différents problèmes cités dans la section 1.3.2, Axia solutions nous a confié la mission de réaliser une solution dédiée à la gestion de projets en suivant les méthodologies agiles, permettant une coopération efficace entre les équipes de développement et une organisation du travail améliorée. Notre solution Axia Agile offre une approche globale qui répond précisément aux exigences des équipes modernes de développement logiciel. Axia Agile se présente comme un environnement collaboratif centralisé qui facilite la productivité et l'efficacité de la collaboration, et augmente la clarté sur le progrès des projets.

#### **Notre application vise à :**

- Facilite l'adoption des méthodologies agiles au sein des équipes de développement.
- Optimiser et améliorer la collaboration des équipes.
- Réduire les risques liés aux projets de développement.
- Guider l'équipe à l'avancement.

## 1.4 Méthodologie de travail

Un projet informatique, peu importe sa taille ou la portée de ses objectifs, exige la mise en place d'une démarche organisationnelle tout au long de son cycle de vie. C'est ainsi qu'est apparue la notion de méthodologie. Nous proposons une étude comparative dans le tableau 1.1 afin de choisir la méthode la plus appropriée.



**TABLE 1.2 – Étude comparative 2 approches**

Thème	Approche classique	Approche agile
Cycle de vie	Cycle de vie en cascade	Itératif et incrémental
Planification	Prédictive où l'identification des besoins est établie au préalable et définitive.	Adaptative et toujours ouverte à tous ajustements en fonction des changements survenus.
Documentation	Produite en quantité importante.	Réduite au strict nécessaire.
Équipe	Une équipe avec des ressources spécialisées, dirigées par un chef de projet.	Une équipe responsabilisée où l'initiative et la communication sont privilégiées, soutenue par le chef projet.
Qualité	Contrôle qualité à la fin du cycle de développement. Le client découvre le produit fini.	Un contrôle qualité précoce et permanent. Le client visualise les résultats tôt et fréquemment.
Changement	Résistance voire opposition au changement. Processus lourds de gestion des changements acceptés.	Accueil favorable au changement inéluctable, intégré dans le processus.
Suivi de lancement	Mesure de la conformité aux plans initiaux. Analyse des écarts.	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire.
Gestion des risques	Processus distinct et rigoureux de gestion des risques.	Gestion des risques intégrés dans le processus global, avec responsabilisation de chacun dans l'identification et la résolution des risques. Pilotage par les risques.

Mesure de succès	Respect des engagements initiaux en termes de budget, de niveau de qualité et de coûts.	Satisfaction du client par la livraison de valeur ajoutée.
------------------	---	--

### 1.4.1 Choix de la méthode

Suite à l'étude comparative que nous avons effectuée dans le tableau 1.1, et après une longue réflexion basée sur les besoins de notre client, nous avons choisi «SCRUM» comme méthodologie de gestion pour notre projet. En effet, cette méthodologie agile est la plus exploitée et la plus populaire dans le monde. Ceci prouve donc son rendement et son efficacité. Cette méthodologie est adaptée aux méthodes incrémentales. Elle est aussi basée sur l'esprit collaboratif.

De plus, la méthode Agile assure la création de produits de haute qualité tout en tenant compte de l'évolution des besoins du client. Elle permet aussi de détecter les problèmes le plus tôt possible, et d'entreprendre des actions correctrices sans trop de pénalités dans les coûts et les délais.

### 1.4.2 Présentation de la méthodologie SCRUM

Scrum offre un cadre souple, précis et parfait pour les projets innovants ou complexes. Cette méthode a pour objectif de favoriser le dialogue entre le client et le prestataire afin d'optimiser la réussite des projets et d'améliorer la productivité des équipes.

Le principe de Scrum est de développer un logiciel de manière incrémentale en préservant une liste totalement transparente des demandes d'évolutions, des fonctionnalités à développer ou de corrections à implémenter (backlog).

Le client reçoit, à chaque livraison, un logiciel avec des fonctionnalités nouvelles et un état de fonctionnement parfait. Pour cela, la méthode se base sur des développements itératifs à une durée de 1 à 4 semaines.

La figure 1.2 montre le cycle de vie de la méthode SCRUM et les relations entre le client et le Product Owner.

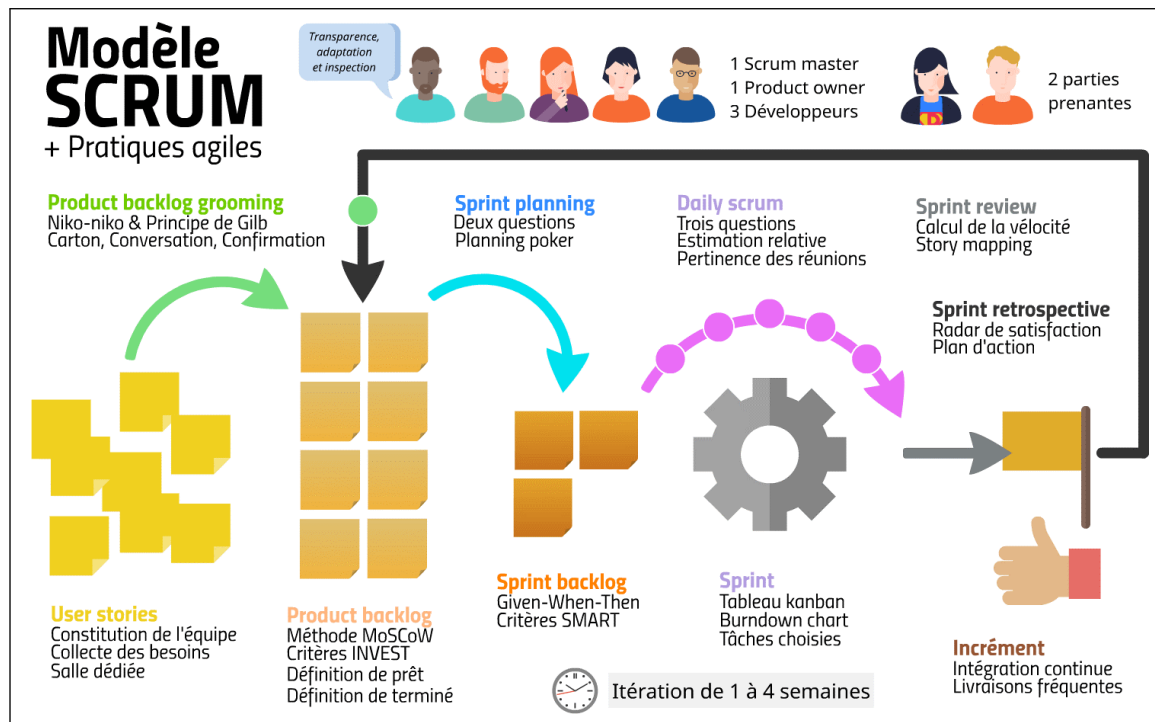


FIGURE 1.4 – Cycle de vie de la méthode SCRUM

## 1.5 Conclusion

Dans ce chapitre, nous avons donné un aperçu du projet en décrivant l'organisme d'accueil « Axia solutions » et le contexte du projet. Nous avons aussi présenté notre méthodologie de travail pour réaliser notre projet, qui est la méthode Scrum agile. Le reste du rapport sera organisé selon cette méthode. Le chapitre suivant sera consacré au lancement du projet et à la spécification de ses besoins.

---

# Spécification et analyse des besoins

## Plan

<b>1</b>	<b>Introduction</b>	<b>15</b>
<b>2</b>	<b>Identification des acteurs du système</b>	<b>15</b>
<b>3</b>	<b>Identification des besoins</b>	<b>14</b>
<b>4</b>	<b>Conception</b>	<b>17</b>
<b>5</b>	<b>Backlog Produit</b>	<b>19</b>
<b>6</b>	<b>Architecture</b>	<b>21</b>
<b>7</b>	<b>Environnement de Travail</b>	<b>24</b>
<b>8</b>	<b>Maquettes développées</b>	<b>28</b>
<b>9</b>	<b>Conclusion</b>	<b>29</b>

### 2.1 Introduction

Ce chapitre est consacré à détecter les besoins fonctionnels et non fonctionnels de notre système, à piloter le projet avec Scrum par l'élaboration du Backlog produit avec une planification des sprints ainsi qu'à décrire l'environnement du travail.

### 2.2 Identification des acteurs et des besoins

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les acteurs les plus pertinents et les fonctionnalités et d'identifier les cas d'utilisations initiaux.

#### 2.2.1 Identification des acteurs du système

Un acteur est l'idéalisation d'un rôle occupé par une personne externe, un processus ou une chose qui interagit avec un système[?]. Pour notre application, nous avons identifié 4 acteurs :

- **Un super administrateur** : Il représente le niveau le plus élevé d'administration. pour observer les utilisateurs créés et les statistiques qui y sont liées et peut créer et gérer les comptes administrateurs.
- **Un administrateur** : Il gère l'ensemble de la plateforme et Il crée des utilisateurs avec l'un des deux rôles possibles : utilisateur standard ou chef de projet. Il définit les permissions pour chaque compte et gère les droits d'accès au système.
- **Un Chef de projet** : Une fois son compte créé et ses permissions définies par l'administrateur, le chef de projet peut utiliser l'application en fonction de ses droits d'accès.
- **Un utilisateur** : Il représente un membre de l'équipe. Ses permissions sont définies par l'administrateur lors de la création de son compte. peut utiliser l'application et participer aux activités du projet en fonction de ses droits d'accès.

### 2.2.2 Besoins fonctionnels

Les besoins fonctionnels représentent les actions qu'un système doit être capable d'exécuter. Il ne sera opérationnel qu'une fois celles-ci effectuées. La solution doit fournir un ensemble de fonctionnalités répondant aux besoins de nos différents acteurs :

- **Authentification :**  
Tout utilisateur, quel que soit son rôle, doit s'authentifier afin de bénéficier des fonctionnalités de l'application après la création de son compte.
- **Gestion des utilisateur :**  
Un super administrateur peut créer, activer ou désactiver et gérer les comptes administrateurs.  
Un administrateur peut créer des utilisateurs avec un rôle spécifique (chef de projet ou utilisateur) et définir des droits d'accès spécifiques pour chaque compte.
- **Gestion du profil :**  
Tout utilisateur, quel que soit son rôle, peut consulter et mettre à jour son profil.
- **Gestion des projets**  
Chaque utilisateur a le droit d'accès pour gérer les projets, peut créer des projets en affectant des membres à ces projets, les modifier et les supprimer.
- **Gestion des tâches :**  
Chaque utilisateur a le droit d'accès pour gérer les tâches, peut créer, consulter, modifier, affecter ou supprimer des tâches.
- **Gestion des colonnes du tableau Kanban :**  
Chaque utilisateur disposant des droits d'accès peut gérer les colonnes du tableau Kanban. Il peut ajouter, supprimer, modifier ou déplacer des colonnes, ainsi qu'organiser les tâches au sein de celles-ci.
- **Gestion des backlog :**  
Chaque utilisateur disposant des droits d'accès pour gérer des backlog, peut créer, modifier, supprimer un backlog.
- **Gestion des sprints :**  
Chaque utilisateur a le droit d'accès pour gérer les sprints, peut créer, modifier, supprimer des sprints, et affecter des tâches à ce sprint depuis le backlog.
- **Gestion des canaux et messagerie collaborative :**  
Chaque utilisateur disposant des droits d'accès peut gérer les canaux de discussion, envoyer des messages, créer ou modifier des canaux en y ajoutant des utilisateurs, supprimer des canaux, ainsi qu'échanger des messages avec les autres utilisateurs au sein de ces canaux.
- **Organisation des réunions :**  
Chaque chef de projet dans l'application peut organiser, gérer des réunions Google Meet, en ajoutant les adresses e-mail des participants invités.

- **Consultation du tableau de bord des projets :**

Chaque chef projet peut consulter le dashboard pour suivre l'avancement des projet et des sprints.

- **Consultation du tableau de bord des utilisateurs :**

Chaque superadmin ou admin peut consulter le dashboard utilisateurs pour observer les utilisateurs créés et les statistiques qui y sont liées.

### 2.2.3 Besoins non fonctionnels

Les besoins non fonctionnels sont des besoins qui expriment la qualité du service rendu en termes de performances, de temps d'exécution, d'évolution... etc. La prise en compte des spécifications non fonctionnelles est essentielle, puisque cela permet de nous guider lors de la conception de l'architecture du système. Les principaux besoins non fonctionnels respectés par notre application sont :

- **L'ergonomie**

Afin de respecter les critères d'ergonomie dans notre application des charges, nous avons opté pour une démarche axée sur l'utilisateur dès la phase de conception. Nous avons commencé notre démarche en élaborant un prototype interactif sur Figma, facile à utiliser et adaptée aux besoins des utilisateurs.

- **L'Extensibilité**

Pour garantir l'extensibilité, nous avons mis en place une structure modulaire et organisée utilisant React et .NET, ce qui facilite les mises à jour futures.

- **La Sécurité**

Afin de satisfaire aux exigences de sécurité de notre application, nous avons choisi d'utiliser une authentification basée sur JWT qui est conservé dans des cookies HttpOnly et Secure avec une expiration courte et un mécanisme de renouvellement.

## 2.3 Conception

La conception est une étape fondamentale dans la réalisation d'un projet. Dans cette section de ce chapitre, nous allons présenter le diagramme de cas d'utilisation ainsi que le diagramme de classes global.

### 2.3.1 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation permet de détailler les fonctionnalités offertes par notre système ainsi que leurs associations avec les acteurs décrits auparavant.

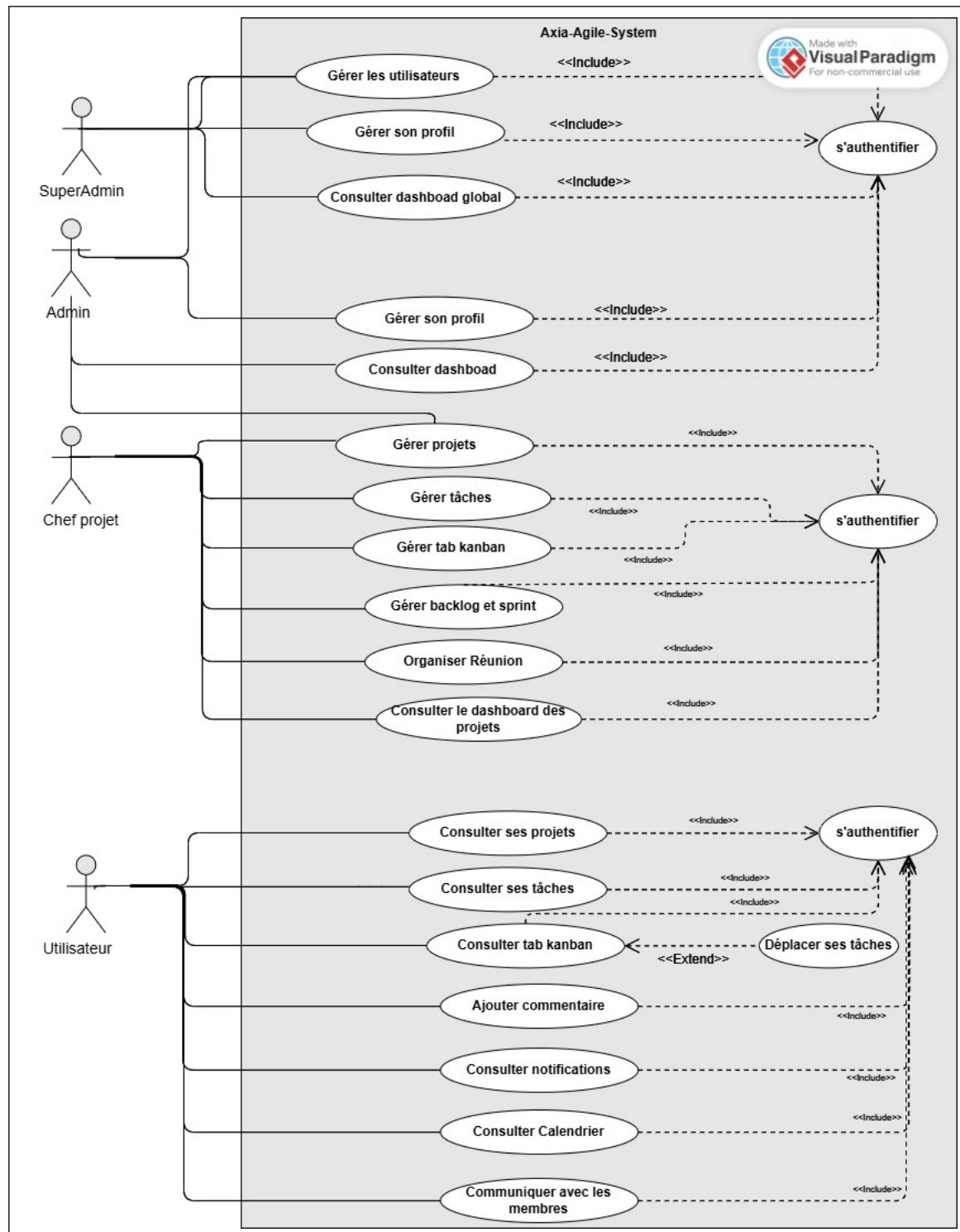


FIGURE 2.1 – Diagramme de cas d'utilisation global



### 2.3.2 Diagramme de classes global

Ci-dessous le diagramme de classes de notre application, il présente un ensemble d'éléments qui montre la structure d'un modèle.

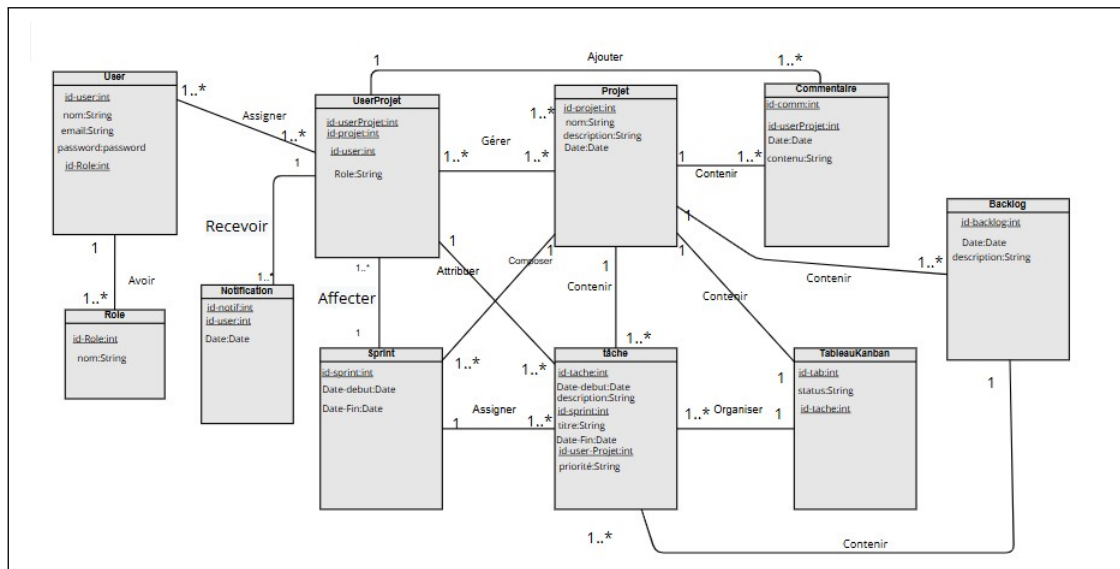


FIGURE 2.2 – Diagramme de classes global

## 2.4 Backlog Produit

Le tableau ci-dessous montre le Backlog produit de notre projet.

**TABLE 2.1 – Product Backlog**

Product Backlog		
Equipe	Nermine Mahdoui	
Date	03/02/2025	
Sprint	Feature	User Story
1	S'authentifier	<ul style="list-style-type: none"><li>– En tant qu'utilisateur de l'application, pour que je puisse m'accéder aux différentes phases de l'application, je dois m'authentifier.</li></ul>
	Gérer utilisateurs	<ul style="list-style-type: none"><li>– En tant que super administrateur de l'application, je peux créer, modifier, supprimer, activer et désactiver des comptes administrateurs.</li><li>– En tant qu'administrateur de l'application, je peux créer modifier, supprimer, activer et désactiver des comptes utilisateurs avec des droits d'accès spécifiques.</li></ul>
	Gérer profil	<ul style="list-style-type: none"><li>– En tant qu'utilisateur de l'application , je peux consulter et mettre à jour mon profil.</li></ul>
	gérer projets	<ul style="list-style-type: none"><li>– En tant que un utilisateur ou chef projet autorisé , je peux consulter, créer, modifier, supprimer des projets.</li></ul>

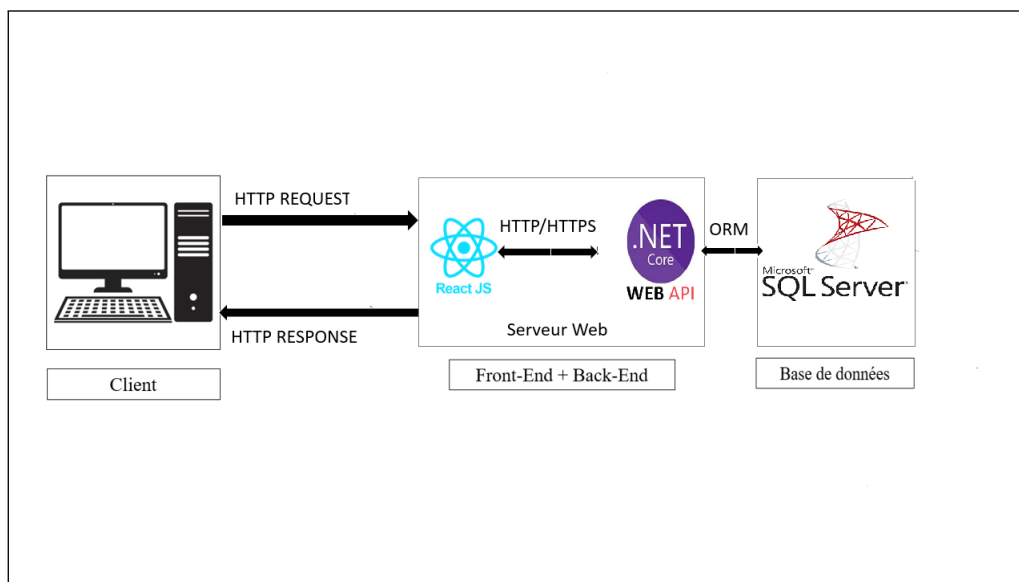
2	<p>Gérer tâches</p> <p>Gérer colonnes Kanban</p> <p>Gérer backlog</p> <p>Gérer sprint</p>	<ul style="list-style-type: none"> <li>– En tant que chef de projet, ou utilisateur autorisé, je peux créer, consulter, modifier, déplacer, supprimer des tâches.</li> <li>– En tant que chef de projet ou utilisateur autorisé, je peux ajouter, modifier, déplacer, et supprimer des colonnes dans le tableau Kanban.</li> <li>– En tant que chef de projet ou utilisateur autorisé, je peux consulter, créer, modifier et supprimer des backlog</li> <li>– En tant que chef de projet ou utilisateur autorisé, je peux consulter, créer, modifier, supprimer et affecter des tâches à un sprint.</li> </ul>
3	<p>Organiser réunions</p> <p>Gérer les canaux et communiquer</p> <p>Consulter tableau de bord</p> <p>Test et validation</p>	<ul style="list-style-type: none"> <li>– En tant que chef de projet, je peux créer, modifier et supprimer des réunions Google Meet via mon interface.</li> <li>– En tant que chef de projet ou utilisateur autorisé, je peux créer des canaux, modifier, et supprimer ces canaux, ainsi que communiquer avec les autres utilisateurs au sein de ces canaux.</li> <li>– En tant que chef de projet, je peux consulter le tableau de bord des projets pour suivre l'avancement des projets et des sprints.</li> <li>– En tant que super administrateur ou administrateur, je peux consulter le tableau de bord des utilisateurs afin d'observer les statistiques liées aux utilisateurs créés.</li> <li>– Création des images docker pour le frontend et le backend et création d'une vue pour le test des API's.</li> </ul>

## 2.5 Architecture

Une architecture d'application décrit les modèles et les techniques utilisés pour concevoir et créer une application. L'architecture fournit une feuille de route ainsi que les meilleures pratiques à suivre pour créer une application bien structurée. [?]

### 2.5.1 Architecture Physique

L'architecture physique de notre application (ou également nommée l'architecture technique d'une application) représente l'ensemble des composantes matérielles constituant l'application web. Dans ce contexte, notre application est constituée principalement d'un client (utilisateur) qui interagit avec l'application via son navigateur en envoyant des requêtes HTTP et recevant des réponses HTTP, un serveur de base de données Microsoft SQL Server qui stocke les données de l'application, accessible depuis le back-end via ORM (Object-Relational Mapping). et un serveur Web. Nous allons détailler l'architecture physique de notre application Web dans la figure ci-dessous.



**FIGURE 2.3 – Architecture physique**

### 2.5.2 Patron d'architecture

Un patron d'architecture est une solution générale et réutilisable à un problème d'architecture récurrent. Dans ce cadre, nous allons présenter le patron d'architecture pour les deux parties frontend et backend.

#### 2.5.2.1 Partie Backend

Pour la partie Backend, nous avons utilisé l'architecture microservices. En effet, l'architecture microservices permet de décomposer une application volumineuse en composants indépendants, chaque élément ayant ses propres responsabilités. Pour diffuser la requête d'un utilisateur unique, une application basée sur des microservices peut appeler plusieurs microservices internes pour composer sa réponse. [?].

- **APIGateway** : Composant servant de point d'entrée unique, gérant le routage des requêtes (ocelot)
- **Bases de données indépendantes** : Chaque microservice possède sa propre base de données.

#### Les avantages de l'architecture Microservices :

- Une meilleure scalabilité grâce à l'indépendance des services.
- Réduction des pannes grâce à l'isolation des services.
- Une plus grande Flexibilité grâce au déploiement indépendant des services

La figure ci-dessous montre comment chaque microservice est connecté à sa propre base de données pour répondre à la requête d'un utilisateur.

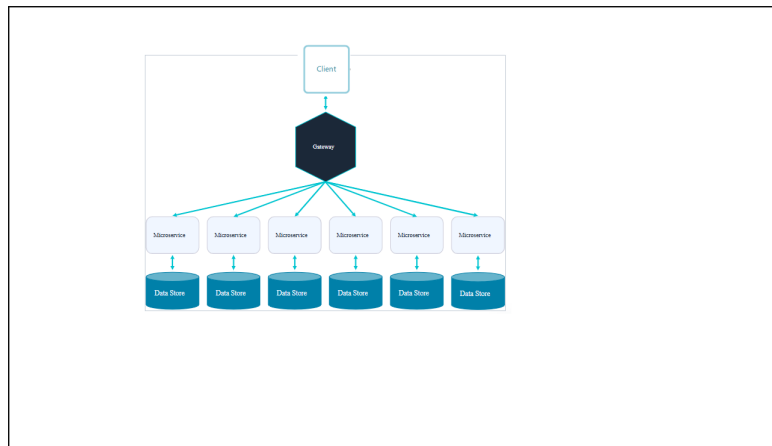


FIGURE 2.4 – Architecture microservices

### 2.5.2.2 Partie Frontend

Pour la partie Frontend, nous avons utilisé l'architecture Redux qui est une librairie Javascript qui nous permet de réduire la complexité du code en appliquant la restriction sur la manière et le temps de la mise à jour de l'état. Elle peut se produire comme le montre la figure ci-dessous.

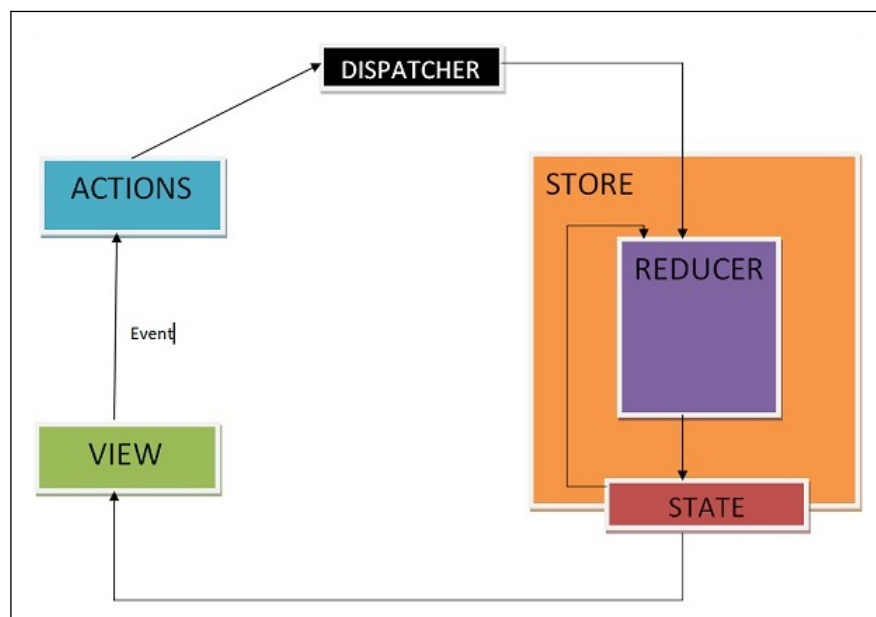


FIGURE 2.5 – Architecture Redux

-**Views** : Ce sont des classes ou des fonctions Javascript appelées composants dans lesquelles nous définissons la logique de l'interface utilisateur, elles acceptent d'autres propriétés (props) venant d'autres composants et renvoient la façon dont notre interface utilisateur prendra forme.

Il y a un composant racine qui enveloppe tous les autres composants comme des enfants qu'on nomme (app.js).

-**Redux store** : Il s'agit d'un conteneur de gestion d'état global pour notre application React. Tout composant connecté au store Redux peut envoyer un appel d'**action**, après que ces derniers transmettent des propriétés aux réducteurs (**reducers**), chaque réducteur (reducer) a une action définie et un état donné. Le rôle des réducteurs (reducers) est de fournir le nouvel état aux **composants**.

## 2.6 Environnement de Travail

Tout au long de cette partie, nous présentons l'environnement logiciel mis à la disposition du présent projet.

### 2.6.1 Environnement Logiciel

Dans ce qui suit, nous présentons les outils logiciels utilisés lors du développement de notre application.

#### -**.Net** [?]

. Net Framework est une plateforme de développement logiciel développée par Microsoft pour construire et faire fonctionner des applications Windows. Le framework .NET comprend des outils de développement, des langages de programmation et des bibliothèques pour créer des applications de bureau et Web. Pour la création du backend de notre plateforme, nous avons adopté .NET Web API en utilisant le langage C#, un framework puissant qui nous permet de créer des services web RESTful robustes et performants.



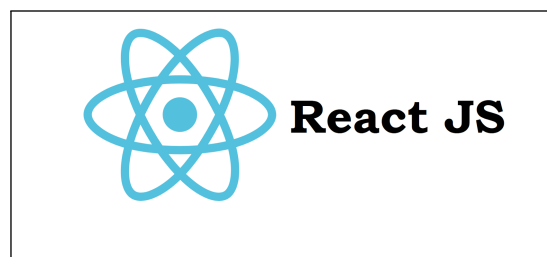
**FIGURE 2.6 – .Net**



**FIGURE 2.7 – .Net web api**

### **-React Js[?]**

Pour la partie frontend, nous avons choisi la bibliothèque JavaScript frontale React Js puisque cette dernière nous permet de créer des interfaces utilisateurs interactives facilement à l'aide de ses bibliothèques telles que ant design. Elle est open source, sécurisée, rapide, scalable et sa popularité s'accroît grâce au soutien de Facebook et d'une communauté dynamique.



**FIGURE 2.8 – React Js**

### **-SQLServer[?]**

Pour la base de données, nous avons choisi SQLServer, un système de gestion de base de données relationnelle (SGBDR) développé par Microsoft.





**FIGURE 2.9 – SQLServer**

### **-Cloud AWS [?]**

Pour le déploiement, nous avons choisi Cloud AWS qui est une plateforme qui nous fournit un service de stockage Cloud. [?] La plateforme AWS nous permet d'effectuer des déploiements rapides dans de nombreuses régions du monde en quelques clics, d'ajouter ou de supprimer de la capacité en toute simplicité. Aussi, L'accès au Cloud est rapide et la capacité est illimitée.



**FIGURE 2.10 – Aws Cloud**

### **- GitHub[?]**

Pour le contrôle de suivi et la gestion de version du projet, nous avons choisi GitHub. C'est une plateforme de développement collaborative qui utilise Git pour la gestion des versions. Elle facilite l'exploitation, le développement, la collaboration entre les équipes, ainsi que la gestion des dépôts.



**FIGURE 2.11 – GitHub**

**- Swagger[?]**

Pour tester les api backend nous avons utilisé swagger qui est un standard incontournable dans le monde du développement d'APIs RESTful. Cette spécification open source, désormais connue sous le nom d'OpenAPI Specification, révolutionne la façon dont les développeurs documentent, testent et intègrent leurs APIs. [?].



**FIGURE 2.12 – Swagger**

**- Figma [?]**

Comme outil de prototypage et design, nous avons utilisé Figma, qui est un outil de design en ligne utilisé pour élaborer des interfaces, maquettes ergonomiques et autres éléments de design UX et UI pour concevoir un site internet.



**FIGURE 2.13 – Figma**

### -Docker [?]

Docker permet d'embarquer une application dans un ou plusieurs containers logiciels qui peuvent s'exécuter sur n'importe quel serveur machine, virtuel ou physique. Il fonctionne sous Linux comme Windows Server. C'est une technologie open source qui a pour but de gérer du dimensionnement de l'infrastructure sous-jacente et faciliter les déploiements d'application. [?].



FIGURE 2.14 – Docker

## 2.7 Maquettes développées

Le maquettage est une méthode de conception d'interface qui nous permet de proposer aux clients des interfaces conformes à leurs attentes et besoins. Elle permet également de s'assurer que les besoins du client sont atteints ou non au projet[?].

Afin de comprendre les besoins de notre application et de déterminer ses différentes fonctionnalités, nous avons élaboré des maquettes avec «Figma» pour modéliser notre solution comme le montre la figure ci-dessous.

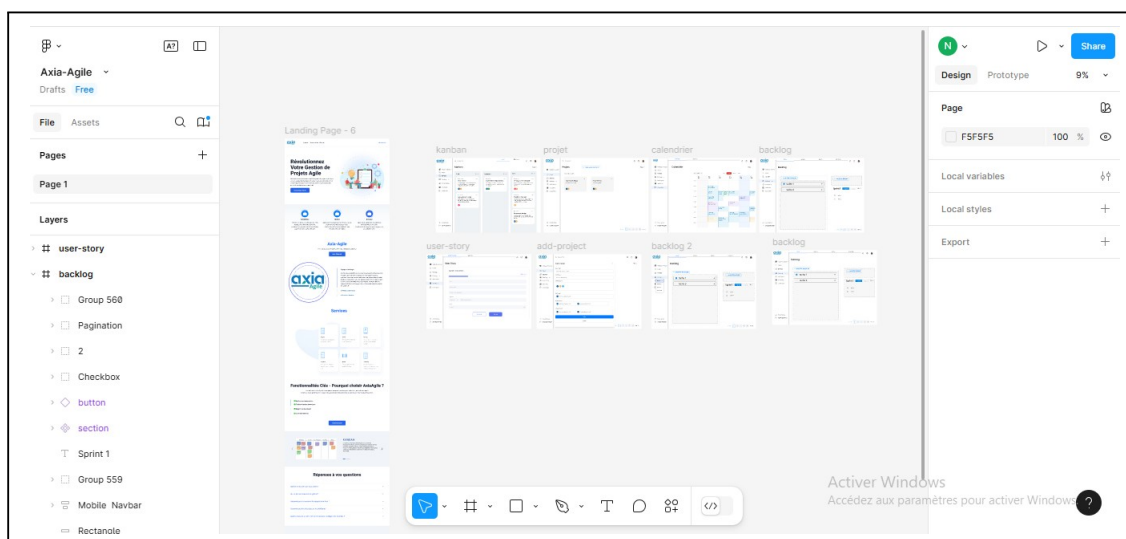


FIGURE 2.15 – Maquettes Axia-Agile

## 2.8 Conclusion

Dans ce chapitre, nous avons planifié notre travail, identifié les besoins fonctionnels et non fonctionnels de notre application et évoqué les rôles des utilisateurs. Par la suite, nous avons présenté le Backlog de notre système et détaillé la phase de planification des sprints. Enfin, nous avons choisi l'architecture de notre projet ainsi que notre environnement de travail.

---

# **Sprint 1 : Fonctionnalités liées à l'utilisateur et projets**

## **Plan**

<b>1</b>	<b>Introduction</b>	<b>31</b>
<b>2</b>	<b>Identification des tâches</b>	<b>31</b>
<b>3</b>	<b>Analyse des besoins</b>	<b>32</b>
<b>4</b>	<b>Réalisation</b>	<b>43</b>
<b>5</b>	<b>Conclusion</b>	<b>46</b>

### 3.1 Introduction

Dans ce chapitre, nous allons concevoir et réaliser les fonctionnalités du sprint 1 de notre projet. Nous allons en premier lieu présenter le sprint backlog, où nous allons détailler les fonctionnalités demandées. L'étape suivante consiste à faire l'analyse et la conception des fonctionnalités spécifiées. Enfin, nous montrons le résultat de ce sprint.

### 3.2 Identification des tâches

Dans cette partie, nous allons présenter notre objectif attendu de ce sprint ainsi que le backlog produit présenté par le tableau tiré du Backlog Product qui montre la planification des tâches du sprint.

#### 3.2.1 Objectif attendu

Suite à ce que nous avons fait durant les parties précédentes, nous avons décidé d'accomplir le but suivant : « Implémenter les fonctionnalités liées à la gestion des comptes utilisateurs(admin, utilisateur, chef projet) , la gestion des profils, et la gestion des projets ». Pour que l'utilisateur soit capable de s'authentifier, rejoindre notre application, gérer son profil,et gérer des projets.

#### 3.2.2 Backlog Sprint 1

Après avoir défini l'objectif de notre sprint, il est temps de détailler les différentes tâches à effectuer afin d'atteindre notre objectif.

**TABLE 3.1 – Backlog Sprint 1**

<b>Nom</b>	<b>Description</b>	<b>Estimation (en jour)</b>
Gérer utilisateurs	En tant que Superadmin, je peux créer, activer,désactiver,modifier,supprimer et définir des autorisations pour les comptes Admin. En tant qu'administrateur de l'application, je peux créer, activer, désactiver, modifier, supprimer et définir des autorisations pour les comptes des utilisateurs et des chefs de projet.	7
S'authentifier	En tant qu'utilisateur de l'application,quel que soit mon rôle, pour que je puisse m'accéder aux différentes phases de l'application, je dois m'authentifier.	2
Consulter profil	En tant qu'utilisateur de l'application, quel que soit mon rôle, je peux consulter mon profil.	1
mettre à jour profil	En tant qu'utilisateur de l'application, quel que soit mon rôle, je peux mettre à jour mon profil (modifier mon mot de passe ou mes informations personnelles).	2
Créer projets	En tant qu'chef projet ou utilisateur autorisé, je peux créer des projets .	3
Consulter projets	En tant que chef de projet ou utilisateur autorisé, je peux consulter les projets dont je suis le créateur ou pour lesquels je suis assigné .	1
Modifier projets	En tant que chef de projet ou utilisateur autorisé, je peux modifier un projet déjà créer .	1

TABLE 3.2 – Backlog du Sprint 1 — Partie 2

Supprimer projets	En tant que chef de projet ou utilisateur autorisé, je peux supprimer un projet déjà créé.	1
Rechercher et filtrer projets	En tant que chef de projet ou utilisateur autorisé, je peux rechercher un projet déjà créé par son nom et le filtrer par date.	1

### 3.3 Analyse des besoins

#### 3.3.1 Diagramme de cas d'utilisation détaillé

Pour mieux comprendre les fonctionnalités à réaliser durant ce sprint, nous allons présenter le diagramme de cas d'utilisation. Dans la figure ci-dessous, nous illustrons le diagramme de cas d'utilisation détaillé de ce premier sprint.

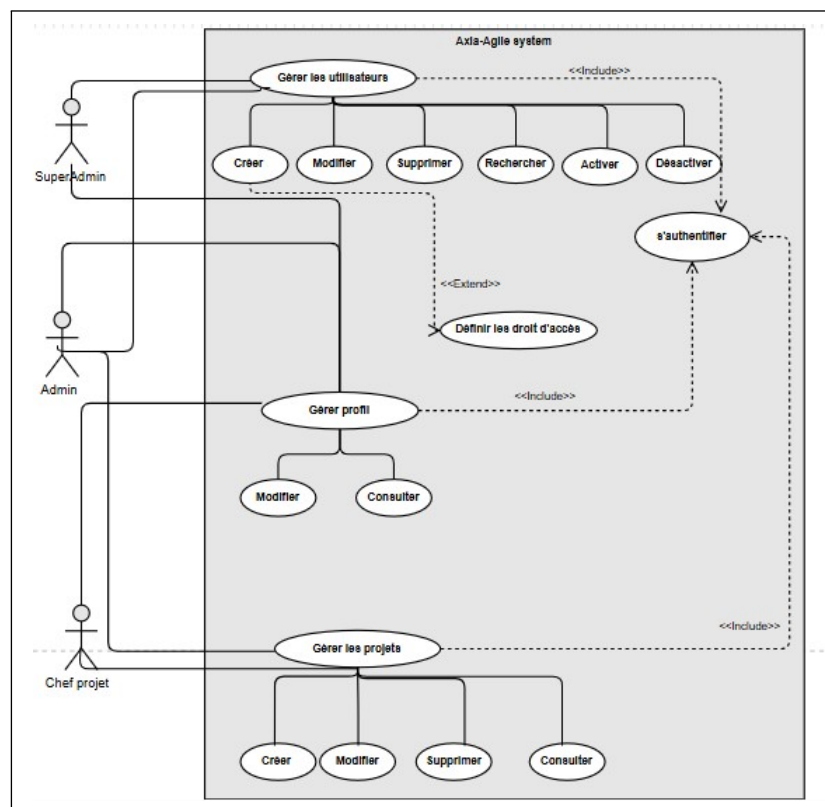


FIGURE 3.1 – Diagramme de cas d'utilisation Sprint1



### 3.3.2 Cas d'utilisation « Gérer utilisateurs »

### 3.3.3 Diagramme de classe sprint 1

Pour mieux comprendre les fonctionnalités à réaliser durant ce sprint, nous allons présenter le diagramme de classes du sprint 1.

Dans la figure ci-dessous, nous illustrons le diagramme de classes de ce premier sprint.

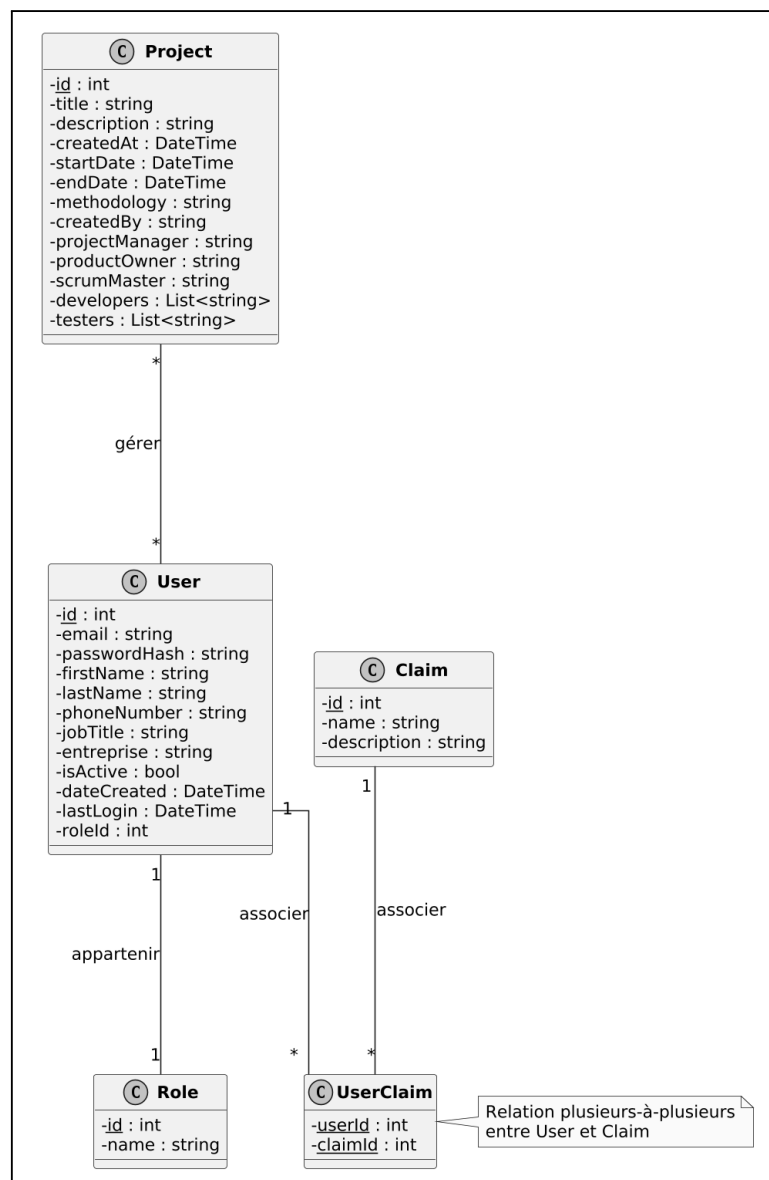


FIGURE 3.2 – Diagramme de classes du sprint 1

### 3.3.3.1 Description textuelle

TABLE 3.3 – Description textuelle du cas d'utilisation « créer utilisateurs »

Cas d'utilisation	Gérer utilisateur
Acteur	Superadmin ou Admin
Précondition	Le Superadmin ou l'Admin doit être authentifié dans le système.
Postcondition	Création de comptes utilisateurs avec des rôles différents.
Scénario nominal	<ul style="list-style-type: none"> <li>– 1. L'admin ou le superadmin ouvre l'interface de gestion des utilisateurs.</li> <li>– 2. Le système affiche le formulaire de création de compte utilisateur.</li> </ul>
	<ul style="list-style-type: none"> <li>– 3. L'utilisateur (admin/superadmin) remplit le formulaire de création de compte.</li> <li>– 4. L'utilisateur(admin/superadmin) définit les autorisations pour le compte.</li> <li>– 5. Le système vérifie les informations saisies.</li> <li>– 6. Le système ajoute l'utilisateur à la base de données.</li> </ul>
Scénario alternatif	<ul style="list-style-type: none"> <li>– 4.1 Les données saisies sont incorrectes ou les champs du formulaire sont vides ou l'email est déjà utilisé.</li> <li>– 4.1.1 Un message d'erreur s'affiche pour recommencer la saisie des données.</li> <li>– 4.1.2 Reprise de l'étape 3 du Scénario nominal.</li> </ul>

### 3.3.3.2 Diagramme de cas d'utilisation du scénario « Gérer utilisateurs »

La figure ci-dessous illustre le diagramme de séquence du scénario « Gérer les utilisateurs » qui montre les étapes d'inscription pour pouvoir s'accéder à l'application.

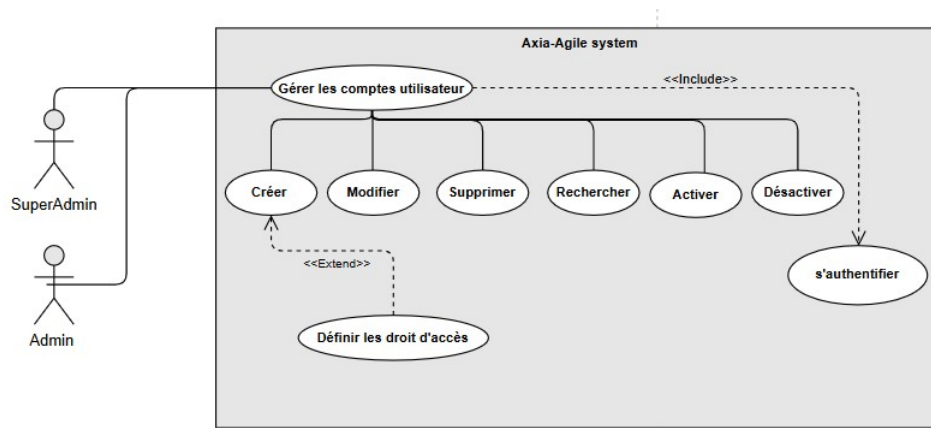


FIGURE 3.3 – Diagramme de cas d'utilisation du scénario « Gérer utilisateurs »

### 3.3.4 Cas d'utilisation « S'authentifier »

#### 3.3.4.1 Description textuelle

TABLE 3.4 – Description textuelle du cas d'utilisation « S'authentifier »

Cas d'utilisation	S'authentifier
Acteur	Utilisateur (superadmin, admin , chef projet , utilisateur)
Précondition	L'utilisateur doit avoir un compte.
Postcondition	Utilisateur authentifié
Scénario nominal	<ul style="list-style-type: none"><li>– 1. L'utilisateur lance l'application.</li><li>– 2. Le système affiche l'interface d'authentification.</li><li>– 3. L'utilisateur saisit ses identifiants (E-mail et mot de passe).</li><li>– 4. Le système vérifie les informations saisies par l'utilisateur.</li><li>– 5. Le système va générer un "Token" pour l'utilisateur authentifié.</li><li>– 6. Le système affiche l'interface "Home".</li></ul>
Scénario alternatif	<ul style="list-style-type: none"><li>– 4.1 Les données saisies sont incorrectes ou les champs du formulaire sont vides ou le compte déjà désactiver.</li><li>– 4.1.1 Un message d'erreur s'affiche pour recommencer la saisie des données.</li><li>– 4.1 Reprise de l'étape 2 du Scénario nominal.</li></ul>

#### 3.3.4.2 Diagramme de séquence du scénario « S'authentifier »

La figure ci-dessous illustre le diagramme de séquence du scénario « S'authentifier » qui montre les étapes d'authentification pour pouvoir s'accéder aux différentes phases de l'application.

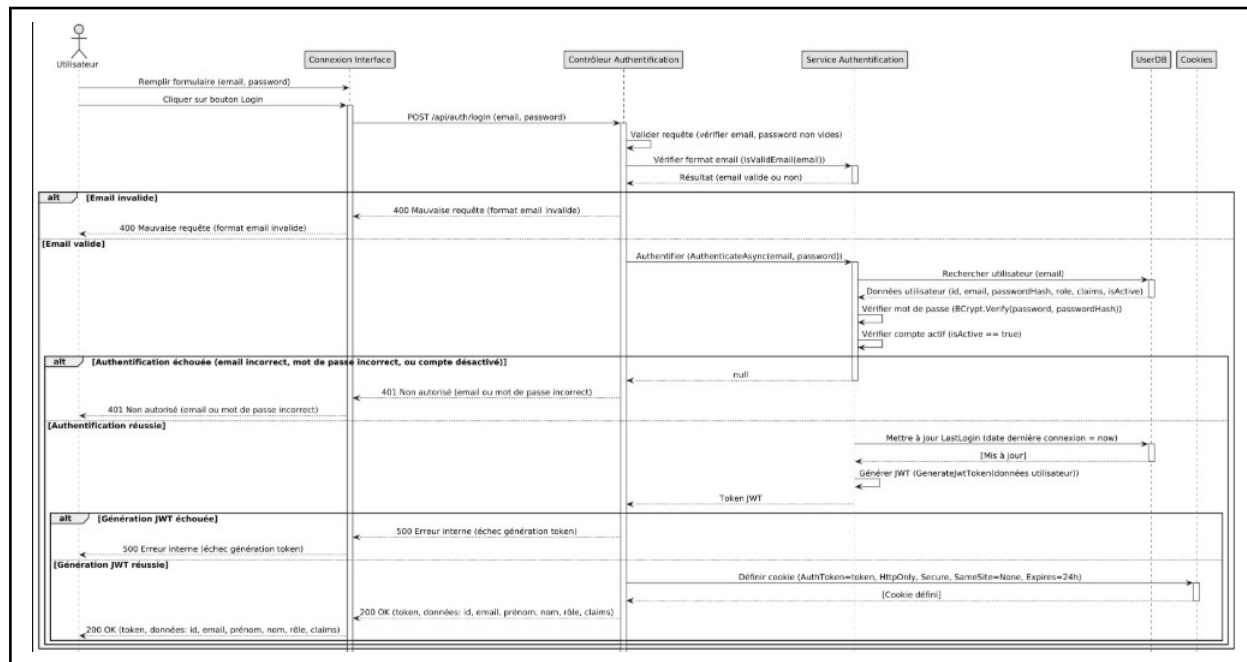


FIGURE 3.4 – Diagramme de séquence du scénario « S'authentifier »

## 3.3.5 Cas d'utilisation « Mettre à jour profil »

### 3.3.5.1 Description textuelle

Nous détaillons dans cette partie le cas d'utilisation « Mettre à jour profil ».

**TABLE 3.5 – Description textuelle du cas d'utilisation « Mettre à jour profil »**

<b>Cas d'utilisation</b>	<b>Mettre à jour profil</b>
Acteur	Utilisateur (Superadmin, Admin, Chef projet ou utilisateur)
Précondition	L'utilisateur doit être authentifié
Postcondition	Profil modifié
Scénario nominal	<ul style="list-style-type: none"> <li>– 1. L'utilisateur s'accède à la page "profil".</li> <li>– 2. L'utilisateur saisit les informations modifiées dans les formulaires de modification.</li> <li>– 3. L'utilisateur valide les modifications.</li> <li>– 4. Le système vérifie les données modifiées par l'utilisateur.</li> <li>– 5. Le système affiche un message de succès de modification.</li> </ul>
Scénario alternatif	<ul style="list-style-type: none"> <li>– 4.1 Les données saisies sont invalides ou vides.</li> <li>– 4.1.1 Un message d'erreur s'affiche pour recommencer la saisie des données.</li> <li>– 4.1.2 Reprise de l'étape 2 du Scénario nominal.</li> </ul>

### 3.3.5.2 Diagramme de séquence du scénario « Mettre à jour profil »

La figure ci-dessous illustre le diagramme de séquence du scénario « Mettre à jour le profil » qui montre les étapes de mise à jour du profil d'un utilisateur.

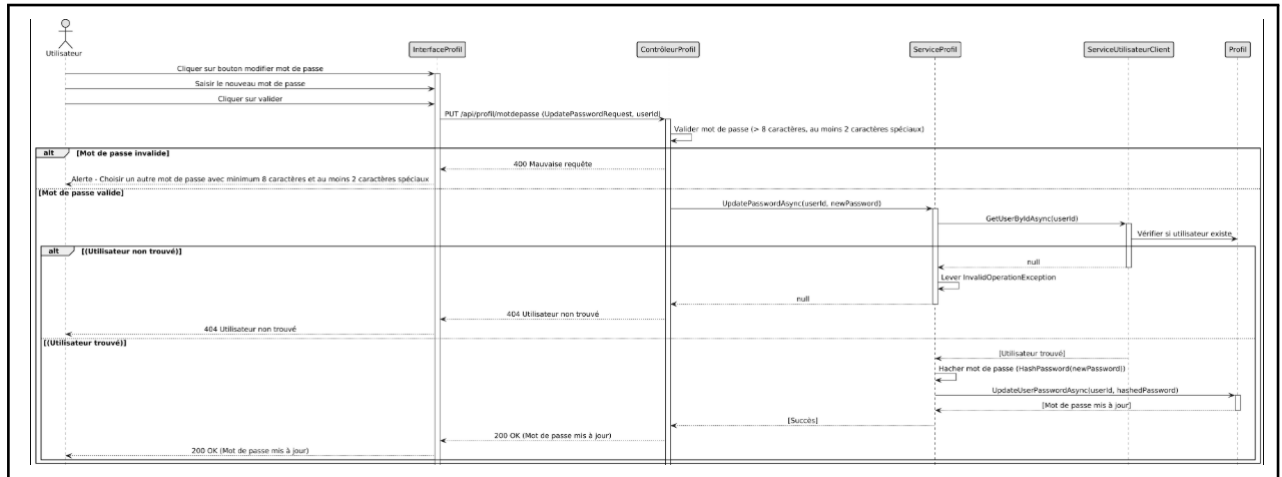


FIGURE 3.5 – Diagramme de séquence du scénario « Mettre à jour profil »

### 3.3.6 Cas d'utilisation «Gérer projets»

#### 3.3.6.1 Description textuelle

TABLE 3.6 – Description textuelle du cas d'utilisation « Gérer projets »

Cas d'utilisation	Gérer projets
Acteur	Utilisateur (Chef projet ou Admin)
Précondition	L'utilisateur doit avoir un compte et doit être authentifié .
Postcondition	Projets gérés
Scénario nominal	<ul style="list-style-type: none"> <li>- 1. L'utilisateur se connecte sur son compte .</li> <li>- 2. L'utilisateur s'accède à la page « Projets ».</li> <li>- 3. Le système affiche les projets de l'utilisateur.</li> <li>- 4. L'utilisateur peut créer des projets en appuyant sur le bouton "créer" .</li> <li>- 5. L'utilisateur peut modifier ou supprimer ses projets en appuyant sur le menu de projets.</li> </ul>
Scénario alternatif	<ul style="list-style-type: none"> <li>- 3.1.1 L'utilisateur n'a aucun projet crée.</li> <li>- 3.1.2 Un message s'affiche "Vous n'avez aucun projet".</li> </ul>

#### 3.3.6.2 Raffinement du cas d'utilisation « Gérer projets »

La figure ci-dessous représente le raffinement du cas d'utilisation « Gérer projets ».



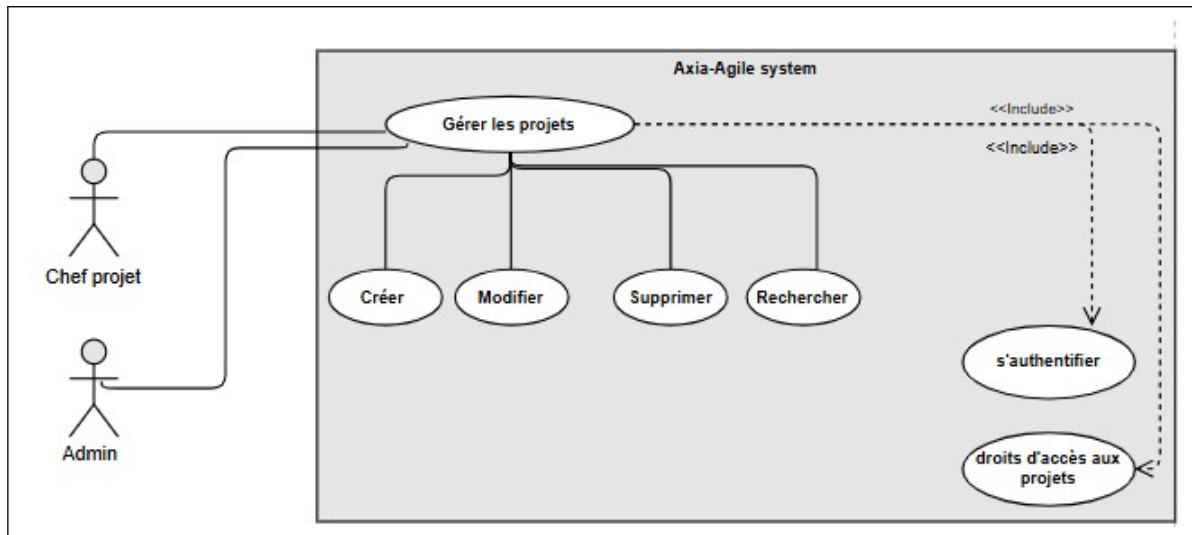


FIGURE 3.6 – Raffinement du cas d'utilisation « Gérer projets »

Le cas d'utilisation « Gérer projets » est composé de 4 sous-cas d'utilisation.

- L'utilisateur peut créer des projets.
- L'utilisateur peut consulter ses projets.
- L'utilisateur peut modifier les information de ses projets.
- L'utilisateur peut supprimer ses projets.

### 3.3.6.3 Diagramme de séquence du scénario « Créer un projet »

La figure ci-dessous illustre le diagramme de séquence du scénario « Créer un projet » qui montre les étapes de création d'un projet.

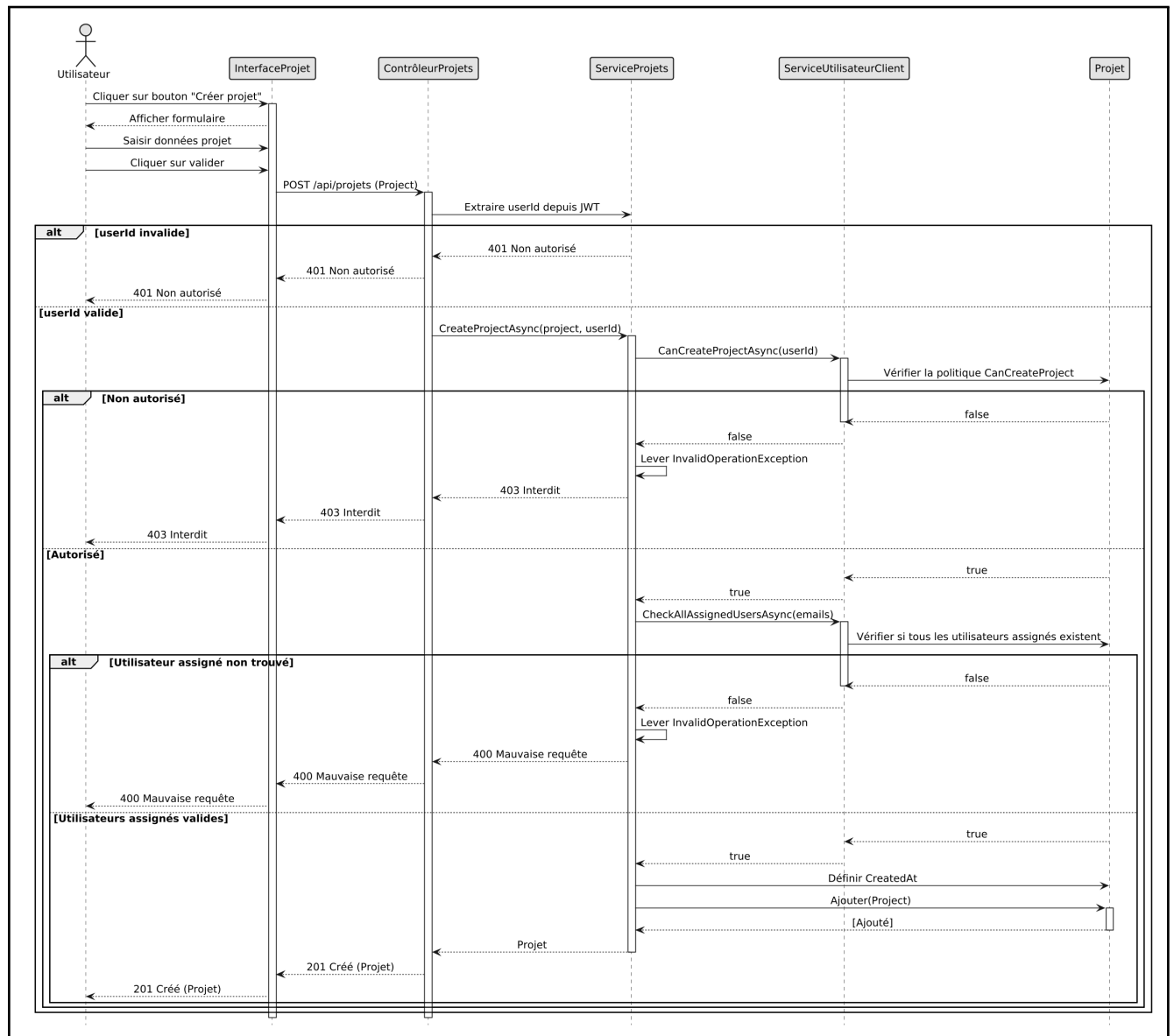


FIGURE 3.7 – Diagramme de séquence du scénario « Créer un projet »

## 3.4 Réalisation

Après avoir établi l'étude conceptuelle du premier sprint, nous présentons quelques captures d'écran de notre application afin de mieux comprendre la réalisation des différentes tâches de ce sprint.

## 3.4.1 Gérer utilisateurs

Les figures ci-dessous présentent les interfaces de la création de comptes utilisateurs où l'admin peut choisir de faire un compte soit utilisateur simple, soit chef de projet, en passant par le formulaire suivant.

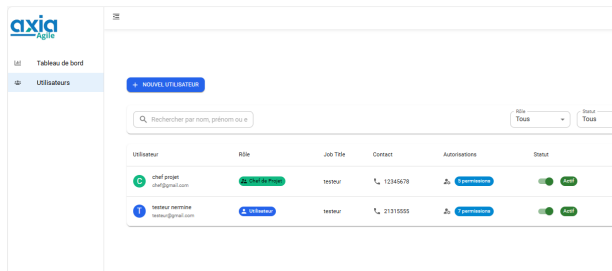


FIGURE 3.8 – Liste des utilisateurs

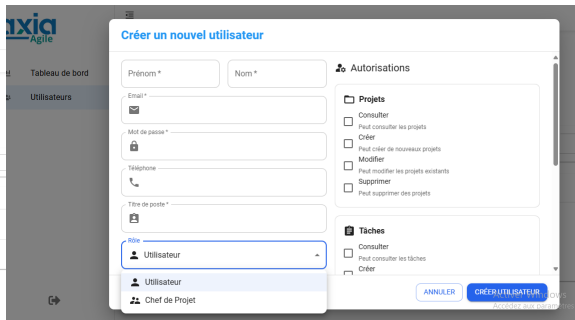


FIGURE 3.9 – Formulaire de création d'utilisateur

## 3.4.2 S'authentifier

La figure 3.13 présente l'interface d'authentification où l'internaute peut se connecter à l'application. En effet, l'internaute appuie sur le bouton « Se connecter » ; si les paramètres sont valides et que le compte n'est pas déjà désactivé, le système affiche la page "Home", sinon le système affiche un message d'erreur.

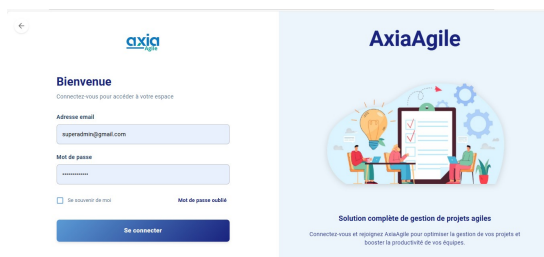


FIGURE 3.10 – Interface login

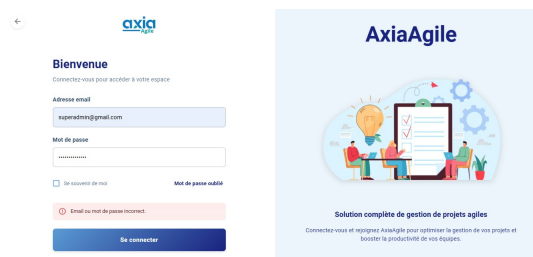
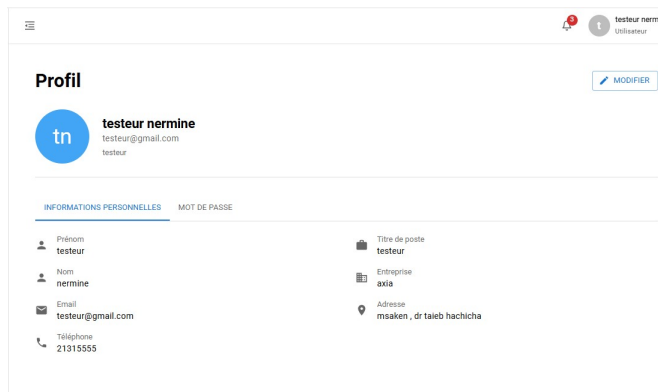


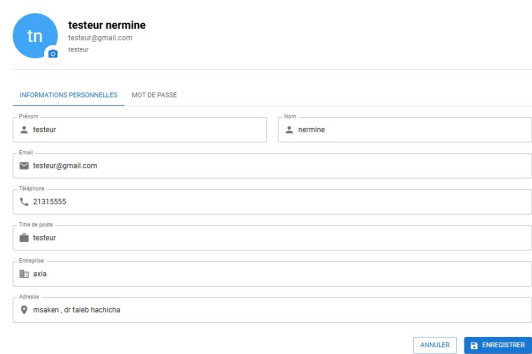
FIGURE 3.11 – Login Erreur

## 3.4.3 Mettre à jour profil

Les figures ci-dessous présentent les interfaces de mise à jour du profil pour l'utilisateur.



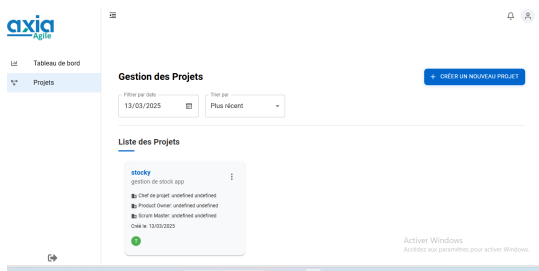
**FIGURE 3.12 – Interface profile «Consulter»**



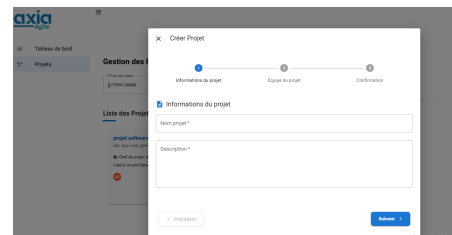
**FIGURE 3.13 – Interface profile «mettre à jour»**

## 3.4.4 Gérer projets

Les figures ci-dessous représentent l'interface de création des projets ainsi que la liste des projets ajoutés.



**FIGURE 3.14 – Interface projets**



**FIGURE 3.15 – Formulaire pour la création de projet**

## 3.5 Conclusion

Tout au long de ce chapitre, nous avons réussi à mentionner les fonctionnalités liées à l'utilisateur et les projets tels que la création de compte, l'authentification, la gestion de profil et la gestion des projets. Dans le chapitre qui suit, notre effort sera consacré à produire le sprint 2.

---

# **Sprint 2 : Gestion des tâches,des backlogs ,des sprints et des colonnes kanban**

## **Plan**

<b>1</b>	<b>Introduction</b>	<b>31</b>
<b>2</b>	<b>Identification des tâches</b>	<b>31</b>
<b>3</b>	<b>Analyse des besoins</b>	<b>32</b>
<b>4</b>	<b>Réalisation</b>	<b>43</b>
<b>5</b>	<b>Conclusion</b>	<b>46</b>

## **4.1 Introduction**

Dans ce chapitre, nous allons passer à la réalisation du sprint 2 de notre projet qui comporte la gestion des tâches, la gestion des backlogs et la gestion des sprints. Nous allons en premier lieu présenter le sprint backlog, où nous allons détailler les fonctionnalités demandées. L'étape suivante consiste à faire l'analyse et la conception de notre application. Finalement, nous allons faire la partie réalisation.

## **4.2 Identification des tâches**

Dans cette partie, nous allons présenter notre objectif attendu de ce sprint ainsi que le backlog produit présenté par le tableau tiré du Backlog Product qui montre la planification des tâches du sprint.

### **4.2.1 Objectif attendu**

Suite à ce que nous avons réalisé dans les parties précédentes, nous avons décidé d'accomplir l'objectif suivant : « Implémenter les fonctionnalités liées aux tâches ». L'utilisateur sera capable de créer, gérer et affecter des tâches à des sprints et au backlog.

### **4.2.2 Backlog du Sprint 2**

Après avoir défini l'objectif de notre sprint, il est temps de détailler les différentes tâches à effectuer afin d'atteindre cet objectif.

**TABLE 4.1 – Backlog du Sprint 2 — Partie 1**

<b>Nom</b>	<b>Description</b>	<b>Estimation (en jours)</b>
Créer un backlog	En tant qu'utilisateur, je peux créer un backlog en y ajoutant un titre et une description.	1
Modifier un backlog	En tant qu'utilisateur, je peux modifier un backlog déjà créé.	1
Supprimer un backlog	En tant qu'utilisateur, je peux supprimer un backlog déjà existant.	1
Créer une tâche	En tant qu'utilisateur, je peux créer une tâche dans un backlog ou dans un tableau Kanban.	1
Modifier une tâche	En tant qu'utilisateur, je peux modifier une tâche existante dans un backlog ou un tableau Kanban.	1
Supprimer une tâche	En tant qu'utilisateur, je peux supprimer une tâche existante dans un backlog ou un tableau Kanban.	1
Déplacer une tâche	En tant qu'utilisateur, je peux déplacer une tâche dans un backlog ou un tableau Kanban.	1
Créer une colonne Kanban	En tant qu'utilisateur, je peux créer une colonne dans un tableau Kanban.	1
Modifier une colonne Kanban	En tant qu'utilisateur, je peux modifier une colonne existante dans un tableau Kanban.	1
Supprimer une colonne Kanban	En tant qu'utilisateur, je peux supprimer une colonne existante dans un tableau Kanban.	1
Filtrer des tâches	En tant qu'utilisateur, je peux filtrer les tâches depuis les backlogs ou les sprints.	1

**TABLE 4.2 – Backlog du Sprint 2 — Partie 2**

Créer un sprint	En tant qu'utilisateur, je peux créer un sprint.	1
Modifier un sprint	En tant qu'utilisateur, je peux modifier un sprint existant.	1
Supprimer un sprint	En tant qu'utilisateur, je peux supprimer un sprint existant.	1
Affecter une tâche à un sprint	En tant qu'utilisateur, je peux affecter des tâches déjà créées dans le backlog à un sprint.	1

## 4.3 Analyse des besoins

### 4.3.1 Diagramme de cas d'utilisation détaillé

Pour mieux comprendre les fonctionnalités à réaliser durant ce sprint, nous allons présenter le diagramme de cas d'utilisation.

Dans la figure ci-dessous, nous illustrons le diagramme de cas d'utilisation détaillé de ce deuxième sprint.



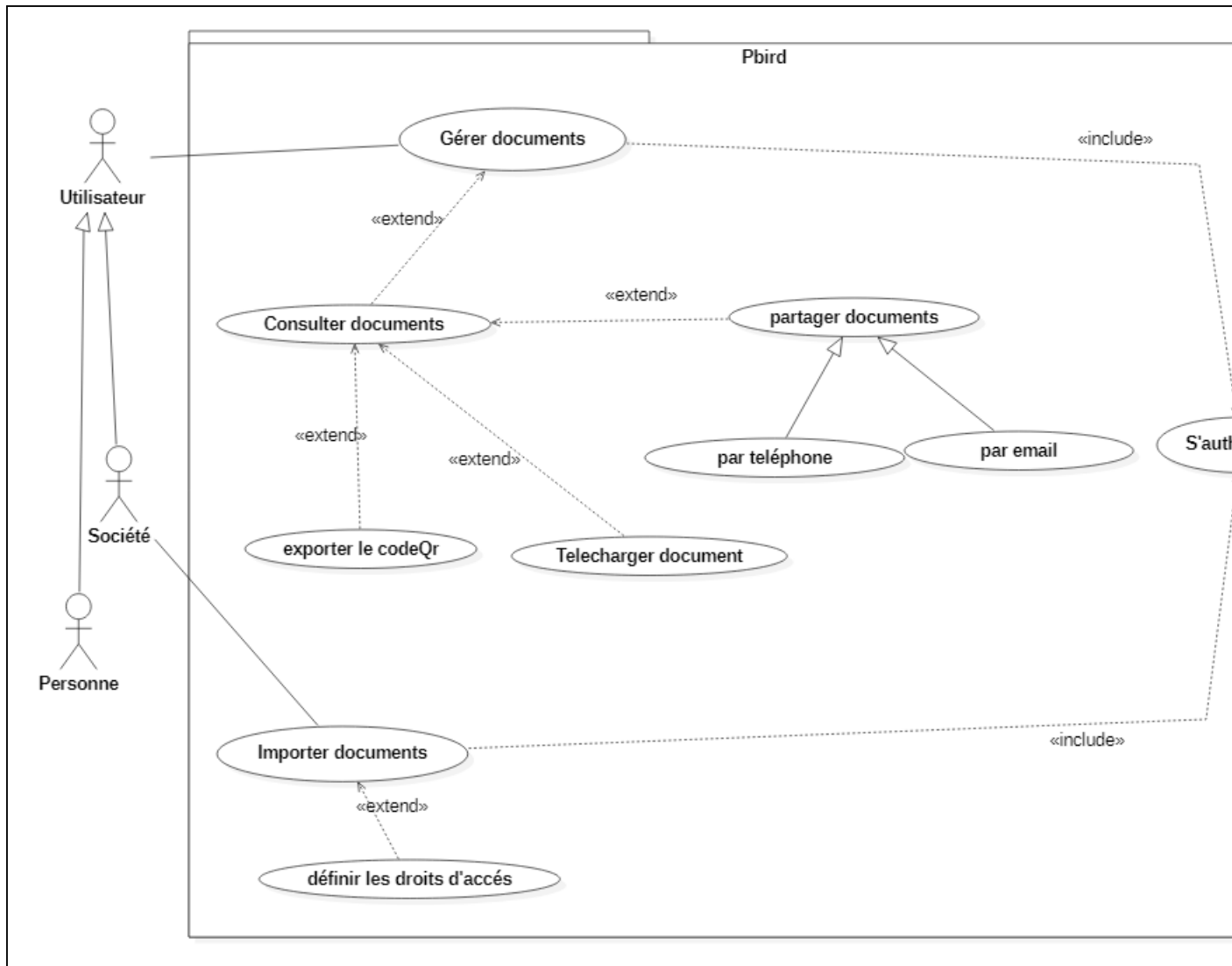


FIGURE 4.1 – Diagramme de cas d'utilisation Sprint2

### 4.3.2 Diagramme de classe sprint 2

Pour mieux comprendre les fonctionnalités à réaliser durant ce sprint, nous allons présenter le diagramme de classes du sprint 2.

Dans la figure ci-dessous, nous illustrons le diagramme de classes de ce deuxième sprint.

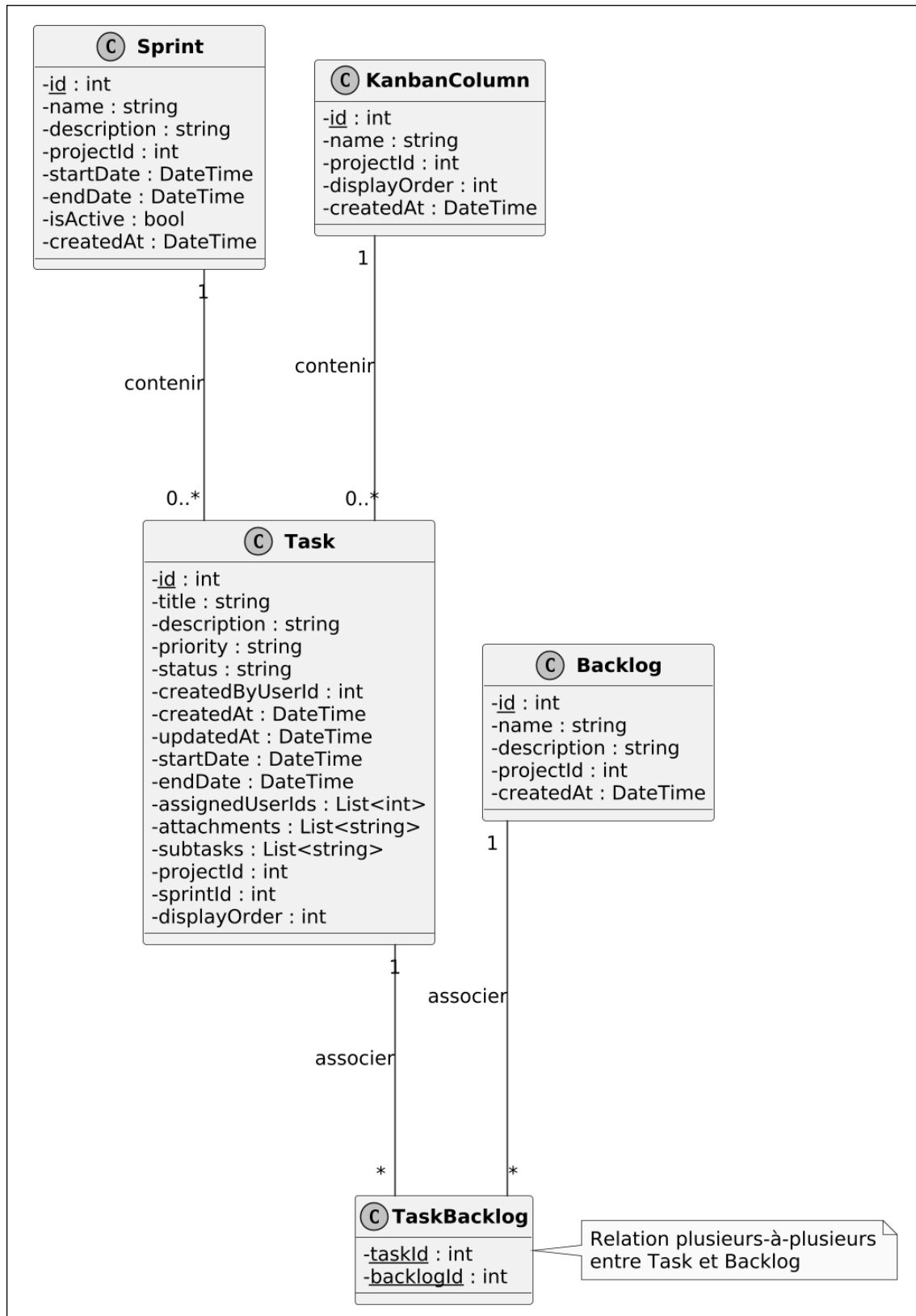


FIGURE 4.2 – Diagramme de classes du sprint 2

### 4.3.3 Cas d'utilisation « Créer une tâche »

#### 4.3.3.1 Description textuelle

TABLE 4.3 – Description textuelle du cas d'utilisation « Créer une tâche »

Cas d'utilisation	Créer une tâche
Acteur	Utilisateur disposant de la permission de créer une tâche
Précondition	L'utilisateur doit avoir un compte, être authentifié et disposer d'une autorisation pour créer une tâche.
Postcondition	Une nouvelle tâche est ajoutée dans la liste des tâches.
Scénario nominal	<ul style="list-style-type: none"><li>– 1. L'utilisateur accède à l'interface de gestion des tâches (Kanban ou backlog).</li><li>– 2. L'utilisateur clique sur le bouton « Créer une tâche ».</li><li>– 3. Le système affiche un formulaire.</li><li>– 4. L'utilisateur saisit les informations de la tâche dans le formulaire et valide la création.</li><li>– 5. Le système affiche un message de succès de création et la nouvelle tâche est bien affichée.</li></ul>
Scénario alternatif	<ul style="list-style-type: none"><li>– 4.1 Les données sont vides ou le titre de la tâche n'est pas saisi.</li><li>– 4.1.1 Affichage d'un message d'erreur pour recommencer la saisie.</li><li>– 4.1.2 Reprise de l'étape 4 du Scénario nominal.</li></ul>

#### 4.3.3.2 Diagramme de séquence du scénario « Créer une tâche »

La figure ci-dessous illustre le diagramme de séquence du cas d'utilisation « Créer une tâche » qui montre le scénario de création des tâches.

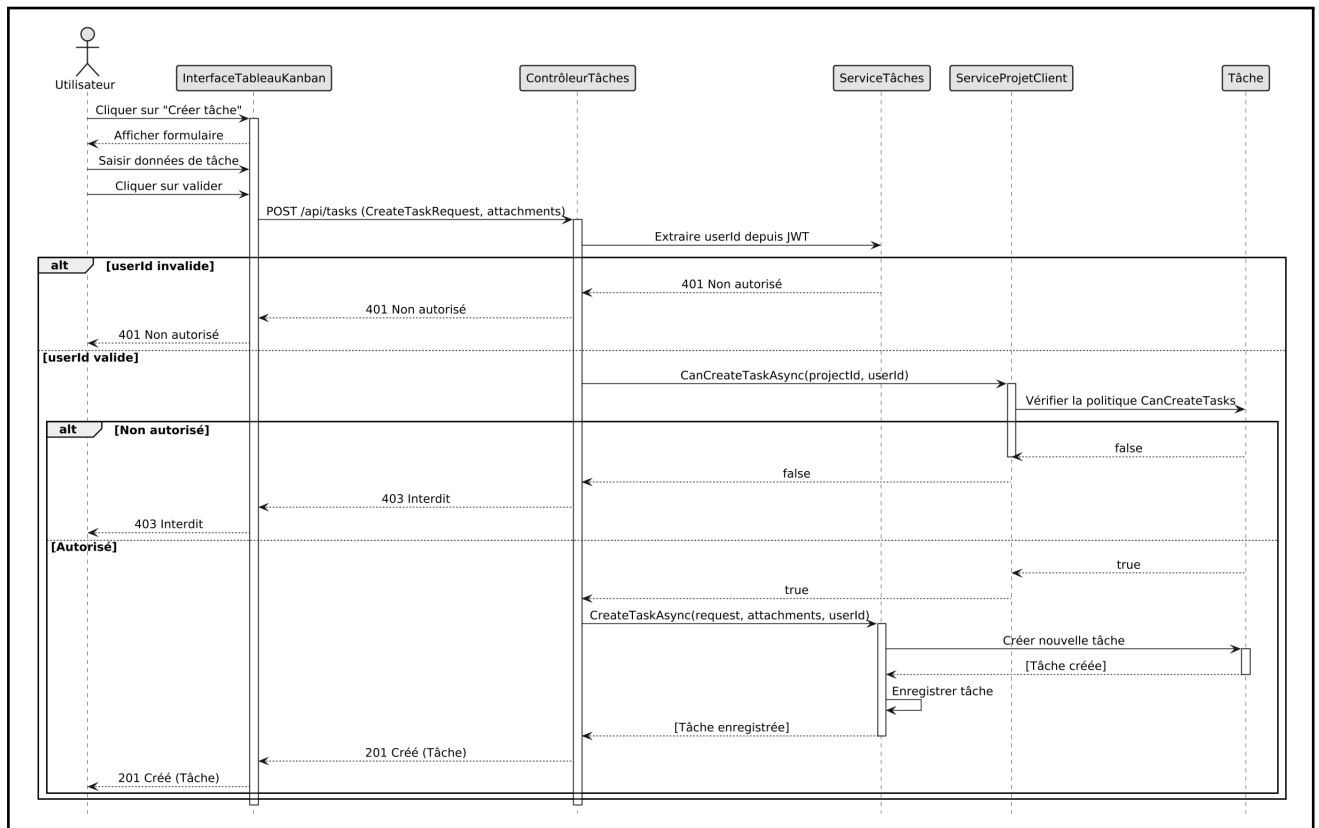


FIGURE 4.3 – Diagramme de séquence du scénario « Créer une tâche »

### 4.3.4 Cas d'utilisation « Déplacer une tâche »

#### 4.3.4.1 Description textuelle

Nous détaillons dans cette partie le cas du déplacement de la tâche.

**TABLE 4.4 – Description textuelle du cas d'utilisation « Déplacer une tâche »**

Cas d'utilisation	Déplacer une tâche
Acteur	Utilisateur disposant de la permission de déplacer une tâche.
Précondition	<p>L'utilisateur doit avoir un compte, être authentifié et disposer d'un droit d'accès pour consulter les tâches.</p> <p>Au moins une tâche et deux colonnes du tableau Kanban doivent déjà être créées.</p>
Postcondition	La tâche est déplacée (son statut est mis à jour).
Scénario nominal	<ul style="list-style-type: none"><li>– 1. L'utilisateur accède à la page "Kanban".</li><li>– 2. Le système affiche les tâches déjà créées.</li><li>– 3. L'utilisateur choisit la tâche qu'il veut déplacer.</li><li>– 4. L'utilisateur clique sur l'icône de drag and drop sur la tâche et la déplace vers la colonne Kanban souhaitée.</li><li>– 5. La tâche est bien déplacée et affichée dans la nouvelle colonne.</li><li>– 6. Le système change le statut de la tâche par le nom de la colonne dans laquelle elle est déplacée.</li></ul>
Scénario alternatif	<ul style="list-style-type: none"><li>– 3.1 L'utilisateur n'a pas le droit nécessaire pour déplacer une tâche.</li></ul>

#### 4.3.4.2 Diagramme de séquence du cas d'utilisation « Déplacer une tâche »

La figure ci-dessous représente le diagramme de séquence du cas d'utilisation « Déplacer une tâche ».

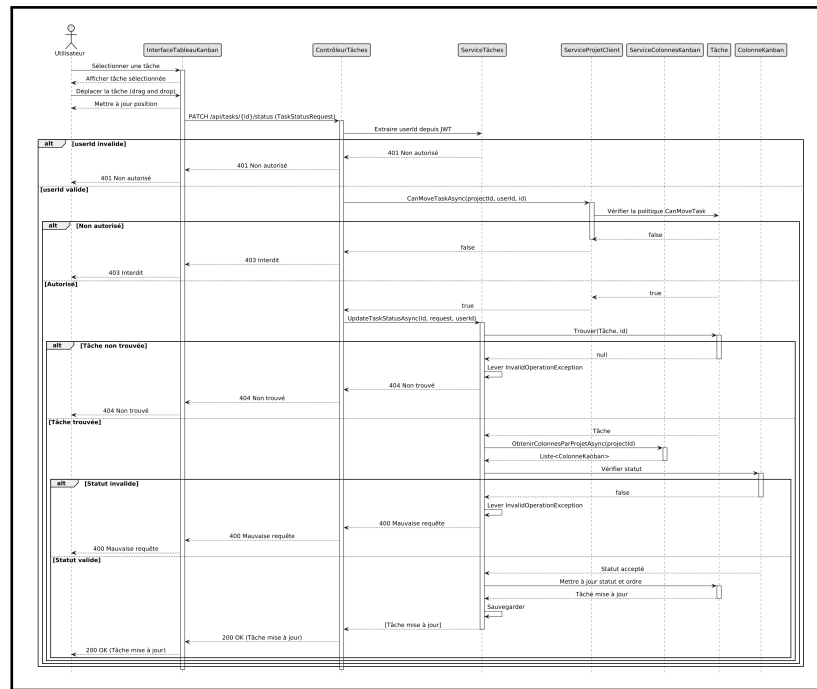


FIGURE 4.4 – Diagramme de séquence du cas d'utilisation « Déplacer une tâche »

## 4.4 Réalisation

Après avoir établi l'étude conceptuelle du deuxième sprint, nous présentons quelques captures d'écran de notre application afin de mieux comprendre la réalisation des différentes tâches de ce sprint.

### 4.4.1 Créer une tâche

Les figures ci-dessous montrent la création de tâche qui peut être faite via l'interface backlog ou l'interface Kanban.

## SPRINT 2 : GESTION DES TÂCHES, DES BACKLOGS, DES SPRINTS ET DES COLONNES KANBAN

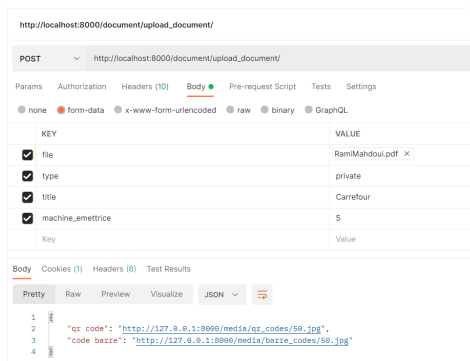


FIGURE 4.5 – Formulaire « Créer une tâche via l'interface Kanban »

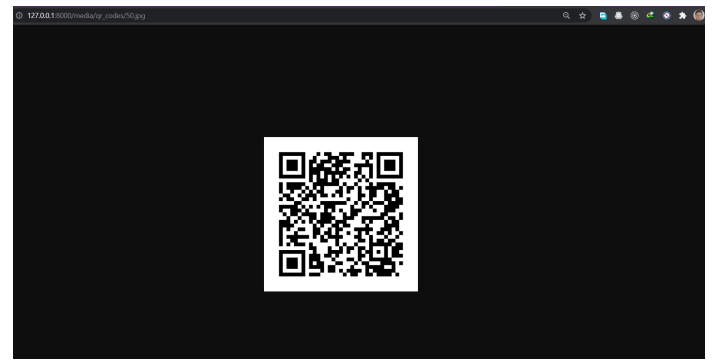


FIGURE 4.6 – Formulaire « Créer une tâche » via l'interface backlog

### 4.4.2 Consulter tâches

Les figures ci-dessous montrent la consultation de tâches, dans le tableau Kanban ou dans le backlog.

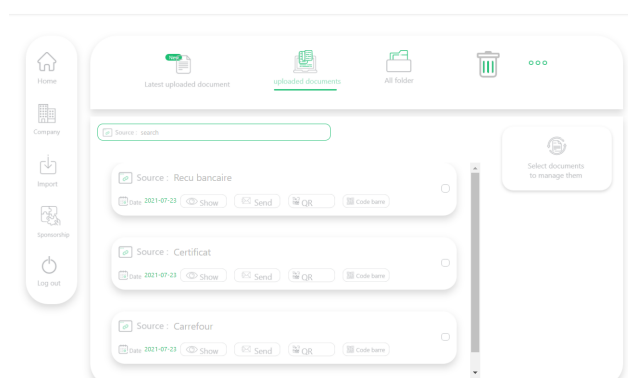


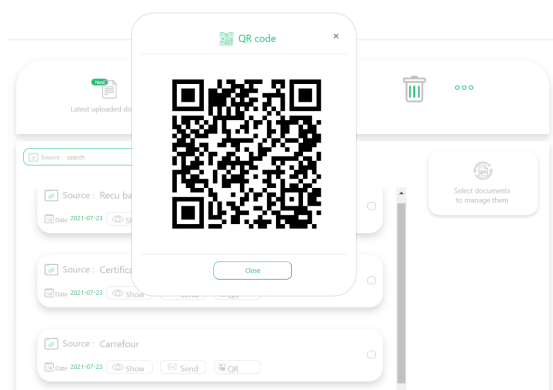
FIGURE 4.7 – Consulter tâches depuis le backlog



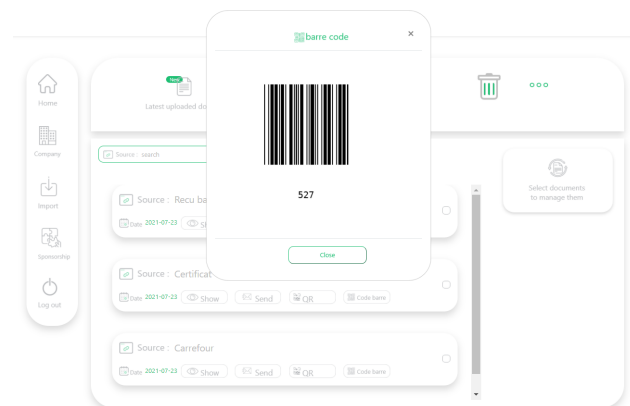
FIGURE 4.8 – Consulter tâches depuis le Kanban

Lors de la consultation des tâches dans le tableau Kanban, l'utilisateur peut filtrer ses tâches soit par le backlog ou sprint, ou bien par une priorité spécifique, comme le montrent les figures ci-dessous.

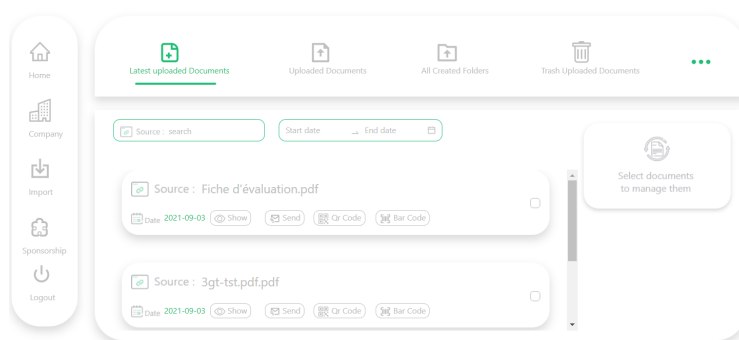
## SPRINT 2 : GESTION DES TÂCHES, DES BACKLOGS, DES SPRINTS ET DES COLONNES KANBAN



**FIGURE 4.9 – Filtrer tâches par sprint**



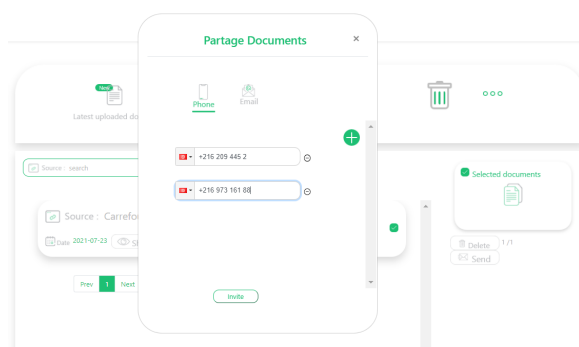
**FIGURE 4.10 – Filtrer tâches par backlog**



**FIGURE 4.11 – Filtrer tâches par priorité spécifique**

### 4.4.3 Créer sprints

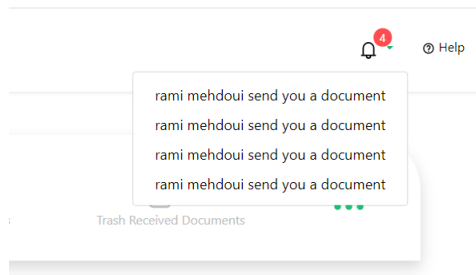
La figure ci-dessous montre le formulaire de création d'un sprint.



**FIGURE 4.12 – Formulaire pour créer un sprint**

Lorsque l'utilisateur crée un sprint, il peut lui affecter des tâches trouvées dans le backlog, comme le présente la figure ci-dessous.





**FIGURE 4.13 – Notification de type document**

## 4.5 Conclusion

Tout au long de ce chapitre, nous avons réussi à mentionner les fonctionnalités liées aux tâches , tableau kanban, sprint et backlog. Dans le chapitre qui suit, notre effort sera consacré à produire le sprint 3.