



DÉDICACES



REMERCIEMENT



TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE	1
1 Etude préalable	2
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	5
1.3.1 Problématique	5
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
1.3.2.4 Objectif du projet	7
1.4 Méthodologie de travail	7
1.4.1 Choix de la méthode	9
1.4.2 Présentation de la méthodologie SCRUM	9
1.4.3 Les rôles dans SCRUM	10
1.5 Conclusion	12
2 Sprint 0 :Spécification et analyse des besoins	13
2.1 Introduction	14
2.2 Identification des acteurs et des besoins	14
2.2.1 Identification des acteurs du système	14
2.2.2 Besoins fonctionnels	14
2.2.3 Besoins non fonctionnels	15
2.3 Conception	16
2.3.1 Diagramme de cas d'utilisation global	16
2.3.2 Diagramme de classes global	17

2.4	Backlog Produit	19
2.5	Architecture	20
2.5.1	Architecture Physique	21
2.5.2	Patron d'architecture	21
2.5.2.1	Partie Backend	22
2.5.2.2	Partie Frontend	23
2.6	Environnement de Travail	24
2.6.1	Environnement Logiciel	24
2.7	Maquettes développées	27
2.8	Conclusion	28
3	Sprint 1 : Fonctionnalités liées à l'utilisateur et projets	29
3.1	Introduction	30
3.2	Identification des tâches	30
3.2.1	Objectif attendu	30
3.2.2	Backlog Sprint 1	30
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
3.3.2.1	Description textuelle	32
3.3.2.2	Diagramme de cas d'utilisation du scénario « Gérer utilisateurs »	33
3.3.3	Cas d'utilisation « S'authentifier »	34
3.3.3.1	Description textuelle	34
3.3.3.2	Diagramme de séquence du scénario « S'authentifier »	34
3.3.4	Cas d'utilisation « Mettre à jour profil »	35
3.3.4.1	Description textuelle	35
3.3.4.2	Diagramme de séquence du scénario « Mettre à jour profil » .	36
3.3.5	Cas d'utilisation «Gérer projets»	38
3.3.5.1	Description textuelle	38
3.3.5.2	Raffinement du cas d'utilisation « Gérer projets »	38
3.4	Réalisation	39
3.4.1	Gérer utilisateurs	39
3.4.2	S'authentifier	40
3.4.3	Mettre à jour profil	40
3.4.4	Gérer projets	41

3.5 Conclusion	41
4 Sprint 2 :Gestion de documents	42
Webographie	43



LISTE DES FIGURES

1.1	Axia solutions	4
1.2	Compétences des consultants Axia solutions	4
1.3	Domaines d'activités pour les clients de Axia solutions	5
1.4	Jira	6
1.5	Trello	6
1.6	Cycle de vie de la méthode SCRUM	11
2.1	Diagramme de cas d'utilisation global	17
2.2	Diagramme de classes global	18
2.3	Architecture physique	21
2.4	Architecture microservices	22
2.5	Architecture Redux	23
2.6	.Net	24
2.7	.Net web api	24
2.8	React Js	25
2.9	SQLServer	25
2.10	Aws Cloud	26
2.11	GitHub	26
2.12	Swagger	26
2.13	Figma	27
2.14	Docker	27
2.15	Maquettes Axia-Agile	28
3.1	Diagramme de cas d'utilisation Sprint1	32
3.2	Diagramme de cas d'utilisation du scénario « Gérer utilisateurs »	33
3.3	Diagramme de séquence du scénario « S'authentifier »	35
3.4	Diagramme de séquence du scénario « Mettre à jour profil »	37
3.5	Raffinement du cas d'utilisation « Gérer projets »	39
3.6	Liste des utilisateurs	40

3.7	Formulaire de création d'utilisateur	40
3.8	Interface login	40
3.9	Login Erreur	40
3.10	Interface profile «Consulter»	41
3.11	Interface profile «mettre à jour»	41
3.12	Interface projets	41
3.13	Formulaire pour la création de projet	41



LISTE DES TABLEAUX

1.1	Étude comparative des méthodologies	8
2.1	Product Backlog	19
3.1	Backlog Sprint 1	31
3.2	Description textuelle du cas d'utilisation « créer utilisateurs »	32
3.3	Description textuelle du cas d'utilisation « S'authentifier »	34
3.4	Description textuelle du cas d'utilisation « Mettre à jour profil »	36
3.5	Description textuelle du cas d'utilisation « Gérer projets »	38



LISTE DES ABRÉVIATIONS

.Net = Dot Net

AWS = Amazon Web Services

OS = Operating System

ORM = Object Relational Mapping

URL = Uniform Resource Locator

API = Application Programming Interface

JWT = JSON Web Token

HTTP = Hyper Text Transfer Protocol

MVT = Model View Template

QR = Quick Response

UML = Unified Modeling Language

JS = JavaScript

UX = User Experience

UI = User Interface



INTRODUCTION GÉNÉRALE

Etude préalable

Plan

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

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 de Axia solutions consistent à fournir des experts qui maîtrisent les technologies présentées ci-dessous.

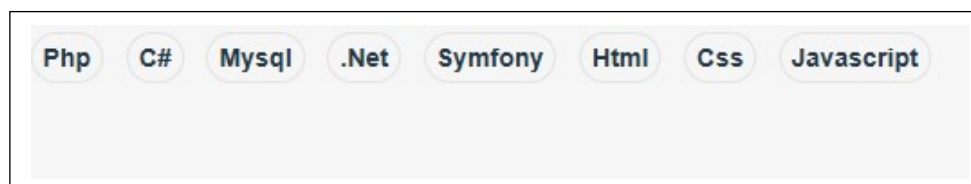


FIGURE 1.2 – Compétences des consultants Axia solutions

L'organisation consacre ses efforts sur le conseil 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.

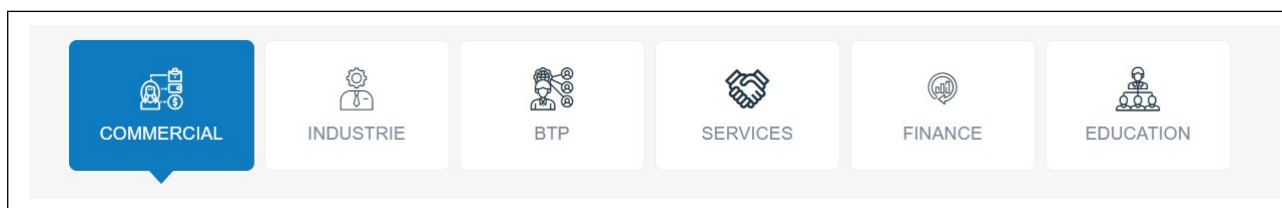


FIGURE 1.3 – Domaines d’activités pour les clients de Axia solutions

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

1.3.2.2 Critique de l'existant

Jira est un outil de gestion du travail pour les équipes de développement qui doivent s'organiser et suivre leur travail.[3].

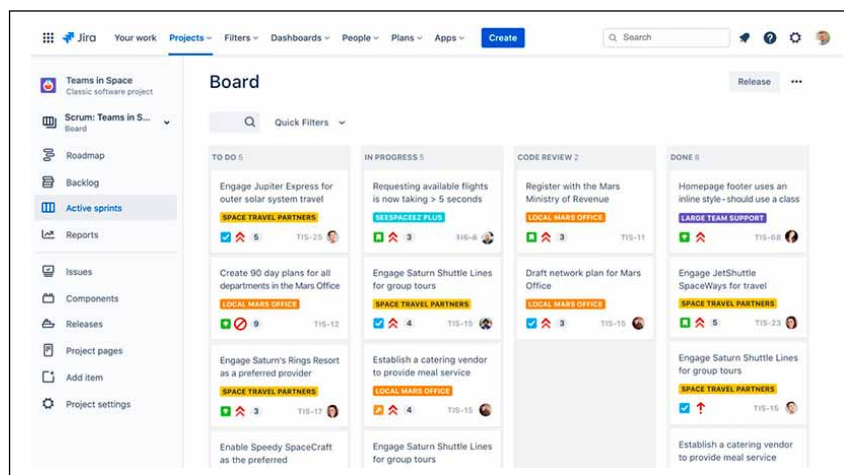


FIGURE 1.4 – Jira

Trello est l'outil de gestion visuelle du travail qui permet aux équipes de proposer des idées, de planifier, de gérer et de célébrer leur travail ensemble de manière collaborative, productive et organisée.[4].



FIGURE 1.5 – Trello

C'est vrai que les solutions mentionnées ci-dessus assurent la gestion agile des projets, qui forme une fonctionnalité primordiale dans notre projet, mais nous ne pouvons pas les exploiter dans notre application puisqu'elles manquent de quelques fonctionnalités telles que :

- Système de messagerie instantanée
- Gestion des réunions

1.3.2.3 Solution proposée

Afin de pallier aux différents problèmes cités dans la section 1.3.1, 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.

1.3.2.4 Objectif du projet

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 1.1 – Étude comparative des méthodologies

Thème	Approche classique	Approche SCRUM 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.

1.4.3 Les rôles dans SCRUM

Scrum est une méthodologie agile qui nécessite la présence d'une équipe adhérente qui oriente le projet au fil de son avancement afin d'atteindre un but intervenant trois rôles majeurs qui sont :

Le Product Owner : Dans la plupart des projets, le responsable produit (Product Owner) est le directeur de l'équipe projet client. Dans notre projet, le Product Owner est présenté par un expert métier afin de :

- Identifier la liste des fonctionnalités du produit.
- Déterminer la date de livraison des versions ainsi que leurs contenus.
- Classer les fonctionnalités en fonction de leur importance et leur valeur ajoutée pour l'entreprise qu'il présente.
- Clarifier les besoins à l'équipe de développement si nécessaire.
- Valider les lots livrés avec l'équipe de développement.
- Assurer la communication entre l'équipe de développement et les parties prenantes.
- Maintenir et ajuster le Product Backlog en fonction de l'évolution des besoins du projet.

Le Scrum Master est présenté par Mr Chouchane Mohamed Mokhtar ; c'est le leader de l'équipe, il accomplit les tâches suivantes :

- S'assurer que Scrum est bien appliquée et respectée.
- Eliminer les obstacles pouvant perturber la progression du travail.
- Encourager l'équipe à apprendre et à progresser pour qu'elle soit productive, fonctionnelle et créative durant le projet.

L'équipe de développement est présentée par Nermin Mahdoui. Elle regroupe tous les rôles habituellement nécessaires à un projet. Elle est responsable de la conception, de l'architecture, du développement, du test... etc. Son rôle principal consiste à :

- Transformer les besoins exprimés dans le Sprint Backlog (est défini ci-dessous) en fonctionnalités utilisables.
- Livrer régulièrement une version fonctionnelle du produit.

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

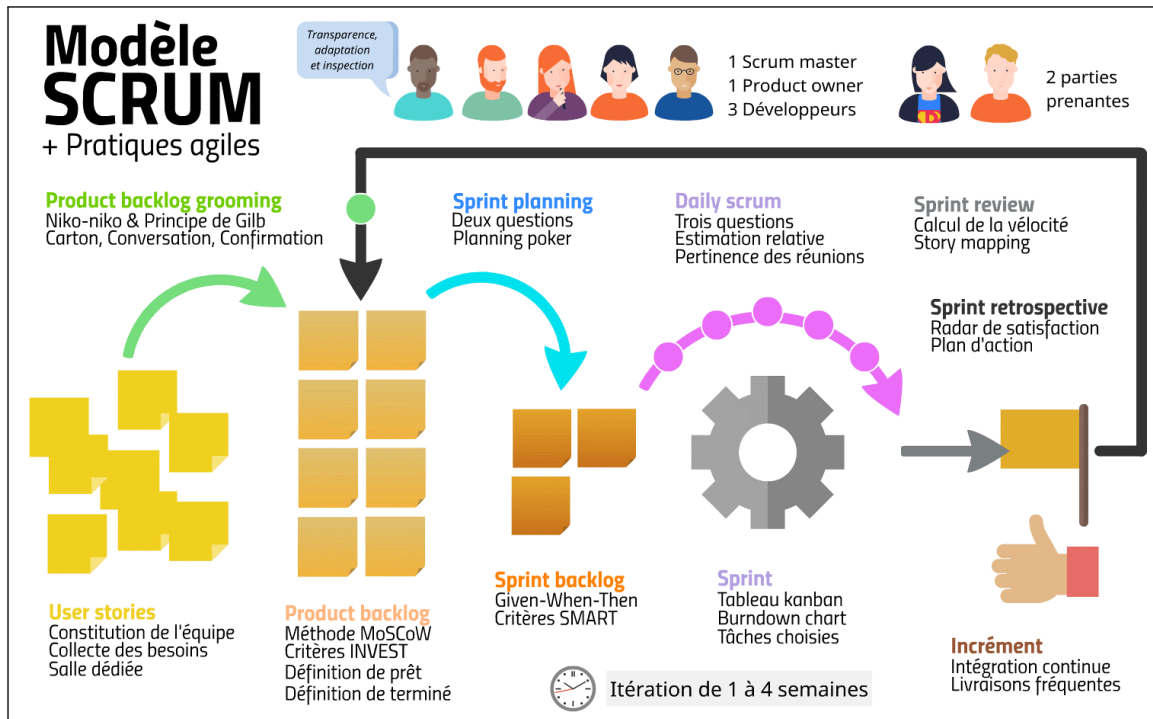


FIGURE 1.6 – Cycle de vie de la méthode SCRUM

Backlog produit (ou catalogue des besoins)

Il présente :

- Les besoins priorisés par le Product Owner.
- Les besoins évalués par l'équipe.

Sprint (itération)

Il exige :

- Développement des fonctionnalités du Backlog de sprint.
- Aucune modification du Backlog de sprint possible.
- Mêlée quotidienne (Rencontre quotidienne).
- Point de contrôle quotidien de l'équipe.
- Intervention régulière – 2 min par personne.

Sprint backlog c'est un extrait du Backlog produit.

Produit livrable est livré au product owner à la fin du sprint.

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.

Sprint 0 :Spécification et analyse des besoins

Plan

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

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[18]. Pour notre application, nous avons identifié 4 acteurs :

- Un super administrateur : Il représente le niveau le plus élevé d'administration.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 par l'administrateur, il peut créer et gérer des projets, affecter des membres et suivre l'avancement des projets.
- Un utilisateur :Il représente un membre de l'équipe.Ses permissions sont définies par l'administrateur lors de la création de son compte. Selon les permissions attribuées, il peut gérer les tâches, mettre à jour leur statut et participer aux activités du projet.

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 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 chef de projet ou admin peut créer des projets en affectant des membres à ces projets, les modifier et les supprimer.

- **Gestion des tâches :**

Chaque chef projet ou utilisateur autorisé peut créer, consulter, modifier, affecter ou supprimer des tâches.

- **Organisation de tableau kanban :**

Chaque chef projet ou admin peut ajouter des colonnes dans le tableau Kanban et créer, organiser des tâches au sein de ces colonnes.

- **Consultation de kanban :**

Chaque utilisateur , peut consulter ses tâches assignées dans le tableau Kanban.

- **Déplacement des tâches :**

Tout utilisateur peut déplacer leurs tâches d'une colonne à l'autre dans le tableau Kanban.

- **Gestion des sprint et backlog :**

Chaque chef projet ou utilisateur autorisé, peut créer et gérer un backlog, et de diviser le backlog en sprints.

- **Consultation dashboard :**

Chaque chef projet ou admin peut consulter le dashboad pour suivre l'avancement des projet et des sprints.

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**
Notre application assure des interfaces bien présentées et bien guidées.
- **La performance**
Le temps d'exécution du système doit être faible.
- **L'Extensibilité**
Le code doit être clair pour qu'il puisse avoir une possibilité d'ajouter des futures évolutions ou améliorations.
- **La fiabilité**
L'application fonctionne sans risque d'erreurs ou de pannes.

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.

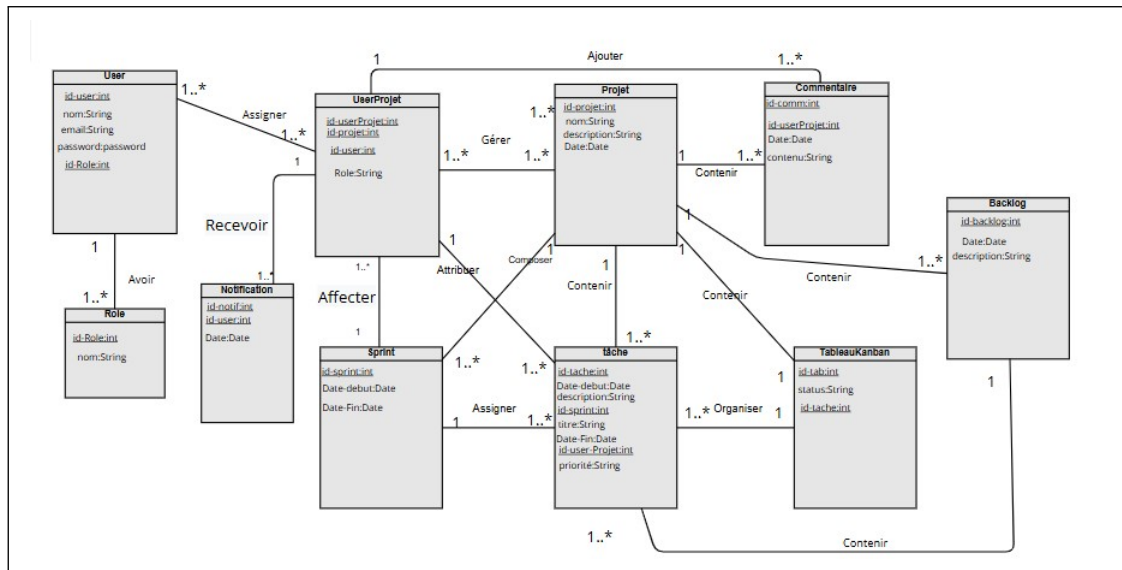


FIGURE 2.2 – Diagramme de classes global

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">– En tant qu'utilisateur de l'application, pour que je puisse m'accéder aux différentes phases de l'application, je dois m'authentifier.
	gérer les utilisateurs	<ul style="list-style-type: none">– En tant que super administrateur de l'application, je peux créer, modifier, supprimer, activer et désactiver des comptes administrateurs.– En tant qu'administrateur de l'application, je peux créer modifier, supprimer, activer et désactiver des comptes utilisateurs avec des droits d'accès spécifiques.
	Gérer profil	<ul style="list-style-type: none">– En tant qu'utilisateur de l'application, je peux consulter et mettre à jour mon profil.
	gérer les projets	<ul style="list-style-type: none">– En tant que chef de projet, administrateur , je peux créer, modifier, supprimer des projets.

2	<p>Gérer les tâches</p> <p>Organiser Tableau kanban</p> <p>Consulter tableau kanban</p> <p>Déplacer les tâches</p>	<ul style="list-style-type: none"> – En tant que chef de projet, ou utilisateur autorisé, je peux créer, modifier, supprimer des tâches. – En tant que chef de projet, je peux ajouter des colonnes dans le tableau Kanban et créer, organiser des tâches au sein de ces colonnes. – En tant qu'utilisateur, je peux consulter mes tâches dans le tableau kanban. – En tant qu'utilisateur, chef projet, je peux déplacer mes tâches d'une colonne à l'autre dans le tableau Kanban.
3	<p>Gérer backlog</p> <p>Gérer sprint</p> <p>Consulter dashboad</p> <p>Test et validation</p>	<ul style="list-style-type: none"> – En tant que chef de projet ou utilisateur autorisé, je peux gérer le backlog(créer, modifier et supprimer) et le diviser en des sprints. – En tant que chef de projet ou utilisateur autorisé, je peux créer, modifier, supprimer et lancer un sprint. – En tant que chef de projet ou admin, je peut consulter le dashboad pour suivre l'avancement des projet et des sprints. – Création des images docker pour le frotend et le backend et création d'un vue pour le test des api's.

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. [12]

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.

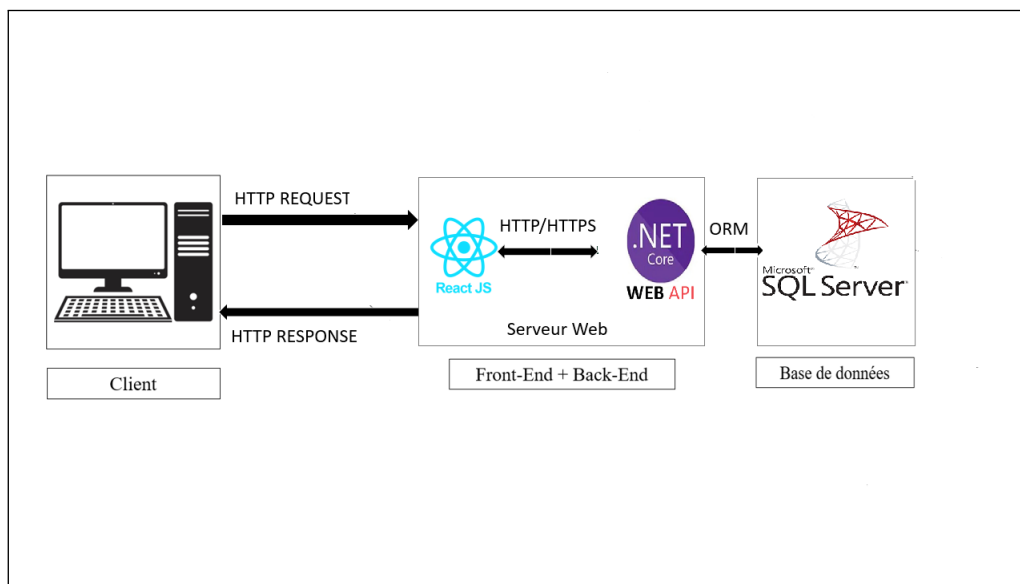


FIGURE 2.3 – Architecture physique

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. [20].

- **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.

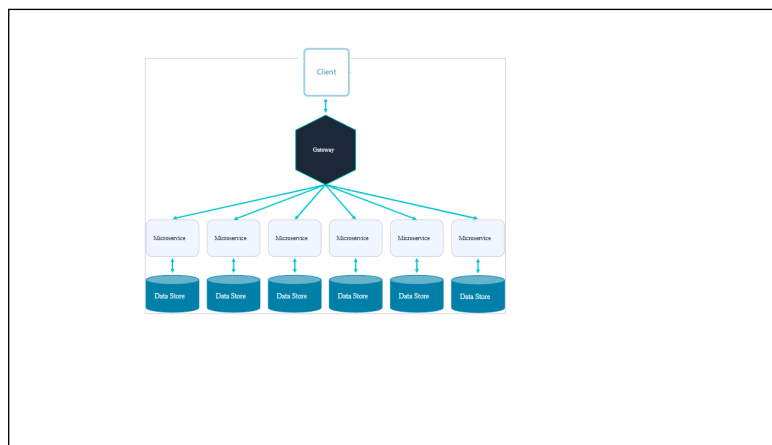


FIGURE 2.4 – Architecture microservices

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.

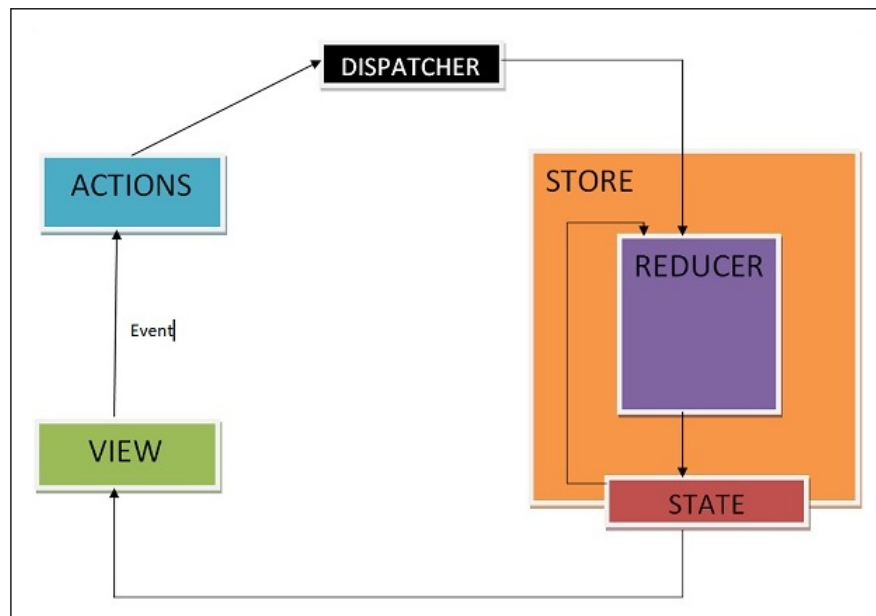


FIGURE 2.5 – Architecture Redux

-**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 [8]

.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[9]

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.

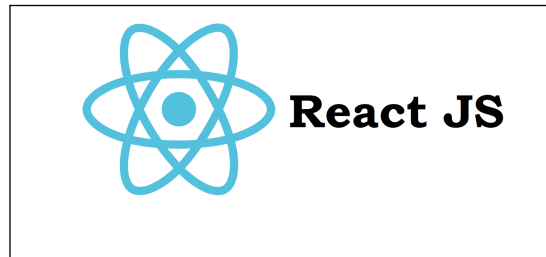


FIGURE 2.8 – React Js

-SQLServer[10]

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 [11]

Pour le déploiement, nous avons choisi Cloud AWS qui est une plateforme qui nous fournit un service de stockage Cloud. [7] 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[12]**

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. [17].



FIGURE 2.12 – Swagger

- Figma [13]

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 [14]

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. [19].



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[15].

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.

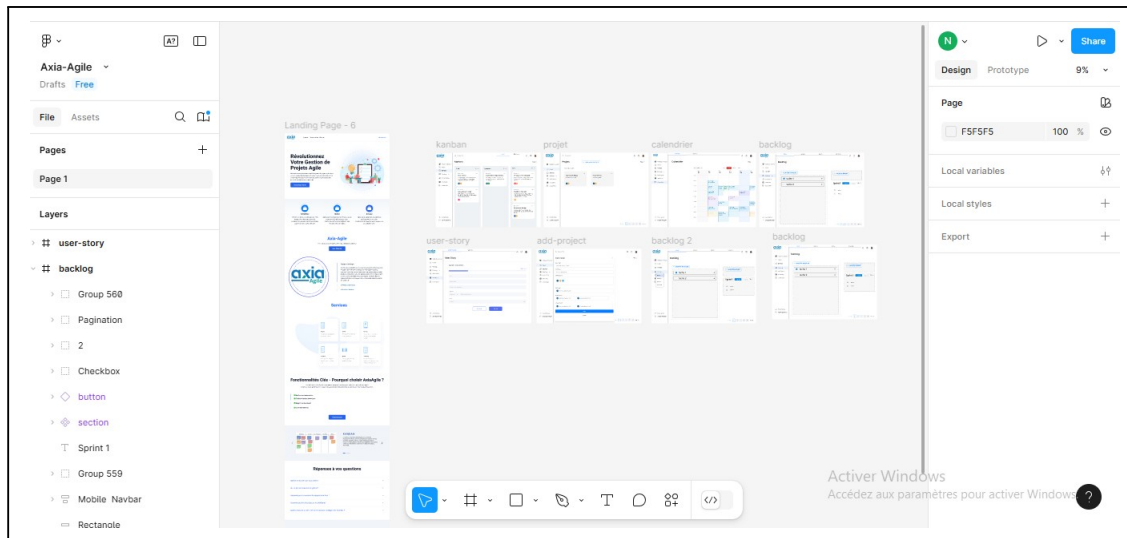


FIGURE 2.15 – Maquettes Axia-Agile

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

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

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

Nom	Description	Estimation (en jour)
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
Gérer profil	En tant qu'utilisateur de l'application, quel que soit mon rôle, je peux consulter et mettre à jour mon profil.	3
Gérer projet	En tant qu'chef projet ou admin de l'application, je peux créer , modifier , supprimer des projets .	3

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.

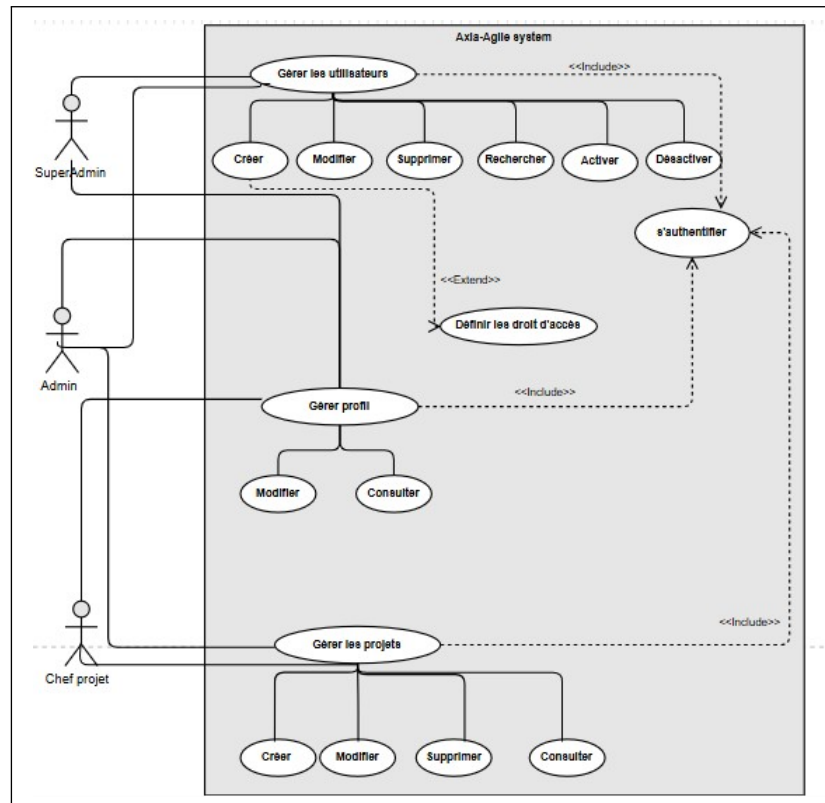


FIGURE 3.1 – Diagramme de cas d'utilisation Sprint1

3.3.2 Cas d'utilisation « Gérer utilisateurs »

3.3.2.1 Description textuelle

TABLE 3.2 – 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"> 1. L'admin ou le superadmin ouvre l'interface de gestion des utilisateurs. 2. Le système affiche le formulaire de création de compte utilisateur.

	<ul style="list-style-type: none"> – 3. L'utilisateur (admin/superadmin) remplit le formulaire de création de compte. – 4. L'utilisateur(admin/superadmin) définit les autorisations pour le compte. – 5. Le système vérifie les informations saisies. – 6. Le système ajoute l'utilisateur à la base de données.
Scénario alternatif	<ul style="list-style-type: none"> – 4.1 Les données saisies sont incorrectes ou les champs du formulaire sont vides ou l'email est déjà utilisé. – 4.1.1 Un message d'erreur s'affiche pour recommencer la saisie des données. – 4.1.2 Reprise de l'étape 3 du Scénario nominal.

3.3.2.2 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.

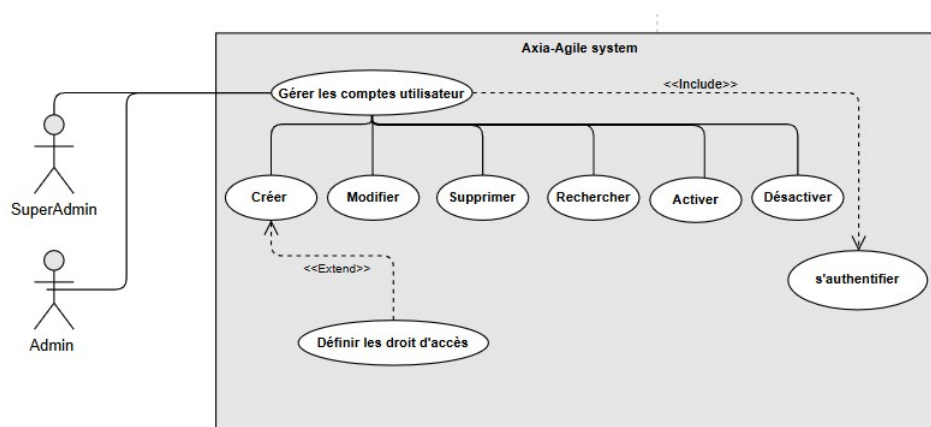


FIGURE 3.2 – Diagramme de cas d'utilisation du scénario « Gérer utilisateurs »

3.3.3 Cas d'utilisation « S'authentifier »

3.3.3.1 Description textuelle

TABLE 3.3 – 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">– 1. L'utilisateur lance l'application.– 2. Le système affiche l'interface d'authentification.– 3. L'utilisateur saisit ses identifiants (E-mail et mot de passe).– 4. Le système vérifie les informations saisies par l'utilisateur.– 5. Le système va générer un "Token" pour l'utilisateur authentifié.– 6. Le système affiche l'interface "Home".
Scénario alternatif	<ul style="list-style-type: none">– 4.1 Les données saisies sont incorrectes ou les champs du formulaire sont vides ou le compte déjà désactiver.– 4.1.1 Un message d'erreur s'affiche pour recommencer la saisie des données.– 4.1 Reprise de l'étape 2 du Scénario nominal.

3.3.3.2 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.

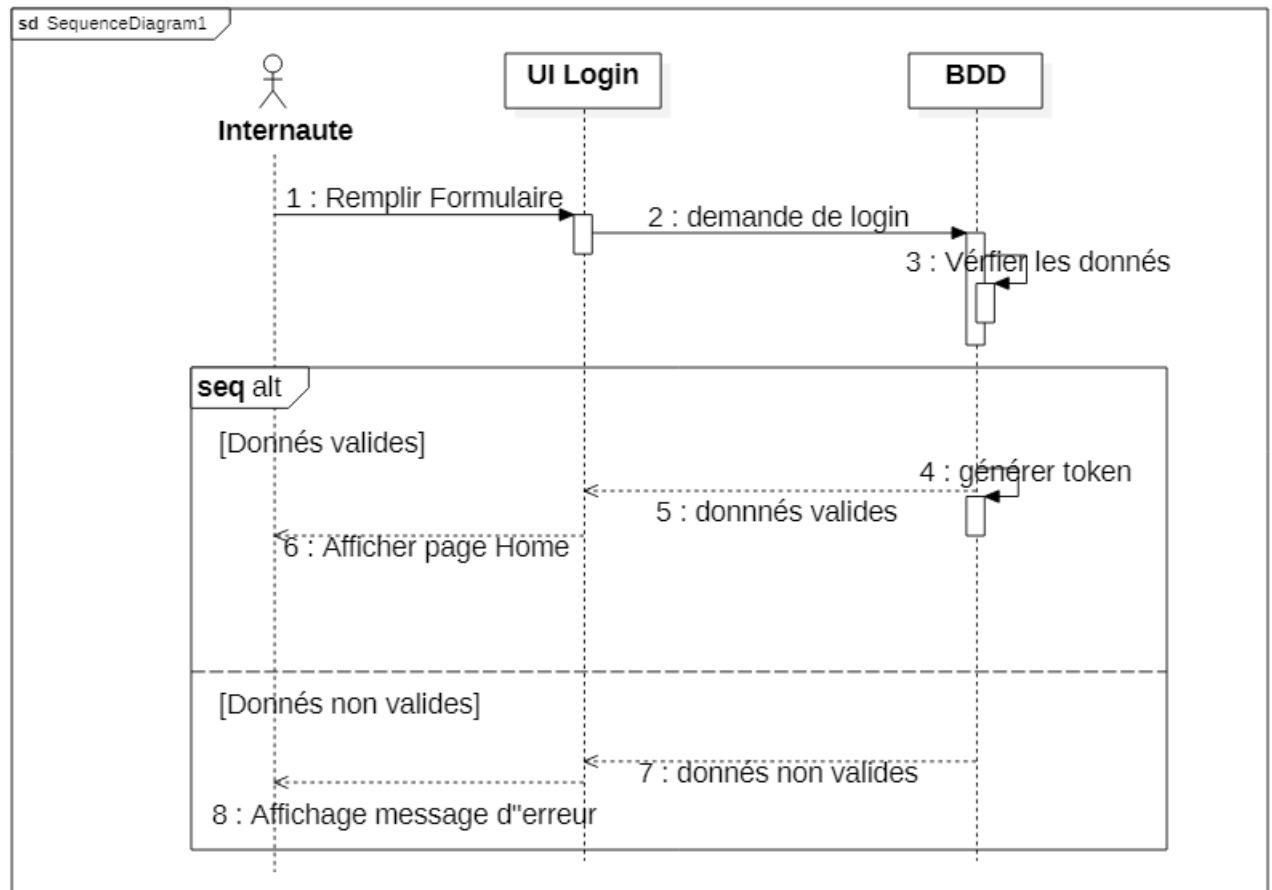


FIGURE 3.3 – Diagramme de séquence du scénario « S'authentifier »

3.3.4 Cas d'utilisation « Mettre à jour profil »

3.3.4.1 Description textuelle

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

TABLE 3.4 – Description textuelle du cas d'utilisation « Mettre à jour profil »

Cas d'utilisation	Mettre à jour profil
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"> – 1. L'utilisateur s'accède à la page "profil". – 2. L'utilisateur saisit les informations modifiées dans les formulaires de modification. – 3. L'utilisateur valide les modifications. – 4. Le système vérifie les données modifiées par l'utilisateur. – 5. Le système affiche un message de succès de modification.
Scénario alternatif	<ul style="list-style-type: none"> – 4.1 Les données saisies sont invalides ou vides. – 4.1.1 Un message d'erreur s'affiche pour recommencer la saisie des données. – 4.1.2 Reprise de l'étape 2 du Scénario nominal.

3.3.4.2 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.

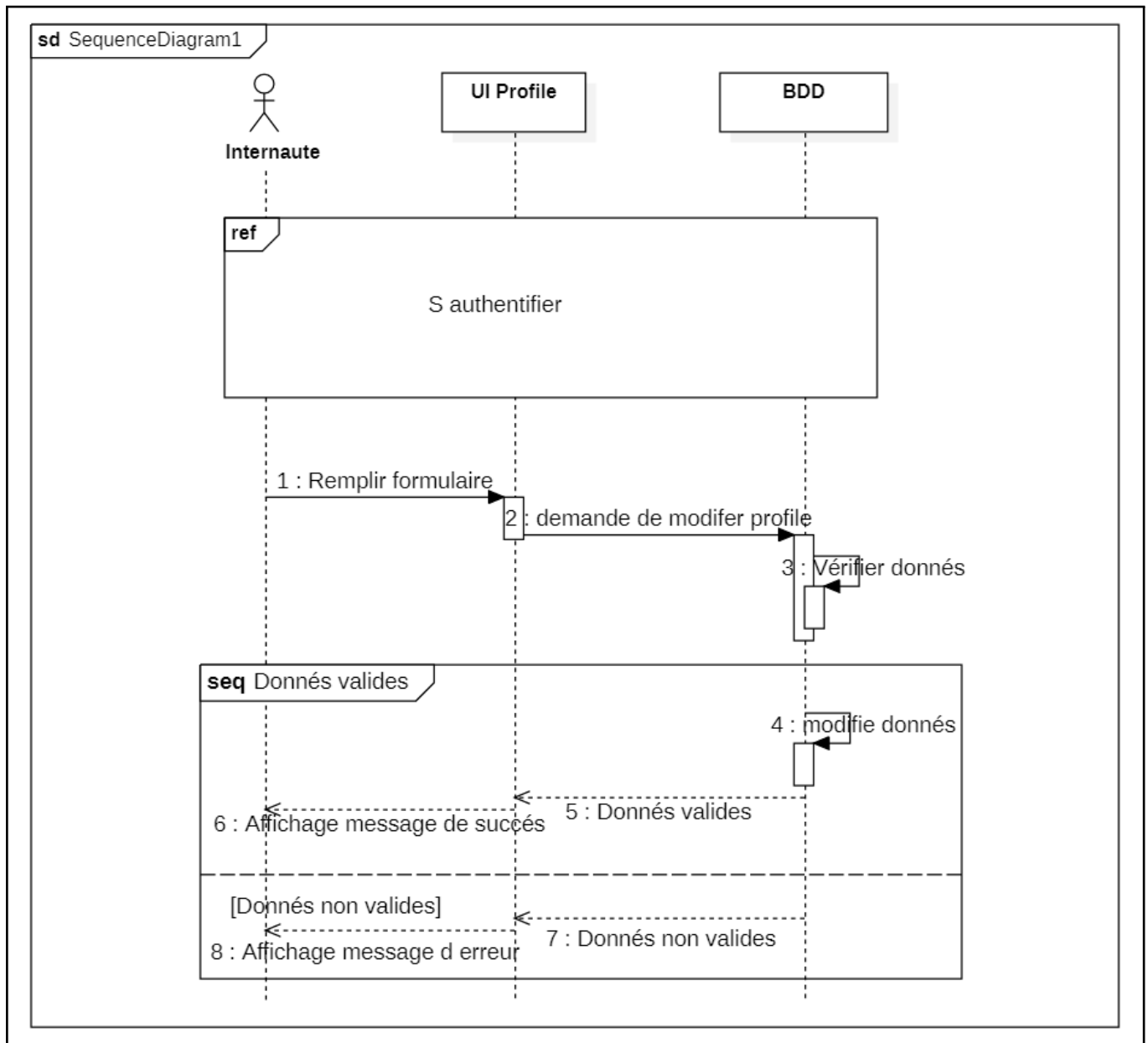


FIGURE 3.4 – Diagramme de séquence du scénario « Mettre à jour profil »

3.3.5 Cas d'utilisation «Gérer projets»

3.3.5.1 Description textuelle

TABLE 3.5 – 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"> - 1. L'utilisateur se connecte sur son compte . - 2. L'utilisateur s'accède à la page « Projets ». - 3. Le système affiche les projets de l'utilisateur. - 4. L'utilisateur peut créer des projets en appuyant sur le bouton "créer" . - 5. L'utilisateur peut modifier ou supprimer ses projets en appuyant sur le menu de projets.
Scénario alternatif	<ul style="list-style-type: none"> - 3.1.1 L'utilisateur n'a aucun projet crée. - 3.1.2 Un message s'affiche "Vous n'avez aucun projet".

3.3.5.2 Raffinement du cas d'utilisation « Gérer projets »

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

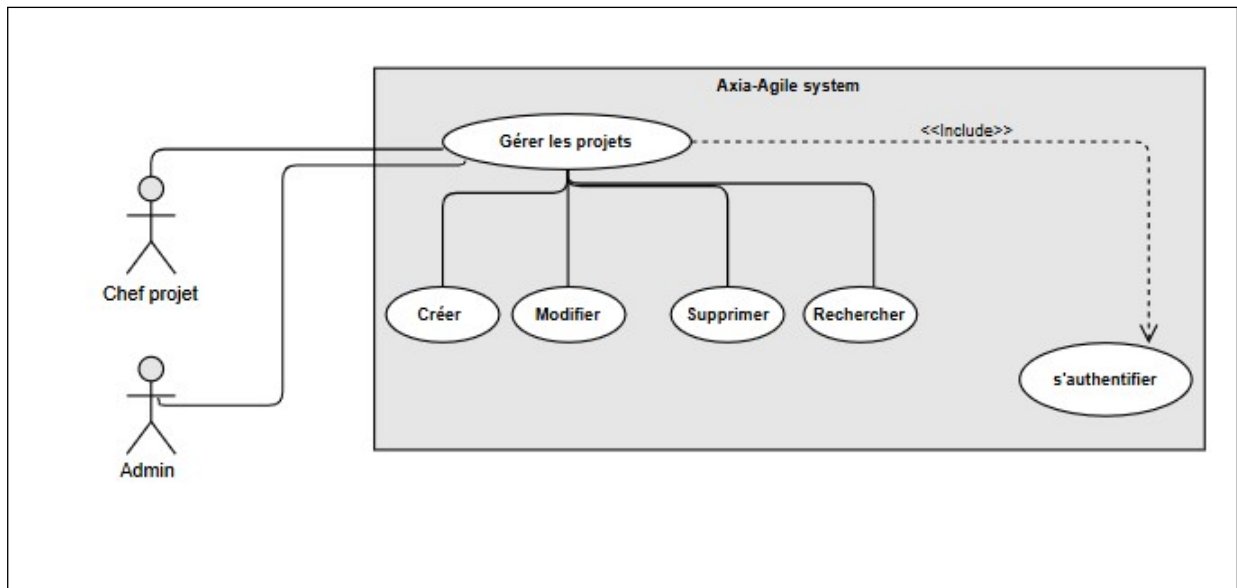


FIGURE 3.5 – Raffinement du cas d'utilisation « Gérer projets »

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.

3.4 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.

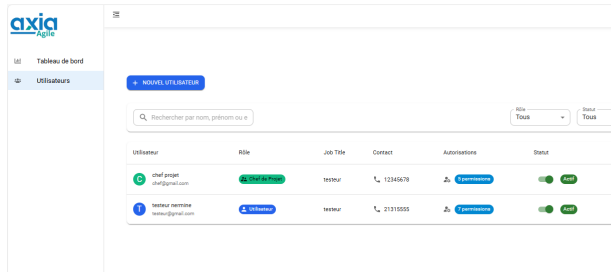


FIGURE 3.6 – Liste des utilisateurs

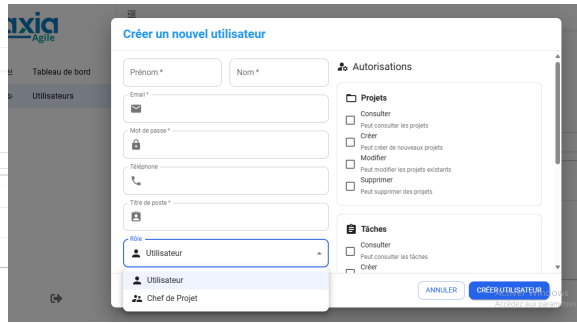


FIGURE 3.7 – Formulaire de création d'utilisateur

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.

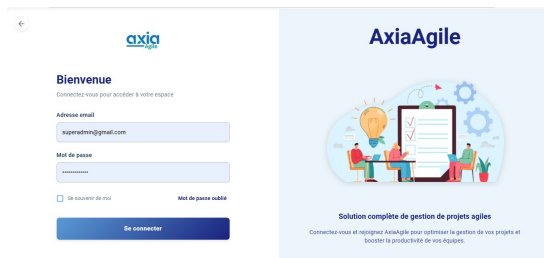


FIGURE 3.8 – Interface login

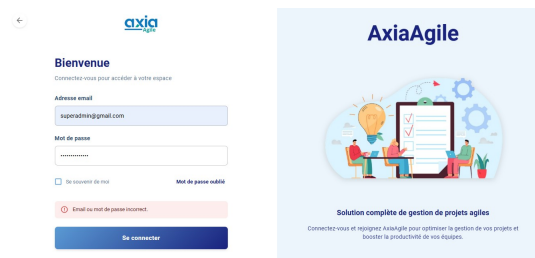


FIGURE 3.9 – Login Erreur

3.4.3 Mettre à jour profil

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

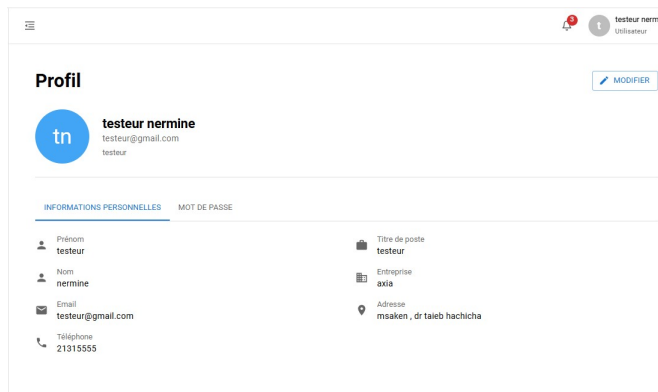


FIGURE 3.10 – Interface profile «Consulter»

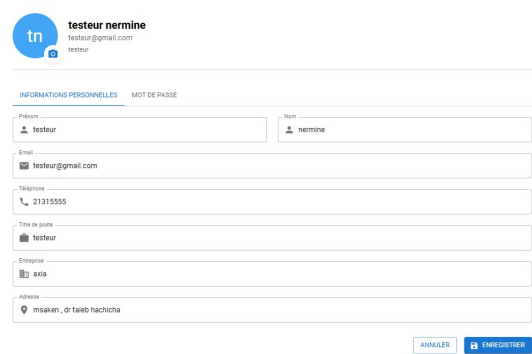


FIGURE 3.11 – Interface profile «mettre à jour»

3.4.4 Gérer projets

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

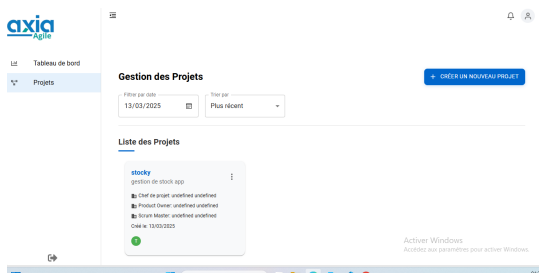


FIGURE 3.12 – Interface projets

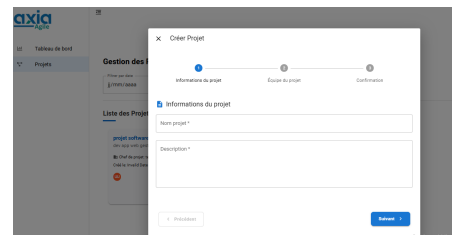


FIGURE 3.13 – Formulaire pour la création de projet

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 de documents

Plan

1	Introduction	48
2	Identification des tâches	49
3	Analyse des besoins	50
4	Réalisation	57
5	Conclusion	60



ANNEXES



Webographie

- [1] [Accès le 04-Mars-2025], <https://axiasolution.com/>
- [2] [Accès le 18-Mars-2025], <https://cloud.google.com/learn/what-is-microservices-architecture?hl=fr>
- [3] [Accès le 10-Mars-2025], <https://www.atlassian.com/fr/software/jira/guides/getting-started/basicsstep-1-c>
- [4] [Accès le 21-Mars-2025] <https://www.kodea.fr/outils/swagger/>
- [5] [Accès le 09-Avril-2021], <https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours04.html>
<https://cloud.google.com/learn/what-is-microservices-architecture?hl=fr>
- [6] [Accès le 10-Avril-2021], <https://framalibre.org/content/visual-studio-code>
- [7] [Accès le 12-Avril-2021], <https://www.lebigdata.fr/amazon-cloud-amazon-web-services>
- [8] [Accès le 21-Mars-2025], <https://www.guru99.com/fr/net-framework.html>
- [9] [Accès le 05-Avril-2021], <https://fr.reactjs.org/>
- [10] [Accès le 05-Avril-2021], <https://www.mongodb.com/try/download/community>
- [11] [Accès le 05-Avril-2021], <https://aws.amazon.com/fr/>
- [12] [Accès le 18-Mars-2025], <https://www.redhat.com/fr/topics/application-development-and-delivery/une-a>
- [13] [Accès le 05-Avril-2021], <https://trello.com/>
- [14] [Accès le 15-Aout-2021], <https://www.docker.com/>
- [15] [Accès le 19-Avril-2021], <https://www.alphalives.com/agence-web-paris/projet-web/maquette-web>
- [16] [Accès le 13-Avril-2021], <https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-a>
[pi-web/](https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-a)
- [17] [Accès le 14-Avril-2021], <https://www.blogdumoderateur.com/tools/slack/>
- [18] [Accès le 14-Avril-2021], <https://www.blogdumoderateur.com/tools/trello/>
- [19] [Accès le 15-Avril-2021], <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1146>
[290-docker-definition-docker-compose-docker-hub-docker-swarm-160919/](https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1146)

[20] [Accès le 17-Avril-2021], <https://explorweb.github.io/cours2018A/cours/django/>

[21] [Accès le 20-Avril-2021], <https://www.twilio.com/docs/usage/api>

[22] [Accès le 5-Juin-2021], <https://aws.amazon.com/fr/s3/>

[23] [Accès le 25-Avril-2021], <https://www.ubidreams.fr/ant-design/>