

## Encadrant Entreprise

J'autorise l'étudiant à déposer son rapport de stage en vue d'une soutenance.

**Signature et cachet :**

## Encadrant TEK-UP

J'autorise l'étudiant à déposer son rapport de stage en vue d'une soutenance.

**Signature :**

# DÉDICACE

Ce travail est dédié :

À mes parents, en guise de reconnaissance et de gratitude  
pour les sacrifices qu'ils ont faits.

À mon frère, à qui je dois tout l'amour, avec tous mes voeux  
de le voir réussir dans sa vie.

À mes ami(e)s, à qui je souhaite le succès, pour l'amitié qui  
nous a toujours unis.

À tous ceux qui me sont chers.

Achref Bouadila

# REMERCIEMENTS

Au terme de ce travail, nous voudrons remercier tous ceux qui, sans leur aide inestimable, ce projet n'aurait jamais été mené à son terme. Nos remerciements s'adressent particulièrement à :

Notre encadrant Monsieur **Khalil Driss**, le fondateur d'**INNOVUP**, à qui nous tenons à exprimer toute notre gratitude pour l'aide qu'il nous a apportée durant toutes les phases de ce stage. Sa disponibilité, son encadrement, sa pédagogie et ses conseils nous ont été précieux pour atteindre les objectifs de ce projet dans les délais convenus.

Notre superviseur Docteur **Hmida Rojbani** pour la qualité de son enseignement, son encouragement et ses conseils qui nous ont été très bénéfiques, sans oublier sa participation au cheminement de ce rapport.

Tous les enseignants qui ont participé à notre évolution scientifique durant les années écoulées à l'**École privée supérieure de technologie et d'ingénierie TEK-UP**

Notre dernier mot s'adresse à tous **les membres du jury** pour l'honneur qu'ils nous font de participer à l'examen de notre travail.

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Contexte général du projet</b>	<b>3</b>
1.1 Présentation de l'entreprise d'accueil . . . . .	4
1.2 Problématique . . . . .	5
1.3 Étude de l'existant . . . . .	5
1.3.1 Description . . . . .	6
1.3.2 Points forts et points faibles . . . . .	7
1.4 Solution proposée . . . . .	7
1.5 Méthodologie de travail . . . . .	8
1.6 Langage de modélisation . . . . .	9
1.7 Concepts théoriques des microservices . . . . .	9
1.7.1 Avantages de l'architecture microservices . . . . .	10
1.7.2 Inconvénients de l'architecture microservices . . . . .	10
<b>2 Analyse préliminaire</b>	<b>12</b>
2.1 Spécification des besoins . . . . .	13
2.1.1 Identification des acteurs . . . . .	13
2.1.2 Les besoins fonctionnels . . . . .	13
2.1.3 Les besoins non fonctionnels . . . . .	14

## TABLE DES MATIÈRES

---

2.1.4	Diagramme de cas d'utilisation global . . . . .	15
2.2	Pilotage de projet avec SCRUM . . . . .	16
2.2.1	Identification des rôles de SCRUM . . . . .	16
2.2.2	Le Product Backlog . . . . .	16
2.2.3	Planification des sprints . . . . .	19
2.3	Environnement de travail . . . . .	19
2.3.1	Environnement matériel . . . . .	20
2.3.2	Environnement logiciel . . . . .	20
2.3.3	Technologies utilisées . . . . .	21
2.4	L'architecture de la solution . . . . .	24
2.4.1	L'architecture l'application en microservices . . . . .	24
2.4.2	L'architecture n-tiers d'un microservice . . . . .	25
<b>3</b>	<b>Sprint 1 : Authentication, Authorisation, Recupera</b>	<b>27</b>
3.1	Analyse du sprint . . . . .	28
3.1.1	Diagrammes de cas d'utilisation . . . . .	28
3.1.2	Sprint Backlog . . . . .	31
3.2	Conception du sprint . . . . .	32
3.2.1	Le diagramme de classes . . . . .	32
3.2.2	les diagrammes de séquences . . . . .	32
3.3	Revue du sprint . . . . .	36
3.3.1	L'interface d'inscription . . . . .	36
3.3.2	L'interface d'authentification . . . . .	36
3.3.3	Les interfaces de récupération de mot de passe . . . . .	37
3.3.4	L'interface de gestion du profil . . . . .	38
<b>4</b>	<b>Sprint 2 : Gestion des projets et L'invitation des membres</b>	<b>39</b>
4.1	Analyse du sprint . . . . .	40
4.1.1	Diagrammes de cas d'utilisation . . . . .	40

## TABLE DES MATIÈRES

---

4.1.2	Sprint Backlog . . . . .	42
4.2	Conception du sprint . . . . .	42
4.2.1	Le diagramme de classes . . . . .	42
4.2.2	les diagrammes de séquences . . . . .	43
4.3	Revue du sprint . . . . .	45
4.3.1	L'interface de création d'un projet . . . . .	45
4.3.2	L'interface pour consulter les détails d'un projet . . . . .	45
4.3.3	L'interface pour inviter un Scrum Master . . . . .	46
4.3.4	L'interface pour inviter les membres d'équipe . . . . .	46
4.3.5	L'interface pour consulter la liste des projets . . . . .	47
<b>5</b>	<b>Sprint 3 : Gestion des sprints et gestion des tâches</b>	<b>48</b>
5.1	Analyse du sprint . . . . .	49
5.1.1	Diagrammes de cas d'utilisation . . . . .	49
5.1.2	Sprint Backlog . . . . .	50
5.2	Conception du sprint . . . . .	51
5.2.1	Le diagramme de classes . . . . .	51
5.2.2	les diagrammes de séquences . . . . .	53
5.3	Revue du sprint . . . . .	55
5.3.1	L'interface de création d'un sprint . . . . .	55
5.3.2	L'interface de création d'une tâche . . . . .	56
5.3.3	L'interface pour consulter « Product Backlog » et « Sprint Backlogs » . . . . .	56
5.3.4	L'interface de la liste des commentaires . . . . .	57
5.3.5	L'interface d'avancement des tâches « Scrumboard » . . . . .	57
<b>6</b>	<b>Sprint 4 : Tableau de bord, système de messagerie et système de notifications</b>	<b>59</b>
6.1	Analyse du sprint . . . . .	60
6.1.1	Diagrammes de cas d'utilisation . . . . .	60

## TABLE DES MATIÈRES

---

6.1.2	Sprint Backlog . . . . .	61
6.2	Conception du sprint . . . . .	62
6.2.1	Le diagramme de classes . . . . .	62
6.2.2	les diagrammes de séquences . . . . .	64
6.3	Revue du sprint . . . . .	65
6.3.1	L'interface de messagerie . . . . .	65
6.3.2	L'interface de la liste des notifications . . . . .	66
6.3.3	L'interface de tableau de bord . . . . .	66
	<b>Conclusion générale et perspectives</b>	<b>68</b>

# Table des figures

1.1	Logo de l'entreprise . . . . .	4
1.2	Jira . . . . .	6
1.3	Vivify Scrum . . . . .	6
1.4	Cycle de vie d'un projet Scrum . . . . .	8
1.5	Architecture monolithique et architecture microservices . . . . .	9
2.1	Diagramme de cas d'utilisation global . . . . .	15
2.2	Logo de Spring Tool Suite . . . . .	20
2.3	Logo de Visual Studio Code . . . . .	20
2.4	Logo de Github . . . . .	20
2.5	Logo de Postman . . . . .	20
2.6	Logo de MySQL Workbench . . . . .	21
2.7	Logo de Visual Paradigm . . . . .	21
2.8	Logo d'Overleaf . . . . .	21
2.9	Logo d'Angular . . . . .	21
2.10	Logo de Spring Boot . . . . .	22
2.11	Logo de RabbitMQ . . . . .	22
2.12	Logo de Websocket . . . . .	22
2.13	Logo de MySQL . . . . .	23
2.14	Logo de Docker . . . . .	23

## TABLE DES FIGURES

---

2.15 L'architecture microservices . . . . .	24
2.16 L'architecture n-tiers d'un microservice . . . . .	25
3.1 Le diagramme de cas d'utilisation global de sprint 1 . . . . .	28
3.2 Le diagramme de classes du sprint 1 . . . . .	32
3.3 Le diagramme de séquence « S'inscrire » . . . . .	33
3.4 Le diagramme de séquence « S'authentifier » . . . . .	34
3.5 Le diagramme de séquence « Autorisation » . . . . .	35
3.6 L'interface d'inscription . . . . .	36
3.7 L'interface d'authentification . . . . .	36
3.8 L'interface initiale de récupération de mot de passe . . . . .	37
3.9 L'email de récupération de mot de passe . . . . .	37
3.10 L'interface changment de mot de passe . . . . .	37
3.11 L'interface de gestion du profil . . . . .	38
4.1 Le diagramme de cas d'utilisation global de sprint 2 . . . . .	40
4.2 Le diagramme de classes du sprint 2 . . . . .	42
4.3 Le diagramme de séquence « Créer un nouveau projet » . . . . .	44
4.4 L'interface de création d'un projet . . . . .	45
4.5 L'interface pour consulter les détails d'un projet . . . . .	45
4.6 L'interface pour inviter un Scrum Master . . . . .	46
4.7 L'interface pour inviter les membres d'équipe . . . . .	46
4.8 L'interface pour consulter la liste des projets . . . . .	47
5.1 Le diagramme de cas d'utilisation global de sprint 3 . . . . .	49
5.2 Le diagramme de classes du sprint 3 . . . . .	52
5.3 Le diagramme de séquence « Démarrer un sprint » . . . . .	53
5.4 Le diagramme de séquence « Créer une nouvelle tâche » . . . . .	54
5.5 L'interface de création d'un sprint . . . . .	55
5.6 L'interface de création d'une tâche . . . . .	56

## TABLE DES FIGURES

---

5.7	L'interface pour consulter « Product Backlog » et « Sprint Backlogs » . . . . .	56
5.8	L'interface de la liste des commentaires . . . . .	57
5.9	L'interface d'avancement des tâches « Scrumboard » . . . . .	57
6.1	Le diagramme de cas d'utilisation global de sprint 4 . . . . .	60
6.2	Le diagramme de classes du sprint 4 . . . . .	63
6.3	Le diagramme de séquence « Envoyer un nouveau message »	64
6.4	L'interface de messagerie . . . . .	65
6.5	L'interface de la liste des notifications . . . . .	66
6.6	L'interface de tableau de bord . . . . .	66

# Liste des tableaux

1.1	Points forts et points faibles des applications « Jira » et « Vivify Scrum » . . . . .	7
2.1	Les rôles de SCRUM . . . . .	16
2.2	Product Backlog . . . . .	18
2.3	Planification des sprints . . . . .	19
3.1	Description textuelle de cas d'utilisation « S'inscrire » . . .	29
3.2	Description textuelle de cas d'utilisation « S'authentifier » .	29
3.3	Description textuelle de cas d'utilisation « Récupérer mot de passe oublié » . . . . .	30
3.4	Description textuelle de cas d'utilisation « Modifier les informations du profil » . . . . .	31
3.5	Backlog du sprint 1 . . . . .	31
4.1	Description textuelle de cas d'utilisation « Créer un nouveau projet » . . . . .	41
4.2	Description textuelle de cas d'utilisation « Inviter un Scrum Master » . . . . .	41
4.3	Backlog du sprint 2 . . . . .	42
5.1	Description textuelle de cas d'utilisation « Créer un nouveau sprint » . . . . .	50

## **LISTE DES TABLEAUX**

---

5.2 Backlog du sprint 3 . . . . .	51
6.1 Description textuelle de cas d'utilisation « Envoyer un nouveau message » . . . . .	61
6.2 Backlog du sprint 4 . . . . .	62

# Introduction générale

**L**e logiciel est devenu un facteur de succès pour toutes les entreprises, ces dernières passent à une fabrication où le composant logiciel est primordial dans leurs environnements, elles imposent certains critères pour le développement de leur logiciel qui est devenu leur projet initial, où il faut bien gérer leur budget pour un bon rendement.

Les entreprises de développement sont toujours à la recherche de la meilleure façon de gérer les projets en intégrant un environnement sain est idéal pour ses développeurs et leur limiter toutes interruptions inutiles pendant leur travail pour gagner en temps et en performance.

Les entreprises présentent une collaboration entre ses employées, ce qui incite leurs managers à envisager sérieusement de limiter les conflits entre eux, ce qui affecte fortement la performance de ces entreprises, nous essayons donc toujours de trouver un système qui assure une bonne entente pour assurer la performance requise, assurant ainsi la survie de l'entreprise.

Le présent rapport sera organisé de la manière suivante :

- Dans un premier chapitre, nous présenterons le contexte général du projet. Il sera constitué d'une présentation de l'entreprise d'accueil, problématique ainsi que d'une étude de l'existant. Ce premier chapitre contiendra aussi une présentation de la méthodologie de travail et du langage de modélisation que nous avons choisi d'adopter.
- Le second chapitre présentera une analyse et une spécification des

## **LISTE DES TABLEAUX**

---

besoins, fonctionnels et non fonctionnels, réalisés à partir de l'identification des acteurs de l'application. Il exposera aussi le diagramme de cas d'utilisation globale, ainsi que le product backlog de l'application, la répartition des sprints, l'environnement de travail et l'architecture de la solution.

- Le troisième, le quatrième, le cinquième et le sixième chapitre porteront sur les sprints, qui seront dédiées à l'analyse détaillée, la conception dynamique et statique ainsi que les revues des sprints afin de bien modéliser notre objectif.

A la fin, nous clôturons le présent rapport par une conclusion générale qui résume notre travail et nous exposons quelques perspectives qui peuvent l'enrichir.

Chapitre **1**

# Contexte général du projet

## Plan

1.1	Présentation de l'entreprise d'accueil . . . . .	4
1.2	Problématique . . . . .	5
1.3	Étude de l'existant . . . . .	5
1.4	Solution proposée . . . . .	7
1.5	Méthodologie de travail . . . . .	8
1.6	Langage de modélisation . . . . .	9
1.7	Concepts théoriques des microservices . . . . .	9

## Introduction

Dans ce chapitre nous présentons, dans une première partie, l'entreprise d'accueil « INNOVUP ». Par la suite, nous décrivons la problématique ainsi qu'une étude de l'existant et une présentation de la solution proposée. Dans un troisième volet, nous définissons la méthode de travail et le langage de modélisation adopté. Nous finissons par quelques concepts théoriques des microservices.

### 1.1 Présentation de l'entreprise d'accueil

INNOVUP est une entreprise tunisienne fondée en 2020 par Khalil Driss, qui est engagée dans le secteur informatique et des nouvelles technologies.

Les principales activités de l'entreprise sont de développer des applications innovantes et intelligentes à forte valeur ajoutée dans la vie des citoyens, et de participer au programme d'innovation et d'informatisation du gouvernement tunisien.

INNOVUP est ouverte à toute sorte de partenariat local ou international et se projette à explorer le marché européen et africain, notamment le projet « Digital Africa ».

La figure 1.1 ci-dessous présente le logo de l'entreprise « INNOVUP ».



**FIGURE 1.1** — Logo de l'entreprise

## **1.2 Problématique**

Comme toutes les entreprises, INNOVUP est confrontée à la problématique de la collaboration et de la coordination entre les acteurs du cycle de vie du projet dans son domaine professionnel d'intégration des applications métiers et de développement spécifique.

Les principales difficultés auxquelles elle fait face sont :

- Collaboration et coordination entre les acteurs qui participent sur le cycle de vie de réalisation des logiciels
- Planification et suivi des tâches
- Maitrise des méthodes agile de développement collaboratif des logiciels.

De plus, partout dans le monde, les projets informatiques ont connu des échecs répétés. Ces échecs sont dus à une mauvaise gestion. Si un projet est en retard de livraison ou que ce dernier dépasse le budget, il est considéré comme un échec.

## **1.3 Étude de l'existant**

Une étude de l'existant s'avère essentielle puisqu'elle fournit une base de référence pour la suite du projet comme elle sert à approfondir l'analyse des dimensions innovantes de notre travail.

Au cours de cette étape, nous procédons à la revue de quelques applications similaires et nous choisissons de faire l'étude de quelques applications de gestion des projets.

## CHAPITRE 1. CONTEXTE GÉNÉRAL DU PROJET

### 1.3.1 Description

Jira [1] est une suite de solutions de gestion du travail agile qui facilite la collaboration entre équipes, du concept au client, et vous permet d'optimiser tous ensemble votre travail.

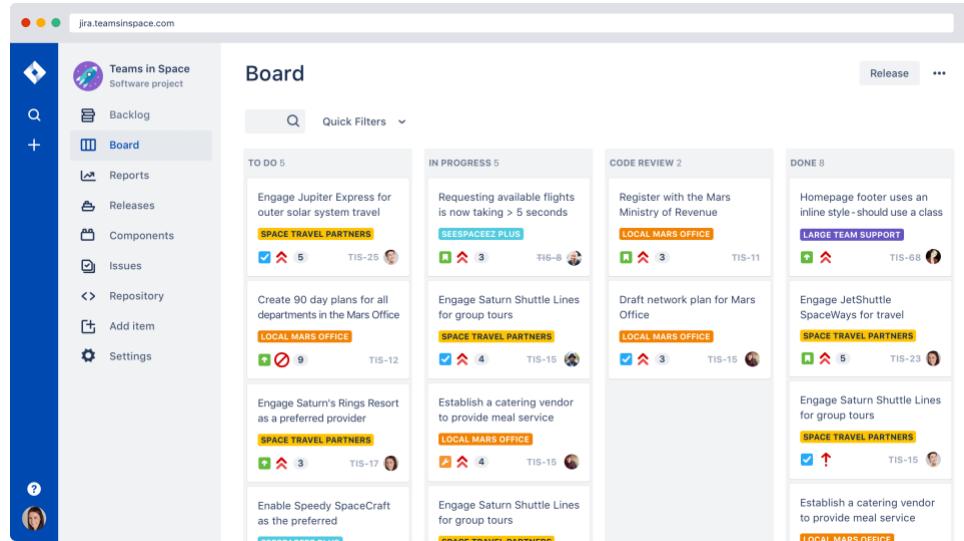


FIGURE 1.2 — Jira

Vivify Scrum [2] est une solution de gestion de projet agile basée sur le cloud pour gérer tous les aspects d'une entreprise. Elle convient aussi bien aux petites équipes agiles qu'aux grandes entreprises.

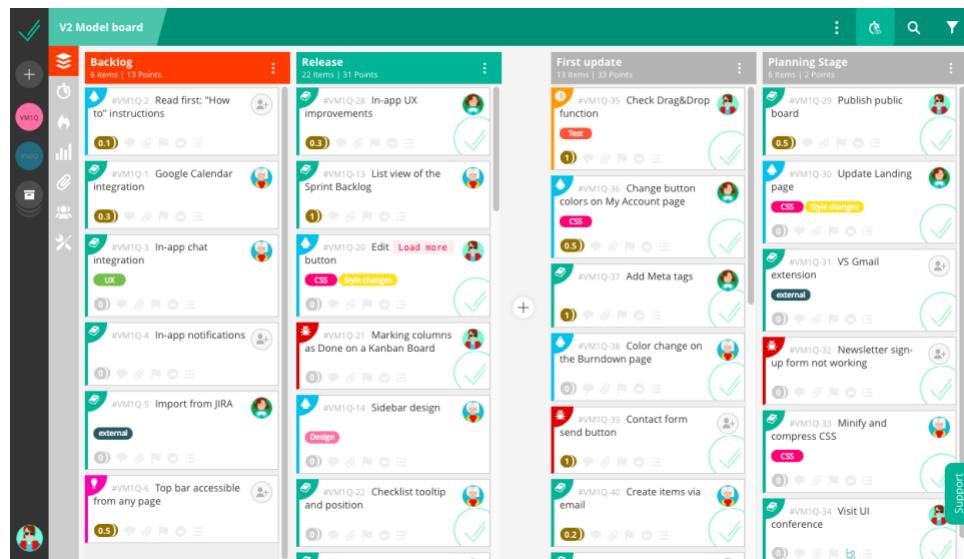


FIGURE 1.3 — Vivify Scrum

### 1.3.2 Points forts et points faibles

Solution	Points forts	Point faibles
Jira	<ul style="list-style-type: none"> <li>— Suivi des tâches très simple et réactif, très adapté au travail en équipe.</li> <li>— Hautement personnalisable.</li> <li>— Association avec d'autres applications (Bitbucket, Trello, ...).</li> </ul>	<ul style="list-style-type: none"> <li>— La configuration d'un projet nécessite pas mal de temps.</li> <li>— Demande un long temps d'adaptation.</li> <li>— Pas d'espace de discussion.</li> <li>— Solution Payante.</li> </ul>
Vivify Scrum	<ul style="list-style-type: none"> <li>— Convient aux entreprises de toute taille.</li> <li>— Bien suivre la méthodologie SCRUM.</li> </ul>	<ul style="list-style-type: none"> <li>— Les interfaces sont parfois compliquées à utiliser.</li> <li>— Pas d'espace de discussion.</li> <li>— Solution Payante.</li> </ul>

TABLE 1.1: Points forts et points faibles des applications « Jira » et « Vivify Scrum »

## 1.4 Solution proposée

Ce projet consiste à concevoir et développer une application web permettant la gestion des projets en se basant sur la méthodologie SCRUM qui garantit une haute disponibilité et une bonne performance, l'utilisation des nouveaux outils de développement est indispensable avec une architecture solide et évolutive.

L'interaction des utilisateurs avec l'application est une partie importante pour les fidéliser, nous avons pensé à mettre en place des interfaces simples avec des actions faciles à utiliser en réduisant le passage entre les interfaces afin de garantir que l'utilisateur accède à l'information la plus rapide possible sans passer par un workflow lourd.

Cette application doit favoriser la communication entre les collaborateurs du projet en mettant en place un espace de discussion et un système de notification en temps réel.

## 1.5 Méthodologie de travail

Les méthodes Agiles représentent un ensemble de pratiques utilisé pour la conception de logiciel. Elles favorisent l’implication continue du client et permettent une grande réactivité à ses demandes. Nous avons choisi la méthodologie **SCRUM** [3] pour la conception et le développement de notre application. Le principe de la méthode Scrum est de développer un logiciel d’une manière incrémentale, progressive et en plusieurs étapes.

Comme le montre la Figure 1.4, pour mettre en place la méthode Scrum, nous commençons par préciser les différentes fonctionnalités de notre solution qui forment le **Product Backlog**. Par la suite, nous passons à la planification des sprints. Un **sprint** indique la période qui dure de 2, 3 ou 4 semaines dans laquelle un travail demandé doit être effectué avant d’être audité.

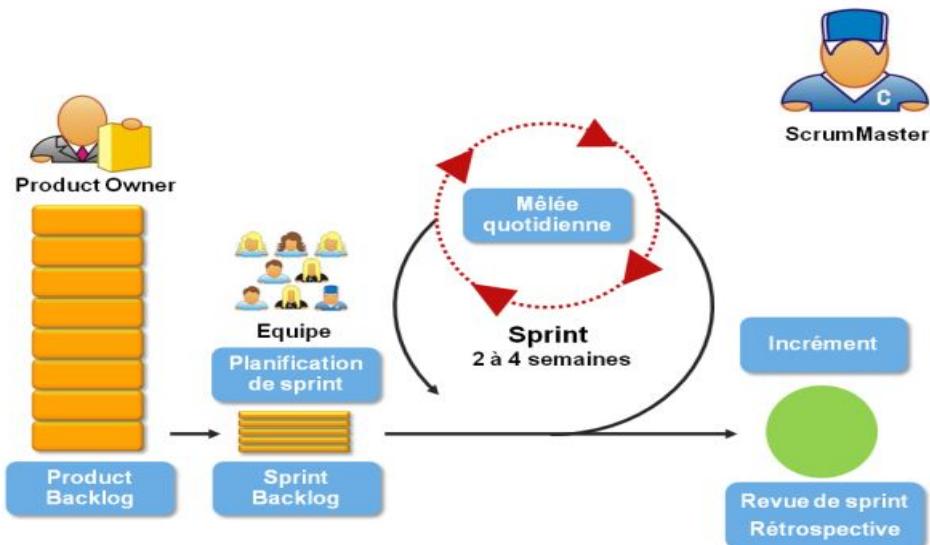


FIGURE 1.4 — Cycle de vie d'un projet Scrum

Pendant un sprint, des **réunions quotidiennes** sont organisées entre les collaborateurs du projet afin de visualiser l'avancement des tâches en cours, les problèmes rencontrés et les prochaines tâches à accomplir. Une **revue** aura lieu à la fin de chaque sprint afin de vérifier à la fois les fonctionnalités réalisées et celles qu'il faut améliorer.

Nous pouvons aussi remarquer d'après la Figure 1.3 l'intervention de trois rôles principaux :

- **Un Product Owner** qui définit le produit à réaliser.
- **Un Scrum Master** qui facilite les échanges et les améliorations.

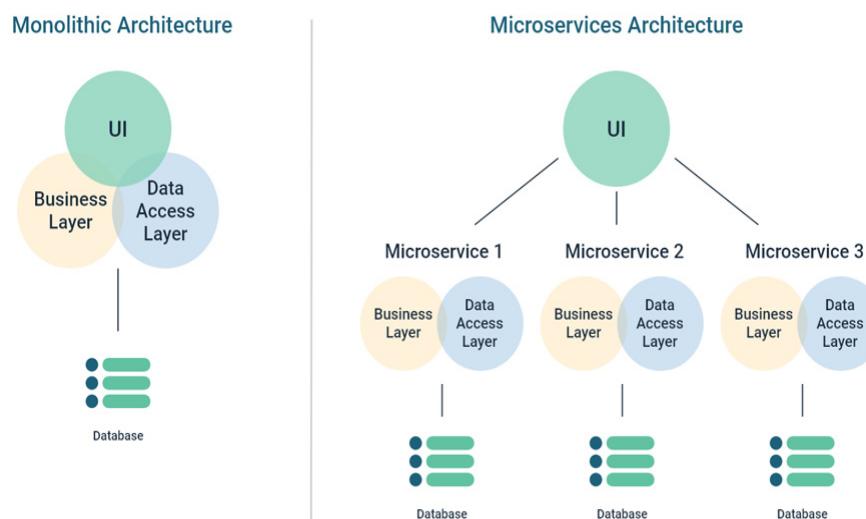
- Une équipe de développement qui possède toutes les compétences nécessaires pour développer le meilleur produit.

## 1.6 Langage de modélisation

Pour faciliter notre tâche nous avons choisi l'UML comme un langage de modélisation. UML [4] est un langage de modélisation graphique, il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.

## 1.7 Concepts théoriques des microservices

Les microservices [5] désignent une approche architecturale du développement d'applications. En fait, cette architecture permet de traiter une application comme un ensemble des petits services chaque service considéré comme un projet à part. Comme présente la figure 1.5, la différence entre architecture microservices et architecture monolithique.



**FIGURE 1.5** — Architecture monolithique et architecture microservices

Chaque service de cette architecture fonctionne moyennant son propre processus qui communique avec des mécanismes légers. Les services sont développés autour des compétences métiers qui sont déployés d'une façon indépendante par un processus automatisé. Ils sont autonomes et isolés mais aussi ils peuvent communiquer entre eux pour pourvoir créer l'ensemble du fonctionnalités nécessaires. Les microservices sont généralement gérés par des petites équipes avec suffisamment d'autonomie. Chaque

équipe peut modifier ou supprimer l'implémentation d'un service dans un microservice avec un impact minimal sur le reste du système. Ils ont plusieurs avantages et inconvénients tout dépend du cas dans lequel ils sont utilisés.

### 1.7.1 Avantages de l'architecture microservices

- **Mise sur le marché plus rapide :** Comme les cycles de développement sont plus courts, l'architecture de microservices permet des déploiements et mises à jour plus agiles.
- **Facilité de déploiement :** Les applications basées sur des microservices sont plus modulaires et légères que les applications monolithiques classiques. Aussi, vous pouvez les déployer plus sereinement.
- **Accessibilité :** Vu que l'application est décomposée en plusieurs éléments, les développeurs peuvent plus facilement comprendre, mettre à jour et améliorer chacun de ces éléments.
- **Résilience :** Lorsque les services sont développés correctement, ces services indépendants n'ont aucun impact les uns sur les autres. Cela signifie que, lorsqu'un élément tombe en panne, l'ensemble de l'application ne cesse pas de fonctionner comme c'est le cas avec le modèle monolithique.
- **Haute évolutivité :** À mesure que la demande pour certains services augmente, vous pouvez étendre les déploiements sur plusieurs serveurs et infrastructures pour répondre à vos besoins.

### 1.7.2 Inconvénients de l'architecture microservices

- **Longue durée de lancement d'application :** Pour démarrer les applications à base de microservices nous devons de lancer chaque service à part.
- **Complexité de faire les tests :** Les tests d'intégration,tout comme les tests complets, peuvent se complexifier et prendre une importance capitale.
- **Déploiement :** Le déploiement aussi un défi, du moins au début. Pour faciliter le déploiement, vous devez au préalable investir largement dans des solutions d'automatisation. En effet, les microservices deviennent vite trop complexes pour envisager un déploiement manuel.
- **Surveillance :** Il est essentiel de disposer d'une vision centralisée du système pour identifier les sources des problèmes.

## **Conclusion**

Dans ce chapitre, nous avons présenté l'organisme d'accueil. Ensuite, nous avons décrit la problématique ainsi qu'une étude de l'existant et nous avons proposé une solution alternative à réaliser. Puis, nous définissons la méthode de travail et le langage de modélisation adopté. Et finalement, nous avons introduit quelques concepts théoriques de l'architecture microservices.

Nous pouvons passer au chapitre suivant qui est réservé à l'analyse préliminaire.

Chapitre **2**

## Analyse préliminaire

### Plan

2.1	Spécification des besoins . . . . .	13
2.2	Pilotage de projet avec SCRUM . . . . .	16
2.3	Environnement de travail . . . . .	19
2.4	L'architecture de la solution . . . . .	24

## Introduction

La phase de spécification des besoins est nécessaire pour la conception de toute application pour fixer dès le début les besoins et la nature des utilisateurs cible et les scénarios d'utilisation du produit à développer.

Dans ce chapitre nous déterminons les fonctionnalités, les acteurs les plus pertinents, et identifions les cas d'utilisations initiaux en traduisant les besoins fonctionnels en diagramme de cas d'utilisations.

## 2.1 Spécification des besoins

### 2.1.1 Identification des acteurs

Les acteurs sont les entités externes qui interagissent avec le système. Notre projet comporte principalement trois acteurs :

- **Product Owner** : c'est un acteur a comme rôle la création et la gestion de détails du projet.
- **Scrum Master** : c'est un acteur qui sélectionne les membres de l'équipe appropriés pour exécuter le projet et surveiller l'avancement des tâches.
- **Membre** : c'est un acteur qui utilise le plus l'application, son rôle est la planification, la répartition et la gestion de déroulement des tâches du projet.

### 2.1.2 Les besoins fonctionnels

Les besoins fonctionnels répondent aux points précis du cahier des charges, et sont donc requis par notre utilisateur final et lui sont indispensables. En d'autres termes, ce sont les besoins obligatoires ou encore les fonctionnalités de l'application.

À cet égard, notre application doit répondre aux besoins fonctionnels suivants :

- **S'inscrire** : chaque utilisateur doit faire l'inscription pour qu'il puisse bénéficier des services de l'application.
- **S'authentifier** : pour assurer la sécurité de notre application chaque utilisateur doit s'authentifier avant d'accéder à son espace.
- **Récupérer le mot de passe oublié** : cette fonctionnalité permet à l'utilisateur de récupérer son mot de passe, s'il l'a oublié.
- **Gérer profil** :
  - \* Consulter le profil.
  - \* Modifier les informations du profil.
  - \* Changer mot de passe.
  - \* Changer photo de profil.
- **Gérer des projets** :
  - \* Créer un nouveau projet.

- \* Modifier un projet.
- \* Supprimer un projet.
- \* Fermer un projet.
- \* Consulter la liste des projets.
- \* Consulter les détails d'un projet .
- \* Attacher des documents à un projet.
- **Inviter un scrum master** : cette fonctionnalité permet le product owner d'inviter un scrum master au projet.
- **Inviter les membres d'équipe** : cette fonctionnalité permet le scrum master de choisir les membres d'équipe d'un projet.
- **Gérer des sprints :**
  - \* Créer un nouveau sprint.
  - \*Modifier un sprint.
  - \* Supprimer un sprint.
  - \* Consulter la liste des sprints.
  - \* Démarrer un sprint.
  - \* Arrêter un sprint.
  - \* Changer l'ordre d'un sprint.
  - \* Affecter une tâche à un sprint.
  - \* Retirer une tâche à partir d'un sprint.
- **Gérer des tâches :**
  - \* Créer une nouvelle tâche.
  - \* Modifier une tâche.
  - \* Supprimer une tâche.
  - \* Consulter les tâches (Poduct Backlog, Sprint Backlog).
  - \* Affecter une tâche à un membre d'équipe.
  - \* Consulter l'avancement des tâches (Scrum Board).
  - \* Commenter une tâche.
- **Discuter avec d'autres collaborateurs** : cette fonctionnalité offre un espace de discussion aux collaborateurs.
- **Notifier un utilisateur** : un système de notifications qui améliore la réactivité des collaborateurs.
- **Consulter les statistiques** : cette fonctionnalité représente un tableau de bord qui contient des chartes statistiques d'un projet.

### 2.1.3 Les besoins non fonctionnels

- **L'ergonomie** : l'application doit présenter des interfaces claires et faciles à utiliser.
- **La performance** : le temps de réponse et le chargement des données doivent être minimaux et la navigation d'une interface à une autre ne doit pas gêner l'utilisateur.
- **La fiabilité** : l'application doit avoir un fonctionnement fiable et assurant l'intégrité des données.

- **La robustesse** : en présence de données invalides, le système doit gérer les exceptions pour garantir un fonctionnement stable.

### 2.1.4 Diagramme de cas d'utilisation global

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision statique et globale du comportement fonctionnel d'un système logiciel. La figure 2.1 ci-dessous représente le diagramme global de notre application.

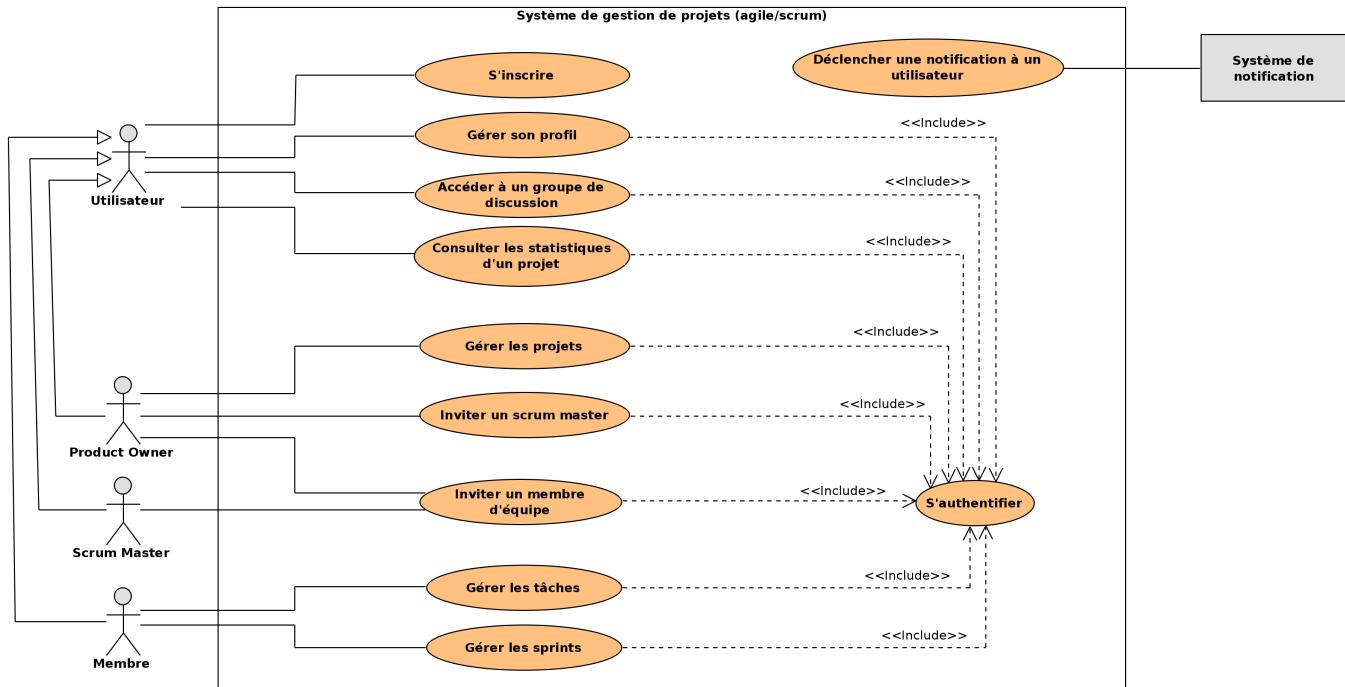


FIGURE 2.1 — Diagramme de cas d'utilisation global

## 2.2 Pilotage de projet avec SCRUM

### 2.2.1 Identification des rôles de SCRUM

Le tableau 2.1 ci-dessous illustre le rôle de chaque membre a été joué dans ce projet suivant la méthodologie SCRUM.

Rôle	Nom & Prénom
<b>Product Owner</b>	Ahmed Ayachi
<b>Scrum Master</b>	Driss Khalil
<b>Membre d'équipe</b>	Bouadila Achref, Mohamed oueslati

TABLE 2.1 — Les rôles de SCRUM

### 2.2.2 Le Product Backlog

Nous commençons par définir le Product Backlog qui est l'ensemble des caractéristiques fonctionnelles qui constituent le produit souhaité. Les caractéristiques fonctionnelles sont appelées des histoires utilisateurs (user story). Les user stories sont caractérisés par :

- **Identifiant** : Il détermine un identifiant unique pour l'histoire en question.
- **Description** : Elle décrit le besoin d'un acteur.
- **Priorité** : Les priorités sont utilisées pour définir l'ordre de réalisation, elles permettent de constituer le flux de stories qui va alimenter l'équipe.
- **Estimation** : Définir l'effort nécessaire pour réaliser la tâche.

La table 2.2 ci-dessous représente notre Product Backlog.

**NB** : Nous allons abrévier les acteurs Product Owner (PO) et Scrum Master (SM).

US-ID	User Story		Priorité	Estimation en (jours)
	En tant que	Je peux		
A.1	Utilisateur	S'inscrire	1	3
B.1	Utilisateur	S'authentifier	2	5
C.1	Utilisateur	Récupérer mon mot de passe	3	5
D.1	Utilisateur	Consulter mon profil	4	3
D.2	Utilisateur	Modifier les informations du profil	5	3
D.3	Utilisateur	Changer ma photo de profil	6	4
D.4	Utilisateur	Changer mon mot de passe	7	2
E.1	PO	Créer un nouveau projet	8	4
E.2	PO	Modifier un projet	9	4
E.3	PO	Supprimer un projet	10	1
E.4	PO	Fermer un projet	11	1
E.5	PO, SM, Membre	Consulter la liste des projets	12	5
E.6	PO, SM, Membre	Consulter les détails d'un projet	13	3
F.1	PO	Inviter un scrum master	14	5
G.1	PO, SM	Inviter les membres d'équipe	15	5
H.1	Membre	Créer un nouveau sprint	16	2
H.2	Membre	Modifier un sprint	17	2
H.3	Membre	Supprimer un sprint	18	1
H.4	PO, SM, Membre	Consulter la liste des sprints	19	3
H.5	Membre	Démarrer un sprint	20	1
H.6	Membre	Arrêter un sprint	21	1
H.7	Membre	Changer l'ordre d'un sprint	22	2
H.8	Membre	Affecter une tâche à un sprint	23	1
H.9	Membre	Retirer une tâche à partir d'un sprint	24	1
I.1	Membre	Créer une nouvelle tâche	25	2
I.2	Membre	Modifier une tâche	26	2
I.3	Membre	Supprimer une tâche	27	1

I.4	PO, SM, Membre	Consulter les tâches (Poduct Backlog, Sprint Backlog)	28	3
I.5	Membre	Affecter une tâche à un membre d'équipe	29	1
I.6	PO, SM, Membre	Consulter l'avancement des tâches (TaskBoard)	30	3
I.7	Membre	Commenter une tâche	31	2
J.1	Utilisateur	Accéder à un groupe de discussion	32	3
J.2	Utilisateur	Consulter la conversation du groupe	33	3
J.3	Utilisateur	Envoyer un nouveau message	34	2
J.4	Utilisateur	Supprimer un message	35	1
K.1	Système	Notifier l'utilisateur après avoir rejoint un nouveau projet.	36	1
K.2	Système	Notifier l'utilisateur lorsque le projet est mis à jour.	37	1
K.3	Système	Notifier l'utilisateur lorsque le projet est fermé.	38	1
K.4	Système	Notifier l'utilisateur de chaque nouveau sprint.	39	1
K.5	Système	Notifier l'utilisateur lorsque le sprint est mis à jour.	40	1
K.6	Système	Notifier l'utilisateur lorsque le sprint est supprimé.	41	1
K.7	Système	Notifier l'utilisateur de chaque nouvelle tâche.	42	1
K.8	Système	Notifier l'utilisateur lorsque la tâche est mise à jour.	43	1
K.9	Système	Notifier l'utilisateur lorsque la tâche est supprimée.	44	1
K.10	Système	Notifier l'utilisateur des nouveaux commentaires.	45	1
K.11	Système	Notifier l'utilisateur des nouveaux messages.	46	1
L.1	PO, SM, Membre	Consulter les statistiques	47	8

TABLE 2.2: Product Backlog

### 2.2.3 Planification des sprints

La planification des sprints est une cérémonie SCRUM. Elle a pour objectif de définir ce qui peut être livré dans chaque sprint et comment y parvenir. La planification des sprints est effectuée en collaboration avec toute l'équipe Scrum.

Le tableau 2.3 représente la planification des sprints que nous allons la suivre par la suite.

Release	Nom de sprint	Fonctionnalité	Durée de sprint
Release 1	Sprint 1	- A : Inscription - B : Authentification - C : Récupération de mot de passe - D : Gestion profil	4 semaines
	Sprint 2	- E : Gestion des projets - F : L'invitation du Scrum Master - G : L'invitation des membres	4 semaines
Release 2	Sprint 3	- H : Gestion des sprints - I : Gestion des tâches	4 semaines
	Sprint 4	- J : Espace de discussion - K : Système de notifications - L : Dashboard	4 semaines

TABLE 2.3 — Planification des sprints

## 2.3 Environnement de travail

Cette partie présente l'environnement matériel mis à la disposition du présent projet ainsi que l'environnement logiciel que nous avons exploité et nous présentons les choix technologiques.

### 2.3.1 Environnement matériel

Durant le stage, nous avons réalisé notre projet en utilisant un ordinateur portable dont la configuration est décrite ci-dessous :

- **Fabricant** : Lenovo.
- **Processeur** : Intel i5-8250U (8) @ 1.600GHz.
- **Mémoire installée (RAM)** : 12,00 Go.
- **Système d'exploitation** : Manjaro Linux 64 bits.

### 2.3.2 Environnement logiciel

- **Spring Tool Suite (STS)** : est un IDE étendu pour Eclipse. Il se spécialise dans le développement des applications Spring [6].



FIGURE 2.2 — Logo de Spring Tool Suite

- **Visual Studio Code (VsCode)** : est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git [7].



FIGURE 2.3 — Logo de Visual Studio Code

- **GitHub** : est une plateforme open source de gestion de versions et de collaboration destinée aux développeurs de logiciels [8].



FIGURE 2.4 — Logo de Github

- **Postman** : est une plateforme d'API pour la création et l'utilisation d'API. Simplifie chaque étape du cycle de vie des API et rationalise la collaboration afin que vous puissiez créer de meilleures API, plus rapidement [9].



FIGURE 2.5 — Logo de Postman

- **MySQL Workbench** : est un outil visuel unifié pour les architectes de bases de données, les développeurs et les administrateurs de bases de données. MySQL Workbench fournit la modélisation des données, le développement SQL et des outils d'administration complets pour la configuration du serveur [10].



**FIGURE 2.6** — Logo de MySQL Workbench

- **Visual Paradigm** : est un outil de conception et de gestion puissant, multiplateforme et pourtant facile à utiliser pour les systèmes informatiques [11].



**FIGURE 2.7** — Logo de Visual Paradigm

- **Overleaf** : est une plateforme en ligne gratuite permettant d'éditer du texte en LATEX sans aucun téléchargement d'application. En outre, elle offre la possibilité de rédiger des documents de manière collaborative [12].



**FIGURE 2.8** — Logo d'Overleaf

### 2.3.3 Technologies utilisées

- **Angular** : est un framework construit avec Typescript. Il a été construit par les développeurs de Google. Ce framework a été utilisé pour surmonter les obstacles rencontrés lors la création des applications clientes à seule page (Single Page Application) [13].



**FIGURE 2.9** — Logo d'Angular

- **Spring Boot** : est un projet basé sur le framework Spring qui fournit la fonctionnalité RAD ( Rapid Application Development ). Il est utilisé pour créer une application autonome basée sur Spring que vous pouvez simplement exécuter car elle nécessite une configuration Spring minimale [14].



**FIGURE 2.10** — Logo de Spring Boot

- **RabbitMQ** : est un logiciel de mise en file d'attente de messages, également appelé courtier de messages (Message Broker) ou gestionnaire de files d'attente. Simplement dit ; c'est un logiciel où sont définies des files d'attente, auxquelles les applications se connectent afin de transférer un ou des messages [15].



**FIGURE 2.11** — Logo de RabbitMQ

- **Websocket** : est une connection bi-directionnel et persistante entre un navigateur Web et un serveur. Une fois qu'une connexion WebSocket est établie, la connexion reste ouverte jusqu'à ce que le client ou le serveur décide de fermer cette connexion. Un cas d'utilisation typique pourrait être lorsqu'une application implique que plusieurs utilisateurs communiquent entre eux, comme dans une application de messagerie [16].



**FIGURE 2.12** — Logo de Websocket

- **MySQL** : est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle [17].



**FIGURE 2.13** — Logo de MySQL

- **Docker** : est un outil pour créer, déployer et gérer des conteneurs d'applications virtualisées sur un système d'exploitation commun.



**FIGURE 2.14** — Logo de Docker

## 2.4 L'architecture de la solution

### 2.4.1 L'architecture l'application en microservices

La figure 2.15 illustre les composants primordiaux pour mettre en place une architecture microservices.

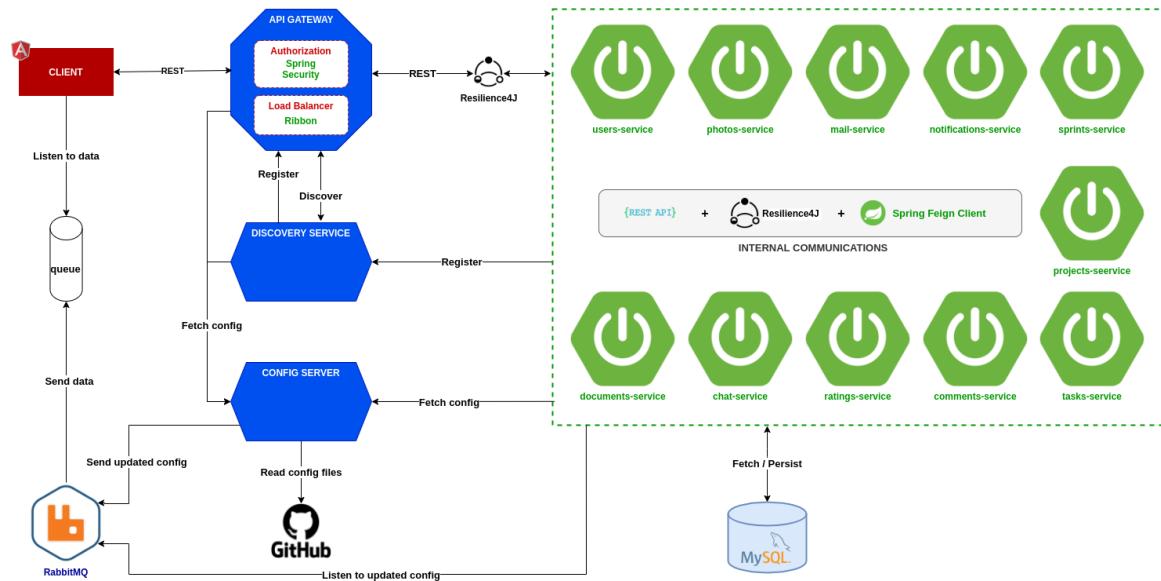


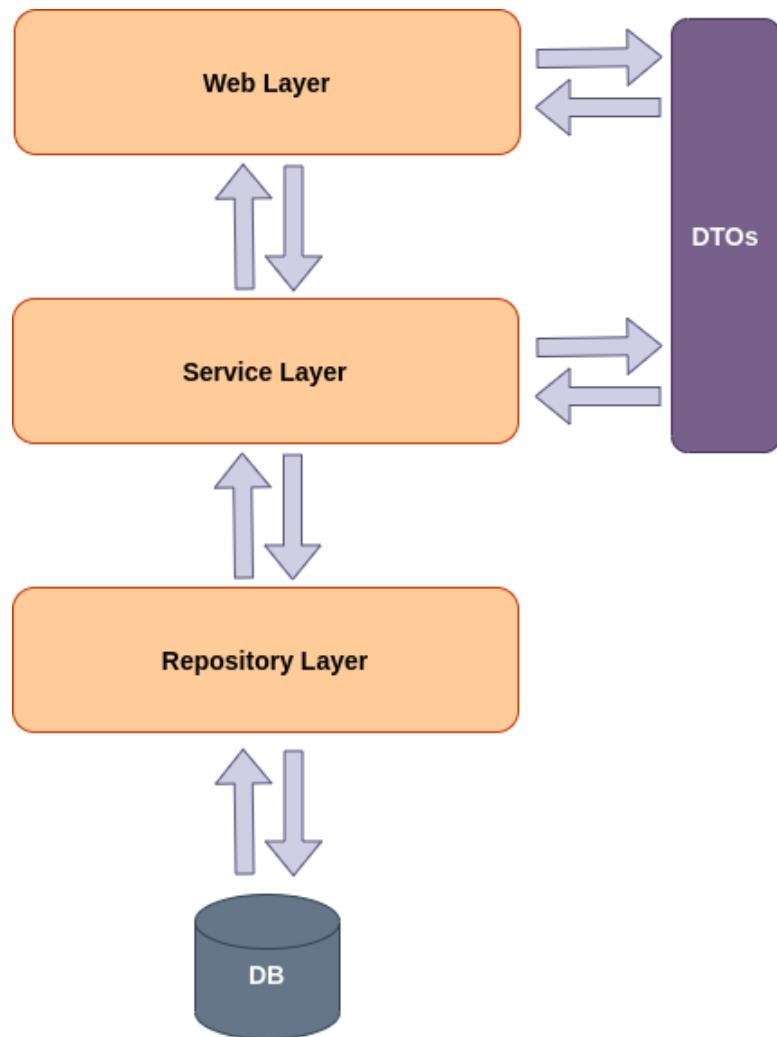
FIGURE 2.15 — L'architecture microservices

Cette architecture est composée de :

- **Serveur de configuration (Config server)** : dont le rôle est de centraliser les fichiers de configuration des différents microservices dans un endroit unique (un dépôt Git dans notre cas).
- **Service de découverte (Discovery Service)** : permettant l'enregistrement des instances de services en vue d'être découvertes par d'autres services.
- **API Passerelle (API Gateway)** : qui présente le point d'entrée de la partie back-end de notre application. Elle encapsule l'architecture interne et fournit des API adaptées pour chaque type de client. l'API Gateway encapsule un composant très important qui est l'équilibrage de charge.
- **Équilibrage de charge (Load Balancer)** : c'est le responsable de router et répartir la charge entre les instances des microservices disponibles. Pour avoir la liste de ces dernières, il consulte le serveur de découverte.
- **Disjoncteur de circuit (Circuit Breaker)** : ce composant est essentiel d'éviter l'échec de l'application au cas où un microservice a cessé de fonctionner.

### 2.4.2 L'architecture n-tiers d'un microservice

La figure 2.16 représente l'architecture d'un seul microservice qui est constituée de trois couches.



**FIGURE 2.16** — L'architecture n-tiers d'un microservice

- **La couche Web** : est la couche supérieure d'une application web. Il est chargé de traiter les entrées de l'utilisateur et de renvoyer la réponse correcte à l'utilisateur. La couche Web doit également gérer les exceptions levées par les autres couches.
- **La couche Service** : réside sous la couche web et contient toute la logique métier de notre application.
- **La couche Repository** : est la couche la plus basse d'une application web. Il est chargé de communiquer avec la base de données utilisée.

Dans cette architecture nous avons utilisé, Les objets de transferts de données (DTOs). Ces objets sont utilisés pour transporter des données entre les différents processus et entre les couches de notre application. Un tel objet aidera à ne pas casser les interfaces à chaque changement de modèle.

## Conclusion

Tout au long de ce chapitre nous avons présenté les besoins fonctionnels et non fonctionnels du projet ainsi que l'environnement de travail et l'architecture de l'application pour donner une vision clairvoyante pour la suite. Dans les prochains chapitres, nous allons présenter les sprints et le travail réalisé dans chacun.

# Chapitre 3

## Sprint 1 : Authentication, Authorisation, Recupera

### Plan

3.1	Analyse du sprint . . . . .	28
3.2	Conception du sprint . . . . .	32
3.3	Revue du sprint . . . . .	36

## Introduction

Dans ce chapitre nous allons implémenter l'authentification, l'autorisation, la récupération de mot passe et la gestion du profil. Ce chapitre explique le cycle de vie du premier sprint, à savoir le Sprint Backlog, la conception et la réalisation.

### 3.1 Analyse du sprint

La phase d'analyse rende les besoins fonctionnels de ce sprint mieux compréhensible et plus clair. Dans ce contexte nous allons entamer à modéliser le diagramme de cas d'utilisation détaillé ainsi que le « Sprint Backlog ».

#### 3.1.1 Diagrammes de cas d'utilisation

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

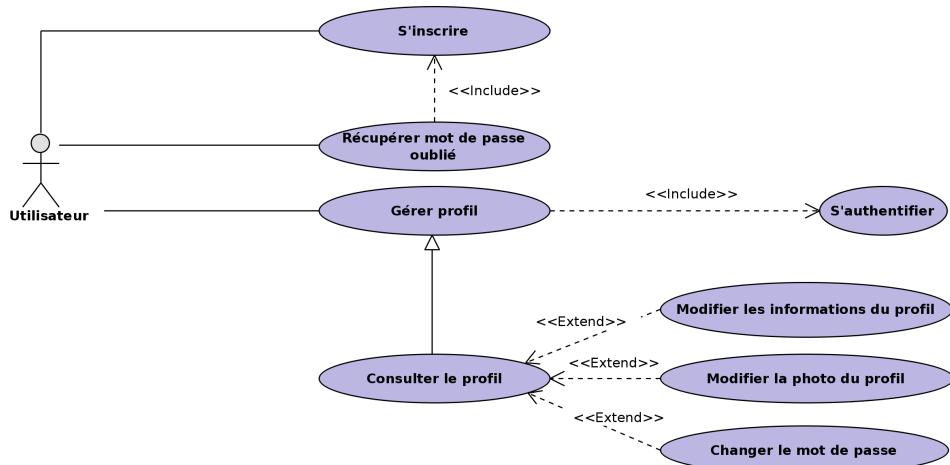


FIGURE 3.1 — Le diagramme de cas d'utilisation global de sprint 1

## CHAPITRE 3. SPRINT 1 : AUTHENTICATION, AUTHORISATION, RECUPERA

---

### 3.1.1.1 Description textuelle de cas d'utilisation « S'inscrire »

<b>Nom du cas d'utilisation</b>	S'inscrire
<b>Acteurs</b>	Utilisateur
<b>Description</b>	Ce cas permet à l'utilisateur de s'inscrire.
<b>Pré-condition</b>	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur clique sur le bouton s'inscrire.</li> <li>2. Le système affiche le formulaire d'inscription.</li> <li>3. L'acteur remplir le formulaire.</li> <li>4. L'acteur confirme l'action.</li> <li>5. Le système vérifie les champs du formulaire.</li> <li>6. Le système créer un nouveau compte.</li> <li>7. Le système redirige l'acteur à l'interface d'authentification.</li> </ol>
<b>Scénario alternative</b>	<p>3.1 : Si les champs sont erronés ou vides.  - Le système les indique dans le formulaire.</p> <p>6.1 : Si le champs email existe déjà.  - Le système affiche un message d'erreur.</p>
<b>Post-condition</b>	L'acteur sera inscrit.

TABLE 3.1 — Description textuelle de cas d'utilisation « S'inscrire »

### 3.1.1.2 Description textuelle de cas d'utilisation « S'authentifier »

<b>Nom du cas d'utilisation</b>	S'authentifier
<b>Acteurs</b>	Utilisateur
<b>Description</b>	Ce cas permet à l'utilisateur de s'authentifier.
<b>Pré-condition</b>	L'acteur doit être inscrit
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur clique sur le bouton s'authentifier.</li> <li>2. Le système affiche le formulaire d'authentification.</li> <li>3. L'acteur saisit son login et mot de passe.</li> <li>4. L'acteur confirme l'action.</li> <li>5. Le système vérifie les champs du formulaire.</li> <li>6. Le système redirige l'acteur vers son espace.</li> </ol>
<b>Scénario alternative</b>	<p>3.1 : Si les champs sont erronés ou vides.  - Le système les indique dans le formulaire.</p> <p>6.1 : Si le login ou le mot de passe est invalide.  - Le système affiche un message d'erreur.</p>
<b>Post-condition</b>	L'acteur sera authentifié.

TABLE 3.2 — Description textuelle de cas d'utilisation « S'authentifier »

**3.1.1.3 Description textuelle de cas d'utilisation « Récupérer mot de passe oublié »**

<b>Nom du cas d'utilisation</b>	Récupérer mot de passe oublié
<b>Acteurs</b>	Utilisateur
<b>Description</b>	Ce cas permet à l'utilisateur de récupérer son mot de passe oublié.
<b>Pré-condition</b>	L'acteur doit être inscrit
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur clique sur le bouton mot de passe oublié.</li> <li>2. Le système affiche le formulaire de demande de récupération.</li> <li>3. L'acteur saisit son email.</li> <li>4. L'acteur confirme l'action.</li> <li>5. Le système vérifie les champs du formulaire.</li> <li>6. Le système envoi un email contenant l'url de récupération de mot de passe.</li> <li>7. L'acteur accède à l'url envoyé.</li> <li>8. Le système affiche le formulaire de récupération.</li> <li>9. L'acteur saisit son nouveau mot de passe.</li> <li>10. L'acteur confirme l'action.</li> <li>11. Le système met à jour le mot de passe et renvoi l'acteur à l'interface d'authentification.</li> </ol>
<b>Scénario alternative</b>	<ul style="list-style-type: none"> <li>3.1 : Si les champs sont erronés ou vides. - Le système les indique dans le formulaire.</li> <li>6.1 : Si l'acteur n'est pas encore inscrit. - Le système affiche un message d'erreur.</li> <li>9.1 : Si les champs sont erronés ou vides. - Le système les indique dans le formulaire.</li> </ul>
<b>Post-condition</b>	L'acteur va avoir un nouveau mot de passe.

**TABLE 3.3** — Description textuelle de cas d'utilisation « Récupérer mot de passe oublié »

**3.1.1.4 Description textuelle de cas d'utilisation « Modifier les informations du profil »**

<b>Nom du cas d'utilisation</b>	Modifier les informations du profil
<b>Acteurs</b>	Utilisateur
<b>Description</b>	Ce cas permet à l'utilisateur de modifier ses informations du profil.
<b>Pré-condition</b>	L'acteur doit être authentifié
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur accède à son profil.</li> <li>2. Le système affiche un formulaire contenant les informations du profil.</li> <li>3. L'acteur modifie les informations qu'il souhaite.</li> <li>4. L'acteur confirme l'action.</li> <li>5. Le système vérifie les champs.</li> <li>6. Le système met à jour les informations du profil.</li> </ol>
<b>Scénario alternative</b>	5.1 : Si les champs sont erronés ou vides. - Le système les indique dans le formulaire.
<b>Post-condition</b>	Les informations de l'acteur seront modifiées.

TABLE 3.4 — Description textuelle de cas d'utilisation « Modifier les informations du profil »

### 3.1.2 Sprint Backlog

Le tableau 3.5 illustre le Backlog du premier sprint.

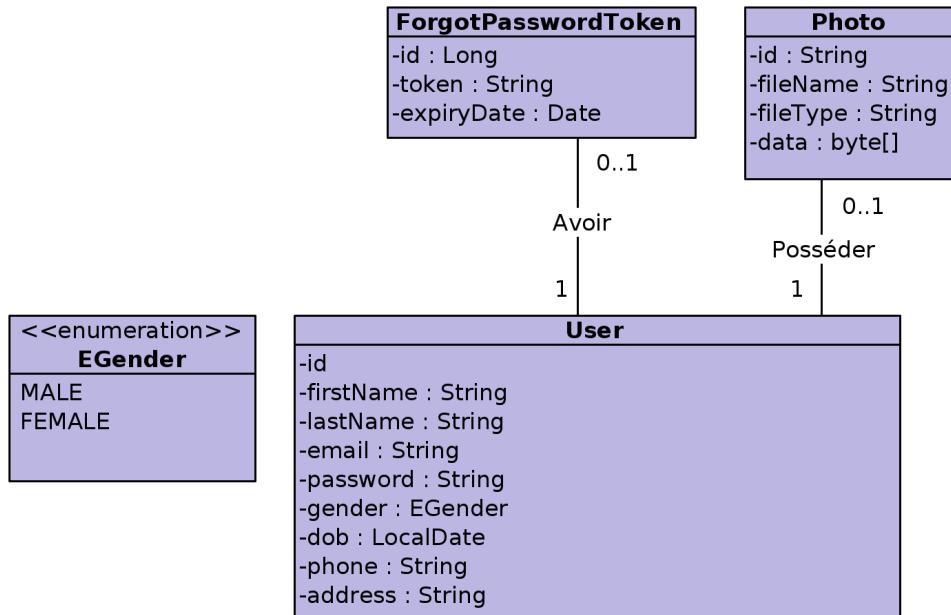
US-ID	User Story		<b>Priorité</b>	<b>Estimation en (jours)</b>
	En tant que	Je peux		
A.1	Utilisateur	S'inscrire	1	3
B.1	Utilisateur	S'authentifier	2	5
C.1	Utilisateur	Récupérer mon mot de passe	3	5
D.1	Utilisateur	Consulter mon profil	4	3
D.2	Utilisateur	Modifier les informations du profil	5	3
D.3	Utilisateur	Changer ma photo de profil	6	4
D.4	Utilisateur	Changer mon mot de passe	7	2

TABLE 3.5: Backlog du sprint 1

## 3.2 Conception du sprint

### 3.2.1 Le diagramme de classes

La figure 3.2 montre le diagramme de classes du premier sprint.



**FIGURE 3.2** — Le diagramme de classes du sprint 1

- User : Cette classe contient les informations des utilisateurs.
- ForgotPasswordToken : Cette classe sert à la récupération de mot de passe.
- Photo : Cette classe contient les photos des utilisateurs.
- EGender : Cette énumération liste les différents genres.

### 3.2.2 les diagrammes de séquences

### 3.2.2.1 Diagramme de séquence « S'inscrire »

La figure 3.3 illustre le diagramme de séquence qui permet à l'utilisateur de s'inscrire

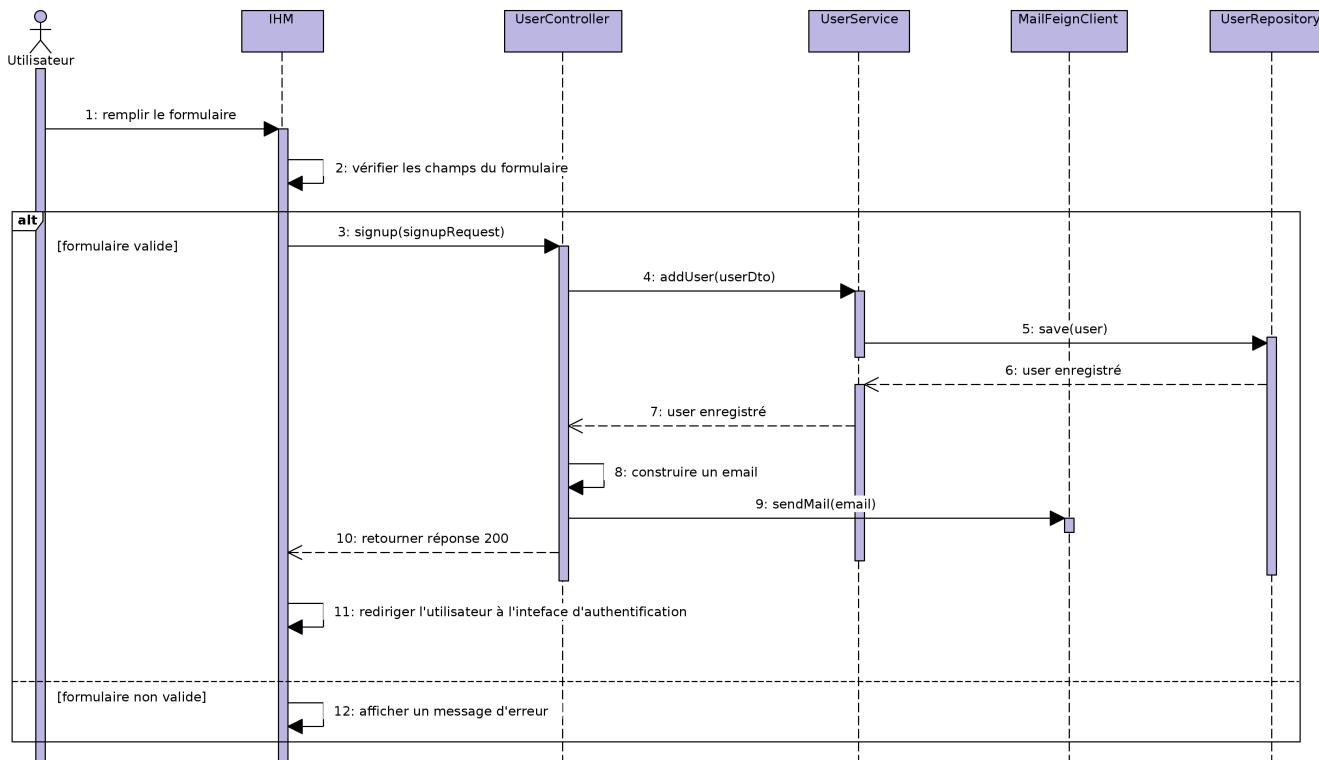
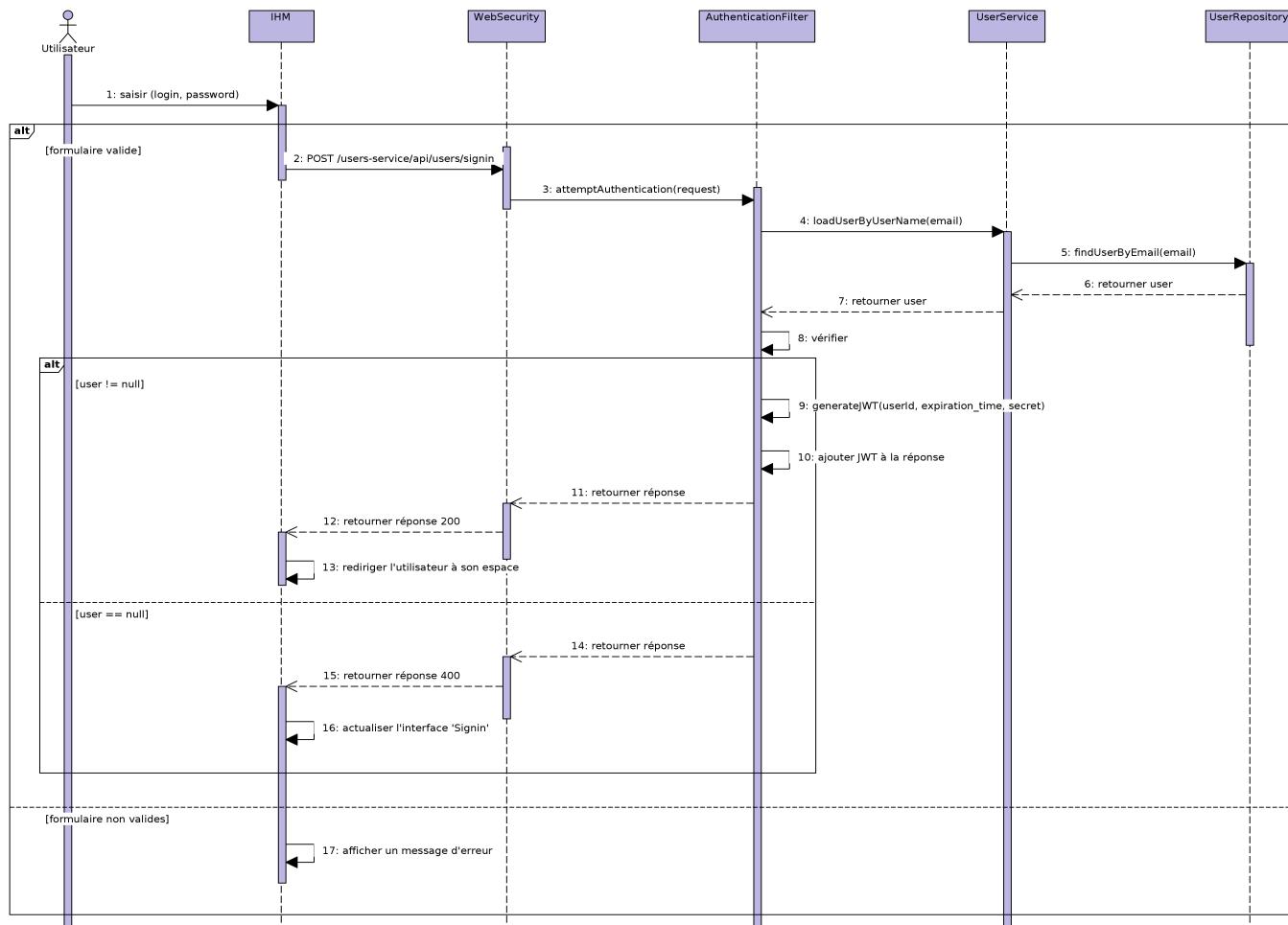


FIGURE 3.3 — Le diagramme de séquence « S'inscrire »

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

La figure 3.4 représente le diagramme de séquence qui permet à l'utilisateur de s'authentifier



**FIGURE 3.4** — Le diagramme de séquence « S'authentifier »

### 3.2.2.3 Diagramme de séquence « Autorisation »

La figure 3.5 montre le diagramme de séquence d'autorisation qui vérifie à chaque requête si l'utilisateur est authentifié ou pas.

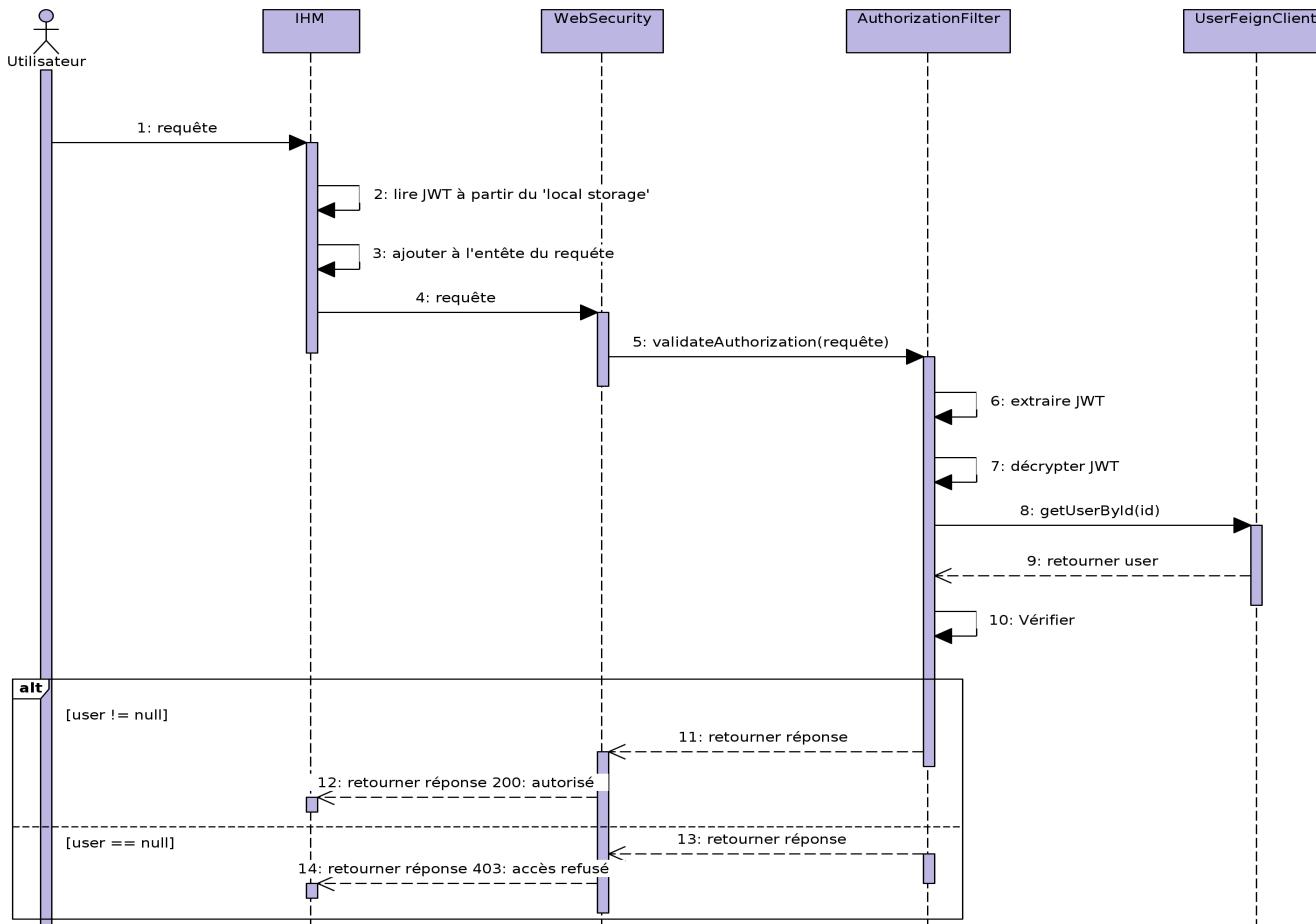


FIGURE 3.5 — Le diagramme de séquence « Autorisation »

### 3.3 Revue du sprint

Pour clôturer, dans cette section nous présentons les interfaces de la revue du premier sprint.

#### 3.3.1 L'interface d'inscription

La figure 3.6 montre l'interface pour la création d'un compte. L'utilisateur doit remplir le nom, prénom, mail et mot de passe.



FIGURE 3.6 — L'interface d'inscription

#### 3.3.2 L'interface d'authentification

L'interface d'authentification est le point d'entrée de notre application comme le montre la figure 3.7.



FIGURE 3.7 — L'interface d'authentification

### 3.3.3 Les interfaces de récupération de mot de passe

Les figures 3.8, 3.9, 3.10 montrent le processus à suivre pour récupérer le mot de passe oublié.



FIGURE 3.8 — L'interface initiale de récupération de mot de passe

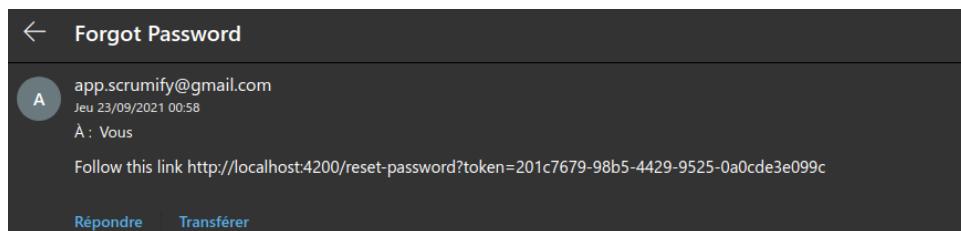


FIGURE 3.9 — L'email de récupération de mot de passe



FIGURE 3.10 — L'interface changement de mot de passe

### 3.3.4 L'interface de gestion du profil

La figure 3.11 illustre l'interface de gestion du profil utilisateur. L'interface contient les informations générales de l'utilisateur, ainsi que la photo du profil.

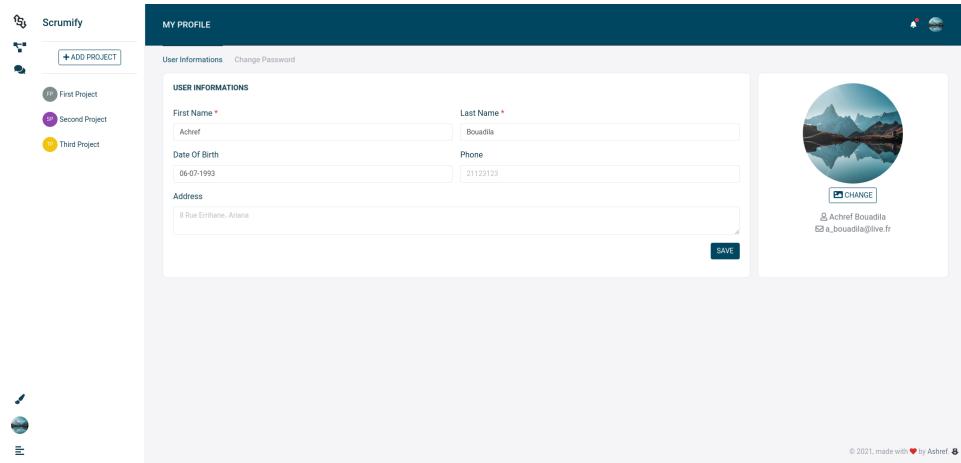


FIGURE 3.11 — L'inferface de gestion du profil

## Conclusion

Dans ce chapitre nous avons présenté les besoins fonctionnels du premier sprint à travers un diagramme de cas d'utilisation, ainsi que le Sprint Backlog, aussi nous avons présenté la conception en utilisant les diagrammes de séquences objet et le diagramme de classes et nous avons clôturé notre chapitre par la présentation des interfaces de ce sprint.

Chapitre **4**

## Sprint 2 : Gestion des projets et L'invitation des membres

### Plan

4.1	Analyse du sprint . . . . .	40
4.2	Conception du sprint . . . . .	42
4.3	Revue du sprint . . . . .	45

## Introduction

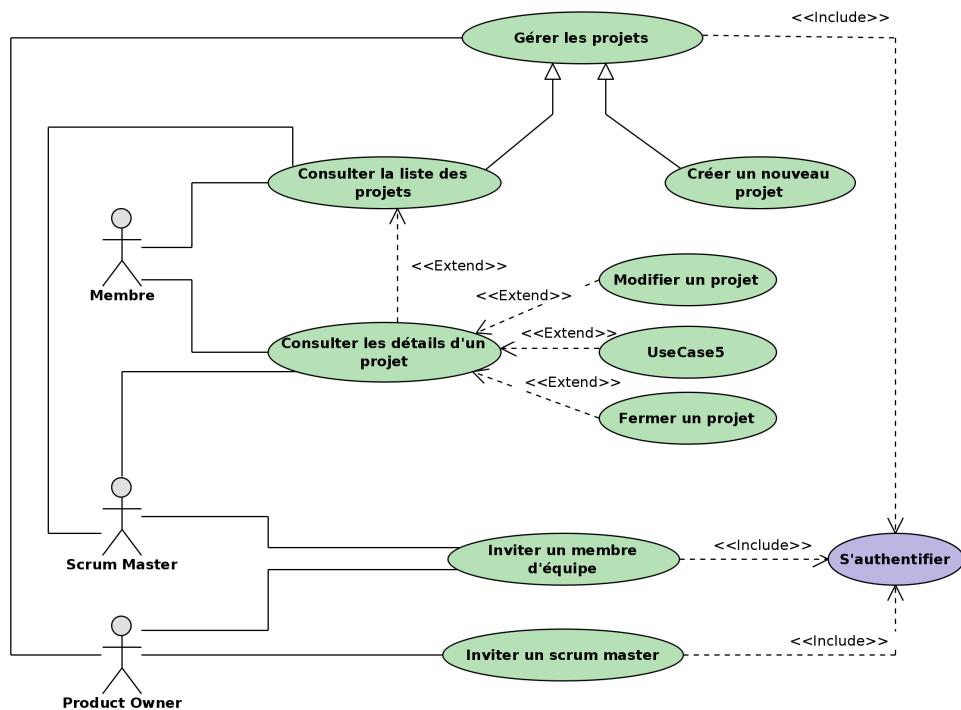
Dans ce chapitre nous allons implémenter la gestion des projets et l'invitation des membres du projet. Ce chapitre explique le cycle de vie du deuxième sprint, à savoir le Sprint Backlog, la conception et la réalisation.

### 4.1 Analyse du sprint

La phase d'analyse rende les besoins fonctionnels de ce sprint mieux compréhensible et plus clair. Dans ce contexte nous allons entamer à modéliser le diagramme de cas d'utilisation détaillé ainsi que le « Sprint Backlog ».

#### 4.1.1 Diagrammes de cas d'utilisation

La figure 4.1 représente le diagramme de cas d'utilisation global du deuxième sprint.



**FIGURE 4.1** — Le diagramme de cas d'utilisation global de sprint 2

## CHAPITRE 4. SPRINT 2 : GESTION DES PROJETS ET L'INVITATION DES MEMBRES

---

### 4.1.1.1 Description textuelle de cas d'utilisation « Créer un nouveau projet »

<b>Nom du cas d'utilisation</b>	Créer un nouveau projet
<b>Acteurs</b>	Product Owner
<b>Description</b>	Ce cas permet au Product Owner de créer un projet.
<b>Pré-condition</b>	L'acteur doit être authentifié
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur clique sur le bouton ajouter.</li> <li>2. Le système affiche le formulaire d'ajout d'un projet.</li> <li>3. L'acteur remplit le formulaire.</li> <li>4. L'acteur confirme l'action.</li> <li>5. Le système vérifie les champs du formulaire.</li> <li>6. Le système ajoute le projet à la base de données.</li> </ol>
<b>Scénario alternative</b>	<p>3.1 : Si les champs sont erronés ou vides. - Le système les indique dans le formulaire.</p>
<b>Post-condition</b>	un nouveau projet sera créé.

TABLE 4.1 — Description textuelle de cas d'utilisation « Créer un nouveau projet »

### 4.1.1.2 Description textuelle de cas d'utilisation « Inviter un Scrum Master »

<b>Nom du cas d'utilisation</b>	Inviter un Scrum Master
<b>Acteurs</b>	Product Owner
<b>Description</b>	Ce cas permet le Product Owner d'associer un Scrum Master au projet.
<b>Pré-condition</b>	L'acteur doit être authentifié
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur clique sur le bouton inviter Scrum Master.</li> <li>2. Le système affiche le formulaire.</li> <li>3. L'acteur sélectionne le Scrum Master qu'il souhaite.</li> <li>4. L'acteur confirme l'action.</li> <li>5. Le système vérifie les champs du formulaire.</li> <li>6. Le système associe le Scrum Master au projet.</li> </ol>
<b>Scénario alternative</b>	<p>3.1 : Si les champs sont erronés ou vides. - Le système les indique dans le formulaire.</p>
<b>Post-condition</b>	un Scrum Master sera associé au projet.

TABLE 4.2 — Description textuelle de cas d'utilisation « Inviter un Scrum Master »

### 4.1.2 Sprint Backlog

Le tableau 4.3 illustre le Sprint Backlog du deuxième sprint.

US-ID	User Story		Priorité	Estimation en (jours)
	En tant que	Je peux		
E.1	PO	Créer un nouveau projet	8	4
E.2	PO	Modifier un projet	9	4
E.3	PO	Supprimer un projet	10	1
E.4	PO	Fermer un projet	11	1
E.5	PO, SM, Membre	Consulter la liste des projets	12	5
E.6	PO, SM, Membre	Consulter les détails d'un projet	13	3
F.1	PO	Inviter un scrum master	14	5
G.1	PO, SM	Inviter les membres d'équipe	15	5

TABLE 4.3: Backlog du sprint 2

## 4.2 Conception du sprint

### 4.2.1 Le diagramme de classes

La figure 4.2 montre le diagramme de classes du deuxième sprint.

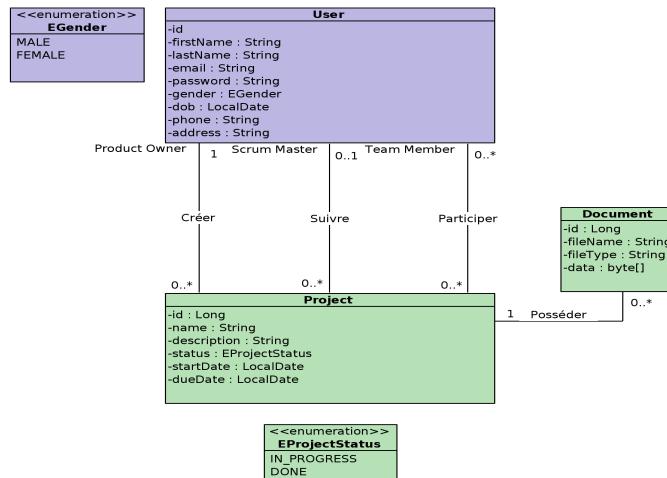


FIGURE 4.2 — Le diagramme de classes du sprint 2

## CHAPITRE 4. SPRINT 2 : GESTION DES PROJETS ET L'INVITATION DES MEMBRES

---

- **Project** : Cette classe contient la liste des projets.
- **Document** : Cette classe comprend les informations des documents.
- **EProjectStatus** : Cette énumération représente les différents états du projet.

### 4.2.2 les diagrammes de séquences

#### 4.2.2.1 Diagramme de séquence « Crée un nouveau projet »

La figure 4.3 illustre le diagramme de séquence qui permet au Product Owner de créer un nouveau projet

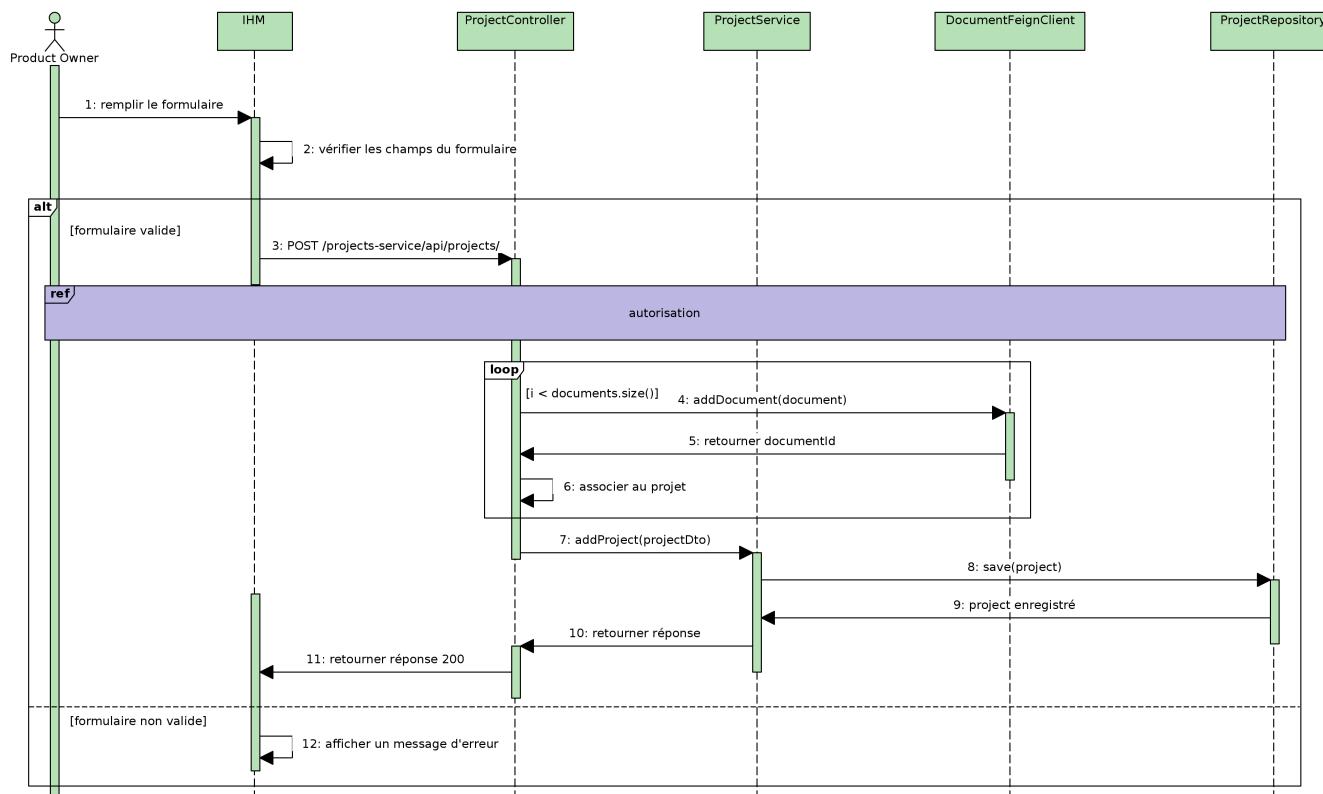


FIGURE 4.3 — Le diagramme de séquence « Crée un nouveau projet »

### 4.3 Revue du sprint

Pour clôturer, dans cette section nous présentons les interfaces de la revue du deuxième sprint.

#### 4.3.1 L'interface de création d'un projet

La figure 4.4 montre l'interface de création du projet. Pour le créer, il faut renseigner le nom du projet, description, date de fin et les documents associés à ce projet.

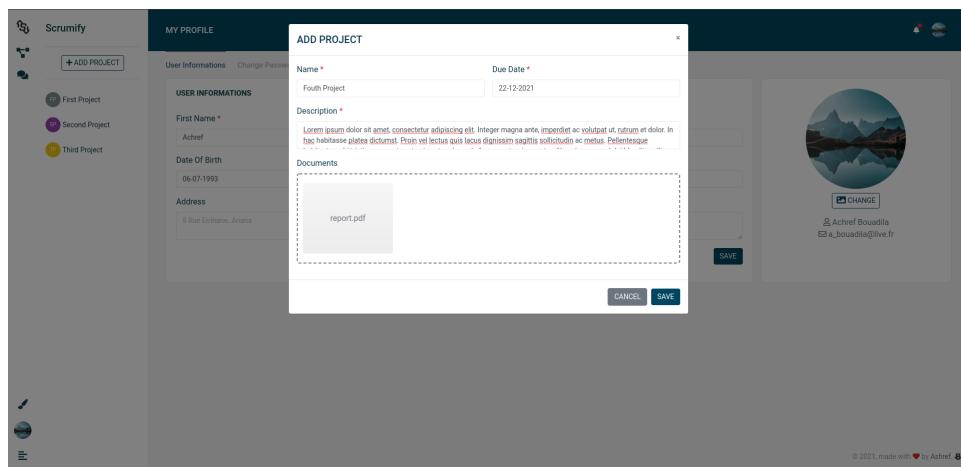


FIGURE 4.4 — L'interface de création d'un projet

#### 4.3.2 L'interface pour consulter les détails d'un projet

Une fois le projet créé, l'utilisateur peut accéder à cette interface pour voir les membres impliqués dans le projet, le statut et les dates de début et fin du projet.

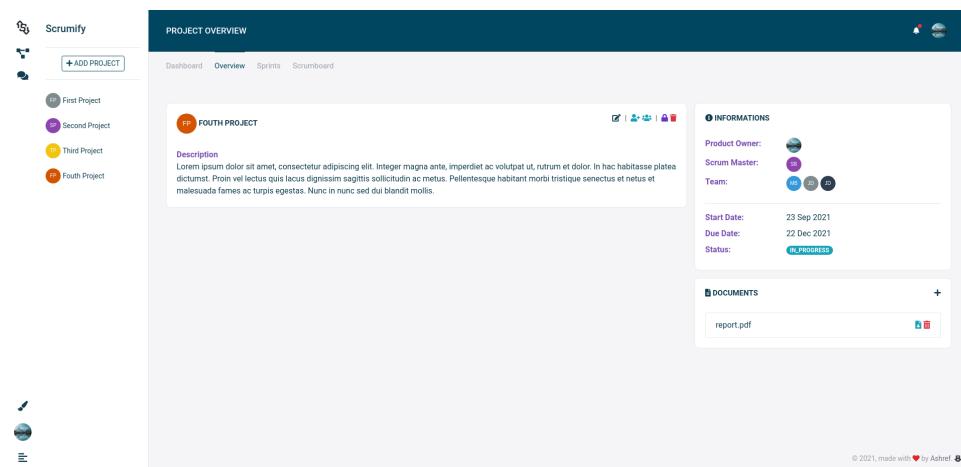


FIGURE 4.5 — L'interface pour consulter les détails d'un projet

## CHAPITRE 4. SPRINT 2 : GESTION DES PROJETS ET L'INVITATION DES MEMBRES

### 4.3.3 L'interface pour inviter un Scrum Master

La figure 4.6 représente l'interface utilisée pour associer un Scrum Master au projet.

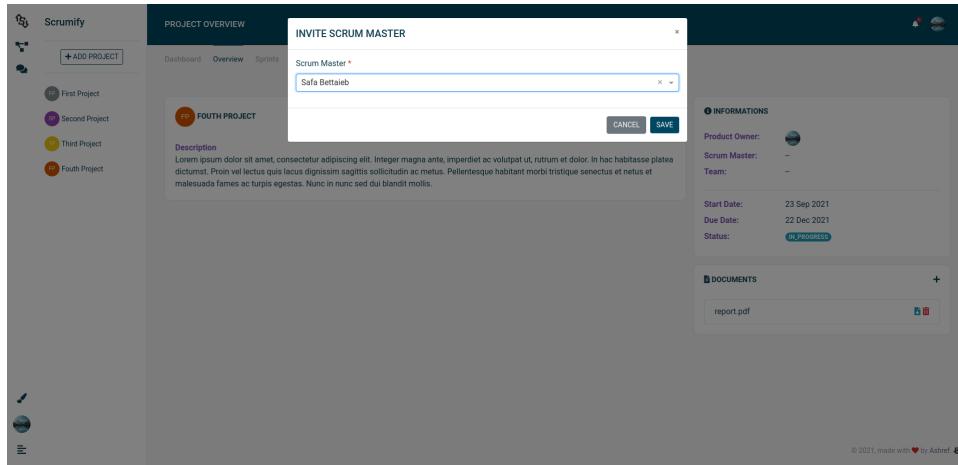


FIGURE 4.6 — L'interface pour inviter un Scrum Master

### 4.3.4 L'interface pour inviter les membres d'équipe

La figure 4.7 illustre l'interface utilisée pour ajouter les membres d'équipe au projet.

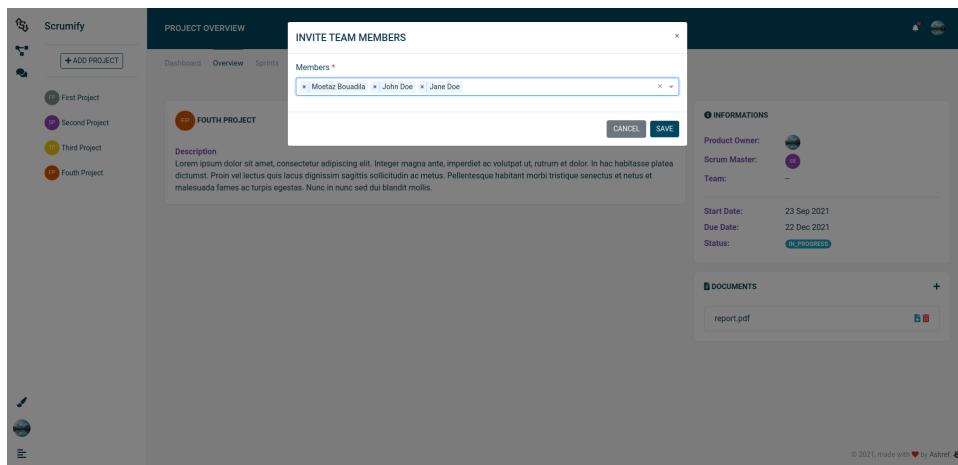


FIGURE 4.7 — L'interface pour inviter les membres d'équipe

## CHAPITRE 4. SPRINT 2 : GESTION DES PROJETS ET L'INVITATION DES MEMBRES

### 4.3.5 L'interface pour consulter la liste des projets

La figure 4.8 montre l'interface qui permet l'utilisateur de consulter la liste des projets créés.

The screenshot shows the 'PROJECTS' page of the Scrumify application. On the left sidebar, there are four project cards: 'First Project' (In Progress), 'Second Project' (In Progress), 'Third Project' (Done), and 'Fourth Project' (In Progress). The main area displays four project cards:

- FIRST PROJECT**: Status: IN\_PROGRESS. Start Date: 30 Jul 2021, Due Date: 22 Aug 2021. Product Owner: [Avatar], Scrum Master: [Avatar], Team: [Avatar]. Progress: 50%.
- SECOND PROJECT**: Status: IN\_PROGRESS. Start Date: 30 Jul 2021, Due Date: 07 Sep 2021. Product Owner: [Avatar], Scrum Master: [Avatar], Team: [Avatar]. Progress: 0%.
- THIRD PROJECT**: Status: DONE. Start Date: 19 Aug 2021, Due Date: 28 Oct 2021. Product Owner: [Avatar], Scrum Master: [Avatar], Team: [Avatar]. Progress: 0%.
- FOURTH PROJECT**: Status: IN\_PROGRESS. Start Date: 30 Jul 2021, Due Date: 22 Aug 2021. Product Owner: [Avatar], Scrum Master: [Avatar], Team: [Avatar]. Progress: 0%.

FIGURE 4.8 — L'interface pour consulter la liste des projets

## Conclusion

Dans ce chapitre nous avons présenté les besoins fonctionnels du deuxième sprint à travers un diagramme de cas d'utilisation, ainsi que le Sprint Backlog, aussi nous avons présenté la conception en utilisant les diagrammes de séquences objet et le diagramme de classes et nous avons clôturé notre chapitre par la présentation des interfaces de ce sprint.

Chapitre **5**

# Sprint 3 : Gestion des sprints et gestion des tâches

## Plan

5.1	Analyse du sprint . . . . .	49
5.2	Conception du sprint . . . . .	51
5.3	Revue du sprint . . . . .	55

## Introduction

Dans ce chapitre nous allons implémenter la gestion des sprints et la gestion des tâches. Ce chapitre explique le cycle de vie du troisième sprint, à savoir le Sprint Backlog, la conception et la réalisation.

### 5.1 Analyse du sprint

La phase d'analyse rende les besoins fonctionnels de ce sprint mieux compréhensible et plus clair. Dans ce contexte nous allons entamer à modéliser le diagramme de cas d'utilisation détaillé ainsi que le « Sprint Backlog ».

#### 5.1.1 Diagrammes de cas d'utilisation

La figure 5.1 représente le diagramme de cas d'utilisation global du troisième sprint.

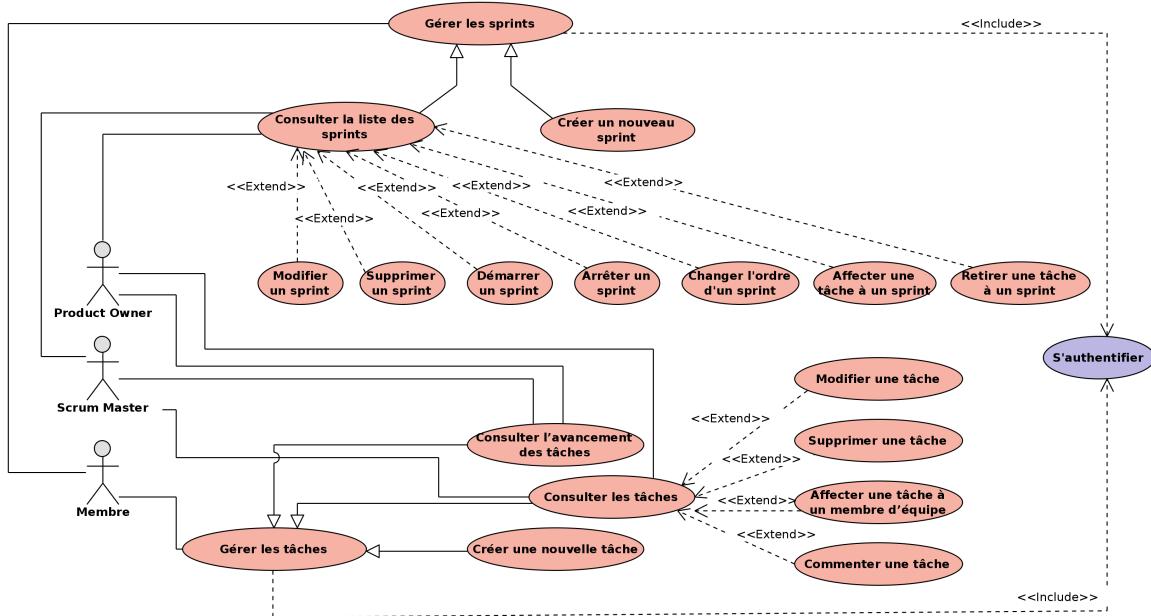


FIGURE 5.1 — Le diagramme de cas d'utilisation global de sprint 3

## CHAPITRE 5. SPRINT 3 : GESTION DES SPRINTS ET GESTION DES TÂCHES

---

### 5.1.1.1 Description textuelle de cas d'utilisation « Créer un nouveau sprint »

<b>Nom du cas d'utilisation</b>	Créer un nouveau sprint
<b>Acteurs</b>	Membre
<b>Description</b>	Ce cas permet au membre de créer un nouveau sprint.
<b>Pré-condition</b>	L'acteur doit être authentifié
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur clique sur le bouton ajouter.</li> <li>2. Le système affiche le formulaire d'ajout d'un projet.</li> <li>3. L'acteur remplir le formulaire.</li> <li>4. L'acteur confirme l'action.</li> <li>5. Le système vérifie les champs du formulaire.</li> <li>6. Le système ajouter le projet à la base de données.</li> </ol>
<b>Scénario alternative</b>	<p>3.1 : Si les champs sont erronés ou vides. - Le système les indique dans le formulaire.</p>
<b>Post-condition</b>	un nouveau projet sera créé.

TABLE 5.1 — Description textuelle de cas d'utilisation « Créer un nouveau sprint »

### 5.1.2 Sprint Backlog

Le tableau 5.2 illustre le Sprint Backlog du troisième sprint.

US-ID	User Story		Priorité	Estimation en (jours)
	En tant que	Je peux		
H.1	Membre	Créer un nouveau sprint	16	2
H.2	Membre	Modifier un sprint	17	2
H.3	Membre	Supprimer un sprint	18	1
H.4	PO, SM, Membre	Consulter la liste des sprints	19	3
H.5	Membre	Démarrer un sprint	20	1
H.6	Membre	Arrêter un sprint	21	1
H.7	Membre	Changer l'ordre d'un sprint	22	2

## CHAPITRE 5. SPRINT 3 : GESTION DES SPRINTS ET GESTION DES TÂCHES

---

H.8	Membre	Affecter une tâche à un sprint	23	1
H.9	Membre	Retirer une tâche à partir d'un sprint	24	1
I.1	Membre	Créer une nouvelle tâche	25	2
I.2	Membre	Modifier une tâche	26	2
I.3	Membre	Supprimer une tâche	27	1
I.4	PO, SM, Membre	Consulter les tâches (Poduct Backlog, Sprint Backlog)	28	3
I.5	Membre	Affecter une tâche à un membre d'équipe	29	1
I.6	PO, SM, Membre	Consulter l'avancement des tâches (TaskBoard)	30	3
I.7	Membre	Commenter une tâche	31	2

TABLE 5.2: Backlog du sprint 3

## 5.2 Conception du sprint

### 5.2.1 Le diagramme de classes

La figure 5.2 montre le diagramme de classes du troisième sprint.

- **Sprint** : Cette classe contient la liste des sprints.
- **Task** : Cette classe contient la liste des tâches.
- **Comment** : Cette classe représente les commentaires liés aux tâches.
- **Rating** : Cette classe sert à évalué les commentaires .
- **ESprintStatus** : Cette énumération illustre les états d'un sprint.
- **ETaskStatus** : Cette énumération montre les états d'une tâche.
- **ETaskPriority** : Cette énumération respésente les priorités des tâches.

## CHAPITRE 5. SPRINT 3 : GESTION DES SPRINTS ET GESTION DES TÂCHES

---

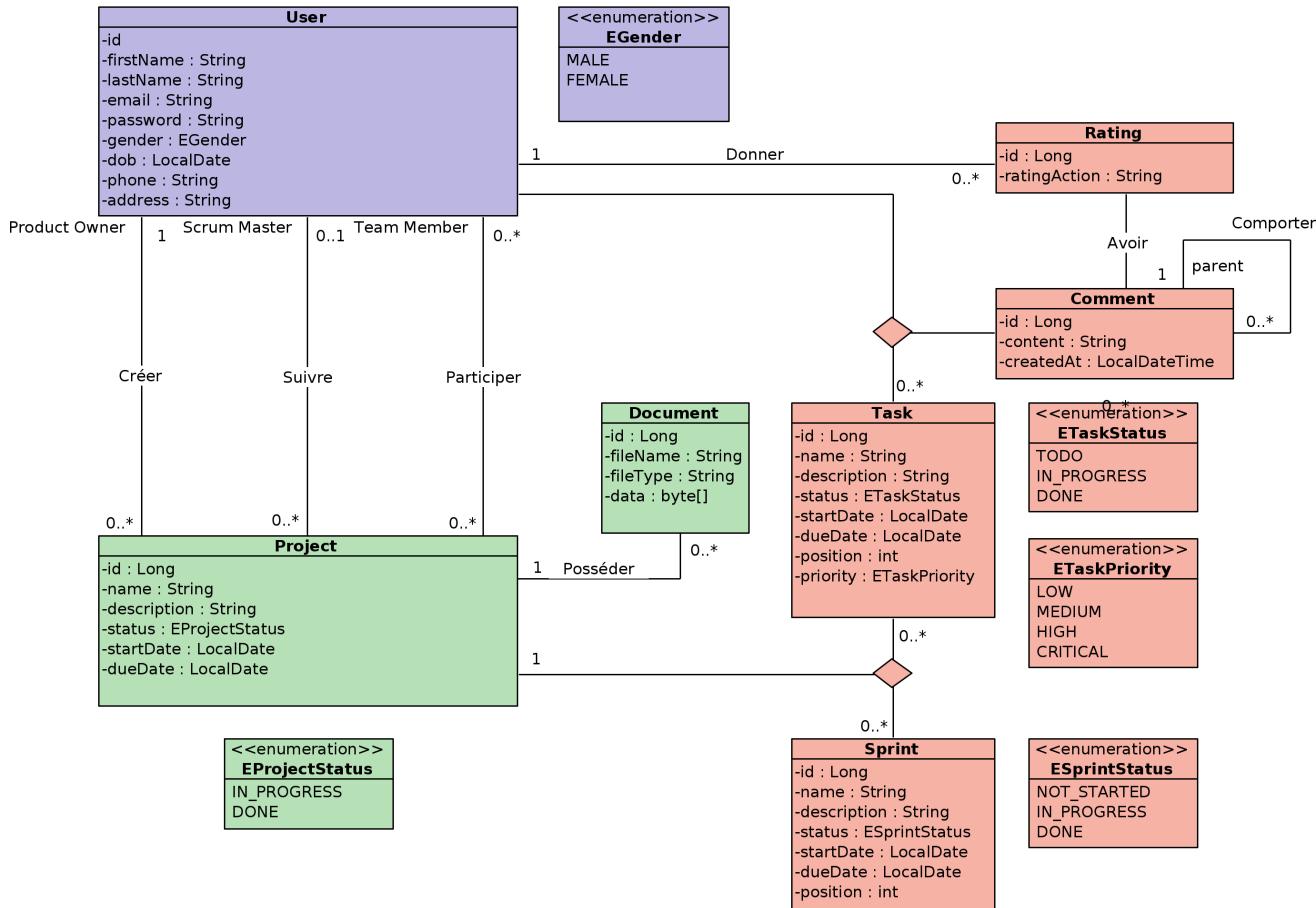
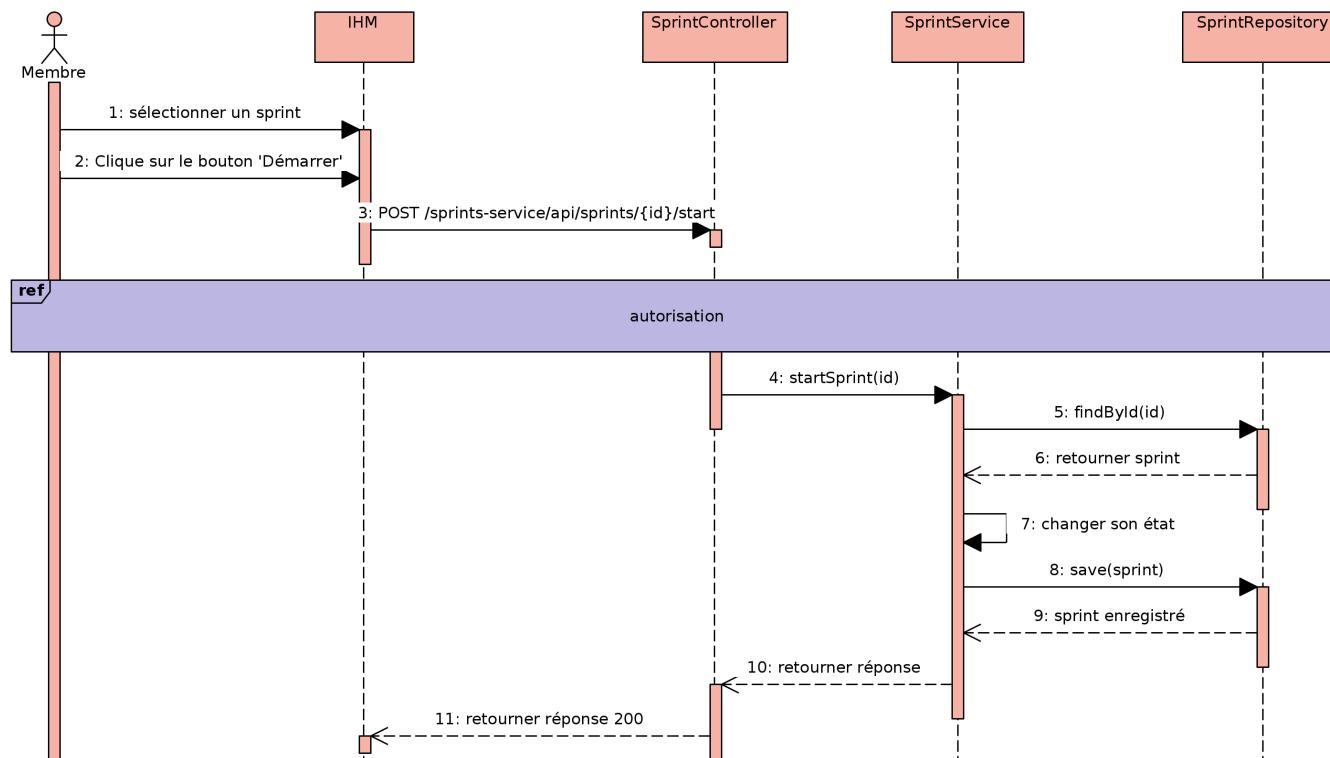


FIGURE 5.2 — Le diagramme de classes du sprint 3

## 5.2.2 les diagrammes de séquences

### 5.2.2.1 Diagramme de séquence « Démarrer un sprint »

La figure 5.3 illustre le diagramme de séquence qui permet de démarrer un sprint.



**FIGURE 5.3** — Le diagramme de séquence « Démarrer un sprint »

### 5.2.2.2 Diagramme de séquence « Crée une nouvelle tâche »

La figure 5.4 représente le diagramme de séquence qui permet de créer un nouvelle tâche.

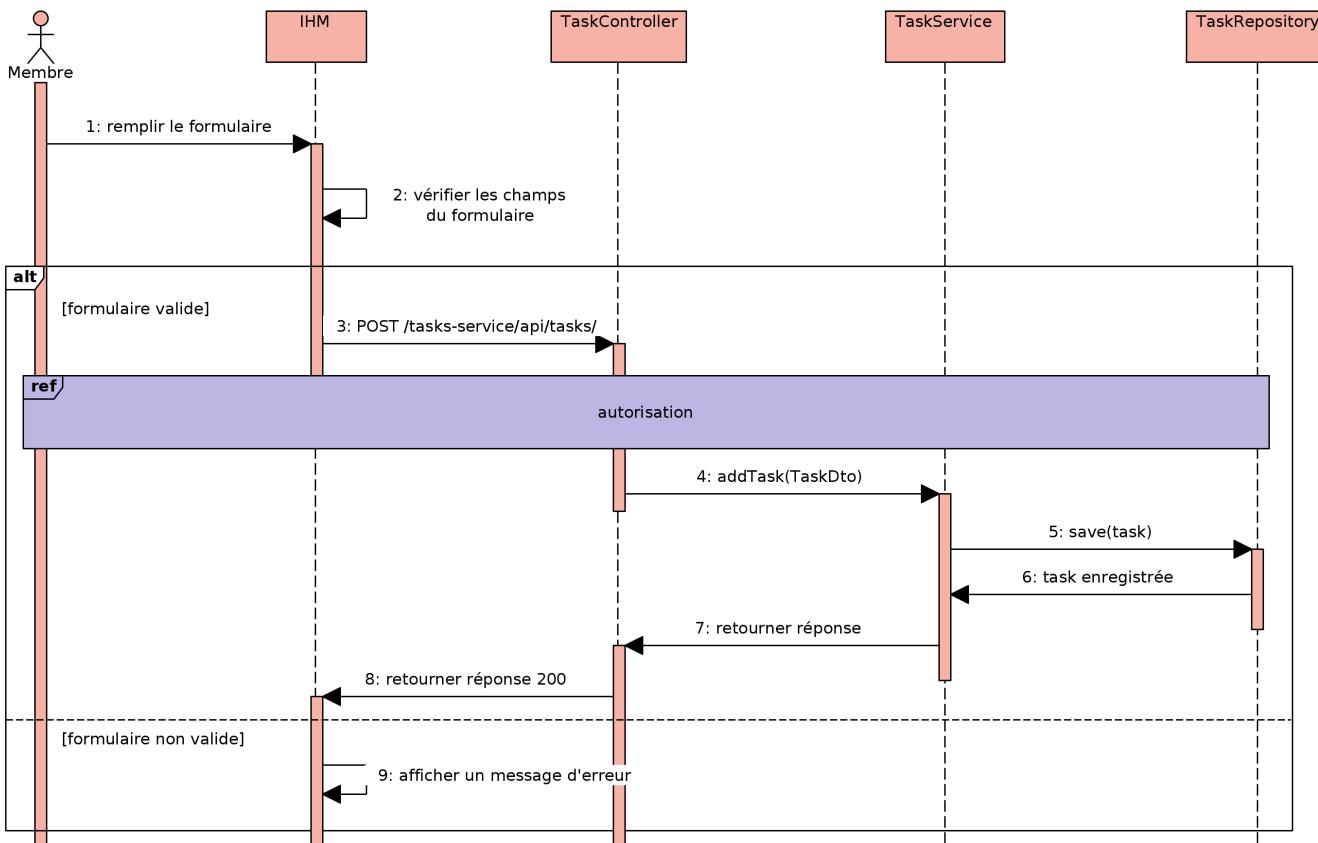


FIGURE 5.4 — Le diagramme de séquence « Crée une nouvelle tâche »

### 5.3 Revue du sprint

Pour clôturer, dans cette section nous présentons les interfaces de la revue du troisième sprint.

#### 5.3.1 L'interface de création d'un sprint

La figure 5.5 représente l'interface dédiée à l'ajout d'un sprint. Il faut renseigner le nom du sprint, date début, date fin et la description.

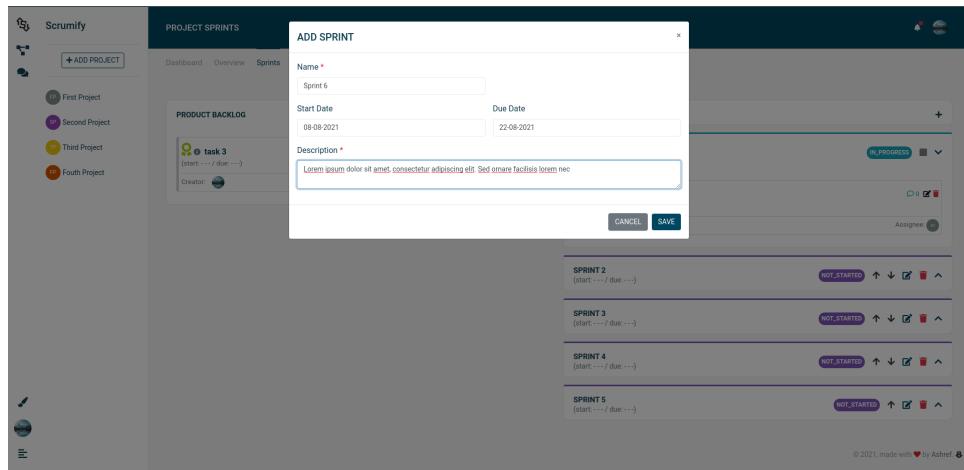


FIGURE 5.5 — L'interface de création d'un sprint

## CHAPITRE 5. SPRINT 3 : GESTION DES SPRINTS ET GESTION DES TÂCHES

### 5.3.2 L'interface de création d'une tâche

La figure 5.6 illustre l'interface dédiée à l'ajout d'une tâche. Il faut renseigner le nom, date début, date fin et la description.

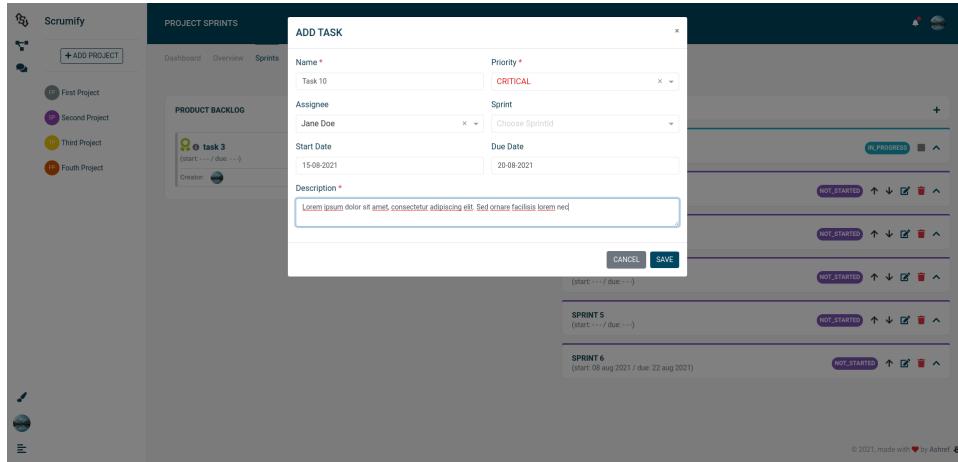


FIGURE 5.6 — L'interface de création d'une tâche

### 5.3.3 L'interface pour consulter « Product Backlog » et « Sprint Backlogs »

La figure 5.7 décrit l'interface qui donne une vue globale sur le Product Backlog, de sprint en cours et les sprints planifiés. Nous pouvons affecter une tâche à un sprint par un simple « Drag & Drop ».

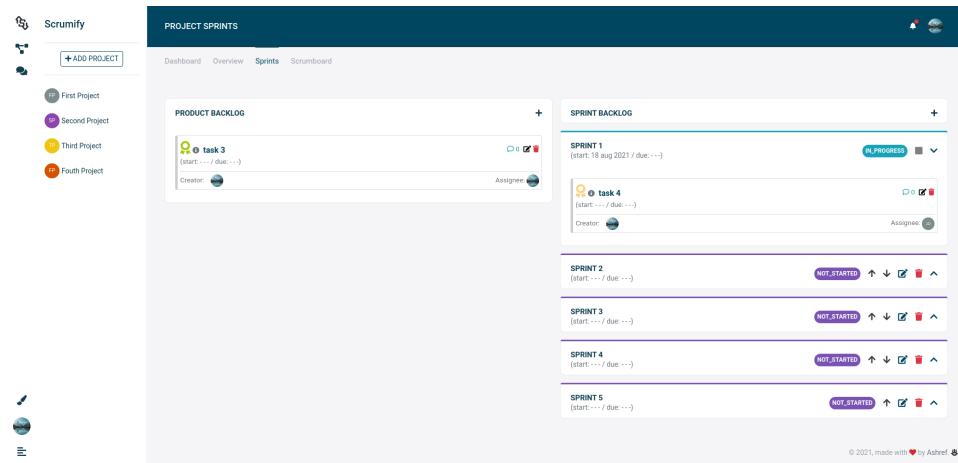


FIGURE 5.7 — L'interface pour consulter « Product Backlog » et « Sprint Backlogs »

## CHAPITRE 5. SPRINT 3 : GESTION DES SPRINTS ET GESTION DES TÂCHES

### 5.3.4 L'interface de la liste des commentaires

L'interaction entre les utilisateurs est nécessaire. Les commentaires est un moyen fiable pour enrichir la tâche et la mieux comprendre.

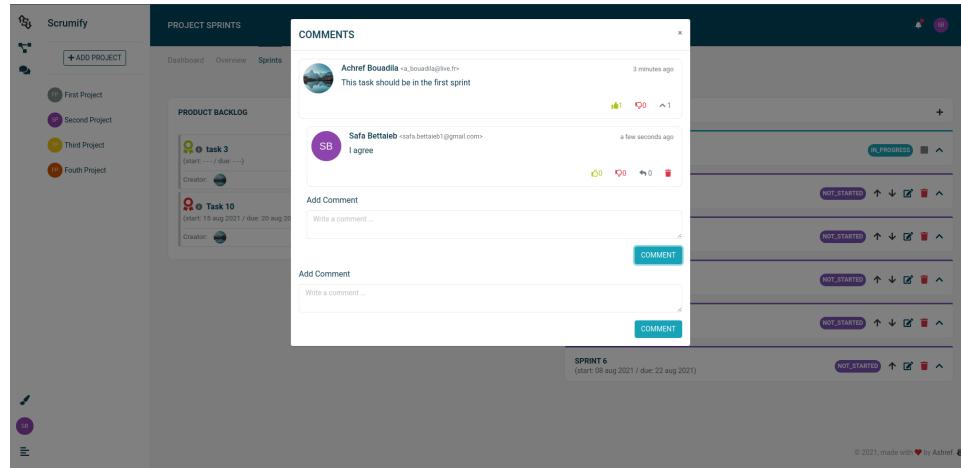


FIGURE 5.8 — L'interface de la liste des commentaires

### 5.3.5 L'interface d'avancement des tâches « Scrumboard »

Le scrumboard présente un graphique qui donne des métriques sur la progression des tâches dans un sprint et permet aux utilisateurs de contrôler le status d'une tâche par un « Drag & Drop ». Une tâche peut avoir trois statuts (à faire, en cours, exécutée).

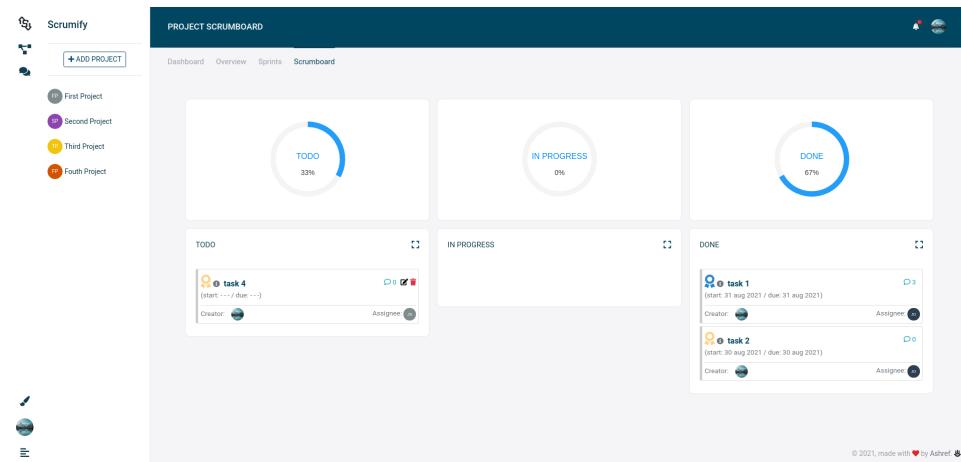


FIGURE 5.9 — L'interface d'avancement des tâches « Scrumboard »

## **Conclusion**

Dans ce chapitre nous avons présenté les besoins fonctionnels du troisième sprint à travers un diagramme de cas d'utilisation, ainsi que le Sprint Backlog, aussi nous avons présenté la conception en utilisant les diagrammes de séquences objet et le diagramme de classes et nous avons clôturé notre chapitre par la présentation des interfaces de ce sprint.

Chapitre **6**

# Sprint 4 : Tableau de bord, système de messagerie et système de notifications

## Plan

6.1	Analyse du sprint . . . . .	60
6.2	Conception du sprint . . . . .	62
6.3	Revue du sprint . . . . .	65

## CHAPITRE 6. SPRINT 4 : TABLEAU DE BORD, SYSTÈME DE MESSAGERIE ET SYSTÈME DE NOTIFICATIONS

---

### Introduction

Dans ce chapitre nous allons implémenter le tableau bord, le système de messagerie et le système de notifications. Ce chapitre explique le cycle de vie du quatrième sprint, à savoir le « Sprint Backlog », la conception et la réalisation.

## 6.1 Analyse du sprint

La phase d'analyse rende les besoins fonctionnels de ce sprint mieux compréhensible et plus clair. Dans ce contexte nous allons entamer à modéliser le diagramme de cas d'utilisation détaillé ainsi que le « Sprint Backlog ».

### 6.1.1 Diagrammes de cas d'utilisation

La figure 6.1 représente le diagramme de cas d'utilisation global du quatrième sprint.

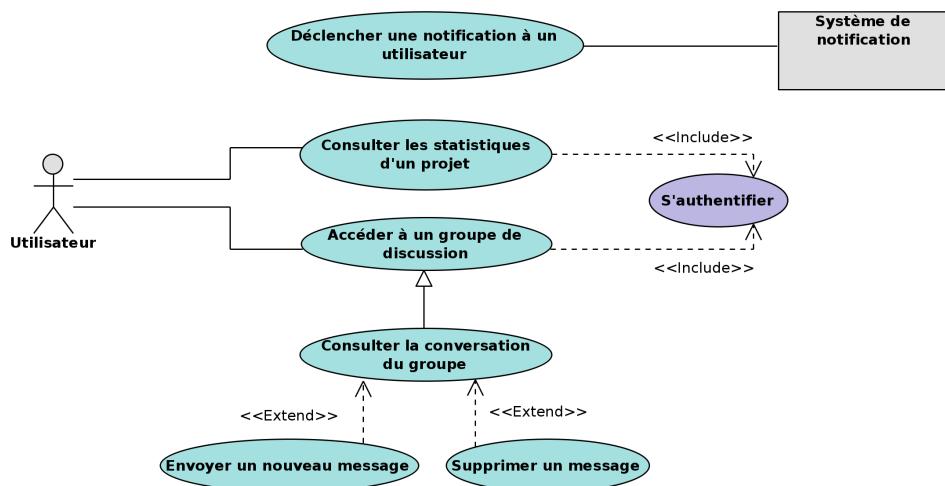


FIGURE 6.1 — Le diagramme de cas d'utilisation global de sprint 4

## CHAPITRE 6. SPRINT 4 : TABLEAU DE BORD, SYSTÈME DE MESSAGERIE ET SYSTÈME DE NOTIFICATIONS

---

### 6.1.1.1 Description textuelle de cas d'utilisation « Envoyer un nouveau message »

<b>Nom du cas d'utilisation</b>	Envoyer un nouveau message
<b>Acteurs</b>	Utilisateur
<b>Description</b>	Ce cas permet au membre d'envoyer un nouveau message.
<b>Pré-condition</b>	L'acteur doit être authentifié
<b>Scénario nominal</b>	<ol style="list-style-type: none"><li>1. L'acteur accède au groupe de discussion.</li><li>2. Le système affiche la conversation du groupe.</li><li>3. L'acteur écrit un message.</li><li>4. L'acteur confirme l'envoi.</li><li>5. Le système sauvegarde le message à la base de données.</li></ol>
<b>Scénario alternative</b>	3.1 : Si les champs sont erronés ou vides. - Le système les indique dans le formulaire.
<b>Post-condition</b>	Le message sera envoyé.

TABLE 6.1 — Description textuelle de cas d'utilisation « Envoyer un nouveau message »

### 6.1.2 Sprint Backlog

Le tableau 6.2 illustre le Sprint Backlog du quatrième sprint.

US-ID	User Story		Priorité	Estimation en (jours)
	En tant que	Je peux		
J.1	Utilisateur	Accéder à un groupe de discussion	32	3
J.2	Utilisateur	Consulter la conversation du groupe	33	3
J.3	Utilisateur	Envoyer un nouveau message	34	2
J.4	Utilisateur	Supprimer un message	35	1
K.1	Système	Notifier l'utilisateur après avoir rejoint un nouveau projet.	36	1
K.2	Système	Notifier l'utilisateur lorsque le projet est mis à jour.	37	1

## CHAPITRE 6. SPRINT 4 : TABLEAU DE BORD, SYSTÈME DE MESSAGERIE ET SYSTÈME DE NOTIFICATIONS

---

K.3	Système	Notifier l'utilisateur lorsque le projet est fermé.	38	1
K.4	Système	Notifier l'utilisateur de chaque nouveau sprint.	39	1
K.5	Système	Notifier l'utilisateur lorsque le sprint est mis à jour.	40	1
K.6	Système	Notifier l'utilisateur lorsque le sprint est supprimé.	41	1
K.7	Système	Notifier l'utilisateur de chaque nouvelle tâche.	42	1
K.8	Système	Notifier l'utilisateur lorsque la tâche est mise à jour.	43	1
K.9	Système	Notifier l'utilisateur lorsque la tâche est supprimée.	44	1
K.10	Système	Notifier l'utilisateur des nouveaux commentaires.	45	1
K.11	Système	Notifier l'utilisateur des nouveaux messages.	46	1
L.1	PO, SM, Membre	Consulter les statistiques	47	8

TABLE 6.2: Backlog du sprint 4

## 6.2 Conception du sprint

### 6.2.1 Le diagramme de classes

La figure 6.2 montre le diagramme de classes du quatrième sprint.

- **Notification** : Cette classe contient toutes les notifications de notre application.
- **ChatGroup** : Cette classe représente les groupes de discussion notre application.
- **Message** : Cette classe comprend les messages de discussion des membres.

## CHAPITRE 6. SPRINT 4 : TABLEAU DE BORD, SYSTÈME DE MESSAGERIE ET SYSTÈME DE NOTIFICATIONS

---

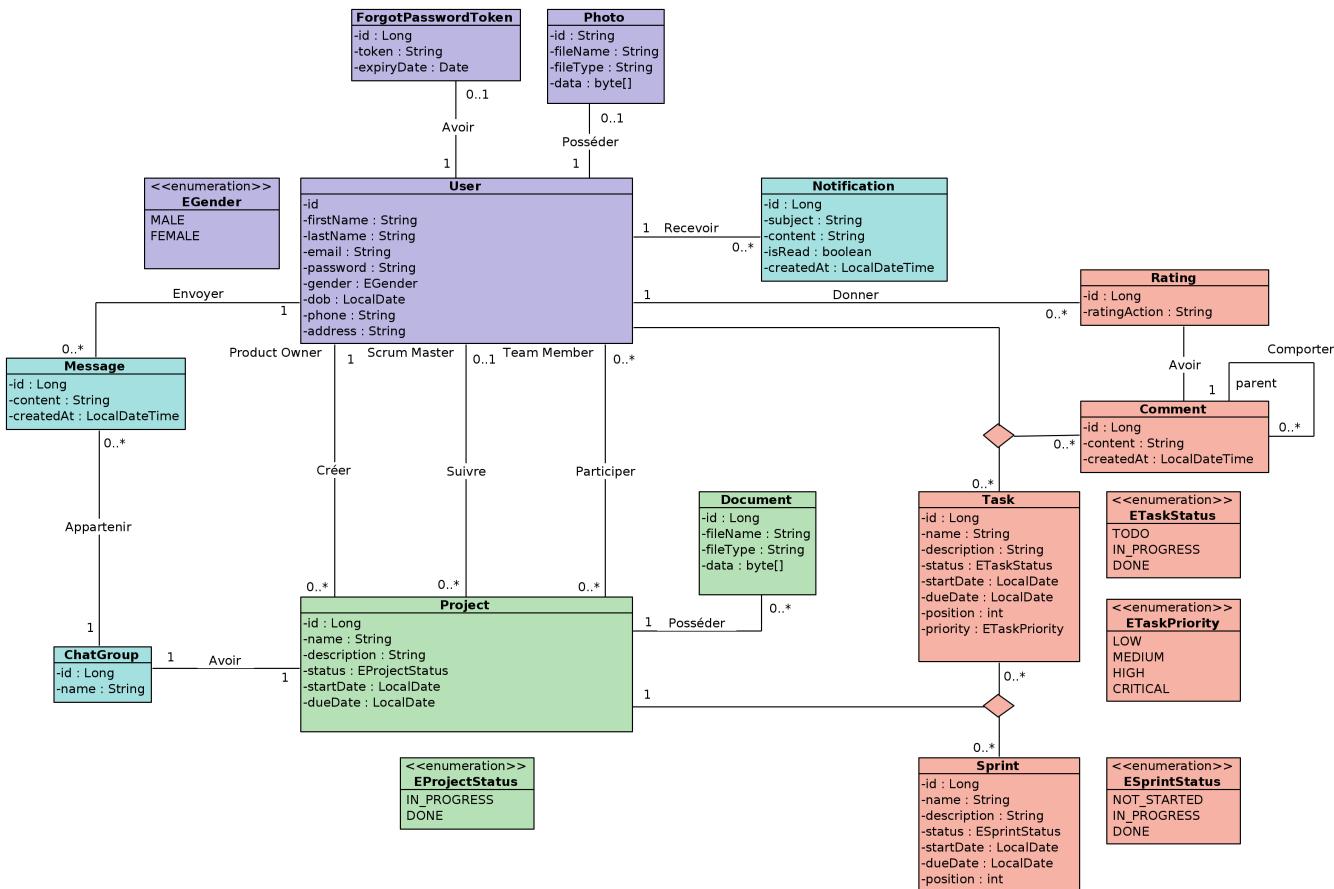
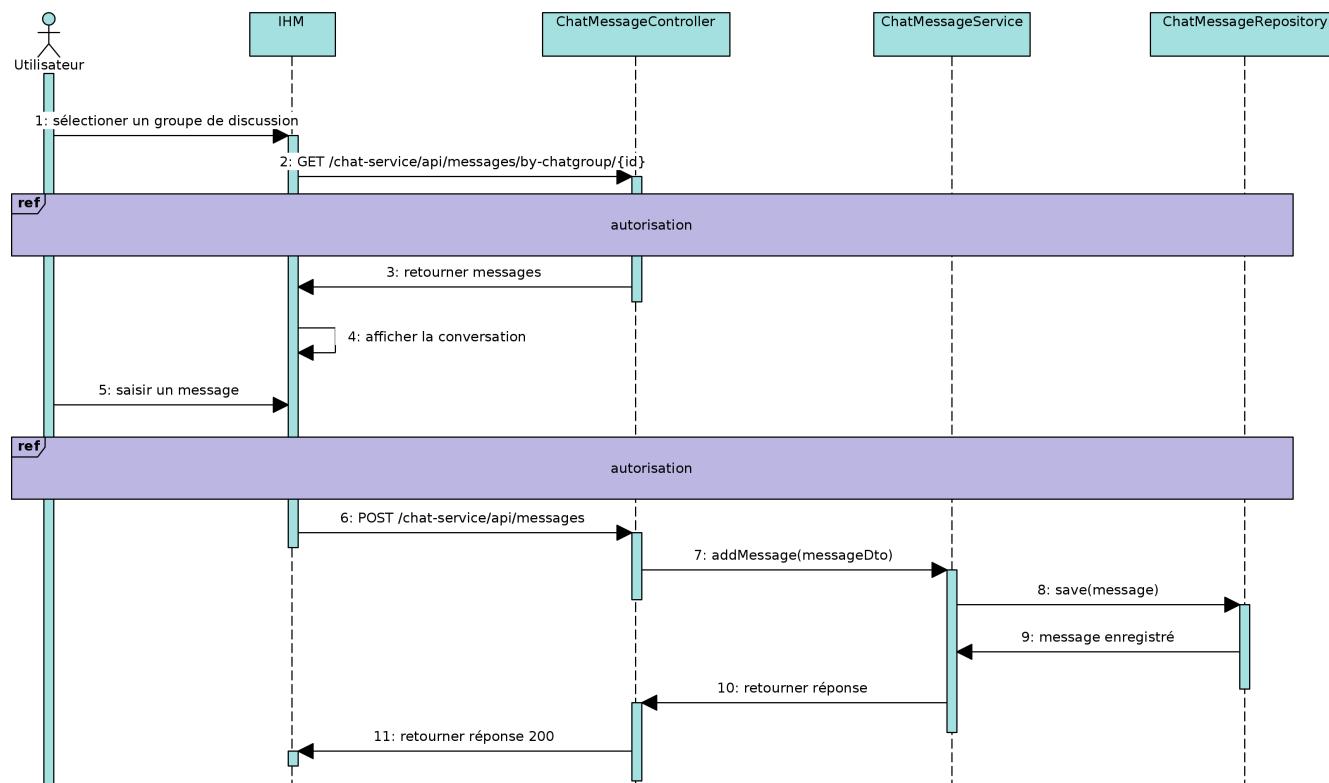


FIGURE 6.2 — Le diagramme de classes du sprint 4

## 6.2.2 les diagrammes de séquences

### 6.2.2.1 Diagramme de séquence « Envoyer un nouveau message »

La figure 6.3 représente le diagramme de séquence qui permet à l'utilisateur d'envoyer un nouveau message.



**FIGURE 6.3** — Le diagramme de séquence « Envoyer un nouveau message »

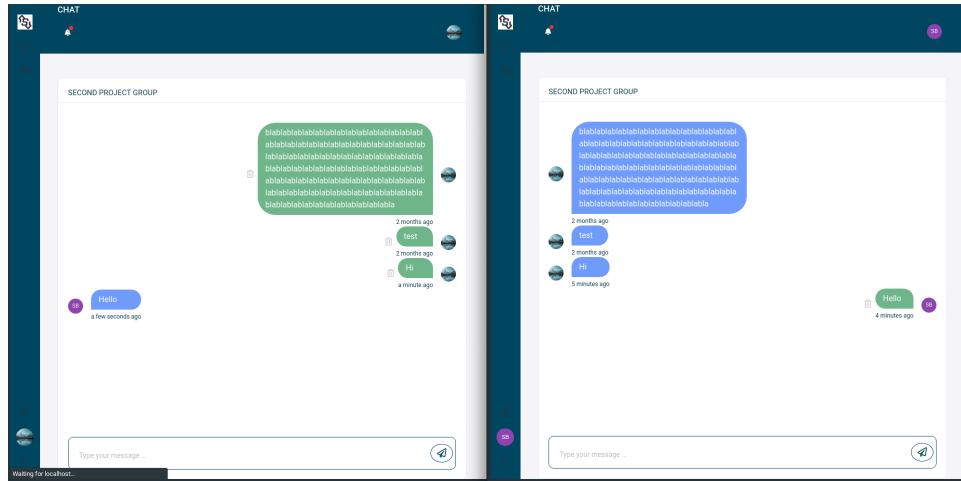
## CHAPITRE 6. SPRINT 4 : TABLEAU DE BORD, SYSTÈME DE MESSAGERIE ET SYSTÈME DE NOTIFICATIONS

## 6.3 Revue du sprint

Pour clôturer, dans cette section nous présentons les interfaces de la revue du quatrième sprint.

### 6.3.1 L'interface de messagerie

Pour interagir rapidement, le système de messagerie est un bon moyen de transmission d'informations. Cette interface montre une conversation entre deux membres.



**FIGURE 6.4** — L'interface de messagerie

## CHAPITRE 6. SPRINT 4 : TABLEAU DE BORD, SYSTÈME DE MESSAGERIE ET SYSTÈME DE NOTIFICATIONS

### 6.3.2 L'interface de la liste des notifications

Cette interface donne une vision clairvoyante sur les projets, sprints, tâches à l'aide du système de notifications.

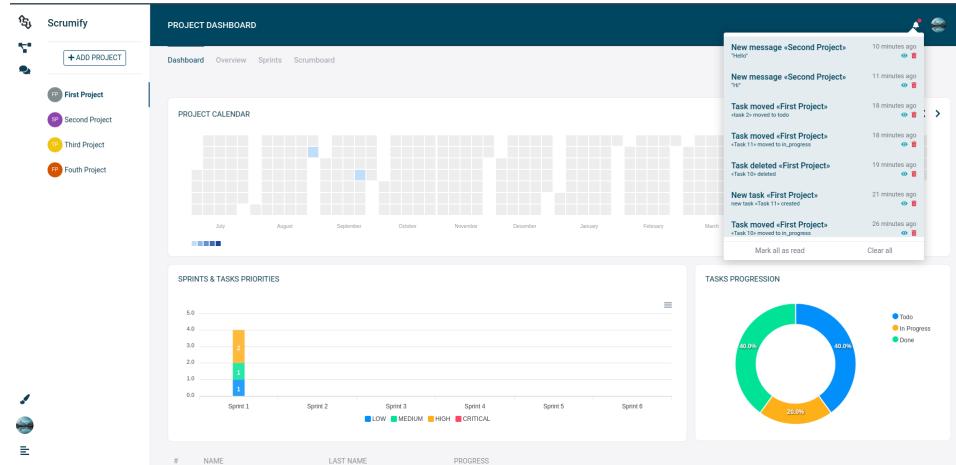


FIGURE 6.5 — L'interface de la liste des notifications

### 6.3.3 L'interface de tableau de bord

Cette interface montre le calendrier d'un projet, la répartition des tâches de chaque sprint en termes de priorité et les membres qui travaillent dessus.

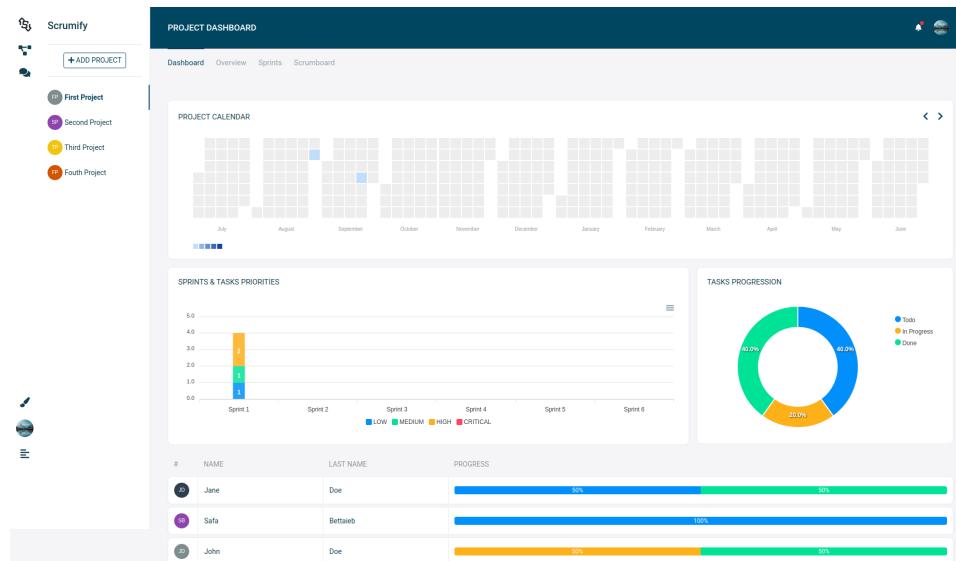


FIGURE 6.6 — L'interface de tableau de bord

## **CHAPITRE 6. SPRINT 4 : TABLEAU DE BORD, SYSTÈME DE MESSAGERIE ET SYSTÈME DE NOTIFICATIONS**

---

### **Conclusion**

Dans ce chapitre nous avons présenté les besoins fonctionnels du quatrième sprint à travers un diagramme de cas d'utilisation, ainsi que le Sprint Backlog, aussi nous avons présenté la conception en utilisant les diagrammes de séquences objet et le diagramme de classes et nous avons clôturé notre chapitre par la présentation des interfaces de ce sprint.

# Conclusion générale et perspectives

**L**e présent rapport a détaillé les étapes effectuées pour mettre en place une application web de gestion des projets basée sur l'architecture microservices. Afin d'aboutir à ce résultat, nous avons d'abord commencé par la présentation de contexte générale du projet . Ensuite, nous avons présenté des différents besoins et exigences relevés ainsi que l'architecture adoptée, pour aboutir par la suite à planification des sprints. Enfin nous avons abordé l'étape de réalisation de ces sprints au cours de laquelle nous avons traduit notre modélisation conceptuelle en une implémentation physique moyennant les différentes technologies et techniques choisies.

Hormis le côté technique et fonctionnel, ce projet a été une opportunité pour appréhender le travail dans une hiérarchie professionnelle au sein d'une grande société et les difficultés inhérentes comme la répartition du temps et des efforts. En participant au développement du projet, nous avons pu découvrir la rigueur nécessaire à la satisfaction du client ainsi que les bonnes pratiques nécessaires à la réalisation d'un produit de qualité.

Cependant, nous pouvons encore améliorer cette application en offrant la possibilité d'intégrer un outil de gestion de versions tels que Github.

Nous souhaitons, enfin, que ce modeste travail apporte l'évolution et la satisfaction aux responsables d'INNOVUP, aux membres du jury et à toute personne intéressée par le domaine de gestion des projets et l'application de méthodologie scrum.

Pour résumer, ce stage m'a fourni une opportunité d'améliorer mes compétences techniques et sociales tout en travaillant avec une équipe professionnelle.

# Netographie

1. Jira, <https://www.atlassian.com/fr/software/jira/guides/getting-started/overview>, dernier accès : 10/05/2021
2. VivifyScrum : <https://www.capterra.fr/software/149624/vivifyscrum>, dernier accès : 10/05/2021
3. Scrum : <https://agiliste.fr/introduction-methodes-agiles/>, dernier accès : 10/05/2021
4. UML : <https://www.lucidchart.com/pages/fr/langage-uml>, dernier accès : 12/05/2021
5. Microservices :  
<https://www.redhat.com/fr/topics/microservices/what-are-microservices>, dernier accès : 15/05/2021
6. Spring Tool Suite : <https://devstory.net/10249/installer-spring-tool-suite-pour-eclipse>, dernier accès : 01/06/2021
7. Visual Studio Code : <https://framalibre.org/content/visual-studio-code>, dernier accès : 02/06/2021
8. Github : <https://www.lemagit.fr/definition/GitHub>, dernier accès : 02/06/2021
9. Posman : <https://www.postman.com/>, dernier accès : 02/06/2021
10. MySQL Workbench : <https://www.mysql.com/fr/products/workbench/>, dernier accès : 02/06/2021
11. Visual Paradigm : [https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963\\_visualparadi.html](https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html), dernier accès : 04/06/2021
12. Overleaf : <https://paris-sorbonne.libguides.com/c.php?g=497641&p=4637541>, dernier accès : 04/06/2021
13. Angular - [https://www.tutorialspoint.com/angular2/angular2\\_overview.htm](https://www.tutorialspoint.com/angular2/angular2_overview.htm), dernier accès : 06/06/2021
14. Spring Boot - <https://www.javatpoint.com/spring-boot-tutorial>, dernier accès : 06/06/2021
15. RabbitMQ : <https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html>, dernier accès : 08/06/2021
16. Websocket : <https://www.baeldung.com/websockets-spring>, dernier accès : 07/06/2021
17. MySQL : <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-mysql/>, dernier accès : 07/06/2021