



# Rapport de Projet de Fin d'Études

Pour obtenir le

**Diplôme d'Ingénieur en Génie Informatique**

**Option : Nouvelles technologies et sécurité**

*Présenté et soutenu publiquement le 06 octobre 2021*

*Par*

**Wissal FERSI**

---

**Conception, développement et déploiement  
d'une application de gestion de projet pour  
« Billcom Consulting »**

---

## Composition du jury

<b>Madame</b>	Narjes HACHANI	<b>Président</b>
<b>Madame</b>	Emna SOUSSI	<b>Rapporteur</b>
<b>Monsieur</b>	Heni OUELHEZI	<b>Encadrant Entreprise</b>
<b>Madame</b>	Ines BAYOUDH SAADI	<b>Encadrant ENSIT</b>

**Année universitaire : 2020-2021**

## Dédicaces

*À mon très cher Père Ali, à ma tendre Mère Sihem, autant de phrases aussi expressives soient-elles ne sauraient exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consentis pour mon éducation et mon bien-être.*

*Votre affection inconditionnée me couvre, votre sagesse me guide et votre présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles. Je vous remercie pour tout le soutien et l'amour que vous portez à mon égard et j'espère que votre bénédiction m'accompagne toujours. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices.*

*Puisse Dieu, le Très Haut, vous accorde santé, bonheur et longue vie.*

*À mon frère et ma sœur, à vous mes amours, nulle dédicace n'exprime ma gratitude de vous avoir dans ma vie, que Dieu le tout puissant vous préserve et vous accorde bonheur et réussite.*

*À ma chère Nesrine, en souvenir des instants dont nous avons partagé les meilleurs et les plus agréables moments. Tu m'as chaleureusement soutenu, réconforté et encouragé tout au long de mon parcours. Puissent nos liens amicaux se consolider et se pérenniser encore plus.*

*À toute ma famille, à tous mes ami(e)s, Je vous dédie ce travail en reconnaissance de l'amour que vous m'offrez quotidiennement et votre bonté exceptionnelle.*

*À tous ceux que j'aime et ceux qui m'aimaient.*

*Wissal Fersi*

## **Remerciements**

Au terme de ce travail, je tenais à remercier toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce projet de fin d'études.

Mr. Heni OUELHEZI, mon encadrant à « Billcom Consulting », à qui je tiens à exprimer toute ma gratitude pour l'aide qu'il m'a apportée durant toutes les phases de ce stage. Sa disponibilité, son encadrement, sa pédagogie et ses conseils m'ont été précieux pour atteindre les objectifs de ce projet dans les délais convenus.

Mme. Ines BAYOUDH SAADI, mon encadrante à L'ENSIT, pour la qualité de son enseignement, son encouragement et ses conseils qui ont été très bénéfiques pour le bon cheminement de ce rapport.

Aux membres du jury pour m'avoir honoré en acceptant d'examiner ce travail.

A tous les professeurs de l'Ensit qui depuis 2018 jusqu'à ce jour ont contribué à notre formation d'ingénieur.

# Table des matières

<i>Introduction générale</i> .....	<b>I</b>
<b>Chapitre 1 : Cadre général du travail .....</b>	<b>3</b>
Introduction .....	3
1. Cadre du projet .....	3
2. Présentation de l'organisme d'accueil.....	3
3. Présentation du projet.....	4
3.1. Etude et critique de l'existant.....	5
3.1.1. Logiciels payants.....	5
3.1.2. Logiciels gratuits .....	6
3.3. Problématique.....	9
3.4. Solution proposée.....	9
4. Méthodologie de travail .....	10
4.1. Etude comparative des méthodologies agiles.....	10
4.2. Méthodologie adoptée : SCRUM.....	11
4.2.1. Les rôles SCRUM .....	11
4.2.2. Les événements SCRUM .....	11
4.2.3. Les artefacts SCRUM.....	12
4.3. Formalisme adopté .....	12
Conclusion.....	13
<b>Chapitre 2 : Phase de Planification .....</b>	<b>14</b>
Introduction .....	14
1. Présentation de l'équipe .....	14
2. Capture des besoins .....	14
2.1. Identification des acteurs.....	14
2.2. Backlog de produit .....	15
2.3. Besoins fonctionnels .....	20
2.4. Besoins non fonctionnels .....	22
3. Analyse et spécifications des besoins.....	23
3.1. Diagramme de cas d'utilisation global.....	23
3.2. Architecture de la solution .....	24
3.2.1. Architecture physique .....	24
3.2.2. Architecture logique .....	24
4. Environnement du travail.....	27
4.1. Environnement matériel .....	27
4.2. Environnement logiciel .....	27
4.2.1. Outils de développement et de modélisation.....	27
4.2.2. Outils de test.....	28
4.2.3. Langages de programmation .....	28
4.2.4. Technologies utilisées .....	29
4.2.5. Outils de déploiement.....	29
5. Planification des sprints .....	29
Conclusion.....	30

<b>Chapitre 3 : Release 1 « Gestion d'accès et des utilisateurs » .....</b>	<b>31</b>
Introduction .....	31
1. Présentation de la release 1 .....	31
2. Sprint 1 : Authentification et gestion d'accès .....	31
2.1. Mécanisme de sécurité .....	32
2.2. Backlog de produit du sprint .....	32
2.3. Analyse.....	33
2.3.1. Diagramme de cas d'utilisation.....	33
2.3.2. Raffinement des cas d'utilisation .....	33
2.3.3. Diagramme de séquence.....	37
2.4. Conception .....	38
2.5. Réalisation.....	40
2.5.1. Interface d'authentification .....	40
2.5.2. Interface de récupération du mot de passe .....	41
2.5.3. Interface de modification du mot de passe.....	42
2.5.4. Interface d'activation et de désactivation d'un compte utilisateur.....	42
3. Sprint 2 : Gestion des utilisateurs et des profils .....	43
3.1. Backlog de produit .....	43
3.2. Analyse.....	43
3.2.1. Diagramme de cas d'utilisation.....	44
3.2.2. Raffinement des cas d'utilisation .....	44
3.3. Conception .....	47
3.3.1. Diagramme de classes .....	47
3.3.2. Diagramme de séquence.....	48
3.4. Réalisation .....	50
3.4.1. Interface d'ajout d'un utilisateur .....	50
3.4.2. Interface de consultation de la liste des utilisateurs .....	51
3.4.3. Interface de recherche sur la liste des utilisateurs .....	52
3.4.4. Interfaces de gestion du rôle d'un utilisateur .....	53
3.4.5. Interface de consultation du profil d'un utilisateur .....	54
Conclusion.....	55
<b>Chapitre 4 : Release 2 « Gestion et planification du projet ».....</b>	<b>56</b>
Introduction .....	56
1. Présentation de la release 2 .....	56
2. Sprint 3 : Gestion et planification du projet et de ses équipes .....	56
2.1. Backlog de produit du sprint .....	56
2.2. Analyse.....	57
2.2.1. Diagramme de cas d'utilisation.....	58
2.2.2. Raffinement des cas d'utilisation .....	58
2.2.3. Diagramme de séquence.....	62
2.3. Conception .....	63
2.3.1. Diagramme de classes .....	63
2.3.2. Diagramme de séquence.....	64

2.4. Réalisation.....	66
2.4.1. Interface de création d'un projet .....	66
2.4.2. Interface de consultation du détail d'un projet.....	67
2.4.3. Opérations sur la liste des projets.....	67
2.4.4. Interface de consultation de la liste des projets/chef de projet.....	69
2.4.5. Interface de création d'une équipe .....	70
2.4.6. Interface de consultation des équipes.....	70
3. Sprint 4 : Gestion et planification des tâches .....	71
3.1. Backlog de produit du sprint .....	71
3.2 Analyse.....	72
3.2.1. Diagramme de cas d'utilisation.....	72
3.2.2. Raffinement des cas d'utilisation .....	72
3.3. Conception .....	75
3.3.1. Diagramme de classes .....	75
3.3.2. Diagramme de séquence.....	76
3.4. Réalisation.....	77
3.4.1. Interface d'ajout d'une tâche.....	77
3.4.2. Interface de consultation de la liste des tâches.....	78
3.4.3. Interface de consultation des détails d'une tâche .....	79
3.4.4. Interface d'ajout des fichiers .....	79
3.4.5. Interface de consultation des tâches entre deux dates .....	79
3.4.6. Interface de modification de la date de livraison de tâche .....	80
Conclusion.....	81
<b>Chapitre 5 : Release 3 « Gestion des évènements, statistiques et déploiement ».....</b>	<b>82</b>
Introduction .....	82
1. Présentation de la release 3 .....	82
2. Sprint 5 : Gestion des événements, des notifications et des rappels .....	82
2.1. Présentation des WebSockets.....	83
2.2. Backlog de produit du sprint 5 .....	83
2.3. Analyse.....	84
2.3.1. Diagramme de cas d'utilisation.....	84
2.3.2. Raffinement des cas d'utilisation .....	85
2.4. Conception .....	87
2.4.1. Diagramme de classes .....	87
2.4.2. Diagramme de séquence.....	88
2.5. Réalisation.....	91
2.5.1. Interface du calendrier.....	91
2.5.2. Interfaces des opérations sur le calendrier .....	92
2.5.3. Interface de gestion des notifications et des rappels .....	93
2.5.4. Interface de la messagerie en temps réel .....	94

3. Sprint 6 : Gestion des tableaux de bord.....	94
3.1. Backlog de produit du sprint 6 .....	94
3.3. Analyse.....	96
3.3.1. Diagramme de cas d'utilisation.....	96
3.3.2. Raffinement des cas d'utilisation .....	96
3.5. Réalisation.....	98
3.5.1. Interfaces des tableaux de bord spécifiques aux projets .....	98
3.5.2. Interfaces des tableaux de bord spécifiques aux équipes .....	100
3.5.3. Interface des tableaux de bord spécifiques à chaque membre .....	102
4. Sprint 7 : Test, déploiement et mise en place d'une chaîne CI/CD .....	103
4.1. Technical Story .....	103
4.2. Tests Automatisés .....	104
4.2.1. Etude comparative des outils de tests.....	104
4.2.2. Développement des tests selenium.....	105
4.3. Déploiement de l'application .....	106
4.3.1. Architecture de déploiement .....	106
4.3.2. Processus de déploiement.....	107
4.4. Mise en place d'une chaîne CI/CD .....	108
4.4.1. Intégration continue/Continous Integration (CI).....	109
4.4.2. Déploiement Continu/Continous Deployment (CD).....	109
4.4.3. Application de la CI/CD sur l'application.....	109
Conclusion.....	114
<i>Conclusion générale</i> .....	<b>115</b>
<i>Référence</i> .....	<b>116</b>
<i>Annexes</i> .....	<b>119</b>
• Release 1 .....	119
• Release 2 .....	124
• Release 3 .....	129

# Table des figures

Figure 1 : Logo de « Billcom Consulting » [1] .....	4
Figure 2 : Organigramme de la société .....	4
Figure 3 : Interface de gestion de tâches de l'outil Asana [3] .....	6
Figure 4 : Dashboard de l'outil Bitrix24 [3] .....	7
Figure 5 : Interface de gestion de tâches de Trello [3] .....	8
Figure 6 : Interface de gestion de tâches de Jira [3] .....	8
Figure 7 : Diagramme de cas d'utilisation global .....	23
Figure 8 : Architecture physique du système [9] .....	24
Figure 9 : Architecture Front-End [10] .....	25
Figure 10 : Diagramme de package.....	26
Figure 11 : Diagramme de cas d'utilisation du premier sprint .....	33
Figure 12 : Diagramme de séquence du scénario « S'authentifier ».....	37
Figure 13 : Diagramme de séquence du scénario « Récupérer mot de passe ».....	38
Figure 14 : Diagramme de classes du sprint numéro 1 .....	39
Figure 15 : Interface d'authentification .....	40
Figure 16 : Les messages des cas d'erreur.....	41
Figure 17 : Récupération du mot de passe oublié .....	41
Figure 18 : Interface de modification du mot de passe .....	42
Figure 19 : Interface d'activation/désactivation d'un compte.....	42
Figure 20 : Diagramme de cas d'utilisation du sprint numéro 2.....	44
Figure 21 : Diagramme de classes du sprint numéro 2 .....	47
Figure 22 : Diagramme de séquence du cas d'utilisation « Ajouter un compte utilisateur »..	49
Figure 23 : Interface d' ajout d'un compte utilisateur .....	50
Figure 24 : Cas d'erreur d'ajout du compte utilisateur.....	51
Figure 25 : Interface de la liste des utilisateurs .....	51
Figure 26 : Operations de recherche appliquée sur la liste .....	52
Figure 27 : Interface de la liste des rôle d'un utilisateur.....	53
Figure 28 : Supprimer un rôle de la liste des rôles .....	53
Figure 29 : Ajout d'un rôle à un utilisateur .....	53
Figure 30 : Consultation du profil utilisateur .....	54
Figure 31 : Mise à jour du profil .....	54
Figure 32 : Diagramme de cas d'utilisation du sprint numéro 3.....	58
Figure 33 : Raffinement du cas d'utilisation « Créer un projet » .....	58
Figure 34 : Raffinement du cas d'utilisation « Créer une équipe » .....	61
Figure 35 : Diagramme de séquence du cas d'utilisation « Supprimer une équipe ».....	62
Figure 36 : Diagramme de classes du sprint numéro 3 .....	63
Figure 37 : Diagramme de séquence du cas d'utilisation « Filtrer la liste des projets .....	65
Figure 38 : Interface du cas d'utilisation « Créer un projet » .....	66
Figure 39 : Interface du détail projet .....	67
Figure 40 : Mettre un projet à l'état « Stopped » .....	67
Figure 41 : Recherche d'un projet .....	69
Figure 42 : Interface des projets en attente .....	69
Figure 43 : Interface de la liste de projet assignés au chef de projet.....	70
Figure 44 : Interface de création d'une équipe.....	70
Figure 45 : Liste des équipes.....	70
Figure 46 : Diagramme des cas d'utilisation du sprint numéro 4 .....	72
Figure 47 : Raffinement du cas d'utilisation « Gérer les commentaires » .....	73

Figure 48 : Diagramme de classes du sprint 4 .....	75
Figure 49 : Diagramme de séquence du cas d'utilisation « Charger des fichiers » .....	76
Figure 50 : Diagramme de séquence du cas d'utilisation « Modifier le statut d'une tâche » ..	77
Figure 51 : Formulaire d'ajout d'une tâche .....	77
Figure 52 : Interface de la liste des tâches d'un membre.....	78
Figure 53 : Ajouter une tâche au statut « To-do ».....	78
Figure 54 : Consultation des détails d'une tâche .....	79
Figure 55 : Interface d'ajout des fichiers .....	79
Figure 56 : Consultation des tâches entre deux date .....	80
Figure 57 : Modification de la date de livraison d'une tâche en cours .....	80
Figure 58 : Diagramme des cas d'utilisation du sprint 5 .....	84
Figure 59 : Diagramme de classes du sprint 5 .....	87
Figure 60 : Diagramme de séquence du cas d'utilisation « Ajouter événement ».....	89
Figure 61 : Diagramme de séquence du cas d'utilisation « Consulter le calendrier ».....	90
Figure 62 : Interface du calendrier .....	91
Figure 63 : Interface du détail d'un événement .....	91
Figure 64 : Interface d'ajout d'un événement .....	92
Figure 65 : Interface de mise à jour d'un événement .....	92
Figure 66 : Interface d'annulation d'un événement .....	93
Figure 67 : Interface de la liste des rappels .....	93
Figure 68 : Interface de la liste des notifications .....	93
Figure 69 : Discussion instantanée entre deux membres .....	94
Figure 70 : Diagramme de cas d'utilisation du sprint 6.....	96
Figure 71 : Interface du tableau de bord des projets ayant le statut « Completed » .....	98
Figure 72 : Statistiques des projets retardés .....	99
Figure 73 : Statistiques des projets en cours .....	99
Figure 74 : Evaluation des charges du projet .....	100
Figure 75 : Interface du tableau de bord de travail de chaque équipe.....	101
Figure 76 : Avancement de chaque membre d'équipe.....	101
Figure 77 : Interface de consultation des jours en avance/en retard des tâches .....	102
Figure 78 : Interface des tableaux de bord globaux des équipes.....	102
Figure 79 : Interface du tableau de bord du membre .....	103
Figure 80 : Interface de résultats des tests selenium .....	105
Figure 81 : Test automatisé du cas d'utilisation « S'authentifier ».....	106
Figure 82 : Architecture de déploiement.....	106
Figure 83 : Génération d'une version de production de l'application angular.....	107
Figure 84 : Interface de déploiement du code sur GitHub .....	107
Figure 85 : Connecter le répertoire Heroku avec GitHub .....	108
Figure 86 : Interface de résultat du déploiement .....	108
Figure 87 : Création des clés privées .....	109
Figure 88 : Création des clés privées-publiques.....	110
Figure 89 : Liaison du projet Jenkins avec GitHub.....	110
Figure 90 : Liaison du projet Jenkins avec « Heroku Repoistory Git » .....	111
Figure 91 : Liaison des événements « push » avec Jenkins .....	111
Figure 92 : Connexion avec la branche master du « Heroku Git Repository ».....	111
Figure 93 : Invocation des tests unitaire.....	112
Figure 94 : Avertissement du Jenkins .....	112
Figure 95 : Résultat des tests unitaire.....	113
Figure 96 : Interface du Jenkins .....	113
Figure 97 : Architecture de déploiement.....	114

Figure 98 : Authentication spring security avec JWT.....	119
Figure 99 : Diagramme d'activités du cas d'utilisation « Désactiver un compte ».....	120
Figure 100 : Diagramme d'activités du cas d'utilisation « Modifier mot de passe ».....	120
Figure 101 : Diagramme de séquence du cas d'utilisation « Consulter les détails d'un utilisateur ».....	121
Figure 102 : Interface de consultation des détails d'un utilisateur.....	122
Figure 103 : Application des tests unitaires sur la couche user service .....	123
Figure 104 : Analyse de SonarLint .....	123
Figure 105 : Documentation de la première release.....	124
Figure 106 : Diagramme de séquence du cas d'utilisation « Consulter détail projet » .....	125
Figure 107 : Diagramme d'activités du cas d'utilisation « Modifier un projet » .....	126
Figure 108 : Interface des cas d'erreurs du cas d'utilisation « Créer un projet ».....	127
Figure 109 : Interface de consultation de la liste des projets .....	127
Figure 110 : Méthode « getProjectBy(id) » testée .....	128
Figure 111 : Analyse du SonarLint .....	128
Figure 112 : Documentation SwaggerUI de la release 2 .....	129
Figure 113 : Diagramme d'activités du cas d'utilisation « Modifier un événement ».....	130

## Table des tableaux

Tableau 1 : Tableau comparatif des logiciels de gestion de projet en termes de fonctionnalité/prix mensuel par utilisateur [2] .....	5
Tableau 2 : Comparaison de différentes méthodes agiles [4] .....	10
Tableau 3 : Identification des acteurs.....	15
Tableau 4 : Backlog de produit .....	16
Tableau 5 : Environnement Matériel.....	27
Tableau 6 : Outils de développement et modélisation .....	27
Tableau 7 : Outils de test.....	28
Tableau 8 : Langages de programmation .....	28
Tableau 9 : Technologies utilisées .....	29
Tableau 10 : Outils de déploiement.....	29
Tableau 11 : Déroulement de stage .....	30
Tableau 12 : Planification des releases .....	30
Tableau 13 : Backlog de produit du sprint 1 .....	32
Tableau 14 : Description textuelle du cas d'utilisation « S'authentifier ».....	34
Tableau 15 : Description textuelle du cas d'utilisation « Modifier le mot de passe » .....	35
Tableau 16 : Description textuelle du cas d'utilisation « Récupérer le mot de passe oublié »	36
Tableau 17 : Backlog de produit du sprint 2 .....	43
Tableau 18 : Description textuelle du cas d'utilisation « Ajouter un compte utilisateur » .....	45
Tableau 19 : Description textuelle du cas d'utilisation « Ajouter un rôle à un utilisateur »....	46
Tableau 20 : Description textuelle du cas d'utilisation « Modifier informations profil » .....	46
Tableau 21 : Backlog de produit du sprint 3 .....	57
Tableau 22 : Description textuelle du cas d'utilisation « Créer un projet ».....	59
Tableau 23 : Description textuelle du cas d'utilisation « Chercher un projet » .....	60
Tableau 24 : Description textuelle du cas d'utilisation « Créer une équipe ».....	61
Tableau 25 : Backlog de produit du sprint numéro 3.....	71
Tableau 26 : Description textuelle du cas d'utilisation « Chercher des tâches » .....	73
Tableau 27 : Description textuelle du cas d'utilisation « Ajouter une tâche ».....	74
Tableau 28 : Backlog de produit du sprint 5 .....	83
Tableau 29 : Description textuelle du cas d'utilisation « Annuler un évènement ».....	85
Tableau 30 : Description textuelle du cas d'utilisation « Marquer notification comme lue » .	86
Tableau 31 : Description textuelle du cas d'utilisation « Consulter la liste des rappels » .....	86
Tableau 32 : Backlog de produit du sprint 6 .....	94
Tableau 33 : Description textuelle du cas d'utilisation « Télécharger un tableau de bord » ...	97
Tableau 34 : Description textuelle du cas d'utilisation « Calculer statistiques projet » .....	97
Tableau 35 : Technical story du sprint 7 .....	104
Tableau 36 : Etude comparative des outils de tests disponibles [33] .....	104

# Introduction générale

Vu la situation sanitaire du pays à cause du covid-19, les entreprises sont parfois obligées de travailler à distance, cette solution peut présenter une complexité administrative en tenant compte de la charge du travail de la société et du nombre de projet qui sont traités en même temps, ainsi chaque chef de projet a besoin d'avoir une idée sur son projet, son état d'avancement, les affectations des tâches et les statistiques relatives à son projet.

En effet, chaque projet présente un ensemble d'activités finalisées par une équipe de projet découpée en sous équipes sous la responsabilité d'un chef de projet dont le but de répondre à un besoin défini dans des délais fixés et d'une certaine qualité, un suivi de travail et une communication entre les membres de projet sont également nécessaires.

Le développement d'un environnement spécifique qui répond aux besoins cités, garantit une méthode de travail collaborative et aide à coordonner et harmoniser les diverses tâches exécutées en toute sécurité demeure une nécessité afin de bien mener le travail.

Dans le cadre de ce projet de fin d'études, notre application a pour principale mission d'aplanir le suivi des projets et d'offrir une représentation visuelle de l'ensemble des informations nécessaires pour la gestion de ces projets.

Une démarche qui vise à organiser de bout en bout le bon déroulement d'un projet et le calcul des indicateurs de performances notamment le plan de la productivité et de l'efficacité va être traitée.

Durant la réalisation de notre projet de fin d'études nous avons eu recours à une méthode de développement qui répondra aux besoins de l'entreprise d'accueil « Billcom Consulting ».

Ce projet consiste à concevoir, réaliser et tester une application de gestion de projet, qui doit être interactive, fiable, conviviale et facile à intégrer dans l'environnement d'une entreprise.

Ce rapport se compose de cinq chapitres qui se présentent comme suit :

Le premier chapitre décrit le cadre général du projet, nous commençons par une présentation de l'organisme d'accueil, l'analyse et la critique de l'existant et nous proposons par la suite, la solution qui va être développée, ainsi que les méthodologies de conception et de développement adoptées. Le second chapitre s'articule sur la phase de planification, nous allons présenter l'équipe du travail, capturer les besoins de la société, analyser et spécifier les besoins pour

pouvoir formuler les fonctionnalités de notre futur système, présenter le choix architectural et la configuration matérielle et logicielle et finalement exposer la planification des releases. Le troisième chapitre est consacré à la gestion d'accès et des utilisateurs, le quatrième chapitre est dédié à la deuxième release et présente la gestion du projet. Pour le dernier chapitre, nous présentons trois sprints, le premier est consacré à la gestion des notifications et des rappels, le deuxième est consacré à la gestion des événements et le dernier sprint est dédié aux tests, déploiement et la mise en place d'une chaîne CI/CD. Le troisième, le quatrième et le cinquième chapitre illustrent le cycle de vie des trois releases en suivant les principes fondamentaux de SCRUM. Nous clôturons par une conclusion générale qui présente un récapitulatif du travail réalisé avec une ouverture sur de nouvelles perspectives.

# **Chapitre 1 : Cadre général du travail**

## **Introduction**

Dans ce chapitre introductif, nous mettons le projet dans son contexte. Nous présentons tout d'abord l'organisme d'accueil. Puis, nous entreprenons avec une étude critique de l'existant qui va nous permettre de proposer notre solution. Enfin nous présentons les objectifs visés et les méthodologies adoptées lors de l'élaboration de ce travail.

### **1. Cadre du projet**

Le présent projet s'intitule « Conception, développement et déploiement d'une application de gestion de projet pour « Billcom Consulting ». Ce travail s'inscrit dans le cadre d'un projet de fin d'études pour l'obtention du diplôme national d'ingénieur en informatique à l'école Nationale Supérieure d'Ingénieurs de Tunis « ENSIT » pour l'année universitaire 2020/2021 au sein de la société « Billcom Consulting » du 5 Avril jusqu'au 15 Septembre.

### **2. Présentation de l'organisme d'accueil**

« Billcom Consulting » est une société spécialisée dans l'intégration de systèmes d'information commercial pour les opérateurs.

Fondée en 2007, elle travaille avec les principaux intégrateurs de systèmes, les aident à fournir des systèmes de pointe à leurs clients tel que Ooreedo, Tunisie Télécom, orange, la poste mobile, etc. Elle a une expérience éprouvée dans l'intégration de divers systèmes et produits bien connus, offrant une grande flexibilité basée sur des normes et réduisant les coûts totaux du projet [1].

« Billcom Consulting » a capitalisé sur son expertise dans le domaine de développement des WebServices, le Customer Relationship Management (CRM) et le Business Intelligence (BI).

Elle offre une large gamme de produits à valeur ajoutée des services informatiques qui comprend :

- La gestion de la relation client (CRM),
- Les services à la clientèle et systèmes de facturation (CCBS),
- L'approvisionnement et médiation,

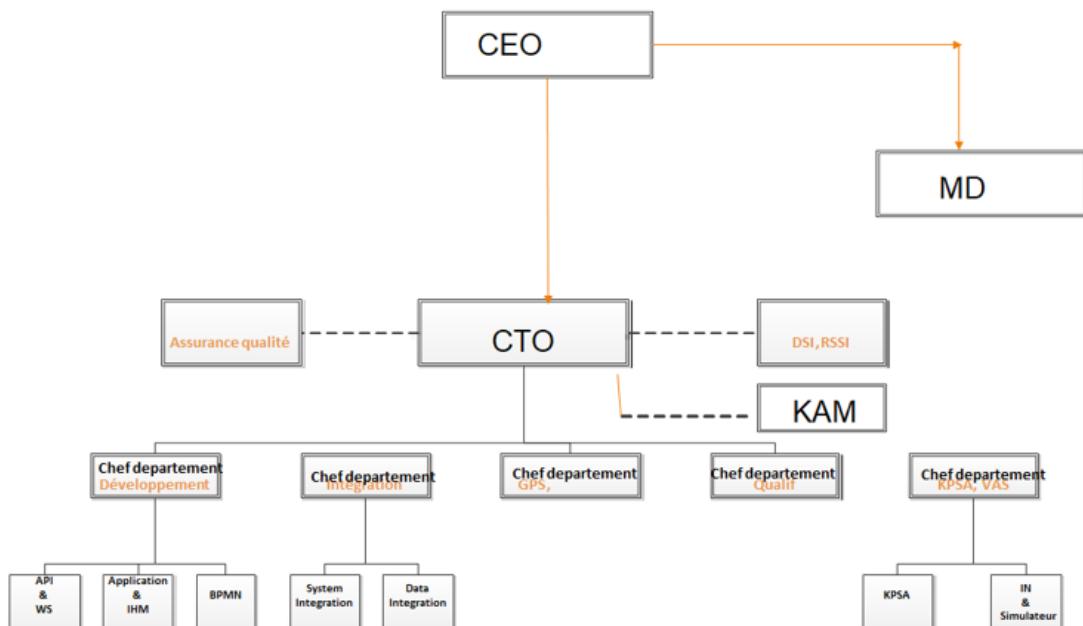
- La Gestion des interconnexions et des fraudes.

La figure 1 présente le logo de « Billcom Consulting ».



**Figure 1 : Logo de « Billcom Consulting » [1]**

« Billcom Consulting » est organisée en fonction des départements et chaque département offre des services spécifiques. La société est structurée comme le montre la figure 2.



**Figure 2 : Organigramme de la société**

Tout du long ce stage, le déroulement du mon projet est effectué dans le département développement.

### 3. Présentation du projet

La gestion des services que fourni « Billcom Consulting » nous a permis d'identifier le volet le plus important sur lequel va se baser notre projet qui consiste à la gestion des projets de la société.

Une gestion de portefeuille de projets, nous a fourni une vue globale sur tous les projets. Cela consiste à suivre l'avancement des projets et gérer tout ce que lui est rattaché pour les finaliser dans les délais et en respectant la charge, tout en mettant en œuvre la bonne organisation du travail, le suivi et la détection des retards de livraison.

En effet, cette gestion offre plusieurs avantages permettant à l'entreprise d'atteindre ses objectifs majeurs tout en assurant une fluidité lors de la réalisation de ces derniers et de lutter contre les problèmes de travail à distance tel que la difficulté de management, le freinage de la communication, la manque de suivi et l'échange des données en toute sécurité.

### **3.1. Etude et critique de l'existant**

Les solutions de gestion de projet existantes sur le marché permettent de planifier et d'optimiser la gestion d'un projet et de suivre son avancement. Ces solutions sont nombreuses et nous allons citer de différents logiciels dans la partie suivante, cependant ces outils ne répondent pas à notre besoin.

#### **3.1.1. Logiciels payants**

Le comparatif des logiciels de gestion de projet en termes de fonctionnalité/prix est présenté par le tableau 1 [2].

**Tableau 1 : Tableau comparatif des logiciels de gestion de projet en termes de fonctionnalité/prix mensuel par utilisateur [2]**

Outils Besoin \	Fitnet Manager	Clarizen	Sciforma	Planzone	Smartsheet
Prix	Prix sur demande	30\$ utilisateur/mois	Prix sur demande	10€ utilisateur/mois	35€ utilisateur/mois
Edition	Premium	Professionnel	Personnalisé	Entreprise	Equipe
Agile Scrum	Oui	Oui	Oui	Oui	Oui
Gantt	Oui	Oui	Non	Oui	Oui
Contrat de travail	Oui	Non	Non	Oui	Oui
Planification de projet	Oui	Non	Oui	Oui	Oui
Portfolio de projet	Non	Oui	Oui	Oui	Oui

Nous nous n'intéresserons pas aux logiciels payants citées dans le tableau 2 puisque la plupart de ces logiciels sont coûteux et la politique de « Billcom Consulting » étant de n'utiliser que des produits gratuits et open source.

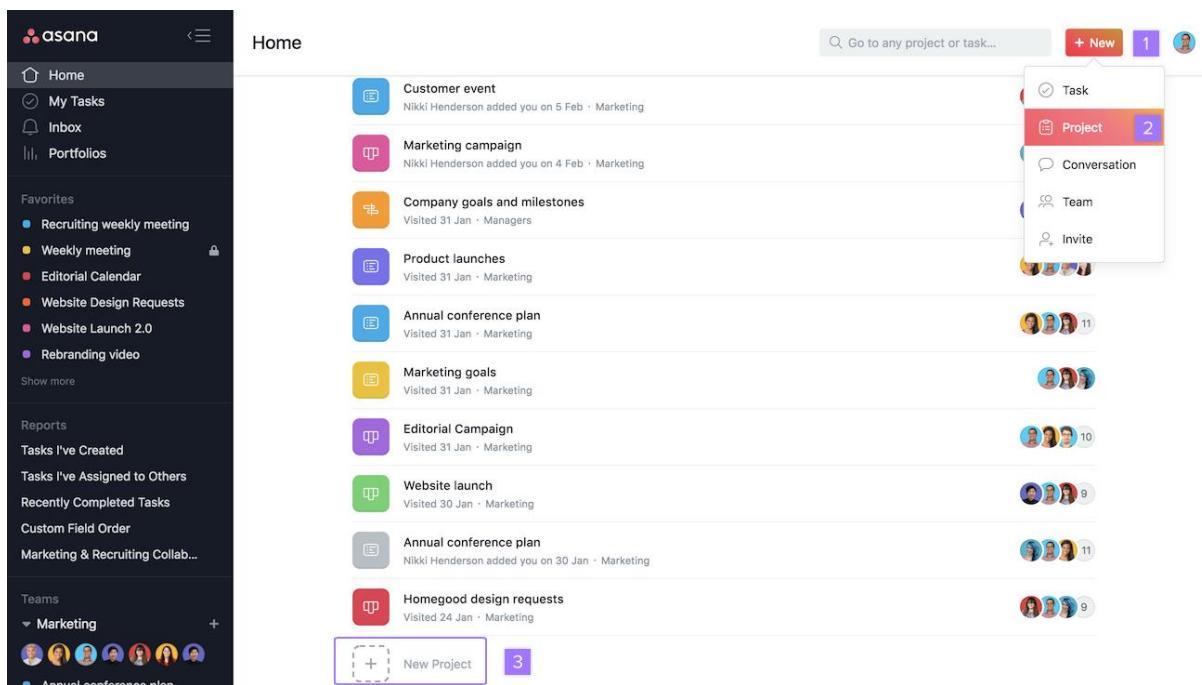
### 3.1.2. Logiciels gratuits

Nous énumérons dans la présente section les logiciels gratuits de gestion du travail.

- **Asana** : C'est un outil de gestion de tâches. Il permet de répartir les tâches à faire entre tous les acteurs de l'entreprise, de collaborer, d'échanger dans le but de travailler plus efficacement. En somme, chacun sait ce qu'il a à faire en arrivant au bureau ou en sortie de réunion : fini les to-do List sur le coin du bureau ou les réunions qui ne débouchent pas sur des actions concrètes.

Asana pose en revanche quelques problèmes d'usage tels que la suppression facile de tâches, l'impossibilité de personnaliser les champs des tâches. Cet outil est donc dédié à la gestion classique de petits projets collaboratifs, mais ne permet pas de gérer des projets complexes [3].

La figure 3 présente l'interface de gestion du projet de l'outil Asana.



**Figure 3 : Interface de gestion de tâches de l'outil Asana [3]**

- **Bitrix24** : C'est un outil de gestion de projet ou en d'autres termes un espace de travail collaboratif. Bitrix24 n'offre pas un service d'administration, de gestion de suivi, ses fonctionnalités sont limitées et il n'offre que la planification des tâches et la discussion entre les membres de l'équipe [3].

La figure 4 présente l'interface de gestion du projet de l'outil Bitrix24.

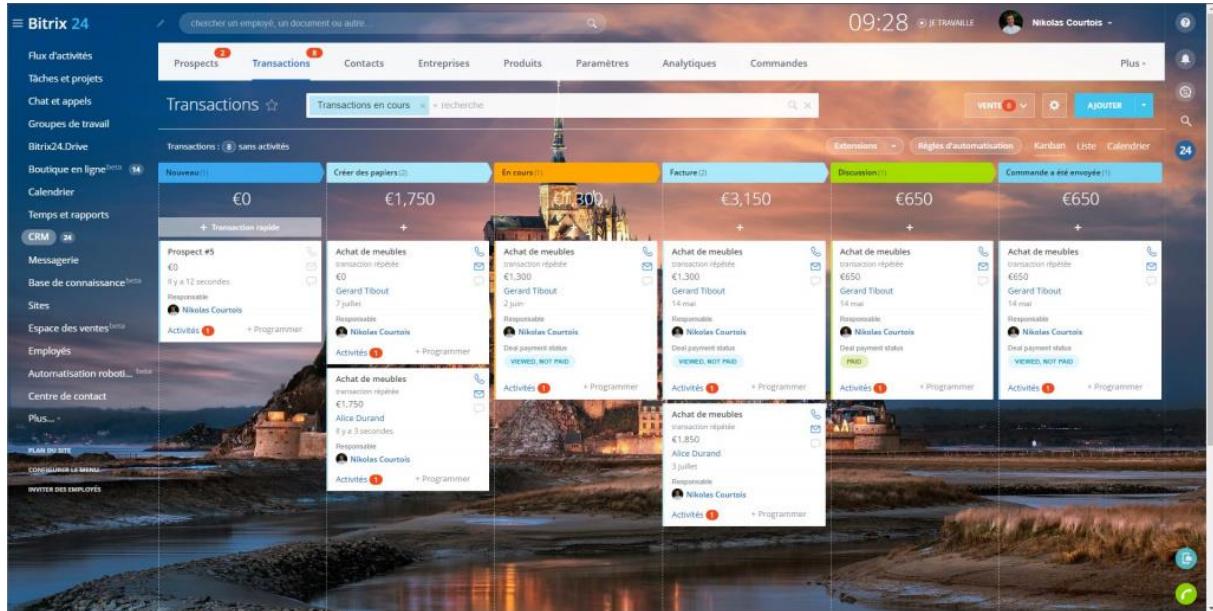


Figure 4 : Dashboard de l'outil Bitrix24 [3]

- **Trello** : Cet outil a compris que la gestion de projet passait avant tout par la collaboration ainsi que la visualisation de l'avancement des tâches en cours. En effet, la solution permet de créer des projets qui prennent la forme de tableaux en plusieurs colonnes (personnalisables). Dans chaque colonne se trouvent les tâches (ou les idées), visualisées sous forme de cartes, que les collaborateurs font progresser de gauche à droite au fur et à mesure de leur avancement.

Trello est entièrement gratuit dans sa première version et dispose d'une application mobile. Cette version gratuite pose un problème de confidentialité et de sécurité des données. L'éditeur avoue lui-même proposer plus de sécurité dans l'offre payante. Il s'apparente plus à un gestionnaire de tâches qu'à un outil de gestion de projet à proprement parlé. En effet, il fait l'impasse sur la gestion des coûts, des temps, des budgets et les fonctions d'administration [3].

La figure 5 présente le logiciel de gestion de projet « Trello ».

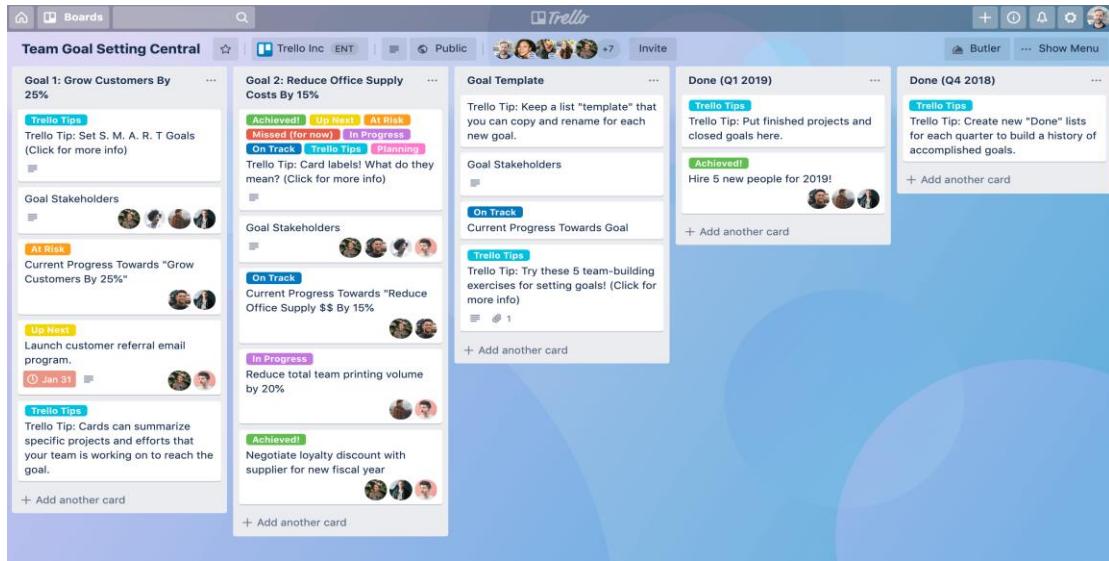


Figure 5 : Interface de gestion de tâches de Trello [3]

- **Jira :** C'est un outil de gestion de projet qu'offre à ses utilisateurs une gamme de fonctionnalités pour gérer leurs projets ainsi que suivre leurs avancements. Ce logiciel peut être adapté aux grands projets avec de grandes équipes. En outre, le tableau de bord proposé n'est pas aussi détaillé qu'aucun le souhaite [3].

La figure 6 présente l'outil de gestion de projet Jira.

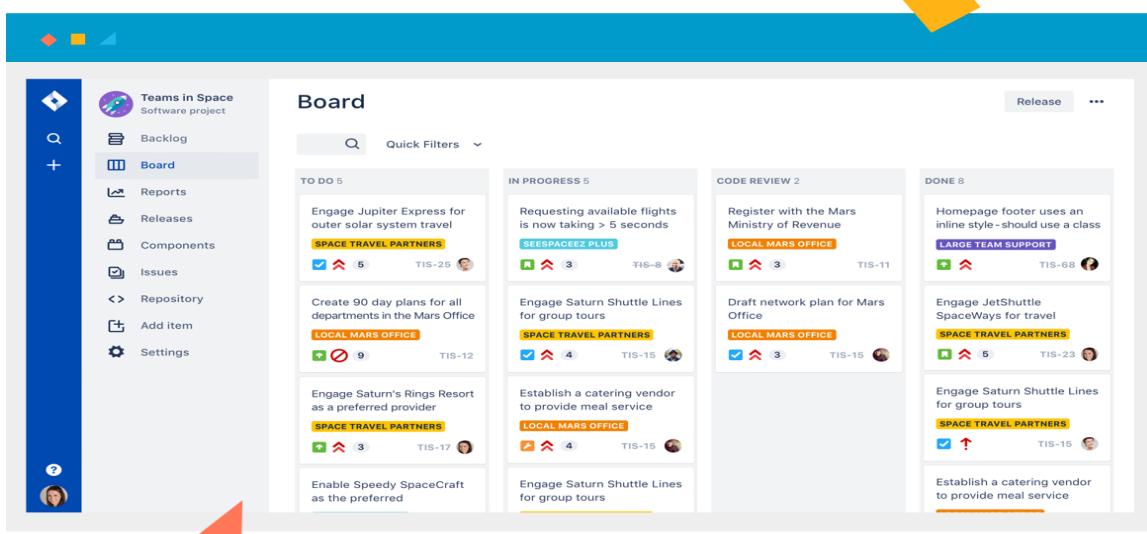


Figure 6 : Interface de gestion de tâches de Jira [3]

Malgré la variété des outils de gestion de projets, ils ne satisfont pas les besoins spécifiques de la société « Billcom Consulting », nous avons traité cette insuffisance par logiciel.

### **3.3. Problématique**

La gestion du portfolio du projet représente un défi pour « Billcom Consulting ». Ce portfolio se trouve face à une augmentation du nombre de projets, et surtout ceux qui sont démarrés en parallèle. En revanche, « Billcom Consulting » ne possède pas un logiciel de gestion de projet spécifique et la politique de la société étant de travailler que avec des logiciels gratuits, en contrepartie, les outils gratuits cités font recours à des fonctionnalités spécifiques payantes, malgré leurs richesses nous a vu que la contrepartie de la gratuité est souvent cachée, la gestion de travail par rôle et la mise en place des tableaux de bord pour le suivi du travail ne font pas partie des fonctionnalités qu'offre les logiciels, la sécurité et la confidentialité des données ne sont pas toujours garanties ainsi le nombre d'utilisateurs est limité.

Egalement, la société doit faire face à plusieurs défis : gérer les projets, gérer les équipes, gérer les tâches, gérer les événements des équipes, mettre en place un système de suivi du travail, éviter le déséquilibre des chargements au sein des équipes, échanger les données en toute sécurité, etc.

Pour remédier aux défis cités auparavant, « Billcom Consulting » a besoin de développer un outil de gestion de projets structuré et bien organisé.

### **3.4. Solution proposée**

Dans le cadre de ce projet de fin d'études et en réponse à la demande de la société, nous proposons à concevoir, développer et déployer une application de gestion de projet pour « Billcom Consulting » qui répond aux besoins de l'organisme d'accueil permettant ainsi :

- La gestion d'accès et l'authentification
- La gestion des utilisateurs
- La gestion et la planification des projets
- La gestion des équipes
- La gestion et la planification des tâches
- La gestion des événements
- La gestion des notifications et la communication en messagerie privée
- La gestion des tableaux de bord

## 4. Méthodologie de travail

Dans cette section nous présentons la méthodologie adoptée, le cycle de développement et le formalisme de conception.

### 4.1. Etude comparative des méthodologies agiles

Les méthodes agiles caractérisent un mode de gestion de projets informatiques privilégiant la communication entre toutes les parties prenantes, clients, développeurs, utilisateurs et autres professionnels du projet, la souplesse en cours de réalisation, et la rapidité de livraison.

Le tableau 2 présente une étude comparative des méthodologies agiles [4].

Tableau 2 : Comparaison de différentes méthodes agiles [4]

Méthode	Description	Points forts	Point faibles
Scrum	<ul style="list-style-type: none"><li>Suppression de hiérarchie avec scrum</li><li>Développement Progressif</li><li>• Suivi quotidien du travail</li></ul>	<ul style="list-style-type: none"><li>Entièrement développé et testé pour de courtes itérations.</li><li>• Processus simple.</li><li>• Augmentation de productivité.</li><li>• Responsabilité collective de l'équipe</li></ul>	<ul style="list-style-type: none"><li>Si l'un des membres de l'équipe quitte, il peut avoir un effet inverse sur le développement du projet</li></ul>
RUP(Rational Unified Process)	<ul style="list-style-type: none"><li>Promu par Rational</li><li>Le RUP est une méthodologie et à la fois un outil prêt à l'emploi.</li></ul>	<ul style="list-style-type: none"><li>Itératif</li><li>• Spécifie le dialogue entre différents intervenants du projet : les plannings, les prototypes...</li></ul>	<ul style="list-style-type: none"><li>Coûteux à personnaliser.</li><li>• Trop lourd.</li></ul>
2TUP(Two Track Unified Process)	<ul style="list-style-type: none"><li>S'articule autour de l'architecture.</li><li>• Suit un cycle de développement en Y.</li><li>• Vise des projets de toutes tailles.</li></ul>	<ul style="list-style-type: none"><li>Donne une place à la technologie et à la gestion du risque.</li><li>• Définit les profils des intervenants, les plannings, les livrables.</li></ul>	<ul style="list-style-type: none"><li>• Trop lourd.</li><li>• Exige trop de documentation.</li></ul>

La méthode que nous avons choisie pour ce projet est la méthode agile SCRUM. Cette dernière est la méthode la plus connue des méthodologies agiles. En effet, elle a fait ses preuves dans des environnements de développement de projets qui connaissent des changements rapides et des exigences émergentes.

## **4.2. Méthodologie adoptée : SCRUM**

SCRUM est un cadre de processus agile permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et itérative des produits de la plus grande valeur possible.

C'est un processus empirique : il se base sur l'expérience du terrain. Il s'appuie sur trois piliers qui sont la transparence, l'inspection et l'adaptation. Le principe de la méthodologie SCRUM est de développer un logiciel de manière incrémentale en maintenant une liste transparente des demandes d'évolutions ou de corrections à implémenter. Avec des livraisons très fréquentes, le client reçoit un logiciel fonctionnel à chaque itération. Plus le projet avance, plus le logiciel est complet et possède toujours de plus en plus de fonctionnalités [5].

### **4.2.1. Les rôles SCRUM**

Les rôles SCRUM sont les suivants :

- **Le product Owner (Directeur du produit)** : est le responsable de l'identification des fonctionnalités du produit, de leur traduction en une liste de priorités, du choix de celles qui doivent figurer en haut de la liste pour le prochain sprint, ainsi que de la réorientation et de l'amélioration continues de la liste.
- **Scrum Master (Meneur de l'équipe)** : aide le groupe de produits à apprendre et à appliquer scrum pour créer de la valeur pour l'entreprise. Le Scrum Master fait tout ce qui est en son pouvoir pour aider l'équipe, le propriétaire du produit et la réussite de l'organisation.
- **Développement Team (équipe de développement)** : est un ensemble de personnes travaillant ensemble pour développer et fournir les augmentations de produit demandées. Il comprend des membres inter-fonctionnels capables d'atteindre les objectifs du sprint [5].

### **4.2.2. Les événements SCRUM**

Les événements SCRUM sont les suivants :

- **Sprint** : Un sprint est la période de temps réelle pendant laquelle l'équipe scrum travaille ensemble pour terminer un incrément. Deux semaines est une durée assez typique pour un sprint, bien que certaines équipes trouvent qu'une semaine de plus est nécessaire.
- **Daily SCRUM** : Il s'agit d'une réunion quotidienne super-courte qui se déroule à la même heure (généralement le matin) et dans un lieu où tout reste simple. Le but du daily scrum

est que tous les membres de l'équipe soient sur la même page, alignés sur l'objectif du sprint, et établissent un plan pour les 24 prochaines heures.

- **Sprint Planning :** Le travail à effectuer pendant le sprint en cours est planifié pendant cette réunion par toute l'équipe de développement. Cette réunion est dirigée par le scrum master et est l'occasion pour l'équipe de décider du but du sprint. Des user stories spécifiques sont ensuite ajoutées au sprint à partir du carnet de produit.
- **Sprint review :** À la fin du sprint, l'équipe se réunit pour une session informelle afin de voir une démonstration ou d'inspecter l'incrément. L'équipe de développement présente les éléments du backlog qui sont terminés aux parties prenantes et à leurs coéquipiers pour les commenter [5].

#### 4.2.3. Les artefacts SCRUM

Les artefacts SCRUM sont les suivants :

- **Product Backlog :** est la liste principale des travaux à effectuer, maintenue par le product owner. Il s'agit d'une liste dynamique de fonctionnalités, d'exigences, d'améliorations et de correctifs servant d'input pour le backlog de sprint. Le carnet de produit est constamment réexaminé, redéfini les priorités et mis à jour par le responsable du produit.
- **Sprint Backlog :** est la liste des éléments, des user stories ou des corrections de bugs sélectionnés par l'équipe de développement pour être implémentée dans le cycle de sprint en cours. Avant chaque sprint, lors de la réunion de planification du sprint, l'équipe choisit les éléments sur lesquels elle travaillera pour le sprint à partir du product backlog [5].

### 4.3. Formalisme adopté

Afin de mener efficacement notre cycle de développement, il est nécessaire de modéliser le produit logiciel. Nous avons choisi le formalisme UML (Unified Modeling Language).

Il représente un support de communication performant, qui facilite la représentation et la compréhension de solutions objet :

- Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation des solutions.
- L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.
- Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.

UML nous fournit des diagrammes pour représenter le logiciel à développer, à savoir, son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel [6].

Tout au long du rapport, nous présenterons quelques diagrammes que nous avons jugés utiles et suffisants pour comprendre le projet à savoir les diagrammes de cas d'utilisation, les diagrammes de classes, les diagrammes de séquence, les diagrammes d'activités.

## **Conclusion**

Ce premier chapitre constitue une étape primordiale pour fixer les repères de notre projet. Après avoir présenté l'organisme d'accueil, nous avons déterminé le cadre du projet, les limites de l'existant qui ont engendré ce travail ainsi que la méthodologie à emprunter lors de ce stage. Dans le prochain chapitre, nous allons aborder la phase de planification de notre projet.

# **Chapitre 2 : Phase de Planification**

## **Introduction**

Après avoir défini le cadre général de notre projet, nous allons nous concentrer dans ce chapitre sur l'identification des besoins, nous allons présenter, par la suite, les acteurs de notre système le backlog de produit du projet. Dans une deuxième partie, nous allons présenter le diagramme de cas d'utilisation global, l'architecture de la solution et l'environnement du travail. Finalement, nous allons exposer la planification des sprints.

### **1. Présentation de l'équipe**

Nous présentons dans cette section l'équipe du travail :

- Client : Monsieur Heni Ouehezi est le « Key Account Manager » de « Billcom Consulting »
- Développeurs : Nous sommes deux développeurs au moment de la rédaction de ce rapport, un salarié développeur monsieur Radhouan Ghribi et un stagiaire développeur.
- Le reste de l'équipe sont dans le service consulting.

### **2. Capture des besoins**

La capture des besoins consiste à identifier les acteurs, exposer le backlog produit du projet, ensuite nous allons décrire les besoins fonctionnels et non fonctionnels que les utilisateurs s'attendent à voir par le système.

#### **2.1. Identification des acteurs**

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser un plus ou une amélioration. Notre étude fonctionnelle sera organisée selon les rôles attribués à chaque acteur qui sont en interaction avec notre système.

Le tableau 3 présente les acteurs de notre projet.

**Tableau 3 : Identification des acteurs**

Acteur	Fonctionnalités
L'administrateur	C'est un utilisateur final de l'application. Il va jouer le rôle d'un administrateur et il aura l'accès à toutes les fonctionnalités de notre système tel que La gestion des utilisateurs, la gestion des rôles, la gestion et la planification du projet, des équipes et des tâches, la gestion des notifications et la gestion des tableaux de bord. Dans notre cas, l'administrateur sera monsieur Heni Ouelhezi.
Le chef de projet	C'est l'acteur responsable à l'affectation des équipes aux projets, il est également responsable du suivi global du projet en terme d'avancement, charge et temps.
Le chef d'équipe	C'est l'acteur responsable du suivi de son équipe, ses fonctionnalités consistent à suivre le travail de son équipe, accorder des tâches aux membres d'équipes, ajouter ou supprimer des membres de l'équipe et gérer les événements et les notifications.
Le membre d'équipe	C'est l'acteur responsable à la gestion de ses tâches, de son profil et des notifications.

## 2.2. Backlog de produit

Nous allons présenter dans cette section le backlog du produit qui est considéré comme élément fondamental de la méthodologie SCRUM. Il s'agit d'une liste de tâches priorisées définissant les besoins métiers de l'utilisateur et un outil de travail principal qui se charge de recueillir les besoins auprès des parties prenantes et de les transformer en liste de fonctionnalités prêtes à être développées par l'équipe de développement [7].

Le backlog de produit présenté dans le tableau 4 comprend les champs suivants :

- ID : C'est l'identifiant unique auto-incrémenté des modules.
- Module : C'est le nom clé pour décrire un ensemble de fonctionnalités regroupées.
- User story : C'est la définition précise et claire de la fonctionnalité souhaitée par l'utilisateur.
- Priorité : C'est la priorité de chaque tâche qui est classée comme suit : "Elevée" ou "Moyenne" ou "Faible".

**Tableau 4 : Backlog de produit**

ID	Module	User Story	Priorité
1	Authentification et gestion d'accès	En tant qu'utilisateur de l'application, je veux être authentifié à l'application.	Elevée
		En tant qu'utilisateur qui tente de se connecter pour la première fois à l'application, je dois être redirigé à une interface de modification du mot de passe avant de se connecter.	Moyenne
		En tant qu'utilisateur, je veux récupérer l'accès à l'application avec l'option mot de passe oublié.	Elevée
		En tant qu'administrateur, je veux activer/désactiver un compte utilisateur.	Elevée
2	Gestion des utilisateurs et des profils	En tant qu'administrateur, je veux créer des comptes utilisateurs.	Elevée
		En tant qu'administrateur, je veux consulter les détails des utilisateurs.	Moyenne
		En tant qu'administrateur, je veux chercher les utilisateurs par filtre.	Moyenne
		En tant qu'administrateur, je veux ajouter/supprimer un ou plusieurs rôles pour les utilisateurs.	Elevée
		En tant qu'utilisateur, je veux modifier mon profil.	Moyenne
		En tant qu'utilisateur de l'application, je veux ajouter des compétences(skill/pourcentage) à mon profil.	Moyenne
3	Gestion et planification du projet	En tant qu'administrateur, je veux créer un projet et l'affecter à un chef de projet.	Elevée
		En tant qu'un chef de projet, je veux consulter les projets qui m'ont été assignés.	Elevée
		En tant qu'un chef projet, je veux assigner des équipes à un projet en appliquant des recherches filtrées sur les utilisateurs ayant le rôle membre/chef d'équipe .	Elevée
		En tant qu'administrateur/chef projet, je veux modifier le statut d'un projet (stopped/paused/pending/processing).	Moyenne

		En tant qu'administrateur, je veux consulter la liste des projets et leurs états d'avancement.	Elevée
		En tant que chef de projet, je veux consulter la liste de mes projets et leurs états d'avancement.	Elevée
		En tant qu'administrateur/chef projet, je veux consulter et modifier les détails du projet.	Elevée
4	Gestion des équipes	En tant que chef d'équipe/chef de projet/administrateur, je veux ajouter des membres à l'équipe en appliquant des recherches filtrées.	Elevée
		En tant que chef d'équipe/chef de projet/administrateur, je veux consulter les détails de chaque équipe.	Moyenne
		En tant que chef d'équipe/chef de projet/administrateur, je veux supprimer des membres de l'équipe.	Moyenne
5	Gestion et planification des tâches	En tant qu'un membre/chef d'équipe, je souhaite ajouter une tâche au tableau.	Elevée
		En tant que chef d'équipe, je veux supprimer une tâches assignée à un membre de l'équipe.	Elevée
		En tant que membre, je veux consulter la liste de mes tâches.	Moyenne
		En tant que membre, je veux modifier l'état d'avancement d'une tâche (to-do/doing/test/done).	Moyenne
		En tant qu'utilisateur de l'application, je veux consulter les détails de ma tâche, télécharger et charger des fichiers.	Moyenne
		En tant que membre/chef d'équipe, je veux ajouter/supprimer des commentaires sur chaque tâche.	Faible
		En tant qu'administrateur/chef de projet/chef d'équipe, je veux consulter les tâches d'un membre et leurs états d'avancement entre deux dates.	Moyenne
		A la création d'une tâche, en tant que chef d'équipe, je veux consulter les tâches qui sont en cours dans cette période et je veux modifier la date de livraison d'une autre tâche.	Elevée
		En tant que chef d'équipe, je veux créer un évènement (réunion/activité...).	Elevée
		En tant que chef d'équipe/membre d'équipe, je veux consulter le calendrier des évènements.	Moyenne

6	Gestion des événements, notifications, rappels et messagerie en temps réels	En tant que chef d'équipe, je veux modifier le jour de l'évènement en utilisant la technique glisser/déplacer.	Elevée
		En tant que chef d'équipe/membre, je veux consulter les détails d'un évènement.	Faible
		En tant que chef d'équipe, je veux modifier les détails d'un évènement.	Elevée
		En tant que chef d'équipe/membre, je veux annuler un évènement.	Elevée
		En tant que membre, je souhaite recevoir des notifications sur mon email en cas d'ajout/de modification d'un évènement.	Faible
		En tant que chef projet, je veux recevoir des notifications relatives à mes projets.	Elevée
		En tant que chef d'équipe, je veux recevoir des notifications relatives à mes équipes.	Elevée
		En tant que membre, je veux recevoir des notifications relatives à mon équipe.	Elevée
		En tant que membre, je veux recevoir des notifications relatives à mes tâches.	Elevée
		En tant que membre, je veux recevoir un rappel de livraison de tâche le jour prévu pour sa livraison.	Elevée
		En tant que membre de l'équipe, je veux recevoir un rappel des événements.	Elevée
		En tant que membre, je veux recevoir un rappel de livraison de tâche.	Elevée
		En tant que chef de projet/chef d'équipe/membre, je veux marquer une notification comme lue/non lue.	Faible
		En tant que membre, je veux envoyer des messages privés en temps réel à un autre membre de l'équipe.	Moyenne
		En tant qu'administrateur, je veux consulter le nombre de projets avec le statut « completed », le nombre de projets « paused », le nombre de projets « stopped » et le nombre de projet « pending ».	Elevée
		En tant qu'administrateur, je veux consulter le nombre de jours gagnés/perdus par rapport à la date de livraison pour chaque projet ayant le statut completed.	Elevée

		En tant qu'administrateur, je veux consulter le nombre de jours gagnés/perdus en totale pour les projets achevés, et le projet ayant le nombre maximal des jours gagnés.	Faible
		En tant qu'administrateur, je veux consulter les projets en retard de livraison et le nombre de jours en retard.	Elevée
		En tant qu'administrateur/chef de projet, je veux choisir un projet en cours et consulter l'état de son avancement, le nombre de jours utilisés et le nombre de jours restants.	Elevée
		En tenant compte de sa charge, en tant qu'administrateur/chef de projet, je veux consulter le temps minimal de la réalisation du projet, le temps optimal, le nombre de ressources nécessaires pour sa réalisation en mois/hommes.	Elevée
		En tant qu'administrateur/chef de projet/chef d'équipe, je veux consulter un tableau de bord représentant le travail pour chaque semaine du mois en cours.	Elevée
7	Gestion des tableaux de bord	En tant qu'administrateur/chef de projet/chef d'équipe, je veux chercher les statistiques du travail de chaque semaine d'un mois donnée.	Elevée
		En tant qu'administrateur/chef de projet/chef d'équipe, je veux consulter le rapport journalier (taux de retard/taux d'avancement) de l'équipe.	Moyenne
		En tant que chef d'équipe/chef de projet/administrateur, je veux consulter les statistiques d'avancement de chaque membre.	Elevée
		En tant que chef d'équipe, je veux consulter les statistiques d'avancement de mes équipes.	Elevée
		En tant que chef d'équipe/chef de projet/administrateur, je veux consulter les tâches en avance/en retard de chaque membre et le nombre de jours en avance/retardés.	Moyenne
		En tant que membre, je veux consulter mon état d'avancement pour chaque équipe dont je fais partie.	Moyenne
		En tant que membre, je veux consulter mon tableau de bord.	Elevée
		En tant que chef d'équipe/membre, je veux chercher les statistiques d'une équipe donnée.	Moyenne
		En tant qu'administrateur/chef de projet/chef d'équipe/membre, je veux télécharger le tableau de bord.	Faible

	Test et déploiement de l'application	En tant que développeur, je dois développer des tests sélénum pour certaines fonctionnalités.	Moyenne
8		En tant que développeur, je dois mettre en place l'intégration continue de l'application.	Moyenne
		En tant que développeur, je dois mettre en place une chaîne de déploiement continu.	Moyenne

### 2.3. Besoins fonctionnels

Les besoins fonctionnels sont les principales exigences que doit fournir notre application à ses utilisateurs, nous allons citer dans ce qui suit l'ensemble des fonctionnalités du système.

**Gérer les utilisateurs :** Le système doit permettre à l'administrateur (le manager) de gérer les utilisateurs :

- Ajouter un utilisateur
- Modifier les détails d'un utilisateur
- Consulter la liste des utilisateurs
- Rechercher les utilisateurs par plusieurs filtres
- Activer/désactiver un utilisateur

**Gérer les rôles :** Le système doit permettre à l'administrateur (le manager) de gérer les rôles :

- Ajouter un ou plusieurs rôles à un utilisateur
- Supprimer un ou plusieurs rôle d'un utilisateur
- Consulter la liste des rôles d'un utilisateur

**Gérer son profil :** Le système doit permettre à l'utilisateur de l'application de gérer son profil :

- Consulter son profil
- Modifier ses informations
- Ajouter des compétences
- Ajouter des fichiers

**Gérer et planifier les projets:** Le système doit permettre à l'administrateur/chef de projet de gérer les projets :

- Ajouter un projet

- Assigner un projet à un chef de projet
- Consulter la liste des projets
- Consulter les détails de chaque projet
- Modifier les informations d'un projet
- Rechercher les projets par plusieurs filtres
- Modifier le statut d'un projet

**Gérer les équipes :** Le système doit permettre à l'administrateur/chef de projet/chef d'équipe de gérer les équipes :

- Ajouter une équipe
- Modifier une équipe
- Consulter les détails de l'équipe
- Supprimer une équipe

**Gérer et planifier les tâches :** Le système doit permettre aux utilisateurs de gérer les tâches :

- Ajouter une tâche
- Accorder une tâche à un utilisateur
- Modifier l'état d'avancement d'une tâche
- Ajouter des commentaires à la tâche
- Afficher les détails de la tâche
- Supprimer une tâche

**Gérer les événements :** Le système doit permettre au chef d'équipe de gérer les événements :

- Ajouter un événement
- Consulter le calendrier de l'évènement
- Modifier un événement
- Supprimer un événement

**Gérer les notifications :** Le système doit permettre aux utilisateurs de gérer les notifications :

- Recevoir des notifications en temps réel

- Modifier le statut des notifications
- Supprimer les notifications
- Recevoir des rappels de livraison de tâche ou d'évènement.
- Envoyer des messages en temps réel

**Gérer les tableaux de bord:** Le système doit permettre aux utilisateurs de gérer les statistiques et le calcul des kpi (key performance indicators) :

- Consulter les tableaux de bord spécifiques au projet
- Consulter les tableaux de bord spécifiques à l'équipe
- Consulter le tableau de bord spécifique au membre de l'équipe
- Chercher des statistiques d'une équipe/projet/période donnée
- Exporter les tableaux de bord

## 2.4. Besoins non fonctionnels

Afin d'assurer un bon fonctionnement de l'application et de garantir la satisfaction de l'utilisateur, des contraintes techniques et ergonomiques doivent être prises en compte tout au long du développement du projet :

- **La maintenabilité** : Le code doit être modulaire, commenté et doit respecter les règles standards et universelles de codage.
- **L'ergonomie et la convivialité** : L'application doit fournir des interfaces simples pour l'utilisateur afin de faciliter l'exploitation et la manipulation des services.
- **L'évolutivité** : Le code doit être extensible et capable d'accueillir les modifications susceptibles d'être ajoutées selon les besoins.
- **La sécurité** : L'accès à l'application ainsi qu'aux données doit être sécurisé. Elle doit donc tenir compte de confidentialité des données des utilisateurs en cryptant tous les mots de passe de leurs comptes, au niveau de la base de données, pour éviter tout accès non autorisé. Il faut aussi sécuriser l'action d'authentification en utilisant l'approche de connexion à base d'échange des jetons (tokens)
- **Le temps réel** : Les fonctionnalités de notifications et messagerie de l'application doivent être gérées en temps réel.

### 3. Analyse et spécifications des besoins

Dans cette partie nous allons présenter le diagramme de cas d'utilisation global et l'architecture globale du projet.

#### 3.1. Diagramme de cas d'utilisation global

Les diagrammes de cas d'utilisation ont pour rôles de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape de modélisation UML pour la conception d'un système. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur et un système. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases) [8].

La figure 7 présente le diagramme de cas d'utilisation global.

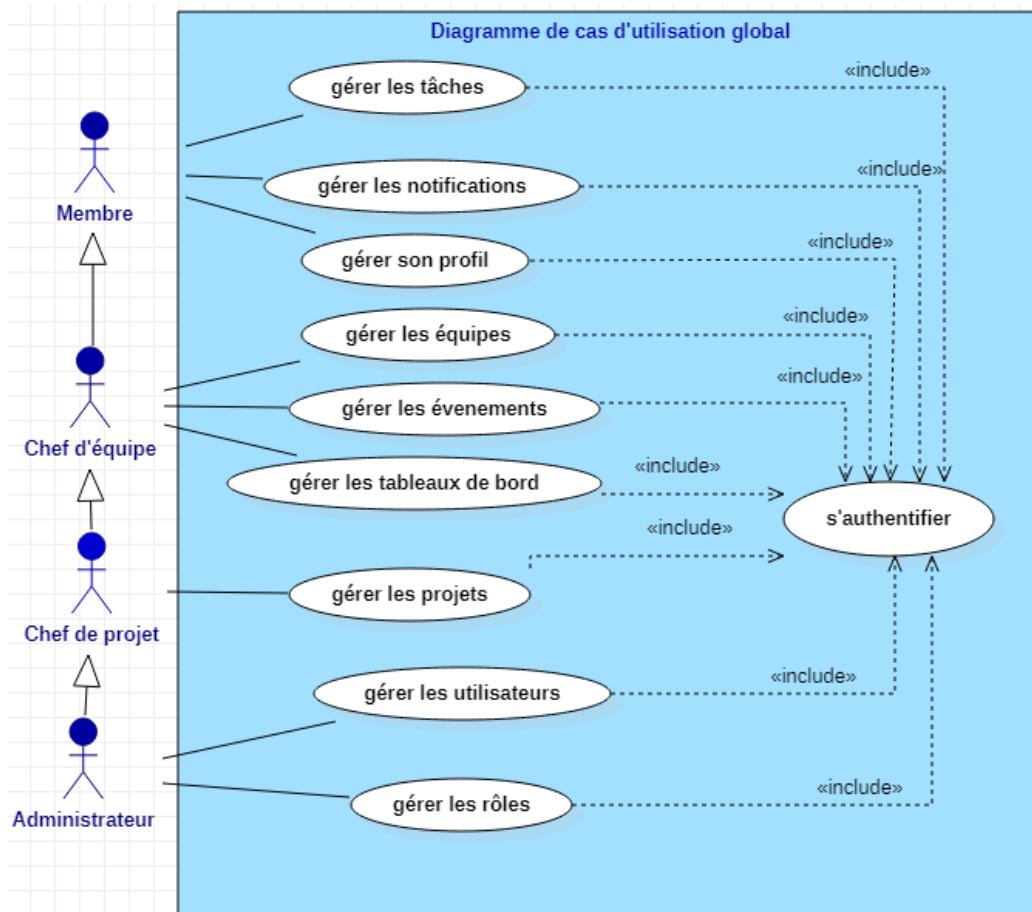


Figure 7 : Diagramme de cas d'utilisation global

## 3.2. Architecture de la solution

Pour chaque système informatique, nous avons besoin de choisir l'architecture de la solution adéquate pour sa réalisation et qui peut assurer un bon fonctionnement et une haute performance. Nous allons présenter dans cette section l'architecture physique et l'architecture logique de ce système.

### 3.2.1. Architecture physique

Dans la présente section, nous allons détailler l'architecture physique de notre application, qui présente l'ensemble des composants matériels que supportent l'application, nous allons adopter une architecture à trois niveaux.

Les trois niveaux de l'architecture sont :

- **Un client léger** : C'est le navigateur web permettant à l'utilisateur d'accéder au site via internet.
- **Un middle tiers** : C'est le serveur d'application qui héberge toutes les couches de l'application à développer.
- **Un tiers de données** : C'est le serveur qui met les données à la disposition des opérateurs d'obsèques tout en assurant des droits accordés à ces derniers [9].

La figure 8 présente l'architecture physique du système.

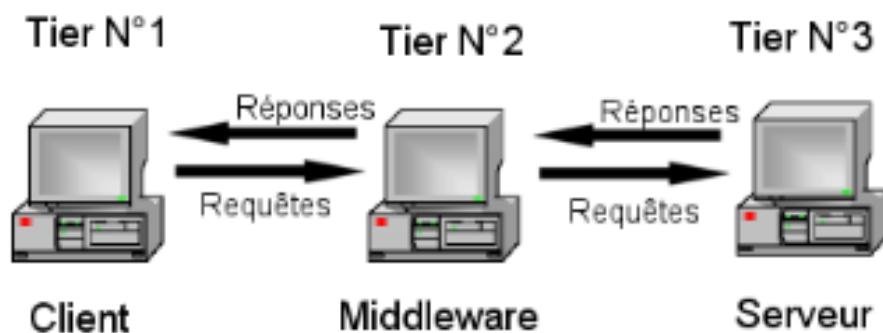


Figure 8 : Architecture physique du système [9]

### 3.2.2. Architecture logique

Par complémentarité à l'architecture physique qui permet de distinguer les différents niveaux physiques de l'application, l'architecture logique s'intéresse au découpage logique de l'application et la façon de regrouper les composants selon les traitements qu'ils effectuent.

### 3.2.2.1. Modèle architectural du Front-End

Pour la partie web du Front-End, nous utilisons Angular8 qui est doté d'une architecture robuste et évolutive.

Cette architecture est basée sur le composant et son modèle associé. Voici ses différents éléments :

- **Modules** : Un module déclare un contexte de compilation pour un ensemble de composants dédiés à un domaine d'application.
- **Components** : Les composants sont essentiellement des classes qui interagissent avec le fichier .html du composant dans le but de définir une vue à afficher dans le navigateur selon un style d'affichage.
- **Template** : Un template est un modèle qui combine du html et du balisage angular qui peut influencer sur la vue avant l'affichage de cette dernière. Une communication entre un composant et son modèle est possible grâce à la liaison de données (data binding).
- **Service** : C'est une classe dédiée à l'injection de dépendance, elle est alors destinée à exécuter les opérations, dont les données, ou la logique n'est pas associée à un modèle spécifique et qui est principalement partagé entre plusieurs composants.
- **Routing (ou routage)** : Le routage permet de définir un chemin de navigation [10].

La figure 9 présente l'architecture du front-end du projet.

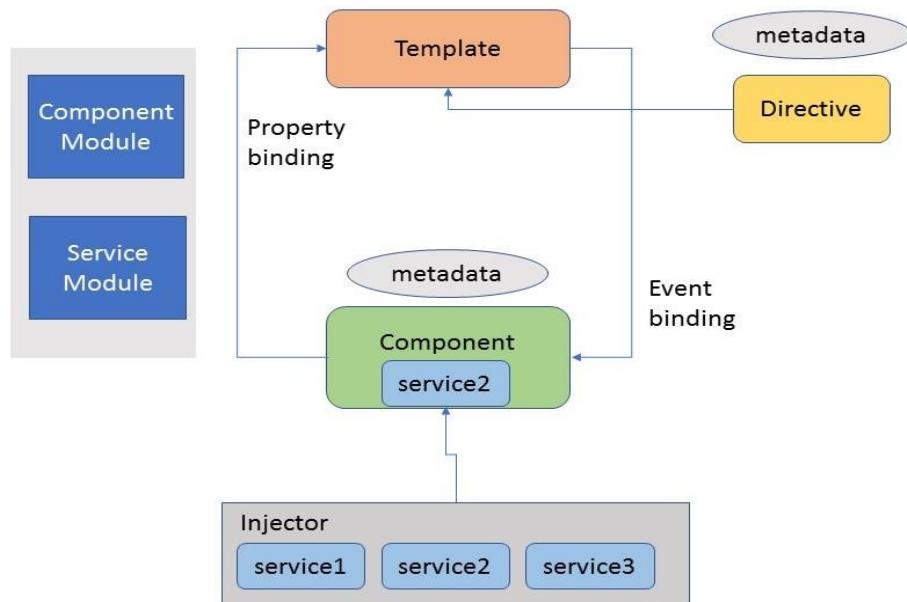


Figure 9 : Architecture Front-End [10]

### 3.2.2.2. Modèle architectural du Back-End

La conception de notre partie backend est modélisée comme suit dans le diagramme de package. Ce diagramme illustre une représentation graphique de l'organisation de l'application qui identifie les liens de dépendance entre les paquets de l'application (figure 10).

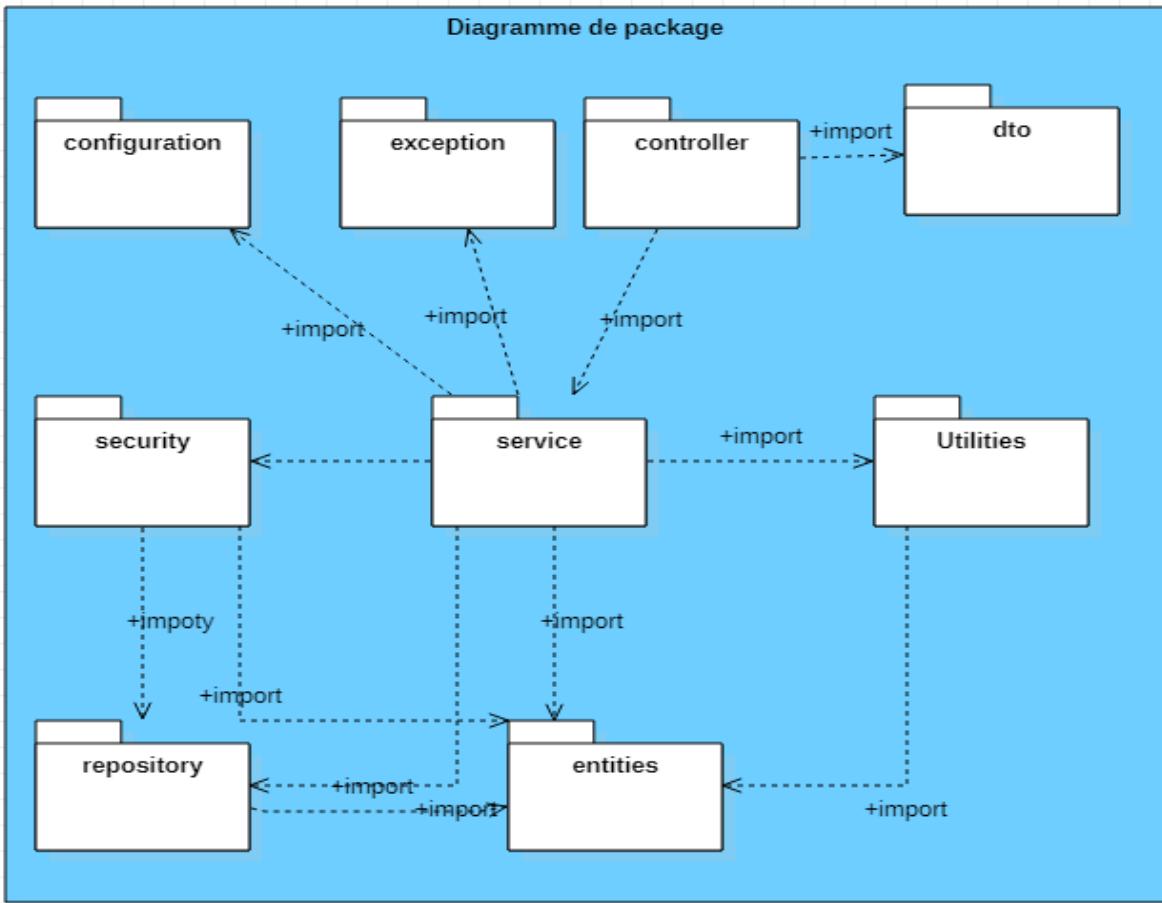


Figure 10 : Diagramme de package

Les packages utilisés pour le développement de notre application sont les suivants :

- **Entities** : C'est le package qui contient les classes entités de l'application.
- **DTO** : C'est le package qui contient les classes qui vont être utilisées pour encapsuler les données et les envoyer vers l'application Front-end, et aussi comme types de retour.
- **Repository** : C'est le répertoire des interfaces héritant de l'interface JpaRepository. L'objectif de ces interfaces consiste à rendre la création de la couche d'accès aux données plus rapide et fluide.
- **Service** : C'est l'ensemble des classes qui implémentent la logique métier de l'application.
- **Controller** : C'est le package qui contient les contrôleurs Rest.

- **Security** : C'est le package où nous avons implémenté le module de sécurité de l'application.
- **Exceptions** : C'est le package qui contient les déclarations des exceptions des modules.
- **Configuration** : C'est le package qui contient les classes de configuration qui doivent être lancés dès le lancement du projet.
- **Utilities** : C'est le package où se fait l'intégration des apis et fonctionnalités externes tels que l'Email et autres.

## 4. Environnement du travail

Cette section est destinée à la présentation de l'environnement de travail qui consiste à présenter l'environnement matériel et logiciel utilisés pour l'implémentation de notre application.

### 4.1. Environnement matériel

Ce projet a été réalisé en utilisant un ordinateur ayant les caractéristiques décrites dans le tableau 5.

**Tableau 5 : Environnement Matériel**

<b>Caractéristique</b>	PC
<b>Processeur</b>	Intel CORE I5
<b>Ram</b>	8GO
<b>Système d'exploitation</b>	Windows 10 professionnel

### 4.2. Environnement logiciel

Dans cette section, nous allons présenter les outils de développement et de modélisation, les langages de programmations et les technologies utilisés.

#### 4.2.1. Outils de développement et de modélisation

Les logiciels que nous avons utilisés pour la réalisation de ce projet sont représentés par le tableau 6.

**Tableau 6 : Outils de développement et modélisation**

Logiciel	Description
<b>Spring tools Suite (STS)</b> 	Spring Tool Suite est une application basée sur Eclipse facilitant la création de projet Spring [11].

<b>Visual Studio Code</b> 	Visual Studio Code un éditeur de code léger mais puissant qui fonctionne sur toutes les plateformes. Il prend en charge la plupart des langages connus comme JavaScript, TypeScript, Python [12].
<b>PostgreSQL</b> 	PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD [13].
<b>StarUML</b> 	StarUML est un logiciel de modélisation UML, qui a été "cédé comme open source" par son éditeur, à la fin de son exploitation commerciale (qui visiblement continue ...), sous une licence modifiée de GNU GPL [14].
<b>PgAdmin</b> 	pgAdmin est un outil de gestion open source pour Postgres. pgAdmin fournit une interface graphique puissante qui simplifie la création, la maintenance et l'utilisation des objets de base de données [15].

#### 4.2.2. Outils de test

Le test du logiciel fait partie du cycle de vie du développement, son objectif est de s'assurer que le code à déployer est de qualité, sans bugs ni erreurs logiques. Afin de faire les tests de l'application nous utilisons les outils dans le tableau 7.

Tableau 7 : Outils de test

Logiciel	Description
<b>SwaggerUI</b> 	C'est une plateforme qui offre des outils permettant de générer la documentation pour son API Web. Il offre également une interface permettant d'explorer et tester les différentes méthodes offertes par le service [16].
<b>Junit 5</b> 	C'est un framework de test unitaire, "Junit" qui est un framework open source pour le développement et l'exécution de tests unitaires avec le langage Java [17].
<b>Selenium</b> 	Selenium WebDriver est un framework web qui vous permet d'exécuter des tests multi-navigateurs. Cet outil est utilisé pour automatiser les tests d'applications Web pour vérifier qu'ils fonctionnent correctement [18].

#### 4.2.3. Langages de programmation

Le tableau 8 présente les langages de programmation pour la partie backend et front-end de l'application.

Tableau 8 : Langages de programmation

Langage	Description
<b>Java</b> 	Java est un langage de programmation orienté objet et une plateforme informatique qui ont été créées par Sun Microsystems en 1995.une plate-forme informatique qui ont été créés par Sun Microsystems en 1995 [19].

 <b>TypeScript</b>	C'est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript [20].
--	--

#### 4.2.4. Technologies utilisées

Le tableau 9 illustre les technologies utilisés pour le développement de l'application.

**Tableau 9 : Technologies utilisées**

<b>Technologies</b>	<b>Description</b>
 <b>Spring Boot 2</b>	Pour le développement du backend nous avons opté pour Spring Boot 2. C'est un Framework java créé par l'équipe pivotal qui permet de simplifier le démarrage et le développement de nouvelles applications spring en réduisant la complexité de configuration [21].
 <b>Angular 8</b>	Angular est un Framework orienté composant qui facilite la création d'une application web. Il permet de créer des applications de type SPA en se basant sur un système de routage agile (sans rafraîchissement de page) [22].
 <b>Chart.js</b>	Chartjs est une bibliothèque de code JavaScript open source simple mais flexible pour les concepteurs et les développeurs. C'est un outil JavaScript de représentation des données sous forme de graphes statistiques [23].

#### 4.2.5. Outils de déploiement

Pour déployer notre application nous avons utilisés les outils décrits dans le tableau 10.

**Tableau 10 : Outils de déploiement**

<b>Outil</b>	<b>Description</b>
 <b>Github</b>	GitHub est une plateforme d'hébergement de code pour le contrôle de version et la collaboration. Il permet de travailler sur des projets de n'importe où avec une équipe distante [24].
 <b>Heroku</b>	Heroku est une plate-forme cloud en tant que service (PaaS) basée sur des conteneurs. Les développeurs utilisent Heroku pour déployer, gérer et faire évoluer des applications modernes. Notre plate-forme est élégante, flexible et facile à utiliser[25].
 <b>Jenkins</b>	Jenkins est un outil open source de serveur d'automatisation. Il aide à automatiser les parties du développement logiciel liées au build, aux tests et au déploiement, et facilite l'intégration continue et la livraison continue [26].

## 5. Planification des sprints

Un sprint est une itération de quelques semaines dans laquelle nous travaillons à produire un incrément du produit potentiellement livrable. Après avoir fixé le backlog du produit, nous répartissons l'ensemble des « stories » sur trois releases.

Nous allons commencer par un tableau descriptif du déroulement général du stage (tableau 11) :

**Tableau 11 : Déroulement de stage**

	5 Avril	5 Mai	5 juin	5 Juillet	5 Aout	15 Septembre
Formation et conception	*					
Développement	*	*	*	*	*	
Test					*	*
Déploiement						*

Nous présentons ensuite les releases à développer tout au long de notre travail par le tableau 12.

**Tableau 12 : Planification des releases**

<b>Id</b>	<b>Release</b>	<b>Date de début</b>	<b>Date de fin</b>
1	Gestion d'accès et des utilisateurs	20/04/2021	20/05/2021
2	Gestion et planification des projets	21/05/2021	01/07/2021
3	Gestion des événements, statistiques et déploiement	01/07/2021	15/09/2021

## Conclusion

Ce chapitre nous a permis de déterminer les acteurs de notre application ainsi que les différents besoins fonctionnels et non fonctionnels, le diagramme de cas d'utilisation et le backlog de produit nous a permis de dégager les différentes fonctionnalités de manière détaillée.

Par la suite, nous avons présenté l'architecture physique et l'architecture logicielle de notre projet et nous avons clôturé ce chapitre par l'exposition de l'environnement logiciel et matériel. Le prochain chapitre sera dédié à la release 1 « Gestion d'accès et des utilisateurs ».

# **Chapitre 3 : Release 1 « Gestion d'accès et des utilisateurs »**

## **Introduction**

Après avoir analysé et spécifié les besoins globaux de notre client, nous détaillerons, dans ce chapitre, les différentes étapes effectuées pour le développement des deux sprints de la première release.

Nous commencerons, tout d'abord, par la présentation du backlog de produit de chaque sprint suivi d'une analyse détaillée, une conception des fonctionnalités et finalement une présentation des interfaces homme-machine réalisées.

### **1. Présentation de la release 1**

Une réunion avec l'équipe scrum s'est effectuée afin de spécifier les fonctionnalités que doit satisfaire cette release.

Notre première release intitulée « Gestion d'accès et des utilisateurs » comportera deux sprints qui se présentent comme suit :

- Sprint 1 : Authentification et gestion d'accès
- Sprint 2 : Gestion des utilisateurs et de leurs profils

Pour chaque sprint nous allons présenter son backlog de produit ainsi qu'une analyse va être explorée en présentant des diagrammes de cas d'utilisation raffinés et une description textuelle de quelques cas d'utilisation.

A la fin de cette release, nous devons avoir notre premier livrable pour notre client, les utilisateurs de l'application peuvent s'authentifier, l'administrateur de l'application (le manger) aura la possibilité de gérer l'accès et les comptes des utilisateurs et l'utilisateur de l'application aura la possibilité de gérer son profil.

### **2. Sprint 1 : Authentification et gestion d'accès**

Afin de sécuriser notre application, nous devons implémenter une couche de sécurité.

Nous allons présenter dans ce qui suit, le mécanisme de sécurité de notre application, le backlog de produit de ce sprint, son diagramme de cas d'utilisation, son diagramme de classes et finalement des captures écrans des interfaces.

## 2.1. Mécanisme de sécurité

Pour fournir une authentification sécurisée à notre application ainsi qu'un support solide d'autorisation nous avons référé à « Spring Security » qui a été livré avec des algorithmes de sécurité en se basant sur le json web token.

Le JWT est un standard ouvert défini dans la RFC 75191. Il permet l'échange sécurisé de jetons entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique, le jeton est composé de trois parties et chacune contient des informations différentes comme suit :

- **Un Header** : Il identifie l'algorithme qui a été utilisé pour générer la signature, ainsi que le type de jeton, dans notre application nous avons utilisé l'algorithme de hachage HS256.
- **Un payload** : Le payload contient les claims (les informations de l'utilisateurs) que l'on souhaite transmettre.
- **Une signature** : C'est la dernière partie du jeton, et est générée à partir du payload et du header.

Le jeton sera envoyé avec chaque requête que le client fera auprès de l'application, qui autorisera, ou non, le client à accéder à ses services suivant la validité de ce dernier [27].

Le diagramme de séquence présenté en annexe par la figure 98 décrit en détail le séquencement du processus de la couche sécurité en utilisant « Spring Security ».

## 2.2. Backlog de produit du sprint

Nous allons énumérer les différents « User Stories » de ce premier sprint « Authentification et gestion d'accès » dans le backlog de produit présenté par le tableau 13.

Tableau 13 : Backlog de produit du sprint 1

ID Module	Module	ID User Story	User Story
1.	Authentification et gestion d'accès	1.	En tant qu'utilisateur de l'application, je veux être authentifié à l'application via mon adresse e-mail et mon mot de passe
		2.	En tant qu'utilisateur, qui tente de se connecter pour la première fois à l'application, je dois modifier mon mot de passe.
		3.	En tant qu'utilisateur, je veux récupérer l'accès à mon compte avec l'option mot de passe oublié.
		4.	En tant qu'administrateur de l'application, je veux activer/désactiver un compte utilisateur.

## 2.3. Analyse

Les « User Stories » que nous avons spécifié dans le backlog de produit précédent nous permettent de mieux comprendre l'objectif de ce premier sprint.

Dans la section suivante, nous allons modéliser les différentes spécifications et fonctionnalités par des diagrammes de cas d'utilisation et des descriptions textuelles de quelques cas et des.

### 2.3.1. Diagramme de cas d'utilisation

Dans le but de décrire les fonctionnalités de ce sprint d'une manière formelle, nous exposons le diagramme de cas d'utilisation dans la figure 11.

L'acteur utilisateur dans ce diagramme de cas d'utilisation peut être un chef de projet, chef d'équipe ou un membre.

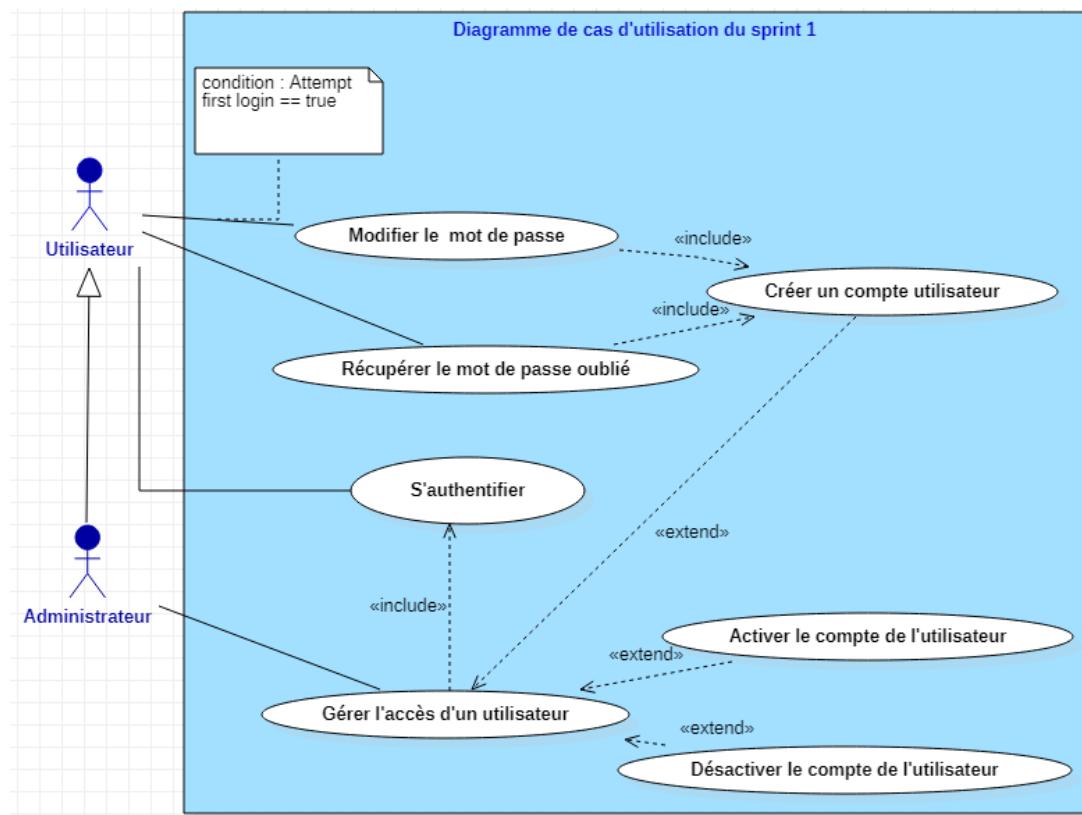


Figure 11 : Diagramme de cas d'utilisation du premier sprint

### 2.3.2. Raffinement des cas d'utilisation

Même si le diagramme de cas d'utilisation donne une représentation simple du système en main et montre les relations entre les acteurs et les cas d'utilisation, nous procédons dans cette partie au raffinement de quelques cas d'utilisation.

### - Description textuelle du cas d'utilisation : Authentification

Le tableau 14 illustre le raffinement du cas d'utilisation « S'authentifier », en présentant l'acteur de la fonctionnalité, les conditions, le scénario nominal et les exceptions de cette fonctionnalité.

**Tableau 14 : Description textuelle du cas d'utilisation « S'authentifier »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	S'authentifier
<b>Acteur</b>	Utilisateur de l'application
<b>Objectif</b>	Accéder à l'application
<b>Résumé</b>	L'utilisateur s'authentifie afin d'accéder à l'application pour gérer ses fonctionnalités selon son rôle (administrateur/chef de projet/chef d'équipe/ membre).
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
<ul style="list-style-type: none"> <li>- L'utilisateur doit avoir un compte utilisateur.</li> <li>- Le compte de l'utilisateur doit avoir le statut actif.</li> <li>- L'utilisateur doit être connecté au moins une fois à l'application.</li> <li>- Le json web token est non expiré.</li> </ul>	L'utilisateur se connecte à l'application.
<b>Scénario</b>	
1- L'utilisateur accède au formulaire d'authentification. 2- L'utilisateur saisit son adresse et son mot de passe dans le formulaire. 3- Le système vérifie le compte depuis la base de données. 4- Le système redirige l'utilisateur vers la page de son profil.	
<b>Exception</b>	
1-Si les informations de connexion (adresse et/ou mot de passe) sont erronées, un message d'erreur s'affiche « veuillez vérifier vos coordonnées ». 2-Si le compte de l'utilisateur est désactivé, un message d'erreur s'affiche. 3- Si l'utilisateur se connecte pour la première fois, une redirection vers la page de modification du mot de passe.	

- **Description textuelle du cas d'utilisation : Modifier le mot de passe**

Le tableau 15 illustre le raffinement du cas d'utilisation « Modifier le mot de passe ».

**Tableau 15 : Description textuelle du cas d'utilisation « Modifier le mot de passe »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Modifier le mot de passe.
<b>Acteur</b>	Un utilisateur de l'application qui tente de se connecter pour la première fois à l'application.
<b>Objectif</b>	Etre authentifié.
<b>Résumé</b>	L'utilisateur modifie son mot de passe pour pouvoir accéder à l'application pour la première fois.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
<ul style="list-style-type: none"> <li>- Un utilisateur tente de se connecter pour la première fois à l'application.</li> <li>- le compte de l'utilisateur doit avoir l'état actif.</li> </ul>	L'utilisateur accède à l'application.
<b>Scénario</b>	
<ol style="list-style-type: none"> <li>1- L'utilisateur accède au formulaire d'authentification.</li> <li>2- L'utilisateur saisit ses informations dans le formulaire.</li> <li>3- Le système redirige l'utilisateur vers la page de modification du mot de passe.</li> <li>4- L'utilisateur modifie son mot de passe.</li> <li>5- Le système redirige l'utilisateur à la page du formulaire d'authentification.</li> <li>6- L'utilisateur saisit ses informations dans le formulaire.</li> <li>7- Le système vérifie les données depuis la base de données.</li> <li>8- Le système redirige l'utilisateur vers la page du profil.</li> </ol>	
<b>Exception</b>	
<ol style="list-style-type: none"> <li>1-Si les deux champs du mot de passe ne sont pas identiques, un message d'erreur s'affiche « les deux champs ne sont pas identiques ».</li> <li>2-Si les informations de connexion (adresse e-mail et/ou mot de passe) sont erronées, un message d'erreur s'affiche « veuillez vérifier vos coordonnées ».</li> <li>3-Si le compte de l'utilisateur est désactivé, un message d'erreur s'affiche. «vous n'avez pas le droit de se connecter à l'application ».</li> </ol>	

- **Description textuelle du cas d'utilisation : Récupérer le mot de passe oublié**

Le tableau 16 illustre le raffinement du cas d'utilisation « Récupérer le mot de passe oublié », en présentant l'acteur de la fonctionnalité, les conditions, le scénario nominal et les exceptions de cette fonctionnalité.

**Tableau 16 : Description textuelle du cas d'utilisation « Récupérer le mot de passe oublié »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Récupérer mot de passe oublié
<b>Acteur</b>	Utilisateur de l'application (administrateur/chef de projet/chef d'équipe/membre).
<b>Objectif</b>	Changer le mot de passe et récupérer l'accès à l'application.
<b>Résumé</b>	L'utilisateur change son mot de passe afin d'accéder à l'application pour gérer ses fonctionnalités selon son rôle(administrateur /chef d'équipe/membre).
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
- le compte de l'utilisateur doit avoir le statut actif.	L'accès à l'application est récupéré.
<b>Scénario</b>	
1- Le système affiche le formulaire d'authentification. 2- L'utilisateur choisit l'option mot de passe oublié. 3- Le système affiche un nouveau formulaire. 4- L'utilisateur entre son adresse e-mail. 5- Le système envoie un code à l'adresse e-mail et indique à l'utilisateur de vérifier sa boîte email. 6- L'utilisateur saisit le code reçu par email dans le formulaire du mot de passe oublié. 7- Le système vérifie le code et redirige l'utilisateur vers le tableau de bord.	
<b>Exception</b>	
1-L'utilisateur saisit un code erroné « vous avez saisi un code erroné ». 2-L'utilisateur saisit une adresse e-mail qui n'existe pas dans la base de données, un message d'erreur s'affiche « Vous avez saisi un e-mail non valide ».	

### 2.3.3. Diagramme de séquence

Les diagrammes de séquence offrent une description des scénarios des cas d'utilisation en mettant l'accent sur la chronologie des interactions entre les acteurs et le système [28].

Dans cette partie, nous allons exposer quelques scénarios dans des diagrammes de séquence.

#### - Diagramme de séquence du scénario s'authentifier

La figure 12 présente le diagramme de séquence système du cas d'utilisation s'authentifier.

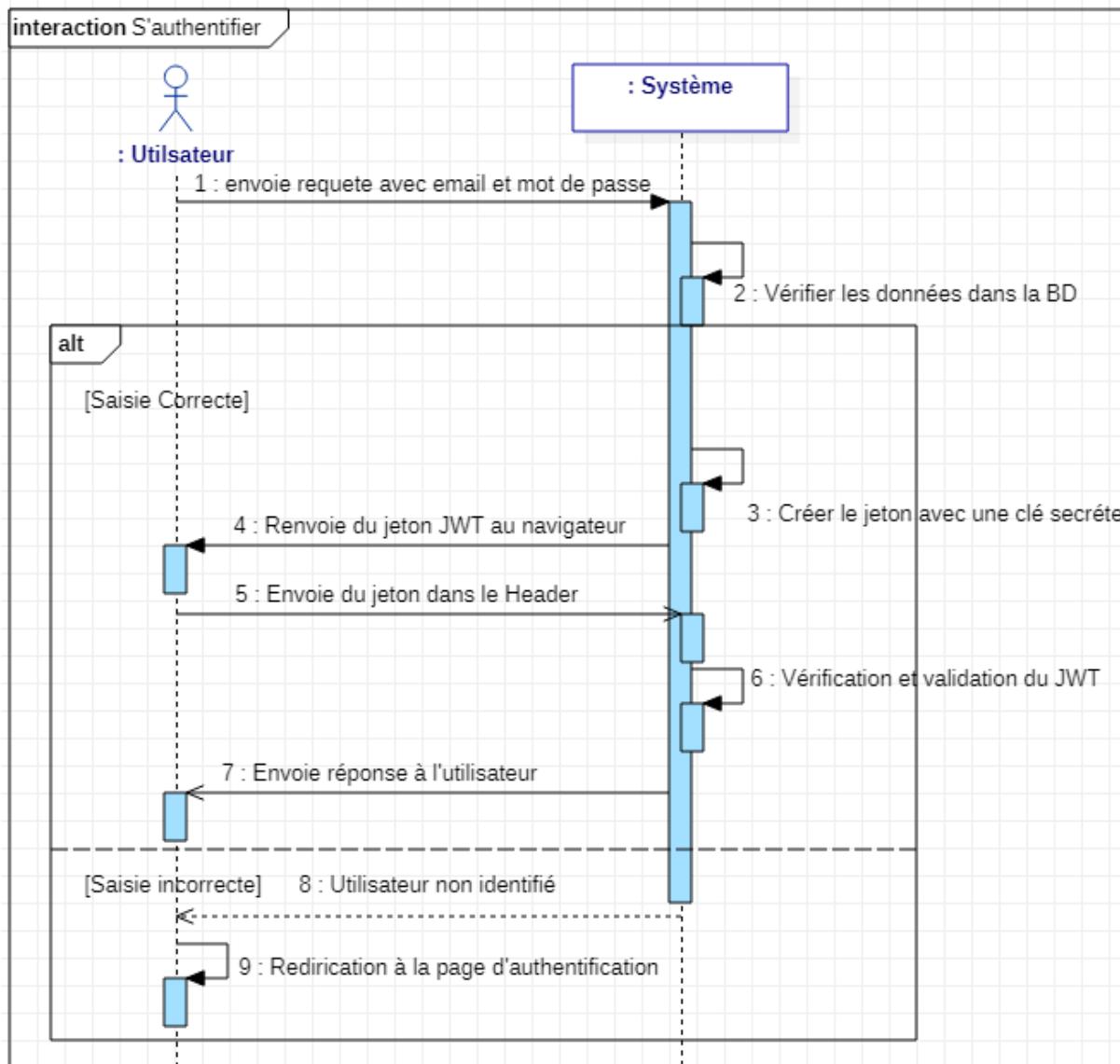


Figure 12 : Diagramme de séquence du scénario « S'authentifier »

### - Diagramme de séquence du scénario récupérer mot de passe oublié

Le diagramme de séquence du cas d'utilisation « Récupérer mot de passe » est illustré par la figure 13.

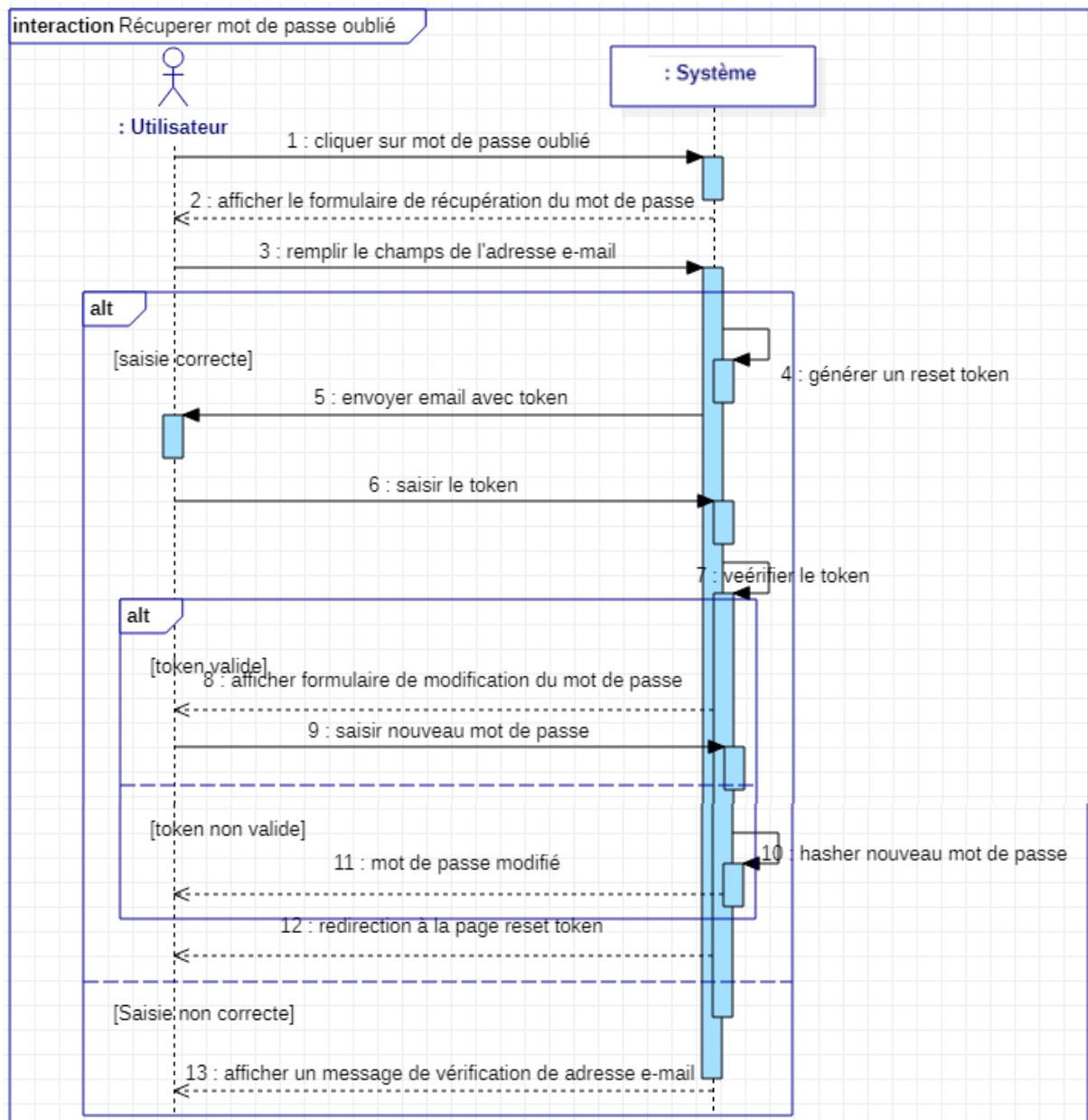
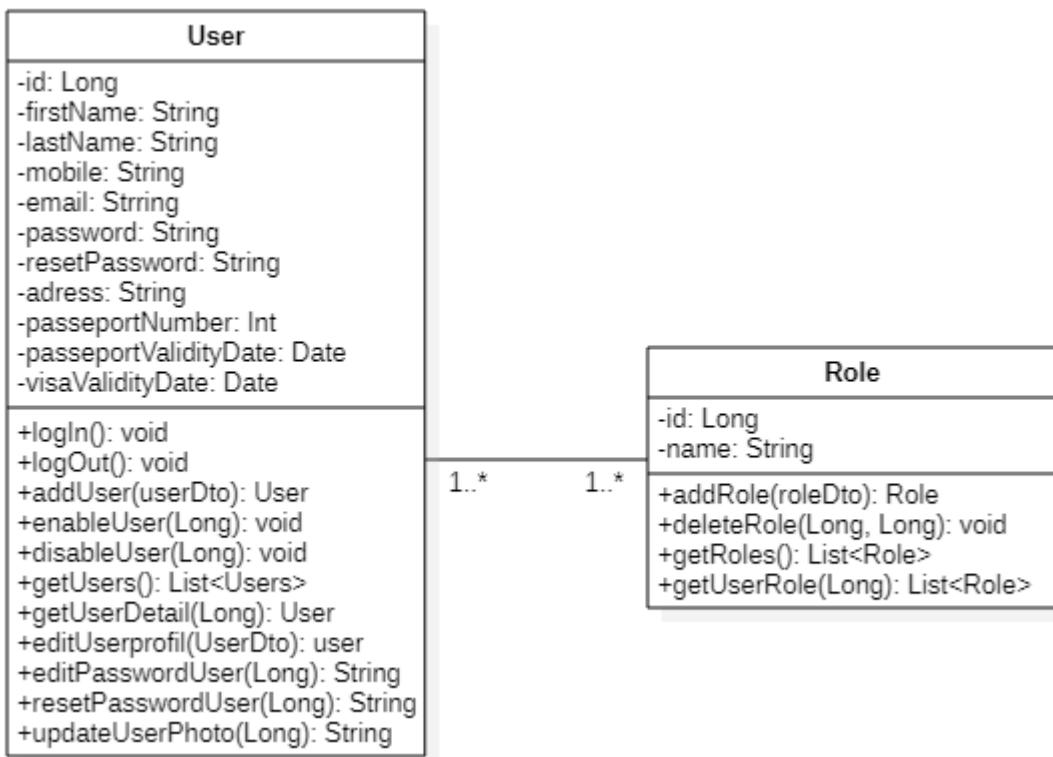


Figure 13 : Diagramme de séquence du scénario « Récupérer mot de passe »

## 2.4. Conception

La conception est une étape critique dans le cycle de vie d'une application, elle vise à développer des modèles détaillés de l'architecture du système et de réduire sa complexité. Elle nous permet de représenter une vue dynamique du système en se référant à un diagramme de classe et une vue statique en présentant des diagrammes d'activités.

Dans ce sprint, nous focalisons sur la gestion d'accès et des utilisateurs. La figure 14 présente le diagramme de classes du sprint numéro 1.



**Figure 14 : Diagramme de classes du sprint numéro 1**

Dans le diagramme de classes de deuxième sprint nous trouvons 2 classes :

- **Classe User** : Cette classe offre 11 méthodes qui sont développées pour gérer un utilisateur, à titre d'exemple la méthode *login()* permettant à un utilisateur de se connecter à l'application, la méthode *enableUser(Long)* et *disableUser(Long)* permettant respectivement d'activer et désactiver le compte d'un utilisateur, la méthode *resetPasswordUser(Long)* permettant de récupérer l'accès à l'application, etc.
- **Classe Role** : L'utilisateur doit avoir le rôle administrateur pour gérer l'accès à l'application.

Le diagramme d'activités du cas d'utilisation « Désactiver un compte » et du cas d'utilisation « Modifier mot de passe » sont présentés respectivement par la figure 99 et la figure 100 à l'annexe.

## 2.5. Réalisation

Dans cette partie nous présenterons les interfaces homme-machine de notre premier sprint, qui est le module « Authentification et gestion d'accès », afin de montrer le mode de fonctionnement de ce dernier.

### 2.5.1. Interface d'authentification

La figure 15 présente l'interface d'authentification. L'utilisateur saisit son adresse e-mail et son mot de passe, s'ils sont corrects, il aura la permission d'accéder aux différentes fonctionnalités de l'application selon son rôle si non un message d'erreur s'affiche.

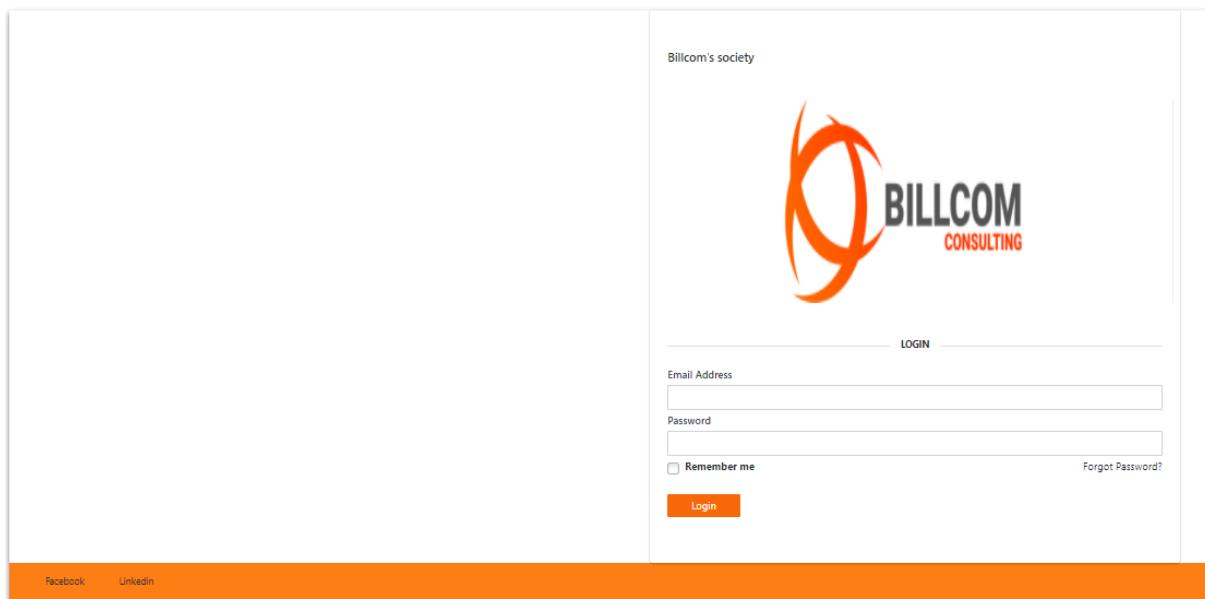


Figure 15 : Interface d'authentification

### Les cas d'erreurs :

Les cas d'erreur que le système peut déclencher :

- Si l'utilisateur saisit une adresse e-mail non valide
- Si un utilisateur ne remplit pas un des champs obligatoires
- Si les données ne sont pas valides

La figure 16 capture quelques cas d'erreurs déclenchées par le système.

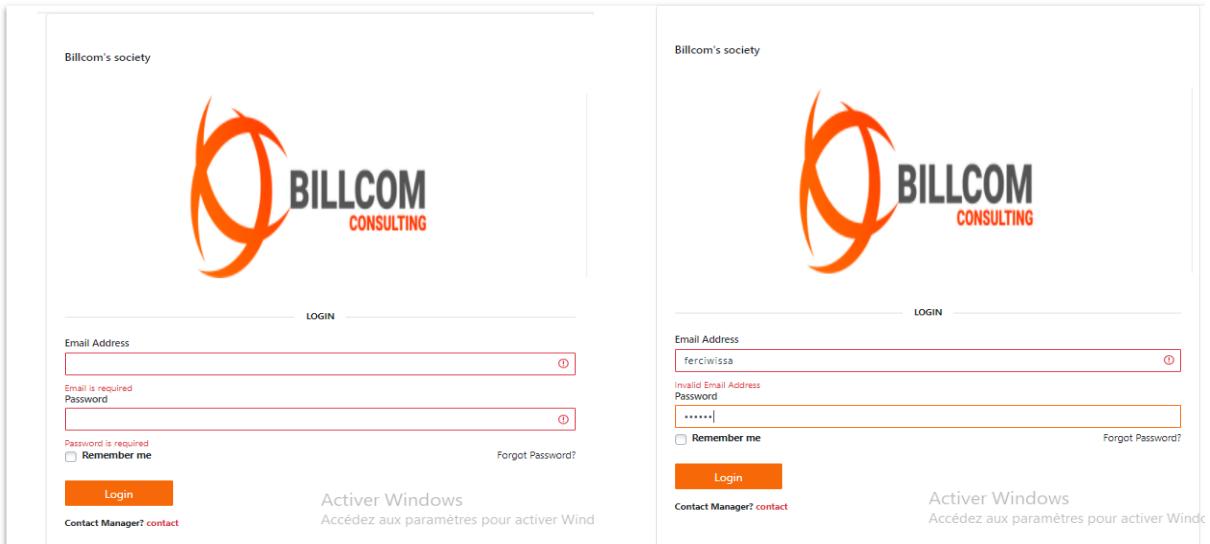


Figure 16 : Les messages des cas d'erreur

### 2.5.2. Interface de récupération du mot de passe

Pour récupérer son mot de passe l'utilisateur doit écrire son e-mail, puis un code va être généré et envoyé automatiquement à l'adresse e-mail de l'utilisateur, ce dernier doit saisir ce code comme l'indique l'interface numéro 17 pour récupérer son accès à l'application, les trois captures 1, 2, 3 présentent les étapes à faire pour ce cas d'utilisation.

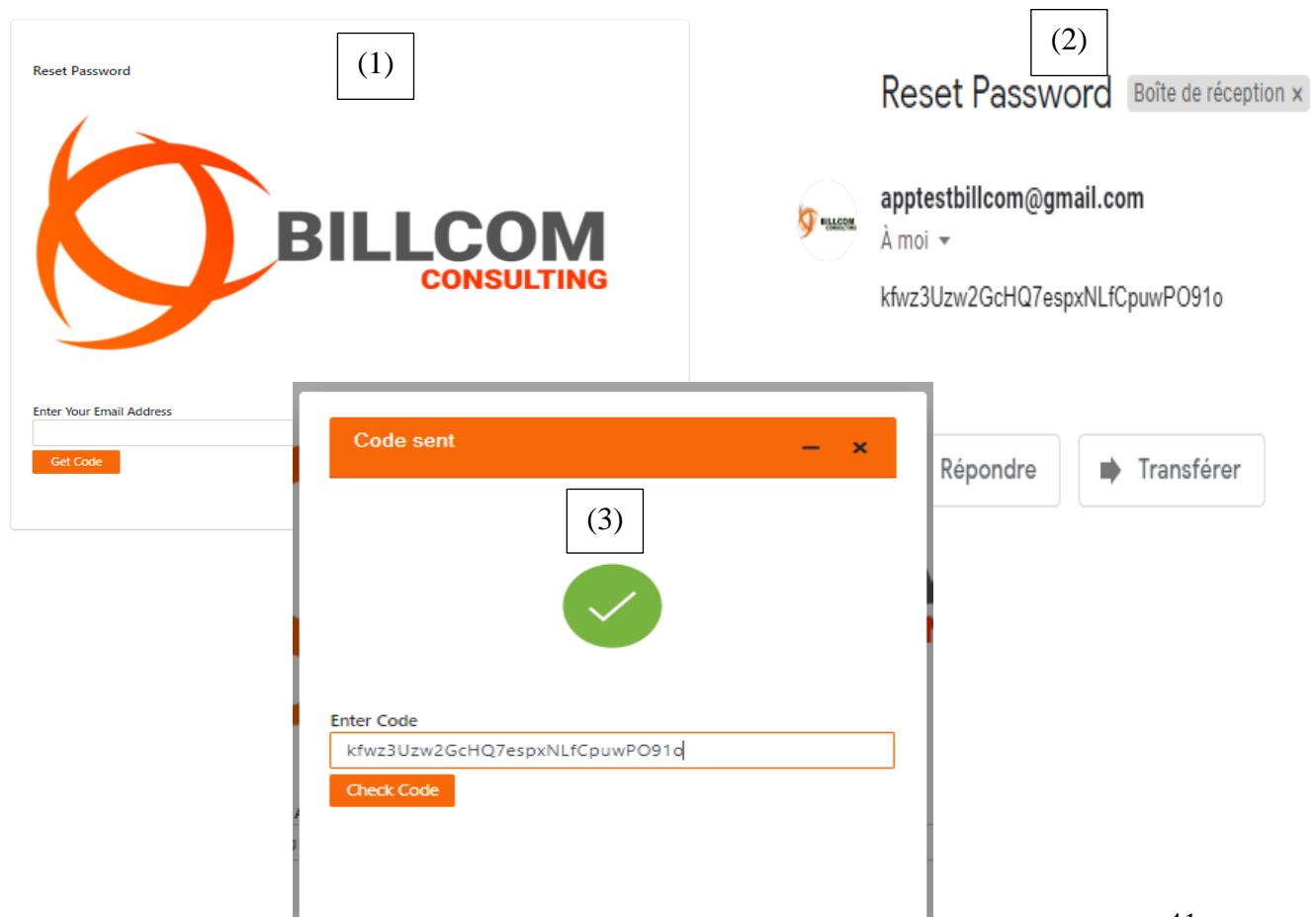


Figure 17 : Récupération du mot de passe oublié

### 2.5.3. Interface de modification du mot de passe

Si l'utilisateur tente de se connecter pour la première fois, il doit modifier son mot de passe pour des raisons de confidentialités de ses données. La figure 18 présente l'interface de modification du mot de passe.

The screenshot shows a login page with the following elements:

- A "Welcome!" message at the top.
- A note below it: "this is your first login! Enter a password that you have receive and update your password please!"
- The Billcom Consulting logo, featuring a stylized orange swirl icon followed by the text "BILLCOM CONSULTING".
- Input fields for "Enter Received Password", "Enter Password", and "Confirm Password".
- An "Update Password" button in orange at the bottom left.
- A "Login" link at the bottom right.

Figure 18 : Interface de modification du mot de passe

### 2.5.4. Interface d'activation et de désactivation d'un compte utilisateur

En se connectant à son application, un administrateur peut activer ou désactiver un compte utilisateur. L'interface de la figure 19 illustre ses fonctionnalités.

The screenshot shows a user management table with the following columns:

First Name	Last Name	Email	Status	Update	Details	Disable	Enable
wissal	fersi	ferciwissal@gmail.com	Active				

Two orange callout boxes with arrows point to the "Disable" and "Enable" icons:

- The "Disable" icon is labeled "Désactiver un compte utilisateur".
- The "Enable" icon is labeled "Activer un compte utilisateur".

Figure 19 : Interface d'activation/désactivation d'un compte

### 3. Sprint 2 : Gestion des utilisateurs et des profils

Pour pouvoir accéder à l'application et gérer les fonctionnalités suivant le rôle, une gestion des utilisateurs et de leurs rôles doit être exploitée ainsi qu'une gestion de leurs profils. Dans ce deuxième sprint de la première release nous allons traiter et développer deux modules :

- Gestion des utilisateurs
- Gestion des profils

#### 3.1. Backlog de produit

Le tableau 17 représente le backlog de notre deuxième sprint « Gestion des utilisateurs et de leurs profil ». Nous exposons dans cette partie les « User Stories » liés à ce sprint pour pouvoir les analyser dans la suite de ce chapitre.

Tableau 17 : Backlog de produit du sprint 2

ID Module	Module	ID User Story	User Story
1.	Gestion des comptes utilisateurs et leurs rôles	1.	En tant qu'administrateur, je veux créer des comptes utilisateurs.
		2.	En tant qu'administrateur, je veux consulter les détails des utilisateurs.
		3.	En tant qu'administrateur, je veux chercher un utilisateur depuis la liste par un filtre.
		4.	En tant qu'administrateur, je veux ajouter un rôle à un utilisateur.
		5.	En tant qu'administrateur, je veux supprimer un rôle d'un utilisateur.
2.	Gestion des profils	1.	En tant qu'utilisateur, je veux consulter mon profil.
		2.	En tant qu'utilisateur, je veux modifier mes informations.
		3.	En tant qu'utilisateur, je veux ajouter des compétences dans mon profil.
		4.	En tant qu'utilisateur je veux charger et télécharger des fichiers dans mon profil.

#### 3.2. Analyse

Au niveau de cette section, nous avons commencé par l'exposition du backlog de produit du sprint, ensuite, nous passons à la phase d'analyse dans laquelle nous présentons le diagramme de cas d'utilisation ainsi, des descriptions textuelles de quelques cas d'utilisation.

### 3.2.1. Diagramme de cas d'utilisation

La figure 20 présente le diagramme de cas d'utilisation du deuxième sprint, l'acteur utilisateur peut être un chef de projet, un chef d'équipe et un membre.

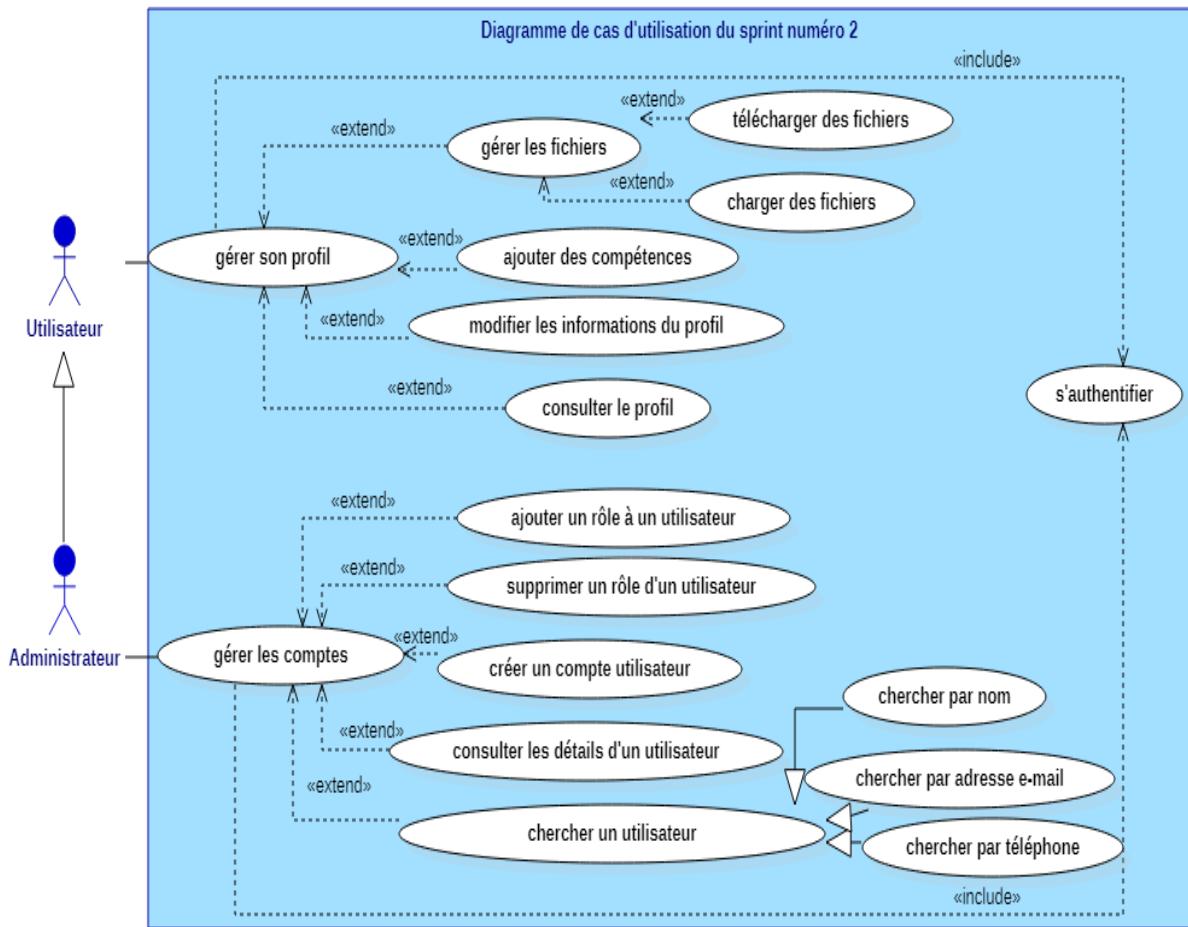


Figure 20 : Diagramme de cas d'utilisation du sprint numéro 2

### 3.2.2. Raffinement des cas d'utilisation

Dans cette partie, nous allons raffiner les cas d'utilisation du notre diagramme en présentant une description textuelle de quelques cas.

#### - Description textuelle du cas d'utilisation : Ajouter un compte utilisateur

Le tableau 18 illustre le raffinement du cas d'utilisation « Ajouter un compte utilisateur », en présentant l'acteur de la fonctionnalité, les conditions, le scénario nominal et les exceptions de cette fonctionnalité.

**Tableau 18 : Description textuelle du cas d'utilisation « Ajouter un compte utilisateur »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Ajouter un compte utilisateur
<b>Acteur</b>	Administrateur de l'application
<b>Objectif</b>	Créer un compte utilisateur
<b>Résumé</b>	Un utilisateur ayant le rôle administrateur ajoute un compte d'un utilisateur avec succès.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
<ul style="list-style-type: none"> <li>- S'authentifier en tant qu'administrateur.</li> <li>- l'email du nouveau compte est inexistant dans la base.</li> <li>- Saisir des informations valides.</li> </ul>	Le compte de l'utilisateur est créé avec succès.
<b>Scénario</b>	
<ol style="list-style-type: none"> <li>1- L'administrateur s'authentifie avec son compte.</li> <li>2- L'administrateur accède à la partie ajout d'un utilisateur.</li> <li>3- L'administrateur remplit les champs de l'utilisateurs à ajouter.</li> <li>4- Le système vérifie la validité des données.</li> <li>5- Le système valide la création du compte de l'utilisateur.</li> <li>6- Le système affiche une notification de création avec succès.</li> <li>7- Le système envoie les informations du compte crée à l'utilisateur par e-mail de l'application.</li> </ol>	
<b>Exception</b>	
<ol style="list-style-type: none"> <li>1-Si les informations de saisie ne sont pas valides, un message d'erreur s'affiche en spécifiant le problème.</li> <li>2-Si l'adresse e-mail de l' utilisateur existe dans la base de données un message d'erreur s'affiche.</li> </ol>	

**- Description textuelle du cas d'utilisation : Ajouter un rôle à l'utilisateur**

Le tableau 19, illustre le raffinement du cas d'utilisation « Ajouter un rôle à un utilisateur », en présentant l'acteur de la fonctionnalité, les conditions, le scénario nominal et les exceptions de cette fonctionnalité.

**Tableau 19 : Description textuelle du cas d'utilisation « Ajouter un rôle à un utilisateur »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Ajouter un rôle à un utilisateur
<b>Acteur</b>	Administrateur de l'application
<b>Objectif</b>	Un utilisateur aura la possibilité de manipuler les fonctionnalités selon le rôle ajouté.
<b>Résumé</b>	Un rôle ajouté à un utilisateur.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
- S'authentifier en tant qu'administrateur.	Le nouveau rôle est associé à l'utilisateur.
<b>Scénario</b>	
1- L'administrateur s'authentifie avec son compte. 2- L'administrateur sélectionne l'utilisateur auquel va ajouter le rôle. 3- L'administrateur clique sur le bouton ajouter rôle. 4- Le système affiche la liste des rôles. 5- L'administrateur choisit le rôle à partir d'une liste. 6- Le système affiche une notification d'ajout avec succès.	
<b>Exception</b>	
Un utilisateur possède déjà le rôle qui a été sélectionné alors il yaura un retour à l'étape numéro 4.	

#### **- Description textuelle du cas d'utilisation : Modifier les informations du profil**

Le tableau 20, illustre le raffinement du cas d'utilisation « Modifier les informations du profil », en présentant l'acteur de la fonctionnalité, les conditions, le scénario nominal et les exceptions de cette fonctionnalité.

**Tableau 20 : Description textuelle du cas d'utilisation « Modifier informations profil »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Modifier les informations du profil.
<b>Acteur</b>	Utilisateur de l'application (administrateur /chef de projet/chef d'équipe/membre).
<b>Objectif</b>	Un profil utilisateur modifié.

<b>Résumé</b>	Un utilisateur authentifié peut modifier des informations de son profil.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
- Un utilisateur authentifié.	Un profil mis à jour.
<b>Scénario</b>	
1- L'utilisateur accède à la page informations utilisateurs. 2- L'utilisateur clique sur le bouton modifier. 3- Le formulaire des informations s'affiche en format input. 4- L'utilisateur entre des modifications sur son profil. 5- L'utilisateur valide son choix. 6- Le système affiche un message de succès.	
<b>Exception</b>	
Un utilisateur possède déjà le rôle qui a été sélectionné alors il y'aura un retour à l'étape numéro 4.	

### 3.3. Conception

Au niveau de cette section, nous allons entamer la phase de conception du premier sprint. Pour se faire, nous présentons la vue dynamique de ce deuxième sprint en faisant recours au diagramme de classes et aux diagrammes de séquence.

#### 3.3.1. Diagramme de classes

La figure 21 présente le diagramme de classes de ce sprint.

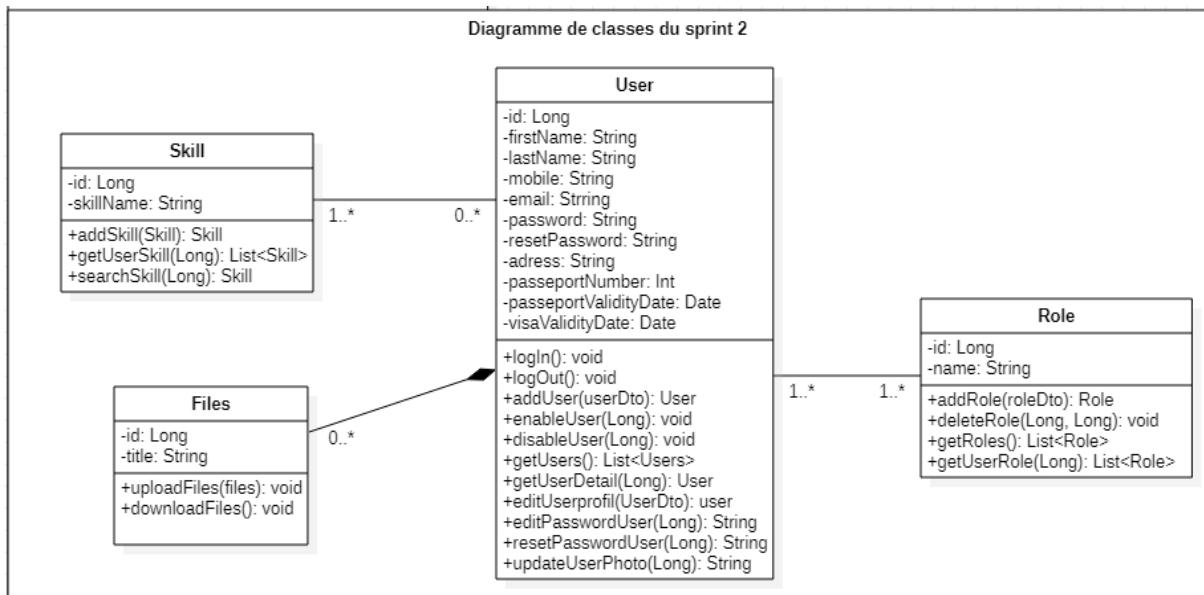


Figure 21 : Diagramme de classes du sprint numéro 2

Dans le diagramme de classes de deuxième sprint nous trouvons 4 classes :

- **Classe User** : En plus que les méthodes citées dans le diagramme de classes du sprint 1, la classe « User » offre la méthode *addUser(userDto)* permettant d'ajouter un utilisateur, la méthode *getUserDetail(Long)* permettant de consulter le profil d'un utilisateur, la méthode *editUserProfil(Long)*, etc.
- **Classe Role** : Pour gérer les rôles, cette classe offre 4 méthodes qui sont : *addRole(roleDto)* cette méthode permet d'ajouter un rôle à un utilisateur ce qui engendre l'ajout des droits d'accès spécifique à ce rôle, *deleteRole(Long, Long)* cette méthode permet de supprimer un rôle d'un utilisateur, et la méthode *getUserRoles(Long)* permettant de consulter la liste des rôles attribuées à un utilisateur donné, etc.
- **Classe Skill** : La classe « Skill » offre trois méthodes qui sont : *addSkill(skill)* pour ajouter une compétence à un utilisateur, *getUserSkill(String)* pour consulter la liste des compétences d'un utilisateur donné, *searchSkill(Long)* pour chercher les utilisateurs ayant une compétence donnée.
- **Classe Files** : C'est une composition liée à la classe « User », elle comporte deux méthodes qui sont : *downloadFiles()* cette méthode offre la possibilité à un utilisateur de charger des fichiers dans son profil et *uploadFiles()* permettant à un utilisateur de télécharger des fichiers de son profil.

### 3.3.2. Diagramme de séquence

Les diagrammes de séquence offrent une description des scénarios des cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets. Dans cette partie, nous allons exposer quelques scénarios dans des diagrammes de séquence.

- Diagramme de séquence du cas d'utilisation « Ajouter un compte »

Le diagramme de séquence du cas d'utilisation « Ajouter un compte » est présenté par la figure 22.

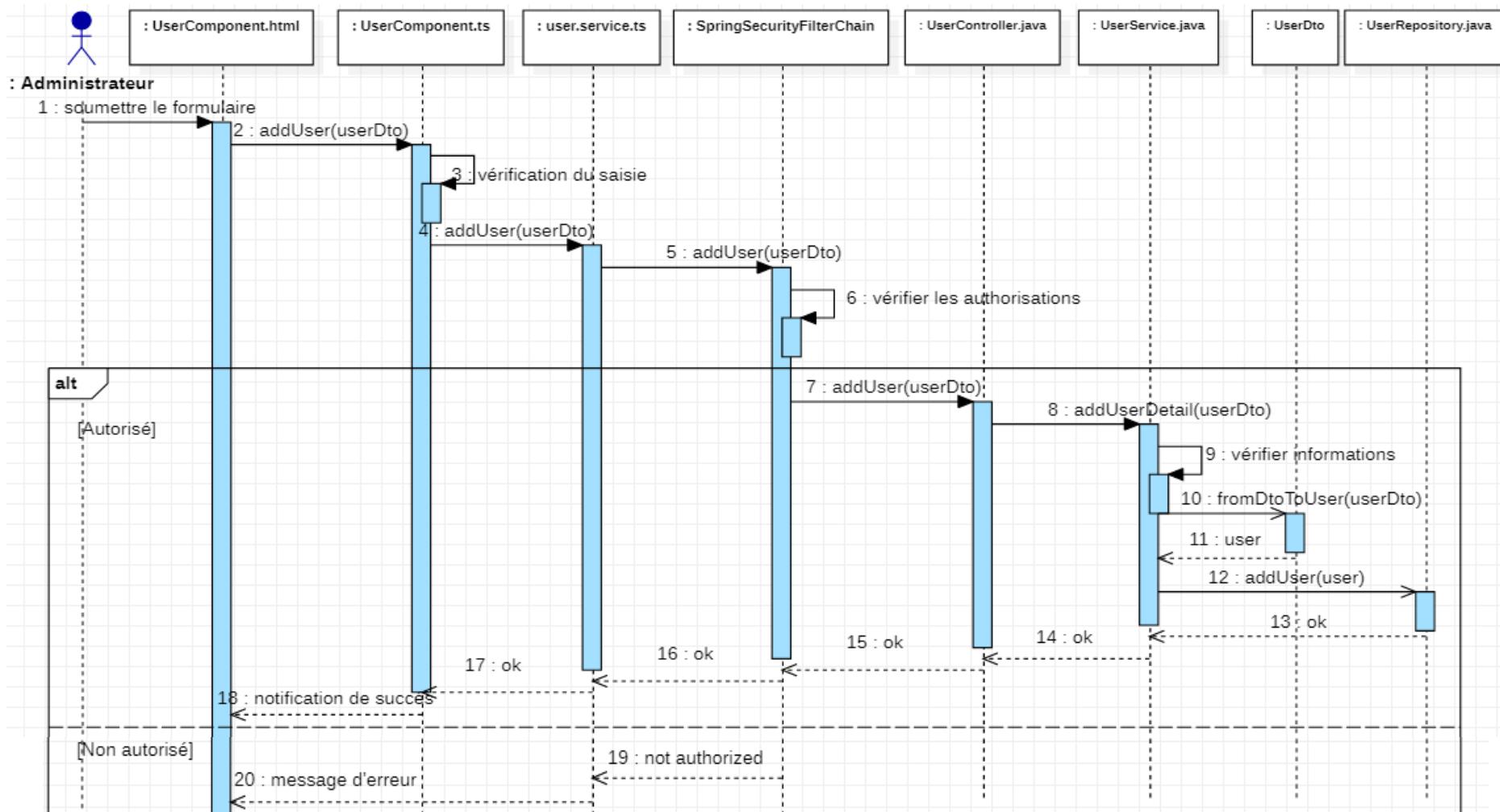


Figure 22 : Diagramme de séquence du cas d'utilisation « Ajouter un compte utilisateur »

Le diagramme de séquence du cas d'utilisation « Consulter détails d'un utilisateur » est présenté en annexe par la figure 101.

## 3.4. Réalisation

Après avoir traité la partie conception, nous exposerons dans ce qui suit quelques interfaces de notre système réalisées au cours de ce deuxième sprint.

### 3.4.1. Interface d'ajout d'un utilisateur

Pour ajouter des utilisateurs à son application, un administrateur authentifié aura le droit de créer des comptes en respectant les contraintes de validation. La figure 23 présente le formulaire de création du compte utilisateur.

L'écran principal affiche une liste d'utilisateurs avec leurs noms et prénoms. Un écran modale intitulé "Create User" permet de saisir de nouveaux détails. Un troisième écran montre une liste d'utilisateurs avec des boutons pour modifier, désactiver ou activer chaque compte.

Figure 23 : Interface d'ajout d'un compte utilisateur

### Les cas d'erreurs :

Le non-respect des validateurs mis par le développeur peut déclencher des erreurs, citons à titre d'exemple les deux cas suivant :

- Si l'administrateur ne respecte pas les validateurs
- Si une adresse email existe dans la base de données

L'interface 24 présente quelques cas d'erreurs déclenchés par le système.

The image displays two identical 'Create User' forms side-by-side, illustrating user input validation errors. Both forms have orange headers and white bodies.

**Left Form Fields:**

- First Name: Marwa
- Last Name: Hamdi
- Mobile: 53789399
- Adresse: Rue montfleury
- email: marwaHamdi (highlighted in red)
- Format Email invalid (error message)
- Password: (redacted)
- Confirm Password: (redacted)
- Confirm Password is required (error message)
- Role: manager

**Right Form Fields:**

- First Name: Marwa
- Last Name: Hamdi
- Mobile: 53789399
- Adresse: Rue montfleury
- email: marwaHamdi@gmail.com
- Password: (redacted)
- Confirm Password: (redacted)
- Passwords must match (error message)
- Role: manager

**Buttons:**

- Left: Register (orange), Cancel (red)
- Right: Register (orange), Cancel (red)

Figure 24 : Cas d'erreur d'ajout du compte utilisateur

### 3.4.2. Interface de consultation de la liste des utilisateurs

L'interface de la figure 25, ne peut être visible que par l'administrateur de l'application. En effet, il aura le droit de consulter la liste des utilisateurs et appliquer des opérations.

The screenshot shows a user management interface with a sidebar and a main content area.

**Sidebar:**

- Billcom logo
- User profile: Heni Ouehezi
- Operations dropdown menu:
  - Profile
  - Employees
  - Charts
  - Projects
  - Team
  - Notification

**Main Content Area:**

First Name	Last Name	Email	Status	Update	Details	Disable	Enable
adem	fersi	ademfersi@gmail.com	Active				
elaa	fersi	elaafersi@gmail.com	Inactive				
Heni	Ouehezi	HeniOuehezi@hotmail.com	Active				
nesrine	werdi	nesrinewerdi@yahoo.com	Active				
radhouan	maghraoui	radhouanMaghi@gmail.com	Active				
rami	feхи	ramifehi@gmail.com	Inactive				

Buttons at the bottom right: Add new user, Search users..., Previous, Next.

Figure 25 : Interface de la liste des utilisateurs

### 3.4.3. Interface de recherche sur la liste des utilisateurs

La recherche d'un utilisateur peut être effectuée par plusieurs manières à savoir :

- La recherche par nom

First Name	Last Name	Email	Status	Update	Details	Disable	Enable
Wifek	Hajji	wifekhajji@gmail.com	Active				
wissal	fersi	ferciwissal@gmail.com	Active				

- La recherche par prénom

First Name	Last Name	Email	Status	Update	Details	Disable	Enable
Heni	Ouelhezi	HeniOuelhezi@hotmail.com	Active				

- La recherche par numéro de téléphone

First Name	Last Name	Email	Status	Update	Details	Disable	Enable
adem	fersi	ademferci@gmail.com	Active				
radhouan	maghraoui	radhouanMaghi@gmail.com	Active				
Yasmine	Ben Slimene	yasmineBenSlimen@gmail.com	Active				

Figure 26 : Opérations de recherche appliquée sur la liste

### 3.4.4. Interfaces de gestion du rôle d'un utilisateur

Chaque utilisateur aura une liste des rôles. En consultant cette liste, un administrateur peut appliquer des opérations selon son besoin. Dans la section ci-après, nous allons présenter la partie gestion des rôles.

#### - Consulter la liste du rôle d'un utilisateur

Chaque utilisateur peut avoir plusieurs rôles, la figure 27 présente la liste des rôles d'un utilisateur sélectionné.

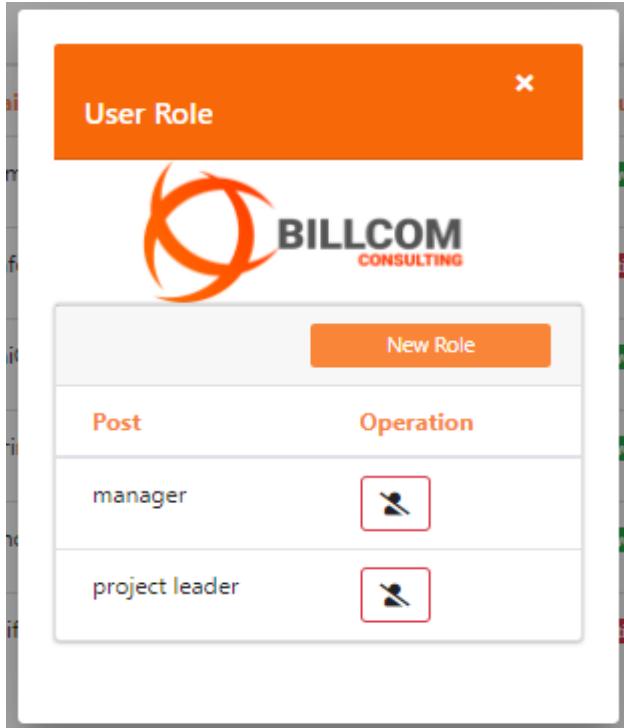


Figure 27 : Interface de la liste des rôle d'un utilisateur

#### - Opérations sur la liste

Un administrateur peut ajouter et/ou supprimer un rôle d'un utilisateur, les figures 28 et 29 illustrent ces deux opérations.

L'écran affiche une interface intitulée "Add Role". En haut à droite, il y a un bouton d'arrêt. Ensuite, le logo "BILLCOM CONSULTING" est visible. En bas à droite, il y a un bouton "Add" et un bouton "Cancel". La liste des rôles est dans une boîte déroulante :

manager
leader
member
project Leader

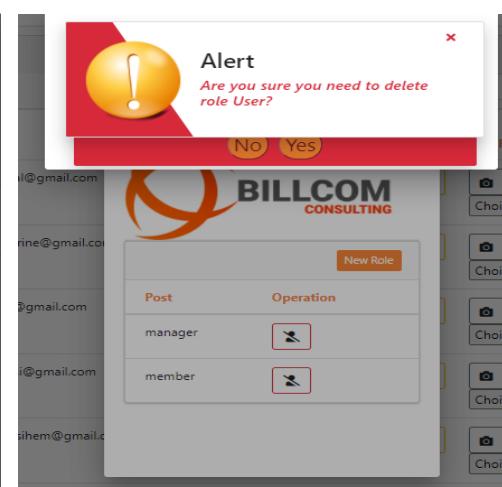


Figure 29 : Ajout d'un rôle à un utilisateur

Figure 28 : Supprimer un rôle de la liste des rôles

### 3.4.5. Interface de consultation du profil d'un utilisateur

Une fois l'utilisateur connecté, il peut consulter ses informations personnelles, ses compétences, ses propres fichiers, et s'il a le rôle membre, il peut consulter son avancement de travail dans chaque équipe.

Figure 30 : Consultation du profil utilisateur

- En cliquant sur le bouton « Add skill », un formulaire s'affiche et l'utilisateur peut ajouter une compétence à son profil également en cliquant sur le bouton « Edit profile », ce dernier peut mettre à jour son profil comme l'indique l'interface 31.

Figure 31 : Mise à jour du profil

L’interface de consultation de détail d’un utilisateur, les tests unitaire, l’analyse du code avec sonarLint et la configuration de ce sprint avec swaggerUI sont présentés par les figures 102, 103, 104 et 105 au niveau de l’annexe.

## Conclusion

Au niveau de ce chapitre, nous avons présenté la première release de notre application qui comporte deux sprint.

Pour chaque sprint, nous avons présenté le backlog produit en premier temps. Ensuite nous avons présenté les différentes fonctionnalités réalisées à travers le diagramme de cas d’utilisation raffiné, les descriptions textuelles de quelques cas. Puis nous avons mis en disposition la conception en présentant un diagramme de classes, des diagrammes de séquence et d’activités. Après, nous avons exposé quelques interfaces graphiques. Et finalement, nous avons présenté la partie test et documentation de la release. Dans le chapitre suivant, nous allons présenter la deuxième release.

# **Chapitre 4 : Release 2 « Gestion et planification du projet »**

## **Introduction**

En se basant sur le même principe de la première release, nous partons par la présentation de la release en se reposant sur le backlog de produit de chaque sprint. Par la suite, une analyse spécifique pour ce sprint sera effectuée, suivie de la phase conception et finalement nous présentons ce que nous avons abouti à réaliser durant cette release.

### **1. Présentation de la release 2**

Cette section vise à exposer les différentes phases de mise en œuvre de la deuxième release, elle consiste à présenter la configuration du projet. Cette release va être décomposée en 2 sprints qui nous permettent à la fin de chacun de réaliser un livrable.

- Sprint 3 : Gestion et planification du projet et de ses équipes
- Sprint 4 : Gestion et planification des tâches

Nous présentons dans ce chapitre la démarche de la gestion du projet du moment où il est créé et affecté à un chef de projet jusqu'à son achèvement. Les fonctionnalités de chaque sprint sont liées à des acteurs différents à savoir le chef de projet, le chef d'équipe, et le membre de l'équipe. Nous présentons pour chaque sprint le backlog de produit, l'analyse contextuelle, la conception et nous clôturons par la présentation de la partie réalisation.

### **2. Sprint 3 : Gestion et planification du projet et de ses équipes**

Dans ce sprint nous allons nous focaliser sur la gestion des projets et de leurs équipes.

#### **2.1. Backlog de produit du sprint**

Avant de se lancer dans ce sprint, nous présentons les « Users Stories » et les tâches à développer. Le tableau 21 décrit le backlog de produit du premier sprint de la release 2.

**Tableau 21 : Backlog de produit du sprint 3**

<b>ID Module</b>	<b>Module</b>	<b>ID User Story</b>	<b>User Story</b>
1.	Gestion et planification du projet	1.	En tant qu'administrateur, je veux créer un projet et l'affecter à un chef de projet.
		2.	En tant qu'administrateur, je veux consulter la liste des projets et leurs états d'avancement.
		3.	En tant que chef de projet, je veux consulter la liste de mes projets et leurs états d'avancement.
		4.	En tant que chef de projet, je veux consulter la liste de mes projets ayant le statut « pending ».
		5.	En tant qu'administrateur et/ou chef de projet, je veux modifier le statut d'un projet.
		6.	En tant qu'administrateur/chef projet, je veux consulter et modifier les détails d'un projet.
2.	Gestion de l'équipe du projet	1.	En tant que chef de projet, je veux créer un ou plusieurs équipe(s) liés à mon projet.
		2.	En tant que chef de projet, je veux supprimer une équipe de mon projet.
		3.	En tant que chef de projet et/ou chef d'équipe, je veux consulter les détails de l'équipe.
		4.	En tant que chef d'équipe/membre, je veux consulter la liste de mes équipes.
		5.	En tant que chef d'équipe, je veux ajouter/supprimer un membre de mon équipe.

## 2.2. Analyse

Après avoir défini les exigences de ce sprint, nous passons au diagramme de cas d'utilisation du troisième sprint, nous présentons par la suite le raffinement de certains cas d'utilisation, nous présentons par la suite des diagrammes de séquence des fonctions essentielles.

### 2.2.1. Diagramme de cas d'utilisation

La figure 32 illustre le diagramme de cas d'utilisation raffiné pour le premier sprint de la release 3 qui définit les cas d'utilisation relatifs aux fonctionnalités attendues.

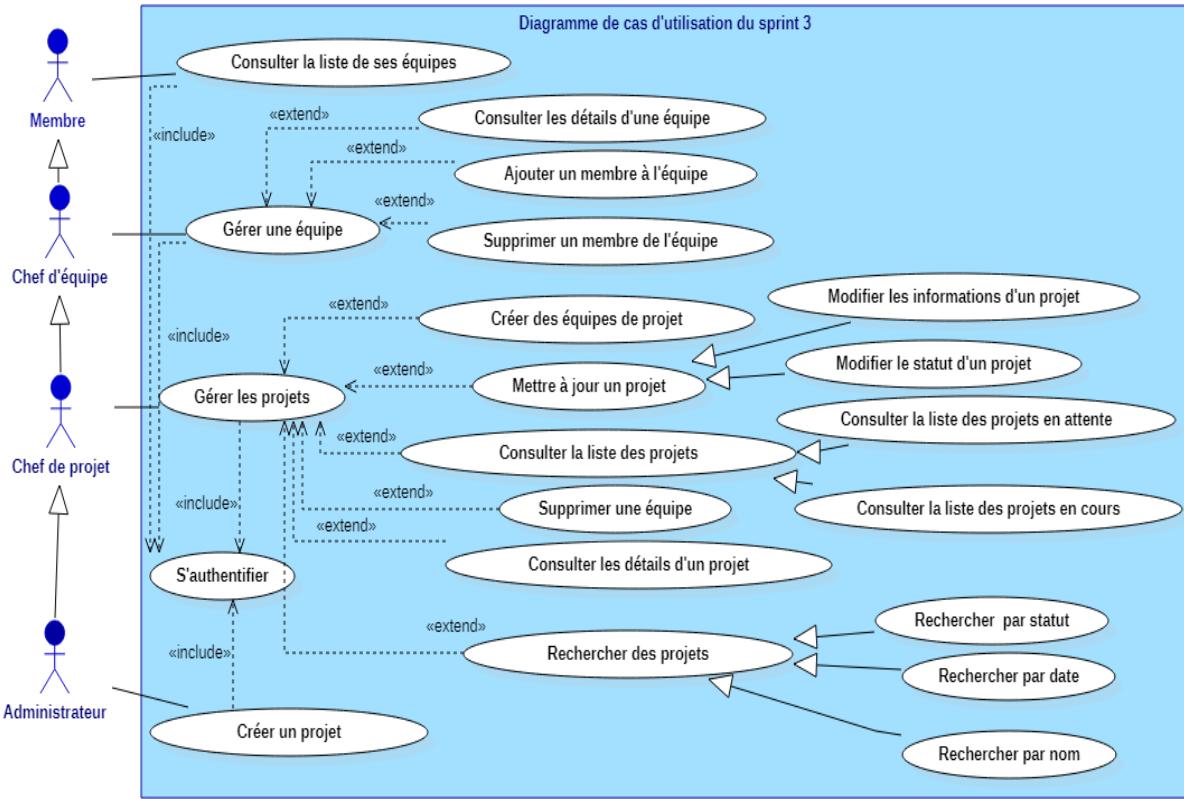


Figure 32 : Diagramme de cas d'utilisation du sprint numéro 3

### 2.2.2. Raffinement des cas d'utilisation

Dans cette section, nous choisissons quelques cas d'utilisation à détailler par une description textuelle afin de mieux expliquer leurs scénarios.

- **Raffinement du cas d'utilisation « Créer un projet »**

La figure 33 présente le raffinement de cas d'utilisation « Créer un projet ».

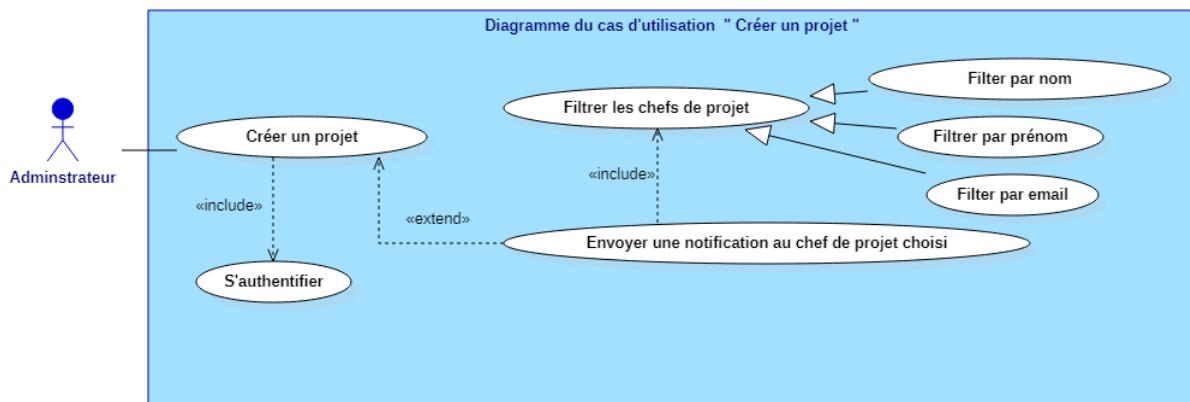


Figure 33 : Raffinement du cas d'utilisation « Créer un projet »

Le tableau 22 est un tableau récapitulatif du cas d'utilisation « Créer un projet ». Nous décrivons dans ce tableau l'objectif ainsi que l'acteur principal de la fonctionnalité et nous introduisons le scénario nominal et les exceptions qui peuvent se produire.

**Tableau 22 : Description textuelle du cas d'utilisation « Créer un projet »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Créer un projet
<b>Acteur</b>	Administrateur
<b>Objectif</b>	Ce cas d'utilisation permet à l'administrateur de créer un projet et l'affecter à un chef de projet.
<b>Résumé</b>	L'administrateur accède à l'interface d'ajout d'un projet et crée un projet.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
- L'utilisateur doit être authentifié et ayant le rôle administrateur.	L'utilisateur a le droit d'accéder à l'interface d'ajout de projet.
<b>Scénario</b>	
1. L'administrateur clique sur le bouton « New Project » 2. Le système affiche un formulaire 3. L'administrateur saisit les champs demandés 4. L'administrateur choisit un chef de projet 5. Le système vérifie les champs et ajoute le projet 5. Le système envoie une notification au chef de projet	
<b>Exception</b>	
1-Si la date de démarrage du projet est inférieure à la date de la création, un message d'erreur sera affiché. 2- Si la date de fin du projet est inférieure à la date de début du projet, un message d'erreur s'affiche. 3-Si l'un des champs (titre de projet, date de début , date de fin, chef de projet) est vide un message d'erreur sera affiché	

- **Description textuelle du cas d'utilisation « Chercher un projet »**

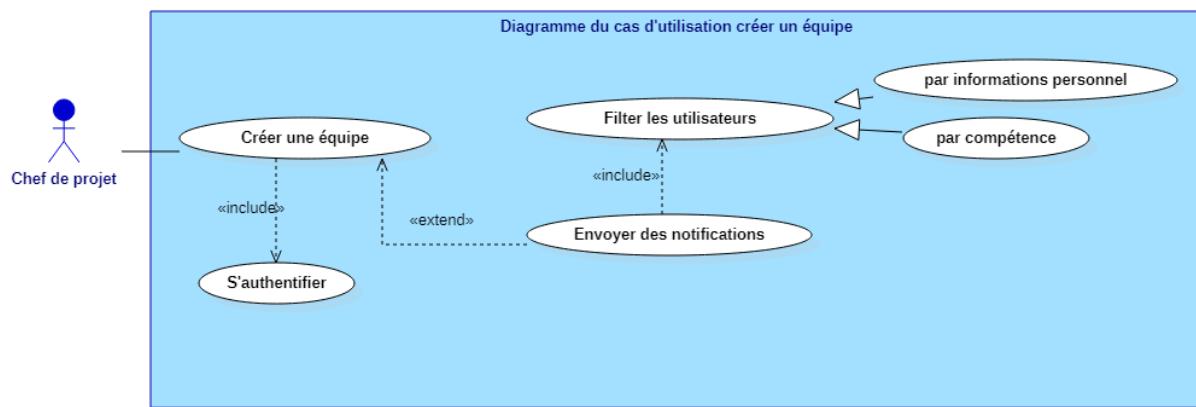
Le tableau 23 présente la description textuelle du cas d'utilisation « Chercher un projet ».

**Tableau 23 : Description textuelle du cas d'utilisation « Chercher un projet »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Chercher un projet
<b>Acteur</b>	Administrateur/chef de projet
<b>Objectif</b>	Le chef de projet/l'administrateur cherche un projet spécifique en choisissant une filtre de recherche (recherche par nom/statut/période).
<b>Résumé</b>	L'administrateur choisit le type de recherche et l'applique pour trouver un projet.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
<ul style="list-style-type: none"> <li>- utilisateur authentifié et ayant le rôle chef de projet et/ou l'administrateur</li> <li>- liste des projets n'est pas vide</li> </ul>	L'utilisateur a le droit d'accéder à l'interface d'ajout de la liste de projet pour la recherche.
<b>Scénario</b>	
<ol style="list-style-type: none"> <li>1- L'administrateur ou chef de projet clique sur le bouton « Filter By »</li> <li>2- L'administrateur ou le chef de projet choisit le type de filtrage (période de temps, statut de projet, nom du projet).</li> <li>3- Le système affiche un formulaire selon son choix.</li> <li>4- L'administrateur ou le chef de projet saisit les champs</li> <li>5- Le système affiche le résultat de recherche.</li> </ol>	
<b>Exception</b>	
1- Si le projet cherché n'existe pas, ou pendant la période choisie, il n'y a pas de projet, le système affiche la liste des projets de nouveau.	

### - Raffinement du cas d'utilisation : Créer une équipe

La figure 34 présente le raffinement du cas d'utilisation « Créer une équipe ».



**Figure 34 : Raffinement du cas d'utilisation « Créer une équipe »**

### - Description textuelle du cas d'utilisation « Créer une équipe »

La description du cas d'utilisation « Créer une équipe » est présentée par le tableau 24.

**Tableau 24 : Description textuelle du cas d'utilisation « Créer une équipe »**

Raffinement du cas d'utilisation	
<b>Cas d'utilisation</b>	Créer une équipe du projet
<b>Acteur</b>	Chef de projet/administrateur
<b>Objectif</b>	Créer des équipes pour un projet.
<b>Résumé</b>	Le chef de projet remplit un formulaire en respectant les exigences afin d'ajouter une équipe à un projet.
Conditions	
<b>Préconditions</b>	<b>Post Conditions</b>
- L'utilisateur doit être connecté et ayant le rôle chef de projet.	Le projet aura des équipes de travail.
Scénario	
1- Le chef de projet clique sur le bouton « Add new team » 2- Le système affiche un formulaire 3- Le chef de projet saisit les champs demandés 4- Le chef de projet clique sur le bouton « Add Leader »	

- 5- Le système affiche la liste des utilisateurs ayant le statut actif et le rôle chef d'équipe.
- 6- Le chef de projet applique une recherche sur la liste.
- 7- Le chef de projet sélectionne un chef d'équipe.
- 8- Le chef de projet clique sur le bouton « Add Member »
- 9- Le système affiche la liste des utilisateurs ayant le statut actif et le rôle membre.
- 10- Le chef de projet choisit le critère de recherche sur la liste des membres et applique sa recherche.
- 11- Le chef de projet sélectionne une liste des membres.
- 12- Le chef de projet valide son choix.
- 13- Le système affiche une notification d'ajout avec succès.

### Exception

- 1- Si la période d'activité de l'équipe est inférieure à la période de l'activité du projet dont elle est attachée, un message d'erreur sera affiché.
- 2- Si la date de fin du travail de l'équipe est inférieure à la date de début du travail de l'équipe, un message d'erreur s'affiche.
- 3- Si l'un des champs obligatoire est vide un message d'erreur sera affiché

### 2.2.3. Diagramme de séquence

La figure 35 décrit le cas d'utilisation « Supprimer une équipe » en exposant un diagramme de séquence système.

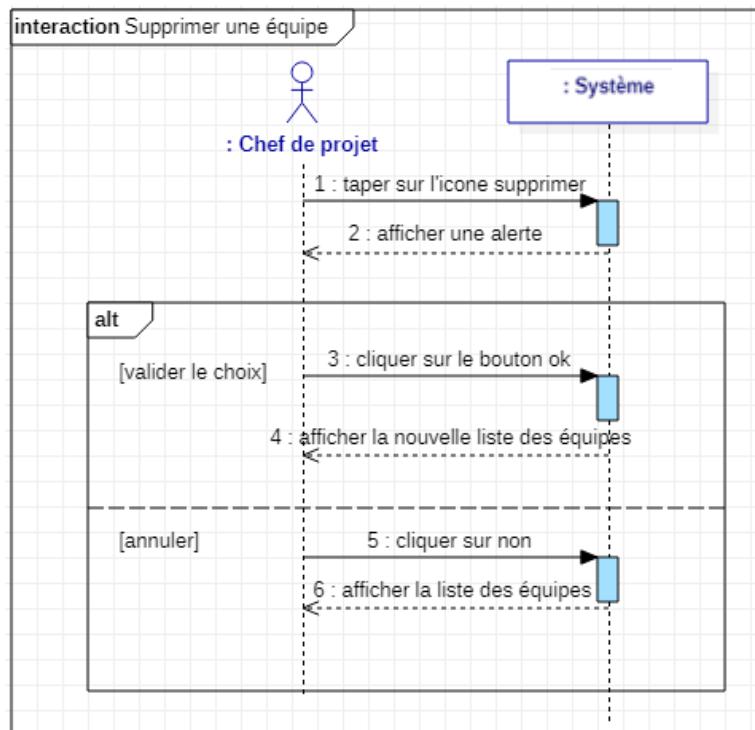


Figure 35 : Diagramme de séquence du cas d'utilisation « Supprimer une équipe »

## 2.3. Conception

Après avoir présenté l'analyse contextuelle de notre sprint, nous entamons la description de la partie conceptuelle. En premier lieu, nous présentons le diagramme des classes de ce sprint et en second lieu nous montrons le diagramme de séquence du scénario « Filtrer la liste projet » et le diagramme de séquence du scénario « Consulter détail d'un projet » et le diagramme d'activités « Modifier un projet ».

### 2.3.1. Diagramme de classes

Le diagramme de classes de troisième sprint est présenté par la figure 36.

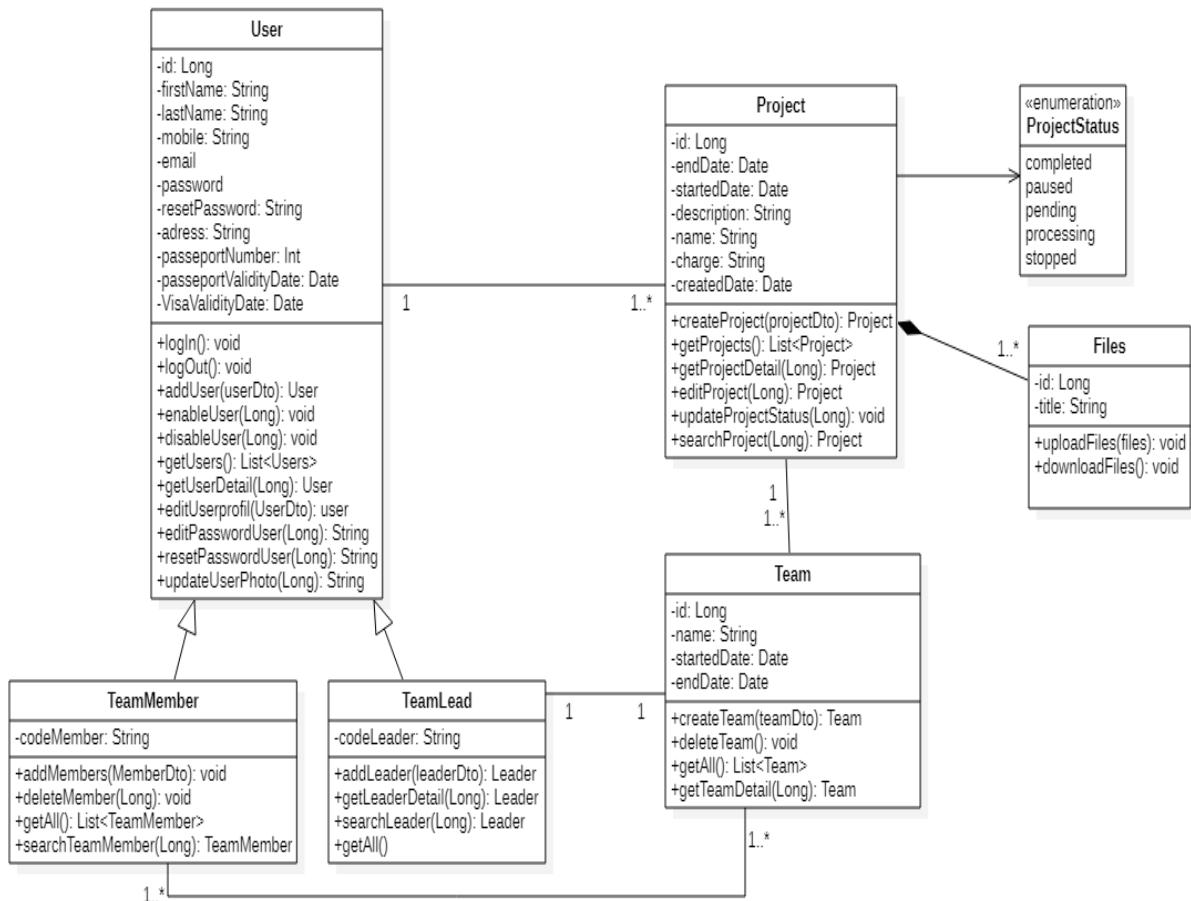


Figure 36 : Diagramme de classes du sprint numéro 3

Le diagramme de classes du sprint 3 présente 7 classes :

- **Classe User** : C'est la classe qui modélise les différents types d'utilisateurs et elle est spécialisée en autres classes à savoir « TeamMember » et « TeamLeader ».
- **Classe TeamMember** : En plus des attributs de l'utilisateur, cette classe a le code (*codeMember*) comme attribut supplémentaire et une association (*ManyToMany*) avec la

classe « Team ». La classe « TeamMember » offre 4 méthodes qui sont : *addMember(Long)* pour ajouter un membre à une équipe, *deleteMember (Long)* pour supprimer un membre de l'équipe, *searchMember(Long)* afin de chercher un membre, *getAll()* pour consulter la liste des membres.

- **Classe TeamLeader:** En plus des attributs de l'utilisateur, cette classe a un attribut spécifique (*codeLeader*) et une association (*OneToOne*) avec la classe « Team ». Cette classe offre 3 méthodes qui sont : *addLeader(LeaderDto)*, *searchLeader(Long)*, *getLeader(Long)*, *getAll()* qui permettent respectivement d'ajouter un chef d'équipe, chercher un chef d'équipe, consulter les détails d'un chef d'équipe et obtenir la liste des chef d'équipe.
- **Classe Project :** Afin de gérer les projets, la classe « Project » offre 6 méthodes, La méthode *createProject(projectDto)* qui offre la possibilité à un administrateur de créer un projet et l'assigner à un chef de projet, *getProjects()* cette méthode permet de consulter la liste des projets, *getProjectDetail(Long)* permettant de consulter les détails d'un projet, *editProject(Long)* pour modifier les détails d'un projet, *updateProjectStatus(Long)* pour modifier le statut d'un projet, *searchProject(Long)* pour chercher un projet spécifique.
- **Classe ProjectStatus :** C'est une classe d'énumération qui contient 5 attributs représentant les différents statuts d'un projet qui sont (*stopped*), (*pendeing*), (*completed*), (*processing*) et (*paused*).
- **Classe Files :** C'est une composition liée à la classe projet, elle comporte deux méthodes qui sont : *downloadFiles()* et *uploadFiles()* qui offrent respectivement la possibilité à un chef de projet de charger et télécharger des fichiers spécifique à un projet.
- **Classe Team :** Afin de gérer les équipes, cette classe offre 4 méthodes qui sont : *createTeam(teamDto)*, *deleteTeam(Long)*, *getAll()* et *getTeamDetail(Long)* et qui offrent respectivement la possibilité d'ajouter une équipe, supprimer une équipe, afficher les détails d'une équipe et consulter la liste des équipes.

### 2.3.2. Diagramme de séquence

Nous allons présenter deux diagrammes de séquences, le premier sera le diagramme de séquence du cas d'utilisation « Filtrer la liste des projets » et le second est le cas d'utilisation « Consulter les détails d'un projet » et il va être présenté au niveau de l'annexe.

### - Diagramme de séquence du cas d'utilisation « Filtrer la liste des projets »

La figure 37 représente le diagramme de séquence du cas d'utilisation « Filtrer la liste des projets ». Lorsque l'administrateur ou le chef de projet clique sur le bouton « Filter By », il doit choisir le filtre, il va avoir la liste des projets filtrée selon les paramètres qu'il a choisis. Il faut mentionner que la méthode du filtre comprend aussi la pagination de la liste.

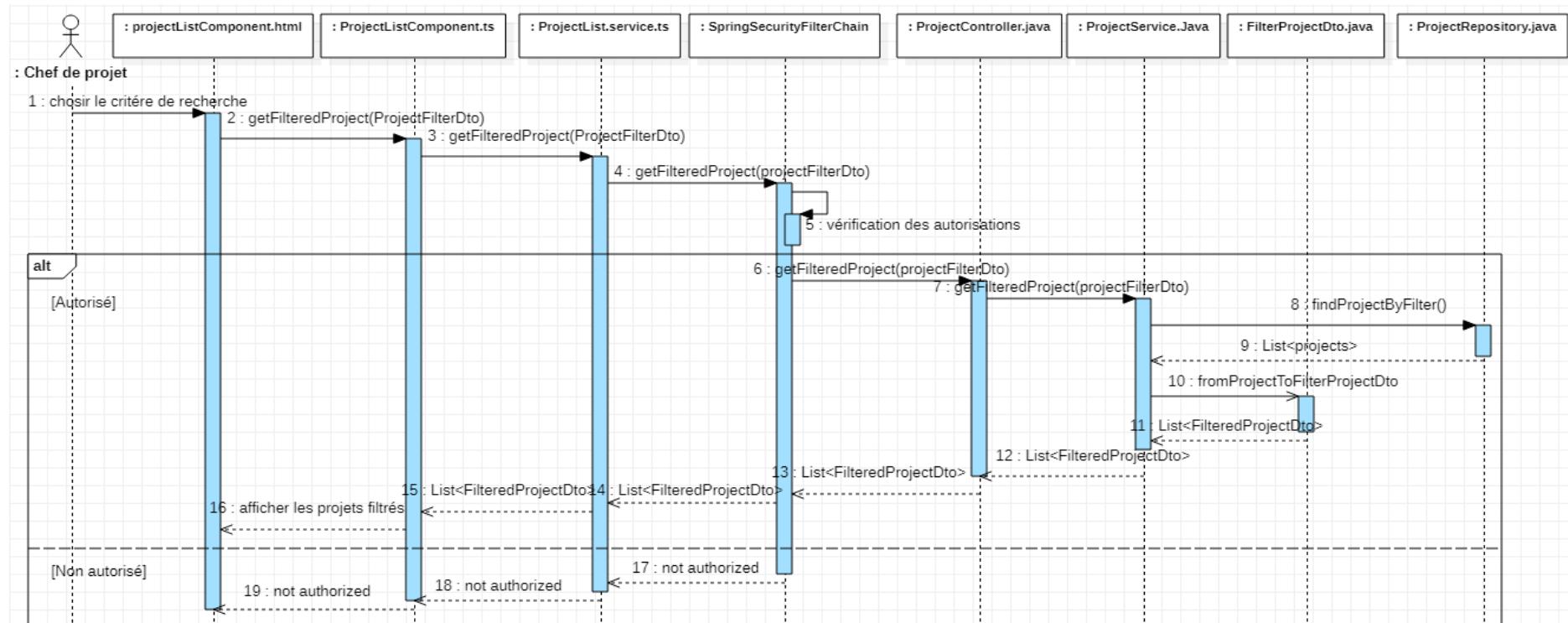


Figure 37 : Diagramme de séquence du cas d'utilisation « Filtrer la liste des projets »

Le diagramme de séquence du cas d'utilisation « Consulter détails d'un projet » et le diagramme d'activité du cas d'utilisation « Modifier un projet » sont présentés respectivement en annexe par la figure 106 et 107.

## 2.4. Réalisation

Nous illustrons le travail que nous avons effectué durant ce sprint par des captures écran.

### 2.4.1. Interface de création d'un projet

La figure 38 présente l'interface d'ajout de projet et l'affectation d'un chef de projet. Comme nous avons cité dans la section précédente, seul l'administrateur possède ce droit d'accès à l'interface, nous illustrons par la suite, quelques cas d'erreur du cas d'utilisation « ajouter un projet ».

The screenshot shows the 'Create Project' dialog box. On the left, the main form contains fields for 'Projet Name' (Développement D'une application mobile de gestion des cartes de fidélités), 'started Date' (05/08/2021 16:41), 'end Date' (08/10/2021 16:41), 'Add files' (Sélect. fichiers travail-demandé.pdf), and 'Description' (L'objectif principal de ce travail de PFE consiste à implémenter une solution mobile basée sur la plate forme J2ME pour la gestion des cartes de fidélité. Cette Application permettra aux clients de conserver toutes leurs cartes de fidélités sur leurs téléphones, consulter leurs soldes des points et géo localiser les enseignes dans lesquelles ils pourraient utiliser ses cartes.). At the bottom are 'Add' and 'Cancel' buttons. On the right, a sidebar titled 'Assign Project to project leader' shows a search bar with 'fersi' and a list of two users: 'ADEM FERSI' and 'CHEDI FERSI'. A callout box labeled 'Recherche filtré des chefs de projet' points to the search bar.

Figure 38 : Interface du cas d'utilisation « Créer un projet »

Pour valider la création d'un projet, le système doit respecter des validateurs, sinon des cas d'erreurs vont être déclenchés. Nous présentons des exemples de cas d'erreur dans la partie annexe du sprint 3 (figure 108).

## 2.4.2. Interface de consultation du détail d'un projet

L'interface du détail de projet est illustrée par la figure 39.

The screenshot shows the 'Smart Travel Guide Application' project detail page. At the top, there's a profile picture of the project leader, Adem Fersi, with his name and email (ademferci@gmail.com) below it. A progress bar indicates 50% completion. The project description states: 'The main objective of this Android project is to provide rich and relevant travel-related information to users on the move. The idea is to create a smart travel guide app that users can access anytime and from anywhere. Since the proposed app is based on request and response, it does not require the continual acquisition of bandwidth.' Below the description, there are tabs for 'Teams' and 'Files'. The 'Teams' tab is active, showing three teams: 'Full Stack' (Team Lead: wissal fersi, contact: 53089499, wissalfersi@gmail.com), 'Test' (Team Lead: radhouan maghraoui, contact: 986683145, radhouanMaghi@gmail.com), and 'Quality' (Team Lead: Yasmine Ben Slimene, contact: 53084498, yasmineSlimen@gmail.com). An orange button at the top right says 'Edit Project'.

Figure 39 : Interface du détail projet

## 2.4.3. Opérations sur la liste des projets

Ayant le rôle administrateur ou chef de projet, l'utilisateur connecté peut appliquer plusieurs opérations sur la liste des projets. Ces opérations seront illustrées par des figures successivement.

- **Modifier le statut d'un projet**

Un projet peut avoir quatre statuts à savoir « stopped, pending, progressing, completed ».

Un chef de projet ou un administrateur aura le droit de choisir trois état qui sont « stopped, pending, progressig » en cliquant sur l'un des icônes de la liste.

Le statut « completed » est mis à jour automatiquement si toutes les tâches du projet ont le statut « Done ».

The screenshot shows a project management application interface. A modal dialog box titled 'Alert' with a yellow exclamation mark icon asks 'Are you sure to update status to STOPPED?'. There are 'No' and 'Yes' buttons at the bottom of the dialog. In the background, a table lists three projects: 'Food spoilage detector' (Status: completed, Progress: 100%, green bar), 'test notif' (Status: pending, Progress: 0%, grey bar), and 'gestion de projet' (Status: processing, Progress: 0%, blue bar). The table has columns for #, Projects, Start Date, Due Date, Status, Team, Progress, and Action. The 'Action' column contains icons for edit, delete, and other project management tasks. Navigation buttons 'Previous' and 'Next' are at the bottom of the table.

Figure 40 : Mettre un projet à l'état « Stopped »

- **Rechercher un projet**

La recherche d'un projet peut être effectuée par plusieurs filtre :

- Recherche par nom

Un administrateur ou un chef de projet peut saisir le nom d'un projet à rechercher et le système l'affiche.

#	Projects	Start Date	Due Date	Status	Team	Progress	Action
1	Computer Game Companion App	11 July,2021	07 August,2021	processing		Progress 50%	

- Recherche par période

L'administrateur ou le chef de projet peut saisir deux dates et le système affiche les projets qui sont présents entre ces deux dates. Les projets sont triés par « started date » du projet.

#	Projects	Start Date	Due Date	Status	Team	Progress	Action
1	Food spoilage detector	10 July,2021	25 July,2021	completed		Progress 100%	
2	Project Management Dashboard	11 July,2021	30 July,2021	processing		Progress 66.66667%	
3	Drawing tool	12 July,2021	06 July,2021	Stopped		Progress 0%	

- Recherche par statut :

Un projet peut avoir quatre statuts, l'administrateur et/ou le chef de projet saisit un statut et le système affiche la liste de projets ayant le statut saisi.

#	Projects	Start Date	Due Date	Status	Team	Progress	Action
1	Food spoilage detector	10 July,2021	25 July,2021	completed		Progress 100%	
2	design and develop a mobile application	12 July,2021	29 August,2021	completed		Progress 100%	

Figure 41 : Recherche d'un projet

#### 2.4.4. Interface de consultation de la liste des projets/chef de projet

Une fois le projet est créé, il aura le statut « pending » et il reste en attente de changement de statut par le chef de projet. En se connectant à son compte, ce dernier aura une interface de consultation de ses projets en état d'attente, comme le montre les interfaces suivantes :

Figure 42 : Interface des projets en attente

- La liste des projets approuvés par le chef de projet est illustrée par la figure 43.

Figure 43 : Interface de la liste de projet assignés au chef de projet

#### 2.4.5. Interface de création d'une équipe

Pour créer une équipe, le chef de projet peut chercher un chef d'équipe et les membres par les opérations présentées par la figure 45.

- Filtrage de la liste des utilisateurs ayant rôle chef d'équipe.

- Filtrage de la liste des utilisateurs ayant rôle membre.

Figure 44 : Interface de création d'une équipe

#### 2.4.6. Interface de consultation des équipes

Un utilisateur connecté et ayant le rôle membre peut consulter la liste de toutes les équipes qu'il appartient.

Figure 45 : Liste des équipes

### 3. Sprint 4 : Gestion et planification des tâches

Le projet a un ensemble de tâches à réaliser, ce sprint est consacré à la gestion des tâches, pour se faire nous commençons par la présentation du backlog de produit du sprint, ensuite, nous entamons la partie conception et nous finissons par la présentation des interfaces.

#### 3.1. Backlog de produit du sprint

Le tableau 25 représente le backlog de produit du sprint intitulé « Gestion et planification des tâches » qui englobe le processus métier de la gestion et la planification des tâches des membres.

Tableau 25 : Backlog de produit du sprint numéro 3

ID Module	Module	ID User Story	User Story
1.	Gestion Et planification des tâches	1.	En tant que chef d'équipe/membre, je veux ajouter une tâche au tableau des tâches.
		2.	En tant que chef d'équipe et/ou membre de l'équipe, je veux consulter la liste de mes tâches.
		3.	En tant que chef d'équipe et/ou membre de l'équipe, je veux modifier le statut d'une tâche avec la technique glisser/déplacer.
		4.	En tant que chef d'équipe, je veux supprimer une tâche assignée à un membre.
		5.	En tant que chef d'équipe/chef de projet/administrateur, je veux consulter les tâches d'un membre d'équipe entre deux dates et leurs états d'avancement.
		6.	A la création d'une tâche, en tant que chef d'équipe, je veux consulter les tâches qui sont en cours dans cette période et je veux modifier la date de livraison d'une autre tâche.
		7.	En tant que membre d'équipe et/ou chef d'équipe, je veux consulter les détails d'une tâche.
		8.	En tant que chef d'équipe et/ou membre de l'équipe, je veux charger/télécharger des fichiers.
		9.	En tant que membre de l'équipe et/ou chef d'équipe, je veux ajouter/supprimer/modifier des commentaires sur chaque tâche.

## 3.2 Analyse

Cette partie a pour objectif de donner une vue d'analyse sur les exigences du quatrième sprint. Nous décrivons les acteurs ainsi que leurs interactions avec les cas d'utilisation, puis nous passons à la présentation de quelques diagrammes de séquence.

### 3.2.1. Diagramme de cas d'utilisation

La figure 46 illustre le diagramme de cas d'utilisation du sprint « Gestion et planification des tâches » qui décrit les fonctionnalités à envisager dans cette partie.

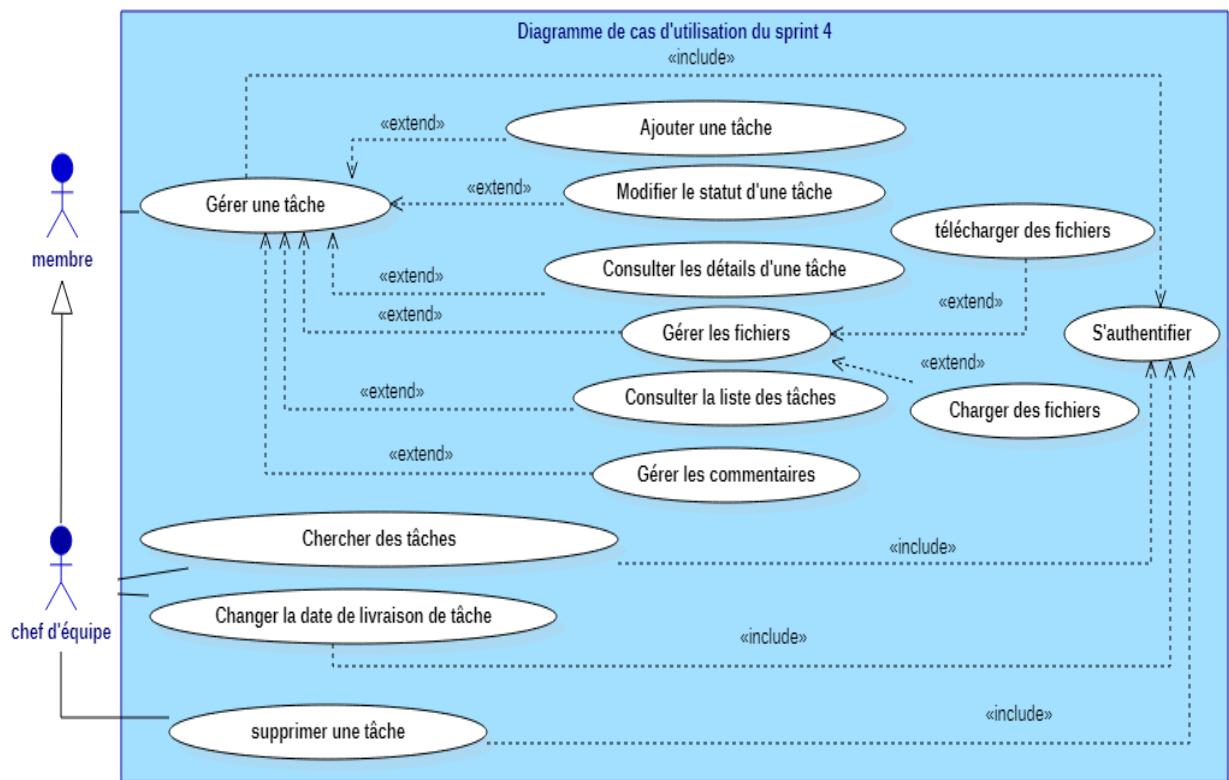


Figure 46 : Diagramme des cas d'utilisation du sprint numéro 4

### 3.2.2. Raffinement des cas d'utilisation

Dans la présente section, nous allons présenter le raffinement des cas d'utilisation en présentant le diagramme de cas d'utilisation raffiné « Gérer les commentaires » et la description textuelle du cas d'utilisation « Chercher des tâches » et « Ajouter une tâche ».

#### - Diagramme du cas d'utilisation « Gérer les commentaires »

Nous présentons par la figure 48 le raffinement du cas d'utilisation « Gérer les commentaires ».

L'acteur utilisateur peut être un chef d'équipe ou un membre.

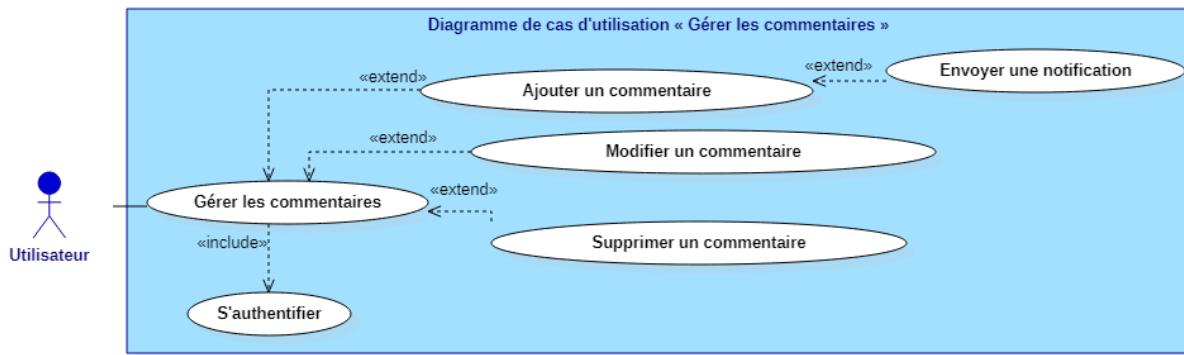


Figure 47 : Raffinement du cas d'utilisation « Gérer les commentaires »

#### - Description textuelle du cas d'utilisation « Chercher des tâches »

Le tableau 26 illustre le raffinement du cas d'utilisation « Chercher des tâches », en présentant l'acteur de la fonctionnalité, les conditions, le scénario nominal et les exceptions de cette fonctionnalité.

Tableau 26 : Description textuelle du cas d'utilisation « Chercher des tâches »

Raffinement du cas d'utilisation	
Cas d'utilisation	Chercher des tâches
Acteur	Chef d'équipe
Objectif	Consulter la liste des tâches d'un membre entre deux dates.
Résumé	Le chef d'équipe sélectionne un membre d'équipe, remplit un formulaire pour visualiser ses tâches entre deux dates avec un calcul des nombre de tâche et le nombre de tâche ayant le statut « done ».
Conditions	
Préconditions	Post Conditions
- L'utilisateur doit être connecté et ayant le rôle chef d'équipe d'une équipe donnée.	La liste des tâches et les statistiques sont affichés.
Scénario	
1- Le chef d'équipe clique sur la liste des membres de l'équipe. 2- Le système affiche la liste des membres. 3- Le chef d'équipe choisit un membre. 4- Le chef d'équipe saisit la période de recherche.	

- |  |
|--|
| 5- Le chef d'équipe valide son choix.      |
| 6- Le système affiche la liste des tâches. |

**Exception**

- |   |
|---|
| 1-Si l'un des champs obligatoire est vide un message d'erreur sera affiché. |
|---|

**- Description textuelle du cas d'utilisation « Ajouter une tâche »**

Une description textuelle du cas d'utilisation « Ajouter une tâche » est illustrée par le tableau 27.

**Tableau 27 : Description textuelle du cas d'utilisation « Ajouter une tâche »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Ajouter une tâche
<b>Acteur</b>	Chef d'équipe/membre d'équipe
<b>Objectif</b>	Assigner une tâche à un membre.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
<ul style="list-style-type: none"> <li>- L'utilisateur doit être connecté et ayant le rôle chef d'équipe/membre d'équipe.</li> <li>- L'utilisateur doit être connecté et ayant le rôle membre d'une équipe donnée.</li> </ul>	La tâche est créé et assignée à un membre d'équipe.
<b>Scénario</b>	
1- Le chef d'équipe ou le membre d'équipe clique sur le bouton « Add New Task ». 2- Le système affiche un formulaire. 3- Le chef d'équipe ou le membre saisit les données. 4- Le système vérifie que le membre d'équipe n'a pas d'autres tâche pendant cette période. 5- Le système affiche un message de succès.	
<b>Scénario alternatif</b>	
1- Le chef d'équipe ou le membre d'équipe clique sur le bouton « Add New Task ». 2- Le système affiche un formulaire. 3- Le chef d'équipe ou le membre saisit les données. 4- Le système vérifie si le membre d'équipe a d'autre tâche pendant cette période. 5- Le système affiche les tâches dont leurs dates de livraison sont pendant la période d tâche à créer. 6- Le chef d'équipe change la date de début et de livraison de la tâche à créer.	

7- Le système affiche un message de succès.

### Exception

1-Si l'un des champs obligatoire est vide un message d'erreur sera affiché.

## 3.3. Conception

Nous passons maintenant à la conception du quatrième sprint. Pour ce faire, nous présentons au premier lieu le diagramme de classes de ce sprint et nous exposons en second lieu le diagramme de séquence de quelques cas d'utilisation.

### 3.3.1. Diagramme de classes

La modélisation du diagramme de classes du sprint 4 est présenté par la figure 49.

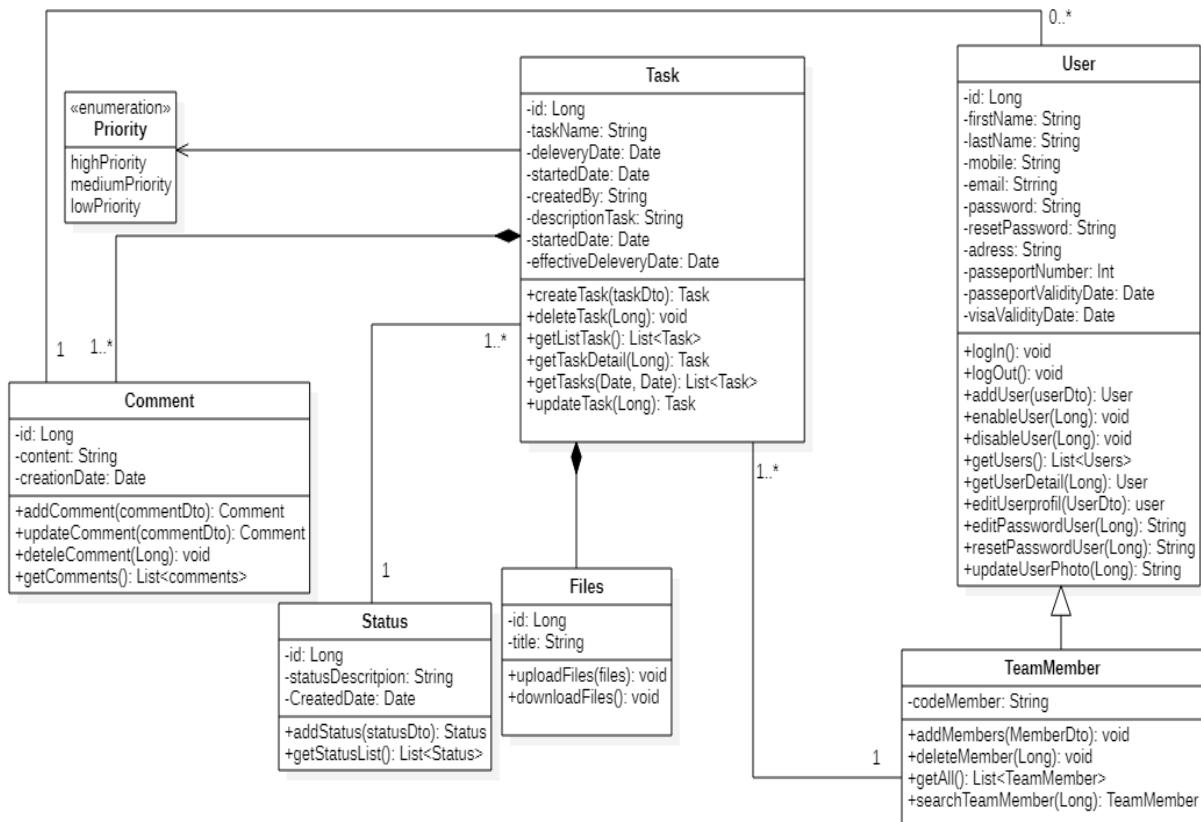


Figure 48 : Diagramme de classes du sprint 4

En plus que la classe « User » et la classe « TeamMember », le diagramme de classes du sprint 4 présente 5 classes supplémentaires qui sont :

- **Classe Task** : La classe « Task » représente la classe principale de ce sprint. Elle offre 6 méthodes, citons la méthode *createTask(taskDto)* qui permet de créer une tâche, la méthode *getListTask(Long)* permettant d'afficher la liste des tâches d'un membre, etc.

- **Classe Priority** : C'est une classe d'énumération contenant 3 attributs représentant les différentes priorités qui peuvent être accordées à une tâche telles que : (*highPriority*), (*lowPriority*) et (*mediumPriority*).
- **Classe Files** : C'est une composition liée à la classe « Task » et comporte deux méthodes qui sont : *downloadFiles()* et *uploadFiles()* permettent respectivement de charger et télécharger des fichiers.
- **Classe Comment**: C'est une composition liée à la classe « Task » et comporte 4 méthodes qui sont : *addComment(commentDto)* pour ajouter un commentaire à une tâche, *deleteComment(Long)* afin de supprimer un commentaire, *updateComment(commentDto)* pour mettre à jour un commentaire et *getComments()* pour consulter la liste des commentaires liée à une tâche.
- **Classe Status** : Cette classe offre deux méthodes qui sont *addStatus(statusDto)* pour ajouter un statut à une tâche et *getListStatus()* pour consulter la liste des statuts.

### 3.3.2. Diagramme de séquence

Après avoir expliqué le comportement statique du système à travers le diagramme de classes, nous exposons dans cette section le diagramme de séquence du cas d'utilisation « Charger des fichiers » et le diagramme de séquence du cas d'utilisation « Modifier le statut d'une tâche ».

#### - Diagramme de séquence du cas d'utilisation « Charger des fichiers »

La figure 49 présente le diagramme de séquence du cas d'utilisation « Charger des fichiers ».

#### - Diagramme de séquence du cas d'utilisation « Modifier le statut d'une tâche »

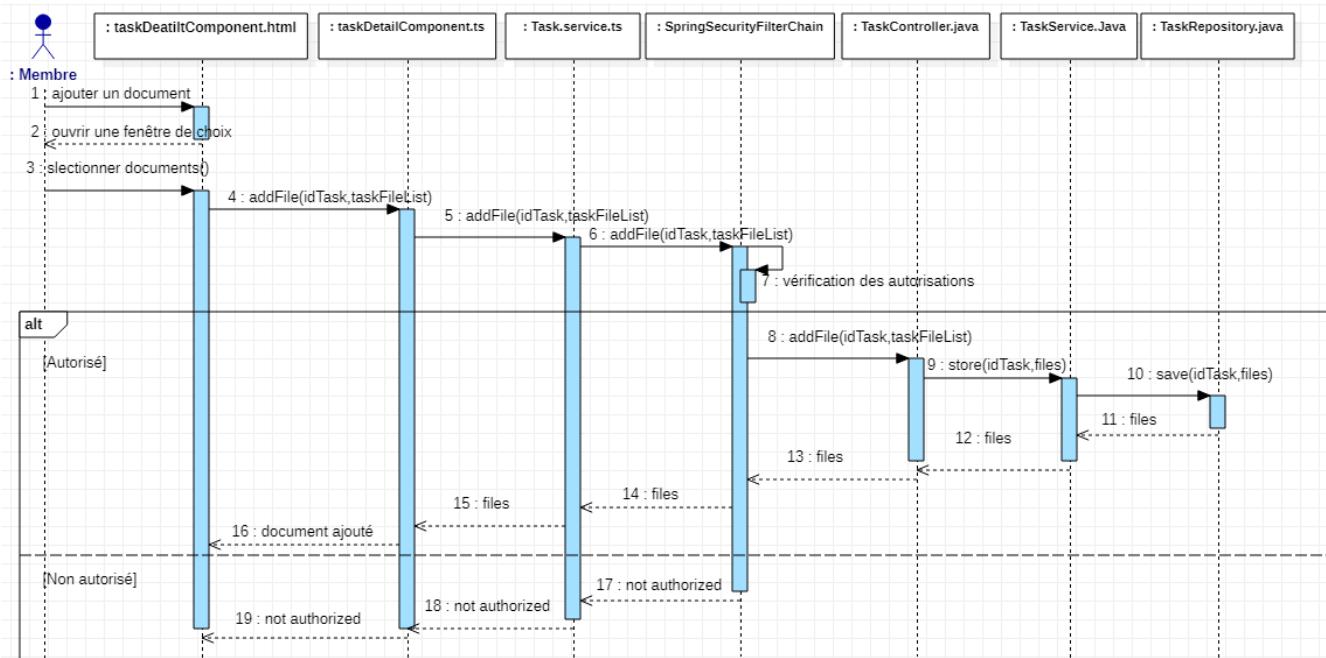


Figure 49 : Diagramme de séquence du cas d'utilisation « Charger des fichiers »

La figure 50 présente le diagramme de séquence du cas d'utilisation « Modifier le statut d'une tâche ».

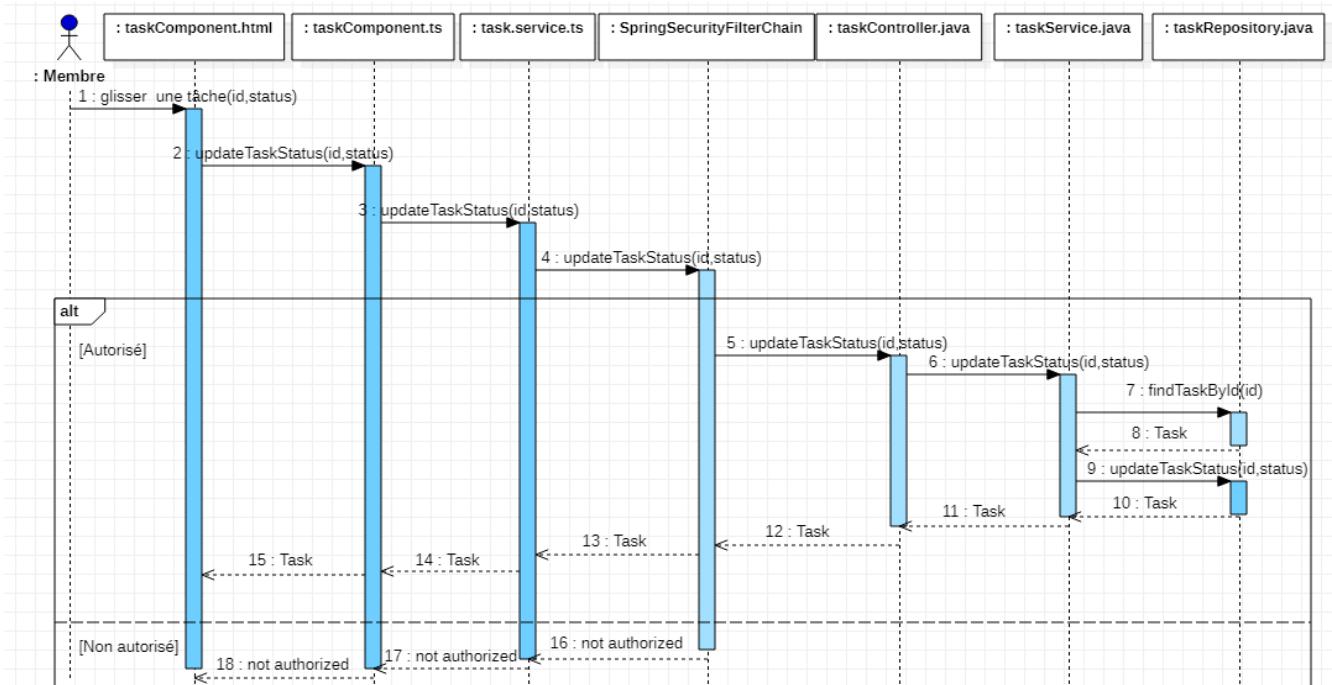


Figure 50 : Diagramme de séquence du cas d'utilisation « Modifier le statut d'une tâche »

## 3.4. Réalisation

Nous exposerons dans cette section les interfaces homme-machine du sprint 4 en détaillant quelques fonctionnalités.

### 3.4.1. Interface d'ajout d'une tâche

Pour ajouter une tâche, le chef d'équipe doit remplir le formulaire illustré par la figure 51. En effet, chaque tâche être affectée à une membre et doit avoir une priorité, un statut et une date de début et de livraison.

Figure 51 : Formulaire d'ajout d'une tâche

### 3.4.2. Interface de consultation de la liste des tâches

En accédant à son espace, un membre peut consulter sa liste des tâches. Ainsi, il peut modifier l'état d'une tâche par la technique glisser/déplacer (figure 52).

The screenshot shows a Kanban board titled "Team / Member's Tasks". At the top, there are status bars for each column: "To-do" (22 September, 21 9:28 am => 09 October, 21 2:28 am), "Doing" (22 September, 21 9:28 am => 09 October, 21 2:28 am), "Test" (22 September, 21 9:28 am => 09 October, 21 2:28 am), and "Done" (22 September, 21 9:28 am => 09 October, 21 2:28 am). The "To-do" column contains one task: "model classes with a class diagram" (due 02 October, 21 8:27 pm). The "Doing" column contains two tasks: "Uses case diagram" (due 28 September, 21 3:25 pm) and "develop automated test" (due 29 September, 21 8:23 pm). The "Test" column contains one task: "Develop automated test for interface 'Us'" (due 29 September, 21 8:23 pm). The "Done" column contains one task: "Develop specification" (due 26 September, 21 10:17 pm). Below the columns, there is a separate section for a task: "create backlog user story" (due 28 September, 21 9:19 am). On the left, there is a sidebar with buttons for "Full stack", "More Detail", "Add New task", "To-do", "Doing", "Test", "Done", and "New Status".

**Figure 52 : Interface de la liste des tâches d'un membre**

- En cliquant sur le bouton « Add new » un membre et/ou un chef d'équipe peut ajouter une tâche avec le statut « to-do » comme le montre la figure 53.

The screenshot shows a modal dialog box titled "add task with ' To-do ' status". The form contains the following fields:
 

- Title :** title
- Priority :** (dropdown menu)
- Started Date :** jj/mm/aaaa --:-- (date picker)
- Due Date :** jj/mm/aaaa --:-- (date picker)
- Description :** Add description to task Here (text area)

 At the bottom, there are two buttons: "Save Task" (orange) and "Cancel" (red).

**Figure 53 : Ajouter une tâche au statut « To-do »**

### 3.4.3. Interface de consultation des détails d'une tâche

En cliquant sur l'icône « view » d'une tâche de la liste, le système redirige le membre à la page détail tâche, le membre peut consulter les fichiers téléchargés, charger des fichiers ainsi faire des commentaires comme l'indique la figure 54.

The screenshot shows the 'Task List / Task Detail' interface. At the top left is the member's profile picture and name 'elaa fersi' with a 'highPriority' badge. To the right is the title 'Conception du diagramme des cas d'utilisation'. Below the title is a brief description: 'La conception est une phase charnière entre la spécification du problème et l'implantation de la solution. Il est donc important de garder la trace des décisions de conception. Merci de les uploader...'. On the left, there are sections for 'Start Date' (11 August, 21 3:25 am), 'Due Date' (13 August, 21 3:25 am), 'Created By' (Wissal ferfi), and 'Attached Files'. The 'Attached Files' section displays nine document icons with names like 'Architecture ...', 'backlog.pdf', 'Cahier des ...', 'Récap2.docx', etc. On the right, there is a 'Comments (3)' section with three entries from users elaa fersi, Wissal ferfi, and elaa fersi, each with 'Reply', 'Update', and 'Remove' links.

Figure 54 : Consultation des détails d'une tâche

### 3.4.4. Interface d'ajout des fichiers

Le membre et ou le chef d'équipe peut télécharger et charger des fichiers à partir de l'interface 55.

The screenshot shows the 'Attached Files' interface. At the top, it says 'Uploading... 24'. Below that is a button labeled 'Select fichiers' with the file path 'OCA-Day 2 (2021-05-09 at 01\_15 GMT-7).mp4'. Below the button is a grid of six document icons with names like 'OCA\_Day1\_A...', 'OCA\_Day1\_B...', etc. A '+' icon is located in the bottom right corner of the grid.

Figure 55 : Interface d'ajout des fichiers

### 3.4.5. Interface de consultation des tâches entre deux dates

Un chef d'équipe peut consulter le nombre des tâches affectées à un membre entre deux dates données et le nombre des tâches ayant le statut « Done », l'interface présente ainsi les détails des tâche (date de début/date de fin/statut)

L'interface 56 présente les tâches du membre « Wissal Fersi » depuis 01/07/2021 jusqu'à 30/07/2021.

The screenshot shows a search interface at the top with fields for 'Member tasks between two date:', 'Choose team member : Wissal ferci', 'From: 07/07/2021 04:14', 'To: 31/07/2021 04:14', and a 'Search' button. Below the search bar is a summary section with icons for 'Number of Task' (3) and 'Done Task' (2). The main area displays three tasks:

- create Ideas Tip Sheet**: Status 'Todo', Started Date: 14 July, 21 9:47 PM, Due Date: 15 July, 21 9:48 PM
- Design diagam class**: Status 'Done', Started Date: 17 July, 21 9:50 PM, Due Date: 18 July, 21 9:50 PM
- Design diagam class**: Status 'Done', Started Date: 17 July, 21 9:50 PM, Due Date: 18 July, 21 9:50 PM

Figure 56 : Consultation des tâches entre deux date

### 3.4.6. Interface de modification de la date de livraison de tâche

A la création d'une tâche et si la date effectuée à la tâche est non disponible, le chef d'équipe peut consulter les tâches qui sont en cours dans cette période et peut modifier la date de livraison d'une autre tâche moins prioritaire, comme le montre la figure 58.

The screenshot shows a task list with two items:

- design a diagrass class** - Team: Test task team, Due Date: 15 July, 21 10:29 AM. A calendar overlay is open, showing July 2021 with the 15th selected. Buttons for 'Change Due date' are visible.
- create a logo for application** - Team: Test task team, Due Date: 15 July, 21 4:36 AM. A 'Change Due date' button is visible.

Figure 57 : Modification de la date de livraison d'une tâche en cours

Les tests unitaires, l’analyse de la qualité du code et la documentation de cette release sont représentés annexe par les figures 110, 111 et 112.

## Conclusion

Nous avons consacré ce chapitre à La description du deuxième release. En effet, nous avons commencé par décrire le backlog de produit de chaque sprint, puis nous avons élaboré le diagramme de cas d’utilisation raffiné qui présente les principales fonctionnalités à réaliser durant chaque sprint avec une description textuelle de quelques cas d’utilisation. Après, nous avons passé à la phase de conception en présentant une vue statique et une vue dynamique du système. Finalement, nous avons présenté un aperçu sur les interfaces qui démontrent la réalisation de notre travail.

# **Chapitre 5 : Release 3 « Gestion des évènements, statistiques et déploiement »**

## **Introduction**

Après avoir développé les fonctionnalités du second release, nous passons maintenant au développement d'autres fonctionnalités à savoir la gestion des évènements, la gestion des notifications et la gestion des statistiques. Ce chapitre illustre le cycle de vie du troisième release, qui englobera trois sprint, en commençant par le backlog de produit de chacun, tout en passant par la spécification fonctionnelle, la conception ainsi que la réalisation.

### **1. Présentation de la release 3**

Le but de cette release est de créer un calendrier afin de gérer les évènements de chaque équipe, nous allons rajouter des fonctionnalités pour la gestion des notifications et des rappels, les membres d'équipe peuvent discuter en temps réel en utilisant le protocole « Web Socket ».

Le troisième objectif de cette release consiste à créer des tableaux de bord pour la gestion des statistiques des projets, des équipes et des membres, Finalement nous allons clôturer avec le troisième sprint qui englobera la partie test, intégration et déploiement.

Nous avons décomposé notre release en trois sprints principaux :

- Sprint 5 : Gestion des évènements et des notifications
- Sprint 6 : Gestion des tableaux de bord
- Sprint 7 : Test, déploiement et mise en place d'une chaîne CI/CD

Nous commençons par la description des fonctionnalités de chaque sprint, suivie de l'analyse et la conception pour enfin présenter la partie réalisation.

### **2. Sprint 5 : Gestion des événements, des notifications et des rappels**

Le présent sprint se focalise sur la gestion des évènements par équipe et gestion des notifications. Un événement représente une réunion ou une activité survenue au cours du travail.

## 2.1. Présentation des WebSockets

Websocket est une technologie qui permet d'ouvrir de façon permanente un canal de communication bidirectionnel entre un client et un serveur. Cette technologie repose sur le protocole TCP qui s'occupe du transport des données [29].

Les webSockets ont plusieurs points forts par rapport aux protocoles HTTP à savoir :

- Une connexion permanente entre le client (navigateur) et le serveur : évite de devoir se connecter/déconnecter à chaque nouvelle requête.
- Les requêtes sont plus rapides et plus légères.
- L'envoi de notifications du serveur au client, sans faire de requête au préalable.

## 2.2. Backlog de produit du sprint 5

Le tableau 28 représente le backlog de produit de notre premier sprint de ce chapitre « Gestion des événements, des notifications et des rappels » où nous listons les différentes exigences à mettre en place dans cette section.

Tableau 28 : Backlog de produit du sprint 5

ID Module	Module	ID User Story	User Story
1.	Gestion des événements	1.	En tant que chef d'équipe, je veux créer un événement.
		2.	En tant que chef d'équipe/ membre d'équipe, je veux consulter le calendrier des événements.
		3.	En tant que chef d'équipe, je veux modifier le jour de l'événement en utilisant la technique glisser/déplacer.
		4.	En tant que chef d'équipe, je veux modifier les détails d'un événement.
		5.	En tant que chef d'équipe, je veux annuler un événement.
		6.	En tant que chef d'équipe/membre, je veux consulter les détails d'un événement.
2.	Gestion des notifications, rappels et messagerie en temps réel	1.	En tant que chef de projet/chef d'équipe/membre, je veux recevoir des notifications relatives respectivement à mes projets/équipes/tâches.
		2.	En tant que membre, je veux recevoir des notifications relatives à la liste des événements.

		3.	En tant que membre, je veux recevoir un rappel le jour de la livraison d'une tâche ou le jour d'un événement.
		4.	En tant que chef de projet/ chef d'équipe/membre, je veux supprimer une notification.
		5.	En tant que chef de projet/ chef d'équipe/membre, je veux marquer une notification comme lue/non lue.
		6.	En tant que chef d'équipe/membre, je veux accéder à une « room chat » et discuter en temps réel avec mon équipe.

## 2.3. Analyse

Les exigences du premier sprint de cette release ont été décrites. Maintenant nous passons au diagramme de cas d'utilisation pour présenter leurs interactions avec les acteurs de notre application, puis nous réalisons quelques raffinements des cas d'utilisation et quelques diagrammes de séquence.

### 2.3.1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation du sprint 5 est illustré par la figure 58.

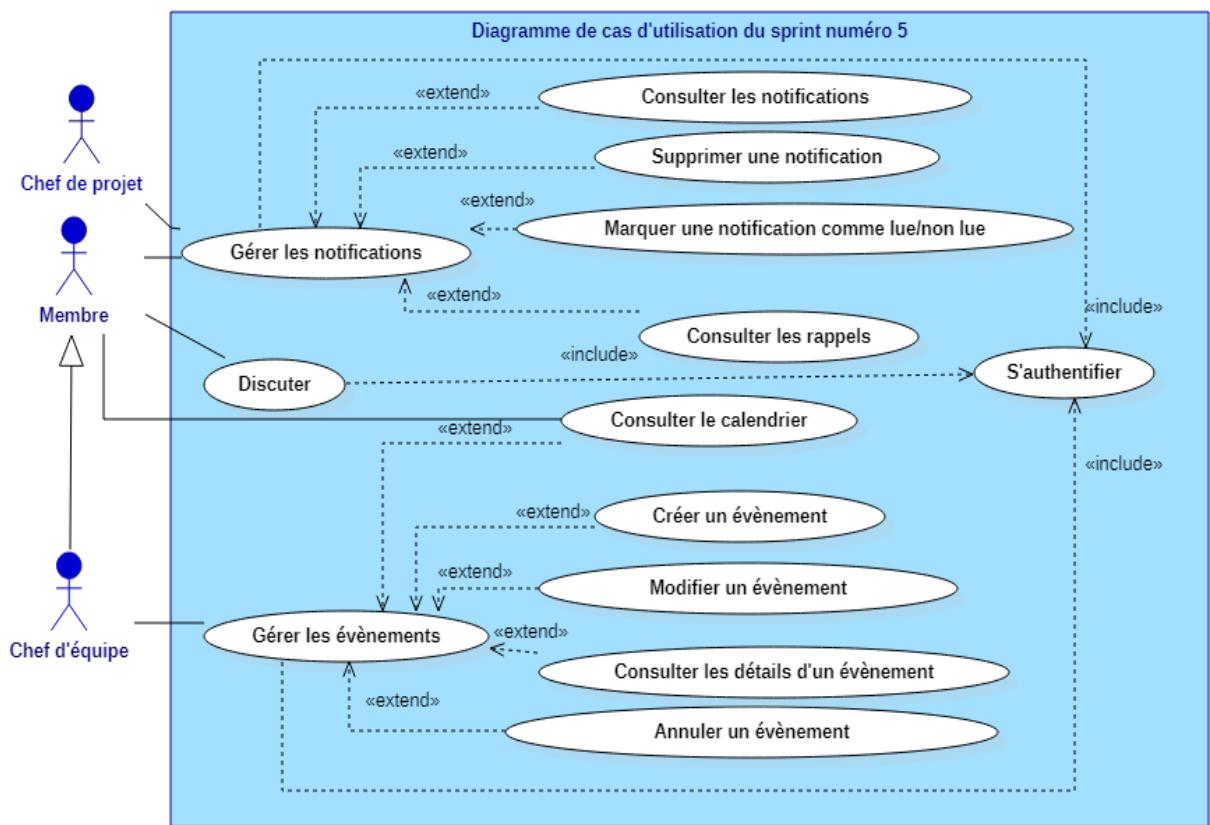


Figure 58 : Diagramme des cas d'utilisation du sprint 5

### 2.3.2. Raffinement des cas d'utilisation

Afin d'expliquer les scénarios, nous présentons la description textuelle de quelques cas d'utilisation.

- **Description textuelle du cas d'utilisation « Annuler un événement »**

Le tableau 29 illustre le raffinement du cas d'utilisation « Annuler un événement », en présentant l'acteur de la fonctionnalité, les conditions, le scénario nominal et les exceptions de cette fonctionnalité.

**Tableau 29 : Description textuelle du cas d'utilisation « Annuler un événement »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Annuler un événement
<b>Acteur</b>	Chef d'équipe
<b>Objectif</b>	Annuler un événement et le supprimer du calendrier.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
- L'utilisateur doit être connecté et ayant le rôle chef d'équipe d'une équipe donnée.	-L'évènement a été supprimé.
<b>Scénario</b>	
1- Le chef d'équipe choisit l'évènement à supprimer 2- Le système affiche un message de confirmation 3- Le chef d'équipe valide son choix 4- Le système vérifie l'existence de l'évènement 5- Le système supprime l'évènement 6- Le système affiche un message de succès	
<b>Scénario alternatif</b>	
1- Le chef d'équipe choisit l'évènement à supprimer. 2- Le système affiche un message de confirmation. 3- Le chef d'équipe annule son choix. 4- Le système annule la suppression.	

- **Description textuelle du cas d'utilisation « Marquer notification comme lue »**

Le tableau 30 illustre le raffinement du cas d'utilisation « Marque une notification comme lue ».

**Tableau 30 : Description textuelle du cas d'utilisation « Marquer notification comme lue »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Marquer notification comme lue
<b>Acteur</b>	Utilisateur
<b>Objectif</b>	Classer une notification comme lue
<b>Résumé</b>	Dans la liste des notifications, les notifications non lues sont affichées en premier lieu avec plus de visibilité, le but est de rendre moins visible une notification lue et décrémenter le nombre des nouvelles notifications .
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
- L'utilisateur doit être connecté	-Notification marquée comme lu et le nombre des notifications non lues est décrémentée de 1.
<b>Scénario</b>	
1- L'utilisateur clique sur l'icône des notifications. 2- Le système affiche une liste des dernières notifications. 3- L'utilisateur choisit une nouvelle notification non lue et clique sur « Marquer comme lue ». 4- Le système supprime l'icône non lue.	

- **Description textuelle du cas d'utilisation « Consulter des rappels »**

Le tableau 31 présente le raffinement du cas d'utilisation « Consulter la liste des rappels ».

**Tableau 31 : Description textuelle du cas d'utilisation « Consulter la liste des rappels »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Consulter la liste des rappels
<b>Acteur</b>	Membre/chef d'équipe
<b>Objectif</b>	Consulter la liste des rappels
<b>Résumé</b>	Si une équipe a une réunion dans des heures, la liste des rappels va être incrémentée et l'utilisateur connecté peut la consulter. Le jour de la livraison d'une tâche le membre d'une équipe va recevoir une notification.

Conditions	
Préconditions	Post Conditions
- L'utilisateur doit être connecté et ayant le rôle membre ou chef d'équipe.	
Scénario	
1- Le système incrémente le nombre des rappels automatiquement. 2- L'utilisateur clique sur l'icône rappels. 3- Le système affiche la liste rappels.	

## 2.4. Conception

Après avoir terminé la description textuelle des cas d'utilisation, nous allons nous focaliser, dans cette partie, sur une étude dynamique et statique présentée par le diagramme de classes relatif à la gestion des événements et le diagramme de classes relatif à la gestion des notifications et les diagrammes de séquence.

### 2.4.1. Diagramme de classes

Une vue statique va être présentée par le diagramme de classes relatif au sprint numéro 5. La figure 60 présentée décrit le système de classes utilisé pour le sprint 5, leurs attributs, les méthodes et les relations entre elles.

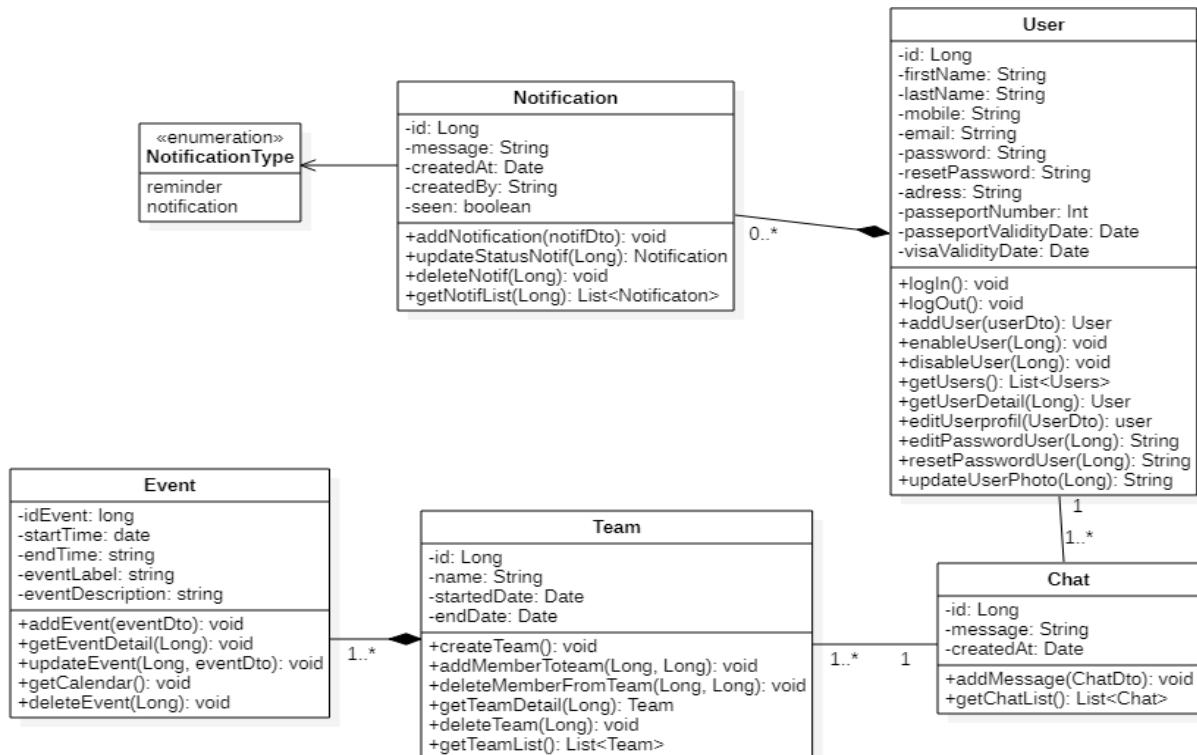


Figure 59 : Diagramme de classes du sprint 5

En plus que la classe « User » et la classe « Team » ce sprint présente quatre nouvelles classes qui sont :

- **Classe Notification** : C'est la classe principale pour la gestion des notifications, elle offre 4 méthodes, la méthode *addNotification(notifiDto)*, *deleteNotification(Long)* permettant respectivement de créer une notification et de la supprimer, la méthode *getNotifList(Long)* permettant de consulter la liste des notifications d'un utilisateur et la méthode *updateStatusNotif(Long)* permettant de modifier le statut d'une notification.
- **Classe NotificationType** : C'est une classe d'énumération, en effet, la notification peut avoir deux types différents, elle peut être une notification ou un rappel.
- **Classe Event** : C'est la classe principale pour la gestion des évènements, elle possède 4 méthodes, la méthode *addEvent(eventDto)* qui permet d'ajouter un nouvel évènement, *updateEvent (Long, eventDto)* pour modifier la date ou les détails d'un évènement, *deleteEvent(Long)* pour annuler un évènement, etc.
- **Classe Chat** : C'est la classe dédiée à la discussion instantanée. Elle offre 2 méthodes qui sont *addMessage(messageDto)* permettant d'envoyer un message au destinataire en temps réel et *getChatList()* pour consulter les messages de la discussion des membres d'une équipe.

#### 2.4.2. Diagramme de séquence

Nous allons présenter deux diagrammes de séquence relatifs au cas d'utilisation « Ajouter un évènement » et le diagramme de séquence « Consulter le calendrier ».

## - Diagramme de séquence du cas d'utilisation « Ajouter un évènement »

La figure 60 présente le diagramme de séquence du cas d'utilisation « Ajouter un évènement ».

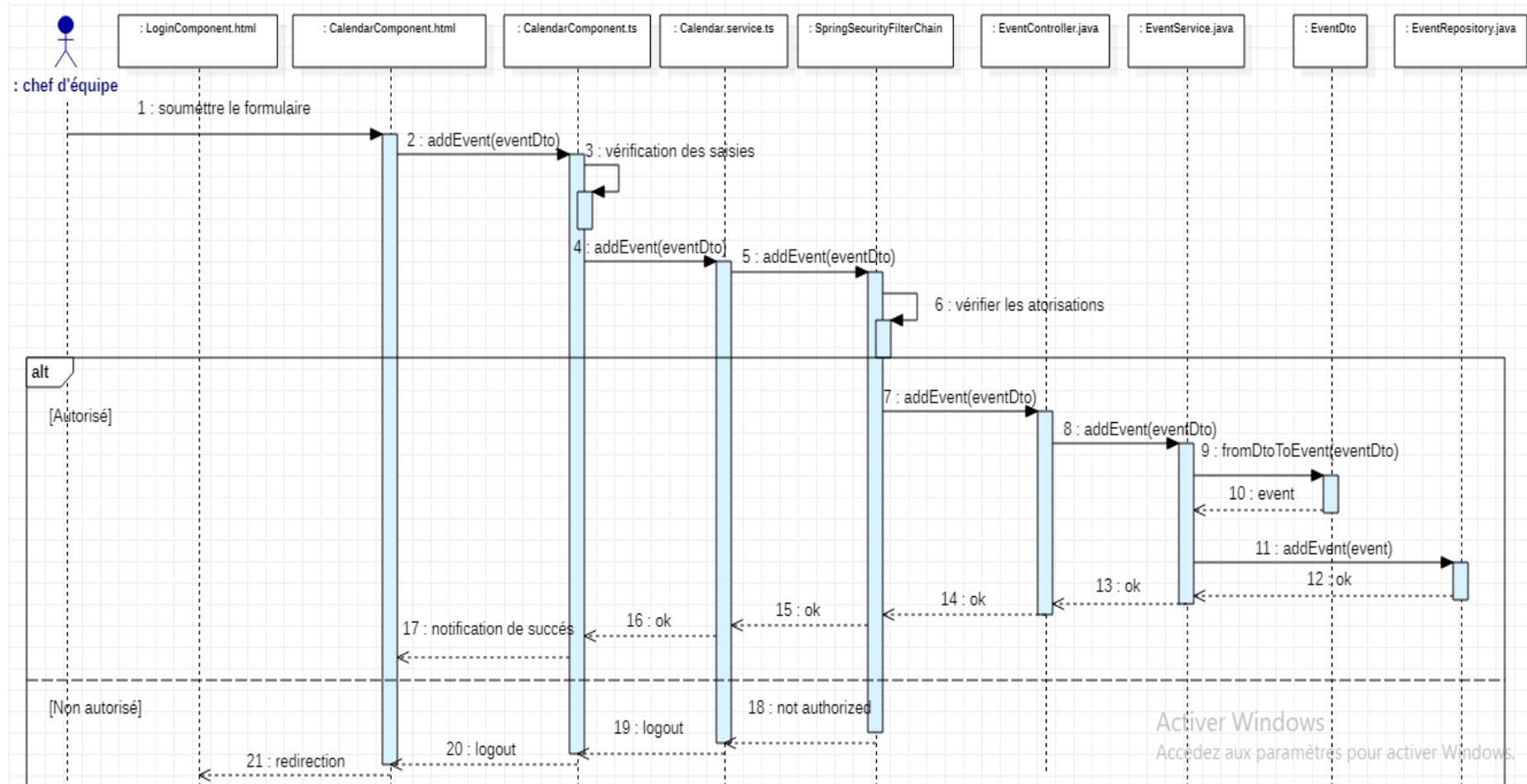


Figure 60 : Diagramme de séquence du cas d'utilisation « Ajouter événement »

## - Diagramme de séquence du cas d'utilisation « Consulter le calendrier »

Pour consulter le calendrier, le chef d'équipes et/ou les membres d'équipe doivent procéder au séquencement présenté par le diagramme de la figure 61.

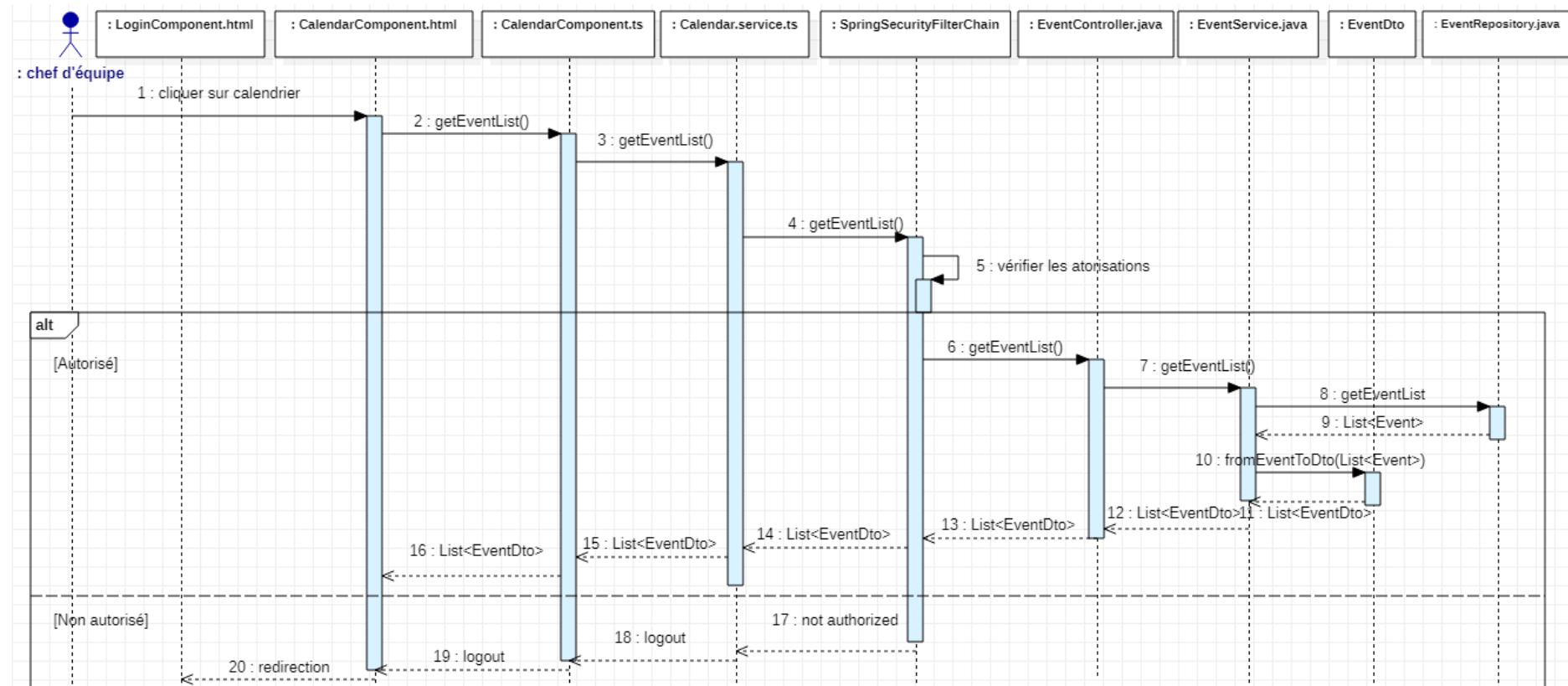


Figure 61 : Diagramme de séquence du cas d'utilisation « Consulter le calendrier »

Le diagramme d'activités du cas d'utilisation « Modifier un événement » est présenté en annexe par la figure 113.

## 2.5. Réalisation

Afin de tester notre sprint, nous présentons dans cette partie des figures qui illustrent les différentes interfaces réalisées.

### 2.5.1. Interface du calendrier

La figure 62 présente le calendrier des événements qui est accessible au chef d'équipe et à chaque membre d'équipe pour qu'il puisse suivre les différentes réunions et activités survenues.

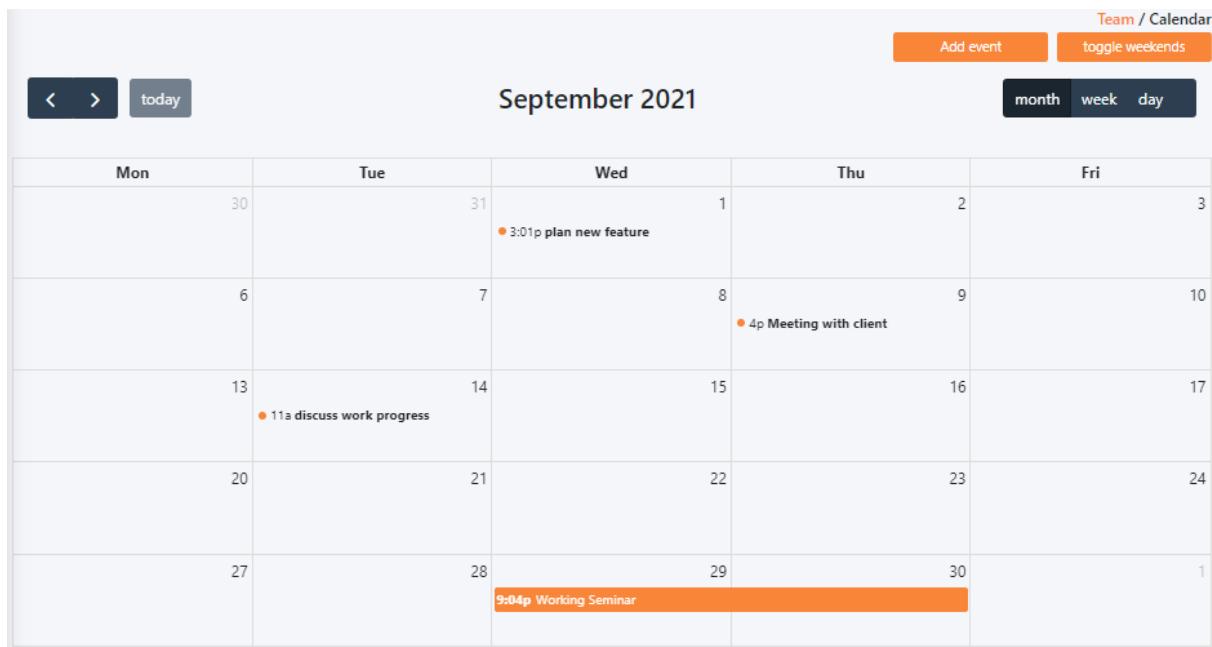


Figure 62 : Interface du calendrier

- En cliquant sur un événement, un formulaire affichant ses détails, les icônes de la mise à jour d'un événement ou de son annulation sont visibles uniquement par le chef d'équipe (figure 63).

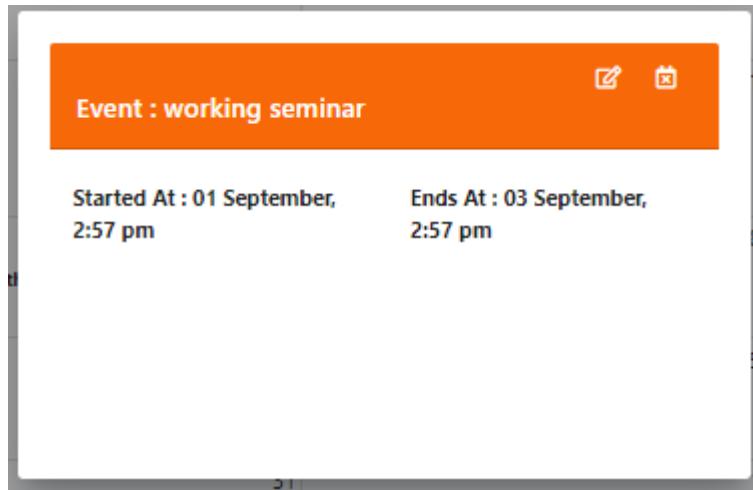


Figure 63 : Interface du détail d'un événement

### 2.5.2. Interfaces des opérations sur le calendrier

Les figures présentent respectivement le formulaire d'ajout d'un rendez-vous au calendrier (figure 64), le formulaire de la mise à jour d'un évènement ainsi l'opération de l'annulation d'un évènement.

The screenshot shows a modal dialog box titled "Add Event". At the top right is a close button (X). Below the title, the text "Event :" is followed by an empty input field. Underneath are two date/time input fields: "Started At :" and "Ends At :", each consisting of a text input field and a calendar icon. At the bottom of the dialog are two buttons: "Add" (orange) and "Cancel" (red).

Figure 64 : Interface d'ajout d'un événement

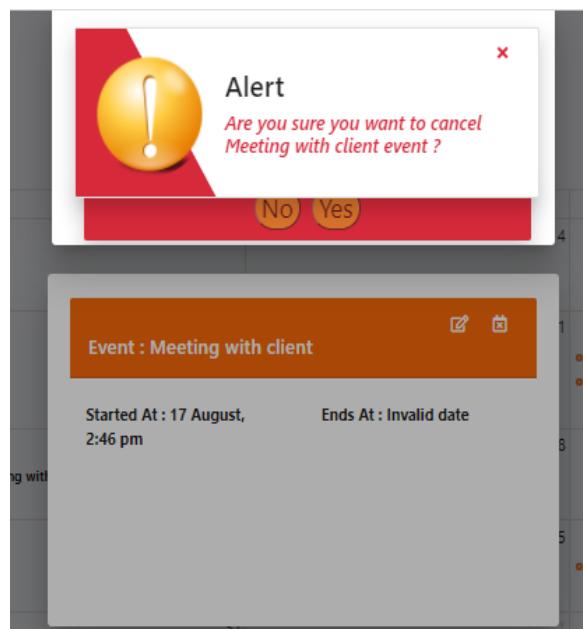
- Pour mettre à jour un événement, le chef d'équipe doit choisir un événement, modifier ses détails et valider son choix. Le formulaire de mise à jour est présenté par la figure 65.

The screenshot shows a modal dialog box titled "Event : Meeting with client". The title bar is orange. Inside, the text "Event :" is followed by an input field containing "Meeting with client". Below this are two rows of text: "Started At : 09 September, 4:00 pm" and "Ends At : 09 September, 5:00 pm". Under each row is a date/time input field with a calendar icon. At the bottom are two buttons: "Update" (orange) and "Cancel" (red).

Figure 65 : Interface de mise à jour d'un événement

- Le chef d'équipe peut également annuler un événement, le système affiche une alerte avant l'annulation de l'évènement pour confirmer le choix ou l'infirmier.

La figure 66 montre l'annulation de l'événement « Meeting with client ».



### 2.5.3. Interface de gestion des notifications et des rappels

Selon son rôle, chaque utilisateur possède un ensemble de notifications pour les grands événements relatifs à ses activités, et en cliquant sur l'icône « notification », il peut consulter la liste de ses notifications, modifier son statut, supprimer une notification (figure 67).

Ainsi, chaque membre d'équipe, peut recevoir des rappels s'il a une tâche à livrer le jour  $j$  ou un événement pour le même jour, les rappels sont cités par date (figure 68).

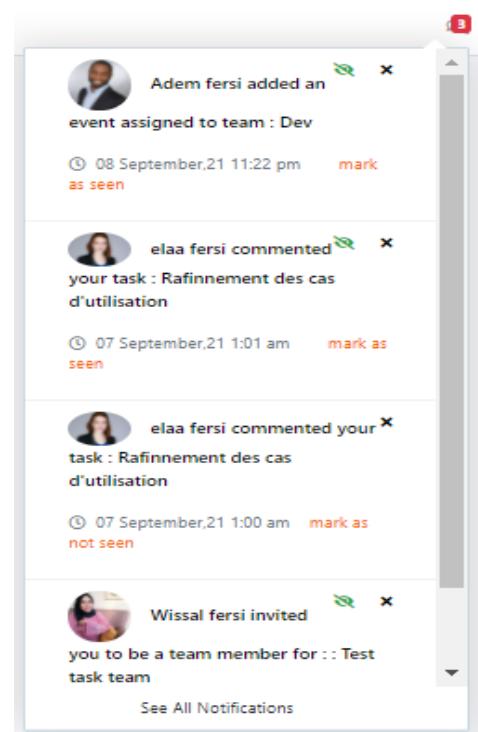
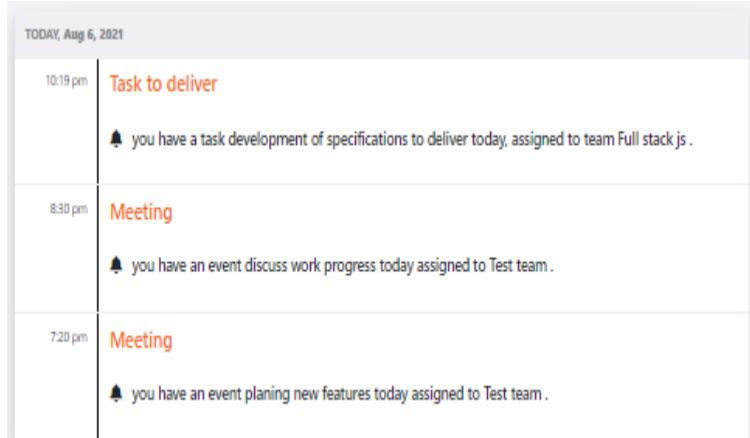


Figure 67 : Interface de la liste des rappels

Figure 68 : Interface de la liste des notifications

#### 2.5.4. Interface de la messagerie en temps réel

Chaque membre d'équipe peut envoyer des messages en temps réel aux autres membres de l'équipe en utilisant le protocole WebSocket. La figure 69 présente l'interface de la discussion instantanée entre le chef d'équipe et le membre de l'équipe.

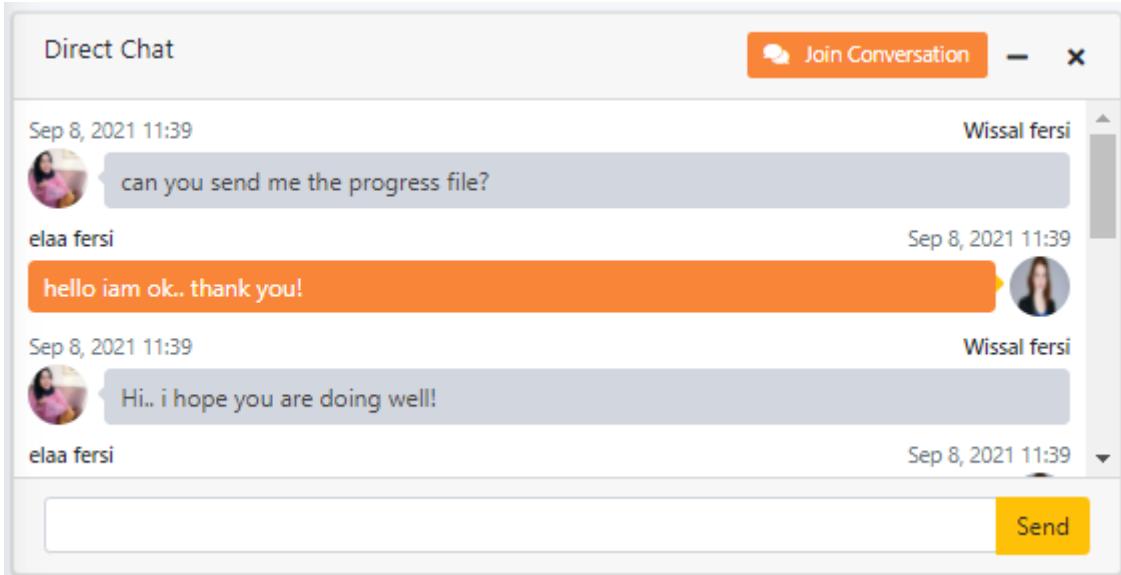


Figure 69 : Discussion instantanée entre deux membres

### 3. Sprint 6 : Gestion des tableaux de bord

Notre objectif est d'avoir une vue globale sur l'avancement du travail depuis le démarrage d'un projet donné jusqu'à sa fin. Dans ce sprint, nous allons calculer des indicateurs de performance et nous allons implémenter des tableaux de bord spécifiques pour l'administrateur, le chef de projet, le chef d'équipe et le membre.

Nous allons tenir compte des indicateurs de performances qui sont le taux de retard et le taux d'avancement.

#### 3.1. Backlog de produit du sprint 6

Le tableau 32 met en disposition le backlog de produit de ce présent sprint où les fonctionnalités sont détaillées.

Tableau 32 : Backlog de produit du sprint 6

ID Module	Module	ID User Story	User Story
		1.	En tant qu'administrateur, je veux consulter le nombre de projets avec le statut « completed », le nombre de projets « paused » et le nombre de projets « stopped ».

		2.	En tant qu'administrateur, je veux consulter le nombre de jours gagnés/perdus par rapport à la date de livraison pour chaque projet achevé.
		3.	En tant qu'administrateur, je veux consulter le nombre de jours gagnés/perdus en totale pour les projets ayant le statut « completed » en mentionnant le projet ayant le nombre maximal des jours gagnés.
		4.	En tant qu'administrateur, je veux consulter les projets en retard de livraison et le nombre de jours en retard de chaque projet.
		5.	En tant qu'administrateur/chef de projet, je veux consulter l'état d'avancement, le nombre de jours passés et le nombre de jours restants pour un projet donné.
1.	Gestion des tableaux de bord	6.	En tenant compte de sa charge, en tant qu'administrateur/chef de projet, je veux consulter le temps minimal de la réalisation du projet, le temps optimal, le nombre de ressources nécessaires en mois/hommes pour la réalisation d'un projet donné.
		7.	En tant qu'administrateur/ chef de projet/chef d'équipe , je souhaite consulter un tableau de bord représentant le travail de chaque semaine du mois en cours.
		8.	En tant qu'administrateur/chef de projet/chef d'équipe, je veux chercher les statistiques du travail de chaque semaine d'un mois donnée.
		9.	En tant qu'administrateur/chef de projet/chef d'équipe, je veux consulter le rapport journalier (taux de retard/taux d'avancement) de l'équipe.
		10.	En tant qu'administrateur/chef d'équipe/chef de projet, je veux consulter les statistiques d'avancement de chaque membre.
		11.	En tant qu'administrateur/chef d'équipe, je veux consulter les tâches en avance/en retard de chaque membre d'équipe et le nombre de jours en avance/retardés.
		12.	En tant que chef d'équipe, je veux consulter les statistiques d'avancement des tâches de mes équipes.
		13.	En tant que membre, je veux consulter mon tableau de bord dans chaque équipe dont je fais partie.
		14.	En tant que chef d'équipe/membre, je veux chercher les statistiques d'une équipe donnée.
		15.	En tant qu'utilisateur de l'application, je veux télécharger les tableaux de bord.

### 3.3. Analyse

Dans cette phase, nous comprenons mieux les besoins de notre projet en termes de fonctions. Tout d'abord nous présentons le diagramme de cas d'utilisation. Ensuite, nous mettons en disposition les descriptions textuelles de quelques cas.

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

Le diagramme de cas d'utilisation du sprint 6 est présenté par la figure 70.

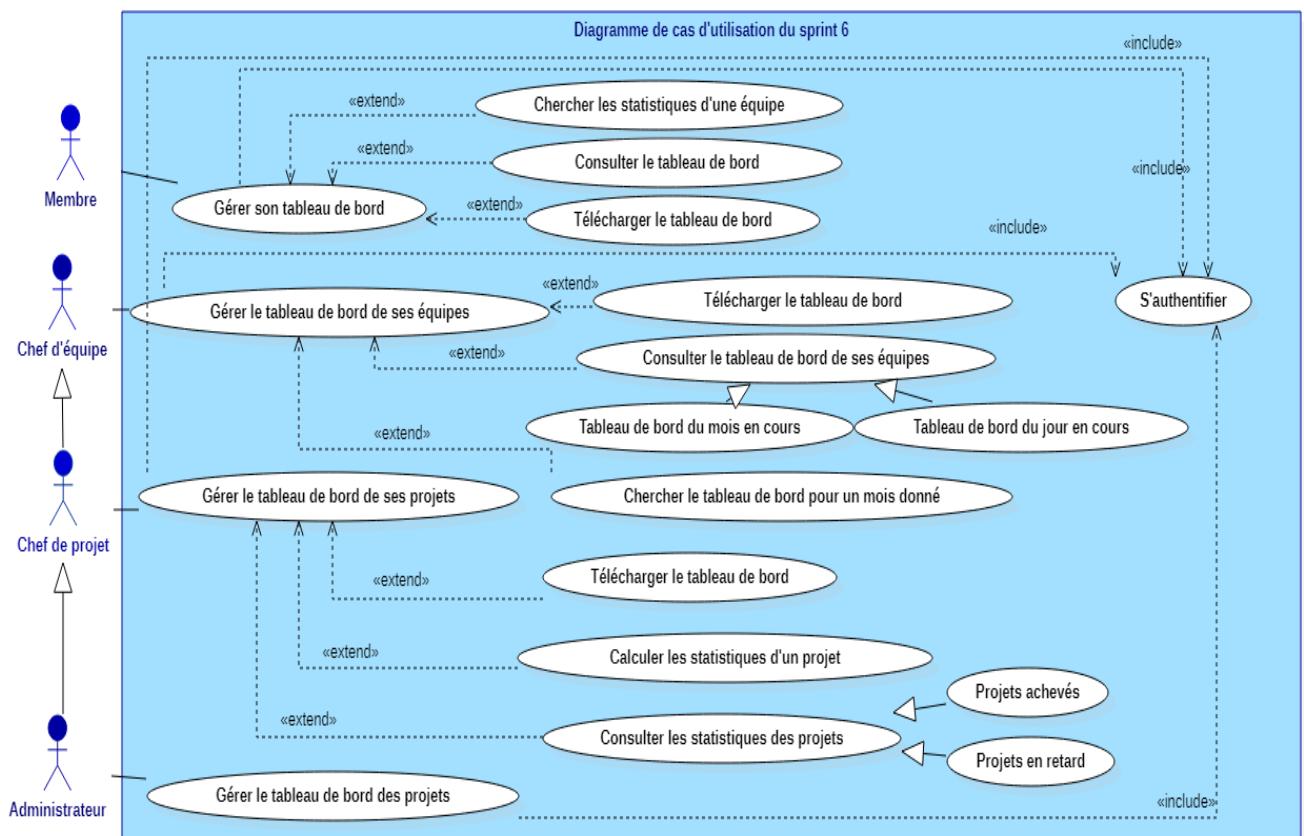


Figure 70 : Diagramme de cas d'utilisation du sprint 6

#### 3.3.2. Raffinement des cas d'utilisation

Dans cette section, nous allons présenter la description textuelle de quelques cas d'utilisation.

##### - Description textuelle du cas d'utilisation « Télécharger le tableau de bord »

Le tableau 33 présente la description textuelle du cas d'utilisation « Télécharger le tableau de bord ».

**Tableau 33 : Description textuelle du cas d'utilisation « Télécharger un tableau de bord »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Télécharger le tableau de bord
<b>Acteur</b>	Administrateur, Chef d'équipe, Chef de projet, Membre
<b>Objectif</b>	Télécharger le tableau de bord
<b>Résumé</b>	Chaque page peut avoir plusieurs tableaux de bord, le but est de télécharger un tableau de bord spécifique.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>
- L'utilisateur doit être connecté	-Le tableau de bord est téléchargé.
<b>Scénario</b>	
1- L'utilisateur clique sur l'icône « charts ». 2- Le système affiche les tableaux de bord spécifique au utilisateur. 3- L'utilisateur choisit le tableau de bord à télécharger et clique sur l'icône « download » 4- Le système télécharge la page.	

- **Description textuelle du cas d'utilisation « Consulter les statistiques d'un projet »**

Le tableau 34 présente la description textuelle du cas d'utilisation « Consulter les statistiques d'un projet choisi ».

**Tableau 34 : Description textuelle du cas d'utilisation « Calculer statistiques projet »**

<b>Raffinement du cas d'utilisation</b>	
<b>Cas d'utilisation</b>	Calculer les statistiques d'un projet choisi
<b>Acteur</b>	Administrateur, Chef de projet
<b>Objectif</b>	Calculer les statistiques d'un projet choisi
<b>Résumé</b>	Un chef de projet ou un administrateur souhaite choisir un projet en cours de développement et consulter ses statistiques.
<b>Conditions</b>	
<b>Préconditions</b>	<b>Post Conditions</b>

- L'utilisateur doit être connecté - Le projet doit avoir le statut « processing »	-Le calcul et l'affichage des statistiques.
<b>Scénario</b>	
1- L'utilisateur clique sur le bouton « choose project ». 2- Le système affiche la liste des projets avec le statut « processing ». 3- L'utilisateur choisit un projet. 4- Le système calcule les statistiques. 5. Le système affiche le résultat.	

### 3.5. Réalisation

Dans ce paragraphe, nous représentons une démonstration du travail effectué tout au long de ce sprint.

#### 3.5.1. Interfaces des tableaux de bord spécifiques aux projets

Cette section est consacrée à la présentation des statistiques des projets selon leurs statuts.

##### - Statistiques des projets ayant le statut « completed »

Le tableau de bord présente le nombre de jours gagnés/perdus pour chaque projet ayant le statut « completed ». Comme le montre la figure 71, le projet « Computer Game Companion App » est en retard de 17 jours.

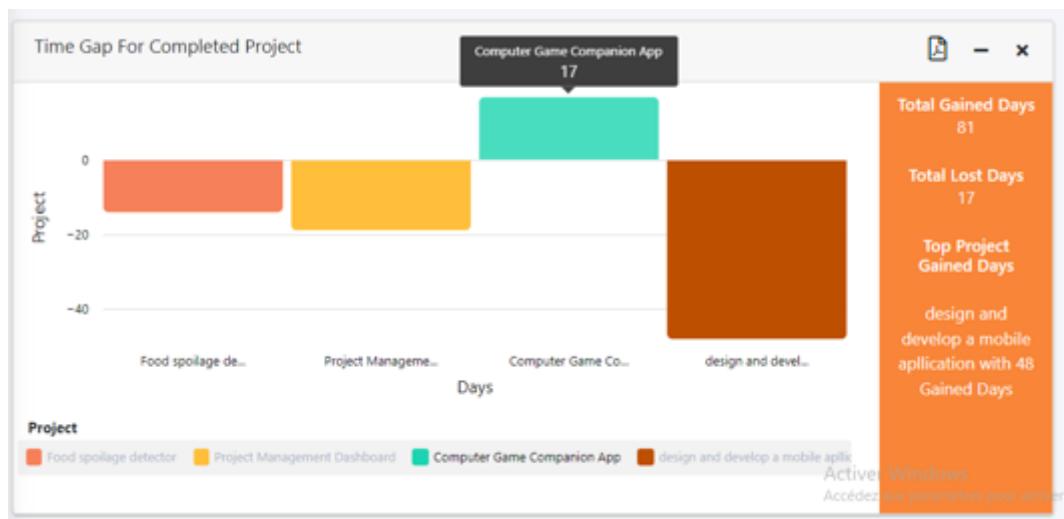


Figure 71 : Interface du tableau de bord des projets ayant le statut « Completed »

### - Statistiques des projets ayant le statut « *delayed* »

Ce tableau de bord présente les projets qui sont en retard de livraison ainsi que le nombre de jours retardés. La figure 72 montre que le projet « Drawing tool » est en retard de 4 jours.

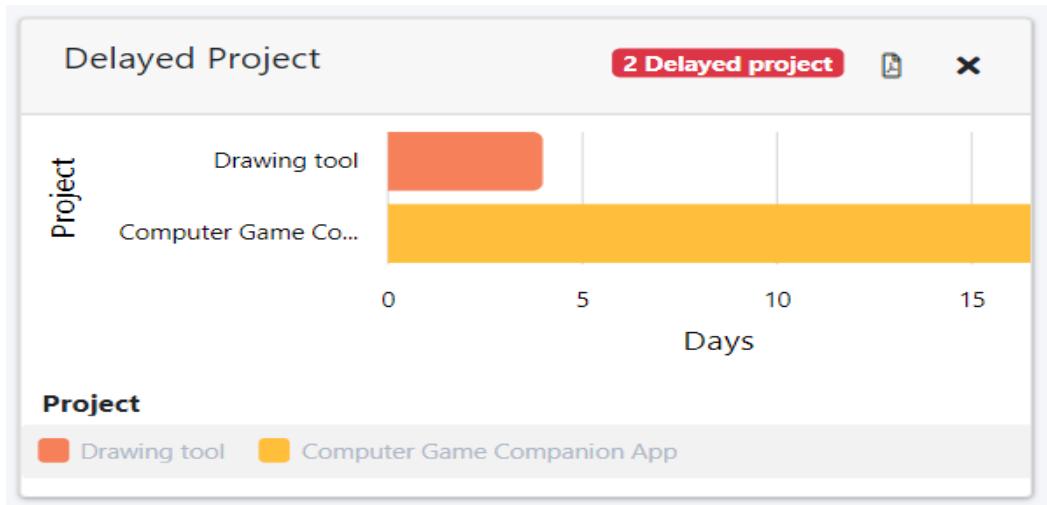


Figure 72 : Statistiques des projets retardés

### - Statistiques des projets ayant le statut « *processing* »

Pour les projets en cours, l'administrateur peut choisir un projet, consulter l'état de son avancement, le nombre de jours passés et le nombre de jours restants.

Exemple : Choix du projet : E-Library Project (figure 73)

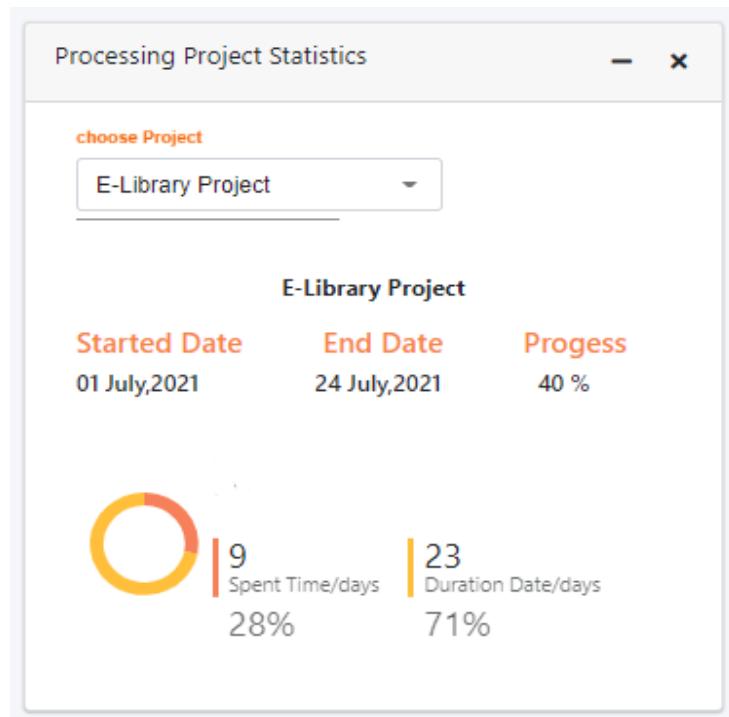


Figure 73 : Statistiques des projets en cours

### - Evaluation des charges du projet

A partir de la charge du projet, l'administrateur/le chef de projet, peut consulter le temps minimal à la réalisation du projet, le temps optimal et le nombre de ressources nécessaires pour contribuer à la réalisation de ce projet. Les règles appliquées sont [30] :

$$T_{min} = (2.5 * \text{charge du projet (mois/hommes)})^{1/3}$$

$$T_{opt} = 1.4 * \text{charge du projet (mois /hommes)}$$

$$\text{Nb Resource : } (\text{charge du projet (mois /hommes)})^{1/2}$$

La figure 74 montre que le projet « Water Quality Monitoring System » est estimé à 9 mois homme temps optimal, il faudra 7 mois/homme équivalent temps plein.

Project : Water Quality Monitoring System		Details
Charge project :	56	
Tmin mois/homme :	7	
Topt mois/homme :	9	
Nb Resource :	7	

Figure 74 : Evaluation des charges du projet

### 3.5.2. Interfaces des tableaux de bord spécifiques aux équipes

Cette section est consacrée à la présentation des statistiques relatives aux équipes.

#### - Statistiques du travail pendant un mois et calcul du taux de retard et d'avancement pour une équipe

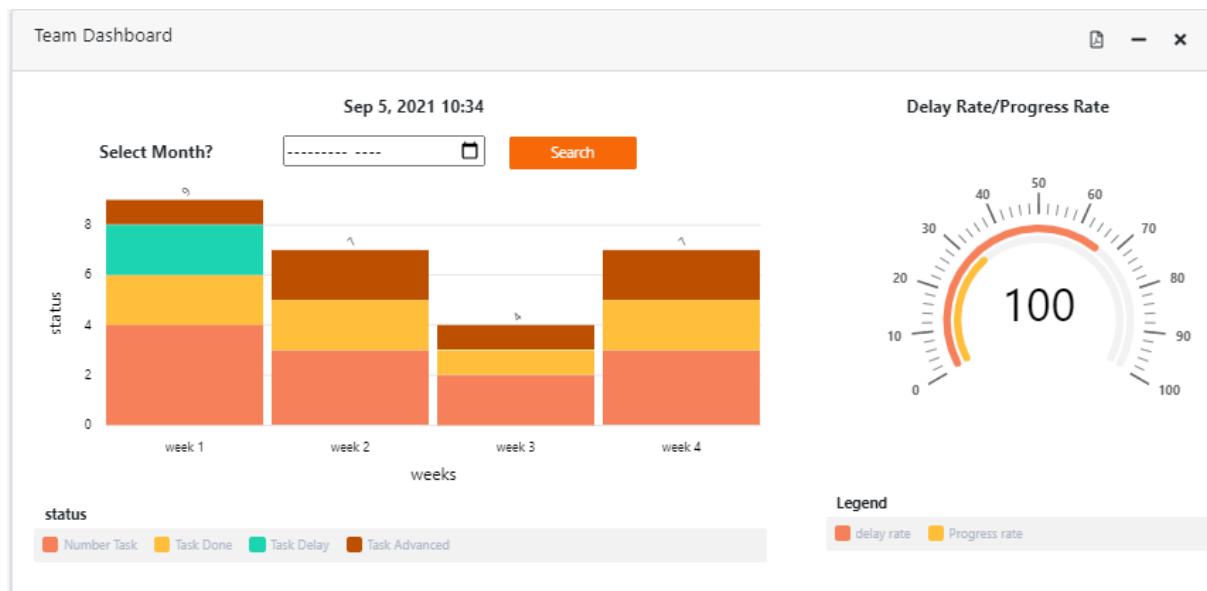
La figure 75 expose les statistiques d'une équipe « Full stack team » pour le mois « septembre », le chef d'équipe peut également chercher et consulter les statistiques d'un mois souhaité. Ce tableau de bord présente les statistiques de chaque semaine du mois, il affiche :

- Le nombre des tâches qu'elles doivent être réalisées.

- Le nombre des tâches réalisées.
- Le nombre des tâches en retard.
- Le nombre des tâches en avance.

Elle présente également le calcul des indicateurs de performance relatives au taux de retard et taux d'avancement pour la journée encours en appliquant les règles suivantes [31] :

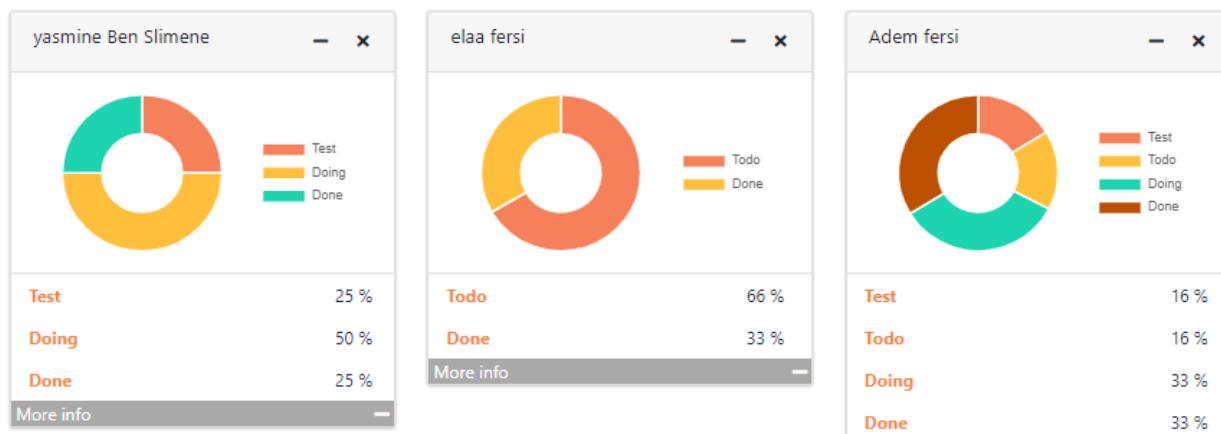
- Taux de retard : (tâches non réalisées / tâches prévues) \* 100
  - Taux d'avancement : (tâches accomplies / tâches prévues) \* 100



**Figure 75 : Interface du tableau de bord de travail de chaque équipe**

#### - Statistiques d'avancement de chaque membre d'équipe

Pour chaque membre, le système affiche le pourcentage des tâches/statut. Ce calcul est illustré par la figure 76.



**Figure 76 : Avancement de chaque membre d'équipe**

- En cliquant sur l'icône « more info », nous pouvons consulter si un membre a des tâches en retard et/ou en avance avec les jours retardés et/ou gagnés comme l'indique la figure 77.

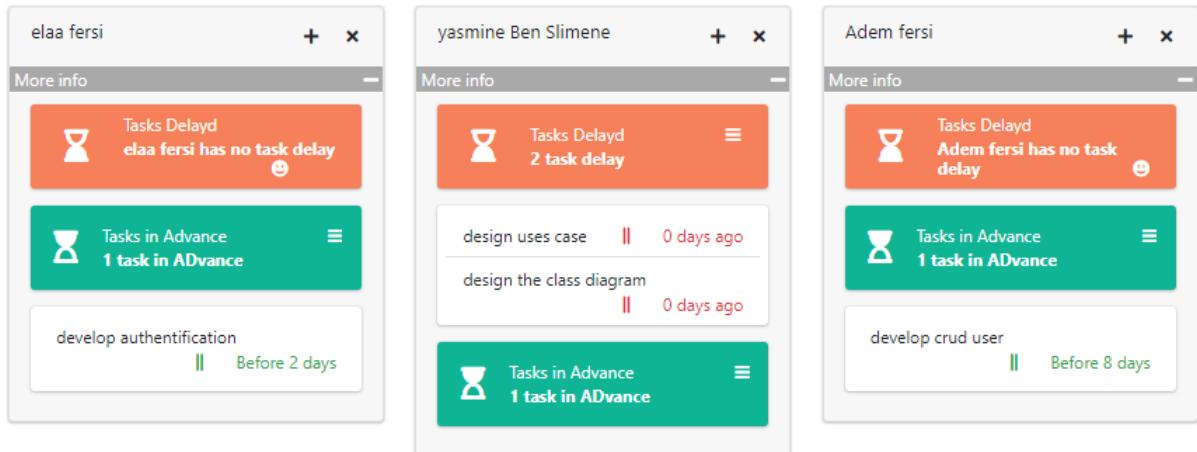


Figure 77 : Interface de consultation des jours en avance/en retard des tâches

### - Statistiques d'avancement des équipes

Pour chaque équipe, le système calcule l'ensemble des tâches de chaque membre, les regroupe par statut et calcul le pourcentage d'avancement par rapport au statut, cette interface donne une vue globale sur l'avancement du travail de chaque équipe et elle est consultée par le chef d'équipe. Il pouvait également chercher le tableau de bord d'une équipe donnée en entrant son nom dans la barre de recherche (Figure 78).

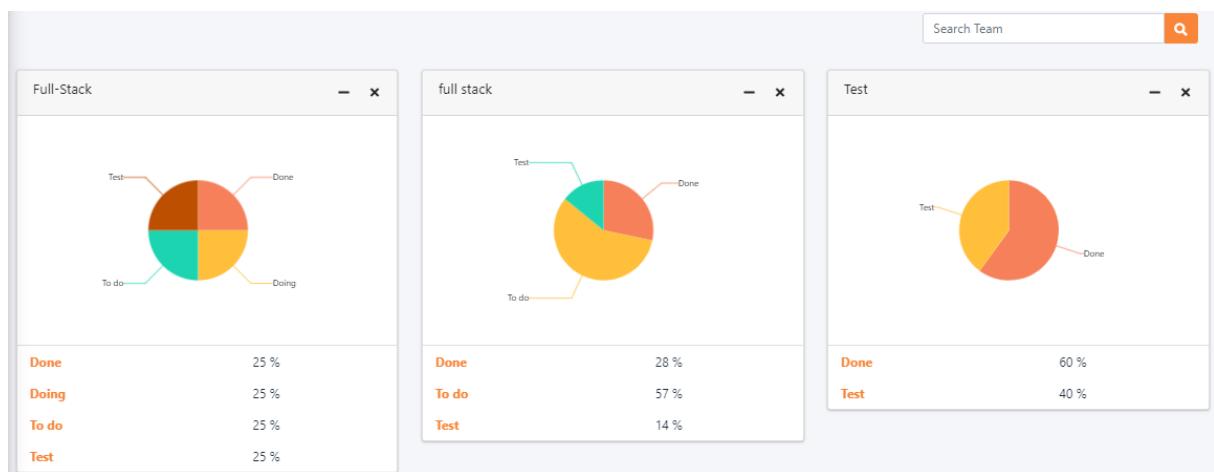
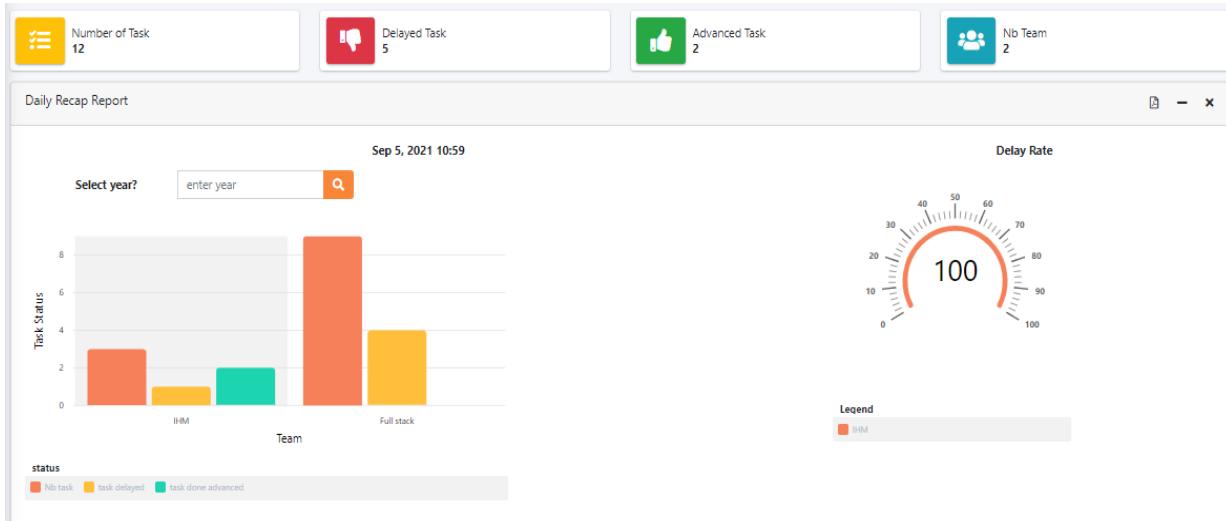


Figure 78 : Interface des tableaux de bord globaux des équipes

### 3.5.3. Interface des tableaux de bord spécifiques à chaque membre

Chaque membre peut consulter son tableau de bord qui représente le nombre de ses tâches dans chaque équipe, ainsi que le nombre des tâches en avance et en retard.

Le tableau de bord spécifique à un membre est représenté par la figure 79.



**Figure 79 : Interface du tableau de bord du membre**

Le membre peut également consulter son avancement par rapport à ses tâches en consultant son profil.

## 4. Sprint 7 : Test, déploiement et mise en place d'une chaîne CI/CD

Dans ce sprint nous allons présenter le processus de déploiement de notre application sur le cloud heroku, nous commençons par l'exposition du technical story de ce sprint et puisque nous avons développé des tests unitaires et les présentés dans les chapitres 3 et 4, dans ce sprint nous allons présenter les tests automatisés développés, les étapes de déploiement de notre application et la mise en place d'une chaîne CI/CD.

### 4.1. Technical Story

Les « Technical Stories » sont des définitions de travail à réaliser équivalentes aux « User Stories » dans le backlog de sprint, mais leur écriture provient de besoins techniques (dans notre cas le test, déploiement et la mise en place d'une chaîne CI/CD pour notre application). Elles sont donc estimées comme les « User Stories », en points relatifs ou en unités de temps. Seulement, à la différence des User Stories, elles n'obéissent pas la plupart du temps à un formalisme rédactionnel particulier [31].

Le tableau 35 illustre les besoins techniques de notre application en présentant le technical story du sprint.

**Tableau 35 : Technical story du sprint 7**

ID Module	Module	ID Technical Story	Technical Story
1.	Test, déploiement Et mise en place d'une chaîne CI/CD	1.	Développement des tests automatisés sélénium pour certaines fonctionnalités.
		2.	Déploiement de l'application sur le cloud heroku.
		3.	Mise en place de l'intégration continue de l'application.
		4.	Mise en place du déploiement continu de l'application.

## 4.2. Tests Automatisés

Les tests automatisés s'exécutent sans l'intervention d'un humain. Ils ont pour objectif de simplifier autant que possible les efforts de test grâce aux scripts. Les tests sont alors exécutés selon ces scripts, les résultats sont signalés et comparés aux résultats des essais antérieurs. Ses principaux intérêt résident dans le fait qu'ils permettent de gagner du temps et de l'argent. En termes de budget, il rend possible des économies sur les charges car l'humain est moins sollicité, si ce n'est pour effectuer la maintenance du test [32].

### 4.2.1. Etude comparative des outils de tests

Le tableau 36 présente une étude comparative entre 3 outils de tests disponibles pour les tests automatisés [33].

**Tableau 36 : Etude comparative des outils de tests disponibles [33]**

Outils	RFT(IBM)	QTP(HP)	SELENIUM
<b>Licence</b>	Requis	Requis	Open source
<b>Prix</b>			Outil open source
<b>Support client</b>	Support dédié d'IBM	Assistance dédiée de HP	Communauté Open Source
<b>Utilisation du matériel lors de l'exécution du script de test</b>			Faible
<b>Compétences en rédaction de code</b>	Nécessaire	Nécessaire	Une connaissance et une expérience en écriture de code sont indispensables
<b>Environnements pris en charge</b>	Windows seulement	Windows seulement	Windows, Linux, Mac, Solaris OS X (uniquement si la prise

			en charge JVM ou JavaScript existe)
<b>Langues prises en charge</b>	C # et Java	Script VB	Java, C #, PHP, Ruby, Python, Perl et JavaScript

#### 4.2.2. Développement des tests selenium

Nous avons opté pour le choix des tests seleniums, et nous avons développé des scripts pour différents scénarios à savoir : l'authentification, l'accès à la liste des utilisateurs et l'ajout d'un utilisateur, l'accès à la liste des projets et l'application des différentes recherches de même pour les tableaux de bord des projets.

La figure 80 présente le résultat d'exécution des différents scénarios.

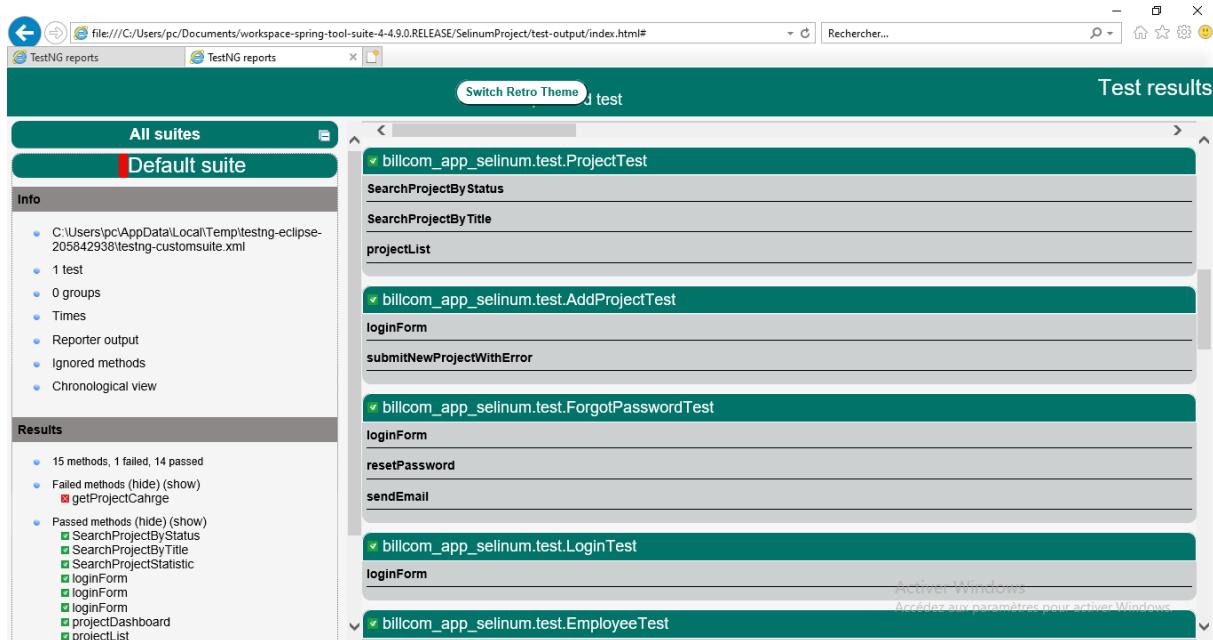


Figure 80 : Interface de résultats des tests selenium

Nous présentons dans la figure 81 un scénario d'exécution du cas d'utilisation « S'authentifier ».

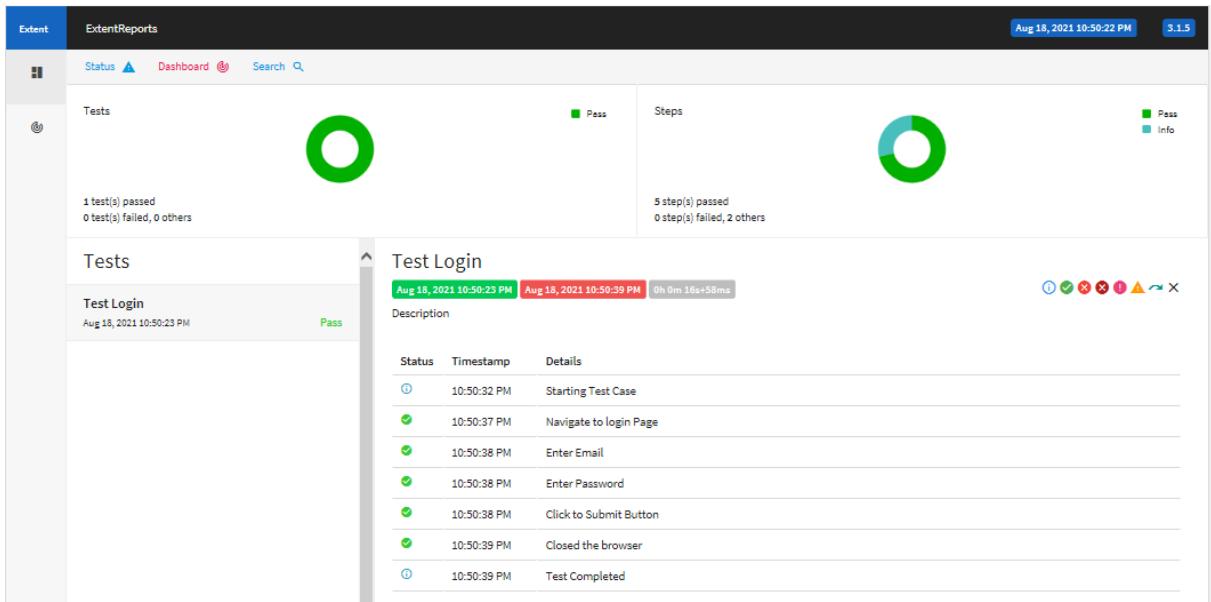


Figure 81 : Test automatisé du cas d'utilisation « S'authentifier »

### 4.3. Déploiement de l'application

Pour la mise en production du notre application, nous avons utilisé la plateforme heroku. Cette plateforme travaille en tant que service (PAAS) permettant de déployer des applications sur le cloud. Nous allons présenter dans ce qui suit l'architecture de déploiement de notre application, et par la suite, nous allons présenter des imprimés écrans sur les étapes de déploiement.

#### 4.3.1. Architecture de déploiement

Afin de déployer notre application sur heroku, nous avons besoin de mettre en place une architecture de déploiement.

En effet, une fois nous avons mis notre code sur GitHub, nous faisons la connexion de notre repository avec le cloud heroku après nous pouvons générer notre build.

L'architecture de déploiement est présentée par la figure 82.

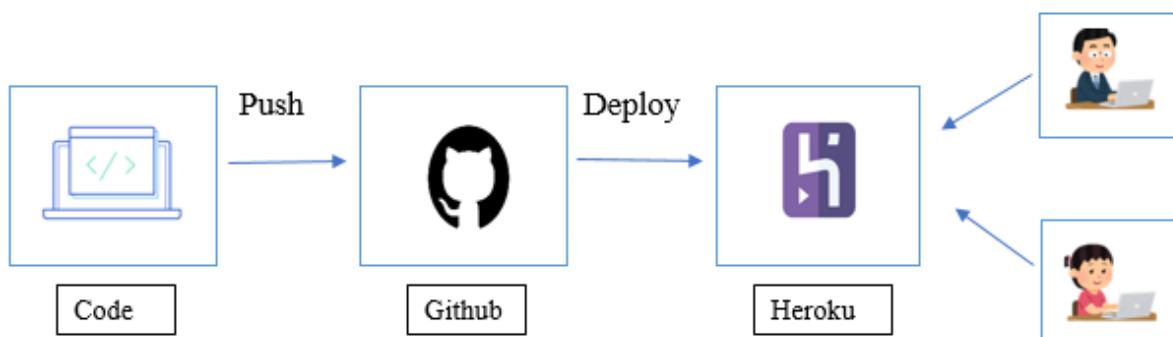


Figure 82 : Architecture de déploiement

### 4.3.2. Processus de déploiement

Nous allons présenter les étapes simples de déploiement de l'application sur le cloud heroku :

1. La génération d'une version de production de l'application angular.

```
PS C:\Users\pc\Documents\billcom-front-web> ng build --prod
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.

chunk {2} polyfills-es2015.de1c487fabf176256383.js (polyfills) 81.9 kB [initial] [rendered]
chunk {0} runtime-es2015.e5938d02aafbeec9aa31.js (runtime) 2.26 kB [entry] [rendered]
chunk {0} runtime-es5.e5938d02aafbeec9aa31.js (runtime) 2.26 kB [entry] [rendered]
chunk {6} 6-es2015.221da577ba173006ceac.js () 17.4 kB [rendered]
chunk {6} 6-es5.221da577ba173006ceac.js () 17.4 kB [rendered]
chunk {5} 5-es2015.94a091638731302829df.js () 214 kB [rendered]
chunk {5} 5-es5.94a091638731302829df.js () 214 kB [rendered]
chunk {1} main-es2015.f2fa5d8f1bbb1d1413c1.js (main) 3.31 MB [initial] [rendered]
chunk {1} main-es5.f2fa5d8f1bbb1d1413c1.js (main) 3.46 MB [initial] [rendered]
chunk {4} styles.0711a7aaa46fee2caf8d.css (styles) 868 kB [initial] [rendered]
chunk {scripts} scripts.3a51fe70f9c165cdfc90.js (scripts) 600 kB [entry] [rendered]
Date: 2021-08-25T00:10:20.442Z - Hash: 6abfdf8af2fb57987d46 - Time: 462722ms
```

Figure 83 : Génération d'une version de production de l'application angular

2. La copie du répertoire de la sortie dans le dossier « static » de l'application du Spring Boot.
3. La création d'une classe de configuration pour rediriger les demandes des fichiers manquants vers index.html sous le dossier « static ».
4. le déploiement le code sur Github.

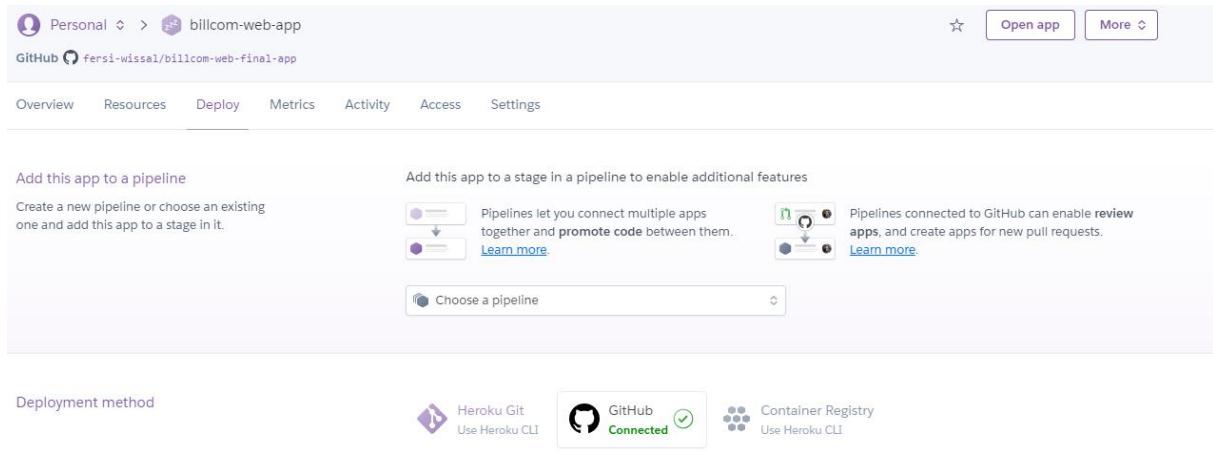
```
$ git add .

pc@DESKTOP-7A54AKS MINGW64 ~/Documents/workspace-spring-tool-suite-4-4.9.0.RELEASE/billcom-web-final (master)
$ git commit -m "Deploy into github"
[master 36ed4e1] Deploy into github
 1 file changed, 3 insertions(+)

pc@DESKTOP-7A54AKS MINGW64 ~/Documents/workspace-spring-tool-suite-4-4.9.0.RELEASE/billcom-web-final (master)
$ git push origin master
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 705 bytes | 70.00 KiB/s, done.
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/fersi-wissal/billcom-web-final-app.git
 5a278c9..36ed4e1 master -> master
```

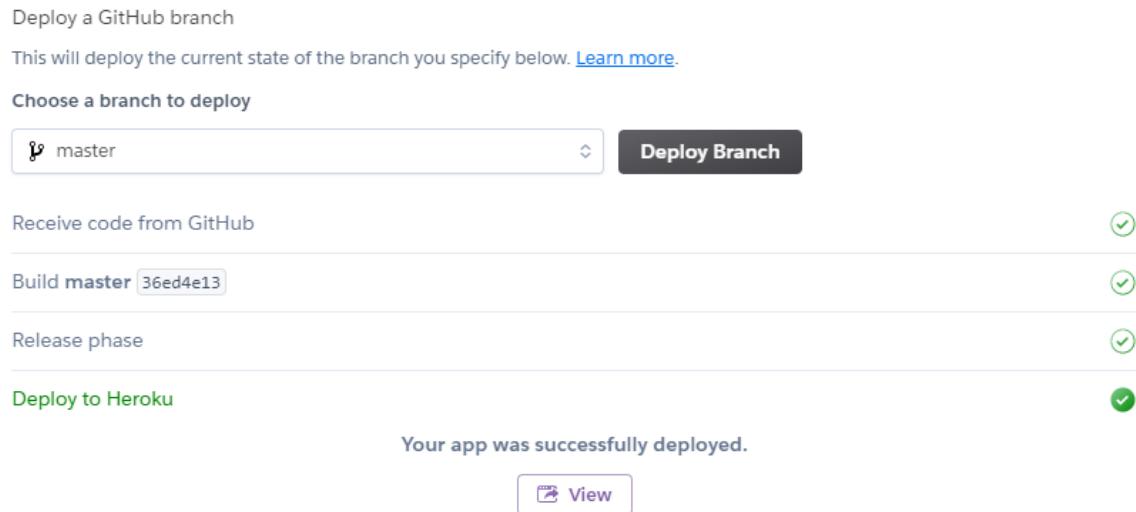
Figure 84 : Interface de déploiement du code sur GitHub

5. La configuration de la base de données
6. La création d'un répertoire dans heroku et le connecter avec GitHub.



**Figure 85 : Connecter le répertoire Heroku avec GitHub**

7. Le choix de la branche et le clique sur le bouton « Deploy Branch ».



**Figure 86 : Interface de résultat du déploiement**

#### 4.4. Mise en place d'une chaîne CI/CD

La configuration manuelle pour générer un AppBundle et le déployer sur le cloud Heroku, est une tâche répétitive qui nécessite de répéter les mêmes étapes à chaque changement. Nous avons donc décidé de l'automatiser en faisant gagner l'équipe du temps et de ressources, et ainsi, les changements seront livrés plus rapidement aux utilisateurs de l'application. Dans les paragraphes qui suivent nous décrivons le processus suivi pour configurer la chaîne CI/CD qui est un ensemble de principes de fonctionnement et de pratiques qui aident les équipes de développement à fournir des changements de code fréquents de manière fiable. Il s'agit de l'automatisation et de la surveillance continues tout au long du cycle de vie de l'application [34].

#### **4.4.1. Intégration continue/Continuous Integration (CI)**

L'intégration continue (CI) est une pratique qui consiste à fusionner tout le code de travail des développeurs de différentes branches et à l'intégrer dans une seule branche de la repository (Git). Après, à partir de cette version de code, un serveur d'intégration génère un build prêt à déployer.

#### **4.4.2. Déploiement Continu/Continuous Deployment (CD)**

Le déploiement continu consiste à automatiser le processus de déploiement d'un build d'une application sur un environnement de production. Ce processus nécessite un serveur d'intégration, son rôle est d'automatiser le processus d'intégration et de déploiement continu. A chaque fois qu'un développeur valide un changement de code, le serveur d'intégration continue détecte le changement et exécute automatiquement une chaîne d'actions qui consiste à : exécuter les tests unitaires, générer un build, après, une fois le build est généré avec succès, il est déployé dans l'environnement de test (staging environment) ou dans l'environnement de production. En cas de problème avec le build (par exemple un test qui échoue), le serveur CI en informe immédiatement et automatiquement les développeurs.

Nous allons utiliser l'outil d'intégration continue Jenkins.

#### **4.4.3. Application de la CI/CD sur l'application**

Tout d'abord, nous développons du code de l'application, puis, à l'aide des commandes git, nous déposons ce code dans notre répertoire sur GitHub.

C'est le moment où Jenkins commence son travail. En effet, Il détecte le changement du code sur GitHub, et lance tout un processus de build.

Ce processus se fait suite à une configuration mise en place au préalable et est présentée par les étapes suivante :

1. La création d'un projet dans l'outil d'intégration Jenkins.
2. La création d'une clé privée comme information d'identification

The screenshot shows the Jenkins global credentials management interface. The title bar says 'Jenkins'. The left sidebar has links for 'Tableau de bord', 'Identifiants', 'System', and 'Identifiants globaux (illimité)'. The main content area is titled 'Identifiants globaux (illimité)' with a subtitle 'Credentials that should be available irrespective of domain specification to requirements matching...'. It lists two entries:

ID	Nom	Type	Description
100c3405-0e05-41a9-b77b-b5537afe44bc	Bilicom-app	SSH Username with private key	
aeeab0d5-c096-43d7-a63e-94c0fe53539	Bilicom-app-heroku	SSH Username with private key	

At the bottom, there are icons for 'Icône', 'S', 'M', and 'L'.

**Figure 87 : Crédit des clés privées**

3. La création des clés privées, publiques en utilisant la méthode RSA et ajouter la clé SSH publique dans le répertoire du GitHub.

```
pc@DESKTOP-7A54AKS MINGW64 ~/ssh
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/pc/.ssh/id_rsa): Billcom-app-key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in Billcom-app-key
Your public key has been saved in Billcom-app-key.pub
The key fingerprint is:
SHA256:zaKQ6ZYvXaHHXLGyJY3bOr12/6GI+paDjrbzi1BdFOQ pc@DESKTOP-7A54AKS
The key's randomart image is:
+---[RSA 3072]---+
| .+.
| o .
| E o
| o. *o=
| +. =SXo
| ..o..B..
| .+.o.o.
| .o++.o++ o . .
| .=*==o+ o...
+---[SHA256]---
```

Figure 88 : Création des clés privées-publiques

4. L'ajout de la clé publique dans « GitHub Configuration » et copier la clé publique au niveau des identifiants globaux du Jenkins.

5. La liaison du projet Jenkins avec l'url du GitHub

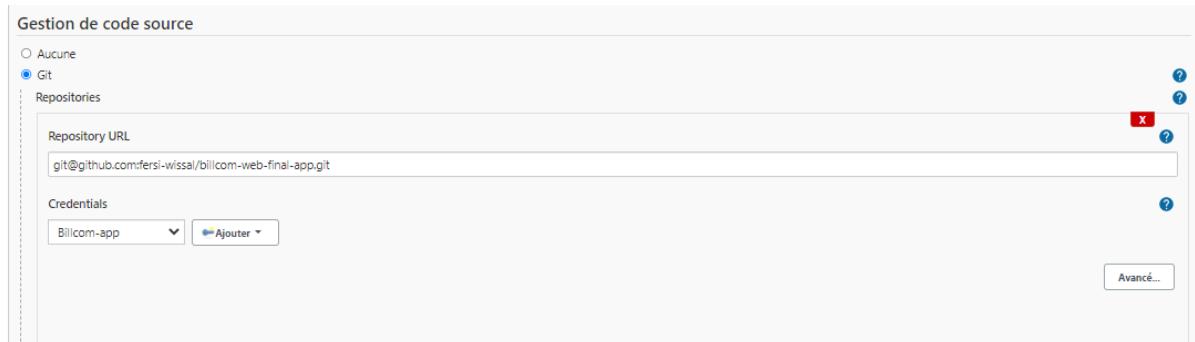
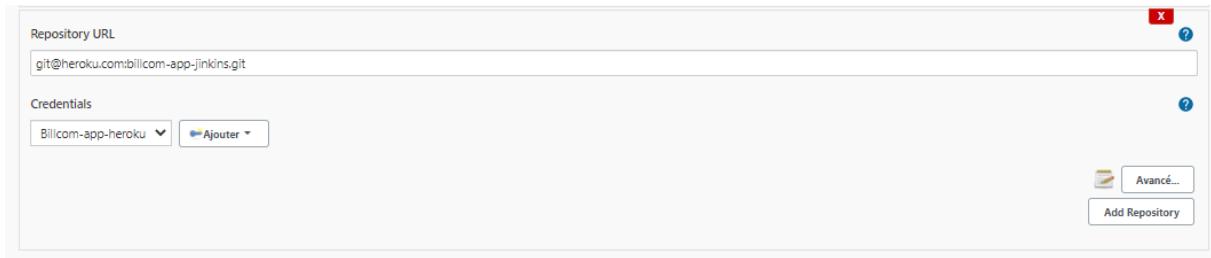


Figure 89 : Liaison du projet Jenkins avec GitHub

A cette étape nous pouvons faire un chargement de notre code à notre projet manuellement.

6. La création d'un répertoire Heroku Git
7. L'ajout d'une branche master dans Heroku Git
8. La création d'une clé publique, privée pour lier entre le projet « Jenkins » et le projet dans « Heroku Git »
9. La connexion du répertoire Heroku avec Jenkins



**Figure 90 : Liaison du projet Jenkins avec « Heroku Repoistory Git »**

10. Une fois les deux répertoires sont connectés correctement avec le projet Jenkins, nous appliquons le principe CI/CD, en effet, une fois il y'a eu une mise à jour de la branche du git, nous devons faire une liaison des événements « push » du git avec Jenkins et les lier avec la branche master du « Heroku Repository Git ». Ces étapes sont présentées par la figure 91 et 92.

**Figure 91 : Liaison des événements « push » avec Jenkins**

**Figure 92 : Connexion avec la branche master du « Heroku Git Repository »**

## 11. L'ajout de l'invocation des tests unitaire avant le chargement sur le « Cloud Heroku ».



**Figure 93 : Invocation des tests unitaire**

## 12. L'avertissement du service Jenkins en utilisant les webhook du git.

```
Administrator : C:\Users\pc\Desktop\Info\PFE\Billcom\ngrok.exe - ngrok http 8080
ngrok by @inconshreveable
(Session Status: online)
Session Expires: 1 hour, 57 minutes
Version: 2.3.40
Region: United States (us)
Web Interface: http://127.0.0.1:4040
Forwarding: http://80c8-196-224-170-245.ngrok.io -> http://localhost:8080
Forwarding: https://80c8-196-224-170-245.ngrok.io -> http://localhost:8080

Connections: ttl     opn      rt1      rt5      p50      p90
              1        0       0.01    0.00    5.02    5.02

HTTP Requests:
-----
POST /github-webhook/          200 OK
```

**Figure 94 : Avertissement du Jenkins**

Maintenant, notre chaîne d'intégration est mise en place, une fois nous avons changé la branche master, Jenkins va être averti et il exécute automatiquement le processus de déploiement.

En effet, Jenkins exécute les tests unitaires en premier lieu et s'assure qu'il n'y a pas d'erreur.

13. La figure 95 présente le résultat des tests unitaires.

## Résultats des tests

0 échecs ( $\pm 0$ )

34 tests ( $\pm 0$ )

A pris 2,3 s.

Ajouter une description

### Tous les tests

Package	Durée	Échec	(diff)	Sauté	(diff)	Pass	(diff)	Total	(diff)
com.billcom.app.entity	0,59 s	0		0		8		8	
com.billcom.app.security	1,3 s	0		0		3		3	
com.billcom.app.service	0,41 s	0		0		23		23	

Figure 95 : Résultat des tests unitaire

Après avoir exécuté les étapes de déploiement continu, l'application sera mise à jour sur le « cloud heroku », et la figure 96 présente le résultat du déploiement continu dans l'interface du Jenkins.

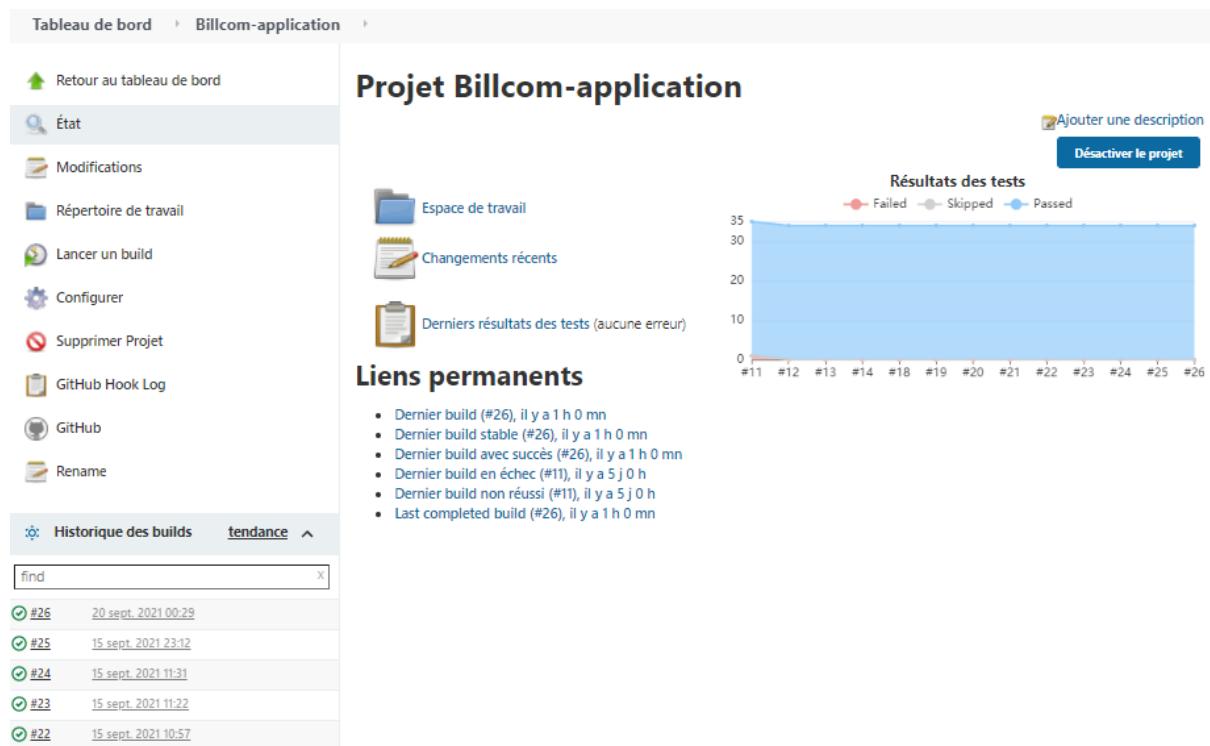


Figure 96 : Interface du Jenkins

La figure 97 présente le processus CI/CD.

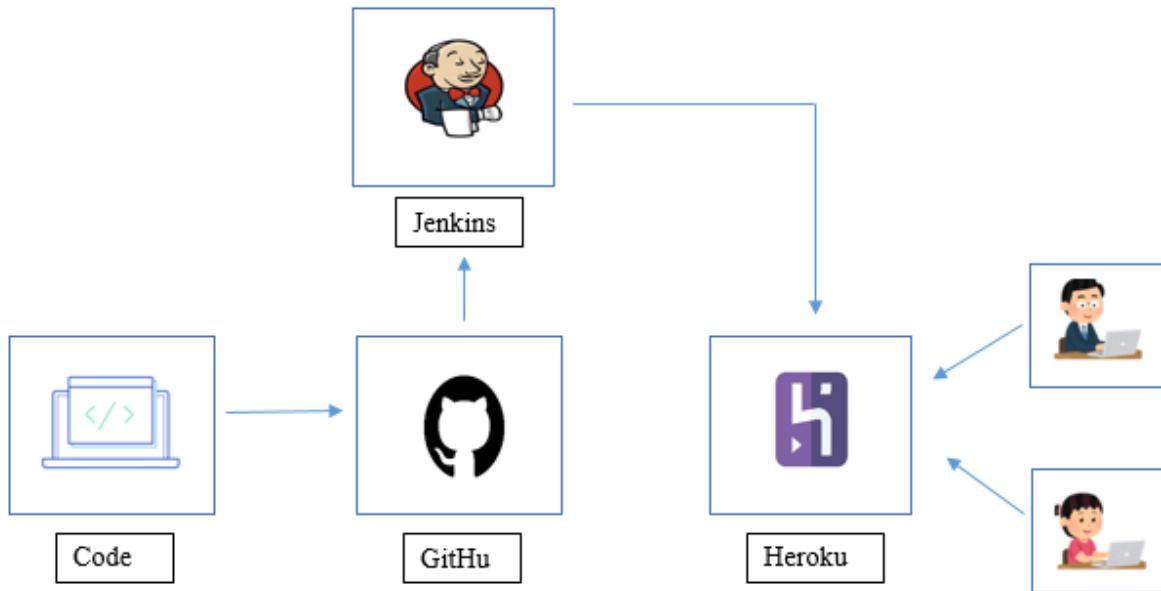


Figure 97 : Architecture de déploiement

## Conclusion

Au niveau de cette dernière release, nous avons présenté trois sprints, la gestion des événements et des notifications, la gestion des tableaux de bord et nous avons clôturer avec le troisième sprint qui est consacré au processus de déploiement de l'application.

# Conclusion générale

Notre projet de fin d'études, réalisé au profit de la société « Billcom Consulting », vise à concevoir, développer, tester et déployer une application web servant à la gestion du projet.

Nous avons commencé par une analyse de la problématique à laquelle fait face « Billcom Consulting » et à travers cette étude, nous avons proposé notre solution en identifiant les objectifs spécifiques de la société tels que la gestion et la planification des projets, la gestion des équipes et de leurs tâches, la mise en place des tableaux de bord, la gestion des utilisateurs et leurs droit d'accès ainsi la mise en place d'une couche de sécurité afin de garantir la confidentialité des données.

Après avoir fixé les objectifs de notre travail, nous avons fait une étude comparative entre les méthodes agiles afin de situer le choix de l'entreprise pour le suivi et la planification du projet, ensuite, nous avons opté pour une analyse minutieuse et une conception détaillée afin de décomposer nos releases. En effet, notre première release a été consacrée à la gestion des utilisateurs et de leurs accès. Par la suite, nous avons détaillé la gestion et la planification du projet dans la deuxième release et notre dernière release a été dédiée à la gestion des évènements, statistiques et au déploiement.

Afin de livrer un produit de qualité, nous avons fait des revues de code en utilisant l'outil sonarLint, nous avons développé également des tests unitaires et des tests automatisés.

Certes, il y'avait des difficultés techniques et conceptuelles auxquelles nous avons dû faire face pendant la période de notre stage, en contrepartie ce dernier a été une opportunité significative au cours de laquelle nous avons été intégrés dans le monde professionnel. Nous avons eu la chance de maîtriser de nouvelles technologies telles que les frameworks Angular 8, Spring Boot 2, se familiariser avec les outils de tests, intégration et déploiement ainsi que la pratique de la méthode agile SCRUM au sein d'une équipe professionnelle et collaborative.

L'application que nous avons présentée n'est que le début d'un processus qui permet de répondre à d'autres attentes de la société « Billcom Consulting », et présente surtout des perspectives pour une éventuelle extension. En effet, en se basant sur ce qui a été fait, l'application peut évoluer en permettant aux clients d'accéder à l'application et d'ajouter leurs feedbacks, nous pouvons également ajouter un module de la gestion des risques, la gestion de la budgétisation des projets et le développement d'une version mobile de notre application.

# Référence

- [1] **Billcom Consulting**, disponible sur : <https://www.billcom-consulting.com/> [Accès le : 10/04/2021]
- [2] **Logiciel payants**, disponible sur : <https://www.appvizer.fr/magazine/operations/gestion-de-projet/top-5-des-logiciels-de-gestion-de-projet-comparatif-1457343003> [Accès le : 15/04/2021]
- [3] **Logiciel gratuits**, disponible sur : <https://www.appvizer.fr/magazine/operations/gestion-de-projet/gestion-de-projet-gratuit> [Accès le : 15/04/2021]
- [4] **Etude comparative des méthodologies agiles**,  
disponible sur: <https://www.atlassian.com/agile> [Accès le : 19/04/2021]
- [5] Schwaber, K., & Sutherland, J. (2011). Le Guide Scrum. *Scrum.org*.
- [6] Muller, P. A., & Gaertner, N. (2000). Modélisation objet avec UML (Vol. 514, p. 235). Paris: Eyrolles.
- [8] **Backlog du produit**  
disponible sur : <https://www.scrumguides.org/docs/scrumguide/v2017/2017-scrum-guide-french.pdf> [Accès le : 22/04/2021]
- [8] Goncalves, G., & Hémery, F. (2000, May). Des cas d'utilisation en UML à la gestion de rôles dans un système d'information. In INFORSID (pp. 367-379).
- [9] **Architecture physique**,  
disponible sur : <https://blog.lecacheur.com/2017/01/26/larchitecture-3-tiers-a-lheure-du-serverless/> [Accès le : 25/04/2021]
- [10] **Architecture front-end**, disponible sur: <https://angular.io/guide/architecture> [Accès le : 25/04/2021]
- [11] **Spring suite tools**, disponible sur : <https://spring.io/tools> [Accès le : 10/05/2021]
- [12] **Visual studio code**, disponible sur : <https://code.visualstudio.com/Docs1> [Accès le : 10/05/2021]
- [13] **PostgreSQL**, disponible sur : <https://www.postgresql.org/> [Accès le : 10/05/2021]
- [14] **StarUML**, disponible sur : <https://staruml.io/> [Accès le : 10/05/2021]

- [15] **pgAdmin**, disponible sur : <https://www.pgadmin.org/> [Accès le : 10/05/2021]
- [16] **Swagger UI**, disponible sur : <https://swagger.io/tools/swagger-ui/> [Accès le : 10/05/2021]
- [17] **Junit**, disponible sur : <https://junit.org/junit5/> [Accès le : 10/05/2021]
- [18] **Selenium**, disponible sur : <https://www.selenium.dev/> [Accès le : 10/05/2021]
- [19] **Java**, disponible sur : <https://www.java.com/fr/> [Accès le : 10/05/2021]
- [20] **TypeScript**, disponible sur : <https://www.typescriptlang.org/> [Accès le : 10/05/2021]
- [21] **Spring Boot**, disponible sur : <https://spring.io/projects/spring-boot> [Accès le : 10/05/2021]
- [22] **Angular**, disponible sur : <https://angular.io/> [Accès le : 10/05/2021]
- [23] **Chart.js**, disponible sur : <https://www.chartjs.org/docs/latest/> [Accès le : 10/05/2021]
- [24] **Heroku**, disponible sur : <https://www.heroku.com/about> [Accès le : 20/07/2021]
- [25] **Jenkins**, disponible sur : <https://www.jenkins.io/> [Accès le : 20/07/2021]
- [26] **GitHub**, disponible sur : <https://github.com/> [Accès le : 25/05/2021]
- [27] **Json Web Token**, disponible sur : <https://jwt.io/> [Accès le : 20/05/2021]
- [28] **Diagramme de séquence**, disponible sur : <http://uml.free.fr/cours/i-p19.html> [Accès le : 20/05/2021]
- [29] **Les webSockets**,  
disponible sur : <https://stacktraceback.com/quest-ce-que-les-websockets-et-a-quoi-ca-sert/> [Accès le : 25/05/2021]
- [30] **Evaluation des projets**,  
disponible sur : <https://www.test-recette.fr/recette/tests-fonctionnels-automatises/> [Accès le : 21/06/2021]
- [31] Madison, J. (2010). Agile architecture interactions. *IEEE software*, 27(2), 41-48.
- [32] **Les tests automatisés**, disponible sur : <https://www.test-recette.fr/recette/tests-fonctionnels-automatises/> [Accès le : 21/06/2021]
- [33] **Etude comparative des outils de tests**, disponible sur : <https://cynoteck.com/fr/blog-post/what-is-selenium-getting-started-with-selenium-automation-testing/> [Accès le : 21/06/2021]

[34] **Mise en place d'une chaîne CI/CD**, disponible sur : <https://www.redhat.com/en/topics/devops/what-is-ci-cd/> [Accès le : 21/06/2021]

# Annexes

## • Release 1

Dans la partie annexe relative à la release 1, nous allons présenter quelques diagrammes de séquence et d'activités de certains cas d'utilisation, le résultat des tests logiciel, la revue du code et la configuration des sprints.

### - Sprint 1 :

Le diagramme de séquence illustrant le scénario de la couche sécurité de notre application est présenté par la figure 98.

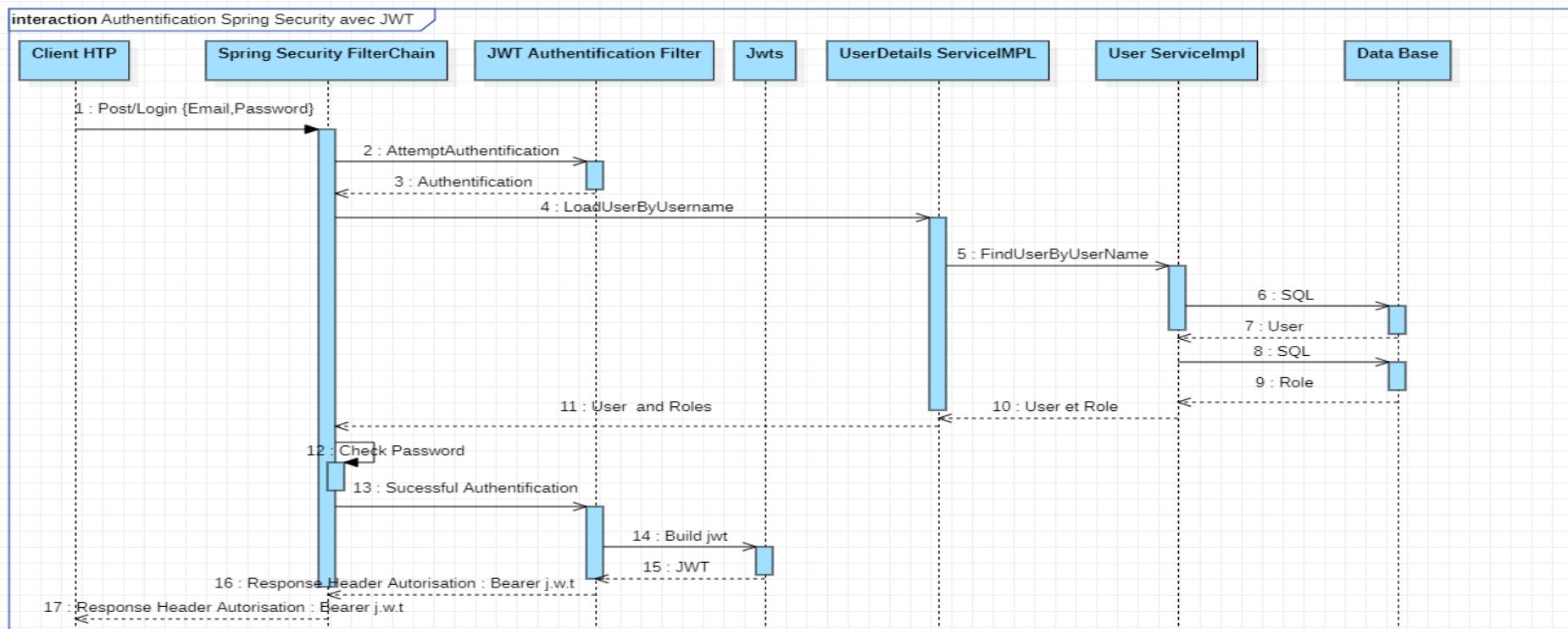
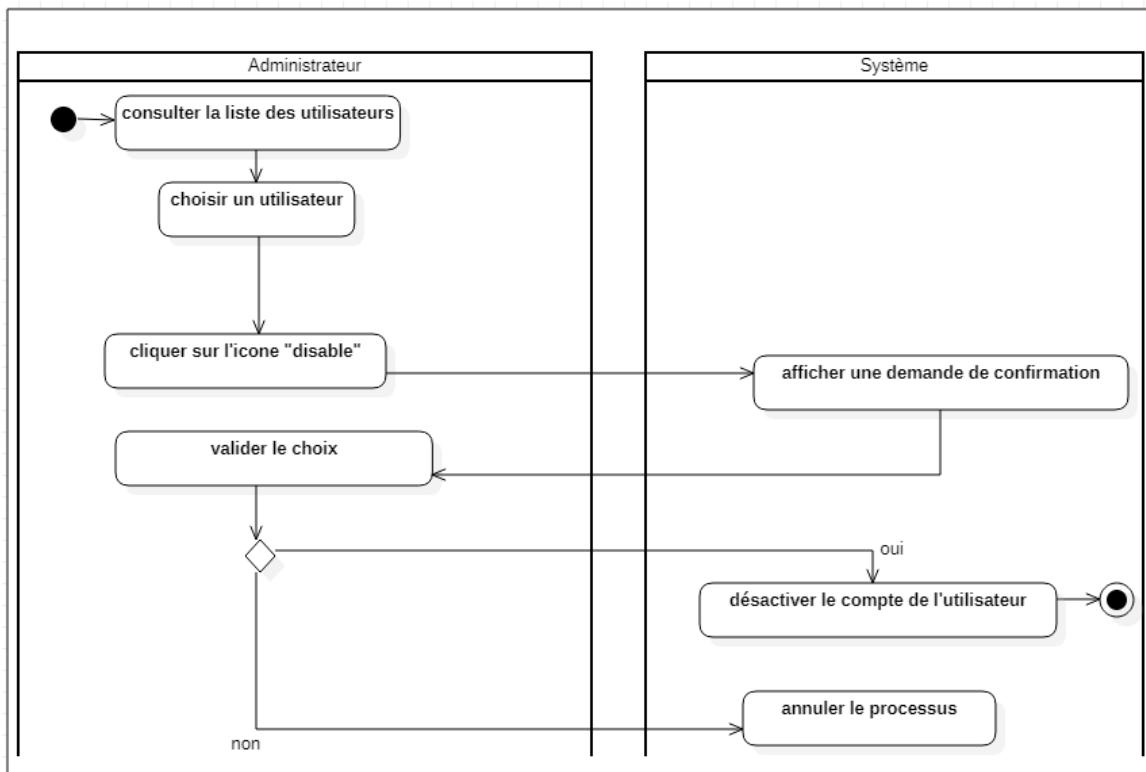


Figure 98 : Authentication spring security avec JWT

### - Diagramme d'activités du cas d'utilisation « Désactiver un compte »

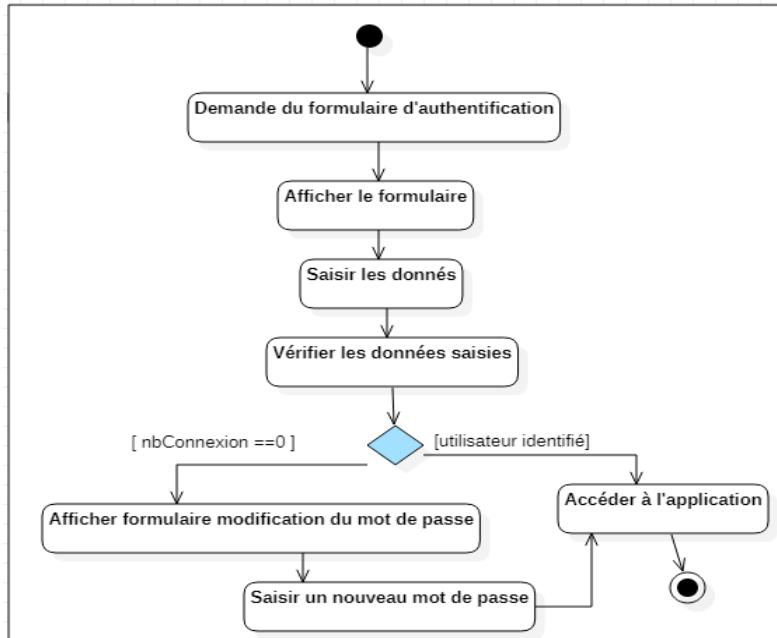
La figure 99 présente le diagramme d'activités du cas d'utilisation « Désactiver un compte ».



**Figure 99 : Diagramme d'activités du cas d'utilisation « Désactiver un compte »**

### - Diagramme d'activités du cas d'utilisation « Modifier mot de passe »

La figure 100 présente le diagramme d'activités du cas d'utilisation « Modifier mot de passe ».



**Figure 100 : Diagramme d'activités du cas d'utilisation « Modifier mot de passe »**

- **Sprint 2 :**

Le diagramme de séquence du cas d'utilisation « Consulter les détails d'un utilisateur » est présenté par la figure 101

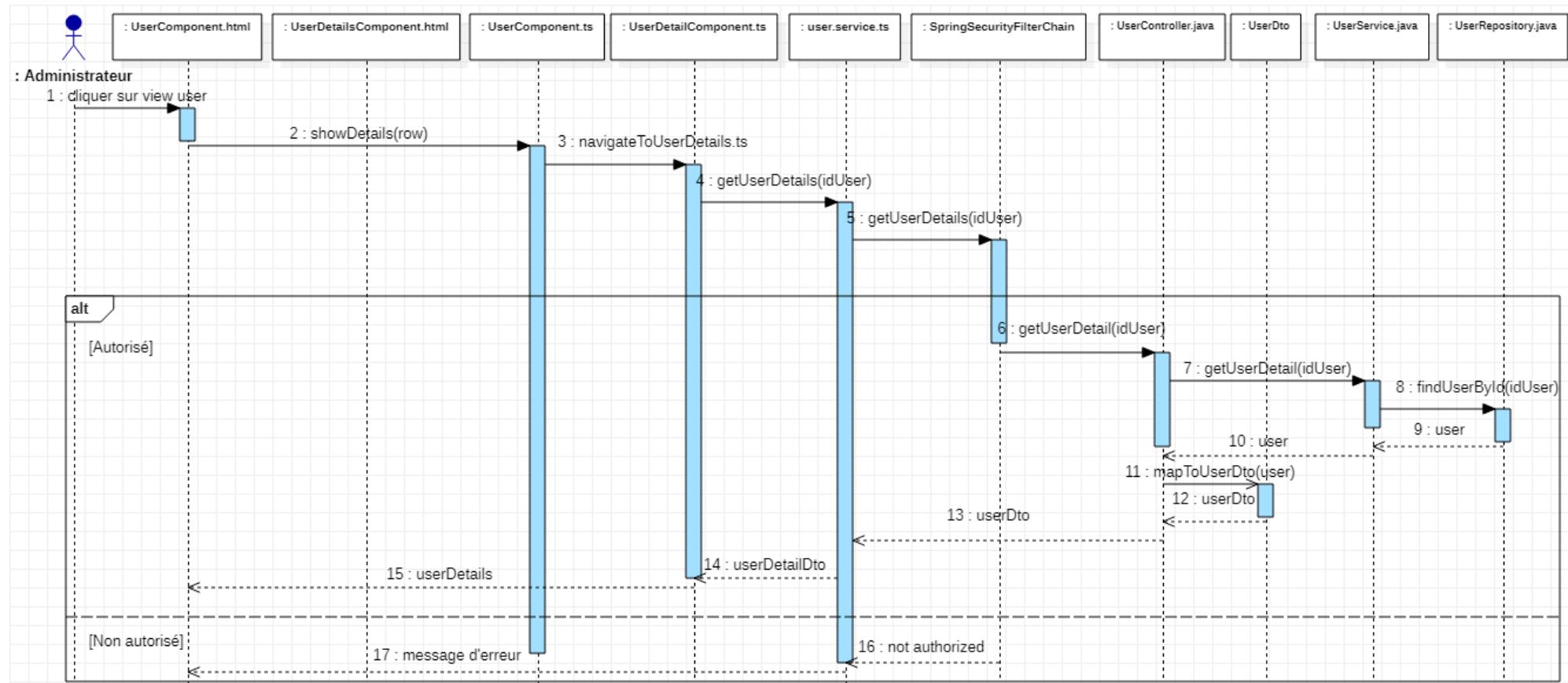


Figure 101 : Diagramme de séquence du cas d'utilisation « Consulter détails utilisateur »

## - Réalisation

En choisissant un utilisateur de la liste des utilisateurs et en cliquant sur l'icône visualiser, l'administrateur peut consulter les détails de ce dernier.



Figure 102 : Interface de consultation des détails d'un utilisateur

## - Test logiciel

Le test du logiciel fait partie du cycle de vie du développement, son objectif est de s'assurer que le code à déployer est de qualité, sans bugs ni erreurs logiques.

Pour ce faire nous avons utilisé le Framework Junit 5, et nous présentons les tests logiciels relatifs à cette release tout en exposant une figure du résultat d'exécution des tests unitaires sur la couche « userService » (figure 103).

The screenshot shows an IDE interface with several tabs and panes. At the top, there's a status bar indicating "Finished after 14,496 seconds". Below it, a "Runs" summary shows 12/12 tests run, 0 errors, and 0 failures. The main pane displays a Java code editor for a test class named `UserServiceTest`. The code contains several `@Test` annotations and assertions related to user service methods like `getCurrentUser` and `getUser`. Below the code editor is a "Coverage" tab showing a table of coverage statistics for different source files. The table includes columns for Element, Coverage, Covered Instructions, Missed Instructions, and Total Instructions. The bottom pane shows a "Console" tab.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
billcom-backend-second-branch	17,1 %	1035	5002	6037
src/main/java	11,9 %	676	4996	5672
src/test/java	98,4 %	359	6	365

**Figure 103 : Application des tests unitaires sur la couche user service**

En plus des tests logiciels effectués, nous avons utilisé SonarLint qui est une extension IDE que nous utilisons afin de détecter et résoudre les problèmes de qualité du code, le résultat de cette analyse est illustré par la figure 104.

The screenshot shows the SonarLint report interface. It lists 16 items found in the `UserService.java` file. Each item has a "Resource" (the file), a "Date" (3 days ago), and a "Description" which includes a small icon and a detailed explanation of the issue. The descriptions cover various code smells and potential improvements, such as making fields final, using static constants, and removing unused imports. The bottom of the report shows the path "File/Billcom-web/src/main/java/com/billcom/app/service/UserService.java (at 02/08/2021 15:03)" and a note about activating Windows.

**Figure 104 : Analyse de SonarLint**

Nous avons travaillé avec l'outil SwaggerUI afin de tester les apis de notre première release, la figure 105 présente les apis « Rest » de sprint 1 et 2 de notre release, ainsi l'ajout du JWT qui est obligatoire pour la majorité des fonctionnalités sauf pour les cas d'utilisation « S'authentifier », « Reset passowrd » et « Update password ».

**Figure 105 : Documentation de la première release**

- **Release 2**
  - **Sprint 1**

Pour avoir une idée sur le séquencement du scénario « Consulter les détails d'un projet » nous présentons dans la figure 106 le diagramme de séquence de ce cas d'utilisation

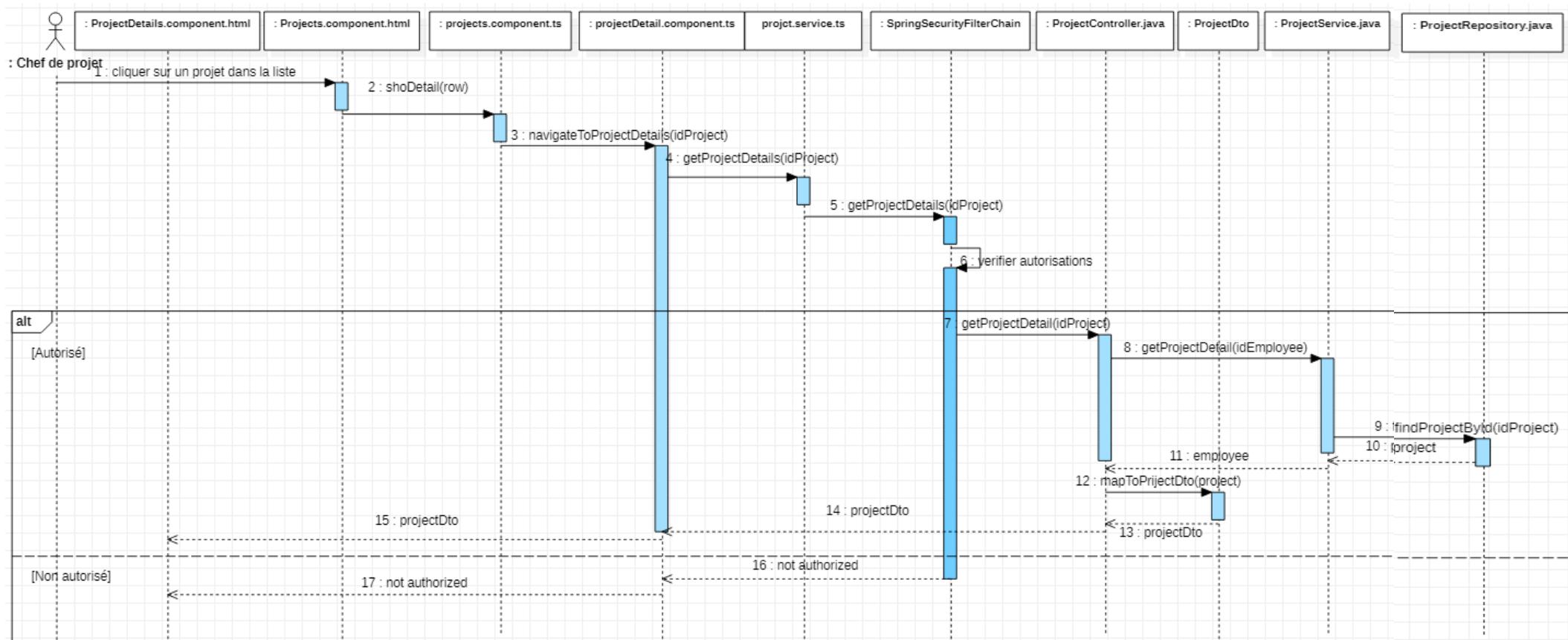


Figure 106 : Diagramme de séquence du cas d'utilisation « Consulter détail projet »

- La figure 107 illustre le diagramme d'activités du cas d'utilisation « Modifier un projet ».

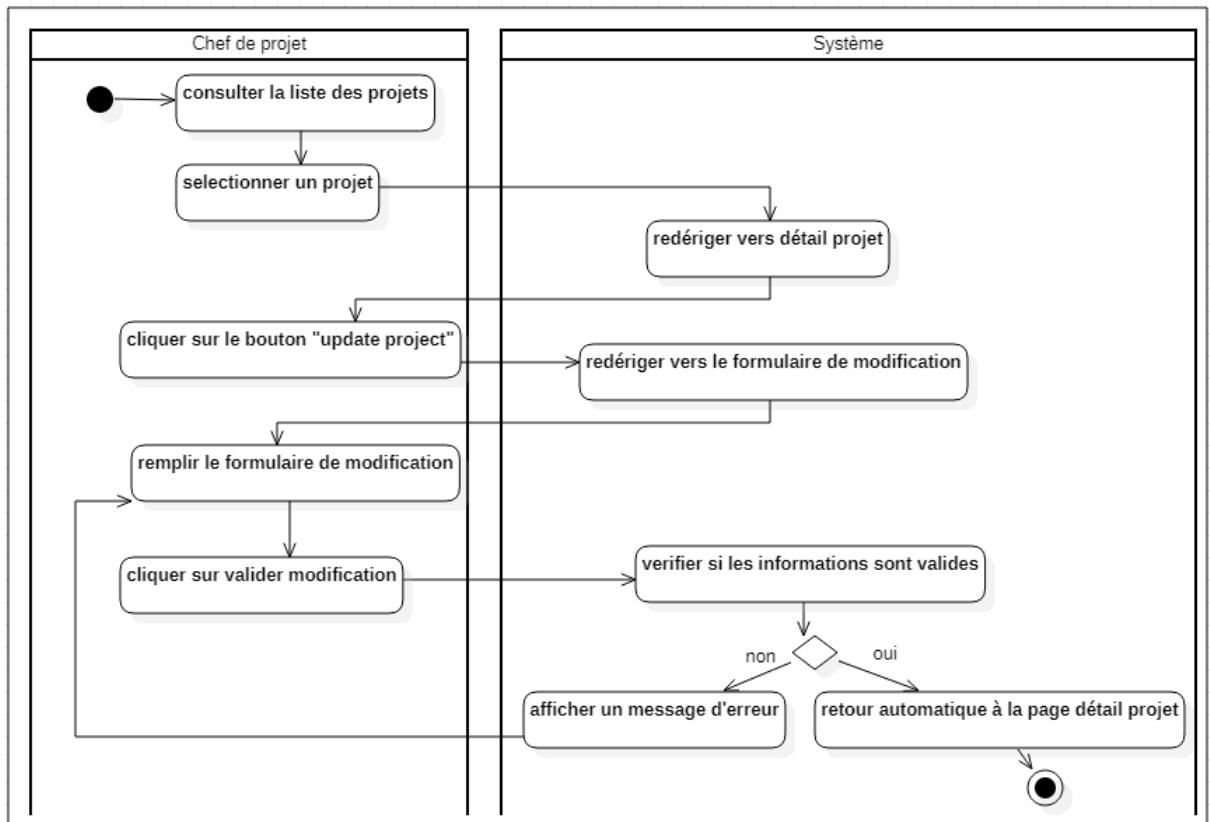


Figure 107 : Diagramme d'activités du cas d'utilisation « Modifier un projet »

- **Réalisation**

### Les cas d'erreurs

Le système est lié à des validateurs pour le formulaire d'ajout du projet, si les exigences ne sont pas respectées, le système déclenche des messages d'erreurs :

- Si les champs sont vides
- Si la date de fin du projet est inférieure à la date de début

Les cas d'erreurs sont présentés par l'interface 108.

The figure consists of two side-by-side screenshots of a 'Create Project' form. Both screenshots have an orange header bar with the title 'Create Project' and a close button 'x' in the top right corner.

**Left Screenshot (Error Case):**

- Project Name:** A red-bordered input field with the placeholder 'name is required'.
- started Date:** A date input field with the placeholder 'jj/mm/aaaa --:--' and a red error message 'enter started date'.
- end Date:** A date input field with the placeholder 'jj/mm/aaaa --:--' and a red error message 'Enter a due date'.
- Add files:** A file selection input field with the placeholder 'Sélect. fichiers' and 'Aucun fichier choisi'.
- Description:** A large text area.
- Assign Project to project leader:** A section with a placeholder 'PROJECT LEADER LIST' and a search bar 'Search Project Leader...'. It shows three user profiles with names: WISSAL FERCI, HENI OUELHEZI, and YASMINE BEN SLIMENE. Each profile has a small orange square icon next to it.
- Bottom Buttons:** 'Add' (orange) and 'Cancel' (red).
- Message:** 'Assign project to Project Leader please'

**Right Screenshot (Error Case):**

- Project Name:** An input field containing 'Rapport PFE application « le Web » agence de voyage'.
- started Date:** A date input field with the value '05/08/2021 17:00'.
- end Date:** A date input field with the value '04/08/2021 17:00' and a red error message 'enter a date after the started date'.
- Add files:** A file selection input field with the placeholder 'Sélect. fichiers' and 'Aucun fichier choisi'.
- Description:** A large text area.
- Assign Project to project leader:** A section with a placeholder 'PROJECT LEADER LIST' and a search bar 'Search Project Leader...'. It shows three user profiles with names: WISSAL FERCI, HENI OUELHEZI, and YASMINE BEN SLIMENE. Each profile has a small orange square icon next to it.
- Bottom Buttons:** 'Add' (orange) and 'Cancel' (red).
- Message:** 'Assign project to Project Leader please'

Figure 108 : Interface des cas d'erreurs du cas d'utilisation « Crée un projet »

## La liste des projets

Un utilisateur connecté et ayant le rôle administrateur, peut accéder à la liste de tous les projets.

Cette interface est illustrée par la figure 109.

The figure is a screenshot of a web-based project management application. The left sidebar includes navigation links for 'Operations' (selected), 'Employees', 'Charts', 'Projects' (selected), 'Team', 'Notification', 'EXAMPLES', 'Mailbox', and 'Pages'.

The main content area displays a summary of project status with four cards:

- Total Project: 23
- Stopped project: 3
- Completed project: 2
- Paused project: 1

Below this is a table with columns: #, Projects, Start Date, Due Date, Status, Team, Progress, and Action. The table contains three rows of data:

#	Projects	Start Date	Due Date	Status	Team	Progress	Action
1	Food spoilage detector	10 July,2021	25 July,2021	completed		Progress 100%	
2	test notif	08 August,2021	22 August,2021	pending	This project does not have team yet!	Progress 0%	
3	gestion de projet	31 July,2021	20 August,2021	processing		Progress 0%	

Annotations highlight specific features:

- Modification statut:** Points to the 'Status' column in the table.
- Pagination:** Points to the page navigation at the bottom, showing pages 1 through 8.

Figure 109 : Interface de consultation de la liste des projets

## - Test logiciel

Nous présentons dans cette section le résultat du test unitaire de la fonctionnalité

« getProjectDetail(id) » relatif au sprint 1 « Gestion et planification des projets ». En effet, à l'aide du plugin EclEmma nous pouvons savoir si la méthode a été testée ou pas (figure 110).

```
--> 264     public ProjectListDto getProjectDetail(long id) {  
265  
266         float progress = getProgressType(id);  
267         return projectRepository.findById(id).map(p -> {  
268             return new ProjectListDto(p, progress);  
269         }).orElseThrow(() -> new NotFoundException("Project Not Found"));  
270     }  
271  
272 }
```

Figure 110 : Méthode « getProjectBy(id) » testée

De même aussi que la release 1, nous faisons toujours des revues de qualité du code à l'aide de l'outil SonarLint. La figure montre l'analyse appliquée sur la partie tâche.

SonarLint Report		
Filter matched 100 of 120 items		
Resource	Date	Description
SecurityUtils.java	8 days ago	Replace this if-then-else statement by a single return statement.
SecurityUtils.java		Define a constant instead of duplicating this literal "manager" 5 times. [+5 locations]
Status.java	21 minutes ago	Use classes from the Java API instead of Sun classes.
Task.java		Remove this unused import 'java.time.LocalDate'.
Task.java		Remove this unused import 'java.time.format.DateTimeFormatter'.
TaskController.java		Replace "@RequestMapping(method = RequestMethod.POST)" with "@PostMapping" [-1]
TaskPriority.java	8 days ago	Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*([A-Z0-9]+)*'
TaskPriority.java		Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*([A-Z0-9]+)*'
TaskPriority.java		Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*([A-Z0-9]+)*'
TaskRepository.java	20 minutes ago	Remove this unused import 'org.springframework.data.jpa.repository.Query'.
TaskRepository.java		Remove this unused import 'com.billcom.app.entity.UserApp'.
TaskRepository.java		Remove this unused import 'java.util.Optional'.
TaskService.java	1 hour ago	Make this anonymous inner class a lambda
TaskService.java	1 hour ago	Remove this unused import 'java.io.Console'.
TaskService.java		Complete the task associated to this TODO comment.
TaskService.java		Complete the task associated to this TODO comment.

Figure 111 : Analyse du SonarLint

## - Documentation

Durant ce sprint, nous avons, de la même façon que la release précédente, généré la documentation des contrôleurs que nous avons implémentés, à l'aide de l'outil SwaggerUI.

La figure 112 présente la documentation de la méthode createTask(taskDto).

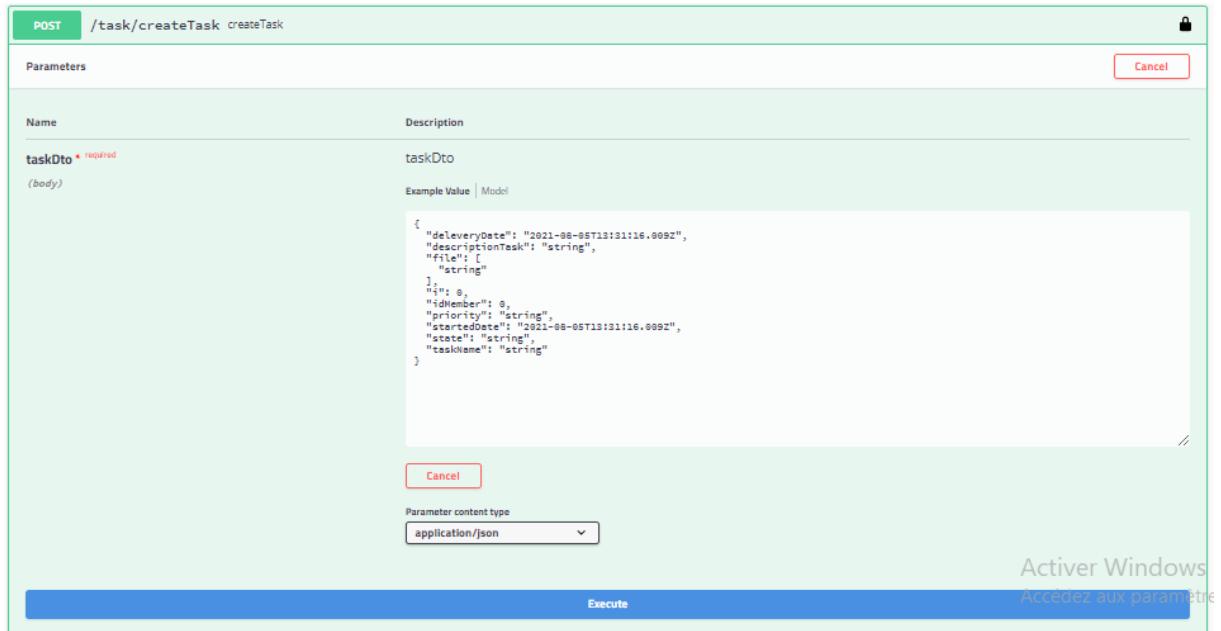
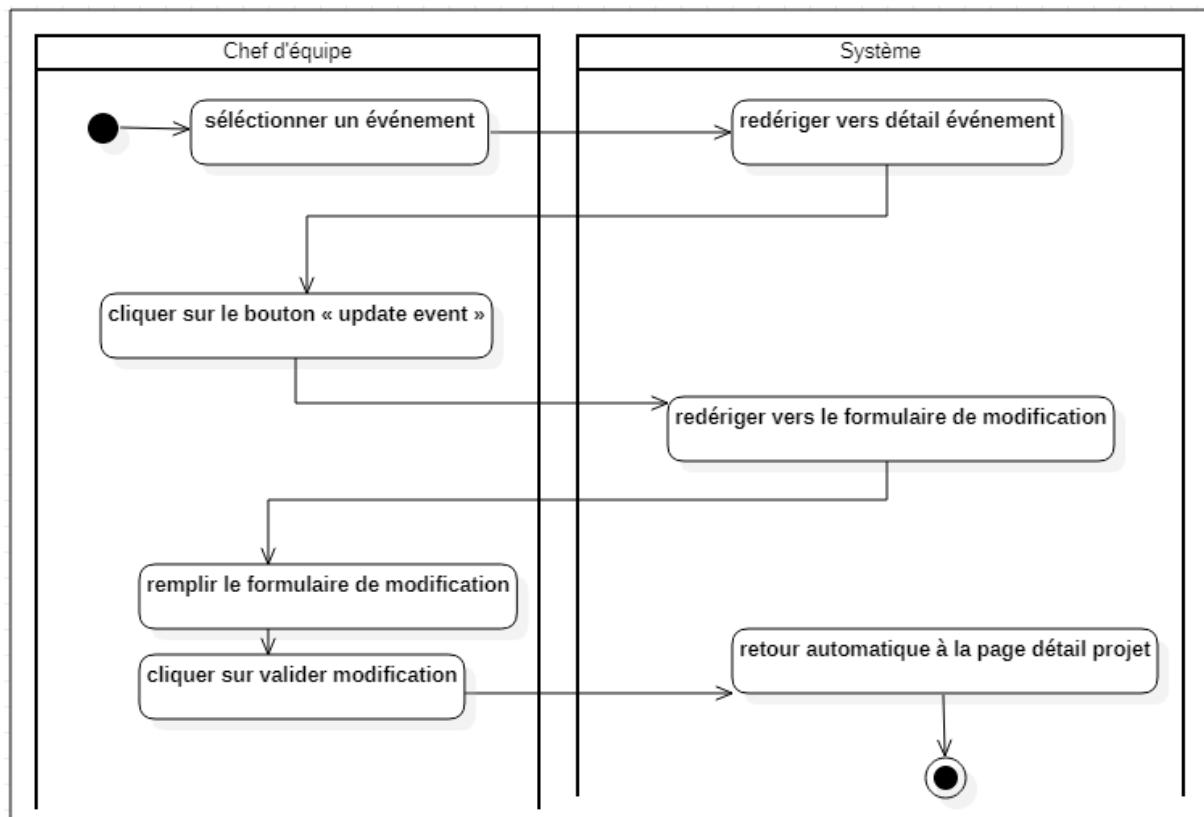


Figure 112 : Documentation SwaggerUI de la release 2

- **Release 3**

- **Sprint 1**

Le diagramme d'activités du cas d'utilisation « Modifier un événement » est présenté par la figure 113.



**Figure 113 : Diagramme d'activités du cas d'utilisation « Modifier un événement »**

## Résumé

Ce projet s'inscrit dans le cadre d'un projet de fin d'études réalisé au sein de la société « Billcom Consulting » afin d'obtenir le diplôme national d'ingénieur en informatique. L'objectif de ce projet consiste à concevoir, développer et déployer une application de gestion de projet. Cette solution permet aux utilisateurs de l'application de planifier les projets et leurs suivi en se référant à des tableaux de bord. Elle offre également l'opportunité de discuter en temps réel et de gérer les notifications. Pour conduire ce projet nous avons utilisé la méthodologie Agile Scrum et pour le réaliser nous avons utilisé les langages Java et TypeScript, les deux frameworks springBoot et angular, pour le tester nous avons utilisé le framework Junit pour les tests unitaire et selenium pour les tests automatisés, et pour le déployer nous avons utilisé la plateforme heroku et l'outil jenkins pour la mise en place de la chaîne CI/CD.

**Mots Clés:** Scum, SpringBoot, angular, webSocket, Chart js, KPI, Heroku, Jenkins, Junit

## Abstract

This project is part of an end-of-studies project carried out within the company « Billcom Consulting » in order to obtain the national diploma in computer engineering. The objective of this project is to design, develop and deploy a project management application. This solution allows users of the application to plan projects and their follow-up by referring to dashboards. It also offers the opportunity to chat in real time and manage notifications. To lead this project we used the Agile Scrum methodology and to carry it out we used the Java and TypeScript languages, both springBoot and Angular frameworks, to test it. We used the Junit framework for unit testing and selenium for automated testing, and to deploy it we used the heroku platform and the jenkins tool for setting up the pipeline CI/CD.

**Keywords:** Scum, SpringBoot, angular, webSocket, Chart js, KPI, Heroku, Jenkins, Junit

## الملخص

هذا المشروع هو مشروع نهاية الدراسات للحصول على الشهادة الوطنية ل الهندسة الاعلامية و تم تنفيذه داخل شركة "Billcom Consulting". الهدف من هذا المشروع هو تصميم و تطوير و اطلاق تطبيق لادارة المشاريع. هذا التطبيق يساعد على تخطيط المشاريع و متابعتها من خلال لوحات مراقبة و يوفر ايضا الفرصة لمستعملي التطبيق التواصل عبر خاصية الارسالات السريعة لقيادة هذا المشروع استخدمنا منهجه Agile Scrum و لتنفيذه استخدمنا لغة Java و TypeScript و لاطلاقه استخدمنا منصة Heroku .

**الكلمات الدالة:** Scum, SpringBoot, angular, webSocket, Chart js, KPI