

Dédicaces

Je dédie ce modeste travail :

À l'âme de ma mère,

À mon père,

*En témoignage de ma reconnaissance pour tous les efforts qu'il a consentis
pour mon éducation.*

À mes soeurs Naima et Nourhen,

En reconnaissance de leurs affections et leurs encouragements.

À mon cher fiancé Mouhamed,

*Merci énormément pour ton soutien plus que précieux, Merci pour ton grand
coeur toutes vos qualités qui seraient trop longues à énumérer. Ma vie ne
serait pas aussi magique sans ton présence et ton amour.*

Je t'aime de tout mon coeur .

À mes amis,

*Tous mes fidèles amis et tous ceux qui sont chers pour leur chaleur, leur
gaieté et leur soutiens moral qu'ils trouvent dans ce travail le témoignage de
ma profonde gratitude.*

*Sans oublier Tous les professeurs que ce soit de primaire, secondaire ou de
l'enseignement supérieur.*

Nesrine Rabah

Remerciements

Avant de commencer la présentation de ce rapport, je profite l'occasion pour remercier à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

En premier lieu, je tiens à remercier du fond du cœur mon encadrant académique **Mr. Nidhal Jelassi** de m'avoir guidée durant toute la période d'encadrement. Je tiens à lui exprimer toute ma reconnaissance et mon admiration .

Je remercie également mon encadrant professionnel **Mme. Zeineb Hadj Ali** et toute la famille **Barsha Technology** pour leur accueil chaleureux et pour l'expérience qu'ils m'ont offert au sein de société.

Je remercie vivement tous les membres du jury qui m'ont honoré par leur présence et en acceptant d'évaluer mon travail.

Finalement, nous tenons aussi, à remercier tous les enseignants qui ont contribué de près ou de loin à notre formation.

Nesrine Rabah

Table des matières

Dédicaces	3
Remerciements	4
Introduction générale	13
1 Cadre général du projet	15
Introduction	16
1.1 Présentation de l'entreprise	16
1.2 Problématique	18
1.3 Etude de l'existant	19
1.3.1 Critique de l'existant	20
1.4 Solution proposée	22
1.5 Identification des acteurs et des cas d'utilisation	23
1.6 Specification des besoins	25
1.6.1 Les besoins fonctionnels	25
1.6.2 Les besoins non fonctionnels	26
1.7 Méthodologie adoptée	27
1.7.1 Méthodes Agiles	27
1.7.2 Tableau Comparatif	28
1.7.3 Choix de la méthode Scrum	30
Conclusion	31
2 ETUDE PRÉALABLE	32
Introduction	34
2.1 Pilotage du projet avec Scrum	34
2.1.1 Équipe et rôle	34
2.1.2 Backlog produit	35

2.1.3	Diagramme de cas d'utilisation général	42
2.2	Planification des Sprints	43
2.3	Architecture de l'Application	44
2.3.1	Couche présentation	45
2.3.2	Couche Métier	45
2.3.3	Couche Données	47
2.4	Environnement de travail	48
2.4.1	Environnement matériel	48
2.4.2	Environnement logiciel	48
2.4.3	Technologie utilisées :	50
	Conclusion	53
3	Sprint 1	54
	Introduction	56
3.1	Backlog du sprint 1	56
3.2	Spécifications fonctionnelles	58
3.2.1	Raffinement de cas d'utilisation « S'inscrire »	58
3.2.2	Raffinement de cas d'utilisation « S'authentifier »	59
3.2.3	Raffinement de cas d'utilisation « Gérer Utilisateurs »	61
3.2.4	Raffinement de cas d'utilisation « Modifier Profil »	64
3.3	Conception	66
3.3.1	Diagramme de classes du premier sprint	66
3.3.2	Diagramme de séquence du cas d'utilisation « S'authentifier »	67
3.3.3	Diagramme de séquence du cas d'utilisation « Modifier utilisateur »	68
3.3.4	Diagramme de séquence du cas d'utilisation « Supprimer utilisateur »	70
3.4	Réalisation	71
3.4.1	Interface d'inscription	71
3.4.2	Interface d'authentification	72
3.4.3	Interface de détail utilisateur	73
3.4.4	Interface de modification utilisateur	74
3.4.5	Interface de profil utilisateur	75
3.4.6	Interface de modification mot de passe	76
	Conclusion	77

4	Sprint 2	78
	Introduction	80
4.1	Backlog du sprint 2	80
4.2	Spécifications fonctionnelles	84
4.2.1	Raffinement de cas d'utilisation « Gérer Projets » . . .	84
4.2.2	Raffinement de cas d'utilisation « Gérer Sprints » . . .	86
4.2.3	Raffinement de cas d'utilisation « Gérer Tâches » . . .	88
4.3	Conception	90
4.3.1	Diagramme de classes du deuxième sprint	90
4.3.2	Diagramme de séquence du cas d'utilisation « Supprimer projet »	91
4.3.3	Diagramme de séquence du cas d'utilisation « Modifier sprint »	92
4.3.4	Diagramme de séquence du cas d'utilisation « Consulter tâche »	93
4.3.5	Diagramme de séquence du cas d'utilisation « Modifier état tâche »	94
4.4	Réalisation	95
4.4.1	Interface d'ajouter projet	95
4.4.2	Interface de modification projet	96
4.4.3	Interface de détail projet	97
4.4.4	Interface liste des sprints	98
4.4.5	Interface de modification sprint	99
4.4.6	Interface de modification d'état sprint	100
4.4.7	Interface de suppression un sprint	101
4.4.8	Interface d'ajouter tâche	102
4.4.9	Interface de modification d'état d'une tâche	103
4.4.10	Interface liste des tâches	103
4.4.11	Interface de détail tâche	104
4.4.12	Interface du Tableau de bord	105
	Conclusion	106
5	Sprint 3	107
	Introduction	108
5.1	Backlog du sprint 3	108
5.2	Spécifications fonctionnelles	110

5.2.1	Raffinement de cas d'utilisation «Gérer Commentaires»	110
5.2.2	Raffinement de cas d'utilisation « Gérer Attachements »	112
5.3	Conception	114
5.3.1	Diagramme de classes du troisième sprint	114
5.3.2	Diagramme de séquence du cas d'utilisation « Ajouter commentaire »	115
5.3.3	Diagramme de séquence du cas d'utilisation « Suppri- mer commentaire »	116
5.3.4	Diagramme de séquence du cas d'utilisation « Ajouter Attachement »	117
5.4	Réalisation	117
5.4.1	Interface gérer commentaires	118
5.4.2	Interface gérer attachemnts	119
	Conclusion	120
6	Sprint 4	121
	Introduction	122
6.1	Backlog du sprint 4	122
6.2	Spécifications fonctionnelles	125
6.2.1	Raffinement de cas d'utilisation «Gérer Documents» .	125
6.2.2	Raffinement de cas d'utilisation «Gérer Evènements» .	126
6.3	Conception	127
6.3.1	Diagramme de classes du quatrième sprint	127
6.3.2	Diagramme de séquence du cas d'utilisation « Ajouter évènement »	129
6.4	Réalisation	129
6.4.1	Interface ajouter document	129
6.4.2	Interface liste des documents	130
6.4.3	Interface liste des documents archivés	131
6.4.4	Interface restaurer document archivé	132
6.4.5	Interface ajouter évènement	133
	Conclusion	134
	Conclusion Générale et Perspectives	135
	Netographie	136

Table des figures

1.1	barsha Technology Logo	16
1.2	Problématique-Outils classiques de gestion des tâches	18
1.3	Problématique-Outils classiques de gestion des documents	19
1.4	Etude de l'existant-Travail d'équipe classique	20
1.5	Critique de l'existant – Microsoft 365 Planner	21
1.6	Fonctionnement de SCRUM	30
2.1	Diagramme de cas d'utilisation général	42
2.2	Architecture de l'Application	44
3.1	Diagramme du cas d'utilisation « S'inscrire »	58
3.2	Diagramme du cas d'utilisation « S'authentifier »	60
3.3	Diagramme du cas d'utilisation «Gérer Utilisateurs »	61
3.4	Diagramme du cas d'utilisation «Modifier Profil »	64
3.5	Diagramme de classes « Premier Sprint »	66
3.6	Diagramme de séquence du cas d'utilisation « S'authentifier »	67
3.7	Diagramme de séquence du cas d'utilisation « Modifier utilisateur »	69
3.8	Diagramme de séquence du cas d'utilisation « Supprimer utilisateur »	70
3.9	Interface d'inscription	71
3.10	Interface d'authentification	72
3.11	Interface de détail utilisateur	73
3.12	Interface de modification utilisateur	74
3.13	Interface de profil utilisateur	75
3.14	Interface de modification mot de passe	76
4.1	Diagramme du cas d'utilisation «Gérer Projets »	84

4.2	Diagramme du cas d'utilisation «Gérer Sprints »	86
4.3	Diagramme du cas d'utilisation « Gérer Tâches »	88
4.4	Diagramme de classes « Deuxième Sprint »	90
4.5	Diagramme de séquence du cas d'utilisation « Supprimer Pro- jet »	91
4.6	Diagramme de séquence du cas d'utilisation « Modifier sprint »	92
4.7	Diagramme de séquence du cas d'utilisation « Consulter tâche »	93
4.8	Diagramme de séquence du cas d'utilisation « Modifier état tâche »	94
4.9	Ajouter projet	95
4.10	l'interface de modification projet	96
4.11	l'interface de détail projet	97
4.12	Interface liste des sprints	98
4.13	l'interface de modification sprint	99
4.14	l'interface de modification d'état sprint	100
4.15	Suppression sprint avec permission	101
4.16	Suppression sprint sans permission	101
4.17	Ajouter tâche	102
4.18	l'interface de modification d'état d'une tâche	103
4.19	Interface liste des tâches	104
4.20	l'interface de détail tâche	104
4.21	l'interface du Tableau de bord	105
5.1	Diagramme du cas d'utilisation « Gérer Commentaires » . . .	110
5.2	Diagramme du cas d'utilisation« Gérer Attachements»	112
5.3	Diagramme de classes« Troisième Sprint »	114
5.4	Diagramme de séquence du cas d'utilisation « Ajouter com- mentaire »	115
5.5	Diagramme de séquence du cas d'utilisation « Supprimer com- mentaire »	116
5.6	Diagramme de séquence du cas d'utilisation « Ajouter Atta- chement »	117
5.7	Interface gérer commentaires	118
5.8	Interface gérer attachemnts	119
6.1	Diagramme du cas d'utilisation « Gérer Documents »	125

6.2	Diagramme du cas d'utilisation « Gérer Evènements »	126
6.3	Diagramme de classes« Quatrième Sprint »	128
6.4	Diagramme de séquence du cas d'utilisation « Ajouter évènement»	129
6.5	Interface ajouter document	130
6.6	Interface liste des documents	131
6.7	Interface liste des documents archivés	131
6.8	Interface restaurer document archivé	132
6.9	Interface ajouter évènement	133

Liste des tableaux

1.1	Tableau Description des acteurs	24
1.2	Scrum VS Kanban VS Scrumban	28
2.1	Backlog Produit	41
2.2	Planification des Sprints	43
2.3	Tableau des mots-clés	47
3.1	Premier sprint Backlog	57
3.2	Description textuelle du cas d'utilisation « S'inscrire »	59
3.3	Description textuelle du cas d'utilisation « S'authentifier » . .	61
3.4	Description textuelle du cas d'utilisation « Désactiver compte utilisateur»	62
3.5	Description textuelle du cas d'utilisation «Supprimer utilisateur»	63
3.6	Description textuelle du cas d'utilisation«Modifier Profil» . . .	65
4.1	Deuxième sprint Backlog	83
4.2	Description textuelle du cas d'utilisation «Ajouter Projet» . .	85
4.3	Description textuelle du cas d'utilisation«Modifier état sprint»	87
4.4	Description textuelle du cas d'utilisation «Supprimer Tâche» .	89
5.1	Troisième sprint Backlog	109
5.2	Description textuelle du cas d'utilisation«Ajouter Commentaire»	111
5.3	Description textuelle du cas d'utilisation«Télécharger Attache- ment»	113
6.1	Quatrième sprint Backlog	124
6.2	Description textuelle du cas d'utilisation«Restaurer Document»	126
6.3	Description textuelle du cas d'utilisation «Ajouter évènement»	127

Introduction générale

Dans le cadre de la formation de Master Professionnel en Ingénierie des systèmes d'information des Entreprises à la Faculté des Sciences Economiques et de Gestion de Tunis, nous sommes appelés à consolider notre formation théorique par des connaissances et des acquis pratiques à travers des projets dont le plus évaluatif en terme de contenu de la formation est celui de fin d'études, c'est dans ce cadre que s'articule notre application que nous sommes appelés à développer (concevoir et réaliser).

De nos jours, toute entreprise est prête à investir des sommes colossales dans l'implantation des technologies logicielles en vue d'améliorer ses services, d'accroître son agilité et sa flexibilité, de réduire les coûts, d'augmenter la production et de faire face aux défis du marché. Vu la croissance des activités au sein des entreprises, la tâche de gérer efficacement toutes les fonctions s'avère de plus en plus complexe et difficile.

De plus, L'équipe de développement a besoin des méthodes axées sur les personnes pour objectif, la satisfaction du client en s'appuyant sur la réalisation d'un logiciel entièrement fonctionnel tout au long de sa fabrication.

Aussi comme la réalisation d'un logiciel, un bon système de gestion documentaire, correctement communiqué à l'ensemble des employés est donc un outil précieux, voire essentiel au bon fonctionnement d'une entreprise. Ce n'est plus à l'utilisateur d'aller à la recherche du document, mais c'est le document qui vient directement vers l'utilisateur.

C'est dans cette optique que se situe notre projet de fin d'études. Il s'agit de développer une application inter service multiplateforme modulaire et qui applique la méthode Agile utilisant « SCRUM » pour bien planifier l'ensemble

des projets.

Le présent rapport s'articule en six chapitres :

- Dans le premier chapitre « cadre du projet », nous expliquons le cadre de notre projet, la problématique posée, l'étude de l'existant et la solution proposée. Ainsi, nous présentons les différents acteurs, les besoins fonctionnels et non fonctionnels . Par la suite, nous présentons la méthodologie choisie pour la réalisation de notre application.
- Dans le deuxième chapitre « Etude préalable », nous commençons par la présentation du pilotage du projet avec Scrum. Ensuite, nous modélisons notre application à travers le diagramme de cas d'utilisation général. Nous présentons, par la suite, la planification des sprints et l'architecture de l'application. Enfin, nous détaillons l'environnement matériel ,l'environnement logiciel et tous les technologies qui nous ont permis d'atteindre ce résultat.
- Du troisième jusqu'au sixième les chapitres sont élaborés selon une même démarche. Ces chapitres sont les réalisations des sprints qui comportent chacun quatre sections : le sprint backlog ,spécifications fonctionnelles, conception et réalisation.
- Enfin, ce rapport sera clôturé par une conclusion générale qui établit le synthèse de travail et quelques perspectives envisageables au présent projet.

1

Cadre général du projet

Sommaire

Introduction	16
1.1 Présentation de l'entreprise	16
1.2 Problématique	18
1.3 Etude de l'existant	19
1.3.1 Critique de l'existant	20
1.4 Solution proposée	22
1.5 Identification des acteurs et des cas d'utilisation	23
1.6 Spécification des besoins	25
1.6.1 Les besoins fonctionnels	25
1.6.2 Les besoins non fonctionnels	26
1.7 Méthodologie adoptée	27
1.7.1 Méthodes Agiles	27
1.7.2 Tableau Comparatif	28
1.7.3 Choix de la méthode Scrum	30
Conclusion	31

Introduction

L'étude de projet est une démarche stratégique visant à organiser le bon déroulement d'un projet et d'assurer la conduite de toutes les phases qui le constituent. Cette étude fera donc l'objet de ce chapitre dans lequel nous allons présenter le cadre général du projet. Il s'agit en effet d'une présentation de l'organisme d'accueil, de la problématique du projet, ainsi que de l'étude de l'existant. Après l'exposition de l'étude de l'existant, nous allons présenter la solution proposée. Puis, Nous allons identifier les fonctionnalités et les acteurs du système. Enfin, nous justifierons notre choix concernant la méthodologie de gestion de projet adoptée pour la réalisation de notre application à travers une étude comparative des principales méthodes agiles.

1.1 Présentation de l'entreprise

Barsha Technology est une société de services technologiques et de production audio visuelle fondée en septembre 2017 par Madame Zeineb Hadj Ali , ces services de base proposés sont le développement d'applications mobile , la création de sites Web , la conception graphique et les applications bureautiques.



FIGURE 1.1 – barsha Technology Logo

Les principaux produits développés par l'équipe Barsha Technology :

- **Faatek :**

est le plus grand site du genre au Royaume d'Arabie Saoudite pour promouvoir les meilleures offres et coupons.

- **Spareapp :**

Plateforme électronique qui vous offre de nombreuses options liées aux pièces de rechange de votre voiture, qu'elles soient d'origine ou commerciale, la seule plateforme qui vous donne des prix différents avant d'acheter des pièces de rechange, vous pouvez donc choisir entre les options disponibles.

- **Alemni :**

Plateforme électronique pour enseigner la langue arabe à travers des formateurs de différentes nationalités et de nombreuses langues et pays. Cette plate-forme vise à diffuser la langue arabe dans le monde entier.

- **YouCan :**

Plateforme de formation électronique qui s'intéresse à la formation à distance ciblant les propriétaires de contenu et ceux qui souhaitent développer leurs compétences. Sa mission : Développer et éduquer les individus dans tous les domaines et partout grâce aux technologies modernes.

- **Barsha Food :**

Application mobile et site web pour la livraison de nourriture

1.2 Problématique

« Le client c'est le roi » ce terme est très utilisé dans le monde professionnel dont la satisfaction de client est le but numéro un de l'entreprise, et le plus connu que le client peut changer d'avis concernant quelques descriptions données d'avance au cahier de charge. De plus même si la réalisation de projet se fait parfaitement elle peut être pas bien compréhensible chez lui d'où la refaire de réalisation encore fois. Tous ça cause la perte de l'effort des employés ainsi le gaspillage de temps d'où la diminution de gain. Ça implique que les méthodes de gestion des projets traditionnelles s'embranchent pas le changement et échoue à délivrer un produit pleinement conforme à l'attendu.

De plus, les tests et le cycle(itération) sont rarement archivés (stockés) d'où la perte de trace d'effort ainsi l'insuivi d'une étape importante au cycle de développement.

Ainsi une mauvaise organisation des tâches peut mener à des résultats chaotiques qui sont catastrophiques pour la productivité des entreprises et une mauvaise communication ou l'absence de communication au sein d'une équipe qui gèrent des tâches peut causer l'échec d'un projet.

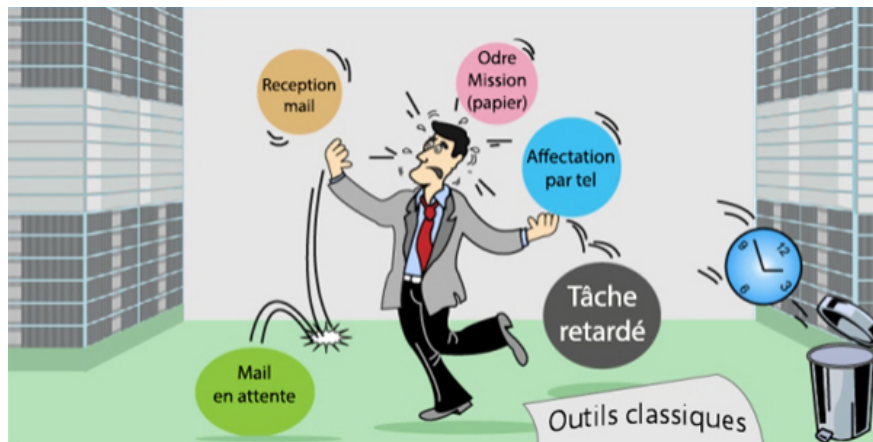


FIGURE 1.2 – Problématique-Outils classiques de gestion des tâches

Comme la gestion des tâches, Le stockage des données physiques reste encore problématique dans la plupart des sociétés et les expose à d'importants risques de perte ou de fuite d'informations de valeur.

La plupart des entreprises gardent leurs archives papier dans des caissons ou dans les tiroirs de leurs bureaux. Cette manière de classer pose des problèmes lors de la recherche d'informations qui peut durer plusieurs minutes, voire plusieurs heures, ce qui rend inefficace les entreprises.

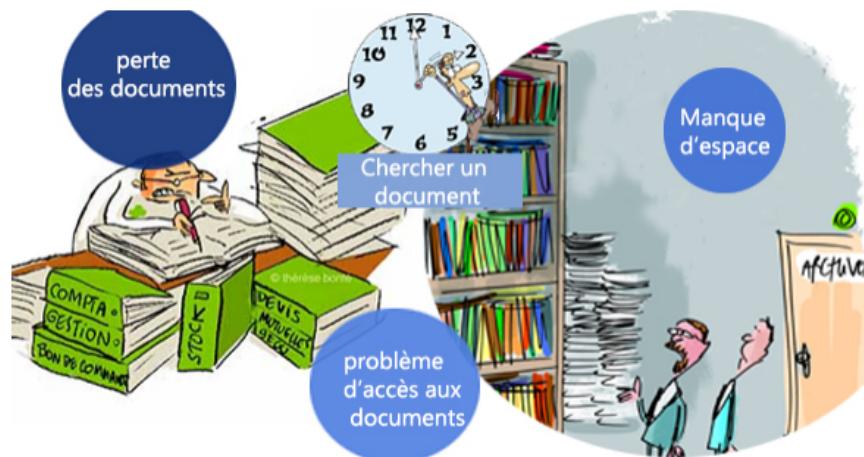


FIGURE 1.3 – Problématique-Outils classiques de gestion des documents

1.3 Etude de l'existant

Dans une approche traditionnelle du monde de génie logiciel, les projets informatiques abordent la gestion de projets comme les autres disciplines.

On y retrouve la planification, l'organisation, le suivi, l'affectation du personnel, les contrôles et la direction les tests et les versions de logiciel.

Au fil du temps, et vu le grand nombre de projet élaborés chaque années, le suivi manuel des projets n'est plus efficace et rentable, il aura des conflits au sein de l'entreprise qui va mener à un retard dans le rendu d'un livrable et donc cela va automatiquement conduire à des pertes financières et des clients. Le projet consiste à résoudre ce problème qui va offrir un espace de travail collaboratif qui organise et facilite le suivi des projets.



FIGURE 1.4 – Etude de l'existant-Travail d'équipe classique

1.3.1 Critique de l'existant

Suite à l'étude de l'existant, nous avons constaté ce qui suit :

- Le processus de gestion des projets étant manuel par conséquent il aura une perte du temps.
- Le processus de gestion des projets n'est pas simple à effectuer manuellement donc cela va conduire à une confusion entre les développeurs.
- Absence d'une solution informatisé pour la synchronisation et le suivi des tâches entre toute l'équipe.
- La suppression des tâches est parfois trop facile : une mauvaise manipulation peut entrainer une suppression sans s'en rendre compte.
- Les solutions alternatives sont payées et ne reponds pas aux besoins spécifique des entreprises.
- Plusieurs organisations ou groupes qui développent des logiciels, ne connaissent que les approches traditionnelles qui sont souvent lourdes et coûteuses à gérer.
- On critique principalement le dépassement de coûts et les délais de livraison, ce qui amène les petites équipes à délaisser ces approches. À défaut, peu de méthodes alternatives sont présentées.

Tenant l'exemple de « Microsoft 365 Planner », c'est l'un des applications fameuses qui permettent d'organiser les projets ainsi de diriger les différents membres d'équipe en ligne (vous pouvez le tester : <https://tasks.office.com>). Mais l'abonnement est encore couteux ainsi que le client ne peut pas s'interagit avec l'application utilisé par l'équipe, cet exclut peut causer le non-satisfaire de client à la livraison du projet.



FIGURE 1.5 – Critique de l'existant – Microsoft 365 Planner

1.4 Solution proposée

Notre solution consiste à mettre en œuvre une application inter service qui facilite au entreprise la gestion des projets, sprints, tâches et La gestion électronique de documents qui permet de remplacer des "documents papier" par leur représentation sous forme de "documents électroniques", en réponse à la fois aux problèmes d'archivage et de recherche des documents complets.

La gestion des projets sur cette outil va être suivant la méthodologie Scrum qui est une méthode Agile, cette méthodologie impose la division d'un projet en plusieurs cycles de travail relativement courts afin de mieux planifier les prochaines étapes de développement du projet et d'évaluer régulièrement les progrès liés au projet.

De plus, nous intégrons le Product-Owner pour utiliser aussi l'application, cela va permet au client d'interagir à n'importe quel moment avec l'entreprise de développement ainsi le suivie de son projet. Aussi, nous tenions notre intérêt à faire une application Web ainsi que la mettre publique sur le net pour que l'utilisateur vive une expérience de qualité et sans limites.

Nous chercherons donc à travers cette application à atteindre les objectifs suivants :

- Gagnez en agilité.
- Centraliser les outils.
- Optimiser la gestion du temps.
- Assigner aisément les différentes tâches avec une date butoir, la description du travail, des commentaires, des pièces jointes. . .
- Mieux communiquer ,Moins d'emails, optimisation du temps, moins de réunions, plus de temps pour la production et les idées.
- Assuré la sécurité de l'information et l'accès au document.
- Un espace sociaux collaboratif qui permet de crée des évènements.

- Doit être disponible et consultable de partout (tablette, mobile ou ordinateur).

1.5 Identification des acteurs et des cas d'utilisation

Appliquant SCRUM on identifie les acteurs principaux de l'application :

- **Administrateur** : Celui qui gère tous les utilisateurs et les composants de système.
- **Scrum Master** : C'est l'intermédiaire entre le Product Owner et l'équipe
- **Chef d'équipe** : Celui qui Facilite la communication au sein de l'équipe.
- **Membre d'équipe** : Tous les membres de l'équipe apportent leur savoir-faire pour accomplir les tâches
- **Product-Owner** : Celui qui joue le rôle du client.

Le tableau 1.1 présente la description des acteurs

Acteur	Définition	Rôle
Administrateur	Il s'agit de la personne qui possède une visibilité totale sur les bases de données. Il spécifie les utilisateurs et les droits de chaque un.	Il est responsable de la gestion de tous les composants de l'application.
Scrum Master	C'est l'intermédiaire entre l'équipe et Product Owner, il gère les projets ainsi l'affectation des sprints au chef d'équipe.	Il s'authentifie afin d'accéder à son propre espace où il pourra créer un Projet, affecter des sprints aux chefs d'équipe, planifier des réunions avec eux.
Chef d'équipe	Celui qui gère l'équipe ainsi les tâches et la modification d'état de sprint.	Il s'authentifie afin d'accéder à l'espace d'équipe où il pourra créer une tâche, modifier état de sprint.
Membre d'équipe	Il peut être soit un développeur, soit un testeur chacun travaille dans un état de tâche (en cours de développement ou en test).	Il fait sa mission (tâche), puis Il s'authentifie afin d'inscrire les changements qu'il a fait.
Product Owner	C'est le client qui attend la finition de projet, il peut s'interagit à n'importe quel moment.	Il se connecte au front end pour consulter ses projets.

TABLE 1.1 – Tableau Description des acteurs

1.6 Specification des besoins

Cette section sert à poser les bases du recueil des besoins du système à réaliser. Pour préciser les différentes tâches des utilisateurs de notre application, nous allons présenter les besoins fonctionnels ainsi que les besoins non fonctionnels.

1.6.1 Les besoins fonctionnels

Mon projet consiste à développer une application entre l'équipe de développement et le Scrum Master et le Product Owner, le système doit permettre aux utilisateurs de gérer d'une manière planifiée et sécurisée les étapes de travail. Les besoins fonctionnels et les attentes vis-à-vis notre application varient d'un acteur à un autre.

- **Les besoins de l'administrateur :**

- Gérer les utilisateurs : Création / suppression, modification des informations personnelles et réinitialisation des mots de passe.
- Gérer les projets et ses sprints et tous les autres composants de system pour l'intervention en cas d'échec ou maintenance.

- **Les besoins d'utilisateur d'application :**

- Gérer son profil.
- Gérer les évènements.
- Gérer les documents.

- **Les besoins de Scrum Master :**

- Gérer les projets, ses sprints et ses tâches.

- **Les besoins de Chef d'équipe :**

- Gérer les tâches .

- **Les besoins de membre d'équipe :**

- Inscrire tous les changements qui concerne les états des tâches .
- Gérer les commentaires.
- Gérer les attachements.

- **Les besoins de Product Owner :**
 - Consulter la progression de son projet à tout moment.

1.6.2 Les besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type SGBD). Les besoins non fonctionnels de notre système, se décrivent comme suit :

- **Contraintes ergonomiques :**
 - La navigation entre les interfaces de notre future application web doit être légère et fluide.
 - Une interface conviviale et ergonomique destinée à tout profil d'utilisateur.
 - La solution doit offrir à l'utilisateur des assistants simples et intuitifs pour ajouter un ou plusieurs documents en une seule opération.
 - La récupération d'accès à l'application doit être simple et rapide.
- **La Fiabilité :**
 - Touche à l'aspect qualité des données et persistance des informations dans l'application .
 - Elle doit fonctionner d'une façon cohérente sans erreur.
 - Les interfaces de l'application doivent s'adapter à la taille des différents écrans qui doit être assuré par un thème responsif.
- **L'Efficacité :**
 - L'application doit permettre l'accomplissement de la tâche avec le minimum de manipulation. Ceci doit être garanti pour qu'elle puisse s'intégrer facilement dans l'entreprise et répondre à leurs besoins.

1.7 Méthodologie adoptée

Sans même en avoir conscience, nous sommes régulièrement en situation de projet. Tous les projets sont dans la nature, à tous les âges, dans tous les domaines, comme organiser une fête, partir en vacances avec la famille, créer son propre emploi, voire sa propre entreprise. Monter un projet c'est tout simplement se fixer un objectif et se donner les moyens de l'atteindre. Les méthodes de conduite de projet se diffèrent selon la méthodologie dont chacun a son point de vue pour réussir un projet durant son cycle de vie.

Dans notre cas, nous allons s'intéresser aux méthodes agiles qui se proclament comme une méthodologie efficace qui représente une approche de gestion de projet et de développement itérative et incrémentale. Pour bien conduire notre projet, nous avons opté pour la méthode agile « Scrum » comme méthodologie de conception et de développement. Après le choix de la méthodologie, nous avons besoin d'un langage de modélisation unifié pour la modélisation de notre projet. Pour cela nous avons choisi l'UML comme un langage de modélisation qui est utilisé par toutes les méthodes agiles y compris la méthode Scrum.

1.7.1 Méthodes Agiles

La méthode Agile est menée dans un esprit collaboratif, elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients, elle se caractérise par une planification adaptative, documentation réduite au strict nécessaire, contrôle qualité précoce et permanent, au niveau du produit et du processus. Le client visualise les résultats tôt et fréquemment. La mesure de succès est comptée par la satisfaction de client. [1]

De nos jours, les méthodes agiles sont légion dans la littérature. Il ne s'agit pas d'en choisir la bonne, mais de sélectionner les meilleures pratiques à mettre en œuvre dans chaque contexte. Elles ont toutefois des affinités avec les différentes typologies de projet et présentent donc un positionnement différent.

1.7.2 Tableau Comparatif

Le choix de la méthode de travail est très important pour fixer les étapes à suivre. Pour ce faire, nous avons établi une comparaison basée par des indicateurs pertinents et sur des études existantes :

Critères	Scrum	Kanban	Scrumban
Itérations	Oui (sprints)	Selon le besoin	Selon le besoin
Rôles	<ul style="list-style-type: none"> • Product Owner • Scrum Master • Team 	Equipe spécialisée	Equipe spécialisée
Changement de périmètre	Doit attendre le sprint suivant	Selon le besoin	Selon le besoin
Planification	Au début de chaque sprint	Kanban board , flux continu	Kanban board avec itérations
Estimation	Oui	Optionnelle	Optionnelle
Règles	Très descriptif	Non descriptif	descriptif
Bords/Artefacts	<ul style="list-style-type: none"> • Product backlog • Scrum board • Burndown chart 	<ul style="list-style-type: none"> • Kanban board • Diagramme des flux cumulés 	<ul style="list-style-type: none"> • Kanban board • Diagramme des flux cumulés

TABLE 1.2 – Scrum VS Kanban VS Scrumban

- **Scrum** : est à ce jour la méthode agile la plus utilisée par les équipes. Ses caractéristiques : des itérations courtes (sprints de 4 semaines maximum), 3 rôles clés dans l'organisation (Scrum master, product owner, équipe de développement) et des timebox. La méthode Scrum est une excellente solution quand il s'agit d'améliorer la productivité d'une équipe, son engagement et la scalabilité des projets. Son caractère cadré et codifié est idéal pour des feuilles de route clairement définies. Scrum est en effet très axé sur le calendrier, contrairement à Kanban. [2]
- **Kanban** : La méthode Kanban est surtout appliquée dans des projets de maintenance corrective ou d'applications en flux. On y trouve par exemple la gestion de tickets ou le support. Le concept de Kanban est en effet basé sur une amélioration continue évolutive. Dans la méthode Kanban, il n'existe pas de rôles attitrés, ces derniers évoluent en fonction des besoins du projet et de l'organisation. [3]
- **Scrumban** : est un mélange des deux concepts « scrum » et « kanban ». Elle conserve l'approche très cadrée de Scrum tout en permettant de mener des projets en flux continu. Scrumban possède une méthode de travail proche de celle de Kanban. Les itérations sont limitées, tout comme les tâches à effectuer, afin de permettre aux équipes de s'adapter et modifier un plan d'actions si nécessaire. La méthode s'appuie également sur une planification basée sur la demande, les équipes Page 10 Chapitre 1 : Cadre du projet ne prévoient plus des tâches pour un sprint complet mais doivent désormais les prioriser au fur et à mesure de l'avancement du projet. [4]

1.7.3 Choix de la méthode Scrum

Pour que la réussite de notre projet soit garantie, nous avons choisi Scrum comme une méthodologie agile. Scrum devrait garantir le succès du projet dans le sens où il permettra d'assurer que le produit final, livré, réponde bien aux besoins du client.). Le principe de base étant d'être toujours prêt à réorienter le projet au fil de son avancement. Il comprend également la possibilité de modifier les exigences formulées et permet d'améliorer la productivité de l'équipe.



FIGURE 1.6 – Fonctionnement de SCRUM

Comme illustré dans la figure ci-dessus cette méthode passe par des étapes, la première étape consiste à formaliser la vision du produit (logiciel) que l'on souhaite réaliser. Cette vision décrit les principaux objectifs, jalons, utilisateurs visés. Elle contribuera à guider et fédérer les acteurs du projet. La suite consiste à établir la liste des exigences fonctionnelles et non fonctionnelles du produit. A la lueur des estimations, la liste ainsi complétée est ordonnancée. Cette liste est appelée le Product Backlog. Le Product Backlog servira à piloter l'équipe de développement et pourra évoluer tout au long du projet. Le

changement est non seulement autorisé mais encouragé afin de pouvoir éliminer les idées de départ qui s'avéreront mauvaises et de prendre en compte les nouvelles idées qui arriveront en cours de route. Cette activité de construction du Product Backlog est collaborative, elle implique le Product Owner et l'équipe de développement.

Conclusion

À travers ce chapitre qui consiste à comprendre le contexte du système. Nous avons présenté l'organisme d'accueil « barsha Technology » en déterminant la problématique . Ensuite, nous avons tenté d'effectuer l'étude de l'existant, puis, nous avons présenté la solution de notre projet. Puis, Nous avons identifié les fonctionnalités et les acteurs les plus pertinents. Enfin, nous avons présenté la méthodologie utilisée pour la réalisation de ce projet. Dans le chapitre qui suit, nous passerons à la phase de l'étude préalable.

2

ETUDE PRÉALABLE

Sommaire

Introduction	34
2.1 Pilotage du projet avec Scrum	34
2.1.1 Équipe et rôle	34
2.1.2 Backlog produit	35
2.1.3 Diagramme de cas d'utilisation général	42
2.2 Planification des Sprints	43
2.3 Architecture de l'Application	44
2.3.1 Couche présentation	45
2.3.2 Couche Métier	45
2.3.3 Couche Données	47
2.4 Environnement de travail	48
2.4.1 Environnement matériel	48
2.4.2 Environnement logiciel	48
Draw.io :	48
IntelliJ IDEA :	49
Visual Studio Code :	49
Maven :	49
Node Package Manager :	49
Wampserver :	50
Postman :	50
Overleaf :	50

2.4.3	Technologie utilisées :	50
	Le Framework Angular :	51
	Le Framework Bootstrap :	51
	Le Framework Spring boot :	51
	Node JS :	52
	JSON Web Token(JWT) :	52
Conclusion	53

Introduction

L'étude préalable constitue une étape primordiale à la réalisation de notre application basé sur la méthode Scrum. Nous commencerons d'abord par la présentation du pilotage du projet avec scrum le backlog du produit et la planification des sprints afin de construire une bonne vision de la solution. Puis, nous définirons tous les besoins qui seront modélisés par le diagramme du cas d'utilisation général. Cette phase sera consacrée également pour la présentation de l'environnement de travail et les technologies utilisées pour la réalisation de notre application.

2.1 Pilotage du projet avec Scrum

Le cadre Scrum consiste en une équipe avec des rôles bien définis

2.1.1 Équipe et rôle

L'équipe Scrum est répartie en trois rôles :

- **Le Product Owner :**

c'est le responsable du produit, il représente les clients et les utilisateurs en transcrivant leurs besoins, définit et priorise les demandes produit.

- **Le Scrum Master :**

c'est le directeur de produit et n'est pas le chef de projet mais il a pour charge de faciliter l'application de Scrum, sa mission est de tout mettre en œuvre pour que l'équipe travaille dans de bonnes conditions et se concentre sur l'objectif du projet. Il porte également une attention particulière au respect des différentes phases de Scrum.

- **Scrum Team :**

L'équipe se gère en toute autonomie et a la charge du développement du produit. Il n'y a pas de notion de hiérarchie, toutes les décisions sont prises ensemble. Elle regroupe les rôles habituellement nécessaires à un projet (architecte, concepteur, développeur, etc.).

Dans le contexte de notre projet :

- **Le Product Owner** : Mme Zeineb Hadj Ali
- **Le Scrum Master** : Mr Mohamed Nidhal Jelassi
- **Le Scrum Team** : Mlle Nesrine Rabah

2.1.2 Backlog produit

Le backlog du produit c'est l'ensemble des fonctionnalités du produit. Il répond à tous les besoins du client. Les éléments du backlog sont classés par priorité ce qui permet de définir l'ordre de réalisation.

Le tableau 2.1.2 représente le backlog de produit de notre application :

id	Fonctionnalité	id stories	User stories	Sprint	Priorité
1	Inscription	1.1	En tant qu'un visiteur, je veux inscrire dans l'application .	1	1
2	Authentification	2.1	En tant qu'un utilisateur, je veux m'authentifier. Afin d'accéder à mon interface.	1	2
3	Gestion des Utilisateurs	3.1	En tant qu'un administrateur ou Scrum Master authentifié, je veux consulter la liste des utilisateurs.	1	4
		3.2	En tant qu'un administrateur ou Scrum Master authentifié, je veux ajouter un utilisateur.	1	3

3		3.3	En tant qu'un administrateur ou Scrum Master authentifié, je veux modifier un utilisateur(rôle,active ou désactive compte).	1	7
		3.4	En tant qu'un administrateur authentifié, je veux supprimer un utilisateur.	1	6
		3.5	En tant qu'un administrateur ou Scrum Master authentifié, je veux afficher le détail de chaque profil utilisateur.	1	8
		3.6	En tant qu'un administrateur authentifié, je veux modifier le mot de passe d'un utilisateur.	1	5
4	Gestion de Profil	4.1	En tant qu'un utilisateur authentifié, je veux consulter mon profile.	1	9
		4.2	En tant qu'un utilisateur authentifié, je veux modifier mon profile.	1	10
5	Gestion des Projets	5.1	En tant qu'un Scrum Master (ou bien Product Owner) authentifié,je veux consulter la liste des projets.	2	12

5	Gestion des Projets	5.2	En tant qu'un Scrum Master authentifié, je veux ajouter un projet.	2	11
		5.3	En tant qu'un Scrum Master authentifié, je veux modifier un projet.	2	15
		5.4	En tant qu'un Scrum Master authentifié, je veux supprimer un projet.	2	14
		3.5	En tant qu'un Scrum Master (ou bien Product Owner) authentifié, je veux afficher le détail de chaque projet.	2	13
6	Gestion des Sprints	6.1	En tant qu'un Scrum Master authentifié(ou bien chef d'équipe affecté et authentifié), je veux consulter la liste des sprints.	2	17
		6.2	En tant qu'un Scrum Master authentifié, je veux ajouter un sprint.	2	16
		6.3	En tant qu'un Scrum Master authentifié, je veux supprimer un sprint.	2	18
		6.4	En tant qu'un Scrum Master authentifié,je veux modifier un sprint(l'état,date...).	2	19

6		6.5	En tant qu'un Scrum Master authentifié(ou bien chef d'équipe affecté et authentifié), je veux afficher le détail de chaque sprint.	2	20
		6.6	En tant qu'un chef d'équipe affecté et authentifié , je veux modifier l'état de sprint.	2	21
7	Gestion des Tâches	7.1	En tant qu'un chef d'équipe authentifié(ou bien member d'équipe affecté et authentifié), je veux consulter la liste des tâches.	2	23
		7.2	En tant qu'un chef d'équipe authentifié, je veux ajouter une tâche.	2	22
		7.3	En tant qu'un chef d'équipe authentifié, je veux modifier une tâche.	2	24
		7.4	En tant qu'un chef d'équipe authentifié, je veux supprimer une tâche.	2	25
		7.5	En tant qu'un chef d'équipe authentifié(ou bien member d'équipe affecté et authentifié), je veux afficher le détail de chaque Tâche.	2	26

7		7.6	En tant qu'un member d'équipe affecté et authentifié , je veux modifier l'état de tâche.	2	27
8	Gestion des Commentaires	8.1	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux consulter la liste des commentaires.	3	29
		8.2	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux ajouter un commentaire.	3	28
		8.3	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux supprimer un commentaire.	3	30
9	Gestion des Attachements	9.1	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux consulter la liste des attachements.	3	32
		9.2	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux ajouter un attachement.	3	31

9		9.3	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux supprimer un attachement.	3	33
		9.4	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux télécharger un attachement.	3	34
10	Gestion des documents	10.6	En tant qu'un utilisateur authentifié, je veux télécharger un document.	4	40
		10.7	En tant qu'un utilisateur authentifié, je veux afficher le détail de chaque document.	4	43
		10.8	En tant qu'un utilisateur authentifié, je veux consulter la liste des documents archivés.	4	41
		10.9	En tant qu'un utilisateur authentifié, je veux restaurer un document archivé.	4	42
11	Gestion des Evènements	11.1	En tant qu'un utilisateur authentifié, je veux consulter la liste des évènements.	4	45

11	11.2	En tant qu'un utilisateur authentifié, je veux ajouter un évènement .	4	44
	11.3	En tant qu'un utilisateur authentifié, je veux modifier un évènement .	4	46
	11.4	En tant qu'un utilisateur authentifié, je veux supprimer un évènement (archivé ou non archiv��).	4	45
	11.5	En tant qu'un utilisateur authentifié, je veux archiver un évènement .	4	47
	11.6	En tant qu'un utilisateur authentifié, je veux afficher le d��tail de chaque ��v��nement .	4	48
	11.7	En tant qu'un utilisateur authentifié, je veux consulter la liste des ��v��nements archiv��s.	4	49
	11.8	En tant qu'un utilisateur authentifié, je veux restaurer un ��v��nement archiv��.	4	50

TABLE 2.1 – Backlog Produit

2.1.3 Diagramme de cas d'utilisation général

Dans cette partie, nous présentons le diagramme des cas d'utilisations global de notre application dans la figure 2.1.

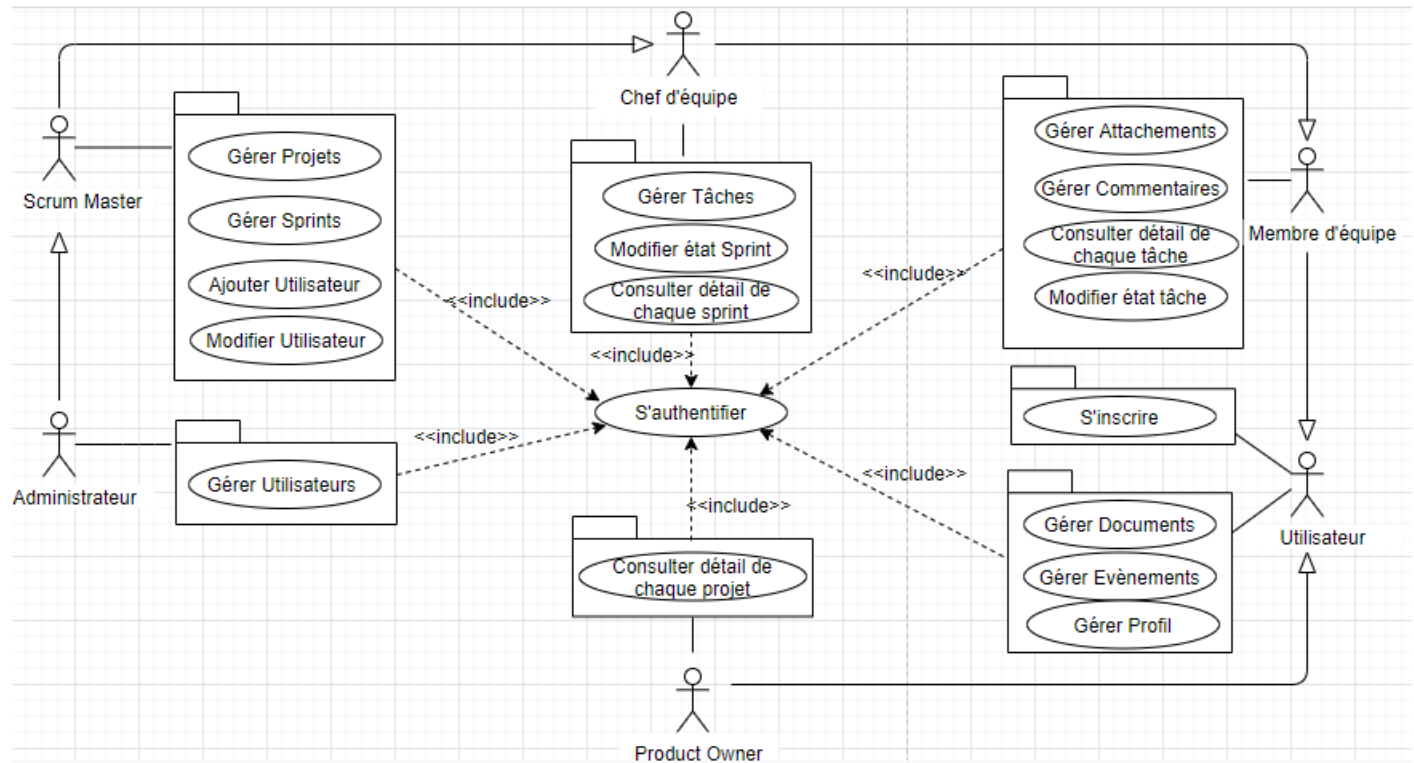


FIGURE 2.1 – Diagramme de cas d'utilisation général

2.2 Planification des Sprints

Dans cette partie nous allons établir la planification des sprints de notre projet. Le tableau 2.2 présente les sprints classés par leurs priorités :

Sprint 1	Sprint 2	Sprint 3	Sprint 4
<ul style="list-style-type: none"> • Inscription • Authentification • Gestion des Utilisateurs • Gestion de Profil 	<ul style="list-style-type: none"> • Gestion des Projets • Gestion des Sprints • Gestion des Tâches 	<ul style="list-style-type: none"> • Gestion des Commentaires • Gestion des Attachements 	<ul style="list-style-type: none"> • Gestion des Documents • Gestion des Événements
04/03/2021-13/05/2021	14/05/2021-03/07/2021	04/07/2021-25/08/2021	26/08/2021-10/09/2021

TABLE 2.2 – Planification des Sprints

2.3 Architecture de l'Application

L'architecture est l'ensemble des aspects techniques et applicatifs qui sont importants pour un logiciel. Les choix architecturaux influent sur la réussite ou l'échec d'un projet.

En effet, nous proposons l'architecture trois-tiers pour la mise en place de notre application. Cette architecture est un modèle logique d'architecture applicative qui vise à modéliser une application comme un empilement de trois couches logicielles telle que présentée dans la figure 2.2.

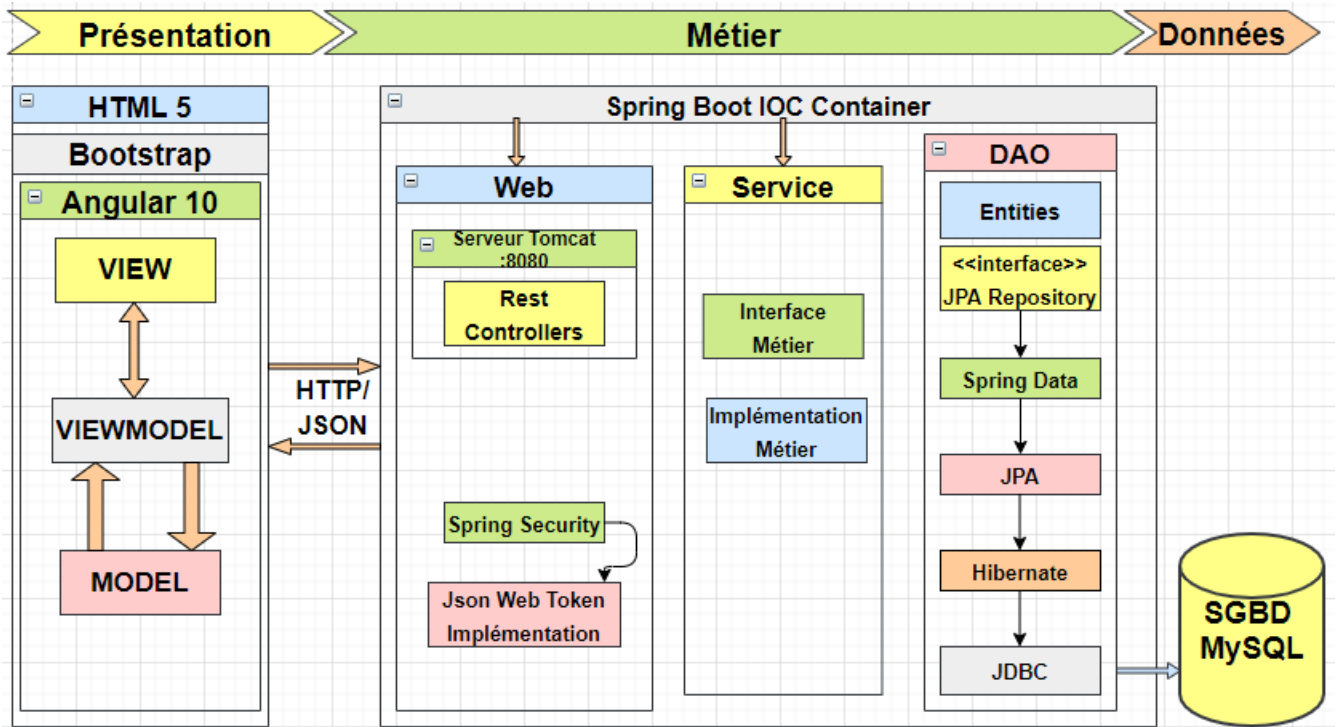


FIGURE 2.2 – Architecture de l'Application

2.3.1 Couche présentation

Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'interface homme-machine. Cette couche est liée au type de client utilisé. nous choisissons pour notre application le patron de conception MVVM (MODEL VIEW VIEWMODEL).

- **Modèle (Model)** : le modèle contient les données. Généralement, ces données proviennent d'une base de données ou d'un service externe comme un API.
- **Vue (View)** : la vue correspond à ce qui est affiché (la page web dans notre cas). La vue contient les différents composants graphiques (boutons, liens, listes) ainsi que le texte.
- **Vue-Modèle (ViewModel)** : ce composant fait le lien entre le modèle et la vue. Il s'occupe de gérer les liaisons de données et les éventuelles conversions. C'est ici qu'intervient le binding.

2.3.2 Couche Métier

Couche métier ou business correspond à la partie fonctionnelle de l'application, celle qui implémente la logique, et qui décrit les opérations que l'application opère sur les données en fonction des requêtes effectuées par les utilisateurs à travers de la couche présentation.

nous pouvons détailler la couche métier en trois autres couches comme la suite :

- **Couche Web** permet d'implémenter la logique présentation web de l'application. Elle possède un serveur d'application Tomcat embarqué afin de faciliter le déploiement de l'application . De plus tous les contrôleurs que nous implémentons et tous les modèles de vues font partie de cette couche .

Dans cette partie on utilise des Framework de sécurité léger comme Spring Security.

- Cette couche reçoit les requêtes Http de client web .
 - Faire appel à la couche Service pour effectuer les traitement .
 - puis envoyer le résultat au client(une réponse Http).
- **Couche Service ou Couche métier** permet d'implémenter la logique métier de l'application.Cette couche utilise généralement une approche orientée objet.
 - **Couche DAO** Data Access Object est un patron de conception. C'est une couche technique qui représente la couche d'accès aux données de l'application et puisque les données sont stockées dans une base de données relationnelle, cette couche utilise un Framework de mapping objet relationnel implémentant la spécification JPA comme Hibernate .

Le tableau 2.3 présente une description textuelle de quelques Mots-clés présenter dans la figure 2.2 :

Mot-clé	Description
Spring Boot IOC Container	Représente le Framework qui permet de faire l'inversion de contrôle.
JPA Repository	Il contient une API pour les opérations CRUD de base et également une API pour la pagination et le tri.
JDBC Java DataBase Connectivity	désigne une API pour permettre un accès aux bases de données avec Java.
Rest Controllers	Le contrôleur intercepte toutes les requêtes et les transmet (grâce à Spring RestTemplate) ensuite au service REST qui lui renvoie une réponse,et c'est cette réponse qui est remise au Model.
Hibernate	Hibernate est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application.

Hibernate	Hibernate permet donc de représenter une base de données en objets Java et vice versa.
Spring Security	est un module incontournable d'une application développée en Spring. Il apporte tout le nécessaire pour sécuriser une application et il a l'avantage d'être vraiment personnalisable. La notion de sécurité informatique n'est pas une mince affaire et sa mise en place est parfois longue et demande d'être constamment adaptée au niveau réseau, serveurs, application... Spring Security n'intervient que sur le domaine applicatif.
Spring Data	C'est un projet supplémentaire de Spring créé il y a quelques années pour répondre aux besoins d'écrire plus simplement l'accès aux données et d'avoir une couche d'abstraction commune à de multiples sources de données. Spring Data s'interface avec plusieurs sources de données parmi lesquelles JPA,

TABLE 2.3 – Tableau des mots-clés

2.3.3 Couche Données

Elle consiste à gérer l'accès aux données du système. La configuration pour les Entités se fait à la fois dans le serveur de l'applications et dans une application autonome réside dans un archive de persistance.

2.4 Environnement de travail

L'implémentation consiste la phase d'achèvement et d'aboutissement du projet. Pour accomplir cette tâche, il faut utiliser des outils adéquats à la réalisation du projet. Ces outils représentent l'environnement de travail.

2.4.1 Environnement matériel

Dans cette partie nous présentons les outils matériels que nous avons utilisés pour réaliser notre projet. Pour la réalisation de notre application nous avons utilisé :

- PC portable : DELL
- Processeur : Intel(R) Core(TM) i5-3340M CPU @ 2.70 GHz 2.70 GHz
- Mémoire installée : 10,0 GO
- Système d'exploitation : Windows 10 Professionnel 64 bits

2.4.2 Environnement logiciel

Dans cette partie nous présentons les différents outils logiciels utilisés pour notre solution.

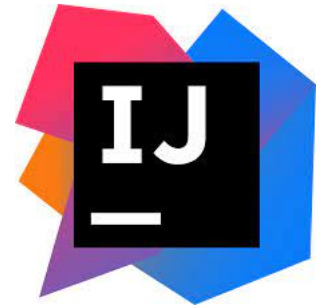
Draw.io :

draw.io est l'un des outils de création de diagrammes les plus riches en fonctionnalités du marché et un concurrent direct de solutions professionnelles de bureau comme Visio. [5]



IntelliJ IDEA :

IntelliJ IDEA est un environnement de développement intégré (IDE) écrit en Java pour le développement de logiciels informatiques. Il est développé par JetBrains (anciennement connu sous le nom d'IntelliJ), et est disponible en tant qu'édition communautaire sous licence Apache 2.

**Visual Studio Code :**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et OS X4. Présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code cross-Platform, open source et gratuit, supportant une dizaine de langages comme Type Script.

**Maven :**

Maven est un outil de construction de projets open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Il permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java. [6]

**Node Package Manager :**

npm est le gestionnaire de paquets officiel de Node.js. Depuis la version 0.6.3 de Node.js, npm fait partie de l'environnement et est donc automatiquement installé par défaut. npm fonctionne avec un terminal et gère les dépendances pour une application. [7]



Wampserver :

WampServer est une plateforme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL¹. [8]

**Postman :**

Est un client REST proposé par Google. Il est disponible sous la forme d'une extension Chrome ou bien d'une application stand-alone. Parmi les nombreuses solutions pour interroger ou tester web services et API, Postman propose de nombreuses fonctionnalités, une prise en main rapide et une interface graphique agréable.

**Overleaf :**

overleaf est une plateforme en ligne gratuite permettant d'éditer du texte en LATEX en temps réel et compilateur PDF. En outre, elle offre la possibilité de rédiger des documents de manière collaborative, de proposer ses documents directement à différents éditeurs (IEEE Journal, Springer, etc.) ou plateformes d'archives ouvertes (arXiv, engrxiv, etc.) pour une éventuelle publication. [9]

**2.4.3 Technologie utilisées :**

Dans cette section, nous allons présenter les technologies et frameworks utilisées pour la réalisation de l'application

1. MySQL : est un système de gestion de bases de données relationnelles.

Le Framework Angular :

Angular est un Framework JavaScript côté client qui permet de réaliser des applications de type «Single Page Application». Le code source d'Angular est écrit en Type Script.



Le Framework Bootstrap :

Est un Framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce Framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement.

Ce Framework est pensé pour développer des sites avec un design responsif, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore.



Le Framework Spring boot :

Spring Boot est un framework de développement JAVA. C'est une déclinaison du framework classique de Spring qui permet essentiellement de réaliser des micro services (ce sont la majeure partie du temps des services web qui sont regroupés en API).

Pour faire simple, Spring Boot est un framework de développement JAVA permettant la création d'API web. [10]



Node JS :

Node JS est une plateforme de développement Open Source orientée client-serveur, elle permet de développer des applications web en Javascript et inclut pour cela le moteur d'exécution de Google V8 qui a considérablement permis d'accélérer l'exécution de code JavaScript.



La plateforme Node JS propose des nombreux modules pour composer son architectures logicielle ces modules sont utilisable grâce à «npm», la gestionnaire de paquets de Node JS.

Node.js contient une bibliothèque de serveur HTTP intégrée, ce qui rend possible de faire tourner un serveur web sans avoir besoin d'un logiciel externe comme Apache ou Lighttpd , et permettant de mieux contrôler la façon dont le serveur web fonctionne. Un programme NodeJS est compilé avant d'être exécuté, ce qui permet d'excellentes performance.

JSON Web Token(JWT) :

est un standard ouvert défini dans la RFC 7519. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité et de l'authenticité des données. Elle s'effectue par l'algorithme HMAC ou RSA.



Conclusion

Dans ce chapitre nous avons présenté le backlog du produit suivi de diagramme de cas d'utilisation général. Nous avons planifié par la suite la réalisation des différents sprints. Nous avons aussi présenté l'environnement de travail et les technologies utilisées pour la réalisation de notre application. Le chapitre suivant sera consacrer pour la réalisation du premier sprint .

3

Sprint 1

Sommaire

Introduction	56
3.1 Backlog du sprint 1	56
3.2 Spécifications fonctionnelles	58
3.2.1 Raffinement de cas d'utilisation « S'inscrire »	58
3.2.2 Raffinement de cas d'utilisation « S'authentifier »	59
3.2.3 Raffinement de cas d'utilisation « Gérer Utilisateurs »	61
3.2.4 Raffinement de cas d'utilisation « Modifier Profil »	64
3.3 Conception	66
3.3.1 Diagramme de classes du premier sprint	66
3.3.2 Diagramme de séquence du cas d'utilisation « S'authentifier »	67
3.3.3 Diagramme de séquence du cas d'utilisation « Modifier utili- sateur »	68
3.3.4 Diagramme de séquence du cas d'utilisation « Supprimer uti- lisateur »	70
3.4 Réalisation	71
3.4.1 Interface d'inscription	71
3.4.2 Interface d'authentification	72
3.4.3 Interface de détail utilisateur	73
3.4.4 Interface de modification utilisateur	74
3.4.5 Interface de profil utilisateur	75
3.4.6 Interface de modification mot de passe	76

Conclusion	77
----------------------	----

Introduction

Après avoir détaillé les exigences de notre projet sous forme d'un Backlog de produit, dans ce chapitre nous allons présenter le premier sprint qui évoque les quatre phases : sprint backlog spécifications fonctionnelles, conception et réalisation.

3.1 Backlog du sprint 1

Le tableau 3.1 présente le backlog du premier sprint :

id	Fonctionnalité	id stories	User stories	Sprint	Priorité
1	Inscription	1.1	En tant qu'un visiteur, je veux inscrire dans l'application .	1	1
2	Authentification	2.1	En tant qu'un utilisateur, je veux m'authentifier. Afin d'accéder à mon interface.	1	2
3	Gestion des Utilisateurs	3.1	En tant qu'un administrateur ou Scrum Master authentifié, je veux consulter la liste des utilisateurs.	1	4
		3.2	En tant qu'un administrateur ou Scrum Master authentifié, je veux ajouter un utilisateur.	1	3
		3.3	En tant qu'un administrateur ou Scrum Master authentifié, je veux modifier un utilisateur(rôle,active ou désactive compte).	1	7

3		3.4	En tant qu'un administrateur authentifié, je veux supprimer un utilisateur.	1	6
		3.5	En tant qu'un administrateur ou Scrum Master authentifié, je veux afficher le détail de chaque profil utilisateur.	1	8
		3.6	En tant qu'un administrateur authentifié, je veux modifier le mot de passe d'un utilisateur.	1	5
4	Gestion de Profil	4.1	En tant qu'un utilisateur authentifié, je veux consulter mon profile.	1	9
		4.2	En tant qu'un utilisateur authentifié, je veux modifier mon profile.	1	10

TABLE 3.1 – Premier sprint Backlog

3.2 Spécifications fonctionnelles

Dans cette section nous présentons le raffinement du diagramme de cas d'utilisation pour chacune des fonctionnalités retenues dans ce sprint.

3.2.1 Raffinement de cas d'utilisation « S'inscrire »

La figure 3.1 illustre le diagramme de cas d'utilisation s'inscrire :

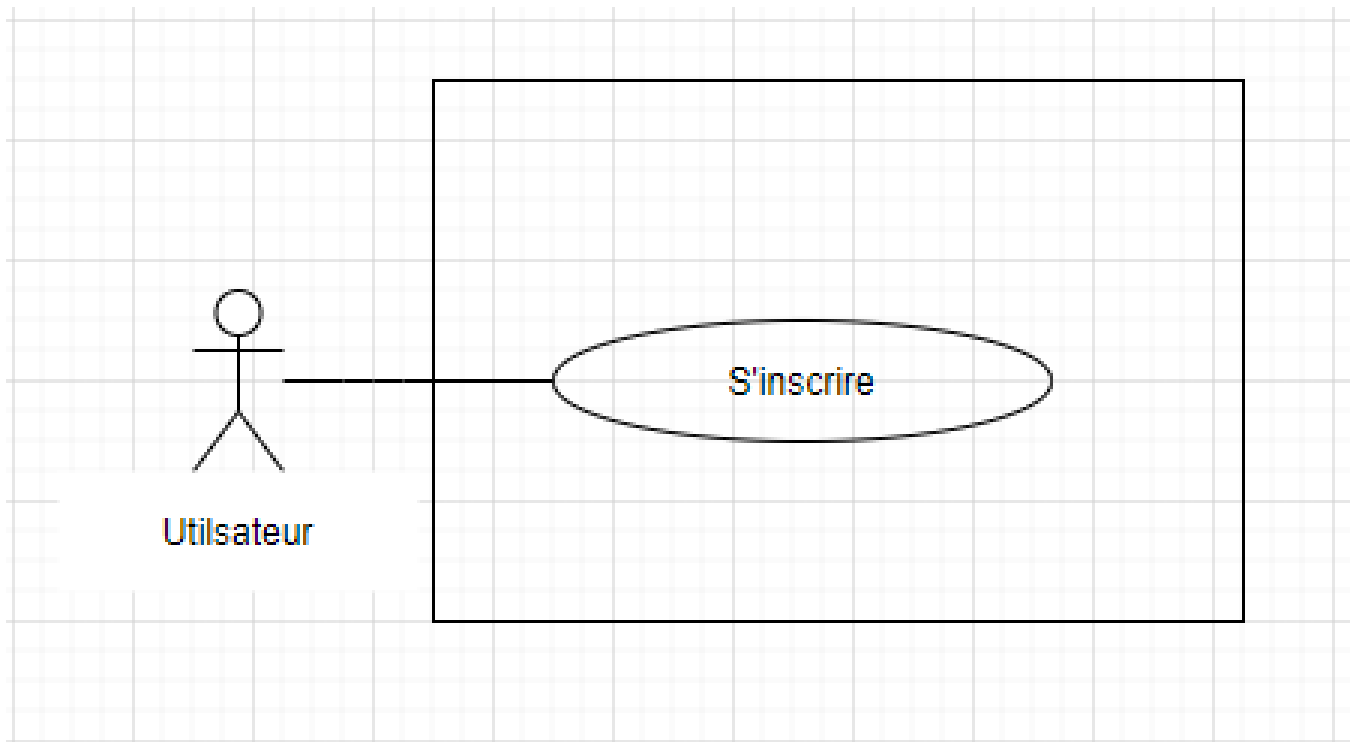


FIGURE 3.1 – Diagramme du cas d'utilisation « S'inscrire »

Le tableau 3.2 présente une description textuelle de cas d'utilisation « S'inscrire »

Titre	S'inscrire
Objectif	Les visiteurs peuvent s'inscrire dans l'application.
Acteur	Utilisateur
Pré condition	L'utilisateur n'est pas encore inscrit.
Post condition	L'utilisateur va créer un compte.
Description du scénario	Le visiteur envoie les différents détails personnels utilisant une formulaire en attendant la réception du mot de passe à travers l'email.
Exception	-Tous les champs sont contrôlés et obligatoires ce pour cela il doit vérifier qu'il a bien rempli le formulaire en respectant le type de chaque champ. -Si le nom ou l'email de l'utilisateur existe le système affiche un message d'erreur.

TABLE 3.2 – Description textuelle du cas d'utilisation « S'inscrire »

3.2.2 Raffinement de cas d'utilisation « S'authentifier »

La figure 3.2 illustre le diagramme de cas d'utilisation s'authentifier :

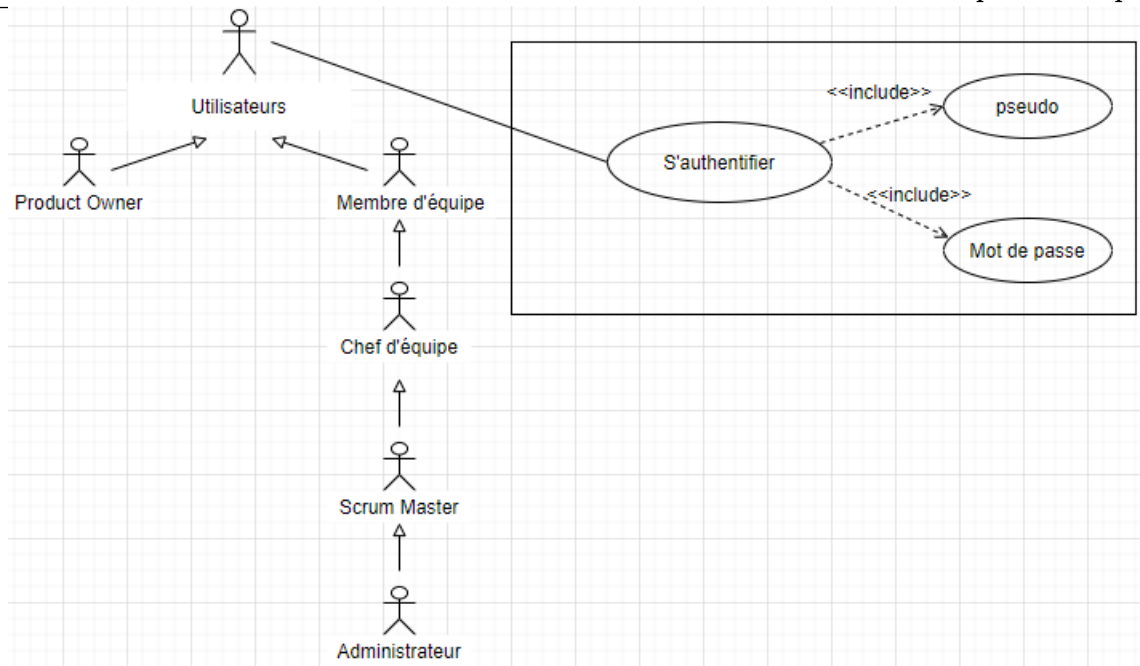


FIGURE 3.2 – Diagramme du cas d'utilisation « S'authentifier »

Le tableau 3.3 présente une description textuelle de cas d'utilisation «S'authentifier »

Titre	S'authentifier
Objectif	L'utilisateur doit s'authentifier pour accéder à sa session.
Acteur	Utilisateur de l'application
Pré condition	L'utilisateur n'a pas encore authentifié.
Post condition	L'utilisateur est identifié selon son profil (administrateur, Scrum Master, chef d'équipe,membre d'équipe) .
Description du scénario	-Le système affiche le formulaire d'authentification.

	<ul style="list-style-type: none"> - L'utilisateur saisit son pseudo et son mot de passe. -Le pseudo et le mot de passe sont valides, l'utilisateur est connecté au système et il peut par la suite accéder aux différentes fonctionnalités de l'application .
Exception	Si un des champs est invalide ou le compte d'utilisateur est désactivé, le système affiche un message d'erreur.

TABLE 3.3 – Description textuelle du cas d'utilisation « S'authentifier »

3.2.3 Raffinement de cas d'utilisation « Gérer Utilisateurs »

La figure 3.3 illustre le diagramme de cas d'utilisation gérer utilisateurs :

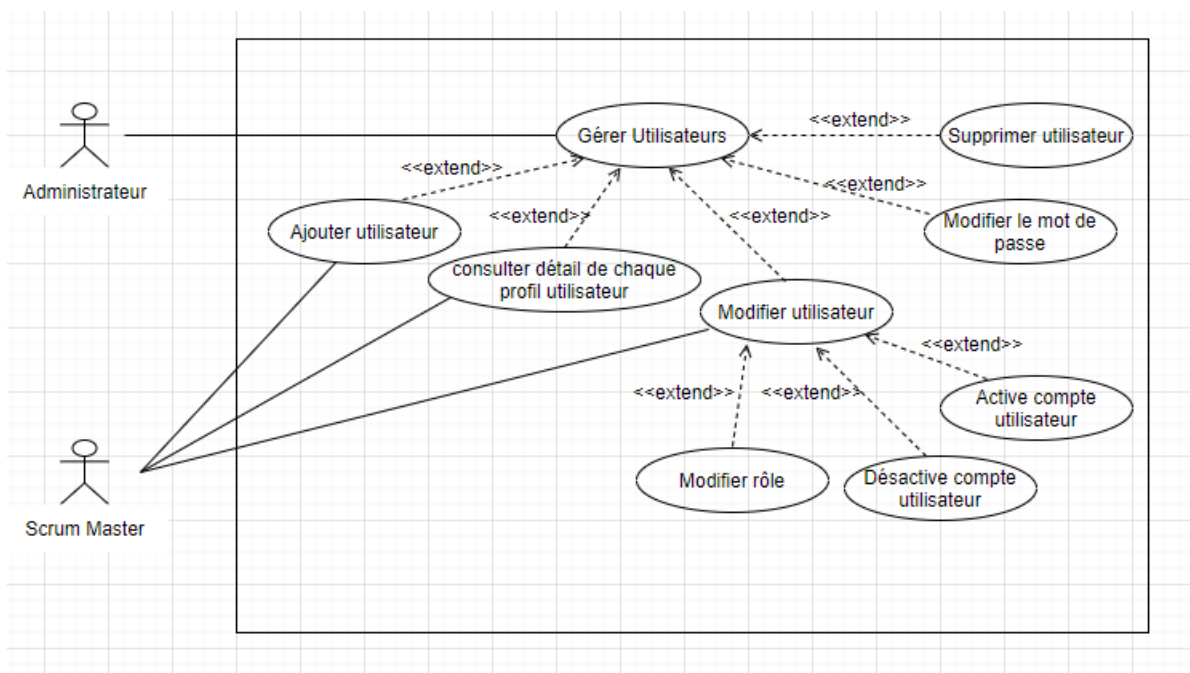


FIGURE 3.3 – Diagramme du cas d'utilisation «Gérer Utilisateurs »

Le tableau 3.4 présente une description textuelle de cas d'utilisation « Désactiver compte utilisateur »

Titre	Désactiver compte utilisateur
Objectif	L'acteur peut désactiver un compte utilisateur.
Acteur	Administrateur/ Scrum Master
Pré condition	-Acteur authentifié. -L'acteur a consulté la liste des utilisateurs.
Post condition	Un compte est désactivé.
Description du scénario	-L'acteur clique sur « modifier utilisateur » puis cocher la case « Active ». -Le système désactive le compte et affiche un message de succès.
Exception	Tous les champs sont contrôlés et obligatoires ce pour cela il doit vérifier qu'il a bien rempli le formulaire de modification.

TABLE 3.4 – Description textuelle du cas d'utilisation « Désactiver compte utilisateur »

Le tableau 3.5 présente une description textuelle de cas d'utilisation « Supprimer utilisateur »

Titre	Supprimer utilisateur
Objectif	L'administrateur peut supprimer un utilisateur.
Acteur	Administrateur
Pré condition	-Administrateur authentifié -L'administrateur a consulté la liste des utilisateurs
Post condition	Un utilisateur est supprimé.
Description du scénario	-L'administrateur clique sur «Supprimer utilisateur» -Le système supprime l'utilisateur et affiche un message de succès.
Exception	-Si l'utilisateur n'a pas la permission de suppression,le système affiche un message d'erreur.

TABLE 3.5 – Description textuelle du cas d'utilisation «Supprimer utilisateur»

3.2.4 Raffinement de cas d'utilisation «Modifier Profil»

La figure 3.4 illustre le diagramme de cas d'utilisation «Modifier Profil » :

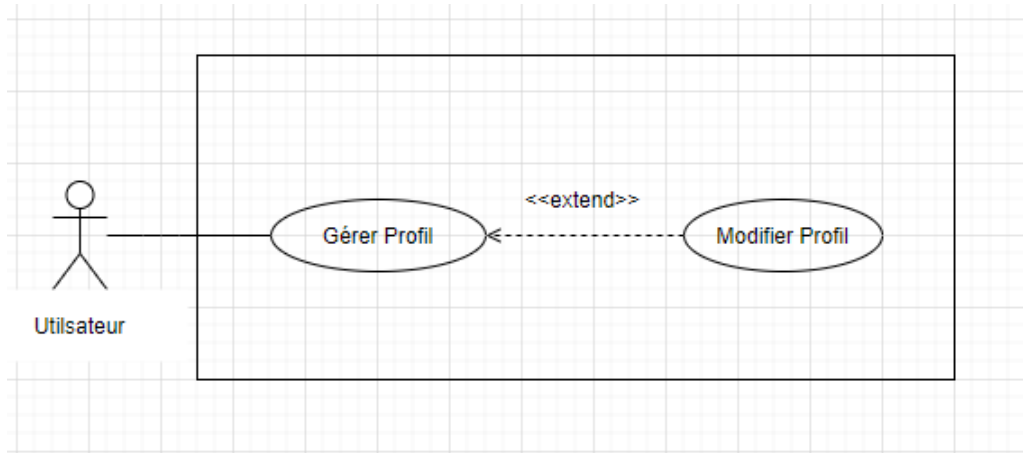


FIGURE 3.4 – Diagramme du cas d'utilisation «Modifier Profil »

Le tableau 3.6 présente une description textuelle de cas d'utilisation «Modifier Profil »

Titre	Modifier profil
Objectif	Les utilisateurs peuvent modifier les informations de leurs profils.
Acteur	Utilisateur
Pré condition	-Utilisateur authentifié -l'utilisateur a consulté son profil.
Post condition	Profil modifié.
Description du scénario	- Le système affiche aux utilisateurs un formulaire qui contient toutes les informations concernant son profil. -L'utilisateur fait les modifications souhaitées puis cliquer sur «Sauvegarder les modifications» -Le système enregistre les modifications .
Exception	-Tous les champs sont obligatoires ce pour cela il doit vérifier qu'il a bien rempli le formulaire .

TABLE 3.6 – Description textuelle du cas d'utilisation«Modifier Profil»

3.3 Conception

Cette partie définit les éléments résultant de l'analyse des spécifications du premier sprint. Afin d'en faciliter la réalisation, cette phase de conception sera définie comme la description du sprint à développer.

3.3.1 Diagramme de classes du premier sprint

La figure 3.5 illustre le diagramme de classe de premier sprint :

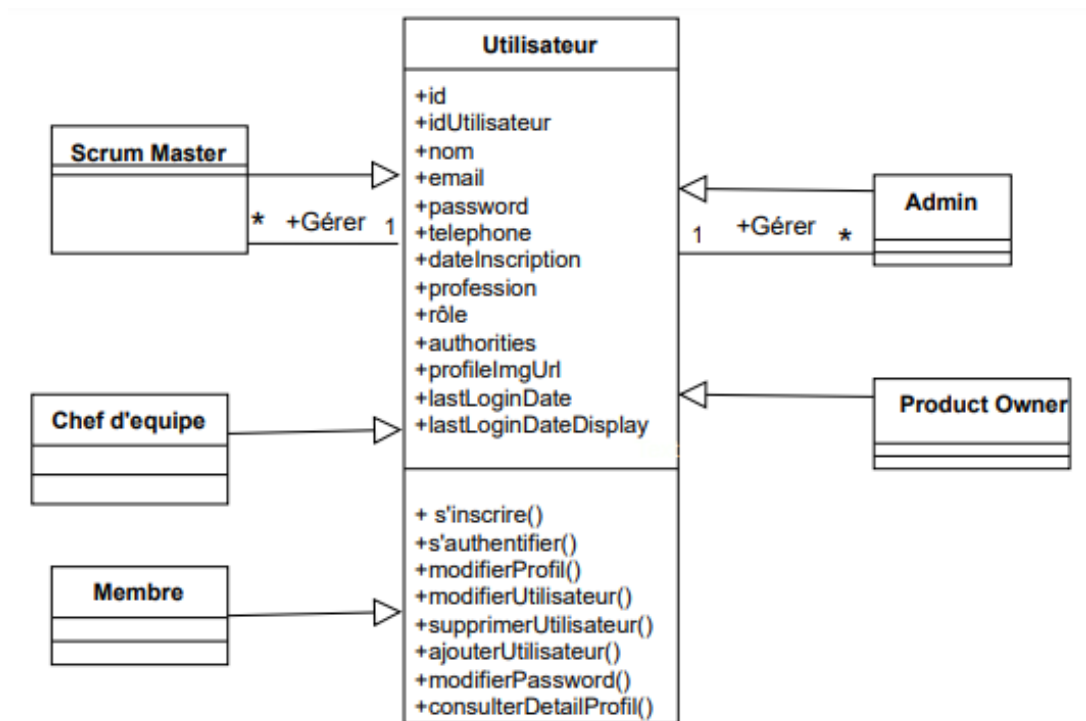


FIGURE 3.5 – Diagramme de classes « Premier Sprint »

3.3.2 Diagramme de séquence du cas d'utilisation « S'authentifier »

Pour accéder à l'application, l'utilisateur doit tout d'abord s'identifier par son login et son mot de passe via le système qui prend en charge de vérifier l'existence des données dans la base de données. S'il est accepté, donc il aura accès à son espace de travail. Sinon, le système affiche un message d'erreur. La figure 3.6 illustre le diagramme de séquence du cas d'utilisation « S'authentifier »

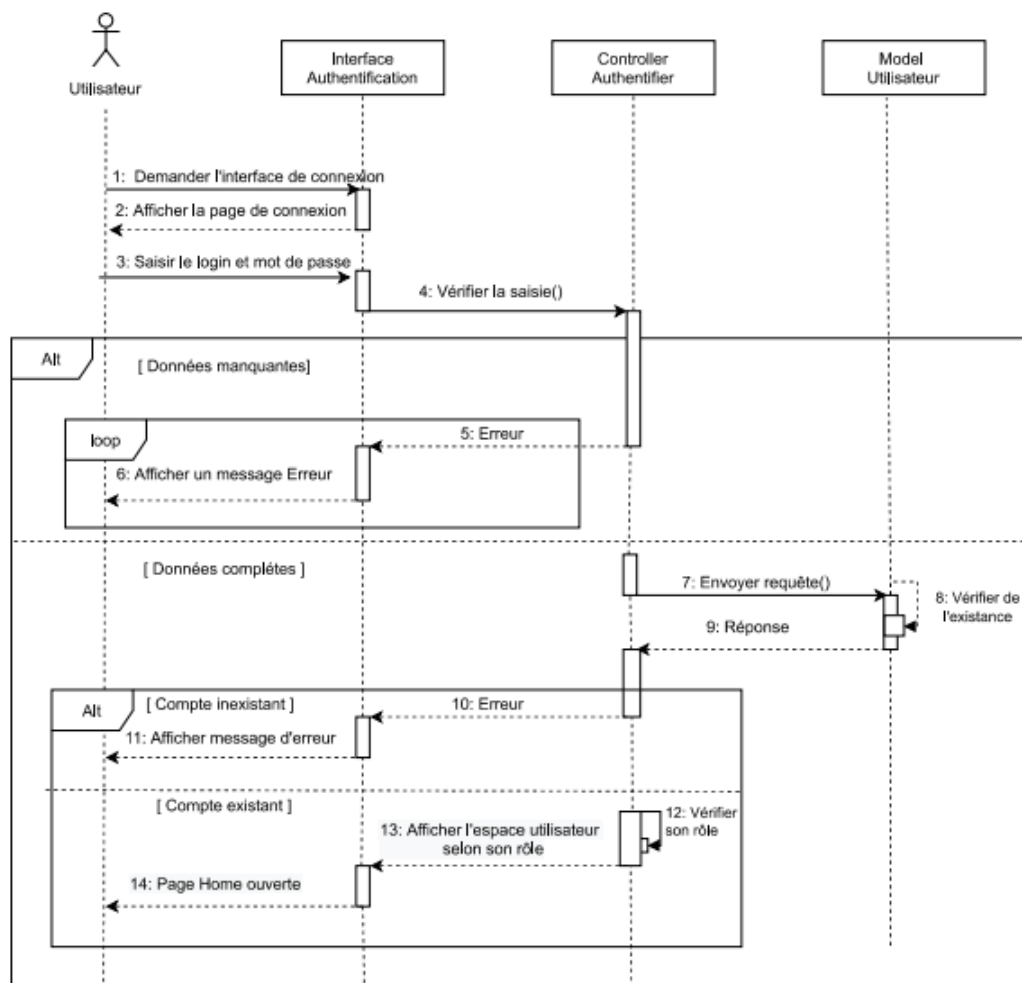


FIGURE 3.6 – Diagramme de séquence du cas d'utilisation « S'authentifier »

3.3.3 Diagramme de séquence du cas d'utilisation « Modifier utilisateur »

Pour modifier un utilisateur, l'acteur du système accède à l'interface modifier utilisateur, et remplit les champs puis clique sur le bouton «Sauvegarder les modifications», ainsi le système vérifie les données saisies et enregistre les modifications.

La figure 3.7 illustre le diagramme de séquence du cas d'utilisation « Modifier utilisateur » :

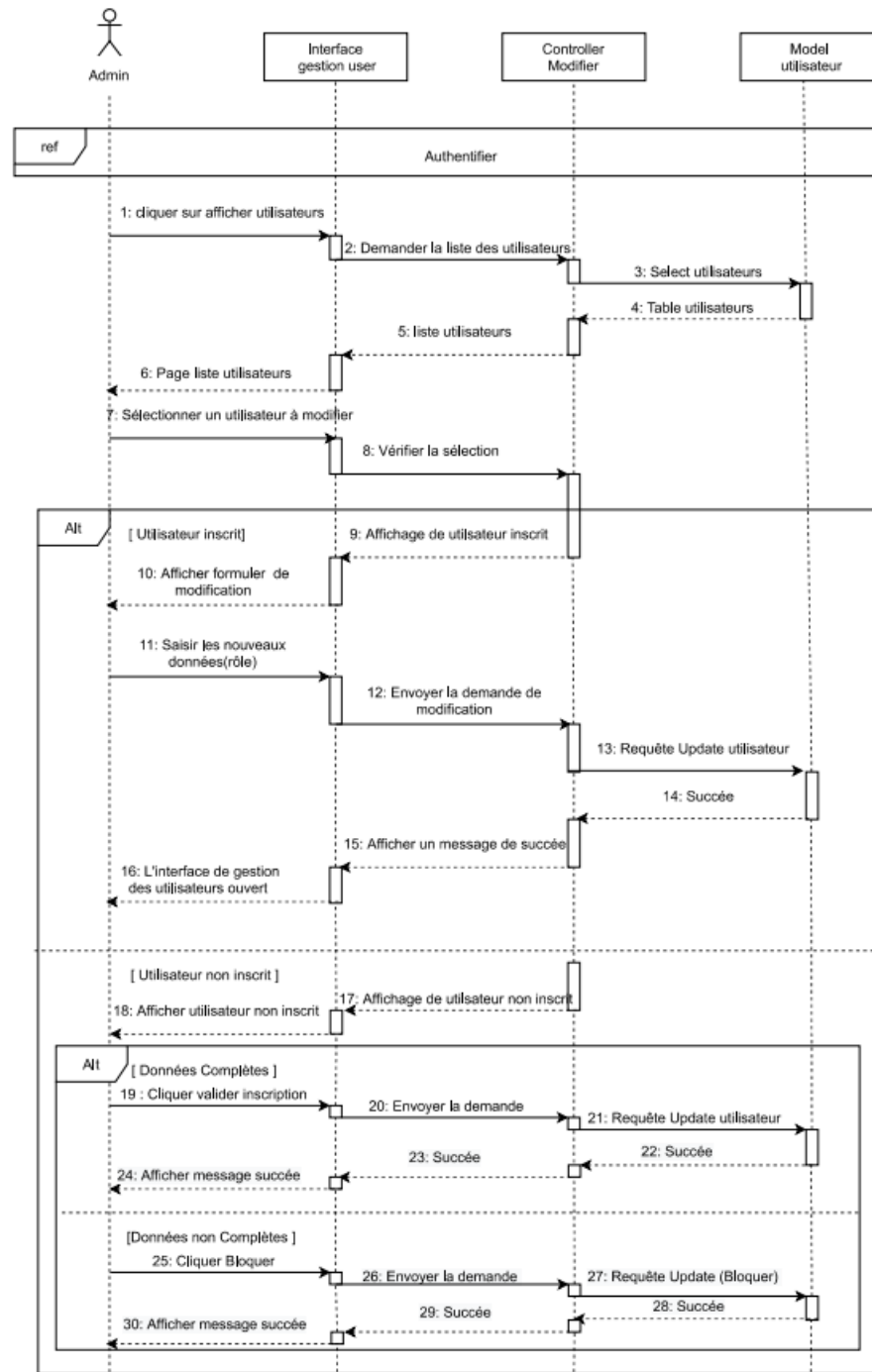


FIGURE 3.7 – Diagramme de séquence du cas d'utilisation « Modifier utilisateur »

3.3.4 Diagramme de séquence du cas d'utilisation « Supprimer utilisateur »

Afin d'établir la suppression d'un utilisateur, l'administrateur du système accède à l'interface gestion des utilisateurs où se trouve la liste des utilisateurs, sélectionne l'utilisateur approprié puis clique sur l'action "supprimer", ceci étant fait le système supprime l'utilisateur.

La figure 3.8 illustre le diagramme de séquence du cas d'utilisation « Supprimer utilisateur » :

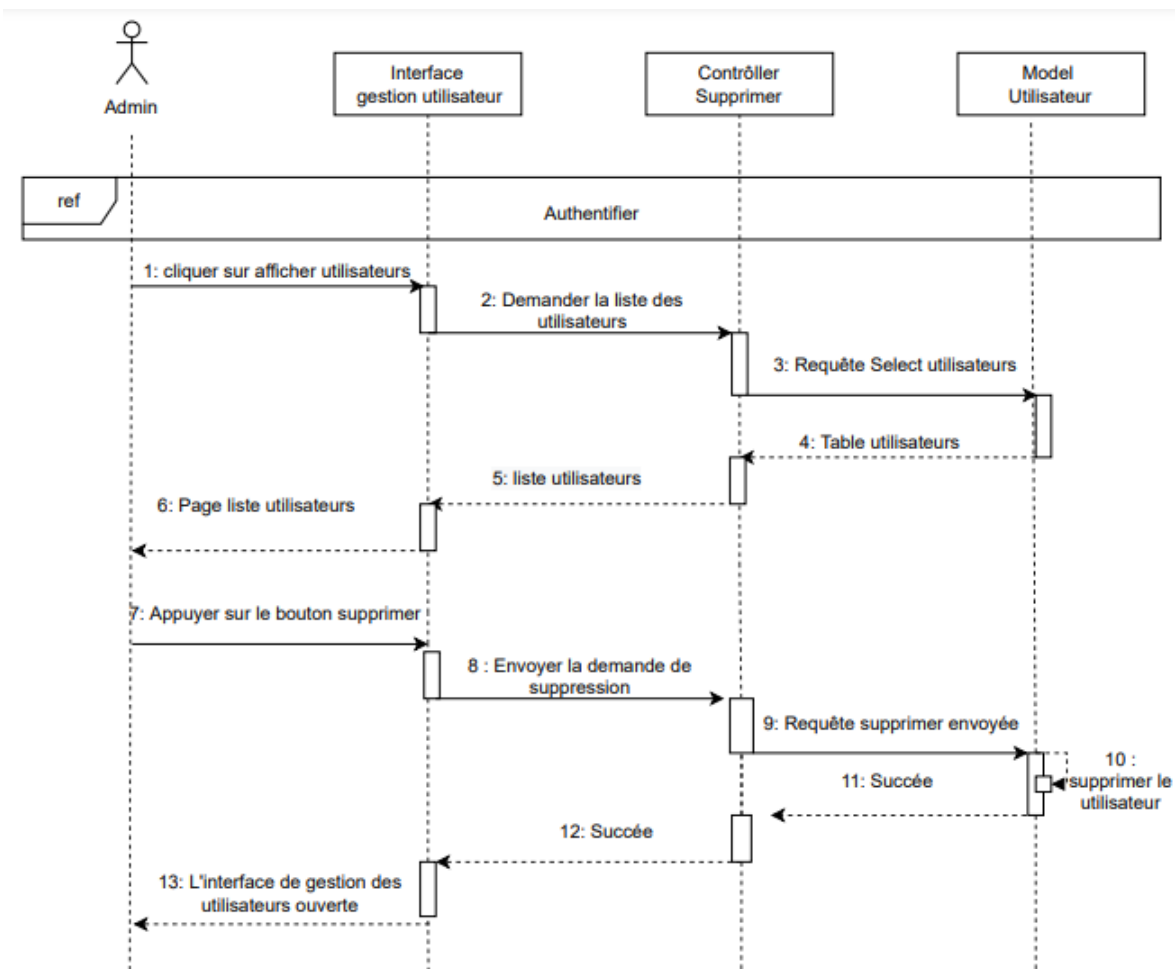


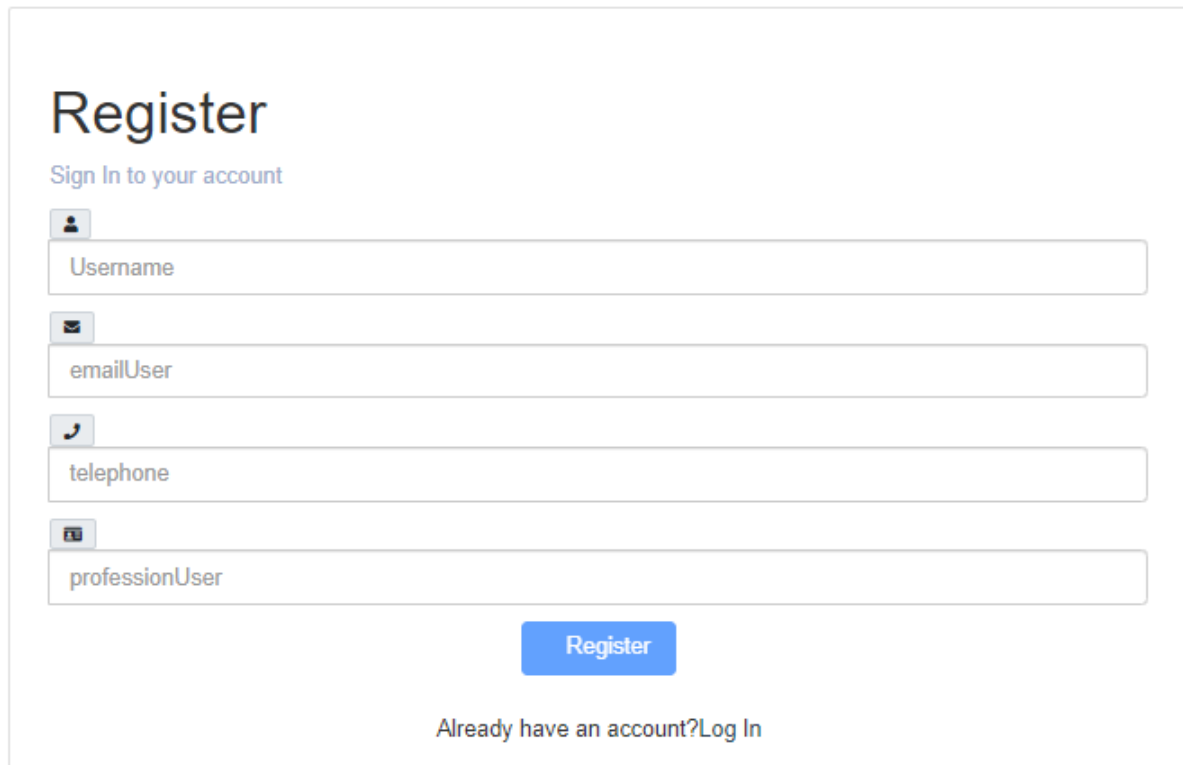
FIGURE 3.8 – Diagramme de séquence du cas d'utilisation « Supprimer utilisateur »

3.4 Réalisation

Passons maintenant à la présentation de la phase de réalisation du premier sprint. Nous exposons quelques interfaces réalisées durant ce sprint. Chaque interface illustre une fonctionnalité donnée.

3.4.1 Interface d'inscription

Pour inscrire à l'application, le visiteur doit remplir le formulaire suivant. La figure 3.9 illustre l'interface d'inscription :

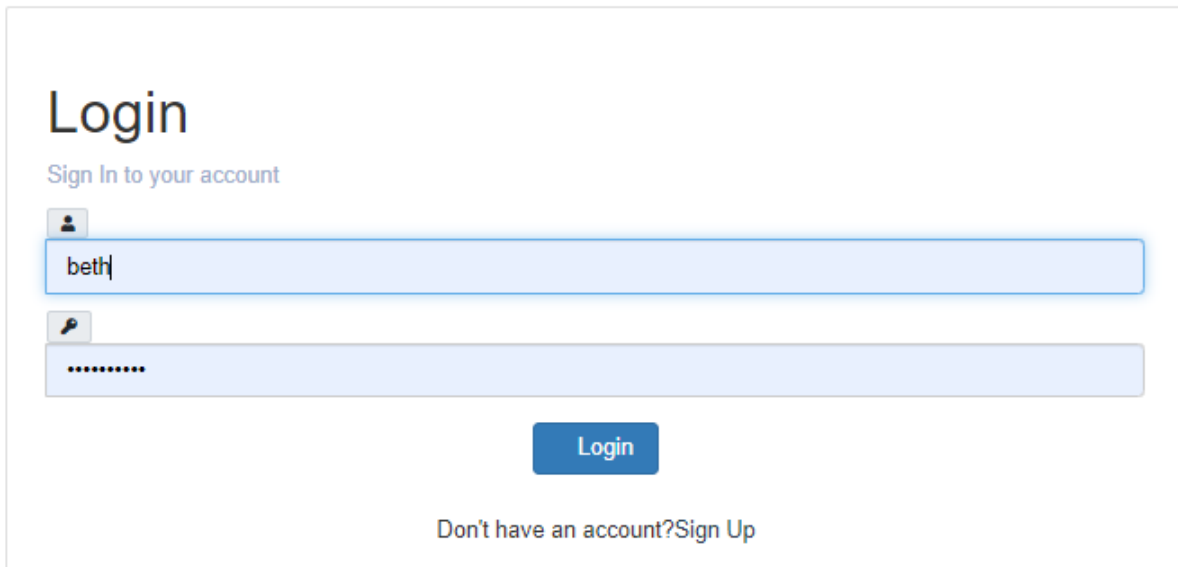


The image shows a web form titled "Register". Below the title is a link "Sign In to your account". The form contains four input fields, each with a small icon to its left: a person icon for "Username", an envelope icon for "emailUser", a telephone handset icon for "telephone", and a document icon for "professionUser". Below these fields is a blue "Register" button. At the bottom of the form is a link "Already have an account? Log In".

FIGURE 3.9 – Interface d'inscription

3.4.2 Interface d'authentification

A fin d'accéder à l'application, l'utilisateur de l'application doit s'authentifier en saisissant son login et son mot de passe. S'il est accepté, donc il aura accès à son espace de travail. Sinon, le système affiche un message d'erreur. La figure 3.10 illustre l'interface d'authentification :



The image shows a login interface with the following elements:

- Title:** "Login" in a large, bold, dark font.
- Subtitle:** "Sign In to your account" in a smaller, blue font.
- Username Field:** A light blue input box with a user icon on the left. The text "beth" is entered.
- Password Field:** A light blue input box with a key icon on the left. The password is masked with dots.
- Login Button:** A blue button with the text "Login" in white.
- Link:** "Don't have an account? Sign Up" in a small, dark font at the bottom.

FIGURE 3.10 – Interface d'authentification

3.4.3 Interface de détail utilisateur

La figure 3.11 illustre l'interface de détail utilisateur :

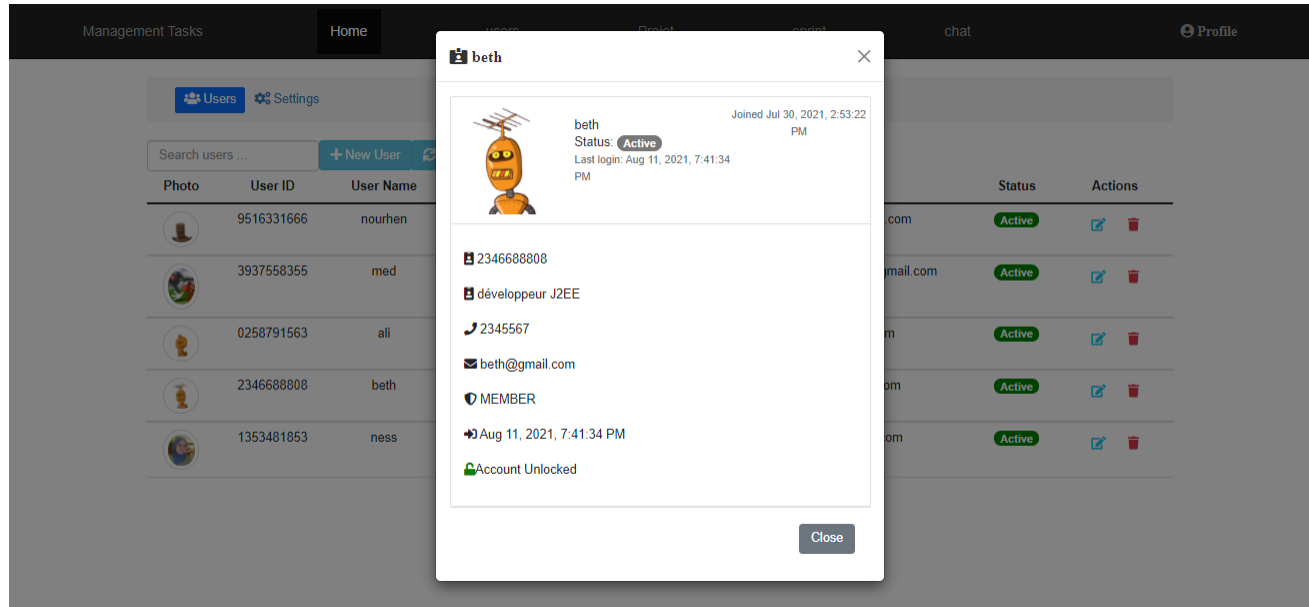


FIGURE 3.11 – Interface de détail utilisateur

3.4.4 Interface de modification utilisateur

A travers cette interface, l'administrateur ou le Scrum Master peuvent modifier un utilisateur approprié.

La figure 3.12 illustre l'interface de modification utilisateur :

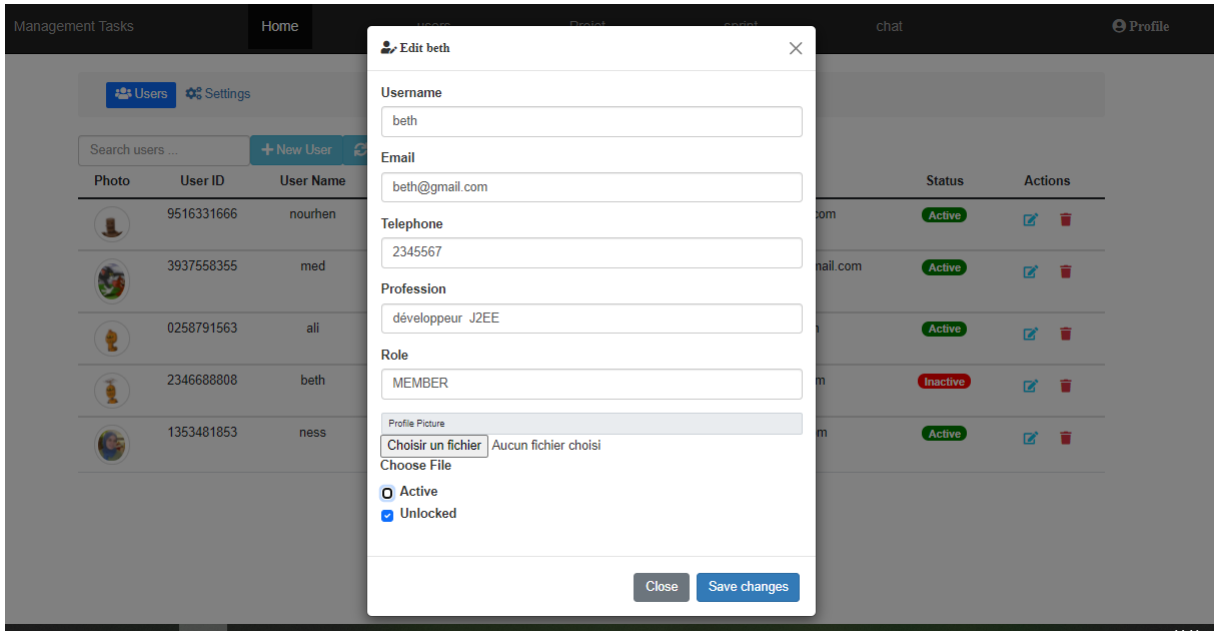


FIGURE 3.12 – Interface de modification utilisateur

3.4.5 Interface de profil utilisateur

A travers cette interface, l'utilisateur authentifié peut consulter son profil. La figure 3.13 illustre l'interface de profil utilisateur :

The screenshot displays a user profile management interface. At the top, a navigation bar includes links for 'Management Tasks', 'Home', 'users', 'Projet', 'sprint', 'chat', and a 'Profile' icon. The profile section on the left shows a user named 'ness' with ID '1353481853', a profile picture, and a 'Change Photo' button. Below this, form fields for 'Username', 'Profession', 'Telephone', 'Email', and 'Role' are shown, with the role currently set to 'ADMIN'. A 'Save Changes' button is at the bottom right of the form. On the right side, a 'Logout' button and a list of 'Permissions From Role' are visible. The permissions list includes actions for users, projects, sprints, and tasks, such as 'user:create', 'user:update', 'user:delete', 'projet:create', 'projet:update', 'projet:delete', 'sprint:create', 'sprint:update', 'sprint:delete', 'task:create', 'task:update', and 'task:delete'.

Permissions From Role
user:create
user:update
user:delete
projet:create
projet:update
projet:delete
sprint:create
sprint:update
sprint:delete
task:create
task:update
task:delete

FIGURE 3.13 – Interface de profil utilisateur

3.4.6 Interface de modification mot de passe

A travers cette interface, l'administrateur peut modifier le mot de passe d'un utilisateur.

La figure 3.14 illustre l'interface de modification mot de passe :

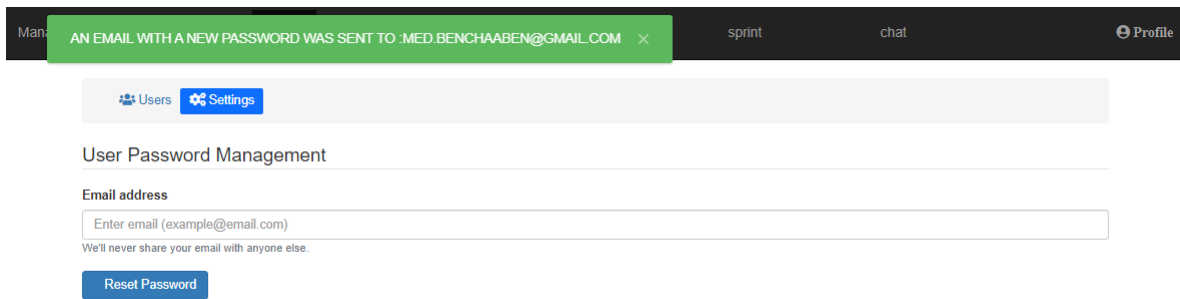


FIGURE 3.14 – Interface de modification mot de passe

Conclusion

Dans ce chapitre nous avons présenté le premier sprint en focalisant sur le sprint backlog, spécifications fonctionnelles, conception suivi par une démonstration de résultat de développement à l'aide des aperçus d'écran. Dans le chapitre suivant, nous allons présenter le deuxième sprint.

4

Sprint 2

Sommaire

Introduction	80
4.1 Backlog du sprint 2	80
4.2 Spécifications fonctionnelles	84
4.2.1 Raffinement de cas d'utilisation « Gérer Projets »	84
4.2.2 Raffinement de cas d'utilisation « Gérer Sprints »	86
4.2.3 Raffinement de cas d'utilisation « Gérer Tâches »	88
4.3 Conception	90
4.3.1 Diagramme de classes du deuxième sprint	90
4.3.2 Diagramme de séquence du cas d'utilisation « Supprimer projet »	91
4.3.3 Diagramme de séquence du cas d'utilisation « Modifier sprint »	92
4.3.4 Diagramme de séquence du cas d'utilisation « Consulter tâche »	93
4.3.5 Diagramme de séquence du cas d'utilisation « Modifier état tâche »	94
4.4 Réalisation	95
4.4.1 Interface d'ajouter projet	95
4.4.2 Interface de modification projet	96
4.4.3 Interface de détail projet	97
4.4.4 Interface liste des sprints	98
4.4.5 Interface de modification sprint	99

4.4.6	Interface de modification d'état sprint	100
4.4.7	Interface de suppression un sprint	101
4.4.8	Interface d'ajouter tâche	102
4.4.9	Interface de modification d'état d'une tâche	103
4.4.10	Interface liste des tâches	103
4.4.11	Interface de détail tâche	104
4.4.12	Interface du Tableau de bord	105
Conclusion	106

Introduction

Après avoir achevé le premier sprint, nous passons à la réalisation du deuxième sprint. Ce chapitre sera consacré en premier lieu à la présentation de son backlog qui sera spécifié à l'aide des diagrammes des cas d'utilisation. En second lieu, nous concevrons ce sprint avec un diagramme de classes, suivi des diagrammes de séquences. Pour finir, nous présentons quelques interfaces graphiques qui correspondent à des fonctionnalités importantes offertes par ce sprint.

4.1 Backlog du sprint 2

Le tableau 4.1 montre notre deuxième sprint backlog :

id	Fonctionnalité	id stories	User stories	Sprint	Priorité
5	Gestion des Projets	5.1	En tant qu'un Scrum Master (ou bien Product Owner) authentifié, je veux consulter la liste des projets.	2	12
		5.2	En tant qu'un Scrum Master authentifié, je veux ajouter un projet.	2	11
		5.3	En tant qu'un Scrum Master authentifié, je veux modifier un projet.	2	15
		5.4	En tant qu'un Scrum Master authentifié, je veux supprimer un projet.	2	14
		3.5	En tant qu'un Scrum Master (ou bien Product Owner) authentifié, je veux afficher le détail de chaque projet.	2	13
6	Gestion des Sprints	6.1	En tant qu'un Scrum Master authentifié (ou bien chef d'équipe affecté et authentifié), je veux consulter la liste des sprints.	2	17

		6.2	En tant qu'un Scrum Master authentifié, je veux ajouter un sprint.	2	16
		6.3	En tant qu'un Scrum Master authentifié, je veux supprimer un sprint.	2	18
		6.4	En tant qu'un Scrum Master authentifié, je veux modifier un sprint(l'état, chef d'équipe a affecter, date...).	2	19
		6.5	En tant qu'un Scrum Master authentifié(ou bien chef d'équipe affecté et authentifié), je veux afficher le détail de chaque sprint.	2	20
		6.6	En tant qu'un chef d'équipe affecté et authentifié, je veux modifier l'état de sprint.	2	21
7	Gestion des Tâches	7.1	En tant qu'un chef d'équipe authentifié(ou bien member d'équipe affecté et authentifié), je veux consulter la liste des tâches.	2	23
		7.2	En tant qu'un chef d'équipe authentifié, je veux ajouter une tâche.	2	22

7	7.3	En tant qu'un chef d'équipe authentifié, je veux modifier une tâche.	2	24
	7.4	En tant qu'un chef d'équipe authentifié, je veux supprimer une tâche.	2	25
	7.5	En tant qu'un chef d'équipe authentifié(ou bien member d'équipe affecté et authentifié), je veux afficher le détail de chaque Tâche.	2	26
	7.6	En tant qu'un member d'équipe affecté et authentifié , je veux modifier l'état de tâche.	2	27

TABLE 4.1 – Deuxième sprint Backlog

4.2 Spécifications fonctionnelles

Dans cette section nous présentons le raffinement du diagramme de cas d'utilisation pour chacune des fonctionnalités retenues dans ce sprint.

4.2.1 Raffinement de cas d'utilisation « Gérer Projets »

La figure 4.1 illustre le diagramme de cas d'utilisation gérer projets :

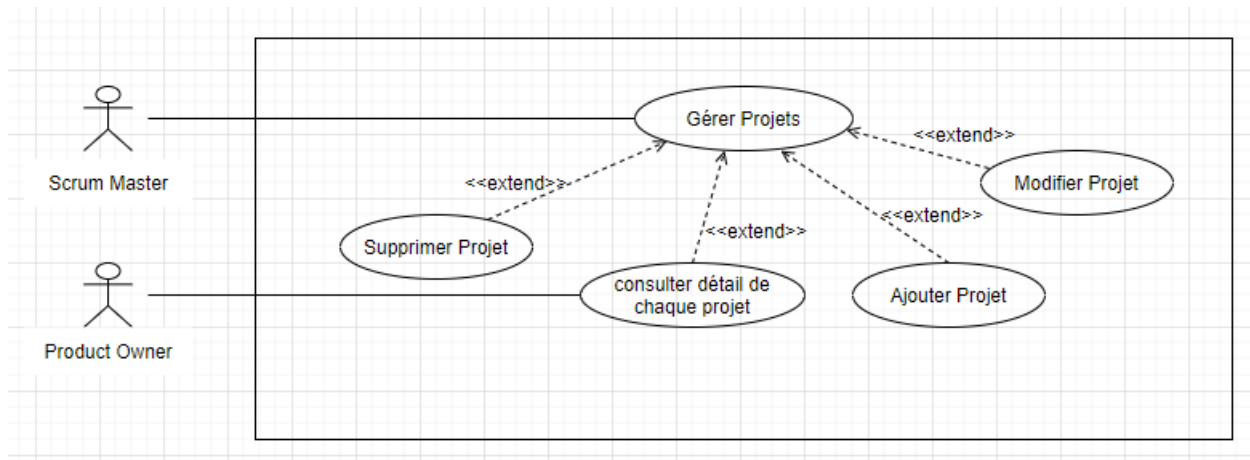


FIGURE 4.1 – Diagramme du cas d'utilisation «Gérer Projets »

Le tableau 4.2 présente une description textuelle de cas d'utilisation « Ajouter Projet »

Titre	Ajouter Projet
Objectif	L'acteur peut ajouter un projet.
Acteur	Administrateur/Scrum Master
Pré condition	-Acteur authentifié -L'acteur a consulté la liste de ses projets.
Post condition	Un projet est ajouté.
Description du scénario	-L'acteur clique sur «Ajouter Projet». -Le système affiche un formulaire d'ajout . -L'acteur remplir le formulaire d'ajout puis cliquer sur «Ajouter». -Le système enregistre le nouveau projet et affiche un message de succès.
Exception	-Tous les champs sont obligatoires ce pour cela il doit vérifier qu'il a bien remplir le formulaire. -Si l'acteur choisit un nom de projet déjà existant ,le système affiche un message d'erreur.

TABLE 4.2 – Description textuelle du cas d'utilisation «Ajouter Projet»

4.2.2 Raffinement de cas d'utilisation «Gérer Sprints»

La figure 4.2 illustre le diagramme de cas d'utilisation gérer sprints :

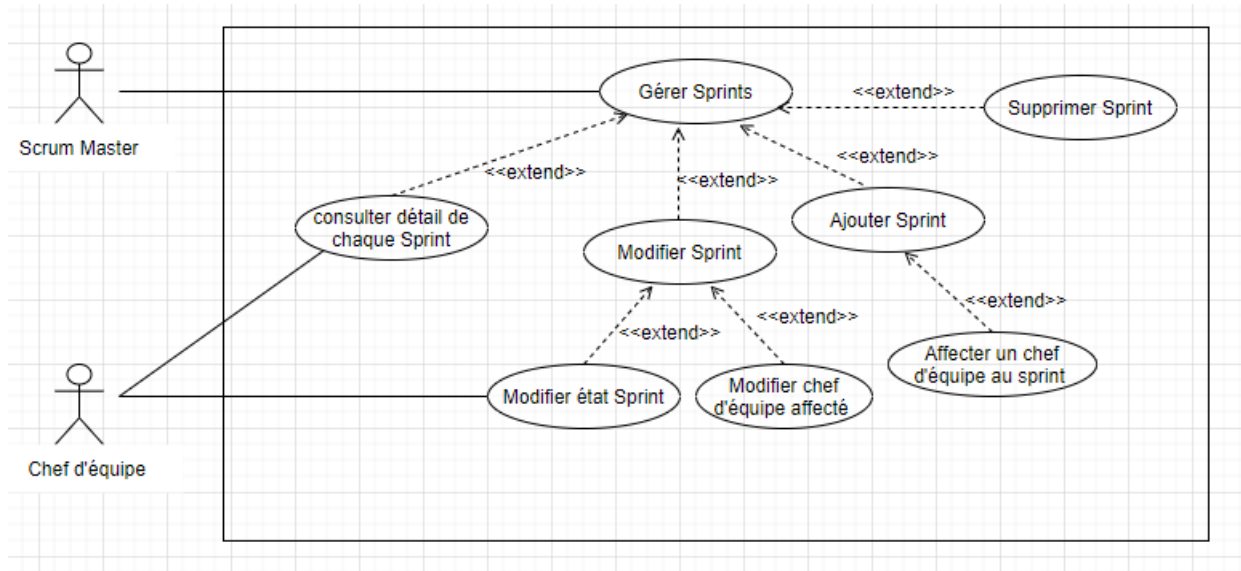


FIGURE 4.2 – Diagramme du cas d'utilisation «Gérer Sprints »

Le tableau 4.3 présente une description textuelle de cas d'utilisation «Modifier état sprint »

Titre	Modifier état sprint
Objectif	L'acteur peut modifier l'état d'un sprint.
Acteur	Administrateur/Scrum Master/Chef d'équipe
Pré condition	-Acteur authentifié et s'il s'agit d'un chef d'équipe, il faut être affecter a ce sprint. -l'acteur a consulté la liste des sprints.
Post condition	état du sprint modifié.
Description du scénario	-L'acteur clique sur «modifier sprint». -L'acteur sélectionne un état souhaité puis cliquer sur «Sauvegarder les modifications». -Le système enregistre les modifications et affiche un message de succès .
Exception	-Tous les champs sont obligatoires ce pour cela il doit vérifier qu'il a bien rempli le formulaire . -La modification d'état du sprint oblige l'affectation de chef d'équipe par le scrum Master. -La gestion des sprints oblige la présence de projet .

TABLE 4.3 – Description textuelle du cas d'utilisation«Modifier état sprint»

4.2.3 Raffinement de cas d'utilisation « Gérer Tâches »

La figure 4.3 illustre le diagramme de cas d'utilisation gérer tâches :

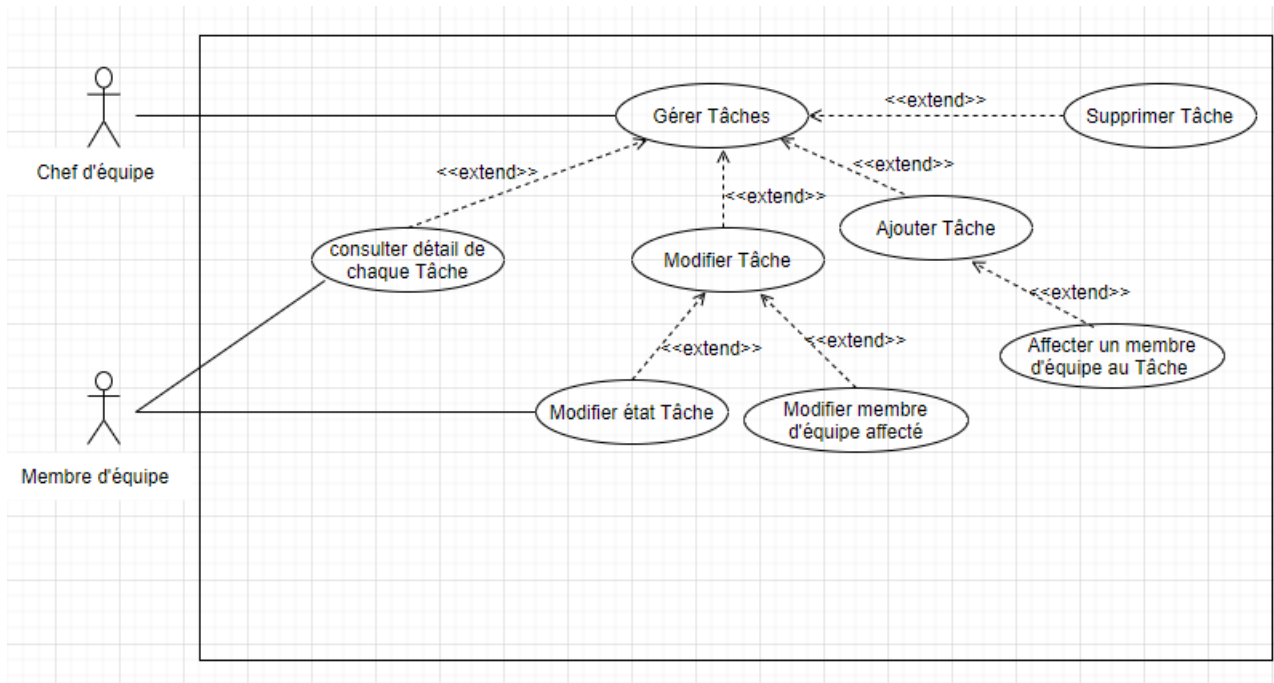


FIGURE 4.3 – Diagramme du cas d'utilisation « Gérer Tâches »

Le tableau 4.4 présente une description textuelle de cas d'utilisation « Supprimer tâche »

Titre	Supprimer tâche
Objectif	L'acteur peut supprimer une tâche.
Acteur	Administrateur/Scrum Master/Chef d'équipe
Pré condition	-Acteur authentifié -L'acteur choisit le sprint pour gérer les tâches à l'intérieur.
Post condition	Une tâche est supprimée.
Description du scénario	-L'acteur clique sur «Supprimer tâche» -Le système supprime la tâche et affiche un message de succès.
Exception	-Si l'acteur n'a pas la permission de suppression,le système affiche un message d'erreur.

TABLE 4.4 – Description textuelle du cas d'utilisation «Supprimer Tâche»

4.3.2 Diagramme de séquence du cas d'utilisation « Supprimer projet »

La figure 4.5 illustre le diagramme de séquence du cas d'utilisation « Supprimer projet » :

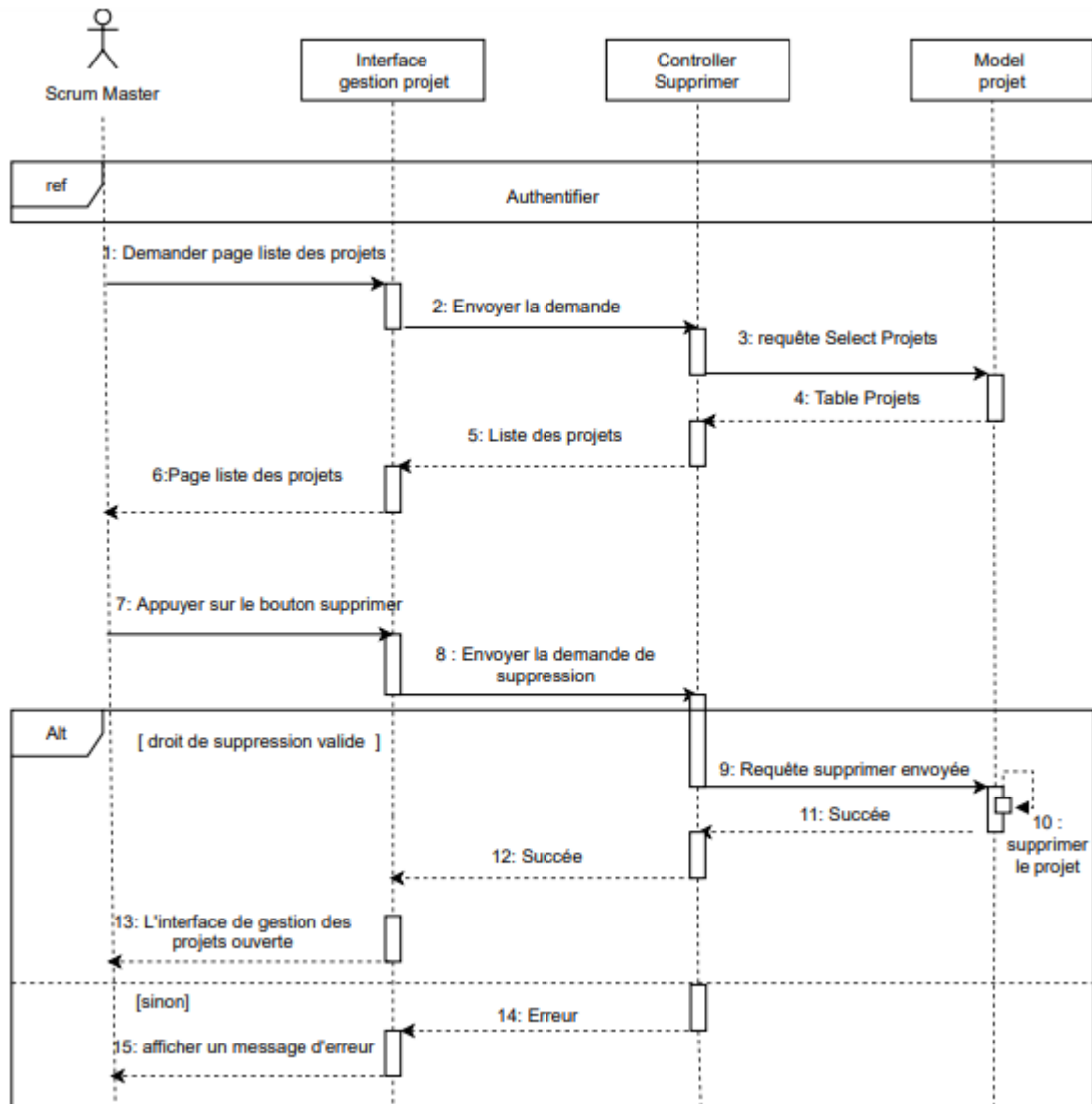


FIGURE 4.5 – Diagramme de séquence du cas d'utilisation « Supprimer Projet »

4.3.3 Diagramme de séquence du cas d'utilisation « Modifier sprint »

La figure 4.6 illustre le diagramme de séquence du cas d'utilisation « Modifier sprint » :

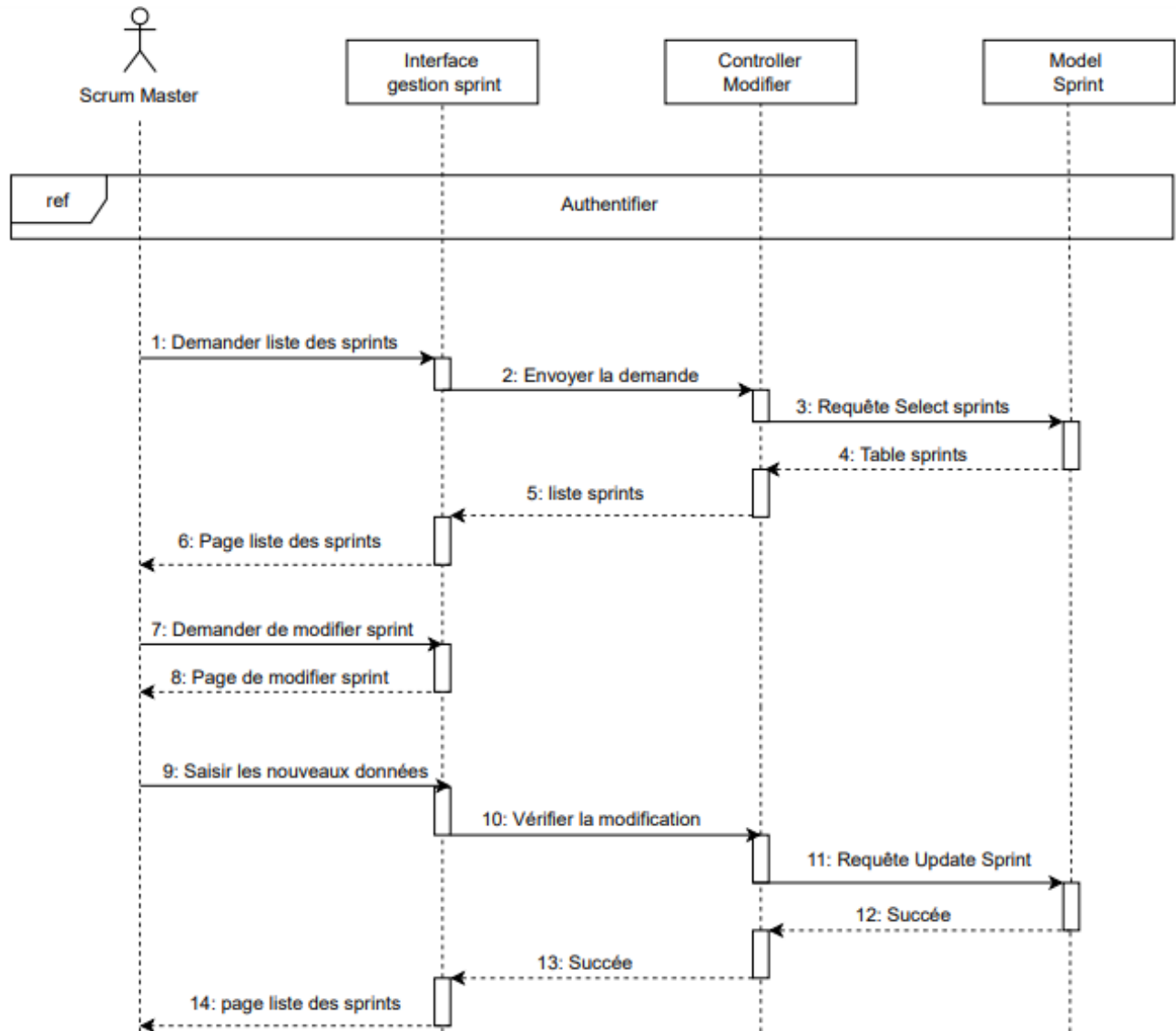


FIGURE 4.6 – Diagramme de séquence du cas d'utilisation « Modifier sprint »

4.3.4 Diagramme de séquence du cas d'utilisation « Consulter tâche »

La figure 4.7 illustre le diagramme de séquence du cas d'utilisation « Consulter tâche » :

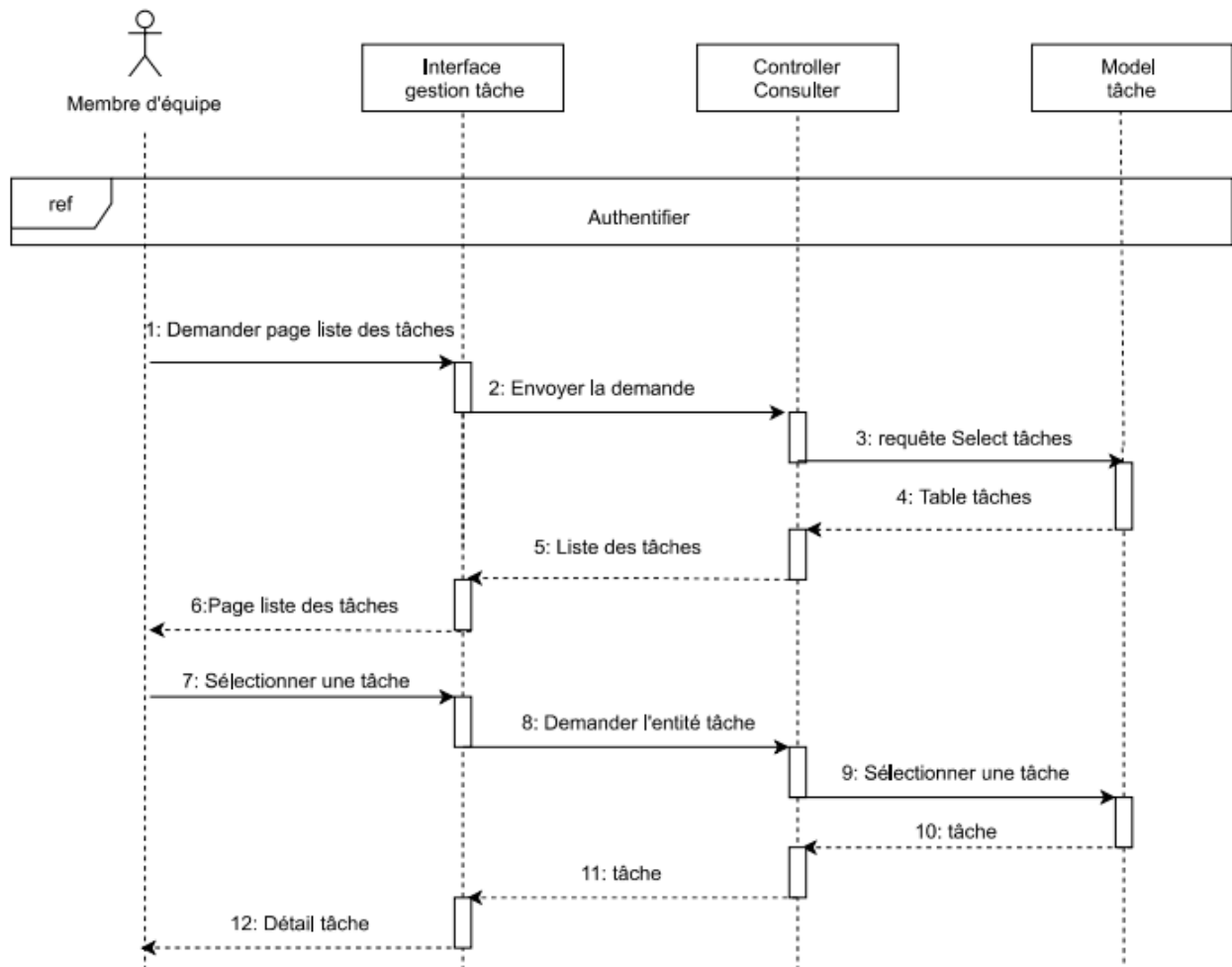


FIGURE 4.7 – Diagramme de séquence du cas d'utilisation « Consulter tâche »

4.3.5 Diagramme de séquence du cas d'utilisation « Modifier état tâche »

La figure 4.8 illustre le diagramme de séquence du cas d'utilisation « Modifier état tâche » :

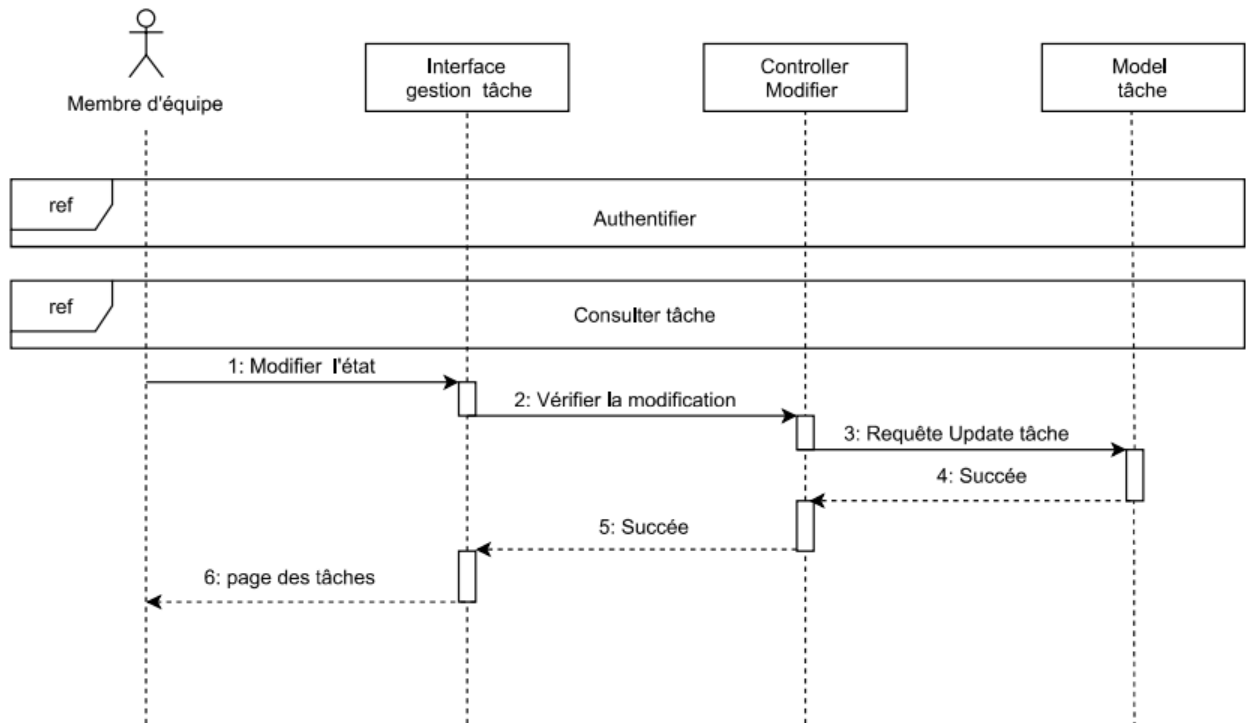


FIGURE 4.8 – Diagramme de séquence du cas d'utilisation « Modifier état tâche »

4.4 Réalisation

Passons maintenant à la présentation de la phase de réalisation du deuxième sprint. Nous exposons quelques interfaces réalisées durant ce sprint. Chaque interface illustre une fonctionnalité donnée.

4.4.1 Interface d'ajouter projet

Cette interface permet à un Scrum Master de créer un projet. La figure 4.9 illustre l'interface d'ajouter projet :

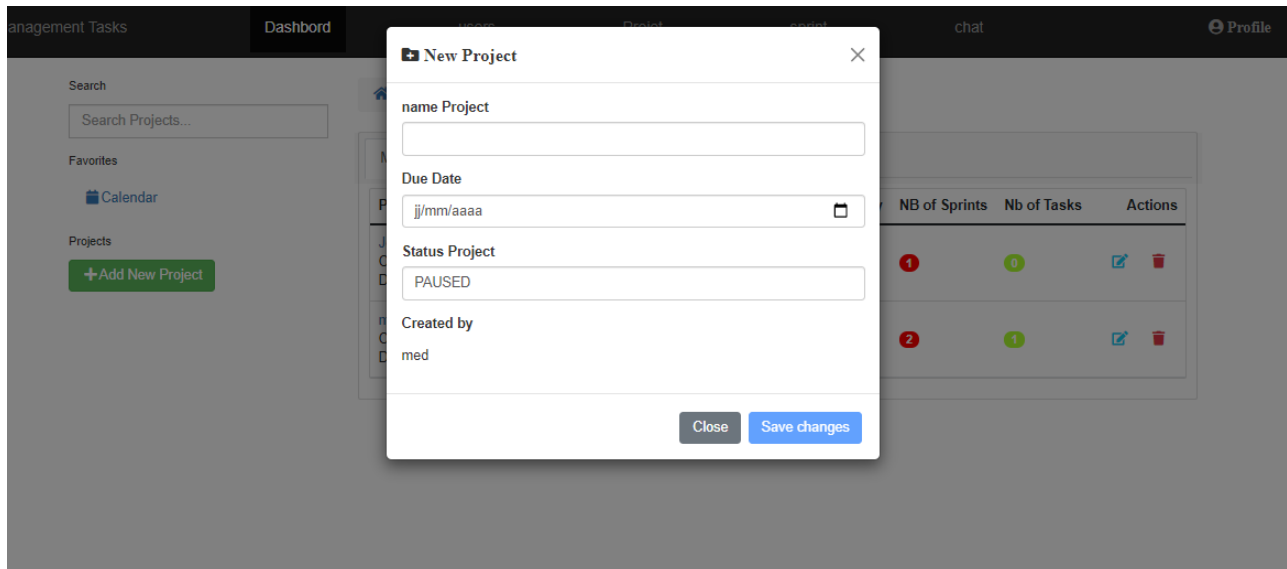


FIGURE 4.9 – Ajouter projet

4.4.2 Interface de modification projet

Cette interface permet à un Scrum Master de modifier un projet. La figure 4.10 illustre l'interface de modification projet :





NB of Sprints	Nb of Tasks	Actions
1	0	 
2	1	 

FIGURE 4.10 – l'interface de modification projet

4.4.3 Interface de détail projet

Cette interface permet aux Scrum Master ou Product owner de consulter détail d'un projet.

La figure 4.20 illustre l'interface de détail projet :

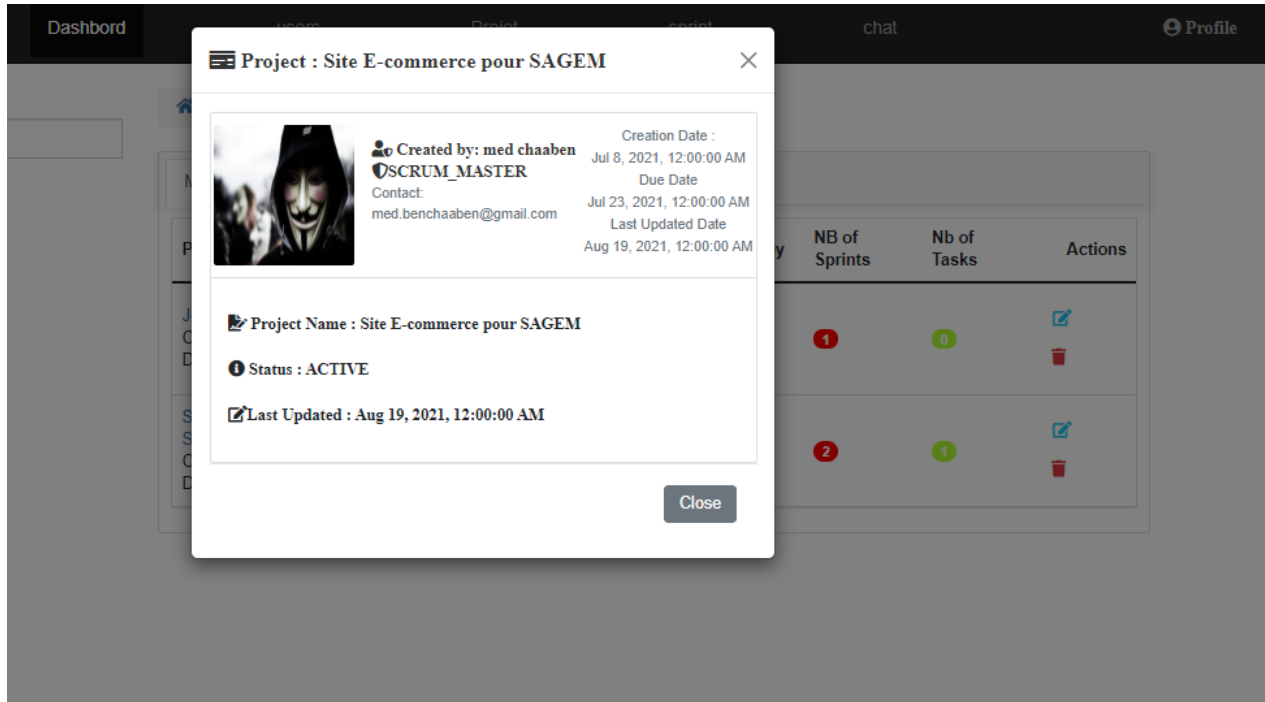


FIGURE 4.11 – l'interface de détail projet

4.4.4 Interface liste des sprints

Cette interface permet à l'acteur de consulter la liste des sprints. La figure 4.19 illustre l'interface liste des sprints :

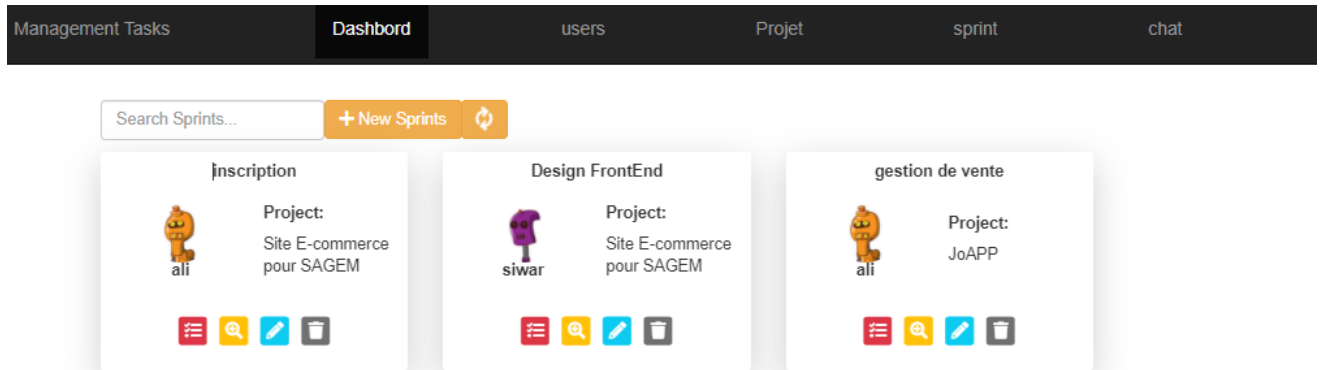


FIGURE 4.12 – Interface liste des sprints

4.4.5 Interface de modification sprint

Cette interface permet le Scrum Master de modifier les informations d'un sprint .

La figure 4.13 illustre l'interface de modification sprint :

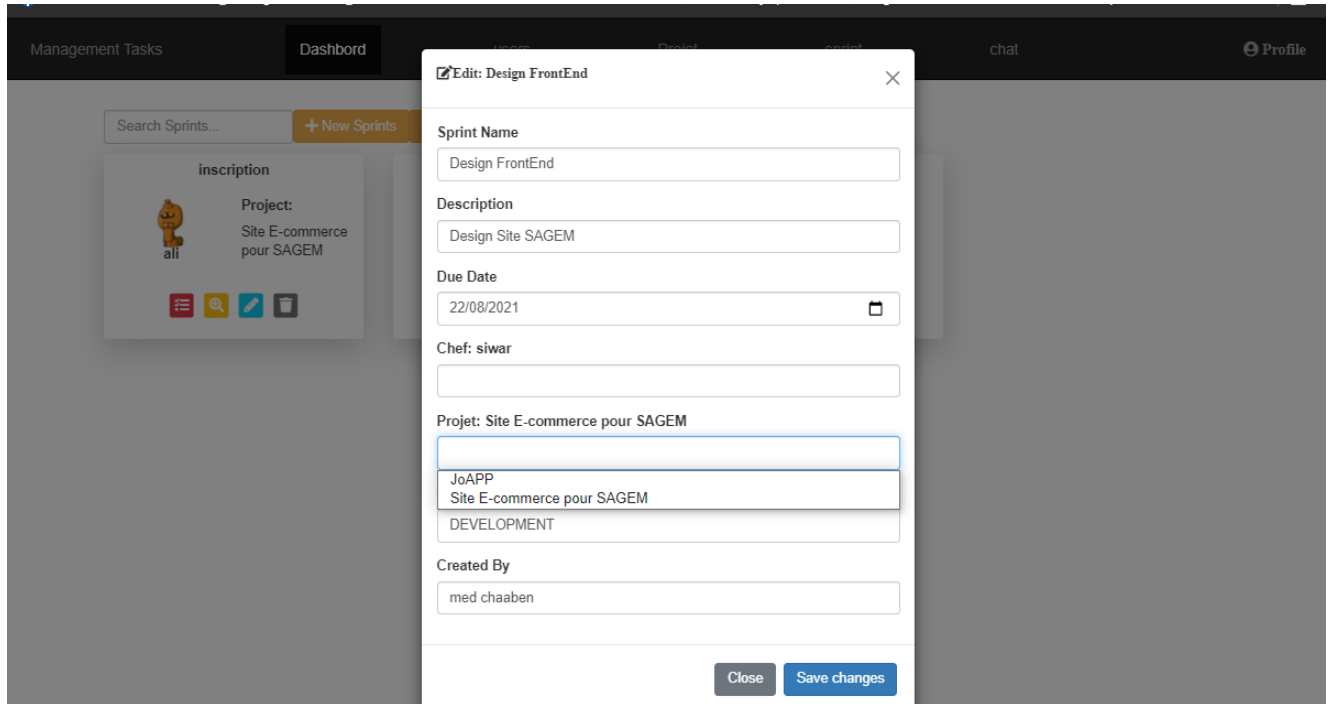


FIGURE 4.13 – l'interface de modification sprint

4.4.6 Interface de modification d'état sprint

La figure 4.14 illustre l'interface de modification d'état sprint par le chef d'équipe affecté :

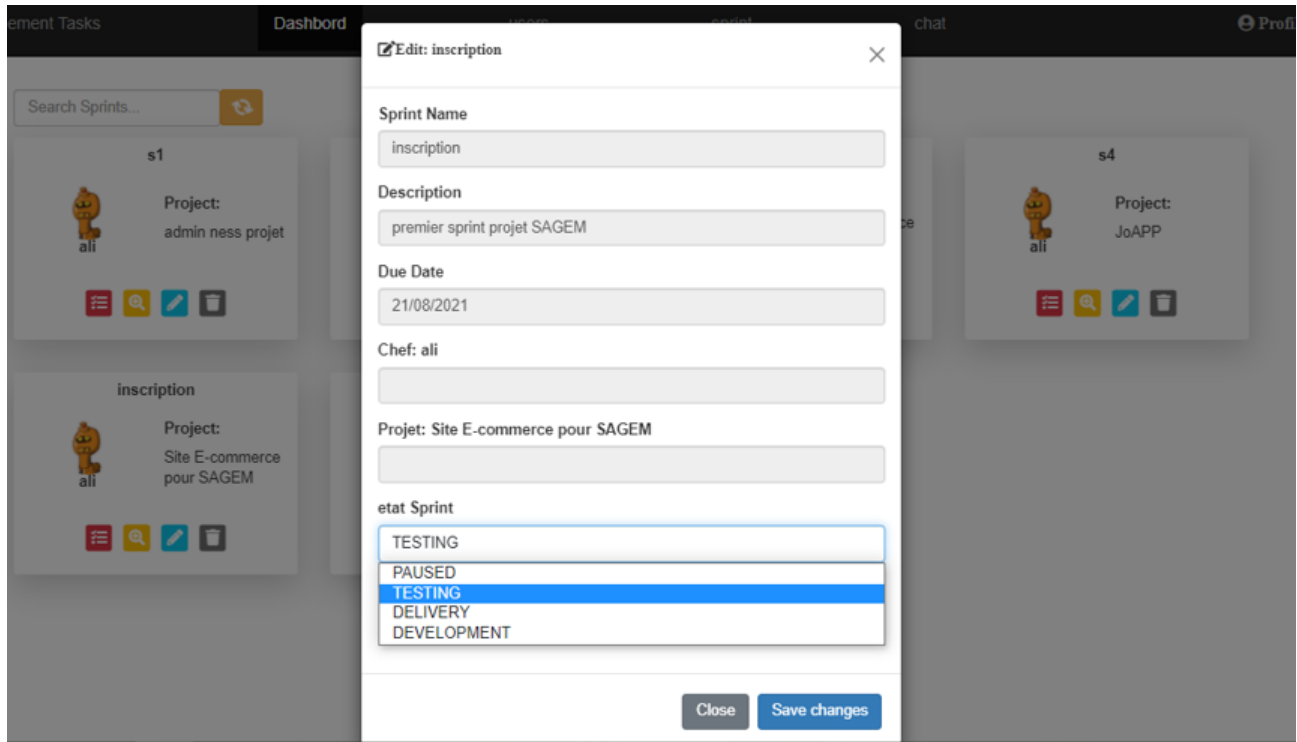


FIGURE 4.14 – l'interface de modification d'état sprint

4.4.7 Interface de suppression un sprint

La figure 4.15 illustre l'interface de suppression sprint avec permission :

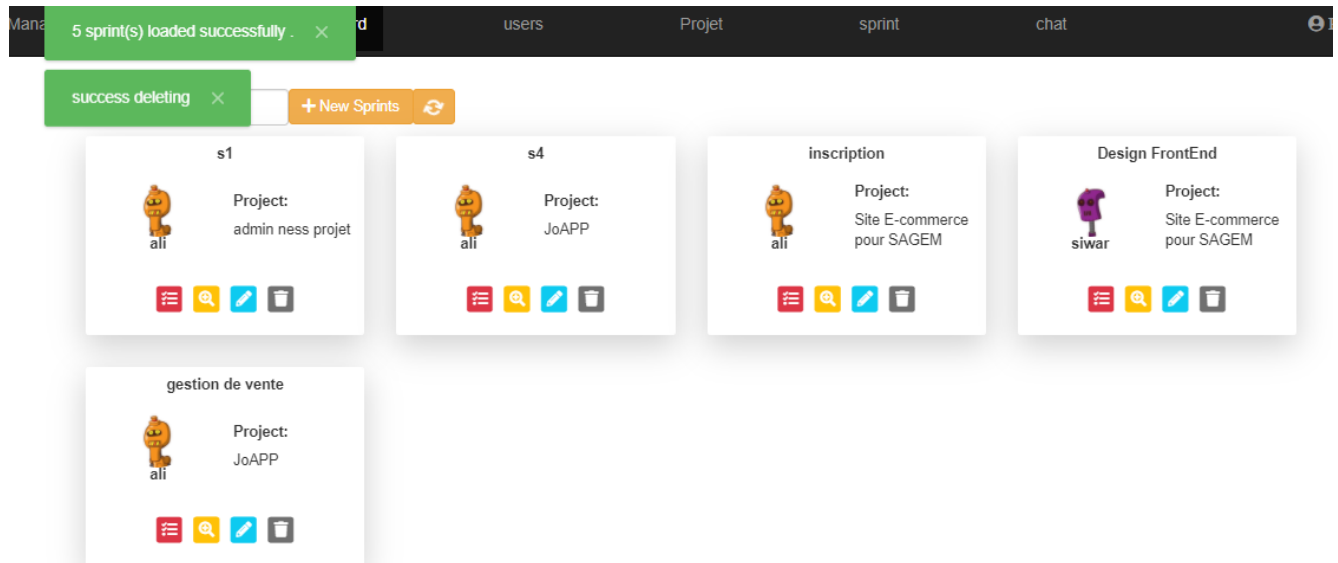


FIGURE 4.15 – Suppression sprint avec permission

La figure 4.16 illustre l'interface de suppression sprint sans permission :

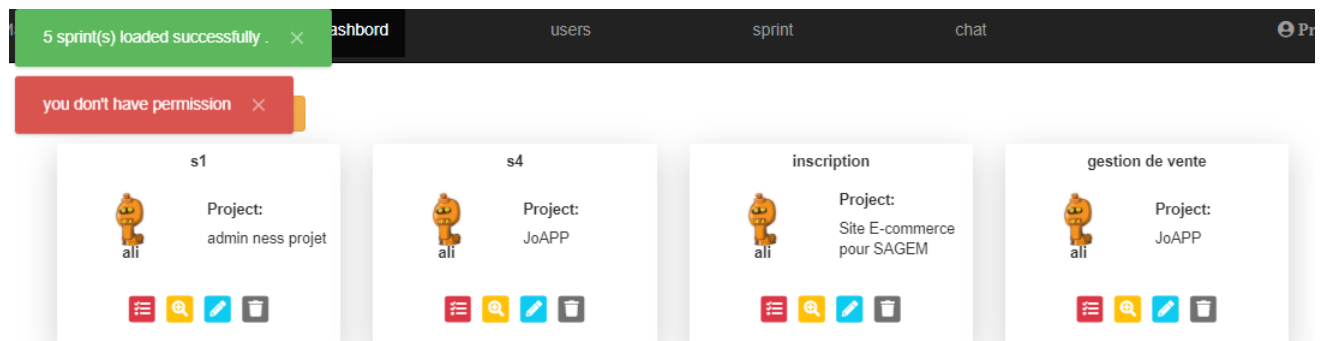


FIGURE 4.16 – Suppression sprint sans permission

4.4.8 Interface d'ajouter tâche

cette interface permet à l'acteur de créer une tâche.
La figure 4.17 illustre l'interface d'ajouter tâche :

The image shows a 'New Task' modal form overlaid on a blurred background of a task management dashboard. The modal has a title bar with a hamburger menu icon, the text 'New Task', and a close button (X). The form contains the following fields and controls:

- Task Name:** A text input field.
- Description:** A text input field.
- Due Date:** A date input field with a placeholder 'jj/mm/aaaa' and a calendar icon.
- Member:** A text input field.
- Sprint:** A text input field.
- etat Task:** A dropdown menu currently showing 'UNSTARTED'.
- Created by:** A text field showing the name 'ali'.
- Archive:** A checkbox with the label 'Archive'.
- Buttons:** At the bottom right, there are two buttons: 'Close' (grey) and 'Save changes' (blue).

The background dashboard includes a 'Management Tasks' header, a 'Dashboard' tab, a 'Search Sprints...' input, a '+ New Tasks' button, and a 'To Do' section with a task card for 'Web app security' assigned to 'beth'.

FIGURE 4.17 – Ajouter tâche

4.4.9 Interface de modification d'état d'une tâche

La figure 4.18 illustre l'interface de modification d'état d'une tâche par le membre d'équipe affecté :

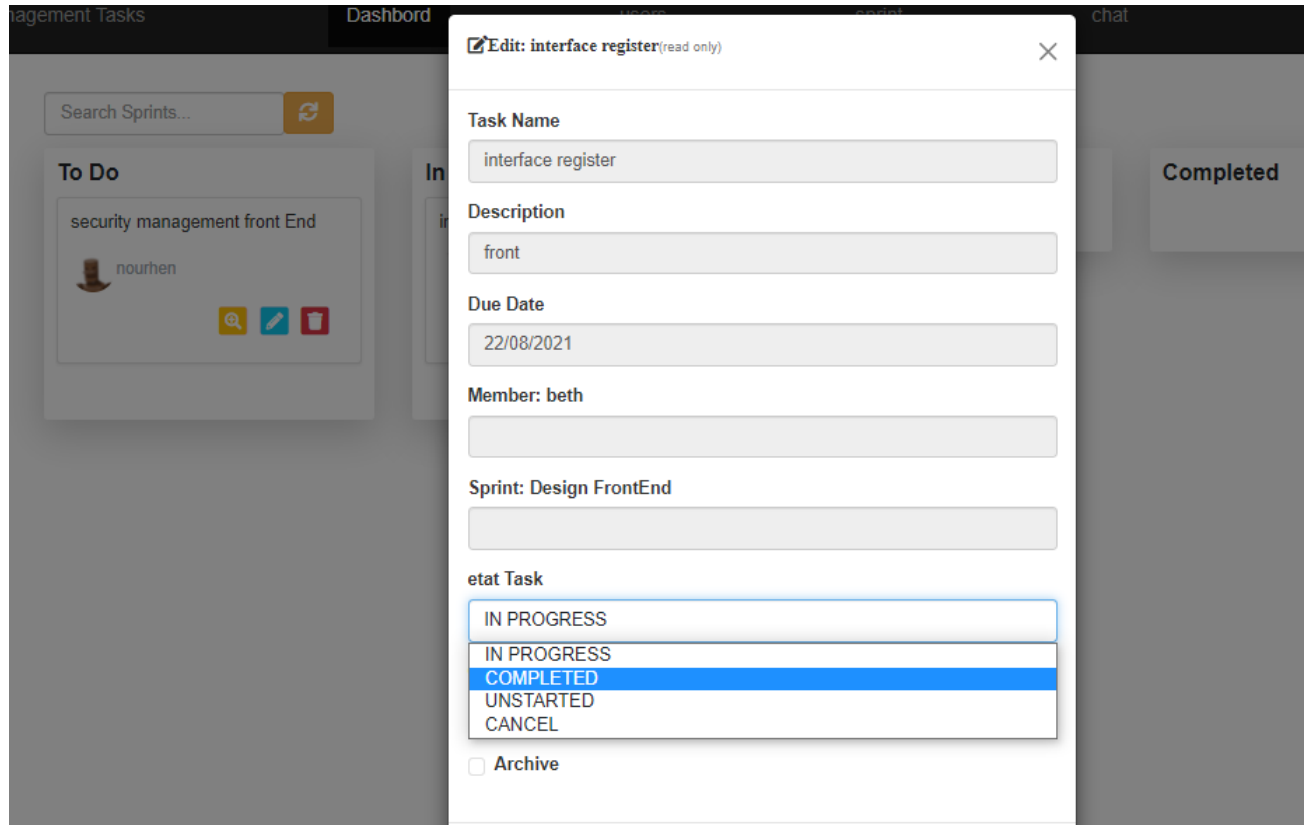


FIGURE 4.18 – l'interface de modification d'état d'une tâche

4.4.10 Interface liste des tâches

cette interface permet aux acteurs de consulter la liste des tâches liée à un sprint .

La figure 4.19 illustre l'interface liste des tâches :

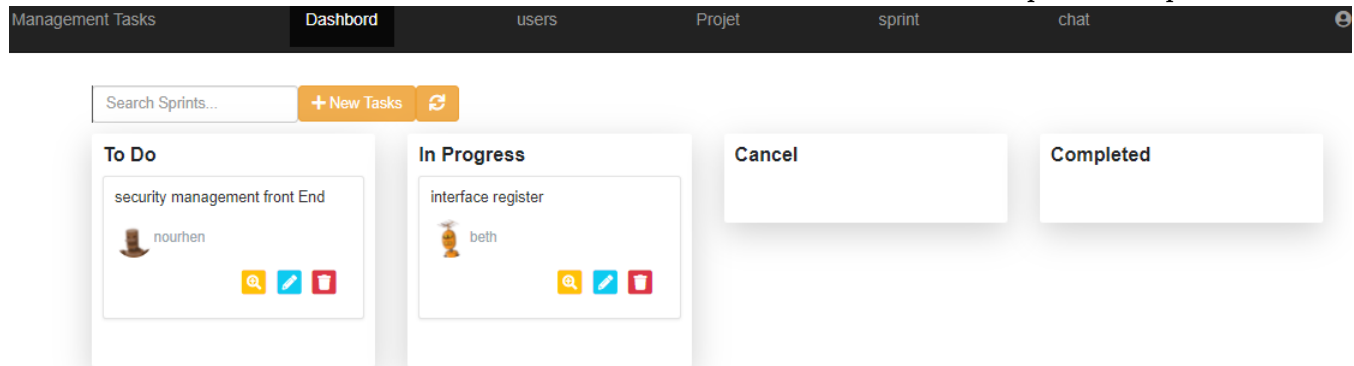


FIGURE 4.19 – Interface liste des tâches

4.4.11 Interface de détail tâche

La figure 4.20 illustre l'interface de détail tâche :

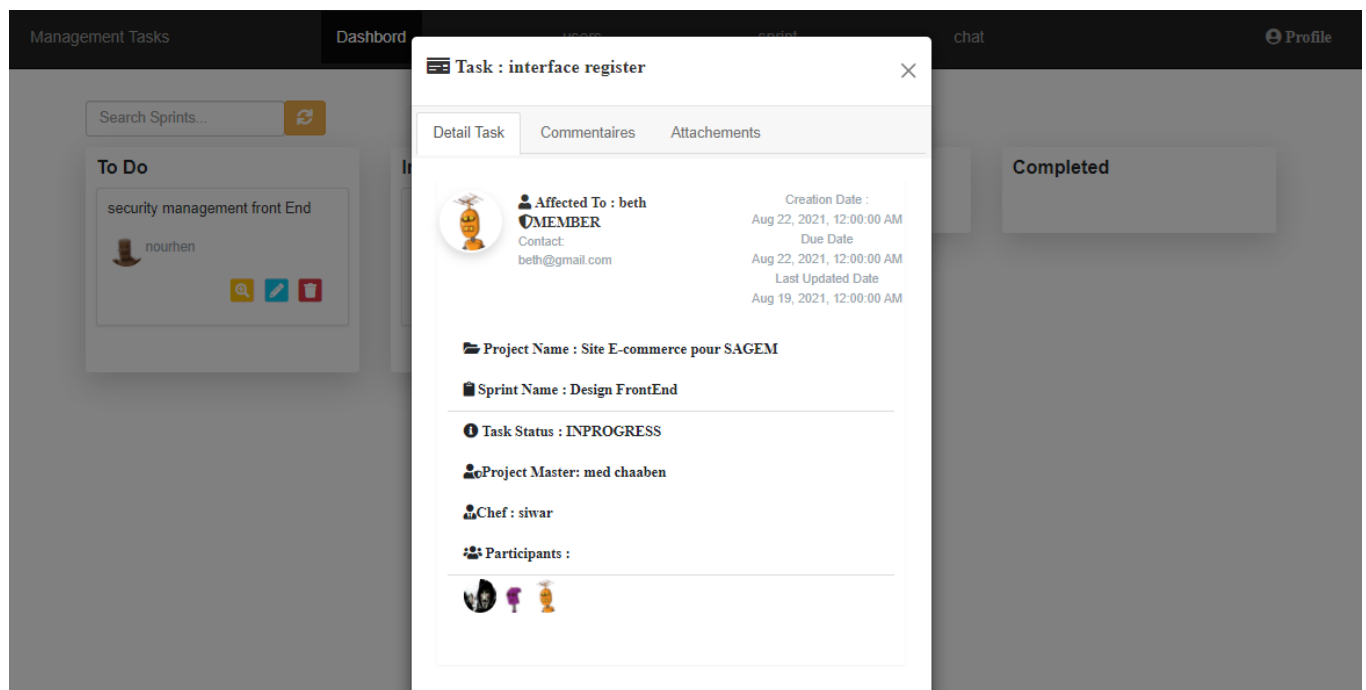


FIGURE 4.20 – l'interface de détail tâche

4.4.12 Interface du Tableau de bord

Cette interface du Tableau de bord accompagné de toutes les mesures clés :

- les nombres des projets selon leurs états.
- les nombres des sprints selon leurs états.
- les nombres des tâches selon leurs états.

La figure 4.21 illustre l'interface du Tableau de bord :

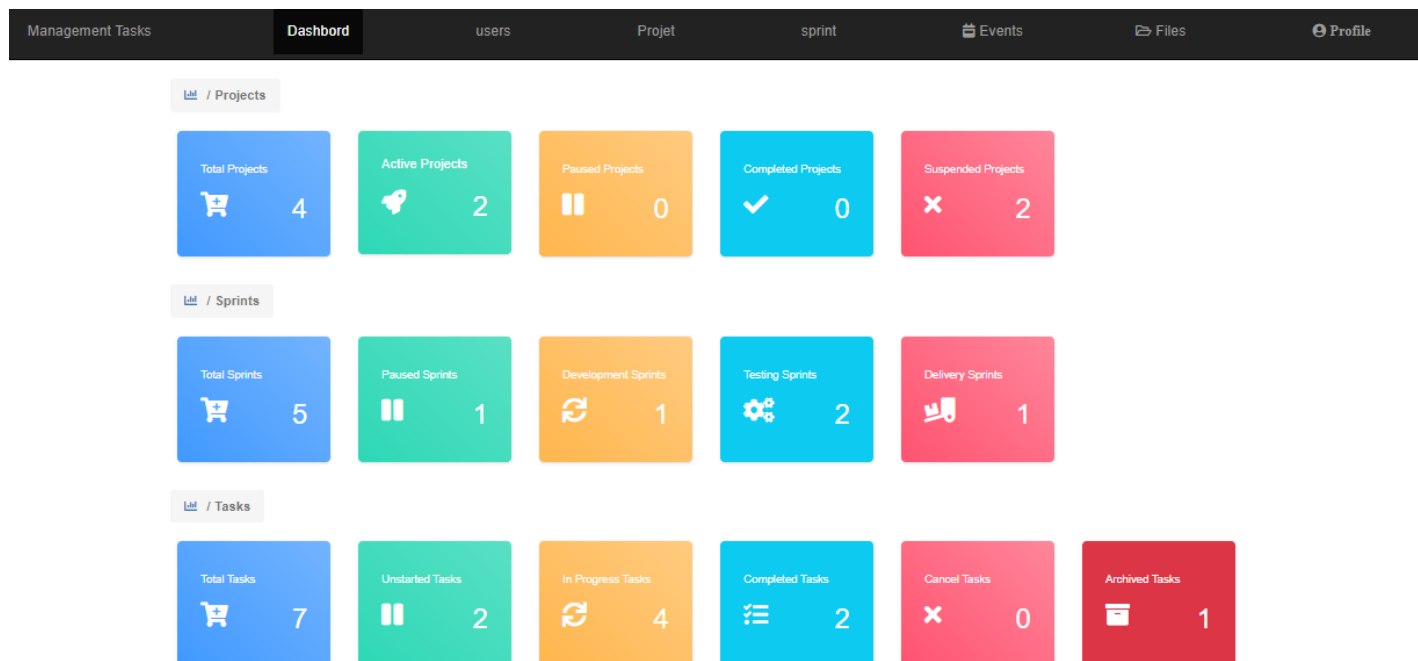


FIGURE 4.21 – l'interface du Tableau de bord

Conclusion

Ce chapitre a présenté la spécification des fonctionnalités du deuxième sprint à l'aide des diagrammes de cas d'utilisation. Ensuite, il a donné l'aspect conceptuel en listant les résultats attendus en termes de fonctionnalités. Ces dernières ont été illustrées par quelques interfaces réalisées. Le chapitre suivant fera l'objet d'une étude détaillée du troisième sprint.

5

Sprint 3

Sommaire

Introduction	108
5.1 Backlog du sprint 3	108
5.2 Spécifications fonctionnelles	110
5.2.1 Raffinement de cas d'utilisation «Gérer Commentaires»	110
5.2.2 Raffinement de cas d'utilisation « Gérer Attachements »	112
5.3 Conception	114
5.3.1 Diagramme de classes du troisième sprint	114
5.3.2 Diagramme de séquence du cas d'utilisation « Ajouter commentaire »	115
5.3.3 Diagramme de séquence du cas d'utilisation « Supprimer commentaire »	116
5.3.4 Diagramme de séquence du cas d'utilisation « Ajouter Attachement »	117
5.4 Réalisation	117
5.4.1 Interface gérer commentaires	118
5.4.2 Interface gérer attachemnts	119
Conclusion	120

Introduction

Le but principal du troisième sprint est de présenter les deux cas d'utilisation : gérer commentaires et gérer attachements. Ceci va être élaboré avec les mêmes démarches du sprint 1 et 2.

5.1 Backlog du sprint 3

Le tableau 5.1 montre notre troisième sprint backlog :

id	Fonctionnalité	id stories	User stories	Sprint	Priorité
8	Gestion des Commentaires	8.1	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux consulter la liste des commentaires.	3	29
		8.2	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux ajouter un commentaire.	3	28
		8.3	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux supprimer un commentaire.	3	30
9	Gestion des Attachements	9.1	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux consulter la liste des attachements.	3	32

9		9.2	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux ajouter un attachement.	3	31
		9.3	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux supprimer un attachement.	3	33
		9.4	En tant qu'un member d'équipe authentifié et affecté à une tâche, je veux télécharger un attachement.	3	34

TABLE 5.1 – Troisième sprint Backlog

5.2 Spécifications fonctionnelles

Dans cette section nous présentons le raffinement du diagramme de cas d'utilisation pour chacune des fonctionnalités retenues dans ce sprint.

5.2.1 Raffinement de cas d'utilisation «Gérer Commentaires»

La figure 5.1 illustre le diagramme de cas d'utilisation « Gérer Commentaires » :

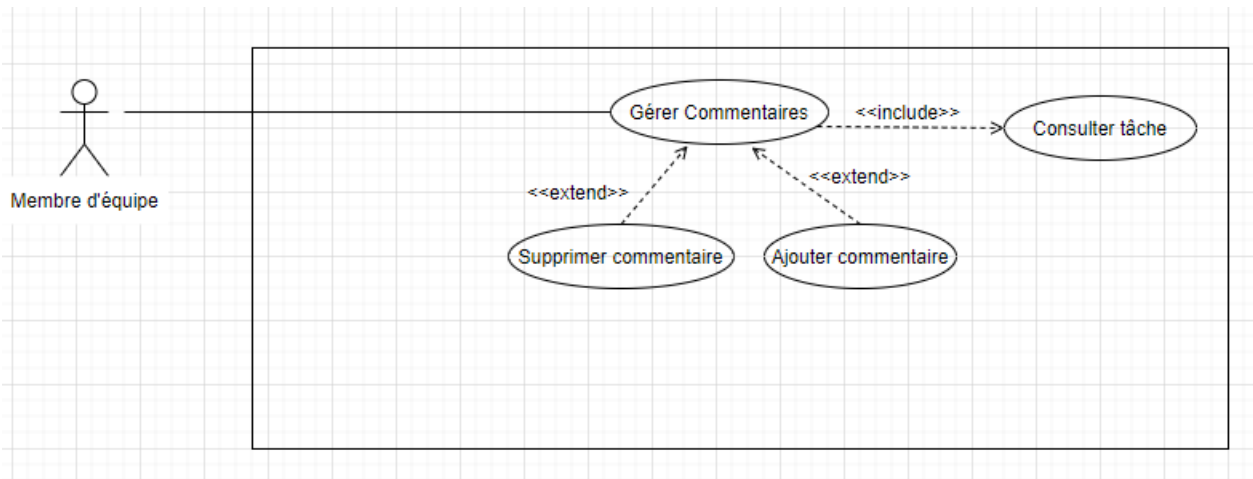


FIGURE 5.1 – Diagramme du cas d'utilisation « Gérer Commentaires »

Le tableau 5.2 présente une description textuelle de cas d'utilisation « Ajouter Commentaire »

Titre	Ajouter Commentaire
Objectif	L'acteur peut ajouter un commentaire.
Acteur	Administrateur/Scrum Master/Chef d'équipe/Membre d'équipe
Pré condition	-Acteur authentifié l'acteur choisit la tâche pour gérer les commentaires à l'intérieur.
Post condition	un commentaire est ajouté.
Description du scénario	-L'acteur remplir le champ commentaire puis clique sur «Ajouter». -Le système enregistre le nouveau commentaire et affiche un message de succès .
Exception	-La gestion des commentaires oblige que le membre d'équipe est affecté à une tâche. -La gestion des commentaires oblige la présence de tâche .

TABLE 5.2 – Description textuelle du cas d'utilisation «Ajouter Commentaire»

5.2.2 Raffinement de cas d'utilisation « Gérer Attachements »

La figure 5.2 illustre le diagramme de cas d'utilisation « Gérer Attachements » :

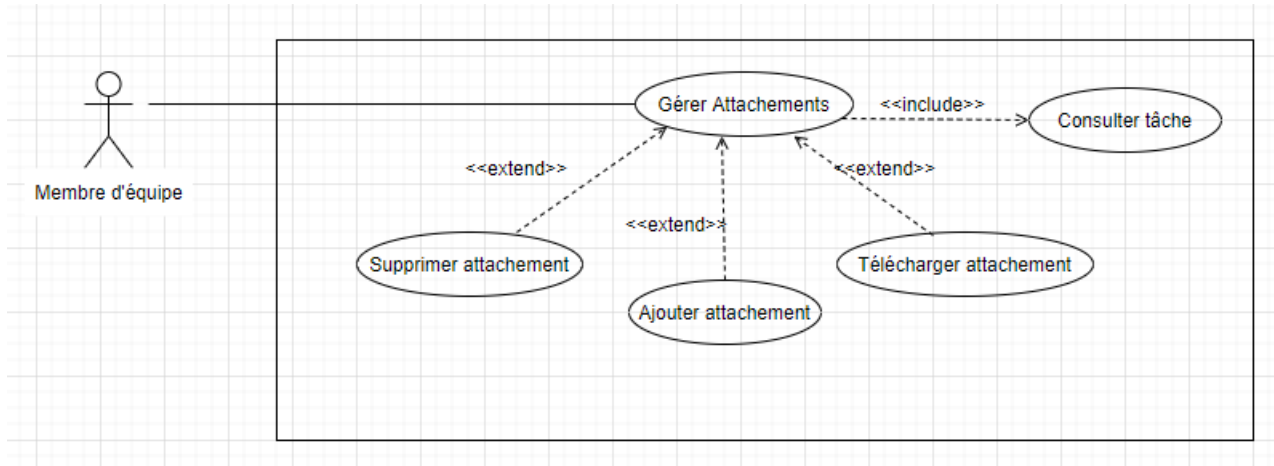


FIGURE 5.2 – Diagramme du cas d'utilisation « Gérer Attachements »

Le tableau 5.3 présente une description textuelle de cas d'utilisation « Télécharger Attachement »

Titre	Télécharger Attachement
Objectif	L'acteur peut télécharger un attachement.
Acteur	Administrateur/Scrum Master/Chef d'équipe/Membre d'équipe
Pré condition	-Acteur authentifié l'acteur choisit la tâche pour gérer les attachements à l'intérieur.
Post condition	un attachement est téléchargé.
Description du scénario	-L'acteur clique sur «Télécharger». -Le système télécharge l'attachement et affiche un message de succès .
Exception	-La gestion des attachements oblige que le membre d'équipe est affecté à une tâche. -La gestion des attachements oblige la présence de tâche .

TABLE 5.3 – Description textuelle du cas d'utilisation«Télécharger Attache-ment»

5.3 Conception

5.3.1 Diagramme de classes du troisième sprint

La figure 5.3 illustre le diagramme de classe de troisième sprint :

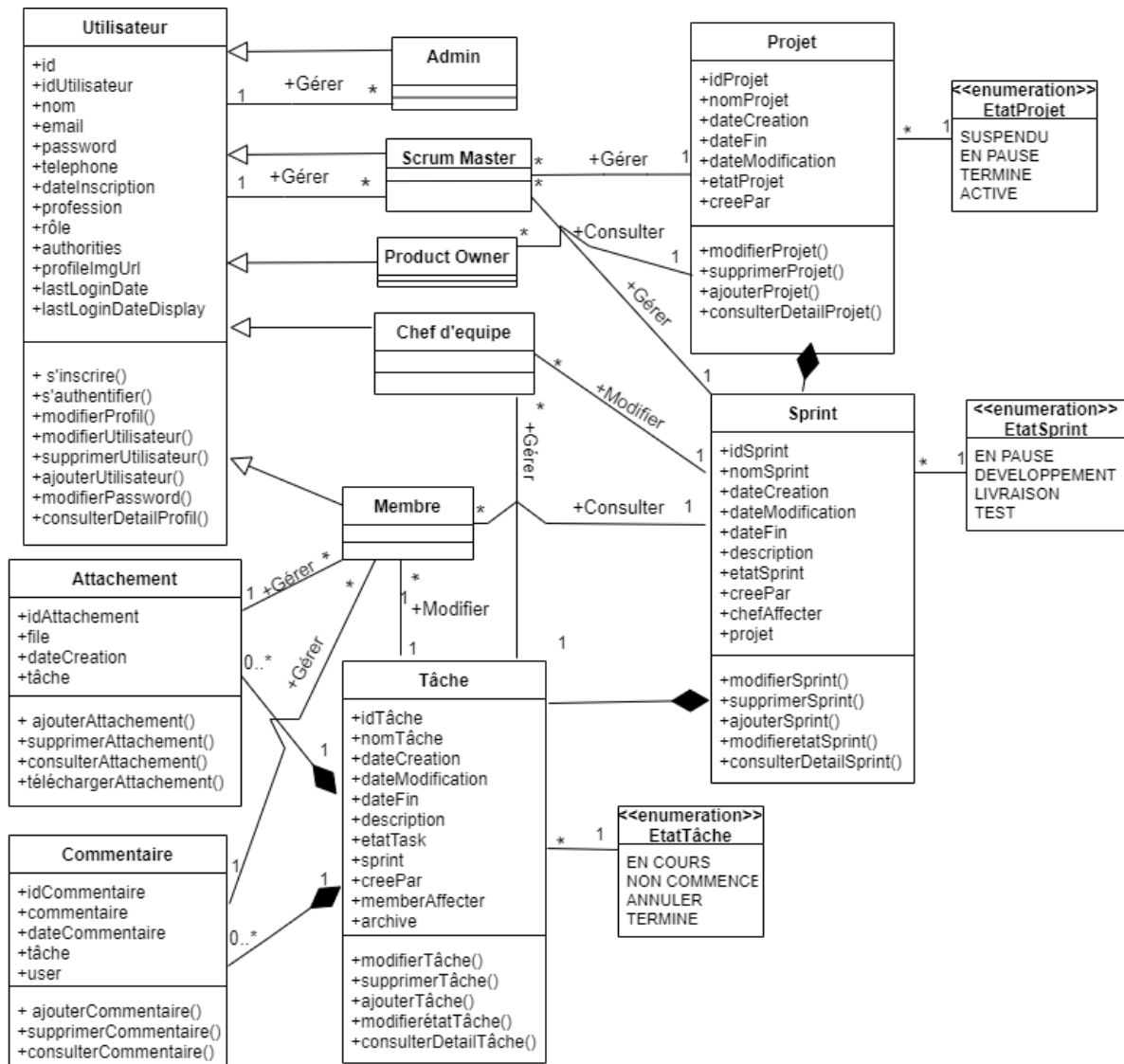


FIGURE 5.3 – Diagramme de classes« Troisième Sprint »

5.3.2 Diagramme de séquence du cas d'utilisation « Ajouter commentaire »

La figure 5.4 illustre le diagramme de séquence du cas d'utilisation « Ajouter commentaire » :

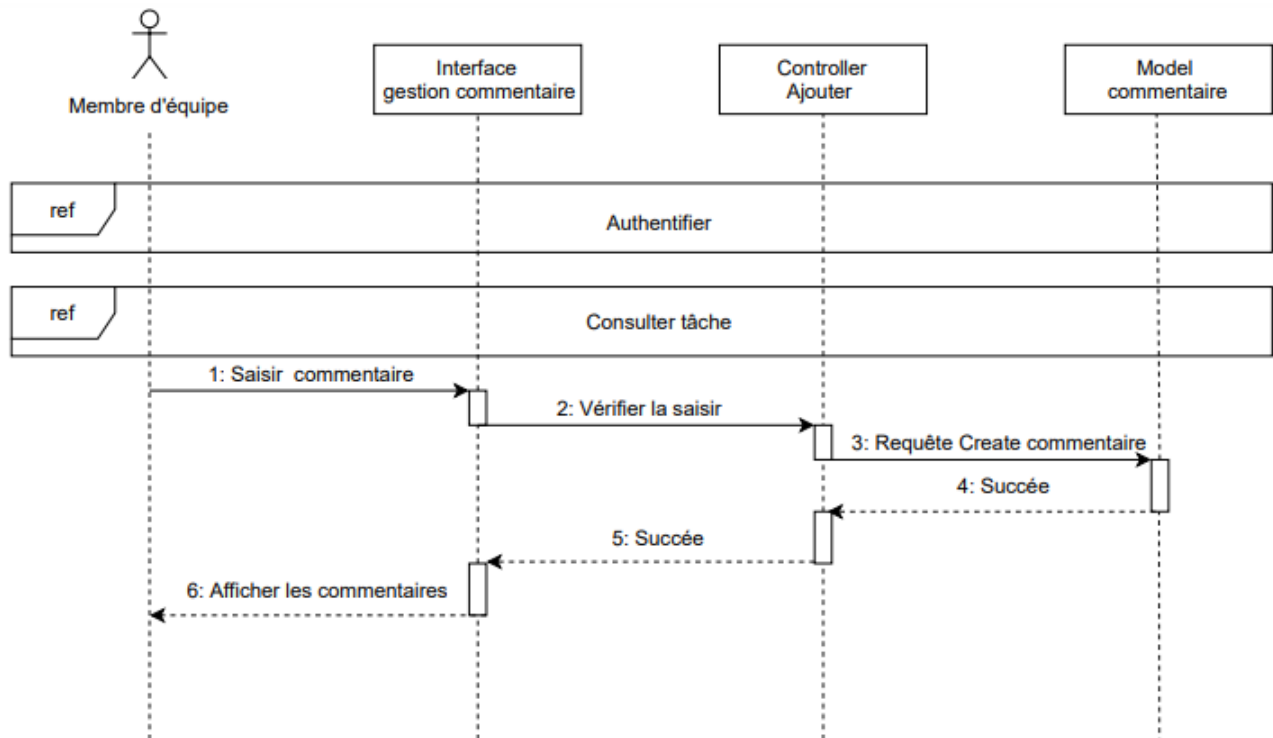


FIGURE 5.4 – Diagramme de séquence du cas d'utilisation « Ajouter commentaire »

5.3.3 Diagramme de séquence du cas d'utilisation « Supprimer commentaire »

La figure 5.5 illustre le diagramme de séquence du cas d'utilisation « Supprimer commentaire » :

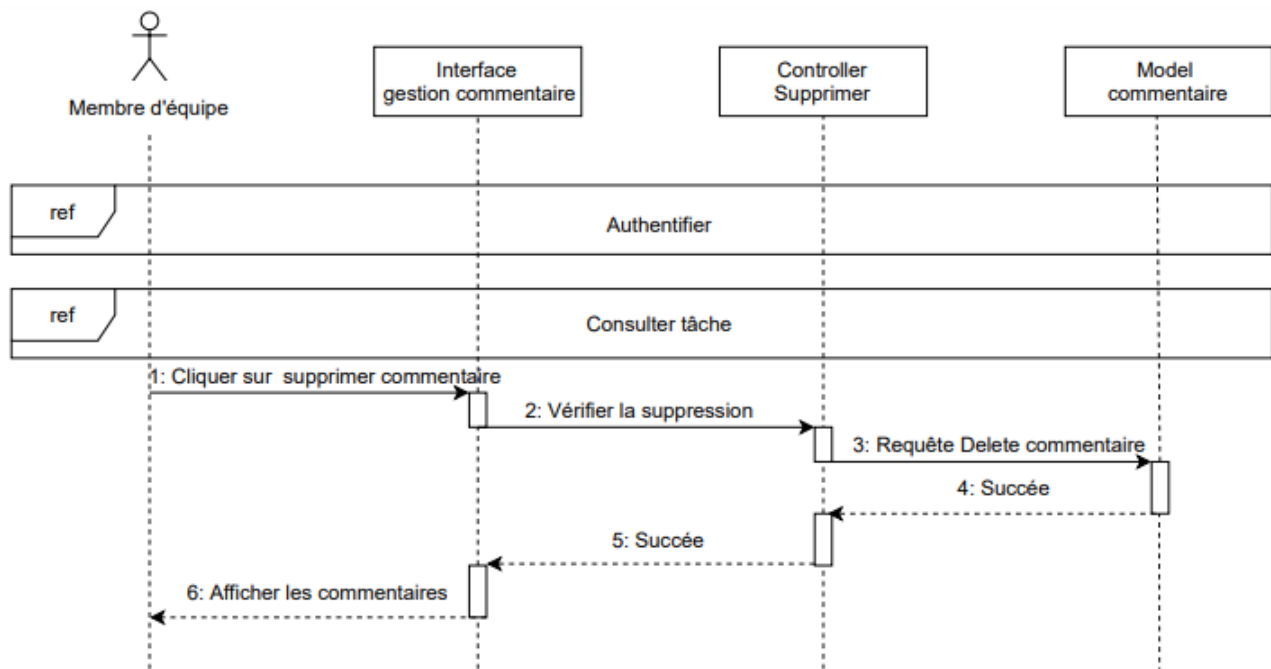


FIGURE 5.5 – Diagramme de séquence du cas d'utilisation « Supprimer commentaire »

5.3.4 Diagramme de séquence du cas d'utilisation « Ajouter Attachement »

La figure 5.6 illustre le diagramme de séquence du cas d'utilisation « Ajouter Attachement » :

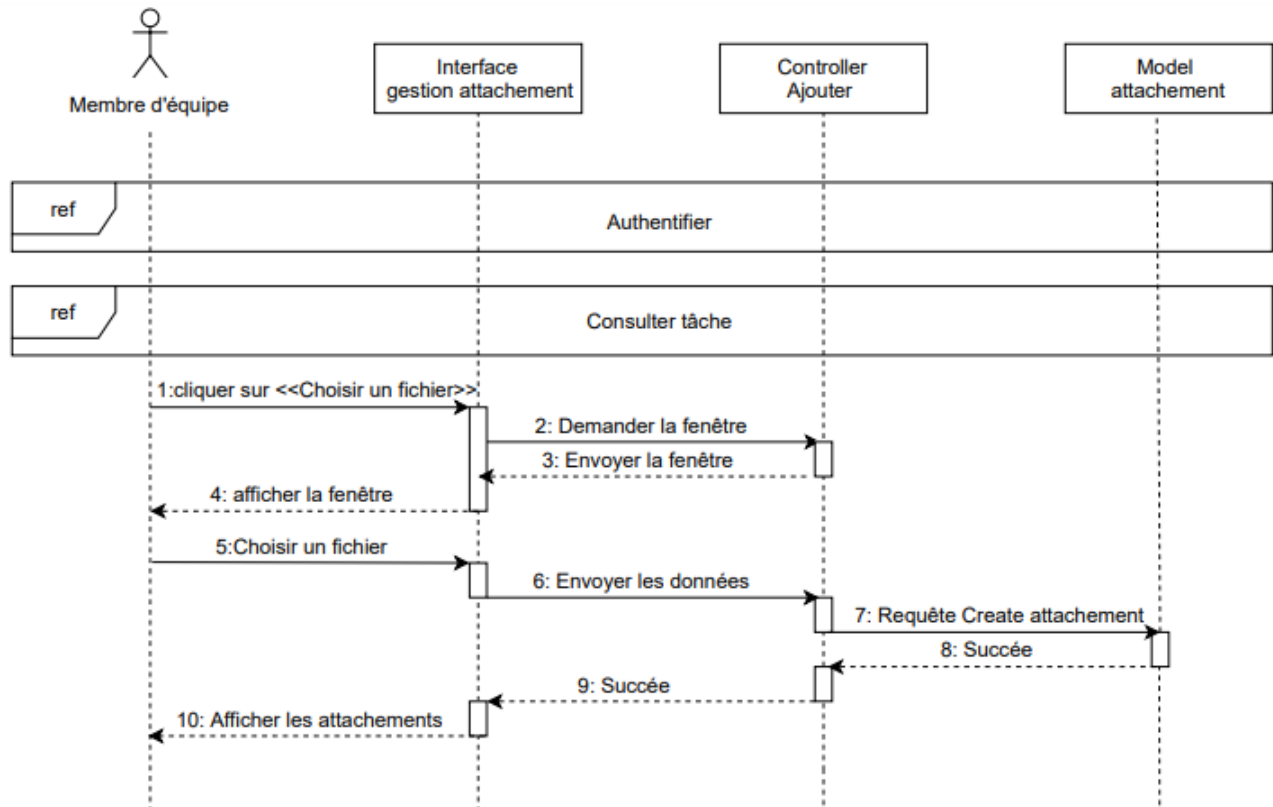


FIGURE 5.6 – Diagramme de séquence du cas d'utilisation « Ajouter Attachement »

5.4 Réalisation

Passons maintenant à la présentation de la phase de réalisation du troisième sprint. Nous exposons quelques interfaces réalisées durant ce sprint. Chaque interface illustre une fonctionnalité donnée.

5.4.1 Interface gérer commentaires

Cette interface permet à l'utilisateur d'ajouter ou supprimer un commentaire.

La figure 5.7 illustre l'interface gérer commentaires :

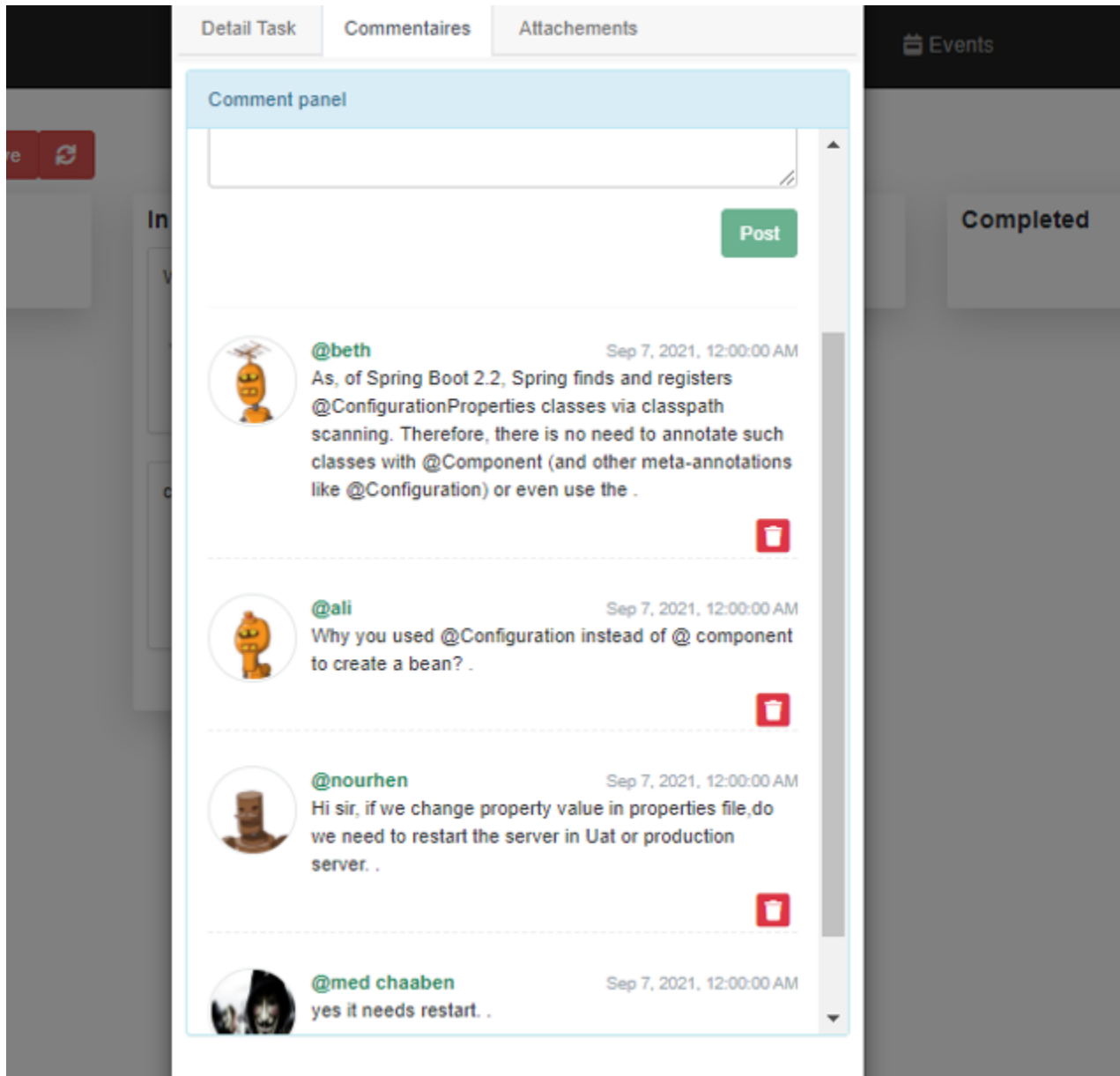


FIGURE 5.7 – Interface gérer commentaires

5.4.2 Interface gérer attachemnts

cette interface permet à l'utilisateur d'ajouter , supprimer ou télécharger un attachement liée à une tâche .

La figure 5.8 illustre l'interface gérer attachemnts :

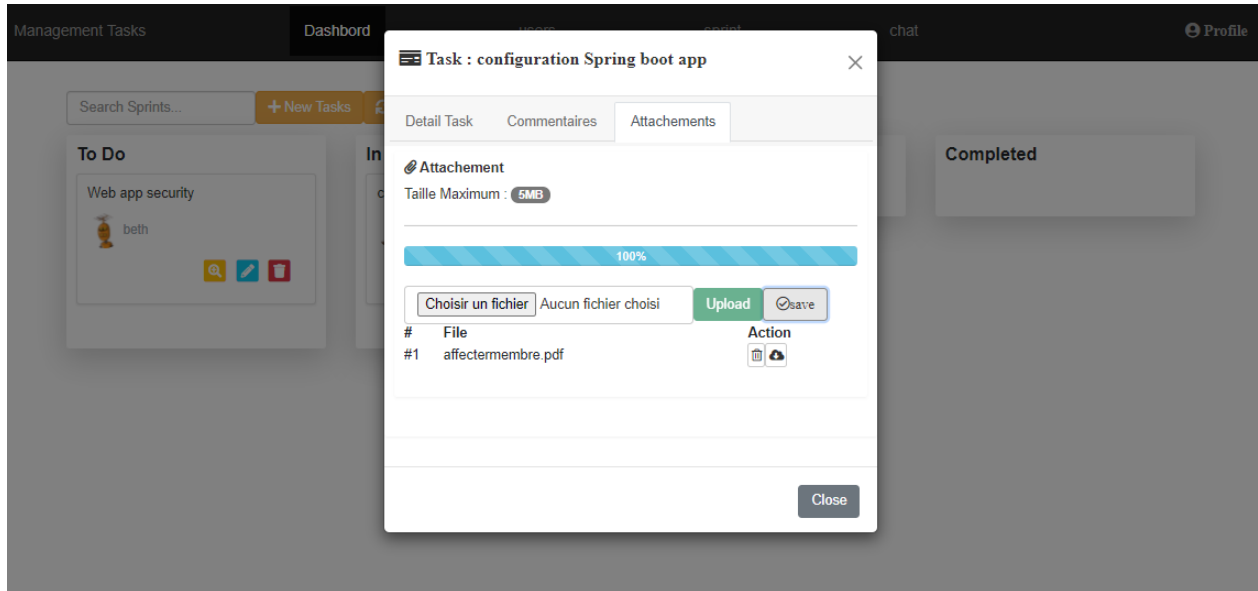


FIGURE 5.8 – Interface gérer attachemnts

Conclusion

Dans ce chapitre nous avons décrit les étapes faites durant le troisième sprint qui a été consacrée pour la gestion des commentaires et des attachements. Nous avons commencé ce chapitre par la spécification et l'analyse des exigences. Ensuite, nous avons effectué une étude conceptuelle. Enfin, nous avons clôturé par la description des étapes de réalisation. Le chapitre suivant fera l'objet d'une étude détaillée du quatrième sprint.

6

Sprint 4

Sommaire

Introduction	122
6.1 Backlog du sprint 4	122
6.2 Spécifications fonctionnelles	125
6.2.1 Raffinement de cas d'utilisation «Gérer Documents»	125
6.2.2 Raffinement de cas d'utilisation «Gérer Evènements»	126
6.3 Conception	127
6.3.1 Diagramme de classes du quaterième sprint	127
6.3.2 Diagramme de séquence du cas d'utilisation « Ajouter évènement »	129
6.4 Réalisation	129
6.4.1 Interface ajouter document	129
6.4.2 Interface liste des documents	130
6.4.3 Interface liste des documents archivés	131
6.4.4 Interface restaurer document archivé	132
6.4.5 Interface ajouter évènement	133
Conclusion	134

Introduction

Au cours de ce chapitre nous allons détailler la réalisation du dernier sprint. En appliquant le même principe que les trois chapitres précédents, nous allons parler de sprint backlog, spécifications fonctionnelles, conception et développement.

6.1 Backlog du sprint 4

Le tableau 6.1 montre notre quatrième sprint backlog :

id	Fonctionnalité	id stories	User stories	Sprint	Priorité
10	Gestion des Documents	10.1	En tant qu'un utilisateur authentifié, je veux consulter la liste des documents(Privé ,Public).	4	36
		10.2	En tant qu'un utilisateur authentifié, je veux ajouter un document.	4	35
		10.3	En tant qu'un utilisateur authentifié, je veux modifier un document.	4	38
		10.4	En tant qu'un utilisateur authentifié, je veux supprimer un document(archivé ou non archivé).	4	37
		10.5	En tant qu'un utilisateur authentifié, je veux archiver un document.	4	39

10		10.6	En tant qu'un utilisateur authentifié, je veux télécharger un document.	4	40
		10.7	En tant qu'un utilisateur authentifié, je veux afficher le détail de chaque document.	4	43
		10.8	En tant qu'un utilisateur authentifié, je veux consulter la liste des documents archivés.	4	41
		10.9	En tant qu'un utilisateur authentifié, je veux restaurer un document archivé.	4	42
11	Gestion des Evènements	11.1	En tant qu'un utilisateur authentifié, je veux consulter la liste des évènements.	4	45
		11.2	En tant qu'un utilisateur authentifié, je veux ajouter un évènement .	4	44
		11.3	En tant qu'un utilisateur authentifié, je veux modifier un évènement .	4	46

11	11.4	En tant qu'un utilisateur authentifié, je veux supprimer un événement (archivé ou non archiv�).	4	45
	11.5	En tant qu'un utilisateur authentifié, je veux archiver un événement .	4	47
	11.6	En tant qu'un utilisateur authentifié, je veux afficher le détail de chaque événement .	4	48
	11.7	En tant qu'un utilisateur authentifié, je veux consulter la liste des événements archiv�s.	4	49
	11.8	En tant qu'un utilisateur authentifié, je veux restaurer un événement archiv�.	4	50

TABLE 6.1 – Quatri me sprint Backlog

6.2 Spécifications fonctionnelles

Dans cette section nous présentons le raffinement du diagramme de cas d'utilisation pour chacune des fonctionnalités retenues dans ce sprint.

6.2.1 Raffinement de cas d'utilisation «Gérer Documents»

La figure 6.1 illustre le diagramme de cas d'utilisation « Gérer Documents » :

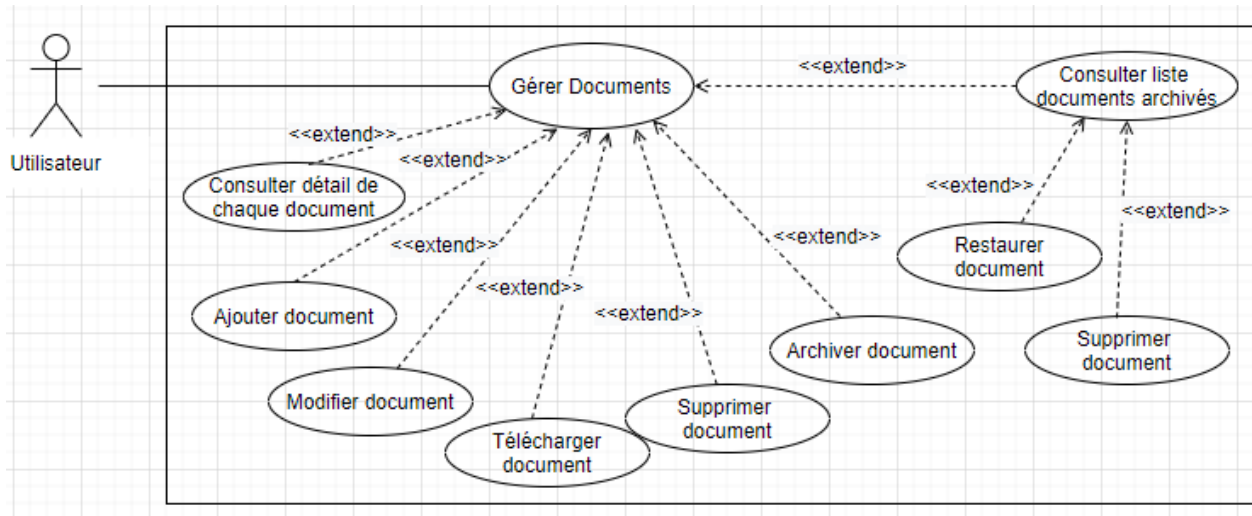


FIGURE 6.1 – Diagramme du cas d'utilisation « Gérer Documents »

Le tableau 6.2 présente une description textuelle de cas d'utilisation « Restaurer Document »

Titre	Restaurer Document
Objectif	L'acteur peut restaurer un document archivé.
Acteur	Utilisateur de l'application
Pré condition	-Acteur authentifié l'acteur a consulté la liste des documents archivés.
Post condition	un document est restauré.
Description du scénario	-L'acteur clique sur «Restaurer». -Le système restaurer le document et affiche un message de succès .

TABLE 6.2 – Description textuelle du cas d'utilisation«Restaurer Document»

6.2.2 Raffinement de cas d'utilisation «Gérer Evènements»

La figure 6.2 illustre le diagramme de cas d'utilisation « Gérer Evènements » :

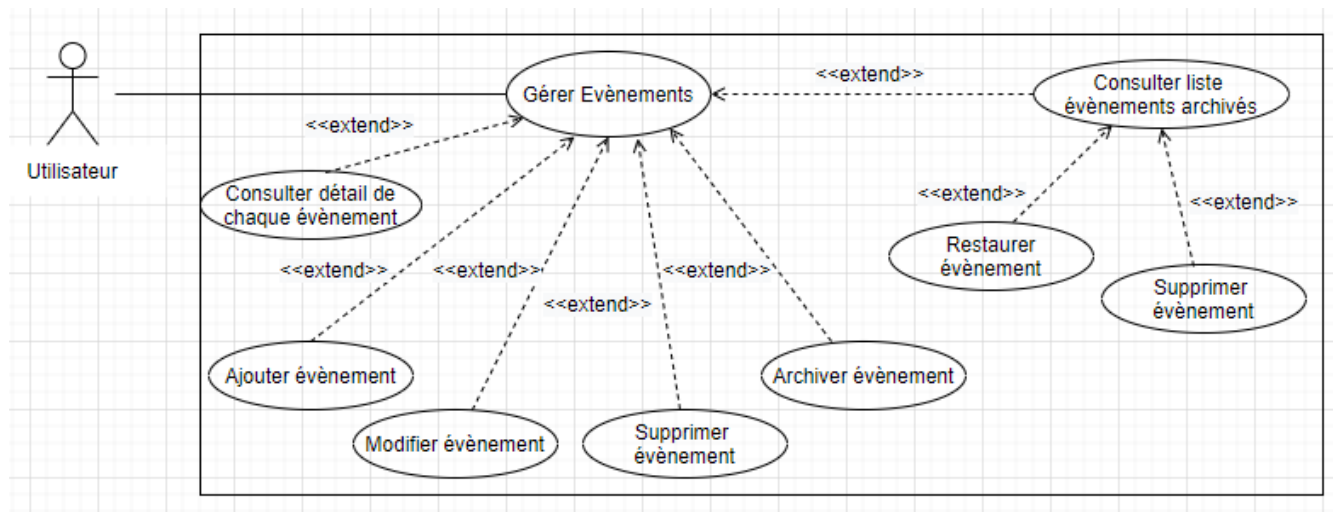


FIGURE 6.2 – Diagramme du cas d'utilisation « Gérer Evènements »

Le tableau 6.3 présente une description textuelle de cas d'utilisation « Ajouter évènement »

Titre	Ajouter Evènement
Objectif	L'acteur peut ajouter un évènement.
Acteur	Utilisateur de l'application.
Pré condition	-Acteur authentifié. -L'acteur a consulté la liste de ses évènements.
Post condition	Un évènement est ajouté.
Description du scénario	-L'acteur clique sur «Ajouter évènement». -Le système affiche un formulaire d'ajout . -L'acteur remplir le formulaire d'ajout puis cliquer sur «Ajouter». -Le système enregistre le nouveau évènement et affiche un message de succès.
Exception	-Tous les champs sont obligatoires ce pour cela il doit vérifier qu'il a bien remplir le formulaire.

TABLE 6.3 – Description textuelle du cas d'utilisation «Ajouter évènement»

6.3 Conception

6.3.1 Diagramme de classes du quaterième sprint

La figure 6.3 illustre le diagramme de classe de quaterième sprint :

6.3.2 Diagramme de séquence du cas d'utilisation « Ajouter évènement »

La figure 6.4 illustre le diagramme de séquence du cas d'utilisation « Ajouter évènement » :

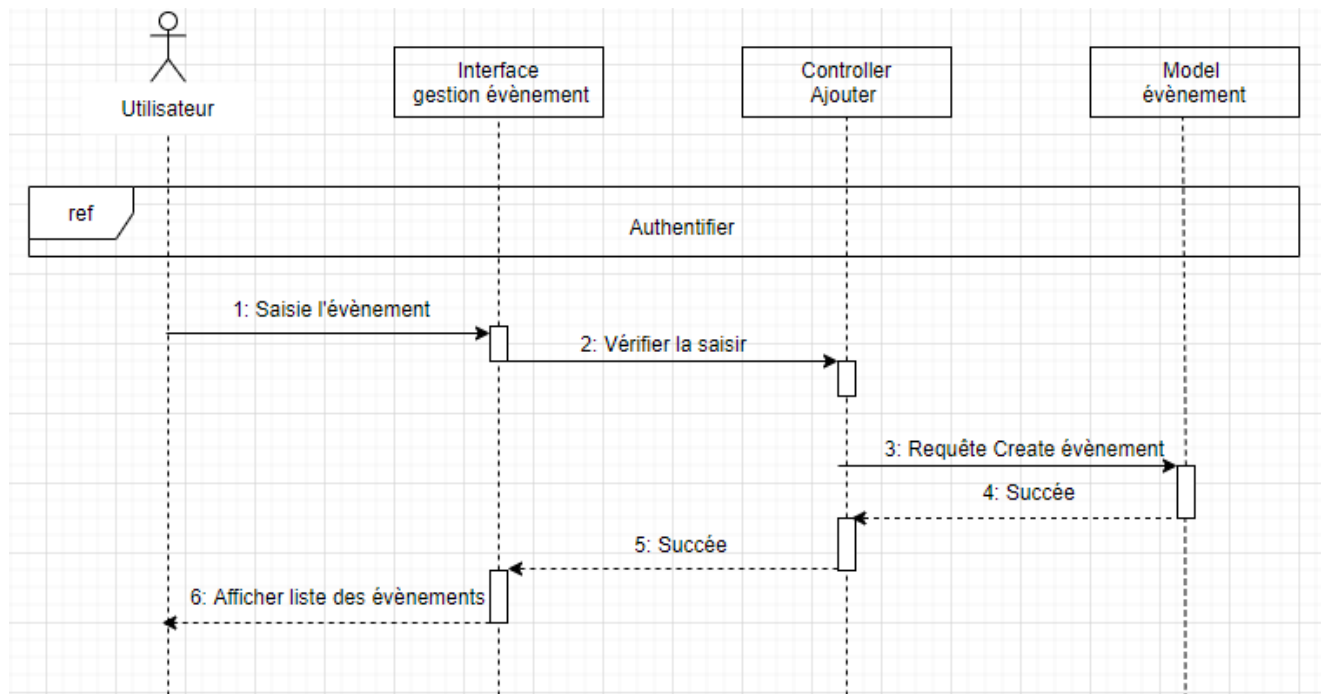


FIGURE 6.4 – Diagramme de séquence du cas d'utilisation « Ajouter évènement »

6.4 Réalisation

Passons maintenant à la présentation de la phase de réalisation du quatrième sprint. Nous exposons quelques interfaces réalisées durant ce sprint. Chaque interface illustre une fonctionnalité donnée.

6.4.1 Interface ajouter document

cette interface permet à l'utilisateur d'ajouter un document .
La figure 6.5 illustre l'interface ajouter document :

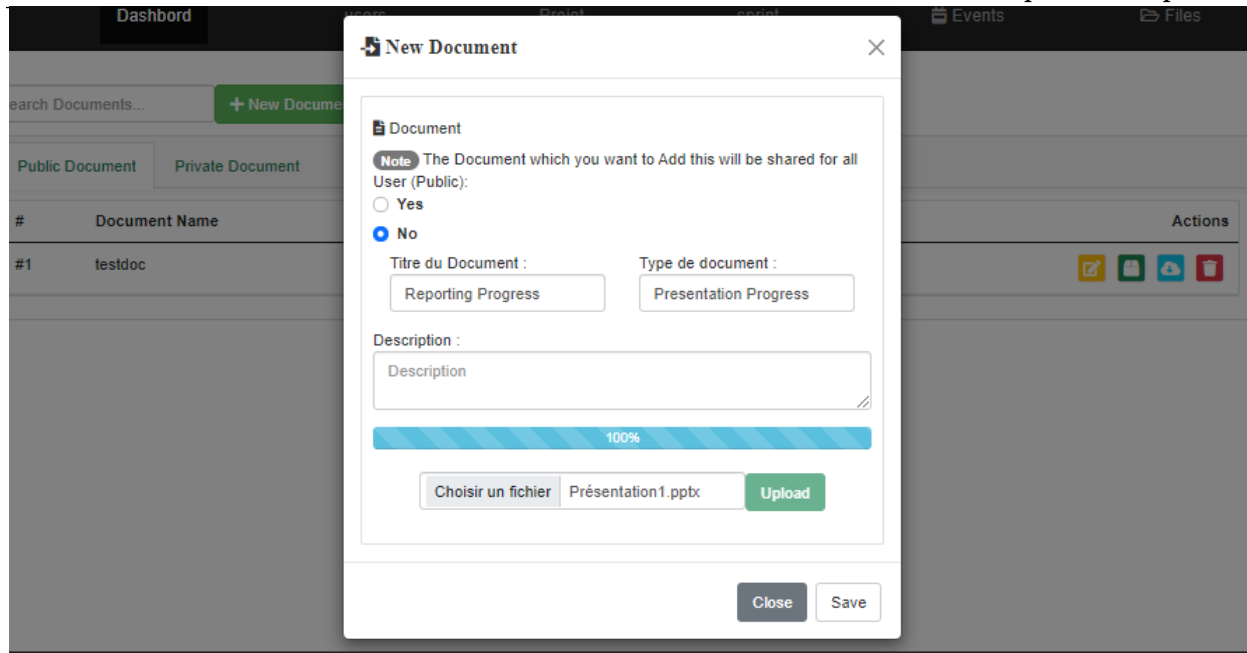


FIGURE 6.5 – Interface ajouter document

6.4.2 Interface liste des documents

cette interface permet à l'utilisateur de consulter la liste des documents(privés et publics) .

La figure 6.6 illustre l'interface liste des documents :

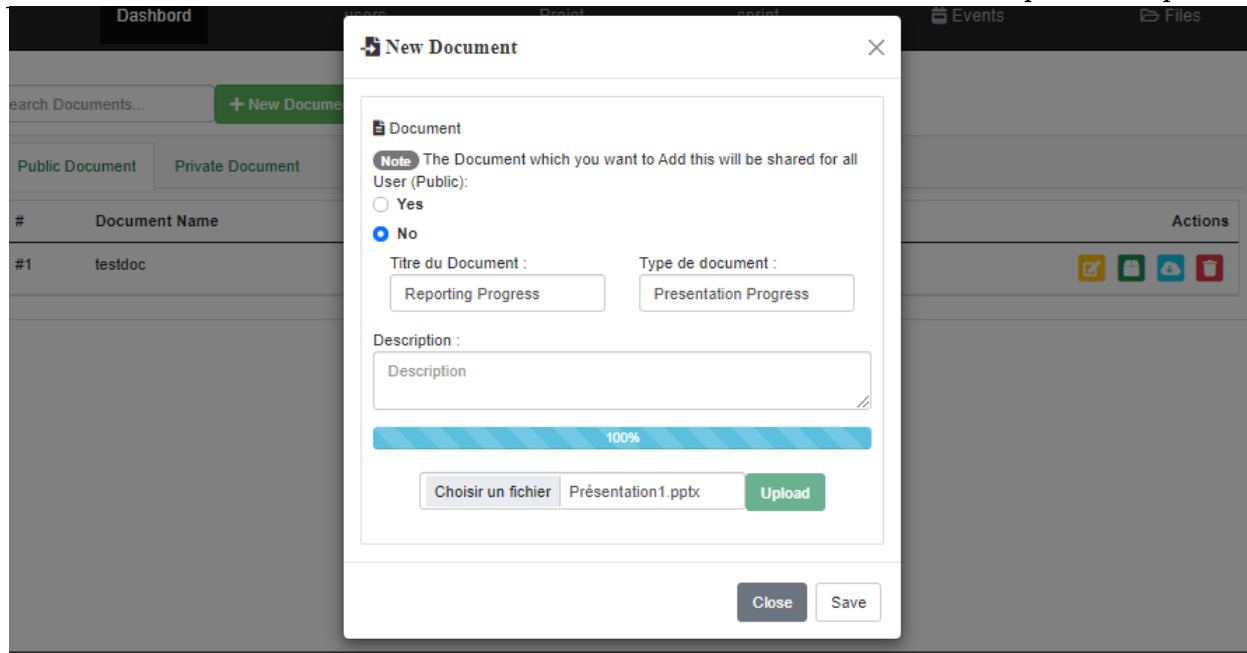


FIGURE 6.6 – Interface liste des documents

6.4.3 Interface liste des documents archivés

cette interface permet à l'utilisateur de consulter la liste des documents archivés .

La figure 6.7 illustre l'interface liste des documents archivés :

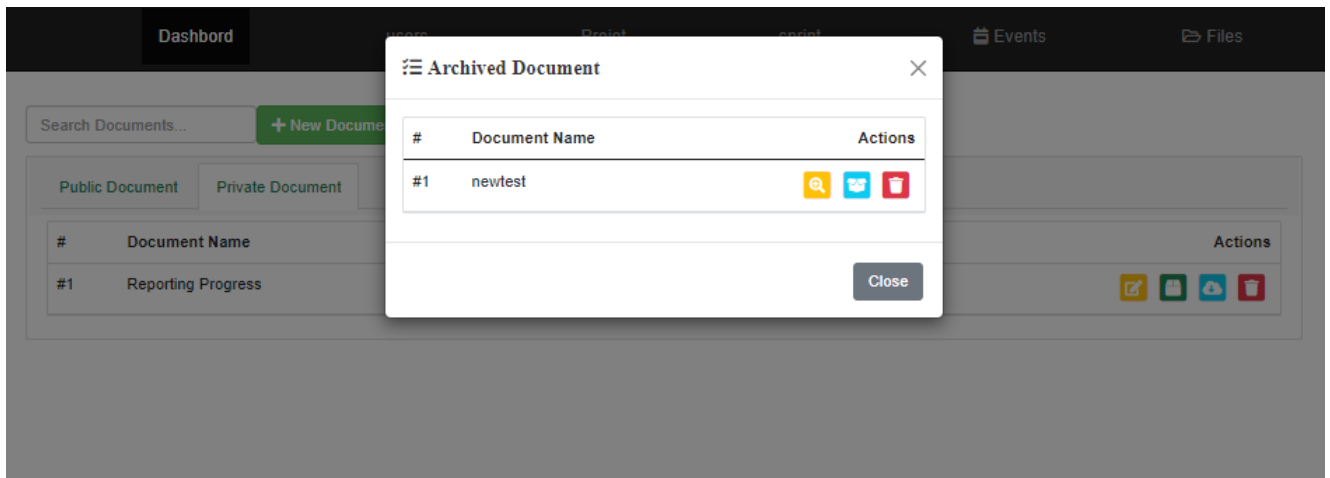


FIGURE 6.7 – Interface liste des documents archivés

6.4.4 Interface restaurer document archivé

cette interface permet à l'utilisateur de restaurer document archivé .
La figure 6.8 illustre l'interface restaurer document archivé :

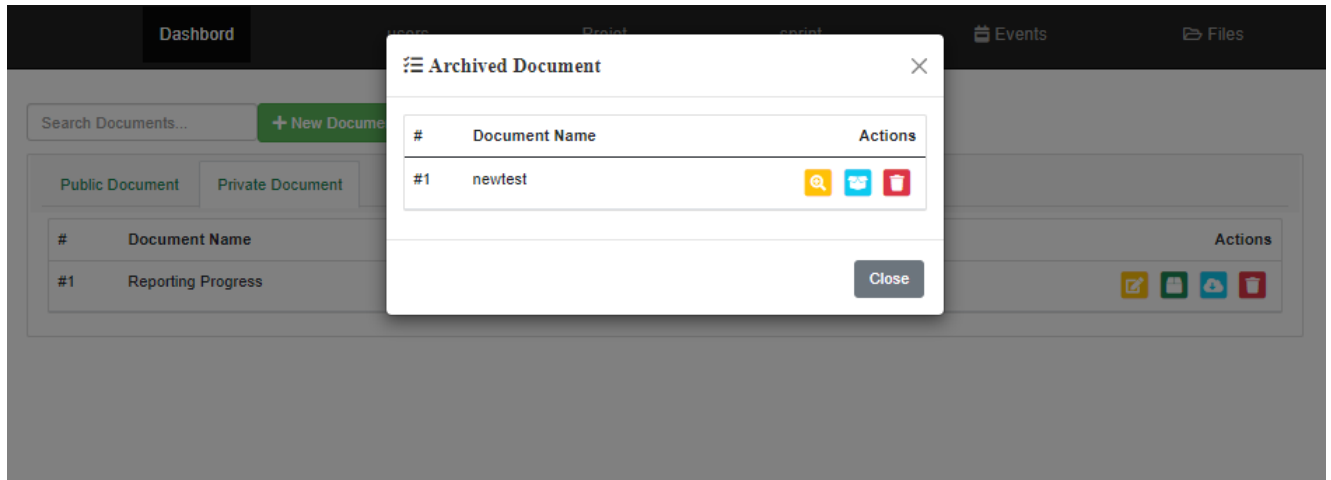


FIGURE 6.8 – Interface restaurer document archivé

6.4.5 Interface ajouter évènement

cette interface permet à l'utilisateur d'ajouter un évènement .
La figure 6.9 illustre l'interface ajouter évènement :

The screenshot shows a 'New Event' modal window. The background interface includes a sidebar with a '+ New Event' button and a top bar with 'ord' and 'Events' labels. The modal form contains the following fields and elements:

- Event Subject:** A text input field containing 'DevOps'.
- Start Date:** A date picker showing '09/09/2021'.
- End Date:** A date picker showing '09/09/2021'.
- Place:** A text input field containing 'barsha Technologie'.
- Time:** A time picker showing '14:00'.
- Categorie:** A text input field containing 'Daily SCRUM'.
- Description:** A text area containing 'Stripe testing' and 'Implementation of DataDog completed'.
- Add Member to Event:** A section with a search bar containing 'a|'. Below the search bar is a list of members with checkboxes:
 - ☒ med chaaben
 - ☐ ali
 - ☐ siwar
 - ☐ bama

At the bottom right of the modal are 'Close' and 'Save' buttons.

FIGURE 6.9 – Interface ajouter évènement

Conclusion

Durant ce chapitre, nous avons décrit les différentes tâches de ce quatrième et dernier sprint à travers le sprint backlog, puis nous avons présenté les spécifications fonctionnelles, conception et enfin nous avons fini par présenter quelques interfaces de l'application .

Conclusion Générale et Perspectives

Dans le cadre de notre projet de fin d'études ,nous avons conçu et développé une application web pour Barsha Technology . Notre projet s'est porté sur la conception et le développement d'une application web chargée de la gestion des projets Scrum.

Nous avons utilisé la méthode agile Scrum pour piloter le projet ce qui a divisé les tâches en quatre sprints comportant chacun : un sprint backlog, des spécifications fonctionnelles, une conception et une réalisation. Nous avons implémenté notre solution en utilisant la technologie Spring boot et Angular.

Malgré les contraintes de temps et les difficultés techniques que nous avons rencontré qui se résument principalement dans la compréhension du sujet et dans la complexité du projet, nous avons réussi à réaliser la totalité de la plateforme web demandé à faire.

Finalement, notre travail ne s'arrête pas à ce niveau, en effet il est possible d'ajouter des nouvelles fonctionnalités à notre application, citons comme exemple la gestion des Product Backlog ,la gestion des notifications et le chat.

Netographie

- [1] <http://www.qualitystreet.fr/2007/11/20/methodes-agiles-un-belle-definition/>.
- [2] <https://www.blogdumoderateur.com/scrum-kanban-scrumban-methode/>
Accédé le 28/05/2020.
- [3] <https://www.blogdumoderateur.com/scrum-kanban-scrumban-methode/>
Accédé le 28/05/2020/.
- [4] <https://www.blogdumoderateur.com/scrum-kanban-scrumban-methode/>
Accédé le 28/05/2020.
- [5] <https://valiantys.com/fr/blog/atlassian-administration-fr/draw-io-addon-confluence/>.
- [6] <https://www.jmdoudoux.fr/java/dej/chap-maven.htm>.
- [7] <https://fr.wikipedia.org/wiki/Npm>.
- [8] <https://www.eprojet.fr/cours/wamp/01-wamp-configuration-de-wamp>.
- [9] <https://paris-sorbonne.libguides.com/c.php?g=497641&p=4637541>.
- [10] <https://www.axopen.com/spring-boot-lyon/>.