

Dédicace

“ À l’achèvement de ce rapport de fin d’études, nous souhaitons exprimer notre reconnaissance particulière envers ceux qui ont joué un rôle central dans notre parcours :

À Dieu, nous sommes reconnaissants de nous avoir dotés de la sagesse et de la force nécessaires pour mener à bien cette étape cruciale de notre vie académique. Nous exprimons notre gratitude pour Sa guidance bienveillante tout au long de notre parcours.

Nous dédions ce rapport avec une affection profonde et une reconnaissance infinie à nos chers parents. Votre dévouement résolu, vos soins inébranlables et votre soutien incommensurable ont constitué le socle de nos réussites. Votre soutien et vos sacrifices incessants nous ont incités à atteindre et à surpasser nos aspirations. Votre exemple inspirant de courage continue de nous motiver. Nous vous considérons comme nos champions et modèles de résilience. Notre reconnaissance envers votre aide et votre gentillesse, passées et présentes, est indescriptible.

Nous exprimons une gratitude indéfectible envers notre ami, Hadi Al Manai. Les soutiens moraux que vous nous avez prodigués au cours de ce périple académique sont inestimables. Votre écoute empathique, vos encouragements constants et votre présence indéfectible ont été une source réconfortante et motivante. Nous sommes au-delà des mots de reconnaissance pour le rôle que vous avez joué comme nos piliers dans les moments difficiles. Merci d’avoir été à nos côtés.

Enfin, nous remercions chaleureusement toutes les personnes qui, de près ou de loin, ont contribué à notre formation et à la réalisation de ce rapport. Vos conseils, votre expertise et votre soutien précieux ont été inestimables. Nous exprimons notre gratitude envers tous ceux qui ont croisé notre chemin et nous ont apporté aide et encouragement.

À tous ceux mentionnés ici, ainsi qu’à tous ceux qui ont contribué de quelque manière que ce soit, nous vous témoignons une reconnaissance profonde. Que vos vies soient emplies de bonheur, de succès et de bénédictions.”

Mohamed Amin Mallek & Zammel Adnen.

Remerciements

Nous débutons nos remerciements en exprimant notre gratitude envers Dieu pour nous avoir dotés de la force, du courage et de la patience nécessaire afin de mener à bien ce travail.

Nous souhaitons adresser nos sincères remerciements à M. Bassem Lamouchi, Président Directeur Général et Fondateur de Tunisian Cloud, pour sa vision inspirante et son soutien continu. Son engagement envers l'innovation et son rôle déterminant dans le domaine ont été une source d'inspiration tout au long de notre parcours.

Un remerciement particulier est également dédié à notre encadreur à l'université, M. Hassen Ben Rebah. Nous tenons à lui exprimer notre gratitude pour avoir accepté de nous encadrer, pour ses conseils éclairés, sa disponibilité et son suivi attentif de notre progression. Son expertise a grandement contribué à la réussite de notre travail dans des conditions optimales.

Nous ne pouvons passer sous silence notre reconnaissance envers tous les membres du jury qui ont apporté leurs précieuses contributions à l'évaluation de ce projet. Leur examen critique a enrichi notre travail et nous sommes honorés de leur implication.

Enfin, nous tenons à remercier toutes les personnes, de près ou de loin, qui ont contribué à la réalisation de ce travail. Leur soutien a été inestimable et a contribué à la réussite de notre projet.

Nos plus sincères remerciements à tous ceux mentionnés ici, ainsi qu'à ceux qui ont joué un rôle, même minime, dans cette réalisation.

Table des matières

Introduction générale.....	11
Chapitre 1: Etude préalable.....	13
1.1 Introduction.....	14
1.2 Cadre général du projet.....	14
1.2.1 Cadre du projet	14
1.2.2 Objectifs du projet.....	14
1.3 Présentation de l'organisme d'accueil	15
1.3.1 Présentation générale.....	15
1.3.2 Services	16
1.3.4 Coordonnées géographiques	16
1.3.5 Structure organisationnelle.....	17
1.4 Analyse et critique de l'existant	17
1.4.1 La plateforme Later	18
1.4.2 La plateforme Fanpagekarma.....	18
1.4.3 La plateforme Tailwind Publish.....	19
1.4.4 La plateforme Buffer.....	20
1.5 Solution proposée	20
1.6 Langage et Méthodologie de conception	22
1.6.1 Choix de la méthodologie de conception	22
1.6.2 Choix du langage de modélisation	28
1.7 Conclusion.....	28
Chapitre 2: Sprint 0 : Planification du projet	29
2.1 Introduction.....	30
2.2 Identification des acteurs du système.....	30
2.3 Identification des besoins	30

2.3.1.	Besoins fonctionnels	30
2.3.2.	Besoins non fonctionnels	31
2.4	Diagramme des cas d'utilisation global	33
2.5	Diagramme de classes global	34
2.6	Structure et découpage du projet	36
2.6.1.	Identification de l'équipe SCRUM	36
2.6.2.	Backlog du produit	36
2.6.3.	Planification des sprints	39
2.7.	Architecture et environnement de développement	40
2.7.1.	Architecture de l'application SocialScope	40
2.7.2.	Environnement de développement	42
2.8.	Conclusion.....	44
Chapitre 3 : Sprint 1 : Fonctionnalités de base de l'administration et de la gestion des utilisateurs.....		45
3.1	Introduction	46
3.2	Objectifs du sprint.....	46
3.3	Backlog du sprint	46
3.4.1	Diagramme des cas d'utilisation	47
3.4.2	Description textuelle des cas d'utilisation.....	48
3.4.3	Diagrammes de séquence	51
3.4.4	Diagramme de classes	53
3.4	Réalisation.....	54
3.4.1	Modélisation des maquettes graphiques.....	54
3.4.2	Développement des interfaces utilisateurs	56
3.4.3	Déploiement des composants logiciels	57
3.5	Test fonctionnel	58
3.6	Conclusion.....	59

Chapitre 4 :Sprint 2: Fonctionnalités de gestion des campagnes et des tâches associées 60

4.1	Introduction	61
4.2	Objectifs du sprint.....	61
4.3	Backlog du sprint	61
4.4	Conception du sprint.....	62
4.4.1	Diagramme des cas d'utilisation	62
4.4.2	Description textuelle des cas d'utilisation.....	63
4.4.3	Diagrammes de séquence	66
4.4.4	Diagramme de classes	68
4.5	Réalisation.....	70
4.5.1	Modélisation des maquettes graphiques.....	70
4.5.2	Développement des interfaces utilisateurs	71
4.6	Test fonctionnel	73
4.7	Conclusion.....	75

Chapitre 5 :Sprint 3: Fonctionnalités de Gestion des publications et de reporting..... 75

5.1	Introduction	77
5.2	Objectifs du sprint.....	77
5.3	Backlog du sprint	77
5.4	Conception du sprint.....	78
5.4.1	Diagramme des cas d'utilisation	78
5.4.2	Description textuelle des cas d'utilisation.....	78
5.4.3	Diagrammes de séquence	81
5.4.4	Diagramme de classes	84
5.5	Réalisation.....	86
5.5.1	Modélisation des maquettes	86
5.5.2	Développement des interfaces utilisateurs	89
5.5.3	Déploiement des composants logiciels	91

5.6 Test fonctionnel	92
5.7 Conclusion.....	93
Chapitre 6: Sprint 4 : Fonctionnalités de collaboration, de Gestion des rôles et des permissions.....	94
6.1 Introduction.....	95
6.2 Objectifs du sprint.....	95
6.3 Backlog du sprint	95
6.4 Conception du sprint.....	96
6.4.1 Diagramme des cas d'utilisation	96
6.4.2 Description textuelle des cas d'utilisation.....	96
6.4.3 Diagrammes de séquence	99
6.4.4 Diagramme de classes	100
6.5 Réalisation.....	102
6.5.1 Modélisation des maquettes	102
6.5.2 Développement des interfaces utilisateurs	103
6.6 Test fonctionnel	104
6.7 Conclusion.....	105
Conclusion Générale	106
Bibliographie.....	107

Liste des Figure

Figure 1- Fonctionnalités d'un système de gestion des médias sociaux	15
Figure 2- Logo Tunisian Cloud	15
Figure 3- Plan d'orientation Tunisian Cloud (Tunis.....	16
Figure 4- Organigramme du Tunisian Cloud.....	17
Figure 5 –Processus Scrum.....	24
Figure 6 – Les intervenant dans SCRUM	25
Figure 7 – Les artéfacts dans SCRUM	26
Figure 8- Diagramme des cas d'utilisation global	34
Figure 9- Diagramme de classes global	35
Figure 10- Répartition des sprints du projet	40
Figure 11- Architecture Client-serveur.....	42
Figure 12- Diagramme des cas d'utilisation du premier sprint	47
Figure 13- Diagramme de séquence « S'authentifier ».....	51
Figure 14- Diagramme de séquence « Modifier profil »	52
Figure 15- Diagramme de séquence « Ajouter utilisateur ».....	53
Figure 16-Diagramme de classes du Sprint 1	53
Figure 17- Prototype relatif à l'histoire utilisateur « S'authentification ».....	54
Figure 18- Prototype relatif à l'histoire utilisateur « Modifier profil »	55
Figure 19- Prototype relatif à l'histoire utilisateur « Ajouter utilisateur »	55
Figure 20- Interface graphique relative à l'histoire utilisateur « S'authentification »....	56
Figure 21- Interface graphique relative à l'histoire utilisateur « Modifier profil »	56
Figure 22- Interface graphique relative à l'histoire utilisateur « Ajouter utilisateur »... 	57
Figure 23- Architecture basique de l'application	58

Figure 24- Diagramme des cas d'utilisation du deuxième sprint.....	62
Figure 25- Diagramme de séquence « Créer campagne»	66
Figure 26- Diagramme de séquence «Créer tâche ».....	67
Figure 27- Diagramme de séquence «Modifier tâche».....	68
Figure 28- Diagramme de classes du Sprint 2	69
Figure 29- Prototype relatif à l'histoire utilisateur « Créer campagne »	70
Figure 30- Prototype relatif à l 'histoire utilisateur « Créer tâche »	71
Figure 31- Prototype relatif à l 'histoire utilisateur « Modifier tâche »	71
Figure 32– Interface graphique relatif à l 'histoire utilisateur « Créer campagne »	72
Figure 33- Interface graphique relatif à l 'histoire utilisateur « Créer tâche »	72
Figure 34- Interface graphique relatif à l 'histoire utilisateur de « Modifier tâche ».....	73
Figure 35- Diagramme des cas d'utilisation du troisième sprint	78
Figure 36- Diagramme de séquence « Se connecter à un compte de réseau social ».....	82
Figure 37- Diagramme de séquence « Créer publication »	83
Figure 38- Diagramme de séquence «Modifier publication»	84
Figure 39- Diagramme de classes du Sprint 3	85
Figure 40- Prototype relatif à l'histoire utilisateur « Se connecter à un compte de réseau social »	86
Figure 41- Prototype relatif à l'histoire utilisateur « Créer publication »	87
Figure 42- Prototype relatif à l'histoire utilisateur « Modifier publication ».....	87
Figure 43- Prototype relatif à l'histoire utilisateur « Consulter les rapports et les statistiques ».....	88
Figure 44- Interface graphique relative à l 'histoire utilisateur « Se connecter à un compte de réseau social »	89
Figure 45- Interface graphique relative à l 'histoire utilisateur « Créer publication » ...	89

Figure 46- Interface graphique relative à l 'histoire utilisateur « Modifier publication »	90
Figure 47- Interface graphique relative à l 'histoire utilisateur « Consulter les rapports et les statistiques »	91
Figure 48- Architecture principale de l'application	92
Figure 49- Diagramme des cas d'utilisation du quatrième sprint	96
Figure 50- Diagramme de séquence du cas d'utilisation «Ajouter Utilisateur à une campagne »	99
Figure 51- Diagramme de séquence du cas d'utilisation « Crée rôle »	100
Figure 52- Diagramme de classes du Sprint 4	101
Figure 53- Prototype relatif à l'histoire utilisateur « Collaborer avec d'autres utilisateurs »	102
Figure 54- Prototype relatif à l'histoire utilisateur « Crée un nouvel rôle utilisateur »	103
Figure 55- Interface graphique relative à l 'histoire utilisateur « Collaborer avec d'autres utilisateurs »	103
Figure 56- Interface graphique relative à l 'histoire utilisateur « Crée un nouvel rôle utilisateur »	104

Liste des Tableaux

Tableau 1 - Les fonctionnalités de la plateforme Later	18
Tableau 2 - Les fonctionnalités de la plateforme Fanpagekarma.....	19
Tableau 3 - Les fonctionnalités de la plateforme Tailwind Publish	19
Tableau 4 - Les fonctionnalités de la plateforme Buffer.....	20
Tableau 5- Avantages différenciés : SocialScope face aux alternatives concurrentes	21
Tableau 6- Équipe du Projet SocialScope	36
Tableau 7- Backlog du produit	39
Tableau 8- Spécifications des environnements de développement	44
Tableau 9- Backlog du sprint 1	47
Tableau 10- Description textuelle du cas d'utilisation « S'authentifier »	48
Tableau 11- Description textuelle du cas d'utilisation « Modifier profil »	49
Tableau 12- Description textuelle du cas d'utilisation « ajouter utilisateur »	50
Tableau 13- Backlog du sprint 2	62
Tableau 14- Description textuelle du cas d'utilisation « Créer campagne ».....	63
Tableau 15- Description textuelle du cas d'utilisation « Créer tâche ».....	64
Tableau 16- Description textuelle du cas d'utilisation « Modifier tâche ».....	65
Tableau 17- Backlog du sprint 3	78
Tableau 18- Description textuelle du cas d'utilisation « Se connecter à un compte de réseau social »	79
Tableau 19- Description textuelle du cas d'utilisation « Créer publication ».....	80
Tableau 20- Description textuelle du cas d'utilisation « Modifier publication »	81
Tableau 21- Backlog du sprint 4	96
Tableau 22- Description textuelle du cas d'utilisation « Ajouter Utilisateur à une campagne ».....	97
Tableau 23- Description textuelle du cas d'utilisation « Créer rôle »	98

Introduction générale

L

E paysage actuel des réseaux sociaux est en constante évolution. Avec plus de 4,88 milliards d'utilisateurs¹ actifs dans le monde, les plateformes sociales sont devenues un élément crucial de la communication et du marketing pour les entreprises et les organisations de toutes tailles.

Cependant, ce paysage dynamique présente également de nombreux défis. La concurrence accrue, l'évolution des algorithmes et la nécessité de créer du contenu engageant peuvent submerger les entreprises qui cherchent à maximiser leur présence sur les réseaux sociaux.

Dans ce contexte, l'analyse des réseaux sociaux s'avère un outil indispensable pour la prise de décision stratégique. En collectant et en analysant des données clés, les entreprises peuvent obtenir une compréhension approfondie de leur audience, de leurs performances et de l'impact de leurs actions.

Dans ce contexte, l'entreprise Tunisian Cloud nous a confié la mission de développer une application Web appelée "SocialScope". Son objectif est de révolutionner la façon dont les entreprises et les organisations gèrent et coordonnent leurs publications sur diverses plateformes de réseaux sociaux telles que Facebook, Instagram, Twitter. Cette application Web sera conçue pour être à la fois intuitive et performante, offrant une gamme complète de fonctionnalités, notamment la gestion des autorisations et des utilisateurs, la création et la gestion de profils d'entreprise, la planification et l'exécution de campagnes, la publication et le suivi de contenu, ainsi que la facilitation de la communication et de la collaboration. De plus, elle fournira des notifications pour les événements clés, assurant ainsi une gestion proactive et efficace des activités sur les réseaux sociaux.

Ce rapport présente six chapitres qui suivent cette structure : Le premier chapitre, "Étude préalable", établit les bases du projet en décrivant son contexte, ses objectifs et l'organisation hôte. Nous y conduisons une analyse approfondie de la situation actuelle, incluant sa description et une critique constructive, tout en exposant notre solution et les méthodologies choisies. Le deuxième chapitre, "Sprint 0 : Planification du projet", se focalise sur le démarrage initial du

¹ <https://www.blogdumoderateur.com/chiffres-reseaux-sociaux/>

projet, en identifiant les parties prenantes, les besoins fonctionnels et non fonctionnels du système, en planifiant les sprints et en configurant l'environnement de développement. Les chapitres trois à six détaillent les différents sprints de développement de La solution "SocialScope", suivant une structure similaire. Chaque chapitre commence par les objectifs du sprint, puis présente le backlog, la conception détaillée des fonctionnalités, leur implémentation pratique et les tests fonctionnels. Enfin, le rapport se conclut par une synthèse générale.

Chapitre 1

Etude préalable



1.1 Introduction

Dans ce chapitre intitulé "Étude préalable", nous commençons par définir le cadre général du projet ainsi que ses objectifs. Ensuite, nous présentons l'entreprise où le stage de fin d'études a été effectué. Après cela, une analyse de l'existant est réalisée, suivie de la proposition de notre solution. Enfin, le choix du langage et de la méthodologie de conception est abordé.

1.2 Cadre général du projet

Dans cette section, l'accent est mis sur la présentation du contexte de notre projet de fin d'études ainsi que sur ses objectifs principaux.

1.2.1 Cadre du projet

Ce projet est réalisé dans le cadre d'un stage de fin d'études en vue de l'obtention du diplôme national de licence appliquée en technologies de l'informatique, avec une spécialité en "Développement des Systèmes d'Information". Son objectif principal est de compléter notre parcours universitaire, d'approfondir les compétences acquises au cours des trois années de formation à l'Institut Supérieur des Études Technologiques de Sfax, et de nous préparer à notre future intégration dans le monde professionnel.

1.2.2 Objectifs du projet

Ce projet vise à concevoir et développer une application web visant à révolutionner l'approche des entreprises dans la gestion de leurs publications sur les réseaux sociaux, incluant Facebook, Instagram, Twitter. Cette application offrira aux utilisateurs la possibilité de gérer efficacement les profils, les entreprises, les campagnes, les publications, la communication et les notifications, tout en favorisant une collaboration optimale. Pour ce faire, notre application s'engage à plusieurs niveaux : premièrement, elle simplifiera le processus de publication en permettant aux utilisateurs de programmer aisément leurs contenus sur différentes plateformes de réseaux sociaux. Ensuite, elle cherchera à maximiser la visibilité des publications en aidant les utilisateurs à cibler un public plus large, en tenant compte des moments les plus opportuns pour publier en fonction des habitudes et des comportements des utilisateurs sur les réseaux sociaux. Elle visera également à améliorer l'engagement avec les followers en favorisant les interactions, les conversations et l'engagement sur les réseaux sociaux. Enfin, notre application s'efforcera de répondre aux besoins spécifiques des utilisateurs en comprenant leurs exigences particulières, qu'ils soient des entreprises, des influenceurs ou des particuliers. Nous nous engageons à fournir des fonctionnalités personnalisées qui aideront chaque acteur à atteindre ses objectifs spécifiques sur les réseaux sociaux.



Figure 1- Fonctionnalités d'un système de gestion des médias sociaux

1.3 Présentation de l'organisme d'accueil

Cette section a deux objectifs principaux : d'une part, présenter de manière générale l'entreprise, et d'autre part, fournir ses coordonnées ainsi que sa structure organisationnelle.

1.3.1 Présentation générale

Fondée en 2014, Tunisian Cloud est une entreprise spécialisée dans l'ingénierie des technologies de l'information et la formation professionnelle. Elle se démarque par son expertise en Cloud Computing et en Sécurité, abordant ainsi les divers défis liés aux systèmes d'information : de la création à la sécurisation en passant par le transport, le stockage, le traitement, l'utilisation et l'analyse des données. Son approche se distingue par la proposition de solutions complètes, répondant aux attentes et aux besoins actuels du marché.

En tant que centre de formation agréé par le ministère de la Formation Professionnelle et de l'Emploi Tunisien, Tunisian Cloud est enregistré sous le numéro 33-042-15. Son équipe se compose d'experts nationaux et internationaux certifiés, jouissant d'une reconnaissance en tant que formateurs agréés par les principaux éditeurs et constructeurs de certifications dans le domaine de l'IT.



Figure 2- Logo Tunisian Cloud

1.3.2 Services

Tunisian Cloud offre une gamme variée de services à ses clients :

- **Service de Conseils en Systèmes d'Information** : analyse approfondie des systèmes d'information des entreprises pour identifier les lacunes et les faiblesses. Élaboration d'un plan d'action et mise en œuvre de solutions technologiques avancées.
- **Service Cloud Computing** : accompagnement des clients dans leur migration vers les modèles et services Cloud les plus adaptés. Intervention en tant que conseiller Cloud, créateur Cloud, fournisseur d'applications Cloud et auditeur Cloud.
- **Service Tests d'Intrusion** : proposition de tests d'intrusion pour évaluer la sécurité des infrastructures informatiques des clients en simulant des attaques.
- **Service Infogérance et Support** : adaptation des services d'infogérance et de support aux besoins et à la taille des clients, leur permettant de confier tout ou partie de la gestion de leur système d'information à un prestataire externe spécialisé.
- **Service Intégration** : possibilité d'intégrer des solutions informatiques dans les locaux des clients à savoir :
 - Mise en place de Réseaux IP, câblage, routeurs, switch
 - Mise en place et configuration de serveurs (Windows et Linux),
 - Mise en place de système de stockage local ou distant
 - Mise en place de système de pointage et gestion intégré des entreprises
 - Administration à distance et dépannage informatique ...
 - Etc...

1.3.4 Coordonnées géographiques

Tunisian Cloud est une société basée à Tunis, ses coordonnées sont :

- Adresse : Immeuble Chammam, rue du Lac Toba, Les Berges du Lac 1053, Tunis, Tunisie
- Email : contact@tunisiancloud.com
- Tel : (00216) 70 279 470

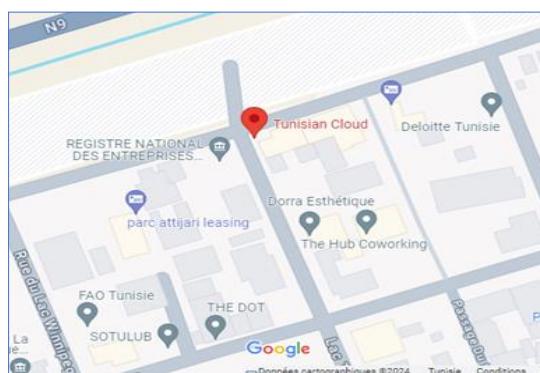


Figure 3- Plan d'orientation Tunisian Cloud (Tunis)

1.3.5 Structure organisationnelle

La structure organisationnelle d'une entreprise représente le type d'organisation entre ses différentes unités qui structurent l'entreprise et le choix d'organisation des ressources humaines et matériels mis en œuvre entre ses différentes unités. Depuis 2014, Tunisian Cloud est structurée comme suit :

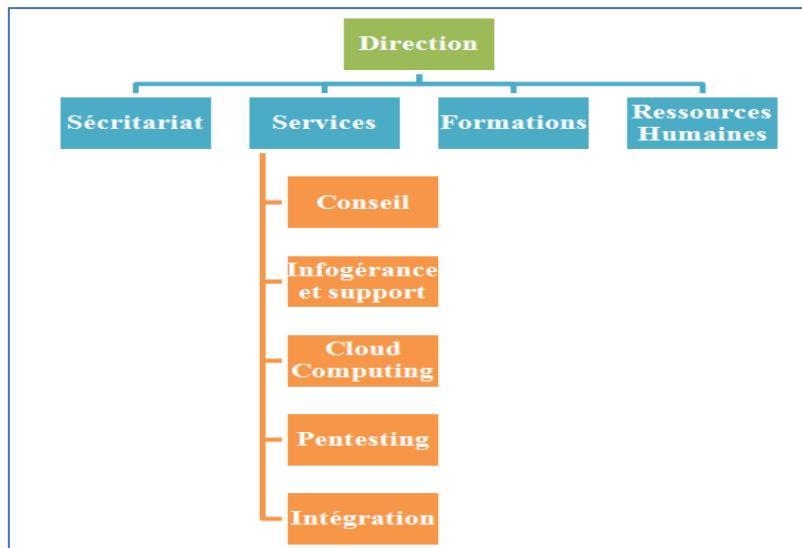


Figure 4- Organigramme du Tunisian Cloud

L'équipe de Tunisian Cloud est caractérisée par un esprit d'équipe fort et collaboratif. Chaque membre apporte ses compétences et son expertise, créant ainsi une synergie qui se traduit par des résultats exceptionnels. Chez Tunisian Cloud, l'esprit d'équipe se manifeste à travers une culture de soutien mutuel, de respect et de confiance. Les membres travaillent en étroite collaboration pour relever les défis et atteindre les objectifs fixés. Ils partagent leurs connaissances et leurs idées, favorisant ainsi un apprentissage continu et une amélioration collective.

Sous la vision et le leadership de M. Bassem Lamouchi, Tunisian Cloud continue de se développer et de fournir des services informatiques de premier ordre à ses clients. Son engagement envers l'excellence et l'innovation en fait un partenaire de confiance pour les entreprises recherchant des solutions informatiques de qualité.

1.4 Analyse et critique de l'existant

Plusieurs systèmes de gestion de publications sur les réseaux sociaux ont été développés, chacun offrant un ensemble de fonctionnalités répondant aux besoins des utilisateurs. Dans la suite de cette section, nous proposons une analyse des avantages et des inconvénients de certaines de ces applications.

1.4.1 La plateforme Later

Later est une plateforme de planification et de gestion des médias sociaux qui se distingue par son interface intuitive et conviviale. Outil puissant pour les utilisateurs individuels comme pour les équipes, Later facilite la planification et la publication de contenu sur une multitude de plateformes sociales.

URL : https://later.com	
Analyse fonctionnelle	<ul style="list-style-type: none"> - Gestion complète des utilisateurs et des permissions. - Création et gestion de publications.
Points forts	<ul style="list-style-type: none"> - Interface utilisateur intuitive et conviviale. - Planification avancée des publications.
Points faibles	<ul style="list-style-type: none"> - Communication et collaboration en temps réel non disponibles.

Tableau 1 - Les fonctionnalités de la plateforme Later

1.4.2 La plateforme Fanpagekarma

Fanpagekarma va au-delà d'être simplement une plateforme d'analyse. C'est un outil de pointe qui permet de suivre en détail les statistiques des pages et des publications sur les réseaux sociaux.

URL: https://www.fanpagekarma.com	
Analyse fonctionnelle	<ul style="list-style-type: none"> - Interface utilisateur intuitive et conviviale. - Gestion complète des utilisateurs et des permissions.

URL: https://www.fanpagekarma.com		
Points forts	<ul style="list-style-type: none"> - Analyse avancée des performances sur les réseaux sociaux. - Suivi détaillé des statistiques des pages et des publications. 	
Points faibles	<ul style="list-style-type: none"> - Création et gestion de publications non disponibles. 	

Tableau 2 - Les fonctionnalités de la plateforme Fanpagekarma

1.4.3 La plateforme Tailwind Publish

Tailwind Publish est une solution avancée de planification des publications sur Pinterest et Instagram, offrant une analyse approfondie des performances des épingles et des publications sur Instagram.

URL : https://www.tailwindapp.com		
Analyse fonctionnelle	<ul style="list-style-type: none"> - Création et gestion d'entreprises et de hashtags. - Définition et planification de Campagnes. 	
Points forts	<ul style="list-style-type: none"> - Planification avancée des publications sur Pinterest et Instagram. - Analyse approfondie des performances des épingles et des publications sur Instagram. 	
Points faibles	<ul style="list-style-type: none"> - Communication et collaboration en temps réel non disponibles. 	

Tableau 3 - Les fonctionnalités de la plateforme Tailwind Publish

1.4.4 La plateforme Buffer

Buffer est une plateforme de gestion des médias sociaux qui propose une interface intuitive et conviviale. Elle permet une planification avancée des publications sur plusieurs plateformes ainsi qu'une communication en temps réel.

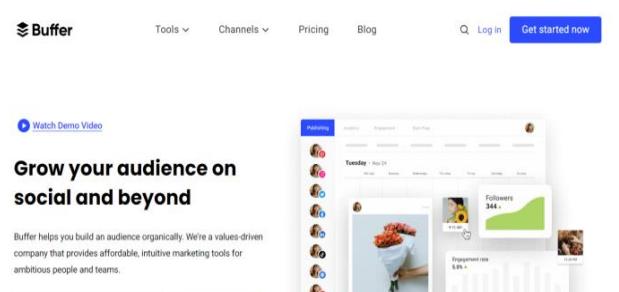
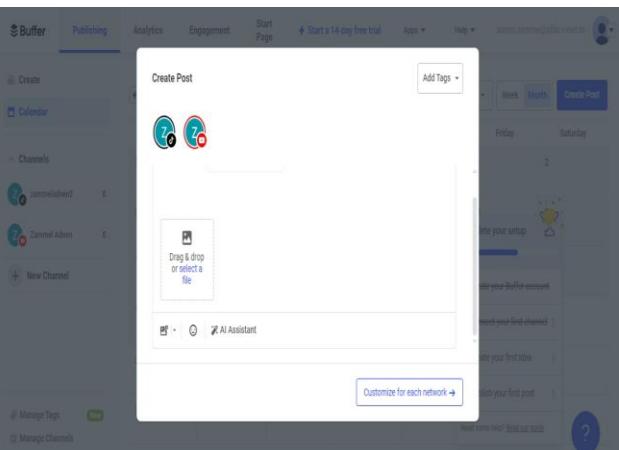
URL : https://buffer.com		
Analyse fonctionnelle	<ul style="list-style-type: none"> - Gestion complète des utilisateurs et des permissions. - Création et gestion de publications. - Communication et collaboration en temps réel. 	
Points forts	<ul style="list-style-type: none"> - Interface utilisateur intuitive et conviviale. - Planification avancée des publications sur plusieurs plateformes de médias sociaux. - Système de notifications intégré. 	
Points faibles	<ul style="list-style-type: none"> - Définition et planification de campagnes non disponibles. - Analyse approfondie des performances limitée sur la version gratuite. 	

Tableau 4 - Les fonctionnalités de la plateforme Buffer

1.5 Solution proposée

Après une analyse approfondie des solutions actuellement disponibles sur le marché, Tunisian Cloud nous a confié la mission de concevoir une application web révolutionnaire destinée à transformer la manière dont les entreprises et les organisations abordent la gestion et la planification de leurs publications sur les réseaux sociaux. Notre solution se distingue par une gestion avancée des utilisateurs et des autorisations, simplifiant ainsi les processus d'inscription, de connexion et de gestion des profils, des rôles et des permissions. De plus, elle offre la possibilité de créer et de gérer des entreprises avec une personnalisation des hashtags, ce qui constitue un atout majeur pour une stratégie de contenu efficace. En outre, notre

application propose une planification stratégique des campagnes, avec des objectifs clairs, un calendrier flexible et une distribution des tâches aux utilisateurs pour une efficacité optimale. La création, la gestion et la programmation des publications sur différentes plateformes sociales sont également simplifiées grâce à notre solution. Enfin, pour favoriser la communication et la collaboration, la solution intègre des fonctionnalités de commentaires et d'affectation de tâches en temps réel, tout en garantissant une réactivité maximale grâce à des alertes instantanées pour les événements importants. Le tableau comparatif suivant met en lumière la valeur ajoutée de notre solution **SocialScope** par rapport aux alternatives existantes:

Solution Fonctionnalités	Later	Fanpagekarma	Tailwind Publish	Buffer	SocialScope Notre Solution
Interface utilisateur intuitive et conviviale	+	+	-	+	+
Gestion complète des utilisateurs et des permissions	+	+	+	+	+
Création et gestion d'entreprises et de hashtags	+	-	+	-	+
Définition et planification de campagnes	+	+	-	+	+
Création et gestion de publications	+	+	+	+	+
Communication et collaboration en temps réel	-	+	-	+	+
Système de notifications intégré	+	-	-	+	+
Rapports et analyses	+	-	-	+	+

Tableau 5- Avantages différenciés : SocialScope face aux alternatives concurrentes

1.6 Langage et Méthodologie de conception

Cette partie se divise en deux sections distinctes : la sélection de la méthodologie de conception et l'usage du langage de modélisation UML. Chacune de ces sous-sections examine respectivement notre approche méthodologique et l'intégration du langage UML dans le processus de conception.

1.6.1 Choix de la méthodologie de conception

Dans le domaine informatique, l'approche méthodique et la gestion efficace de projet sont cruciales pour atteindre les objectifs fixés. Les équipes de développement sont confrontées à des défis tels que la gestion du temps, des coûts de développement et des risques d'erreurs. Pour y faire face, l'adoption d'une méthodologie de développement est impérative. Cette méthodologie permet de planifier le travail, de répartir les tâches et d'établir des échéances claires, assurant ainsi une vue d'ensemble du processus de développement. Elle garantit le respect des délais de livraison, l'amélioration de la qualité du logiciel et facilite la gestion de toutes les phases du développement. Pour gérer efficacement les problèmes et les tâches, une approche structurée et méthodique, basée sur des méthodes éprouvées, est nécessaire. L'intégration de bonnes méthodes adaptées aux exigences du projet est essentielle pour des résultats optimaux. Dans cette perspective, l'adoption d'une méthodologie robuste pour la gestion du cycle de vie du projet, comme les méthodes agiles, est vivement recommandée en raison de leur flexibilité et de leur capacité à s'adapter aux évolutions des besoins du projet.

1.6.1.1 Concept agile

Le concept agile, dans le domaine de la gestion de projet et du développement logiciel, désigne une approche itérative et collaborative qui met l'accent sur la flexibilité, la réactivité aux changements et la livraison continue de valeur. Les méthodes agiles sont spécifiquement conçues pour s'adapter aux évolutions des besoins des clients et de l'environnement du projet, ce qui les oppose souvent aux approches traditionnelles plus rigides et planifiées. Elles favorisent la collaboration, la communication et la livraison continue de produits fonctionnels. Les méthodes agiles sont particulièrement adaptées aux projets complexes et évolutifs, où les exigences peuvent fluctuer rapidement. Elles reposent généralement sur quatre valeurs fondamentales :

- Priorité aux individus et aux interactions plutôt qu'aux processus et aux outils: les méthodes agiles mettent l'accent sur la coopération et la communication au sein de l'équipe, reconnaissant que les personnes sont essentielles à la réussite du projet. Elles favorisent les échanges directs plutôt que de se concentrer uniquement sur les procédures et les outils.

- Livraison de logiciels fonctionnels plutôt que sur une documentation exhaustive: les méthodes agiles encouragent la fourniture continue de produits fonctionnels, privilégiant la production rapide de résultats concrets plutôt que la documentation détaillée. Cela permet d'obtenir rapidement des retours d'utilisateurs et de s'adapter aux besoins changeants du projet.
- Collaboration avec le client plutôt que négociation contractuelle : les méthodes agiles favorisent une étroite collaboration avec le client tout au long du projet, établissant une relation de partenariat où le client participe au processus de développement et contribue à la prise de décision grâce à des retours réguliers.
- Réactivité aux changements plutôt que suivi strict d'un plan : les méthodes agiles reconnaissent que les besoins et les priorités peuvent évoluer au cours du projet. Elles encouragent à être réactif et à s'adapter aux changements plutôt que de s'en tenir rigoureusement à un plan initial, permettant ainsi de maximiser la valeur livrée en répondant aux besoins réels au fur et à mesure qu'ils se manifestent.

1.6.1.2 Principales Méthodes Agiles et Choix de Scrum

Parmi les méthodes agiles les plus renommées, citons Scrum, Kanban et Extreme Programming (XP). Voici un aperçu des principes de chacune de ces méthodologies :

- Scrum : c'est l'une des méthodes agiles les plus populaires, basée sur des cycles de développement itératifs appelés "sprints". Elle implique des rôles clés tels que le Product Owner, le Scrum Master et l'équipe de développement pour planifier, exécuter et évaluer les tâches. Les réunions régulières, telles que la réunion de planification du sprint et la rétrospective, favorisent la collaboration et la transparence.
- Kanban : c'est une méthode visuelle axée sur la gestion du flux de travail. Les tâches sont représentées par des cartes et déplacées à travers différentes colonnes pour indiquer leur statut. Kanban met l'accent sur la limitation du travail en cours pour améliorer la productivité et la gestion des priorités.
- Extreme Programming (XP) : c'est une méthode agile axée sur l'ingénierie logicielle. Elle met l'accent sur la qualité du code, les tests automatisés, la collaboration étroite entre les développeurs et les clients, ainsi que l'itération rapide du développement. XP encourage également la communication continue et la rétroaction régulière pour améliorer le processus de développement.
- Lean : Bien que le Lean ne soit pas exclusivement une méthode agile, il est souvent associé aux principes agiles. Le Lean se concentre sur l'élimination des gaspillages, l'amélioration

continue et la création de valeur pour le client. Il met l'accent sur la réduction des délais, l'optimisation des processus et la satisfaction des besoins réels du client.

Chacune de ces approches offre des solutions uniques pour gérer les projets de manière flexible, itérative et collaborative. Scrum, en particulier, se distingue par sa popularité et sa capacité à répondre aux besoins changeants des projets. Fondée sur des cycles de développement itératifs appelés "sprints", Scrum met l'accent sur la collaboration, la transparence et la livraison continue de produits fonctionnels. Avec des rôles clés tels que le Product Owner, le Scrum Master et l'équipe de développement, Scrum offre une structure claire pour planifier, exécuter et évaluer les tâches de manière efficace. Les réunions régulières, telles que la réunion de planification du sprint et la rétrospective, favorisent la communication et permettent des ajustements rapides en réponse aux retours d'utilisateurs et aux changements de priorités.

C'est pourquoi, pour atteindre les objectifs du projet « **SocialScope** », notre choix s'est porté sur la méthode Scrum. Son approche flexible et axée sur la collaboration semble parfaitement adaptée pour répondre aux exigences dynamiques de notre projet. En adoptant Scrum, nous sommes convaincus de pouvoir tirer parti de ses principes éprouvés pour assurer une gestion efficace et une livraison réussie de nos résultats.

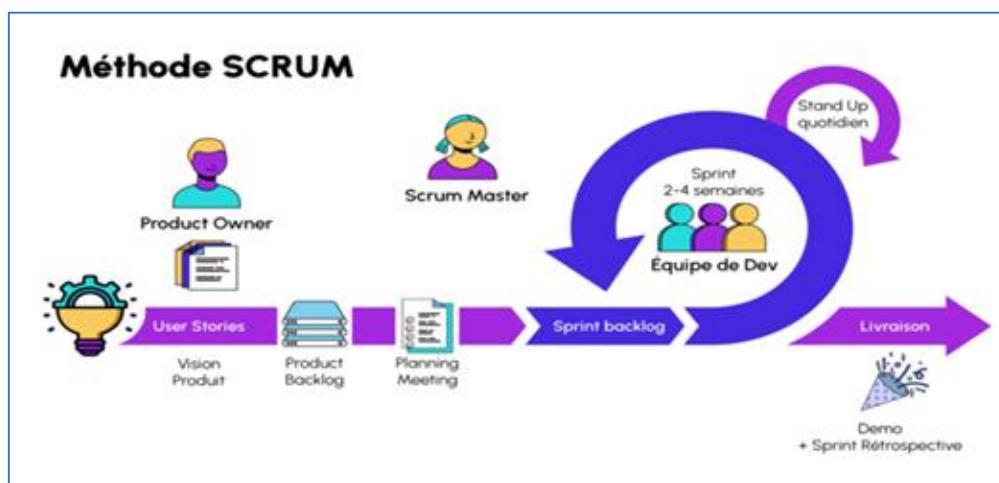


Figure 5 –Processus Scrum

1.6.1.3 La méthodologie Scrum

La méthode Scrum, créée en 2002, tire son nom du rugby, où "Scrum" signifie "la mêlée". Elle se distingue par son approche de travail flexible plutôt qu'une méthode rigide. L'un de ses principaux avantages est sa capacité à produire rapidement un résultat final en s'appuyant sur une équipe cohérente. Cette équipe travaille à atteindre un objectif évolutif qui se transforme au fil de cycles itératifs appelés "Sprints".

La durée d'un sprint peut varier de 15 à 30 jours, et à la fin de chaque sprint, l'équipe présente un produit conforme aux spécifications initiales. Cette approche permet au propriétaire du produit de réévaluer les priorités des fonctionnalités à mesure que le développement progresse. De plus, la fin de chaque sprint offre l'opportunité de prendre des décisions quant à la livraison ou de lancer un nouveau sprint pour améliorer le produit.

1.6.1.4 Liste des intervenants

Dans Scrum, plusieurs intervenants clés jouent des rôles spécifiques dans la gestion du projet. Voici les principaux intervenants dans Scrum :

- Product Owner : cet intervenant est responsable de définir et de prioriser le backlog du produit. En étroite collaboration avec les parties prenantes, il comprend leurs besoins et leurs attentes. Le Product Owner prend des décisions sur les fonctionnalités à développer, définit les objectifs du projet et s'assure que la valeur est ajoutée au produit.
- Scrum Master : cet intervenant facilite la mise en œuvre de Scrum et veille à ce que l'équipe respecte les principes et les processus de Scrum. Il aide à résoudre les obstacles et les problèmes rencontrés par l'équipe, facilite les réunions et encourage la collaboration. Le Scrum Master joue un rôle de coach pour l'équipe et garantit que Scrum est correctement compris et appliqué.
- Équipe de développement : cette équipe réalise les tâches et crée le produit. Elle est auto-organisée et pluridisciplinaire, possédant ainsi les compétences nécessaires pour mener à bien les différentes tâches du projet. En collaboration avec le Product Owner pour comprendre les exigences et avec le Scrum Master pour garantir le suivi du processus de Scrum, l'équipe de développement assure le bon déroulement du projet.

Ces intervenants jouent des rôles clés dans la mise en œuvre réussie de Scrum. Leur collaboration et leur interaction sont essentielles pour assurer le succès du projet et la livraison d'un produit de haute qualité.

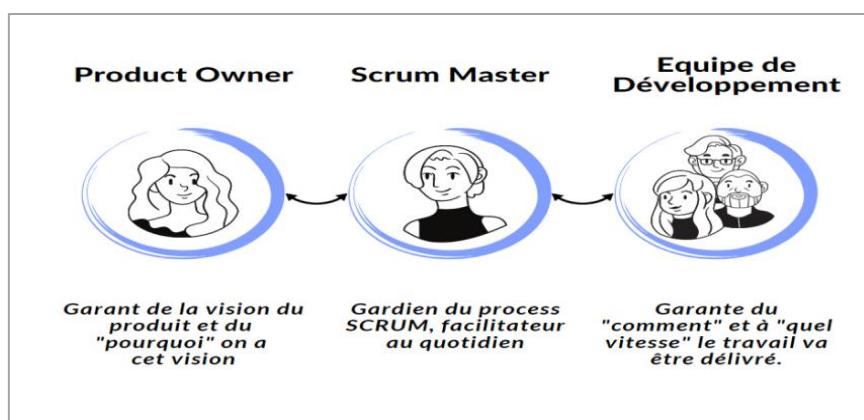


Figure 6 – Les intervenants dans SCRUM

1.6 Liste des artefacts

Dans Scrum, plusieurs artefacts clés sont utilisés pour faciliter la gestion du projet et la communication au sein de l'équipe à savoir :

- Product Backlog (Carnet de produits) : Le Product Backlog est une liste ordonnée de toutes les fonctionnalités, améliorations et corrections de bugs qui doivent être réalisées pour le produit. Géré par le Product Owner, il est constamment mis à jour pour refléter les besoins et les priorités du projet. Le Product Backlog est utilisé pour planifier les sprints et déterminer les fonctionnalités à inclure dans chaque itération.
- Sprint Backlog (Carnet d'itération) : Le Sprint Backlog est une liste des tâches spécifiques qui doivent être réalisées pendant un sprint donné. Créé par l'équipe de développement à partir du Product Backlog, il sélectionne les fonctionnalités prioritaires à inclure dans le sprint. Le Sprint Backlog est utilisé pour suivre les progrès de l'équipe pendant le sprint et garantir que toutes les tâches sont terminées à sa fin.
- Incrément (Graphique d'avancement) : L'Incrément est le résultat du travail accompli par l'équipe de développement à la fin de chaque sprint. Il s'agit d'une version fonctionnelle et utilisable du produit incluant toutes les fonctionnalités terminées pendant le sprint. L'Incrément est inspecté lors de la revue de sprint et peut être déployé ou livré aux parties prenantes si elles le souhaitent.

Ces artefacts aident à organiser le travail, à suivre les progrès et à faciliter la communication au sein de l'équipe. Ils permettent de garder une vision claire des objectifs du projet et de garantir que le travail est effectué de manière efficace et itérative.



Figure 7 – Les artefacts dans SCRUM

- Création du backlog du produit : le Product Owner est chargé de créer et de maintenir le backlog du produit. En collaboration étroite avec les parties prenantes, il identifie les besoins et les exigences du projet, alimentant ainsi le backlog avec une liste ordonnée de fonctionnalités, d'améliorations et de corrections de bugs nécessaires pour le produit.
- Planification du sprint : l'équipe de développement et le Product Owner se réunissent pour planifier le sprint. Ils sélectionnent les fonctionnalités prioritaires à inclure dans le sprint à partir du backlog du produit. L'équipe de développement estime le temps nécessaire pour chaque fonctionnalité et détermine la quantité de travail réalisable pour le sprint.
- Création du sprint backlog : une fois les fonctionnalités sélectionnées, l'équipe de développement crée le sprint backlog. Ce dernier consiste en une liste des tâches spécifiques à réaliser pendant le sprint. Chaque tâche est estimée en termes de temps et de complexité.
- Réunion de planification du sprint : l'équipe de développement se réunit pour discuter du sprint backlog et planifier l'exécution des tâches. Les membres répartissent les tâches, estiment le temps nécessaire pour chacune d'elles et définissent les critères d'acceptation pour chaque fonctionnalité.
- Exécution du sprint : pendant le sprint, l'équipe de développement travaille sur les tâches du sprint backlog. Des réunions quotidiennes de stand-up sont tenues pour discuter des progrès, des obstacles et de la coordination des efforts. L'équipe s'engage à terminer toutes les tâches du sprint backlog d'ici la fin du sprint.
- Revue de sprint : à la fin du sprint, l'équipe de développement présente le travail accompli lors d'une revue de sprint. Ils démontrent les fonctionnalités terminées et recueillent les commentaires des parties prenantes. Le Product Owner décide si le travail répond aux critères d'acceptation et si l'incrément est prêt à être livré.
- Rétrospective de sprint : Après la revue de sprint, l'équipe de développement tient une rétrospective pour réfléchir sur le processus et identifier les améliorations à apporter. Ils discutent des points forts, des points faibles et des actions à prendre pour améliorer la productivité et la qualité du travail.

Ces étapes de planification permettent à l'équipe de développement de gérer efficacement le projet, de s'adapter aux changements et de livrer de la valeur de manière itérative. La planification est un processus continu tout au long du projet, avec de nouvelles fonctionnalités ajoutées au backlog du produit et de nouveaux sprints planifiés régulièrement.

1.6.2 Choix du langage de modélisation

Une fois la méthodologie choisie, il est crucial de sélectionner un langage de modélisation unifié qui réponde efficacement aux exigences de notre projet. C'est pourquoi nous avons décidé d'opter pour UML comme langage de modélisation.

1.6.2.2 Le langage UML

UML (Unified Modeling Language) est un langage de conception orienté objet qui offre une approche complète pour modéliser divers systèmes d'information tels que des applications web, des sites web, des bases de données, etc. Ce langage propose une gamme de diagrammes organisés en vues statiques, représentant la structure physique du système à modéliser, ainsi qu'en vues dynamiques, illustrant le fonctionnement du système (par exemple, le diagramme des séquences).

1.6.2.2 Raisons de l'Adoption d'UML

UML privilégie la création de modèles plutôt qu'une approche qui sépare strictement l'analyse de la conception. Les modèles d'analyse et de conception se distinguent uniquement par leur niveau de détail, sans différence dans les concepts utilisés. UML ne présente pas de spécificités de modélisation propres à une activité particulière (analyse, conception, etc.) ; le langage reste uniforme à tous les niveaux d'abstraction. Cette approche simplifiée facilite la transition entre les niveaux d'abstraction, encourageant ainsi une approche non linéaire. Les allers-retours entre différents niveaux d'abstraction sont facilités, et l'unicité du langage assure la traçabilité entre les modèles de différents niveaux. Nous avons choisi d'adopter UML pour plusieurs raisons :

- UML est un standard reconnu.
- Il permet une modélisation à un niveau très élevé et indépendant des langages et des environnements.
- La notation UML facilite la compréhension et la communication des modèles orientés objet.

1.7 Conclusion

Dans ce chapitre, nous avons exposé le cadre général de notre projet en identifiant sa problématique et en proposant une solution pour répondre à cette situation. Nous avons également dévoilé le langage et la méthodologie de conception qui seront employés dans les prochains chapitres de ce rapport, tout en justifiant notre choix.

Chapitre 2

Sprint 0 : Planification du projet



2.1 Introduction

Dans ce chapitre intitulé "Planification du Projet", nous commencerons par explorer la phase essentielle de la capture des besoins. L'objectif premier consistera à identifier les différents intervenants impliqués dans le projet, tout en mettant en évidence les exigences fonctionnelles et non fonctionnelles que la solution doit satisfaire. Ensuite, une modélisation détaillée de toutes les fonctionnalités sera réalisée à l'aide d'un diagramme de cas d'utilisation global. Enfin, la méthodologie Scrum sera adoptée pour la gestion dynamique du projet, garantissant ainsi une approche agile et collaborative tout au long de son développement.

2.2 Identification des acteurs du système

Chaque système informatique offre des fonctionnalités et des informations pertinentes aux utilisateurs, leur permettant ainsi d'interagir efficacement avec celui-ci. Les utilisateurs de notre système peuvent être catégorisés de la manière suivante :

- **Utilisateur** : L'utilisateur final du système, qui interagit avec le système pour accomplir des tâches spécifiques selon son rôle et ses autorisations définis. Cela peut inclure des actions telles que la gestion de profils, la consultation de publications, la collaboration avec d'autres utilisateurs et la réception de notifications pertinentes...
- **Admin** : l'administrateur du système, responsable de la gestion des utilisateurs, des rôles et des configurations du système.

2.3 Identification des besoins

La spécification des besoins met en lumière les fonctionnalités essentielles qu'un système doit offrir afin de répondre aux besoins de ses utilisateurs. Dans cette partie, l'accent est mis sur la description détaillée des exigences fonctionnelles et non fonctionnelles du système « SocialScope ».

2.3.1. Besoins fonctionnels

Après notre étude et en accord avec les besoins fixés par Tunisian Cloud, les fonctionnalités de notre système "**SocialScope**" se présentent comme suit :

- **Gestion des utilisateurs** : l'application permet aux administrateurs de créer des comptes pour les utilisateurs, qui peuvent ensuite se connecter et gérer leurs profils. Les administrateurs ont également le pouvoir de définir les rôles des utilisateurs, offrant ainsi un contrôle personnalisé sur l'accès et les autorisations.
- **Gestion des entreprises** : l'application permet de créer, modifier et supprimer des profils entrepris, ainsi que d'associer des hashtags spécifiques à chacune pour une meilleure stratégie de contenu.

- **Définition et planification des campagnes²** : l'application permet aux utilisateurs de définir leurs objectifs de campagne, de planifier leurs actions à l'avance avec un calendrier intégré, et aux administrateurs d'attribuer équitablement les tâches pour chaque campagne.
- **Gestion des publications** : l'application permet aux utilisateurs de créer, modifier et supprimer des contenus sur les réseaux sociaux. Ils peuvent aussi planifier la publication à des moments précis et choisir le type de publication pour une diffusion optimale sur différentes plateformes.
- **Communication et collaboration** : l'application permet aux utilisateurs de laisser des commentaires sur les publications, favorisant ainsi la collaboration et la communication en équipe. De plus, les administrateurs peuvent attribuer des tâches spécifiques à chaque utilisateur et suivre leur avancement pour une gestion efficace des projets.
- **Réception de notifications** : l'application envoie des notifications aux utilisateurs pour les événements importants comme les nouvelles publications, les commentaires et les tâches attribuées, assurant ainsi une gestion proactive et une réponse rapide sur la plateforme.
- **Consulter les statistiques**: l'application propose aux utilisateurs une fonction de génération de rapports pour suivre les performances des publications et des campagnes, ainsi qu'une analyse des performances pour évaluer leur efficacité et prendre des décisions informées pour l'avenir.

2.3.2. Besoins non fonctionnels

Les besoins non fonctionnels d'une application déterminent les limites et les exigences liées à son utilisation, à son évolutivité et à ses performances techniques. Notre application doit répondre aux exigences suivantes en matière de besoins non fonctionnels :

- **Performance** : pour garantir l'enregistrement précis des données, il est impératif que les réponses du système soient très rapides. À cette fin, l'application doit implémenter une fonction dédiée facilitant le chargement des fichiers volumineux (images ou vidéos) depuis la machine locale de l'utilisateur vers la plateforme.
- **Sécurité** : la sécurité revêt une importance capitale pour ce type d'application informatique. Il ainsi impératif de veiller à ce que toutes les informations confidentielles soient rigoureusement protégées. Dans cette optique, plusieurs mesures vont être mises en place:
 - Les mots de passe sont cryptés avant d'être stockés dans la base de données, assurant ainsi une protection adéquate des données sensibles.

² *Campagne* : en anglais *campaign*, c'est un terme qui désigne le plus souvent une série coordonnée d'activités pour atteindre un objectif précis, comme une campagne publicitaire ou une campagne politique.

- Les fichiers sont sécurisés grâce au contrôle de leur taille et de leur type, éliminant ainsi les vulnérabilités potentielles. À titre d'exemple, en limitant la taille des fichiers téléchargés à 8 Mo, la sécurité est renforcée en évitant les fichiers excessivement volumineux, tandis que la vérification des types de fichiers autorisés prévient les risques liés à l'exécution de code malveillant.
- Les variables d'entrée de l'application sont soumises à des mesures de sécurité strictes pour garantir leur conformité avec les données enregistrées dans la base de données. Des techniques telles que la maximisation et la minimisation du nombre de caractères sont appliquées pour prévenir les attaques par injection SQL Par exemple, dans la fonction d'inscription des utilisateurs, les annotations binding telles que "required", "min" et "max", ainsi que la validation du format email, jouent un rôle crucial dans la sécurisation des champs de saisie. Ces mesures visent à éviter les champs vides, à limiter les longueurs excessives des chaînes de caractères, et à valider les adresses email, réduisant ainsi le risque d'introduction de code malveillant dans les requêtes SQL.
- Les URL, avec les vérifications supplémentaires effectuées, en prenant en compte la session en cours, constituent une mesure significative pour minimiser le risque d'exposition de données personnelles. Dans chaque interaction sur la plateforme, telle que la planification d'une publication ou la création d'une tâche, l'identification de chaque utilisateur se fait à travers l'ID (ou les ID) des entreprises auxquelles il appartient. Cette approche garantit que seuls les utilisateurs autorisés peuvent effectuer des actions spécifiques, renforçant ainsi la sécurité et préservant la confidentialité des données personnelles.
- Les risques de bogues sont réduits grâce à l'utilisation de constantes pour gérer les limitations et la pagination, assurant ainsi le bon fonctionnement de la plateforme. Par exemple, lors de l'affichage des publications, le système permet à l'utilisateur de choisir entre des listes regroupées en 5, 10, 20 ou 50 publications par page. En cas d'échec ou d'entrée invalide, l'affichage par défaut est limité à 10 publications par page. Cette approche garantit une expérience utilisateur stable et cohérente, tout en minimisant les erreurs potentielles liées à la pagination et à la gestion des données.
- **Fiabilité :** la fiabilité de la plateforme est une priorité absolue pour garantir une expérience utilisateur optimale. L'application s'efforce de maintenir une disponibilité continue de ses services, minimisant ainsi les temps d'arrêt et assurant une accessibilité maximale pour tous les utilisateurs. À cette fin, un gestionnaire d'exceptions est déployé pour faire face aux

éventuelles pannes matérielles ou logicielles, prévenant ainsi la propagation d'erreurs à travers tous les services du système.

- **Maintenance :** le code de l'application doit être bien documenté pour expliquer ses différentes parties et leur utilisation. Chaque section du package doit être décrite en détail pour aider les développeurs à comprendre son fonctionnement. Les commentaires doivent également fournir des informations sur les dépendances du package et sur l'utilisation des fonctions et des structures incluses. Enfin, la date de la dernière mise à jour doit être indiquée pour informer les développeurs de la fraîcheur du code. Ces commentaires bien documentés sont essentiels pour faciliter la maintenance en aidant les développeurs à comprendre, modifier et étendre le code plus facilement.
- **Convivialité :** l'application s'efforce de fournir une interface utilisateur conviviale, même pour les débutants. Elle doit intégrer des fonctionnalités intuitives telles que des icônes claires et reconnaissables, ainsi que des menus organisés offrant une variété d'options de mise en page et de choix d'affichage. De plus, des instructions claires et des info-bulles peuvent être mises en place pour aider les utilisateurs à comprendre les fonctionnalités de l'interface. Ces efforts visent à rendre l'application plus accessible et à offrir une expérience utilisateur satisfaisante et agréable pour tous, quel que soit leur niveau d'expérience.

2.4 Diagramme des cas d'utilisation global

Un cas d'utilisation (Use case) est une manière de représenter de manière fonctionnelle le système, illustrant les interactions entre les acteurs et le système afin de générer un résultat concret et significatif. Son objectif est de définir les différentes fonctionnalités du système et d'approfondir la compréhension de son fonctionnement.

Le schéma suivant illustre le diagramme de cas d'utilisation global de notre système :

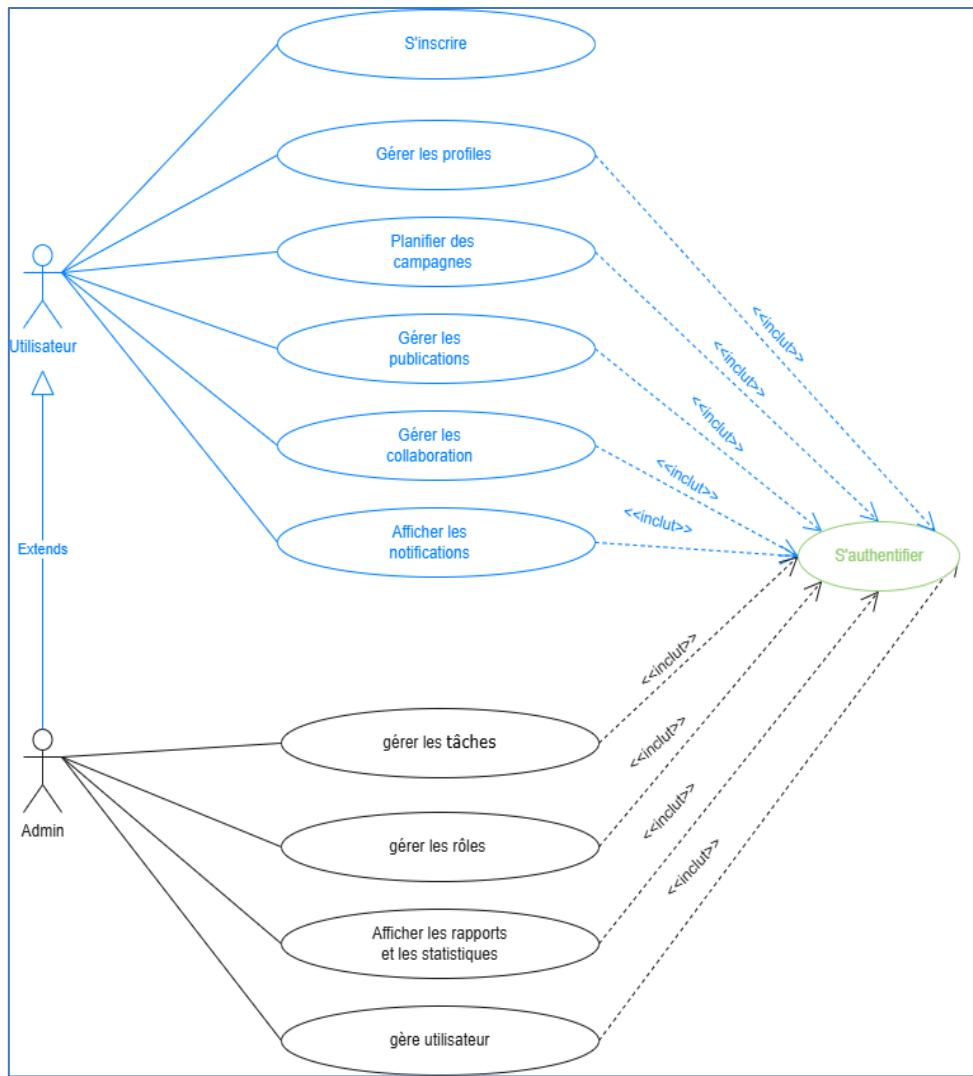


Figure 8- Diagramme des cas d'utilisation global

2.5 Diagramme de classes global

Un diagramme de classes est une représentation visuelle des composants fondamentaux d'un système orienté objet. Il met en évidence les classes, leurs attributs et opérations, ainsi que les relations entre ces entités. Les classes sont présentées sous forme de rectangles à trois lignes, indiquant respectivement le nom de la classe, ses attributs et ses méthodes. Les relations entre les classes sont également visualisées, ainsi que les sous-classes dérivées d'autres classes. Les diagrammes de classes sont largement utilisés par les ingénieurs logiciels pour documenter l'architecture des logiciels, permettant une meilleure compréhension de la structure globale du système et facilitant la communication entre les équipes de développement.

La figure ci-dessus illustre le diagramme de classe global de notre application SocialScope :

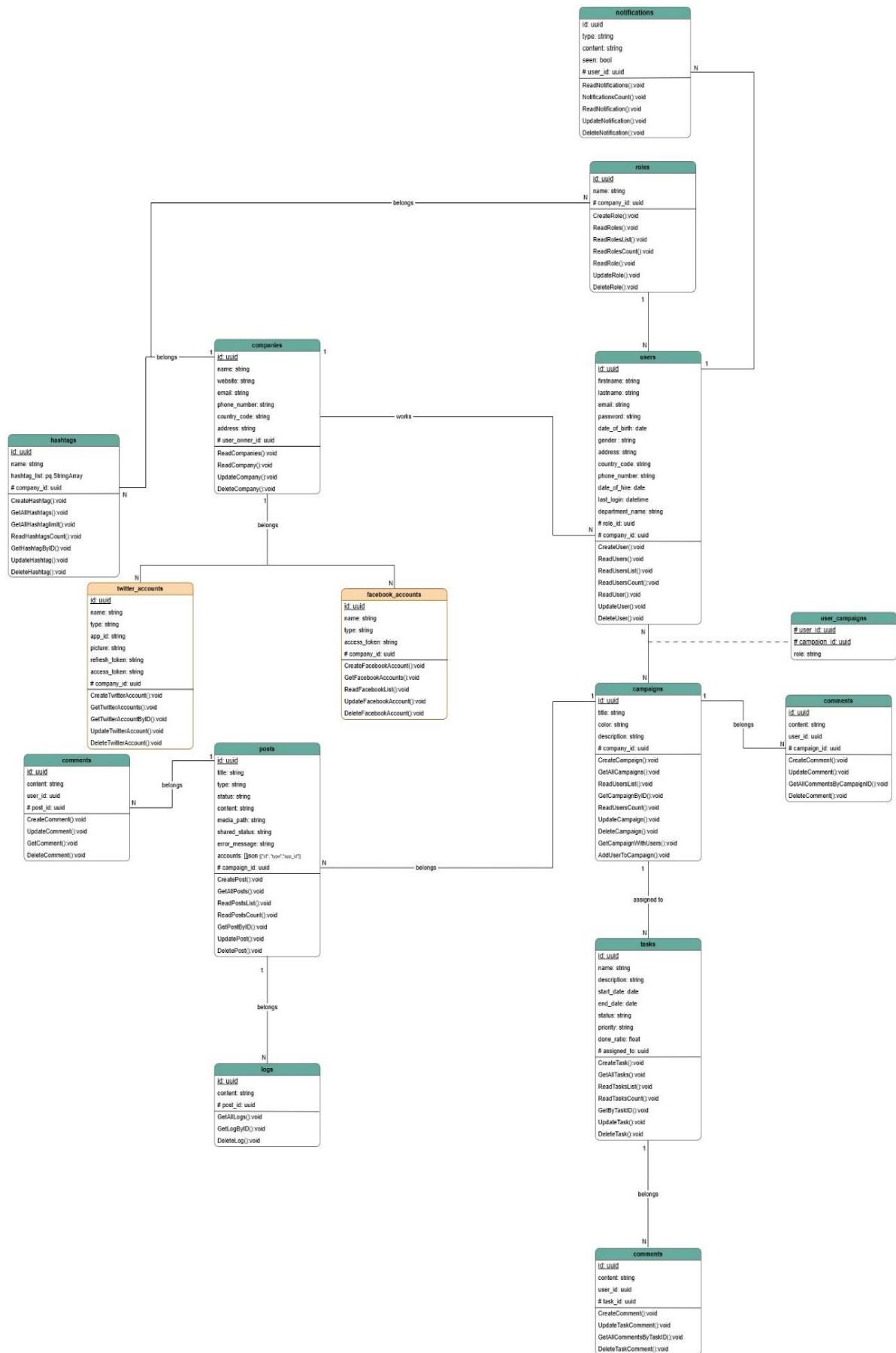


Figure 9- Diagramme de classes global

2.6 Structure et découpage du projet

Cette section se penche sur la structure et le découpage du projet, en détaillant les différentes composantes et leur organisation. L'objectif est d'assurer une gestion efficace du développement.

2.6.1. Identification de l'équipe SCRUM

L'équipe joue un rôle crucial dans Scrum en optimisant la productivité et la flexibilité. Elle doit être auto-organisée, posséder un large éventail de compétences et disposer de la latitude nécessaire pour accomplir ses tâches de manière efficace. Dans le contexte de notre projet, nous identifions :

		
<p><i>Product Owner :</i> <i>Hedi Manai</i></p>	<p><i>Scrum Master :</i> <i>Hassan Ben Rebah</i></p>	<p><i>Development team :</i> <i>Zammel Adnen</i> & <i>Mohamed Amin Mallek</i></p>

Tableau 6- Équipe du Projet SocialScope

2.6.2. Backlog du produit

Le tableau ci-dessus représente le backlog de produit initial, où sont répertoriées une série de "histoire utilisateurs" détaillant les fonctionnalités de notre système. Ces histoire utilisateurs sont organisées selon leur niveau de complexité. En effet, les plus simples et les plus claires sont privilégiés en premier, tandis que les plus complexes sont envisagées ultérieurement.

ID	Fonctionnalité	Histoire utilisateur (User Story)	Priorité	Complexité
1	Authentification	En tant qu'administrateur, je veux pouvoir me connecter à l'espace de l'administration afin de gérer mon espace de travail.	1	Faible
2		En tant qu'utilisateur, je veux pouvoir me connecter au système.	2	Faible

ID	Fonctionnalité	Histoire utilisateur (User Story)	Priorité	Complexité
3	Gestion des profils	En tant qu'utilisateur, je veux pouvoir modifier mon profil.	3	Faible
4		En tant qu'utilisateur, je veux pouvoir mettre à jour mes préférences.	4	Faible
5	Gestion des utilisateurs	En tant qu'administrateur, je veux pouvoir ajouter un nouvel utilisateur.	5	Moyenne
6		En tant qu'administrateur, je veux pouvoir modifier les informations relatives à un utilisateur existant.	6	Moyenne
7		En tant qu'administrateur, je veux pouvoir supprimer un utilisateur.	7	Moyenne
8	Gestion des rôles	En tant qu'administrateur, je veux pouvoir créer un nouvel rôle utilisateur.	25	Moyenne
9		En tant qu'administrateur, je veux pouvoir modifier les autorisations d'un rôle existant.	26	Moyenne
10		En tant qu'administrateur, je veux pouvoir supprimer un rôle utilisateur.	27	Faible
11	Gestion d'hashtag	En tant qu'utilisateur, je veux pouvoir créer un hashtag.	21	Moyenne
12		En tant qu'utilisateur, je veux pouvoir consulter les hashtags à différents niveaux.	15	Faible
13		En tant qu'utilisateur, je veux pouvoir modifier un hashtag.	25	Moyenne
14		En tant qu'utilisateur, je veux pouvoir supprimer un hashtag.	7	Faible
15	Gestion des campagnes	En tant qu'utilisateur, je veux pouvoir créer une campagne.	8	Faible
16		En tant qu'utilisateur, je veux pouvoir consulter ma liste de campagnes.	9	Faible

ID	Fonctionnalité	Histoire utilisateur (User Story)	Priorité	Complexité
17	Gestion des campagnes	En tant qu'utilisateur, je veux pouvoir modifier une campagne.	10	Faible
18		En tant qu'utilisateur, je veux pouvoir supprimer une campagne.	11	Faible
19		En tant qu'utilisateur, je veux pouvoir échanger des messages avec les autres utilisateurs de campagne.	12	Moyenne
20	Gestion des tâches	En tant qu'administrateur, je veux pouvoir créer une tâche avec une durée spécifiée.	13	Moyenne
21		En tant qu'utilisateur, je veux pouvoir consulter les tâches à différents niveaux.	14	Faible
22		En tant qu'administrateur, je veux pouvoir modifier une tâche.	15	Faible
23		En tant qu'administrateur, je veux pouvoir supprimer une tâche.	16	Faible
24		En tant qu'utilisateur, je veux pouvoir échanger des messages pour une tâche donnée.	17	Moyenne
25	Gestion des publications	En tant qu'utilisateur, je veux pouvoir me connecter à un nouveau compte de réseau social.	18	Moyenne
26		En tant qu'utilisateur, je veux pouvoir créer une nouvelle publication sur les réseaux sociaux.	19	Élevée
27		En tant qu'utilisateur, je veux pouvoir modifier une publication existante non encore publiée.	20	Moyenne
28		En tant qu'utilisateur, je veux pouvoir supprimer une publication existante non encore publié.	21	Faible

ID	Fonctionnalité	Histoire utilisateur (User Story)	Priorité	Complexité
29	Gestion des collaborations	En tant qu'utilisateur, je veux pouvoir collaborer avec d'autres utilisateurs dans le cadre d'un projet.	22	Moyenne
30	Affichage des notifications	En tant qu'utilisateur, je veux pouvoir recevoir une notification sur les activités importantes.	23	Faible
31	Affichage des rapports et des statistiques	En tant qu'administrateur, je veux pouvoir consulter des rapports et des statistiques sur l'utilisation du système.	24	Élevée

Tableau 7- Backlog du produit

Où :

- **ID** : Représente l'identifiant unique de l'user story.
- **Fonctionnalité** : facilite l'organisation des histoires utilisateur en les regroupant par thème.
- **Complexité** : permet de classifier les histoire utilisateurs en termes de dépendance. Elle se décline en trois niveaux : faible, moyenne et élevée (Exemple : une user story peut être simple en soi, mais si elle dépend de la réalisation préalable d'autres tâches, elle peut devenir plus complexe à planifier et à exécuter).
- **Priorité** : classe les histoires utilisateur en fonction de leur valeur métier et de l'ordre de réalisation.

2.6.3. Planification des sprints

La création des sprints représente une étape fondamentale dans la méthodologie de gestion de projet agile. Elle implique la découpe du projet en cycles de développement courts et itératifs, favorisant une planification optimale, une exécution efficace et une adaptation fluide aux éventuels changements. Dans ce processus, les histoire utilisateurs sont soigneusement évaluées et réparties en fonction de leur complexité, de l'estimation du temps nécessaire à leur réalisation, ainsi que des objectifs spécifiques de chaque sprint.

Après avoir minutieusement estimé les histoire utilisateurs, nous avons procédé à la répartition de notre release, débutant le 22 janvier 2024 et se clôturant le 30 Avril 2024, sur quatre sprints. Chaque sprint est conçu pour avoir une durée de trois semaines, offrant ainsi un équilibre entre la flexibilité des itérations courtes et la capacité à produire des résultats tangibles.

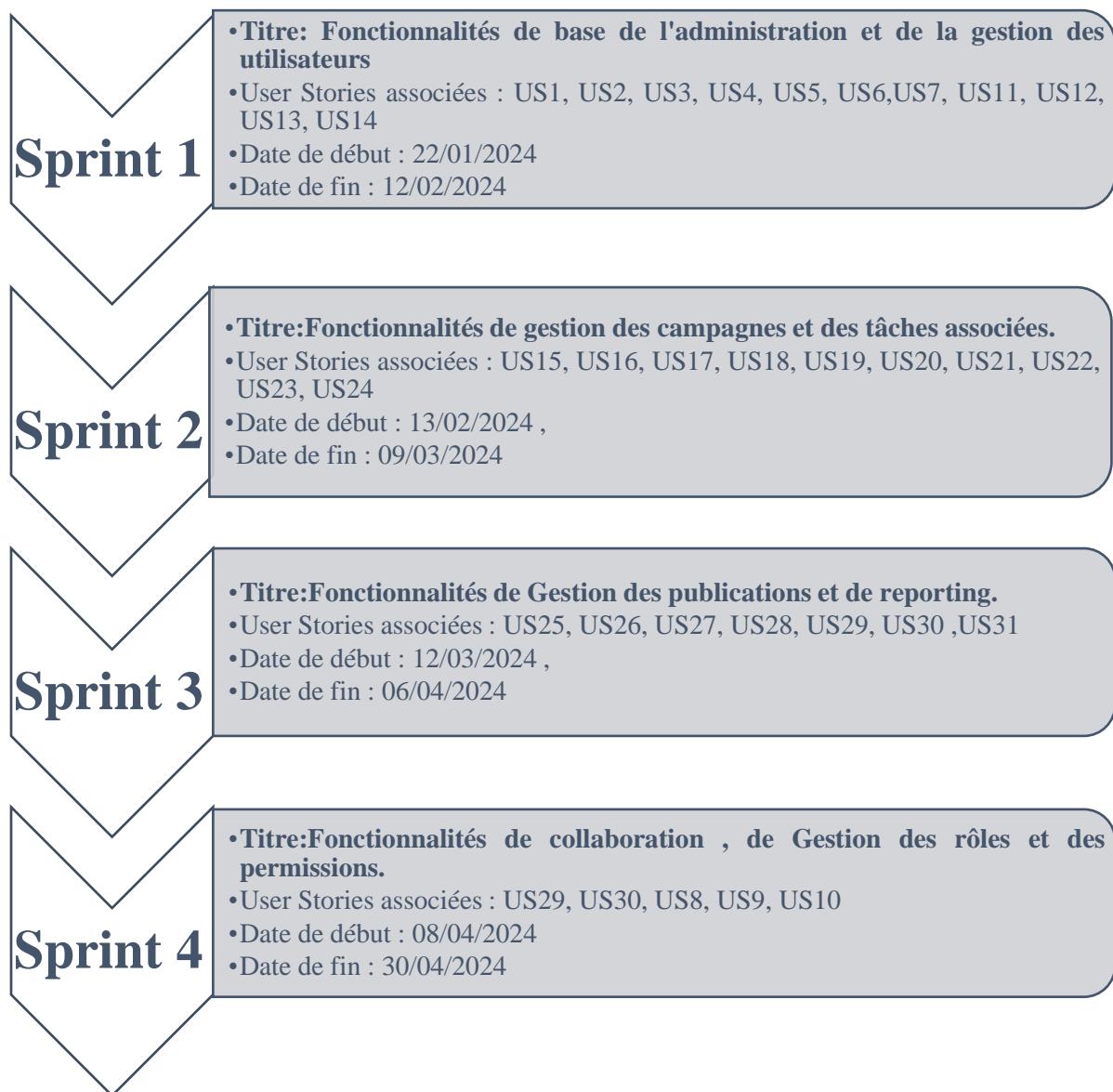


Figure 10- Répartition des sprints du projet

2.7. Architecture et environnement de développement

Cette partie se subdivise en deux sections distinctes : l'Architecture de l'application SocialScope et l'environnement ainsi que les outils de développement utilisés.

2.7.1. Architecture de l'application SocialScope

Lors de la conception de tout système informatique, le choix de son architecture revêt une importance cruciale. Cette décision impacte directement son bon fonctionnement, ses performances optimales, ainsi que sa facilité de réutilisation et d'interconnexion avec d'autres systèmes. C'est dans cette optique que nous avons opté pour le modèle client-serveur lors du développement de notre application web, SocialScope. Ce choix s'est justifié par son efficacité avérée dans la gestion des interactions entre les différents composants de notre système. Dans la section suivante, nous détaillerons cette architecture nommée client-serveur.

2.7.1.1. Définition de l'architecture client serveur

L'architecture client-serveur représente un modèle fondamental dans le domaine de l'informatique, où les tâches et les responsabilités sont distribuées entre les clients et les serveurs. Dans ce cadre, les clients adressent des requêtes aux serveurs pour accéder à des ressources ou des services, tandis que les serveurs y répondent en fournissant les données demandées. Cette architecture se distingue par les caractéristiques suivantes :

- Distribution des responsabilités : les clients et les serveurs possèdent des rôles distincts. Les clients initient les requêtes, tandis que les serveurs y répondent en fournissant les services demandés.
- Communication: Le client envoie des requêtes HTTP aux points d'extrémité de l'API via le module http Client, puis le serveur les reçoit, les analyse et prend des actions appropriées. Ensuite, le serveur envoie une réponse JSON au client, qui traite les données pour mettre à jour l'interface utilisateur.
- Modularité et évolutivité : l'architecture client-serveur encourage la modularité et l'évolutivité des systèmes, car les composants peuvent être développés, déployés et mis à jour indépendamment les uns des autres.
- Centralisation des données : les serveurs agissent souvent comme des points centraux pour stocker et gérer les données, offrant ainsi une source unique d'information pour les clients.
- Interopérabilité : ce modèle facilite l'interopérabilité entre les divers types de clients et de serveurs, puisqu'ils peuvent être mis en œuvre dans une variété d'environnements et de langages de programmation.

2.7.1.2 Les composantes clés de l'architecture client-serveur

Les éléments essentiels du modèle de conception logicielle client-serveur sont les suivants :

- Client: Cette couche gère l'interface utilisateur et interagit avec l'API via des services. Elle comprend des composants responsables de l'affichage des données, de la gestion des entrées utilisateur et du déclenchement des appels API.
- Serveur: Cette couche est construite pour définir les points d'extrémité de l'API RESTful. Elle reçoit les requêtes du client, les traite, interagit avec les sources de données (bases de données, fichiers, etc.) et renvoie des réponses au format JSON.
- Protocole de communication: ce composant définit les règles et les formats utilisés pour la communication entre le client et le serveur grâce à représentation de données en format JSON. Un exemple bien connu est le protocole HTTP (Hypertext Transfer Protocol).

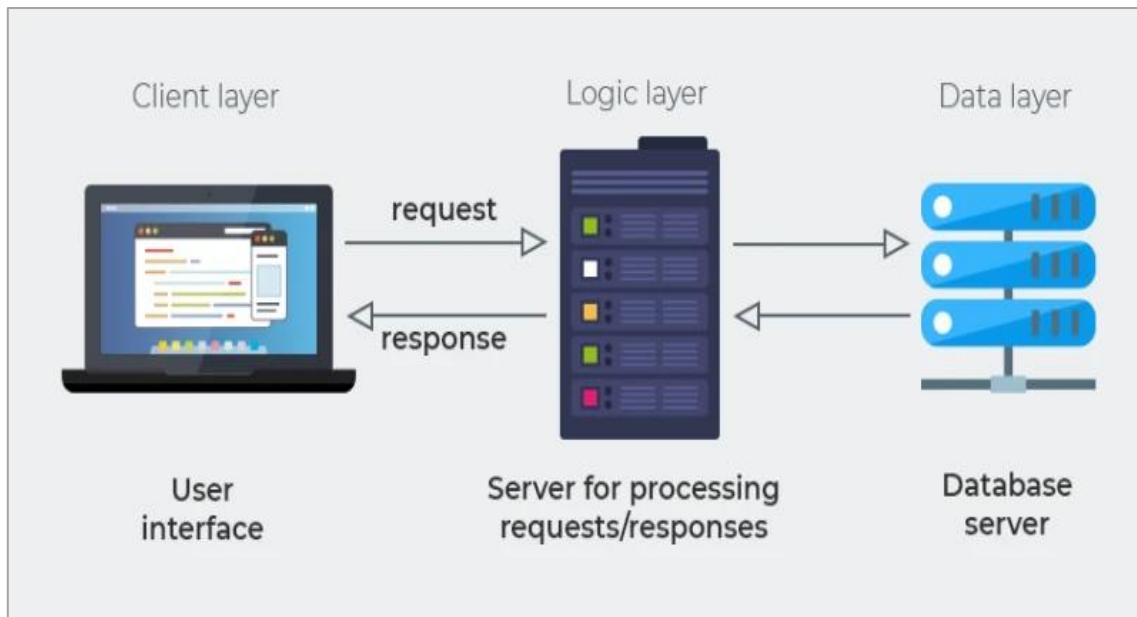


Figure 11- Architecture Client-serveur

2.7.2. Environnement de développement

Dans cette partie, nous allons présenter l'environnement matériel et logiciel utilisé pour la réalisation de notre projet.

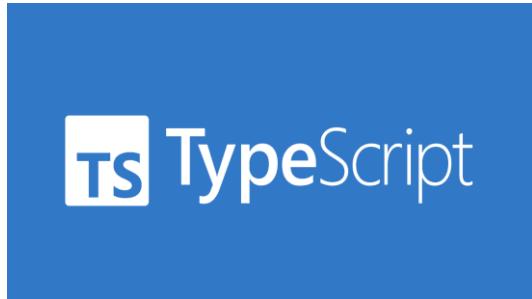
2.7.2.1. Environnement logiciel

Dans notre projet, nous avons utilisé une gamme d'outils tant pour le développement du front-end que pour celui du back-end.

- **Pour la partie Front-End :**



Angular est un framework JavaScript open-source développé par Google, idéal pour créer des applications web dynamiques et interactives. Il propose une architecture basée sur les composants, une liaison de données bidirectionnelle et une gestion efficace des dépendances. Avec Angular, il est possible de développer des Single Page Applications (SPAs) hautement performantes et évolutives. Ce framework complet pour le développement utilise TypeScript comme langage principal et offre une gamme étendue de fonctionnalités pour la création d'applications web. Il intègre notamment un système de liaison de données bidirectionnelle, une injection de dépendances intégrée et des fonctionnalités prêtes à l'emploi telles que la navigation, les formulaires et la gestion des états de manière native. En raison de sa robustesse et de sa capacité à gérer des projets complexes sur le long terme, Angular convient parfaitement aux grandes équipes de développement.



TypeScript

TypeScript développé par Microsoft, est un sur-ensemble typé de JavaScript, offrant des fonctionnalités de typage statique en plus des fonctionnalités de JavaScript de base. En ajoutant des types aux variables, il permet de détecter les erreurs plus tôt dans le processus de développement. Cependant, pour être utilisé dans les navigateurs, il doit être compilé en JavaScript. Souvent associé à des frameworks comme Angular, TypeScript est couramment utilisé pour construire des applications web robustes et évolutives, en particulier dans les projets de grande envergure où il améliore la maintenabilité et la lisibilité du code, malgré sa syntaxe et sa structure plus strictes qui peuvent nécessiter une période d'apprentissage plus longue pour les nouveaux développeurs.



Figma est un éditeur de graphiques vectoriels et un outil de prototypage. Il est principalement basé sur le web, avec des fonctionnalités hors ligne supplémentaires activées par des applications de bureau pour macOS et Windows.

- Pour la partie Back-End :



Golang

Le langage de programmation **Go**, également connu sous le nom de **Golang**, est largement utilisé pour le développement backend en raison de ses nombreux avantages. Sa syntaxe simple et concise permet de créer rapidement des applications robustes et performantes. De plus, son système de gestion des dépendances simplifié facilite le processus de développement et de maintenance du code. Grâce à son support natif de la concurrence et des canaux, Go est idéal pour les applications nécessitant une haute disponibilité et une gestion efficace des requêtes concurrentes. En outre, sa vaste bibliothèque standard fournit une gamme complète d'outils pour gérer divers aspects du développement backend, ce qui en fait un choix populaire parmi les développeurs pour la création d'application RESTful.



PostgreSQL est un système de gestion de base de données (SGBD) relationnelle et objet. Il est distribué sous une licence de type BSD, ce qui signifie qu'il est libre et gratuit à utiliser. PostgreSQL est reconnu pour sa robustesse, sa fiabilité et sa conformité aux standards ANSI SQL. Il offre des fonctionnalités avancées telles que la gestion de données géospatiales, la réPLICATION, et le support de nombreuses extensions. En comparaison avec d'autres SGBD, PostgreSQL est réputé pour sa stabilité et sa capacité à gérer des charges de travail complexes et volumineuses.

2.7.2.2 Environnement logiciel

Le tableau suivant présente les ressources matérielles exploitées dans le cadre de la réalisation de notre projet.

	Asus TUF	Dell E5570
Processeur	Ryzen 5 4500 H	Intel Core i5-6200U
Carte graphique	GTX 1650	(intégrée)
RAM	16 Go	8 Go

Tableau 8- Spécifications des environnements de développement

2.8. Conclusion

Tout au long de ce chapitre, les acteurs et les besoins du système ont été identifiés. Un diagramme global des cas d'utilisation a été élaboré pour visualiser les interactions principales. La structure du projet a été définie, ainsi qu'un diagramme de classe global pour organiser le code de manière efficace. Les choix architecturaux et les outils de développement ont été sélectionnés avec soin, avec Golang pour le backend et Angular pour le frontend. Ces décisions posent des bases solides pour la suite du projet, assurant une direction claire et cohérente.

Chapitre 3

Sprint 1 : Fonctionnalités de base de l'administration et de la gestion des utilisateurs



3.1 Introduction

Dans ce chapitre intitulé "Sprint 1 : fonctionnalités de base de l'administration et de la gestion des utilisateurs", nous nous penchons sur le premier sprint du projet. Nous débutons par l'exploration des objectifs clés de ce sprint ainsi que du backlog des fonctionnalités à implémenter. Ensuite, nous passons en revue en détail les différentes histoires utilisateurs qui orienteront le développement, avant d'entrer dans la phase de conception. Enfin, nous esquissons succinctement les différentes étapes de réalisation et de test, pour conclure sur les premiers résultats obtenus à l'issue de ce premier sprint.

3.2 Objectifs du sprint

L'objectif principal du sprint 1 est de déployer les fonctionnalités fondamentales permettant à l'administrateur de superviser les utilisateurs du système, tout en offrant à ces utilisateurs la possibilité de gérer leurs propres profils et leurs préférences.

3.3 Backlog du sprint

Dans ce tableau, nous présentons le backlog de notre premier sprint ainsi que l'estimation de l'effort de développement pour chaque fonctionnalité à réaliser.

ID	Fonctionnalité	Histoire utilisateur	Estimation
1	Authentification	En tant qu'administrateur, je veux me connecter à l'espace de l'administration afin de gérer mon espace de travail.	4 jours
2		En tant qu'utilisateur, je veux me connecter au système.	2 jours
3	Gestion des profils	En tant qu'utilisateur, je veux modifier mon profil.	2 jours
4	Gestion des utilisateurs	En tant qu'administrateur, je veux ajouter de nouveaux utilisateurs.	4 jours
5		En tant qu'administrateur, je veux modifier les informations des utilisateurs existants.	3 jours
6		En tant qu'administrateur, je veux supprimer des utilisateurs.	2 jours

ID	Fonctionnalité	Histoire utilisateur	Estimation
7	Gestion d'hashtag	En tant qu'utilisateur, je veux pouvoir créer un hashtag.	2 jours
8		En tant qu'utilisateur, je veux pouvoir consulter les hashtags à différents niveaux.	1 jours
9		En tant qu'utilisateur, je veux pouvoir modifier un hashtag.	2 jours
10		En tant qu'utilisateur, je veux pouvoir supprimer un hashtag.	1 jours

Tableau 9- Backlog du sprint 1

3.4 Conception du sprint

La phase de conception du sprint constitue une étape essentielle dans le processus de développement, où nous élaborons les fondations techniques et fonctionnelles de notre projet à travers différents artefacts tels que les diagrammes des cas d'utilisation, les descriptions textuelles des cas d'utilisation, les diagrammes de séquence et le diagramme de classes.

3.4.1 Diagramme des cas d'utilisation

La figure 12 ci-dessous présente le diagramme de cas d'utilisation de notre premier sprint.

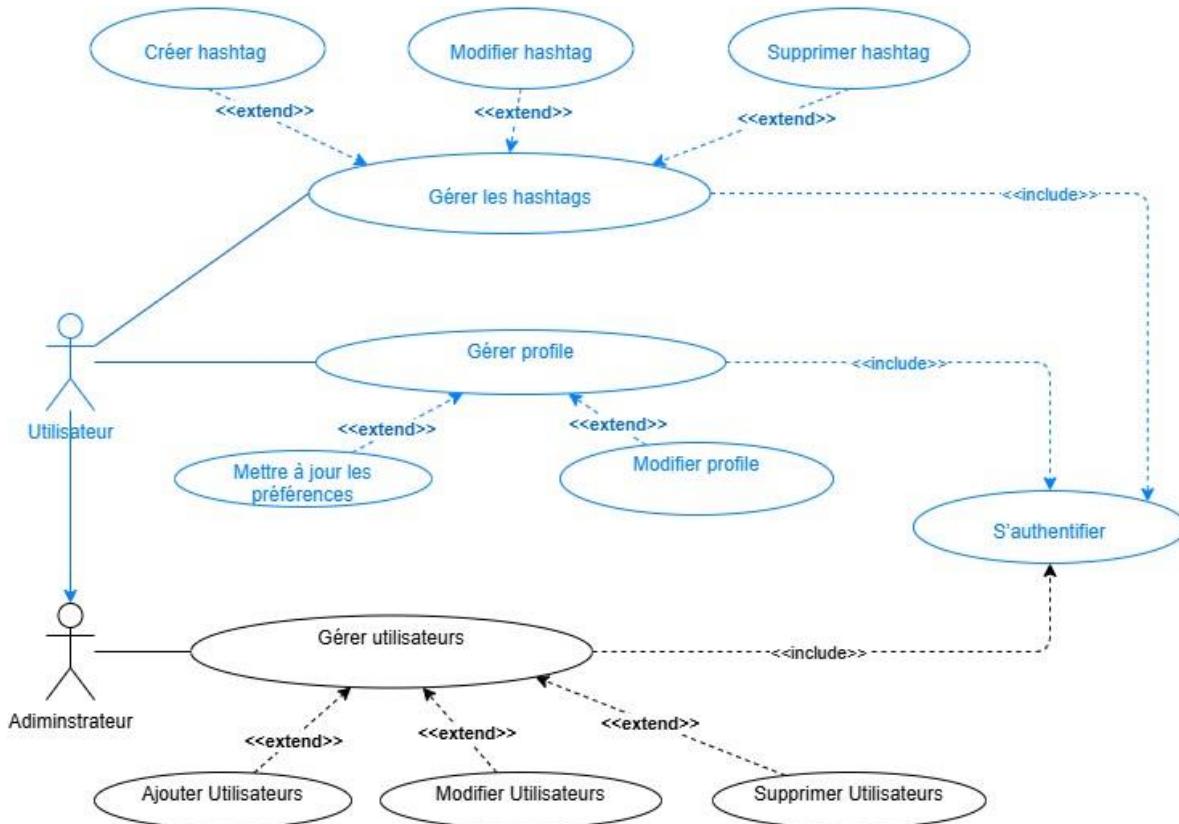


Figure 12- Diagramme des cas d'utilisation du premier sprint

3.4.2 Description textuelle des cas d'utilisation

Cette section se concentre sur l'explication détaillée de certains cas d'utilisation pour mieux comprendre notre système et ses interactions avec les acteurs. En simplifiant ces cas d'utilisation, nous avons pu fournir des descriptions détaillées des scénarios possibles, servant ainsi de base pour créer facilement les diagrammes de séquence système.

3.4.2.1 Description textuelle du cas d'utilisation « S'authentifier »

Cas d'utilisation : S'authentifier
Résumé : Ce cas d'utilisation permet à l'acteur de se connecter à la plateforme.
Acteur : Utilisateur, Admin
Pré-condition : L'utilisateur doit être enregistré dans le système.
Post-condition : L'utilisateur a accès aux fonctionnalités appropriées, qu'il soit administrateur ou utilisateur standard.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<ul style="list-style-type: none"> 01. L'utilisateur demande l'interface de connexion du système. 02. Le système affiche le formulaire d'authentification (email et mot de passe). 03. L'utilisateur saisit son e-mail et son mot de passe. 04. Le système vérifie la validité des identifiants fournis. 05. Le système ouvre l'espace de travail correspondant au rôle de l'utilisateur.
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
Dans le cas où les identifiants fournis ne sont pas valides, ce scénario démarre au point 04 du scénario nominal.
<ul style="list-style-type: none"> 05 : Le système informe l'utilisateur du type d'erreur , puis le scénario reprend au point 03 du scénario nominal.

Tableau 10- Description textuelle du cas d'utilisation « s'authentifier »

3.4.2.2 Description textuelle du cas d'utilisation « Modifier profile »

Cas d'utilisation : Modifier profile
Résumé : Ce cas d'utilisation permet à un acteur de modifier ses informations personnelles.
Acteur : Utilisateur, Admin
Pré-condition : L'utilisateur doit être authentifié et avoir accès à la fonctionnalité de modification de profil.
Post-condition : Les informations personnelles de l'utilisateur sont modifiées.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<p>01. L'utilisateur accède à la section "profil" de l'interface utilisateur.</p> <p>02. Le système affiche les champs de formulaire pré-remplis avec les informations actuelles de l'utilisateur.</p> <p>03. L'utilisateur effectue les modifications nécessaires dans les champs de formulaire.</p> <p>04. Le système vérifie la validité des données soumises.</p> <p>05. Le système enregistre les modifications apportées et affiche un message de confirmation à l'utilisateur.</p>
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
Dans le cas où les données sont invalides, ce scénario démarre au point 04 du scénario nominal.
05 le système affiche un message d'erreur indiquant les champs à corriger, puis le scenario reprend au point 03 du scénario nominal.

Tableau 11- Description textuelle du cas d'utilisation « Modifier profil »

3.4.2.3 Description textuelle du cas d'utilisation « Ajouter utilisateur »

Cas d'utilisation : Ajouter utilisateur
Résumé : Ce cas d'utilisation permet à un acteur d'ajouter un utilisateur à l'entreprise.
Acteur : Admin
Pré-condition : L'administrateur doit être authentifié et avoir accès à la fonctionnalité d'ajout d'utilisateurs.
Post-condition : L'administrateur a ajouté avec succès un nouvel utilisateur via son compte Gmail.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<p>01. L'administrateur accède à la fonctionnalité "Company" dans l'interface d'administration.</p> <p>02. Le système affiche un formulaire comprenant un champ pour l'adresse e-mail du nouvel utilisateur.</p> <p>03. L'administrateur saisit l'adresse e-mail du nouvel utilisateur.</p> <p>04. Le système vérifie la validité de l'adresse e-mail.</p> <p>05. Le système envoie une invitation à l'utilisateur via son compte Gmail.</p> <p>06. L'utilisateur accepte l'invitation et génère son mot de passe.</p> <p>07. L'utilisateur soumet son mot de passe via le lien d'invitation.</p> <p>08. Le système enregistre les informations de l'utilisateur et lui accorde les autorisations appropriées.</p> <p>09. Le système affiche un message de confirmation à l'administrateur indiquant que l'utilisateur a été ajouté avec succès.</p>
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
Dans le cas où l'adresse e-mail est invalide ou déjà associée à un compte, ce scénario démarre au point 04 du scénario nominal.
05 le système affiche un message d'erreur et invite l'administrateur à saisir une adresse e-mail différente., puis le scenario reprend au point 03 du scénario nominal.

Tableau 12- Description textuelle du cas d'utilisation « ajouter utilisateur »

3.4.3 Diagrammes de séquence

Dans cette section, nous nous sommes attelés à présenter quelques diagrammes de séquence. Ces diagrammes offrent une représentation visuelle des interactions entre les différents composants d'un système ou entre les acteurs et le système lui-même. En se concentrant sur la séquence chronologique des événements, ces diagrammes permettent de mieux comprendre le flux d'exécution des actions au sein du système. En explorant ces séquences d'actions, nous avons cherché à clarifier les processus impliqués dans divers scénarios d'utilisation, offrant ainsi une perspective plus approfondie sur le fonctionnement global du système.

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

Dans le diagramme ci-dessous, nous détaillons l'interaction entre l'utilisateur et les composants du système afin de pouvoir s'authentifier.

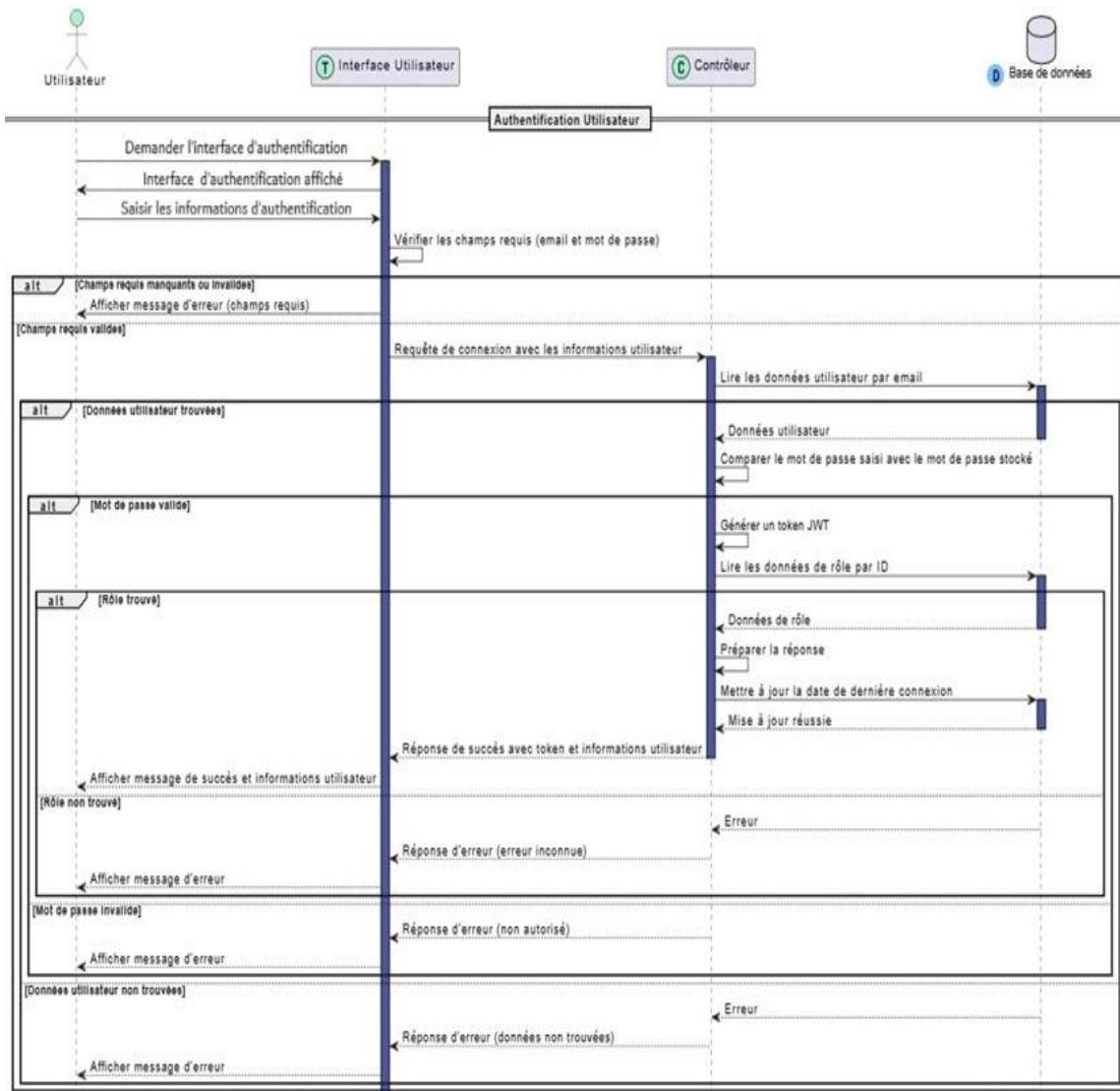


Figure 13- Diagramme de séquence « S'authentifier »

3.4.3.2 Diagramme de séquence du cas d'utilisation « Modifier profile »

Dans le diagramme présenté ci-dessous, nous éludions l'interaction spécifique entre l'utilisateur et les différentes composantes du système pour modifier les informations relatives à son profil.

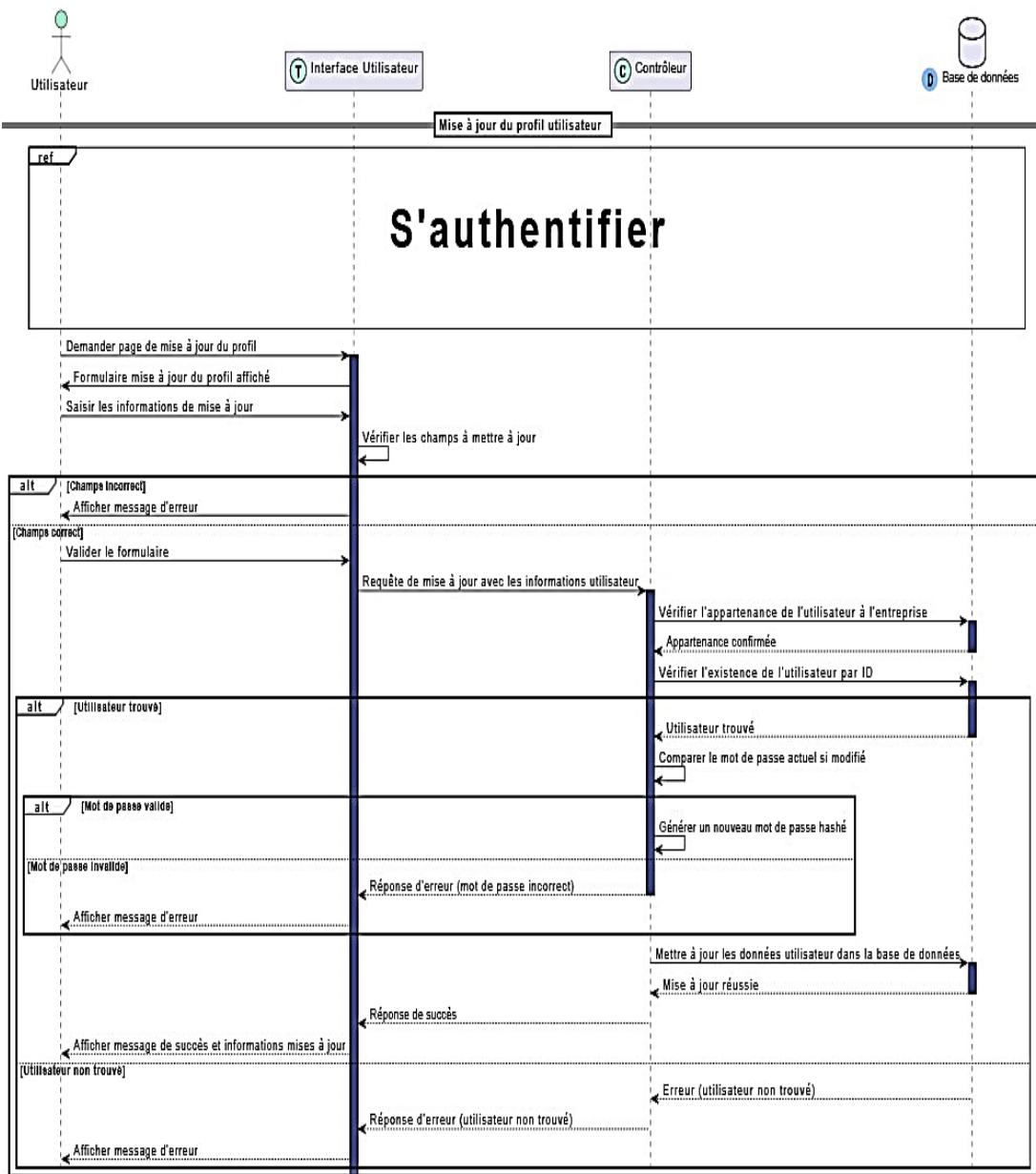


Figure 14- Diagramme de séquence « Modifier profil »

3.4.3.3 Diagramme de séquence du cas d'utilisation « ajouter utilisateur »

Dans le diagramme présenté ci-dessous, nous éludions l'interaction spécifique entre l'utilisateur et les différentes composantes du système, visant à permettre l'ajout d'un utilisateur.

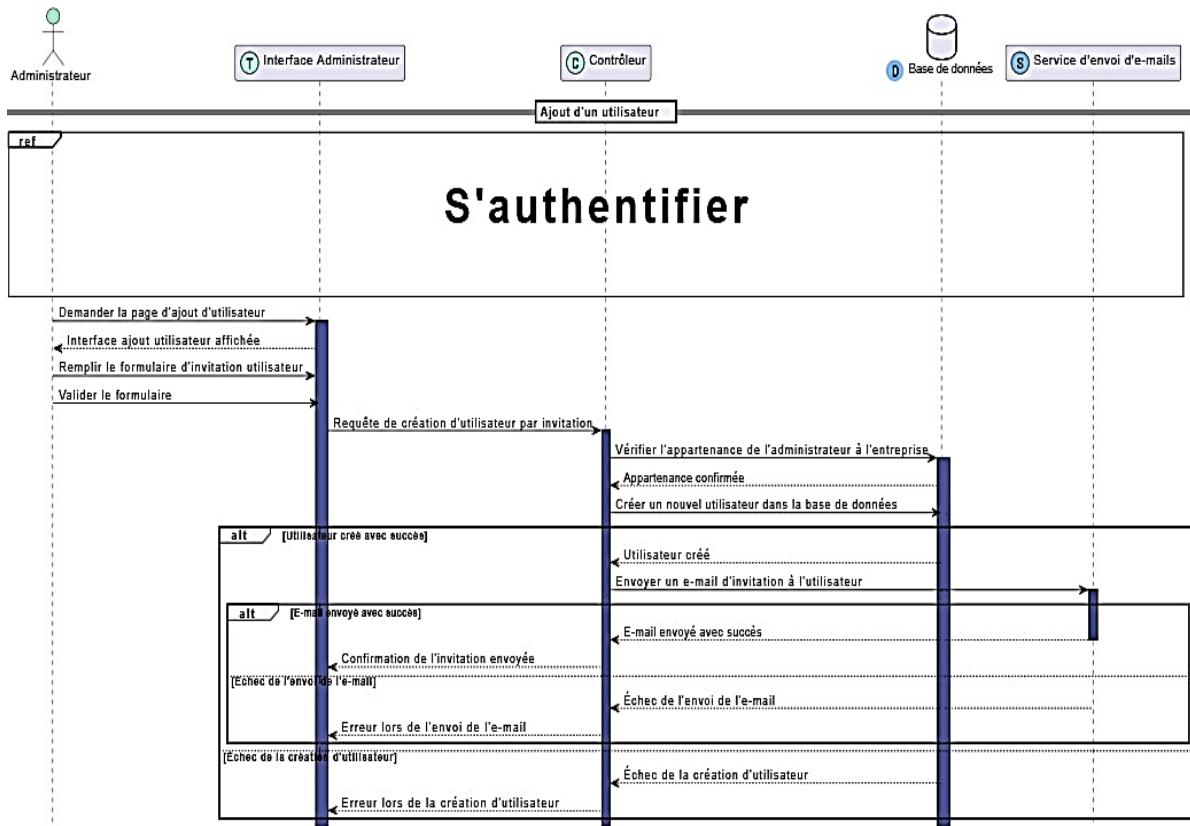


Figure 15- Diagramme de séquence « Ajouter utilisateur »

3.4.4 Diagramme de classes

Le diagramme de séquence précédemment exposé nous a offert un aperçu dynamique de notre système. Pour une présentation plus statique des entités et de leurs relations, nous avons consacré cette section à définir un diagramme faisant partie des diagrammes structurels d'UML le diagramme de classes, tel qu'illustré dans la figure ci-dessous :

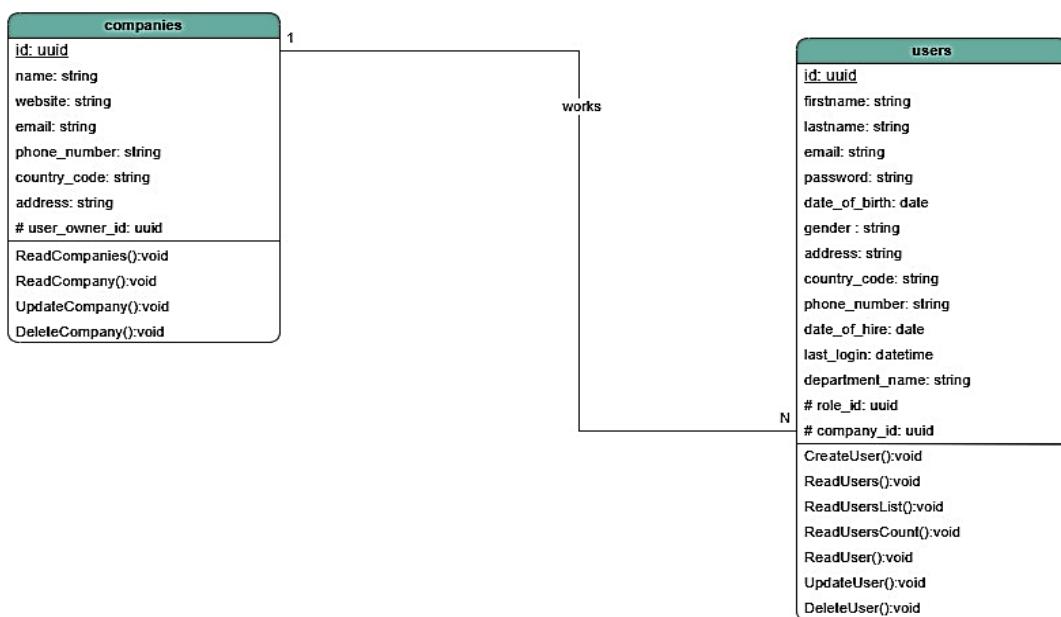


Figure 16-Diagramme de classes du Sprint 1

3.4 Réalisation

Durant cette étape, nous donnons vie aux concepts élaborés lors de la phase de conception. Notre but est de concrétiser les idées précédemment exposées. Nous nous focaliserons sur la modélisation des maquettes, le développement des interfaces utilisateurs et le déploiement des composants logiciels. Cette phase représente notre engagement à mettre en œuvre les fonctionnalités spécifiées dans le backlog du sprint.

3.4.1 Modélisation des maquettes graphiques

La création des maquettes d'IHM présente plusieurs avantages. Tout d'abord, elles permettent de concrétiser le fonctionnement futur de l'application lors des présentations au client. Ensuite, la collaboration avec le client dans la conception de l'interface garantit une réponse adéquate à ses attentes, renforçant ainsi la satisfaction client. De plus, la création de prototypes clarifie les besoins du produit et favorise une approche interactive dans le processus de développement. Dans la suite, nous présenterons les maquettes des histoires utilisateurs les plus significatives du premier sprint.

3.4.1.1 Prototype relatif à l'histoire utilisateur « S'authentifier »

Le prototype présenté ci-dessous correspond à l'histoire utilisateur « s'authentifier », elle répond à deux scénarios d'utilisation différents à savoir :

- En tant qu'administrateur, l'utilisateur souhaite se connecter à l'espace d'administration pour gérer son espace de travail.
- En tant qu'utilisateur standard, l'utilisateur souhaite se connecter au système."

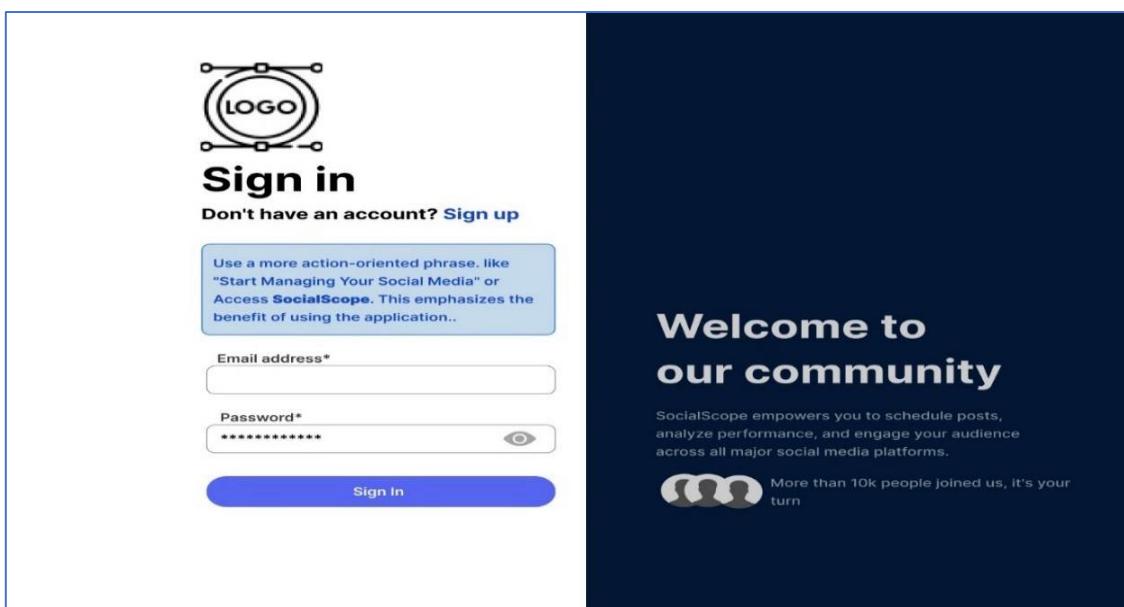


Figure 17- Prototype relatif à l'histoire utilisateur « S'authentification »

3.4.1.2 Prototype relatif à l'histoire utilisateur « modifier profil »

Le prototype présenté ci-dessous correspond à l'histoire utilisateur « modifier profil ». Il répond au scénario d'utilisation : En tant qu'utilisateur régulier, je peux modifier mon profil et mettre à jour mes informations personnelles.

Le prototype de l'interface 'Modifier profil' est un formulaire de modification de profil. Il comprend les champs suivants :

- Profile picture : Un champ pour charger une photo de profil avec un placeholder d'un utilisateur.
- User ID : Un champ de texte pour l'ID de l'utilisateur.
- Password : Un champ de mot de passe et un bouton 'Change Password'.
- First Name, Last Name, Email : Champs de texte pour le prénom, le nom et l'adresse email.
- Country : Un champ avec un placeholder 'Country-name' et un bouton 'Swap'.
- Date of Hire : Un champ de date pour la date d'embauche.
- Last Login : Un champ pour la dernière connexion.
- Company : Un champ de texte pour l'entreprise.
- Role : Un champ de texte pour le rôle.
- Update : Un bouton pour enregistrer les modifications.

Figure 18- Prototype relatif à l'histoire utilisateur « Modifier profil »

3.4.1.3 Prototype relatif à l'histoire utilisateur « Ajouter utilisateur »

Le prototype présenté ci-dessous correspond à l'histoire utilisateur « ajouter utilisateur ». Il répond au scénario d'utilisation : En tant qu'administrateur, je peux ajouter d'autres utilisateurs.

Le prototype de l'interface 'Ajouter utilisateur' permet d'inviter de nouveaux membres à l'équipe. Il comprend les étapes suivantes :

- Add team member** : Titre de la section.
- Invite team member** : Section où l'utilisateur peut inviter un membre existant.
- First Name**, **Last Name**, **Email** : Champs pour saisir les détails du nouvel utilisateur.
- What permissions should the new team member have?** : Section pour définir les permissions du nouvel utilisateur.
- Invitation** et **Cancel** : Boutons pour envoyer l'invitation ou annuler.
- Team member** : Section qui affiche les membres déjà ajoutés avec des options de suppression ('Swap' et trois points).

Figure 19- Prototype relatif à l'histoire utilisateur « Ajouter utilisateur »

3.4.2 Développement des interfaces utilisateurs

Nous allons s'intéresser dans cette partie à présenter quelques interfaces utilisateurs de notre premier sprint.

3.4.2.1 Interface graphique relative à l'histoire utilisateur « S'authentification »

Cette interface sert à l'authentification, permettant à l'utilisateur (qu'il soit administrateur ou utilisateur standard) d'entrer ses identifiants (nom d'utilisateur et mot de passe) afin de se connecter à l'application.

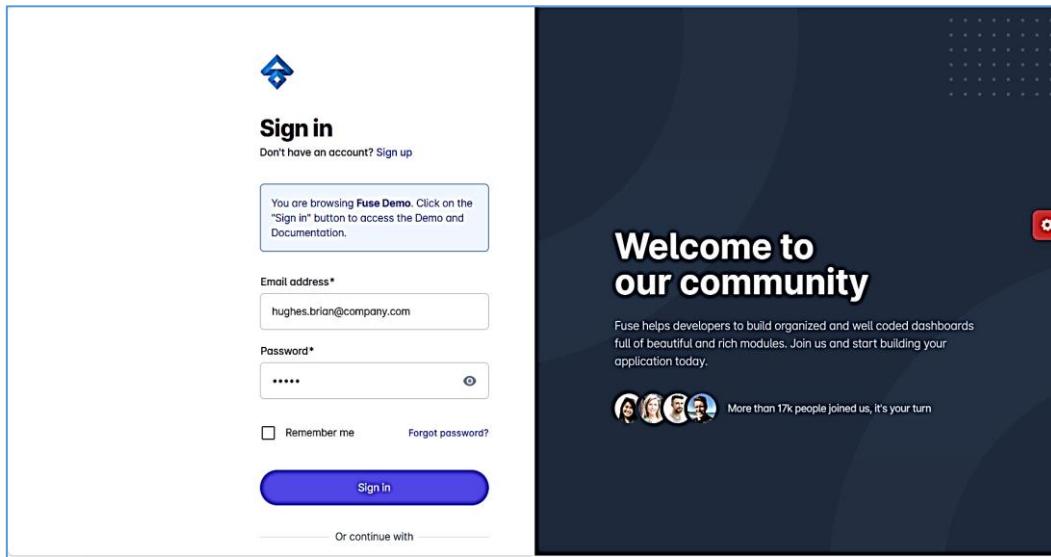


Figure 20- Interface graphique relative à l'histoire utilisateur « S'authentification »

3.4.2.3 Interface graphique relative à l'histoire utilisateur « Modifier profil »

Cette interface est destinée à la modification du profil, permettant à l'utilisateur (administrateur ou utilisateur standard) de mettre à jour ses informations personnelles en fonction de ses besoins.

The image shows a detailed view of a user profile modification form. At the top, there are fields for 'Profile picture' (with a placeholder image of a man in a suit), 'User ID' (containing '3cebb892-e677-493e-915b-2c5ea225e258'), and 'Password' (with a 'Change Password' link). Below these are sections for personal information: 'First name' (empty), 'Last name' (empty), and 'Email' (empty). There are also fields for 'Country' (set to 'Tunisia'), 'Date of Hire' (placeholder 'mm/dd/yyyy'), and 'Last Login' (empty). Further down are fields for 'Company' (placeholder 'undefined') and 'Role' (set to 'Manager'). At the bottom right is a blue 'Update' button.

Figure 21- Interface graphique relative à l'histoire utilisateur « Modifier profil »

3.4.2.4 Interface graphique relative à l'histoire utilisateur « Ajouter utilisateur »

Cette interface est spécifiquement dédiée à l'ajout d'utilisateurs, une fonctionnalité réservée à l'administrateur. Elle offre à l'administrateur la possibilité de créer de nouveaux comptes utilisateur en saisissant les informations nécessaires et en définissant les autorisations appropriées.

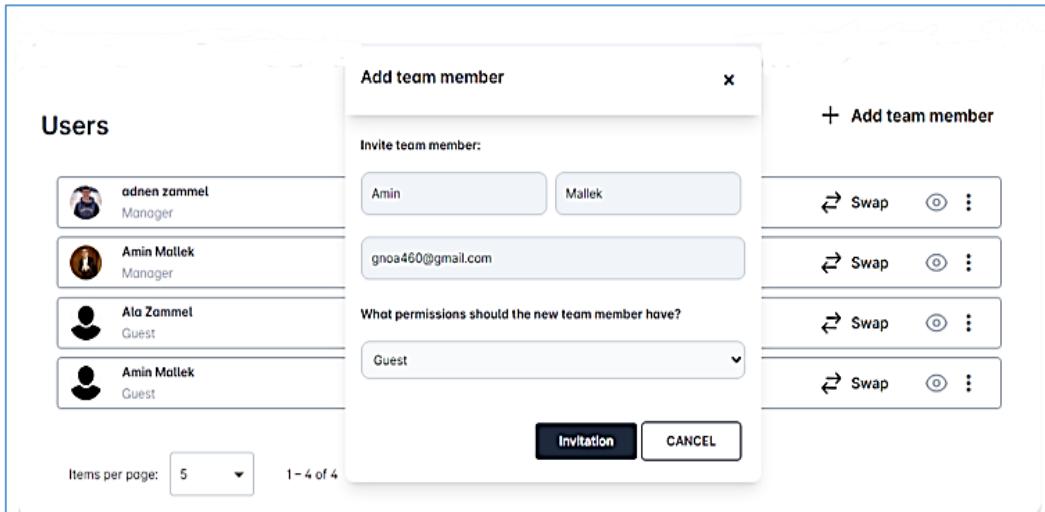


Figure 22- Interface graphique relative à l'histoire utilisateur « Ajouter utilisateur »

3.4.3 Déploiement des composants logiciels

Angular et **Golang** forment une alliance robuste pour le développement d'applications web modernes. Ensemble, ils créent un environnement propice à un développement efficace, maintenable, testable et facilement déployable. La combinaison de ces deux framework divise l'architecture de notre application en trois couches principales :

- **Frontend** : L'application Angular assure l'interaction utilisateur, fournissant une expérience dynamique et réactive.
- **Backend** : Le serveur **Golang** agit comme une passerelle entre la base de données PostgreSQL et l'application Angular. Il offre des services REST sans état, facilitant ainsi la communication entre le frontend et la base de données, et garantissant un échange fluide de données. Pour cela, le backend utilise les frameworks **Gin** et **Gorm**. Gin est un framework web pour Golang qui gère le routage et le traitement des requêtes HTTP, tandis que Gorm est un ORM simplifiant l'interaction avec les bases de données.
- **Base de données** : PostgreSQL stocke de manière persistante les données de l'application. Elle constitue le référentiel central pour les entités manipulées par l'application Angular via le backend Golang.

La communication entre les différents composants se déroule comme suit :

- Le serveur Golang fournit une API REST à l'application Angular.
- Ce serveur communique avec la base de données PostgreSQL à l'aide de la bibliothèque de pilotes PostgreSQL en Golang.
- L'application Angular envoie des requêtes HTTP au backend Golang et reçoit des réponses grâce au client HTTP intégré d'Angular.

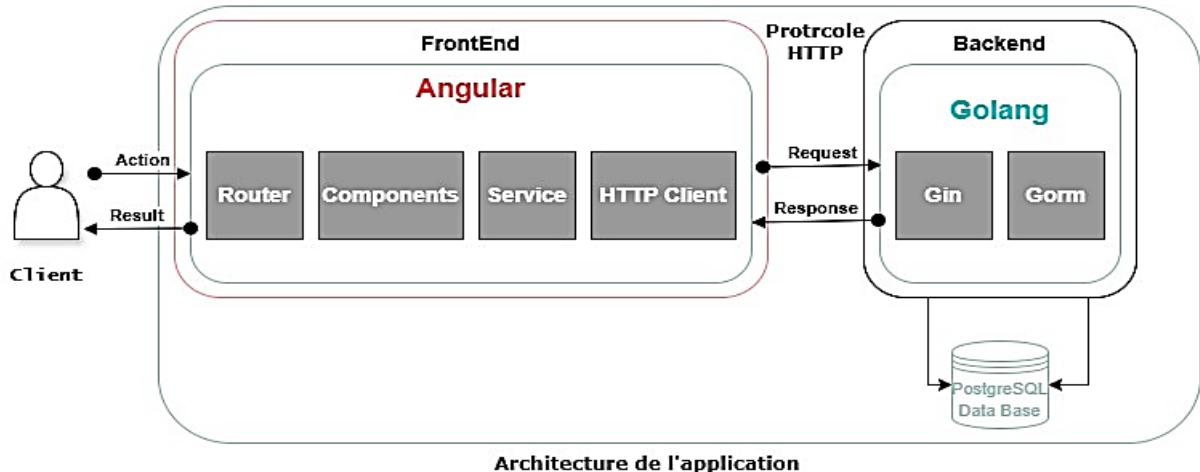


Figure 23- Architecture basique de l'application

3.5 Test fonctionnel

Après la réalisation du premier sprint, nous entamons désormais la présentation des tâches de finalisation. Cette phase vise à évaluer notre avancement et déterminer si nous avons atteint les objectifs fixés, tout en nous préparant pour le prochain sprint. Dans cette partie, nous mettons l'accent sur quelques scénarios de tests fonctionnels pertinents du premier sprint. Nous les présentons dans le tableau suivant pour une meilleure lisibilité et compréhension :

Cas de test	Démarche	Comportement attendu	Résultat
Test d'authentification	Après l'accès à l'interface d'authentification, l'utilisateur entre ses coordonnées et valide.	Redirection vers la page de tableau de bord si l'utilisateur existe. Sinon un message d'erreur s'affiche.	Conforme
Test de modification de profil	Après l'accès à l'interface de modification de profil, l'utilisateur met à jour les champs nécessaires et clique sur le bouton "Update".	Le client envoie des données au serveur pour mettre à jour son profil. Le serveur valide et enregistre ces données. Si aucune erreur ne survient, le client reçoit un message de succès, si non reçoit un message d'erreur.	Conforme

Cas de test	Démarche	Comportement attendu	Résultat
Test d'ajout d'utilisateur	Après l'accès à l'interface d'ajout d'utilisateur, l'utilisateur saisir les données et clique sur le bouton "invitation".	Après soumission des informations, le système envoie une invitation à l'adresse e-mail spécifiée. Si l'envoi réussit, un message de succès est affiché. Sinon, un message d'erreur est affiché.	Conforme

3.6 Conclusion

Dans ce chapitre, nous avons abordé la spécification des besoins, la conception avec des diagrammes de séquences détaillés et un diagramme de classes, ainsi que la réalisation des histoires utilisateurs pour le premier sprint. Après avoir testé et validé ces fonctionnalités avec le Scrum Master et le Product Owner lors de la revue du sprint, nous sommes prêts à passer à la prochaine étape : le sprint 2.

Chapitre 4

Sprint 2 : Fonctionnalités de gestion des campagnes et des tâches associées.



4.1 Introduction

Dans ce chapitre intitulé "Sprint 2 : Fonctionnalités de gestion des campagnes et des tâches associées", nous plongerons dans le deuxième sprint du projet, centré sur la gestion des campagnes et des tâches associées. Nous débutons par la définition des objectifs spécifiques de ce sprint et l'examen du backlog des fonctionnalités à mettre en œuvre. Ensuite, nous explorons en détail les histoires utilisateurs associées à ce sprint, qui serviront de guide tout au long du processus de développement. Puis, nous procéderons à la conception approfondie des fonctionnalités identifiées, en accordant une attention particulière aux besoins et aux exigences des utilisateurs. Enfin, nous détaillerons les étapes de réalisation et de test prévues pour ce sprint, avant de conclure sur les résultats escomptés à l'issue de cette phase du projet.

4.2 Objectifs du sprint

Ce sprint vise à mettre en place les fonctionnalités fondamentales pour la gestion des campagnes et des tâches associées. Cela inclut la création, la consultation, la modification et la suppression des campagnes et des tâches, ainsi que la fonctionnalité de messagerie pour les utilisateurs de campagne afin d'assurer une meilleure coordination.

4.3 Backlog du sprint

Pour ce deuxième sprint, nous avons préparé un tableau Sprint Backlog des tâches clés à réaliser. Chaque fonctionnalité est associée à une estimation de l'effort de développement, permettant la planification et la priorisation des activités de sprint.

ID	Fonctionnalité	Histoire utilisateur	Estimation
1	Gestion des campagnes	En tant qu'utilisateur, je veux pouvoir créer une campagne.	4 jours
2		En tant qu'utilisateur, je veux pouvoir consulter ma liste de campagnes.	2 jours
3		En tant qu'utilisateur, je veux pouvoir modifier une campagne.	3 jours
4		En tant qu'utilisateur, je veux pouvoir supprimer une campagne.	2 jours
5		En tant qu'utilisateur, je veux pouvoir échanger des messages avec les autres utilisateurs de campagne.	3 jours

ID	Fonctionnalité	Histoire utilisateur	Estimation
6	Gestion des tâches	En tant qu'administrateur, je veux pouvoir créer une tâche avec une durée spécifiée.	4 jours
7		En tant qu'utilisateur, je veux pouvoir consulter les tâches à différents niveaux.	2 jours
8		En tant qu'administrateur, je veux pouvoir modifier une tâche.	3 jours
9		En tant qu'administrateur, je veux pouvoir supprimer une tâche.	2 jours
10		En tant qu'utilisateur, je veux pouvoir échanger des messages pour une tâche donnée.	2 jours

Tableau 13- Backlog du sprint 2

4.4 Conception du sprint

Dans cette partie, nous nous concentrons sur la présentation de la conception de notre deuxième sprint.

4.4.1 Diagramme des cas d'utilisation

La figure 23 ci-dessous illustre le diagramme de cas d'utilisation de notre deuxième sprint.

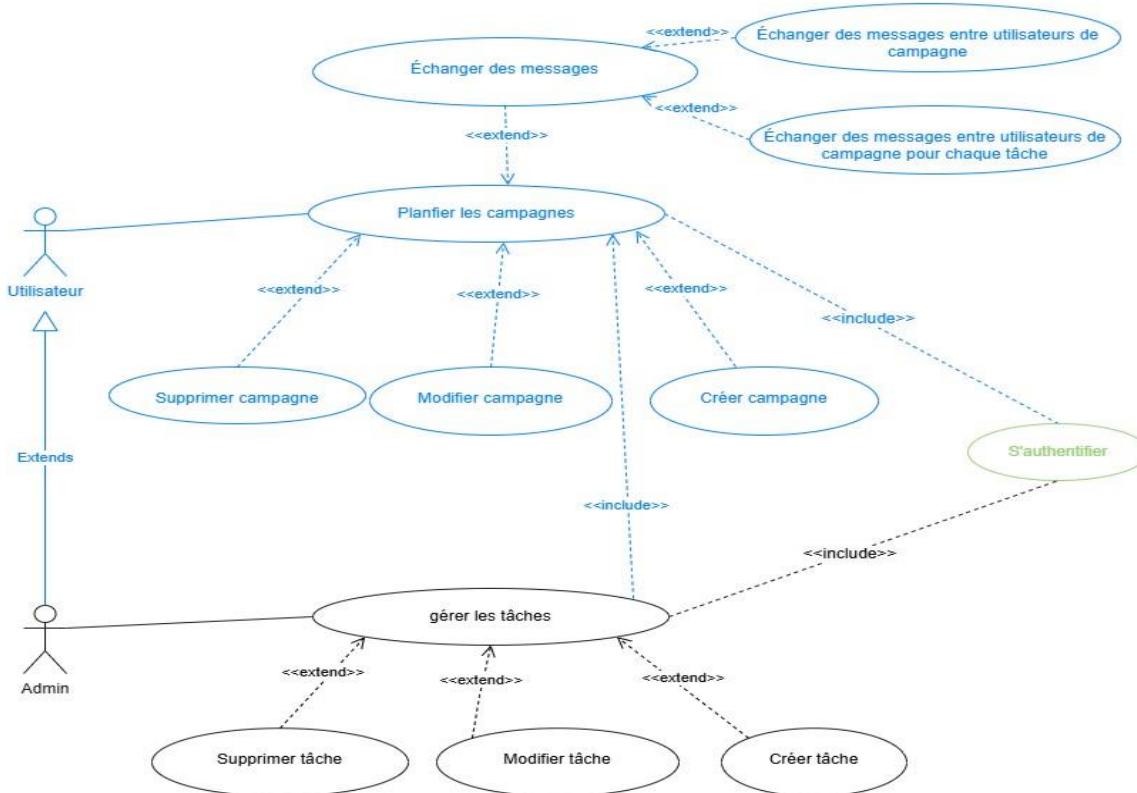


Figure 24- Diagramme des cas d'utilisation du deuxième sprint

4.4.2 Description textuelle des cas d'utilisation

Dans cette section, nous approfondirons par l'explication sur des cas d'utilisation pour le deuxième sprint, offrant ainsi une compréhension approfondie du système et de ses interactions avec les acteurs impliqués. En décomposant ces cas d'utilisation de manière détaillée, nous fournissons des descriptions exhaustives des différents scénarios envisageables, jetant ainsi les bases nécessaires à la création aisée des diagrammes de séquence système.

4.4.2.1 Description textuelle du cas d'utilisation « Créer campagne »

Cas d'utilisation : Créer campagne
Résumé : Ce cas d'utilisation permet à l'acteur de créer une campagne.
Acteur : Utilisateur, Admin
Pré-condition : L'utilisateur doit être authentifié et avoir accès à la fonctionnalité de création campagne.
Post-condition : La campagne est ajoutée dans le système et affichée dans la liste des campagnes des tâches déjà créées.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<ul style="list-style-type: none"> 01. L'utilisateur sollicite l'interface de création de campagne. 02. Le système affiche le formulaire de création de campagne. 03. L'utilisateur saisit les informations requises. 04. Le système vérifie la validité des informations fournies. 05. La campagne est ajoutée au système et apparaît dans la liste des campagnes.
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
En cas d'informations incorrectes ou manquantes, ce scénario démarre au point 04 du scénario nominal.
<ul style="list-style-type: none"> 05 : Le système informe l'utilisateur du type d'erreur, puis le scénario reprend au point 03 du scénario nominal.

Tableau 14- Description textuelle du cas d'utilisation « Créer campagne »

4.4.2.2 Description textuelle du cas d'utilisation « Créer Tâche »

Cas d'utilisation : Créer tâche
Résumé : Ce cas d'utilisation permet à l'acteur de créer une tâche.
Acteur : Admin
Pré-condition : L'administrateur doit être authentifié et avoir accès à la fonctionnalité « Créer tâche ».
Post-condition : La tâche est créée dans le système et affichée dans la liste des tâches des tâches déjà créées.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<ul style="list-style-type: none"> 01. L'administrateur sollicite l'interface de création de tâche. 02. Le système affiche le formulaire de création de tâche. 03. L'administrateur saisit les informations dans le formulaire. 04. Le système vérifie la validité des informations de la tâche. 05. La nouvelle tâche est créée dans le système et apparaît dans la liste des tâches dans l'interface de campagne.
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
En cas d'informations incorrectes ou manquantes, ce scénario démarre au point 04 du scénario nominal.
<ul style="list-style-type: none"> 05 : Le système informe l'administrateur du type d'erreur, puis le scénario reprend au point 03 du scénario nominal.

Tableau 15- Description textuelle du cas d'utilisation « Créer tâche »

4.4.2.2 Description textuelle du cas d'utilisation « Modifier tâche »

Cas d'utilisation : Modifier tâche
Résumé : Ce cas d'utilisation permet à l'acteur de modifier une tâche.
Acteur : Admin
Pré-condition : L'administrateur doit être authentifié et avoir accès à la fonctionnalité « Modifier tâche ».
Post-condition : La tâche a été modifiée avec succès et apparaît dans la liste des tâches déjà créées.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<ul style="list-style-type: none"> 01. L'administrateur accède à l'interface de campagne sélectionnée. 02. Le système affiche une liste des tâches créées dans la campagne sélectionnée. 03. L'administrateur sélectionne la tâche qu'il souhaite modifier. 04. Le système affiche un formulaire pré-rempli avec les informations actuelles pour la tâche sélectionnée. 05. L'administrateur effectue les modifications nécessaires dans les champs de formulaire. 06. Le système vérifie la validité des données soumises. 07. Le système enregistre les modifications et affiche à nouveau la tâche dans la liste.
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
En cas d'informations incorrectes ou manquantes, ce scénario démarre au point 06 du scénario nominal.
<ul style="list-style-type: none"> 07. Le système bloque la modification de la tâche, puis le scénario reprend au point 05 du scénario nominal.

Tableau 16- Description textuelle du cas d'utilisation « Modifier tâche »

4.4.3 Diagrammes de séquence

Dans cette section, nous avons créé des diagrammes de séquence pour illustrer comment les différentes parties du système interagissent dans quelques cas d'utilisation du sprint 2. Ces schémas décrivent l'ordre des actions, ce qui facilite la compréhension du fonctionnement du système. Notre objectif était de clarifier les processus dans différentes situations, offrant ainsi une meilleure vue d'ensemble du système.

4.4.1.1 Diagramme de séquence du cas d'utilisation « Créez campagne »

Dans le diagramme présenté ci-dessous, nous éludions l'interaction spécifique entre l'utilisateur et les différentes composantes du système, visant à permettre la création d'une campagne.

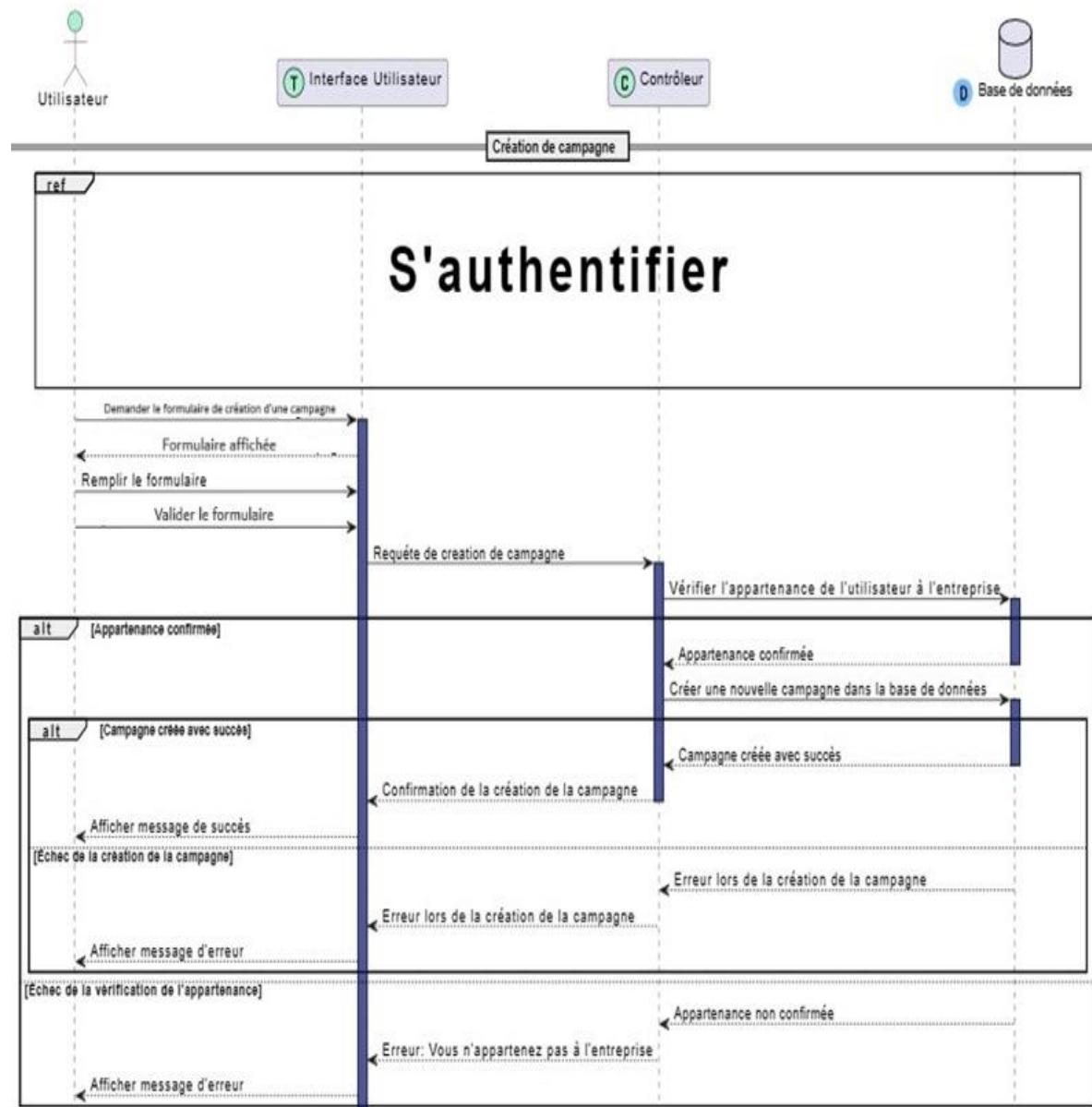


Figure 25- Diagramme de séquence « Créez campagne »

4.4.3.2 Diagramme de séquence du cas d'utilisation « Crée tâche »

Dans le diagramme suivant, nous éludions l'interaction spécifique entre l'administrateur et les différentes composantes du système afin de créer une tâche donnée.

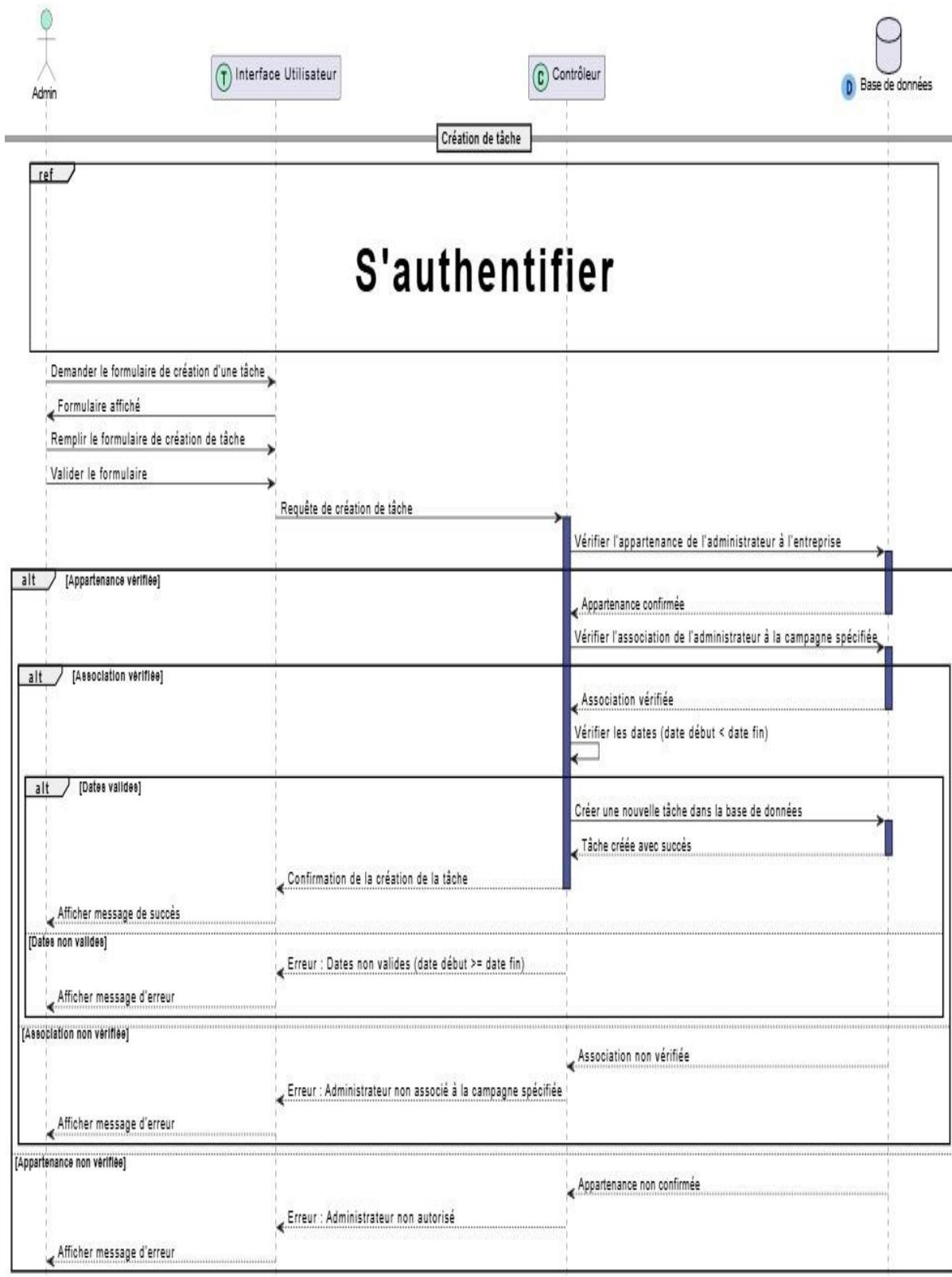


Figure 26- Diagramme de séquence «Créer tâche»

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

Dans le diagramme suivant, nous éludions l'interaction spécifique entre l'administrateur et les différentes composantes du système afin de modifier une tâche donnée.

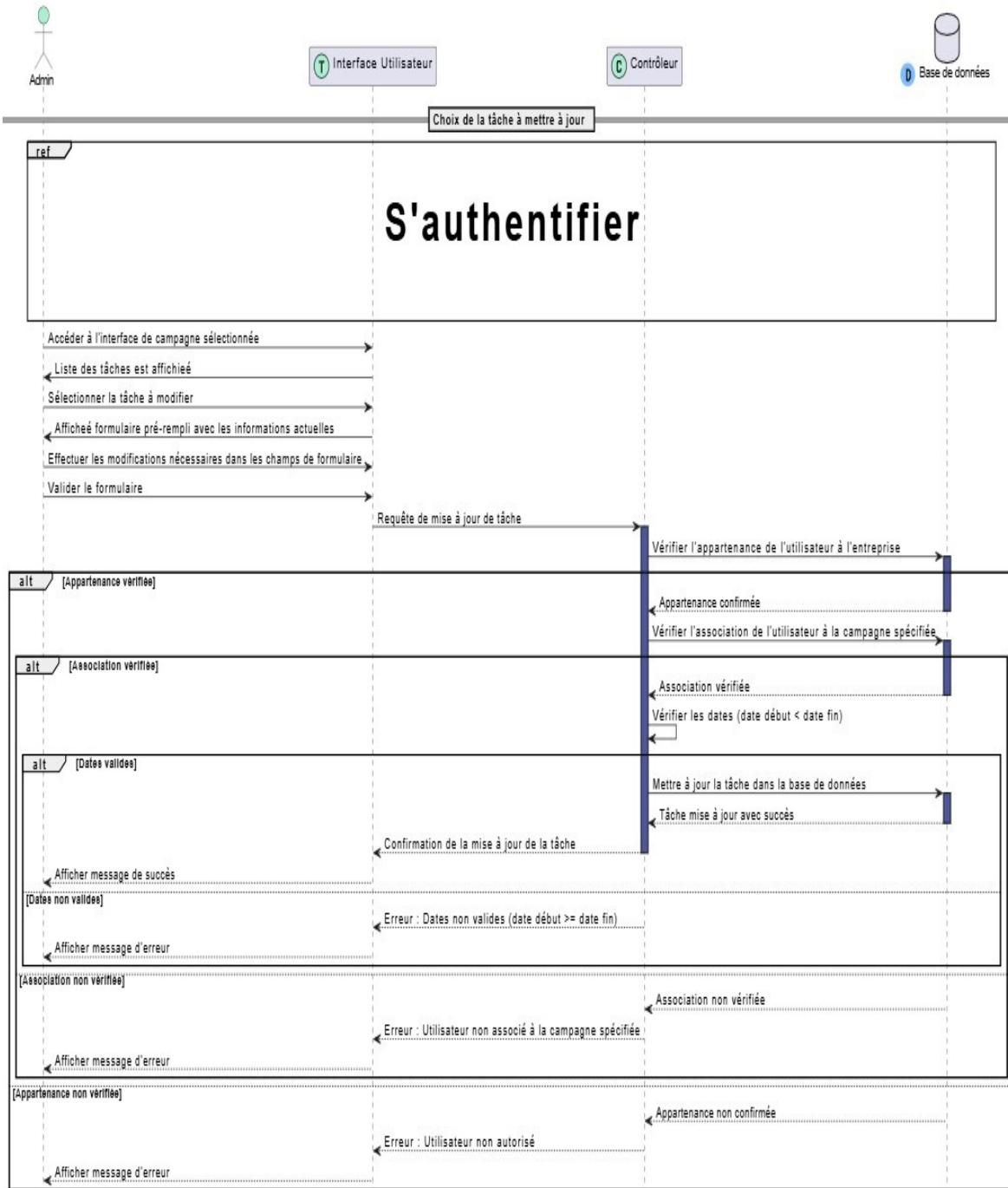


Figure 27- Diagramme de séquence «Modifier tâche»

4.4.4 Diagramme de classes

Après la présentation de quelques diagrammes de séquence du sprint 2, mettant en lumière les interactions dynamiques entre les acteurs et le système dans des contextes spécifiques, nous avons complété cette vision dynamique par la conception d'un diagramme de classes. Ce dernier offre une représentation structurée des entités et de leurs relations dans ce sprint, offrant ainsi

une perspective détaillée sur l'organisation interne du système, comme illustré dans la figure ci-dessous :

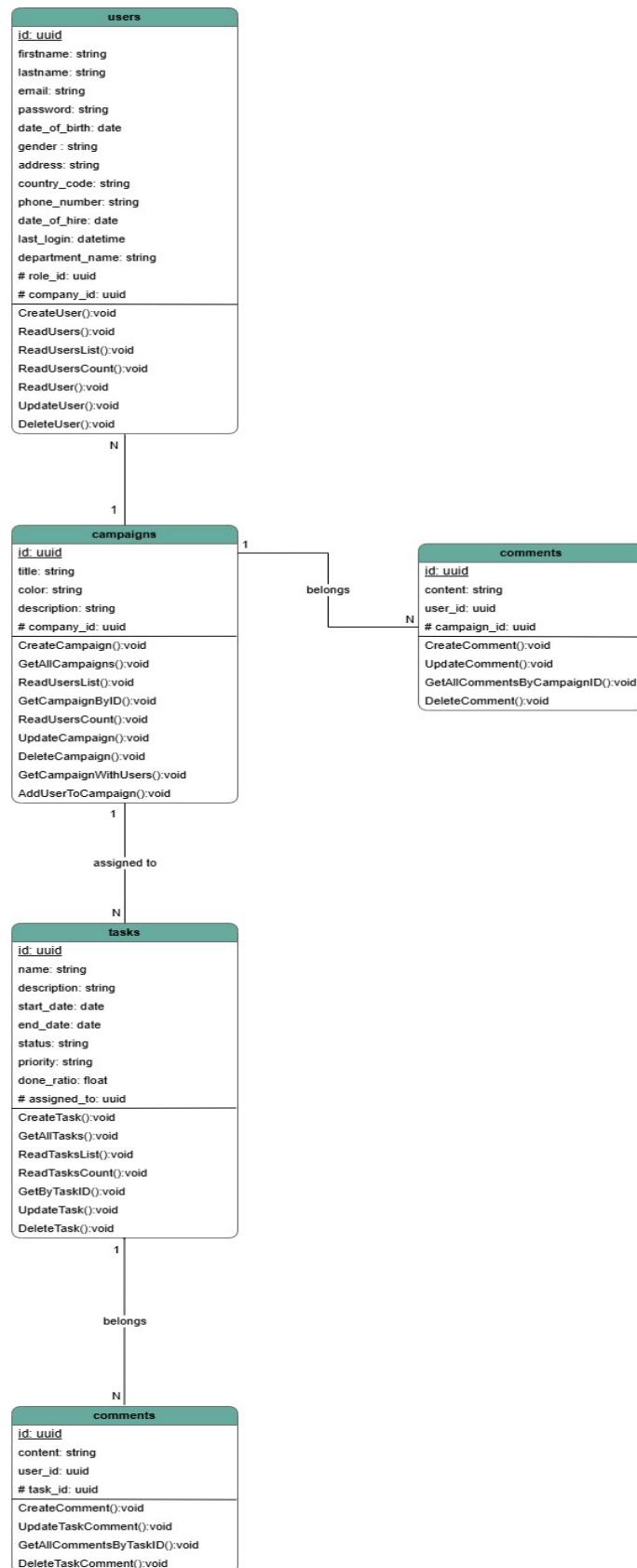


Figure 28- Diagramme de classes du Sprint 2

4.5 Réalisation

Dans cette phase, nous concrétisons nos idées en passant de la conception à la réalisation concrète. Notre objectif est de transformer les concepts discutés précédemment en actions tangibles. Nous mettrons particulièrement l'accent sur la création visuelle des maquettes et la conception des interfaces utilisateurs. Cette étape témoigne de notre engagement à mettre en œuvre les fonctionnalités définies dans le backlog du sprint.

4.5.1 Modélisation des maquettes graphiques

La création de maquettes d'interface utilisateur offre de nombreux avantages. En premier lieu, cela permet au client de visualiser l'apparence de l'application. En collaborant avec lui pour ajuster l'interface, on s'assure de répondre adéquatement à ses besoins. Les maquettes contribuent à clarifier les attentes du produit et favorisent la collaboration. Dans cette perspective, voici quelques exemples de maquettes graphiques illustrant les histoires utilisateur du deuxième sprint.

4.5.1.1 Prototype relatif à l'histoire utilisateur « Créer campagne »

Le prototype présenté ci-dessous correspond à l'histoire utilisateur « Créer campagne ». Il répond au scénario d'utilisation : En tant qu'utilisateur, je veux pouvoir créer une campagne.

The prototype is a wireframe for a 'campaigns - Add New' screen. It features a blue header bar with the title 'campaigns - Add New' and a close button ('X'). Below the header are three input fields: 'Color*' with a red color swatch, 'Description' with a text area containing two double slashes ('//'), and 'Title*' with an empty text area. At the bottom is a grey 'Add' button.

Figure 29- Prototype relatif à l'histoire utilisateur « Créer campagne »

4.5.1.2 Prototypes relatifs à l'histoire utilisateur « Créer tâche »

Le prototype présenté ci-dessous correspond à l'histoire utilisateur « Créer tâche ». Il répond au scénario d'utilisation : En tant qu'administrateur, je veux pouvoir créer une tâche avec une durée spécifiée.

Tasks

Provide feedback
 Change image
 Change text
 Other

What is it about? (optional)

Date start Date end

To whom

Priority

Invite Guest

Create Task

Figure 30- Prototype relatif à l 'histoire utilisateur « Créer tâche »

4.5.1.3 Prototype relatif à l'histoire utilisateur « Modifier tâche »

Le prototype présenté ci-dessous correspond à l'histoire utilisateur « modifier tâche ». Il répond au scénario d'utilisation : En tant qu'administrateur, je veux pouvoir modifier une tâche.

Tasks

Create New Task

Provide feedback Deadline: 4/20/24 to 4/30/24

Provide feedback Deadline: 4/20/24 to 4/30/24 Assigned To Adnen Zammel

Provide feedback Deadline: 4/20/24 to 4/30/24 Assigned To Adnen Zammel

Provide feedback Deadline: 4/20/24 to 4/30/24 Assigned To Adnen Zammel

Update
Delete

Previous Next

Tasks

Create New Task

Provide feedback
 Change image
 Change text
 Other

What is it about? (optional)

Date start Date end

To whom

Priority

Invite Guest

Update Task

Figure 31- Prototype relatif à l 'histoire utilisateur « Modifier tâche »

4.5.2 Développement des interfaces utilisateurs

Dans cette section, nous nous concentrerons sur la présentation des interfaces utilisateur conçues pour notre deuxième sprint.

4.5.2.1 Interface graphique relative à l'histoire utilisateur «Créer campagne.»

Cette interface est destinée à la création d'une campagne, permettant à l'utilisateur (administrateur ou utilisateur standard) de fournir les informations nécessaires pour la création d'une campagne.

L'interface 'campaigns - Add new' est une fenêtre de saisie de données. Elle comporte un champ 'Color*' avec un curseur de couleur, un champ 'Description' avec un éditeur de texte, un champ 'Title*' avec un éditeur de texte et un bouton 'Add' en bas à droite.

Figure 32– Interface graphique relatif à l'histoire utilisateur «Créer campagne»

4.5.2.2 Interface graphique relative à l'histoire utilisateur «Créer tâche»

Cette interface est exclusivement conçue pour créer une nouvelle tâche, fonctionnalité réservée à l'administrateur. Il offre à l'administrateur la possibilité de créer de nouvelles tâches assignées à un utilisateur en saisissant les informations requises et en définissant la durée de la tâche.

L'interface 'Create new task' est divisée en deux panneaux. Le panneau gauche, intitulé 'Tasks', contient une liste déroulante d'options : 'Provide feedback', 'Change image', 'Change text' et 'Other'. En dessous, il y a un champ 'What is it about? (optional)' avec un éditeur de texte et des champs 'Date start' et 'Date end' avec curseurs de date. Le panneau droit, également intitulé 'Create new task', contient des champs pour 'mm/dd/yyyy' avec calendriers, 'To whom' avec un champ déroulant, 'Priority' avec un champ déroulant, un bouton '+ Invite Guest' et un bouton 'Create Task'.

Figure 33- Interface graphique relatif à l'histoire utilisateur «Créer tâche»

4.5.2.3 Interface graphique relative à l'histoire utilisateur « Modifier tâche »

Cette interface est spécialement conçue pour la modification des tâches, offrant à l'administrateur la possibilité de mettre à jour les informations relatives à chaque tâche existante dans la campagne.

The figure illustrates the graphical interface for the 'Modifier tâche' (Edit Task) user story, showing three distinct views:

- Top View (List of Tasks):** A card-based list of five tasks. Each card includes a small icon, the task name, its deadline range, the assignee, and a set of actions (Edit, Delete). A dashed arrow points from this view down to the 'Create new task' form.
- Middle View (Task Detail):** A detailed view of a specific task titled 'Provide feedback'. It shows the deadline (5/20/24 to 5/30/24), assignee (Mohamed Amin Mallek), and a checkbox indicating it is completed. Below this, there's a 'Description' section with a rich-text editor containing the text 'create rapport'.
- Bottom View (Create New Task):** A form for creating a new task. It includes fields for 'To whom' (set to 'Mallek Mohamed Amin'), 'Priority' (set to 'Medium'), and two date pickers for 'Date start' (5/20/2024) and 'Date end' (5/30/2024). At the bottom are 'Invite Guest' and 'Update Task' buttons.

Figure 34- Interface graphique relatif à l'histoire utilisateur de « Modifier tâche »

4.6 Test fonctionnel

Après avoir terminé le deuxième sprint, nous abordons maintenant la phase de finalisation. Cette étape consiste à évaluer notre progression, vérifier si les objectifs ont été atteints, et nous préparer pour le sprint suivant. Nous mettrons en lumière quelques scénarios de tests

fonctionnels pertinents, présentés dans le tableau ci-dessous pour une meilleure lisibilité et compréhension :

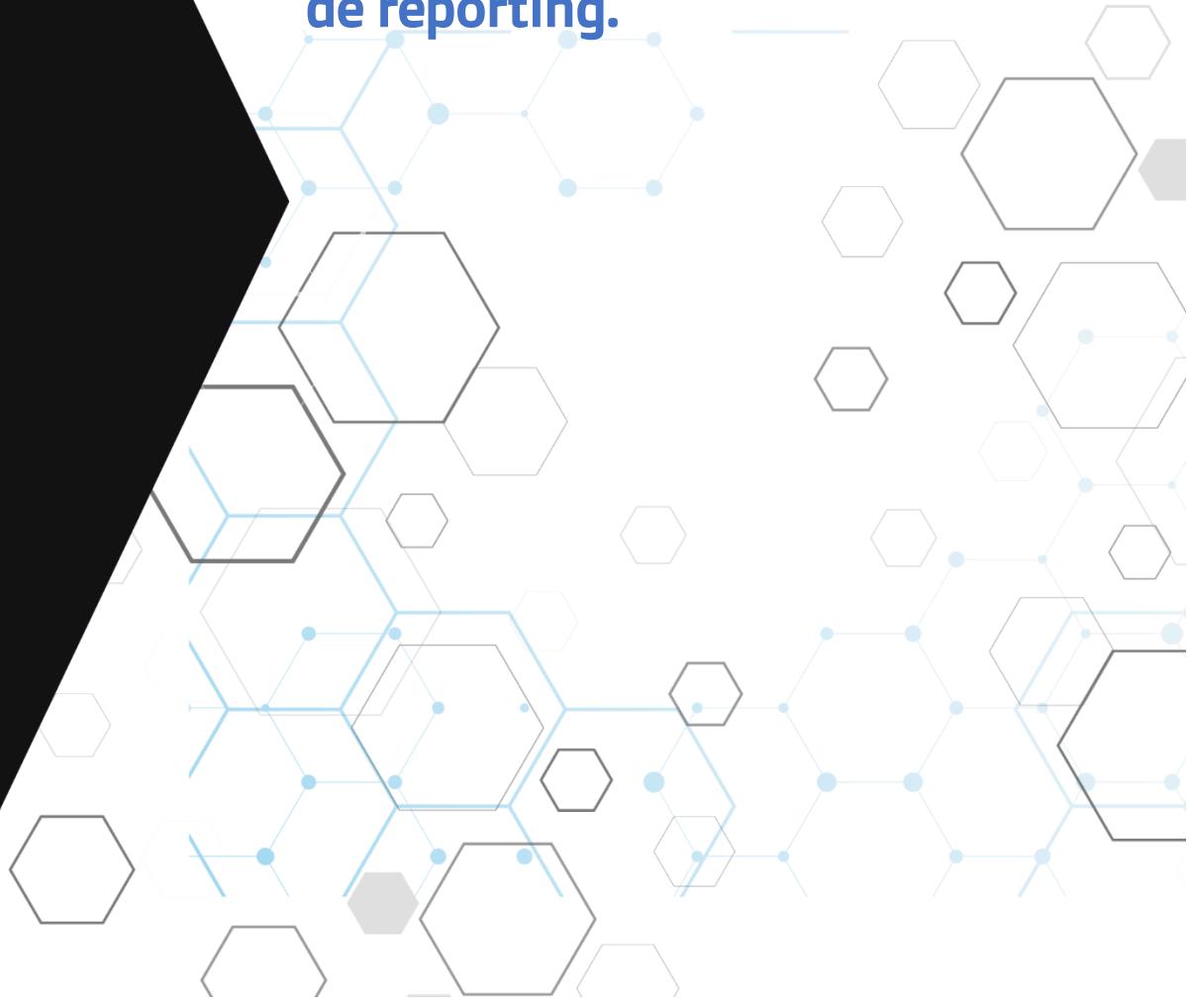
Cas de test	Démarche	Comportement attendu	Résultat
Test de création d'une tâche avec une durée spécifiée.	Après l'accès à l'interface de création d'une tâche, l'administrateur saisit les informations de la tâche, choisit la date de début et la date de fin, ainsi que l'utilisateur affecté à la tâche, et clique sur le bouton "Create Task".	Redirection vers la liste des tâches. Si la tâche est créée, une nouvelle tâche sera planifiée avec une durée et des niveaux spécifiés.	Conforme
Test de création d'une campagne	Après avoir cliqué sur l'icône de création de campagne, un menu déroulant s'ouvre. L'utilisateur saisit les informations nécessaires pour la campagne et clique sur le bouton "Create Campaign".	Redirection vers la liste des campagnes. Si une campagne est créée, une nouvelle campagne s'affiche avec une couleur et un nom distincts.	Conforme
Test de modification d'une tâche	Après avoir accédé à la liste des tâches, l'administrateur sélectionne la tâche à modifier, et l'interface passe au formulaire de modification de tâche. L'administrateur met à jour les champs nécessaires et clique sur le bouton « Update Task ».	Redirige vers la liste des tâches, si la tâche est modifiée l'administrateur note que les informations sur la tâche seront modifiées avec succès.	Conforme

4.7 Conclusion

Dans ce chapitre, nous avons spécifié les besoins et conçu les solutions à l'aide de diagrammes de séquence et de classes pour le sprint 2. Nous avons également réalisé les histoire utilisateurs, testé les fonctionnalités, et validé les résultats avec le Scrum Master et le Product Owner lors de la revue du sprint. Forts de ces validations, nous sommes prêts à entamer la prochaine étape du projet : sprint 3.

Chapitre 5

**Sprint 3 : Fonctionnalités de
Gestion des publications et
de reporting.**



5.1 Introduction

Dans ce chapitre intitulé "Sprint 3 : Fonctionnalités de Gestion des Publications et de Reporting", nous abordons actuellement le troisième sprint du projet, qui se concentre sur les fonctionnalités essentielles de gestion des publications et de reporting. Ce sprint représente une étape cruciale du projet, car il introduit des fonctionnalités clés qui permettent aux utilisateurs de créer, modifier, supprimer et planifier des publications sur les réseaux sociaux, ainsi que de générer des rapports détaillés pour analyser leurs campagnes. Nous commençons par définir les objectifs spécifiques de ce sprint et examinons le backlog des fonctionnalités à mettre en œuvre. Ensuite, nous décrivons en détail les user stories associées à ce sprint, qui guident notre processus de développement. Nous procédons à une conception approfondie des fonctionnalités identifiées, en accordant une attention particulière aux besoins et aux exigences des utilisateurs. Enfin, nous détaillons les étapes de réalisation et de test prévues pour ce sprint, avant de conclure sur les résultats escomptés à l'issue de cette phase du projet.

5.2 Objectifs du sprint

L'objectif de ce sprint serait de développer les fonctionnalités liées à la gestion des publications sur les réseaux sociaux ainsi qu'à la consultation des rapports et des statistiques d'utilisation du système. Cela inclut la possibilité pour les utilisateurs de se connecter à de nouveaux comptes de réseaux sociaux, de créer de nouvelles publications, de modifier les publications existantes qui n'ont pas encore été publiées, et de supprimer ces publications. De plus, l'administrateur devrait être en mesure de consulter des rapports et des statistiques sur l'utilisation du système, permettant ainsi une analyse approfondie de l'activité.

5.3 Backlog du sprint

Pour ce troisième sprint, nous avons préparé un tableau Sprint Backlog des tâches clés à réaliser. Chaque fonctionnalité est associée à une estimation de l'effort de développement, permettant la planification et la priorisation des activités de sprint.

ID	Fonctionnalité	Histoire utilisateur	Estimation
1	Gestion des publications	En tant qu'utilisateur, je veux pouvoir me connecter à un nouveau compte de réseau social.	3 jours
2		En tant qu'utilisateur, je veux pouvoir créer une nouvelle publication sur les réseaux sociaux.	7 jours
3		En tant qu'utilisateur, je veux pouvoir modifier une publication existante non encore publiée.	5 jours

ID	Fonctionnalité	Histoire utilisateur	Estimation
4	Gestion des publications	En tant qu'utilisateur, je veux pouvoir supprimer une publication existante non encore publié.	3 jours
5	Affichage des rapports et des statistiques	En tant qu'administrateur, je veux pouvoir consulter des rapports et des statistiques sur l'utilisation du système.	7 jours

Tableau 17- Backlog du sprint 3

5.4 Conception du sprint

Dans cette partie, nous nous concentrerons sur la présentation de la conception de notre troisième sprint.

5.4.1 Diagramme des cas d'utilisation

La figure 32 suivante illustre le diagramme de cas d'utilisation de notre troisième sprint.

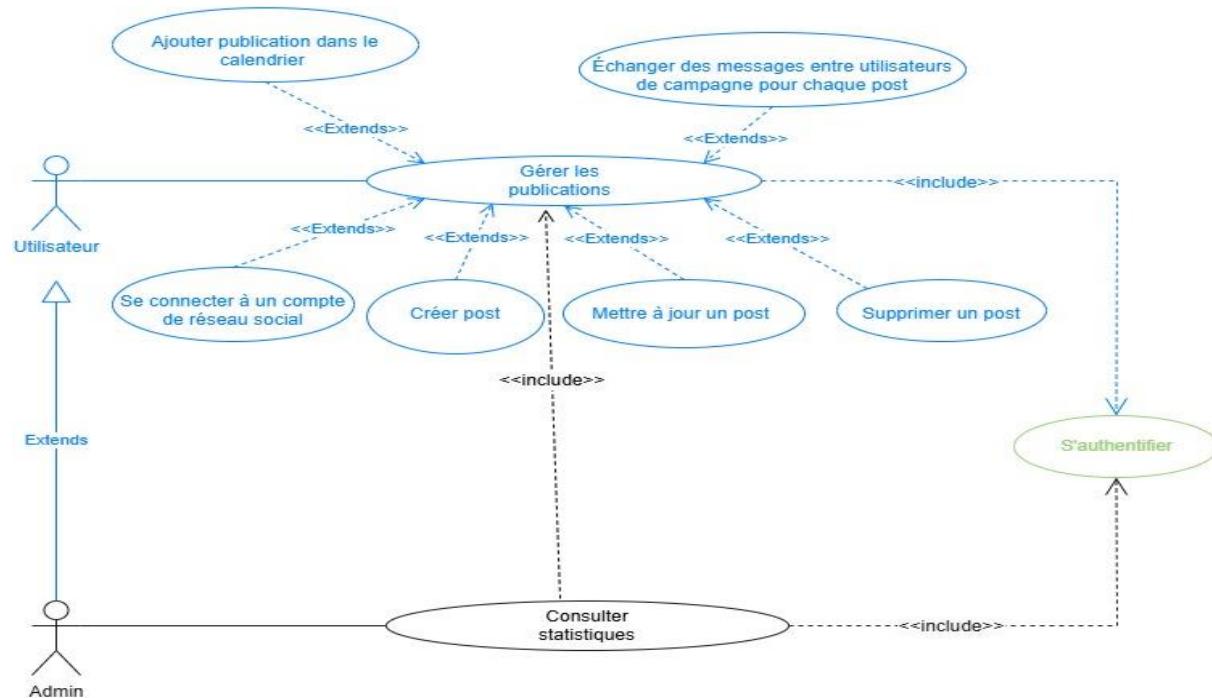


Figure 35- Diagramme des cas d'utilisation du troisième sprint

5.4.2 Description textuelle des cas d'utilisation

Pour le troisième sprint, nous nous concentrerons sur l'exploration détaillée des cas d'utilisation, en particulier ceux relatifs à la gestion des publications et à l'affichage des rapports et des statistiques. En analysant minutieusement ces cas d'utilisation, nous visons à clarifier les interactions du système avec les utilisateurs, facilitant ainsi la création de diagrammes de séquence précis et exhaustifs. Cette approche nous permettra d'assurer une compréhension complète et une implémentation efficace des fonctionnalités prévues.

5.4.2.1 Description textuelle du cas d'utilisation « Se connecter à un compte de réseau social »

Cas d'utilisation : Se connecter à un compte de réseau social
Résumé : Ce cas d'utilisation permet à un acteur de se connecter à un compte de réseau social.
Acteur : Utilisateur, Admin
Pré-condition : L'utilisateur doit être authentifié et avoir accès à l'interface « Connect Channels ».
Post-condition : Le compte sera ajouté à la liste des comptes déjà connectés.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<p>01. L'utilisateur demande l'interface de connexion au compte.</p> <p>02. Le système affiche une liste des types de réseaux sociaux auxquels il est possible de se connecter.</p> <p>03. L'utilisateur sélectionne le type de réseau social à ajouter (par exemple, Facebook, Instagram, Twitter).</p> <p>04. Le système redirige l'utilisateur vers la page d'authentification du réseau social sélectionné.</p> <p>05. L'utilisateur saisit ses identifiants de connexion (nom d'utilisateur et mot de passe) pour le réseau social choisi.</p> <p>06. Le réseau social demande l'autorisation pour que votre plateforme accède aux informations de l'utilisateur.</p> <p>07. L'utilisateur accorde les autorisations nécessaires.</p> <p>08. Le réseau social renvoie un jeton d'authentification à votre plateforme.</p> <p>09. Le système enregistre ce jeton et les informations pertinentes du compte et redirige l'utilisateur vers la liste des comptes connectés.</p>
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
Echec de Se connecter à un compte de réseau social, ce scénario démarre au point 04 du scénario nominal.
05 : Le système affiche un message d'erreur, puis le scénario reprend au point 03 du scénario nominal.

Tableau 18- Description textuelle du cas d'utilisation « Se connecter à un compte de réseau social »

5.4.1.1 Description textuelle du cas d'utilisation « Créer publication »

Cas d'utilisation : Créer publication
Résumé : Ce cas d'utilisation permet à l'acteur de créer une publication avec différents statuts (Brouillon, Publier, Programmer).
Acteur : Utilisateur, Admin
Pré-condition : L'utilisateur doit être authentifié et avoir accès à l'interface de création publication.
Post-condition : La publication créée est affichée dans un calendrier avec deux vues.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<ul style="list-style-type: none"> 01. L'utilisateur sollicite l'interface de création de publication 02. Le système affiche une modale de création de publication. 03. L'utilisateur saisit les informations préférées de la publication ou du statut. 04. Le système vérifie la validité des informations de la publication ou du statut. 05. La nouvelle publication ou le statut est créé dans le système et apparaît dans un calendrier avec deux vues.
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
En cas d'informations incorrectes ou manquantes, ce scénario démarre au point 04 du scénario nominal.
<ul style="list-style-type: none"> 05 : Le système informe l'utilisateur du type d'erreur, puis le scénario reprend au point 03 du scénario nominal.

Tableau 19- Description textuelle du cas d'utilisation « Créer publication »

5.4.2.3 Description textuelle du cas d'utilisation « Modifier publication »

Cas d'utilisation : Modifier publication
Résumé : Ce cas d'utilisation permet à l'acteur de modifier une publication qui n'a pas encore été publiée.
Acteur : Utilisateur, Admin
Pré-condition : L'utilisateur doit être authentifié et avoir accès à l'interface de modification d'une publication
Post-condition : La publication modifiée s'affiche dans la liste des publications de l'interface de campagne associée.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<ul style="list-style-type: none"> 01. L'utilisateur accède à l'interface de campagne sélectionnée. 02. Le système affiche la liste des publications créées dans la campagne sélectionnée. 03. L'utilisateur sélectionne la publication qu'il souhaite modifier. 04. Le système affiche une modale de modification de la publication sélectionnée. 05. L'utilisateur effectue les modifications nécessaires dans les champs de modale. 06. Le système vérifie la validité des informations de la publication. 07. La publication est modifiée dans le système et apparaît dans la liste des publications dans l'interface de campagne associée.
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
En cas d'informations incorrectes ou manquantes, ce scénario démarre au point 06 du scénario nominal.
<ul style="list-style-type: none"> 07 : Le système informe l'utilisateur du type d'erreur, puis le scénario reprend au point 04 du scénario nominal.

Tableau 20- Description textuelle du cas d'utilisation « Modifier publication »

5.4.3 Diagrammes de séquence

Dans cette section, nous avons élaboré des diagrammes de séquence pour le sprint 3 afin de démontrer les interactions entre les différentes composantes du système dans le cadre des cas

d'utilisation de gestion des publications et de reporting. Ces diagrammes détaillent la succession des actions et événements, ce qui aide à visualiser clairement le fonctionnement du système. Notre but était de clarifier les processus pour différents scénarios, fournissant ainsi une compréhension approfondie et complète du système.

5.4.3.1 Diagramme de séquence du cas d'utilisation « Se connecter à un compte de réseau social »

Dans le diagramme suivant, nous éludions l'interaction spécifique entre l'utilisateur et les différentes composantes du système, visant à permettre la connexion à un compte de réseau social.

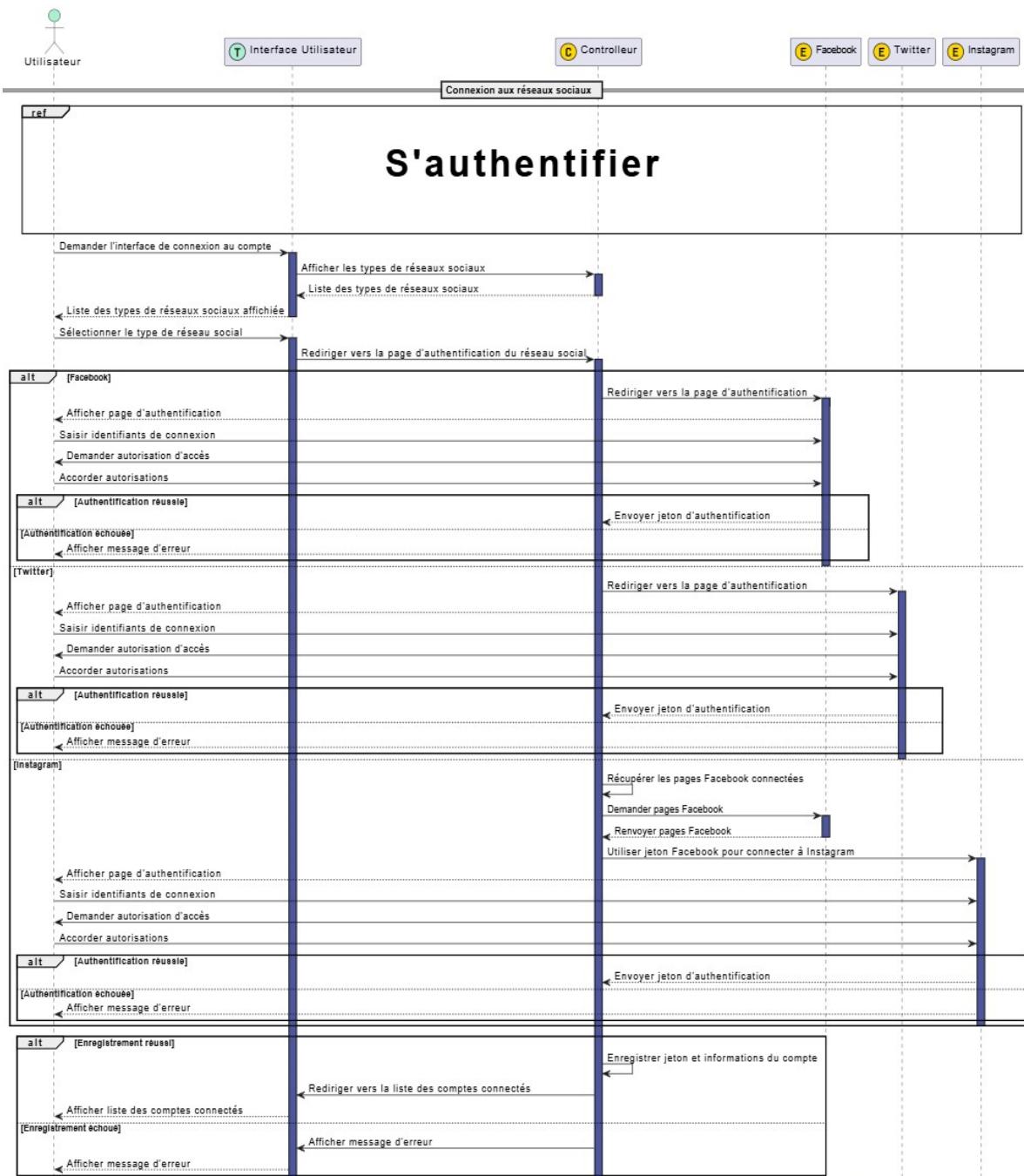


Figure 36- Diagramme de séquence « Se connecter à un compte de réseau social »

5.4.3.2 Diagramme de séquence du cas d'utilisation « Crée publication »

Dans le diagramme suivant, nous éludions l'interaction spécifique entre l'utilisateur et les différentes composantes du système, visant à permettre la création d'une publication.

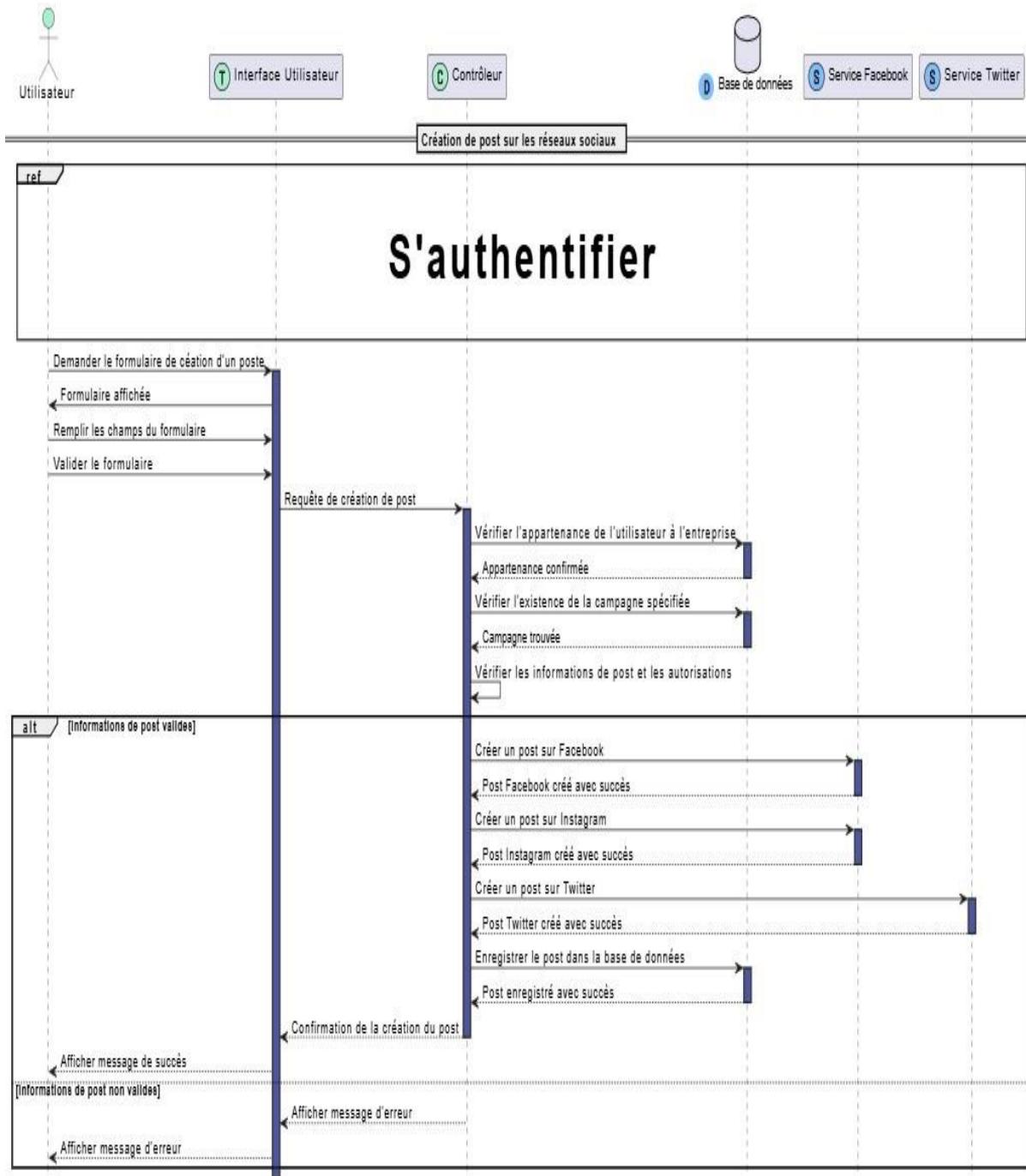


Figure 37- Diagramme de séquence « Crée publication »

5.4.3.3 Diagramme de séquence du cas d'utilisation « Modifier publication »

Dans le diagramme présenté ci-dessous, nous éludions l'interaction spécifique entre l'utilisateur et les différentes composantes du système, visant à permettre la modification d'une publication.

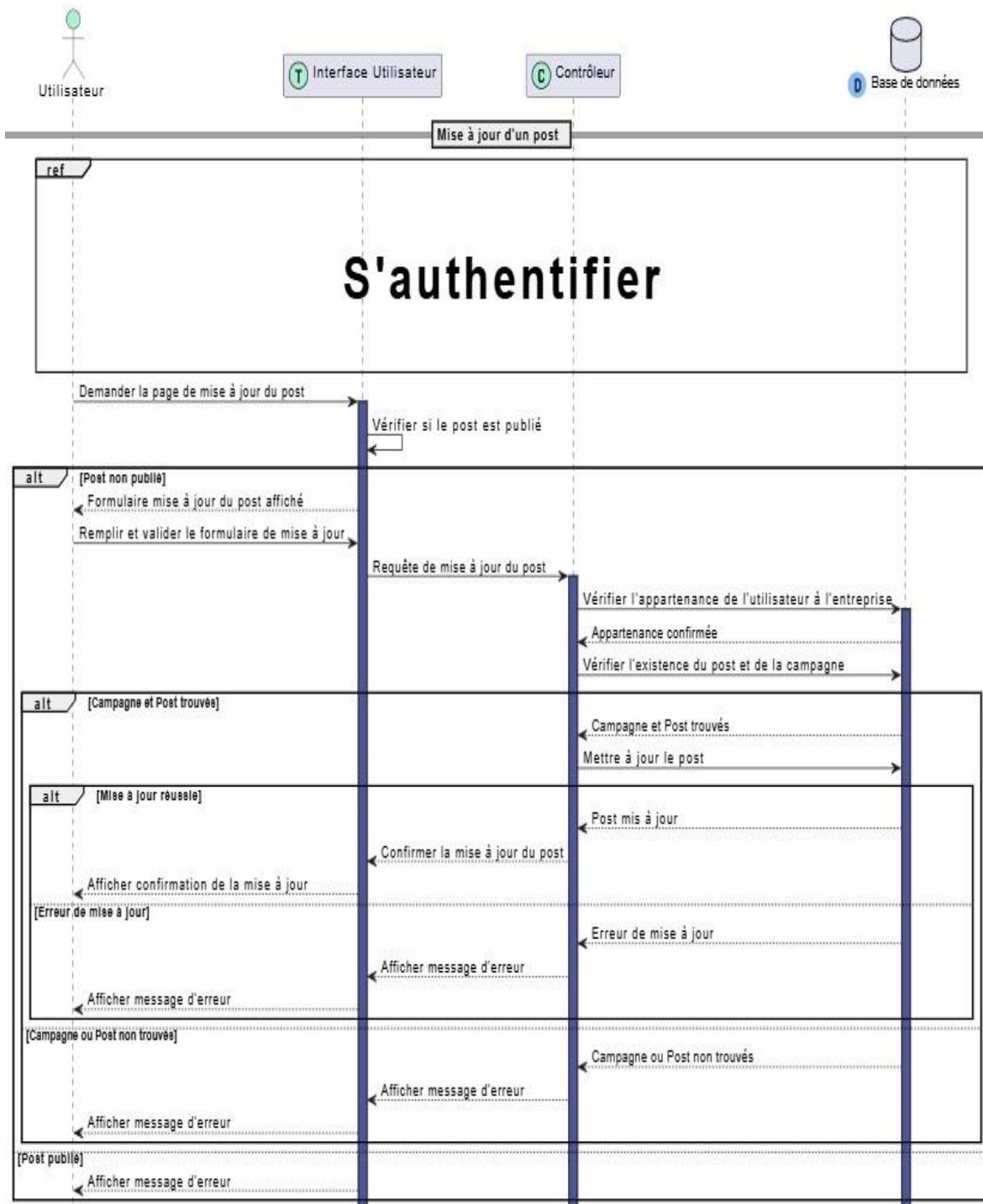


Figure 38- Diagramme de séquence «Modifier publication»

5.4.4 Diagramme de classes

Après avoir présenté les diagrammes de séquence, qui illustrent les interactions dynamiques entre les acteurs et le système dans des contextes spécifiques, nous avons complété cette vision par la conception d'un diagramme de classes. Pour le sprint 3, nous adoptons une approche similaire. Le diagramme de classes ci-dessous (figure 39) fournit une représentation structurée

des entités et de leurs relations, offrant ainsi une compréhension claire et organisée des composants clés et de leur interconnexion dans ce sprint.

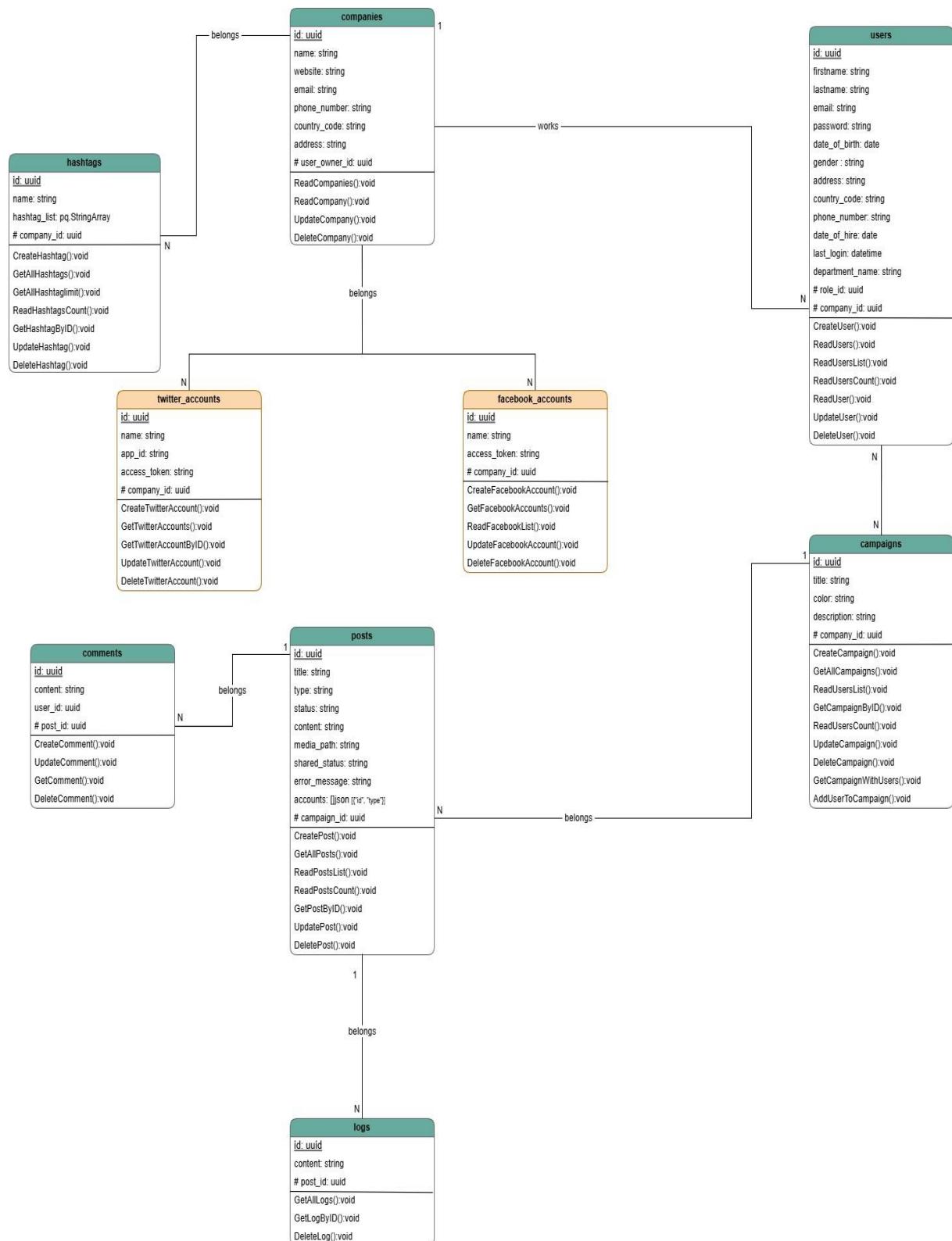


Figure 39- Diagramme de classes du Sprint 3

5.5 Réalisation

Dans cette phase, nous passons de la conception à la réalisation concrète de nos idées. Notre objectif est de transformer les concepts discutés en actions tangibles, avec un accent particulier sur la création visuelle des prototypes et la conception des interfaces utilisateur. Cette étape démontre notre engagement à mettre en œuvre les fonctionnalités définies dans le backlog du sprint.

5.5.1 Modélisation des maquettes

La création de maquettes d'interface utilisateur simplifie la visualisation de l'application pour le client et encourage la collaboration en clarifiant les attentes du produit. Dans ce contexte, nous présentons plusieurs exemples de maquettes graphiques illustrant des scénarios utilisateur notamment : se connecter aux réseaux sociaux, créer une publication, modifier publication et consulter les rapports et les statistiques.

5.5.1.1 Le Prototype relatif à l'histoire utilisateur « Se connecter à un compte de réseau social »

Le prototype ci-dessous correspond à l'histoire utilisateur « Se connecter à un compte de réseau social ». Il répond au scénario d'utilisation : En tant qu'utilisateur, je souhaite pouvoir me connecter à un nouveau compte de réseau social.

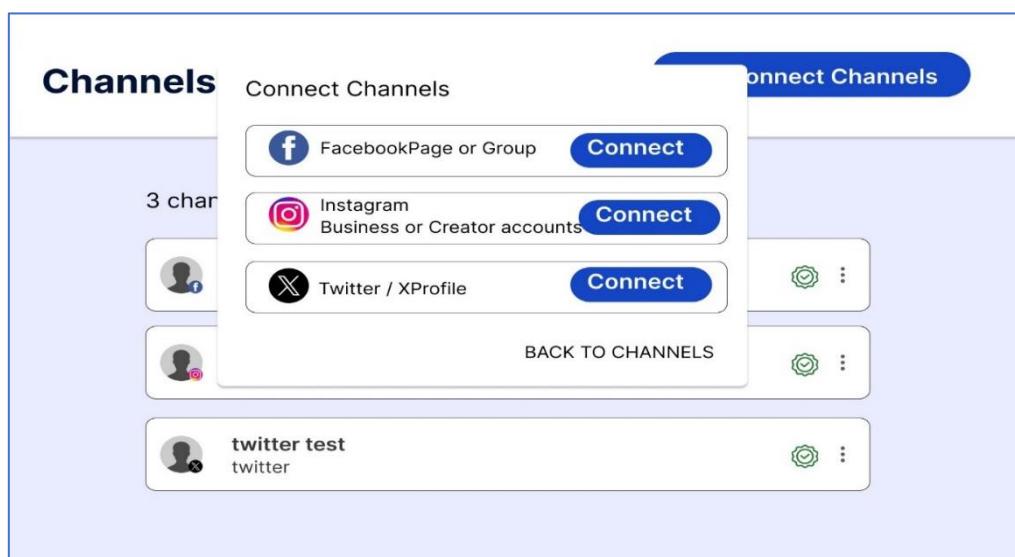


Figure 40- Prototype relatif à l'histoire utilisateur « Se connecter à un compte de réseau social »

5.5.1.2 Le Prototype relatif à l'histoire utilisateur « Crée publication »

Le prototype exposé ci-dessous correspond à l'histoire utilisateur « Crée publication ». Il répond au scénario d'utilisation : En tant qu'utilisateur, je veux pouvoir créer une nouvelle publication sur les réseaux sociaux.

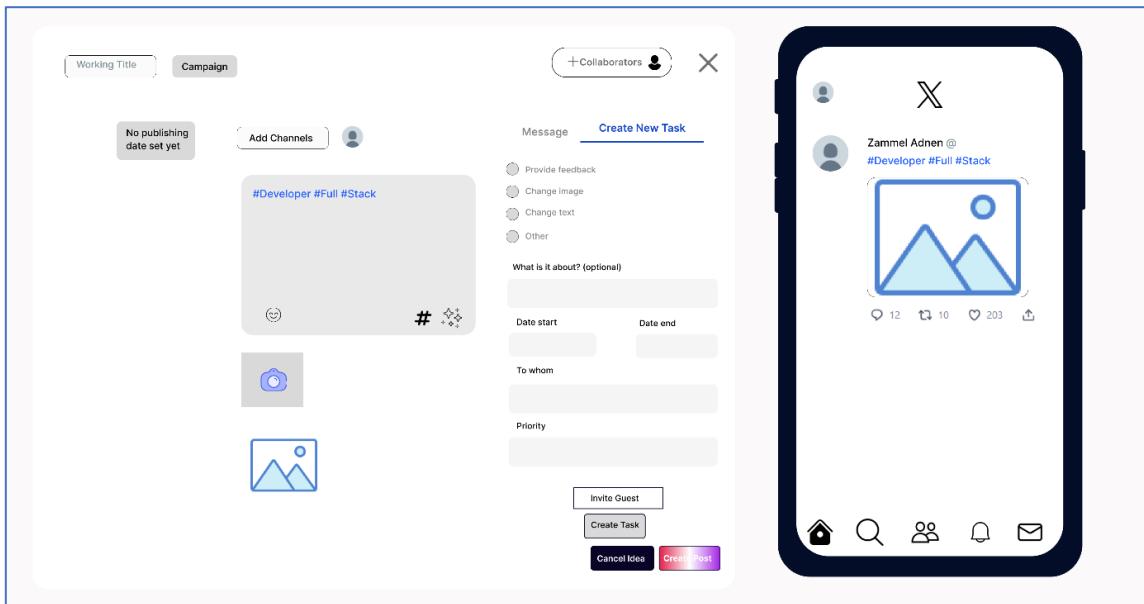


Figure 41- Prototype relatif à l'histoire utilisateur « Crée publication »

5.5.1.4 Le Prototype relatif à l'histoire utilisateur « Modifier publication »

Le prototype ci-dessous illustre l'histoire utilisateur « Modifier la publication ». Il répond au scénario d'utilisation : En tant qu'utilisateur, je veux pouvoir modifier une publication existante non encore publiée.

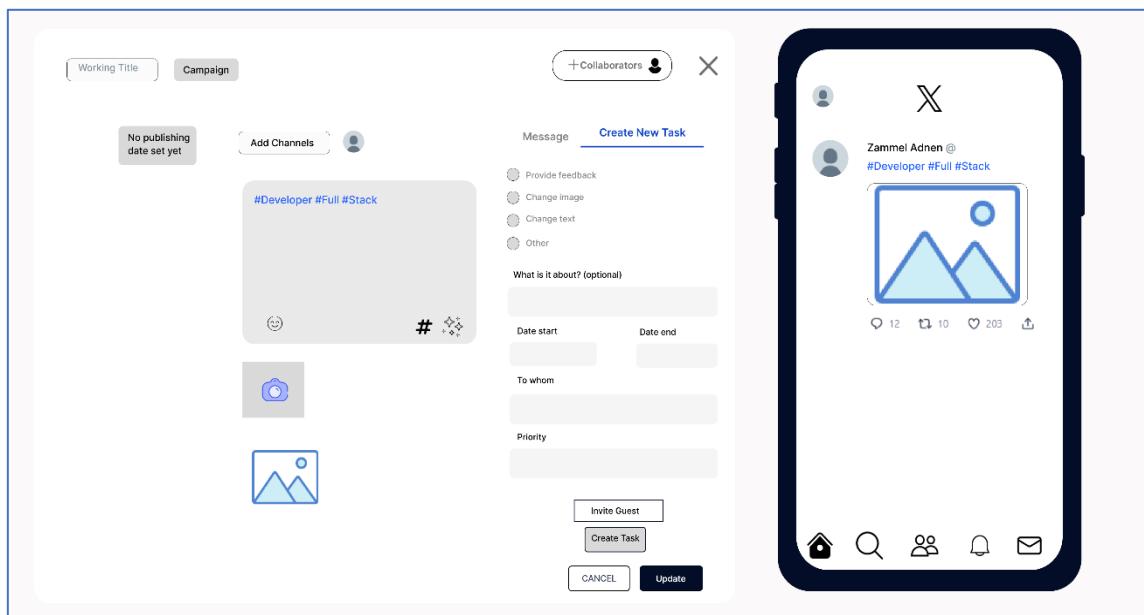


Figure 42- Prototype relatif à l'histoire utilisateur « Modifier publication »

5.5.1.4 Le Prototype relatif à l'histoire utilisateur « consulter les rapports et les statistiques »

Le prototype ci-dessous illustre l'histoire utilisateur « Consulter des rapports et des statistiques ». Il répond au scénario d'utilisation : En tant qu'administrateur, je veux pouvoir consulter des rapports et des statistiques sur l'utilisation du système.

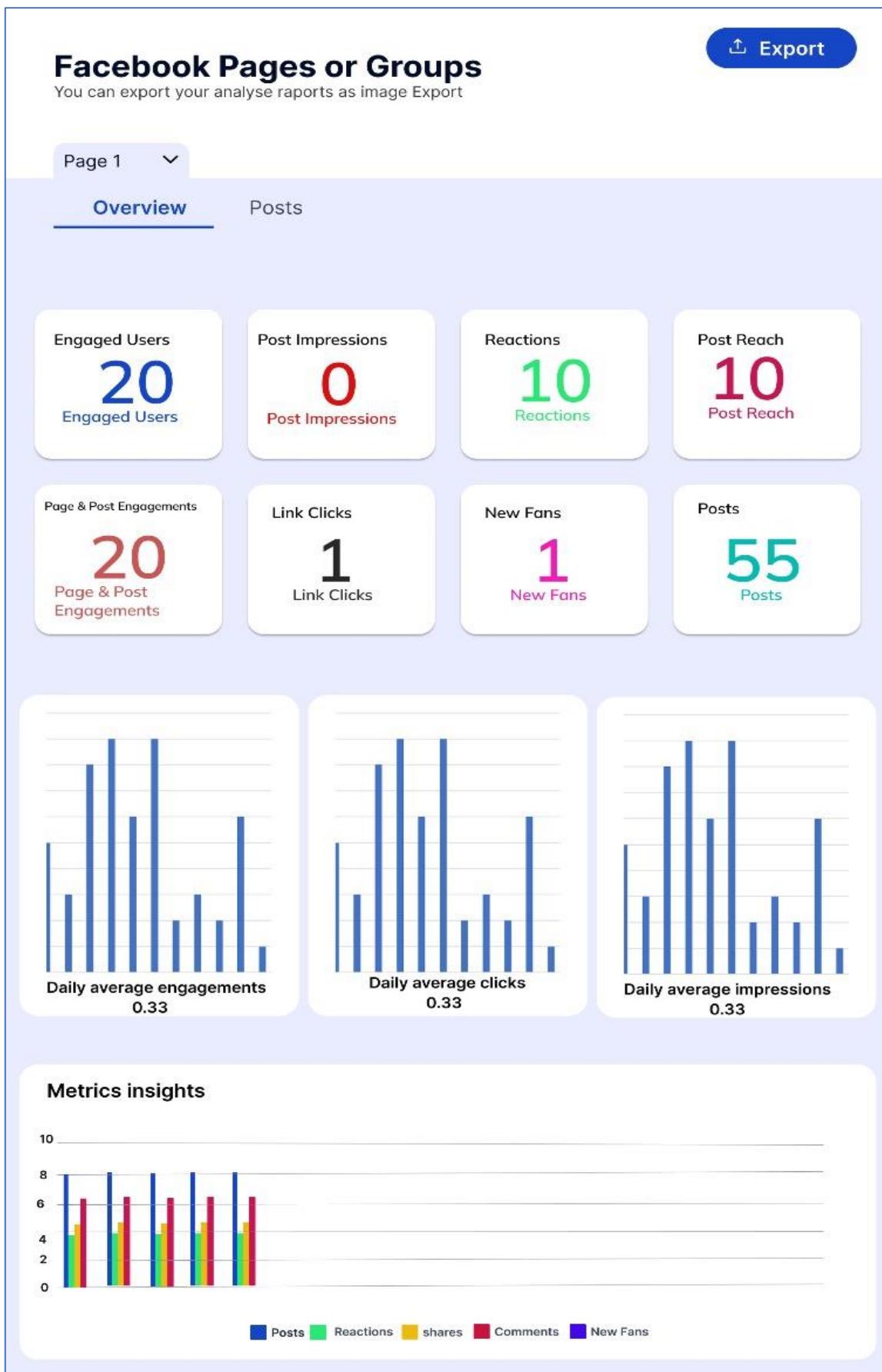


Figure 43- Prototype relatif à l'histoire utilisateur « Consulter les rapports et les statistiques »

5.5.2 Développement des interfaces utilisateurs

Dans cette partie, nous nous concentrerons sur la présentation de quelques interfaces utilisateurs développées lors de notre troisième sprint.

5.5.2.1 Interface graphique relative à l'histoire utilisateur « Se connecter à un compte de réseau social »

Cette interface est destinée à se connecter à un compte de réseau social, offrant à l'utilisateur (administrateur ou utilisateur standard) la possibilité d'ajouter à un autre compte.

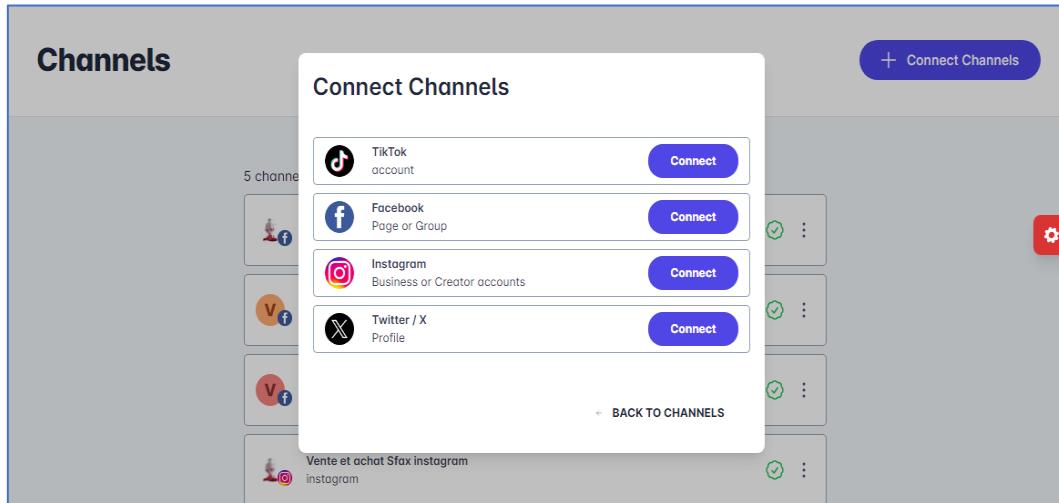


Figure 44- Interface graphique relative à l'histoire utilisateur « Se connecter à un compte de réseau social »

5.5.2.2 Interface graphique relative à l'histoire utilisateur « Crée publication »

Cette interface est spécialement conçue pour la création de publications, permettant à l'utilisateur (administrateur ou utilisateur standard) de saisir et de publier du contenu sur les réseaux sociaux.

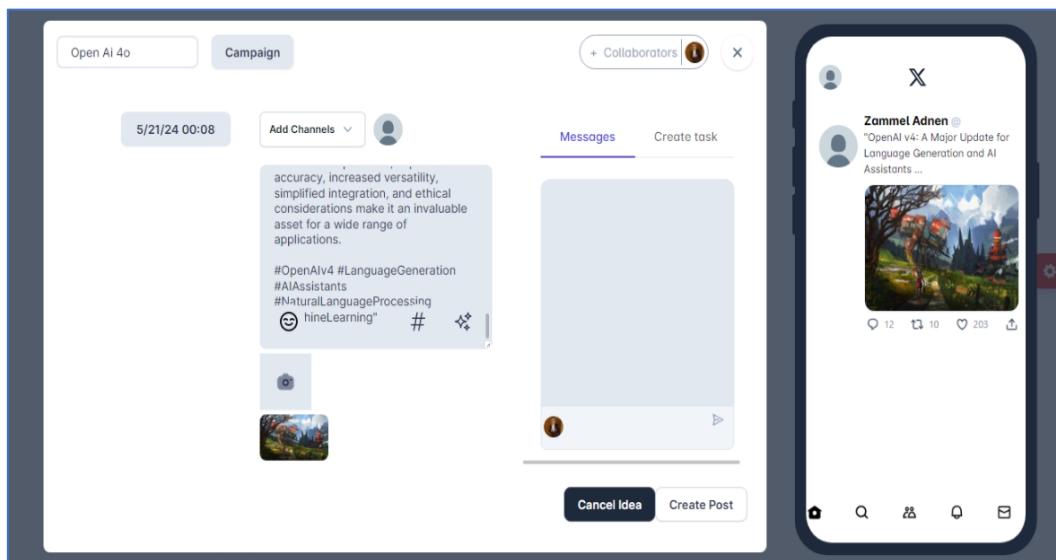


Figure 45- Interface graphique relative à l'histoire utilisateur « Crée publication »

5.5.2.4 Interface graphique relative à l 'histoire utilisateur « Modifier publication »

Cette fonctionnalité est dédiée à la modification des publications existantes, offrant à l'utilisateur (administrateur ou utilisateur standard) la possibilité de mettre à jour le contenu ou les paramètres d'une publication avant sa diffusion sur les réseaux sociaux.

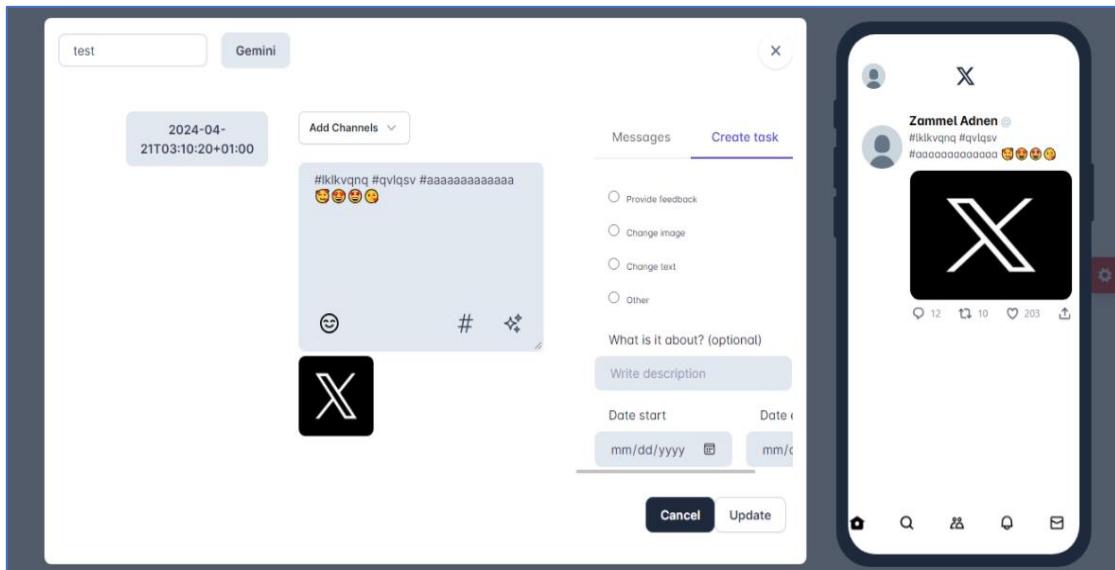


Figure 46- Interface graphique relative à l 'histoire utilisateur « Modifier publication »

5.5.2.4 Interface graphique relative à l 'histoire utilisateur « Consulter les rapports et les statistiques »

Cette fonctionnalité est dédiée à la consultation des rapports et des statistiques, offrant à l'administrateur la possibilité d'accéder à des données détaillées sur l'utilisation du système pour une analyse approfondie et une prise de décision éclairée.

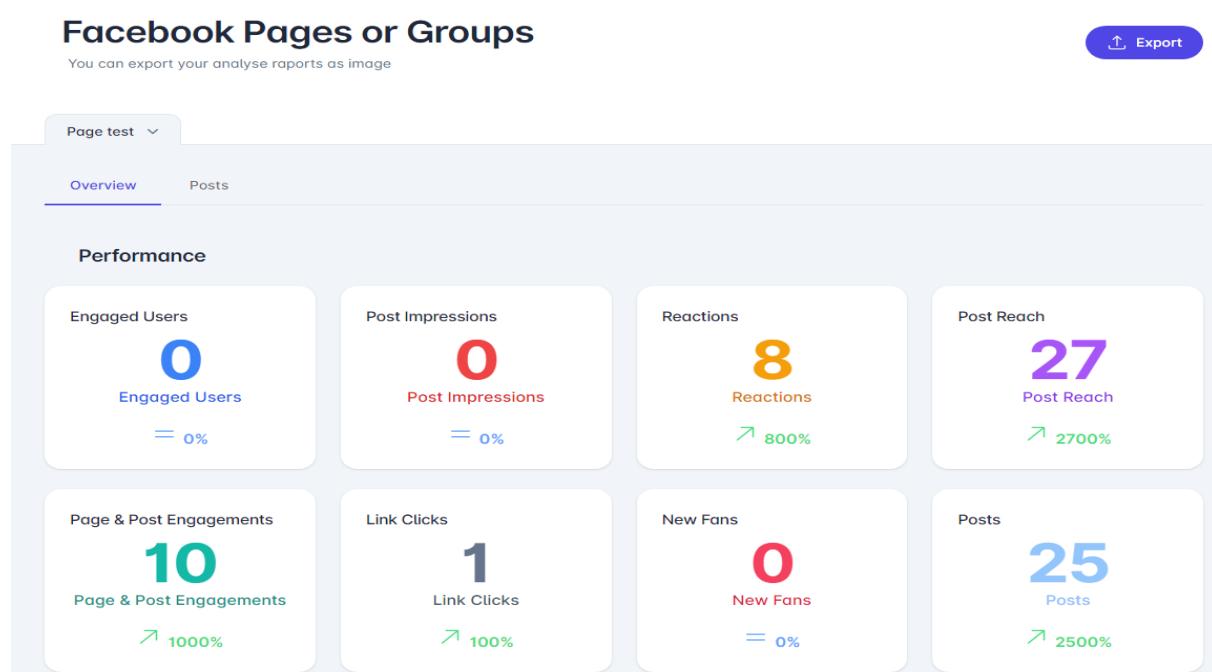




Figure 47- Interface graphique relative à l 'histoire utilisateur « Consulter les rapports et les statistiques »

5.5.3 Déploiement des composants logiciels

Dans le cadre de notre projet, l'application "SocialScop" requiert une interaction avec divers réseaux sociaux, offrant ainsi aux utilisateurs la possibilité de gérer l'ensemble de leurs publications. La figure 48 ci-dessous illustre l'architecture complète de notre système, mettant en évidence les différentes API utilisées pour favoriser une intégration harmonieuse avec les plateformes Facebook, Instagram et Twitter :

- API Facebook Graph : cette API sert de pont vers l'écosystème de Facebook, facilitant les interactions avec la vaste gamme de données et de fonctionnalités de cette plateforme. Elle renforce ainsi des fonctionnalités telles que la création et la planification de publications.
- API Twitter v2 : avec un accès étendu aux données et fonctionnalités de Twitter, l'API v2 permet une intégration transparente avec les comptes Twitter, facilitant des tâches telles que la création et la planification de tweets.

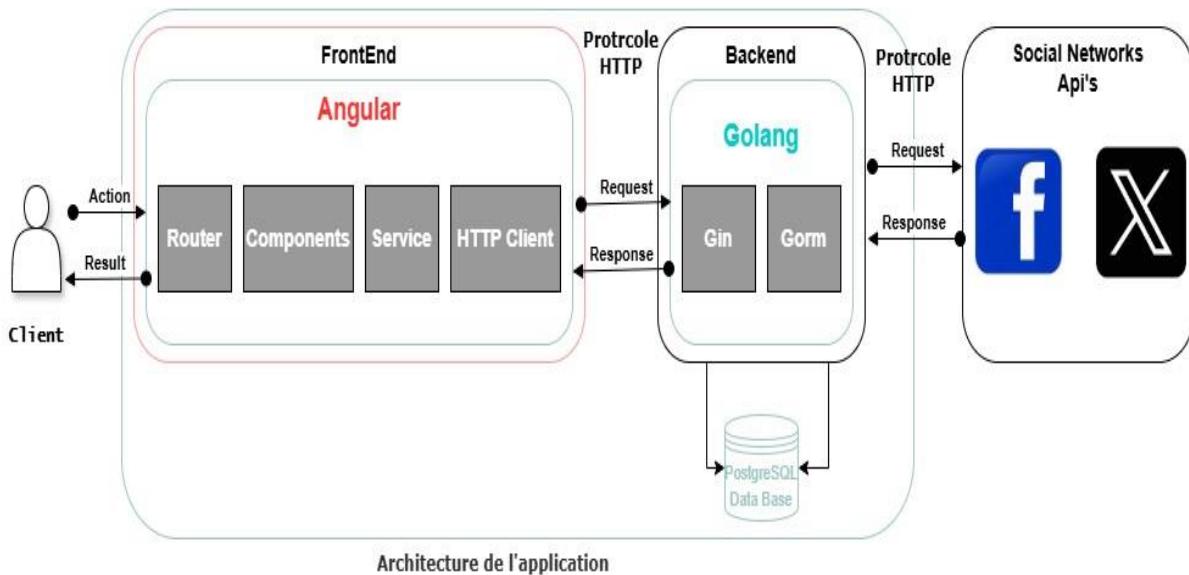


Figure 48- Architecture principale de l'application

5.6 Test fonctionnel

Après avoir achevé le sprint 3, nous entamons la phase de finalisation pour évaluer notre avancement et nous préparer pour le prochain sprint. Cette étape comprend la présentation des principaux scénarios de tests fonctionnels du ce sprint dans un tableau pour une meilleure lisibilité et compréhension.

Cas de test	Démarche	Comportement attendu	Résultat
Test de connexion d'un compte de réseau social.	<p>L'utilisateur accède d'abord à l'interface de connexion au compte du réseau social. Le système affiche alors une liste des différents types de réseaux sociaux disponibles. Ensuite, l'utilisateur sélectionne le type de réseau social auquel il souhaite se connecter. Une fois cette sélection effectuée, il est automatiquement redirigé vers la page d'authentification correspondante. Après avoir correctement saisi ses identifiants de connexion, l'utilisateur accorde les autorisations demandées par le réseau social.</p>	<p>Le compte est ajouté à la liste des comptes connectés, confirmant ainsi que l'utilisateur s'est connecté avec succès à un compte de réseau social.</p>	Conforme

Cas de test	Démarche	Comportement attendu	Résultat
Test de création d'une publication	L'utilisateur accède à l'interface de création de publication. Ensuite, il sélectionne la campagne, rédige ou génère le titre, écrit ou génère l'hashtag, crée le contenu et ajoute éventuellement un fichier média (image ou vidéo) pour la publication, ainsi que le ou les comptes de réseaux sociaux associés.	Redirection vers la page de calendrier avec la vue choisie. Si la publication est créée, elle s'affiche dans le calendrier selon le type de réseau social (Facebook [Bleu], Instagram [Rose], Twitter [Jaune]).	Conforme
Test de modification d'une publication	L'utilisateur accède à la liste des publications. Ensuite, il sélectionne la publication qu'il souhaite modifier, ce qui déclenche l'ouverture de l'interface modale de la modification de la publication. L'utilisateur met ensuite à jour les champs nécessaires dans la modale, puis valide l'action de modification.	Redirection vers la liste des publications, si la publication est modifiée et de ce fait les informations sur la publication seront modifiées avec succès.	Conforme

5.7 Conclusion

Dans ce chapitre, nous avons détaillé les exigences et conçu des solutions robustes pour le sprint 3 en utilisant des diagrammes de séquence et de classes. Nous avons également implémenté les histoires utilisateurs, effectué des tests rigoureux des fonctionnalités, et validé les résultats en collaboration avec le Scrum Master et le Product Owner lors de la revue de sprint. Grâce à ces validations, nous avons confirmé l'efficacité et la performance des solutions développées. Forts de ces accomplissements, nous sommes prêts à avancer vers la prochaine phase du projet : le sprint 4.

Chapitre 6

Sprint 4 : Fonctionnalités de collaboration, de Gestion des rôles et des permissions.



6.1 Introduction

Dans ce chapitre intitulé "Sprint 4 : Gestion des Rôles et Collaborations", nous explorons le dernier sprint du projet, axé sur les fonctionnalités essentielles de gestion des rôles, des collaborations et des notifications. Ce sprint marque la finalisation de notre projet en introduisant des outils clés pour l'administration et la communication au sein de la plateforme. Nous définissons les objectifs spécifiques de ce sprint, examinons le backlog des fonctionnalités à mettre en œuvre, et décrivons les user stories associées qui orientent notre développement. Enfin, nous abordons les étapes de réalisation et de test, ainsi que les résultats attendus de cette phase.

6.2 Objectifs du sprint

Ce sprint final se concentre sur la gestion des rôles, des collaborations et des notifications. Les principaux objectifs consistent à permettre la création, la modification et la suppression des rôles utilisateurs, ainsi qu'à faciliter la collaboration entre les utilisateurs et la gestion des notifications importantes. Les administrateurs auront la possibilité de gérer les autorisations des rôles et de supprimer des rôles si nécessaires. De plus, les utilisateurs pourront collaborer efficacement sur des projets et recevoir des notifications sur les activités clés. Ce sprint vise à renforcer l'administration et la communication au sein de la plateforme, assurant ainsi une gestion optimisée et une collaboration fluide.

6.3 Backlog du sprint

Pour ce quatrième sprint, nous avons établi un tableau Sprint Backlog des tâches essentielles à accomplir. Chaque fonctionnalité est accompagnée d'une estimation de l'effort de développement, facilitant ainsi la planification et la priorisation des activités du sprint.

ID	Fonctionnalité	Histoire utilisateur	Estimation
8	Gestion des rôles	En tant qu'administrateur, je veux pouvoir créer un nouvel rôle utilisateur.	5 jours
9		En tant qu'administrateur, je veux pouvoir modifier les autorisations d'un rôle existant.	4 jours
10		En tant qu'administrateur, je veux pouvoir supprimer un rôle utilisateur.	4 jours
25	Gestion des collaborations	En tant qu'utilisateur, je veux pouvoir collaborer avec d'autres utilisateurs dans le cadre d'un projet.	6 jours

ID	Fonctionnalité	Histoire utilisateur	Estimation
26	Affichage des notifications	En tant qu'utilisateur, je veux pouvoir recevoir une notification sur les activités importantes.	3 jours

Tableau 21- Backlog du sprint 4

6.4 Conception du sprint

Dans cette partie, nous mettons l'accent sur la présentation de la conception de notre troisième sprint.

6.4.1 Diagramme des cas d'utilisation

La figure 46 ci-dessous illustre le diagramme de cas d'utilisation de quatrième sprint.

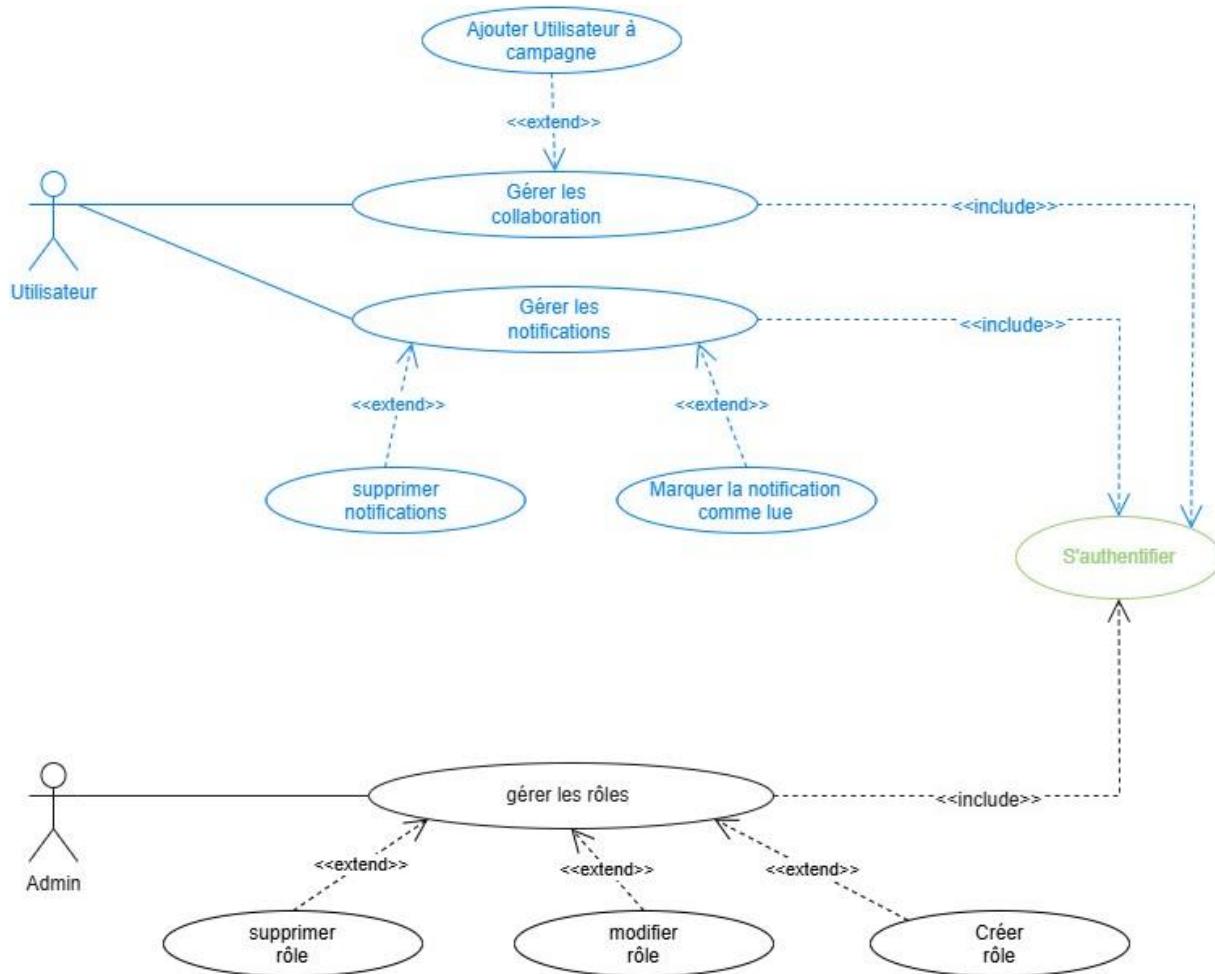


Figure 49- Diagramme des cas d'utilisation du quatrième sprint

6.4.2 Description textuelle des cas d'utilisation

Pour le quatrième sprint, nous mettrons l'accent sur l'examen approfondi de quelques cas d'utilisation ceux concernant la gestion des rôles et des collaborations ainsi que l'affichage des notifications à savoir : ajouter utilisateur à une campagne et créer un rôle pour une entreprise donnée.

6.4.2.1 Description textuelle du cas d'utilisation « Ajouter Utilisateur à une campagne »

Cas d'utilisation : Ajouter Utilisateur à une campagne
Résumé : Ce cas d'utilisation permet à un acteur d'ajouter un utilisateur à une campagne.
Acteur : Utilisateur, Admin
Pré-condition : L'utilisateur doit être authentifié et avoir accès à l'interface de campagne choisie
Post-condition : L'utilisateur sera intégré au groupe d'utilisateurs déjà inclus dans la campagne.
Scénario Nominal
<p>DESCRIPTION DU SCÉNARIO NOMINAL</p> <p><<DEBUT>></p> <ul style="list-style-type: none"> 01. L'utilisateur demande l'interface d'ajout utilisateur. 02. Le système affiche une liste des utilisateurs existants dans l'entreprise. 03. L'utilisateur sélectionne l'utilisateur à ajouter dans la campagne. 04. L'utilisateur valide la sélection. 05. Le système vérifie l'ajout de l'utilisateur à la campagne. 06. Le système intègre l'utilisateur ajouté au groupe des utilisateurs déjà existants. <p><<FIN>></p>
Scénario Alternatif
<p>DESCRIPTION DU SCÉNARIO ALTERNATIF</p> <p>En cas d'échec d'ajout d'utilisateur, ce scénario démarre au point 05 du scénario nominal.</p> <ul style="list-style-type: none"> 06. Le système affiche un message d'erreur, puis le scénario reprend au point 03 du scénario nominal.

Tableau 22- Description textuelle du cas d'utilisation « Ajouter Utilisateur à une campagne »

6.4.2.2 Description textuelle du cas d'utilisation « Créer rôle »

Cas d'utilisation : Créer rôle
Résumé : Ce cas d'utilisation permet à un acteur d'attribuer un rôle pour une entreprise donnée.
Acteur : Admin
Pré-condition : L'administrateur doit être authentifié et avoir accès à la fonctionnalité de Créer rôles.
Post-condition : L'administrateur a créé avec succès un nouveau rôle.
Scénario Nominal
DESCRIPTION DU SCÉNARIO NOMINAL
<<DEBUT>>
<ol style="list-style-type: none"> 01. L'administrateur accède à la fonctionnalité "Company" dans l'interface d'administration et demande formulaire de créer rôle. 02. Le système affiche un formulaire contenant un champ pour le nom du rôle. 03. L'administrateur saisit le nom du rôle et valide le formulaire. 04. Le système valide le nom du rôle. 05. Le système affiche un message de confirmation à l'administrateur indiquant que le rôle a été créé avec succès.
<<FIN>>
Scénario Alternatif
DESCRIPTION DU SCÉNARIO ALTERNATIF
Dans le cas où le nom est invalide ou déjà associée, ce scénario démarre au point 04 du scénario nominal.
<ol style="list-style-type: none"> 05. Le système informe l'utilisateur du type d'erreur, puis le scénario reprend au point 03 du scénario nominal.

Tableau 23- Description textuelle du cas d'utilisation « Créer rôle »

6.4.3 Diagrammes de séquence

Dans ce volet, nous présentons le diagramme de séquence de certains cas d'utilisation du sprint 4, notamment l'ajout d'un utilisateur à une campagne et la création d'un rôle pour une entreprise donnée.

6.4.3.1 Diagramme de séquence du cas d'utilisation « Ajouter Utilisateur à une campagne »

Dans ce diagramme, nous élucidons l'interaction spécifique entre l'utilisateur et les différentes composantes du système, visant à permettre l'ajout d'un utilisateur à une campagne.

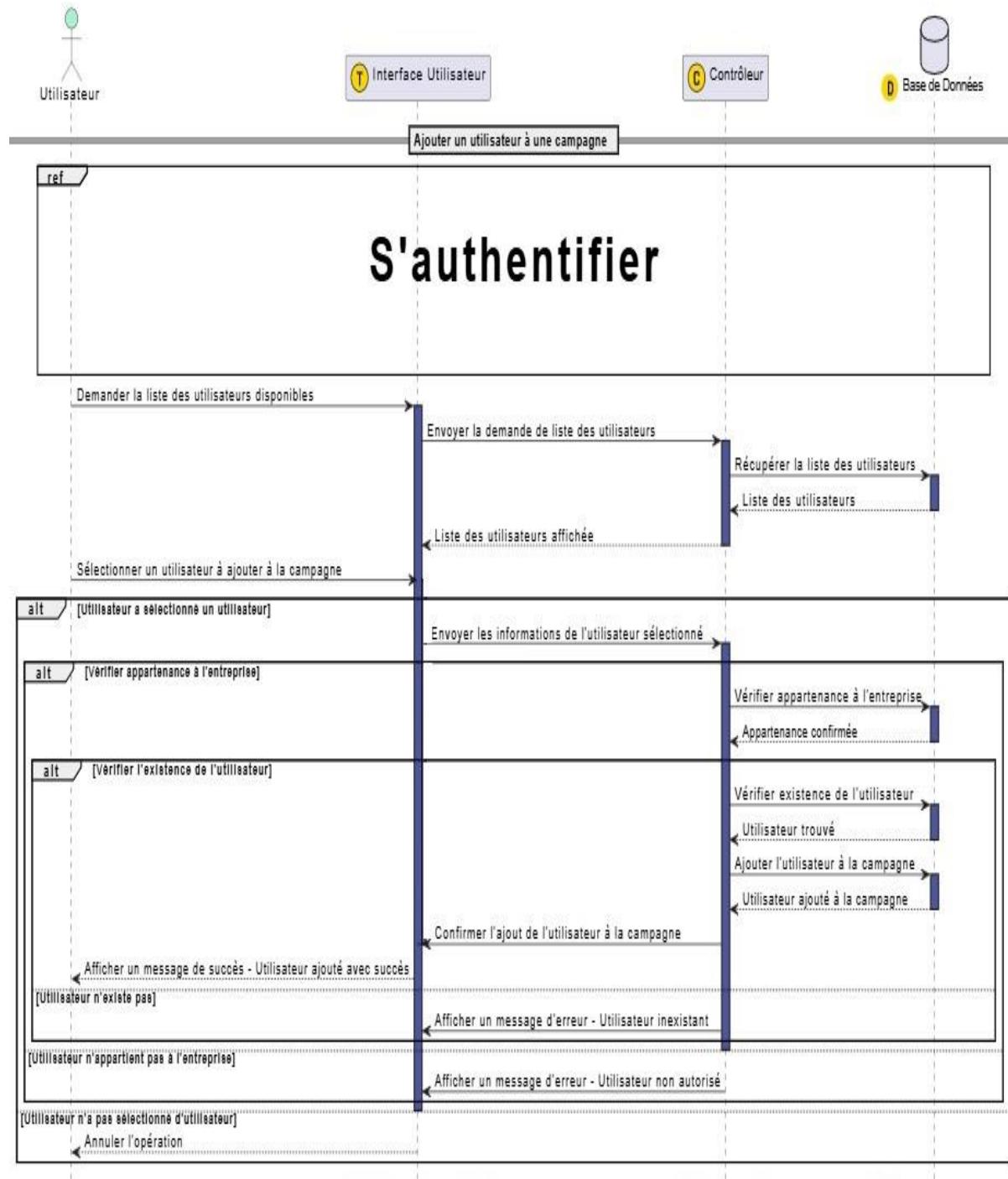


Figure 50- Diagramme de séquence du cas d'utilisation «Ajouter Utilisateur à une campagne »

6.4.3.2 Diagramme de séquence du cas d'utilisation « Crée rôle »

Dans le diagramme présenté ci-dessous, nous éludions l'interaction spécifique entre l'utilisateur et les différentes composantes du système, visant à permettre la création d'un rôle.

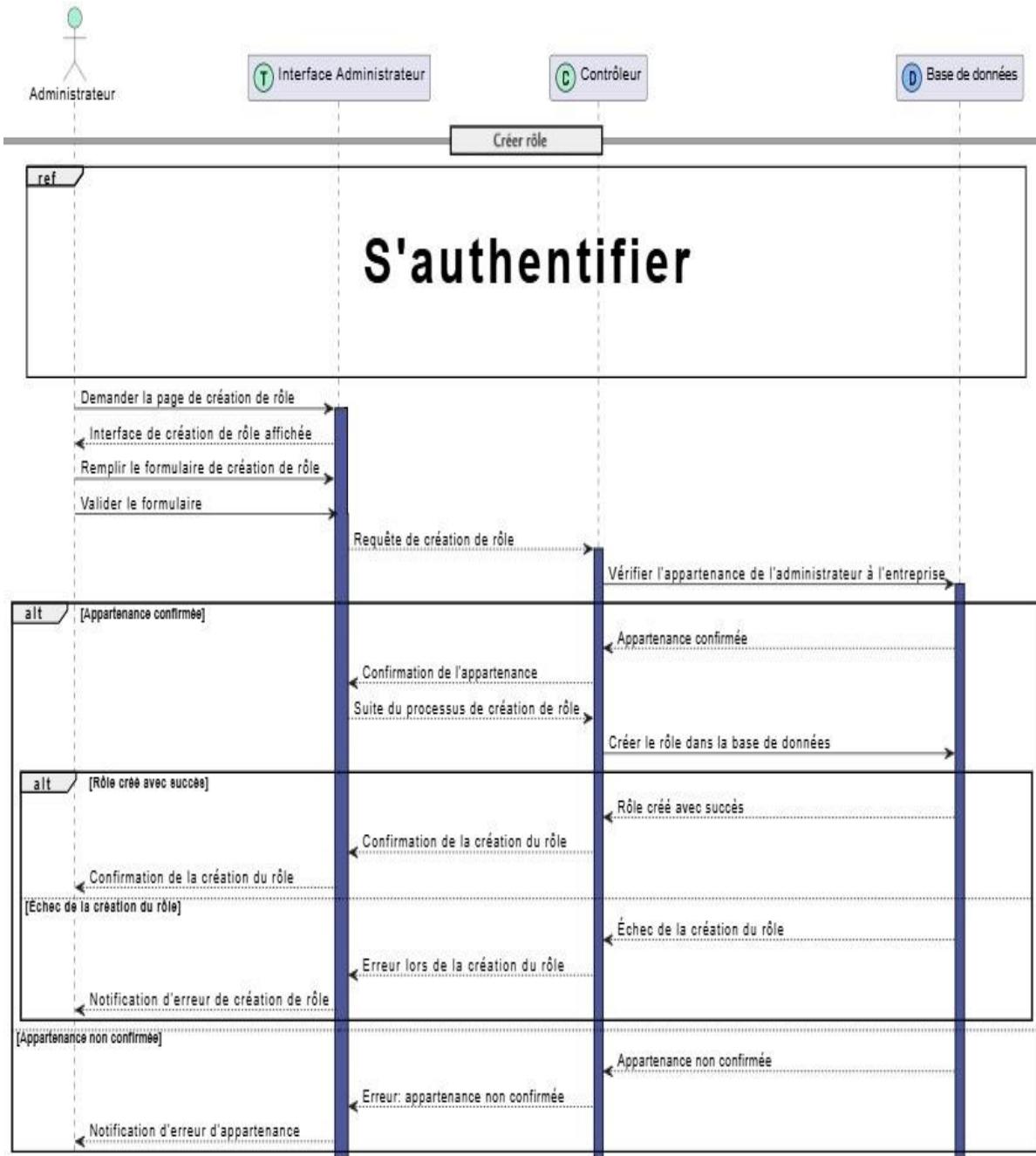


Figure 51- Diagramme de séquence du cas d'utilisation « Crée rôle »

6.4.4 Diagramme de classes

Pour le sprint 4, nous avons suivi une méthodologie similaire en présentant d'abord les diagrammes de séquence pour illustrer les interactions dynamiques entre les acteurs et le système dans des contextes spécifiques. Par la suite, nous avons complété cette vision par la conception d'un diagramme de classes présenté dans la figure 52.

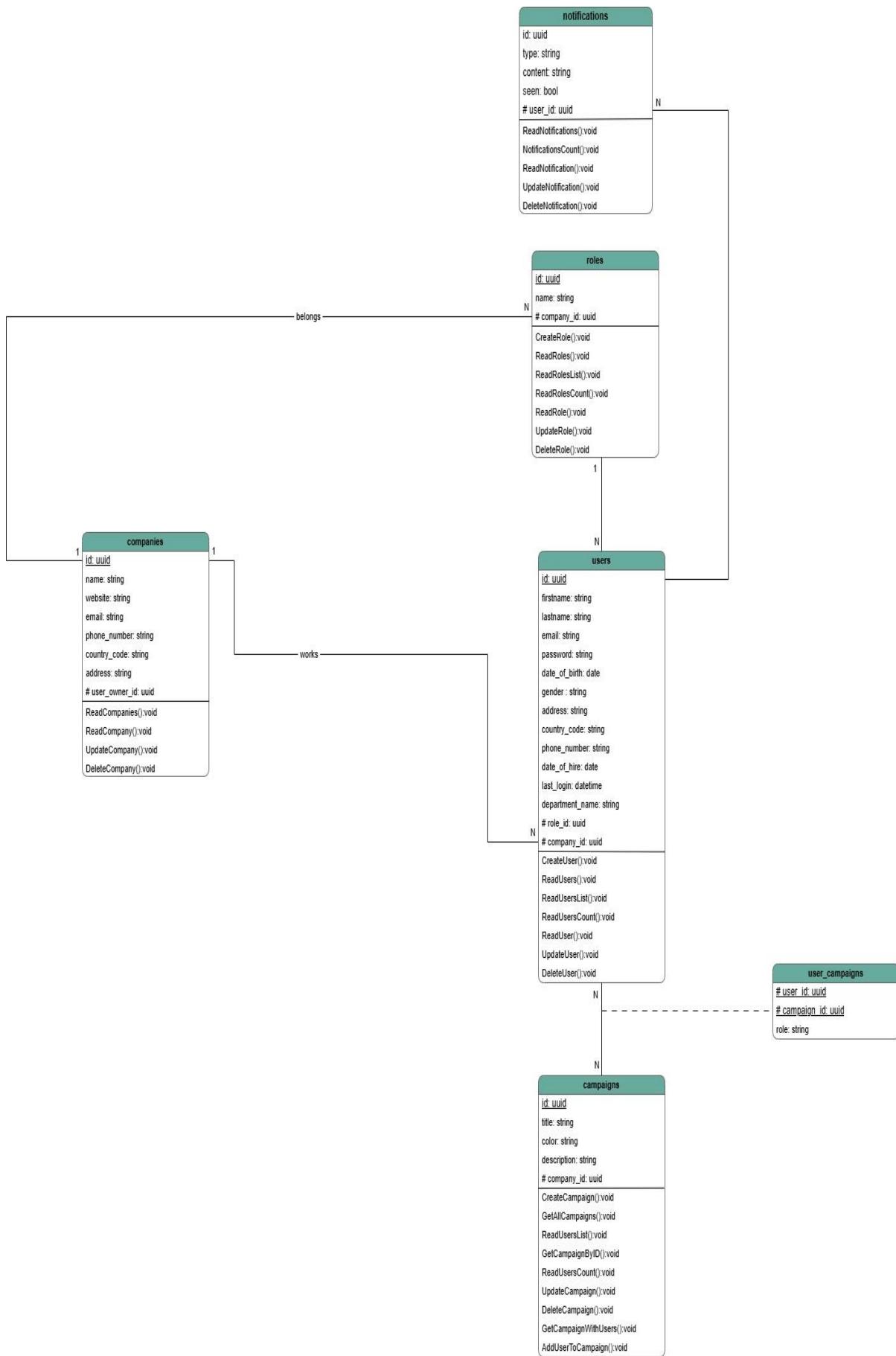


Figure 52- Diagramme de classes du Sprint 4

6.5 Réalisation

Pendant cette du sprint 4, nous donnons vie à nos idées en les transformant en actions tangibles, passant ainsi de la phase de conception à celle de la réalisation. Notre objectif principal est de concrétiser les fonctionnalités établies dans le backlog du sprint à travers l'élaboration des maquettes graphiques puis des interfaces utilisateur.

6.5.1 Modélisation des maquettes

La création de maquettes d'interface utilisateur offre plusieurs avantages. En premier lieu, elle permet de visualiser l'apparence de l'application pour le client, ce qui facilite l'adaptation de l'interface pour répondre à ses besoins spécifiques. De plus, les maquettes aident à clarifier les attentes du produit et encouragent la collaboration. Vous trouverez ci-dessous quelques exemples de maquettes graphiques de notre quatrième sprint.

6.5.1.1 Le Prototype relatif à l'histoire utilisateur « Collaborer avec d'autres utilisateurs »

Le prototype suivant illustre l'histoire utilisateur "Collaborer avec d'autres utilisateurs". Il répond au scénario d'utilisation : En tant qu'utilisateur, je souhaite pouvoir collaborer avec d'autres utilisateurs dans le cadre d'un projet.

Edit collaborators

Collaborators are informed about important actions according to their notification settings.

Select all

<input type="checkbox"/>	user 1
<input checked="" type="checkbox"/>	user 2
<input type="checkbox"/>	user 3
<input checked="" type="checkbox"/>	user 4

CANCEL SAVE

Figure 53- Prototype relatif à l'histoire utilisateur « Collaborer avec d'autres utilisateurs »

6.5.1.2 Le Prototype relatif à l'histoire utilisateur « Créer un nouvel rôle utilisateur »

Le prototype suivant illustre l'histoire utilisateur "Créer un nouvel rôle utilisateur". Il répond au scénario d'utilisation : En tant qu'administrateur, je souhaite pouvoir créer un nouvel rôle utilisateur.

add Role

Name

Company Name

mm/dd/yyyy --:-- --

SAVE CANCEL

Figure 54- Prototype relatif à l'histoire utilisateur « Crée un nouvel rôle utilisateur »

6.5.2 Développement des interfaces utilisateurs

Dans cette partie, nous nous concentrerons sur la présentation de quelques interfaces utilisateur développées lors de notre quatrième sprint.

6.5.2.1 Interface graphique relative à l'histoire utilisateur « Collaborer avec d'autres utilisateurs »

Cette fonctionnalité est dédiée à la collaboration entre utilisateurs, permettant à l'utilisateur (administrateur ou utilisateur standard) de travailler conjointement avec d'autres utilisateurs sur un projet, en partageant des tâches et des informations pour une meilleure coordination et efficacité.

Edit collaborators

Collaborators are informed about important actions according to their notification settings.

Select all

	Amin Mallek	<input type="checkbox"/>
	Mohamed Amin Mallek	<input type="checkbox"/>
	Biranji Dubey	<input type="checkbox"/>
	Adnen Zammel	<input type="checkbox"/>
	Ala Zammel	<input type="checkbox"/>

CANCEL SAVE

Figure 55- Interface graphique relative à l'histoire utilisateur « Collaborer avec d'autres utilisateurs »

6.5.2.2 Interface graphique relative à l 'histoire utilisateur « Crée un nouvel rôle utilisateur »

Cette fonctionnalité est dédiée à la création de nouveaux rôles utilisateur, permettant à l'administrateur de définir des rôles spécifiques avec des permissions distinctes, afin de mieux structurer les responsabilités et accès au sein du système.

The screenshot shows a modal dialog titled "add Role". It contains three input fields: "Name" (placeholder "Name"), "Company Name" (placeholder "Company Name"), and a date input field (placeholder "mm/dd/yyyy --:-- --"). At the bottom right are two buttons: a dark blue "Save" button and a white "CANCEL" button.

Figure 56- Interface graphique relative à l 'histoire utilisateur « Crée un nouvel rôle utilisateur »

6.6 Test fonctionnel

Dans ce sprint final, notre priorité est de nous concentrer sur les tests fonctionnels pour assurer la qualité du produit avant sa livraison. Cela comprend la mise en place et l'exécution des principaux scénarios de test spécifiques à ce sprint, afin de valider l'intégralité des fonctionnalités développées.

Cas de test	Démarche	Comportement attendu	Résultat
Test d'ajout d'un utilisateur à une campagne.	L'utilisateur se rend dans l'interface "Ajouter un utilisateur" et vérifie si la liste des utilisateurs s'affiche correctement. Ensuite, il sélectionne un utilisateur donné. Après avoir confirmé que la liste des utilisateurs de la campagne est mise à jour, il peut continuer avec les actions requises.	L'utilisateur est ajouté avec succès à la liste des utilisateurs de la campagne.	Conforme

Cas de test	Démarche	Comportement attendu	Résultat
Test de création d'un nouvel rôle utilisateur	<p>L'utilisateur se rend sur la fonction "Entreprise" dans l'interface d'administration.</p> <p>Il sollicite alors le formulaire de création de rôle et vérifie que celui-ci s'affiche correctement, incluant un champ dédié au nom du rôle. Après avoir saisi le nom du rôle, l'utilisateur valide le formulaire..</p>	<p>Accès à la liste des rôles. Si le rôle est créé, il sera ajouté à la liste existante.</p>	Conforme

6.7 Conclusion

Dans ce chapitre, nous avons conclu le dernier sprint de notre projet. Cette étape s'est achevée par l'évaluation d'une nouvelle livraison fonctionnelle, rigoureusement testée et validée par le propriétaire du produit.

Conclusion Générale

À travers ce projet de fin d'études intitulé SocialScope, nous avons exploré et mis en œuvre une solution complète de gestion des médias sociaux, visant à centraliser et optimiser les interactions des utilisateurs avec diverses plateformes sociales. Tout au long des différents sprints, nous avons suivi une méthodologie Agile, nous permettant d'adapter continuellement notre approche et de répondre efficacement aux besoins changeants des utilisateurs.

Chaque sprint a été marqué par des avancées significatives. Nous avons débuté par une analyse approfondie des besoins, suivie par la conception et le développement de fonctionnalités essentielles telles que la création et la gestion de publications, la génération de rapports et de statistiques, et la gestion des rôles et des collaborations. Chaque phase a été validée par des tests rigoureux et des retours constructifs de notre Scrum Master et du Product Owner, assurant ainsi la qualité et la robustesse de notre solution.

L'utilisation de technologies modernes telles qu'Angular pour le frontend et Golang pour le backend, couplée à l'intégration d'APIs comme Facebook Graph API et Twitter API v2, a permis de créer une application performante et scalable. PostgreSQL a été choisi pour sa fiabilité en tant que base de données, garantissant une gestion efficace et sécurisée des données.

L'un des points forts de ce projet a été notre capacité à travailler en équipe, à collaborer et à surmonter les défis techniques et organisationnels. Chaque membre de l'équipe a contribué avec son expertise et son dévouement, faisant de SocialScope une solution aboutie et prête à être déployée dans des environnements réels.

En conclusion, ce projet de fin d'études a été une expérience qui a non seulement élargi nos compétences techniques et méthodologiques, mais qui a également démontré notre capacité à gérer de manière efficace un projet complexe d'A à Z. Nous sommes fiers des résultats obtenus et sommes convaincus que SocialScope offrira une réelle valeur ajoutée aux utilisateurs dans la gestion de leurs campagnes sur les réseaux sociaux. Ce succès nous motive à continuer sur cette lancée, à explorer de nouvelles opportunités et à continuer à innover dans le domaine des technologies de l'information.

C'est avec une grande satisfaction et un regard optimiste vers l'avenir que nous clôturons ce projet, prêts à relever de nouveaux défis et à apporter une contribution significative au monde professionnel.

Bibliographie

- [1] Manifeste Agile : Valeurs et principes par Blog Gestion de Projet. Adresse en ligne:
<https://blog-gestion-de-projet.com/>
- [2] Rôles dans Scrum par Atlassian. Adresse en ligne:
<https://www.atlassian.com/agile/scrum/roles>
- [3] Introduction à UML par Edrawsoft. Adresse en ligne:
<https://www.edrawsoft.com/uml-diagrams.html>
- [4] UML 2 par Saliha Yacoub. Adresse en ligne:
<https://www.slideshare.net/slideshow/chapter2-uml-and-uml-diagramspdf/266241159>
- [5] Besoins fonctionnels et non fonctionnels par Savoir Plus. Adresse en ligne:
<https://savoir.plus/besoins-fonctionnels-non-fonctionnels/>
- [6] Architecture Client-Serveur par OpenClassrooms. Adresse en ligne:
<https://openclassrooms.com/fr/courses/7210131-definissez-votre-architecture-logicielle-grace-aux-standards-reconnus/7370196-apprenez-larchitecture-client-serveur>
- [7] App.diagrams.net. Outil en ligne: <https://app.diagrams.net/>
- [8] Go (Golang) - Le Language de programmation de Google. Adresse en ligne:
<https://go.dev/>
- [9] Go Solutions proposées par Google. Adresse en ligne: <https://go.dev/>
- [10] Go Documentation de la bibliothèque standard. Adresse en ligne:
<https://go.dev/src/go/doc/comment/std.go>

- [11] Source d'informations sur les packages et modules Go Package Documentation.
Adresse en ligne: <https://pkg.go.dev/about>
- [12] Ce guide de création, publication sur votre page Facebook en tant que page.
Adresse en ligne: <https://developers.facebook.com/docs/pages-api/posts/>
- [13] Twitter API v2 Guide de migration. Adresse en ligne:
<https://developer.twitter.com/en/docs/twitter-api/migrate/overview>
- [14] Swagger Go API (Swaggo) by SoberKoder. Adresse en ligne:
<https://www.soberkoder.com/tags/swagger/>
- [15] TypeScript - JavaScript qui évolue. Adresse en ligne:
<https://www.typescriptlang.org/docs/handbook/intro.html>
- [16] Angular Documentation. Adresse en ligne: <https://angular.io/docs>
- [17] Angular CLI - L'outil CLI pour Angular Package on npm. Adresse en ligne:
<https://www.npmjs.com/package/@angular/cli>
- [18] Tailwind CSS: Un framework CSS axé sur l'utilitaire. Adresse en ligne:
<https://v2.tailwindcss.com/docs>
- [19] Est une bibliothèque officielle de composants Flowbite. Adresse en ligne:
<https://flowbite-svelte.com/docs/components/device-mockups>
- [20] Bibliothèque d'icônes SVG gratuite par Tailwind Labs. Adresse en ligne:
<https://heroicons.com/>