

# Dédicaces

Je dédie ce travail à :

À la mémoire de mon cher père Samir, dont son existence a été une source de force inébranlable. Tout ce que j'ai accompli est le fruit de son soutien, et je lui serai éternellement reconnaissante.

À ma chère mère Wassila, une source inépuisable de tendresse et d'amour, un symbole de courage et de force inébranlable. Malgré toutes les difficultés, elle n'a jamais cessé de m'encourager et de prier pour moi.

À toute ma famille, à tous mes proches et mes amis Un grand merci pour vos encouragements continus, votre soutien moral, pour les agréables moments que nous avons passés. Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite.

À tout ceux qui m'ont donné un jour un coup de pousse, de près ou de loin. Je dédie ce modeste travail, en signe de ma vive reconnaissance et mon profond estime.

# **Remerciements**

Je tiens à exprimer ma sincère gratitude envers toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet, et je souhaite partager ces mots de reconnaissance avant de commencer mon rapport.

En premier lieu, je souhaite exprimer ma reconnaissance envers les membres du jury qui ont accepté d'évaluer mon projet de stage de fin d'études.

Ensuite, je tiens à exprimer mes vifs remerciements à mon encadrante technique au sein de l'entreprise SOFRECOM Tunisie, Mme. Nouha ARBI. Son accompagnement constant et ses précieux conseils ont été essentiels tout au long de ce projet. J'ai eu la chance d'avoir une superviseure qui a été une source constante de motivation et d'expertise.

Je n'oublie pas de remercier mon encadrant académique de l'ESPRIT, Mr. Mohamed Selmène BEN YAHIA. Ses conseils avisés, sa disponibilité pour les échanges et ses remarques constructives ont grandement enrichi mon travail.

Je tiens également à exprimer ma gratitude envers le corps enseignant de L'École supérieure privée d'ingénierie et de technologie , qui nous a fourni des bases solides pour aborder notre première expérience professionnelle.

Enfin, je souhaite exprimer ma profonde gratitude envers l'ensemble de l'équipe technique et administrative de SOFRECOM, ainsi qu'à l'équipe SCRIBE. Ils ont créé des conditions de travail exceptionnelles, un environnement propice à la réussite, et une atmosphère agréable qui ont grandement contribué à la réalisation de ce projet.

# Table des matières

<b>Liste des figures</b>	<b>vi</b>
<b>Liste des tableaux</b>	<b>ix</b>
<b>Terminologie</b>	<b>xi</b>
<b>Liste des abréviations</b>	<b>xiv</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Contexte général</b>	<b>3</b>
1.1 Organisme d'accueil . . . . .	3
1.1.1 SIGA . . . . .	3
1.1.2 Secteur d'activité . . . . .	4
1.2 Présentation du Projet . . . . .	4
1.2.1 Présentation du Portail de Gestion des Approbations . . . . .	4
1.2.2 Spécificités du Portail de Gestion des Approbations . . . . .	5
1.2.3 Problématique . . . . .	5
1.2.4 Situation actuelle et Critiques . . . . .	5
1.3 Solution proposée . . . . .	6
1.4 Méthodologie de gestion de projet . . . . .	8
1.4.1 Méthodologies agiles . . . . .	8
1.4.2 Méthode Scrum . . . . .	8
1.5 Langage de modélisation : UML . . . . .	11
1.6 Outils adoptés pour la gestion de projet . . . . .	11
<b>2 Sprint 0 : Analyse et Spécification des besoins</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Capture des besoins . . . . .	13
2.2.1 Identification des acteurs . . . . .	13
2.2.2 Identification des besoins fonctionnels . . . . .	14
2.2.3 Identification des cas d'utilisation . . . . .	15
2.2.4 Identification des besoins non fonctionnels . . . . .	16
2.3 Diagramme de cas d'utilisation globale . . . . .	17
2.4 Diagramme de classe global . . . . .	18

2.5	Backlog du Produit . . . . .	18
2.6	Planification de projet . . . . .	20
2.6.1	Diagramme de Gantt . . . . .	20
2.7	Environnement de travail . . . . .	21
2.7.1	Environnement matériel . . . . .	21
2.7.2	Environnement logiciel . . . . .	22
2.8	Choix technologiques . . . . .	24
2.8.1	Frontend . . . . .	24
2.8.2	Backend . . . . .	24
2.8.3	Workflow . . . . .	24
2.8.4	Base de données . . . . .	25
2.8.5	CI/CD . . . . .	25
2.8.6	Conteneurisation . . . . .	25
2.8.7	Orchestration . . . . .	26
<b>3</b>	<b>Sprint 1 : Accès et administration de base</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Backlog de Sprint 1 . . . . .	27
3.3	Raffinement des cas d'utilisation . . . . .	28
3.3.1	Identification des acteurs du premier sprint . . . . .	28
3.3.2	Raffinement du cas d'utilisation «S'authentifier» . . . . .	28
3.3.3	Raffinement du cas d'utilisation «Se déconnecter» . . . . .	29
3.3.4	Raffinement du cas d'utilisation «Gérer les utilisateurs» . . . . .	29
3.3.5	Raffinement du cas d'utilisation «Mise en place et configuration de Camunda» . . . . .	32
3.4	Conception . . . . .	33
3.4.1	Diagramme de classe du sprint 1 . . . . .	33
3.4.2	Conception du cas d'utilisation «S'authentifier» . . . . .	34
3.4.3	Conception du cas d'utilisation «Se déconnecter» . . . . .	35
3.4.4	Conception du cas d'utilisation «Gérer les utilisateurs» . . . . .	37
3.5	Réalisation . . . . .	39
3.5.1	S'authentifier . . . . .	39
3.5.2	Se déconnecter . . . . .	39
3.5.3	Gérer les utilisateurs . . . . .	40
3.6	Conclusion . . . . .	44
<b>4</b>	<b>Sprint 2 : Gestion des demandes</b>	<b>45</b>
4.1	Introduction . . . . .	45
4.2	Backlog du Sprint 2 . . . . .	45
4.3	Raffinement de cas d'utilisation . . . . .	46
4.3.1	Identification des acteurs du deuxième sprint . . . . .	46
4.3.2	Raffinement du Cas d'Utilisation «Gérer son Profil» . . . . .	46
4.3.3	Raffinement du Cas d'Utilisation «Réinitialiser son Mot de Passe» . . . . .	47
4.3.4	Raffinement du Cas d'Utilisation «Soumettre une Demande» . . . . .	48
4.3.5	Raffinement du Cas d'Utilisation «Consulter ses Demandes» . . . . .	49

4.3.6	Raffinement du Cas d'Utilisation «Consulter ses Congés» . . . . .	50
4.3.7	Raffinement du Cas d'Utilisation «Traiter une Demande» . . . . .	51
4.4	Conception . . . . .	51
4.4.1	Diagramme de classe du sprint 2 . . . . .	51
4.4.2	Conception du Cas d'Utilisation «Gérer son Profil» . . . . .	52
4.4.3	Conception du Cas d'utilisation «Réinitialiser son mot de passe» . . . . .	54
4.4.4	Conception du Cas d'Utilisation «Soumettre une Demande» . . . . .	56
4.4.5	Conception du Cas d'Utilisation «Consulter ses demandes» . . . . .	59
4.4.6	Diagramme de Séquence . . . . .	59
4.4.7	Conception du Cas d'Utilisation «Consulter ses congés» . . . . .	60
4.4.8	Conception du Cas d'Utilisation «Traiter une demande» . . . . .	61
4.5	Réalisation . . . . .	63
4.5.1	Gérer son profil . . . . .	63
4.5.2	Réinitialiser son mot de passe . . . . .	63
4.5.3	Soumettre une demande . . . . .	66
4.5.4	Consulter ses demandes . . . . .	67
4.5.5	Consulter ses congés . . . . .	68
4.5.6	Traiter une demande . . . . .	68
4.6	Conclusion . . . . .	69
<b>5</b>	<b>Sprint 3 : Suivi et Supervision</b>	<b>70</b>
5.1	Introduction . . . . .	70
5.2	Backlog du Sprint 3 . . . . .	70
5.3	Raffinement de cas d'utilisation . . . . .	71
5.3.1	Identification des acteurs du troisième sprint . . . . .	71
5.3.2	Raffinement du cas d'utilisation «Recevoir une notification» . . . . .	71
5.3.3	Raffinement du cas d'utilisation «Consulter les processus métiers» . . . . .	72
5.3.4	Raffinement du cas d'utilisation «Consulter les demandes d'approbation» .	73
5.3.5	Raffinement du cas d'utilisation «Consulter les membres de l'équipe» . .	75
5.3.6	Raffinement du cas d'utilisation «Consulter ses crédits» . . . . .	75
5.4	Conception . . . . .	76
5.4.1	Diagramme de classe du sprint 3 . . . . .	77
5.4.2	Conception du cas d'utilisation «Recevoir une notification» . . . . .	78
5.4.3	Conception du cas d'utilisation «Consulter les processus métiers» . . . . .	79
5.4.4	Conception du cas d'utilisation «Consulter les demandes d'approbation» .	80
5.4.5	Conception du cas d'utilisation «Consulter les membres de l'équipe» . .	81
5.4.6	Conception du cas d'utilisation «Consulter ses crédits» . . . . .	82
5.5	Réalisation . . . . .	83
5.5.1	Recevoir une notification . . . . .	83
5.5.2	Consulter les processus métiers . . . . .	83
5.5.3	Consulter les demandes d'approbation . . . . .	84
5.5.4	Consulter les membres de l'équipe . . . . .	86
5.5.5	Consulter ses crédits . . . . .	86
5.6	Conclusion . . . . .	87

<b>6 Sprint 4 : Analyse et Améliorations</b>	<b>88</b>
6.1 Introduction . . . . .	88
6.2 Backlog du Sprint 4 . . . . .	88
6.3 Raffinement de cas d'utilisation . . . . .	89
6.4 Raffinement de cas d'utilisation . . . . .	89
6.4.1 Identification des acteurs du quatrième sprint . . . . .	89
6.4.2 Raffinement du cas d'utilisation «Consulter le calendrier d'équipe» . . . . .	89
6.4.3 Raffinement du cas d'utilisation «Consulter les tâches accomplies» . . . . .	90
6.4.4 Raffinement du cas d'utilisation «Consulter les rapports» . . . . .	91
6.4.5 Raffinement du cas d'utilisation «Implémentation d'un chatbot» . . . . .	92
6.5 Conception . . . . .	94
6.5.1 Diagramme de classe du sprint 4 . . . . .	94
6.5.2 Conception du Cas d'Utilisation «Consulter le calendrier d'équipe» . . . . .	94
6.5.3 Conception du Cas d'Utilisation «Consulter les tâches accomplies» . . . . .	96
6.5.4 Conception du Cas d'Utilisation «Consulter les rapports» . . . . .	96
6.5.5 Conception du Cas d'Utilisation «Implémentation d'un chatbot» . . . . .	97
6.6 Réalisation . . . . .	99
6.6.1 Consulter le calendrier d'équipe . . . . .	99
6.6.2 Consulter les tâches accomplies . . . . .	100
6.6.3 Consulter les rapports . . . . .	100
6.6.4 Implémentation d'un chatbot . . . . .	101
6.7 Conclusion . . . . .	103
<b>Conclusion générale</b>	<b>104</b>
<b>Nétographie</b>	<b>105</b>
<b>Bibliographie</b>	<b>109</b>

# Liste des figures

1.1 Logo de SIGA . . . . .	3
1.2 Le cycle de vie en Scrum . . . . .	9
1.3 Interface de ClickUp . . . . .	11
1.4 Interface de ClickUp . . . . .	12
2.1 Les acteurs . . . . .	14
2.2 Diagramme de cas d'utilisation globale . . . . .	17
2.3 Diagramme de classe globale . . . . .	18
2.4 Diagramme de Gantt . . . . .	21
2.5 Logo de Git . . . . .	22
2.6 Logo de Github . . . . .	22
2.7 Logo de WampServer . . . . .	22
2.8 Logo de phpMyAdmin . . . . .	23
2.9 Logo de VSCode . . . . .	23
2.10 Logo de IntelliJ . . . . .	23
2.11 Logo d'Angular . . . . .	24
2.12 Logo de Spring Boot . . . . .	24
2.13 Logo de Camunda . . . . .	24
2.14 Logo de MySQL . . . . .	25
2.15 Logo d'ArgoCD . . . . .	25
2.16 Logo de Docker . . . . .	25
2.17 Logo de Kubernetes . . . . .	26
3.1 Diagramme du cas d'utilisation «S'authentifier» . . . . .	28
3.2 Diagramme du cas d'utilisation «Se déconnecter» . . . . .	29
3.3 Diagramme du cas d'utilisation «Gérer les utilisateurs» . . . . .	29
3.4 Diagramme de classe du sprint 1 . . . . .	33
3.5 Diagramme de classe du cas d'utilisation «S'authentifier» . . . . .	34
3.6 Diagramme de séquence du cas d'utilisation «S'authentifier» . . . . .	35
3.7 Diagramme de classe du cas d'utilisation «Se déconnecter» . . . . .	36
3.8 Diagramme de séquence du cas d'utilisation «Se déconnecter» . . . . .	36
3.9 Diagramme de classe du cas d'utilisation «Gérer les utilisateurs» . . . . .	37
3.10 Diagramme de séquence du cas d'utilisation «Gérer les utilisateurs» . . . . .	38
3.11 Interface du cas d'utilisation «S'authentifier» . . . . .	39
3.12 Interface du cas d'utilisation «Se déconnecter» . . . . .	40

3.13	Interface du cas d'utilisation «Ajouter un utilisateur» . . . . .	40
3.14	Interface du cas d'utilisation «Consulter les utilisateurs» . . . . .	41
3.15	Interface du cas d'utilisation «Consulter les détails d'un utilisateur» . . . . .	41
3.16	Interface du cas d'utilisation «Modifier un utilisateur» . . . . .	42
3.17	Interface du cas d'utilisation «Supprimer un utilisateur» . . . . .	42
3.18	Interface d'intégration de Camunda . . . . .	43
3.19	Interface de configuration de Camunda . . . . .	43
3.20	Interface cockpit de Camunda . . . . .	44
4.1	Diagramme du cas d'utilisation «Gérer le profil» . . . . .	46
4.2	Diagramme du cas d'utilisation «Réinitialiser son Mot de Passe» . . . . .	47
4.3	Diagramme du cas d'utilisation «Soumettre une Demande» . . . . .	48
4.4	Diagramme du cas d'utilisation «Consulter ses Demandes» . . . . .	49
4.5	Diagramme du cas d'utilisation «Consulter ses Congés» . . . . .	50
4.6	Diagramme du cas d'utilisation «Traiter une Demande» . . . . .	51
4.7	Diagramme de classe du Sprint 2 . . . . .	52
4.8	Diagramme de classe du cas d'utilisation «Gérer son Profil» . . . . .	52
4.9	Diagramme de séquence du cas d'utilisation «Gérer son Profil» . . . . .	53
4.10	Diagramme de classe du cas d'utilisation «Réinitialiser son Mot de Passe» . . . . .	54
4.11	Diagramme de séquence du cas d'utilisation «Réinitialiser son Mot de Passe» . . . . .	55
4.12	Diagramme de classe du cas d'utilisation «Soumettre une demande» . . . . .	56
4.13	Diagramme de séquence du cas d'utilisation «Soumettre une demande» . . . . .	57
4.14	Diagramme BPMN demande de congé . . . . .	58
4.15	Diagramme BPMN demande d'autorisation . . . . .	58
4.16	Diagramme de classe du cas d'utilisation «Consulter ses demandes» . . . . .	59
4.17	Diagramme de séquence du cas d'utilisation «Consulter ses demandes» . . . . .	59
4.18	Diagramme de classe du cas d'utilisation «Consulter ses congés» . . . . .	60
4.19	Diagramme de séquence du cas d'utilisation «Consulter ses congés» . . . . .	60
4.20	Diagramme de classe du cas d'utilisation «Traiter une demande» . . . . .	61
4.21	Diagramme de séquence du cas d'utilisation «Traiter une demande» . . . . .	62
4.22	Interface du cas d'utilisation «Gérer son profil» . . . . .	63
4.23	Page "Mot de passe oublié ?" . . . . .	64
4.24	Confirmation de l'envoi du lien de réinitialisation . . . . .	64
4.25	E-mail de réinitialisation de mot de passe . . . . .	65
4.26	Page de création d'un nouveau mot de passe . . . . .	65
4.27	Page de création d'une nouvelle demande de congé . . . . .	66
4.28	Page de création d'un nouvelle demande d'autorisation . . . . .	67
4.29	Page de consultation des demandes personnelles . . . . .	67
4.30	Page de consultation des congés acceptés . . . . .	68
4.31	Page de traitement des demandes . . . . .	69
5.1	Diagramme du cas d'utilisation «Recevoir une notification» . . . . .	71
5.2	Diagramme du cas d'utilisation «Consulter les processus métiers» . . . . .	72
5.3	Diagramme du cas d'utilisation «Consulter les demandes d'approbation» . . . . .	73
5.4	Diagramme du cas d'utilisation «Consulter les membres de l'équipe» . . . . .	75

5.5	Diagramme du cas d'utilisation «Consulter ses crédits» . . . . .	76
5.6	Diagramme de classe du Sprint 3 . . . . .	77
5.7	Diagramme de classe du cas d'utilisation «Recevoir une notification» . . . . .	78
5.8	Diagramme de séquence du cas d'utilisation «Recevoir une notification» . . . . .	78
5.9	Diagramme de classe du cas d'utilisation «Consulter les processus métiers» . . . . .	79
5.10	Diagramme de séquence du cas d'utilisation «Consulter les processus métiers» . . . . .	79
5.11	Diagramme de classe du cas d'utilisation «Consulter les demandes d'approbation» . . . . .	80
5.12	Diagramme de séquence du cas d'utilisation «Consulter les demandes d'approbation» . . . . .	80
5.13	Diagramme de classe du cas d'utilisation «Consulter les membres de l'équipe» . . . . .	81
5.14	Diagramme de séquence du cas d'utilisation «Consulter les membres de l'équipe» . . . . .	81
5.15	Diagramme de classe du cas d'utilisation «Consulter ses crédits» . . . . .	82
5.16	Diagramme de séquence du cas d'utilisation «Consulter ses crédits» . . . . .	82
5.17	Interface du cas d'utilisation «Recevoir une notification» . . . . .	83
5.18	Interface du cas d'utilisation «Consulter les processus métiers» . . . . .	84
5.19	Interface du cas d'utilisation «Consulter les demandes d'approbation» . . . . .	85
5.20	Interface du cas d'utilisation «Consulter les demandes d'approbation» . . . . .	85
5.21	Interface du cas d'utilisation «Consulter les membres de l'équipe» . . . . .	86
5.22	Interface du cas d'utilisation «Consulter ses crédits» . . . . .	87
6.1	Diagramme du cas d'utilisation «Consulter le calendrier d'équipe» . . . . .	89
6.2	Diagramme du cas d'utilisation «Consulter les tâches accomplies» . . . . .	90
6.3	Diagramme du cas d'utilisation «Consulter les rapports» . . . . .	91
6.4	Diagramme de classe du Sprint 4 . . . . .	94
6.5	Diagramme de classe du cas d'utilisation «Consulter le calendrier d'équipe» . . . . .	95
6.6	Diagramme de séquence du cas d'utilisation «Consulter le calendrier d'équipe» . . . . .	95
6.7	Diagramme de classe du cas d'utilisation «Consulter les tâches accomplies» . . . . .	96
6.8	Diagramme de séquence du cas d'utilisation «Consulter les tâches accomplies» . . . . .	96
6.9	Diagramme de classe du cas d'utilisation «Consulter les rapports» . . . . .	97
6.10	Diagramme de séquence du cas d'utilisation «Consulter les rapports» . . . . .	97
6.11	Diagramme de classe du cas d'utilisation «Implémentation d'un chatbot» . . . . .	98
6.12	Diagramme de séquence du cas d'utilisation «Implémentation d'un chatbot» . . . . .	98
6.13	Interface du cas d'utilisation «Consulter le calendrier d'équipe» . . . . .	99
6.14	Interface du cas d'utilisation «Consulter le calendrier d'équipe (voir details)» . . . . .	99
6.15	Interface du cas d'utilisation «Consulter les tâches accomplies» . . . . .	100
6.16	Interface du cas d'utilisation «Consulter les rapports» . . . . .	100
6.17	Interface du cas d'utilisation «Consulter les rapports» . . . . .	101
6.18	Interface du cas d'utilisation «Implémentation d'un chatbot» . . . . .	102
6.19	Interface du cas d'utilisation «Implémentation d'un chatbot» . . . . .	102
6.20	Interface du cas d'utilisation «Implémentation d'un chatbot» . . . . .	103

# Liste des tableaux

2.1	tab:Les rôles des acteurs . . . . .	14
2.2	Backlog du Produit - Partie 1 . . . . .	19
2.3	Backlog du Produit - Partie 2 . . . . .	20
2.4	Spécifications des machines utilisées . . . . .	21
3.1	Backlog du Sprint 1 . . . . .	27
3.2	Description textuelle du cas d'utilisation «S'authentifier» . . . . .	28
3.3	Cas d'utilisation : Se déconnecter . . . . .	29
3.4	Description textuelle du cas d'utilisation « Ajouter un utilisateur » . . . . .	30
3.5	Description textuelle du cas d'utilisation « Consulter les utilisateurs » . . . . .	30
3.6	Description textuelle du cas d'utilisation « Consulter les détails d'un utilisateur » .	31
3.7	Description textuelle du cas d'utilisation « Modifier un utilisateur » . . . . .	31
3.8	Description textuelle du cas d'utilisation « Supprimer un utilisateur » . . . . .	32
3.9	Description textuelle du cas d'utilisation «Mise en place et configuration de Camunda» . . . . .	32
4.1	Backlog du Sprint 2 . . . . .	45
4.2	Description textuelle du Cas d'utilisation «Consulter le Profil» . . . . .	46
4.3	Description textuelle du Cas d'utilisation «Modifier le Profil» . . . . .	47
4.4	Description textuelle du Cas d'utilisation «Réinitialiser son Mot de Passe» . . . . .	48
4.5	Description textuelle du Cas d'utilisation «Soumettre une Demande» . . . . .	49
4.6	Description textuelle du Cas d'utilisation «Consulter ses Demandes» . . . . .	50
4.7	Description textuelle du Cas d'utilisation «Consulter ses Congés» . . . . .	50
4.8	Description textuelle du Cas d'utilisation «Traiter une Demande» . . . . .	51
5.1	Backlog du Sprint 3 : Suivi et Supervision . . . . .	70
5.2	Description textuelle du Cas d'utilisation «Recevoir une notification par mail» . . .	71
5.3	Description textuelle du Cas d'utilisation «Recevoir une notification push» . . . . .	72
5.4	Description textuelle du Cas d'utilisation «Consulter les processus métiers» . . . . .	73
5.5	Description textuelle du Cas d'utilisation «Consulter les demandes d'approbation»	74
5.6	Description textuelle du Cas d'utilisation «Consulter les détails d'une demande d'approbation» . . . . .	74
5.7	Description textuelle du Cas d'utilisation «Consulter les membres de l'équipe» . .	75
5.8	Description textuelle du Cas d'utilisation «Consulter ses crédits» . . . . .	76
6.1	Backlog du Sprint 4 : Fonctionnalités avancées . . . . .	88

6.2	Description textuelle du Cas d'utilisation «Consulter le calendrier d'équipe» . . . . .	90
6.3	Description textuelle du Cas d'utilisation «Consulter les tâches accomplies» . . . . .	91
6.4	Description textuelle du Cas d'utilisation «Consulter les rapports» . . . . .	92
6.5	Description textuelle du Cas d'utilisation «Implémentation d'un chatbot» . . . . .	93

# Terminologie

**Automatisation** : c'est lorsque un processus est complètement automatisé et ne requiert aucune intervention d'un utilisateur.

**Environnement de développement** : n'est pas seulement un IDE (Integrated Development Environment) comme Eclipse ou Visual Studio mais les bibliothèques, les fichiers de configuration et les serveurs en font partie aussi.

**Intégration** : est la combinaison des parties de code source séparées dont le but est de déterminer comment elles fonctionnent comme un tout.

**Build** : présente l'ensemble d'activités dont l'objectif est la construction de l'application. Il comprend plusieurs tâches comme :

- La mise à jour du code source depuis le gestionnaire de version
- La compilation du code source de l'application
- La génération des artefacts

**Une tâche (Job)** : est un ensemble d'instructions que le runner doit exécuter. [1] L'ensemble des jobs est défini dans un fichier YAML. Nous pouvons voir en temps réel le résultat d'une tâche afin que les développeurs puissent savoir si le job a été exécuté avec succès ou bien il a échoué. Un job peut être automatiquement lancé lorsqu'un commit est poussé ou bien manuellement exécuté. En effet, le job manuel peut être utile, par exemple, pour automatiser un déploiement, mais ne le déployer que lorsque quelqu'un l'approuve manuellement. En plus, il existe des moyens qui limitent le nombre de personnes pouvant exécuter un job.

**Artefact (Artifact)** : est généré par un job en cas d'exécution avec succès. Les utilisateurs peuvent télécharger cet artefact pour le tester ou bien l'utiliser directement [1].

**Pipeline** : est un groupe de jobs exécutés par étapes. En effet, tous les jobs d'une même étape sont exécutés en parallèle s'il y a suffisamment de Runners simultanés disponibles [3]. Le pipeline passe à l'étape suivante lorsque les jobs en cours terminent leur exécution sans erreurs. En cas d'échec d'au moins un job, l'exécution de pipeline s'arrête.

**Runner** : est une machine virtuelle, un container Docker ou un cluster de containers. Son rôle principal est de personnaliser l'environnement d'exécution du job [3]. En particulier, GitLab Runner est un projet Open Source utilisé dans la fonctionnalité d'intégration continue GitLab. Ce dernier communique avec le Runner à travers une API pour exécuter les scripts écrits dans la configuration GitLab-CI.

**GOROCO** : permet de récupérer la valeur exacte de la version [4]. Il se décompose en:

1. Generation: la génération du logiciel
2. Revision: les services pack (le cas d'un nombre impair implique que la version a encore des Bugs à éliminer pour la révision suivante)
3. Correction: les mises à jour

# Liste des abréviations

**BRMC:** Cloud privé d'Orange France

**CI:** Continuous Integration

**CD:** Continuous Deployment

**DSIF:** Direction des Services d'Information France

**DTSI:** Direction Technique et du Système d' Information

**IHM:** Interface Homme-Machine

**IT:** Information Technology

**ODE:** Orange Deployment Engine

**PLA:** Produit Logiciel Applicatif

**PLI:** Produit Logiciel Infrastructure

**SCM:** Software Configuration Management

**SI:** Systèmes d'Information

**SQAAS:** SonarQube As A Service

**URL:** Uniform Resource Locator

**YML:** Yet Another Markup Language

# Introduction générale

À une époque où les technologies de l'information redessinent les contours de notre quotidien, leur influence sur les organisations est devenue incontournable. Depuis janvier 2023, le nombre d'internautes a franchi la barre des 5,181 milliards à l'échelle mondiale, illustrant une adoption massive du numérique. Cette révolution technologique impose aux entreprises de repenser leurs processus pour gagner en agilité et en efficacité. Parmi ces processus, la gestion des approbations – qu'il s'agisse de demandes de congés, d'achats ou de budgets – se distingue par sa complexité. Souvent fragmentée, impliquant plusieurs acteurs et sujette aux erreurs, elle représente un défi que les outils traditionnels peinent à relever.

C'est dans ce contexte que SIGA, une entreprise spécialisée dans les solutions informatiques innovantes, s'impose comme un acteur clé pour accompagner les organisations vers une transformation numérique réussie. Reconnue pour son expertise, SIGA développe des systèmes qui simplifient les opérations tout en répondant aux exigences d'un monde en constante évolution. Mon projet de fin d'études s'inscrit pleinement dans cette mission : la création d'un « Portail de Gestion des Approbations avec Workflows Dynamiques et Déploiement Automatisé ». Ce portail vise à centraliser et fluidifier la gestion des demandes, en permettant aux utilisateurs de les soumettre via une interface intuitive, de suivre leur progression en temps réel, et de bénéficier d'une validation automatisée grâce à des workflows dynamiques.

Pour concrétiser cette vision, j'ai mobilisé un éventail de technologies modernes : Angular pour une interface utilisateur conviviale, Spring Boot pour un backend robuste, et Camunda pour orchestrer des processus flexibles et évolutifs. En adoptant une approche DevOps, j'ai intégré un pipeline CI/CD avec Jenkins pour automatiser les tests et le déploiement, tout en utilisant Docker pour la conteneurisation et Kubernetes pour garantir la scalabilité. Ces choix techniques reflètent un double objectif : offrir une expérience utilisateur optimale tout en assurant une solution fiable et maintenable pour les équipes techniques.

Ce rapport retrace les étapes de cette réalisation au sein de SIGA, structuré en sept chapitres qui suivent la progression du projet, depuis son contexte jusqu'à son déploiement final.

Le premier chapitre explore le contexte général du projet. Nous présenterons SIGA, ses ambitions, et les spécificités du portail, avant d'analyser la problématique des processus d'approbation et de poser les bases du projet.

Le deuxième chapitre, correspondant au Sprint 0, se concentre sur l'analyse et la spécification des besoins. Nous définirons les exigences fonctionnelles et non fonctionnelles, ainsi que les bases de la modélisation des workflows.

Le troisième chapitre, Sprint 1, aborde l'accès et l'administration de base du portail. Nous détaillerons la mise en place des fonctionnalités initiales, telles que l'authentification et la gestion des utilisateurs, en utilisant Angular et Spring Boot.

Le quatrième chapitre, Sprint 2, est dédié à la gestion des demandes. Nous présenterons la soumission, le suivi et la validation des demandes, en intégrant les workflows dynamiques avec Camunda.

Le cinquième chapitre, Sprint 3, se focalise sur le suivi et la supervision. Nous explorerons les fonctionnalités permettant aux utilisateurs de suivre les processus, consulter les demandes d'approbation et gérer les crédits de congés.

Le sixième chapitre, Sprint 4, met l'accent sur l'analyse et les améliorations. Nous décrirons les outils d'analyse avancés, comme la consultation du calendrier d'équipe, des tâches accomplies et des rapports, ainsi que l'intégration d'un chatbot pour assister les administrateurs.

Le septième chapitre conclut le projet en abordant la partie DevOps et le déploiement. Nous détaillerons l'utilisation de 'ArgoCD' pour l'intégration continue, Docker pour la conteneurisation, et Kubernetes pour l'orchestration, garantissant une scalabilité et une fiabilité optimales, accompagnées de la documentation associée.

En conclusion de cette introduction, ces sept chapitres forment un récit cohérent qui reflète les différentes étapes de notre travail. Ensemble, ils illustrent comment ce portail, né d'une problématique concrète, a évolué pour devenir une solution complète, alignée sur les besoins des utilisateurs et les standards technologiques actuels. Ce projet au sein de SIGA a été une opportunité unique de conjuguer créativité, rigueur technique et collaboration, dans un effort pour rendre les processus d'approbation plus simples et plus intelligents.

# Chapitre 1

## Contexte général

### Introduction

Dans ce premier chapitre, nous allons poser les bases de notre travail en explorant son contexte général. Nous débuterons par une présentation de l'organisme d'accueil, la société SIGA, en mettant en lumière son domaine d'expertise et son rôle. Ensuite, nous introduirons le projet en détail, avant d'examiner l'existant afin d'identifier ses limites et les opportunités d'amélioration que notre solution vise à concrétiser. Enfin, nous exposerons la méthodologie suivie pour concevoir ce projet ainsi que les grandes lignes de la solution proposée.

### 1.1 Organisme d'accueil

#### 1.1.1 SIGA

Créée en 1996, SIGA (Système Informatique et Gestion Automatisée) s'est imposée comme un acteur majeur dans le domaine du développement de logiciels et des solutions informatiques en Tunisie. Basée à Tunis, l'entreprise regroupe une équipe de plus de vingt ingénieurs et six consultants seniors, cumulant entre 5 et 25 ans d'expérience dans la conception et l'intégration de systèmes d'information. Depuis sa fondation, SIGA se distingue par son expertise multidisciplinaire, offrant des services qui couvrent l'ensemble du cycle de vie d'un produit informatique : de l'analyse des besoins à la mise en œuvre, en passant par le développement, la formation et la maintenance.



Figure 1.1: Logo de SIGA

### **1.1.2 Secteur d'activité**

SIGA déploie son expertise dans divers secteurs, proposant des solutions technologiques adaptées aux besoins spécifiques de ses clients. Ses principaux domaines d'intervention incluent :

- **Banque et assurance** : développement de systèmes pour optimiser la gestion des processus financiers et administratifs.
- **Industrie** : conception de logiciels pour améliorer la gestion de la production et des ressources.
- **Transports** : mise en place de solutions facilitant la logistique et la coordination opérationnelle.
- **Télécommunications** : création de systèmes pour rationaliser les interactions et la gestion des données.
- **Autres secteurs** : accompagnement à la digitalisation des opérations pour des entreprises variées, avec une présence étendue en Tunisie et dans plusieurs pays africains (Mali, Côte d'Ivoire, Tchad, Mauritanie).

## **1.2 Présentation du Projet**

Le projet que j'ai réalisé s'inscrit dans le cadre de mon projet de fin d'études en dernière année à l'École Supérieure Privée d'Ingénierie et de Technologie (ESPRIT). J'ai effectué mon stage au sein de SIGA, une entreprise reconnue pour son expertise dans le développement de logiciels et de solutions informatiques innovantes en Tunisie.

Dans cette section, nous présentons notre projet en détaillant, dans un premier temps, un aperçu du « Portail de Gestion des Approbations avec Workflows Dynamiques et Déploiement Automatisé » et ses spécificités. Dans un second temps, nous aborderons la problématique, l'étude de l'existant, notre contribution ainsi que la méthodologie adoptée pour mener à bien cette mission.

### **1.2.1 Présentation du Portail de Gestion des Approbations**

Le projet « Portail de Gestion des Approbations avec Workflows Dynamiques et Déploiement Automatisé » est une solution conçue pour centraliser et optimiser la gestion des processus d'approbation au sein des organisations. Son objectif principal est de simplifier la soumission, le suivi et la validation des demandes – telles que les congés ou les autorisations – en offrant une interface intuitive et des mécanismes automatisés. Ce portail se distingue par sa capacité à adapter dynamiquement les workflows en fonction des besoins spécifiques de chaque demande, garantissant ainsi une flexibilité et une personnalisation optimales.

Le portail prend en charge divers types de demandes et génère des outputs variés, notamment des notifications par email ou push notification, des statuts en temps réel, et des rapports analytiques sous forme de tableaux de bord. À terme, il vise à gérer un volume significatif de requêtes quotidiennes tout en assurant une traçabilité complète des actions effectuées. Développé avec Angular pour le frontend, Spring Boot pour le backend, et Camunda pour l'orchestration des workflows, le système repose sur une architecture moderne déployée via Docker et orchestrée par Kubernetes. Il est hébergé sur plusieurs environnements distincts : développement (DEV), tests (TEST) et production (PROD), avec une infrastructure évolutive adaptée à chaque contexte.

Chez SIGA, ce projet joue un rôle clé dans la digitalisation des processus internes, et toute interruption pourrait affecter la fluidité des opérations, rendant sa disponibilité une priorité essentielle.

### 1.2.2 Spécificités du Portail de Gestion des Approbations

Le portail est un outil de gestion des processus qui permet aux utilisateurs de soumettre et de suivre des demandes via différents canaux et fonctionnalités, notamment :

- **Workflows dynamiques:** des processus personnalisables définis avec Camunda.
- **Suivi en temps réel:** un statut actualisé et un historique des actions.
- **Reporting:** des tableaux de bord pour analyser les processus.
- **Notifications:** des alertes automatiques par email ou temps réel.

### 1.2.3 Problématique

Dans de nombreuses organisations, les processus d'approbation sont encore réalisés de manière manuelle. Cela engendre plusieurs problématiques :

- **Lenteur des traitements:** Les délais sont souvent rallongés à cause des allers-retours entre les différents acteurs.
- **Manque de traçabilité :** Il est difficile de connaître l'état d'avancement d'une demande ou de retracer les actions passées.
- **Risque d'erreurs humaines :** En raison du manque d'automatisation, certaines étapes peuvent être oubliées ou mal exécutées.

### 1.2.4 Situation actuelle et Critiques

Actuellement, pour obtenir une approbation (congé, autorisation...), le processus suit généralement les étapes suivantes :

1. Remplissage manuel d'un formulaire papier ou d'un document bureautique (Word, Excel).
2. Transmission physique ou par email au supérieur hiérarchique.
3. Approbation manuelle, parfois verbale, sans traçabilité.
4. Notification au demandeur uniquement si ce dernier relance.

## Inconvénients:

- **Non-centralisation** : Les demandes sont éparpillées (emails, papiers, discussions).
- **Perte de temps** : Les relances sont fréquentes.
- **Absence de supervision globale** : Aucune visibilité d'ensemble pour l'administration.
- **Aucune possibilité d'analyse** : Pas de données statistiques sur les demandes, les délais, les refus, etc.

## 1.3 Solution proposée

Afin de répondre aux limites identifiées dans le processus actuel, il est proposé de mettre en place un **Portail Web de Gestion des Approbations** moderne, centralisé et automatisé. Ce portail repose sur une architecture modulaire et des technologies éprouvées pour assurer une solution performante, évolutive et facilement maintenable.

### Automatisation des workflows avec Camunda BPM

Le cœur de la solution repose sur l'intégration de **Camunda BPM**, un moteur de workflow open-source basé sur le standard BPMN (Business Process Model and Notation). Chaque type de demande (congé, achat, autorisation, etc.) est modélisé sous forme de processus graphique. Ces workflows définissent les différentes étapes à suivre, les rôles impliqués et les règles de validation.

L'utilisation de Camunda permet :

- La **définition claire et visuelle** des processus métiers.
- L'**exécution automatique** des étapes selon les règles définies.
- La **flexibilité** de modifier les processus sans redéployer l'application.
- L'**assignation dynamique** des tâches aux utilisateurs concernés.

### Digitalisation de la soumission des demandes

L'interface utilisateur, développée avec **Angular**, permet aux utilisateurs de soumettre leurs demandes via des formulaires en ligne ergonomiques et adaptés au type de demande. Chaque formulaire est connecté à une instance de processus Camunda qui orchestre automatiquement les étapes suivantes. Cette digitalisation permet de :

- Réduire les erreurs de saisie grâce à des contrôles intégrés.
- Gagner du temps en supprimant les supports papier.
- Standardiser la soumission et le traitement des demandes.

## **Suivi en temps réel et notifications automatiques**

Chaque utilisateur peut suivre en temps réel le statut de ses demandes (approuvée, rejetée ou en cours de traitement). Le portail intègre un système de **notifications en temps réel** directement dans l'interface de l'application, basé sur des technologies de type WebSocket:

- Alerte à l'approbateur lorsqu'une tâche lui est assignée.
- Notification au demandeur une fois la décision prise.
- Rappels pour les tâches non traitées dans un délai défini.

Ce mécanisme renforce la réactivité des acteurs et limite les retards.

## **Infrastructure moderne, scalable et résiliente**

L'ensemble de l'application est conteneurisé à l'aide de **Docker** afin de garantir une portabilité maximale et un déploiement simplifié. Le déploiement se fait sur un cluster **Kubernetes** pour bénéficier :

- D'une haute disponibilité des services.
- D'un équilibrage de charge automatique.
- D'une scalabilité horizontale en fonction de la charge.

## **Pipeline DevOps pour l'intégration et le déploiement continu**

Un pipeline CI/CD est mis en place avec **ArgoCD** pour automatiser :

- La compilation et les tests des composants backend et frontend.
- La construction des images Docker.
- Le déploiement automatique sur l'environnement de production.

Cela permet de garantir la rapidité, la fiabilité et la répétabilité du processus de livraison.

## **Analyse des performances et tableaux de bord**

Des tableaux de bord sont mis à disposition des administrateurs pour visualiser :

- Le nombre de demandes traitées par type ou par service.
- Les délais moyens de traitement.
- Les goulets d'étranglement dans les workflows.

Ces indicateurs permettent d'optimiser les processus métier et de prendre des décisions basées sur des données concrètes.

## **1.4 Méthodologie de gestion de projet**

Afin d'assurer le bon déroulement de notre projet, tout en respectant les contraintes de qualité, de délais et les attentes fonctionnelles, il est essentiel d'adopter une méthodologie adaptée qui structure et optimise les différentes phases de travail.

### **1.4.1 Méthodologies agiles**

Les méthodologies agiles représentent une nouvelle génération d'approches de gestion de projet, axées sur des cycles itératifs, adaptatifs et incrémentaux. Elles permettent une réévaluation régulière des besoins, facilitent l'évolution des exigences et garantissent une livraison progressive et maîtrisée des fonctionnalités. [2].

Ces méthodes reposent sur des principes forts, tels que :

- L'acceptation du changement tout au long du cycle de développement.
- La collaboration constante avec le client, considéré comme partie prenante active.
- L'accent mis sur la valeur fonctionnelle livrée à chaque itération.
- La priorité donnée aux interactions humaines plutôt qu'aux processus rigides.

Les méthodes agiles favorisent ainsi la réactivité, l'amélioration continue et l'adaptabilité face à un environnement en perpétuelle évolution.

### **1.4.2 Méthode Scrum**

Parmi les différentes approches agiles, la méthodologie Scrum s'est imposée comme une référence incontournable. Elle fournit un cadre structuré mais souple permettant d'organiser efficacement le travail d'équipe, tout en maximisant la productivité et la qualité du produit livré.

Nous avons opté pour Scrum, car cette méthodologie répond parfaitement aux besoins des projets complexes en nous offrant une flexibilité optimale et une forte capacité d'adaptation face aux imprévus.

Le contexte méthodologique de Scrum s'articule, tel que le montre la figure 1.2, autour de la définition des fonctions, d'un tempo bien rythmé, d'éléments concrets et des réunions particulières.

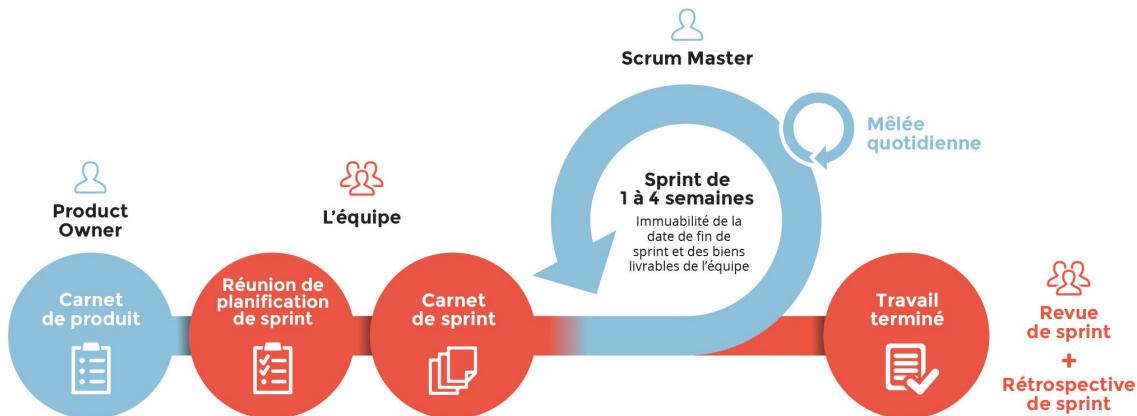


Figure 1.2: Le cycle de vie en Scrum

### Les Rôles :

Scrum définit trois rôles clés, indispensables à la réussite du projet :

- **Le Product Owner** : Il est fréquent qu'un spécialiste de domaine assume le rôle de représentant officiel du client au sein d'un projet Scrum. Son rôle principal consiste à définir les besoins fonctionnels, établit les priorités dans le Product Backlog et veille à la satisfaction du client.
- **Le Scrum Master** : Garant de la méthode Scrum, il facilite le travail de l'équipe en supprimant les obstacles et en s'assurant que les pratiques agiles sont correctement appliquées.
- **L'Équipe** : Les membres responsable de la conception, du développement, des tests et de la livraison des fonctionnalités à la fin de chaque sprint. Ils sont auto-organisés et multidisciplinaires.

### Le Rythme Itératif :

- **Sprint** : Le travail est organisé en cycles courts appelés sprints, d'une durée généralement comprise entre 2 et 4 semaines. À chaque sprint, une version incrémentale du produit est développée, testée et potentiellement livrable.

### Les Artefacts :

- **Product Backlog** : Liste hiérarchisée de toutes les fonctionnalités à développer, rédigées sous forme de User Stories. Ce backlog est régulièrement mis à jour en fonction des retours du client.
- **Sprint Backlog** : Sous-ensemble du Product Backlog sélectionné pour être réalisé lors d'un sprint donné. Il contient les User Stories détaillées et les tâches associées.

## les Evénements :

- **Sprint Planning** : Réunion de planification marquant le début de chaque sprint. L'équipe sélectionne les éléments à développer et définit les objectifs à atteindre.
- **Daily Scrum (Mêlée quotidienne)** : Courte réunion quotidienne, souvent appelée "stand-up" en raison de sa brièveté (15 minutes maximum), visant à synchroniser les efforts de l'équipe. Chaque membre partage trois points : ce qu'il a accompli depuis la dernière mêlée, ce qu'il prévoit de faire avant la prochaine, et les éventuels blocages rencontrés. Facilitée par le Scrum Master, cette rencontre n'est pas destinée à résoudre les problèmes sur-le-champ, mais à les identifier pour un traitement ultérieur. Elle favorise la transparence, renforce la collaboration et permet d'ajuster rapidement le plan du sprint si nécessaire.
- **Definition of Done (DoD)** : Ensemble de critères définissant qu'une tâche est considérée comme terminée (tests réussis, documentation rédigée, code relu, etc.).
- **Definition of Ready (DoR)** : Ce concept repose sur une compréhension commune des préparatifs requis pour une tâche, intégrant une liste de contrôle pour établir les User Stories. Pour être considérées comme prêtes à être mises en œuvre, les User Stories doivent répondre à plusieurs exigences, notamment la description exhaustive des scénarios d'utilisation, la disponibilité des ressources essentielles, des critères d'acceptation clairement définis et vérifiables, une taille adaptée, une présentation à l'équipe par le propriétaire du produit, des méthodes de test fonctionnelles, une estimation fiable de leur taille relative basée sur l'expérience utilisateur, ainsi que la spécification des appareils ciblés.
- **Revue de Sprint** : Réunion organisée à la fin de chaque sprint pour évaluer le travail accompli. L'équipe présente l'incrément terminé au Product Owner et aux parties prenantes, qui valident sa conformité aux attentes. Si nécessaire, des ajustements sont apportés au "Product Backlog" pour refléter l'évolution des besoins ou des priorités. D'une durée maximale de 4 heures pour un sprint d'un mois, cette étape vise à garantir que le projet progresse efficacement, tout en maintenant un alignement constant avec les objectifs stratégiques et les exigences changeantes.
- **Rétrospective de Sprint** : Rencontre tenue immédiatement après la revue, dédiée à l'amélioration continue. L'équipe, guidée par le Scrum Master, analyse le sprint écoulé en identifiant ce qui a bien fonctionné, les difficultés rencontrées, et les opportunités d'optimisation. Des actions concrètes sont définies pour le sprint suivant, comme ajuster les processus ou résoudre des blocages récurrents. Limitée à 3 heures pour un sprint d'un mois, cette réunion renforce l'auto-organisation de l'équipe et assure une progression constante dans la qualité et l'efficacité du travail.

## 1.5 Langage de modélisation : UML

Le langage UML (Unified Modeling Language) est un langage standardisé de modélisation utilisé dans la conception orientée objet des systèmes logiciels. Il permet de représenter, visualiser, spécifier, construire et documenter de manière formelle les différents aspects d'un système, tant structurels que comportementaux.

UML repose sur un ensemble de diagrammes graphiques permettant de décrire les éléments clés d'un système : les classes, les objets, les interactions, les activités, les cas d'utilisation, etc. Ces représentations facilitent la communication entre les différents acteurs du projet (développeurs, analystes, chefs de projet) et garantissent une compréhension commune du fonctionnement global du système.

Dans le cadre de notre projet, nous avons choisi d'utiliser UML pour modéliser les différentes composantes de notre application. Ce choix s'explique par la richesse expressive d'UML, sa capacité à structurer efficacement un système complexe, et sa large adoption dans l'industrie comme outil de conception et de documentation. La figure 1.3 présente le logo du language de modélisation UML.



Figure 1.3: Interface de ClickUp

## 1.6 Outils adoptés pour la gestion de projet

Pour assurer le succès de notre projet et garantir une livraison réussie de notre produit, nous avons mis en place un ensemble d'outils de gestion de projet stratégiques, parmi lesquels nous pouvons citer :

- **Git** : Un système de contrôle de version permettant de suivre les modifications apportées aux fichiers de code, facilitant ainsi une collaboration fluide et efficace entre plusieurs contributeurs.
- **GitLab** : Une plateforme de gestion de dépôts basée sur Git, qui simplifie le suivi et l'organisation de nos ressources tout en optimisant leur gestion.
- **Microsoft Teams** : Un outil collaboratif de communication qui renforce les interactions au sein de l'équipe, favorisant une coordination transparente et une communication améliorée.

- **ClickUp** : Une solution open source de gestion de projet, essentielle pour planifier, suivre et gérer les tâches, contribuant ainsi à une administration efficace et structurée de l'ensemble du projet. La figure 1.4 présente l'interface de ClickUp.

Name	Assignee	Due date	Priority	Comments	Date de tache
ajout de recherche et sort	SA				Feb 11
Reimplémentation du process congé	SA				Feb 12
Ajout interfaces user liste congé et demand...	SA				Feb 17
Ajout de la traduction	SA				Feb 18
ajout interface user congé	SA				Feb 20
ajout liste demande user	SA				Feb 20
ajout autorisation	SA				Feb 24
implementation autorisation front	SA				Feb 26
Ajout calendrier d'équipe	SA				Mar 3
Integration camunda modeler dans NG	SA				Mar 6
Ajout de notification avec webSocket	SA				Mar 7
Correction RBAC for Modeler and rh Confirm	SA				Mar 10
Correction des retours de qqe api backend	SA				Mar 11
Preparation présentation pour restitution e...	SA				Mar 12
Faire des recherches sur l'architecture de d...	SA				Mar 13
Début redaction du rapport	SA				Mar 14
Optimisation du code	SA				Mar 17
redaction rapport	SA				Mar 18
recherches sur argoCD	SA				Mar 21
Conception et rapport	SA				Mar 24
Ajout de mise à jour nom et synchronisation	SA				Mar 25

Figure 1.4: Interface de ClickUp

## Conclusion

Dans ce premier chapitre, nous avons introduit l'organisme d'accueil tout en offrant un aperçu général de la problématique et des objectifs du projet.

Nous avons également décrit la méthodologie retenue ainsi que le formalisme encadrant le processus mis en œuvre pour le développement de notre application.

Le chapitre suivant se concentrera sur l'analyse de la solution existante, en mettant en lumière ses limites et en proposant une nouvelle solution à concevoir.

# Chapitre 2

## Sprint 0 : Analyse et Spécification des besoins

### 2.1 Introduction

Ce chapitre est consacré à l'analyse des besoins pour le développement du Portail de Gestion des Approbations. Nous y identifions les exigences fonctionnelles et non fonctionnelles du système, puis nous illustrons la modélisation à l'aide de diagrammes UML tels que les cas d'utilisation et le diagramme de classes.

Nous présentons ensuite le Backlog produit défini selon la méthodologie Scrum, avant de conclure par une présentation des principaux outils utilisés dans cette phase.

### 2.2 Capture des besoins

Dans cette section nous présentons les acteurs du système, les besoins fonctionnels et non fonctionnels présents dans le projet.

#### 2.2.1 Identification des acteurs

Notre système contient quatre acteurs ,illustrés dans la figure 2.1, qui sont :

- **Administrateur** : Gère l'ensemble du système, incluant les utilisateurs et les configurations globales (e.g., permissions, intégrations).
- **Manager** : Soumet, consulte et traite les demandes (validation/rejet) des employés qu'il supervise.
- **RH (Ressources Humaines)** : Gère les congés, soumet, consulte et traite les demandes, tout en assurant le suivi des politiques RH.
- **Utilisateur** : Employé qui soumet des demandes (congés, absences) et consulte ses propres demandes et soldes de congés.

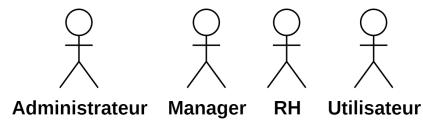


Figure 2.1: Les acteurs

## 2.2.2 Identification des besoins fonctionnels

Les besoins fonctionnels sont les interactions entre l'acteur(Une personne,un matériel ou un logiciel) et le système de manière à exploiter les fonctionnalités de ce dernier. Notre système offre aux acteurs la possibilité de faire certaines tâches.

Le tableau 2.1 illustre le rôle de chaque acteur dans ce système.

Tableau 2.1: tab:Les rôles des acteurs

Administrateur	<ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Se déconnecter.</li> <li>- Gérer les utilisateurs.</li> <li>- Consulter les processus métiers.</li> <li>- Consulter les demandes d'approbations.</li> </ul>
Utilisateur	<ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Se déconnecter.</li> <li>- Consulter ses demandes.</li> <li>- Consulter ses congés.</li> <li>- Soumettre une demande.</li> <li>- Recevoir une notifications.</li> <li>- Consulter les rapports.</li> <li>- Gérer son profil.</li> <li>- Consulter le calendrier d'équipe.</li> <li>- Consulter les membres d'équipe.</li> <li>- Consulter ses crédits.</li> </ul>
Manager / RH	<ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Se déconnecter.</li> <li>- Consulter ses demandes.</li> <li>- Consulter ses congés.</li> <li>- Soumettre une demande.</li> <li>- Recevoir une notifications.</li> <li>- Consulter les rapports.</li> <li>- Gérer son profil.</li> <li>- Consulter le calendrier d'équipe.</li> <li>- Consulter les membres d'équipe.</li> <li>- Consulter ses crédits.</li> <li>- Consulter ses taches accomplies.</li> <li>- Traiter une Demande.</li> </ul>

### 2.2.3 Identification des cas d'utilisation

- **S'authentifier:** L'authentification est le cas d'utilisation qui permet à tous les utilisateurs (Administrateur, Utilisateur, Manager, RH) d'accéder à la plateforme. Il contient deux champs à remplir, qui sont le nom d'utilisateur et le mot de passe, et un bouton de validation pour accéder aux fonctionnalités de la plateforme en cas de succès de l'authentification.
- **Se déconnecter:** Les utilisateurs (Administrateur, Utilisateur, Manager, RH) peuvent se déconnecter du système en cliquant sur le bouton "Déconnexion". Le système vérifie l'existence et la conformité des coordonnées de l'utilisateur avant de le quitter.
- **Gérer les utilisateurs:** À travers ce cas d'utilisation, l'Administrateur est le seul à avoir le droit de gérer les utilisateurs de la plateforme, en incluant l'ajout, la consultation, la modification et la suppression des utilisateurs
- **Consulter les processus métiers:** L'Administrateur a la possibilité de consulter les différents processus métiers définis dans le système pour mieux comprendre les opérations internes et leur suivi.
- **Consulter les demandes d'approbation:** L'Administrateur peut accéder aux demandes d'approbation soumises pour les consulter.
- **Consulter ses demandes:** Les Utilisateurs, les Managers et les RH peuvent consulter l'historique de leurs demandes dans la plateforme.
- **Consulter ses congés:** Les utilisateurs, les Managers et les RH peuvent consulter la liste de leurs congés, passés et à venir, ainsi que leur solde restant.
- **Soumettre une demande:** Les Utilisateurs, les Managers et les RH peuvent soumettre une nouvelle demande via la plateforme. Cette demande peut concerner des congés, des absences ou d'autres requêtes spécifiques.
- **Recevoir une notification:** Ce cas permet aux Utilisateurs, Managers et aux RH de recevoir des notifications concernant les actions importantes, comme la validation d'une demande, ou des rappels de tâches.
- **Consulter les rapports:** Les Utilisateurs, les Managers et les RH peuvent consulter les rapports concernant leurs activités, comme les demandes soumises, les congés, les tâches accomplies, etc.
- **Gérer son profil:** Ce cas permet à chaque acteur (Utilisateur, Manager, RH) de gérer son profil personnel en modifiant ses informations comme son mot de passe, son adresse e-mail, etc.
- **Consulter le calendrier d'équipe:** Ce cas permet à l'Utilisateur, au Manager et au RH de consulter le calendrier de leur équipe, afin de voir les absences, les congés et les événements à venir.

- **Consulter les membres de l'équipe:** Les Utilisateurs, les Managers et les RH peuvent consulter la liste des membres de leur équipe, leurs informations et les tâches qui leur sont attribuées.
- **Consulter ses crédits:** Ce cas permet aux Utilisateurs, aux Managers et aux RH de consulter le solde de leurs crédits de congé ou autres types de crédits sur la plateforme.
- **Consulter ses tâches accomplies:** Les Managers et les RH peuvent consulter les tâches accomplies par ses subordonnés, pour le suivi des activités et des performances.
- **Traiter une demande:** Le Manager et l'RH sont en charge de traiter les demandes soumises, en les validant ou en les rejetant, selon les critères définis dans la plateforme.

#### **2.2.4 Identification des besoins non fonctionnels**

Les besoins non fonctionnels décrivent les contraintes et les exigences de qualité que doit respecter le système. Ils garantissent la performance, la fiabilité et la sécurité de l'application. Voici les principaux besoins non fonctionnels identifiés pour notre projet :

- **Performance :** Le système doit offrir une bonne réactivité, avec un temps de réponse inférieur à 2 secondes pour l'affichage des principales fonctionnalités. Il doit également supporter un nombre élevé de connexions simultanées sans dégradation notable des performances.
- **Sécurité :** Le système doit garantir la confidentialité, l'intégrité et la disponibilité des données. L'authentification des utilisateurs doit être sécurisée, et les accès doivent être strictement contrôlés selon le rôle de chaque utilisateur.
- **Fiabilité :** Le système doit être capable de fonctionner de manière continue sans erreurs critiques, avec un taux de disponibilité supérieur à 99.5%.
- **Scalabilité :** Le système doit pouvoir évoluer facilement pour prendre en charge un plus grand nombre d'utilisateurs ou de données sans refonte majeure de l'architecture.
- **Traçabilité :** Le système doit enregistrer les actions importantes effectuées par les utilisateurs (audit logs) pour permettre un suivi en cas d'incident ou d'investigation.

## 2.3 Diagramme de cas d'utilisation globale

Le diagramme de cas d'utilisation globale illustré dans la figure 2.2 consiste à décrire les fonctions fondamentales et identifie , également la relation et les interactions entre le système et ses acteurs.

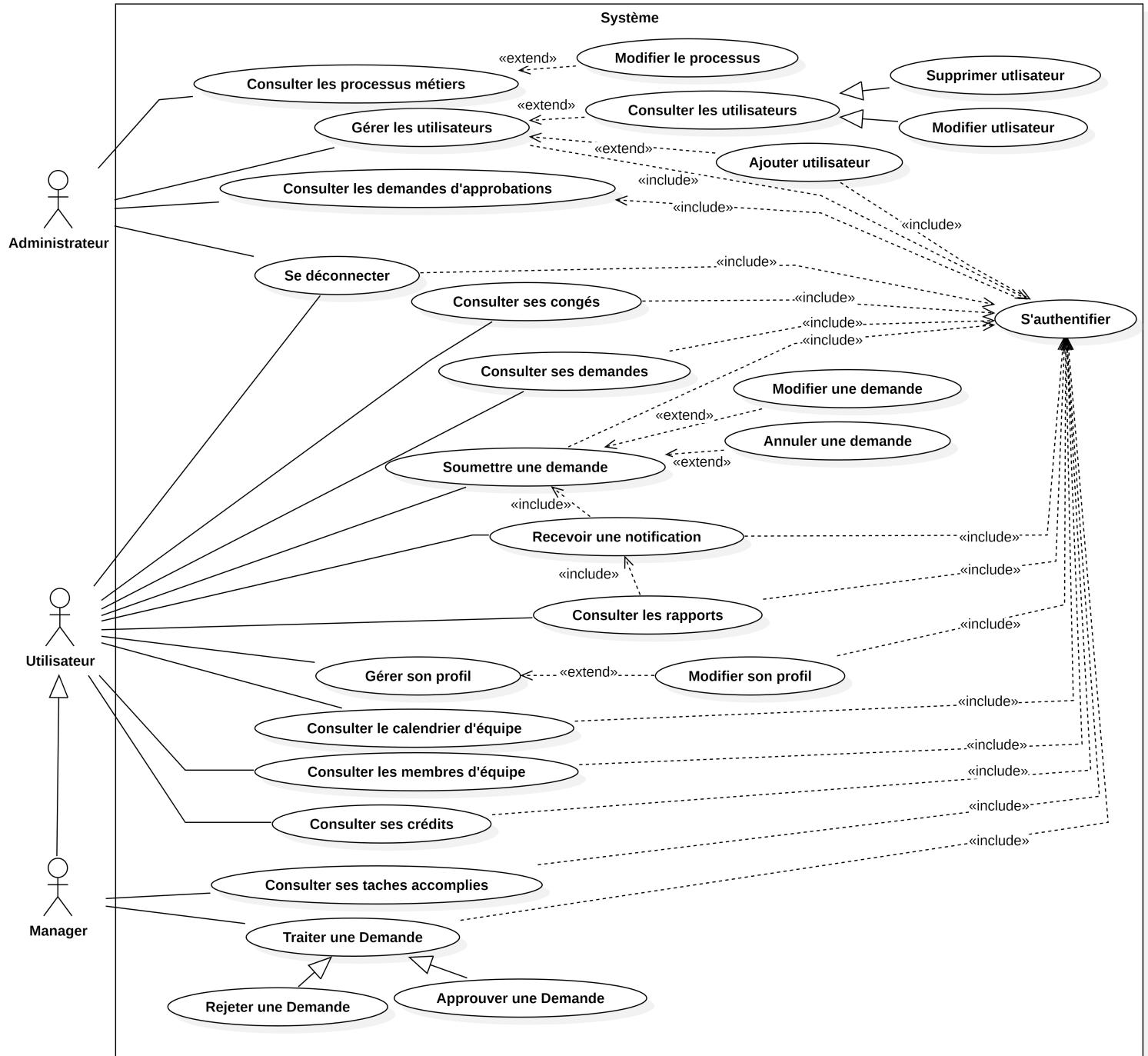


Figure 2.2: Diagramme de cas d'utilisation globale

## 2.4 Diagramme de classe global

Le diagramme de classe globale illustré dans la figure 2.3 montre la constitution du système à l'aide de la modélisation de ses classes, ses attributs, ses opérations et l'identification des relations entre ses objets.

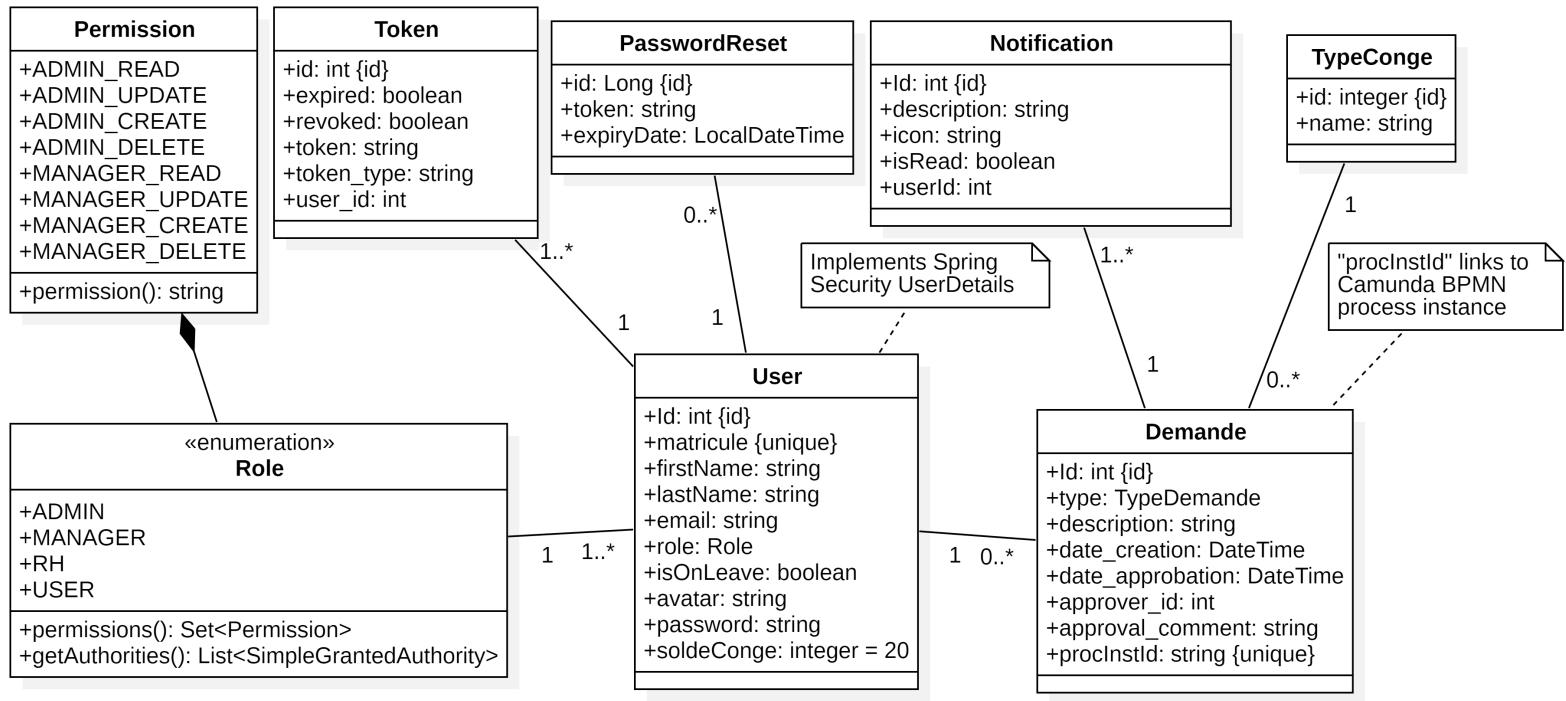


Figure 2.3: Diagramme de classe globale

## 2.5 Backlog du Produit

Le backlog du produit est un ensemble de tâches priorisées qui caractérise un produit. C'est l'un des éléments indispensables de la méthodologie Scrum. C'est l'outil de travail du Product Owner. Ce qui nous mène à le modéliser dans le tableau 2.2 et 2.3 []:

Sprint	User Story	Description	Priorité
Sprint 1 : Accès et Administration de Base	<ul style="list-style-type: none"> <li>• S'authentifier</li> <li>• Se déconnecter</li> <li>• Gérer les utilisateurs</li> <li>• Mise en place et configuration de Camunda</li> </ul>	<ul style="list-style-type: none"> <li>• Permet à tous les utilisateurs d'accéder à la plateforme via nom d'utilisateur et mot de passe.</li> <li>• Permet aux utilisateurs de quitter le système après vérification des coordonnées.</li> <li>• L'Administrateur peut ajouter, consulter, modifier et supprimer des utilisateurs.</li> <li>• Permet de configurer et d'intégrer le moteur Camunda dans l'application Spring Boot pour orchestrer les processus métiers.</li> </ul>	1 1 1 1
Sprint 2 : Gestion des Demandes	<ul style="list-style-type: none"> <li>• Gérer son profil</li> <li>• Réinitialiser son mot de passe</li> <li>• Soumettre une demande</li> <li>• Consulter ses demandes</li> <li>• Consulter ses congés</li> <li>• Traiter une demande</li> </ul>	<ul style="list-style-type: none"> <li>• Chaque acteur modifie ses informations personnelles (mot de passe, e-mail, etc.).</li> <li>• En cas d'oubli, l'utilisateur peut réinitialiser son mot de passe via un lien sécurisé.</li> <li>• Les Utilisateurs, Managers et RH soumettent des demandes (congés, absences, etc.).</li> <li>• Les Utilisateurs, Managers et RH consultent l'historique de leurs demandes.</li> <li>• Les Utilisateurs, Managers et RH consultent leurs congés et leur solde.</li> <li>• Les Managers et RH valident ou rejettent les demandes soumises.</li> </ul>	2 2 2 2 2 2
Sprint 3 : Suivi et Supervision	<ul style="list-style-type: none"> <li>• Recevoir une notification</li> <li>• Consulter les processus métiers</li> <li>• Consulter les demandes d'approbation</li> <li>• Consulter les membres de l'équipe</li> <li>• Consulter ses crédits</li> </ul>	<ul style="list-style-type: none"> <li>• Les Utilisateurs, Managers et RH reçoivent des alertes sur des actions importantes.</li> <li>• L'Administrateur consulte les processus métiers définis dans le système.</li> <li>• L'Administrateur accède aux demandes soumises pour consultation.</li> <li>• Les Utilisateurs, Managers et RH accèdent aux infos des membres de leur équipe.</li> <li>• Les Utilisateurs, Managers et RH vérifient leur solde de congés ou crédits.</li> </ul>	2 3 3 3 2

Tableau 2.2: Backlog du Produit - Partie 1

Sprint	User Story	Description	Priorité
Sprint 4 : Analyse et Améliorations	<ul style="list-style-type: none"> <li>• Consulter le calendrier d'équipe</li> <li>• Consulter les tâches accomplies</li> <li>• Consulter les rapports</li> <li>• Exporter les rapports</li> <li>• Implémentation d'un chatbot</li> </ul>	<ul style="list-style-type: none"> <li>• Les Utilisateurs, Managers et RH consultent les absences et événements d'équipe.</li> <li>• Les Managers et RH suivent les tâches réalisées par leurs subordonnés.</li> <li>• Les Utilisateurs, Managers et RH consultent des rapports sur leurs activités.</li> <li>• Permet d'exporter les rapports (PDF, Excel) pour une utilisation hors ligne.</li> <li>• Permet de développer et d'intégrer un chatbot dans l'application pour assister les utilisateurs dans leurs interactions (consultation, soumission de demandes, etc.).</li> </ul>	3 3 4 4 4
Sprint 5 : Déploiement et Pipeline DevOps	<ul style="list-style-type: none"> <li>• Déployer l'application</li> <li>• Configurer le pipeline CI/CD</li> <li>• Surveiller le déploiement</li> <li>• Gérer les versions</li> </ul>	<ul style="list-style-type: none"> <li>• Permet de déployer l'application sur un serveur ou une plateforme cible.</li> <li>• Met en place un pipeline d'intégration et de déploiement continu pour automatiser les mises à jour.</li> <li>• Les administrateurs vérifient l'état et les performances de l'application déployée.</li> <li>• Permet de suivre et de gérer les différentes versions de l'application déployée.</li> </ul>	5 5 5 5

Tableau 2.3: Backlog du Produit - Partie 2

## 2.6 Planification de projet

Cette section a pour objectif de présenter la planification de notre projet. Pour ce faire, nous utiliserons un diagramme de Gantt qui permettra de visualiser les différentes étapes du projet sous forme de tâches à réaliser, facilitant ainsi le suivi et l'organisation du travail.

### 2.6.1 Diagramme de Gantt

Le diagramme de Gantt présenté par la figure 2.4 nous permet de visualiser les différentes tâches ainsi que leurs durées.

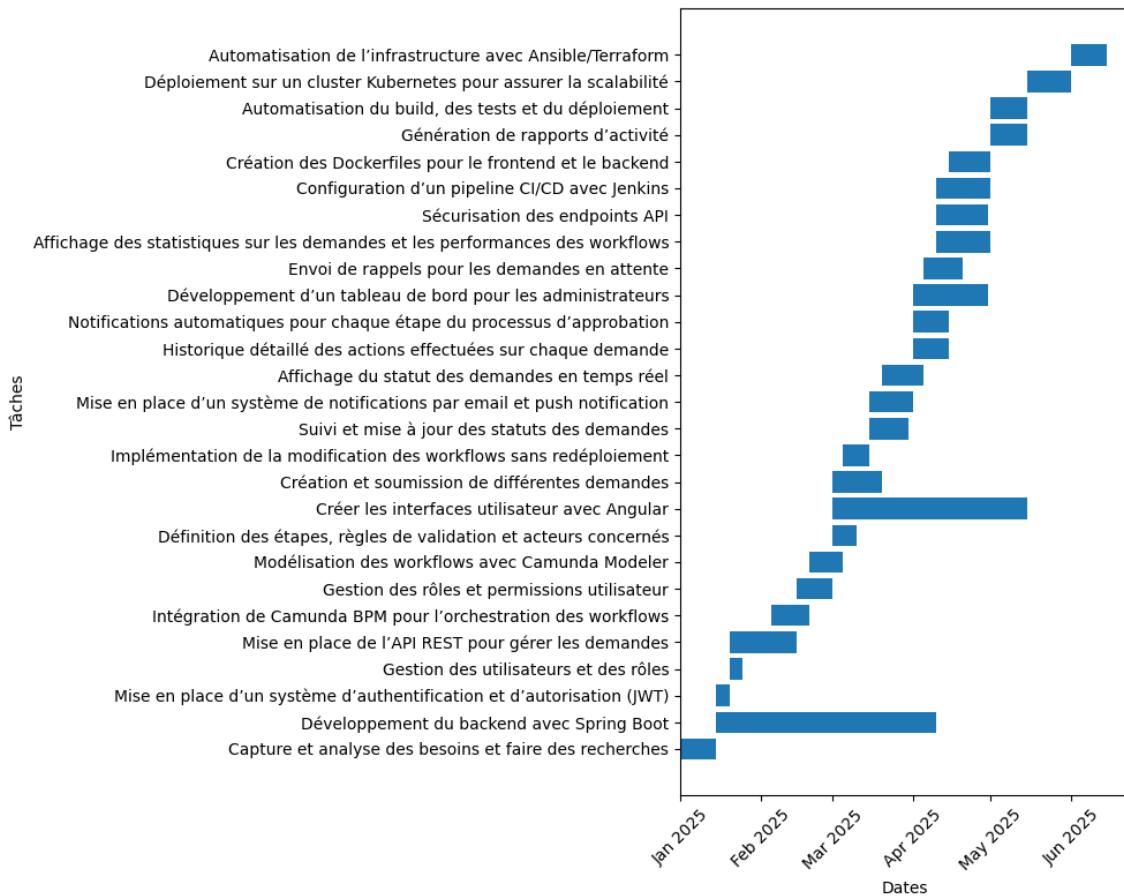


Figure 2.4: Diagramme de Gantt

## 2.7 Environnement de travail

Dans cette section, nous examinerons les différentes outils technologiques adoptées dans le cadre de ce projet.

### 2.7.1 Environnement matériel

Durant ce projet, nous avons utilisé deux machines pour bien mener notre projet. Les caractéristiques techniques de ces machines sont présentées dans le tableau 2.4.

Tableau 2.4: Spécifications des machines utilisées

	<b>Machine 1</b>	<b>Machine 2</b>
<b>Marque</b>	Gigabyte AERO 15	Lenovo IdeaPad
<b>Système d'exploitation</b>	Windows 11 64 bit	Windows 10 64 bit
<b>Processeur</b>	I7-11800H 2.3 GHz	I7-9750H 2.6 GHz
<b>RAM</b>	16 GB 3200MHz	8 GB
<b>Disque Dur</b>	1 TB	500 GB

## 2.7.2 Environnement logiciel

Dans cette section, nous présentons l'environnement logiciel utilisé dans ce projet :

- **Git**: Dont le logo est présenté dans la figure 2.5, cet outil constitue un système de gestion de versions distribué. Il permet de tracer l'évolution des fichiers et des répertoires d'un projet, d'accéder à des versions précédentes et de consulter en détail l'historique des modifications réalisées.



Figure 2.5: Logo de Git

- **Github**: Dont le logo est exposé dans la figure 2.6, est une plateforme collaborative basée sur Git, utilisée pour héberger du code, suivre les changements et faciliter le travail d'équipe sur des projets de développement logiciel.



Figure 2.6: Logo de Github

- **WampServer**: Dont le logo est illustré dans la figure 2.7, WampServer est un environnement de développement web local pour Windows. Il regroupe Apache, MySQL et PHP, permettant aux développeurs de tester et d'exécuter des applications web en local avant leur mise en production.



Figure 2.7: Logo de WampServer

- **phpMyAdmin:** Dont le logo est représenté dans la figure 2.8, phpMyAdmin est une interface web permettant d’administrer facilement des bases de données MySQL ou MariaDB. Il offre aux utilisateurs un accès simplifié pour gérer les tables, exécuter des requêtes SQL, importer ou exporter des données, le tout sans avoir à utiliser la ligne de commande.



Figure 2.8: Logo de phpMyAdmin

- **Visual Studio Code (VSCode):** Représenté dans la figure 2.9, VSCode est un éditeur de code source léger et extensible, populaire pour le développement Angular. Il offre une riche palette d’extensions, telles que des outils de linting, de débogage, et des intégrations Git, ce qui le rend idéal pour travailler sur des applications JavaScript et TypeScript, comme celles développées avec Angular.



Figure 2.9: Logo de VSCode

- **IntelliJ IDEA:** Représenté dans la figure 2.10, IntelliJ IDEA est un environnement de développement intégré (IDE) robuste, principalement utilisé pour le développement d’applications Java, telles que celles créées avec Spring. Il offre des fonctionnalités avancées comme la gestion des dépendances, le débogage intégré, et la prise en charge complète de Spring, ce qui facilite le développement et le déploiement d’applications Spring Boot.



Figure 2.10: Logo de IntelliJ

## 2.8 Choix technologiques

### 2.8.1 Frontend

- **Angular:** Représenté dans la figure 2.11, Angular est un framework de développement web permettant de créer des applications web dynamiques et interactives.



Figure 2.11: Logo d'Angular

### 2.8.2 Backend

- **Spring Boot:** Représenté dans la figure 2.12, Spring Boot est un framework Java qui facilite le développement d'applications Spring en simplifiant la configuration et le déploiement, tout en offrant des outils prêts à l'emploi comme un serveur embarqué et une sécurité intégrée.



Figure 2.12: Logo de Spring Boot

### 2.8.3 Workflow

- **Camunda:** Dont le logo est représenté dans la figure 2.13, Camunda est une plateforme open-source de gestion des processus métier (BPMN), de gestion des décisions (DMN) et de gestion des cas (CMMN). Camunda est particulièrement utilisé pour l'orchestration des workflows complexes et s'intègre facilement dans des architectures Java et microservices.



Figure 2.13: Logo de Camunda

#### 2.8.4 Base de données

- **MySQL:** Représenté dans la figure 2.14, MySQL est un système de gestion de bases de données relationnelles open-source. Il permet de stocker, gérer et interroger des données de manière structurée à l'aide de SQL.



Figure 2.14: Logo de MySQL

#### 2.8.5 CI/CD

- **ArgoCD:** Représenté dans la figure 2.15, ArgoCD est un outil open-source d'intégration et de déploiement continu (CI/CD) spécialement conçu pour Kubernetes. Il permet de déployer et gérer des applications dans des environnements Kubernetes en suivant la méthodologie GitOps.



Figure 2.15: Logo d'ArgoCD

#### 2.8.6 Conteneurisation

- **Docker:** Représenté dans la figure 2.16, Docker est une plateforme open-source permettant de créer, déployer et exécuter des applications dans des conteneurs. Ces conteneurs encapsulent l'application et ses dépendances dans un environnement léger et portable.



Figure 2.16: Logo de Docker

### 2.8.7 Orchestration

- **Kubernetes:** Représenté dans la figure 2.17, Kubernetes est un système open-source d'orchestration de conteneurs qui automatise le déploiement, la gestion, la mise à l'échelle et l'administration des applications conteneurisées. Il permet de gérer efficacement les clusters de conteneurs Docker, offrant des fonctionnalités telles que l'autoscaling, la gestion des ressources et le monitoring des applications.



Figure 2.17: Logo de Kubernetes

## Conclusion

Dans ce deuxième chapitre, nous avons détaillé les besoins fonctionnels et non fonctionnels du projet, ainsi que les différents acteurs et leurs interactions avec le système à travers des diagrammes de cas d'utilisation et de classes globaux. Nous avons également présenté le backlog du produit, structuré en sprints, et défini les priorités pour chaque fonctionnalité. Nous avons également présenté les choix technologiques retenus pour le développement du projet, couvrant l'environnement matériel, l'environnement logiciel. Le chapitre suivant sera consacré à la présentation du Sprint 1, qui couvrira les premières étapes du développement, notamment l'accès et l'administration de base du système.

# Chapitre 3

## Sprint 1 : Accès et administration de base

### 3.1 Introduction

Après avoir tracer les grandes lignes de notre projet, concentrons-nous maintenant sur le premier sprint. Dans ce qui suit, nous expliquerons chaque fonctionnalité de ce sprint en detaillant ses différents besoins, ainsi que sa conception .

### 3.2 Backlog de Sprint 1

Le tableau 3.1 représente le backlog du premier sprint. Ce tableau détaille les cas d'utilisation, leurs priorités, estimations et tâches associées.

Tableau 3.1: Backlog du Sprint 1

Cas d'utilisation	Priorité	Tâche
En tant qu'utilisateur ou admin, je peux m'authentifier	1	Authentifier l'utilisateur a l'aide du token JWT
En tant qu'utilisateur ou admin, je peux me déconnecter	1	Déconnecter l'utilisateur du session
En tant qu'admin, je peux gérer les utilisateurs	1	Ajouter, consulter, modifier, supprimer
En tant que développeur, je peux configurer et intégrer Camunda	1	Configurer dépendances Camunda

### 3.3 Raffinement des cas d'utilisation

Cette section raffine les cas d'utilisation du Sprint 1 en identifiant les acteurs et en détaillant chaque cas d'utilisation.

#### 3.3.1 Identification des acteurs du premier sprint

Les acteurs de ce sprint sont :

**Administrateur** : Acteur principal, il peut s'authentifier, se déconnecter, et gérer les utilisateurs (ajouter, consulter, modifier, supprimer).

**Utilisateur** : Tout acteur ayant un compte, pouvant s'authentifier et se déconnecter.

**Développeur** : Acteur technique qui configure le moteur Camunda dans l'application pour activer les workflows.

#### 3.3.2 Raffinement du cas d'utilisation «S'authentifier»

La figure 3.1 illustre le diagramme de cas d'utilisation « S'authentifier ».

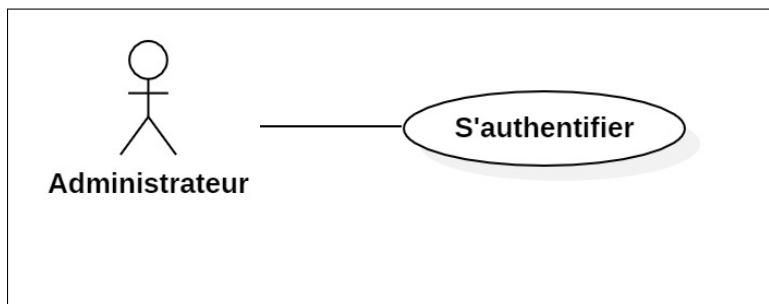


Figure 3.1: Diagramme du cas d'utilisation «S'authentifier»

Le tableau 3.2 illustre la description détaillée du cas d'utilisation ‘s’authentifier’.

Tableau 3.2: Description textuelle du cas d'utilisation «S'authentifier»

<b>Cas d'utilisation</b>	S'authentifier
<b>Acteur</b>	Utilisateur
<b>Pré-conditions</b>	Système en marche, utilisateur inscrit
<b>Post-conditions</b>	Utilisateur authentifié, redirigé selon son rôle
<b>Scénario de base</b>	<ol style="list-style-type: none"><li>1. Affichage de l'interface de connexion</li><li>2. Saisie de l'email et du mot de passe</li><li>3. Clic sur « Se connecter »</li><li>4. Vérification des identifiants</li><li>5. Redirection vers l'accueil</li></ol>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>- Erreur si identifiants incorrects</li><li>- Redirection vers l'authentification si le token JWT est expiré</li></ul>

### 3.3.3 Raffinement du cas d'utilisation «Se déconnecter»

La figure 3.2 illustre le diagramme de cas d'utilisation « Se déconnecter ».

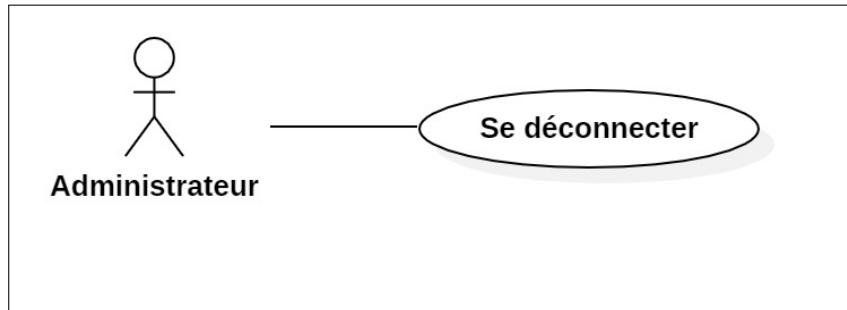


Figure 3.2: Diagramme du cas d'utilisation «Se déconnecter»

Le tableau 3.3 illustre la description détaillée du cas d'utilisation 'Se déconnecter'.

Tableau 3.3: Cas d'utilisation : Se déconnecter

Cas d'utilisation	Se déconnecter
Acteur	Utilisateur
Pré-conditions	Système en marche, utilisateur connecté
Post-conditions	Utilisateur déconnecté
Scénario de base	1. L'utilisateur clique sur « Se déconnecter » 2. Le système redirige vers l'interface d'authentification
Exceptions	- Échec de déconnection (erreur API)

### 3.3.4 Raffinement du cas d'utilisation «Gérer les utilisateurs»

La figure 3.3 illustre le diagramme de cas d'utilisation « Gérer les utilisateurs ».

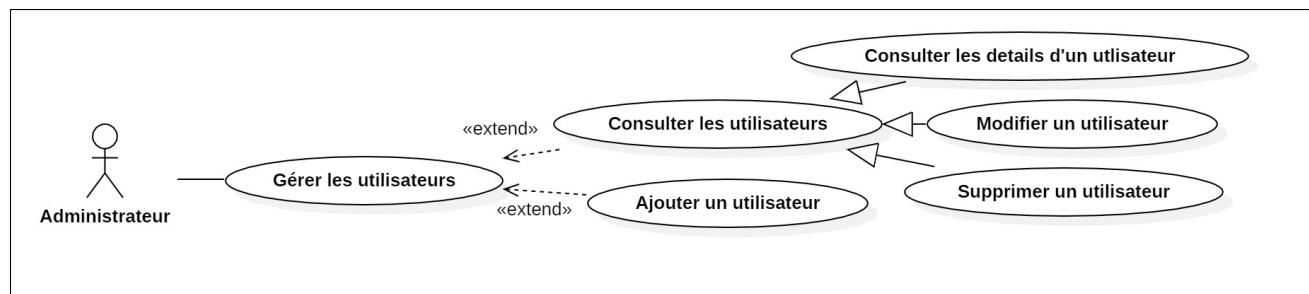


Figure 3.3: Diagramme du cas d'utilisation «Gérer les utilisateurs»

Le tableau 3.4 illustre la description détaillée du cas d'utilisation « Ajouter un utilisateur ».

Tableau 3.4: Description textuelle du cas d'utilisation « Ajouter un utilisateur »

<b>Cas d'utilisation</b>	Ajouter un utilisateur
<b>Acteur</b>	Admin
<b>Pré-conditions</b>	Le système est en marche. L'admin est authentifié.
<b>Post-conditions</b>	L'utilisateur est ajouté.
<b>Scénario de base</b>	<ol style="list-style-type: none"> <li>1. Le système affiche l'interface d'ajout.</li> <li>2. L'admin saisit les données du nouvel utilisateur (nom, email, rôle, etc.).</li> <li>3. L'admin clique sur le bouton « Ajouter ».</li> <li>4. Le système enregistre les informations.</li> <li>5. Le système affiche un message de succès.</li> </ol>
<b>Exceptions</b>	Échec d'ajout (entrées invalides, problème avec l'API POST, ou erreur de base de données).

Le tableau 3.5 illustre la description détaillée du cas d'utilisation « Consulter les utilisateurs ».

Tableau 3.5: Description textuelle du cas d'utilisation « Consulter les utilisateurs »

<b>Cas d'utilisation</b>	Consulter les utilisateurs
<b>Acteur</b>	Admin
<b>Pré-conditions</b>	Le système est en marche. L'admin est authentifié.
<b>Post-conditions</b>	La liste des utilisateurs est consultée.
<b>Scénario de base</b>	<ol style="list-style-type: none"> <li>1. Le système affiche la liste des utilisateurs.</li> <li>2. L'admin consulte la liste.</li> </ol>
<b>Exceptions</b>	Échec de consultation (problème avec l'API GET, ou erreur de base de données).
<b>Extensions</b>	Modifier un utilisateur. Supprimer un utilisateur.

Le tableau 3.6 illustre la description détaillée du cas d'utilisation « Consulter les détails d'un utilisateur ».

Tableau 3.6: Description textuelle du cas d'utilisation « Consulter les détails d'un utilisateur »

<b>Cas d'utilisation</b>	Consulter les détails d'un utilisateur
<b>Acteur</b>	Admin
<b>Pré-conditions</b>	Le système est en marche. L'admin est authentifié.
<b>Post-conditions</b>	Les informations détaillées de l'utilisateur sont affichées.
<b>Scénario de base</b>	<ol style="list-style-type: none"> <li>L'admin clique sur le bouton pour afficher les détails d'un utilisateur depuis la liste.</li> <li>Le système affiche les informations détaillées de l'utilisateur (nom, email, rôle, date de création, etc.).</li> </ol>
<b>Exceptions</b>	Échec de consultation (problème avec l'API GET, ou erreur de base de données).

Le tableau 3.7 illustre la description détaillée du cas d'utilisation « Modifier un utilisateur ».

Tableau 3.7: Description textuelle du cas d'utilisation « Modifier un utilisateur »

<b>Cas d'utilisation</b>	Modifier un utilisateur
<b>Acteur</b>	Admin
<b>Pré-conditions</b>	Le système est en marche. L'admin est authentifié.
<b>Post-conditions</b>	L'utilisateur est modifié.
<b>Scénario de base</b>	<ol style="list-style-type: none"> <li>Le système affiche l'interface de modification.</li> <li>L'admin saisit les nouvelles données.</li> <li>L'admin clique sur le bouton « Modifier ».</li> <li>Le système enregistre les modifications.</li> <li>Le système affiche un message de succès.</li> </ol>
<b>Exceptions</b>	Échec de modification (problème avec l'API PUT, ou erreur de base de données).

Le tableau 3.8 illustre la description détaillée du cas d'utilisation « Supprimer un utilisateur ».

Tableau 3.8: Description textuelle du cas d'utilisation « Supprimer un utilisateur »

<b>Cas d'utilisation</b>	Supprimer un utilisateur
<b>Acteur</b>	Admin
<b>Pré-conditions</b>	Le système est en marche. L'admin est authentifié.
<b>Post-conditions</b>	L'utilisateur est supprimé.
<b>Scénario de base</b>	<ol style="list-style-type: none"> <li>1. L'admin clique sur le bouton « Supprimer » d'un utilisateur.</li> <li>2. Le système supprime les données correspondantes.</li> <li>3. Le système affiche un message de succès.</li> </ol>
<b>Exceptions</b>	Échec de suppression (problème avec l'API DELETE, ou erreur de base de données).

### 3.3.5 Raffinement du cas d'utilisation «Mise en place et configuration de Camunda»

Le tableau 3.9 illustre la description détaillée du cas d'utilisation « Mise en place et configuration de Camunda ».

Tableau 3.9: Description textuelle du cas d'utilisation «Mise en place et configuration de Camunda»

<b>Cas d'utilisation</b>	Mise en place et configuration de Camunda
<b>Acteur</b>	Développeur
<b>Pré-conditions</b>	Le projet backend (Spring Boot) est accessible. L'environnement de développement est configuré (Java, Maven, IDE).
<b>Post-conditions</b>	Le moteur Camunda est intégré, configuré et fonctionnel dans l'application Spring Boot.
<b>Scénario de base</b>	<ol style="list-style-type: none"> <li>1. Le développeur identifie la version de Camunda compatible avec la version de Spring Boot utilisée dans le projet.</li> <li>2. Le développeur ajoute les dépendances Camunda nécessaires dans le fichier <code>pom.xml</code>.</li> <li>3. Le développeur configure les propriétés Camunda dans le fichier <code>application.properties</code>.</li> <li>4. Le développeur initialise et configure le moteur de workflow dans la classe de configuration Spring.</li> <li>5. Le développeur crée un processus BPMN simple pour tester l'intégration.</li> <li>6. Le système valide le bon fonctionnement du moteur.</li> </ol>
<b>Exceptions</b>	Erreurs d'intégration : dépendances manquantes, mauvaise configuration, problèmes de base de données ou de compatibilité.

## 3.4 Conception

La conception est une phase importante pour la bonne réalisation d'un projet. Dans cette partie nous allons exposer la conception des cas d'utilisations de ce sprint qui se traduit par un diagramme de classe globale de ce sprint suivi par les diagrammes de classes et les diagrammes de séquences de chaque cas d'utilisation.

### 3.4.1 Diagramme de classe du sprint 1

La figure 3.4 illustre le diagramme de classe du Sprint 1. Il est formé de :

- **Classe "Permission"** : C'est une énumération qui définit les permissions des différents rôles (ADMIN, MANAGER, etc.).
- **Classe "Role"** : C'est une énumération qui permet d'identifier les rôles des utilisateurs de la plateforme.
- **Classe "Token"** : C'est la classe qui gère les jetons d'authentification des utilisateurs.
- **Classe "User"** : C'est la classe qui représente tous les utilisateurs ayant accès à la plate-forme.
- **Interface "Implements Spring Security UserDetailsService"** : Indique que la classe User implémente cette interface pour gérer l'authentification et l'autorisation.

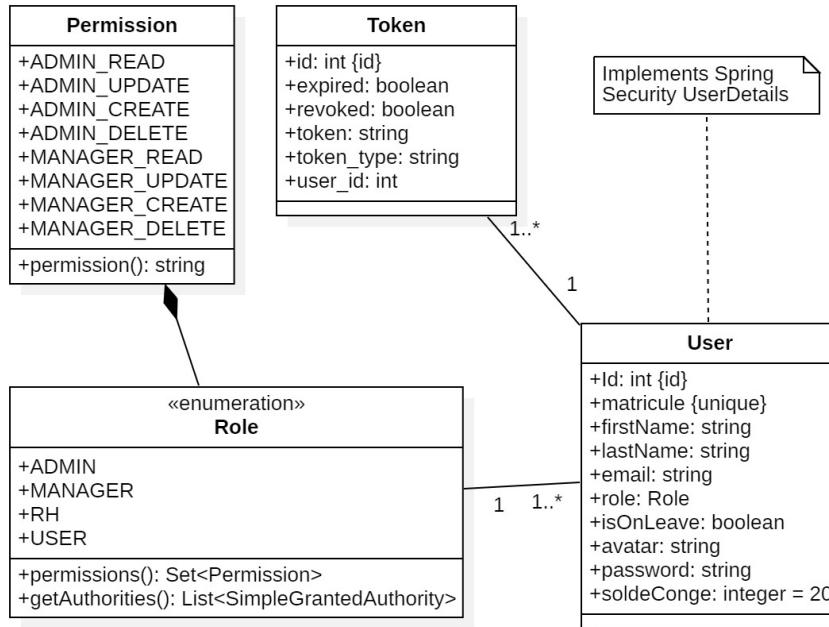


Figure 3.4: Diagramme de classe du sprint 1

### 3.4.2 Conception du cas d'utilisation «S'authentifier»

#### Diagramme de classe

La figure 3.5 illustre le diagramme de classes du cas d'utilisation s'authentifier.

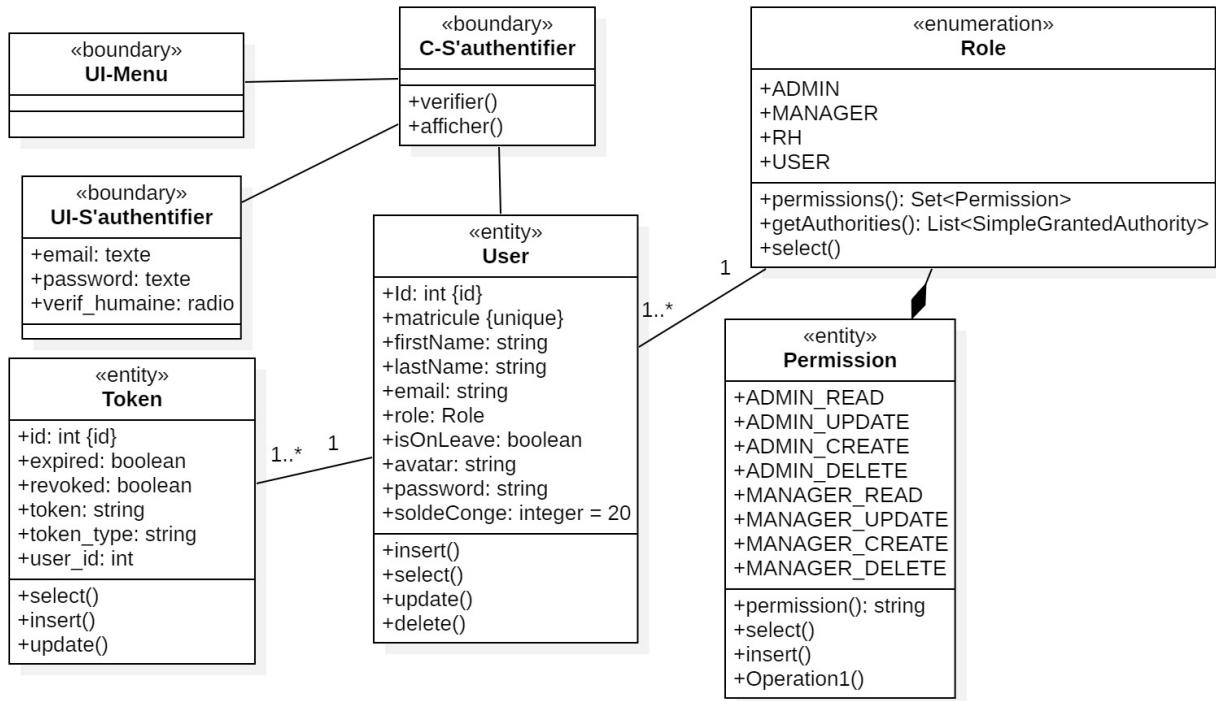


Figure 3.5: Diagramme de classe du cas d'utilisation «S'authentifier»

#### Diagramme de séquence

Au lancement de l'application, une interface d'authentification apparaîtra. Ensuite, les utilisateurs peuvent entrer leurs adresses mail et mots de passe, et vérifier qu'ils ne sont pas un robot (recaptcha), qui seront envoyés via l'interface d'authentification au contrôleur d'authentification, qui à son tour vérifie les informations dans l'entité *Utilisateur*.

Si les coordonnées sont correctes, ils seront amenés à un espace approprié selon leurs rôles. Sinon, un message d'erreur d'authentification sera affiché.

La figure 3.6 modélise le diagramme de séquence du cas d'utilisation « S'authentifier ».

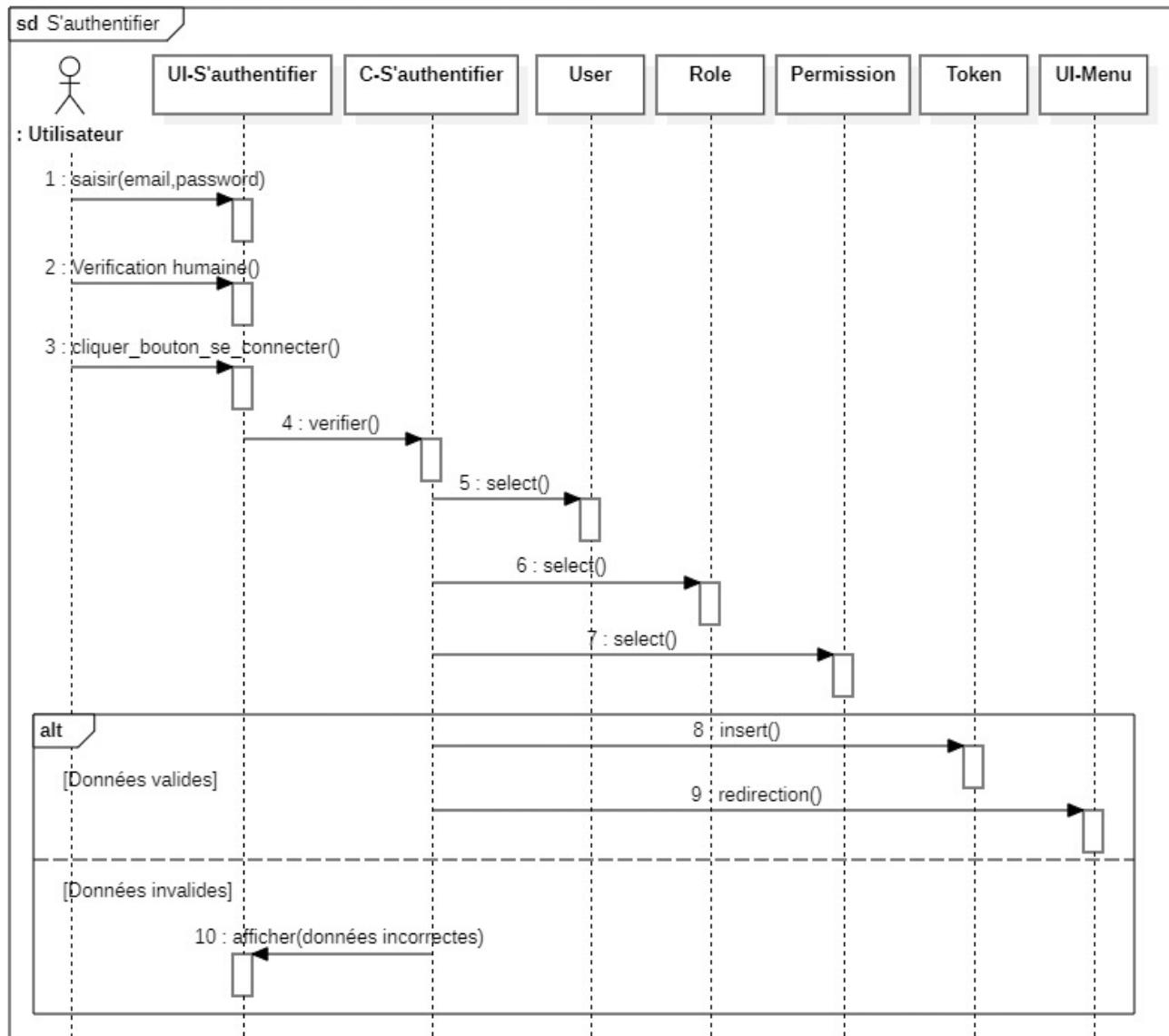


Figure 3.6: Diagramme de séquence du cas d'utilisation «S'authentifier»

### 3.4.3 Conception du cas d'utilisation «Se déconnecter»

#### Diagramme de classe

La figure 3.7 illustre le diagramme de classe du cas d'utilisation se déconnecter.

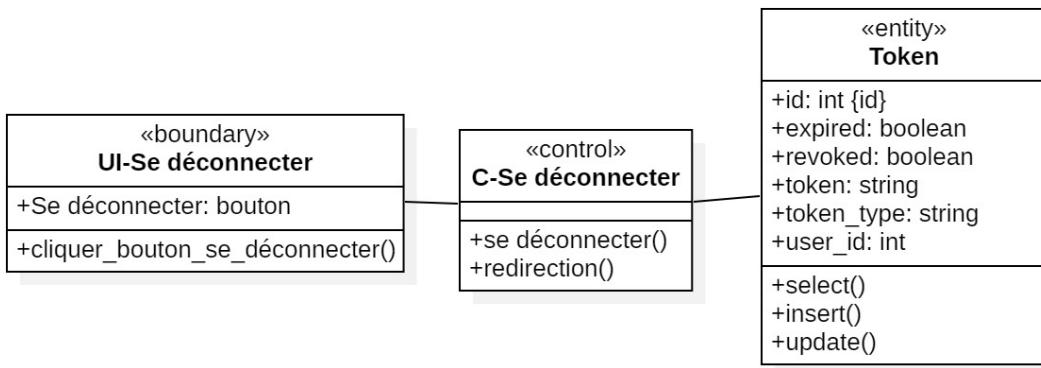


Figure 3.7: Diagramme de classe du cas d'utilisation «Se déconnecter»

### Diagramme de séquence

Si l'utilisateur souhaite se déconnecter, il clique sur le bouton de déconnexion dans le menu. Le système supprime les données obtenues avec l'authentification de l'utilisateur et retire son token. Ce dernier sera finalement redirigé vers l'interface d'authentification.

La figure 3.8 illustre le diagramme de séquence du cas d'utilisation se déconnecter.

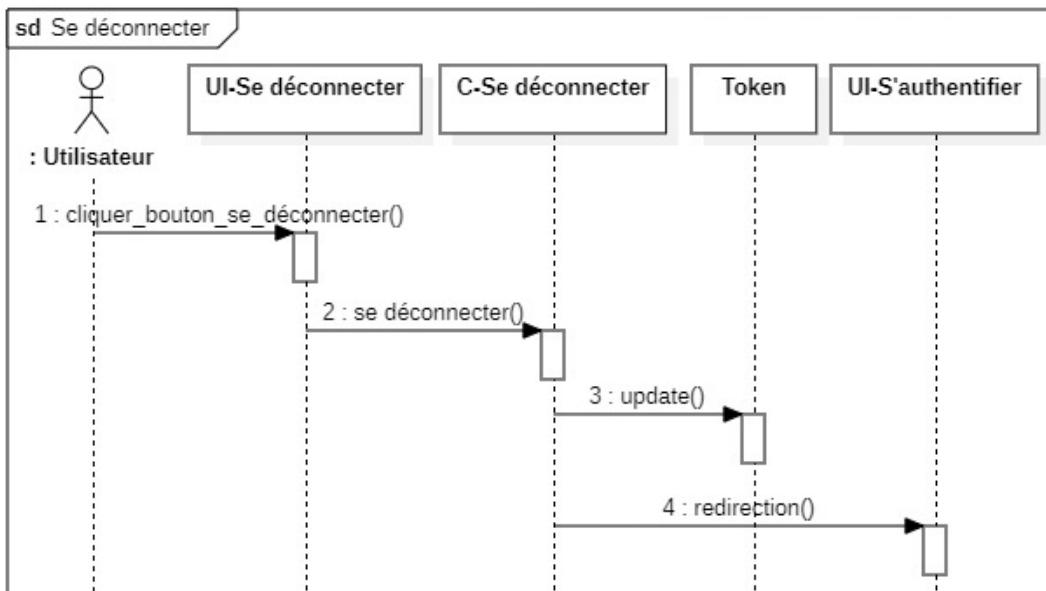


Figure 3.8: Diagramme de séquence du cas d'utilisation «Se déconnecter»

### 3.4.4 Conception du cas d'utilisation «Gérer les utilisateurs»

#### Diagramme de classe

La figure 3.9 illustre le diagramme de classe du cas d'utilisation gérer les utilisateurs.

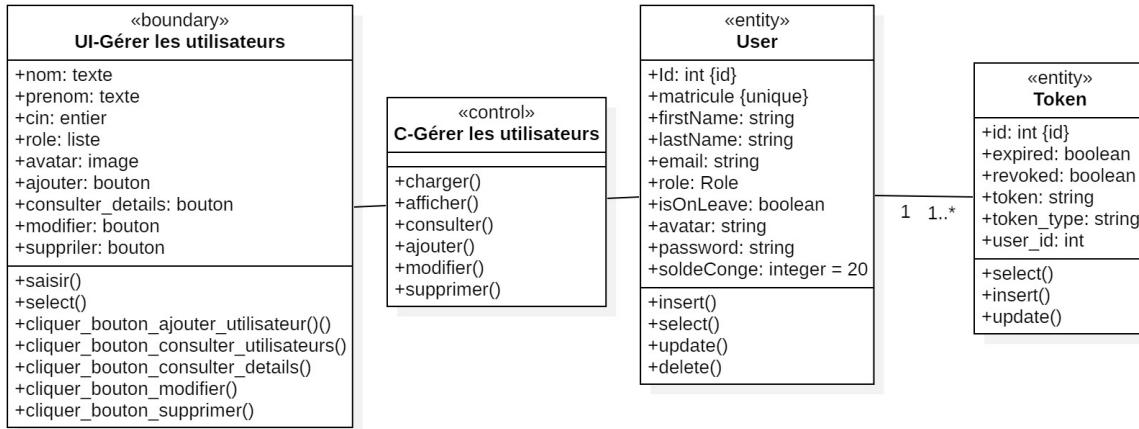


Figure 3.9: Diagramme de classe du cas d'utilisation «Gérer les utilisateurs»

#### Diagramme de séquence

Seul l'administrateur dispose des droits nécessaires pour gérer les utilisateurs depuis son interface dédiée. Cette gestion inclut un ensemble d'opérations essentielles telles que l'ajout de nouveaux utilisateurs, la consultation de la liste des utilisateurs existants, l'affichage des détails d'un utilisateur spécifique, la modification de ses informations, ainsi que la suppression d'un compte utilisateur. Toutes ces actions sont déclenchées depuis l'interface d'administration, puis transmises au contrôleur via des requêtes HTTP. Ce dernier assure la logique métier en interagissant avec les entités du système, notamment l'entité "User" pour manipuler les données des utilisateurs, et l'entité "Token" pour révoquer les tokens en cas de suppression d'un utilisateur.

La figure 3.10 illustre le diagramme de séquence du cas d'utilisation « Gérer les utilisateurs ».

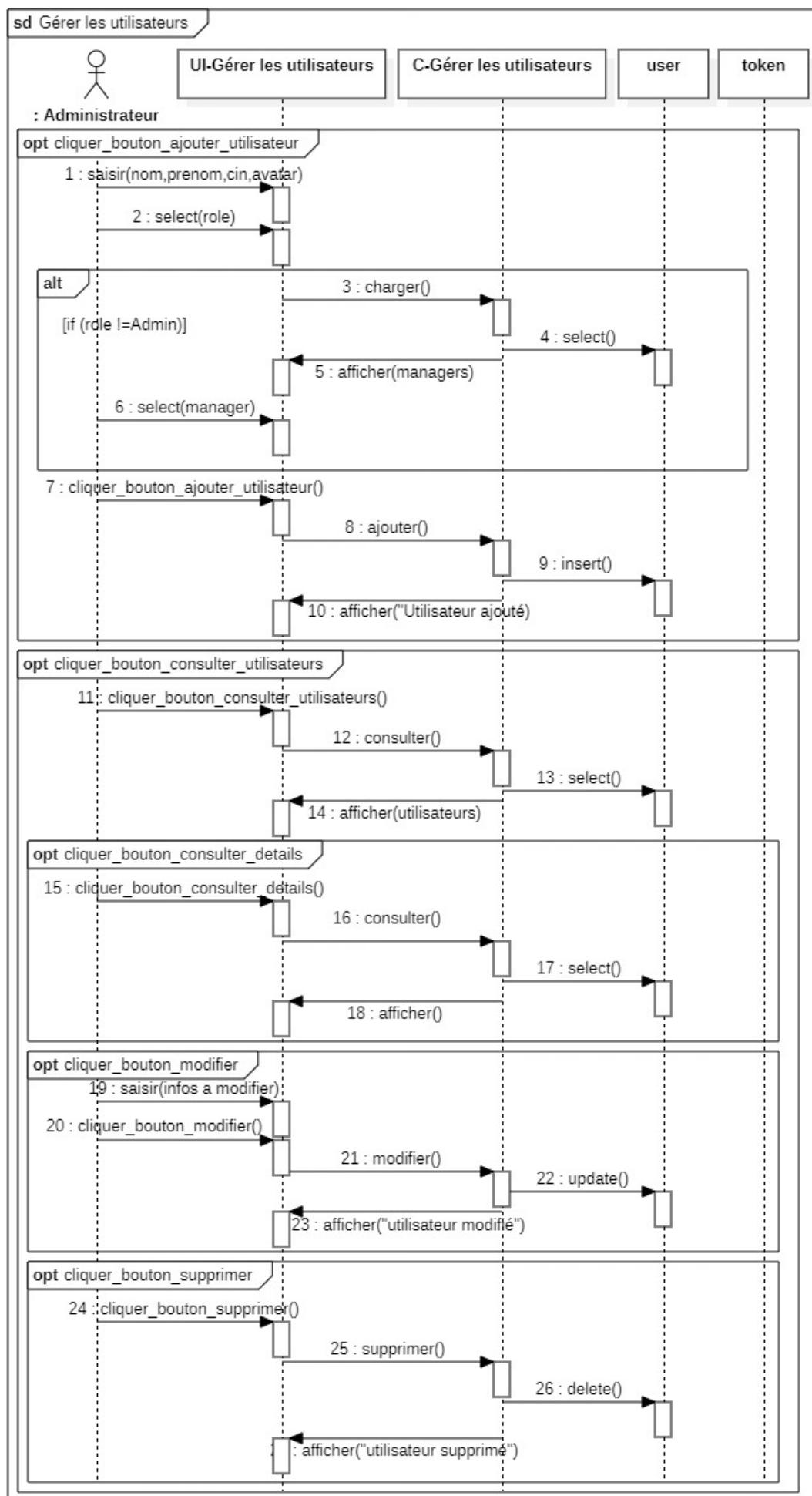


Figure 3.10: Diagramme de séquence du cas d'utilisation «Gérer les utilisateurs»<sup>38</sup>

## 3.5 Réalisation

Dans cette partie, nous présentons les modules de notre premier sprint en utilisant des captures d'écran.

### 3.5.1 S'authentifier

L'utilisateur n'est pas autorisé à accéder à l'application, sauf s'il a un compte. Donc, afin d'explorer les fonctionnalités de l'application, l'utilisateur est invité à s'authentifier avec un email et un mot de passe comme le montre la figure 3.11 ci-dessous. L'utilisateur ne peut pas créer un compte puisque les comptes sont attribués par un administrateur.

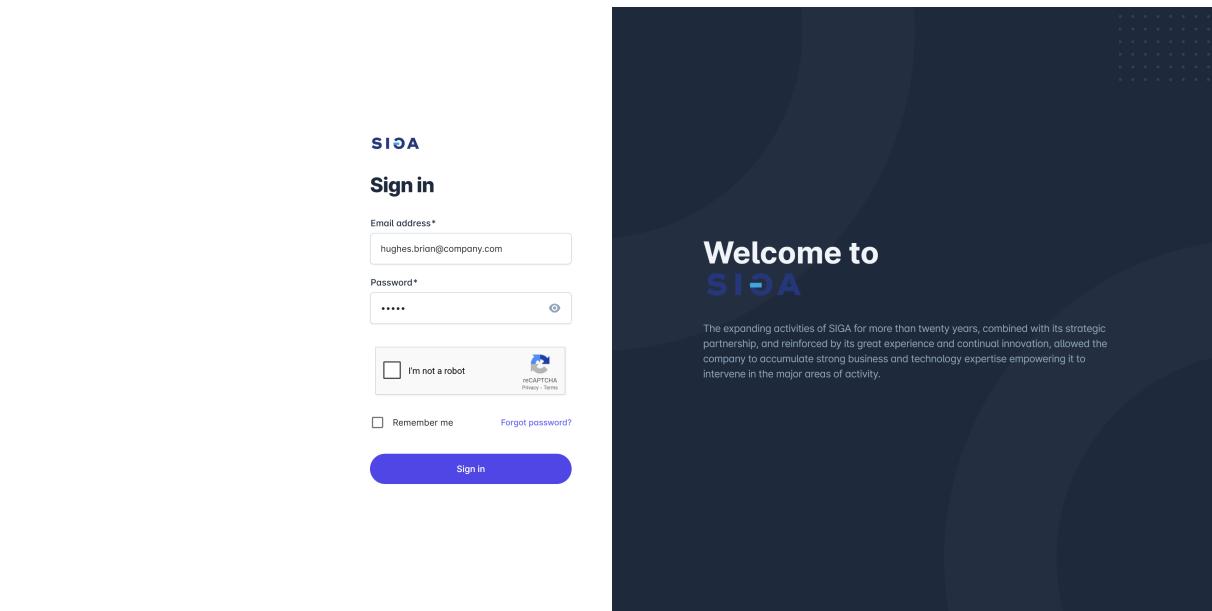


Figure 3.11: Interface du cas d'utilisation «S'authentifier»

### 3.5.2 Se déconnecter

L'utilisateur a la possibilité de se déconnecter de l'application lorsqu'il le souhaite, comme le montre la figure 3.12. Cette action met fin à sa session en cours et le redirige vers la page d'authentification.

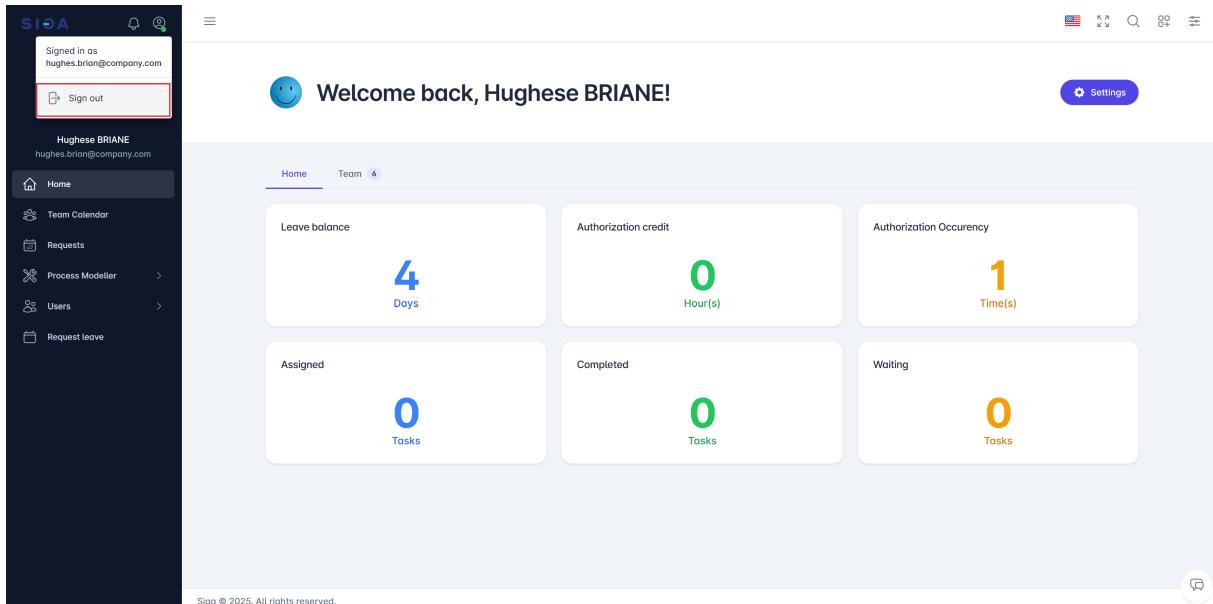


Figure 3.12: Interface du cas d'utilisation «Se déconnecter»

### 3.5.3 Gérer les utilisateurs

La gestion des utilisateurs permet à l'administrateur de consulter, ajouter, modifier, supprimer et consulter les détails des utilisateurs.

#### Ajouter un utilisateur

L'administrateur peut ajouter un nouvel utilisateur à l'application via l'interface dédiée accessible depuis le menu latéral, comme illustré dans la figure 3.13. Il suffit de renseigner les informations demandées telles que le prénom, le nom, l'adresse e-mail, le mot de passe, le rôle ainsi qu'un avatar, puis de cliquer sur Create account pour finaliser l'ajout.

Figure 3.13: Interface du cas d'utilisation «Ajouter un utilisateur»

## Consulter les utilisateurs

L'administrateur peut consulter la liste des utilisateurs existants via l'onglet Users > List dans le menu latéral, comme le montre la figure 3.14. Cette interface permet d'avoir un aperçu global des comptes enregistrés, incluant leurs informations de profil et leur rôle dans le système.

Matricule	Firstname	Lastname	Email	Role	Leave balance	Manager Matricule	Details
2501EMP001	mourad	BenFlen	self.abidi@esprit.tn	USER	20	2501EMP002	
2501EMP002	iheb	Youfli	iheb@esprit.tn	MANAGER	23		
2501EMP003	Seifeddine	Abidi	seifeddine.abidi@esprit.tn	RH	12	2501EMP002	
2501EMP004	Ahmed	jaballah	imed.jaballah@esprit.tn	ADMIN	5	2501EMP002	
2501EMP005	Hughese	BRIANE	hughes.briane@company.com	ADMIN	4	2501EMP002	
2501EMP006	string	string	string	MANAGER	16	2501EMP002	
2502EMP001	imed	jaballah	self.briane@company.com	MANAGER	21	2501EMP002	

Figure 3.14: Interface du cas d'utilisation «Consulter les utilisateurs»

## Consulter les détails d'un utilisateur

L'administrateur peut consulter les détails d'un compte spécifique en sélectionnant un utilisateur dans la liste, comme illustré dans la figure 3.15. Cette action affiche les informations complètes du profil sélectionné.

Figure 3.15: Interface du cas d'utilisation «Consulter les détails d'un utilisateur»

## Modifier un utilisateur

L'administrateur peut modifier les informations d'un compte existant en accédant à son profil depuis la liste des utilisateurs, comme le montre la figure 3.16. Une fois les champs mis à jour (nom, e-mail, rôle, etc.), il peut enregistrer les modifications pour qu'elles soient prises en compte immédiatement.

The screenshot shows the SIDA application's 'Users list' page. On the left is a sidebar with navigation links: Home, Team Calendar, Requests, Process Modeler, Users (selected), List, Add, and Request leave. The main area displays a table with columns: Matricule, Firstname, Lastname, Email, Role, Leave balance, Manager Matricule, and Details. One row is selected, showing details for a user named 'moured' with Matricule '2501EMP001'. The 'Details' section contains input fields for Firstname ('moured'), Lastname ('BenFlen'), Matricule ('2501EMP001'), Role ('USER'), Solde Conge ('20'), Manager Matricule ('2501EMP002'), and Email ('seif.abidi@esprit.tn'). At the bottom right of the table, there is a blue 'Update' button, which is highlighted with a red box.

Figure 3.16: Interface du cas d'utilisation «Modifier un utilisateur»

## Supprimer un utilisateur

L'administrateur peut supprimer un compte utilisateur en accédant à la liste des utilisateurs et en sélectionnant l'option de suppression associée, comme illustré dans la figure 3.17. Une confirmation est demandée avant la suppression définitive afin d'éviter toute action involontaire.

This screenshot is identical to Figure 3.16, showing the 'Users list' page in the SIDA application. The user 'moured' is selected for modification. The 'Delete' button at the bottom left of the table is highlighted with a red box.

Figure 3.17: Interface du cas d'utilisation «Supprimer un utilisateur»

## Intégrer Camunda

Afin d'intégrer le moteur de workflow Camunda au projet, l'utilisateur doit ajouter les dépendances nécessaires dans le fichier pom.xml, comme illustré dans la figure 3.18. Ces dépendances permettent d'activer le moteur BPM, l'interface web de gestion, ainsi que l'accès aux API REST de Camunda.

```
<dependency>
    <groupId>org.camunda.bpm.springboot</groupId>
    <artifactId>camunda-bpm-spring-boot-starter</artifactId>
    <version>7.20.0</version>
</dependency>
<dependency>
    <groupId>org.camunda.bpm.springboot</groupId>
    <artifactId>camunda-bpm-spring-boot-starter-webapp</artifactId>
    <version>7.20.0</version>
</dependency>
<dependency>
    <groupId>org.camunda.bpm.springboot</groupId>
    <artifactId>camunda-bpm-spring-boot-starter-rest</artifactId>
    <version>7.20.0</version>
</dependency>
```

Figure 3.18: Interface d'intégration de Camunda

## Configurer Camunda

Après avoir ajouté les dépendances nécessaires, l'utilisateur doit configurer Camunda dans le fichier application.properties, comme montré dans la figure 3.19. Cette configuration permet de personnaliser des paramètres essentiels tels que l'authentification, l'accès à l'interface web ou encore le comportement du moteur de workflow au démarrage.

```
camunda.bpm.admin-user.id= demo
camunda.bpm.admin-user.password= demo
camunda.bpm.authorization.enabled=true
camunda.bpm.history-level=full
camunda.bpm.generate-unique-process-instance-id=true
camunda.bpm.job-execution.enabled=true
camunda.bpm.database.schema-update=true
camunda.bpm.process-engine-name=default
camunda.bpm.enabled=true
camunda.bpm.webapp.enabled=true
camunda.bpm.webapp.cors.enabled=true
camunda.bpm.webapp.cors.allowed-origins=http://localhost:4200
camunda.bpm.webapp.ui-authorization=false
```

Figure 3.19: Interface de configuration de Camunda

## Interface de Camunda

Une fois l'application démarrée, l'utilisateur ayant les droits appropriés peut accéder à l'interface Cockpit de Camunda, comme illustré dans la figure 3.20. Cette interface web permet de surveiller les processus en cours d'exécution, d'examiner les incidents, et d'analyser les performances du moteur BPMN en temps réel.

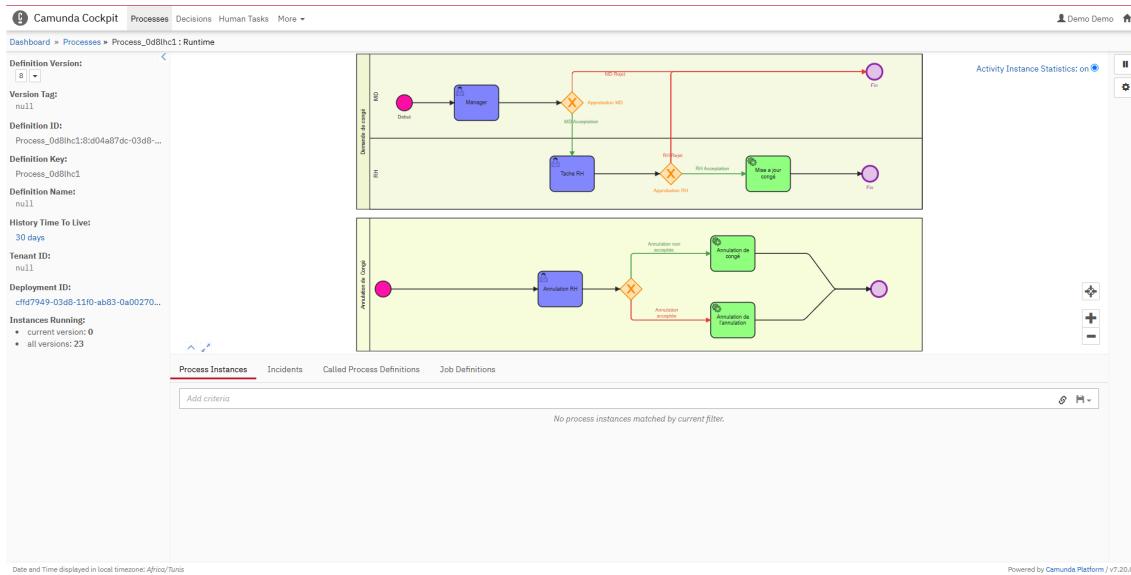


Figure 3.20: Interface cockpit de Camunda

## 3.6 Conclusion

Le premier sprint du projet a permis de poser des bases solides pour la gestion des accès et l'administration de la plateforme. En implémentant les fonctionnalités clés d'authentification, de déconnexion, de gestion des utilisateurs et d'intégration de Camunda, nous avons établi une infrastructure robuste et sécurisée, essentielle pour les sprints futurs. Les cas d'utilisation ont été soigneusement raffinés, conçus et réalisés, avec des interfaces utilisateur intuitives et des mécanismes techniques fiables, tels que l'utilisation des tokens JWT pour l'authentification, renforcés par les adresses IP des clients et les informations du client-agent (navigateur) pour ajouter une couche supplémentaire de sécurité. L'intégration du moteur de workflow Camunda a également permis de structurer et gérer les processus de manière fluide et évolutive.

Cette étape a non seulement répondu aux besoins fonctionnels prioritaires, mais a également renforcé la qualité, la sécurité et la fiabilité du système grâce à des pratiques de développement rigoureuses. Les fondations établies dans ce sprint garantiront une évolutivité et une maintenabilité optimales pour les prochaines itérations du projet.

# Chapitre 4

## Sprint 2 : Gestion des demandes

### 4.1 Introduction

Après avoir établi les fondations d'accès et d'administration dans le Sprint 1, le Sprint 2 se concentre sur la gestion des demandes au sein de la plateforme. Ce chapitre détaille les fonctionnalités prévues pour permettre aux utilisateurs, managers et RH de gérer efficacement les demandes et congés, tout en offrant des outils pour la consultation et le traitement de ces demandes. Nous aborderons les besoins, la conception et les étapes de réalisation de ce sprint.

### 4.2 Backlog du Sprint 2

Tableau 4.1: Backlog du Sprint 2

Cas d'utilisation	Priorité	Tâche
Gérer son profil	2	Modifier ses informations (nom, prénom, e-mail, etc.)
Réinitialiser son mot de passe	2	Utilisateur réinitialise via un lien sécurisé
Soumettre une demande	2	Utilisateurs, Managers, RH soumettent des demandes
Consulter ses demandes	2	Utilisateurs, Managers, RH consultent l'historique
Consulter ses congés	2	Utilisateurs, Managers, RH consultent congés, solde
Traiter une demande	2	Managers, RH valident ou rejettent les demandes

## 4.3 Rafinement de cas d'utilisation

Cette partie consiste à analyser et spécifier les besoins de ce deuxième sprint à travers l'identification des acteurs et le raffinement des cas d'utilisations.

### 4.3.1 Identification des acteurs du deuxième sprint

Les acteurs de ce sprint sont :

**Utilisateur** : Soumet des demandes (congés, absences) et consulte ses propres demandes et congés.

**Manager** : Soumet, consulte et traite les demandes (validation/rejet).

**RH (Ressources Humaines)** : Soumet, consulte et traite les demandes, gère les congés.

### 4.3.2 Raffinement du Cas d'Utilisation «Gérer son Profil»

La figure 4.1 illustre le diagramme de cas d'utilisation « Gérer le profil ».

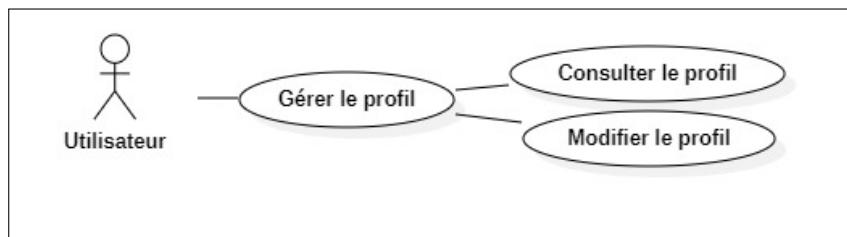


Figure 4.1: Diagramme du cas d'utilisation «Gérer le profil»

Le tableau 4.2 illustre la description textuelle du cas d'utilisation « Consulter le Profil ».

Tableau 4.2: Description textuelle du Cas d'utilisation «Consulter le Profil»

<b>Cas d'utilisation</b>	Consulter le Profil
<b>Acteur</b>	Utilisateur, Manager, RH
<b>Pré-conditions</b>	Système en marche. Utilisateur authentifié.
<b>Post-conditions</b>	Profil consulté.
<b>Scénario de Base</b>	1. Le système affiche l'interface du profil de l'utilisateur. 2. L'utilisateur consulte la liste des informations de son profil.
<b>Exceptions</b>	Échec de consultation (problème dans l'API GET, problème dans la base de données).

Le tableau 4.3 illustre la description textuelle du cas d'utilisation « Modifier le Profil ».

Tableau 4.3: Description textuelle du Cas d'utilisation «Modifier le Profil»

<b>Cas d'utilisation</b>	Modifier le Profil
<b>Acteur</b>	Utilisateur
<b>Pré-conditions</b>	Système en marche. Utilisateur authentifié.
<b>Post-conditions</b>	Profil modifié.
<b>Scénario de Base</b>	<ol style="list-style-type: none"><li>1. Le système affiche l'interface du profil de l'utilisateur.</li><li>2. L'utilisateur modifie les informations choisies.</li><li>3. L'utilisateur clique sur le bouton « Modifier ».</li><li>4. Le système modifie ces informations.</li><li>5. Le système affiche un message de succès de modification.</li></ol>
<b>Exceptions</b>	Échec de modification (problème dans l'API PUT, problème dans la base de données).

### 4.3.3 Raffinement du Cas d'Utilisation «Réinitialiser son Mot de Passe»

La figure 4.2 illustre le diagramme de cas d'utilisation « Réinitialiser son Mot de Passe ».

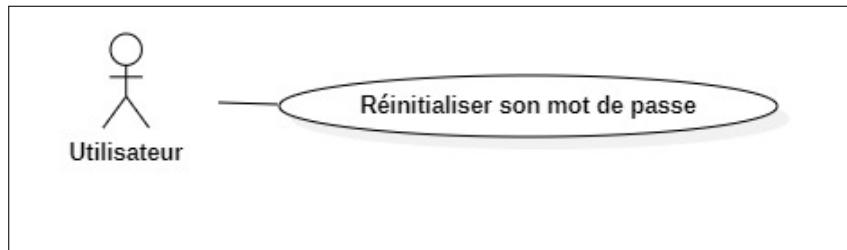


Figure 4.2: Diagramme du cas d'utilisation «Réinitialiser son Mot de Passe»

Le tableau 4.4 illustre la description textuelle du cas d'utilisation « Réinitialiser son Mot de Passe ».

Tableau 4.4: Description textuelle du Cas d'utilisation «Réinitialiser son Mot de Passe»

<b>Cas d'utilisation</b>	Réinitialiser son Mot de Passe
<b>Acteur</b>	Utilisateur
<b>Pré-conditions</b>	Système en marche. Utilisateur inscrit avec un e-mail valide.
<b>Post-conditions</b>	Mot de passe réinitialisé.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur « Mot de passe oublié » sur la page de connexion.</li> <li>2. Il entre son adresse e-mail et soumet la demande.</li> <li>3. Le système envoie un lien sécurisé à l'e-mail de l'utilisateur.</li> <li>4. L'utilisateur clique sur le lien et entre un nouveau mot de passe.</li> <li>5. Le système enregistre le nouveau mot de passe et affiche un message de succès.</li> </ol>
<b>Exceptions</b>	Échec de réinitialisation (e-mail invalide, lien expiré, problème API).

#### 4.3.4 Raffinement du Cas d'Utilisation «Soumettre une Demande»

La figure 4.3 illustre le diagramme de cas d'utilisation « Soumettre une Demande ».

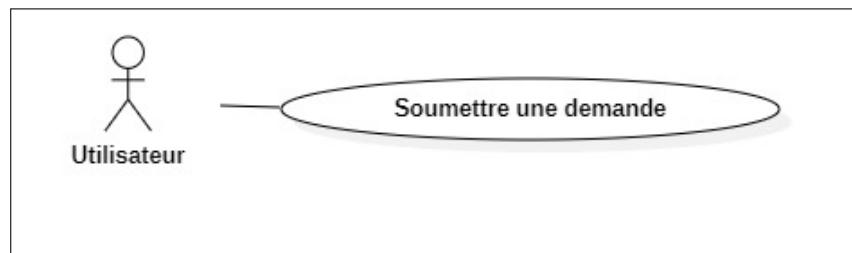


Figure 4.3: Diagramme du cas d'utilisation «Soumettre une Demande»

Le tableau 4.5 illustre la description textuelle du cas d'utilisation « Soumettre une Demande ».

Tableau 4.5: Description textuelle du Cas d'utilisation «Soumettre une Demande»

<b>Cas d'utilisation</b>	Soumettre une Demande
<b>Acteur</b>	Utilisateur, Manager, RH
<b>Pré-conditions</b>	Système en marche. Acteur authentifié.
<b>Post-conditions</b>	Demande enregistrée et parties concernées notifiées.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'acteur accède au formulaire de soumission.</li> <li>2. L'acteur sélectionne le type de demande :             <ol style="list-style-type: none"> <li>a. <i>Demande de Congé</i> : Formulaire avec les champs (dates de début et fin, type de congé, motif).</li> <li>b. <i>Demande d'Autorisation</i> : Formulaire avec les champs (date, durée).</li> </ol> </li> <li>3. Il remplit les détails selon le formulaire correspondant.</li> <li>4. Il soumet la demande.</li> <li>5. Le système enregistre la demande et notifie les parties concernées.</li> </ol>
<b>Exceptions</b>	Échec de soumission (données invalides, erreur API).

### 4.3.5 Raffinement du Cas d'Utilisation «Consulter ses Demandes»

La figure 4.4 illustre le diagramme de cas d'utilisation « Consulter ses Demandes ».

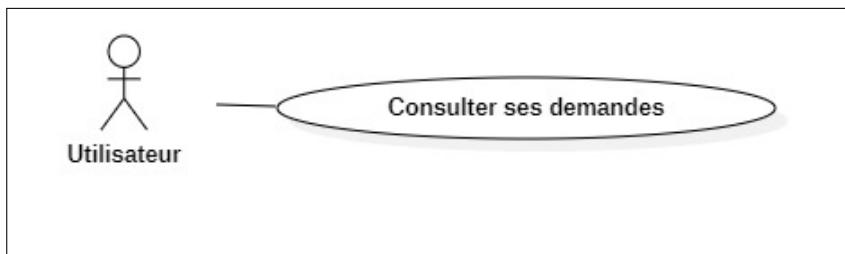


Figure 4.4: Diagramme du cas d'utilisation «Consulter ses Demandes»

Le tableau 4.6 illustre la description textuelle du cas d'utilisation « Consulter ses Demandes ».

Tableau 4.6: Description textuelle du Cas d'utilisation «Consulter ses Demandes»

<b>Cas d'utilisation</b>	Consulter ses Demandes
<b>Acteur</b>	Utilisateur, Manager, RH
<b>Pré-conditions</b>	Système en marche. Acteur authentifié.
<b>Post-conditions</b>	Historique des demandes affiché.
<b>Scénario de Base</b>	1. L'acteur accède à la section « Mes Demandes ». 2. Le système affiche la liste des demandes avec leur statut.
<b>Exceptions</b>	Échec de consultation (erreur API).

#### 4.3.6 Raffinement du Cas d'Utilisation «Consulter ses Congés»

La figure 4.5 illustre le diagramme de cas d'utilisation « Consulter ses Congés ».

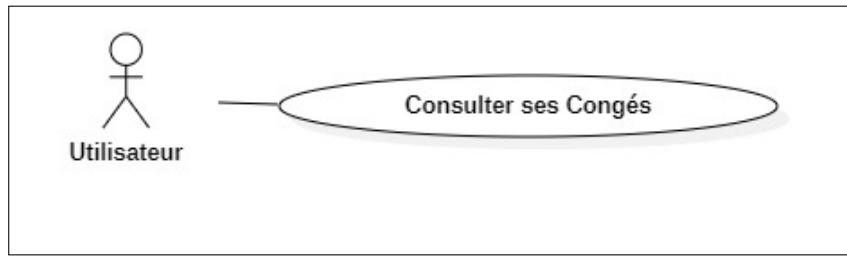


Figure 4.5: Diagramme du cas d'utilisation «Consulter ses Congés»

Le tableau 4.7 illustre la description textuelle du cas d'utilisation « Consulter ses Congés ».

Tableau 4.7: Description textuelle du Cas d'utilisation «Consulter ses Congés»

<b>Cas d'utilisation</b>	Consulter ses Congés
<b>Acteur</b>	Utilisateur, Manager, RH
<b>Pré-conditions</b>	Système en marche. Acteur authentifié.
<b>Post-conditions</b>	Solde et historique des congés affichés.
<b>Scénario de Base</b>	1. L'acteur accède à la section « Mes Congés ». 2. Le système affiche le solde et l'historique des congés.
<b>Exceptions</b>	Échec de consultation (erreur API).

### 4.3.7 Raffinement du Cas d'Utilisation «Traiter une Demande»

La figure 4.6 illustre le diagramme de cas d'utilisation « Traiter une Demande ».

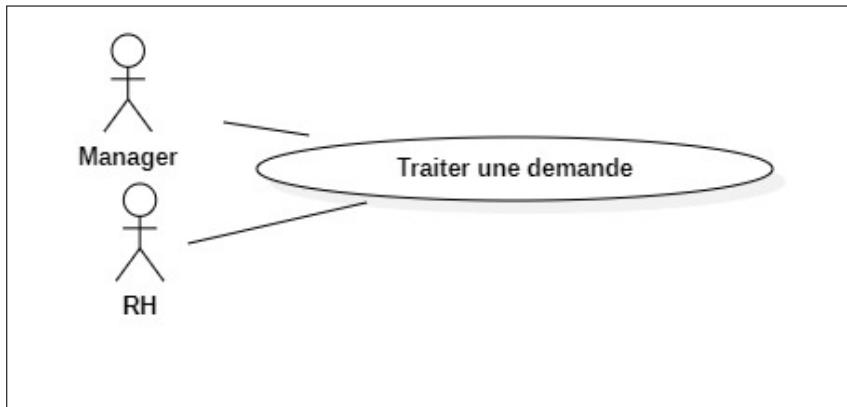


Figure 4.6: Diagramme du cas d'utilisation «Traiter une Demande»

Le tableau 4.8 illustre la description textuelle du cas d'utilisation « Traiter une Demande ».

Tableau 4.8: Description textuelle du Cas d'utilisation «Traiter une Demande»

Cas d'utilisation	Traiter une Demande
Acteur	Manager, RH
Pré-conditions	Système en marche. Acteur authentifié.
Post-conditions	Statut de la demande mis à jour et émetteur notifié.
Scénario de Base	1. L'acteur accède à la liste des demandes en attente. 2. Il sélectionne une demande. 3. Il choisit de valider ou rejeter. 4. Le système met à jour le statut et notifie l'émetteur.
Exceptions	Échec de traitement (erreur API).

## 4.4 Conception

La conception du Sprint 2 vise à modéliser les interactions entre les demandes, les congés et les notifications, en s'appuyant sur une architecture orientée objet et une intégration avec Camunda pour la gestion des processus.

### 4.4.1 Diagramme de classe du sprint 2

La figure 4.7 illustre le diagramme de classe du Sprint 2.

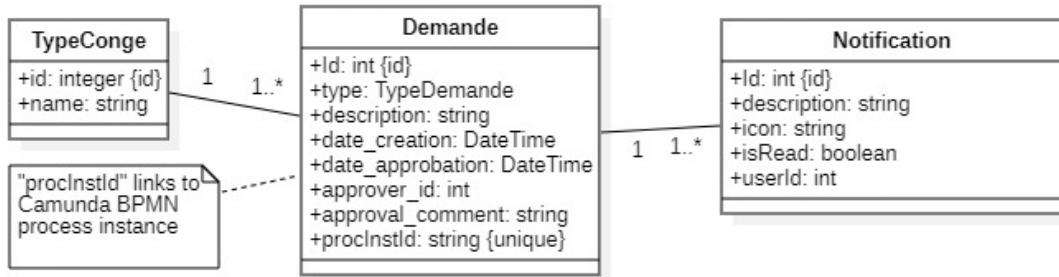


Figure 4.7: Diagramme de classe du Sprint 2

Le diagramme de classe est formé de :

- **Classe "TypeConge"** : Représente les différents types de congés disponibles sur la plate-forme.
- **Classe "Demande"** : Gère les demandes soumises par les utilisateurs, comme les congés ou autorisations.
- **Classe "Notification"** : Gère les notifications envoyées aux utilisateurs pour les informer sur leurs demandes.
- **Note sur procInstId** : L'attribut procInstId est un identifiant unique qui lie une instance de la classe (e.g., TypeConge et Demande) à une instance de processus BPMN dans Camunda. Cela permet de suivre et de gérer le workflow associé (e.g., approbation de demande) via le moteur Camunda.

#### 4.4.2 Conception du Cas d'Utilisation «Gérer son Profil»

La figure 4.8 illustre le diagramme de classe du cas d'utilisation « Gérer son Profil ».

##### Diagramme de Classe

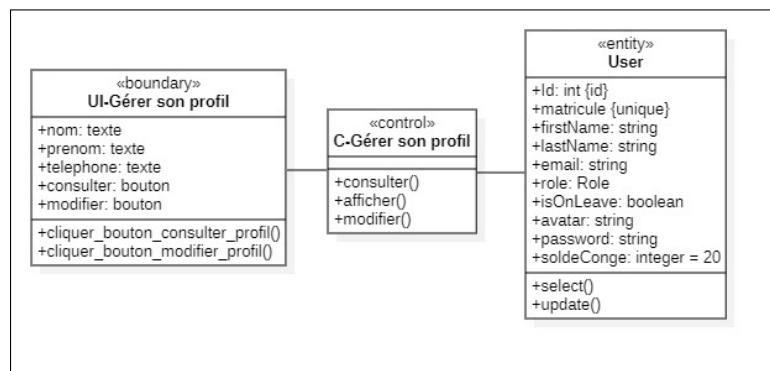


Figure 4.8: Diagramme de classe du cas d'utilisation «Gérer son Profil»

## Diagramme de Séquence

Tous les acteurs (Utilisateur, Manager, RH) ont la possibilité de gérer leurs profils à partir de leurs interfaces respectives. Lorsqu'un acteur consulte ou modifie son profil, l'interface communique avec le contrôleur de profil, qui interagit avec l'entité « User » pour récupérer ou mettre à jour les informations dans la base de données.

Ce scénario est présenté dans la figure 4.9, intitulé « diagramme de séquence du cas d'utilisation Gérer son Profil ».

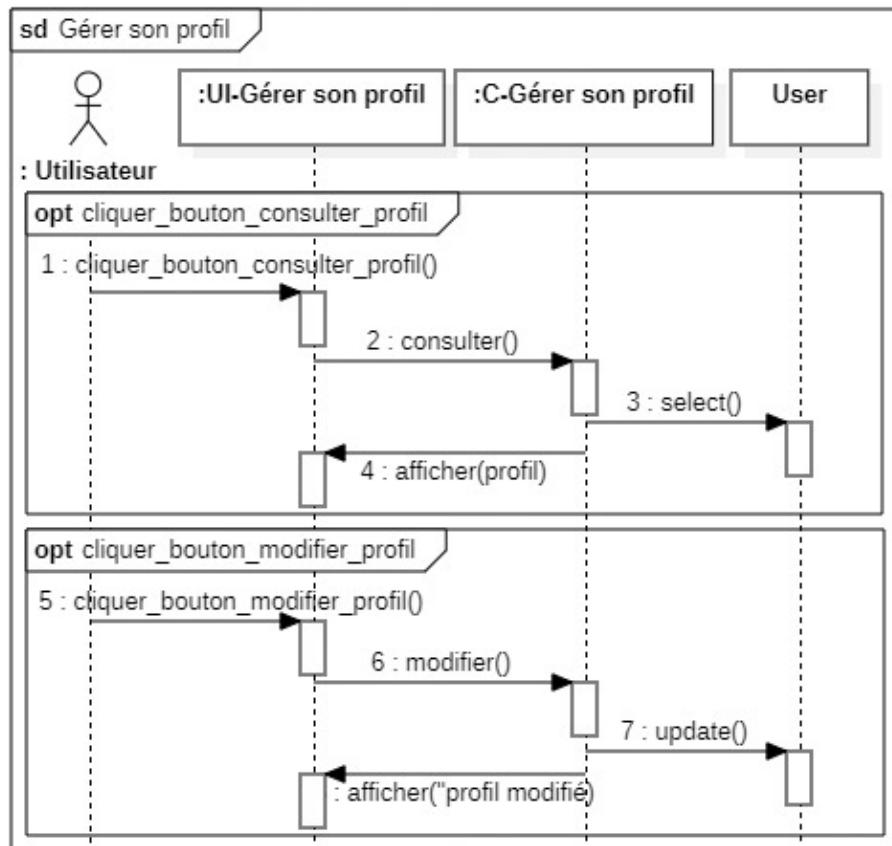


Figure 4.9: Diagramme de séquence du cas d'utilisation «Gérer son Profil»

#### 4.4.3 Conception du Cas d'utilisation «Réinitialiser son mot de passe»

##### Diagramme de Classe

La figure 4.10 illustre le diagramme de classe du cas d'utilisation « Réinitialiser son mot de passe ».

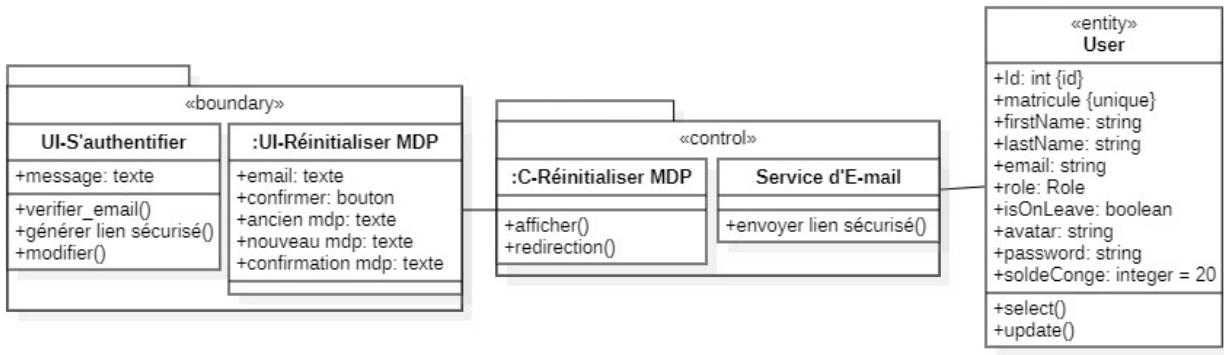


Figure 4.10: Diagramme de classe du cas d'utilisation «Réinitialiser son Mot de Passe»

##### Diagramme de Séquence

L'utilisateur qui a oublié son mot de passe initie le processus via l'interface de connexion. L'interface communique avec le contrôleur, qui vérifie l'e-mail de l'utilisateur dans l'entité « User ». Si l'e-mail n'existe pas, le système renvoie un message : « Si votre e-mail existe dans la base, vous recevrez un e-mail pour réinitialiser votre mot de passe », afin de ne pas divulguer l'existence de l'e-mail pour des raisons de sécurité. Si l'e-mail existe, le contrôleur génère un lien sécurisé et envoie une notification par e-mail via un service. Un attaquant potentiel recevant le lien sans connaître le propriétaire de l'e-mail ne peut pas modifier le mot de passe, car l'utilisateur doit resaisir son adresse e-mail pour confirmer son identité. Une fois le lien utilisé, l'utilisateur soumet un nouveau mot de passe, qui est mis à jour dans la base de données via le contrôleur. Après cette mise à jour, le lien devient expiré pour des raisons de sécurité. Ce scénario est présenté dans la figure 4.11, intitulé diagramme de séquence du cas d'utilisation « Réinitialiser son Mot de Passe ».

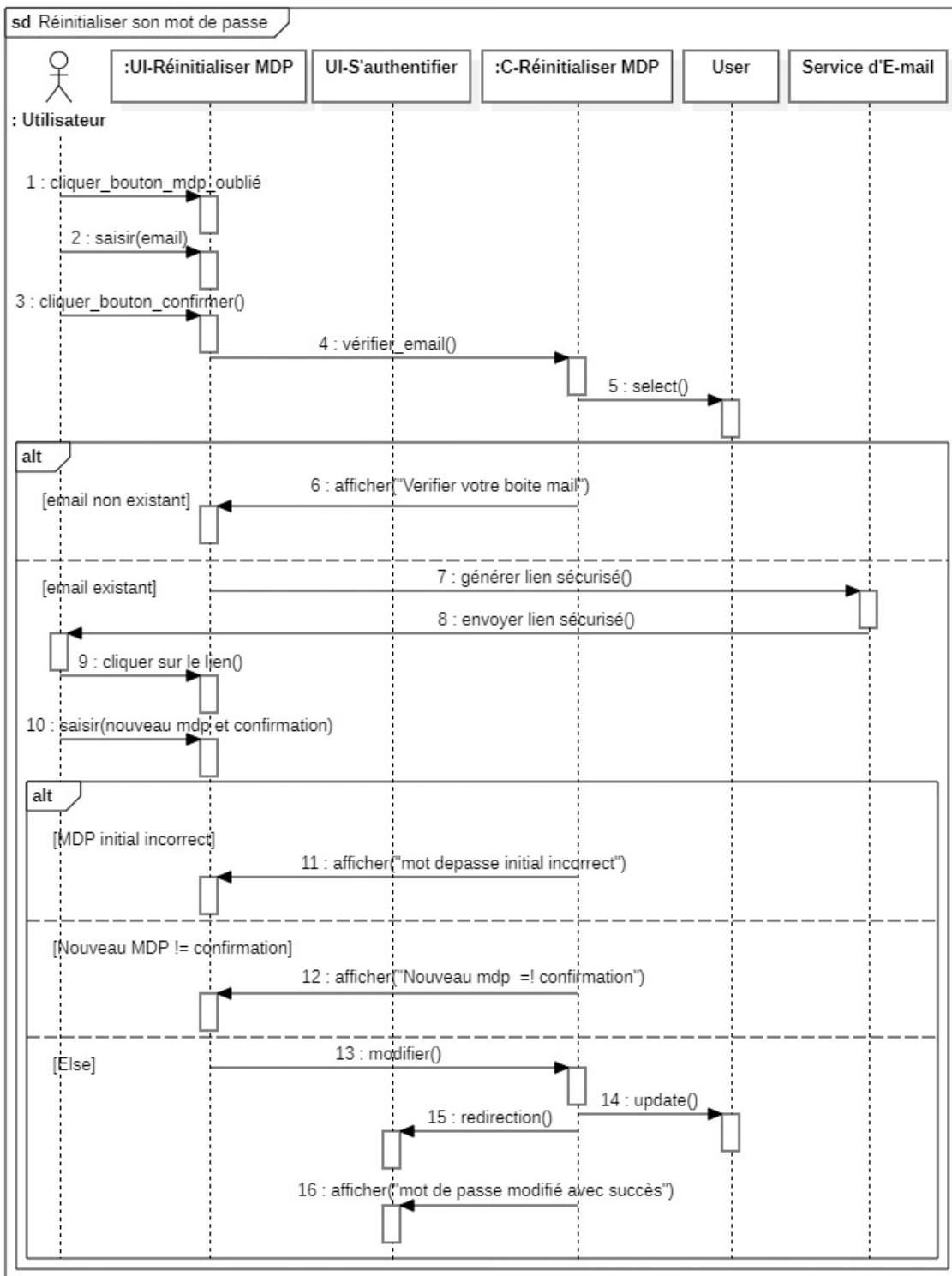


Figure 4.11: Diagramme de séquence du cas d'utilisation «Réinitialiser son Mot de Passe»

#### 4.4.4 Conception du Cas d'Utilisation «Soumettre une Demande»

##### Diagramme de Classe

La figure 4.12 illustre le diagramme de classes du cas d'utilisation « Soumettre une demande ».



Figure 4.12: Diagramme de classe du cas d'utilisation «Soumettre une demande»

##### Diagramme de Séquence

Ce scénario décrit le processus de soumission d'une demande de congé par un utilisateur via l'interface de l'application. L'utilisateur commence par remplir un formulaire de demande (étape 1). La saisie est ensuite contrôlée (étape 2) pour vérifier la validité des données. En cas d'erreur, un message est affiché (étape 3) et l'utilisateur peut corriger sa saisie, dans une boucle de validation.

Une fois les données correctement saisies (étape 4), l'utilisateur clique sur le bouton de confirmation (étape 5). Cette action est capturée par le contrôleur de soumission (**C-Soumission demande**), qui déclenche la méthode `confirmer()` (étape 6). Le contrôleur interagit ensuite avec le module de gestion des demandes (**Demande**) pour insérer la nouvelle demande dans la base de données via `insert()` (étape 7).

Parallèlement, un processus BPMN est initié dans Camunda BPM afin d'orchestrer la suite du workflow de validation (par exemple, affectation de la demande à un manager pour approbation). Enfin, un message de confirmation est retourné à l'interface pour informer l'utilisateur que sa demande a été créée avec succès (étape 8).

La figure 4.13, intitulé diagramme de séquence du cas d'utilisation « Soumettre une Demande ».

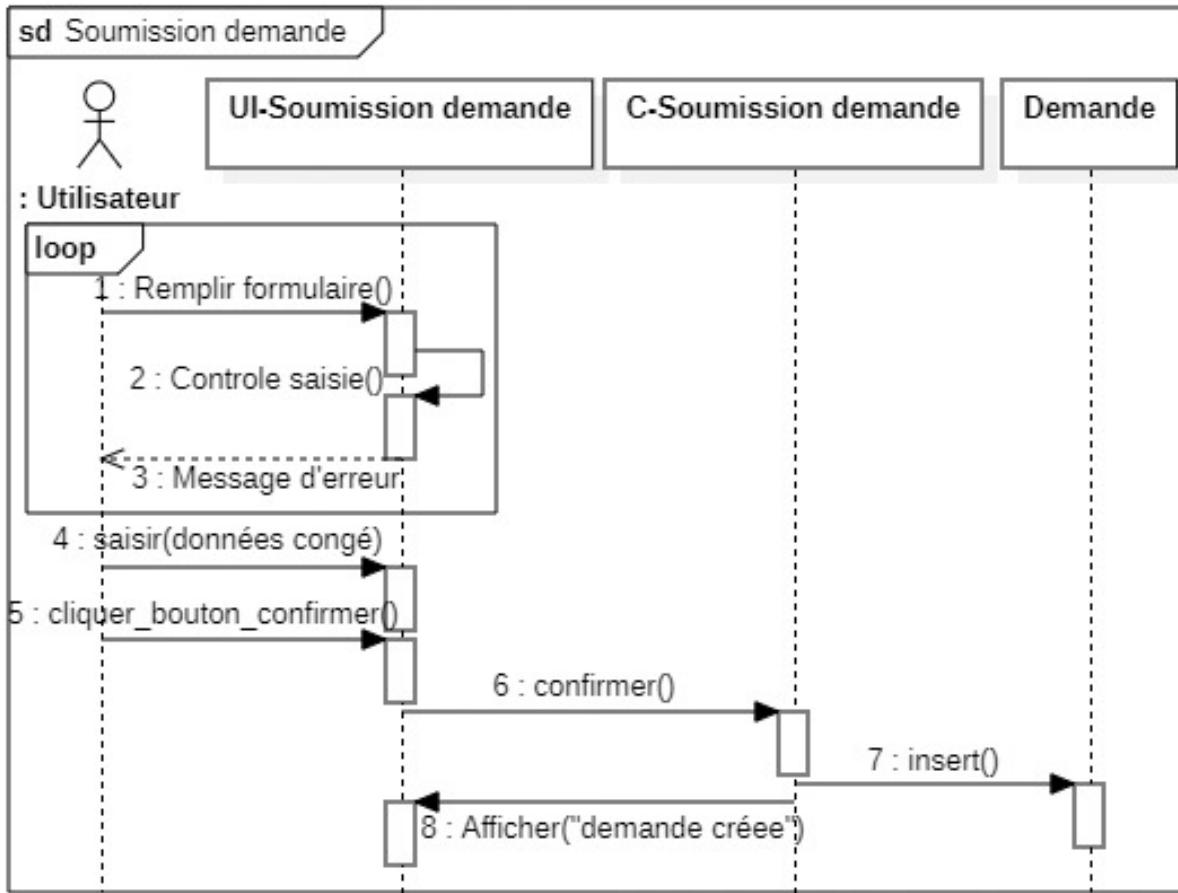


Figure 4.13: Diagramme de séquence du cas d'utilisation «Soumettre une demande»

### Diagramme Business Process Model and Notation

Le diagramme BPMN ci-dessous illustre le processus métier de gestion des demandes et annulations de congé dans la plateforme. Il est structuré en deux grands processus : Demande de congé et Annulation de congé, chacun réparti entre plusieurs rôles (ou swimlanes), notamment le Manager, le service RH, et l'initiateur de l'annulation.

**Demande de congé :** Le processus débute par la soumission d'une demande par l'utilisateur. Celle-ci est d'abord examinée par le Manager, qui peut soit approuver, soit rejeter la demande.

En cas de rejet, le processus s'arrête immédiatement.

En cas d'approbation, la demande est transmise au RH pour une seconde validation. Si le RH rejette la demande, le processus s'arrête également.

En cas de double approbation (Manager + RH), une tâche de type service est exécutée pour mettre à jour les données du congé dans le système, marquant la fin du processus.

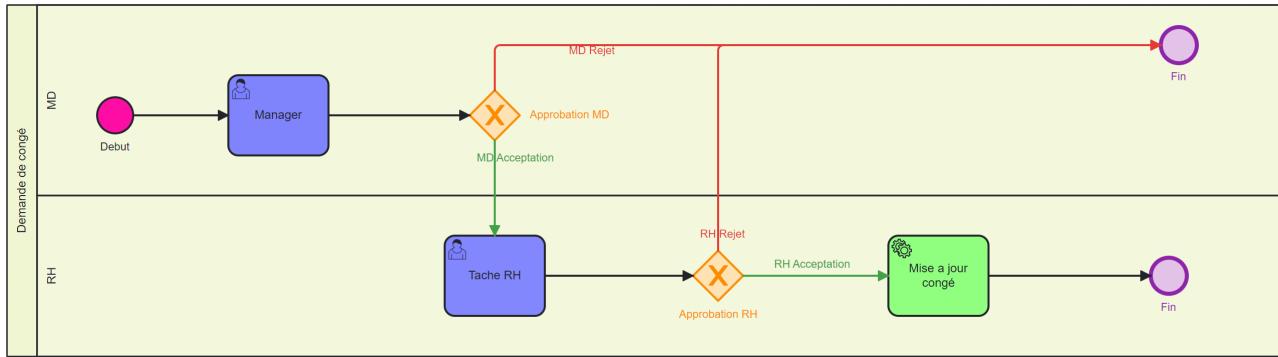


Figure 4.14: Diagramme BPMN demande de congé

**Demande d'autorisation :** Le processus débute par un événement de départ (**Début**). Ensuite, une tâche utilisateur intitulée **Demande autorisation** est initiée, dans laquelle un utilisateur soumet une requête d'autorisation.

Un **gateway exclusif** (losange avec croix) permet de rediriger le processus selon le résultat de la demande :

Si la demande est refusée (chemin rouge), le processus se termine immédiatement par un événement de fin.

Si la demande est approuvée (chemin vert), une tâche de service **Update autorisation** est exécutée. Celle-ci permet de mettre à jour les informations d'autorisation dans le système. Le processus se termine ensuite par un événement de fin classique.

Ce processus est typiquement orchestré par un moteur BPM tel que **Camunda**, permettant d'associer automatiquement des règles métiers et des traitements conditionnels à chaque étape. L'utilisation du gateway permet de gérer de façon claire et contrôlée les différentes issues de la demande.

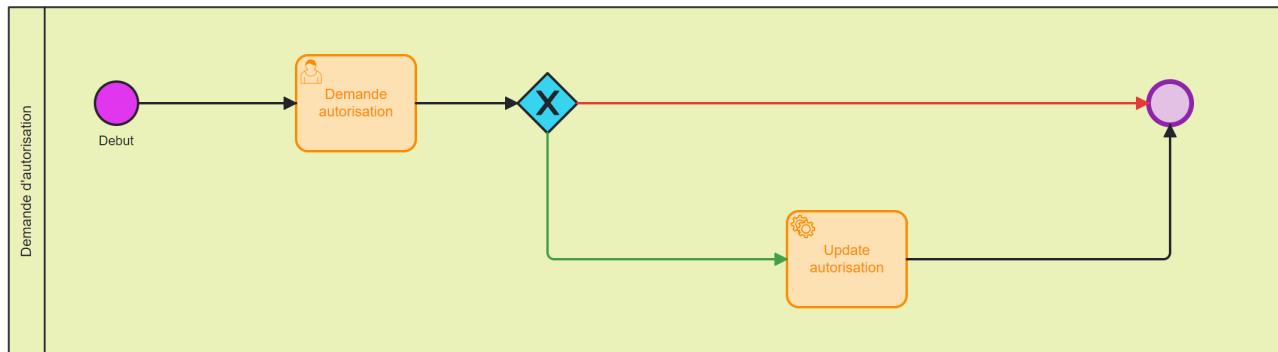


Figure 4.15: Diagramme BPMN demande d'autorisation

#### 4.4.5 Conception du Cas d'Utilisation «Consulter ses demandes»

##### Diagramme de Classe

La figure 4.16 illustre le diagramme de classes du cas d'utilisation « Consulter ses demandes ».



Figure 4.16: Diagramme de classe du cas d'utilisation «Consulter ses demandes»

#### 4.4.6 Diagramme de Séquence

Les utilisateurs, les managers et les RH ont la possibilité de consulter leurs demandes à partir de leurs interfaces respectives. Les actions qu'ils réalisent pour consulter leurs demandes sont exécutées suite à la communication entre l'interface utilisateur et le contrôleur de consultation, qui interagit avec l'entité "Demande" pour récupérer les informations correspondantes depuis la base de données. Ce scénario est présenté dans la figure 4.17.

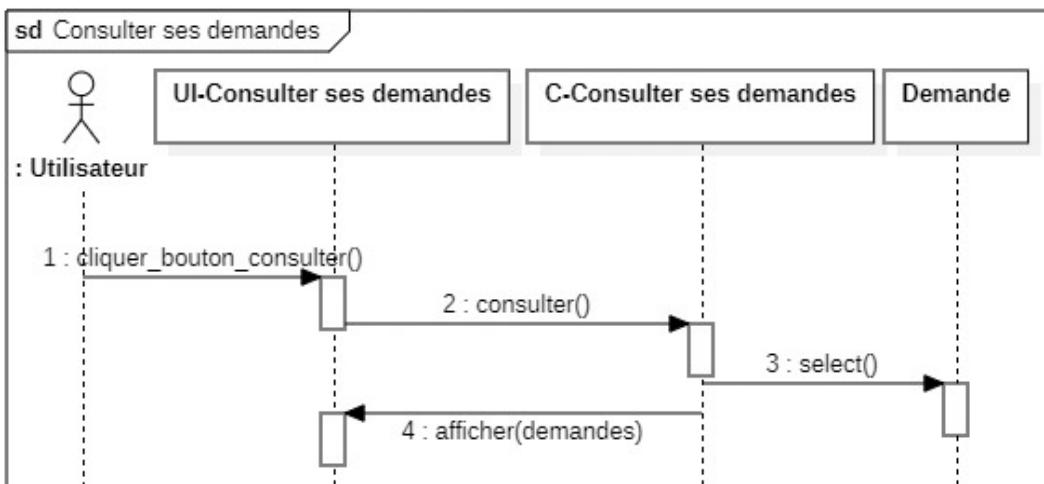


Figure 4.17: Diagramme de séquence du cas d'utilisation «Consulter ses demandes»

#### 4.4.7 Conception du Cas d'Utilisation «Consulter ses congés»

##### Diagramme de Classe

La figure 4.18 illustre le diagramme de classes du cas d'utilisation « Consulter ses congés ».



Figure 4.18: Diagramme de classe du cas d'utilisation «Consulter ses congés»

##### Diagramme de Séquence

Les utilisateurs, les managers et les RH ont la possibilité de consulter leurs congés à partir de leurs interfaces respectives. Les actions qu'ils réalisent pour consulter leurs congés sont exécutées suite à la communication entre l'interface utilisateur et le contrôleur de consultation, qui interagit avec l'entité "Demande" pour récupérer les informations correspondantes depuis la base de données(Les demandes approuvées). Ce scénario est présenté dans la figure 4.19.

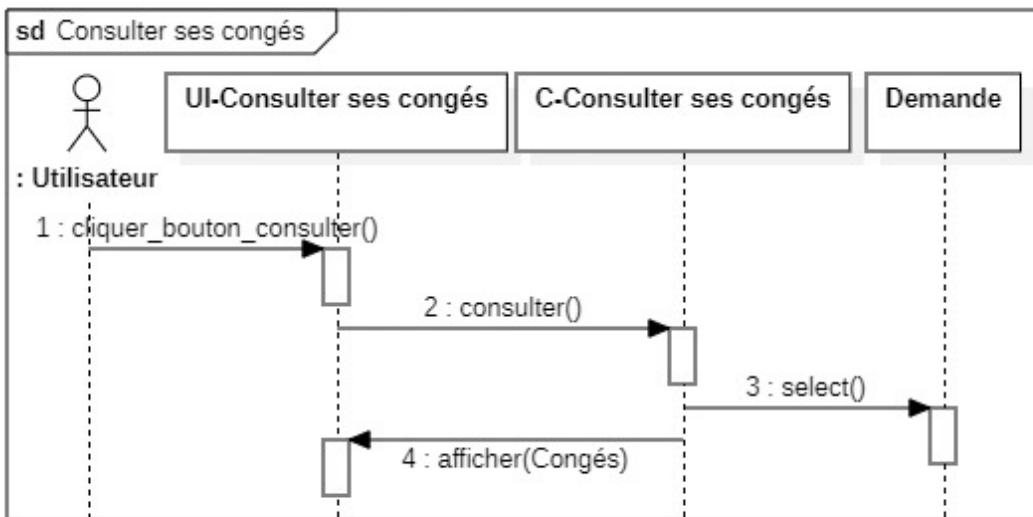


Figure 4.19: Diagramme de séquence du cas d'utilisation «Consulter ses congés»

#### 4.4.8 Conception du Cas d'Utilisation «Traiter une demande»

##### Diagramme de Classe

La figure 4.20 illustre le diagramme de classes du cas d'utilisation « Traiter une demande ».

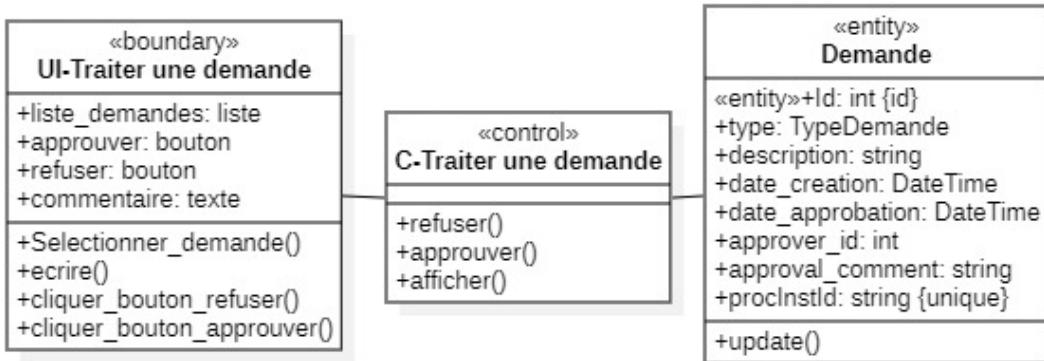
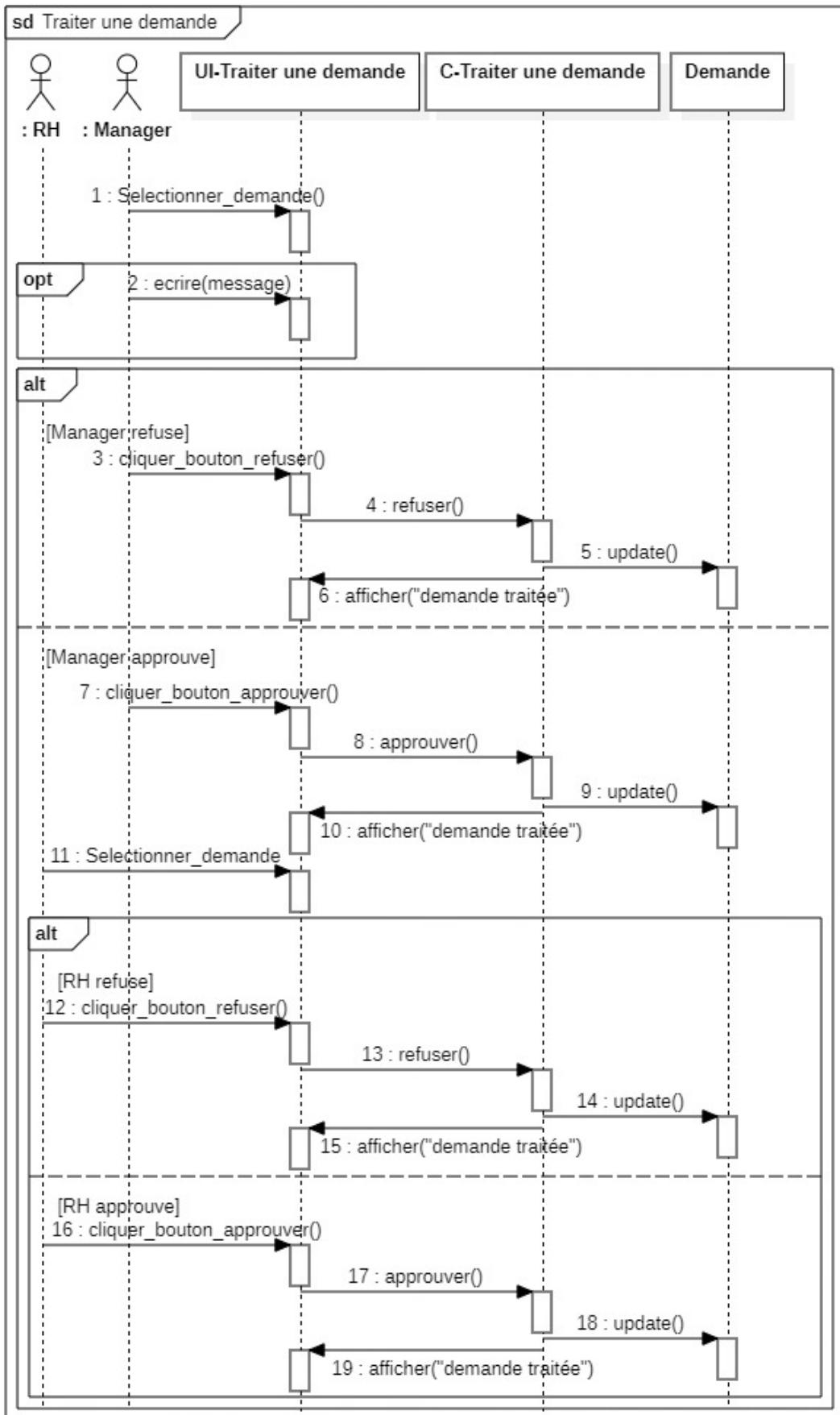


Figure 4.20: Diagramme de classe du cas d'utilisation «Traiter une demande»

##### Diagramme de Séquence

Les managers et les responsables RH ont la possibilité de traiter les demandes qui leur sont adressées. Lors du traitement d'une demande, le manager doit d'abord approuver ou rejeter la demande. Si le manager rejette, la demande est automatiquement refusée. Si le manager approuve, la demande est transmise au responsable RH qui, à son tour, doit également approuver ou rejeter. Si le responsable RH rejette, la demande est refusée ; sinon, elle est approuvée. Ce scénario est illustré dans la figure 4.21.



## 4.5 Réalisation

Dans cette partie, nous présentons les modules de notre premier sprint en utilisant des captures d'écran.

### 4.5.1 Gérer son profil

L'interface de gestion du profil présentée dans la figure 4.22 permet aux utilisateurs de consulter et de modifier leurs informations personnelles. Ils ont la possibilité de mettre à jour leur photo de profil, ainsi que de corriger leur nom et prénom en cas d'erreur. Cette fonctionnalité garantit que les données utilisateur restent exactes et à jour, offrant ainsi une expérience personnalisée et fiable.

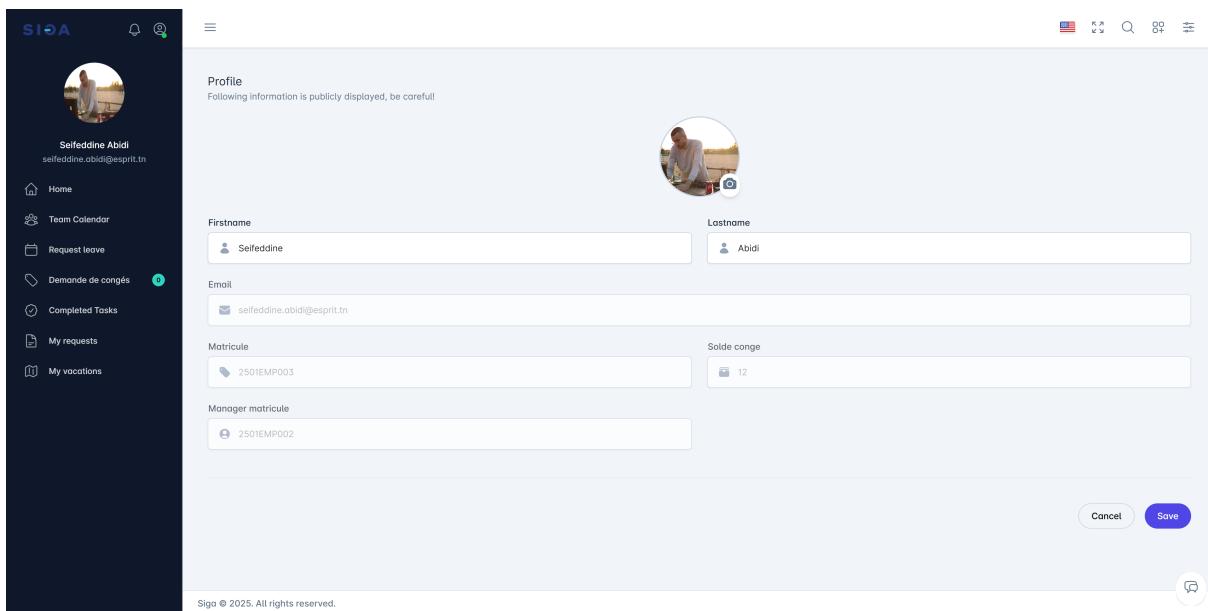


Figure 4.22: Interface du cas d'utilisation «Gérer son profil»

### 4.5.2 Réinitialiser son mot de passe

Pour réinitialiser le mot de passe, on doit suivre ces étapes :

- Accédez à la page "Mot de passe oublié ?"** : Depuis la page de connexion, cliquez sur "Forgot password?".
- Entrez votre adresse e-mail** : Saisissez l'adresse e-mail associée à votre compte dans le champ prévu ("Email address").
- Envoyez la demande** : Cliquez sur le bouton "Send reset link". Un message confirmera l'envoi d'un e-mail si votre adresse est enregistrée dans le système.
- Consultez votre boîte de réception** : Vous recevrez un e-mail intitulé "Password Reset Request" avec un bouton "Reset Password". Cliquez sur ce bouton.

5. **Créez un nouveau mot de passe** : Sur la page de réinitialisation, entrez votre e-mail, un nouveau mot de passe, et confirmez ce mot de passe dans les champs correspondants.
6. **Validez** : Cliquez sur "Reset your password" pour enregistrer votre nouveau mot de passe.

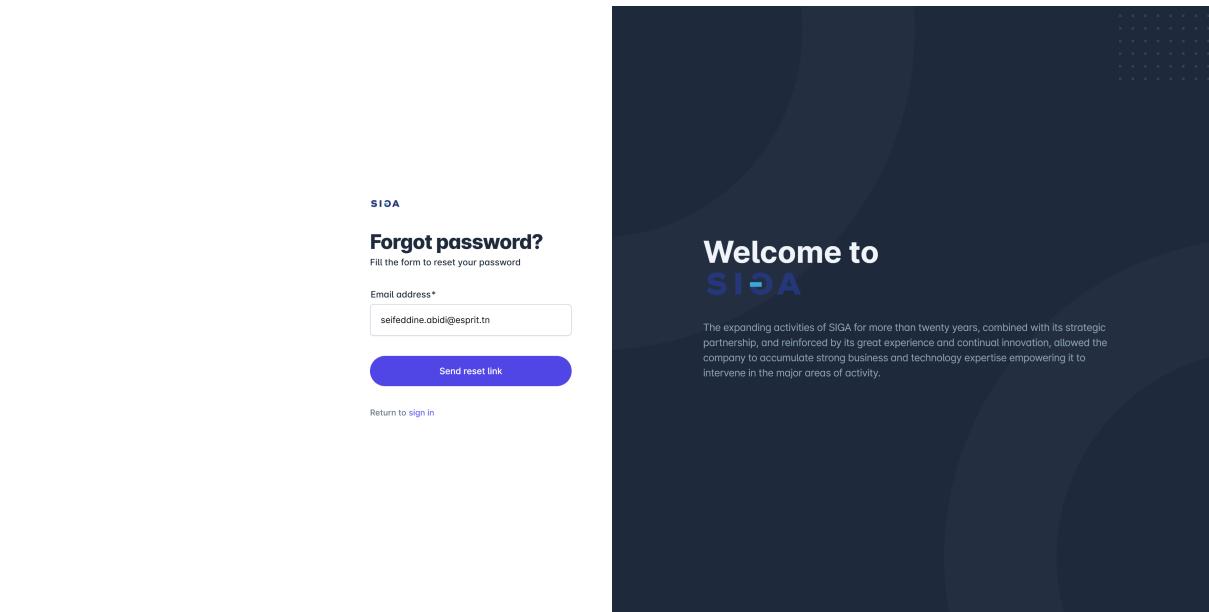


Figure 4.23: Page "Mot de passe oublié ?"

**Forgot password?**  
Fill the form to reset your password

Password reset sent! You'll receive an email if you are registered on our system.

Email address\*

Send reset link

Return to [sign in](#)

Figure 4.24: Confirmation de l'envoi du lien de réinitialisation

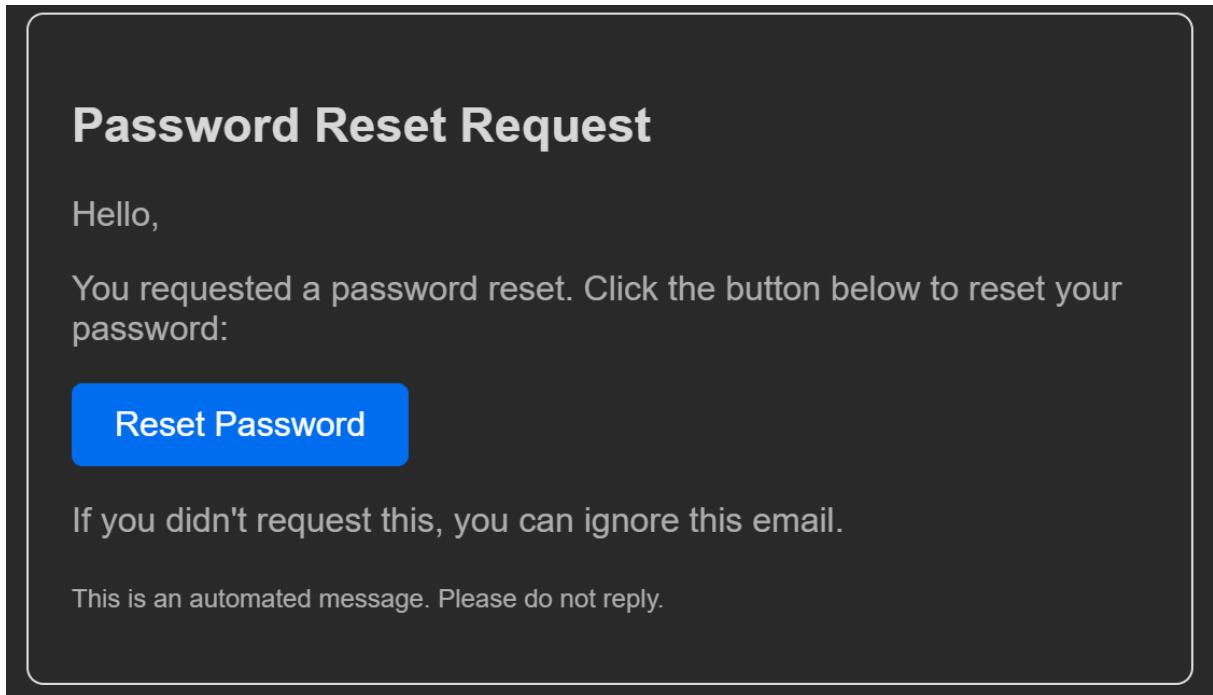


Figure 4.25: E-mail de réinitialisation de mot de passe

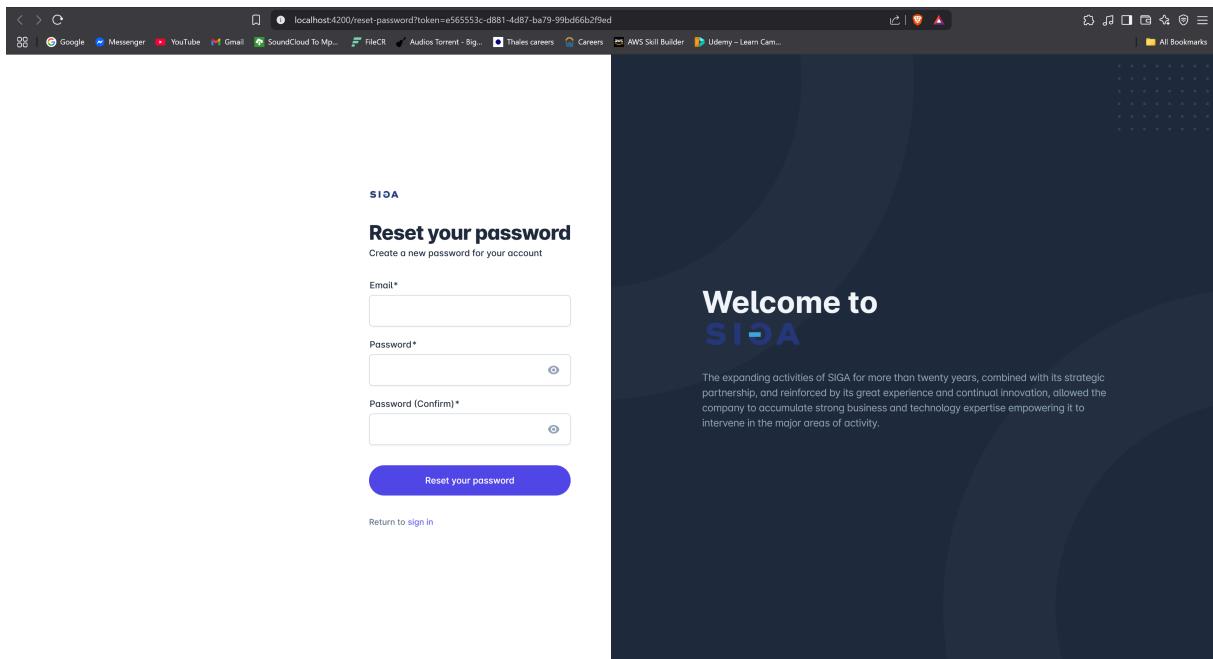


Figure 4.26: Page de création d'un nouveau mot de passe

### 4.5.3 Soumettre une demande

- 1. Sélectionner le type de la demande :** Commencez par choisir le type de demande que vous souhaitez soumettre via un menu déroulant ou une liste (par exemple, "Leave request" pour une demande de congé comme illustré dans la figure 4.27 et 4.28).
- 2. Remplir les informations nécessaires :** Indiquez les détails requis, tels que les dates de début et de fin, ainsi que les horaires si pertinents.
- 3. Ajouter des options supplémentaires :** Si applicable, précisez des préférences ou conditions spécifiques (par exemple, un départ ou un retour en milieu de journée).
- 4. Soumettre la demande :** Une fois toutes les informations saisies, validez votre demande en cliquant sur un bouton de soumission. Vous pouvez également annuler si nécessaire.

The screenshot shows a web-based application interface for creating a new request. At the top, there is a navigation bar with icons for search, refresh, and other functions. Below the navigation is a breadcrumb trail: 'Request > Add'. The main title is 'Add Request'. A dropdown menu is open, showing 'Leave request' as the selected option. The form itself has a light gray background and contains the following fields:

- A section labeled 'Request' with the sub-instruction 'Add here your request details and submit it.'
- 'Start Date\*' field with a date input box and a calendar icon.
- 'End Date\*' field with a date input box and a calendar icon.
- Two checkboxes:
  - Departure After Midday
  - Return After Midday
- At the bottom right of the form area are 'Cancel' and 'Create request' buttons, along with a small message icon.

Figure 4.27: Page de création d'une nouvelle demande de congé

Request &gt; Add

**Add Request**

Authorization request

**Authorization**  
Add here your request details and submit it.

Date \*

Start Time \*

End Time \*

**Cancel** **Submit** 

Figure 4.28: Page de création d'un nouvelle demande d'autorisation

**4.5.4 Consulter ses demandes**

Requests > List

**Requests list**

Motricule	Email	Manager	Request Date	Start Date	End Date	Number of days	GoAfterMidday	BackAfterMidday	Status
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-22 10:19:19	2025-04-30T00:00:00	2025-04-05T00:00:00	26 days	<span>NO</span>	<span>NO</span>	<span>ACCEPTED</span>
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 16:16:40	2025-05-05T12:00:00	2025-05-05T13:00:00	1 day	<span>NO</span>	<span>NO</span>	<span>ACCEPTED</span>
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 13:29:28	2025-04-07T12:00:00	2025-04-07T13:00:00	1 day	<span>NO</span>	<span>NO</span>	<span>ACCEPTED</span>
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 13:18:22	2025-05-01T12:00:00	2025-05-01T14:00:00	1 day	<span>NO</span>	<span>NO</span>	<span>ACCEPTED</span>
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 11:15:53	2025-04-30T11:00:00	2025-04-30T12:00:00	1 day	<span>NO</span>	<span>NO</span>	<span>REJECTED</span>
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 10:01:33	2025-04-29T12:00:00	2025-04-29T13:00:00	1 day	<span>NO</span>	<span>NO</span>	<span>ACCEPTED</span>
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 09:59:37	2025-04-25T12:00:00	2025-04-25T13:00:00	1 day	<span>NO</span>	<span>NO</span>	<span>ACCEPTED</span>
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 09:46:08	2025-04-28T12:00:00	2025-04-28T13:00:00	1 day	<span>NO</span>	<span>NO</span>	<span>ACCEPTED</span>
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 09:32:24	2025-04-27T12:00:00	2025-04-27T13:00:00	1 day	<span>NO</span>	<span>NO</span>	<span>ACCEPTED</span>



Figure 4.29: Page de consultation des demandes personnelles

L'interface pour consulter ses demandes sur une plateforme comme celle présentée est généralement conçue pour être intuitive et organisée. Une fois dans cette section, l'utilisateur voit une liste ou un tableau répertoriant ses demandes soumises. Chaque entrée affiche des informations clés telles que la date de soumission, les dates demandées (début et fin), et l'état de la demande (en attente, approuvée ou rejetée). Des options de tri peuvent être présentes, permettant de classer les demandes par statut ou par date.

#### 4.5.5 Consulter ses congés

L'interface pour consulter ses congés, c'est-à-dire les demandes de congés qui ont été acceptées, est conçue pour fournir une vue organisée et spécifique des congés approuvés. Une fois dans cette section, l'utilisateur voit une liste ou un tableau répertoriant uniquement ses demandes de congés ayant été acceptées. Chaque entrée affiche des informations clés telles que les dates de début et de fin du congé, la date d'approbation, et éventuellement la durée.

Matricule	Email	Manager	Request Date	Start Date	End Date	Number of days	GoAfterMidday	BackAfterMidday
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-22 10:19:19	2025-04-30T00:00:00	2025-04-05T00:00:00	26 days	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 16:16:40	2025-05-05T12:00:00	2025-05-05T13:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 13:29:28	2025-04-07T12:00:00	2025-04-07T13:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 13:18:22	2025-05-01T12:00:00	2025-05-01T14:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 10:01:33	2025-04-29T12:00:00	2025-04-29T13:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 09:59:37	2025-04-25T12:00:00	2025-04-25T13:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 09:46:08	2025-04-28T12:00:00	2025-04-28T13:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 09:32:24	2025-04-27T12:00:00	2025-04-27T13:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-04-10 08:11:48	2025-04-09T00:00:00	2025-04-14T00:00:00	6 days	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-03-07 10:36:03	2025-03-07T16:00:00	2025-03-07T17:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-02-28 15:37:20	2025-03-31T12:00:00	2025-03-31T13:00:00	1 day	NO	NO
2501EMP005	hughes.brian@company.com	Iheb Yousfi	2025-02-28 15:03:45	2025-02-21T12:00:00	2025-02-21T13:00:00	1 day	NO	NO

Figure 4.30: Page de consultation des congés acceptés

#### 4.5.6 Traiter une demande

L'interface pour traiter une demande est conçue pour permettre l'approbation ou le refus d'une demande. Le manager ou l'RH peut choisir entre deux options : accepter ou refuser la demande. Une fois la décision prise, il peut cliquer sur un bouton de validation, ce qui envoie une notification à l'utilisateur qui a soumis la demande. Les notifications peuvent être envoyées par e-mail ou par un message de notification sur la plateforme.

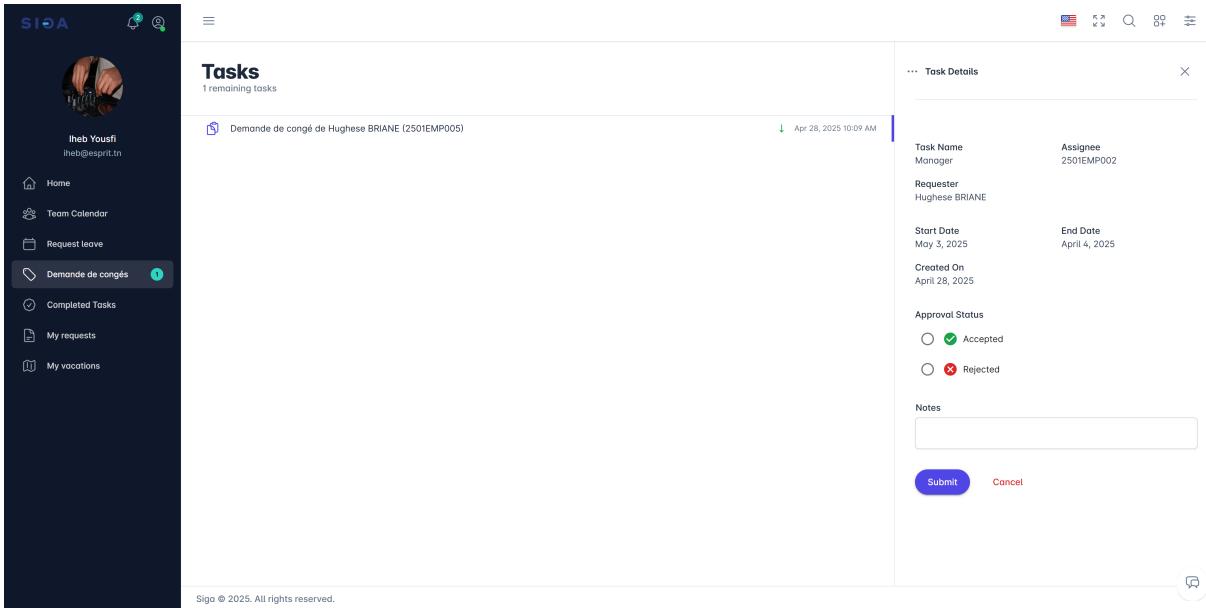


Figure 4.31: Page de traitement des demandes

## 4.6 Conclusion

Le Sprint 2 a permis de développer des fonctionnalités essentielles pour la gestion des demandes et congés, répondant aux besoins des utilisateurs, managers et RH.

Le backlog a défini les cas d'utilisation prioritaires, raffinés via des diagrammes UML (cas d'utilisation, classes, séquences) et BPMN pour modéliser les interactions et workflows.

La conception, basée sur une architecture orientée objet et intégrant Camunda, a soutenu une réalisation réussie, avec des interfaces intuitives pour gérer les profils, réinitialiser les mots de passe, soumettre, consulter et traiter les demandes.

Ce sprint a établi une gestion efficace des demandes, ouvrant la voie à de futures améliorations comme des notifications avancées ou des tableaux de bord analytiques.

# Chapitre 5

## Sprint 3 : Suivi et Supervision

### 5.1 Introduction

Fort des avancées réalisées dans les Sprints 1 et 2, qui ont mis en place les bases de l'accès, de l'administration et de la gestion des demandes, le Sprint 3 se focalise sur le suivi et la supervision au sein de la plateforme. Ce chapitre présente les fonctionnalités conçues pour permettre aux utilisateurs, managers, RH et administrateurs de superviser efficacement les processus, de consulter les demandes d'approbation, de suivre leurs équipes et de consulter les crédits de congés. Nous détaillerons les besoins identifiés, la conception de ces fonctionnalités et les étapes de leur mise en œuvre dans ce sprint.

### 5.2 Backlog du Sprint 3

Le tableau 5.1 représente le backlog du troisième sprint. Ce tableau détaille les cas d'utilisation, leurs priorités, estimations et tâches associées.

Tableau 5.1: Backlog du Sprint 3 : Suivi et Supervision

Cas d'utilisation	Priorité	Tâche
Recevoir une notification	2	Utilisateurs, Managers et RH reçoivent des alertes sur des actions importantes.
Consulter les processus métiers	3	Administrateur consulte les processus métiers définis dans le système.
Consulter les demandes d'approbation	3	Administrateur accède aux demandes soumises pour consultation.
Consulter les membres de l'équipe	3	Utilisateurs, Managers et RH accèdent aux informations des membres de leur équipe.
Consulter ses crédits	2	Utilisateurs, Managers et RH vérifient leur solde de congés ou crédits.

## 5.3 Raffinement de cas d'utilisation

Cette partie consiste à analyser et spécifier les besoins de ce troisième sprint à travers l'identification des acteurs et le raffinement des cas d'utilisations.

### 5.3.1 Identification des acteurs du troisième sprint

Les acteurs de ce sprint sont :

**Utilisateur, Manager et RH** : Consulte les membres de son équipe, reçoit des notifications et consulte ses crédits.

**Administrateur** : Consulte les processus métiers et les demandes d'approbation.

### 5.3.2 Raffinement du cas d'utilisation «Recevoir une notification»

La figure 5.1 illustre le diagramme de cas d'utilisation « Recevoir une notification »



Figure 5.1: Diagramme du cas d'utilisation «Recevoir une notification»

Tableau 5.2: Description textuelle du Cas d'utilisation «Recevoir une notification par mail»

<b>Cas d'utilisation</b>	Recevoir une notification par mail
<b>Acteur</b>	Utilisateur, Manager, RH
<b>Pré-conditions</b>	Système en marche. Acteur authentifié. Acteur a un e-mail valide configuré. Une action importante a été déclenchée (ex. : demande traitée).
<b>Post-conditions</b>	L'acteur reçoit une notification par e-mail.
<b>Scénario de Base</b>	1. Une action importante est effectuée (ex. : une demande est validée ou rejetée). 2. Le système génère une notification avec les détails de l'action. 3. Le système envoie un e-mail à l'adresse de l'acteur concerné. 4. L'acteur consulte l'e-mail dans sa boîte de réception.
<b>Exceptions</b>	Échec d'envoi (e-mail invalide, problème de serveur SMTP, erreur API).

Tableau 5.3: Description textuelle du Cas d'utilisation «Recevoir une notification push»

<b>Cas d'utilisation</b>	Recevoir une notification push
<b>Acteur</b>	Utilisateur, Manager, RH
<b>Pré-conditions</b>	Système en marche. Manager authentifié. Manager connecté à l'application avec une session WebSocket active. Une demande est soumise ou traitée (ex. : validation/rejet par RH).
<b>Post-conditions</b>	Le manager reçoit une notification push en temps réel.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. Une demande est soumise par un utilisateur ou traitée par un autre acteur (ex. : RH).</li> <li>2. Le système génère une notification avec les détails (ex. : type de demande, statut).</li> <li>3. Le serveur Spring WebSocket envoie la notification au client du manager via la connexion WebSocket.</li> <li>4. La notification s'affiche en temps réel dans l'interface du manager (ex. : pop-up ou badge).</li> </ol>
<b>Exceptions</b>	Échec de réception (connexion WebSocket interrompue, manager hors ligne, erreur serveur).

### 5.3.3 Raffinement du cas d'utilisation «Consulter les processus métiers»

La figure 5.2 illustre le diagramme de cas d'utilisation « Consulter les processus métiers ».

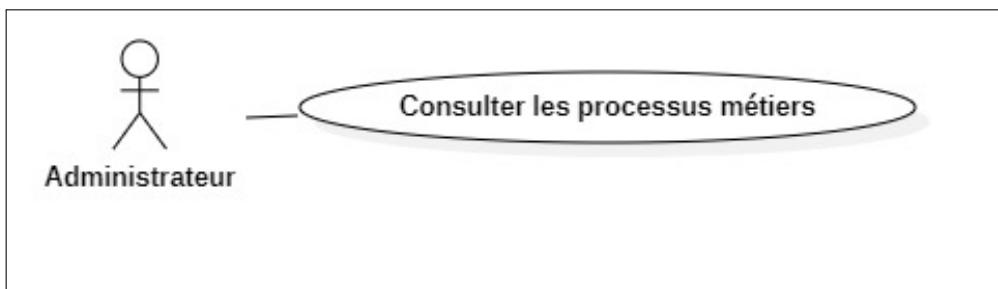


Figure 5.2: Diagramme du cas d'utilisation «Consulter les processus métiers»

Tableau 5.4: Description textuelle du Cas d'utilisation «Consulter les processus métiers»

<b>Cas d'utilisation</b>	Consulter les processus métiers
<b>Acteur</b>	Administrateur
<b>Pré-conditions</b>	<ul style="list-style-type: none"> <li>• Système en marche</li> <li>• Administrateur authentifié</li> <li>• Processus métiers déjà définis dans le système</li> </ul>
<b>Post-conditions</b>	L'administrateur visualise la liste des processus métiers configurés.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'administrateur accède à la section "Processus métiers"</li> <li>2. Le système récupère et affiche la liste des processus</li> <li>3. L'administrateur peut visualiser un processus spécifique</li> <li>4. Le système affiche les modèle BPMN du processus sélectionné</li> </ol>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Aucun processus défini → Message "Aucun processus disponible"</li> <li>• Erreur de chargement → Notification d'erreur technique</li> </ul>

### 5.3.4 Raffinement du cas d'utilisation «Consulter les demandes d'approbation»

La figure 5.2 illustre le diagramme de cas d'utilisation « Consulter les demandes d'approbation ».



Figure 5.3: Diagramme du cas d'utilisation «Consulter les demandes d'approbation»

Tableau 5.5: Description textuelle du Cas d'utilisation «Consulter les demandes d'approbation»

<b>Cas d'utilisation</b>	Consulter les demandes d'approbation
<b>Acteur</b>	Administrateur
<b>Pré-conditions</b>	<ul style="list-style-type: none"> <li>• Système en marche</li> <li>• Administrateur authentifié</li> <li>• Demandes soumises par les utilisateurs</li> </ul>
<b>Post-conditions</b>	L'administrateur visualise les demandes en attente ou historiques.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'administrateur accède à la section "Demandes d'approbation"</li> <li>2. Le système récupère les demandes (filtrables par statut/date/type)</li> <li>3. L'administrateur sélectionne une demande pour en voir les détails</li> <li>4. Optionnel : Export des données en PDF</li> </ol>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Aucune demande disponible → Afficher "Aucune demande trouvée"</li> <li>• Données corrompues → Proposer une réinitialisation du filtre</li> </ul>

Tableau 5.6: Description textuelle du Cas d'utilisation «Consulter les détails d'une demande d'approbation»

<b>Cas d'utilisation</b>	Consulter les détails d'une demande d'approbation
<b>Acteur</b>	Administrateur
<b>Pré-conditions</b>	<ul style="list-style-type: none"> <li>• Système en marche</li> <li>• Utilisateur authentifié et autorisé</li> <li>• Demande existante dans le système</li> </ul>
<b>Post-conditions</b>	L'utilisateur visualise tous les détails de la demande sélectionnée.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur sélectionne une demande dans la liste</li> <li>2. Le système affiche les informations principales</li> <li>3. L'utilisateur clique sur "Voir détails complets"</li> <li>4. Le système affiche les détails de la demande</li> </ol>
<b>Extensions</b>	<ul style="list-style-type: none"> <li>• Impression des avis de congé : génération d'un PDF</li> </ul>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Accès non autorisé → Redirection vers page d'erreur</li> <li>• Pièces jointes corrompues → Notification spécifique</li> </ul>

### 5.3.5 Raffinement du cas d'utilisation «Consulter les membres de l'équipe»

La figure 5.4 illustre le diagramme de cas d'utilisation « Consulter les membres de l'équipe ».

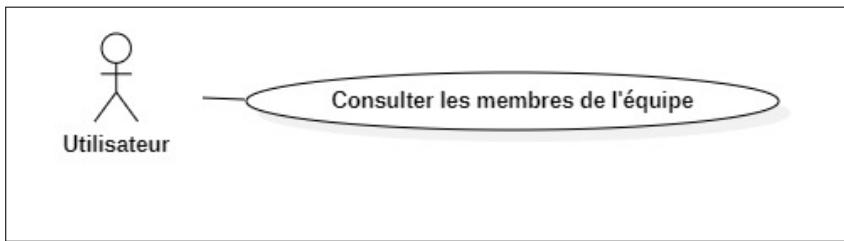


Figure 5.4: Diagramme du cas d'utilisation «Consulter les membres de l'équipe»

Tableau 5.7: Description textuelle du Cas d'utilisation «Consulter les membres de l'équipe»

<b>Cas d'utilisation</b>	Consulter les membres de l'équipe
<b>Acteurs</b>	Utilisateurs, Managers, RH
<b>Pré-conditions</b>	<ul style="list-style-type: none"><li>• Système en marche</li><li>• Utilisateur authentifié</li></ul>
<b>Post-conditions</b>	Visualisation des profils des membres de l'équipe.
<b>Scénario de Base</b>	<ol style="list-style-type: none"><li>1. L'utilisateur accède à la section "Équipe"</li><li>2. Le système affiche la liste des membres</li><li>3. L'utilisateur clique sur un membre pour voir plus de détails</li><li>4. Optionnel : Recherche par nom/fonction</li></ol>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• Équipe vide → Afficher "Aucun membre trouvé"</li><li>• Accès refusé → Rediriger vers une page d'erreur 403</li></ul>

### 5.3.6 Raffinement du cas d'utilisation «Consulter ses crédits»

La figure 5.5 illustre le diagramme de cas d'utilisation « Consulter ses crédits ».

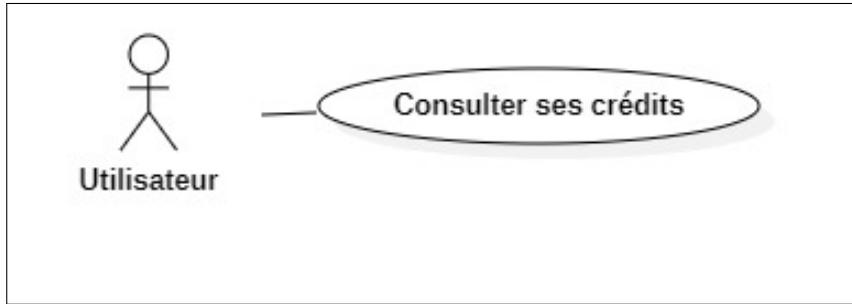


Figure 5.5: Diagramme du cas d'utilisation «Consulter ses crédits»

Tableau 5.8: Description textuelle du Cas d'utilisation «Consulter ses crédits»

<b>Cas d'utilisation</b>	Consulter ses crédits
<b>Acteurs</b>	Utilisateurs, Managers, RH
<b>Pré-conditions</b>	<ul style="list-style-type: none"> <li>• Système en marche</li> <li>• Utilisateur authentifié</li> <li>• Crédits/congés déjà attribués</li> </ul>
<b>Post-conditions</b>	Visualisation du solde de crédits(congés, autorisations) disponibles.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la section "Menu principal"</li> <li>2. Le système affiche le solde disponible</li> </ol>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Données non chargées → Afficher "Données non chargées"</li> <li>• Crédits non attribués → Afficher "Solde non disponible"</li> </ul>

## 5.4 Conception

Dans cette partie nous allons exposer la conception des cas d'utilisations de ce sprint qui se traduit par un diagramme de classe globale de ce sprint suivi par les diagrammes de classes et les diagrammes de séquences de chaque cas d'utilisation. La figure 5.6 illustre le diagramme de classe du Sprint 2.

### 5.4.1 Diagramme de classe du sprint 3

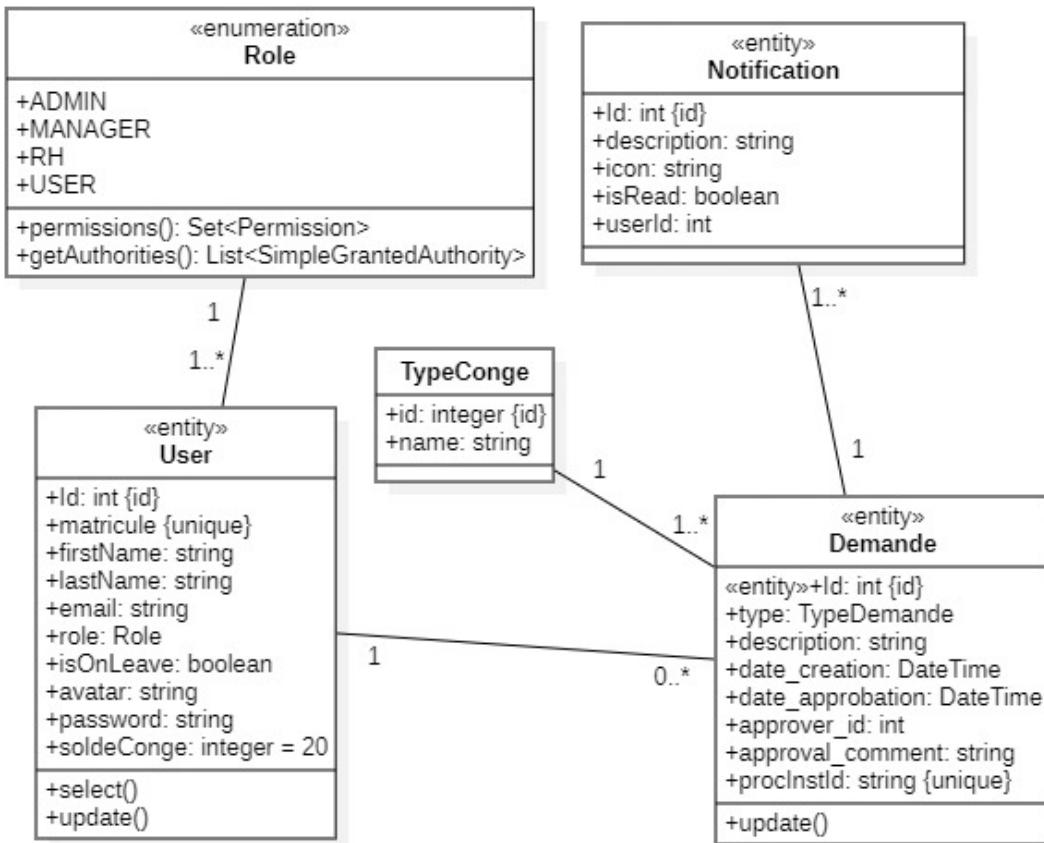


Figure 5.6: Diagramme de classe du Sprint 3

Le diagramme présente l'architecture objet centrale avec :

- **Énumération "Role"** : Définit les rôles système.
- **Classe "User"** : Modélise les utilisateurs.
- **Classe "TypeConge"** : Structure les types de congés disponibles.
- **Classe "Notification"** : Gère le système d'alertes.
- **Classe "Demande"** : Traite les requêtes utilisateurs.

## 5.4.2 Conception du cas d'utilisation «Recevoir une notification»

### Diagramme de Classe

La figure 5.7 illustre le diagramme de classes du cas d'utilisation « Recevoir une notification ».



Figure 5.7: Diagramme de classe du cas d'utilisation «Recevoir une notification»

### Diagramme de Séquence

L'utilisateur déclenche une action qui initialise une notification via le contrôleur. Le contrôleur charge les données, envoie un email si l'action nécessite une notification par email. Sinon, il insère une notification push. Ce scénario est modélisé dans la figure 5.8.

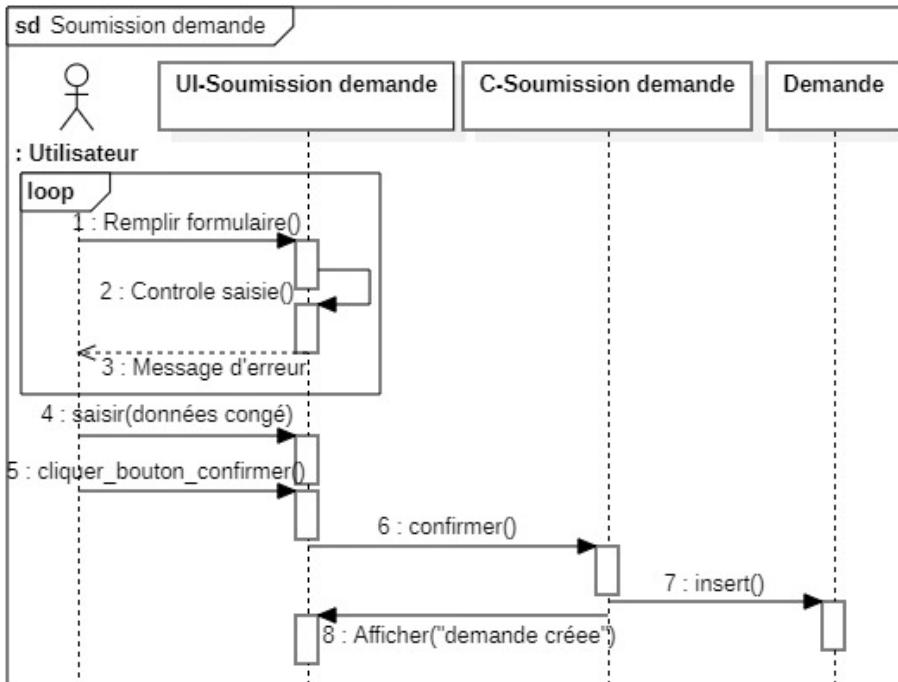


Figure 5.8: Diagramme de séquence du cas d'utilisation «Recevoir une notification»

### 5.4.3 Conception du cas d'utilisation «Consulter les processus métiers»

#### Diagramme de Classe

La figure 5.9 illustre le diagramme de classes du cas d'utilisation « Consulter les processus métiers ».

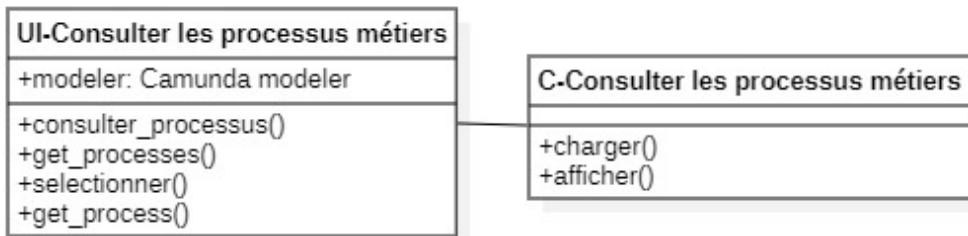


Figure 5.9: Diagramme de classe du cas d'utilisation «Consulter les processus métiers»

#### Diagramme de Séquence

L'administrateur se connecte, clique sur "Processus métiers" dans le menu principal, sélectionne un processus dans la liste affichée, et consulte le diagramme BPMN interactif avec ses étapes et règles associées.Ce scénario est modélisé dans la figure 5.10.

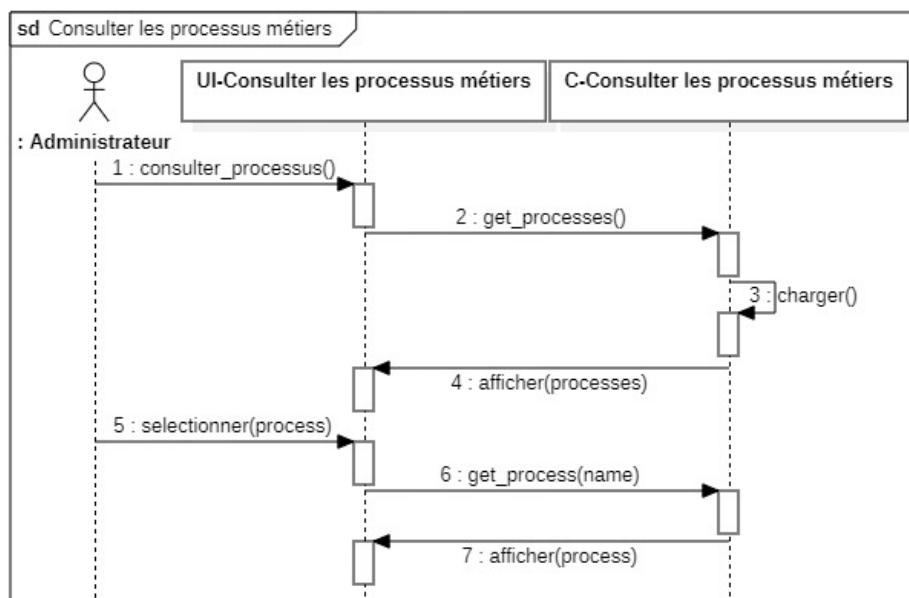


Figure 5.10: Diagramme de séquence du cas d'utilisation «Consulter les processus métiers»

## 5.4.4 Conception du cas d'utilisation «Consulter les demandes d'approbation»

### Diagramme de Classe

La figure 5.11 illustre le diagramme de classes du cas d'utilisation « Consulter les demandes d'approbation ».

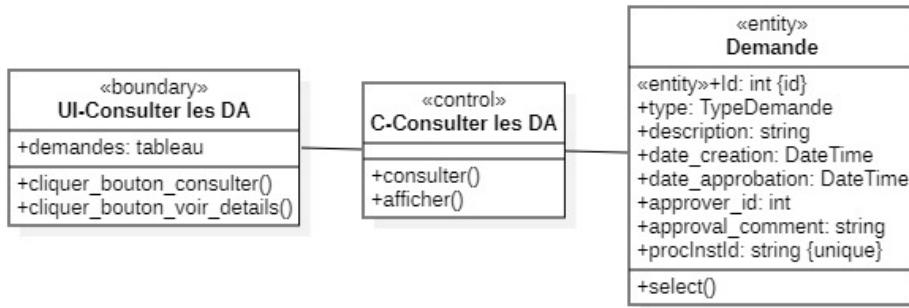


Figure 5.11: Diagramme de classe du cas d'utilisation «Consulter les demandes d'approbation»

### Diagramme de Séquence

L'administrateur consulte la liste des demandes, clique sur une demande pour afficher les détails complets (historique, documents joints, commentaires).

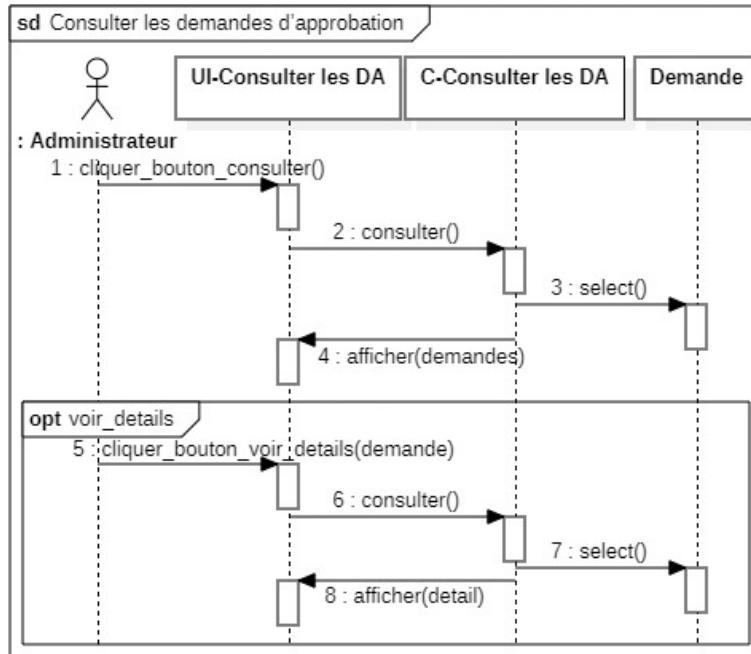


Figure 5.12: Diagramme de séquence du cas d'utilisation «Consulter les demandes d'approbation»

## 5.4.5 Conception du cas d'utilisation «Consulter les membres de l'équipe»

### Diagramme de Classe

La figure 5.13 illustre le diagramme de classes du cas d'utilisation « Consulter les membres de l'équipe ».

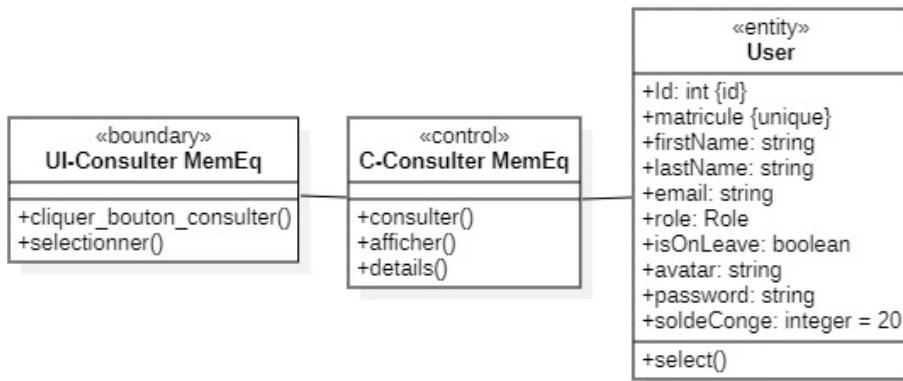


Figure 5.13: Diagramme de classe du cas d'utilisation «Consulter les membres de l'équipe»

### Diagramme de Séquence

L'utilisateur consulte la liste des membres de son équipe, clique sur un collaborateur pour afficher ses détails (email).

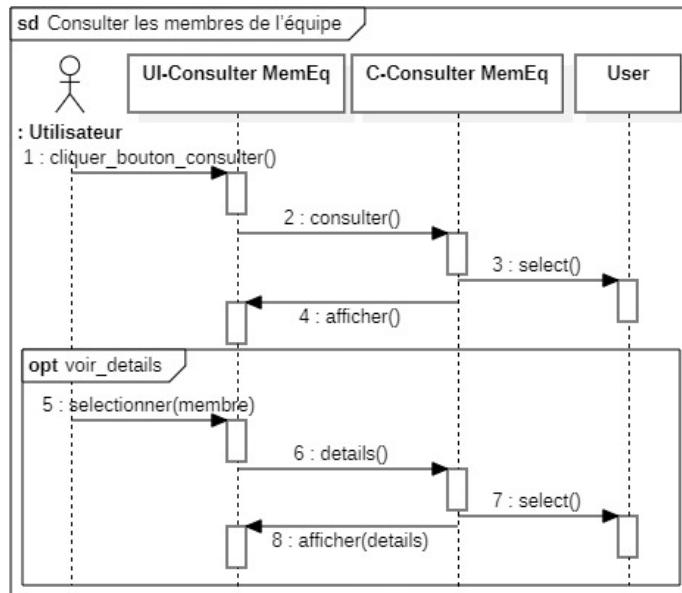


Figure 5.14: Diagramme de séquence du cas d'utilisation «Consulter les membres de l'équipe»

## 5.4.6 Conception du cas d'utilisation «Consulter ses crédits»

### Diagramme de Classe

La figure 5.15 illustre le diagramme de classes du cas d'utilisation « Consulter ses crédits ».

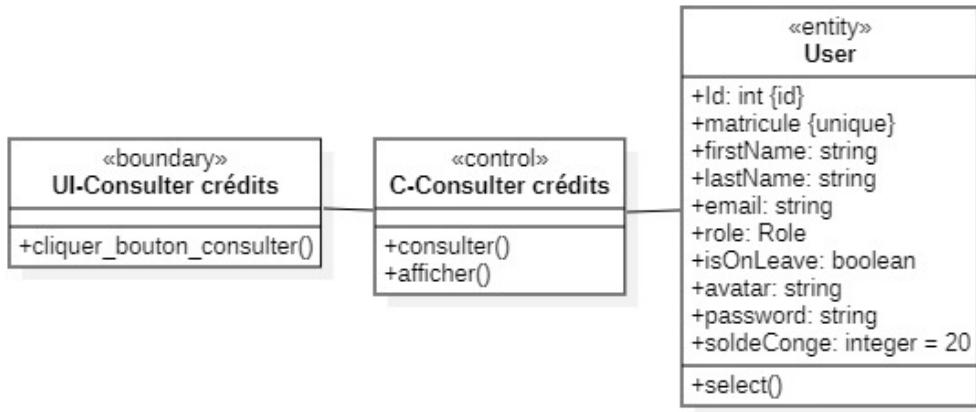


Figure 5.15: Diagramme de classe du cas d'utilisation «Consulter ses crédits»

L'utilisateur clique sur le bouton du menu principal, il consulte son solde de crédits disponibles (congés, autorisation) dans son espace personnel.

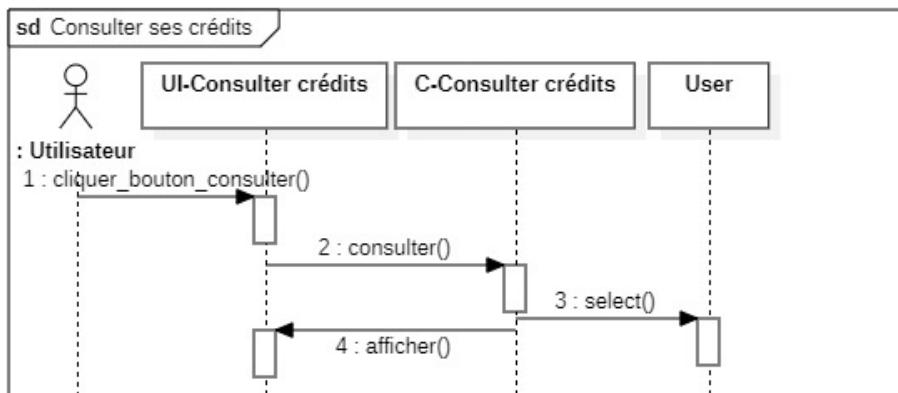


Figure 5.16: Diagramme de séquence du cas d'utilisation «Consulter ses crédits»

## 5.5 Réalisation

Dans cette partie, nous présentons les modules de notre troisième sprint en utilisant des captures d'écran.

### 5.5.1 Recevoir une notification

La figure 5.17 présente le centre de notifications unifié.

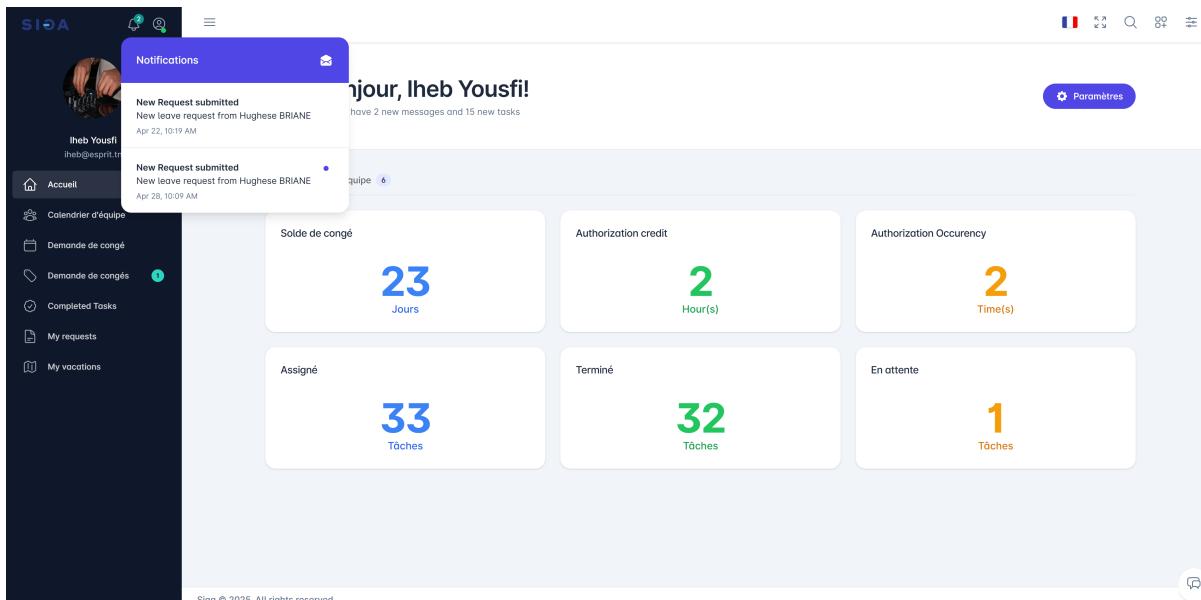


Figure 5.17: Interface du cas d'utilisation «Recevoir une notification»

### 5.5.2 Consulter les processus métiers

La figure 5.18 présente l'interface permettant aux administrateurs de consulter les workflows métiers.

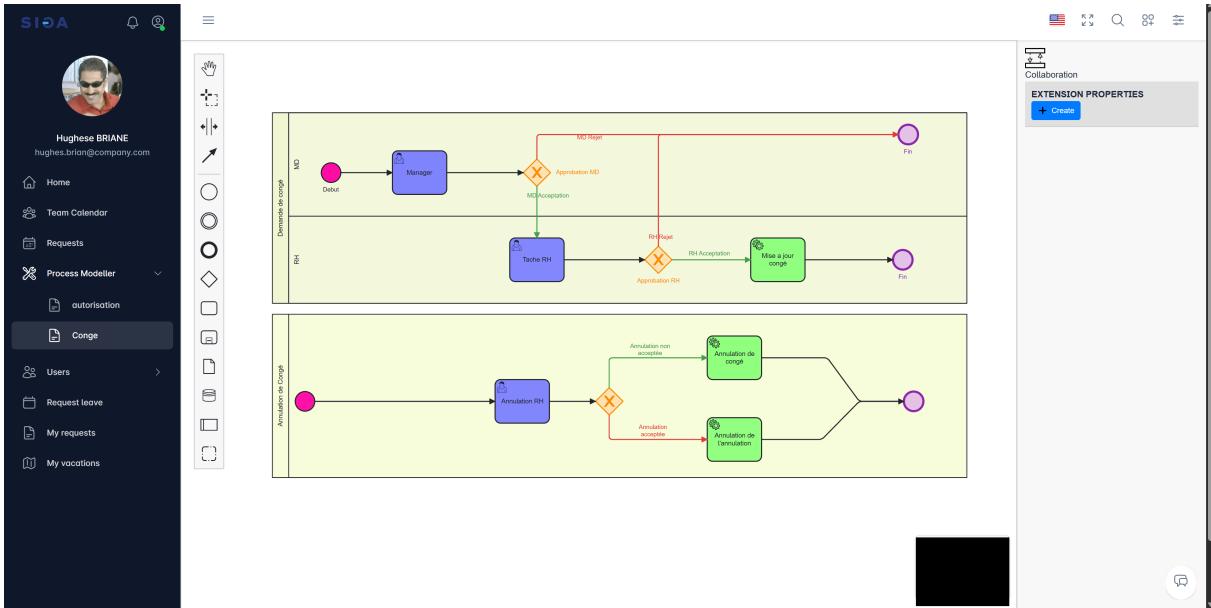


Figure 5.18: Interface du cas d'utilisation «Consulter les processus métiers»

### 5.5.3 Consulter les demandes d'approbation

Les figures 5.19 et 5.20 illustrent l'interface de consultation des demandes avec :

- Les détails de chaque demande
- L'historique complet des décisions
- Filtrage multi-critères
- Recherche dynamique sur toutes les colonnes

Figure 5.19: Interface du cas d'utilisation «Consulter les demandes d'approbation»

Figure 5.20: Interface du cas d'utilisation «Consulter les demandes d'approbation»

## 5.5.4 Consulter les membres de l'équipe

La figure 5.21 illustre l'annuaire interactif offrant une vue tabulaire des collaborateurs (photo, email).

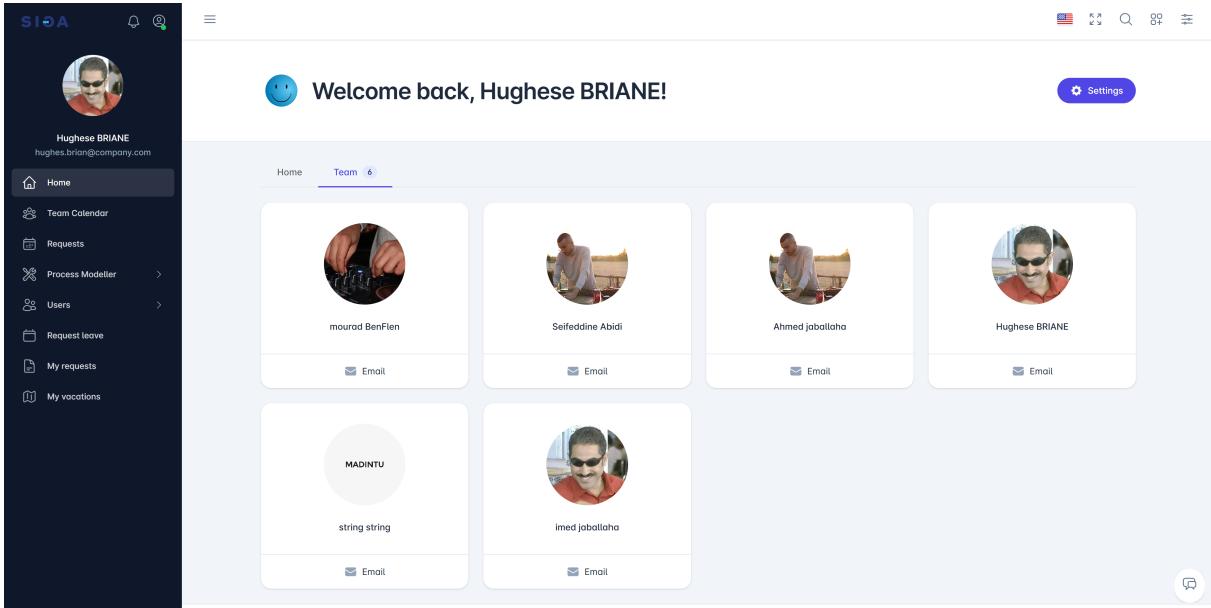


Figure 5.21: Interface du cas d'utilisation «Consulter les membres de l'équipe»

## 5.5.5 Consulter ses crédits

Le cas d'utilisation permet aux employés de visualiser leur solde de congés et autres crédits disponibles.

La figure 5.22 illustre l'interface dédiée.

**Scénario utilisateur :**

1. L'utilisateur se connecte à son espace personnel
2. Il clique sur l'onglet «Accueil»
3. Le système affiche :
  - Le solde de congé (jours disponibles)
  - Le solde d'autorisation (heures disponibles)
  - L'occurrence d'autorisation

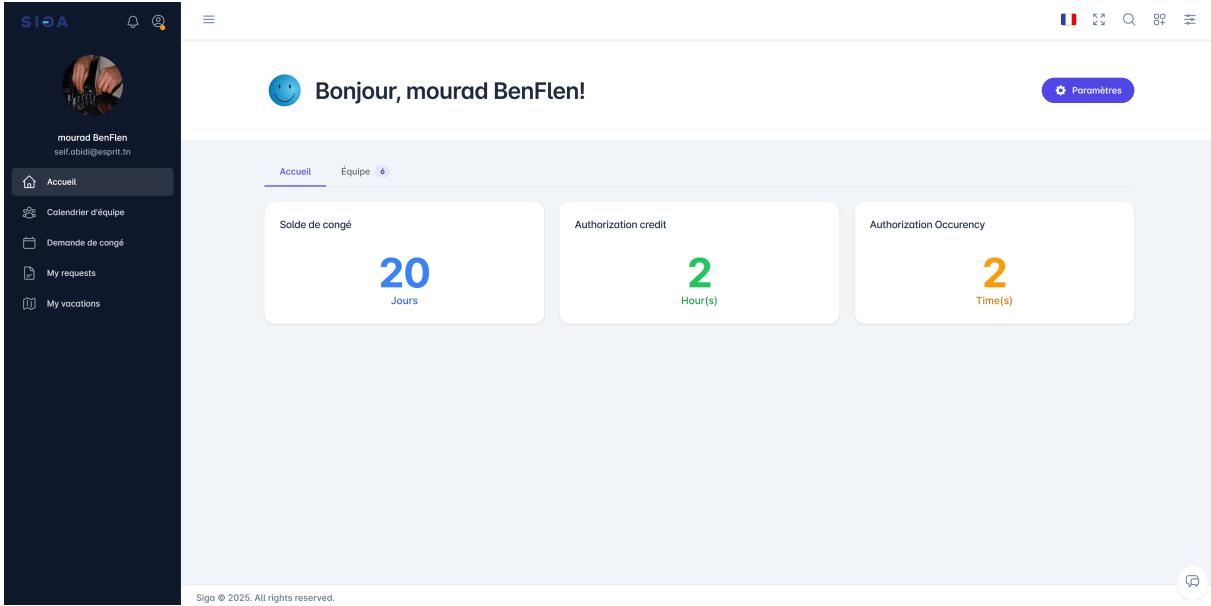


Figure 5.22: Interface du cas d'utilisation «Consulter ses crédits»

## 5.6 Conclusion

Le Sprint 3 a marqué une étape clé dans le développement de la plateforme en mettant l'accent sur le suivi et la supervision. Les fonctionnalités implémentées, telles que la réception de notifications, la consultation des processus métiers, des demandes d'approbation, des membres de l'équipe et des crédits, ont permis aux utilisateurs, managers, RH et administrateurs de mieux superviser les processus et de suivre efficacement leurs activités.

Grâce à une conception rigoureuse, illustrée par des diagrammes de classes et de séquences, ainsi qu'une réalisation soignée avec des interfaces intuitives, ce sprint a répondu aux besoins identifiés tout en posant des bases solides pour les évolutions futures.

# Chapitre 6

## Sprint 4 : Analyse et Améliorations

### 6.1 Introduction

Fort des progrès réalisés lors des sprints précédents, qui ont établi une base solide pour la gestion, le suivi et la supervision, le Sprint 4 se concentre sur l'analyse des activités et l'amélioration de l'expérience utilisateur au sein de la plateforme. Ce chapitre détaille les nouvelles fonctionnalités destinées à fournir des outils d'analyse avancés, tels que la consultation du calendrier d'équipe, des tâches accomplies et des rapports, ainsi que leur exportation. De plus, l'intégration d'un chatbot vise à assister les administrateurs dans leurs statistiques. Nous explorerons les besoins, la conception et la mise en œuvre de ces améliorations pour optimiser l'efficacité et l'interactivité du système.

### 6.2 Backlog du Sprint 4

Le tableau 3.1 représente le backlog du quatrième sprint. Ce tableau détaille les cas d'utilisation, leurs priorités, estimations et tâches associées.

Tableau 6.1: Backlog du Sprint 4 : Fonctionnalités avancées

Cas d'utilisation	Priorité	Tâche
Consulter le calendrier d'équipe	3	<ul style="list-style-type: none"><li>Les Utilisateurs, Managers et RH consultent les congés et autorisations</li></ul>
Consulter les tâches accomplies	3	<ul style="list-style-type: none"><li>Les Managers et RH suivent les tâches réalisées par eux même</li></ul>
Consulter les rapports	4	<ul style="list-style-type: none"><li>Génération de rapports d'activité (congés)</li></ul>
Implémentation d'un chatbot	4	<ul style="list-style-type: none"><li>Assistance interactive pour administrateurs</li></ul>

## 6.3 Raffinement de cas d'utilisation

Cette partie consiste à analyser et spécifier les besoins de ce quatrième sprint à travers l'identification des acteurs et le raffinement des cas d'utilisations.

## 6.4 Raffinement de cas d'utilisation

Cette partie consiste à analyser et spécifier les besoins de ce quatrième sprint à travers l'identification des acteurs et le raffinement des cas d'utilisations.

### 6.4.1 Identification des acteurs du quatrième sprint

Les acteurs de ce sprint sont :

**Utilisateur, Manager et RH** : Consultent le calendrier d'équipe, les rapports d'activité, exportent les rapports et interagissent avec le chatbot pour assistance.

**Manager et RH** : Suivent les tâches accomplies par eux-mêmes.

**Administrateur** : Interagit avec le chatbot pour une assistance interactive.

### 6.4.2 Raffinement du cas d'utilisation «Consulter le calendrier d'équipe»

La figure 6.1 illustre le diagramme de cas d'utilisation « Consulter le calendrier d'équipe ».

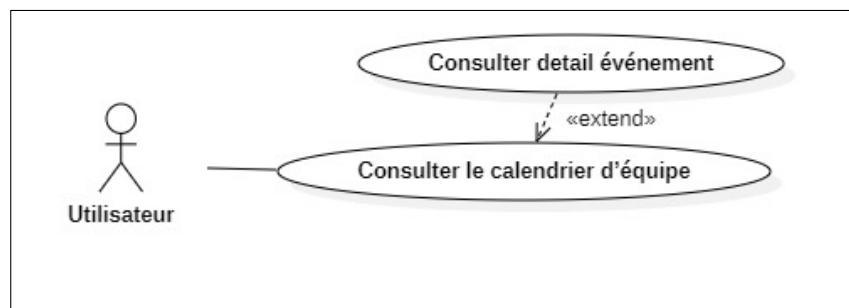


Figure 6.1: Diagramme du cas d'utilisation «Consulter le calendrier d'équipe»

Tableau 6.2: Description textuelle du Cas d'utilisation «Consulter le calendrier d'équipe»

<b>Cas d'utilisation</b>	Consulter le calendrier d'équipe
<b>Acteurs</b>	Utilisateur, Manager, RH
<b>Pré-conditions</b>	<ul style="list-style-type: none"> <li>• Système en marche</li> <li>• Acteur authentifié</li> <li>• Données d'absences et d'autorisations enregistrées dans le système</li> </ul>
<b>Post-conditions</b>	L'acteur visualise les congés et autorisations de l'équipe dans un calendrier.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'acteur accède à la section « Calendrier d'équipe »</li> <li>2. Le système affiche un calendrier avec les congés et autorisations de l'équipe</li> <li>3. Optionnel : L'acteur clique sur un événement pour voir les détails</li> </ol>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Aucune donnée disponible → Afficher « Aucun événement trouvé »</li> <li>• Problème de chargement → Notification d'erreur technique</li> </ul>

#### 6.4.3 Raffinement du cas d'utilisation «Consulter les tâches accomplies»

La figure 6.2 illustre le diagramme de cas d'utilisation « Consulter les tâches accomplies ».

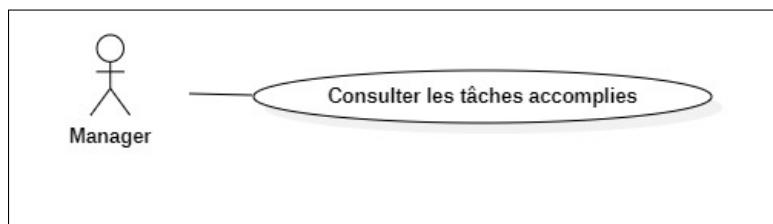


Figure 6.2: Diagramme du cas d'utilisation «Consulter les tâches accomplies»

Tableau 6.3: Description textuelle du Cas d'utilisation «Consulter les tâches accomplies»

<b>Cas d'utilisation</b>	Consulter les tâches accomplies
<b>Acteurs</b>	Manager, RH
<b>Pré-conditions</b>	<ul style="list-style-type: none"> <li>• Système en marche</li> <li>• Acteur authentifié</li> <li>• Tâches enregistrées et marquées comme accomplies</li> </ul>
<b>Post-conditions</b>	L'acteur visualise la liste des tâches accomplies par lui-même.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'acteur accède à la section « Tâches accomplies »</li> <li>2. Le système affiche la liste des tâches marquées comme terminées</li> <li>3. L'acteur peut filtrer par période ou type de tâche</li> </ol>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Aucune tâche accomplie → Afficher « Aucune tâche trouvée »</li> <li>• Erreur de chargement → Notification d'erreur technique</li> </ul>

#### 6.4.4 Raffinement du cas d'utilisation «Consulter les rapports»

La figure 6.3 illustre le diagramme de cas d'utilisation « Consulter les rapports ».

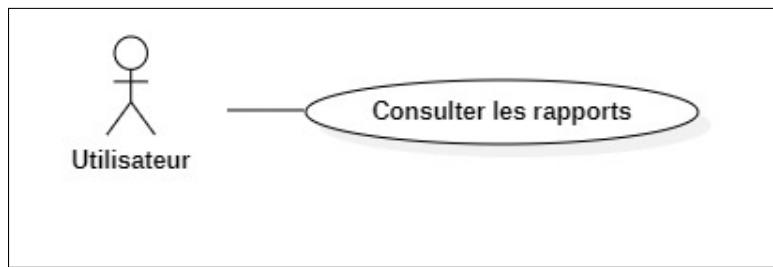


Figure 6.3: Diagramme du cas d'utilisation «Consulter les rapports»

Tableau 6.4: Description textuelle du Cas d'utilisation «Consulter les rapports»

<b>Cas d'utilisation</b>	Consulter les rapports
<b>Acteurs</b>	Utilisateur, Manager, RH
<b>Pré-conditions</b>	<ul style="list-style-type: none"> <li>• Système en marche</li> <li>• Acteur authentifié</li> <li>• Données d'activités (ex. : congés) disponibles</li> </ul>
<b>Post-conditions</b>	L'acteur visualise un rapport d'activité généré.
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'acteur accède à la section « Mes congés » et clique sur la case « Générer un rapport »</li> <li>2. Le système génère un avis de congé</li> <li>3. Le rapport est affiché avec des graphiques</li> </ol>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Aucune donnée disponible → Afficher « Aucun rapport généré »</li> <li>• Erreur de génération → Notification d'erreur technique</li> </ul>

#### 6.4.5 Raffinement du cas d'utilisation «Implémentation d'un chatbot»

Le cas d'utilisation « Implémentation d'un chatbot » a pour objectif de fournir une assistance interactive aux administrateurs en intégrant un outil d'intelligence artificielle capable d'analyser et de présenter des statistiques sur les demandes traitées dans le système.

Pour ce faire, un modèle d'IA est téléchargé via Ollama, implanté et intégré au backend à l'aide d'une API REST, puis entraîné sur les données des demandes pour générer des analyses pertinentes, comme le nombre de demandes approuvées ou rejetées par période.

Une interface web est ensuite créée et connectée au backend, permettant à l'administrateur d'interagir avec le chatbot pour obtenir des informations ou des actions assistées, améliorant ainsi l'efficacité de la gestion des processus administratifs.

Tableau 6.5: Description textuelle du Cas d'utilisation «Implémentation d'un chatbot»

<b>Cas d'utilisation</b>	Implémentation d'un chatbot
<b>Acteurs</b>	Administrateur
<b>Pré-conditions</b>	<ul style="list-style-type: none"> <li>• Système en marche</li> <li>• Administrateur authentifié</li> <li>• Serveur Ollama en marche</li> <li>• Backend opérationnel avec une API pour intégrer le chatbot</li> </ul>
<b>Post-conditions</b>	<ul style="list-style-type: none"> <li>• Le chatbot est intégré au backend et le backend à l'interface web</li> <li>• Le chatbot fournit des statistiques sur les demandes (ex. : nombre de demandes approuvées/rejetées par période)</li> <li>• L'administrateur interagit avec le chatbot via une interface web pour obtenir des informations ou des actions assistées</li> </ul>
<b>Scénario de Base</b>	<ol style="list-style-type: none"> <li>1. L'équipe met en place un modèle d'IA adapté via Ollama de traitement du langage naturel comme LLaMA</li> <li>2. Le modèle est implémenté dans l'environnement du backend et intégré via une API REST pour communiquer avec le modèle Ollama</li> <li>3. Le modèle est entraîné avec les données des demandes pour fournir des statistiques (ex. : analyse du nombre de demandes par statut ou période)</li> <li>4. Une interface web front-end est développée pour permettre à l'administrateur d'interagir avec le chatbot</li> <li>5. L'interface web est intégrée avec le backend via des appels API pour envoyer les requêtes de l'administrateur et afficher les réponses du chatbot</li> <li>6. L'administrateur ouvre l'interface du chatbot dans l'application</li> <li>7. L'administrateur pose une question (ex. : « Quelles sont les statistiques des demandes ce mois-ci ? »)</li> <li>8. Le chatbot analyse la demande, interroge les données via le backend, et répond avec des statistiques (ex. : « 20 demandes approuvées, 5 rejetées »)</li> <li>9. Optionnel : Le chatbot propose des actions (ex. : « Voulez-vous exporter ces données ? »)</li> </ol>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Erreur de connexion entre front-end et backend → Message « Échec de communication avec le serveur »</li> <li>• Demande incomprise par le chatbot → Réponse générique « Veuillez reformuler votre demande »</li> </ul>

## 6.5 Conception

Dans cette partie, nous allons détailler la conception des cas d'utilisation du Sprint 4, dédié à l'analyse et aux améliorations, à travers un diagramme de classe global de ce sprint, suivi des diagrammes de classes et des diagrammes de séquences spécifiques au cas d'utilisation, afin de modéliser les interactions et les structures nécessaires à leur réalisation.

### 6.5.1 Diagramme de classe du sprint 4

La figure 6.4 illustre le diagramme de classe du Sprint 4.

Le diagramme de classe est formé de :

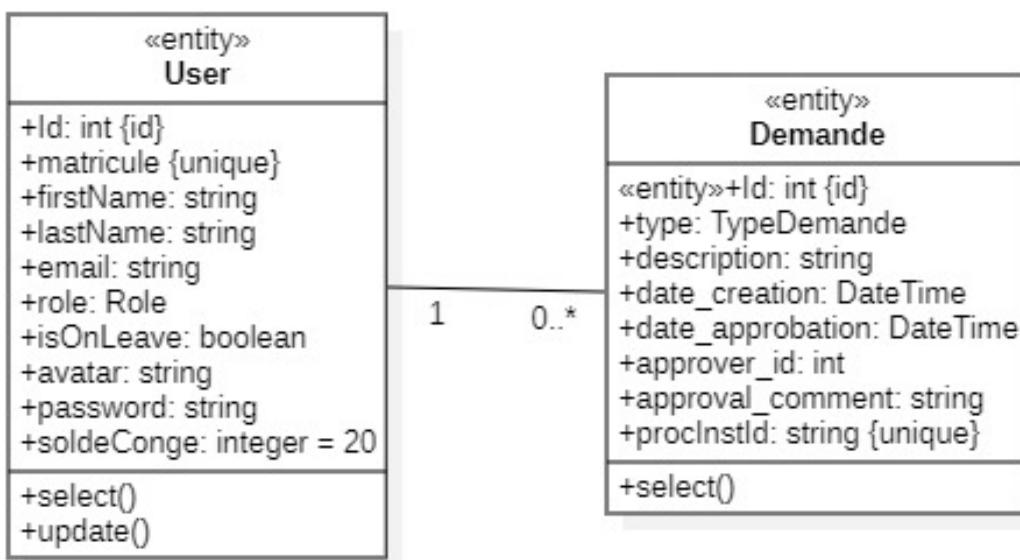


Figure 6.4: Diagramme de classe du Sprint 4

- **Classe "Demande"** : Gère les demandes soumises par les utilisateurs, comme les congés ou autorisations.
- **Classe "User"** : C'est la classe qui représente tous les utilisateurs ayant accès à la plate-forme.

### 6.5.2 Conception du Cas d'Utilisation «Consulter le calendrier d'équipe»

La figure 6.5 illustre le diagramme de classe du cas d'utilisation « Consulter le calendrier d'équipe ».

## Diagramme de Classe

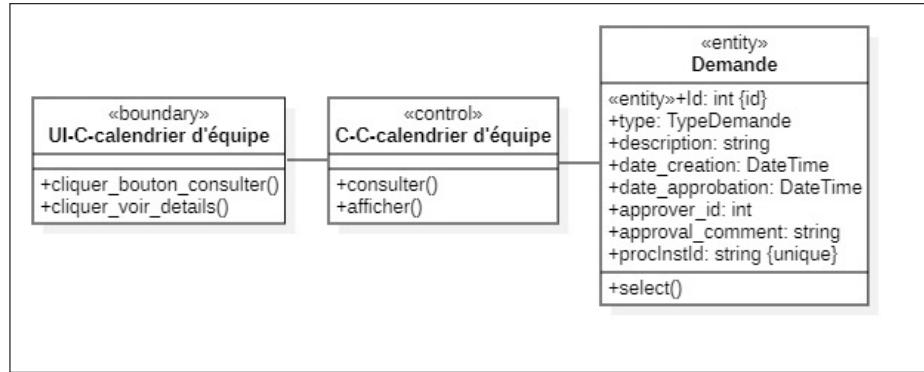


Figure 6.5: Diagramme de classe du cas d'utilisation «Consulter le calendrier d'équipe»

## Diagramme de Séquence

L'utilisateur se connecte, accède à la section « Calendrier d'équipe », consulte les événements (congés et autorisations) de son équipe, clique sur un événement, et visualise les détails tels que la date de début et la date de fin. Ce scénario est modélisé dans la figure 6.6.

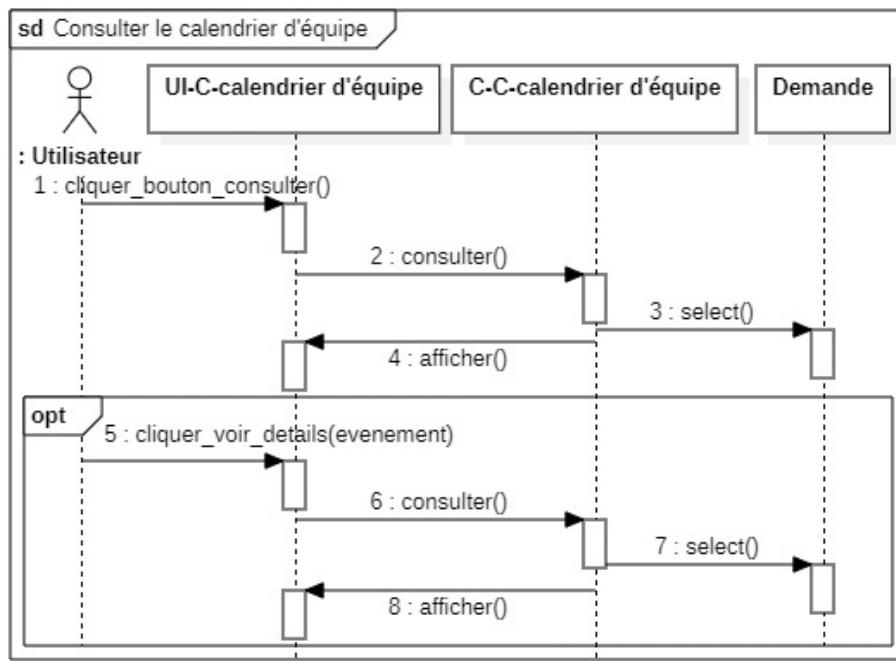


Figure 6.6: Diagramme de séquence du cas d'utilisation «Consulter le calendrier d'équipe»

### 6.5.3 Conception du Cas d'Utilisation «Consulter les tâches accomplies»

La figure 6.7 illustre le diagramme de classe du cas d'utilisation « Consulter les tâches accomplies ».

#### Diagramme de Classe

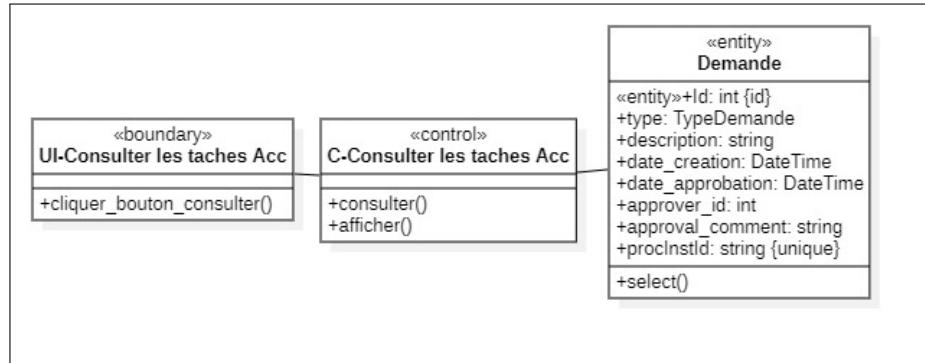


Figure 6.7: Diagramme de classe du cas d'utilisation «Consulter les tâches accomplies»

#### Diagramme de Séquence

Le manager ou RH se connecte, accède à la section « Tâches accomplies », consulte la liste des tâches qu'il a réalisées organisés par période. Ce scénario est modélisé dans la figure 6.8.

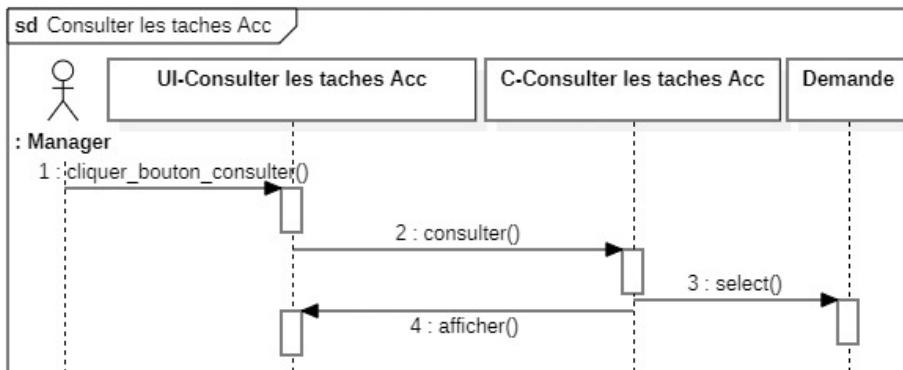


Figure 6.8: Diagramme de séquence du cas d'utilisation «Consulter les tâches accomplies»

### 6.5.4 Conception du Cas d'Utilisation «Consulter les rapports»

La figure 6.9 illustre le diagramme de classe du cas d'utilisation « Consulter les rapports ».

## Diagramme de Classe

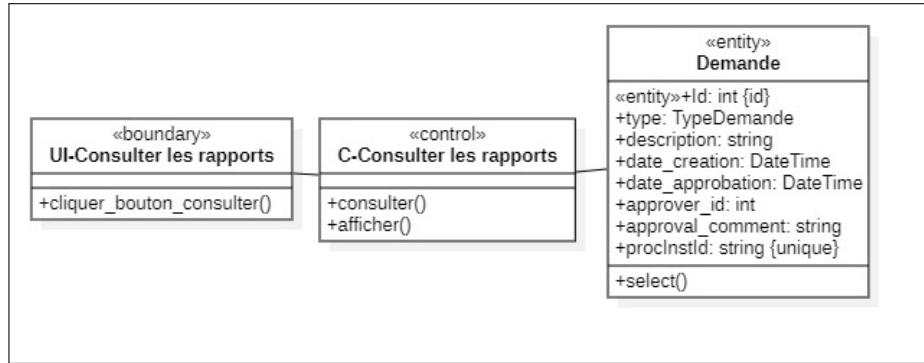


Figure 6.9: Diagramme de classe du cas d'utilisation «Consulter les rapports»

## Diagramme de Séquence

L'utilisateur, manager ou RH se connecte, accède à la section « Rapports », sélectionne un type de rapport (ex. : congés), choisit une période, et visualise le rapport généré sous forme de tableaux ou graphiques. Ce scénario est modélisé dans la figure 6.10.

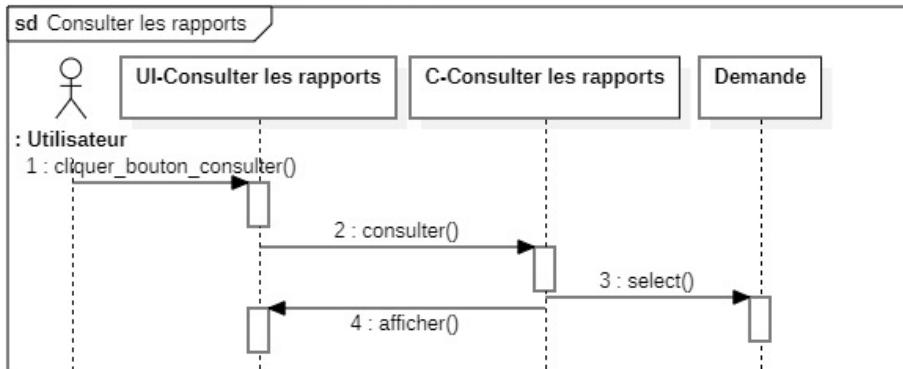


Figure 6.10: Diagramme de séquence du cas d'utilisation «Consulter les rapports»

## 6.5.5 Conception du Cas d'Utilisation «Implémentation d'un chatbot»

La figure 6.11 illustre le diagramme de classe du cas d'utilisation « Implémentation d'un chatbot ».

## Diagramme de Classe

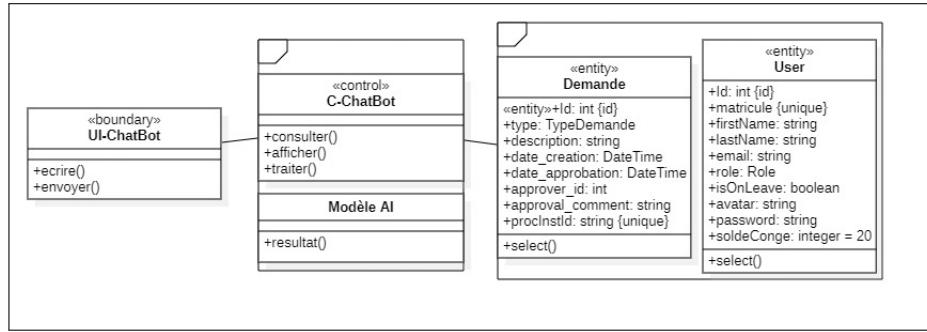


Figure 6.11: Diagramme de classe du cas d'utilisation «Implémentation d'un chatbot»

## Diagramme de Séquence

L'administrateur se connecte, ouvre l'interface du chatbot, pose une question (ex. : « Quelles sont les statistiques des congés ce mois-ci ? »), le modèle d'IA, qui connaît l'architecture des tables sauf « Demande » et « User », traduit cette interrogation en une requête SQL (ex. : SELECT COUNT(\*) FROM DEMANDE WHERE MONTH(date\_debut) = 6), l'envoie au contrôleur, qui exécute la requête et renvoie le résultat (ex. : « 15 congés ce mois-ci ») à la vue pour affichage.

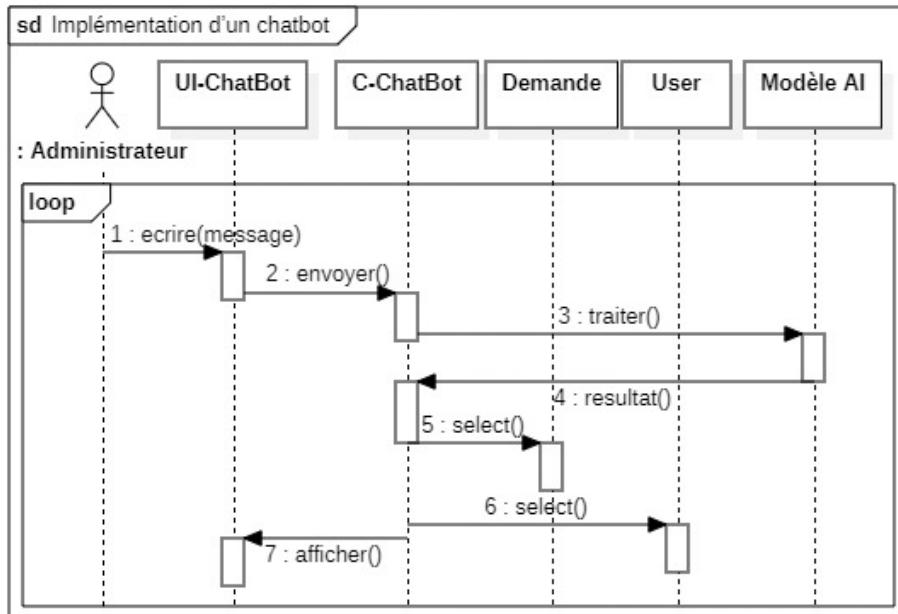


Figure 6.12: Diagramme de séquence du cas d'utilisation «Implémentation d'un chatbot»

## 6.6 Réalisation

### 6.6.1 Consulter le calendrier d'équipe

Les figure 6.13 et 6.14 présentent les interfaces permettant aux utilisateurs, managers et RH de consulter les événements de leur équipe, ainsi que les détails de ces événements.

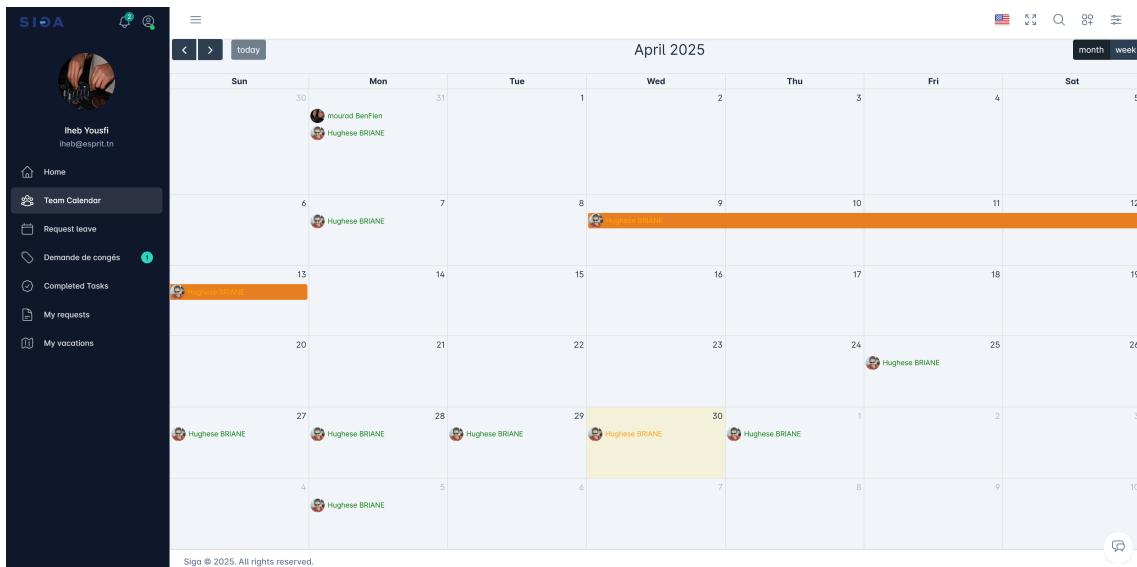


Figure 6.13: Interface du cas d'utilisation «Consulter le calendrier d'équipe»

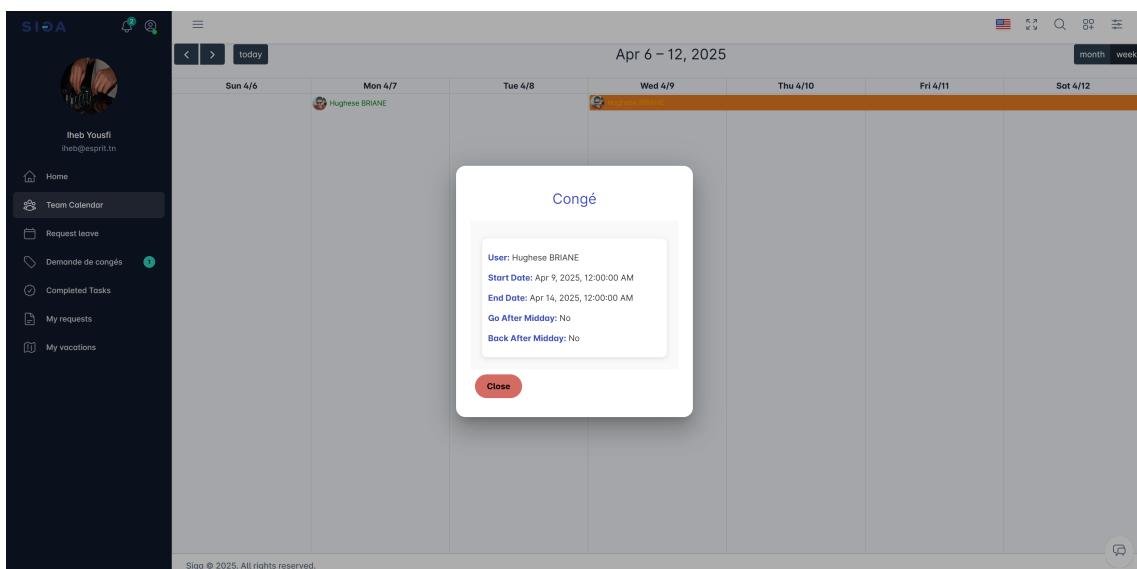


Figure 6.14: Interface du cas d'utilisation «Consulter le calendrier d'équipe (voir details)»

## 6.6.2 Consulter les tâches accomplies

La figure 6.15 illustre l'interface permettant aux managers et RH de suivre leurs tâches réalisées.

The screenshot shows a dark-themed user interface for managing tasks. On the left, a sidebar displays the user's profile (Iheb Yousfi, iheb@esprit.tn) and navigation links: Home, Team Calendar, Request leave, Demande de congés, Completed Tasks (selected), My requests, and My vacations. The main area is titled 'All Tasks' with the sub-instruction 'Tasks are listed here as individual items, starting with the most recent.' It features two time-based filters: '8 Days Ago' and '20 Days Ago'. Below these filters, a list of tasks is presented with the following details:

- Approved Manager task**: Task ID: 2501EMP005, Owner: 2501EMP005, Start Time: Apr 22, 10:19 AM.
- Approved Demande autorisation task**: Task ID: 2501EMP005, Owner: 2501EMP005, Comment: ff, Start Time: Apr 10, 4:16 PM.
- Approved Demande autorisation task**: Task ID: 2501EMP005, Owner: 2501EMP005, Comment: ff, Start Time: Apr 10, 1:30 PM.
- Approved Demande autorisation task**: Task ID: 2501EMP005, Owner: 2501EMP005, Start Time: Apr 10, 1:29 PM.
- Rejected Demande autorisation task**: Task ID: 2501EMP005, Owner: 2501EMP005, Start Time: Apr 10, 11:16 AM.

Figure 6.15: Interface du cas d'utilisation «Consulter les tâches accomplies»

## 6.6.3 Consulter les rapports

Les figures 6.16 et 6.17 présentent les interfaces de consultation des rapports, accessible aux utilisateurs, managers et RH.

The screenshot shows a dark-themed user interface for managing requests. On the left, a sidebar displays the user's profile (Hughese BRIANE, hughese.brian@company.com) and navigation links: Home, Team Calendar, Requests (selected), Process Modeler, Users, Request leave, My requests, and My vacations. The main area is titled 'Requests > List' and 'Requests list'. It includes a search bar ('Search requests') and a dropdown for 'All Types'. A table provides a detailed view of a single request:

Requester	Matricule	Manager	Request Date	Start Date	End Date	Number of days	Status	GoAfterMidday	BockAfterMidday	Type	Details
Hughese BRIANE	2501EMP005	2501EMP002	2025-02-28 11:48	2025-03-02 00:00	2025-03-03 00:00	2 days	ACCEPTED	NO	NO	Congé	<a href="#">View Details</a>

Below the table, the 'Task Details' section shows the following information:

- Validator:** 2501EMP002, **Owner:** 2501EMP005
- Description:** jowek bethy, **Start Time:** 2025-02-28 11:48:34
- End Time:** 2025-02-28 11:49:00, **Task status:** completed

The 'Tache RH' section shows similar details:

- Validator:** 2501EMP003, **Owner:** 2501EMP005
- Description:** Soudh, **Start Time:** 2025-02-28 11:49:00
- End Time:** 2025-02-28 11:49:01, **Task status:** completed

At the bottom, there are pagination controls ('Items per page: 10', '1 - 10 of 19') and a footer note ('Sipaa © 2025. All rights reserved.').

Figure 6.16: Interface du cas d'utilisation «Consulter les rapports»



Figure 6.17: Interface du cas d'utilisation «Consulter les rapports»

#### 6.6.4 Implémentation d'un chatbot

Les figure 6.18, 6.19 et 6.20 présentent l'implementation et l'interface du chatbot intégré pour assister les administrateurs.

**Scénario utilisateur :**

1. L'administrateur se connecte à son espace
2. Il ouvre l'interface du chatbot via un bouton dédié
3. Le système affiche une fenêtre de discussion où :
  - L'administrateur peut poser des questions (ex. : « Statistiques des congés ce mois-ci ? »)
  - Le chatbot répond avec des données (ex. : « 15 congés ce mois-ci »)
  - Optionnel : insertion directe des requêtes SQL

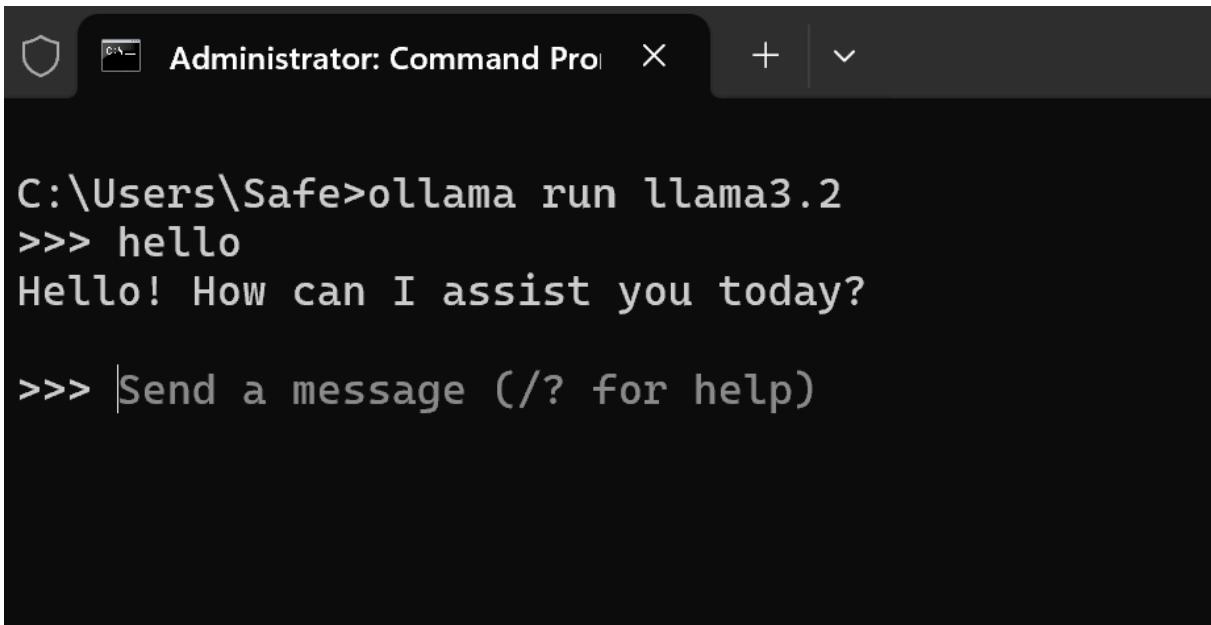


Figure 6.18: Interface du cas d'utilisation «Implémentation d'un chatbot»

```
import java.util.HashMap;
import java.util.Map;

@Service 3 usages + Seifeddine ABIDI
public class OllamaService {

    private final RestTemplate restTemplate; 5 usages
    private final String ollamaApiUrl = "http://localhost:11436/api/generate"; 3 usages
    private final String queryApiUrl = "http://localhost:8080/query/execute"; 1 usage
    private final ObjectMapper objectMapper = new ObjectMapper(); no usages

    public OllamaService(RestTemplateBuilder builder) { + Seifeddine ABIDI
        this.restTemplate = builder.build();
    }

    public String askOllama(String prompt) { 1 usage + Seifeddine ABIDI
        try {
            // Step 1: Generate SQL query or detect non-data question
            String ollamaPrompt = "Your name is Bird. You're helping with a leave management system with tables USER (id, matricule, first_name, last_name, email, solde_con
                "If the prompt asks for data, return a single SELECT SQL query. If it's not about data, return 'CHAT'. " +
                "\nPrompt: " + prompt;

            Map<String, Object> ollamaBody = new HashMap<>();
            ollamaBody.put("model", "llama3.2");
            ollamaBody.put("prompt", ollamaPrompt);
            ollamaBody.put("stream", false);

            HttpHeaders headers = new HttpHeaders();
            headers.setContentType(MediaType.APPLICATION_JSON);
            HttpEntity<Map<String, Object>> ollamaRequest = new HttpEntity<>(ollamaBody, headers);

            ResponseEntity<Map> ollamaResponse = restTemplate.postForEntity(ollamaApiUrl, ollamaRequest, Map.class);
            if (!ollamaResponse.getStatusCode().is2xxSuccessful() || ollamaResponse.getBody() == null) {
                return "Whoops, couldn't connect. Wanna try again?";
            }
        } catch (Exception e) {
            return "An error occurred while generating the response. Please try again later.";
        }
    }
}
```

Figure 6.19: Interface du cas d'utilisation «Implémentation d'un chatbot»

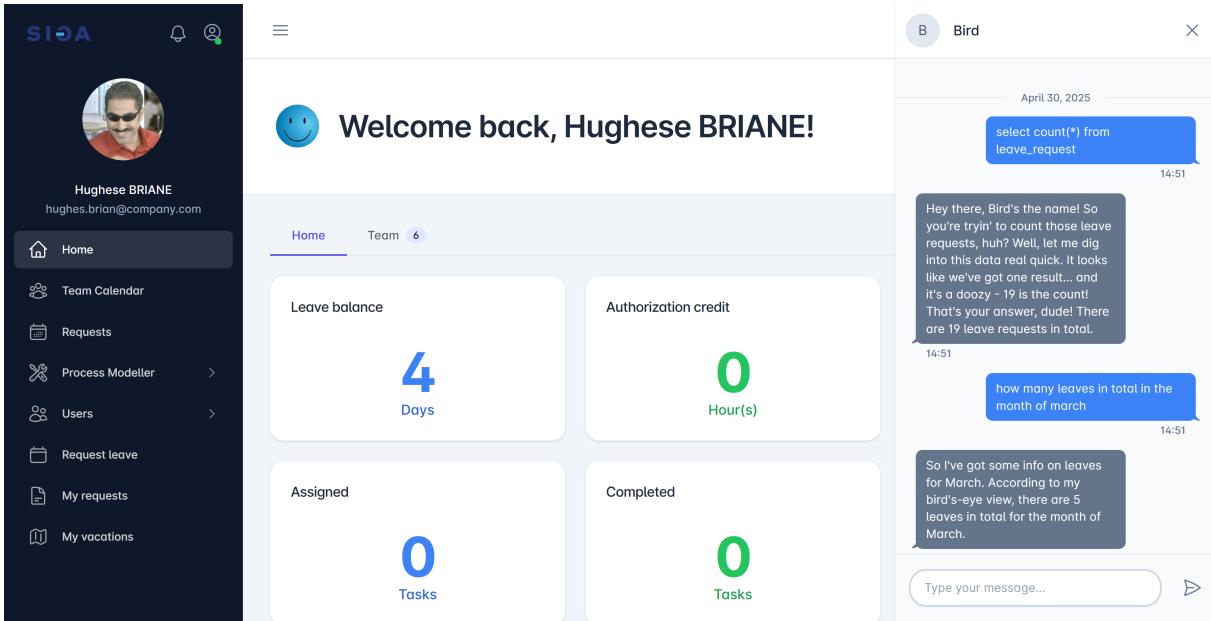


Figure 6.20: Interface du cas d'utilisation «Implémentation d'un chatbot»

## 6.7 Conclusion

Le Sprint 4 a permis de consolider les avancées des sprints précédents en introduisant des outils d'analyse et des améliorations significatives pour optimiser l'expérience utilisateur au sein de la plateforme.

Les fonctionnalités développées, telles que la consultation du calendrier d'équipe, des tâches accomplies et des rapports d'activité, offrent aux utilisateurs, managers et RH une vision claire et structurée de leurs données, renforçant ainsi leur capacité à suivre et analyser les processus.

L'intégration d'un chatbot interactif pour les administrateurs marque une avancée notable en matière d'assistance, en fournissant des statistiques pertinentes via des requêtes SQL générées dynamiquement.

Grâce à une conception détaillée et une réalisation soignée, ce sprint a répondu aux besoins identifiés tout en ouvrant la voie à de futures évolutions pour encore plus d'interactivité et d'efficacité dans la gestion des ressources humaines.

# Conclusion générale

Le présent document synthétise le projet de fin d'études réalisé au sein de l'entreprise Sofrecom Tunisie, intitulé "La refonte d'une chaîne CI/CD pour une application de gestion centralisée de correspondances". Le projet s'est articulé autour de plusieurs phases :

La première phase consistait à améliorer la chaîne CI existante en suivant les meilleures pratiques de l'équipe DevOps d'Orange. La deuxième partie impliquait la mise en place d'un mécanisme de rollback vers la version précédente en cas d'échec de déploiement pour le projet "IHM-ScribeFront". Le troisième chapitre se focalisait sur la séparation des Frontend et Backend du projet "IHMAAdminCatalogue" en mettant en place leurs propres chaînes CI/CD.

Le chapitre suivant abordait la création d'une nouvelle chaîne CI/CD pour le projet "Provider", tandis que le chapitre subséquent se concentrerait sur l'optimisation du temps d'exécution de la chaîne CI/CD pour le projet "Scribe". Enfin, le dernier chapitre avait pour objectif de réorganiser et simplifier le pipeline de PROD/PREPROD pour le projet "Scribe".

Cette réalisation a suivi une méthodologie Scrum, et malgré les défis rencontrés en cours de route, nous avons atteint nos objectifs et satisfait aux exigences du projet. Nous avons veillé à ce que notre prototype soit extensible, permettant l'ajout de serveurs de déploiement sans difficultés, en testant le déploiement sur les mêmes serveurs utilisés par l'équipe SCRIBE. De plus, nous avons appliqué avec succès la même automatisation DevOps à un nouveau projet (IHM Scribe) pour démontrer sa généralisabilité.

Dans les étapes à venir, nous prévoyons de migrer vers notre solution en déployant notre prototype sur tous les autres groupes de serveurs SCRIBE, à l'exception des groupes "ihm" et "was".

Notre stage chez Sofrecom Tunisie a été une expérience exceptionnellement enrichissante. Nous avons eu le privilège de travailler aux côtés d'une équipe collaborative et hautement compétente, ce qui a contribué à notre développement tant sur le plan académique que professionnel.

# Nétographie

## Sites Web

- ref1 Sofrecom Tunisie. Consulté le 15/05/2023.  
<https://www.sofrecom.com/fr>
- ref2 La méthodologie DevOps. Consulté le 15/05/2023.  
<https://openclassrooms.com/fr/courses/6093671-decouvrez-la-methode-devops/6183233-decouvrez-la-methode-devops>
- ref3 Guide to continuous integration. Consulté le 18/05/2023.  
<https://about.gitlab.com/2018/01/22/a-beginners-guide-to-continuous-integration/>
- ref4 Gitlab CI/CD. Consulté le 18/05/2023.  
<https://www.supinfo.com/articles/single/6822-integration-continue-gitlab-gitlab-ci-cd>
- ref5 La méthode GOROCO. Consulté le 16/05/2023.  
<http://tumador.blogspot.com/2009/08/goroco-method.html>
- ref6 DevOps pour les nuls. Consulté le 17/05/2023.  
<https://pdfslide.net/technology/devops-pour-les-nuls.html>
- ref7 Internet et réseaux sociaux. Consulté le 15/05/2023.  
<https://www.blogdumoderateur.com/chiffres-cles-internet-reseaux-sociaux-monde-avril-2023/>
- ref8 Processus DevOps. Consulté le 20/05/2023.  
<https://keltio.fr/devops/les-differentes-etapes-du-devops/>
- ref9 DEVOPS ADOPTION AND IMPLEMENTATION IN SOFTWARE DEVELOPMENT PRACTICE. Consulté le 25/05/2023.
- ref10 Méthodes agiles. Consulté le 30/05/2023.  
<https://www.qualitystreet.fr/2010/01/13/manifeste-agile/>
- ref11 Le Guide Scrum. Consulté le 27/05/2023.
- ref12 Git. Consulté le 27/05/2023.  
<https://robertgreiner.com/getting-started-with-git-and-tortoisegit-on-windows/>

ref13 GitLab et Gitlab CI. Consulté le 03/06/2023.  
<https://blog.eleven-labs.com/fr/introduction-gitlab-ci/>

ref14 Docker. Consulté le 03/06/2023.  
<https://docs.docker.com/get-started/>

ref15 Ansible. Consulté le 03/06/2023.  
<https://medium.com/formcept/configuration-management-andcontinuousdeployment>

ref16 DevOps Store. Consulté le 03/06/2023.  
<https://devops-store.rd.francetelecom.fr/dos>

ref17 VS Code. Consulté le 03/06/2023.  
<https://github.com/Microsoft/vscode>

ref18 Red et Eclipse. Consulté le 03/06/2023.  
<https://marketplace.eclipse.org/content/red-robot-editor>

ref19 Robot Framework. Consulté le 06/06/2023.  
<https://deusyss.developpez.com/tutoriels/Python/Robotframework/>

ref20 Tests d'acceptation. Consulté le 06/06/2023.  
[https://gayerie.dev/docs/testing/test\\_acceptation/test\\_acceptation.html](https://gayerie.dev/docs/testing/test_acceptation/test_acceptation.html)

ref21 Qualité de code avec Sonar. Consulté le 07/06/2023.  
[https://linsolas.developpez.com/articles/java/qualite/sonar/?page=page\\_1](https://linsolas.developpez.com/articles/java/qualite/sonar/?page=page_1)

ref22 Test et analyse de code. Consulté le 07/06/2023.  
<https://plazza.orange.com/groups/techlead-devops/blog/2019/08/16/sonarqube-comment-tester-et-analyser>

ref23 Tests automatisés. Consulté le 09/06/2023.  
[http://mattischneider.fr/software/watai/WATAI-web\\_testing\\_whitepaper.pdf](http://mattischneider.fr/software/watai/WATAI-web_testing_whitepaper.pdf)

ref24 Formation Robot Framework. Consulté le 09/06/2023.  
<https://plazza.orange.com/docs/DOC-673969>

ref25 Robot Framework. Consulté le 11/06/2023.  
<https://www.linkedin.com/pulse/robot-framework-test-automation-smart-way-agha-anas/>

ref26 Keywords Robot Framework. Consulté le 11/06/2023.  
[https://devops-store.rd.francetelecom.fr/dos?page\\_id=1150](https://devops-store.rd.francetelecom.fr/dos?page_id=1150)

ref27 Architecture test. Consulté le 11/06/2023.  
<https://medium.com/arcadia-software-development/automate-test-api-with-robot-framework-3035af1c9e2>

ref28 Composants de test. Consulté le 12/06/2023.  
<https://www.linkedin.com/pulse/robot-framework-test-automation-smart-way-agha-anas/>

ref29 Architecture Sonarqube. Consulté le 12/06/2023.  
<https://plazza.orange.com/docs/DOC-1255071>

- ref30 ODE et New Delivery. Consulté le 13/06/2023.  
<https://plazza.orange.com/docs/DOC-2089482>
- ref31 Sonar Quality Gate et Profile. Consulté le 14/06/2023.  
<https://docs.sonarqube.org/latest/instance-administration/quality-profiles/>
- ref32 Intégration de template ODE. Consulté le 14/06/2023.  
<https://orange-deployment-engine.pages.gitlab.tech.orange/ode-documentation/1-principes-fonctionnement/>
- ref33 Ode Deployment. Consulté le 14/06/2023.  
<https://plazza.orange.com/people/rahendatriprakosa.rosmedi@orange.com/blog/2021/04/20/>
- ref34 Structure de rôle Ansible. Consulté le 15/06/2023.  
<https://devops-store.rd.francetelecom.fr/dos?p=3280>
- ref35 JaCoCo-Maven Plugin. Consulté le 15/06/2023.  
<https://www.lambdatest.com/blog/reporting-code-coverage-using-maven-and-jacoco-plugin/>
- ref36 Cycle de vie Maven. Consulté le 15/06/2023.  
<https://www.baeldung.com/maven-goals-phases#:~:text=A>
- ref37 Maven enforcer plugin. Consulté le 15/06/2023.  
<https://fullstackcode.dev/2023/01/09/complete-guide-to-maven-enforcer-plugin/>
- ref38 Maven deploy plugin. Consulté le 15/06/2023.  
<https://maven.apache.org/plugins/maven-deploy-plugin/>
- ref39 Maven CI friendly plugin. Consulté le 16/06/2023.  
<https://developpaper.com/ci-friendly-versions-revision-under-maven-multi-module-architecture/>
- ref40 Maven SCM plugin. Consulté le 18/06/2023.  
<https://subscription.packtpub.com/book/application-development/9781785286124/8/ch08lv11sec84/using-the-maven-scm-plugin>
- ref41 Architecture Protractor. Consulté le 20/06/2023.  
<https://pfs-ltaas.pages.gitlab.si.francetelecom.fr/ltaas-wiki/protractor/>
- ref42 Tests unitaires Angular. Consulté le 20/06/2023.  
<https://medium.com/swlh/angular-unit-testing-jasmine-karma-step-by-step-e3376d110ab4>
- ref43 GitLab docs. Consulté le 28/06/2023.  
<https://docs.gitlab.com/ee/ci/>

# Bibliography

- [1] GitLab. A Beginner’s Guide to Continuous Integration, 2018. URL <https://about.gitlab.com/2018/01/22/a-beginners-guide-to-continuous-integration/>. Consulté le 18/05/2023.
- [2] Quality Street. Manifeste Agile, 2010. URL <https://www.qualitystreet.fr/2010/01/13/manifeste-agile/>. Consulté le 30/05/2023.
- [3] Supinfo. Intégration continue avec Gitlab et Gitlab CI/CD, 2023. URL <https://www.supinfo.com/articles/single/6822-integration-continue-gitlab-gitlab-ci-cd>. Consulté le 18/05/2023.
- [4] Tumador. La méthode GOROCO, 2009. URL <http://tumador.blogspot.com/2009/08/goroco-method.html>. Consulté le 16/05/2023.

# Bibliographie

- [1] Abdelaziz ABDELLATIF. *Ingénierie des méthodes et des processus - Méthodes agiles*, 2000.