



République Tunisienne
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
École Supérieur Privée d'ingénierie et de technologie
TEK-UP



RAPPORT DE PROJET DE FIN D'ANNÉE

Spécialité : Génie Logiciel et Systèmes d'Information

Réalisé par

Benhjal Eya

Gestion des Tickets et des Workflows

Encadrant professionnel : **M. Mechergui Bassem**

Encadrant académique : **: Mme Sawssen JALEL**

Entreprise ou le travail a été réalisé : Treetronix

Année universitaire :

2024 - 2025

Dédicace

À mes chers parents,

Je ne saurais trouver les mots justes pour exprimer toute ma reconnaissance et ma profonde gratitude pour vos sacrifices inestimables et votre soutien constant.

Merci papa, tu as toujours été mon modèle, celui sur lequel je me suis appuyé tout au long de ces années.

Merci maman, pour ton amour inconditionnel et ton appui indéfectible.

À mes chères sœurs, Rania et Asma, et à mes chers frères, Houssem et Ala,
Merci pour vos encouragements, votre présence rassurante et votre aide sincère.

À mes amies,

Merci pour les merveilleux moments partagés.

Je vous dédie ce travail en vous souhaitant de réaliser encore mieux l'année prochaine,
et un avenir rempli de belles réussites.

À mon fiancé ,

Merci pour ton amour, ta patience et ton soutien inestimable.
Ta présence à mes côtés m'a donné la force d'aller jusqu'au bout.

Je te dédie cette réussite avec tout mon cœur.

À tous mes proches,

Je vous dédie ce travail avec toute ma gratitude et mes remerciements. Toute ma reconnaissance...

Remerciements

Il est particulièrement agréable, avant de présenter cette oeuvre, d'exprimer toute ma gratitude envers les personnes qui de près ou de loin m'ont apporté leur aide inestimable lors de la réalisation de ce projet.

Je tiens tout particulièrement à remercier mes tuteurs de stage dans la société

Treetronix ,

M.Mechergui Bassem

J'adresse, mes sincères remerciements à mon encadrante académique,

M.Jalel Sawssen ,

Je tiens aussi à remercier tous les membres du jury pour avoir bien voulu examiner et juger ce travail.

Merci.

Table des matières

Introduction générale	1
1 Présentation du cadre du projet	3
1.1 Présentation du projet	4
1.1.1 Présentation de l'entreprise 3S	4
1.1.2 Les filiales du Groupe 3S	4
1.1.3 Contexte et Objectifs du projet	5
1.2 Étude de l'existant	6
1.3 Analyse et critique de l'existant	6
1.3.1 Étude de l'application Zammad Community	6
1.3.2 Étude de l'application Erex	8
1.4 Synthèse pour le choix de la solution	10
1.5 Méthodologie adoptée	11
1.5.1 Méthodologie SCRUM	11
1.5.2 Méthode de modélisation :UML	12
1.6 Environnement de développement	12
1.6.1 Environnement matériel	12
1.6.2 Environnement logiciel	13
2 Analyse et spécification des besoins	19
2.1 Spécifications des Besoins	20
2.1.1 Les Besoins fonctionnels	20
2.1.2 Les besoins non fonctionnels	22
2.1.3 Description des acteurs	22
2.2 Diagramme de structure des données global	24
2.3 Planification du backlog du produit	24
2.4 Planification de travail	32
2.4.1 Répartition des releases	32
2.5 Architecture de l'application	32
2.5.1 Architecture logique	32
2.5.2 Architecture physique	34
2.5.3 Architecture de API REST	34

2.6	Difficultés rencontrées	35
3	Release 1 : Étude et réalisation de la Release 1	36
3.0.1	Objectifs du sprint 1	37
3.1	sprint 1 :Module inscription et authentification	37
3.1.1	Backlog du sprint 1	37
3.1.2	Le diagramme de cas d'utilisation de sprint 1	39
3.1.3	Diagramme de structure de données de sprint 1	39
3.1.4	Conception	40
3.1.5	Réalisation	41
3.1.6	Interfaces d'inscription	41
3.1.7	Interfaces de verification de création compte	42
3.1.8	Interface de confirmation du mot de passe oublié	42
3.1.9	Interface de reset password	43
3.1.10	Interface de login	44
3.2	Sprint 2 :Module gestion des départements et gestion des catégories	45
3.2.1	Objectifs du sprint 3	45
3.2.2	Le diagramme de cas d'utilisation de sprint 2	49
3.2.3	Diagramme de Structure des Données sprint 2	50
3.2.4	Conception	51
3.2.5	Réalisation	56
4	Release 2 : Étude et réalisation de la Release 2	60
4.1	sprint 3 :Module gestion des utilisateurs et des rôles	61
4.1.1	Objectifs du sprint 3	61
4.1.2	Backlog du sprint 3	61
4.1.3	Spécification des besoins fonctionnels	63
4.1.4	Diagramme de Structure des Données sprint 3	64
4.1.5	Conception	64
4.1.6	Réalisation	66
4.2	Sprint 4 :Module gestion du profil et des horaires	70
4.2.1	Objectifs du sprint 4	70
4.2.2	Backlog du sprint 4	70
4.2.3	Spécification des besoins fonctionnels	73
4.2.4	Diagramme de Structure des Données sprint 4	74

4.2.5 Conception	75
4.2.6 Réalisation	77
Conclusion générale	83
Bibliographie	84
Webographie	85

Table des figures

1.1	Logo 3S	4
1.2	Emplacement de startup	5
1.3	interface de l'application Zammad	7
1.4	interface Erxes	9
1.5	Schéma d'Agile Scrum	11
1.6	Logo de systeme de modelisation	12
1.7	logo vs code	13
1.8	Logo mongodb	13
1.9	Logo mongodb compass	14
1.10	Logo trello	14
1.11	Logo github	15
1.12	Logo draw.io	15
1.13	Logo docker	15
1.14	Logo postman	16
1.15	Logo vite	16
1.16	Logo node js	16
1.17	Logo express js	16
1.18	Logo react	17
1.19	Logo tailwind css	17
1.20	logo socket.io	18
1.21	Logo jwt	18
2.1	Diagramme de cas d'utilisation global	23
2.2	Diagramme de structure des données global	24
2.3	Architecture logique Node.js	33
2.4	l'architecture logique de react	34
2.5	Architecteur physique	34
2.6	Architecture de API REST	35
3.1	Diagramme de cas d'utilisation de sprint 1	39
3.2	Diagramme structure de données de sprint 1	40
3.3	Diagramme de séquence "Inscription"	40

3.4	Diagramme de séquence "Authentification"	41
3.5	Interfaces d'inscription	42
3.6	Mail de verification compte	42
3.7	Interface pour la réinitialisation du mot de passe oublié	43
3.8	mail de réinitialisation	43
3.9	Interface de reset password	44
3.10	Interface de login	44
3.11	Le diagramme de cas d'utilisation de sprint 2	50
3.12	Diagramme de Structure des Données sprint 2	51
3.13	diagramme de séquence de cas d'utilisation "ajouter un département"	51
3.14	diagramme de séquence de cas d'utilisation "modifier un département"	52
3.15	Diagramme de séquence de cas d'utilisation "supprimer un département"	52
3.16	Diagramme de séquence de cas d'utilisation "lister les départements"	53
3.17	Diagramme de séquence du cas d'utilisation "Ajouter une catégorie"	53
3.18	Diagramme de séquence de cas d'utilisation "Modifier catégorie"	54
3.19	Diagramme de séquence de cas d'utilisation "Supprimer catégorie"	55
3.20	Diagramme de séquence de cas d'utilisation "lister les categories"	55
3.21	Interface d'ajout département	56
3.22	Interface de modification département	56
3.23	Interface de supression un department	57
3.24	Interface de liste des départements	57
3.25	Interface d'ajout catégorie	58
3.26	Interface de modification de catégorie	58
3.27	Interface de supression de catégorie	59
3.28	Interface de liste des categories	59
4.1	Diagramme de cas d'utilisation "gestion des utilisateurs et des rôles"	63
4.2	Diagramme de Structure des Données sprint 3	64
4.3	Diagramme de séquence de cas d'utilisation "Ajouter utilisateur"	65
4.4	diagramme de séquence de cas d'utilisation "modifier un utilisateur"	65
4.5	diagramme de séquence de cas d'utilisation "supprimer un utilisateur"	66
4.6	Formulaire d'ajout d'utilisateur	67
4.7	L'email contenant les coordonnées de connexion	67
4.8	Interface affiche le détails d'un utilisateur "Client"	68

4.9 Interface Changement de rôle	68
4.10 Interface de modifier les données d'un utilisateur	69
4.11 Interface de la liste des utilisateurs	69
4.12 Interface de suppression un utilisateur	70
4.13 Diagramme cas d'utilisation du sprint "Gestion du profil et des horaires".	74
4.14 Diagramme de Structure des Données sprint 4	74
4.15 Diagramme de séquence du cas d'utilisation "afficher profil"	75
4.16 Diagramme de séquence du cas d'utilisation "update profil".	75
4.17 diagramme de séquence du cas d'utilisation "Supprimer compétence"	76
4.18 Diagramme de séquence "Modifier horaire"	76
4.19 Diagramme de séquence "Supprimer"	77
4.20 Interface de profil agent partie "Informations"	78
4.21 Interface de profil agent partie "sécurité"	78
4.22 Interface de profil agent partie "compétence"	79
4.23 Interface de profil agent partie disponibilités	79
4.24 Interface de gestion horaires	80
4.25 Interface de profil client partie "Informations"	80
4.26 Interface de profil client partie "Sécurité"	81

Liste des tableaux

1.1	Points forts et points faibles de l'application Zammad	8
1.2	Points forts et points faibles de l'application erex	10
1.3	Les acteurs principaux dans SCRUM	11
1.4	Environnement matériel	12
2.1	Acteurs du système	22
2.2	Backlog de produit	25
2.3	Répartition des releases et des sprints selon les dépendances fonctionnelles	32
3.1	Backlog de sprint 1	37
3.2	Backlog de sprint 2	45
4.1	Backlog de sprint 3	61
4.2	Backlog de sprint 3	70

Liste des abréviations

- **API** = Application Programming Interface
- **DAO** = Data Access Object
- **DI** = Dependency Injection
- **DTO** = Data Transfer Object
- **GLSI** = Génie Logiciel et Système d'Information
- **GRH** = Gestion des Ressources Humaines
- **HRM** = Human Ressource Management
- **HTTP** = Hypertext Transfer Protocol
- **IT** = Information Technology
- **JSON** = JavaScript Object Notation
- **JWT** = JSON Web Token
- **REST** = Representational state transfer
- **RH** = Ressources Humaines
- **UML** = Unified Modeling Language

Introduction générale

Dans un environnement professionnel en constante évolution, la gestion efficace des demandes, incidents et processus internes représente un enjeu stratégique pour les entreprises et les organisations. Qu'il s'agisse de traiter des réclamations clients, de suivre les interventions techniques, ou de piloter les tâches entre différents services, la capacité à organiser et automatiser ces flux d'information devient essentielle.

C'est dans cette optique qu'un système de gestion des tickets et des workflows s'impose comme une solution incontournable. Il permet non seulement de centraliser les demandes, d'en assurer le suivi et la traçabilité, mais également de structurer les processus métiers selon des règles prédéfinies, favorisant ainsi la réactivité, la transparence et l'efficacité opérationnelle.

Ce projet vise à concevoir et développer une plateforme qui répond à ces besoins. Elle devra offrir une interface intuitive pour la création, l'affectation et le traitement des tickets, tout en intégrant un moteur de workflow configurable pour automatiser les enchaînements de tâches. Cette solution contribuera à l'amélioration de la communication entre les équipes, à la réduction des délais de traitement, et à l'optimisation des ressources internes.

- Dans un premier chapitre, nous présenterons le cadre général du projet. Il sera constitué d'une présentation du contexte général du projet ainsi que d'une étude de l'existant. Ce premier chapitre contiendra aussi une présentation de la méthode de gestion de projet et de la méthode de conception que nous avons choisi d'adopter. Ensuite, l'environnement de développement.
- Le second chapitre présentera une analyse et une spécification des besoins, fonctionnels et non fonctionnels, réalisés à partir de l'identification des acteurs de l'application. Il exposera aussi le diagramme de cas d'utilisation globale, ainsi que le diagramme de classes de l'application, le backlog de produit et l'architecture de l'application.
- Le troisième chapitre portera sur la Release 1 : «gestion du département et catégories », qui a consisté en la réalisation d'un sprint 1 incluant la création de compte de l'agent, compte de client et la gestion des utilisateurs, ainsi la réalisation d'un sprint 2, portant l'inscription, l'authentification et la gestion des rôles .
- Dans un quatrième chapitre, nous présenterons la Release 2, « gestion des tickets », qui comprend le Sprint 3, portant sur la gestion des réclamations par le client et l'assignation de tickets à un agent et la résolution de réclamation .
- Enfin, dans un cinquième et dernier chapitre, nous présenterons le sprint 4, « gestion des profils,

gestion des notifications et gestion des workflows portant sur la réalisation d'un dashboard de suivi des détails des contenus.

PRÉSENTATION DU CADRE DU PROJET

Plan

1	Présentation du projet	4
2	Étude de l'existant	6
3	Analyse et critique de l'existant	6
4	Synthèse pour le choix de la solution	10
5	Méthodologie adoptée	11
6	Environnement de développement	12

Introduction

Ce chapitre introductif présente le projet intitulé « Gestion des tickets et des workflows ». Nous débutons par une présentation du Groupe 3S, puis nous détaillons la startup filiale dans laquelle nous avons réalisé notre stage de projet de fin d'étude.

Ensuite, nous expliquons le contexte et les objectifs du projet. Enfin, nous présentons la méthodologie du travail que nous adoptons ainsi que l'environnement de travail.

1.1 Présentation du projet

1.1.1 Présentation de l'entreprise 3S

STANDARD SHARING SOFTWARE est une société tunisienne à vocation technologique aux nombreuses filiales spécialisées également dans la formation aux nouvelles technologies, les réseaux , le développement des logiciels web et mobile, les codes-barres, les centres d'appel et le commerce électronique. 3S est un intégrateur d'infrastructures et de services qui est parvenu, au fil de ses 30 ans d'existence, à se positionner en tant que leader sur le marché de l'intégration et des services à valeur ajoutée IT (Technologies de l'information) .



FIGURE 1.1 : Logo 3S

1.1.2 Les filiales du Groupe 3S

Les filiales du Groupe 3S Le Groupe 3S est un acteur majeur dans les domaines des technologies de l'information (IT) et des énergies renouvelables (CleanTech), avec une forte orientation vers l'innovation. Il regroupe une douzaine de filiales, dont Treetronix, GlobalNet, DVANCIA Training et Africa Solar. Notre stage au sein du groupe nous a permis de mieux comprendre sa structure fonctionnelle et la coordination entre ses différentes entités.

1.1.2.1 Présentation de la startup Treetronix

Treetronix est une startup technologique tunisienne, fondée en 2018 et membre du Groupe 3S. Spécialisée dans le développement de solutions innovantes, elle intervient dans les domaines de l'électronique embarquée, des systèmes d'information, de l'Internet des objets (IoT) et de l'intelligence

artificielle.

Treetronix conçoit des technologies adaptées à des secteurs clés comme l'agriculture intelligente, la logistique et les villes intelligentes.

Situé est basée à Charguia 2, Tunis, dans l'Immeuble Nefissa.



FIGURE 1.2 : Emplacement de startup

1.1.3 Contexte et Objectifs du projet

1.1.3.1 Contexte

La gestion des tickets et des workflows occupe une place centrale dans les entreprises, car elle permet d'optimiser le traitement des réclamations tout en assurant une meilleure coordination entre les différents acteurs (clients, agents, administrateurs). Elle contribue à structurer les processus internes, à clarifier les responsabilités et à suivre chaque demande de manière rigoureuse.

Dans ce contexte, notre projet de fin d'études consiste à concevoir et développer une application complète de gestion des tickets et des workflows.

1.1.3.2 Objectifs du projet

L'objectif principal est de mettre en place un système permettant de résoudre les réclamations clients sous forme de tickets, ainsi que la modélisation des workflows régissant leur traitement, présentés sous forme de tableau Kanban. Ce projet vise à :

- Gérer les départements.

- Gérer les catégories.
- Gérer les tickets.
- Gérer les utilisateurs .
- Permettre aux clients de soumettre des réclamations, avec la possibilité d'ajouter des pièces jointes.
- Permettre à l'administrateur d'assigner les tickets aux agents en fonction du département auquel ils appartiennent, avec l'envoi d'un e-mail à l'agent pour l'informer qu'un nouveau ticket lui a été attribué.
- Suivre l'évolution des tickets grâce au changement de statuts (en attente, en cours, résolu) .
- Offrir à l'administrateur une vision globale et un contrôle sur l'ensemble des opérations à l'aide d'un tableau de bord et de statistiques
- Gérer les profils utilisateurs selon les rôles attribués.
- Envoyer des notifications par e-mail automatiques en fonction des actions effectuées, notamment au client lorsque le ticket est résolu.
- Assurer une assistante virtuelle ,via chatbot pour aider le client à résoudre ses problème .
- Automatiser le workflow de traitement, depuis de la création du ticket à sa résolution sous forme de tableaux kanban.
- Assurer une traçabilité claire dans la résolution des demandes.

1.2 Étude de l'existant

L'étude de l'existant est une étape indispensable, elle permet d'extraire les forces et les faiblesses des applications existantes. Cela nous aidera à la réalisation de notre projet. Nous avons choisi d'analyser deux applications existantes.

- L'application « Zammad Community »
- L'application « Erxes »

1.3 Analyse et critique de l'existant

1.3.1 Étude de l'application Zammad Community

Adresse (URL) :<https://zammad.org/>

1.3.1.1 Description

Cette application a pour but le gestion des tickets multi-canaux, gestion des utilisateurs et des agents, Interface utilisateur moderne, Rapports et Statistiques.... Le public ciblé de cette application est les entreprises (petite, moyenne ou grande entreprise).

Zammad dispose d'une application Web, mais pas d'application mobile native officielle

- **Interface du site :** C'est l'interface d'accueil qui contient la description du site.

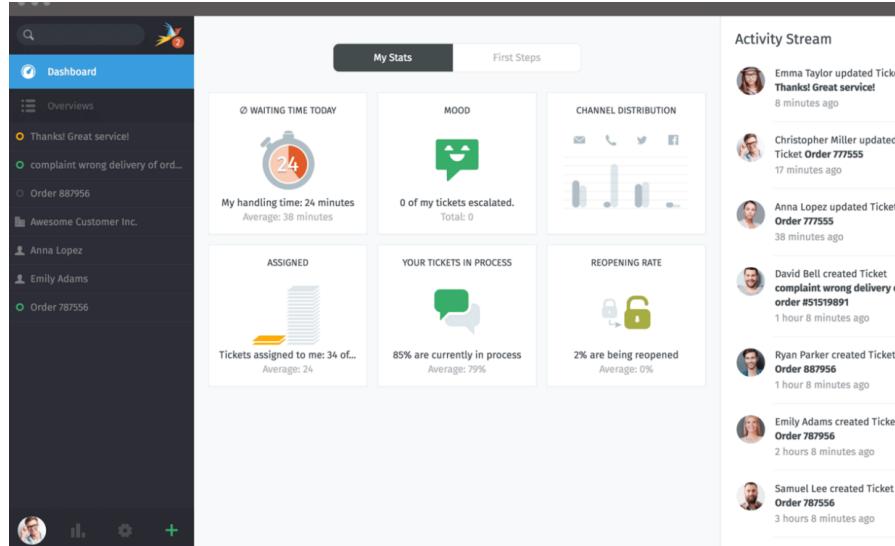


FIGURE 1.3 : interface de l'application Zammad

1.3.1.2 Étude fonctionnelle

Les principales fonctionnalités proposées par l'application, tout en les reliant aux acteurs qui en bénéficient.

- **Client :**

- Créer un ticket via formulaire, email ou chat.
- Lire les messages clients et y répondre.
- Changer le statut du ticket.
- Ajouter des pièces jointes à une réponse.
- Rechercher des tickets ou informations via un moteur de recherche global.

- **Agent :**

- Voir tous les tickets attribués à lui-même.
- Répondre aux tickets .
- Ajouter des notes internes.
- Changer l'état d'un ticket : Nouveau, Ouvert, Fermé, En attente, etc.

- Suivre ses performances via le tableau de bord.

- **Manager :**

- Assigner les tickets manuellement aux agents.
- Accès aux statistiques et rapport.
- Ajouter des notes internes.
- Superviser la charge de travail par agent/groupe.
- Définir des règles de priorité.

- **Administrateur :**

- Accès complet à toutes les fonctionnalités de l'application.
- Gestion des rôles et des utilisateurs.
- Configuration des canaux de communication.
- Sauvegarde et restauration.
- Définir des règles de priorité.

1.3.1.3 Points forts et points faibles

Points forts	Points faibles
<ul style="list-style-type: none">— Facile à utiliser— Open source— Multicanal (e-mail, chat, réseaux sociaux, etc.)— Interface moderne et intuitive— Statistiques et rapports avancés	<ul style="list-style-type: none">— Absence de gestion avancée de base de connaissances— Intégrations tierces limitées— Moins adapté aux très grandes structures— Version cloud payante

TABLEAU 1.1 : Points forts et points faibles de l'application Zammad

1.3.2 Étude de l'application Erex

Adresse (URL) : <https://erxes.io/>

1.3.2.1 Description

Fournir un environnement unifié pour recevoir, traiter et suivre les demandes de support ou incidents via un système de tickets.

- **Interface du site :** La figure représente l'interface d'accueil du site Erxes.

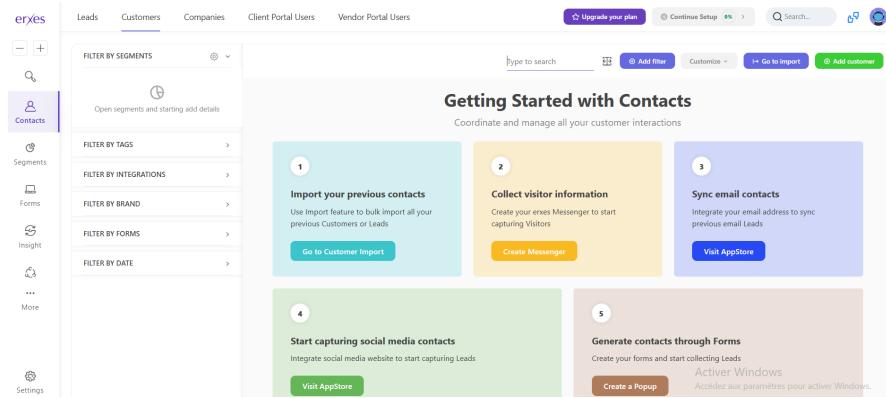


FIGURE 1.4 : interface Erxes

1.3.2.2 Étude fonctionnelle

Les principales fonctionnalités proposées par l'application, tout en les reliant aux acteurs qui en bénéficient.

- **Client :**

- Envoyer un message via le chat intégré ou par email.
- Poser des questions à l'équipe.
- Consulter la base de connaissances si activée.

- **Agent :**

- Créer des tickets et les suivre dans le module support.
- Accès au module "Inbox" (boîte de réception).
- Accès au module CRM (gérer les contacts et clients potentiels).
- Gérer les formulaires et les audiences.

- **Agent responsable d'équipe :**

- Toutes les fonctionnalités des agents.
- Valider ou corriger les contenus de la base de connaissances.
- Visualiser les performances (temps de réponse, taux de conversion).
- Accès aux statistiques avancées.

- **Administrateur :**

- Accès complet à tous les modules.
- Création et gestion des utilisateurs et rôles.
- Configuration des modules.
- Gestion des politiques de sécurité et permissions.
- Connexion des canaux.

1.3.2.3 Points forts et points faibles

Points forts	Points faibles
<ul style="list-style-type: none"> — Multicanal avec une interface unifiée — Sécurité (gestion des rôles,...) — Interface web moderne et responsive 	<ul style="list-style-type: none"> — Absence d'application mobile native — Moins d'intégrations avancées que les outils commerciaux — Dépendance à des compétences techniques pour l'installation serveur

TABLEAU 1.2 : Points forts et points faibles de l'application erex

1.4 Synthèse pour le choix de la solution

Après l'analyse de deux applications, nous avons conclu que l'application de la gestion des tickets et des workflows doit répondre aux besoins suivants :

- Utiliser le maximum possible d'interface simple et bien organiser pour faciliter l'expérience de Client.
- Assurer que le maximum des fonctionnalités seront disponible.
- Répartir l'application selon le système des rôles.
- Resoudre les problèmes de réclamation de maniere simple et efficace.
- Réaliser une application spécifique ciblant la gestion des tickets et des workflows pour une startup répondant à certains besoins fonctionnels.

1.5 Méthodologie adoptée

1.5.1 Méthodologie SCRUM

La méthodologie choisie pour ce projet est Scrum, une méthode Agile largement reconnue et adaptée à la gestion de projets itératifs. Scrum a été retenue en raison de sa flexibilité, de son approche collaborative et de sa capacité à s'adapter rapidement aux évolutions des besoins du projet. Cette méthode permet d'assurer une communication continue, une gestion optimale des priorités et une réactivité accrue aux défis rencontrés.

Scrum favorise également une meilleure visibilité sur l'avancement du projet et permet de détecter les problèmes dès qu'ils apparaissent.

L'équipe de projet est composée de :

Rôle	Nom
Scrum Master	Jalel Sawssen
Développeur	Ben hjal Eya
Product Owner	Mechergui Bassem

TABLEAU 1.3 : Les acteurs principaux dans SCRUM

Le Daily Scrum est une réunion brève qui se tient chaque jour pendant un sprint. Elle permet à chaque membre de l'équipe de partager rapidement les tâches accomplies et celles à venir. La réunion se déroule devant le Scrum Board, qui affiche l'état d'avancement du sprint. D'une durée maximale de 15 minutes, le Daily Scrum aborde également les obstacles rencontrés et les solutions envisagées. Comme illustré dans la Figure suivante, la méthodologie Scrum commence par la définition des différentes fonctionnalités du produit dans le backlog de produits. Ensuite, elle se poursuit avec la planification du sprint pour établir le plan détaillé de l'itération.

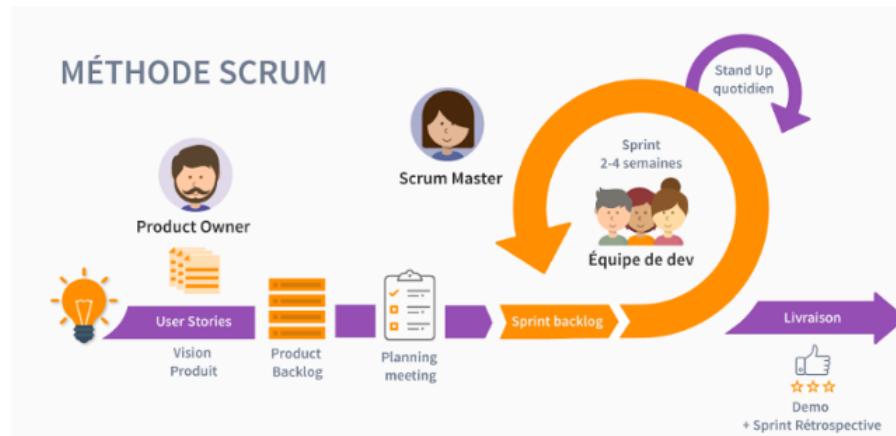


FIGURE 1.5 : Schéma d'Agile Scrum

Pendant le sprint, les membres de l'équipe tiennent des réunions quotidiennes pour discuter

de l'avancement des tâches en cours, des difficultés rencontrées et des tâches à venir. Une fois qu'une partie du produit est prête, une vérification de la conformité du produit fini est effectuée. En résumé, Scrum est une approche de gestion de projet agile qui se caractérise par des itérations courtes, une forte collaboration avec le client, une transparence totale, et une capacité d'adaptation continue pour répondre aux changements du marché et aux besoins évolutifs des utilisateurs.

1.5.2 Méthode de modélisation :UML

Le langage de modélisation unifié (Unified Modeling Language), est un langage d'analyse, de modélisation et de conception orienté objet. C'est une boîte à outils qui permet d'améliorer progressivement une méthode de travail, tout en préservant un mode de fonctionnement. Le support logiciel choisi pour la conception et la modélisation UML est Enterprise architecte, jugé le support le plus complet avec tous les éléments, les relations et les diagrammes UML.



FIGURE 1.6 : Logo de système de modélisation

1.6 Environnement de développement

1.6.1 Environnement matériel

Pour développer ce logiciel, nous avons utilisé comme environnement matériel un ordinateur portable qui possède comme caractéristiques :

TABLEAU 1.4 : Environnement matériel

Caractéristiques	Eya - ordinateur
Processeur	ASUS i5-6006U CPU @ 3.00GHz
Mémoire installée	12.00 Go
Type de système	Windows 11 64 bits

1.6.2 Environnement logiciel

Dans cette section, nous détaillons les logiciels et technologies utilisés lors de la réalisation du projet, en mettant l'accent sur les choix techniques qui ont permis de garantir l'efficacité, la fiabilité et la performance de l'application.

1.6.2.1 Outils et frameworks :

Outils :

- **Visual Studio** : Visual Studio Code est éditeur de texte open source, gratuit et multiplateforme (Windows, Mac et Linux), développé par Microsoft. Principalement conçu pour le développement d'application avec JavaScript, TypeScript et Node.js, l'éditeur peut s'adapter à d'autres types de langages grâce à un système d'extension bien fourni.



FIGURE 1.7 : logo vs code

- **MongoDB** : MongoDB est un programme de base de données multiplateforme orienté document et disponible en source ouverte. Classé comme un programme de base de données NoSQL, MongoDB utilise des documents de type JSON avec des schémas facultatifs. MongoDB est développé par MongoDB.



FIGURE 1.8 : Logo mongodb

- **MongoDB Compass** : MongoDB Compass est l'interface graphique officielle (GUI) de MongoDB,

développée par MongoDB Inc. Elle permet aux développeurs de visualiser, interagir et analyser leurs données stockées dans MongoDB sans avoir besoin d'utiliser la ligne de commande.

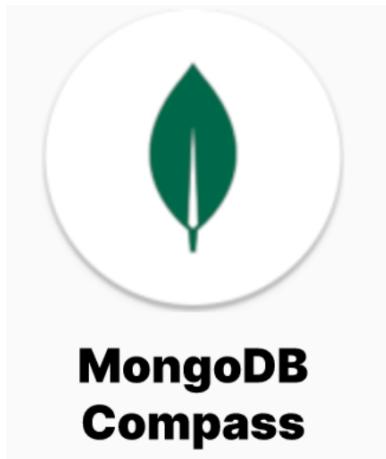


FIGURE 1.9 : Logo mongodb compass

- **Trello** Trello est un outil de gestion de projet en ligne, inspiré de la méthode Kanban de Toyota. Il permet de gérer les tâches et les workflows de manière visuelle et collaborative, facilitant le suivi de l'avancement du projet.



FIGURE 1.10 : Logo trello

- **Github** GitHub est une plateforme de développement logiciel basée sur Git, un système de gestion de version décentralisé, qui permet aux développeurs de collaborer sur des projets de code source.



FIGURE 1.11 : Logo github

- **Draw.io** Draw.io est un outil en ligne utilisé pour la création de diagrammes, tels que des diagrammes de processus, des cartes mentales et des diagrammes UML. Il facilite la modélisation visuelle des idées et des architectures de projets.



FIGURE 1.12 : Logo draw.io

- **Docker** Docker est une plateforme de conteneurisation qui permet d'assurer la cohérence des environnements de développement en isolant les applications dans des conteneurs. Il garantit ainsi une compatibilité entre les environnements de développement, de test et de production.



FIGURE 1.13 : Logo docker

- **Postman** Postman est une solution pour interroger ou tester des webservices et API. Il permet de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en Collections. Nous avons utilisé cet outil pour tester le fonctionnement de la partie Backend.[1]



FIGURE 1.14 : Logo postman

Vite : Vite est un outil de développement rapide (build tool) conçu pour les projets front-end modernes. Il a été utilisé dans ce projet pour gérer le bundling, le hot-reloading, et le serveur de développement local.



FIGURE 1.15 : Logo vite

Frameworks et bibliothèques :

- **Node.js** : Node.js est un environnement d'exécution JavaScript back-end open source, multiplateforme qui s'exécute sur le moteur V8 et exécute le code JavaScript en dehors d'un navigateur Web.



FIGURE 1.16 : Logo node js

- **Express.js** : Express.js, ou simplement Express, est un framework d'application Web back-end pour Node.js, publié en tant que logiciel gratuit et open source sous la licence MIT. Il est conçu pour créer des applications Web et des API. Il a été appelé le framework de serveur standard de facto pour Node.js.



FIGURE 1.17 : Logo express js

- **React** : React est une bibliothèque JavaScript frontale open source permettant de créer des interfaces utilisateur ou des composants d'interface utilisateur. Il est maintenu par Facebook et une communauté de développeurs individuels et d'entreprises. React peut être utilisé comme base dans le développement d'applications monopages ou mobiles.

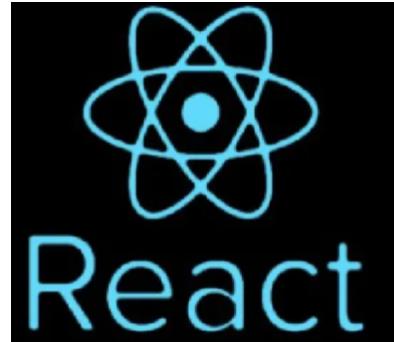


FIGURE 1.18 : Logo react

- **Tailwind CSS** : Tailwind est un framework CSS "utility-first" qui fournit un catalogue complet de classes et d'outils CSS qui vous permet de commencer facilement à styliser votre site Web ou votre application.



FIGURE 1.19 : Logo tailwind css

- **Socket.IO** : Socket.IO est une bibliothèque JavaScript pour les applications Web en temps réel. Il permet une communication bidirectionnelle en temps réel entre les clients Web et les serveurs. Il se comporte en deux parties : une bibliothèque côté client qui s'exécute dans le navigateur et une bibliothèque côté serveur pour Node.js.



Socket.IO

FIGURE 1.20 : logo socket.io

- **JSON Web Token** : JSON Web Token est une norme Internet proposée pour la création de données avec une signature facultative et un cryptage facultatif dont la charge utile contient JSON qui revendique un certain nombre de revendications. Les jetons sont signés à l'aide d'un secret privé ou d'une clé publique ou privée.

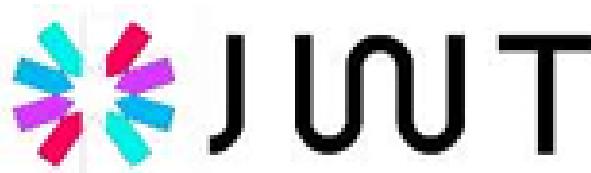


FIGURE 1.21 : Logo jwt

Conclusion

Dans ce chapitre, nous avons présenté le projet dans son cadre général, en introduisant la startup Treetronix, le contexte et les objectifs du projet, la méthodologie de travail adoptée pour la réalisation de l'application, ainsi que notre environnement de développement.

Nous passons maintenant à la spécification des besoins.

ANALYSE ET SPÉCIFICATION DES BESOINS

Plan

1	Spécifications des Besoins	20
2	Diagramme de structure des données global	24
3	Planification du backlog du produit	24
4	Planification de travail	32
5	Architecture de l'application	32
6	Difficultés rencontrées	35

Introduction

Dans ce chapitre, nous analysons les besoins fonctionnels et non fonctionnels de notre application, en identifiant les différents acteurs du système. Cette analyse est appuyée par deux éléments essentiels de modélisation le diagramme de cas d'utilisation, qui illustre les interactions entre les acteurs et le système, et un diagramme de structure des données, adapté au modèle NoSQL de MongoDB. Ce dernier remplace le diagramme de classes traditionnellement utilisé dans les bases de données relationnelles. De plus, nous aborderons la planification du projet

2.1 Spécifications des Besoins

2.1.1 Les Besoins fonctionnels

Le futur système permet à :

- «L'ADMINISTRATEUR » de :
 - S'inscrire et se connecter à son espace personnel
 - Gestion des Utilisateurs par Département :créer, modifier, supprimer (avec confirmation), lister, consulter les détails et rechercher (par filtre nom) les utilisateurs.
 - Gérer les départements : Ajouter, modifier, lister, rechercher et supprimer des départements.
 - Gérer les catégories associées aux départements : créer, la liste des catégories.
 - Gérer les tickets :
 - Création de tickets associés à un département et une catégorie
 - Modification des tickets existants
 - Affichage de la liste des tickets avec son status (nouveaux, en cours, résolu)
 - Suppression de tickets
 - Assignation des tickets :
 - Attribution aux agents par département
 - Notification automatique par email des nouvelles assignations
 - Résolution de ticket :
 - Passage à l'état "Résolu"
 - Notification automatique au client par email
 - Gérer les commentaires : ajouter, modifier ,supprimer et consulter des commentaires sur les tickets pour faciliter le suivi et la collaboration.

- Visualiser et de déplacer les tickets entre les colonnes Kanban (Nouveaux, En cours, Résolus) par glisser-déposer(drag and drop) pour gérer leur cycle de vie.
- Suivre l'activité globale : nombre de tickets ouverts, résolus, temps moyen de résolution, agents actifs accéder à l'historique et à la traçabilité de tous les fonctionnalités.
- L'utilisateur « AGENT » de :
 - S'inscrire et se connecter à son espace personnel
 - Lister les tickets assignés
 - Consulter le détail d'un ticket : description, catégorie, historique, pièces jointes..
 - Modifier le statut du ticket selon son avancement.
 - Ajouter des notes internes
 - Indiquer et suivre le temps passé sur chaque ticket.
 - Clôturer un ticket lorsqu'il est résolu.
- Assistance par chatbot :
 - Interaction avec un chatbot intelligent
 - Résolution guidée des problèmes selon la catégorie du ticket.
- Rechercher des tickets dans l'historique (nouveaux,ancienne..).
- Gérer son profil (competance, bio..).
- Visualiser et de déplacer les tickets entre les colonnes Kanban (Nouveaux, En cours, Résolus) par glisser-déposer pour gérer leur cycle de vie.
- Envoi automatique d'une notification email à l'agent lorsqu'un ticket lui est assigné.
- Gérer les commentaires : ajouter, modifier ,supprimer et consulter des commentaires sur les tickets pour faciliter le suivi et la collaboration.
- L'utilisateur « CLIENT » de :
 - Se connecter à son espace personnel
 - Créer un nouveau ticket
 - Suivre l'état de ses tickets (en attente, en cours, résolu)
 - Voir le détail d'un ticket si celui-ci a été complété par l'agent ou non.
 - Commentaires sur tickets :ajouter, modifier, répondre, supprimer.
 - Envoi automatique d'une notification par email Lorsque le ticket passe à l'état "résolu" .
 - Rechercher des tickets dans l'historique.

- Gérer son profil (nom, email, photo de profil, mot de passe,...).
- Assistance par chatbot :
 - Interaction avec un chatbot intelligent
 - Résolution guidée des problèmes selon la catégorie du ticket

2.1.2 Les besoins non fonctionnels

Les besoins non fonctionnels doivent assurer :

- **Ergonomie** : Nous nous engageons à développer une interface utilisateur conviviale et ergonomique, facile à naviguer pour tous, quel que soit leur niveau de compétence technique
- **Sécurité** : Assurer une authentification sécurisée et la protection des données sensibles.
- **Performance** : Garantir une réponse rapide des systèmes, notamment pour la détection en temps réel.
- **Fiabilité** : assurer un fonctionnement continu sans interruption.
- **Utilisabilité** : Offrir une interface intuitive, facilitant l'accès aux fonctionnalités pour tous les utilisateurs.
- **Scalabilité** : Permettre l'intégration de nouveaux utilisateurs et fonctionnalités sans compromettre la performance.

2.1.3 Description des acteurs

Dans ce tableau, nous identifions les acteurs principaux du système. Pour ce faire, il faut discerner quels sont les différents rôles joués par les utilisateurs.

TABLEAU 2.1 : Acteurs du système

Acteurs	Description du rôle
client	Un utilisateur qui peut faire une réclamation par un ticket.
Administrateur	Le responsable de l'application et de l'interaction entre les autres acteurs.
Agent	Un utilisateur qui peut entrer dans la plateforme et résoudre les problèmes liés aux réclamations.

2.1.3.1 Diagramme de cas d'utilisation global

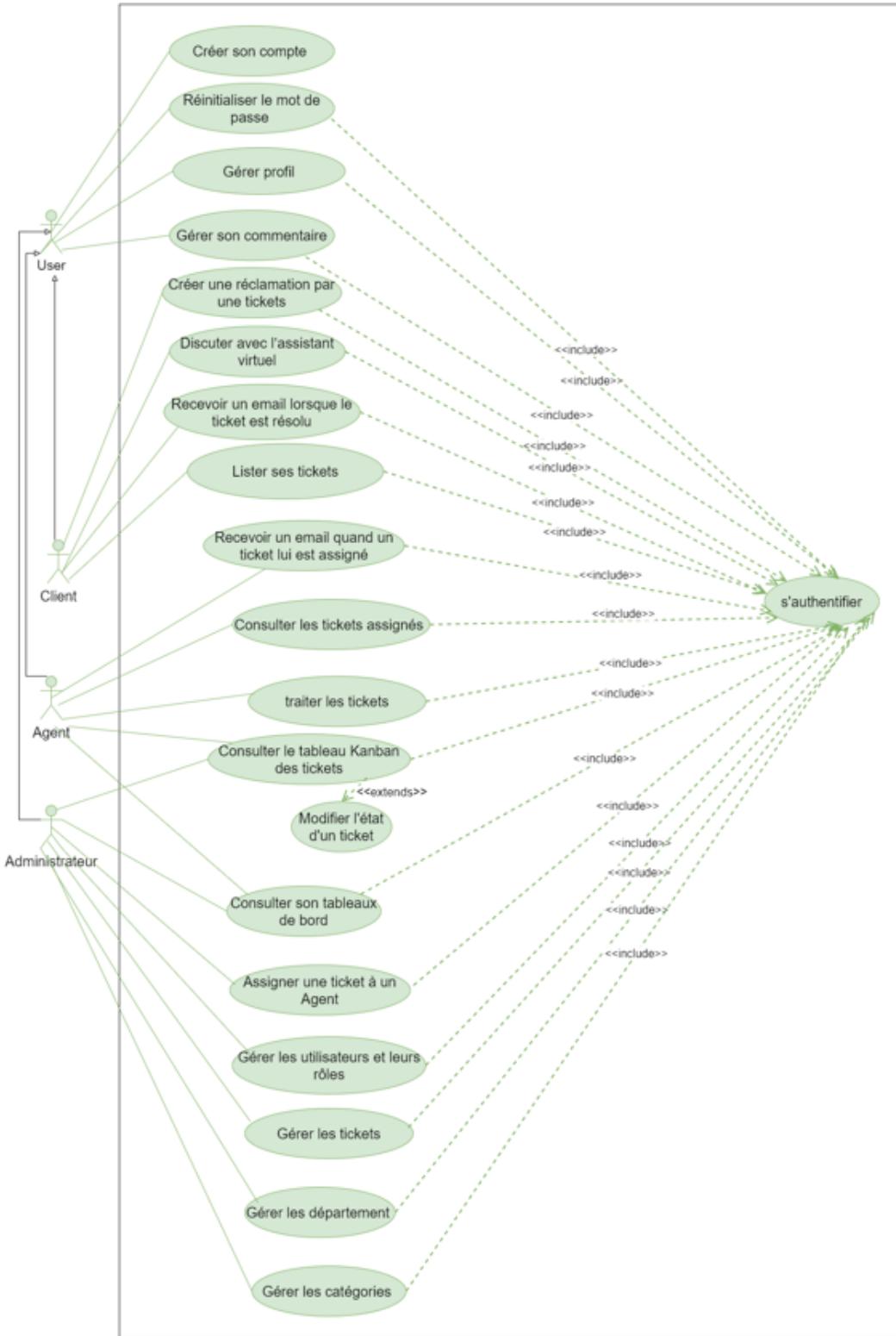


FIGURE 2.1 : Diagramme de cas d'utilisation global

2.2 Diagramme de structure des données global

Ci-dessous dans la figure suivante , nous représentons le Diagramme de structure des données global de notre application.

Le diagramme suivant représente la structure de nos entités principales, leurs relations, et les références entre documents (MongoDB).

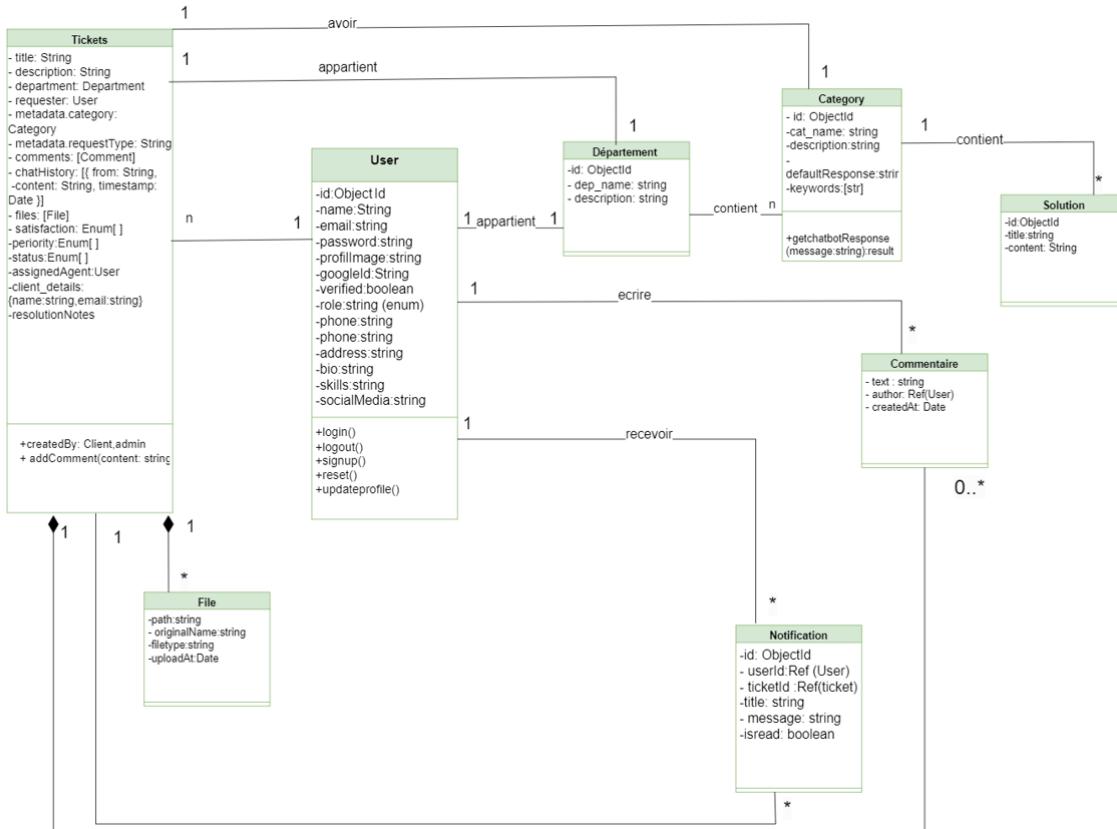


FIGURE 2.2 : Diagramme de structure des données global

2.3 Planification du backlog du produit

TABLEAU 2.2 : Backlog de produit

ID	Fonctionnalité	Sous-ID	User Story	Priorité	Complexité
A	Inscription et Authentification	A.1	En tant qu'utilisateur, je veux pouvoir m'inscrire avec une validation par email.	Haute	3
		A.2	En tant qu'utilisateur, je veux m'authentifier et accéder au application.	Haute	2
		A.3	En tant qu'utilisateur, je veux pouvoir réinitialiser mon mot de passe en cas d'oubli.	Moyenne	2
		A.4	En tant qu'utilisateur, je veux recevoir un lien pour récupérer mon mot de passe oublié.	Moyenne	1
		A.5	En tant qu'utilisateur, je veux pouvoir me déconnecter de mon compte.	Faible	1
B	Gestion des Départements	B.1	En tant qu'administrateur, je veux créer des départements.	Haute	2
		B.2	En tant qu'administrateur, je veux modifier des départements.	Moyenne	2
		B.3	En tant qu'administrateur, je veux supprimer des départements.	Moyenne	2
		B.4	En tant qu'administrateur, je peux voir la liste des départements.	Moyenne	2
		B.5	En tant qu'administrateur, je veux pouvoir rechercher un département dans la liste en utilisant un filtre par nom.	Moyenne	1

ID	Fonctionnalité	Sous-ID	User Story	Priorité	Complexité
C	Gestion des Utilisateurs	C.1	En tant qu'administrateur, je veux voir détails d'un utilisateur.	Haute	2
		C.2	En tant qu'administrateur je peux ajouter utilisateurs	Moyenne	2
		C.3	En tant qu'administrateur je peux supprimer un utilisateur.	Moyenne	2
		C.4	En tant qu'administrateur, je veux pouvoir lister les utilisateurs existants.	Moyenne	2
		C.5	En tant qu'administrateur, je veux pouvoir modifier le rôle ou les données d'un utilisateur existant.	Haute	3
		C.6	En tant qu'administrateur, je peux filtrer les utilisateurs par nom pour faciliter leur recherche. .	Moyenne	1
D	Gestion des Tickets	D.1	En tant qu'client, je veux créer un ticket afin soumettre une réclamation.	Haute	3
		D.2	En tant qu'agent, je souhaite consulter la liste des tickets qui me sont assignés.	Moyenne	2
		D.3	En tant qu' agent, je veux consulter les détails d'un ticket et mettre à jour son statut, afin de marquer le ticket comme résolu et déclencher l'envoi d'un e-mail au client.	Haute	3

ID	Fonctionnalité	Sous-ID	User Story	Priorité	Complexité
		D.4	En tant qu'client, je veux suivre l'état de mon ticket et sa détails.	Moyenne	2
		D.5	En tant que client, je veux pouvoir ajouter mon niveau de satisfaction (satisfait, très satisfait, insatisfait..) dans les détails d'un ticket, une fois que le ticket a été résolu.	Moyenne	2
		D.6	En tant qu'client, je veux voir la listes du mes tickets.	Moyenne	2
		D.7	En tant qu'administrateur je peux créer un tickets.	Haute	4
		D.8	En tant qu' administrateur, je veux assigner un ticket à un agent en fonction de son département et de sa disponibilité, afin que un e-mail automatique soit envoyé à l'agent concerné.	Moyenne	4
		D.9	En tant qu'administrateur je veux modifier un tickets	Moyenne	3
		D.10	En tant qu'administrateur je peux supprimer un tickets.	Moyenne	2
E	Kanban	E.1	En tant qu'administrateur, je veux visualiser le cycle de vie de tickets via un tableau de kanban afin de suivre leur progression dans les différentes étapes.	haute	3

ID	Fonctionnalité	Sous-ID	User Story	Priorité	Complexité
		E.2	En tant qu' administrateur, je veux pouvoir glisser et déposer les tickets dans le tableau Kanban, afin de réorganiser facilement leur position ou changer leur statut (nouveau, en cours, résolu).	Moyenne	3
		E.3	En tant qu'agent, je veux visualiser le cycle de vie de tickets via un tableau de kanban.	Moyenne	3
		E.4	En tant qu' agent, je veux pouvoir glisser-déposer les tickets dans le tableau Kanban afin de réorganiser leur position ou changer leur statut (nouveau, en cours, résolu), et également modifier les données d'un ticket.	Moyenne	3
F	Notifications	F.1	En tant qu'agent, je veux recevoir une notification par un email après l'assignation d'un ticket par l'administrateur, contenant les détails du ticket.	Moyenne	3
		F.2	En tant qu'client, je veux être notifié par email lors des changements de statut à résolu.	Moyenne	3

ID	Fonctionnalité	Sous-ID	User Story	Priorité	Complexité
G	Tableau de bord	G.1	En tant qu'administrateur, je veux visualiser des informations globales sur les fonctionnalités (comme le nombre total de tickets, de tickets résolus,...)	Haute	3
		G.2	En tant qu'agent, je veux avoir accès à un tableau de bord personnalisé pour visualiser les tickets qui me sont assignés, les tickets résolus,..	Haute	3
H	Gestion des catégories	H.1	En tant qu'administrateur, je veux créer des catégories.	Moyenne	2
		H.2	En tant qu'administrateur, je veux modifier des catégories.	Moyenne	2
		H.3	En tant qu'administrateur, je veux supprimer des categories.	Moyenne	2
		H.4	En tant qu'administrateur, je veux voir la liste des categories.	Moyenne	2
		H.5	En tant qu'administrateur, je peux filtrer les catégories pour faciliter leur recherche.	Moyenne	2
I	Gestion des commentaires	I.1	En tant qu'utilisateur, je veux écrire un commentaire pour mon ticket.	Moyenne	3
		I.2	En tant qu'utilisateur, je veux modifier un commentaire pour mon ticket.	Moyenne	2

ID	Fonctionnalité	Sous-ID	User Story	Priorité	Complexité
		I.3	En tant qu'utilisateur, je veux supprimer un commentaire pour mon ticket.	Moyenne	2
		I.4	En tant qu'utilisateur, je veux voir la liste des commentaire pour chaque ticket.	Moyenne	2
J	Gestion des Profiles	J.1	En tant qu'agent, je veux ajouter mes compétences, bio, LinkedIn, twitter...	Moyenne	3
		J.2	En tant qu'agent, je veux modifier mes informations sur mon profil.	Moyenne	2
		J.3	En tant qu'agent, je veux supprimer mes informations.	Moyenne	2
		J.4	En tant qu'client, je veux modifier mes informations.	Moyenne	2
		J.5	En tant qu'admin, je veux voir liste complète des profils utilisateurs (bio, compétences, etc.).	Moyenne	2
		J.6	En tant qu'admin , je veux supprimer les informations de profil d'un utilisateur.	Moyenne	2
		J.7	En tant qu'admin, je veux les champs du profil d'un utilisateur.	Moyenne	2
		J.8	En tant qu'admin, je veux voir les détails complets d'un profil utilisateur.	Moyenne	2

ID	Fonctionnalité	Sous-ID	User Story	Priorité	Complexité
		J.9	En tant qu'agent, je veux modifier ou ajouter mon statut de disponibilité (Absent, Occupé, Disponible, Hors ligne) et définir ma capacité maximale de tickets.	Moyenne	2
K	Chatbot Intelligent	K.1	En tant qu'client, je veux interagir avec un chatbot qui analyse mes messages .	Moyenne	3
		K.2	En tant qu'client, je veux interagir avec un chatbot pour recevoir des conseils pour resoudre mon problème.	Moyenne	2
L	Gestion des Horaires	L.1	En tant qu'agent, je veux ajouter mes horaires de disponibilité pour chaque jour de la semaine.	Moyenne	2
		L.2	En tant qu'agent, je veux modifier mes horaires de disponibilité existants pour tenir compte de tout changement dans mon emploi du temps.	Moyenne	2
		L.3	En tant qu'agent, je veux supprimer mes horaires de disponibilité afin de ne pas apparaître comme disponible quand je ne le suis pas.	Moyenne	2

2.4 Planification de travail

2.4.1 Répartition des releases

Un plan de release est une projection des sprints futurs, présentée sous forme de tableau.

Release ID	Contenu des Sprints
Release 1	<ul style="list-style-type: none">• Sprint 1 : Inscription, Authentification• Sprint 2 : Gestion des départements et Gestion des catégories
Release 2	<ul style="list-style-type: none">• Sprint 3 : Gestion des utilisateurs et des rôles• Sprint 4 :Gestion du profil et des horaires• Sprint 5 : Gestion Tickets côté client/agent + Notifications• Sprint 6 : Gestion Tickets côté Admin + Commentaires
Release 3	<ul style="list-style-type: none">• Sprint 7 : Chatbot et vue Kanban• Sprint 8 : Tableau de bord

TABLEAU 2.3 : Répartition des releases et des sprints selon les dépendances fonctionnelles

2.5 Architecture de l'application

2.5.1 Architecture logique

L'architecture logique de notre application est divisée en deux parties, une partie Back-end et une partie Front-end.

La partie Front-end est développée avec React.js, une bibliothèque JavaScript modulaire pour la création d'interfaces utilisateur. L'interface est construite à partir de composants fonctionnels ou de classe. Chaque composant gère son propre état et interagit avec des services d'abstraction pour la logique métier.

Ces services utilisent des appels HTTP pour communiquer avec le Back-end. Les échanges de

données entre le Back-end et le Front-end se font au format JSON.

La partie Back-end repose sur Node.js, un environnement JavaScript côté serveur, et Express.js, un framework léger et flexible .

Les contrôleurs reçoivent les requêtes HTTP pour appliquer la logique métier, et interagissent avec une base de données.

2.5.1.1 Architecture logique nodejs

La figure représente l'architecture logique de nodejs

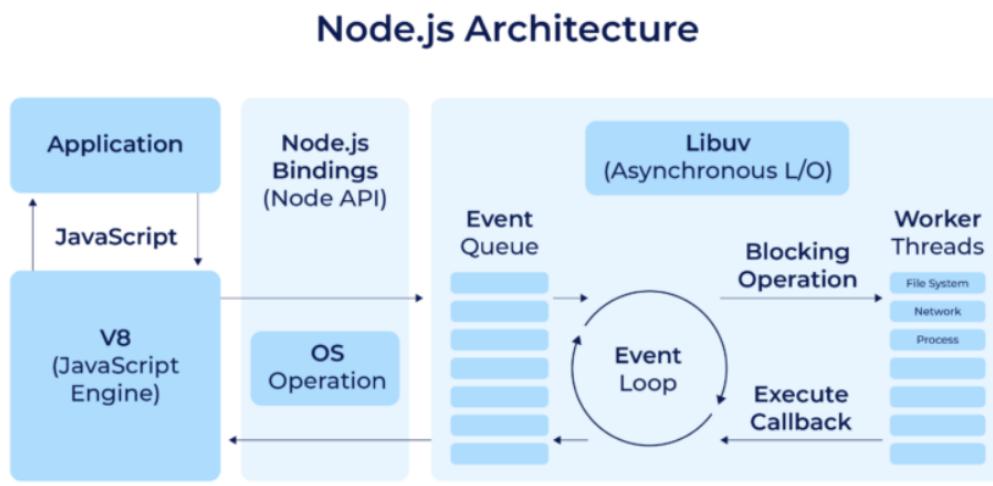


FIGURE 2.3 : Architecture logique Node.js

2.5.1.2 Architecture logique de react

Une application React moderne s'organise généralement autour des éléments clés suivants :

Composants : Ce sont les briques de base de l'interface, divisés en composants "intelligents" (gérant la logique et l'état) et "présentationnels" (affichage uniquement).

Hooks : Ils encapsulent la logique métier réutilisable (comme les services dans Angular) et gèrent l'état local.

Gestion d'état global (Redux, Context API) : Pour partager des données entre composants éloignés, similaire aux services injectables d'Angular.

Couche API : Axios ou Fetch pour communiquer avec le backend, centralisée dans des fichiers dédiés.

Routing (React Router) : Gère la navigation entre les vues.

La figure 2.4 représente l'architecture logique d'react :

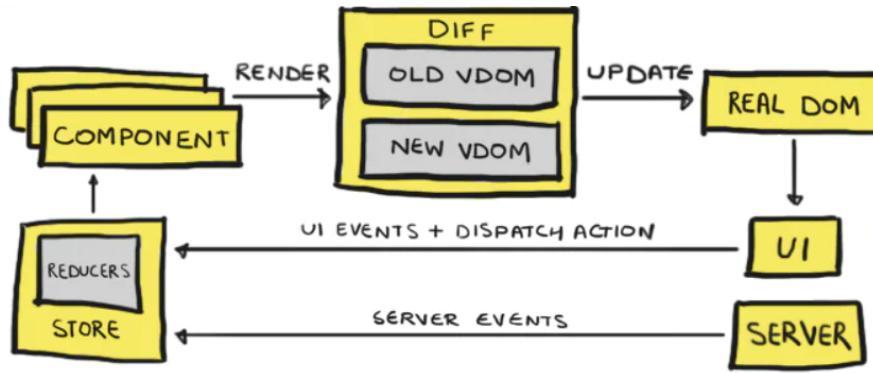


FIGURE 2.4 : l'architecture logique de react

2.5.2 Architecture physique

Notre application repose sur une architecture 3-tiers. La figure , représente les interactions entre les différents niveaux, ainsi que l'architecture physique que nous avons adopté

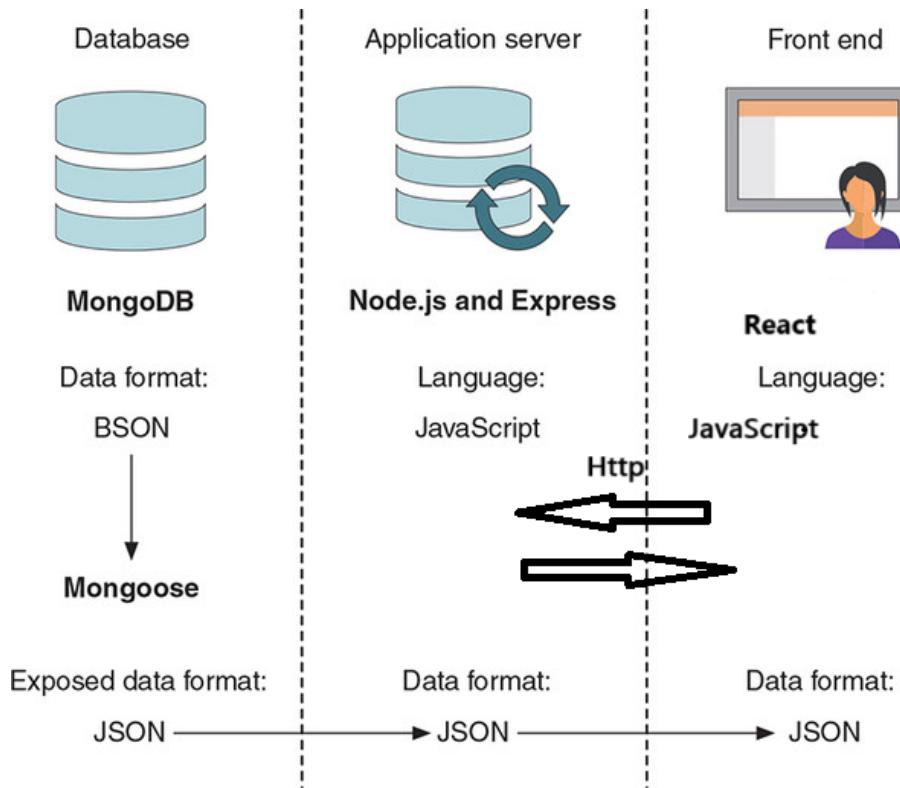


FIGURE 2.5 : Architecteur physique

2.5.3 Architecture de API REST

REST utilise la structure des requêtes HTTP et pour communiquer avec une API REST, on peut seulement utiliser les termes GET, POST, PUT et DELETE. Pour chaque donnée, il y a quatre types d'opérations possibles (CRUD) auxquels correspondent ces quatre mots clés :

- Créer – POST
- Afficher - GET

- Mettre à jour - PUT
- Supprimer - DELETE

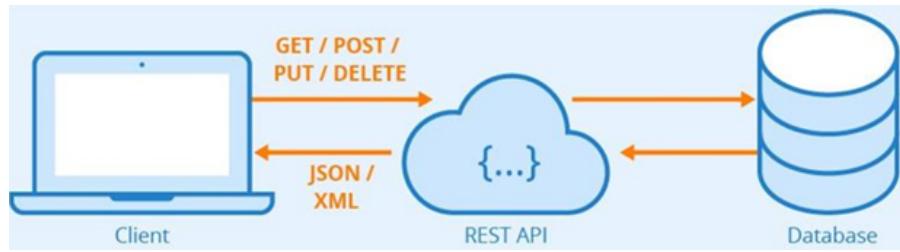


FIGURE 2.6 : Architecture de API REST

2.6 Difficultés rencontrées

La principale difficulté rencontrée est que je travaille seul sur le projet, sans équipe de développement et sans maquettes préétablies. J'ai dû tout développer à partir de zéro, en assurant à la fois la partie Back-end avec Node.js et la partie Front-end avec React. Cela comprend la conception de l'interface, le développement des fonctionnalités, l'architecture du code ainsi que l'intégration complète entre le Front-end et le Back-end. Ce travail en autonomie demande beaucoup de temps et de rigueur, surtout en l'absence de base existante ou de support visuel initial.

Conclusion

La phase de spécification des besoins fonctionnel et non fonctionnel est une phase très importante dans le cycle de vie d'un projet, car elle offre une vue plus claire du système ,aussi le diagramme de cas d'utilisation globale et le diagramme de classe. Par conséquent, nous avons introduit le backlog produit et la planification des releases.

Dans le prochaine chapitre, nous nous concentrerons sur la conception et le développement de la première Release.

RELEASE 1 : ÉTUDE ET RÉALISATION DE LA RELEASE 1

Plan

1	sprint 1 :Module inscription et authentification	37
2	Sprint 2 :Module gestion des départements et gestion des catégories	45

Introduction

Après avoir spécifié et analysé les besoins, nous passons à la conception et à la réalisation des différents modules de l'application. Dans ce chapitre, nous détaillons la conception et le développement. Les éléments relatifs aux premier et deuxième sprints du backlog produit sont les suivants :

Sprint 1 : Inscription et authentification .

Sprint 2 : Gestion des départements. et Gestion des catégories .

3.0.1 Objectifs du sprint 1

L'objectif du premier sprint est de développer le module «Inscription et Authentification » qui permet de faire l'inscription et l'authentification.

3.1 sprint 1 :Module inscription et authentification

3.1.1 Backlog du sprint 1

Le backlog du sprint 1 décrit les fonctionnalités à développer, organisées par user stories. Chaque user story est décomposée en tâches spécifiques, accompagnées de leur priorité et d'une estimation de l'effort requis.

TABLEAU 3.1 : Backlog de sprint 1

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
A	En tant qu'utilisateur, je veux pouvoir m'inscrire avec confirmation par e-mail.	A.1	Créer le modèle User, la route d'inscription, le controller de création de compte, la génération du token de confirmation et l'envoi de l'e-mail.	Haute	Moyenne
		A.2	Créer le formulaire d'inscription avec gestion de l'état de confirmation affichage des erreurs .		Complexe

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
		A.3	Afficher un message de succès après l'inscription en utilisant location.state pour passer les données.	Moyenne	Simple
B	En tant qu'utilisateur, je veux pouvoir m'authentifier pour accéder au système.	B.1	Créer route de login, controller de vérification des identifiants, génération et stockage du token, redirection selon le rôle.	Haute	Complexe
		B.2	Créer le formulaire de login avec validation des champs, gestion des erreurs d'identifiants.	Moyenne	Moyenne
C	En tant qu'utilisateur, je veux pouvoir me déconnecter de mon compte.	C.1	Ajouter un bouton de déconnexion visible dans l'interface utilisateur.	Moyenne	Simple
		C.2	Supprimer le token ou la session, et réinitialiser l'état du contexte utilisateur.	Moyenne	Simple
D	En tant qu'utilisateur, je veux réinitialiser mon mot de passe en cas d'oubli.	D.1	Créer la route de demande de mot de passe oublié, le controller pour générer un token sécurisé, et envoyer l'e-mail de récupération.	Moyenne	Moyenne
		D.2	Créer le formulaire de réinitialisation du mot de passe qui accepte un token valide, avec validation des nouveaux mots de passe.	Moyenne	Moyenne

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
		D.3	Afficher un message de succès ou d'échec après soumission du formulaire de réinitialisation.	Moyenne	Simple

3.1.2 Le diagramme de cas d'utilisation de sprint 1

Le diagramme de cas d'utilisation de notre premier Sprint 1 est présenté dans la figure suivante .

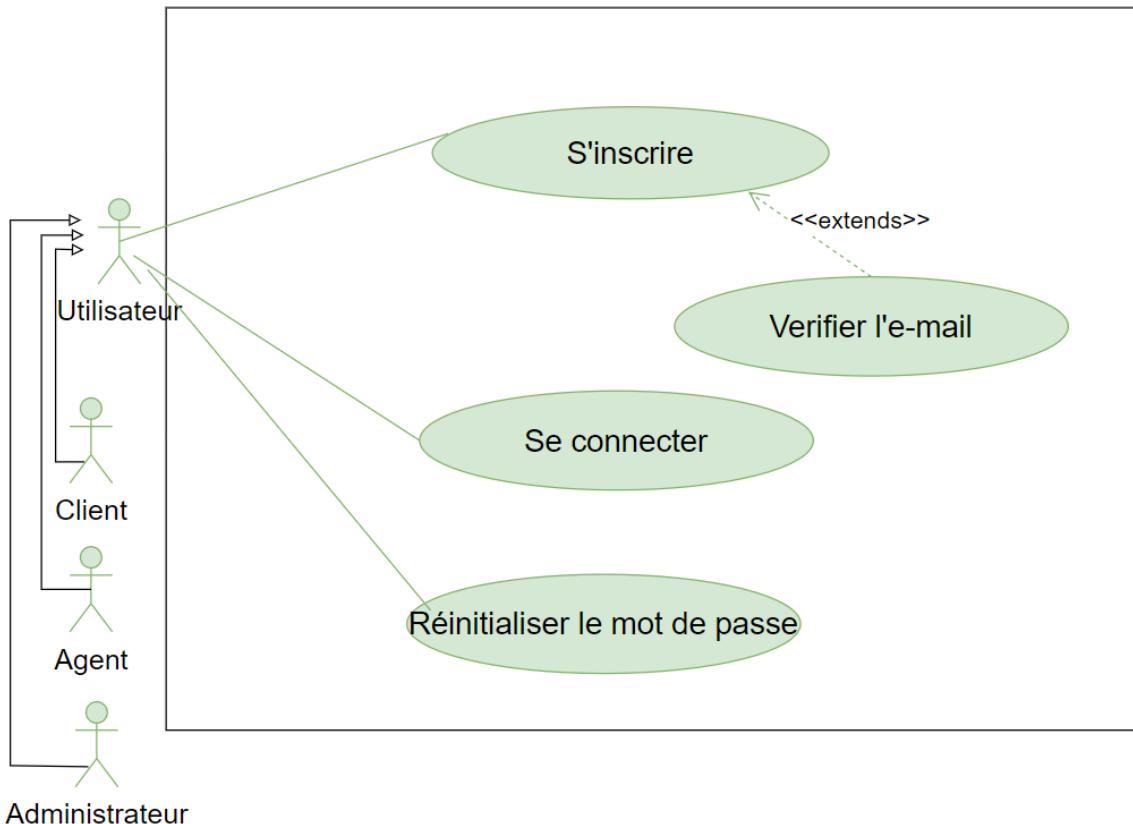


FIGURE 3.1 : Diagramme de cas d'utilisation de sprint 1

3.1.3 Diagramme de structure de données de sprint 1

La figure suivante représente le diagramme de classe du sprint "inscription et authentification"

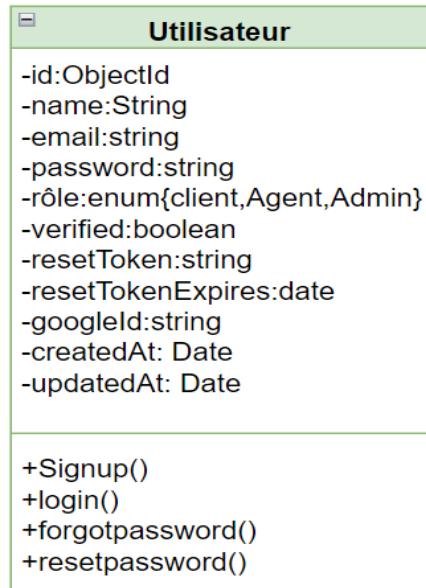


FIGURE 3.2 : Diagramme structure de données de sprint 1

3.1.4 Conception

3.1.4.1 Diagramme de séquence de cas d'utilisation "Inscription"

La figure illustre le diagramme de séquence du cas d'utilisation de l'inscription.

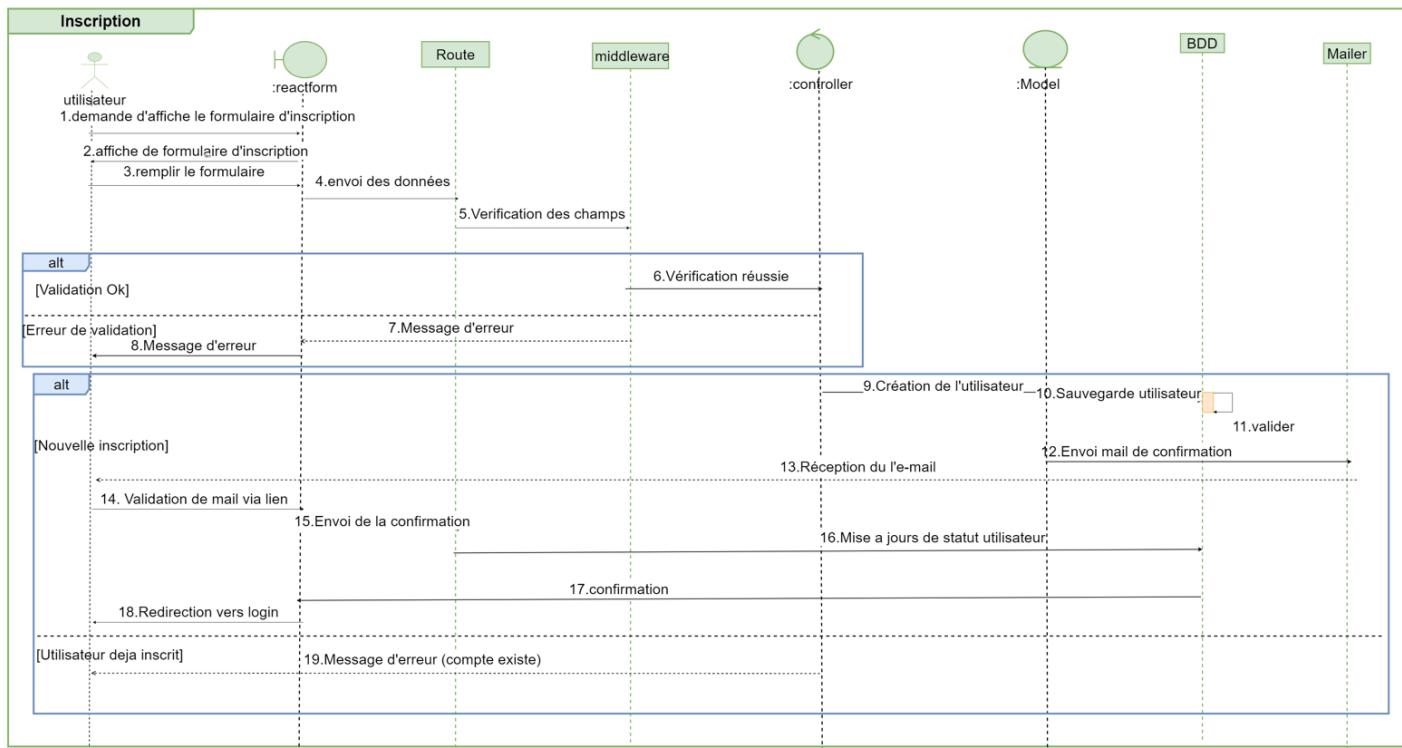
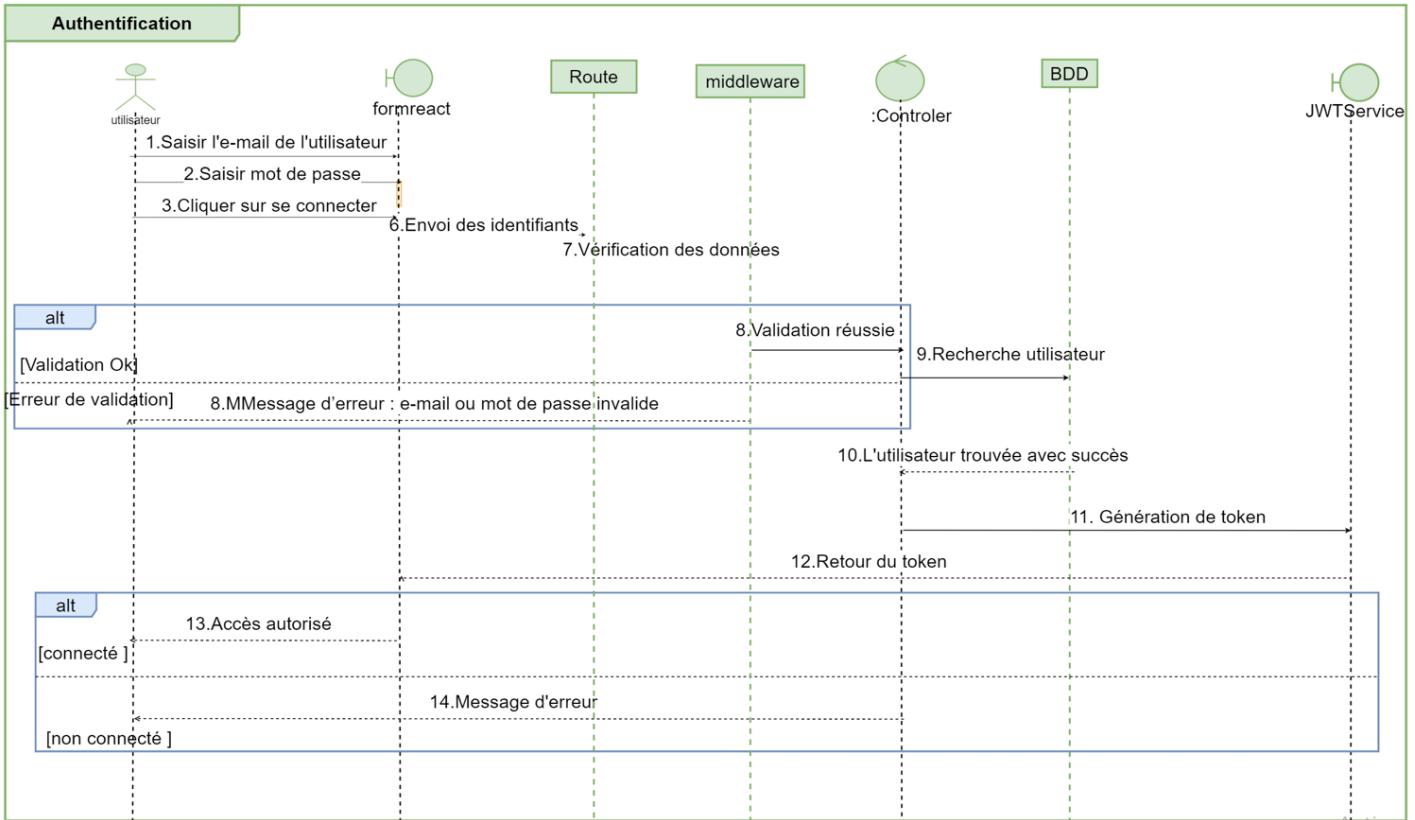


FIGURE 3.3 : Diagramme de séquence "Inscription"

3.1.4.2 Diagramme de séquence objet "Authentification"

La figure illustre le diagramme de séquence du cas d'utilisation Authentification.

**FIGURE 3.4 :** Diagramme de séquence "Authentification"

3.1.5 Réalisation

Pour mieux comprendre le fonctionnement de notre projet, nous allons présenter les interfaces de sprint 1 .

Voir les figures :

3.1.6 Interfaces d'inscription

L'interface suivante permet à un utilisateur de créer un compte. Par exemple, un nouveau compte peut être créé avec le nom 'agenteya', auquel le rôle 'client' est attribué automatiquement.

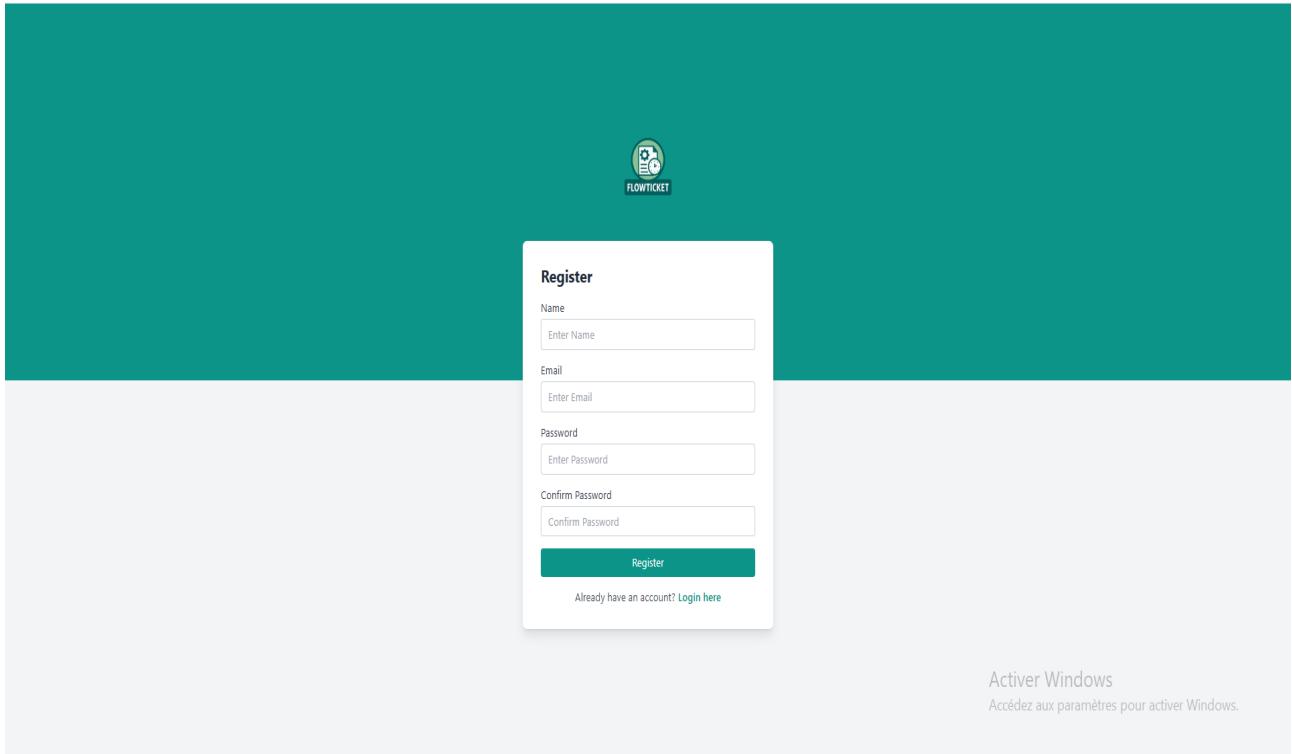


FIGURE 3.5 : Interfaces d'inscription

3.1.7 Interfaces de vérification de création compte

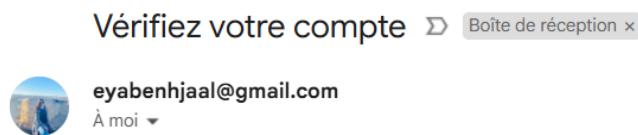


FIGURE 3.6 : Mail de vérification compte

3.1.8 Interface de confirmation du mot de passe oublié

La figure suivante présente l'interface où l'utilisateur peut saisir son e-mail pour recevoir un mail contenant un lien pour réinitialiser son mot de passe ici on a un message pour une liaison de reninitialisation envoyée à votre email.

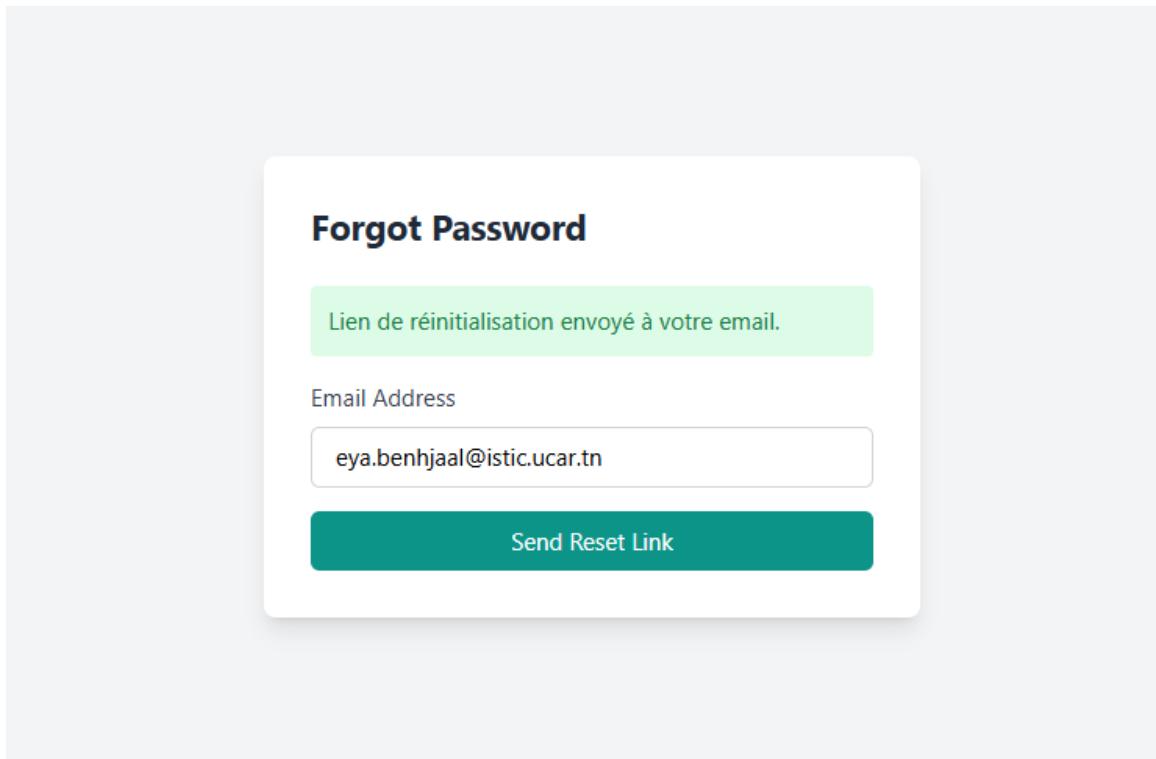


FIGURE 3.7 : Interface pour la réinitialisation du mot de passe oublié

Ceci est l'e-mail de réinitialisation. Lorsqu'on clique sur le lien, l'utilisateur est redirigé vers l'interface de réinitialisation du mot de passe.



FIGURE 3.8 : mail de réinitialisation

3.1.9 Interface de reset password

L'interface suivante permet à l'utilisateur de réinitialiser son mot de passe.

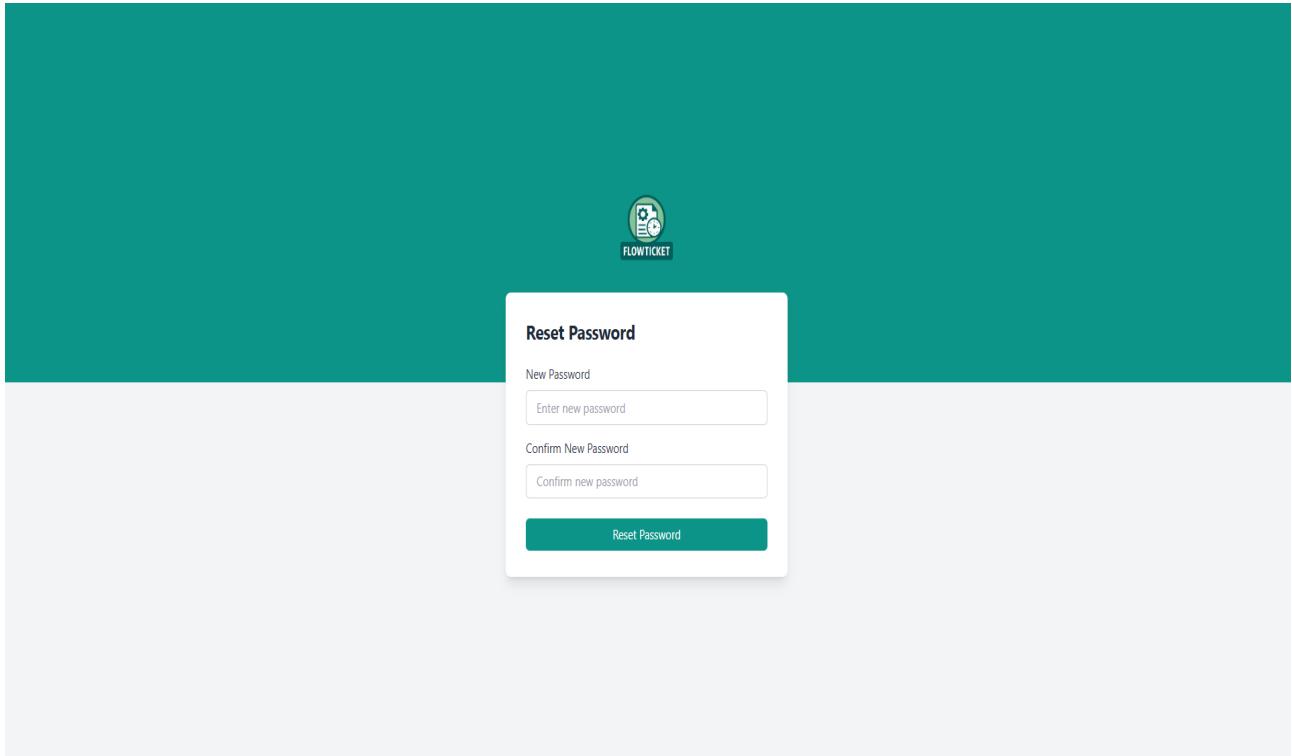


FIGURE 3.9 : Interface de reset password

3.1.10 Interface de login

L'interface suivante permet à l'utilisateur de se connecter à son espace c'est apres le register on a le rediger a elle ou apres renisialiser le mdp.

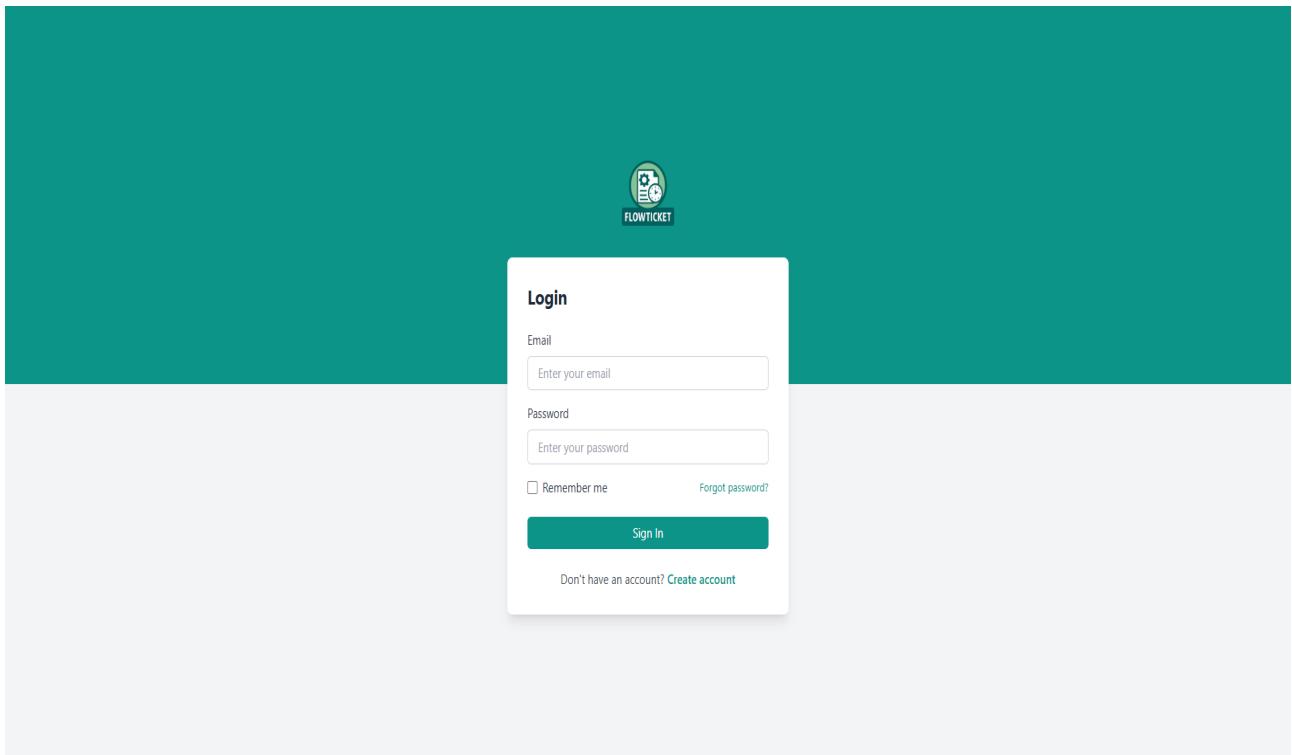


FIGURE 3.10 : Interface de login

3.2 Sprint 2 :Module gestion des départements et gestion des catégories

Cette section présente les étapes de réalisation du sprint 2 "Gestion des départements et des catégories", destiné à l'administrateur.

3.2.1 Objectifs du sprint 3

L'objectif du troisième sprint est de développer Le module "gestion des départements et gestion des catégories ". Ce module permet, d'une part, aux administrateurs de gérer les départements ainsi que les catégories en les associant à un département, et d'autre part, aux clients aux clients de choisir une catégorie lors de la création d'un ticket.

TABLEAU 3.2 : Backlog de sprint 2

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
A	En tant qu'administrateur, je veux créer des départements.	A.1	Créer le modèle du département	Haute	Moyenne
		A.2	Créer le contrôleur d'ajout	Haute	Moyenne
		A.3	Définir la route POST pour la création de department	Moyenne	Moyenne
		A.4	Créer le formulaire frontend d'ajout de département	Haute	Moyenne
		A.5	Gérer l'état local des champs (useState)	Moyenne	Moyenne
		A.6	Envoyer la requête POST avec Axios	Haute	Moyenne
B	En tant qu'administrateur, je veux modifier des départements.	B.1	Créer le contrôleur de modifier département	Haute	Moyenne
		B.2	Définir la route PUT pour la modification de department	Moyenne	Moyenne

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
		B.3	Définir la route GET pour récupérer les données de department	Moyenne	Moyenne
		B.4	Créer un formulaire d'édition côté frontend .	Moyenne	Moyenne
		B.5	Appeler l'API GET pour récupérer les données à modifier	Moyenne	Moyenne
		B.6	Mettre en place l'appel PUT pour éditer les champs	Haute	Moyenne
C	En tant qu'administrateur, je veux supprimer des départements.	C.1	Créer le contrôleur de suppression de département	Haute	Moyenne
		C.2	Définir la route DELETE pour la suppression	Haute	Moyenne
		C.3	Ajouter un bouton Delete côté frontend	Moyenne	Faible
		C.4	Appeler l'API DELETE depuis React	Haute	Moyenne
		C.5	Mettre à jour l'état après suppression	Moyenne	Moyenne
		C.6	Ajouter une confirmation avant suppression	Faible	Faible
D	En tant qu'administrateur, je peux voir la liste des départements.	D.1	Créer une API pour récupérer tous les départements	Moyenne	Moyenne
		D.2	Créer un composant React pour afficher la liste des départements	Moyenne	Moyenne

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
		D.3	Intégrer un tableau interactif avec ‘react-data-table-component’	Moyenne	Moyenne
		D.4	Utiliser ‘useEffect’ et ‘axios’ pour récupérer les données de l’API	Moyenne	Moyenne
		D.5	Gérer l’état de chargement (‘depLoading’) pendant la récupération des données	Moyenne	Faible
		D.6	Gérer les erreurs d’appel API côté client	Moyenne	Faible
		D.7	Ajouter un bouton pour créer un nouveau département	Moyenne	Faible
		D.8	Afficher les boutons modifier/supprimer pour chaque ligne	Moyenne	Faible
E	En tant qu’administrateur, je veux rechercher un département par nom.	E.1	Ajouter un champ de recherche dans le composant ‘DepartmentList’	Moyenne	Faible
		E.2	Implémenter une fonction de filtrage	Moyenne	Moyenne
		E.3	Mettre à jour dynamiquement la liste	Moyenne	Moyenne
F	En tant qu’administrateur, je veux créer des catégories.	F.1	Définir le modèle ou le schéma Mongoose de la catégorie avec relation vers le département	Haute	Moyenne
		F.2	Implémenter le contrôleur ‘addCategory’ côté backend	Haute	Moyenne
		F.3	Définir la route POST	Moyenne	Moyenne

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
		F.4	Créer le formulaire React pour l'ajout de catégorie avec champ département	Haute	Moyenne
		F.5	Remplir dynamiquement le champ département (fetch API)	Moyenne	Moyenne
		F.6	Envoyer les données à l'API à l'aide de Axios	Haute	Moyenne
G	En tant qu'administrateur, je veux modifier des catégories.	G.1	Implémenter 'updateCategory' dans le contrôleur backend	Haute	Moyenne
		G.2	Définir la route PUT dans Express	Moyenne	Moyenne
		G.3	Définir la route GET par id pour pré-remplir les champs de catégorie	Moyenne	Moyenne
		G.4	Créer un formulaire React pour modifier une catégorie	Moyenne	Moyenne
		G.5	Appeler l'API GET par id pour récupérer les données dans le formulaire	Moyenne	Moyenne
		G.6	Appeler l'API PUT pour mettre à jour les données	Haute	Moyenne
H	En tant qu'administrateur, je veux supprimer des catégories.	H.1	Implémenter 'deleteCategory' dans le contrôleur backend	Haute	Moyenne
		H.2	Définir la route DELETE	Haute	Moyenne
		H.3	Ajouter un bouton Supprimer côté React	Moyenne	Faible

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
		H.4	Envoyer la requête DELETE via Axios	Haute	Moyenne
		H.5	Mettre à jour l'interface après suppression	Moyenne	Moyenne
		H.6	Ajouter une boîte de confirmation avant suppression	Faible	Faible
I	En tant qu'administrateur, je veux voir la liste des catégories.	I.1	Implémenter l'API pour retourner et lister toutes les catégories avec départements	Moyenne	Moyenne
		I.2	Créer un composant React pour afficher la liste	Moyenne	Moyenne
		I.3	Afficher les données avec 'react-data-table-component'	Moyenne	Moyenne
		I.4	Utiliser 'useEffect' pour charger les données via Axios	Moyenne	Moyenne
J	En tant qu'administrateur, je peux filtrer les catégories pour faciliter leur recherche.	J.1	Créer l'API qui permet de filtrer les catégories	Moyenne	Moyenne
		J.2	Ajouter un champ pour écrire le nom de la catégorie	Moyenne	Moyenne
		J.3	Mettre à jour dynamiquement la liste des catégories selon le filtre sélectionné	Moyenne	Moyenne

3.2.2 Le diagramme de cas d'utilisation de sprint 2

Le diagramme de cas d'utilisation de notre deuxième sprint 2 est présenté dans la figure suivante

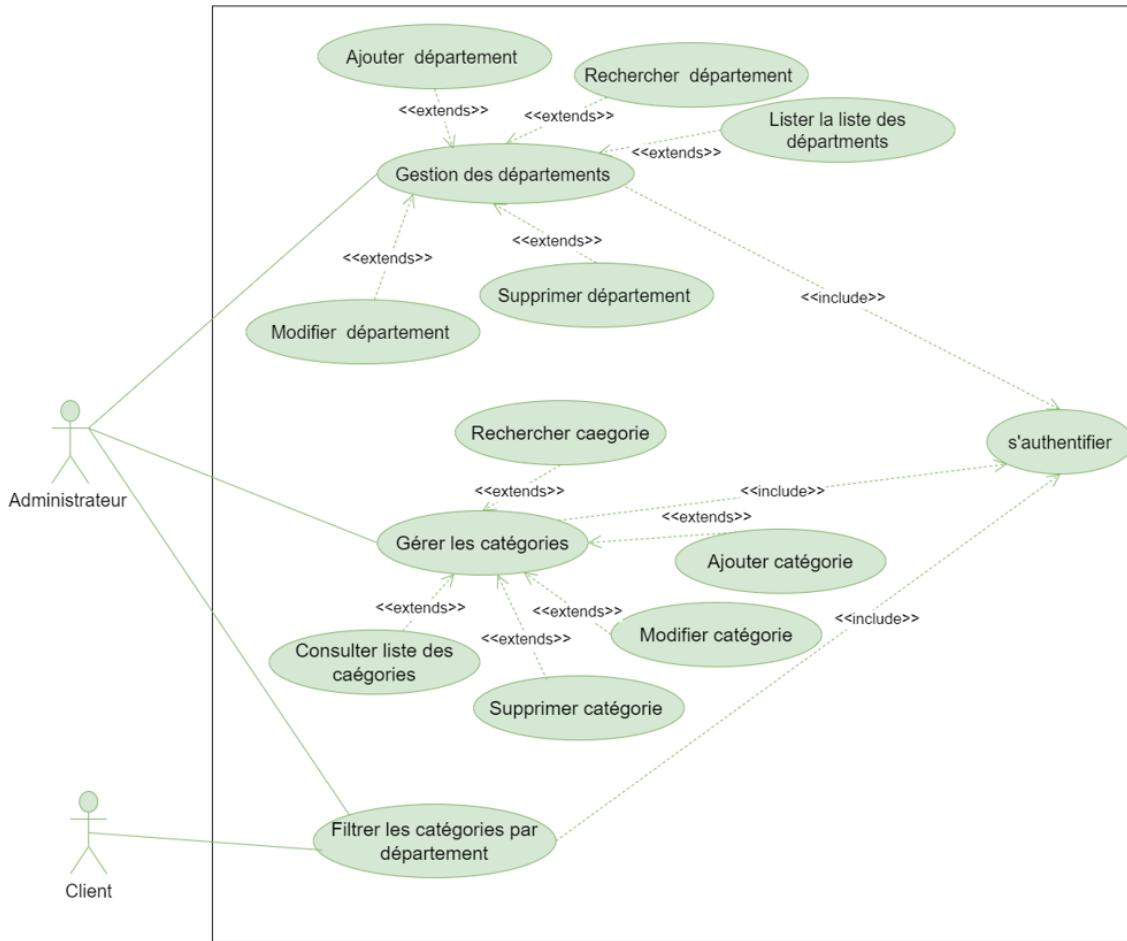


FIGURE 3.11 : Le diagramme de cas d'utilisation de sprint 2

3.2.3 Diagramme de Structure des Données sprint 2

La figure suivante représente le diagramme de Structure des Données du sprint "gestion des départements et gestion des catégories"

- Un département peut avoir plusieurs catégories.
- Une catégorie peut avoir plusieurs tickets .

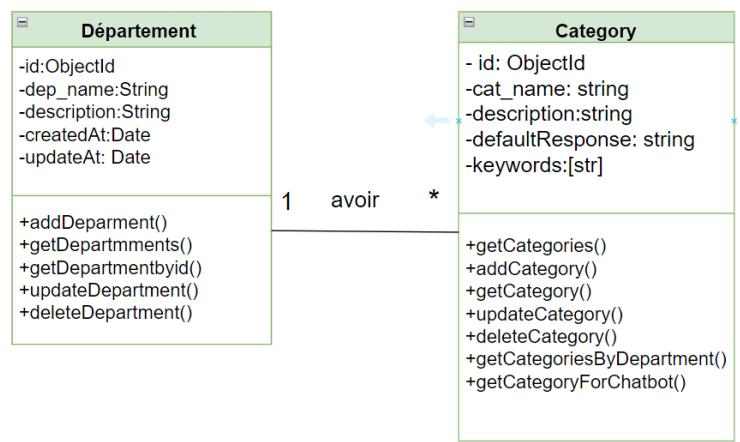


FIGURE 3.12 : Diagramme de Structure des Données sprint 2

3.2.4 Conception

3.2.4.1 Diagrammes des séquences "Gestion des départements"

Dans la figure suivante, on a le diagramme de séquence du cas d'utilisation "Ajouter un département".

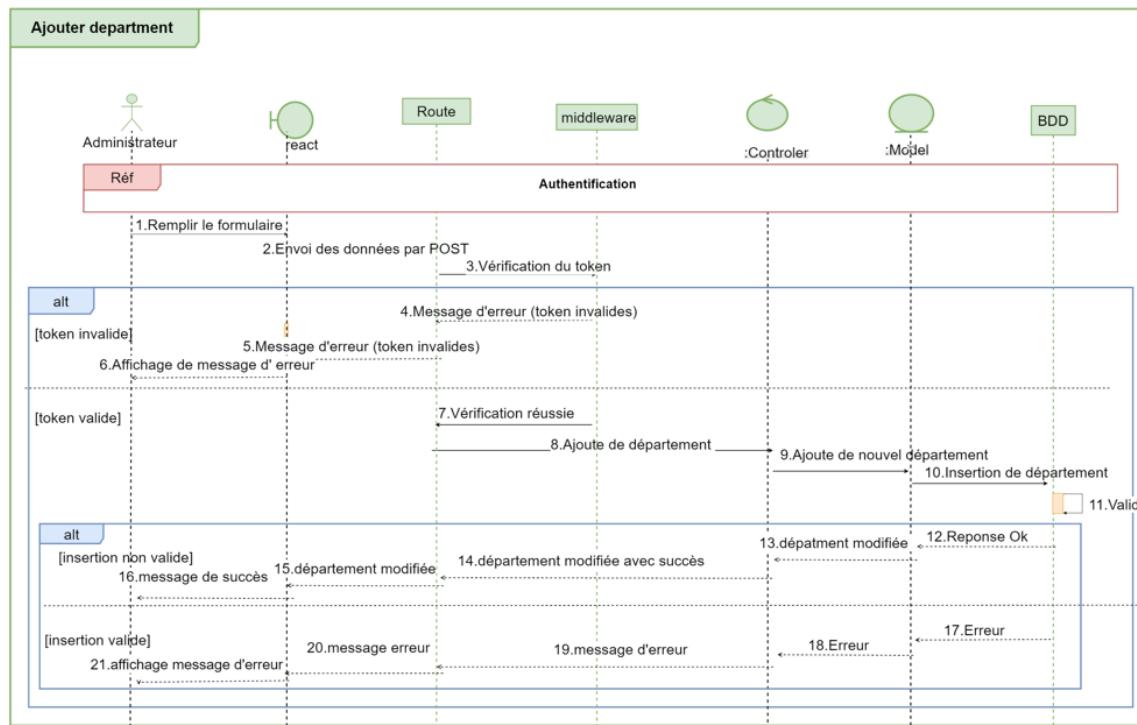


FIGURE 3.13 : diagramme de séquence de cas d'utilisation "ajouter un département"

Dans la figure suivante, on a le diagramme de séquence du cas d'utilisation "Modifier un département".

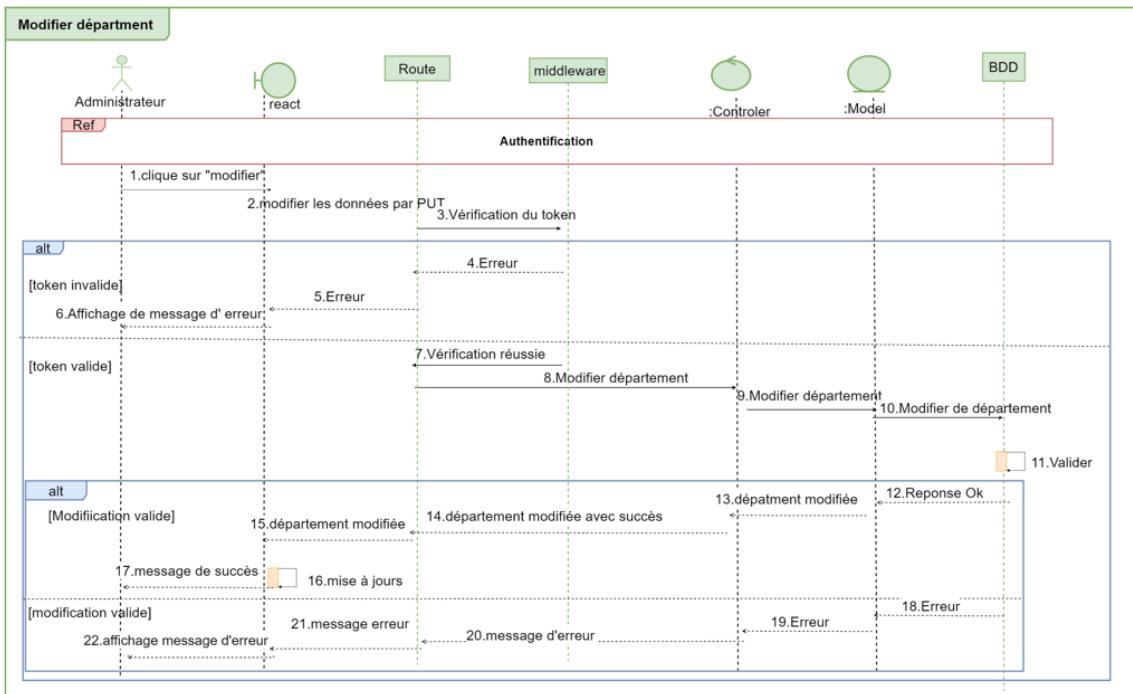


FIGURE 3.14 : diagramme de séquence de cas d'utilisation "modifier un département"

Dans la figure suivante ona le diagramme de séquence de cas d'utilisation "supprimer un département"

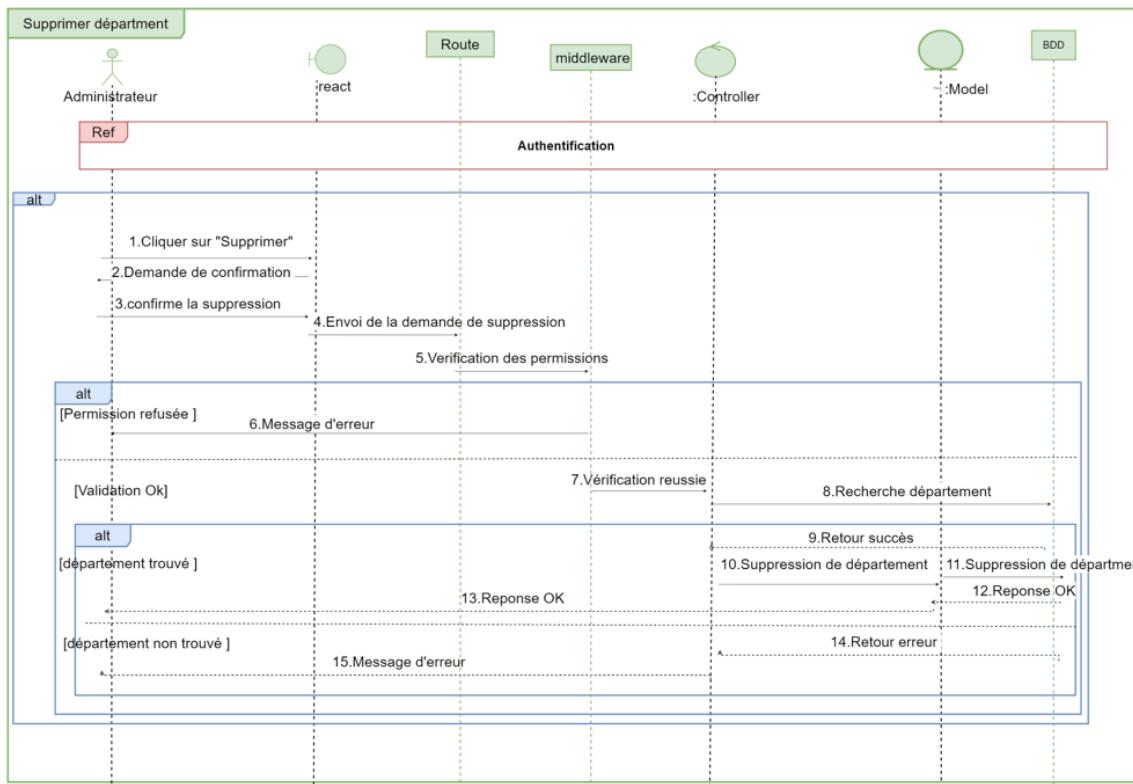


FIGURE 3.15 : Diagramme de séquence de cas d'utilisation "supprimer un département"

dans la figure suivante ona le diagramme de séquence de cas d'utilisation "lister les départements"

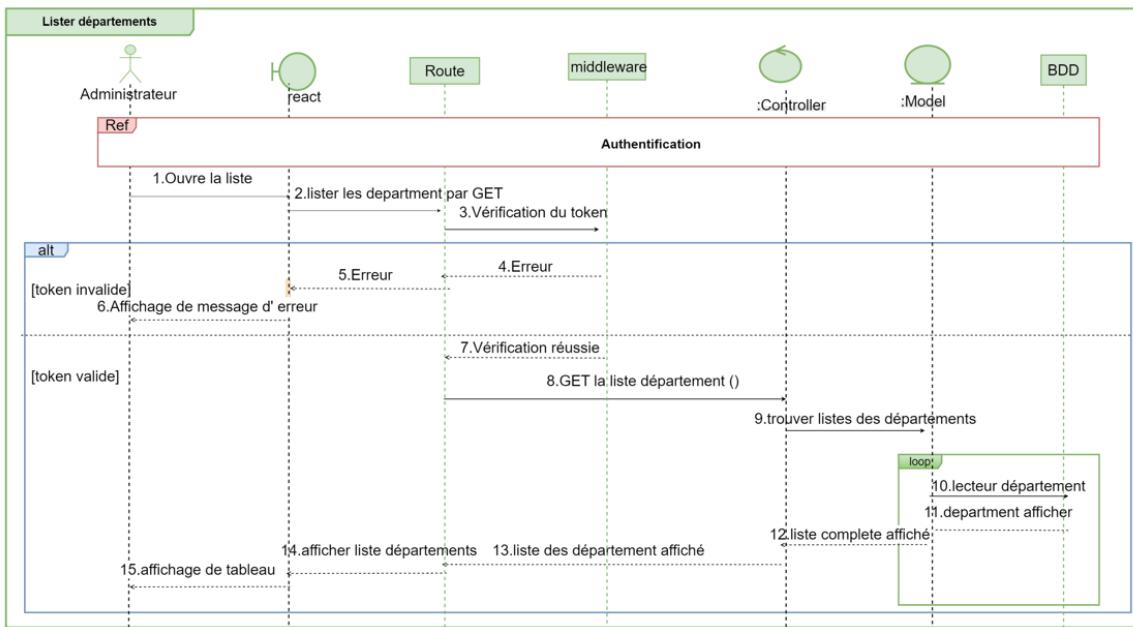


FIGURE 3.16 : Diagramme de séquence de cas d'utilisation "lister les départements"

3.2.4.2 Diagramme de séquence "Gestion des catégories"

Dans cette partie nous allons présenter les diagrammes séquences pour la gestion des catégories.

3.2.4.3 Diagramme de séquence du cas d'utilisation "Ajouter une catégorie"

La figure 3.17 illustre le diagramme de séquence correspondant à l'ajout de catégories.

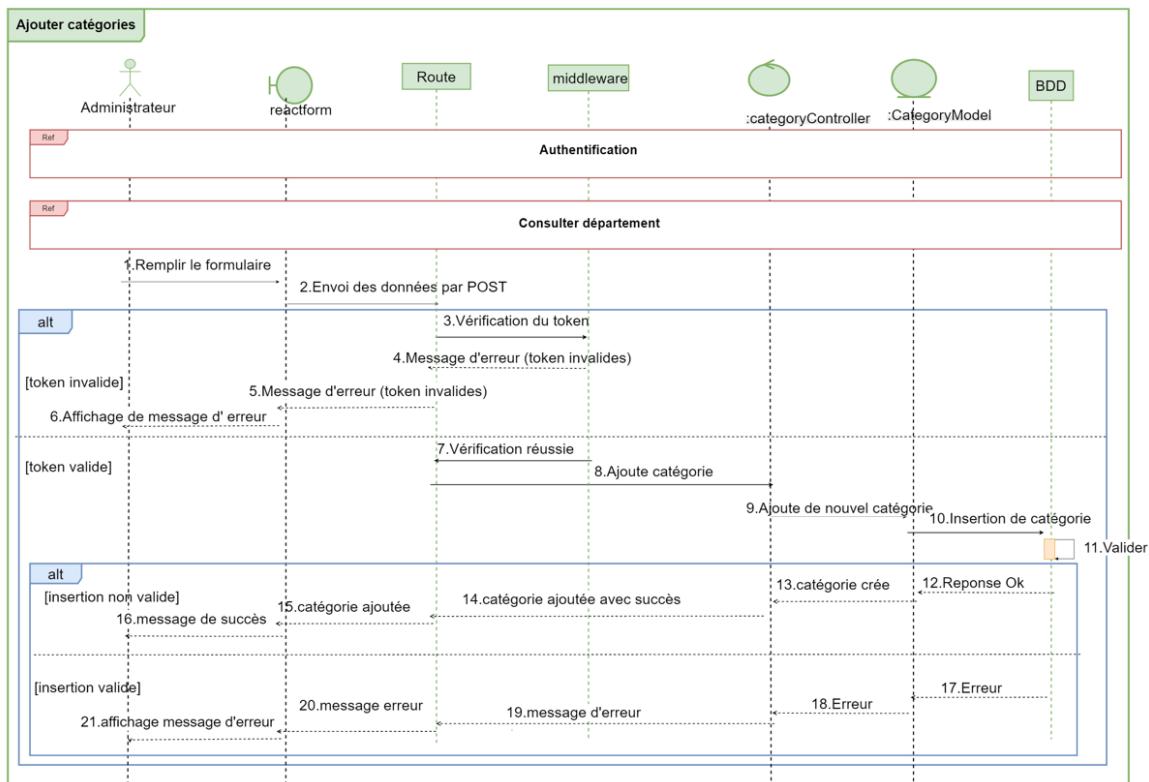


FIGURE 3.17 : Diagramme de séquence du cas d'utilisation "Ajouter une catégorie"

3.2.4.4 Diagramme de séquence de cas d'utilisation "Modifier une catégorie"

La figure 3.18 représente le diagramme de séquence de la modification d'une catégories.

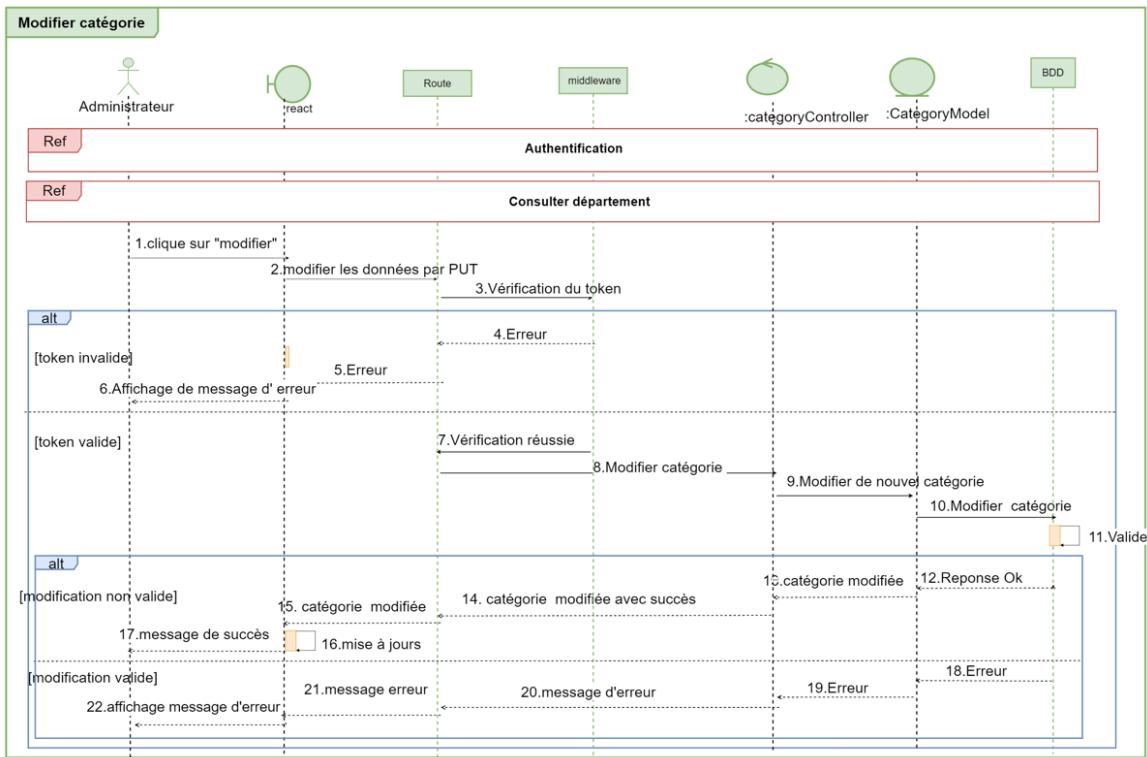


FIGURE 3.18 : Diagramme de séquence de cas d'utilisation "Modifier catégorie"

3.2.4.5 Diagramme de séquence de cas d'utilisation "Supprimer une catégorie"

La figure 3.19 représente le diagramme de séquence de la suppression d'une catégories.

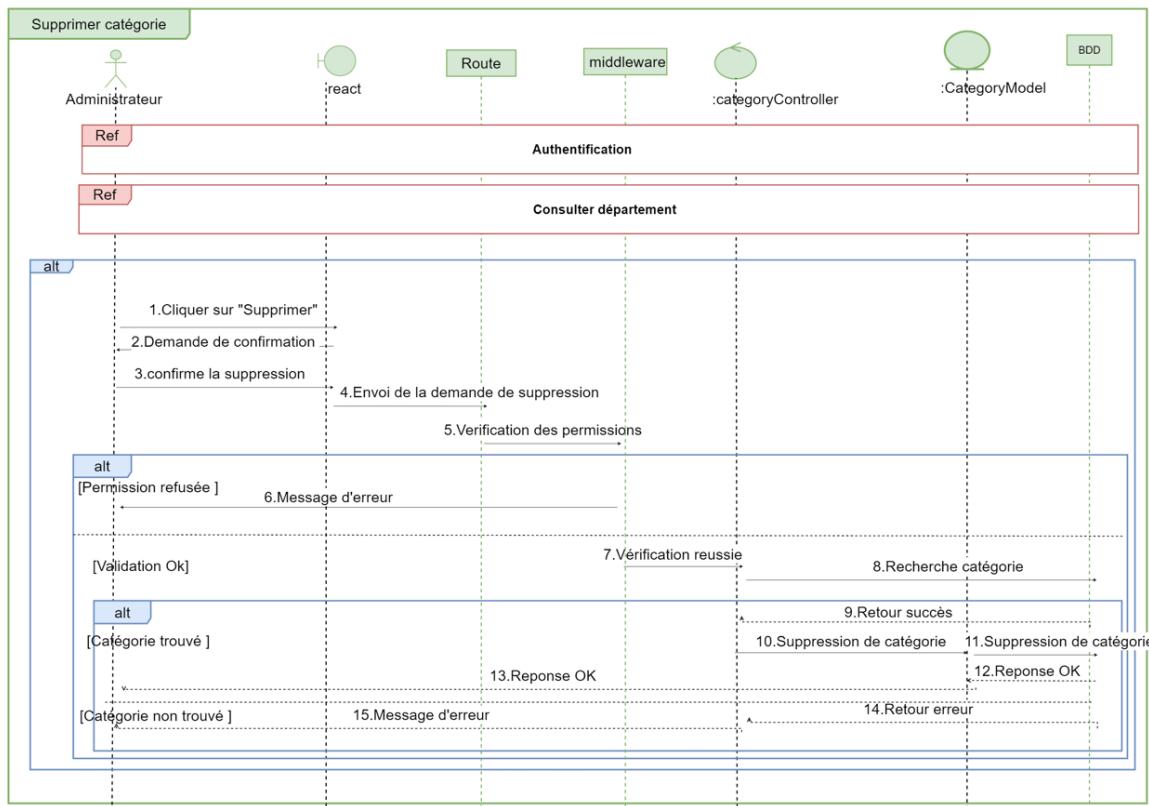


FIGURE 3.19 : Diagramme de séquence de cas d'utilisation "Supprimer catégorie"

3.2.4.6 Diagramme de séquence de cas d'utilisation "lister les categories"

La figure 3.20 représente le diagramme de séquence de la liste des catégories .

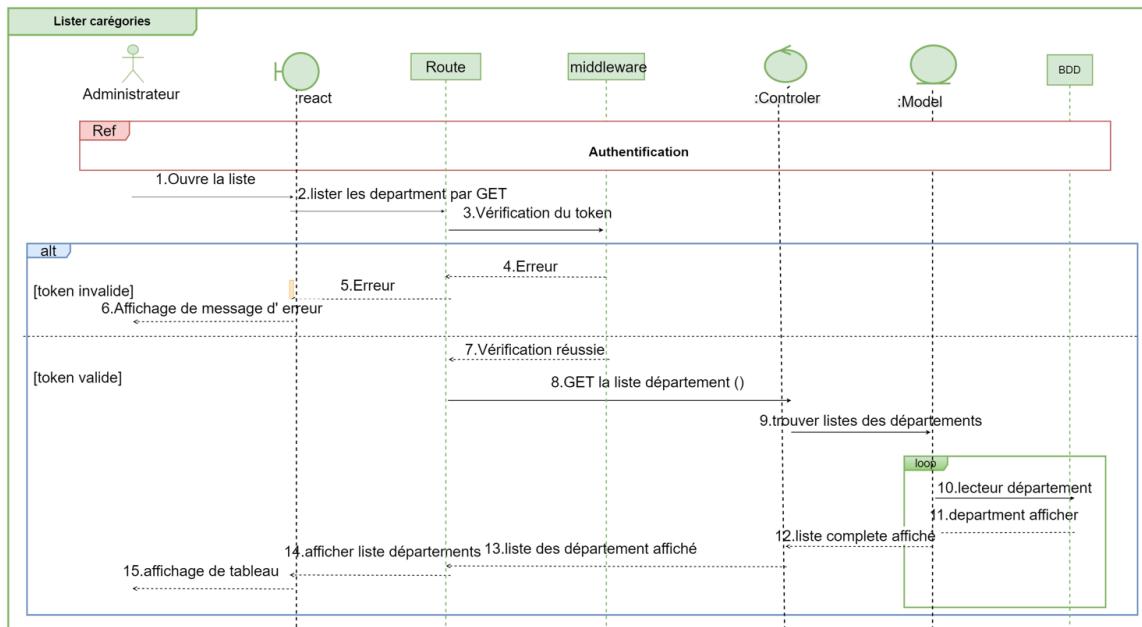


FIGURE 3.20 : Diagramme de séquence de cas d'utilisation "lister les categories"

3.2.5 Réalisation

3.2.5.1 Gestion des départements

- L'ajout d'un département : Les interfaces suivantes représentent la gestion des départements effectuée par l'administrateur. La figure 3.21 représente l'interface d'ajout d'un nouveau département avec un formulaire à remplir.

Bienvenue
eya benhjal

Add New Department

DepartmentName
Département Informatique

Description
Responsable du traitement et de la résolution des demandes liées aux problèmes techniques et aux services informatiques.

Add Department

Department added successfully!

FIGURE 3.21 : Interface d'ajout département

- La modification d'un département :

La figure 3.22 représente l'interface de modification d'un département.

Bienvenue
eya benhjal

Edit Department

DepartmentName
Département Informatique (IT)

Description
Créer des solutions informatiques efficaces, sécurisées et évolutives.

Edit Department

Département mis à jour avec succès!

FIGURE 3.22 : Interface de modification département

- La suppression d'un département :

La figure 3.23 représente l'interface de suppression d'un département. Après confirmation de la suppression, un message s'affiche indiquant que le département a été supprimé avec succès.

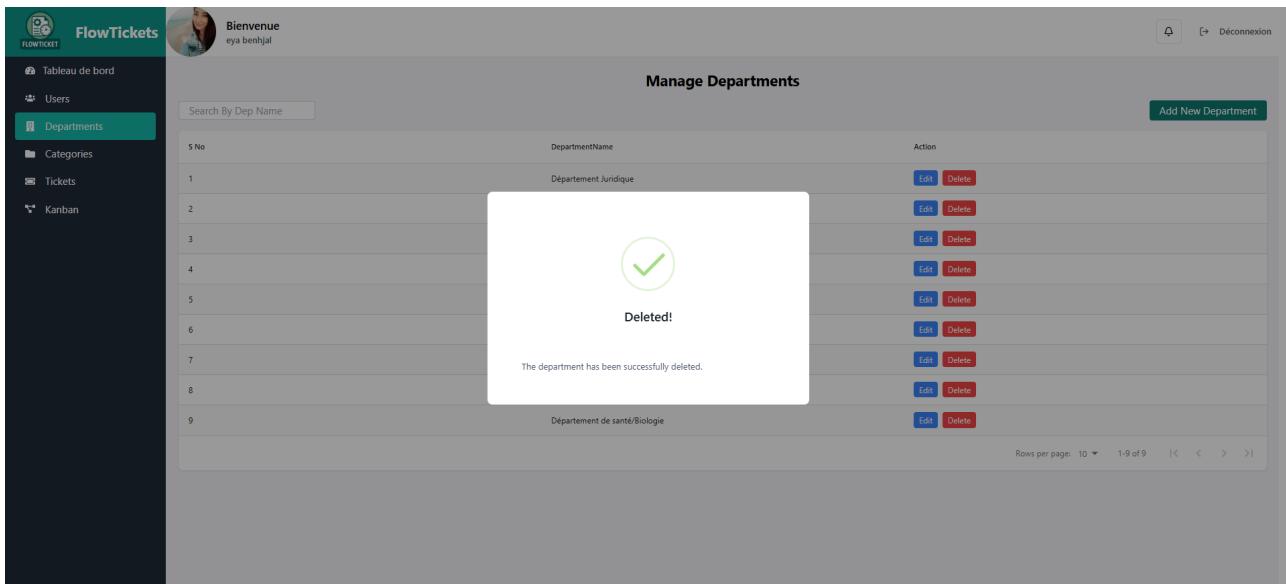


FIGURE 3.23 : Interface de suppression d'un département

- La liste des départements : La figure 3.24 illustre l'interface administrateur affichant la liste des départements, ainsi qu'un champ de recherche permettant de filtrer les départements par nom.

S No	DepartmentName	Action
1	Département Juridique	[Edit] [Delete]
2	Département électronique	[Edit] [Delete]
3	Département des Ressources Humaines (RH)	[Edit] [Delete]
4	Département Marketing et Communication	[Edit] [Delete]
5	Département Qualité / Contrôle qualité	[Edit] [Delete]
6	Département Recherche et Développement (R&D)	[Edit] [Delete]
7	Département Logistique et Approvisionnement	[Edit] [Delete]
8	Département Commercial / Ventes	[Edit] [Delete]
9	Département de santé/Biologie	[Edit] [Delete]
10	Département Informatique (IT)	[Edit] [Delete]

FIGURE 3.24 : Interface de liste des départements

3.2.5.2 Gestion des catégories

Les interfaces suivantes représentent la gestion des catégories effectuée par l'administrateur.

- L'ajout d'une catégorie :

La figure 3.25 représente l'interface d'ajout d'une nouvelle catégorie avec un formulaire à remplir.

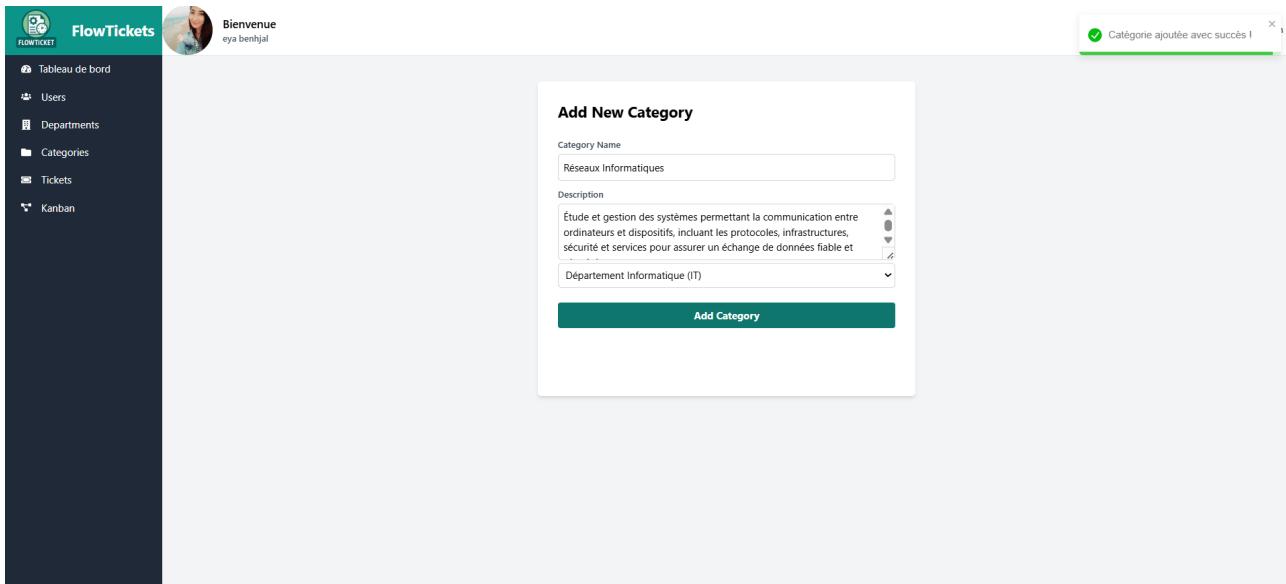


FIGURE 3.25 : Interface d'ajout catégorie

- La modification d'une catégorie :

La figure 3.26 représente l'interface de modification d'une catégorie .

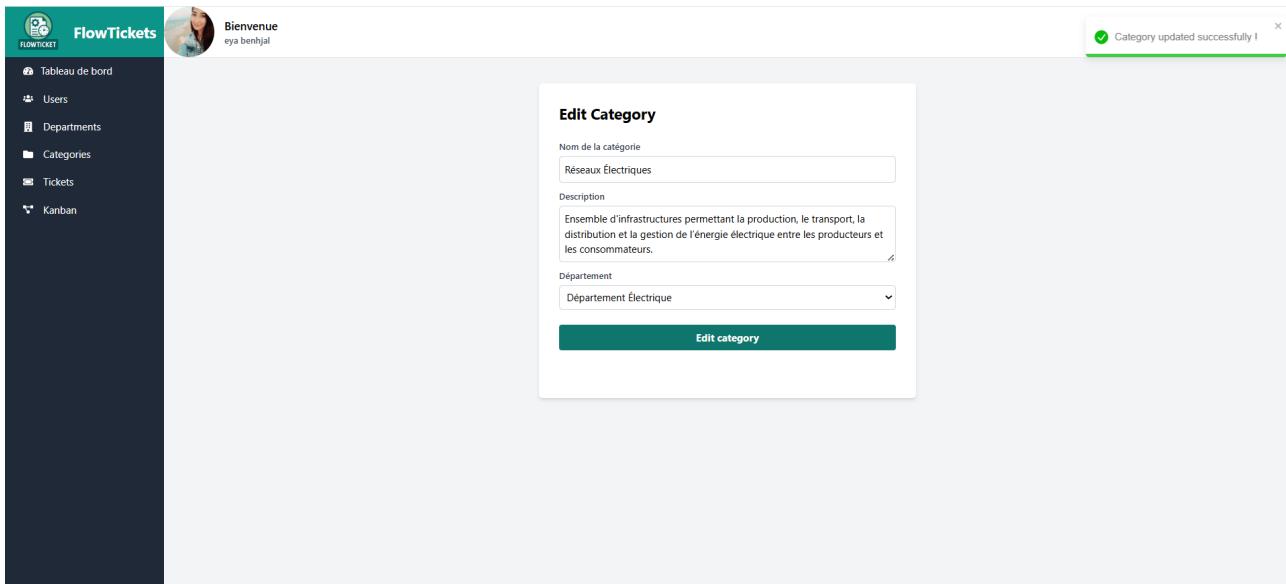


FIGURE 3.26 : Interface de modification de catégorie

- La suppression d'une catégorie :

La figure 3.27 représente l'interface de suppression d'un catégorie. Après confirmation de la suppression, un message s'affiche indiquant que la catégorie a été supprimé avec succès.

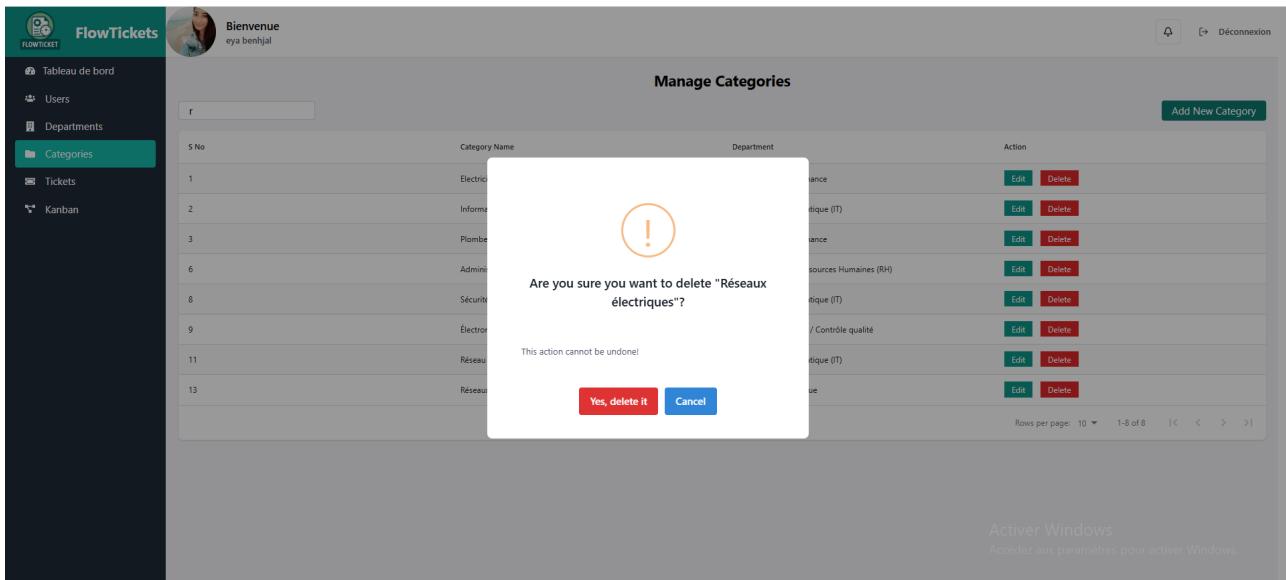


FIGURE 3.27 : Interface de suppression de catégorie

- La liste des catégories :

La figure 3.28 illustre l'interface administrateur affichant la liste des départements, accompagnée d'un champ de recherche permettant de filtrer les catégories par nom.

No	Category Name	Department	Action
1	Électricité	département maintenance	Edit Delete
2	Informatique	Département Informatique (IT)	Edit Delete
3	Plomberie	département maintenance	Edit Delete
4	Télévision	Département Commercial / Ventes	Edit Delete
5	Finance	département Finance	Edit Delete
6	Administratif	Département des Ressources Humaines (RH)	Edit Delete
7	Santé	Département de santé/Biologie	Edit Delete
8	Sécurité informatique	Département Informatique (IT)	Edit Delete
9	Électroménager	Département Qualité / Contrôle qualité	Edit Delete
10	Téléphone mobile	Département Commercial / Ventes	Edit Delete

FIGURE 3.28 : Interface de liste des catégories

Conclusion

Au cours de ce chapitre, nous avons présenté la réalisation de la première release, qui comprend le Sprint 1 : Inscription et authentification, ainsi que le Sprint 2 : Gestion des départements et gestion des catégories. Pour ce faire, nous sommes passés par les étapes d'analyse, de conception et de réalisation de ces deux premiers sprints.

RELEASE 2 : ÉTUDE ET RÉALISATION DE LA RELEASE 2

Plan

1	sprint 3 :Module gestion des utilisateurs et des rôles	61
2	Sprint 4 :Module gestion du profil et des horaires	70

Introduction

Dans ce chapitre, nous allons présenter les différentes étapes de réalisation du troisième sprint "Gestion des utilisateurs et des rôles", et du quatrième sprint "gestion profil et horaire " et cinquième sprint "Tickets client/agent + Notifications " et sixième sprint "Tickets Admin + Commentaires".

4.1 sprint 3 :Module gestion des utilisateurs et des rôles

Dans cette section nous allons présenter les différents étapes de la réalisation du sprint "Gestion des utilisateurs et des rôles".

4.1.1 Objectifs du sprint 3

L'objectif du troisième sprint est de développer Le module "Gestion des utilisateurs et des rôles" permet, d'une part, aux administrateurs de gérer les Gestion des utilisateurs et des rôles en fonction du département ou de les associer à un département.

4.1.2 Backlog du sprint 3

TABLEAU 4.1 : Backlog de sprint 3

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
A	En tant qu'administrateur, je peux ajouter des utilisateurs.	A.1	Implémenter l'ajout d'utilisateur côté backend avec validation des champs, gestion de l'image (avec Multer), et association au département.	Haute	Moyenne
			Créer un formulaire d'ajout côté frontend avec envoi des informations via FormData.		
			Afficher les messages d'erreur ou succès à partir de la réponse du serveur.		

		A.4	Rediriger automatiquement vers la liste des utilisateurs après un ajout réussi.	Faible	Faible
B	En tant qu'administrateur, je peux supprimer un utilisateur.	B.1	Implémenter une route DELETE backend pour la suppression d'un utilisateur.	Haute	Moyenne
		B.2	Ajouter un bouton de suppression dans la liste avec une confirmation.	Moyenne	Simple
C	En tant qu'administrateur, je veux pouvoir lister les utilisateurs existants.	C.1	Implémenter une API pour récupérer tous les utilisateurs.	Moyenne	Moyenne
		C.2	Afficher les utilisateurs avec DataTable côté frontend.	Moyenne	Complexe
D	En tant qu'administrateur, je veux pouvoir modifier le rôle ou les données d'un utilisateur existant.	D.1	Créer le contrôleur pour la mise à jour de l'utilisateur.	Haute	Moyenne
		D.2	Définir la route PUT dans l'API en utilisant multipart/form-data.	Haute	Faible
		D.3	Définir la route GET pour récupérer les données d'un utilisateur spécifique.	Moyenne	Faible
		D.4	Créer un formulaire de modification côté client pré-rempli avec les informations de l'utilisateur sélectionné.	Moyenne	Moyenne

	D.5	Gérer les erreurs du formulaire côté frontend.	Moyenne	Moyenne
	D.6	Rediriger vers la liste des utilisateurs après la mise à jour.	Faible	Faible
	D.7	Afficher un message de succès après modification.	Faible	Faible
E	En tant qu'administrateur, je veux rechercher un utilisateur existant par son nom.	E.1	Ajouter un champ de recherche dynamique dans DataTable côté frontend.	Moyenne
		E.2	Filtrer dynamiquement les utilisateurs affichés selon leur nom.	Moyenne

4.1.3 Spécification des besoins fonctionnels

Cette section résume le raffinement des cas d'utilisation du sprint 3, faciliter la compréhension des fonctionnalités attendues. La figure 4.1 représente le diagramme de cas d'utilisation du sprint 3 "gestion des utilisateurs et des rôles"

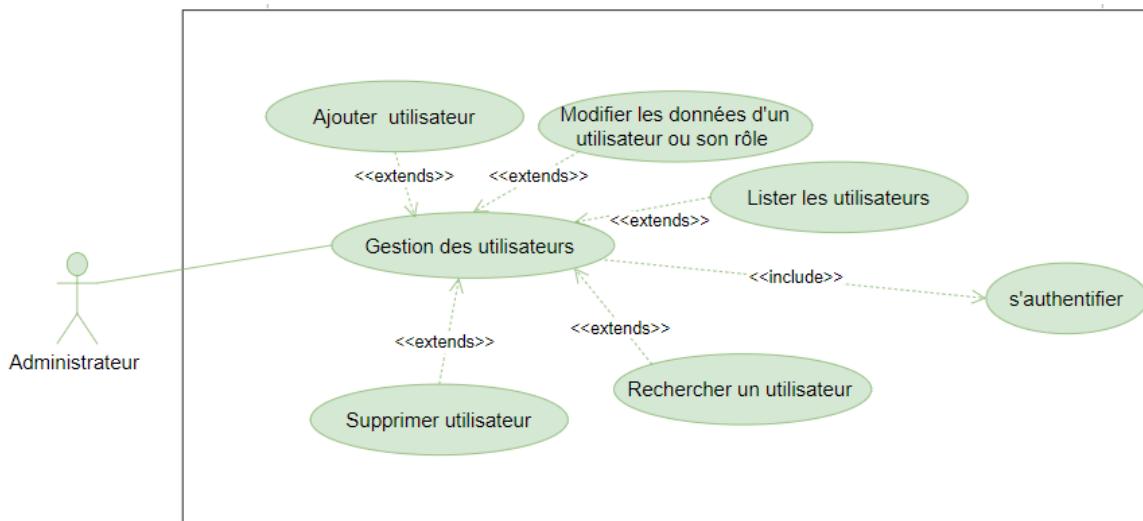


FIGURE 4.1 : Diagramme de cas d'utilisation "gestion des utilisateurs et des rôles"

4.1.4 Diagramme de Structure des Données sprint 3

La figure 4.2 représente le diagramme de classe du sprint "Gestion des utilisateurs et rôles".

- Un département peut avoir plusieurs utilisateurs.

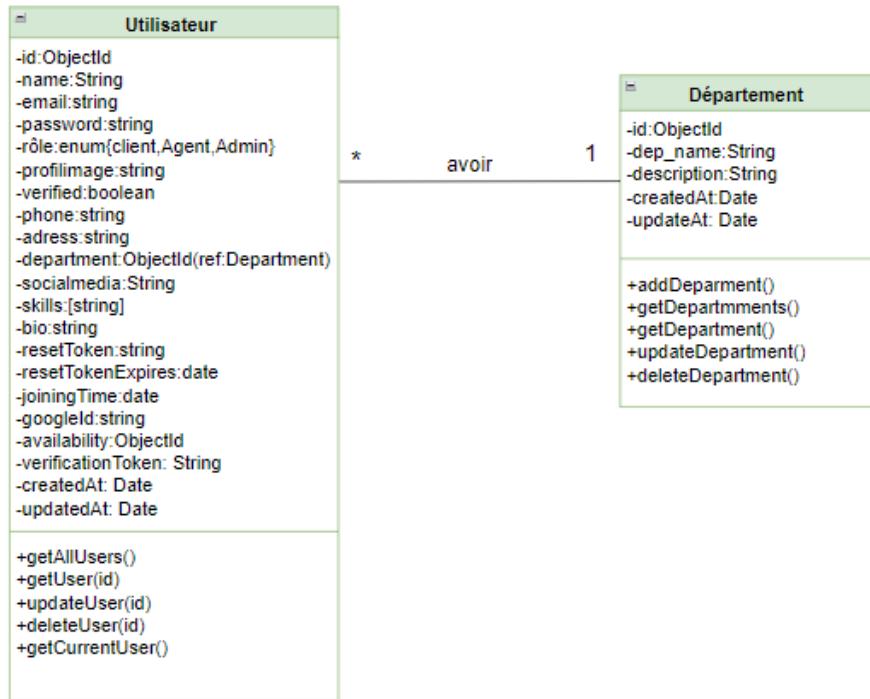


FIGURE 4.2 : Diagramme de Structure des Données sprint 3

4.1.5 Conception

4.1.5.1 Diagrammes de séquence "Gestion des utilisateurs "

Les figures suivantes illustre le diagramme de séquence de Gestion des utilisateurs et rôle.

L'interface suivante représente le diagramme de séquence cas d'utilisation de l'ajout d'un utilisateur qui peut être réalisé par l'administrateur.

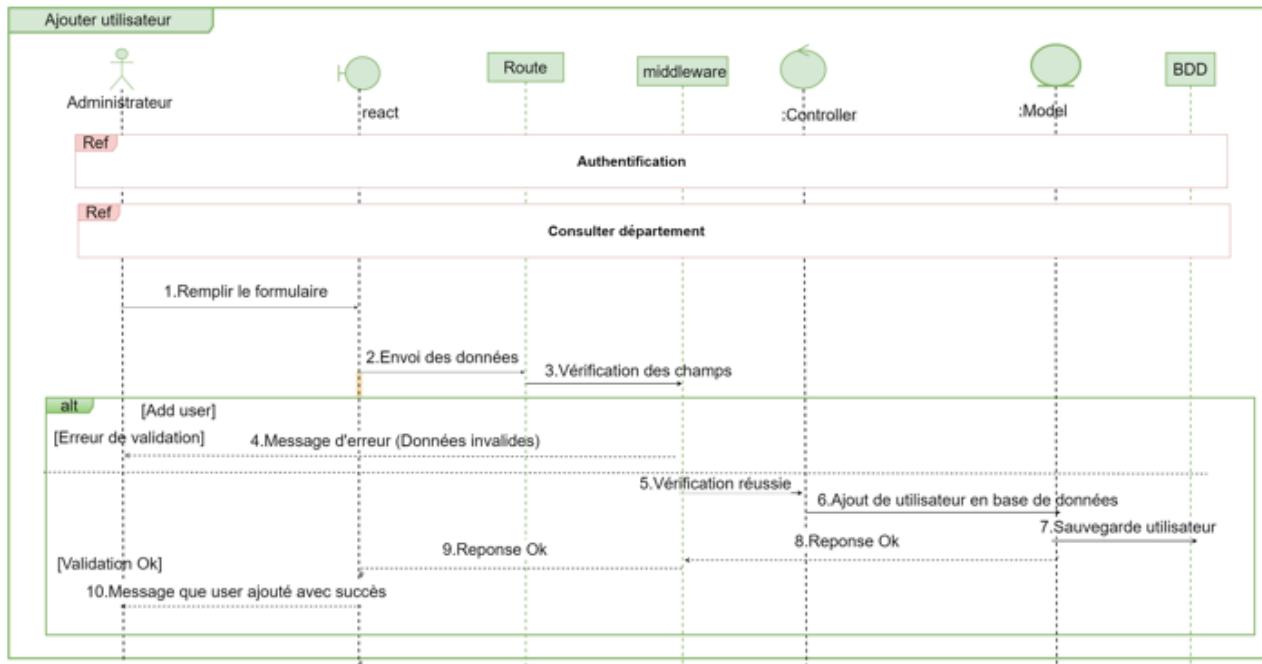


FIGURE 4.3 : Diagramme de séquence de cas d'utilisation "Ajouter utilisateur"

L'interface suivante représente le diagramme de séquence de cas d'utilisation modifier un utilisateur qui peut être réalisé par l'administrateur.

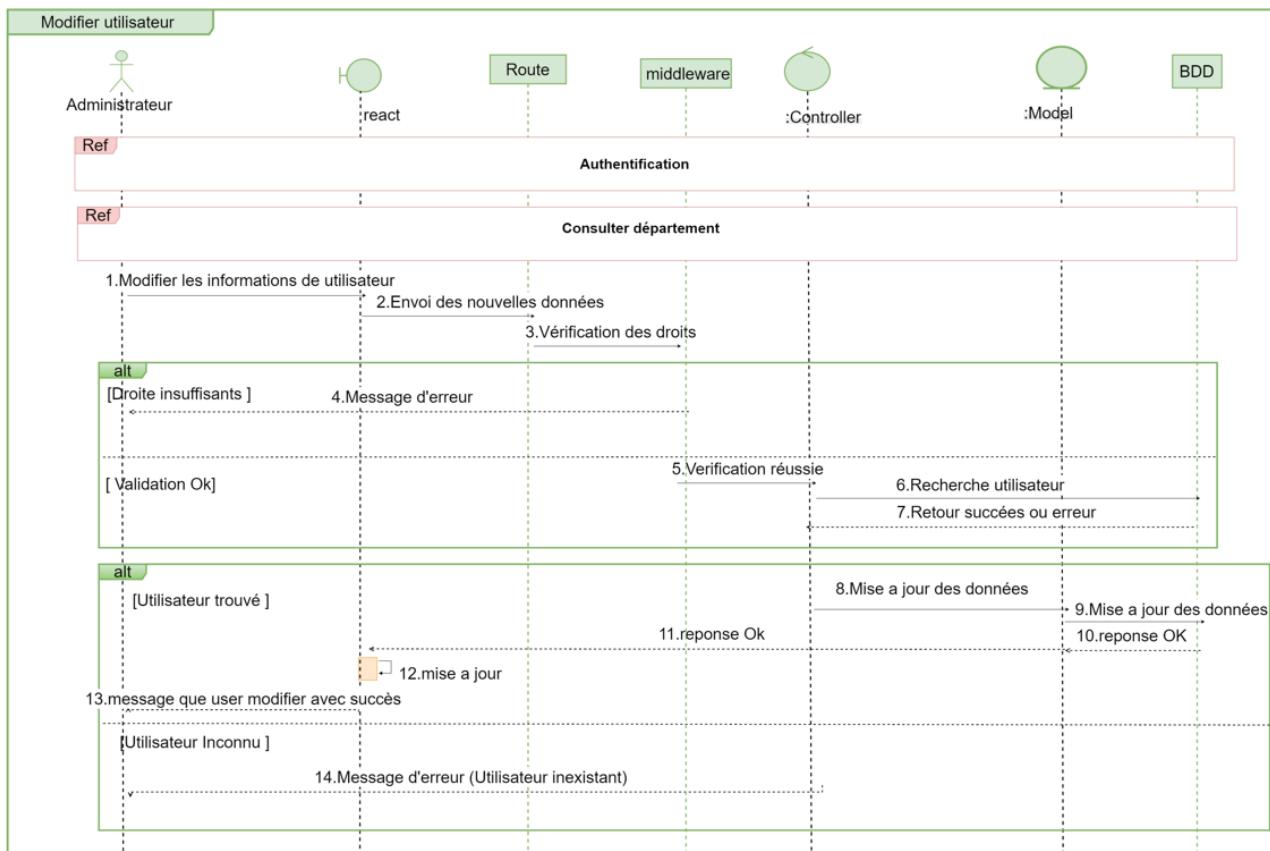


FIGURE 4.4 : diagramme de séquence de cas d'utilisation "modifier un utilisateur"

L'interface suivante représente le diagramme de séquence de cas d'utilisation supprimer un utilisateur

qui peut être réalisé par l'administrateur.

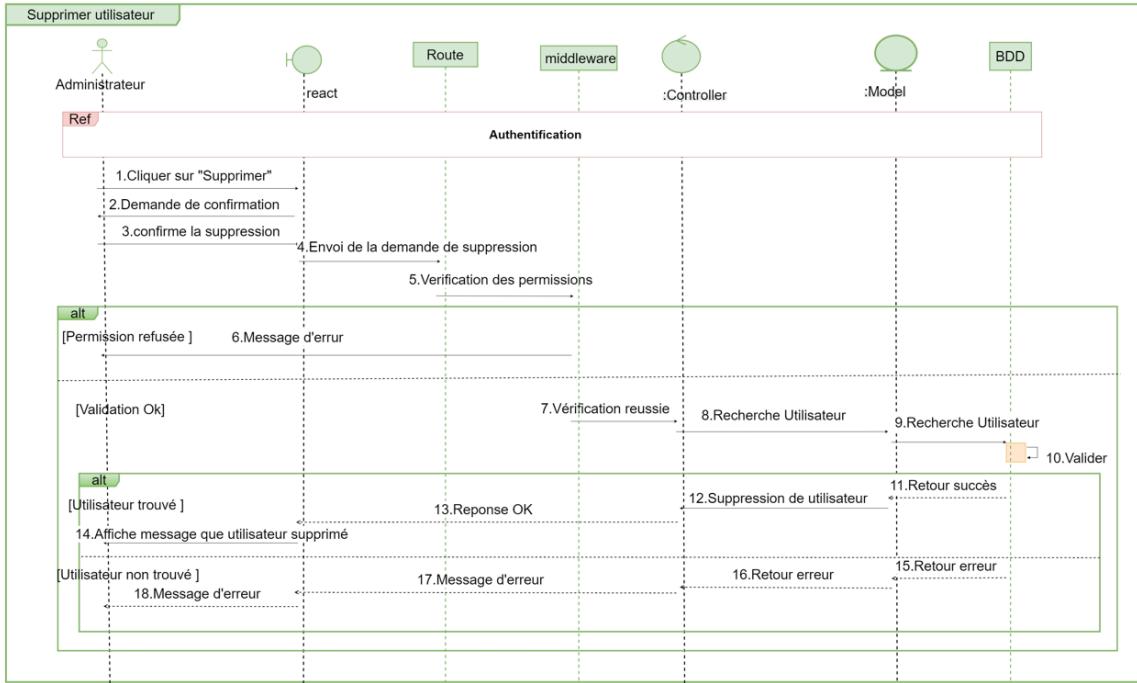


FIGURE 4.5 : diagramme de séquence de cas d'utilisation "supprimer un utilisateur"

4.1.6 Réalisation

4.1.6.1 Gestion des utilisateurs

- L'ajout d'un utilisateur :

La figure 4.6 présente l'interface dédiée à l'ajout d'un nouvel utilisateur via un formulaire. Cette interface permet de renseigner les informations nécessaires, telles que le nom ,le rôle et l'adresse e-mail, et d'affecter l'utilisateur à un département spécifique. Dans cet exemple, un utilisateur ayant le rôle de client et portant le nom SalhaBH a été ajouté avec succès.

FIGURE 4.6 : Formulaire d'ajout d'utilisateur

- L'envoi d'un mail pour le nouvelle utilisateur :

Avec l'ajout de l'utilisateur l'application envoie un email instantanément contenant les coordonnées de connexion (nom d'utilisateur et mot de passe) vers l'adresse mail fournie dans le formulaire comme nous indique la figure 4.7.

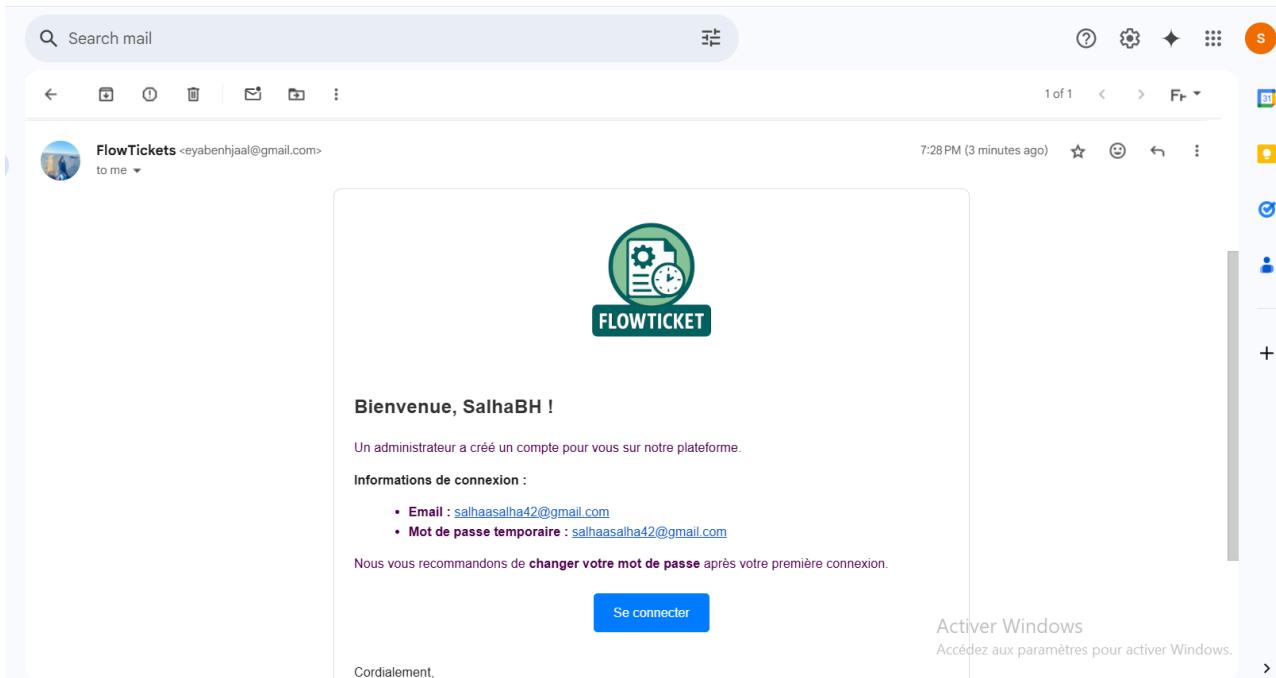


FIGURE 4.7 : L'email contenant les coordonnées de connexion

- Détails de profil utilisateur : Maintenant on va voir les détails de données d'un utilisateur . La figure 4.8 représente la page profil utilisateur.

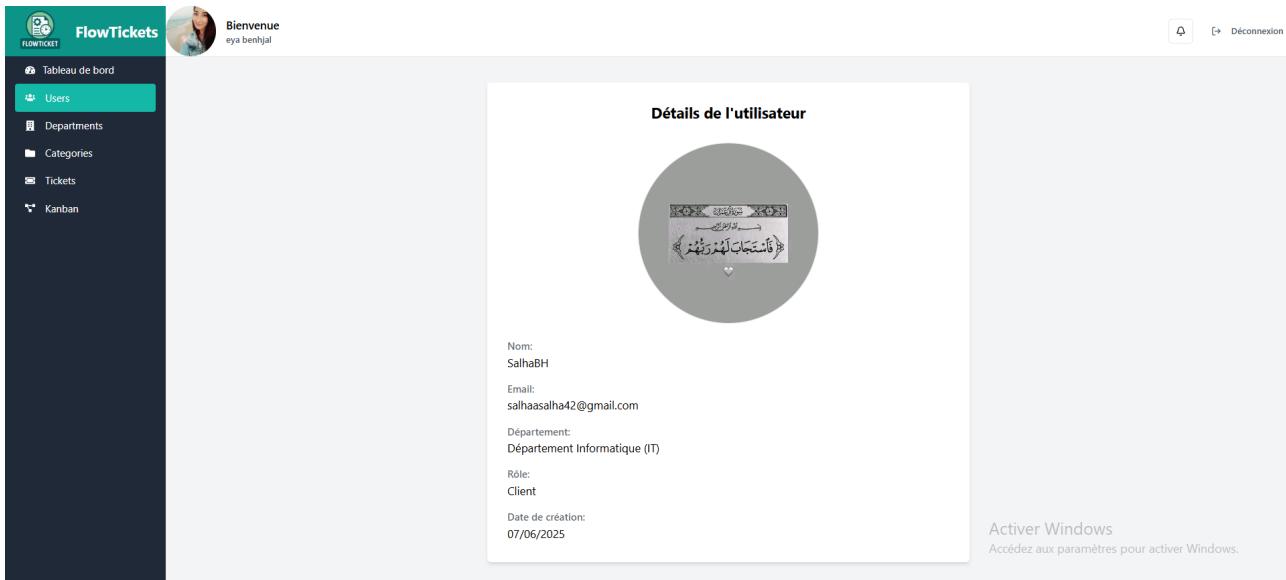


FIGURE 4.8 : Interface affiche le détails d'un utilisateur "Client"

- Modifier rôle et données :

Ici, on peut modifier le rôle d'un utilisateur. Si l'on sélectionne “Agent”, les champs Bio et Compétences sont s'affichent .

En revanche, si l'on choisit “Client” , les champs spécifiques à l'agent se désactivés .

Une fenêtre pop-up permet également de confirmer le changement de rôle. La figure 4.9 illustre le cas où l'administrateur modifie le rôle d'un client pour lui attribuer celui à agent.

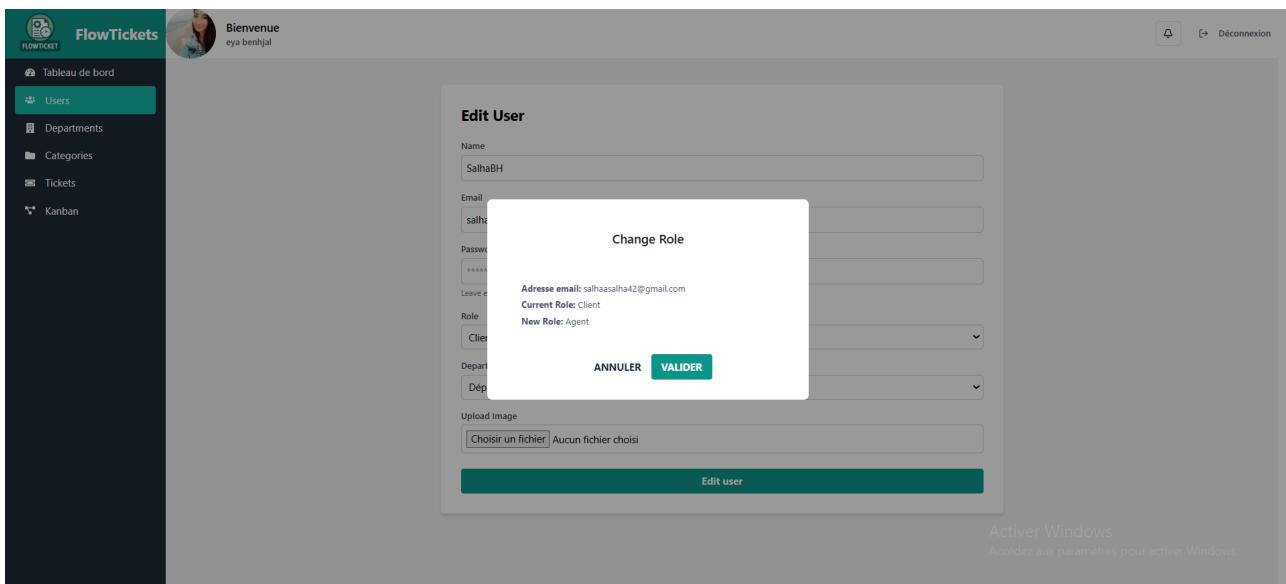


FIGURE 4.9 : Interface Changement de rôle

La page de modification des données utilisateurs permet à l'administrateur de compléter et de modifier les informations d'un utilisateur. Elle lui offre également la possibilité de changer les coordonnées de connexion, comme l'illustre la capture suivante (voir figure 4.10).

FIGURE 4.10 : Interface de modifier les données d'un utilisateur

- Lister les utilisateurs :

La figure suivante illustre l'interface administrateur affichant la liste des utilisateurs, ainsi qu'un

champ de recherche permettant de filtrer les utilisateurs par nom.

on a dans la liste des utilisateurs de rôle :

— Client

— Admin

— Agent

#	image	Nom	Email	Rôle	Département	Actions
1		clienteyaaa	eya.benhjal@gmail.com	Client	Département informatique (IT)	<button>View</button> <button>Edit</button> <button>Delete</button>
2		eya benhjal	eyabenhjal@gmail.com	Admin	Département informatique (IT)	<button>View</button> <button>Edit</button> <button>Delete</button>
3		agenteya	eya.benhjal@istic.ucar.tn	Agent	Département informatique (IT)	<button>View</button> <button>Edit</button> <button>Delete</button>
4		malek	malek.benzarti@groupeleclerc.com.tn	Client	Département Informatique (IT)	<button>View</button> <button>Edit</button> <button>Delete</button>
5		Chayma	eyabenhjal@istic.ucar.tn	Client	département maintenance	<button>View</button> <button>Edit</button> <button>Delete</button>
6		SalhaBH	salhaasalha42@gmail.com	Agent	Département Informatique (IT)	<button>View</button> <button>Edit</button> <button>Delete</button>
7		mouradiEya	eyab.enhjal@istic.ucar.tn	Client	Département des Ressources Humaines (RH)	<button>View</button> <button>Edit</button> <button>Delete</button>
8		chayma	chaymabh52@gmail.com	Agent	Département des Ressources Humaines (RH)	<button>View</button> <button>Edit</button> <button>Delete</button>

FIGURE 4.11 : Interface de la liste des utilisateurs

- Supprimer un utilisateur En revenant à l'interface de la liste des utilisateurs, l'administrateur peut cliquer sur le bouton “Delete” pour supprimer un utilisateur, après confirmation.

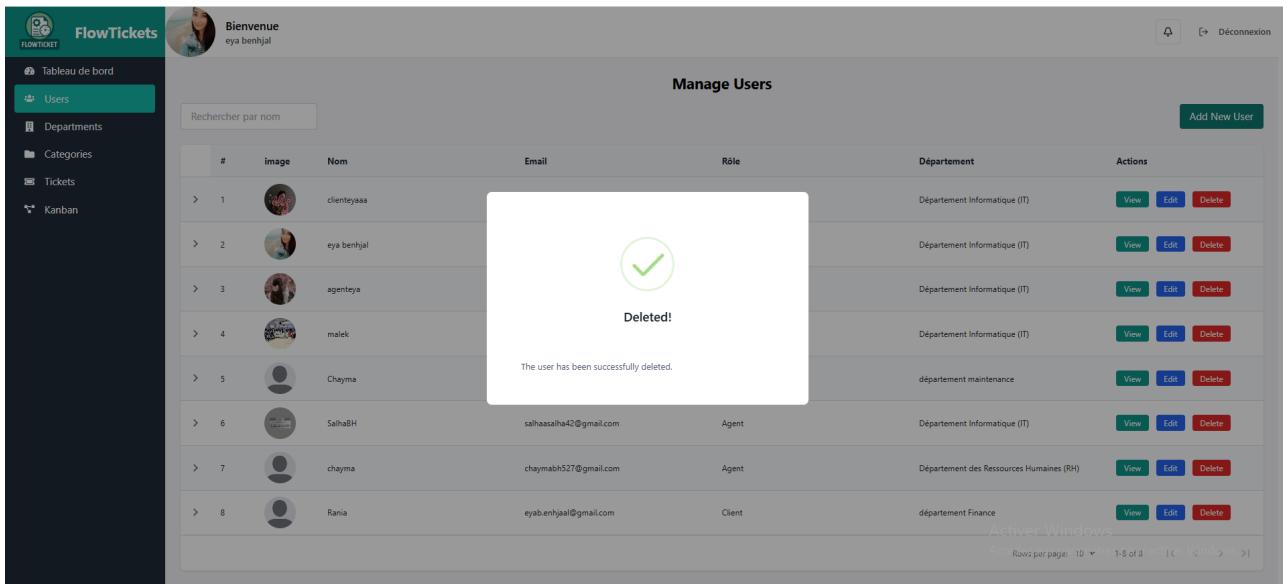


FIGURE 4.12 : Interface de suppression un utilisateur

4.2 Sprint 4 :Module gestion du profil et des horaires

Dans cette section nous allons présenter les différents étapes de la réalisation du sprint "Gestion du profil et des horaires".

4.2.1 Objectifs du sprint 4

L'objectif du quatrième sprint est de développer le module « Gestion du profil et des horaires » qui permet aux utilisateurs de faire gérer son profil et de gérer ses horaires de travail . De plus, permettant au administrateur de consulter les listes des profils de tous les agents et les clients,aussi le modifier.

4.2.2 Backlog du sprint 4

TABLEAU 4.2 : Backlog de sprint 3

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
A	En tant qu'agent, je veux ajouter mes compétences, bio, LinkedIn, Twitter...	A.1	Créer le formulaire d'ajout de compétences et réseaux sociaux	Haute	Moyenne
		A.2	Modifier le modèle User pour inclure bio, socialMedia, skills	Haute	Moyenne

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
		A.3	Enregistrement via PUT /api/profile/up avec upload image	Haute	Moyenne
B	En tant qu'agent, je veux modifier mes informations sur mon profil	B.1	updateProfile dans profileControllers.js	Haute	Moyenne
		B.2	Route PUT /up avec middleware upload	Haute	Moyenne
		B.3	Onglet Informations avec formulaire complet	Haute	Moyenne
		B.4	Association via handlePersonalInfoSubmit	Haute	Moyenne
C	En tant qu'agent, je veux supprimer mes informations	C.1	removeSkill dans profileControllers.js	Haute	Moyenne
		C.2	Boutons sur chaque compétence	Haute	Moyenne
		C.3	Confirmation implicite via toast	Moyenne	Basse
D	En tant qu'agent, je veux modifier mon statut de disponibilité	D.1	Champs dans modèle User	Haute	Moyenne
		D.2	Onglet Disponibilités avec interface complète	Haute	Moyenne
		D.3	Gestion via handleSaveAvailability	Haute	Moyenne
E	En tant que client, je veux modifier mes informations	E.1	API spécifique client manquante	Moyenne	Moyenne
		E.2	Formulaire client séparé à créer	Moyenne	Moyenne

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
F	En tant qu'admin, je veux voir la liste complète des profils utilisateurs	F.1	Créer l'API pour récupérer tous les profils	Moyenne	Moyenne
		F.2	Ajouter la possibilité de filtrer les profils par catégorie	Moyenne	Moyenne
		F.3	Intégrer le listing dans le tableau de bord admin	Moyenne	Moyenne
G	En tant qu'admin, je veux modifier les champs du profil d'un utilisateur	G.1	Créer un formulaire d'édition de profil pour l'admin	Moyenne	Moyenne
		G.2	Ajouter l'appel à l'API de mise à jour profil admin	Moyenne	Moyenne
H	En tant qu'admin, je veux supprimer les informations de profil d'un utilisateur	H.1	Ajouter bouton de suppression dans le tableau admin	Moyenne	Moyenne
		H.2	Implémenter 'deleteUser' côté backend	Moyenne	Moyenne
		H.3	Ajouter alerte de confirmation avant suppression	Moyenne	Basse
I	En tant qu'admin, je veux voir les détails complets d'un profil utilisateur	I.1	Créer une page avec les informations détaillées d'un utilisateur	Moyenne	Moyenne
		I.2	Relier cette page au tableau de bord admin (bouton "View")	Moyenne	Basse

ID	User Story	Sub-ID	Tâches	Priorité	Complexité
J	En tant qu'agent, je veux enregistrer mes horaires de disponibilité	J.1	Créer l'interface d'ajout d'horaires	Moyenne	Moyenne
		J.2	Créer le modèle <code>Availability</code>	-	Moyenne
		J.3	Créer la route POST <code>/availability</code>	-	Moyenne
		J.4	Créer le contrôleur et la méthode <code>createOrUpdateAvailability</code>	-	Moyenne
J	En tant qu'agent, je veux enregistrer mes horaires de disponibilité	J.1	Créer l'interface d'ajout d'horaires	Moyenne	Moyenne
		J.2	Créer le modèle	Moyenne	Moyenne
		J.3	Créer la route backend	Moyenne	Moyenne
		J.4	Créer le contrôleur associé	Moyenne	Moyenne
K	En tant qu'agent, je veux modifier mes horaires de disponibilité	K.1	Créer l'interface de modification d'horaires	Moyenne	Moyenne
		K.2	Implémenter la mise à jour dans le backend	Moyenne	Moyenne
L	En tant qu'agent, je veux supprimer un horaire de disponibilité	L.1	Ajouter un bouton de suppression d'un créneau	Moyenne	Moyenne
		L.2	Créer la route de suppression	Moyenne	Moyenne
		L.3	Implémenter la confirmation et suppression dans le backend	Moyenne	Basse

4.2.3 Spécification des besoins fonctionnels

La figure 4.13 représente le diagramme cas d'utilisation du sprint "Gestion du profil et des horaires".

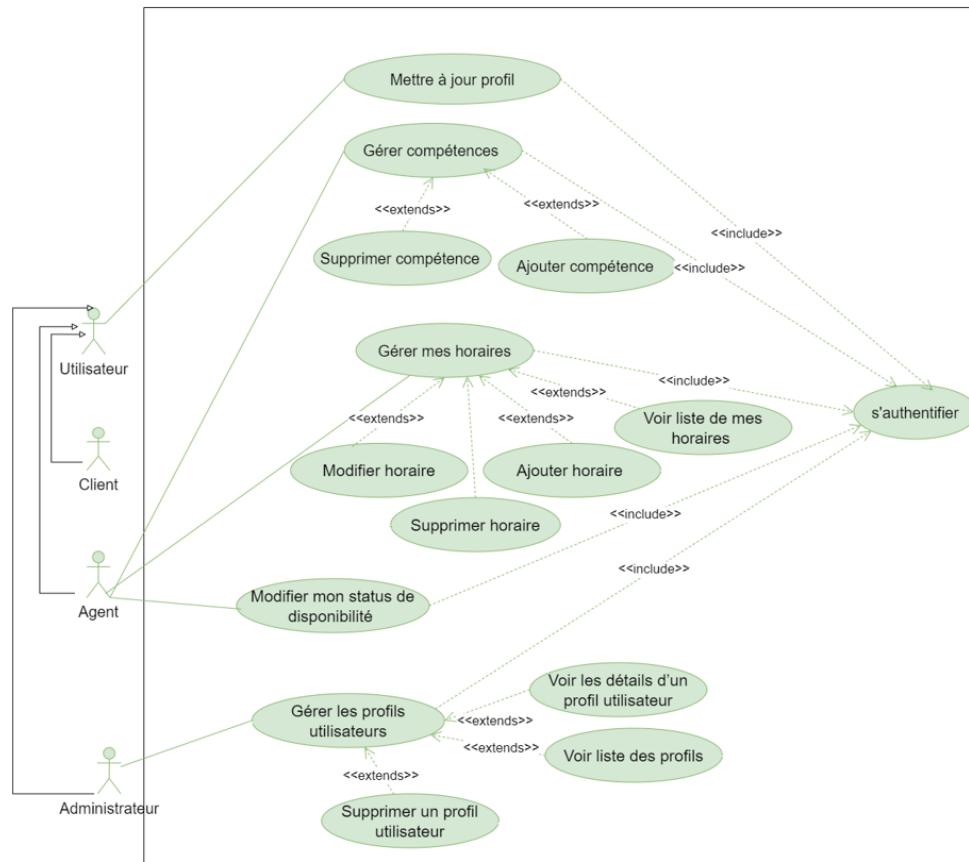


FIGURE 4.13 : Diagramme cas d'utilisation du sprint "Gestion du profil et des horaires".

4.2.4 Diagramme de Structure des Données sprint 4

La figure 4.14 représente le de structure des données du sprint "Gestion du profil et des horaires".

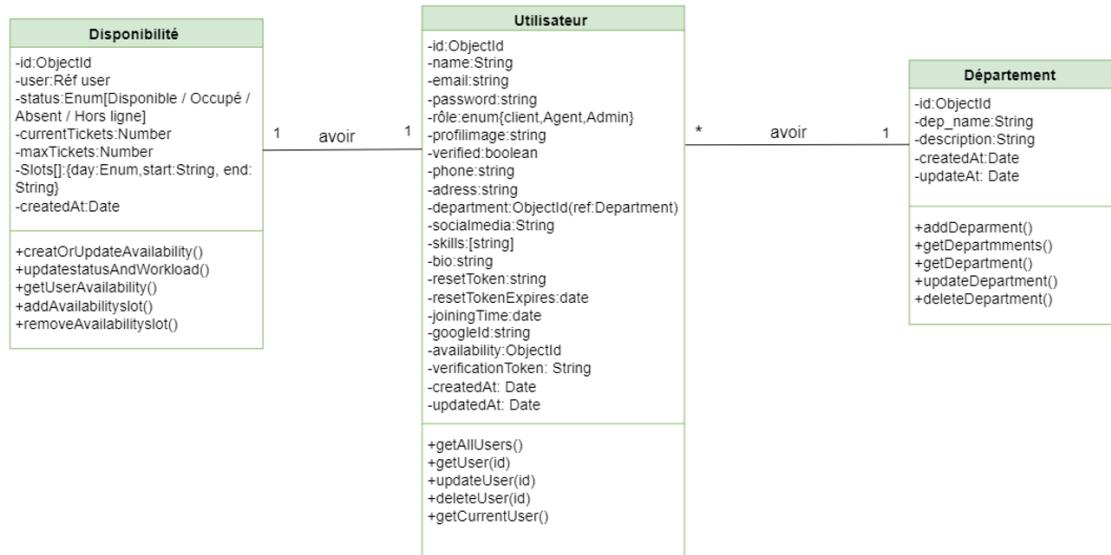


FIGURE 4.14 : Diagramme de Structure des Données sprint 4

4.2.5 Conception

4.2.5.1 Diagramme de séquence "Afficher profil"

La figure 4.15 représente le diagramme de séquence du cas d'utilisation "afficher profil".

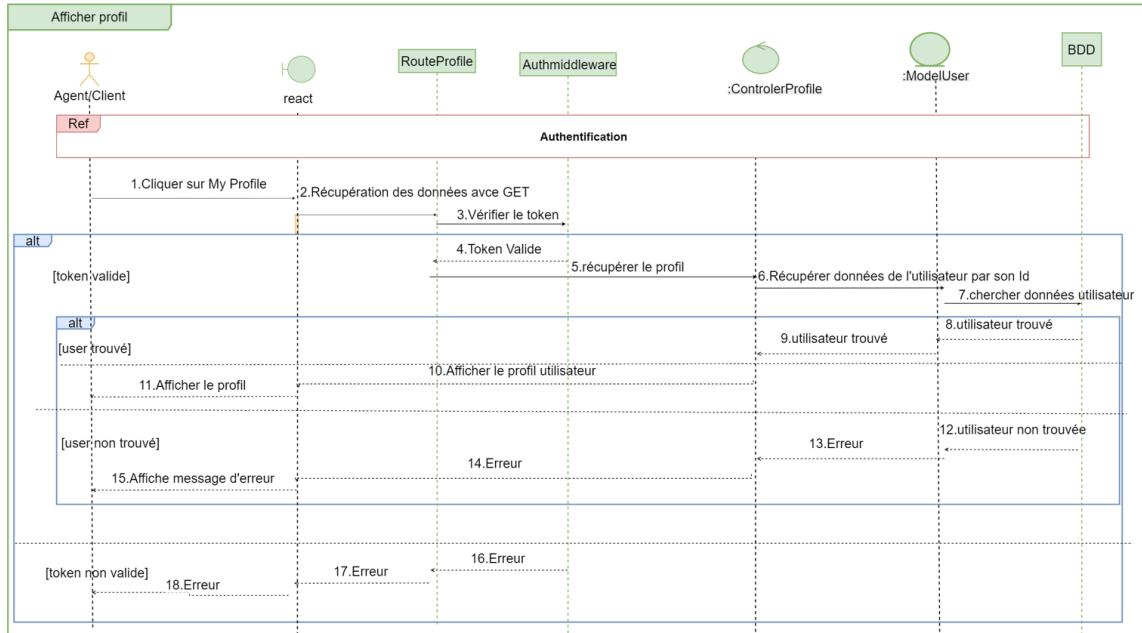


FIGURE 4.15 : Diagramme de séquence du cas d'utilisation "afficher profil"

4.2.5.2 Diagramme de séquence "Modifier profil"

La figure 4.16 représente le diagramme de séquence du cas d'utilisation "Modifier profil".

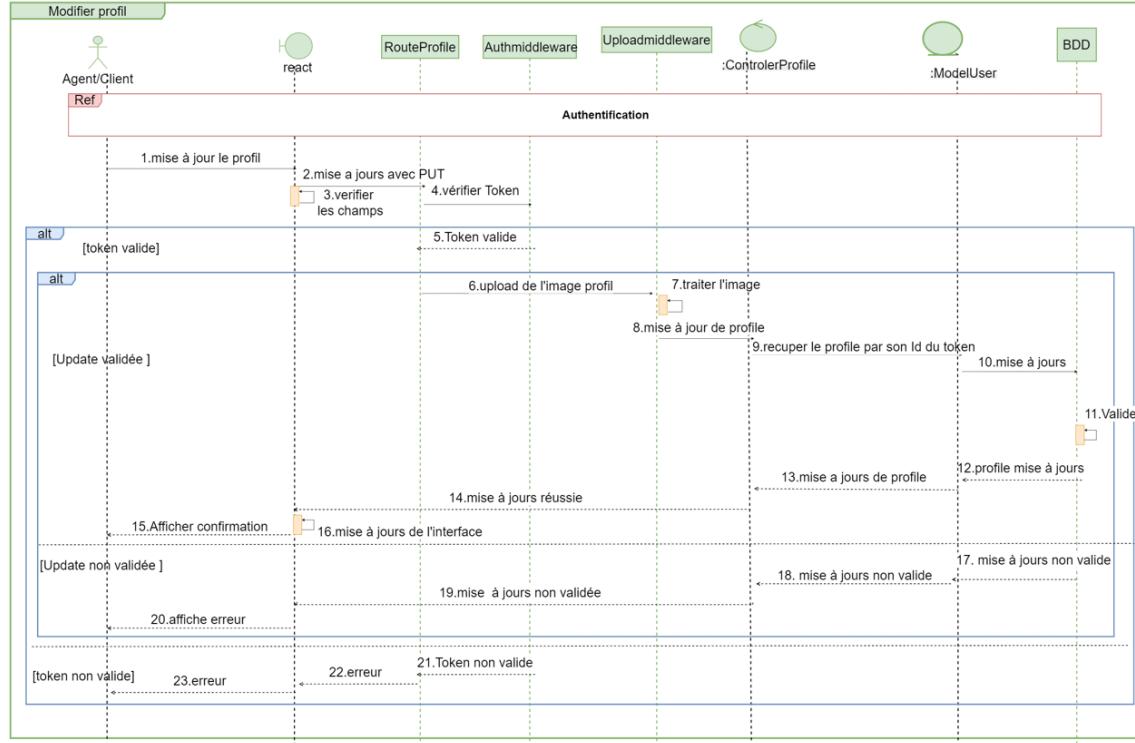


FIGURE 4.16 : Diagramme de séquence du cas d'utilisation "update profil".

4.2.5.3 Diagramme de séquence "Supprimer compétence"

La figure 4.17 représente le diagramme de séquence du cas d'utilisation "Supprimer compétence".

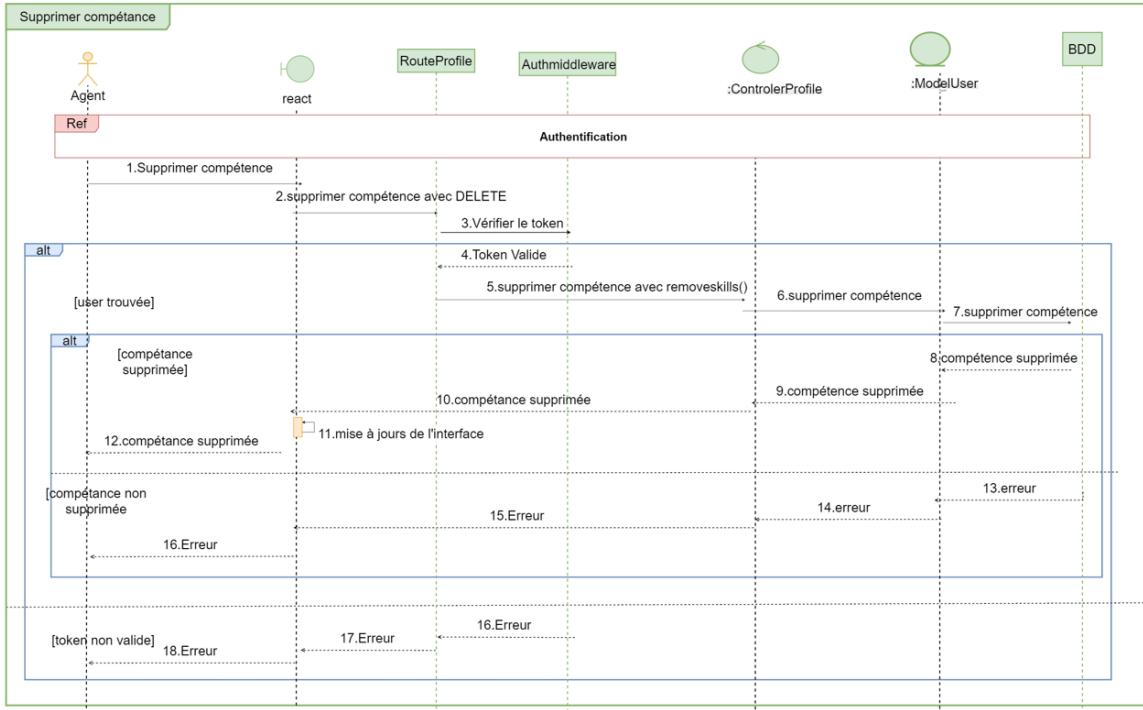


FIGURE 4.17 : diagramme de séquence du cas d'utilisation "Supprimer compétence"

4.2.5.4 Diagramme de séquence "Modifier horaires"

La figure 4.18 représente le diagramme de séquence du cas d'utilisation "Modifier horaires".

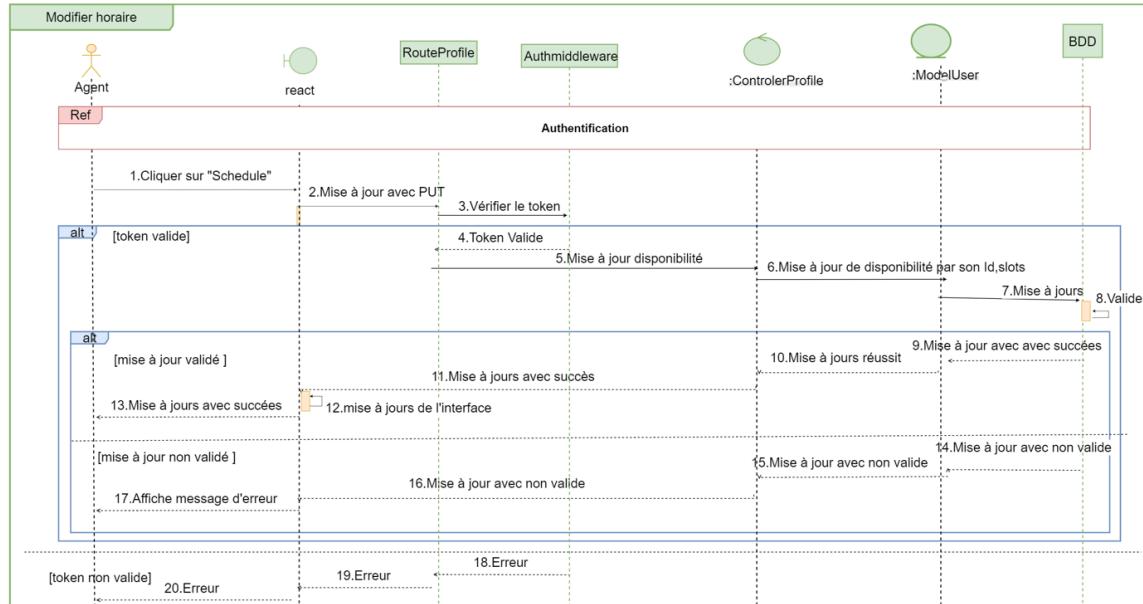


FIGURE 4.18 : Diagramme de séquence "Modifier horaire"

4.2.5.5 Diagramme de séquence "Supprimer horaires"

La figure 4.18 représente le diagramme de séquence du cas d'utilisation "Supprimer horaires".

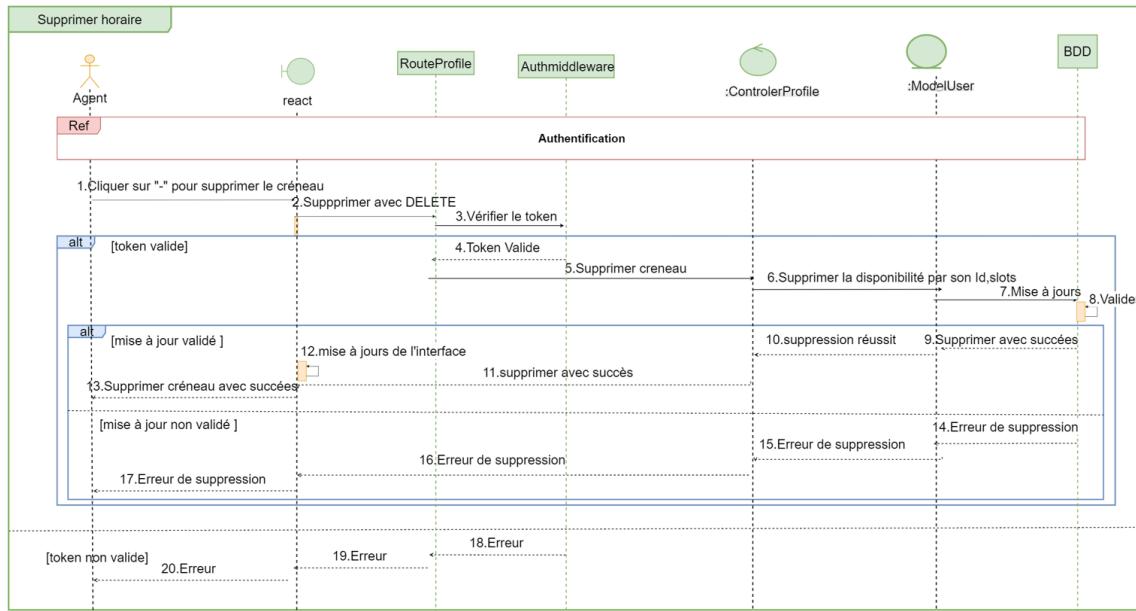


FIGURE 4.19 : Diagramme de séquence "Supprimer"

4.2.6 Réalisation

4.2.6.1 Gestion des profils et des horaires

Pour la réalisation du quatrième sprint, nous avons implémenté la gestion des profils et la gestion des horaires.

- Gestion du profil agent et de ses horaires

Nous avons commencé par nous connecter avec le compte d'un agent, puis avec celui d'un client. Les figures suivantes montrent que l'agent nommé "agenteya" dispose de champs supplémentaires pour la gestion de son profil, tels qu'une biographie professionnelle, des compétences, des liens vers ses profils LinkedIn et Twitter, ainsi que des informations de disponibilité. Ces éléments ne sont pas présents dans l'interface du client.

La figure ?? illustre la section dédiée aux informations du profil agent.

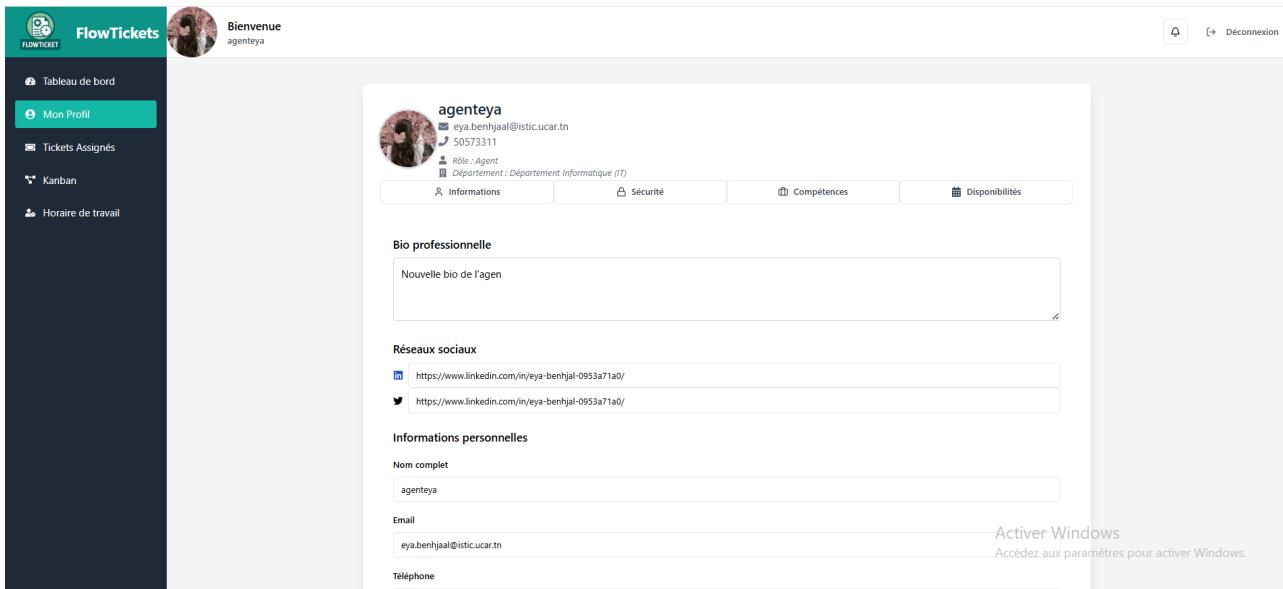


FIGURE 4.20 : Interface de profil agent partie "Informations"

La figure 4.21 montre que la section dédiée à la sécurité du profil agent permet de modifier son mot de passe.

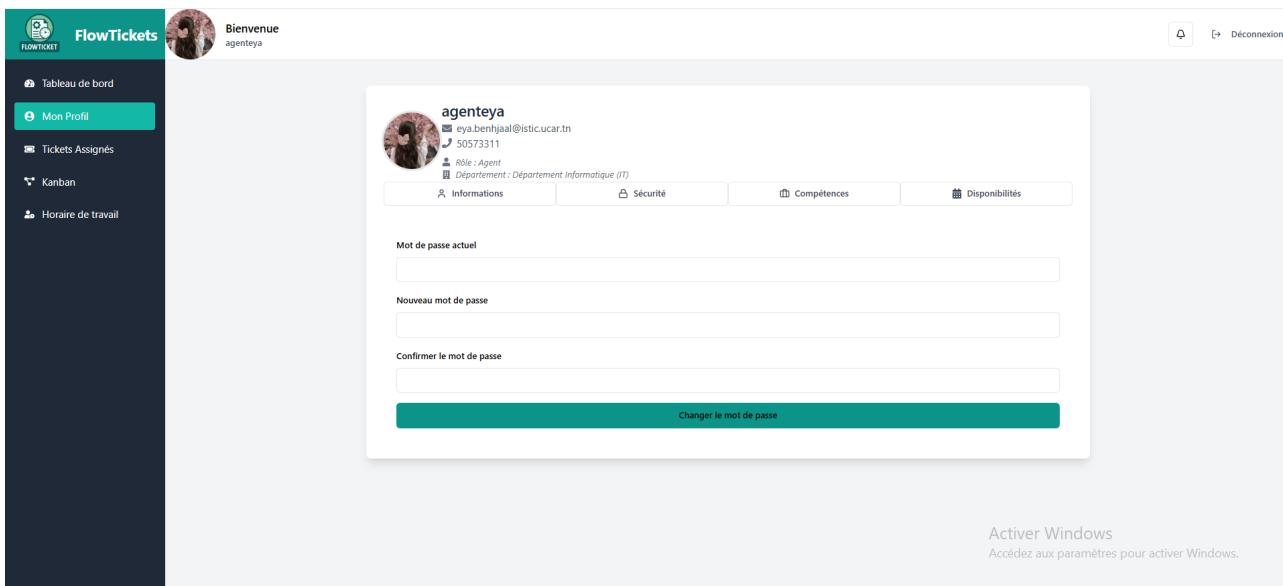


FIGURE 4.21 : Interface de profil agent partie "sécurité"

La figure 4.22 montre que la section dédiée aux compétences du profil agent permet d'ajouter ou de supprimer une compétence.

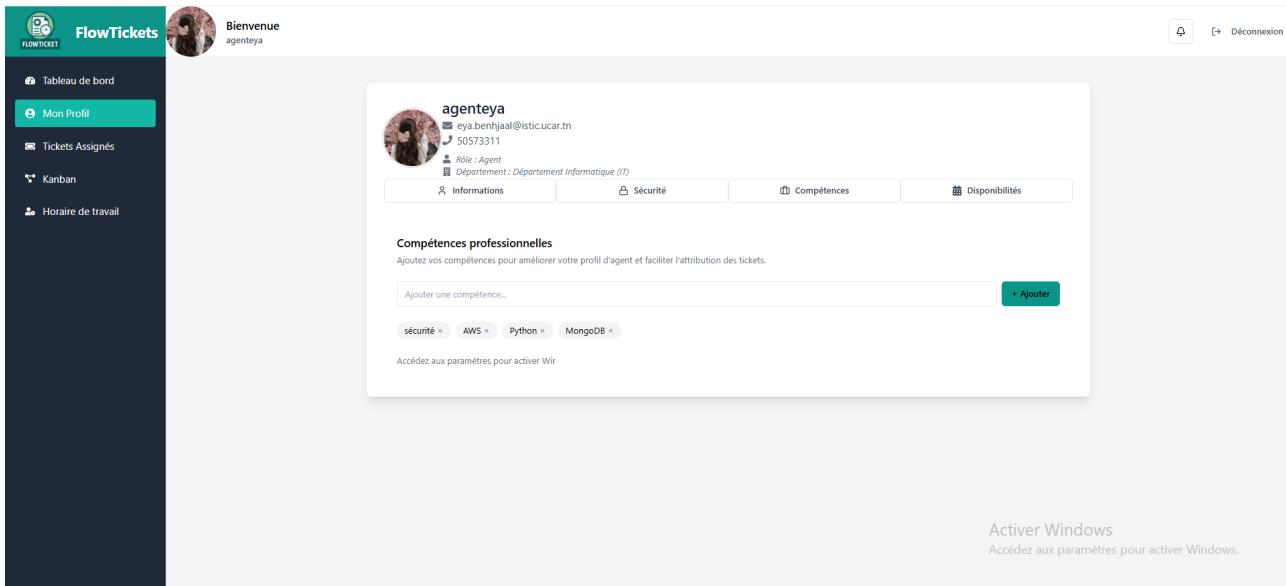


FIGURE 4.22 : Interface de profil agent partie "compétence"

La figure 4.23 montre que la section dédiée aux disponibilités du profil agent permet d'indiquer son statut (Disponible, Occupé, Absent, Hors ligne), de modifier sa capacité maximale, ainsi que de consulter ses tickets actuels.

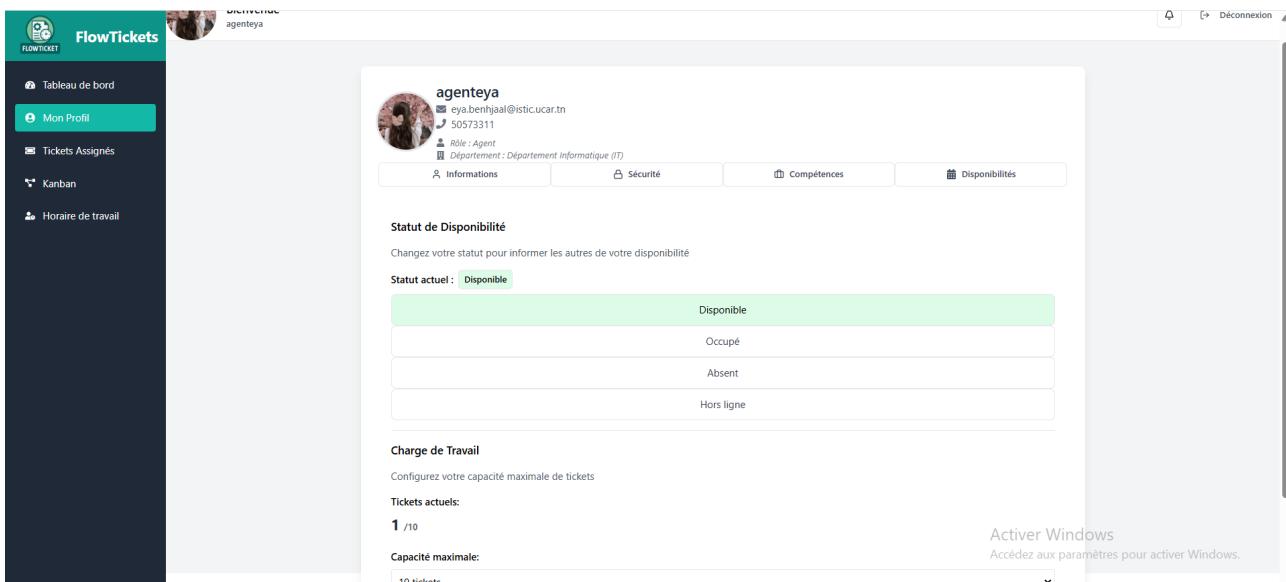


FIGURE 4.23 : Interface de profil agent partie disponibilités

La figure 4.24 montre que la section dédiée aux disponibilités permet à l'agent d'ajouter un horaire à l'aide du bouton « + », de le supprimer avec le bouton « - », et de modifier les horaires. Les interrupteurs « switchs » permettent d'activer ou de désactiver sa disponibilité pour chaque jour de la semaine.

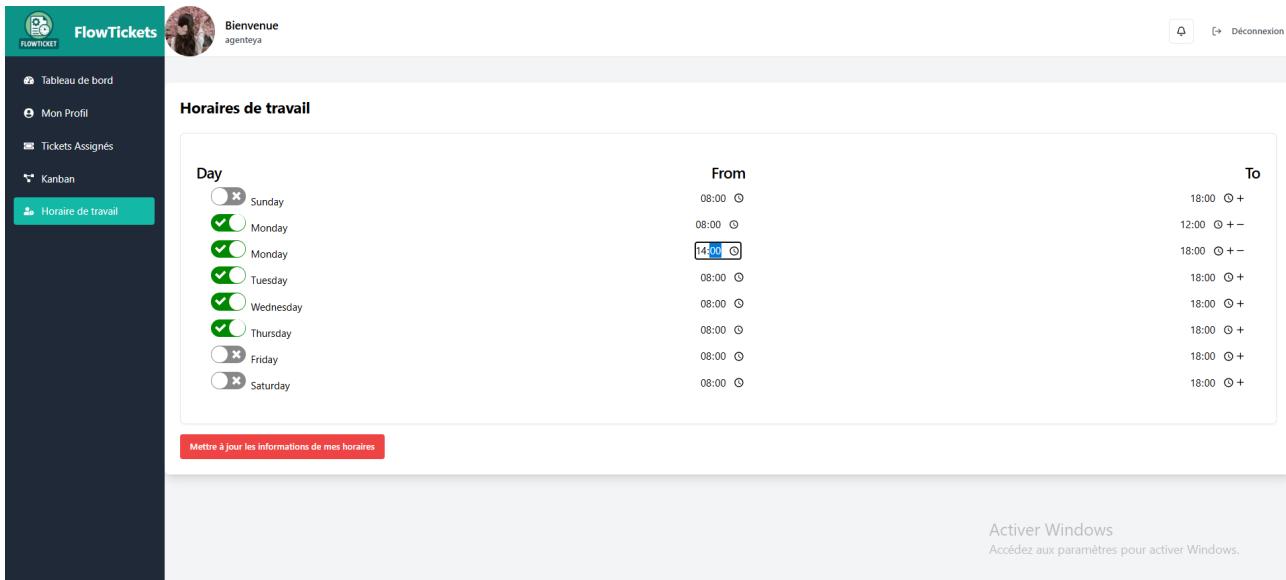


FIGURE 4.24 : Interface de gestion horaires

- Gestion du profil client La figure 4.25 illustre la section dédiée aux informations du profil client

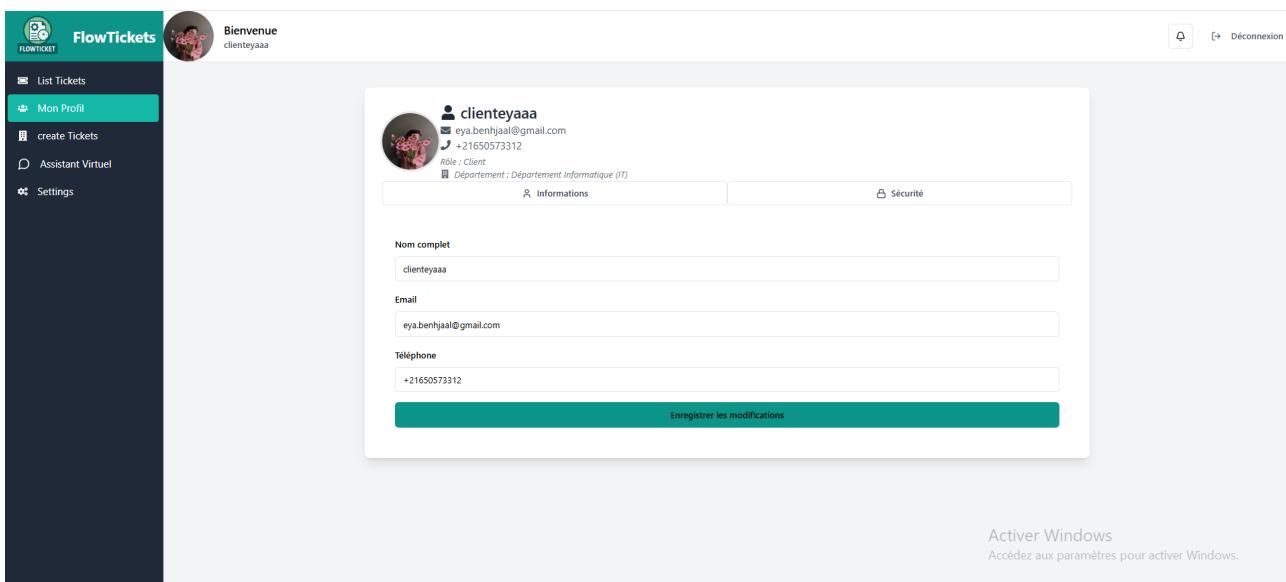


FIGURE 4.25 : Interface de profil client partie "Informations"

La figure 4.26 montre que la section dédiée à la sécurité du profil agent permet de modifier son mot de passe.

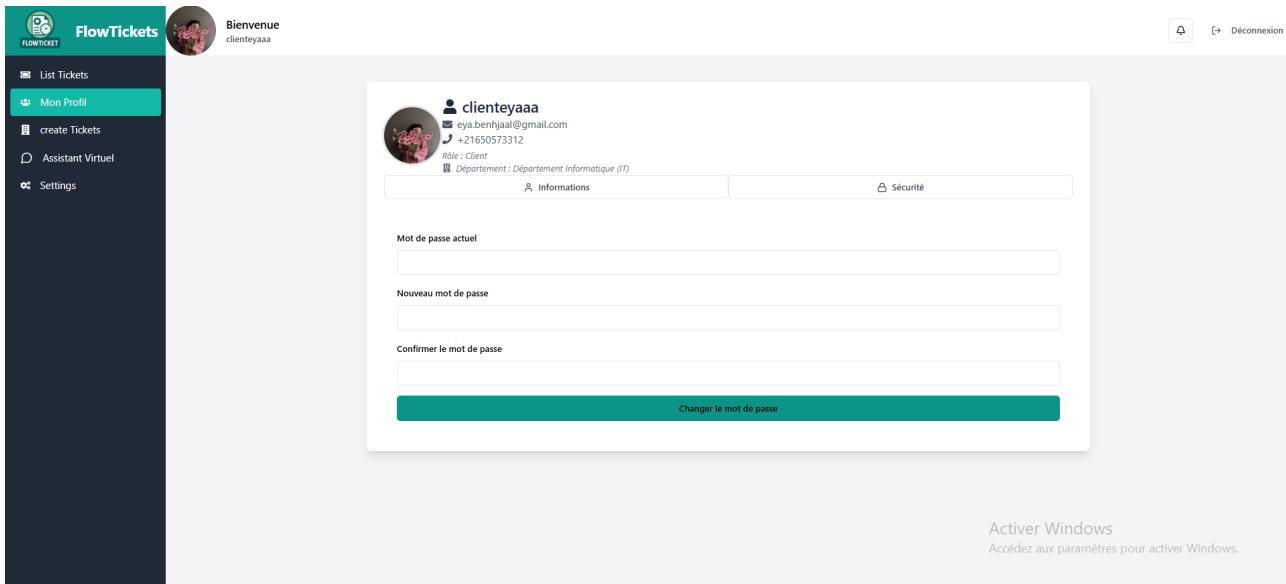


FIGURE 4.26 : Interface de profil client partie "Sécurité"

Conclusion

Au cours de ce chapitre, nous avons présenté la réalisation de la deuxième Release . Pour ce faire, nous avons passé par l'analyse, la conception et la réalisation des deux sprints "Gestion des utilisateurs et des rôles" et "Gestion du profil et des horaires".

z

Conclusion générale

Bibliographie

[B1] Pierre Pezziardi, Référentiel des Pratiques Agiles, édition ebook.2013

Webographie

- [1] « Postman. » [Accès le 26/04/2020]. (26/04/2020), adresse : <https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-api-web>.

Résumé

Le présent rapport synthétise le travail effectué dans le cadre du projet de fin d'études pour l'obtention du diplôme national d'ingénieur en informatique au sein de l'entreprise Xtendplex. L'objectif de ce travail est la conception et l'implémentation d'une application web de gestion des ressources humaines. Cette application consiste d'une part, de simplifier et d'accélérer le processus RH, et d'autre part, d'améliorer la communication entre les employés.

Mots clés : Ressources humaines, Spring Boot, Angular, Application web

Abstract

This document summarizes the work carried out as part of the end-of-study project for obtaining the national software engineering degree during the internship completed within the company Xtendplex. The main idea behind this project is to design and implement a Human Resources management web application. This web application is mainly used to accelerate the Human Resources process and improve the communication between the team members.

Keywords : Human ressource, Spring Boot, Angular, Web application