

**IIR**

## **INGÉNIERIE INFORMATIQUE & RÉSEAUX**

**ANNÉE UNIVERSITAIRE**

2022/2023

**OPTION**

**MÉTHODES INFORMATIQUES APPLIQUÉES  
À LA GESTION DES ENTREPRISES ( MIAGE )**

# **PROJET DE FIN D'ÉTUDES**

## **THÈME**

Analyse, conception, développement et déploiement de la plateforme de la ville intelligente pour la gestion des réclamations.

**RÉALISÉ PAR**

TIRGANI Badreddine

**ENCADRÉ PAR**

LAZAR Hajar

CHAKIB Hanaa

**SOUTENU LE**

18/09/2023

## Dédicace

*À l'âme de mon cher père, que Dieu ait son âme, je dédie ce projet avec une profonde gratitude. Ses valeurs et son amour inconditionnel ont toujours été une source d'inspiration pour moi. Sa bienveillance et ses conseils éclairés continuent de guider ma vie et mes choix.*

*Je souhaite également dédier ce projet à tous ceux qui m'ont soutenu et encouragé. À ma mère, dont l'amour inébranlable m'a permis de grandir et de me réaliser. À mon frère Saad, qui a toujours été là pour me conseiller et me soutenir durant mes études. À mes professeurs, dont la patience et la passion pour l'enseignement ont façonné ma formation. Et enfin, à mes amis et à tous ceux qui me sont chers, pour leur présence constante et leur support indéfectible.*

*J'espère sincèrement que ce travail sera à la hauteur des attentes de toutes ces personnes qui ont joué un rôle crucial dans ma vie. Que chaque lecteur en tire satisfaction et puisse trouver une source d'inspiration et de connaissance à travers ces pages.*

# **Remerciements**

*Avant de commencer ce rapport, je tiens tout d'abord à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué de manière directe ou indirecte à l'élaboration de ce projet.*

*Je souhaite exprimer ma sincère reconnaissance à ma famille pour leur soutien émotionnel inestimable tout au long de ma période de stage.*

*Mes remerciements s'adressent également à l'équipe pédagogique de l'EMSI, ainsi qu'à mes enseignants et aux professionnels intervenants responsables de ma formation.*

*Je tiens à exprimer ma gratitude envers mon encadrante, Mme Lazar, pour son aide, son soutien et ses précieux conseils concernant les missions abordées dans ce rapport.*

*En particulier, je voudrais exprimer ma profonde gratitude envers les personnes suivantes : M. Abdelkader, M. Sami et Mme Hanaa, pour leur soutien précieux et leur expertise tout au long de mon stage. Leurs conseils, leur disponibilité et leur confiance ont été des éléments cruciaux dans mon développement professionnel.*

# **Abstract**

The project presented in this document is part of my final internship in Information Engineering and Networks, specializing in MIAGE, at the Moroccan School of Engineers Sciences. I carried out this internship within Smart Way, a subsidiary of the IT Road Consulting group, from 06/03/2023 to 06/09/2023. This internship report outlines the various stages of design and implementation of the Smart City platform. This digital solution aims to improve the management of complaints and interventions in water and electricity distribution companies, within the framework of urban planning. This platform offers a user-friendly and intuitive interface that allows agents to quickly report issues such as power outages, water leaks, or drinking water cuts. Municipal authorities can thus efficiently handle complaints and resolve issues in a timely manner.

The goal of this application is to provide a maintainable, secure, and innovative application based on Spring Boot, Spring Data, Spring MVC, Spring Security, Java 17, React.js, JavaScript, deployed using the Docker virtualization system and Docker Compose, along with GitLab's Continuous Integration/Continuous Deployment (CI/CD) for development and deployment.

To achieve the expected results, our work process is based on agile development methods, using GitLab as a version control tool, enabling continuous delivery of features. We also use the JIRA project management platform for task distribution within the team and tracking the progress of each task.

**Keywords:** JIRA, management, Spring Boot, React Native, React Js, GitLab, JWT.

# Résumé

Le projet présenté dans ce document s'inscrit dans le cadre de mon stage de fin d'études en Ingénierie d'Information et Réseaux, option MIAGE, à l'École Marocaine des Sciences des Ingénieurs. J'ai effectué ce stage au sein de Smart Way, une filiale du groupe IT Road Consulting, du 06/03/2023 au 06/09/2023.

Ce rapport de stage présente les différentes étapes de conception et de réalisation de la plateforme Smart City. Cette solution numérique vise à améliorer la gestion des réclamations et des interventions dans les régies de distribution d'eau et d'électricité, dans le cadre de l'aménagement urbain. Cette plateforme offre une interface conviviale et intuitive permettant aux agents de signaler rapidement les problèmes tels que les pannes d'électricité, les fuites d'eau ou les coupures d'eau potable. Les autorités municipales peuvent ainsi gérer efficacement les plaintes et résoudre les problèmes dans les meilleurs délais.

L'objectif de cette application est de fournir une application maintenable, sécurisée et innovante basée sur Spring Boot, Spring Data, Spring MVC, Spring Security, Java 17, React.js, JavaScript, et déployée en utilisant le système de virtualisation Docker et Docker Compose, ainsi que le développement et le déploiement continu CI/CD de GitLab.

Pour atteindre les résultats attendus, notre processus de travail est basé sur les méthodes de développement agiles, en utilisant GitLab comme outil de contrôle, permettant une livraison continue des fonctionnalités. Nous utilisons également la plateforme de gestion de projet JIRA pour la distribution des tâches au sein de l'équipe et le suivi de l'avancement de chaque tâche.

Mots-clés : JIRA, gestion, Spring Boot, React native , React Js, gitlab, JWT.

# Glossaire

ALM	Gestion du Cycle de Vie des Applications (Application Lifecycle Management)
API	Interface de Programmation Applicative (Application Programming Interface)
BO	Bureau de Gestion (Back Office)
CI/CD	Intégration Continue / Déploiement Continu (Continuous Integration / Continuous Deployment)
JWT	Jeton Web JSON (JSON Web Token)
JS	Langage de Script JavaScript
MVC	Modèle-Vue-Contrôleur (Model-View-Controller)
RFC	Demande de Commentaires (Request for Comments)
UML	Langage de Modélisation Unifié (Unified Modeling Language)

# Table de matières

Dédicace .....	i
Remerciements .....	ii
Abstract .....	iii
Résumé .....	iv
Glossaire.....	v
Table de matières .....	vi
Liste des tableaux .....	vii
Liste des figures .....	viii
Introduction Générale.....	1
Chapitre 1 .....	2
Contexte Général.....	2
1.1    Introduction .....	3
1.2    Présentation de l'entreprise.....	3
1.3    Secteurs d'activité: .....	4
1.4    Structure organisationnelle.....	4
1.5    Présentation Du Projet.....	6
1.5.1.    Contexte et objectifs du projet.....	6
1.5.2.    Description de la Smart City plateforme .....	6
1.6    Objectifs du rapport.....	6
1.7    Identification des besoins des régies de distribution de l'eau et de l'électricité .....	10
1.8    Analyse des problèmes existants et des lacunes à combler.....	10

1.9	Spécifications des besoins .....	10
1.10	Besoins fonctionnels .....	11
1.11	Besoins non fonctionnels .....	15
1.12	Planification et Gestion du projet.....	16
1.12.1	Méthode Agile Scrum.....	16
1.12.2	Vision produit :.....	16
1.12.3	Le Backlog du produit Smart City.....	18
1.12.4	Planning meeting .....	18
1.12.5	Stand-up quotidien.....	19
1.13	Conclusion :.....	20
	Chapitre 2 .....	20
	Analyse et conception .....	20
2.1	Introduction .....	21
2.2	Diagramme de cas d'utilisation : .....	22
2.2.1	Diagramme de cas d'utilisation générale : .....	22
2.2.2	Diagrammes de cas d'utilisation pour l'acteur Membre et chef d'équipe :.....	23
2.2.3	Diagrammes de cas d'utilisation pour l'acteur Opérateur : .....	24
2.2.4	Diagrammes de cas d'utilisation pour l'acteur Ordonnanceur et Intermédiaire :.....	24
2.2.5	Diagramme de cas d'utilisation pour l'acteur Chef de départements :.....	25
2.2.6	Diagrammes de cas d'utilisation textuelles : .....	25
2.3	Diagrammes de séquence : .....	29
2.4	Diagrammes de Class : .....	32
2.5	Conclusion :.....	33
	Chapitre 3 .....	34

Etude technique .....	34
3.1    Introduction : .....	35
3.2    Architecture globale de la plateforme .....	35
3.3    Choix des technologies et justification des décisions prises .....	35
3.4    Modèle de conception MVC : .....	36
3.5    Technologies utilisées .....	37
Chapitre 4 .....	42
Réalisation.....	42
4.1    Interfaces graphiques : .....	43
Conclusion Générale .....	67
Références .....	68

# Liste des tableaux

Tableau 2.1: Diagramme textuel d'authentification .....	25
Tableau 2.2: Diagramme textuel gestion des interventions .....	27

# Liste des figures

Figure 1.1 IT Road Consulting.....	3
Figure 1.2 Projet Roadmap.....	7
Figure 1.3 Schéma des actions par profil .....	8
Figure 1.4 Schéma des modules de l'application.....	9
Figure 1.5 Méthode Scrum.....	16
Figure 1.6 Backloag du projet .....	17
Figure 1.7 Diagramme de Gantt.....	19
Figure 2.1 Cas d'utilisation générale .....	23
Figure 2.2: Cas d'utilisation pour l'acteur membre et chef d'équipe.....	23
Figure 2.3: Cas d'utilisation pour l'acteur Opérateur .....	24
Figure 2.4: Cas d'utilisation pour l'acteur Ordonnanceur et Intermédiaire .....	24
Figure 1.5: Cas d'utilisation pour l'acteur Chef de départements.....	25
Figure 2.6 : Diagramme de séquence d'authentification .....	29
Figure 2.7 : Diagramme séquence Ajouter Réclamation .....	30
Figure 2.8: Diagramme séquence Ajouter Intervention .....	31
Figure2.9 Diagramme de classe .....	32
Figure 3.1: Modèle MVC .....	36
Figure 3.2 : Logo springboot.....	37
Figure 3.3 : Logo ReactJs.....	38
Figure 3.4: Logo React native .....	38
Figure 3.5 Logo Docker .....	39

Figure3.6: Logo git.....	39
Figure 3.7: Logo Gitlab.....	40
Figure 3.8: Logo Jira .....	40
Figure 3.9: Json Web Token .....	41
Figure.4.1: Authentification page Login .....	43
Figure 4.2: Mot de passe oublié .....	44
Figure 4.3: Réception d'un e-mail de récupération de mot de passe suite à la validation de l'utilisateur .....	45
Figure 4.4:Dashboard .....	45
Figure 4.5: Administration Paramétrage Famille Type Famille .....	46
Figure 4.6: Ajouter Famille .....	47
Figure 4.7: Activé la modification des familles .....	47
Figure 4.8: Modifier une famille .....	48
Figure 4.9: Activer la modification des types pour une famille donne .....	48
Figure 4.10: Modification de Typé sélectionné .....	49
Figure 4.11: Suppression de la famille .....	50
Figure 4.12:Suppression d'un personnel .....	50
Figure 4.13: Ajouter un nouveau personnel .....	51
Figure 4.14: Lister les personnels .....	51
Figure 4.15: Filtrage des personnels par gérance .....	52
Figure 4.16: Filtrage des personnels par contenu exemple login nom rôle.....	53
Figure 4.17: Modification d'un personnel.....	53
Figure 4.18: Ajouter une nouvelle équipe .....	54
Figure 4.19: Modifier une équipe sélectionner .....	55
Figure 4.20: Lister les Equipe existants .....	55

Figure 4.21 : Ajouter un nouveau matériel ..... 56

Figure 4.22: Popup pour l'ajout du matériel ..... 56

Figure 4.23: Lister les matériels existants ..... 57

Figure 4.24: Modifier un Matériel..... 58

Figure 4.25: Filtrage du matériel par gérances..... 58

Figure 4.26: Ajouter une nouvelle déclamation par adresse ..... 59

Figure 4.27: Ajouter une nouvelle réclamation par Numéro du contrat étape 1 ..... 60

Figure 4.28: Ajouter une nouvelle réclamation par Numéro du contrat étape 2 ..... 60

Figure 4.29:Filtrer les réclamations par statuts ..... 61

Figure 4.30: Filtrage des réclamations existantes par statut..... 62

Figure 4.31: Suppression d'une réclamation donnée ..... 62

Figure 4.32: Ajouter plusieurs interventions dans une réclamations sans contrats..... 63

Figure 4.33: Ajouter plusieurs interventions dans une réclamations avec contrats ..... 63

Figure 4.34: Ajouter une nouvelle intervention dans une réclamation ..... 64

Figure 4.35:Modification d'une réclamation contenant le contrat client..... 65

Figure 4.36: Modification d'une réclamation sans contrat client..... 65

Figure 4.37: Lister les interventions par Gérances..... 66

# Introduction Générale

Le sujet de ce rapport porte sur la Smart City plateforme, une solution numérique novatrice qui vise à améliorer la gestion des réclamations et des interventions au sein des régies de distribution de l'eau et de l'électricité, dans le contexte de l'aménagement urbain. Cette solution offre une interface conviviale et intuitive permettant aux agents de signaler rapidement divers problèmes tels que les pannes électriques, les fuites d'eau ou encore les coupures d'eau potable. Grâce à cette solution, les autorités municipales peuvent gérer efficacement les plaintes et résoudre les problèmes dans les meilleurs délais.

Cette solution est hautement personnalisable, offrant ainsi aux autorités municipales la possibilité d'adapter facilement la plateforme en fonction de leurs besoins spécifiques. Les données sont stockées de manière sécurisée dans une base de données, garantissant ainsi la confidentialité et l'intégrité des informations.

Le présent rapport constitue une sorte de bilan de tout un travail de la période de stage pfe, Il se compose de 4 chapitres : le premier vise à présenter l'organisme d'accueil, ses activités, et une description du projet ainsi que la méthodologie de travail adopté. Dans le deuxième chapitre nous avons modéliser la structure et le comportement de système à travers les différents diagrammes (diagramme de cas d'utilisation, de séquence, de classe). Dans le troisième chapitre nous avons décrit l'architecture que nous avons adoptée, et l'environnement technique sur lequel nous avons travaillé. Le dernier chapitre présente l'ensemble des interfaces qui décrivent le fonctionnement de l'application.

# **Chapitre 1**

---

## **Contexte Général**

---

## **1.1Introduction**

Dans le cadre de ma dernière année du cycle d'ingénieur en informatique et réseau option MIAGE à l'école EMSI Marrakech aussi ma double diplomation M2 en Big Data Option MBDS à l'université cote d'azur Nice France j'ai effectué un stage de fin d'étude d'une durée de 6 mois

Ce rapport vise à explorer en détail le projet Smart City que j'ai réalisé au sein de l'entreprise It Road Group. L'objectif principal de ce projet était de concevoir et développer une plateforme Smart City complète. Cette plateforme offre une interface conviviale et intuitive pour gérer les réclamations et les interventions dans un environnement urbain. Les agents des régies autonomes ont la possibilité de signaler rapidement des problèmes tels que des pannes électriques, des fuites d'eau ou des coupures d'eau potable. Grâce à cette solution, les autorités municipales sont en mesure de gérer de manière efficace les plaintes et de résoudre les problèmes dans les plus brefs délais, contribuant ainsi à améliorer la qualité de vie des citoyens.

## **1.2Présentation de l'entreprise**

IT Road Consulting est une société de conseil spécialisée dans les technologies de l'information et l'innovation. Nous travaillons en étroite collaboration avec nos clients pour les accompagner dans leur transformation numérique et les aider à tirer parti des opportunités offertes par les nouvelles technologies. (1)



**Figure 1.1 IT Road Consulting**

### **1.3 Secteurs d'activité:**

Le groupe IT Road est actif dans plusieurs secteurs d'activité, notamment :

**Conseil en technologies de l'information** : IT Road fournit des services de conseil stratégique et technologique aux entreprises pour les aider à optimiser leurs systèmes d'information, à mettre en œuvre des solutions innovantes et à améliorer leur performance.

**Développement de logiciels** : IT Road se spécialise dans le développement de logiciels sur mesure pour répondre aux besoins spécifiques de ses clients. Cela comprend la conception, le développement, le déploiement et la maintenance de solutions logicielles personnalisées.

**Services d'infrastructure informatique** : IT Road propose des services d'infrastructure informatique complets, tels que la gestion des serveurs, le stockage des données, la virtualisation, la sécurité réseau et la gestion des systèmes d'exploitation.

**Solutions de sécurité informatique** : IT Road offre des solutions de sécurité informatique pour protéger les données et les systèmes de ses clients contre les cyberattaques. Cela comprend la gestion des pare-feux, la détection des intrusions, la gestion des identités et des accès, et la sensibilisation à la sécurité.

**Intégration de systèmes** : IT Road aide les entreprises à intégrer leurs systèmes informatiques existants, à assurer l'interopérabilité des applications et à optimiser leurs processus métier grâce à des solutions d'intégration adaptées.

**Services cloud** : IT Road propose des services de cloud computing, tels que l'infrastructure en tant que service (IaaS), la plateforme en tant que service (PaaS) et le logiciel en tant que service (SaaS), permettant aux entreprises de bénéficier de l'évolutivité, de la flexibilité et de la disponibilité des solutions cloud.

### **1.4 Structure organisationnelle**

La structure organisationnelle du groupe IT Road peut varier en fonction de sa taille, de ses filiales et de ses divisions. Cependant, voici une structure organisationnelle typique qui peut être observée :

- Direction générale : La direction générale est responsable de la vision stratégique, de la prise de la décision globale et de la supervision de l'ensemble du groupe IT Road.
- Divisions ou départements fonctionnels : Le groupe IT Road peut être organisé en différentes divisions ou départements fonctionnels, tels que :
  - Département des ventes et du marketing : Responsable de la promotion des services du groupe, de l'acquisition de nouveaux clients et de la gestion des relations avec les clients existants.
  - Département de la gestion de projet : Responsable de la planification, de l'exécution et de la gestion des projets clients, en veillant au respect des délais, des budgets et des objectifs fixés.
  - Département de la recherche et développement : Responsable de l'innovation technologique, de la recherche de nouvelles solutions, du développement de nouveaux produits ou services et de l'amélioration continue des offres existantes.
  - Département des opérations : Responsable de la gestion des opérations quotidiennes, de la prestation des services, de la gestion des infrastructures et des systèmes, et de la supervision des équipes techniques.
  - Département des ressources humaines : Responsable de la gestion des ressources humaines, du recrutement, de la formation, de la gestion des talents et des relations employés.
  - Département des finances : Responsable de la gestion financière du groupe, y compris la comptabilité, la planification budgétaire, la gestion des risques et le reporting financier.

- Filiales et divisions spécifiques : Selon la structure du groupe IT Road, il peut y avoir des filiales ou des divisions spécifiques qui se concentrent sur des secteurs d'activité spécifiques ou des régions géographiques particulières. Chaque filiale peut avoir sa propre structure organisationnelle interne, dirigée par un responsable ou un directeur.

Cette structure organisationnelle est généralement conçue pour faciliter la coordination, la communication et la prise de décision efficace au sein du groupe IT Road, tout en permettant une spécialisation et une expertise dans différents domaines d'activité.

## **1.5 Présentation Du Projet**

### **1.5.1. Contexte et objectifs du projet**

Dans le cadre de notre engagement envers le développement des villes intelligentes, nous avons entrepris la conception et le développement d'une solution numérique innovante appelée Smart City plateforme. Cette plateforme a pour objectif d'améliorer la gestion des réclamations et des interventions dans les régies de distribution de l'eau et de l'électricité, contribuant ainsi à l'aménagement urbain durable et à la satisfaction des citoyens.

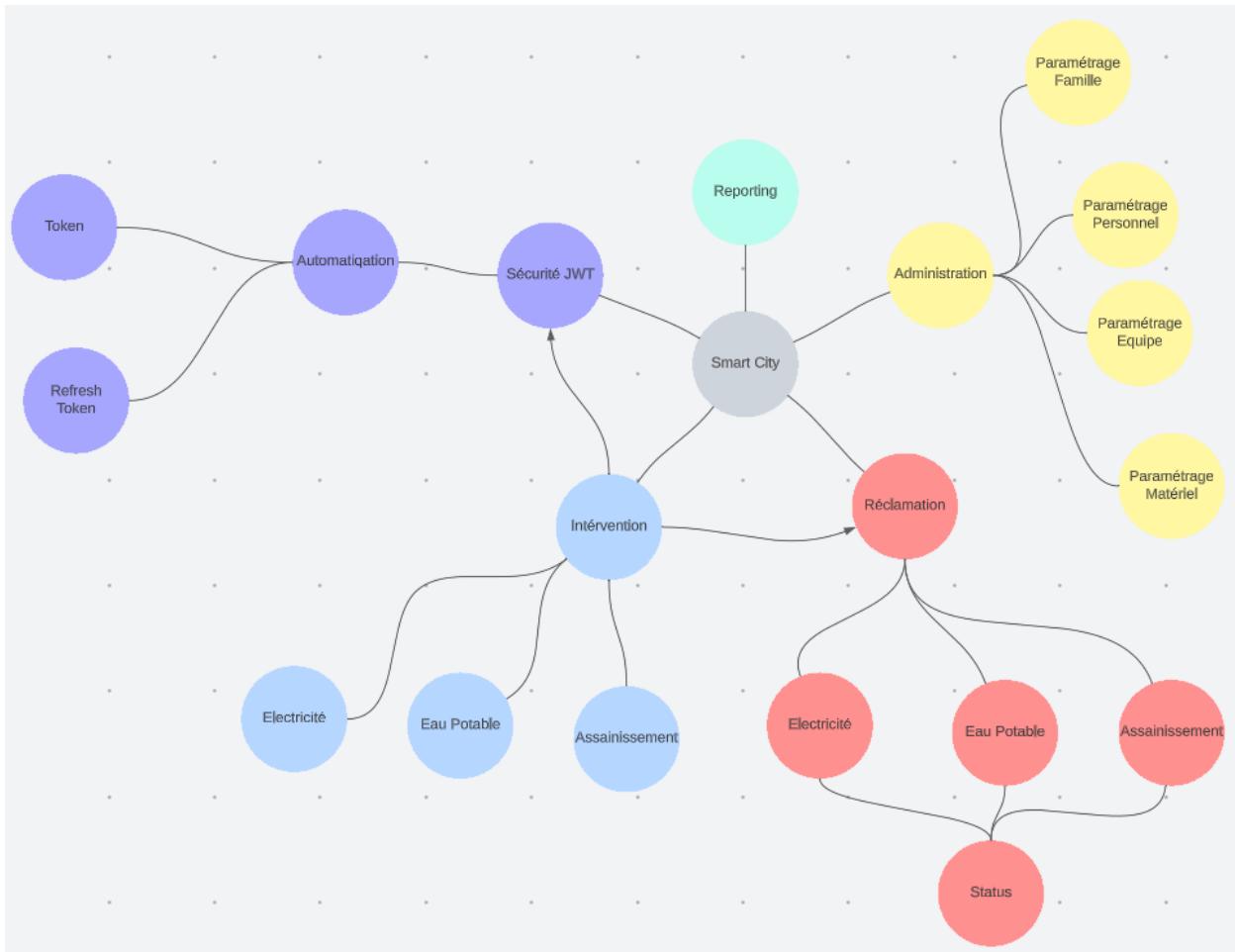
### **1.5.2. Description de la Smart City plateforme**

La Smart City plateforme est une solution numérique complète qui offre une interface conviviale et intuitive pour la gestion des réclamations et des interventions. Elle permet aux agents des régies autonomes de signaler rapidement les problèmes tels que les pannes d'électricité, les fuites d'eau ou les coupures d'eau potable. Les autorités municipales peuvent ainsi gérer efficacement les plaintes et résoudre les problèmes dans les meilleurs délais, améliorant ainsi la qualité de vie des citoyens.

## **1.6 Objectifs du rapport**

L'objectif de ce rapport est d'examiner de manière approfondie la création et l'évolution de la plateforme Smart City par IT Road Consulting. Nous allons plonger dans l'analyse des exigences spécifiques des services de distribution, la conception de la solution, ainsi que le processus de développement en exploitant des technologies avancées telles que Java 17, Spring Boot, React JS

et React Native. De plus, nous allons explorer la mise en œuvre de la plateforme en détaillant l'infrastructure déployée ainsi que les outils de gestion de version et de collaboration qui ont été astucieusement mis en place.



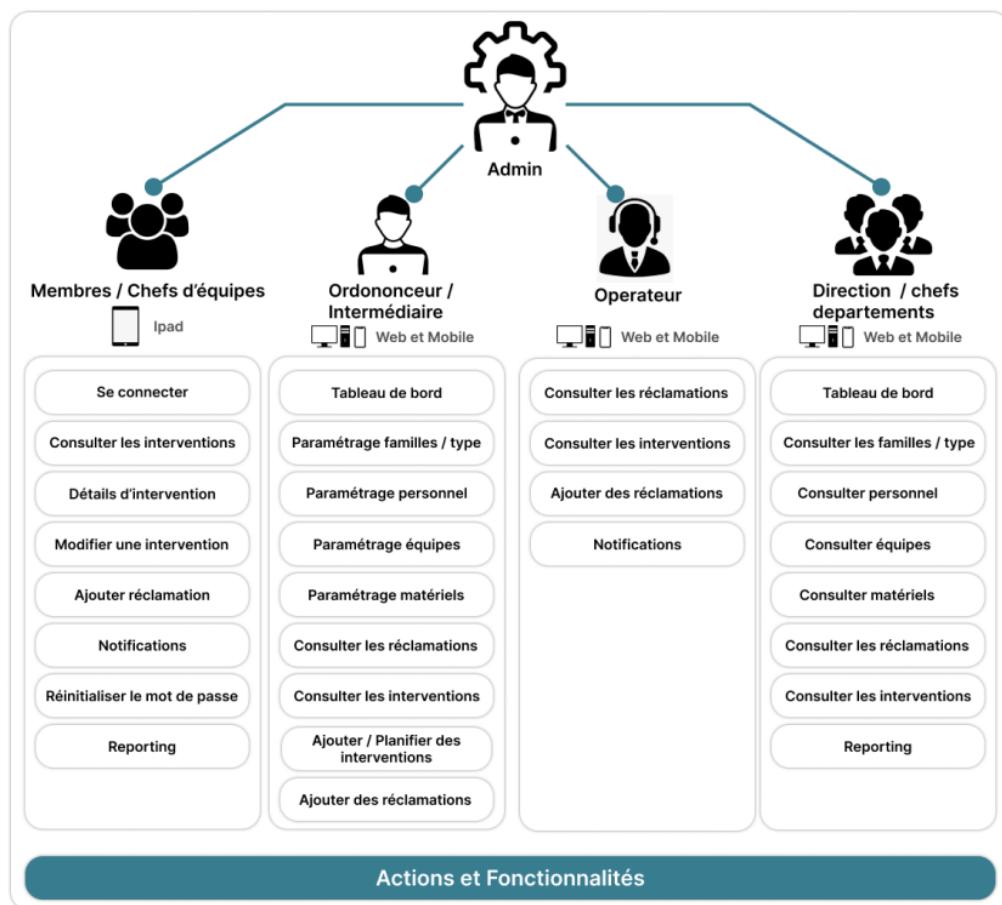
**Figure 1.2 Projet Roadmap**

Le système est doté de différentes fonctionnalités réparties en fonction des rôles des utilisateurs. Les membres et les chefs d'équipes, ont la capacité de se connecter, de consulter les interventions, d'accéder aux détails d'intervention, de modifier les interventions existantes, d'ajouter des réclamations, de recevoir des notifications, de réinitialiser leur mot de passe, et de générer des rapports pour une gestion plus efficace.

Les ordonnanceurs et les intermédiaires, bénéficient d'un tableau de bord pour une vue d'ensemble. Ils peuvent également paramétrier les familles/types, le personnel, les équipes et les matériels. En outre, ils ont la possibilité de consulter les réclamations, les interventions, de planifier de nouvelles interventions, d'ajouter des réclamations, et de rester informés grâce aux notifications.

Les opérateurs, ont des rôles plus axés sur la gestion des réclamations et des interventions. Ils peuvent consulter les réclamations, les interventions, ajouter de nouvelles réclamations, et recevoir des notifications pour une réactivité optimale.

Finalement, la direction et les chefs de départements, profitent d'un tableau de bord global et peuvent consulter les familles/types, le personnel, les équipes, les matériels, ainsi que les réclamations et les interventions. Ils ont également la possibilité de générer des rapports pour prendre des décisions éclairées. Ce système offre une gamme complète d'outils pour satisfaire les besoins de chaque rôle au sein de l'organisation.

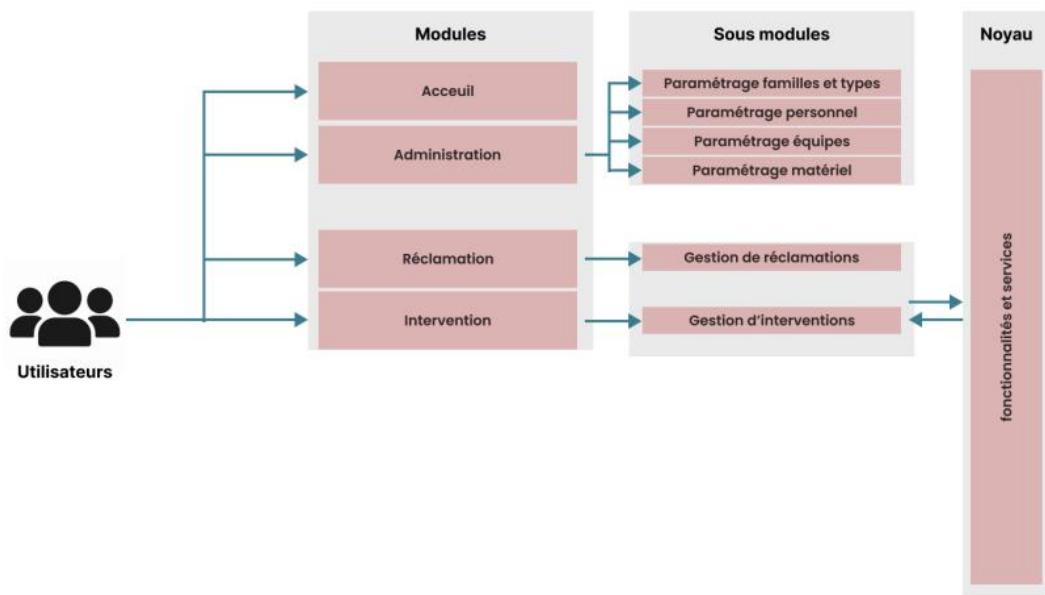


**Figure 1.3 Schéma des actions par profil**

Le projet est composé de 10 modules, regroupés en différentes catégories. Dans la première catégorie, "Administration", nous avons les modules de paramétrage de la famille, du personnel, de l'équipe et du matériel. Ces modules sont essentiels pour configurer et gérer les informations de base du projet.

La deuxième catégorie concerne les réclamations par gérance, avec des modules dédiés à l'électricité, à l'eau potable et à l'assainissement. Ces modules permettent de suivre et de gérer les réclamations liées à ces services, assurant ainsi une meilleure gestion et résolution des problèmes.

Enfin, la troisième catégorie comprend les modules d'interventions par gérance, également axés sur l'électricité, l'eau potable et l'assainissement. Ces modules sont conçus pour faciliter la gestion des interventions techniques nécessaires dans ces domaines.



**Figure 1.4 Schéma des modules de l'application.**

## **1.7 Identification des besoins des régies de distribution de l'eau et de l'électricité**

L'analyse des besoins vise à comprendre les exigences spécifiques des régies de distribution de l'eau et de l'électricité en termes de gestion des réclamations et des interventions. Cette section se concentrera sur l'identification des fonctionnalités clés requises par les régies pour améliorer leur efficacité opérationnelle et répondre aux attentes des citoyens. Des entretiens avec les parties prenantes, l'étude de la législation en vigueur et l'analyse des bonnes pratiques seront effectués pour établir une liste exhaustive des besoins.

## **1.8 Analyse des problèmes existants et des lacunes à combler**

Certaines lacunes subsistent dans l'application, notamment en ce qui concerne la partie mobile ainsi que la sécurisation de l'accès aux différents services de l'application.

## **1.9 Spécifications des besoins**

Afin d'adopter une approche ciblée et efficiente, l'équipe de développement du projet Smart City a été segmentée en trois entités distinctes : la conception, le développement et le déploiement continu. J'ai été désigné pour superviser le développement de toutes les facettes du projet. Ma principale responsabilité réside dans l'élaboration de la composante frontale en utilisant React.js, garantissant ainsi une interface utilisateur réactive, conviviale et attrayante. En parallèle, il m'incombe également de gérer le développement de l'infrastructure backend en utilisant Spring et Java 17. Mon objectif majeur consiste à mettre en place les fonctionnalités essentielles pour assurer le parfait fonctionnement du système.

De plus, J'ai déployé le projet en utilisant Docker. Cette technologie de virtualisation facilite le déploiement, la gestion des conteneurs et garantit une meilleure portabilité de l'application.

## 1.10 Besoins fonctionnels

### - Système d'authentification

La première fonctionnalité essentielle de notre projet est la mise en place d'un système d'authentification sécurisé. Ce système permet aux utilisateurs d'accéder en toute sécurité à leurs comptes, de gérer leurs informations personnelles et de bénéficier d'une expérience d'authentification fluide.

Pour cela, nous utilisons Spring Security, une solution robuste et largement utilisée pour la gestion de l'authentification et de l'autorisation dans les applications Java. Spring Security nous permet de mettre en place des mécanismes avancés tels que la gestion des espaces par rôle.

En utilisant Spring Security, nous pouvons attribuer des rôles spécifiques aux utilisateurs et définir les autorisations correspondantes pour chaque rôle. Il est également possible de permettre à un utilisateur d'avoir plusieurs rôles, offrant ainsi une flexibilité accrue dans la gestion des autorisations.

Cette fonctionnalité d'authentification sécurisée est cruciale pour garantir la confidentialité et l'intégrité des données des utilisateurs, ainsi que pour assurer une expérience utilisateur personnalisée en fonction des rôles et des autorisations attribués.

### - Gestion des utilisateurs

La gestion des utilisateurs est une fonctionnalité essentielle du système, permettant aux administrateurs de gérer efficacement les utilisateurs dans le frontend. En implémentant ces fonctionnalités, le système offre une interface conviviale et sécurisée pour la gestion des utilisateurs, simplifiant ainsi l'administration et le contrôle des accès.

Les fonctionnalités clés de la gestion des utilisateurs comprennent :

**Ajout de nouveaux utilisateurs** : Les administrateurs peuvent ajouter de nouveaux utilisateurs en fournissant les informations requises telles que le nom, l'adresse e-mail et le rôle associé.

**Attribution des rôles** : Les administrateurs peuvent affecter des rôles spécifiques aux utilisateurs pour définir leurs autorisations et leurs niveaux d'accès.

**Affichage de la liste des utilisateurs** : Une liste des utilisateurs enregistrés est affichée, fournissant leurs informations essentielles.

**Recherche et filtrage des utilisateurs** : Les administrateurs peuvent effectuer des recherches et appliquer des filtres pour trouver rapidement des utilisateurs spécifiques en fonction de critères tels que le nom ou le rôle.

**Modification des informations des utilisateurs** : Les administrateurs ont la possibilité de modifier les informations des utilisateurs existants, comme le nom ou l'adresse e-mail.

**Suppression des utilisateurs** : Les administrateurs peuvent supprimer des utilisateurs du système. Une confirmation est généralement demandée avant la suppression pour éviter les suppressions accidentnelles.

Ces fonctionnalités permettent une gestion efficace des utilisateurs, offrant aux administrateurs un contrôle total sur les accès et les informations des utilisateurs. Cela garantit une interface conviviale et sécurisée pour la gestion des utilisateurs dans le frontend du système.

#### - **Gestion des familles et des types de famille :**

Le système comprend également la gestion des familles, des types de famille et des équipes. Ces fonctionnalités permettent une organisation et une gestion efficace des données liées aux familles et aux équipes dans le système.

**Ajout de familles** : Les administrateurs peuvent ajouter de nouvelles familles en fournissant les informations nécessaires telles que le nom de la famille, l'adresse, les membres de la famille, etc.

**Attribution de types de famille** : Les familles peuvent être catégorisées en fonction de différents types, tels que famille monoparentale, famille nombreuse, etc. Les administrateurs peuvent attribuer des types de famille spécifiques à chaque famille en fonction de leurs caractéristiques.

**Affichage et filtrage des familles** : Une liste des familles enregistrées est affichée, permettant aux utilisateurs de visualiser et de filtrer les familles en fonction de critères spécifiques, tels que le type de famille ou l'adresse.

### - Gestion des équipes :

**Création d'équipes** : Les administrateurs peuvent créer des équipes en spécifiant un nom d'équipe, une description et les membres qui y sont associés.

**Gestion des membres de l'équipe** : Les administrateurs ont la possibilité d'ajouter ou de supprimer des membres de l'équipe, d'attribuer des rôles spécifiques aux membres de l'équipe et de gérer les autorisations de chaque membre.

**Affichage des équipes** : Une liste des équipes enregistrées est affichée, fournissant des informations telles que le nom de l'équipe, la description et les membres associés, le superviseur, le chef d'équipe la gérance ....

### - Gestion des matériels :

La gestion des matériels englobe plusieurs fonctionnalités clés pour permettre aux utilisateurs de gérer efficacement les équipements et les ressources matérielles. Voici les principales fonctionnalités incluses :

**Ajout de matériels** : Les utilisateurs peuvent ajouter de nouveaux matériels en renseignant les informations nécessaires telles que le nom, la description, le numéro de série, etc.

**Modification des matériels** : Les utilisateurs ont la possibilité de modifier les informations des matériels existants, que ce soit pour mettre à jour les détails, les attributs ou les caractéristiques.

**Filtrage des matériels** : Le système offre la possibilité de filtrer les matériels en fonction de critères spécifiques tels que la catégorie, l'état, la gérance, ou tout autre attribut pertinent. Cela facilite la recherche et la visualisation des matériels correspondants aux besoins de l'utilisateur.

**Suppression logique** : Au lieu de supprimer définitivement les matériels, le système utilise une suppression logique qui permet de les marquer comme étant supprimés tout en conservant les données associées. Cette approche permet de récupérer facilement les matériels supprimés si nécessaire.

**Gestion des gérances** : Les matériels peuvent être attribués à des gérances spécifiques, permettant ainsi une organisation efficace et un suivi précis de l'affectation des équipements.

#### - **Gestion des réclamations :**

La partie de gestion des réclamations comprend plusieurs fonctionnalités pour permettre aux utilisateurs de filtrer, ajouter, modifier et supprimer des réclamations, ainsi que d'ajouter des interventions associées. Voici les détails de ces fonctionnalités :

**Filtrage des réclamations** : Les utilisateurs ont la possibilité de filtrer les réclamations en fonction de critères tels que la gérance, le statut ou le contenu de la réclamation. Cela permet de visualiser facilement les réclamations correspondant aux besoins spécifiques de l'utilisateur.

**Ajout d'une réclamation** : Les utilisateurs peuvent ajouter une réclamation en utilisant soit le numéro de contrat associé, soit l'adresse concernée. Cela permet de créer une nouvelle réclamation en fournissant les informations pertinentes telles que la description du problème, la date, etc.

**Modification et suppression** : Les utilisateurs ont la possibilité de modifier les détails d'une réclamation existante, tels que la description, la date ou d'autres informations pertinentes. De plus, au lieu de supprimer définitivement une réclamation, le système utilise une suppression logique pour la marquer comme supprimée tout en conservant les données associées. Cela permet de restaurer facilement une réclamation supprimée si nécessaire.

**Ajout de plusieurs interventions** : Les utilisateurs peuvent ajouter plusieurs interventions à une réclamation spécifique. Cela permet de suivre et de gérer les différentes étapes de résolution de la réclamation, en fournissant des informations sur les actions entreprises, les dates, les responsables, etc.

Ces fonctionnalités de gestion des réclamations permettent aux utilisateurs de filtrer, ajouter, modifier et supprimer des réclamations, ainsi que d'ajouter des interventions associées. Cela facilite la gestion efficace des réclamations, en garantissant un suivi précis, une communication transparente et une résolution appropriée des problèmes signalés.

## **1.11 Besoins non fonctionnels**

En plus des besoins fonctionnels, il est important de prendre en compte les besoins non fonctionnels dans la gestion des réclamations. Voici quelques exemples de besoins non fonctionnels importants à considérer :

**Sécurité** : Assurer la confidentialité, l'intégrité et la sécurité des données liées aux réclamations, en mettant en place des mesures de protection telles que l'authentification, l'autorisation, le cryptage des données sensibles, etc.

**Performance** : Garantir des temps de réponse rapides et une haute disponibilité du système, en optimisant les performances du système, en gérant efficacement les requêtes et en évitant les goulots d'étranglement.

**Convivialité** : Offrir une interface utilisateur conviviale, intuitive et ergonomique pour faciliter la saisie des réclamations, la navigation dans le système et la visualisation des informations.

**Scalabilité** : Assurer que le système puisse gérer un grand nombre de réclamations et d'utilisateurs, en permettant l'expansion et l'adaptation du système selon les besoins futurs.

**Fiabilité** : Garantir un fonctionnement fiable du système, en minimisant les temps d'arrêt, en implémentant des sauvegardes régulières des données et en prévoyant des mécanismes de récupération en cas de défaillance.

**Intégration** : Permettre l'intégration avec d'autres systèmes ou services pertinents, tels que des systèmes de gestion de l'inventaire, des systèmes de facturation, etc., pour assurer la cohérence des données et faciliter les processus métier.

**Évolutivité** : Permettre une évolutivité du système, en facilitant l'ajout de nouvelles fonctionnalités, l'adaptation aux changements technologiques et l'intégration de nouvelles exigences métier.

**Accessibilité** : Assurer l'accessibilité du système pour les utilisateurs ayant des besoins spécifiques, en respectant les normes d'accessibilité et en fournissant des fonctionnalités telles que le support pour les lecteurs d'écran, les options de contraste, etc.

En tenant compte de ces besoins non fonctionnels, le système de gestion des réclamations peut offrir une expérience utilisateur optimale, tout en répondant aux exigences de sécurité, de performance, de convivialité et d'intégrité des données.

## 1.12 Planification et Gestion du projet

Pour bien structurer et assurer le bon déroulement d'un projet, il faut adopter une démarche pour gérer son cycle de vie. Alors dans cette partie, nous allons décrire la méthode adoptée lors du développement de cette application.

### 1.12.1 Méthode Agile Scrum

Dans le cadre du projet Smart City, nous avons choisi de travailler selon la méthode Agile Scrum, qui favorise la flexibilité, la collaboration et l'itération continue. Cette approche nous permet de mieux répondre aux exigences changeantes du projet et d'assurer une livraison régulière de fonctionnalités.

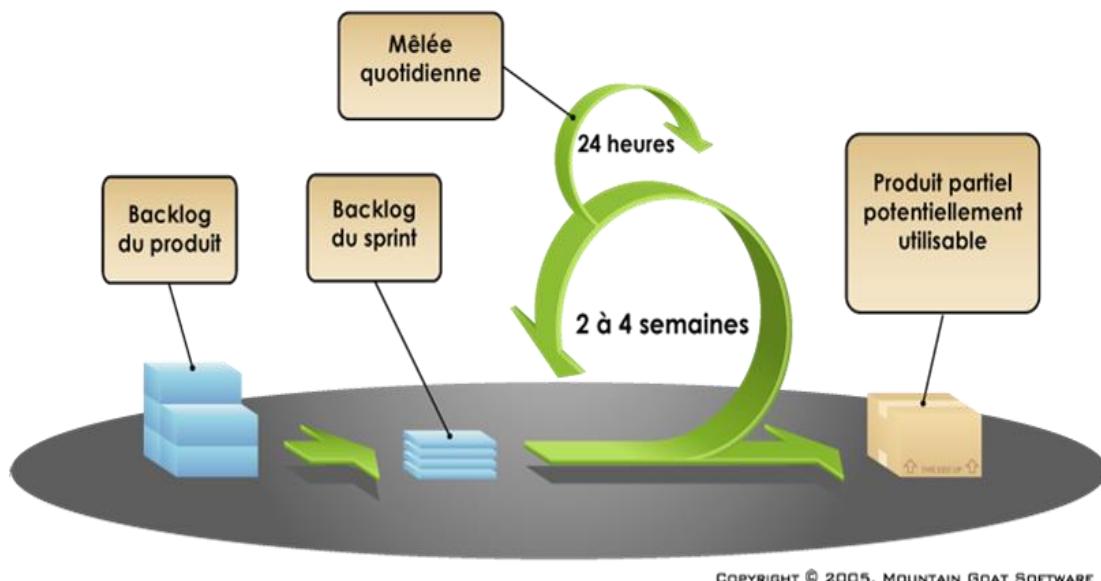


Figure 1.5 Méthode Scrum

### 1.12.2 Vision produit :

La vision du produit Smart City est de créer une solution complète et innovante pour améliorer la qualité de vie et la durabilité dans les villes. Notre produit vise à intégrer les dernières technologies disponibles pour offrir une plateforme intelligente et connectée, capable de gérer efficacement les différents aspects de la vie urbaine.

Notre objectif est de permettre aux autorités municipales, aux services publics et aux citoyens de bénéficier des avantages de la transformation numérique dans la gestion des services urbains tels que l'énergie, l'eau, les transports, les déchets, la sécurité, etc.

Nous visons à fournir des fonctionnalités avancées telles que la surveillance en temps réel, l'analyse des données, la gestion des ressources, la planification intelligente, la communication bidirectionnelle avec les citoyens, et bien plus encore. Notre produit vise également à favoriser la participation citoyenne et à promouvoir la durabilité environnementale.

En offrant une interface conviviale et intuitive, notre produit Smart City permettra aux utilisateurs d'accéder facilement aux informations et aux services pertinents, favorisant ainsi une meilleure prise de décision et une gestion efficace des ressources.

Dans l'ensemble, notre vision du produit Smart City est de transformer les villes en des environnements plus intelligents, plus durables et plus agréables à vivre, en exploitant le potentiel des dernières technologies pour répondre aux besoins des citoyens et des autorités municipales.

The screenshot shows the Jira Software interface for the 'Smart City' project. The left sidebar includes navigation links for 'Smart City' (Projet logiciel), 'PLANIFICATION' (Chronologie, Backlog, Tableau), and 'DÉVELOPPEMENT' (Code, Pages de projet, Ajouter un raccourci, Paramètres du projet). The main area displays a Kanban board titled 'Projets / Smart City'. The board has five columns: 'Todo', 'In Progress', 'Blocked', 'Testing', and 'Review/QA'. Each column contains several items, each with a small icon, a title, and a brief description. The 'Tableau' link in the sidebar is highlighted, indicating the current view. The top navigation bar includes links for 'Votre travail', 'Projets', 'Filtres', 'Tableaux de bord', 'Équipes', 'Appli', and 'Créer'. A search bar and various icons are also present at the top right.

**Figure 1.6 Backlog du projet**

### **1.12.3 Le Backlog du produit Smart City**

Le Backlog du produit Smart City est une liste ordonnée des fonctionnalités, des améliorations et des tâches à réaliser pour atteindre la vision du produit. Voici quelques exemples d'éléments qui pourraient figurer dans le Backlog :

1. Système d'authentification et de gestion des utilisateurs
2. Gestion des réclamations avec filtrage par gérance, statut et contenu
3. Ajout de réclamations par numéro de contrat ou par adresse
4. Modification et suppression (soft delete) des réclamations
5. Ajout de plusieurs interventions à une réclamation
6. Gestion des familles et des types de famille
7. Gestion des équipes avec attribution des rôles
8. Gestion des matériels avec ajout, modification et filtrage
9. Rapports et analyses des données liées aux réclamations et aux matériels
10. Optimisation des performances du système pour une meilleure réactivité
11. Intégration avec des systèmes externes tels que la facturation ou la gestion de l'inventaire
12. Amélioration de l'interface utilisateur pour une meilleure convivialité
13. Renforcement des mesures de sécurité pour assurer la confidentialité des données
14. Mise en place d'un processus de sauvegarde régulière des données
15. Évaluation et sélection des technologies pour le déploiement en utilisant Docker

Ce Projet est évolutif et sera continuellement mis à jour en fonction des priorités, des retours des utilisateurs et de l'évolution des besoins. Il permet de garder une trace organisée des éléments à réaliser et d'assurer une progression cohérente vers la réalisation de la vision du produit Smart City.

### **1.12.4 Planning meeting**

Les réunions de planification sont des événements clés dans la méthodologie Agile Scrum utilisée pour le projet Smart City. Ces réunions sont organisées avant le début de chaque sprint et impliquent l'équipe de développement, le chef de projet et les parties prenantes. L'objectif

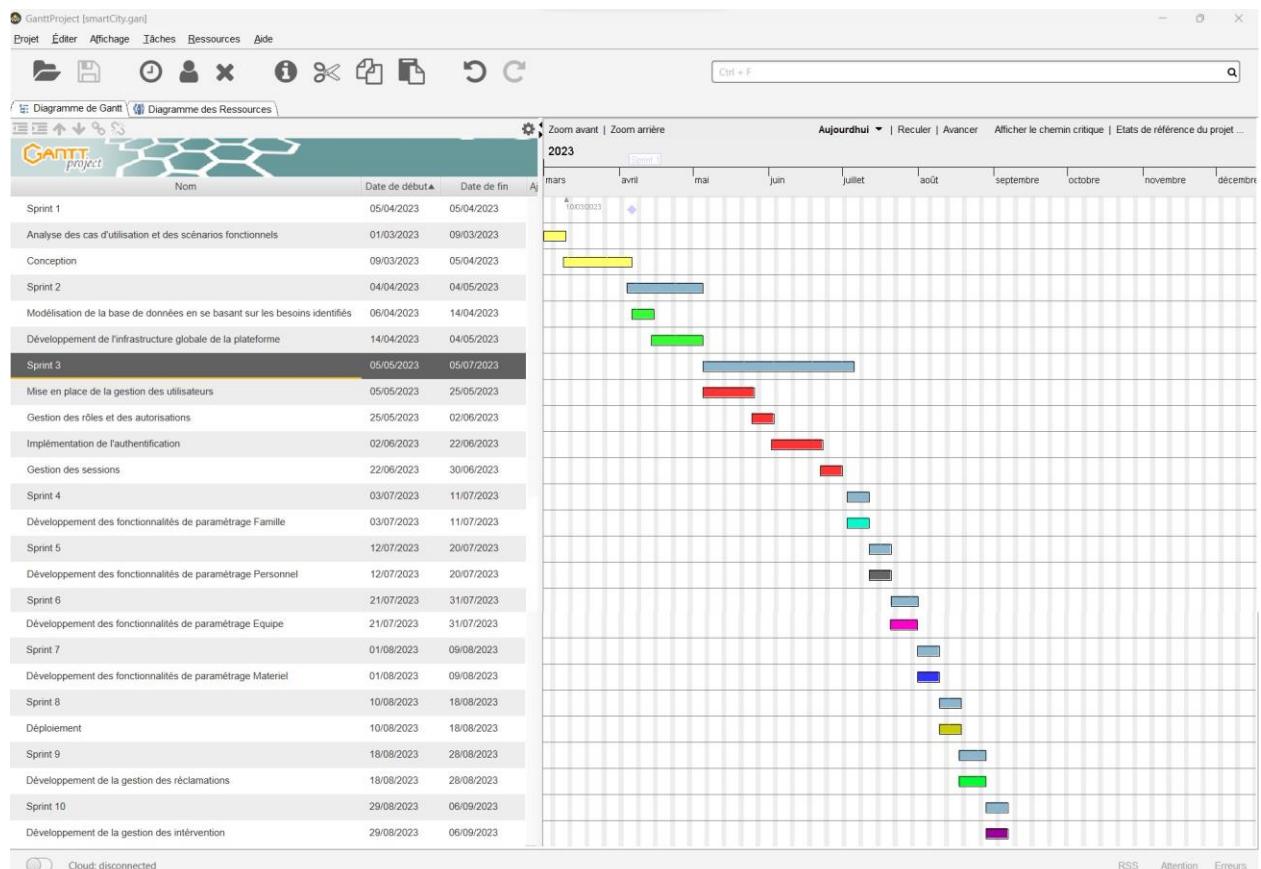
Principal de ces réunions est de définir les objectifs spécifiques du sprint, de discuter et d'évaluer les user stories, de répartir les tâches et de créer un plan de travail détaillé pour le sprint à venir.

### 1.12.5 Stand-up quotidien

Les réunions de stand-up quotidiennes sont des réunions courtes et régulières qui ont lieu tous les jours ouvrables. Ces réunions sont organisées pour permettre à l'équipe de partager rapidement les progrès réalisés depuis la dernière réunion, les tâches sur lesquelles chacun travaille et les éventuels obstacles rencontrés. Les réunions de stand-up favorisent la communication, la coordination et la résolution rapide des problèmes, permettant ainsi à l'équipe de rester alignée et de suivre l'avancement du projet de manière efficace.

Sprints :

Voici une proposition de séquence de sprints en prenant en compte les différentes étapes du projet Smart City :



**Figure 1.7 Diagramme de Gantt**

### **1.13 Conclusion :**

En résumé, l'approche itérative et bien organisée des sprints prévus pour le projet Smart City permet à notre équipe de développement de progresser méthodiquement. Chacun des sprints se concentre sur des objectifs spécifiques et des fonctionnalités clés à élaborer, ce qui favorise une avancée graduelle. Cette stratégie nous permet de relever avec succès les défis uniques de chaque sprint et de nous adapter efficacement aux évolutions des exigences du projet.

## **Chapitre 2**

---

### **Analyse et conception**

---

## **2.1Introduction**

Dans ce chapitre, nous présenterons l'analyse et la conception de l'application. Nous utiliserons des outils tels que le langage de modélisation UML pour décrire de manière claire et compréhensible les solutions adoptées pour résoudre la problématique et répondre aux besoins du système. L'utilisation d'UML nous permettra de créer différents types de diagrammes qui décrivent la structure et le comportement du système ainsi que des objets qui y sont impliqués. Les diagrammes que nous utiliserons sont les suivants :

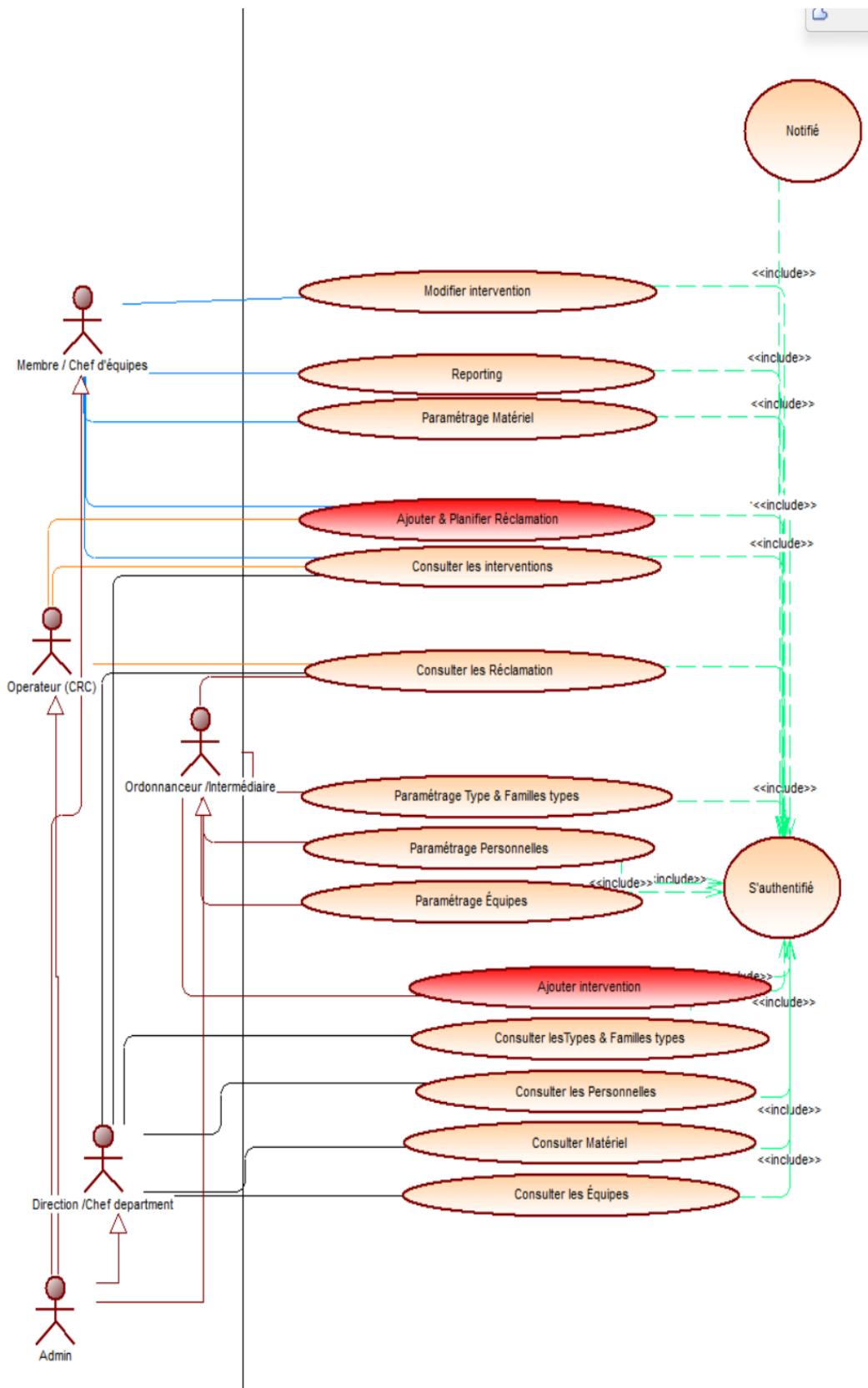
- **Diagramme de cas d'utilisation** : il permet de visualiser les interactions entre les acteurs et les fonctionnalités du système.
- **Description textuelle des cas d'utilisation** : il fournit une description détaillée des actions et des interactions dans un langage textuel.
- **Diagramme de séquence** : il représente l'ordre chronologique des messages échangés entre les objets lors de l'exécution d'un scénario.
- **Diagramme de classe** : il montre la structure statique du système en identifiant les classes, les attributs et les relations entre les objets.
- **Diagramme d'activité** : il présente le flux d'activités et de décisions dans un processus ou un scénario.

Grâce à ces diagrammes, nous pourrons mieux comprendre et représenter les différentes facettes de l'application, ce qui facilitera le développement et la mise en œuvre des fonctionnalités nécessaires.

## 2.2 Diagramme de cas d'utilisation :

### 2.2.1 Diagramme de cas d'utilisation générale :

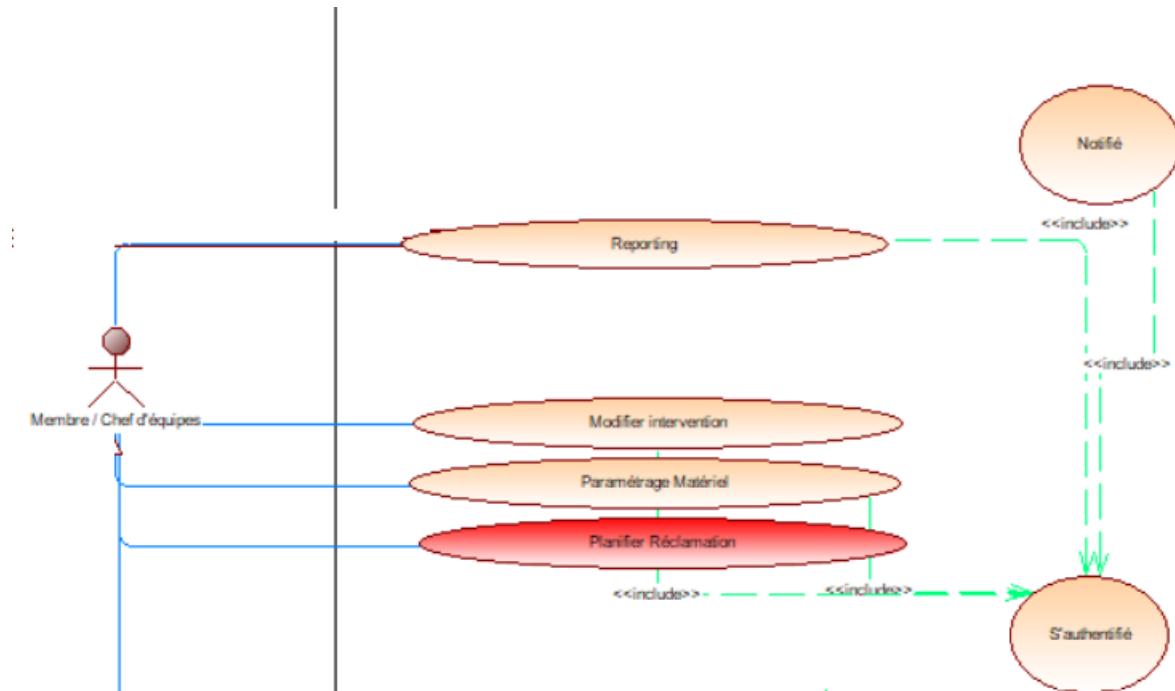
La figure suivante présente le diagramme de cas d'utilisation général.



**Figure 2.1 Cas d'utilisation générale**

## 2.2.2 Diagrammes de cas d'utilisation pour l'acteur Membre et chef d'équipe :

La figure ci-dessous illustre le diagramme de cas d'utilisation pour les acteurs "Membre" et "Chef d'équipe".



**Figure 2.2: Cas d'utilisation pour l'acteur membre et chef d'équipe**

### 2.2.3 Diagrammes de cas d'utilisation pour l'acteur Opérateur :

La figure ci-dessous illustre le diagramme de cas d'utilisation pour l'acteurs "Operateur Centre de relation Clients"

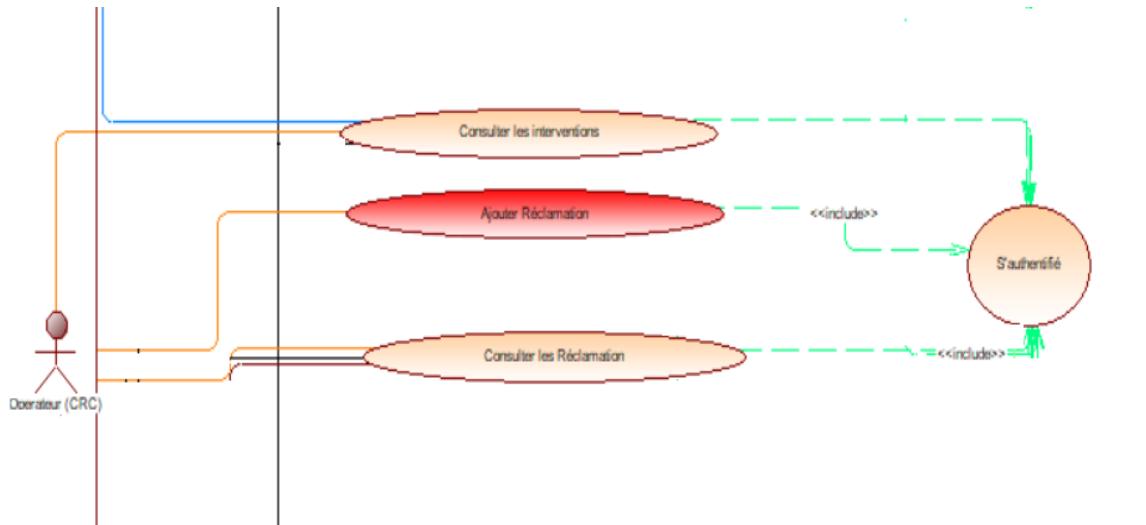


Figure 2.3: Cas d'utilisation pour l'acteur Opérateur

### 2.2.4 Diagrammes de cas d'utilisation pour l'acteur Ordonnanceur et Intermédiaire :

La figure ci-dessous illustre le diagramme de cas d'utilisation pour les acteurs " Ordonnanceur " et " Intermédiaire ".

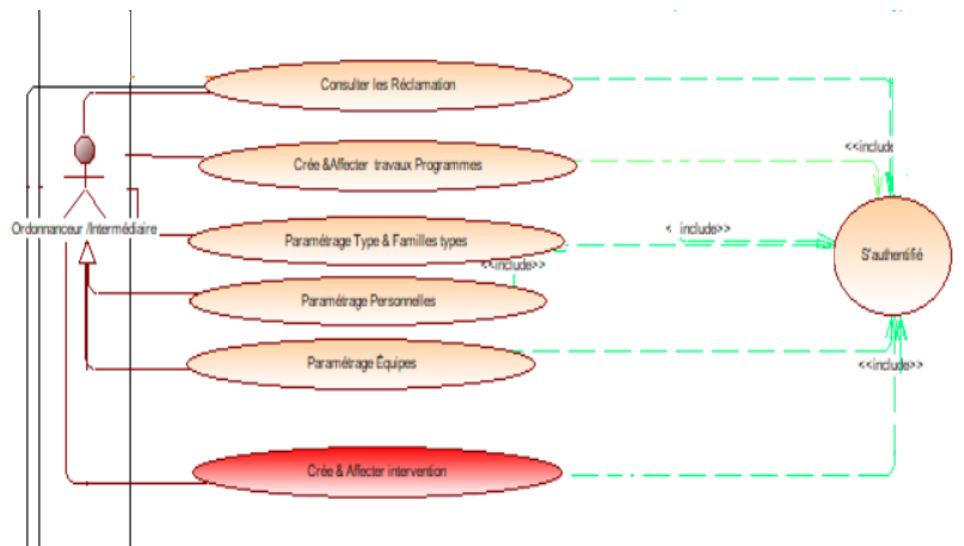
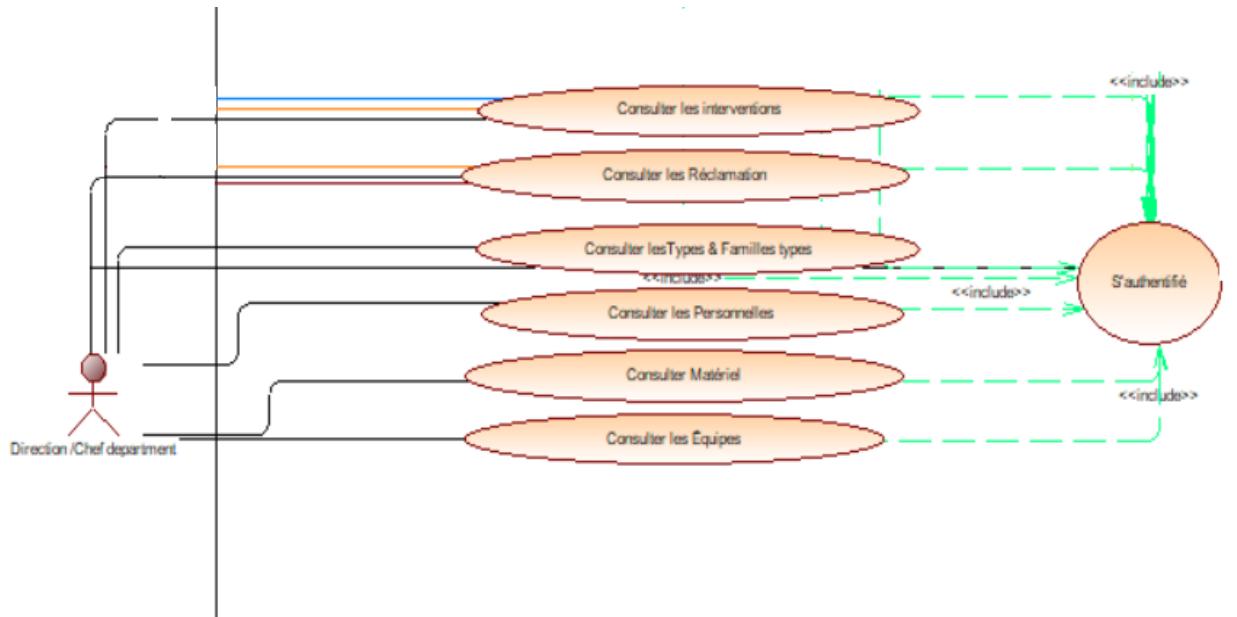


Figure 2.4: Cas d'utilisation pour l'acteur Ordonnanceur et Intermédiaire

## 2.2.5 Diagramme de cas d'utilisation pour l'acteur Chef de départements :

La figure ci-dessous illustre le diagramme de cas d'utilisation pour les acteurs " **Chef de départements** ".



**Figure 1.5: Cas d'utilisation pour l'acteur Chef de départements**

## 2.2.6 Diagrammes de cas d'utilisation textuelles :

**Tableau 2.1: Diagramme textuel d'authentification**

Le diagramme ci-dessous représente le schéma textuel d'authentification

<b>Titre</b>	Authentification
<b>Acteurs principaux</b>	L'utilisateur de l'application qui souhaite accéder à des fonctionnalités protégées.
<b>Scénario principal</b>	<p>L'utilisateur tente d'accéder à une fonctionnalité protégée ou à une ressource nécessitant une authentification.</p> <p>Le serveur reçoit la demande de l'utilisateur et vérifie si celui-ci est déjà authentifié.</p> <p>Si l'utilisateur n'est pas authentifié, le serveur renvoie une réponse</p>

	<p>indiquant que l'accès est refusé et demande à l'utilisateur de s'authentifier.</p> <p>L'utilisateur fournit ses informations d'identification (nom d'utilisateur et mot de passe) au serveur.</p> <p>Le serveur vérifie les informations d'identification en comparant avec celles stockées dans une base de données ou un système d'authentification.</p> <p>Si les informations d'identification sont valides, le serveur génère un JSON Web Token (JWT) contenant des informations supplémentaires sur l'utilisateur, telles que son identifiant et ses rôles.</p> <p>Le serveur renvoie le JWT à l'utilisateur en tant que réponse à la demande d'authentification réussie.</p> <p>L'utilisateur inclut le JWT dans les en-têtes de ses futures requêtes pour prouver son identité lors de l'accès à des fonctionnalités protégées.</p> <p>Le serveur vérifie le JWT à chaque demande pour s'assurer de l'identité de l'utilisateur et accorde l'accès aux ressources protégées si le JWT est valide.</p>
<b>Exceptions</b>	<p>Si l'utilisateur fournit des informations d'identification incorrectes lors de la demande d'authentification, le serveur renvoie une réponse d'erreur indiquant que l'authentification a échoué.</p> <p>Si le JWT est expiré, le serveur renvoie une réponse d'erreur indiquant que l'accès est refusé et demande à l'utilisateur de se reconnecter.</p> <p>Si le JWT est malformé ou invalide, le serveur renvoie une réponse d'erreur indiquant que l'accès est refusé et demande à l'utilisateur de s'authentifier à nouveau.</p> <p>Si l'utilisateur tente d'accéder à des fonctionnalités protégées sans fournir de JWT ou avec un JWT invalide, le serveur renvoie une réponse d'erreur indiquant que l'accès est refusé.</p>

**Tableau 2.2: Diagramme textuel gestion des interventions**

<b>Titre</b>	Gestion des interventions
<b>Acteurs principaux</b>	CRC, Ordonnanceur
<b>Scénario principal</b>	<p>L'utilisateur crée une nouvelle intervention en fournissant les informations pertinentes sur l'intervention, telles que la description, la date, l'emplacement, etc.</p> <p>Le système enregistre les détails de l'intervention dans la base de données.</p> <p>La base de données confirme la réussite de l'enregistrement.</p> <p>Le système informe l'utilisateur que l'intervention a été créée avec succès.</p> <p>L'utilisateur consulte les interventions existantes pour afficher la liste des interventions enregistrées</p> <p>Le système envoie une requête à la base de données pour récupérer les données des interventions.</p> <p>La base de données renvoie les données des interventions au système.</p> <p>Le système affiche la liste des interventions existantes à l'utilisateur.</p> <p>L'utilisateur souhaite mettre à jour les détails d'une intervention existante en fournissant les nouveaux détails.</p> <p>Le système met à jour les détails de l'intervention dans la base de données.</p> <p>La base de données confirme la réussite de la mise à jour.</p> <p>Le système informe l'utilisateur que l'intervention a été mise à jour avec succès.</p> <p>L'utilisateur souhaite supprimer une intervention existante.</p> <p>Le système supprime l'intervention de la base de données.</p> <p>La base de données confirme la réussite de la suppression.</p> <p>Le système informe l'utilisateur que l'intervention a été supprimée avec succès.</p>
<b>Exceptions</b>	Erreur de validation des données : Lors de la création ou de la mise à jour d'une intervention, des erreurs de validation peuvent survenir si certaines données fournies par l'utilisateur sont manquantes, incorrectes ou ne respectent pas les contraintes

définies (par exemple, une date incorrecte, une description vide, etc.). Le système doit informer l'utilisateur des erreurs de validation et demander une correction avant de poursuivre.

**Intervention non trouvée :** Lorsque l'utilisateur souhaite consulter, mettre à jour ou supprimer une intervention spécifique, le système doit vérifier si l'intervention avec l'identifiant fourni existe réellement dans la base de données. Si l'intervention n'est pas trouvée, le système doit renvoyer une erreur indiquant que l'intervention n'existe pas.

**Erreur de connexion à la base de données :** Lors de l'enregistrement, de la mise à jour ou de la suppression d'une intervention, le système peut rencontrer des problèmes de connexion à la base de données, par exemple, en raison d'une panne du serveur de base de données. Dans ce cas, le système doit renvoyer une erreur indiquant qu'il est actuellement impossible d'accéder à la base de données.

**Accès non autorisé :** Certaines interventions peuvent être restreintes à des utilisateurs spécifiques ou à des rôles particuliers. Si un utilisateur tente d'accéder à une intervention pour laquelle il n'a pas les autorisations requises, le système doit renvoyer une erreur indiquant que l'accès est non autorisé.

**Conflit de mise à jour :** Lors de la mise à jour d'une intervention, il est possible qu'une autre personne ait déjà effectué une mise à jour simultanée. Dans ce cas, il peut y avoir un conflit de mise à jour. Le système doit gérer ces situations en fournissant des mécanismes de résolution des conflits, tels que des verrous d'édition ou des versions de l'intervention.

**Erreurs du serveur :** Les erreurs internes du serveur peuvent survenir pour diverses raisons, telles que des problèmes de configuration, des bogues logiciels ou des erreurs inattendues. Le système doit être capable de détecter ces erreurs et de fournir des messages d'erreur appropriés à l'utilisateur.

## 2.3 Diagrammes de séquence :

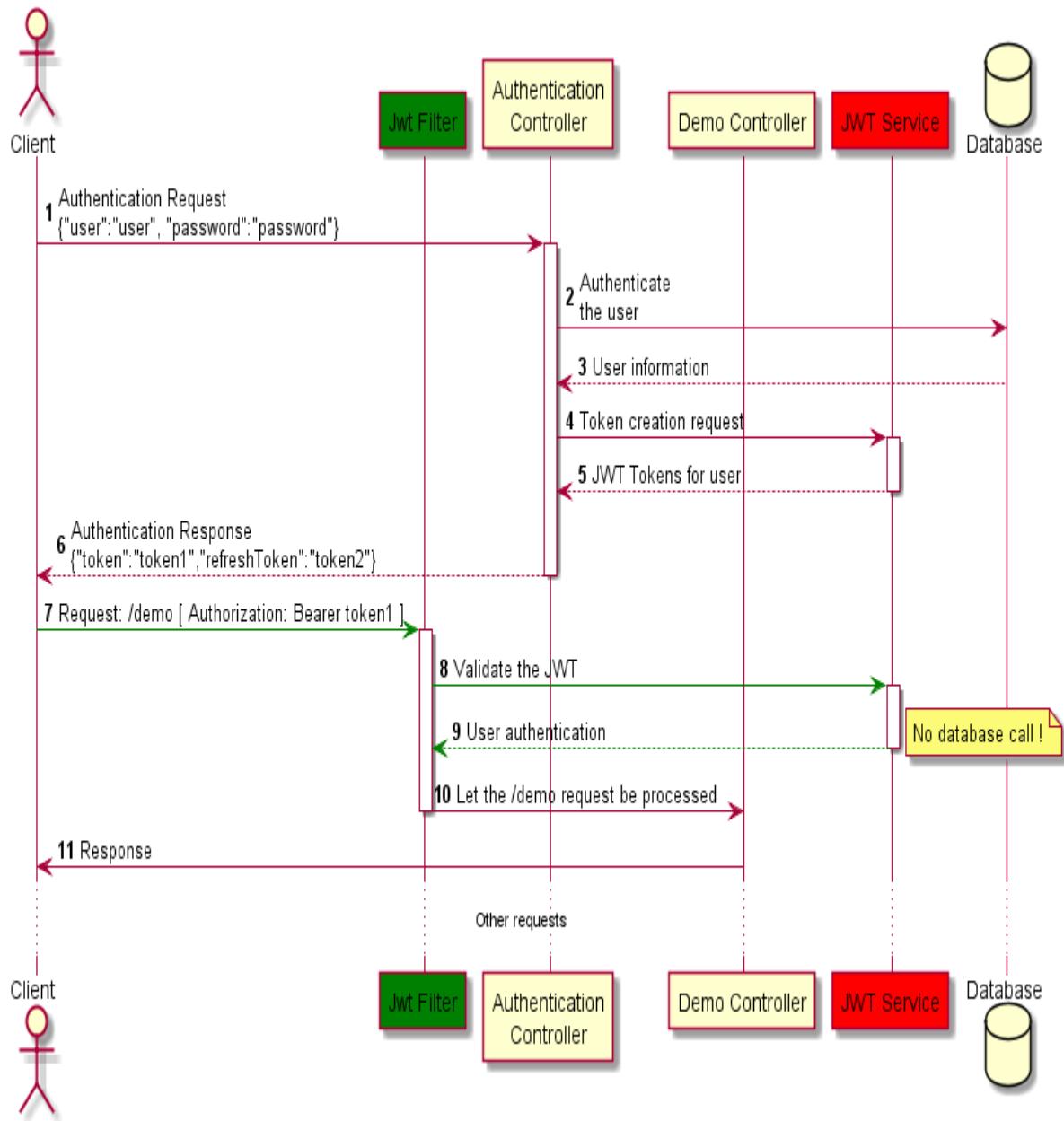


Figure 2.6 : Diagramme de séquence d'authentification

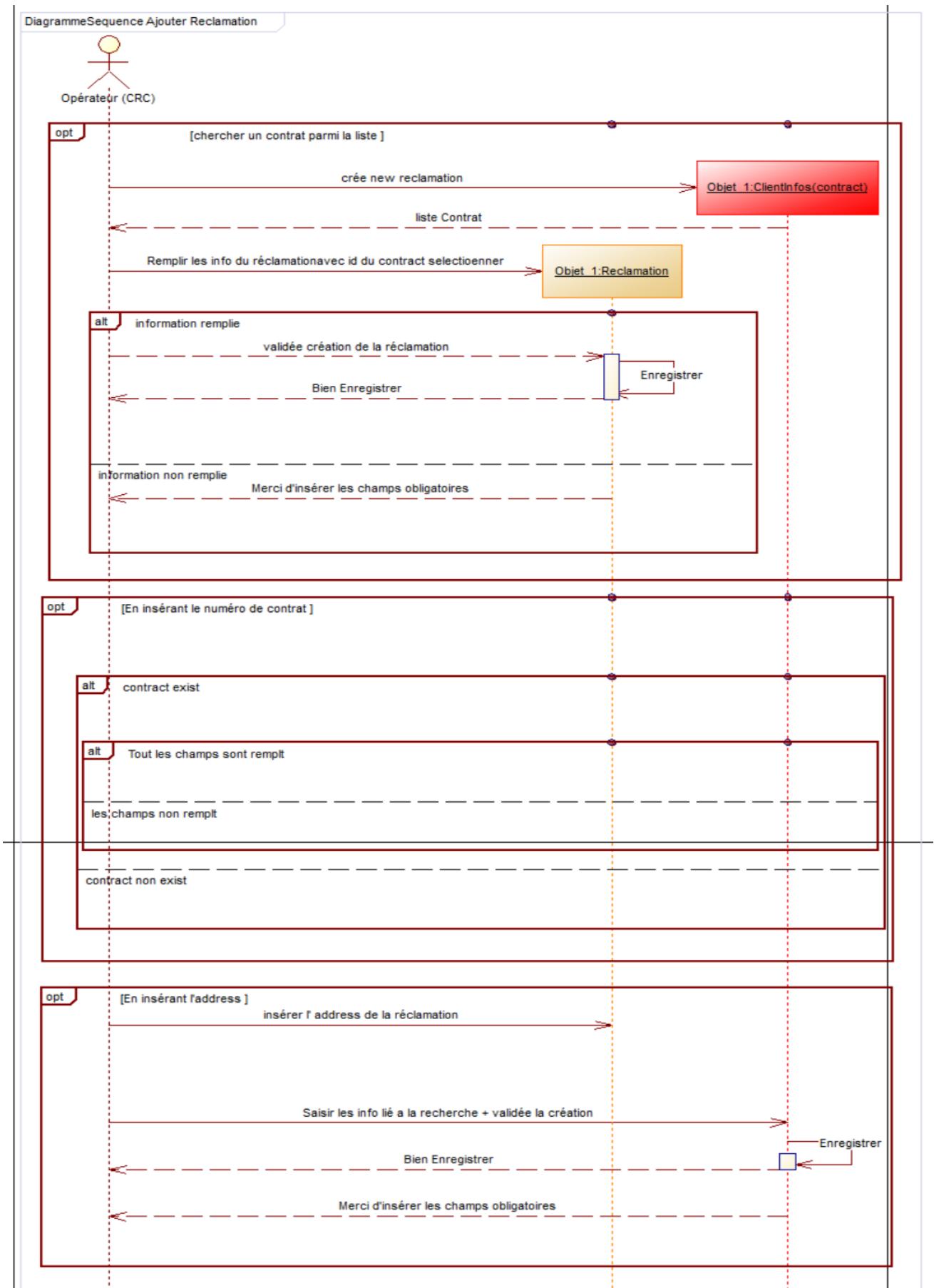


Figure 2.7 : Diagramme séquence Ajouter Réclamation

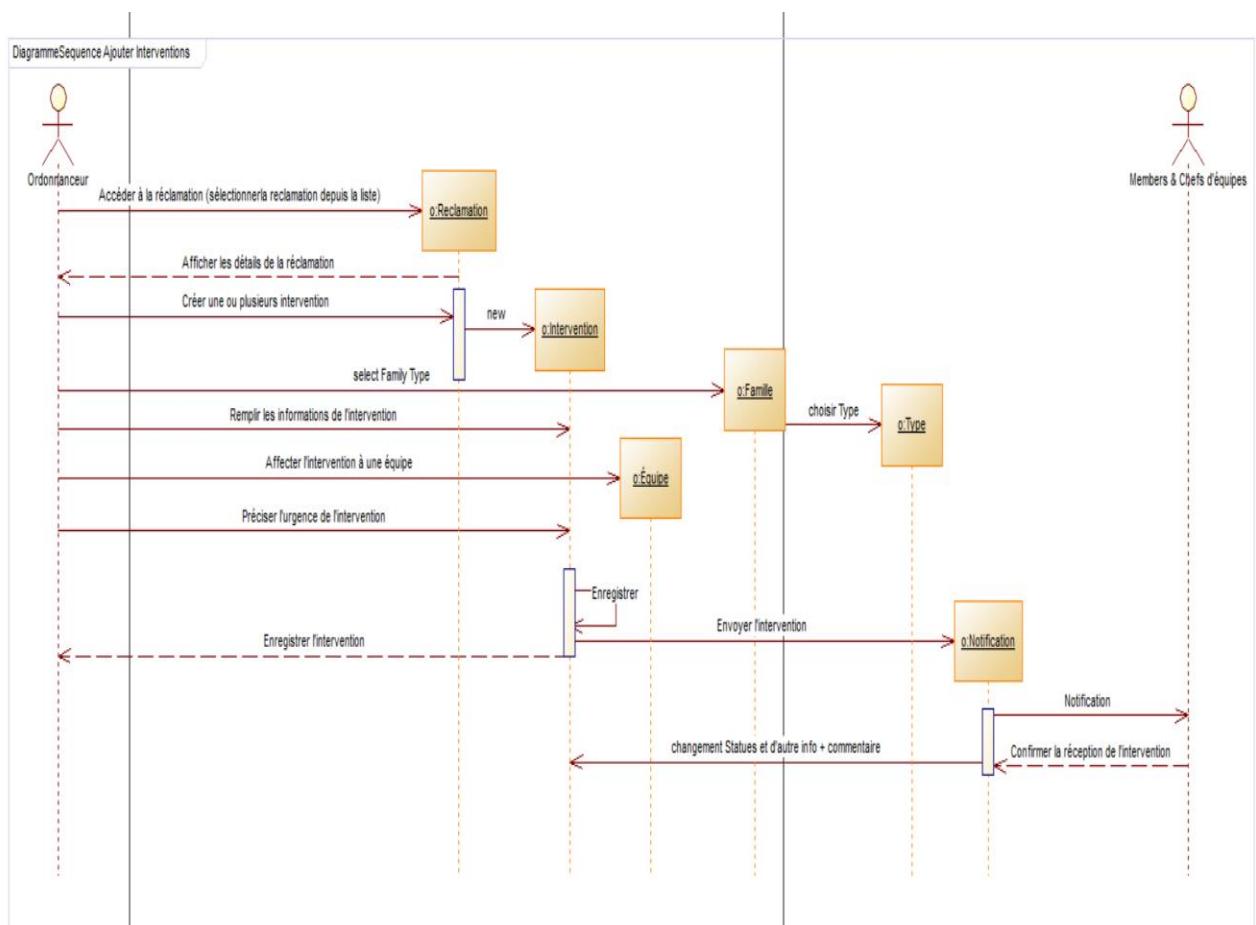
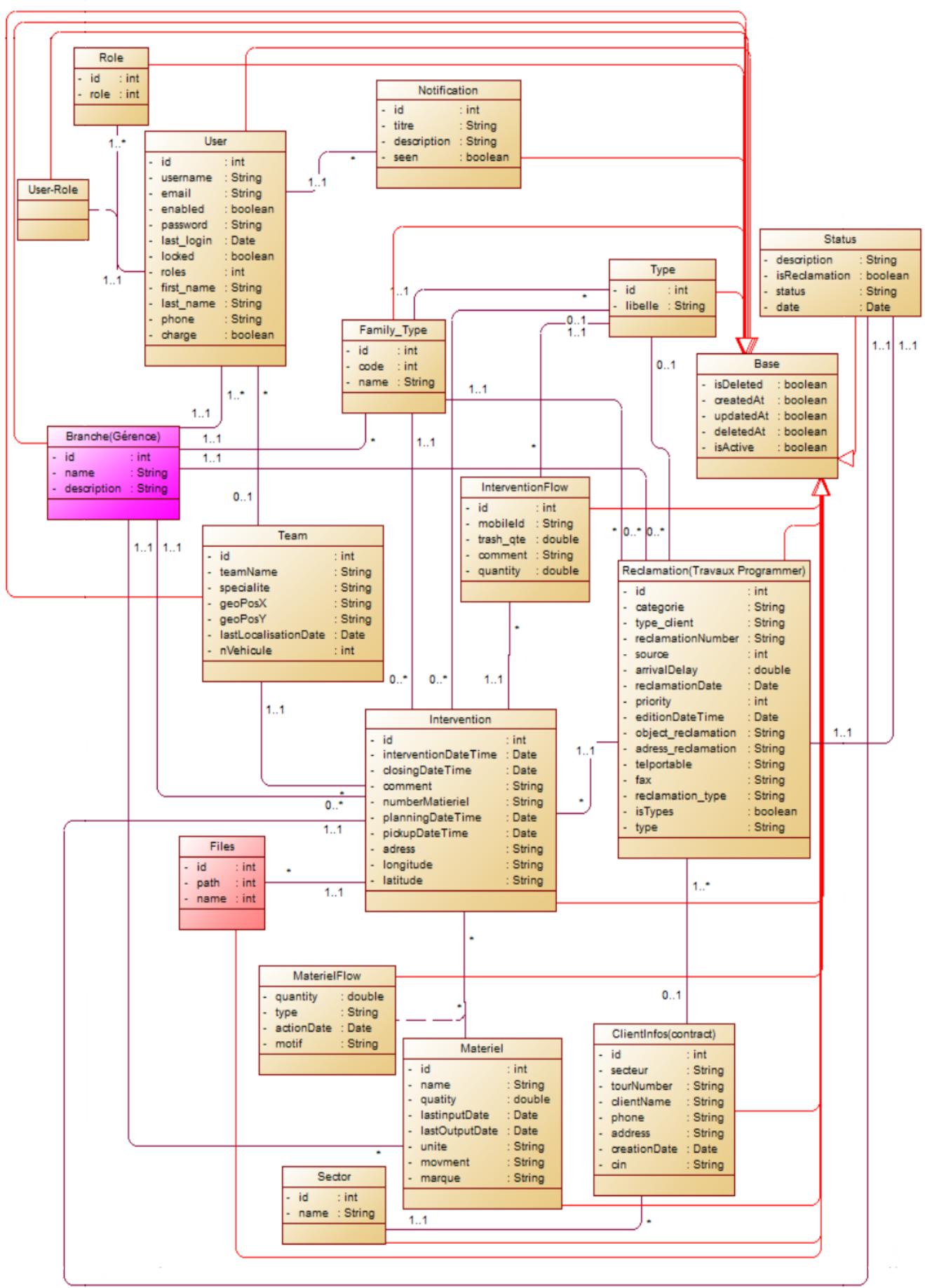


Figure 2.8: Diagramme séquence Ajouter Intervention

## 2.4 Diagrammes de Classe :

Figure2.9 Diagramme de classe



## **2.5Conclusion :**

En somme, l'analyse et la conception de notre projet Smart City ont permis de mettre en lumière plusieurs aspects critiques nécessitant une attention particulière. Les lacunes identifiées, telles que les défis liés à la partie mobile et à la sécurisation de l'accès aux divers services de l'application, constituent des axes de développement essentiels pour garantir la cohérence, la performance et la sécurité de l'ensemble du système. En intégrant ces constatations dans notre approche de développement, nous sommes mieux préparés à optimiser les prochaines étapes de notre projet afin de créer une plateforme Smart City robuste, fonctionnelle et répondant aux besoins actuels et futurs.

## **Chapitre 3**

---

### **Etude technique**

---

### **3.1 Introduction :**

Dans cette section, nous décrirons l'architecture globale de la Smart City plateforme. Nous présenterons les différents composants de la solution, leur rôle et leur interaction.

### **3.2 Architecture globale de la plateforme**

Le frontend, élaboré au moyen de ReactJs, établit une communication avec le BackEnd grâce à des API. Le BackEnd, construit sur les bases de Spring Boot, Spring Data et Spring MVC, assume la responsabilité du stockage des données dans une base de données, avec un accent particulier sur la sécurité via Spring Security. En outre, il exécute des requêtes pour récupérer les informations requises. Concernant le déploiement de notre application, nous optons pour la conteneurisation via Docker Compose, où chaque service est représenté par une image hébergée dans un unique conteneur.

### **3.3 Choix des technologies et justification des décisions prises**

Nous discuterons des choix technologiques effectués pour le développement de la plateforme. Cela inclut les langages de programmation, les Framework et les bibliothèques utilisés. Nous justifierons nos décisions en prenant en compte des critères tels que la robustesse, la maintenabilité, la performance et la compatibilité avec les besoins spécifiques du projet. Nous expliquerons également comment ces technologies interagissent avec l'architecture globale de la plateforme et comment elles contribuent à la réalisation des objectifs fixés.

La conception de la solution joue un rôle crucial dans le développement de la Smart City plateforme. En décrivant en détail l'architecture globale, la modélisation des données et des fonctionnalités, ainsi que les choix technologiques effectués, nous poserons les bases solides pour la mise en œuvre de la plateforme. Cette conception permettra de garantir la cohérence, la performance et la fonctionnalité de la solution, en répondant aux besoins identifiés lors de l'analyse précédente.

### 3.4 Modèle de conception MVC :

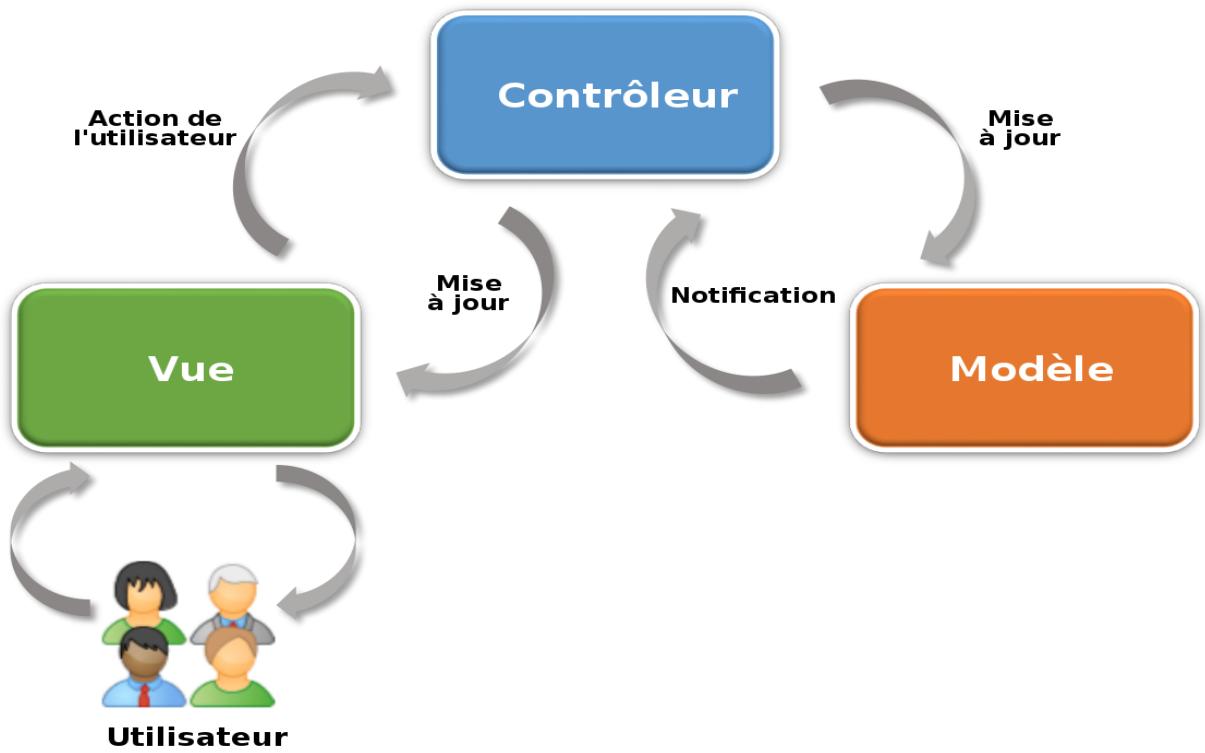


Figure 3.1: Modèle MVC

Le modèle MVC (Modèle-Vue-Contrôleur) adopté par Spring MVC est un modèle de conception architectural qui divise une application en trois composants principaux : le Modèle, la Vue et le Contrôleur. Chacun de ces composants joue un rôle spécifique dans le processus de développement et permet une séparation claire des responsabilités, ce qui facilite la maintenance, la réutilisation du code et la scalabilité de l'application.

- Modèle :

Le modèle représente les données de l'application et définit la logique métier. Il encapsule les données et les traitements associés, tels que l'accès à la base de données, la logique de calcul, etc. Le modèle est généralement implémenté sous forme de classes POJO (Plain Old Java Object) qui sont indépendantes de l'interface utilisateur. Spring MVC facilite la gestion du modèle en permettant l'utilisation d'annotations telles que `@ModelAttribute`, `@RequestBody`, etc., pour lier les données de la requête HTTP aux objets du modèle.

- Vue (View) :

La vue est responsable de la présentation des données au client. Elle définit la manière dont les informations du modèle sont affichées à l'utilisateur. Dans Spring MVC, la vue est généralement mise en œuvre à l'aide de templates (par exemple, JSP, Thymeleaf) qui peuvent intégrer les données du modèle pour générer des pages HTML dynamiques. La vue n'a pas de logique métier complexe et doit se concentrer uniquement sur l'interface utilisateur.

- Contrôleur (Controller) :

Le contrôleur agit comme un intermédiaire entre la vue et le modèle. Il gère les requêtes HTTP entrantes, traite les données, interagit avec le modèle et détermine quelle vue doit être affichée en réponse à une requête. Dans Spring MVC, les contrôleurs sont mis en œuvre en tant que classes Java annotées avec @Controller, et les méthodes de ces classes sont associées aux différentes URL et actions de l'application à l'aide d'annotations telles que @RequestMapping, @GetMapping, @PostMapping, etc. (2)

### 3.5 Technologies utilisées



Figure 3.2 : Logo springboot

**Spring Boot** : est un Framework open-source basé sur Spring qui facilite le développement rapide d'applications Java. Il fournit un environnement d'exécution préconfiguré qui permet aux développeurs de se concentrer sur la logique métier de leur application plutôt que sur la configuration. (3)

### - Frontend

Le frontend de la Smart City plateforme est conçu pour offrir une expérience utilisateur conviviale et réactive. Dans cette section, nous décrirons les technologies et les bonnes pratiques utilisées pour le développement frontend.

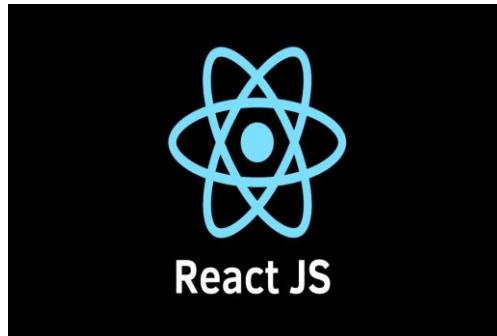


Figure 3.3 : Logo ReactJs

### **JavaScript et ReactJs JS pour le développement frontend**

Nous avons opté pour JavaScript, un langage de programmation largement utilisé pour le développement web, afin de créer le frontend de la plateforme. Pour tirer parti des avantages de la réactivité et de la modularité, nous avons utilisé le Framework ReactJs JS. ReactJs JS permet de construire des interfaces utilisateur interactives et dynamiques en utilisant des composants réutilisables.

### - Mobile

La Smart City plateforme comprend également une version mobile pour une accessibilité optimale. Nous avons choisi d'utiliser React Native, un framework de développement d'applications mobiles, pour créer cette version.

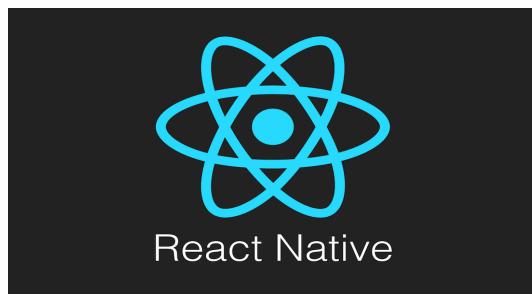


Figure 3.4: Logo React native

### **Utilisation de React Native pour le développement d'applications mobiles**

React Native permet de développer des applications mobiles multiplateformes en utilisant du code JavaScript. Il offre une approche de développement efficace, car une grande partie du code

peut être partagée entre les plateformes iOS et Android, ce qui réduit les efforts de développement et de maintenance.

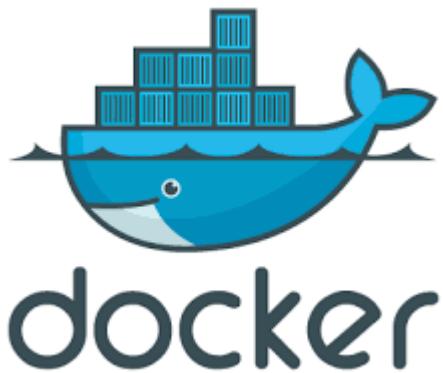


Figure 3.5 Logo Docker

**Docker** : est une technologie de conteneurisation open-source qui permet de créer, déployer et exécuter des applications dans des conteneurs légers et portables. Un conteneur Docker encapsule une application et toutes ses dépendances, y compris les bibliothèques, les dépendances système et le code, dans une seule unité logicielle autonome. Cela permet aux applications de s'exécuter de manière cohérente et fiable dans différents environnements, qu'il s'agisse d'un environnement de développement local, d'un serveur de production ou d'un cloud public. (4)



Figure 3.6: Logo git

Git : est un système de contrôle de version distribué open-source largement utilisé dans le développement logiciel. Il permet de suivre les modifications apportées au code source d'un

projet au fil du temps et facilite la collaboration entre les membres d'une équipe de développement. (5)



**Figure 3.7:** Logo Gitlab

**GitLab :** est une plateforme de gestion du cycle de vie des applications (Application Life cycle Management - ALM) basée sur Git. C'est un outil open-source complet qui offre des fonctionnalités de gestion de code source, de suivi des problèmes, de gestion des tâches, d'intégration continue (CI/CD), de déploiement, de gestion des versions, et bien plus encore. GitLab permet aux équipes de développement de collaborer efficacement sur des projets logiciels, de suivre les modifications apportées au code et de livrer des logiciels de manière plus rapide et fiable. (5)



**Figure 3.8:** Logo Jira

**Jira :** est une plateforme de gestion de projet et de suivi des problèmes développée par Atlassian. C'est un outil populaire utilisé par les équipes de développement logiciel, les équipes de projet, les équipes de support client et d'autres équipes pour planifier, suivre et gérer leur travail de manière collaborative et transparente. (6)



**Figure 3.9: Json Web Token**

**JWT (JSON Web Token)** : est un standard ouvert (RFC 7519) pour la création de jetons d'accès basés sur JSON qui sont utilisés pour l'authentification et l'autorisation dans les applications web et mobiles. Les JWT sont souvent utilisés pour sécuriser les API et les ressources en permettant l'échange d'informations entre différentes parties de manière sécurisée et compacte. (7)

# **Chapitre 4**

---

## **Réalisation**

---

## 4.1 Interfaces graphiques d'authentifications:

La figure représente l'interface d'authentification de connexion, conçue pour offrir une expérience utilisateur fluide et sécurisée. Cette page de connexion met en œuvre la technologie JWT (JSON Web Token) pour gérer de manière efficace et sécurisée l'authentification des utilisateurs.

L'aspect innovant de cette solution réside dans la gestion transparente des jetons d'accès et de rafraîchissement. En arrière-plan, le système met en œuvre un processus automatisé de renouvellement des jetons d'accès et de rafraîchissement. Les jetons d'accès, qui contiennent les informations d'identification de l'utilisateur, sont périodiquement renouvelés pour garantir la sécurité des données tout en minimisant les interruptions pour l'utilisateur final.

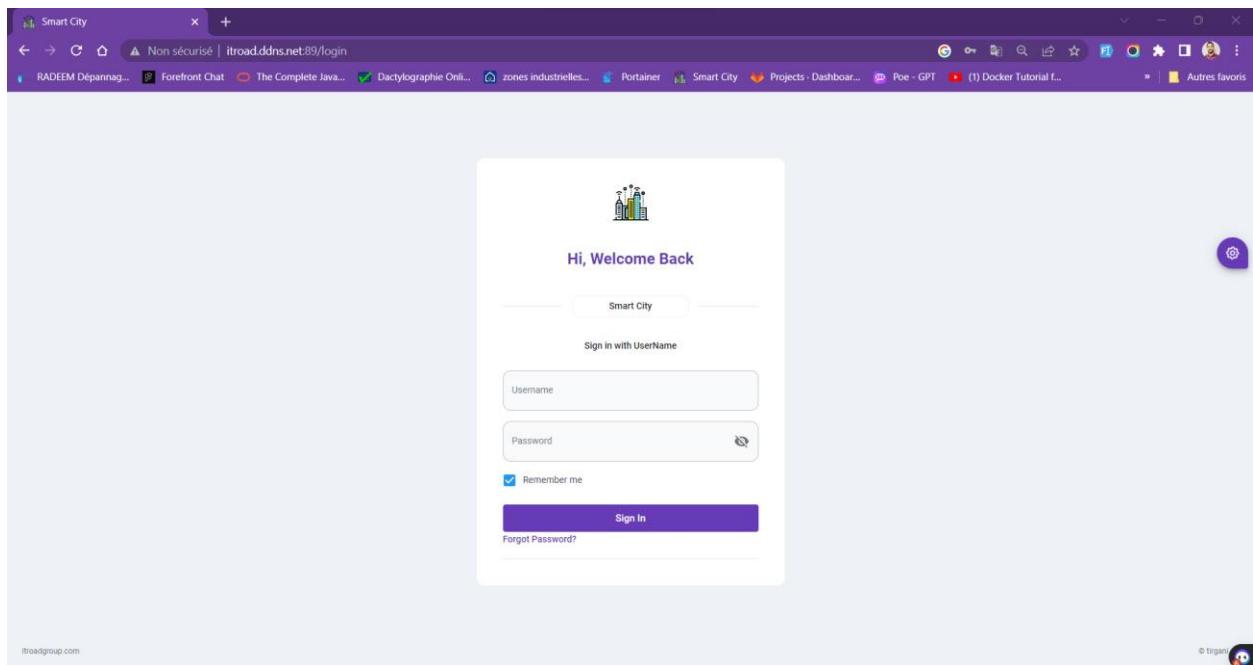
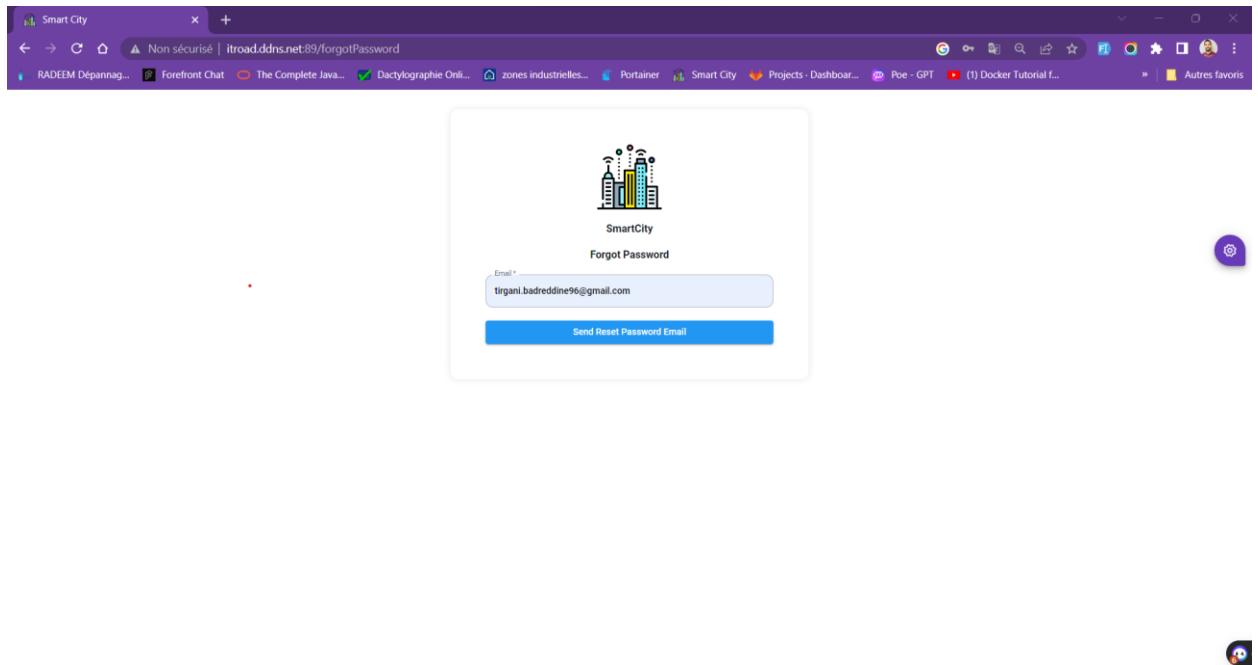


Figure 4.1: Authentification page Login

La représentation visuelle décrit une étape de récupération de mot de passe au sein d'une interface utilisateur conviviale. Lorsqu'un utilisateur oublie son mot de passe, il peut saisir son adresse e-mail associée au compte. Une fois l'adresse e-mail soumise, le système effectue une vérification pour s'assurer de sa validité.

Ensuite, le système déclenche un processus automatisé en utilisant le protocole ICMP (Internet Control Message Protocol) de Google. Ce processus consiste à envoyer un e-mail au propriétaire

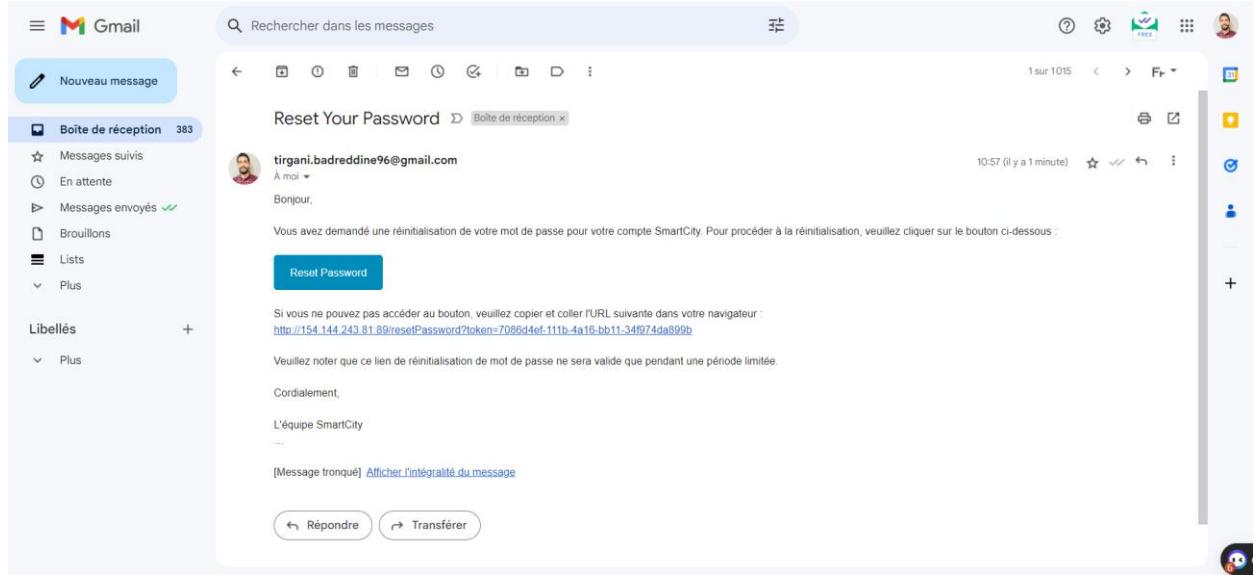
de l'adresse e-mail fournie. Cet e-mail contient un lien unique permettant de réinitialiser le mot de passe.



**Figure 4.2: Mot de passe oublié**

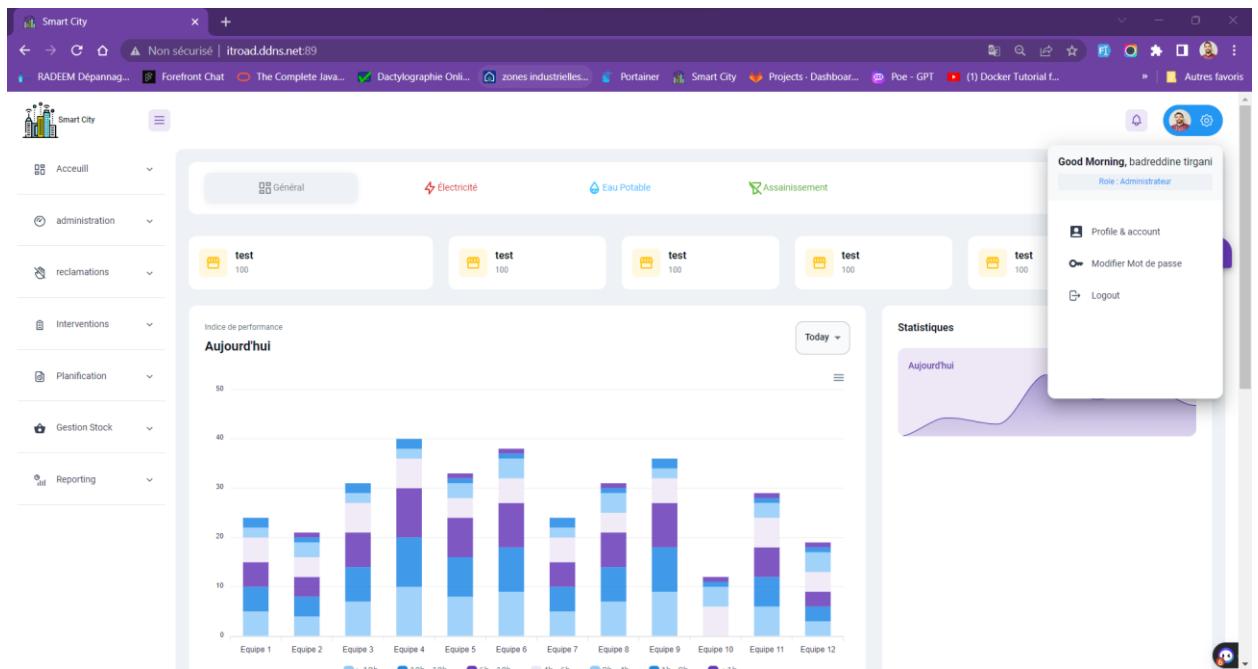
La particularité réside dans le jeu temporel introduit pour assurer la sécurité et la protection des données. Une fois que l'e-mail de réinitialisation est envoyé, le lien qu'il contient a une durée de validité prédéterminée. Le destinataire de l'e-mail doit cliquer sur ce lien dans un laps de temps défini pour pouvoir réinitialiser son mot de passe avec succès. Cela renforce la sécurité en évitant que des liens de réinitialisation de mot de passe restent valides indéfiniment, réduisant ainsi les risques potentiels liés à des accès non autorisés.

En somme, la figure illustre un processus de récupération de mot de passe basé sur l'adresse e-mail, impliquant l'utilisation du protocole ICMP de Google pour envoyer un e-mail de réinitialisation avec une durée de validité temporelle. Cette approche renforce la sécurité des comptes utilisateurs tout en offrant une expérience fluide de récupération de mot de passe.



**Figure 4.3: Réception d'un e-mail de récupération de mot de passe suite à la validation de l'utilisateur.**

La figure présente un tableau de bord dynamique qui offre un aperçu en temps réel des statistiques les plus récentes de l'application, ainsi que la localisation GPS des différentes interventions. Ce tableau de bord fournit aux utilisateurs une vue holistique et informative de l'activité de l'application.



**Figure 4.4: Dashboard**

La figure présente l'interface d'administration dédiée au paramétrage des familles de types

d'interventions, organisées en fonction des différentes gérances. Les gérances spécifiques qui sont mises en évidence dans cette illustration sont l'électricité, l'eau potable et l'assainissement.

## 4.2 Interfaces graphiques partie d'administrations :

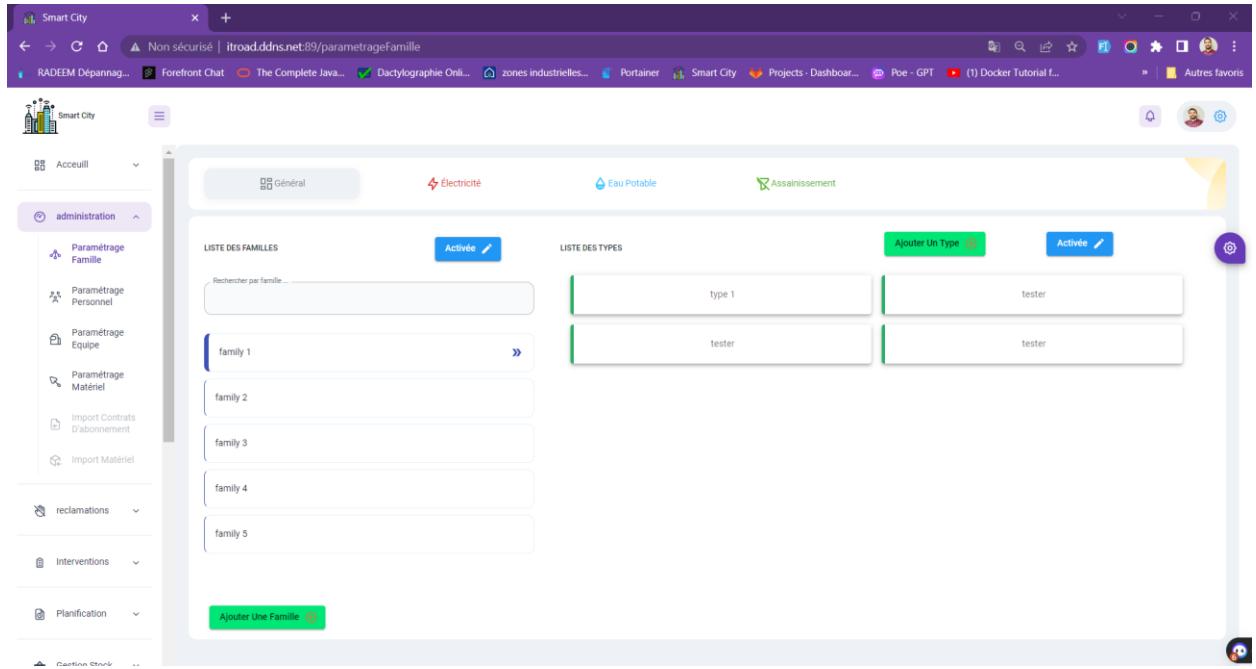


Figure 4.5: Administration Paramétrage Famille Type Famille

La figure présente l'interface permettant d'ajouter une nouvelle famille dans le système. Cette fonctionnalité est conçue pour permettre aux administrateurs de créer facilement de nouvelles catégories ou familles en fonction des besoins spécifiques de l'application.

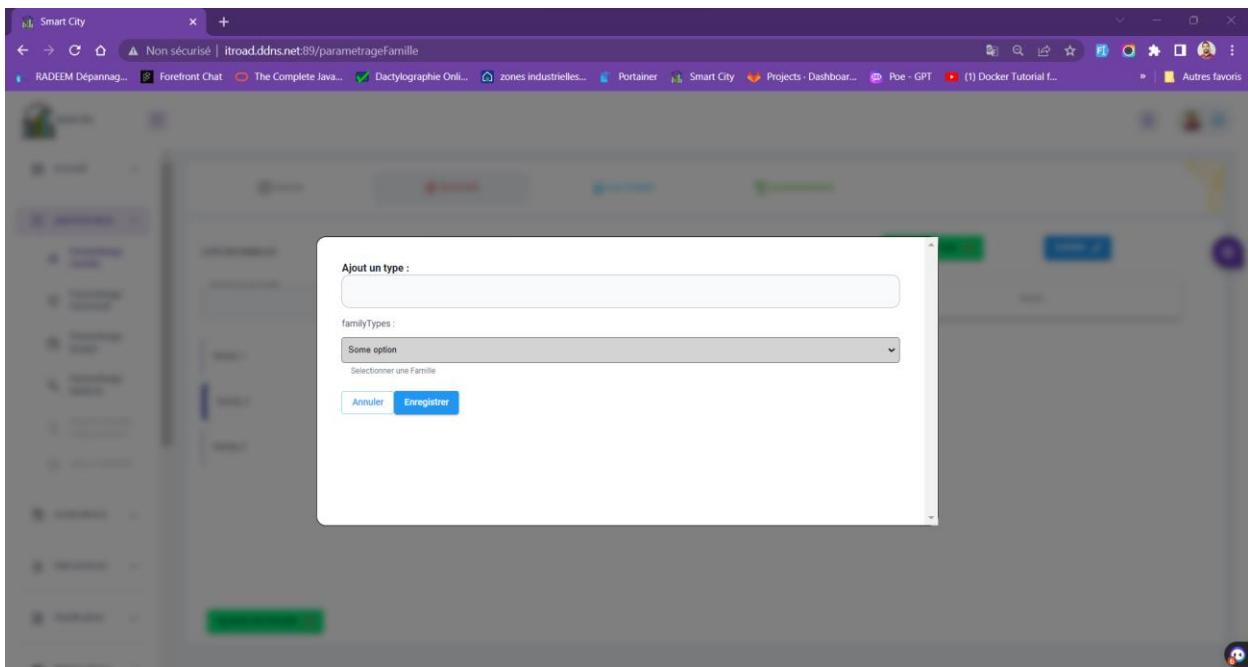


Figure 4.6: Ajouter Famille

La figure présente une interface permettant d'activer la modification des familles dans le système. Cette fonctionnalité est conçue pour permettre aux administrateurs d'apporter des modifications aux familles existantes, en ajustant les détails et les paramètres selon les besoins de l'application.

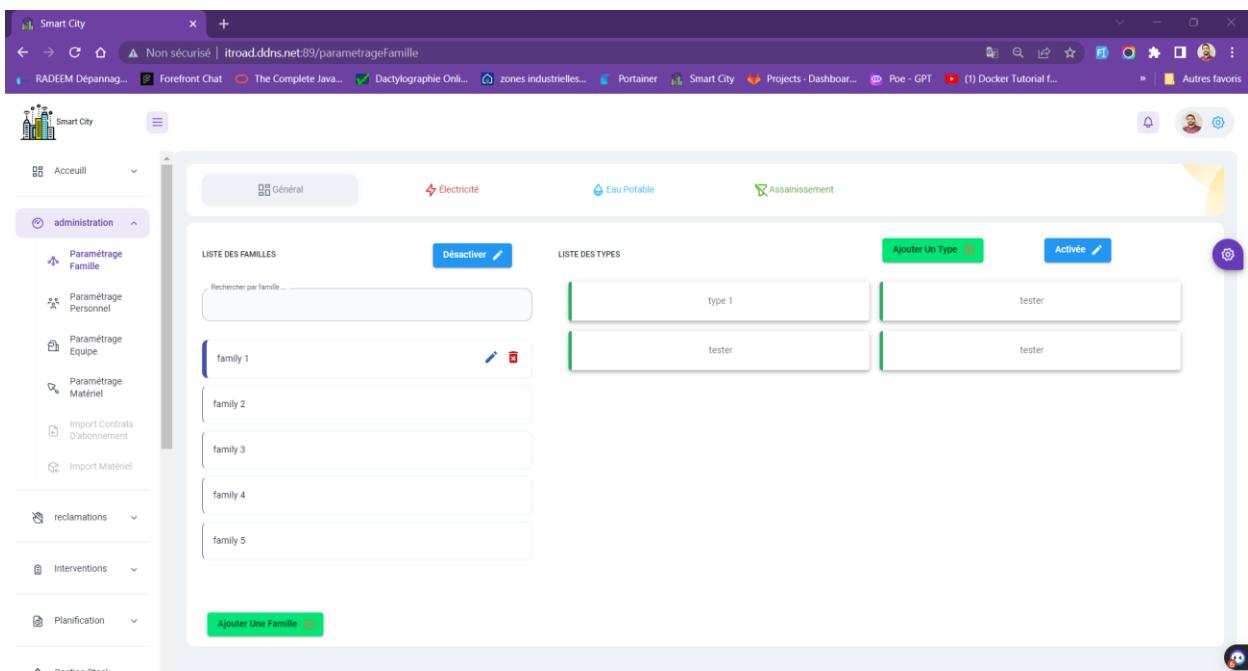
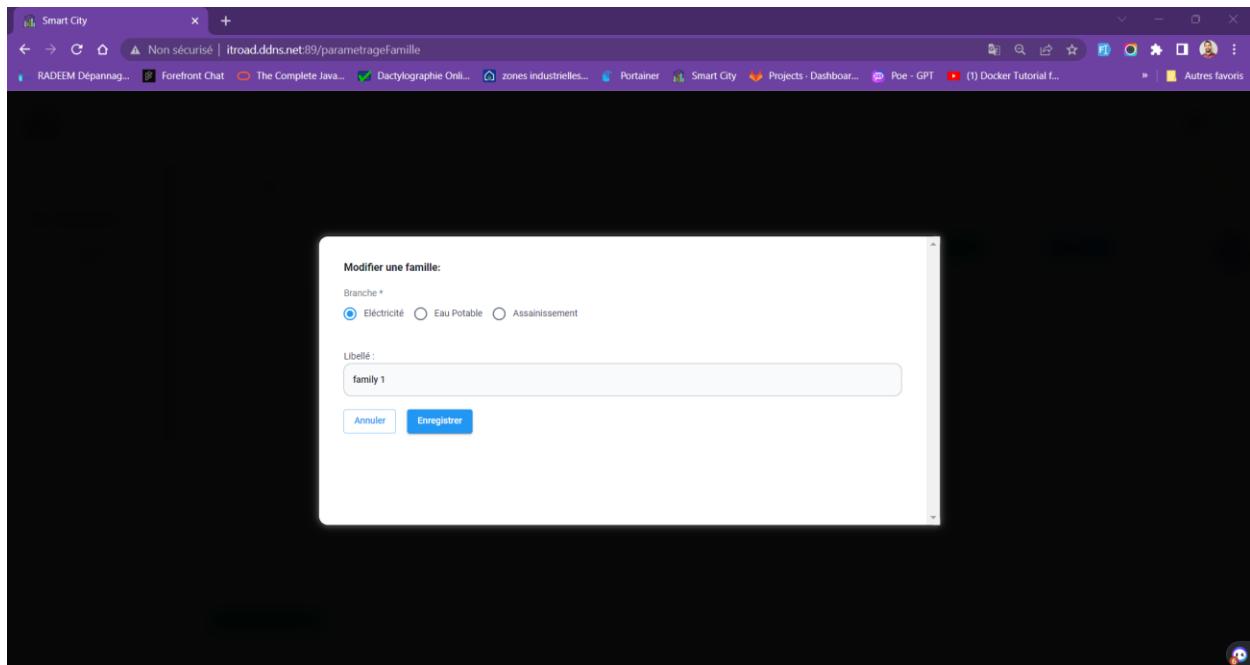


Figure 4.7: Activé la modification des familles

La figure présente une interface permettant de modifier une famille existante dans le système. Cette fonctionnalité est conçue pour permettre aux administrateurs de mettre à jour les détails et les paramètres d'une famille spécifique en fonction des besoins changeants de l'application.

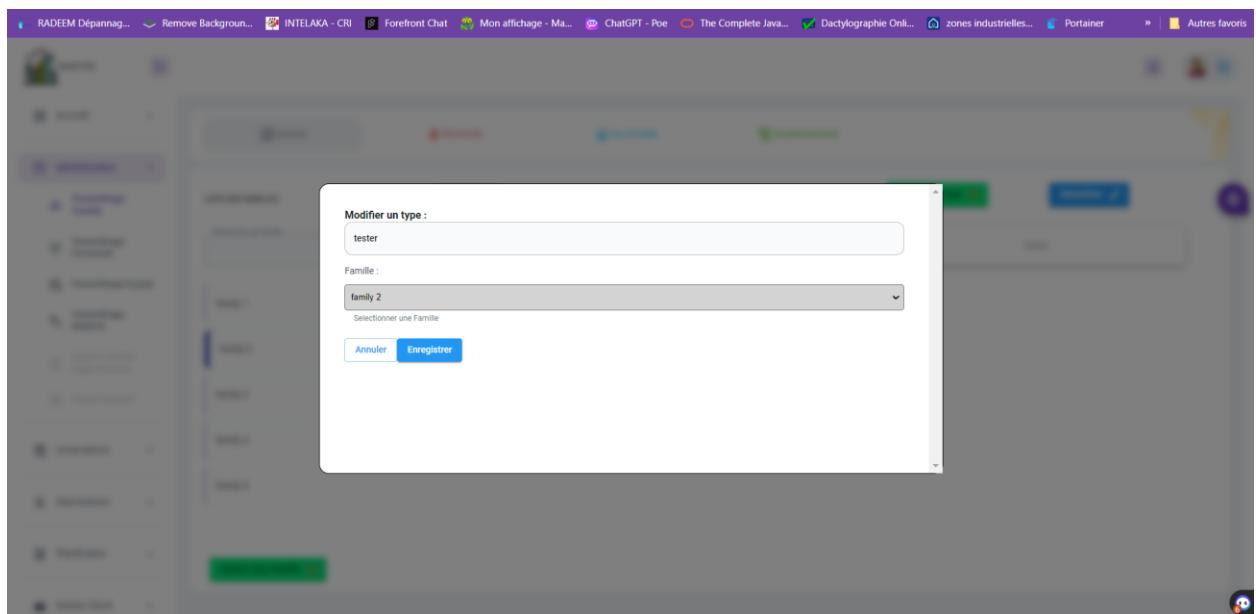


**Figure 4.8: Modifier une famille**

La figure représente une interface permettant d'activer la modification des types pour une famille spécifique dans le système. Cette fonctionnalité est conçue pour permettre aux administrateurs d'ajuster et de personnaliser les types d'interventions au sein d'une famille donnée, en fonction des besoins spécifiques de cette famille.

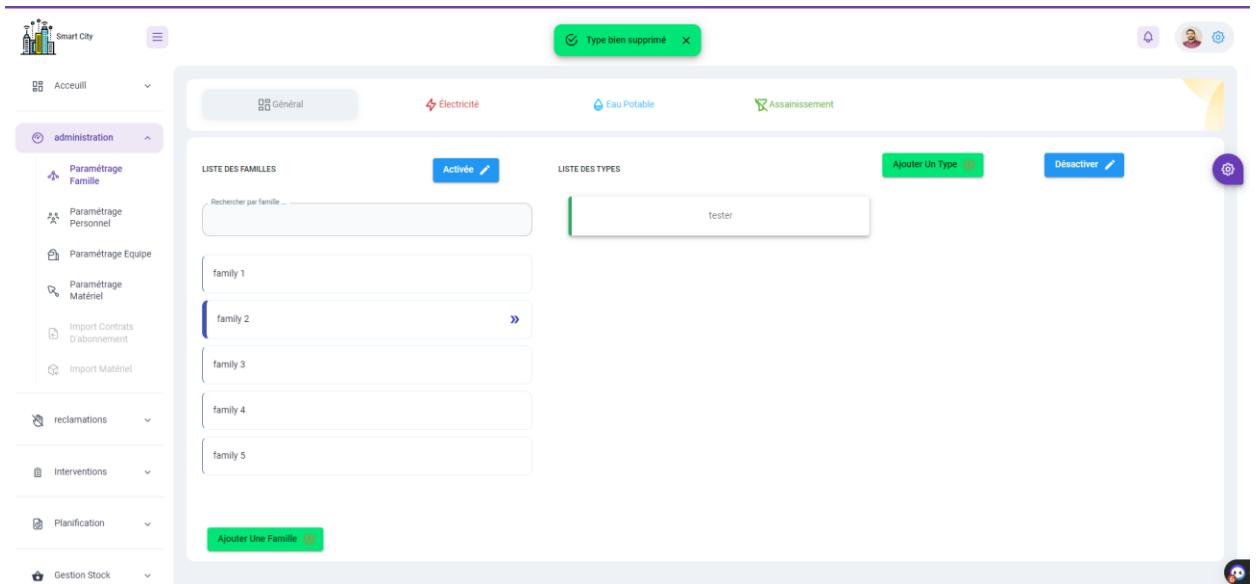
**Figure 4.9: Activer la modification des types pour une famille donne**

La figure représente une interface de modification pour un type d'intervention spécifique au sein d'une famille donnée dans le système. Cette interface est conçue pour permettre aux administrateurs de personnaliser les détails et les paramètres d'un type d'intervention particulier.



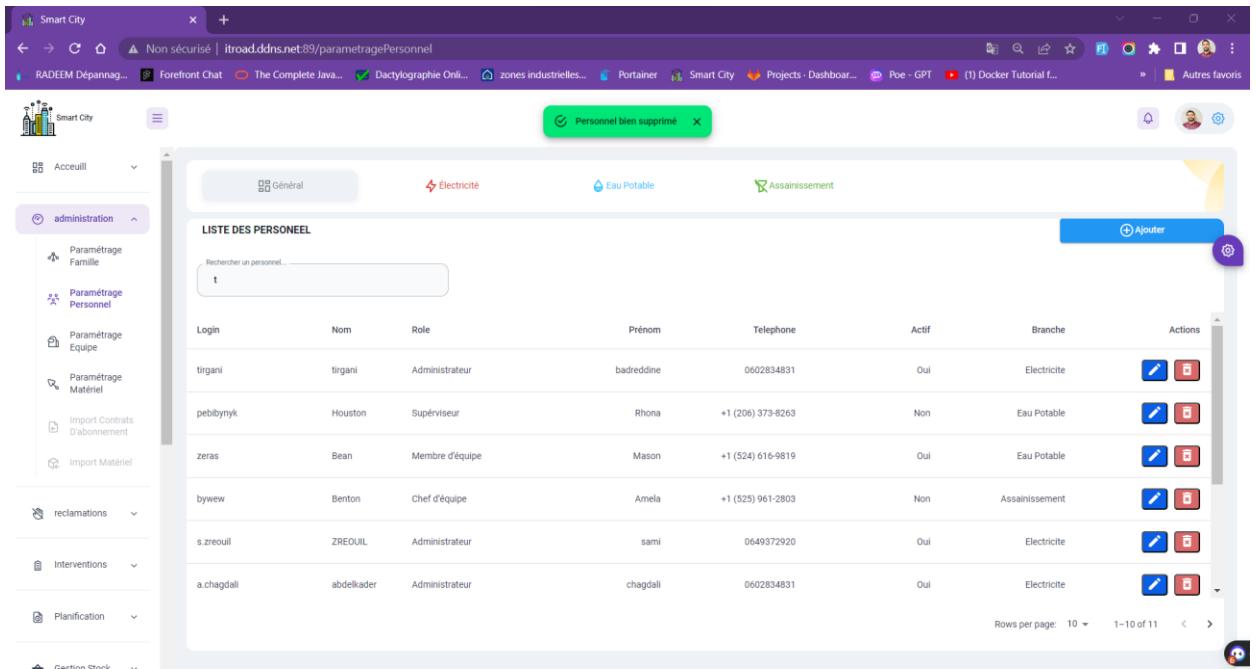
**Figure 4.10: Modification de Typé sélectionné**

La figure présente une interface permettant la suppression d'une famille spécifique dans le système. Cette fonctionnalité est conçue pour permettre aux administrateurs de supprimer des catégories de familles lorsqu'elles ne sont plus nécessaires ou pertinentes pour l'application.



**Figure 4.11: Suppression de la famille**

La figure représente une interface permettant la suppression d'un membre du personnel spécifique dans le système. Cette fonctionnalité est conçue pour permettre aux administrateurs de supprimer des membres du personnel lorsqu'ils ne sont plus actifs, ne font plus partie de l'organisation, ou pour d'autres raisons.



**Figure 4.12: Suppression d'un personnel**

La figure présente une interface permettant d'ajouter un nouveau membre du personnel dans le système. Cette fonctionnalité est conçue pour permettre aux administrateurs d'intégrer de nouveaux membres du personnel à l'organisation ou à l'application.

**Figure 4.13: Ajouter un nouveau personnel**

La figure représente une interface où les personnels sont listés et organisés par gérances spécifiques. Cette fonctionnalité permet aux administrateurs d'avoir une vue d'ensemble des membres du personnel répartis par départements ou catégories spécifiques, facilitant ainsi la gestion et la communication au sein de l'organisation.

**Figure 4.14: Lister les personnels**

La figure représente une interface permettant le filtrage des personnels en fonction des différentes gérances au sein de l'organisation. Cette fonctionnalité offre aux administrateurs la possibilité de visualiser uniquement les membres du personnel associés à une gérance spécifique, facilitant ainsi la gestion et la communication ciblée.

The screenshot shows a web application titled "Smart City" with a purple header bar. The URL is "Non sécurisé | itroad.ddns.net:89/parametragePersonnel". The left sidebar has a tree menu with categories like "administration", "reclamations", "Interventions", "Planification", and "Carte des sites". Under "administration", "Paramétrage Personnel" is selected. The main content area has tabs at the top: "Général", "Électricité", "Eau Potable" (which is active), and "Assainissement". Below is a table titled "LISTE DES PERSONNEL" with columns: Login, Nom, Role, Prénom, Telephone, Actif, Branche, and Actions. The table contains three rows of data. At the bottom right of the table, there are buttons for "Rows per page: 10" and "1-10 of 11".

Login	Nom	Role	Prénom	Telephone	Actif	Branche	Actions
pebibynyk	Houston	Superviseur	Rhona	+1 (206) 373-8263	Non	Eau Potable	
zeros	Bean	Membre d'équipe	Mason	+1 (524) 616-9819	Oui	Eau Potable	
b.khalil	bio	Membre d'équipe	khalil	0602834831	Oui	Eau Potable	

**Figure 4.15: Filtrage des personnels par gérance**

La figure représente une interface permettant le filtrage des personnels en fonction de différents critères tels que le nom, le rôle, le login, etc. Cette fonctionnalité permet aux administrateurs de rechercher rapidement des membres du personnel en utilisant des mots-clés spécifiques pour afficher les informations pertinentes.

The screenshot shows the 'LISTE DES PERSONNEL' (List of Personnel) page. The search bar at the top contains the name 'tirgani'. The table below lists personnel with the following columns: Login, Nom, Role, Prénom, Telephone, Actif, Branche, and Actions. One row is highlighted for the user 'tirgani'. The 'Actions' column for this row contains edit and delete icons.

**Figure 4.16: Filtrage des personnels par contenu exemple login nom rôle**

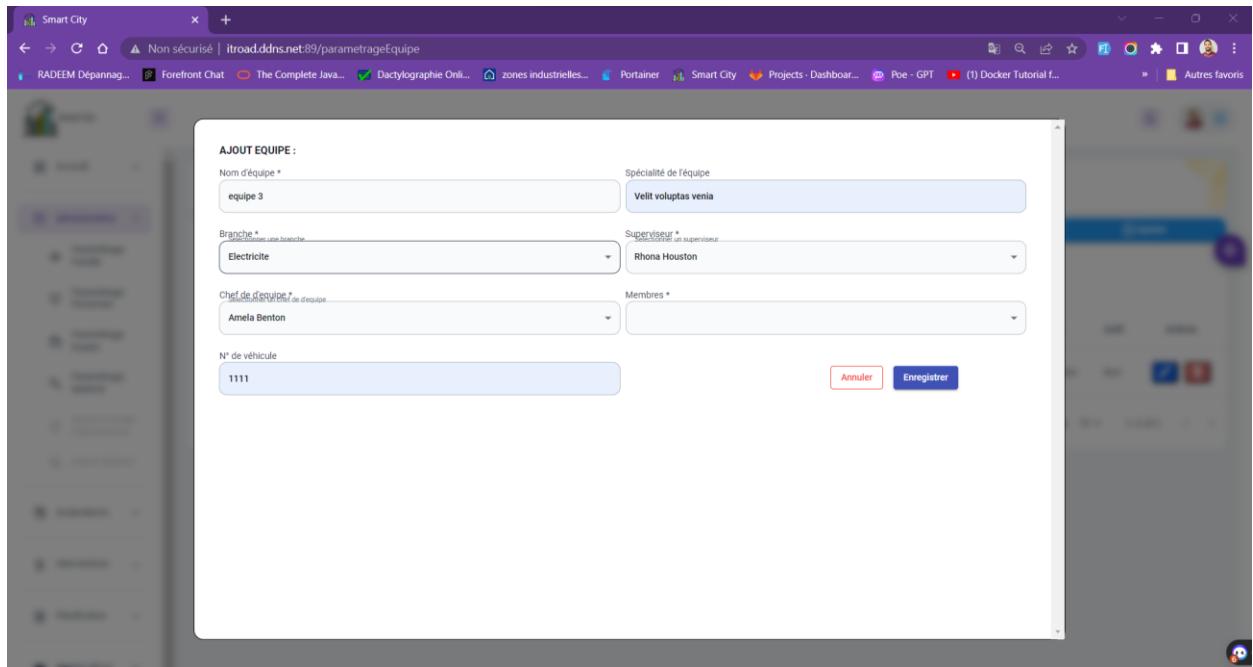
La figure représente une interface de modification d'un membre du personnel dans le système, en prenant en compte les attributs suivants : Nom, CIN (Carte d'identité nationale), nom d'utilisateur, mot de passe, branche, rôle et statut actif/inactif. Cette interface permet aux administrateurs de mettre à jour les informations d'un membre du personnel de manière précise et efficace.

The screenshot shows the 'MODIFIER PERSONNEL' (Edit Personnel) form. The form fields are as follows:

- Nom: badreddine
- CIN: E7435639
- Prénom: tirgani
- Matricule: ABB9866G
- Nom d'utilisateur: tirgani
- Tél: 0602834831
- Email: tirgani.badreddine96@gmail.com
- Selectionner un Profil: Administrateur
- Mot de passe: (empty)
- Confirmation du mot de passe: (empty)
- Branches: Electricité (radio button selected), Eau potable, Assainissement
- Actif: checked

**Figure 4.17: Modification d'un personnel**

La figure représente une interface permettant d'ajouter une nouvelle équipe dans le système, en prenant en compte les attributs suivants : nom, branche, chef d'équipe, superviseur, spécialité, membres de l'équipe et numéro de véhicule. Cette interface facilite l'ajout et la gestion des équipes au sein de l'organisation.



**Figure 4.18: Ajouter une nouvelle équipe**

La figure représente une interface permettant la modification d'une équipe sélectionnée dans le système. Cette fonctionnalité offre aux administrateurs la possibilité de mettre à jour les informations et les attributs d'une équipe spécifique, ce qui permet une gestion plus précise et efficace.

**MODIFIER EQUIPE :**

- Nom d'équipe \* : Equipe 2
- Spécialité de l'équipe : Commodo alias est om
- Branche \* : Électricité
- Superviseur \* : Rhona Houston
- Chef de l'équipe \* : Amela Benton
- Membres \* : Mason Bean, Jessamine Finley
- N° de véhicule : 63

**Actions :** Annuler, Enregistrer

**Figure 4.19: Modifier une équipe sélectionner**

La figure représente une interface permettant de répertorier les équipes existantes dans le système, offrant la possibilité de filtrer les équipes par gérance ou par nom. Cette fonctionnalité permet aux utilisateurs de visualiser les équipes de manière organisée et de les rechercher facilement en utilisant des critères spécifiques.

**Général**    **Électricité**    **Eau Potable**    **Assainissement**

**Ajouter**

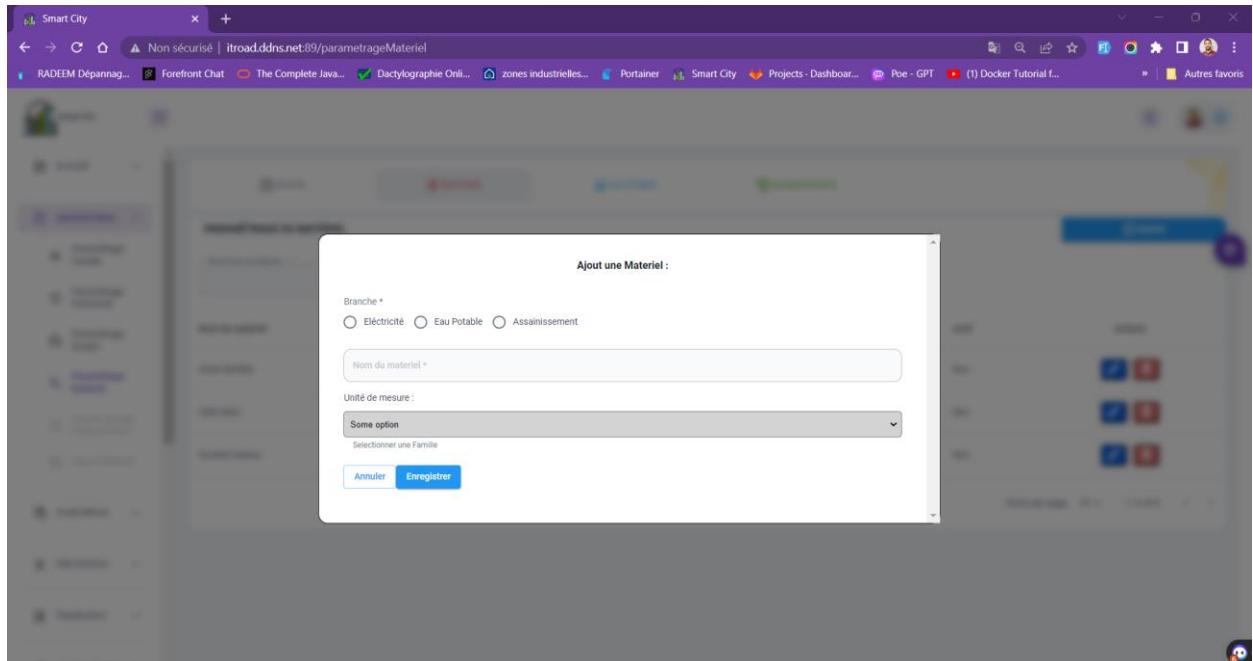
**LISTE DES EQUIPE**

Equipe	chef d'équipe	Members d'équipe	Utilisateur d'édition	Gerence	Spécialité d'équipe	N° de véhicule	superviseur	Actif	Actions
Equipe 1	Amela Benton	1. khallil bio 2. essadik idriss 3. Garrison Stephens 4. Phyllis Peck	tirgani badreddine	Electricité	Ratione cum quisquam	36	Rhona Houston	Non	
Equipe 2	Amela Benton	1. Mason Bean 2. Jessamine Finley	tirgani badreddine	Electricité	Commodo alias est om	63	Rhona Houston	Non	

Rows per page: 10 ▾ 1-2 of 2 < >

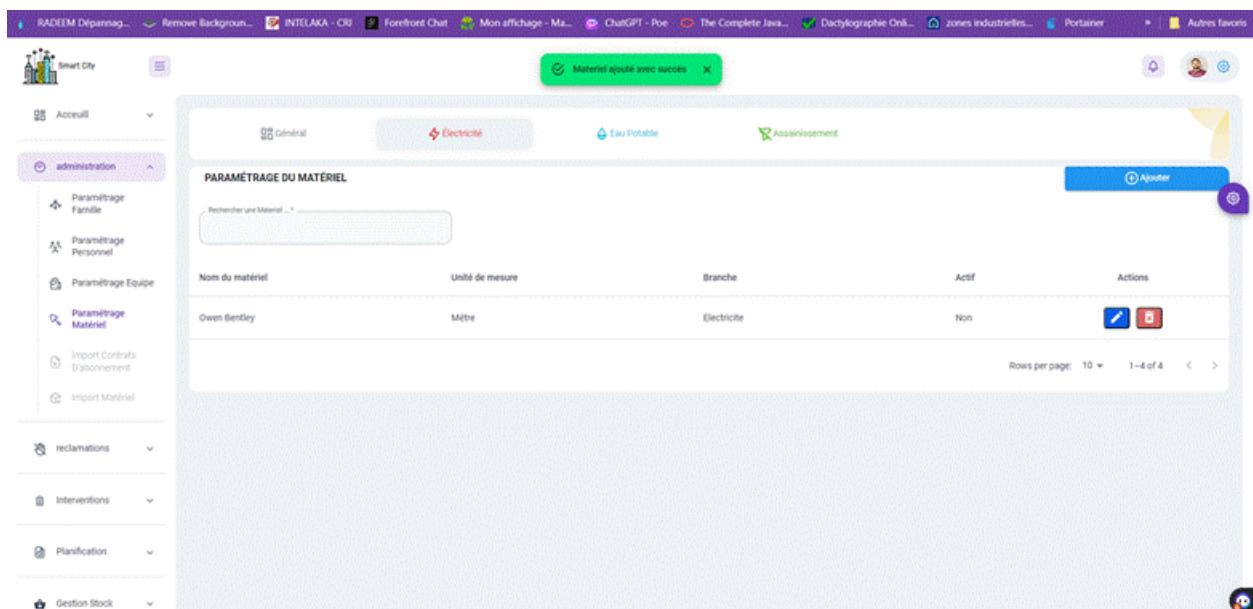
**Figure 4.20: Lister les Equipe existants**

La figure représente une interface permettant d'ajouter un nouveau matériel dans le système, en prenant en compte les variables suivantes : la branche, le numéro et l'unité du matériel. Cette fonctionnalité facilite l'intégration de nouveaux matériaux dans l'organisation.



**Figure 4.21 : Ajouter un nouveau matériel**

La figure représente une fenêtre contextuelle (popup) dédiée à l'ajout d'un nouveau matériel dans le système. Cette fenêtre contextuelle offre une expérience utilisateur ciblée et pratique pour intégrer de nouveaux matériaux à l'organisation.



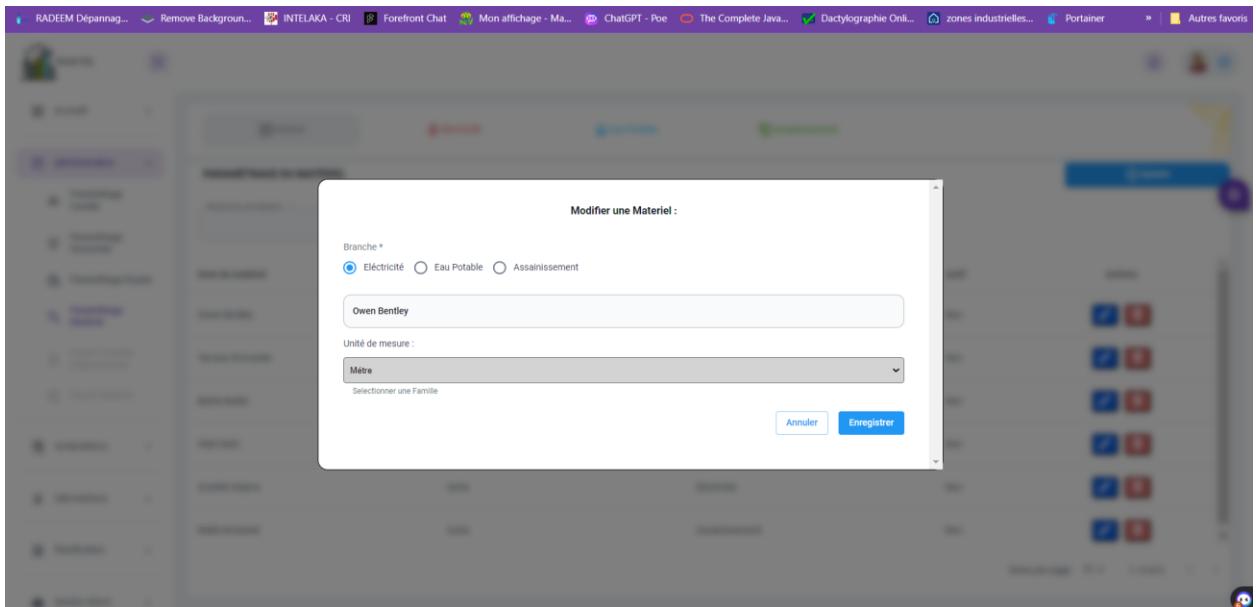
**Figure 4.22: Popup pour l'ajout du matériel**

La figure représente une interface permettant de répertorier les matériels existants dans le système, avec la possibilité de filtrer les matériels par gérance ou par nom. Cette fonctionnalité offre aux utilisateurs une vue organisée des matériels et la capacité de rechercher des matériels spécifiques en utilisant des critères de filtrage.

Nom du matériel	Unité de mesure	Branche	Actif	Actions
Owen Bentley	Mètre	Electricité	Non	
Tamara Schroeder	Unité	Eau Potable	Non	
Blythe Butler	Unité	Assainissement	Non	
Hilel Hahn	Autre	Electricité	Non	
Scarlett Adams	Autre	Electricité	Non	
Malik McDaniel	Autre	Assainissement	Non	

**Figure 4.23: Lister les matériels existants**

La figure représente une interface permettant la modification d'un matériel existant dans le système. Cette fonctionnalité offre aux utilisateurs la possibilité de mettre à jour les informations et les attributs d'un matériel spécifique, permettant ainsi une gestion plus précise et efficace des équipements.



**Figure 4.24: Modifier un Matériel**

La figure représente une interface permettant le filtrage du matériel par gérance dans le système. Cette fonctionnalité offre aux utilisateurs la possibilité de visualiser uniquement le matériel associé à une gérance spécifique, facilitant ainsi la gestion ciblée des équipements.

Nom du matériel	Unité de mesure	Branche	Actif	Actions
Owen Bentley	Mètre	Electricité	Non	
Hillel Hahn	Autre	Electricité	Non	
Scarlett Adams	Autre	Electricité	Non	

**Figure 4.25: Filtrage du matériel par gérances.**

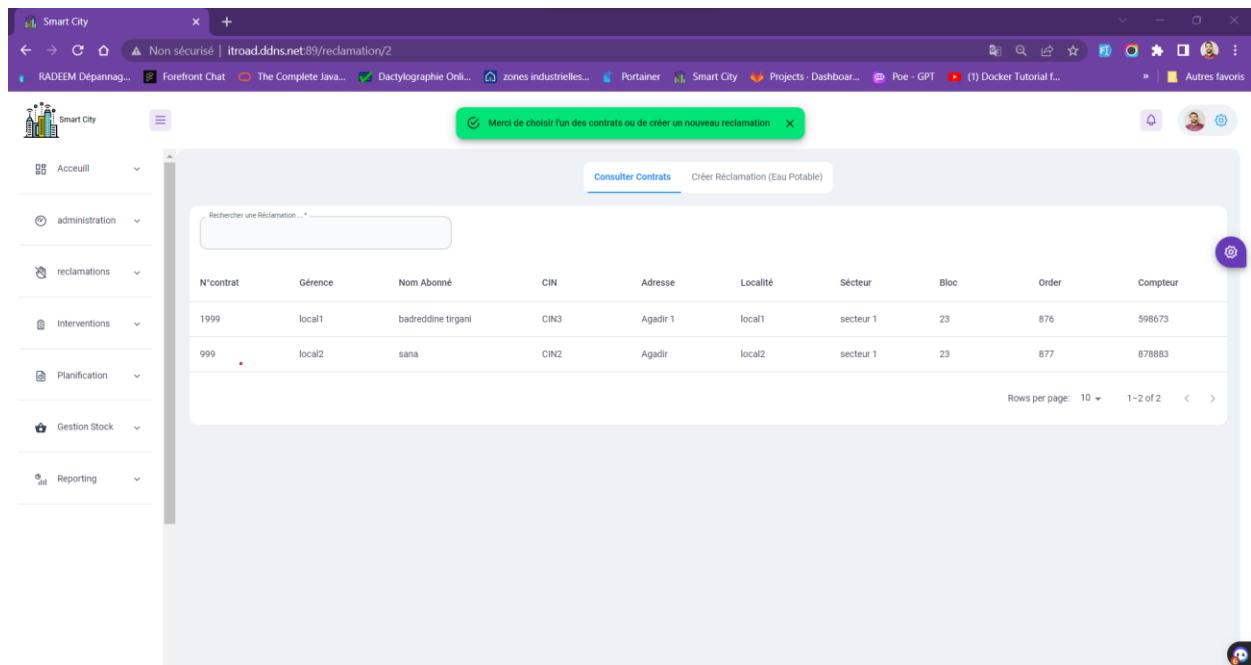
La figure représente une interface permettant d'ajouter une nouvelle réclamation par adresse dans le système, avec la possibilité de traiter les cas où le réclamateur ne possède pas de contrat. Cette fonctionnalité permet aux utilisateurs de soumettre des réclamations liées à une adresse spécifique, même si le réclamateur n'a pas de contrat enregistré.

### 4.3 Interfaces graphiques partie Gestion Réclamations :

The screenshot shows a web browser window for the 'Smart City' application. The URL is 'Non sécurisé | itroad.ddns.net:89/reclamation/2'. The page displays a form titled 'Créer Réclamation (Eau Potable)'. On the left, there is a sidebar with navigation links: Accueil, administration, reclamations, Interventions, Planification, Gestion Stock, and Reporting. The main form has tabs for 'Client' and 'Adresse', with 'Adresse' selected. It contains fields for Nom / Prénom, Tel. Portable, Catégorie Réclamation, Type du Client, Origine de réclamation, Objet de Réclamation, Famille, Type, Adresse de la Réclamation (with 'Agadir 1' entered), Observations, and Réclamation Urgente (with a checkbox). At the bottom are 'Annuler' and 'Enregistrer' buttons. The top right of the form has a 'Information Réclamation' section with a gear icon.

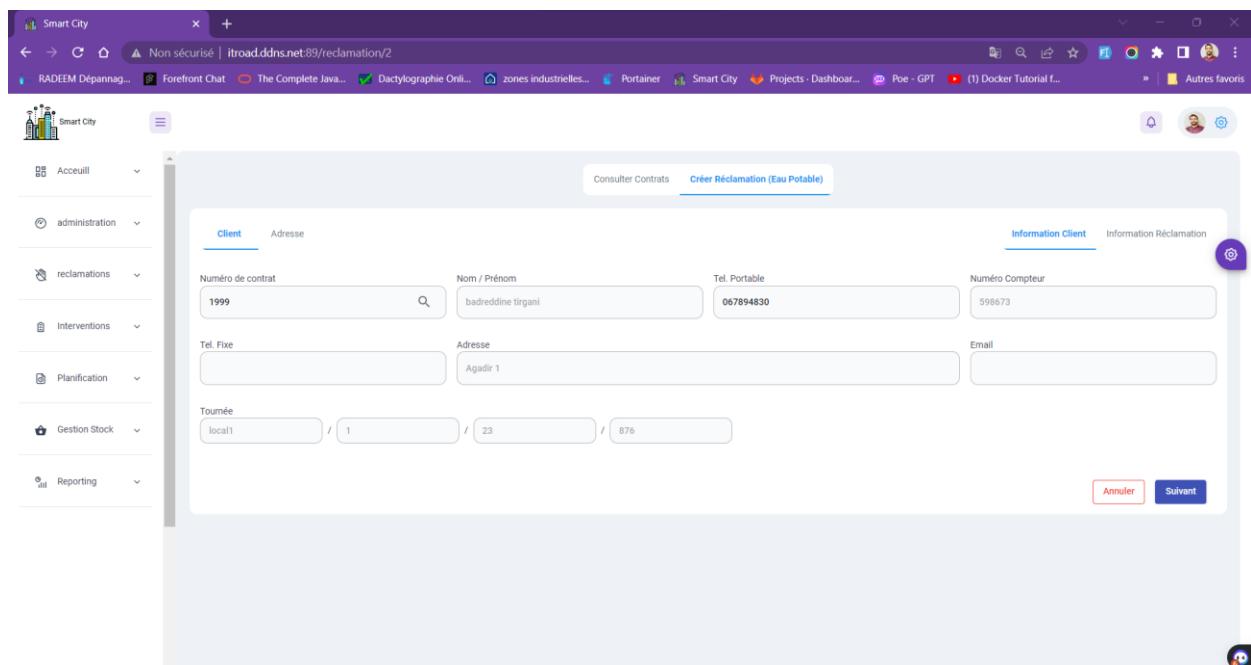
Figure 4.26: Ajouter une nouvelle déclamation par adresse

La figure représente la première étape d'une interface permettant d'ajouter une nouvelle réclamation en utilisant le numéro du contrat dans le système. Cette étape initiale permet aux utilisateurs de spécifier le contrat concerné avant de fournir plus de détails sur la réclamation.



**Figure 4.27: Ajouter une nouvelle réclamation par Numéro du contrat étape 1**

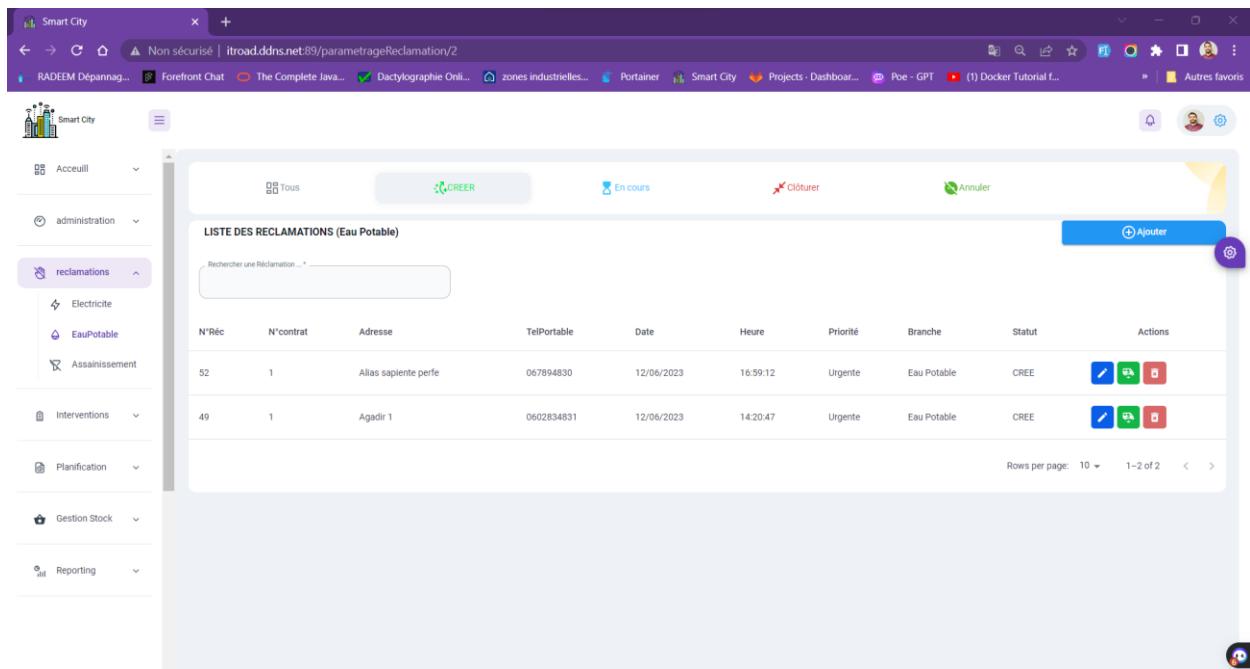
La figure représente la deuxième étape d'une interface guidée pour ajouter une nouvelle réclamation en utilisant le numéro du contrat dans le système. Cette étape permet aux utilisateurs de fournir des détails spécifiques sur la réclamation une fois que le contrat a été identifié.



**Figure 4.28: Ajouter une nouvelle réclamation par Numéro du contrat étape 2**

La figure représente une interface permettant de filtrer les réclamations par statuts dans le

système. Cette fonctionnalité offre aux utilisateurs la possibilité de visualiser les réclamations en fonction de leur état actuel, facilitant ainsi le suivi et la gestion des réclamations.



The screenshot shows the 'Smart City' application interface. On the left, there is a sidebar with navigation links: Accueil, administration, reclamations (selected), Electricite, EauPotable (highlighted in purple), Assainissement, Interventions, Planification, Gestion Stock, and Reporting. The main content area is titled 'LISTE DES RECLAMATIONS (Eau Potable)'. It features a search bar labeled 'Rechercher une Reclamation...'. Below the search bar are four filter buttons: 'Tous' (All), 'CREER' (Create), 'En cours' (In progress), and 'Clôturer' (Close). A blue 'Ajouter' (Add) button is located in the top right corner of the list area. The list table has columns: N°Réc, N°contrat, Adresse, TelPortable, Date, Heure, Priorité, Branche, Statut, and Actions. Two rows of data are visible:

N°Réc	N°contrat	Adresse	TelPortable	Date	Heure	Priorité	Branche	Statut	Actions
52	1	Alias sapiente perfe	067894830	12/06/2023	16:59:12	Urgente	Eau Potable	CREE	
49	1	Agadir 1	0602834831	12/06/2023	14:20:47	Urgente	Eau Potable	CREE	

At the bottom right of the list area, there are buttons for 'Rows per page' (set to 10), '1-2 of 2', and navigation arrows. The status bar at the bottom right shows the date as 'Mercredi 14 Juin 2023'.

**Figure 4.29:Filtrer les réclamations par statuts**

La figure représente une interface permettant de filtrer les réclamations existantes par statut dans le système. Cette fonctionnalité offre aux utilisateurs la possibilité de visualiser les réclamations en fonction de leur état actuel, ce qui facilite la gestion et le suivi des réclamations

**Figure 4.30: Filtrage des réclamations existantes par statut.**

La figure représente une interface permettant la suppression d'une réclamation donnée dans le système. Cette fonctionnalité offre aux utilisateurs la possibilité de retirer une réclamation spécifique, facilitant ainsi la gestion et l'organisation des réclamations.

**Figure 4.31: Suppression d'une réclamation donnée**

Cette figure représente la situation où plusieurs interventions sont ajoutées à une réclamation sans qu'il n'existe de contrats préalables. On peut observer plusieurs flèches reliant la réclamation au groupe d'interventions.

#### 4.4 Interfaces graphiques partie gestion Interventions :

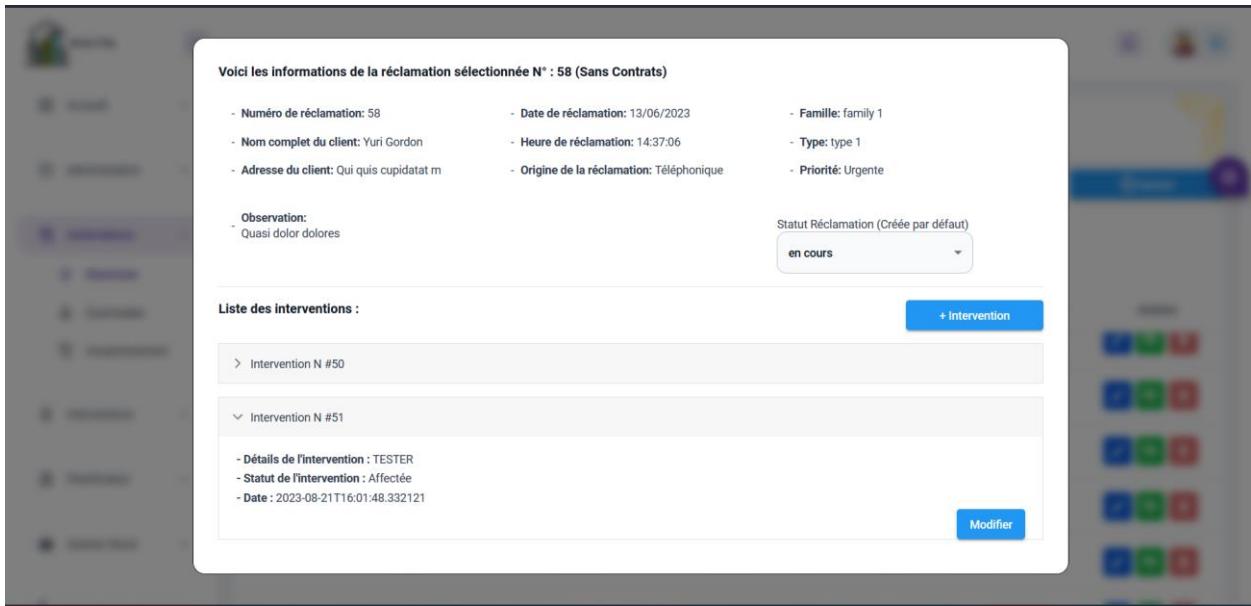


Figure 4.32: Ajouter plusieurs interventions dans une réclamations sans contrats

Cette figure illustre le processus d'ajout de plusieurs interventions à une réclamation préalablement encadrée par des contrats.

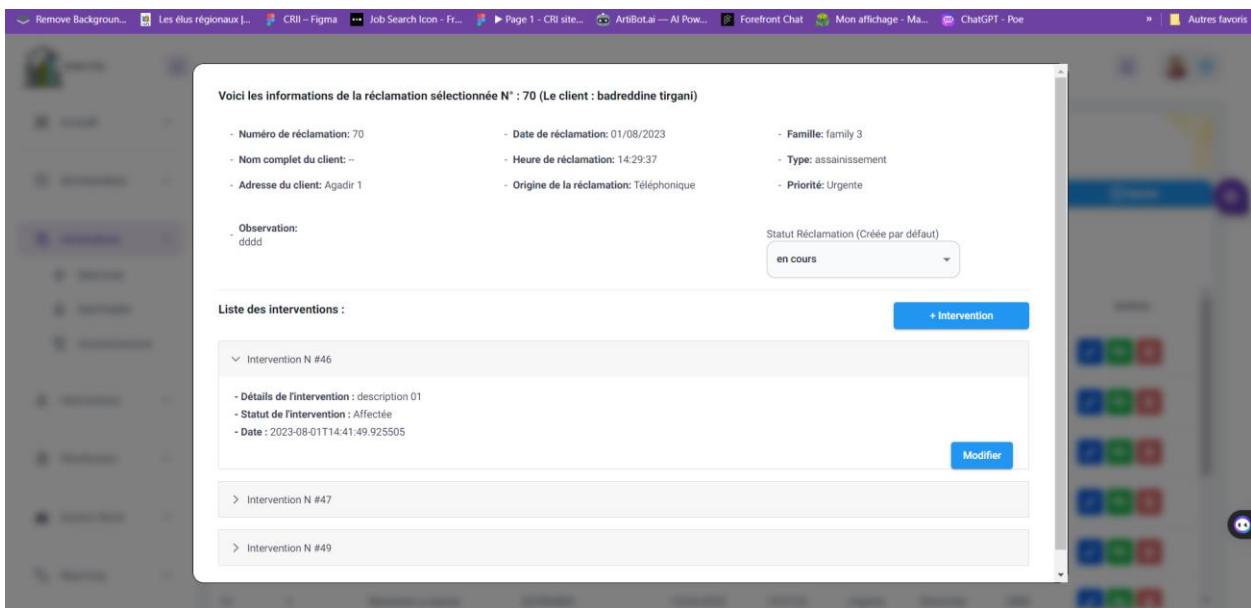


Figure 4.33: Ajouter plusieurs interventions dans une réclamations avec contrats

## 4.1 Interfaces graphiques partie Gestion des Interventions :

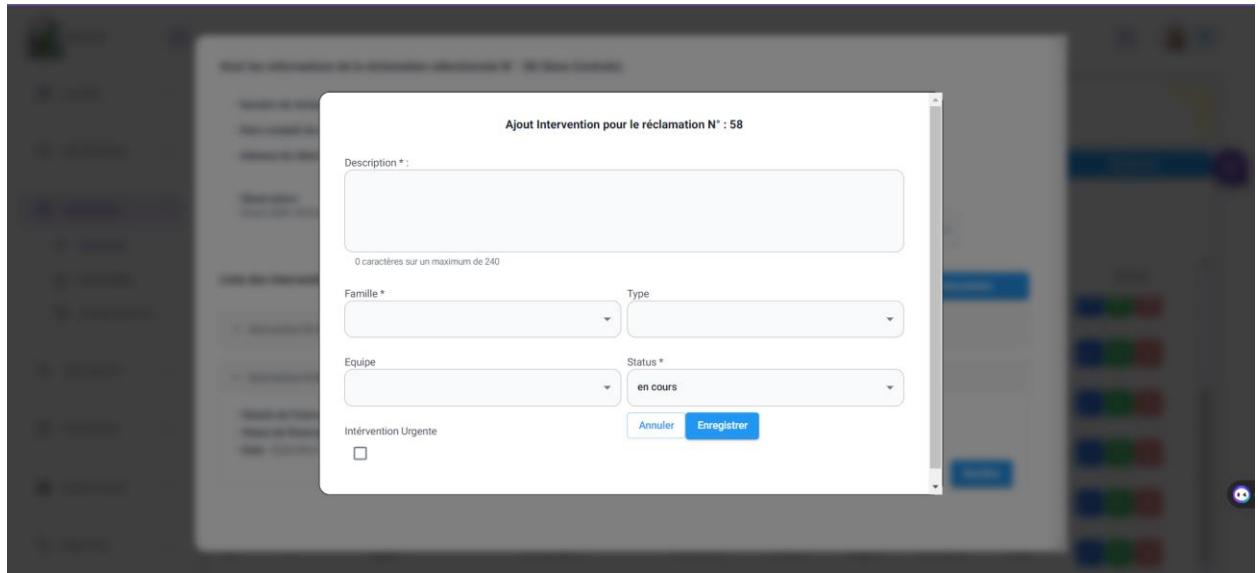


Figure 4.34: Ajouter une nouvelle intervention dans une réclamation

La figure représente une interface permettant la modification d'une réclamation existante qui inclut les détails du contrat client dans le système. Cette fonctionnalité offre aux utilisateurs la possibilité de mettre à jour les informations spécifiques liées à la réclamation tout en tenant compte des détails du contrat du client.

MODIFIER - Réclamation N'Réc: 62

Numéro de contrat: 999 Tournée: local2 / / 23 / 877

Nom / Prénom: sana Tel. Portable: 0602834831 Catégorie Réclamation\*: Réclamation Client Type du Client\*: Ministère

Origine de réclamation\*: SMS Objet de Réclamation: ddd Famille\*: family 1 Type: tester

Adresse de la Réclamation\*: Agadir Observations: eeeeeeee

Réclamation Urgente:

**Annuler** **Enregistrer**

**Figure 4.35: Modification d'une réclamation contenant le contrat client.**

La figure représente une interface permettant la modification d'une réclamation existante qui ne contient pas de contrat client dans le système. Cette fonctionnalité offre aux utilisateurs la possibilité de mettre à jour les informations spécifiques liées à la réclamation sans nécessiter les détails d'un contrat client.

MODIFIER - Réclamation N'Réc: 58

Nom / Prénom: Yuri Gordon Tel. Portable: Nobis voluptatibus m Catégorie Réclamation\*: Réclamation Client Type du Client\*: Normal

Origine de réclamation\*: Téléphonique Objet de Réclamation: Sunt asperiores et Famille\*: family 1 Type: type 1

Adresse de la Réclamation\*: Qui quis cupidatat m Observations: Quasi dolor dolores

Réclamation Urgente:

**Annuler** **Enregistrer**

**Figure 4.36: Modification d'une réclamation sans contrat client**

La figure représente une interface permettant de répertorier les interventions existantes dans le système, offrant la possibilité de filtrer les interventions par gérance. Cette fonctionnalité permet

aux utilisateurs de visualiser les interventions de manière organisée et de les rechercher facilement en utilisant des critères spécifiques.

The screenshot shows a web-based application interface for managing interventions. On the left, there is a sidebar with various menu items: Accueil, administration, reclamations, Interventions (selected), Electricité, Eau Potable, Assainissement, Intervention Par Equipe, Planification, and Gestion Stock. The main content area has tabs for Tous, CREER (Create), En cours (In progress), Clôturer (Close), and Annuler (Cancel). Below these tabs is a search bar labeled "Rechercher une Réclamation ...". The main table lists 18 interventions, each with columns for N°intervention, N°réclamation, Description, Equipe, Date, Heure, Adresse, Branche, Statut, and Actions (represented by a blue edit icon). The interventions are numbered 13 to 18, all belong to the Electricité branch, and have the status CREE.

N°intervention	N°réclamation	Description	Equipe	Date	Heure	Adresse	Branche	Statut	Actions
13	13	string	-	20/06/2023	16:43:42	string	Electricité	CREE	
14	14	string	-	20/06/2023	16:44:10	string	Electricité	CREE	
15	15	string	-	20/06/2023	16:46:31	string	Electricité	CREE	
16	16	string	-	20/06/2023	17:25:07	string	Electricité	CREE	
17	17	string	-	20/06/2023	17:33:45	string	Electricité	CREE	
18	18	string	-	20/06/2023	17:55:46	string	Electricité	CREE	

Figure 4.37: Lister les interventions par Gérances

# Conclusion Générale

Ce rapport résume le travail effectué durant mon projet de fin d'études au sein d'IT Road Consulting, pour l'obtention du titre « Ingénieur d'Etat » en Ingénierie d'Information et Réseaux, option MIAGE. Le projet a pour objectif d'améliorer la gestion des réclamations et des interventions dans les régies de distribution d'eau et d'électricité, dans le cadre de l'aménagement urbain.

Afin de mener à bien notre projet, nous avons commencé par nous familiariser avec le contexte général du projet en définissant la problématique à laquelle le projet doit répondre. Par la suite, nous avons pu déterminer les spécifications fonctionnelles que le système doit satisfaire. Nous avons enchaîné avec la conception et la modélisation UML pour finir avec une étude technique qui va nous permettre la mise en œuvre du projet.

Nous avons tiré un grand bénéfice de ce projet de fin d'études, aussi bien au niveau technique qu'au niveau professionnel. En effet, nous avons pu consolider notre formation notamment côté Développement Web et nous avons améliorer notre méthodologie de travail et développer notre manière de communication.

En guise de conclusion, cette expérience enrichissante n'est pour nous que le début d'une carrière que nous voulons exercer correctement

# Références

- [1]. Groupe. *IT Road Consulting*. [En ligne] <https://www.itroad.ma/groupe/>.
- [2]. Modele View Controller. *bdpedia*. [En ligne] <http://orm.bdpedia.fr/mvc.html>.
- [3]. Docs. *Docker*. [En ligne] <https://www.docker.com/>.
- [4]. *git --distributed-even-if-your-workflow-isnt*. *git*. [En ligne] <https://git-scm.com/>.
- [5]. pourquoi gitlab. *gitlab*. [En ligne] <https://about.gitlab.com/fr-fr/>.
- [6]. Jira. *BDM tools*. [En ligne] <https://www.blogdumoderateur.com/tools/jira/>.
- [7]. JWT. *jwt*. [En ligne] <https://jwt.io/>.

