

Ministère de l'Enseignement supérieur et de la Recherche scientifique

Université de Sousse

Institut Supérieur de Gestion de Sousse



PROJET DE FIN D'ÉTUDES

En vue de l'obtention du diplôme de licence en Business Computing avec une spécialisation en
Business Intelligence

Conception et développement d'un site web Étude et mise en place d'une solution d'éclairage public intelligent

Organisme d'accueil :



Élaboré par :

Fedi Belloumi

Encadrant pédagogique :

Dr Hamdi Hassen

Encadrant professionnel :

Mr Bassem Mechergui

Année universitaire : 2025–2026

Remerciements

Avant tout, je remercie Dieu le Tout-Puissant de m'avoir donné la santé, la volonté et le courage nécessaires pour mener ce travail à terme.

J'adresse mes sincères remerciements à l'Institut Supérieur de Gestion de Sousse pour la qualité de la formation offerte ainsi que pour les connaissances et les compétences acquises tout au long de mon parcours universitaire.

J'exprime également ma profonde gratitude à tous mes professeurs pour leurs conseils, leur disponibilité et leurs efforts durant mes années d'études.

Mes remerciements les plus respectueux s'adressent particulièrement à mon encadrant, Monsieur Hamdi Hassen, pour son accompagnement, son soutien, ses orientations précieuses et ses encouragements tout au long de la réalisation de ce projet.

Je remercie chaleureusement ma famille Belloumi pour son soutien moral, sa patience et sa confiance permanente.

Enfin, j'adresse mes remerciements à mes camarades et amis pour leur aide, leur esprit de collaboration et les agréables moments partagés durant cette expérience universitaire.

À tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce travail, j'exprime ma profonde reconnaissance.

Table des matières

Liste des figures

Liste des tableaux

Introduction Générale	1
1 Cadre général du projet	3
Plan	3
1.1 Introduction	3
1.2 Présentation de l'organisme d'accueil	4
1.2.1 Description de la société	4
1.2.2 Contact	4
1.3 Cadre général du projet	4
1.4 Analyse de l'existant	5
1.4.1 Étude de l'existant	5
1.4.2 Critique de l'existant	5
1.4.3 Solution proposée et objectif du projet	6
1.5 Méthodologie de gestion de projet	6
1.5.1 Présentation de Scrum	6
1.5.2 Présentation de processus unifié	7
1.5.3 Comparaison entre PU et SCRUM	8
1.5.4 Les rôles de SCRUM	9
1.5.5 Fonctionnement de SCRUM	9
1.6 Langages de modélisation	10
1.6.1 UML	10
1.7 Conclusion	11
2 Sprint 0 : Expression des besoins et étude technique	12

Plan	12
2.1 Introduction	12
2.2 Identification des besoins	13
2.2.1 Identification des acteurs	13
2.2.2 Besoins fonctionnels	13
2.2.3 Besoins non fonctionnels	15
2.2.4 Les besoins décisionnels	15
2.2.5 Pilotage du projet avec Scrum	16
2.2.6 Backlog produit	17
2.2.7 Diagramme de cas d'utilisation globale	19
2.2.8 Planification des sprints	21
2.3 Environnement du travail	21
2.3.1 Environnement matériel	21
2.4 Architecture générale de l'application	25
2.5 Diagramme de déploiement	26
2.6 Conclusion	27
 3 Sprint 1 : Étude,analyse et réalisation du Sprint 1	 28
Plan	28
3.1 Introduction	28
3.2 Backlog du Sprint 1	29
3.3 Spécification des besoins du Sprint 1	31
3.3.1 Diagramme de cas d'utilisation	32
3.3.2 Description textuelle des cas d'utilisations	33
3.4 Conception	36
3.4.1 Diagrammes de séquences sprint 1	36
3.4.2 Diagramme de séquence du scénario « Se connecter »	37
3.4.3 Diagramme de séquence du scénario « S'authentifier »	38
3.4.4 Diagramme de Classes du sprint 1	40
3.5 Réalisation et tests	41
3.5.1 Tests	51
3.6 Utilisation des outils de suivi Scrum	53
3.6.1 Scrum Board	53
3.6.2 Burn Down Chart	54
3.7 Conclusion	54

4 Sprint 2 :Étude, analyse et réalisation du Sprint 2	55
Plan	55
4.1 Introduction	55
4.2 Backlog du sprint 2	56
4.3 Spécification de besoins du sprint 2	57
4.3.1 Diagramme de cas d'utilisation	57
4.4 Conception	61
4.4.1 Diagrammes de séquences sprint 2	61
4.4.2 Diagramme de séquence du scénario « Créer une zone »	62
4.4.3 Diagramme de séquence du scénario « Consulter la carte interactive»	63
4.4.4 Diagramme de Classes du sprint 2	64
4.5 Réalisation et tests	65
4.5.1 Tests	70
4.6 Utilisation des outils de suivi Scrum	72
4.6.1 Scrum Board	72
4.6.2 Burn Down Chart	73
4.7 Conclusion	73
5 Sprint 3 : Étude, analyse et réalisation du Sprint 3	74
Plan	74
5.1 Introduction	74
5.2 Backlog du Sprint 3	75
5.3 Spécification de besoins du sprint 3	78
5.3.1 Diagramme de cas d'utilisation	78
5.4 Conception	82
5.4.1 Diagrammes de séquences du sprint 3	82
5.4.2 Diagramme de séquence du scénario « Supprimer une notification »	83
5.4.3 Diagramme de séquence du scénario «Modifier ticket »	84
5.4.4 Diagramme de Classes du sprint 3	86
5.5 Réalisation et tests	87
5.5.1 Tests	91
5.6 Utilisation des outils de suivi Scrum	93
5.6.1 Scrum Board	93
5.6.2 Burn Down Chart	94
5.7 Conclusion	94

6 Sprint 4 : Étude, analyse et réalisation du Sprint 4	95
Plan	95
6.1 Introduction	95
6.2 Backlog du Sprint	96
6.2.1 Diagramme de cas d'utilisation	100
6.2.2 Diagramme de séquence du scénario « Consulter le tableau du bord »	105
6.2.3 Diagramme de séquence du scénario « Communiquer avec le chatbot »	106
6.2.4 Diagramme de Classes du sprint 4	107
6.3 Spécification décisionnelle	108
6.3.1 Définition des objectifs	109
6.3.2 Construction du Tableau de bord	111
6.3.3 Chaîne BI	112
6.3.4 Data Mart	113
6.4 Réalisation et tests	121
6.5 Réalisation et tests	131
6.6 Utilisation des outils de suivi Scrum	132
6.6.1 Scrum Board	133
6.6.2 Burn Down Chart	134
6.7 Conclusion	134
6.8 Diagramme de calsse globale de l'application	135
6.9 Schema textuelle de base de données	136
 Conclusion Générale	 137
 Webographie	 137

Table des figures

1.1	Logo Treetronix [1]	4
1.2	Logo Scrum [19]	7
1.3	Rôles de Scrum [19]	9
1.4	Logo UML [2]	11
2.1	Diagramme des cas d'utilisation global	20
2.2	Logo Visual Studio Code [11]	22
2.3	Logo Node.js [3]	22
2.4	Logo Express.js [4]	22
2.5	Logo MongoDB [5]	23
2.6	Logo InfluxDB [6]	23
2.7	Logo React [7]	23
2.8	Logo TypeScript [8]	23
2.9	Logo Python [9]	24
2.10	Logo FastAPI [10]	24
2.11	Logo Draw.io [15]	24
2.12	Logo Postman [12]	24
2.13	Logo Git [13]	25
2.14	Logo GitHub [14]	25
2.15	Architecture physique du système	26
2.16	Diagramme de déploiement	27
3.1	Diagramme de cas d'utilisation du Sprint 1	32
3.2	Diagramme de séquence du scénario « Se connecter »	37
3.3	Diagramme de séquence du scénario « S'authentifier »	39
3.4	Diagramme de Classes du sprint 1	40
3.5	Interface d'accueil	42
3.6	Interface Consulter les statistiques publiques	43

3.7	Interface d'inscription	44
3.8	Interface vérification email	45
3.9	Interface de connexion	46
3.10	Interface d'authentification à deux facteurs	47
3.11	Interface mot de passe oublié	48
3.12	Interface réinitialiser le mot de passe	49
3.13	Interface changer le mot de passe	50
3.14	Interface gestion des utilisateurs	50
3.15	Scrum Board sprint 1	53
3.16	Burn down chart à la fin de 1er sprint	54
4.1	Diagramme de cas d'utilisation du Sprint 2	58
4.2	Diagramme de séquence du scénario « Créer une zone »	62
4.3	Diagramme de séquence du scénario « Consulter la carte interactive »	63
4.4	Diagramme de Classes du sprint 2	64
4.5	Interface carte interactive	65
4.6	Interface Gestion des Lampadaires	66
4.7	Interface Nouveau Lampadaire	66
4.8	Interface Modifier Lampadaire	67
4.9	Interface Gestion des Zones	68
4.10	Interface Nouvelle des Zones	68
4.11	Interface Modifier Zones	69
4.12	Tableau Scrum Sprint 2	72
4.13	Burn down à la fin de 2ème Sprint	73
5.1	Diagramme de cas d'utilisation du Sprint 3	78
5.2	Diagramme de séquence du scénario « Consulter et Supprimer une notification »	83
5.3	Diagramme de séquence du scénario « Modifier ticket »	85
5.4	Diagramme de Classes du sprint 3	86
5.5	Interface Notifications	88
5.6	Interface Alertes	89
5.7	Interface Tickets	90
5.8	Interface Données IoT	90
5.9	Tableau Scrum Sprint 3	93
5.10	Burn down à la fin de 3ème Sprint	94

6.1	Diagramme de cas d'utilisation du Sprint 4	100
6.2	Diagramme de séquence du scénario «Consulter le tableau du bord»	105
6.3	Diagramme de séquence du scénario «Diagramme de séquence Interagir avec le chatbot»	106
6.4	Diagramme de Classes du sprint 4	107
6.5	Schéma en étoile – Data Mart Smart Lighting	118
6.6	Interface Tableau de bord	122
6.7	Interface Tableau de bord	122
6.8	Interface Analytiques – Vue d'ensemble	123
6.9	Interface Analytiques – FACT Énergi	124
6.10	Interface Analytiques – FACT Pannes	125
6.11	Interface Analytiques – FACT Alertes	126
6.12	Interface Analytiques – FACT IoT Live	127
6.13	Interface Prédiction IA	128
6.14	Interface Optimisation IA	129
6.15	Interface Chatbot IA	130
6.16	Tableau Scrum Sprint 4	133
6.17	Burn down à la fin de 4ème Sprint	134
6.18	Diagramme de classe globale	135

Liste des tableaux

1.1	Étude de l'existant	5
1.2	Comparaison entre SCRUM et PU	8
2.1	Equipe Scrum	17
2.2	Backlog produit	19
2.3	Environnement matériel	21
3.2	Backlog détaillé du Sprint 1	31
3.3	Description du cas d'utilisation « S'inscrire »	34
3.4	Description du cas d'utilisation « S'authentifier »	35
3.5	Tests du Sprint 1	52
4.2	Backlog détaillé du Sprint 2	57
4.3	Description du cas d'utilisation « Créer une zone »	59
4.4	Description du cas d'utilisation « Consulter la carte interactive »	60
4.5	Tests du Sprint 2	71
5.1	Description du cas d'utilisation « Consulter et Supprimer les notifications »	77
5.2	Description du cas d'utilisation « Consulter et Supprimer les notifications »	80
5.3	Description du cas d'utilisation « Modifier un ticket »	81
5.4	Tests du Sprint 3	92
6.2	Backlog détaillé du Sprint 4	99
6.3	Description du cas d'utilisation « Consulter le tableau de bord »	102
6.4	Description du cas d'utilisation « Communiquer avec le chatbot »	104
6.5	Définition des objectifs	110
6.6	Construction du Tableau de bord	111
6.7	Table de faits – FAIT_CONSOMMATION	114
6.8	Table de dimension – DIM_TEMPS	114

6.9	Table de dimension – DIM_LAMPE	115
6.10	Table de dimension – DIM_ZONE	116
6.11	Table de dimension – DIM_USER	116
6.12	Table de dimension – DIMALERTE	117
6.13	Table de dimension – DIM_PREDICTION	117
6.14	Construction du Tableau de bord	120
6.15	Tests du Sprint 4	132

Introduction Générale

Dans un contexte mondial marqué par une urbanisation croissante et une demande énergétique en constante augmentation, les villes sont confrontées à des défis majeurs en matière de gestion intelligente des ressources. Le concept de *Smart City* (ville intelligente) s'inscrit dans cette dynamique en intégrant les technologies de l'information et de la communication afin d'optimiser les services urbains, d'améliorer la qualité de vie des citoyens et de réduire l'impact environnemental.

Parmi les domaines clés de la ville intelligente, l'éclairage public représente une composante essentielle. En effet, il joue un rôle crucial dans la sécurité des citoyens, la mobilité urbaine et le confort nocturne. Cependant, les systèmes d'éclairage traditionnels présentent plusieurs limites, notamment une consommation énergétique élevée, une absence d'adaptation aux conditions réelles, telles que le trafic ou la luminosité naturelle, ainsi qu'un manque d'automatisation dans la détection et la gestion des pannes.

Avec l'émergence de l'Internet des Objets (IoT), il est désormais possible de connecter les infrastructures urbaines, telles que les lampadaires, afin de collecter des données en temps réel et de permettre une gestion plus intelligente et dynamique. Ces données, combinées aux techniques d'intelligence artificielle (IA), offrent de nouvelles perspectives en matière de prédiction, d'optimisation énergétique et de maintenance préventive.

C'est dans ce contexte que s'inscrit notre projet de fin d'études, qui consiste à concevoir et développer une plateforme intelligente de gestion de l'éclairage public, basée sur l'IoT et l'intelligence artificielle. Cette plateforme vise à superviser le fonctionnement des lampadaires connectés, à centraliser les données issues du terrain, à détecter les anomalies, à assister les opérateurs dans la maintenance et à optimiser la consommation énergétique grâce à des mécanismes d'analyse et de prédiction.

La solution proposée repose sur une architecture moderne intégrant un backend basé sur Node.js et MongoDB pour la gestion des données métier, une base de données InfluxDB pour le stockage des données IoT en temps réel, un frontend React pour la visualisation et le pilotage, ainsi qu'un microservice Python développé avec FastAPI pour les traitements d'intelligence artificielle.

L'un des objectifs majeurs de ce projet est la mise en place d'un système intelligent capable non seulement de superviser l'éclairage public, mais aussi d'anticiper certains

comportements du système, comme la consommation énergétique future, le besoin d'éclairage ou encore le risque de panne. En complément, un module d'optimisation énergétique permet d'ajuster l'intensité lumineuse en fonction de plusieurs paramètres, tels que le trafic, la luminosité ambiante, la consommation estimée et le niveau de risque, afin de réduire les coûts énergétiques tout en maintenant un niveau de sécurité satisfaisant.

Pour mener à bien ce projet, nous avons adopté une approche Agile en suivant la méthodologie **SCRUM**, fondée sur une gestion itérative et incrémentale du travail à travers plusieurs sprints. Cette méthode nous a permis d'assurer un développement progressif et maîtrisé des fonctionnalités, tout en favorisant une communication continue entre les différentes parties prenantes, notamment les encadrants et les développeurs. Sur le plan technique, la solution proposée repose sur une architecture moderne et modulaire intégrant un backend en **Node.js** [3] avec **Express** [4], une base de données **MongoDB** [5] pour les données métier, une base **InfluxDB** [6] pour les données IoT en temps réel, un frontend en **React** [7], ainsi qu'un microservice **Python FastAPI** [9, 10] dédié aux traitements d'intelligence artificielle.

Ce rapport retrace l'ensemble des étapes suivies pour la conception et la réalisation de cette plateforme intelligente de gestion de l'éclairage public. Il est structuré en plusieurs chapitres :

- Le premier chapitre présente le cadre général du projet, incluant le contexte, la problématique, l'étude de l'existant ainsi que la solution proposée.
- Le deuxième chapitre est consacré au Sprint 0, portant sur l'expression des besoins fonctionnels et non fonctionnels, l'identification des acteurs, l'étude de l'environnement de travail et la modélisation initiale du système.
- Le troisième chapitre détaille les travaux réalisés au cours du Sprint 1, notamment la mise en place du socle du système, la gestion des zones, des lampadaires, des alertes et des tickets, ainsi que l'intégration des premières fonctionnalités backend.
- Le quatrième chapitre présente les réalisations du Sprint 2, en mettant l'accent sur le tableau de bord, l'exploitation des données IoT, les visualisations analytiques et les indicateurs de suivi.
- Le cinquième chapitre expose les travaux du Sprint 3, consacrés à l'intégration de l'intelligence artificielle, notamment la prédiction de la consommation énergétique, la prédiction du besoin d'éclairage, l'estimation du risque de panne, ainsi que l'optimisation énergétique intelligente.

Enfin, une conclusion générale vient clore ce rapport en mettant en lumière les principaux apports du projet, les compétences acquises au cours de sa réalisation, ainsi que les perspectives d'amélioration et d'évolution futures de la solution proposée.

Chapitre 1

Cadre général du projet

Plan

1. Introduction
2. Contexte du projet
3. Problématique
4. Étude de l'existant
5. Critique de l'existant
6. Solution proposée
7. Méthodologie adoptée
8. Conclusion

1.1 Introduction

Ce premier chapitre introductif vise à placer le projet dans son contexte global. Nous commencerons par présenter l'organisme d'accueil. Par la suite, nous présenterons le projet en lui-même. Puis, nous exposerons la problématique abordée. Ensuite, nous réaliserons une étude de l'existant afin d'identifier les limites des solutions actuelles. Enfin, nous présentons la solution proposée ainsi que la méthodologie adoptée pour mener à bien ce projet.

1.2 Présentation de l'organisme d'accueil

1.2.1 Description de la société

Treetronix Technology est une startup spécialisée en développement et l'intégration de l'électronique embarquée, développement des systèmes d'information, développement des systèmes robotiques et développement des solutions pour l'Internet des objets (IoT).

1.2.2 Contact

- **Site web** : <http://www.treetronix.com/>
- **Téléphone** : +216 71 111 100
- **Siège social** : Immeuble Nefissa , 31 Rue des Entrepreneurs, charguia 2, Tunis 2035



FIGURE 1.1 – Logo Treetronix [1]

1.3 Cadre général du projet

Ce projet a été réalisé dans le cadre de mon projet de fin d'études en vue de l'obtention du diplôme de Licence en Informatique de Gestion à l'Institut Supérieur de Gestion de Sousse. Ce projet a pour objectif principal la conception et le développement d'une application web intégrant une plateforme de Business Intelligence (BI). Avec l'évolution rapide des technologies numériques et la croissance des villes, le concept de Smart City s'impose comme une solution innovante pour améliorer la gestion des infrastructures urbaines. L'objectif principal est d'exploiter les technologies de l'information et de la communication afin d'optimiser les ressources, de réduire les coûts et d'améliorer la qualité de vie des citoyens. Dans ce contexte, l'éclairage public constitue un élément essentiel des infrastructures urbaines. Cependant, les systèmes traditionnels sont souvent inefficaces, énergivores et difficiles à maintenir. L'intégration de l'Internet des Objets (IoT) permet de connecter les lampadaires et de collecter des données en temps réel, ouvrant ainsi la voie à une gestion intelligente et automatisée de l'éclairage public.

1.4 Analyse de l'existant

L'analyse de l'existant constitue une étape essentielle pour comprendre le fonctionnement du système actuel. Notre mission consiste à identifier les insuffisances et les limites du système en place, à proposer des solutions adaptées et à définir les objectifs à atteindre afin d'améliorer et d'optimiser le système existant.

1.4.1 Étude de l'existant

Étude de l'existant

Nom du site	Désignation	Avantages	Inconvénients
StreetLight Control Software	Plateforme de gestion de l'éclairage public basée sur le cloud	Contrôle à distance, tableaux de bord en temps réel, gestion des alertes	Coût élevé, pas de module IA intégré, dépendance au cloud
StreetlightOps	Solution de supervision et de maintenance des réseaux d'éclairage public	Suivi des interventions, gestion des tickets de maintenance, rapports automatiques	Interface peu intuitive, absence de prédiction énergétique, pas de carte interactive
Liveable Cities	Plateforme smart city intégrant la gestion de l'éclairage urbain	Vision globale smart city, indicateurs environnementaux, intégration IoT	Complexité d'utilisation, coût très élevé, solution non spécialisée en éclairage
Smart Lighting Platform	Plateforme intelligente de gestion de l'éclairage public développée dans ce projet	Module IA intégré, prédiction énergétique, carte interactive, tableau de bord analytique, open source	En cours de développement, déploiement limité à l'environnement de test

TABLE 1.1 – Étude de l'existant

1.4.2 Critique de l'existant

À partir de cette étude, plusieurs limites peuvent être identifiées :

- Absence d'intelligence prédictive.
- Manque d'optimisation énergétique en temps réel.
- Faible intégration des données IoT.

- Difficulté de maintenance et de supervision centralisée.

Ces insuffisances justifient le besoin d’une solution plus avancée, intégrant l’IoT et l’intelligence artificielle.

1.4.3 Solution proposée et objectif du projet

Pour répondre à ces problématiques, nous proposons le développement d’une plateforme intelligente de gestion de l’éclairage public.

Cette solution repose sur :

- L’utilisation de capteurs IoT pour collecter des données en temps réel.
- Une plateforme backend pour la gestion des lampadaires, des zones, des alertes et des tickets.
- Un dashboard interactif permettant la visualisation des données et le contrôle du système.
- Un module d’intelligence artificielle permettant la prédiction et l’optimisation énergétique.

L’objectif est de créer un système capable d’ajuster automatiquement l’intensité lumineuse, de détecter les anomalies et de proposer des recommandations intelligentes pour améliorer l’efficacité énergétique.

1.5 Méthodologie de gestion de projet

Les méthodologies de gestion de projet regroupent un ensemble structuré de techniques et de procédures permettant de planifier, organiser et piloter efficacement un projet. Elles visent à optimiser l’utilisation des ressources tout en réduisant les risques liés à son déroulement. La réussite d’un projet repose également sur une collaboration efficace entre l’ensemble des parties prenantes. Ainsi, il est indispensable d’adopter une démarche de travail claire, cohérente et bien structurée.

1.5.1 Présentation de Scrum

Scrum est un cadre de gestion de projet innovant qui met l’accent sur la collaboration, l’adaptabilité et le développement itératif afin d’améliorer la réussite des projets. Au coeur de cette méthode se trouve l’équipe Scrum, composée de membres qui travaillent en étroite collaboration pour concevoir et livrer des logiciels ou d’autres types de livrables à forte valeur ajoutée pour l’entreprise. Face à des problématiques complexes et évolutives, Scrum permet de les aborder de manière progressive, efficace et créative, en assurant la livraison continue de produits à forte valeur. Ce cadre repose sur une approche itérative

et incrémentale, où le produit est développé par étapes successives appelées « sprints ». Chaque sprint permet de livrer une partie fonctionnelle du produit, répondant ainsi aux besoins réels du client. Scrum offre ainsi une méthode flexible, évolutive et efficace pour la gestion de projets.

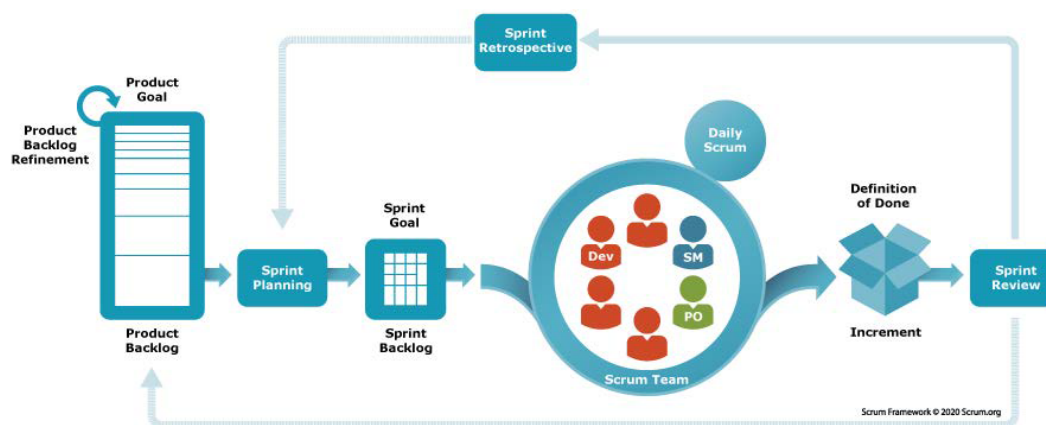


FIGURE 1.2 – Logo Scrum [19]

1.5.2 Présentation de processus unifié

Définition de processus unifié

Le Processus Unifié (PU) se présente comme une famille de méthodologies dédiées au développement logiciel orienté objet. Son approche se distingue par sa nature itérative et incrémentale, s'articulant autour de plusieurs concepts clés :

- **Itératif et Incrémental** : Le développement progresse par cycles successifs, chaque itération produisant une version fonctionnelle et améliorée du logiciel. Cette approche permet une meilleure gestion des changements et une adaptation aux besoins des utilisateurs.
- **Piloté par les cas d'utilisation** : Les fonctionnalités du logiciel sont définies et priorisées en fonction des cas d'utilisation réels des utilisateurs. Cette focalisation garantit que le logiciel répond aux besoins et attentes des utilisateurs finaux.
- **Accent sur l'architecture et les modèles UML** : L'architecture du logiciel est au cœur du processus, définie et affinée au fil des itérations. L'utilisation de modèles UML permet de visualiser et de documenter l'architecture du logiciel, facilitant la communication et la compréhension entre les différents acteurs du projet.

Les avantages et les inconvénients

Le processus unifié se caractérise par plusieurs avantages :

- Une structure claire avec un processus bien défini.
- Une bonne traçabilité ainsi qu'une documentation complète.

— Une adaptation efficace aux projets complexes et à long terme.

Cependant, il présente également certaines limites :

— Un manque de flexibilité et d'adaptabilité face aux changements.

— Un processus parfois lourd et bureaucratique.

— Une forte dépendance à une planification détaillée.

1.5.3 Comparaison entre PU et SCRUM

TABLE 1.2 – Comparaison entre SCRUM et PU

	SCRUM	PU
Philosophie et structure	Approche itérative et agile, privilégiant des sprints courts et des livraisons fréquentes.	Approche méthodique et séquentielle, découpant le projet en phases distinctes (conception, analyse, implémentation, tests, etc.).
Rôle des équipes	Des équipes auto-organisées et flexibles, avec des rôles moins formels (Scrum Master, Product Owner, équipe de développement).	Une armée bien structurée avec des rôles précis (chef de projet, analystes, développeurs, testeurs).
Gestion du changement	Adaptabilité et flexibilité accrues grâce aux sprints courts et à la rétrospective.	Changements difficiles à intégrer en raison de la structure séquentielle.
Estimation et planification	Estimation approximative au début de chaque sprint, avec des ajustements possibles en cours de route.	Estimation détaillée du temps et des ressources avant le démarrage du projet.
Outils et documentation	Outils de gestion de projet traditionnels et documentation complète.	Outils légers pour la gestion du backlog et le suivi des sprints.

C'est pourquoi j'ai choisi la méthode SCRUM. Particulièrement adaptée à notre objectif, elle favorise une collaboration efficace et efficiente grâce à : Des cycles courts et itératifs, Une meilleure communication et implication de tous les participants, une gestion des risques optimisée

1.5.4 Les rôles de SCRUM

Scrum repose sur trois rôles principaux afin d'assurer une collaboration efficace et une livraison réussie du projet :

- « Product Owner » : Il représente les besoins du client et des parties prenantes. Il est chargé de prioriser les tâches, de définir le Product Backlog, de valider les livrables et d'ajuster la vision du produit en fonction des retours.
- « Scrum Master » : Il veille au bon respect de la méthodologie Scrum. Il anime les réunions, facilite la communication entre les membres de l'équipe et aide à lever les obstacles tout en favorisant l'amélioration continue.
- « Équipe de développement » : Elle regroupe les développeurs, testeurs et autres intervenants techniques. Elle réalise les tâches du sprint, s'auto-organise et collabore étroitement afin d'atteindre les objectifs fixés



FIGURE 1.3 – Rôles de Scrum [19]

1.5.5 Fonctionnement de SCRUM

Artefacts :

- Product Backlog : Il représente la feuille de route du projet. Il regroupe et priorise l'ensemble des fonctionnalités à développer.
- Sprint Backlog : Il s'agit d'un plan de travail détaillé qui contient les fonctionnalités sélectionnées pour le sprint en cours ainsi que les tâches nécessaires à leur réalisation.
- Sprint Burndown Chart : C'est un graphique de suivi qui permet de visualiser l'avancement du sprint et d'identifier rapidement les écarts ou points de blocage.

Cérémonies :

- Sprint Planning : L'équipe se réunit afin de planifier le sprint, sélectionner les éléments du backlog et estimer l'effort nécessaire pour leur réalisation.
- Daily Scrum : Réunion quotidienne courte durant laquelle chaque membre partage son avancement, les difficultés rencontrées et les actions prévues.
- Sprint Review : L'équipe présente le travail réalisé aux clients et parties prenantes afin de recueillir leurs retours et ajuster le produit si nécessaire.
- Sprint Rétrospective :

Moment d'analyse collective permettant d'évaluer le sprint écoulé et d'identifier les améliorations à apporter pour les prochains cycles. En résumé, Scrum est une méthode agile puissante permettant de gérer efficacement les projets complexes et évolutifs. Elle favorise la collaboration, l'adaptation continue et l'amélioration progressive, tout en assurant la livraison régulière de produits à forte valeur ajoutée.

1.6 Langages de modélisation

1.6.1 UML

La modélisation étant un élément clé du cycle de vie de toute application, il est crucial de sélectionner un langage de modélisation qui réponde à vos besoins et exigences spécifiques. C'est dans cette optique que nous avons choisi UML (Unified Modeling Language). UML est un langage visuel puissant composé d'une série de diagrammes, chacun offrant une perspective unique sur le projet. Cela permet d'avoir une compréhension holistique et précise de l'application à développer. La flexibilité d'UML réside dans la variété de ses diagrammes, qui permettent de représenter différents aspects de l'application, tels que son fonctionnement, son déploiement, les actions possibles, et bien plus encore. Nous avons choisi UML également en raison de son statut de standard de modélisation pour les applications orientées objet. Son adoption à grande échelle favorise une meilleure collaboration et une compréhension commune entre les équipes. Sa flexibilité, sa puissance et son adoption universelle en font un choix judicieux pour la modélisation de votre application .



FIGURE 1.4 – Logo UML [2]

Pourquoi choisir UML ?

- Amélioration de la communication et de la compréhension : UML fournit un langage commun entre les équipes et les parties prenantes, ce qui facilite la collaboration et la compréhension globale du système.
- Meilleure conception et architecture : Il permet de visualiser et d'analyser l'architecture du système afin d'identifier les faiblesses et d'améliorer la qualité de la conception.
- Développement et maintenance facilités : La documentation claire et structurée offerte par UML simplifie le développement ainsi que la maintenance du système.
- Réduction des risques et des coûts : La détection précoce des erreurs de conception permet de limiter les risques et de réduire les coûts liés aux corrections ultérieures.
- Amélioration de la qualité du logiciel : UML contribue à la création de logiciels plus robustes, fiables et évolutifs.

1.7 Conclusion

Dans cette section, nous avons effectué une analyse détaillée du projet. Nous avons d'abord présenté l'organisme d'accueil, puis étudié la situation actuelle ainsi que les solutions proposées. Ensuite, nous avons exposé la méthodologie adoptée pour la réalisation de notre application. Les détails de cette étude seront développés dans le chapitre suivant.

Chapitre 2

Sprint 0 : Expression des besoins et étude technique

Plan

1. Introduction
2. Identification des besoins
3. Environnement du travail
4. Architecture générale de l'application
5. Diagramme de déploiement
6. Conclusion

2.1 Introduction

Ce chapitre traite de la phase d'analyse, une étape clé pour garantir la réussite du projet. Nous commençons par identifier les acteurs ainsi que leurs rôles, puis nous priorisons et validons les besoins fonctionnels, non fonctionnels et les besoins décisionnels afin d'atteindre les objectifs fixés. Ensuite, nous définissons la hiérarchisation des fonctionnalités à développer en sprints, déterminons leur durée et répartissons les tâches entre les différentes équipes. Enfin, nous présentons l'environnement de travail ainsi que l'architecture adoptée pour notre projet.

2.2 Identification des besoins

2.2.1 Identification des acteurs

Le système met en jeu plusieurs acteurs principaux :

- **Administrateur** : responsable de la gestion globale du système, des zones, des lampadaires, des utilisateurs, des alertes et du suivi global.
- **Décideur** :
- **Opérateur** :
- **Visiteur** :
- **Citoyen** :
- **Technicien** : responsable du traitement des tickets et des interventions de main tenance.
- **Système IoT** : responsable de l'envoi automatique des données issues des lampadaires connectés.

2.2.2 Besoins fonctionnels

Les besoins fonctionnels du système sont les suivants :

1. **S'authentifier** : Ce CU permet à tout utilisateur de s'identifier via ses identifiants afin d'accéder de manière sécurisée à son espace personnel.
2. **Consulter la page d'accueil** : Ce CU permet à l'utilisateur de visualiser la page principale de l'application, présentant un aperçu général du système et les accès rapides aux différentes fonctionnalités.
3. **Consulter les statistiques publiques** : Ce CU permet à tout visiteur de consulter les statistiques générales du réseau d'éclairage sans nécessiter d'authentification.
4. **Consulter les analytiques** : Ce CU permet à l'utilisateur connecté d'accéder à des analyses détaillées des données du système .
5. **Créer un compte** : Ce CU permet à un nouvel utilisateur de s'inscrire sur la plateforme en renseignant ses informations personnelles afin d'obtenir un accès au système.
6. **Consulter la prédiction IA** : Ce CU permet à l'utilisateur de visualiser les prévisions générées par les modèles d'intelligence artificielle.
7. **Consulter l'optimisation IA** : Ce CU permet à l'utilisateur de consulter les recommandations d'optimisation produites par l'IA.
8. **Consulter la carte interactive** : Ce CU permet à l'utilisateur de visualiser la localisation géographique des lampadaires .

9. **Réinitialisation du mot de passe** : Ce CU permet à un utilisateur ayant oublié son mot de passe de le réinitialiser via une procédure de code de confirmation.
10. **Déconnecter** : Ce CU permet à l'utilisateur de mettre fin à sa session en cours et de quitter son espace personnel de manière sécurisée.
11. **Gérer les notifications** : Ce CU permet à l'utilisateur de consulter, supprimer les notifications reçues.
12. **Gérer les utilisateurs** : Ce CU permet à l'administrateur d'ajouter, désactiver et supprimer les comptes utilisateurs, ainsi que de gérer leurs rôles et leurs droits d'accès au système.
13. **Gérer les alertes** : Ce CU permet à l'administrateur de consulter les alertes du système.
14. **Gérer les paramètres** : Ce CU permet à l'administrateur de configurer les paramètres généraux du système, tels que les seuils de consommation.
15. **Gérer les zones** : Ce CU permet à l'administrateur de créer, consulter, modifier et supprimer les zones géographiques d'éclairage, afin d'organiser le réseau en secteurs gérables.
16. **Gérer les lampadaires** : Ce CU permet à l'administrateur d'ajouter, modifier, rechercher, consulter et supprimer les lampadaires du réseau.
17. **Gérer les données IoT** : Ce CU permet au système de collecter, traiter et stocker en temps réel les données issues des capteurs IoT installés sur les lampadaires.
18. **Gérer les tickets** : Ce CU permet de gérer automatiquement les tickets de maintenance lors de la détection d'une anomalie, en assurant leur suivi depuis l'ouverture jusqu'à la résolution.
19. **Consulter le tableau de bord** : Ce CU permet à l'utilisateur de visualiser en temps réel les données du système sous forme de graphiques, courbes et statistiques, offrant une vue synthétique de l'état du réseau d'éclairage.
20. **Communiquer avec le chatbot** : Ce CU permet à l'utilisateur d'interagir avec un assistant conversationnel intégré, capable de répondre aux questions, d'orienter vers les fonctionnalités du système et de fournir des analyses basées sur les modèles d'IA.
21. **Changer mot de passe** : Ce CU permet à l'utilisateur de changer mot de passe.

2.2.3 Besoins non fonctionnels

Les exigences non fonctionnelles concernent des aspects visibles ou ressentis par l'utilisateur, sans décrire directement les fonctionnalités du système. Elles englobent des contraintes liées à l'environnement technique et à l'implémentation, ainsi que des critères de performance, d'extensibilité et de fiabilité. Ci-dessous sont présentées les exigences non fonctionnelles de notre système.

- **Sécurité d'accès au système** : Le système doit garantir un accès sécurisé aux utilisateurs afin de protéger les données et les fonctionnalités contre toute utilisation non autorisée.
- **Performance** : Le système doit assurer un traitement rapide et efficace des données en temps réel afin de garantir une bonne réactivité.
- **Simplicité d'utilisation de l'interface** : L'interface doit être intuitive, claire et facile à utiliser pour faciliter la navigation des utilisateurs.
- **Fiabilité des données** : Les données collectées et affichées doivent être exactes, cohérentes et fiables afin d'assurer une bonne prise de décision.
- **Évolutivité du système** : Le système doit être flexible et capable d'évoluer afin d'intégrer de nouvelles fonctionnalités et répondre aux futurs besoins.

2.2.4 Les besoins décisionnels

Pour l'administrateur :

- Supervision globale de l'état du réseau d'éclairage public (zones, lampadaires, pannes, alertes, tickets)
- Suivi du taux de disponibilité (Uptime) et de la puissance totale consommée en kWh
- Analyse de la tendance énergétique du réseau sur une période sélectionnée
- Visualisation de la consommation par zone géographique
- Suivi des alertes récentes et des incidents critiques du réseau
- Gestion et supervision des comptes utilisateurs de la plateforme

Pour le décideur :

- Analyse de l'évolution temporelle de la consommation énergétique via la courbe de tendance (FACT Énergie)
- Comparaison des performances énergétiques par zone géographique via le diagramme en barres et le camembert de répartition (FACT Énergie)
- Identification des zones les plus critiques et analyse de la criticité par zone (FACT Pannes)
- Consultation des prédictions de consommation énergétique sur 7 jours générées par le microservice IA (Prédiction IA)

- Exploitation des recommandations d'optimisation de l'intensité lumineuse et estimation des économies réalisables (Optimisation IA)

Pour l'opérateur :

- Surveillance en temps réel de l'état des lampadaires via les métriques IoT collectées depuis InfluxDB (FACT IoT Live)
- Consultation des alertes actives, leur répartition par type et par sévérité (FACT Alertes)
- Suivi du taux de résolution des alertes et du temps moyen de résolution MTTR (FACT Pannes)
- Consultation des recommandations d'optimisation de l'intensité lumineuse par zone (Optimisation IA)
- Suivi de l'activité détaillée de chaque capteur via le tableau de données IoT en temps réel (FACT IoT Live)

Pour le technicien :

- Consultation des lampadaires en panne ou en cours de maintenance via le tableau d'activité des capteurs (FACT IoT Live)
- Suivi de l'évolution des pannes par jour et identification des zones les plus critiques (FACT Pannes)
- Consultation de la répartition des pannes par sévérité (High, Critical, Medium) pour prioriser les interventions (FACT Pannes)
- Accès aux données IoT en temps réel pour le diagnostic des équipements défaillants (tension, courant, intensité, température) (FACT IoT Live)

2.2.5 Pilotage du projet avec Scrum

Equipe SCRUM

Cette section présente les différents intervenants ayant contribué aux différentes phases du projet ainsi qu'à la rédaction du rapport de stage.

L'équipe Scrum est une équipe agile composée de plusieurs membres travaillant en étroite collaboration au cours de sprints afin de développer un produit à forte valeur ajoutée.

Chaque équipe Scrum repose sur trois rôles essentiels, qui peuvent être assurés par une ou plusieurs personnes :

Rôle scrum	Personne affectée	Ses responsabilités
Scrum master	Mr Bassem Merchergui	A travers différentes réunions, il est chargé d'aider à résoudre les problèmes, aussi d'assurer la méthode de Scrum correctement et suivre l'avancement de projet en temps réel.
Product Owner	Mr Hamdi Hassen	Définir les besoins fonctionnels et non fonctionnels de projet et fournir la vision de produit fini.
L'équipe de développement	Fedi Balloumi	Ayant le rôle de réaliser et livrer chaque sprint afin d'assurer la finalisation de projet.

TABLE 2.1 – Equipe Scrum

2.2.6 Backlog produit

Le **Product Backlog** est une liste évolutive et priorisée qui regroupe l'ensemble des éléments nécessaires à la réalisation du produit. Il représente la référence principale pour tout changement et oriente l'équipe de développement dans ses choix et ses priorités. Sa gestion est assurée par le **Product Owner**, qui est responsable de son contenu, de sa mise à jour et de son organisation, en fonction de la vision du produit et des besoins des utilisateurs.

Le Product Backlog est généralement structuré sous forme de liste ordonnée contenant, pour chaque élément, les informations suivantes :

- ▷ **ID** : identifiant unique de l'histoire utilisateur.
- ▷ **User Story** : description concise de la fonctionnalité attendue, exprimée du point de vue de l'utilisateur.
- ▷ **Priorité** : importance métier de l'histoire utilisateur, permettant de la classer selon les besoins du client.
- ▷ **Complexité** : estimation de l'effort et du temps nécessaires pour la réalisation de l'histoire utilisateur.

ID	User Story	Priorité	Complexité
1	En tant qu'utilisateur, je veux m'authentifier afin d'accéder de manière sécurisée à mon espace personnel.	1	Élevé
2	En tant que visiteur, je veux consulter la page d'accueil afin de visualiser un aperçu général du système et accéder rapidement aux fonctionnalités.	1	Moyen

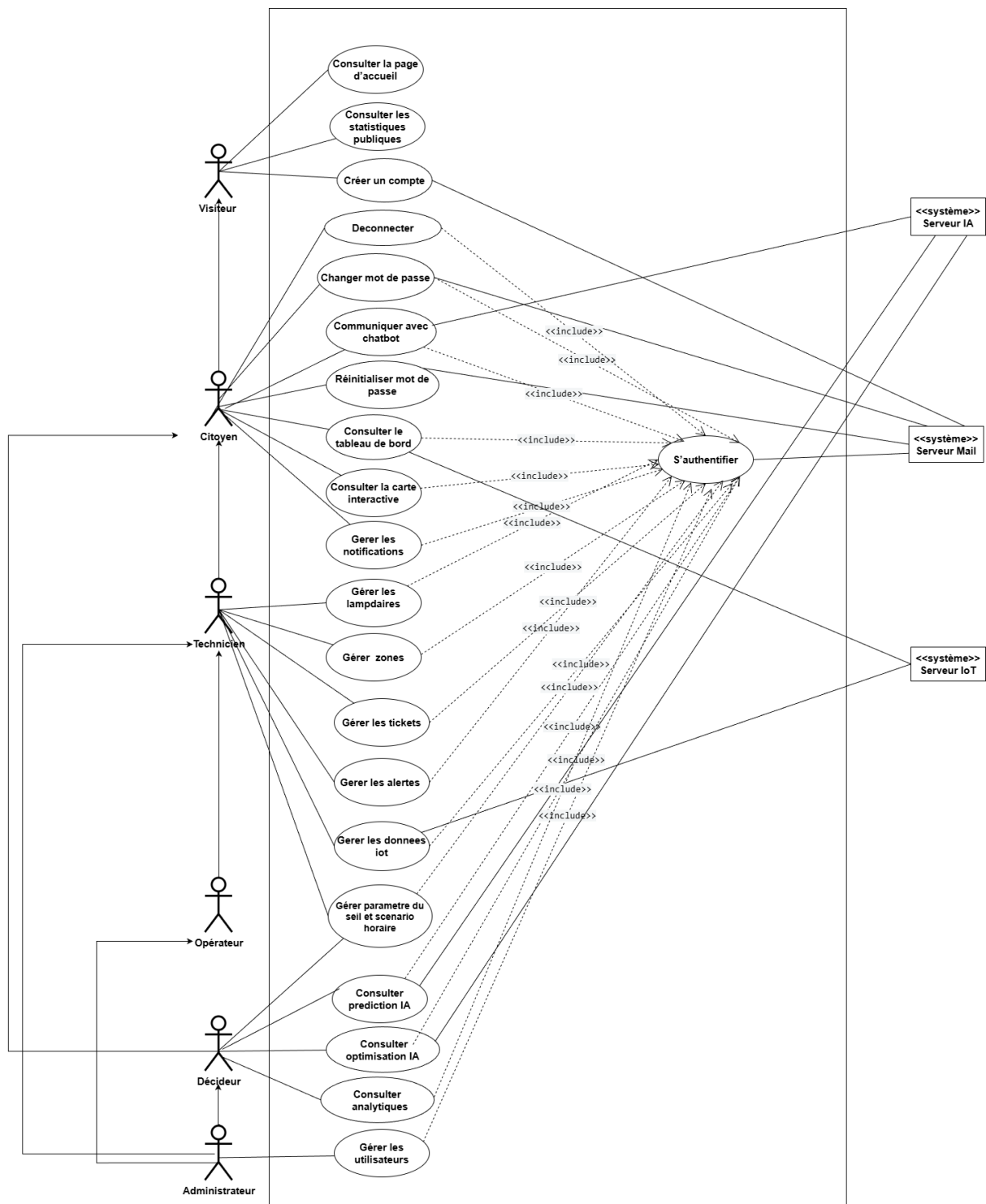
ID	User Story	Priorité	Complexité
3	En tant que visiteur, je veux consulter les statistiques publiques du réseau d'éclairage sans nécessiter d'authentification.	2	Moyen
4	En tant qu'utilisateur, je veux créer un compte afin d'obtenir un accès sécurisé à la plateforme.	1	Élevé
5	En tant qu'utilisateur, je veux réinitialiser mon mot de passe via une procédure de code de confirmation.	2	Moyen
6	En tant qu'utilisateur, je veux me déconnecter afin de mettre fin à ma session de manière sécurisée.	1	Faible
7	En tant qu'administrateur, je veux gérer les utilisateurs afin de contrôler les accès et les rôles au système.	1	Élevé
8	En tant qu'utilisateur, je veux gérer les zones afin de créer, consulter, modifier et supprimer les zones géographiques d'éclairage.	1	Élevé
9	En tant qu'utilisateur, je veux gérer les lampadaires afin d'ajouter, modifier, consulter et supprimer les lampadaires du réseau.	1	Élevé
10	En tant qu'utilisateur, je veux consulter la carte interactive afin de visualiser la localisation géographique des lampadaires.	1	Élevé
11	En tant qu'utilisateur, je veux consulter les alertes afin de consulter les anomalies détectées sur le réseau.	1	Élevé
12	En tant qu'utilisateur, je veux gérer les tickets de maintenance afin de suivre les interventions depuis l'ouverture jusqu'à la résolution.	1	Élevé
13	En tant qu'utilisateur, je veux gérer les données IoT afin de collecter et traiter en temps réel les données des capteurs des lampadaires.	1	Élevé

ID	User Story	Priorité	Complexité
14	En tant qu'utilisateur, je veux gérer les notifications afin de consulter et supprimer les notifications reçues.	2	Moyen
15	En tant qu'utilisateur, je veux consulter le tableau de bord afin de visualiser en temps réel les données du système sous forme de graphiques et statistiques.	1	Élevé
16	En tant qu'utilisateur, je veux consulter les analytiques afin d'accéder à des analyses détaillées des données du système.	1	Élevé
17	En tant qu'utilisateur, je veux consulter la prédiction IA afin de visualiser les prévisions de consommation générées par les modèles d'IA.	1	Élevé
18	En tant qu'utilisateur, je veux consulter l'optimisation IA afin de visualiser les recommandations d'optimisation produites par l'IA.	1	Élevé
19	En tant qu'utilisateur, je veux communiquer avec le chatbot afin d'obtenir des réponses et des analyses basées sur les modèles d'IA.	2	Élevé
20	En tant qu'administrateur, je veux gérer les paramètres du système afin de configurer les seuils de consommation et les règles du réseau.	2	Moyen
21	En tant qu'utilisateur je veux changer mot de passe.	2	Moyen

TABLE 2.2 – Backlog produit

2.2.7 Diagramme de cas d'utilisation globale

Le diagramme de cas d'utilisation globale illustre les interactions principales entre les différents acteurs du système et la plateforme. Comme illustré dans la figure 2.1.



2.2.8 Planification des sprints

La planification des sprints débute une fois le Product Backlog finalisé. Cette étape consiste à sélectionner les fonctionnalités à développer pour chaque sprint et à estimer la durée nécessaire à leur réalisation. Le projet est ainsi découpé en quatre sprints, chacun ayant une charge de travail définie dès le départ.

- **Sprint 1** : Ce sprint couvre la mise en place du socle fonctionnel de la plateforme, notamment l'authentification sécurisée (inscription, connexion, vérification email, double authentification 2FA), la réinitialisation du mot de passe et la gestion des comptes utilisateurs et changer mot de passe.
- **Sprint 2** : Ce sprint est consacré à la gestion des infrastructures d'éclairage public, incluant la gestion des zones géographiques (CRUD), la gestion des lampadaires (CRUD) et la visualisation sur carte interactive.
- **Sprint 3** : Ce sprint introduit les modules de supervision opérationnelle, notamment la gestion des alertes, la gestion des tickets de maintenance, la consultation des données IoT en temps réel et le système de notifications.
- **Sprint 4** : Ce sprint constitue l'aboutissement du développement de la plateforme en intégrant l'intelligence artificielle et la business intelligence, à travers la prédiction de la consommation énergétique, l'optimisation intelligente des lampadaires, le tableau de bord avancé, les tableaux analytiques et le chatbot IA.

2.3 Environnement du travail

Dans cette étape, nous présentons les outils qui ont joué un rôle essentiel dans notre application.

2.3.1 Environnement matériel

Notre application a été développée par :

Caractéristique	Machine
Marque	Asus
Système d'exploitation	Windows 11
Processeur	AMD Ryzen 7
Mémoire RAM	32 Go

TABLE 2.3 – Environnement matériel

Environnement logiciel

Visual Studio Code (VS Code) : est un éditeur de code open source développé par Microsoft. Disponible gratuitement sur Windows, MacOS et Linux, il offre aux développeurs un environnement de développement intégré (IDE) complet pour mener à bien leurs projets techniques.

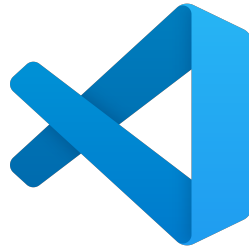


FIGURE 2.2 – Logo Visual Studio Code [11]

Node.js : Basé sur le moteur JavaScript V8 de Chrome, est une plateforme serveur qui autorise les développeurs à concevoir des applications côté serveur, web rapides et évolutives en utilisant JavaScript, tant pour le côté serveur que pour le côté client.



FIGURE 2.3 – Logo Node.js [3]

Express : Framework minimaliste et flexible pour Node.js, utilisé pour structurer le backend, gérer les requêtes et les réponses HTTP.



FIGURE 2.4 – Logo Express.js [4]

MongoDB : Est un système de gestion de base de données NoSQL orienté documents, conçu pour stocker de grands volumes de données de manière flexible et performante.



FIGURE 2.5 – Logo MongoDB [5]

InfluxDB : Est un système de gestion de base de données open source optimisé pour les données de séries temporelles (Time Series Database - TSDB).



FIGURE 2.6 – Logo InfluxDB [6]

React JS : Est une bibliothèque JavaScript open source conçue par Meta pour créer des interfaces utilisateur interactives et performantes, principalement pour les applications web monopages.



FIGURE 2.7 – Logo React [7]

TypeScript : Est un langage de programmation open source développé par Microsoft qui s'exécute comme un sur-ensemble typé de JavaScript.



FIGURE 2.8 – Logo TypeScript [8]

Python : Est un langage de programmation interprété, puissant et réputé facile à apprendre grâce à sa syntaxe élégante et proche de l'anglais. Idéal pour débiter, il est largement utilisé pour le développement web, l'automatisation, la science des données et l'intelligence artificielle.



FIGURE 2.9 – Logo Python [9]

FastAPI : Est un framework web moderne, open-source et hautement performant conçu pour créer des API RESTful avec Python 3.7+. Il repose sur deux piliers majeurs : Starlette pour la gestion des serveurs web asynchrones et Pydantic pour la validation automatique des données.



FIGURE 2.10 – Logo FastAPI [10]

Draw.io : Est un outil de conception de diagrammes en ligne, gratuit et open source, permettant de créer des diagrammes UML, des schémas d'architecture, des organigrammes et des maquettes de manière intuitive et collaborative.



FIGURE 2.11 – Logo Draw.io [15]

Postman : Est la plateforme leader mondiale pour concevoir, tester et gérer des API. Elle simplifie le cycle de vie du développement logiciel en remplaçant l'écriture de code de test par une interface graphique intuitive.



FIGURE 2.12 – Logo Postman [12]

Git : Est un système de contrôle de version distribué, incontournable pour le développement logiciel, permettant de gérer l'historique des modifications de code de manière rapide et sécurisée.



FIGURE 2.13 – Logo Git [13]

GitHub : Est une plateforme web d'hébergement et de gestion de code source pour les développeurs. Elle repose sur le logiciel de gestion de versions Git et sert d'outil collaboratif à l'échelle mondiale.



FIGURE 2.14 – Logo GitHub [14]

2.4 Architecture générale de l'application

L'architecture de notre application repose sur le modèle 3-tiers, une approche largement adoptée dans le développement des applications web modernes. Ce modèle structure le système en trois couches distinctes et indépendantes, favorisant ainsi la maintenabilité, la modularité et l'évolutivité de la plateforme.

La première couche, dite de **présentation**, est assurée par un frontend développé avec React et TypeScript. Elle constitue le point de contact direct avec l'utilisateur et offre une interface interactive permettant la visualisation des tableaux de bord, la gestion des zones et des lampadaires, le suivi des alertes et des tickets, ainsi que l'interaction avec l'assistant intelligent.

La deuxième couche, dite de **logique métier**, repose sur un backend implémenté avec Node.js et Express. Organisée selon une architecture modulaire distinguant les controllers, les services, les routes et les models, elle centralise le traitement des requêtes, l'application des règles métier, la gestion de l'authentification et la communication avec les services externes.

La troisième couche, dite de **données**, s'appuie sur MongoDB, une base de données NoSQL orientée documents, assurant le stockage persistant des informations relatives aux utilisateurs, aux zones, aux lampadaires, aux notifications et aux historiques d'analyse.

En complément de cette architecture 3-tiers, la plateforme intègre un **microservice d'intelligence artificielle** développé avec FastAPI. Ce service indépendant est dédié à la prédiction de la consommation énergétique, à l'optimisation intelligente de

l'éclairage et à l'analyse des données issues du système IoT. Il communique avec le backend via une API REST, garantissant ainsi un couplage faible et une grande flexibilité d'évolution. La figure suivante illustre cette architecture en mettant en évidence les interactions entre les différentes couches ainsi que l'intégration du service IA.

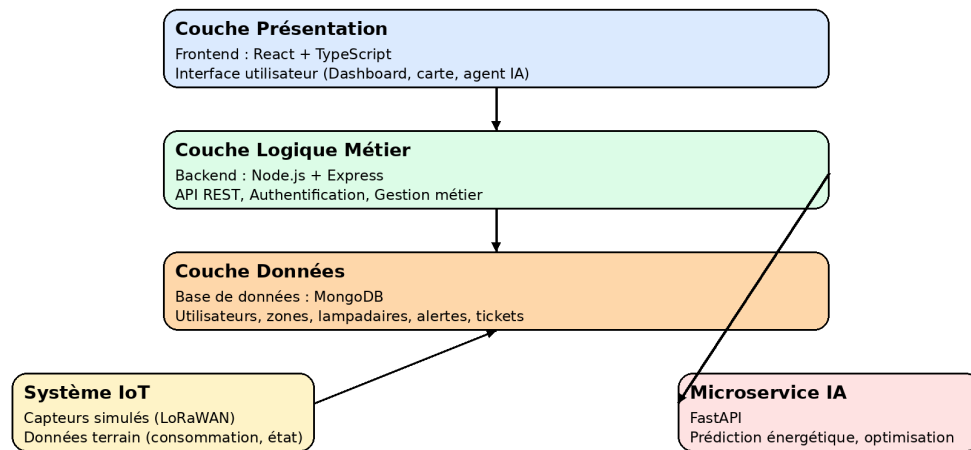


FIGURE 2.15 – Architecture physique du système

2.5 Diagramme de déploiement

Le diagramme de déploiement en UML est une représentation statique décrivant l'architecture physique d'un système. Il met en évidence la répartition des composants logiciels sur les différents nœuds matériels et les communications établies entre ces derniers.

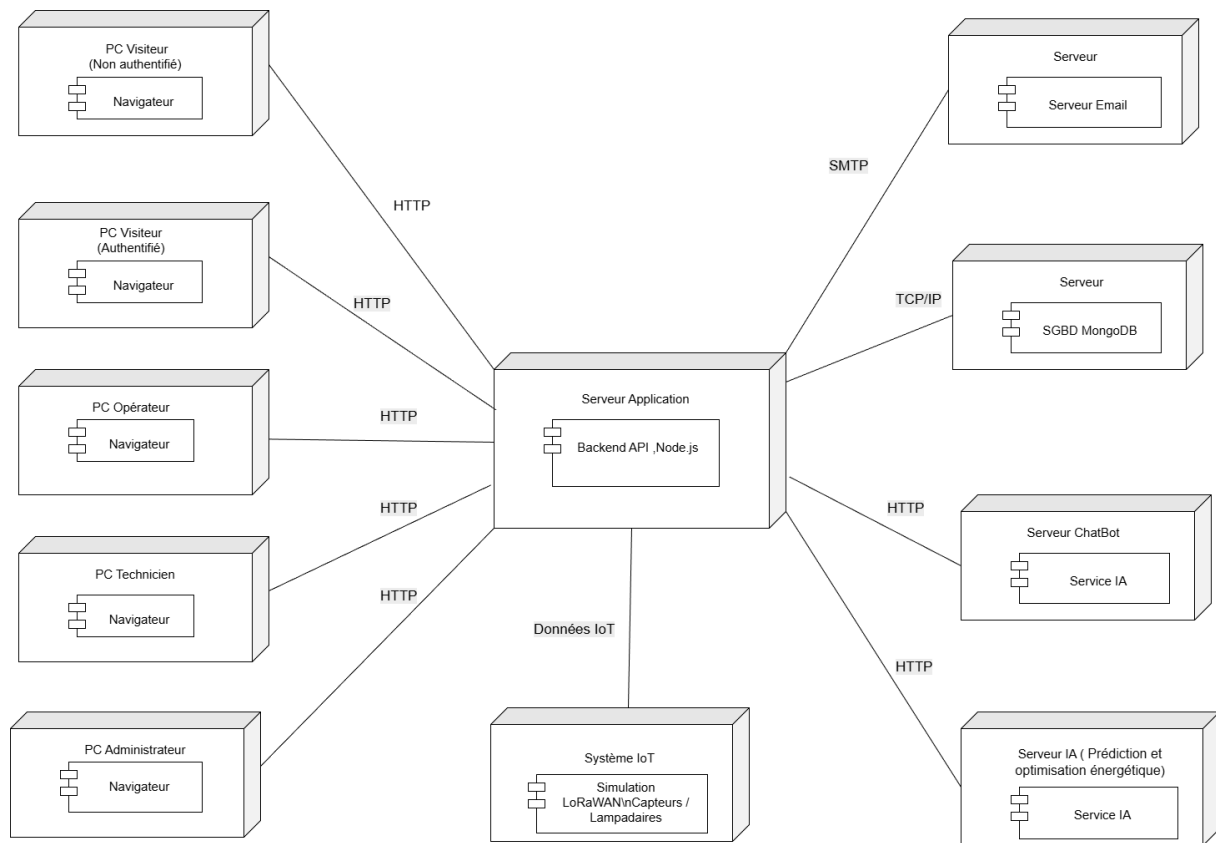


FIGURE 2.16 – Diagramme de déploiement

2.6 Conclusion

Dans ce chapitre, nous avons présenté le Sprint 0 consacré à l'analyse des besoins et à l'étude technique. Cette étape a permis d'identifier les acteurs, de préciser les besoins, de structurer le backlog produit et de définir les bases techniques nécessaires au développement du projet.

Chapitre 3

Sprint 1 : Étude,analyse et réalisation du Sprint 1

Plan

1. Introduction
2. Backlog du Sprint 1
3. Spécification de besoins du sprint 1
4. Conception
5. Réalisation et tests
6. Utilisation des outils de suivi Scrum
7. Conclusion

3.1 Introduction

Dans ce chapitre, nous présentons les travaux réalisés lors du Sprint 1. Ce sprint constitue une étape fondamentale dans le développement de notre plateforme Smart Lighting, puisqu'il vise à mettre en place le socle technique du système. Il inclut la configuration de l'environnement de développement, la mise en place de l'architecture backend ainsi que l'implémentation des fonctionnalités de base nécessaires au bon fonctionnement de la plateforme.

3.2 Backlog du Sprint 1

Le Sprint 2 est consacré à la gestion des zones et des lampadaires, ainsi qu'à leur visualisation sur une carte interactive. Les fonctionnalités définies dans ce sprint permettent aux différents acteurs du système, à savoir l'administrateur, l'opérateur et le technicien, d'effectuer les opérations de création, de consultation, de modification et de suppression sur les zones et les lampadaires. Ce sprint intègre également une interface cartographique facilitant la localisation et la consultation des équipements.

Le tableau ci-dessous présente les tâches planifiées et réalisées durant le Sprint 1, en lien avec les User Stories définies. Ce sprint est principalement consacré à l'authentification et à la gestion des entités de base du système.

ID US	User Story	ID tâche	Tâche	Statut	Complexité
1	En tant que visiteur, je veux consulter la page d'accueil.	1.1	Conception de la page d'accueil	Terminé	Faible
		1.2	Développement interface page d'accueil	Terminé	Moyen
		1.3	Intégration des données (KPI, carte)	Terminé	Moyen
		1.4	Test de la page d'accueil	Terminé	Faible
2	En tant que visiteur, je veux m'inscrire.	2.1	Conception formulaire d'inscription	Terminé	Faible
		2.2	Développement interface inscription	Terminé	Moyen
		2.3	Développement API signup	Terminé	Élevé
		2.4	Envoi email de vérification	Terminé	Élevé
		2.5	Validation email (code)	Terminé	Moyen
		2.6	Test inscription	Terminé	Faible

ID US	User Story	ID tâche	Tâche	Statut	Complexité
3	En tant qu'utilisateur, je veux m'authentifier.	3.1	Conception interface login	Terminé	Faible
		3.2	Développement API login	Terminé	Élevé
		3.3	Implémentation JWT	Terminé	Élevé
		3.4	Génération code 2FA	Terminé	Élevé
		3.5	Validation code 2FA	Terminé	Moyen
		3.6	Test authentification	Terminé	Faible
4	En tant qu'utilisateur, je veux réinitialiser mon mot de passe.	4.1	Conception interface reset	Terminé	Faible
		4.2	Développement processus reset	Terminé	Élevé
		4.3	Envoi email de réinitialisation	Terminé	Moyen
		4.4	Validation code reset	Terminé	Moyen
		4.5	Modification mot de passe	Terminé	Moyen
		4.6	Test reset password	Terminé	Faible
5	En tant qu'administrateur, je veux gérer les comptes utilisateurs.	5.1	Affichage liste utilisateurs	Terminé	Moyen
		5.2	Activer / Désactiver compte	Terminé	Moyen
		5.3	Gestion des rôles	Terminé	Élevé
		5.4	Consulter date de création	Terminé	Faible

ID US	User Story	ID tâche	Tâche	Statut	Complexité
		5.5	Consulter dernière connexion	Terminé	Faible

TABLE 3.2 – Backlog détaillé du Sprint 1

3.3 Spécification des besoins du Sprint 1

La rédaction des spécifications fonctionnelles constitue une étape essentielle dans tout projet informatique. Elles sont regroupées dans une section spécifique du cahier des charges et permettent de définir, décrire et clarifier les fonctionnalités du site, de l'application ou du logiciel concerné. À ce stade, il est important d'élaborer un diagramme de cas d'utilisation ainsi que d'identifier les différents scénarios associés, afin de vérifier que les fonctionnalités à développer correspondent bien à celles définies dans le backlog.

3.3.1 Diagramme de cas d'utilisation

Dans le but de mieux illustrer les fonctionnalités développées lors de ce sprint, le diagramme de cas d'utilisation est présenté dans la figure ci-après.

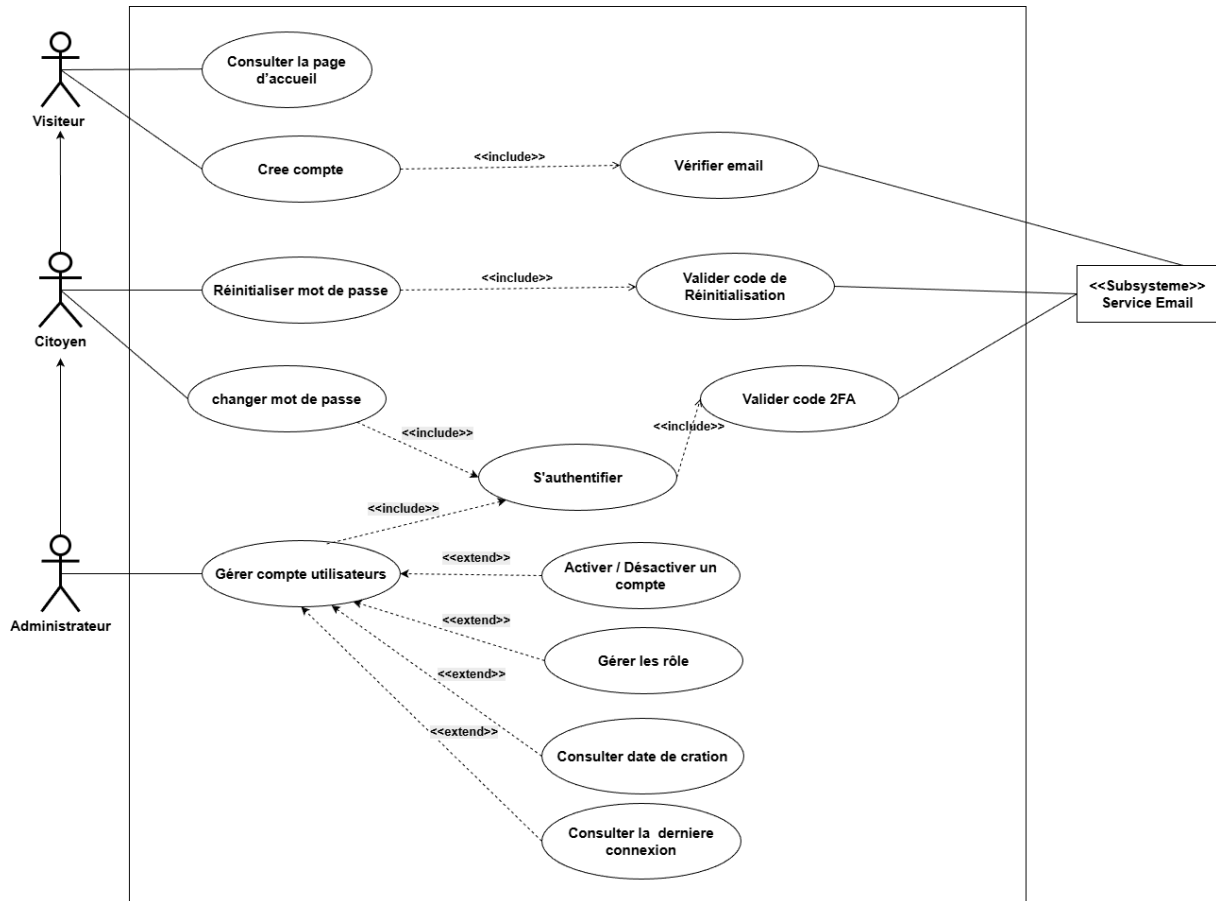


FIGURE 3.1 – Diagramme de cas d'utilisation du Sprint 1

3.3.2 Description textuelle des cas d'utilisations

Cette section a pour objectif de fournir une description textuelle détaillée des principaux cas d'utilisation du système.

Description de cas d'utilisation « S'inscrire »

Le tableau suivant présente la description du cas d'utilisation « S'inscrire ».

Cas d'utilisation	Cree compte
Acteurs	Visiteur
Description brève du CU	Permet au visiteur de créer un compte afin d'accéder aux fonctionnalités de la plateforme.
Préconditions	Le visiteur ne possède pas encore de compte.
Post conditions	Un compte utilisateur est créé et enregistré dans la base de données.
Scénario de base	1. Visiteur accède à la page d'accueil 2. Visiteur clique button "inscription" 3. Visiteur accède à la page d'inscription. 4. Le système affiche le formulaire d'inscription. 5. Visiteur saisit ses informations (nom, email, mot de passe). 6. Visiteur clique button "Cree un compte" 7. Le système vérifie les données saisies. 8. Le système affiche l'interface "vérification de l'email" 9. Le système envoie un email de vérification. 10. Visiteur saisit code de vérification et clique bouton "vérifier mon email" 11. Le système affiche l'interface "Authentification a deux facteurs" 12. Le système envoie un email de vérification. 13. Visiteur saisit code de vérification et clique bouton "Valider le code" 14. Système affiche l'espace de l'utilisateur
Autre scénario	7.1. Si les champs sont invalides → affichage d'un message d'erreur. 13.1. Si le code invalide → visiteur clique bouton 'Renvoyer le code' → serveur mail renvoyer le code .

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

Description de cas d'utilisation « S'authentifier »

Le tableau suivant présente la description du cas d'utilisation « S'authentifier».

Cas d'utilisation	S'authentifier
Acteurs	Utilisateur,Administrateur
Description brève du CU	Permet à L'acteur d'accéder à son espace personnel via une authentification sécurisée.
Préconditions	L'acteur possède un compte valide.
Post conditions	L'acteur est authentifié et accède au système.
Scénario de base	1.L'acteur accède à la page d'accueil. 2.L'acteur clique bouton "Se connecter". 3.Le système affiche l'interface connexion 4. L'acteur accède à la page de connexion. 5. Le système affiche le formulaire de connexion. 6. L'acteur saisit son email et son mot de passe. 7.L'acteur clique bouton "Se connecter" 8. Le système vérifie les identifiants. 9. Le système génère un code 2FA. 10. Le système envoie le code par email. 11. L'acteur saisit le code 2FA. 12.L'acteur clique bouton "" 13. Le système valide le code et affiche l'espace de l'utilisateur. 14. L'acteur est connecté.
Autre scénario	8.1. Identifiants incorrects →Le système affiche un message d'erreur. 13. Code 2FA invalide → accès refusé.

TABLE 3.4 – Description du cas d'utilisation « S'authentifier »

3.4 Conception

Cette section présente les diagrammes de séquence et de classes élaborés pour modéliser le comportement dynamique et la structure statique du système au cours du Sprint .

3.4.1 Diagrammes de séquences sprint 1

Cette section expose les différents diagrammes de séquence réalisés dans le cadre du Sprint 1.

3.4.2 Diagramme de séquence du scénario « Se connecter »

Le diagramme de séquence suivant présente le déroulement du processus d'authentification d'un utilisateur. Après la vérification des identifiants, un code de double authentification est généré et transmis par email.

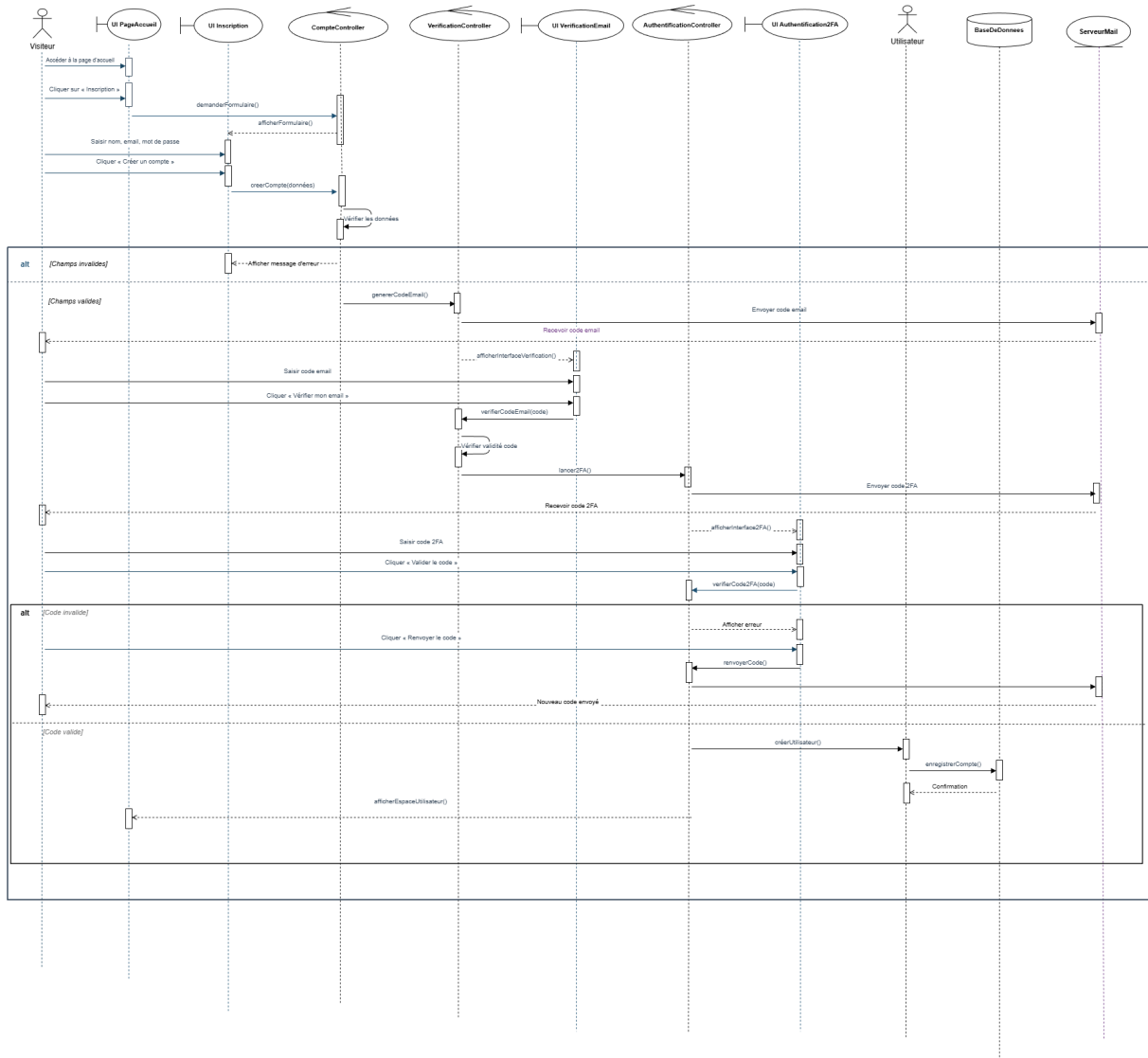


FIGURE 3.2 – Diagramme de séquence du scénario « Se connecter »

3.4.3 Diagramme de séquence du scénario « S’authentifier »

Le diagramme de séquence suivant illustre le processus d’authentification d’un utilisateur au sein de la plateforme. Après la saisie de l’email et du mot de passe, le système vérifie d’abord l’existence du compte ainsi que l’état de vérification de l’adresse email. Si les identifiants sont valides, un code de double authentification (2FA) est généré puis envoyé par email. L’accès au système n’est autorisé qu’après validation correcte de ce code. Une fois l’authentification confirmée, le système génère un jeton JWT et redirige l’utilisateur vers son tableau de bord.



3.4.4 Diagramme de Classes du sprint 1

Le diagramme de classes présenté dans la figure met en évidence les principales entités du système ainsi que les relations qui les lient dans le cadre du Sprint 1.

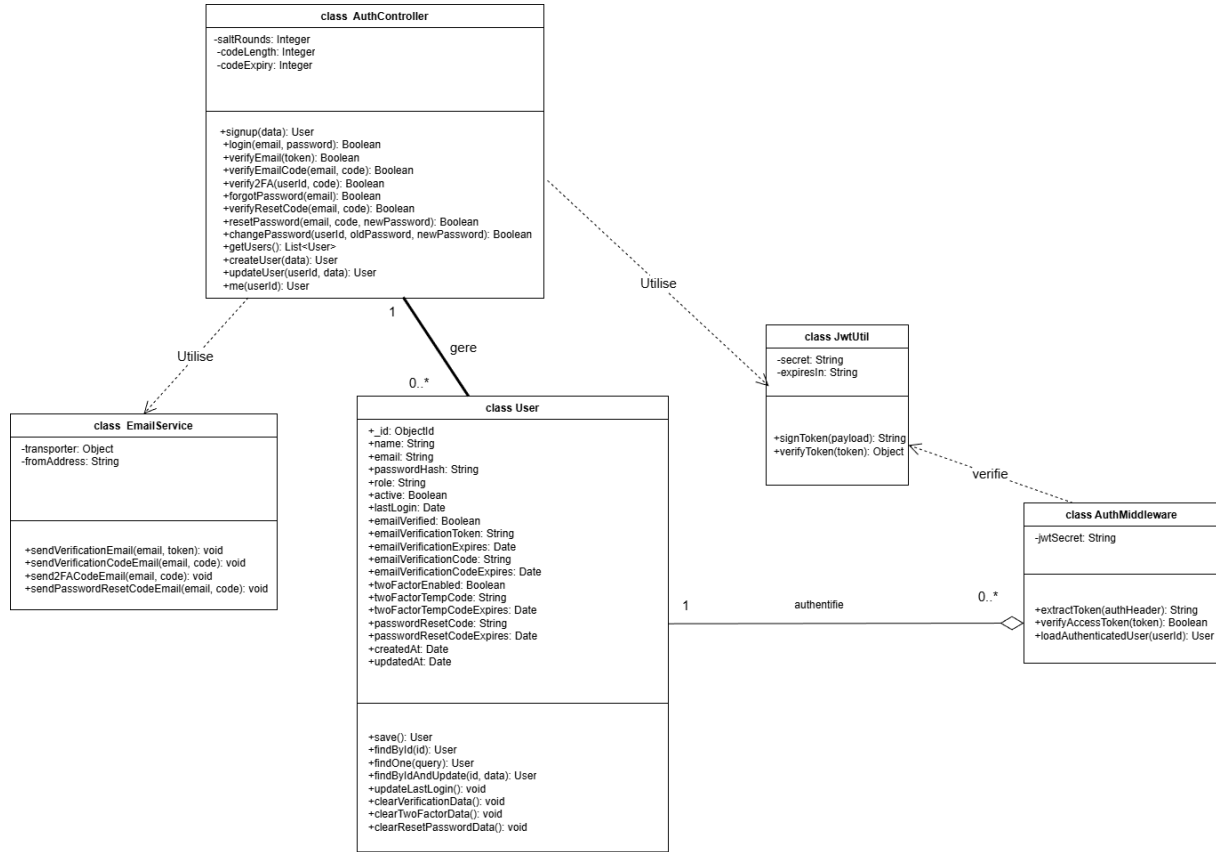


FIGURE 3.4 – Diagramme de Classes du sprint 1

3.5 Réalisation et tests

Cette section décrit le processus de mise en œuvre du système ainsi que les différents tests réalisés afin de valider son bon fonctionnement et sa conformité aux exigences spécifiées.

Cette section présente les différentes interfaces réalisées durant ce sprint.

Dans cette section, nous présentons les interfaces développées au cours de ce sprint.

Interface d'accueil

L'interface d'accueil constitue le point d'entrée principal de la plateforme Smart Lighting. Elle est accessible à tous les utilisateurs, y compris les visiteurs non authentifiés. Cette page a pour objectif de présenter une vue globale du système ainsi que ses principales fonctionnalités.

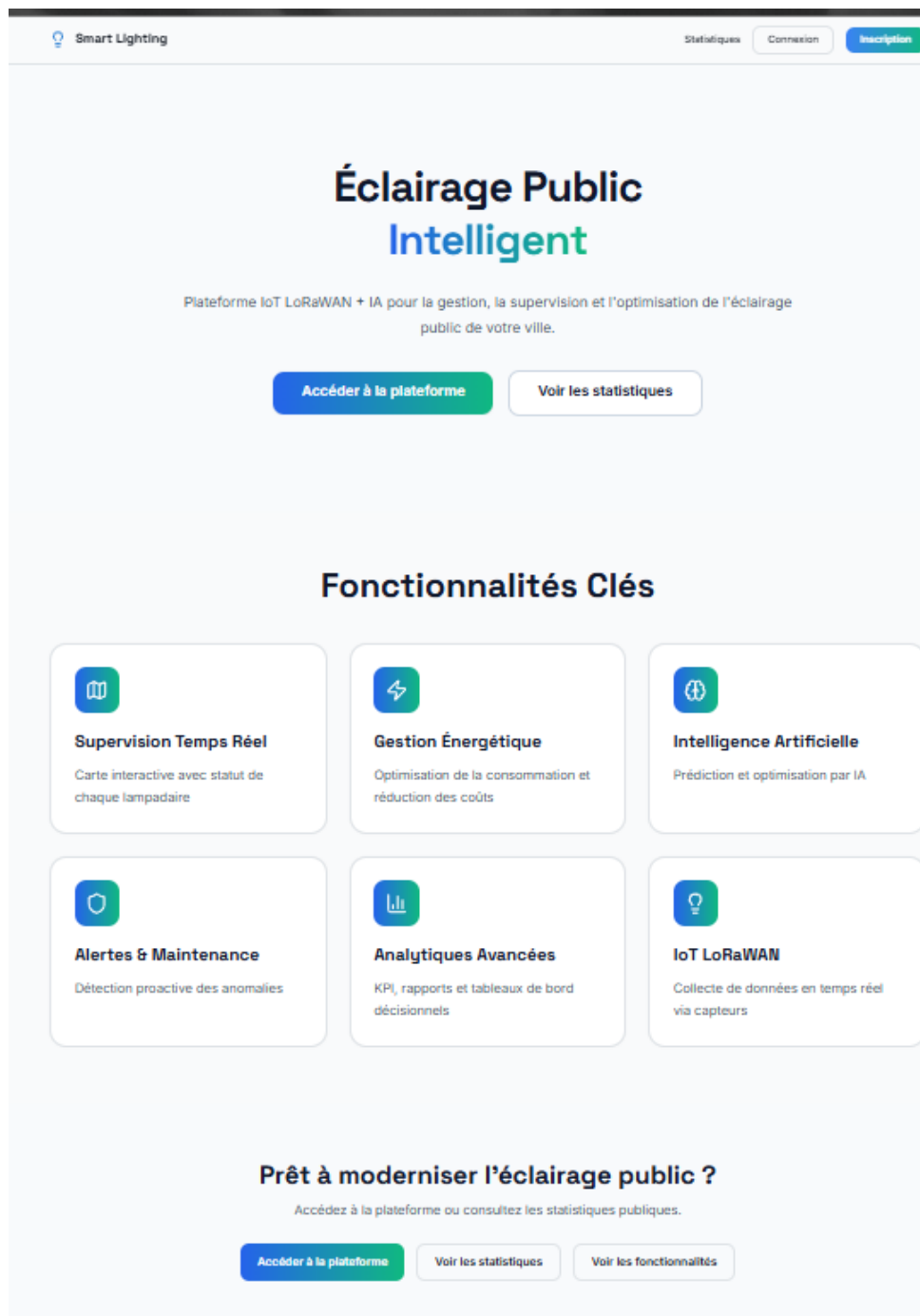


FIGURE 3.5 – Interface d'accueil

Interface Consulter les statistiques publiques

L'interface de consultation des statistiques publiques représente la vitrine analytique de la plateforme. Accessible sans authentification.

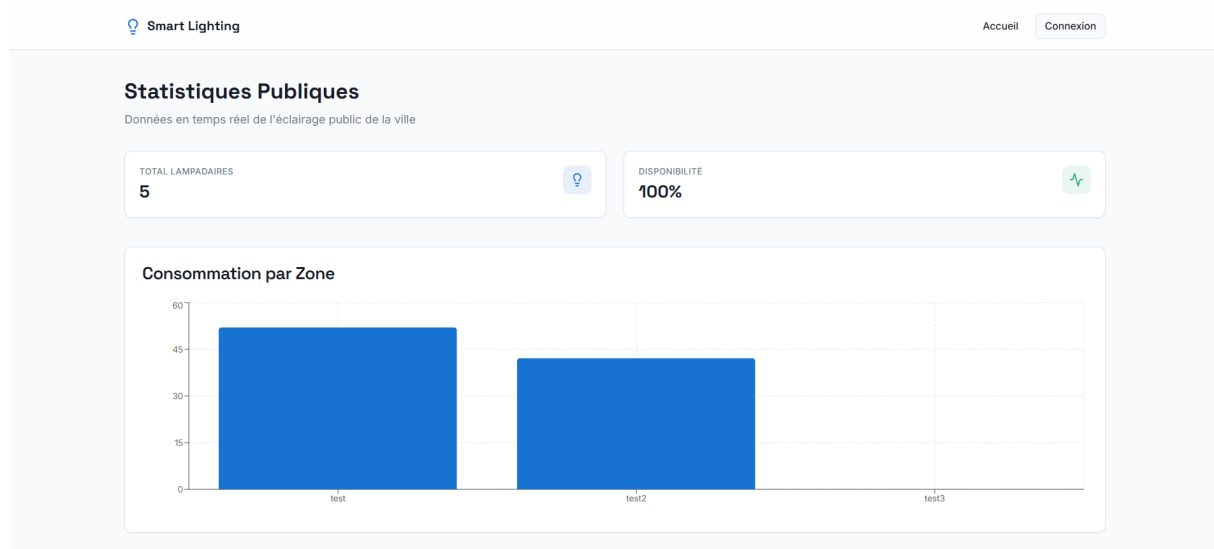
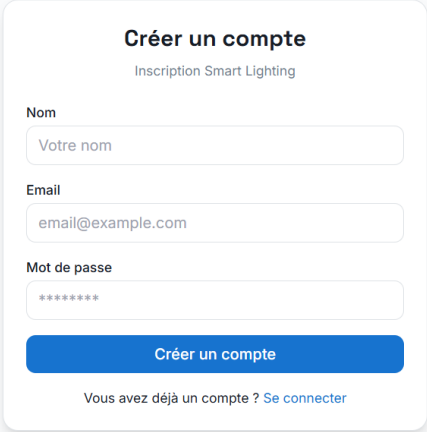


FIGURE 3.6 – Interface Consulter les statistiques publiques

Interface d'inscription

L'interface d'inscription permet aux utilisateurs de créer un nouveau compte afin d'accéder aux fonctionnalités de la plateforme Smart Lighting. Elle est accessible aux visiteurs non authentifiés et constitue une étape essentielle du processus d'authentification.

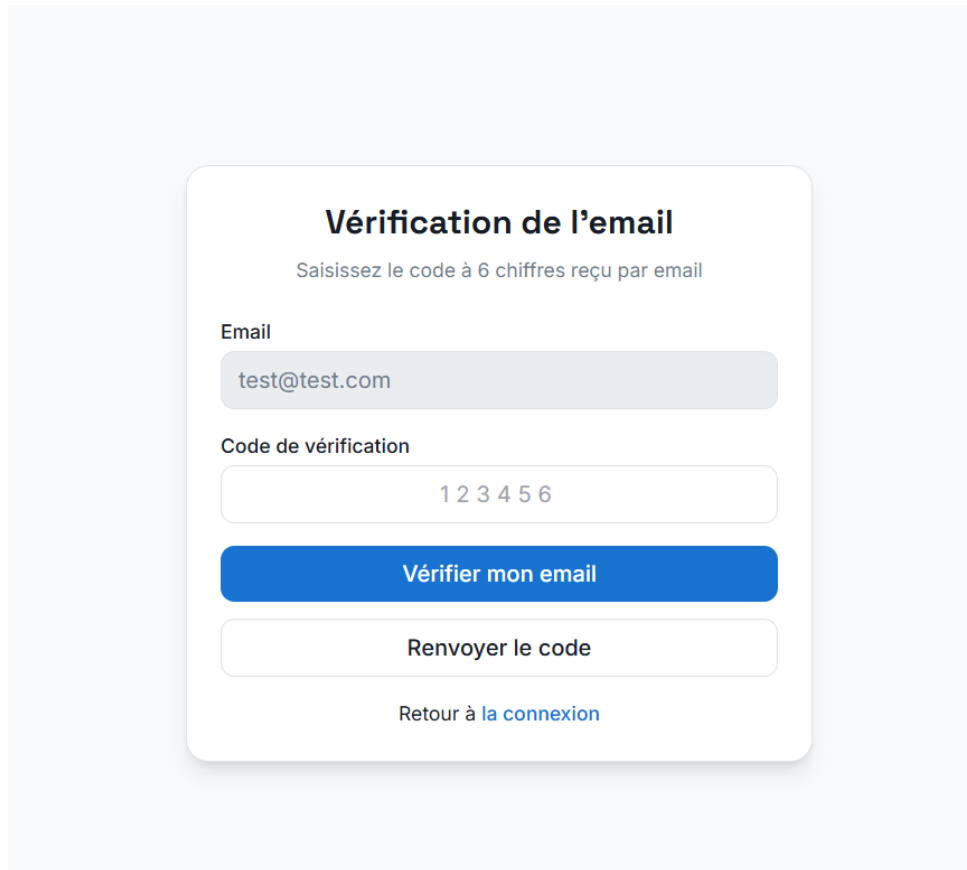


The image shows a registration form titled "Créer un compte" (Create an account) for "Inscription Smart Lighting". The form is centered on a light gray background. It contains three input fields: "Nom" (Name) with the placeholder "Votre nom", "Email" with the placeholder "email@example.com", and "Mot de passe" (Password) with a masked input "*****". Below the fields is a blue button labeled "Créer un compte". At the bottom, there is a link: "Vous avez déjà un compte ? [Se connecter](#)".

FIGURE 3.7 – Interface d'inscription

Interface vérification email

L'interface de vérification de l'email permet de confirmer l'identité de l'utilisateur après la phase d'inscription. Elle constitue une étape essentielle du processus de sécurisation des comptes au sein de la plateforme Smart Lighting.



The image shows a web interface for email verification. It features a white card with rounded corners on a light gray background. The card has the title "Vérification de l'email" in bold black text, followed by the instruction "Saisissez le code à 6 chiffres reçu par email". Below this, there is a label "Email" and a text input field containing "test@test.com". Another label "Code de vérification" is followed by a six-digit input field with the numbers "1 2 3 4 5 6" displayed inside. There are two buttons: a prominent blue button labeled "Vérifier mon email" and a white button with a gray border labeled "Renvoyer le code". At the bottom of the card, there is a link "Retour à la connexion" in blue text.

FIGURE 3.8 – Interface vérification email

Interface de connexion

L'interface de connexion permet aux utilisateurs déjà inscrits d'accéder à leur compte au sein de la plateforme Smart Lighting. Elle constitue une étape essentielle du processus d'authentification.

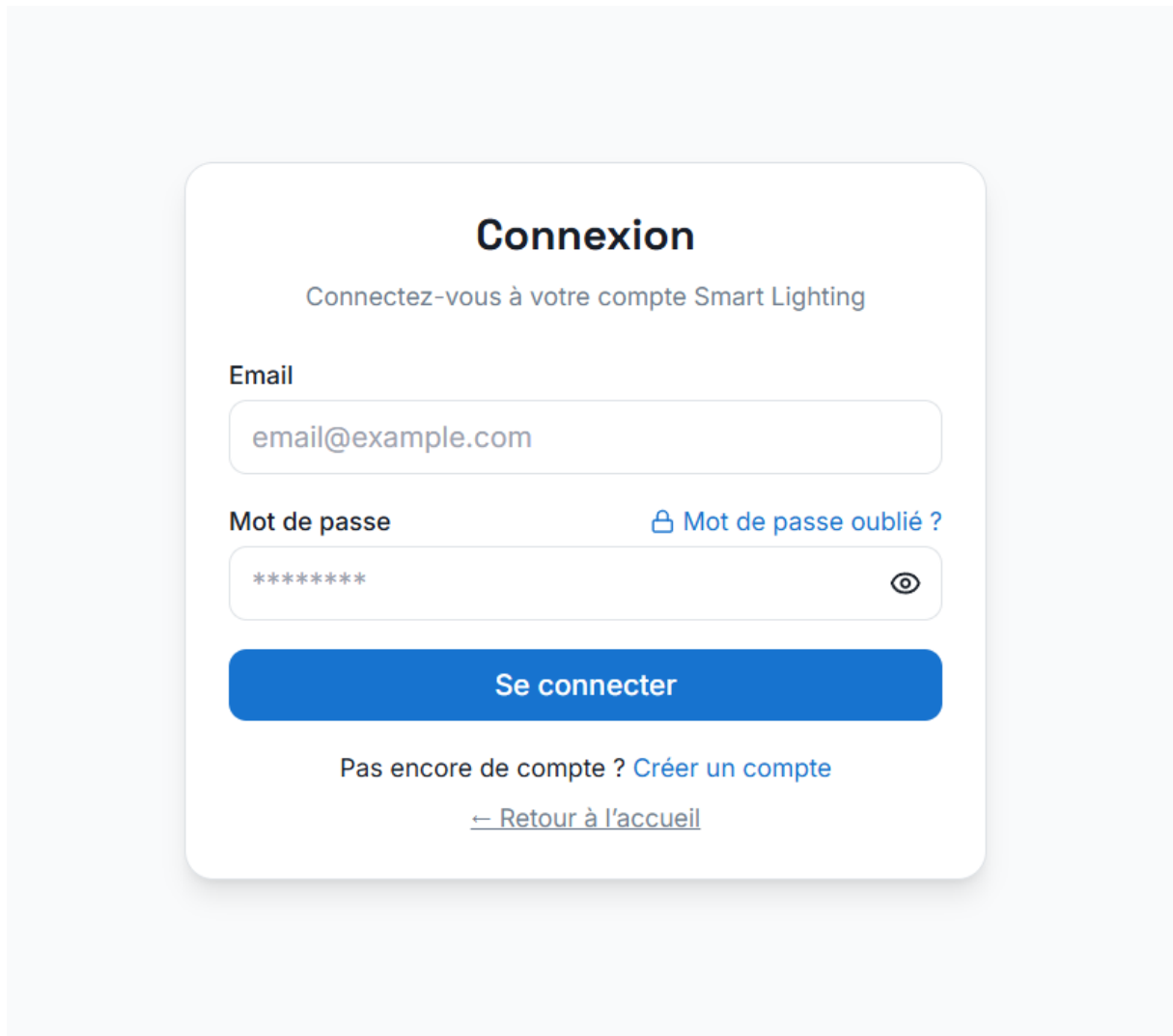
The image shows a login form titled "Connexion" for the "Smart Lighting" platform. The form is centered on a light gray background. It includes fields for "Email" (containing "email@example.com") and "Mot de passe" (containing "*****"). A "Mot de passe oublié ?" link is next to the password field. A blue "Se connecter" button is below the fields. At the bottom, there is a link "Créer un compte" for users who do not have an account, and a "Retour à l'accueil" link with a left arrow.

FIGURE 3.9 – Interface de connexion

Interface d'authentification à deux facteurs

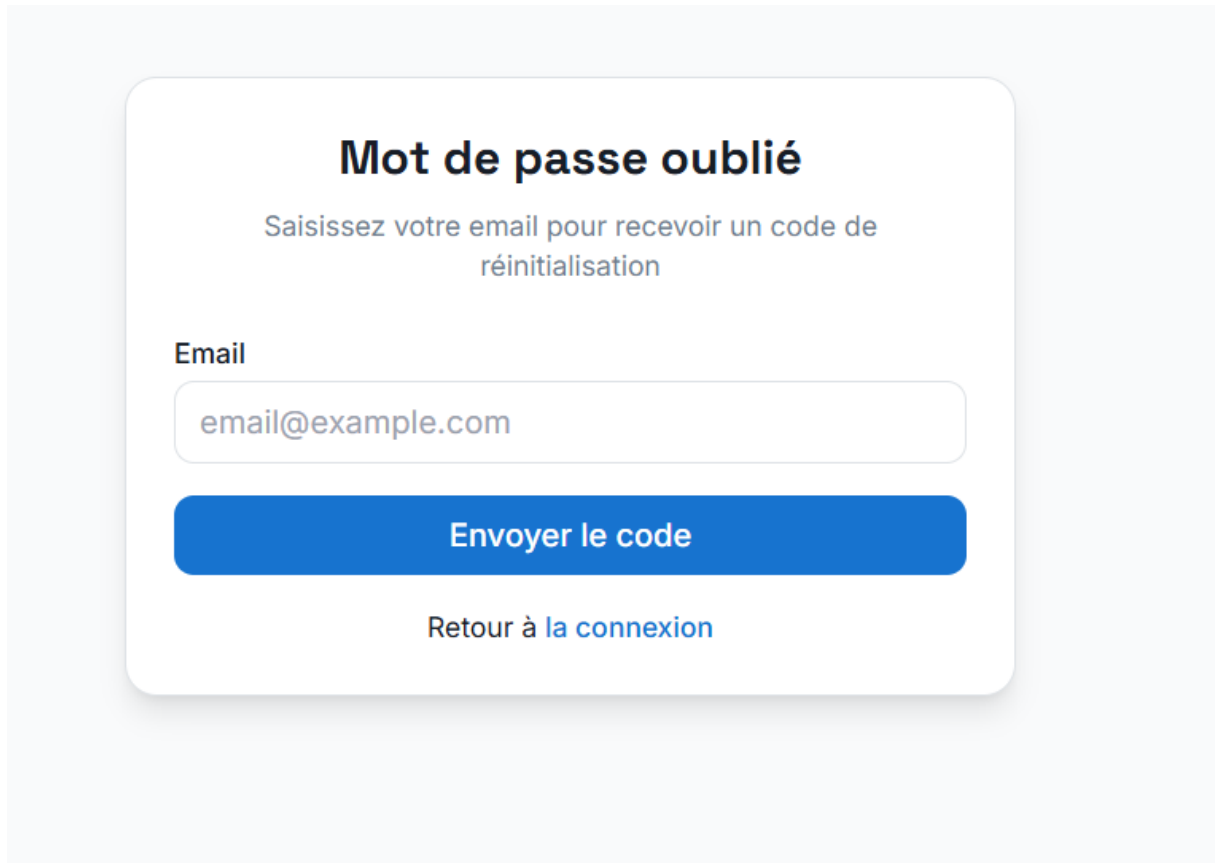
L'interface d'authentification à deux facteurs (2FA) constitue une étape supplémentaire de sécurité dans le processus de connexion. Elle permet de vérifier l'identité de l'utilisateur après la validation de ses identifiants (email et mot de passe).

The image shows a user interface for two-factor authentication (2FA). It features a white rounded rectangle centered on a light gray background. At the top, the title 'Authentification à deux facteurs' is displayed in bold black text. Below the title, a subtitle 'Saisissez le code à 6 chiffres reçu par email' is shown in a smaller, gray font. Underneath, the label 'Code 2FA' is positioned above a text input field. The input field contains six placeholder digits: '1 2 3 4 5 6'. Below the input field, there are two buttons: a prominent blue button labeled 'Valider le code' and a white button with a gray border labeled 'Renvoyer le code'. At the bottom of the interface, there is a link that reads 'Retour à la connexion'.

FIGURE 3.10 – Interface d'authentification à deux facteurs

Interface mot de passe oublié

L'interface « Mot de passe oublié » permet à un utilisateur ayant perdu son mot de passe d'initier le processus de réinitialisation de son compte. Elle constitue la première étape du mécanisme de récupération d'accès.



Mot de passe oublié

Saisissez votre email pour recevoir un code de réinitialisation

Email

email@example.com

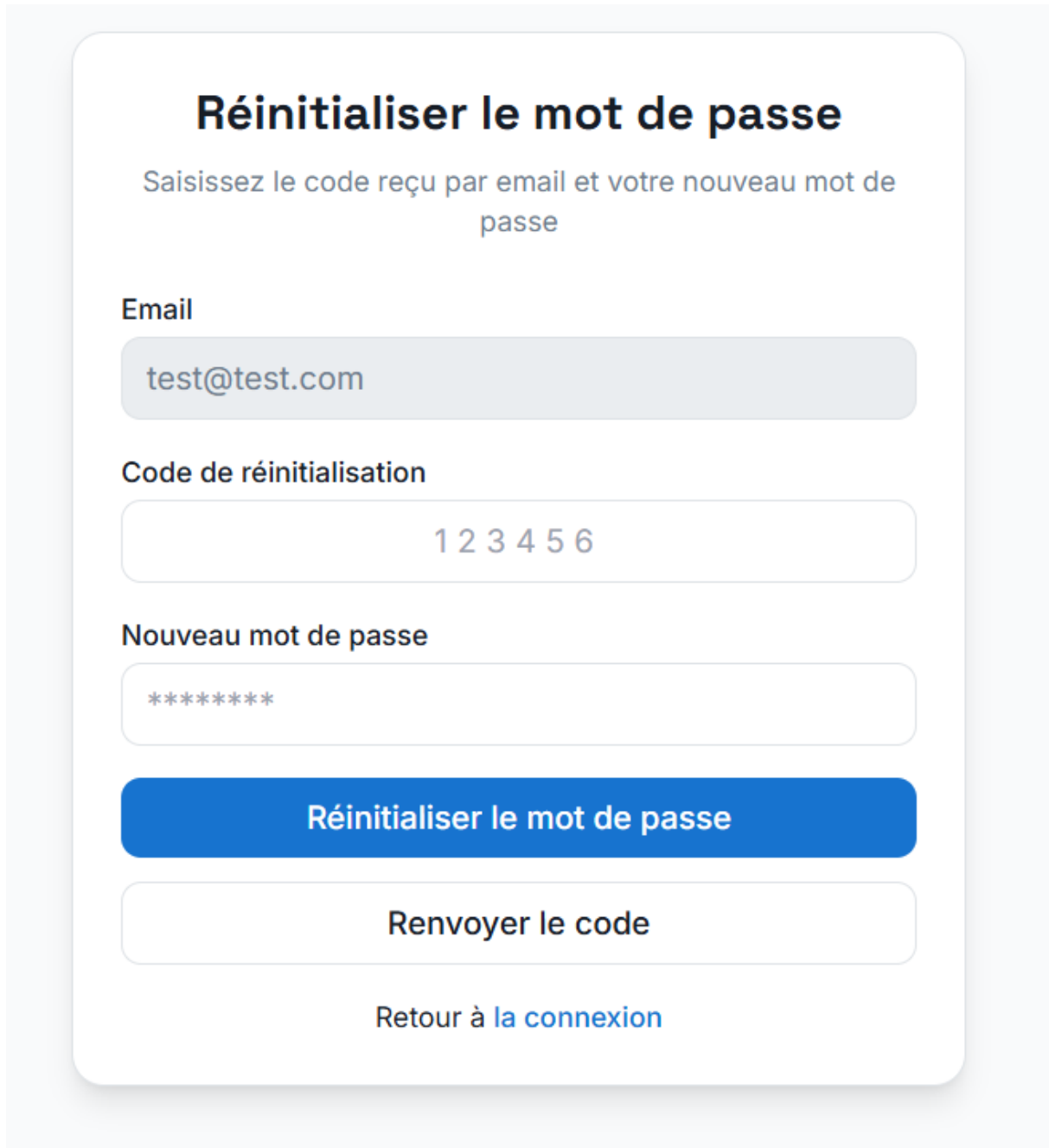
Envoyer le code

[Retour à la connexion](#)

FIGURE 3.11 – Interface mot de passe oublié

Interface réinitialiser le mot de passe

L'interface de réinitialisation du mot de passe permet à l'utilisateur de définir un nouveau mot de passe après validation d'un code de sécurité reçu par email. Elle constitue la deuxième étape du processus de récupération d'accès.



The image shows a web interface for password reset. It features a white card with rounded corners on a light gray background. The card has a title 'Réinitialiser le mot de passe' in bold black text, followed by a subtitle 'Saisissez le code reçu par email et votre nouveau mot de passe' in a smaller, lighter blue font. Below the subtitle are three input fields: 'Email' containing 'test@test.com', 'Code de réinitialisation' containing '1 2 3 4 5 6', and 'Nouveau mot de passe' containing '*****'. At the bottom of the card are two buttons: a prominent blue button labeled 'Réinitialiser le mot de passe' and a white button with a gray border labeled 'Renvoyer le code'. Below the white button is a link 'Retour à la connexion' in blue text.

Réinitialiser le mot de passe

Saisissez le code reçu par email et votre nouveau mot de passe

Email

test@test.com

Code de réinitialisation

1 2 3 4 5 6

Nouveau mot de passe

Réinitialiser le mot de passe

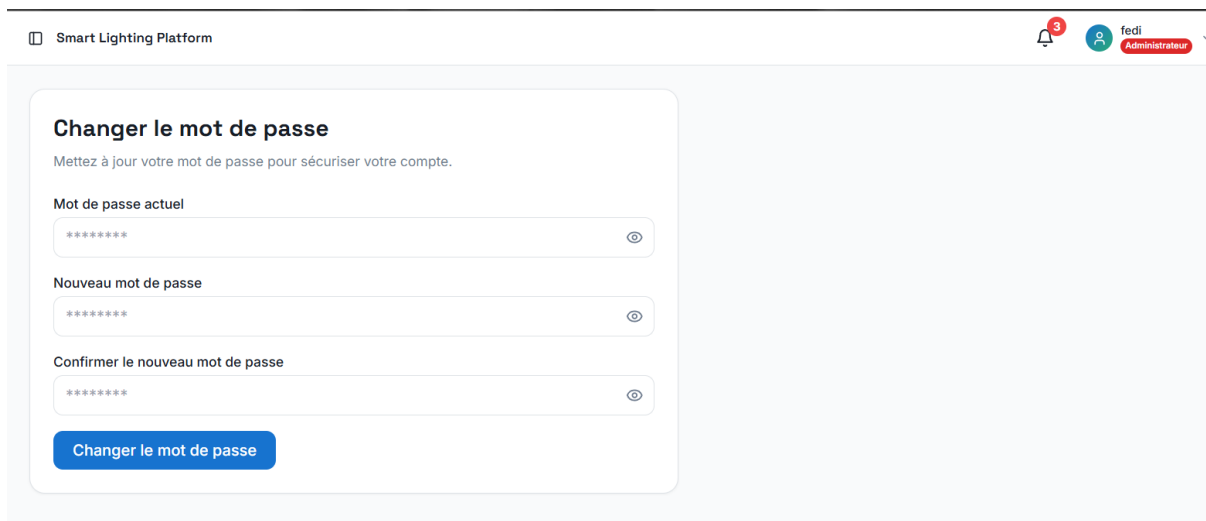
Renvoyer le code

[Retour à la connexion](#)

FIGURE 3.12 – Interface réinitialiser le mot de passe

Interface changer le mot de passe

L'interface « Changer le mot de passe » permet à un utilisateur authentifié de modifier son mot de passe actuel afin de renforcer la sécurité de son compte. Elle est accessible depuis l'espace personnel de l'utilisateur au sein de la plateforme Smart Lighting.



Smart Lighting Platform

Changer le mot de passe

Mettez à jour votre mot de passe pour sécuriser votre compte.

Mot de passe actuel

Nouveau mot de passe

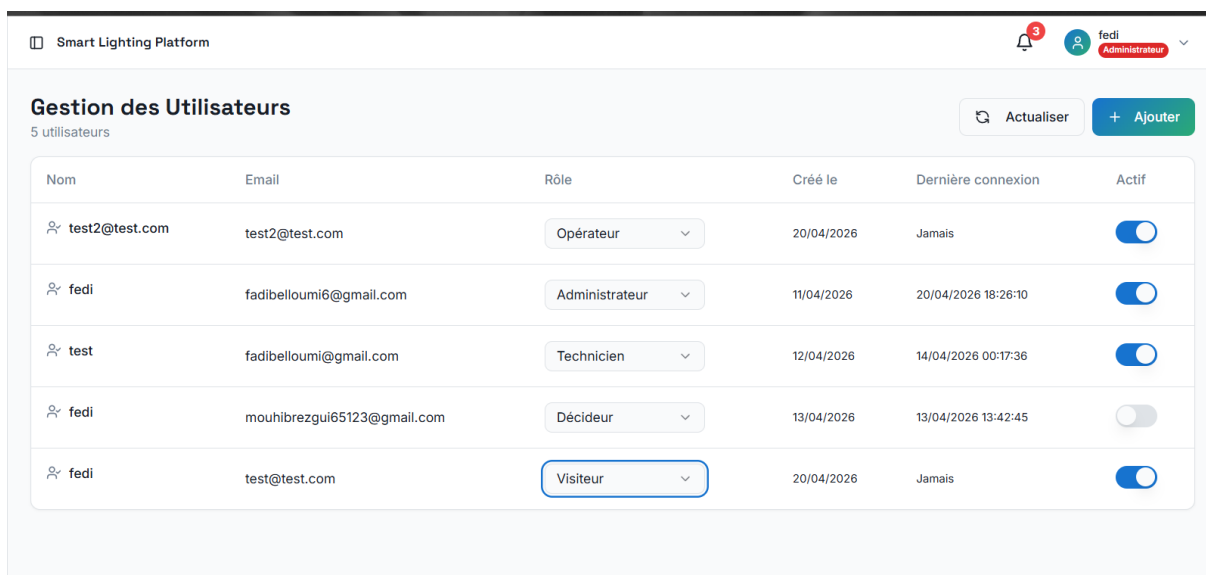
Confirmer le nouveau mot de passe

Changer le mot de passe

FIGURE 3.13 – Interface changer le mot de passe

Interface gestion des utilisateurs

L'interface de gestion des utilisateurs permet à l'administrateur de superviser et de gérer l'ensemble des comptes utilisateurs de la plateforme Smart Lighting. Elle offre une vue globale des utilisateurs ainsi que des fonctionnalités de contrôle des accès et des rôles.



Smart Lighting Platform

Gestion des Utilisateurs

5 utilisateurs

Actualiser + Ajouter

Nom	Email	Rôle	Créé le	Dernière connexion	Actif
test2@test.com	test2@test.com	Opérateur	20/04/2026	Jamais	☒
fedi	fadibelloumi6@gmail.com	Administrateur	11/04/2026	20/04/2026 18:26:10	☒
test	fadibelloumi@gmail.com	Technicien	12/04/2026	14/04/2026 00:17:36	☒
fedi	mouhibrezgui65123@gmail.com	Décideur	13/04/2026	13/04/2026 13:42:45	☐
fedi	test@test.com	Visiteur	20/04/2026	Jamais	☒

FIGURE 3.14 – Interface gestion des utilisateurs

3.5.1 Tests

Le tableau suivant présente les tests effectués pour le Sprint 1.

Cas de test	Démarche	Comportement attendu	Résultat
Test de consultation de la page d'accueil	Accéder à la page d'accueil sans authentification et consulter les KPI et la carte affichés.	Le système affiche la page d'accueil avec les statistiques publiques et la carte interactive accessibles sans authentification.	Conforme
Test d'inscription	Accéder à la page d'inscription, remplir le formulaire avec les informations requises (nom, email, mot de passe), puis soumettre.	Le système valide les données, enregistre le compte, envoie un email de vérification et redirige l'utilisateur vers la page de confirmation.	Conforme
Test de vérification email	Accéder à la boîte mail, récupérer le code de vérification reçu, puis le saisir dans la page de validation.	Le système vérifie le code, active le compte utilisateur et redirige vers la page de connexion.	Conforme
Test d'authentification	Accéder à la page de connexion, saisir l'email et le mot de passe valides, puis soumettre le formulaire.	Le système vérifie les identifiants, génère un code 2FA, l'envoie par email et redirige vers la page de vérification 2FA.	Conforme
Test de vérification 2FA	Récupérer le code 2FA reçu par email, le saisir dans la page de vérification, puis valider.	Le système vérifie le code, génère un token JWT et redirige l'utilisateur vers le tableau de bord.	Conforme

Cas de test	Démarche	Comportement attendu	Résultat
Test de réinitialisation du mot de passe	Accéder à la page « Mot de passe oublié », saisir l'adresse email, puis suivre le processus de réinitialisation via le code reçu par email.	Le système envoie un code de réinitialisation, vérifie le code saisi et permet la modification du mot de passe.	Conforme
Test de gestion du compte	Accéder à la page « Mon compte », modifier le mot de passe actuel en saisissant l'ancien et le nouveau mot de passe, puis valider.	Le système vérifie l'ancien mot de passe, enregistre le nouveau et affiche un message de confirmation.	Conforme
Test de gestion des utilisateurs	Se connecter en tant qu'administrateur, accéder à la page « Utilisateurs », consulter la liste des comptes et modifier le rôle ou le statut d'un utilisateur.	Le système affiche la liste complète des utilisateurs avec leurs informations et applique les modifications de rôle et de statut correctement.	Conforme
Test d'activation et désactivation d'un compte	Accéder à la page « Utilisateurs » en tant qu'administrateur, sélectionner un compte et basculer son statut actif/inactif.	Le système met à jour le statut du compte et l'utilisateur concerné ne peut plus se connecter si son compte est désactivé.	Conforme

TABLE 3.5 – Tests du Sprint 1

3.6 Utilisation des outils de suivi Scrum

Dans cette partie nous allons présenter les outils de suivi de la méthodologie adoptée Scrum : le Scrum board et le Burn down chart.

3.6.1 Scrum Board

À la fin de la deuxième semaine du premier sprint, l'état d'avancement de notre projet est illustré dans la figure ci-dessous. Six cas d'utilisation ont déjà été terminés, notamment : s'inscrire, réinitialiser mot de passe, s'authentifier, ajouter expériences, et ajouter compétences. Cinq cas d'utilisation sont en cours de réalisation, parmi lesquels : Gérer comptes utilisateurs, changer mot de passe, consulter compte, modifier compte et consulter profil CV. Enfin, trois cas d'utilisation restent à accomplir : consulter détails compte recruteurs, approuver compte, et consulter détails compte candidats, afin d'atteindre l'ensemble des objectifs fixés pour ce sprint.

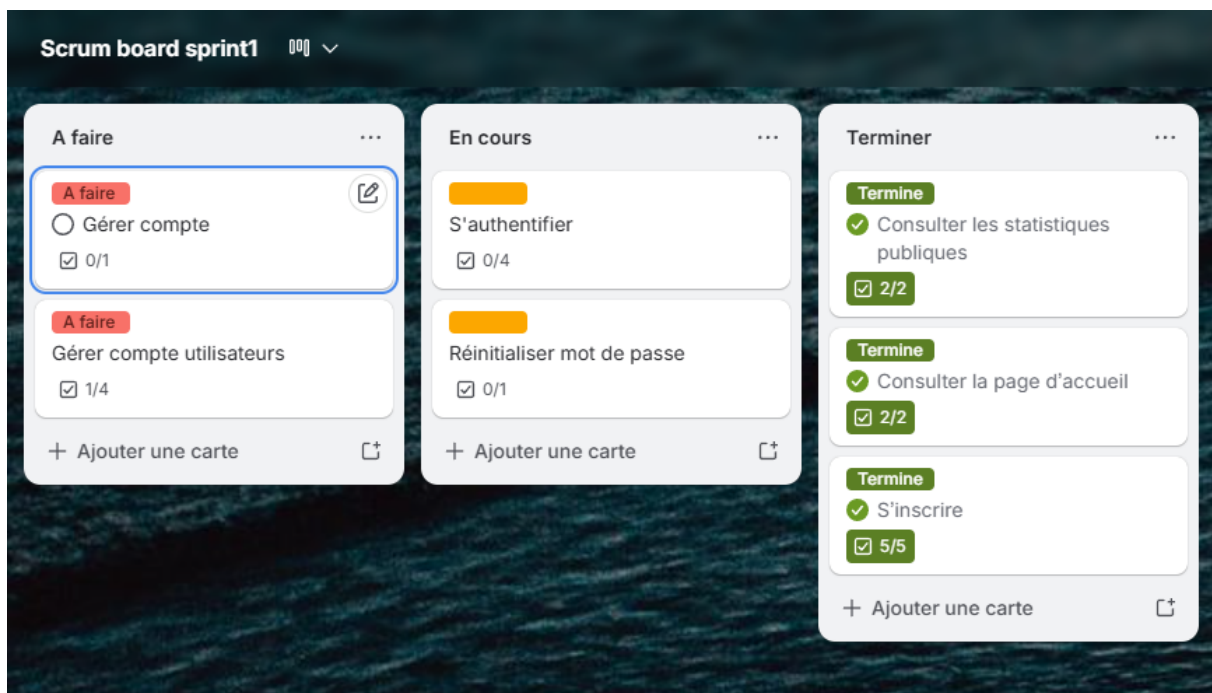


FIGURE 3.15 – Scrum Board sprint 1

3.6.2 Burn Down Chart

La figure 3.16 présente le Burn Down Chart illustrant l'évolution du premier sprint. Les fluctuations observées à certains moments sur ce graphique reflètent la capacité de l'équipe à s'adapter aux contraintes du projet ainsi qu'à intégrer de nouvelles technologies au fur et à mesure de leur avancement.

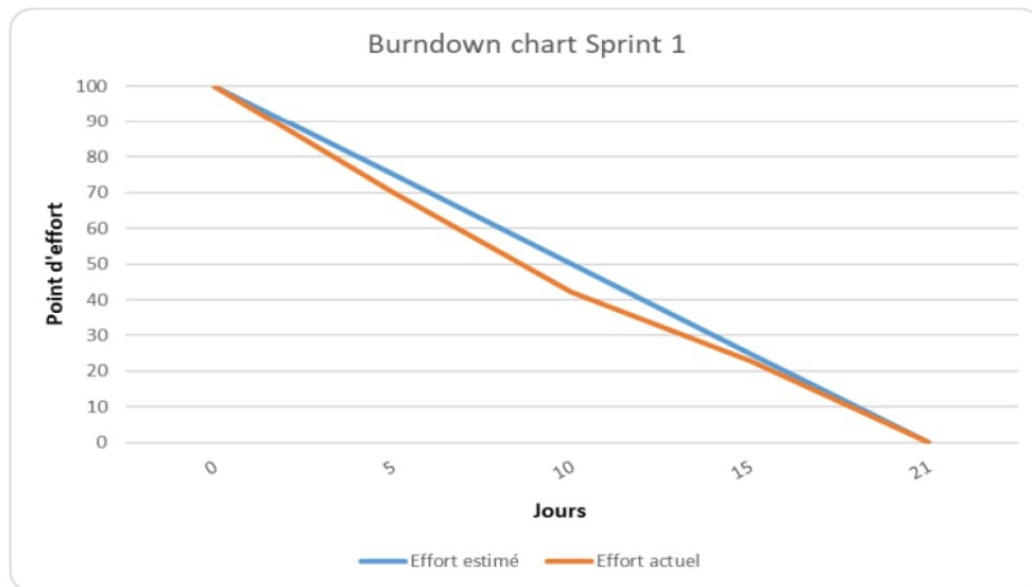


FIGURE 3.16 – Burn down chart à la fin de 1er sprint

3.7 Conclusion

Ce premier sprint a permis de poser les bases essentielles de l'application, notamment la gestion des comptes utilisateurs, l'authentification, la modification du profil et la réinitialisation de mot de passe. Toutes les tâches prévues ont été réalisées avec succès et testées. Grâce à l'utilisation de la méthode Scrum, l'équipe a su organiser efficacement le travail, respecter les délais et livrer une première version stable du système.

Chapitre 4

Sprint 2 :Étude, analyse et réalisation du Sprint 2

Plan

1. Introduction
2. Backlog du Sprint
3. Spécification de besoins du sprint 2
4. Conception
5. Réalisation et tests
6. Utilisation des outils de suivi Scrum
7. Conclusion

4.1 Introduction

L'objectif du Sprint 2 est de concevoir et de implémenter un module de gestion intelligente des infrastructures d'éclairage public. Ce module doit permettre la gestion des entités géographiques appelées zones, ainsi que la gestion des lampadaires associés à ces zones. Il doit aussi offrir une interface cartographique interactive permettant de localiser les équipements, d'afficher leurs informations essentielles et de simplifier les opérations de supervision. Ce sprint vise ainsi à établir une représentation structurée et exploitable du réseau d'éclairage dans la plateforme.

4.2 Backlog du sprint 2

Le Sprint 2 est consacré à la gestion des zones et des lampadaires, ainsi qu'à leur visualisation sur une carte interactive. Les fonctionnalités définies dans ce sprint permettent aux différents acteurs du système, à savoir l'administrateur, l'opérateur et le technicien, d'effectuer les opérations de création, de consultation, de modification et de suppression sur les zones et les lampadaires. Ce sprint intègre également une interface cartographique facilitant la localisation et la consultation des équipements.

ID US	User Story	ID tâche	Tâche	Statut	Complexité
1	En tant qu'administrateur / opérateur / technicien, je veux gérer les zones afin d'organiser le réseau d'éclairage.	1.1	Conception du modèle Zone	Terminé	Faible
		1.2	Développement API CRUD des zones	Terminé	Élevé
		1.3	Développement interface gestion des zones	Terminé	Moyen
		1.4	Test du module de gestion des zones	Terminé	Faible
2	En tant qu'administrateur / opérateur / technicien, je veux gérer les lampadaires afin de structurer le réseau d'éclairage.	2.1	Conception du modèle Lampadaire	Terminé	Faible
		2.2	Développement API CRUD lampadaires	Terminé	Élevé
		2.3	Association lampadaire à une zone	Terminé	Moyen

ID US	User Story	ID tâche	Tâche	Statut	Complexité
		2.4	Développement in- terface lampadaires	Terminé	Moyen
		2.5	Test du module lam- padaires	Terminé	Faible
3	En tant qu'admini- strateur / opéra- teur / technicien, je veux consulter la carte interactive afin de localiser les équipements.	3.1	Intégration bi- bliothèque carte (Leaflet)	Terminé	Élevé
		3.2	Affichage des lampa- daires sur la carte	Terminé	Moyen
		3.3	Ajout des popups d'information	Terminé	Moyen
		3.4	Test de la carte inter- active	Terminé	Faible

TABLE 4.2 – Backlog détaillé du Sprint 2

4.3 Spécification de besoins du sprint 2

Cette section présente les besoins fonctionnels traités lors du deuxième sprint, en mettant en évidence les principaux cas d'utilisation associés.

4.3.1 Diagramme de cas d'utilisation

Dans le but de mieux illustrer les fonctionnalités développées lors de ce sprint, le diagramme de cas d'utilisation est présenté dans la figure ci-après.

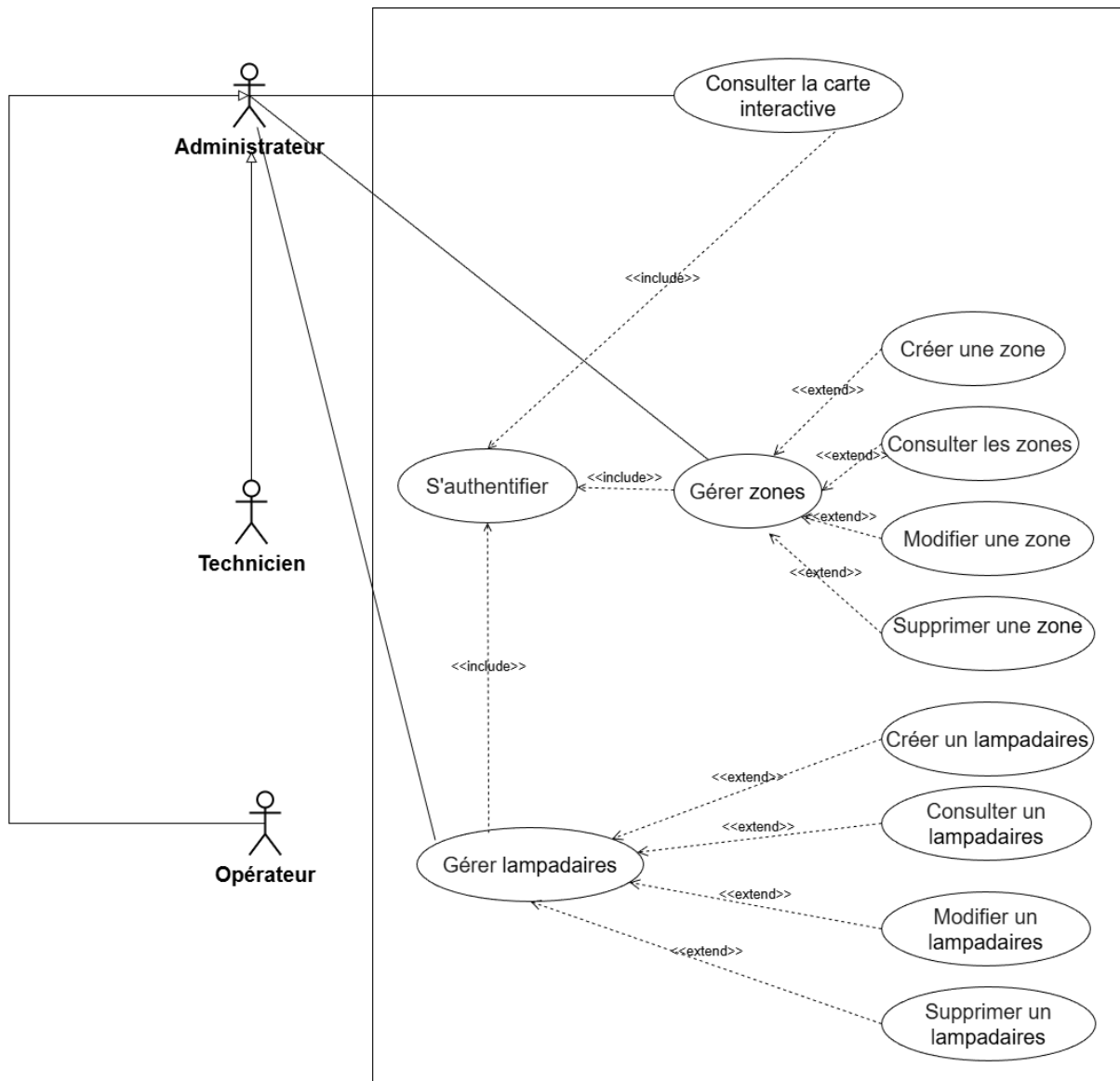


FIGURE 4.1 – Diagramme de cas d'utilisation du Sprint 2

Description du cas d'utilisation « Créer une zone »

Le tableau suivant présente la description du cas d'utilisation « Créer une zone ».

Cas d'utilisation	Créer une zone
Acteurs	Administrateur, Opérateur , Technicien
Description brève du CU	Permet à l'acteur de créer une nouvelle zone géographique afin de regrouper les lampadaires par secteur dans la plateforme.
Préconditions	L'utilisateur est authentifié et dispose des droits nécessaires.
Post conditions	La zone est créée et enregistrée dans la base de données. Elle devient disponible pour y associer des lampadaires.
Scénario de base	1. L'utilisateur s'authentifie. 2. Le système affiche l'espace de l'utilisateur. 3. L'utilisateur accède à la page de gestion des zones. 4. Le système affiche la liste des zones existantes. 5. L'utilisateur clique sur le bouton « Ajouter ». 6. Le système affiche le formulaire de création. 7. L'utilisateur saisit les informations requises (nom, description, localisation). 8. L'utilisateur clique sur le bouton « Enregistrer ». 9. Le système valide les données saisies. 10. Le système affiche une alerte de succès. 11. L'utilisateur clique sur le bouton « OK ». 12. Le système enregistre la zone dans la base de données.
Autre scénario	9.1. Si un champ obligatoire est manquant → affichage d'un message d'erreur de validation. 9.2. Si une zone avec le même nom existe déjà → le système refuse la création et notifie l'utilisateur.

TABLE 4.3 – Description du cas d'utilisation « Créer une zone »

Description du cas d'utilisation « Consulter la carte interactive »

Le tableau suivant présente la description du cas d'utilisation « Consulter la carte interactive ».

Cas d'utilisation	Consulter la carte interactive
Acteurs	Administrateur, Opérateur, Technicien
Description brève du CU	Permet à l'acteur de visualiser l'ensemble des zones et des lampadaires géolocalisés sur une carte interactive afin d'assurer la supervision spatiale du réseau d'éclairage public.
Préconditions	L'utilisateur est authentifié. Des zones et/ou des lampadaires ont été préalablement enregistrés dans le système.
Post conditions	L'utilisateur visualise la carte avec les marqueurs représentant les zones et les lampadaires, accompagnés de leurs informations de statut.
Scénario de base	1. L'utilisateur s'authentifie. 2. L'utilisateur accède à la page de la carte interactive. 3. L'utilisateur clique bouton " Carte interactive". 4. Le système charge les données géographiques des zones et des lampadaires depuis la base de données. 5. Le système affiche la carte avec les marqueurs positionnés aux coordonnées GPS de chaque entité. 6. L'utilisateur peut zoomer, dézoomer et naviguer sur la carte. 7. Le système affiche une popup avec les informations détaillées de l'entité (nom, statut, zone associée).
Autre scénario	4.1. Si aucune donnée n'est disponible → le système affiche la carte vide avec un message informatif. 5.2. Si le chargement des données échoue → le système affiche un message d'erreur et propose de réessayer.

TABLE 4.4 – Description du cas d'utilisation « Consulter la carte interactive »

4.4 Conception

Cette section présente les diagrammes de séquence et de classes élaborés pour modéliser le comportement dynamique et la structure statique du système au cours du Sprint 2.

4.4.1 Diagrammes de séquences sprint 2

Cette section expose les différents diagrammes de séquence réalisés dans le cadre du Sprint 2.

4.4.2 Diagramme de séquence du scénario « Créer une zone »

La figure illustre le diagramme de séquence du scénario « Créer une zone », en présentant les interactions entre l'utilisateur et le système.

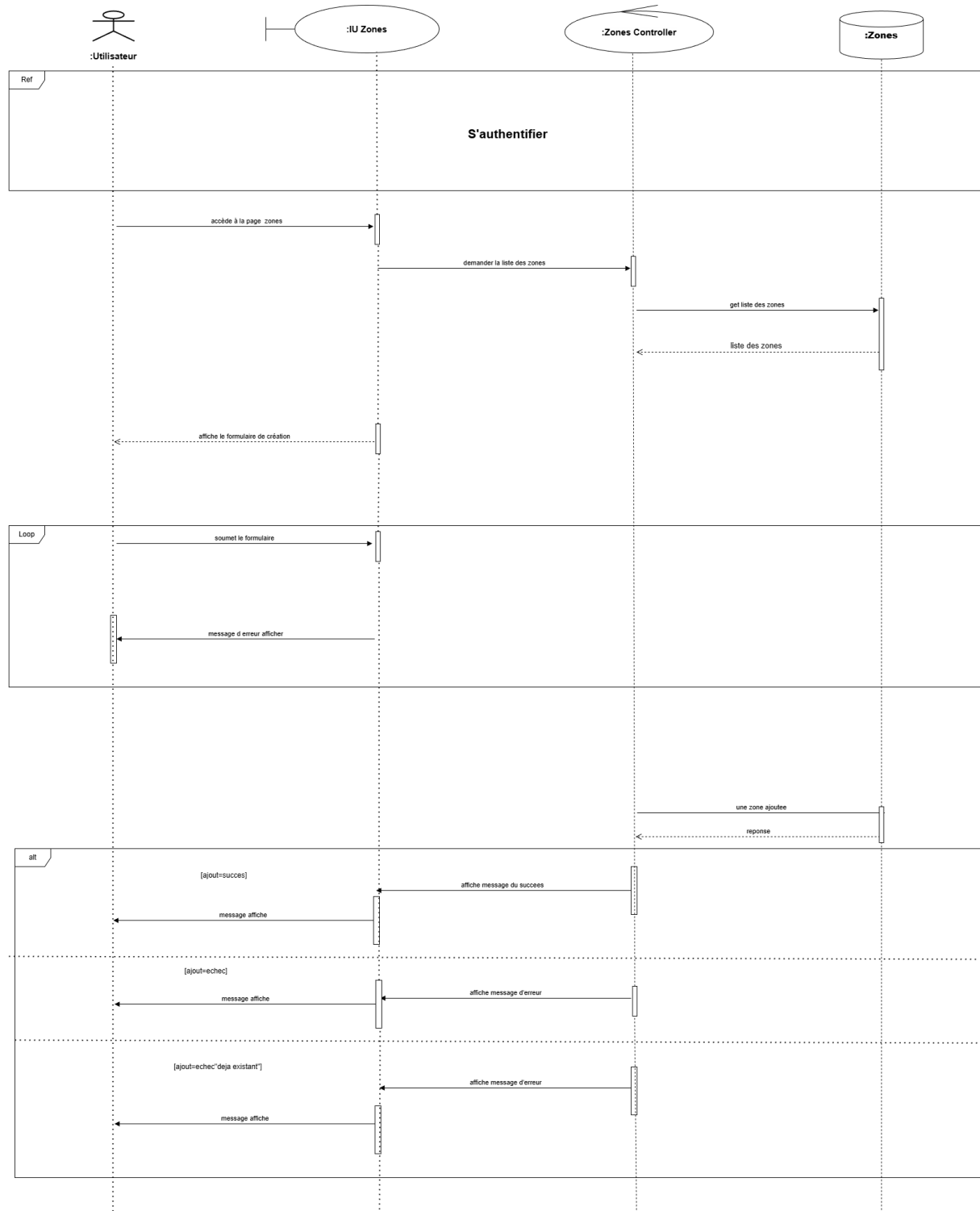


FIGURE 4.2 – Diagramme de séquence du scénario « Créer une zone »

4.4.3 Diagramme de séquence du scénario « Consulter la carte interactive »

La figure illustre le diagramme de séquence du scénario « Consulter la carte interactive », en présentant les interactions entre l'utilisateur et le système.

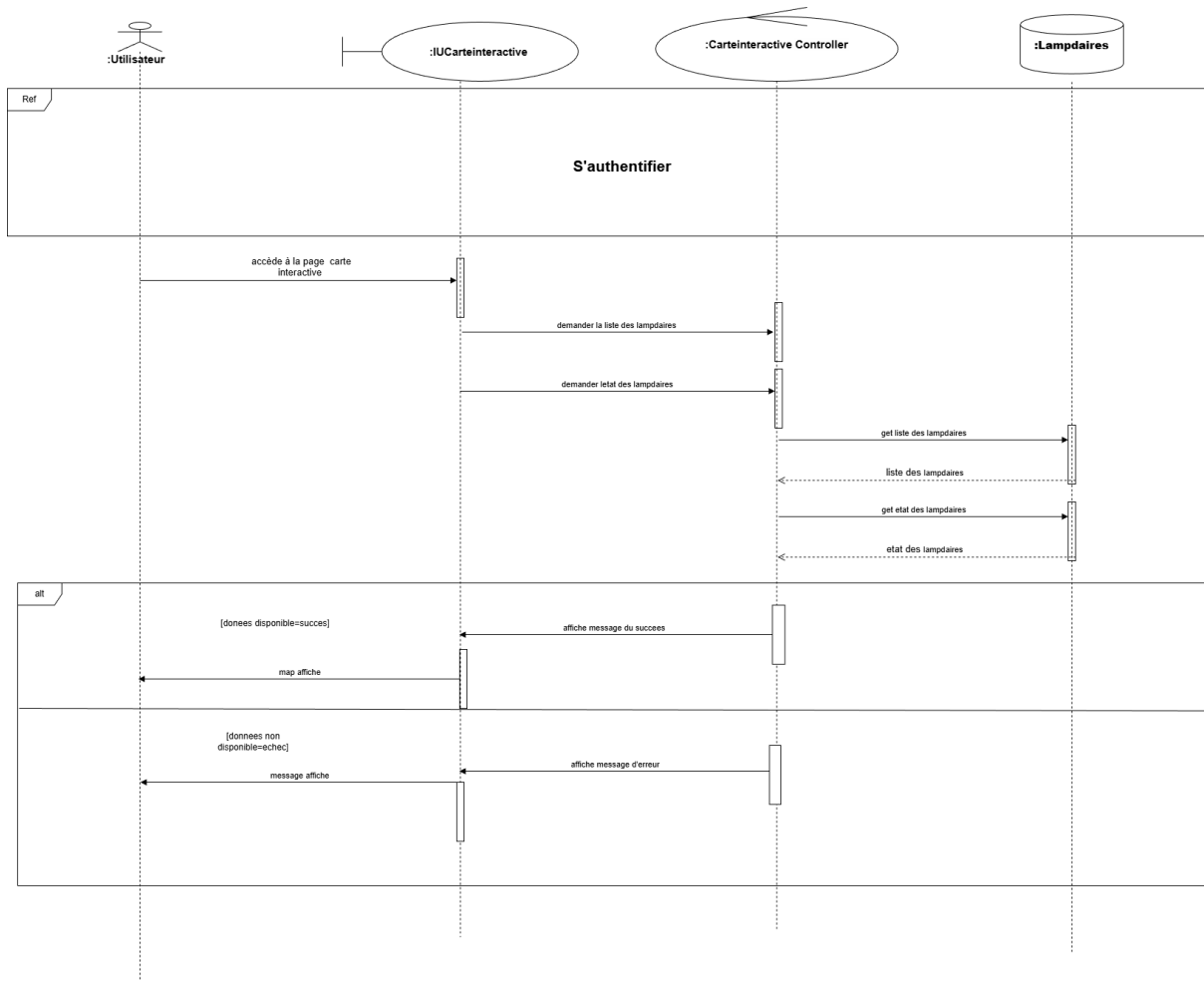


FIGURE 4.3 – Diagramme de séquence du scénario « Consulter la carte interactive »

4.4.4 Diagramme de Classes du sprint 2

Le diagramme de classes présenté dans la figure met en évidence les principales entités du système ainsi que les relations qui les lient dans le cadre du Sprint 2.

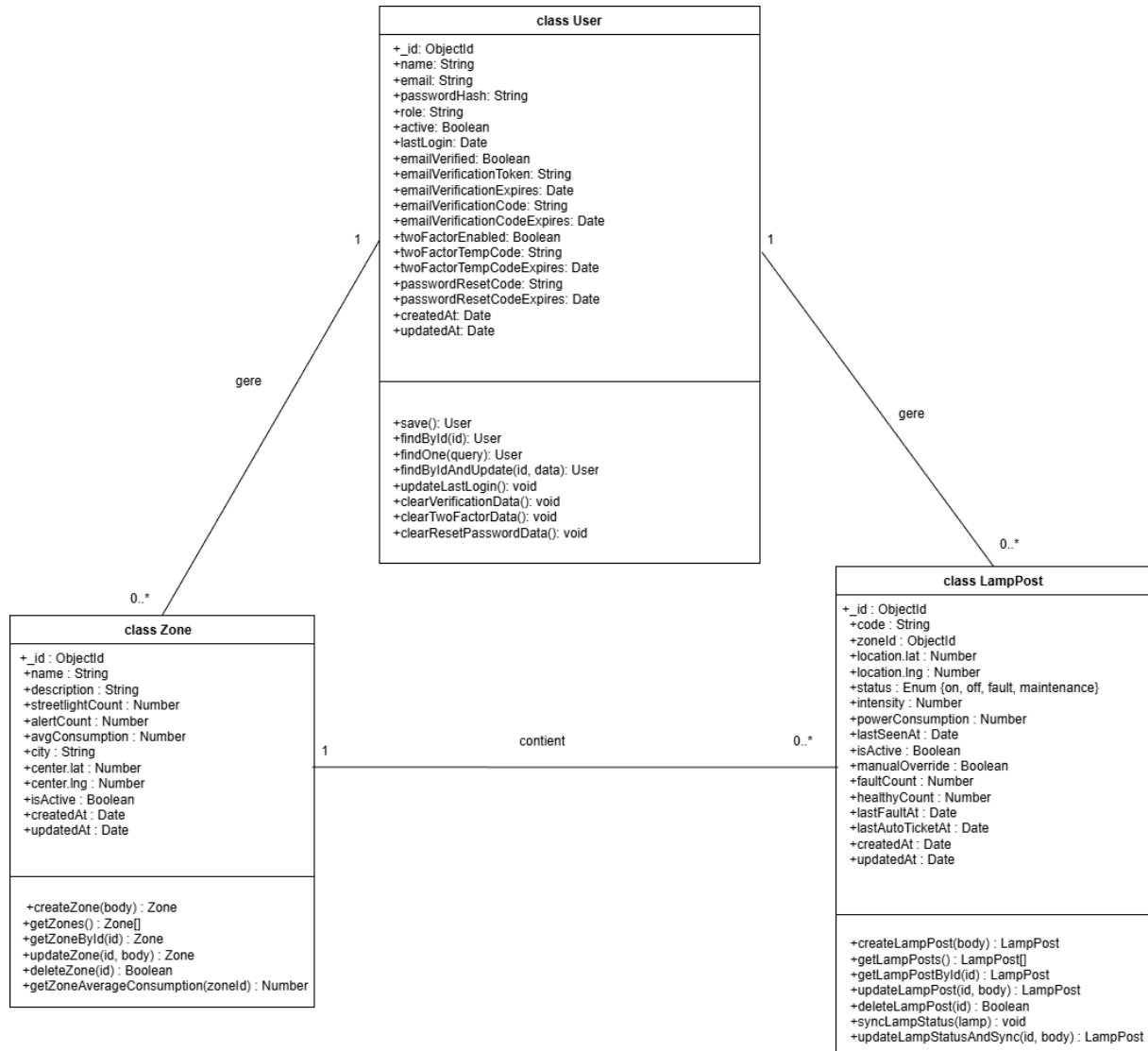


FIGURE 4.4 – Diagramme de Classes du sprint 2

4.5 Réalisation et tests

Cette section décrit le processus de mise en œuvre du système ainsi que les différents tests réalisés afin de valider son bon fonctionnement et sa conformité aux exigences spécifiées.

Cette section présente les différentes interfaces réalisées durant ce sprint.

Dans cette section, nous présentons les interfaces développées au cours de ce sprint.

Interface Carte Interactive

L'interface « Carte Interactive » permet à l'utilisateur de visualiser l'ensemble des zones et des lampadaires géolocalisés sur une carte dynamique. Elle constitue le principal outil de supervision spatiale du réseau d'éclairage public.

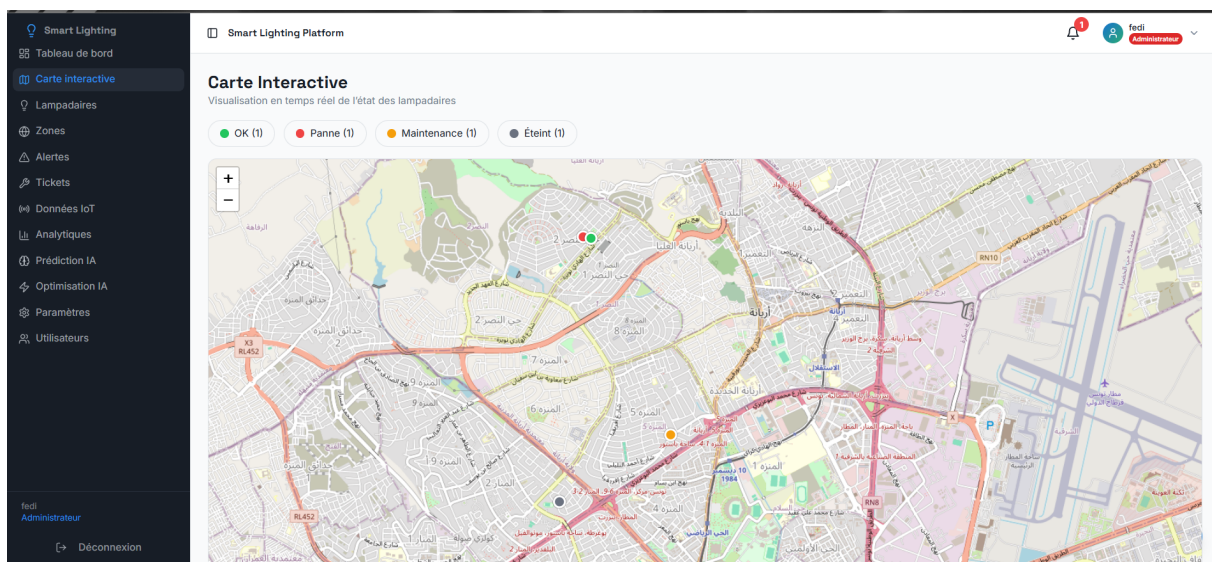


FIGURE 4.5 – Interface carte interactive

Interface Gestion des Lampadaires

L'interface « Gestion des Lampadaires » permet à l'utilisateur de consulter, ajouter, modifier et supprimer les lampadaires enregistrés dans le système. Elle constitue le principal outil de gestion des équipements d'éclairage public de la plateforme.

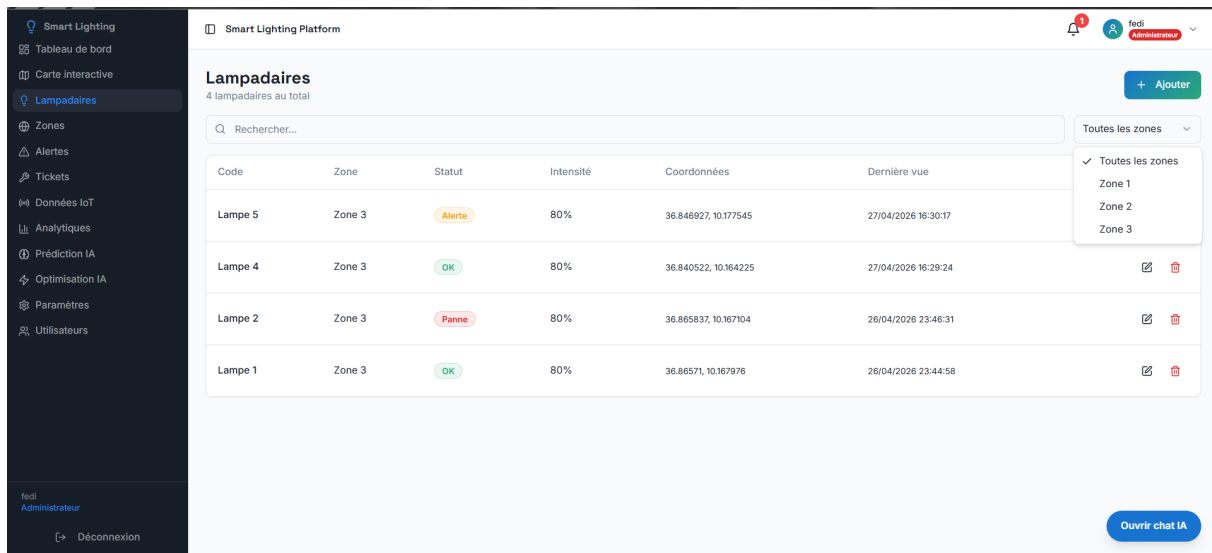


FIGURE 4.6 – Interface Gestion des Lampadaires

Interface Nouveau Lampadaire

L'interface « Nouveau Lampadaire » permet à l'utilisateur d'enregistrer un nouveau lampadaire dans le système en renseignant ses informations techniques et en l'associant à une zone géographique existante.

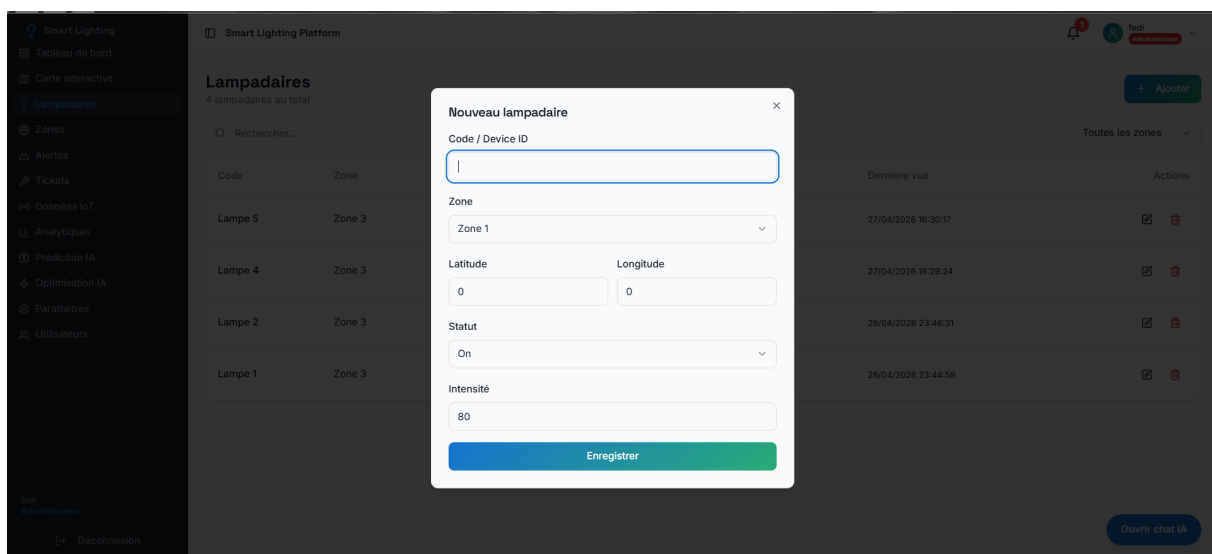


FIGURE 4.7 – Interface Nouveau Lampadaire

Interface Modifier Lampadaire

L'interface « Modifier Lampadaire » permet à l'utilisateur de mettre à jour les informations d'un lampadaire existant dans le système en modifiant ses caractéristiques techniques ou sa zone d'affectation.

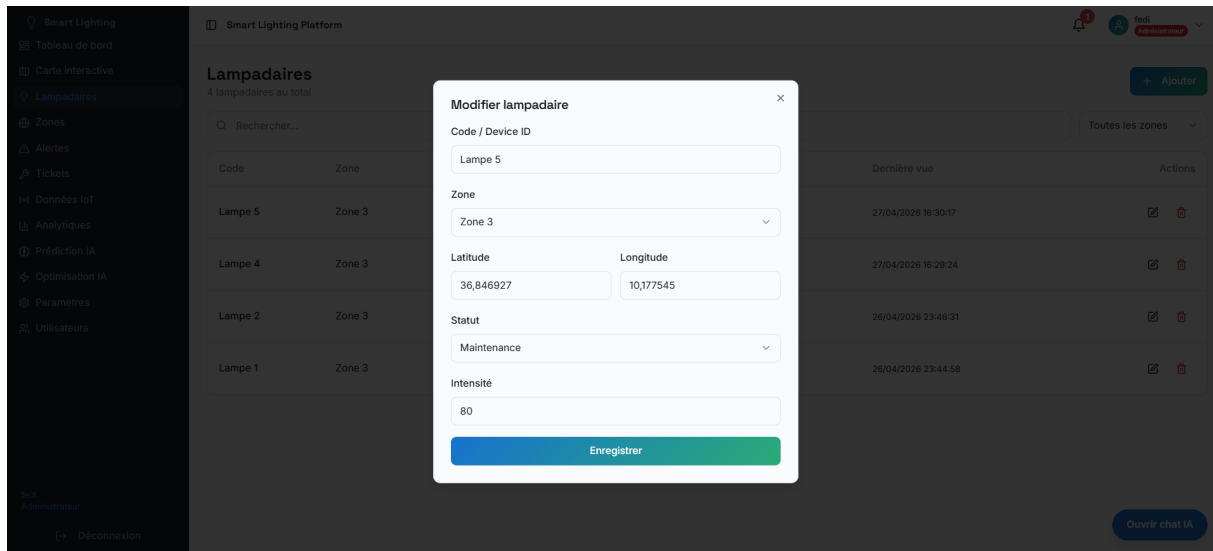


FIGURE 4.8 – Interface Modifier Lampadaire

Interface Gestion des Zones

L'interface « Gestion des Zones » permet à l'utilisateur de consulter, ajouter, modifier et supprimer les zones géographiques enregistrées dans le système. Elle constitue le principal outil d'organisation spatiale du réseau d'éclairage public.

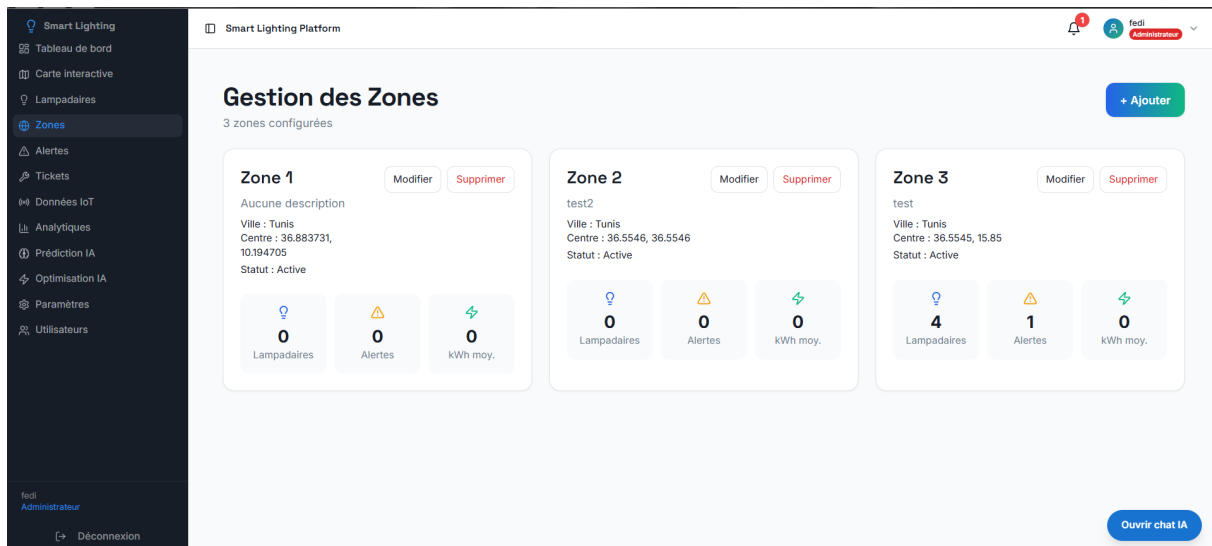


FIGURE 4.9 – Interface Gestion des Zones

Interface Nouvelle Zone

L'interface « Nouvelle Zone » permet à l'utilisateur de créer une nouvelle zone géographique dans le système en renseignant ses informations descriptives et sa localisation sur la carte.

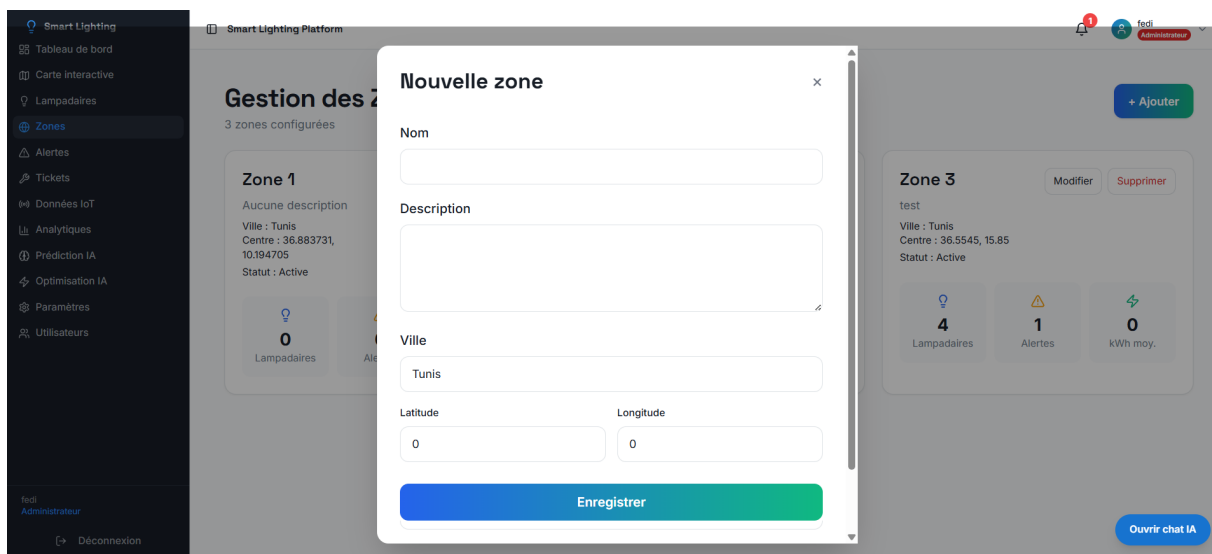


FIGURE 4.10 – Interface Nouvelle des Zones

Interface Modifier Zone

L'interface « Modifier Zone » permet à l'utilisateur de mettre à jour les informations d'une zone géographique existante dans le système en modifiant ses attributs descriptifs ou sa localisation.

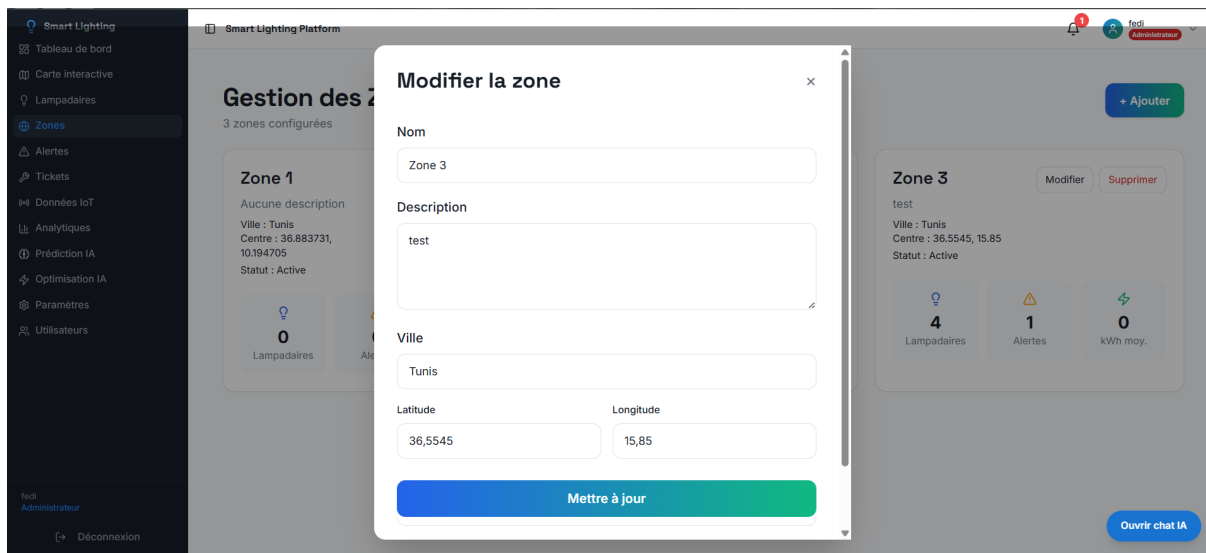


FIGURE 4.11 – Interface Modifier Zones

4.5.1 Tests

Le tableau suivant présente les tests effectués pour le Sprint 2.

Cas de test	Démarche	Comportement attendu	Résultat
Test de création d'une zone	Accéder à la page « Gestion des zones », cliquer sur « Ajouter une zone », remplir le formulaire avec le nom, la description et la localisation, puis soumettre.	Le système valide les données, enregistre la zone dans la base de données et affiche un message de confirmation.	Conforme
Test de modification d'une zone	Accéder à la liste des zones, sélectionner une zone existante, modifier ses informations, puis valider les modifications.	Le système enregistre les modifications et actualise la liste des zones avec les nouvelles informations.	Conforme
Test de suppression d'une zone	Accéder à la liste des zones, sélectionner une zone, cliquer sur « Supprimer », puis confirmer la suppression dans la boîte de dialogue.	Le système supprime la zone de la base de données et actualise la liste des zones.	Conforme
Test d'ajout d'un lampadaire	Accéder à la page « Gestion des lampadaires », cliquer sur « Ajouter un lampadaire », sélectionner une zone, renseigner les informations techniques, puis soumettre.	Le système valide les données, enregistre le lampadaire, l'associe à la zone sélectionnée et met à jour la carte interactive.	Conforme
Test de modification d'un lampadaire	Accéder à la liste des lampadaires, sélectionner un lampadaire existant, modifier ses paramètres techniques, puis valider les modifications.	Le système enregistre les modifications et actualise la liste des lampadaires avec les nouvelles informations.	Conforme
Test de suppression d'un lampadaire	Accéder à la liste des lampadaires, sélectionner un lampadaire, cliquer sur « Supprimer », puis confirmer la suppression.	Le système désactive le lampadaire (soft delete), le retire de la carte interactive et actualise la liste.	Conforme

Cas de test	Démarche	Comportement attendu	Résultat
Test de consultation des zones et lampadaires	Accéder à la page de gestion, consulter la liste des zones et des lampadaires, puis utiliser les filtres et la recherche.	Le système affiche la liste complète des zones et lampadaires avec leurs informations et les filtres fonctionnent correctement.	Conforme
Test de la carte interactive	Accéder à la page « Carte interactive », visualiser les marqueurs des zones et lampadaires, puis cliquer sur un marqueur.	Le système affiche la carte avec tous les marqueurs géolocalisés et les popups d'information au clic.	Conforme

TABLE 4.5 – Tests du Sprint 2

4.6 Utilisation des outils de suivi Scrum

Dans cette partie nous allons présenter les outils de suivi de la méthodologie adoptée Scrum : le Scrum board et le Burn down chart.

4.6.1 Scrum Board

À la fin du Sprint 2, l'état d'avancement du projet est illustré dans la figure ci-dessous. Quatre cas d'utilisation ont déjà été terminés, notamment : consulter la carte, consulter les zones, créer un lampadaire et consulter un lampadaire. Cinq cas d'utilisation sont en cours de réalisation, parmi lesquels : modifier une zone, créer une zone, supprimer une zone, modifier un lampadaire et supprimer un lampadaire. Enfin, deux cas d'utilisation restent à accomplir : gérer les lampadaires et gérer les zones, afin d'atteindre l'ensemble des objectifs fixés pour ce sprint.

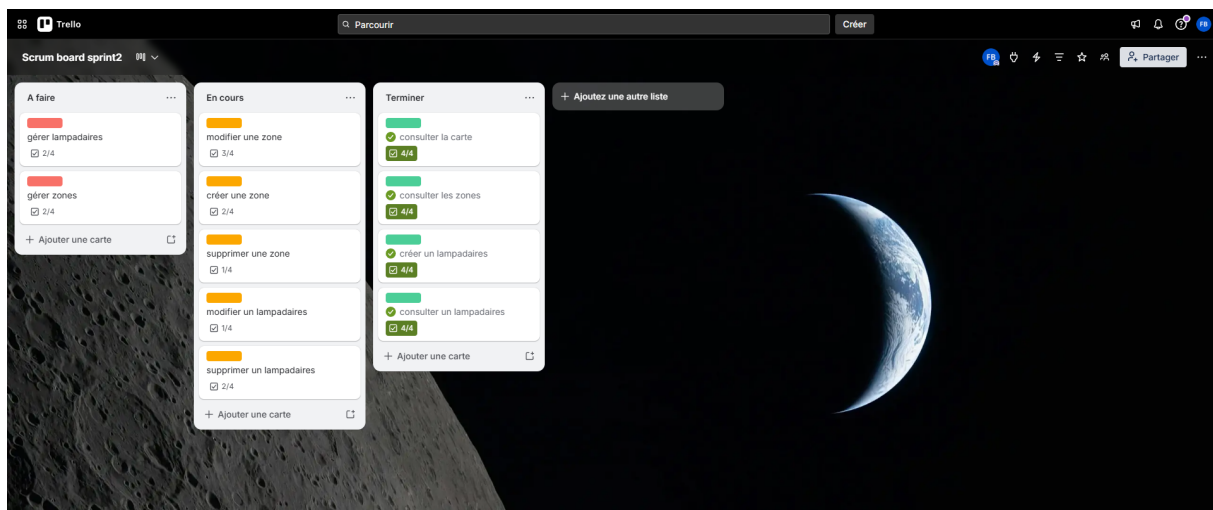


FIGURE 4.12 – Tableau Scrum Sprint 2

4.6.2 Burn Down Chart

La FIGURE 51 ci-dessous montre l'évolution de notre avancement au cours de ce sprint. Au départ, le travail a progressé de façon régulière, mais des obstacles rencontrés

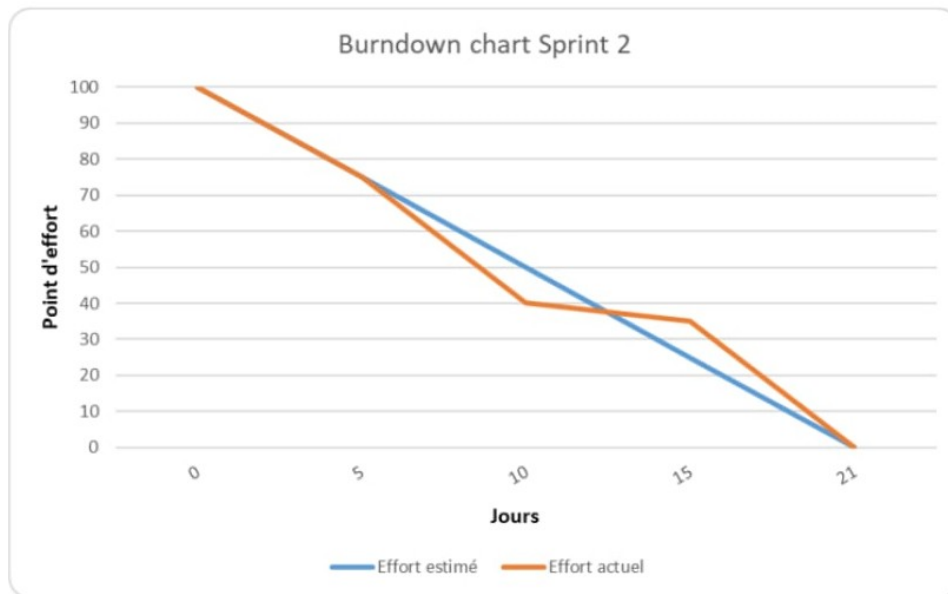


FIGURE 4.13 – Burn down à la fin de 2ème Sprint

4.7 Conclusion

Ce sprint a permis de concevoir et d'implémenter les modules fondamentaux de gestion des infrastructures d'éclairage public de la plateforme Smart Lighting. La mise en place de la gestion des zones et des lampadaires, couplée à une interface cartographique interactive, offre désormais une représentation structurée, exploitable et visuellement intuitive du réseau d'éclairage public. Ces fonctionnalités constituent le socle métier sur lequel s'appuieront les sprints suivants, notamment la gestion des alertes, des tickets, la gestion des tickets de maintenance, la consultation des données IoT en temps réel et le système de notifications

Chapitre 5

Sprint 3 : Étude, analyse et réalisation du Sprint 3

Plan

1. Introduction
2. Backlog du Sprint
3. Spécification des besoins du Sprint 3
4. Conception
5. Réalisation et tests
6. Utilisation des outils de suivi Scrum
7. Conclusion

5.1 Introduction

Le Sprint 3 a pour objectif de concevoir et d'implémenter les modules de supervision opérationnelle de la plateforme Smart Lighting. Il introduit quatre composants clés : la gestion des alertes, permettant de détecter automatiquement les anomalies du réseau ; la gestion des tickets de maintenance, assurant le suivi des interventions de leur ouverture jusqu'à leur résolution ; la consultation des données IoT en temps réel, offrant une visualisation des métriques collectées par les capteurs ; et le système de notifications, informant les acteurs de tout événement critique survenant sur le réseau. Ce sprint constitue une étape charnière vers une supervision proactive et efficace de la plateforme.

5.2 Backlog du Sprint 3

Le tableau suivant présente l'ensemble des User Stories planifiées et réalisées dans le cadre du Sprint 3.

Cas d'utilisation	Consulter et Supprimer les notifications
Acteurs	Administrateur, Opérateur, Technicien
Description brève du CU	Permet à l'acteur de consulter l'ensemble des notifications générées par le système concernant les événements critiques survenus sur le réseau d'éclairage public, et de supprimer celles qui ne sont plus nécessaires.
Préconditions	L'utilisateur est authentifié. Des notifications ont été générées par le système.
Post conditions	L'utilisateur visualise la liste des notifications avec leurs détails et leur statut. La notification supprimée est retirée définitivement de la base de données.
Scénario de base	1. L'utilisateur accède à la page des notifications. 2. Le système envoie une requête au serveur pour récupérer les notifications. 3. Le serveur retourne la liste des notifications. 4. Le système affiche les notifications avec leur type, message et statut. 5. L'utilisateur consulte les détails d'une notification. 6. L'utilisateur sélectionne une notification à supprimer. 7. Le système affiche une boîte de dialogue de confirmation. 8. L'utilisateur confirme la suppression. 9. Le système supprime la notification de la base de données. 10. Le système affiche un message de confirmation.
Autre scénario	1. Si aucune notification n'existe → le système affiche un message informatif. 2. Si le chargement échoue → le système affiche un message d'erreur et propose de réessayer. 3. Si une erreur serveur survient → le système notifie l'utilisateur et annule l'opération.

Cas d'utilisation	Consulter et Supprimer les notifications
--------------------------	--

TABLE 5.1 – Description du cas d'utilisation « Consulter et Supprimer les notifications »

5.3 Spécification de besoins du sprint 3

Cette section présente les besoins fonctionnels traités lors du troisième sprint, en mettant en évidence les principaux cas d'utilisation associés.

5.3.1 Diagramme de cas d'utilisation

Dans le but de mieux illustrer les fonctionnalités développées lors de ce sprint, le diagramme de cas d'utilisation est présenté dans la figure ci-après.

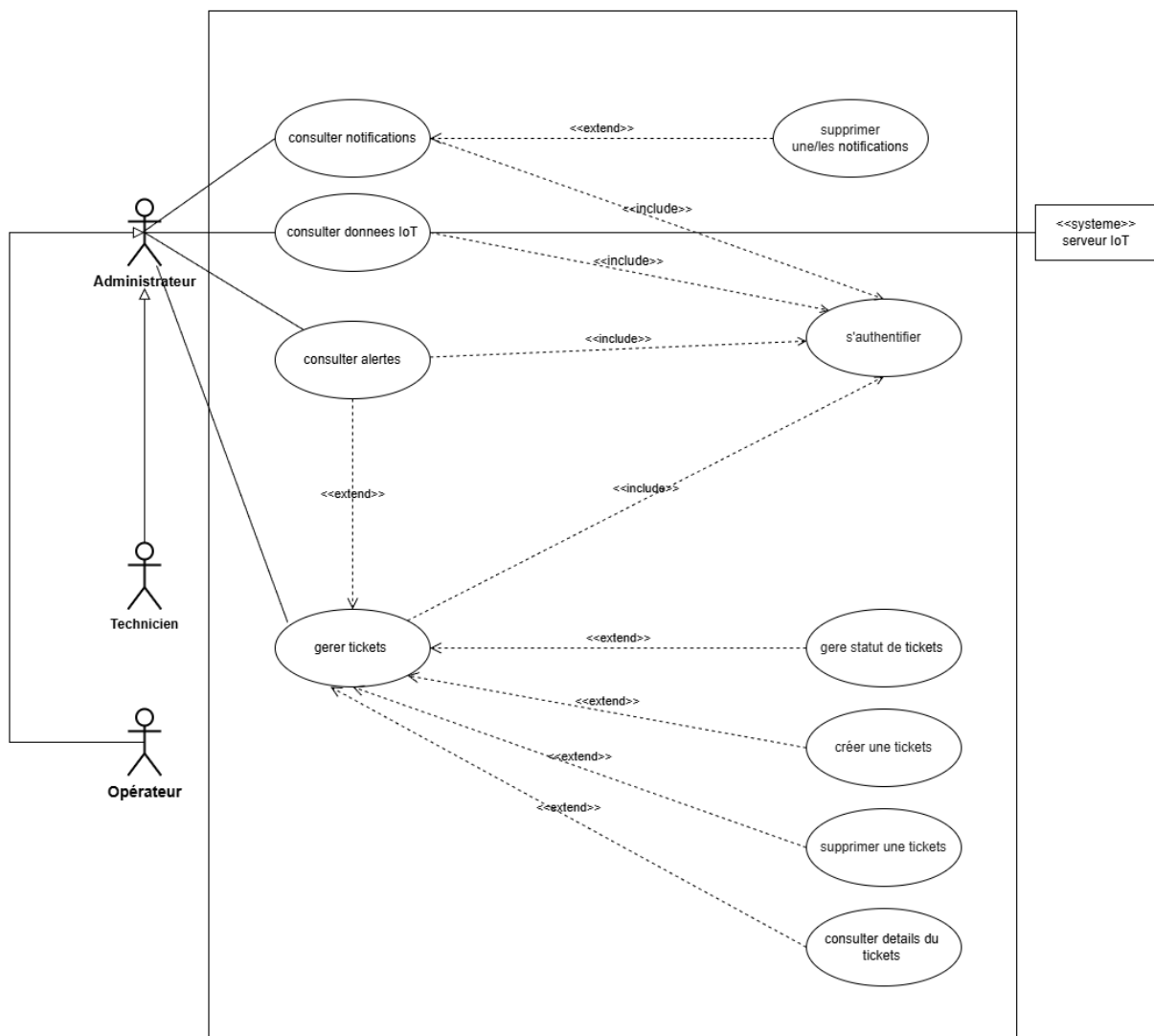


FIGURE 5.1 – Diagramme de cas d'utilisation du Sprint 3

Description du cas d'utilisation « Supprimer les notifications »

Le tableau suivant présente la description du cas d'utilisation « Supprimer les notifications ».

Cas d'utilisation	Supprimer les notifications
Acteurs	Administrateur, Opérateur, Technicien
Description brève du CU	Permet à l'acteur de consulter l'ensemble des notifications générées par le système concernant les événements critiques survenus sur le réseau d'éclairage public, et de supprimer celles qui ne sont plus nécessaires.
Préconditions	L'utilisateur est authentifié. Des notifications ont été générées par le système.
Post conditions	L'utilisateur visualise la liste des notifications avec leurs détails et leur statut. La notification supprimée est retirée définitivement de la base de données.
Scénario de base	1. L'utilisateur s'authentifier. 2. Le système affiche l'espace de l'utilisateur. 3. L'utilisateur clique bouton "Notification". 4. Le système envoie une requête au serveur pour récupérer les notifications. 5. Le serveur retourne la liste des notifications. 6. Le système affiche les notifications avec leur type, message et statut. 7. L'utilisateur consulte les détails d'une notification. 8. L'utilisateur sélectionne une notification à supprimer. 9. L'utilisateur clique bouton "X". 10. Le système supprime la notification de la base de données. 11. Le système affiche un message de confirmation.

Autre scénario	6.1. Si aucune notification n'existe → le système affiche un message informatif. 9.1. Si l'utilisateur clique sur « Effacer tout » → le système supprime toutes les notifications et affiche un message de succès.
-----------------------	---

TABLE 5.2 – Description du cas d'utilisation « Consulter et Supprimer les notifications »

Description du cas d'utilisation « Modifier un ticket »

Le tableau suivant présente la description du cas d'utilisation « Modifier un ticket ».

Cas d'utilisation	Modifier un ticket
Acteurs	Administrateur, Opérateur, Technicien
Description brève du CU	Permet à l'acteur de mettre à jour les informations d'un ticket de maintenance existant, notamment son titre, sa description, son statut et sa priorité, afin d'assurer un suivi précis et actualisé des interventions sur le réseau d'éclairage public.
Préconditions	L'utilisateur est authentifié. Le ticket à modifier existe dans la base de données.
Post conditions	Les informations du ticket sont mises à jour et persistées dans la base de données. La liste des tickets est actualisée en conséquence.

Scénario de base	1. L'utilisateur s'authentifier. 2. Le système affiche l'espace de l'utilisateur. 3. L'utilisateur accède à la page de gestion des tickets. 4. Le système récupère et affiche la liste des tickets existants avec leur statut et priorité. 5. L'utilisateur sélectionne le ticket à modifier. 6. Le système récupère les données du ticket et affiche le formulaire pré-rempli. 7. L'utilisateur modifie les informations souhaitées (titre, description, statut, priorité). 8. L'utilisateur soumet le formulaire et clique bouton "modifier". 9. Le système valide les données saisies. 10. Le système met à jour le ticket dans la base de données. 11. Le système affiche un message du succès.
Autre scénario	7.1. Si un champ obligatoire est manquant → le système affiche un message d'erreur de validation.

TABLE 5.3 – Description du cas d'utilisation « Modifier un ticket »

5.4 Conception

Cette section présente les diagrammes de séquence et de classes élaborés pour modéliser le comportement dynamique et la structure statique du système au cours du Sprint .

5.4.1 Diagrammes de séquences du sprint 3

Cette section expose les différents diagrammes de séquence réalisés dans le cadre du Sprint 3.

5.4.2 Diagramme de séquence du scénario « Supprimer une notification »

Le diagramme de séquence suivant présente le déroulement du processus de suppression d'une notification. Après authentification, l'utilisateur accède à la liste des notifications générées par le système et peut sélectionner une notification afin de la supprimer.

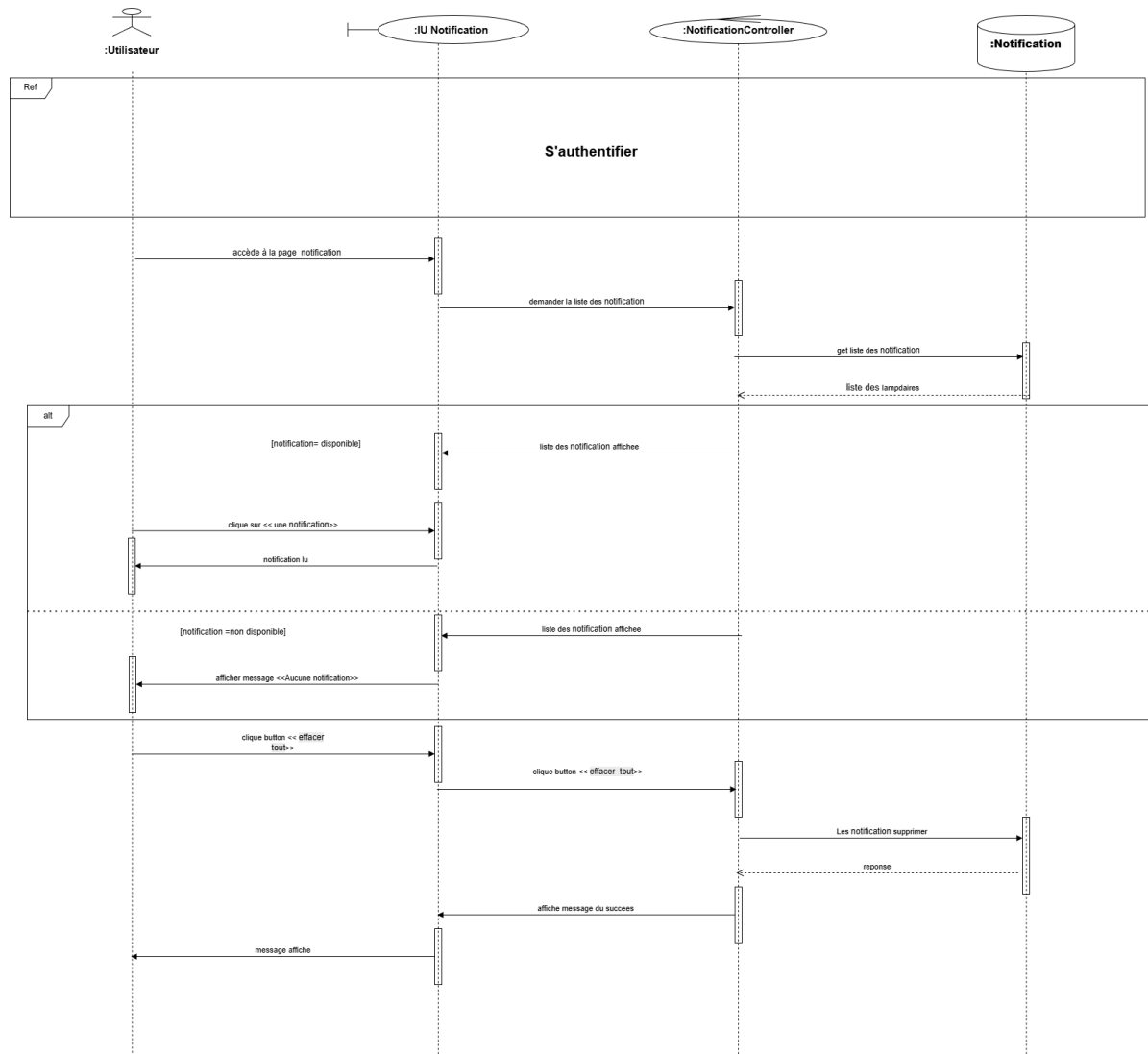


FIGURE 5.2 – Diagramme de séquence du scénario « Consulter et Supprimer une notification »

5.4.3 Diagramme de séquence du scénario «Modifier ticket »

Le diagramme de séquence suivant présente le déroulement du processus de modification d'un ticket de maintenance. Après authentification, l'utilisateur sélectionne le ticket à modifier, le système affiche le formulaire pré-rempli avec les données actuelles, et l'utilisateur met à jour les informations souhaitées avant de valider les modifications.

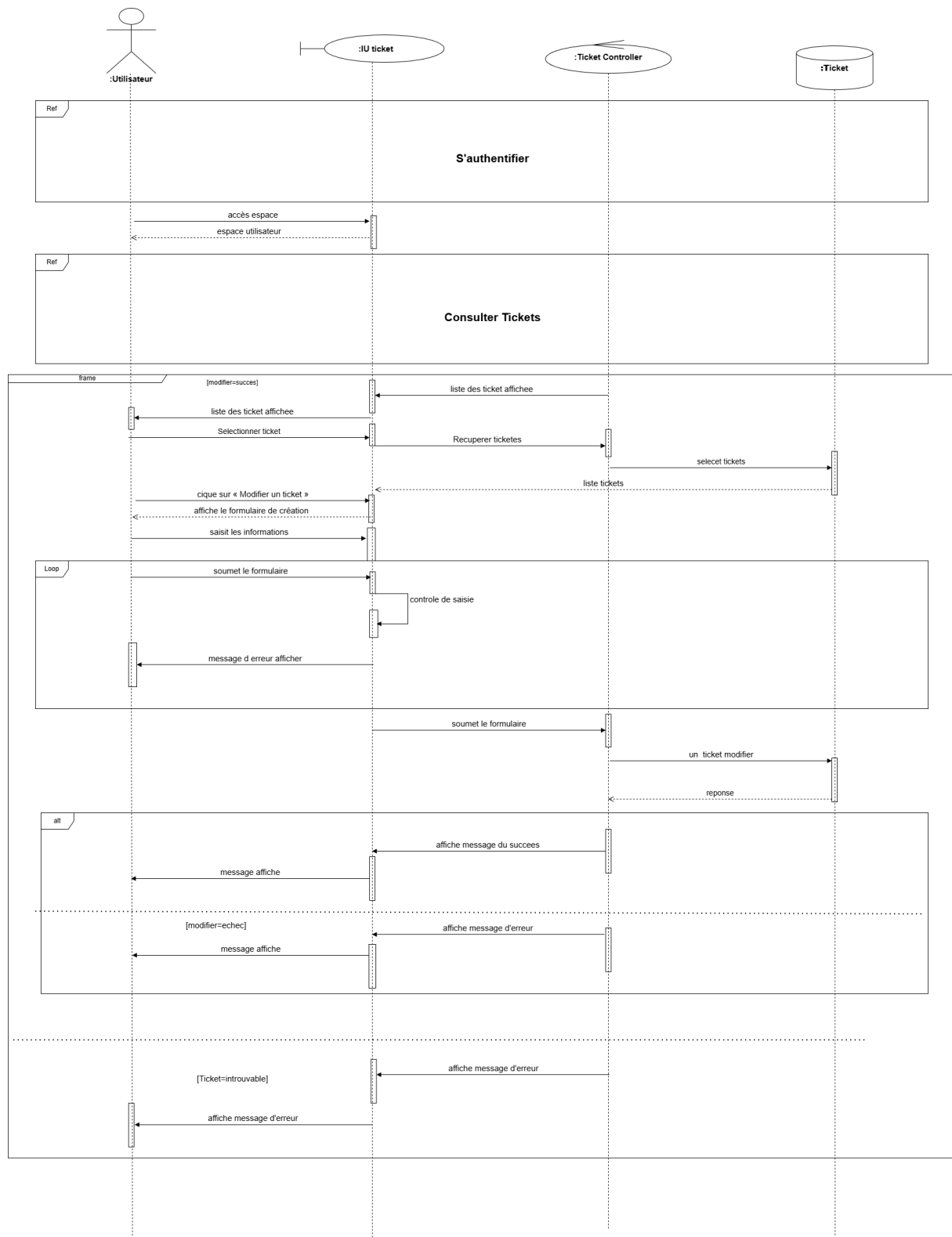


FIGURE 5.3 – Diagramme de séquence du scénario «Modifier ticket»

5.4.4 Diagramme de Classes du sprint 3

Le diagramme de classes présenté dans la figure met en évidence les principales entités du système ainsi que les relations qui les lient dans le cadre du Sprint 3.

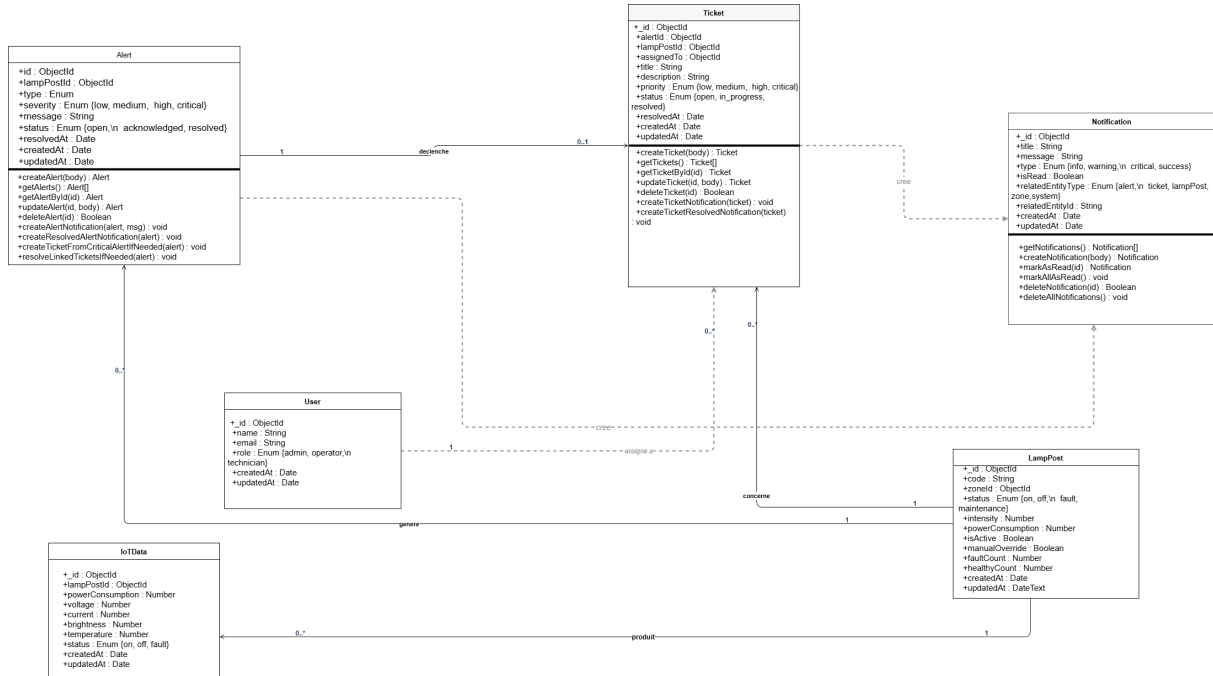


FIGURE 5.4 – Diagramme de Classes du sprint 3

5.5 Réalisation et tests

Cette section décrit le processus de mise en œuvre du système ainsi que les différents tests réalisés afin de valider son bon fonctionnement et sa conformité aux exigences spécifiées.

Interface Notifications

L'interface « Notifications » permet à l'utilisateur de consulter l'ensemble des notifications générées par le système et de supprimer celles qui ne sont plus nécessaires. Elle constitue le principal outil de suivi des événements critiques du réseau d'éclairage public.

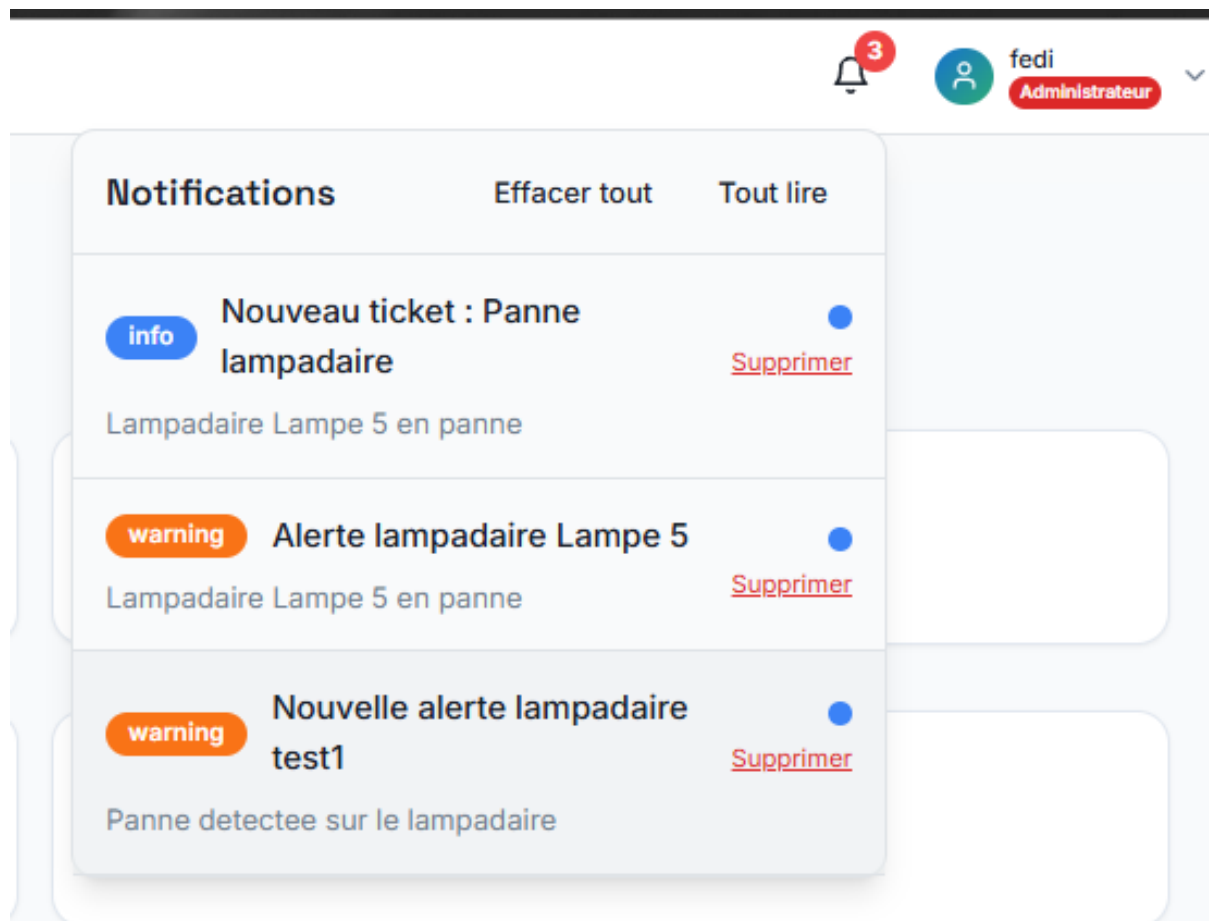


FIGURE 5.5 – Interface Notifications

Interface Alertes

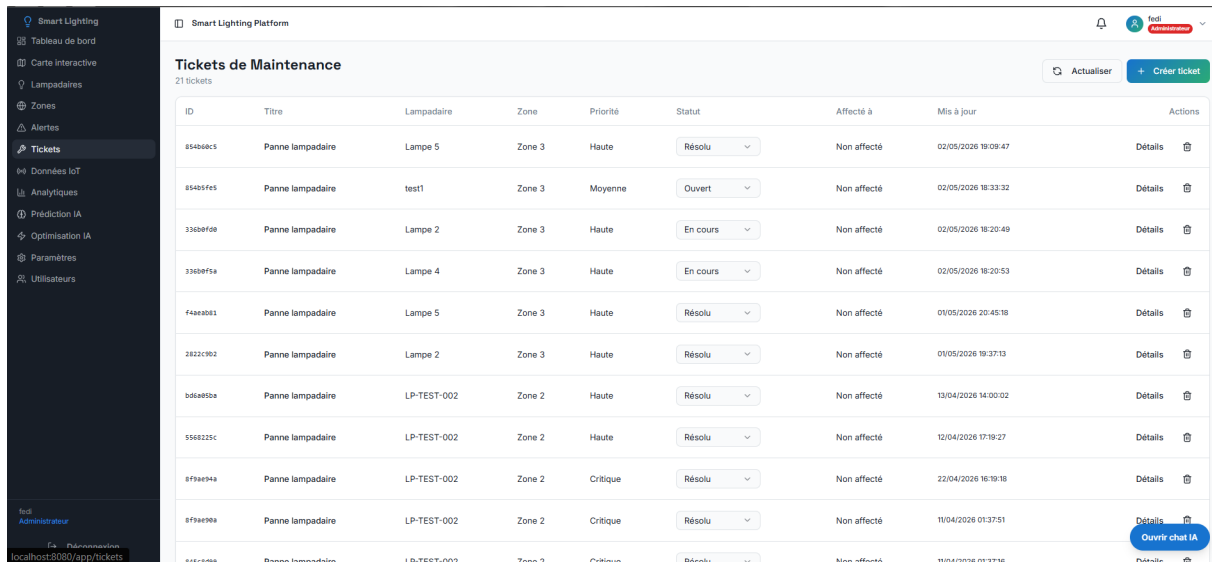
L'interface « Alertes » permet à l'utilisateur de consulter l'ensemble des alertes générées automatiquement par le système suite à la détection d'anomalies sur les lampadaires. Elle constitue le principal outil de supervision des incidents du réseau d'éclairage public.

Sévérité	Lampadaire	Zone	Type	Message	Date	État	Action
Critique	Lampe 5	Zone 3	fault	Lampadaire Lampe 5 en panne	02/05/2026 18:42:28	Résolu	Ticket
Warning	test1	Zone 3	fault	Panne detectee sur le lampadaire	02/05/2026 18:33:32	Active	Ticket
Critique	Lampe 2	Zone 3	fault	Lampadaire Lampe 2 en panne	01/05/2026 18:37:23	Acquiescé	Ticket
Critique	Lampe 4	Zone 3	fault	Lampadaire Lampe 4 en panne	01/05/2026 19:36:58	Acquiescé	Ticket
Critique	Lampe 5	Zone 3	fault	Lampadaire Lampe 5 en panne	30/04/2026 22:37:23	Résolu	Ticket
Critique	Lampe 2	Zone 3	fault	Lampadaire Lampe 2 en panne	27/04/2026 16:30:30	Résolu	Ticket
Critique	LP-TEST-002	Zone 2	fault	Lampadaire LP-TEST-002 en panne	13/04/2026 13:59:19	Résolu	Ticket
Critique	LP-TEST-002	Zone 2	fault	Lampadaire LP-TEST-002 en panne	12/04/2026 17:17:55	Résolu	Ticket
Critique	LP-TEST-002	Zone 2	fault	Lampadaire LP-TEST-002 en panne	11/04/2026 01:37:58	Résolu	Ticket
Critique	LP-TEST-002	Zone 2	fault	Lampadaire LP-TEST-002 en panne	11/04/2026 01:37:19	Résolu	Ticket

FIGURE 5.6 – Interface Alertes

Interface Tickets

L'interface « Tickets » permet à l'utilisateur de consulter, créer, modifier et supprimer les tickets de maintenance associés aux incidents détectés sur le réseau d'éclairage public.

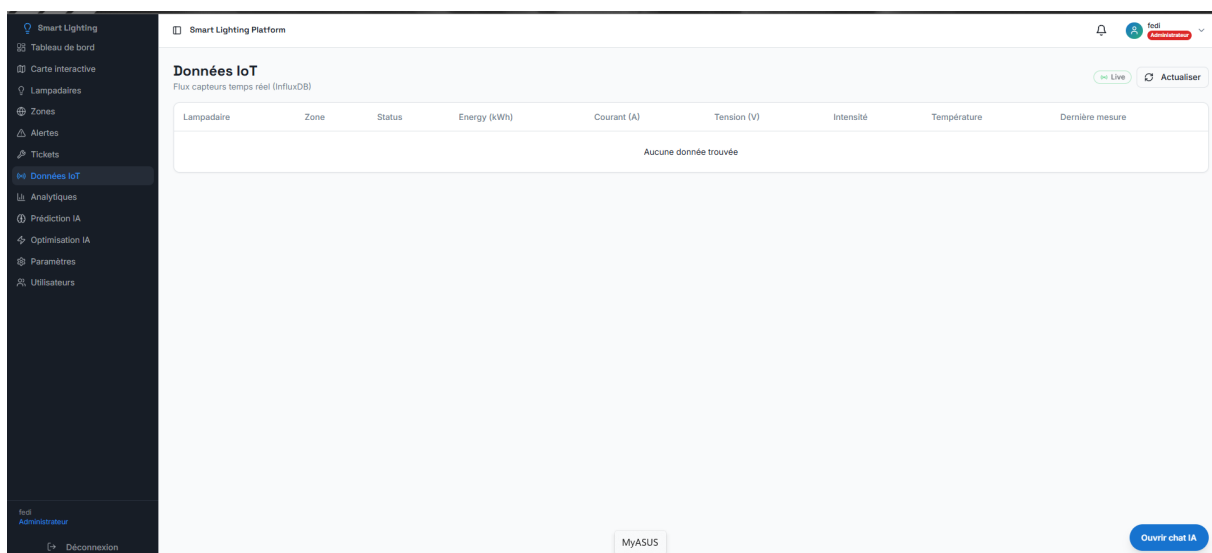


ID	Titre	Lampadaire	Zone	Priorité	Statut	Affecté à	Mis à jour	Actions
854b46c5	Panne lampadaire	Lampe 5	Zone 3	Haute	Résolu	Non affecté	02/05/2026 19:09:47	Détails
854b5f65	Panne lampadaire	test1	Zone 3	Moyenne	Ouvert	Non affecté	02/05/2026 18:33:32	Détails
336b9f60	Panne lampadaire	Lampe 2	Zone 3	Haute	En cours	Non affecté	02/05/2026 18:20:49	Détails
336b9f5a	Panne lampadaire	Lampe 4	Zone 3	Haute	En cours	Non affecté	02/05/2026 18:20:53	Détails
f44eb0b1	Panne lampadaire	Lampe 5	Zone 3	Haute	Résolu	Non affecté	01/05/2026 20:45:18	Détails
2b22c102	Panne lampadaire	Lampe 2	Zone 3	Haute	Résolu	Non affecté	01/05/2026 19:37:13	Détails
b06eb5ba	Panne lampadaire	LP-TEST-002	Zone 2	Haute	Résolu	Non affecté	13/04/2026 14:00:02	Détails
556b225c	Panne lampadaire	LP-TEST-002	Zone 2	Haute	Résolu	Non affecté	12/04/2026 17:19:27	Détails
8f94e94a	Panne lampadaire	LP-TEST-002	Zone 2	Critique	Résolu	Non affecté	22/04/2026 16:19:18	Détails
8f94e90a	Panne lampadaire	LP-TEST-002	Zone 2	Critique	Résolu	Non affecté	11/04/2026 01:37:51	Détails
845c5979	Panne lampadaire	LP-TEST-002	Zone 2	Critique	Résolu	Non affecté	11/04/2026 01:37:16	Détails

FIGURE 5.7 – Interface Tickets

Interface Données IoT

L'interface « Données IoT » permet à l'utilisateur de visualiser en temps réel les métriques collectées par les capteurs des lampadaires, telles que la consommation énergétique, la tension, le courant et la luminosité.



Lampadaire	Zone	Status	Energy (kWh)	Courant (A)	Tension (V)	Intensité	Température	Dernière mesure
Aucune donnée trouvée								

FIGURE 5.8 – Interface Données IoT

5.5.1 Tests

Le tableau suivant présente les tests effectués pour le Sprint 3.

Cas de test	Démarche	Comportement attendu	Résultat
Test de consultation des alertes	Accéder à la page « Alertes », consulter la liste des alertes générées par le système et cliquer sur une alerte pour afficher ses détails.	Le système affiche la liste des alertes avec leur type, niveau de criticité, lampadaire concerné et statut de traitement.	Conforme
Test d'acquiescement d'une alerte	Sélectionner une alerte ouverte dans la liste, puis cliquer sur le bouton « Acquiescer » pour signaler la prise en charge.	Le système met à jour le statut de l'alerte à « acknowledged » et actualise la liste des alertes.	Conforme
Test de résolution d'une alerte	Sélectionner une alerte acquiescée dans la liste, puis cliquer sur le bouton « Résoudre » pour clôturer le traitement.	Le système met à jour le statut de l'alerte à « resolved », enregistre la date de résolution et actualise la liste.	Conforme
Test de création d'un ticket	Accéder à la page « Tickets », cliquer sur « Créer un ticket », remplir le formulaire avec le titre, la description, le lampadaire et la priorité, puis soumettre.	Le système valide les données, crée le ticket dans la base de données et affiche un message de confirmation.	Conforme
Test de modification d'un ticket	Accéder à la liste des tickets, sélectionner un ticket existant, modifier son statut ou sa priorité, puis valider les modifications.	Le système enregistre les modifications, met à jour le statut du lampadaire associé et affiche un message de confirmation.	Conforme
Test de suppression d'un ticket	Accéder à la liste des tickets, sélectionner un ticket, cliquer sur « Supprimer », puis confirmer la suppression dans la boîte de dialogue.	Le système supprime le ticket de la base de données et actualise la liste des tickets.	Conforme

Test de consultation des données IoT	Accéder à la page « Données IoT », sélectionner un lampadaire et une période, puis consulter les métriques affichées.	Le système affiche les métriques IoT en temps réel (consommation, tension, intensité, température) collectées depuis InfluxDB.	Conforme
Test de consultation des notifications	Accéder à la page « Notifications » et consulter la liste des notifications générées par le système.	Le système affiche la liste des notifications avec leur type, message, date et statut de lecture.	Conforme
Test de suppression des notifications	Sélectionner une notification dans la liste, cliquer sur « Supprimer », puis confirmer la suppression.	Le système supprime la notification de la base de données et actualise la liste des notifications.	Conforme

TABLE 5.4 – Tests du Sprint 3

5.6 Utilisation des outils de suivi Scrum

Dans cette partie nous allons présenter les outils de suivi de la méthodologie adoptée Scrum : le Scrum board et le Burn down chart.

5.6.1 Scrum Board

À la fin du Sprint 3, l'état d'avancement du projet est illustré dans la figure ci-dessous. Trois cas d'utilisation ont déjà été terminés, notamment : consulter les notifications, consulter les données IoT et consulter les alertes. Trois cas d'utilisation sont en cours de réalisation, parmi lesquels : gérer les tickets, gérer le statut des tickets et consulter les détails des tickets. Enfin, deux cas d'utilisation restent à accomplir : créer un ticket et supprimer un ticket, afin d'atteindre l'ensemble des objectifs fixés pour ce sprint.

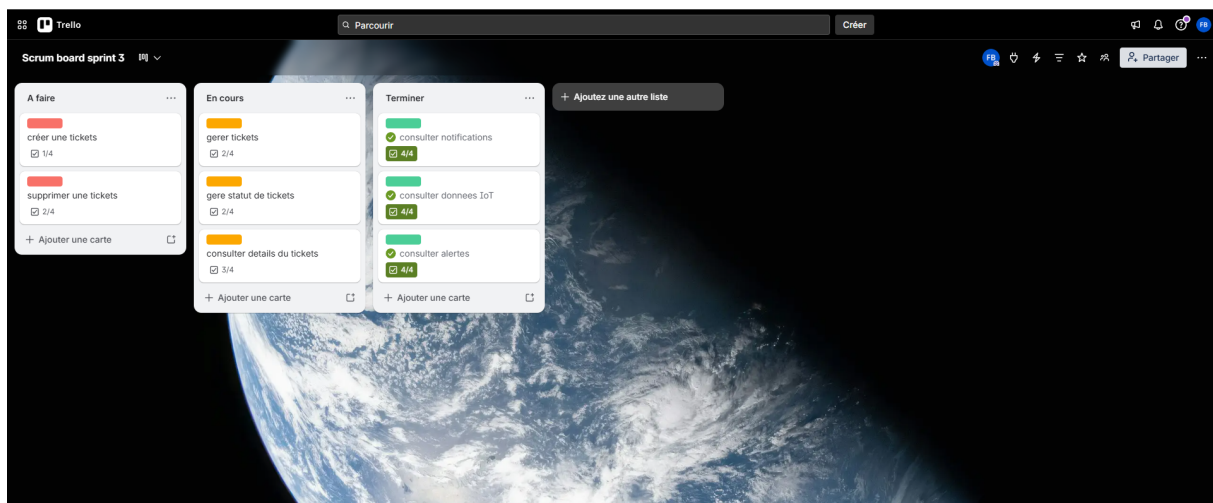


FIGURE 5.9 – Tableau Scrum Sprint 3

5.6.2 Burn Down Chart

La FIGURE 66 ci-dessous présente l'évolution du travail durant le troisième sprint. Au cours de la première semaine, la progression a été lente en raison des difficultés rencontrées lors du développement de la fonctionnalité de génération des Tickets. Toutefois, nous avons su nous adapter par la suite

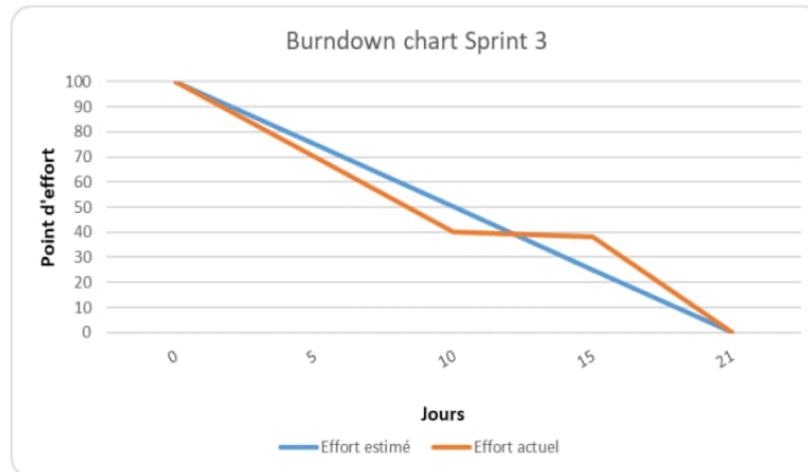


Figure 66 : Burn down à la fin de 3ème Sprint

FIGURE 5.10 – Burn down à la fin de 3ème Sprint

5.7 Conclusion

Ce sprint a permis de concevoir et d'implémenter les modules de supervision opérationnelle de la plateforme Smart Lighting, notamment la gestion des alertes, la gestion des tickets de maintenance, la consultation des données IoT en temps réel ainsi que le système de notifications. L'ensemble de ces fonctionnalités dote la plateforme des outils nécessaires à une supervision proactive et efficace du réseau d'éclairage public, et constitue le socle sur lequel s'appuiera le Sprint 4 pour l'intégration de l'intelligence artificielle.

Chapitre 6

Sprint 4 : Étude, analyse et réalisation du Sprint 4

Plan

1. Introduction
2. Backlog du Sprint
3. Spécification des besoins du Sprint 3
4. Conception
5. Réalisation et tests
6. Utilisation des outils de suivi Scrum
7. Conclusion

6.1 Introduction

Le Sprint 4 constitue l'aboutissement du développement de la plateforme Smart Lighting en fusionnant l'intelligence artificielle et la business intelligence. Il intègre un microservice FastAPI dédié à la prédiction de la consommation énergétique et à l'optimisation intelligente des lampadaires, ainsi qu'un tableau de bord avancé offrant une visualisation dynamique des indicateurs clés de performance et une analyse décisionnelle du réseau d'éclairage public.

6.2 Backlog du Sprint

Le tableau suivant présente l'ensemble des User Stories planifiées et réalisées dans le cadre du Sprint 4.

ID US	User Story	ID tâche	Tâche	Statut	Complexité
1	En tant qu'utilisateur, je veux consulter le tableau de bord afin de visualiser les indicateurs clés du réseau en temps réel.	1.1	Conception du tableau de bord	Terminé	Faible
		1.2	Développement API récupération des KPI	Terminé	Élevé
		1.3	Développement interface tableau de bord	Terminé	Élevé
		1.4	Test du module tableau de bord	Terminé	Faible
2	En tant qu'administrateur / décideur / opérateur, je veux consulter les analytiques afin d'analyser les données du réseau d'éclairage public.	2.1	Conception des tableaux analytiques	Terminé	Faible
		2.2	Développement FACT Énergie	Terminé	Élevé
		2.3	Développement FACT Pannes	Terminé	Élevé

		2.4	Développement FACT Alertes	Terminé	Élevé
		2.5	Développement FACT IoT Live	Terminé	Élevé
		2.6	Test du module ana- lytiques	Terminé	Faible
3	En tant qu'admin- istrateur / déci- deur / opérateur, je veux comparer les zones afin d'identi- fier les secteurs les plus critiques du ré- seau.	3.1	Conception du mo- dule comparaison zones	Terminé	Faible
		3.2	Développement API comparaison par zone	Terminé	Moyen
		3.3	Développement in- terface comparaison zones	Terminé	Moyen
		3.4	Test du module com- paraison zones	Terminé	Faible
4	En tant qu'admin- istrateur / déci- deur / opérateur, je veux consulter la prédiction IA afin d'anticiper la consommation énergétique du réseau.	4.1	Conception du mi- croservice FastAPI	Terminé	Faible
		4.2	Développement end- point /predict	Terminé	Élevé

		4.3	Intégration backend Node.js vers FastAPI	Terminé	Élevé
		4.4	Développement interface prédiction IA	Terminé	Moyen
		4.5	Test du module prédiction IA	Terminé	Faible
5	En tant qu'administrateur / décideur / opérateur, je veux consulter l'optimisation IA afin d'améliorer l'efficacité énergétique du réseau.	5.1	Développement endpoint /optimize	Terminé	Élevé
		5.2	Intégration recommandations dans le backend	Terminé	Élevé
		5.3	Développement interface optimisation IA	Terminé	Moyen
		5.4	Test du module optimisation IA	Terminé	Faible
6	En tant que citoyen / technicien / opérateur / décideur / administrateur, je veux communiquer avec le chatbot afin d'obtenir des informations sur le réseau.	6.1	Conception du module chatbot IA	Terminé	Faible
		6.2	Développement endpoint /chat	Terminé	Élevé

		6.3	Intégration chatbot dans le frontend	Terminé	Moyen
		6.4	Test du module chatbot	Terminé	Faible
7	En tant qu'administrateur, je veux gérer les paramètres afin de configurer les seuils et les règles du système.	7.1	Conception du module paramètres	Terminé	Faible
		7.2	Développement API gestion paramètres	Terminé	Moyen
		7.3	Développement interface paramètres	Terminé	Moyen
		7.4	Test du module paramètres	Terminé	Faible
8	En tant qu'utilisateur, je veux me déconnecter afin de mettre fin à ma session de manière sécurisée.	8.1	Développement API déconnexion	Terminé	Faible
		8.2	Développement interface déconnexion	Terminé	Faible
		8.3	Test de la déconnexion	Terminé	Faible

TABLE 6.2 – Backlog détaillé du Sprint 4

6.2.1 Diagramme de cas d'utilisation

Dans le but de mieux illustrer les fonctionnalités développées lors de ce sprint, le diagramme de cas d'utilisation est présenté dans la figure ci-après.

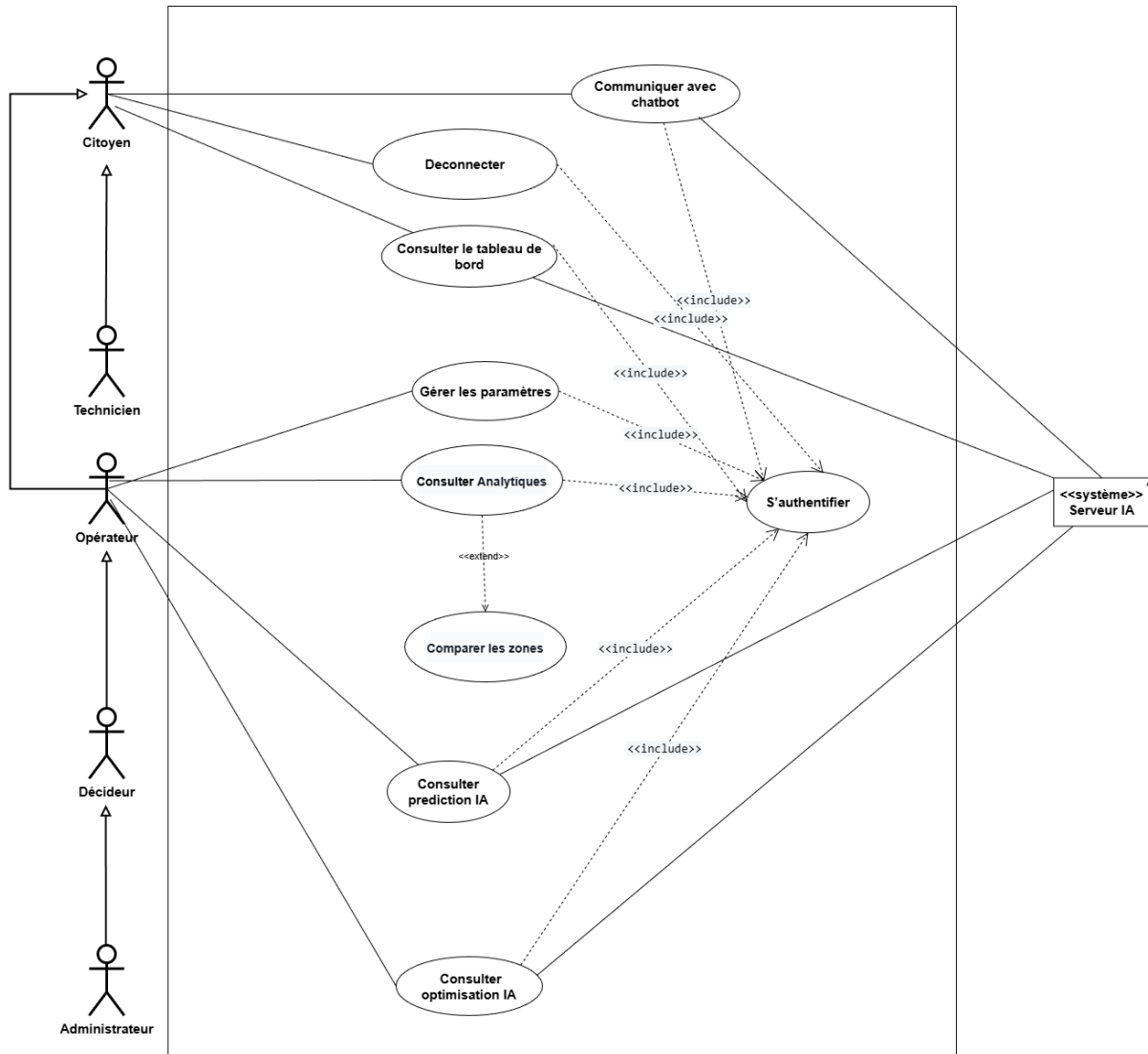


FIGURE 6.1 – Diagramme de cas d'utilisation du Sprint 4

Description du cas d'utilisation « Consulter le tableau de bord »

Le tableau suivant présente la description du cas d'utilisation « Consulter le tableau de bord ».

Cas d'utilisation	Consulter le tableau de bord
Acteurs	Administrateur, Décideur, Opérateur, Technicien, Citoyen
Description brève du CU	Permet à l'acteur de visualiser une vue centralisée et synthétique de l'ensemble des indicateurs clés de performance du réseau d'éclairage public, incluant les données IoT, les prédictions IA et les statistiques globales.
Préconditions	L'utilisateur est authentifié. Des données IoT et des résultats IA sont disponibles dans le système.
Post conditions	L'utilisateur visualise le tableau de bord avec l'ensemble des KPI, graphiques et indicateurs du réseau mis à jour.
Scénario de base	1. L'utilisateur s'authentifie. 2. L'utilisateur accède à la page du tableau de bord. 3. Le système envoie des requêtes au backend pour récupérer les données globales du réseau. 4. Le système récupère les données IoT, les alertes, les tickets et les résultats IA depuis MongoDB. 5. Le système affiche les KPI principaux (consommation totale, lampadaires actifs, alertes, taux d'optimisation). 6. Le système affiche les graphiques dynamiques d'évolution de la consommation énergétique. 7. L'utilisateur interagit avec les graphiques (filtrage par zone, par période). 8. Le système met à jour les visualisations en fonction des filtres appliqués.

Autre scénario	1. Si aucune donnée n'est disponible → le système affiche un message informatif et un tableau de bord vide. 2. Si le chargement des données échoue → le système affiche un message d'erreur et propose de réessayer. 3. Si le microservice IA est indisponible → le tableau de bord s'affiche sans les données de prédiction.
-----------------------	---

TABLE 6.3 – Description du cas d'utilisation « Consulter le tableau de bord »

Description du cas d'utilisation « Communiquer avec le chatbot »

Le tableau suivant présente la description du cas d'utilisation « Communique avec le chatbot ».

Cas d'utilisation	Communiquer avec le chatbot
Acteurs	Administrateur, Décideur, Opérateur, Technicien, Visiteur
Description brève du CU	Permet à l'acteur d'interagir avec le chatbot intelligent de la plateforme afin d'obtenir des informations, des recommandations et une assistance sur le réseau d'éclairage public.
Préconditions	L'utilisateur accède à la plateforme. Le microservice IA est opérationnel.
Post conditions	L'utilisateur reçoit une réponse pertinente du chatbot et peut poursuivre la conversation ou consulter l'historique.
Scénario de base	1. L'utilisateur s'authentifie. 2. L'utilisateur accède à l'interface du chatbot. 3. Le système affiche l'interface de conversation avec l'historique des échanges précédents. 4. L'utilisateur saisit sa question ou sa demande. 5. Le système transmet la requête au microservice IA via l'endpoint <code>/chat</code>. 6. Le microservice IA traite la requête et génère une réponse contextuelle. 7. Le système affiche la réponse du chatbot dans l'interface de conversation. 8. L'utilisateur peut poursuivre la conversation ou consulter l'historique des échanges.
Autre scénario	1. Si le microservice IA est indisponible → le système affiche un message d'erreur et invite l'utilisateur à réessayer. 2. Si la requête est vide → le système bloque l'envoi et notifie l'utilisateur. 3. Si la réponse du chatbot dépasse le délai imparti → le système affiche un message de timeout.

TABLE 6.4 – Description du cas d'utilisation « Communiquer avec le chatbot »

6.2.2 Diagramme de séquence du scénario «Consulter le tableau du bord »

Le diagramme de séquence suivant présente le déroulement du processus de consultation du tableau de bord. Après authentification, l'utilisateur accède à une vue centralisée des indicateurs clés de performance du réseau d'éclairage public, regroupant les données IoT, les statistiques globales et les résultats des prédictions IA, avec la possibilité de filtrer les visualisations par zone ou par période.

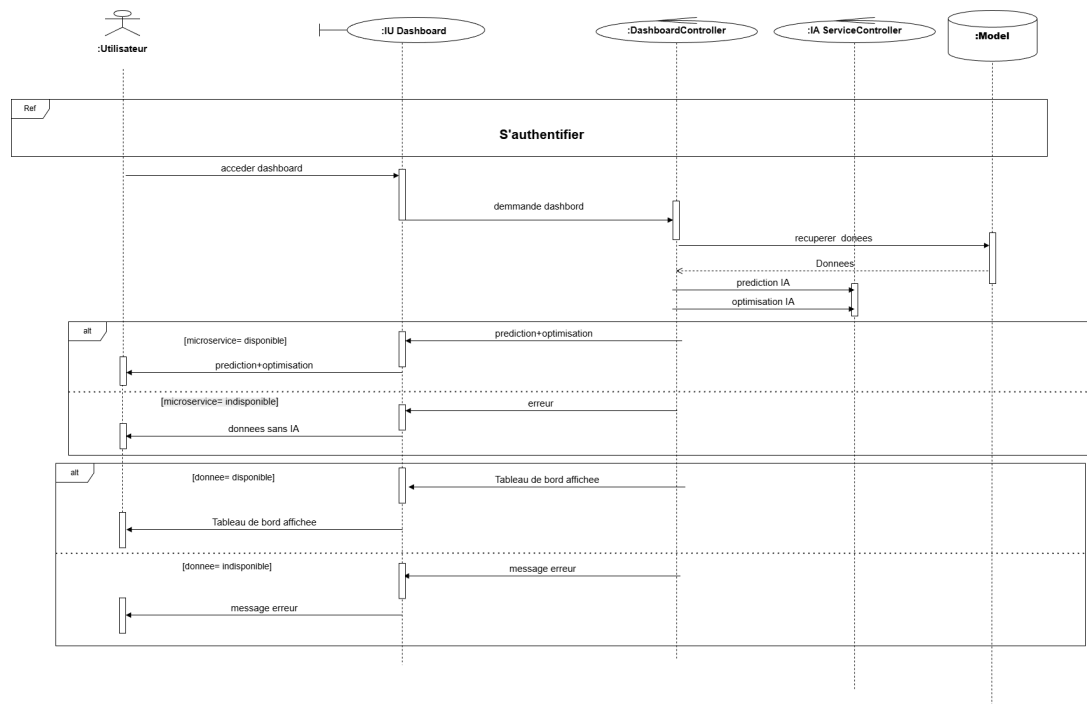


FIGURE 6.2 – Diagramme de séquence du scénario «Consulter le tableau du bord»

6.2.3 Diagramme de séquence du scénario «Communiquer avec le chatbot »

Le diagramme de séquence suivant présente le déroulement du processus d'interaction avec le chatbot. L'utilisateur saisit sa question dans l'interface de conversation, le système transmet la requête au microservice IA via l'endpoint `/chat`, qui génère une réponse contextuelle retournée et affichée dans l'interface.

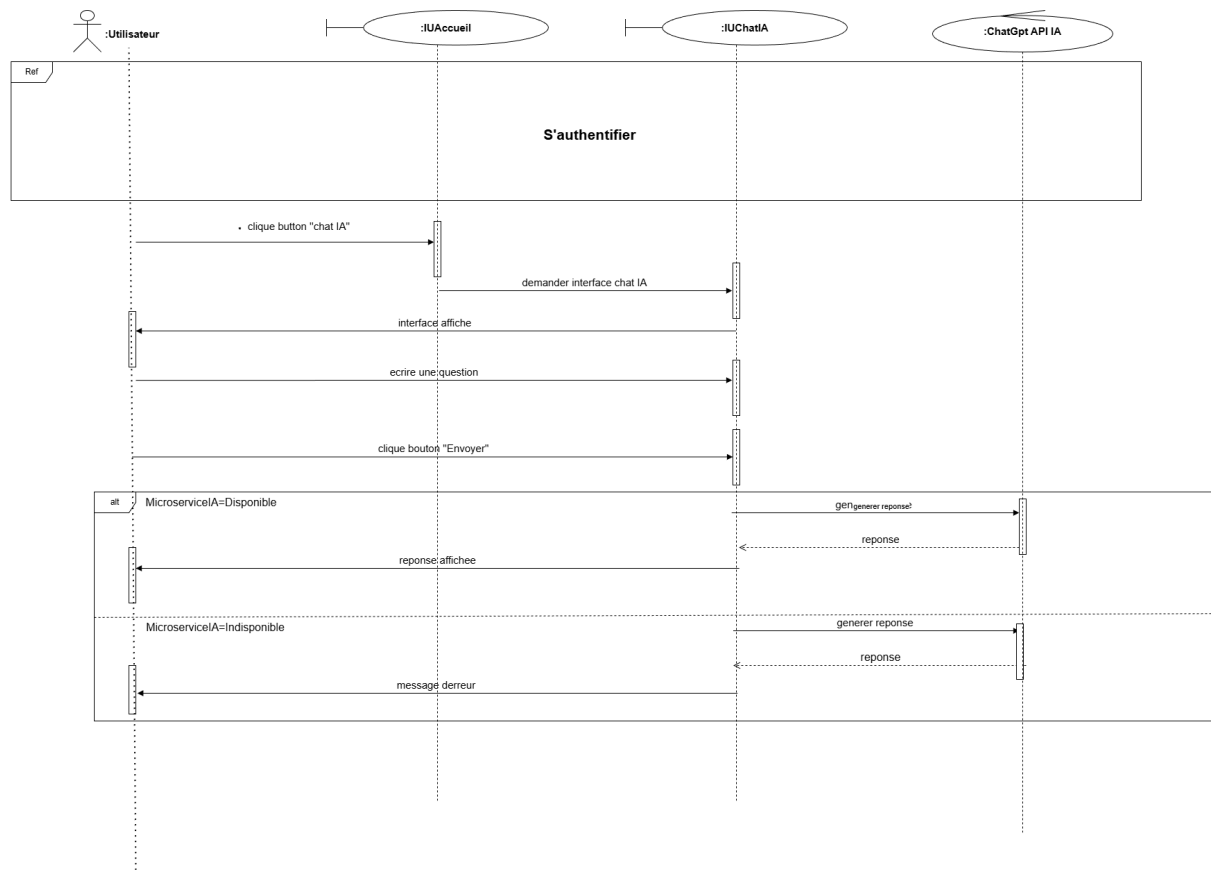


FIGURE 6.3 – Diagramme de séquence du scénario «Diagramme de séquence Interagir avec le chatbot»

6.2.4 Diagramme de Classes du sprint 4

Le diagramme de classes présenté dans la figure met en évidence les principales entités du système ainsi que les relations qui les lient dans le cadre du Sprint 4.

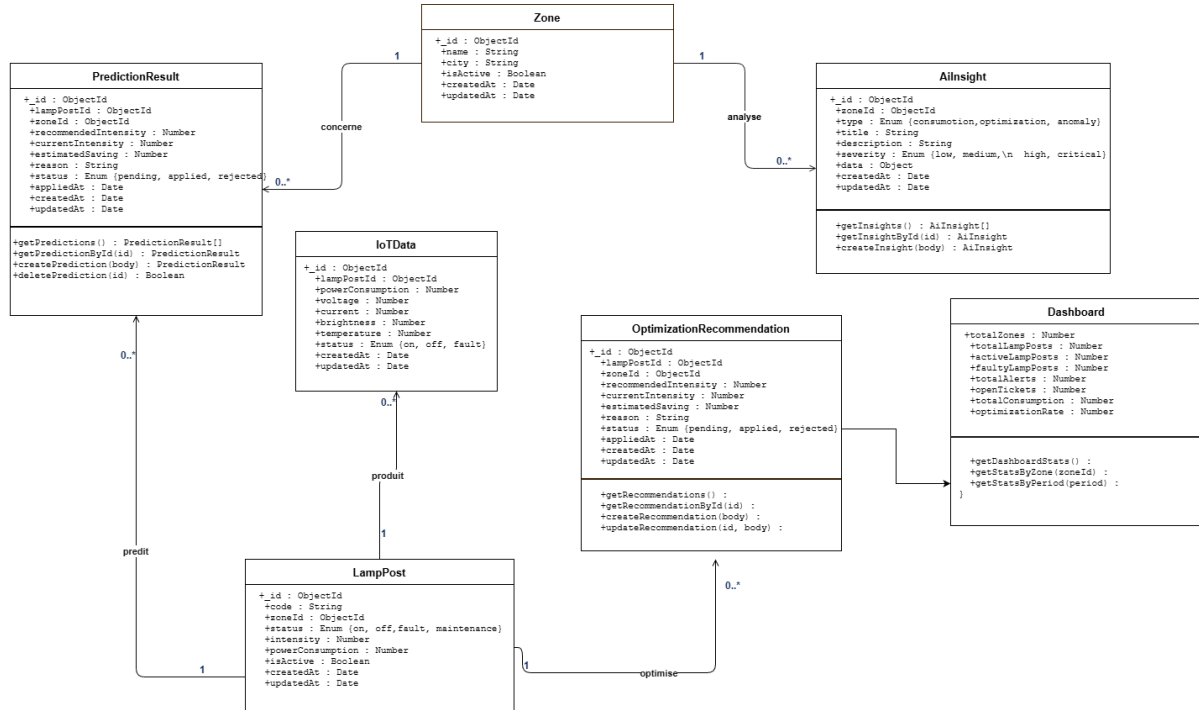


FIGURE 6.4 – Diagramme de Classes du sprint 4

6.3 Spécification décisionnelle

Cette section comprend la définition des objectifs décisionnels, la construction du tableau de bord ainsi que la mise en place de la chaîne BI, afin de permettre une analyse pertinente des données du réseau d'éclairage public et un pilotage efficace de sa consommation énergétique. La démarche adoptée s'inspire de la méthode GIMSI, qui propose une approche structurée pour la mise en oeuvre de projets décisionnels en se basant sur l'identification des besoins métier, la modélisation des indicateurs de performance, et la mise en place d'outils de pilotage adaptés au contexte de la gestion intelligente de l'éclairage public.

6.3.1 Définition des objectifs

Le tableau suivant présente les objectifs décisionnels de chaque acteur de la plateforme Smart Lighting.

Acteur	Objectifs
Administrateur	- Connaître le nombre total de zones et de lampadaires. - Connaître le nombre de lampadaires actifs, en panne et en maintenance. - Connaître la consommation énergétique totale du réseau. - Connaître le nombre total d'alertes ouvertes et résolues. - Connaître le nombre de tickets de maintenance en cours. - Connaître les prédictions de consommation énergétique générées par l'IA. - Connaître les recommandations d'optimisation par zone.
Décideur	- Connaître les indicateurs clés de performance du réseau. - Connaître la consommation énergétique par zone géographique. - Connaître l'évolution temporelle de la consommation énergétique. - Connaître le taux d'optimisation du réseau. - Connaître les zones à forte consommation ou problématiques. - Connaître les prédictions futures de consommation pour anticiper les besoins.
Opérateur	- Connaître l'état en temps réel des lampadaires du réseau. - Connaître le nombre d'alertes actives par zone. - Connaître les recommandations d'optimisation des lampadaires. - Connaître l'historique des interventions de maintenance. - Connaître la consommation énergétique par lampadaire.
Technicien	- Connaître les lampadaires en panne ou en maintenance. - Connaître les tickets de maintenance qui lui sont assignés. - Connaître l'historique des pannes par lampadaire.

6.3.2 Construction du Tableau de bord

Le tableau suivant présente les indicateurs retenus, le type de visualisation choisi et la maquette correspondante pour chaque composant du tableau de bord de la plateforme Smart Lighting.

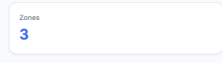
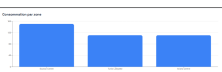
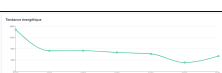
Indicateur	Choix du graphique	Maquette
Nombre total de zones	Carte de KPI	
Nombre total de lampadaires	Carte de KPI	
Nombre de pannes	Carte de KPI	
Nombre d'alertes actives	Carte de KPI	
Tickets ouverts	Carte de KPI	
Energie consommée (24h)	Carte de KPI	
Consommation par zone	Diagramme en barres verticales	
Tendance énergétique	Courbe temporelle	

TABLE 6.6 – Construction du Tableau de bord

6.3.3 Chaîne BI

Dans cette partie, nous mettons en oeuvre la chaîne décisionnelle de type ETL (Extract – Transform – Load), adaptée au contexte d’un tableau de bord interactif développé avec React et Recharts. Bien que ces outils soient orientés front-end, la logique BI reste indispensable pour garantir une visualisation fiable et pertinente des données du réseau d’éclairage public.

Extraction des données (Extract)

L’extraction constitue la première étape de la chaîne BI. Elle a pour but de collecter les données nécessaires à partir des sources de données de la plateforme Smart Lighting, à savoir MongoDB pour les données structurelles (zones, lampadaires, alertes, tickets) et InfluxDB pour les métriques IoT temporelles (consommation, tension, courant, intensité, température). Cette étape est réalisée grâce à des requêtes exécutées côté serveur avec Node.js, permettant de récupérer les informations pertinentes depuis les différentes collections. Ces données sont ensuite transmises à l’application front-end via des API REST conçues pour répondre aux besoins spécifiques de chaque indicateur du tableau de bord.

Transformation des données (Transform)

Après leur extraction, les données brutes nécessitent des traitements afin de pouvoir être utilisées dans les visualisations. Dans le cadre de la plateforme Smart Lighting, ces transformations sont réalisées en deux temps : d’abord côté serveur avec Node.js, puis côté client dans l’application React. Les opérations réalisées comprennent le nettoyage des données IoT (suppression des valeurs aberrantes, correction des formats temporels), le filtrage (par zone géographique, par période, par statut de lampadaire), ainsi que l’agrégation (calculs de consommation totale, regroupements par zone, calcul du taux de disponibilité et du taux de résolution des alertes). Le microservice FastAPI intervient également dans cette étape en transformant les données IoT historiques en prédictions de consommation et en recommandations d’optimisation.

Chargement des données (Load)

Une fois les données transformées, elles sont prêtes à être exploitées dans l'interface utilisateur. Dans la plateforme Smart Lighting, le chargement consiste à transmettre les données depuis le serveur Node.js vers le front-end développé avec React. Ces données sont ensuite intégrées dans les composants graphiques grâce à la bibliothèque Recharts, utilisée pour générer les visualisations dynamiques (cartes de KPI, courbes temporelles, diagrammes en barres, donut charts). Elles sont gérées dans l'état local des composants via les hooks `useState` et `useEffect`, garantissant un affichage dynamique, réactif et actualisé des indicateurs clés du tableau de bord. Ce processus permet aux acteurs de la plateforme une lecture claire, instantanée et décisionnelle de l'état du réseau d'éclairage public.

6.3.4 Data Mart

Un Data Mart est un sous-ensemble d'un entrepôt de données orienté vers un domaine métier spécifique. Dans le cadre de la plateforme Smart Lighting, le Data Mart est centré sur l'analyse de la consommation énergétique et la supervision du réseau d'éclairage public. Il adopte un **schéma en étoile** composé d'une table de faits centrale `FAIT_CONSOMMATION` reliée à cinq tables de dimensions.

Table de faits

La table de faits `FAIT_CONSOMMATION` centralise les mesures quantitatives collectées par les capteurs IoT des lampadaires. Elle constitue le noyau du schéma en étoile et contient les indicateurs de performance clés du réseau.

Attribut	Type	Description
fact_id	ObjectId (PK)	Identifiant unique du fait
time_id	ObjectId (FK)	Référence vers DIM_TEMPS
lamp_id	ObjectId (FK)	Référence vers DIM_LAMPE
zone_id	ObjectId (FK)	Référence vers DIM_ZONE
user_id	ObjectId (FK)	Référence vers DIM_USER
alert_id	ObjectId (FK)	Référence vers DIMALERTE
pred_id	ObjectId (FK)	Référence vers DIM_PREDICTION

consommation_kwh	Number	Consommation énergétique en kWh
tension_v	Number	Tension électrique en Volts
courant_a	Number	Courant électrique en Ampères
luminosite	Number	Intensité lumineuse en %
temperature	Number	Température du capteur en °C
nb_alertes	Number	Nombre d'alertes générées

TABLE 6.7 – Table de faits – FAIT_CONSOMMATION

Tables de dimensions

DIM_TEMPS

Attribut	Type	Description
time_id	ObjectId (PK)	Identifiant unique de la date
date	Date	Date complète de la mesure
heure	Number	Heure de la mesure (0-23)
jour_semaine	String	Jour de la semaine
mois	Number	Mois de l'année (1-12)
trimestre	Number	Trimestre de l'année (1-4)
annee	Number	Année de la mesure
est_weekend	Boolean	Indique si la mesure est un weekend
est_nuit	Boolean	Indique si la mesure est de nuit

TABLE 6.8 – Table de dimension – DIM_TEMPS

DIM_LAMPE

Attribut	Type	Description
----------	------	-------------

lamp_id	ObjectId (PK)	Identifiant unique du lampadaire
zone_id	ObjectId (FK)	Référence vers DIM_ZONE
code	String	Code d'identification du lampadaire
statut	String	Statut du lampadaire
intensite	Number	Intensité lumineuse configurée
latitude	Number	Latitude GPS du lampadaire
longitude	Number	Longitude GPS du lampadaire
fault_count	Number	Nombre de pannes enregistrées
manual_override	Boolean	Mode de contrôle manuel

TABLE 6.9 – Table de dimension – DIM_LAMPE

DIM_ZONE

Attribut	Type	Description
zone_id	ObjectId (PK)	Identifiant unique de la zone
nom	String	Nom de la zone géographique
ville	String	Ville d'appartenance
latitude	Number	Latitude du centre de la zone
longitude	Number	Longitude du centre de la zone
nb_lampes	Number	Nombre de lampadaires dans la zone
consommation_moy	Number	Consommation moyenne de la zone
est_active	Boolean	Etat d'activation de la zone

TABLE 6.10 – Table de dimension – DIM_ZONE

DIM_USER

Attribut	Type	Description
user_id	ObjectId (PK)	Identifiant unique de l'utilisateur
nom	String	Nom complet de l'utilisateur
email	String	Adresse email de l'utilisateur
role	String	Rôle de l'utilisateur dans le système
two_factor_enabled	Boolean	Activation de la double authentification
created_at	Date	Date de création du compte

TABLE 6.11 – Table de dimension – DIM_USER

DIMALERTE

Attribut	Type	Description
alert_id	ObjectId (PK)	Identifiant unique de l'alerte
lamp_id	ObjectId (FK)	Référence vers DIM_LAMPE
type_alerte	String	Type de l'alerte détectée
severite	String	Niveau de criticité de l'alerte
statut	String	Statut de traitement de l'alerte
resolved_at	Date	Date de résolution de l'alerte
created_at	Date	Date de création de l'alerte

TABLE 6.12 – Table de dimension – DIM_ALERTERTE

DIM_PREDICTION

Attribut	Type	Description
pred_id	ObjectId (PK)	Identifiant unique de la prédiction
type_prediction	String	Type de prédiction générée
strategie_ia	String	Stratégie IA utilisée
intensite_recommandee	Number	Intensité lumineuse recommandée
niveau_risque	String	Niveau de risque de panne estimé
confiance	Number	Indice de confiance du modèle IA
source	String	Source du modèle de prédiction
created_at	Date	Date de génération de la prédiction

TABLE 6.13 – Table de dimension – DIM_PREDICTION

Schéma en étoile

La figure 6.5 illustre le schéma en étoile du Data Mart de la plateforme Smart Lighting. La table de faits FAIT_CONSOMMATION est reliée aux cinq tables de dimensions DIM_TEMPS, DIM_LAMPE, DIM_ZONE, DIM_USER, DIMALERTE et DIM_PREDICTION via des clés étrangères, permettant ainsi une analyse multidimensionnelle des données du réseau d'éclairage public.

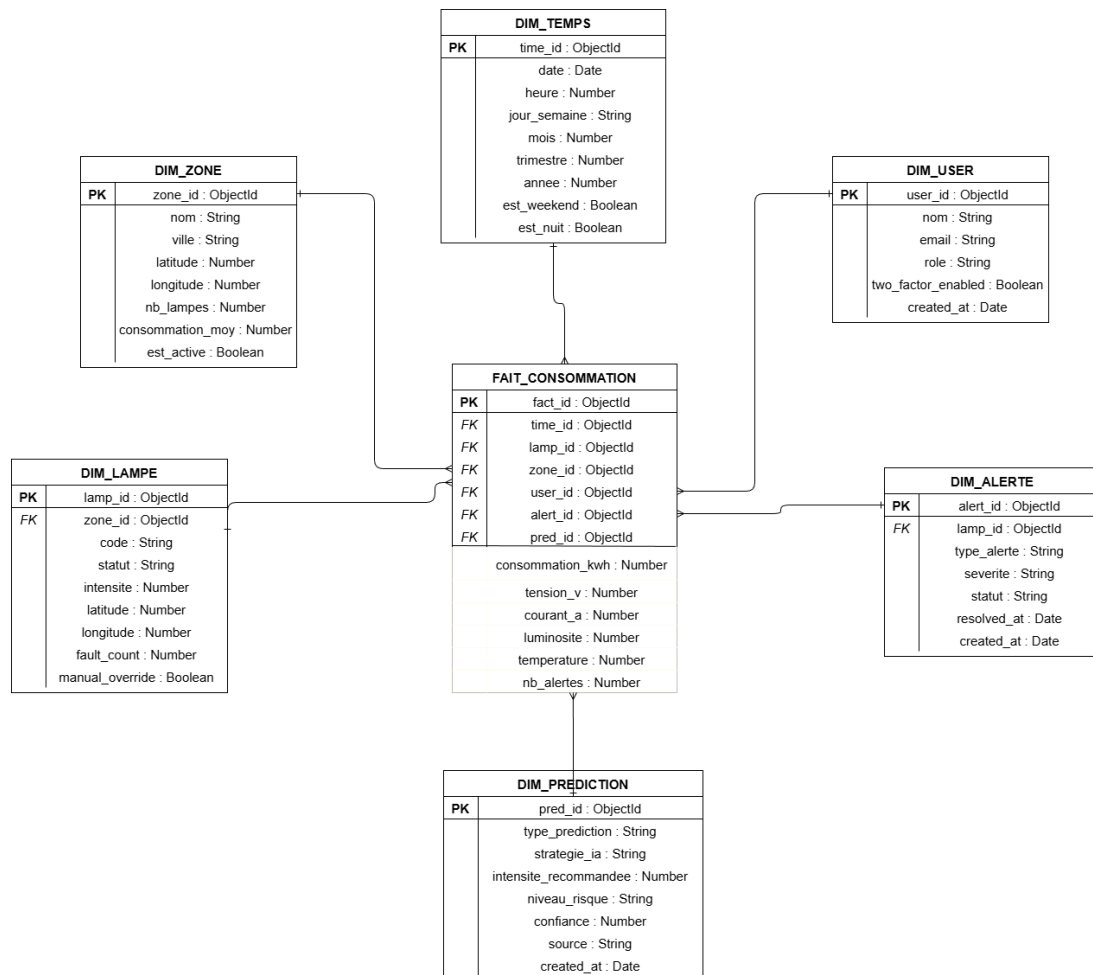


FIGURE 6.5 – Schéma en étoile – Data Mart Smart Lighting

Indicateur	Extract	Transform	Load
Nombre total de zones	Extraction depuis la collection Zone via API Node.js	Filtrage des zones actives, suppression des doublons	Affichage dans une carte KPI avec Recharts
Nombre total de lampadaires	Extraction depuis la collection LampPost via API Node.js	Filtrage des lampadaires actifs, comptage total	Affichage dans une carte KPI avec Recharts
Nombre de pannes	Extraction depuis la collection LampPost avec statut fault	Filtrage par statut, comptage des lampadaires en panne	Affichage dans une carte KPI avec Recharts
Nombre d'alertes actives	Extraction depuis la collection Alert avec statut open	Filtrage des alertes ouvertes et critiques, comptage	Affichage dans une carte KPI avec Recharts
Tickets ouverts	Extraction depuis la collection Ticket avec statut open ou in_progress	Filtrage par statut, comptage des tickets en cours	Affichage dans une carte KPI avec Recharts
Energie consommée (24h)	Extraction depuis InfluxDB via API Node.js	Agrégation de la consommation sur 24h, calcul du total en kWh	Affichage dans une carte KPI avec Recharts
Consommation par zone	Extraction depuis InfluxDB groupée par zoneId	Regroupement par zone, agrégation de la consommation moyenne	Affichage dans un diagramme en barres verticales avec Recharts
Tendance énergétique	Extraction depuis InfluxDB sur la période sélectionnée	Tri par date, calcul de la moyenne mobile de consommation	Affichage dans une courbe temporelle avec Recharts

Statut réseau	Extraction depuis la collection LampPost avec tous les statuts	Classification des lampadaires par statut, comptage par catégorie	Affichage dans un donut chart avec Recharts
Pannes par jour	Extraction depuis la collection Alert groupée par date	Regroupement par jour, comptage des pannes et critiques	Affichage dans une courbe temporelle avec Recharts
Répartition par sévérité	Extraction depuis la collection Alert avec champ severity	Classification par sévérité, comptage par niveau	Affichage dans un diagramme en barres horizontales avec Recharts
Alertes par type	Extraction depuis la collection Alert avec champ type	Regroupement par type, comptage par catégorie	Affichage dans un diagramme en barres verticales avec Recharts
Alertes par sévérité	Extraction depuis la collection Alert avec champ severity	Classification par sévérité, calcul des pourcentages	Affichage dans un donut chart avec Recharts
Métriques IoT Live	Extraction depuis InfluxDB en temps réel par lampPostId	Nettoyage des valeurs aberrantes, calcul des moyennes	Affichage dans un tableau de données avec Recharts
Distribution statuts IoT	Extraction depuis InfluxDB avec champ status	Classification par statut (on, off, fault), comptage	Affichage dans un donut chart avec Recharts

TABLE 6.14 – Construction du Tableau de bord

6.4 Réalisation et tests

Cette section décrit le processus de mise en œuvre du système ainsi que les différents tests réalisés afin de valider son bon fonctionnement et sa conformité aux exigences spécifiées.

Interface Tableau de bord

L'interface « Tableau de bord » constitue le point d'entrée principal de la plateforme Smart Lighting. Elle offre à l'utilisateur une vue globale et centralisée de l'ensemble des indicateurs clés du réseau d'éclairage public, regroupant les KPI opérationnels, les métriques IoT en temps réel, les prédictions énergétiques générées par le microservice IA, ainsi que les graphiques d'évolution de la consommation par zone et de la tendance énergétique du réseau.

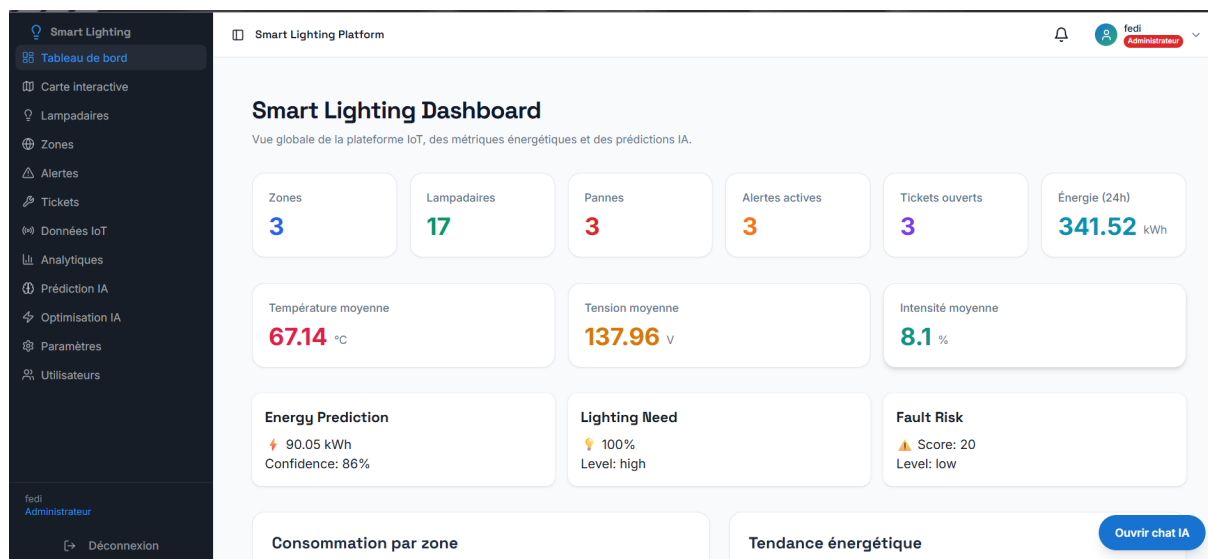


FIGURE 6.6 – Interface Tableau de bord

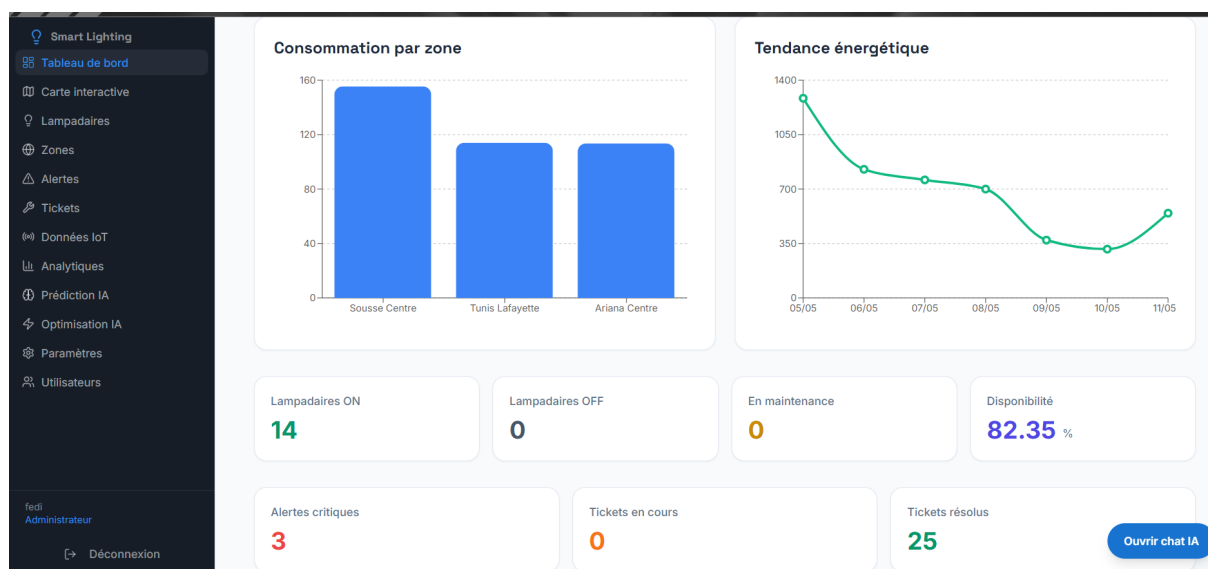


FIGURE 6.7 – Interface Tableau de bord

Interface Analytiques – Vue d’ensemble

L’interface « Analytiques – Vue d’ensemble » constitue le tableau de bord décisionnel centralisé de la plateforme Smart Lighting. Elle offre à l’utilisateur une vision synthétique et globale de l’état du réseau d’éclairage public, en agrégeant les indicateurs clés tels que le nombre de lampadaires actifs, le taux de disponibilité, les alertes ouvertes, les tickets actifs et la puissance totale consommée, accompagnés de graphiques dynamiques illustrant la tendance énergétique, le statut réseau, la consommation par zone et les alertes récentes.

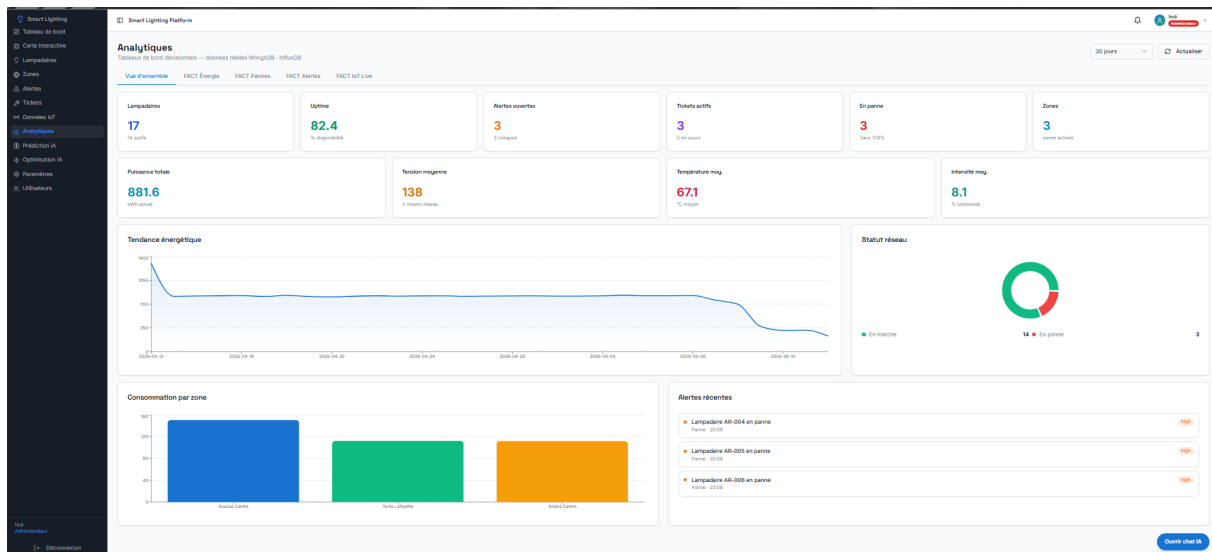


FIGURE 6.8 – Interface Analytiques – Vue d’ensemble

Interface Analytiques – FACT Énergie

L'interface « FACT Énergie » constitue le module d'analyse énergétique de la plateforme Smart Lighting. Elle permet à l'utilisateur de visualiser les indicateurs clés de consommation énergétique du réseau, notamment l'énergie totale consommée en kWh, la tension moyenne, l'intensité lumineuse moyenne et le nombre de zones analysées, accompagnés d'une courbe temporelle de tendance de consommation, d'un diagramme de consommation par zone et d'un camembert de répartition géographique.

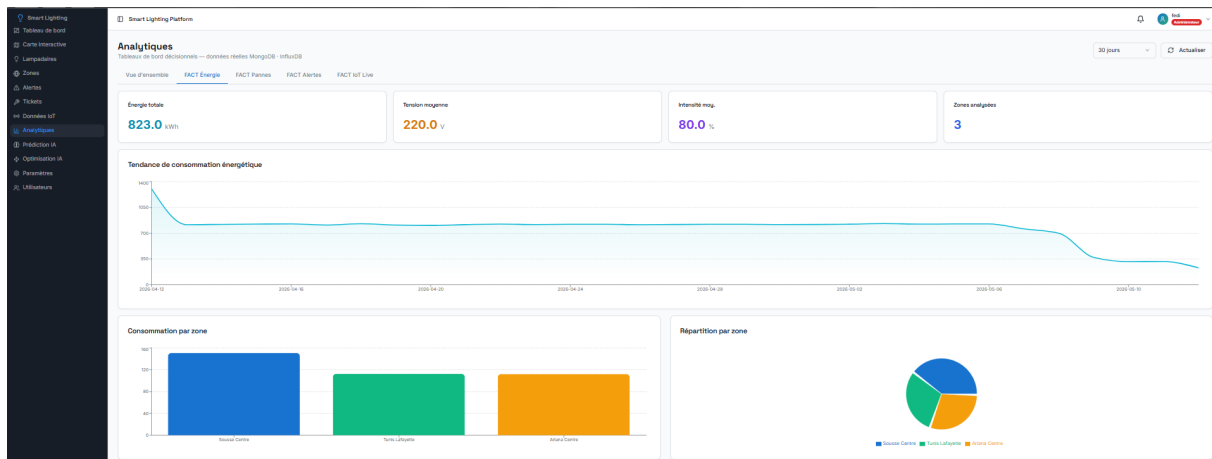


FIGURE 6.9 – Interface Analytiques – FACT Énergi

Interface Analytiques – FACT Pannes

L'interface « FACT Pannes » constitue le module d'analyse des incidents de la plateforme Smart Lighting. Elle offre à l'utilisateur une vue détaillée des pannes survenues sur le réseau, en affichant le total des pannes, le nombre de pannes critiques, le taux de résolution et le temps moyen de résolution (MTTR), accompagnés d'une courbe d'évolution des pannes par jour, d'un diagramme des zones les plus critiques, d'une répartition par sévérité et d'un indicateur de criticité par zone.

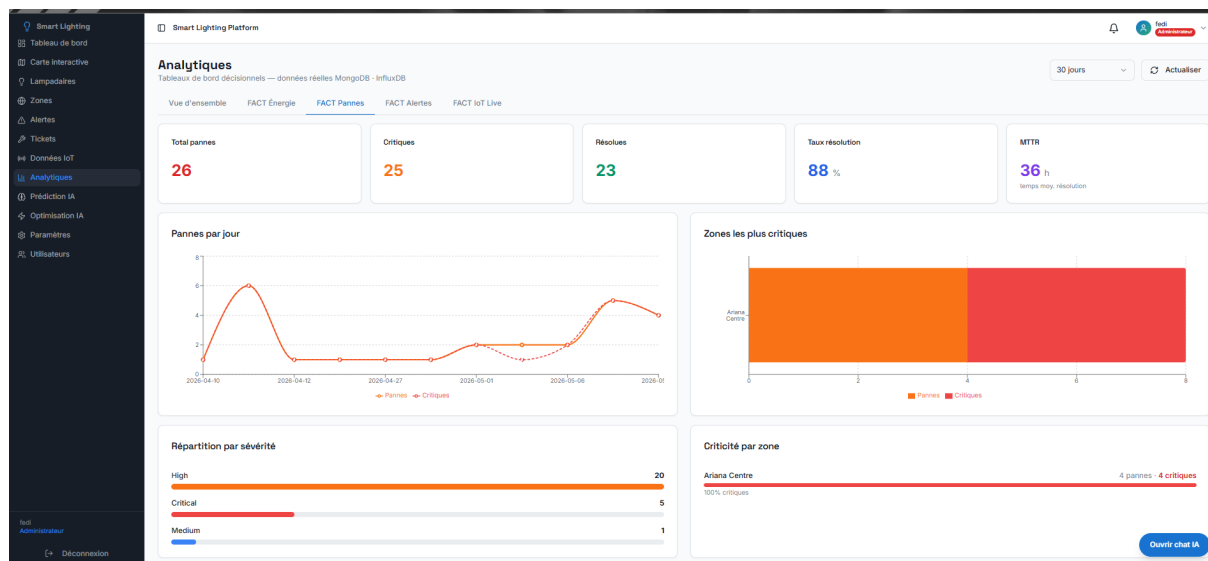


FIGURE 6.10 – Interface Analytiques – FACT Pannes

Interface Analytiques – FACT Alertes

L'interface « FACT Alertes » constitue le module d'analyse des alertes de la plateforme Smart Lighting. Elle permet à l'utilisateur de visualiser l'ensemble des alertes générées par le système, en affichant le total des alertes, le nombre d'alertes ouvertes et résolues, les critiques actives et le taux de résolution, accompagnés d'une courbe de tendance des alertes, d'un diagramme de répartition par type, d'un donut chart de distribution par sévérité et d'un diagramme de comparaison des alertes par zone géographique.

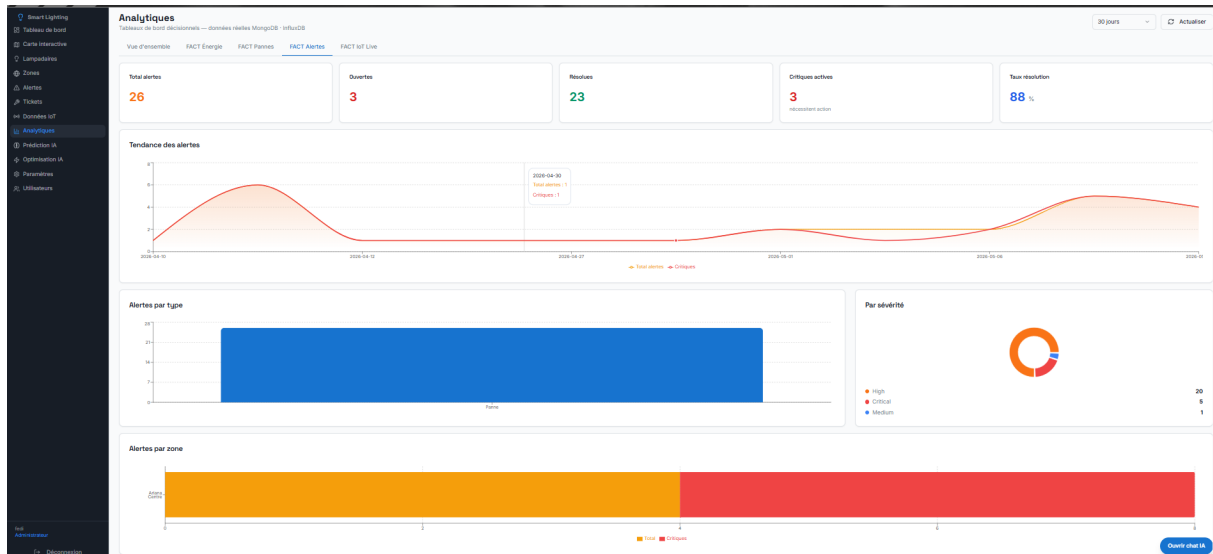


FIGURE 6.11 – Interface Analytiques – FACT Alertes

Interface Analytiques – FACT IoT Live

L'interface « FACT IoT Live » constitue le module de supervision temps réel des capteurs de la plateforme Smart Lighting. Elle offre à l'utilisateur une visualisation dynamique des métriques collectées en temps réel par les capteurs des lampadaires via InfluxDB, notamment la tension, le courant, l'intensité lumineuse, la température et l'énergie moyennes, accompagnées d'une courbe de tendance énergétique IoT, d'un donut chart de distribution des statuts et d'un tableau détaillé de l'activité de chaque capteur.

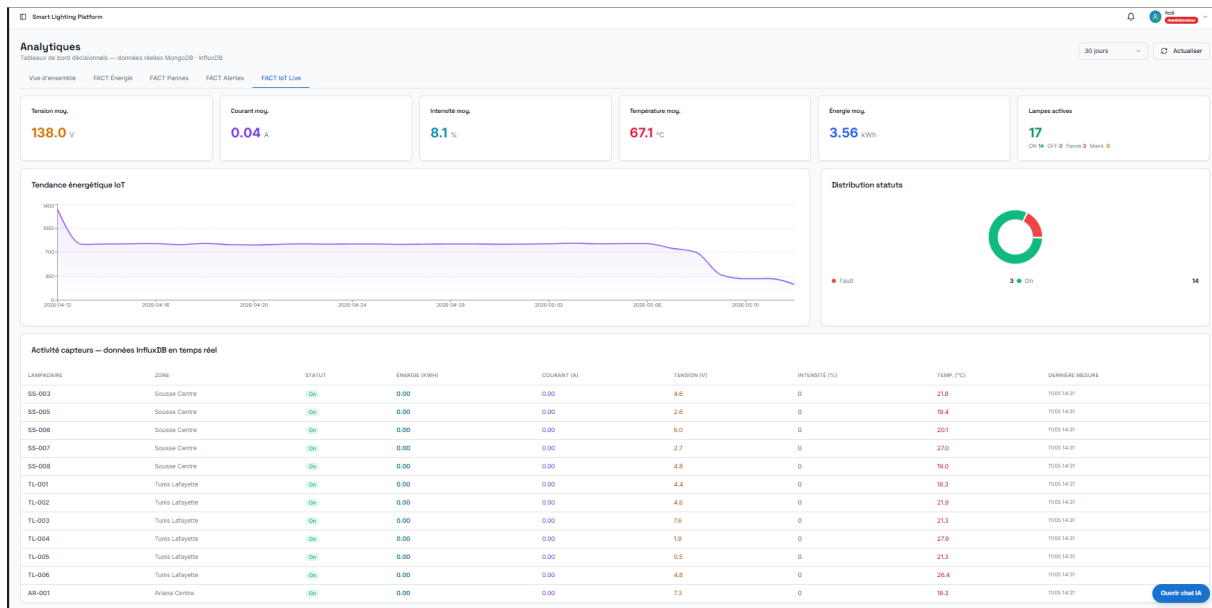


FIGURE 6.12 – Interface Analytiques – FACT IoT Live

Interface Prédiction IA

L'interface « Prédiction IA » constitue le module de supervision prédictive de la plateforme Smart Lighting. Elle permet à l'utilisateur de consulter les prédictions de consommation énergétique générées par le microservice FastAPI pour une zone et un horizon temporel sélectionnés. Elle affiche les indicateurs clés tels que l'énergie prédite en kWh, la fiabilité du modèle, l'intensité recommandée et l'économie potentielle réalisable, accompagnés d'une projection sur 7 jours comparant la courbe prévue et la courbe optimisée, d'une analyse du besoin d'éclairage avec justification IA, d'une évaluation du risque de panne par zone et de la méthode de prédiction utilisée avec son échantillon historique.

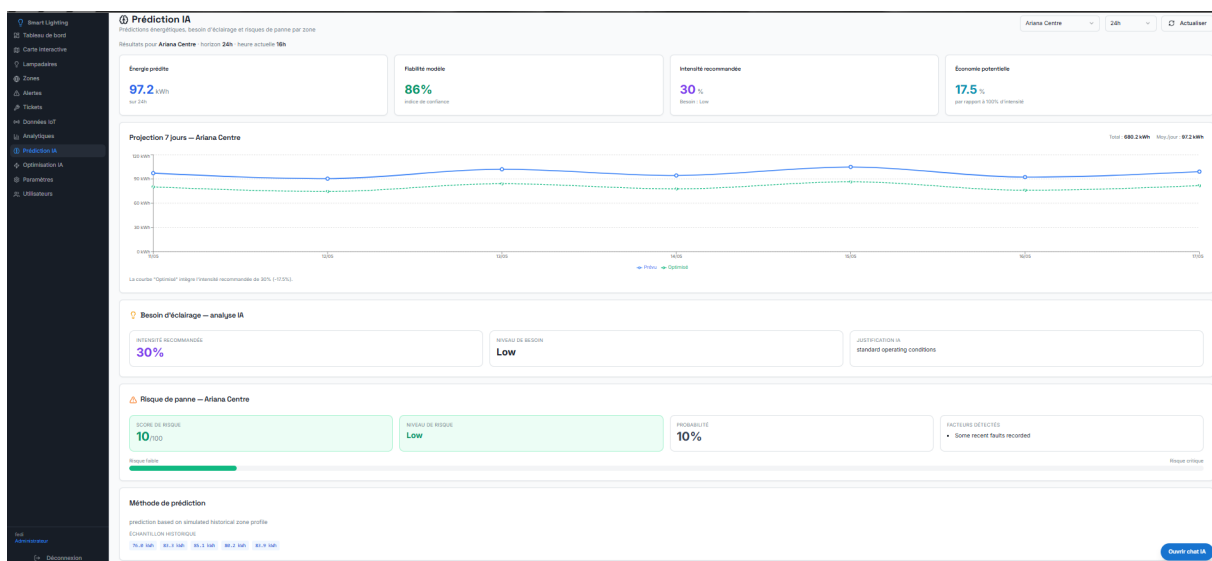


FIGURE 6.13 – Interface Prédiction IA

Interface Optimisation IA

L’interface « Optimisation IA » constitue le module d’optimisation intelligente de la plateforme Smart Lighting. Elle permet à l’utilisateur de consulter les recommandations d’optimisation générées par le microservice FastAPI pour une zone géographique sélectionnée. Elle affiche les indicateurs clés tels que l’économie estimée en pourcentage, la réduction de consommation en kWh, le CO₂ évité et l’économie financière estimée en TND, accompagnés d’un diagramme en barres comparant l’intensité lumineuse avant et après optimisation par le microservice IA sur un cycle de 24 heures, ainsi qu’un panneau de recommandation IA détaillant la stratégie adoptée, le scénario d’optimisation, l’intensité recommandée, l’intensité optimisée et le niveau de risque de panne associé.

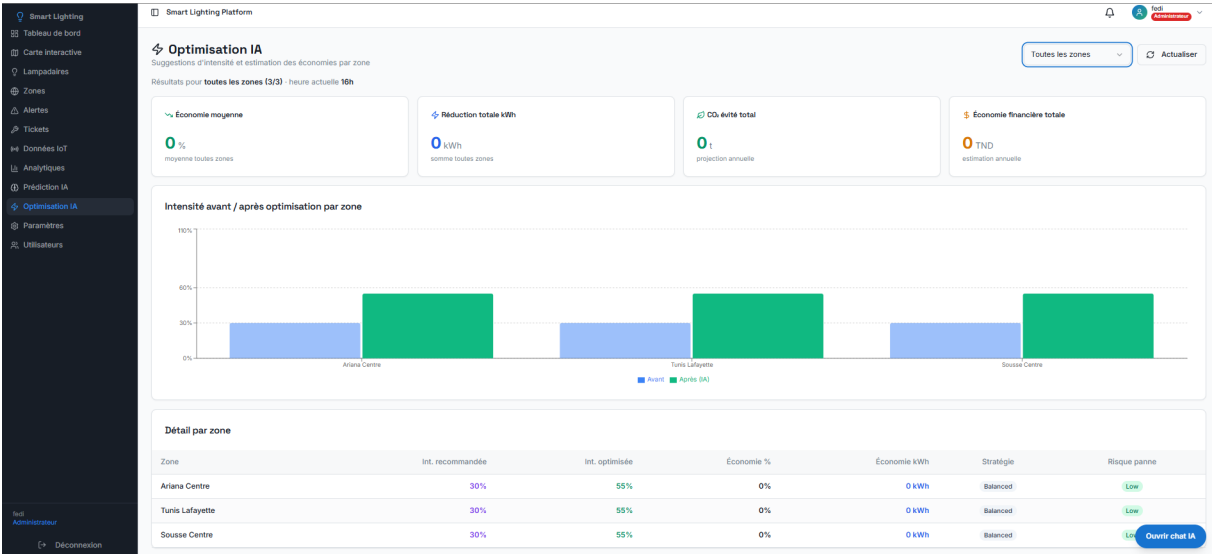


FIGURE 6.14 – Interface Optimisation IA

Interface Chatbot IA

L'interface « AI Smart Assistant » constitue le module d'assistance intelligente de la plateforme Smart Lighting. Accessible depuis toutes les pages de la plateforme via un bouton flottant, elle offre à l'utilisateur un assistant conversationnel intelligent permettant d'obtenir des analyses, des recommandations et des informations en temps réel sur le réseau d'éclairage public. Elle propose des raccourcis d'analyse rapide par zone, des actions IA directes vers les modules de prédiction et d'optimisation, ainsi qu'une zone de saisie libre permettant à l'utilisateur de formuler toute requête relative à l'état du réseau. Le micro-service IA traite chaque requête et retourne une réponse contextuelle détaillée incluant le nombre de lampadaires, les alertes ouvertes, les tickets actifs et une analyse complète de la zone concernée.

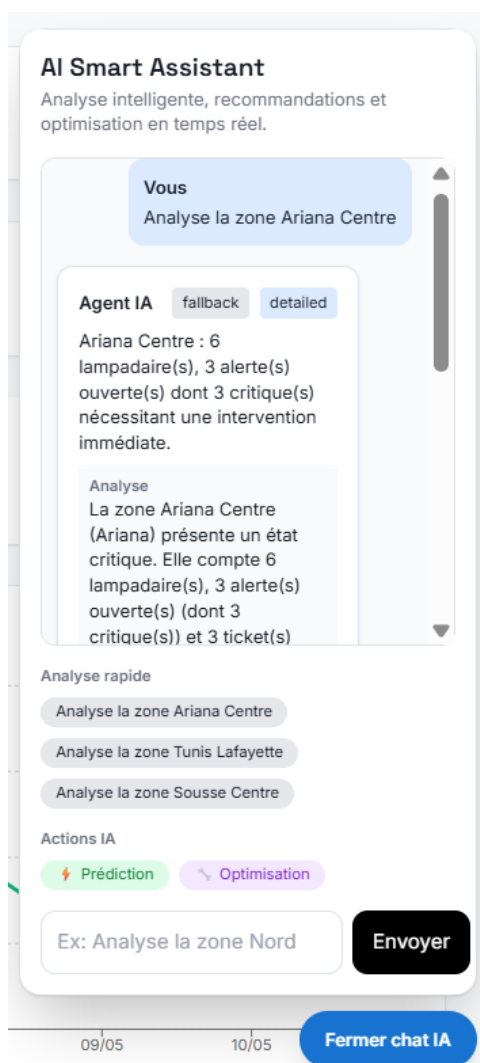


FIGURE 6.15 – Interface Chatbot IA

6.5 Réalisation et tests

Le tableau suivant présente les tests effectués pour le Sprint 4.

Cas de test	Démarche	Comportement attendu	Résultat
Test de consultation du tableau de bord	Accéder à la page « Tableau de bord » et consulter les KPI affichés ainsi que les graphiques de consommation par zone et de tendance énergétique.	Le système affiche les indicateurs clés du réseau avec les graphiques dynamiques mis à jour en temps réel.	Conforme
Test de la prédiction IA	Accéder à la page « Prédiction IA », sélectionner une zone et un horizon temporel, puis consulter les prédictions générées par le microservice FastAPI.	Le microservice IA retourne les prédictions de consommation avec la fiabilité du modèle et l'économie potentielle.	Conforme
Test de l'optimisation IA	Accéder à la page « Optimisation IA », sélectionner une zone, puis consulter les recommandations d'optimisation générées.	Le système affiche le diagramme comparatif d'intensité avant et après optimisation ainsi que la recommandation IA.	Conforme
Test du chatbot IA	Cliquer sur le bouton « Ouvrir chat IA », saisir une question relative au réseau, puis attendre la réponse générée par le microservice IA.	Le microservice IA traite la requête et affiche une réponse contextuelle incluant l'état du réseau et les recommandations.	Conforme
Test des analytiques Vue d'ensemble	Accéder à la page « Analytiques », consulter l'onglet « Vue d'ensemble » et vérifier les KPI globaux et les graphiques affichés.	Le système affiche les indicateurs globaux du réseau avec la tendance énergétique, le statut réseau et les alertes récentes.	Conforme
Test de l'analyse FACT Énergie	Accéder à l'onglet « FACT Énergie », consulter les KPI énergétiques et visualiser la courbe de tendance et la répartition par zone.	Le système affiche l'énergie totale, la tension moyenne, l'intensité et les graphiques d'évolution de la consommation.	Conforme

Cas de test	Démarche	Comportement attendu	Résultat
Test de l'analyse FACT Pannes	Accéder à l'onglet « FACT Pannes », consulter les KPI des pannes et visualiser l'évolution par jour et la criticité par zone.	Le système affiche le total des pannes, le taux de résolution, le MTTR et les graphiques de répartition par sévérité.	Conforme
Test de l'analyse FACT Alertes	Accéder à l'onglet « FACT Alertes », consulter les KPI des alertes et visualiser la tendance et la répartition par type et sévérité.	Le système affiche le total des alertes, les critiques actives, le taux de résolution et les graphiques de distribution.	Conforme
Test de l'analyse FACT IoT Live	Accéder à l'onglet « FACT IoT Live », consulter les métriques IoT en temps réel et visualiser l'activité détaillée de chaque capteur.	Le système affiche les métriques IoT avec la tendance énergétique, la distribution des statuts et le tableau d'activité des capteurs.	Conforme

TABLE 6.15 – Tests du Sprint 4

6.6 Utilisation des outils de suivi Scrum

Dans cette partie nous allons présenter les outils de suivi de la méthodologie adoptée Scrum : le Scrum board et le Burn down chart.

6.6.1 Scrum Board

À la fin du Sprint 4, l'état d'avancement du projet est illustré dans la figure ci-dessous. Deux cas d'utilisation ont déjà été terminés, notamment : communiquer avec le chatbot et consulter le tableau de bord. Deux cas d'utilisation sont en cours de réalisation, parmi lesquels : consulter la prédiction IA et consulter l'optimisation IA. Enfin, un cas d'utilisation reste à accomplir : consulter les analytiques, afin d'atteindre l'ensemble des objectifs fixés pour ce sprint.

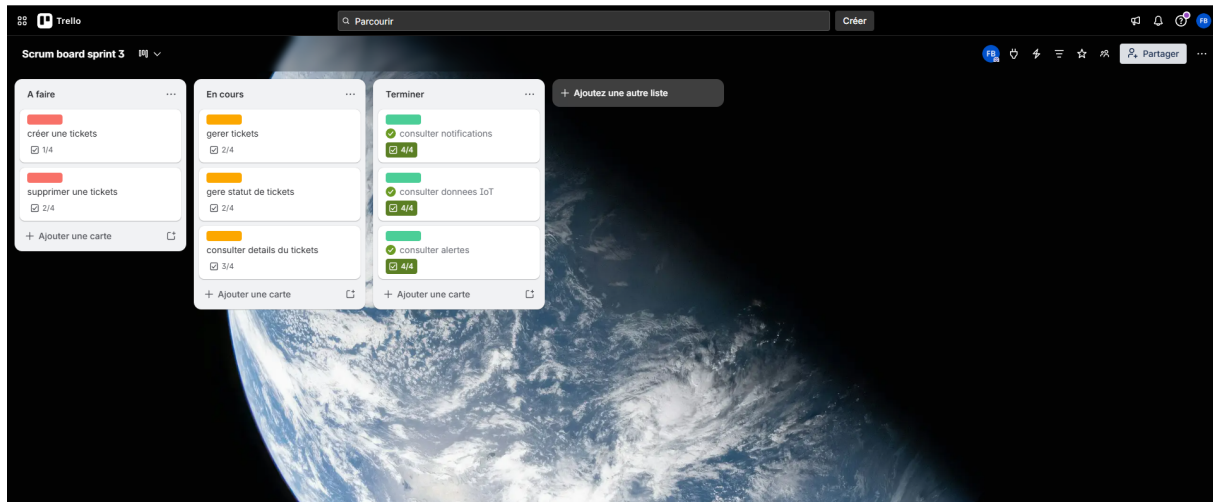


FIGURE 6.16 – Tableau Scrum Sprint 4

6.6.2 Burn Down Chart

l'évolution du travail durant le Sprint 4. Au cours de la première semaine, la progression a été légèrement en dessous de la courbe idéale en raison des difficultés rencontrées lors du développement du microservice IA et de l'intégration des endpoints de prédiction et d'optimisation. Toutefois, l'équipe a su s'adapter par la suite en accélérant le rythme de développement, permettant ainsi de rattraper le retard accumulé et d'atteindre l'ensemble des objectifs fixés pour ce sprint.

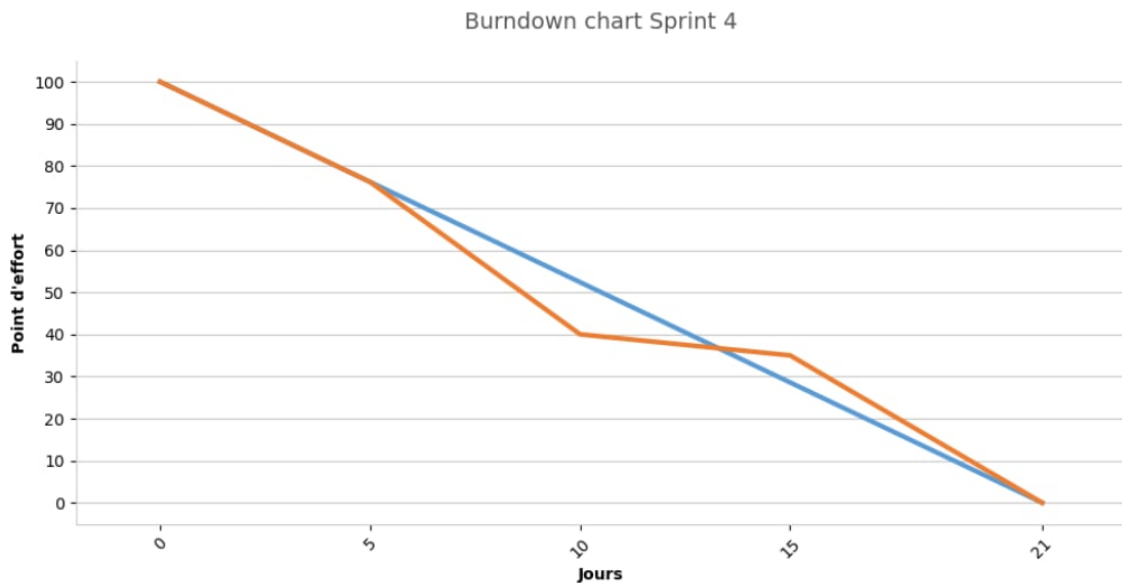


FIGURE 6.17 – Burn down à la fin de 4ème Sprint

6.7 Conclusion

Ce sprint a constitué l'aboutissement du développement de la plateforme Smart Lighting en fusionnant l'intelligence artificielle et la business intelligence. L'intégration du microservice FastAPI a permis de doter la plateforme de capacités prédictives et d'optimisation intelligente du réseau d'éclairage public, tandis que le tableau de bord avancé et les tableaux analytiques offrent aux acteurs une vision décisionnelle complète et exploitable des données du réseau. Le chatbot IA vient compléter ces fonctionnalités en offrant une assistance contextuelle et interactive en temps réel. L'ensemble de ces modules marque le passage de la plateforme vers une supervision intelligente, prédictive et décisionnelle, ouvrant la voie à de futures améliorations et extensions du système.

6.8 Diagramme de calsse globale de l'application

Le diagramme de classe global de l'application présente une vue d'ensemble structurée de l'architecture du système de gestion de l'éclairage public intelligent. Il permet de modéliser les principales entités du domaine, ainsi que leurs attributs et relations, afin de mieux comprendre l'organisation des données et des composants du système. Ce diagramme met en évidence les interactions entre les objets métier (zones, lampadaires, alertes, tickets, utilisateurs, etc.) ainsi que les services et contrôleurs assurant le traitement des différentes fonctionnalités de la plateforme.

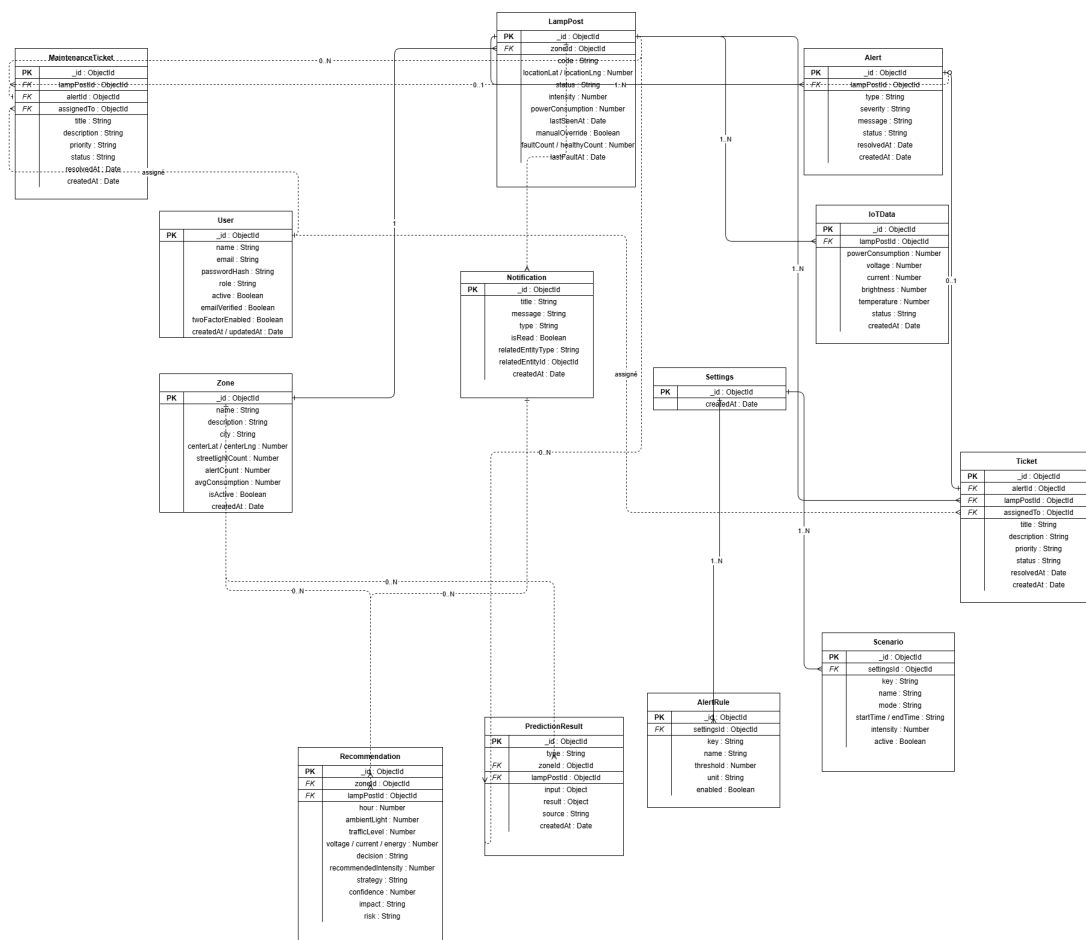


FIGURE 6.18 – Diagramme de classe globale

6.9 Schema textuelle de base de données

- User(_id, name, email, passwordHash, role, active, lastLogin, emailVerified, twoFactorEnabled, createdAt, updatedAt)
- Zone(_id, name, description, streetlightCount, alertCount, avgConsumption, city, centerLat, centerLng, isActive, createdAt)
- LampPost(_id, code, #zoneId, locationLat, locationLng, status, intensity, powerConsumption, lastSeenAt, manualOverride, faultCount, healthyCount, lastFaultAt)
- Alert(_id, #lampPostId, type, severity, message, status, resolvedAt)
- IoTData(_id, #lampPostId, powerConsumption, voltage, current, brightness, temperature, status, createdAt)
- Ticket(_id, #alertId, #lampPostId, #assignedTo, title, description, priority, status, resolvedAt)
- MaintenanceTicket(_id, #lampPostId, #alertId, title, description, priority, status, #assignedTo, resolvedAt)
- Notification(_id, title, message, type, isRead, relatedEntityType, relatedEntityId, createdAt)
- Settings(_id)
- AlertRule(_id, #settingsId, key, name, threshold, unit, enabled)
- Scenario(_id, #settingsId, key, name, mode, startTime, endTime, intensity, active)
- Recommendation(_id, #zoneId, #lampPostId, hour, ambientLight, trafficLevel, voltage, current, energy, decision, recommendedIntensity, strategy, confidence, impact, risk)
- PredictionResult(_id, type, #zoneId, #lampPostId, input, result, source)

Conclusion Générale

Ce projet de fin d'études avait pour objectif la conception et le développement d'une plateforme intelligente de gestion de l'éclairage public, baptisée Smart Lighting Platform. Développée selon une méthodologie agile Scrum, la plateforme a été réalisée en quatre sprints successifs, chacun apportant une couche fonctionnelle supplémentaire au système. Le premier sprint a permis de mettre en place le socle sécuritaire de la plateforme, en implémentant un système d'authentification robuste intégrant la vérification par email, la double authentification (2FA) et la gestion des comptes utilisateurs. Le deuxième sprint a introduit les modules fondamentaux de gestion des infrastructures, notamment la gestion des zones géographiques, des lampadaires et la visualisation sur carte interactive. Le troisième sprint a enrichi la plateforme de modules de supervision opérationnelle, à travers la gestion des alertes, des tickets de maintenance, la consultation des données IoT en temps réel et le système de notifications. Enfin, le quatrième sprint a constitué l'aboutissement du projet en fusionnant l'intelligence artificielle et la business intelligence, permettant la prédiction de la consommation énergétique, l'optimisation intelligente des lampadaires, un tableau de bord analytique avancé et un assistant conversationnel intelligent.

Sur le plan technique, la plateforme repose sur une architecture 3-tiers moderne combinant React et TypeScript pour le frontend, Node.js et Express pour le backend, MongoDB et InfluxDB pour la persistance des données, et un microservice FastAPI dédié à l'intelligence artificielle. Cette architecture garantit la modularité, la maintenabilité et l'évolutivité de la solution.

Ce projet nous a permis de consolider nos compétences en développement web full-stack, en intégration IoT, en intelligence artificielle et en business intelligence. Il nous a également permis d'acquérir une expérience concrète dans la gestion de projets agiles et dans la conception de systèmes distribués à grande échelle.

En perspective, plusieurs axes d'amélioration peuvent être envisagés, notamment le déploiement de la plateforme dans un environnement de production réel, l'intégration de capteurs IoT physiques en remplacement du simulateur LoRaWAN, l'amélioration des modèles de prédiction par l'utilisation de données historiques plus volumineuses, ainsi que l'extension des fonctionnalités décisionnelles pour une gestion encore plus proactive et intelligente du réseau d'éclairage public.

Bibliographie

- [1] Treetronix, *Site officiel*, <http://www.treetronix.com/>,
- [2] Object Management Group, *Unified Modeling Language*, <https://www.uml.org/>
- [3] OpenJS Foundation, *Node.js – Documentation officielle*, <https://nodejs.org/docs/latest/api/>
- [4] OpenJS Foundation, *Express.js – Documentation officielle*, <https://expressjs.com>
- [5] MongoDB Inc., *MongoDB – Documentation officielle*, <https://www.mongodb.com/docs/>,
- [6] InfluxData, *InfluxDB – Documentation officielle*, <https://docs.influxdata.com/>,
- [7] Meta Platforms, *React – Documentation officielle*, <https://react.dev/>,
- [8] Microsoft, *TypeScript – Documentation officielle*, <https://www.typescriptlang.org/docs/>,
- [9] Python Software Foundation, *Python – Documentation officielle*, <https://docs.python.org/3/>,
- [10] Sebastián Ramírez, *FastAPI – Documentation officielle*, <https://fastapi.tiangolo.com/>
- [11] Microsoft, *Visual Studio Code – Documentation officielle*, <https://code.visualstudio.com/docs>
- [12] Postman Inc., *Postman – Documentation officielle*, <https://learning.postman.com/docs/getting-started/overview>,
- [13] Software Freedom Conservancy, *Git – Documentation officielle*, <https://git-scm.com/doc>
- [14] GitHub Inc., *GitHub – Documentation officielle*, <https://docs.github.com/en>
- [15] JGraph Ltd, *Draw.io – Outil de modélisation*, <https://www.drawio.com/>
- [16] Recharts Group, *Recharts – Bibliothèque de visualisation*, <https://recharts.github.io/>
- [17] Volodymyr Agafonkin, *Leaflet – Bibliothèque cartographique*, <https://leafletjs.com/>
- [18] Auth0, *JSON Web Tokens – Introduction*, <https://www.jwt.io/introduction>

- [19] Scrum.org, *Le Guide Scrum*, <https://www.scrum.org/resources/scrum-guide>
- [20] Alain Fernandez, *Méthode GIMSI – Système décisionnel*, <https://www.piloter.org/mesurer/methode/methode-GIMSI-phases.htm>