



Faculté des Sciences
de Bizerte

République Tunisienne
Ministère de l'Enseignement Supérieur et
de la Recherche Scientifique

—
Université de Carthage



Département
Informatique

Rapport de Projet de Fin d'Etudes

Pour l'obtention d'une Licence en Sciences de l'Informatique

Parcours : Génie Logiciel et Système d'Information

Intitulé :

DEVELOPPEMENT D'UNE APPLICATION MOBILE (OPM MOBILE)

Réalisé par :

Benhadjmbarek Abderazek
Kaabi Ali

Au sein de

Prologic Tunisie

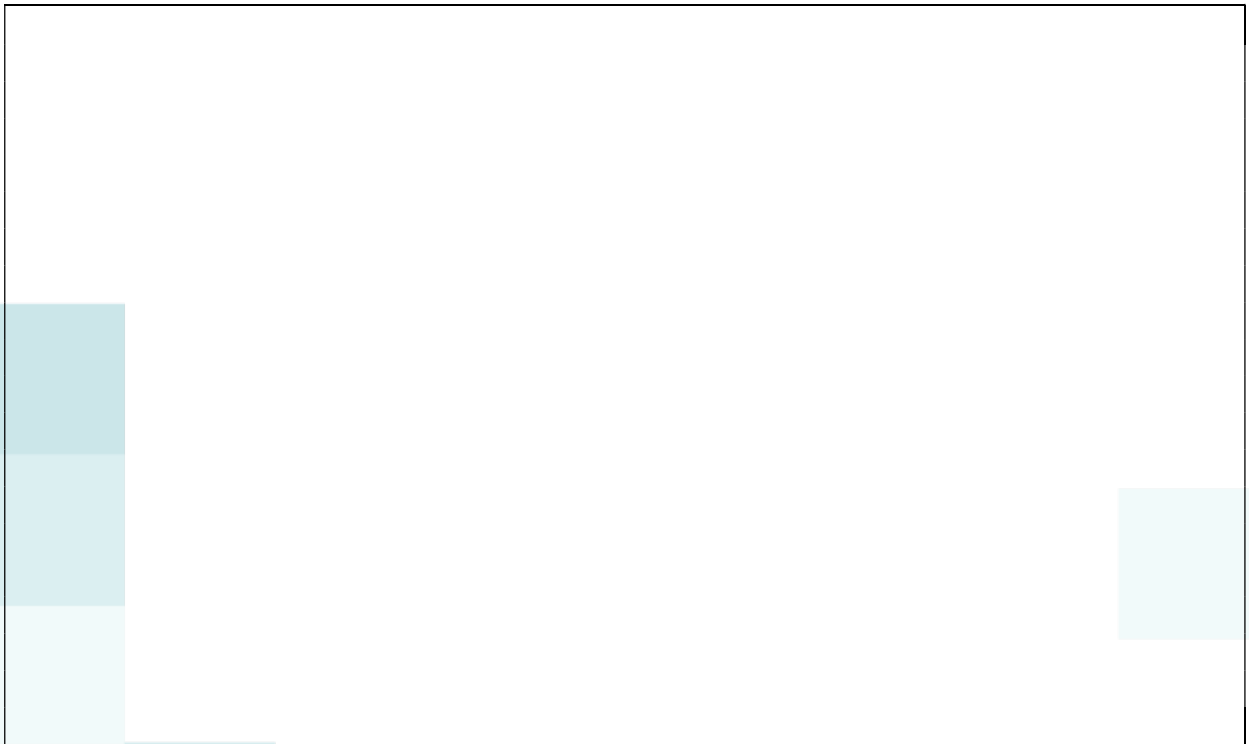
Encadré par :

Horchani Amir (Prologic Tunisie)
Amiri Samya (FSB)

Année Universitaire 2024-2025



Cadre réservé à l'encadrant académique (signature obligatoire avant dépôt)



Dédicace

*Ce travail représente bien plus qu'un simple projet académique.
Il est le fruit d'un long parcours, de sacrifices, de patience et de soutien inestimable.*

Je dédie ce mémoire avec une profonde gratitude et une immense affection :

À ma chère mère, **Hajer Dhaouadi**,
qui a toujours été ma source d'amour inconditionnel et de prière silencieuse.
Merci pour ta sagesse, et ta force invisible qui m'a porté jusqu'ici.

À mon père, **Touhami Benhadjmbarek**,
pour son soutien discret et sa confiance indéfectible.
Ta détermination et ton courage sont un exemple pour moi chaque jour.

À mes deux frères,
pour leur encouragement constant et leur présence rassurante.
Merci d'avoir toujours cru en moi, même dans les moments de doute.

Et à mes amis fidèles,
qui m'ont accompagné tout au long de ce parcours universitaire avec humour et solidarité.
Votre présence a été essentielle dans chaque étape de ce projet.

Benhadjmbarek Abderazek

Dédicace

*Ce travail représente bien plus qu'un simple accomplissement académique.
Il incarne des années d'efforts, de persévérance, de doutes surmontés et surtout de
soutien inestimable.*

Je dédie ce mémoire avec tout mon amour, ma reconnaissance et mon respect :

À ma mère, **Monia Kaabi**,
pour son amour inconditionnel, ses prières silencieuses, et sa présence rassurante dans les
moments de faiblesse.
Ta tendresse, ta patience et ta force intérieure m'ont porté plus que tu ne peux l'imaginer.

À mon père, **Mongi Kaabi**,
dont le courage, la discrétion et les sacrifices constants ont tracé devant moi le chemin de
l'effort et de la dignité.
Merci pour ta foi en moi et ton soutien inébranlable.

À mes deux frères,
pour leur complicité, leur énergie positive et leur encouragement indéfectible.
Merci d'avoir toujours été là, fiers, présents et bienveillants, même dans l'ombre.

À mes amis,
pour leur présence sincère, leur écoute attentive, et leur soutien moral tout au long de ce
parcours.
Merci pour les moments de partage, les encouragements dans les instants de doute, et les
sourires qui ont allégé bien des journées.

Chacun de vous a contribué à faire de ce projet une réussite, et je vous en serai toujours
reconnaissant.

Kaabi Ali

Remerciements

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet de fin d'études ainsi qu'à la rédaction de ce rapport.

Nous remercions tout particulièrement M. Horchani Amir, notre encadrant au sein de l'entreprise, pour son accompagnement attentif et ses précieux conseils tout au long de cette expérience. Son expertise et sa confiance ont été pour nous d'un grand soutien et nous ont permis d'évoluer dans un cadre professionnel stimulant.

Nous exprimons également notre sincère reconnaissance à Mme Amiri Samya, notre encadrante académique, ses orientations éclairées et ses encouragements constants. Sa bienveillance et son implication ont été essentielles à la réussite de ce travail.

Nous remercions également tous les membres du personnel de l'entreprise pour leur accueil chaleureux, ainsi que nos camarades et amis pour leur soutien moral et leurs encouragements tout au long de ce parcours.

Enfin, notre reconnaissance va aux membres du jury pour le temps qu'ils consacreront à l'évaluation de ce rapport et pour l'intérêt qu'ils porteront à notre travail.

Table des matières

Introduction Générale	13
Chapitre 1 Étude préalable	14
1.1 Introduction	14
1.2 Contexte général du projet	14
1.2.1 Cadre du projet	14
1.2.2 Présentation de l'organisme d'accueil	14
1.2.3 Problématiques du OPM Mobile	15
1.2.4 Description de l'existant	15
1.2.4.1 Etude de l'existant	15
1.2.4.2 Critique de l'existant	16
1.2.4.3 Solutions du OPM Mobile	16
1.3 Analyse des besoins	17
1.3.1 Identificateur des acteurs	17
1.3.1.1 Besoins fonctionnels	17
1.3.1.2 Besoins non fonctionnels	19
1.3.2 Diagramme de cas d'utilisation	19
1.3.3 Diagramme de classes globale pour opm mobile	20
1.4 Méthodologie de développement	21
1.4.1 Méthodologie de gestion de projet	21
1.4.2 Méthode agile SCRUM	22
1.4.2.1 Les rôles dans la méthode SCRUM	23
1.4.2.2 Préparation de sprint	23
Chapitre 2 Sprint 0 : Mise en place du projet	24
2.1 Introduction	24
2.2 Environnement et choix techniques	24
2.2.1 Environnement du travail	24
2.2.1.1 Environnement matériel	24
2.2.1.2 Environnement logiciel	24
2.2.2 Technologies utilisées	26
2.2.3 Structure générale du système	28
2.2.3.1 Vue matérielle de l'application	28
2.2.3.2 Vue fonctionnelle de l'application	29

2.3	Mise en œuvre du « Sprint 0 »	30
2.3.1	Backlog du produit	30
2.3.2	Planification des sprints	32
Chapitre 3	Sprint 1	33
3.1	Introduction	33
3.2	Objectifs du Sprint	33
3.3	Backlog du sprint 1	33
3.4	Diagramme d'activité	34
3.4.1	Diagramme d'activité d'user story N°1	34
3.4.2	Diagramme d'activité d'user story N°8	34
3.5	Les Scénario	35
3.5.1	Sénario de user story N°1	35
3.5.2	Sénario de user story N°8	36
3.6	Conception du Sprint	36
3.6.1	Diagramme de cas d'utilisation du sprint1	36
3.6.2	Diagramme de classe du sprint1	37
3.6.3	Vue dynamique du sprint 1	38
3.6.3.1	Diagramme de séquence objet	39
3.6.3.1.1	Diagramme de séquence objet de user story N°1	39
3.6.3.1.2	Diagramme de séquence objet de user story N°8	40
3.7	Réalisation de sprint1	40
3.7.1	Interface de user story N°1	40
3.7.2	Interface de user story N°3 + N°4	41
3.7.3	Interface de user story N°8	42
Chapitre 4	Sprint 2	43
4.1	Introduction	43
4.2	Objectifs du Sprint 2	43
4.3	Backlog du sprint 2	43
4.4	Diagramme d'activité	44
4.4.1	Diagramme d'activité d'user story N°14	44
4.4.2	Diagramme d'activité d'user story N°20	44
4.5	Les Scénario	45
4.5.1	Sénario de user story N°14	45
4.5.2	Sénario de user story N°18	46
4.5.3	Sénario de user story N°20	47
4.6	Conception du Sprint 2	48
4.6.1	Diagramme de cas d'utilisation du sprint 2	48
4.6.2	Diagramme de classe du sprint 2	49

4.6.3	Vue dynamique du sprint 2	50
4.6.3.1	Diagramme de séquence objet N°14 et N°18	50
4.6.3.2	Diagramme de séquence objet de user story N°20	51
4.7	Réalisation de sprint1	52
4.7.1	Interface de user story N°14	52
4.7.2	Interface de user story N°18	53
4.7.3	Interface de user story N°20	54
Chapitre 5	Sprint 3	55
5.1	Introduction	55
5.2	Objectifs du Sprint 3	55
5.3	Backlog du sprint 3	55
5.4	Diagramme d'activité	56
5.4.1	Diagramme d'activité d'user story N°21	56
5.4.2	Diagrammes d'activités de user story N°24	57
5.4.3	Diagrammes d'activités de user story N°28	57
5.4.4	Diagrammes d'activités de user story N°30 et N°31	58
5.5	Les Scénario	59
5.5.1	Sénario de user story N°21	59
5.5.2	Sénario de user story N°24	60
5.5.3	Sénario de user story N°32	60
5.6	Conception du Sprint 3	61
5.6.1	Diagramme de cas d'utilisation du sprint 3	61
5.6.2	Diagramme de classe du sprint 3	62
5.6.3	Vue dynamique du sprint 3	62
5.6.3.1	Diagramme de séquence objet N°21	62
5.6.3.2	Diagramme de séquence objet de user story N°24	63
5.6.3.3	Diagramme de séquence objet de user story N°32	64
5.7	Réalisation de sprint1	65
5.7.1	Interface de user story N°21	65
5.7.2	Interface de user story N°24	66
5.7.3	Interface de user story N°32	67
Chapitre 6	Sprint 4	68
6.1	Introduction	68
6.2	Objectifs du Sprint 4	68
6.3	Backlog du sprint 4	68
6.4	Diagramme d'activité	69
6.4.1	Diagrammes d'activités de user story N°35	69
6.4.2	Diagrammes d'activités de user story N°41	69
6.4.3	Diagrammes d'activités de user story N°42	70

6.5	Les Scénario	71
6.5.1	Sénario de user story N°35	71
6.5.2	Sénario de user story N°41	72
6.5.3	Sénario de user story N°42	72
6.6	Conception du Sprint 4	73
6.6.1	Diagramme de cas d'utilisation du sprint 4	73
6.6.2	Diagramme de classe du sprint 4	73
6.6.3	Vue dynamique du sprint 4	74
	6.6.3.1 Diagramme de séquence objet N°35	74
	6.6.3.2 Diagramme de séquence objet N°38 et N°39	75
	3.6.2.3 Diagramme de séquence objet de user story N°41	76
	3.6.2.4 Diagramme de séquence objet de user story N°42	77
6.7	Réalisation de sprint 4	78
6.7.1	Interface de user story N°35 et N°37	78
6.7.2	Interface de user story N°38 et N°39	79
6.7.3	Interface de user story N°41	80
6.7.4	Interface de user story N°42	81

Table des figures

1.1	Logo de Prologic Tunisie	15
1.2	Logo de Timesheet	16
1.3	Logo de Leadersoft	16
1.4	Diagramme de cas d'utilisation globale	20
1.5	Diagramme de cas d'utilisation global	21
1.6	Présentation de la méthode agile SCRUM	22
2.1	VS Code logo	25
2.2	Android studio logo	25
2.3	Draw io logo	25
2.4	Postman logo	26
2.5	Figma logo	26
2.6	MongoDB logo	26
2.7	Flutter logo	26
2.8	Dart logo	27
2.9	Node.JS logo	27
2.10	Express.JS logo	27
2.11	Javascript logo	28
2.12	Python logo	28
2.13	Architecture en Trois Couches d'une Application Mobile Moderne	29
3.1	Diagramme d'activité « Authentification d'utilisateur »	34
3.2	Diagramme « Création des dossiers des clients »	35
3.3	Diagramme de cas d'utilisation du sprint 1	37
3.4	Diagramme de classe du sprint 1	38
3.5	Diagramme d'objet séquence «L'authentification»	39
3.6	Diagramme d'objet séquence «Création d'un dossiers des contrats»	40
3.7	Authentificationn	41
3.8	L'activation et la desactivations des comptes utilisateurs	41
3.9	Creation d'un dossier pour un client	42
4.1	Diagramme d'activité « Création du contrat pour un client »	44
4.2	Diagramme d'activité « Création du type support pour les contracts »	45
4.3	Diagramme de cas d'utilisation du sprint 2	49
4.4	Diagramme de classe montre la conception du sprint 2	50
4.5	Diagramme de séquence objet «création d'un contrat avec l'ajout des sites et fichiers pour cette contrat»	51
4.6	Diagramme de séquence objet «création d'un type de support»	52
4.7	Interface de user story N°14	53
4.8	Interface de user story N°18	53
4.9	Interface de user story N°20	54

5.1	Diagramme d'activité «la création d'un équipement»	56
5.2	Diagramme d'activité «la création d'un ticket»	57
5.3	Diagramme d'activité «Réception des notifications chez les utilisateurs» .	58
5.4	Diagramme d'activité «résolution d'un ticket»	59
5.5	diagramme de cas d'utilisation du sprint 3	61
5.6	Une présentation du diagramme de classes de conception du sprint 3 . . .	62
5.7	Diagramme de séquence objet «la création d'un équipement»	63
5.8	Diagramme de séquence objet «la création d'un ticket»	64
5.9	Diagramme de séquence objet «la résolution d'un ticket»	65
5.10	Interface de user story N°21	66
5.11	Interface de user story N°24	66
5.12	Interface de user story N°32	67
6.1	Diagrammes d'activités« planification des visites préventives»	69
6.2	Diagrammes d'activités«l'accées au tableau de bord»	70
6.3	Diagrammes d'activités«génération du rapport du ticket»	71
6.4	diagramme de cas d'utilisation du sprint 4	73
6.5	Une présentation du diagramme de classes de conception du sprint 4 . . .	74
6.6	Diagramme de séquence «Planification du visite péventive»	75
6.7	Diagramme de séquence «les outils des techniciens : knowledge base et chatbot»	76
6.8	Diagramme de séquence «tableau de bord»	77
6.9	Diagramme de séquence «création du rapport d'un ticket»	77
6.10	Interface de la création du visite préventive	78
6.11	Interface du calendrier	79
6.12	interface du knowledge base	80
6.13	Interface du chatbot	80
6.14	Interface du table de bord	81
6.15	Interface de la création d'un rapport d'un ticket	81
6.16	Interface de téléchargement du rapport	82

Liste des tableaux

1.1	Gestion classique et agile d'une application mobile	22
2.1	Les machines	24
2.2	Tableau récapitulatif des User Stories du projet OPM Mobile	32
2.3	Planification des sprints	32
3.1	Backlog du sprint 1	34
3.2	Description de l'authentification d'utilisateurs	35
3.3	Description de la création des dossiers des contrats pour les clients	36
4.1	Tableau du backlog du sprint 2	43
4.2	Description textuelle de la création d'un contrat	46
4.3	Description textuelle de l'ajout des sites et fichiers pour un contrat	47
4.4	Description textuelle de la Création des types de support	48
5.1	Backlog du sprint 3	56
5.2	Description textuelle de la création d'un équipement	59
5.3	Description textuelle de la création d'un ticket	60
5.4	Description textuelle sur la résolution d'un ticket	60
6.1	Backlog du sprint 4	68
6.2	Description textuelle de la création d'une visite préventive	71
6.3	Description textuelle de tableau de bord	72
6.4	Description textuelle pour la création d'un rapport d'un ticket	72

Introduction Générale

Au cours des dernières années, le monde a connu une révolution technologique sans précédent, transformant profondément notre mode de vie. L'essor fulgurant des smartphones et des applications mobiles illustre parfaitement cette transformation. Devenues omniprésentes, ces applications jouent désormais un rôle central, tant dans la sphère personnelle que professionnelle. Elles facilitent la communication, optimisent les tâches quotidiennes, et ouvrent de nouvelles perspectives en matière de commerce et de services.

Cependant, malgré cette évolution, la majorité des applications actuelles restent limitées à des services spécifiques. Ce fonctionnement en silos oblige les utilisateurs à multiplier les applications pour répondre à leurs besoins, ce qui peut s'avérer inefficace, chronophage et peu ergonomique. Dans ce contexte, une nouvelle tendance se dessine : le développement d'applications tout-en-un, capables de regrouper divers services au sein d'une même plateforme, offrant ainsi une expérience utilisateur plus fluide et complète.

C'est dans cette optique que s'inscrit notre projet de fin d'études, réalisé au sein de la société Prologic Tunisie. Le projet porte sur la conception et le développement d'une application mobile nommée « OPM Mobile », visant à mettre en relation les clients avec des techniciens et commerciaux, en centralisant plusieurs services dans une seule interface intuitive.

Enfin, l'introduction décrit brièvement la structure du rapport :

- Le **chapitre 1** : Étude préalable Une mise en contexte du projet, une analyse de l'existant, des critiques pertinentes, ainsi qu'une proposition de solution.
- Le **chapitre 2** Sprint 0 - Mise en place du projet Présentation des outils de développement, des technologies utilisées, de l'architecture système, ainsi que l'élaboration du backlog produit et la planification des sprints.
- Les **chapitres 3 à 6** Déroulement des sprints Détail de chaque sprint, avec une analyse approfondie, une conception technique, et une évaluation des livrables à chaque étape.
- Enfin, une conclusion générale viendra récapituler le travail accompli, évaluer les résultats obtenus et proposer des perspectives d'amélioration et d'évolution du projet.

Chapitre 1

Étude préalable

1.1 Introduction

Ce premier chapitre est dédié à l'étude préalable de notre projet. Nous y présentons l'organisme d'accueil dans lequel le stage de fin d'études a été effectué. Ensuite, nous présentons les principales problématiques identifiées, en analysant certaines solutions existantes afin d'en dégager les forces et les limites. Cette analyse nous aidera à déterminer les exigences fonctionnelles et non fonctionnelles du projet. Enfin, nous décrivons la méthodologie adoptée pour sa mise en œuvre.

1.2 Contexte général du projet

1.2.1 Cadre du projet

Ce projet entre dans le cadre de la préparation du stage de fin d'études pour l'obtention de la licence fondamentale en sciences d'informatique à la Faculté des sciences de Bizerte (FSB). Il a été effectué au sein de la société Prologic Tunisie durant une période de 4 mois.

1.2.2 Présentation de l'organisme d'accueil

Prologic Tunisie, créée en 1985 par des passionnés de nouvelles technologies, dispose de 35 ans d'expertise dans le conseil, la vente et le service informatique. Cette société est aujourd'hui un acteur majeur du marché tunisien, offrant aux entreprises de toutes tailles des solutions IT complètes — de la conception à la maintenance — reposant sur des stratégies évolutives et à haute valeur ajoutée.

Les services que Prologic Tunisie offre sont les suivants :

- Services Cloud
- IT as a Service
- ICT Housing
- SaaS (Software as a Service)
- PaaS (Platform as a Service) : solutions de sauvegarde pour les entreprises



FIGURE 1.1 – Logo de Prologic Tunisie

1.2.3 Problématiques du OPM Mobile

La gestion actuelle des projets, des interventions et des tickets au sein de l'entreprise se caractérise par une approche largement manuelle et fragmentée, reposant sur l'utilisation de plusieurs outils non intégrés. Cette condition provoque un suivi inefficace, une perte de temps massive et un manque de visibilité pour l'ensemble des acteurs impliqués, tels que les administrateurs, les responsables de projets (PMO), les techniciens, les commerciaux et les clients.

Parmi les principales difficultés identifiées, on note :

1. Une gestion manuelle et dispersée des données relatives aux tickets, contrats et interventions .
2. Une absence de centralisation des outils de travail, rendant la coordination complexe.
3. Une traçabilité limitée des actions réalisées par les utilisateurs.
4. Des retards et pertes de temps dans le suivi des projets et des interventions sur le terrain .
5. Un manque de fonctionnalités avancées, telles que les notifications automatiques et les statistiques en temps réel, qui pourraient améliorer la réactivité et l'efficacité des équipes.

1.2.4 Description de l'existant

1.2.4.1 Etude de l'existant

Pour l'instant, la gestion des projets, interventions et tickets est effectuée de manière manuelle ou à l'aide de plusieurs outils non intégrés. Cette approche entraîne une perte de temps considérable, une inefficacité dans le suivi des actions, ainsi qu'un manque de transparence pour les différents participants, y compris les administrateurs, pmo, techniciens, commerciaux et clients.

Dans notre pays tunisie, certaines entreprises locales proposent des solutions de gestion, Cependant celles-ci ne satisfont pas toujours aux besoins spécifiques du domaine de l'infogérance.

Timesheet est une outil tunisienne centré essentiellement sur la gestion du temps et des tâches des employés. Cette interface facile à utiliser permet un suivi quotidien des activités. Cependant, elle présente des limites lorsqu'il s'agit de gérer des interventions complexes, des contrats multiples ou une coordination entre plusieurs acteurs tels que les clients et les techniciens.

Timesheet

FIGURE 1.2 – Logo de Timesheet

Leadersoft offre une solution ERP intégrale, englobant différents modules comme la comptabilité, la facturation, la gestion des affaires et la gestion de projet. Bien que solide, option cible essentiellement les grandes entreprises et pourrait être vue comme inflexible, coûteuse, et difficile à personnaliser pour des besoins spécifiques de gestion de tickets ou de sites dispersés sur plusieurs localisations.



FIGURE 1.3 – Logo de Leadersoft

1.2.4.2 Critique de l'existant

Les principales limites de l'existant sont :

- **Absence d'un système centralisé** : les informations sont éparpillées entre divers outils (Excel, emails, appels téléphoniques), rendant difficile la consolidation des données.
- **Manque de mobilité** : aucune solution mobile intuitive n'est disponible pour permettre un accès rapide aux données sur le terrain, en particulier pour les techniciens.
- **Suivi des interventions inefficace** : il n'existe pas de système en temps réel permettant de suivre les tickets, interventions ou SLA.
- **Solutions locales limitées** : les outils comme Timesheet et Leadersoft ne couvrent pas de manière optimale tous les besoins liés à l'externalisation des services, notamment la gestion des contrats par site, les interventions programmées ou les tableaux de bord dynamiques.

1.2.4.3 Solutions du OPM Mobile

L'objectif de OPM Mobile est de centraliser et automatiser ces processus à travers une application mobile intuitive, optimisant la gestion des dossiers, contrats, tickets et interventions, tout en intégrant un système de notifications et de statistiques en temps réel.

Parmi les solutions :

1. Centralisation des processus dans une seule application mobile.

2. Authentification sécurisée et gestion des rôles via JWT pour un accès personnalisé aux fonctionnalités.
3. Système de notifications et calendrier pour améliorer le suivi des tâches et le respect des délais.
4. Messagerie intégrée et partage de fichiers pour faciliter la communication entre les techniciens et les clients.
5. Dashboard et statistiques en temps réel pour un meilleur pilotage de l'activité et des performances.

1.3 Analyse des besoins

1.3.1 Identificateur des acteurs

Administrateur système, PMO, technicien, commercial et client sont les acteurs qui interagissent avec notre système :

- Administrateur système (gère le système)
- PMO (gestionnaire de projet)
- Technicien (acteur clé de la plateforme)
- Commercial (acteur stratégique de la plateforme)
- Client (acteur actif dans l'écosystème OPM Mobile)

1.3.1.1 Besoins fonctionnels

Les besoins fonctionnels correspondent aux services et capacités que le futur système doit fournir et adaptés aux différents profils d'utilisateurs.

Administrateur de système :

- Gérer les utilisateurs
- Accéder au dashboard
- Gérer les contrats
- Gérer les dossiers des clients
- Gérer les contrats sous les dossiers des clients
- Gérer l'équipe sous les contrats
- Gérer les visites préventives
- Gérer les tickets
- Gérer les sites sous les contrats
- Gérer les équipements sous les sites
- Gérer les fichiers de contrat
- Gérer les types de support
- Accéder au calendrier
- Accéder à la base de connaissances
- Gérer son propre profil

Client :

- Gérer son compte
- Accéder au dashboard
- Gérer les tickets
- Accéder au chat de ticket
- Gérer les fichiers de contrat
- Accéder au calendrier
- Gérer son propre profil

Technicien :

- Créer son compte
- Gérer son compte
- Accéder au dashboard
- Gérer les tickets
- Accéder au chat de ticket
- Accéder au calendrier
- Accéder à la base de connaissances
- Gérer son propre profil

Commercial :

- Gérer son compte
- Accéder au dashboard
- Consulter les contrats
- Consulter les dossiers des clients
- Consulter les contrats sous les dossiers des clients
- Consulter l'équipe sous les contrats
- Consulter les visites préventives
- Consulter les sites sous les contrats
- Consulter les équipements sous les sites
- Consulter les fichiers de contrat
- Accéder au calendrier
- Gérer son propre profil

PMO :

- Gérer son compte
- Accéder au dashboard
- Gérer les contrats

- Gérer les dossiers des clients
- Gérer les contrats sous les dossiers des clients
- Gérer l'équipe sous les contrats
- Gérer les visites préventives
- Gérer les tickets
- Gérer les sites sous les contrats
- Gérer les équipements sous les sites
- Gérer les fichiers de contrat
- Gérer les types de support
- Accéder au calendrier
- Gérer son propre profil

1.3.1.2 Besoins non fonctionnels

Les besoins non fonctionnels représentent les exigences liées à la qualité, la performance et les conditions d'utilisation du système, sans décrire des fonctionnalités spécifiques.

- **Sécurité** : Stockage sécurisé des mots de passe (hachage avec bcrypt) et l'archivage sécurisé des données sensibles (contrats, tickets, etc).
- **Compatibilité et Portabilité** : L'application Flutter doit fonctionner sur iOS, Web et Android.
- **Interopérabilité** : Intégration d'API externes (ex. : OpenMapStreet pour la localisation) et une Capacité d'importation/exportation de données au format PDF.
- **Maintenabilité** : Code structuré et modulaire. pour permettre des évolutions faciles (la simplicité d'ajout des nouvelles fonctionnalités ou correction des bugs).
- **Conformité** : Gestion du consentement utilisateur pour les données personnelles et respect des SLA (ex. : résolution des tickets sous 24 h) dans le cadre de la conformité réglementaire (RGPD).

1.3.2 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est une représentation concrète d'une interaction entre un utilisateur et le système, visant à atteindre un objectif précis. En analysant en détail les spécifications, on identifie ces cas d'utilisation pour organiser et modéliser.

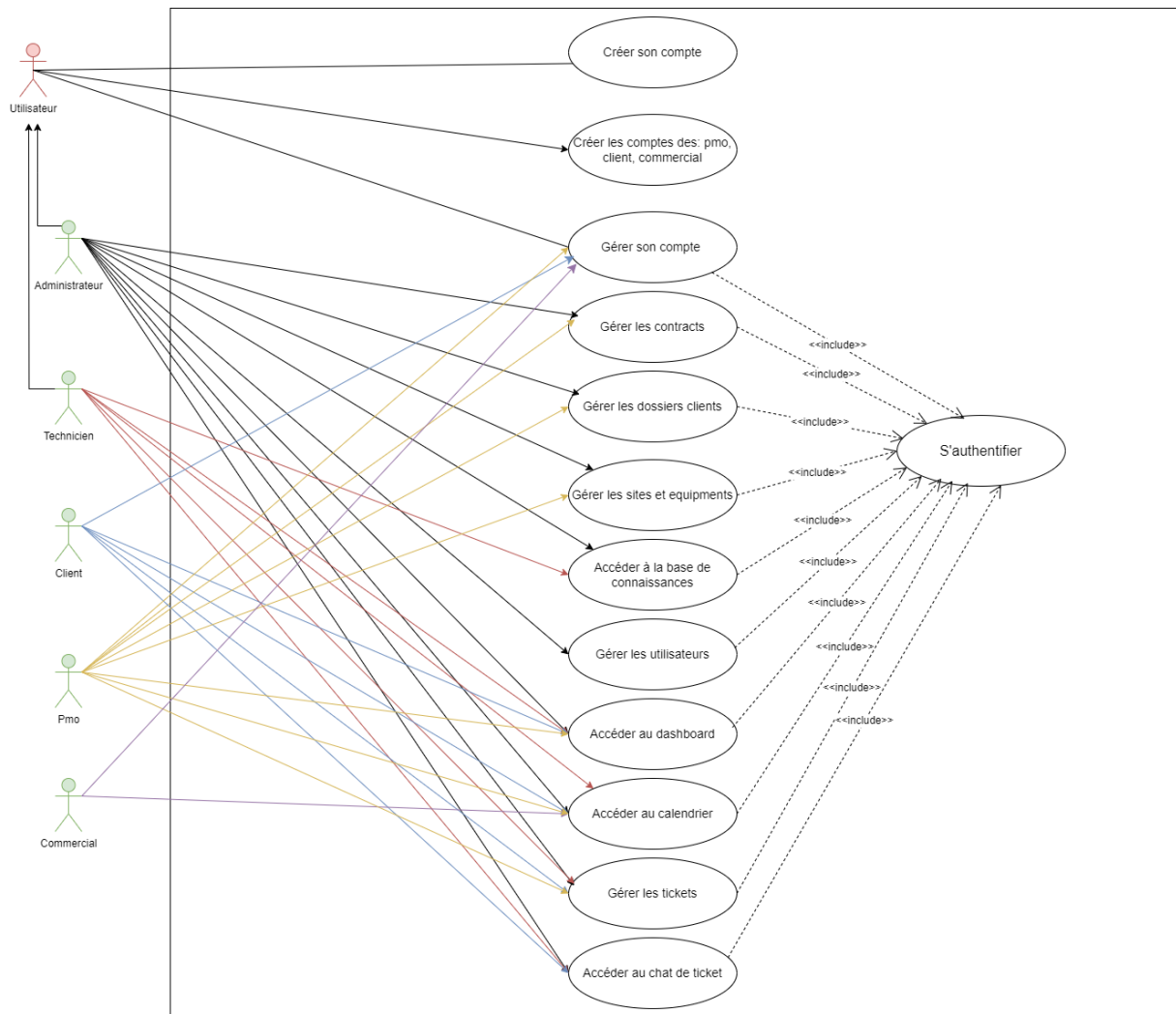


FIGURE 1.4 – Diagramme de cas d'utilisation globale

1.3.3 Diagramme de classes globale pour opm mobile

Ce schéma UML représente un système de gestion de contrats, tickets et interventions techniques. Il détermine des éléments cruciaux comme Utilisateur, Contrat, Ticket, Visite, Equipement et Support, ainsi que leurs relations. Chaque utilisateur se voit attribuer un rôle (administrateur, pmo, technicien, client, commercial) qui détermine ses permissions. Les dossiers contiennent des contrats qui sont associés à des sites et équipements spécifiques. Les tickets servent à gérer les incidents, avec un suivi via messages, solutions et visites techniques. Le système comprend aussi une base de connaissance, un calendrier et un tableau de bord pour le suivi opérationnel et décisionnel.

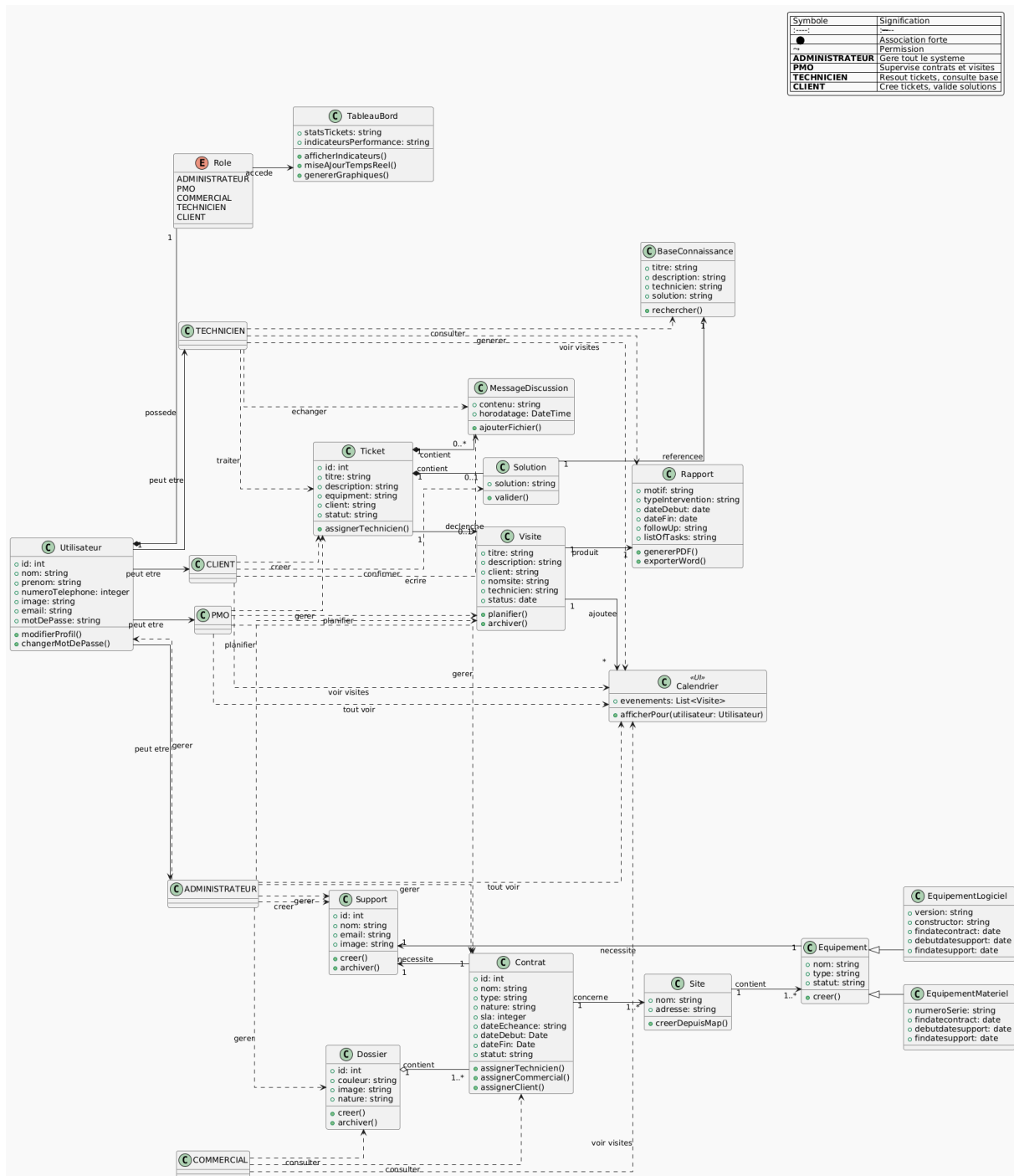


FIGURE 1.5 – Diagramme de cas d'utilisation global

1.4 Méthodologie de développement

1.4.1 Méthodologie de gestion de projet

Les projets informatiques évoluent constamment en complexité, ce qui rend indispensable l'usage de méthodologies structurées. Ces approches permettent de guider chaque phase du projet — de la planification initiale jusqu'à la réalisation finale — tout en optimisant les ressources, le temps et les coûts. Une méthodologie de gestion de projet constitue un ensemble cohérent de règles, d'outils et de pratiques adoptés par les pro-

fessionnels pour organiser et piloter efficacement leur travail. On distingue généralement deux grandes familles : les approches traditionnelles dites prédictives, et les approches agiles, plus flexibles et adaptatives.

TABLE 1.1 – Gestion classique et agile d’une application mobile

Aspect	Méthode Classique	Méthode Agile
Structure du projet	Étapes fixes et linéaires.	Cycles courts avec livraisons fréquentes.
Planification	Tout est défini dès le début.	Plan évolutif selon les retours.
Gestion des risques	Gérés au départ.	Suivi en continu.
Organisation de l’équipe	Structurée et hiérarchisée.	Autonome et collaborative.
Gestion de la qualité	Contrôle en fin de cycle.	Tests réguliers et précoces.
Réaction au changement	Peu accepté.	Favorisé et intégré.
Suivi du progrès	Basé sur le plan initial.	Basé sur les fonctions livrées.
Critère de réussite	Respect du cahier des charges.	Satisfaction et valeur pour l’utilisateur.

Les méthodes agiles les plus populaires aujourd’hui :

- Scrum
- Kanban
- XP (Extreme Programming)
- SAFe (Scaled Agile Framework)
- Lean

1.4.2 Méthode agile SCRUM

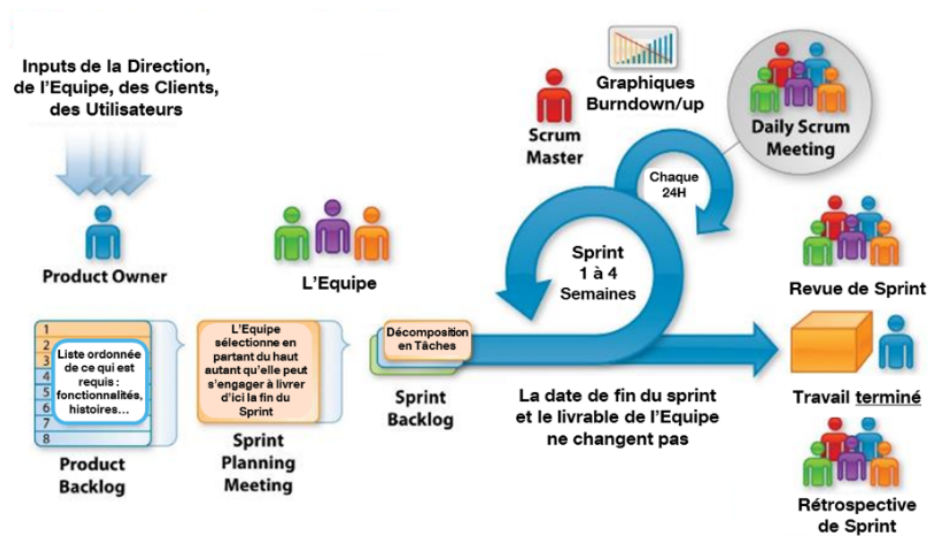


FIGURE 1.6 – Présentation de la méthode agile SCRUM

Scrum est une méthode Agile qui organise le travail en cycles courts appelés sprints. Elle favorise la collaboration entre l'équipe et le client pour fournir rapidement des fonctionnalités. Scrum utilise des rôles et réunions régulières pour s'adapter aux changements et améliorer le produit.

1.4.2.1 Les rôles dans la méthode SCRUM

Scrum s'appuie sur trois rôles essentiels qui coopèrent pour produire rapidement et efficacement des fonctionnalités à forte valeur ajoutée :

- **Product Owner** : Il porte la vision du produit, gère le backlog et hiérarchise les fonctionnalités en fonction de la valeur apportée aux utilisateurs et aux parties prenantes.
- **Scrum Master** : Il facilite le processus Scrum, supprime les obstacles, veille au respect des bonnes pratiques et aide l'équipe à s'améliorer en continu.
- **Équipe de développement** : Groupe de professionnels multifonctionnels (développeurs, testeurs, designers...) qui conçoivent, codent et livrent les incréments fonctionnels à chaque sprint.

1.4.2.2 Préparation de sprint

Dans Scrum, la préparation du sprint (Sprint Planning) est l'atelier qui lance chaque itération. Elle consiste à :

- **Sélectionner les éléments du backlog** : l'équipe choisit les user stories ou tâches prioritaires que le Product Owner a mises en avant.
- **Définir l'objectif du sprint** : un but clair et partagé, qui guide les efforts de l'équipe pendant toute la durée du sprint.
- **Découper et estimer le travail** : pour chaque élément, on détaille les tâches à accomplir et on estime leur charge, afin d'ajuster la portée du sprint à la capacité de l'équipe.

Conclusion En somme, Scrum organise le travail en courts sprints pour délivrer rapidement des fonctionnalités utilisables. Ses rôles dédiés et ses cérémonies régulières assurent une collaboration fluide et une vision partagée. Cette agilité permet à l'équipe de s'ajuster en permanence et de maximiser la valeur pour les utilisateurs

Chapitre 2

Sprint 0 : Mise en place du projet

2.1 Introduction

Ce chapitre détaille l'environnement matériel et logiciel mis en place pour lancement du projet. Il expose également les choix techniques effectués afin de préparer une base solide pour le développement de l'application.

2.2 Environnement et choix techniques

Cette partie décrit l'environnement de développement ainsi que les technologies choisies pour la réalisation du projet, les choix techniques ont été guidés par les besoins en performance, sécurité, et portabilité de l'application.

2.2.1 Environnement du travail

2.2.1.1 Environnement matériel

Machine	Système d'exploitation	RAM	Processeur	GPU
PC ASUS Tuf Gaming	Windows 11 64 bits	16 Go	Intel® Core™ i5-10300H @ 2,50 GHz	GTX 1650, 4 Go VRAM
PC ASUS Tuf Gaming	Windows 11 64 bits	16 Go	Intel® Core™ i5-11300H @ 2,50 GHz	GTX 1650, 4 Go VRAM

TABLE 2.1 – Les machines

2.2.1.2 Environnement logiciel

Visual Studio Code : L'IDE *Visual Studio Code* est un panneau de lancement créatif que vous pouvez utiliser pour modifier, déboguer et générer du code, puis publier une application. En plus de l'éditeur et du débogueur standard fournis par la plupart des IDE, Visual Studio Code inclut des compilateurs, des outils de complétion de code, des concepteurs graphiques et bien d'autres fonctionnalités pour améliorer le processus de développement de logiciels.

[1]

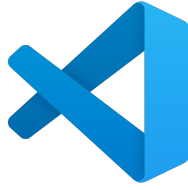


FIGURE 2.1 – VS Code logo

Android Studio :

Android Studio est l'environnement de développement intégré (IDE) officiel pour la création d'applications pour le système d'exploitation Android. Conçu et maintenu par Google, il est mis à disposition des développeurs afin de leur faciliter le processus de développement d'applications mobiles. Basé sur le logiciel IntelliJ IDEA de JetBrains, Android Studio offre une multitude d'outils et de fonctionnalités adaptées au développement Android. [2]



FIGURE 2.2 – Android studio logo

Draw.io :

Draw.io est une suite technologique pour la création d'applications de diagrammes, et le logiciel de diagrammes en ligne le plus utilisé au monde. [3]



FIGURE 2.3 – Draw io logo

Postman :

Postman est un outil qui permet aux développeurs de tester et de documenter les API (Application Programming Interface). Il permet de créer des requêtes HTTP (GET, POST, PUT, etc.), de gérer les paramètres et les entêtes, de vérifier les réponses et les statuts de retour, et de sauvegarder des requêtes pour une utilisation ultérieure. [4]



FIGURE 2.4 – Postman logo

Figma :

Figma est une plateforme collaborative pour éditer des graphiques vectoriels et faire du prototypage. Elle permet de concevoir des design systems pour faciliter la création de sites web et d'applications mobiles. C'est une solution à destination des UI et UX designers et des développeurs. [5]



FIGURE 2.5 – Figma logo

MongoDB :

MongoDB est un programme de gestion de base de données NoSQL open source. NoSQL (Not only SQL) est utilisé comme alternative aux bases de données relationnelles traditionnelles. Les bases de données NoSQL sont très utiles pour travailler avec de grands ensembles de données distribuées. MongoDB est un outil qui peut gérer des informations orientées documents, stocker ou récupérer des informations. [6]



FIGURE 2.6 – MongoDB logo

2.2.2 Technologies utilisées

Flutter :

Flutter est un framework UI open source développé par Google pour créer des applications multiplateformes en utilisant le langage de programmation Dart. Il permet aux développeurs d'écrire du code une seule fois et de le déployer sur plusieurs plateformes. [7]



FIGURE 2.7 – Flutter logo

Dart :

Dart est un langage orienté objet moderne apparu en 2010 à l'initiative de deux développeurs de Google : Lars Bak et Kasper Lund. À l'origine, ce projet avait comme objectif de remplacer JavaScript à moyen ou long terme, en raison des difficultés à faire évoluer ce dernier. Google a donc conçu Dart à partir de cette idée et la version 1.0 est sortie le 10 octobre 2011. [8]



FIGURE 2.8 – Dart logo

Node.Js :

Node.js est un environnement d'exécution JavaScript côté serveur, construit sur le moteur JavaScript V8 de Google. Il permet de créer des applications web évolutives, rapides et performantes, en utilisant une architecture orientée événements qui facilite la gestion des requêtes simultanées. [9]



FIGURE 2.9 – Node.JS logo

Express.Js :

Express.js est un framework d'applications web Node.js minimal et flexible qui offre une liste de fonctionnalités pour créer facilement des applications web et mobiles. Il simplifie le développement d'applications côté serveur grâce à une API intuitive pour le routage, le middleware et les utilitaires HTTP. [10]



FIGURE 2.10 – Express.JS logo

Javascript :

JavaScript est un langage de programmation principalement utilisé sur Internet, en complément de HTML et CSS. Il utilise des scripts pour créer du contenu dynamique. Il complète ainsi les deux autres langages de base du Web et peut stocker des valeurs, faire des opérations ou encore exécuter du code selon certains événements. C'est un langage orienté prototype, c'est-à-dire semblable à un langage orienté objet, mais sans classe. [11]



FIGURE 2.11 – Javascript logo

Python :

Python est un langage de programmation polyvalent, intuitif et puissant qui ouvre la porte à de nombreuses opportunités professionnelles dans le domaine du numérique. Facile à apprendre et à utiliser, ce langage de programmation qui permet de développer rapidement des applications dans des domaines variés tels que le développement web, l'analyse de données, l'intelligence artificielle ou l'automatisation.[12]



FIGURE 2.12 – Python logo

2.2.3 Structure générale du système

Cette partie expose l'architecture générale de l'application mobile, en distinguant deux vues complémentaires : une vue matérielle qui illustre les composants physiques et logiciels impliqués, et une vue fonctionnelle qui détaille l'organisation logique du système selon une architecture en couches.

2.2.3.1 Vue matérielle de l'application

- **Client Mobile (Frontend) :** L'application mobile, développée avec Flutter, sert d'interface pour les utilisateurs (Admin, PMO, Technicien, Client, Commercial). Elle permet de créer des tickets, consulter des contrats et interagir via des requêtes HTTPS avec le serveur. L'accès est sécurisé par JWT et adapté aux permissions de chaque rôle. La messagerie en temps réel et l'intégration de cartes (OpenStreetMap) améliorent l'expérience utilisateur.
- **Serveur d'application :** Basé sur Node.js et Express, il gère la logique métier et les règles de gestion. Il expose une REST API pour communiquer avec le client et la base de données. Des middlewares sécurisent les échanges. Il intègre des services externes (géolocalisation) et génère des rapports PDF/Word.
- **Serveur de base de données :** MongoDB stocke les données de l'application sous forme de documents JSON flexibles et scalables. Les collections (utilisateurs, tickets, contrats) gèrent des structures variées sans schéma rigide, adaptées aux besoins

évolutifs. L'intégration avec Node.js via Mongoose simplifie les opérations CRUD et assure la cohérence des données.

2.2.3.2 Vue fonctionnelle de l'application

- **Couche présentation** :L'interface utilisateur, développée en Flutter, offre des écrans adaptés à chaque rôle (Admin, Commercial, PMO, technicien, client). Elle permet de créer des tickets, consulter des contrats, et interagir via un chat intégré. Les données s'affichent dynamiquement (dashboard, calendrier) avec des mises à jour en temps réel. La communication avec le backend se fait via des appels API REST sécurisés par JWT.
- **Couche métier** :Le backend gère l'authentification, les permissions RBAC, et les règles métier. Il expose des API REST pour les opérations CRUD sur les contrats, tickets,..etc. Des middlewares valident les données et sécurisent les échanges (des fichiers). Il intègre aussi des services externes comme le mailing et la géolocalisation.
- **Couche données** :MongoDB stocke les documents structurés (utilisateurs, tickets, contrats...) avec des schémas flexibles. Les collections gèrent les relations via des références et des agrégations pour les statistiques. Les index optimisent les requêtes fréquentes (ex : tickets ouverts). La scalabilité horizontale assure une gestion efficace des données en croissance.

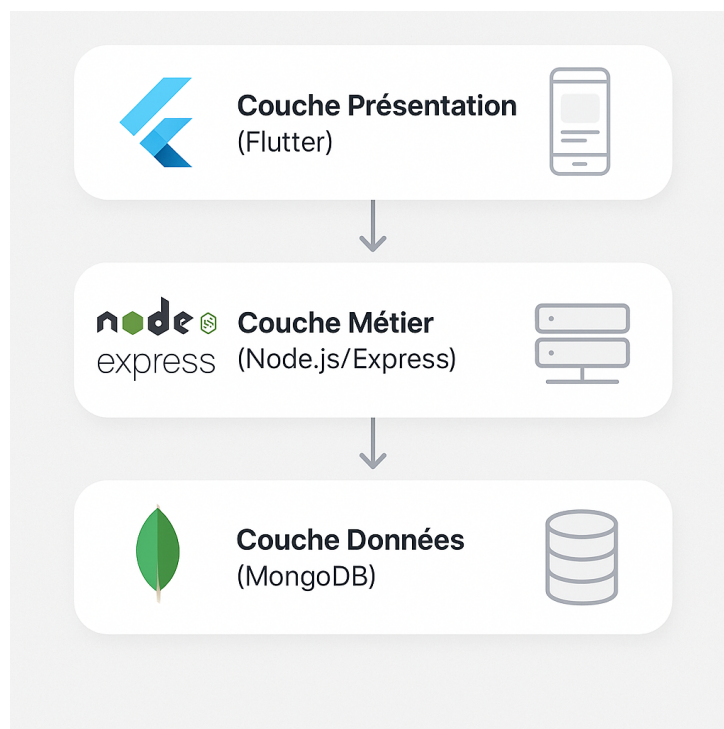


FIGURE 2.13 – Architecture en Trois Couches d'une Application Mobile Moderne

2.3 Mise en œuvre du « Sprint 0 »

2.3.1 Backlog du produit

Dans le contexte de Scrum, le Backlog Produit centralise et priorise les éléments à développer, selon les attributs :

- **Id** :Identifiant unique
- **User Story** : est une expression claire et concise d’une exigence utilisateur, formulée sous forme de besoin.
- **Story Points** :Estimation basée sur Fibonacci

[1] : Très facile [2] : Facile [3] : Moyenne [5] : dur [8] : Très dur

ID	User Story	Story Points
1	En tant qu'utilisateur, je veux m'authentifier.	3
2	En tant qu'utilisateur, je veux gérer mon profil et changer mon mot de passe.	2
3	En tant qu'administrateur, je veux accepter un utilisateur pour qu'il accède aux fonctionnalités dédiées.	3
4	En tant qu'administrateur, je veux désactiver temporairement un utilisateur sans le supprimer.	3
5	En tant qu'administrateur, je veux valider des utilisateurs.	2
6	En tant que technicien, je veux m'inscrire sur la plateforme.	3
7	En tant qu'utilisateur, je veux recevoir un mail après inscription pour confirmer mon enregistrement.	5
8	En tant qu'administrateur, je veux créer un dossier client pour structurer les contrats.	2
9	En tant qu'administrateur, je veux créer les comptes des Pmos, Clients et Commerciaux.	2
10	En tant qu'utilisateur, je veux recevoir un email d'acceptation ou refus	5
11	En tant qu'administrateur, je veux créer un dossier client pour structurer les contrats.	2
12	En tant que PMO, je veux consulter et modifier les dossiers des clients pour assurer le suivi.	2
13	En tant que commercial, je veux consulter les dossiers clients associés à mes contrats.	2
14	En tant qu'administrateur, je veux créer un contrat selon les informations d'un client.	3
15	En tant que PMO, je veux consulter et gérer les contrats sous les dossiers pour les adapter.	3
16	En tant que commercial, je veux consulter les contrats des clients.	2
17	En tant qu'administrateur, je veux attribuer des techniciens et un commercial à un contrat.	3
18	En tant qu'administrateur, je veux créer un site client avec localisation via OpenStreetMap et télécharger des fichiers pour un contrat.	3

ID	User Story	Story Points
19	En tant que PMO, je veux consulter et gérer les sites associés aux contrats.	3
20	En tant qu'administrateur, je veux créer des types de support (DELL, HP, etc.) pour catégoriser les équipements.	5
21	En tant qu'administrateur, je veux associer des équipements à un site.	3
22	En tant que PMO, je veux consulter et gérer les équipements rattachés aux sites.	3
23	En tant que PMO, je veux gérer les types de support.	3
24	En tant qu'administrateur, je veux créer un ticket.	3
25	En tant que client, je veux créer un ticket.	3
26	En tant que PMO, je veux consulter et gérer les tickets clients.	3
27	En tant que technicien, je veux assigner un ticket, si je suis un TeamLeader	3
28	En tant que technicien, je veux assigner un technicien a un ticket, si je suis un TeamLeader	3
29	En tant qu'utilisateur, je veux recevoir des notifications importantes.	5
30	En tant que technicien, je veux résoudre le problème d'un ticket.	3
31	En tant que client, je veux ouvrir une discussion avec le technicien qui est assigné à mon ticket pour résoudre mon problème.	5
32	En tant que technicien, je veux ouvrir une discussion avec le client afin de soumettre une solution.	5
33	En tant que technicien, je veux envoyer une solution chez le client pour valider la solution du ticket.	3
34	En tant que client, je veut confirmer la solution proposer par le technicien pour fermer la ticket avec succès.	3
35	En tant qu'administrateur, je veux accéder aux discussions entre le technicien et le client à fin de suivre l'avancement.	2
36	En tant qu'administrateur, je veux planifier des visites de maintenance préventive.	3
37	En tant que PMO, je veux planifier des visites de maintenance préventive.	3
38	En tant qu'utilisateur, je veux accéder au calendrier pour consulter les visites de maintenance préventive.	2
39	En tant que technicien, je veux accéder à la base de connaissances pour résoudre un ticket plus efficacement.	2
40	En tant que technicien, je veux accéder à un chabot intégré dans la base de connaissances.	2
41	En tant que technicien, je veux générer un rapport après une visite pour documenter l'intervention.	3
42	En tant qu'utilisateur, je veux consulter les indicateurs clés sur le tableau de bord.	3
43	En tant que technicien, je veux générer un rapport pour un ticket en PDF.	3

ID	User Story	Story Points
----	------------	--------------

TABLE 2.2 – Tableau récapitulatif des User Stories du projet OPM Mobile

2.3.2 Planification des sprints

L'utilisation des User Stories dans notre initiative représente une approche méthodique pour traduire les besoins métiers en fonctionnalités concrètes. Elles nous ont donné la possibilité d'organiser le développement de manière agile et centrée sur l'utilisateur. Chaque User Story a été soigneusement formulée afin de répondre à une exigence précise, selon le rôle de l'acteur concerné. Cette organisation a facilité la répartition des tâches, l'estimation des charges et la priorisation des livrables. Elle constitue ainsi un pilier fondamental dans la réussite et la fluidité de l'exécution du projet.

Sprint	User Story	Début	Fin
Sprint 0	#####	18-Feb-25	3-Mar-25
Sprint 1	1–13	4-Mar-25	24-Mar-25
Sprint 2	14–20	25-Mar-25	13-Apr-25
Sprint 3	21–35	14-Apr-25	3-May-25
Sprint 4	36–43	4-May-25	29-May-25

TABLE 2.3 – Planification des sprints

Conclusion Dans ce chapitre, nous avons présenté de manière structurée l'ensemble des éléments techniques nécessaires à la mise en place de notre application. Nous avons aussi précisé la structure globale du système en passant par son architecture, tant logique que physique. Ces fondations nous offriront la possibilité d'aborder avec méthode et efficacité la phase de développement. Le chapitre suivant sera consacré à la description des travaux réalisés lors du premier sprint, en mettant l'accent sur la conception et l'implémentation initiale.

Chapitre 3

Sprint 1

3.1 Introduction

Dans ce chapitre, nous exposons la totalité des tâches effectuées au cours du premier sprint. Nous commencerons par exposer le contenu du backlog spécifique à cette phase, puis nous décrirons la conception des différentes user stories retenues. Enfin, nous illustrerons le travail accompli à travers les interfaces utilisateurs développées durant cette période.

3.2 Objectifs du Sprint

Le Sprint 1, qui s'est déroulé du 11 au 29 mars 2025, a marqué le début du développement fonctionnel après la mise en place de l'environnement technique. Il visait à implémenter l'accès sécurisé, la gestion des utilisateurs et les premières opérations sur les dossiers clients. Durant ce sprint, j'ai travaillé sur le front-end avec Flutter, le back-end avec Node.js, et l'intégration avec MongoDB.

3.3 Backlog du sprint 1

ID	User Story	Story Points
1	En tant qu'utilisateur, je veux m'authentifier.	3
2	En tant qu'utilisateur, je veux gérer mon profil et changer mon mot de passe.	2
3	En tant qu'administrateur, je veux accepter un utilisateur pour qu'il accède aux fonctionnalités dédiées.	3
4	En tant qu'administrateur, je veux désactiver temporairement un utilisateur sans le supprimer.	3
5	En tant qu'administrateur, je veux valider des utilisateurs.	2
6	En tant que technicien, je veux m'inscrire sur la plateforme.	3
7	En tant qu'utilisateur, je veux recevoir un mail après inscription pour confirmer mon enregistrement.	5
8	En tant qu'administrateur, je veux créer un dossier client pour structurer les contrats.	2
9	En tant qu'administrateur, je veux créer les comptes des Pmos, Clients et Commerciaux.	2
10	En tant qu'utilisateur, je veux recevoir un email d'acceptation ou refus	5

11	En tant qu'administrateur, je veux créer un dossier client pour structurer les contrats.	2
12	En tant que PMO, je veux consulter et modifier les dossiers des clients pour assurer le suivi.	2
13	En tant que commercial, je veux consulter les dossiers clients associés à mes contrats.	2

TABLE 3.1 – Backlog du sprint 1

3.4 Diagramme d'activité

3.4.1 Diagramme d'activité d'user story N°1

Ce diagramme d'activité illustre le déroulement du processus d'authentification d'un utilisateur au sein de l'application. Il présente de manière séquentielle les différentes étapes, depuis la saisie des informations de connexion jusqu'à la validation ou le refus de l'accès, en fonction de la conformité des identifiants. Ce schéma permet ainsi de visualiser clairement la logique de traitement mise en œuvre pour sécuriser l'accès à la plateforme.

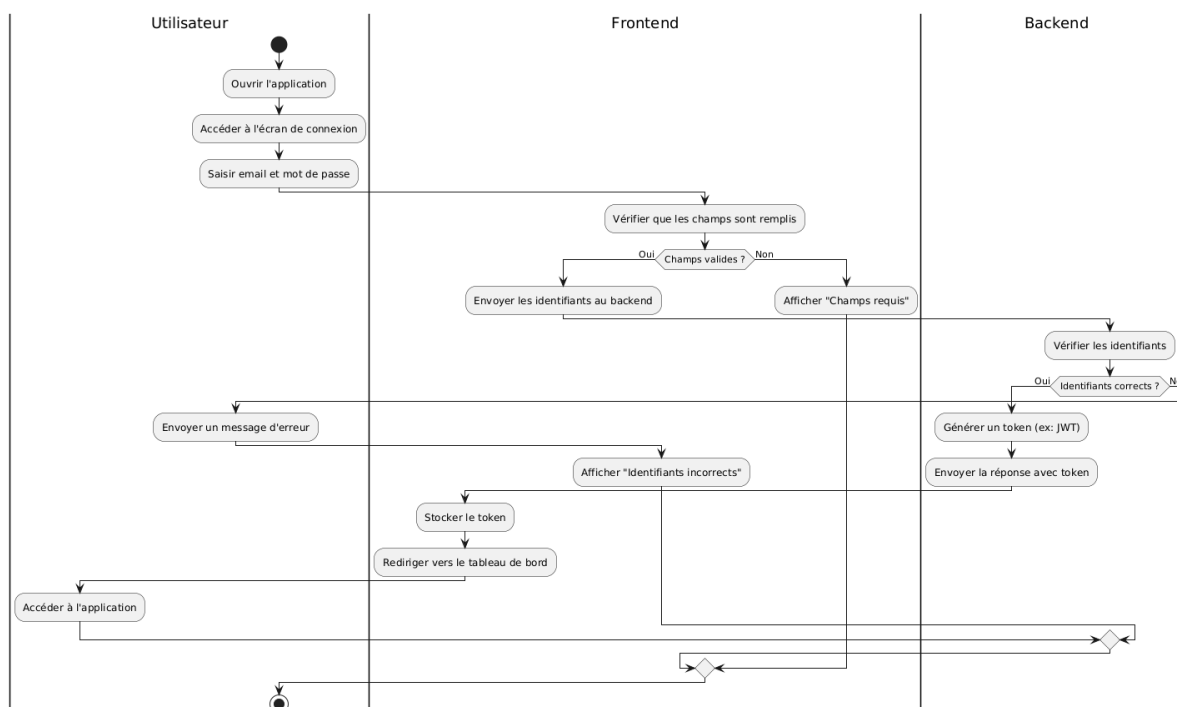


FIGURE 3.1 – Diagramme d'activité « Authentification d'utilisateur »

3.4.2 Diagramme d'activité d'user story N°8

Le diagramme présente les étapes de la création des dossiers des clients

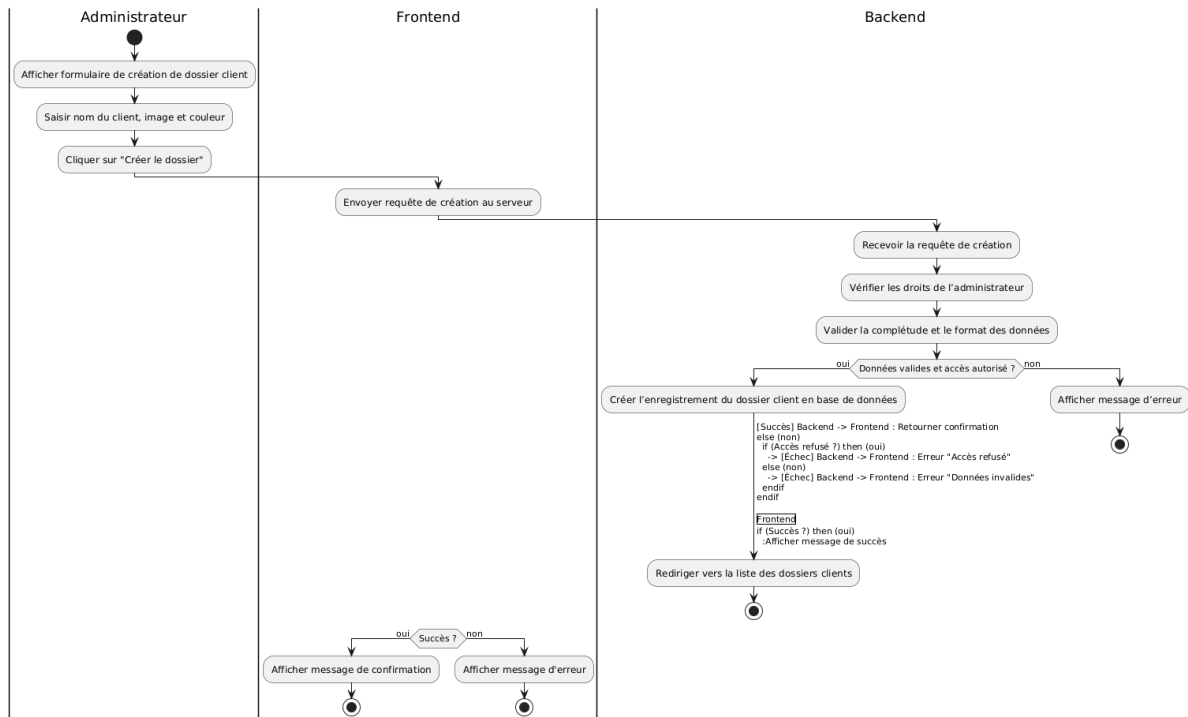


FIGURE 3.2 – Diagramme « Création des dossiers des clients »

3.5 Les Scénario

3.5.1 S  nario de user story N  1

Cette sc  nario pr  sente l'authentification des utilisateurs. Elle d  taille les conditions d'acc  s, le d  roulement de la connexion, aussi le traitement des erreurs en cas d'identifiants invalides.

Cas d'utilisation	En tant qu'utilisateur, je veux m'authentifier
Acteurs	Utilisateurs
Pr��condition	Tout acteur doit ��tre inscrit
Post-condition	Acteur authentifi��
Sc��nario nominal	L'acteur saisit son email et son mot de passe puis clique sur le bouton « Login ». Le syst��me v��rifie les donn��es et affiche automatiquement la premi��re page de l'application.
Sc��nario d'exception	Si l'email ou le mot de passe est incorrect, le syst��me informe l'acteur que ses donn��es sont invalides (compar��es �� celles enregistr��es dans la base de donn��es), puis affiche un message d'erreur.

TABLE 3.2 – Description de l'authentification d'utilisateurs

3.5.2 Scénario de user story N°8

Cette scénario décrit la création d'un dossier client par un administrateur qui doit être connecté. Dès que le système détecte un champ obligatoire vide, il va afficher un message d'erreur de création chez l'administrateur.

Cas d'utilisation	En tant qu'administrateur, je veux créer un dossier client pour structurer les contrats.
Acteurs	Administrateur
Précondition	Pour l'affectation de cette opération, l'administrateur doit être connecté
Post-condition	Création du dossier client effectuée
Scénario nominal	L'administrateur choisit un nom, une couleur et une image pour le dossier après communication avec le client.

TABLE 3.3 – Description de la création des dossiers des contrats pour les clients

3.6 Conception du Sprint

3.6.1 Diagramme de cas d'utilisation du sprint1

Ce diagramme de cas d'utilisation représente les échanges entre les différents types d'utilisateurs (Utilisateur, Administrateur, Technicien, PMO, Commercial) et le système. Il se compose en trois sections : gestion de compte, administration des utilisateurs et gestion des dossiers clients. utilisateur a accès uniquement aux fonctionnalités qui correspondent à son rôle spécifique.

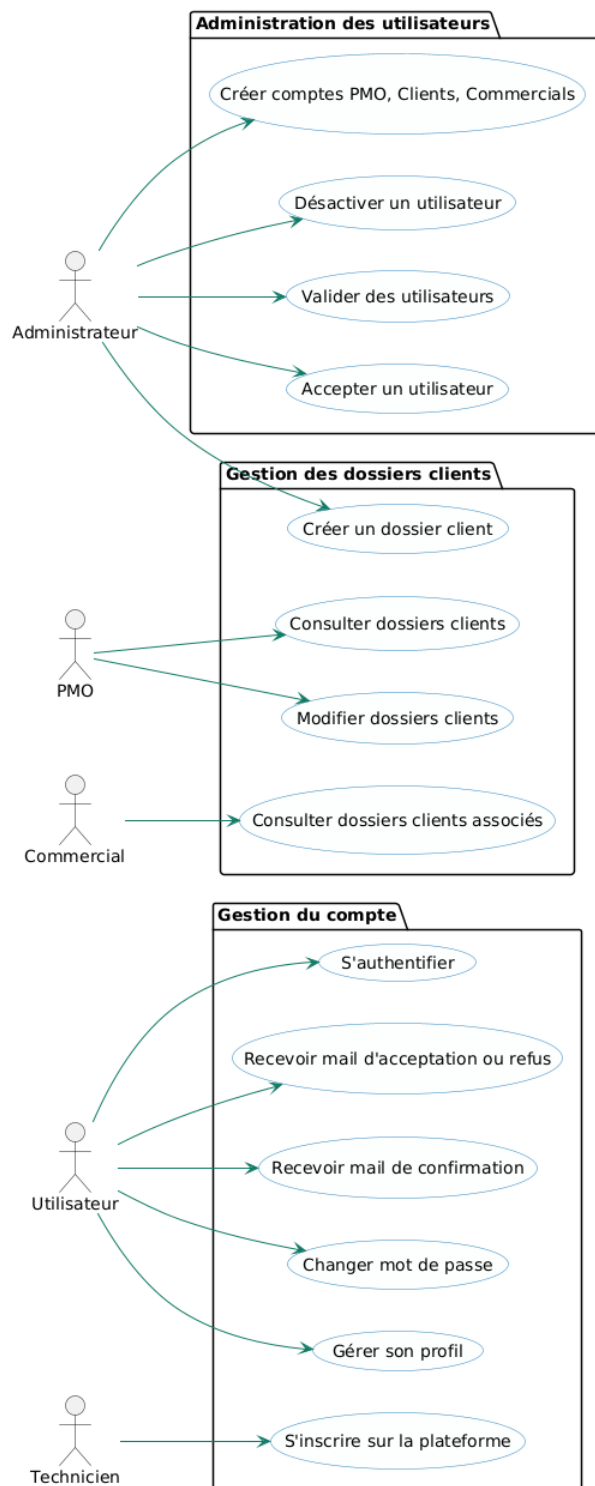
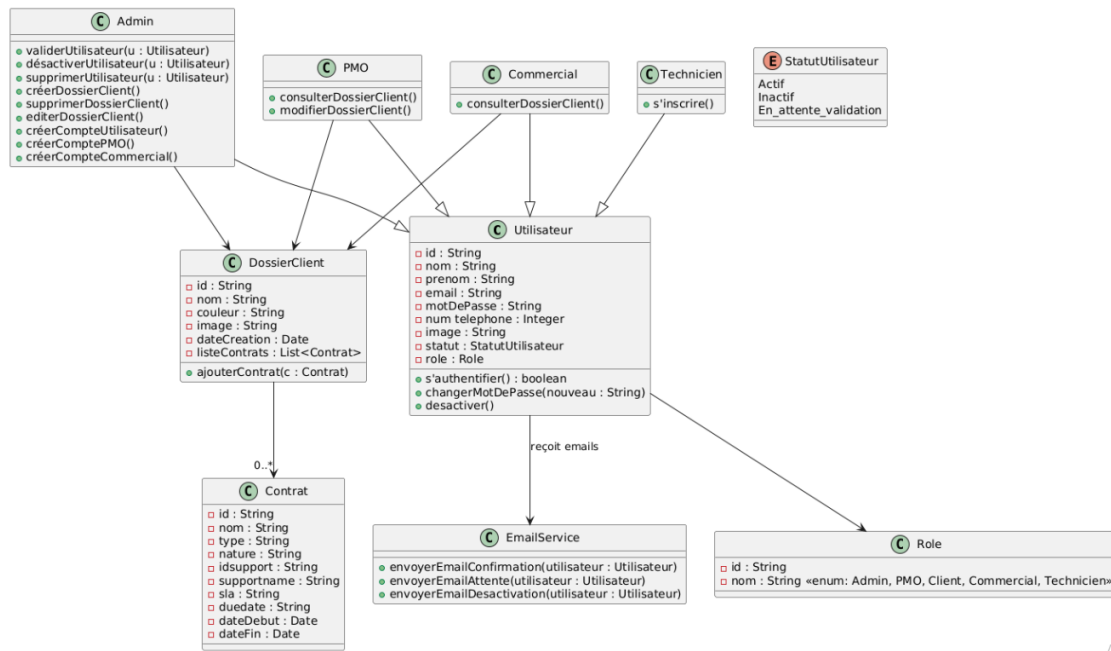


FIGURE 3.3 – Diagramme de cas d'utilisation du sprint 1

3.6.2 Diagramme de classe du sprint1

C'est une présentation du diagramme de classe du sprint 1 qui montre les classes et les interfaces utilisées pour la création de notre application



Activate Win

FIGURE 3.4 – Diagramme de classe du sprint 1

3.6.3 Vue dynamique du sprint 1

Ce diagramme illustre les interactions clés entre les intervenants (administrateur, technicien, PMO, commercial et client) et le système OPM Mobile lors du premier sprint. Il met en évidence les flux d'authentification sécurisée (JWT), la création centralisée des comptes par l'admin (PMO, commercial et client). Les notifications instantanées (emails, alertes) et la gestion des rôles (RBAC) y sont modélisées, reflétant les user stories prioritaires. Cette vue dynamique montre également les états des comptes (actif/en attente).

3.6.3.1 Diagramme de séquence objet

3.6.3.1.1 Diagramme de séquence objet de user story N°1

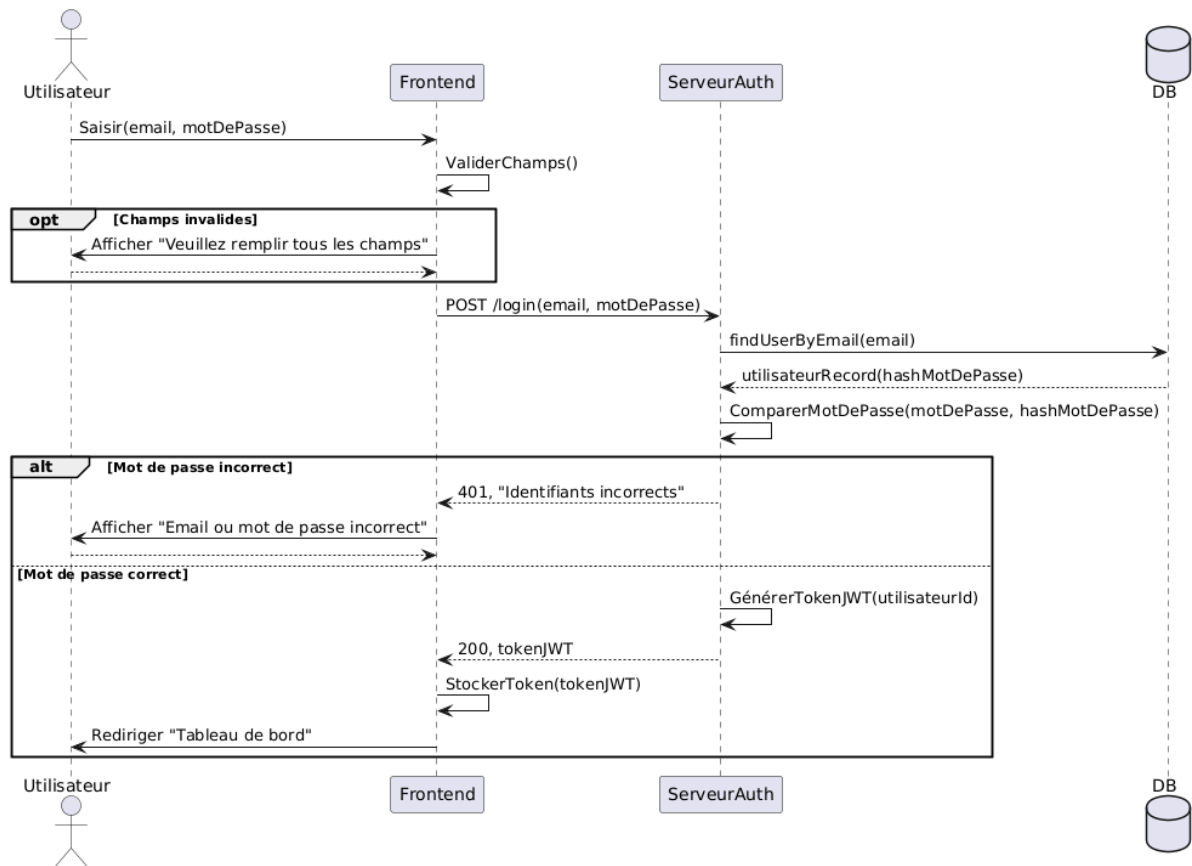


FIGURE 3.5 – Diagramme d'objet séquence «L'authentification»

3.6.3.1.2 Diagramme de séquence objet de user story N°8

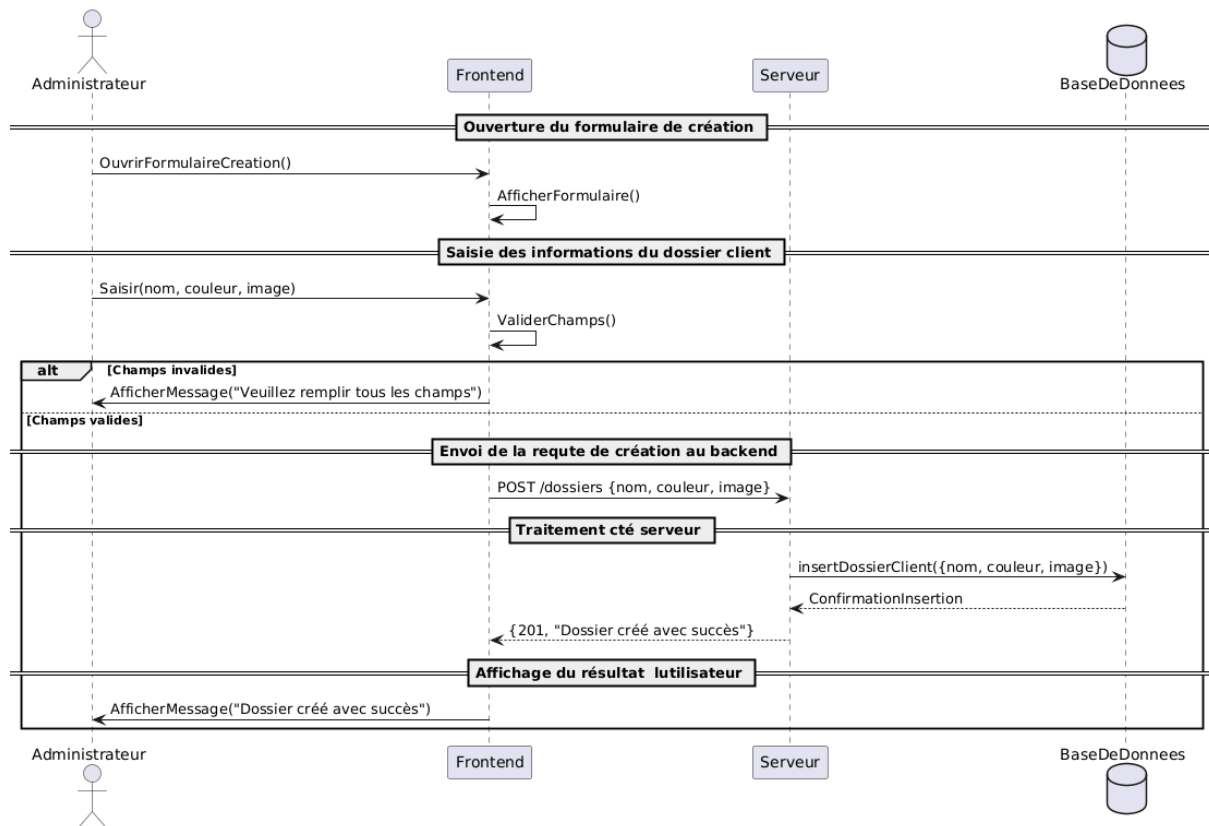


FIGURE 3.6 – Diagramme d'objet séquence «Création d'un dossiers des contrats»

3.7 Réalisation de sprint1

3.7.1 Interface de user story N°1

Cette interface décrit l'authentification de tout utilisateur va accéder à l'application

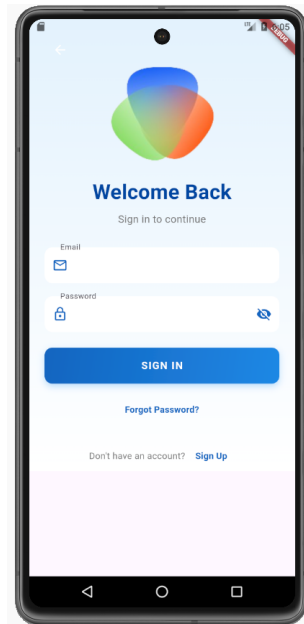


FIGURE 3.7 – Authentificationn

3.7.2 Interface de user story N°3 + N°4

Cette interface présente le déroulement de l'activation ou désactivation des comptes via l'administrateur.

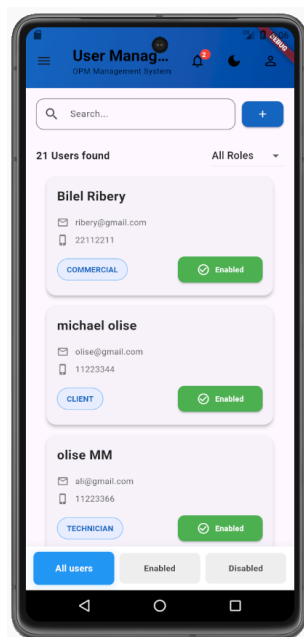


FIGURE 3.8 – L'activation et la desactivations des comptes utilisateurs

3.7.3 Interface de user story N°8

Cette interface présente la création de dossiers de contrats clients selon les besoins du client

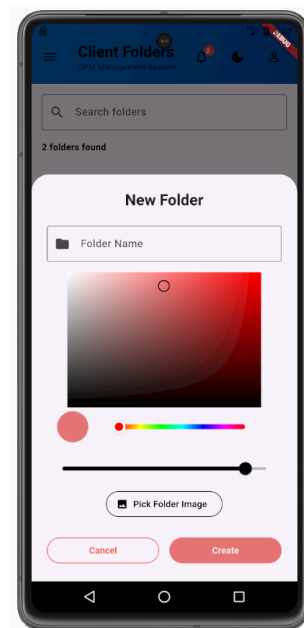


FIGURE 3.9 – Creation d'un dossier pour un client

Conclusion Ce chapitre est consacré à la conception et la mise en œuvre de la première sprint. La partie suivante sera consacrée au lancement détaillé du deuxième sprint

Chapitre 4

Sprint 2

4.1 Introduction

Dans ce chapitre, nous allons travailler sur le deuxième sprint. Nous commencerons par la présentation du back log du sprint, puis analyser et concevoir les user stories. Enfin, une présentation pour les interfaces.

4.2 Objectifs du Sprint 2

Ce deuxième sprint est consacré à l'implémentation des fonctionnalités liées à la gestion des dossiers client (pour le pmo), des contrats, des sites, ainsi que des types support. Donc ce sprint a un objectif de renforcer la structure organisationnelle du système en offrant aux différents rôles un accès ciblé a ces éléments essentiels à la gestion de projet

4.3 Backlog du sprint 2

ID	User Story	Story Points
14	En tant qu'administrateur, je veux créer un contrat selon les informations d'un client.	3
15	En tant que PMO, je veux consulter et gérer les contrats sous les dossiers pour les adapter.	3
16	En tant que commercial, je veux consulter les contrats des clients.	2
17	En tant qu'administrateur, je veux attribuer des techniciens et un commercial à un contrat.	3
18	En tant qu'administrateur, je veux créer un site client avec localisation via OpenStreetMap et télécharger des fichiers pour un contrat.	3
19	En tant que PMO, je veux consulter et gérer les sites associés aux contrats.	3
20	En tant qu'administrateur, je veux créer des types de support (DELL, HP, etc.) pour catégoriser les équipements.	5

TABLE 4.1 – Tableau du backlog du sprint 2

4.4 Diagramme d'activité

4.4.1 Diagramme d'activité d'user story N°14

Le diagramme ci-dessus illustre le processus complet suivi par l'administrateur pour créer un nouveau contrat. Le système présente d'abord le formulaire, l'administrateur saisit toutes les informations nécessaires, puis le système vérifie la validité des données. En cas d'échec, un message d'erreur est affiché et l'utilisateur peut corriger son formulaire (boucle de réessaie). En cas de succès, le contrat est enregistré et une confirmation est affichée.

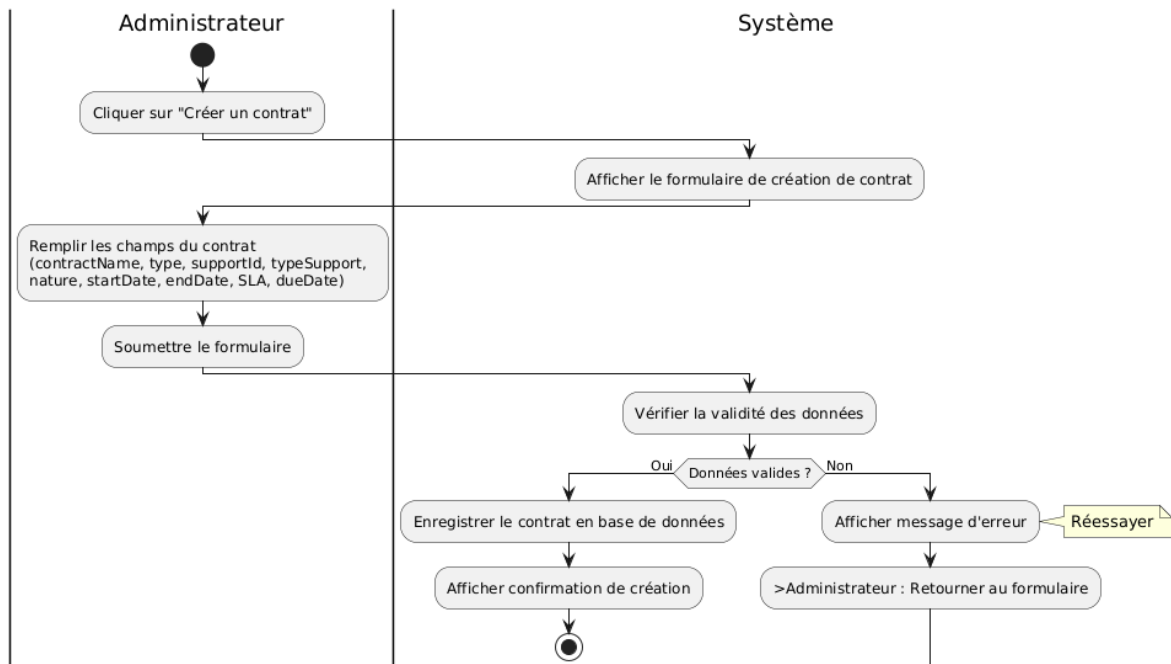


FIGURE 4.1 – Diagramme d'activité « Création du contrat pour un client »

4.4.2 Diagramme d'activité d'user story N°20

Le diagramme illustre l'administrateur qui ouvre le formulaire de création d'un type de support (nom, email, image) et soumet les données. Le système vérifie la validité des informations puis, si elles sont conformes, enregistre le nouveau type de support et affiche une confirmation. En cas d'erreur, un message est affiché et l'administrateur peut corriger et resoumettre le formulaire (boucle de réessai).

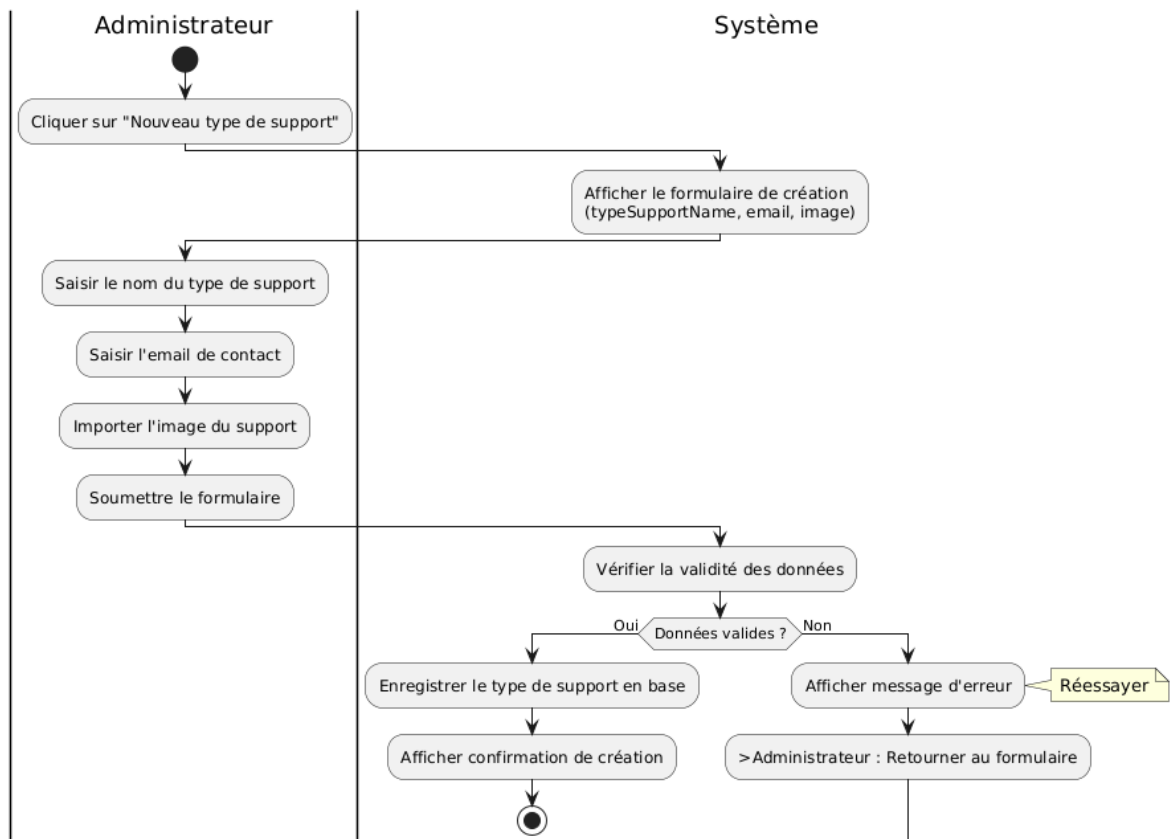


FIGURE 4.2 – Diagramme d’activité « Création du type support pour les contracts »

4.5 Les Scénario

4.5.1 Sénaire de user story N°14

Ce scénario décrit la création d’un contrat par un administrateur via un formulaire contenant des informations départ du client de cette contrat. Dès que un champ obligatoire est vide le système va afficher un message d’erreur chez l’admin

Cas d'utilisation	En tant qu'administrateur, je veux créer un contrat selon les informations d'un client.
Acteur	Administrateur
Précondition	L'administrateur doit être connecté
Post-condition	Création du contrat d'un client
Scénario nominal	L'administrateur accède à l'interface de gestion des contrats pour la création d'un nouveau contrat. Il remplit un formulaire en saisissant toutes les informations requises, telles que le nom du contrat, le type, le support avec un Id, la nature, les dates (début, fin, échéance) ainsi que les paramètres SLA. Une fois le formulaire complété, il le soumet pour validation. Enfin, le système vérifie la conformité des données saisies, enregistre le contrat en base de données puis un message s'affiche confirmant le succès de l'opération.
Scénario d'exception	L'administrateur remplit le formulaire de création de contrat mais oublie de saisir certaines informations obligatoires (nom du contrat, type, etc.). Alors, le système détecte les champs manquants ou invalides, bloque le processus de création et affiche un message d'erreur précisant les éléments à corriger. L'administrateur doit alors compléter ou rectifier les données avant de pouvoir valider à nouveau le formulaire.

TABLE 4.2 – Description textuelle de la création d'un contrat

4.5.2 Scénario de user story N°18

Ce scénario décrit l'ajout d'un site à un contrat avec géolocalisation via OpenStreet-Map avec le téléchargement de fichiers. Si le système détecte un erreur de localisation ou de format de fichier, il bloque que ce la création du site ou le téléchargement des fichiers.

Cas d'utilisation	En tant qu'administrateur, je veux créer un site client avec localisation via OpenStreetMap avec le téléchargement des fichiers pour un contrat.
Acteur	Administrateur
Précondition	L'administrateur doit être connecté
Post-condition	L'ajout des sites et fichiers pour un contrat
Scénario nominal	L'administrateur accède à la section « Sites » d'un contrat donné, puis clique sur « Ajouter un site ». Le système affiche un formulaire de création dans lequel il saisit les informations du site (nom et adresse) et utilise l'outil de localisation intégré basé sur OpenStreetMap pour sélectionner précisément l'emplacement géographique. Ensuite, il téléverse les fichiers nécessaires associés au site (contrat signé, escalation matrix, etc.). Dès que les données sont vérifiées, il valide le formulaire. Enfin, le système enregistre le site et les fichiers liés dans la base de données, puis affiche un message confirmant la création réussie.
Scénario d'exception	L'administrateur tente de créer un nouveau site client, mais omet de sélectionner un emplacement sur la carte OpenStreetMap ou téléverse un fichier dans un format non supporté. Lors de la validation du formulaire, le système détecte ces erreurs, empêche la soumission et affiche un message d'erreur précisant les éléments incorrects ou manquants. L'administrateur doit alors corriger les données, par exemple en choisissant une localisation valide sur la carte et en téléversant un fichier conforme, avant de pouvoir finaliser la création du site.

TABLE 4.3 – Description textuelle de l'ajout des sites et fichiers pour un contrat

4.5.3 Scénario de user story N°20

Cette scénario décrit la création de types de support par un administrateur connecté. Elle présente les préconditions, le déroulement normal, et les exceptions liées à cette fonctionnalité.

Cas d'utilisation	En tant qu'administrateur, je veux créer des types de support (DELL, HP, etc.).
Acteur	Administrateur
Précondition	L'administrateur doit être connecté
Post-condition	La création de types de support pour l'utiliser dans les contrats
Scénario nominal	L'administrateur accède à la section « Types de support » depuis le tableau de bord. Il clique sur « Ajouter un type de support » pour ouvrir le formulaire de création. Il saisit le nom du type de support (ex. : DELL ou HP), renseigne l'adresse email de contact du fournisseur et téléverse une image représentative. Après vérification des informations, il valide le formulaire. Le système enregistre alors le nouveau type de support dans la base de données et affiche un message de confirmation. Ce type de support pourra ensuite être utilisé lors de la création ou de la modification des contrats.
Scénario d'exception	Si l'administrateur tente de créer un type de support déjà existant (ex : "DELL"), le système détecte le doublon, bloque l'enregistrement et affiche un message d'erreur clair ("Ce type de support existe déjà"). L'administrateur doit alors modifier le nom ou annuler l'opération.

TABLE 4.4 – Description textuelle de la Création des types de support

4.6 Conception du Sprint 2

4.6.1 Diagramme de cas d'utilisation du sprint 2

Ce diagramme de cas d'utilisation présente les fonctionnalités du sprint 2, axées sur la gestion des contrats, des sites clients et des types de support. Les acteurs principaux (Administrateur, PMO, Commercial) interagissent avec le système selon leurs rôles assignés. Chaque fonctionnalité est regroupée pour illustrer une structure modulaire et claire du système.

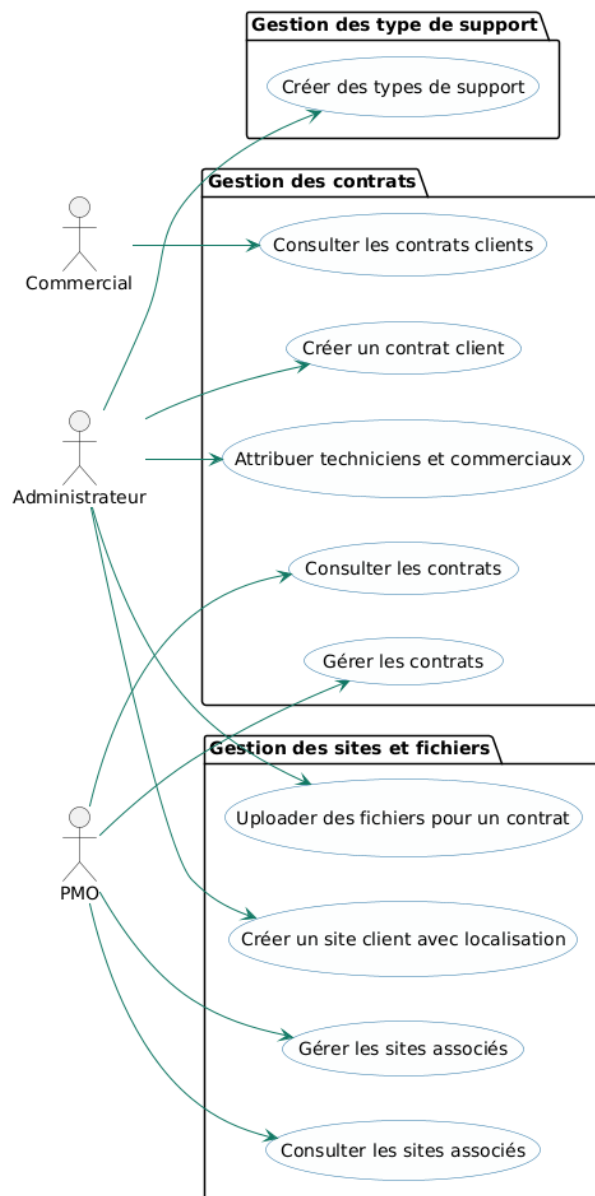


FIGURE 4.3 – Diagramme de cas d'utilisation du sprint 2

4.6.2 Diagramme de classe du sprint 2

Une présentation du diagramme de classes de conception du sprint 2 :

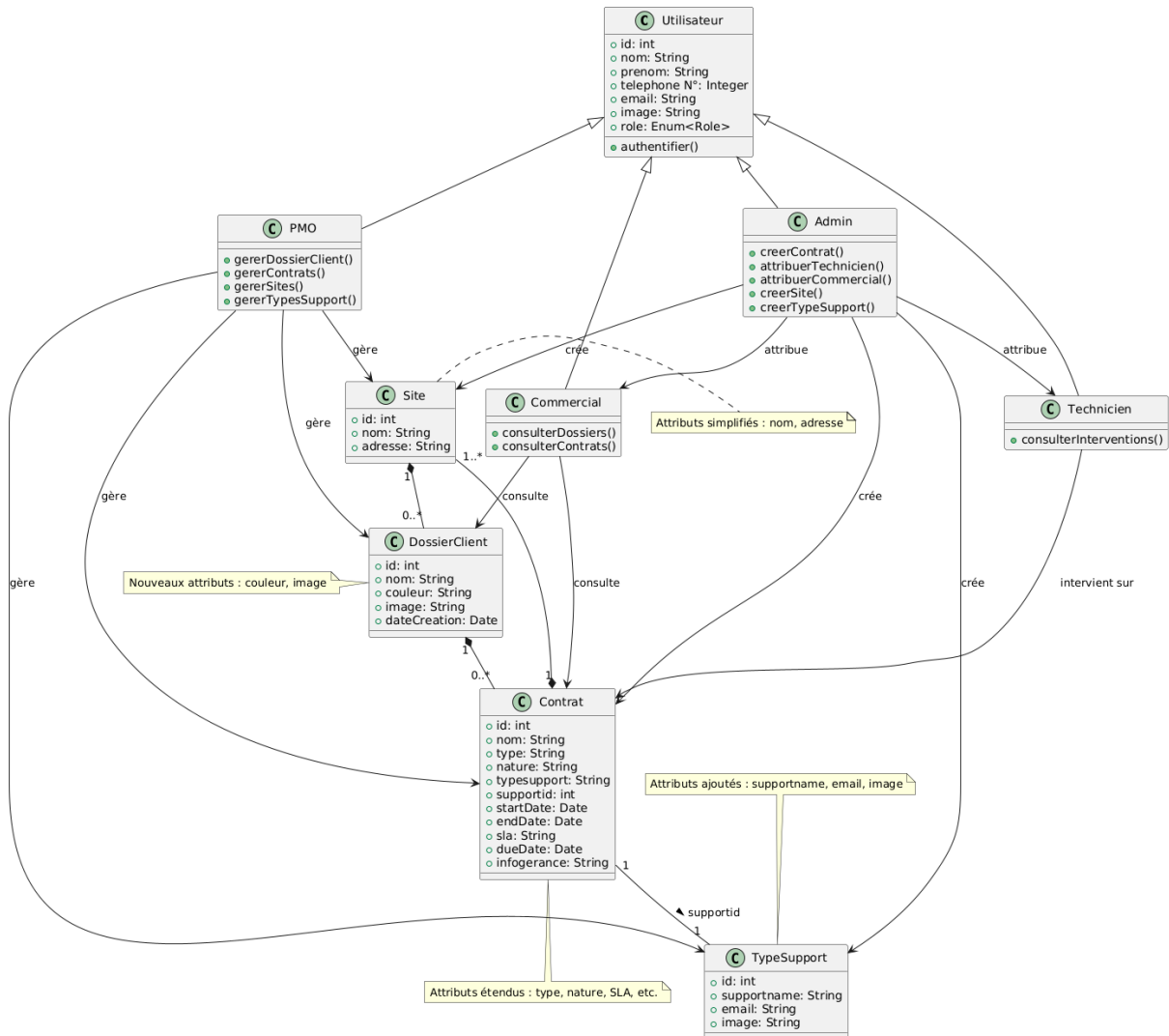


FIGURE 4.4 – Diagramme de classe montre la conception du sprint 2

4.6.3 Vue dynamique du sprint 2

4.6.3.1 Diagramme de séquence objet N°14 et N°18

User story N°14

L'administrateur crée un contrat personnalisé en saisissant les informations client (nom, type, dates, etc) via une interface sécurisée. Le système valide les données et les enregistre en base pour une gestion centralisée.

User story N°18

L'administrateur ajoute un site client en intégrant une localisation via OpenStreetMap et télécharge des fichiers associés (plans, documents). Les coordonnées et fichiers sont stockés en base et liés au contrat correspondant.

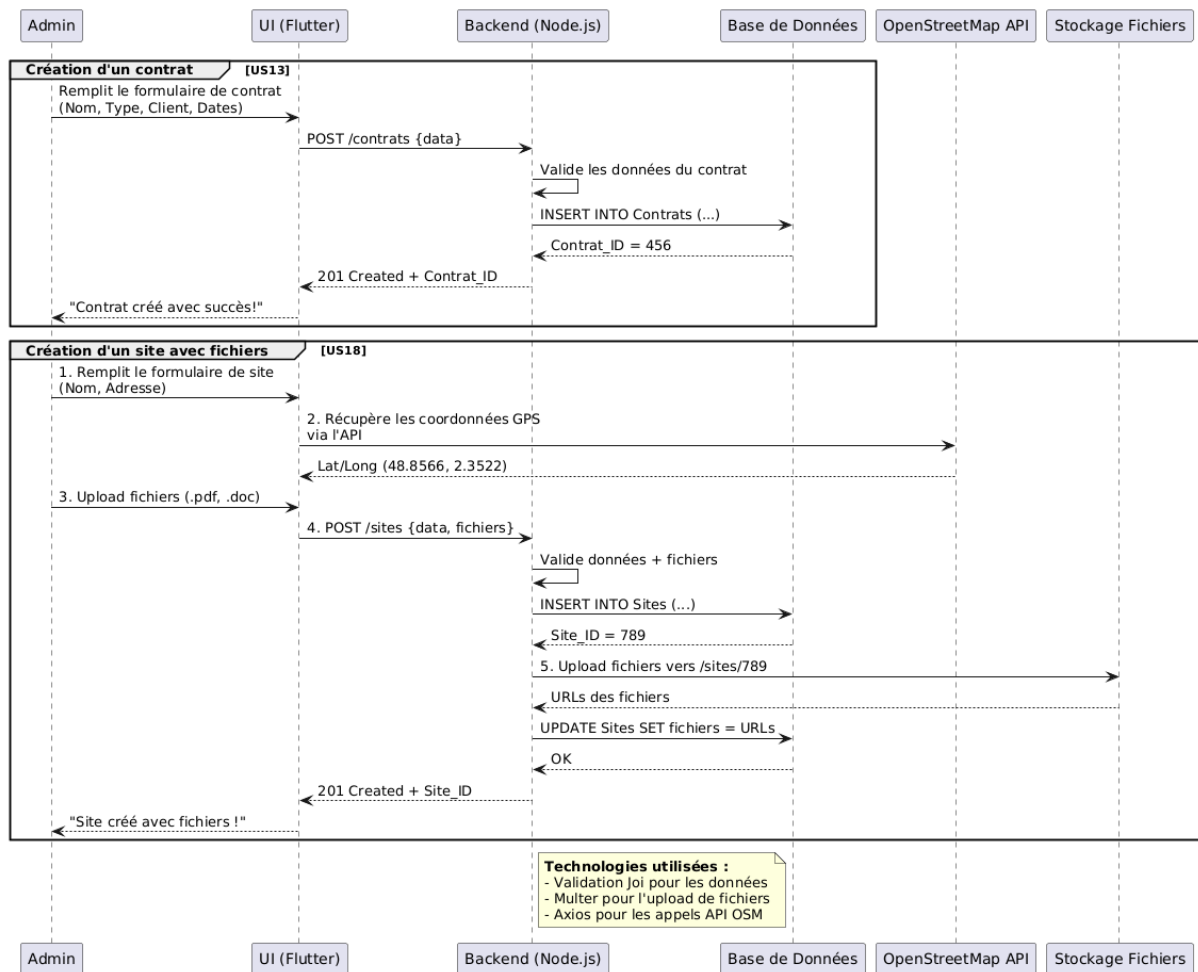


FIGURE 4.5 – Diagramme de séquence objet «création d'un contrat avec l'ajout des sites et fichiers pour cette contrat»

4.6.3.2 Diagramme de séquence objet de user story N°20

Ce diagramme illustre le processus de génération d'un type de support (ex : "DELL PowerEdge") par un administrateur dans l'application OPM Mobile. L'administrateur saisit les informations requises (nom et image) via une interface Flutter, déclenchant une requête API vers le backend Node.js. Ce dernier valide d'abord le format des données (avec Joi) et vérifie l'unicité du nom en interrogeant la base de données. Si le nom est unique, le type de support est enregistré en base, et une confirmation s'affiche à l'administrateur. En cas de doublon, une erreur 409 est renvoyée, invitant l'utilisateur à modifier le nom. Le diagramme met en évidence l'intégration fluide entre le frontend, le backend, et la base de données, ainsi que la gestion robuste des erreurs pour assurer l'intégrité des données.

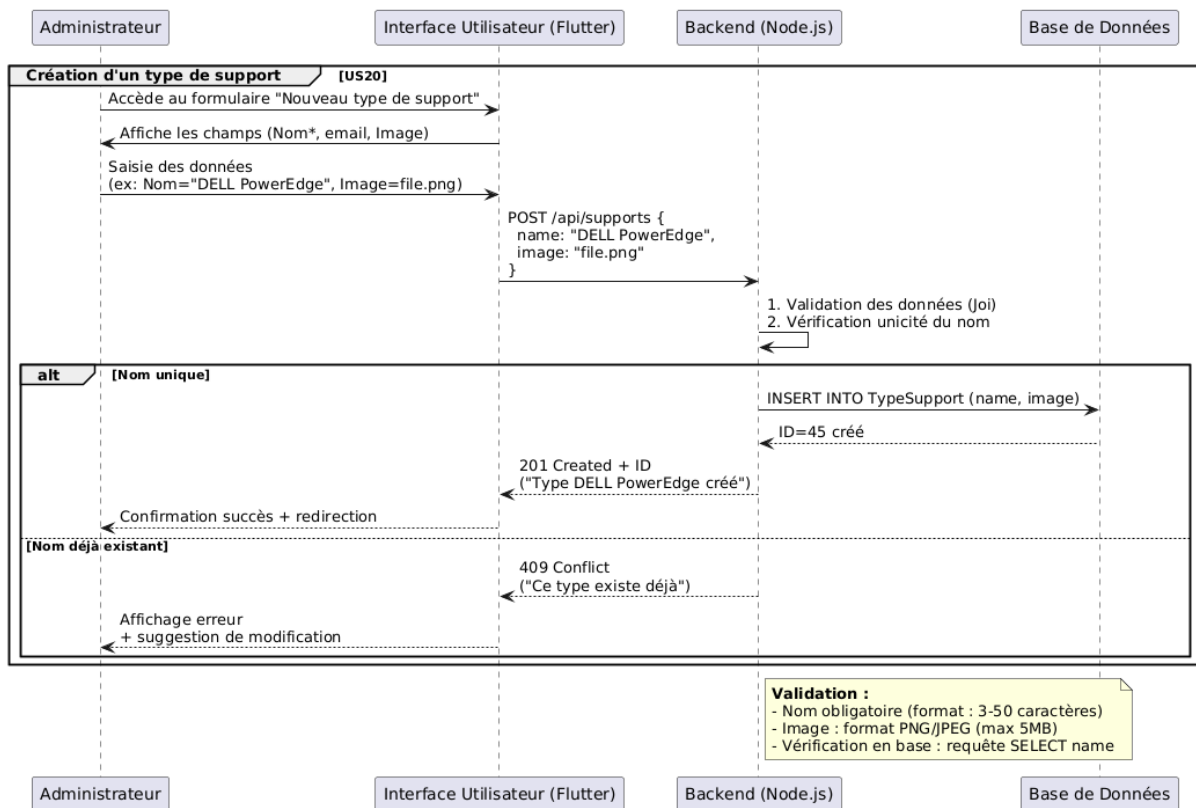


FIGURE 4.6 – Diagramme de séquence objet «création d'un type de support»

4.7 Réalisation de sprint1

4.7.1 Interface de user story N°14

Cette interface décrit la création d'un contrat du n'importe quel client à l'aide d'un formulaire qui contient des champs obligatoires. Dès que le système détecte un champs vides il va affiche un message d'erreur pour l'administrateur.

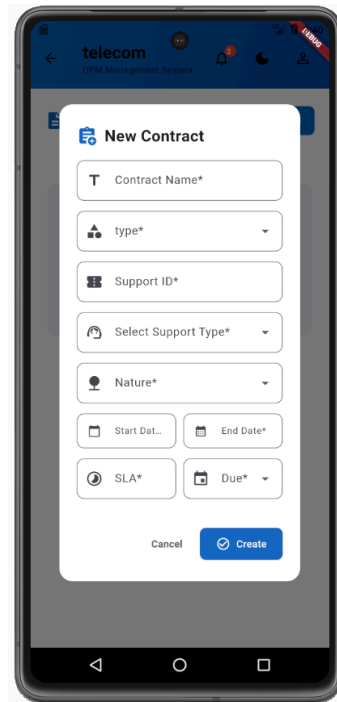


FIGURE 4.7 – Interface de user story N°14

4.7.2 Interface de user story N°18

Cette interface décrit l'ajout d'un ou des sites pour un contract. Dès que le système détecte un erreur de localisation ou champ vide il va afficher un message d'erreur pour l'administrateur.

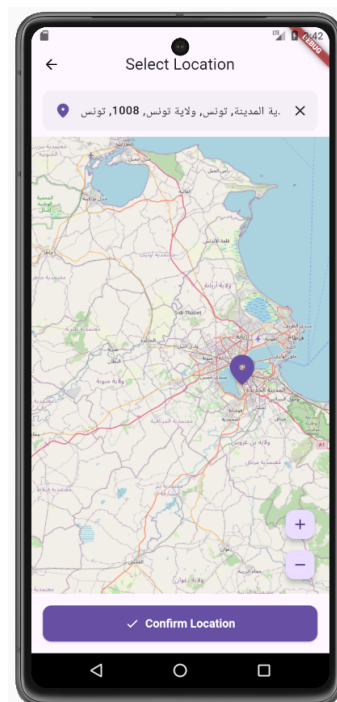


FIGURE 4.8 – Interface de user story N°18

4.7.3 Interface de user story N°20

Cette interface décrit La création d'un type de support via un administrateur. Dès que le système détecte un champ vide il va afficher un message d'erreur de création vers l'administrateur.

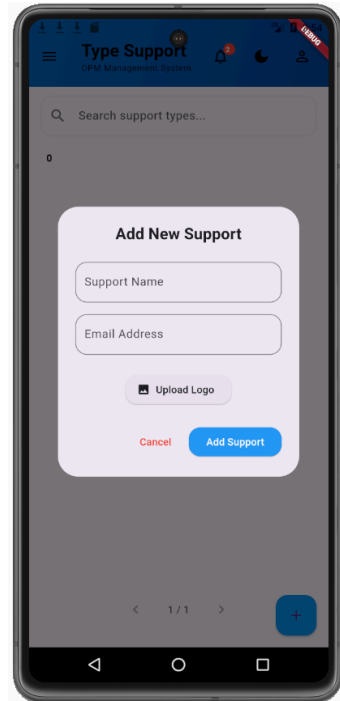


FIGURE 4.9 – Interface de user story N°20

ConclusionCe sprint a permis de finaliser la gestion centralisée des dossiers clients, contrats, sites et types de support,tout en incorporant des fonctionnalités clés comme la géolocalisation et le téléchargement de fichiers. Les tests unitaires et utilisateurs ont validé la robustesse des mécanismes d'attribution de rôles et de validation des données. Ces réalisations constituent une base solide pour aborder le Sprint 3, axé sur la gestion des tickets et du chat en temps réel.

Chapitre 5

Sprint 3

5.1 Introduction

Dans ce chapitre, nous allons exposer le travail réalisé durant le troisième sprint. Pour ce faire, nous commençons tout d'abord par la présentation du backlog du sprint, suivi par une phase d'analyse et de conception des user stories identifiés et nous présentons par la suite, les interfaces homme-machine réalisées.

5.2 Objectifs du Sprint 3

Ce sprint est élaboré pour offrir aux clients de consulter les services disponibles ainsi que leurs prestataires associés et de permettre également aux clients de gérer leurs listes des favoris. En outre, il devrait donner aux prestataires la possibilité de créer, consulter et modifier leurs comptes.

5.3 Backlog du sprint 3

ID	User Story	Story Points
21	En tant qu'administrateur, je veux créer des équipements à un site.	3
22	En tant que PMO, je veux consulter et gérer les équipements rattachés aux sites.	3
23	En tant que PMO, je veux gérer les types de support.	3
24	En tant qu'administrateur, je veux créer un ticket.	3
25	En tant que client, je veux créer un ticket.	3
26	En tant que PMO, je veux consulter et gérer les tickets clients.	3
27	En tant que technicien, je veux assigner un ticket, si je suis un TeamLeader	3
28	En tant que technicien, je veux assigner un technicien a un ticket, si je suis un TeamLeader	3
29	En tant qu'utilisateur, je veux recevoir des notifications importantes.	5
30	En tant que technicien, je veux résoudre le problème d'un ticket.	3
31	En tant que client, je veux ouvrir une discussion avec le technicien qui est assigné à mon ticket pour résoudre mon problème.	5
32	En tant que technicien, je veux ouvrir une discussion avec le client afin de soumettre une solution.	5

33	En tant que technicien, je veux envoyer une solution chez le client pour valider la solution du ticket.	3
34	En tant que client, je veux confirmer la solution proposer par le technicien pour fermer la ticket avec succès.	3
35	En tant qu'administrateur, je veux accéder aux discussions entre le technicien et le client à fin de suivre l'avancement.	2

TABLE 5.1 – Backlog du sprint 3

5.4 Diagramme d'activité

5.4.1 Diagramme d'activité d'user story N°21

Le diagramme représente l'administrateur qui ouvre le formulaire d'association d'un équipement à un site, informe le type et les informations de l'équipement, puis soumet. Le système valide les données :

- Si elles sont valides, l'équipement est associé au site et une confirmation est affichée.
- En cas d'erreur de saisie, un message d'erreur est affiché et l'administrateur peut corriger et resoumettre (boucle de réessai).

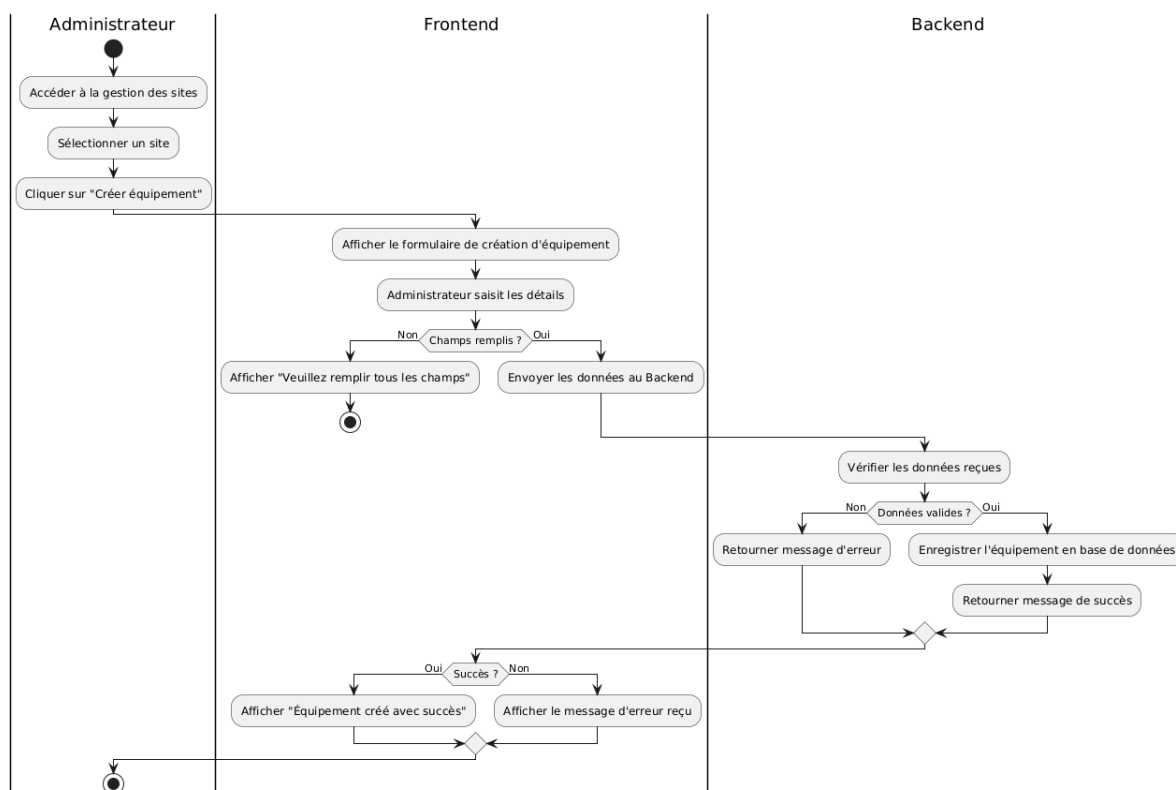


FIGURE 5.1 – Diagramme d'activité «la création d'un équipement»

5.4.2 Diagrammes d'activités de user story N°24

Le diagramme représente le client qui ouvre le formulaire de création d'un ticket (titre, description, priorité et, éventuellement, pièce jointe), saisie les champs, puis soumet. Le système vérifie les informations :

- Si elles sont valides, il crée le ticket et affiche un message de confirmation.
- Sinon, il affiche un message d'erreur et le client peut corriger et resoumettre le formulaire.

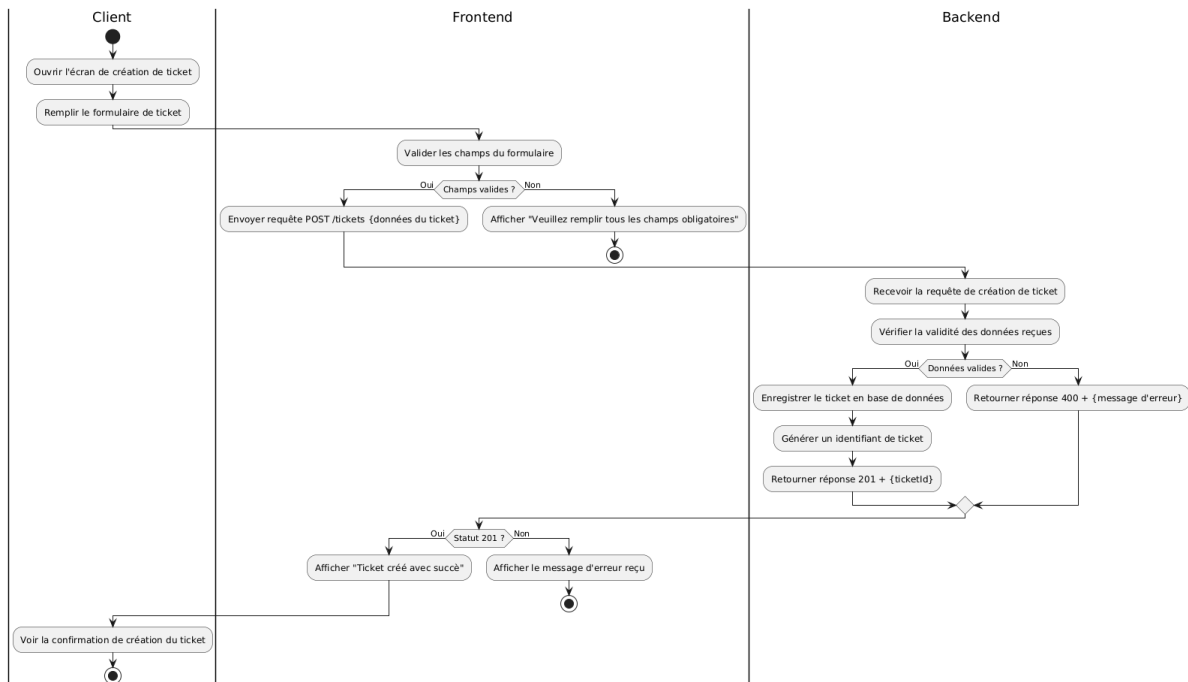


FIGURE 5.2 – Diagramme d'activité «la création d'un ticket»

5.4.3 Diagrammes d'activités de user story N°28

Le diagramme présente le système qui détecte un événement important, génère une notification et l'envoyer à l'utilisateur. L'utilisateur peut consulter la notification, puis soit l'archiver, soit la supprimer.

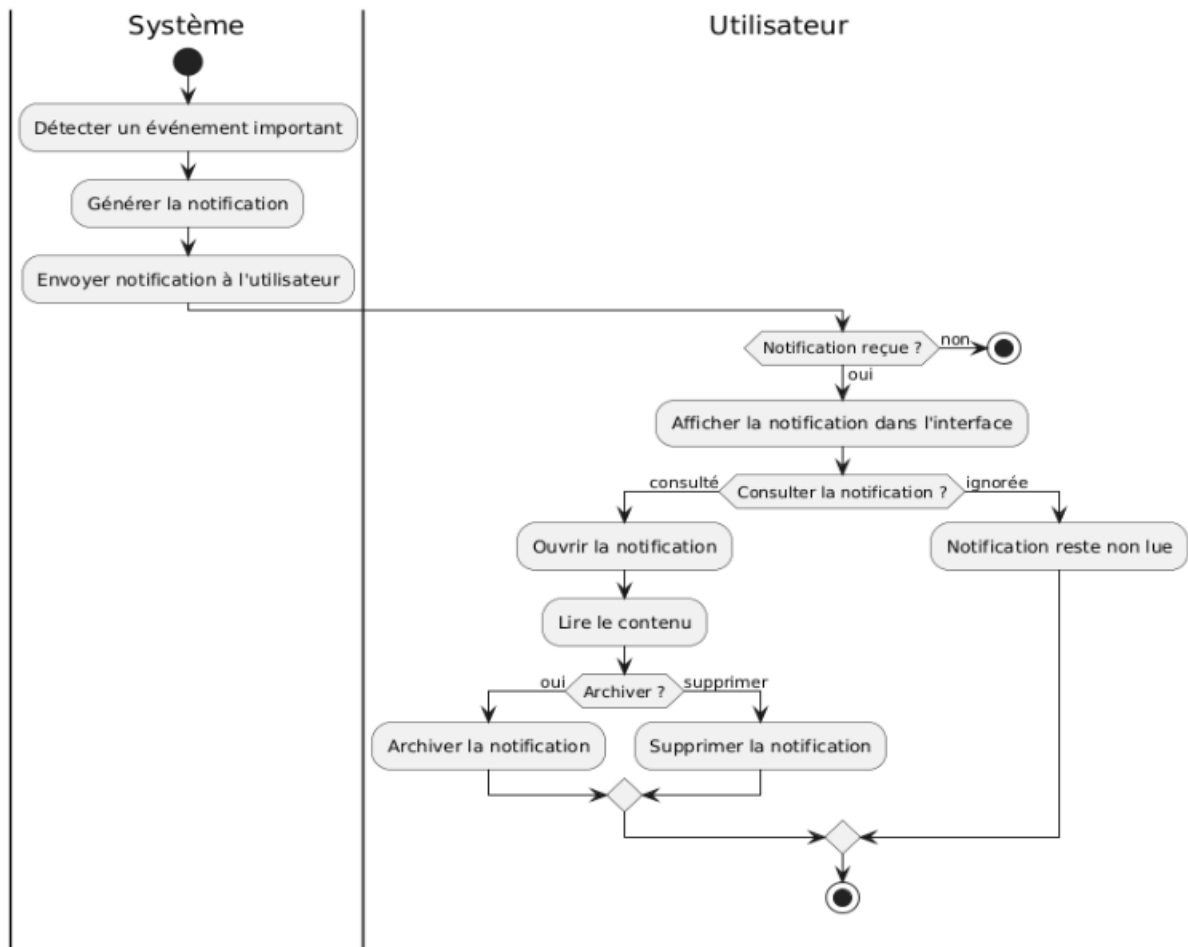


FIGURE 5.3 – Diagramme d’activité «Réception des notifications chez les utilisateurs»

5.4.4 Diagrammes d’activités de user story N°30 et N°31

Lorsqu’un ticket est attribué à un technicien, une session de discussion liée à ce ticket est automatiquement créée et le client peut l’accéder depuis son espace ticket pour poser des questions et . De même, le technicien peut y accéder pour répondre ou proposer une solution. Une fois la solution proposée, le client peut la consulter et la confirmer.

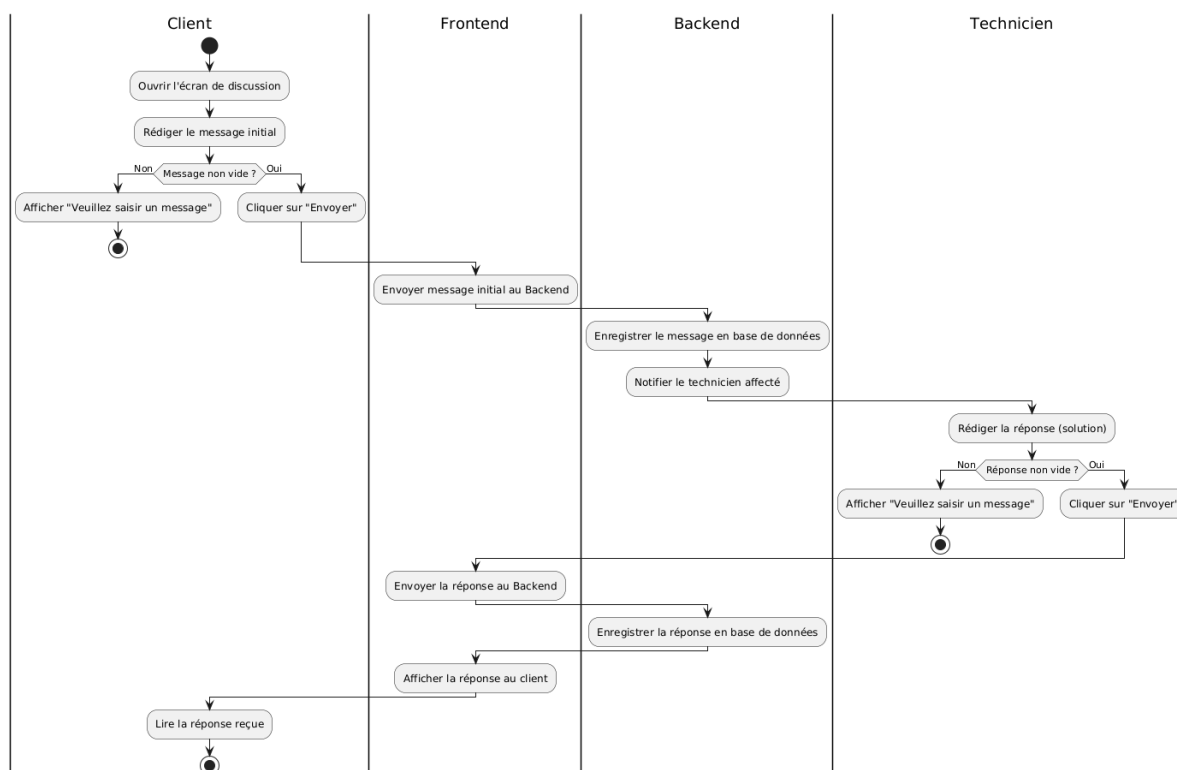


FIGURE 5.4 – Diagramme d'activité «résolution d'un ticket»

5.5 Les Scénario

5.5.1 Sénario de user story N°21

Ce scénario décrit l'ajout d'un équipement (soit matériel ou logiciel) à un site par un admin. Si les données sont invalides le système sera affiché un message d'erreur chez l'administrateur

Cas d'utilisation	En tant qu'administrateur, je veux créer des équipements (Soft/Hard) à un site.
Acteur	Administrateur
Précondition	L'administrateur est connecté et un site existe déjà dans le système.
Post-condition	L'équipement est associé au site et existe dans la liste.
Scénario nominal	L'administrateur accède à la section « Sites », sélectionne un site, clique sur « le bouton d'ajout ». Le système affiche un formulaire. L'administrateur remplit les champs nécessaires d'un équipement. Le système valide, enregistre l'association et affiche « Équipement associé avec succès ».
Scénario d'exception	Si un champ obligatoire est manquant ou invalide, le système empêche la soumission et affiche « Veuillez renseigner tous les champs obligatoires ». L'administrateur corrige puis resoumet.

TABLE 5.2 – Description textuelle de la création d'un équipement

5.5.2 Scénario de user story N°24

Ce scénario décrit la création d'un ticket par un client à l'aide d'un formulaire. Si les champs obligatoires du formulaire sont vides, le système va affiché un message d'erreurs chez le client.

Cas d'utilisation	En tant que client, je veux créer un ticket. Acteur Client
Acteur	Client
Précondition	Le client est authentifié et accède à la rubrique « Tickets ».
Post-condition	Le ticket est créé avec un identifiant unique et apparaît dans la liste.
Scénario nominal	Le client clique sur « Nouveau ticket », remplit titre, description, equipment, télécharger un fichier(optionnel), puis soumet. Le système vérifie, crée le ticket et affiche « Ticket créé avec succès » avec le numéro.
Scénario d'exception	Si titre ou description est vide, le système bloque la soumission et affiche « Tous les champs marqués d'un * sont obligatoires ». Le client complète puis soumet à nouveau.

TABLE 5.3 – Description textuelle de la création d'un ticket

5.5.3 Scénario de user story N°32

Ce scénario décrit la résolution d'un ticket à l'aide d'une interaction avec un technicien et client. Une solution sera enregistré dès que le client la validée.

Cas d'utilisation	En tant que technicien, je veux envoyer une solution chez le client pour valider la solution du ticket.
Acteur	Technicien
Précondition	Le technicien est connecté et un ticket lui est assigné.
Post-condition	Le ticket passe au statut « Résolu » avec historique de résolution.
Scénario nominal	Le technicien accède à ses tickets, choisit un ticket assigné pour lui et clique sur « Take action ». Une icône dédiée au chat sera mise à disposition, permettant au technicien d'engager une conversation avec le client afin d'identifier et résoudre le problème. Une fois la solution validée par le client, le technicien la saisira dans le champ prévu à cet effet et l'enregistrera pour clôturer officiellement le ticket.
Scénario d'exception	Si le commentaire est vide, le système empêche la validation et affiche « Le commentaire de résolution est requis ». Le technicien complète et resoumet.

TABLE 5.4 – Description textuelle sur la résolution d'un ticket

5.6 Conception du Sprint 3

5.6.1 Diagramme de cas d'utilisation du sprint 3

Ce diagramme englobe les cas d'utilisation du sprint 3, qui se concentrent sur la gestion des équipements, la résolution des tickets et la communication entre les utilisateurs. Il souligne les interactions spécifiques entre les rôles (Administrateur, PMO, Technicien, Client, Utilisateur) et le système. Les fonctionnalités sont organisées en modules distincts.

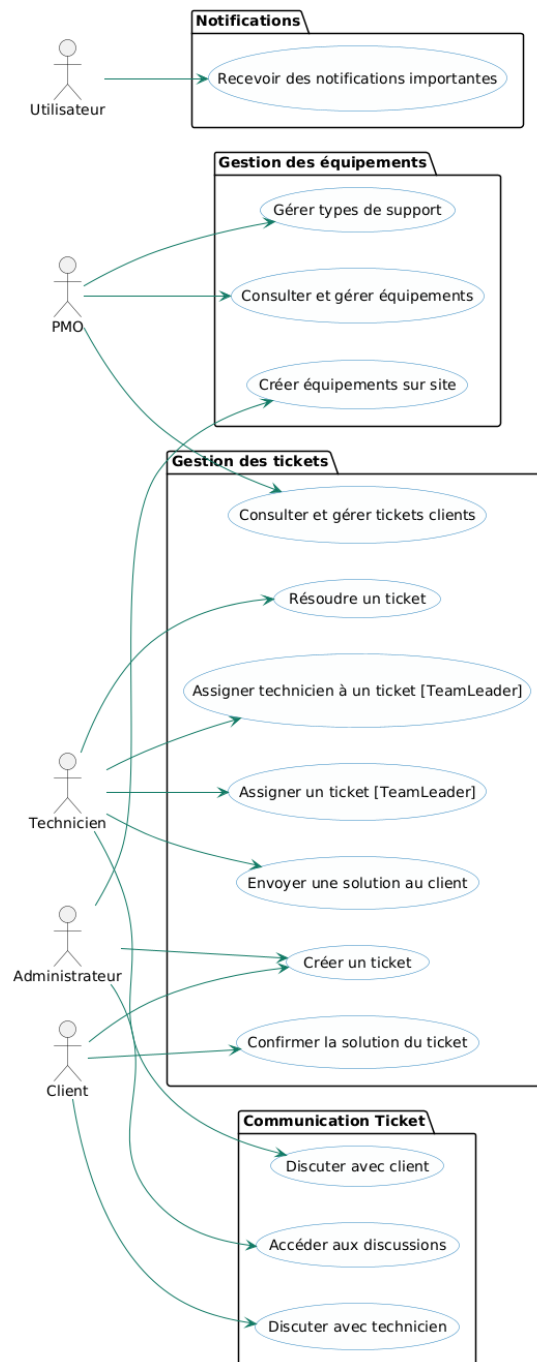


FIGURE 5.5 – diagramme de cas d'utilisation du sprint 3

5.6.2 Diagramme de classe du sprint 3

Une présentation du diagramme de classes de conception du sprint 3 :

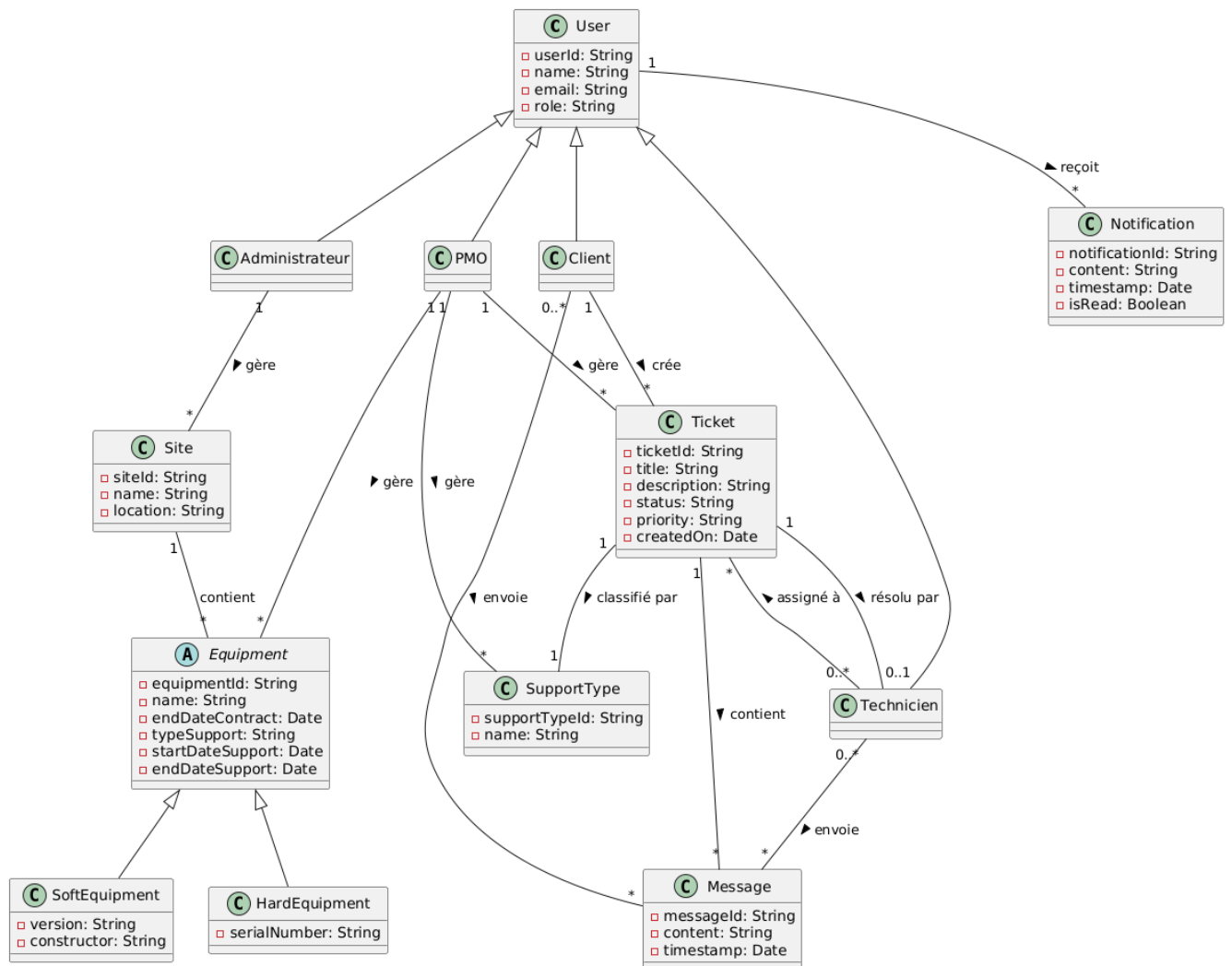


FIGURE 5.6 – Une présentation du diagramme de classes de conception du sprint 3

5.6.3 Vue dynamique du sprint 3

5.6.3.1 Diagramme de séquence objet N°21

Ce diagramme illustre l'interaction entre l'administrateur, l'interface, l'API et la base de données pour associer un équipement à un site. Après authentification, l'administrateur sélectionne un site et un équipement, puis valide l'association. L'opération est traitée côté serveur et confirmée via l'interface.

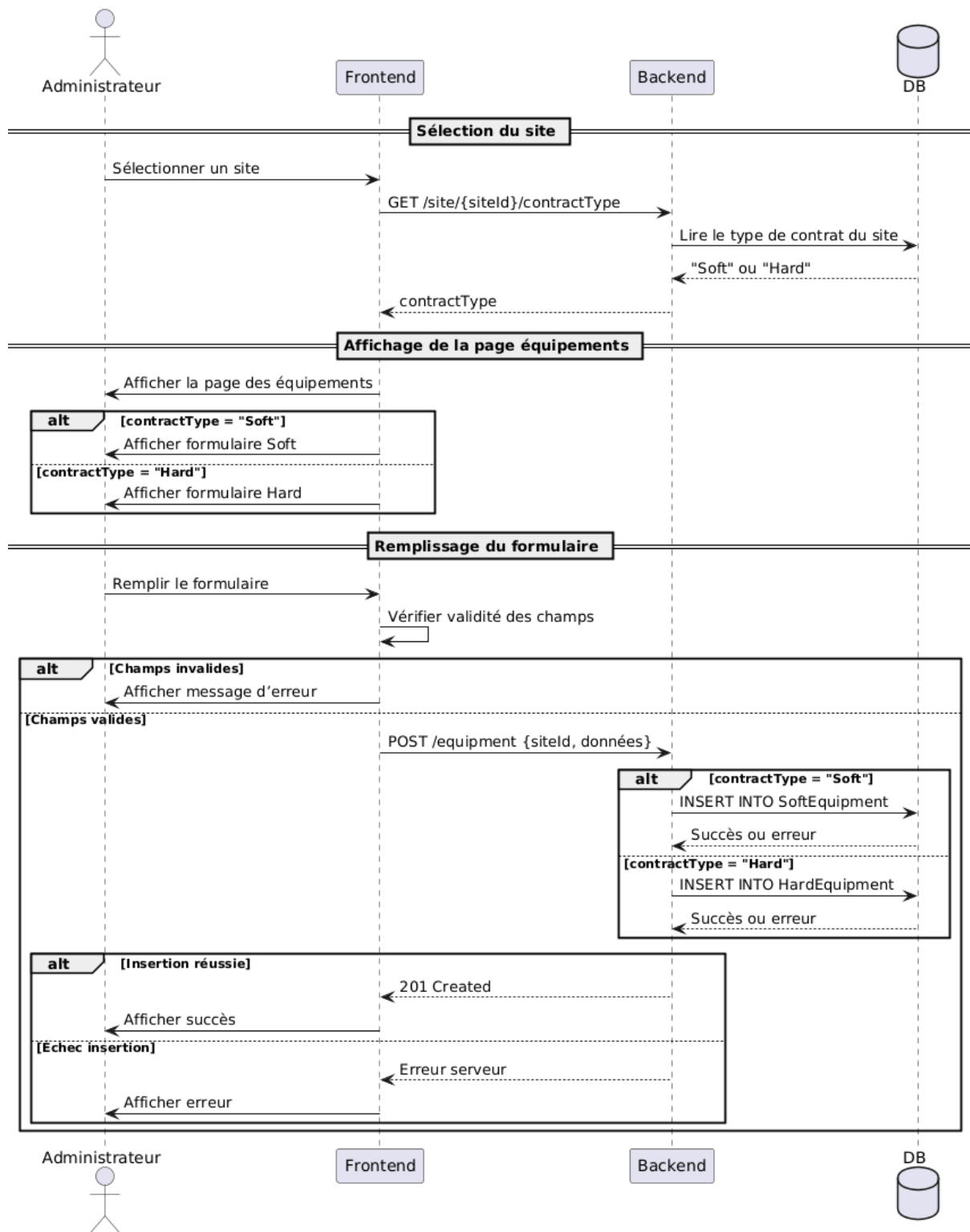


FIGURE 5.7 – Diagramme de séquence objet «la création d'un équipement»

5.6.3.2 Diagramme de séquence objet de user story N°24

Ce diagramme montre comment un client se connecte à l'application, accède à l'interface de création de ticket, puis envoie les informations nécessaires via l'API. Le backend enregistre le ticket dans la base de données. Une confirmation est ensuite renvoyée à l'interface.

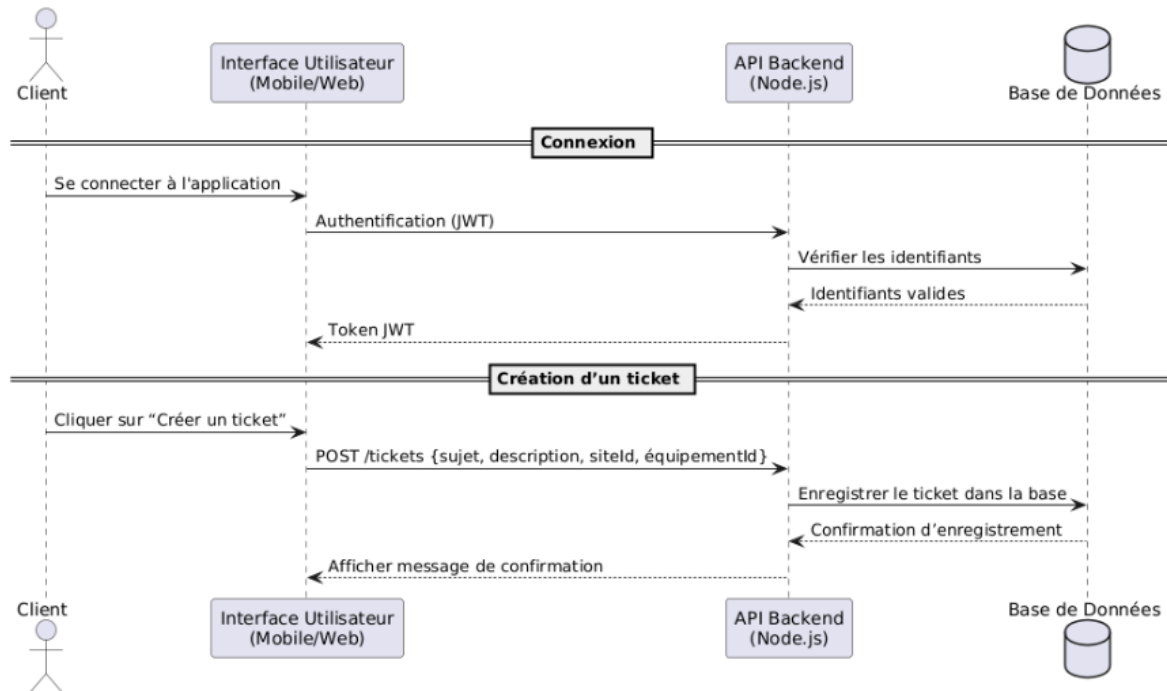


FIGURE 5.8 – Diagramme de séquence objet «la création d'un ticket»

5.6.3.3 Diagramme de séquence objet de user story N°32

Ce diagramme décrit comment un technicien accède à ses tickets assignés, sélectionne un ticket et soumet une solution via l'interface. L'API met à jour le statut du ticket dans la base de données. Une confirmation de résolution est ensuite affichée à l'utilisateur.

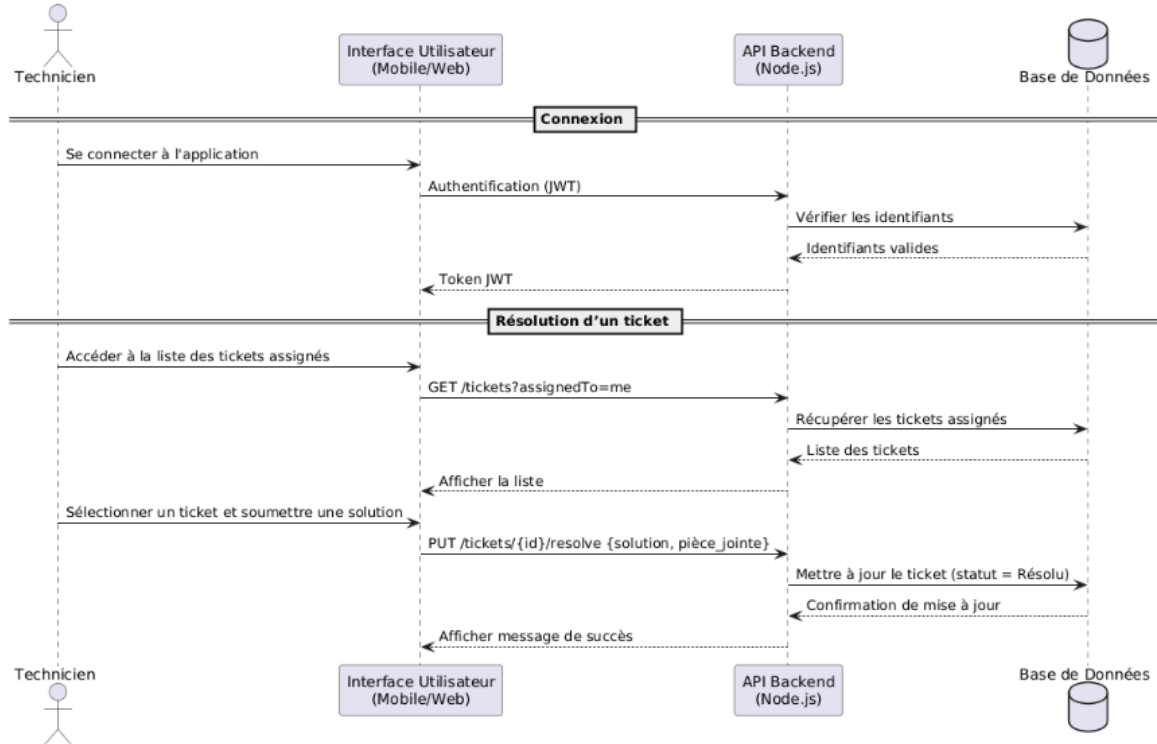


FIGURE 5.9 – Diagramme de séquence objet «la résolution d'un ticket»

5.7 Réalisation de sprint1

5.7.1 Interface de user story N°21

Cette interface permet à l'administrateur d'ajouter un équipement via un formulaire associé à un site sélectionné.

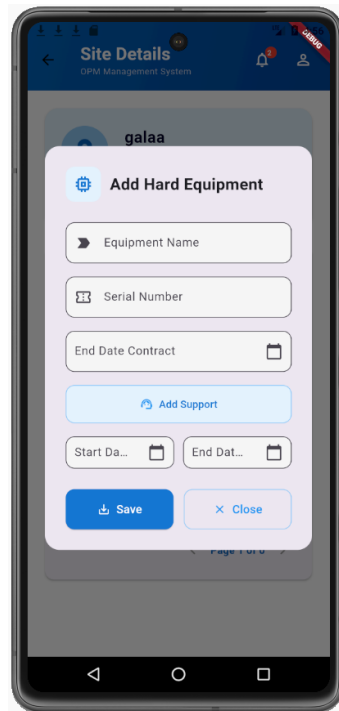


FIGURE 5.10 – Interface de user story N°21

5.7.2 Interface de user story N°24

Cette interface affiche un formulaire avec des champs obligatoires pour faire la création d'un ticket via un client.

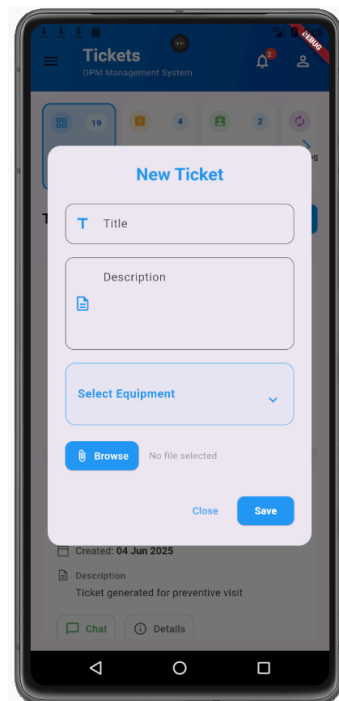


FIGURE 5.11 – Interface de user story N°24

5.7.3 Interface de user story N°32

Cette interface permet l’affichage d’une discussion entre un technicien et un client. Le but de cette discussion est la validation du client pour une solution de cette ticket



FIGURE 5.12 – Interface de user story N°32

Conclusion Durant ce sprint, plusieurs fonctionnalités importantes ont été développées, notamment la gestion des équipements, des tickets et des échanges entre utilisateurs. Les différents rôles définis dans l’application ont été suivis et mis en œuvre selon les exigences. Le traitement complet d’un ticket, depuis sa création jusqu’à sa clôture, a été couvert. Le travail réalisé correspond aux user stories prévues avec un bon niveau de cohérence. Ce sprint représente une étape stable et aboutie dans l’avancement du projet.

Chapitre 6

Sprint 4

6.1 Introduction

Ce quatrième sprint met l'accent sur l'amélioration de la maintenance préventive et l'enrichissement de l'expérience utilisateur. Il propose des fonctionnalités clés telles que la planification des visites, la l'élaboration des rapports et l'intégration d'un chatbot. Le tableau suivant met présente les user stories sélectionnées et leur complexité évaluée.

6.2 Objectifs du Sprint 4

Cette sprint a pour objectif d'optimiser la gestion des maintenances, l'efficacité des techniciens et le pilotage du système. Elle propose la planification des visites préventives, un calendrier consultable, un chabot intégré dans la base de connaissances, la génération de rapports d'intervention et un tableau de bord fournissant aux utilisateurs une vue claire sur le système.

6.3 Backlog du sprint 4

ID	User Story	Story Points
36	En tant qu'administrateur, je veux planifier des visites de maintenance préventive.	3
37	En tant que PMO, je veux planifier des visites de maintenance préventive.	3
38	En tant qu'utilisateur, je veux accéder au calendrier pour consulter les visites de maintenance préventive.	2
39	En tant que technicien, je veux accéder à la base de connaissances pour résoudre un ticket plus efficacement.	2
40	En tant que technicien, je veux accéder à un chabot intégré dans la base de connaissances.	2
41	En tant que technicien, je veux générer un rapport après une visite pour documenter l'intervention.	3
42	En tant qu'utilisateur, je veux consulter les indicateurs clés sur le tableau de bord.	3
43	En tant que technicien, je veux générer un rapport pour un ticket en PDF.	3

TABLE 6.1 – Backlog du sprint 4

6.4 Diagramme d'activité

6.4.1 Diagrammes d'activités de user story N°35

L'administrateur accède à l'espace de planification des visites préventives, sélectionne le site du client, choisit une date et assigne un technicien. Le système vérifie les données : si elles sont valides, la visite est enregistrée et visible dans le calendrier. En cas d'erreur, un message d'erreur est affiché.

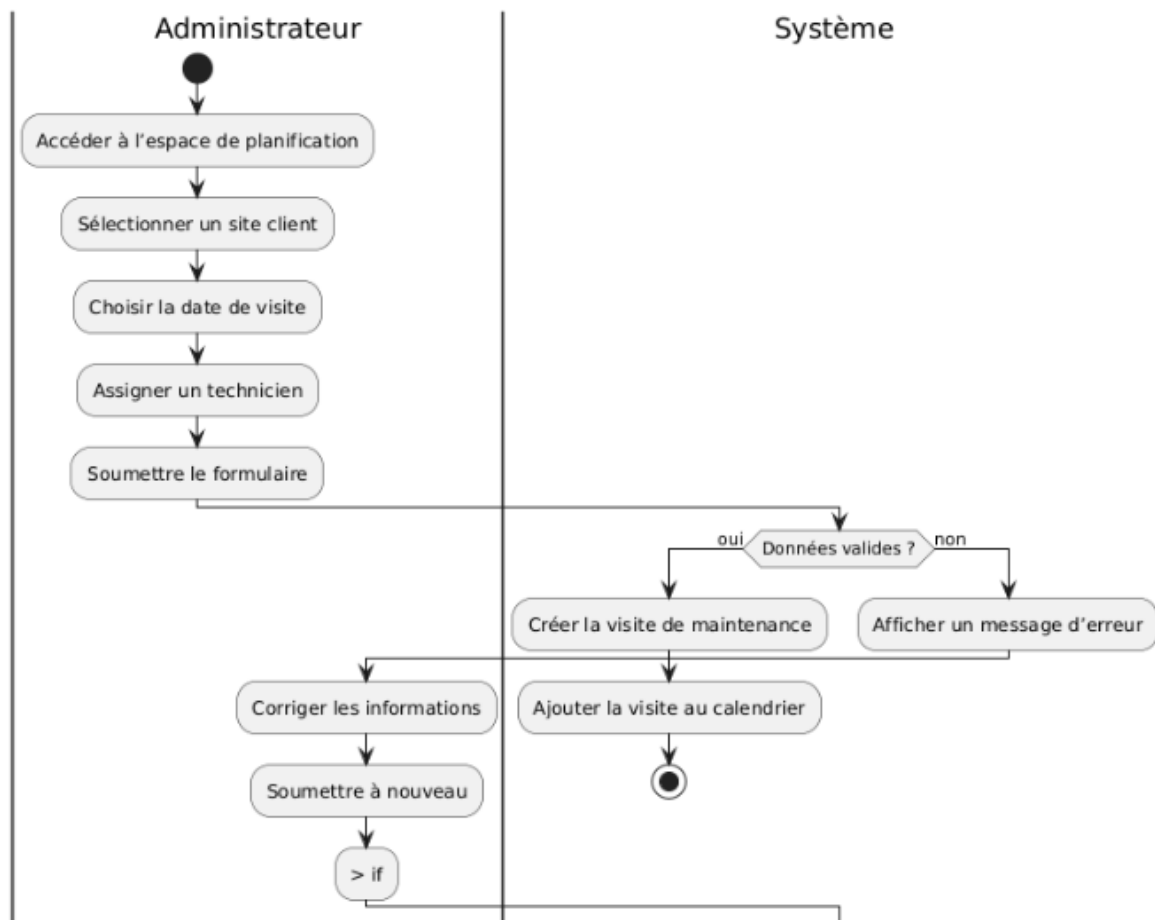


FIGURE 6.1 – Diagrammes d'activités« planification des visites préventives»

6.4.2 Diagrammes d'activités de user story N°41

Au moment où L'utilisateur accède au tableau de bord, le système récupère automatiquement les données et génère des statistiques visuelles dynamiques (graphiques, indicateurs de performance, évolution des tickets, etc.). L'utilisateur visualise ces données pour évaluer les performances du système et prendre des décisions.

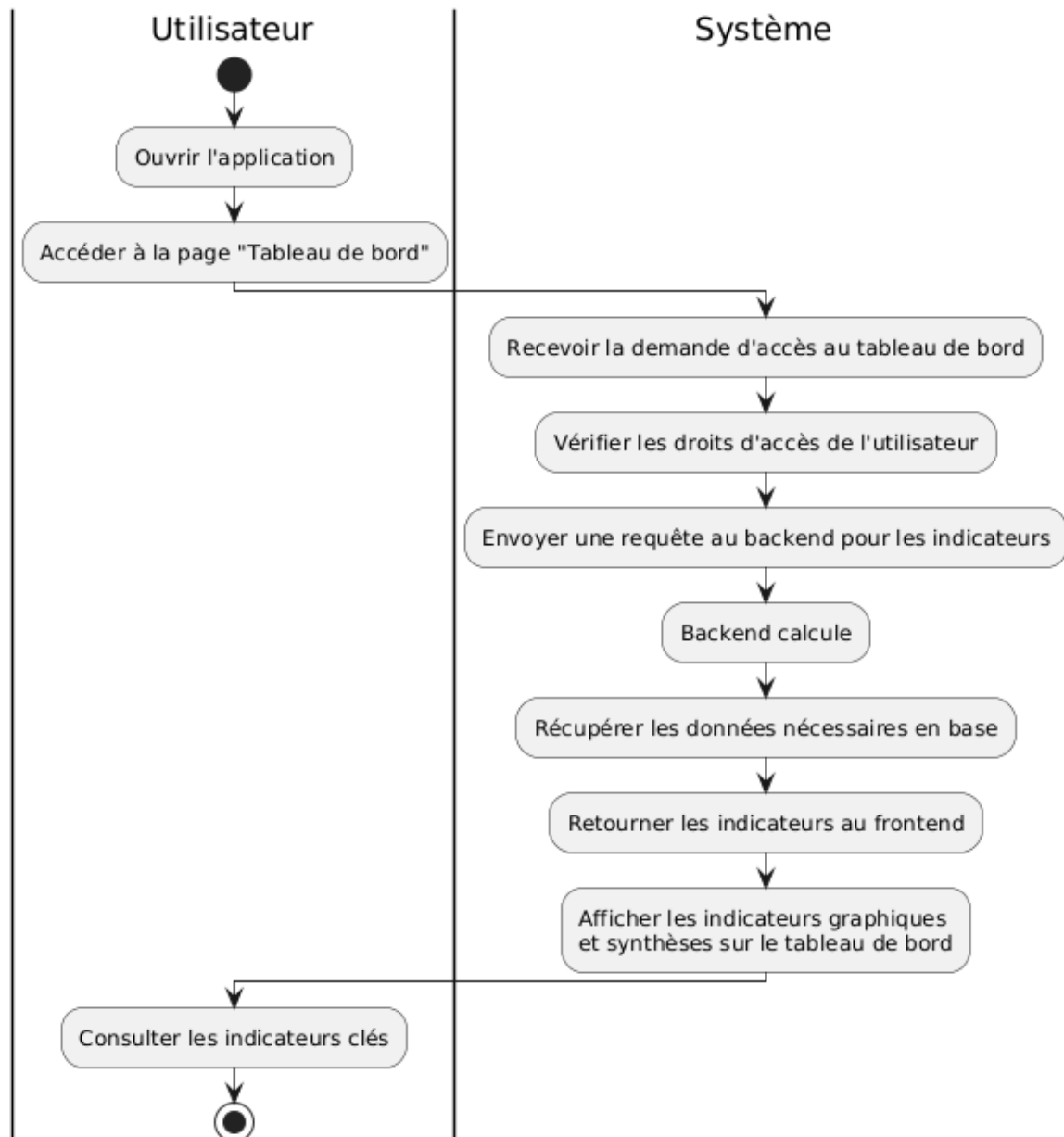


FIGURE 6.2 – Diagrammes d’activités«l’accées au tableau de bord»

6.4.3 Diagrammes d’activités de user story N°42

Le technicien accède à l’un de ses tickets, clique sur l’option « Générer un rapport », remplit les champs requis, puis valide. Le système génère un fichier PDF avec les données du ticket et du technicien, puis le rend téléchargeable ou consultable dans l’historique du ticket.

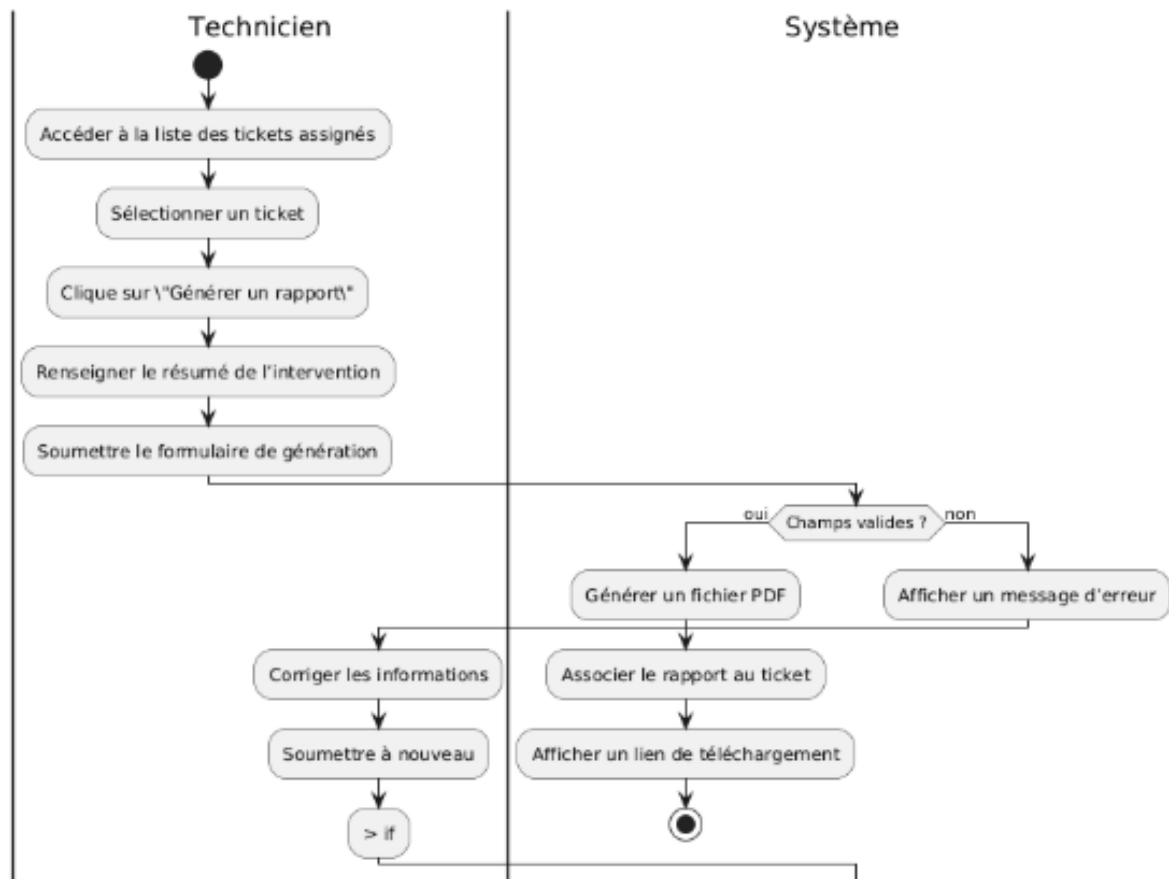


FIGURE 6.3 – Diagrammes d'activités«génération du rapport du ticket»

6.5 Les Scénario

6.5.1 Sénario de user story N°35

Ce tableau de scénario décrit la planification d'une visite préventive par un administrateur connecté. Puis cette visite sera affiché dans une calendrier.

Cas d'utilisation	En tant qu'administrateur, je veux planifier des visites de maintenance préventive.
Acteur	Administrateur
Précondition	L'administrateur est connecté à la plateforme.
Post-condition	La visite est enregistrée dans le calendrier du système.
Scénario nominal	L'administrateur planifie une visite préventive en remplissant les champs requis(site, date, technicien...).Puis, le système enregistre cette visite dans le calendrier.
Scénario d'exception	Si un champ obligatoire est manquant ou invalide, le système empêche la soumission.

TABLE 6.2 – Description textuelle de la création d'une visite préventive

6.5.2 Scénario de user story N°41

Ce tableau de scénario décrit l’affichage des indicateurs clés sur le tableau de bord dès que l’utilisateur se connecte. En cas d’erreur de chargement, le système va envoyer un message d’erreur à l’utilisateur

Cas d’utilisation	En tant qu’utilisateur, je veux consulter les indicateurs clés sur le tableau de bord.
Acteur	Utilisateur
Précondition	L’utilisateur est connecté à la plateforme.
Post-condition	Les indicateurs s’affichent dynamiquement selon les données du système.
Scénario nominal	Au moment où l’utilisateur accède au tableau de bord, le système récupère automatiquement les données les plus récentes. Les indicateurs clés (KPI) tels que le nombre de tickets résolus, les visites réalisées et affichés sous forme de graphiques et tableaux dynamiques.
Scénario d’exception	Si une erreur survient dans le chargement des données (indisponibilité temporaire, données incomplètes), le système affiche un message d’erreur.

TABLE 6.3 – Description textuelle de tableau de bord

6.5.3 Scénario de user story N°42

Ce tableau de scénario décrit la génération d’un rapport PDF du ticket par un technicien qui lui est assigné. Si les données sont invalides, va envoyer un message d’erreur au technicien.

Cas d’utilisation	En tant que technicien, je veux générer un rapport pour un ticket en PDF.
Acteur	Technicien
Précondition	Le technicien est connecté et le ticket est assigné à lui.
Post-condition	Un fichier PDF est généré et attaché au ticket concerné.
Scénario nominal	Le technicien sélectionne l’un de ses tickets. Il clique ensuite sur « Générer un rapport », ce qui fait apparaître un formulaire. Il remplit un résumé de son intervention. Une fois toutes les données saisies, il valide le formulaire. Le système vérifie les champs, puis génère un fichier PDF contenant les détails du ticket et les éléments fournis.
Scénario d’exception	Si le résumé est manquant ou qu’un fichier joint est dans un format non autorisé, le système empêche la génération du PDF et affiche un message d’erreur.

TABLE 6.4 – Description textuelle pour la création d’un rapport d’un ticket

6.6 Conception du Sprint 4

6.6.1 Diagramme de cas d'utilisation du sprint 4

Ce diagramme représente les cas d'utilisation du sprint 4, axés sur la planification des visites de maintenance, l'accès à la base de connaissances avec chatbot, et la génération de rapports par les techniciens. Il met également en lumière la consultation du calendrier et des indicateurs clés par les utilisateurs. Les rôles et fonctionnalités sont structurés de manière à offrir une lecture claire et modulaire.

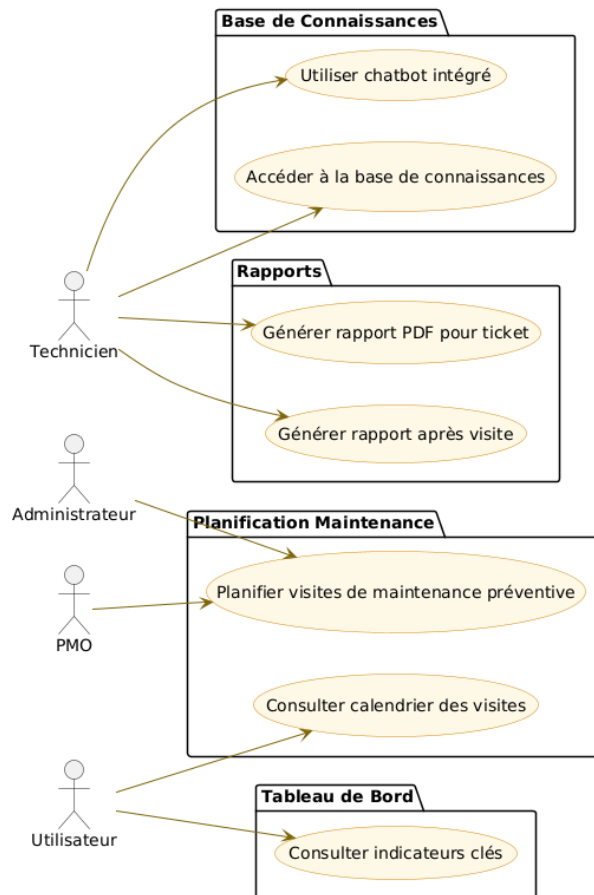


FIGURE 6.4 – diagramme de cas d'utilisation du sprint 4

6.6.2 Diagramme de classe du sprint 4

Une présentation du diagramme de classes de conception du sprint 4, qui montre la création d'une visite préventive, consultation du tableau de bord et les outils qui facilite la résolution d'un ticket pour un technicien (base de connaissance et chatbot) :

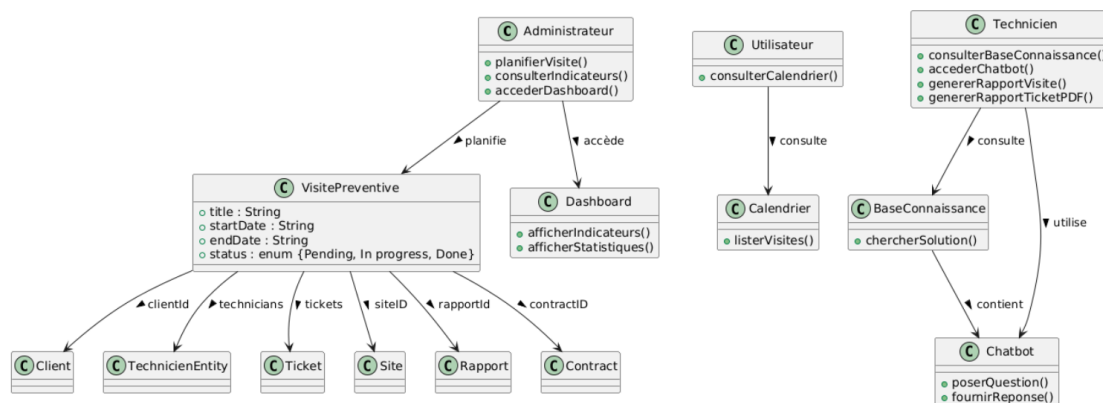


FIGURE 6.5 – Une présentation du diagramme de classes de conception du sprint 4

6.6.3 Vue dynamique du sprint 4

6.6.3.1 Diagramme de séquence objet N°35

Le diagramme montre comment un administrateur planifie ou modifie une visite préventive. La visite est enregistrée en base, puis une notification est envoyée aux techniciens. Elle est ensuite affichée ou mise à jour dans le calendrier intégré de l'application.

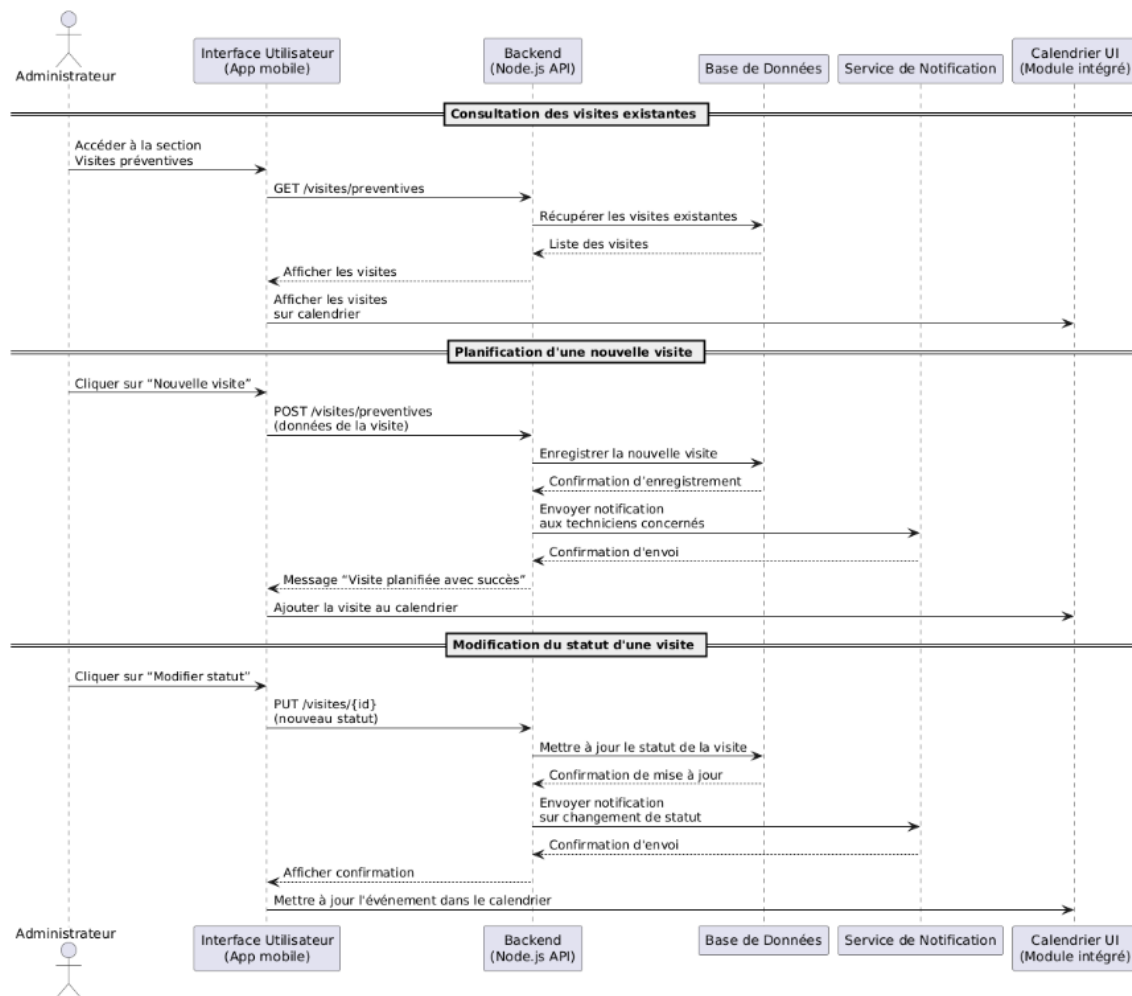


FIGURE 6.6 – Diagramme de séquence «Planification du visite péventive»

6.6.3.2 Diagramme de séquence objet N°38 et N°39

Ce diagramme montre comment le technicien accède à la base de connaissances pour consulter des articles, puis interagit avec un chatbot intégré pour obtenir des réponses ciblées. Le backend et la base de données sont sollicités pour fournir les contenus et suggestions en temps réel.

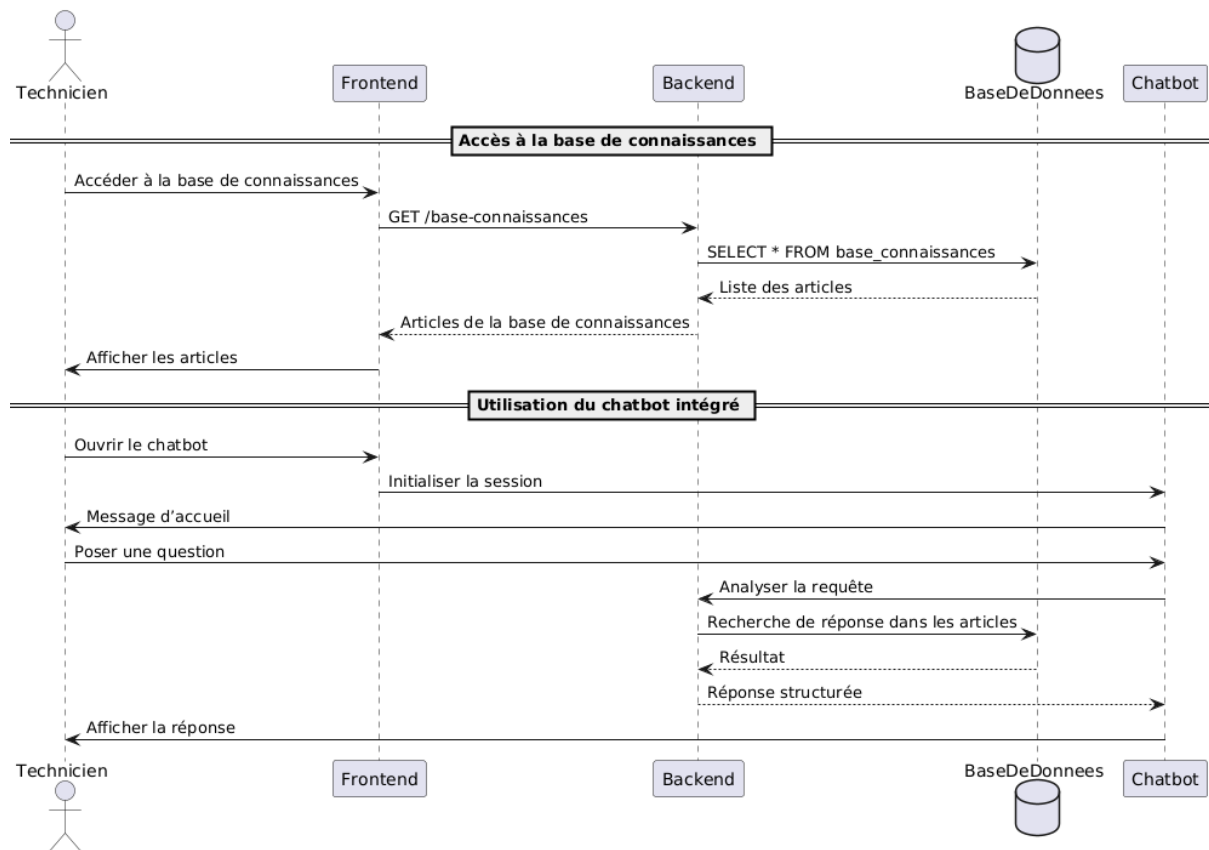


FIGURE 6.7 – Diagramme de séquence «les outils des techniciens : knowledge base et chatbot»

3.6.2.3 Diagramme de séquence objet de user story N°41

L'utilisateur accède à l'interface pour consulter le tableau de bord. Le contrôleur demande au service statistiques de générer les KPI à partir des données. Les indicateurs sont retournés et affichés à l'utilisateur via l'interface.

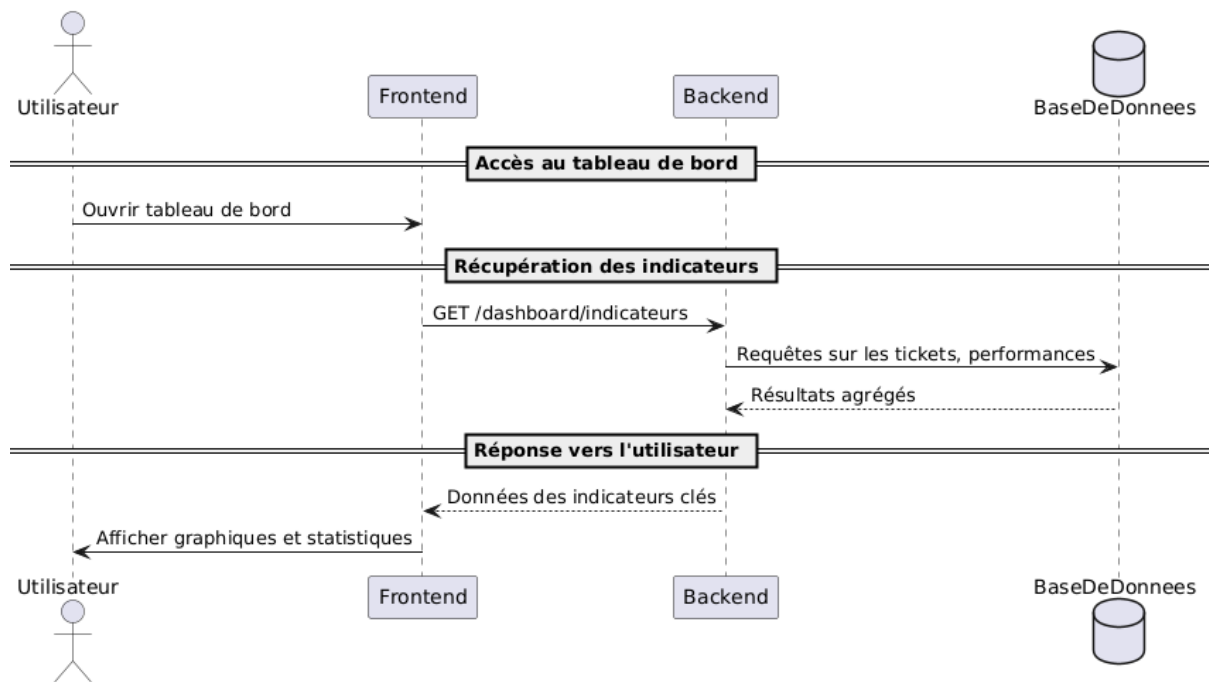


FIGURE 6.8 – Diagramme de séquence «tableau de bord»

3.6.2.4 Diagramme de séquence objet de user story N°42

Le technicien sélectionne un ticket et consulte ses détails. Il clique sur "Générer PDF", ce qui déclenche la création du rapport via un service dédié. Le rapport PDF est généré, puis proposé en téléchargement à l'utilisateur.

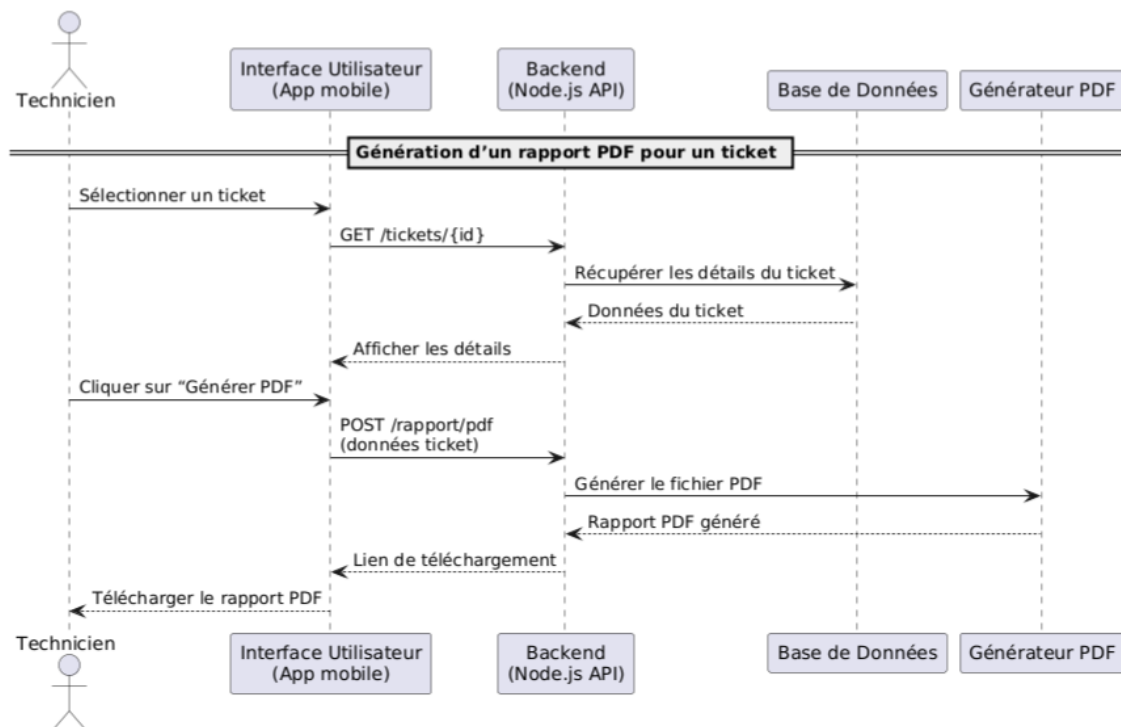


FIGURE 6.9 – Diagramme de séquence «création du rapport d'un ticket»

6.7 Réalisation de sprint 4

6.7.1 Interface de user story N°35 et N°37

Ces deux interfaces montrent la creation d'une visite préventive par l'administrateur. Ensuite cette visite sera affiché dans la calendrier de chaque utilisateur en fonction de son rôle.

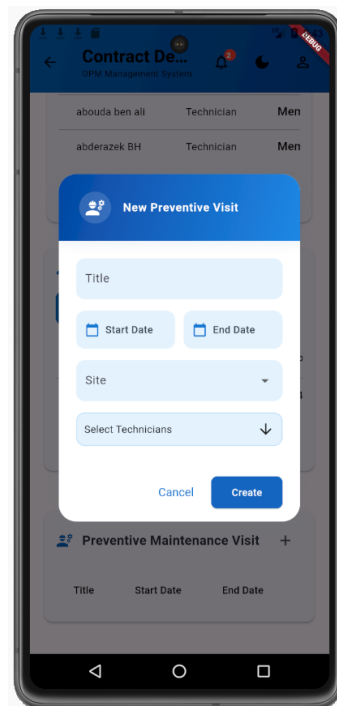


FIGURE 6.10 – Interface de la création du visite préventive

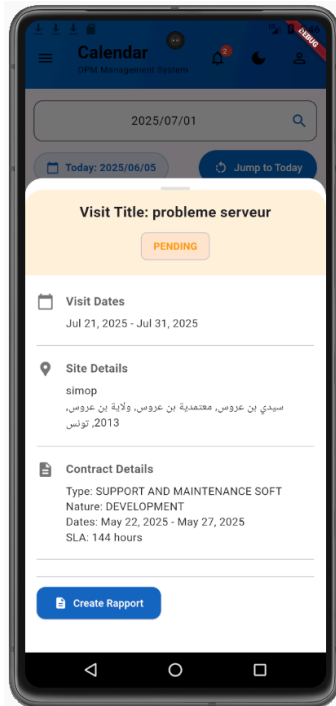


FIGURE 6.11 – Interface du calendrier

6.7.2 Interface de user story N°38 et N°39

Ces deux interfaces décrivent les outils qui facilite la résolution d'un ticket pour le technicien.

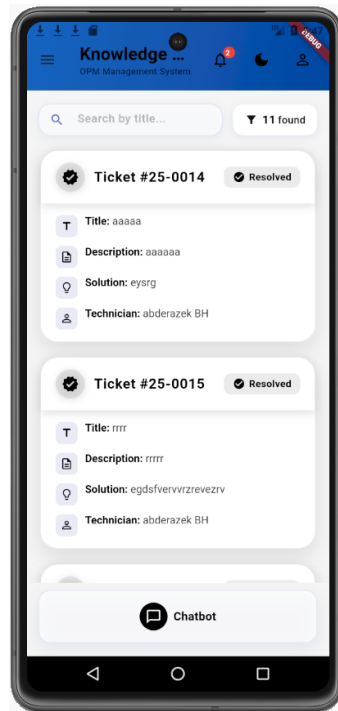


FIGURE 6.12 – interface du knowledge base

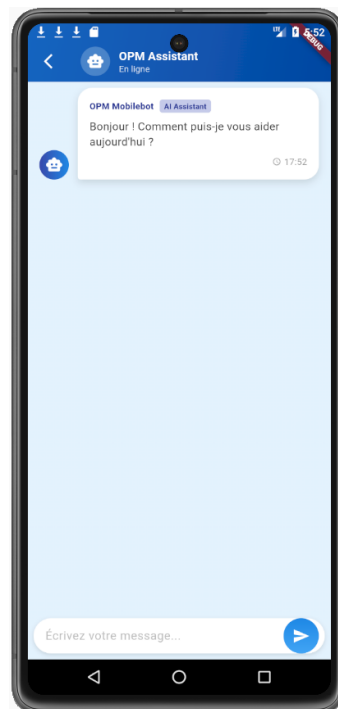


FIGURE 6.13 – Interface du chatbot

6.7.3 Interface de user story N°41

Cette interface indique l’affichage d’un tableau de bord dès que l’utilisateur se connecte sur l’application

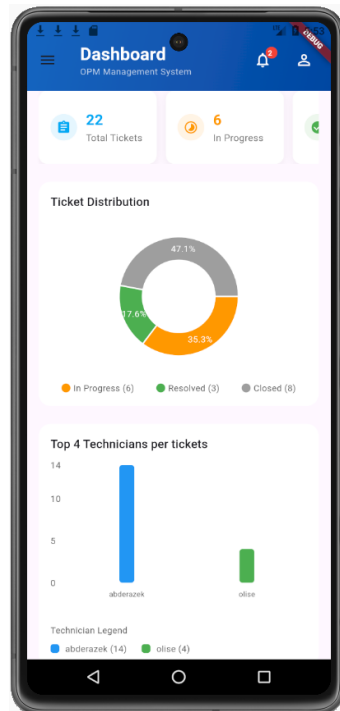


FIGURE 6.14 – Interface du table de bord

6.7.4 Interface de user story N°42

Ces interfaces montrent la génération du rapport d'un ticket à l'aide d'un technicien. Ensuite, la possibilité de l'affichage et le téléchargement de ce rapport

Add Rapport

Motif (Project Framework)

Type d'intervention

Start Date: Start Time:

End Date: End Time:

List of Tasks

Enter each task on a new line

Follow-Up

Add notes, observations, and next steps

Cancel Save

FIGURE 6.15 – Interface de la création d'un rapport d'un ticket



FIGURE 6.16 – Interface de téléchargement du rapport

Conclusion Ce sprint a contribué à améliorer plusieurs fonctionnalités clés liées à la maintenance, à la documentation technique et à la visualisation des données. La planification des visites de maintenance préventive est désormais accessible aux profils administrateur et PMO, alors que les utilisateurs peuvent en suivre l'évolution via un calendrier intégré. Du point de vue technique, l'accès à la base de connaissances et au chatbot permet d'améliorer la rapidité du traitement des tickets. La génération de rapports d'intervention en format PDF assure une meilleure traçabilité des actions réalisées. Enfin, l'intégration du tableau de bord permet aux utilisateurs de consulter en temps réel les indicateurs clés de performance.

Conclusion Générale

La développement de l'application mobile OPM représente une progression stratégique dans la transformation numérique des processus de gestion des projets externalisés. En centralisant toutes les fonctions essentielles, comme la gestion des utilisateurs, des contrats, des tickets, des interventions et des équipements. La solution développée permet de résoudre avec efficacité les limites liées à la dispersion des outils, à l'insuffisance de réactivité et au défaut de traçabilité.

S'appuyant sur une technologie avancée alliant Flutter pour le développement multiplateforme de l'interface utilisateur et Node.js pour la robustesse et la scalabilité du backend, l'application offre une expérience homogène, sécurisée (grâce à l'authentification JWT), et ajustée aux besoins spécifiques des différents profils d'utilisateurs (administrateurs, pmo, techniciens, clients, commercial).

Dans une logique d'amélioration continue et d'expansion fonctionnelle, plusieurs axes de développement futur sont prévus :

L'incorporation d'un module de facturation et de paiement, pour automatiser le cycle financier lié aux prestations et contrats de service.

L'implémentation en mode multi-entreprise (version SaaS), afin de mutualiser la plateforme pour différentes entités tout en assurant une séparation logique et sécurisée des données.

L'instauration d'un système de notation et de satisfaction client, dans une optique de qualité de service et de retour d'expérience utilisateur.

De plus, les bases techniques de ce projet ouvrent la voie à l'intégration progressive de technologies avancées telles que l'intelligence artificielle pour l'optimisation des affectations.

En somme, OPM mobile s'impose comme un levier essentiel vers une gestion de projet moderne, intelligente et collaborative, parfaitement en phase avec les défis actuels de performance, de flexibilité et de transformation digitale des organisations.

Nétographie

[1] : <https://visualstudio.microsoft.com/fr/>

[2] : <https://www.v-labs.fr/glossaire/android-studio/>

[3] : <https://www.drawio.com/about>

[4] : <https://bility.fr/definition-postman/>

[5] : <https://www.blogdumoderateur.com/tools/figma/>

[6] : <https://www.techtarget.com/searchdatamanagement/definition/MongoDB>

[7] : <https://www.browserstack.com/guide/what-is-flutter>

[8] : <https://www.editions-eni.fr/livre/flutter-developpez-vos-applications-mobiles-multiplateformes-avec-dart-9782409025273/dart-les-fondamentaux>

[9] : <https://www.appvizer.fr/magazine/services-informatiques/gestion-informatique/nodejs-glossaire>

[10] : <https://www.geeksforgeeks.org/express-js/>

[11] : <https://www.futura-sciences.com/tech/definitions/internet-javascript-509/>

[12] : <https://www.wildcodeschool.com/lexique/definition-python>

Résumé

Dans le cadre de la réalisation de l'application mobile OPM, plusieurs outils et technologies ont été utilisés pour assurer le bon déroulement du développement, aussi bien au niveau frontend que backend. Dans le domaine mobile, j'ai opté pour Flutter comme

framework principal, en utilisant le langage Dart, Cela m'a permis de créer une interface utilisateur moderne et responsive. L'application a été élaborée en employant environnements de développement Visual Studio Code et Android Studio, qui m'ont donné la possibilité de coder, simuler et tester les différentes fonctionnalités dans un cadre Android. Sur le plan du serveur, j'ai utilisé Node.js associé au framework Express.js afin de

construire des API REST efficaces, assurant la communication entre l'application mobile et la base de données. Les données sont stockées dans MongoDB, une base de données orientée documents, offrant ainsi une structuration flexible et évolutive des données. Le

processus de développement a été soutenu par divers outils, comme Postman pour l'essai des API, Draw.io pour la création des diagrammes fonctionnels et Figma pour la conception des maquettes de l'interface. Finalement, Python a été utilisé ponctuellement pour automatiser certaines tâches internes en rapport avec le projet.