

**Université de Tunis**  
**École Supérieure des Sciences Économiques et Commerciales de Tunis**



Projet de fin d'études en vue de l'obtention de  
la Licence Business Computing  
Parcours E-Business

---

**Conception et développement  
d'une application de gestion de parc**

---

Organisme d'accueil : Digital Identity



Réalisé par :

Balti Firas

Mimouni Mohamed Aziz

Supervisé par :

**Encadrante Académique**

Mme. Chamakhi Hanen

**Encadrant Professionnel**

Mr. Mbarki Mohamed Seif





# Table des matières

<b>Table des matières</b>	<b>4</b>
<b>Table des figures</b>	<b>8</b>
<b>Liste des tableaux</b>	<b>11</b>
<b>Introduction générale</b>	<b>12</b>
<b>1 Etude préliminaire du projet</b>	<b>13</b>
Introduction . . . . .	13
1.1 Présentation de l'organisme d'accueil . . . . .	13
1.1.1 Organigramme de l'entreprise . . . . .	14
1.1.2 Les activités de Digital Identity : . . . . .	14
1.2 Cadre du projet . . . . .	15
1.2.1 Problématique . . . . .	15
1.2.2 Solution proposée . . . . .	15
1.3 Méthodologie et langage de conception . . . . .	16
1.3.1 Méthodes agiles . . . . .	16
1.3.2 Principes du framework Scrum . . . . .	17
1.3.3 Rôles du framework Scrum . . . . .	18
1.3.4 Langage de modélisation . . . . .	18
1.4 Environnement de travail . . . . .	19
1.4.1 Langages et framework utilisés . . . . .	19
1.4.2 Environnement matériel . . . . .	21
1.4.3 Environnement logiciel . . . . .	21
1.5 Organisation du travail . . . . .	24
Conclusion . . . . .	25
<b>2 Planification du projet</b>	<b>26</b>
Introduction . . . . .	26
2.1 Spécification des besoins . . . . .	26
2.1.1 Identification des acteurs . . . . .	26
2.1.2 Besoins fonctionnels . . . . .	28

## TABLE DES MATIÈRES

---

2.1.3	Besoins non fonctionnels . . . . .	29
2.1.4	Diagramme de cas d'utilisation . . . . .	30
2.2	Pilotage du projet avec Scrum . . . . .	31
2.2.1	Équipe et rôles . . . . .	31
2.2.2	Contraintes et exigences . . . . .	31
2.2.3	Risques . . . . .	31
2.2.4	Backlog du produit . . . . .	31
2.3	Conception de l'application . . . . .	33
2.3.1	Prototypes de l'application . . . . .	33
2.3.2	Charte Graphique . . . . .	37
2.3.3	Plan de l'application . . . . .	37
2.3.4	Diagramme de classe du domaine . . . . .	43
2.3.5	Diagramme de Gantt . . . . .	44
	Conclusion . . . . .	44
<b>3</b>	<b>Sprint 1 : Gestion des accès</b>	<b>45</b>
	Introduction . . . . .	45
3.1	Sprint Backlog . . . . .	45
3.2	Analyse et conception . . . . .	46
3.2.1	Diagramme de cas d'utilisation du sprint 1 . . . . .	46
3.2.2	Diagramme de séquence système du cas d'utilisation «S'authentifier» . . . . .	47
3.2.3	Description textuelle du cas d'utilisation “S'authentifier” . . . . .	49
3.2.4	Raffinement du cas d'utilisation « Gérer les utilisateurs » . . . . .	50
3.2.5	Diagramme de séquence système du cas d'utilisation «Ajouter utilisateur» . . . . .	50
3.2.6	Description textuelle du cas d'utilisation “Ajouter utilisateur” . . . . .	51
3.3	Diagramme de classes du sprint 1 . . . . .	52
3.4	Réalisation . . . . .	53
3.4.1	Authentification . . . . .	53
3.4.2	Profil . . . . .	54
3.4.3	Gestion des utilisateurs . . . . .	55
3.4.4	Création d'un nouvel utilisateur . . . . .	56
3.5	Rétrospective . . . . .	56
3.5.1	Burdown Chart . . . . .	57
3.5.2	Task Bord . . . . .	57
3.6	Tests unitaires . . . . .	58
	Conclusion . . . . .	59
<b>4</b>	<b>Sprint 2 : Gestion des Tâches</b>	<b>60</b>
	Introduction . . . . .	60
4.1	Sprint Backlog . . . . .	60

## TABLE DES MATIÈRES

---

4.2	Analyse et conception . . . . .	61
4.2.1	Diagramme de cas d'utilisation du sprint 2 . . . . .	61
4.2.2	Raffinement du cas d'utilisation « Gérer les véhicules » . . . . .	61
4.2.3	Description textuelle du cas d'utilisation « Ajouter véhicules » . . . . .	62
4.2.4	Diagramme de séquence système du cas d'utilisation « Ajouter des véhicules »	63
4.2.5	Raffinement du cas d'utilisation « Créer les taches » . . . . .	64
4.2.6	Description textuelle du cas d'utilisation « Créer des Tâches pour les Mécaniciens » . . . . .	64
4.2.7	Description textuelle du cas d'utilisation « Créer des Tâches pour les Chauffeurs » . . . . .	65
4.2.8	Diagramme de séquence système du cas d'utilisation «Créer les taches» .	67
4.2.9	Raffinement du cas d'utilisation « Consulter les taches pour les chauffeurs »	68
4.2.10	Description textuelle du cas d'utilisation « Consulter les taches pour les chauffeurs » . . . . .	68
4.2.11	Diagramme d'activité du cas d'utilisation «Consulter les taches des chauffeurs» . . . . .	69
4.2.12	Raffinement du cas d'utilisation « Consulter les taches pour les mécaniciens »	70
4.2.13	Description textuelle du cas d'utilisation « Consulter les taches pour les mécaniciens » . . . . .	70
4.3	Diagramme de classes du sprint 2 . . . . .	71
4.4	Réalisation . . . . .	72
4.4.1	Interface de création des taches . . . . .	72
4.4.2	Consultation du checklist du chauffeur . . . . .	74
4.4.3	Consultation des taches du mécanicien . . . . .	74
4.5	Rétrospective . . . . .	74
4.5.1	Burdown Chart . . . . .	75
4.5.2	Task Bord . . . . .	75
4.6	tests unitaires . . . . .	76
	Conclusion . . . . .	77
<b>5</b>	<b>Sprint 3 : Suivi et Cr éation des Rapports</b>	<b>78</b>
	Introduction . . . . .	78
5.1	Sprint Backlog . . . . .	78
5.2	Analyse et conception . . . . .	79
5.2.1	Diagramme de cas d'utilisation du sprint 3 . . . . .	79
5.2.2	Description textuelle du cas d'utilisation « Suivi en Temps Réel par Géolocalisation » . . . . .	80
5.2.3	Description textuelle du cas d'utilisation « Planifier les déplacements » .	82
5.2.4	Diagramme de séquence système du cas d'utilisation « Planifier les déplacements » . . . . .	82

## TABLE DES MATIÈRES

---

5.2.5	Description textuelle du cas d'utilisation « Consulter Notifications » . . . . .	83
5.2.6	Diagramme d'activité du cas d'utilisation « Consulter Notifications » . . . . .	83
5.2.7	Raffinement du cas d'utilisation « Générer des rapports » . . . . .	84
5.2.8	Description textuelle du cas d'utilisation « Générer des rapports » . . . . .	84
5.2.9	Diagramme de séquence système du cas d'utilisation « Générer des rapports »	85
5.3	Diagramme de classes du sprint 3 . . . . .	86
5.4	Réalisation . . . . .	87
5.4.1	Suivi en Temps Réel par Géolocalisation . . . . .	87
5.4.2	La planification des déplacements . . . . .	87
5.4.3	Les Notifications . . . . .	88
5.4.4	Génération d'un Rapport . . . . .	89
5.5	Rétrospective . . . . .	90
5.5.1	Burdown Chart . . . . .	90
5.5.2	Task Bord . . . . .	90
5.6	Tests unitaires . . . . .	91
	Conclusion . . . . .	92

# Table des figures

1.1	Logo de l'entreprise « DIGITAL IDENTITY » . . . . .	13
1.2	Organigramme de Digital identity . . . . .	14
1.3	Les activités de Digital Identity . . . . .	14
1.4	Mode de fonctionnement de la méthodologie Scrum . . . . .	17
1.5	UML . . . . .	18
1.6	Java . . . . .	19
1.7	Python . . . . .	19
1.8	MySQL . . . . .	19
1.9	ReactJS . . . . .	20
1.10	Node.js . . . . .	20
1.11	REST API . . . . .	20
1.12	Figma . . . . .	21
1.13	Android Studio . . . . .	21
1.14	Overleaf . . . . .	22
1.15	Spyder . . . . .	22
1.16	Jira . . . . .	22
1.17	Slack . . . . .	23
1.18	Material UI . . . . .	23
1.19	Visme . . . . .	23
1.20	Postman . . . . .	23
1.21	Création du backlog avec Jira . . . . .	24
1.22	Dépôt GitHub . . . . .	24
2.1	Diagramme des acteurs . . . . .	27
2.2	Diagramme de cas d'utilisation global . . . . .	30
2.3	Ecran de chargement - Web . . . . .	33
2.4	Ecran de chargement - Mobile . . . . .	33
2.5	Interface d'authentification - Web . . . . .	34
2.6	Interface d'authentification - Mobile . . . . .	34
2.7	Interface de profil- Web . . . . .	34
2.8	Interface de profil - Mobile . . . . .	34

## TABLE DES FIGURES

---

2.9	Interface de gestion des utilisateurs . . . . .	35
2.10	Interface de création d'un nouvel utilisateur . . . . .	35
2.11	map- Web . . . . .	36
2.12	map - Mobile . . . . .	36
2.13	la palette de couleurs . . . . .	37
2.14	logo . . . . .	37
2.15	Flux utilisateur : Définition des éléments . . . . .	38
2.16	Vue globale sur les flux d'utilisateurs . . . . .	38
2.17	Organigramme des interfaces du l'administrateur . . . . .	39
2.18	Organigramme des interfaces du chauffeur . . . . .	40
2.19	Organigramme des interfaces du mécanicien . . . . .	41
2.20	Organigramme des interfaces du chef d'équipe . . . . .	42
2.21	Diagramme de Gantt . . . . .	44
3.1	Diagramme de cas d'utilisation global du sprint 1 . . . . .	46
3.2	Diagramme de séquence système du cas d'utilisation «S'authentifier» - Web . . .	47
3.3	Diagramme de séquence système du cas d'utilisation «S'authentifier» - Mobile . .	48
3.4	Analyse de cas d'utilisation « Gérer les utilisateurs » . . . . .	50
3.5	Diagramme de séquence système du cas d'utilisation «Ajouter utilisateur» . . .	50
3.6	Interface d'authentification Web . . . . .	53
3.7	Echec de connexion . . . . .	53
3.8	Interface d'authentification Mobile . . . . .	54
3.9	Echec de connexion . . . . .	54
3.10	Interface de profil - Web . . . . .	54
3.11	Interface de profil - Mobile . . . . .	54
3.12	Interface de gestion des utilisateurs . . . . .	55
3.13	Interface de création d'un nouvel utilisateur . . . . .	56
3.14	Burdown Chart du Sprint 1 . . . . .	57
3.15	Task board du Sprint 1 . . . . .	57
3.16	Code souce du test unitaire du cas d'utilisation «S'authentifier » . . . . .	58
3.17	Résultat du premier test . . . . .	58
3.18	Résultat du deuxième test . . . . .	59
4.1	Diagramme de cas d'utilisation global du sprint 2 . . . . .	61
4.2	Raffinement de cas d'utilisation « Gérer les véhicules » . . . . .	61
4.3	Raffinement de cas d'utilisation « Ajout des véhicules » . . . . .	63
4.4	Raffinement de cas d'utilisation « Créer les tâches » . . . . .	64
4.5	Diagramme de séquence système du cas d'utilisation «Créer les tâches» . . . .	67
4.6	Analyse de cas d'utilisation « Consulter les tâches pour les chauffeurs » . . . .	68
4.7	Diagramme d'activité du cas d'utilisation « Consulter les tâches des chauffeurs »	69

## TABLE DES FIGURES

---

4.8	Analyse de cas d'utilisation « Consulter les taches pour les mécaniciens » . . . . .	70
4.9	interface de création des taches . . . . .	72
4.10	Création des taches pour les chauffeurs . . . . .	73
4.11	Création des taches pour les mécaniciens . . . . .	73
4.12	interface de consultation du checklist pour les chauffeurs . . . . .	74
4.13	interface de consultation des taches pour les mécaniciens . . . . .	74
4.14	Burdown Chart du Sprint 2 . . . . .	75
4.15	Task board du Sprint 2 . . . . .	75
4.16	Code souce du test unitaire du cas d'utilisation « Créer taches » . . . . .	76
4.17	Réultat du test de réussite . . . . .	76
4.18	Réultat du test d'échec . . . . .	77
5.1	Diagramme de cas d'utilisation global du sprint 3 . . . . .	79
5.2	Diagramme d'activité du cas d'utilisation « Consulter Notifications » . . . . .	83
5.3	Raffinement du cas d'utilisation « Générer des rapports » . . . . .	84
5.4	Diagramme de séquence système du cas d'utilisation « Générer des rapports » . .	85
5.5	Suivi en Temps Réel par Géolocalisation . . . . .	87
5.6	Vue du trajet . . . . .	87
5.7	Carte de voyage . . . . .	88
5.8	Interface de Consultation des notifications - web » . . . . .	88
5.9	Interface de Consultation des notifications - Mobile » . . . . .	88
5.10	Génération de Rapport - Interface Initiale . . . . .	89
5.11	Génération de Rapport - Détails Entrés . . . . .	89
5.12	Génération de Rapport - Rapport Sauvegardé . . . . .	89
5.13	Burdown Chart du Sprint 3 . . . . .	90
5.14	Task board du Sprint 3 . . . . .	90
5.15	Code souce du test unitaire du cas d'utilisation « Générer des rapports » . . . .	91
5.16	Réultat du test de réussite . . . . .	91
5.17	Réultat du test d'échec . . . . .	92

# Liste des tableaux

1.1	Description des machines de développement utilisées . . . . .	21
2.1	Backlog du produit . . . . .	33
3.1	Backlog du sprint 1 . . . . .	46
3.2	Description textuelle du cas d'utilisation “S'authentifier” . . . . .	49
3.3	Description textuelle du cas d'utilisation “Ajouter utilisateur” . . . . .	51
4.1	Backlog du sprint 2 . . . . .	60
4.2	Description textuelle du cas d'utilisation “Ajouter véhicules ” . . . . .	63
4.3	Description textuelle du cas d'utilisation “Créer des Tâches pour les Mécaniciens ”	65
4.4	Description textuelle du cas d'utilisation “Créer des Tâches pour les chauffeurs ”	66
4.5	Description textuelle du cas d'utilisation “Consulter les taches des chauffeurs” . .	69
4.6	Description textuelle du cas d'utilisation “Consulter les tâches des mécaniciens” .	71
5.1	Backlog du sprint 3 . . . . .	78
5.2	Description textuelle du cas d'utilisation “Suivi en Temps Réel par Géolocalisation”	81
5.3	Description textuelle du cas d'utilisation “ Planifier les déplacements ” . . . . .	82
5.4	Description textuelle du cas d'utilisation “ Consulter Notifications ” . . . . .	83
5.5	Description textuelle du cas d'utilisation “ Générer des rapports ” . . . . .	85

# Introduction générale

Dans le contexte actuel de la gestion d'entreprise axée sur l'efficacité opérationnelle et la rentabilité, l'optimisation des ressources et des actifs est devenue un impératif stratégique pour les organisations modernes. Parmi ces actifs essentiels, les parcs d'entreprise, qu'ils comprennent des véhicules, des flottes de transport ou d'autres équipements, représentent des investissements substantiels nécessitant une gestion rigoureuse et proactive.

Face à cette exigence croissante, notre projet s'attache à répondre aux besoins de gestion des parcs d'entreprise à travers le développement d'une application mobile novatrice et fonctionnelle, intégrant les principes de la GMAO (Gestion de Maintenance Assistée par Ordinateur). Cette solution numérique vise à rationaliser les processus de gestion en offrant aux parties prenantes une plateforme intégrée pour surveiller, entretenir et optimiser les parcs d'entreprise de manière efficace et transparente.

Dans le cadre de ce rapport, nous présentons une analyse approfondie du développement de cette application, mettant en lumière nos objectifs stratégiques, les fonctionnalités clés de l'application, ainsi que les défis rencontrés et les stratégies d'atténuation mises en œuvre. Nous explorerons également les implications potentielles de cette solution pour les entreprises utilisatrices, notamment en termes d'amélioration de l'efficacité opérationnelle, de réduction des coûts et d'optimisation des performances.

# Chapitre 1

## Etude préliminaire du projet

### Introduction

Dans le cadre de la mise en place d'une application de gestion des parkings des entreprises, il est essentiel de réaliser une étude préalable approfondie afin de cerner les besoins, les contraintes et les objectifs liés à ce projet. Cette étude préalable permettra de définir les contours du projet et de poser les bases pour son développement et sa mise en œuvre. Nous commençons par placer le projet dans son cadre général et d'exposer le contexte de travail ainsi que les objectifs à atteindre.

#### 1.1 Présentation de l'organisme d'accueil

Digital Identity, connue sous l'abréviation « DIGID » est une entreprise tunisienne Fondée en 2019, spécialisée dans le conseil et le développement de solutions technologiques spécifiques à l'échelle nationale et internationale. Ses objectifs sont de permettre à ses clients de se concentrer sur leurs activités principales, en s'appuyant sur des solutions fiables et sur un partenaire crédible et inébranlable.

DIGID est spécialisée dans le conseil et la mise en œuvre de solutions logicielles de gestion intégrée.



FIGURE 1.1 – Logo de l'entreprise « DIGITAL IDENTITY »

## CHAPITRE 1. ETUDE PRÉLIMINAIRE DU PROJET

### 1.1.1 Organigramme de l'entreprise

L'organigramme de Digital Identity figure 1.2 présente une vue parfaite de l'organisation des liens hiérarchiques entre les différentes équipes. En effet, c'est une traduction schématique des objectifs, des missions et des relations fonctionnelles.

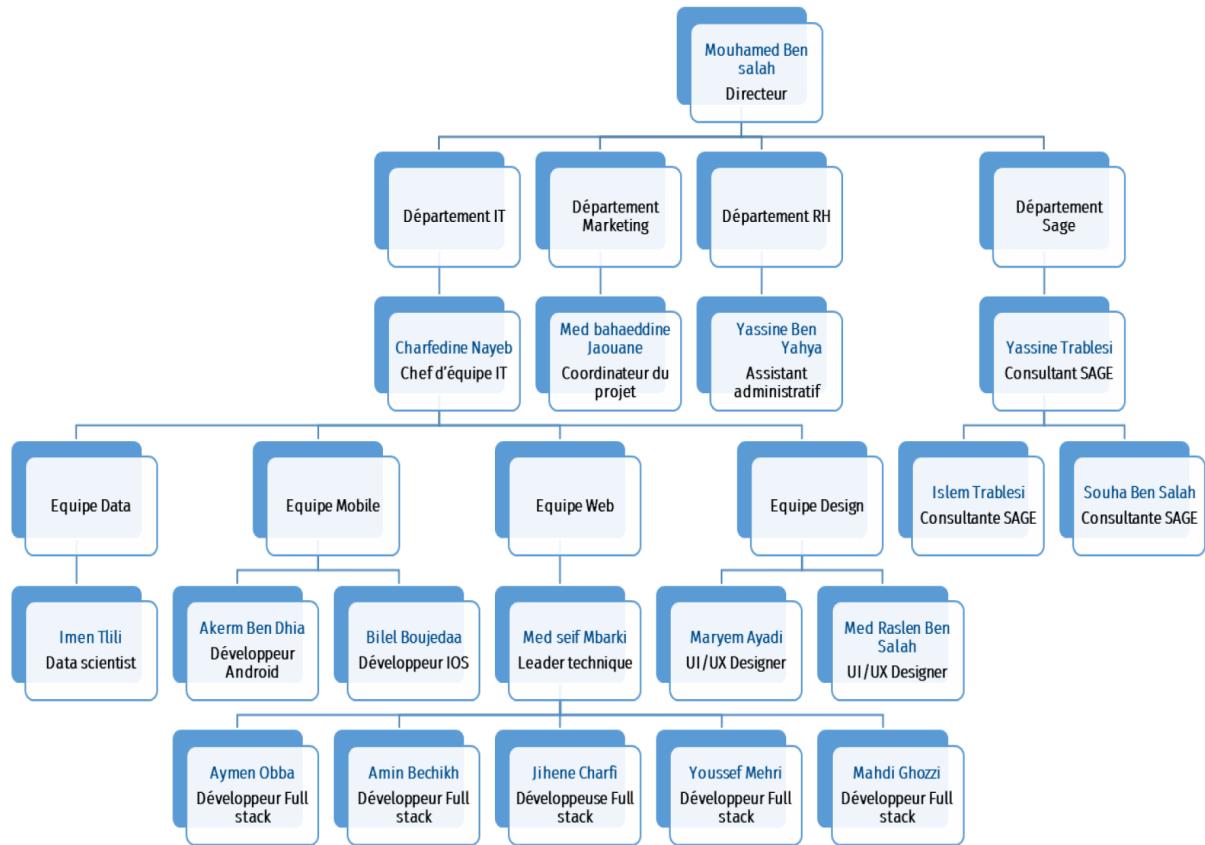


FIGURE 1.2 – Organigramme de Digital identity

### 1.1.2 Les activités de Digital Identity :



FIGURE 1.3 – Les activités de Digital Identity

### **1.2 Cadre du projet**

Dans cette section, nous allons aborder en détail la problématique soulevée et nous allons présenter une solution pertinente qui a été proposée pour y remédier.

#### **1.2.1 Problématique**

À une époque où la gestion des parcs de véhicules d'entreprise est devenue de plus en plus complexe en raison de la diversité des véhicules, des itinéraires et des exigences opérationnelles, ainsi que des pressions technologiques et environnementales croissantes, il devient impératif de concevoir, développer et déployer une plateforme intégrée véritablement innovante et robuste pour relever ces défis multidimensionnels. Comment mettre en place une telle plateforme, capable de fournir une surveillance en temps réel, une analyse prédictive et des fonctionnalités d'optimisation avancées, tout en garantissant une efficacité opérationnelle maximale, une réduction des coûts et une durabilité environnementale accrue pour les entreprises ? De plus, comment concilier ces objectifs avec les exigences de sécurité, de conformité réglementaire et de flexibilité, afin de permettre aux entreprises de prospérer dans un contexte économique en perpétuelle mutation, de maintenir leur compétitivité et d'assurer une gestion stratégique de leurs actifs de transport dans un monde en constante évolution ?

#### **1.2.2 Solution proposée**

Notre solution de gestion de flotte de véhicules d'entreprise repose sur une plateforme intégrée et technologiquement avancée, conçue pour répondre aux défis croissants de gestion des parcs de véhicules. Dans le cadre de cette solution, nous avons intégré une dimension Business Intelligence (BI) afin d'offrir une analyse approfondie des données et des informations stratégiques aux gestionnaires. Cette intégration du volet BI enrichit notre plateforme en lui permettant de fournir une surveillance en temps réel des véhicules, une analyse prédictive des données et des outils d'optimisation avancés.

Grâce à des tableaux de bord interactifs et des rapports personnalisables, les gestionnaires auront une vue d'ensemble précise de la performance opérationnelle de leur flotte. Ces outils leur permettront de visualiser et de suivre en temps réel des indicateurs clés de performance (KPI) tels que le taux d'utilisation des véhicules, les coûts de maintenance, les temps d'immobilisation, etc. De plus, les rapports générés par le système BI fourniront des insights stratégiques pour la prise de décision, en identifiant les tendances, les patterns et les opportunités d'optimisation.

En combinant ces fonctionnalités avancées avec une approche centrée sur les besoins des entreprises, notre solution vise à fournir une réponse complète et efficace aux défis de gestion des flottes de véhicules d'entreprise. Elle permettra aux gestionnaires de prendre des décisions éclairées et proactives pour maximiser l'efficacité opérationnelle de leur flotte, réduire les coûts associés et maintenir leur compétitivité dans un environnement commercial en constante évolution.

### **1.3 Méthodologie et langage de conception**

La conception est cruciale dans le développement d'un système informatique pour répondre aux besoins du client. Les choix conceptuels concernant l'approche, le langage de modélisation et le processus de développement sont issus de réflexions collectives. Nous avons choisi la méthodologie Scrum, une méthode agile, pour réaliser notre projet.

#### **1.3.1 Méthodes agiles**

Parmi les méthodes agiles les plus couramment utilisées de nos jours, on peut citer :

- Extrême Programming (XP)
- Scrum
- Agile Unified Process (Agile UP ou AUP)

Ces méthodes agiles se caractérisent par leur approche itérative et incrémentale, leur focalisation sur les besoins des utilisateurs, leur flexibilité pour répondre aux changements, leur collaboration étroite avec les parties prenantes du projet, ainsi que leur orientation vers une livraison régulière de fonctionnalités à valeur ajoutée. Chacune de ces méthodes agiles a ses propres pratiques, rôles, cérémonies et outils, mais elles partagent toutes les mêmes valeurs et principes fondamentaux de l'agilité.

Dans notre cas ,nous avons choisi d'adopter le framework Scrum pour le développement de notre application, car elle nous a semblé la plus adaptée à notre travail. Scrum est un framework agile qui offre une organisation flexible nous permettant de découper notre projet en plusieurs sprints. Cette approche itérative nous aidera à mieux gérer notre travail en nous concentrant sur des objectifs concrets à atteindre à chaque étape du développement.

### 1.3.2 Principes du framework Scrum

Scrum est un cadre de travail pour la gestion de projet agile qui est largement utilisé à travers le monde. Il comprend des rôles bien définis, un rythme itératif, des réunions chronométrées et des artefacts tels que le product backlog, le sprint backlog et le graphique d'avancement (Burndown Chart). Ces éléments constituent les piliers du processus de gestion de projet agile offert par Scrum.

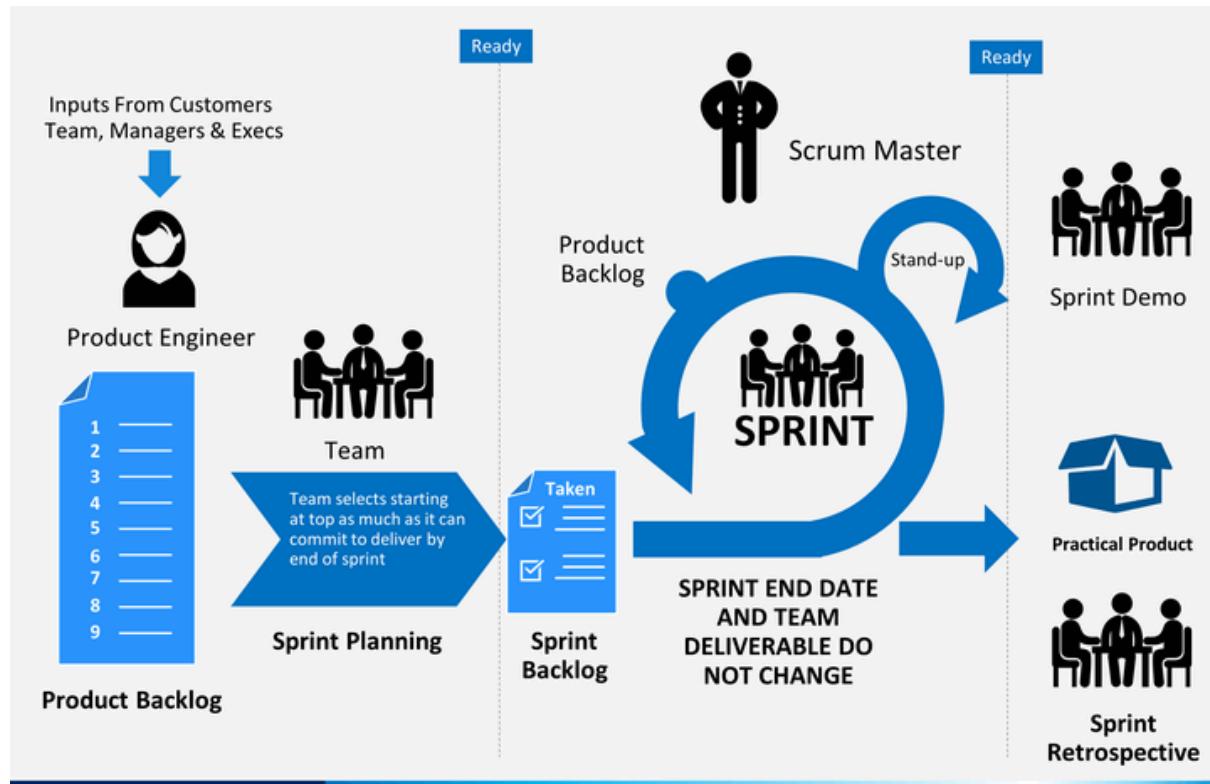


FIGURE 1.4 – Mode de fonctionnement de la méthodologie Scrum

**Sprint** : Tout projet Scrum est organisé autour de «Sprints» de développement pendant lesquels une équipe de développement travaille sur un ensemble de fonctionnalités ou d'objectifs spécifiques.

**User story** : Les fonctionnalités requises sont présentées sous la forme de "User Stories" dans une liste organisée.

**Backlog du produit** : Les user stories sont organisées dans le backlog du produit.

**Sprint backlog** : Le sprint backlog est une liste des éléments du product backlog sélectionnés pour être développés pendant un sprint exacte. C'est une liste de tâches à accomplir pendant le sprint. Les membres de l'équipe de développement se concentrent sur ces tâches pour les livrer à la fin du sprint.

**Planning poker** : Pendant une réunion nommée "Réunion du Planning Poker", les user stories de chaque sprint sont estimées en points et priorisées.

**La mêlée quotidienne** : La mêlée quotidienne permet aux membres de l'équipe de se synchroniser régulièrement, de signaler rapidement les problèmes et les défis qu'ils rencontrent et de suivre

## **CHAPITRE 1. ETUDE PRÉLIMINAIRE DU PROJET**

---

l'avancement du sprint en temps réel. Cette réunion rapide et efficace est un moyen pour l'équipe de travailler de manière collaborative et d'assurer la réussite du projet.

**Revue de sprint :** L'objectif de la revue de sprint est d'inspecter l'incrément produit au cours du sprint fini.

**Rétrospective :** La rétrospective est une réunion organisée à la fin de chaque sprint pour évaluer ce qui a bien fonctionné et ce qui peut être amélioré pour le prochain sprint. L'objectif est d'améliorer continuellement les processus de travail et la collaboration de l'équipe Scrum.

### **1.3.3 Rôles du framework Scrum**

Il existe trois rôles dans l'organisation d'un projet agile suivant la méthodologie Scrum :

- **Product Owner :** Le Product Owner est celui qui porte la vision du produit à réaliser et travaille en interaction avec les Développeurs. Il apporte ses connaissances pour aider l'équipe de développement à comprendre les besoins et les attentes du client, tout en veillant à ce que le produit final réponde à ces exigences.
- **Scrum Master :** Le Scrum Master est un rôle clé dans le cadre de la méthode Scrum. Il est responsable de s'assurer que l'équipe Scrum suit les principes et les pratiques de Scrum. Le Scrum Master facilite les réunions et les événements Scrum, et aide l'équipe à résoudre les obstacles ou les problèmes qui peuvent les entraver .
- **Scrum Team :** Équipe autogérée et multidisciplinaire constituée de développeurs, testeurs... chargés de transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables.

### **1.3.4 Langage de modélisation**

Afin de visualiser la conception de notre système, nous avons utilisé le langage de modélisation unifié UML (Unified Modeling Language) qui permet de modéliser les besoins du logiciel à développer.



FIGURE 1.5 – UML

### **1.4 Environnement de travail**

Dans cette section, nous détaillons l'environnement de travail à savoir l'environnement matériel, l'environnement logiciel ainsi que les langages et framework utilisés.

#### **1.4.1 Langages et framework utilisés**

Les langages et Frameworks utilisés pour l'implémentation de notre solution sont les suivants :

- **Java** : « C'est un Langage de programmation polyvalent et orienté objet, largement utilisé pour le développement d'applications Android et d'entreprises en raison de sa portabilité et de sa robustesse»



FIGURE 1.6 – Java

- **Python** : « C'est un Langage de programmation populaire connu pour sa simplicité syntaxique, sa polyvalence et sa grande variété de bibliothèques, idéal pour le développement rapide d'applications, l'analyse de données et l'automatisation des tâches.»

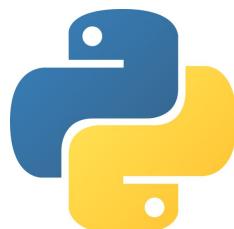


FIGURE 1.7 – Python

- **MySQL** : « C'est un système de gestion de base de données relationnelle open source largement utilisé dans le développement d'applications web et mobiles en raison de sa fiabilité, de sa performance et de sa facilité d'utilisation. Si vous avez besoin d'assistance pour intégrer MySQL à votre application ou pour d'autres aspects de votre projet.»



FIGURE 1.8 – MySQL

## CHAPITRE 1. ETUDE PRÉLIMINAIRE DU PROJET

---

- **ReactJS** : « C'est une bibliothèque JavaScript open-source développée par Facebook pour créer des interfaces utilisateur interactives. Elle permet de construire des composants réutilisables et optimise les mises à jour de l'interface grâce au Virtual DOM.»



FIGURE 1.9 – ReactJS

- **Node.js** : « C'est une plateforme d'exécution JavaScript construite sur le moteur V8 de Chrome, permettant d'exécuter du code JavaScript côté serveur. Il est réputé pour son modèle non bloquant et basé sur des événements, idéal pour des applications réseau performantes.»

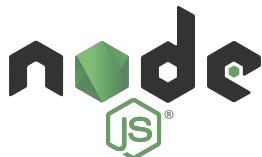


FIGURE 1.10 – Node.js

- **REST API :**

« Une API REST (Representational State Transfer) est une interface de programmation d'application qui utilise les méthodes HTTP standard pour permettre aux clients d'accéder et de manipuler des ressources sur un serveur. »

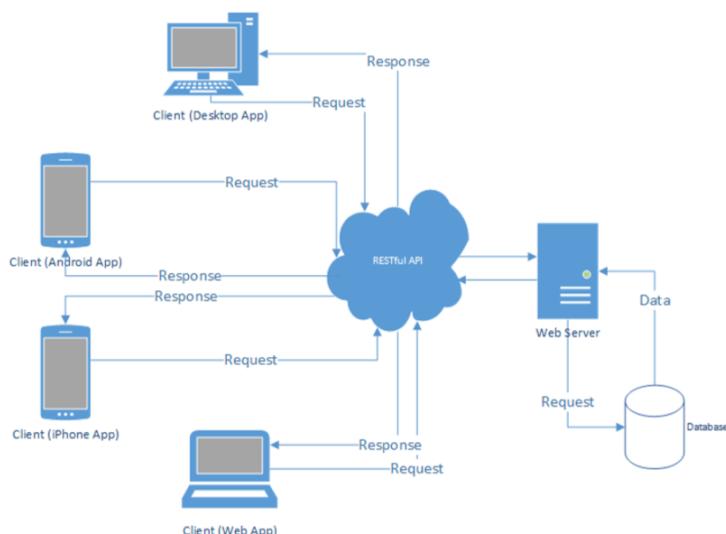


FIGURE 1.11 – REST API

### **1.4.2 Environnement matériel**

Le tableau 1.1 montre les caractéristiques techniques des machines utilisées pour la réalisation de notre application.

Ordinateur	ASUS TUF F15	ASUS TUF F15
Propriétaire	Balti Firas	Mimouni Med Aziz
RAM	32 GO	32 GO
Processeur	i5 11éme	i5 11éme
Système d'exploitation	Windows 11	windows 11

TABLE 1.1 – Description des machines de développement utilisées

### **1.4.3 Environnement logiciel**

Les logiciels utilisés pour l'implémentation de notre solution sont les suivants :

- **Figma** : C'est un outil de conception d'interface utilisateur (UI) et d'expérience utilisateur (UX) en ligne. Il permet aux designers de créer des designs d'interface utilisateur . Figma est également utilisé pour collaborer en temps réel. Nous l'avons utilisé dans notre projet pour concevoir les maquettes et les diagrammes de notre application.



FIGURE 1.12 – Figma

- **Android Studio** : C'est un environnement de développement intégré (IDE) développé par Google pour la création d'applications Android. Il est disponible gratuitement pour les développeurs Android. Nous l'avons utilisé dans notre cas pour bénéficier d'un émulateur android servant à déboguer notre application.



FIGURE 1.13 – Android Studio

## **CHAPITRE 1. ETUDE PRÉLIMINAIRE DU PROJET**

---

- **Overleaf** : C'est un éditeur en ligne de LaTeX, un langage de composition de documents scientifiques et techniques. Il permet aux utilisateurs de créer, de modifier et de collaborer sur des documents LaTeX en temps réel, Tout au long de notre stage de projet de fin d'étude, nous l'avons utilisé pour travailler ensemble sur un seul document et suivre les modifications apportées par chacune de nous en temps réel.



FIGURE 1.14 – Overleaf

- **Spyder** : C'est un environnement de développement interactif (IDE) spécialement conçu pour Python, offrant une interface conviviale et des fonctionnalités avancées telles que l'édition de code, l'exploration de variables et la gestion de projets. Nous avons choisi d'utiliser Spyder dans notre cas pour sa simplicité d'utilisation et sa compatibilité avec Python, ce qui nous a permis de développer et de déboguer efficacement notre code .



FIGURE 1.15 – Spyder

- **Jira** : C'est un outil de gestion de projet, conçu pour aider les équipes à planifier, suivre et gérer leurs tâches et leurs projets. Il offre une interface conviviale pour la création de backlogs, l'attribution de tâches, le suivi des progrès et la gestion des délais. Grâce à ses fonctionnalités de collaboration , Jira permet aux équipes de travailler de manière coordonnée et de respecter les délais de manière transparent.



FIGURE 1.16 – Jira

## **CHAPITRE 1. ETUDE PRÉLIMINAIRE DU PROJET**

---

- **Slack** : C'est une plateforme de communication collaborative propriétaire (SaaS). Nous avons utilisé Slack lors de notre période de stage pour assurer et faciliter la communication avec l'équipe de l'entreprise d'accueil.



FIGURE 1.17 – Slack

- **Material UI** : C'est une bibliothèque de composants React qui facilite la création d'interfaces utilisateurs avec des éléments préconçus basés sur les principes du Material Design de Google. Elle permet une personnalisation poussée et un développement rapide.



FIGURE 1.18 – Material UI

- **Visme** : C'est une plateforme de création visuelle en ligne, nous avons utilisé cette outil pour la création de nos burndown charts.



FIGURE 1.19 – Visme

- **Postman** : C'est un outil qui simplifie le test et le développement d'API en permettant d'envoyer des requêtes HTTP et d'analyser les réponses via une interface utilisateur conviviale.



FIGURE 1.20 – Postman

## CHAPITRE 1. ETUDE PRÉLIMINAIRE DU PROJET

### 1.5 Organisation du travail

Pour assurer une organisation optimale tout au long du développement de notre projet, nous avons mis en place une série de processus et de pratiques collaboratives. Nous avons débuté par des séances de brainstorming, où nous avons généré et partagé des idées. Ces propositions ont ensuite été soumises à des réunions quotidiennes avec notre encadrant professionnel pour validation et conseils. En parallèle, nous avons consulté notre encadrant académique pour le filtrage et le développement des idées. Une fois la phase de conception achevée, nous avons entamé la création du backlog sur Jira, ce qui nous a permis de planifier et d'organiser efficacement les différentes tâches à réaliser. Ensuite, nous avons progressé vers la phase de codage.

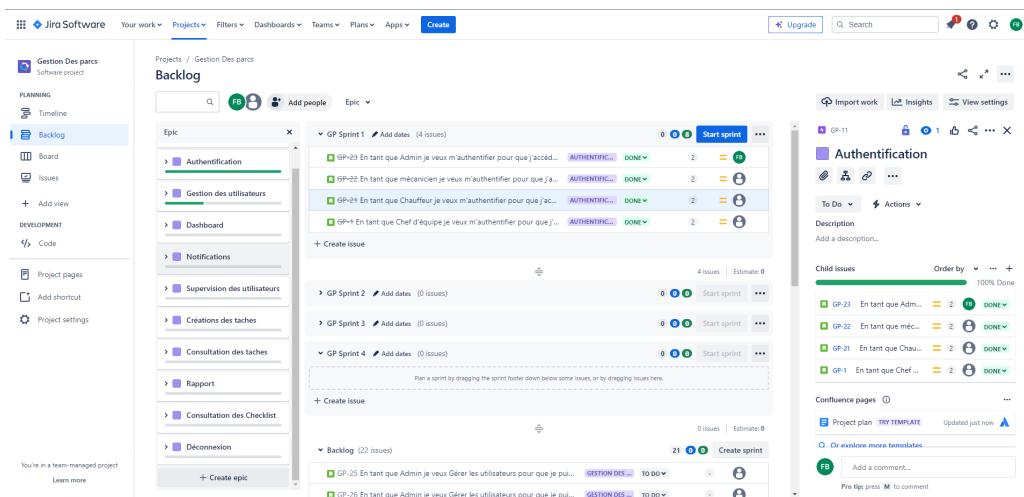


FIGURE 1.21 – Création du backlog avec Jira

Pour faciliter la collaboration et le partage du code en temps réel, nous avons opté pour l'utilisation de GitHub. Chacun de nous a pu "pusher" ses changements une fois les tâches accomplies, ce qui nous a permis de suivre l'avancement de chacun et de garantir le respect des délais de réalisation de nos sprints.

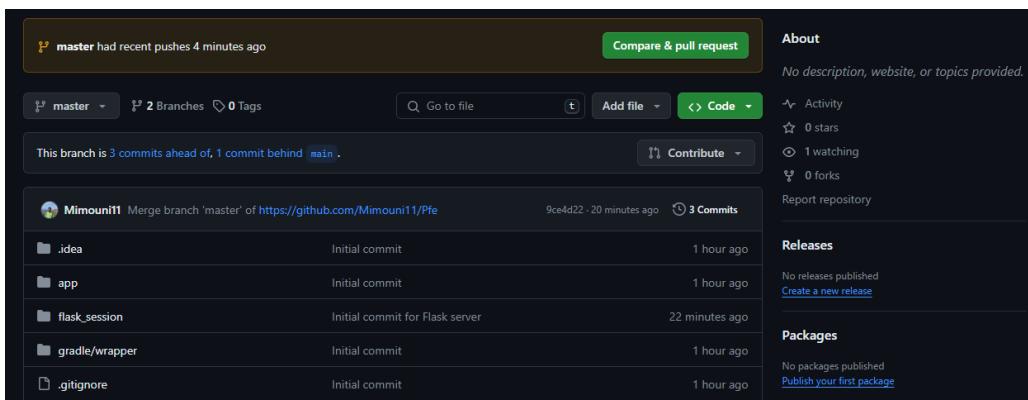


FIGURE 1.22 – Dépôt GitHub

## **Conclusion**

Dans ce chapitre, nous avons introduit l'organisme d'accueil, clarifié le cadre du projet et décrit les choix méthodologiques ainsi que l'architecture. Le prochain chapitre se concentrera sur la planification du projet et la définition des fonctionnalités de l'application.

## Chapitre 2

# Planification du projet

Ce chapitre a pour objectif de présenter les différents acteurs impliqués dans le projet, les besoins fonctionnels et non fonctionnels requis par le client, ainsi que le backlog du produit. En d'autres termes, nous allons fournir une vue d'ensemble des éléments clés nécessaires pour comprendre le projet dans son ensemble.

## 2.1 Spécification des besoins

Dans cette partie du chapitre, nous allons spécifier les différents acteurs et les divers besoins fonctionnels et non fonctionnels de notre application.

### 2.1.1 Identification des acteurs

Cette étape vise à identifier les acteurs interagissant directement avec le système étudié. Un acteur représente un rôle joué par une entité externe, pouvant être un utilisateur humain, un dispositif matériel ou un autre système. Notre système est destiné à être utilisé par quatre profils d'acteurs. La figure 2.1 présente les acteurs dans notre application :

- **Administrateur** : L'administrateur est responsable de la gestion globale du système. Il a des priviléges étendus, y compris la configuration du système, la gestion des utilisateurs et des autorisations, ainsi que la surveillance et la maintenance du système dans son ensemble.
- **Chef d'équipe** : Le chef d'équipe supervise à la fois les chauffeurs et les mécaniciens, assurant ainsi la coordination efficace des opérations sur le terrain et dans l'atelier. Il utilise l'application pour assigner des tâches, suivre les progrès des chauffeurs et des mécaniciens, gérer les horaires et les itinéraires, ainsi que pour recevoir des alertes sur les problèmes ou les retards.
- **Chauffeur** : Le chauffeur est l'utilisateur chargé de conduire les véhicules de l'entreprise pour effectuer diverses tâches, telles que le transport de marchandises ou la réalisation de missions spécifiques. Il utilise l'application pour signaler son statut, ses itinéraires, et pour

## **CHAPITRE 2. PLANIFICATION DU PROJET**

---

recevoir des notifications sur les tâches à effectuer.

- **Mécanicien :** Le mécanicien est chargé de la maintenance et de la réparation des véhicules de l'entreprise. Il utilise l'application pour signaler les problèmes, demander des pièces de rechange, et pour recevoir des tâches de maintenance préventive ou corrective.



FIGURE 2.1 – Diagramme des acteurs

### **2.1.2 Besoins fonctionnels**

La spécification des besoins fonctionnels détermine les exigences fonctionnelles ainsi que les principaux objectifs et fonctionnalités de l'application. En d'autres termes, elle spécifie ce que le système étudié doit faire.

- **S'authentifier :** Toute personne inscrite devra s'authentifier en saisissant son nom d'utilisateur ainsi que son mot de passe afin d'accéder aux fonctionnalités du statut sous lequelle elle est inscrite.
- **Gérer profil :** Chaque acteur qu'il soit administrateur, chef d'équipe, chauffeur ou même mécanicien peut gérer son propre compte.
- **Gestion des utilisateurs :** Permet à l'administrateur de gérer les comptes des utilisateurs de l'application, y compris la création, la modification et la suppression des comptes. Lors de la création d'un utilisateur (chef d'équipe, chauffeur ou mécanicien), un email contenant les paramètres de connexion ainsi qu'un manuel d'utilisation de l'application en PDF, adapté à chaque type d'utilisateur, sera envoyé automatiquement.
- **Gestion des rôles :** Permet à l'administrateur d'attribuer des rôles spécifiques à chaque utilisateur, déterminant ainsi leurs permissions et leurs accès au système.
- **Gestion des Véhicules :** L'administrateur peut ajouter de nouveaux véhicules, supprimer des véhicules existants et visualiser la liste complète des véhicules dans le système .
- **Notifications :** Les notifications sont catégorisées (accidents, maintenance, demande de pièce ...) pour une gestion proactive des événements.
- **Création des tâches :** Permet à l'administrateur de créer, attribuer, suivre et gérer les tâches assignées aux chauffeurs, mécaniciens et chefs d'équipe.
- **Suivi des tâches :** Permet au chef d'équipe de suivre l'état et la progression des tâches assignées à son équipe, en vérifiant leur avancement et en s'assurant de leur complétion.
- **Surveillance en temps réel des déplacements des véhicules :** Donne au chef d'équipe la capacité de surveiller les déplacements en temps réel des véhicules de l'entreprise, afin de coordonner efficacement les opérations.

- **Consultation de checklist** : Permet au chauffeur de consulter les listes de vérification nécessaires avant de commencer un voyage ou une mission, garantissant ainsi la conformité et la sécurité .
- **Planification des voyages** : Autorise le chauffeur à planifier ses itinéraires et ses voyages à l'avance, en tenant compte des horaires, des destinations et d'autres paramètres pertinents.
- **Confirmation des tâches** : Permet au chauffeur de confirmer la réalisation des tâches qui lui sont assignées, telles que la livraison de marchandises ou la maintenance préventive des véhicules.
- **Consultation des tâches** : Donne au mécanicien la possibilité de consulter les tâches de maintenance assignées, y compris les réparations à effectuer et les inspections à réaliser..
- **Rapport** : Permet au mécanicien de générer des rapports détaillés sur les travaux effectués, les pièces utilisées et l'état général des véhicules, contribuant ainsi à la gestion proactive de la maintenance.

### 2.1.3 Besoins non fonctionnels

La définition des besoins non fonctionnels est cruciale pour déterminer les caractéristiques et la qualité requises du système. Ils n'affectent pas directement les fonctionnalités de l'application, mais peuvent avoir un impact sur sa performance et son efficacité globale. Il est important de ne pas les négliger lors de la conception et du développement de l'application. Les principaux besoins non fonctionnels de cette application sont :

- **Simplicité d'utilisation** : Assurer une interface utilisateur intuitive et conviviale pour toutes les catégories d'utilisateurs, minimisant ainsi la courbe d'apprentissage et favorisant l'adoption de l'application.
- **Sécurité des données** : Assurer la confidentialité, l'intégrité et la disponibilité des données sensibles telles que les informations sur les véhicules, les tâches et les utilisateurs.
- **Évolutivité** : Concevoir l'application pour qu'elle puisse s'adapter à l'augmentation du nombre d'utilisateurs, de véhicules et de fonctionnalités sans compromettre les performances ou la fiabilité du système.
- **Disponibilité** : Assurer la disponibilité continue de l'application, minimisant ainsi les temps d'arrêt et les interruptions des opérations.

### 2.1.4 Diagramme de cas d'utilisation

La figure 2.2 illustre le diagramme de cas d'utilisation général de notre application.

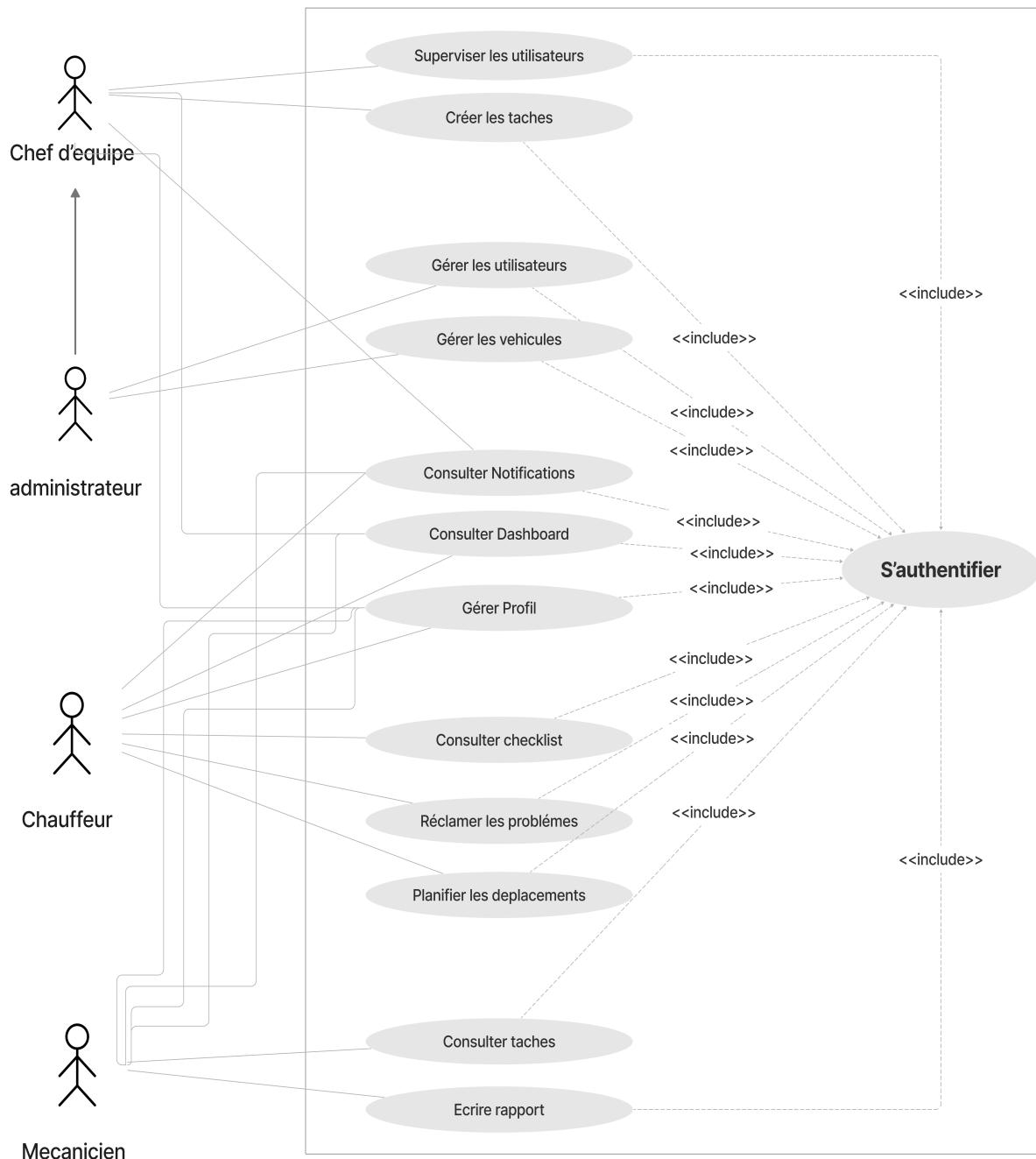


FIGURE 2.2 – Diagramme de cas d'utilisation global

### **2.2 Pilotage du projet avec Scrum**

Cette section comportant l'équipe du projet et ses rôles, le backlog du produit et le découpage du projet en sprint, présente le pilotage du projet avec Scrum .

#### **2.2.1 Équipe et rôles**

L'équipe responsable du pilotage du projet est composée de :

- **Product owner :** Digital Identity.
- **Scrum master :** C'est un membre de l'équipe de développement de Digid Mr Baha jaouane .
- **Scrum team :** C'est l'équipe de développement qui englobe nous mêmes Mimouni med aziz et Balti firas.

#### **2.2.2 Contraintes et exigences**

La réussite du développement et de l'implémentation d'une solution technique dépend de l'identification des besoins réels des parties prenantes. Certaines fonctionnalités et besoins sont plus importants que d'autres, et il est crucial de les réaliser en premier pour assurer la continuité et la clarté de l'application. Pour cela, il est essentiel d'établir un ordre de priorité afin de coordonner les tâches à effectuer. Ce processus implique de classer les besoins en trois catégories selon leur importance : élevée, moyenne et faible, tout au long du cycle de vie de la solution technique.

#### **2.2.3 Risques**

Les principaux obstacles qui peuvent entraver notre progression sont la complexité de l'application et les exigences à respecter. Il est également important de prendre en compte la représentation des fonctionnalités similaires au cours de deux sprints consécutifs, afin d'éviter toute redondance inutile. Pour minimiser les risques, il est donc préférable de regrouper ces fonctionnalités similaires ensemble, plutôt que de les aborder séparément.

#### **2.2.4 Backlog du produit**

Le backlog du produit (Table 2.1) montre la liste des exigences et des fonctionnalités qui doivent être réalisées pour un produit donné ainsi que leurs degrés de priorité et de risque. Il sert de document de référence pour l'équipe de développement, les parties prenantes et les clients, afin de s'assurer que les travaux sont bien alignés sur les objectifs du projet et les besoins des utilisateurs finaux.

## CHAPITRE 2. PLANIFICATION DU PROJET

---

ID	USER STORIES	PRIORITE	RISQUE
<b>US1</b>	En tant qu'administrateur,, chauffeur, chef d'équipe et mécanicien je veux m'authentifier pour que j'accède a mon espace.	Elevée	Faible
<b>US2</b>	En tant qu'administrateur,, chauffeur, chef d'équipe et mécanicien je veux me déconnecter pour que je quitte l'application.	Elevée	Faible
<b>US3</b>	En tant qu'administrateur,, chauffeur, chef d'équipe et mécanicien je veux modifier mon profil.	Elevée	Faible
<b>US4</b>	En tant qu'administrateur, je veux pouvoir gérer les rôles des utilisateurs, afin de contrôler l'accès aux fonctionnalités de l'application	Elevée	Faible
<b>US5</b>	En tant qu'administrateur, je veux ajouter, supprimer et visualiser les véhicules .	Elevée	Faible
<b>US6</b>	En tant qu'administrateur, je veux Gérer les utilisateurs pour que je puisse ajouter/modifier/supprimer des utilisateurs.	Elevée	Faible
<b>US7</b>	En tant que Chef d'équipe je veux suivre le progrès et la localisation des voiture .	Elevée	Faible
<b>US8</b>	En tant que Chef d'équipe je veux Créer des taches pour les chauffeurs et les mécanicien .	Elevée	Faible
<b>US9</b>	En tant que Mécanicien je veux Consulter les détails des interventions .	Elevée	Faible
<b>US10</b>	En tant que Chauffeur je veux consulter la checklist avant et après le départ .	Elevée	Faible
<b>US11</b>	En tant qu'administrateur, chauffeur, mécanicien ou chef d'équipe, je veux recevoir des notifications pour être informé en temps réel des mises à jour.	Elevée	Faible

## **CHAPITRE 2. PLANIFICATION DU PROJET**

---

<b>US12</b>	En tant que Mécanicien je veux Générer des rapports pour que je puisse documenter les détails de chaque intervention.	Elevée	Faible
<b>US13</b>	En tant qu'administrateur, je veux avoir accès à un tableau de bord pour que je puisse superviser l'ensemble des opérations.	Elevée	Faible
<b>US14</b>	En tant que Chef d'équipe, je veux un tableau de bord pour suivre les déplacements en temps réel des chauffeurs et gérer les missions de manière proactive.	Elevée	Faible
<b>US15</b>	En tant que Chauffeur je veux Consulter un tableau de bord pour que je puisse avoir une vue claire de mes missions actuelles	Elevée	Faible
<b>US16</b>	En tant que Mécanicien, je veux consulter un tableau de bord pour avoir une vue d'ensemble des opérations de maintenance.	Elevée	Faible

TABLE 2.1 – Backlog du produit

### **2.3 Conception de l'application**

Dans cette section, nous présentons les prototypes de l'applications , la charte graphique, le plan de l'application ainsi que le diagramme de classes du domaine et le diagramme de gantt.

#### **2.3.1 Prototypes de l'application**

Les figure 2.3 et 2.4 illustrent les écrans de chargement dans l'application.

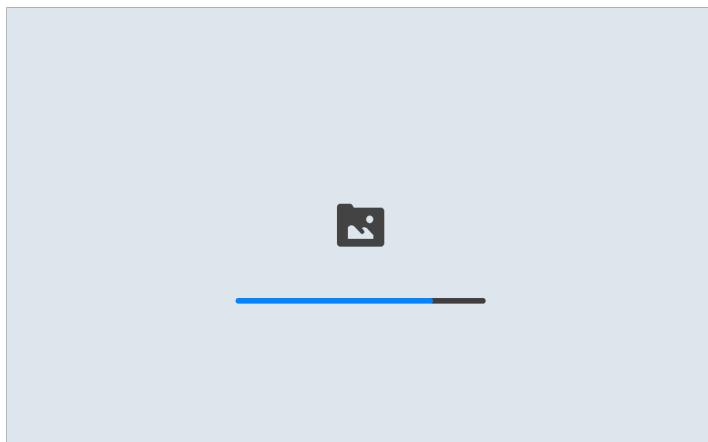


FIGURE 2.3 – Ecran de chargement - Web

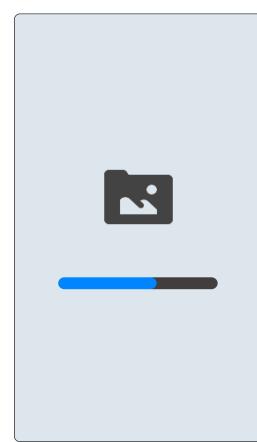


FIGURE 2.4 – Ecran de chargement - Mobile

## CHAPITRE 2. PLANIFICATION DU PROJET

Les figure 2.5 et 2.6 illustrent les pages permettant à l'utilisateur de s'authentifier dans l'application.

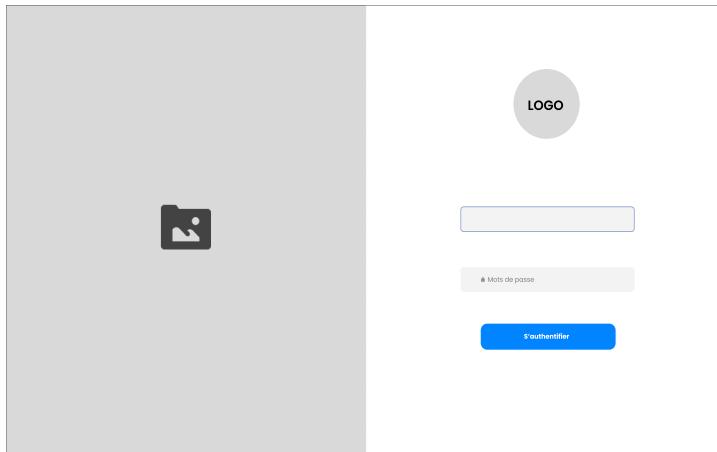


FIGURE 2.5 – Interface d'authentification - Web

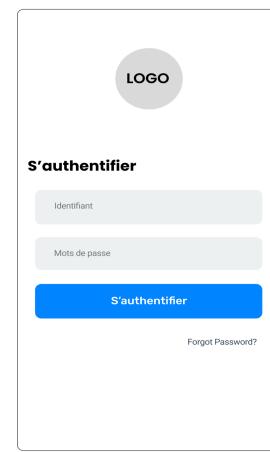


FIGURE 2.6 – Interface d'authentification - Mobile

Les figure 2.7 et 2.8 illustrent les pages permettant à l'utilisateur de modifier leur profils.

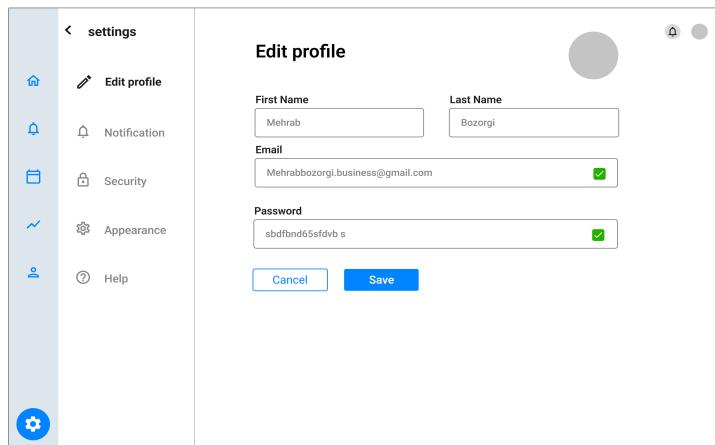


FIGURE 2.7 – Interface de profil- Web

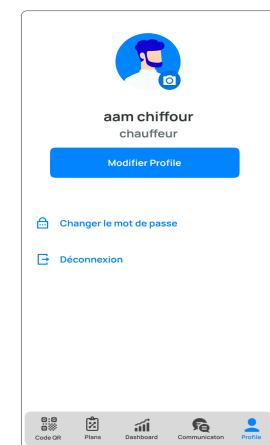


FIGURE 2.8 – Interface de profil - Mobile

## CHAPITRE 2. PLANIFICATION DU PROJET

---

En plus, les figure 2.9 et 2.10 illustrent les interfaces de gestion des utilisateurs par l'administrateur.

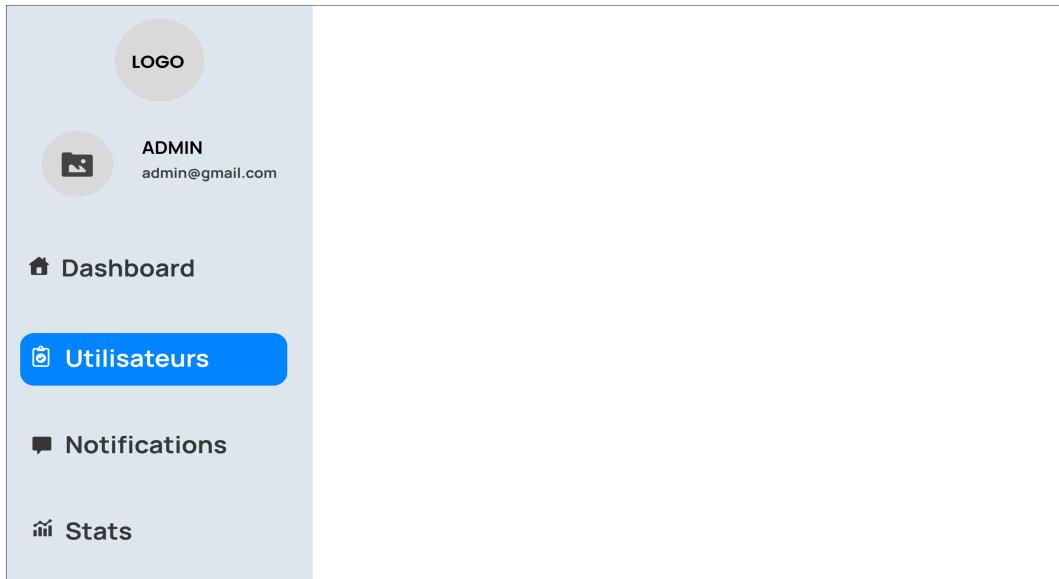


FIGURE 2.9 – Interface de gestion des utilisateurs

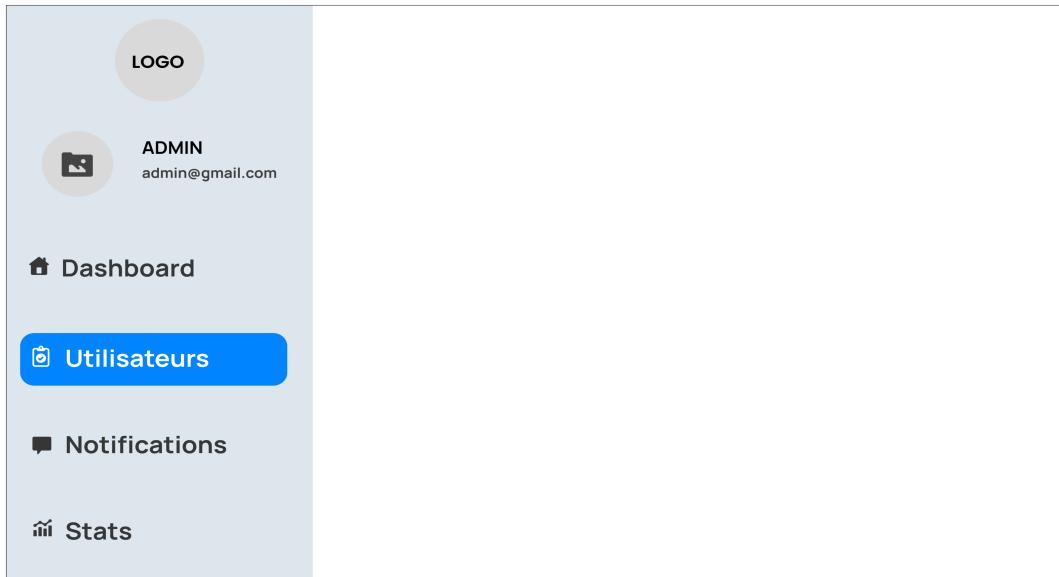


FIGURE 2.10 – Interface de création d'un nouvel utilisateur

## CHAPITRE 2. PLANIFICATION DU PROJET

---

Les figure 2.11 et 2.12 illustrent les pages permettant ....

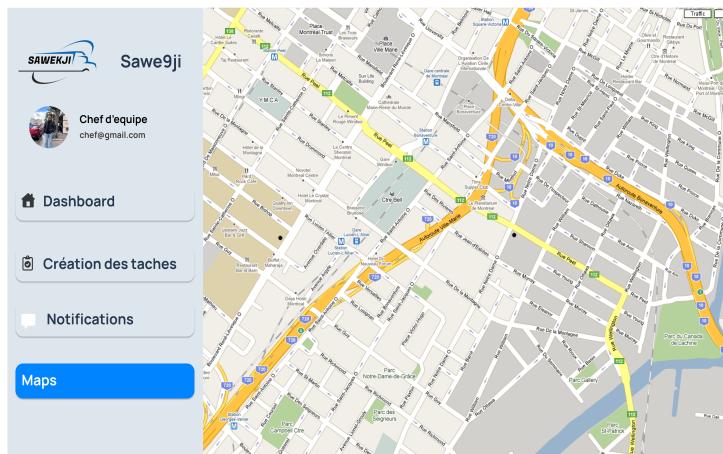


FIGURE 2.11 – map- Web



FIGURE 2.12 – map - Mobile

### 2.3.2 Charte Graphique

La phase de création de la charte graphique a été précédée par une séance de brainstorming afin de choisir la palette adéquate ainsi que le nom et le logo de l'application.



FIGURE 2.13 – la palette de couleurs



FIGURE 2.14 – logo

### 2.3.3 Plan de l'application

Dans cette section nous exposerons, dans les figures 2.5, 2.6, 2.7 et 2.8 l'organigramme des interfaces de l'application de nos profils d'acteurs à savoir : L'administrateur de l'application, le chauffeur, le mécanicien et le chef d'équipe .

#### Diagramme de flux utilisateur

Le flux utilisateur est une représentation visuelle du chemin qu'un utilisateur suit pour atteindre l'objectif d'application . Voici les éléments d'un flux utilisateur :

- **Flèches d'option de transition** : Ces flèches représentent les différentes options de choix que l'utilisateur peut faire pour avancer ou reculer dans le flux utilisateur.
- **Page** : Cela représente une seule page avec laquelle un utilisateur interagit.
- **Section dans la page** : Il s'agit de zones spécifiques sur une page, telles qu'un formulaire ou un menu de navigation.
- **Point final** : C'est le résultat final du flux utilisateur.
- **Entrée d'informations** : Ce sont les entrées requises de l'utilisateur, telles que remplir un formulaire ou saisir des informations de connexion.
- **Décision** : Ils représentent un choix ou une décision que l'utilisateur doit prendre.

- **Traitement des données** : Cela représente tout traitement de données qui se produit pendant le flux utilisateur.

**Flux utilisateur ( Définition des éléments ) :**

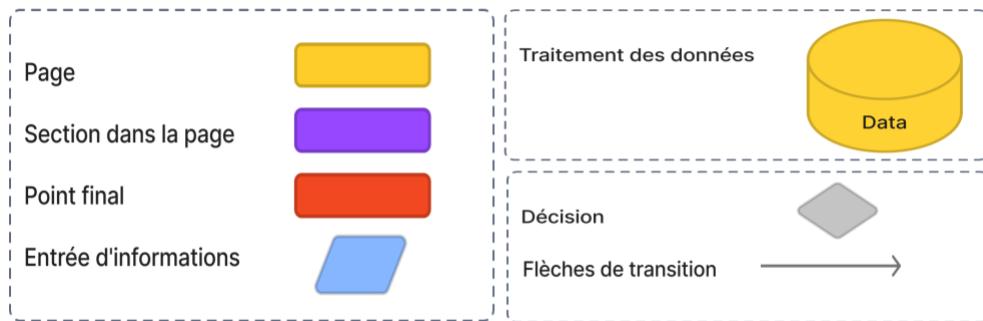


FIGURE 2.15 – Flux utilisateur : Définition des éléments

**Vue globale sur les flux d'utilisateurs :**

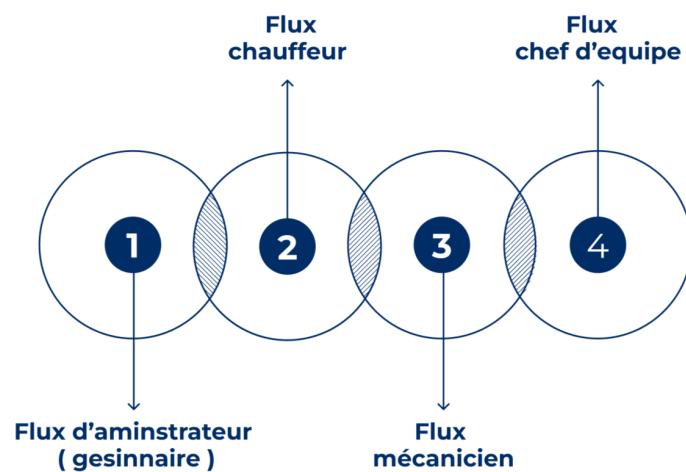


FIGURE 2.16 – Vue globale sur les flux d'utilisateurs

**Flux administrateur**

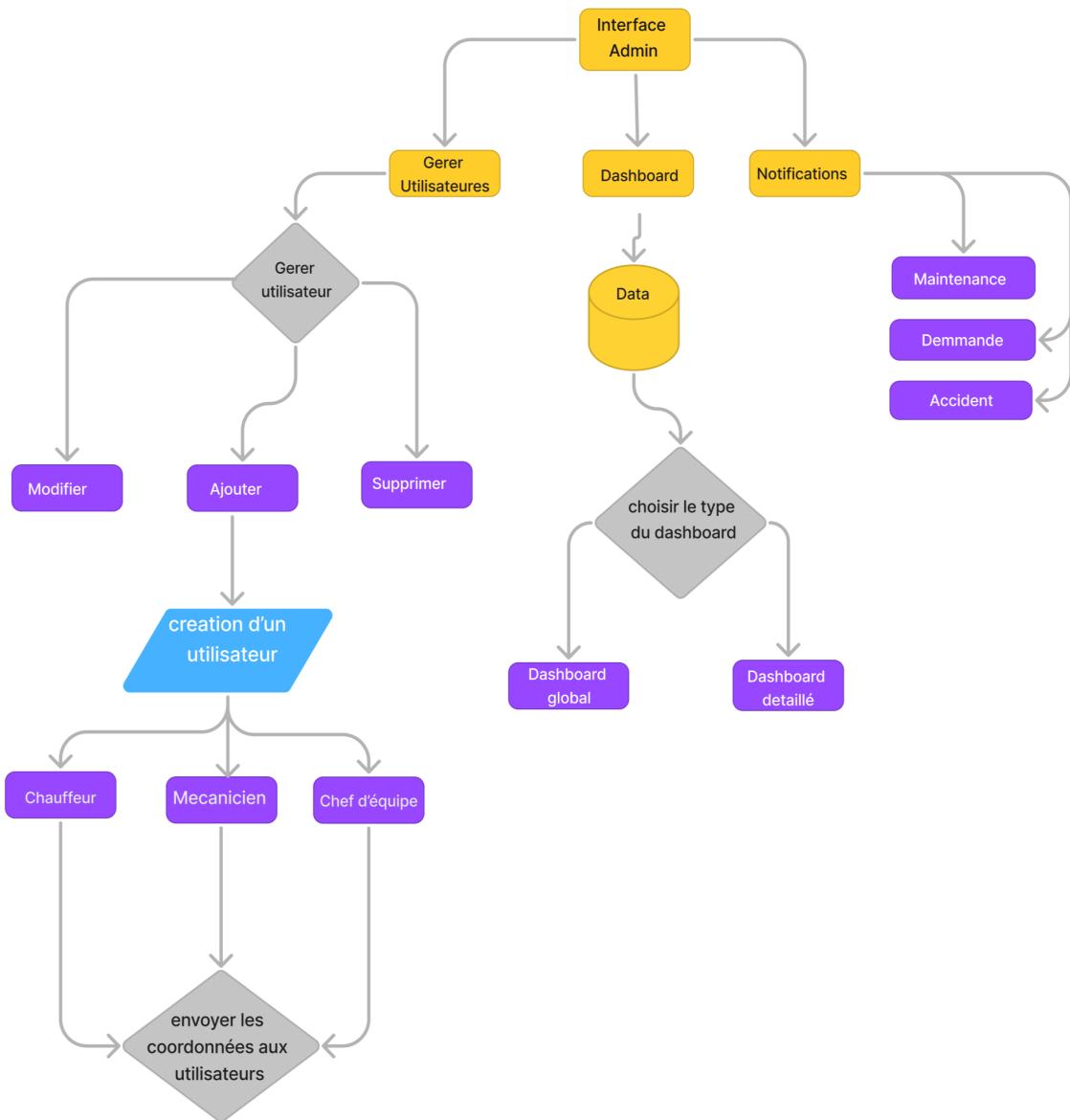


FIGURE 2.17 – Organigramme des interfaces du l'administrateur

**Flux chauffeur**

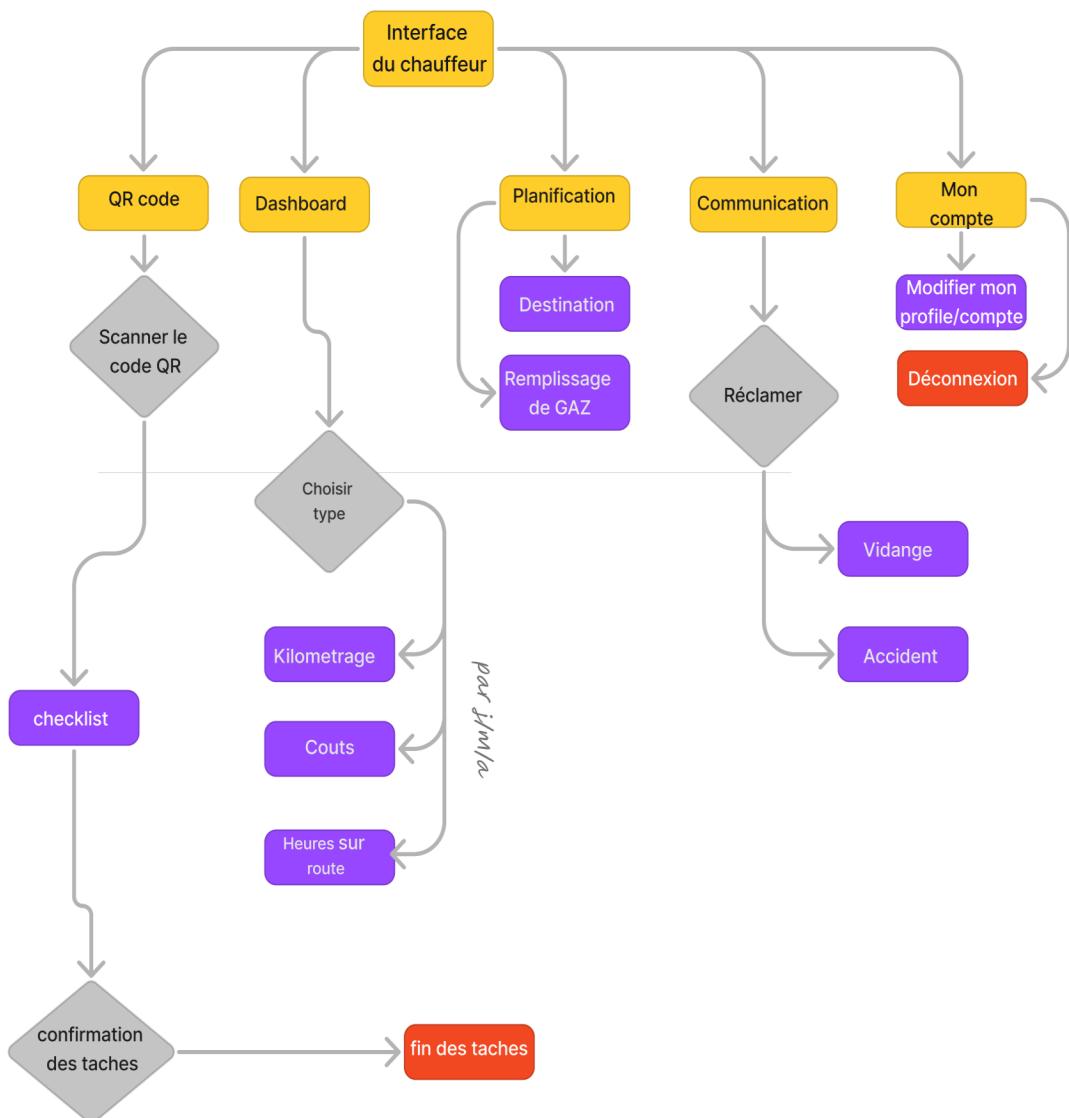


FIGURE 2.18 – Organigramme des interfaces du chauffeur

**Flux mécanicien**

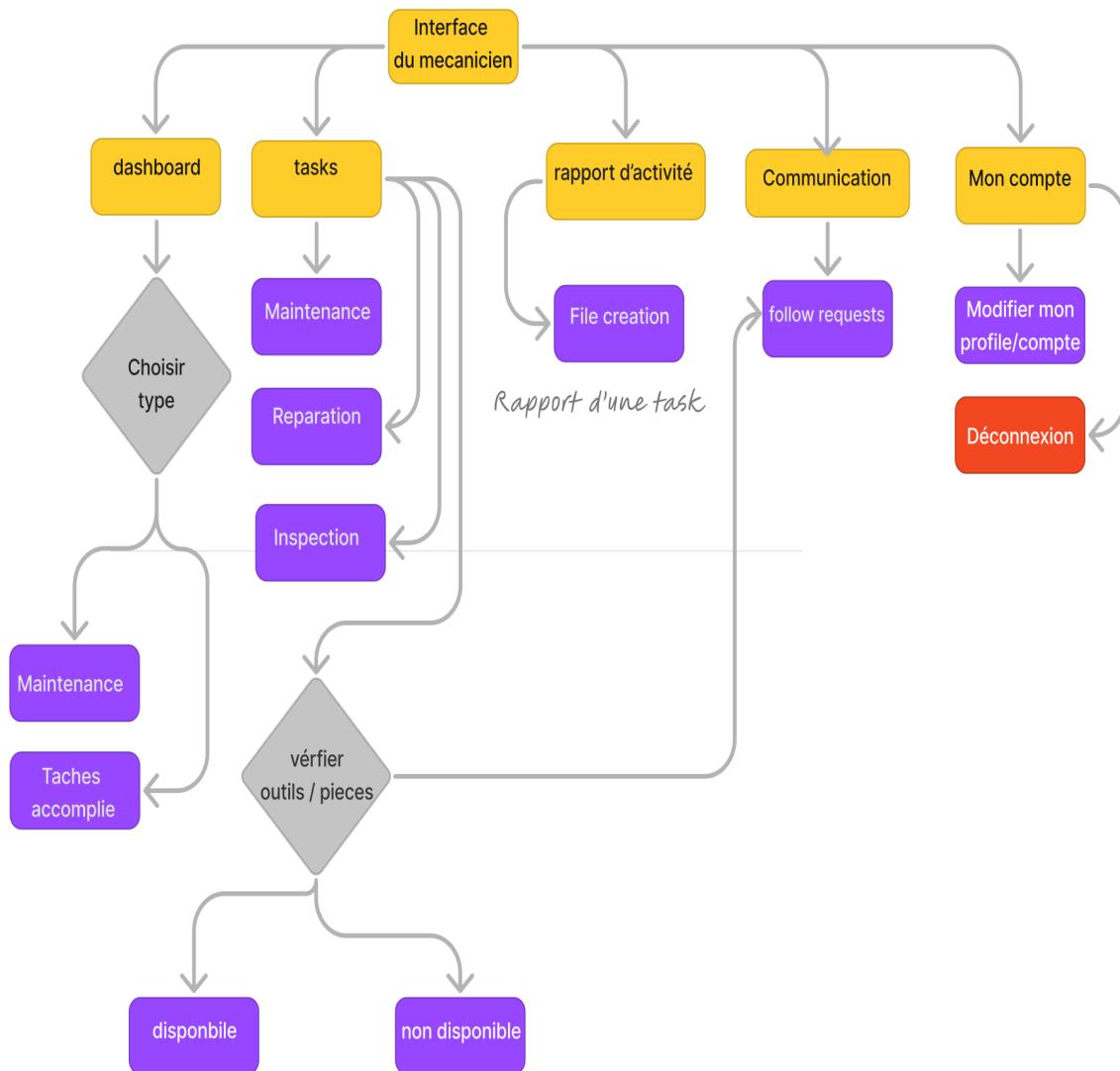


FIGURE 2.19 – Organigramme des interfaces du mécanicien

**Flux chef d'équipe**

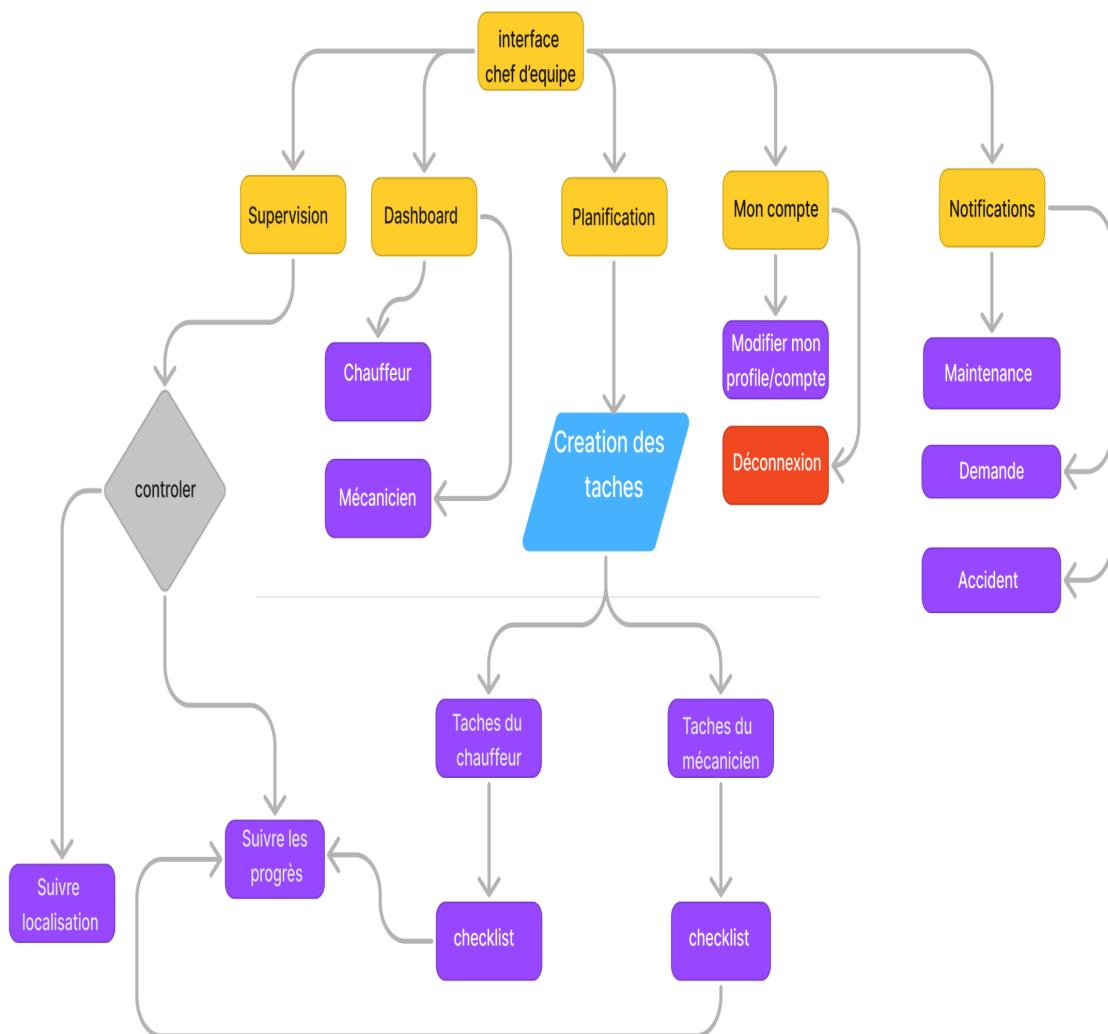


FIGURE 2.20 – Organigramme des interfaces du chef d'équipe

### **2.3.4 Diagramme de classe du domaine**

La figure 2.21 montre le diagramme de classes du domaine de notre application.

## CHAPITRE 2. PLANIFICATION DU PROJET

---

### 2.3.5 Diagramme de Gantt

Par cette figure 2.20 nous présentons une planification prévisionnelle du projet .

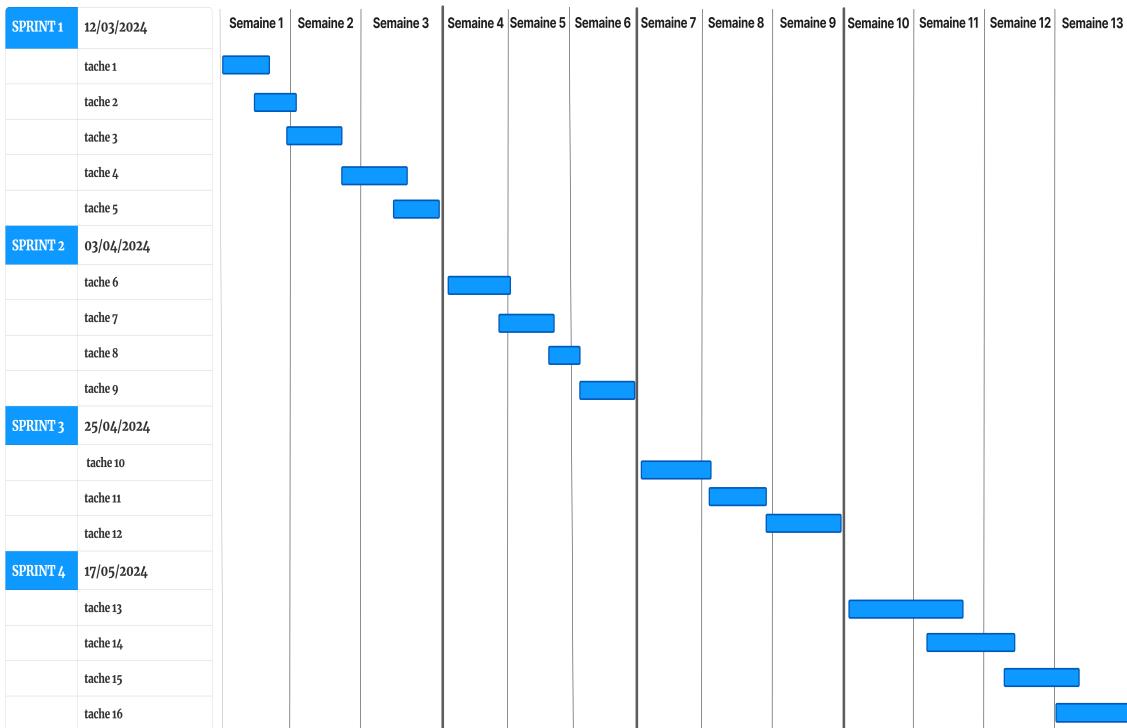


FIGURE 2.21 – Diagramme de Gantt

## Conclusion

Durant ce chapitre, en plus de l'identification des acteurs, des besoins fonctionnels et non fonctionnels de notre application, nous avons exposé le premier artéfact de la méthodologie Scrum à savoir le Backlog du produit qui comporte la liste des exigences fonctionnelles et techniques de notre application. Nous avons également identifié les différents rôles au sein de l'équipe du projet et enfin nous avons exploré le flux utilisateurs de notre application, mettant en évidence les différentes interactions entre les utilisateurs et les fonctionnalités clés de l'application. Dans le chapitre suivant, nous allons commencer notre premier sprint.

# Chapitre 3

## Sprint 1 : Gestion des accès

### Introduction

Dans ce sprint, l'équipe s'est concentrée sur la mise en place des fonctionnalités de base, notamment l'authentification des utilisateurs, la déconnexion sécurisée, la gestion des utilisateurs et la définition des rôles. Ces éléments sont essentiels pour assurer la sécurité et la convivialité de l'application, ainsi que pour permettre une gestion efficace des utilisateurs et de leurs autorisations.

### 3.1 Sprint Backlog

Le backlog du sprint présente un ensemble des fonctionnalités extraites par l'équipe scrum à partir du backlog de produit et qui vont être réalisés pendant ce sprint. Les tâches à achever pendant notre premier sprint sont présentées dans le tableau 3.1 suivant :

ID	Fonctionnalités	User Story	Story points
<b>US1</b>	Authentification	En tant qu'admin, chauffeur, chef d'équipe et mécanicien je veux m'authentifier pour que j'accède à mon espace.	2
<b>US2</b>	Déconnexion	En tant qu'admin, chauffeur, chef d'équipe et mécanicien je veux me déconnecter pour que je quitte l'application.	1
<b>US3</b>	Modification profil	En tant qu'admin, chauffeur, chef d'équipe et mécanicien je veux modifier mon profil pour modifier photo et mots de passe.	3

<b>US4</b>	Gestion des utilisateurs	En tant qu'admin je veux Gérer les utilisateurs pour que je puisse les ajouter, modifier et supprimer .	5
<b>US5</b>	Gestion des rôles	En tant qu'admin, je veux pouvoir gérer les rôles et les autorisations des utilisateurs, afin de contrôler l'accès aux fonctionnalités de l'application.	2

TABLE 3.1 – Backlog du sprint 1

## 3.2 Analyse et conception

La section "Analyse et Conception" présente les méthodes de conception et la structure de notre application. Elle présente les diagrammes de cas d'utilisation, qui identifient les interactions des utilisateurs avec le système, les diagrammes de séquences système, qui illustrent l'ordre des opérations et la coordination entre les composants du système, et les diagrammes de classes participantes, qui dépeignent la structure objet et les relations entre les classes. Ces diagrammes sont essentiels pour comprendre le fonctionnement interne et l'architecture de l'application, afin de répondre aux besoins des utilisateurs

### 3.2.1 Diagramme de cas d'utilisation du sprint 1

Ce diagramme illustre les différentes interactions que les utilisateurs peuvent avoir avec notre application. Il sert à identifier clairement les fonctionnalités accessibles par chaque type d'utilisateur et à définir les exigences système nécessaires pour ces interactions.

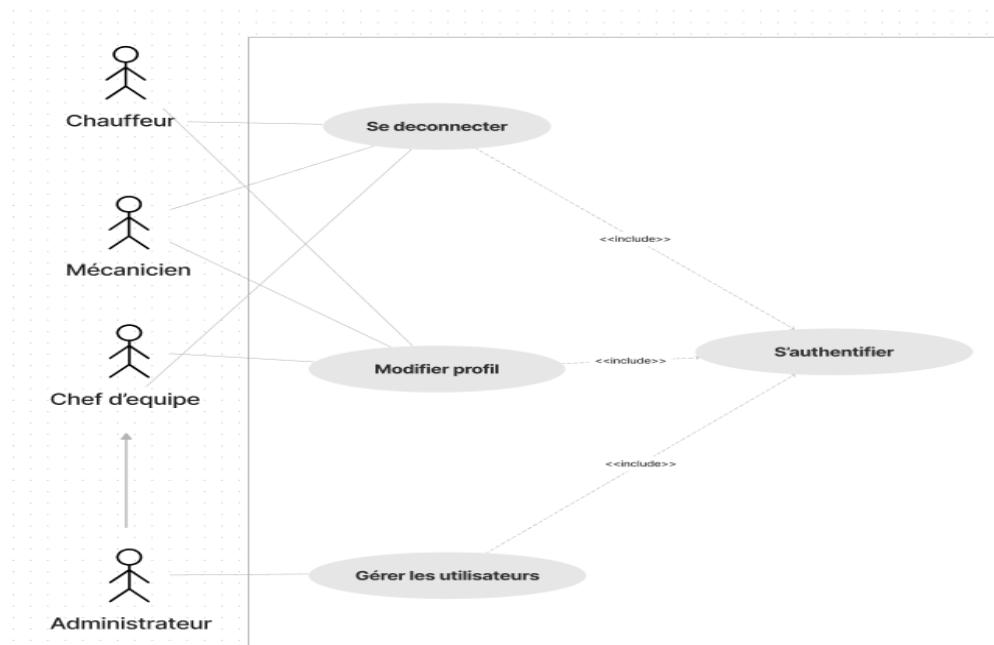


FIGURE 3.1 – Diagramme de cas d'utilisation global du sprint 1

### 3.2.2 Diagramme de séquence système du cas d'utilisation «S'authentifier»

Un diagramme de séquences système est une représentation graphique qui illustre les interactions entre les acteurs ou les composants d'un système informatique pour accomplir une série d'actions ou de processus. Il met en évidence la chronologie des échanges de messages entre ces entités, fournissant ainsi une vue détaillée de la façon dont le système fonctionne pour atteindre ses objectifs.

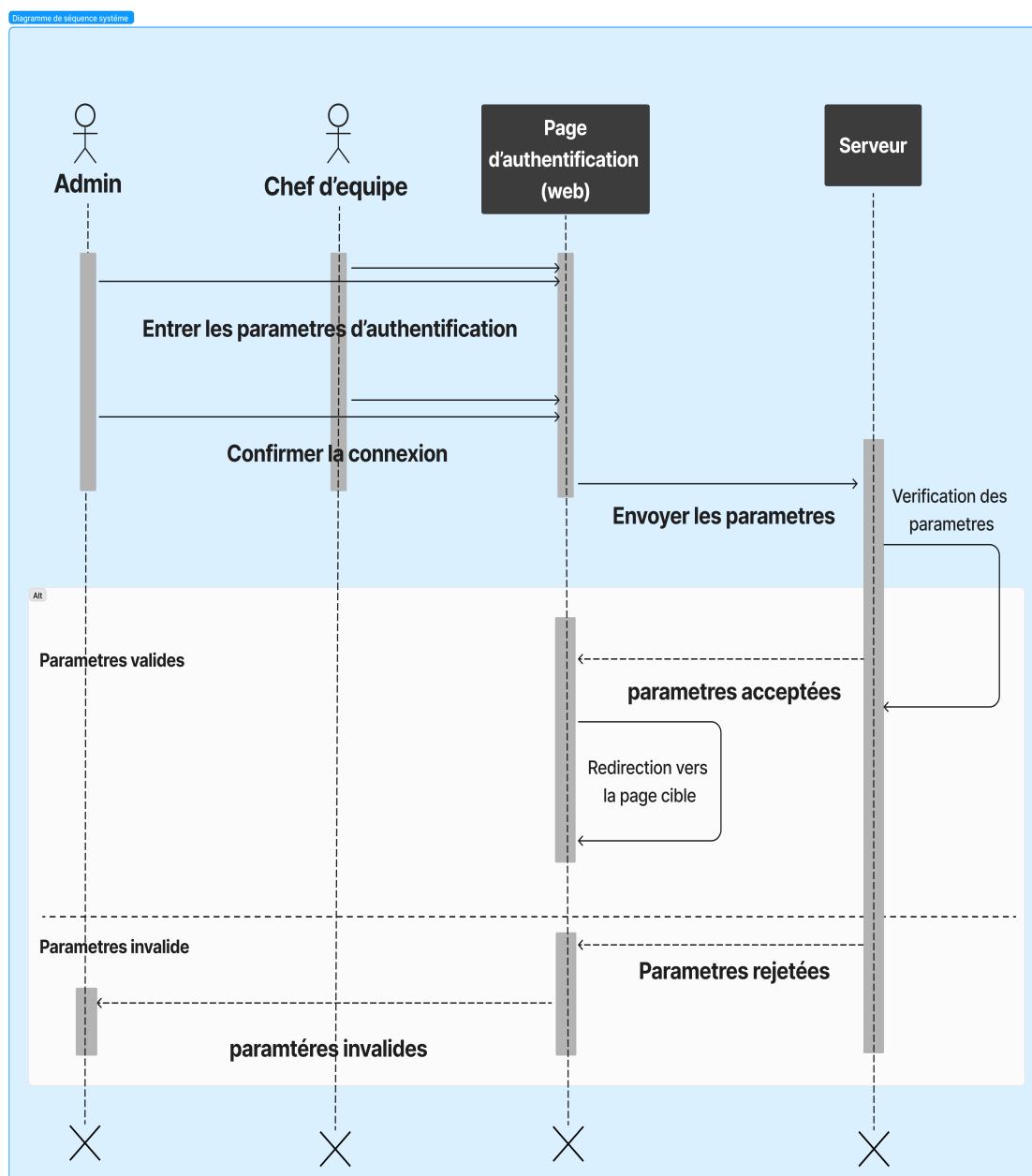


FIGURE 3.2 – Diagramme de séquence système du cas d'utilisation «S'authentifier» - Web

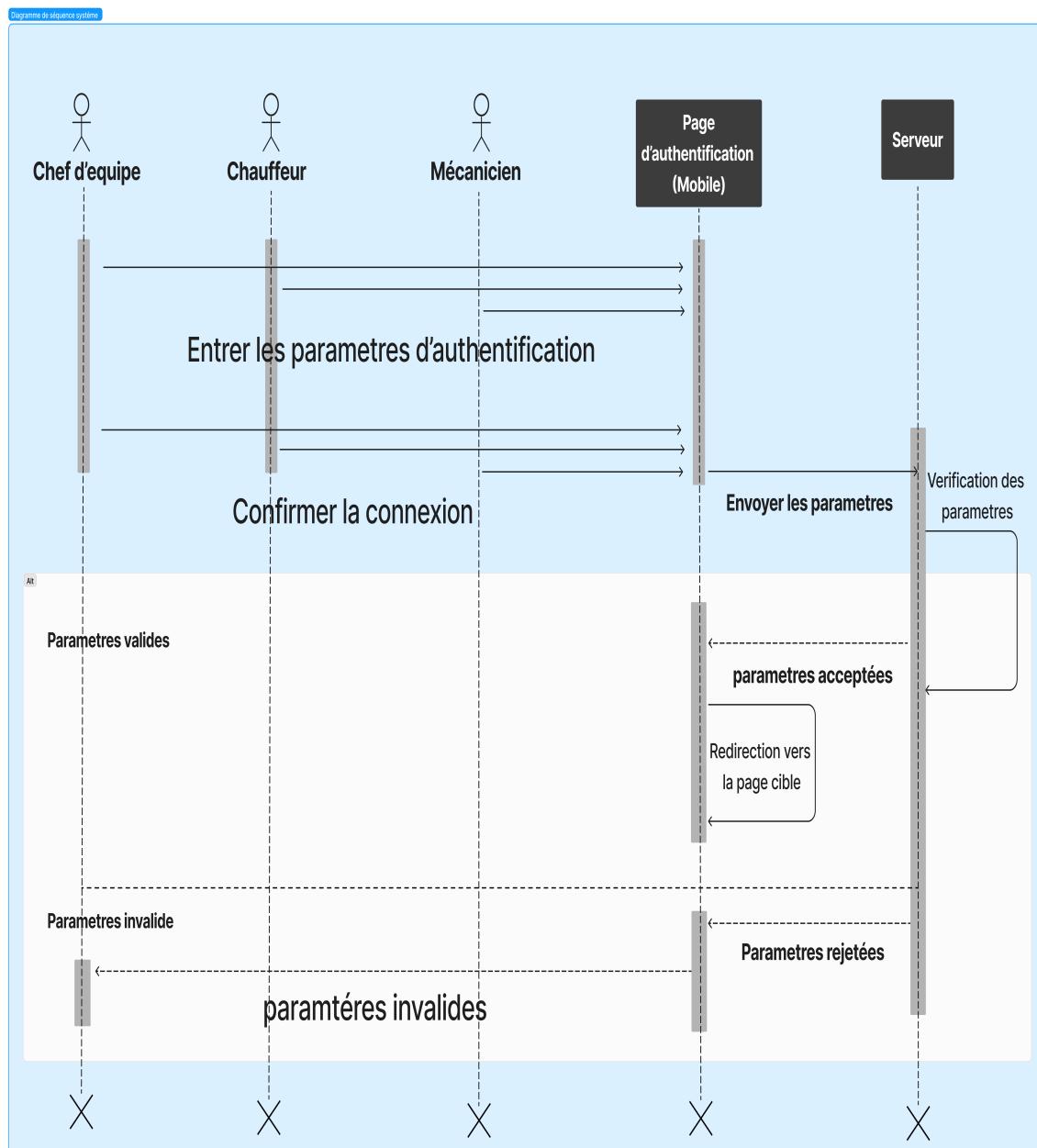


FIGURE 3.3 – Diagramme de séquence système du cas d'utilisation «S'authentifier» - Mobile

### 3.2.3 Description textuelle du cas d'utilisation “S’authentifier”

<b>Cas d'utilisation</b>	S'authentifier
<b>Acteur</b>	Admin, Chef d'équipe, Chauffeur et Mécanicien
<b>Pré-condition</b>	L'utilisateur accède à l'interface d'authentification de l'application.
<b>Post-condition</b>	L'utilisateur est authentifié et accède à son espace sécurisé dans l'application.
<b>Scénario nominal</b>	<p>1- L'utilisateur saisit son identifiant et son mot de passe.</p> <p>2- L'utilisateur appuie sur le bouton de confirmation.</p> <p>3- Le système vérifie les informations d'identification fournies par l'utilisateur.</p> <p>4- L'utilisateur est redirigé vers son espace sécurisé dans l'application.</p>
<b>Scénario alternatif</b>	<p>A1 identifiants / mot de passe erroné :</p> <p>l'enchaînement A1 démarre au point 3 du scénario nominal.</p> <p>4- un message d'erreur va être afficher.</p> <p>Le scénario nominal reprend au point 1.</p>

TABLE 3.2 – Description textuelle du cas d'utilisation “S’authentifier”

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

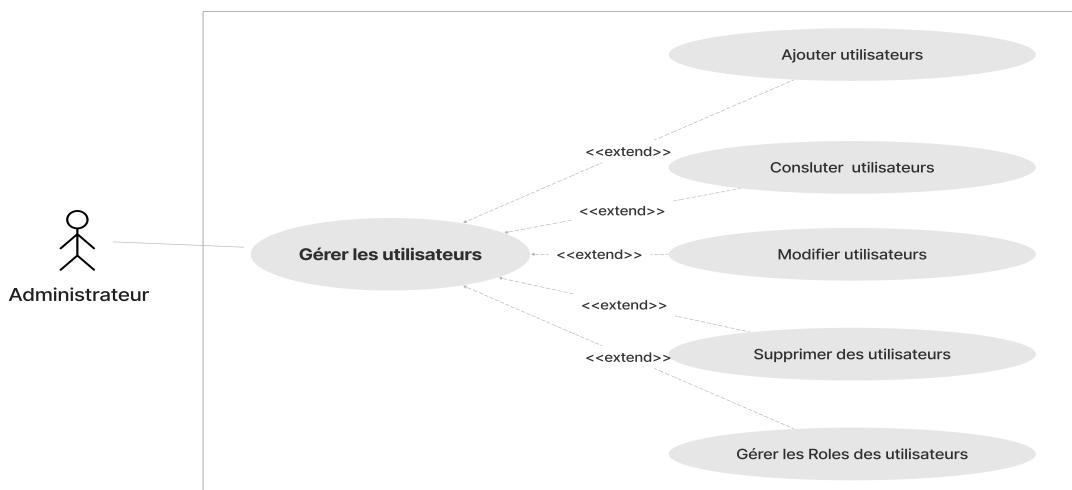


FIGURE 3.4 – Analyse de cas d'utilisation « Gérer les utilisateurs »

### 3.2.5 Diagramme de séquence système du cas d'utilisation « Ajouter utilisateur »

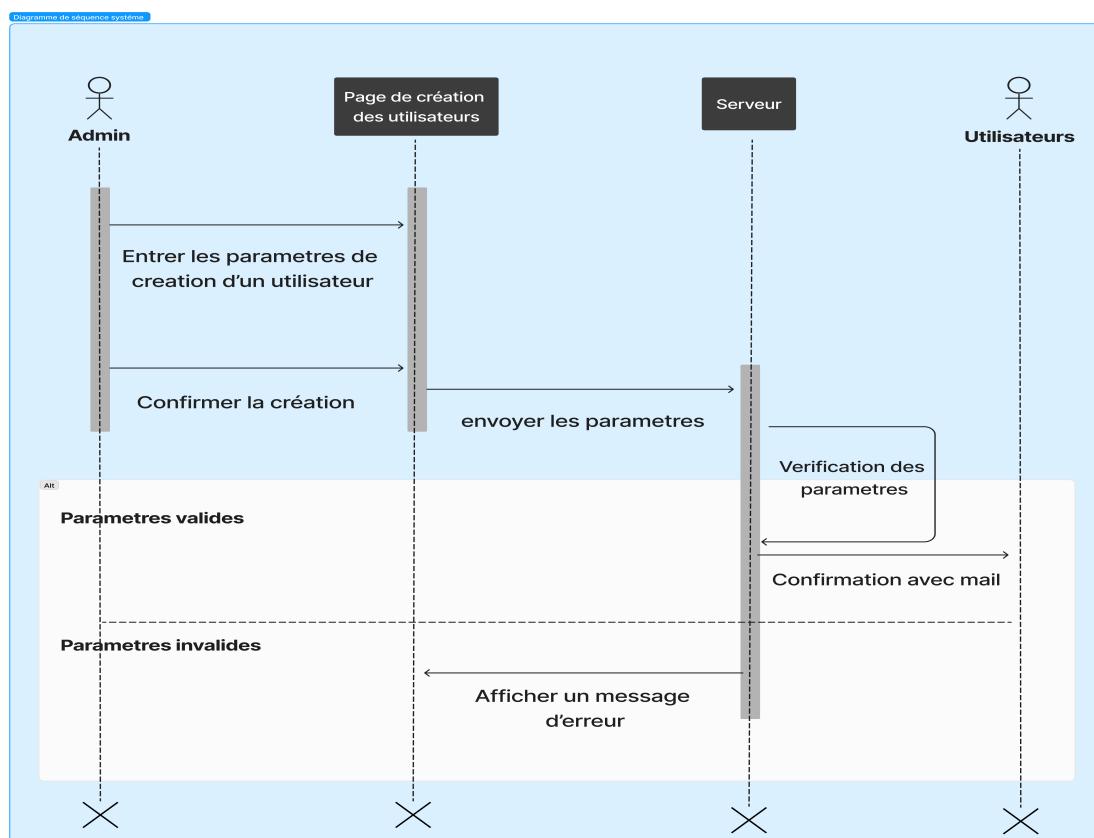


FIGURE 3.5 – Diagramme de séquence système du cas d'utilisation « Ajouter utilisateur »

**3.2.6 Description textuelle du cas d'utilisation “Ajouter utilisateur”**

<b>Cas d'utilisation</b>	Ajouter Utilisateur
<b>Acteur</b>	Administrateur
<b>Pré-condition</b>	L'administrateur est authentifié dans le système et a accès à l'interface de gestion des utilisateurs.
<b>Post-condition</b>	Le nouvel utilisateur est ajouté avec succès dans le système et peut accéder aux fonctionnalités appropriées.
<b>Scénario nominal</b>	<p>1- L'administrateur accède à l'interface de gestion des utilisateurs.</p> <p>2- L'administrateur ajoute un nouvel utilisateur en saisissant ses informations dans un formulaire dédié.</p> <p>3- Le nouvel utilisateur est ajouté à la base de données.</p> <p>4- Le système envoie un e-mail au nouvel utilisateur contenant ses informations de connexion pour accéder au système.</p>
<b>Scénario alternatif</b>	<p>A1 Les informations sont incorrectes :</p> <p>l'échancement A1 démarre au point 3 du scénario nominal.</p> <p>4- Le système affiche un message pour vérifier les informations saisies.</p> <p>Le scénario nominal reprend au point 2.</p>

TABLE 3.3 – Description textuelle du cas d'utilisation “Ajouter utilisateur”

### **3.3 Diagramme de classes du sprint 1**

### 3.4 Réalisation

Dans cette section, nous mettons en lumière les interfaces de l'application que nous avons développée. À travers des captures d'écran détaillées, nous illustrons les fonctionnalités clés mises à disposition des utilisateurs, démontrant ainsi l'application pratique des concepts et des designs discutés dans les sections précédentes.

#### 3.4.1 Authentification

Cette interface permet aux utilisateurs de se connecter à l'application. L'utilisateur est accueilli par un formulaire de connexion simple, comprenant des champs pour saisir son identifiant et son mot de passe. Une fois les informations d'identification saisies, l'utilisateur peut appuyer sur le bouton 'confirmer' pour accéder à son espace sécurisé, si les informations d'identification saisies sont incorrectes, le système affiche un message d'erreur.

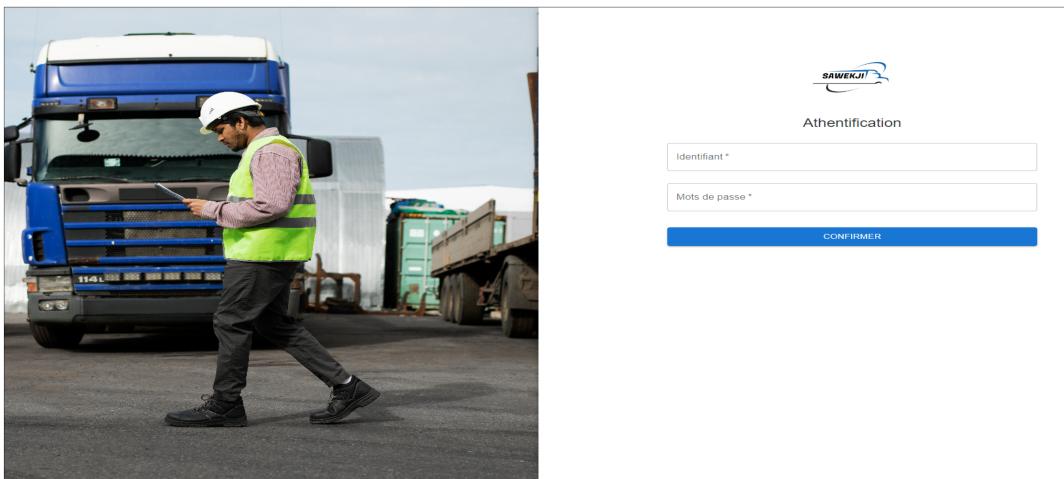


FIGURE 3.6 – Interface d'authentification Web

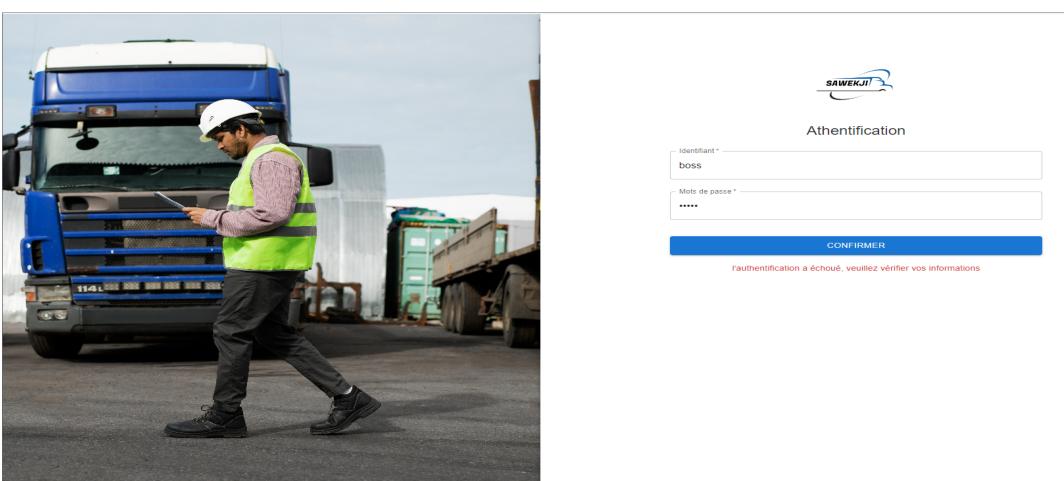


FIGURE 3.7 – Echec de connexion

## CHAPITRE 3. SPRINT 1 : GESTION DES ACCÈS

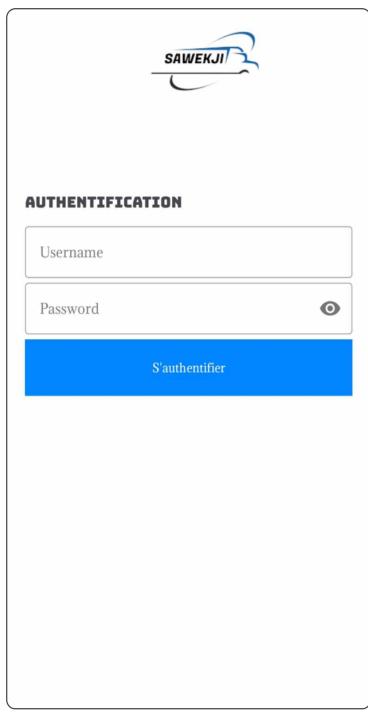


FIGURE 3.8 – Interface d’authentification Mobile

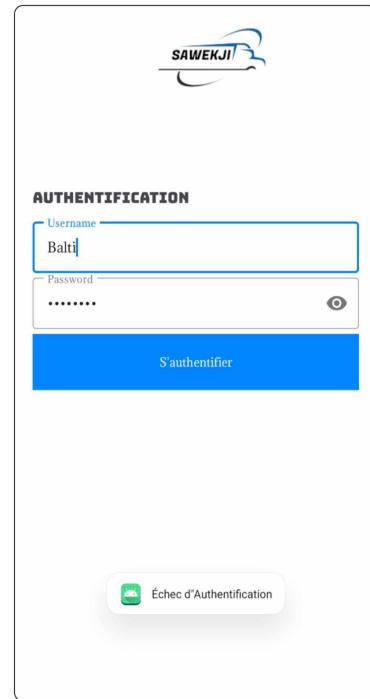


FIGURE 3.9 – Echec de connexion

### 3.4.2 Profil

Permet à l’administrateur, chef d’équipe, chauffeur et mécanicien de changer leur photo de profil et leur mot de passe.

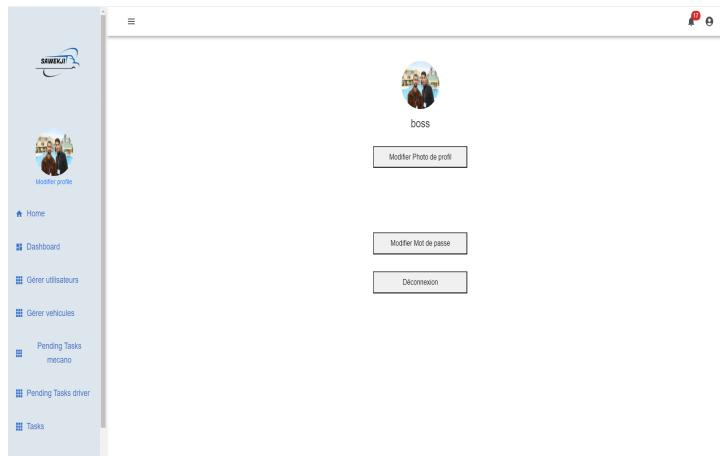


FIGURE 3.10 – Interface de profil - Web

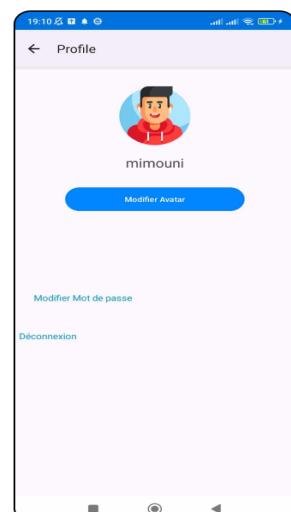


FIGURE 3.11 – Interface de profil - Mobile

### 3.4.3 Gestion des utilisateurs

Lorsque l'administrateur accède à cette interface, il est présenté avec un tableau répertoriant tous les utilisateurs de l'application. Chaque ligne du tableau représente un utilisateur et affiche des informations telles que le nom, l'adresse e-mail et le rôle. L'administrateur a la possibilité de supprimer un utilisateur en cliquant sur l'icône de suppression correspondante dans la ligne de l'utilisateur. De plus, en cliquant sur l'icône de modification, l'administrateur peut mettre à jour les informations d'un utilisateur existant.

ID	Username	Email	Role	Action
2	balti		mecano	
4	boss		admin	
33	cheritou	nourcherni357@gmail.com	chef	
34	mamam	mamamama@gmail.com	Driver	
36	mimouni	medazizmimouni35@gmail.com	driver	
55	kaka		chef	
57	hanen	hanen.chamathi@yahoo.fr	Chef	
58	a		mecano	
61	neji	neji@gmail.com	chef	
62		boss		
63	kada	hamzaakadaa1@gmail.com	chef	

FIGURE 3.12 – Interface de gestion des utilisateurs

### 3.4.4 Création d'un nouvel utilisateur

Lorsque l'administrateur clique sur le bouton "Ajouter Utilisateur" dans l'interface précédente, un formulaire d'ajout d'utilisateur s'affiche à l'écran. Ce formulaire permet à l'administrateur de saisir les informations nécessaires pour créer un nouvel utilisateur, telles que le nom, l'adresse e-mail et le mot de passe initial. L'administrateur peut également sélectionner le rôle de l'utilisateur parmi les options disponibles. Une fois que toutes les informations requises sont fournies, l'administrateur peut soumettre le formulaire pour créer un nouvel utilisateur dans le système.

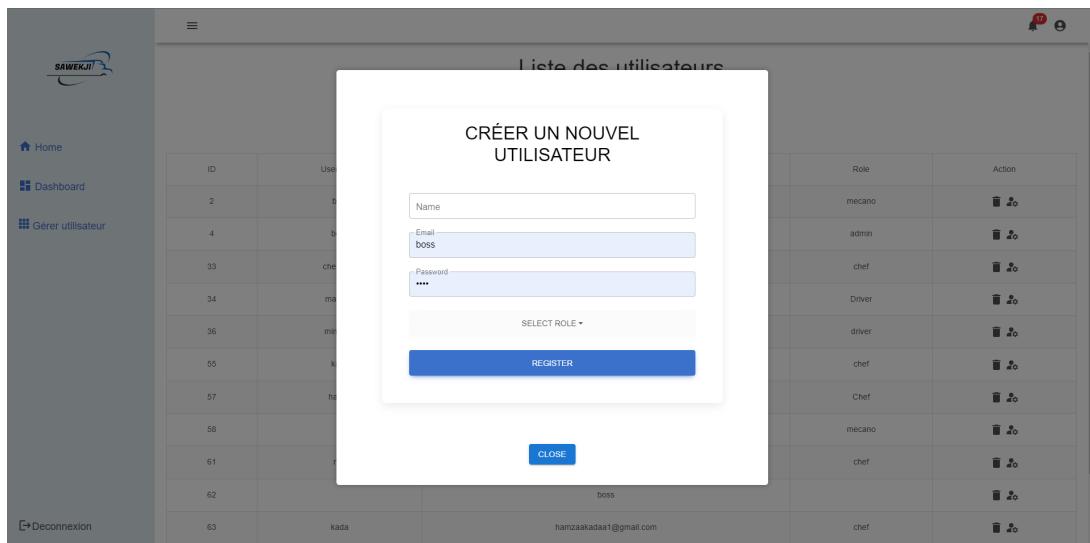


FIGURE 3.13 – Interface de création d'un nouvel utilisateur

## 3.5 Rétrospective

Dans cette section de rétrospective, nous examinerons de près les performances de notre équipe pendant le Sprint 1 en utilisant deux outils essentiels de la méthodologie Scrum : le burndown chart et le task board. Ces outils nous permettent de visualiser et d'évaluer notre progression tout au long du sprint.

## CHAPITRE 3. SPRINT 1 : GESTION DES ACCÈS

### 3.5.1 Burdown Chart

Le burndown chart est un outil essentiel utilisé dans la gestion de projet agile pour suivre la progression des tâches. Il illustre graphiquement la quantité de travail restant à accomplir par l'équipe au cours du sprint. Dans notre cas, nous avons planifié un sprint de 3 semaines, avec une moyenne de 5 heures de travail par jour.

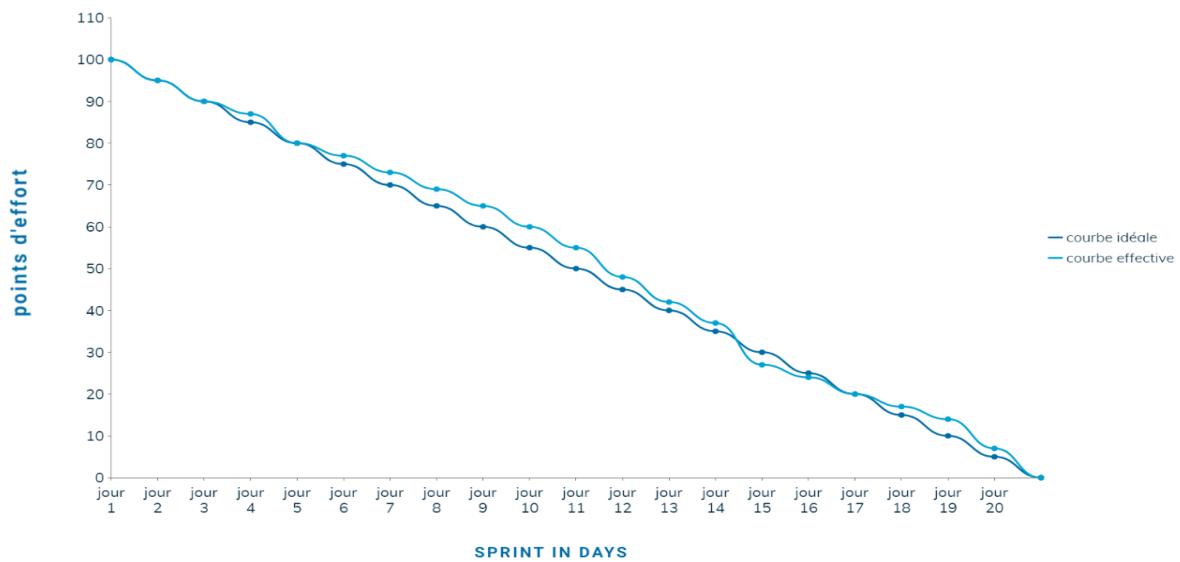


FIGURE 3.14 – Burdown Chart du Sprint 1

### 3.5.2 Task Bord

La figure 3.14 présente le "Task Bord" correspondant au 6ème jour du sprint 1.

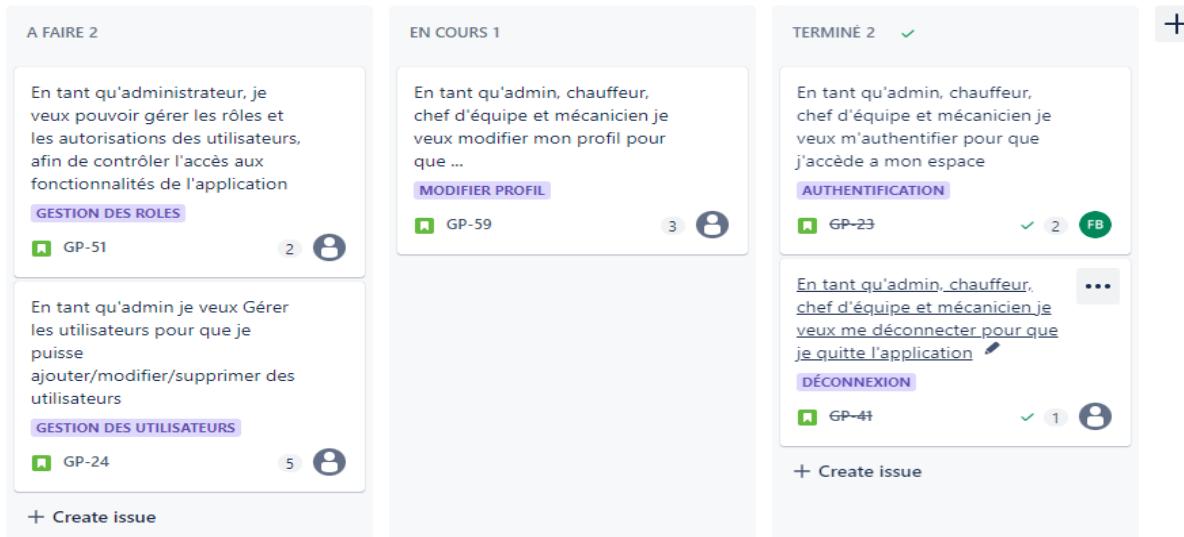


FIGURE 3.15 – Task board du Sprint 1

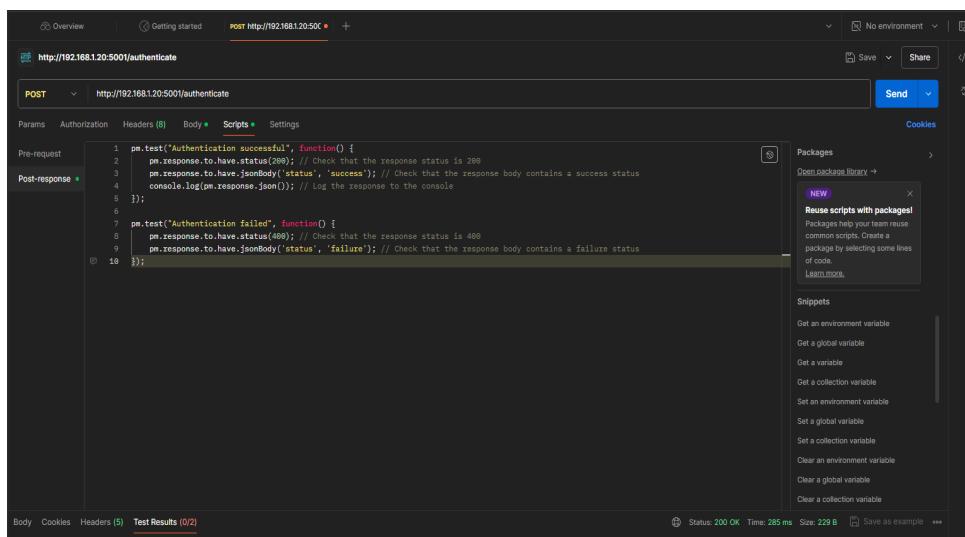
## 3.6 Tests unitaires

Dans cette section, nous utiliserons l'outil Postman pour effectuer des tests unitaires sur notre application. Les tests unitaires consistent à vérifier le bon fonctionnement de chaque unité de code de manière isolée. Nous définirons des scénarios de test, configurerons des requêtes HTTP dans Postman, et écrirons des scripts pour valider les résultats renvoyés par chaque unité de code. Ces tests nous permettront de détecter les erreurs dès le stade de développement .

### Test unitaires du cas d'utilisation « S'authentifier » :

Nous présentons ci-dessous le code source et le résultat de deux cas de test :

1. Toutes les données sont valides.
2. Mots de passe incorrecte .

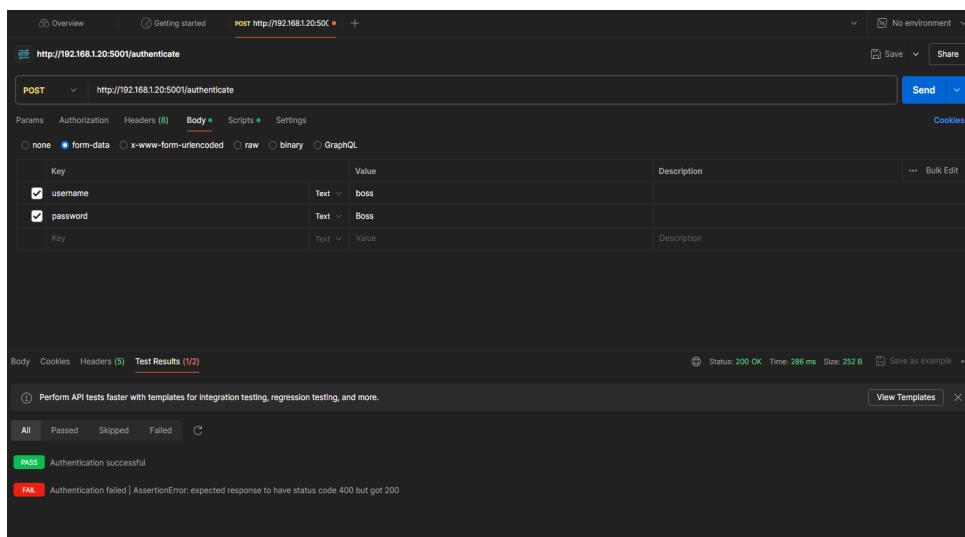


```

POST http://192.168.1.20:5001/authenticate
pm.test("Authentication successful", function() {
    pm.response.to.have.status(200); // Check that the response status is 200
    pm.response.to.have.jsonbody("status", "success"); // Check that the response body contains a success status
    console.log(pm.response.json()); // Log the response to the console
});
pm.test("Authentication failed", function() {
    pm.response.to.have.status(400); // Check that the response status is 400
    pm.response.to.have.jsonBody("status", "failure"); // Check that the response body contains a failure status
});

```

FIGURE 3.16 – Code source du test unitaire du cas d'utilisation «S'authentifier »



Key	Value	Description
username	boss	
password	Boss	

**Test Results (1/2)**

All Passed Skipped Failed

**PASS** Authentication successful

**FAIL** Authentication failed | Assertion error: expected response to have status code 400 but got 200

FIGURE 3.17 – Résultat du premier test

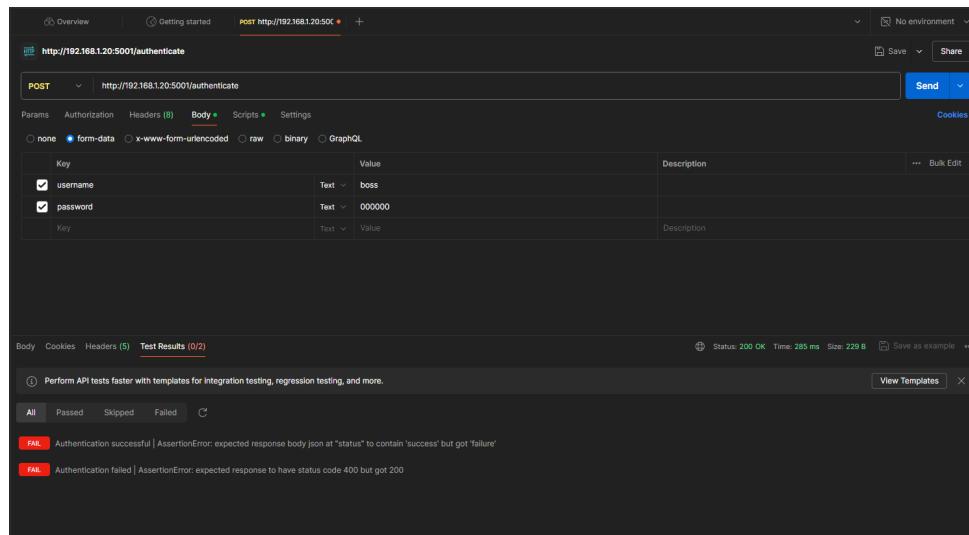


FIGURE 3.18 – Resultat du deuxième test

## Conclusion

Dans ce chapitre, nous avons examiné en détail le premier Sprint, en décrivant le contenu du Sprint Backlog, le diagramme de cas d'utilisation et les séquences système des cas d'utilisation principaux. Nous avons également illustré quelques fonctionnalités de l'application à travers un scénario, et présenté d'autres outils de Scrum comme le Task Board et le burndown Chart. Dans le prochain chapitre, nous continuerons à explorer les fonctionnalités du deuxième Sprint.

## Chapitre 4

# Sprint 2 : Gestion des Tâches

### Introduction

Dans ce sprint, nous nous concentrons sur le développement des fonctionnalités de création et de consultation des tâches dans notre application. Nous avons élaboré un backlog, des cas d'utilisation, des diagrammes de séquence et des descriptions textuelles pour guider notre travail. Nous avons également conçu un diagramme de classe participant pour planifier l'implémentation des fonctionnalités, ainsi que des captures d'écran des interfaces utilisateur. Nous examinerons également le burndown chart et le task board pour suivre l'avancement du sprint et discuterons de la mise en œuvre des tests unitaires pour garantir la qualité du code..

#### 4.1 Sprint Backlog

ID	Fonctionnalités	User Story	Story points
<b>US6</b>	Gestion des véhicules	En tant qu'administrateur, je veux ajouter, supprimer et visualiser les véhicules pour que la gestion du parc soit efficace et à jour.	5
<b>US7</b>	Créations des taches	En tant que Chef d'équipe je veux Créer des taches pour les chauffeurs et les mécanicien.	5
<b>US8</b>	Consultation des taches	En tant que Chauffeur je veux consulter la checklist pour que je puisse vérifier tous les éléments essentiels et confirmer l'exécution des tâches .	3
<b>US9</b>	Consultation des taches	En tant que Mécanicien je veux Consulter les détails des interventions pour que je puisse obtenir toutes les informations nécessaires sur les tâches assignées.	3

TABLE 4.1 – Backlog du sprint 2

## 4.2 Analyse et conception

Dans cette section, nous détaillerons le processus d'analyse et de conception pour les fonctionnalités de création et de consultation des tâches.

### 4.2.1 Diagramme de cas d'utilisation du sprint 2

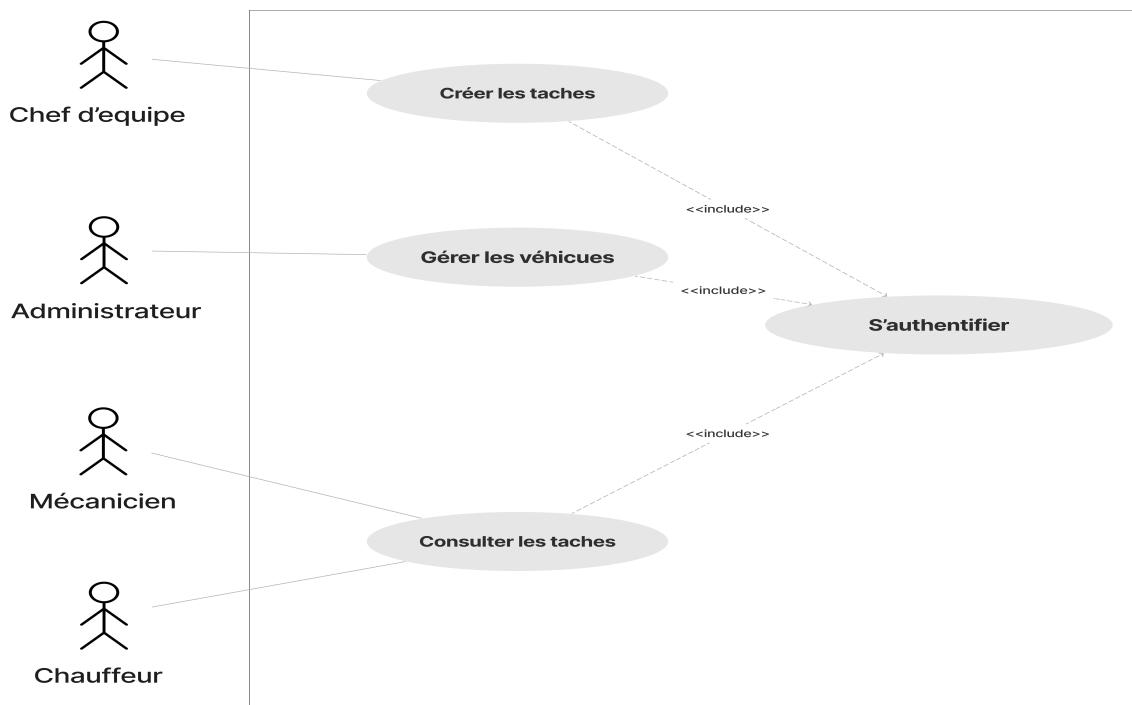


FIGURE 4.1 – Diagramme de cas d'utilisation global du sprint 2

### 4.2.2 Raffinement du cas d'utilisation « Gérer les véhicules »

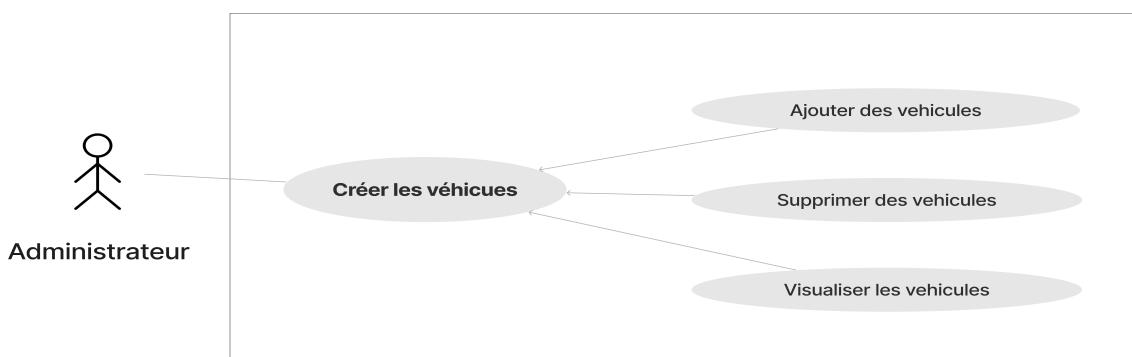


FIGURE 4.2 – Raffinement de cas d'utilisation « Gérer les véhicules »

**4.2.3 Description textuelle du cas d'utilisation « Ajouter véhicules »**

<b>Cas d'utilisation</b>	Ajouter des Véhicules
<b>Acteur</b>	Administrateur
<b>Pré-condition</b>	1- L'administrateur doit être connecté et a les droits nécessaires pour ajouter des véhicules.
<b>Post-condition</b>	Le véhicule est ajouté à la base de données.
<b>Scénario nominal</b>	<p>1- L'administrateur accède à la section "Gestion des Véhicules" dans le système.</p> <p>2- L'administrateur remplit les champs requis pour le nouveau véhicule.</p> <p>3- L'administrateur clique sur "Confirmer" pour envoyer les informations du nouveau véhicule.</p> <p>4- Le système vérifie que la plaque d'immatriculation n'est pas déjà utilisée.</p> <p>5- Le système ajoute le véhicule à la base de données.</p> <p>6- Le système affiche un message de confirmation indiquant que le véhicule a été ajouté avec succès.</p>
<b>Scénario alternatif</b>	<p>A1 Immatriculation déjà utilisée :</p> <p>L'échainement A1 démarre au point 4 du scénario nominal.</p> <p>5- Le système affiche un message pour vérifier les informations du nouveau véhicule .</p> <p>Le scénario nominal reprend au point 2.</p> <p>A2 Ajout du véhicule échoué dans la base de données :</p> <p>L'échainement A2 démarre au point 5 du scénario nominal.</p>

	<p>6- Le système affiche un message que le véhicule n'a pas été ajouté à la base de données</p> <p>Le scénario nominal reprend au point 4.</p>
--	--

TABLE 4.2 – Description textuelle du cas d'utilisation “Ajouter véhicules”

#### 4.2.4 Diagramme de séquence système du cas d'utilisation « Ajouter des véhicules »

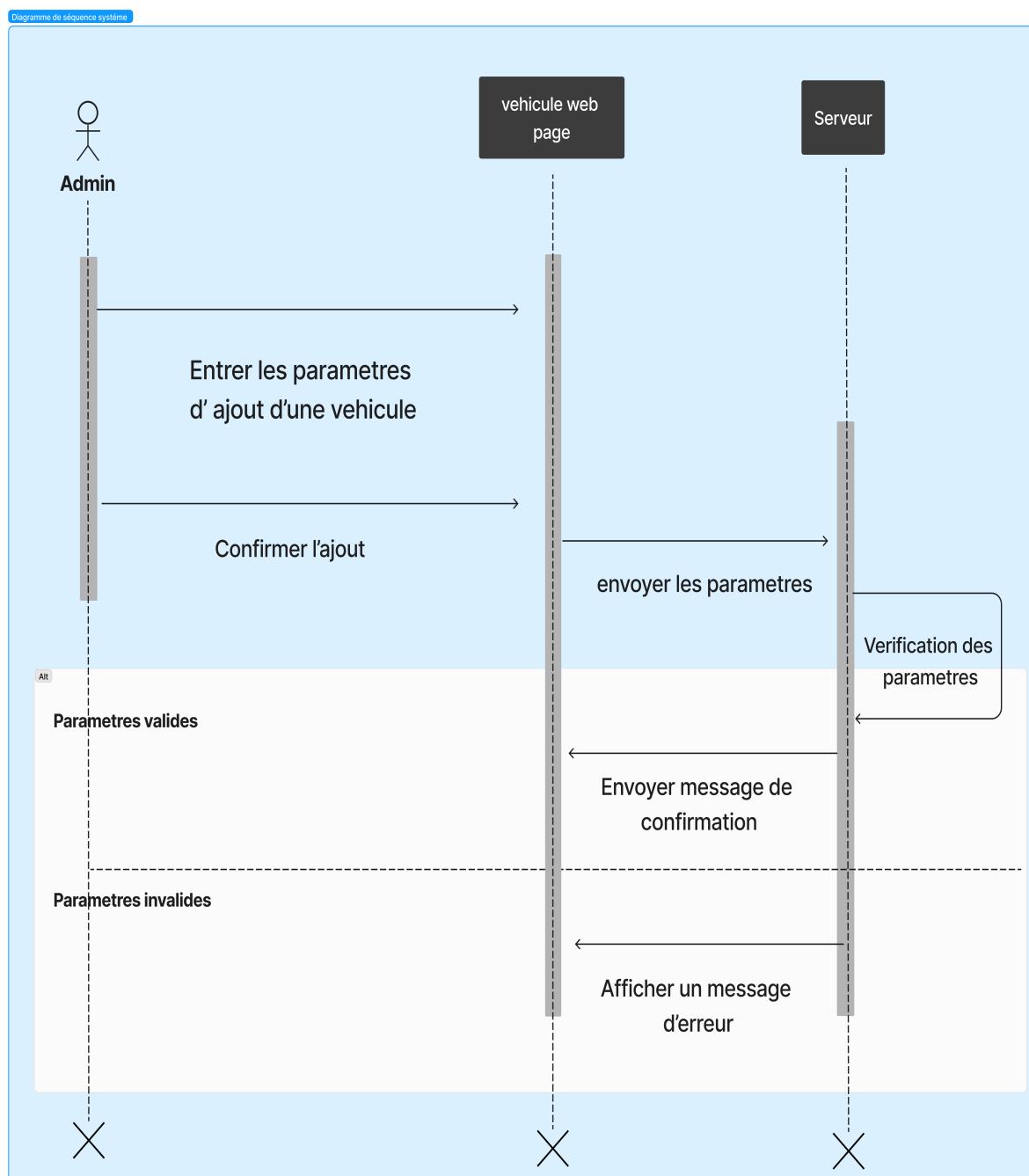


FIGURE 4.3 – Raffinement de cas d'utilisation « Ajout des véhicules »

#### 4.2.5 Raffinement du cas d'utilisation « Créer les tâches »

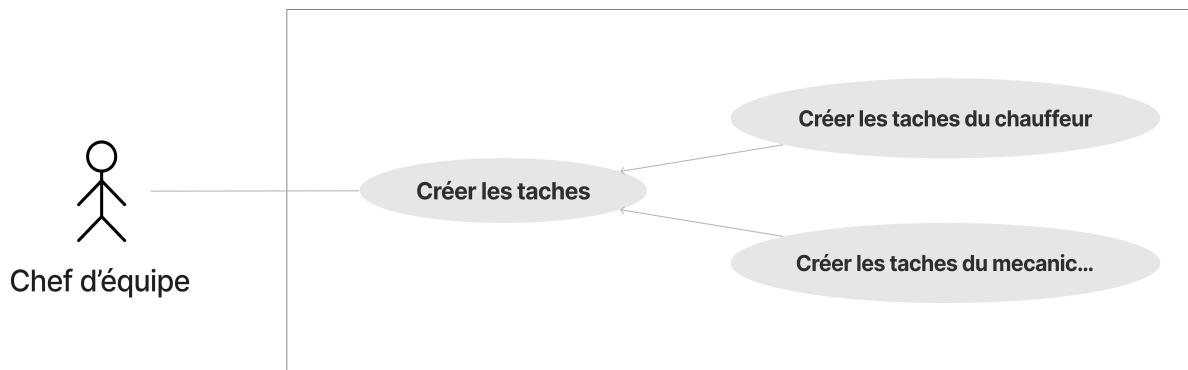


FIGURE 4.4 – Raffinement de cas d'utilisation « Créer les tâches »

#### 4.2.6 Description textuelle du cas d'utilisation « Créer des Tâches pour les Mécaniciens »

<b>Cas d'utilisation</b>	Créer des Tâches pour les Mécaniciens
<b>Acteur</b>	Chef d'Équipe
<b>Pré-condition</b>	1-Le chef d'équipe est authentifié dans le système.  2-Il existe une liste de mécaniciens disponibles dans le système.
<b>Post-condition</b>	Les tâches sont créées et assignées aux mécaniciens sélectionnés avec succès.
<b>Scénario nominal</b>	1- Le chef d'équipe accède à l'interface de création de tâches pour les mécaniciens.  2- Le chef d'équipe choisit une date pour laquelle les tâches seront planifiées.  3- Le système affiche une liste des mécaniciens disponibles pour cette date.  4- Le chef d'équipe sélectionne un mécanicien parmi la liste pour lui assigner une tâche.  5- Le chef d'équipe saisit les détails de la tâche, y compris le titre, la description.

	<p>6- Le chef d'équipe confirme la création de la tâche et l'assignation du mécanicien.</p> <p>7- Le système enregistre la tâche créée et envoie une notification au mécanicien assigné.</p>
<b>Scénario alternatif</b>	<p>.</p> <p>.</p> <p>.</p> <p>.</p> <p>.</p>

TABLE 4.3 – Description textuelle du cas d'utilisation “Créer des Tâches pour les Mécaniciens”

#### **4.2.7 Description textuelle du cas d'utilisation « Créer des Tâches pour les Chauffeurs »**

<b>Cas d'utilisation</b>	Créer des Tâches pour les Chauffeurs
<b>Acteur</b>	Chef d'Équipe
<b>Pré-condition</b>	<p>1- Le chef d'équipe est authentifié et a accès à l'interface de gestion des tâches..</p> <p>2- Les informations sur les véhicules et les chauffeurs sont disponibles dans le système..</p>
<b>Post-condition</b>	La tâche est correctement créée et assignée au chauffeur avec toutes les informations nécessaires
<b>Scenario nominal</b>	<p>1- Le chef d'équipe accède à l'interface de création des tâches pour les chauffeurs.</p> <p>2- Il sélectionne le modèle de véhicule souhaité pour la tâche.</p>

## CHAPITRE 4. SPRINT 2 : GESTION DES TÂCHES

---

	<p>3- Il choisit la date pour laquelle la tâche est à planifier.</p> <p>4- Le système affiche les véhicules disponibles qui correspondent au modèle sélectionné et sont libres à la date choisie.</p> <p>5- Le chef d'équipe sélectionne un véhicule par sa matricule, puis saisit l'ID de la tâche et une description détaillée de celle-ci.</p> <p>6- Le chef d'équipe recherche les chauffeurs qualifiés pour conduire le modèle de véhicule choisi et qui sont disponibles à la date spécifiée.</p> <p>7- Il assigne la tâche à un chauffeur sélectionné et met à jour le QR code du véhicule pour refléter les détails de la nouvelle tâche.</p> <p>8- Le système enregistre la tâche avec toutes les informations nécessaires et informe le chauffeur assigné.</p>
<b>Scénario alternatif</b>	<p>.</p> <p>.</p> <p>.</p> <p>.</p> <p>.</p>

TABLE 4.4 – Description textuelle du cas d'utilisation “Créer des Tâches pour les chauffeurs ”

#### 4.2.8 Diagramme de séquence système du cas d'utilisation «Créer les tâches»

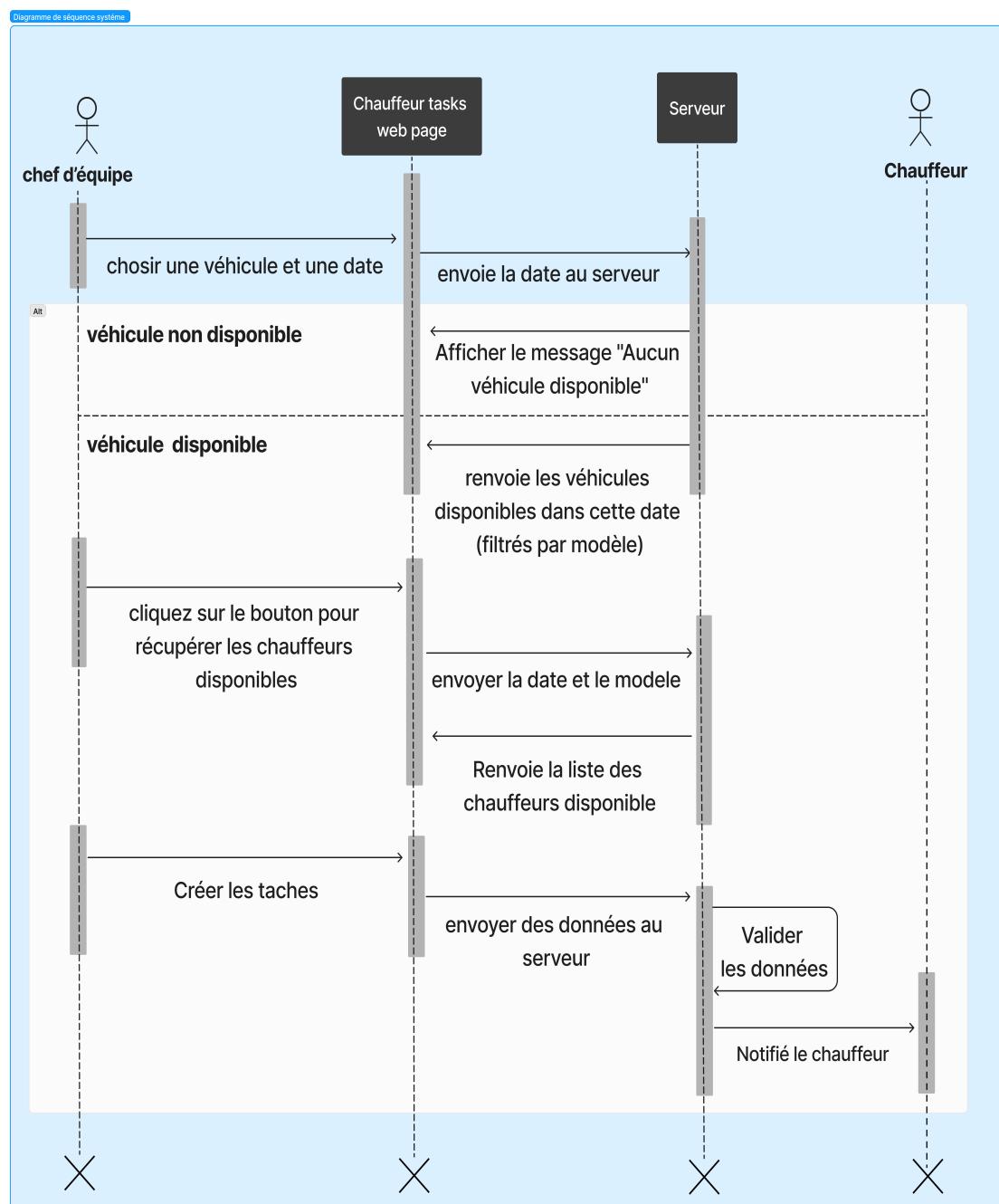


FIGURE 4.5 – Diagramme de séquence système du cas d'utilisation «Créer les tâches»

**4.2.9 Raffinement du cas d'utilisation « Consulter les tâches pour les chauffeurs »**

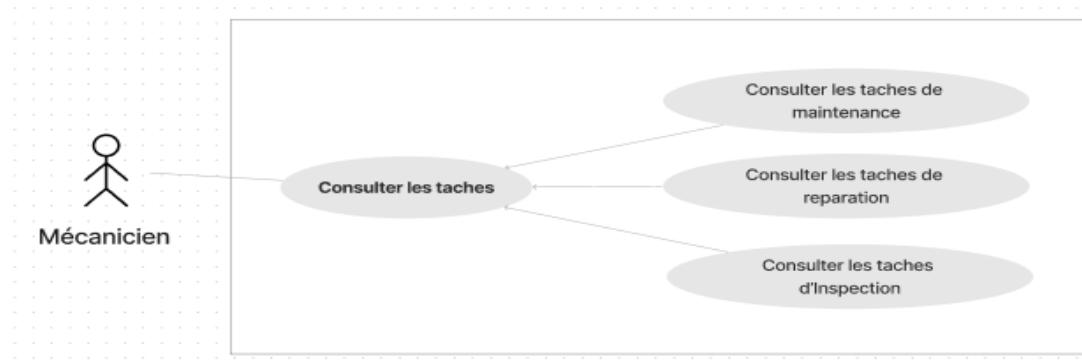


FIGURE 4.6 – Analyse de cas d'utilisation « Consulter les tâches pour les chauffeurs »

**4.2.10 Description textuelle du cas d'utilisation « Consulter les tâches pour les chauffeurs »**

<b>Cas d'utilisation</b>	Consulter les tâches des chauffeurs
<b>Acteur Principal</b>	Chauffeur
<b>Acteurs Secondaires</b>	Chef d'Équipe
<b>Pré-condition</b>	1- Le chauffeur doit être connecté à l'application . 2- Des tâches doivent être assignées au chauffeur.
<b>Post-condition</b>	1- Le chauffeur a consulté les détails de ses tâches. 2- Le chauffeur peut confirmer la réalisation des tâches. 3- Le chef d'équipe est informé de l'avancement ou de l'achèvement des tâches.
<b>Scenario nominal</b>	1- Lorsqu'une nouvelle tâche est assignée, le chauffeur reçoit une notification sur son application mobile. 2- Le chauffeur est invité à scanner un code QR pour accéder à sa checklist quotidienne. 3- Le système affiche la checklist quotidienne du chauffeur, comprenant toutes les tâches assignées avec des détails spécifiques pour chaque tâche.

	<p>4- Une fois qu'une tâche est accomplie, le chauffeur confirme sa réalisation directement depuis l'application en marquant la tâche comme "Terminée".</p> <p>5- Le système notifie automatiquement le chef d'équipe de l'avancement ou de l'achèvement des tâches confirmées par le chauffeur.</p>
<b>Scénario alternatif</b>	<p>.</p> <p>.</p> <p>.</p> <p>.</p> <p>.</p>

TABLE 4.5 – Description textuelle du cas d'utilisation “Consulter les tâches des chauffeurs”

#### 4.2.11 Diagramme d'activité du cas d'utilisation «Consulter les tâches des chauffeurs»

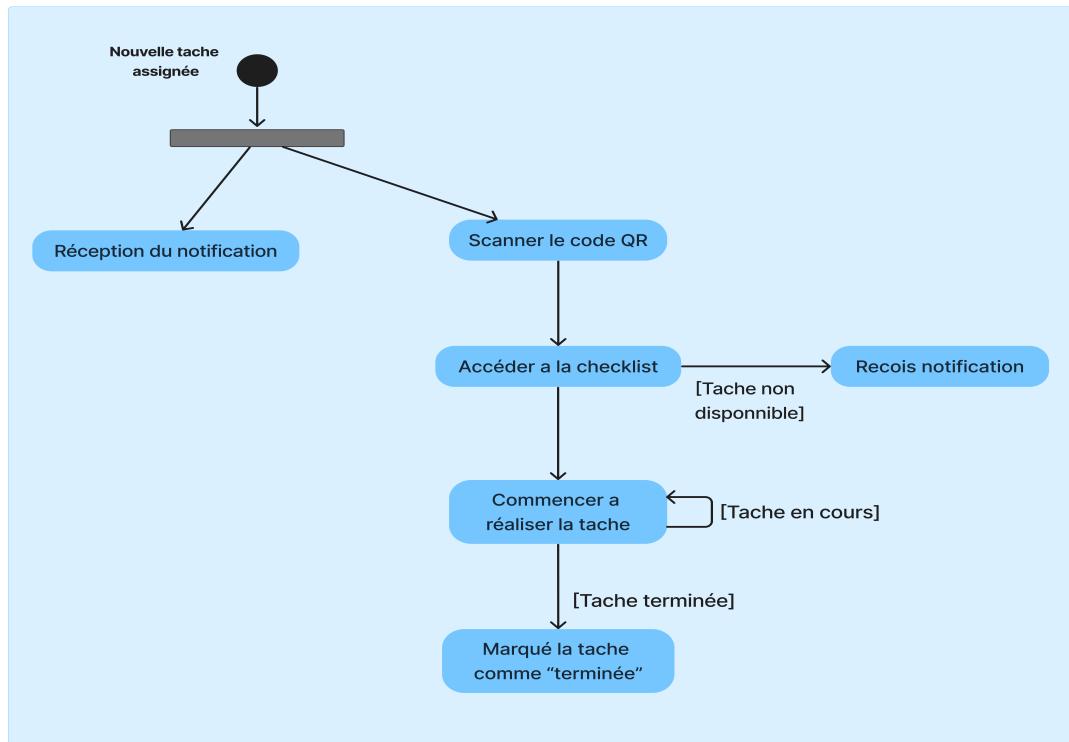


FIGURE 4.7 – Diagramme d'activité du cas d'utilisation « Consulter les tâches des chauffeurs »

**4.2.12 Raffinement du cas d'utilisation « Consulter les tâches pour les mécaniciens »**

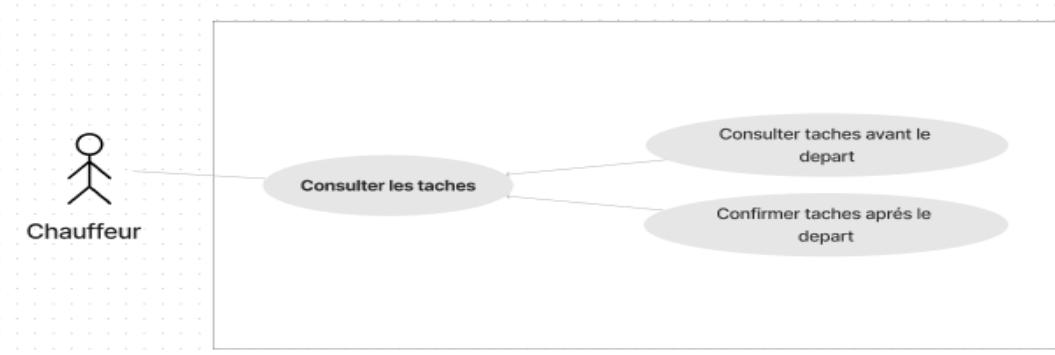


FIGURE 4.8 – Analyse de cas d'utilisation « Consulter les tâches pour les mécaniciens »

**4.2.13 Description textuelle du cas d'utilisation « Consulter les tâches pour les mécaniciens »**

<b>Cas d'utilisation</b>	Consulter les tâches des mécaniciens
<b>Acteur Principal</b>	Mécanicien
<b>Acteurs Secondaires</b>	Chef d'Équipe
<b>Pré-condition</b>	1- Le mécanicien doit être authentifié dans le système. 2- Des tâches doivent être assignées au mécanicien.
<b>Post-condition</b>	1- Le mécanicien est informé des nouvelles tâches. 2- Les tâches sont consultées et confirmées une fois réalisées. 3- Le chef d'atelier est notifié de la réalisation des tâches.
<b>Scénario nominal</b>	1- Le mécanicien reçoit une notification sur son appareil mobile lui indiquant qu'une nouvelle tâche lui a été assignée. 2- Il accède à la liste des tâches disponibles. 3- Après avoir pris connaissance des détails, le mécanicien se met au travail pour effectuer la tâche.

	<p>4- Une fois la tâche terminée, il marque la tâche comme réalisée dans l'application.</p> <p>5- Le système envoie automatiquement une notification à son chef pour l'informer de l'achèvement de la tâche.</p> <p>6- Le chef vérifie la réalisation de la tâche dans le système et valide l'achèvement.</p>
<b>Scénario alternatif</b>	<p>.</p> <p>.</p> <p>.</p> <p>.</p>

TABLE 4.6 – Description textuelle du cas d'utilisation “Consulter les tâches des mécaniciens”

### 4.3 Diagramme de classes du sprint 2

### 4.4 Réalisation

Dans cette section, nous présentons les avancées réalisées durant le sprint 2. À travers des captures d'écran et des descriptions détaillées, nous illustrons les nouvelles fonctionnalités intégrées et les améliorations apportées aux interfaces existantes.

#### 4.4.1 Interface de création des tâches

Cette interface permet au chef d'équipe de créer des tâches pour les chauffeurs et les mécaniciens. Le chef d'équipe peut choisir le type de tâche à créer en sélectionnant soit une tâche pour un chauffeur, soit une tâche pour un mécanicien.

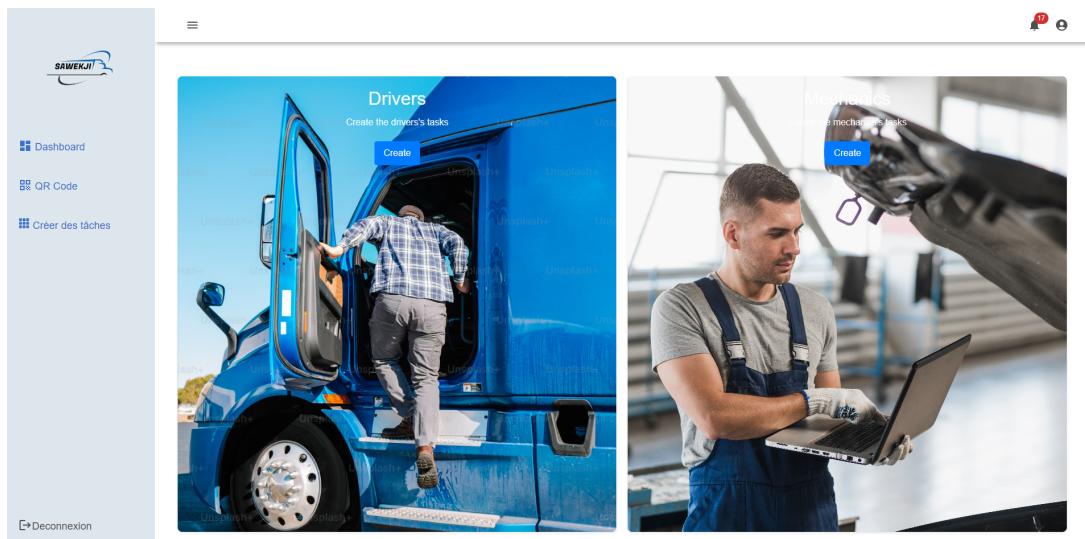


FIGURE 4.9 – interface de création des tâches

## CHAPITRE 4. SPRINT 2 : GESTION DES TÂCHES

Cette interface permet au chef d'équipe de créer des tâches spécifiques pour les chauffeurs. Le processus inclut la sélection du modèle de voiture, la date de la tâche, une description détaillée de la tâche, et enfin, l'assignation de la tâche à un chauffeur spécifique. Cette capture d'écran montre les différentes options et champs à remplir pour une création de tâche complète et précise.

The screenshot shows a user interface for creating driver tasks. On the left is a sidebar with a logo and three menu items: 'Dashboard', 'QR Code', and 'Créer des tâches'. Below the sidebar is a 'Déconnexion' button. The main area has a header with the date '05/06/2024' and a button 'Fetch Free Mechanics'. A table lists three drivers: 'ena', 'haroun', and 'houssem'. Each row contains a 'Name' column, a 'Tasks' column with an input field 'Enter tasks...', and an 'Action' column with a 'Save' button. The table has a light gray background with white borders for the rows and columns.

FIGURE 4.10 – Création des tâches pour les chauffeurs

Cette interface est dédiée à la création de tâches pour les mécaniciens. Le chef d'équipe peut sélectionner la date de la tâche, puis visualiser la liste des mécaniciens disponibles pour cette date. Ensuite, il peut assigner la tâche à un mécanicien particulier. Cette capture d'écran met en évidence les étapes de sélection et d'assignation, offrant une vue claire et fonctionnelle pour la gestion des interventions mécaniques.

The screenshot shows a user interface for creating mechanic tasks. On the left is a sidebar with a logo and three menu items: 'Dashboard', 'QR Code', and 'Créer des tâches'. Below the sidebar is a 'Déconnexion' button. The main area features a large image of a white Isuzu pickup truck with a 'Pick Up' button and a 'CHOOSE' button below it. Above the image is a date input '05/06/2024' and a dropdown 'pickup'. To the right is a button 'FETCH FREE TRUCKS'. Below the image is a table for task creation. The table has columns: 'Matricule', 'Type', 'Task ID', 'Task Description', and 'Actions'. It contains two rows: one for '20155' (Type: pickup) and another for '899' (Type: pickup). Each row includes an 'Enter Task ID' input, a 'Task Description' input, and a row of buttons: 'Fetch Free Drivers', 'Select Driver', and 'Save'. At the bottom of the table is a 'Go to QR Code' button.

FIGURE 4.11 – Création des tâches pour les mécaniciens

### 4.4.2 Consultation du checklist du chauffeur

Cette interface mobile permet aux chauffeurs de consulter leurs tâches assignées. Lorsqu'une nouvelle tâche est créée, le chauffeur reçoit une notification et est invité à scanner un code QR pour accéder à sa checklist quotidienne. Cette checklist comprend toutes les tâches à accomplir, avec des détails spécifiques pour chaque tâche. Le chauffeur peut alors confirmer la réalisation des tâches directement depuis l'application, ce qui notifie automatiquement le chef d'équipe de l'avancement ou de l'achèvement des tâches.

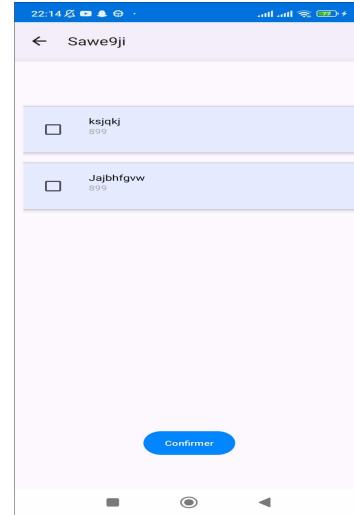


FIGURE 4.12 – interface de consultation du checklist pour les chauffeurs

### 4.4.3 Consultation des tâches du mécanicien

Cette interface mobile est destinée aux mécaniciens pour la gestion de leurs tâches assignées. À chaque nouvelle tâche, le mécanicien reçoit une notification l'invitant à consulter les détails de la tâche sur l'application. L'interface affiche les nouvelles tâches avec les informations nécessaires. Le mécanicien peut alors confirmer la réalisation de chaque tâche, ce qui envoie une notification au chef d'équipe pour informer de l'état d'avancement.

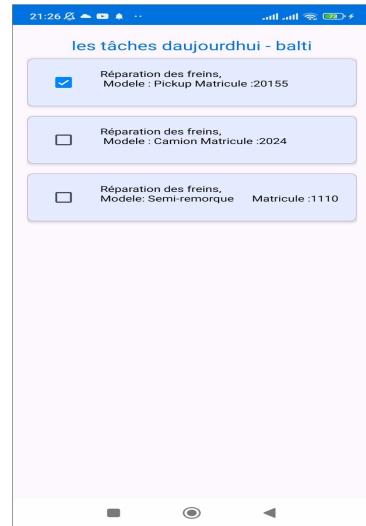


FIGURE 4.13 – interface de consultation des tâches pour les mécaniciens

## 4.5 Rétrospective

Dans cette rétrospective du Sprint 2, nous évaluerons les performances de notre équipe en utilisant le burndown chart et le task board. Ces outils nous aident à identifier les succès, les défis rencontrés, et les possibilités d'amélioration pour les prochains sprints.

### 4.5.1 Burdown Chart

Nous avons planifié un sprint de 3 semaines, avec une moyenne de 6 heures de travail par jour.

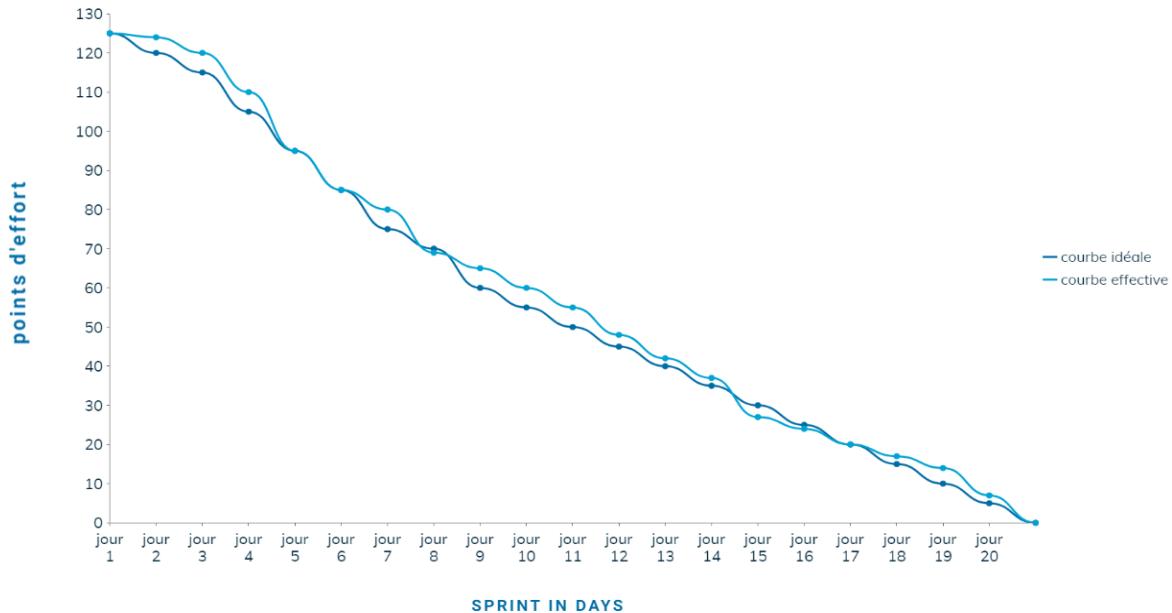


FIGURE 4.14 – Burdown Chart du Sprint 2

### 4.5.2 Task Bord

La figure 4.14 présente le "Task Bord" correspondant au 9ème jour du sprint 2.



FIGURE 4.15 – Task board du Sprint 2

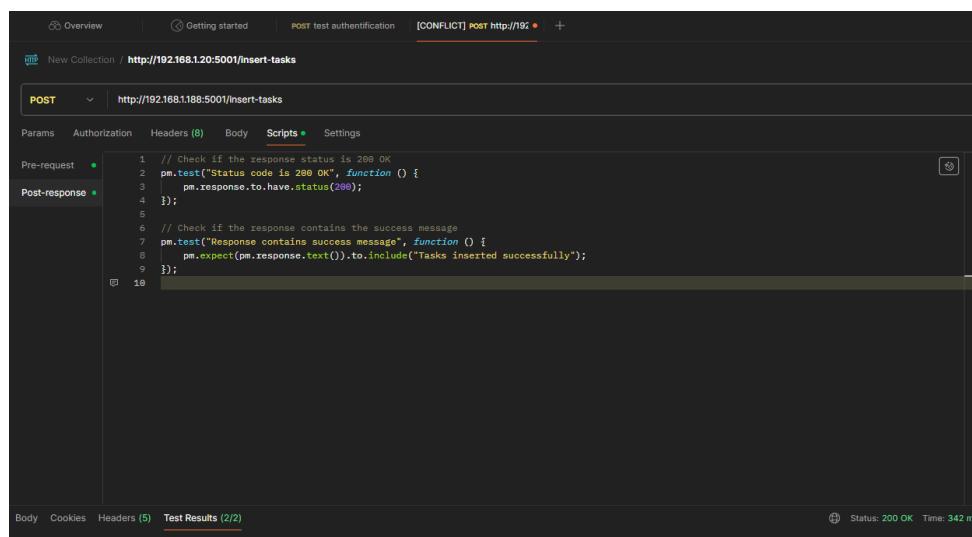
## 4.6 tests unitaires

### Test unitaires du cas d'utilisation « Créeer taches » :

Nous présentons ci-dessous le code source et le résultat de deux cas de test :

**1. Test de réussite :** Le test est considéré comme réussi lorsque le chef d'équipe crée des tâches en utilisant des ID valides existants.

**2. Test d'échec :** Le test est considéré comme échoué si le chef d'équipe tente d'utiliser un ID inexistant lors de la création d'une nouvelle tâche.



The screenshot shows the Postman interface with a collection named "New Collection / http://192.168.1.20:5001/insert-tasks". A POST request is defined with the URL "http://192.168.1.188:5001/insert-tasks". The "Scripts" tab is selected, displaying the following code:

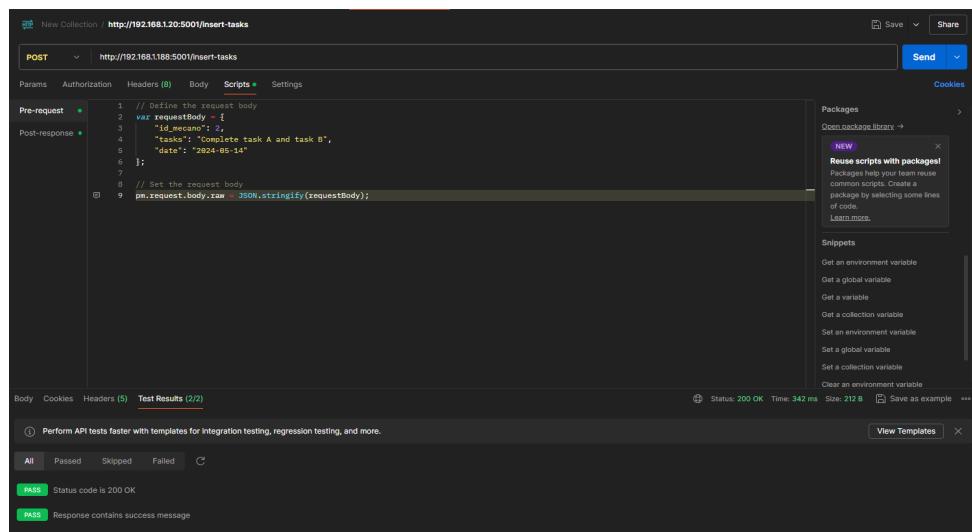
```

1 // Check if the response status is 200 OK
2 pm.test("Status code is 200 OK", function () {
3     pm.response.to.have.status(200);
4 });
5
6 // Check if the response contains the success message
7 pm.test("Response contains success message", function () {
8     pm.expect(pm.response.text()).to.include("Tasks inserted successfully");
9 });
10

```

The "Test Results" section shows 2/2 tests passed. The status bar at the bottom indicates "Status: 200 OK Time: 342 ms".

FIGURE 4.16 – Code souce du test unitaire du cas d'utilisation « Créeer taches »



The screenshot shows the Postman interface with the same collection and request setup as Figure 4.16. The "Scripts" tab now displays a modified version of the code, including a "requestBody" variable definition:

```

1 // Define the request body
2 var requestBody = {
3     "id_mecano": 2,
4     "tasks": "complete task A and task B",
5     "date": "2024-05-14"
6 };
7
8 // Set the request body
9 pm.request.body.raw = JSON.stringify(requestBody);

```

A context menu is open over the "requestBody" variable, showing options like "Open package library", "Reuse scripts with packages!", and "Snippets". The "Test Results" section shows 2/2 tests passed. The status bar at the bottom indicates "Status: 200 OK Time: 342 ms Size: 212 B".

FIGURE 4.17 – Résultat du test de réussite

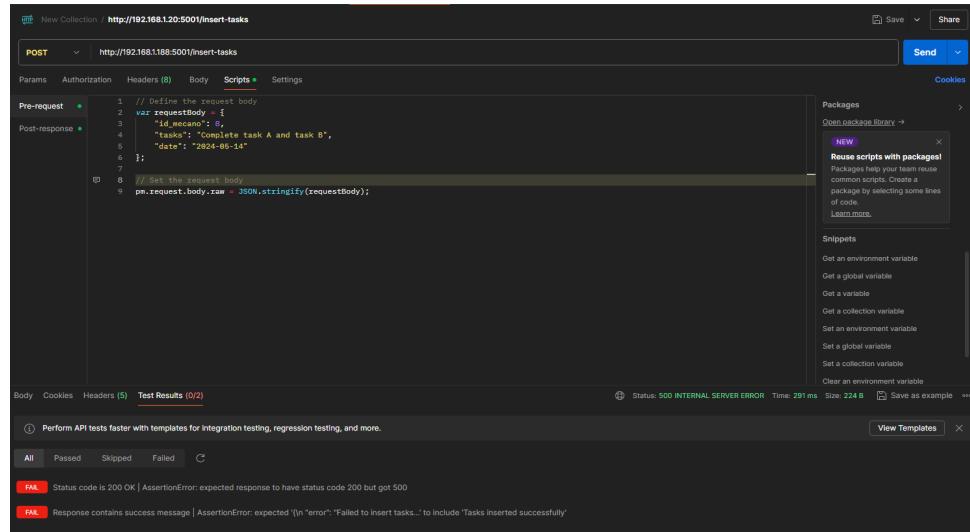


FIGURE 4.18 – Résultat du test d'échec

## Conclusion

Dans le sprint 2, nous avons implémenté avec succès la création des tâches par le chef d'équipe et leur consultation par les chauffeurs et les mécaniciens. Les diagrammes de cas d'utilisation et les séquences système ont été efficacement déployés, accompagnés de descriptions textuelles et de captures d'interfaces qui illustrent clairement les avancées réalisées. Les tests unitaires ont confirmé la fiabilité des nouvelles fonctionnalités. Dans le prochain chapitre, nous explorerons le Sprint 3 : 'Suivi et Crédit des Rapports'.

## Chapitre 5

# Sprint 3 : Suivi et Creation des Rapports

## Introduction

Le sprint 3 se concentre sur l'amélioration de la communication et de la visibilité au sein de l'équipe à travers quatre principales fonctionnalités : les notifications, le suivi en temps réel des véhicules, la planification des déplacements et la génération de rapports détaillés pour les interventions. Ce sprint inclut des sections détaillées avec des diagrammes pour clarifier les besoins et processus. La phase de "Realisation" se concentre sur l'implémentation des fonctionnalités, tandis que la "Retrospective" et les "Tests unitaires" évaluent les méthodes de développement pour assurer la qualité du produit.

### 5.1 Sprint Backlog

ID	Fonctionnalités	User Story	Story points
<b>US6</b>	Suivi en temps réel	En tant que Chef d'équipe je veux suivre le progrès et la localisation des voitures .	8
<b>US7</b>	Planification des déplacements	En tant que chauffeur, je veux planifier des voyages pour optimiser mes itinéraires et mes arrêts.	5
<b>US8</b>	Notifications	En tant qu'administrateur, chauffeur, mécanicien ou chef d'équipe, je veux recevoir des notifications pour être informé en temps réel des mises à jour.	5
<b>US9</b>	Creation des rapports	En tant que Mécanicien je veux Générer des rapports pour que je puisse documenter les détails de chaque intervention .	3

TABLE 5.1 – Backlog du sprint 3

## 5.2 Analyse et conception

Nous mettrons en avant le diagramme de cas d'utilisation globale, illustrant les interactions principales des utilisateurs avec le système. Ensuite, pour chaque fonctionnalité identifiée, nous élaborerons des cas d'utilisation spécifiques, accompagnés de diagrammes de séquence système pour détailler l'ordre des opérations. Enfin, nous fournirons une description textuelle détaillée pour chaque cas d'utilisation, décrivant les étapes, les actions et les résultats attendus.

### 5.2.1 Diagramme de cas d'utilisation du sprint 3

Ce diagramme de cas d'utilisation présente les interactions des différents acteurs avec le système pour le Sprint 3, qui se concentre sur les fonctionnalités de suivi en temps réel, de consultation des notifications et de création de rapports.

Les actions des utilisateurs nécessitent une authentification avant d'accéder à leurs fonctionnalités spécifiques.

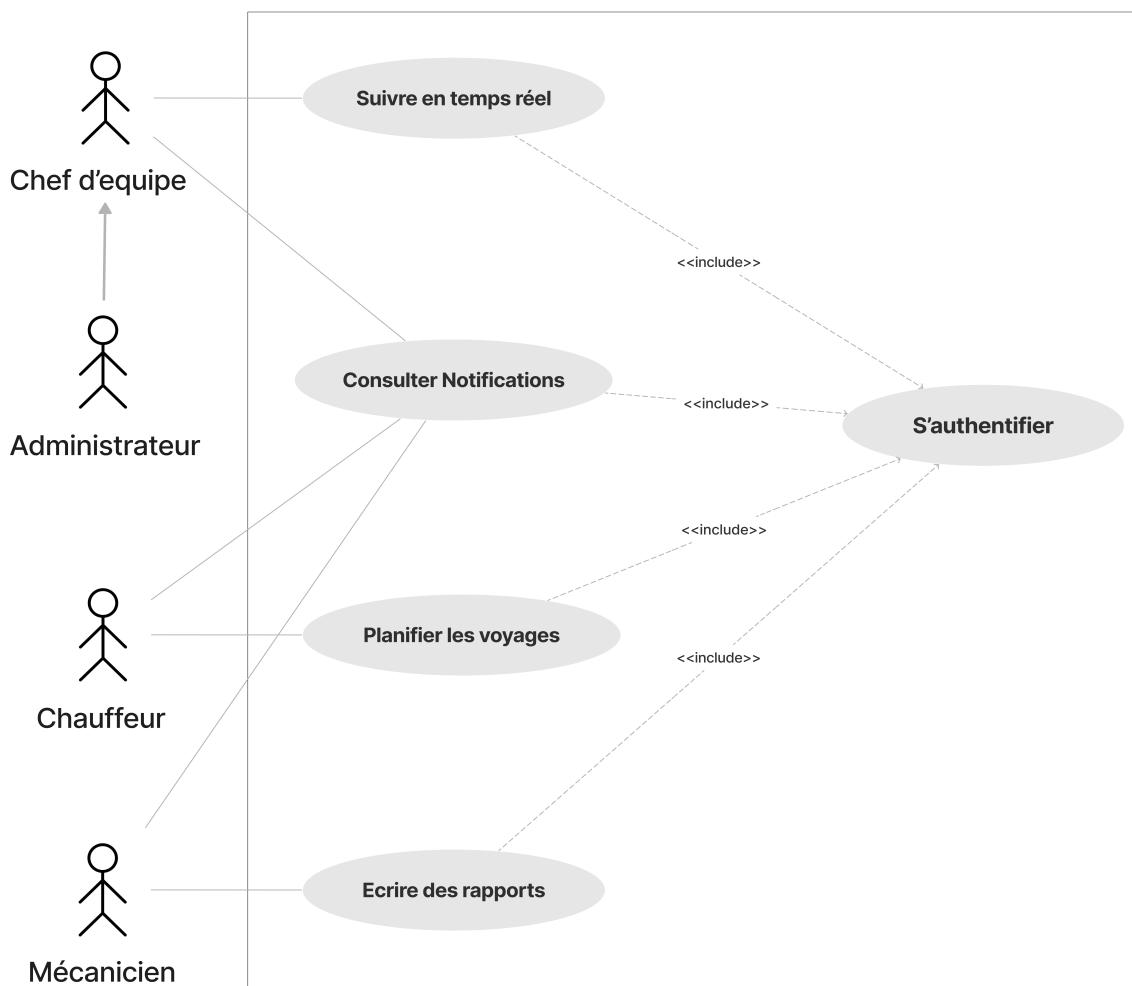


FIGURE 5.1 – Diagramme de cas d'utilisation global du sprint 3

**5.2.2 Description textuelle du cas d'utilisation « Suivi en Temps Réel par Géolocalisation »**

<b>Cas d'utilisation</b>	Suivre en temps réel
<b>Acteur</b>	Chef d'équipe et Chauffeur
<b>Pré-condition</b>	<p>1- Le chauffeur doit être authentifié dans l'application mobile.</p> <p>2- Le chauffeur doit avoir activé le suivi GPS sur son téléphone.</p> <p>3- Le chef d'équipe doit être authentifié dans l'application mobile et avoir accès à l'interface de suivi en temps réel.</p>
<b>Post-condition</b>	<p>1- Le chef peut suivre la position du camion en temps réel sur la carte.</p> <p>2- Le chauffeur peut pinner les lieux où il fait une pause ou où il remplit du carburant.</p> <p>3- Les pins apparaissent en temps réel sur la carte du chef.</p>
<b>Scenario nominal</b>	<p>1- Le chauffeur s'authentifie dans l'application mobile (appel au cas d'utilisation "S'authentifier").</p> <p>2- Le chauffeur active le suivi GPS.</p> <p>3- Le chauffeur sélectionne le trajet optimal sur la carte .</p> <p>4- Pendant le trajet, le chauffeur peut localiser et sélectionner une station de carburant à proximité du trajet prévu.</p> <p>5- Toutes les actions sont mises à jour en temps réel sur la carte.</p> <p>6- Le chef s'authentifie dans l'application (appel au cas d'utilisation "S'authentifier").</p> <p>7- Le chef accède à l'interface de suivi en temps réel.</p>

## **CHAPITRE 5. SPRINT 3 : SUIVI ET CRÉATION DES RAPPORTS**

---

	<p>8- Le système vérifie la position actuelle du camion et l'affiche sur la carte.</p> <p>9- Le système vérifie l'emplacement, l'enregistre, et le marque sur la carte.</p> <p>10- Le chef voit l'emplacement marqué en temps réel sur la carte.</p>
<b>Scénario alternatif</b>	<p>A1 Position GPS non disponible :</p> <p>l'enchainement A1 démarre au point 5 du scénario nominal.</p> <p>6- Un message d'erreur s'affiche, indiquant que la position GPS n'est pas disponible..</p> <p>Le scénario nominal reprend au point 2.</p>

TABLE 5.2 – Description textuelle du cas d'utilisation “Suivi en Temps Réel par Géolocalisation”

**5.2.3 Description textuelle du cas d'utilisation « Planifier les déplacements »**

<b>Cas d'utilisation</b>	Planifier les déplacements
<b>Acteur</b>	Chauffeur
<b>Pré-condition</b>	Le chauffeur doit être connecté à l'application et avoir accès à la carte interactive.
<b>Post-condition</b>	Les détails du trajet choisi, des pauses et des arrêts pour le carburant sont enregistrés dans le système.
<b>Scénario nominal</b>	<p>1- Le chauffeur sélectionne le trajet optimal sur la carte .</p> <p>2- Pendant le trajet, le chauffeur peut localiser et sélectionner une station de carburant à proximité du trajet prévu.</p> <p>3- Toutes les actions sont mises à jour en temps réel sur la carte.</p>
<b>Scénario alternatif</b>	<p>A1 Station de carburant fermée ou inaccessible :</p> <p>l'enchaînement A1 démarre au point 1 du scénario nominal.</p> <p>2- le chauffeur recherche une autre station de carburant disponible à proximité sur la carte.</p> <p>Le scénario nominal reprend au point 2.</p>

TABLE 5.3 – Description textuelle du cas d'utilisation “ Planifier les déplacements ”

**5.2.4 Diagramme de séquence système du cas d'utilisation « Planifier les déplacements »**

### 5.2.5 Description textuelle du cas d'utilisation « Consulter Notifications »

<b>Cas d'utilisation</b>	Consulter les notifications
<b>Acteur</b>	Administrateur, Chef d'équipe, Chauffeur et Mécanicien
<b>Pré-condition</b>	1- L'utilisateur doit être authentifié dans l'application mobile.  2- L'utilisateur doit avoir accès à l'interface de consultation des notifications.
<b>Post-condition</b>	L'utilisateur peut voir toutes les notifications.
<b>Scenario nominal</b>	1- L'utilisateur s'authentifie dans l'application (appel au cas d'utilisation "S'authentifier").  2- L'utilisateur accède à l'interface de consultation des notifications.  3- Le système affiche la liste des notifications, triées par ordre chronologique.

TABLE 5.4 – Description textuelle du cas d'utilisation “ Consulter Notifications ”

### 5.2.6 Diagramme d'activité du cas d'utilisation « Consulter Notifications »

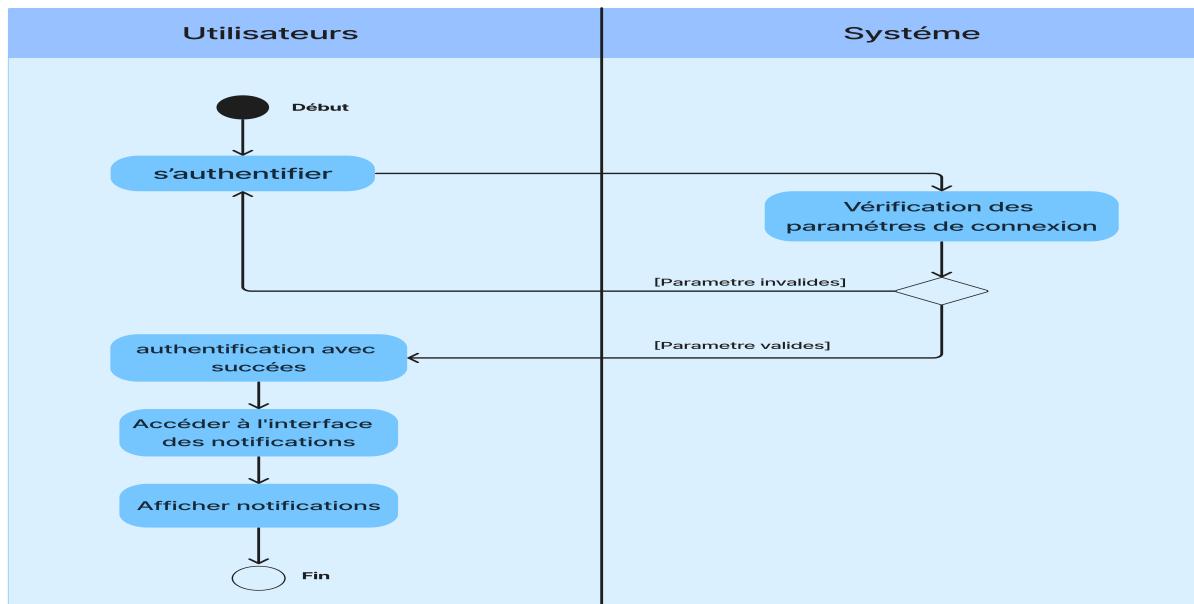


FIGURE 5.2 – Diagramme d'activité du cas d'utilisation « Consulter Notifications »

### 5.2.7 Raffinement du cas d'utilisation « Générer des rapports »

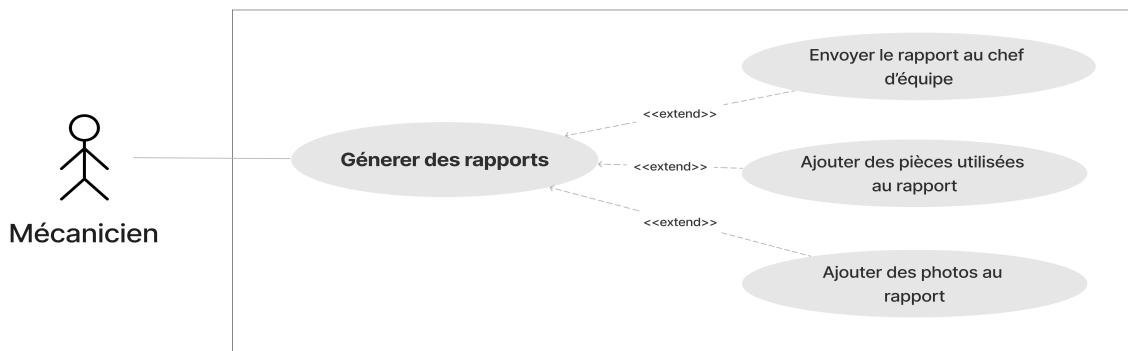


FIGURE 5.3 – Raffinement du cas d'utilisation « Générer des rapports »

### 5.2.8 Description textuelle du cas d'utilisation « Générer des rapports »

<b>Cas d'utilisation</b>	Écrire des rapports
<b>Acteur</b>	Mécanicien
<b>Pré-condition</b>	Le mécanicien doit être authentifié dans l'application mobile et a l'accès à l'interface de saisie des rapports.
<b>Post-condition</b>	<p>1-Le rapport est enregistré dans le système avec tous les détails requis.</p> <p>2- Les rapports existants s'affichent dans l'interface en dessous du formulaire de saisie.</p>
<b>Scénario nominal</b>	<p>1- Le mécanicien s'authentifie dans l'application mobile (appel au cas d'utilisation "S'authentifier").</p> <p>2- Le mécanicien accède à l'interface de saisie des rapports.</p> <p>3- Le mécanicien saisit la matricule du véhicule.</p> <p>4- Le mécanicien entre les détails de l'intervention (description des travaux effectués, pièces remplacées).</p> <p>5- Le mécanicien appuie sur le bouton "Confirmer le rapport".</p> <p>6- Le système vérifie la matricule du véhicule.</p>

## CHAPITRE 5. SPRINT 3 : SUIVI ET CRÉATION DES RAPPORTS

	<p>7- Le système enregistre les informations saisies et génère un rapport.</p>
<b>Scénario alternatif</b>	<p>A1 Erreur de saisie de la matricule :</p> <p>l'enchaînement A1 démarre au point 6 du scénario nominal.</p> <p>4- un message d'erreur va être afficher.</p> <p>Le scénario nominal reprend au point 3.</p>

TABLE 5.5 – Description textuelle du cas d'utilisation “ Générer des rapports ”

### 5.2.9 Diagramme de séquence système du cas d'utilisation « Générer des rapports »

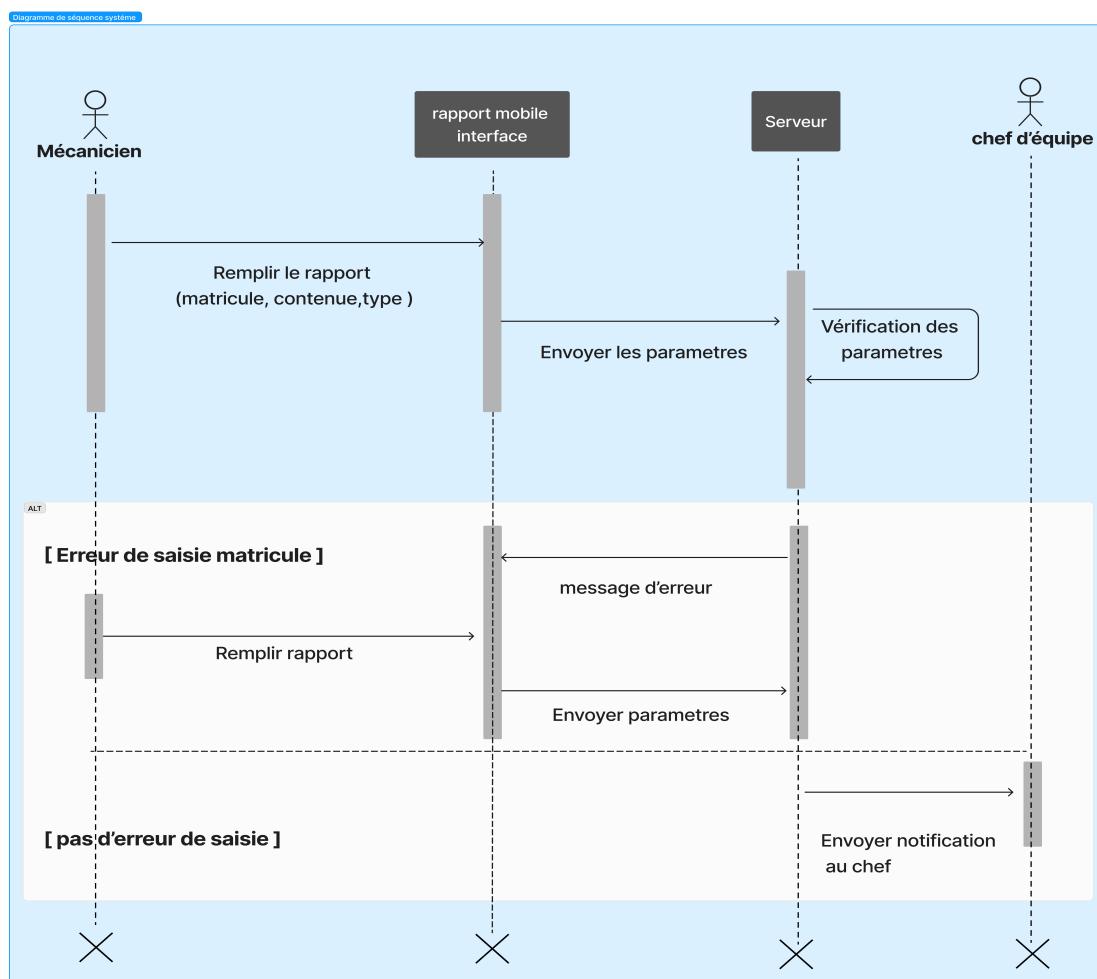


FIGURE 5.4 – Diagramme de séquence système du cas d'utilisation « Générer des rapports »

### **5.3 Diagramme de classes du sprint 3**

### 5.4 Réalisation

Cette section présente les interfaces développées lors du sprint 3.

#### 5.4.1 Suivi en Temps Réel par Géolocalisation

L’interface web permet au chef d’équipe de suivre en temps réel la position des véhicules sur la carte et de surveiller les itinéraires des chauffeurs.

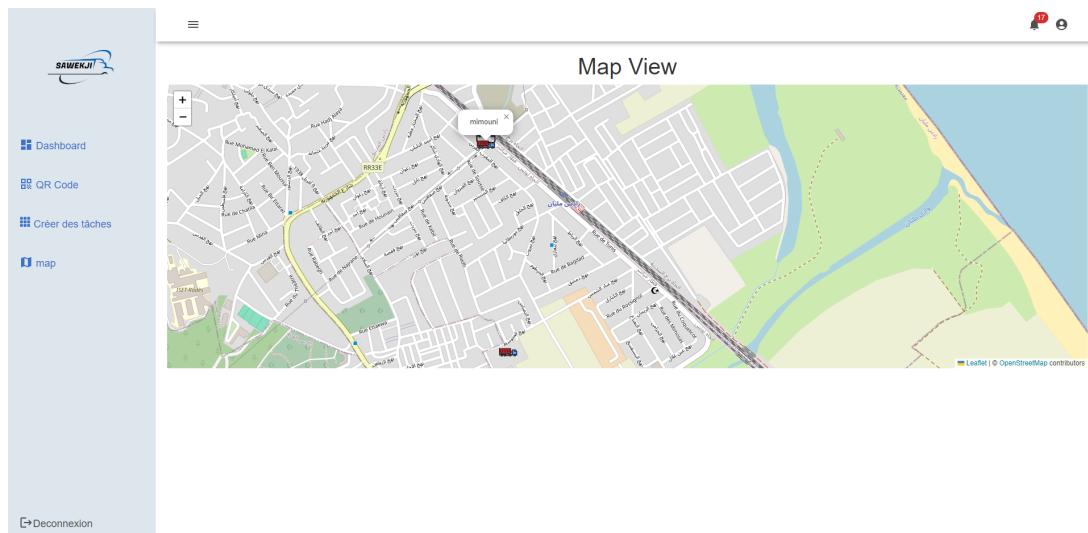
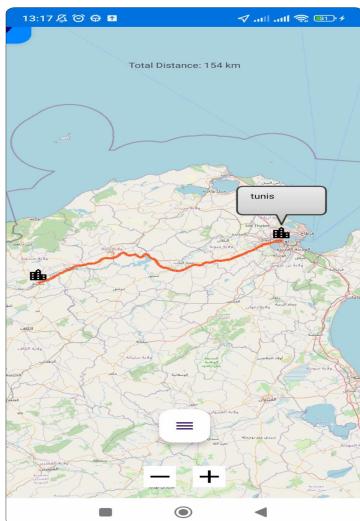


FIGURE 5.5 – Suivi en Temps Réel par Géolocalisation

#### 5.4.2 La planification des déplacements



Cette interface offre une vue d’ensemble du trajet, avec le chemin complet mis en surbrillance, la distance totale du voyage affichée en haut, et des informations sur les points de passage et la destination finale.

FIGURE 5.6 – Vue du trajet

Cette interface affiche une carte avec des stations-service (icônes de pompe à essence) pour les ravitaillements, des aires de repos pour les pauses du chauffeur, et la position actuelle du véhicule .

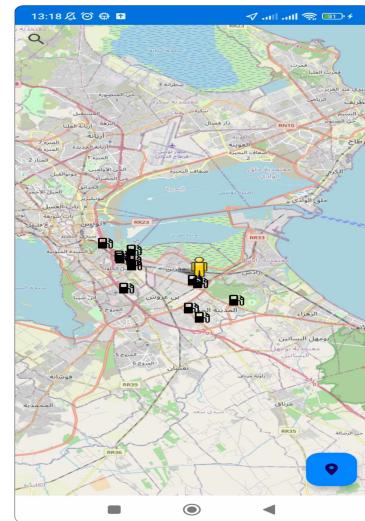


FIGURE 5.7 – Carte de voyage

#### 5.4.3 Les Notifications

L’interface web est principalement destinée au chef d’équipe et permet de consulter les notifications relatives à l’avancement des chauffeurs et des mécaniciens. Lorsqu’un chauffeur ou un mécanicien réalise une tâche, une notification est envoyée au chef d’équipe, qui peut alors confirmer la réalisation de la tâche. Cela permet un suivi en temps réel des activités . L’interface mobile, quant à elle, est destinée aux chauffeurs, aux mécaniciens et au chef d’équipe. Elle permet de notifier ces utilisateurs à chaque fois qu’une tâche leur est assignée ou lorsqu’ils doivent accomplir une tâche spécifique. Le chef d’équipe reçoit également des notifications mobiles pour suivre l’avancement des tâches en temps réel, ce qui lui permet de rester informé même lorsqu’il n’est pas devant son ordinateur.

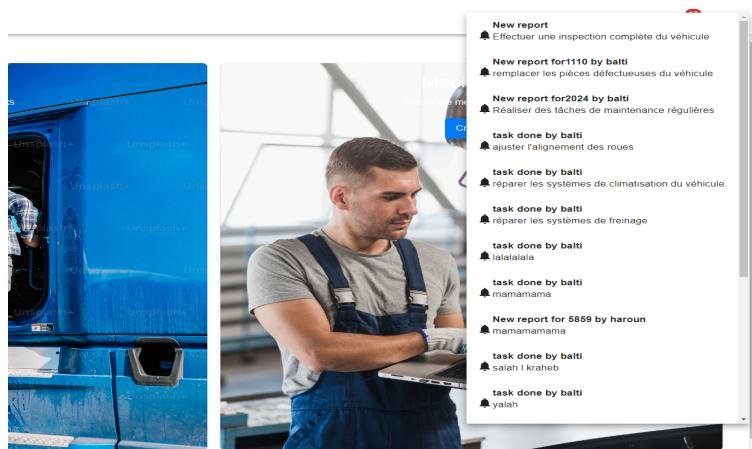


FIGURE 5.8 – Interface de Consultation des notifications - web »



FIGURE 5.9 – Interface de Consultation des notifications - Mobile »

### 5.4.4 Génération d'un Rapport

Cette interface permet au mécanicien d'entrer le matricule du véhicule et de rédiger un rapport. Deux boutons sont disponibles : "Importer image" pour ajouter une photo et "Sauvegarder" pour enregistrer le rapport.

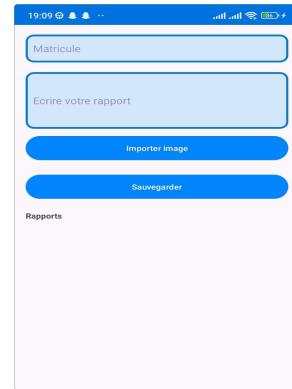


FIGURE 5.10 – Génération de Rapport - Interface Initiale

Le mécanicien a saisi le matricule et les détails du rapport dans cette interface. Les champs de texte affichent les informations saisies.

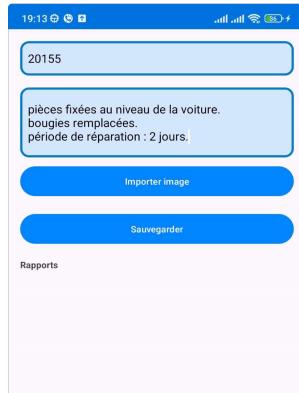


FIGURE 5.11 – Génération de Rapport - Détails Entrés

Cette interface affiche le rapport sauvegardé avec le matricule et les détails du rapport. Les champs pour entrer un nouveau rapport sont toujours accessibles, ainsi que les boutons pour importer une image ou sauvegarder.



FIGURE 5.12 – Génération de Rapport - Rapport Sauvegardé

## 5.5 Rétrospective

Pour cette rétrospective du Sprint 3, nous évaluerons les performances de notre équipe, le burndown chart et le task board nous permettront d'identifier les succès, les défis rencontrés, ainsi que les opportunités d'amélioration pour le prochain sprint.

### 5.5.1 Burdown Chart

Nous avons planifié un sprint de 3 semaines, avec une moyenne de 7 heures de travail par jour.

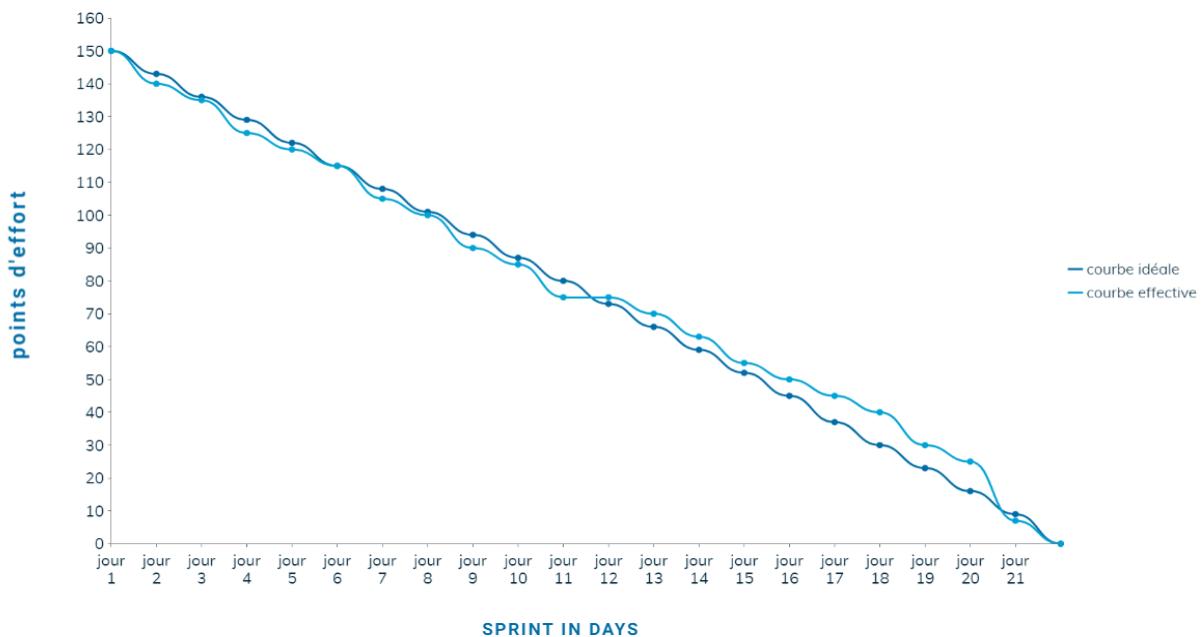


FIGURE 5.13 – Burdown Chart du Sprint 3

### 5.5.2 Task Bord

La figure 4.\*\* présente le "Task Bord" correspondant au 3ème jour du sprint 3.

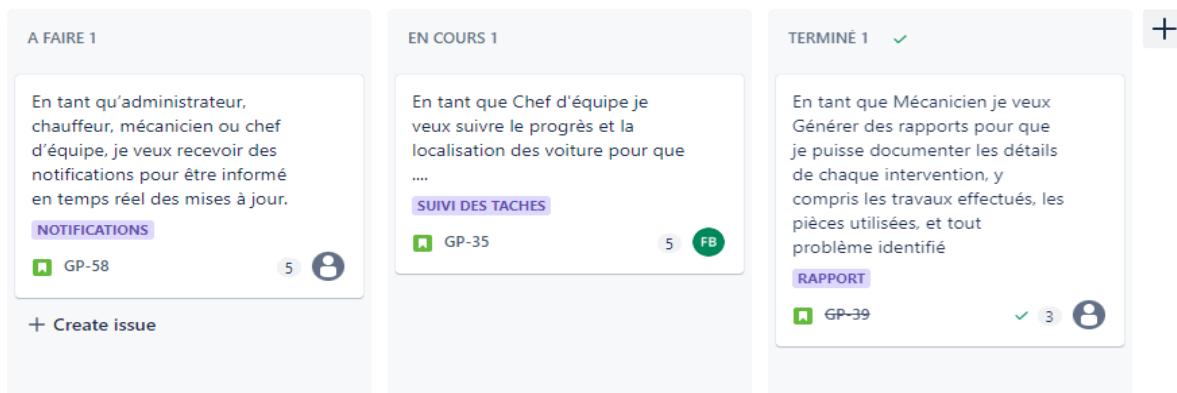


FIGURE 5.14 – Task board du Sprint 3

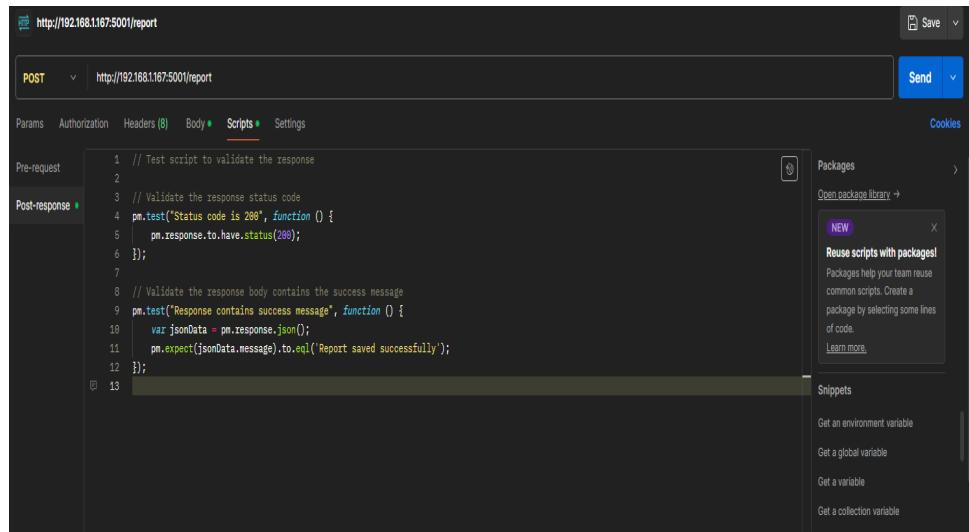
## 5.6 Tests unitaires

### Test unitaires du cas d'utilisation « Générer des rapports » :

Nous présentons ci-dessous le code source et le résultat de deux cas de test :

**1. Test de réussite :** Le test est considéré comme réussi lorsque ...

**2. Test d'échec :** Le test est considéré comme échoué si ...



The screenshot shows the Postman interface with a POST request to `http://192.168.1.167:5001/report`. The 'Scripts' tab is selected, displaying the following test script:

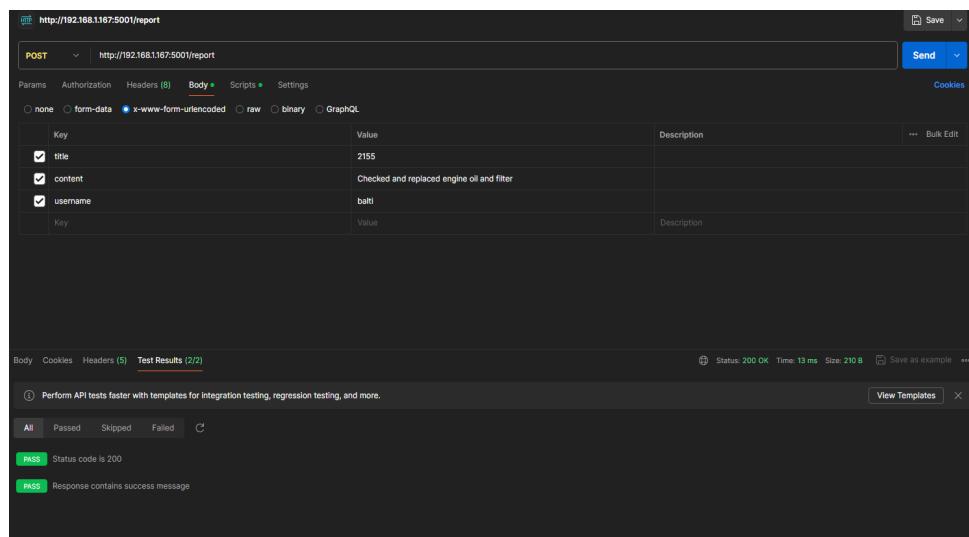
```

1 // Test script to validate the response
2
3 // Validate the response status code
4 pm.test("Status code is 200", function () {
5     pm.response.to.have.status(200);
6 });
7
8 // Validate the response body contains the success message
9 pm.test("Response contains success message", function () {
10    var jsonData = pm.response.json();
11    pm.expect(jsonData.message).to.eql('Report saved successfully');
12 });
13

```

The right sidebar shows a 'Reuse scripts with packages!' dialog and a 'Snippets' section.

FIGURE 5.15 – Code source du test unitaire du cas d'utilisation « Générer des rapports »



The screenshot shows the Postman interface after running the test. The 'Test Results' tab indicates 2/2 tests passed. The results show:

- Passed: Status code is 200
- Passed: Response contains success message

The status bar at the bottom shows: Status 200 OK Time: 13 ms Size: 210 B Save as example View Templates

FIGURE 5.16 – Résultat du test de réussite

## CHAPITRE 5. SPRINT 3 : SUIVI ET CRÉATION DES RAPPORTS

---

The screenshot shows a POST request to `http://192.168.1.167:5001/report`. The 'Body' tab is selected, showing three parameters: 'title' (volvo), 'content' (Checked and replaced engine oil and filter), and 'username' (bali). The 'Test Results' tab indicates 5 tests, with two failing assertions: 'Status code is 200 | AssertionError: expected response to have status code 200 but got 500' and 'Response contains success message | AssertionError: expected undefined to deeply equal 'Report saved successfully''.

FIGURE 5.17 – Résultat du test d'échec

## Conclusion

Ce sprint a consolidé le suivi en temps réel, la planification des déplacements, les notifications et la génération de rapports. Les interfaces web et mobile ont renforcé la coordination, permettant au chef d'équipe de suivre les véhicules et aux chauffeurs de gérer efficacement les itinéraires. La simplification de la génération de raports a amélioré l'efficacité des mécaniciens. Les ajustements basés sur les outils de suivi préparent efficacement la transition vers le dernier sprint.