

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 avec un tableau de bord interactif**

Organisme d'accueil : Digital Identity



Réalisé par :

Balti Firas

Mimouni Mohamed Aziz

Supervisé par :

Encadrante Académique

Mme. Chamakhi Hanen

Encadrant Professionnel

Mr. Neyeb Charfedine

ATTESTATION

ATTESTATION

Table des matières

Table des matières	4
Table des figures	9
Liste des tableaux	13
Introduction générale	15
1 Etude préliminaire du projet	17
Introduction	17
1.1 Présentation de l'organisme d'accueil	17
1.1.1 Organigramme de l'entreprise	18
1.1.2 Les activités de Digital Identity	18
1.2 Cadre du projet	19
1.2.1 Problématique	19
1.2.2 Solution proposée	20
1.3 Méthodologie et langage de conception	20
1.3.1 Méthodes agiles	20
1.3.2 Principes du framework Scrum	21
1.3.3 Rôles du framework Scrum	22
1.3.4 Langage de modélisation	23
1.4 Environnement de travail	23
1.4.1 Langages et framework utilisés	23
1.4.2 Environnement matériel	25
1.4.3 Environnement logiciel	26
1.5 Architecture du Système	29
1.5.1 Architecture logique front-end	29
1.5.2 Architecture physique	29
1.6 Organisation du travail	30
Conclusion	31

TABLE DES MATIÈRES

2 Planification du projet	32
Introduction	32
2.1 Spécification des besoins	32
2.1.1 Identification des acteurs	32
2.1.2 Besoins fonctionnels	33
2.1.3 Besoins non fonctionnels	35
2.1.4 Diagramme de cas d'utilisation	36
2.2 Pilotage du projet avec Scrum	37
2.2.1 Équipe et rôles	37
2.2.2 Contraintes et exigences	37
2.2.3 Risques	37
2.2.4 Backlog du produit	37
2.3 Conception de l'application	39
2.3.1 Prototypes de l'application	39
2.3.2 Charte Graphique	43
2.3.3 Plan de l'application	43
2.3.4 Diagramme de Gantt	49
Conclusion	49
3 Sprint 1 : Gestion des accès	50
Introduction	50
3.1 Backlog du sprint 1	50
3.2 Analyse et conception	51
3.2.1 Diagramme de cas d'utilisation du sprint 1	51
3.2.2 Description textuelle du cas d'utilisation "S'authentifier"	52
3.2.3 Diagramme de séquence système du cas d'utilisation «S'authentifier»	53
3.2.4 Raffinement du cas d'utilisation « Gérer les utilisateurs »	54
3.2.5 Description textuelle du cas d'utilisation "Ajouter utilisateur"	54
3.2.6 Diagramme de séquence système du cas d'utilisation «Ajouter utilisateur»	55
3.3 Diagramme de classes du sprint 1	56
3.4 Réalisation	57
3.4.1 Authentification	57
3.4.2 Profil	58
3.4.3 Gestion des utilisateurs	59
3.4.4 Crédit d'un nouvel utilisateur	59
3.5 Rétrospective	60
3.5.1 Burndown Chart	60
3.5.2 Task Bord	61
3.6 Tests unitaires	61
Conclusion	62

TABLE DES MATIÈRES

4 Sprint 2 : Gestion des Tâches	63
Introduction	63
4.1 Backlog du sprint 2	63
4.2 Analyse et conception	64
4.2.1 Diagramme de cas d'utilisation du sprint 2	64
4.2.2 Raffinement du cas d'utilisation « Gérer les véhicules »	64
4.2.3 Description textuelle du cas d'utilisation « Ajouter véhicules »	65
4.2.4 Diagramme de séquence système du cas d'utilisation « Ajouter des véhicules »	66
4.2.5 Raffinement du cas d'utilisation « Créer les taches »	67
4.2.6 Description textuelle du cas d'utilisation « Créer des Tâches pour les Mécaniciens »	67
4.2.7 Description textuelle du cas d'utilisation « Créer des Tâches pour les Chauffeurs »	68
4.2.8 Diagramme de séquence système du cas d'utilisation «Créer les tâches pour les chauffeurs»	70
4.2.9 Raffinement du cas d'utilisation « Consulter les taches pour les chauffeurs »	71
4.2.10 Description textuelle du cas d'utilisation « Consulter les taches pour les chauffeurs »	71
4.2.11 Diagramme d'activité du cas d'utilisation «Consulter les taches des chauffeurs»	72
4.2.12 Raffinement du cas d'utilisation « Consulter les taches pour les mécaniciens »	73
4.2.13 Description textuelle du cas d'utilisation « Consulter les taches pour les mécaniciens »	73
4.3 Diagramme de classes du sprint 2	74
4.4 Réalisation	75
4.4.1 Interface de création des taches	75
4.4.2 Consultation du checklist du chauffeur	76
4.4.3 Consultation des taches du mécanicien	77
4.4.4 Validation des taches du mécanicien par le chef d'équipe	77
4.5 Rétrospective	78
4.5.1 Burdown Chart	78
4.5.2 Task Bord	78
4.6 tests unitaires	79
Conclusion	80
5 Sprint 3 : Suivi et Crédit des Rapports	81
Introduction	81
5.1 Backlog du sprint 3	81
5.2 Analyse et conception	82
5.2.1 Diagramme de cas d'utilisation du sprint 3	82

5.2.2	Description textuelle du cas d'utilisation « Suivi en Temps Réel par Géolocalisation »	83
5.2.3	Description textuelle du cas d'utilisation « Planifier les déplacements »	84
5.2.4	Description textuelle du cas d'utilisation « Consulter Notifications »	85
5.2.5	Diagramme d'activité du cas d'utilisation « Consulter Notifications »	85
5.2.6	Raffinement du cas d'utilisation « Générer des rapports »	86
5.2.7	Description textuelle du cas d'utilisation « Générer des rapports »	86
5.2.8	Diagramme de séquence système du cas d'utilisation « Générer des rapports »	87
5.3	Diagramme de classes du sprint 3	88
5.4	Réalisation	89
5.4.1	Suivi en Temps Réel par Géolocalisation	89
5.4.2	La planification des déplacements	89
5.4.3	Les Notifications	90
5.4.4	Génération d'un Rapport	91
5.5	Rétrospective	92
5.5.1	Burdown Chart	92
5.5.2	Task Bord	92
5.6	Tests unitaires	93
	Conclusion	94
6	Sprint 4 : Implémentation des tableaux de bord	95
6.1	Backlog du sprint 4	95
6.2	Conception du data warehouse	96
6.2.1	Choix du modèle	96
6.3	Définition des indicateurs de performance	96
6.4	Schéma en constellation	97
6.4.1	Les tables de fait	97
6.4.2	Les tables de dimensions	98
6.5	Data mart : Chauffeur	101
6.5.1	Schéma conceptuel du DataMart « Chauffeur »	101
6.6	Data mart : Véhicules	102
6.6.1	Schéma conceptuel du DataMart « Véhicules »	102
6.7	Data mart : Mécanicien	103
6.7.1	Schéma conceptuel du DataMart « Mécanicien »	103
6.8	Data mart Utilisateurs	104
6.8.1	Schéma conceptuel du DataMart « Utilisateurs »	104
6.9	Le développement de l'ETL	105
6.9.1	Extraction des données	106
6.9.2	Transformation des données	106
6.9.3	Chargement des données	108

TABLE DES MATIÈRES

6.10 Choix des graphiques	110
6.11 Publication du tableau de bord	113
6.11.1 Présentation des tableaux de bord dans l'application web	113
6.11.2 Présentation des tableaux de bord dans l'application mobile	113
Conclusion	113
Conclusion générale	114
Webographie	115

Table des figures

1.1	Logo de l'entreprise « DIGITAL IDENTITY »	17
1.2	Organigramme de Digital identity	18
1.3	Les activités de Digital Identity	18
1.4	Mode de fonctionnement de la méthodologie Scrum	21
1.5	UML	23
1.6	Java	23
1.7	Python	23
1.8	MySQL	24
1.9	Firebase	24
1.10	ReactJS	24
1.11	Node.js	24
1.12	Flask	25
1.13	REST API	25
1.14	Figma	26
1.15	Android Studio	26
1.16	Overleaf	26
1.17	Spyder	27
1.18	Jira	27
1.19	Slack	27
1.20	Visme	27
1.21	Material UI	28
1.22	Postman	28
1.23	Instagantt	28
1.24	Adobe Photoshop	28
1.25	Model–view–viewmodel	29
1.26	Architecture physique	29
1.27	Création du backlog avec Jira	30
1.28	Dépôt GitHub	31
2.1	Diagramme des acteurs	33
2.2	Diagramme de cas d'utilisation global	36

TABLE DES FIGURES

2.3	Ecran de chargement - Web	40
2.4	Ecran de chargement - Mobile	40
2.5	Interface d'authentification - Web	40
2.6	Interface d'authentification - Mobile	40
2.7	Interface de profil- Web	41
2.8	Interface de profil - Mobile	41
2.9	Interface de gestion des utilisateurs	41
2.10	Interface de création d'un nouvel utilisateur	42
2.11	map - Web	42
2.12	map - Mobile	42
2.13	la palette de couleurs	43
2.14	logo	43
2.15	Flux utilisateur : Définition des éléments	44
2.16	Vue globale sur les flux d'utilisateur	44
2.17	Organigramme des interfaces du l'administrateur	45
2.18	Organigramme des interfaces du chauffeur	46
2.19	Organigramme des interfaces du mécanicien	47
2.20	Organigramme des interfaces du chef d'équipe	48
2.21	Diagramme de Gantt	49
3.1	Diagramme de cas d'utilisation global du sprint 1	51
3.2	Diagramme de séquence système du cas d'utilisation « S'authentifier »	53
3.3	Analyse de cas d'utilisation « Gérer les utilisateurs »	54
3.4	Diagramme de séquence système du cas d'utilisation «Ajouter utilisateur»	55
3.5	Diagramme de classes du sprint 1	56
3.6	Interface d'authentification Web	57
3.7	Echec de connexion	57
3.8	Interface d'authentification Mobile	58
3.9	Echec de connexion	58
3.10	Interface de profil - Web	58
3.11	Interface de profil - Mobile	58
3.12	Interface de gestion des utilisateurs	59
3.13	Interface de création d'un nouvel utilisateur	60
3.14	Burdown Chart du Sprint 1	60
3.15	Task board du Sprint 1	61
3.16	Code souce du test unitaire du cas d'utilisation «S'authentifier »	61
3.17	Réultat du premier test	62
3.18	Réultat du deuxième test	62
4.1	Diagramme de cas d'utilisation global du sprint 2	64

TABLE DES FIGURES

4.2 Raffinement de cas d'utilisation « Gérer les véhicules »	64
4.3 Raffinement de cas d'utilisation « Ajout des véhicules »	66
4.4 Raffinement de cas d'utilisation « Créer les taches »	67
4.5 Diagramme de séquence système du cas d'utilisation «Créer les tâches pour les chauffeurs»	70
4.6 Analyse de cas d'utilisation « Consulter les taches pour les chauffeurs »	71
4.7 Diagramme d'activité du cas d'utilisation « Consulter les taches des chauffeurs »	72
4.8 Analyse de cas d'utilisation « Consulter les taches pour les mécaniciens »	73
4.9 Diagramme de classes du sprint 2	74
4.10 interface de création des taches	75
4.11 Création des taches pour les chauffeurs	75
4.12 Création des taches pour les mécaniciens	76
4.13 interface de consultation du checklist pour les chauffeurs	76
4.14 interface de consultation des taches pour les mécaniciens	77
4.15 interface de consultation des taches pour les mécaniciens	77
4.16 Burdown Chart du Sprint 2	78
4.17 Task board du Sprint 2	78
4.18 Code souce du test unitaire du cas d'utilisation « Crée taches »	79
4.19 Resultat du test de réussite	79
4.20 Resultat du test d'échec	80
5.1 Diagramme de cas d'utilisation global du sprint 3	82
5.2 Diagramme d'activité du cas d'utilisation « Consulter Notifications »	85
5.3 Raffinement du cas d'utilisation « Générer des rapports »	86
5.4 Diagramme de séquence système du cas d'utilisation « Générer des rapports »	87
5.5 Diagramme de classes du sprint 3	88
5.6 Suivi en Temps Réel par Géolocalisation	89
5.7 Vue du trajet	89
5.8 Carte de voyage	90
5.9 Interface de Consultation des notifications - web »	90
5.10 Interface de Consultation des notifications - Mobile »	90
5.11 Génération de Rapport - Interface Initiale	91
5.12 Génération de Rapport - Détails Entrés	91
5.13 Génération de Rapport - Rapport Sauvegardé	91
5.14 Burdown Chart du Sprint 3	92
5.15 Task board du Sprint 3	92
5.16 Code souce du test unitaire du cas d'utilisation « Générer des rapports »	93
5.17 Resultat du test de réussite	93
5.18 Resultat du test d'échec	94

TABLE DES FIGURES

6.1	Dimension Users	98
6.2	Dimension Time	99
6.3	Dimension véhicules	99
6.4	Dimension task	99
6.5	Schéma de constellation	100
6.6	Schéma en étoile : Chauffeur	101
6.7	Schéma en étoile : Véhicules	102
6.8	Schéma en étoile : Mécanicien	103
6.9	Schéma en étoile : Utilisateurs	104
6.10	Schéma du flux de travail ETL	105
6.11	Code Source : Extraction des données	106
6.12	Transformation de la table de dimension 'users'	107
6.13	Transformation de la table de dimension 'users'	107
6.14	Chargement des Données dans la Table DimUser	108
6.15	Chargement des Données dans la Table FactDriverTasks :	109
6.16	Nivo Chart	110
6.17	Graphique en barres	110
6.18	Graphique linéaire	111
6.19	Graphique en secteurs	111
6.20	Heatmap	112

Liste des tableaux

1.1	Description des machines de développement utilisées	25
2.1	Backlog du produit	39
3.1	Backlog du sprint 1	50
3.2	Description textuelle du cas d'utilisation “S'authentifier”	52
3.3	Description textuelle du cas d'utilisation “Ajouter utilisateur”	55
4.1	Backlog du sprint 2	63
4.2	Description textuelle du cas d'utilisation “Ajouter véhicules ”	66
4.3	Description textuelle du cas d'utilisation “Créer des Tâches pour les Mécaniciens ”	68
4.4	Description textuelle du cas d'utilisation “Créer des Tâches pour les chauffeurs ”	69
4.5	Description textuelle du cas d'utilisation “Consulter les taches des chauffeurs” . .	72
4.6	Description textuelle du cas d'utilisation “Consulter les tâches des mécaniciens” .	74
5.1	Backlog du sprint 3	81
5.2	Description textuelle du cas d'utilisation “Suivi en Temps Réel par Géolocalisation”	84
5.3	Description textuelle du cas d'utilisation “ Planifier les déplacements ”	84
5.4	Description textuelle du cas d'utilisation “ Consulter Notifications ”	85
5.5	Description textuelle du cas d'utilisation “ Générer des rapports ”	87
6.1	Backlog du sprint 4	95
6.2	Indicateurs de performance	97

Liste des Acronymes

GMAO : Gestion de la Maintenance Assistée par Ordinateur

API : Interface de Programmation Applicative

REST : Representational State Transfer

UML : Unified Modeling Language

MVVM : Modèle-Vue-VueModèle

HTTP : Hypertext Transfer Protocol

SQL : Structured Query Language

KPI : Key Performance Indicator

ETL : Extraction, Transformation, and Loading

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. En effet, dans un environnement économique de plus en plus concurrentiel, les entreprises doivent non seulement maximiser l'utilisation de leurs ressources, mais également minimiser les coûts liés à leur maintenance et leur exploitation. 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. Ces investissements nécessitent une gestion rigoureuse et proactive pour assurer leur durabilité, leur efficacité et leur rentabilité à long terme.

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. L'objectif est de fournir un outil qui non seulement facilite la gestion quotidienne, mais aussi anticipe les besoins futurs grâce à des fonctionnalités avancées de suivi et d'analyse des données.

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 détaillerons comment notre solution permet une gestion proactive grâce à des notifications automatisées pour les maintenances, des rapports détaillés sur l'utilisation et l'état des actifs, et des tableaux de bord personnalisables pour une vision claire et en temps réel des performances.

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. En réduisant les temps d'arrêt des équipements et en prolongeant leur durée de vie, notre application vise à générer des économies significatives et à améliorer la productivité globale des parcs d'entreprise. De plus, en fournissant des données pré-

cises et exploitables, elle permet aux gestionnaires de prendre des décisions éclairées, d'optimiser les processus et de mieux planifier les investissements futurs.

Enfin, ce rapport abordera les perspectives d'évolution de notre application, avec des plans pour intégrer des améliorations continues basées sur les retours d'expérience des utilisateurs et les besoins émergents des entreprises. Notre vision est de créer une solution qui évolue avec les besoins des entreprises, offrant une flexibilité et une scalabilité pour répondre aux défis de demain.

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 »

1.1.1 Organigramme de l'entreprise

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

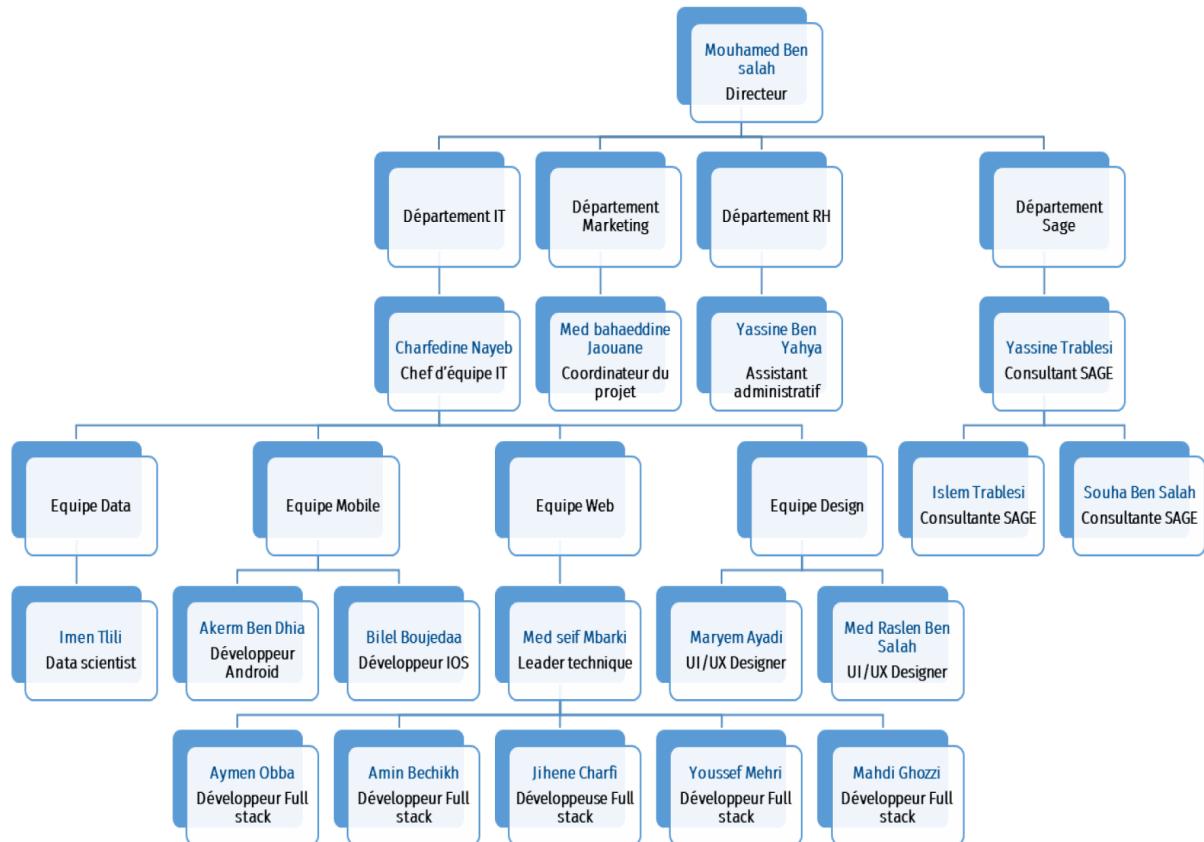


FIGURE 1.2 – Organigramme de Digital identity

1.1.2 Les activités de Digital Identity



FIGURE 1.3 – Les activités de Digital Identity

Digital Identity se spécialise dans plusieurs domaines clés :

- **Audit et Conseils** : Spécialisée dans les services d'audit et de conseil de haute qualité, Digital Identity aide les entreprises à relever des défis complexes et à atteindre leurs objectifs grâce à une expertise approfondie et une approche personnalisée. Ses recommandations stratégiques optimisent les performances et favorisent une croissance durable.
- **Implémentation de Systèmes d'Informations** : En tant que fournisseur de confiance, Digital Identity facilite l'adoption de solutions logicielles robustes, garantissant une intégration fluide, une mise en œuvre efficace et une formation complète pour rationaliser les opérations des entreprises et optimiser leur productivité.
- **Développement d'Applications** : Digital Identity excelle dans le développement d'applications de bureau et mobiles sur mesure, mettant l'accent sur la convivialité et l'utilisation des technologies avancées. Ses solutions personnalisées permettent aux organisations d'optimiser leurs flux de travail de manière transparente.
- **Infrastructure des Technologies de l'Information** : Elle propose des services complets pour assurer des environnements informatiques fiables, sécurisés et évolutifs, en se spécialisant dans l'architecture réseau, l'intégration de systèmes et la cybersécurité pour garantir des performances optimales.

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

Dans le contexte actuel, la gestion des parcs de véhicules d'entreprise est confrontée à plusieurs défis critiques. La diversité des véhicules et des itinéraires, combinée aux exigences opérationnelles spécifiques de chaque entreprise, rend la gestion des flottes complexe et nécessite des solutions personnalisées. Les entreprises doivent également faire face à des pressions technologiques, telles que l'intégration de nouvelles technologies de suivi et de maintenance prédictive, ainsi qu'à des pratiques durables.

Pour répondre à ces défis, une plateforme intégrée capable de fournir une surveillance en temps réel et des analyses prédictives est essentielle. Une telle plateforme doit permettre une optimisation avancée des ressources, en minimisant les coûts et en maximisant l'efficacité opérationnelle. De plus, elle doit intégrer des fonctionnalités de sécurité et de conformité réglementaire pour assurer des opérations sécurisées et conformes aux exigences légales.

1.2.2 Solution proposée

Notre solution propose une plateforme de gestion de flotte de véhicules d'entreprise qui intègre des technologies avancées pour offrir des fonctionnalités complètes et personnalisables. Cette plateforme permet de surveiller les performances de la flotte en temps réel, de prédire les besoins de maintenance et d'optimiser les itinéraires et l'utilisation des véhicules. Elle offre une gestion centralisée et simplifiée des actifs de transport, réduisant ainsi les coûts opérationnels et augmentant l'efficacité.

La dimension Business Intelligence (BI) de la plateforme permet d'analyser les données de manière approfondie, fournissant aux gestionnaires des informations stratégiques essentielles pour la prise de décision. Les tableaux de bord interactifs offrent une vue d'ensemble en temps réel des performances de la flotte, permettant aux gestionnaires de suivre des indicateurs clés tels que le taux d'utilisation des véhicules, les coûts de maintenance et les temps d'immobilisation. Les rapports personnalisables fournissent des insights détaillés sur les tendances et les opportunités d'optimisation, facilitant ainsi une gestion proactive et informée.

En intégrant ces fonctionnalités avec une approche centrée sur les besoins des entreprises, notre solution vise à optimiser la gestion des flottes de véhicules, réduisant les coûts et améliorant l'efficacité opérationnelle. Elle permet aux entreprises de rester compétitives en offrant une gestion stratégique et évolutive de leurs actifs de transport, capable de s'adapter aux défis futurs grâce à des technologies émergentes telles que l'intelligence artificielle et l'Internet des objets (IoT).

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 approches agile se caractérisent par une méthode itérative et incrémentale, une focalisation sur les besoins des utilisateurs, une flexibilité pour s'adapter aux changements, une collaboration étroite avec toutes les parties prenantes du projet, et une livraison régulière de fonctionnalités à forte valeur ajoutée. Chaque méthode agile présente ses propres pratiques, rôles, cérémonies et outils, tout en partageant les mêmes valeurs et principes fondamentaux de l'agilité.

Dans notre contexte, nous avons opté pour le framework Scrum pour le développement de notre application, car il nous semble le mieux adapté à notre travail. Scrum offre une structure agile et flexible qui nous permet de diviser notre projet en sprints. Cette approche itérative nous aide à gérer efficacement notre travail en nous concentrant sur des objectifs clairs 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.

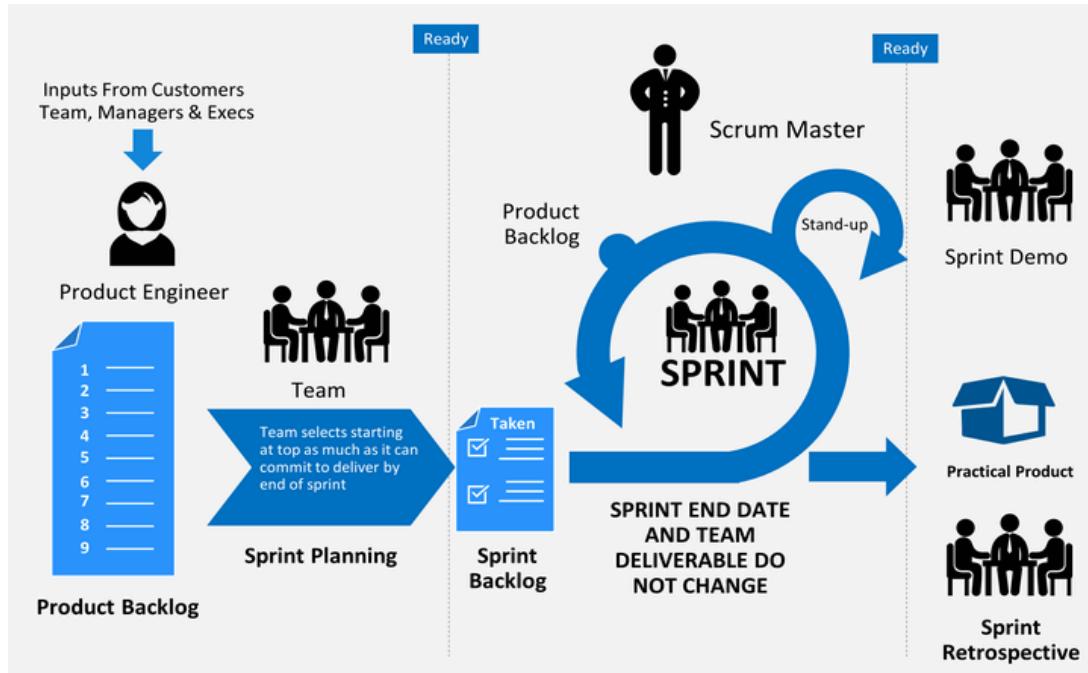


FIGURE 1.4 – Mode de fonctionnement de la méthodologie Scrum

?

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 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** : Equipe 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.»

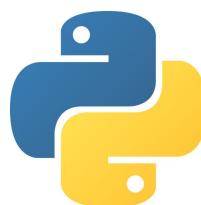


FIGURE 1.7 – Python

- **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

- **Firebase** : « Nous avons utilisé Firebase pour implémenter la fonctionnalité en temps réel de notre application. Cela permet de synchroniser instantanément les données, assurant un suivi en temps réel des véhicules.»



FIGURE 1.9 – Firebase

- **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 1.10 – 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.»

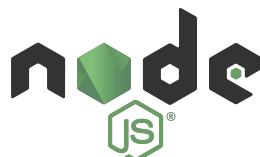


FIGURE 1.11 – Node.js

- **Flask** : « Nous avons choisi d'utiliser Flask pour créer le serveur. Flask est un micro-framework web léger et flexible pour Python qui facilite le développement. »



FIGURE 1.12 – Flask

- **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. »

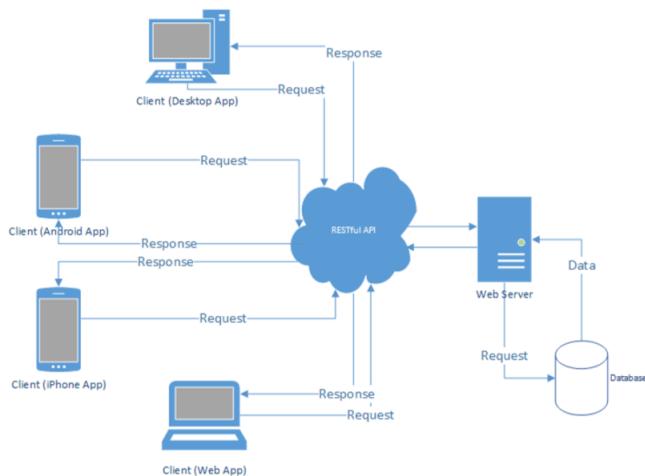


FIGURE 1.13 – REST API

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 11ème	i5 11ème
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 1.14 – 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 1.15 – Android Studio

- **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 1.16 – 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 1.17 – 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 1.18 – Jira

- **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 1.19 – Slack

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



FIGURE 1.20 – Visme

- **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 1.21 – Material UI

- **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 1.22 – Postman

- **Instagantt** : C'est un outil en ligne intuitif pour la gestion de projets qui permet de créer des diagrammes de Gantt de manière simple et visuelle.



FIGURE 1.23 – Instagantt

- **Adobe Photoshop** : Nous avons utilisé Photoshop pour la création du logo de notre application. Cet outil nous a permis de concevoir un logo professionnel et attrayant, reflétant l'identité visuelle de notre projet.



FIGURE 1.24 – Adobe Photoshop

1.5 Architecture du Système

1.5.1 Architecture logique front-end

En architecture logicielle, MVVM (Model View ViewModel) est un Design pattern (Modèle de conception) visant à séparer la logique de présentation d'une application en 3 couches :

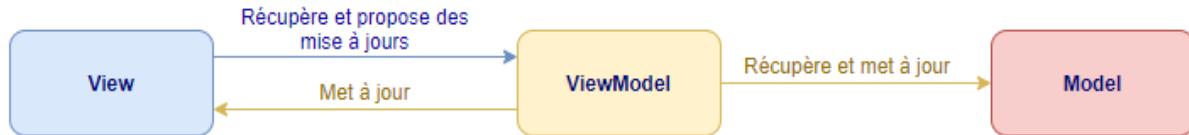


FIGURE 1.25 – Model–view–viewmodel

- **Model (Modèle)** : Le modèle représente la logique métier de l'application et contient les données provenant de bases de données ou d'API externes, optimisées pour la persistance ou la transmission. Il est indépendant de l'interface utilisateur et ne prend pas en compte la manière dont les données seront affichées.
- **View (Vue)** : La vue est ce que l'utilisateur voit à l'écran. Elle décrit l'interface graphique et fait le lien entre les actions de l'utilisateur et le modèle de vue, définissant l'agencement des composants graphiques et gérant les liaisons de données connectant ces composants aux données du modèle de vue.
- **ViewModel (Modèle de Vue)** : Le modèle de vue agit comme un médiateur entre le modèle et la vue. Il transforme les données du modèle en formats facilement affichables et manipulables par la vue, permettant à la vue de s'adapter indépendamment des modifications apportées aux données métier et au modèle.

1.5.2 Architecture physique

La figure 1.26 montre l'architecture physique de l'application, incluant les technologies pour le développement web et mobile, et leur connexion à une base de données MySQL.

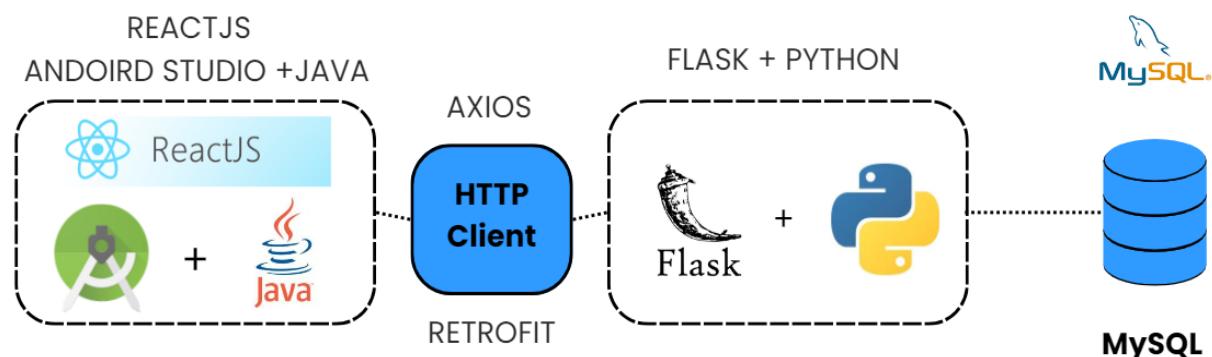


FIGURE 1.26 – Architecture physique

Fonctionnement :

- 1- L'utilisateur interagit avec l'interface utilisateur web développée avec ReactJS ou l'application mobile Android.
- 2- Les actions de l'utilisateur déclenchent des requêtes HTTP envoyées via Axios pour le web ou Retrofit pour le mobile.
- 3- Les requêtes sont reçues par le backend Flask, qui traite les demandes en exécutant du code Python.
- 4- Le backend Flask interagit avec la base de données MySQL pour récupérer ou stocker des données.
- 5- Les réponses sont renvoyées au client HTTP, qui les transmet à l'interface utilisateur pour mise à jour.

Cette architecture permet une séparation claire des responsabilités entre le frontend et le backend, assurant ainsi une meilleure maintenabilité et évolutivité de l'application.

1.6 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 1.27 – 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.

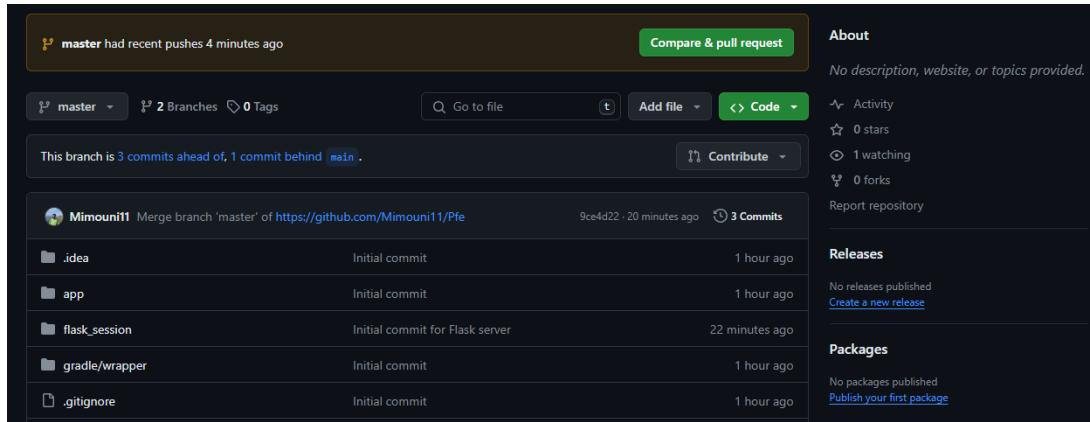


FIGURE 1.28 – Dépôt GitHub

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

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 lequel 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 les déplacements :** 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.

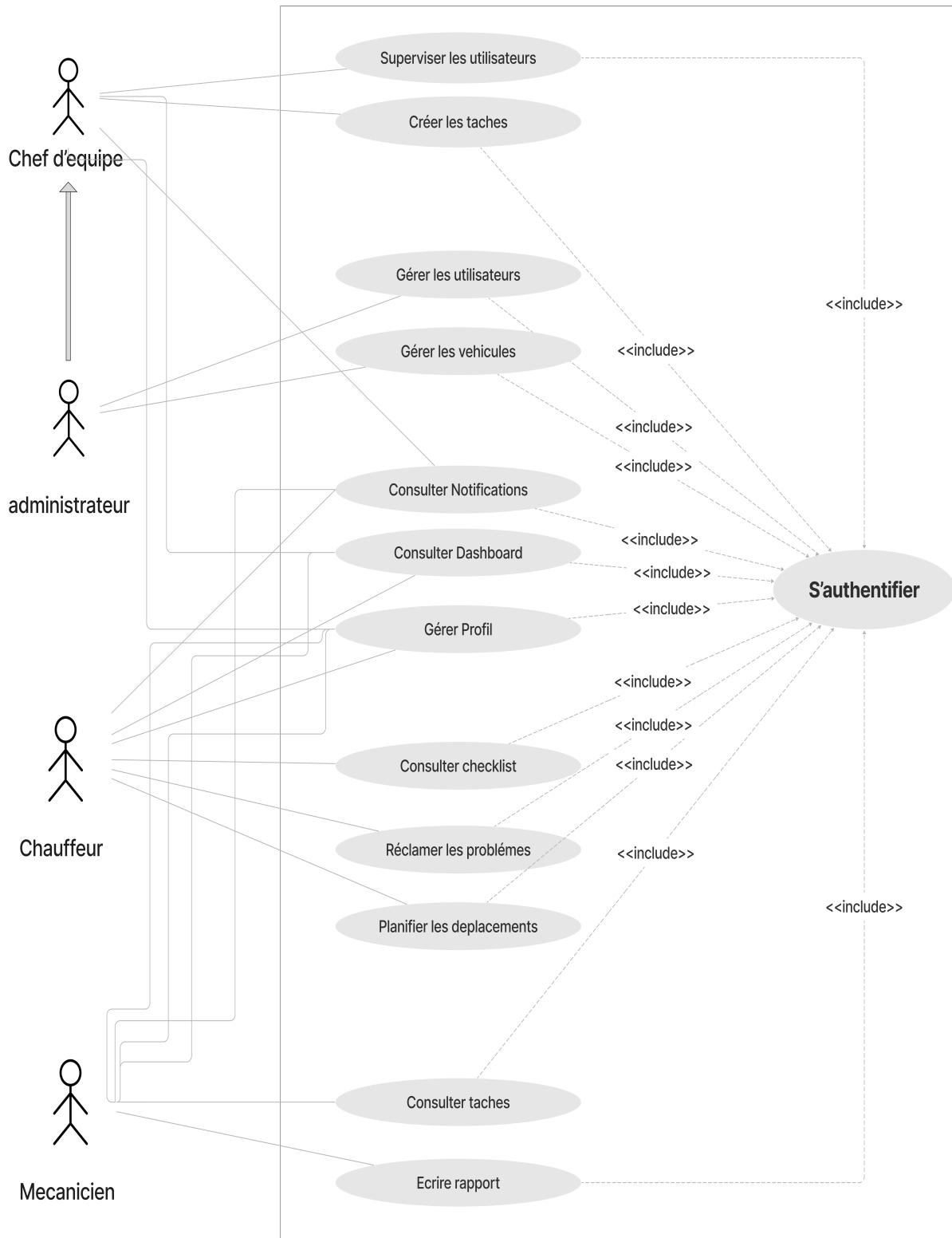


FIGURE 2.2 – Diagramme de cas d'utilisation global

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.

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

US12	En tant que chauffeur, je veux planifier les déplacements pour optimiser les itinéraires et les arrêts.	Faible	Faible
US13	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.	Moeynne	Moyen
US14	En tant que Mécanicien je veux Générer des rapports pour que je puisse documenter les détails de chaque intervention.	Moeynne	Moyen
US15	En tant qu'administrateur, je veux avoir accès à un tableau de bord pour que je puisse superviser l'ensemble des opérations.	Elevée	Elevé
US16	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	Elevé
US17	En tant que Chauffeur je veux Consulter un tableau de bord pour que je puisse avoir une vue claire de mes missions actuelles	Elevée	Elevé
US18	En tant que Mécanicien, je veux consulter un tableau de bord pour avoir une vue d'ensemble des opérations de maintenance.	Elevée	Elevé

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 ecrans de chargement dans l'application.

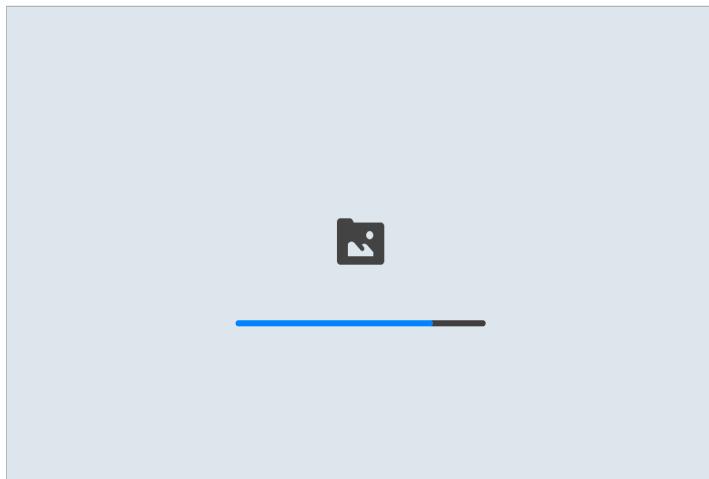


FIGURE 2.3 – Ecran de chargement - Web

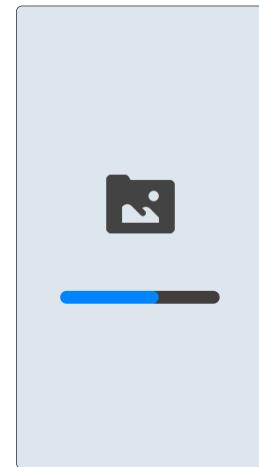


FIGURE 2.4 – Ecran de chargement - Mobile

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

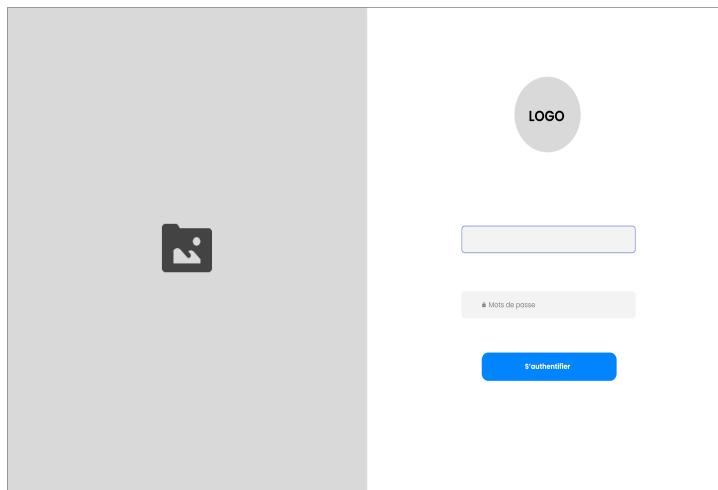


FIGURE 2.5 – Interface d'authentification - Web

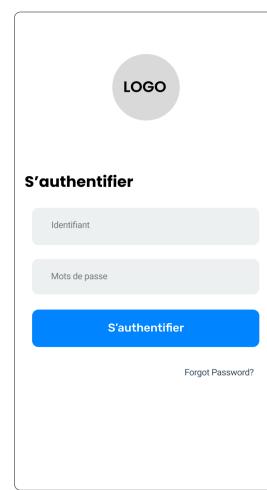


FIGURE 2.6 – Interface d'authentification - Mobile

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

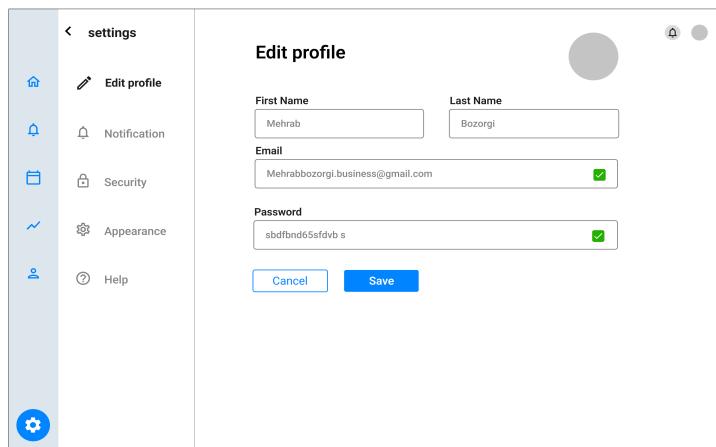


FIGURE 2.7 – Interface de profil- Web



FIGURE 2.8 – Interface de profil - Mobile

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

This screenshot shows a user management interface. On the left is a sidebar with 'LOGO', 'ADMIN' (username), and email 'admin@gmail.com'. It also has buttons for 'Dashboard', 'Utilisateurs' (selected), 'Notifications', and 'Stats'. The main area is titled 'Gerer utilisateurs' and shows a table of users. The columns are 'Nombre' (Name), 'Apellido' (Last Name), and 'Estado' (Status). The table contains the following data:

Nombre	Apellido	Estado
Fernando	hamander	Activo
Francisco	Coronado	Inactivo
Alberto	Morales	Activo
Mauricio	Perez	Inactivo
Fernando	Bautista	Activo
Juan	Mendoza	Activo

FIGURE 2.9 – Interface de gestion des utilisateurs

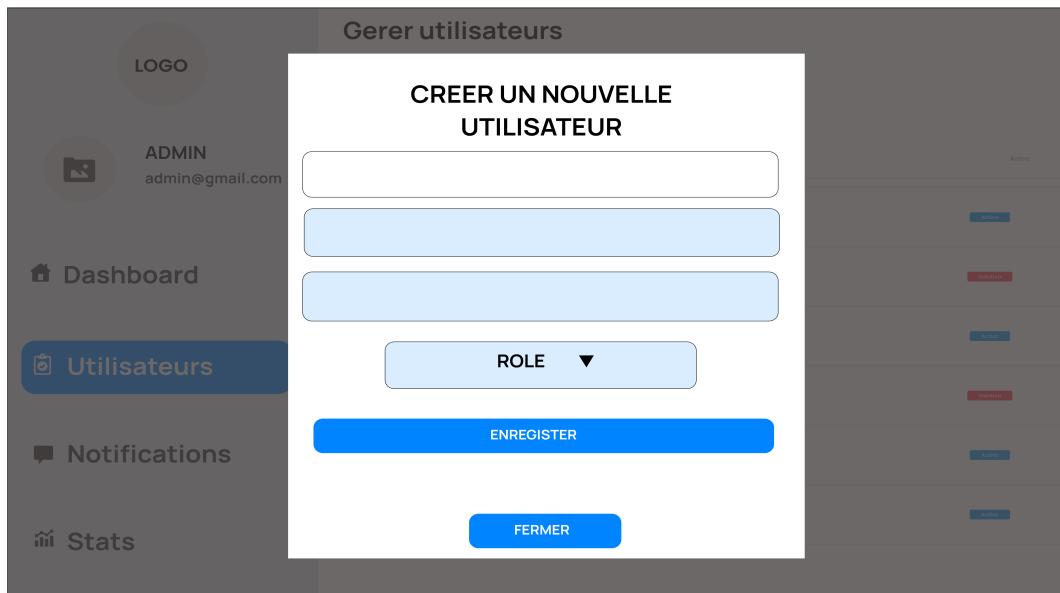


FIGURE 2.10 – Interface de création d'un nouvel utilisateur

Les figures 2.11 et 2.12 illustrent les interfaces du map pour l'admin et le chauffeur

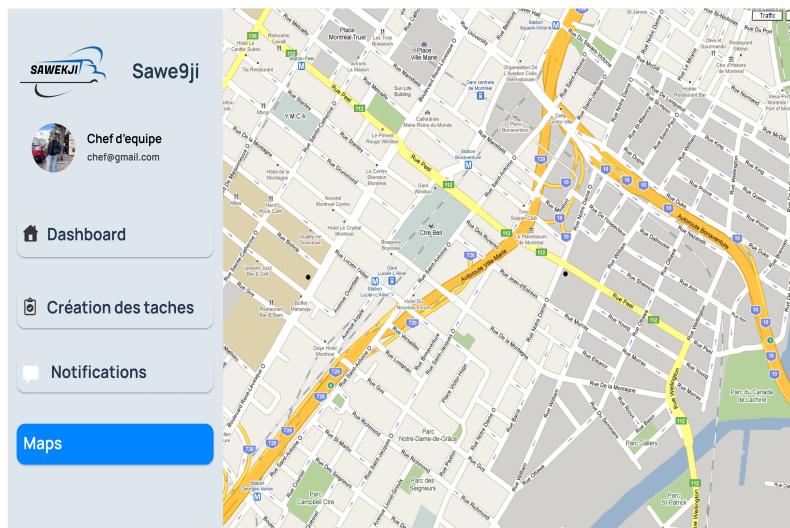


FIGURE 2.11 – map - Web



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) :

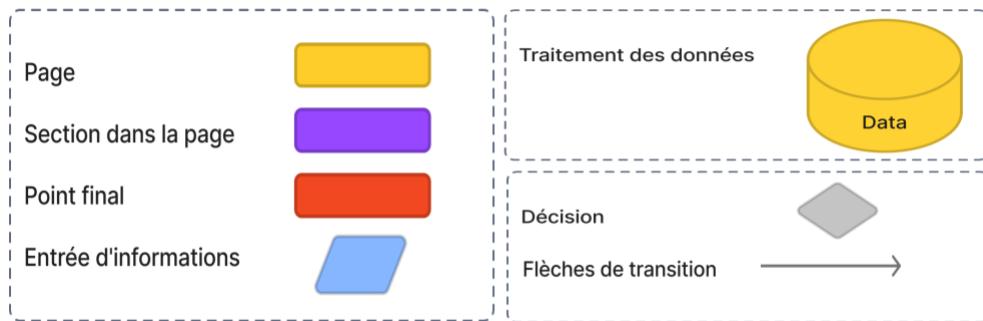


FIGURE 2.15 – Flux utilisateur : Définition des éléments

Vue globale sur les flux d'utilisateurs :

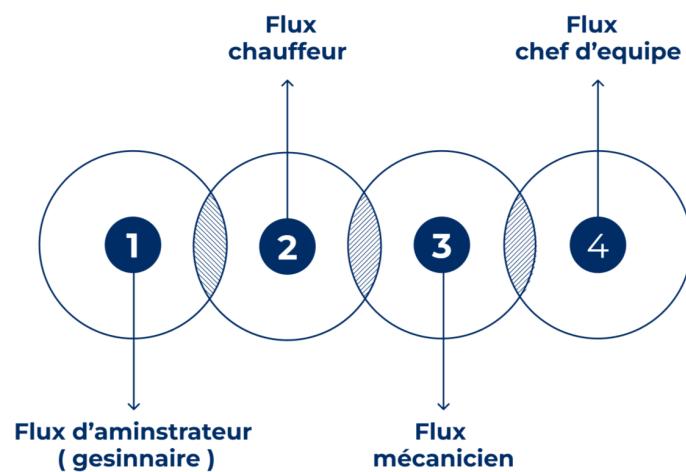


FIGURE 2.16 – Vue globale sur les flux d'utilisateurs

Flux administrateur

Ce diagramme montre les principales fonctionnalités accessibles à un administrateur dans l'interface admin, divisées en quatre sections :

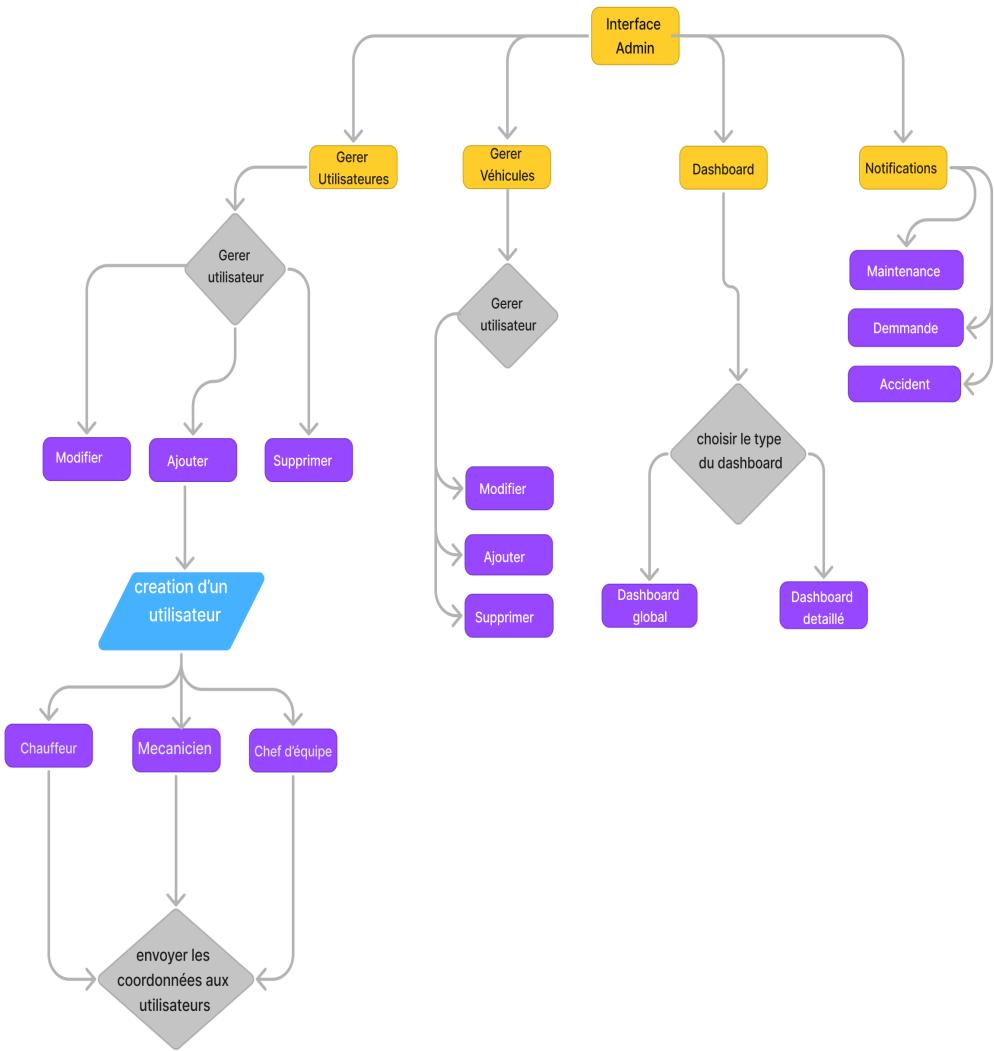


FIGURE 2.17 – Organigramme des interfaces du l'administrateur

Gérer Utilisateurs :

- Modifier, ajouter, ou supprimer des utilisateurs.
- Lors de l'ajout, choisir le type d'utilisateur (Chauffeur, Mécanicien, Chef d'équipe) et envoyer les coordonnées.

Gérer Véhicules :

- Modifier, ajouter, ou supprimer des véhicules.

Dashboard :

- Choisir entre un dashboard global pour une vue d'ensemble et un dashboard détaillé pour des analyses spécifiques.

Notifications :

- Gérer et visualiser des notifications de maintenance, de demandes, et d'accidents.

Flux chauffeur

Ce diagramme montre les principales fonctionnalités accessibles à un chauffeur dans l'application.

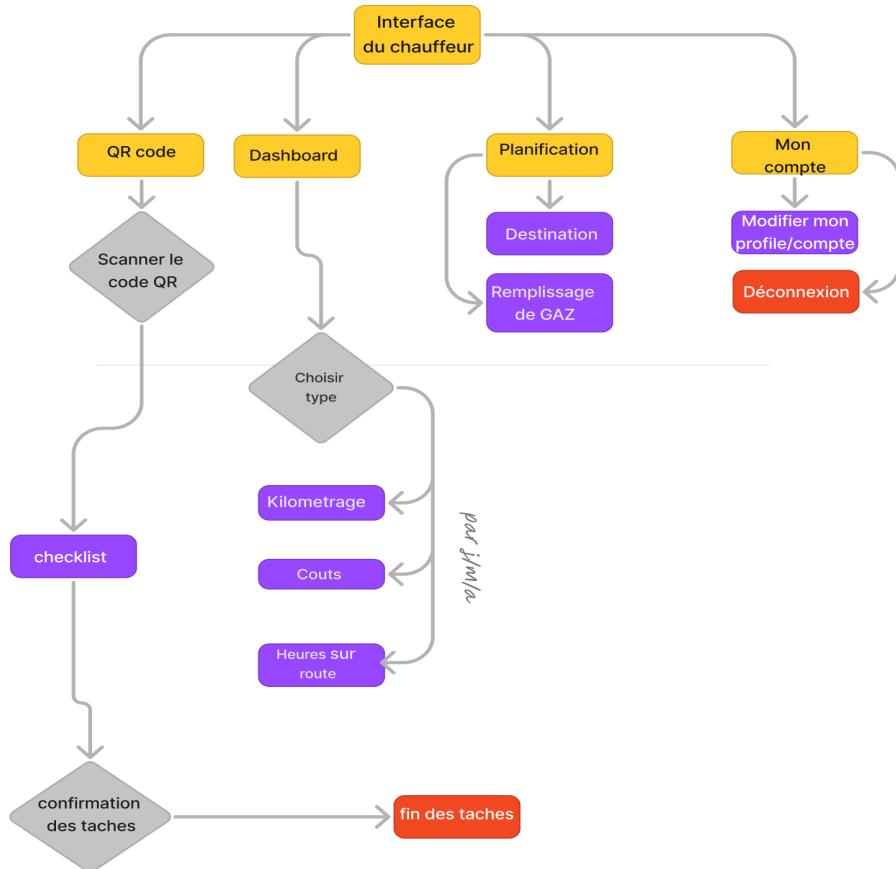


FIGURE 2.18 – Organigramme des interfaces du chauffeur

QR Code :

- Scanner le code QR pour démarrer une tâche.
- Effectuer une checklist des tâches à réaliser.
- Confirmer les tâches effectuées pour terminer la tâche.

Dashboard :

- Choisir entre différents types de données à consulter, telles que :Kilométrage, Heures sur route...

Planification :

- Planifier des destinations et le remplissage de gaz.

Mon Compte :

- Modifier le profil ou le compte.
- Se déconnecter de l'interface.

Flux mécanicien

Ce diagramme montre les principales fonctionnalités accessibles à un mécanicien dans l'interface utilisateur dédiée.

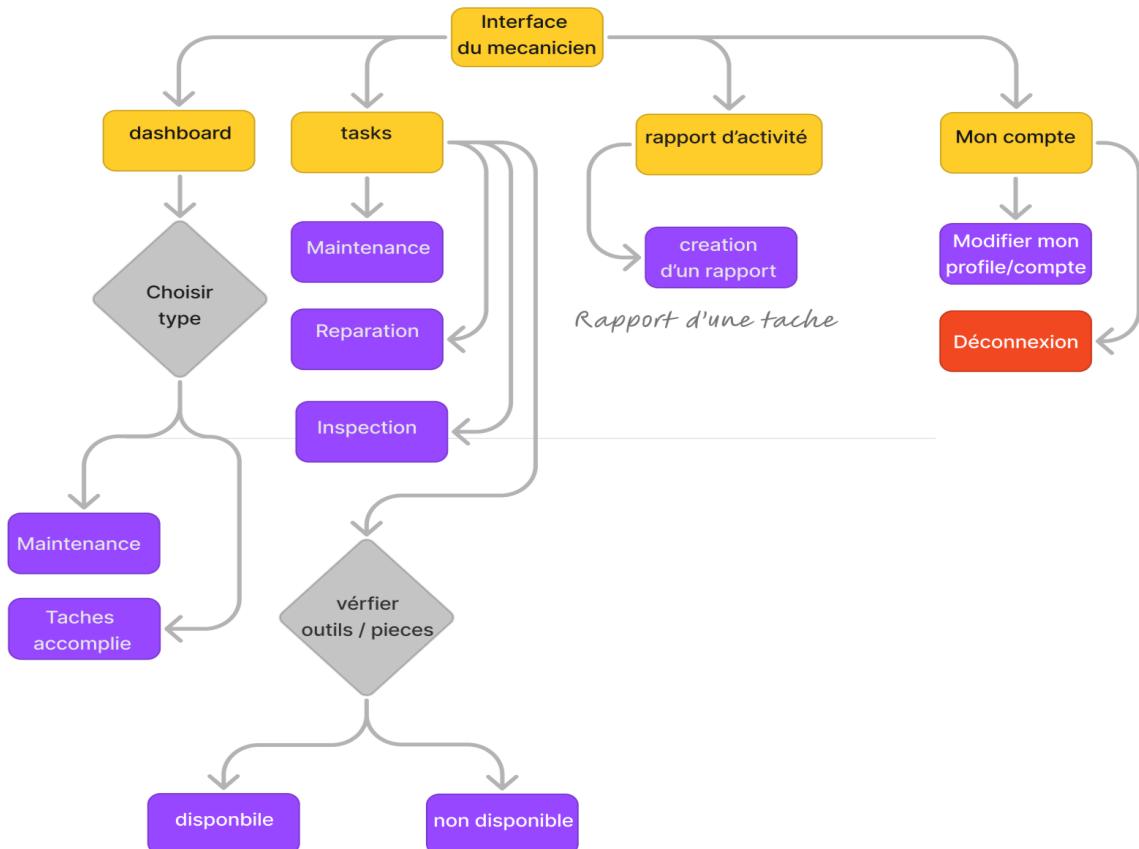


FIGURE 2.19 – Organigramme des interfaces du mécanicien

Dashboard :

-Choisir le type de tâche à consulter.

Les Tâches :

-Choisir entre différentes tâches à accomplir, telles que : Maintenance, Réparation ou Inspection
-Effectuer la maintenance et vérifier les outils ou pièces nécessaires.

Si les outils/pièces sont disponibles, accomplir la tâche.

Si les outils/pièces ne sont pas disponibles, signaler la non-disponibilité.

Rapport d'Activité :

-Créer un rapport d'une tâche effectuée.

Mon Compte :

-Modifier le profil ou le compte.

-Se déconnecter de l'interface.

Flux chef d'équipe

Ce diagramme montre les principales fonctionnalités accessibles à un chef d'équipe dans l'interface utilisateur dédiée.

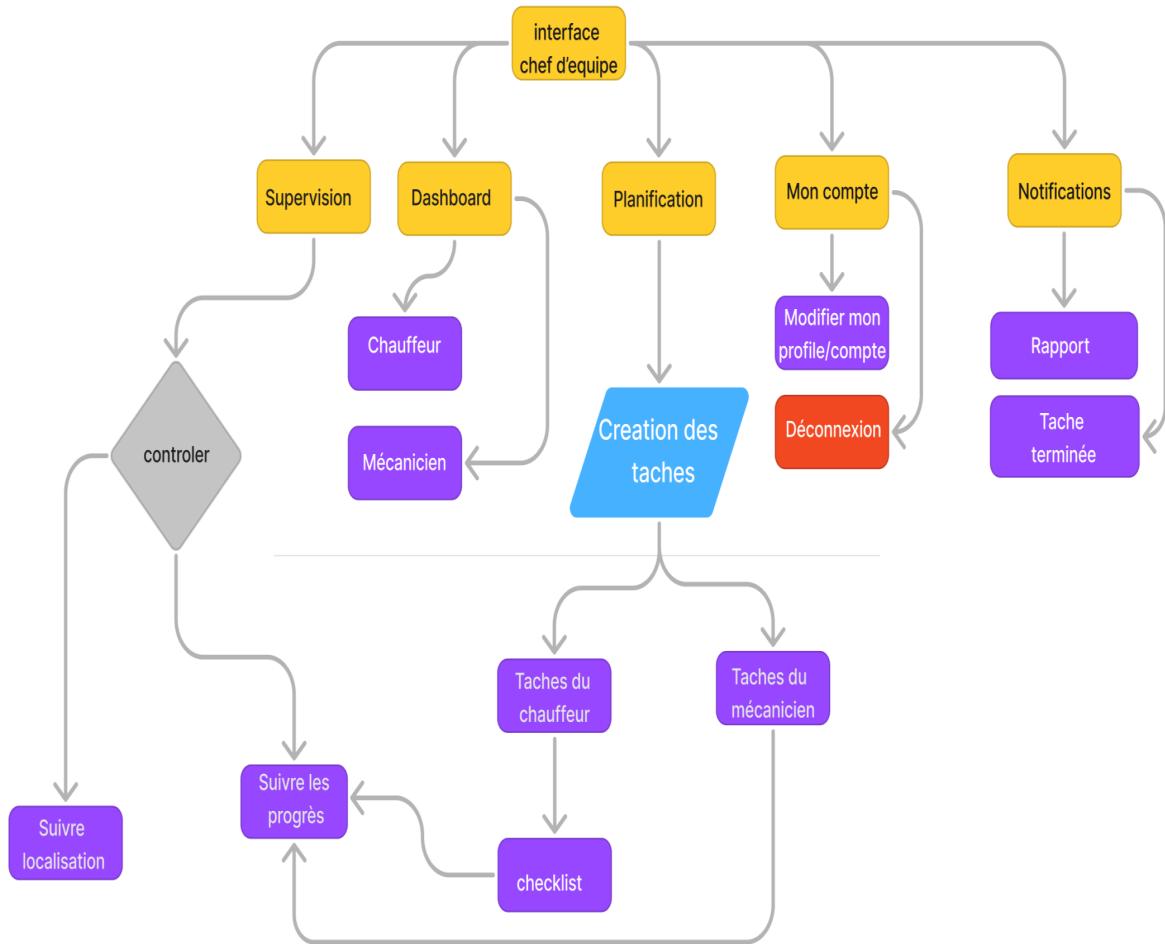


FIGURE 2.20 – Organigramme des interfaces du chef d'équipe

Supervision :

-Contrôler : Suivre la localisation et les progrès des chauffeurs et des mécaniciens.

Dashboard :

-Consulter les données spécifiques des chauffeurs et des mécaniciens.

Planification :

-Création des tâches : Planifier les tâches pour les chauffeurs et les mécaniciens.

Mon Compte :

-Modifier le profil ou le compte.

-Se déconnecter de l'interface.

Notifications :

-Gérer et visualiser les notifications importantes : Rapport / Tâche terminée

2.3.4 Diagramme de Gantt

Par cette figure 2.21 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 artefact 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 Backlog du sprint 1

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
US1	Authentification	En tant qu'admin, chauffeur, chef d'équipe et mécanicien je veux m'authentifier pour que j'accède à mon espace.	3
US2	Déconnexion	En tant qu'admin, chauffeur, chef d'équipe et mécanicien je veux me déconnecter pour que je quitte l'application.	1
US3	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
US4	Gestion des utilisateurs	En tant qu'admin je veux Gérer les utilisateurs pour que je puisse les ajouter, modifier et supprimer .	5
US5	Gestion des rôles	En tant qu'admin, je veux pouvoir gérer les rôles et les autorisations des utilisateurs.	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.

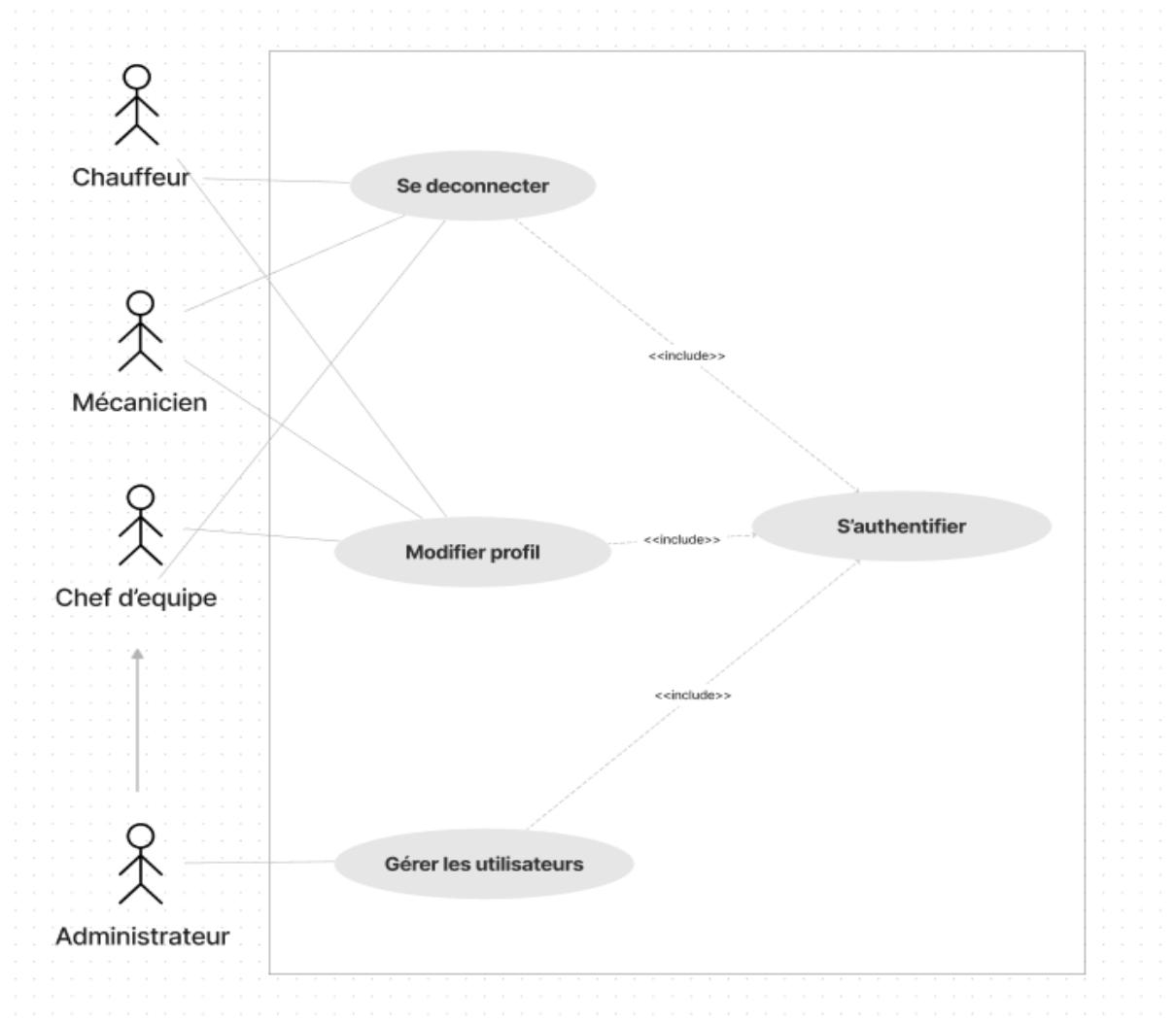


FIGURE 3.1 – Diagramme de cas d'utilisation global du sprint 1

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

Cas d'utilisation	S'authentifier
Acteur	Administrateur, Chef d'équipe, Chauffeur et Mécanicien
Pré-condition	L'utilisateur accède à l'interface d'authentification de l'application.
Post-condition	L'utilisateur est authentifié et accède à son espace sécurisé dans l'application.
Scenario nominal	<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>
Scénario alternatif	<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”

3.2.3 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.

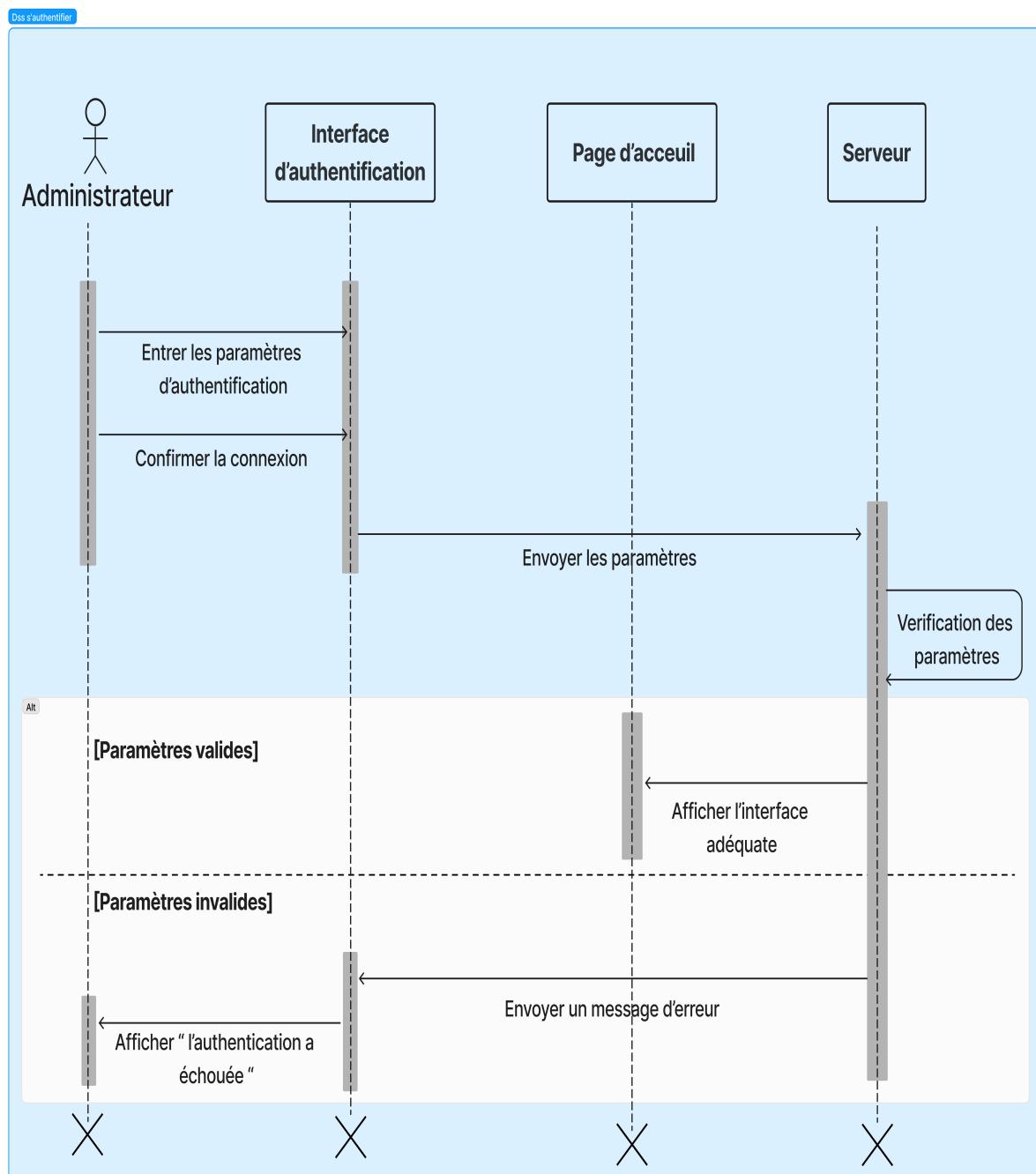


FIGURE 3.2 – Diagramme de séquence système du cas d'utilisation « S'authentifier »

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

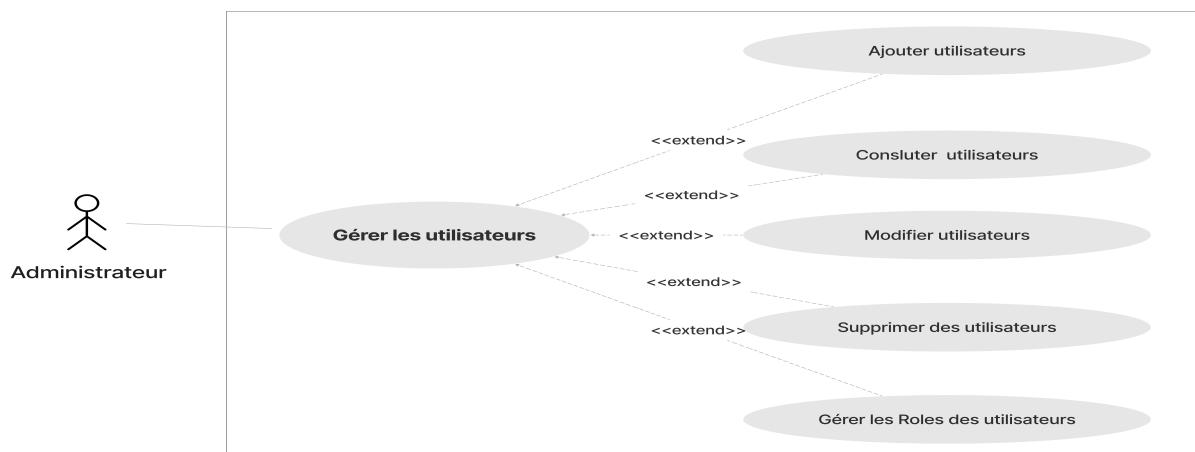


FIGURE 3.3 – Analyse de cas d'utilisation « Gérer les utilisateurs »

3.2.5 Description textuelle du cas d'utilisation “Ajouter utilisateur”

Cas d'utilisation	Ajouter Utilisateur
Acteur	Administrateur
Pré-condition	L'administrateur est authentifié dans le système .
Post-condition	Le nouvel utilisateur est ajouté avec succès dans le système et peut accéder aux fonctionnalités appropriées.
Scénario nominal	<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 système vérifie les informations saisies.</p> <p>4- Le nouvel utilisateur est ajouté à la base de données.</p> <p>5- Le système envoie un e-mail au nouvel utilisateur contenant ses informations de connexion pour accéder au système.</p>
Scénario alternatif	<p>A1 Erreur de saisie des informations :</p> <p>l'échappement A1 démarre au point 3 du scenario 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”

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

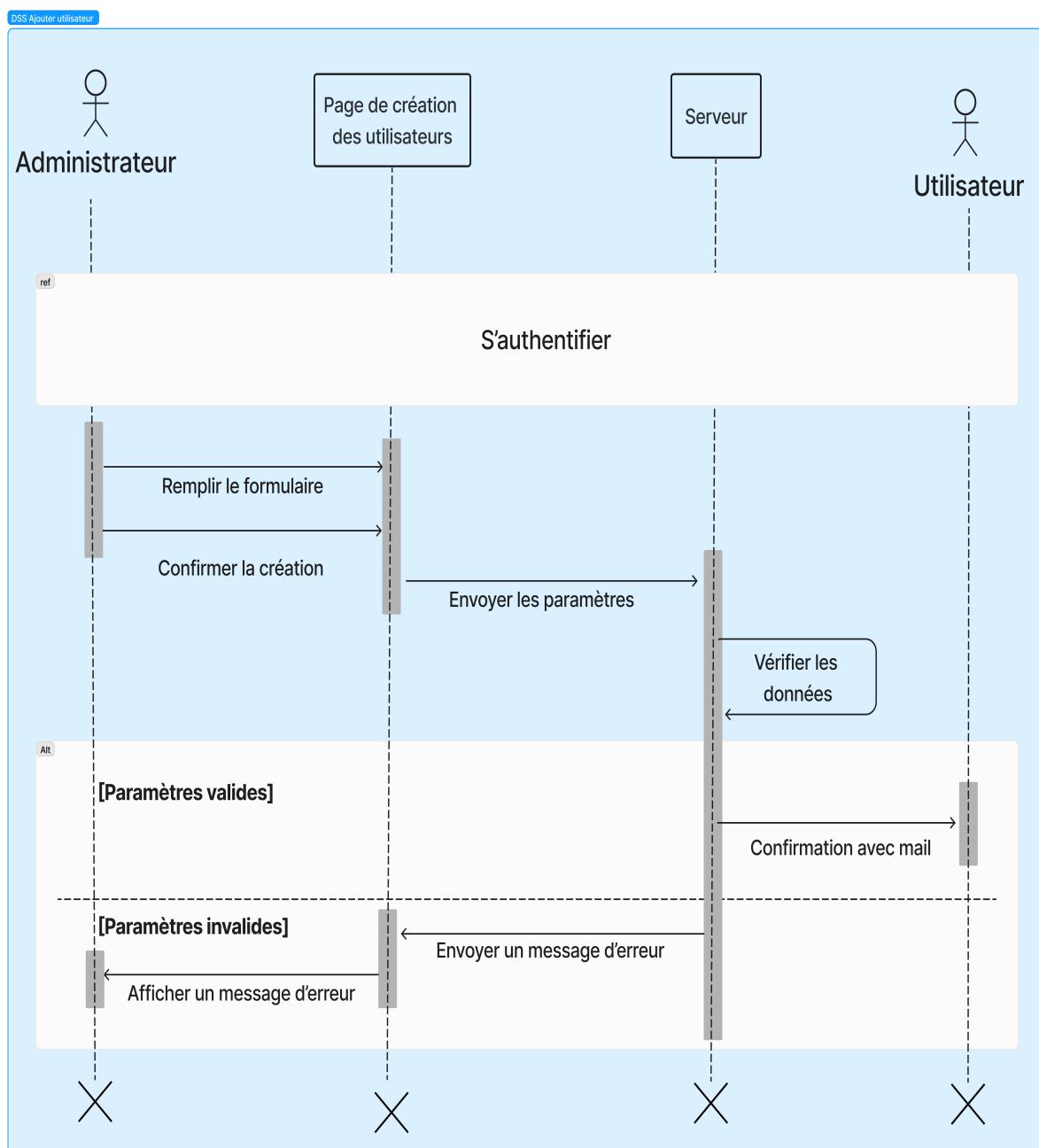


FIGURE 3.4 – Diagramme de séquence système du cas d'utilisation «Ajouter utilisateur»

3.3 Diagramme de classes du sprint 1

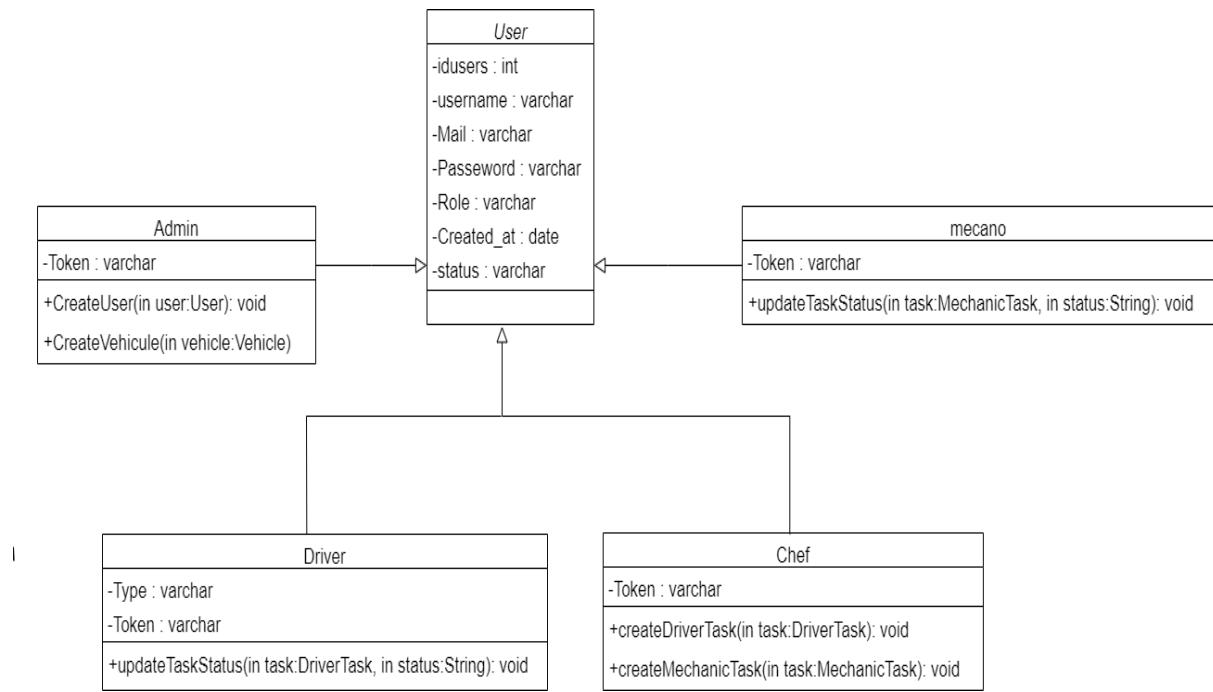


FIGURE 3.5 – Diagramme de classes du sprint 1

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.

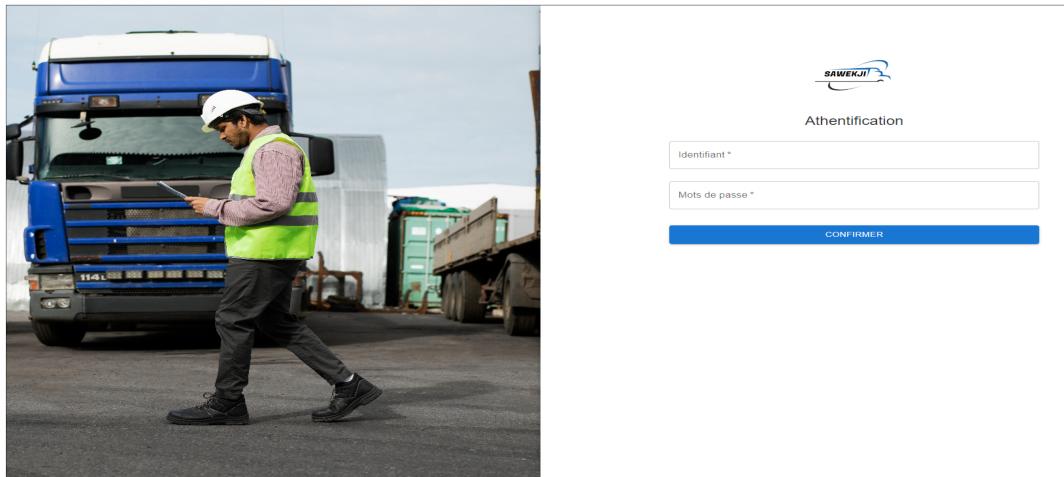


FIGURE 3.6 – Interface d'authentification Web

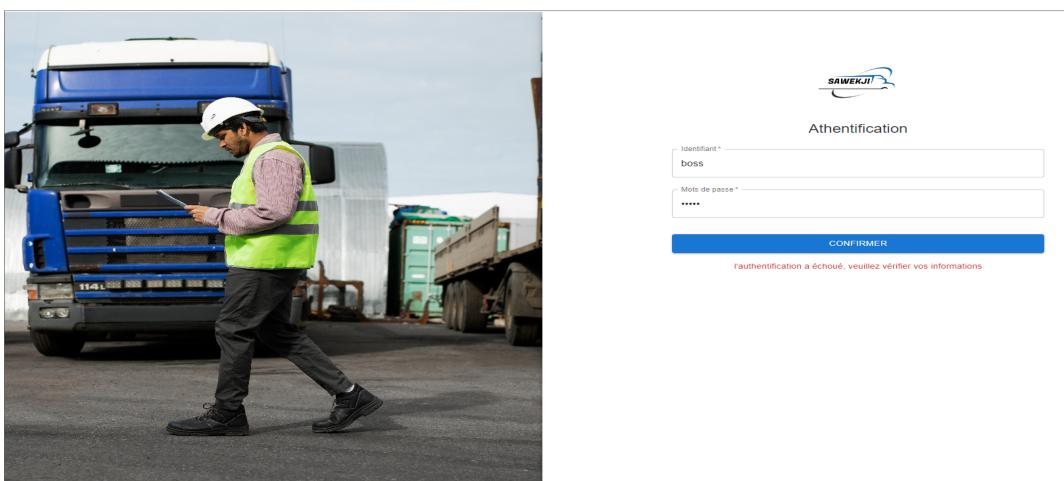


FIGURE 3.7 – Echec de connexion

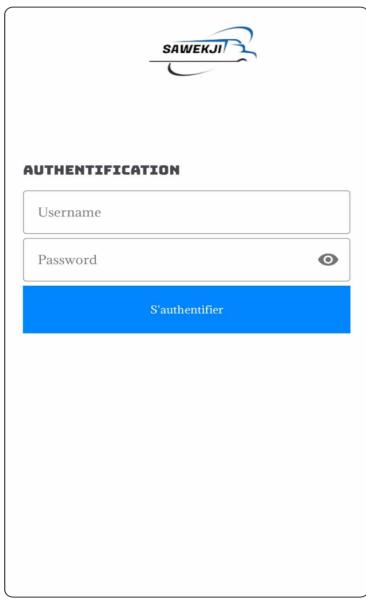


FIGURE 3.8 – Interface d'authentification Mobile

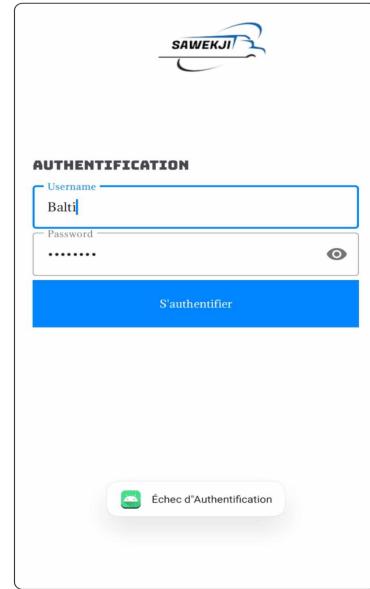


FIGURE 3.9 – Echec de connexion

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.

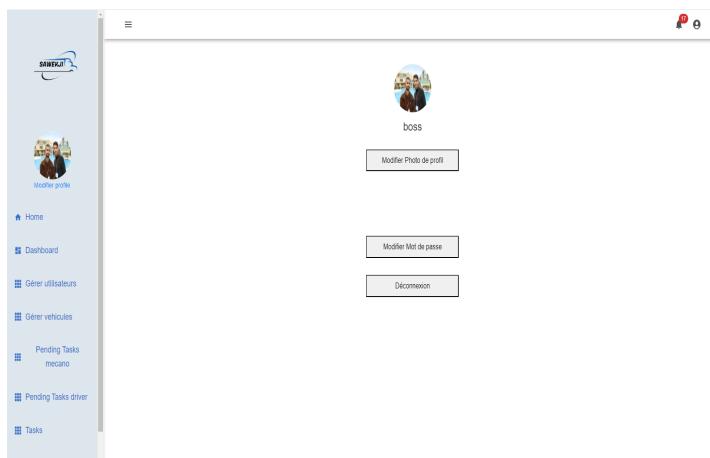


FIGURE 3.10 – Interface de profil - Web

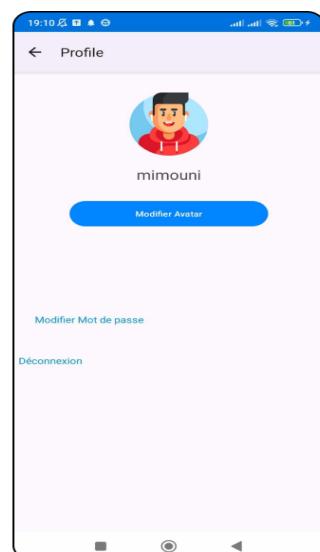
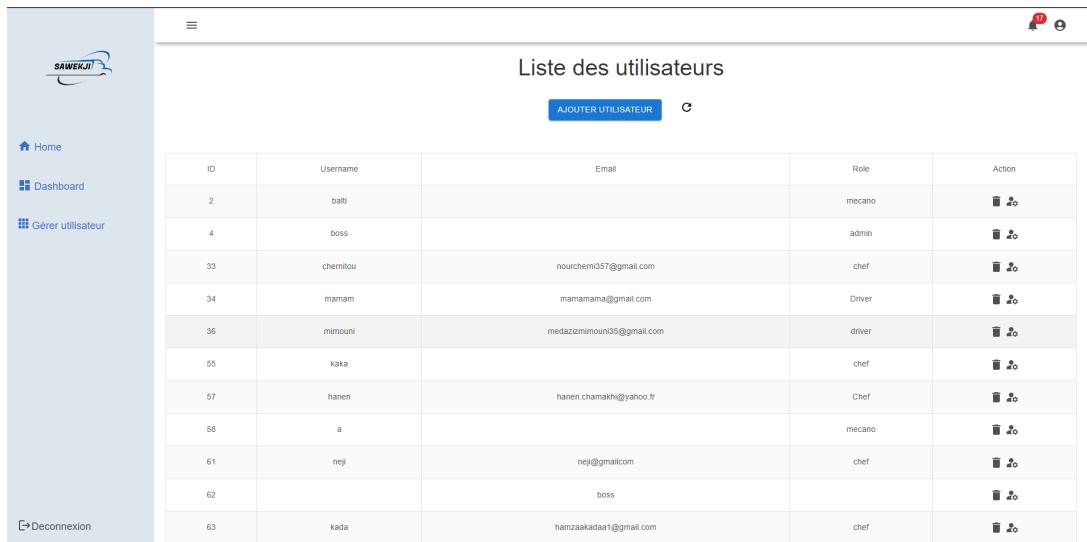


FIGURE 3.11 – Interface de profil - Mobile

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.



The screenshot shows a web-based application interface for managing users. On the left, there is a sidebar with navigation links: Home, Dashboard, Gérer utilisateur (selected), and Déconnexion. The main content area has a title "Liste des utilisateurs" and a "AJOUTER UTILISATEUR" button. Below is a table with the following data:

ID	Username	Email	Role	Action
2	balti		mecano	
4	boss		admin	
33	chermitou	nourcherni35@gmail.com	chef	
34	mamam	mamamama@gmail.com	Driver	
36	mimouni	medaizzmimouni35@gmail.com	driver	
55	kaka		chef	
57	hanen	hanen.chamakhi@yahoo.fr	Chef	
58	a		mecano	
61	neji	neji@gmail.com	chef	
62		boss		
63	kada	hamzaakadaaf@gmail.com	chef	

FIGURE 3.12 – Interface de gestion des utilisateurs

3.4.4 Crédation 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.

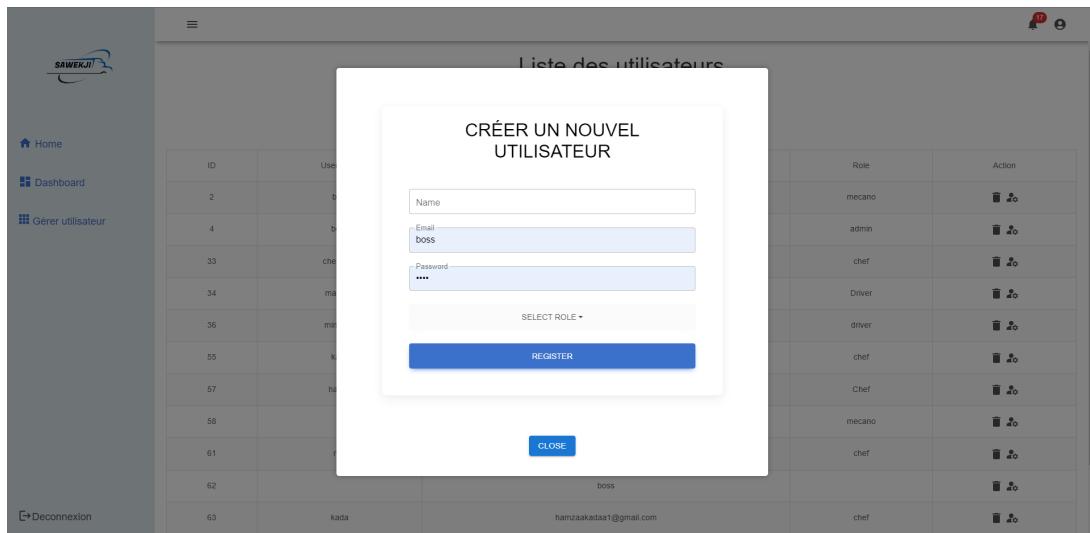


FIGURE 3.13 – Interface de création d'un nouvel utilisateur

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.

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.

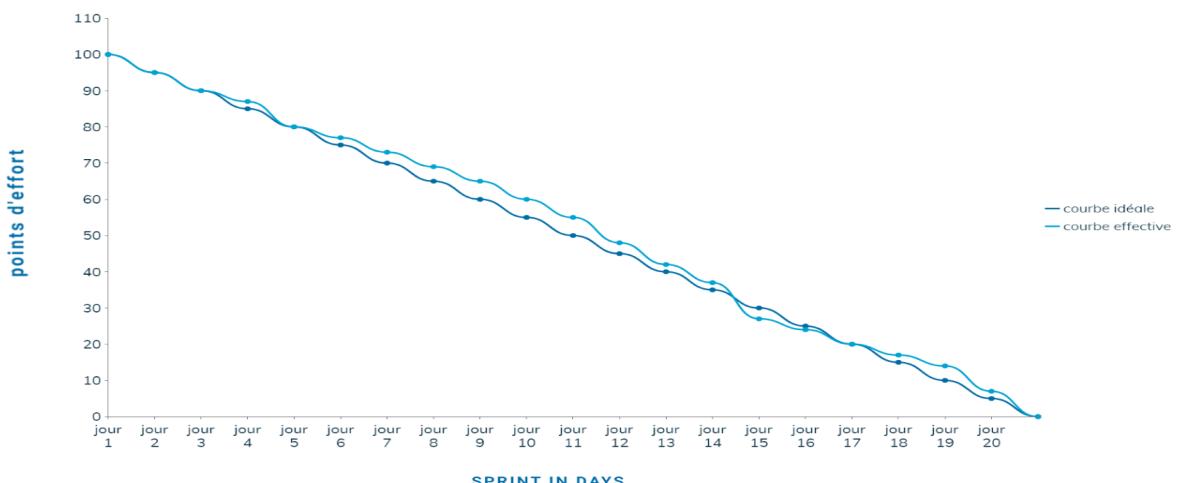


FIGURE 3.14 – Burndown Chart du Sprint 1

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 .

```
http://192.168.1.199:5001/admin/authenticate

Headers (9) Body Scripts Settings
1 pm.test("Authentication successful", function() {
2     pm.response.to.have.status(200); // Check that the response status is 200
3     pm.response.to.have.jsonBody('status', 'success'); // Check that the response body contains a success status
4     console.log(pm.response.json()); // Log the response to the console
5 });
6
7 pm.test("Authentication failed", function() {
8     pm.response.to.have.status(400); // Check that the response status is 400
9     pm.response.to.have.jsonBody('status', 'failure'); // Check that the response body contains a failure status
10});
```

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

POST http://192.168.1.199:5001/admin/authenticate

Params Authorization Headers (9) **Body** Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL

	Key	Value
<input checked="" type="checkbox"/>	username	Text <input type="text" value="kaka"/>
<input checked="" type="checkbox"/>	password	Text <input type="text" value="0000"/>
	Key	Text <input type="text" value="Value"/>

Body Cookies (1) Headers (7) **Test Results (1/2)**

Perform API tests faster with templates for integration testing, regression testing, and more.

All Passed Skipped Failed ⌂

PASS Authentication successful

FAIL Authentication failed | AssertionError: expected response to have status code 400 but got 200

FIGURE 3.17 – Résultat du premier test

POST http://192.168.1.199:5001/admin/authenticate

Params Authorization Headers (9) **Body** Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL

	Key	Value
<input checked="" type="checkbox"/>	username	Text <input type="text" value="kaka"/>
<input checked="" type="checkbox"/>	password	Text <input type="text" value="000000"/>
	Key	Text <input type="text" value="Value"/>

Body Cookies (1) Headers (7) **Test Results (0/2)**

Perform API tests faster with templates for integration testing, regression testing, and more.

All Passed Skipped Failed ⌂

FAIL Authentication successful | AssertionError: expected response body json at "status" to contain "success" but got "failure"

FAIL Authentication failed | AssertionError: expected response to have status code 400 but got 200

FIGURE 3.18 – Résultat 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 Backlog du sprint 2

ID	Fonctionnalités	User Story	Story points
US6	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
US7	Créations des tâches	En tant que Chef d'équipe je veux Créer des tâches pour les chauffeurs et les mécanicien.	5
US8	Consultation des tâches	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
US9	Consultation des tâches	En tant que Mécanicien je veux Consulter les détails des interventions pour que je puisse obtenir toutes les informations sur les tâches assignées.	3
US10	Validation des tâches	En tant que Chef d'équipe je veux valider les tâches du mécanicien	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

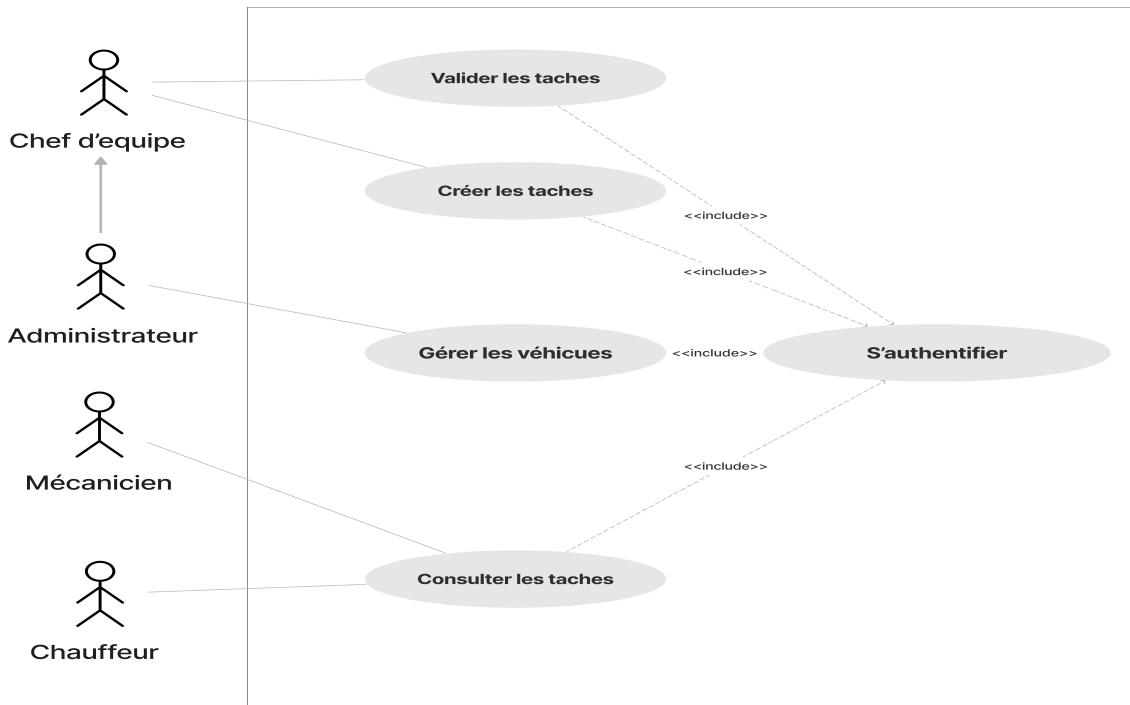


FIGURE 4.1 – Diagramme de cas d'utilisation global du sprint 2

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

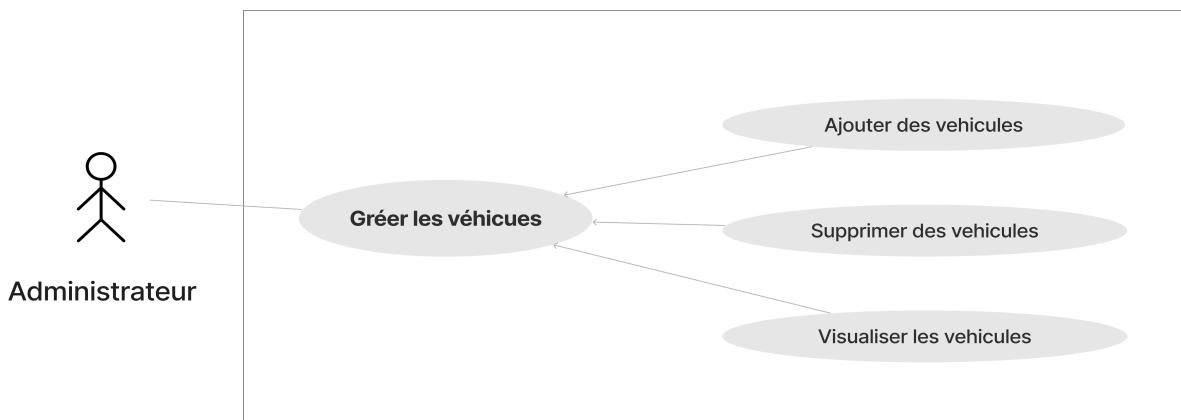


FIGURE 4.2 – Raffinement de cas d'utilisation « Gérer les véhicules »

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

Cas d'utilisation	Ajouter des Véhicules
Acteur	Administrateur
Pré-condition	1- L'administrateur doit être connecté .
Post-condition	Le véhicule est ajouté à la base de données.
Scénario nominal	<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>
Scénario alternatif	<p>A1 Immatriculation déjà utilisée :</p> <p>L'échainement A1 démarre au point 4 du scenario 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 scenario 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 »

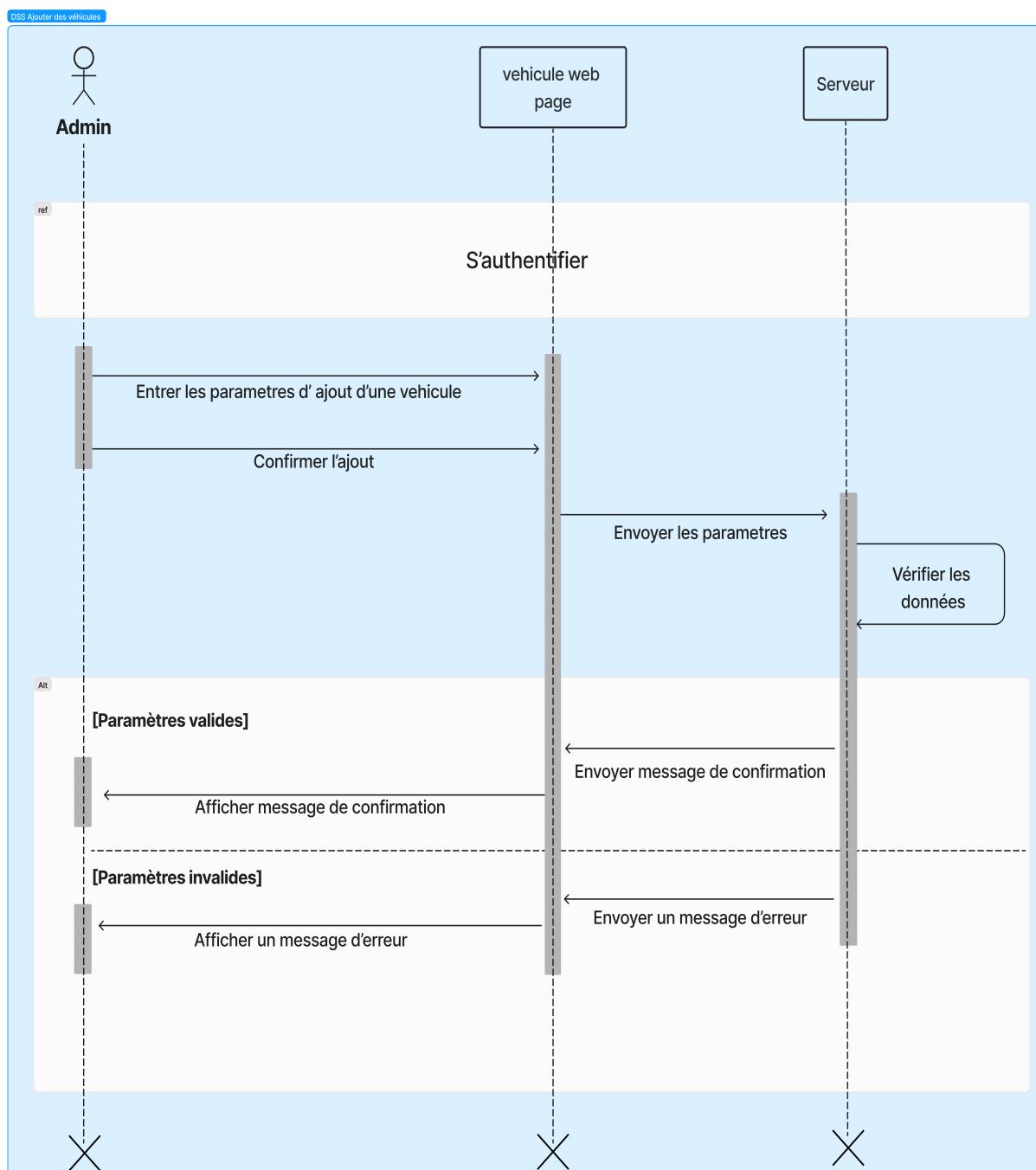


FIGURE 4.3 – Raffinement de cas d'utilisation « Ajout des véhicules »

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

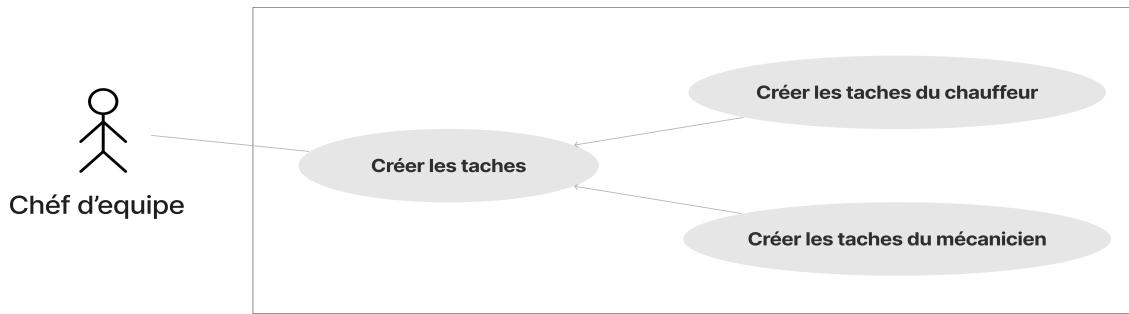


FIGURE 4.4 – Raffinement de cas d'utilisation « Créer les tâches »

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

Cas d'utilisation	Créer des Tâches pour les Mécaniciens
Acteur	Chef d'Équipe
Pré-condition	1-Le chef d'équipe est authentifié dans le système. 2-Il existe une liste de mécaniciens disponibles dans le système.
Post-condition	Les tâches sont créées et assignées aux mécaniciens sélectionnés avec succès.
Scénario nominal	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 vérifie la disponibilité de cette date. 4- Le système affiche une liste des mécaniciens disponibles pour cette date. 5- Le chef d'équipe sélectionne un mécanicien parmi la liste pour lui assigner une tâche. 6- 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>
Scénario alternatif	<p>A1 Date non disponible</p> <p>L'enchaînement A1 démarre au point 3 du scénario nominal.</p> <p>4- un message d'erreur indiquant que pas de mécanicien disponible pour cette date.</p> <p>Le scénario nominal reprend au point 2.</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 »

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

	<p>3- Le chef 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- Le système vérifie les choix du chef d'équipe.</p> <p>8- Le chef 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>9- Le système enregistre la tâche avec toutes les informations nécessaires et informe le chauffeur assigné.</p>
Scénario alternatif	<p>A1 Chauffeur qualifié non disponible</p> <p>L'enchaînement A1 démarre au point 7 du scénario nominal.</p> <p>4- un message d'erreur pour refaire le choix du chauffeur.</p> <p>Le scénario nominal reprend au point 6.</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 pour les chauffeurs»

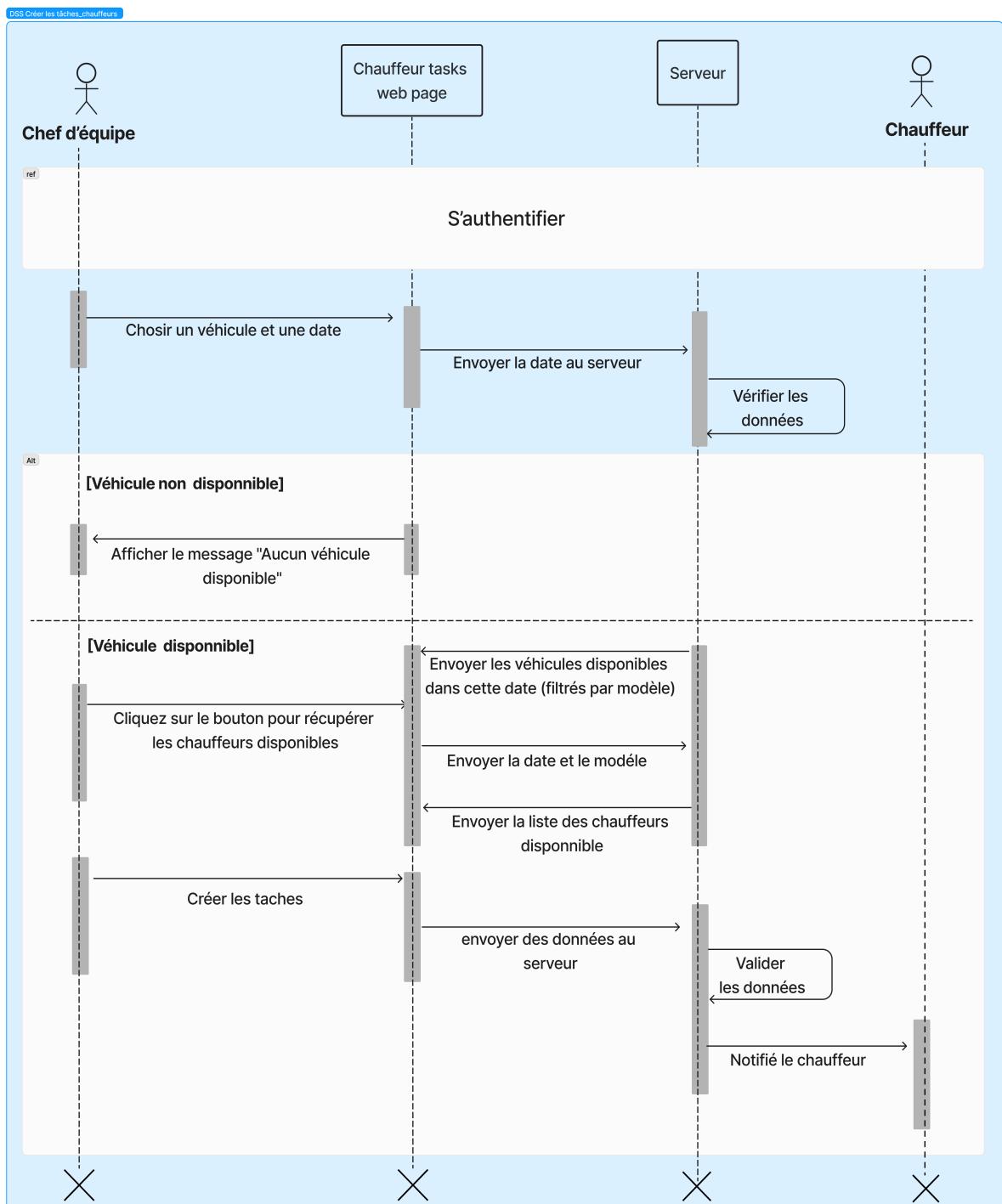


FIGURE 4.5 – Diagramme de séquence système du cas d'utilisation «Créer les tâches pour les chauffeurs»

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

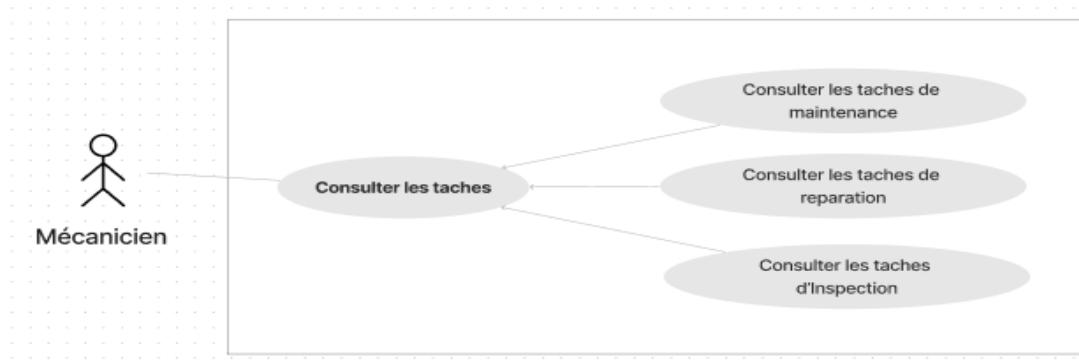


FIGURE 4.6 – Analyse de cas d'utilisation « Consulter les tâches pour les chauffeurs »

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

Cas d'utilisation	Consulter les tâches des chauffeurs
Acteur Principal	Chauffeur
Acteurs Secondaires	Chef d'Équipe
Pré-condition	1- Le chauffeur doit être connecté à l'application . 2- Des tâches doivent être assignées au chauffeur.
Post-condition	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.
Scénario nominal	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>
Scénario alternatif	<p>A1 Absence de Checklist</p> <p>L'enchaînement démarre au point 3 du scénario nominal.</p> <p>4- Un message d'erreur s'affiche pour réaccéder à la checklist.</p> <p>Le scénario reprend au point 2.</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 taches des chauffeurs»

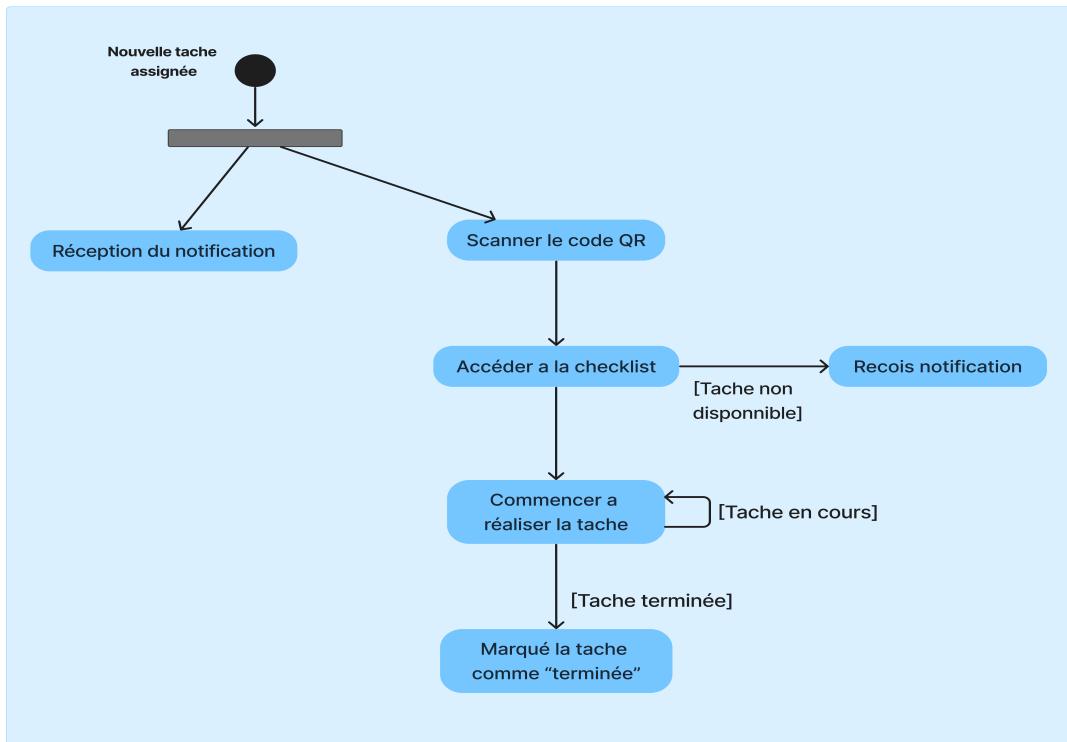


FIGURE 4.7 – Diagramme d'activité du cas d'utilisation « Consulter les tâches des chauffeurs »

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

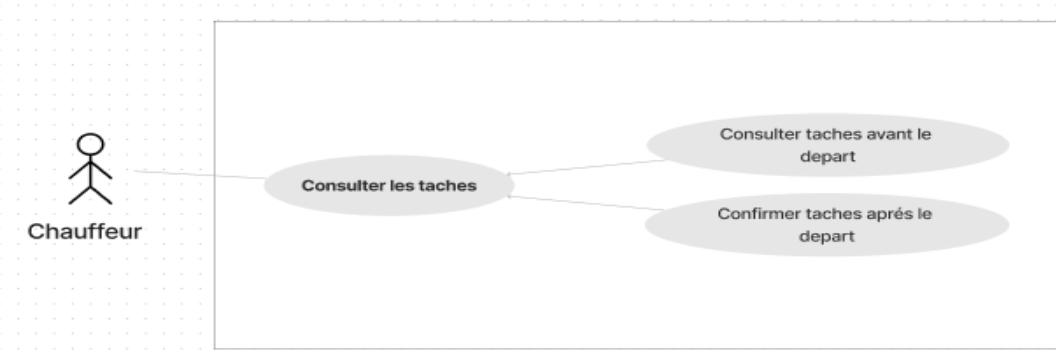


FIGURE 4.8 – Analyse de cas d'utilisation « Consulter les tâches pour les mécaniciens »

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

Cas d'utilisation	Consulter les tâches des mécaniciens
Acteur Principal	Mécanicien
Acteurs Secondaires	Chef d'Équipe
Pré-condition	1- Le mécanicien doit être authentifié dans le système. 2- Des tâches doivent être assignées au mécanicien.
Post-condition	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.
Scénario nominal	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>
Scénario alternatif	<p>A1 Absence de taches</p> <p>L'enchaînement démarre au point 3 du scénario nominal.</p> <p>4- Un message d'erreur s'affiche pour réaccéder à la checklist.</p> <p>Le scénario reprend au point 2.</p>

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

4.3 Diagramme de classes du sprint 2

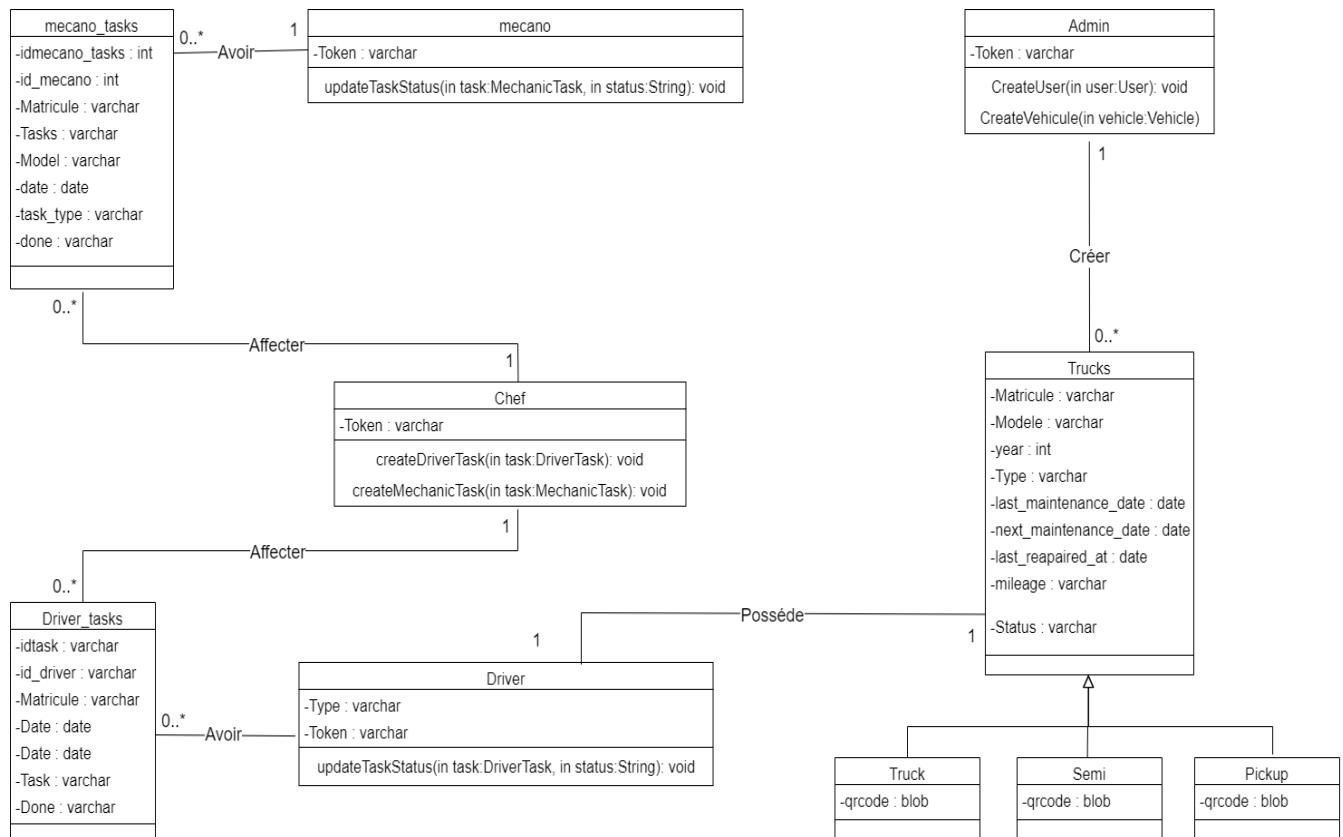


FIGURE 4.9 – Diagramme de classes du sprint 2

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.

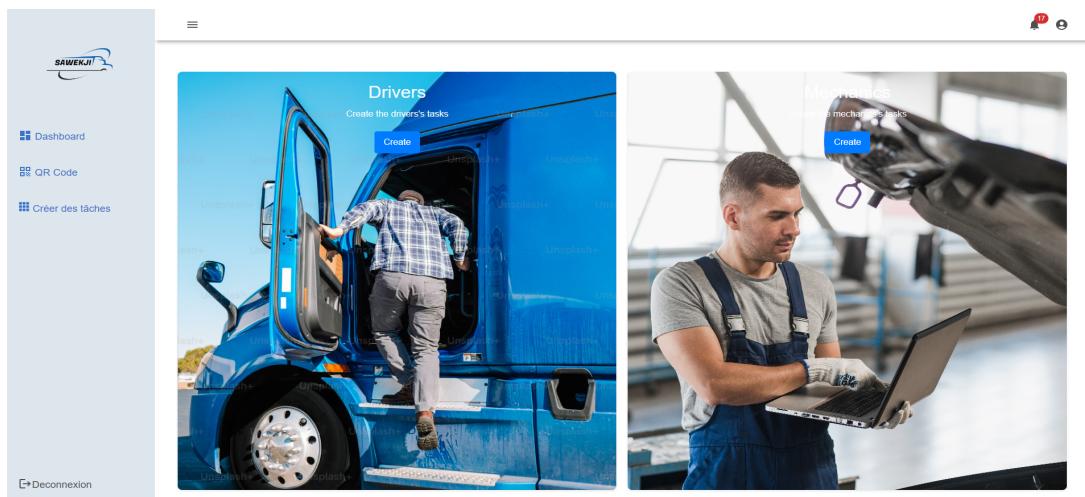


FIGURE 4.10 – interface de création des tâches

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

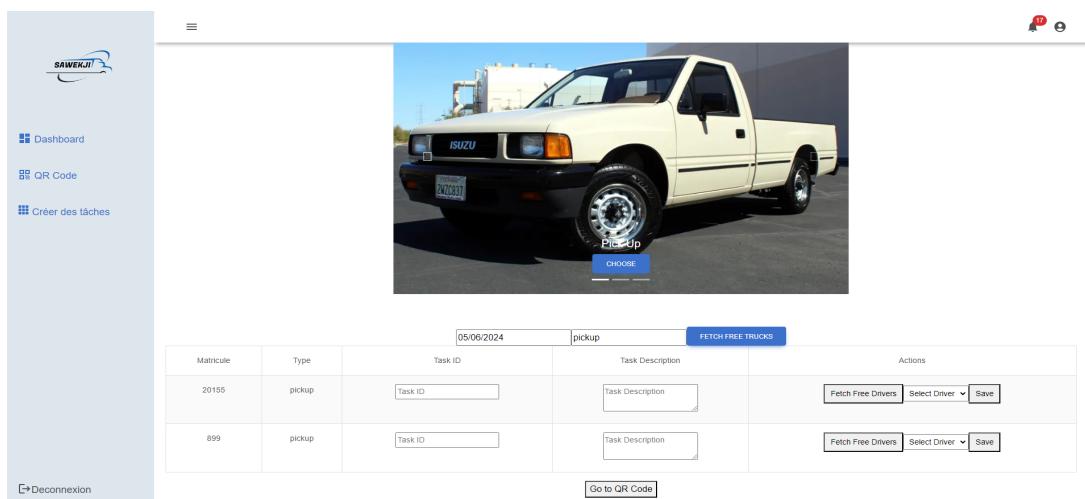


FIGURE 4.11 – Crédit : SAWEKJI

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.

Name	Tasks	Action
ena	<input type="text" value="Enter tasks..."/>	<button>Save</button>
haroun	<input type="text" value="Enter tasks..."/>	<button>Save</button>
houssem	<input type="text" value="Enter tasks..."/>	<button>Save</button>

FIGURE 4.12 – 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 4.13 – 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 4.14 – interface de consultation des tâches pour les mécaniciens

4.4.4 Validation des tâches du mécanicien par le chef d'équipe

Cette interface permet au chef d'équipe de confirmer les tâches effectuées par les mécaniciens. Elle affiche une liste des tâches en attente de validation. Chaque tâche est présentée par sa description et le nom du mécanicien responsable de la tâche.

Le chauffeur, après avoir vérifié sur le terrain l'accomplissement de la tâche, il peut confirmer la tâche en appuyant sur le bouton "Confirm Task" correspondant.

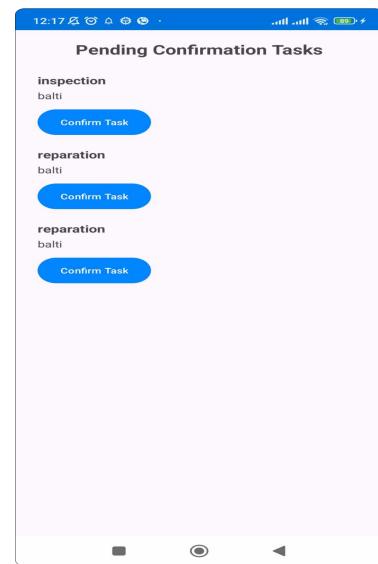


FIGURE 4.15 – interface de consultation des tâches pour les mécaniciens

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.

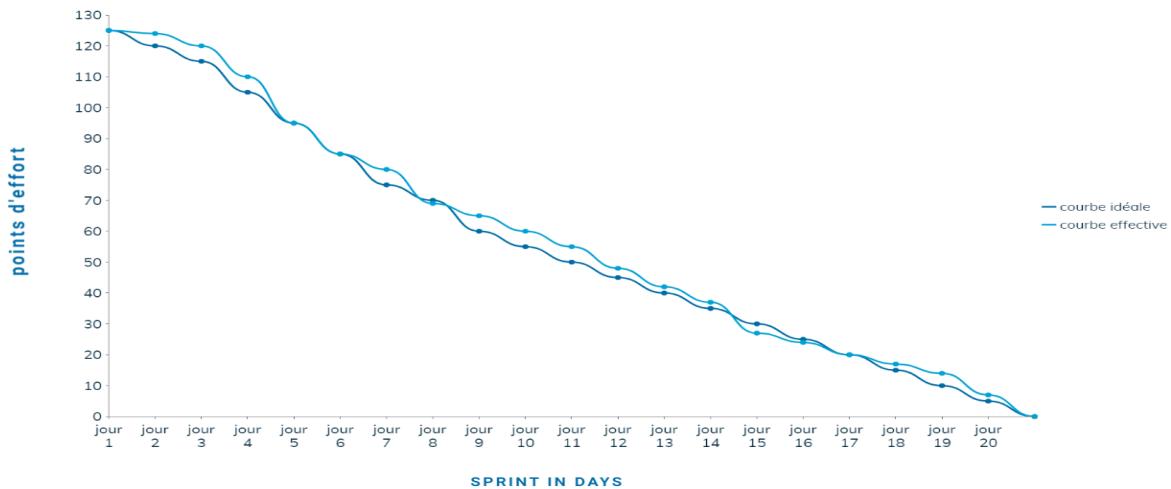


FIGURE 4.16 – Burdown Chart du Sprint 2

4.5.2 Task Bord

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



FIGURE 4.17 – Task board du Sprint 2

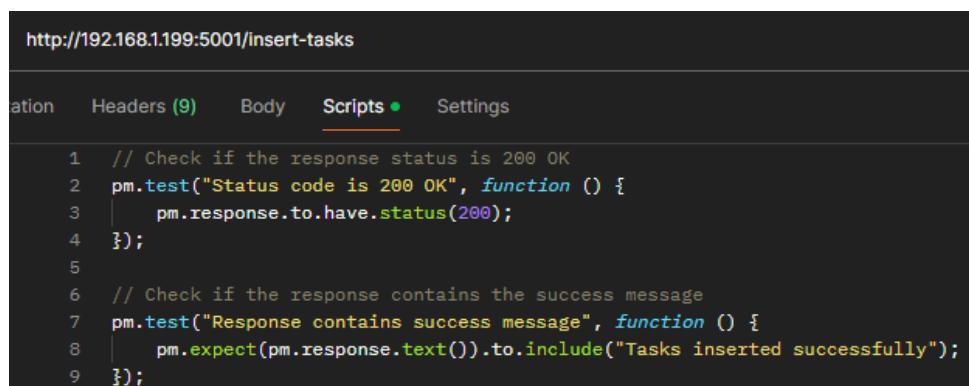
4.6 tests unitaires

Test unitaires du cas d'utilisation « Créer 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.



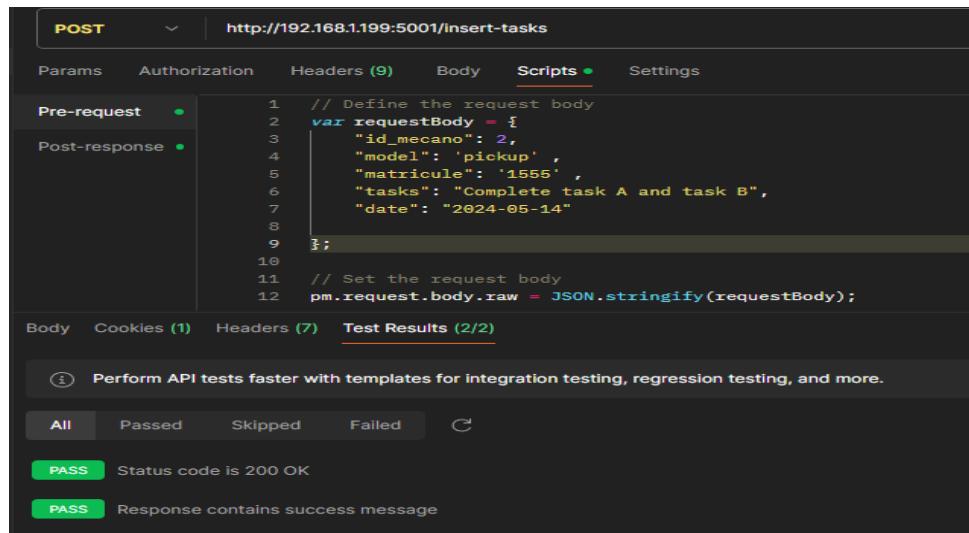
```
http://192.168.1.199:5001/insert-tasks

Action Headers (9) Body Scripts • Settings

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 });


```

FIGURE 4.18 – Code souce du test unitaire du cas d'utilisation « Créer taches »



POST http://192.168.1.199:5001/insert-tasks

Params	Authorization	Headers (9)	Body	Scripts •	Settings
Pre-request				<pre>1 // Define the request body 2 var requestBody = { 3 "id_mecano": 2, 4 "model": 'pickup', 5 "matricule": '1555', 6 "tasks": "Complete task A and task B", 7 "date": "2024-05-14" 8 }; 9 10 // Set the request body 11 pm.request.body.raw = JSON.stringify(requestBody);</pre>	
Post-response					
Body	Cookies (1)	Headers (7)	Test Results (2/2)		

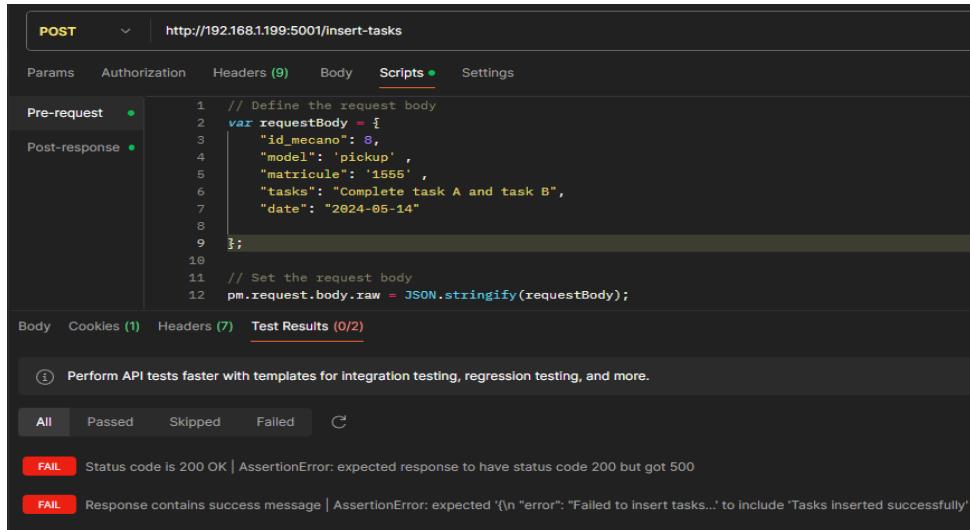
(i) Perform API tests faster with templates for integration testing, regression testing, and more.

All Passed Skipped Failed C

PASS Status code is 200 OK

PASS Response contains success message

FIGURE 4.19 – Résultat du test de réussite



The screenshot shows a Postman test run for a POST request to `http://192.168.1.199:5001/insert-tasks`. The Pre-request script contains the following code:

```

1 // Define the request body
2 var requestBody = {
3   "id_mecano": 8,
4   "model": 'pickup',
5   "matricule": '1555',
6   "tasks": "Complete task A and task B",
7   "date": "2024-05-14"
8 };
9 ;
10
11 // Set the request body
12 pm.request.body.raw = JSON.stringify(requestBody);

```

The Test Results section shows two failing assertions:

- FAIL** Status code is 200 OK | Assertion Error: expected response to have status code 200 but got 500
- FAIL** Response contains success message | Assertion Error: expected '{\n "error": "Failed to insert tasks..."' to include 'Tasks inserted successfully'

FIGURE 4.20 – 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'implementation des fonctionnalités, tandis que la "Retrospective" et les "Tests unitaires" ´evaluent les methodes de developpement pour assurer la qualite du produit.

5.1 Backlog du sprint 3

ID	Fonctionnalités	User Story	Story points
US11	Suivi en temps reel	En tant que Chef d'équipe je veux suivre le progrès et la localisation des voitures .	8
US12	Planification des deplacements	En tant que chauffeur, je veux planifier des voyages pour optimiser mes itineraires et mes arrêts.	5
US13	Notifications	En tant qu'administrateur, chauffeur, mecanicien ou chef d'équipe, je veux recevoir des notifications pour tre informe en temps reel des mises à jour.	5
US14	Creation des rapports	En tant que Mecanicien je veux Générer des rapports pour que je puisse documenter les details 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.

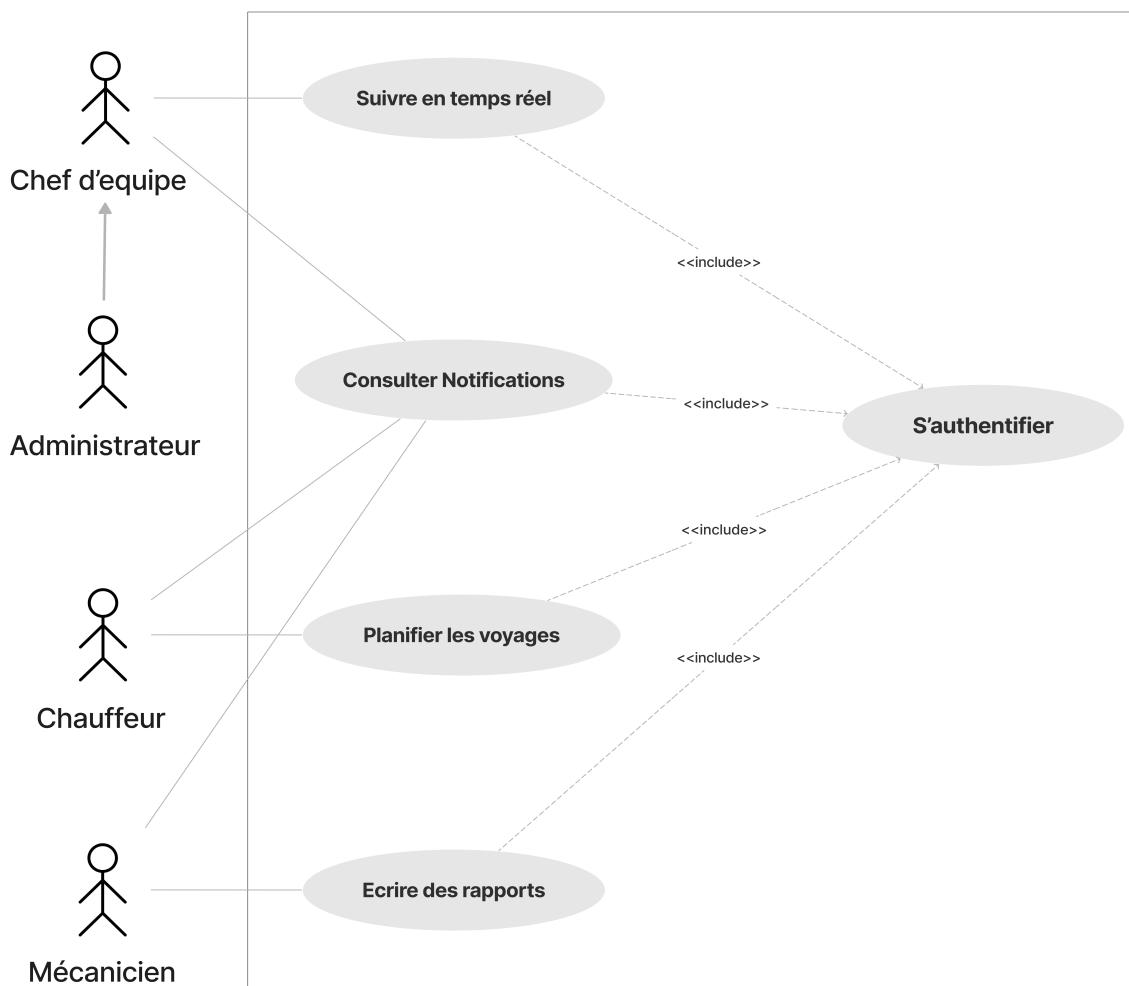


FIGURE 5.1 – Diagramme de cas d'utilisation global du sprint 3

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

Cas d'utilisation	Suivre en temps réel
Acteur	Chef d'équipe et Chauffeur
Pré-condition	<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>
Post-condition	<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>
Scénario nominal	<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 planifie les déplacements avant son départ (appel au cas d'utilisation "Planifier les déplacements").</p> <p>4- Le chef s'authentifie dans l'application (appel au cas d'utilisation "S'authentifier").</p> <p>5- Le chef accède à l'interface de suivi en temps réel.</p> <p>6- Le système vérifie la position actuelle du camion et l'affiche sur la carte.</p>

	7- Le chef voit l'emplacement marqué en temps réel sur la carte.
Scénario alternatif	<p>A1 Position GPS non disponible :</p> <p>l'enchaînement 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 »

Cas d'utilisation	Planifier les déplacements
Acteur	Chauffeur
Pré-condition	Le chauffeur doit être connecté à l'application .
Post-condition	Les détails du trajet choisi, des pauses et des arrêts pour le carburant sont enregistrés dans le système.
Scénario nominal	<p>1- Le chauffeur sélectionne le trajet optimal sur la carte .</p> <p>2- Pendant le trajet, le chauffeur peut localiser les différentes stations de carburant à proximité du trajet prévu.</p> <p>3- Le système vérifie le trajet, l'enregistre, et le marque sur la carte.</p> <p>4- Toutes les actions sont mises à jour en temps réel sur la carte.</p>
Scénario alternatif	<p>A1 Erreur de vérification du trajet :</p> <p>l'enchaînement A1 démarre au point 3 du scénario nominal.</p> <p>4- Un message d'erreur s'affiche pour vérifier le trajet sélectionné.</p> <p>Le scénario nominal reprend au point 1.</p>

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

5.2.4 Description textuelle du cas d'utilisation « Consulter Notifications »

Cas d'utilisation	Consulter les notifications
Acteur	Administrateur, Chef d'équipe, Chauffeur et Mécanicien
Pré-condition	1- L'utilisateur doit être authentifié dans l'application mobile. 2- L'utilisateur doit avoir accès à l'interface de consultation des notifications.
Post-condition	L'utilisateur peut voir toutes les notifications.
Scénario nominal	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 ”

5.2.5 Diagramme d'activité du cas d'utilisation « Consulter Notifications »

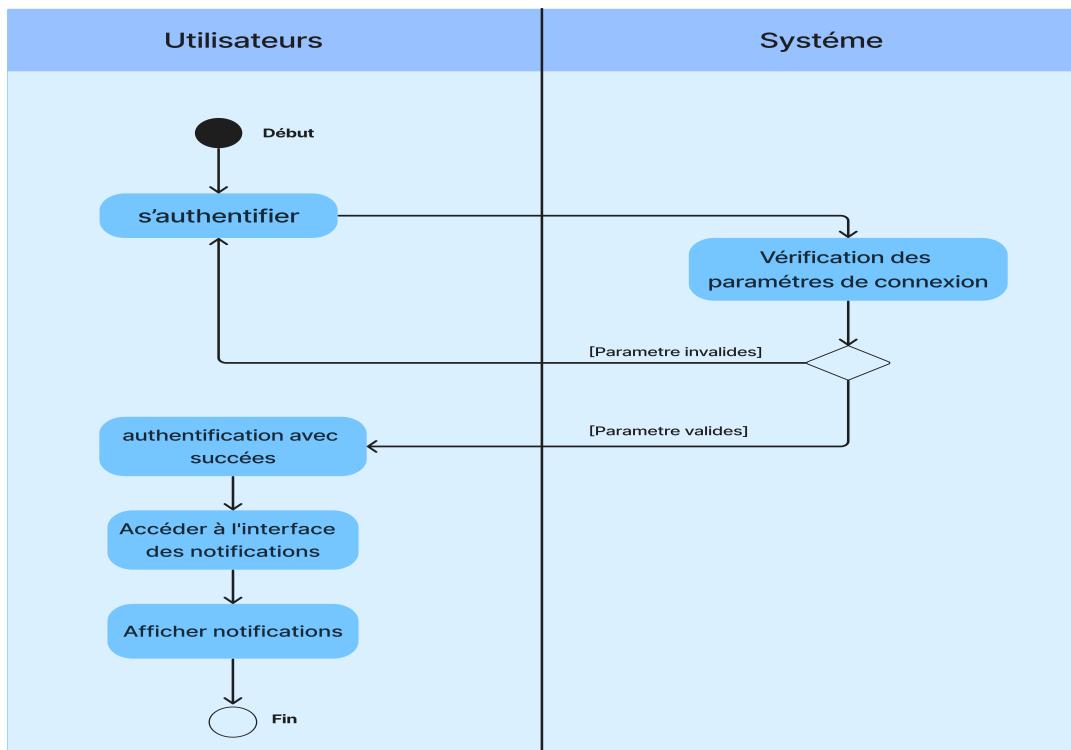


FIGURE 5.2 – Diagramme d'activité du cas d'utilisation « Consulter Notifications »

5.2.6 Raffinement du cas d'utilisation « Générer des rapports »

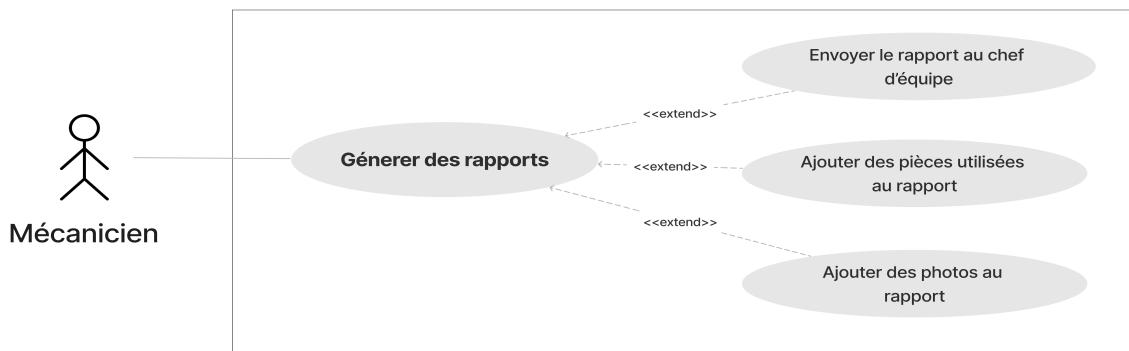


FIGURE 5.3 – Raffinement du cas d'utilisation « Générer des rapports »

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

Cas d'utilisation	Écrire des rapports
Acteur	Mécanicien
Pré-condition	Le mécanicien doit être authentifié dans l'application mobile et a l'accès à l'interface de saisie des rapports.
Post-condition	<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>
Scénario nominal	<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>

	<p>7- Le système enregistre les informations saisies et génère un rapport.</p>
Scénario alternatif	<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 ”

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

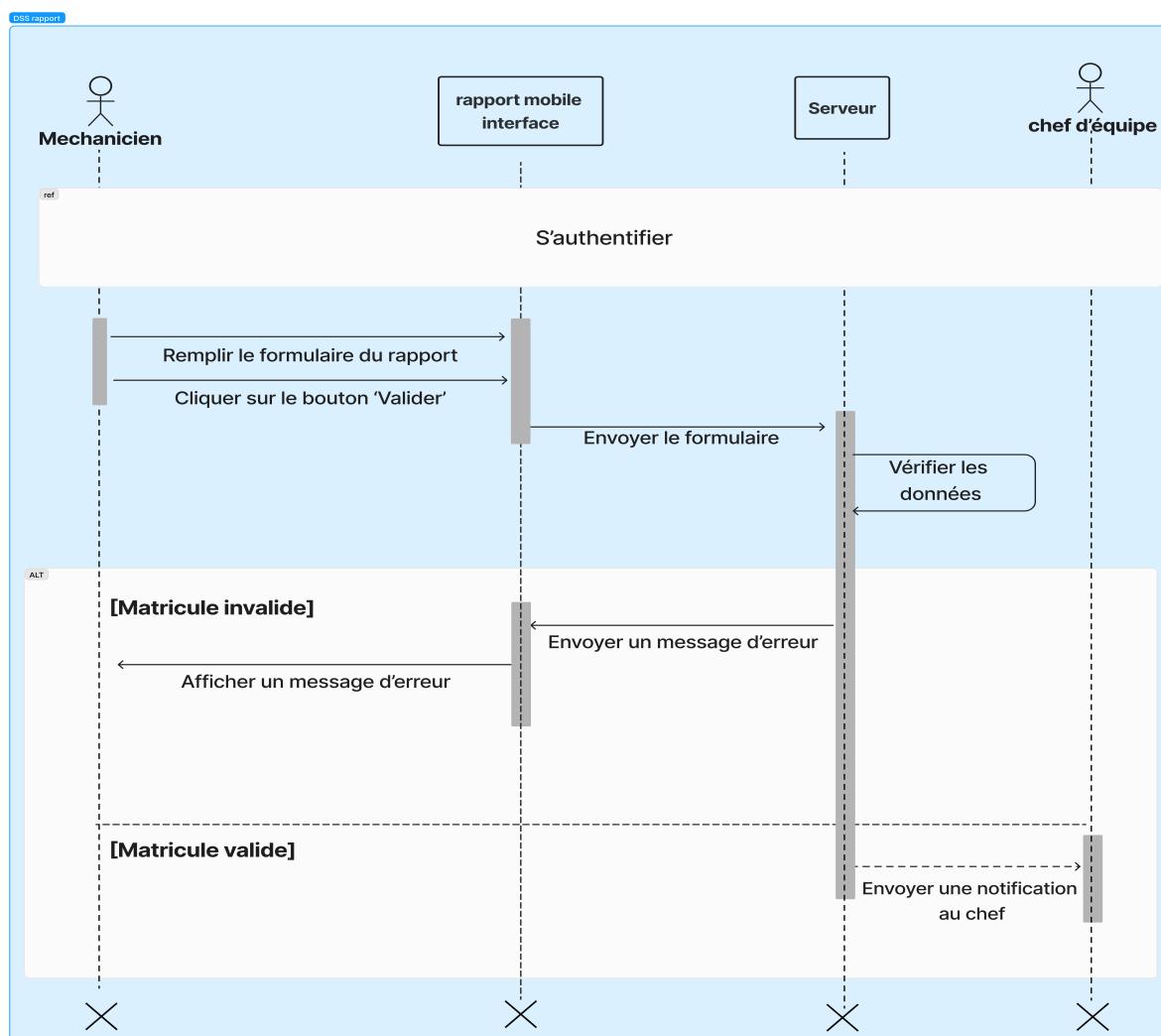


FIGURE 5.4 – Diagramme de séquence système du cas d'utilisation « Générer des rapports »

5.3 Diagramme de classes du sprint 3

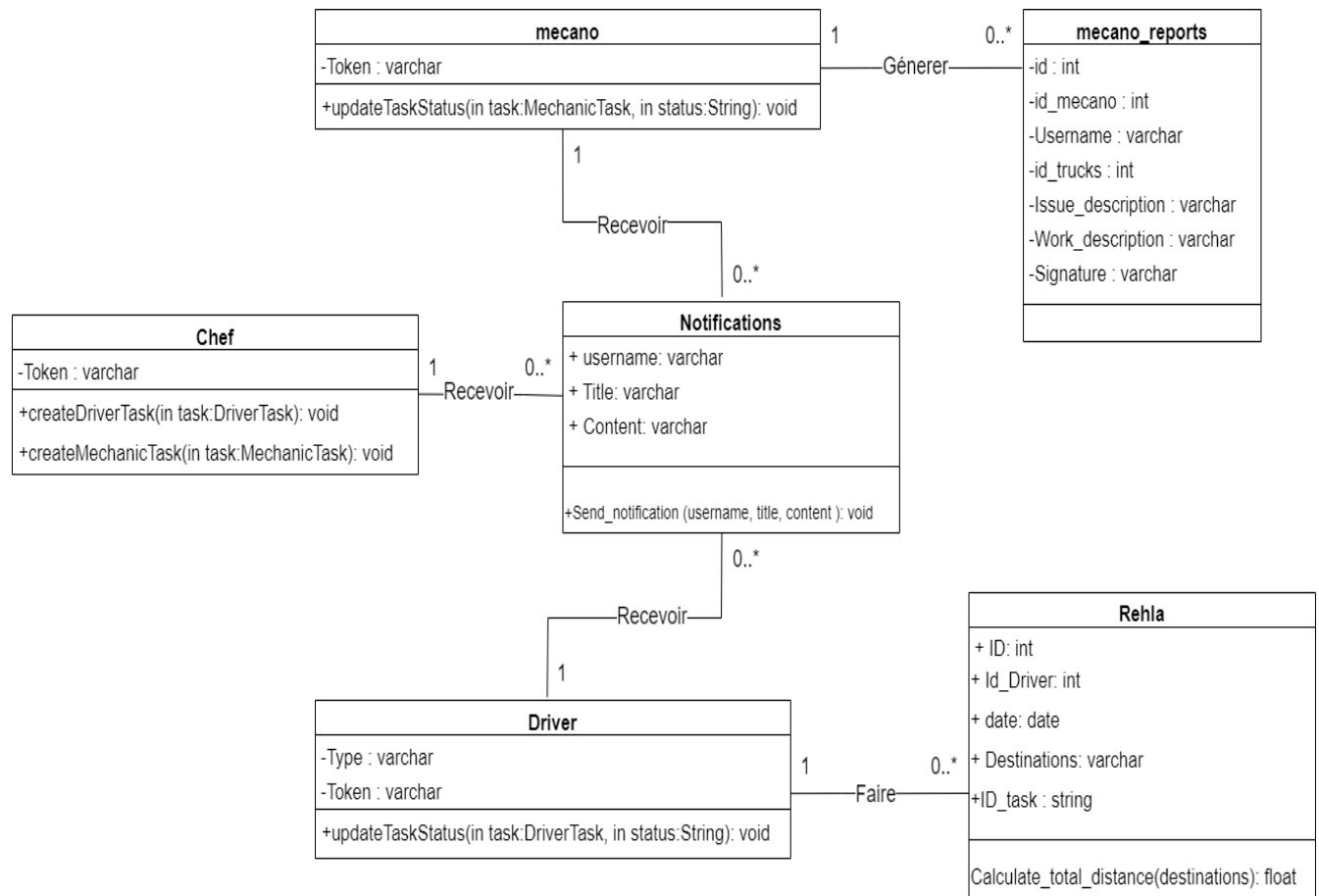


FIGURE 5.5 – Diagramme de classes du sprint 3

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.

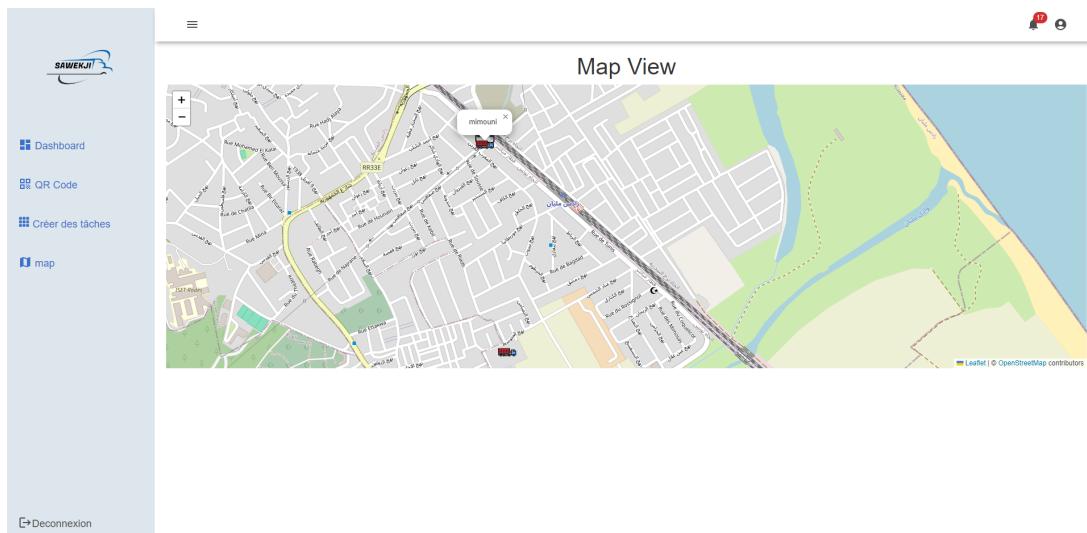
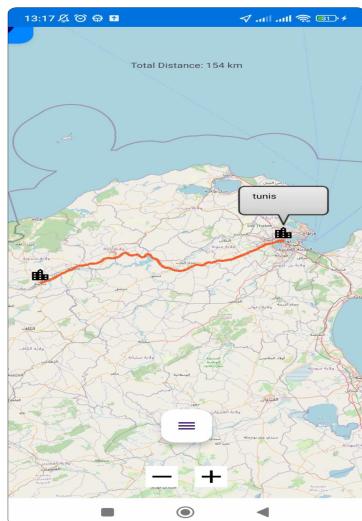


FIGURE 5.6 – Suivi en Temps Réel par Géolocalisation

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 5.7 – 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 .

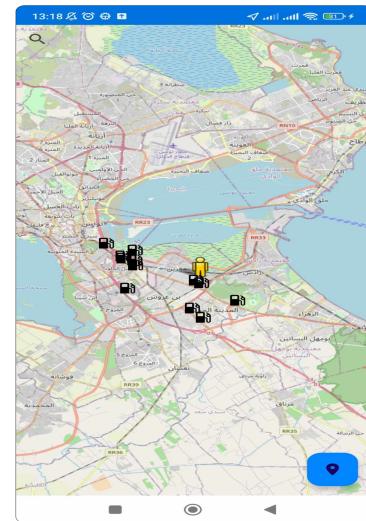


FIGURE 5.8 – Carte de voyage

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.

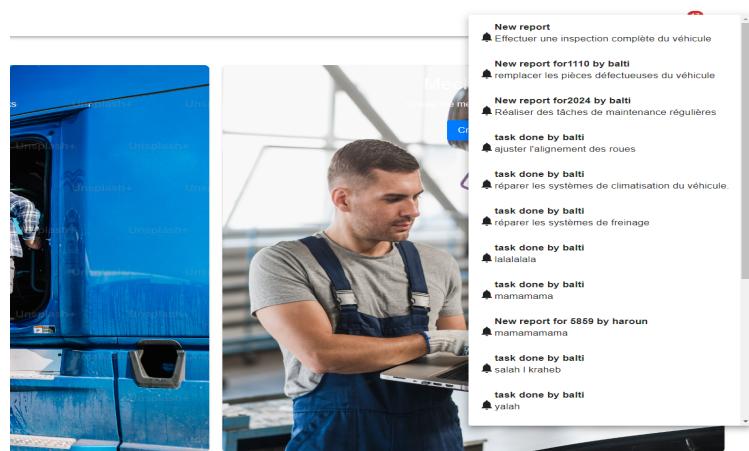


FIGURE 5.9 – Interface de Consultation des notifications - « web »



FIGURE 5.10 – Interface de Consultation des notifications - « Mobile »

5.4.4 Génération d'un Rapport

Cette interface permet aux mécaniciens de saisir les informations d'un véhicule et de rédiger un rapport. Elle propose des champs pour l'ID du véhicule, la description du problème, le travail effectué et la signature du mécanicien. Des boutons permettent d'importer des photos et de sauvegarder le rapport.



FIGURE 5.11 – Génération de Rapport - Interface Initiale

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

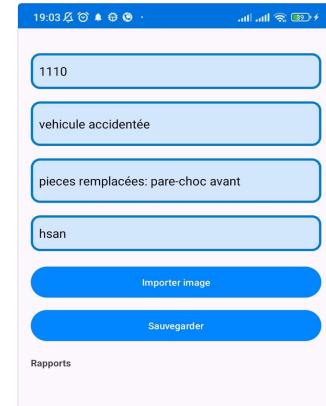


FIGURE 5.12 – Génération de Rapport - Détails Entrés

Cette interface affiche le rapport sauvegardé avec le matricule, les détails du problème initial et le travail effectué.

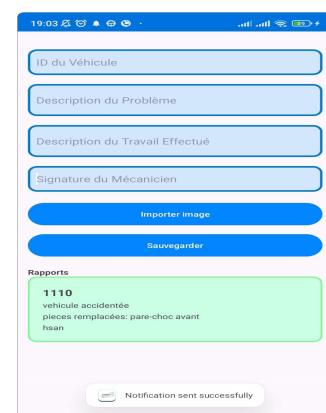


FIGURE 5.13 – Génération de Rapport - Rapport Sauvegardé

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.

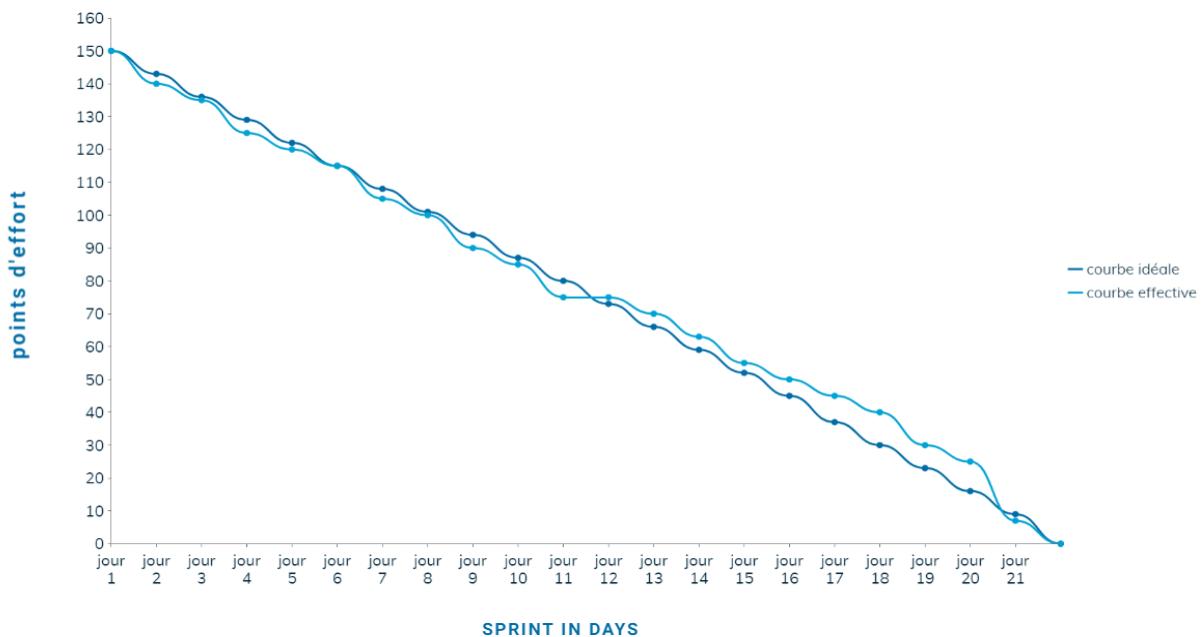


FIGURE 5.14 – Burdown Chart du Sprint 3

5.5.2 Task Bord

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

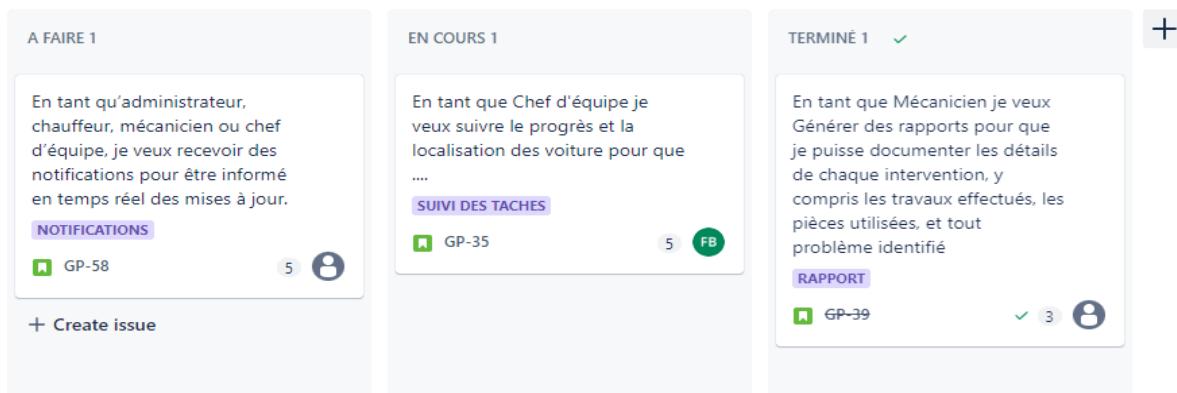


FIGURE 5.15 – Task board du Sprint 3

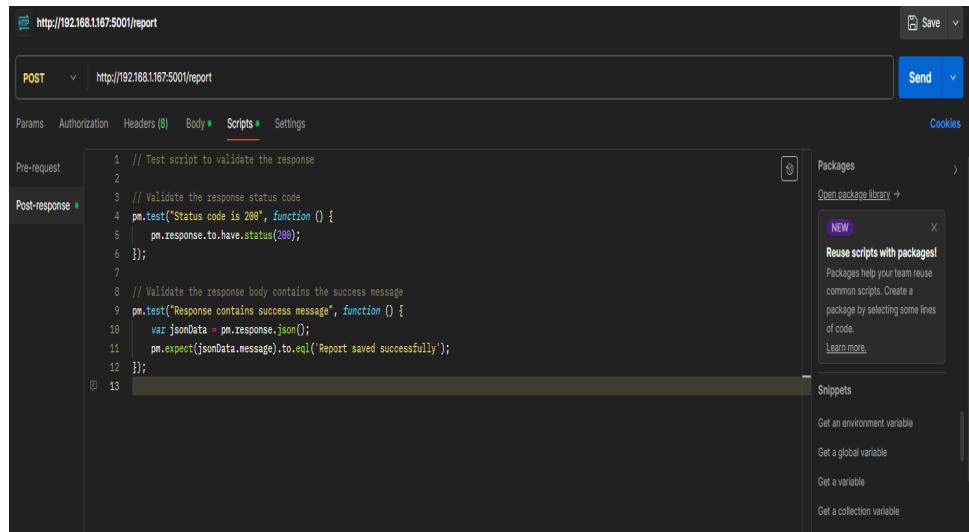
5.6 Tests unitaires

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 test script for validating a report generation endpoint. The URL is `http://192.168.1.167:5001/report`. The 'Scripts' tab is selected, containing the following code:

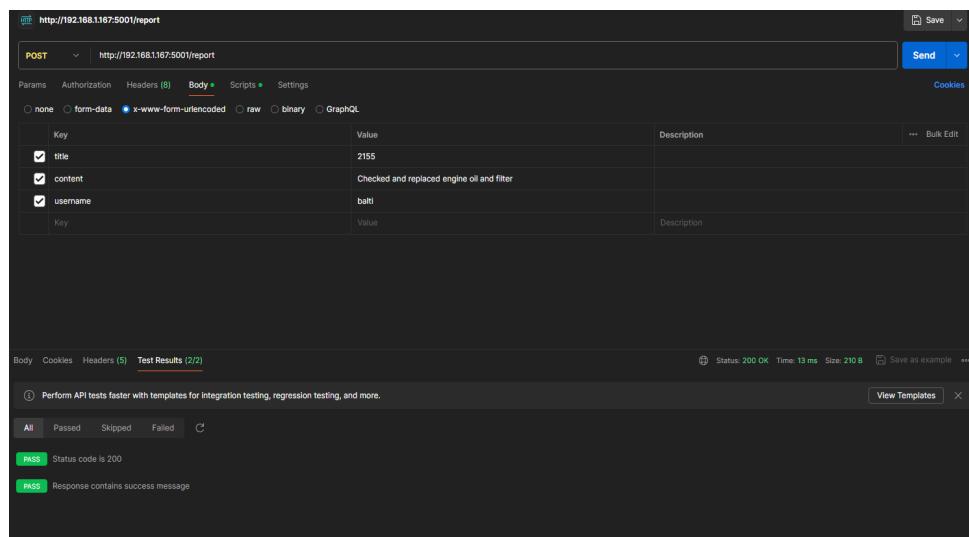
```

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

```

A floating 'Reuse scripts with packages!' modal is open on the right side.

FIGURE 5.16 – Code source du test unitaire du cas d'utilisation « Générer des rapports »



The screenshot shows the Postman interface after a successful test run. The URL is `http://192.168.1.167:5001/report`. The 'Body' tab is selected, showing form-data parameters: title (2155), content (Checked and replaced engine oil and filter), and username (bati). The 'Test Results' section shows two successful tests:

- PASS Status code is 200
- PASS Response contains success message

The status bar at the bottom indicates: Status 200 OK, Time: 13 ms, Size: 210 B.

FIGURE 5.17 – Résultat du test de réussite

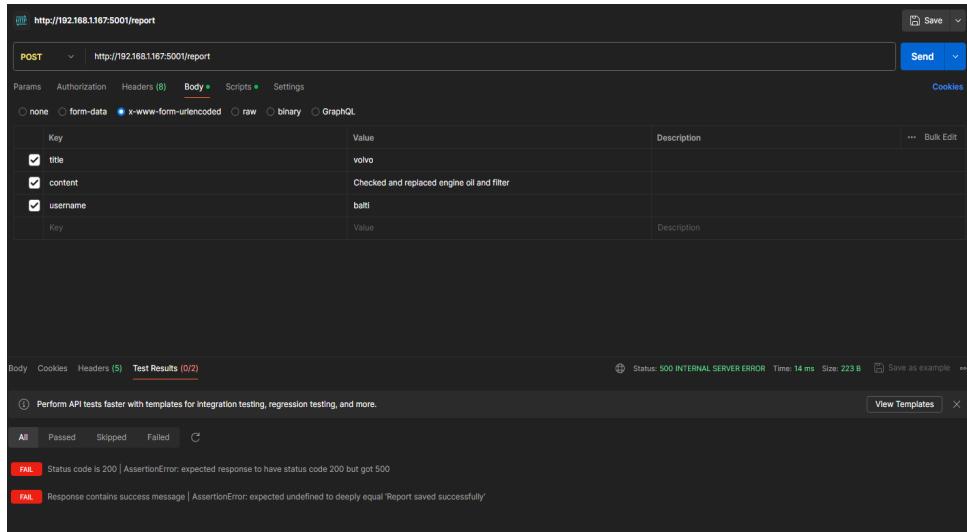


FIGURE 5.18 – Resultat du test d'échec

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 rapports a amélioré l'efficacité des mécaniciens. Les ajustements basés sur les outils de suivi préparent efficacement la transition vers le dernier sprint.

Chapitre 6

Sprint 4 : Implémentation des tableaux de bord

Introduction

Dans ce chapitre, nous détaillerons les étapes et processus de l'implémentation des tableaux de bord durant le sprint 4 de notre projet. Ces tableaux de bord, essentiels pour une visualisation claire des données, facilitent la prise de décisions éclairées pour les différentes parties prenantes. Ce sprint a mis l'accent sur les fonctionnalités de consultation adaptées aux profils spécifiques d'utilisateurs, chacun ayant des besoins particuliers pour améliorer l'efficacité opérationnelle et la gestion proactive des missions.

6.1 Backlog du sprint 4

ID	Fonctionnalités	User Story	Story points
US11	Consulter dashboard	En tant qu'administrateur je veux Consulter le Dashboard pour que je puisse superviser l'ensemble des opérations du parc	8
US12		En tant que Chef d'équipe je veux Consulter le Dashboard pour que je puisse gérer les missions de manière proactive	8
US13		En tant que Chauffeur je veux Consulter le Dashboard pour que je puisse avoir une vue claire de mes missions actuelles	8
US14		En tant que Mécanicien je veux Consulter le Dashboard pour que je puisse avoir une vue d'ensemble des opérations de maintenance	8

TABLE 6.1 – Backlog du sprint 4

6.2 Conception du data warehouse

Il y a en gros trois modélisations possibles pour organiser les données stockées dans un Data Warehouse

Modèle en Etoile :

Un schéma en étoile, ou modèle de données "en étoile", est une structure multidimensionnelle utilisée pour stocker des données atomiques ou agrégées, généralement dans des entrepôts de données ou des datamarts. Souvent considéré comme un modèle dénormalisé, le modèle en étoile offre l'avantage d'économiser les jointures lors des requêtes, ce qui le rend particulièrement optimisé pour les analyses.

Modèle en flocon de neige :

Les données des tables de dimension sont normalisées, ce qui signifie que les données sont stockées dans des tables de dimension séparées sans redondance, ça veut dire que les données qu'appartiennent à une dimension ne se répètent pas.

Modèle en Constellation :

Lorsque plusieurs schémas en étoile partagent des dimensions (voir dimensions conformes) on parle de schéma en constellation.

6.2.1 Choix du modèle

Nous avons choisi le modèle en constellation car notre entrepôt de données repose sur quatre tables de faits qui partagent une ou plusieurs dimensions communes entre elles. Pour les DataMart nous avons opté pour le modèle en étoile .

6.3 Définition des indicateurs de performance

Les indicateurs de performance (KPI) sont des mesures quantifiables qui permettent d'évaluer la réussite d'une organisation, d'un projet, d'un processus ou d'une initiative particulière par rapport à ses objectifs stratégiques ou opérationnels . Nous allons définir ces indicateurs dans le tableau 6.2.

Indicateur	Description
Taux d'achèvement des tâches	Nombre de tâches terminées / Nombre total de tâches
Taux de réparation des véhicules par catégorie	Nombre de véhicules réparés / Nombre total de véhicules par catégorie
Kilomètres parcourus par semaine	Somme des kilomètres parcourus chaque semaine
Tâches terminées par semaine	Nombre de tâches terminées chaque semaine
Tâches terminées par semaine	Nombre de tâches terminées chaque semaine

Kilomètres par véhicule (filtrés par matricule)	Somme des kilomètres parcourus par véhicule (filtrés par matricule)
Taux de panne par modèle de véhicule	Nombre de véhicules en panne / Nombre total de véhicules par modèle
Nombre de nouveaux utilisateurs par période	Nombre de nouveaux utilisateurs ajoutés pendant une période donnée
Nombre total d'utilisateurs actifs par semaine	Nombre total d'utilisateurs actifs chaque semaine

TABLE 6.2 – Indicateurs de performance

6.4 Schéma en constellation

Nous allons détailler la conception de l'entrepot de données.

6.4.1 Les tables de fait

nous avons quatre sujets d'analyse, donc il est essentiel de mettre en place quatre tables de faits distinctes.

- **Fait taches-Chauffeur :**

- **ID-task** : Identifiant unique de chaque tâche effectuée par le chauffeur
- **ID-driver** : Identifiant unique de chaque chauffeur
- **Date** : La date de réalisation de la tâche
- **vehicule-ID** : Identifiant unique de chaque véhicule
- **ID-destination** : Identifiant de la destination de la tâche

- **Fait gestion des Véhicules :**

- **Vehicule-ID** : Identifiant unique de chaque véhicule
- **Last-maintenance-date-key** : Clé de date indiquant la dernière fois qu'un véhicule a été entretenu
- **next-maintenance-date-key** : Clé de date prévue pour la prochaine maintenance
- **Last-repaired-at-key** : Clé de localisation ou de service indiquant où le véhicule a été réparé pour la dernière fois
- **Maintenance-interval** : Interval spécifié entre les maintenances
- **Time-since-last-repair** : Durée écoulée depuis la dernière réparation

- **Fait taches-Mécanicien :**
 - **User-ID** : Identifiant de l'utilisateur (mécanicien) responsable de l'activité.
 - **Task-ID** : Identifiant de la tâche spécifique effectuée par le mécanicien.
 - **Vehicule-ID** : Identifiant du véhicule sur lequel l'activité a été effectuée.
 - **Task-date-key** : Clé pour enregistrer la date et l'heure de l'activité.
 - **Total-tasks** : Nombre total de tâches assignées au mécanicien.
 - **Total-done-tasks** : Nombre de tâches terminées par le mécanicien.
 - **Percentage-tasks-done** : Pourcentage de tâches terminées par rapport au nombre total de tâches assignées au mécanicien.
- **Fait gestion des utilisateur :**
 - **User-ID** : Identifiant unique de l'utilisateur.
 - **Last-time-active** : Date et heure de la dernière activité de l'utilisateur.
 - **Hire-date** : Date à laquelle l'utilisateur a été embauché.
 - **Status** : Statut actuel de l'utilisateur (actif / inactif).
 - **Activity-count** : Nombre total d'activités effectuées par l'utilisateur.
 - **Active-users-count** : Nombre total d'utilisateurs actifs sur une période donnée.
 - **Inactive-users-count** : Nombre total d'utilisateurs inactifs sur une période donnée.
 - **New-users-count** : Nombre total de nouveaux utilisateurs ajoutés sur une période donnée.
 - **Avg-activity-per-day** : Moyenne des activités par jour pour chaque utilisateur.

6.4.2 Les tables de dimensions

Dans cette section, nous allons présenter les tables de dimensions essentielles pour l'analyse du processus.

Cette table Contient des informations d'identification et de contact sur les utilisateurs, ainsi que leur rôle et statut dans le système, utilisées pour gérer l'accès et suivre l'activité des utilisateurs

Dim-Users
- Users-ID pk
- Username
- Email
- Role
- Status

FIGURE 6.1 – Dimension Users

Cette table Fournit des détails structurés sur les dates pour les analyses temporelles, incluant des informations sur les jours, mois, années, et trimestres, essentielles pour le reporting et les analyses périodiques.

Dim-Time
- Date-key
- day
- month
- year
- quarter
- day-of-week
- day-name

FIGURE 6.2 – Dimension Time

La table de dimension "Dim-vehicules" est conçue pour enregistrer les détails clés des véhicules, incluant leur identifiant unique, modèle, année de fabrication, type, kilométrage actuel, et statut opérationnel. Elle soutient ainsi la gestion efficace et la maintenance des véhicules au sein du système.

Dim-vehicules
- vehicules-id pk
- model
- year
- type
- mileage
- status

FIGURE 6.3 – Dimension vehicles

La table de dimension "Dim-task" est conçue pour stocker des informations essentielles sur les tâches, incluant leur identifiant unique, une description détaillée, l'entité ou l'utilisateur assigné, le type de tâche, et un indicateur d'état pour marquer si la tâche est terminée.

Dim-task
- task-id
- description
- task-for
- task-type
- done

FIGURE 6.4 – Dimension task

La figure 3.1 montre la conception du schéma de constellation



FIGURE 6.5 – Schéma de constellation

le schéma en constellation que nous avons élaboré structure efficacement les données de notre entrepôt grâce à des tables de faits et de dimensions bien définies. Cette architecture permet une analyse approfondie et multidimensionnelle des divers aspects de la gestion de flotte, notamment les tâches des chauffeurs, la maintenance des véhicules, les activités des mécaniciens et la gestion des utilisateurs. Elle offre une base solide pour le calcul des indicateurs de performance clés (KPI) et soutient les décisions stratégiques avec des données fiables, assurant une précision optimale dans le traitement des informations.

Dans les sections suivantes, nous explorerons en détail chaque datamart individuel, en commençant par le datamart dédié à l'analyse des chauffeurs. Nous examinerons les spécificités de chaque table de faits et de dimensions utilisées, ainsi que les KPI calculés,

6.5 Data mart : Chauffeur

L'objectif du data mart "Chauffeur" est de fournir une vue analytique détaillée sur les performances et les activités des chauffeurs. Il permet de suivre et d'analyser les tâches effectuées, les distances parcourues, et les interactions avec les véhicules.

6.5.1 Schéma conceptuel du DataMart « Chauffeur »

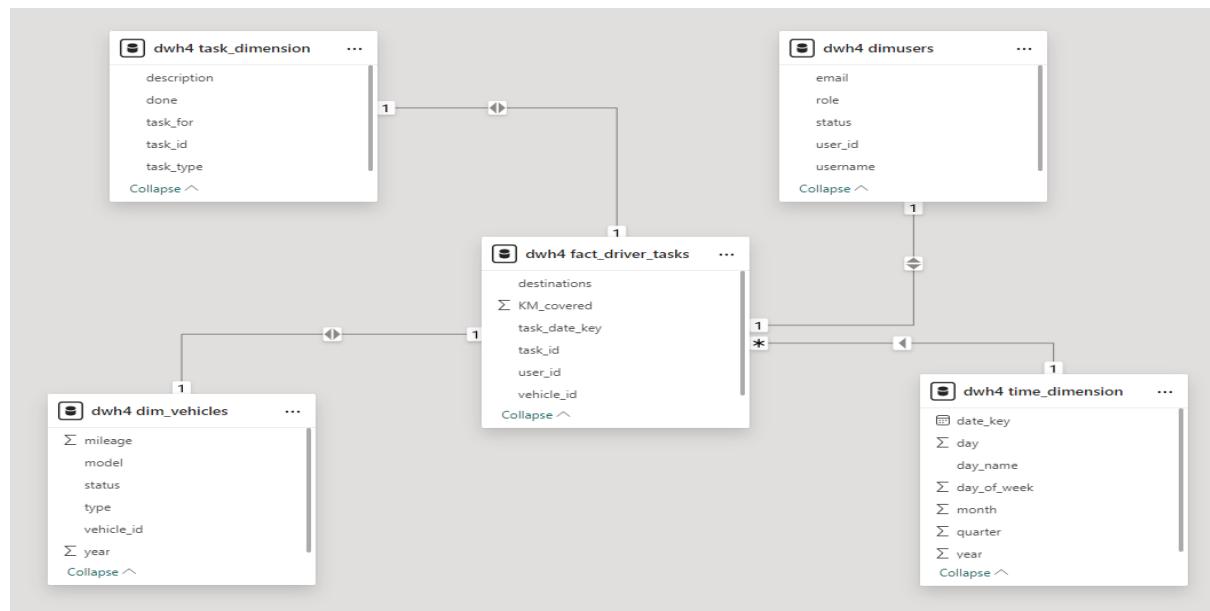


FIGURE 6.6 – Schéma en étoile : Chauffeur

Table de faits : Nous utilisons la table de faits **dwh4-driver-tasks-fact** pour suivre et analyser les statistiques et les métriques relatives aux tâches des chauffeurs, telles que les dates des tâches, les kilomètres parcourus et les destinations associées.

Tables de dimensions utilisées :

- La table de dimensions **dwh4-time-dimension** nous aide à organiser les données temporelles avec des attributs comme le jour, le mois et l'année.
- La table de dimensions **dwh4-dim-drivers** contient des informations sur les chauffeurs, telles que leur identifiant, leur nom d'utilisateur et leur type.
- La table de dimensions **dwh4-dim-vehicles** contient des informations sur les véhicules utilisés par les chauffeurs, comme le modèle, le kilométrage et le type de véhicule.
- La table de dimensions **dwh4-dim-destinations** contient des informations sur les destinations, avec des attributs tels que l'identifiant de la destination et le nom de la destination.

→ Avec ce datamart, nous pouvons calculer des KPI essentiels tels que le taux d'achèvement des tâches par chauffeur, les kilomètres parcourus par semaine, les tâches terminées par semaine

6.6 Data mart : Véhicules

L'objectif du data mart 'véhicules' est la gestion de la maintenance des véhicules pour garantir leur bon fonctionnement et prolonger leur durée de vie.

6.6.1 Schéma conceptuel du DataMart « Véhicules »

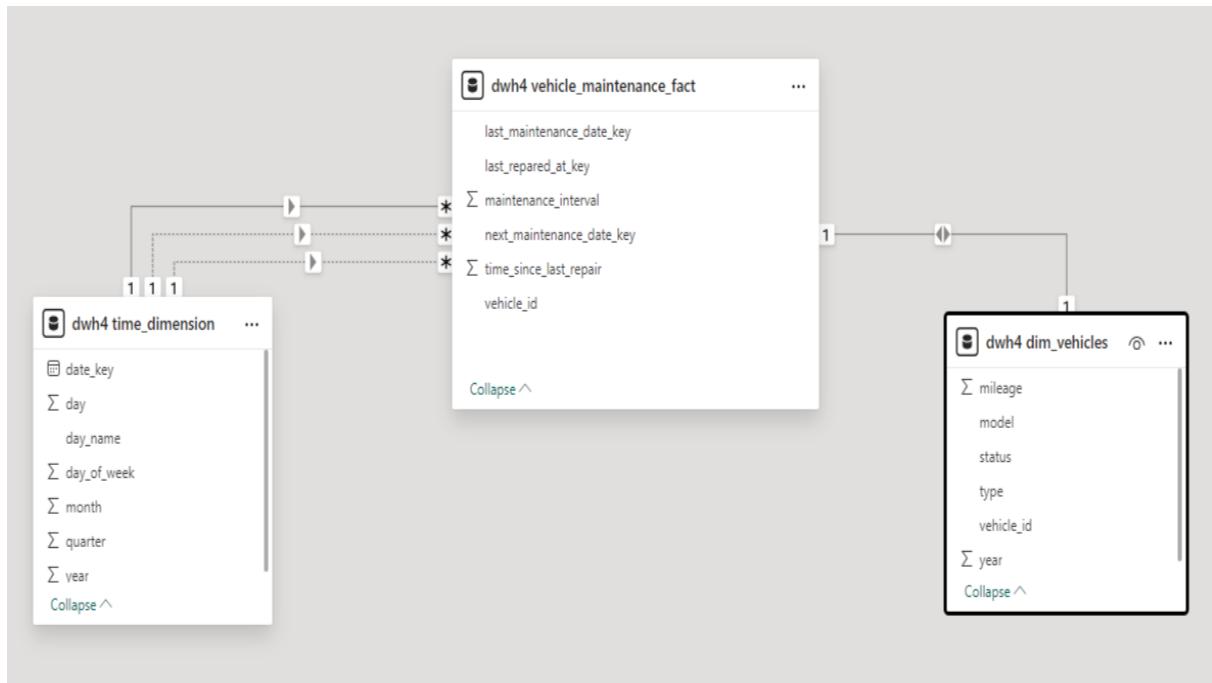


FIGURE 6.7 – Schéma en étoile : Véhicules

Table de faits : Nous utilisons la table de faits dwh4-vehicle-maintenance-fact pour suivre et analyser les statistiques et les métriques relatives à la maintenance des véhicules, telles que les dates de maintenance et de réparation, l'intervalle de maintenance et le temps écoulé depuis la dernière réparation.

Tables de dimensions utilisées :

- La table de dimensions **dwh4-time-dimension** nous aide à organiser les données temporelles avec des attributs comme le jour, le mois et l'année.
- La table de dimensions **dwh4-dim-vehicles** contient des informations sur les véhicules, comme -le modèle, le kilométrage et le type de véhicule.

→Avec ce datamart, nous pouvons calculer des KPI essentiels tels que le taux de réparation des véhicules par catégorie, les kilomètres parcourus par semaine, les kilomètres par véhicule (filtrés par matricule), et le taux de panne par modèle de véhicule.

6.7 Data mart : Mécanicien

L'objectif du data mart 'Mécanicien' est le suivi et optimisation de l'exécution des tâches mécaniques.

6.7.1 Schéma conceptuel du DataMart « Mécanicien »

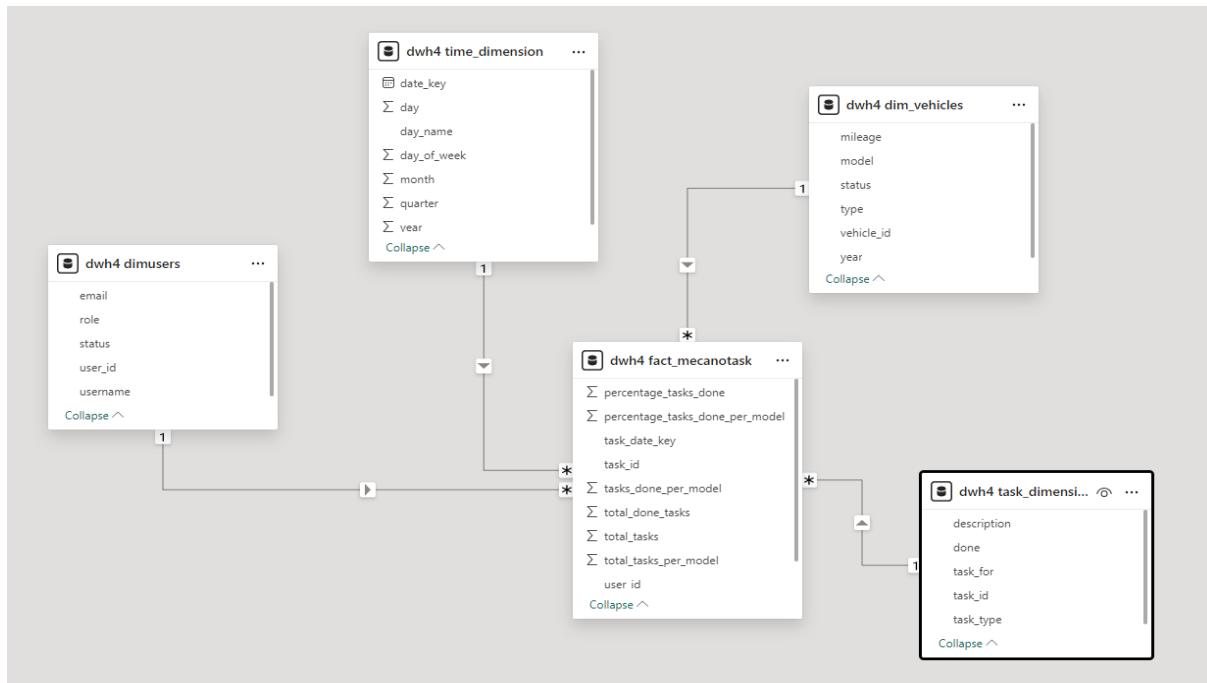


FIGURE 6.8 – Schéma en étoile : Mécanicien

Table de faits : Nous utilisons la table de faits **dwh4-fact-mecanotask** pour suivre et analyser les statistiques et les métriques relatives à l'exécution des tâches, telles que le pourcentage des tâches terminées et le nombre total de tâches terminées.

Tables de dimensions utilisées :

- La table de dimensions **dwh4-time-dimension** nous aide à organiser les données temporelles avec des attributs comme le jour, le mois et l'année.
- La table de dimensions **dwh4-dimusers** contient des informations détaillées sur les utilisateurs, telles que leur email, leur rôle et leur statut.
- La table de dimensions **dwh4-dim-vehicles** contient des informations sur les véhicules, comme le modèle, le kilométrage et le type de véhicule.
- La table de dimensions **dwh4-task-dimension** donne des détails sur les tâches, telles que la description, le type et pour qui la tâche est destinée.

→ Avec ce datamart, nous pouvons calculer des indicateurs essentiels comme le taux d'achèvement des tâches et le nombre de tâches terminées par semaine.

6.8 Data mart Utilisateurs

L'objectif du data mart des sinistres est le Suivi de la gestion des utilisateurs et de leur engagement.

Suivi de la gestion des utilisateurs et de leur engagement.

6.8.1 Schéma conceptuel du DataMart « Utilisateurs »

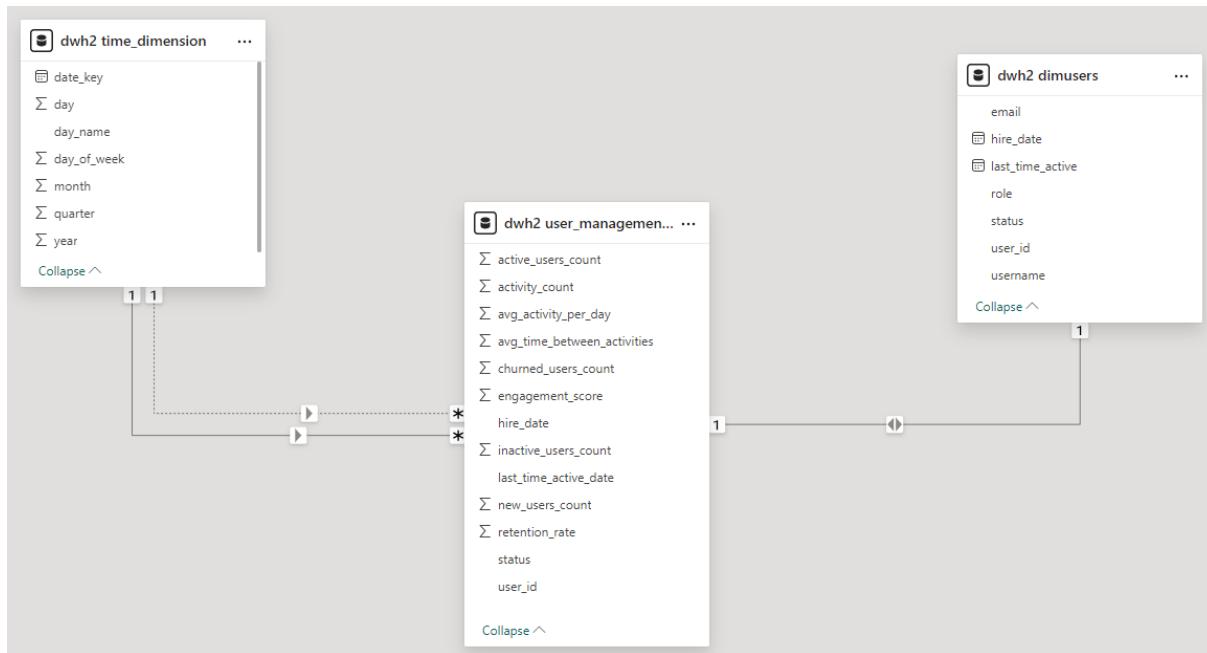


FIGURE 6.9 – Schéma en étoile : Utilisateurs

Table de faits : Nous utilisons la table de faits **dwh2-user-management** pour suivre les statistiques et les métriques relatives aux utilisateurs, comme le nombre d'utilisateurs actifs, les activités, et le taux de rétention.

Tables de dimensions utilisées :

- La table de dimensions **dwh2-time-dimension** nous aide à organiser les données temporelles avec des attributs comme le jour, le mois, et l'année.
- La table de dimensions **dwh2-dimusers** contient des informations détaillées sur les utilisateurs, telles que leur email, la date d'embauche, et leur rôle.

→Avec ce datamart, nous pouvons calculer des indicateurs essentiels tels que le nombre de nouveaux utilisateurs par période et le nombre total d'utilisateurs actifs par semaine.

6.9 Le développement de l'ETL

L'ETL est un processus crucial en gestion des données qui consiste à extraire des données de sources variées, les transformer selon les besoins analytiques de l'entreprise, et les charger dans une structure de stockage finale comme un entrepôt de données ou un lac de données. Ce processus supporte les activités d'analyse et de prise de décision en entreprise.

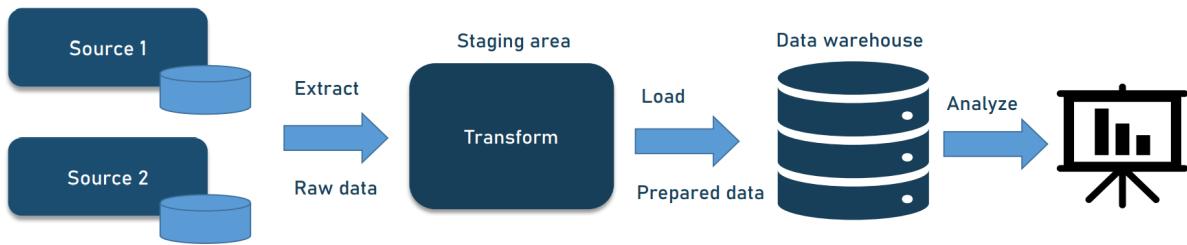
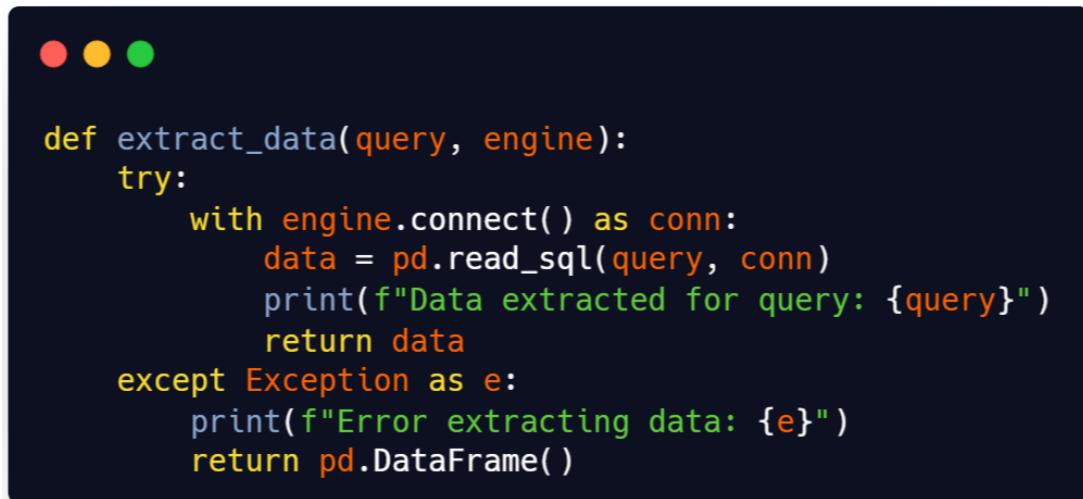


FIGURE 6.10 – Schéma du flux de travail ETL

- **Source** : La source de données d'où les données sont extraites.
- **Extract** : La phase d'extraction où les données brutes sont collectées.
- **Transform** : La phase de transformation où les données sont nettoyées et transformées.
- **Load** : La phase de chargement où les données transformées sont stockées dans le Data Warehouse.
- **Data Warehouse** : Le stockage centralisé des données prêtes pour l'analyse.
- **Analyze** : L'étape finale où les données sont analysées et utilisées pour générer des insights.

6.9.1 Extraction des données

L'extraction des données est la première étape du processus ETL. Le code ci-dessous montre une fonction en Python utilisée pour extraire des données d'une base de données à l'aide de la bibliothèque pandas et d'un moteur de base de données SQLAlchemy.



```
def extract_data(query, engine):
    try:
        with engine.connect() as conn:
            data = pd.read_sql(query, conn)
            print(f"Data extracted for query: {query}")
            return data
    except Exception as e:
        print(f"Error extracting data: {e}")
        return pd.DataFrame()
```

FIGURE 6.11 – Code Source : Extraction des données

Cette fonction, extract-data, prend en entrée une requête SQL et un moteur de base de données. Elle établit une connexion à la base de données, exécute la requête SQL et charge les résultats dans un DataFrame pandas. Cela permet de manipuler facilement les données pour les étapes suivantes. En cas d'erreur, la fonction capture l'exception et retourne un DataFrame vide, tout en affichant un message d'erreur pour faciliter le débogage.

6.9.2 Transformation des données

La phase de transformation du processus ETL consiste à nettoyer, enrichir et formater les données extraites pour les rendre cohérentes et prêtes à être chargées dans le Data Warehouse. Cela inclut la suppression des doublons, la correction des erreurs, la standardisation des formats et l'agrégation des données. Cette étape assure la qualité et l'intégrité des données pour les analyses futures.

Transformation du table de dimension 'users' : transform-dim-users

Cette fonction transforme les données des utilisateurs en renommant et en sélectionnant les colonnes nécessaires. Par exemple, elle renomme idusers en user-id, Mail en email et created-at en hire-date. Cette étape permet de standardiser les noms de colonnes pour une intégration cohérente dans le Data Warehouse.

```
● ● ●

def transform_dim_users(users):
    print("Transforming DimUsers...")
    return users[['idusers', 'Username', 'Mail', 'Role',
    'created_at']].rename(columns={
        'idusers': 'user_id',
        'Mail': 'email',
        'created_at': 'hire_date'
    })
```

FIGURE 6.12 – Transformation de la table de dimension 'users'

Transformation du table de fait Tâches du chauffeur : transform-fact-driver-tasks

Cette fonction enrichit les données des tâches des conducteurs en y intégrant les kilomètres parcourus à partir d'une autre source de données. Elle effectue une jointure et remplit les valeurs manquantes avec zéro, puis renomme et sélectionne les colonnes pour préparer les données pour le Data Warehouse.

```
● ● ●

def transform_fact_driver_tasks(driver_tasks, rehla):
    print("Transforming FactDriverTasks...")
    driver_tasks_with_km = driver_tasks.merge(
        rehla[['id_D', 'date', 'KM']],
        left_on=['id_driver', 'date'],
        right_on=['id_D', 'date'],
        how='left'
    )
    driver_tasks_with_km['KM'] =
    driver_tasks_with_km['KM'].fillna(0)
    fact_driver_tasks = driver_tasks_with_km[['idtask',
    'id_driver', 'matricule', 'date', 'KM']].rename(columns={
        'idtask': 'task_id',
        'id_driver': 'driver_id',
        'matricule': 'vehicle_id',
        'date': 'task_date',
        'KM': 'km_covered'
    })
    return fact_driver_tasks
```

FIGURE 6.13 – Transformation de la table de dimension 'users'

Ces exemples illustrent comment les données sont préparées, améliorées et structurées pour assurer leur cohérence et les rendre prêtes à être intégrées dans le Data Warehouse.

6.9.3 Chargement des données

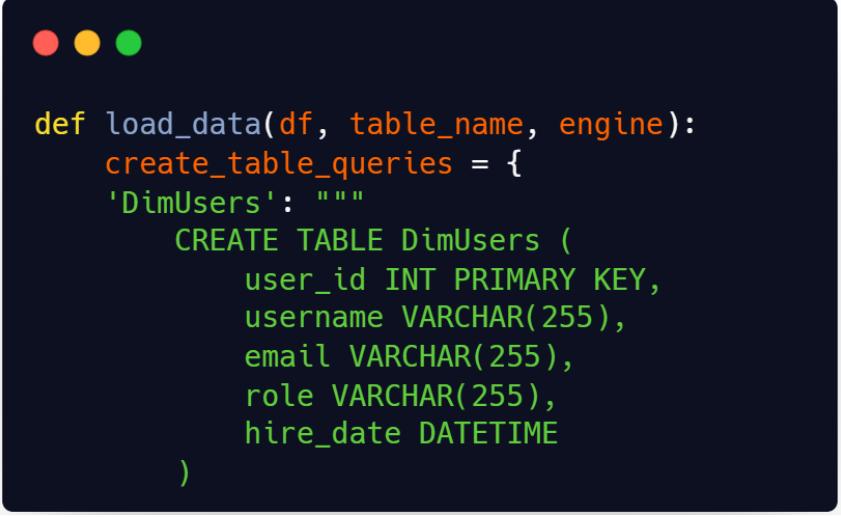
La dernière étape du processus ETL consiste à intégrer les données extraites et transformées dans le Data Warehouse.

La fonction load-data est définie avec trois paramètres : df (le DataFrame contenant les données à charger), table-name (le nom de la table cible) et engine (l'objet moteur de connexion à la base de données).

Requêtes de création de table : Un dictionnaire create-table-queries est défini pour contenir les requêtes SQL de création de tables pour chaque table cible. Ces requêtes définissent la structure de chaque table avec les colonnes et leurs types de données.

Chargement des Données dans la Table DimUsers :

-Insertion des Utilisateurs : Les données transformées des utilisateurs, généralement stockées dans un DataFrame (df), sont insérées dans la table DimUsers. Chaque enregistrement contient les champs comme user-id, username, email, role, et hire-date.



```
def load_data(df, table_name, engine):
    create_table_queries = {
        'DimUsers': """
            CREATE TABLE DimUsers (
                user_id INT PRIMARY KEY,
                username VARCHAR(255),
                email VARCHAR(255),
                role VARCHAR(255),
                hire_date DATETIME
            )
        """
    }
```

FIGURE 6.14 – Chargement des Données dans la Table DimUser

Chaque utilisateur est identifié par un user-id unique et les autres champs sont utilisés pour fournir des détails importants sur chaque utilisateur.

Chargement des Données dans la Table FactDriverTasks :

-Insertion des Tâches des chauffeurs : Les données transformées des tâches des conducteurs, généralement stockées dans un DataFrame (df), sont insérées dans la table FactDriverTasks. Chaque enregistrement contient des champs comme task-id, driver-id, vehicle-id, task-date, et km-covered.



```
'FactDriverTasks': """
    CREATE TABLE FactDriverTasks (
        task_id VARCHAR(255) PRIMARY
    KEY,
        driver_id INT,
        vehicle_id VARCHAR(255),
        task_date DATETIME,
        km_covered FLOAT
    )
"""

```

FIGURE 6.15 – Chargement des Données dans la Table FactDriverTasks :

Chaque tâche est identifiée par un task-id unique et les autres champs capturent des informations cruciales telles que le conducteur (driver-id), le véhicule (vehicle-id), la date de la tâche (task-date), et les kilomètres parcourus (km-covered).

6.10 Choix des graphiques

Nivo Chart est une bibliothèque de visualisation de données basée sur React, conçue pour créer des graphiques interactifs et réactifs. Elle offre une large gamme de types de graphiques, y compris des barres, des lignes, des pie charts, des treemaps, des heatmaps, et bien d'autres. En utilisant Nivo Chart, vous pouvez facilement créer des visualisations qui non seulement présentent les données de manière claire et attrayante, mais aussi facilitent l'interaction et l'analyse des informations. [7]



FIGURE 6.16 – Nivo Chart

Avec l'aide de Nivo Chart, nous avons opté pour les types de graphiques suivants pour représenter les différents indicateurs dans notre rapport :

Graphique en barres (Bar Chart) : Idéal pour comparer des valeurs discrètes, notamment pour montrer le taux d'achèvement des tâches et le nombre de tâches terminées par semaine. Leur simplicité et leur lisibilité sont parfaites pour des comparaisons claires.

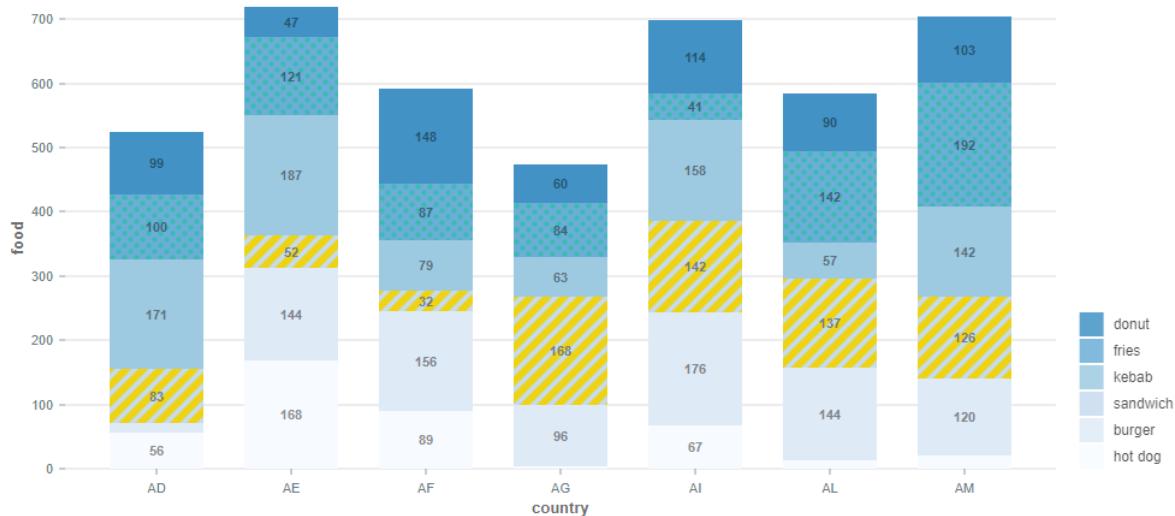


FIGURE 6.17 – Graphique en barres

Graphique linéaire (Line Chart) : Permet de visualiser des tendances sur une période de temps. Utilisé pour montrer l'évolution des kilomètres parcourus par semaine, mettant en évidence les variations et les tendances continues.

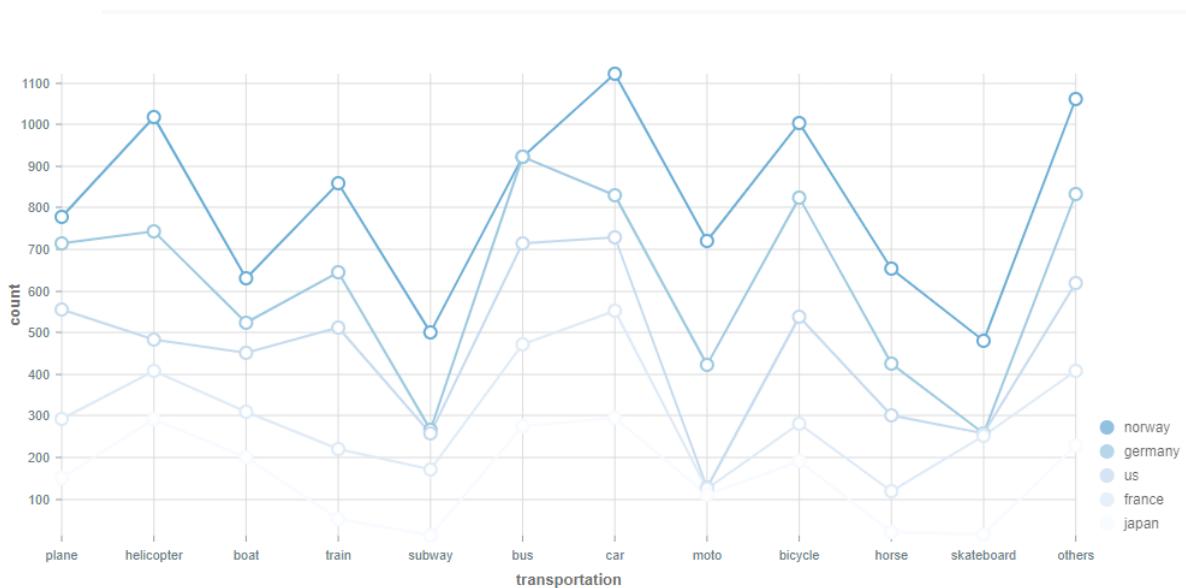


FIGURE 6.18 – Graphique linéaire

Graphique en secteurs (Pie Chart) : Montre les proportions de différentes catégories dans un ensemble. Utile pour visualiser la répartition des types de tâches ou des pannes par catégorie de véhicule.

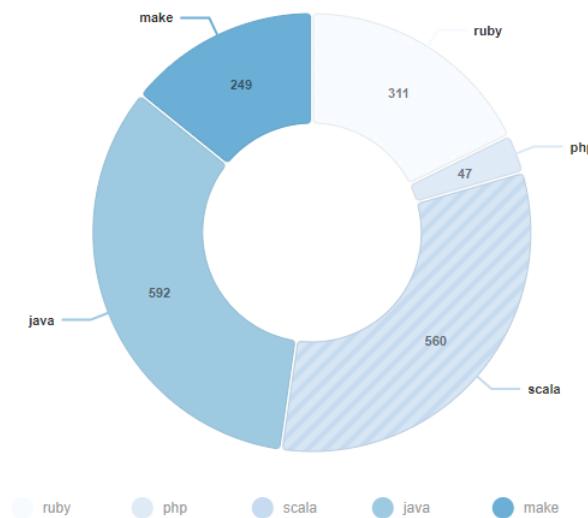


FIGURE 6.19 – Graphique en secteurs

Heatmap : Excellente pour montrer la densité ou la fréquence d'événements dans un espace défini. Particulièrement utile pour représenter la fréquence des réparations ou l'utilisation des véhicules sur différentes périodes.



FIGURE 6.20 – Heatmap

En utilisant ces différents types de graphiques, nous pouvons fournir une vue d'ensemble complète et intuitive des performances et des indicateurs clés, facilitant ainsi la prise de décision basée sur les données.

6.11 Publication du tableau de bord

6.11.1 Présentation des tableaux de bord dans l'application web

6.11.2 Présentation des tableaux de bord dans l'application mobile

Conclusion

Ce chapitre a décrit l'implémentation des tableaux de bord durant le sprint 4, en se concentrant sur les besoins des différents utilisateurs. Le choix d'une architecture de données adaptée a permis une gestion efficace des données et des visualisations personnalisées pour chaque profil. Les tableaux de bord créés facilitent la prise de décisions éclairées grâce à des KPI pertinents. En résumé, ce sprint a abouti à des outils analytiques robustes et essentiels pour l'amélioration continue des opérations.

Conclusion générale

Ce projet représente une avancée majeure dans la gestion des données et l'analyse au sein de notre organisation. À travers plusieurs sprints bien planifiés et exécutés, nous avons réussi à développer et à implémenter des tableaux de bord dynamiques et personnalisés, répondant aux besoins spécifiques de divers utilisateurs, allant des administrateurs aux mécaniciens.

L'un des points forts de ce projet a été la définition précise des besoins des utilisateurs via des user stories détaillées. Cela a permis de concevoir des solutions sur mesure qui optimisent la prise de décision et la gestion opérationnelle. Les tableaux de bord créés fournissent des indicateurs clés de performance (KPI) pertinents et facilement accessibles, permettant une vue d'ensemble claire et précise des opérations.

Le choix de l'architecture de données a également joué un rôle crucial dans le succès du projet. L'approche adoptée a permis une gestion efficace et fluide des données, assurant leur intégrité et leur disponibilité pour des analyses en temps réel. Les processus d'ETL (Extraction, Transformation, Chargement) ont été méticuleusement conçus pour garantir que les données alimentant les tableaux de bord sont toujours à jour et exactes.

En outre, ce projet a mis en évidence l'importance de la collaboration interfonctionnelle et de la communication continue entre les différentes équipes. Chaque sprint a été une occasion de recueillir des retours, d'ajuster les fonctionnalités et de s'assurer que les solutions développées répondaient parfaitement aux attentes des utilisateurs finaux.

En conclusion, ce projet a non seulement amélioré la capacité de notre organisation à gérer et analyser les données, mais a également renforcé notre culture de prise de décision basée sur des données fiables et pertinentes. Les outils analytiques robustes développés au cours de ce projet continueront à soutenir l'amélioration continue des opérations, contribuant ainsi à la croissance et à l'efficacité de l'organisation.

Webographie

[1] Organigramme de Digital identity

<https://digidco.com/> (consulté le 22/02/2024)

[2] Les activités de Digital Identity

<https://digidco.com/> (consulté le 22/02/2024)

[3] méthodologie Scrum

<https://www.sketchbubble.com/en/powerpoint-scrum-process.html> (consulté le 06/03/2024)

[4] Définition des indicateurs de performance

<https://www.tableau.com/fr-fr/learn/articles/how-to-choose-your-kpi#:~:text=Les%20indicateurs%20cl%C3%A9s%20performance%2C%20ou,la%20strat%C3%A9gie%20et%20l'op%C3%A9rationnel.> (consulté le 27/05/2024)

[5] Schéma du flux de travail ETL

<https://www.altexsoft.com/blog/data-pipeline-components-and-types/> (consulté le 01/06/2024)

[6] MVVM

<https://www.arkance-systems.fr/blog/pattern-mvvm/#:~:text=D%C3%A9finition,li%C3%A9s%20%C3%A0%20la%20logique%20m%C3%A9tier.> (consulté le 22/03/2024)

[7] Nivo chart

<https://nivo.rocks/> (consulté le 03/06/2024)

Resumé du projet

Français :

Le projet de fin d'études vise à concevoir et développer une application de gestion de parc, accompagnée d'un tableau de bord interactif. Cette application, développée pour l'organisme Digital Identity, a pour objectif principal d'améliorer la gestion des ressources et des équipements. Elle permet de surveiller en temps réel l'état des différentes ressources du parc, d'optimiser la planification des interventions de maintenance, et de fournir des rapports analytiques détaillés. Ces rapports aident à la prise de décisions stratégiques en fournissant des données précises et actualisées. Le projet adopte une méthodologie agile, en particulier le framework Scrum, ce qui permet une gestion de projet flexible et adaptative. Les langages de programmation utilisés et l'infrastructure matérielle choisie sont adaptés pour garantir la performance et la scalabilité de l'application.

English :

The final year project aims to design and develop a fleet management application, equipped with an interactive dashboard. This application, created for the organization Digital Identity, has the primary goal of enhancing the management of resources and equipment. It enables real-time monitoring of the status of various resources, optimizes the scheduling of maintenance interventions, and provides detailed analytical reports. These reports aid in strategic decision-making by offering precise and up-to-date data. The project employs an agile methodology, specifically the Scrum framework, which allows for flexible and adaptive project management. The programming languages used and the chosen hardware infrastructure are suited to ensure the application's performance and scalability.