**République Algérienne Démocratique et Populaire**
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**
**Université M'Hamed Bougara – Boumerdès**




Faculté des Sciences
Département d'Informatique

**Domaine** : Mathématiques Informatique
**Filière** : Informatique
**Spécialité** : Systemes informatiques
*N° de l'Arrêté d'habilitation de la spécialité : arrêté n °872 du 26/07/2016*

**Mémoire de fin d'études en vue de l'obtention du Diplôme de Licence Académique**

# *Thème*

# Développement d'une application mobile dédié à la gestion de réservation de véhicules

**Présenté par :**
Sellali Djoumana
Younes Syrine
Abed Chahd

**Stage Pratique réalisé au : Sonatrach-Exploration**

**Soutenu le 04/06/2026 Devant le jury composé de :**

| | | |
|---|---|---|
| Rezki Mohamed Amine | : | Examinateur |
| Baroudi Mohammed Yassine | : | Encadreur |
| Hadj Arab Massinissa | : | promoteur (thèmes externes) |

# Remerciement

*Nous rendons **Grâce à Allah**, tout puissant et miséricordieux, de nous avoir donné le courage, la volonté et la patience pour achever ce modeste projet et surmonter tous les obstacles.*

*Nous tenons tout particulièrement à remercier **M. Baroudi Mohammed Yassin** notre encadreur interne de ce mémoire, pour avoir accepté de nous encadrer et pour le précieux temps qu'il nous a consacré afin de nous orienter et nous conseiller. Sa disponibilité, sa rigueur et ses précieux conseils ont été d'une grande aide tout au long de la réalisation de ce projet. Nous tenons également à remercier **M. Hadj Arab Massinissa***

*promoteur externe de SONATRACH, pour sa coopération, son accompagnement et ses précieux conseils qui ont grandement contribué à l'enrichissement de ce travail.*

*Nos vifs remerciements vont à **M. Rezki Mohamed Amine** notre examinateur, qui a honoré ce travail en l'évaluant avec attention et bienveillance.*

*Nous tenons à remercier chaleureusement**M. Hassani Zaid** pour son professionnalisme et son soutien tout au long de nos années d'études.*

*Enfin, nous adressons nos sincères remerciements à la Faculté des Sciences de**l'Université M'Hamed Bougara - Boumerdes** pour le cadre d'apprentissage qu'elle nous a offert durant nos trois années d'études.*

# Dédicace

*Je dédie ce modeste travail, fruit de longues années d'efforts, de patience et de persévérance, à ceux qui ont toujours illuminé mon chemin par leur amour, leur présence et leurs encouragements.*

***À mon très cher père,***
*aucun mot ne saurait exprimer toute ma reconnaissance pour les sacrifices que tu as consentis afin de me voir réussir.*
*Tu as été ma force dans les moments difficiles, mon soutien dans les instants de doute et la lumière qui a guidé chacun de mes pas.*
*Que Dieu te protège et t'accorde santé et bonheur.*
*Ce travail est avant tout le tien.*

***À ma chère mère,***
*pour ton amour, tes sacrifices, tes prières et ton soutien tout au long de mon parcours.*
*Que Dieu te protège et te garde auprès de moi.*
*Ce travail t'est dédié avec tout mon amour.*

***À mon cher frère Chakib,***
*merci pour ton soutien, tes conseils, ta présence et tous les moments où tu as su me redonner confiance.*
*Ta présence dans ma vie est une immense bénédiction et une source de fierté pour moi.*

***À ma chère sœur Hadil,***
*pour toute la tendresse, l'amour et la joie que tu apportes dans ma vie.*
*Merci d'avoir toujours cru en moi et d'avoir été à mes côtés dans chaque étape de mon parcours.*

***À ma binôme Syrine,***
*merci pour ta patience, ton sérieux, ta compréhension et tous les efforts partagés durant ce travail.*
*Les moments de stress, de fatigue et de réussite resteront gravés dans ma mémoire.*
*Je te souhaite beaucoup de succès et de bonheur dans ton avenir.*

**Sellali Djoumana**

# Dédicace

*À moi-même, avant tout... À celle qui n'a jamais renoncé, même dans les moments les plus difficiles. À celle qui a affronté la fatigue, les nuits blanches, les doutes et les obstacles avec courage et persévérance. Je me dédie ce travail, reflet de mes efforts, de ma patience et de ma détermination. Aujourd'hui, je suis fière du chemin parcouru et de tout ce que j'ai accompli.*

***À ma très chère maman,***
*Mon premier repère, ma source infinie de tendresse et mon plus bel exemple de force. Merci pour ton amour inconditionnel, tes sacrifices silencieux et tes prières qui ont toujours illuminé mon chemin. Tu as été mon refuge dans chaque moment de faiblesse et ma plus grande motivation pour avancer.*

***À mon précieux papa,*** *Toi qui as tant donné et tant sacrifié pour notre bonheur et notre réussite. Merci pour ton soutien sans faille, ta confiance et tous les efforts que tu as consentis pour me voir réussir. Tu resteras toujours ce pilier solide sur lequel je peux m'appuyer avec fierté et sécurité.*

***À mon cher frère Walid et à ma tendre sœur Chaima,*** *Merci pour votre présence, votre affection et votre soutien constant. Votre amour et vos encouragements ont rendu ce parcours plus beau et plus léger.*

***À notre petite princesse Rahil,***
*La nouvelle lumière de notre famille. Ton arrivée a rempli nos cœurs d'une immense joie et apporté un souffle de bonheur et d'espoir. Que ton sourire illumine toujours nos vies*

*À toutes les personnes qui ont cru en moi, Merci pour chaque mot encourageant, chaque geste sincère et chaque présence bienveillante qui m'ont aidée à avancer et à ne jamais perdre confiance.*
*Enfin, à* ***ma chère amie et binôme Joumana*** *Merci d'avoir partagé avec moi cette aventure de fin d'études avec patience, courage et complicité. Merci pour tous les moments de stress, de rire, de doute et de motivation que nous avons traversés ensemble. Ce parcours restera un souvenir précieux gravé dans ma mémoire.*

*Je dédie ce travail à tous ceux qui occupent une place particulière dans mon cœur.*

**Younes Syrine**

# Table des matières

**Introduction Générale** **i**

**1 Présentation de l'organisme d'accueil** **1**
1.1 Présentation Générale . . . . . 1
1.2 Contexte institutionnel . . . . . 2
1.2.1 Présentation générale de Sonatrach . . . . . 2
1.2.2 Organisation structurelle . . . . . 2
1.3 Présentation de l'unité d'accueil : Division Exploration – Boumerdès . . . . 3
1.3.1 Rôle stratégique de la division . . . . . 3
1.3.2 Positionnement de la Direction Data Management . . . . . 4
1.4 Analyse organisationnelle du Département Technologie de l'Information . . 5
1.5 Délimitation du périmètre d'étude . . . . . 5
1.6 Conclusion . . . . . 6

**2 Étude de l'existant** **7**
2.1 Introduction . . . . . 7
2.2 Gestion de réservation . . . . . 7
2.2.1 Pourquoi la gestion de réservation . . . . . 7
2.3 Étude de poste de travail . . . . . 8
2.4 Description du projet . . . . . 8
2.4.1 Problématique . . . . . 8
2.4.2 Objectifs . . . . . 9
2.5 Conclusion . . . . . 9

**3 Analyse et conception** **10**
3.1 Introduction . . . . . 10
3.2 Présentation d'UML . . . . . 10
3.3 Justification de l'utilisation d'UML . . . . . 10
3.4 Spécification des besoins . . . . . 10
3.4.1 Les besoins fonctionnels . . . . . 11
3.4.2 Les besoins non fonctionnels . . . . . 11
3.5 Diagramme de Cas d'utilisation . . . . . 11
3.5.1 Diagramme de cas d'utilisation générale du système . . . . . 11
3.5.2 Diagramme de cas d'utilisation : chef de département . . . . . 12
3.5.3 Diagramme de cas d'utilisation : chef de parc . . . . . 12
3.5.4 Diagramme de cas d'utilisation : employé . . . . . 13
3.5.5 Diagramme de cas d'utilisation :chauffeur . . . . . 13
3.6 Elaboration de diagramme de classe : . . . . . 14
3.6.1 Diagramme de classe de système : . . . . . 14
3.6.2 Règles de gestion : . . . . . 15
3.7 Diagramme de séquences : . . . . . 16

3.7.1 Diagramme de séquence : "Authentification" . . . . . 17
3.7.2 Diagramme de séquence : "consulter mission" . . . . . 18
3.7.3 Diagramme de séquence : "Demande de réservation" . . . . . 18
3.7.4 Diagramme de séquence : "Traiter la demande" . . . . . 19
3.7.5 Diagramme de séquence : "l'affectation des chauffeurs et des véhicules" 20
3.8 Conclusion . . . . . 20

**4 Implementation** **21**
4.1 Introduction . . . . . 21
4.2 L'architecture MVVM . . . . . 21
4.3 Environnement de développement : . . . . . 21
4.4 Environnement matériel . . . . . 21
4.5 Environnement logiciel : . . . . . 22
4.6 présentation des interfaces graphiques : . . . . . 24
4.6.1 Les interfaces d'authentification : . . . . . 24
4.6.2 Les interfaces chef de parc : . . . . . 25
4.6.3 Les interfaces de chef département : . . . . . 29
4.6.4 Les interfaces de l'employé : . . . . . 31
4.6.5 Les interfaces de chauffeur : . . . . . 34

**Conclusion Générale** **37**
4.7 Perspectives . . . . . 37

# Table des figures


1.1 Logo de Sonatrach. . . . . 2
1.2 Organigramme de la macrostructure de SONATRACH . . . . . 3
1.3 L'organigramme de la division . . . . . 4
1.4 Organigramme de la direction Data Management. . . . . 5

3.1 Diagramme de cas d'utilisation générale . . . . . 12
3.2 Diagramme de cas d'utilisation chef de département . . . . . 12
3.3 Diagramme de cas d'utilisation chef de parc . . . . . 13
3.4 Diagramme de cas d'utilisation employé . . . . . 13
3.5 Diagramme de cas d'utilisation chauffeur . . . . . 14
3.6 Diagramme de classe de système . . . . . 15
3.7 Diagramme de séquence d'authentification . . . . . 17
3.8 Diagramme de séquence associe a la consultation des missions . . . . . 18
3.9 Diagramme de séquence associe a la demande de réservation . . . . . 18
3.10 Diagramme de séquence associe au traitement des demandes par le chef de département . . . . . 19
3.11 Diagramme de séquence associe a l'affectation des chauffeurs et des véhicules 20

4.2 l'interface d'authentification . . . . . 25
4.6 les interfaces du chef de parc . . . . . 29
4.7 Interfaces du chef département . . . . . 30
4.9 les Interfaces de l'Espace Employé . . . . . 32
4.11 les interfaces d'employe . . . . . 34
4.13 Les interfaces du chauffeur . . . . . 36

# Introduction Générale

Dans un contexte marqué par la transformation numérique et l'évolution rapide des technologies de l'information, les organisations sont de plus en plus amenées à moderniser leurs systèmes de gestion afin d'améliorer leur performance et leur efficacité opérationnelle. La digitalisation des processus métiers constitue aujourd'hui un levier essentiel permettant d'optimiser l'utilisation des ressources, de faciliter la communication entre les acteurs et de garantir une meilleure traçabilité des opérations.

Dans ce cadre, notre projet de fin d'études s'inscrit dans une démarche de conception et de développement d'une application mobile dédiée à la gestion des réservations de véhicules au sein de la division Exploration de Sonatrach. En effet, malgré l'existence d'un processus organisationnel structuré, la gestion actuelle repose en grande partie sur des méthodes manuelles, ce qui engendre plusieurs contraintes telles que le manque de visibilité, les risques d'erreurs ainsi que des délais de traitement relativement longs.

L'objectif principal de ce travail est donc de proposer une solution informatique moderne permettant de faciliter et d'optimiser le processus de réservation des véhicules, tout en améliorant la coordination entre les différents intervenants (employés, chefs de département, chefs de parc et chauffeurs).

Afin de mener à bien ce projet, notre démarche s'est articulée autour de quatre chapitres principaux. Le premier chapitre est consacré à la présentation de l'organisme d'accueil ainsi que du contexte général du projet. Le deuxième chapitre porte sur l'étude de l'existant, permettant d'analyser le fonctionnement actuel et d'identifier ses limites. Le troisième chapitre est dédié à l'analyse et à la conception de la solution proposée à travers l'utilisation du langage UML. Enfin, le dernier chapitre présente l'implémentation de l'application, en détaillant les outils, les technologies utilisées ainsi que les différentes interfaces réalisées.

# Chapitre 1

Présentation de l'organisme d'accueil

# Chapitre 1

# Présentation de l'organisme d'accueil

## 1.1 Présentation Générale

Ce chapitre marque le début de notre approche méthodologique. Il s'agit de définir le contexte organisationnel du projet, d'examiner l'environnement actuel et de déterminer les exigences fonctionnelles et non fonctionnelles du système à réaliser.

Nous adoptons les étapes suivantes dans notre démarche :

— Introduction au cadre institutionnel.
— Définition de la portée de l'étude.
— Étude de l'état actuel.
— Identification et détermination des besoins.

Cette organisation favorise une appréhension graduelle et méthodique du problème examiné.

## 1.2 Contexte institutionnel

### 1.2.1 Présentation générale de Sonatrach




FIGURE 1.1 – Logo de Sonatrach.

Sonatrach est l'entreprise nationale algérienne dédiée à l'exploration, la production, le transport, la transformation et la vente d'hydrocarbures. Établie le 31 décembre 1963, elle se positionne comme un acteur crucial dans l'économie du pays et joue un rôle prépondérant dans le domaine de l'énergie à l'échelle africaine et mondiale.

Sonatrach opère sur toute la chaîne de valeur des hydrocarbures, englobant les activités en amont (exploration et production) ainsi qu'en aval (raffinage, pétrochimie et commercialisation). Elle est également impliquée dans des initiatives concernant les énergies renouvelables et la transition vers une énergie durable.

En raison de son envergure économique et organisationnelle, Sonatrach incarne un contexte complexe défini par :

— Une structure organisationnelle centralisée
— Un éventail d'activités diversifiées
— Une forte reliance sur les systèmes d'information
— Une administration massive et stratégique des données techniques.

### 1.2.2 Organisation structurelle

La structure organisationnelle de Sonatrach est basée sur une gouvernance qui repose sur :

— Une direction générale sous la responsabilité du Président-Directeur Général .
— De départements d'entreprise responsables des fonctions stratégiques (finances, stratégie, ressources humaines, numérisation)
— De directions centrales en charge de la gestion opérationnelle

— D'activités sectorielles spécialisées (exploration-production, transport, liquéfaction, raffinage, commercialisation).

Cette structure organisationnelle assure la concordance entre la stratégie globale du groupe et ses opérations quotidiennes.

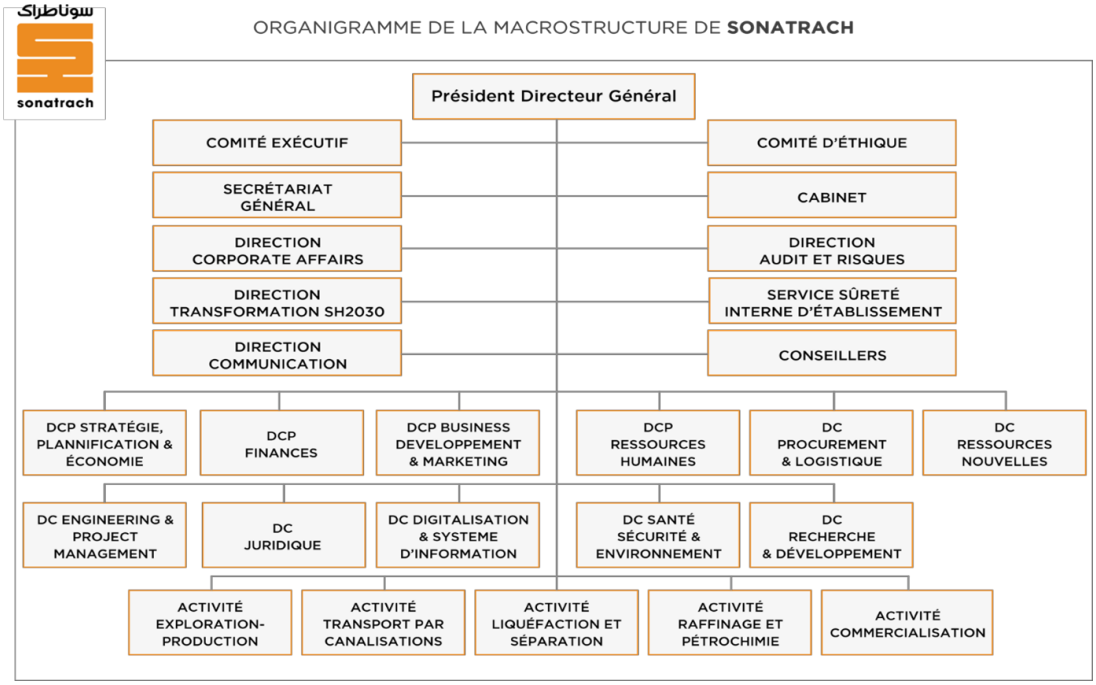



FIGURE 1.2 – Organigramme de la macrostructure de SONATRACH

## 1.3 Présentation de l'unité d'accueil : Division Exploration – Boumerdès

### 1.3.1 Rôle stratégique de la division

La mission principale de la Division Exploration est de garantir le renouvellement et l'expansion des réserves nationales d'hydrocarbures. Elle supervise les opérations de recherche, d'étude géologique et de traitement géophysique.

Ses tâches essentielles comprennent :

— La conception et la mise en application des programmes d'exploration
— Effectuer des recherches géologiques et géophysiques
— Contribuer aux initiatives d'exploration à l'échelle nationale et internationale
— Renforcer l'expertise scientifique et technique dans le secteur pétrolier.

Cette division sert de pôle stratégique pour la production et l'analyse des données techniques.

L'organigramme de la division est présenté par la figure ci-dessous :

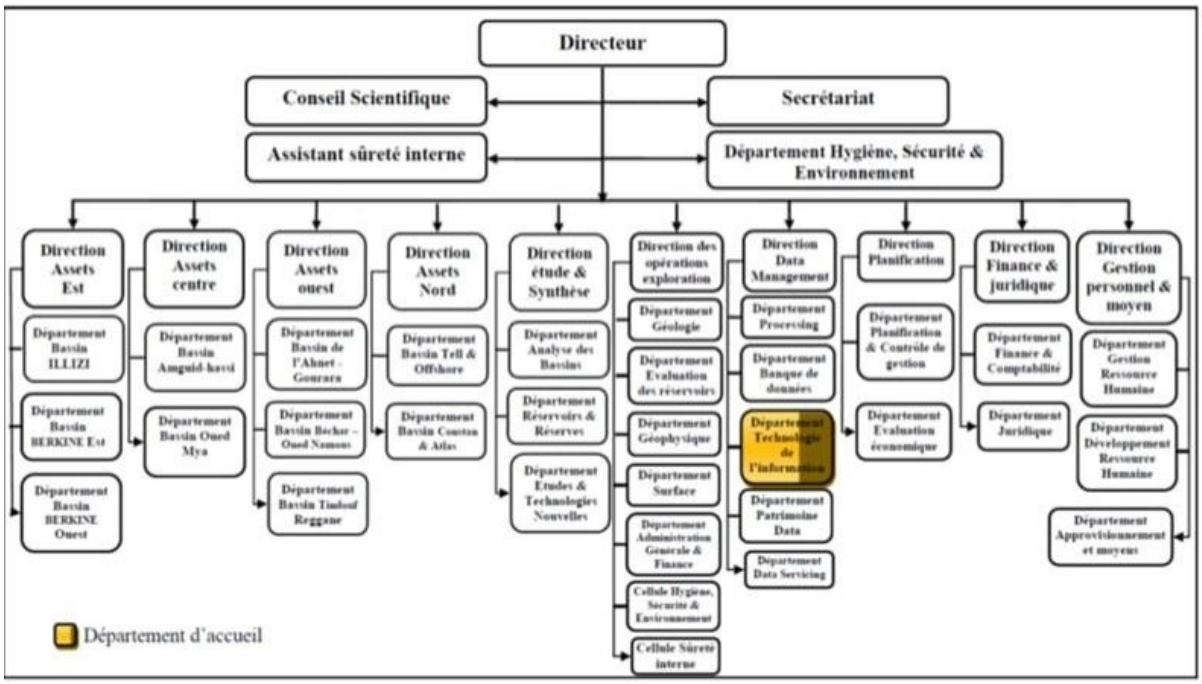



FIGURE 1.3 – L'organigramme de la division

### 1.3.2 Positionnement de la Direction Data Management

Dans la Division Exploration, le Service de Gestion des Données joue un rôle crucial dans l'organisation, la structuration et l'exploitation des données.

Elle est notamment responsable de :

— La gestion des bases de données techniques
— L'analyse et la réanalyse des données sismiques
— L'administration des systèmes d'information
— La protection des infrastructures informatiques
— La préservation du patrimoine informationnel.

Son opération est basée sur divers départements spécialisés, y compris le Département de Technologie de l'Information, qui représente le noyau du soutien informatique. L'organigramme de la direction Data Management est présenté par la figure ci-dessous :

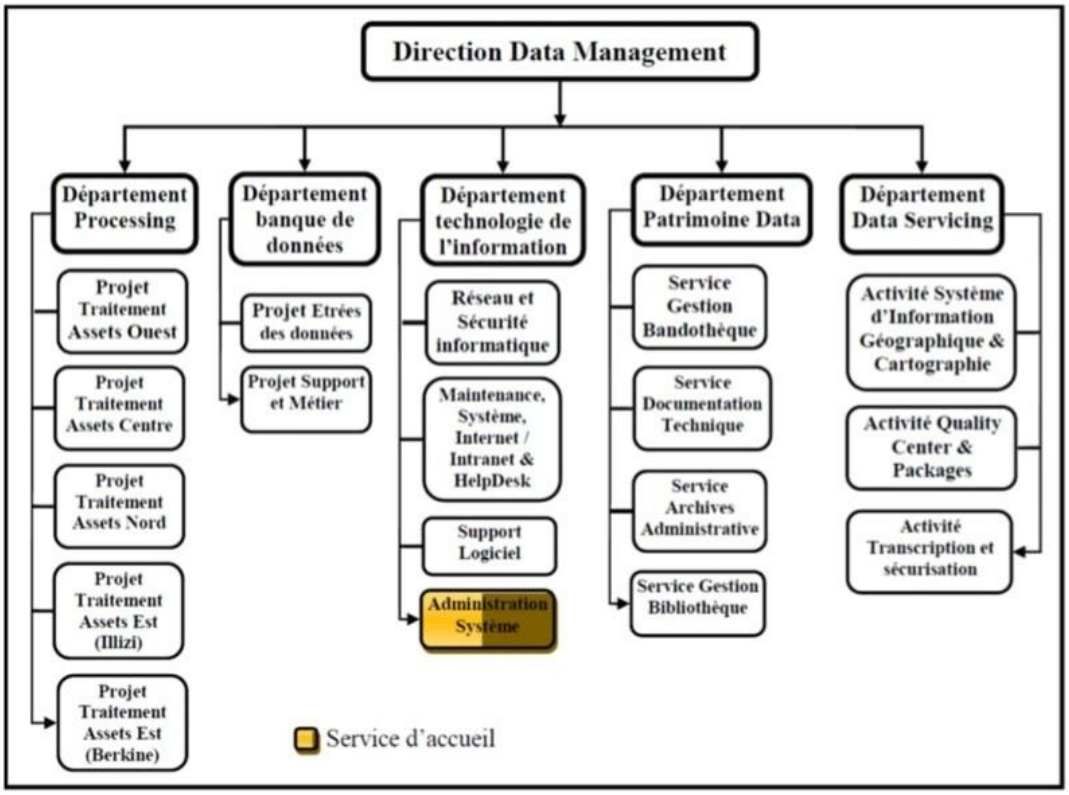



FIGURE 1.4 – Organigramme de la direction Data Management.

## 1.4 Analyse organisationnelle du Département Technologie de l'Information

C'est le Département de Technologie de l'Information qui se charge de la gestion et du maintien en état opérationnel des ressources numériques de la division.

Ses tâches principales comprennent :

— Gestion des systèmes et des réseaux
— Administration des équipements et des programmes informatiques
— Mise en application de la politique de sécurité numérique
— Support technique pour les utilisateurs
— Gestion des plateformes spécialisées en géologie et géophysique

Ce département constitue le cadre immédiat dans lequel s'intègre notre projet.

## 1.5 Délimitation du périmètre d'étude

Pour garantir la cohérence de l'étude, le champ d'application est restreint aux processus associés à la gestion et au traitement des données au sein de la Direction Data Management, spécifiquement dans le Département Technologie de l'Information.

L'analyse se concentrera sur les points suivants :

— Les courants d'informations
— Les échanges entre les intervenants
— Les techniques actuellement en place
— Les contraintes du système actuel.

## 1.6 Conclusion

Ce chapitre a permis de contextualiser le projet dans son cadre organisationnel et stratégique. L'analyse graduelle, qui s'étend du contexte général de Sonatrach jusqu'à l'unité opérationnelle spécifique, a facilité la compréhension de l'environnement du système examiné. Ceci représente une étape cruciale pour commencer, dans le prochain chapitre, l'étude approfondie de la situation actuelle et la définition précise des besoins selon une approche méthodique structurée.

# Chapitre 2

# Étude de l'existant

## 2.1 Introduction

Dans le cadre de la réalisation de notre application mobile dédiée à la gestion des réservations de véhicules au sein de Sonatrach, il est essentiel d'effectuer une étude de l'existant afin de comprendre les méthodes actuellement utilisées.

Cette analyse permet d'identifier les différents acteurs impliqués dans le processus, d'étudier le fonctionnement actuel de la gestion des réservations, ainsi que de mettre en évidence les insuffisances du système en place.

L'objectif de ce chapitre est donc de présenter une vision claire du système actuel, qui servira de base pour la conception d'une solution informatique adaptée et performante.

## 2.2 Gestion de réservation

La gestion des réservations de véhicules s'organise autour d'un processus hiérarchique faisant intervenir plusieurs acteurs. L'employé, selon ses besoins professionnels, initie une demande de réservation. Celle-ci est soumise à l'approbation du chef de département, qui l'examine et statue sur sa validation. En cas de refus, la procédure est interrompue. Si la demande est acceptée, elle est transmise au chef de parc, garant des ressources matérielles et humaines. Ce dernier évalue la disponibilité des véhicules et des chauffeurs, puis affecte un véhicule et un chauffeur à la mission. Une fois ces affectations réalisées, la mission est officiellement validée et confiée au chauffeur désigné. Toutefois, ce circuit, bien que structuré, repose encore largement sur des opérations manuelles, ce qui peut entraîner des difficultés de gestion et de suivi.

### 2.2.1 Pourquoi la gestion de réservation

La gestion des réservations occupe une place centrale au sein des organisations disposant de ressources limitées, notamment les véhicules de service. Une gestion rigoureuse permet non seulement d'améliorer l'organisation des déplacements professionnels, mais aussi d'optimiser l'utilisation des ressources disponibles et de réduire les temps morts.

Dans le cadre spécifique de Sonatrach, où les missions exigent une coordination étroite entre plusieurs intervenants, la gestion des réservations devient un levier majeur de performance. Elle contribue à prévenir les conflits de planification, à fluidifier la communication entre les services et à renforcer la traçabilité des opérations.

Dès lors, la mise en place d'un système de gestion performant apparaît comme une condition indispensable pour exploiter au mieux les ressources disponibles et accroître la productivité globale de l'organisme.

## 2.3 Étude de poste de travail

Afin de concevoir une solution adaptée aux besoins réels des utilisateurs, il est essentiel de réaliser une analyse descriptive des postes de travail. Cette étude met en relief les tâches spécifiques à chaque profil (Employé, Chauffeur, Chef de département, Chef de parc) et précise leur périmètre d'action ainsi que les fonctionnalités du système qu'ils sont amenés à exploiter.

| Poste | Activités principales |
|---|---|
| **Employé** | - Effectuer une demande de réservation (type de trajet : court ou long)<br>- Modifier ou annuler une demande existante<br>- Suivre le statut des demandes (en attente, validée, refusée)<br>- Gérer son profil |
| **Chef de Département** | - Consulter les demandes des employés<br>- Valider une demande (transfert au Chef de parc)<br>- Refuser une demande<br>- Consulter l'historique des décisions |
| **Chef de Parc** | - Gérer les véhicules et les chauffeurs (ajout, modification, affectation)<br>- Traiter les demandes validées<br>- Confirmer ou annuler selon disponibilité<br>- Saisir le kilométrage<br>- Consulter l'historique global |
| **Chauffeur** | - Consulter son planning de missions<br>- Signaler des incidents ou commentaires<br>- Consulter l'historique des missions |

TABLE 2.1 – postes de travail et les activites principales des acteurs

## 2.4 Description du projet

dans cette partie du projet, nous nous concentrons sur la problematique et les objectifs lies a la gestion des reservations des vehicules sous la division exploration .

### 2.4.1 Problématique

Malgré l'existence d'un processus organisationnel structuré, la gestion actuelle des réservations présente plusieurs limites.

En effet, l'utilisation de méthodes manuelles entraîne un manque de traçabilité, une difficulté d'accès à l'information en temps réel, ainsi qu'un risque élevé d'erreurs et de conflits dans les réservations.

De plus, l'absence d'un système centralisé complique la coordination entre les différents intervenants et allonge les délais de traitement des demandes.

Ainsi, la problématique principale de ce projet peut être formulée comme suit :

Comment concevoir et mettre en place une application mobile permettant d'optimiser la gestion des réservations de véhicules, tout en améliorant la traçabilité, la communication et l'efficacité du processus ?

### 2.4.2 Objectifs

Afin de répondre à la problématique identifiée, ce projet vise à atteindre les objectifs suivants :

- Développer une application mobile permettant de faciliter la gestion des réservations de véhicules
- Automatiser le processus de traitement des demandes
- Assurer une meilleure traçabilité des opérations
- Optimiser l'utilisation des véhicules et des chauffeurs
- Réduire les délais de traitement des demandes
- Améliorer la communication entre les différents acteurs

Ces objectifs permettront de proposer une solution efficace et adaptée aux besoins de l'organisme.

## 2.5 Conclusion

Ce chapitre a permis de présenter une étude détaillée du système existant de gestion des réservations de véhicules. Il a mis en évidence le fonctionnement actuel ainsi que les différents acteurs impliqués dans le processus.

L'analyse a également permis d'identifier plusieurs limites qui affectent l'efficacité du système, ce qui justifie la nécessité de proposer une solution informatisée.

Dans le chapitre suivant, nous nous concentrerons sur l'analyse, la conception et la modélisation de l'application à l'aide du langage UML (Unified Modeling Language).

# Chapitre 3

# Analyse et conception

## 3.1 Introduction

Avant de passer au code, il est indispensable de passer par une phase d'analyse et de conception. C'est l'étape où l'on prend le temps de bien comprendre les besoins du système et de définir qui va utiliser l'application et comment. L'idée ici est d'avoir une vision claire des objectifs du projet pour éviter les erreurs plus tard. Pour structurer tout cela, nous avons choisi d'utiliser le langage UML, qui est parfait pour représenter les différents aspects de l'application de manière visuelle et organisée.

## 3.2 Présentation d'UML

L'UML (Unified Modeling Language) est devenu un standard dans le monde du dé veloppement logiciel. C'est un langage graphique qui permet de modéliser des systèmes, surtout quand on travaille en programmation orientée objet. Ce qui est intéressant avec UML, c'est qu'il propose beaucoup de diagrammes différents, mais on n'est pas obligé de tous les utiliser. En général, pour bien décrire un projet, deux ou trois diagrammes suffisent. Dans notre cas, nous allons nous concentrer sur les plus essentiels : le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme de classe.1

## 3.3 Justification de l'utilisation d'UML

Le choix d'UML est motivé par plusieurs raisons stratégiques. D'abord, il permet de structurer la pensée avant le codage, évitant ainsi des erreurs de conception coûteuses. Ensuite, c'est un langage universel qui facilite la communication entre les développeurs et simplifie la maintenance future du logiciel. Enfin, disposer de schémas précis sert de guide technique, garantissant un développement plus rapide et une meilleure organisation du code.

## 3.4 Spécification des besoins

La spécification des besoins constitue une étape essentielle dans le développement de toute application. Elle permet de définir de manière claire et précise les attentes des utilisateurs, en distinguant les besoins fonctionnels et les besoins non fonctionnels.

### 3.4.1 Les besoins fonctionnels

Les besoins fonctionnels représentent les différentes fonctionnalités que le système doit assurer afin de répondre aux exigences des utilisateurs. Dans le cadre de notre application, les principaux besoins fonctionnels identifiés sont les suivants :

- Consulter l'état des demandes en cours.
- Visualiser l'historique des véhicules utilisés lors des missions précédentes.
- Gérer les utilisateurs ainsi que leurs droits d'accès au système.
- Configurer et assurer la maintenance du système ;
- Administrer la base de données et effectuer des sauvegardes régulières.

### 3.4.2 Les besoins non fonctionnels

Les besoins non fonctionnels correspondent aux contraintes et aux exigences de qualité auxquelles le système doit se conformer. Ils ne décrivent pas directement les fonctionna lités, mais définissent les critères de performance, de sécurité et d'ergonomie. Dans notre cas, ces besoins peuvent être résumés comme suit :

- **Sécurité :** L'accès à l'application est conditionné par une authentification préa lable des utilisateurs.
- **Ergonomie :** L'application doit offrir une interface simple, claire et conviviale, facilitant son utilisation.
- **Intégrité des données :** Des mécanismes de validation doivent être mis en place afin de garantir la cohérence et la fiabilité des données.
- **Cohérence :**Le système doit adopter une uniformité au niveau de la terminologie, de la notation et de la structure pour améliorer la lisibilité.
- **Apprenabilité :** L'application doit être facile à prendre en main par les utilisa teurs, même novices.
- **Flexibilité :** Le système doit pouvoir s'adapter aux différents besoins et profils d'utilisateurs.
- **Robustesse :** Le système doit être capable de fonctionner correctement même en présence d'erreurs ou de situations imprévues.

## 3.5 Diagramme de Cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML qui décrit les fonctionnalités d'un système du point de vue de l'utilisateur. Il met en évidence les relations entre les **acteurs** et les **cas d'utilisation**, afin de mieux comprendre les besoins fonctionnels du système2.

### 3.5.1 Diagramme de cas d'utilisation générale du système

la figure suivante représente le diagramme de cas d'utilisation générale de système :

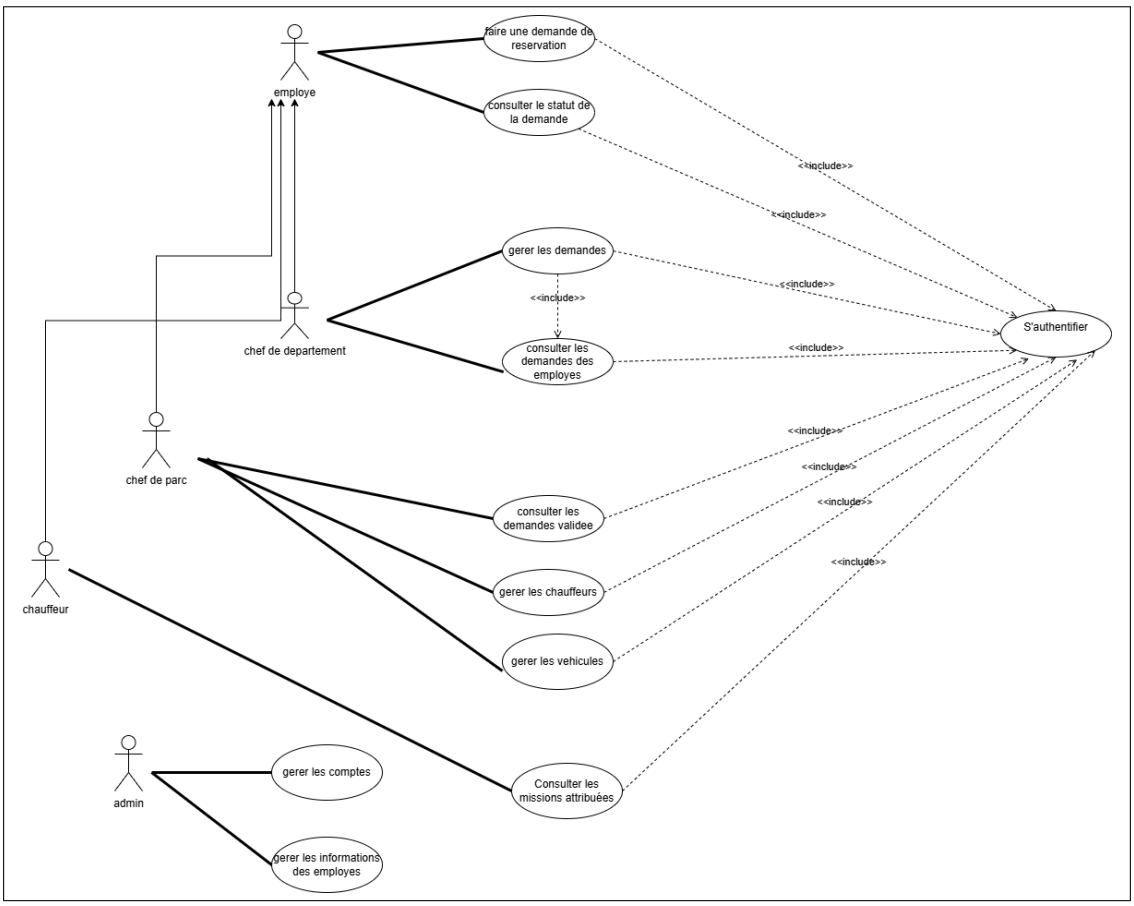



FIGURE 3.1 – Diagramme de cas d'utilisation générale

### 3.5.2 Diagramme de cas d'utilisation : chef de département

la figure suivante représente le diagramme de cas d'utilisation chef de département :

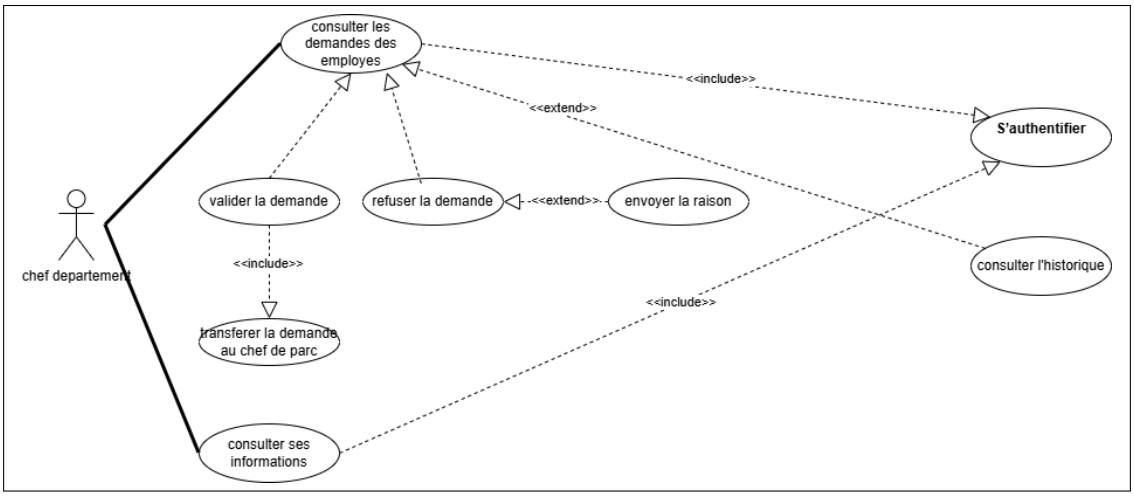



FIGURE 3.2 – Diagramme de cas d'utilisation chef de département

### 3.5.3 Diagramme de cas d'utilisation : chef de parc

la figure suivante représente le diagramme de cas d'utilisation chef de parc :

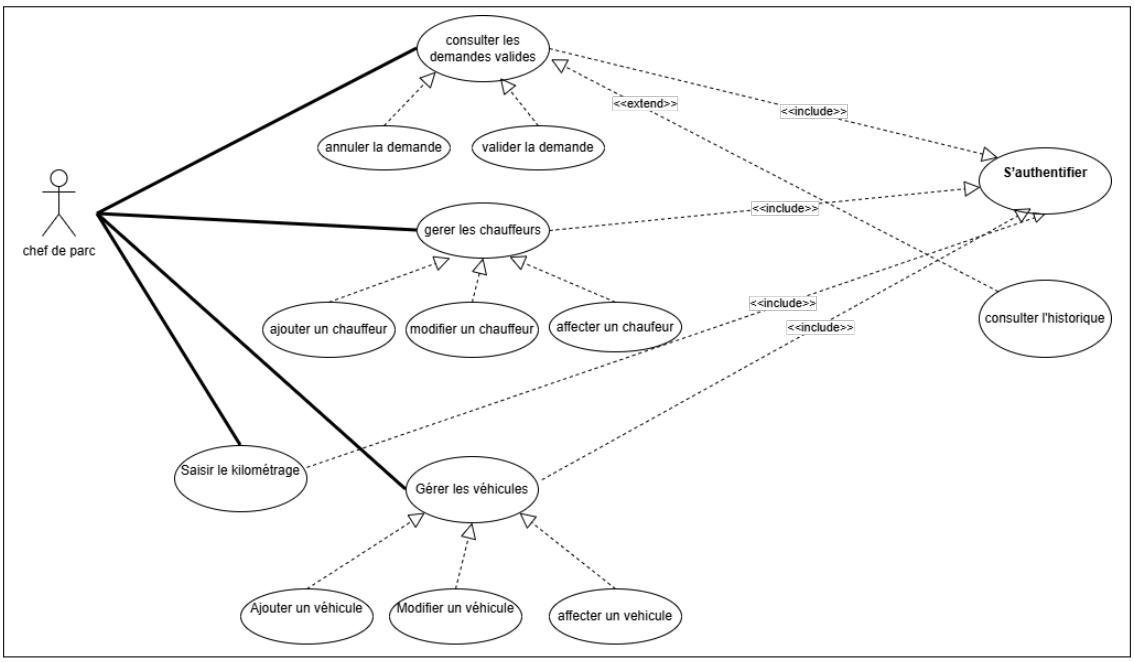



FIGURE 3.3 – Diagramme de cas d'utilisation chef de parc

### 3.5.4 Diagramme de cas d'utilisation : employé

la figure suivante représente le diagramme de cas d'utilisation employé :

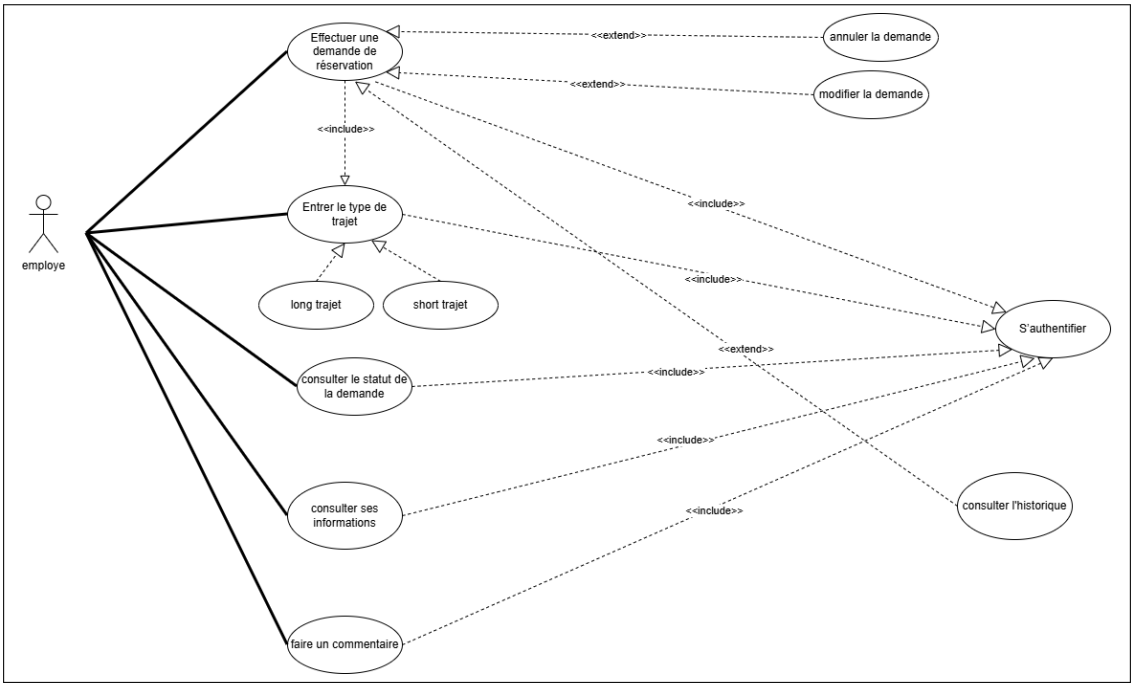



FIGURE 3.4 – Diagramme de cas d'utilisation employé

### 3.5.5 Diagramme de cas d'utilisation :chauffeur

la figure suivante représente le diagramme de cas d'utilisation chauffeur :

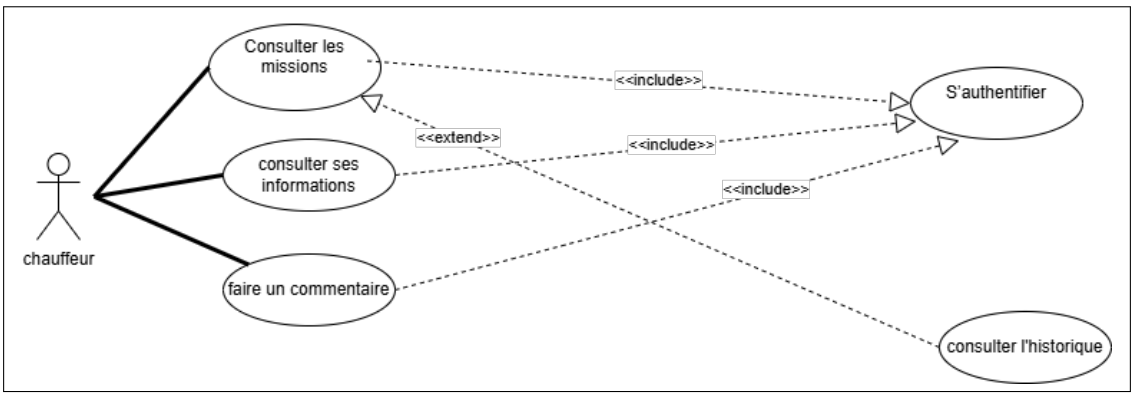



FIGURE 3.5 – Diagramme de cas d'utilisation chauffeur

## 3.6 Elaboration de diagramme de classe :

Le diagramme de classes est un diagramme structurel du langage de modélisation UML (**Unified Modeling Language**). Il constitue l'un des diagrammes les plus utilisés dans la conception orientée objet, car il offre une vue statique et globale du système à développer.

Il permet de représenter l'ensemble des **classes** du système, chacune étant définie par ses **attributs** (les données qu'elle contient) et ses **méthodes** (les opérations qu'elle peut effectuer). Il décrit également les différentes **relations** qui existent entre ces classes, notamment :

- **L'association :** une relation simple entre deux classes.
- **L'héritage :** une classe fille qui hérite des attributs et méthodes d'une classe mère.
- **L'agrégation :** une relation de type "contient" faible.
- **La composition :** une relation de type "contient" forte, où la classe fille ne peut exister sans la classe mère.

### 3.6.1 Diagramme de classe de système :

cette figure représente le diagramme de classe :

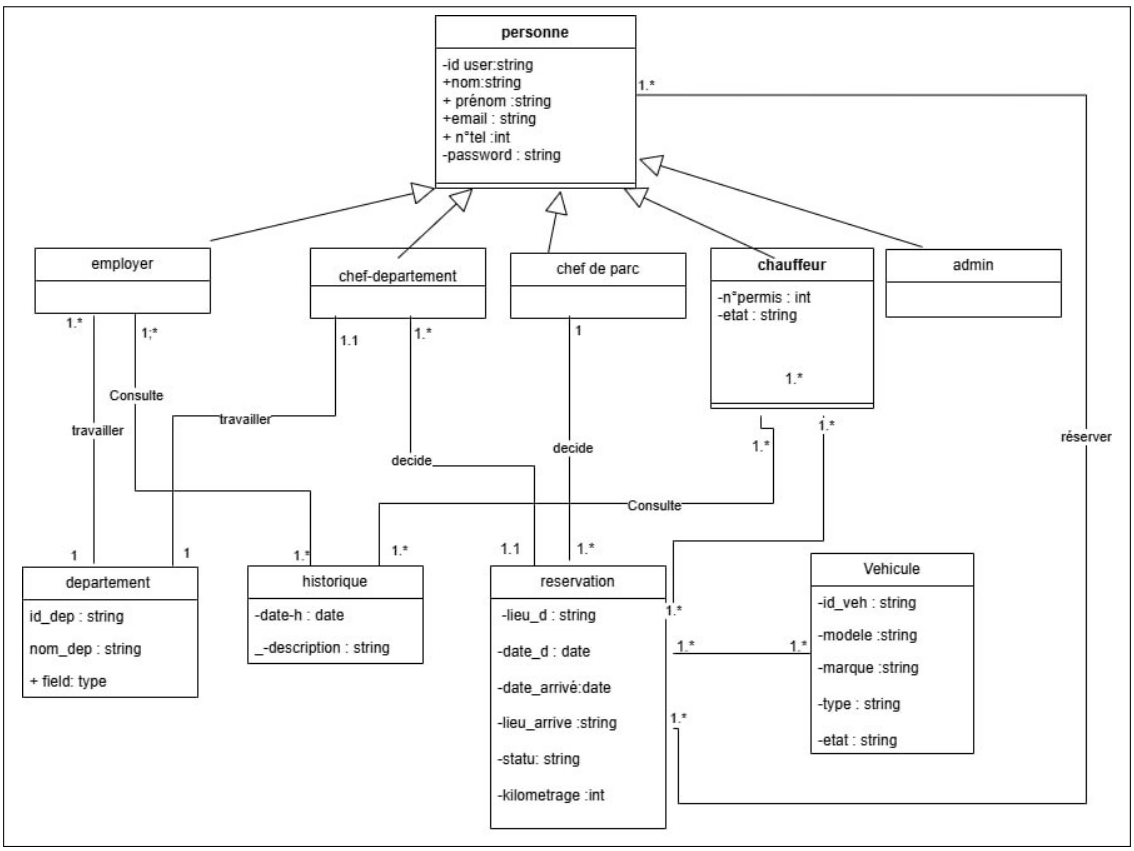



FIGURE 3.6 – Diagramme de classe de système

### 3.6.2 Règles de gestion :

voici les Règles de gestion de notre système :

- Un utilisateur peut avoir le rôle d'employé, chef de département, chef de parc ou chauffeur.
- Le chef de parc peut consulter les demandes validées par le chef de département afin de les accepter ou les refuser.
- Chaque chauffeur possède un état dans le système (disponible, en congé, occupé, etc.).
- Chaque véhicule possède un état dans le système (disponible, en maintenance, réservé, etc.).
- Le chef de parc gère les chauffeurs et les véhicules à travers les opérations d'ajout, modification et affectation.
- Le chef de département peut valider une demande de réservation puis la transmettre au chef de parc, ou bien la refuser.
- Une réservation doit être effectuée par un seul chauffeur.
- Un chauffeur peut effectuer une ou plusieurs réservations.
- Une réservation doit être associée à un seul véhicule.
- Un véhicule peut être utilisé dans une ou plusieurs réservations.
- Chaque action importante effectuée dans le système doit être enregistrée dans l'historique.
- Toutes les opérations importantes nécessitent une authentification.
- L'administrateur gère les comptes utilisateurs du système.
- Un chef de département, un chef de parc ou un chauffeur peut également être un employé.

## 3.7 Diagramme de séquences :

Le diagramme de séquence est un diagramme comportemental du langage de modélisation **UML (Unified Modeling Language)**. Il représente de manière chronologique et ordonnée les interactions entre les **acteurs** (utilisateurs ou systèmes externes) et les **objets** du système, en illustrant précisément qui envoie quoi à qui et dans quel ordre 3.

Chaque participant est représenté par une **ligne de vie** verticale, et les échanges entre eux sont modélisés sous forme de **messages**. On distingue plusieurs types de messages :

- **Message synchrone :** l'expéditeur envoie un message et *attend* la réponse avant de poursuivre son exécution.
- **Message asynchrone :** l'expéditeur envoie un message *sans attendre* la réponse et continue son exécution.
- **Message de retour :** représente la réponse renvoyée par le récepteur après traitement.
- **Message réflexif :** un objet s'envoie un message à lui-même.

De plus, le diagramme de séquence utilise des **fragments combinés** pour modéliser les interactions complexes :

- **ALT :** représente une alternative entre plusieurs scénarios possibles, équivalent à un *if/else*.
- **OPT :** représente un bloc optionnel exécuté uniquement si une condition est vérifiée, équivalent à un *if* sans else.
- **LOOP :** représente une boucle répétée tant qu'une condition reste vraie.
- **REF :** permet de faire référence à un autre diagramme de séquence afin d'éviter toute répétition.

### 3.7.1 Diagramme de séquence : "Authentification"

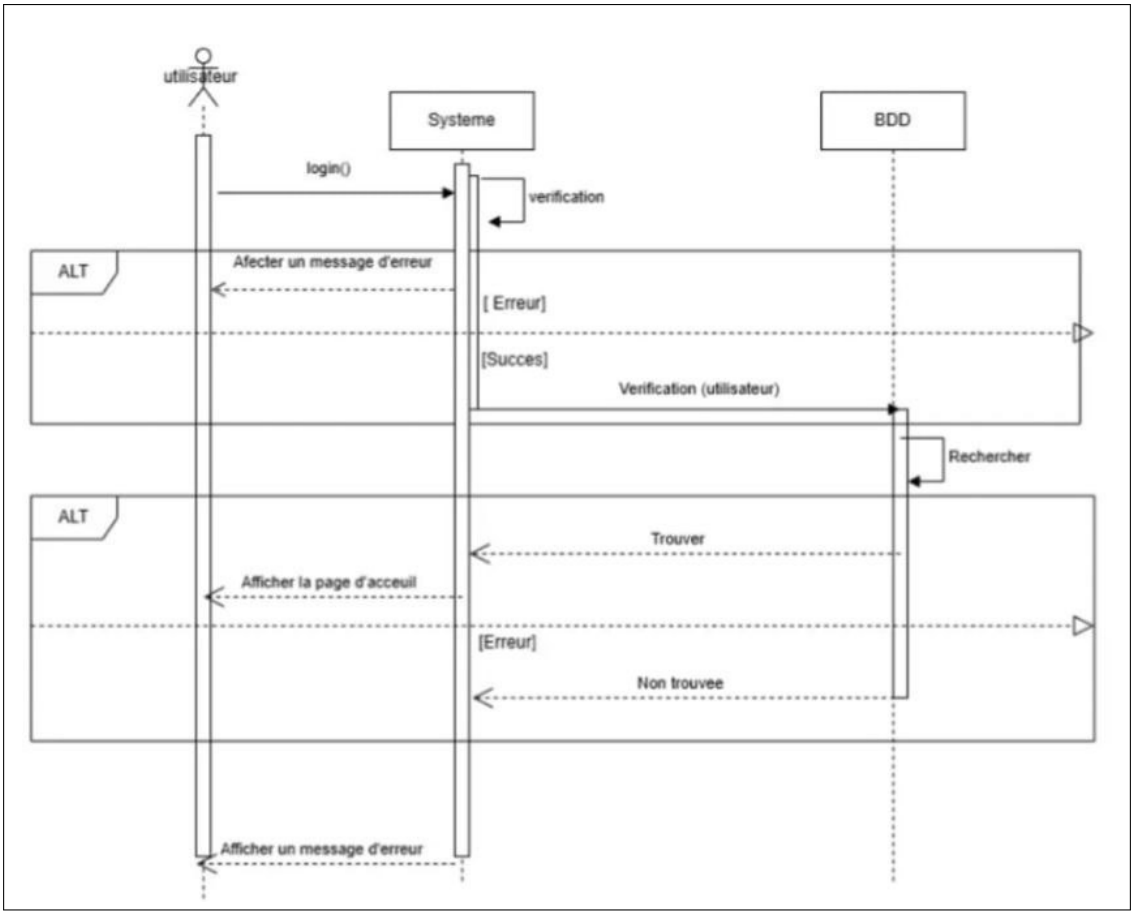



FIGURE 3.7 – Diagramme de séquence d'authentification

### 3.7.2 Diagramme de séquence : "consulter mission"

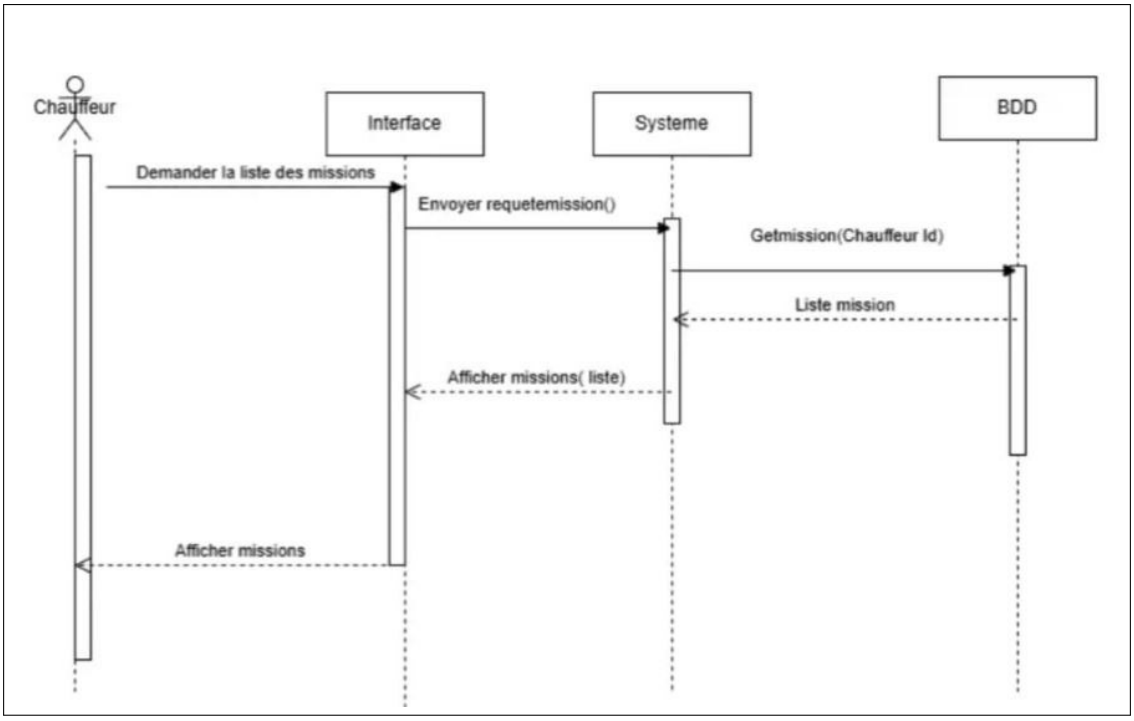



FIGURE 3.8 – Diagramme de séquence associe a la consultation des missions

### 3.7.3 Diagramme de séquence : "Demande de réservation"

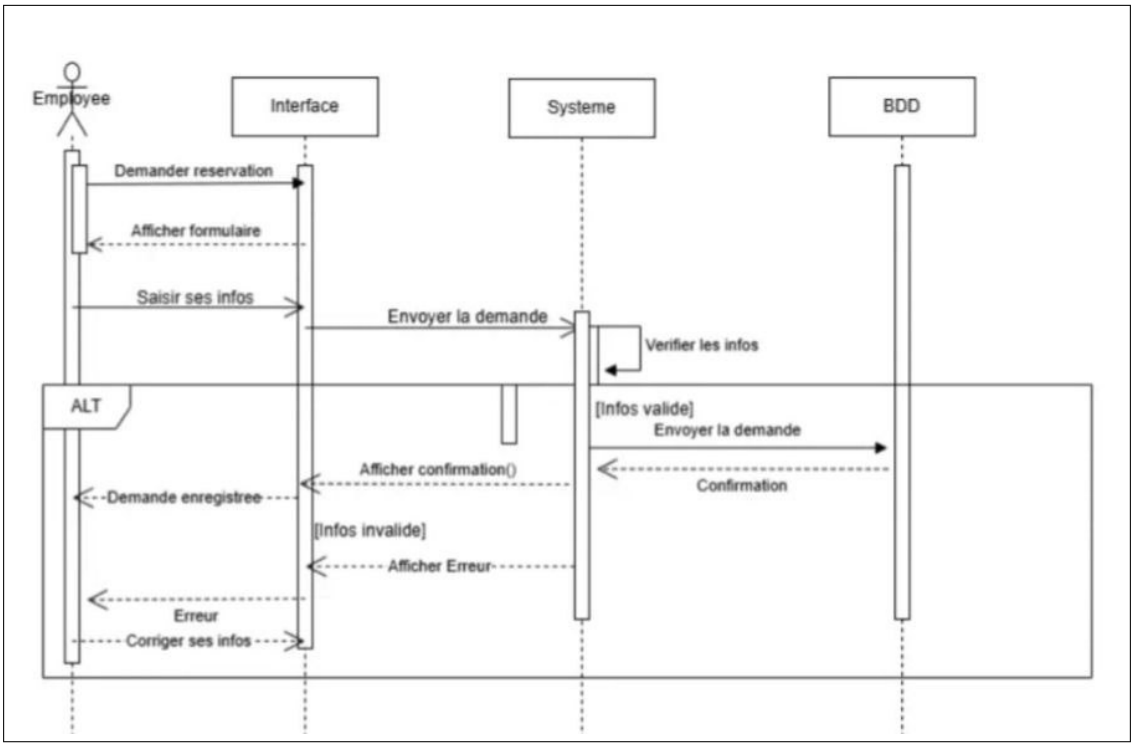



FIGURE 3.9 – Diagramme de séquence associe a la demande de réservation

### 3.7.4 Diagramme de séquence : "Traiter la demande"

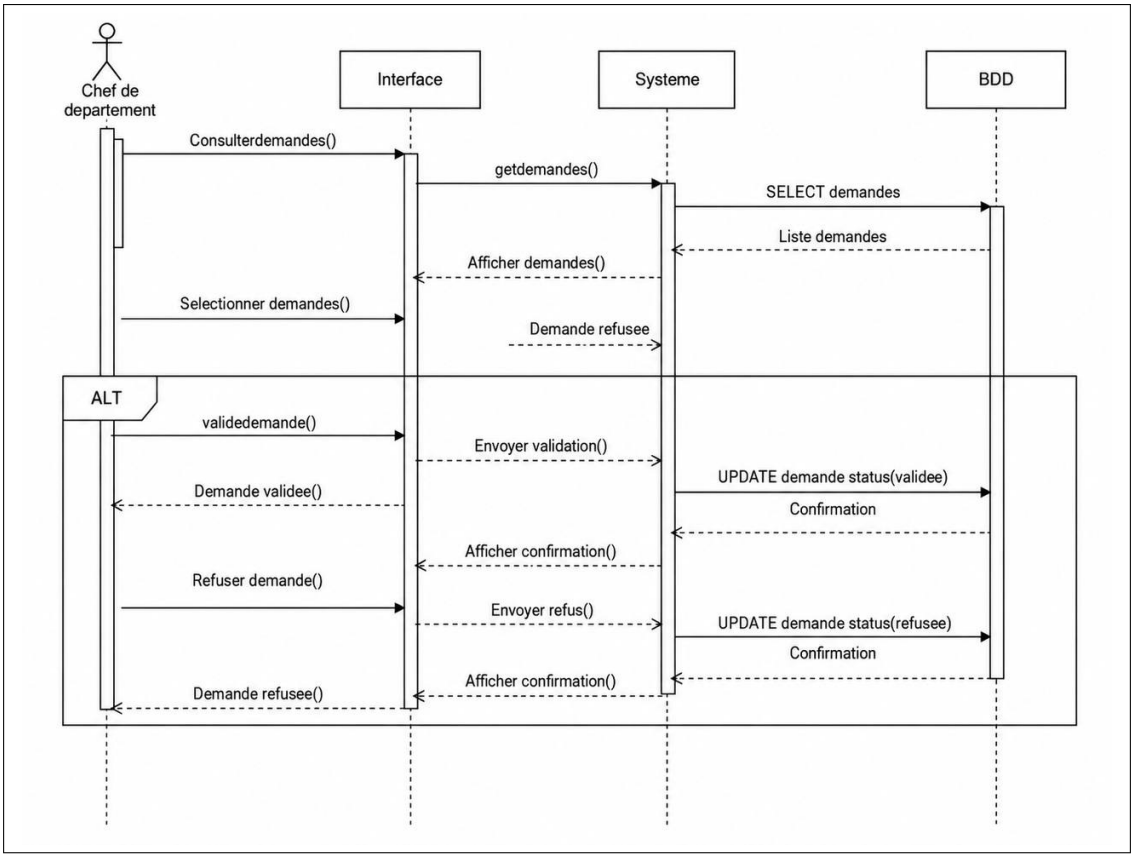



FIGURE 3.10 – Diagramme de séquence associe au traitement des demandes par le chef de département

### 3.7.5 Diagramme de séquence : "l'affectation des chauffeurs et des véhicules"

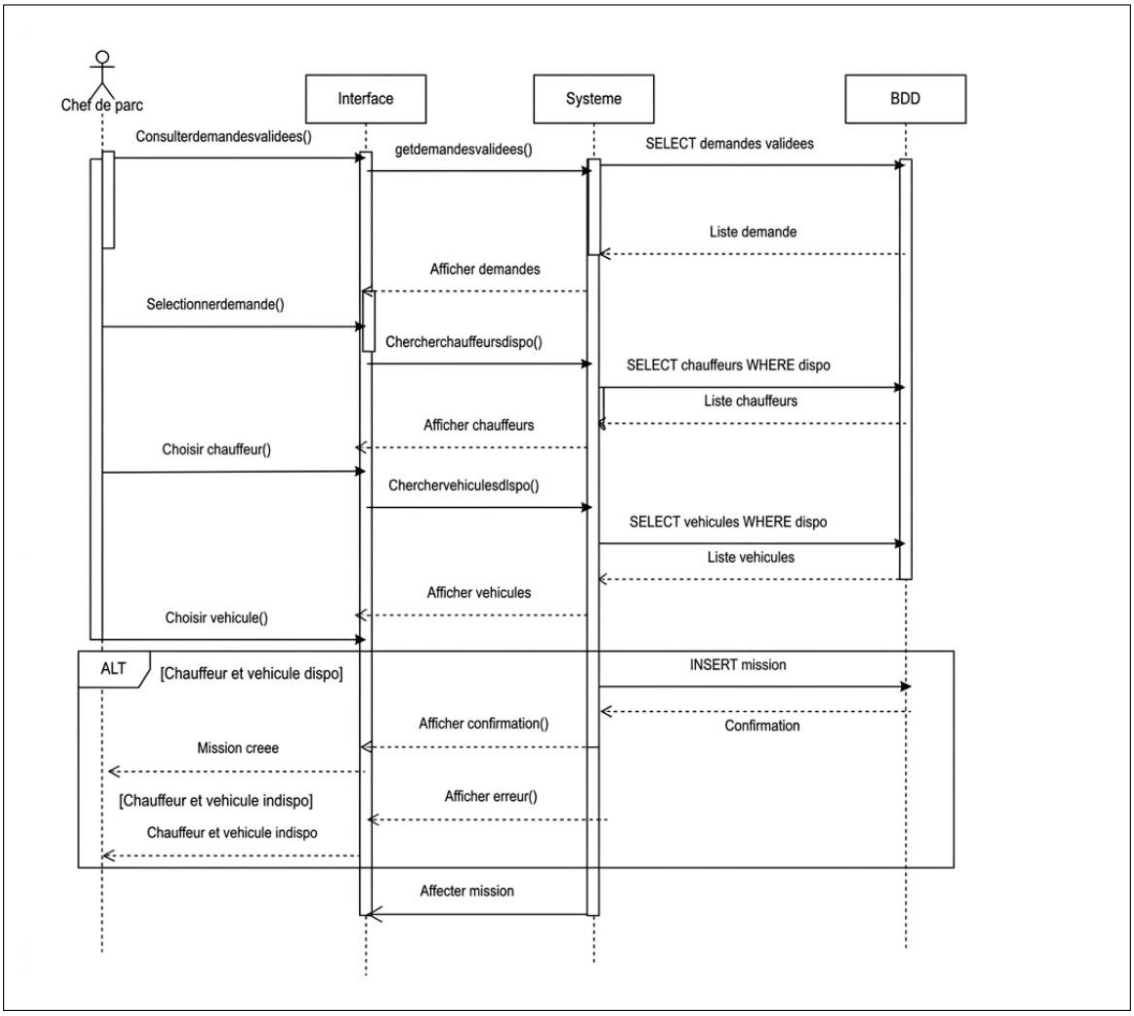



FIGURE 3.11 – Diagramme de séquence associe a l'affectation des chauffeurs et des véhicules

## 3.8 Conclusion

Dans ce chapitre, nous avons examiné l'approche de modélisation des traitements en utilisant les diagrammes **UML**, tels que le **diagramme de cas d'utilisation**, le **diagramme de classes** et le **diagramme de séquence**. Ces outils nous ont fourni une représentation détaillée et précise du fonctionnement de notre application, offrant ainsi une vision complète de son architecture et de ses interactions. Le chapitre qui suit sera consacré à la **mise en œuvre** et à la **réalisation** de notre application.

# Chapitre 4

# Implementation

## 4.1 Introduction

Après avoir présenté l'étude conceptuelle dans le chapitre précédent, il est maintenant nécessaire de passer à l'étape de réalisation de notre application.

Dans ce chapitre, nous allons présenter la partie pratique du projet. Nous commencerons par décrire l'environnement de travail, aussi bien logiciel que matériel, utilisé durant le développement de l'application. Ensuite, nous présenterons quelques interfaces graphiques réalisées afin de montrer le résultat final de notre travail.

## 4.2 L'architecture MVVM

L'architecture MVVM (Model-View-ViewModel) est un modèle de conception utilisé pour organiser le code d'une application de manière claire et structurée. Elle repose sur trois composants principaux : le Model, chargé de la gestion des données ; la View, responsable de l'affichage de l'interface utilisateur ; et le ViewModel, qui sert d'intermédiaire entre les deux. Cette architecture permet de séparer les différentes responsabilités, d'améliorer la maintenance du code et de faciliter l'évolution de l'application. Dans notre projet, elle a été adoptée pour assurer une meilleure organisation du développement.

## 4.3 Environnement de développement :

Dans cette partie , nous présentons l'environnement de travail (logiciel et matériel )

## 4.4 Environnement matériel

Au cours de notre projet nous avons utilise matériels suivants :

| PC / Caractéristiques | DELL Latitude 5490 | DELL latitude E5470 |
|---|---|---|
| **Processeur** | Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz | Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz |
| **Mémoire vive** | 8 Go | 4 Go |
| **Disque dur** | 238 SSD | 119 SSD |
| **Type de système** | Systeme d'exploitation 64 bits, processur x64 | Systeme d'exploitation 64 bits, processur x64 |
| **Carte graphique** | Intel(R) UHD Graphics 620 (128 Mo) | Intel(R) HD Graphics 520 (128 Mb) |

TABLE 4.1 – Environnement matériel utilisé durant le développement

## 4.5 Environnement logiciel :

Lors de la conception de notre application, nous avons utilisé plusieurs environnements, langages, outils et technologies, chacun jouant un rôle spécifique dans la phase de réalisation. Nous allons donc les présenter en expliquant leurs fonctions respectives. langages de programmation : les langages de programmation utilises sont :

- **dart** :développé par Google. Il s'agit d'un langage moderne, orienté objet et principalement utilisé avec le framework Flutter pour le développement d'applications mobiles, web et desktop. Dart se distingue par sa simplicité, ses bonnes performances et sa capacité à créer des interfaces dynamiques et fluides. Il a été choisi pour sa compatibilité avec Flutter et sa rapidité de développement4.
- **Java** :est un langage orienté objet largement utilisé pour le développement d'applications web, mobiles et desktop. Il se distingue par sa robustesse, sa sécurité et sa portabilité grâce au principe Write Once, Run Anywhere. Dans notre projet, Java a été utilisé principalement pour le développement de la partie back-end, notamment avec le framework Spring Boot, afin d'assurer la gestion des données et des services de l'application5.

Framework et technologies :

- **Flutter** :est un framework open source développé par Google, utilisé pour la création d'applications mobiles, web et desktop à partir d'un seul code source. Il repose sur le langage Dart et permet de concevoir des interfaces graphiques modernes, fluides et performantes. Flutter se distingue par sa rapidité de développement grâce à la fonctionnalité Hot Reload, qui permet de visualiser instantanément les modifications apportées au code6.
- **Spring boot** :est un framework Java open source basé sur la plateforme Spring, conçu pour faciliter le développement d'applications web et de services back-end. Il permet de créer rapidement des applications robustes, sécurisées et performantes grâce à sa configuration automatique et à son architecture simplifiée. Spring Boot intègre plusieurs outils utiles, tels que la gestion des dépendances, la connexion aux bases de données et la création d'API REST. Dans notre projet, il a été utilisé

pour développer la partie serveur de l'application et assurer la communication avec la base de données7.

- **Figma** :est un outil de conception d'interfaces graphiques utilisé pour créer des maquettes d'applications et de sites web. Il permet de réaliser des designs modernes, de prototyper les interfaces et de collaborer facilement en équipe. Dans notre projet, Figma a été utilisé pour concevoir les interfaces de l'application avant le développement8.
- **Stitch** :est un outil de conception assisté par intelligence artificielle permettant de créer rapidement des interfaces graphiques modernes et interactives. Il facilite la génération de maquettes et d'éléments visuels adaptés aux besoins des applications. Dans notre projet, Stitch a été utilisé pour aider à la conception de certaines interfaces9 .
- **Postman** :est un outil utilisé pour tester et gérer les API. Il permet d'envoyer des requêtes HTTP, de vérifier les réponses du serveur et de valider le bon fonctionnement des services web. Dans notre projet, Postman a été utilisé pour tester les API développées avec Spring Boot et assurer la communication entre le front-end et le back-end10.
- **GitHub** :est une plateforme de gestion de versions basée sur Git, utilisée pour stocker, organiser et partager le code source d'un projet. Elle facilite la collaboration entre les membres d'une équipe, le suivi des modifications et la sauvegarde du travail. Dans notre projet, GitHub a été utilisé pour gérer le code source et coordonner le développement de l'application11.

Outils de développement

- **Visuel studio code** :est un éditeur de code développé par Microsoft, utilisé pour programmer avec plusieurs langages et technologies, notamment Flutter et Spring Boot. Il se distingue par sa simplicité, sa rapidité et ses nombreuses extensions. Dans notre projet, il a été utilisé pour le développement et la gestion du code source de l'application12.
- **Android Studio** :est un environnement de développement intégré (IDE) spécialement conçu pour la création d'applications Android. Il offre plusieurs outils utiles, tels que l'émulateur Android, la gestion du SDK et le débogage des applications. Dans notre projet, Android Studio a été utilisé pour exécuter, tester et configurer l'application mobile développée avec Flutter13.

outils de gestion de bases de données :

- **phpMyAdmin** :est un outil de gestion de bases de données MySQL et MariaDB accessible via une interface web. Il permet de créer, modifier et administrer facilement les bases de données ainsi que leurs tables. Dans notre projet, phpMyAdmin a été utilisé pour gérer la base de données de l'application15.
- **XAMPP** :est un environnement de développement local qui regroupe plusieurs outils, notamment Apache, MySQL, PHP et phpMyAdmin. Il permet de créer et de tester des applications web sur un serveur local. Dans notre projet, XAMPP a été utilisé pour héberger localement la base de données et faciliter son administration16.

outils de modélisation UML :

- **Draw.io** :est un outil de conception de diagrammes en ligne utilisé pour créer différents schémas, tels que les diagrammes UML, organigrammes et modèles de bases de données. Il se distingue par sa simplicité et sa facilité d'utilisation. Dans notre projet, Draw.io a été utilisé pour réaliser les diagrammes de conception du système17.

## 4.6 présentation des interfaces graphiques :

Dans cette partie nous présentons notre travail en montrant des captures d'écran des interfaces développées de l'application.

### 4.6.1 Les interfaces d'authentification :

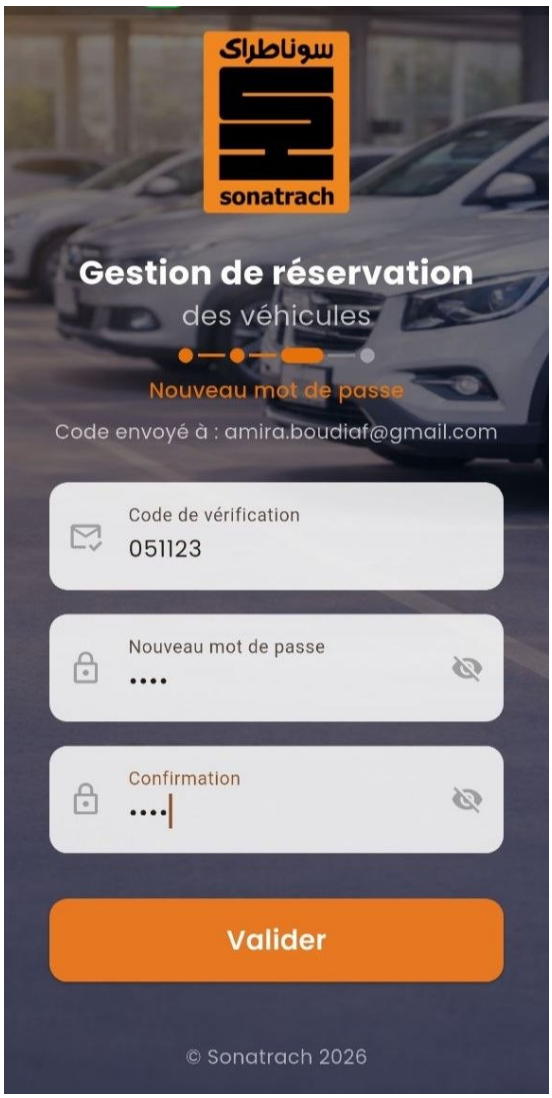
Les figures présentent le processus d'authentification en trois étapes. L'utilisateur commence par une première connexion où il renseigne ses informations personnelles ainsi qu'un code unique. Étant donné que tous les comptes sont initialement créés par l'administrateur avec un mot de passe par défaut identique, chaque utilisateur est obligé de changer son mot de passe avant de pouvoir accéder à l'application. Une fois le nouveau mot de passe validé, l'utilisateur est redirigé vers l'écran de connexion classique pour s'authentifier normalement.



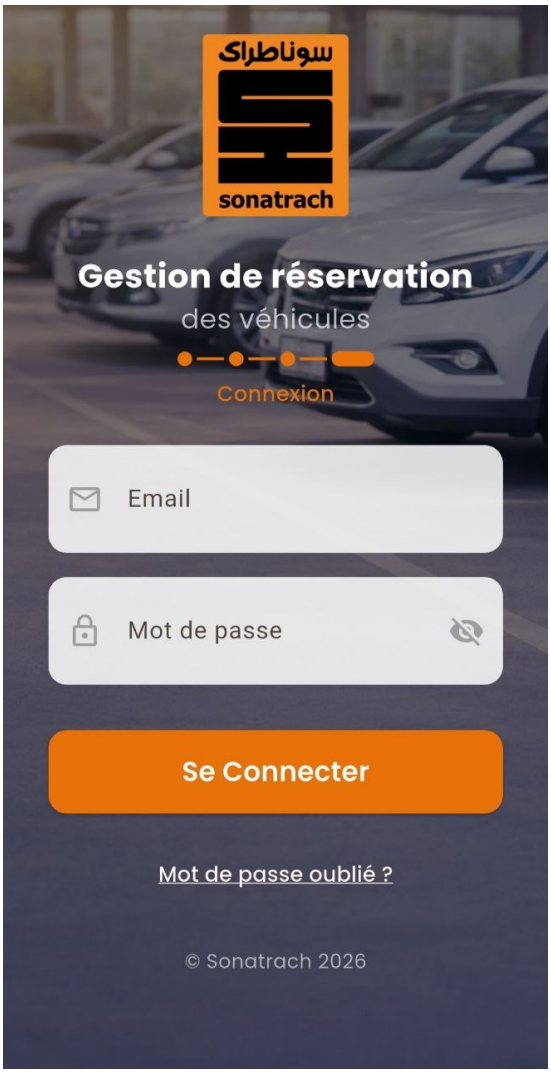

(a) Interface première connexion




(b) Récupération du mot de passe




(a) Réinitialiser le mot de passe




(b) Interface de connexion classique

FIGURE 4.2 – l'interface d'authentification

### 4.6.2 Les interfaces chef de parc :

Les figures présentent l'ensemble des fonctionnalités accessibles au Chef de Parc après authentification.Les figures présentent l'ensemble des fonctionnalités accessibles au Chef de Parc après authentification. Une fois connecté, l'utilisateur ayant le double rôle choisit d'accéder à l'espace Chef de Parc ou à l'espace Employé. L'interface se compose de cinq onglets : Demandes, où le chef consulte les demandes de réservation préalablement validées par le Chef de Département, et peut attribuer un chauffeur et un véhicule avant d'accepter ou de refuser la mission ; Acceptées, affichant les missions validées avec le chauffeur et le véhicule assignés ; Véhicules, permettant de visualiser le statut de chaque véhicule du parc (disponible, en mission ou en maintenance), d'ajouter un nouveau véhicule via un formulaire (marque, modèle, type et état), ainsi que de modifier ou supprimer un véhicule existant ; Chauffeurs, offrant une vue sur la disponibilité et le numéro de permis de chaque chauffeur, avec la possibilité d'ajouter un nouveau chauffeur (prénom, nom, email, téléphone, numéro de permis, mot de passe et état), de le modifier ou de le supprimer ; et enfin Historique, qui retrace toutes les actions effectuées par le Chef de Parc.




(a) Interface de sélection du rôle




(b) Interface de Consultation des demandes validées




(a) Interface de saisie du kilométrage



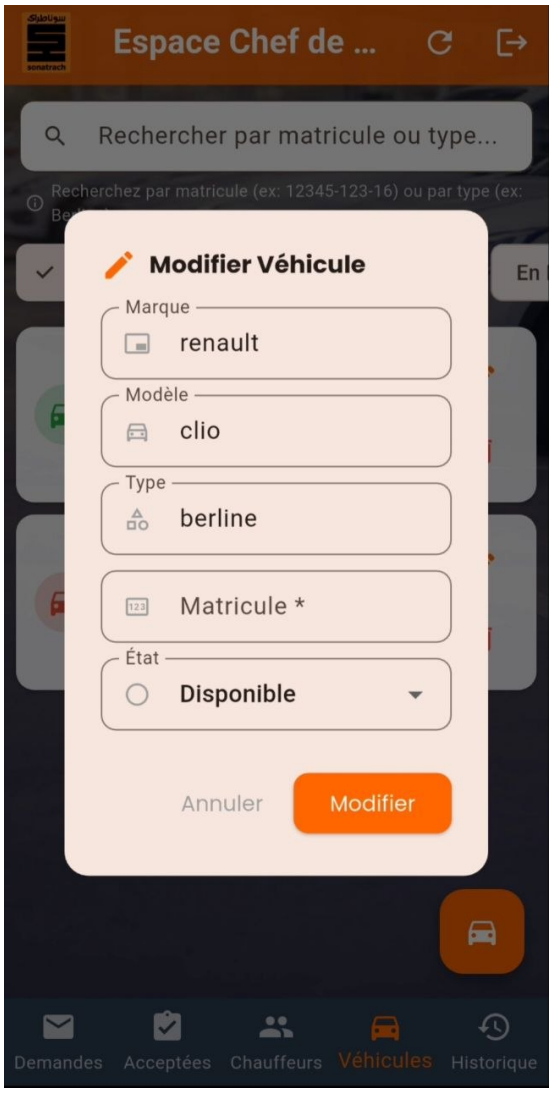

(b) Interface de gestion des chauffeurs



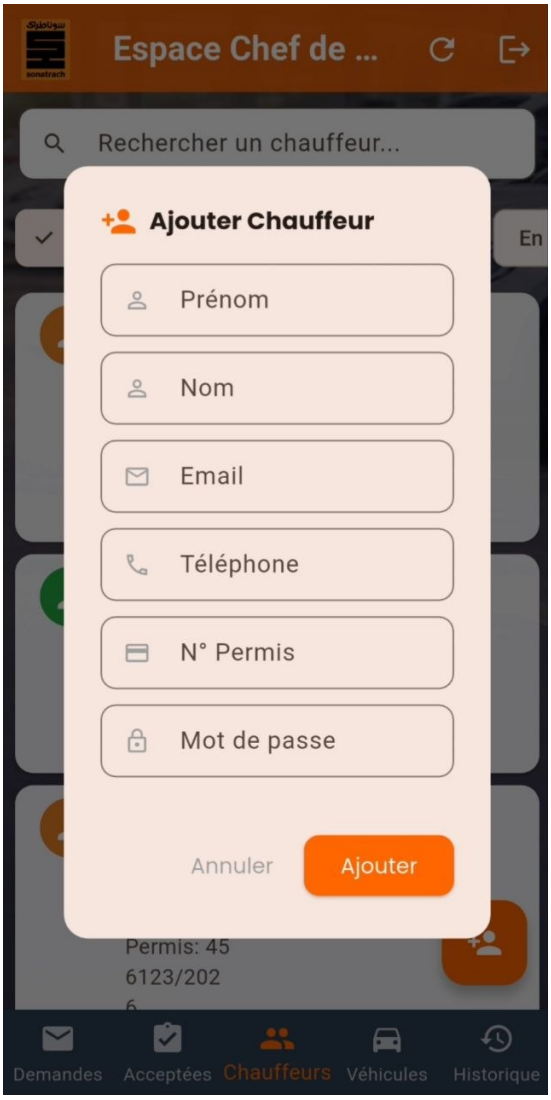

(a) Interface historique




(b) Interface de gestion de véhicules




(a) Interface ajouter véhicule




(b) Interface ajouter chauffeur

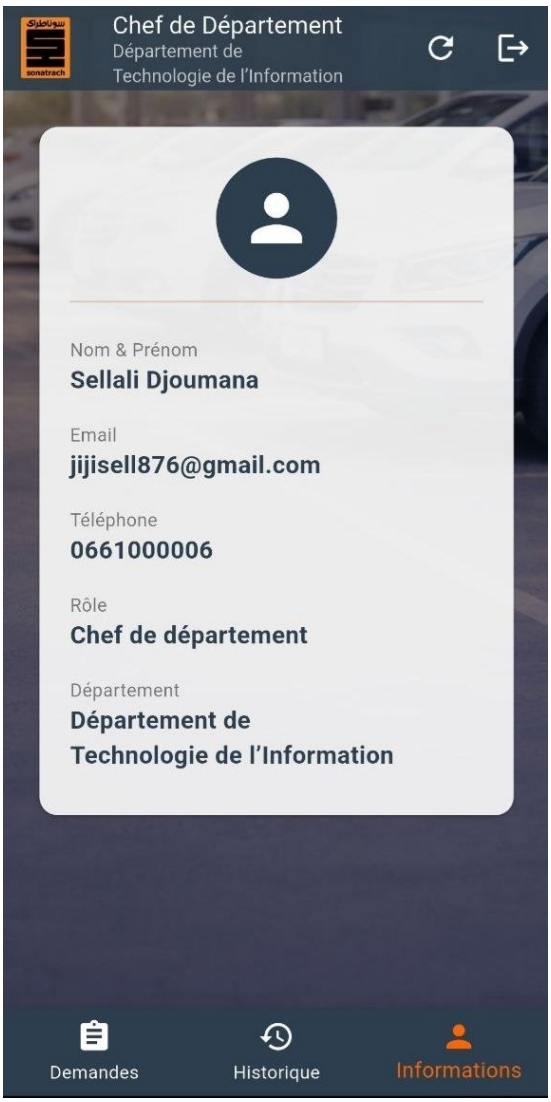
FIGURE 4.6 – les interfaces du chef de parc

### 4.6.3 Les interfaces de chef département :

Les figures présentent l'espace dédié au Chef de Département. Après authentification, l'utilisateur ayant un double rôle choisit entre l'espace Chef de Département (supervision) ou l'espace Employé (réservation). L'interface du Chef de Département se compose de trois onglets : Demandes, pour consulter et valider les demandes de réservation soumises par les employés de son département ; Historique, retraçant toutes ses actions ; et Informations, affichant ses données personnelles telles que son nom, email, téléphone , rôle et département.

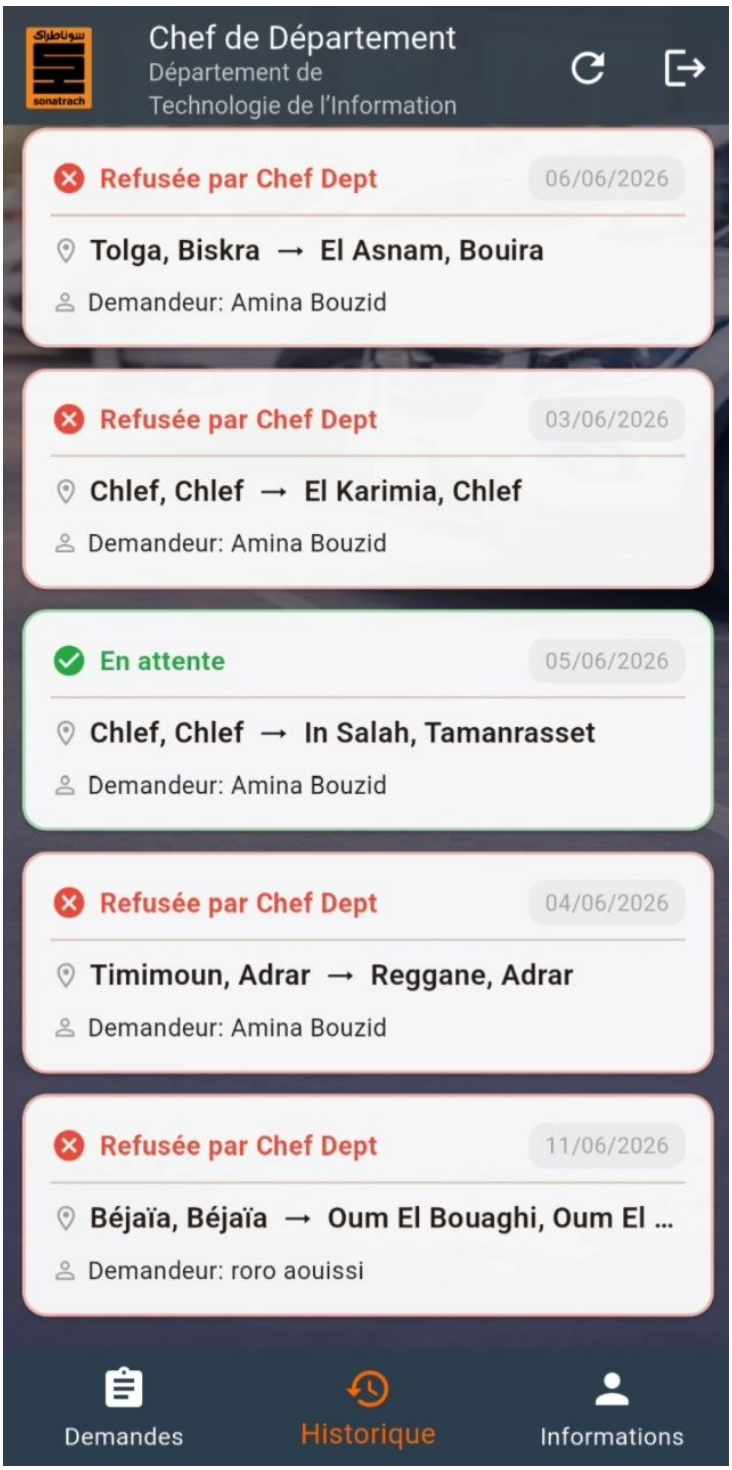



(a) Interface de sélection du rôle




(b) Interface profil du chef département

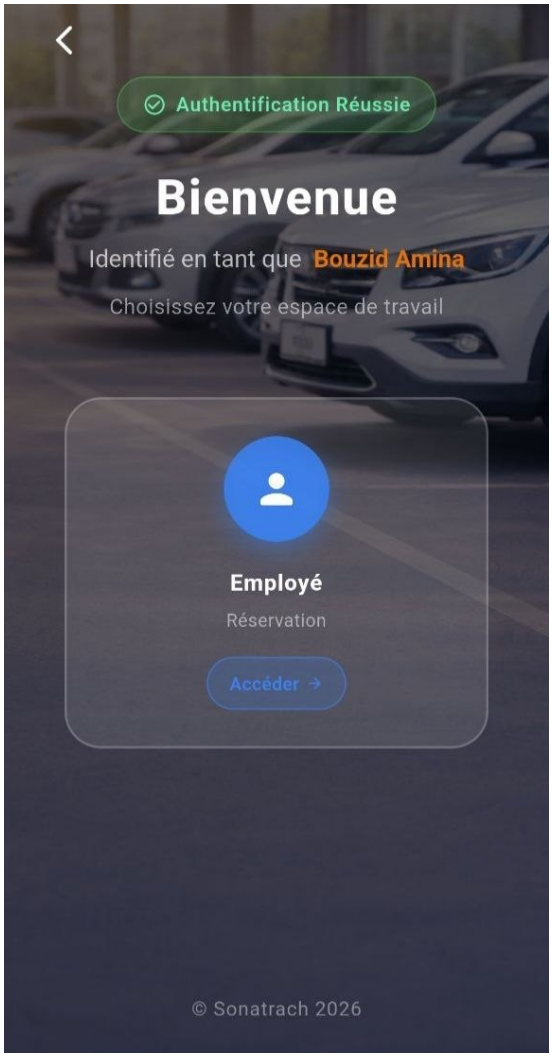
FIGURE 4.7 – Interfaces du chef département




(a) Interface historique du chef département

### 4.6.4 Les interfaces de l'employé :

Les figures présentent l'espace dédié à l'Employé. Après authentification, l'utilisateur accède uniquement à l'espace Employé (réservation). L'interface propose quatre onglets : Réservations, où l'employé soumet une demande en choisissant entre un formulaire Court Trajet (wilaya de départ, destination, dates et heures) ou un formulaire Long Trajet (avec en plus le choix du type de véhicule) ; Statut, pour suivre l'état de ses demandes ; Historique, retraçant ses réservations passées ; et Profil, affichant ses informations personnelles.

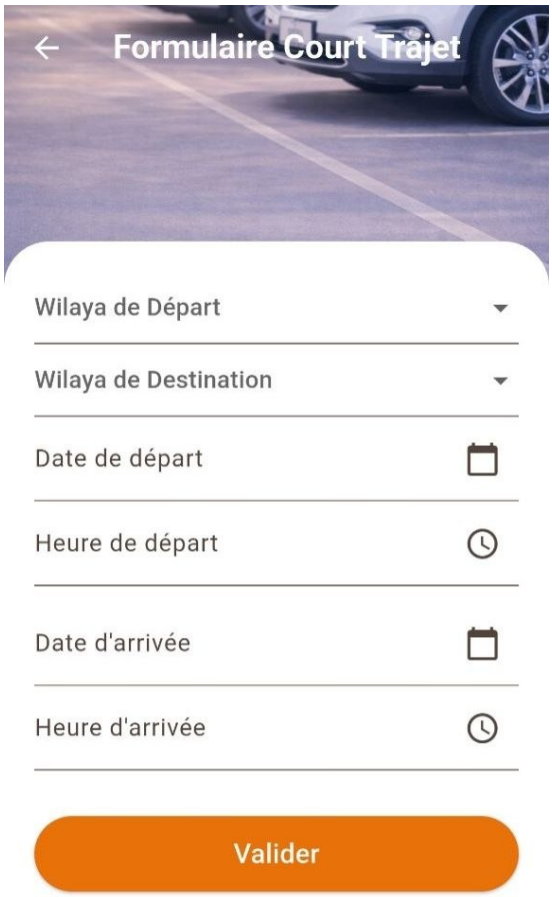



(a) Interface d'accueil — Espace Employé

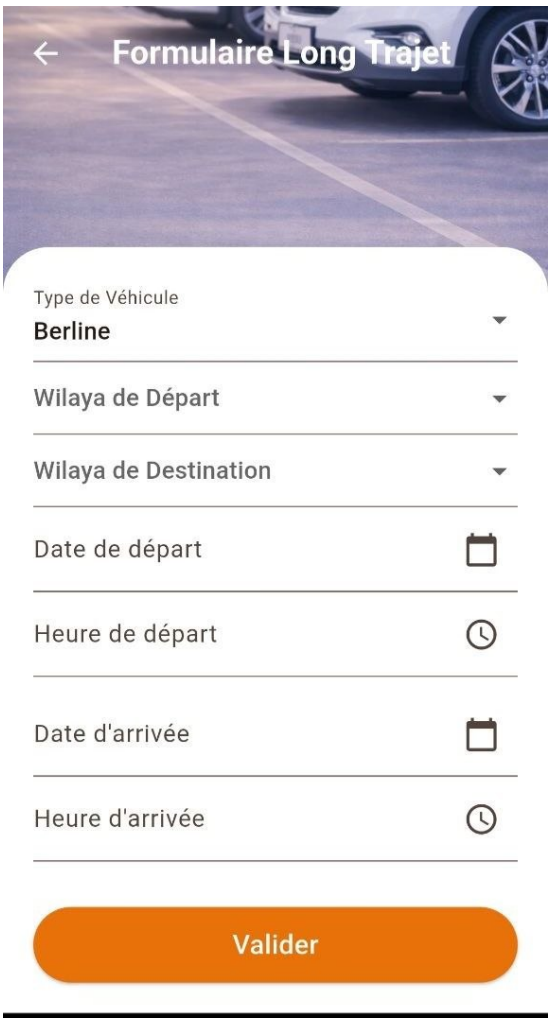



(b) Interface de sélection du type de réservation

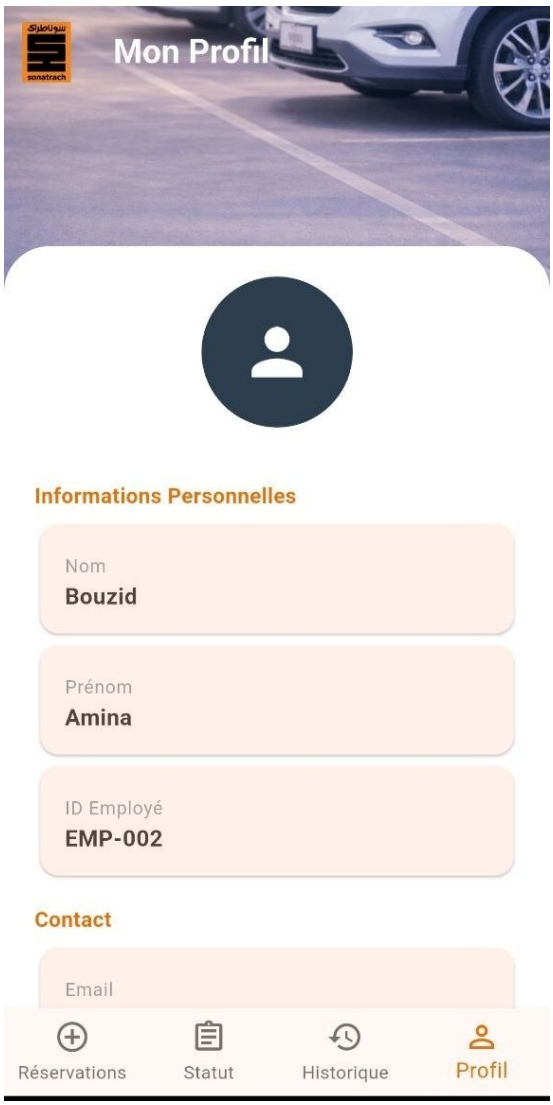
FIGURE 4.9 – les Interfaces de l'Espace Employé




(a) interface Formulaire de réservation — court Trajet




(b) interface Formulaire de réservation — Long Trajet

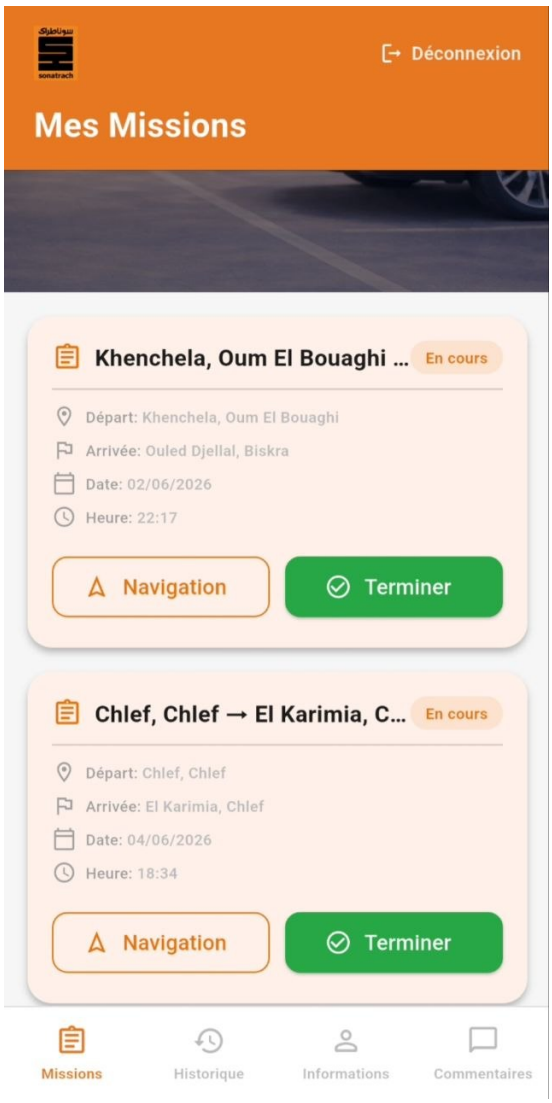



(a) interface du profil employé

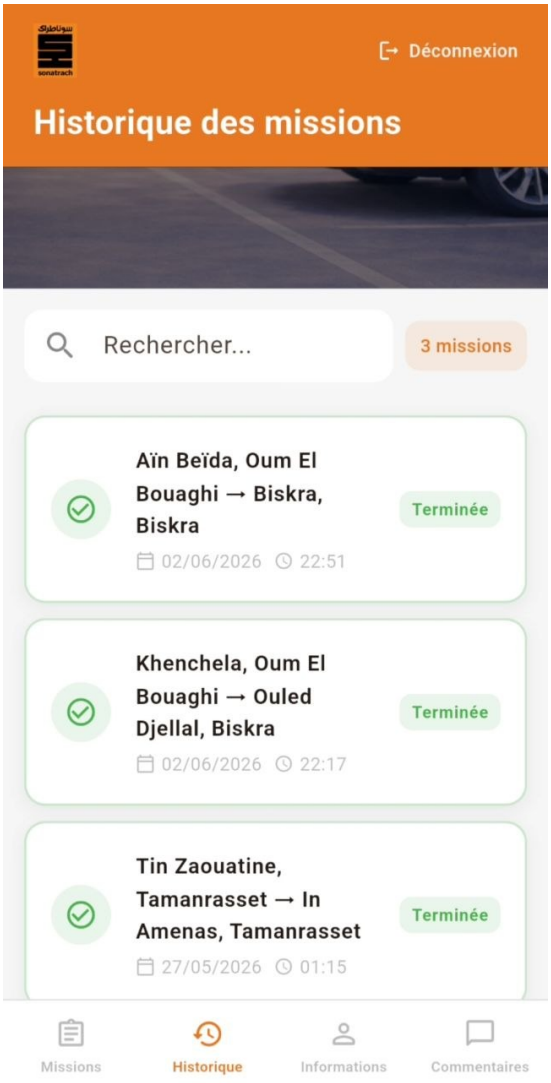
FIGURE 4.11 – les interfaces d'employe

### 4.6.5 Les interfaces de chauffeur :

L'espace Chauffeur se compose de quatre onglets : Missions, pour consulter les missions qui lui sont assignées ; Historique, retraçant l'ensemble de ses missions passées ; Informations, affichant ses données personnelles (nom, email, téléphone, type de permis et état de disponibilité) avec possibilité de les actualiser ; et Commentaires, permettant d'envoyer un commentaire au Chef de Parc



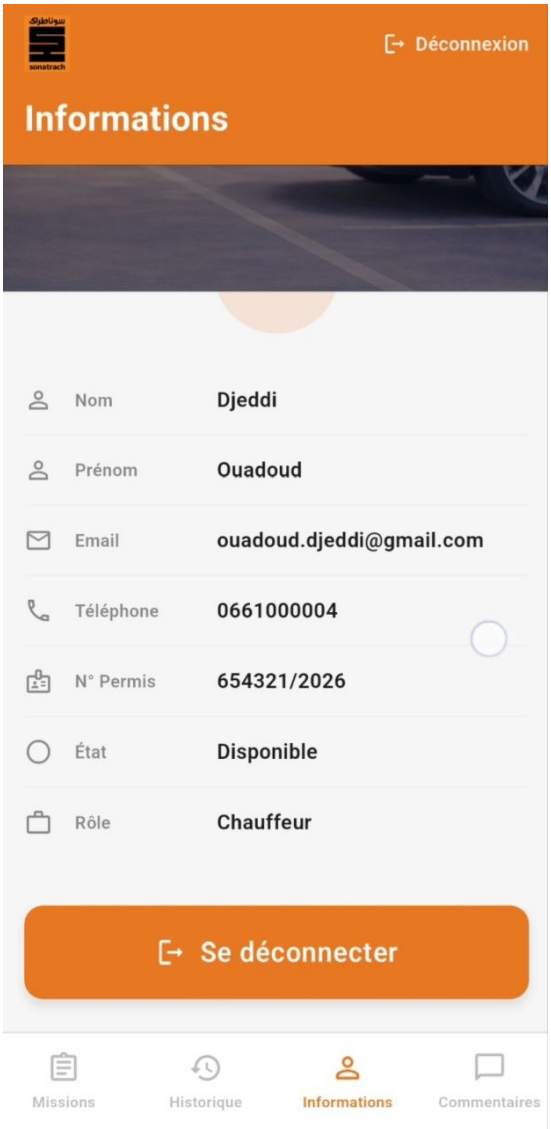

(a) Interface des missions




(b) Interface historique




(a) Interface d'envoi de commentaire




(b) Interface profil

FIGURE 4.13 – Les interfaces du chauffeur

# Conclusion Générale

Au terme de ce travail, nous avons mené à bien la conception et le développement d'une application mobile dédiée à la gestion des réservations de véhicules au sein de la division Exploration de Sonatrach. Ce projet nous a permis de mettre en pratique les connaissances acquises durant notre parcours académique, tout en nous confrontant à des problématiques réelles du monde professionnel.

L'étude de l'existant a révélé les limites du système actuel, notamment la dépendance aux méthodes manuelles, le manque de traçabilité et les difficultés de coordination entre les différents acteurs. Ces constats ont justifié la nécessité de proposer une solution informatisée capable d'améliorer la gestion des ressources et d'optimiser les processus de réservation.

La phase d'analyse et de conception, réalisée à l'aide du langage UML, nous a permis de modéliser le système de manière claire et structurée, en identifiant les besoins fonctionnels et non fonctionnels, ainsi que les interactions entre les différents intervenants. Cette étape a constitué une base solide pour le développement de l'application.

La mise en œuvre de la solution, en s'appuyant sur des technologies modernes telles que Flutter pour le développement mobile et Spring Boot pour le back-end, a permis de concevoir une application performante, ergonomique et adaptée aux besoins des utilisateurs. L'adoption de l'architecture MVVM a également contribué à assurer une meilleure organisation du code et à faciliter sa maintenance.

Ainsi, l'application développée apporte une amélioration significative en termes de gestion des réservations, en offrant une meilleure traçabilité, une réduction des délais de traitement et une optimisation de l'utilisation des véhicules et des chauffeurs.

Enfin, ce projet ouvre la voie à plusieurs perspectives d'amélioration .

## 4.7 Perspectives

Cette première version opérationnelle ouvre plusieurs pistes d'amélioration et d'extension :

- **Notifications en temps réel** Intégrer des notifications push pour alerter automatiquement les acteurs à chaque changement d'état d'une demande (soumission, validation, refus, affectation), afin de réduire les délais de traitement.
- **Géolocalisation et suivi des trajets**Un module de géolocalisation permettrait de suivre en temps réel la position des véhicules, d'optimiser les affectations selon la proximité des chauffeurs et de consulter l'historique des itinéraires.
- **Tableau de bord analytique**Mettre en place un tableau de bord affichant des indicateurs clés : taux d'utilisation des véhicules, délais de traitement, statistiques

des chauffeurs, pour appuyer la prise de décision stratégique.

- **Gestion de la maintenance**Ajouter un module de suivi technique des véhicules avec alertes automatiques sur les échéances d'entretien (vidange, révision, contrôle technique), pour prévenir les pannes et garantir la disponibilité du parc.
- **Déploiement à l'échelle de l'entreprise** Étendre l'application à l'ensemble des divisions de Sonatrach, avec une architecture multi-parc et un hébergement cloud sécurisé pour assurer la scalabilité et la disponibilité du service.
- **Intelligence artificielle pour les affectations** À terme, intégrer des algorithmes de recommandation pour suggérer automatiquement le chauffeur et le véhicule les plus adaptés à chaque mission, selon la disponibilité, la distance et l'historique

# Bibliographie


[1] Martin Fowler. *UML Distilled : A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Professional, 3e édition, 2003.

[2] Kurt Bittner, Ian Spence. *Use Case Modeling*. Addison-Wesley Professional, 1re édition, 2002.

[3] Pascal Roques. *UML 2 par la pratique*. Eyrolles, Paris, 7e édition, 2011.

[4] Gilad Bracha. *The Dart Programming Language Specification*. Addison-Wesley Professional, 1re édition, 2015.

[5] Herbert Schildt. *Java : The Complete Reference*. Oracle Press, New York, 11e édition, 2018.

[6] Flutter Documentation. Disponible sur : `https://docs.flutter.dev/`. Consulté le 10 mars 2026.

[7] Spring Boot Documentation. Disponible sur : `https://docs.spring.io/spring-boot/index.html`. Consulté le 10 mars 2026.

[8] Figma. Disponible sur : `https://www.figma.com/`. Consulté le 10 mars 2026.

[9] Google Stitch AI. Disponible sur : `https://stitch.withgoogle.com/`. Consulté le 10 mars 2026.

[10] Postman. Disponible sur : `https://www.postman.com/`. Consulté le 15 avril 2026.

[11] Scott Chacon, Ben Straub. *Pro Git*. Apress, 2e édition, 2014.

[12] Microsoft Visual Studio Code. Disponible sur : `https://code.visualstudio.com/`. Consulté le 15 mars 2026.

[13] Android Studio. Disponible sur : `https://developer.android.com/studio`. Consulté le 15 mars 2026.

[14] LaTeX Project. Disponible sur : `https://www.latex-project.org/`. Consulté le 01 mai 2026.

[15] phpMyAdmin. Disponible sur : `https://www.phpmyadmin.net/`. Consulté le 30 avril 2026.

[16] Apache Friends - XAMPP. Disponible sur : `https://www.apachefriends.org/`. Consulté le 30 avril 2026.

[17] Draw.io. Disponible sur : `https://www.diagrams.net/`. Consulté le 08 mars 2026.