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 : Systèmes d'informatique
N° de l'Arrêté d'habilitation de la spécialité : arrêté n °872 du 26/07/2016

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

Thème

Développement d'une application Android « Carnet d'entretien automobile »

Présenté par :
BOURZAM SALSABIL
ADJAL ADEM
TAFTIST MOUAAD ABDELLAH
ZERROUK HOURIA

Soutenu le .../.../ …Devant le jury composé de
BERRICHI ALI : Examinateur
BITIT RAHMOUNE : Encadreur

Dédicace :

Grâce à Dieu qui m'a permis de mener à bien ce travail,

Je dédie ce mémoire à ma chère mère et à mon cher père, dont l'amour, la patience et le soutien ont été ma force et mon inspiration à chaque étape de ma vie.
À mes frères bien-aimés, qui ont toujours été à mes côtés, partageant mes joies et mes défis, et m'encourageant à persévérer.
Et à mes amis proches, qui m'ont accompagné tout au long de mon parcours universitaire, m'apportant aide, soutien et encouragements, et rendant cette aventure plus riche et plus agréable

Salsabil

Je dédie ce mémoire à mes chers parents, ma mère et mon père, pour leur patience, leur amour, leur soutien et leurs encouragements.
Sans oublier tous les professeurs, que ce soit du primaire, du moyen, du secondaire ou de l'enseignement supérieur.

Mouaad Abdellah

Je dédie ce travail à mes chers parents, ma mère et mon père, ainsi qu'à mes grands-parents maternels, qui ne m'ont jamais ménagé leur amour, leur soutien et leurs encouragements.
J'adresse également mes remerciements profonds à tous mes professeurs, à toutes les étapes de mon parcours scolaire, pour leur savoir et leur inspiration.
Je n'oublie pas mes amis et mes frères, qui ont toujours été un soutien et un encouragement tout au long de mon chemin.

houria

Je dédie ce travail à mes chers parents, ma mère et mon père, ainsi qu'à mes grands-parents maternels, qui ne m'ont jamais ménagé leur amour, leur soutien et leurs encouragements.
J'adresse également mes remerciements profonds à tous mes professeurs, à toutes les étapes de mon parcours scolaire, pour leur savoir et leur inspiration.
Je n'oublie pas mes amis et mes frères, qui ont toujours été un soutien et un encouragement tout au long de mon chemin.

Adem

Remerciements :

Grâce à Dieu qui nous a accordé force, patience et volonté, nous avons pu réaliser ce modeste travail et trouver le chemin de la créativité et de l'accomplissement.

Nous exprimons notre profonde gratitude à notre encadreur M. RAHMOUNE BITIT, dont les conseils avisés, les orientations précieuses et les remarques constructives ont été un guide essentiel à chaque étape de l'élaboration de ce mémoire. Nous remercions également notre examinateur M.BERRICHI ALI pour avoir accepté d'évaluer notre travail et pour l'enrichir par ses remarques et suggestions précieuses, contribuant ainsi à en améliorer la qualité.

Nos remerciements s'étendent spécialement à nos parents, pour leur amour infini, leur patience, leurs sacrifices et toutes les valeurs qu'ils nous ont transmises. Ils ont été notre source de force et d'inspiration tout au long de ce parcours. À nos frères et sœurs qui ont toujours été à nos côtés dans les moments de défis et de joie, nous adressons également notre reconnaissance pour leur soutien et leur encouragement constants.

Enfin, nous tenons à remercier toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail. Leur présence et leur appui ont joué un rôle important dans l'aboutissement de ce projet.

Table des matières :

Introduction générale **1**
Chapitre I : Analyse des besoins **1**
1.1 Introduction 1
1.2 Carnet d'entretien du véhicule 1
1.2.1. Définition du carnet d'entretien 1
1.2.2. Composants du carnet d'entretien 1
1.3. Applications mobiles des carnets d'entretien 2
1.3.1. Importance des applications de carnet d'entretien 2
1.3.2 Problématique et limites des solutions traditionnelles : 2
1.3.3 Les applications de carnet d'entretient existantes: 3
1.4 Analyse des besoins des utilisateurs : 5
1.4.1 Besoins fonctionnels: 5
1.4.2 Besoins non fonctionnels : 5
1.5 Définition d'UML (Langage de modélisation unifié) : 5
1.5.1 Types de diagrammes UML 5
1.5.2 Rôle de l'UML dans le développement logiciel : 6
1.5.3 Phases d'utilisation de l'UML 6
1.6. Diagramme de cas d'utilisation **7**
1.6.1 Définition du diagramme de cas d'utilisation 7
1.6.2 Diagramme de cas d'utilisation de notre système 7
1.7 Description textuelle : **7**
1.7.1 Description textuelle du cas d'utilisation « Créer un compte » 8
1.7.2 Description textuelle du cas d'utilisation « Authentification » 8
1.7.3 Description textuelle du cas d'utilisation « créer un carnet » 9
1.7.4 c ..... **Erreur ! Signet non défini.**
1.7.5 Description textuelle du cas d'utilisation « Contacter un service » 10
Chapitre II : Conception **13**
2.1. Introduction 13
2.2. Diagrammes de séquence : 13
2.2.1 Définition du diagramme de séquence 13
2.2.2 Diagramme de séquence « Créer compte » 13
2.2.4 Diagramme de séquence « Consulte le carnet » 14
2.2.5 Diagramme de séquence « Ajouter le carnet » 15
2.2.6 Diagramme de séquence « Modifier le carnet » 15

2.2.7 Diagramme de séquence « Supprimer le carnet » 16
2.2.8 Diagramme de séquence « Gérer une révision » 17
2.3. Diagramme de classe : 17
2.3.1 Définition de diagramme de classe : 17
2.3.2 Dictionnaire de données : 17
2.3.3 Les règles de gestion : 19
2.3.4 Présentation de diagramme de classe : 20
2.3.5 Les règles de transformation vers le modèle relationnel : ..... **Erreur ! Signet non défini.**
2.4. Modèle Relationnel : 21
2.5. Conclusion : 21
Chapitre III : Implémentation **22**
3.1 Introduction 22
3.2 Environnement de développement 22
3.2.2 Environment logiciel 22
3.4 Présentation des interfaces graphiques 23
3.4.1 Authentification et inscription 24
3.4.2 Accueil 25
3.4.3 Historique 25
3.4.4 Ajout et Modification de véhicule : 26
3.4.6 Liste des véhicules et Consultation 27
3.4.7 Localisation et Partage 28
Conclusion générale **29**
Webographie : **30**

Liste des tableaux :

Tableau 1: Tableau comparative des applications mobile << Carnet d'entretien Auto>> 4
Tableau 2: Description textuelle du cas d'utilisation<<Créer un compte>> 8
Tableau 3: Descriptions textuelle du cas d'utilisation <<Authentification>> 8
Tableau 4: Description textuelle du cas d'utilisation<< créer du carnet>> 9
Tableau 5: Description textuelle du cas d'utilisation << consulter carnet >> 9
Tableau 6: Description textuelle du d'utilisation<<contacter un service>> 10
Tableau 7: Dictionnaire de données. 19

Liste des figures :

Figure 1:Diagramme de cas d'utilisation du système 7
Figure 2:Diagramme de séquence « créer un compte 13
Figure 3 : Diagramme de séquence « Authentification ». 14
Figure 4: Diagramme de séquence « Consulte le c 14
Figure 5:Diagramme de séquence « Ajouter le carnet ». 15
Figure 6:Diagramme de séquence « Modifier le carnet ». 15
Figure 7:Diagramme de séquence « Supprimer une révision » 16
Figure 8:Diagramme de séquence « Gérer une révision » 17
Figure 9: Diagramme de classe. 20
Figure 10: Logo de notre application 23
Figure 11: d'authentification 24
Figure 12:Interface d'inscription 24
Figure 13:récupération du mot de passe 24
Figure 14: La page d'accueil 25
Figure 15: Interface de l'historique des entretiens 25
Figure 16: Interface du carnet 26
Figure 17: Interface d'ajout de véhicule 26
Figure 18:modification 26
Figure 19:Interface de la liste des véhicules 27
Figure 20:Interface de consultation des détails du véhicule 27
Figure 21:Interface de localisation 28
Figure 22: Interface de partage de la position 28

Introduction générale

Dans un contexte marqué par une évolution technologique rapide, la numérisation est devenue un élément essentiel et incontournable dans les différents domaines de la vie quotidienne. Elle a largement contribué à l'amélioration de la qualité des services et à la facilitation de la gestion de diverses activités. Ce progrès a touché plusieurs secteurs, notamment celui de la gestion des services personnels, qui a connu une transformation significative grâce à l'adoption des solutions numériques.

Les applications mobiles représentent l'une des principales manifestations de cette transformation, en offrant des moyens efficaces pour organiser et suivre les tâches quotidiennes de manière simple et rapide. Parmi les domaines ayant bénéficié de ces avancées, on trouve celui de la maintenance automobile, où il est devenu nécessaire d'assurer un suivi régulier de l'état du véhicule afin de préserver ses performances et prolonger sa durée de vie.

Dans le cadre de projet de fin d'étude, nous avons choisi de développer une application mobile Android dédiée à la gestion de l'entretien automobile, afin de répondre aux besoins croissants des utilisateurs en matière d'organisation et de suivi de leurs véhicules.
L'objectif principal de notre projet est de développer une une solution simple, pratique et efficace permettant à l'utilisateur de centraliser toutes les informations relatives à son véhicule en un seul endroit. En effet, l'application offre la possibilité d'enregistrer les différentes opérations de maintenance, telles que les vidanges, les révisions périodiques, le changement de pièces et les réparations diverses.

En outre, l'application propose un système de rappel intelligent qui notifie l'utilisateur des échéances importantes, contribuant ainsi à éviter les oublis et à assurer un entretien régulier du véhicule. Elle intègre également une fonctionnalité de géolocalisation basée sur Google Maps, permettant de localiser facilement les stations-service, les garages et autres services automobiles à proximité.

Par ailleurs, une attention particulière a été accordée à la conception de l'interface utilisateur, afin de garantir une utilisation simple, intuitive et accessible à tous les profils d'utilisateurs. L'application vise ainsi à améliorer l'expérience utilisateur tout en facilitant la gestion quotidienne du véhicule.

Ce mémoire est structuré en trois chapitres principaux :

- Le premier chapitre est consacré à la présentation du projet et à l'analyse des besoins.
- Le deuxième chapitre porte sur la conception du système à l'aide des diagrammes UML de séquence et de classes.
- Le troisième chapitre est consacré à la phase d'implémentation en présentant l'environnement de développement, les outils utilisés ainsi que les différentes fonctionnalités offertes par l'application.

Chapitre I : Analyse des besoins

1.1 Introduction

Ce chapitre est consacré à la présentation de notre projet de manière systématique, en exposant son cadre général et en analysant soigneusement les exigences du système. Il inclut également l'étude des besoins et des attentes des utilisateurs afin de déterminer les caractéristiques et les fonctionnalités sur lesquelles l'application s'appuiera pour garantir une performance efficace et une bonne réactivité dans différents contextes d'utilisation.

1.2 Carnet d'entretien du véhicule

1.2.1. Définition du carnet d'entretien

Le carnet d'entretien est un document officiel utilisé pour organiser et suivre la maintenance technique d'un véhicule. Ce carnet permet d'enregistrer périodiquement l'état du véhicule conformément aux calendriers fixés par le constructeur ou aux recommandations des ingénieurs de maintenance, ce qui contribue à préserver les performances et la sécurité du véhicule [1].

1.2.2. Composants du carnet d'entretien

1. Les pièces mécaniques et leurs fonctions

Le carnet d'entretient contient une liste des différentes pièces du véhicule qui sont classées en deux catégories :

- **Essentielles :** comme les freins, le moteur, les pneus et les rétroviseurs. Elles sont des composants qui influencent directement la sécurité du véhicule et de ses utilisateurs.
- **Non-essentielles :** des pièces qui n'affectent pas directement la sécurité, comme certains accessoires intérieurs ou éléments décoratifs

Chaque pièce possède des caractéristiques spécifiques telles que : le nom, la durée de vie, les spécifications techniques, le numéro de référence.

2. Informations de base du véhicule :

Des informations qui concernent le véhicule comme :

- La marque
- Le modèle
- Le type de véhicule
- Le numéro de châssis
- Le numéro d'immatriculation
- La date de première mise en circulation

3. Fonctions principales du carnet d'entretien

1. Suivi de la durée de vie des pièces : Le carnet indique le moment où chaque pièce doit être remplacée en fonction de sa durée de vie afin d'éviter les pannes ou les accidents.

2. Enregistrement des opérations de maintenance : Chaque opération d'entretien, remplacement de pièce ou réparation est enregistrée avec la date et des notes explicatives.

3. Planification des inspections futures : Le carnet fournit un calendrier indiquant les dates des futures opérations de maintenance selon les intervalles planifiés

4. Amélioration de la gestion du véhicule : Il aide le propriétaire à savoir ce qui a été fait et ce qui doit encore être réalisé, réduisant ainsi la probabilité de pannes soudaines ou d'accidents causés par l'usure des pièces.

1.3. Applications mobiles des carnets d'entretien

1.3.1. Importance des applications de carnet d'entretien

Les applications mobiles de carnet d'entretien automobile constituent un outil moderne et efficace pour la gestion des véhicules. Elles permettent aux utilisateurs de suivre toutes les opérations d'entretien et de réparation de manière précise et organisée. Ce type d'applications offre les avantages suivants :

1. **Assurer la sécurité du véhicule et des utilisateurs** : Permet une surveillance continue des pièces essentielles du véhicule et réduit les risques d'accidents causés par des défaillances techniques, protégeant ainsi le conducteur, les passagers et les piétons.
2. **Conformité légale :** Aide à éviter les problèmes juridiques tels que les amendes, les poursuites judiciaires ou même l'emprisonnement pouvant résulter de l'utilisation d'un véhicule avec des pièces endommagées ou défectueuses.
3. **Encourager l'entretien régulier** : Incite les utilisateurs à effectuer la maintenance à temps et à respecter les intervalles recommandés pour chaque pièce du véhicule et contribue à prolonger la durée de vie du véhicule et à assurer des performances optimales.

1.3.2 Problématique et limites des solutions traditionnelles :

Malgré l'importance du carnet d'entretien papier, plusieurs limites apparaissent :

- **Risque de perte ou de détérioration :** le carnet peut être perdu ou endommagé.
- **Omissions ou erreurs :** les utilisateurs peuvent oublier d'enregistrer certaines opérations de maintenance.
- **Absence de rappels automatiques :** il n'existe aucun système d'alerte pour les entretiens à venir.
- **Difficulté d'accès aux informations :** il est difficile de consulter rapidement toutes les informations ou d'y accéder à distance.
- **Historique limité :** il n'est pas possible de stocker des photos, des factures ou d'autres documents liés à la maintenance.

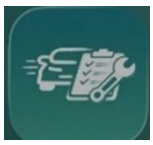
Ces limites rendent le carnet d'entretien papier moins pratique et moins fiable, ce qui justifie la création d'une application mobile moderne.

1.3.3 Les applications de carnet d'entretient existantes :



1. MY Garage – Car Assistant : Application pour suivre la maintenance de votre voiture et enregistrer toutes les interventions et dépenses, avec des rappels pour les opérations importantes comme le changement d'huile ou le renouvellement de l'assurance [2]. Cette application est caractérisée par :

- **Gestion du carburant** : enregistrement de chaque remplissage de carburant et calcul de la consommation du véhicule.
- **Suivi de la maintenance** : enregistrement de toutes les opérations de maintenance avec envoi de rappels.
- **Gestion des dépenses** : suivi de toutes les dépenses liées au véhicule.
- **Rapports et statistiques** : affichage de graphiques sur la consommation de carburant et les dépenses.
- **Gestion de plusieurs véhicules** : possibilité d'ajouter plusieurs voitures ou motos.



2. CarLog – Car Maintenance Tracker : Application pour suivre et gérer facilement la maintenance de votre voiture, enregistrer les services, coûts et kilométrage, avec rappels automatiques pour les entretiens importants [3]. Cette application est caractérisée par :

- **Suivi de la maintenance** : enregistrement de toutes les opérations de maintenance et réparations.
- **Gestion des dépenses** : suivi des dépenses liées au véhicule.
- **Suivi** de **la consommation** de **carburant** : enregistrement de chaque remplissage et analyse de la consommation.
- **Rappels de maintenance** : notifications pour rappeler les opérations de maintenance à venir.
- **Gestion de plusieurs véhicules** : possibilité de suivre plusieurs voitures dans.



3. Drivvo – Gestion des véhicules : Application pour suivre et gérer facilement les coûts et l'entretien de véhicule (carburant, maintenance, dépenses, rappels et statistiques) [4]. Les caractéristiques de cette application sont :

- **Gestion du carburant** : enregistrement de chaque remplissage de carburant et calcul de la consommation du véhicule.
- **Suivi de la maintenance** : enregistrement de toutes les opérations de maintenance avec envoi de rappels.
- **Gestion des dépenses** : suivi de toutes les dépenses liées au véhicule (carburant, maintenance)
- **Rapports et statistiques** : affichage de graphiques sur la consommation de carburant et les dépenses



4. TRACK (VehicleTrack) : Application qui permet de suivre la voiture ou la flotte de véhicules en temps réel (route, trajets, localisation et données d'efficacité) pour mieux gérer les déplacements [5]. Elle offre les fonctionnalités suivantes :

- **Suivi en temps réel via GPS** : localisation de chaque véhicule directement sur la carte.

- **Analyse des trajets** : affichage des parcours, distances parcourues et durée des trajets.
- **Alertes et notifications** : notifications en cas de dépassement de vitesse, sortie de zone ou arrêt inattendu.
- **Gestion d'un parc automobile** : adapté aux entreprises disposant de plusieurs véhicules.
- **Rapports et statistiques** : analyse de la performance des véhicules et des conducteurs, gestion efficace du parc automobile.

Le tableau suivant résume les caractéristiques des applications mobiles étudiées :

Application :	**Avantages :**	**Points faibles :**
Drivvo	• Interface claire et facile à utiliser. • Possibilité de gérer plusieurs véhicules • Adapté à un usage personnel et familial	• Personnels, pas pour flottes. • Certaines fonctions payantes. • Pas de suivi GPS en temps réel.
CarLog	• Application simple et légère • Possibilité de gérer plusieurs véhicules	• Interface simple et peu avancée. • Pas de suivi GPS. • Non adaptée aux parcs automobiles
My Garage	• Interface simple et intuitive. • Possibilité de gérer plusieurs véhicules facilement.	• Fonctionnalités limitées pour entreprises. • Pas de suivi en temps réel ni analyse des trajets. • Moins disponible sur Android
VehicleTrack	• Suivi en temps réel via GPS • Analyse détaillée des trajets et performances des véhicules	• Conçu pour les entreprises, pas pour les particuliers. • Certaines fonctionnalités payantes. • Gestion limitée du carburant et des dépenses

Tableau 1: Tableau comparative des applications mobile << Carnet d'entretien Auto>>

Synthèse *:*

Les applications étudiées offrent des interfaces simples, des rappels automatiques et la gestion de plusieurs véhicules. Elles présentent cependant des limites, comme la gestion d'un seul véhicule pour certaines, l'absence de fonctionnalités essentielles et la nécessité d'une version payante pour accéder à toutes les options. Globalement, elles restent utiles mais peuvent être améliorées pour un suivi plus complet.

Pour notre application, nous prévoyons d'ajouter de nouvelles fonctionnalités inédites qui ne sont pas présentes dans les applications existantes :

- Notifications automatiques pour les rendez-vous de maintenance, adaptées à l'état du véhicule
- Possibilité de partager temporairement les données du véhicule pour que la famille ou les employés puissent les consulter en cas d'incident.
- Analyse visuelle des distances parcourues, de la consommation et des trajets, facile à comprendre pour l'utilisateur.
- Notifications sur l'état météo pour aider le conducteur à planifier ses trajets en fonction du climat.

Ces améliorations visent à rendre l'application plus complète, interactive, et adaptée aux besoins réels des utilisateurs.

1.4 Analyse des besoins des utilisateurs :

Pour concevoir une application de gestion de l'entretien des véhicules, les besoins des utilisateurs se divisent en deux catégories :

1.4.1 Besoins fonctionnels :

- Création et gestion sécurisée des comptes utilisateurs.
- Gestion complète des véhicules : ajout, modification, suppression.
- Gestion des pièces et de leurs caractéristiques (référence, date d'achat, durée de vie…).
- Consultation de l'état du véhicule et de son carnet de maintenance.
- Notifications automatiques pour les entretiens et la fin de vie des pièces.
- Exportation des données pour partage ou archivage.

1.4.2 Besoins non fonctionnels :

- Sécurité et protection des données.
- Ergonomie et facilité d'utilisation.
- Performance et réactivité du système.
- Stockage sécurisé et sauvegarde des données.
- Compatibilité avec les systèmes Android et évolutivité future.

1.5 Définition d'UML (Langage de modélisation unifié) :

UML (Langage de modélisation unifié) : est un langage de modélisation standardisé, reconnu mondialement dans le domaine du développement logiciel. UML permet de représenter visuellement un système à travers des diagrammes, facilitant ainsi la communication entre développeurs, analystes et utilisateurs. Il transforme les besoins des utilisateurs en modèles clairs et compréhensibles et aide à planifier efficacement la conception et la réalisation d'une application [6].

1.5.1 Types de diagrammes UML

Les diagrammes UML se divisent en deux grandes catégories :

1. Diagrammes statiques :

Ces diagrammes représentent la structure du système et ses composants sans considérer l'évolution dans le temps. Ils permettent de :

- Visualiser les classes et leurs attributs.
- Montrer les relations entre les classes (associations, héritages, dépendances).
- Illustrer l'architecture générale du système.

2. Diagrammes dynamiques :

Ces diagrammes illustrent le comportement du système face aux événements et aux interactions avec l'utilisateur ou l'environnement. Ils permettent de :

- Comprendre les flux d'interaction entre acteurs et système.
- Montrer la séquence des opérations.
- Étudier les scénarios possibles de fonctionnement.

1.5.2 Rôle de l'UML dans le développement logiciel :

UML fournit aux développeurs un ensemble de techniques et de bonnes pratiques à appliquer tout au long du cycle de vie d'un logiciel, de l'analyse à l'implémentation. Il permet de:

- Clarifier les besoins des utilisateurs.
- Définir les fonctionnalités et contraintes du système.
- Organiser la logique et l'architecture logicielle.
- Faciliter la communication entre les équipes de développement et les clients

1.5.3 Phases d'utilisation de l'UML

L'utilisation d'UML se fait généralement en trois phases :

Phase d'analyse : cette phase consiste à:

- Décrire en détail les cas d'utilisation, c'est-à-dire les interactions possibles entre le système et ses utilisateurs.
- Créer les diagrammes de séquence pour visualiser l'enchaînement des actions.
- Identifier clairement les besoins fonctionnels et non fonctionnels.

Phase de conception : C'est la deuxième phase de cycle de développement avec UML. Elle consiste à :

- Décrire la logique interne du système et la façon dont les composants interagissent
- Produire des diagrammes de classes et des diagrammes de séquence détaillés.
- Préparer la structure qui servira pour l'implémentation.

Phase d'implémentation : La dernière phase consiste à:

- Mettre en œuvre les fonctionnalités prévues en respectant la conception.
- Vérifier que le logiciel respecte bien les besoins identifiés et les scénarios d'utilisation.

1.6. Diagramme de cas d'utilisation

1.6.1 Définition du diagramme de cas d'utilisation : Les diagrammes de cas d'utilisation présentent les fonctionnalités générales du système. Ils indiquent toutes les interactions entre le système et les acteurs (utilisateurs), montrant ce que fait le système et comment les acteurs l'utilisent, de manière générale, sans détailler le fonctionnement interne du système. Le diagramme de cas d'utilisation est utilisé lors de la spécification des besoins pour présenter le comportement attendu du système avec son environnement [7]

1.6.2 Diagramme de cas d'utilisation de notre système:

La figure suivante représente le diagramme de cas d'utilisation de notre application. Elle montre l'acteur (propriétaire du véhicule) et leur interactions avec le système.

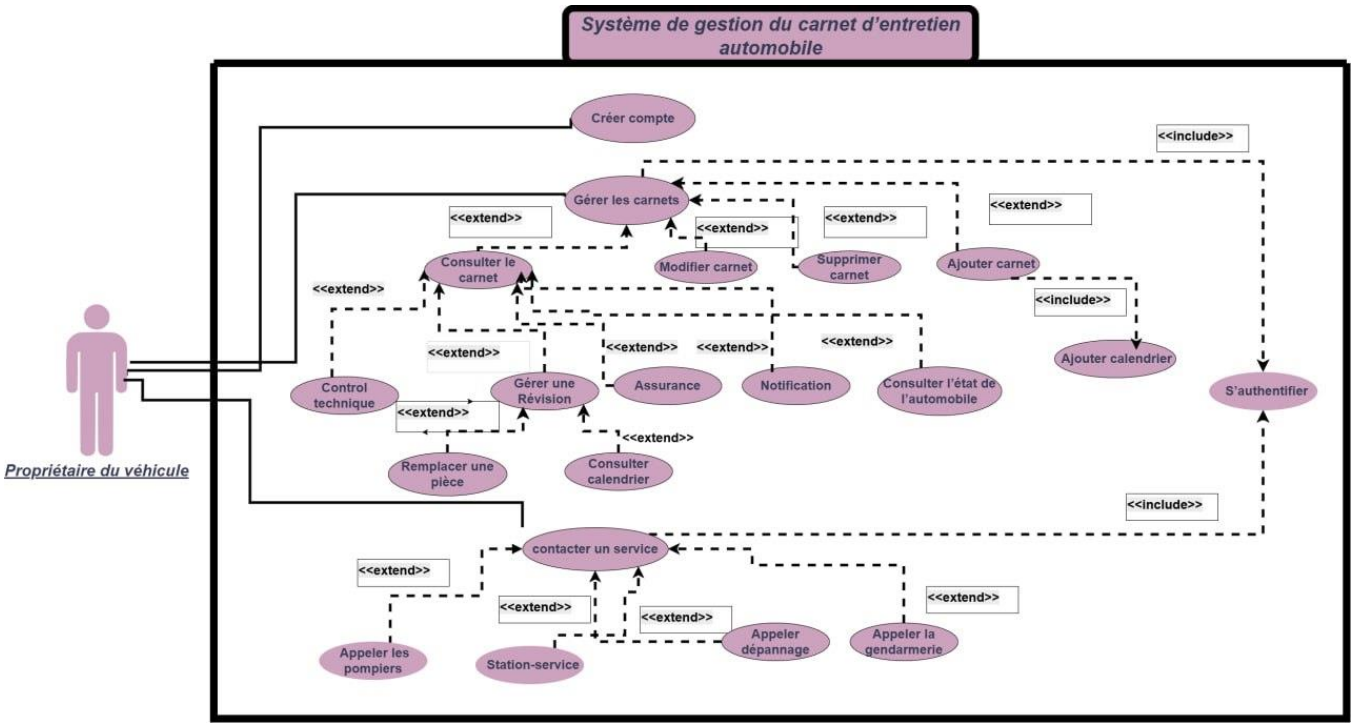


Figure 1:Diagramme de cas d'utilisation du système

1.7 Description textuelle :

Dans cette partie, nous allons expliquer de manière textuelle les fonctionnalités du système pour chaque cas d'utilisation en utilisant des tableaux.

1.7.1 Description textuelle du cas d'utilisation « Créer un compte »

Cas d'utilisation : créer un compte	
Titre	Créer un compte
But	Permettre à l'utilisateur de créer un nouveau compte
Acteur	Propriétaire du véhicule
Préconditions	Avoir un email valide et connexion Internet
Postconditions	Compte créé avec succès
Enchaînement nominal	
1. Ouvrir l'application 2. Cliquer sur créer compte 3. Saisir les informations 4. Valider 5. Compte crée	
Enchaînement alternatif	
1. À l'étape 4 ; si les informations saisies sont incorrectes, le système affiche un message d'erreur et revient l'étape 3.	

Tableau 2: Description textuelle du cas d'utilisation<<Créer un compte>>

1.7.2 Description textuelle du cas d'utilisation « Authentification »

Cas d'utilisation : Authentification	
Titre	Authentification
But	Permettre à l'utilisateur d'accéder à son compte
Acteur	Propriétaire du véhicule
Préconditions	Compte déjà créé
Postconditions	Accès au système
Enchaînement nominal	
1. Entrer email et mot de passe 2. Vérification 3. Accès autorisé	
Enchaînement alternatif	
1. À l'étape 2 ; si les informations saisies sont incorrectes, le système affiche un message d'erreur et revient à l'étape 1.	

Tableau 3: Descriptions textuelle du cas d'utilisation <<Authentification>>

1.7.3 Description textuelle du cas d'utilisation « Gérer les carnets »

Cas d'utilisation : Gérer les carnets	
Titre	Gérer les carnets
But	Gérer les informations du carnet d'entretien
Acteur	Propriétaire du véhicule
Préconditions	Être connecté
Postconditions	Carnet mis à jour
Enchaînement nominal	
1. L'utilisateur accède au carnet. 2. Le système affiche les options (consulter, ajouter, modifier). 3. L'utilisateur choisit une action. 4. Le système exécute l'action demandée. 5. Le système met à jour les informations si nécessaire	
Enchaînement alternatif	
1. À l'étape 3, si aucune donnée n'existe, le système affiche un message d'erreur	

Tableau 4: Description textuelle du cas d'utilisation<< créer du carnet>>

1.7.4 Description textuelle du cas d'utilisation « Consulter le carnet »

Cas d'utilisation : consulter le carnet	
Titre	Consulter le carnet
But	Voir les informations du véhicule
Acteur	Propriétaire du véhicule
Préconditions	Le carnet est disponible
Postconditions	Informations affichées
Enchaînement nominal	
1. L'utilisateur sélectionne consulter le carnet. 2. Le système récupère les informations du carnet. 3. Le système affiche les informations.	
Enchaînement alternatif	
1. À l'étape 2, si le carnet n'existe pas, le système affiche un message d'erreur.	

Tableau 5: Description textuelle du cas d'utilisation << consulter carnet >>

1.7.5 Description textuelle du cas d'utilisation « Contacter un service »

Cas d'utilisation : Contacter un *service*	
Titre	Contacter un service
But	Permettre à l'utilisateur de joindre des services d'assistance ou d'urgence
Acteur	Propriétaire du véhicule
Préconditions	L'utilisateur doit être connecté au système (Authentifié).
Postconditions	Mise en relation avec le service demandée.
Enchaînement nominal	
1. L'utilisateur accède à la rubrique "Services". 2. Le système affiche les services disponibles (Pompiers, Gendarmerie, etc.). 3. L'utilisateur sélectionne le service souhaité. 4. Le système établit la connexion (Appel ou affichage des coordonnées)	
Enchaînement alternatif	

Tableau 6: Description textuelle du d'utilisation<<contacter un service>>

1.8. Conclusion :

Dans ce chapitre, nous avons présenté notre projet en détaillant ses besoins fonctionnels et non fonctionnels. Nous avons ensuite procédé à l'analyse des besoins de notre future application à travers le diagramme UML de cas d'utilisation ainsi que leurs descriptions textuelles. Cette étude nous a permis de clarifier le fonctionnement attendu de l'application et de préparer la phase suivante. Le chapitre suivant sera consacré à la conception de notre travail.

Chapitre II : Conception

2.1. Introduction

Après avoir défini les grandes lignes des besoins et des exigences de notre système, nous passons maintenant à une phase fondamentale du cycle de vie du système, visant à analyser et concevoir le système de manière plus détaillée et précise. Ce chapitre a pour objectif de formaliser les besoins et comportements attendus du système de manière méthodique, en utilisant les diagrammes de séquence et les diagrammes de classes, afin d'assurer la clarté des processus et des interactions entre les utilisateurs et le système.

2.2. Diagrammes de séquence :

2.2.1 Définition du diagramme de séquence

Le diagramme de séquence est l'un des diagrammes d'interaction fondamentaux en ingénierie logicielle, montrant l'ordre d'exécution des opérations entre le système et les acteurs de manière temporelle précise. Ce diagramme permet de comprendre le déroulement interne des opérations du système et de définir les interactions entre les différents composants, ce qui améliore la qualité de la conception et assure sa conformité aux besoins fonctionnels [8].

2.2.2 Diagramme de séquence « Créer compte »

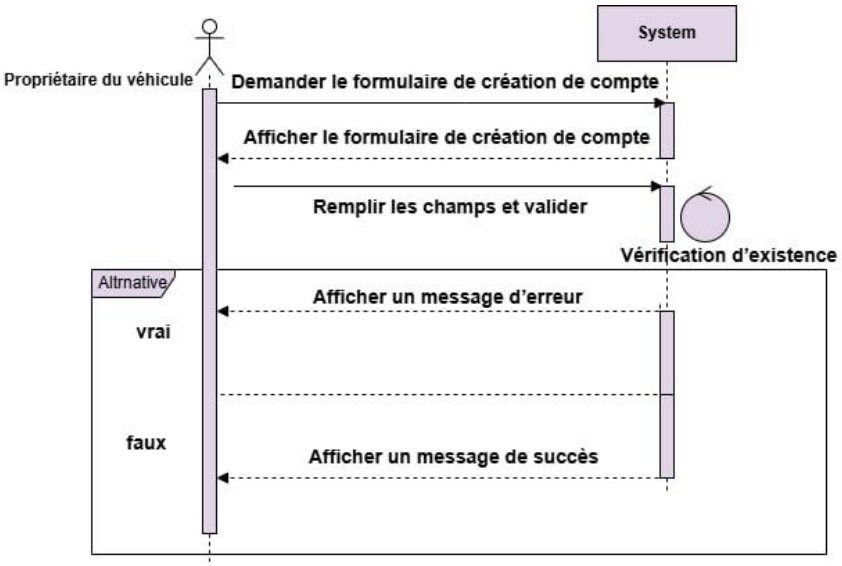


Figure 2:Diagramme de séquence « créer un compte

2.2.3 Diagramme de séquence "Authentification"

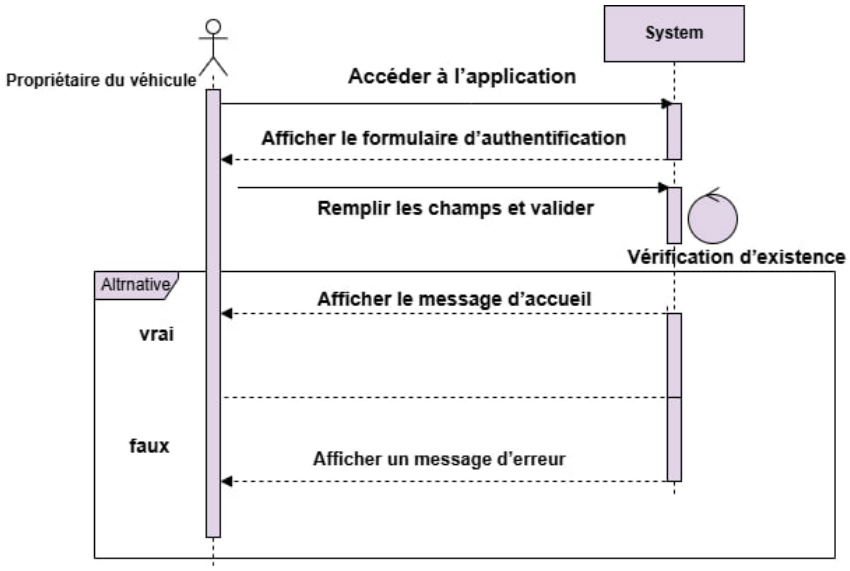


Figure 3 : Diagramme de séquence « Authentification ».

2.2.4 Diagramme de séquence « Consulte le carnet »

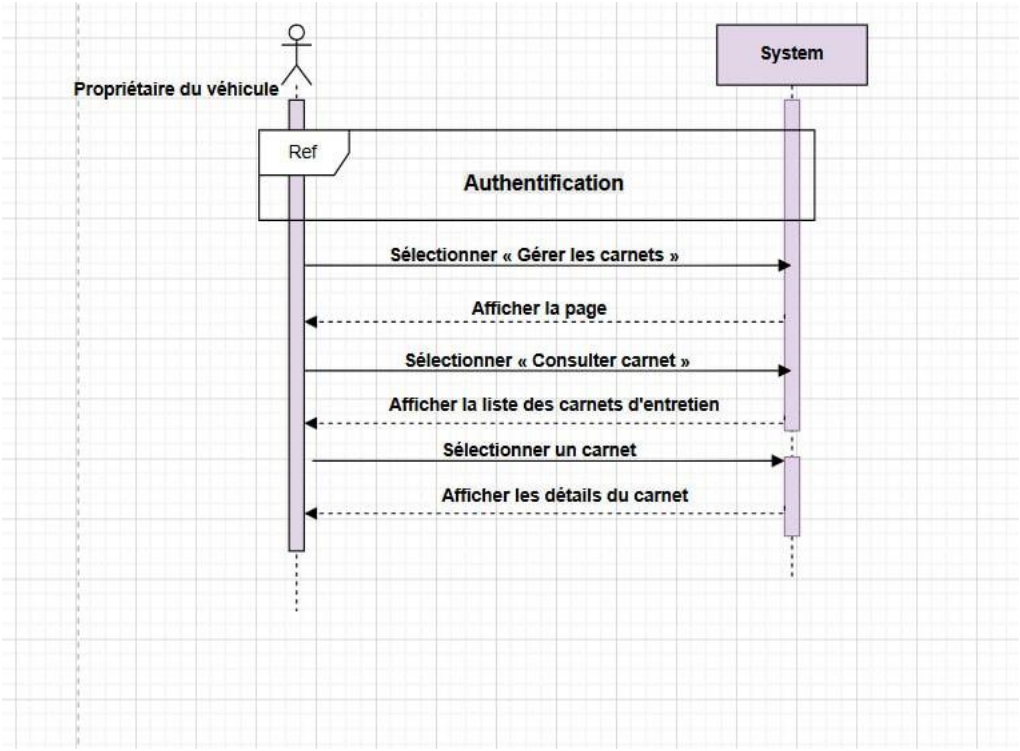


Figure 4: Diagramme de séquence « Consulte le c

2.2.5 Diagramme de séquence « Ajouter le carnet »

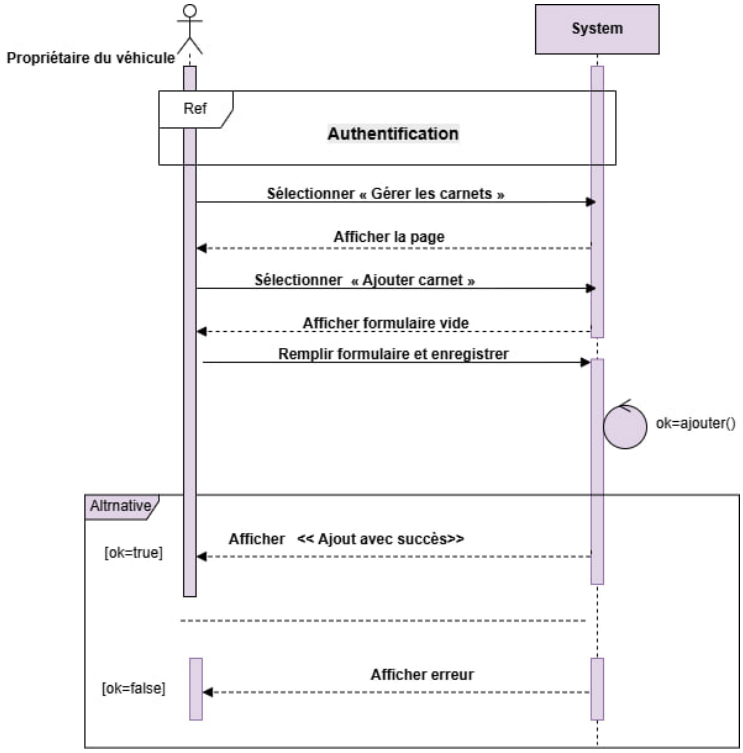


Figure 5:Diagramme de séquence « Ajouter le carnet ».

2.2.6 Diagramme de séquence « Modifier le carnet »

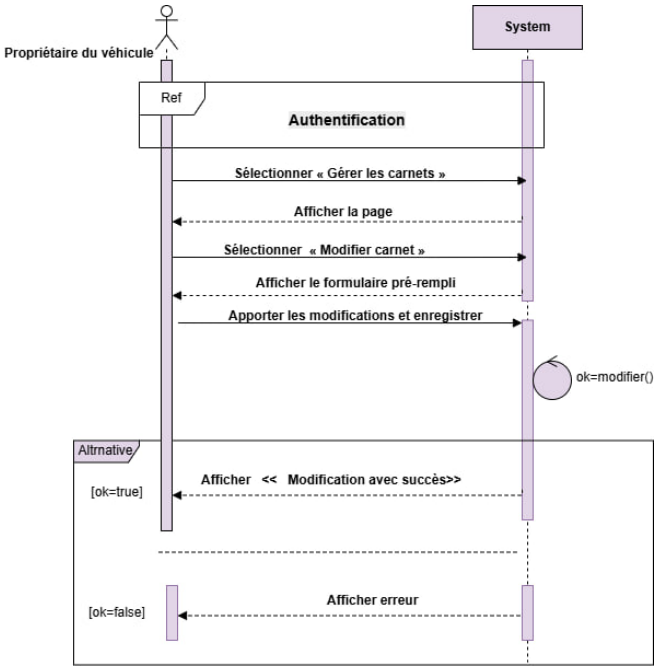


Figure 6:Diagramme de séquence « Modifier le carnet ».

2.2.7 Diagramme de séquence « Supprimer le carnet »

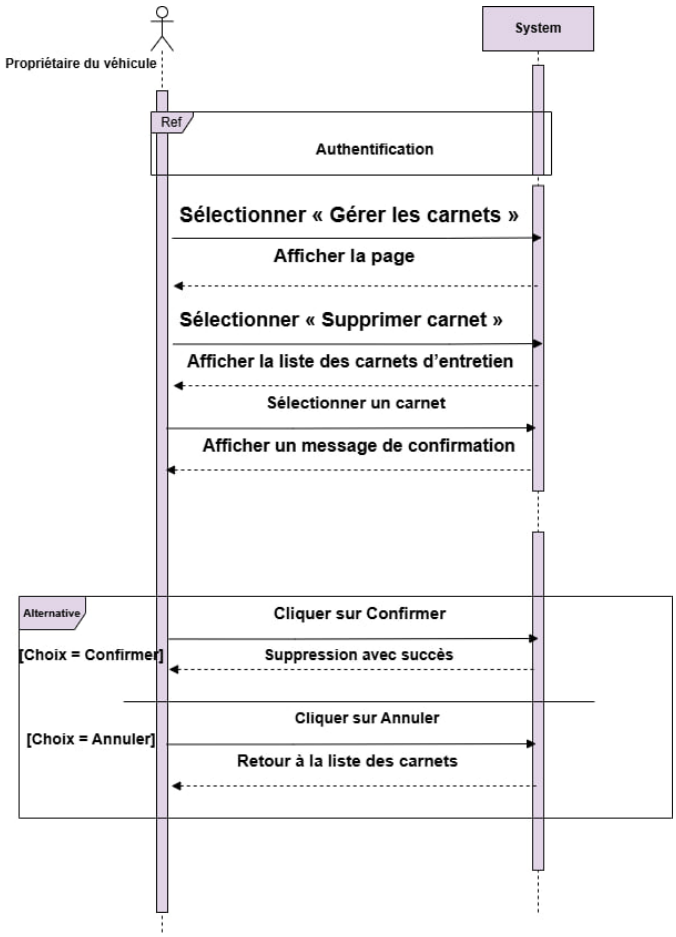


Figure 7:Diagramme de séquence « Supprimer une révision »

2.2.8 Diagramme de séquence « Gérer une révision »

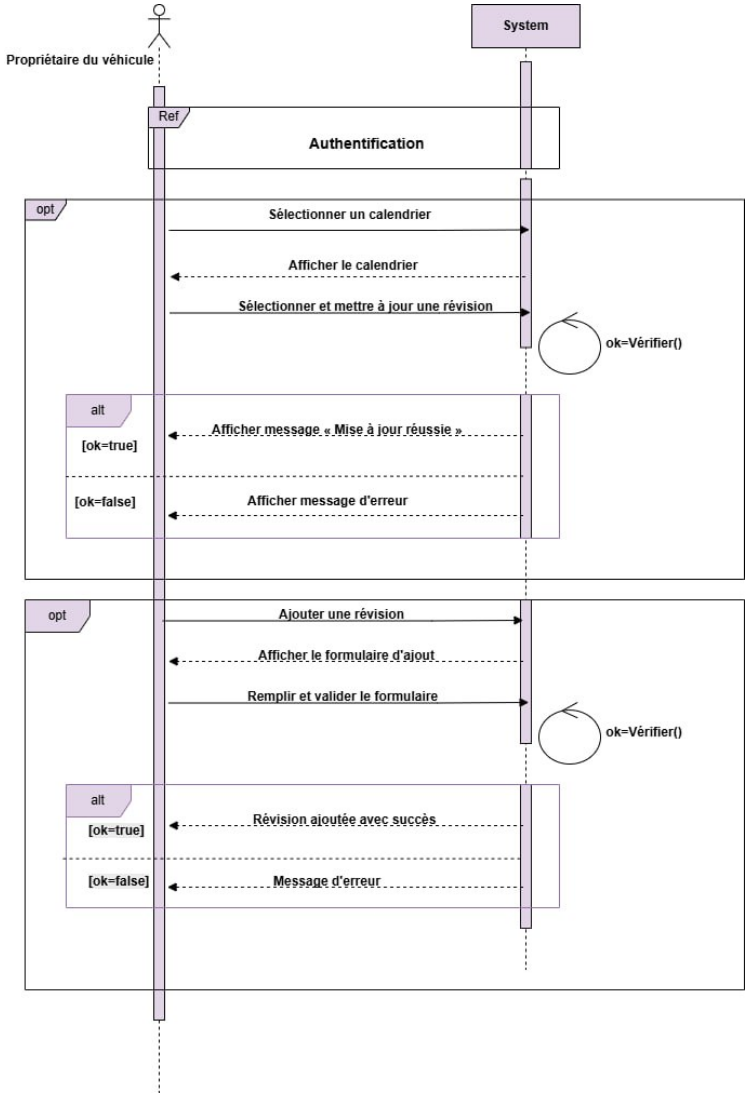


Figure 8:Diagramme de séquence « Gérer une révision »

2.3. Diagramme de classe :

2.3.1 Définition de diagramme de classe :

Le diagramme de classes représente la structure interne de notre système. Il constitue un élément fondamental dans la conception et la modélisation de notre application, car il permet de décrire les différentes classes, leurs attributs, ainsi que les relations entre elles [9].

2.3.2 Dictionnaire de données :

Toutes les données collectées et identifiées sont saisies dans le dictionnaire de données, qui constitue une référence essentielle pour comprendre la structure du système. Le dictionnaire contient le nom de la classe, les attributs, la description, le type de données et la taille, ce qui

facilite le processus de conception et permet de construire une base de données cohérente et fiable.

Class :	Attributs :	Désignation :	Type :	Taille
Utilisateur	id_util	Id d'utilisateur	Intger	10
	nom	Nom du propriétaire	String	10
	email	Prénom du propriétaire	String	10
	motDepasse	Mot de passe	Chaîne de caractères	10
	role	Type d'utilisateur (ex: Admin, Client)	String	10
Carnet	Id_carnet	Identifiant unique du carnet/véhicule.	Intger	10
	immatriculation	Immatriculation du véhicule	Chaîne de caractères	10
	Marque	Marque du véhicule	Chaîne de caractères	10
	Model	Modèle du véhicule	String	10
	dateCreation	Date d'ajout du véhicule dans le système.	Date	10
	etat	État actuel (ex: En marche, En panne).	String	10
	titreEvenement	Libellé de l'événement	String	10
	descriptionEvenement	Détails sur l'événement.	String	10
	dateDebutEvenement	Date de début d'un événement lié au véhicule	Date	10
	dateFinEvenement	Date de fin d'un événement.	Date	6
Revision	Id_rev	Identifiant unique de la révision.	Int	10
	Type	Type de révision (ex: Vidange, Freins).	Chaîne de caractères	10
	coût	Montant de la révision.	Double	10
	description	Notes techniques sur l'intervention.	Date	14
	date	Date de l'intervention.	Date	8
	Id_assru	Identifiant unique du contrat.	Int	10
	dateDebut	Date de prise d'effet.	Date	10

Assurance	dateFin	Date d'expiration.	Date	10
	compagnie	Nom de l'assureur.	string	10
	numeroPolice	Référence du contrat d'assurance.	string	10
ControleTechnique	Id_ct	Identifiant unique du contrôle.	Int	10
	datePassage	Date à laquelle le contrôle a été fait	Date	10
	resultat	Résultat (ex:Favorable, Défavorable)	String	10
	prochaineEcheance	Date du prochain contrôle obligatoire.	Date	10
Notification	Id_notif	Identifiant du message	Int	10
	message	Contenu de l'alerte.	String	10
	date	Date d'envoi.	Date	7
	estLue	État de lecture (Vrai/Faux).	Boolean	10
Service	Id_serv	Identifiant du service tiers.	Int	10
	nom	Nom de l'établissement.	String	10
	adresse	Localisation physique.	Chaîne de caractères	10
	telephone	Contact téléphonique.	String	10
	type	Catégorie (Gendarmerie, Pompier, etc.)	String	10
Gendarmerie	brigade	Nom de la brigade spécifique.	String	50
	secteur	Zone de compétence	String	50
StationService	carburantsDispo	Types de carburants (ex : Sans Plomb, Diesel)	String	100
Depannage	zoneIntervention	Rayon d'action du dépanneur	String	10

Tableau 7: Dictionnaire de données.

2.3.3 Les règles de gestion :

Notre système définit un ensemble de règles de gestion essentielles pour organiser les données et garantir le bon déroulement des opérations :

- Chaque carnet est associé à un seul véhicule et un seul propriétaire.
- Un propriétaire peut posséder plusieurs carnets, et chaque carnet peut contenir plusieurs pièces.
- Chaque pièce existe dans un seul carnet.
- Plusieurs utilisateurs peuvent localiser plusieurs stations-service.
- Chaque pièce peut être ajoutée, modifiée, consultée ou remplacée dans un carnet.

2.3.4 Présentation de diagramme de classe :

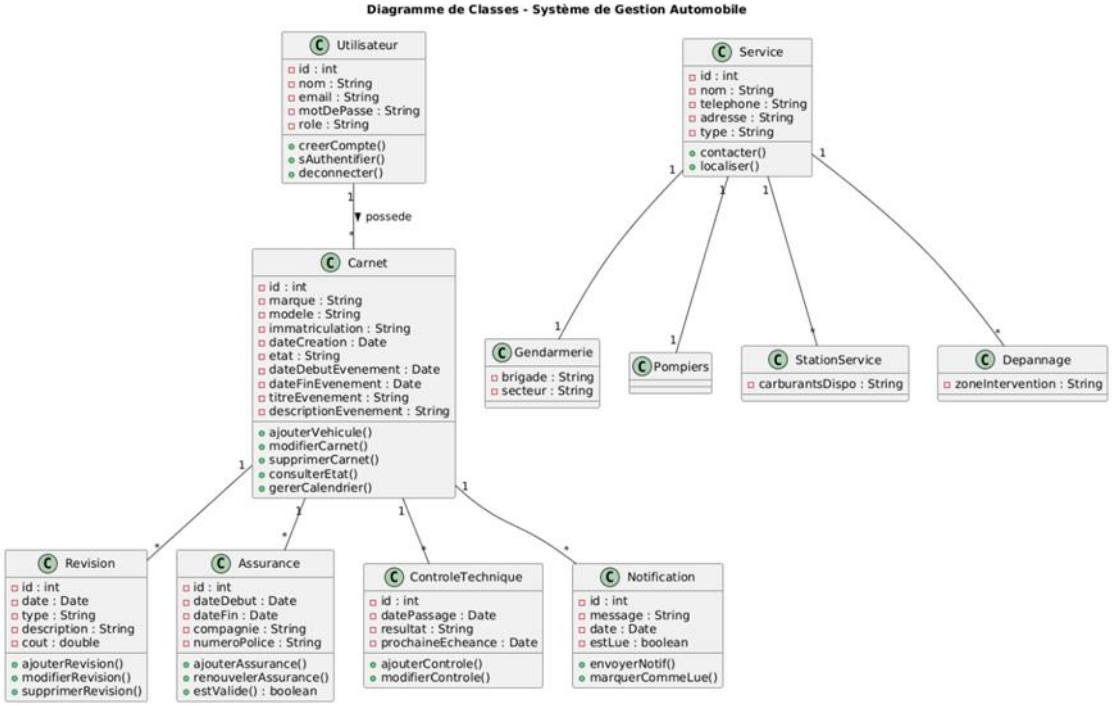


Figure 9: Diagramme de classe.

2.3. Règles de passage du Diagramme de Classes au Modèle Relationnel

. Pour obtenir un modèle relationnel cohérent et normalisé, nous appliquons les règles de transformation UML suivantes :

- **Chaque Classe devient une table** : Les attributs de la classe deviennent les colonnes de la table.
- **Pour les associations (1...*) :** La clé primaire de la classe côté 1 est ajoutée comme clé étrangère (marquée par un #) dans la table côté *.
- **Pour les associations (1...1) :** Chaque table garde sa propre clé primaire pour préserver son identité, et on lie les tables via une clé étrangère (#) qui pointe vers la table principale.
- **Correction syntaxique :** Les espaces dans les noms des tables sont supprimés (ex : Control Technique devient ControlTechnique) pour respecter les normes SQ

2.4. Modèle Relationnel :

Le Modèle relationnel traduit les liens entre les tables. Les clés primaires sont soulignées et les clés étrangères sont marquées par un #.

Utilisateur (**id_util,** nom, email, motDePasse, role)
Carnet (**id _carnet**, marque, modele, immatriculation, dateCreation, etat, dateDebEv, dateFinEv, titreEv, descEv, **#id_util**)
Révision (id_rev, date, type, description, cout, **#id_carnet**)
Assurance (**id_assur**, dateDebut, dateFin, compagnie, numeroPolice, **#id_carnet**)
Control Technique (**id_ct**, datePassage, resultat, prochaineEcheance, **#id_carnet**)
Notification (**id_notif**, message, date, estLue, **#id_carnet**)
Gestion des Services (Spécialisation) :
Service (**id_serv**, nom, telephone, adresse, type)
Gendarmerie (#id_serv, brigade, secteur)
Station-service (#id_serv, carburantsDispo)
Dépannage (#id_serv, zoneIntervention)
Pompiers (#id_serv)

Points clés de la structure :

Lien Utilisateur -> Carnet : La clé #id_util dans la table Carnet permet de savoir à qui appartient le véhicule.
Lien Carnet -> Maintenance : Toutes les interventions (Révisions, Assurances, ControleTechnique, Notification) pointent vers #id_carnet.
Liaison des Services (Associations) : Chaque table spécifique (Gendarmerie, Pompiers, Station-Service, Dépannage) possède sa propre clé primaire et contient une clé étrangère #id_serv qui pointe vers la table Service. Cela permet de lier chaque entité spécifique aux informations communes (nom, tel, adresse) stockées dans la table Service, tout en respectant les multiplicités du diagramme

2.5. Conclusion :

Ce chapitre a porté sur la phase de conception, incluant les diagrammes de séquence et de classes, ainsi que la construction d'un modèle relationnel normalisé et optimisé pour la gestion des services. Après cette étape de modélisation, nous passons à la phase d'implémentation et de développement pratique de l'application, qui sera présentée en détail dans le chapitre suivant

Chapitre III : Implémentation

3.1 Introduction

Après l'étude conceptuelle présentée dans le chapitre précédent, nous entamons la phase d'implémentation de notre application. Nous y présenterons l'environnement de développement, les langages utilisés, ainsi que les différentes interfaces de l'application

3.2 Environnement de développement

Cette section présente les outils logiciels et les langages de programmation adoptés pour l'implémentation de l'application.

3.2.2 Environnent logiciel

Android Studio

Android Studio est l'environnement de développement intégré (IDE) officiel dédié à la conception d'applications Android. Il offre un ensemble complet d'outils facilitant le développement, le test et le déploiement des applications mobiles. Cet environnement permet notamment l'édition du code source en Java ou Kotlin, ainsi que la gestion des fichiers de configuration XML. Il intègre également des fonctionnalités avancées, telles que la prise en charge des applications multilingues et des outils de prévisualisation des interfaces utilisateur en temps réel [10].



Android SDK

Android Software Développement Kit (SDK) désigne un ensemble d'outils et de ressources mis à disposition des développeurs pour concevoir des applications destinées à la plateforme Android. Il comprend généralement un compilateur assurant la traduction du code source en langage machine, un éditeur de liens permettant l'assemblage des différents modules en un programme exécutable, ainsi que des bibliothèques facilitant le développement.



De nos jours, le SDK constitue un élément essentiel dans le développement des applications, notamment mobiles [11].

SQLite

SQLite est une bibliothèque logicielle implémentant un système de gestion de base de données relationnelle, léger, autonome et sans serveur. Elle ne nécessite aucune configuration préalable et prend en charge les transactions, ce qui la rend particulièrement adaptée aux applications mobiles [12].



Java

Java est un langage de programmation orienté objet et modulaire, inspiré du langage ADA. Sa syntaxe est largement inspirée du langage C, ce qui facilite son apprentissage et son utilisation pour le développement d'applications multiplateformes [13].

XML

XML est un métalangage utilisé pour décrire des documents et structurer des données, reconnu pour sa flexibilité et son efficacité [14].

JDK

Java Développement Kit (JDK) désigne un ensemble de bibliothèques fondamentales du langage Java, accompagné d'outils permettant de compiler le code source et de le transformer en bytecode exécutable par la machine virtuelle Java (JVM) [15].

3.3 Présentation de notre logo

La figure suivante représente le logo de notre application :

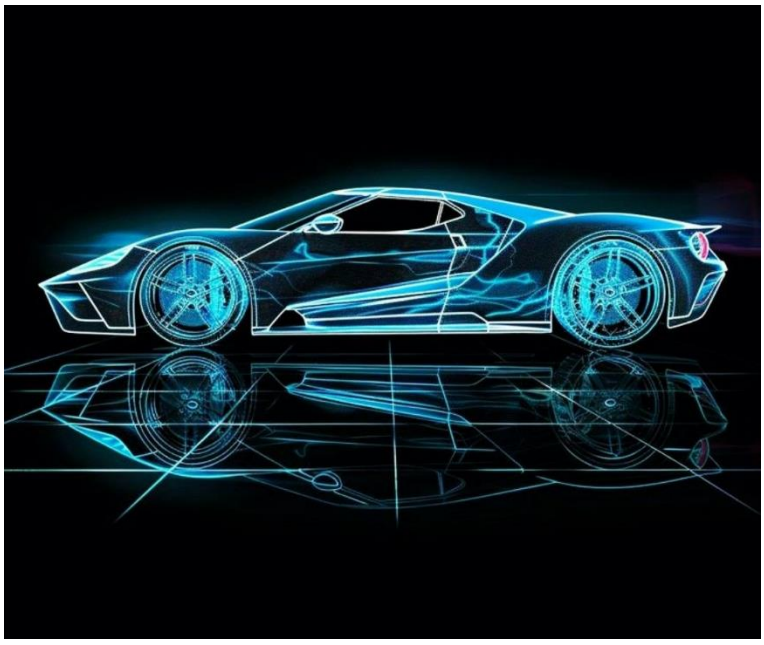


Figure 10: Logo de notre application

3.4 Présentation des interfaces graphiques

Dans cette partie nous allons présenter quelques interfaces graphiques à travers des captures d'écran pour montrer les différentes fonctionnalités de notre système.

1.4.1 Authentification et inscription

Les figures suivantes représentent les interfaces d'authentification, d'inscription et de récupération du mot de passe permettant à l'utilisateur de se connecter au système, de créer un nouveau compte ou de réinitialiser son mot de passe en cas d'oubli.

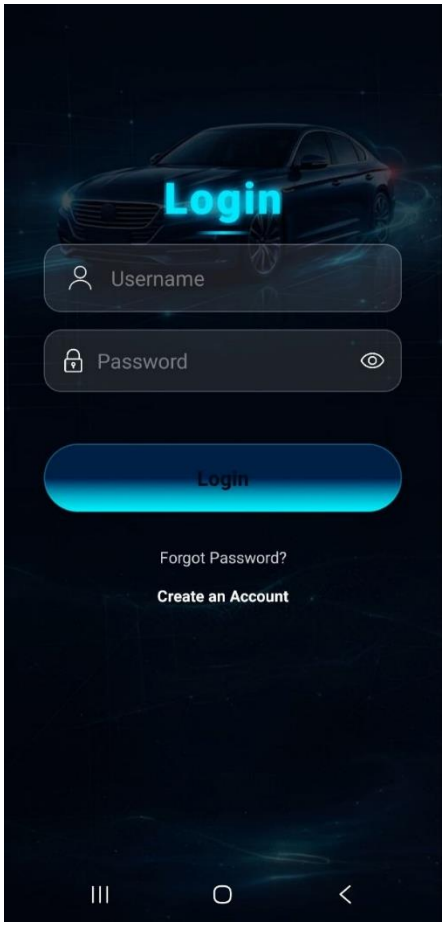


Figure 11: d'authentification

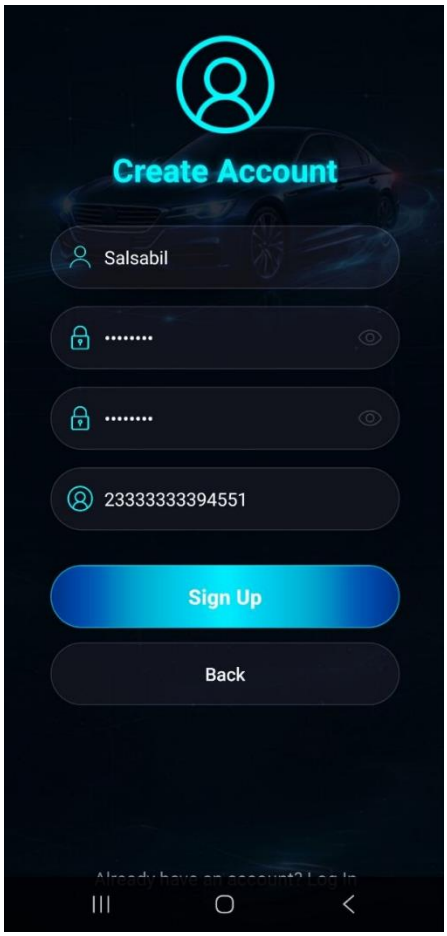


Figure 12:Interface d'inscription

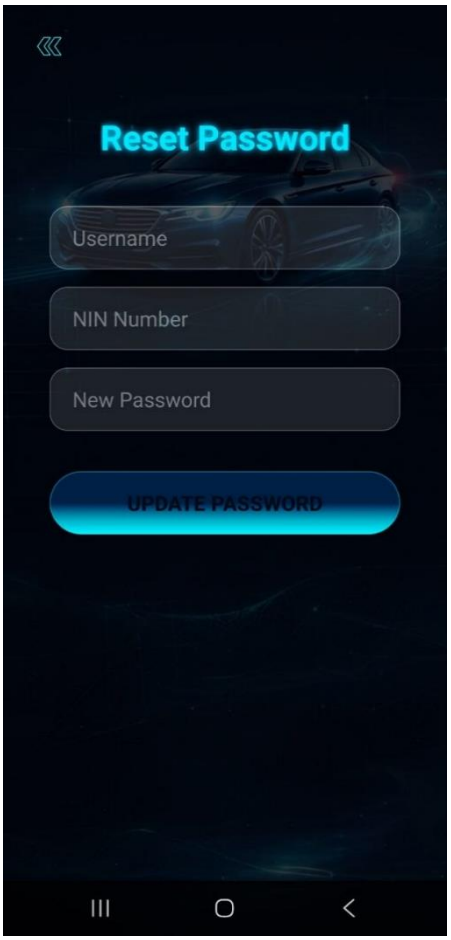


Figure 13:récupération du mot de passe

.

3.4.2 Accueil

La figure suivante représente la page d'accueil de notre application qui permet à l'utilisateur d'accéder aux différents espaces :

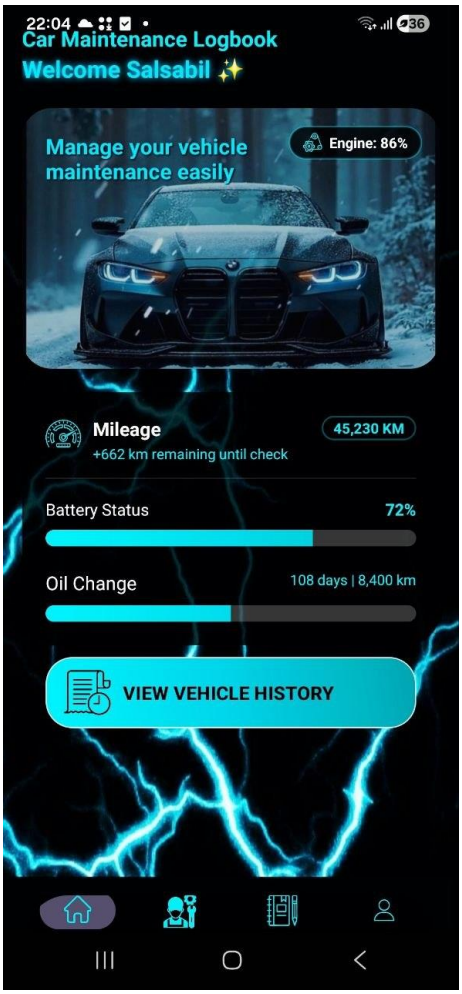


Figure 14: La page d'accueil

3.4.3 Historique :

La figure suivante représente la page d'historique de notre application qui permet à l'utilisateur de consulter l'archive de toutes les opérations de maintenance et de réparation effectuées sur le véhicule.



Figure 15: Interface de l'historique des entretiens

3.4.4 Ajout et Modification de véhicule :

La figure suivante représente la page d'ajout et de modification qui permet à l'utilisateur d'enregistrer un nouveau véhicule ou de mettre à jour ses informations existantes.

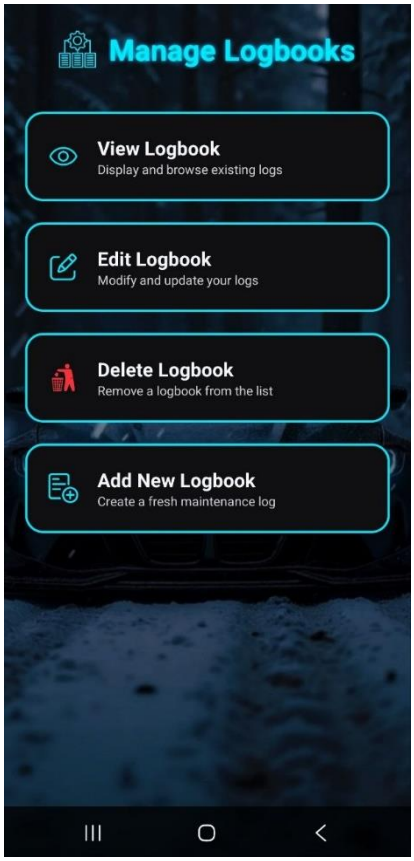


Figure 16: Interface du carnet

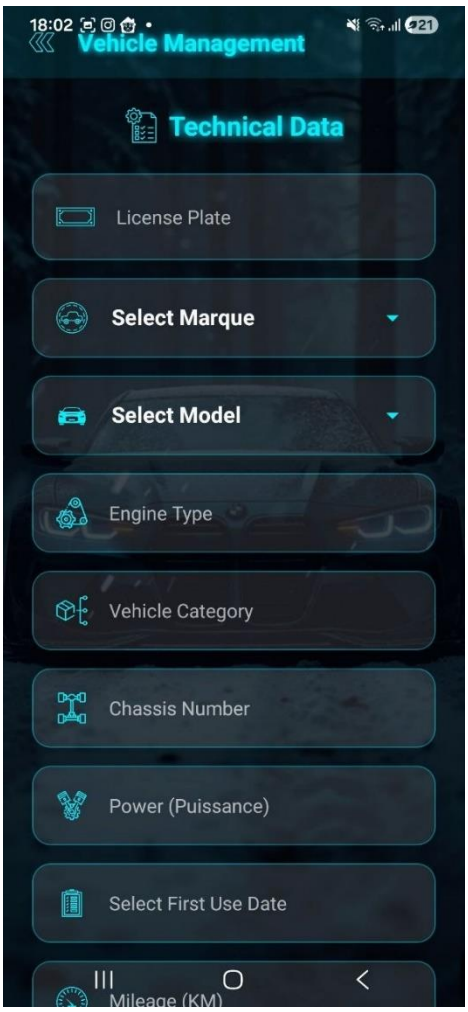


Figure 17: Interface d'ajout de véhicule



Figure 18:modification

3.4.6 Liste des véhicules et Consultation

Les figures suivantes représentent les interfaces de la liste des véhicules enregistrés et de la consultation de leurs détails de maintenance



Figure 19:Interface de la liste des véhicules



Figure 20:Interface de consultation des détails du véhicule

3.4.7 Localisation et Partage

Les figures suivantes représentent les interfaces de recherche de services sur la carte et de partage de la position actuelle :

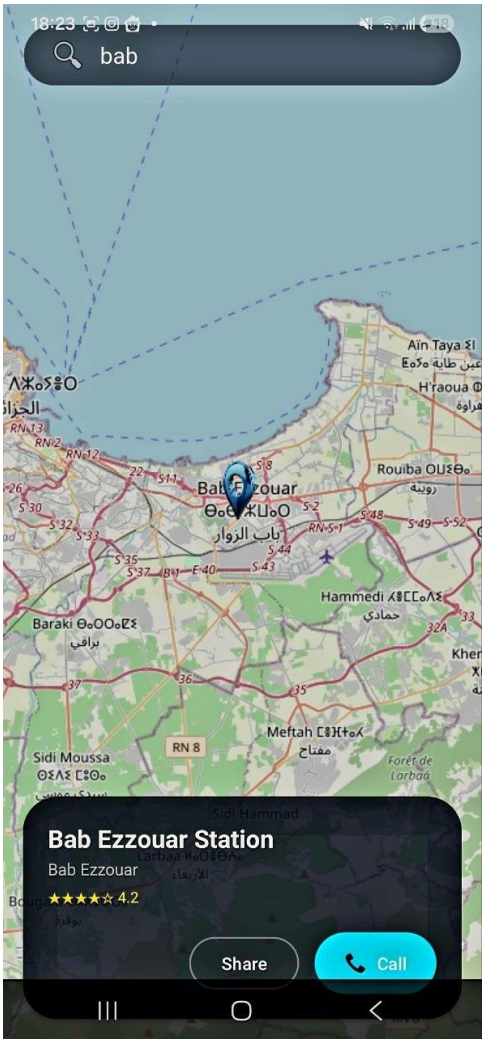


Figure 21:Interface de localisation

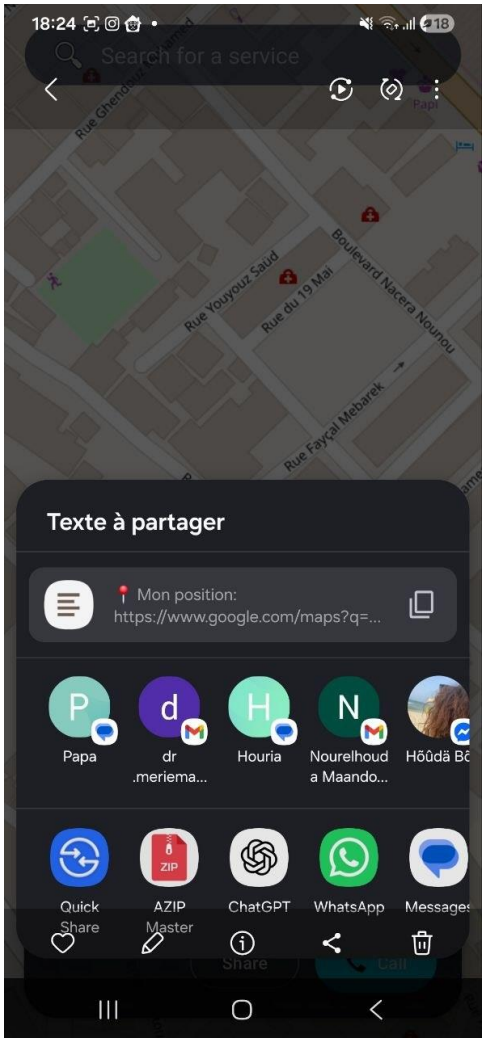


Figure 22: Interface de partage de la position

Conclusion générale

En conclusion de ce travail, il est important de souligner que ce projet s'inscrit dans le cadre de la transformation numérique que connaît le monde actuel, où les solutions technologiques deviennent indispensables pour améliorer la qualité de vie et faciliter les activités quotidiennes. La réalisation de ce projet nous a permis d'acquérir une expérience pratique à travers les différentes étapes du développement d'une application mobile, allant de l'analyse des besoins jusqu'à la phase de conception, puis à l'implémentation.

L'analyse des besoins nous a permis de définir clairement les exigences des utilisateurs et de construire une vision globale du système. La phase de conception a contribué à établir une structure logique et organisée du système à l'aide des diagrammes UML, facilitant ainsi le développement. Quant à la phase d'implémentation, elle a permis de concrétiser le projet en une application fonctionnelle offrant aux utilisateurs un moyen simple et efficace de suivre l'entretien de leurs véhicules grâce à des interfaces intuitives et une expérience utilisateur fluide.

L'intérêt de ce travail réside dans le fait qu'il propose une solution numérique permettant de réduire les oublis liés à l'entretien des véhicules, en organisant les informations et en rappelant les échéances importantes, ce qui contribue à préserver les performances du véhicule et à réduire les coûts liés aux pannes imprévues.

Malgré les résultats satisfaisants obtenus, ce projet reste ouvert à plusieurs perspectives d'amélioration. En effet, il pourrait être enrichi par l'intégration de fonctionnalités intelligentes telles que des systèmes de prédiction des pannes ou la connexion à des services externes afin de fournir des informations plus complètes et précises, augmentant ainsi la performance et la valeur ajoutée de l'application.

Webographie :

1. Service-Public.fr, "Contrôle technique d'un véhicule particulier", https://www.service-public.fr/particuliers/vosdroits/F2851 (consulté le 12/05/2026)
2. TOPBAS, "My Garage - Car Assistant", https://apps.apple.com/fr/app/my-garage-car-assistant/id6757397493 (consulté le 12/05/2026)
3. Jo App (Jordanie), "CarLog – Car Maintenance Tracker", https://play.google.com/store/apps/details?id=com.joapp.carlog (consulté le 12/05/2026)
4. Drivvo, "Drivvo – Gestion des véhicules : Carburant, Maintenance et Rappels", https://www.drivvo.com (consulté le 12/05/2026)
5. VehicleTrack, "VehicleTrack-MonitorYourFleet : Realtime GPS Tracking", https://apps.apple.com/fr/app/vehicletrack-monitoryourfleet/id1158866599 (consulté le 12/05/2026)
6. Object Management Group (OMG), "Unified Modeling Language (UML) Specification", Version 2.5.1, https://www.omg.org/spec/UML/ (consulté le 12/05/2026)
7. Object Management Group (OMG), "UML Use Case Diagrams Overview", https://www.omg.org (consulté le 12/05/2026)
8. Sommerville, I., Software Engineering, 10th Edition, Pearson Education, https://www.pearson.com/en-us/subject-catalog/p/software-engineering/9780133943030.html (consulté le 12/05/2026)
9. Larman, C., Applying UML and Patterns, 3rd Edition, Prentice Hall, Pearson Education, https://www.pearson.com/en-us/subject-catalog/p/applying-uml-and-patterns/P200000006xxx (consulté le 12/05/2026)
10.
11. Google Android Developers, "Android Studio : The Official IDE for Android", https://developer.android.com/studio (consulté le 12/05/2026)
12. Google Developers, "Android SDK Tools & Libraries Overview", https://developer.android.com/tools (consulté le 12/05/2026)
13. SQLite Consortium, https://www.sqlite.org (Consulté le 12/05/2026)
14. Oracle Corporation, "Java Platform, Standard Edition", https://www.oracle.com/java (Consulté le 12/05/2026)
15. World Wide Web Consortium (W3C), https://www.w3.org (consulté le 12/05/2026)
16. Oracle Corporation, "Java SE Documentation", https://docs.oracle.com/en/java/javase/ (consulté le 12/05/2026)