



PRESIDENCE DE LA REPUBLIQUE

MINISTERE DE LA PLANIFICATION
DU DEVELOPPEMENT ET DE LA
COOPERATION



Institut Africain d'Informatique
Représentation du TOGO
(IAI-TOGO)
07 BP 12456 Lome 07
Tel : (+228) 22 20 47 00
Courriel : iaitogo@gmail.com
Site-Web : new.iai-togo.tg/officiel

COMPAGNIE ÉNERGIE ÉLECTRIQUE
DU TOGO
426, AVENUE MAMA FOUSSENI
B.P. 42 LOME
TEL : 22 21 27 43 / 44
FAX : 22 21 64 98

MEMOIRE DE FIN DE FORMATION POUR L'OBTENTION DU DIPLOME D'INGENIEUR DES TRAVAUX INFORMATIQUES

OPTION : Génie Logiciel et Systèmes d'Information

DEVELOPPEMENT ET MISE EN ŒUVRE D'UNE APPLICATION POUR LA GESTION DU PARC AUTOMOBILE DE LA CEET

Du 27 Mai 2024 au 17 Août 2024

Rédigé et soutenu par :

FLINDJO Namtante

Etudiant en troisième année

Année académique : 2023 – 2024.

Superviseur :

Monsieur Kossivi GAVI

Professeur à l'IAI-Togo

Maitre de stage :

Monsieur Simon EDOH

Chef Service Etude et
Veille Technologique de
la CEET

Dédicaces

- ❖ À mes parents, pour leur amour inconditionnel et leur soutien constant. Vous avez toujours cru en moi et m'avez donné la force de persévérer.
- ❖ À mes tuteurs, Monsieur et Madame LOADJA pour leurs conseils éclairés et leur encadrement bienveillant. Votre contribution a été essentielle à la réussite de ce projet.

Ce travail est le fruit de nombreux efforts collectifs et de précieux soutiens. À chacun de vous, je dédie cette réussite avec une profonde gratitude et reconnaissance.



Remerciements

Ô Dieu Tout-Puissant, pour toutes les bénédictions accordées tout au long de mon parcours, je te rends grâce. Depuis le commencement jusqu'à aujourd'hui, ta présence constante m'a guidé à travers les épreuves et les joies, m'illuminant de ton amour inconditionnel. Merci infiniment.

- ❖ Je souhaite exprimer ma profonde gratitude à l'Institut Africain d'Informatique (IAI) ainsi qu'à tout son personnel pour leur engagement envers l'excellence académique et leur dévouement à la formation de futurs experts en informatique, qui ont été essentiels à mon développement.
- ❖ Un merci particulier à Monsieur AGBETI, Représentant Résident de l'IAI, pour son engagement envers notre réussite.
- ❖ Je tiens également à exprimer mes sincères remerciements à la CEET pour l'accueil et l'expérience enrichissante que j'y ai vécue, en particulier à la Direction de la Cellule de Transformation Digitale.
- ❖ À M. Daniel YENTCHABRE, pour son soutien indéfectible tout au long de mon stage. Votre guidance et vos encouragements ont été d'une valeur inestimable.
- ❖ Je suis profondément reconnaissant envers Monsieur EDOH Simon, mon maître de stage, pour son encadrement rigoureux, ses précieux conseils et son expertise, ainsi qu'à Monsieur GAVI Kossivi, mon superviseur, pour son accompagnement attentif, ses orientations pertinentes et son soutien indéfectible.
- ❖ Enfin, je remercie de tout cœur ma famille et mes amis, dont les sacrifices, l'amour et la foi inébranlable en moi ont été ma force motrice tout au long de ce parcours.

Aujourd'hui, je suis fier d'être diplômé de l'IAI, mais ce n'est que le début d'un nouveau chapitre. Je m'engage à porter haut les valeurs de cette institution d'excellence, à mettre mes compétences au service de la société et à contribuer positivement au développement de l'Afrique.

Application de gestion du parc automobile de la CEET**Sommaire**

| | |
|--|-------------|
| Résumé..... | iv |
| Glossaire | v |
| Liste des figures | vi |
| Liste des tableaux | viii |
| Liste des participants..... | ix |
| Introduction..... | 1 |
| 1. Partie I : Cahier des charges | 2 |
| 1.1. PRÉSENTATIONS | 3 |
| 1.2. THÈME DU STAGE | 7 |
| 1.3. ETUDE DE L'EXISTANT..... | 9 |
| 1.4. CRITIQUE DE L'EXISTANT | 11 |
| 1.5. PROPOSITIONS ET CHOIX DE SOLUTIONS..... | 12 |
| 1.6. PLANNING PRÉVISIONNEL DE RÉALISATION | 15 |
| 2. Partie II : Analyse et Conception..... | 18 |
| 2.1. CHOIX DE LA METHODE D'ANALYSE ET JUSTIFICATION | 19 |
| 2.2. CHOIX DE L'OUTIL DE MODÉLISATION ET JUSTIFICATION | 21 |
| 2.3. ETUDE DÉTAILLÉE DE LA SOLUTION | 21 |
| 3. Partie III : réalisation et Mise en œuvre | 41 |
| 3.1. MATÉRIELS ET LOGICIELS UTILISÉS..... | 42 |
| 3.2. ARCHITECTURE MATÉRIELLE ET LOGICIELLE DE L'APPLICATION | 46 |
| 3.3. SÉCURITÉ DE L'APPLICATION | 48 |
| 3.4. MISE EN PLACE DE LA BASE DE DONNÉES | 49 |
| 4. Partie IV : Guide d'exploitation | 58 |
| 4.1. CONFIGURATION LOGICIELLE ET MATÉRIELLE | 59 |
| 4.2. DÉPLOIEMENT ET SUIVI..... | 60 |
| 4.3. MAINTENANCE | 62 |
| 5. Partie V : Guide d'utilisation..... | 64 |
| 5.1. DESCRIPTION TEXTUELLE DU LOGICIEL | 65 |
| 5.2. PLAN DE NAVIGATION | 66 |
| 5.3. PRESENTATION DES DIFFERENTES INTERFACES DE L'APPLICATION..... | 69 |
| 5.4. PRESENTATION DES ETATS | 74 |
| Conclusion | 77 |
| Bibliographie indicative | I |
| Webographie indicative | II |

Résumé

Ce mémoire porte sur le développement et la mise en œuvre d'une application pour la gestion du parc automobile de la Compagnie Énergie Électrique du Togo (CEET). L'objectif principal de ce projet est d'améliorer la gestion logistique de la société en optimisant le suivi, l'entretien et l'utilisation des véhicules de service.

L'application développée permet une gestion centralisée et en temps réel des caractéristiques des véhicules, de leur inventaire, de la planification des entretiens, ainsi que de la gestion des commandes et des acquisitions. En intégrant des fonctionnalités telles que la gestion des horaires, le suivi des kilomètres parcourus, et la surveillance des états des véhicules, l'application vise à réduire les coûts opérationnels, à minimiser les risques d'accidents, et à prolonger la durée de vie des véhicules.

Le projet a été réalisé en suivant une méthodologie rigoureuse, incluant la phase de planification, l'analyse des besoins, la conception du système, l'implémentation et les tests. Les résultats obtenus démontrent une amélioration significative de l'efficacité et de la fiabilité des processus internes liés à la gestion du parc automobile.

En conclusion, l'application développée constitue un outil stratégique pour la CEET, offrant une solution complète et intégrée pour la gestion de ses véhicules, et contribuant ainsi à la performance globale de la société.

Glossaire

| Expressions / Acronymes | Significations |
|--------------------------------|---|
| IAI | Institut Africain d'Informatique |
| CEET | Compagnie Énergie Électrique du Togo |
| GLSI | Génie Logiciel et Système d'Information |
| ASR | Administration des Systèmes et Réseaux |
| MTWI | Multimédia Technologie Web et Infographie |
| CENETI | Centre National d'Etude et de Traitements Informatiques |
| ORM | Object-Relational Mapping |

Liste des figures

| | |
|--|----|
| <i>Figure 1 : Carte de la géolocalisation de l'IAI-Togo</i> | 4 |
| <i>Figure 2 : Organigramme de la CEET</i> | 5 |
| <i>Figure 3 : Organigramme de la Cellule de Transformation Digitale</i> | 5 |
| <i>Figure 4 : Carte de la géolocalisation de la CEET</i> | 6 |
| <i>Figure 5 : Logo de la Société Fleet Complete</i> | 12 |
| <i>Figure 6 : Diagramme de Gantt des activités</i> | 17 |
| <i>Figure 7 : Logo UML</i> | 20 |
| <i>Figure 8 : Processus 2TUP</i> | 20 |
| <i>Figure 9 : Logo draw.io</i> | 21 |
| <i>Figure 10 : Cas d'utilisation MECANICIEN</i> | 24 |
| <i>Figure 11 : Cas d'utilisation EXPLOITANT</i> | 24 |
| <i>Figure 12 : Cas d'utilisation CHEF SERVICE</i> | 25 |
| <i>Figure 13 : Cas d'utilisation MAGASINIER</i> | 25 |
| <i>Figure 14 : Cas d'utilisation CONTROLEUR</i> | 26 |
| <i>Figure 15 : Cas d'utilisation POMPISTE</i> | 26 |
| <i>Figure 16 : Cas d'utilisation GENERAL</i> | 27 |
| <i>Figure 17 : Diagramme de séquence : EXPLOITATION</i> | 35 |
| <i>Figure 18 : Diagramme de séquence : MAINTENANCE</i> | 36 |
| <i>Figure 19 : Diagramme de séquence : APPROVISIONNEMENT</i> | 36 |
| <i>Figure 20 : Diagramme d'activités <<Connexion>></i> | 37 |
| <i>Figure 21 : Diagramme d'activités <<Sortie>></i> | 37 |
| <i>Figure 22 : Diagramme d'activités <<Approvisionnement>></i> | 38 |
| <i>Figure 23 : Diagramme d'activités <<Maintenance>></i> | 38 |
| <i>Figure 24 : Diagramme de classe</i> | 40 |
| <i>Figure 25 : Logo Visual Studio Code</i> | 43 |
| <i>Figure 26 : Logo XAMPP</i> | 43 |
| <i>Figure 27 : Logo MySQL</i> | 43 |
| <i>Figure 28 : Logo PHP</i> | 44 |
| <i>Figure 29 : Logo Laravel</i> | 44 |
| <i>Figure 30 : Logo HTML & CSS</i> | 44 |
| <i>Figure 31 : Logo JavaScript</i> | 45 |
| <i>Figure 32 : Logo JQuery</i> | 45 |
| <i>Figure 33 : Logo GitHub</i> | 45 |
| <i>Figure 34 : Logo Apache</i> | 46 |
| <i>Figure 35 : Architecture matérielle 3-tiers</i> | 47 |
| <i>Figure 36 : Architecture logicielle MVC</i> | 48 |

Application de gestion du parc automobile de la CEET

| | |
|---|----|
| <i>Figure 37 : Migration <<vehicules>></i> | 50 |
| <i>Figure 38 : Migration <<affectations>></i> | 51 |
| <i>Figure 39 : Migration <<fiche_sorties>></i> | 52 |
| <i>Figure 40 : Migration <<demandes>></i> | 53 |
| <i>Figure 41 : Migration <<fiche_maintenances>></i> | 54 |
| <i>Figure 42 : Migration <<approvisionnements>></i> | 55 |
| <i>Figure 43 : Migration <<users>></i> | 56 |
| <i>Figure 44 : Migration <<pieces>></i> | 57 |
| <i>Figure 45 : Page de création d'une nouvelle base de données</i> | 61 |
| <i>Figure 46 : Tableau de bord administrateur</i> | 66 |
| <i>Figure 47 : Interface de gestion du parc pour l'administrateur et chef service</i> | 66 |
| <i>Figure 48 : Plan de navigation administrateur et chef service</i> | 67 |
| <i>Figure 49 : Interface et plan de navigation exploitation</i> | 68 |
| <i>Figure 50 : Interface et plan de navigation maintenance</i> | 68 |
| <i>Figure 51 : Interface et plan de navigation consommation</i> | 69 |
| <i>Figure 52 : Page d'authentification</i> | 70 |
| <i>Figure 53 : Interface d'ajout d'un véhicule</i> | 71 |
| <i>Figure 54 : Interface de redirection pour affectation d'un nouveau véhicule</i> | 71 |
| <i>Figure 55 : Interface d'affectation d'un véhicule</i> | 72 |
| <i>Figure 56 : Interface d'ajout d'une demande de sortie</i> | 73 |
| <i>Figure 57 : Interface de création d'une fiche de sortie</i> | 74 |
| <i>Figure 58 : Interface de terminaison d'une fiche de sortie</i> | 74 |
| <i>Figure 59 : Liste des affectations de véhicules</i> | 74 |
| <i>Figure 60 : Liste des demandes de sortie</i> | 75 |
| <i>Figure 61 : Liste des sorties en cours</i> | 76 |
| <i>Figure 62 : Statiques et état actuel du parc</i> | 76 |

Liste des tableaux

| | |
|--|----|
| <i>Tableau 1 : Estimation des coûts des propositions de solution</i> | 14 |
| <i>Tableau 2 : Tableau des différentes activités et les périodes d'exécution</i> | 16 |
| <i>Tableau 3 : Tableau récapitulatif des cas d'utilisations</i> | 22 |
| <i>Tableau 4 : Caractéristiques du matériel utilisé</i> | 42 |
| <i>Tableau 5 : Configuration logicielle de notre application</i> | 59 |
| <i>Tableau 6 : Configurations matérielles pour notre application</i> | 60 |
| <i>Tableau 7 : Actions à mener en cas d'erreurs</i> | 63 |

Liste des participants

Le tableau ci-dessous présente la liste des personnes ayant contribué à la réalisation de ce projet.

| Nom et Prénoms | Fonction | Rôle |
|----------------------------|--|-----------------|
| M. FLINDJO Namtante | Etudiant en troisième année de Génie Logiciel et Systèmes d'Information à l'Institut Africain d'Informatique | Réalisateur |
| M. Simon EDOH | Chef service Etude et Veille technologique de la CEET | Maitre de stage |
| M. Kossivi GAVI | Enseignant à l'Institut Africain d'Informatique | Superviseur |

Introduction

Dans un contexte où l'efficacité et la réactivité constituent des critères essentiels pour la compétitivité des entreprises, la gestion optimale des ressources logistiques s'impose comme un enjeu majeur. La Compagnie Énergie Électrique du Togo (CEET), en tant qu'acteur clé du secteur énergétique au Togo, dépend largement de la performance et de la disponibilité de son parc automobile pour assurer ses opérations quotidiennes. Cependant, la gestion traditionnelle de ce parc, souvent caractérisée par une approche manuelle et décentralisée, présente de nombreuses limites, notamment en termes de suivi, de maintenance, et de gestion des coûts.

C'est dans ce contexte que s'inscrit ce projet de mémoire, intitulé "**Développement et mise en œuvre d'une application pour la gestion du parc automobile de la CEET**", qui m'a été confié lors de mon stage à la CEET. L'objectif principal de ce projet est de concevoir et de déployer une solution technologique qui permettra d'optimiser la gestion de la flotte de véhicules de la CEET. Cette application vise à centraliser les informations, automatiser les processus de maintenance et de suivi, gérer l'exploitation des véhicules, et fournir des outils analytiques pour une meilleure gestion des coûts.

Afin de présenter de manière détaillée et optimale les différentes étapes et résultats obtenu lors de la réalisation de ce projet, ce document sera structuré comme suit :

- ⌚ La première partie qui présentera le cadre de travail et les objectifs et résultats attendus pour ce projet ainsi que les différentes études portées sur ce dernier.
- ⌚ La seconde partie présentera les outils et les résultats d'analyse et de conception.
- ⌚ La troisième partie portera sur les méthodes de mise en œuvre et l'architecture utilisée.
- ⌚ Les quatrième et cinquième partie seront portée respectivement sur la rédaction du guide d'exploitation et du guide d'utilisation.
- ⌚ Et enfin une conclusion qui fera le point sur toute notre étude.

1. Partie I : Cahier des charges

1.1.PRÉSENTATIONS

1.1.1. Brève présentation de l’Institut Africain d’Informatique (IAI-Togo)

L’Institut Africain d’Informatique (IAI-Togo) est un établissement d’enseignement supérieur spécialisé en informatique et technologies de l’information. Il fait partie du réseau IAI, un groupe d’établissements publics d’enseignement supérieur africains dédiés à la formation et à la recherche en informatique et en technologies de l’information.

Fondé en 2002, l’IAI Togo se situe à Lomé, au Togo. L’école propose une variété de programmes de premier cycle, de deuxième cycle et de formation professionnelle. Son corps professoral est composé de professionnels expérimentés issus du milieu universitaire et de l’industrie.

L’IAI Togo est un acteur de premier plan dans le domaine de la formation en informatique et technologies de l’information au Togo. Ses diplômés sont très recherchés par les employeurs des secteurs public et privé.

IAI-Togo propose des formations en informatique appliquée, en génie logiciel et en systèmes d’information (GLSI), en réseaux et télécommunications (ASR), et en multimédia (MTWI). Ces programmes visent à préparer les étudiants à répondre aux besoins croissants en expertise informatique sur le continent africain. L’institut est également impliqué dans divers projets de recherche et de développement en collaboration avec des partenaires nationaux et internationaux, contribuant ainsi à l’innovation technologique en Afrique.

L’IAI-TOGO est situé dans l’enceinte de l’immeuble CENETI sis au quartier administratif, 15 Rue de la Kozah, 07 BP 12456 Lomé 07, TOGO.

Application de gestion du parc automobile de la CEET



Figure 1 : Carte de la géolocalisation de l'IAI-Togo

1.1.2. Présentation du cadre de stage

La Compagnie Énergie Électrique du Togo (CEET), fondée en 1968, est l'entité nationale en charge de la production, de la distribution et de la commercialisation de l'électricité au Togo. Basée à Lomé, la CEET est une entreprise publique placée sous la supervision du ministère de l'Énergie et des Mines. Avec un effectif de plus de 1 500 employés et un réseau de plus de 50 agences à travers le pays, la CEET dessert plus de 1,5 million de clients. La mission de la CEET est triple :

- ✖️ Fournir une électricité fiable et abordable à l'ensemble des citoyens togolais.
- ✖️ Promouvoir le développement du secteur électrique du pays.
- ✖️ Contribuer au développement économique et social du Togo.

Afin de parvenir à la réalisation de sa mission, la CEET a mis en place plusieurs solutions tel que la création de la Direction de la Cellule de la Transformation Digital. Le but est de développer et de mettre en œuvre des solutions numériques pour améliorer l'efficacité de l'entreprise et le service client.

Organigramme de la Compagnie Énergie Électrique du Togo

Application de gestion du parc automobile de la CEET

L'organigramme de la CEET reflète une structure hiérarchique et fonctionnelle conçue pour favoriser une communication fluide, une prise de décision rapide, et une coordination efficace entre ses différents départements.

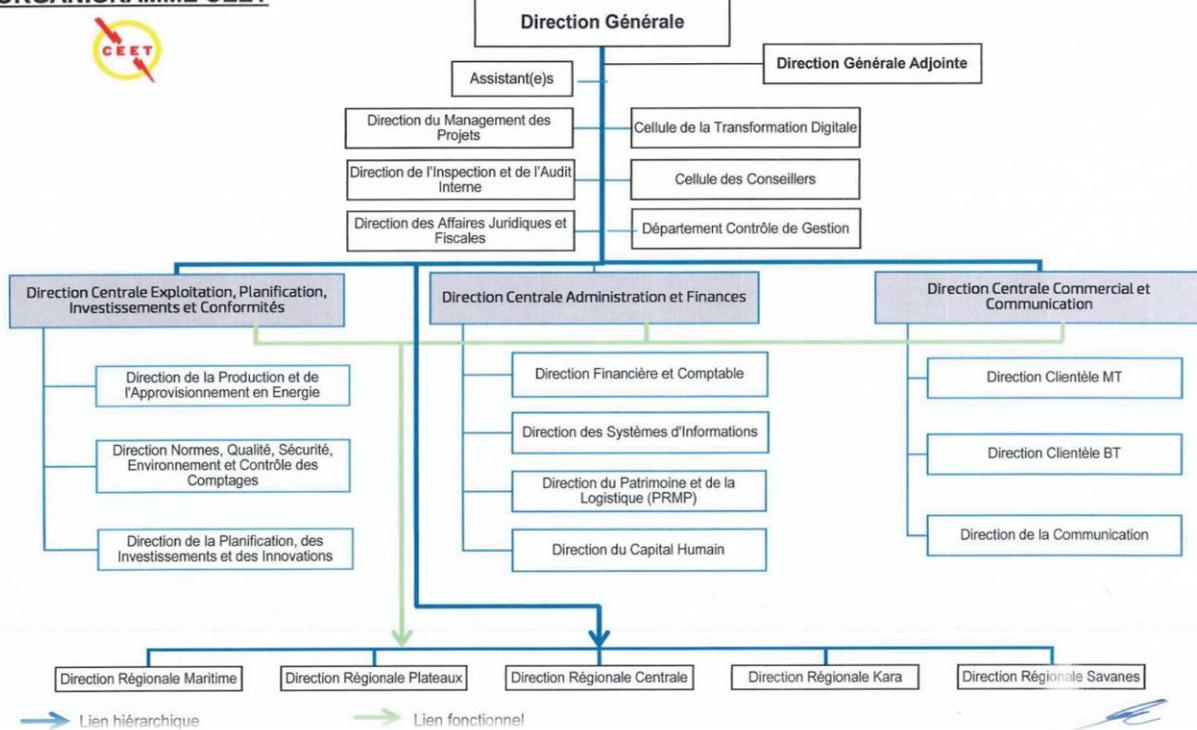
ORGANIGRAMME CEET

Figure 2 : Organigramme de la CEET

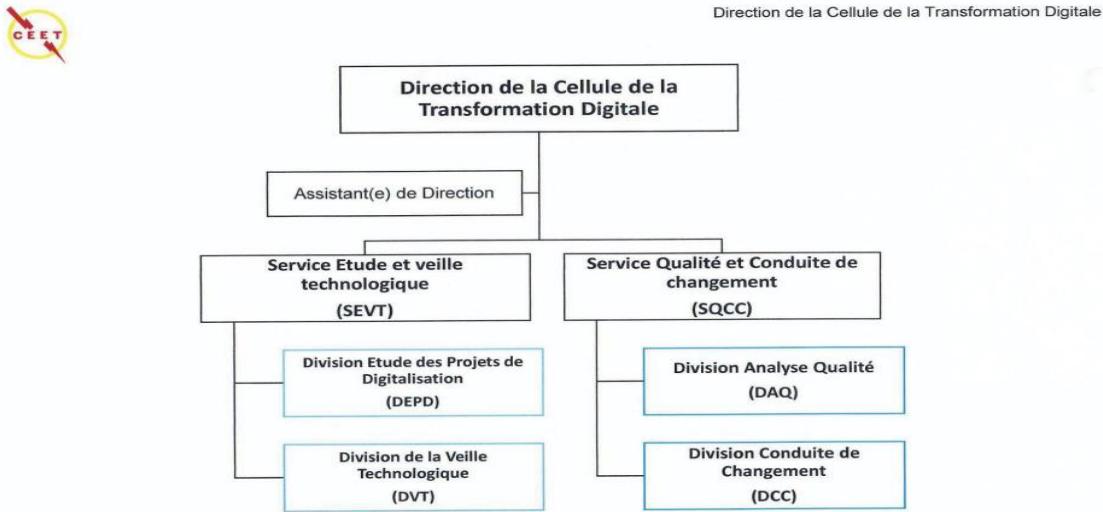


Figure 3 : Organigramme de la Cellule de Transformation Digitale

Plan de géolocalisation de la Compagnie Énergie Électrique du Togo

Le siège social de la CEET est situé à Lomé. Voici les coordonnées et les contacts pour toute correspondance ou demande d'information :

Adresse : Avenue de la Nouvelle Marche, Lomé, Togo

Téléphone : +228 22 21 22 22

Courriel : contact@ceet.tg

Site Web : www.ceet.tg

La CEET dispose de plus de 50 agences réparties stratégiquement à travers le pays. Ces agences sont implantées dans les principales villes et régions pour assurer une couverture optimale et un accès facilité aux services électriques pour tous les citoyens togolais



Figure 4 : Carte de la géolocalisation de la CEET

1.2. THÈME DU STAGE

1.2.1. Présentation du sujet

La CEET est l'entité nationale en charge de la production, de la distribution et de la commercialisation de l'électricité au Togo. De ce fait la mobilité de ses agents et leurs présences sur le terrain devient un facteur très important dans son principal objectif, qui est de fournir sa population en électricité et de s'assurer de la disponibilité et du fonctionnement des installations publics (lampadaire, panneaux publicitaires, feux tricolores ...).

Pour parer à ses défis, la CEET se doit d'avoir un parc automobile dynamique et hautement équipé, mais aussi bien organisé et bien géré. D'où la nécessité de mettre en place un système informatique de gestion de son parc automobile.

Le projet « **Développement et mise en œuvre d'une application de gestion du parc automobile de la CEET** » se présente comme la solution adéquate à ce problème. Car elle aura pour but d'optimiser la gestion du parc afin de la rendre plus efficace pour les besoins de la Compagnie.

1.2.2. Problématique du sujet

La Compagnie Énergie Électrique du Togo (CEET) joue un rôle crucial dans la fourniture d'électricité à l'ensemble du territoire togolais. Pour accomplir cette mission, la CEET s'appuie sur une flotte de véhicules utilisée pour diverses opérations, telles que la maintenance des infrastructures, la réponse aux urgences, et le transport de personnel et de matériel. La gestion du parc automobile de la CEET est donc essentielle pour assurer la mobilité et le dynamisme de ses opérations.

Cependant, la gestion actuelle du parc automobile, présente certains manquements dans son fonctionnement. Ces défis se posent sous plusieurs angles :

1. Comment optimiser l'utilisation des véhicules pour assurer leur disponibilité pour les missions cruciales tout en minimisant les temps d'arrêt ?
2. Comment suivre et contrôler efficacement la consommation de carburant pour éviter les gaspillages et réduire les coûts ?
3. Comment garantir une maintenance régulière et préventive des véhicules pour éviter les pannes et prolonger leur durée de vie ?

Application de gestion du parc automobile de la CEET

4. Comment assurer une gestion efficace des stocks de pièces détachées pour éviter les ruptures et les retards dans les réparations ?

La problématique centrale de ce projet peut ainsi être formulée comme suit : Comment le développement et la mise en œuvre d'une application peuvent-ils améliorer la gestion du parc automobile de la CEET, afin de renforcer l'efficacité opérationnelle de la compagnie ?

1.2.3. Intérêt du sujet**1.2.3.1. Objectifs**

Pour la réalisation de ce projet, nous nous sommes fixés des objectifs afin de toujours garder notre but en perspective. Ces objectifs peuvent être exposés comme suite :

- ✖ **Optimisation des coûts** : Réduire les dépenses liées à l'entretien, à la réparation et de l'exploitation des véhicules
- ✖ **Sécurité** : Garantir la sécurité des utilisateurs grâce à une maintenance régulière et conforme aux prescriptions du constructeur.
- ✖ **Efficacité opérationnelle** : Assurer une disponibilité maximale des véhicules pour les besoins de l'entreprise.
- ✖ **Durabilité** : Permettre une longévité du véhicule.

1.2.3.2. Résultats

Pour ce projet, les résultats attendus sont :

Optimisation des coûts :

- ✖ Réduction significative des dépenses d'entretien et de réparation grâce à une maintenance préventive efficace.
- ✖ Baisse des coûts d'exploitation par une meilleure gestion de la consommation de carburant.

Sécurité :

- ✖ Augmentation de la sécurité des utilisateurs grâce à des inspections régulières et à la conformité des véhicules aux normes de sécurité.

Application de gestion du parc automobile de la CEET

- ☛ Diminution du nombre d'accidents et des pannes en raison d'une maintenance systématique.

Efficacité opérationnelle :

- ☛ Amélioration de la disponibilité des véhicules, réduisant ainsi les temps d'arrêt et augmentant la productivité.
- ☛ Optimisation des missions et des déplacements grâce à une gestion centralisée et une allocation efficace des ressources.

Durabilité :

- ☛ Prolongation de la durée de vie des véhicules grâce à une gestion proactive et préventive de leur maintenance.

1.3. ETUDE DE L'EXISTANT

L'étude de l'existant est une étape cruciale dans le développement de tout projet. Cette phase consiste à analyser en détail les systèmes, processus et outils actuels utilisés pour identifier leurs forces, faiblesses, opportunités d'amélioration et menaces. Cette phase permet de partir sur des bases solides en comprenant parfaitement les défis actuels.

Dans notre cas, elle nous a permis d'établir le processus de fonctionnement actuel du parc automobile de la CEET, qui se présente comme tel :

1. Acquisition de véhicules

- ☛ Une équipe du parc automobile vérifie les véhicules lors de la réception et met à jour l'inventaire.
- ☛ Les véhicules sont ensuite répartis dans les différentes directions en fonction des besoins par le processus d'affectation.

2. Exploitation des véhicules

- ☛ Les véhicules sont alloués en fonction des besoins (missions, réparations de matériels, achats, déplacements, feuilles de route).
- ☛ La division d'exploitation sélectionne le véhicule approprié et affecte un chauffeur.

3. Consommation de carburant

- ☛ La consommation est gérée par une citerne située dans l'enceinte du parc automobile.

Application de gestion du parc automobile de la CEET

- ✖ Les chauffeurs obtiennent du carburant en présentant des bons d'essence fournis par le chef de la division d'exploitation.
- ✖ En plus des bons de carburant, des cartes d'essence de compagnies comme Total ou Cap sont utilisées pour les missions.

4. Maintenance des véhicules

- ✖ Lorsqu'une voiture tombe en panne, le chauffeur la conduit au parc automobile.
- ✖ Le responsable enregistre les observations de l'utilisateur et effectue un diagnostic.
- ✖ Une fiche de maintenance est remplie et transmise au garage de mécanique.
- ✖ Une fiche de sortie de magasin ou de produits pétroliers est rédigé pour fournir les pièces détachées nécessaires pour la réparation.
- ✖ La fiche est envoyée au magasinier, qui sort les pièces nécessaires.
- ✖ Un mécanicien effectue la réparation du véhicule.
- ✖ Le véhicule passe une inspection finale avant d'être remis en service.

5. Gestion des stocks de pièces détachées

- ✖ Les sorties de pièces ne sont autorisées qu'en présence d'une fiche de sortie.
- ✖ Chaque pièce a un numéro de référence d'article.
- ✖ Le réapprovisionnement est initié par le chef de la section mécanique via un bon d'achat transmis au département des achats.

6. Gestion des documents

- ✖ La gestion des documents relatifs à chaque véhicule est faite par le parc automobile.
- ✖ Elle est également responsable des assurances, des rappels de visites techniques, et des réparations de routine pour tous les véhicules de la compagnie sur tout le territoire.

L'étude de l'existant révèle que le parc automobile de la CEET est actuellement géré avec des méthodes principalement manuelles et décentralisées, entraînant quelques insuffisances. Cependant, il existe des points forts dont la rigueur des processus de vérification et de maintenance, ainsi que dans la gestion centralisée des documents.

1.4. CRITIQUE DE L'EXISTANT

L'analyse du processus actuel de gestion du parc automobile de la CEET révèle à la fois des forces et des faiblesses notables.

1.4.1. Forces

- ✖ **Rigueur des Vérifications** : À la réception des véhicules, une équipe dédiée procède à des vérifications et contrôles minutieux, garantissant que les nouveaux véhicules sont en bon état avant d'être intégrés au parc.
- ✖ **Processus de Maintenance Structuré** : Le processus de diagnostic et de réparation est bien défini, impliquant plusieurs niveaux de validation et de contrôle, ce qui garantit que les véhicules sont correctement entretenus.
- ✖ **Gestion Centralisée des Documents** : La centralisation des documents relatifs aux véhicules (assurances, rappels de visites techniques, etc.) facilite le suivi et les interventions nécessaires.

1.4.2. Faiblesses

- ✖ **Gestion décentralisée** : Les processus sont longs et susceptibles d'erreurs humaines.
- ✖ **Indisponibilité des Véhicules** : Le risque d'indisponibilité des véhicules pour les missions urgentes est élevé en raison de la gestion manuelle.
- ✖ **Suivi de la Consommation de Carburant** : Le suivi manuel de la consommation de carburant, via des bons et des cartes d'essence, peut entraîner des gaspillages, des fraudes et des abus. Il est difficile de contrôler et d'optimiser la consommation sans une solution automatisée.
- ✖ **Complexité du Processus de Maintenance** : Le processus de maintenance est complexe et nécessite de nombreuses étapes et signatures.
- ✖ **Gestion des Stocks de Pièces Détachées** : La gestion manuelle des pièces détachées expose l'entreprise à des risques de ruptures de stock et de retards dans les réparations.
- ✖ **Gestion Documentaire Manuelle** : La gestion manuelle des documents peut entraîner des désorganisations et des pertes d'information.

Bien que la CEET ait mis en place un cadre de gestion relativement structuré pour son parc automobile, les méthodes manuelles et décentralisées actuelles

Application de gestion du parc automobile de la CEET

présentent des limites importantes. Soulignant ainsi le besoin urgent d'une solution automatisée et centralisée pour optimiser la gestion du parc automobile.

1.5. PROPOSITIONS ET CHOIX DE SOLUTIONS

Dans l'optique de trouver une solution sûre et durable face aux diverses insuffisances relevées plus haut, la Compagnie Énergie Électrique du Togo a mise en place deux (02) potentiels solutions :

- ✖ Une solution existante sur le marché : Fleet Complete



Figure 5 : Logo de la Société Fleet Complete

- ✖ Une solution développée sur mesure

1.5.1. Evaluation technique des solutions

1.5.1.1. Première solution : Fleet Complete

Fleet Complete est une solution canadienne de gestion de flotte largement utilisée qui offre une variété de fonctionnalités adaptées à la gestion des parcs automobiles. Cette solution propose des outils pour le suivi des véhicules en temps réel, la gestion des entretiens, le suivi de la consommation de carburant et des rapports analytiques pour optimiser les coûts opérationnels.

Avantages :

- ✖ Une solution prête à l'emploi avec déploiement rapide.
- ✖ Un large éventail de fonctionnalités adaptées à la gestion de flotte.
- ✖ Support technique régulier fourni par l'éditeur.

Inconvénients :

- ✖ Coût élevé pour les frais d'abonnement mensuels et coûts supplémentaires pour des fonctionnalités avancées.
- ✖ Les fonctionnalités ne sont pas entièrement adaptées aux besoins spécifiques de la CEET.
- ✖ Une Dépendance vis-à-vis d'un fournisseur externe pour les mises à jour et le support.

1.5.1.2. Deuxième solution : Développement d'une application

La deuxième solution consiste à développer une application personnalisée pour la gestion du parc automobile de la CEET. Cette application serait conçue pour répondre spécifiquement aux besoins et aux processus opérationnels de la CEET.

Avantages :

- ✖ L'application sera entièrement personnalisée pour répondre aux exigences spécifiques de la CEET.
- ✖ La CEET conserve le contrôle complet sur les fonctionnalités, les mises à jour et la maintenance de l'application.
- ✖ Éventuellement moins coûteux à long terme en éliminant les frais d'abonnement.

Inconvénients :

- ✖ Nécessite un temps de développement et de déploiement un peu plus long.
- ✖ Besoin de ressources internes pour la maintenance et les mises à jour.

1.5.2. Evaluations financières des solutions proposées

Pour faire notre évaluation financière, nous prenons une période d'étude de cinq (05) ans. En prenant en compte toutes les informations sur les différentes solutions, nous obtenons les résultats suivants : Les coûts sont exprimés en Francs CFA

Tableau 1 : Estimation des coûts des propositions de solution

| | Coûts initiaux | Coûts récurrents | Coûts sur cinq (05) ans |
|-------------------|--|--|---|
| Solution 1 | Frais de configuration et d'installation : 6.000.000 FCFA | <p>Abonnement mensuel : 30.000 par véhicule</p> <p>Nombre de véhicules : 200</p> <p>Coût mensuel total : 30.000 * 200 = 6.000.000</p> <p>Coût annuel total : 6.000.000 * 12 = 72.000.000 FCFA</p> | <p>Coût de configuration initial : 6.000.000</p> <p>Coût d'abonnement pour 5 ans : 72.000.000 * 5 = 360.000.000</p> <p>Coût total sur 5 ans : 6.000.000 + 360.000.000 = 366.000.000 FCFA</p> |
| Solution 2 | <p>1. Analyse et Conception : 750.000 FCFA</p> <ul style="list-style-type: none"> o Étude de faisabilité et conception technique. <p>2. Développement : 2.500.000 FCFA</p> <ul style="list-style-type: none"> o Développement Backend et Frontend, intégration des APIs, et tests. <p>3. Étude de terrain et déploiement : 500.000 FCFA</p> <ul style="list-style-type: none"> o Étude de terrain, adaptation, et déploiement. <p>4. Formation des utilisateurs : 750.000 FCFA</p> <ul style="list-style-type: none"> o Formation initiale (50 heures) et support post-formation (20 heures). <p>5. Frais divers : 500.000 FCFA</p> <ul style="list-style-type: none"> o Licences, outils de développement, et frais administratifs. <p>Total Estimé : 5.000.000 FCFA</p> | <ul style="list-style-type: none"> • Maintenance corrective : Réparation des bugs et problèmes techniques identifiés après la mise en production. • Maintenance évolutive : Ajout de nouvelles fonctionnalités ou améliorations basées sur les retours utilisateurs. • Mises à jour : Mise à jour régulière du système pour assurer sa compatibilité avec les nouvelles technologies et standards. • Support technique : Assistance continue aux utilisateurs pour résoudre les problèmes d'utilisation ou répondre à leurs questions. • Contrat de maintenance : Un contrat annuel définissant les services inclus, les délais d'intervention, et les conditions financières, avec possibilité de renouvellement automatique ou révision annuelle. <p>Total : 1 000 000 FCFA par an soit 20% du cout initial</p> | <p>Coût initial : 5.000.000</p> <p>Coût de maintenance et support pour 5 ans : 1.000.000 * 5 = 5.000.000</p> <p>Coût total sur 5 ans : 5.000.000 + 5.000.000 = 10.000.000 FCFA</p> |

1.5.3. Choix de la solution

Les deux solutions présentent des avantages et des inconvénients distincts.

Fleet Complete offre une solution rapide et complète, mais avec des coûts élevés et une personnalisation limitée. En revanche, une application développée sur mesure permet une personnalisation totale et un contrôle complet, malgré un investissement initial et un temps de développement plus importants. En plus notre évaluation

Application de gestion du parc automobile de la CEET

financière nous a permis d'avoir une estimation des couts des deux solutions sur cinq (05) ans :

- ☒ La première solution coûtera environ 366.000.000 FCFA sur cinq ans¹
- ☒ La seconde solution coûtera environ 10.000.000 FCFA sur cinq ans

En tenant compte de ces facteurs, il est clair que le développement d'une application interne constitue la solution idéale. Cette approche non seulement optimise les coûts à long terme, mais assure également une flexibilité maximale et une adéquation parfaite avec les processus opérationnels de la CEET. Par conséquent, nous procèderons à la mise en œuvre de la solution développée en interne.

1.6. PLANNING PRÉVISIONNEL DE RÉALISATION

Pour assurer une mise en œuvre structurée et efficace de la solution développée en interne, un planning prévisionnel détaillé a été élaboré, couvrant toutes les étapes clés du projet.

Nous avons un tableau présentant les différentes activités ainsi que leur date de début et de fin et la durée prévu pour leur réalisation.

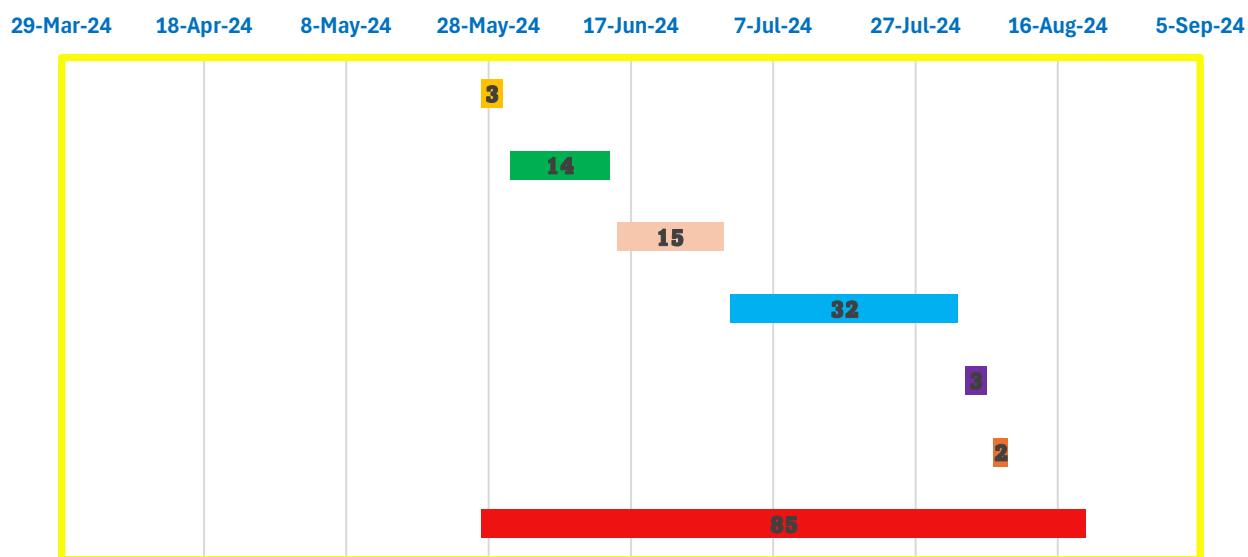
Également un diagramme de Gantt dont la couleur de chaque bande correspond à une activité énoncé dans le tableau de la même couleur.

¹ Source des données financières de la solution fleet complete : <https://www.fleetcomplete.com/fr-ca/>

Tableau 2 : Tableau des différentes activités et les périodes d'exécution

| N° | Activité | Date Début | Durée (jrs) | Date Fin |
|----|---|------------|-------------|------------|
| 1 | Gestation et planification. Choix du thème ; Etude du thème ; Choix des technologies ; Élaboration du cahier des charges ; Réalisation du planning prévisionnel des activités ; | 27-mai-24 | 3 | 30-mai-24 |
| 2 | Analyse et spécification. Rédaction des questionnaires pour les interviews ; Collecte des informations ; Interviews ; | 31-mai-24 | 14 | 14-juin-24 |
| 3 | Conception. Proposition d'architecture ; Elaborations des diagrammes ; Création de la maquette ; | 15-juin-24 | 15 | 30-juin-24 |
| 4 | Implémentation. | 01-juil-24 | 32 | 02-aout-24 |
| 5 | Vérification et Installation. Tests et résolution des bugs. Déploiement ; Résolution des problèmes de déploiement. | 03-aout-24 | 3 | 06-aout-24 |
| 6 | Formation des utilisateurs. | 07-aout-24 | 2 | 09-aout-24 |
| 7 | Rédaction du mémoire. | 27-mai-24 | 85 | 18-aout-24 |

Diagramme de Gantt

Figure 6 : Diagramme de Gantt des activités



2. Partie II : Analyse et Conception

Application de gestion du parc automobile de la CEET

La phase d'analyse et de conception représente un jalon critique dans le développement de solutions informatiques, marquant la transition entre la compréhension des besoins et la formalisation de ces besoins en un cadre structuré. Dans cette partie du mémoire, nous explorerons en détail les méthodologies et les approches utilisées pour analyser les exigences du système de gestion du parc automobile de la CEET. Nous décrirons également la conception de l'architecture du système, les modèles de données, et les interfaces utilisateur, en mettant en lumière comment ces éléments s'intègrent pour former une solution cohérente et efficace. Cette étape est cruciale pour assurer que le système final répond aux attentes fonctionnelles.

2.1. CHOIX DE LA METHODE D'ANALYSE ET JUSTIFICATION

La méthodologie d'analyse et de conception joue un rôle fondamental dans le développement de systèmes informatiques. Cette étape cruciale permet d'aligner précisément le développement du système avec les exigences spécifiques du client.

On distingue principalement deux (02) types d'approches d'analyse :

- ◉ L'approche systémique ou fonctionnelle
- ◉ L'approche orientée objet

2.1.1. Approche systémique ou fonctionnelle

L'approche systémique ou fonctionnelle, structure le système en divers sous-systèmes interactifs, facilitant l'analyse par la simplification des composants complexes. Des méthodologies telles que MERISE supportent cette approche.

2.1.2. Approche orientée objet

L'approche orientée objet envisage le système comme un ensemble d'objets interagissant, où chaque objet représente une entité avec des attributs et des comportements spécifiques. Cette approche est essentielle pour les systèmes modernes car elle permet une modularité accrue, facilite la maintenance par l'encapsulation, et améliore la réutilisabilité des composants via des classes bien définies.

Synthèse et choix de l'approche :

Avec les défis posés par la complexité croissante des systèmes informatiques contemporains, l'approche fonctionnelle montre ses limites, notamment en matière de flexibilité et d'adaptabilité. L'approche orientée objet, en revanche, offre une réponse

Application de gestion du parc automobile de la CEET

robuste à ces défis grâce à sa capacité à modéliser des systèmes complexes de manière intuitive et alignée sur les réalités opérationnelles, rendant la maintenance et l'évolution du système plus gestionnaires.

Ce qui nous amené à utiliser l'approche orientée objet comme méthode d'analyse.

2.1.3. Langage de modélisation

L'UML a été choisi comme le langage de modélisation standard pour ce projet en raison de sa capacité à fournir une variété de diagrammes, supportant ainsi une visualisation complète des aspects statiques et dynamiques du système. UML aide à clarifier les exigences et à documenter les structures de manière cohérente et compréhensible.



Figure 7 : Logo UML

2.1.4. Processus Two Track Unified Process (2TUP)

La méthode 2TUP, qui s'inscrit dans le cadre des processus unifiés, est conçue pour s'adapter aux cycles de vie des projets en fournissant une structure claire divisée en deux grands aspects :

- ⌚ Aspect technique
- ⌚ Aspect fonctionnel

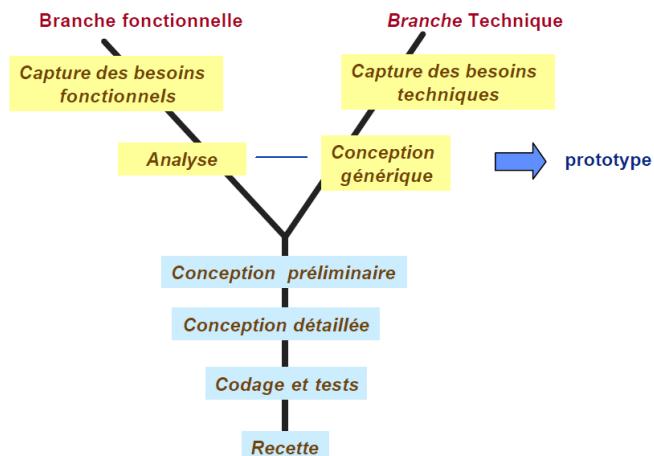


Figure 8 : Processus 2TUP

2.1.5. Synthèse

La combinaison de l'approche orientée objet avec les outils UML et du processus 2TUP crée une fondation solide pour le développement du système de gestion du parc automobile de la CEET. Cette configuration assure non seulement l'alignement avec les objectifs stratégiques mais également l'adaptabilité nécessaire pour répondre aux évolutions futures des exigences.

2.2. CHOIX DE L'OUTIL DE MODÉLISATION ET JUSTIFICATION

Après une évaluation approfondie des options disponibles, nous avons sélectionné draw.io comme notre outil de modélisation privilégié. Draw.io se distingue par sa simplicité d'utilisation, sa compatibilité étendue avec divers formats de fichier, et sa capacité à intégrer facilement des diagrammes UML, des flux de processus, et des architectures système. En outre, étant une plateforme basée sur le web, draw.io facilite la collaboration en temps réel entre les membres de l'équipe, permettant ainsi des mises à jour et des révisions instantanées, ce qui est crucial pour maintenir l'alignement et la précision tout au long du cycle de développement du projet.



Figure 9 : Logo draw.io

2.3. ETUDE DÉTAILLÉE DE LA SOLUTION

Cette étape est essentielle pour visualiser la structure et le fonctionnement du système proposé. Nous détaillerons les diagrammes de classes, de séquences, d'activités, de cas d'utilisations et les descriptions textuelles, qui ensemble, illustrent les interactions, les processus et les comportements au sein du système.

2.3.1. Modélisation des aspects fonctionnels de la solution

Cette sous-section se concentre sur la modélisation des aspects fonctionnels du système proposé pour la gestion du parc automobile de la CEET. Nous utilisons des diagrammes de cas d'utilisation pour décrire les fonctionnalités et les interactions au sein du système.

2.3.1.1. Diagrammes de cas d'utilisations



Les acteurs :

Un acteur représente une entité extérieure qui interagit avec le système, typiquement un utilisateur humain ou un autre système. Dans notre cas, les différents acteurs seront : l'administrateur, le chef service, le contrôleur, l'exploitant, pompiste, le magasinier, le mécanicien.



Les cas d'utilisations :

Un cas d'utilisation en UML est une description des actions ou des séquences d'actions, y compris les variantes, que le système effectue en interagissant avec un ou plusieurs acteurs.

Tableau 3 : Tableau récapitulatif des cas d'utilisations

| Acteurs | Cas d'utilisations | | |
|---|------------------------|-----------------------------------|--|
| Administrateur / Chef de service | Gérer les utilisateurs | Créer un utilisateur | |
| | | Modifier un utilisateur | |
| | | Supprimer un utilisateur | |
| Exploitant / Administrateur / Chef de service | Gérer les véhicules | Créer un véhicule | |
| | | Modifier un véhicule | |
| | | Supprimer un véhicule | |
| Administrateur / Chef de service | Gérer les sorties | Créer une fiche de sortie | |
| | | Modifier une fiche de sortie | |
| | | Supprimer une fiche de sortie | |
| Exploitant / Administrateur / Chef de service | | Créer une fiche de maintenance | |
| | | Modifier une fiche de maintenance | |
| Contrôleur / Administrateur / Chef de service | Gérer les maintenances | | |
| Administrateur / Chef de service | | | |

Application de gestion du parc automobile de la CEET

| | | |
|---|--------------------------------------|---|
| | | Supprimer une fiche de maintenance |
| Pompiste / Administrateur / Chef de service | Gérer les approvisionnement | Créer une fiche d'approvisionnement |
| Administrateur / Chef de service | | Modifier une fiche d'approvisionnement |
| | | Supprimer une fiche d'approvisionnement |
| Exploitation / Administrateur / Chef de service | Gérer les documents | Créer un document |
| Administrateur / Chef de service | | Supprimer un document |
| | | Modifier un document |
| Mécanicien / Administrateur / Chef de service | Gérer les bons de sorties magasin | Créer un bon de sortie |
| Administrateur / Chef de service | | Modifier un bon de sortie |
| | | Supprimer un bon de sortie |
| Mécanicien / Administrateur / Chef de service | Gérer les achats | Créer une fiche d'achat |
| Administrateur / Chef de service | | Modifier une fiche d'achat |
| | | Supprimer une fiche d'achat |
| Magasinier / Administrateur / Chef de service | Gérer le magasin de pièces détachées | Ajouter une pièce |
| Administrateur / Chef de service | | Modifier une pièce |
| | | Supprimer une pièce |
| Administrateur | Gérer les rôles | |
| Administrateur / Chef de service | Voir les statistiques | |



Diagrammes de cas d'utilisation par acteurs :

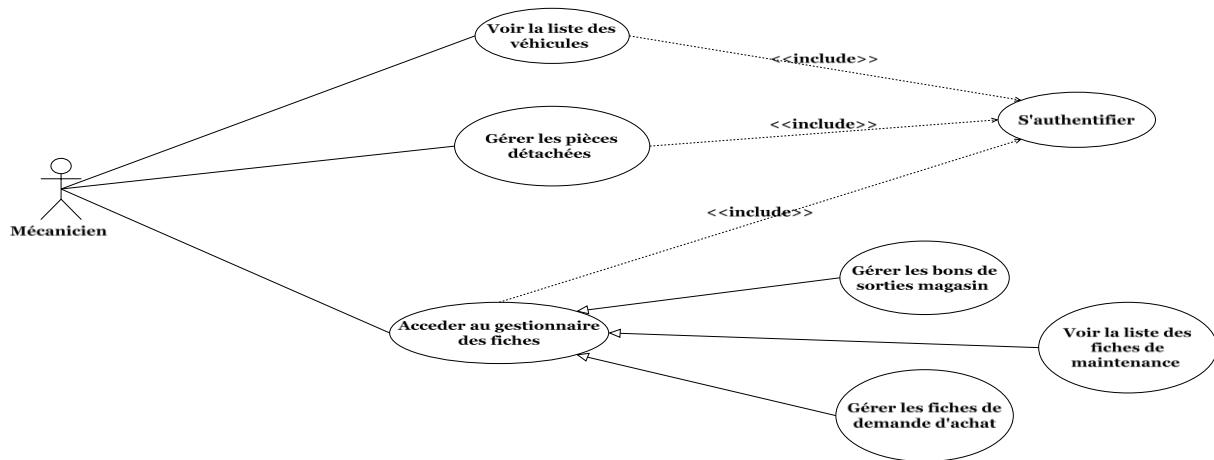


Figure 10 : Cas d'utilisation MECANICIEN

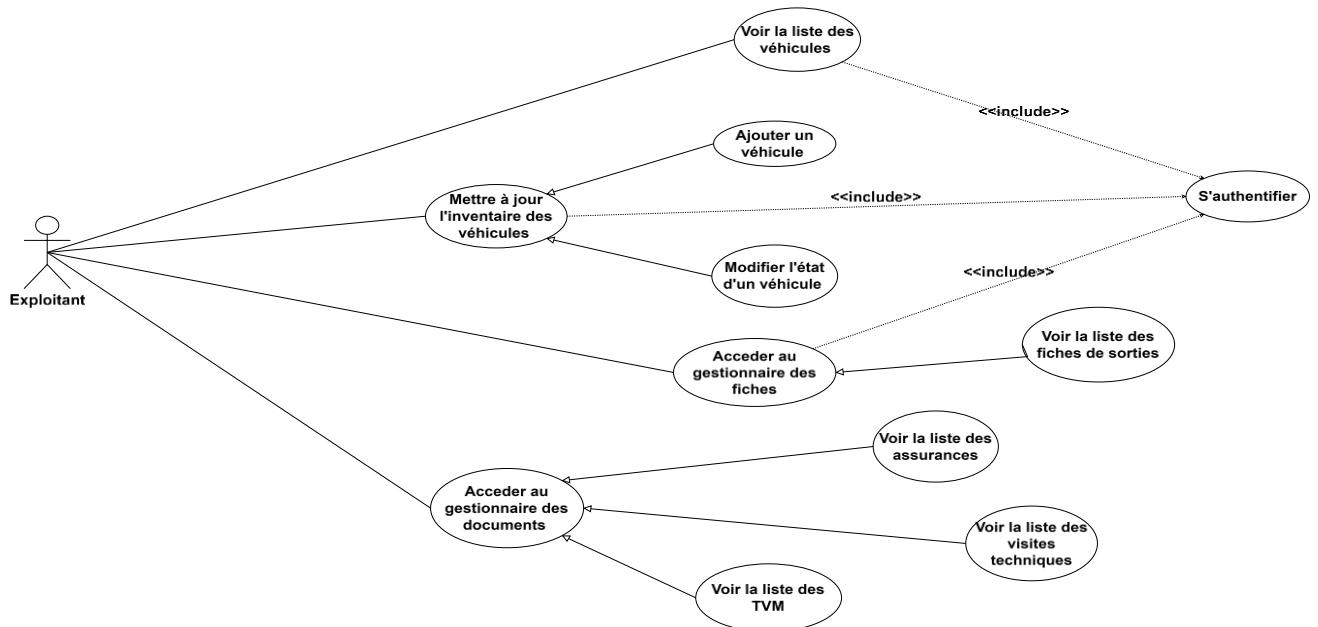


Figure 11 : Cas d'utilisation EXPLOITANT

Application de gestion du parc automobile de la CEET



Figure 12 : Cas d'utilisation CHEF SERVICE

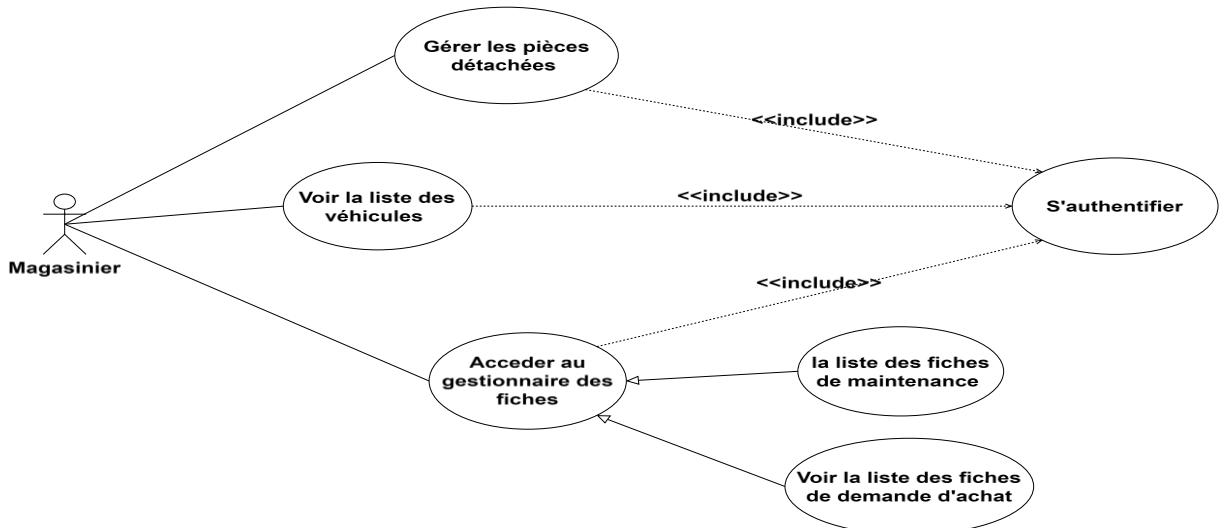


Figure 13 : Cas d'utilisation MAGASINIER

Application de gestion du parc automobile de la CEET

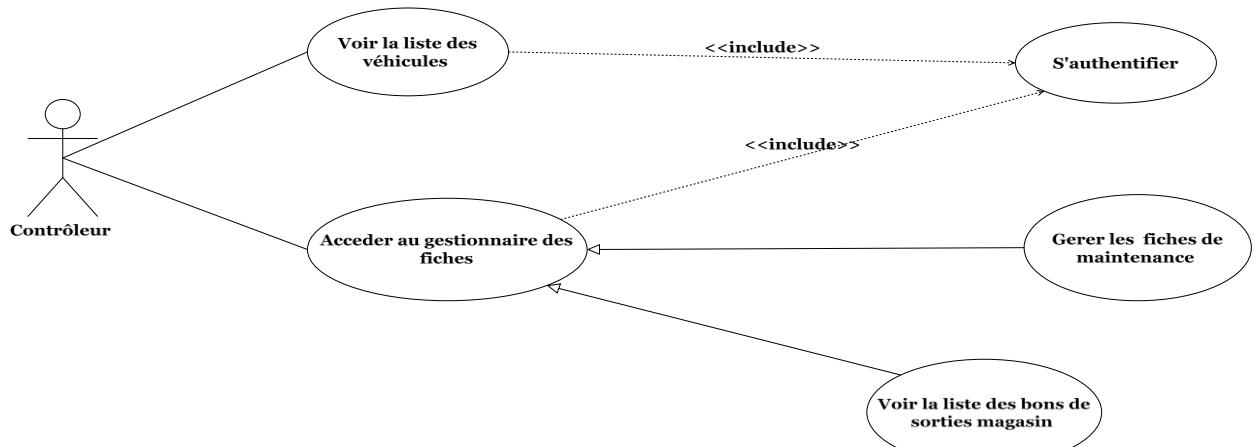
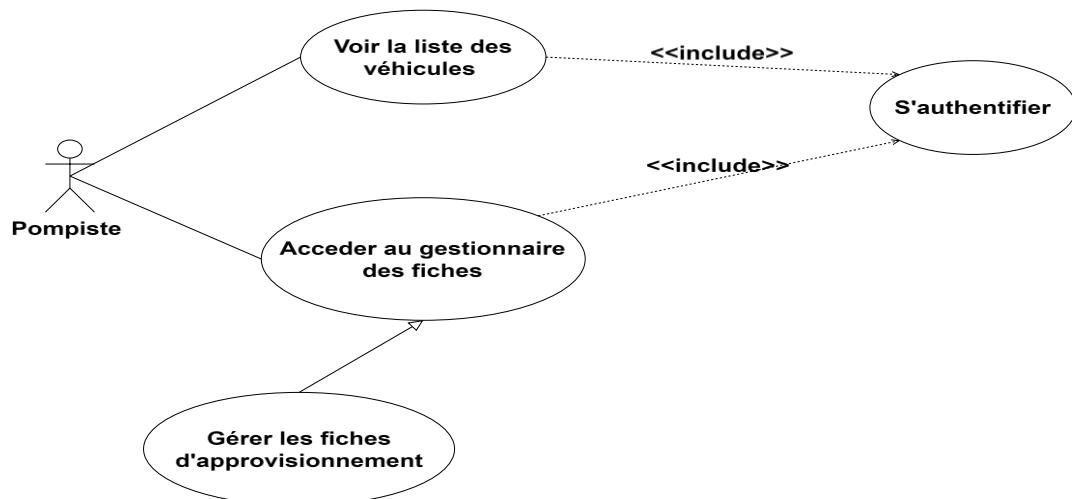
Figure 14 : Cas d'utilisation CONTROLEURFigure 15 : Cas d'utilisation POMPISTE

Diagramme de cas d'utilisation général

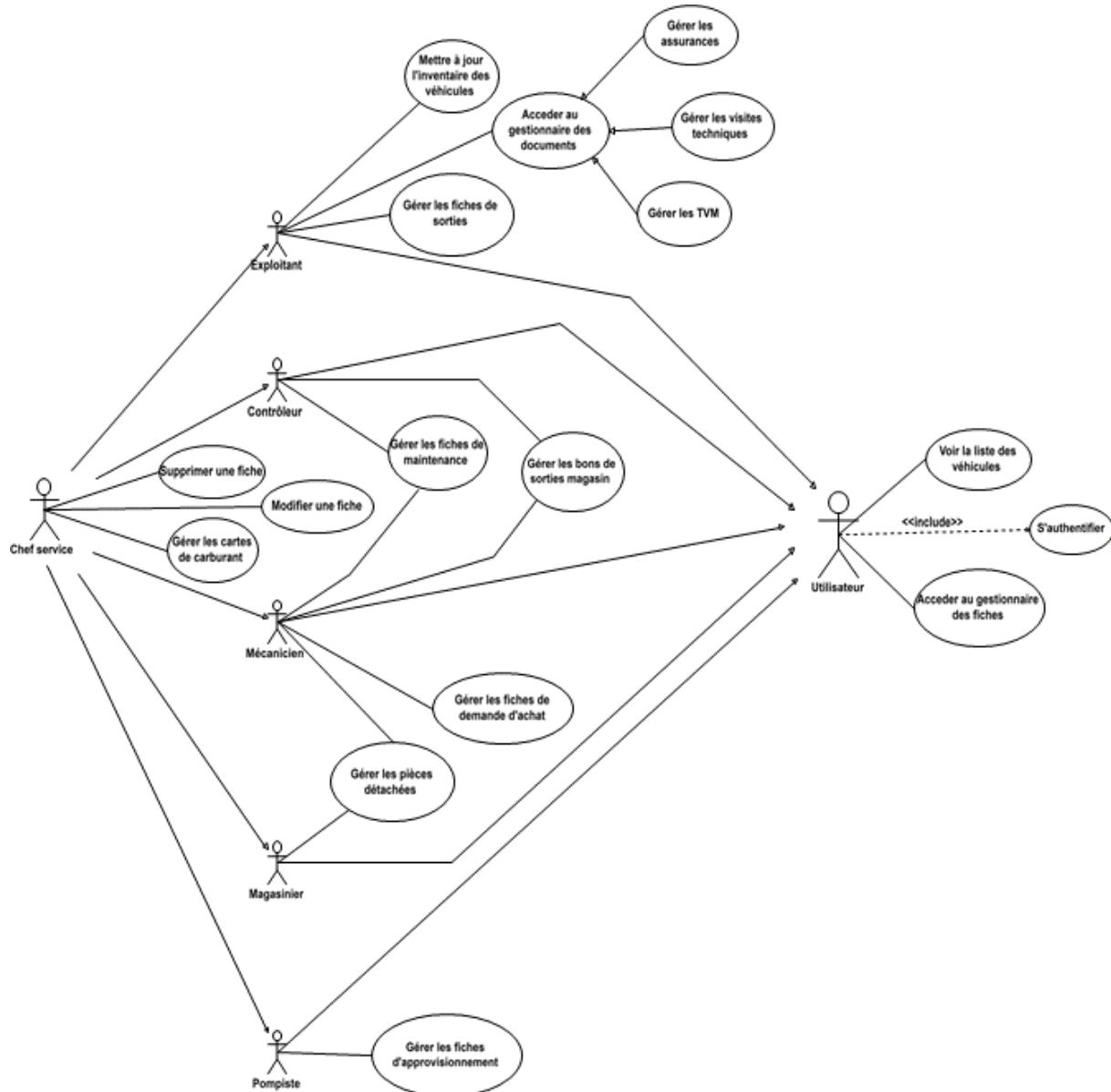


Figure 16 : Cas d'utilisation GENERAL

2.3.1.2. Descriptions textuelles de quelques cas d'utilisations

Ces descriptions fournissent une explication détaillée des processus, des interactions et des comportements attendus au sein du système.



Créer une fiche d'approvisionnement

Titre : Approvisionnement.

Acteurs : Chef de Service, Pompiste, Système.

Résumé : La création d'une fiche d'approvisionnement d'un véhicule pour gérer la consommation de carburant.

Auteur : FLINDJO Namtante.

Version : 1.0

Date de création : 22 juillet 2024

Acteurs principaux : Chef service, Pompiste.

Contexte : Ce cas d'utilisation décrit le processus de création d'une fiche d'approvisionnement en carburant pour les véhicules de l'entreprise. Les acteurs impliqués sont le Chef de Service et le Pompiste, qui interagissent avec le Système pour enregistrer les approvisionnements en carburant.

Préconditions :

- ⊕ L'acteur doit être authentifié et autorisé à créer une fiche d'approvisionnement.
- ⊕ Les informations sur les véhicules et les cartes de carburant doivent être disponibles dans le système.

Scénario Nominal

1. Initiation du processus :

- Le Chef de Service ou le Pompiste initie la création d'une fiche d'approvisionnement en carburant en accédant au système. (E1)
- Ils sélectionnent l'option pour créer une nouvelle fiche d'approvisionnement.

2. Récupération des informations du véhicule :

- Le système demande l'identifiant du véhicule à approvisionner.
- L'acteur entre l'identifiant du véhicule. (A2)
- Le Système récupère les informations correspondantes au véhicule à partir de la base de données. (A1)

3. Vérification de la présence d'un carnet de bon lorsque la création est initiée par le Pompiste :

- Une condition alternative vérifie si le véhicule dispose d'un carnet de bon :
 - Oui : Le système demande les informations de la fiche du carnet de bon. L'acteur les fournit.

Application de gestion du parc automobile de la CEET

- Non : Le système demande les informations du bon de carburant. L'acteur les fournit. (E3)
4. Récupération des informations de la carte de carburant, si le processus a été initié par le chef service et si le véhicule en dispose. (E2)
5. Création de la Fiche d'Approvisionnement :
- Le système crée une nouvelle fiche d'approvisionnement en utilisant les informations fournies.
 - Les informations incluent l'identifiant du bénéficiaire et d'autres détails pertinents.
6. Mise à Jour des Informations :
- Le Système met à jour les informations de la carte de carburant liée au véhicule.
 - Le système enregistre les informations du reçu de la station-service.
7. Le système confirme la création de la fiche d'approvisionnement et informe l'acteur que l'opération a été réalisée avec succès.

Scénarios Alternatifs

A1. Erreur de Validation :

- Si les informations fournies ne sont pas valides ou complètes, le système affiche un message d'erreur et demande à l'acteur de corriger les informations.

A2. Véhicule Non Reconnu :

- Si l'identifiant du véhicule n'existe pas dans la base de données, le système informe l'acteur que le véhicule n'est pas reconnu et demande un identifiant valide.

Scénarios d'exceptions

E1. Problème de connexion au système :

- Si le système est inaccessible, l'acteur reçoit un message d'erreur et doit réessayer plus tard.

E2. Carte de carburant non valide :

- Si la carte de carburant est expirée ou bloquée, le système informe l'acteur et demande une carte valide.

E3. Informations manquantes dans le carnet de bon :

- Si les informations du carnet de bon sont incomplètes, le système affiche une erreur et demande de les compléter.

Postconditions

- La fiche d'approvisionnement est enregistrée dans le système.
- Les informations de la carte de carburant sont mises à jour.
- Un reçu de la station-service est enregistré.



Créer une fiche de sortie

Titre : Création d'une Fiche de Sortie

Acteurs : Exploitant, Système

Résumé : Ce cas d'utilisation décrit le processus de création d'une fiche de sortie pour un véhicule par un exploitant.

Auteur : FLINDJO Namtante

Version : 1.0

Date de création : 22 juillet 2024

Acteurs principaux : Exploitant

Contexte : Ce cas d'utilisation détaille comment un exploitant crée une fiche de sortie pour un véhicule. Le système vérifie la disponibilité du véhicule et demande une date ultérieure en cas d'indisponibilité.

Préconditions :

- L'exploitant doit être authentifié et autorisé à créer une fiche de sortie.
- Les informations sur les véhicules doivent être disponibles dans le système.

Scénario Nominal

1. Initiation du processus :

- L'exploitant initie la création d'une fiche de sortie en accédant au système. (E1)
- Il sélectionne l'option pour créer une nouvelle fiche de sortie.

2. Vérification de la disponibilité du véhicule :

- Le système demande l'identifiant du véhicule à sortir.
- L'exploitant entre l'identifiant du véhicule. (E2)

- Le système vérifie la disponibilité du véhicule.

3. Disponibilité du véhicule :

- Si le véhicule est disponible, le système passe à l'étape de création de la fiche de sortie.
- Si le véhicule n'est pas disponible, une condition alternative est activée. (A1)

4. Création de la fiche de sortie :

- Le système crée une fiche de sortie avec les informations fournies.
- Les informations incluent l'identifiant du bénéficiaire et d'autres détails pertinents.

5. Confirmation et enregistrement :

- Le système confirme la création de la fiche de sortie et informe l'exploitant que l'opération a été réalisée avec succès.
- La fiche de sortie est enregistrée dans le système.

Scénario Alternatifs

A1. Vérification de la planification :

- Le système demande à l'exploitant s'il souhaite planifier la sortie à une date ultérieure.
 - Si l'exploitant choisit de planifier, il entre une date ultérieure.
 - Le système vérifie la validité de la date.

A2. Retour au gestionnaire de fiches :

- Si l'exploitant choisit de ne pas planifier, il est redirigé vers le gestionnaire de fiches.

Scénarios Alternatifs généraux

• Erreur de Validation :

- Si les informations fournies ne sont pas valides ou complètes, le système affiche un message d'erreur et demande à l'exploitant de corriger les informations.

• Véhicule Non Reconnu :

- Si l'identifiant du véhicule n'existe pas dans la base de données, le système informe l'exploitant que le véhicule n'est pas reconnu et demande un identifiant valide.

Scénarios d'Exceptions

E1. Problème de connexion au système :

- Si le système est inaccessible en raison d'un problème technique, l'exploitant reçoit un message d'erreur indiquant que la connexion a échoué.

E2. Véhicule déjà sorti :

- Si le véhicule sélectionné est déjà sorti ou n'est pas disponible pour d'autres raisons, le système informe l'exploitant de la situation.

Postconditions :

- La fiche de sortie est enregistrée dans le système.

Les informations pertinentes sont mises à jour dans la base de données du système.



[Créer une fiche de maintenance](#)

Titre : Création d'une Fiche de Maintenance

Acteurs : Mécanicien, Contrôleur, Système

Résumé : La création d'une fiche de maintenance pour un véhicule, incluant le choix des pièces nécessaires et la génération d'un bon de sortie du magasin.

Auteur : FLINDJO Namtante

Version : 1.0

Date de création : 22 juillet 2024

Acteurs principaux : Mécanicien, Contrôleur

Contexte : Ce cas d'utilisation décrit le processus de création d'une fiche de maintenance pour les véhicules de l'entreprise. Les acteurs impliqués sont le Mécanicien et le Contrôleur, qui interagissent avec le Système pour enregistrer les besoins en pièces de rechange pour la maintenance.

Préconditions :

1. L'acteur doit être authentifié et autorisé à créer une fiche de maintenance.
2. Les informations sur les véhicules, départements et pièces doivent être disponibles dans le système. (A1)

Scénario Nominal :

1. Initiation du processus :

Application de gestion du parc automobile de la CEET

- Le Mécanicien ou le Contrôleur initie la création d'une fiche de maintenance en accédant au système.
- Ils sélectionnent l'option pour créer une nouvelle fiche de maintenance.

2. Récupération des informations de maintenance :

- Le système demande l'identifiant du véhicule et d'autres informations pertinentes.
- L'acteur entre les informations requises.
- Le Système récupère les informations correspondantes à partir de la base de données.

3. Création de la Fiche de Maintenance :

- Le système crée une nouvelle fiche de maintenance en utilisant les informations fournies.
- Les informations incluent l'identifiant du bénéficiaire (véhicule) et d'autres détails pertinents.
- Le système vérifie et associe le département responsable.

4. Création du Bon de Sortie Magasin :

- Après la création de la fiche de maintenance, le mécanicien peut créer un bon de sortie magasin.
- Le système demande les pièces nécessaires pour la maintenance.
- L'acteur sélectionne les pièces requises à partir de la liste fournie par le système.

5. Choix des pièces :

- Le système affiche la liste des pièces disponibles.
- L'acteur choisit les pièces nécessaires pour la maintenance. (A2), (E1)

6. Validation et Confirmation :

- Le système crée un bon de sortie magasin en utilisant les informations des pièces sélectionnées.
- Le bon de sortie est enregistré dans le système.

7. Mise à Jour des Informations :

- Le Système met à jour les informations de la fiche de maintenance avec les détails des pièces utilisées.
- Le système enregistre les informations du bon de sortie du magasin.

8. Confirmation de la Création :

- Le système confirme la création de la fiche de maintenance et du bon de sortie magasin.
- L'acteur est informé que l'opération a été réalisée avec succès.

Scénarios Alternatifs

A1. Véhicule Non Reconnu :

- Si l'identifiant du véhicule n'existe pas dans la base de données, le système informe l'acteur que le véhicule n'est pas reconnu et demande un identifiant valide.

A2. Pièces Non Disponibles :

- Si les pièces sélectionnées ne sont pas disponibles, le système informe l'acteur et propose des alternatives ou demande de sélectionner d'autres pièces.

Scénarios d'Exceptions

E1. Pièces en rupture de stock :

- Si les pièces nécessaires pour la maintenance ne sont pas disponibles en stock, le système informe l'acteur de la situation.

Postconditions :

1. La fiche de maintenance est enregistrée dans le système.
2. Les informations des pièces utilisées sont mises à jour.
3. Un bon de sortie magasin est enregistré.
4. L'acteur reçoit une confirmation de la création de la fiche de maintenance et du bon de sortie magasin.

2.3.2. Modélisation de l'aspect dynamique de la solution

Pour donner suite à la modélisation des aspects fonctionnels, nous abordons maintenant la dimension dynamique de notre solution. Cette phase capture les interactions entre les acteurs et le système, ainsi que la séquence des messages échangés, à travers l'utilisation de diagrammes de séquences et d'activités système. Étant donné la complexité des dynamiques système, plusieurs diagrammes de séquences

Application de gestion du parc automobile de la CEET

seront utilisés pour couvrir différentes sous-fonctions du système, assurant ainsi une représentation complète et détaillée des interactions dynamiques.

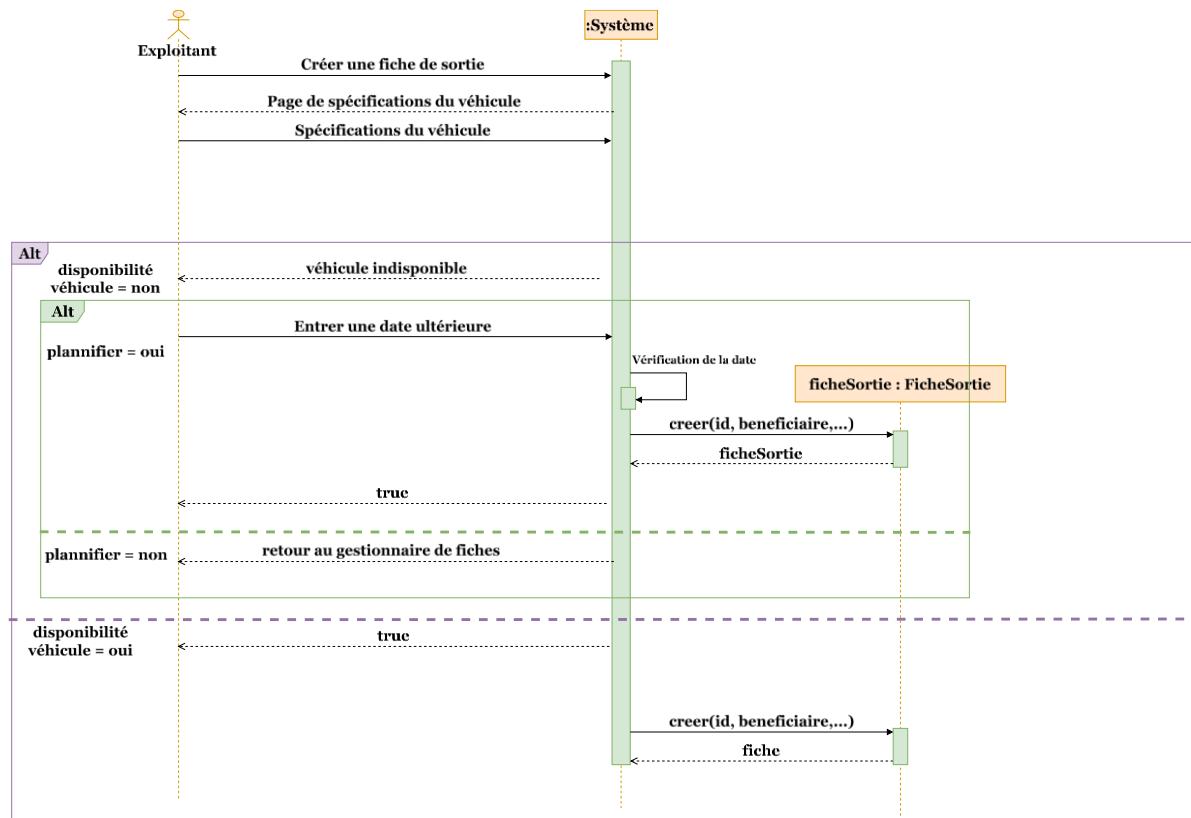

Diagramme de séquence Exploitation


Figure 17 : Diagramme de séquence : EXPLOITATION


Diagramme de séquence Maintenance

Application de gestion du parc automobile de la CEET

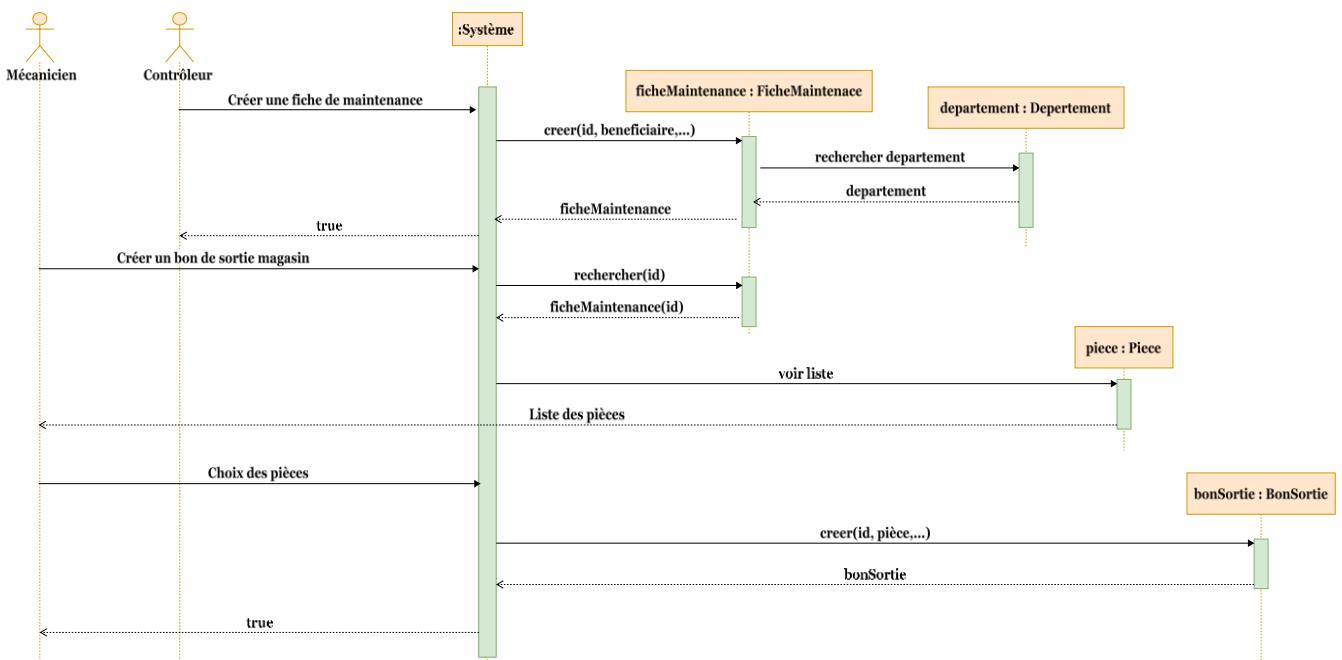


Figure 18 : Diagramme de séquence : MAINTENANCE



Diagramme de séquence APPROVISIONNEMENT

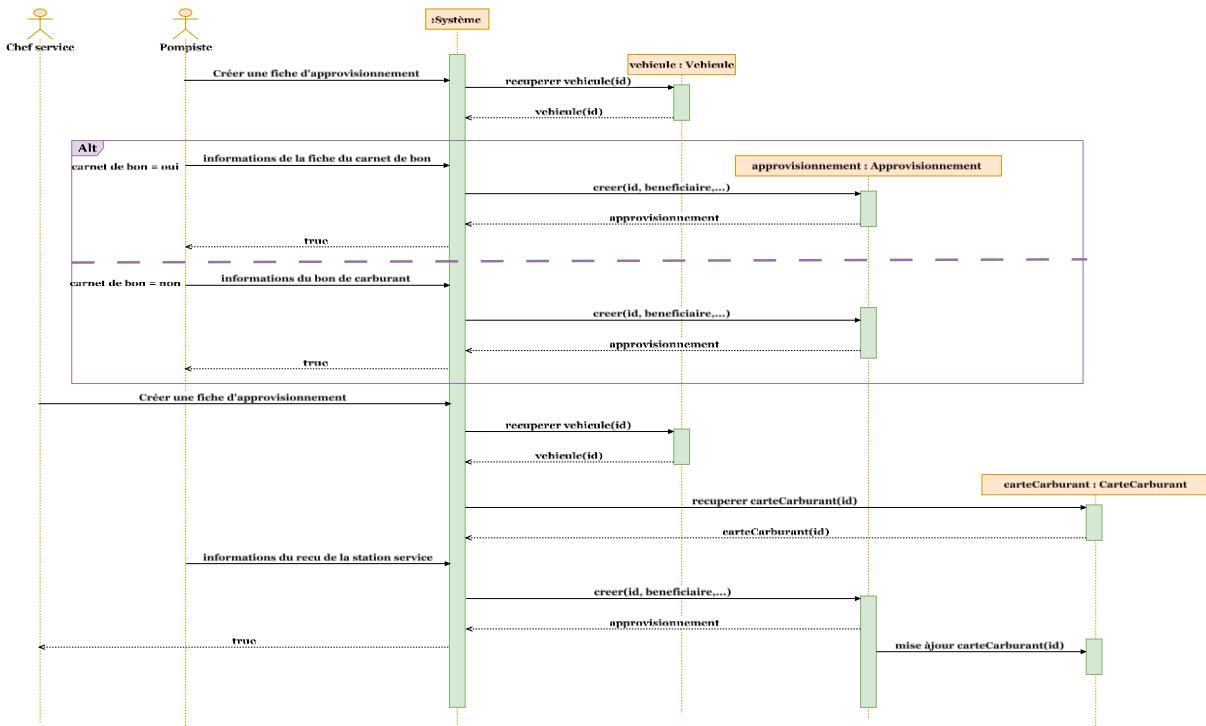
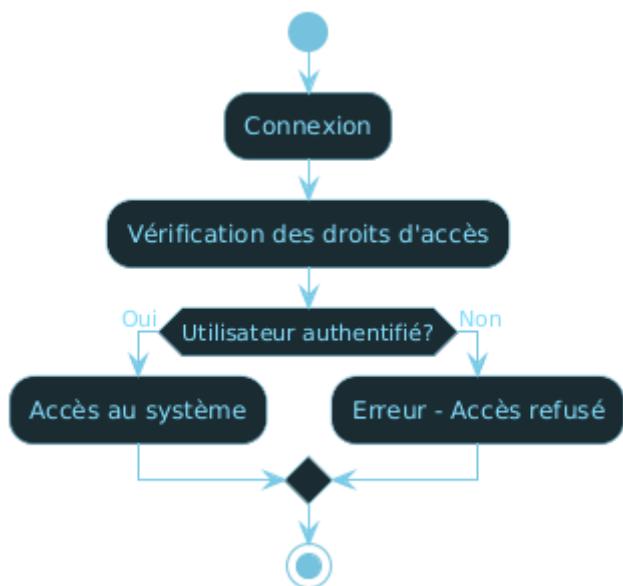
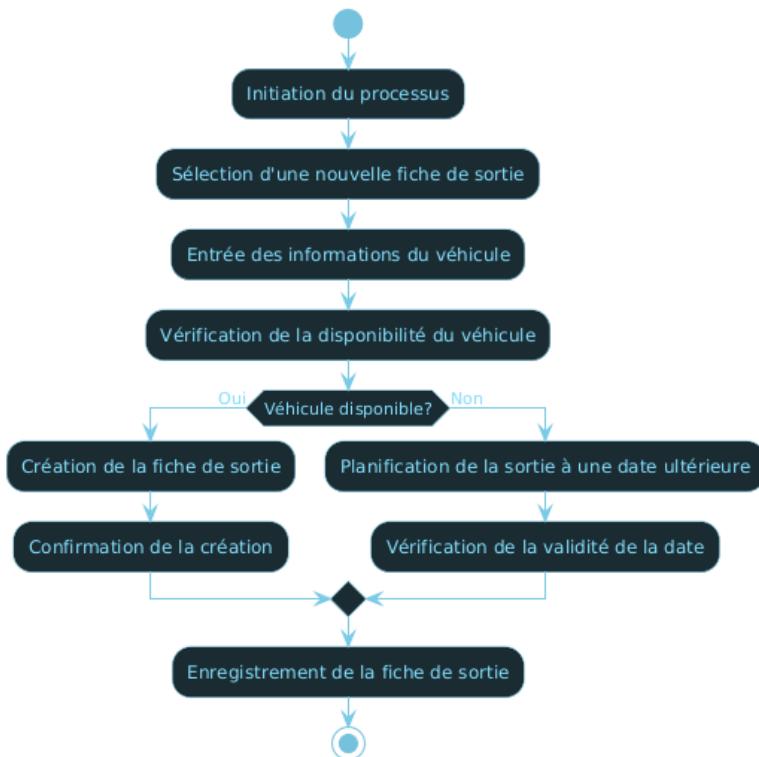


Figure 19 : Diagramme de séquence : APPROVISIONNEMENT

Application de gestion du parc automobile de la CEET

Diagramme d'activités : Authentification*Figure 20 : Diagramme d'activités <<Connexion>>*Diagramme d'activités : Sortie de véhicule*Figure 21 : Diagramme d'activités <<Sortie>>*

Application de gestion du parc automobile de la CEET

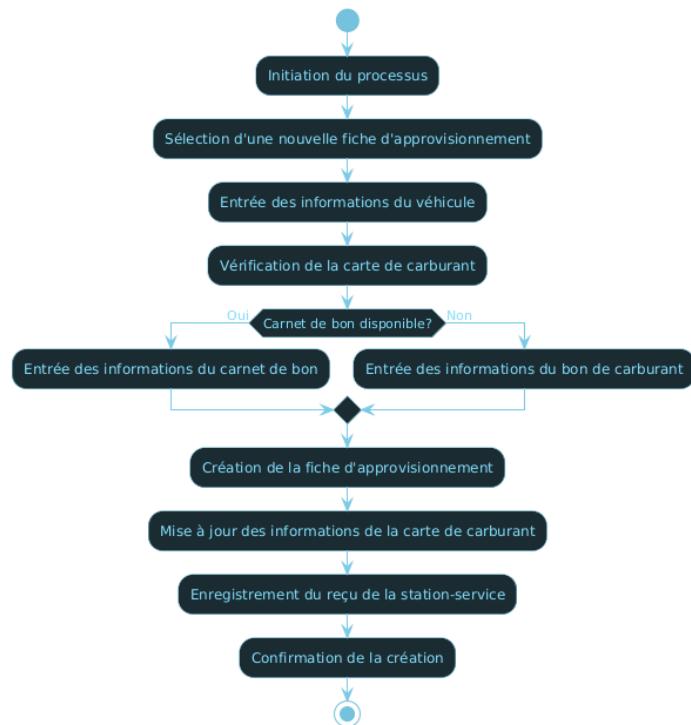
Diagramme d'activités Approvisionnement

Figure 22 : Diagramme d'activités <<Approvisionnement>>

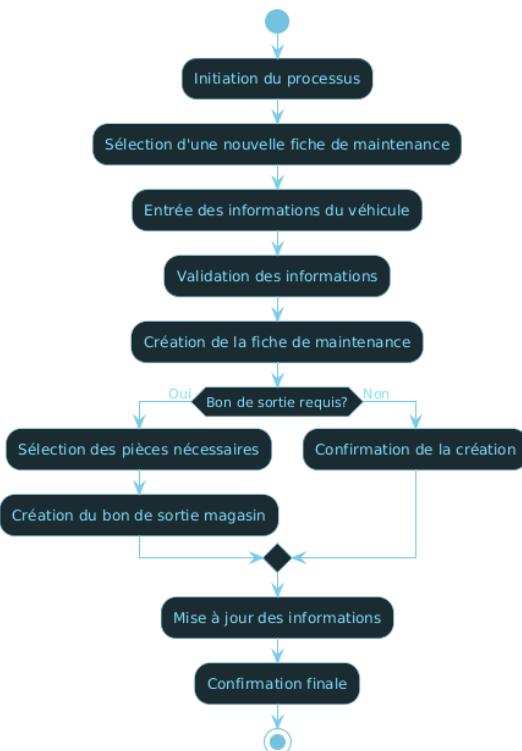
Diagramme d'activités : Maintenance

Figure 23 : Diagramme d'activités <<Maintenance>>

2.3.3. Modélisation des aspects statiques de la solution

Cette section se consacre à la structure logique du système, abordant les aspects statiques à travers les diagrammes de classes et d'objets. Les diagrammes de classes détaillent les relations et interactions entre les composants du système, tandis que les diagrammes d'objets illustrent des exemples spécifiques de ces interactions dans des scénarios réels. Ensemble, ces modélisations fournissent une vue claire de l'architecture interne et de la structuration des données au sein de l'application.

Application de gestion du parc automobile de la CEET

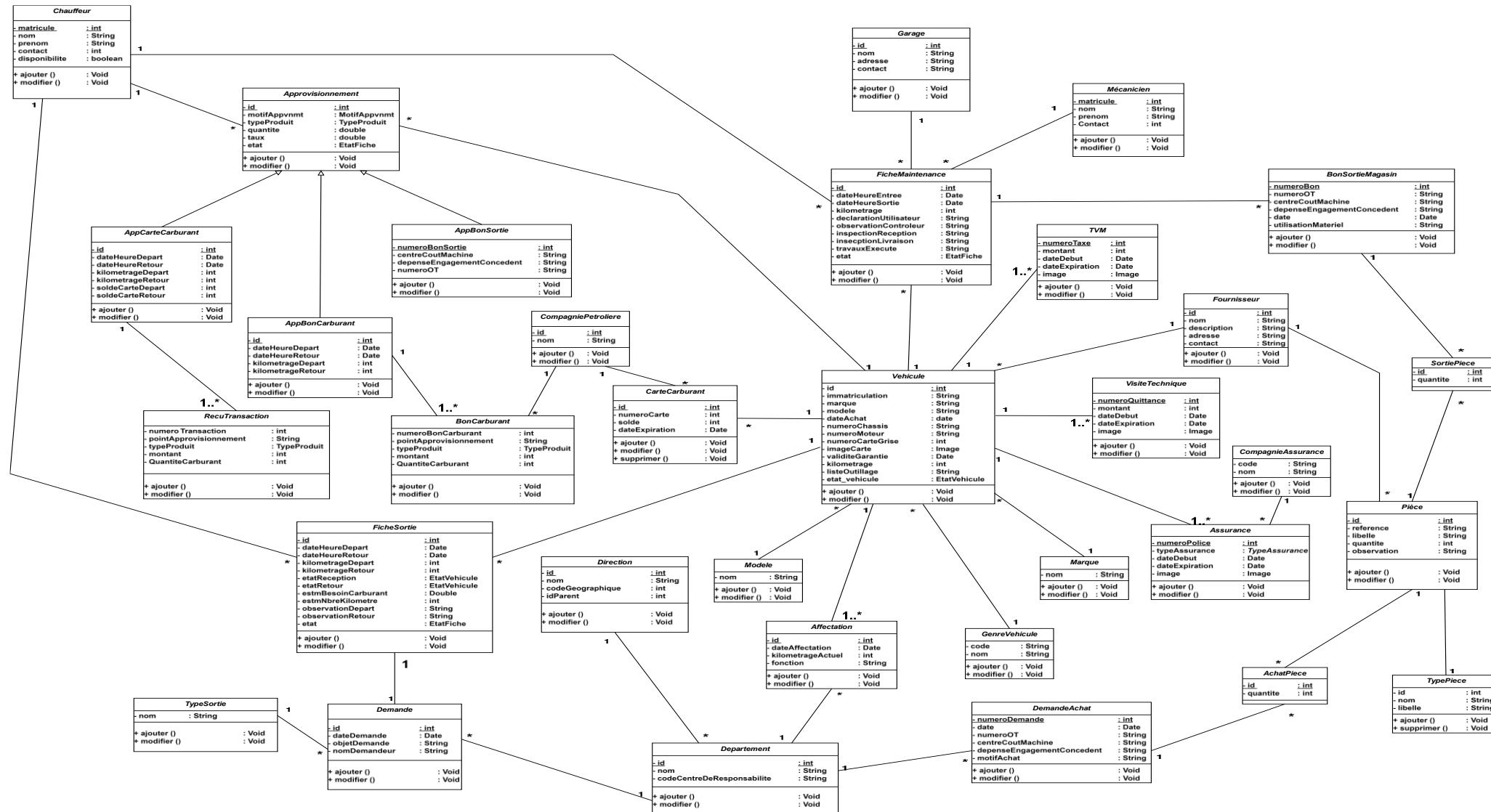


Figure 24 : Diagramme de classe

3. Partie III : réalisation et Mise en œuvre

La troisième partie de ce mémoire se concentre sur la phase de réalisation et de mise en œuvre de notre système de gestion du parc automobile pour la CEET. Cette étape cruciale combine la théorie avec la pratique. Nous y détaillerons les ressources matérielles et logicielles mises en jeu, l'architecture déployée pour soutenir l'application, les mesures de sécurité intégrées, ainsi que la structuration de la base de données qui soutiendra toutes les fonctionnalités de l'application.

3.1. MATERIELS ET LOGICIELS UTILISES

3.1.1. Matériels utilisés

Pour la réalisation et la mise en œuvre de notre solution, nous avons utilisé un ordinateur portable dont les caractéristiques techniques sont renseignées dans le tableau ci-dessous.

Tableau 4 : Caractéristiques du matériel utilisé

| Éléments | Caractéristiques |
|-------------------------|--|
| Marque | Hewlett-Packard (HP) |
| Modèle | Victus by HP gaming Laptop 15-fa 100 |
| Processeur et fréquence | 13th Gen Intel(R) Core(TM) i5-13420H, 5000 MHz, 8 cœur(s), 12 processeur(s) logique(s) |
| Type de système | Système d'exploitation 64 bits, processeur x64 |
| Mémoire RAM | 16,00 Go |
| Disque dur | 512,00 Go |
| Système d'exploitation | Microsoft Windows 11 Famille |
| Chipset graphique | Nvidia GEFORCE RTX |

3.1.2. Logiciels utilisés



Logiciels et environnement de travail

- Visual Studio Code :

Est un éditeur de code source léger et puissant, développé par Microsoft, il est utilisé pour son support étendu des extensions, facilitant le



Figure 25 : Logo Visual Studio Code

développement en PHP, HTML, CSS, et JavaScript. Son interface intuitive et ses fonctionnalités de débogage intégrées en font un choix idéal pour un projet nécessitant un environnement de développement flexible et personnalisable.

- XAMPP :

Est une solution tout-en-un qui regroupe Apache, MySQL, et PHP, utilisée pour créer un environnement de serveur local. XAMPP simplifie le processus de développement en offrant une plateforme stable pour tester et déployer des applications web sans la nécessité de configurer un serveur distinct.



Figure 26 : Logo XAMPP

- MySQL :

Un système de gestion de bases de données relationnelles performant, choisi pour sa robustesse et sa compatibilité avec Laravel. MySQL est essentiel pour gérer les données critiques du projet, assurant des transactions sécurisées et une gestion efficace des données volumineuses.



Figure 27 : Logo MySQL



Langages et technologies

- PHP :

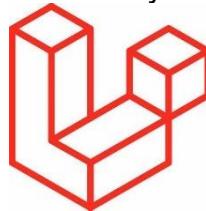
Est un langage de script côté serveur, essentiel pour le développement dynamique des fonctionnalités de l'application. PHP a été choisi pour sa flexibilité et sa compatibilité étroite avec le Framework Laravel.



[Figure 28 : Logo PHP](#)

- Laravel :

Est un Framework PHP moderne utilisé pour sa structure MVC, qui facilite le développement rapide et structuré d'applications web complexes. Laravel 11 a été sélectionné pour ses fonctionnalités avancées, telles que l'authentification, les migrations de bases de données, et l'intégration avec MySQL.



[Figure 29 : Logo Laravel](#)

- HTML & CSS :

Les langages de base pour la création de la structure et du style des pages web. HTML et CSS sont utilisés pour construire l'interface utilisateur, en assurant une présentation propre et réactive.



[Figure 30 : Logo HTML & CSS](#)

- JavaScript :

Est un langage de programmation essentiel pour ajouter de l'interactivité et des fonctionnalités dynamiques à l'application. JavaScript est utilisé en conjonction avec jQuery pour manipuler le DOM, gérer les événements, et améliorer l'expérience utilisateur.



[Figure 31 : Logo JavaScript](#)

- JQuery :

Est une bibliothèque JavaScript légère utilisée pour simplifier les manipulations DOM, la gestion des événements, et les animations. JQuery est utilisé pour optimiser les interactions utilisateur et rendre l'application plus dynamique.



[Figure 32 : Logo JQuery](#)

- GitHub :

C'est Un système de contrôle de version distribué, utilisé pour suivre les modifications du code source tout au long du développement. GitHub est essentiel pour la gestion collaborative du code et pour assurer l'intégrité du projet à chaque étape.



[Figure 33 : Logo GitHub](#)

- Apache :

C'est un serveur HTTP utilisé dans l'environnement XAMPP pour servir les pages web de l'application. Apache a été choisi pour sa stabilité, sa



Figure 34 : Logo Apache

flexibilité, et sa capacité à gérer des applications web à grande échelle.

3.2. ARCHITECTURE MATERIELLE ET LOGICIELLE DE L'APPLICATION

3.2.1. Architecture matérielle

Pour la mise en œuvre de notre application de gestion du parc automobile, nous avons opté pour une architecture client/serveur, précisément une architecture à trois niveaux, communément appelée architecture 3-tiers. Cette architecture repose sur trois principaux composants interdépendants :



La présentation : Ce niveau gère l'interaction directe avec les utilisateurs. Il se traduit par les interfaces utilisateur à travers lesquelles les acteurs du système accèdent aux différentes fonctionnalités de l'application. La présentation est assurée par le client, c'est-à-dire la machine que l'utilisateur utilise pour interagir avec l'application.



Les traitements : Il s'agit du cœur logique de l'application, là où sont exécutés les traitements des données et la logique métier. Ce niveau est pris en charge par un serveur intermédiaire. Dans notre cas, c'est le serveur Apache qui assure la gestion des traitements, aussi bien locaux que globaux, en reliant directement la couche présentation via le protocole http.



Les données : Ce niveau concerne la gestion centralisée des données, comprenant tous les mécanismes permettant le stockage, la manipulation et l'accès aux informations traitées par l'application. Ces données sont hébergées et gérées par un serveur de bases de données. Pour notre projet, nous utilisons un serveur équipé de MySQL, garantissant ainsi une gestion efficace et sécurisée des données.

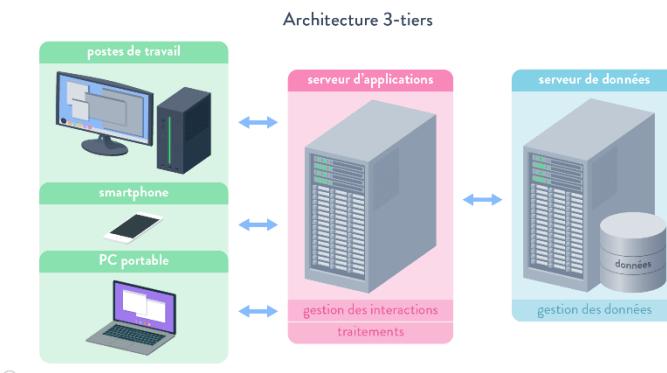


Figure 35 : Architecture matérielle 3-tiers

3.2.2. Architecture logicielle

Pour structurer efficacement le code de notre application, nous avons adopté le modèle de conception MVC (Modèle-Vue-Contrôleur). Ce modèle permet une séparation claire des préoccupations en décomposant l'application en trois parties :



Modèle : Cette couche encapsule la logique métier ainsi que l'accès aux données. Elle gère les règles de gestion et les interactions avec la base de données. Dans notre application, le modèle est construit en suivant les principes de l'orientation objet, ce qui facilite la maintenance et l'évolution du système.



Vue : La vue est responsable de tout ce qui concerne l'affichage et l'interface utilisateur. Elle gère la présentation des données et la collecte des saisies utilisateur, assurant ainsi une expérience utilisateur fluide et intuitive.



Contrôleur : Le contrôleur agit comme un intermédiaire entre le modèle et la vue. Il orchestre les interactions entre l'utilisateur et l'application, en traitant les entrées de l'utilisateur, en appelant les opérations appropriées du modèle, et en mettant à jour la vue en conséquence.

Cette architecture logicielle MVC permet une gestion modulaire du code, facilitant les tests, la maintenance, et les futures évolutions de l'application.

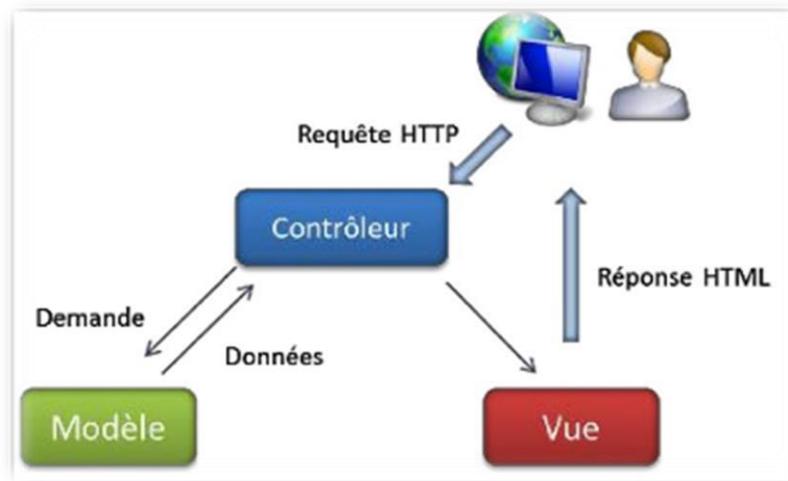


Figure 36 : Architecture logicielle MVC

3.3. SECURITE DE L'APPLICATION

La sécurité de notre application a été intégrée de manière rigoureuse et méthodique à plusieurs niveaux afin de garantir la protection des données et la confidentialité des utilisateurs. Nous avons mis en place des mécanismes robustes pour contrôler les accès et assurer une sécurité optimale.



Authentification : L'authentification constitue la première ligne de défense en vérifiant l'identité des utilisateurs. Cette étape est cruciale pour s'assurer que chaque utilisateur accède uniquement aux fonctionnalités qui lui sont réservées. L'authentification est réalisée à travers des couples email/mot de passe, garantissant ainsi que seuls les utilisateurs légitimes puissent se connecter à l'application.



Autorisation : Une fois l'utilisateur authentifié, l'autorisation détermine ses droits d'accès au sein de l'application. Ces droits sont attribués en fonction des rôles ou profils des utilisateurs, permettant de restreindre ou de permettre l'accès à certaines fonctionnalités ou données en fonction des responsabilités de chaque utilisateur. Ce système de gestion des droits, couplé au système de Guard (Multi-Authentification) de Laravel, renforce la sécurité en s'assurant que chaque utilisateur n'accède qu'aux ressources pertinentes à son profil.



Cryptographie : La cryptographie joue un rôle clé dans la protection des données sensibles. Côté client, elle est utilisée pour masquer les mots de passe lors de la saisie, empêchant ainsi toute interception par des regards indiscrets. Côté serveur, les mots de passe sont chiffrés à l'aide de l'algorithme de hachage Bcrypt, disponible dans PHP 8, garantissant que les mots de passe stockés ne puissent pas être compromis. Ce double niveau de protection assure une sécurité renforcée des informations utilisateur tout au long du processus de traitement des données.

3.4. MISE EN PLACE DE LA BASE DE DONNEES

Pour la création de notre base de données, nous avons utilisé l'ORM Eloquent (Object-Relational Mapping est une technique de programmation qui permet de manipuler les données d'une base de données relationnelle en utilisant des objets du langage de programmation. L'ORM facilite l'interaction entre le code et la base de données en permettant aux développeurs de travailler avec des objets plutôt qu'avec des tables et des requêtes SQL.) ; intégré dans le Framework Laravel. Eloquent permet de gérer les interactions avec la base de données de manière fluide et intuitive, en se basant sur des modèles PHP qui représentent les tables de la base de données.

L'un des grands avantages d'Eloquent est la gestion des migrations. Les migrations sont des fichiers PHP qui définissent la structure de la base de données, incluant la création, la modification ou la suppression des tables et des colonnes. Elles permettent de synchroniser facilement la structure de la base de données avec les évolutions du code, sans avoir à écrire manuellement des scripts SQL. Voici quelques fichiers de migrations pour les principales tables de notre système.



Migration de la table « véhicules »

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('vehicules', function (Blueprint $table) {
15             $table->id();
16             $table->string('immatriculation')->unique();
17             $table->string('numero_moteur')->unique();
18             $table->enum('type_moteur', ['essence', 'diesel', '100% électrique']);
19             $table->string('numero_chassis')->unique();
20             $table->integer('nombre_place');
21             $table->date('date_achat');
22             $table->bigInteger('numero_carte_grise')->unique();
23             $table->string('image_carte_grise')->unique();
24             $table->date('validite_garantie')->nullable();
25             $table->integer('kilometrage');
26             $table->enum('etat_vehicule', ['neuf', 'bon etat', 'indisponible', 'en maintenance',
27                 'etat passable', 'mauvais etat', 'rebut'])->default('neuf');
28             $table->foreignId('id_genre_vehicule')->constrained(table: 'genre_vehicules');
29             $table->foreignId('id_fournisseur')->constrained(table: 'fournisseurs');
30             $table->foreignId('id_marque')->constrained(table: 'marques');
31             $table->foreignId('id_modele')->constrained(table: 'modeles');
32             $table->timestamps();
33
34             $table->softDeletes();
35             $table->unsignedBigInteger('created_by')->nullable();
36             $table->unsignedBigInteger('updated_by')->nullable();
37         });
38     }
39
40     /**
41      * Reverse the migrations.
42      */
43     public function down(): void
44     {
45         Schema::dropIfExists('vehicules');
46     }
47 };
48

```

Figure 37 : Migration <>vehicules>>



Migration de la table « affectations »

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      */
12      public function up(): void
13  {
14          Schema::create('affectations', function (Blueprint $table) {
15              $table->id();
16              $table->string('numero_affectation')->unique();
17              $table->date('date_affectation');
18              $table->integer('kilometrage');
19              $table->string('liste_outillage');
20              $table->string('fonction');
21              $table->foreignId('id_departement')->constrained(table :'departements');
22              $table->foreignId('id_vehicule')->constrained(table :'vehicules');
23              $table->timestamps();
24
25              $table->softDeletes();
26              $table->unsignedBigInteger('created_by')->nullable();
27              $table->unsignedBigInteger('updated_by')->nullable();
28          });
29      }
30
31      /**
32      * Reverse the migrations.
33      */
34      public function down(): void
35  {
36          Schema::dropIfExists('affectations');
37      }
38  };
39

```

[Figure 38 : Migration <>affectations>>](#)



Migration de la table « fiche_sorties »

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('fiche_sorties', function (Blueprint $table) {
15             $table->id();
16             $table->string('numero_sortie');
17             $table->dateTime('date_heure_depart');
18             $table->dateTime('date_heure_retour')->nullable();
19             $table->integer('kilometrage_depart');
20             $table->integer('kilometrage_retour')->nullable();
21             $table->enum('etat_depart', ['bon etat', 'etat passable', 'mauvais etat']);
22             $table->enum('etat_retour', ['bon etat', 'etat passable', 'mauvais etat'])->nullable();
23             $table->double('estimation_besoin_carburant');
24             $table->integer('estimation_nombre_kilometre');
25             $table->string('observation_depart');
26             $table->string('observation_retour')->nullable();
27             $table->double('quantite_carburant_consomme')->nullable();
28             $table->enum('etat_fiche', ['en cours', 'terminée']);
29             $table->foreignId('id_chauffeur')->constrained(table: 'chauffeurs');
30             $table->foreignId('id_vehicule')->constrained(table: 'vehicules');
31             $table->foreignId('id_demande')->constrained(table: 'demandes');
32             $table->foreignId('id_approvisionnement')->nullable()->constrained(table: 'approvisionnements');
33             $table->timestamps();
34
35             $table->softDeletes();
36             $table->unsignedBigInteger('created_by')->nullable();
37             $table->unsignedBigInteger('updated_by')->nullable();
38         });
39     }
40
41     /**
42      * Reverse the migrations.
43      */
44     public function down(): void
45     {
46         Schema::dropIfExists('fiche_sorties');
47     }
48 };
49

```

Figure 39 : Migration <>fiche_sorties>>



Migration de la table « demandes »

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      */
12      public function up(): void
13  {
14          Schema::create('demandes', function (Blueprint $table) {
15              $table->id();
16              $table->string('numero_demande')->unique();
17              $table->date('date_demande');
18              $table->string('nom_demandeur');
19              $table->string('objet_demande');
20              $table->enum('status',[ 'validée', 'non validée'])->default('non validée');
21              $table->integer('contact_demandeur');
22              $table->dateTime('date_heure_depart');
23              $table->dateTime('date_heure_retour');
24              $table->string('adresse_destination');
25              $table->foreignId('id_departement')->constrained(table: 'departements');
26              $table->foreignId('id_type_sortie')->constrained(table: 'type_sorties');
27              $table->timestamps();
28
29              $table->softDeletes();
30              $table->unsignedBigInteger('created_by')->nullable();
31              $table->unsignedBigInteger('updated_by')->nullable();
32          });
33      }
34
35      /**
36      * Reverse the migrations.
37      */
38      public function down(): void
39  {
40          Schema::dropIfExists('demandes');
41      }
42  };
43

```

Figure 40 : Migration <>demandes>>



Migration de la table « fiche_maintenances »

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('fiche_maintenances', function (Blueprint $table) {
15             $table->id();
16             $table->string('numero_maintenance')->unique();
17             $table->date('date_heure_entree');
18             $table->date('date_heure_sortie');
19             $table->integer('kilometrage');
20             $table->string('declaration_utilisateur');
21             $table->string('observation_controleur');
22             $table->string('inspection_reception');
23             $table->string('inspection_livraison');
24             $table->string('travaux_execute');
25             $table->enum('etat_fiche', ['en cours', 'termine']);
26             $table->foreignId('id_chauffeur')->constrained(table : 'chauffeurs');
27             $table->foreignId('id_vehicule')->constrained(table : 'vehicules');
28             $table->foreignId('id_mecanicien')->constrained(table : 'mecaniciens');
29             $table->foreignId('id_garage')->constrained(table : 'garages');
30             $table->timestamps();
31
32             $table->softDeletes();
33             $table->unsignedBigInteger('created_by')->nullable();
34             $table->unsignedBigInteger('updated_by')->nullable();
35         });
36     }
37
38     /**
39      * Reverse the migrations.
40      */
41     public function down(): void
42     {
43         Schema::dropIfExists('fiche_maintenances');
44     }
45 };
46

```

Figure 41 : Migration <<fiche_maintenances>>

Application de gestion du parc automobile de la CEET



Migration de la table « approvisionnements »

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('approvisionnements', function (Blueprint $table) {
15             $table->id();
16             $table->string('numero_appvnmt')->unique();
17             $table->enum('motif_appvnmt', ['mission', 'feuille_de_route', 'reapprovisionnement']);
18             $table->enum('type_produit', ['essence', 'gasoil']);
19             $table->double('quantite');
20             $table->double('taux_consommation');
21             $table->enum('etat_fiche', ['en_cours', 'termine']);
22             $table->foreignId('id_chauffeur')->constrained(table : 'chauffeurs');
23             $table->foreignId('id_vehicule')->constrained(table : 'vehicules');
24             $table->timestamps();
25
26             $table->softDeletes();
27             $table->unsignedBigInteger('created_by')->nullable();
28             $table->unsignedBigInteger('updated_by')->nullable();
29         });
30     }
31
32     /**
33     * Reverse the migrations.
34     */
35     public function down(): void
36     {
37         Schema::dropIfExists('approvisionnements');
38     }
39 };
40

```

Figure 42 : Migration <<approvisionnements>>

Application de gestion du parc automobile de la CEET



Migration de la table « users »

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      */
12      public function up(): void
13  {
14          Schema::create('users', function (Blueprint $table) {
15              $table->id();
16              $table->string('matricule')->unique();
17              $table->string('nom');
18              $table->string('prenom');
19              $table->string('email')->unique();
20              $table->string('contact')->unique()->nullable();
21              $table->timestamp('email_verified_at')->nullable();
22              $table->string('password');
23              $table->foreignId('id_departement')->constrained(table: 'departements');
24              $table->foreignId('id_profil')->constrained(table: 'profils');
25              $table->timestampls();
26
27              $table->softDeletes();
28              $table->unsignedBigInteger('created_by')->nullable();
29              $table->unsignedBigInteger('updated_by')->nullable();
30          });
31
32          Schema::create('password_reset_tokens', function (Blueprint $table) {
33              $table->string('email')->primary();
34              $table->string('token');
35              $table->timestamp('created_at')->nullable();
36              $table->softDeletes();
37              $table->unsignedBigInteger('created_by')->nullable();
38              $table->unsignedBigInteger('updated_by')->nullable();
39          });
30
41          Schema::create('sessions', function (Blueprint $table) {
42              $table->string('id')->primary();
43              $table->foreignId('user_id')->nullable()->index();
44              $table->string('ip_address', 45)->nullable();
45              $table->text('user_agent')->nullable();
46              $table->longText('payload');
47              $table->integer('last_activity')->index();
48              $table->softDeletes();
49              $table->unsignedBigInteger('created_by')->nullable();
50              $table->unsignedBigInteger('updated_by')->nullable();
51          });
52      }
53
54      /**
55      * Reverse the migrations.
56      */
57      public function down(): void
58  {
59          Schema::dropIfExists('users');
60          Schema::dropIfExists('password_reset_tokens');
61          Schema::dropIfExists('sessions');
62      }
63  };

```

Figure 43 : Migration <>users>>



Migration de la table « pieces »

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('pieces', function (Blueprint $table) {
15             $table->id();
16             $table->string('reference')->nullable(false);
17             $table->string('nom');
18             $table->double('quantite');
19             $table->string('observation')->nullable(true);
20             $table->foreignId('id_type_piece')->constrained(table : 'type_pieces');
21             $table->foreignId('id_fournisseur')->constrained(table : 'fournisseurs');
22             $table->timestamps();
23
24             $table->softDeletes();
25             $table->unsignedBigInteger('created_by')->nullable();
26             $table->unsignedBigInteger('updated_by')->nullable();
27         });
28     }
29
30     /**
31     * Reverse the migrations.
32     */
33     public function down(): void
34     {
35         Schema::dropIfExists('pieces');
36     }
37 };
38

```

[Figure 44 : Migration <>pieces>>](#)

4. Partie IV : Guide d'exploitation

Application de gestion du parc automobile de la CEET

La phase de réalisation et de mise en œuvre, détaillée dans la partie précédente de ce document, nous a permis de concrétiser le système de gestion du parc automobile de la CEET, tel qu'il avait été conçu initialement. À présent, il est primordial de procéder au déploiement de ce système pour le rendre pleinement opérationnel. Cette section vise à détailler les étapes de déploiement, les configurations matérielles et logicielles requises, ainsi que les procédures de suivi et de maintenance qui garantiront le bon fonctionnement et la pérennité du système. Cette partie sera structurée en trois sous-sections : la première présentera les configurations matérielles et logicielles nécessaires au déploiement du système, la deuxième détaillera le processus de déploiement et les méthodes de suivi mises en place, et la troisième abordera les aspects liés à la maintenance pour résoudre les éventuels problèmes rencontrés après la mise en production.

4.1. CONFIGURATION LOGICIELLE ET MATÉRIELLE

4.1.1. Configuration logicielle

Pour assurer le bon fonctionnement de notre application, plusieurs composants logiciels sont nécessaires. Voici les configurations logicielles recommandées :

Tableau 5 : Configuration logicielle de notre application

| Couches logicielles | Configurations recommandées |
|-----------------------------------|---|
| Poste utilisateur | <ul style="list-style-type: none"> • Navigateur Web : Google Chrome, Microsoft Edge • Système d'exploitation (OS) : Windows 10, 11 |
| Serveur d'application | <ul style="list-style-type: none"> • Composer pour la gestion des dépendances • PHP 8 pour le langage serveur • Apache pour le serveur web • Extensions PHP : PDO, open_ssl, mbstring, gd, pdo_pgsql, pgsql |
| Serveur de base de données | <ul style="list-style-type: none"> • SGBD : MySQL pour la gestion des données |

4.1.2. Configuration matérielle

Les exigences matérielles dépendent des rôles spécifiques des machines dans l'infrastructure de l'application.

Application de gestion du parc automobile de la CEET

| Couches matérielles | Configurations minimales | Configurations recommandées |
|-----------------------------------|---|--|
| Poste utilisateur | <ul style="list-style-type: none"> ❖ Processeur : 1.8 GHz, 64 bits ❖ Disque dur : 128 Go ❖ RAM : 2 Go ❖ Port RJ45 : Actif | <ul style="list-style-type: none"> ❖ Processeur : ≥ 2.2 GHz, 64 bits ❖ Disque dur : 256 Go ❖ RAM : 4 Go ❖ Port RJ45 : Actif |
| Serveur d'application | <ul style="list-style-type: none"> ❖ Processeur : Core i5, 2.0 GHz ❖ Disque dur : 500 Go ❖ RAM : 4 Go | <ul style="list-style-type: none"> ❖ Processeur : Core i7, 3.0 GHz ❖ Disque dur : 1 To ❖ RAM : 8 Go |
| Serveur de base de données | <ul style="list-style-type: none"> ❖ Processeur : Core i5, 2.5 GHz ❖ Disque Dur : 2 To ❖ Mémoire RAM : 8 Go ❖ Port RJ45 : Actif | <ul style="list-style-type: none"> ❖ Processeur : Core i7, 3.6 GHz ❖ Disque Dur : 4 To ❖ Mémoire RAM : 16 Go ❖ Port RJ45 : Actif |

*Tableau 6 : Configurations matérielles pour notre application***4.2. DÉPLOIEMENT ET SUIVI****4.2.1. Déploiement de l'application**

Le déploiement de l'application consiste à mettre en place l'application auprès des postes utilisateurs afin de rendre le système opérationnel. Les étapes suivantes décrivent le processus de déploiement de l'application de gestion du parc automobile de la CEET :

Création de la base de données :

- On se connecte au système de gestion de base de données MySQL.

Application de gestion du parc automobile de la CEET

- Créez une nouvelle base de données en utilisant le panneau de configuration MySQL.

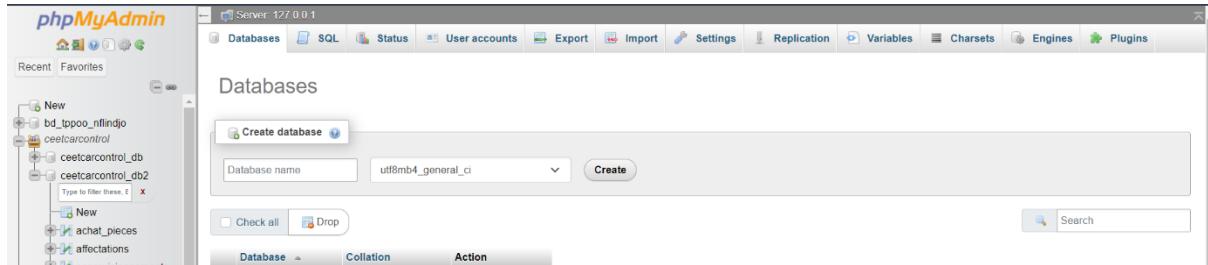


Figure 45 : Page de création d'une nouvelle base de données



Installation du projet Laravel :

- Exécutez la commande suivante pour installer Laravel sur le serveur :
- ```
composer create-project --prefer-dist laravel/laravel gestion_parc_auto
```
- Exécution des migrations : Après avoir configuré la base de données, exécutez les migrations pour créer les tables :

```
php artisan migrate
```

- Mise à jour des dépendances :

```
composer update
```



### Configuration du fichier « .env » :

- Modification du fichier .env situé à la racine de votre projet Laravel pour y insérer les paramètres suivants :

**APP\_ENV** : indique l'environnement (*production, développement*).

**APP\_DEBUG** : désactivez-le (*false*) pour la production afin d'éviter que les utilisateurs voient des messages d'erreur détaillés.

**APP\_URL** : l'URL de l'application.

**DB\_DATABASE** : le nom de la base de données.

**DB\_USERNAME** : le nom d'utilisateur de la base de données.

**DB\_PASSWORD** : le mot de passe de la base de données.

Ces étapes garantiront que notre application est correctement déployée et prête à être utilisée.

## Application de gestion du parc automobile de la CEET

### 4.2.2. Suivie de l'application

Une fois l'application déployée, il est essentiel d'assurer un suivi régulier pour éviter les problèmes qui pourraient entraîner une interruption de service ou une perte de données. Voici les mesures de suivi recommandées :



#### Sauvegarde :

Nous effectuons des sauvegardes régulières de la base de données pour garantir la récupération des données en cas de défaillance.

La sauvegarde peut être logique (réalisée à chaud sans arrêter la base de données) et effectuée via des outils comme phpMyAdmin ou physique.



#### Restauration :

Nous nous assurons de pouvoir restaurer la base de données à partir des sauvegardes en cas de besoin. La restauration doit être effectuée lorsque le serveur est en mode mono-utilisateur pour éviter toute perturbation.

### 4.3. MAINTENANCE

La maintenance de l'application est essentielle pour corriger les erreurs, mettre à jour le système et assurer la sécurité. Voici quelques bonnes pratiques mises en place pour la maintenance de notre système :

- **Surveillance des performances :**

- Utiliser des outils de monitoring pour surveiller les performances du serveur et identifier les goulets d'étranglement potentiels.

- **Gestion des incidents :**

- Mettre en place un plan de gestion des incidents pour réagir rapidement aux problèmes qui pourraient survenir. Pour cela, nous définissons ici les actions à mener en cas de certaines erreurs. Ces erreurs sont scindées en deux parties : les erreurs courantes et les autres erreurs.

## Application de gestion du parc automobile de la CEET

*Tableau 7 : Actions à mener en cas d'erreurs*

| <b>Code de l'erreur</b> | <b>Nom de l'erreur</b> | <b>Description</b>          | <b>Action à mener</b>                           |
|-------------------------|------------------------|-----------------------------|-------------------------------------------------|
| <b>404</b>              | Internal server error  | Ressource introuvable       | Vérifier si l'URL est valide.                   |
| <b>408</b>              | Request Time-out       | Temps de requête dépassé    | Actualiser la page et reprendre l'action.       |
| <b>500</b>              | Internal server error  | Le serveur est inaccessible | Actualisez la page et contacter la maintenance. |
| <b>Autres erreurs</b>   |                        | Erreurs non mentionnées     | Actualisez la page et contacter la maintenance. |

En suivant ces directives, on assure une gestion efficace de la maintenance de notre application, garantissant ainsi sa pérennité et son bon fonctionnement.



## **5. Partie V : Guide d'utilisation**

## Application de gestion du parc automobile de la CEET

Cette section présente en détail les fonctionnalités et les rôles au sein de l'application **CarControl**, conçue pour optimiser la gestion du parc automobile de la Compagnie Énergie Électrique du Togo (CEET). L'objectif est de centraliser toutes les opérations liées aux véhicules, de l'acquisition à l'exploitation, en passant par la maintenance et le suivi des consommations. Les différents espaces de travail sont organisés de manière à assurer une gestion efficace et une traçabilité complète des opérations.

### 5.1. DESCRIPTION TEXTUELLE DU LOGICIEL

CarControl est une application web dédiée à la gestion optimisée du parc automobile de la Compagnie Énergie Électrique du Togo (CEET). Cette application permet de centraliser et de suivre l'ensemble des opérations liées à la gestion des véhicules, de l'acquisition à l'exploitation, en passant par la maintenance et le suivi des consommations.

L'application est organisée autour de quatre espaces de travail principaux :

- ➡ **Administrateur et le Chef Service** : Ce rôle est chargé de la gestion globale du système, incluant la création et la gestion des utilisateurs, l'attribution des droits d'accès, et la supervision des opérations effectuées au sein de l'application. L'administrateur a également la responsabilité de configurer les paramètres de l'application, tels que les types de véhicules, les types de carburant, et les catégories d'opérations.
- ➡ **Responsable de la division exploitation** : Ce rôle est central dans la gestion opérationnelle du parc automobile. Il est responsable de l'affectation des véhicules, de la création des fiches de sortie, du suivi des déplacements et de l'exploitation des véhicules. Il gère également l'inventaire des véhicules et assure la traçabilité de chaque mission ou course.
- ➡ **Responsable de la division maintenance** : Chargé de la maintenance préventive et corrective des véhicules, ce rôle gère les fiches de maintenance, organise les réparations, suit les interventions techniques, et supervise la gestion des pièces détachées. Il veille également à la planification des visites techniques et au respect des échéances pour les révisions périodiques.
- ➡ **Gestionnaire des consommations** : Ce rôle est spécialisé dans le suivi et le contrôle des consommations de carburant. Il gère les cartes de carburant, les bons de sortie de produits pétroliers, et assure la validation des

## Application de gestion du parc automobile de la CEET

approvisionnements en carburant. Il analyse également les taux de consommation des véhicules et produit des rapports pour optimiser les coûts liés à l'exploitation.

Notre application CarControl offre une vision complète et intégrée de la gestion du parc automobile, facilitant la coordination entre les différentes divisions, assurant une traçabilité des opérations, et contribuant à une meilleure gestion des ressources de la CEET.

## **5.2. PLAN DE NAVIGATION**

 Administrateur et chef service :

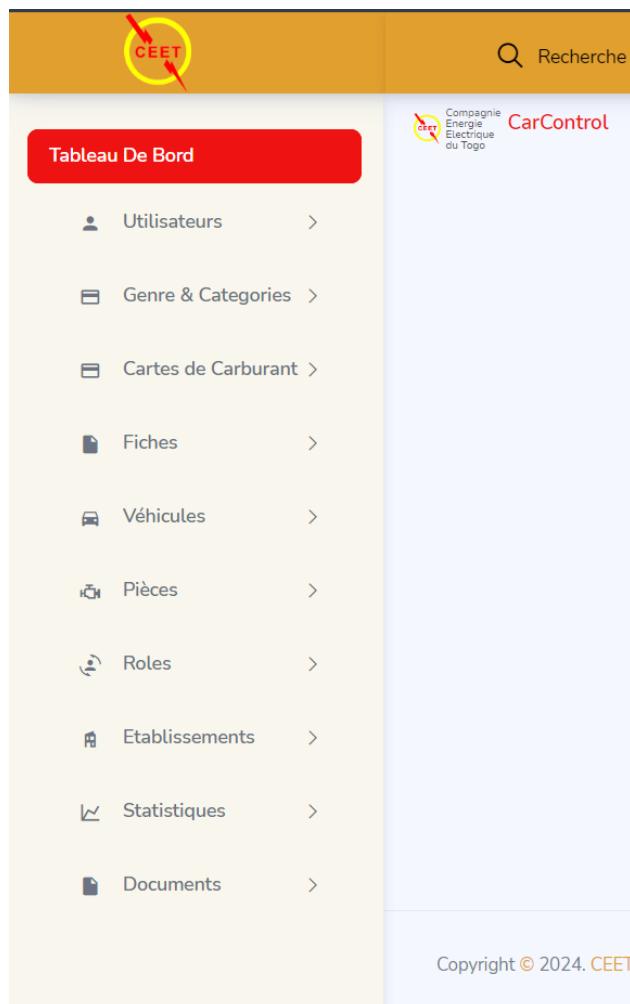


Figure 46 : Tableau de bord administrateur



Figure 47 : Interface de gestion du parc pour l'administrateur et chef service

## Application de gestion du parc automobile de la CEET

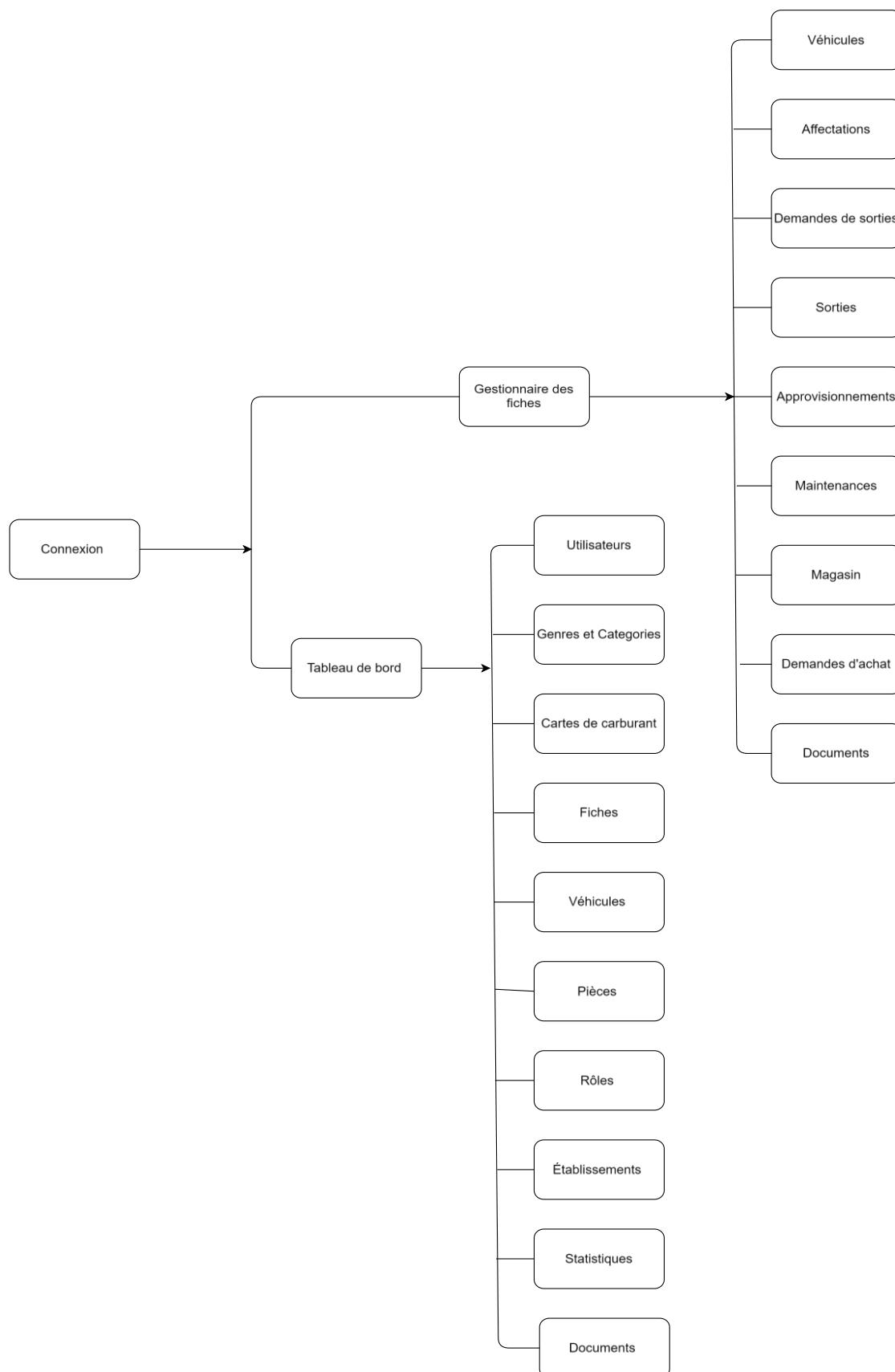


Figure 48 : Plan de navigation administrateur et chef service

## Application de gestion du parc automobile de la CEET

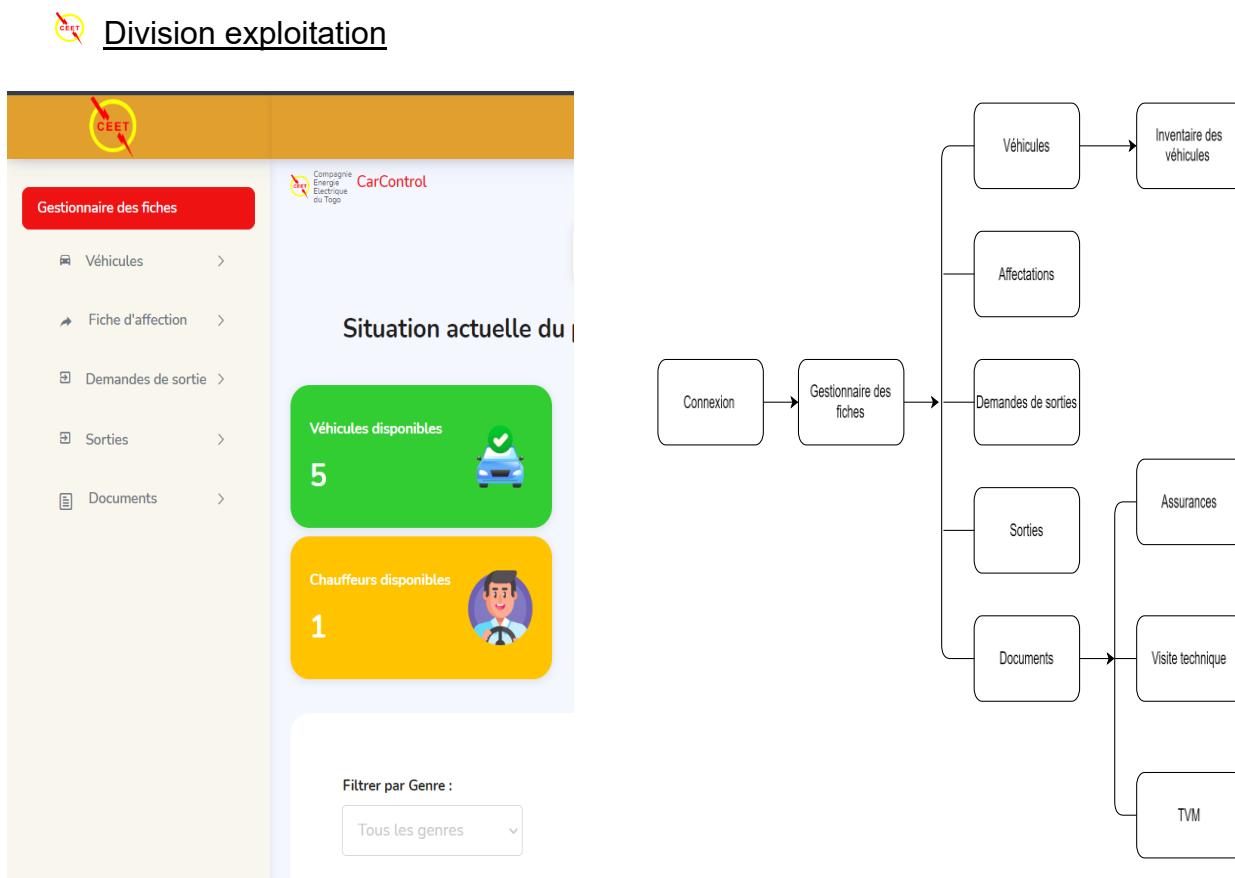


Figure 49 : Interface et plan de navigation exploitation

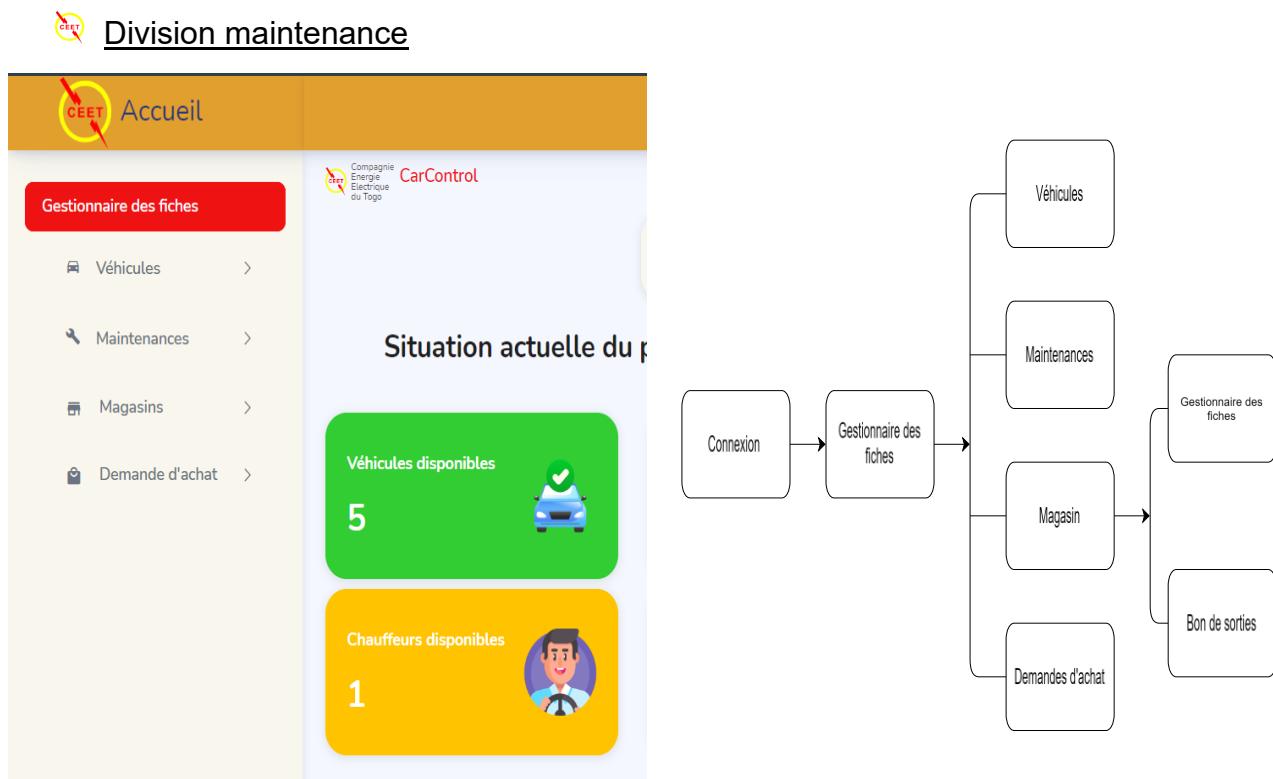
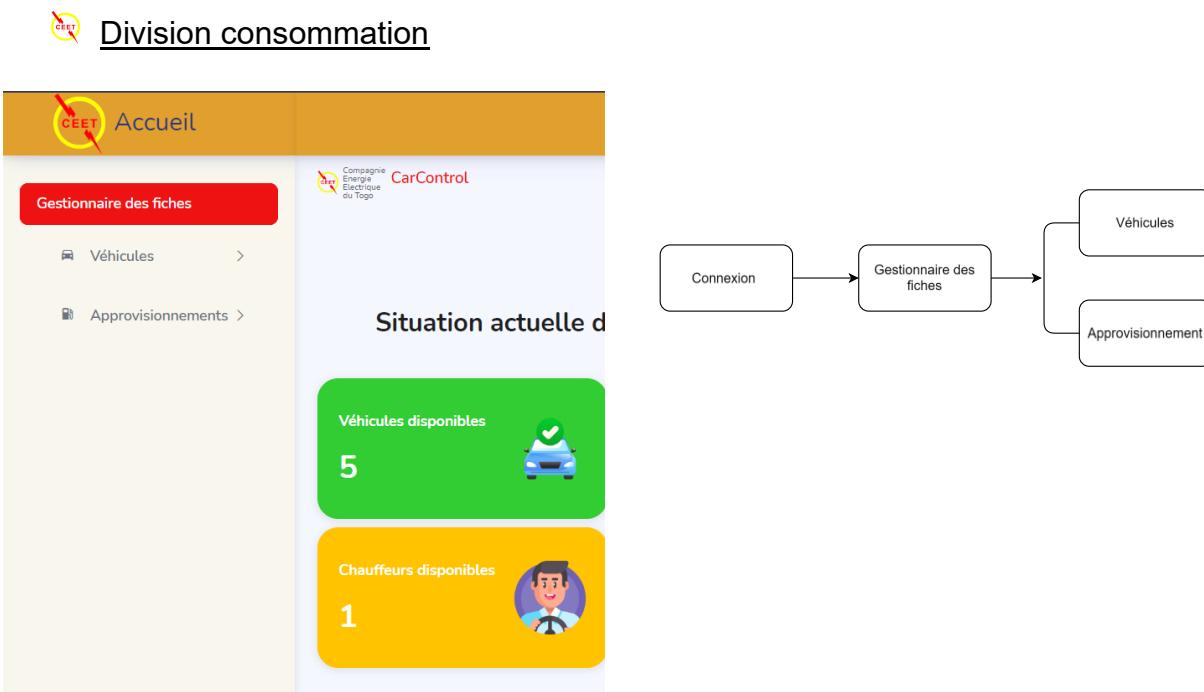


Figure 50 : Interface et plan de navigation maintenance

## Application de gestion du parc automobile de la CEET

*Figure 51 : Interface et plan de navigation consommation*

### 5.3. PRÉSENTATION DES DIFFÉRENTES INTERFACES DE L'APPLICATION

Cette section du document est destinée à guider l'utilisateur à travers les principales fonctionnalités de l'application **CarControl**, dédiée à la gestion du parc automobile de la CEET. À travers une description textuelle claire et concise, nous explorerons les diverses interfaces de l'application, en illustrant les processus et interactions que l'utilisateur pourra effectuer au quotidien. Cette partie se veut pratique et accessible, permettant à chaque utilisateur de se familiariser avec les outils mis à sa disposition.



#### Authentification :

L'accès à l'application CarControl commence par une phase d'authentification. Cette étape est cruciale pour sécuriser l'accès aux données sensibles et garantir que seuls les utilisateurs autorisés puissent interagir avec le système.

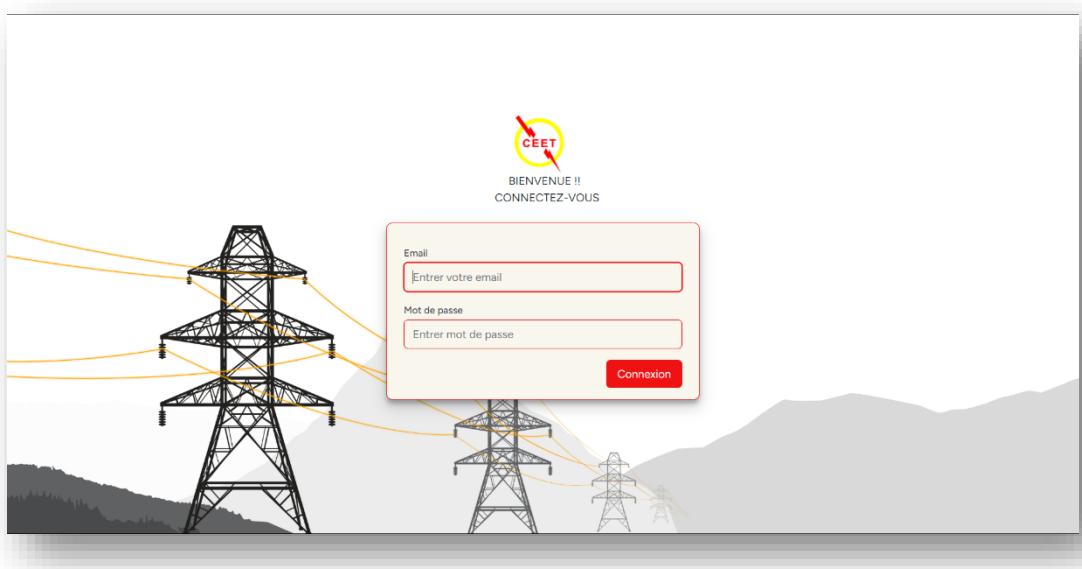
Pour se connecter à l'application, l'utilisateur doit suivre les étapes suivantes :

- **Email** : Saisir votre email d'utilisateur. Ce champ est sensible à la casse, ce qui signifie qu'il faut respecter les majuscules et minuscules.
- **Mot de passe** : Saisir votre mot de passe.

## Application de gestion du parc automobile de la CEET

- Connexion** : Cliquer sur le bouton « Connexion » pour valider les informations saisies.

Si tous les champs sont correctement remplis et les informations sont valides, l'utilisateur est redirigé vers la page d'accueil correspondant à son rôle (Administrateur, Responsable de l'exploitation, etc.). En cas d'erreur, un message d'alerte apparaîtra, indiquant que les informations sont incorrectes.



*Figure 52 : Page d'authentification*

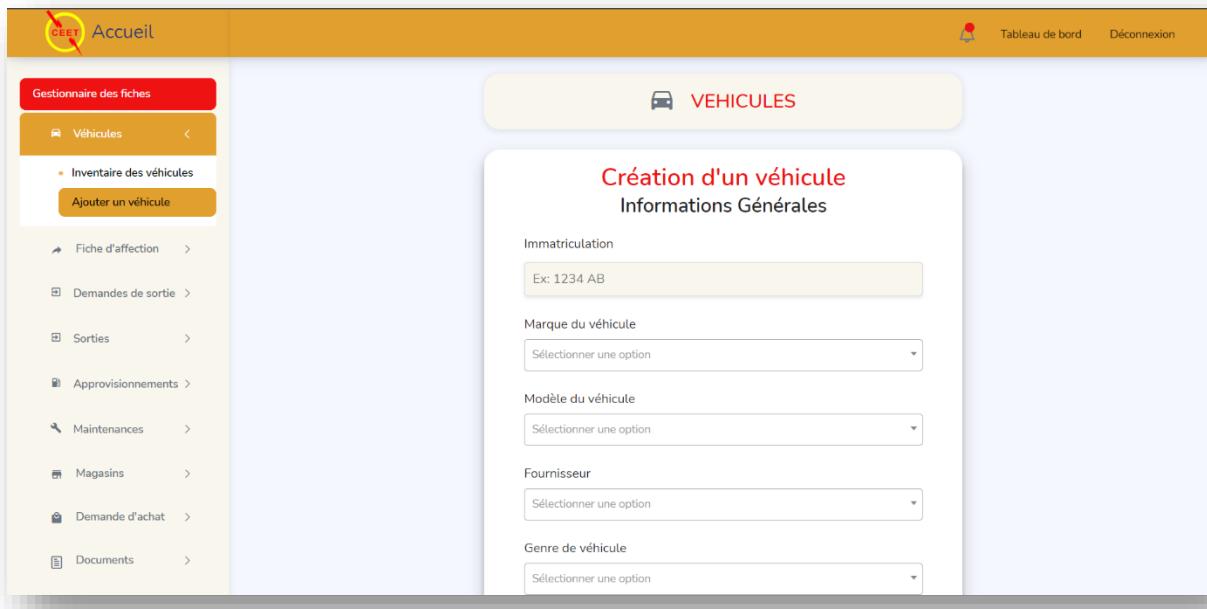
### Gestion des véhicules

Cette fonctionnalité permet au responsable de l'exploitation de gérer l'acquisition et l'affectation des véhicules aux différentes directions ou départements de la CEET.

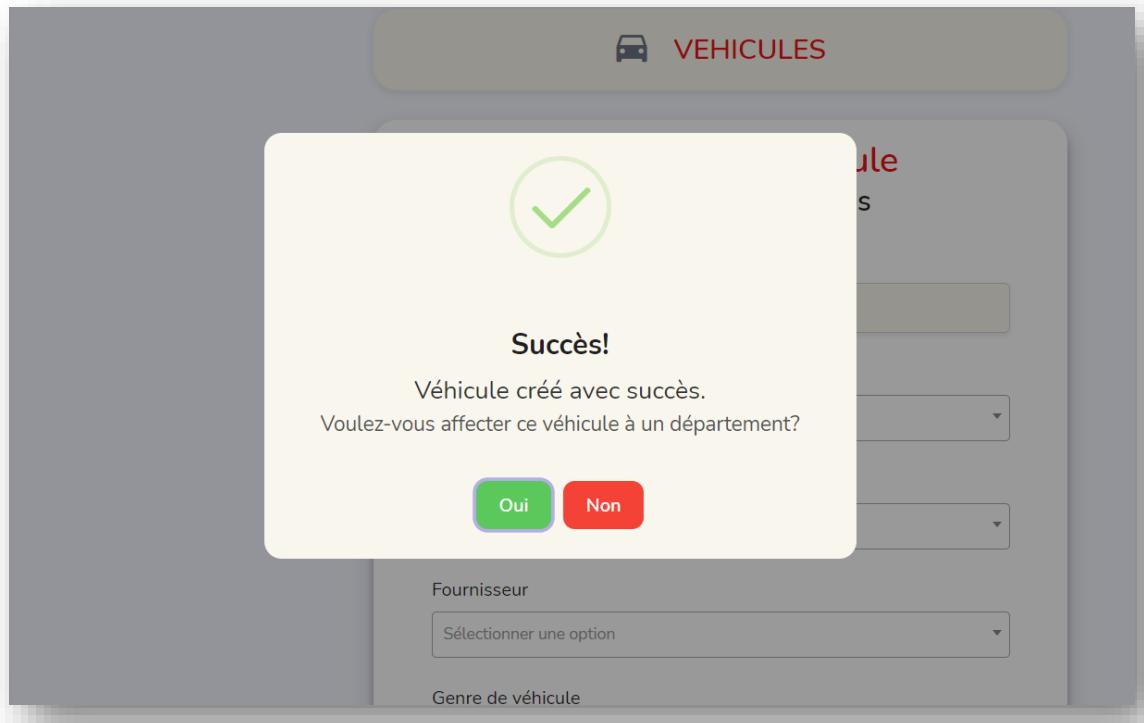
Pour enregistrer un véhicule :

- Formulaire d'ajout d'un véhicule** : Renseigner les informations du véhicule, les informations de la carte grise.

## Application de gestion du parc automobile de la CEET

*Figure 53 : Interface d'ajout d'un véhicule*

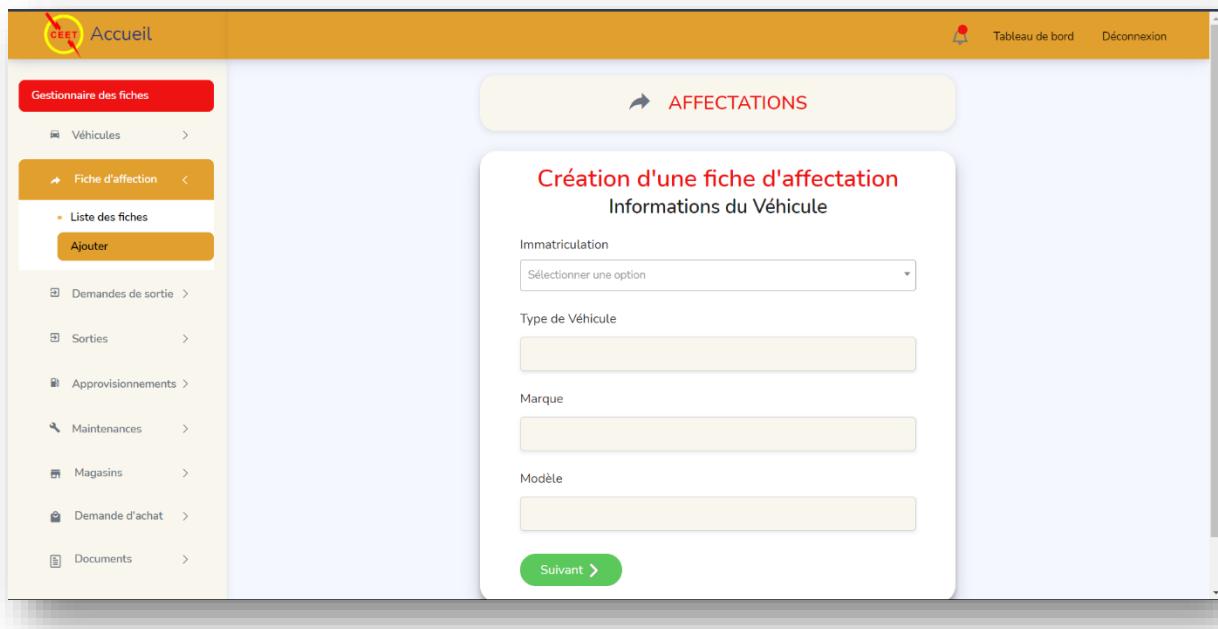
- Redirection pour l'affection :** Après l'ajout d'un nouveau véhicule, on a la possibilité d'être redirigé directement sur la page d'affectation pour affecter le véhicule à un département.

*Figure 54 : Interface de redirection pour affectation d'un nouveau véhicule*

Pour affecter un véhicule :

- **Formulaire d'affectation** : Sélectionner le véhicule à affecter, choisir le département ou la direction bénéficiaire, et spécifier les dates d'affectation.
- **Validation** : Cliquer sur « Valider » pour finaliser l'affectation.

Une fois l'affectation validée, un message de succès confirme l'opération, et le véhicule est mis à jour dans le système comme étant en service dans le département sélectionné.



*Figure 55 : Interface d'affectation d'un véhicule*

### Gestion de la maintenance

Le module de gestion de la maintenance permet de suivre les réparations et entretiens des véhicules du parc. Pour enregistrer une maintenance :

- **Création de fiche de maintenance** : Le responsable remplit une fiche de maintenance en indiquant les détails du véhicule, la nature de la panne, et les travaux à effectuer.
- **Suivi des réparations** : Le système suit l'avancement des réparations et met à jour le statut du véhicule une fois les travaux terminés.



### Gestion de la consommation de carburant

CarControl inclut un module pour la gestion des consommations de carburant, permettant de suivre et de contrôler les réapprovisionnements des véhicules. Pour enregistrer un réapprovisionnement :

- **Formulaire d'approvisionnement** : Remplir les détails du véhicule, le type de carburant utilisé, le kilométrage, et le montant de carburant ajouté.
- **Validation** : Cliquer sur « Enregistrer » pour sauvegarder l'opération.

Le système compare ensuite les consommations réelles avec les estimations, ce qui permet de détecter les anomalies potentielles.



### Gestion des sorties

La gestion des sorties est l'une des fonctionnalités principales de notre système. Incluant les demandes de sorties jusqu'à la terminaison d'une fiche de sortie. Ainsi, on a un formulaire d'ajout d'une demande :

- **Formulaire de demande de sortie** : Remplir les détails sur le demandeur, son département puis sur les motifs de la sortie.
- **Validation** : Cliquer sur le bouton « Terminer » pour enregistrer la demande ou redirigé directement sur l'assignation d'une fiche de sortie.

Figure 56 : Interface d'ajout d'une demande de sortie

Après l'ajout d'une demande, si le véhicule et le chauffeur sont disponibles, alors on passe à la création d'une fiche de sortie. Qui début à la sortie du véhicule et prend fin au retour du véhicule

## Application de gestion du parc automobile de la CEET

- ❖ **Formulaire de départ** : Remplir les informations pour initialiser la sortie, en sélectionnant d'abord la fiche de demande pour laquelle on veut faire la sortie.

The screenshot shows the 'Sortie de Véhicule' creation form. The main title is 'Création d'une fiche de sortie' (Creation of a departure record). Below it is the subtitle 'Informations du Véhicule'. The form contains several dropdown menus and input fields:

- Véhicule:** Sélectionner une option
- Type de Véhicule:** (empty field)
- État du Véhicule au Départ:** Sélectionner une option
- Chauffeur:** Sélectionner une option
- Numéro de Demande:** Sélectionner une option

The left sidebar shows a navigation menu with categories like 'Gestionnaire des fiches', 'Sorties' (selected), and 'Demandes de sortie'.

*Figure 57 : Interface de création d'une fiche de sortie*

- ❖ **Formulaire de retour** : Quand le véhicule revient, on termine le formulaire avec les informations de retour.

The screenshot shows the 'Compléter la fiche de retour' (Complete return record) interface. The main title is 'Compléter la fiche de retour'. The form contains the following fields:

- Date et Heure de Retour:** 16/08/2024 08:42
- Kilométrage de Retour:** (empty field)
- État du Véhicule au Départ:** (empty field)

The left sidebar shows a navigation menu with categories like 'Gestionnaire des fiches', 'Sorties' (selected), and 'Demandes de sortie'.

*Figure 58 : Interface de terminaison d'une fiche de sortie*

#### 5.4. PRESENTATION DES ETATS

*Figure 59 : Liste des affectations de véhicules*

## Application de gestion du parc automobile de la CEET

**AFFECTATIONS**

| N° | Numéro Affectation | Date Affectation | Kilométrage | Fonction                  | Département         | Véhicule   | Date de création de la fiche |
|----|--------------------|------------------|-------------|---------------------------|---------------------|------------|------------------------------|
| 20 | AFF010             | 01-03-2024 00:00 | 38000 km    | Tourisme                  | Logistique          | TG-3243-JJ | 12-08-2024 17:28             |
| 19 | AFF009             | 01-02-2024 00:00 | 35000 km    | Transport scolaire        | Informatique        | TG-1223-II | 12-08-2024 17:28             |
| 18 | AFF008             | 01-01-2024 00:00 | 32000 km    | Service de messagerie     | Ressources Humaines | TG-9202-HH | 12-08-2024 17:28             |
| 17 | AFF007             | 01-12-2023 00:00 | 29000 km    | Livraison express         | Comptabilité        | TG-7181-GG | 12-08-2024 17:28             |
| 16 | AFF006             | 01-11-2023 00:00 | 26000 km    | Inspection de site        | Ressources Humaines | TG-5161-FF | 12-08-2024 17:28             |
| 15 | AFF005             | 01-10-2023 00:00 | 23000 km    | Patrouille de sécurité    | Maintenance         | TG-3141-EE | 12-08-2024 17:28             |
| 14 | AFF004             | 01-09-2023 00:00 | 20000 km    | Transport de matériel     | Comptabilité        | TG-1121-DD | 12-08-2024 17:28             |
| 13 | AFF003             | 01-08-2023 00:00 | 18000 km    | Transport de VIP          | Parc Auto           | TG-9101-CC | 12-08-2024 17:28             |
| 12 | AFF002             | 01-07-2023 00:00 | 15000 km    | Livraison de marchandises | Maintenance         | TG-5678-BB | 12-08-2024 17:28             |
| 11 | AFF001             | 01-06-2023 00:00 | 12000 km    | Transport de personnel    | Parc Auto           | TG-1234-AA | 12-08-2024 17:28             |

**DEMANDES DE SORTIE**

| N° | Numéro demande | Nom du demandeur     | Département  | Date de demande | Objet de la demande  | Adresse de Destination | Date et heure de départ |
|----|----------------|----------------------|--------------|-----------------|----------------------|------------------------|-------------------------|
| 1  | DEM000014/24   | Odit ipsum aut magni | Logistique   | 2023-08-22      | Deserunt dolores tem | Sint soluta quia rat   | 2014-04-10 15:30:00     |
| 2  | DEM000013/24   | KOUDJI Junior        | Informatique | 2011-06-24      | Ducimus quisquam su  | Est consequatur volu   | 1979-05-04 04:26:00     |
| 3  | DEM000012/24   | KOUDJI Junior        | Informatique | 2011-06-24      | Ducimus quisquam su  | Ut in ullamco et rep   | 1995-03-20 17:27:00     |
| 4  | DEM000010/24   | Libero natus labore  | Ventes       | 2007-04-19      | Quisquam harum sunt  | Recusandae Quis sim    | 2014-06-21 01:09:00     |
| 5  | DEM000009/24   | Mirima dolor modi la | Maintenance  | 2010-04-08      | Officia ullamco qui  | Elit error sint am     | 1996-01-02 03:00:00     |
| 6  | DEM000008/24   | Nobis natus a nihil  | Parc Auto    | 1992-09-05      | Est rerum porro qui  | Sapiente et ipsa ut    | 2006-09-06 09:44:00     |

*Figure 60 : Liste des demandes de sortie*

## Application de gestion du parc automobile de la CEET

| N° | Numéro de Sortie | Véhicule   | Chauffeur        | Date de Départ   | Estimation Carburant | Estimation Kilomètres | Date de Création | État de la fiche |
|----|------------------|------------|------------------|------------------|----------------------|-----------------------|------------------|------------------|
| 1  | SORT000012/24    | TG-4578-JK | Bouchard Louis   | 25-01-1976 21:42 | 10 L                 | 70 km                 | 13-08-2024 22:19 | En Cours         |
| 2  | SORT000010/24    | TG-1121-DD | Gagnon Sophie    | 05-12-1982 16:08 | 81 L                 | 6 km                  | 13-08-2024 15:23 | En Cours         |
| 3  | SORT000009/24    | TG-5161-FF | Gauthier Nicolas | 27-05-1975 01:08 | 30 L                 | 13 km                 | 13-08-2024 10:48 | En Cours         |
| 4  | SORT000008/24    | TG-5161-FF | Gauthier Nicolas | 27-05-1975 01:08 | 30 L                 | 13 km                 | 13-08-2024 10:48 | En Cours         |
| 5  | SORT000005/24    | TG-7181-GG | Roy Julien       | 04-10-1984 09:37 | 16 L                 | 74 km                 | 13-08-2024 08:35 | En Cours         |
| 6  | SORT000005/24    | TG-7181-GG | Roy Julien       | 04-10-1984 09:37 | 16 L                 | 74 km                 | 13-08-2024 08:35 | En Cours         |

Figure 61 : Liste des sorties en cours
Figure 62 : Statiques et état actuel du parc

**Conclusion**

La conception et la réalisation de l'application de gestion du parc automobile de la CEET marquent l'aboutissement d'un projet ambitieux, fruit d'une réflexion rigoureuse et d'une exécution minutieuse. Ce projet s'inscrit dans un contexte où l'efficience opérationnelle et la rationalisation des ressources sont devenues des impératifs stratégiques. En réponse à ces exigences, notre application se positionne non seulement comme un outil de gestion mais aussi comme un levier de transformation digitale pour l'entreprise.

Le parcours que nous avons emprunté pour aboutir à ce résultat a été jalonné de défis, tant techniques qu'organisationnels. Mais à chaque étape, une analyse fine et une approche méthodique nous ont permis de surmonter les obstacles. La phase de conception, appuyée par les principes de l'approche orientée objet, a permis de modéliser un système robuste, évolutif, et aligné sur les besoins spécifiques de la CEET.

Ce projet ne se contente pas d'apporter une solution immédiate aux problématiques de gestion du parc automobile ; il anticipe également les besoins futurs en intégrant des fonctionnalités modulables et évolutives. Chaque décision de conception a été guidée par une vision à long terme, assurant que l'application pourra s'adapter aux évolutions technologiques et aux besoins changeants de l'entreprise. En cela, notre travail se distingue par sa durabilité et sa pertinence dans un environnement en constante mutation.

De plus, la mise en œuvre d'une architecture solide, combinée à une sécurité renforcée, garantit non seulement la performance de l'application mais aussi la protection des données sensibles, un enjeu crucial à l'ère du numérique.

En conclusion, ce mémoire n'est pas une simple documentation technique ; c'est le récit d'une transformation, d'une idée qui a pris forme pour devenir une pierre angulaire du fonctionnement opérationnel de la CEET. Avec cette application, nous n'avons pas seulement développé un outil ; nous avons construit un pont vers l'avenir de la gestion efficace des ressources au sein de l'entreprise.

**Bibliographie indicative****Ouvrages**

- ✓ Introduction à UML 2 de Pierre Gérard ;
- ✓ Gestion de projets 2 Tracks Unified Process d'El MAZOUZI Nadia, FILALI Abderrahmane et TAMAGNON Olivier.

**Note de cours**

- ✓ Cours d'Introduction au Génie Logiciel de M. SANI Koffi (2023-2024) ;
- ✓ Cours d'outil de programmation web de M. DOUTI & M. AKOGO (2023 – 2024).

**Anciens mémoires consultés**

- ✓ CONCEPTION ET RÉALISATION D'UNE APPLICATION DE GESTION DES SCELLÉS DE SÉCURITÉ (SCEAUX) DES EQUIPEMENTS DE COMPTAGE DE LA CEET, de SONHOUIN Billali (2020 – 2021)

**Webographie indicative**

- ✓ <https://www.google.com/> : Tout au long du projet ;
- ✓ <https://laravel.com/docs/11.x> / : 03/06/2024 – 15/08/2024, pour consulter la documentation officielle de Laravel ;
- ✓ <https://stackoverflow.com/> : 03/06/2024 – 15/08/2024, pour consulter les éventuelles solutions des erreurs lors de la programmation ;
- ✓ <https://themewagon.com/theme-category/admin-dashboard/>, tout le long du projet ; c'est le site du Template web utilisé ;
- ✓ <https://www.youtube.com/> : Tout au long du projet, pour suivre de différents tutoriels en fonction des besoins ;
- ✓ <https://www.php.net/manual/fr/> : 03/07/2024 – 13/08/2024, pour consulter le manuel de PHP (Rechercher certaines fonctions) ;
- ✓ <https://chatgpt.com/> : Tout le long du projet

## **Table des matières**

|                                                                                  |           |
|----------------------------------------------------------------------------------|-----------|
| Résumé.....                                                                      | iv        |
| Glossaire .....                                                                  | v         |
| Liste des figures .....                                                          | vi        |
| Liste des tableaux .....                                                         | viii      |
| Liste des participants.....                                                      | ix        |
| Introduction.....                                                                | 1         |
| <b>1. Partie I : Cahier des charges .....</b>                                    | <b>2</b>  |
| 1.1. PRÉSENTATIONS .....                                                         | 3         |
| 1.1.1. Brève présentation de l’Institut Africain d’Informatique (IAI-Togo) ..... | 3         |
| 1.1.2. Présentation du cadre de stage.....                                       | 4         |
| 1.2. THÈME DU STAGE .....                                                        | 7         |
| 1.2.1. Présentation du sujet .....                                               | 7         |
| 1.2.2. Problématique du sujet .....                                              | 7         |
| 1.2.3. Intérêt du sujet.....                                                     | 8         |
| 1.2.3.1. Objectifs .....                                                         | 8         |
| 1.2.3.2. Résultats .....                                                         | 8         |
| 1.3. ETUDE DE L’EXISTANT.....                                                    | 9         |
| 1.4. CRITIQUE DE L’EXISTANT .....                                                | 11        |
| 1.4.1. Forces .....                                                              | 11        |
| 1.4.2. Faiblesses .....                                                          | 11        |
| 1.5. PROPOSITIONS ET CHOIX DE SOLUTIONS.....                                     | 12        |
| 1.5.1. Evaluation technique des solutions .....                                  | 12        |
| 1.5.1.1. Première solution : Fleet Complete .....                                | 12        |
| 1.5.1.2. Deuxième solution : Développement d’une application .....               | 13        |
| 1.5.2. Evaluations financières des solutions proposées .....                     | 13        |
| 1.5.3. Choix de la solution.....                                                 | 14        |
| 1.6. PLANNING PRÉVISIONNEL DE RÉALISATION .....                                  | 15        |
| <b>2. Partie II : Analyse et Conception .....</b>                                | <b>18</b> |
| 2.1. CHOIX DE LA METHODE D’ANALYSE ET JUSTIFICATION .....                        | 19        |
| 2.1.1. Approche systémique ou fonctionnelle .....                                | 19        |
| 2.1.2. Approche orientée objet .....                                             | 19        |
| 2.1.3. Langage de modélisation.....                                              | 20        |
| 2.1.4. Processus Two Track Unified Process (2TUP) .....                          | 20        |
| 2.1.5. Synthèse.....                                                             | 21        |
| 2.2. CHOIX DE L’OUTIL DE MODÉLISATION ET JUSTIFICATION .....                     | 21        |

**Application de gestion du parc automobile de la CEET**

|                                       |                                                               |           |
|---------------------------------------|---------------------------------------------------------------|-----------|
| 2.3.                                  | ETUDE DÉTAILLÉE DE LA SOLUTION .....                          | 21        |
| 2.3.1.                                | Modélisation des aspects fonctionnels de la solution .....    | 21        |
| 2.3.1.1.                              | Diagrammes de cas d'utilisations .....                        | 22        |
| 2.3.1.2.                              | Descriptions textuelles de quelques cas d'utilisations .....  | 27        |
| 2.3.2.                                | Modélisation de l'aspect dynamique de la solution .....       | 34        |
| 2.3.3.                                | Modélisation des aspects statiques de la solution .....       | 39        |
| <b>3.</b>                             | <b>Partie III : réalisation et Mise en œuvre .....</b>        | <b>41</b> |
| 3.1.                                  | MATERIELS ET LOGICIELS UTILISES.....                          | 42        |
| 3.1.1.                                | Matériels utilisés .....                                      | 42        |
| 3.1.2.                                | Logiciels utilisés.....                                       | 43        |
| 3.2.                                  | ARCHITECTURE MATERIELLE ET LOGICIELLE DE L'APPLICATION .....  | 46        |
| 3.2.1.                                | Architecture matérielle.....                                  | 46        |
| 3.2.2.                                | Architecture logicielle .....                                 | 47        |
| 3.3.                                  | SECURITE DE L'APPLICATION .....                               | 48        |
| 3.4.                                  | MISE EN PLACE DE LA BASE DE DONNEES .....                     | 49        |
| <b>4.</b>                             | <b>Partie IV : Guide d'exploitation .....</b>                 | <b>58</b> |
| 4.1.                                  | CONFIGURATION LOGICIELLE ET MATÉRIELLE .....                  | 59        |
| 4.1.1.                                | Configuration logicielle.....                                 | 59        |
| 4.1.2.                                | Configuration matérielle.....                                 | 59        |
| 4.2.                                  | DÉPLOIEMENT ET SUIVI.....                                     | 60        |
| 4.2.1.                                | Déploiement de l'application.....                             | 60        |
| 4.2.2.                                | Suivie de l'application .....                                 | 62        |
| 4.3.                                  | MAINTENANCE .....                                             | 62        |
| <b>5.</b>                             | <b>Partie V : Guide d'utilisation.....</b>                    | <b>64</b> |
| 5.1.                                  | DESCRIPTION TEXTUELLE DU LOGICIEL .....                       | 65        |
| 5.2.                                  | PLAN DE NAVIGATION .....                                      | 66        |
| 5.3.                                  | PRESENTATION DES DIFFERENTES INTERFACES DE L'APPLICATION..... | 69        |
| 5.4.                                  | PRESENTATION DES ETATS .....                                  | 74        |
| <b>Conclusion .....</b>               | <b>77</b>                                                     |           |
| <b>Bibliographie indicative .....</b> | <b>I</b>                                                      |           |
| <b>Webographie indicative .....</b>   | <b>II</b>                                                     |           |