

REPUBLIQUE TUNISIENNE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR,
DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DE GABES



Institut Supérieur de l'Informatique de Médenine

RAPPORT DE PROJET DE FIN D'ETUDES

Présenté en vue de l'obtention du

Diplôme : **Mastère Professionnel en Ingénierie des Logiciels et des
Connaissance**

Conception et Développement d'une Application web Pour le Service De Remorquage

Par
Tombari Ahmed

Entreprise / Organisme d'accueil
ALTAIR SOLUTION

Jury

Président : Mr. BEN
OTHMAN Soufiene
Rapporteur : Mr.
SASSI Aymen
Encadreur : Mme. SAADAOUI Zakia
Encadreur : Mr. ABBASSI Sami

Remerciements

Je tiens tout d'abord à remercier mon Dieu le tout puissant et miséricordieux, qui ma donnée la force et la patience d'accomplir ce modeste travail.

Ensuite, Nous tenons à remercier vivement tous les membres du jury, pour l'attention qu'ils ont porté à notre travail.

Je souhaite exprimer mon respect et mon entière gratitude à mon encadrante **Mme. Zakia SAADAoui de l'ISIMed** pour son encouragement. Son écoute, ses conseils et sa confiance m'ont permis de mener à bien ce projet.

Je tiens à remercier vivement mon encadrant en entreprise, **Mr. Sami ABBASSI**, pour l'accueil, le temps passé ensemble et le partage de ses expertises au quotidien. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions. Il fut d'une aide précieuse dans les moments les plus délicats.

Enfin, un grand merci à tous les enseignants à l'**ISIMed** qui nous ont encouragé tout au long de la préparation de ce travail et pour les connaissances et le savoir-faire acquis durant toute ma formation universitaire.

Dédicaces

*Du plus profond de mon cœur et avec le plus grand plaisir de
ce monde, je dédie ce travail :*

*A mes chers parents que j'aime énormément ;
Qui m'ont toujours encouragé et soutenu tout au long de
mes études et.*

*Que dieu les protège et leur préserve la santé et le
bonheur.*

*A mes chères sœurs,
Qui m'ont toujours accordé leurs attentions.*

*A tous mes amis
A tous ceux qui me sont chers,
Ses nombreux sacrifices et ses efforts sont ici
couronnés.*

*Qu'ils trouvent dans ce travail le témoignage de ma profonde
reconnaissance et de mon amour infini.*

 ...Ahmed

Sommaire

1.1	Introduction Générale	1
2.1	Chapitre 1 :Étude préalable	3
1.	Introduction	3
2.	Présentation de l'organisme d'accueil	3
3.	Description du projet et ses objectifs	3
4.	Etude de l'existant et critiques.....	3
5.	Solution proposée	5
6.	Conclusion	6
3.1	Chapitre 2 : Spécification des besoins.....	7
1.	Introduction	7
2.	Besoins fonctionnels.....	7
7.	Conclusion	20
4.1	Chapitre3 :Etude conceptuelle.....	21
1.	Introduction	21
2.	Langage de modélisation UML	21
3.	Conception de l'application : vue statique.....	21
4.	Conception de l'application : Vue dynamique.....	24
5.	Conclusion	30
	Chapitre 4 : Réalisation	31
1.	Introduction	31
2.	Spécification de l'environnement de travail	31
2.1	L'architecture utilisée.....	31
1.1.1	Architecture de l'application web.....	31
2.2	Environnement matériel.....	32
2.3	Spécification de l'environnement logiciel.....	33
2.4	Technologies et langages utilisés	34
2.5	Techniques utilisées pour la géolocalisation	35
	Leaflet Maps	35
	Map box	35
2.6	Techniques utilisées pour la Notification	36
2.7	Protocol et format de données	36
2.7.1	Protocol de communication.....	36
2.7.2	Format de données	36
3.	Réalisation : les interfaces.....	37
3.1	Partie Administrateur :	37
2.1.1	Interface login de l'administrateur	37

3.1.2	Interface de tableau de bord de l'administrateur	38
3.1.3	Interface de List utilisateurs de l'administrateur	38
3.1.4	Interface de List véhicules de l'administrateur	39
3.1.5	Interface de List camions de l'administrateur	39
3.1.6	Interface de List chauffeurs de l'administrateur	40
3.1.7	Interface de List demandes des remorquages de l'administrateur	40
3.1.8	Interface de List de nouveau utilisateurs de l'administrateur	41
3.2	Partie Client :.....	42
3.2.1	Interface login de Client	42
3.3	Partie Chauffeur :.....	47
3.3.1	Interface login de Chauffeur	47
4.	Conclusion.....	52
	Conclusion Générale.....	53

Liste de figures

Figure 1. Logo de l'organisme d'accueil.....	2
Figure 2. diagramme cas utilisation global.....	7
Figure 3. Diagramme de cas d'utilisation « Gérer véhicules ».....	8
Figure 4. Diagramme de cas d'utilisation « Gérer demandes »	9
Figure 5. Diagramme de cas d'utilisation « Gérer profil »	10
Figure 6. Diagramme de cas d'utilisation « Contacter l'agence ».....	11
Figure 7. Diagramme de cas d'utilisation « Traiter la demande de remorquage ».....	12
Figure 8. Diagramme de cas d'utilisation « Valider compte de client »	13
Figure 9. Diagramme de cas d'utilisation « Gérer les Chauffeurs ».....	14
Figure 10. Diagramme de cas d'utilisation « Gérer les Camions ».....	15
Figure 11. Diagramme de cas d'utilisation « Gérer les ».....	16
Figure 12. Diagramme de cas d'utilisation « Géolocaliser des Utilisateurs ».....	17
Figure 13. Diagramme de cas d'utilisation « Remorquer une Véhicule »	18
Figure 14. Diagramme du class.....	21
Figure 15. Diagramme de séquence cas utilisation « Authentifier ».....	22
Figure 16. Diagramme de séquence pour le cas utilisation « Gérer Véhicule »	23
Figure 17. Diagramme de séquence pour le cas utilisation « Gérer Demande »	24
Figure 18. Diagramme de séquence pour le cas utilisation « Gérer Profil ».	25
Figure 19. Diagramme de séquence pour le cas utilisation « Gérer Chauffeur ».....	26
Figure 20. Diagramme de séquence de cas utilisation « Gérer Camion ».....	27
Figure 21. Diagramme de séquence pour le cas d'utilisation « Gérer Véhicule »	28
Figure 22. Etat transition de l'objet « remorquage ».....	29
Figure 23. Etat transition de l'objet « demande »	29
Figure 24. Diagramme de déploiement.....	30

Liste des tableaux

Tableau 1.Avantages et inconvénients des applications existantes.	3
Tableau 2.Description textuelle du cas d'utilisation " Gérer véhicules "	8
Tableau 3.Description textuelle du cas d'utilisation " Gérer demandes "	9
Tableau 4.Description textuelle du cas d'utilisation " Gérer profil "	10
Tableau 5.Description textuelle du cas d'utilisation " Contacter l'agence ".....	11
Tableau 6.Description textuelle du cas d'utilisation " Traiter la demande de remorquage"	12
Tableau 7.Description textuelle du cas d'utilisation " Valider compte de client ".....	13
Tableau 8.Description textuelle du cas d'utilisation " Gérer les Chauffeurs "	14
Tableau 9.Description textuelle du cas d'utilisation " Gérer les camions "	15
Tableau 10.Description textuelle du cas d'utilisation " Gérer les Véhicules "	16
Tableau 11.Description textuelle du cas d'utilisation " Géolocaliser des Utilisateurs " .	17
Tableau 12.Description textuelle du cas d'utilisation " Remorquer une Véhicule "	18

Introduction Générale

Aujourd'hui, suite à l'évolution exponentielle des technologies de communications et l'émergence des traitements automatique de l'information, la technologie des solutions web et est devenue un vrai l'intérêt. Son succès provient de sa facilité d'utilisation et de sa rapidité d'échange de l'information en temps réel.

Dans ce contexte, il nous été imploré de développer une application web. Pour le service de remorquage des voitures au cas de panne afin d'élargir l'espace des canaux de communication avec les clients.

En effet, notre application web ALTAIR REMORQUAGE est considérée comme une application permettant de faciliter la possibilité d'assistance à la clientèle pour le remorquage express.

Ce travail sera représenté par quatre chapitres :

- Le premier chapitre intitulé « Étude préalable », nous allons présenter L'entreprise d'accueil et notre application.
- Le second chapitre intitulé « Spécification des besoins », nous commencerons D'analyser l'existant et déterminer et nous identifions les besoins fonctionnels Et les besoins non fonctionnels.
- Le chapitre trois intitulé « Etude conceptuelle », nous tenterons d'approfondir la Compréhension de la structure interne des éléments et leurs relations les uns par rapport aux autres et d'obtenir une spécification, une analyse et une conception détaillées.
- Le dernier chapitre intitulé « Réalisation », nous allons détaillée l'architecture de notre Application en présentant l'environnement matériel et logiciel utilisé et mettre des vues sur ce dernière dans son état final.

Enfin, ce rapport sera entouré par une conclusion qui montre les bénéfices acquis durant l'élaboration de ce projet avec quelques perspectives

Chapitre 1 : Étude préalable

1. Introduction

Dans ce chapitre, nous commençons, en premier lieu, par présenter l'organisme d'accueil. Nous procédons en quatre sections. Nous commençons par la présentation de l'organisme d'accueil. La deuxième section sera pour la description générale du projet et ses objectifs. La troisième, mettra l'accent sur l'étude de l'existant et les critiques. La quatrième section sera pour les solutions proposées.

2. Présentation de l'organisme d'accueil

Dans cette section, nous présentons la société ALTAIR SOLUTION ainsi que ses activités principales.

2.1 Présentation de l'organisme d'accueil

ALTAIR SOLUTION est une jeune société à technopole Manouba. Ses activités sont diversifiées, à très haute valeur technologique, et orientées vers les marchés de l'avenir : Le développement web, mobile, ERP, apprentissage en ligne... En outre, elle a recours à des environnements technologiques sophistiqués pour offrir des services d'ingénierie de haute qualité. De ce fait, elle nous a offert l'environnement idéal pour développer nos compétences et approfondir notre formation.



Figure 1. Logo de l'organisme d'accueil

3. Description du projet et ses objectifs

3.1 Etude de l'existant et critiques

Pour pouvoir créer un projet performant, nous devons passer par une étape d'étude de l'existant pour nous collaborer à suggérer une conception évolutive à notre projet. En effet, dans

cette partie nous allons essayer de donner un aperçu sur l'existant en présentant deux applications : « Route4Me » et « Towbook » qui se rapprochent de notre application dont l'objectif est de mettre en place notre application.

Tableau 1. Avantages et inconvénients des applications existantes.

Application	Description	Avantages	Inconvénients
Route4Me	<p>*Route4Me est un logiciel de planification d'itinéraire qui offre aux petites et moyennes entreprises une plateforme intelligente à partir de laquelle créer, gérer et partager des itinéraires optimisés.</p> <p>* En intégrant des fonctions de cartographie, de recherche et de catégorisation, Route4Me vise à fournir une solution de planification d'itinéraire conviviale.</p>	<p>*Planification des itinéraires en quelques secondes.</p> <p>*Suivi des chauffeurs en temps réel et savoir s'ils sont à l'heure sur leur itinéraire.</p>	<p>*Application payée.</p> <p>*Certaines fonctionnalités nécessaires sont inactivées.</p> <p>*l'angle de mappage est un peu décalé.</p>
Towbook	<p>*Est un Outil de gestion web destiné aux entreprises de</p>	<p>* Automatisation de la facturation</p>	<p>* quelques bugs qui font planter l'application de</p>

<p>remorquage et utilisé pour la répartition et la confiscation des véhicules, la facturation et bien plus.</p> <p>* Avec Towbook, le transfert des appels de répartition sur les téléphones de vos chauffeurs et la gestion des relevés de facturation en quelques clics sont simplement disponibles.</p> <p>* Towbook offre également des applications Android et iOS (iPhone/iPad).</p>	<p>* Soumission automatique aux clubs automobiles</p> <p>* Suivi GPS des chauffeurs</p> <p>* Rapports faciles à utiliser</p> <p>* Suivi des commissions</p> <p>* Service client de 1ère classe</p>	<p>temps en temps et parfois difficile à charger.</p> <p>* Mauvaise ergonomie (les interfaces sont incompressibles).</p>
--	--	--

3.2 Solution proposée

Pour ce faire, et après l'étude des applications déjà existantes Altair Solution propose la conception et la réalisation d'une application web de système de remorquage, basé sur le positionnement Géolocalisation par satellite (GPS), qui aide l'utilisateur à déterminer sa position. Afin d'apporter une valeur ajoutée et un meilleur service aux clients, nous proposons une application web à usage simple qui assure la disponibilité et facilite la communication et l'exploitation des services d'assistance voiture en panne pour les clients. Cette dernière permet interaction intelligente et efficace avec les clients.

4. Conclusion

Dans ce premier chapitre, tout d'abord nous avons commencé par la présentation l'organisme d'accueil et ses services offertes, par la suite le contexte du projet : description générale dès l'objective de l'application, étude de l'existant et critiques et la solution proposée. Le chapitre suivant sera les spécifications des besoins détaillé de l'application.

Chapitre 2 : Spécification des besoins

1. Introduction

La spécification des besoins présente la première étape dans le cycle de développement d'un tel projet informatique du fait qu'elle nous permet de mieux comprendre le système à développer.

En effet, les objectifs représentent les fonctionnalités qui devraient être présentées dans l'application, ce qui permet aussi de valider l'application au fur et à mesure du développement.

2. Besoins fonctionnels

2.1 Identificateurs des acteurs

Un acteur représente une personne qui interagit avec le système. Nous allons, tout d'abord, identifier les différents acteurs de notre application puis dans un deuxième lieu de modéliser des exigences pour chaque acteur. Dans cette application, trois acteurs peuvent être distingués :

- Administrateur : Utilisateur de l'application web (partie back office).
- Client : Utilisateur de l'application web (demander un remorquage).
- Chauffeur : Utilisateur de l'application web (remorquer les voitures en pannes).

Acteur	Description
Client	C'est un acteur qui a besoin d'utiliser l'application pour Demander un camion de remorquage et suivre cette demande.
Administrateur	L'administrateur effectue la plupart des tâches et qui est en contact avec les clients et les chauffeurs via notre application.
Chauffeur	Le chauffeur modifier la disponibilité d'un camion et remorquer les voitures en panne.

2.2 Spécifications des fonctionnels

Cette étape consiste à exprimer le système de point de vue utilisateur, c'est-à-dire l'aspect métier du système. L'application doit fournir les fonctionnalités suivantes à :

- ❖ **L'administrateur** : Acteur de l'application web (back office), il est chargé de
 - S'authentifier : L'application permet aux utilisateurs de s'authentifier afin de profiter certaines fonctionnalités de l'application
 - Gérer profile : L'application offre la possibilité de gérer votre compte à travers les fonctionnalités de supprimer, modifier un compte afin de mieux utiliser ses fonctionnalités.
 - Gérer les Voitures : le système permet à l'administrateur de gérer les voitures avec les fonctionnalités d'ajout, suppression et modification.
 - Gérer les camions de remorquage : le système permet à l'administrateur de gérer ses camions avec les fonctionnalités d'ajout, suppression et modification.
 - Gérer les chauffeurs : le système permet à l'administrateur de gérer ses chauffeurs avec les fonctionnalités d'ajout, suppression et modification.
 - Traiter les demandes de remorquage : l'application offre à l'utilisateur D'abord, de localiser la position actuelle d'un client. Ensuite, la consultation des réseaux des camions de remorquage en se basant sur la demande et la position reçu.
 - Valider les comptes des clients : L'application permet aux l'administrateur de valider ou bloquée les comptes des clients
- ❖ **Le client** : Acteur de l'application. Il est chargé de :
 - S'authentifier : L'application permet aux utilisateurs de s'authentifier afin de profiter certaine fonctionnalité de l'application.
 - Gérer Véhicules : le client peut ajouter, modifier et supprimer des véhicules.
 - Gérer profile : un client peut modifier, désactiver son profil.
 - Gérer Demandes : le client peut envoyer, modifier ou annuler une demande de remorquage.
 - Contacter l'agence : le système donne au client la possibilité de contacter l'agence
- ❖ **Le chauffeur** : Acteur de l'application. Il est chargé de :
 - S'authentifier : L'application permet aux utilisateurs de s'authentifier afin de profiter certaine fonctionnalité de l'application.

- Assurer le mécanisme de remorquage : Le chauffeur remorqué les voitures en panne
- Modifier l'état d'un camion : le chauffeur met à jour le statut de camion de remorquage.

2.3 Modélisation des besoins fonctionnels

Après la présentation des besoins fonctionnels, nous passons à la modélisation par la définition de ces cas d'utilisation.

2.3.1 Identification des cas d'utilisations

Acteurs et cas d'utilisation sont les concepts UML fondamentaux pour les spécifications des exigences. Dans cette section, nous les identifierons à partir de l'expression initiale des besoins de notre étude de cas. Nous structurerons, relierons et classerons ensuite ces cas d'utilisation et élaborerons les représentations graphiques UML associée. La figure ci-dessous présente le digramme de cas utilisation global.

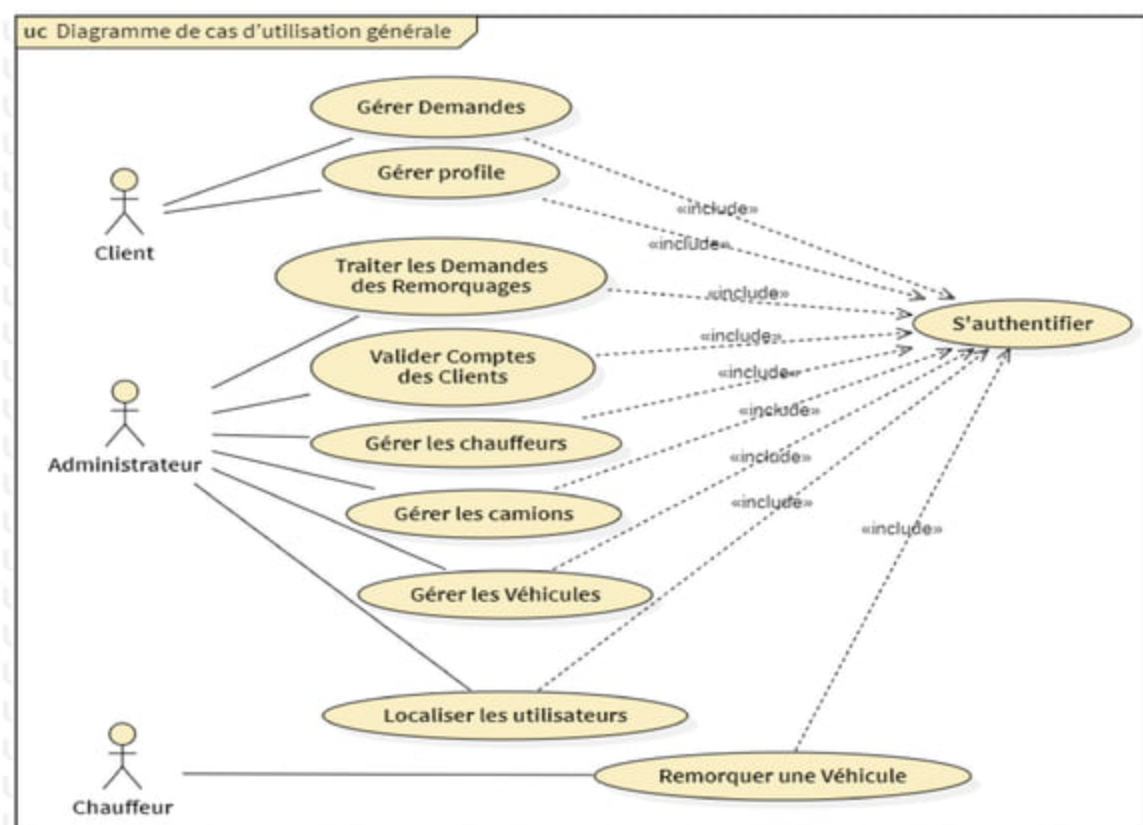


Figure 2. Diagramme cas utilisation global

2.3.2 Raffinement des cas d'utilisation

Afin de mieux comprendre un fonctionnement d'un cas d'utilisation, nous allons présenter une table descriptive pour les principaux cas d'utilisations pour notre application.

2.3.2.1 Analyse du cas d'utilisation « Gérer demandes »

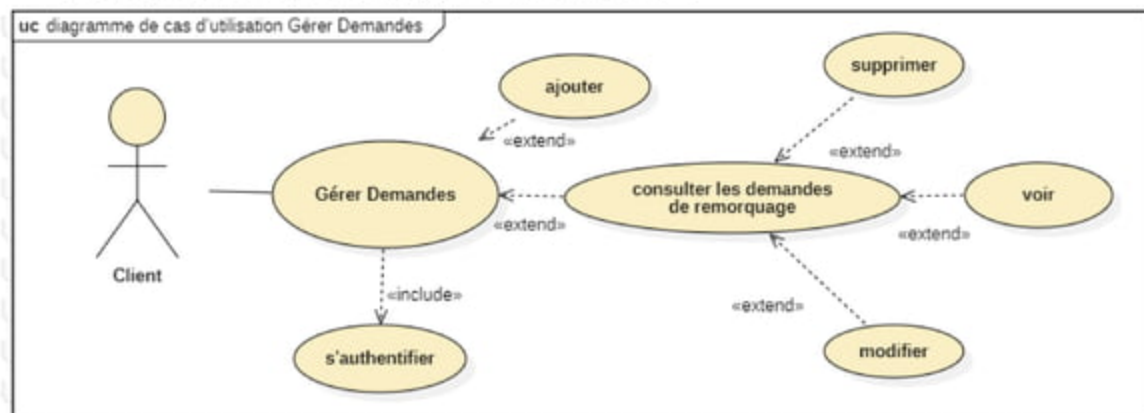


Figure 4. Diagramme de cas d'utilisation « Gérer demandes »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « Gérer demandes ».

Tableau 3. Description textuelle du cas d'utilisation " Gérer demandes "

Cas d'utilisation	Gérer demandes
Description	Ce cas d'utilisation permet au Client de Gérer son véhicules (demander, voir, modifier et supprimer d'une demande).
Acteurs	Le Client
Préconditions	L'action de demande est changée
Post conditions	L'ouverture de la session privée
Scénario principal	<ol style="list-style-type: none"> 1. Le système affiche l'interface relative à « espace de client » 2. Le système offre la possibilité d'appeler l'administrateur de service de remorquage ou d'envoyer une demande à travers des informations relatives au nom, prénom, nombre de personne, type de voiture, type de panne et la position actuelle, il effectué la tâche de suppression au de modification des Demandes. 3. Le système effectue l'envoi de demande avec succès.

2.3.2.2 Analyse de cas utilisation « Gérer profil »

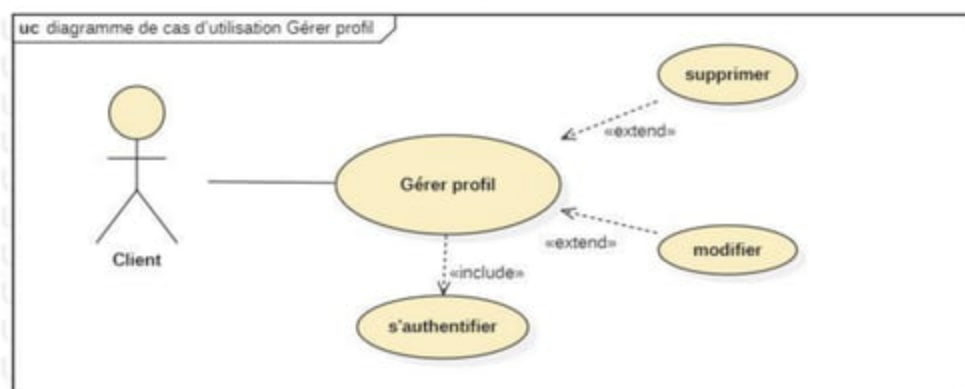


Figure 5. Diagramme de cas d'utilisation « Gérer profil »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « Gérer profil ».

Tableau 4. Description textuelle du cas d'utilisation " Gérer profil "

Cas d'utilisation	Gérer profil
Description	Ce cas d'utilisation permet à l'utilisateur de gérer son profil (Modifier et supprimer).
Acteurs	Le Client
Préconditions	L'utilisateur authentifié
Post conditions	Mis à jour effectué
Scénario principal	<ol style="list-style-type: none"> 1. Le cas d'utilisation se déclenche lorsque l'acteur demande une mise-à-jour de son compte. 2. L'acteur accède à son profile. 3. Il effectue les modifications nécessaires (changer photo du profil, modification des coordonnées personnelles ...) ou suppression du compte 4. Validation d'action

2.3.2.3 Analyse du cas d'utilisation « Traiter la demande de remorquage »

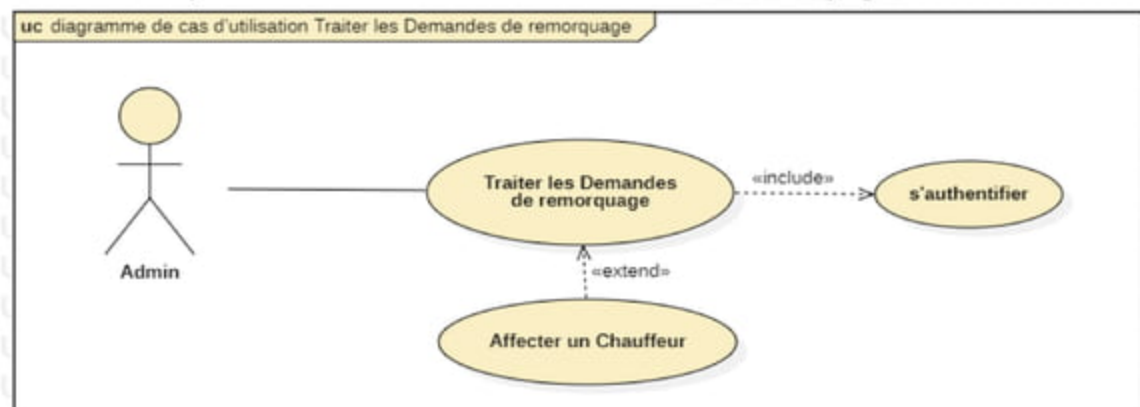


Figure 7. Diagramme de cas d'utilisation « Traiter la demande de remorquage »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « **Traiter la demande de remorquage** ».

Tableau 6. Description textuelle du cas d'utilisation " Traiter la demande de remorquage "

Cas d'utilisation	Traiter la demande de remorquage
Description	Ce cas d'utilisation permet à l'utilisateur Traiter la demande de remorquage
Acteurs	Administrateur
Préconditions	Administrateur authentifié
Post conditions	L'ouverture de la session privée
Scénario principal	<ol style="list-style-type: none"> 1. L'Administrateur accède à la liste des demandes. 2. Il choisit une demande. 3. Il choisit un chauffeur disponible après il clique sur le bouton «accepter» pour accepter et affecter un Chauffeur à la demande envoyer par Client .
Scénario alternatif	E1 : si l'état de déchet est « en attente » : 1. L'application affiche un message « en attente » quand l'administrateur ne fait pas aucune action

2.3.2.4 Analyse du cas d'utilisation « Valider compte de client »

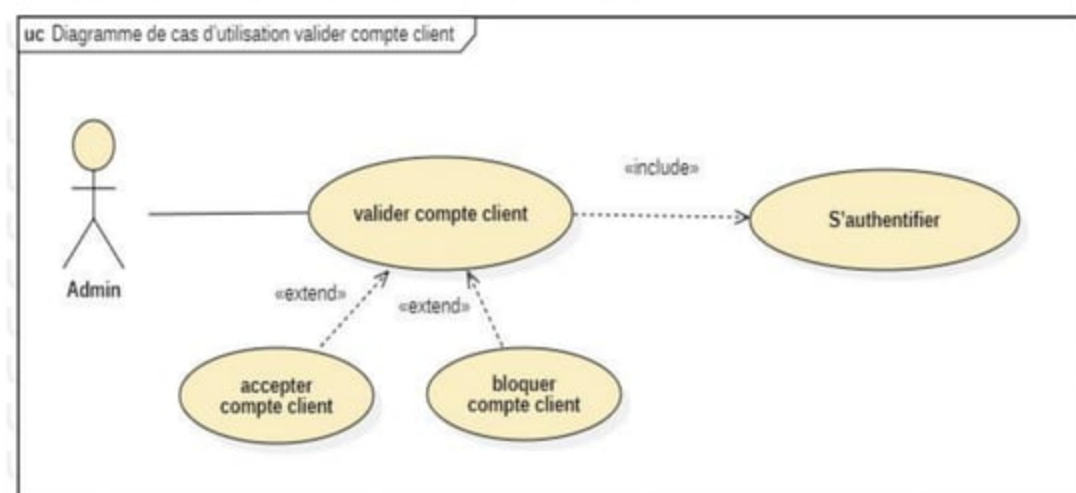


Figure 8. Diagramme de cas d'utilisation « Valider compte de client »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « **Valider compte de client** ».

Tableau 7. Description textuelle du cas d'utilisation " Valider compte de client "

Cas d'utilisation	Valider compte de client
Description	Ce cas d'utilisation permet à l'administrateur de Valider les comptes des Clients.
Acteurs	Administrateur
Préconditions	Administrateur authentifié
Post conditions	Mis-à-jour effectué
Scénario principal	<ol style="list-style-type: none"> 1. L'Administrateur accède à la liste des Clients. 2. L'administrateur choisi l'utilisateur dans la liste des Nouveau Clients. 3. Il choit l'action (accepter ou supprimer ou bloquer). 4. Validation du mis-à-jour

2.3.2.5 Analyse du cas d'utilisation « Gérer les Chauffeurs »

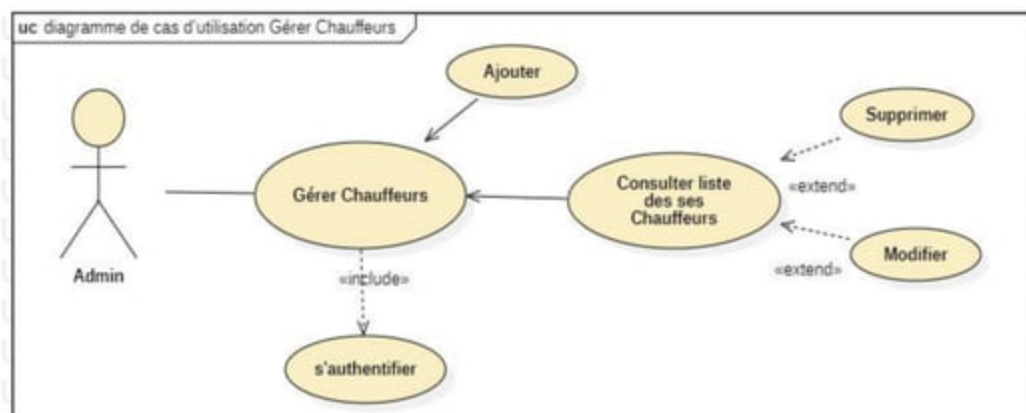


Figure 9. Diagramme de cas d'utilisation « Gérer les Chauffeurs »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « Gérer les Chauffeurs ».

Tableau 8. Description textuelle du cas d'utilisation " Gérer les Chauffeurs "

Cas d'utilisation	Gérer les chauffeurs
Description	Ce cas d'utilisation permet à l'administrateur de Gérer les Chauffeurs (ajouter, modifier et supprimer d'un chauffeur).
Acteurs	Administrateur
Préconditions	Administrateur authentifié
Post conditions	L'ouverture de la session privée
Scénario principal	<ol style="list-style-type: none"> 1. L'Administrateur accède à la liste des Chauffeurs. 2. L'administrateur effectuée la tâche d'ajout, suppression au de modification d'un Chauffeur. 3. Validation du mis-à-jour

2.3.2.6 Analyse du cas d'utilisation « Gérer les Camions »

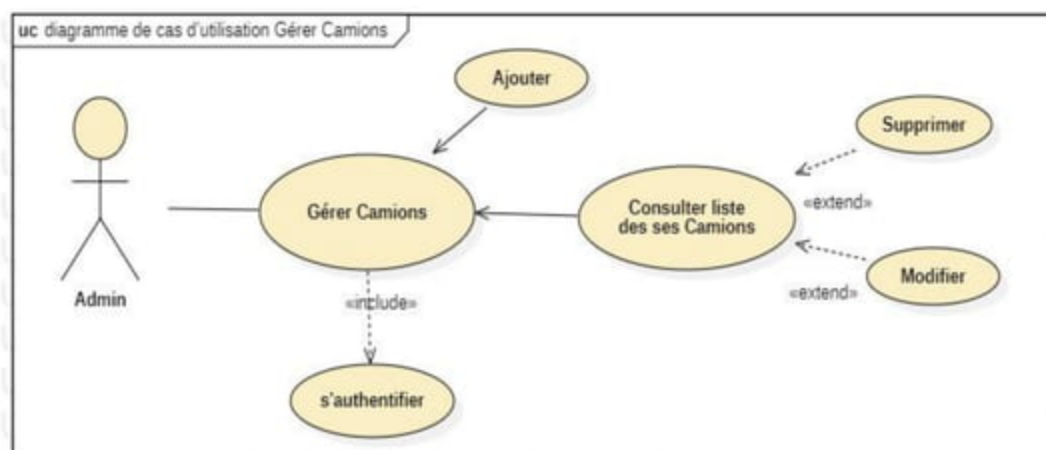


Figure 10. Diagramme de cas d'utilisation « Gérer les Camions »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « **Gérer les Camions** ».

Tableau 9. Description textuelle du cas d'utilisation " Gérer les camions "

Cas d'utilisation	Gérer les Camions
Description	Ce cas d'utilisation permet à l'administrateur de Gérer les Camions (ajouter, modifier et supprimer d'un chauffeur).
Acteurs	Administrateur
Préconditions	Administrateur authentifié
Post conditions	L'ouverture de la session privée
Scénario principal	<ol style="list-style-type: none"> 1. L'Administrateur accède à la liste des Camions. 2. L'administrateur effectuée la tâche d'ajout, suppression au de modification d'un Camion. 3. Validation du mis-à-jour

2.3.2.7 Analyse du cas d'utilisation « Gérer les Véhicules »

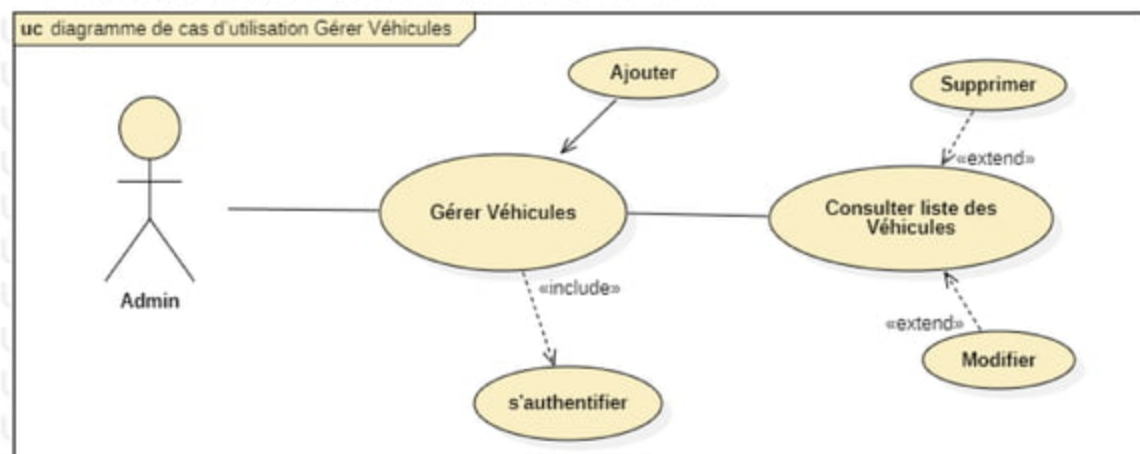


Figure 11. Diagramme de cas d'utilisation « Gérer les véhicules »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « Gérer les Véhicules ».

Tableau 10. Description textuelle du cas d'utilisation " Gérer les Véhicules "

Cas d'utilisation	Gérer les Véhicules
Description	Ce cas d'utilisation permet à l'administrateur de Gérer les Véhicules (ajouter, modifier et supprimer).
Acteurs	Administrateur
Préconditions	Administrateur authentifié
Post conditions	L'ouverture de la session privée
Scénario principal	<ol style="list-style-type: none"> 1. L'Administrateur accède à la liste des Véhicules. 2. L'administrateur effectuée la tâche d'ajout, suppression au de modification d'un Véhicule. 3. Validation du mis-à-jour

2.3.2.8 Analyse du cas d'utilisation « Géolocaliser des Utilisateurs »

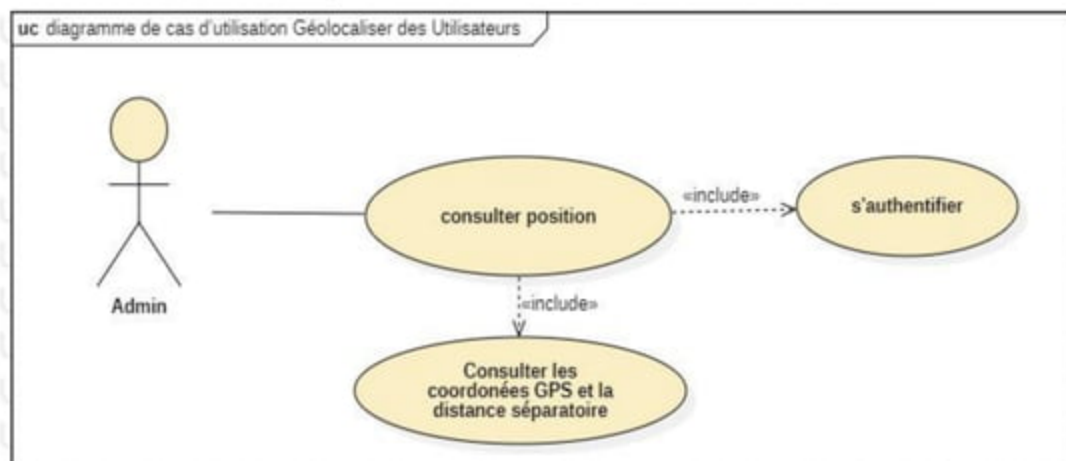


Figure 12. Diagramme de cas d'utilisation « Géolocaliser des Utilisateurs »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « **Géolocaliser des Utilisateurs** ».

Tableau 11. Description textuelle du cas d'utilisation " Géolocaliser des Utilisateurs "

Cas d'utilisation	Géolocalisation des utilisateurs
Description	Ce cas d'utilisation permet à l'administrateur de Géolocaliser des Utilisateurs (Client, Chauffeur, Administrateur).
Acteurs	Administrateur
Préconditions	Administrateur authentifié
Post conditions	L'ouverture de la session privée
Scénario principal	<ol style="list-style-type: none"> 1. L'Administrateur accède à la liste de localisation. 2. Localiser des toutes les Utilisateurs 3. Validation du mis-à-jour

2.3.2.9 Analyse du cas d'utilisation « Remorquer une Véhicule »

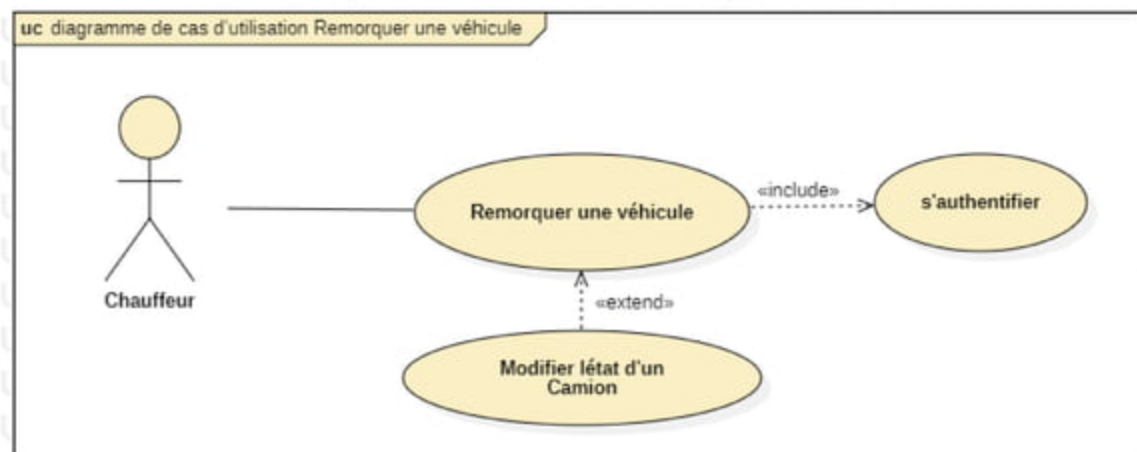


Figure 13. Diagramme de cas d'utilisation « Remorquer une Véhicule »

Le tableau ci-dessous illustre l'analyse détaillé du cas d'utilisation « **Remorquer une Véhicule** ».

Tableau 12. Description textuelle du cas d'utilisation " Remorquer une Véhicule "

Cas d'utilisation	Remorquer une Véhicule
Description	Ce cas d'utilisation permet de remorquer et Modifier l'état d'un Camion (Disponible / En Service).
Acteurs	Chauffeur
Préconditions	Chauffeur authentifié
Post conditions	L'ouverture de la session privée
Scénario principal	<ol style="list-style-type: none"> 1. Le Chauffeur accède à la liste des Camions. 2. Le Chauffeur choisit le Camion. 3. Le Chauffeur Modifier l'état d'un camion (Disponible, En Service).

2.4 Besoins non fonctionnels

Les besoins non fonctionnels ont les besoins qui caractérisent le système. Ce sont des besoins qui Résumant les contraintes et les caractéristiques techniques imposées par l'application.

Les besoins que notre application doit garantir sont :

- ❖ **La sécurité** : l'échange des données entre l'application et la base de données sera à travers un service web.
- ❖ **La rapidité** : L'application doit être optimisée pour avoir un court temps de réponse.
- ❖ **L'ergonomie** : L'application doit présenter des interfaces intuitives et compréhensibles par les utilisateurs.
- ❖ **La maintenabilité** : L'application doit être facile à maintenir.
- ❖ **La fiabilité** : L'application doit garantir la pertinence des données lors de l'échange avec la base de données.

3. Conclusion

Tout au long de ce chapitre, nous avons détaillé les besoins fonctionnels et non fonctionnels de notre Projet qui va être utilisé dans la réalisation de l'application. L'étude conceptuelle fait l'objet du chapitre suivant.

Chapitre 3 : Etude conceptuelle

1. Introduction

Ce chapitre est essentiellement axé sur la phase de conception détaillée. En effet, L'objectif de cette phase est de reprendre le modelé d'analyse et le refaire d'une façon plus raffinée pour dégager les modelés statique et dynamique.

2. Langage de modélisation UML

L'UML est un langage qui possède la capacité de satisfaire tous les besoins de la conception et il permet de représenter le système selon les différentes vues complémentaire grâce aux diagrammes.

Pour cette raison, nous l'avons choisi comme langage pour modéliser les fonctionnalités que notre système doit offrir d'une part et pour représenter les interactions entre ses différentes composantes d'autre part.

En effet, l'UML représente 14 diagrammes destinés à l'analyse et à la conception d'un système d'information. C'est ainsi, pour représenter la vue statique du système on opte le diagramme de classes [6]. En revanche, les diagrammes de séquences et ceux d'état-transition sont choisis pour la vue dynamique.

3. Conception de l'application : vue statique

La vue statique représente des schémas de conception à caractère stable à l'immobile pour les différents processus de l'application. Cette vue se traduit en termes d'acteurs.

3.1 Diagramme de classe

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que la différente relation entre celles-ci. Les diagrammes de classes fait partie de la partie statique d'UML car il fait abstraction des Aspects temporels et dynamiques. Une classe décrit les responsabilités, les comportements et le type d'un ensemble d'objets. Les objets sont des instances de classes et les liens sont des instances de relations.

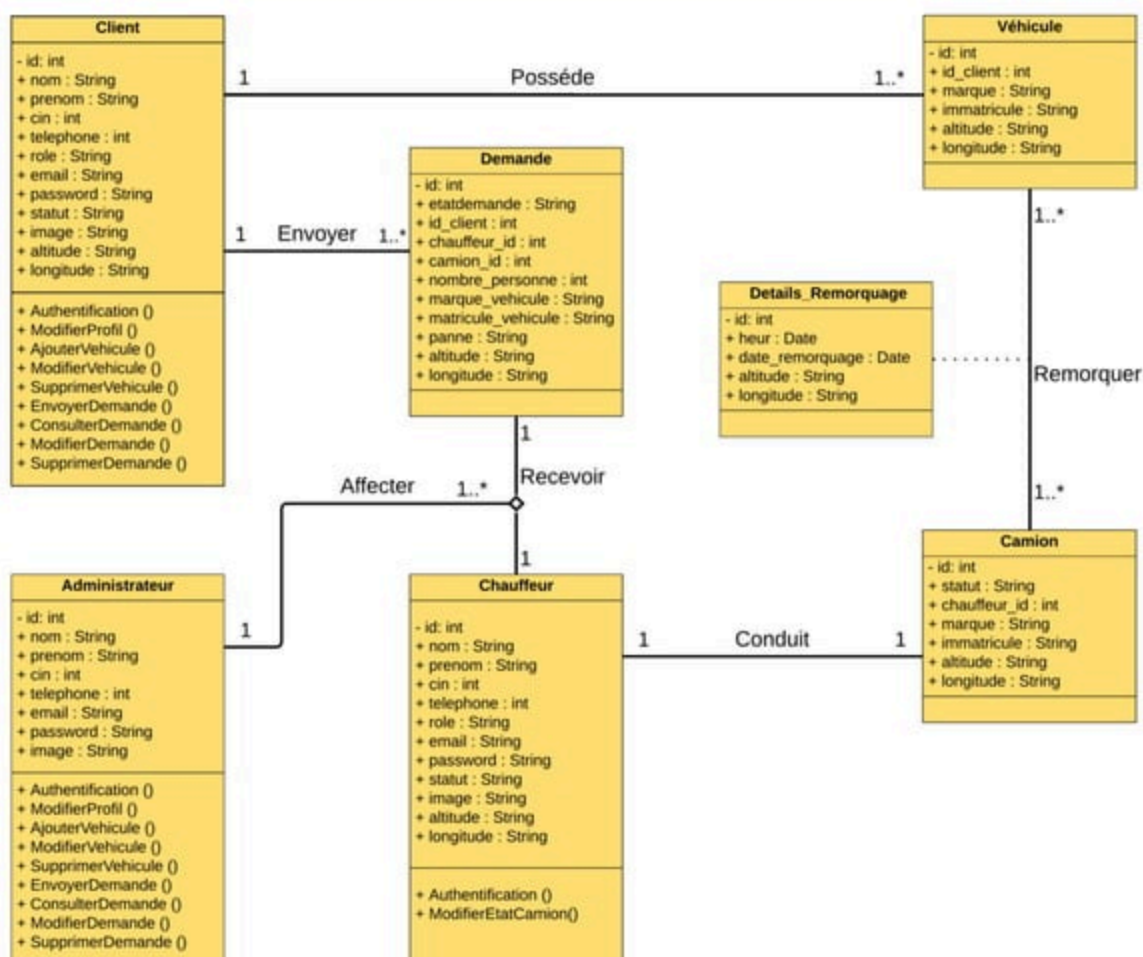


Figure 14. Diagramme du class

Dans notre application, nous avons utilisé les classes suivantes :

❖ **La classe « Client »** : contient les informations concernant les Clients de l'application web : son nom, prénom, cin, email, téléphone, password.

❖ **La classe « Administrateur »** : contient les informations concernant les Administrateurs de l'application web : son nom, prénom, cin, email, téléphone, password.

❖ **La classe « Chauffeur »** : contient les informations concernant les Chauffeurs de l'application web : son nom, prénom, cin, email, téléphone, password.

❖ **La classe « Véhicule »** : contient toutes les informations pour chaque véhicule tel que : le marque de véhicule, le matricule et les serie.

❖ **La classe « Demande »** : contient les informations concernant chaque demande de remorquage de véhicule envoyé par les clients aux administrateurs.

Une demande est caractérisée par : id de demande, mat véhicule, id_client, l'attitude, longitude, nombre_personne, type_voiture, la date d'envoi et l'action (en attente, accepter, refuser). L'action par défaut est « en attente ».

❖ **La classe « Camion »** : contient toutes les informations pour chaque Camions tell que: id_camion, le modèle de camion, id_chauffeur, le statut de camion (disponible, en service).

❖ **La classe « Remorquer »** : contient les informations pour chaque remorquage tell que: Heur, date de remorquage, géolocalisation (altitude, longitude).

4. Conception de l'application : Vue dynamique

La vue dynamique met l'accent sur le comportement dynamique du système en montrant les collaborations entre les objets et les modifications apportées à états internes d'objets. Durant cette phase, nous allons décrire le comportement général du système en présentant les diagrammes de séquences et les diagrammes d'état-transition.

4.1 Diagramme de séquences :

Le diagramme de séquence décrit la séquence temporelle des échanges de messages entre les objets et l'acteur pour réaliser une certaine tâche. Les diagrammes de séquence représentent une façon excellente pour documenter les scénarios d'utilisation, identifier les différents objets et vérifier leur utilisation.

4.1.1 Diagramme de séquence « Authentifier »

La figure ci-dessous illustre le diagramme de séquence de cas utilisation « Authentifier ».

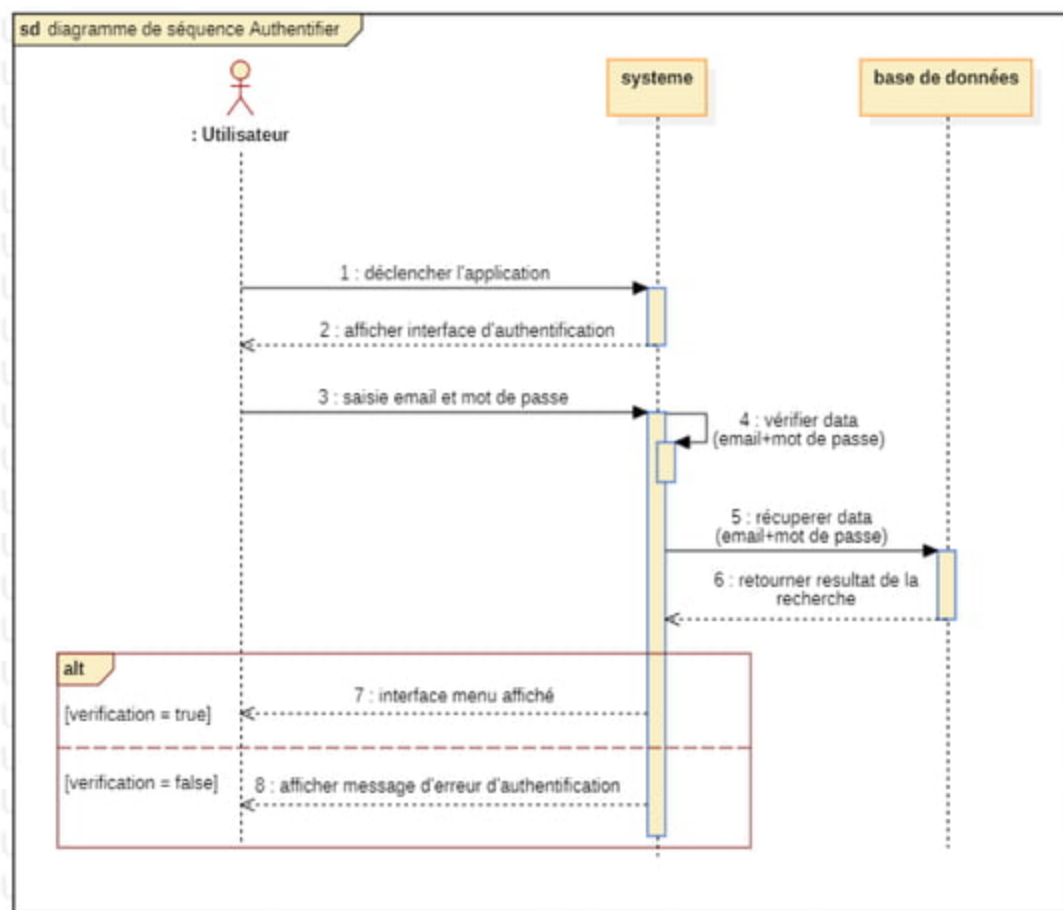


Figure 15. Diagramme de séquence cas utilisation « Authentifier »

4.1.2 Diagramme de séquence « Gérer Demande »

La figure ci-dessous illustre le diagramme de séquence pour le cas utilisation « Gérer Demande ».

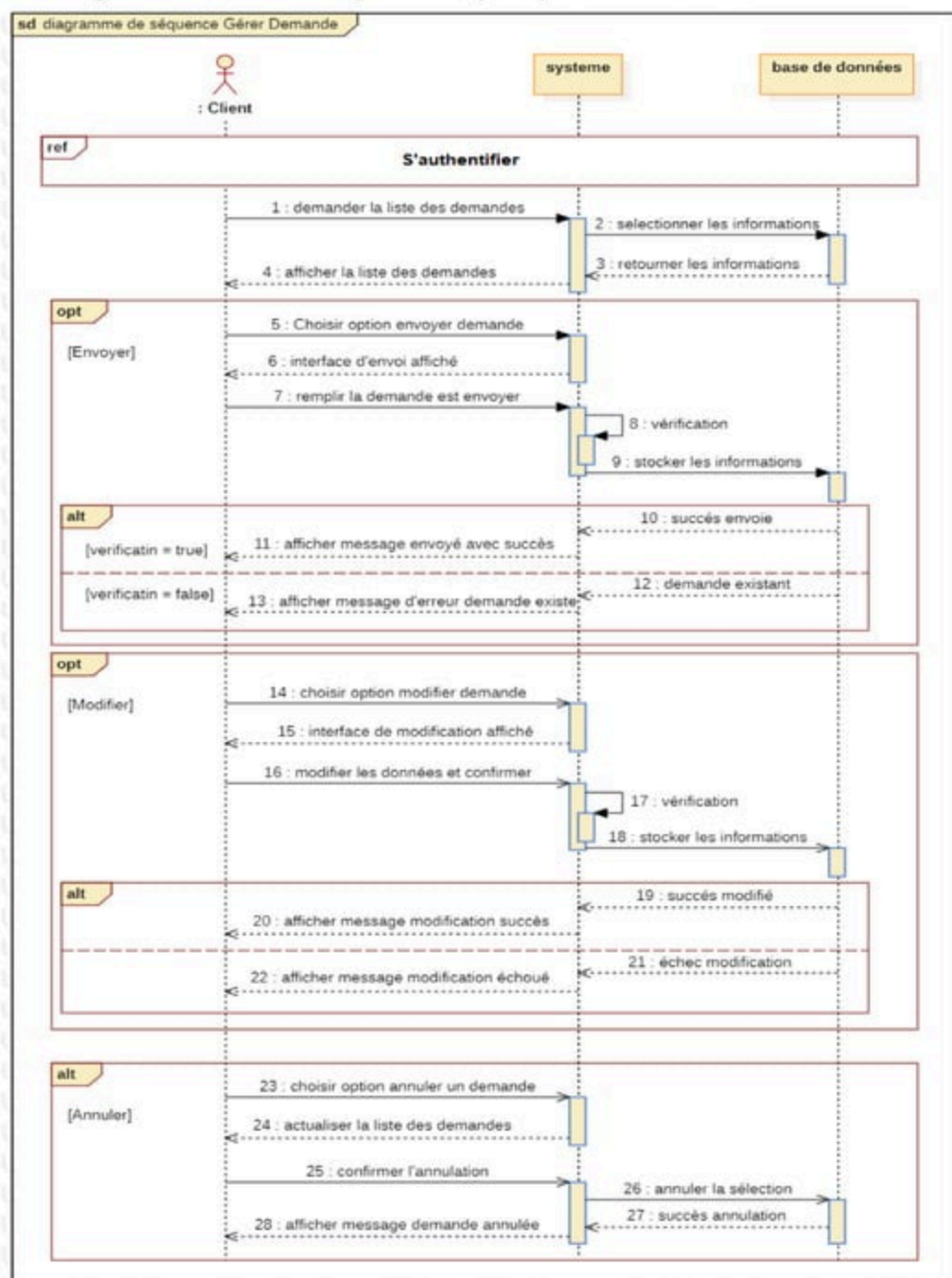


Figure 17. Diagramme de séquence pour le cas utilisation « Gérer Demande »

4.1.3 Diagramme de séquence « Gérer Profil »

La figure ci-dessous illustre le diagramme de séquence pour le cas utilisation « Gérer Profil ».

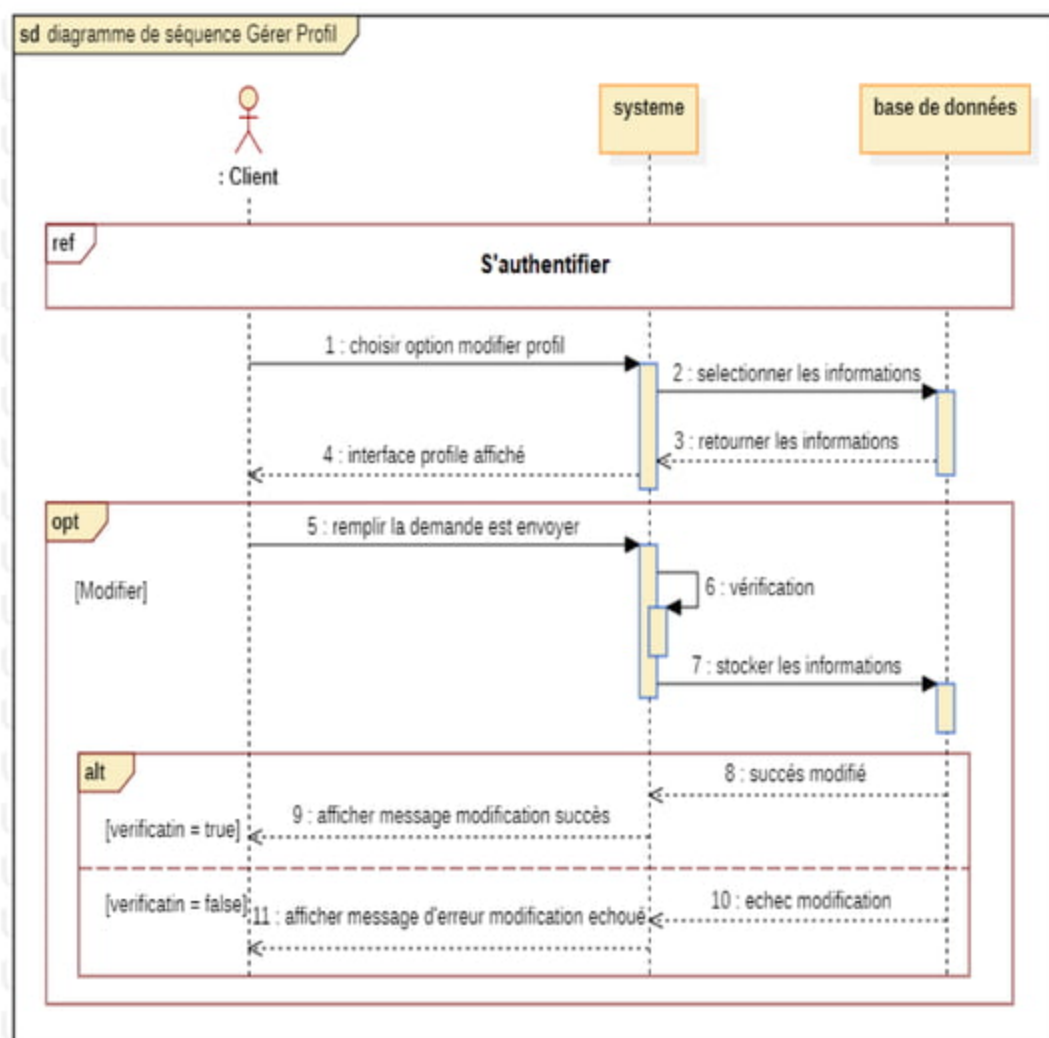


Figure 18. Diagramme de séquence pour le cas utilisation « Gérer Profil ».

4.1.4 Diagramme de séquence « Gérer Chauffeur »

La figure ci-dessous illustre le diagramme de séquence pour le cas utilisation « Gérer Chauffeur ».

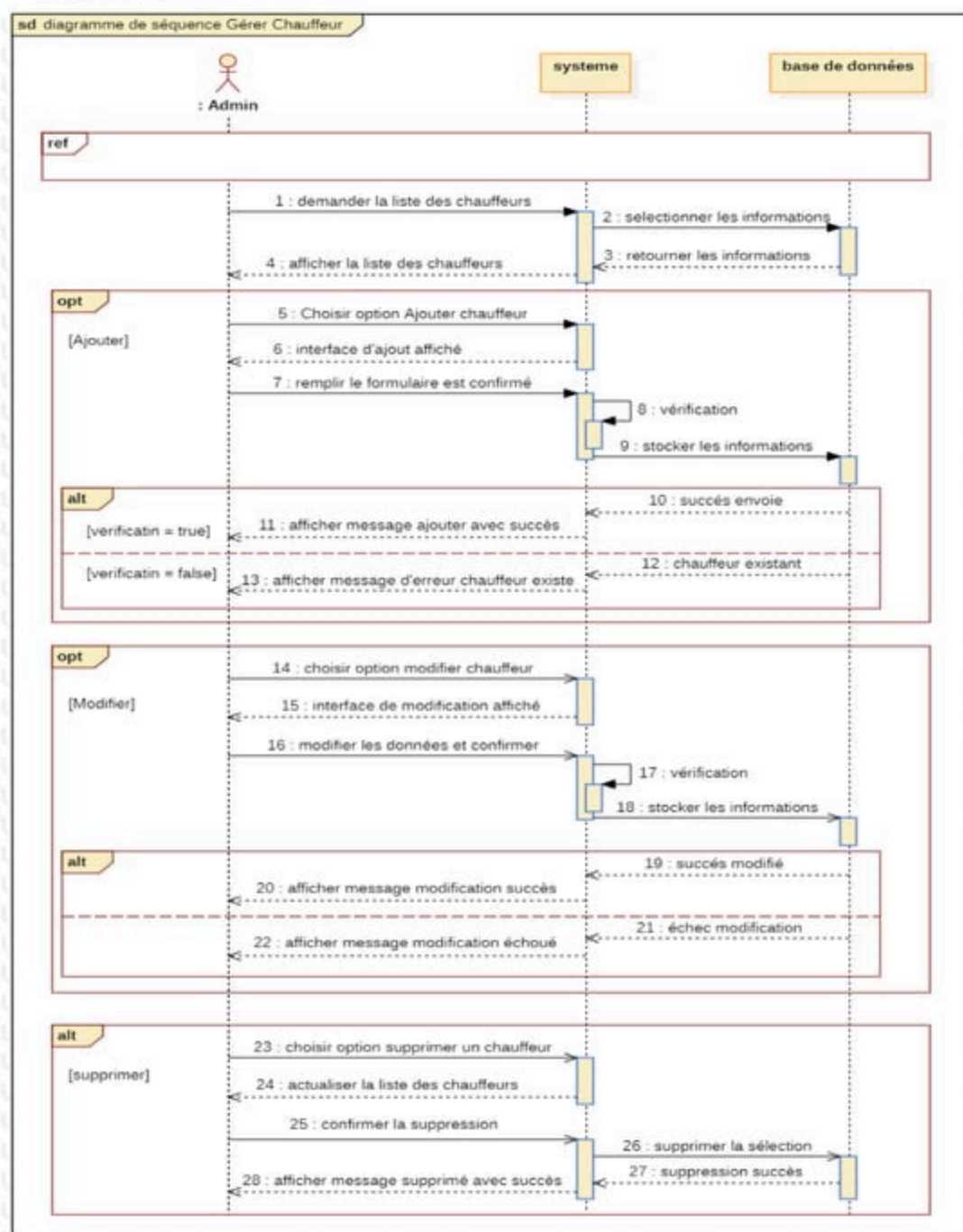


Figure 19. Diagramme de séquence pour le cas utilisation « Gérer Chauffeur ».

4.1.5 Diagramme de séquence « Gérer Camion »

La figure ci-dessous illustre le diagramme de séquence de cas utilisation « Gérer Camion ».

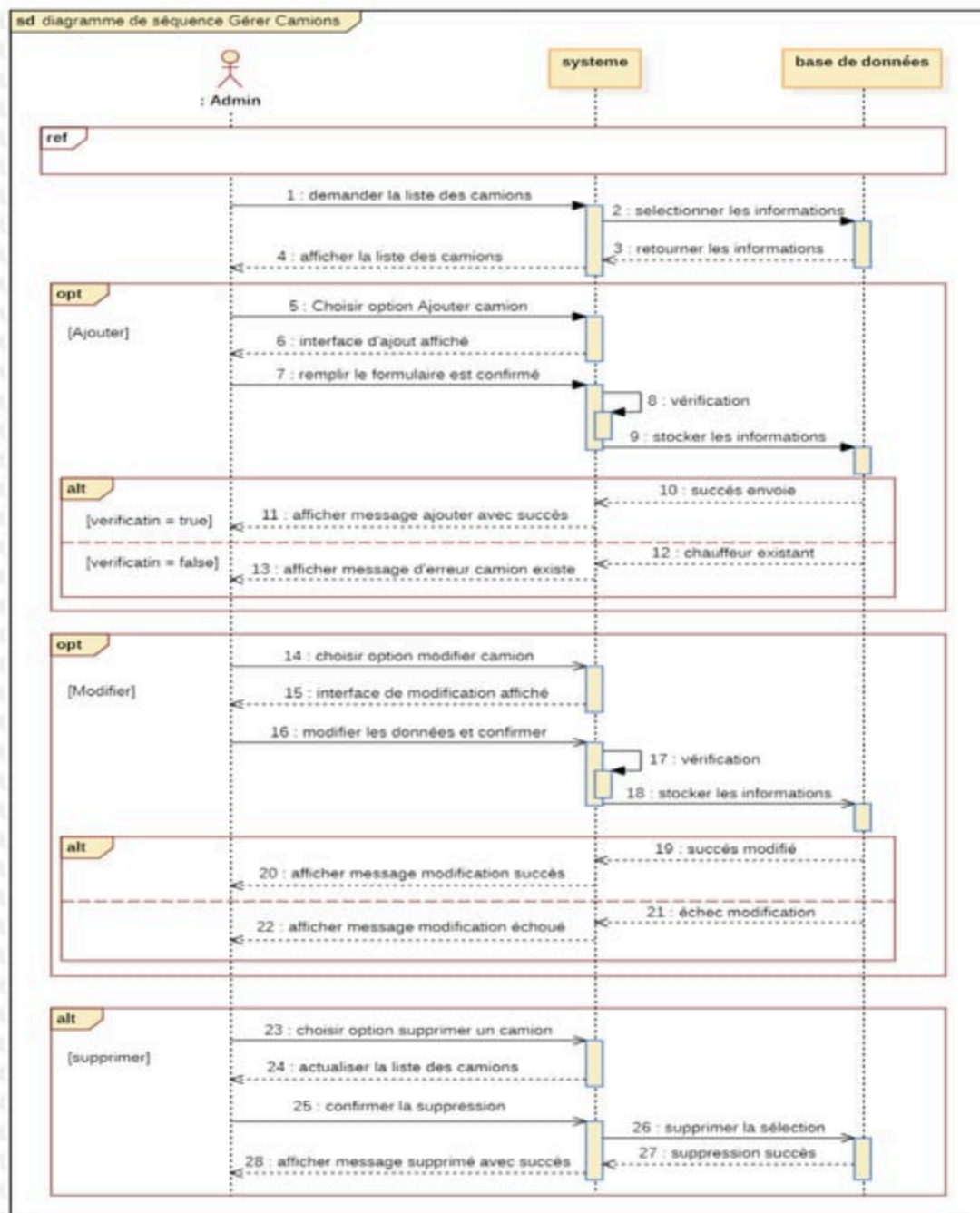


Figure 20. Diagramme de séquence de cas utilisation « Gérer Camion »

4.1.6 Diagramme de séquence « Gérer Véhicule »

La figure ci-dessous illustre le diagramme de séquence pour le cas d'utilisation « Gérer Véhicule »

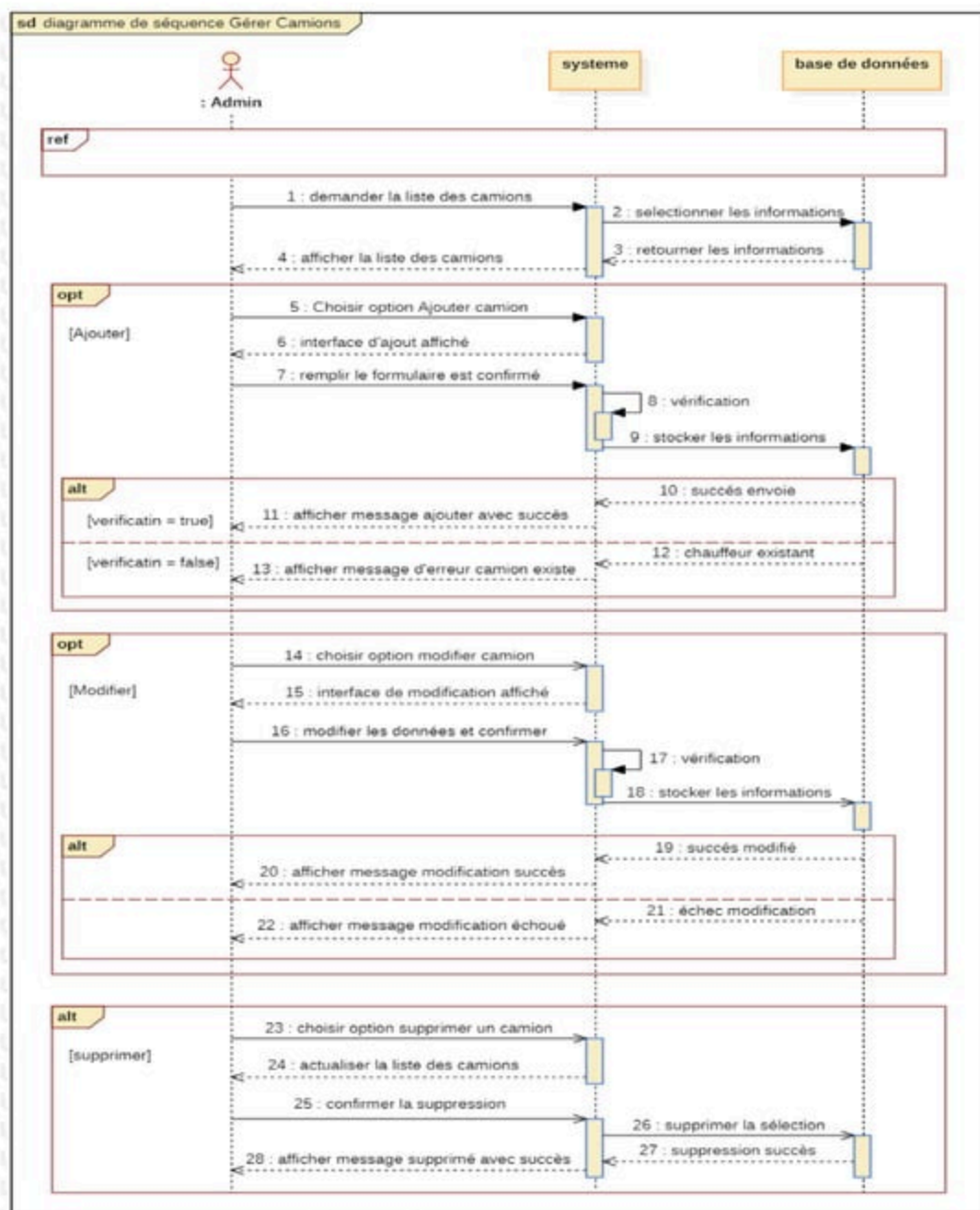


Figure 21. Diagramme de séquence pour le cas d'utilisation « Gérer Véhicule »

4.1.7 Diagramme de séquence « Traiter les Demandes des Remorquages »

La figure ci-dessous illustre le diagramme de séquence pour le cas d'utilisation

« Traiter les Demandes des Remorquages »

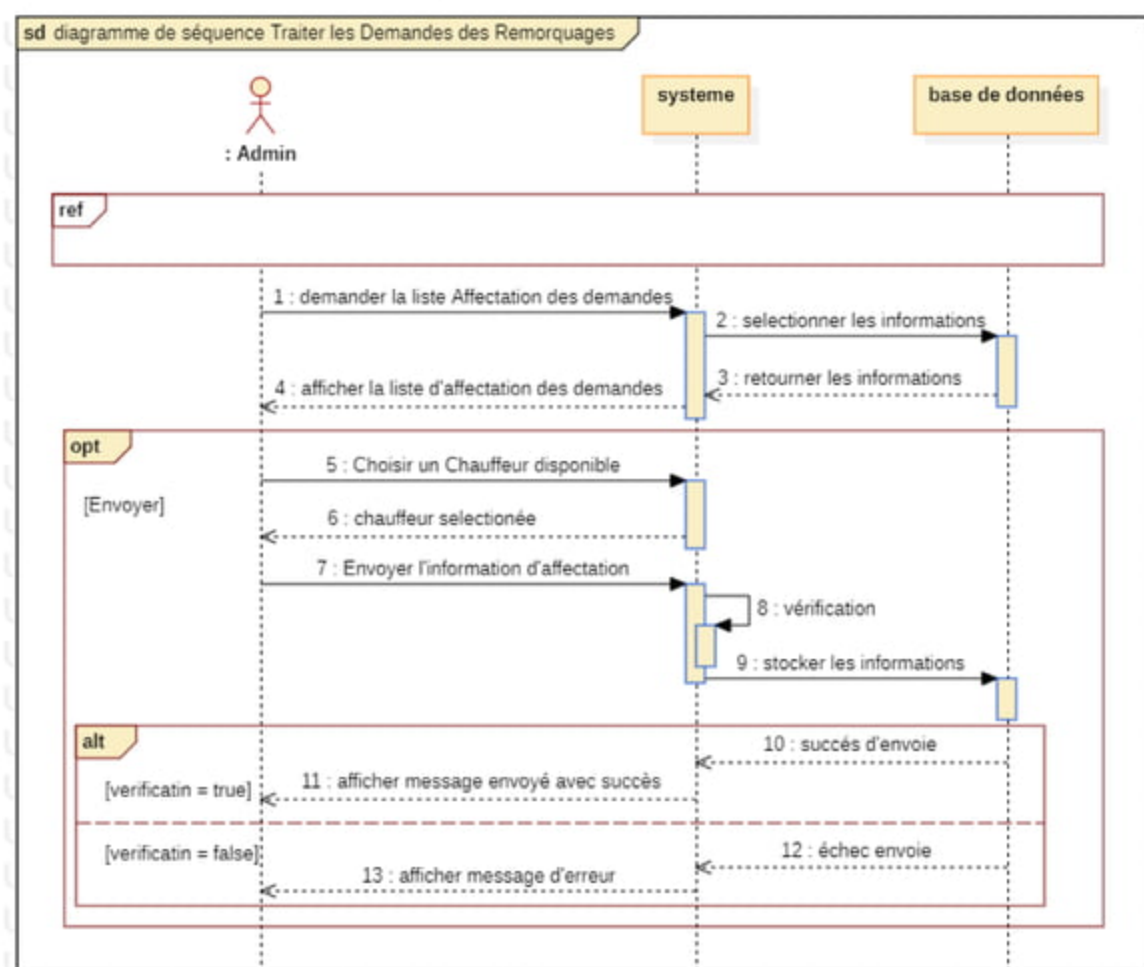


Figure 22. Diagramme de séquence pour le cas d'utilisation « Traiter les Demandes des Remorquages »

4.1.8 Diagramme de séquence « Modifier l'état d'un Camion »

La figure ci-dessous illustre le diagramme de séquence pour le cas d'utilisation « Modifier l'état d'un camion »

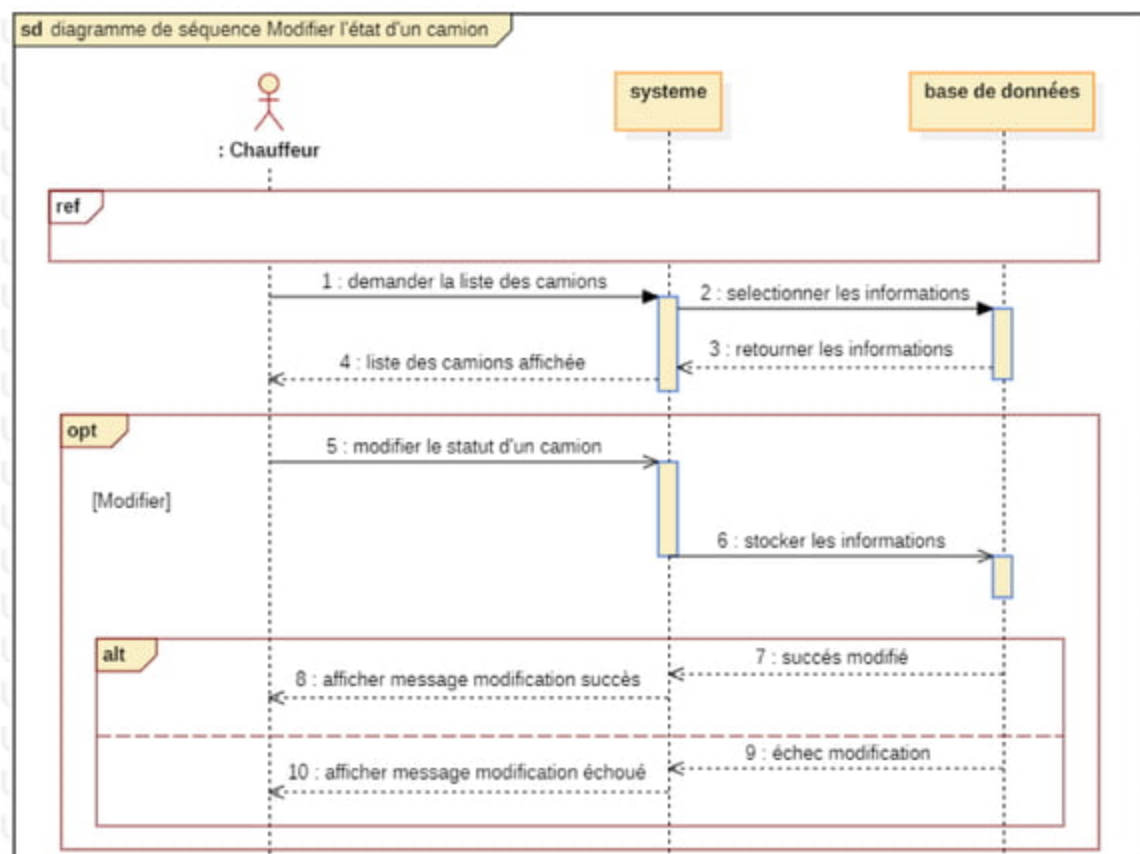


Figure 23. Diagramme de séquence pour le cas d'utilisation « Modifier l'état d'un Camion »

4.2 Diagrammes d'état/transition

Les diagrammes d'états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions.

Le passage d'un état vers l'autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une transition et entraînent le début immédiat d'une autre (automatique) [8].

4.2.1 Diagramme d'état/transition : objet remorquage

La figure ci-dessous illustre l'état transition de l'objet « remorquage ».

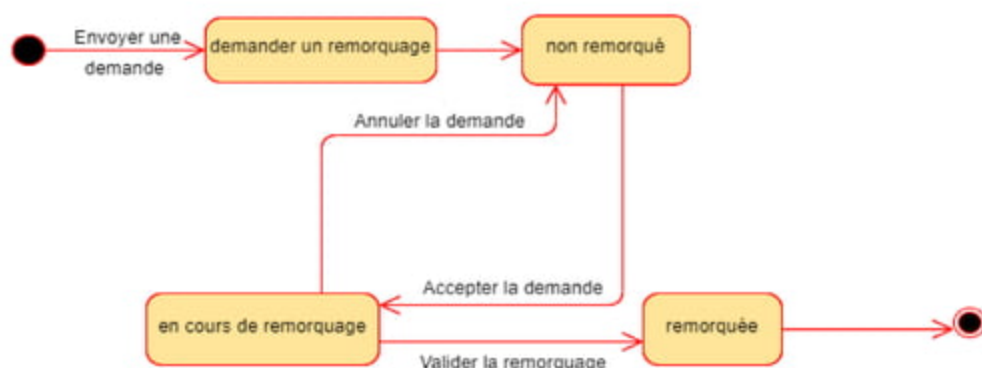


Figure 24. Etat transition de l'objet « remorquage »

4.2.2 Diagramme d'état/transition : objet demande

La figure ci-dessous illustre l'état transition de l'objet « demande ».

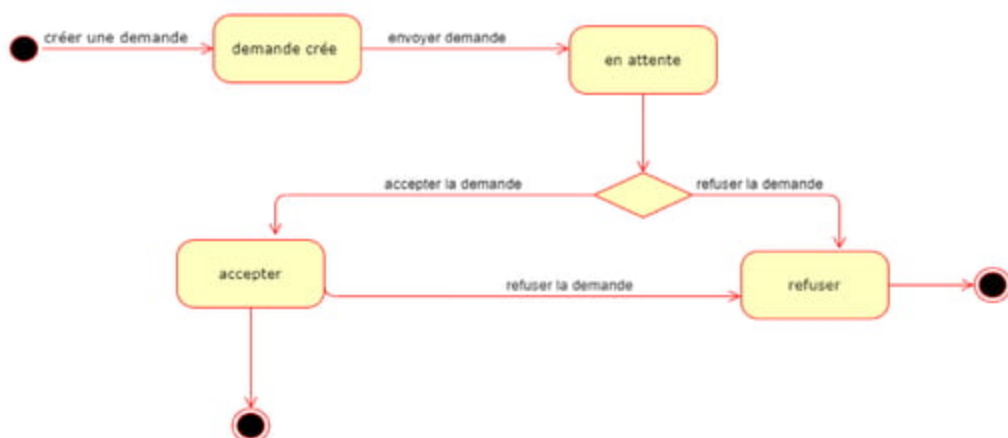


Figure 25. Etat transition de l'objet « demande »

4.3 Diagrammes de déploiement

Un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts [9]. La figure ci-dessous illustre le diagramme de déploiement.

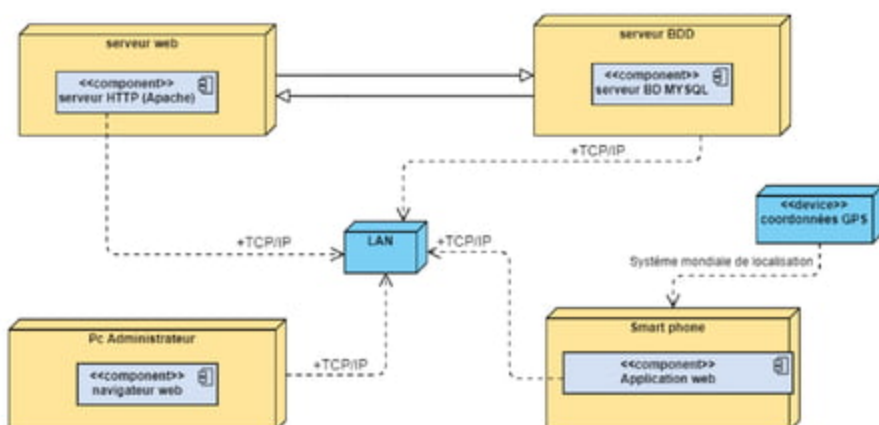


Figure 26. Diagramme de déploiement

5. Conclusion

Au cours de ce chapitre, nous avons détaillé le modèle conceptuel de notre projet qui va être utilisé dans le développement de l'application. Dans le prochain chapitre, nous allons Étudier la partie réalisation qui sera traduite par la présentation de l'environnement du travail et les interfaces réalisées.

Chapitre 4 : Réalisation

1. Introduction

Ce chapitre est réservé à la partie pratique de la réalisation de notre application. Il comporte la partie réalisation et la partie implémentation.

Dans ce chapitre, nous commençons par la présentation de l'environnement logiciel, les outils et les langages utilisés. Ensuite, nous allons présenter quelques interfaces de l'application avec des scénarios.

2. Spécification de l'environnement de travail

2.1 L'architecture utilisée

4. Architecture de l'application web

Nous avons développé la partie web en se basant sur un patron de conception MVC (Modèle, Vue, Contrôleur) comme l'illustre la figure 4.2.

MVC est modèle d'organisation du code telle que :

- Le modèle est chargé de gérer les données.
- La vue est chargée de la mise en forme pour l'utilisateur.
- Le contrôleur est chargé de gérer l'ensemble.

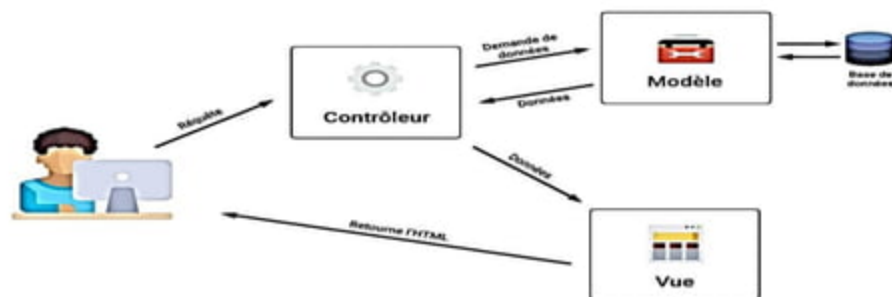


Figure 27. Architecture MVC

Le modèle MVC décrit une manière d'architecturer une application informatique en la décomposant en trois sous-parties :

- ✓ **La partie Modèle :** La couche Model représente la partie de l'application qui exécute la logique métier. Cela signifie qu'elle est responsable de récupérer les données, de les convertir selon des concepts chargés de sens pour votre application, tels que le traitement, la validation, l'association et beaucoup d'autres tâches concernant la manipulation des données.
- ✓ **La partie Vue :** La Vue retourne une présentation des données venant du model. Etant séparée par les Objets Model, elle est responsable de l'utilisation des informations dont elle dispose pour produire une interface de présentation de votre application. Par exemple, de la même manière que la couche Model retourne un ensemble de données, la Vue utilise ces données pour fournir une page HTML les contenant.
- ✓ **La partie Contrôleur :** La couche Controller gère les requêtes des utilisateurs. Elle est responsable de retourner une réponse avec l'aide mutuelle des couches Model et Vue.

2.2 Environnement matériel

Pour la réalisation de notre projet, nous avons utilisé deux ordinateurs portables pour pouvoir installer et utiliser les logiciels nécessaires pour la réalisation de l'application.

Le tableau ci-dessous représente les configurations des matériels utilisés.

Le tableau ci-dessous représente les configurations des matériels utilisés.


Nom de matériel	Description
Un PC LENOVO : 	<ul style="list-style-type: none"> ➤ Processeur : Intel Core i5 ➤ Mémoire vive : 6.00 Go ➤ Système d'exploitation : Windows 10

Tableau 4 : Environnement matériel

2.3 Spécification de l'environnement logiciel

Logiciel	Description
	Nicepage est un nouvel outil de construction de sites qui déclare être différent des autres outils les plus populaires sur le marché. Pour un maximum de liberté créative, ce constructeur fonctionne avec un éditeur avancé en drag-and-drop où il suffit de placer les éléments avec la souris.
	XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation [10].
	Visual Studio Code Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git[11].
	Word est un logiciel de traitement de texte couvre des notions assez différentes en pratique : un éditeur de textes interactif et un compilateur pour un langage de mise en forme des textes [12].
	Adobe Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur, lancé en 1990 sur MacOS puis en 1992 sur Windows. Édité par Adobe, il est principalement utilisé pour le traitement des photographies numériques [13].
	StarUML est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous licence modifiée de GNU GPL [14].
	Postman est un logiciel qui se focalise sur les tests des API. Il est devenu très populaire pour tester les Micro services, notamment grâce à sa simplicité et ses fonctionnalités très spécialisées [15].

Table 4.1 : Logiciels utilisés

2.4 Technologies et langages utilisés

Les langages et les technologies que nous avons utilisés pour le développement de notre application sont définis dans le tableau 4.2 :

Nom	Description
	Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems [16].
 ANGULAR 10	Angular est un framework MVC / MVVM côté client, développé en JavaScript, par Google libre et open-source qui permet d'améliorer, au même titre que JQUERY, la syntaxe de javascript ainsi que la productivité du développeur. Angular est basé sur la logique MVC. (modèle vue controller). Créé en 2009 par Google ce framework est devenu incontournable pour les applications web.
	Le HTML est un langage dit de « marquage » (de « structuration » ou de « balisage ») dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents [18].
	CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelées fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML. [19].
	JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web [20].
	jQuery est une bibliothèque JavaScript gratuite, libre et multiplateforme. Compatible avec l'ensemble des navigateurs Web (Internet Explorer, Safari, Chrome, Firefox, etc.), elle a été conçue et développée en 2006 pour faciliter l'écriture de scripts. Il s'agit du framework JavaScript le plus connu et le plus utilisé. Il permet d'agir sur les codes HTML, CSS, JavaScript et AJAX et s'exécute essentiellement côté client [21].
	PHP est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. PHP est un langage impératif orienté objet [17].



	<p>Bootstrap est un framework (une librairie) développé par Twitter et mis en open-source en février 2012.</p> <p>Ce langage utilise les principaux langages de développement web (HTML, CSS & Javascript). Il s'agit d'un code qui raccourcit différentes fonctionnalités (plugins utilisant la bibliothèque JQuery), permettant au développeur de gagner du temps et de réaliser simplement des codes complexes [22].</p>
	<p>Laravel est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet</p>

Table 4.2: Technologies et les langages utilisés

2.5 Techniques utilisées pour la géolocalisation




Nom	Description
<p>Google Maps</p> 	<p>Google Maps est un service gratuit de carte géographique et de plan en ligne. Le service a été créé par Google. Il s'agit d'une forme de géoportail. Lancé en 2004 aux États-Unis et au Canada et en 2005 en Grande Bretagne (sous le nom de Google Local), Google Maps a été lancé jeudi 27 avril 2006, simultanément en France, Allemagne, Espagne et Italie. [24].</p>
<p>Leaflet Maps</p> 	<p>Leaflet est une bibliothèque JavaScript libre de cartographie en ligne développée par Vladimir Agafonkin de CloudMade et de nombreux contributeurs. Elle est notamment utilisée par le projet de cartographie libre et ouverte OpenStreetMap2.</p>
<p>Map box</p> 	<p>Mapbox est une entreprise américaine spécialisée dans la cartographie en ligne. Elle fournit les cartes de sites, services et médias tels que Foursquare, Pinterest, The Financial Times, Etsy, Le Monde et Snapchat</p>

Table 4.3: Technique de géolocalisation

2.6 Techniques utilisées pour la Notification



Nom	Description
<p>Pusher</p> 	<p>Pusher est une API hébergée permettant d'ajouter rapidement, facilement et en toute sécurité des fonctionnalités évolutives en temps réel via WebSockets aux applications Web et mobiles. Il permet à un serveur d'envoyer des messages au client, et inversement.</p>
<p>Laravel Notification Channels</p> 	<p>Laravel prend en charge l'envoi de notifications sur divers canaux de distribution les notifications doivent être des messages courts et informatifs qui informent les utilisateurs de quelque chose qui s'est produit dans votre application</p>

Table 4.3: Technique de la Notification

2.7 Protocol et format de données

2.7.1 Protocol de communication

Tout au long de la phase de développement de notre application, nous avons utilisé le protocole de communication « HTTP » pour le transfert de données entre le client et le serveur. En fait, le protocole « HTTP » est le protocole le plus utilisé sur Internet. Ce protocole comprend plusieurs types des requête (GET, POST, HEAD, PUT, DELETE) parmi lesquelles nous avons utilisé deux types :

- Les requêtes GET permettent de demander une ressource située à un URL spécifiée.
- Les requêtes POST permettent d'envoyer des données à un URL spécifiée.

2.7.2 Format de données

JSON (JavaScript Object Notation) est un format de données textuelles. Il offre une représentation structurée de l'information et il est simple à mettre en œuvre par un développeur.

JSON est utilisé dans notre application comme un intermédiaire entre l'application et le script PHP pour interroger et récupérer les données depuis de la base de données [27]

3. Réalisation : les interfaces

L'interface homme/machine (IHM) représente un élément indispensable pour l'utilisation de tout système d'information. En effet, le présent système comprend des interfaces interactives, lisibles et faciles à manipuler.

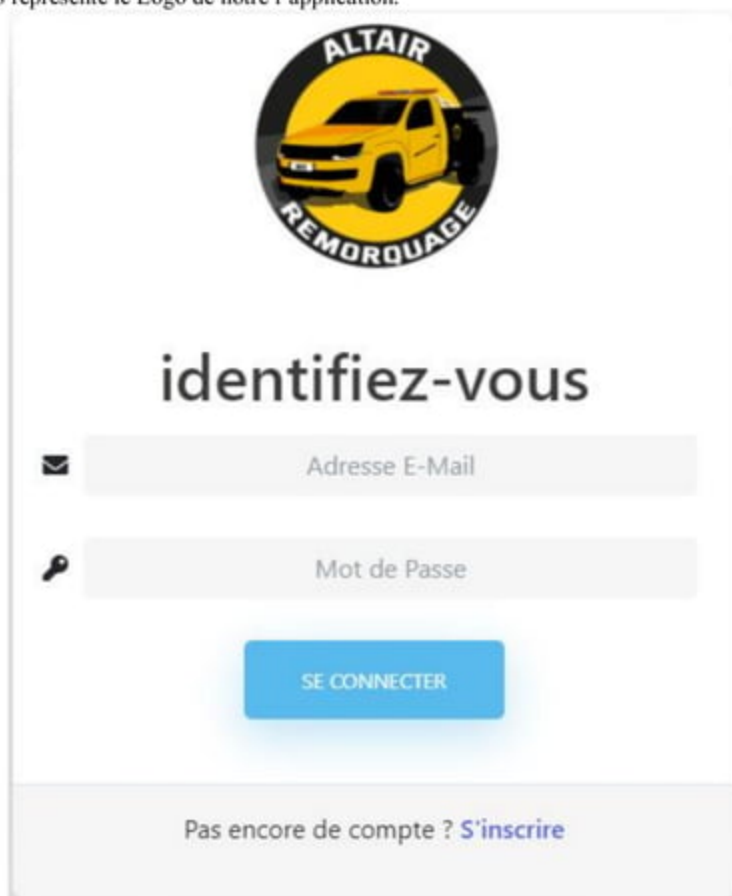
Dans cette partie, nous présentons les interfaces de notre application web.

3.1 Partie Administrateur :

L'application Web développée est destinée à l'administrateur

3.1.1 Interface login de l'administrateur

La figure 28 représente le Logo de notre l'application.



ALTAIR
REMORQUAGE

identifiez-vous

✉ Adresse E-Mail

🔑 Mot de Passe

SE CONNECTER

Pas encore de compte ? [S'inscrire](#)

Figure 28 : Interface Login Administrateur

3.1.2 Interface de tableau de bord de l'administrateur

Une fois l'administrateur est authentifié, le tableau de bord s'affiche (figure 29) contient tous les statistiques de l'application (nombres d'utilisateurs, nombres des véhicules...).

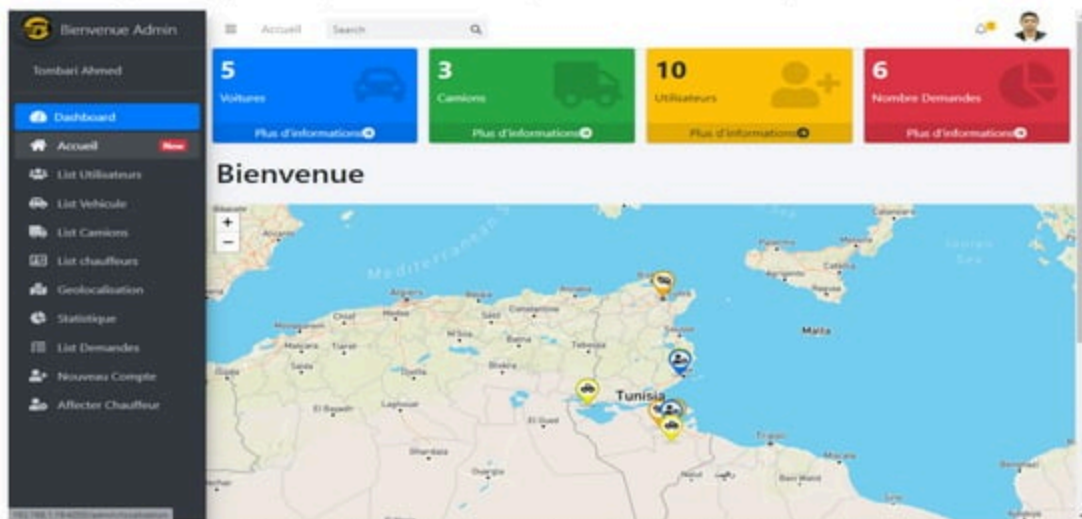


Figure 29: Interface tableau de bord pour Administrateur

3.1.3 Interface de List utilisateurs de l'administrateur

L'administrateur peut consulter la liste de tous les utilisateurs (client et chauffeurs) stockés dans la base de données. Ils sont affichés dans un tableau avec un système de pagination et une barre de recherche comme l'illustre la figure 30.

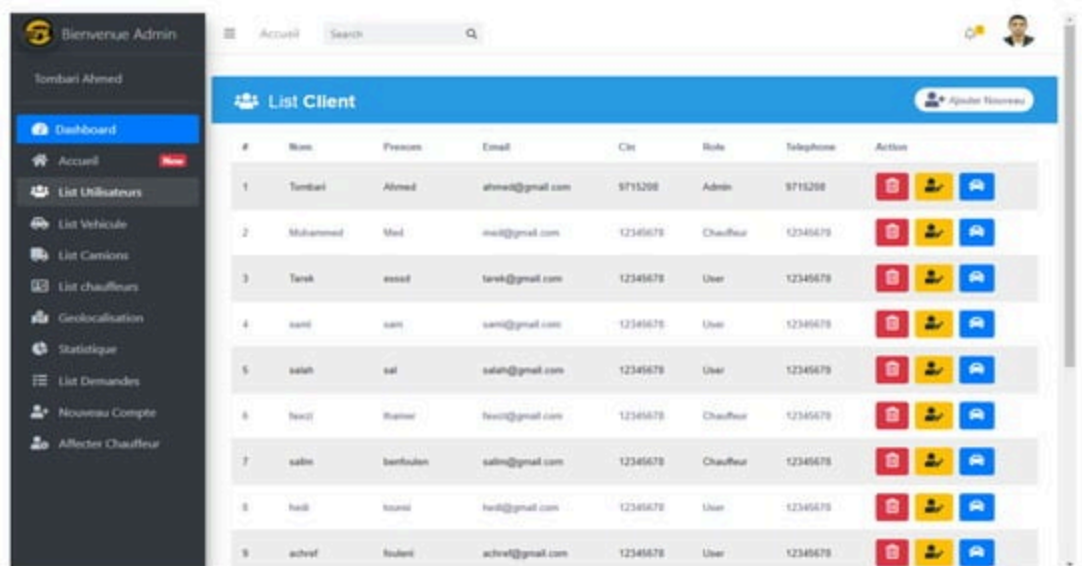


Figure 30 : Interface de List de l'administrateur

3.1.4 Interface de List véhicules de l'administrateur

La figure 31 représente la liste des Véhicules stockés dans la base de données. L'administrateur peut consulter les détails de chaque Véhicules (matricule, marque,...).

The screenshot shows the 'List Vehicle' page of an admin dashboard. The left sidebar contains a menu with options like Dashboard, Accueil, List Utilisateurs, List Vehicule, List Camions, List chauffeurs, Geolocalisation, Statistique, List Demandes, Nouveau Compte, and Affecter Chauffeur. The main content area has a header 'List Vehicle' and a table with the following data:

#	Marque	Matricule	Date Sale	Date Paiement	Tarif	Action
1	12345678	2104567	2021-11-10 23:34			[Edit] [Delete] [Add]
2	7754321	2113576	2021-11-10 23:43			[Edit] [Delete] [Add]
3	20212222	2903214	2021-11-11 12:48			[Edit] [Delete] [Add]
4	20222222	2791456	2021-11-11 12:49			[Edit] [Delete] [Add]
5	22220529	2301987	2021-11-11 12:53			[Edit] [Delete] [Add]

Below the table, it says 'Affichage 5 hors de 25 entrées'. At the bottom, there is a map of North Africa with a red location pin.

Figure 31 : Interface de List véhicules de l'administrateur

3.1.5 Interface de List camions de l'administrateur

L'administrateur peut consulter la liste de tous les camions stockés dans la base de données. Ils sont affichés dans un tableau comme l'illustre la figure 32.

The screenshot shows the 'List Camions' page of an admin dashboard. The left sidebar is the same as in Figure 31. The main content area has a header 'List Camions' and a table with the following data:

#	Numero	Matricule	Marque	Statut	Action
1	0012345	12034567	mercedes	En service	[Edit] [Delete] [Add]
2	0030987	2103054	iveco	En service	[Edit] [Delete] [Add]
3	004568	2004567	volvo	Disponible	[Edit] [Delete] [Add]

Below the table, it says 'Affichage 3 hors de 25 entrées'. At the bottom, there is a map of North Africa with a red location pin.

Figure 32 : Interface de List camions de l'administrateur

3.1.6 Interface de List chauffeurs de l'administrateur

La figure 33 représente la liste des chauffeurs stockés dans la base de données. L'administrateur peut consulter les détails de chaque Chauffeur (nom, prénom, email, cin...).

#	Nom	Prénom	Email	Cin	Role	Telephone	Action
2	Mohammad	Maid	meid@gmail.com	12345678	Chauffeur	12345678	[Icons: Delete, Edit, Add]
6	Isaaci	Thomas	isaaci@gmail.com	12345678	Chauffeur	12345678	[Icons: Delete, Edit, Add]
7	salim	benfouda	salim@gmail.com	12345678	Chauffeur	12345678	[Icons: Delete, Edit, Add]

Affichage 3 hors de 25 entrées

Figure 33 : Interface de List chauffeurs de l'administrateur

3.1.7 Interface de Historique des demandes des remorquages de l'administrateur

La figure 34 représente l'espace de liste de demandes (marque, panne, état, emplacement géographique...).

Id	Marque	Panne	Date	Etat	Chauffeur
1	audi r8	Panne Carburant	2021-11-10 23:45	Remorquage Terminé	[Icon: User]
2	Mercedes	Casseion	2021-11-10 23:45	Remorquage Terminé	[Icon: User]
3	Mercedes	Panne Carburant	2021-11-11 11:07	En Attente	[Icon: User]
4	audi r8	Remorquage	2021-11-11 11:07	Remorquage Terminé	[Icon: User]
5	ford	Accident	2021-11-11 12:53	Accepté	[Icon: User]
6	chevrolet	Panne Carburant	2021-11-11 22:41	En Attente	[Icon: User]

Affichage 5 hors de 25 entrées

Figure 34 : Interface de Historique des remorquages de l'administrateur

3.1.8 Interface de List de nouveau utilisateurs de l'administrateur

L'administrateur peut consulter la liste de tous les nouveaux utilisateurs stockés dans la base de données. Ils sont affichés dans un tableau comme l'illustre la figure 35

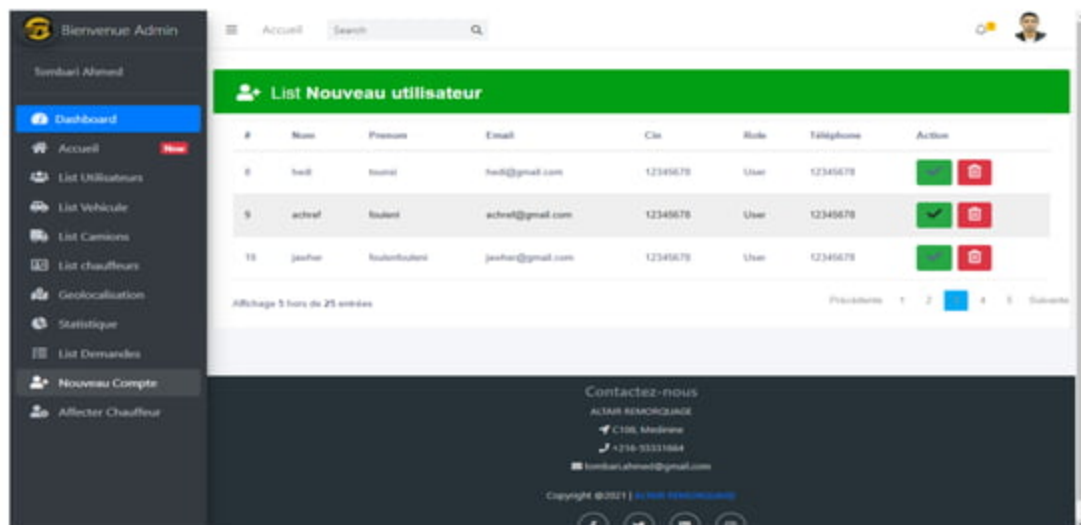


Figure 35 : Interface de List de nouveau utilisateurs de l'administrateur

3.1.9 Interface d'affectation des chauffeurs a des demandes

L'administrateur peut consulter la liste de tous les nouveaux utilisateurs stockés dans la base de données. Ils sont affichés dans un tableau comme l'illustre la figure 36

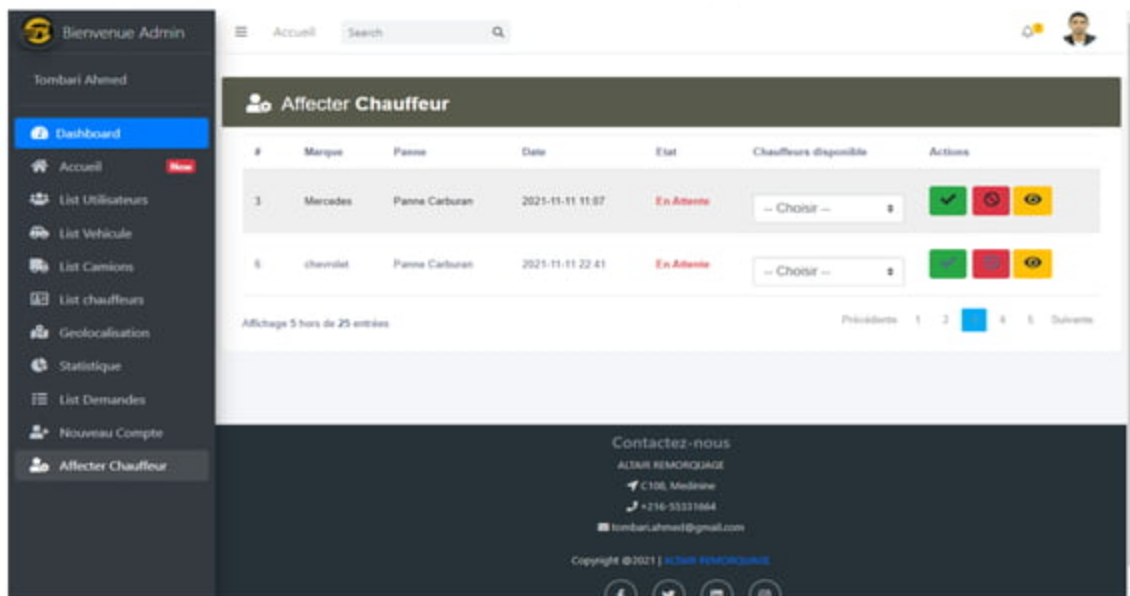


Figure 36 : Interface d'affectation de chauffeurs à la demande de l'administrateur

3.2 Partie Client :

L'application Web développée est destinée à Client.

3.2.1 Interface login de Client

La figure 37 représente l'interface de connexion de l'utilisateurs (Client).

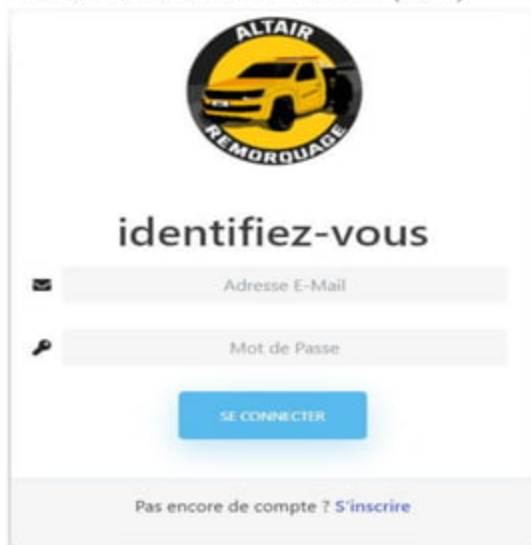


Figure 37 : interface de connexion

La figure 38 représente l'interface de création du compte.



Figure 38 : interface de création de compte

La figure 39 représente l'interface de Menu de client.

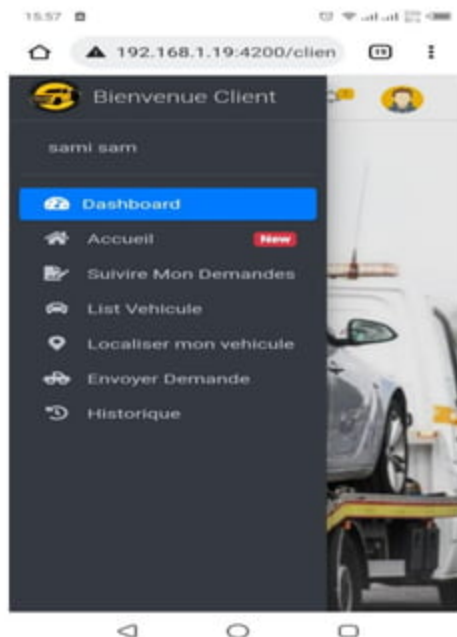


Figure 39 : interface Menu de client

La figure 40 représente l'interface de suivi des demandes.

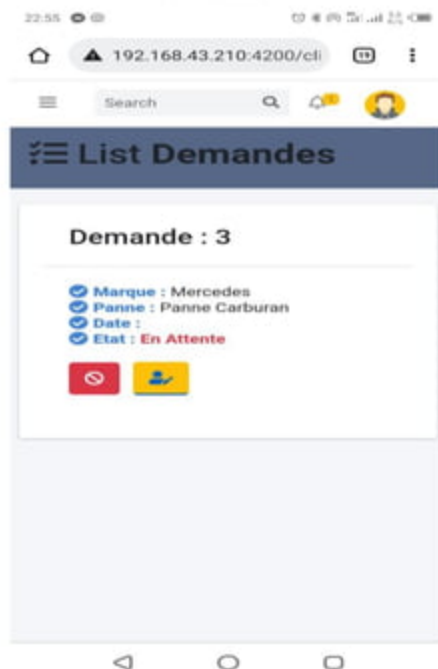


Figure 40 : interface de suivi de demande

La figure 41 représente l'interface de List de véhicules.

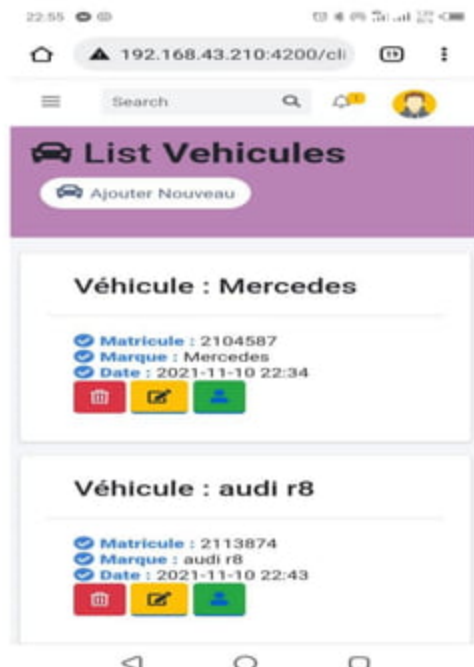


Figure 41 : interface List des véhicules

La figure 42 représente l'interface d'ajout d'un véhicule.



Figure 42 : Formulaire d'ajouter un véhicule

La figure 43 représente l'interface interface d'envoyer des demandes.

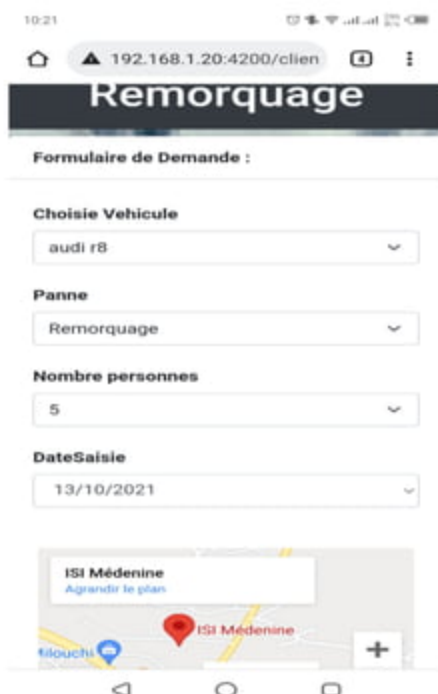


Figure 43 : interface d'envoi des demandes

La figure 44 représente l'interface interface de modifier profil.

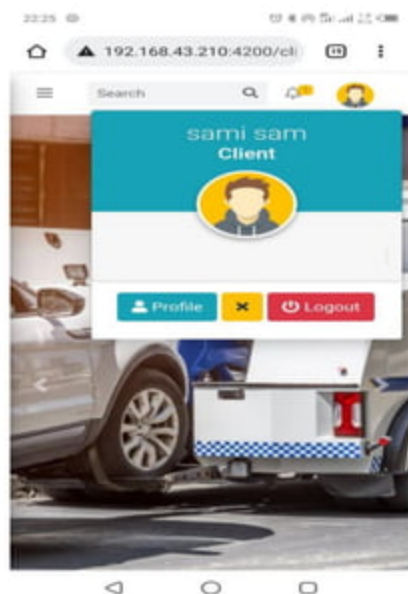
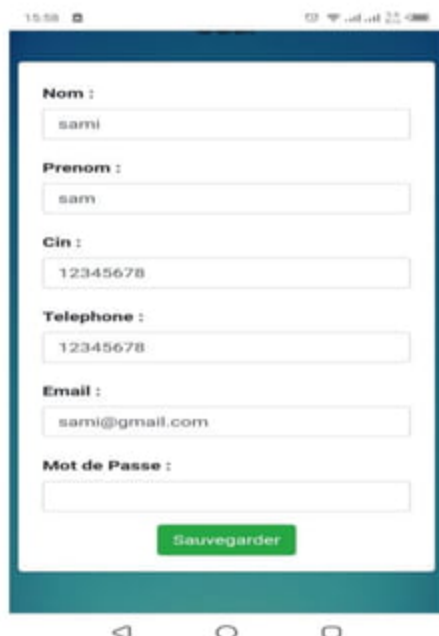


Figure 44 : interface de modifier profil

La figure 45 représente l'interface de Formulaire de modifier de profil.



15:58

Nom :
sami

Prenom :
sam

Cin :
12345678

Telephone :
12345678

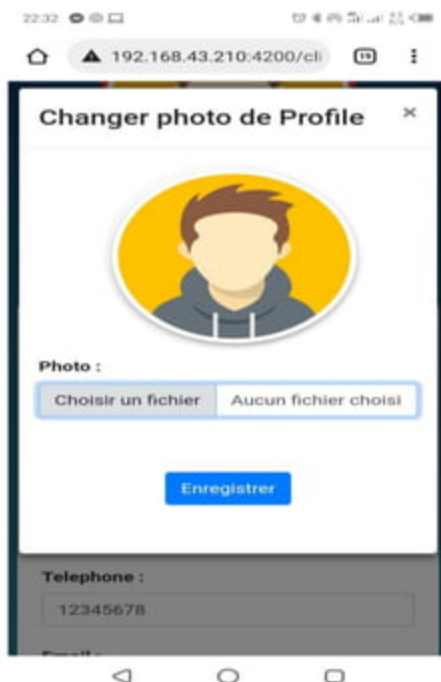
Email :
sami@gmail.com

Mot de Passe :

Sauvegarder

Figure 45 : Formulaire de modifier de profil

La figure 46 représente l'interface de changer de photo de profil.



22:32

192.168.43.210:4200/cli

Changer photo de Profile

Photo :

Choisir un fichier Aucun fichier choisi

Enregistrer

Telephone :
12345678

Figure 46 : interface de changement de photo de profil

La figure 47 représente l'interface d'Histoire des demandes Terminée

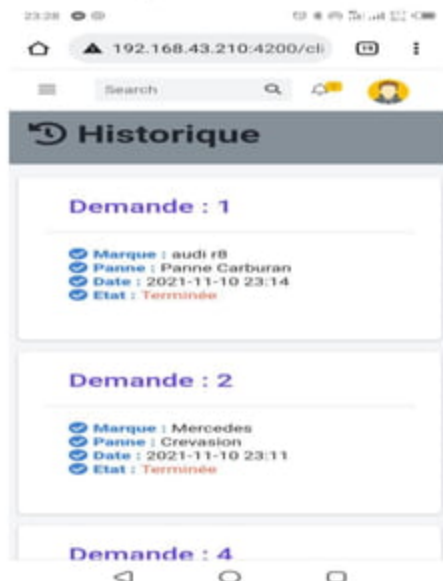


Figure 47 : interface d'Histoire des Demandes Terminée

3.3 Partie Chauffeur :

L'application Web développée est destinée à Client.

3.3.1 Interface login de Chauffeur

La figure 48 représente l'interface de connexion de l'utilisateurs (Chauffeur).



Figure 48 : interface de connexion

La figure 49 représente l'interface d'accueil de chauffeur.

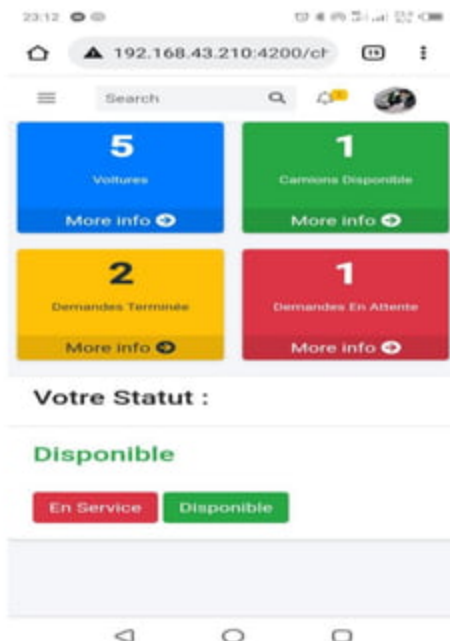


Figure 49 : interface d'accueil

La figure 50 représente l'interface Menu de chauffeur.

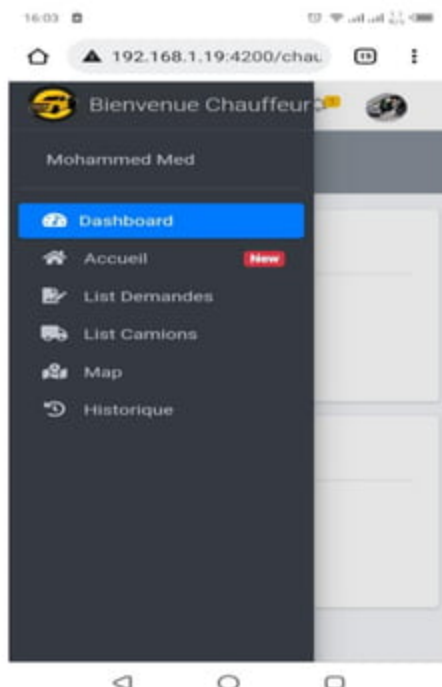


Figure 50 : interface Menu

La figure 51 représente l'interface liste des demandes.



Figure 51 : interface List de demande

La figure 52 représente l'interface de modification de l'état de camion.

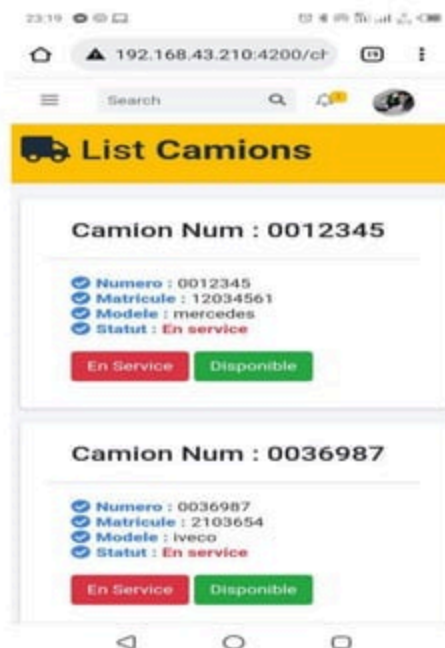


Figure 52 : interface de modification de l'état de camion

Les figures 53 représente l'interface de modification de profil.

Figure 53 : interface modifier profil

Les figures 54 représente la Formulaire de modification de profil

Figure 54 : formulaire de modifier profil

Les figures 55 représente la l'interface de changement de photo de profil

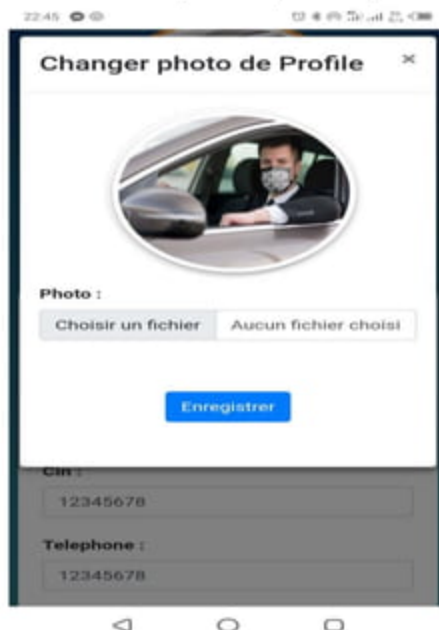


Figure 55 : changer photo de profil

Les figures 56 représente là l'interface d'Histoire de remorquages Terminée

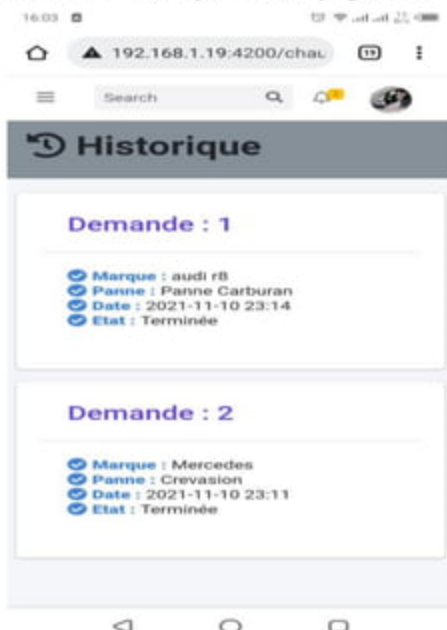


Figure 56 : Historique de Remorquages Terminée

4. Conclusion

Au cours de ce chapitre, nous avons présentés quelques interfaces de l'application web et mobile montrant en valeur la fonctionnalité de l'application.

Conclusion Générale

En guise de conclusion, ce projet m'a donné la possibilité de découvrir de nouvelles approches de développement dédiées aux web et d'utiliser des nouvelles technologies. Ce projet m'a permis aussi d'approfondir mes connaissances, acquises tout au long de ma formation universitaire, dans les langages de programmation. L'application « Système de remorquage » qui assure la disponibilité et facilite la communication et l'exploitation des services d'assistance voiture en panne pour les clients. La réalisation de ce travail a nécessité au début de faire une collecte d'informations qui nous a permis à son tour de préparer une étude théorique au cours de laquelle nous avons étudié et critiqué l'existant. Cette étude nous a amené à proposer « remorquage de véhicule » comme solution. La phase de conception détaillée a couvert tous les aspects conceptuels : statique et dynamique de l'application en se basant sur les différents diagrammes UML. Lors de la réalisation du projet, nous avons utilisé les technologies les plus adaptés à notre choix technique dont nous avons énuméré dans le chapitre des réalisations. Enfin, nous avons enchainé avec une série de captures d'écran illustrant le fonctionnement de l'application. A l'issue de ce travail, plusieurs perspectives peuvent être envisagées comme l'ajout d'un module intelligent basé sur l'évaluation des profils des utilisateurs (les commentaires et la note du profil) pour aider les clients et les chauffeurs à la prise de décision.

Bibliographie

- [1] Site web de Plan Bleu, <https://planbleu.org/>
- [2] Site web de <https://app.towbook.com/>
- [3] Site web de <https://www.route4me.com/>
- [4] <https://www.lucidchart.com/pages/fr/langage-uml>
- [5] <https://laurent-audibert.developpez.com/Cours-UMLdiagramme-classes>
- [6] <https://laurent-audibert.developpez.com/Cours-UML/diagramme-etats-transitions>
- [7] <https://www.json.org/json-fr.html>
- [8] <https://mathias-seguy.developpez.com/tutoriels/android/utiliser-retrofit/>
- [9] <https://www.techno-science.net/definition/375.html>
- [10] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203587-jquery-definition/>
- [11] <https://agency-inside.com/2016/06/definition-webmarketing-bootstrap/>
- [12] <https://angular.io/>
- [13] <https://laravel.com/>
- [14] <https://www.commentcamarche.net/contents/498-html-langage> Projet de fin d'étude
Bibliographie 2019/2020 Page 52
- [15] <http://glossaire.infowebmaster.fr/css/>
- [16] https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [17] <http://inf1410.telug.ca/telugDownload.php?file=2014/01/INF1410-PresentationStarUML>
- [18] <https://openclassrooms.com/fr/courses/4668056-construisezdes-microservices/5123020-testez-votre-api-grace-a-postman>
- [19] https://java.com/fr/download/help/whatis_java.html
- [20] <https://www.php.net/manual/fr/intro-what-is.php>
- [21] <https://www.africmemoire.com/part.5-ii-3-1-le-langage-de-modelisation-uml949.html>
- [22] <https://www.encyclopedie.fr/definition/Xampp>
- [23] <https://framalibre.org/content/visual-studio-code>
- [24] <https://sites.google.com/site/googsitedetest/word>
- [25] <https://leafletjs.com/>
- [26] <https://www.mapbox.com/>
- [27] <https://pusher.com/>

Acronymes

API : Application Programming Interface.

CSS : Cascading Style Sheet.

HTML : Hypertext Markup Language.

HTTP : HyperText Transfer Protocol.

JSON : JavaScript Object Notation.

MYSQL : My Structured Query Language.

NOSQL : Not Only Structured Query Language

PHP : Hypertext Preprocessor.

UML : Unified Modeling Language

Résumé

Ce projet a été réalisé au sein de la société Altair Solution pour l'obtention du diplôme de Mastère professionnel en Ingénierie des logiciels et des connaissances à l'institut supérieur d'informatique de Médénine. L'objectif de ce projet est de concevoir et développer une application web pour le système de remorquage. La réalisation de ce projet est basée sur le Framework « Angular » qui est basé sur le langage Typescript dans la partie Frontend, le format léger d'échange de données et Laravel dans la partie back-end avec un système de gestion de base de données MySQL.

Mots clés : Laravel, JSON, Angular ...

Abstract

This project was carried out within the company Altair Solution to obtain the professional master's degree in software and knowledge engineering at the higher institute of computer science in Medenine. The objective of this project is to design and develop a web application for the towing system. The realization of this project is based on the "Angular" Framework which is based on the Typescript language in the Frontend part, the light data exchange format and Laravel in the back-end part with a database management system MySQL.

Keywords : Laravel, JSON, Angular ...

ملخص

تم تنفيذ هذا المشروع ضمن شركة Altair Solution للحصول على درجة الماجستير المهنية في هندسة البرمجيات والمعرفة من المعهد العالي لعلوم الحاسوب بمدنين. الهدف من هذا المشروع هو تصميم وتطوير تطبيق ويب لنظام السحب. يعتمد تحقيق هذا المشروع على إطار العمل "Angular" الذي يعتمد على لغة الأنواع في الجزء الأمامي، وتنسيق تبادل البيانات الخفيف Laravel في الجزء الخلفي مع نظام إدارة قاعدة البيانات MySQL.

