

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



MÉMOIRE DE MASTER INFORMATIQUE
Spécialité Génie Logiciel

Thème:

Conception et réalisation d'une application web
d'un module de planification d'un TMS. Cas
d'étude : Bejaia Logistique.

Présenté par:

Mlle MERABET Nedjima
Mlle BOUYAHMED Maïssa

Soutenue le 14 octobre 2021 devant le jury composé de

Encadrant :	Mr. AMROUN Kamal	Professeur
Président de jury :	Mme. ALOUI Soraya	M.C.A
Examineur :	Mr. SAADI Mustapha	M.C.A

Année universitaire : 2020/2021

Remerciements

En guise de reconnaissance, on tient à témoigner nos sincères remerciements à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce modeste travail.

On tient tout d'abord à remercier Mr HERROUG Imade chef de département informatique à Bejaia Logistique pour la qualité de son encadrement tout au long du déroulement de notre stage, pour sa disponibilité, sa rigueur et son partage d'expertise.

Nos sincères gratitudee à Mr AZOUG Hocine, directeur commercial à Bejaia Logistique, de nous avoir accepté pour effectuer ce stage de projet de fin d'études, pour son temps précieux, sa disponibilité, et ses précieux conseils durant tout ce temps.

On remercie également toute l'équipe des planificateurs pour leur accueil en particulier Mr MEHIDI Mohand qui nous a énormément aidé à comprendre le rôle des planificateurs au sein d'une entreprise de transport.

On tient à remercier l'ensemble du personnel de Bejaia Logistique pour leurs conseils plein de sens et pour le suivi et l'intérêt qu'ils ont porté à nos travaux.

Enfin, dans l'impossibilité de citer tous les noms, nos sincères remerciements vont à tous ceux et celles, qui de près ou de loin, ont permis par leurs conseils et leurs compétences la réalisation de ce mémoire.

Dédicaces

Je dédie ce modeste travail

A mon père Zahir MERABET et ma mère Melaaz BENLARBI pour tout leur amour, leurs sacrifices, leur confiance, et leur soutien tout au long de mon parcours d'études.

A mes frères Riad, Abderraouf, Rayane et à ma sœur Maya.

A l'âme de mes chers grands-parents Djamila et Youcef qui m'ont quitté durant la réalisation de ce projet.

A mes grands-parents et mes tantes de la famille Benlarbi pour leur soutien et encouragement.

MERABET Nedjima

Dédicaces

A ma très chère mère

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force, ton amour a fait de moi ce que je suis aujourd'hui.

A mon très cher père

Je dédie ce travail marquant de ma vie à la mémoire de mon père disparu trop tôt. J'espère que, du monde qui est sein maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'une fille qui a toujours priée pour le salut de son âme. Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde!

A mon chère frère Bouyahmed Lounis

Ta bonté, ton précieux soutien, ton encouragement tout au long de mes années d'études, ton amour et ton affection, ont été pour moi l'exemple de persévérance. Que ce travail soit l'expression de mon estime pour toi et que Dieu te protège.

A ma très chère sœur Bouyahmed Sarah

Aucune dédicace ne peut exprimer mon amour et ma gratitude de t'avoir comme sœur. Je ne pourrais jamais imaginer la vie sans toi, tu es la sœur qui assure son rôle comme il faut, je n'oublierais jamais ton encouragement et ton soutien tout au long de mes études. Une sœur comme on ne peut trouver nulle part ailleurs, Puisse Allah te protéger.

A mes très chères cousines

Selma et Mina bouyahmed, Selma et Safia Amarouche, sans oublier ma très chère Belahcene Souad. Qui ont été là à me soutenir tout au long de cette année.

A Ouatah Abdelhamid

Qui m'a aidé et supporté dans les moments difficiles.

BOUYAHMED Maïssa

Table des matières

Table des matières	v
Liste des figures	vii
Liste des tableaux	viii
Liste des abréviations	ix
Introduction générale	2
1 Organisme d'accueil et étude des besoins	3
1.1 Généralités sur la logistique	4
1.1.1 Définition et historique de la logistique	4
1.1.2 Types des logistiques	4
1.1.2.1 Logistique interne	4
1.1.2.2 Logistique externe	4
1.1.2.3 Logistique industrielle	5
1.1.2.4 Logistique de distribution	5
1.1.3 La chaîne logistique (Supply Chain)	6
1.1.3.1 TMS	6
1.2 Présentation de l'organisme d'accueil	7
1.2.1 Historique et présentation de Bejaia Logistique	7
1.2.2 Missions et activités de Bejaia Logistique	8
1.2.3 Problématique	8
1.2.4 la structure de l'entreprise	10
1.3 Conclusion	11

2	Analyse et spécification des besoins	12
2.1	Introduction	12
2.2	Démarche de développement	12
2.2.1	Processus Unifié (UP)	12
2.2.2	Présentation UML	13
2.3	Spécification des besoins	14
2.3.1	Besoins fonctionnel	14
2.3.2	Besoins non fonctionnel	14
2.4	Analyse des besoins	15
2.4.1	Identification des acteurs	15
2.4.2	Diagramme de contexte dynamique	16
2.4.2.1	Identification des messages échangés	17
2.4.3	Identification des cas d'utilisation	18
2.4.4	Diagrammes des cas d'utilisation et leurs descriptions	20
2.4.4.1	Diagramme de cas d'utilisation global	20
2.4.4.2	Diagramme de cas d'utilisation - Administrateur -	22
2.4.4.3	Diagramme de cas d'utilisation - Ressources humaines -	22
2.4.4.4	Diagramme de cas d'utilisation - Planificateur -	22
2.4.4.5	Diagramme de cas d'utilisation - Commercial -	24
2.4.4.6	Diagramme de cas d'utilisation - Chargé de l'exploitation -	24
2.4.4.7	Diagramme de cas d'utilisation - Approvisionneur -	26
2.4.4.8	Diagramme de cas d'utilisation - Chargé de la maintenance -	26
2.4.5	Description textuelle des cas d'utilisation	27
2.5	Conclusion	29
3	Conception	30
3.1	Introduction	30
3.2	Diagramme de séquence système	30
3.2.1	Diagramme de séquence système « Authentification »	30
3.2.2	Diagramme de séquence système « Ajouter un utilisateur »	31
3.2.3	Diagramme de séquence système « Rechercher un utilisateur »	33
3.2.4	Diagramme de séquence système « Modifier un utilisateur »	34

3.2.5	Diagramme de séquence système «Supprimer un utilisateur»	35
3.2.6	Diagramme de séquence système « Planifier les commandes »	36
3.2.7	Diagramme de séquence système « Vérification des nouvelles commandes »	37
3.2.8	Diagramme de séquence système « Planification automatique»	38
3.3	Diagramme de classes	39
3.4	Modèle relationnel	40
3.5	Conclusion	41
4	Réalisation	42
4.1	Introduction	42
4.2	Langage et environnements de développement	42
4.2.1	Langages de développement	42
4.2.1.1	Technologies Back end	42
4.2.1.2	Technologies Front end	43
4.2.2	SGBDR (Système de Gestion de Base de Données Relationnelles) et Serveurs web	44
4.2.2.1	WampServer	44
4.2.2.2	Apache Tomcat	45
4.2.2.3	MySQL	45
4.2.2.4	PhpMyAdmin	45
4.2.3	Les composants logiciels utilisés	46
4.2.3.1	Spring boot	46
4.2.3.2	Bootstrap	46
4.2.3.3	Thymeleaf	46
4.2.3.4	Hibernate	47
4.2.4	Environnement de modélisation et de développement	47
4.2.4.1	IntelliJ IDEA	47
4.3	Architecture de développement	48
4.3.1	Le Patron Modèle-Vue-Contrôleur	48
4.3.2	Implémentation d'un projet en J2EE suivant le model MVC	50
4.4	Captures d'écran de l'application	51
4.5	Conclusion	55

Conclusion générale	56
Bibliographie	56

Table des figures

1.1	La chaine logistique	6
1.2	Logo de BL et Ifri	7
1.3	Organigramme de BL	10
2.1	Les différents diagrammes de L’UML	13
2.2	Diagramme de contexte.	18
2.3	diagramme de cas d’utilisation globale	21
2.4	Diagramme de cas d’utilisation – Administrateur -	22
2.5	Diagramme de cas d’utilisation - Ressources humaines –	22
2.6	Diagramme de cas d’utilisation - Planificateur -	23
2.7	Diagramme de cas d’utilisation - Commercial -	24
2.8	Diagramme de cas d’utilisation – Chargé de l’exploitation -	25
2.9	Diagramme de cas d’utilisation - Approvisionneur -	26
2.10	Diagramme de cas d’utilisation – Chargé de la maintenance –	26
3.1	Diagramme de séquence système « Authentification »	31
3.2	Diagramme de séquence système « Ajouter un utilisateur »	32
3.3	Diagramme de séquence système « Rechercher un utilisateur »	33
3.4	Diagramme de séquence système « Modifier un utilisateur »	34
3.5	Diagramme de séquence système « Supprimer un utilisateur »	35
3.6	Diagramme de séquence système « Planifier les commandes »	36
3.7	Diagramme de séquence système « Vérification des nouvelles commandes »	37
3.8	Diagramme de séquence système « Planification automatique »	38
3.9	Diagramme de classes	39
4.1	Logo JavaScript	44

4.2	Logo MySQL	45
4.3	Logo SpringBoot	46
4.4	Logo Thymeleaf	47
4.5	Logo IntelliJ IDEA	47
4.6	Mise en application du MVC	49
4.7	Architecture de J2EE	50
4.8	Interface d'authentification.	51
4.9	Interface liste des utilisateurs.	52
4.10	Interface formulaire d'ajout.	52
4.11	Interface de modification.	53
4.12	Interface planificateur.	53
4.13	L'interface de vérification des nouvelles commandes.	54
4.14	Interface liste des disponibilités.	54
4.15	Interface de validation d'une planification.	55

Liste des tableaux

2.1	Tableaux des différents acteurs du système.	16
2.2	Les différents messages échangés entre les acteurs.	17
2.3	Identification des cas d'utilisation.	19

Liste des abréviations

BL	Bejaia Logistique .
BDD	Base De Données.
CSS	Cascading Style Sheet.
HTML	Hyper Text Markup Language.
HTTP	Hyper Text Transfer Protocol.
IDEA	Integrated Development Environment.
JEE	Java Enterprise Edition.
JS	Java Script.
JSP	JavaServer Pages.
MVC	Modèle Vue Contrôleur.
PHP	Personal Home Page.
SGBD	System de Gestion de Base de Données.
SQL	Structured Query Language.
TMS	Transport Management System.
UML	Unified Modeling language.
UP	Unified Process.

Résumé

En vue de l'obtention du diplôme master professionnel en génie logiciel, ce travail consiste en la conception et la réalisation d'une application web d'un module de planification d'un TMS (Transport Management System) destinée à l'entreprise « Bejaia logistique ».

La modélisation a été réalisée en suivant le processus unifié UP et en utilisant le langage de modélisation UML. Concernant la réalisation de l'application, l'implémentation s'est faite à l'aide de l'environnement de développement IntelliJ IDEA, en utilisant le langage Java EE sous le Framework Spring boot, et enfin MySQL comme gestionnaire de base de données.

Mots clés : *TMS, UP, UML, MVC, MySQL, Java EE, Spring boot.*

Abstract

With a view to obtaining the professional master's degree in software engineering, this work consists of the design and production of a web application of a TMS (Transport Management System) planning module intended for the company "Bejaia logistics".

The modeling was performed following the unified process UP and using the UML modeling language. Concerning the realization of the application, the implementation was done using the development environment IntelliJ IDEA, using the Java EE language under the Spring Boot Framework, and finally MySQL as database manager.

Keywords : *TMS, UP, UML, MVC, MySQL, Java EE, Spring boot.*

Introduction générale

L'évolution des technologies de l'information et de la communication a fait évoluer l'ensemble des systèmes d'informations par l'accélération de leur rythme et l'ampleur de leur généralisation. Et comme tout autre secteur, le domaine logistique ne fait pas exception de cette évolution capricieuse.

Les systèmes d'informations sont désormais l'élément central du dispositif logistique. En effet, les chaînes logistiques deviennent de plus en plus complexes au fil du temps, ainsi leur gestion nécessite de nouvelles méthodes et technologies pour mieux s'y adapter (la réduction des coûts, la traçabilité, la planification, la fiabilité. . .).

Durant notre stage à Bejaia logistique qui est spécialisé dans le domaine du transport routier, nous avons pu assimiler le rôle important que joue leur système de gestion des transports TMS pour accomplir toutes les tâches quotidiennes. Néanmoins, au tant qu'informaticiennes, nous avons pu remarquer que le service de planification n'exploite pas le TMS et ne bénéficie pas totalement de cet atout.

L'objectif principal dans ce projet de fin de cycle master est de concevoir et réaliser une application web qui consiste en un module de planification d'un TMS qui sera destiné à Bejaia logistique.

Notre mémoire sera divisé en quatre chapitres.

Dans le premier chapitre intitulé « Organisme d'accueil et étude des besoins », nous définirons comme première partie la logistique et ces notions de base, et comme deuxième partie nous présenterons l'organisme d'accueil, et nous extrairons la problématique.

Dans le deuxième chapitre intitulé « Analyse et spécification des besoins », nous allons identifier les besoins du Bejaia logistique, et déterminer les acteurs de notre système.

Dans le troisième chapitre qui concerne la conception, nous présenterons la solution que nous souhaitons développer, ainsi que la manière dans laquelle notre système devra être déployé.

Dans le dernier chapitre dédié à la réalisation de l'application, nous présenterons dans un premier lieu l'environnement de développement et les différents outils et langages de programmation utilisés, ensuite nous exposerons des captures d'écran qui représentent les interfaces de notre application.

Enfin, nous clôturerons ce mémoire par une conclusion générale.

Chapitre 1

Organisme d'accueil et étude des besoins

Introduction

Une incompréhension réside souvent dans l'emploi courant des deux concepts transport et logistique qui sont souvent confondus bien qu'il existe des différences notables entre ces deux notions.

Le transport et la logistique sont deux concepts distincts mais entremêlés qui jouent un rôle clé dans le mouvement des biens et des personnes. Le transport représente une des plus importantes activités humaines mondiales, il peut être défini étant le déplacement d'un bien ou d'une personne entre deux points géographiques différents, mais ce dernier est tout simplement une partie de la logistique il ne pourrait y avoir de transport sans logistique.

Dans ce présent chapitre nous définirons en premier lieu ce qu'est réellement une logistique ainsi que ces types, ensuite nous présenterons notre organisme d'accueil spécialisé dans ce domaine.

1.1 Généralités sur la logistique

1.1.1 Définition et historique de la logistique

Le terme logistique provient du mot grec Logistikos qui signifie l'art du raisonnement et qui est relatif aux moyens d'optimisation d'une opération ou d'un processus.

La logistique existe depuis des millénaires, elle est née dans un milieu militaire lors des préparations en prévision d'une bataille pour mettre à disposition les moyens de transports et les équipements et tout ce qui concerne les denrées alimentaires. Comme tout autre mode de transport, le transport routier de marchandise a connu plusieurs étapes durant son développement, à commencer par la marche à pied puis aux animaux et enfin l'arrivée de la roue et du moteur qui ont bouleversé l'essor du transport routier de marchandises.

Aujourd'hui la logistique est un élément central dans une entreprise, en effet, plusieurs années après la seconde guerre mondiale, le terme logistique fut emprunté au contexte militaire et appliqué à l'entreprise, la logistique de celle-ci consiste à coordonner les produits en circulation de manière à ce que les produits circulent en continu, d'une part pour diminuer les délais de livraison, en d'autre part pour optimiser les coûts, en d'autres termes gérer tout ce qui concerne le transport et le stockage des produits de l'entreprise [1].

1.1.2 Types des logistiques

1.1.2.1 Logistique interne

C'est la partie de la logistique qui se réfère à l'organisation de tous les flux de matières et d'information ayant lieu à l'intérieur des propres frontières de l'entreprise, autrement dit elle se préoccupe de l'organisation et de la gestion des flux interne de l'entreprise. Exemples : les mouvements de stocks dans l'entrepôt, la gestion des stocks, etc [2] [3].

1.1.2.2 Logistique externe

Elle-même se devise en :

- **Logistique d'approvisionnement** : qui représente l'organisation et la gestion des flux des fournisseurs vers l'entreprise.

- **Logistique de distribution** : qui représente l'organisation et la gestion des flux de l'entreprise jusqu'aux clients.

Cette dernière couvre donc le transport des produits jusqu'à leurs destinations finales [2].

1.1.2.3 Logistique industrielle

Se focalise sur la production et la satisfaction de la demande d'où vient aussi le nom logistique de production. Elle consiste donc à mettre en place des méthodes classiques qui permettent de gérer de façon optimale les flux de la production. Par conséquent elle englobe l'ensemble des infrastructures ciblées à la production des produits (usines, ateliers, sites..), auxquelles il est nécessaire d'associer différents entrepôts :

- Entrepôt pour la logistique amont (stockage des matières premières et produits nécessaires à la production).
- Les entrepôts intermédiaires et atelier (qui contiennent les encours de production).
- Entrepôts pour la logistique aval (stockage des produits finis avant leur livraison au client) [4].

1.1.2.4 Logistique de distribution

La phase de distribution signifie le transport et la livraison des marchandises ou des produits finis à ses clients. Cette dernière intervient après toutes les activités d'achats et d'approvisionnement des matériels et produits, et après les processus de fabrication et de production.

Par déduction, la politique de la logistique de la distribution réside principalement dans l'organisation de l'acheminement de la marchandise, elle rassemble donc les activités mises en pratique pour satisfaire les commandes clients, ce type de logistique intègre : l'administration des ventes (réception des commandes client, engagement sur les délais et planification des livraisons), la tenue des stocks et des magasins de produits finis, la préparation des commandes (prélèvement, conditionnement et emballage, suremballage), l'expédition et enfin la gestion des réseaux de distribution (plateformes de groupage et de dégroupage) [5].

1.1.3 La chaîne logistique (Supply Chain)

Aussi appelée « Supply Chain » la chaîne logistique n'est pas une fonction de l'entreprise, ni un service achetable à un prestataire de service, ni un module informatique, c'est en fait une démarche de fonctionnement qui vise à assurer une gestion de l'ensemble des processus qui permet à un ou plusieurs systèmes clients ou fournisseurs de prendre en compte et de répondre aux attentes des clients finaux.

La Supply Chain est une chaîne d'approvisionnement qui peut être divisée en trois flux : flux physique (déplacement et le stockage des marchandises), flux d'information qui permettra le fonctionnement du flux physique (quoi ? où ? comment ? combien ? et quand ?), et le flux financier qui est une forte stratégie d'approvisionnement à part entière qui a pour objectif de rendre accessible ses produits au bon moment, au bon endroit et au meilleur prix. Elle contient donc toutes les activités associées au flux et à la transformation des biens, depuis les matières premières jusqu'au produit fini livré à l'utilisateur, ainsi que les flux d'informations associées [6].

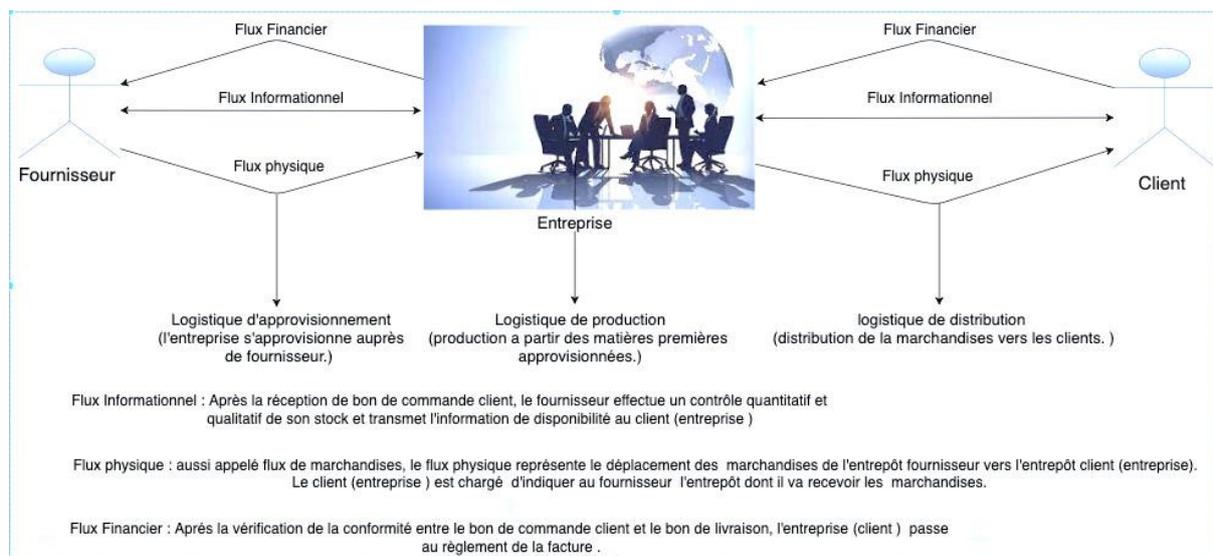


FIGURE 1.1 – La chaîne logistique

1.1.3.1 TMS

Le TMS (Transport Management System) est une plateforme logistique qui utilise des NTIC (Nouvelles Technologies de l'Information et de Communication), elle permet non seulement d'automatiser et d'optimiser les missions liées au transport, mais aussi de proposer des solutions de traçabilité des livraisons, et cela en couvrant plusieurs tâches

telles que la gestion d'une flotte de camions et de chauffeurs, l'organisation des plannings de chargement, les livraisons, leurs déchargements et la facturation [7].

Le TMS apporte de nombreux avantages à l'entreprise :

- Meilleure conformité sur les importations et exportations.
- Amélioration du service client et de la satisfaction client avec des mises à jour en temps réel.
- Réduction des coûts pour l'entreprise et le client final.
- Gain de temps.
- Simplification des processus de la Supply Chain [8].

1.2 Présentation de l'organisme d'accueil

1.2.1 Historique et présentation de Bejaia Logistique

La Sarl Bejaia logistique est une entreprise d'une notoriété nationale qui exerce dans le domaine du transport routier et l'un des acteurs les plus indispensables dans ce domaine qui a été sélectionnée grâce à son succès, cette dernière est située au pied de la montagne IFRI dans la commune d'Ouzellaguen. Elle est implantée à la zone industrielle Ahrik Ighzar Amokrane, dans la wilaya de Bejaia au nord-est de l'Algérie.

Bejaia logistique offre quotidiennement des solutions appropriées et efficaces à une clientèle professionnelle très diversifiée, dont, entreprises du BTP (Bâtiments et Travaux Publics), collectivités et administrations, industries, PME (les petites et moyennes entreprises), grâce à leurs capacités, aux compétences de leurs collaborateurs et aux investissements technologiques [9].



FIGURE 1.2 – Logo de BL et Ifri

1.2.2 Missions et activités de Bejaia Logistique

- Les activités de la Sarl BL sont étendues du transport public de marchandises, location d'engins, de véhicule avec ou sans chauffeur, du matériel pour bâtiments et travaux publics.
- BL a des missions très diverses liées aux activités qu'elle pratique tous les jours comme :
 - Assurer la satisfaction des clients.
 - Assurer la sécurité des personnes et des biens en rapport avec les activités du transport.
 - Assurer la gestion (conduite, exploitation et maintenance) et le développement de ses réseaux d'activités.
 - Élargir son réseau d'activités en dehors du pays à des zones pas encore exploitées.

BL place la satisfaction du client, le respect des engagements et l'intégrité de la propriété du client au centre de ses valeurs et sa politique qualité.

1.2.3 Problématique

Comme nous l'avons précédemment signalé, Bejaia logistique (BL) est une entreprise qui bénéficie d'une image de qualité, c'est la référence algérienne dans le domaine du transport routier. Dans ce cas, le TMS représente un outil indispensable de gestion informatique des transports.

Lors des réunions qui ont eu lieu avec les différents services de BL, nous avons pu cerner un ensemble d'exigences pour les différents utilisateurs spécialement les planificateurs, en effet durant notre stage au sein du service informatique, nous avons constaté l'absence du module le plus essentiel qui s'agit bien de la planification d'où nous avons recensé quelques problèmes dont :

- L'absence d'un système informatique pour la planification des missions ainsi qu'une base de données.
- Toutes les informations sont saisies manuellement et stockées dans des fichiers Excel.
- Insécurité des informations.

- Risque d'erreur lors de la collecte des données.
- Charge de travail énorme pour les planificateurs.
- Difficulté de récupération de l'historique des planifications.
- Difficulté d'échange d'information entre les différents planificateurs, ce qui provoque une perte de temps.
- Plusieurs planificateurs font les mêmes tâches, ce qui réduit l'exploitation des compétences de chaque employé.

1.2.4 la structure de l'entreprise

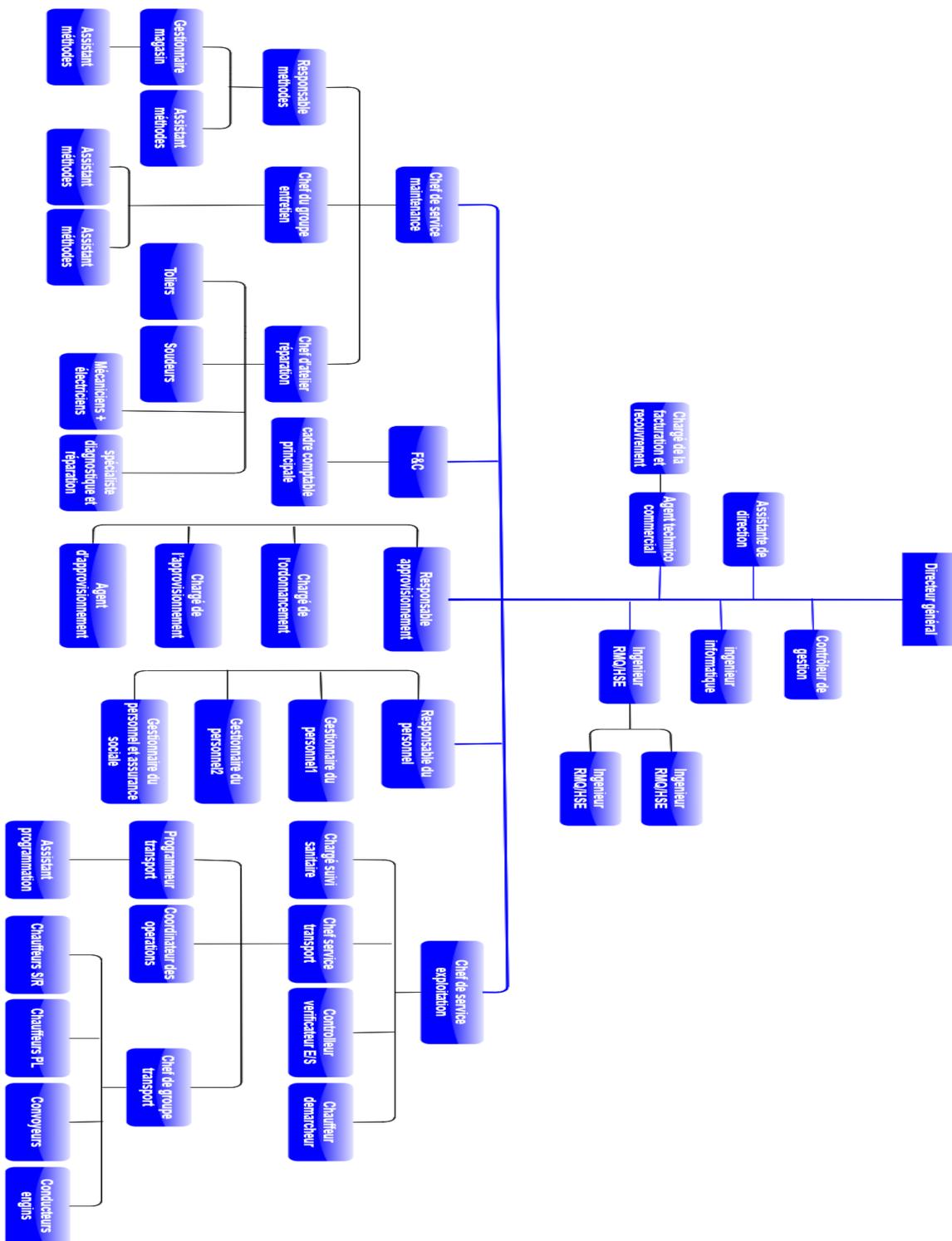


FIGURE 1.3 – Organigramme de BL

1.3 Conclusion

Dans la première section de ce chapitre, nous avons vu quelques notions sur la logistique telle que la Supply chain et le TMS (Transport Management System) qui représente le cœur de notre projet

Dans la deuxième section, nous avons commencé par présenter l'entreprise Bejaia logistique ainsi que son fonctionnement interne. Ensuite, nous sommes parvenus à cerner la problématique.

Le chapitre suivant sera consacré à la spécification et analyse des besoins.

Chapitre 2

Analyse et spécification des besoins

2.1 Introduction

Avant de développer un système d'information, il est nécessaire de choisir une méthode bien définie afin de bien organiser le travail à effectuer en citant les différentes tâches et les différentes étapes. Pour cela nous avons consacré ce chapitre à l'analyse et la spécification des besoins ou nous avons opté pour le processus UP qui est centré sur l'architecture UML et piloté par des cas d'utilisations. Dans ce chapitre nous identifierons toutes les fonctionnalités de l'application selon les exigences fonctionnelles et non fonctionnelles, puis les entités externes qui vont interagir avec le système, ensuite élaborer le diagramme de contexte ainsi que le diagramme de cas d'utilisation pour spécifier les besoins de notre système.

2.2 Démarche de développement

2.2.1 Processus Unifié (UP)

Pour la réalisation de notre application, notre choix a été porté sur le Processus Unifié (UP). En effet, ce dernier est un processus de conception et de réalisation de logiciels, il se caractérise par une démarche itérative et incrémental, ce processus est conduit par les cas d'utilisation et piloté par les risques, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel [10].

2.2.2 Présentation UML

UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins. Ce dernier est un langage visuel qui sert principalement à représenter les actions exécutées par le logiciel et son fonctionnement [11].

UML propose treize diagrammes pour la description d'un système. Ces diagrammes sont divisés en deux catégories :

- **Diagrammes Structurels** : qui permettent de visualiser, spécifier, construire et documenter l'aspect statique ou structurel du système d'information.
- **Diagrammes comportementaux** : qui permettent de modéliser les aspects dynamiques du système, c'est-à-dire les différents éléments qui sont susceptibles de subir des modifications [12].

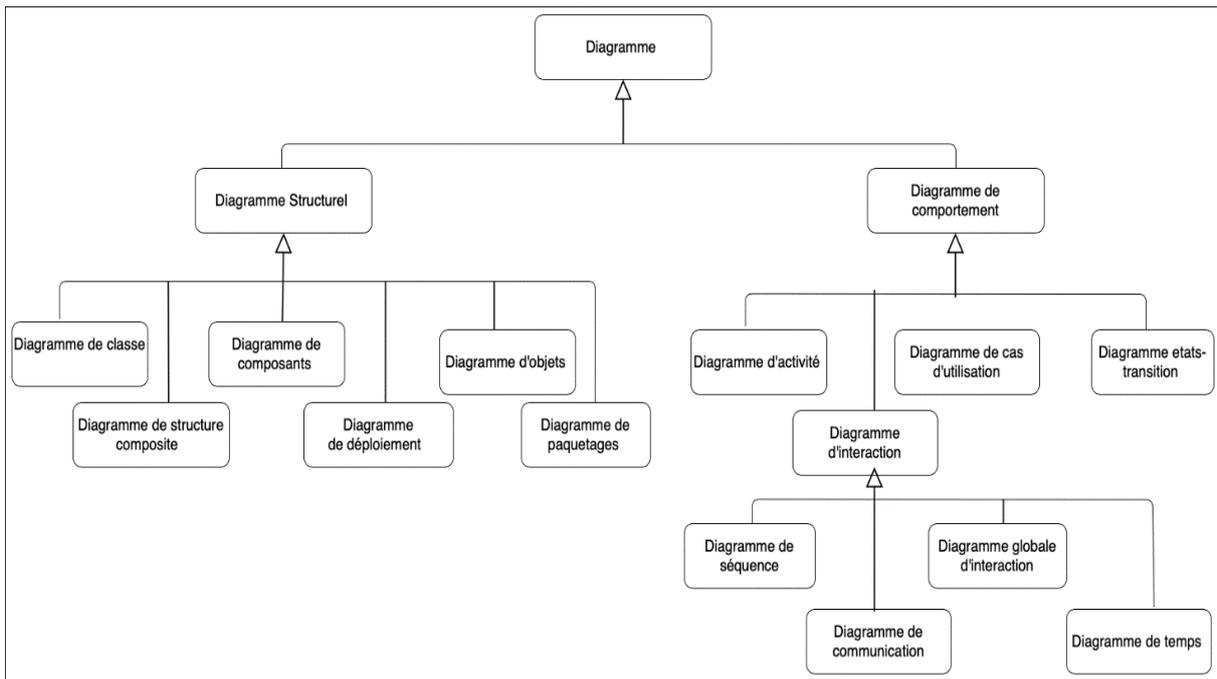


FIGURE 2.1 – Les différents diagrammes de L'UML

2.3 Spécification des besoins

L'expression des besoins permet de décrire un projet de manière globale avec une approche technique et exhaustive avant même de le commencer.

Afin de pouvoir clarifier les besoins des utilisateurs de notre application, nous présenterons les besoins fonctionnels et non fonctionnels [13].

2.3.1 Besoins fonctionnel

Les besoins fonctionnels représentent les différentes fonctionnalités du système, ils expriment les exigences auxquelles l'application doit répondre.

- Automatiser le processus de planifications des missions.
- Avoir une base de données pour le stockage des chauffeurs et des véhicules.
- L'authentification des utilisateurs par un login et un mot de passe.
- Mise à jour des utilisateurs par l'administrateur et attributions des privilèges d'accès.
- Facilitation des recherches.
- Le planificateur pourrait consulter les commandes non planifiées via l'application.
- Possibilité de planifier automatiquement les missions.
- Établir un diagramme de Gantt pour la vérification des disponibilités chauffeurs et véhicules.

2.3.2 Besoins non fonctionnel

Une fois les besoins fonctionnels sont bien définis, les besoins non fonctionnels qui représentent le comportement et la performance de l'application, doivent être pris en compte tout au long du processus de développement à savoir :

- **Authentification** : chaque utilisateur doit s'authentifier par un nom et un mot de passe unique pour accéder à l'application.
- **Extensibilité** : l'application doit être extensible c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.
- **Rapidité du traitement** : l'application doit assurer un temps de réponse minime, tout en répondant aux exigences de l'utilisateur.

- **Sécurité** : les données recueillies par l'application ne doivent être accessibles que par le personnel autorisé.
- **Ergonomie** : l'interface doit être simple à utiliser, les informations doivent être lisibles.
- **La convivialité** : le futur logiciel doit être facile à utiliser, en effet les interfaces utilisateurs doivent être conviviales autrement dit simples, ergonomiques et adaptées à l'utilisateur.

2.4 Analyse des besoins

L'analyse des besoins est une des premières techniques de gestion de projet à mettre en œuvre, elle permet de traiter la partie fonctionnelle du projet. Dans cette partie nous identifierons dans un premier temps les acteurs impliqués, nous définirons ensuite les messages échangés entre ces derniers avec le système pour établir un diagramme de contexte, et pour finir nous spécifierons les cas d'utilisations ainsi que leurs descriptions textuelles.

2.4.1 Identification des acteurs

Les acteurs sont les utilisateurs directs de l'application. Un acteur représente un rôle joué par une entité externe (une personne, un dispositif matériel ou un autre système) qui interagit directement avec le système étudié.

Pour notre système, les acteurs identifiés sont : Administrateur, RH, Planificateur, client, commercial, chargé de l'exploitation, approvisionneur, chargé de la maintenance.

Le tableau suivant représente les différents acteurs et leurs rôles :

Acteurs	Rôles
Administrateur	<ul style="list-style-type: none"> • Gérer les utilisateurs qui peuvent utiliser le système.
Ressource humaine	<ul style="list-style-type: none"> • Déclarer les disponibilités des chauffeurs. • Gérer les employés. • Gérer les congés maladies. • Gérer les accidents de travail. • Gérer les assurances et des déplacements a la CNAS. • Gérer les sanctions. • Gérer les traitements de pointage.
Planificateur	<ul style="list-style-type: none"> • Planifier les commandes. • Localiser et suivre les camions et les chauffeurs en temps réel. • Gérer les missions. • Élaborer la feuille de route. • Demander l'affrètement en cas de besoin.
Client	<ul style="list-style-type: none"> • Passer une commande au sein de l'entreprise. • Effectuer le paiement.
Commercial	<ul style="list-style-type: none"> • Recevoir les commandes clients. • Préparer des factures de recouvrement à discuter. • Traiter les différents documents clients. • Gérer les commandes.
Approvisionneur	<ul style="list-style-type: none"> • Approvisionner les produits. • Gérer les sous-traitants. • Gérer l'affrètement.
Chargé de l'exploitation	<ul style="list-style-type: none"> • Gérer les véhicules. • Gérer les trajets.
Chargé de la maintenance	<ul style="list-style-type: none"> • Réparer les véhicules. • Déclarer l'indisponibilité des véhicules.

TABLE 2.1 – Tableaux des différents acteurs du système.

2.4.2 Diagramme de contexte dynamique

Le diagramme de contexte constitue une étape intermédiaire entre le cahier des charges et la construction des premiers cas d'utilisations, il permet de visualiser tous les messages

échangés entre le système et les acteurs.

Ce dernier présente deux versions un diagramme de contexte statique et un autre dynamique. Dans cette partie nous allons élaborer le diagramme de contexte dynamique qui met en évidence les missions de chaque catégorie d'acteur vis-à-vis du système [14].

2.4.2.1 Identification des messages échangés

Le tableau suivant illustre les différents messages (informations, objets, services) échangés entre les acteurs de notre système et le système.

ID Messages Acteur - Système	ID Messages Système - Acteur
<ul style="list-style-type: none"> - M1 Demande de l'authentification. - M2 Mise à jour des Chauffeurs. - M3 Mise à jour des Missions. - M4 Mise à jour des Commandes. - M5 Mise à jour des Clients. - M6 Mise à jour des Trajets. - M7 Mise à jour des Véhicules. - M8 Mise à jour des sous-traitants. - M9 Mise à jour des Services Clients. - M10 Mise à jour des destinations. - M11 Mise à jour des utilisateurs. - M12 Demande d'affrètement. - M13 Demande de L'élaboration de la feuille de route. - M14 Demande de passer une commande. - M15 Mise à jour feuille de route. - M16 Mise à jour des affrètements. - M17 Demande de planification des commandes. - M18 Mise à jour des disponibilités chauffeurs. - M19 Mise à jour des disponibilités véhicules. 	<ul style="list-style-type: none"> - M1' Interface de l'authentification. - M2' Interface gestion Chauffeur. - M3' Interface gestion Mission. - M4' Interface gestion Commande. - M5' Interface gestion Client. - M6' Interface gestion Trajet. - M7' Interface gestion Véhicules. - M8' Interface gestion sous-traitants. - M9' Interface gestion Services Clients. - M10' Interface gestion Destination. - M11' Interface gestion Utilisateur. - M12' Interface Service Affrètement. - M13' Interface Feuille de route. - M14' Interface Commande. - M15' Interface gestion feuille de route. - M16' Interface gestion des affrètements. - M17' Interface de planification commandes. - M18' Interface de gestion des indisponibilités chauffeurs. - M19' Interface de gestion des indisponibilités véhicules.

TABLE 2.2 – Les différents messages échangés entre les acteurs.

Le diagramme suivant représente le diagramme de contexte permettant de présenter et de visualiser tous les messages échangés entre les acteurs du système identifiés auparavant.

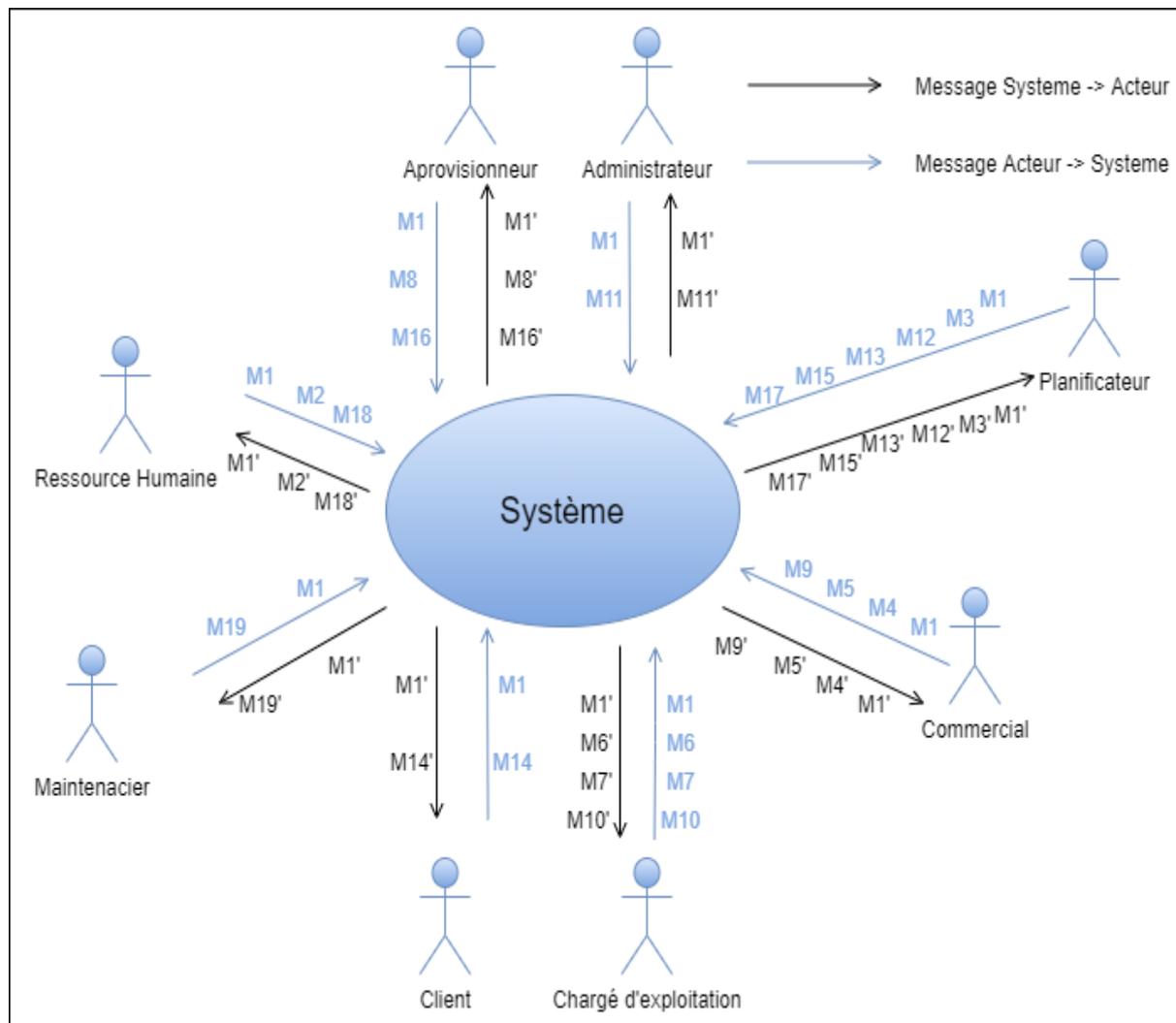


FIGURE 2.2 – Diagramme de contexte.

Tous les messages (Système Acteurs) identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme appelé diagramme de contexte, qui constitue une étape intermédiaire entre le cahier des charges et la construction des premiers cas d'utilisations, il permet de visualiser tous les messages échangés entre le système et les acteurs.

2.4.3 Identification des cas d'utilisation

Un cas d'utilisation correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur. Le but de ce dernier est d'éviter de tom-

ber dans la dérive d'une approche fonctionnelle, ils ne doivent en aucun cas décrire des solutions d'implémentation [11] [15].

Le tableau suivant liste les différents cas d'utilisation associés à notre système.

Numéro	Cas d'utilisation	Acteur
01	S'authentifier	Administrateur
		Chargé d'exploitation
		Chargé de la maintenance
		Planificateur
		Commercial
		Approvisionneur
		Client
02	Gérer les chauffeurs	Ressource humaine
03	Déclarer les indisponibilités chauffeurs	Ressource humaine
04	Gérer les missions	Planificateur
05	Planifier les commandes	Planificateur
06	Élaboration de la feuille de route	Planificateur
07	Demander l'affrètement	Planificateur
08	Gérer les commandes	Commercial
09	Gérer les clients	Commercial
10	Gérer service client	Commercial
11	Gérer les trajets	Chargé d'exploitation
12	Gérer destination	Chargé d'exploitation
13	Gérer les véhicules	Chargé d'exploitation
14	Gérer les sous-traitants	Approvisionneur
15	Gérer affrètement	Approvisionneur
16	Gérer chauffeur externes	Approvisionneur
17	Gérer chauffeur internes	Approvisionneur
18	Déclarer les indisponibilités véhicules	Chargé de la maintenance

TABLE 2.3 – Identification des cas d'utilisation.

2.4.4 Diagrammes des cas d'utilisation et leurs descriptions

Les diagrammes de cas d'utilisations servent à avoir une vision globale du comportement fonctionnel d'un système logiciel et cela en identifiant les différentes actions entre le système et ses acteurs. Les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, néanmoins, ils ne montrent pas comment le système fonctionne en interne [16].

2.4.4.1 Diagramme de cas d'utilisation global

Le diagramme suivant représente le diagramme de cas d'utilisation global de notre système.

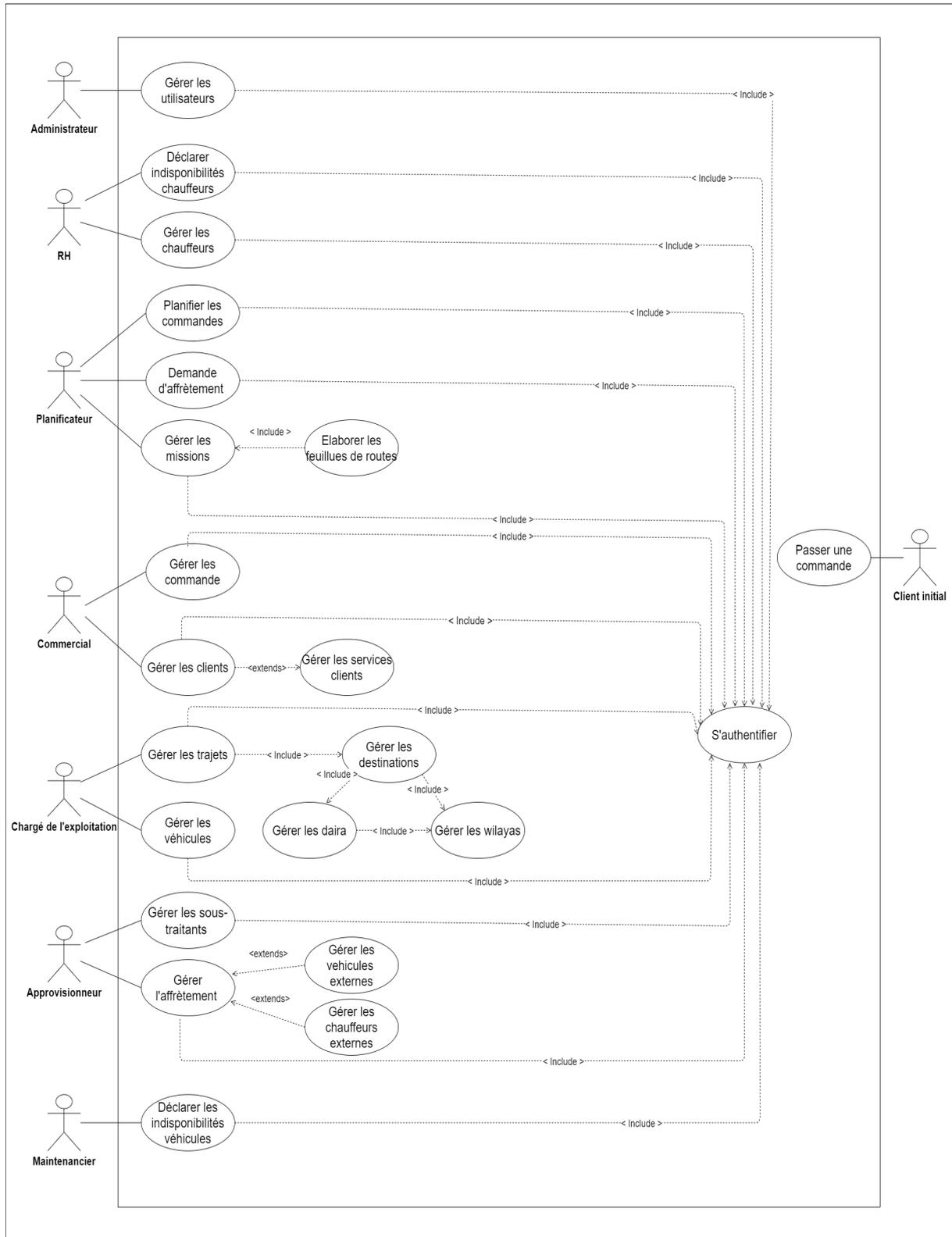


FIGURE 2.3 – diagramme de cas d'utilisation globale

2.4.4.2 Diagramme de cas d'utilisation - Administrateur -

La figure suivante représente le diagramme de cas d'utilisation détaillé associé à l'acteur « Administrateur ».

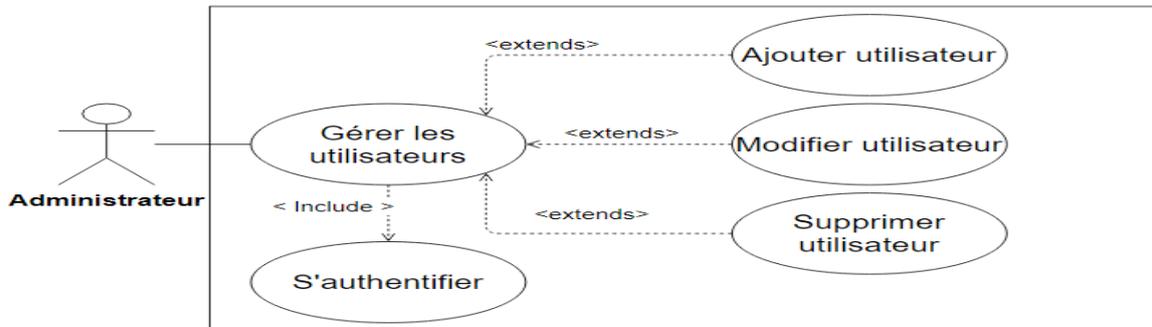


FIGURE 2.4 – Diagramme de cas d'utilisation – Administrateur -

2.4.4.3 Diagramme de cas d'utilisation - Ressources humaines -

Le diagramme suivant représente tous les cas d'utilisation associé à l'acteur RH.

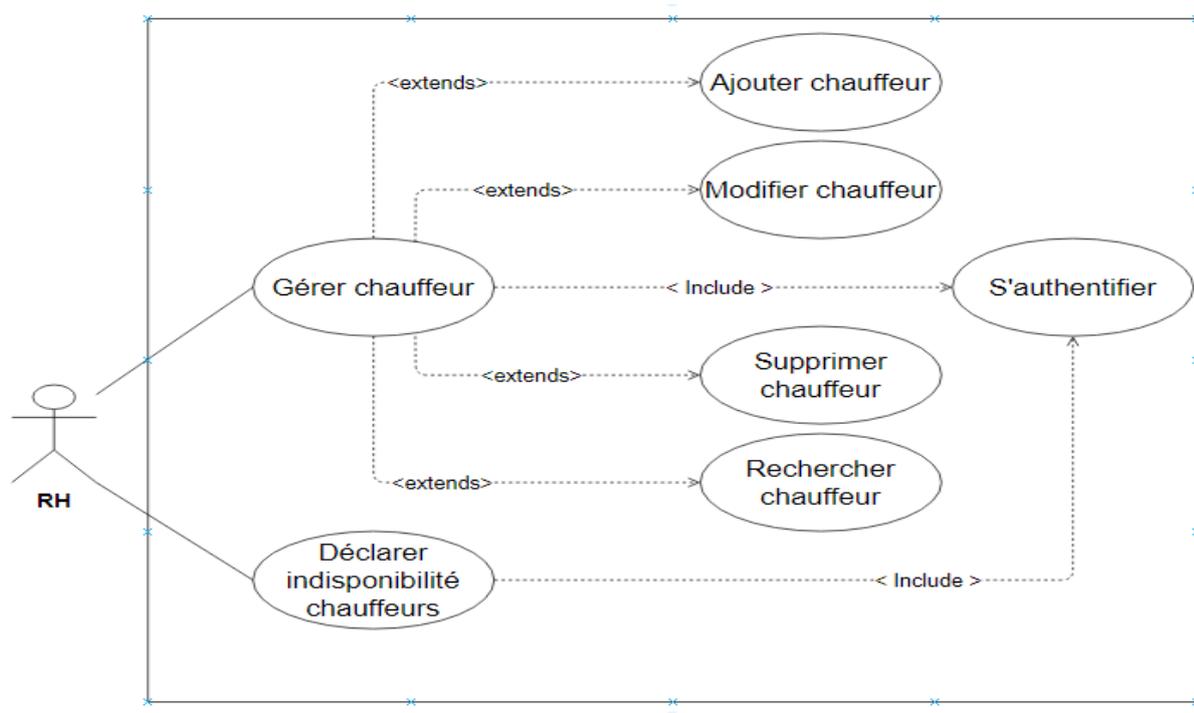


FIGURE 2.5 – Diagramme de cas d'utilisation - Ressources humaines -

2.4.4.4 Diagramme de cas d'utilisation - Planificateur -

Le diagramme ci-dessous représente les différentes tâches réalisées par un planificateur.

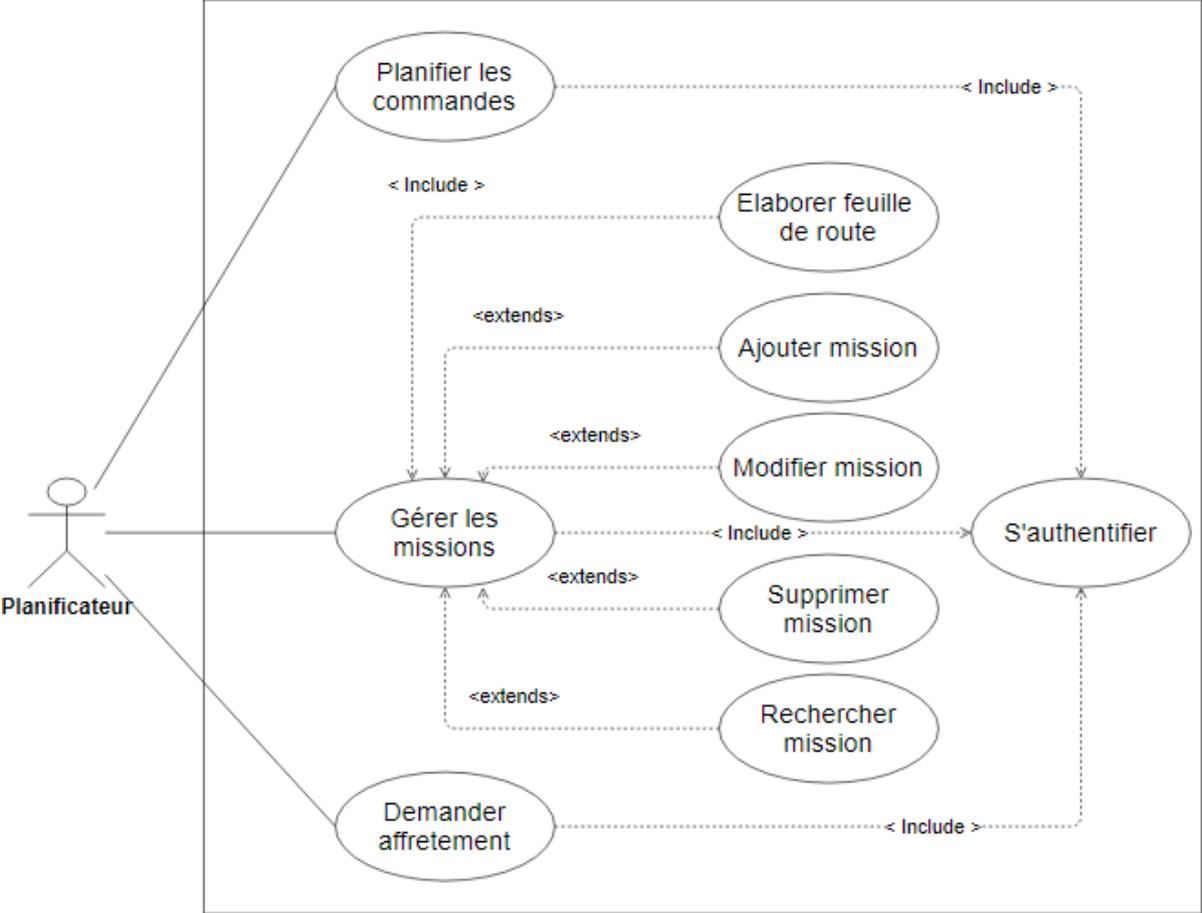


FIGURE 2.6 – Diagramme de cas d'utilisation - Planificateur -

2.4.4.5 Diagramme de cas d'utilisation - Commercial -

Le diagramme suivant représente les cas d'utilisation associé au commercial, ce dernier réalise en effet toutes les tâches représentées dans la figure ci-dessous.

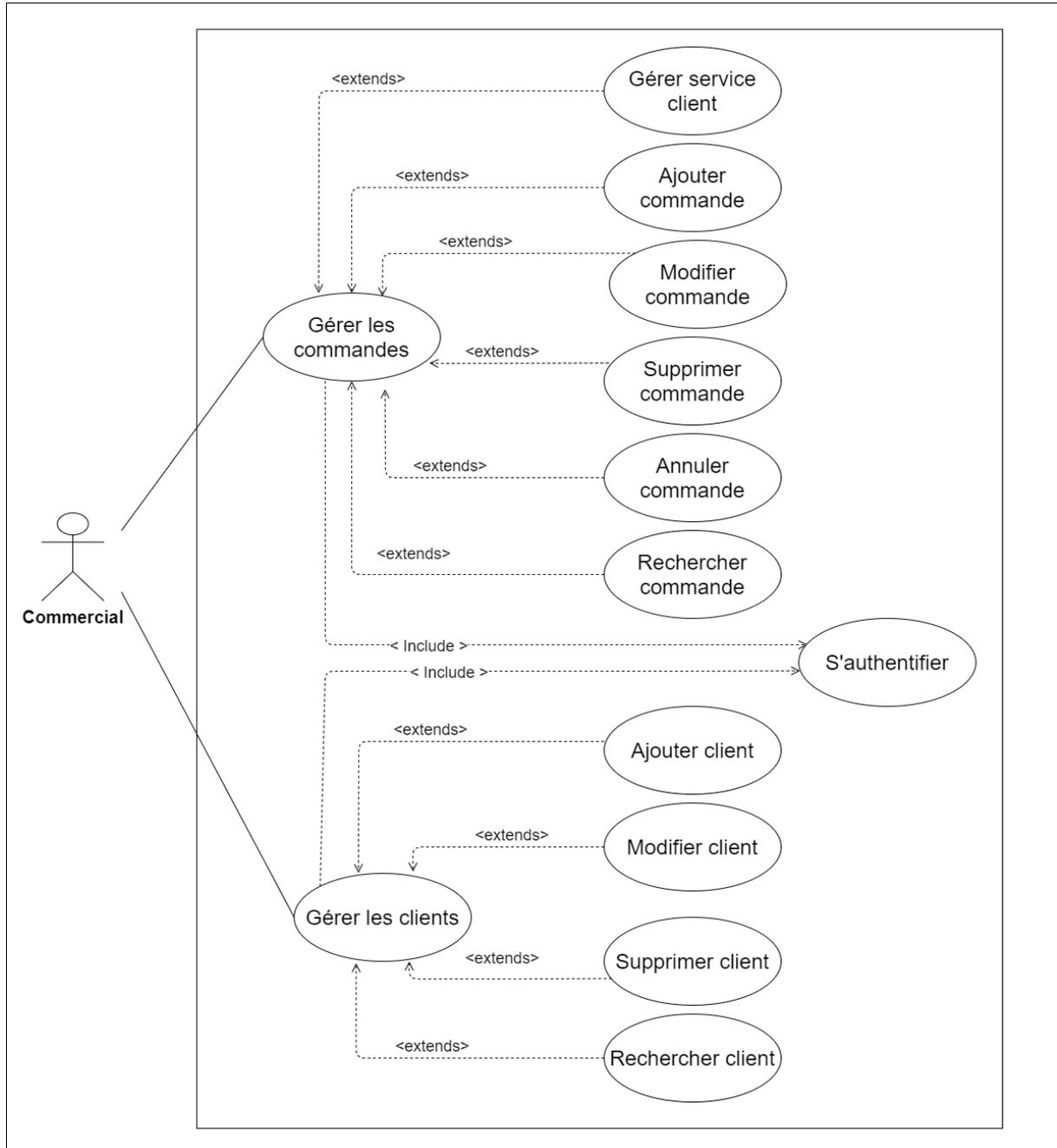


FIGURE 2.7 – Diagramme de cas d'utilisation - Commercial -

2.4.4.6 Diagramme de cas d'utilisation – Chargé de l'exploitation -

La figure suivante représente le diagramme de cas d'utilisation détaillé de l'acteur « Chargé d'exploitation ».

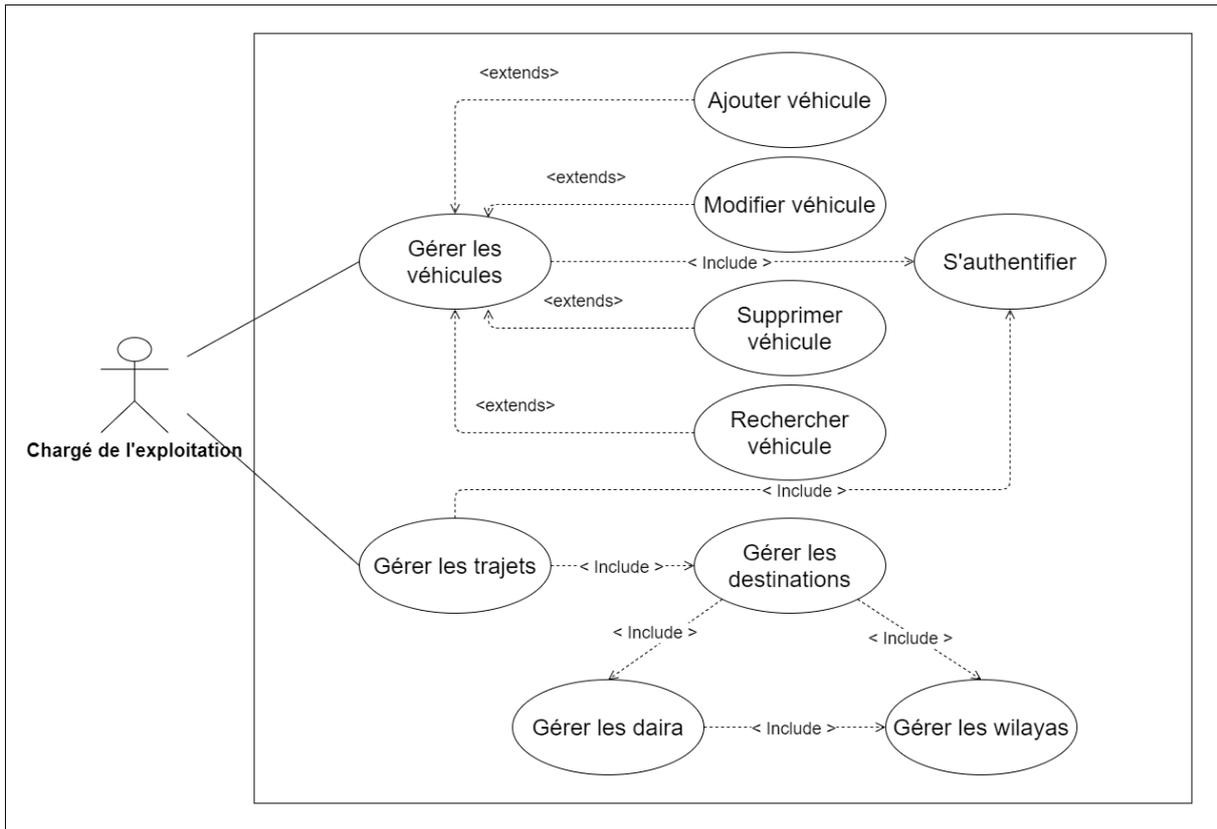


FIGURE 2.8 – Diagramme de cas d'utilisation – Chargé de l'exploitation -

2.4.4.7 Diagramme de cas d'utilisation - Approvisionneur -

Le diagramme suivant représente les différentes tâches réalisées par un approvisionneur de BL.

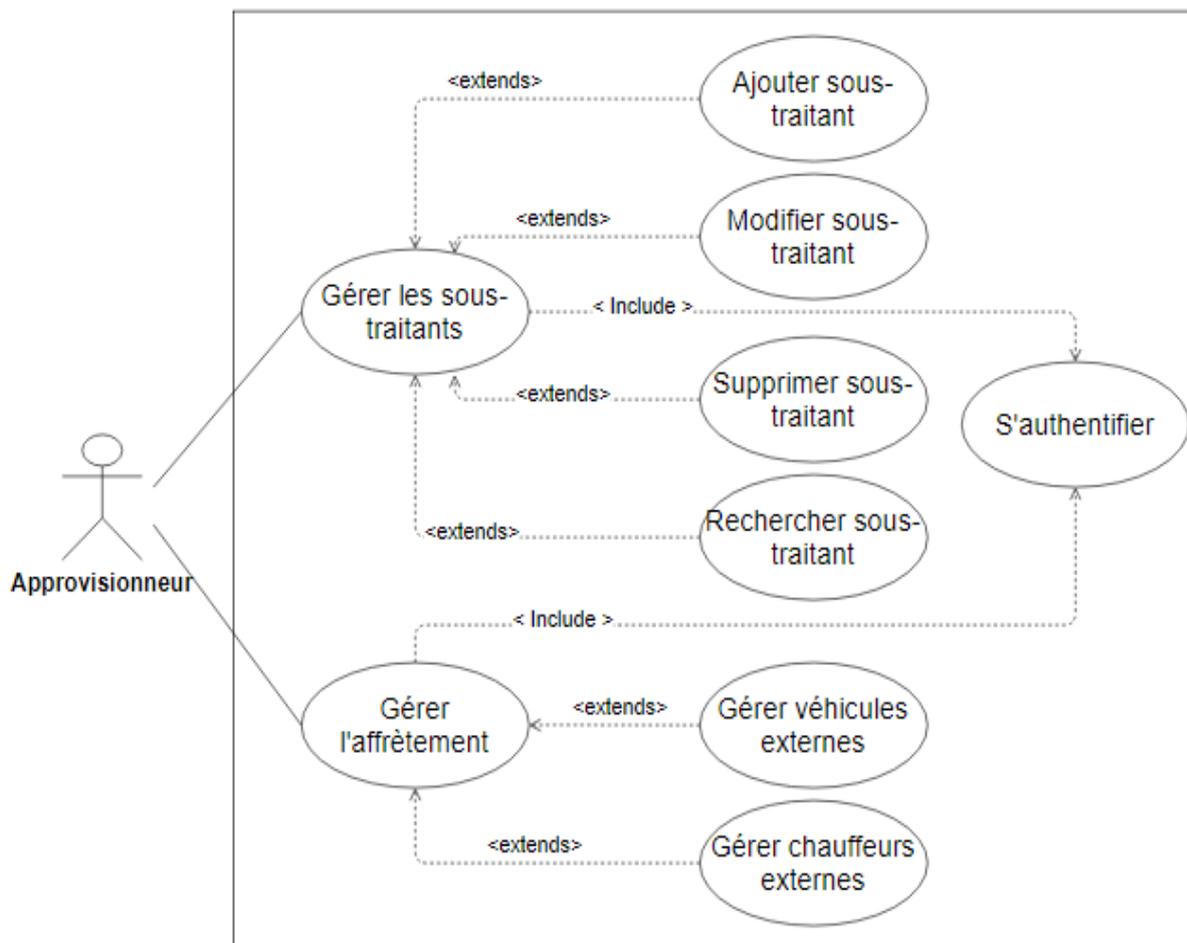


FIGURE 2.9 – Diagramme de cas d'utilisation - Approvisionneur -

2.4.4.8 Diagramme de cas d'utilisation – Chargé de la maintenance –

La figure suivante représente le diagramme de cas d'utilisation détaillé associé au Chargé de la maintenance.

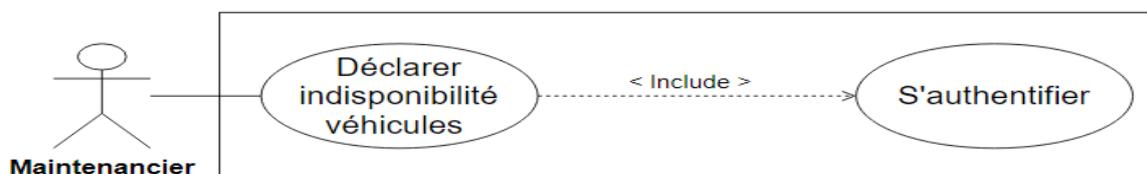


FIGURE 2.10 – Diagramme de cas d'utilisation – Chargé de la maintenance –

2.4.5 Description textuelle des cas d'utilisation

Il est nécessaire d'adjoindre une description textuelle à chaque cas d'utilisation afin de bien comprendre le détail de ce qu'il recouvre. Cette description est composée de plusieurs points dont les plus importants sont :

- Le scénario nominal qui est un enchaînement d'actions typiques dans le cas où tout se passe comme prévu.
- Le scénario alternatif qui représente un ensemble d'actions qui s'exécutent si les conditions du scénario nominal ne sont pas validées.

Dans ce qui suit, nous donnerons une description textuelle de quelques cas d'utilisation.

1. Authentification :

- **Objectif** : permet à l'utilisateur de s'identifier et d'accéder à son espace de travail selon les privilèges accordés.
- **Acteur** : tous les acteurs
- **Scénario nominal** :
 1. Le système affiche l'interface d'authentification.
 2. L'utilisateur saisit son nom et son mot de passe.
 3. Le système vérifie les informations introduites.
 4. Le système affiche l'interface adéquate à chaque type de l'utilisateur.
- **Scénario alternatif** :
 1. Nom d'utilisateur ou mot de passe incorrect.
 - Le système affiche un message d'erreur et demande d'introduire les informations correctes.

2. Gestion des chauffeurs :

- **Objectif** : permet d'ajouter, modifier, ou de supprimer un chauffeur du système.
- **Acteur** : RH
- **Scénario nominal 1** :
 1. RH remplit le formulaire d'ajout et le valide.

2. Le système affiche un message de confirmation.

● **Scénario alternatif 1 :**

1. L'un des champs du formulaire est vide.

- Le système affiche un message d'erreur en indiquant le champ concerné.

2. Le chauffeur existe déjà.

- Le système affiche un message indiquant que le chauffeur existe déjà.

● **Scénario nominal 2 :**

1. RH choisit le chauffeur à supprimer.

2. Le système affiche un message de confirmation.

● **Scénario alternatif 2 :**

1. RH choisit le chauffeur concerné par la modification.

2. RH modifie les informations concernées.

3. Le système affiche un message de confirmation.

3. **Gestion des véhicules :**

● **Objectif :** permet d'ajouter, modifier, ou de supprimer un véhicule du système.

● **Acteur :** Chargé de l'exploitation.

● **Scénario nominal 1 :**

1. Le chargé d'exploitation remplit le formulaire d'ajout et le valide.

2. Le système affiche un message de confirmation.

● **Scénario alternatif 1 :**

1. L'un des champs du formulaire est vide.

- Le système affiche un message d'erreur en indiquant le champ concerné.

2. Le chauffeur existe déjà.

- Le système affiche un message indiquant que le véhicule existe déjà.

● **Scénario nominal 2 :**

1. RH choisit le véhicule à supprimer.

2. Le système affiche un message de confirmation.

● **Scénario nominal 3 :**

1. RH choisit le véhicule concerné par la modification.
2. RH modifie les informations concernées.
3. Le système affiche un message de confirmation.

4. Recherche de clients :

- **Objectif** : Rechercher un client à l'aide de son identifiant.
- **Acteur** : commercial.
- **Scénario nominal** :
 1. Le commercial introduit l'identifiant du client (Numéro de la carte d'identité) et valide la recherche.
 2. Le système affiche une interface avec les informations du client.
- **Scénario alternatif** :
 1. Aucun résultat de recherche, le client n'existe pas.
 - Le système affiche un message indiquant que le client n'existe pas.

2.5 Conclusion

Ce chapitre nous a permis dans un premier lieu de spécifier les besoins de notre application en définissant les différents besoins fonctionnels et non fonctionnels, et de les analyser dans un deuxième lieu. Cette analyse s'est faite en décrivant le contexte de notre projet, et en spécifiant les différents cas d'utilisations grâce aux diagrammes des cas d'utilisation.

Dans ce qui suit nous entamerons la conception de notre projet en présentant les diagrammes des interactions et le diagramme des classes.

Chapitre 3

Conception

3.1 Introduction

Après avoir fini l'étape de spécification des besoins, nous entamons à présent les phases d'analyse et de conception. Pour cela, nous commencerons par illustrer les différents diagrammes de séquences système qui représenteraient les interactions chronologiques entre les acteurs et le système, par la suite nous détaillerons la conception de notre application en présentant le diagramme de classes.

3.2 Diagramme de séquence système

Un diagramme de séquence est un diagramme d'interaction UML qui représente les différentes interactions entre les acteurs et le système selon un ordre chronologique. Il comprend un groupe d'objets organisés horizontalement, des messages échangés entre les lignes de vie qui sont représentés par des flèches du haut vers le bas, et peut contenir différents fragments qui peuvent s'imbriquer les uns dans les autres comme l'opérateur conditionnel « alt » et l'opérateur d'itération « loop » [17].

Dans ce qui suit, nous présenterons les diagrammes de séquences relatifs aux cas d'utilisation présentés dans le chapitre précédent.

3.2.1 Diagramme de séquence système « Authentification »

Pour accéder à son espace de travail, l'utilisateur doit saisir son nom et son mot de passe et valide l'opération. Deux cas peuvent se présenter, si les champs sont vides, le

système affichera un message d'erreur, sinon le système procède à la vérification d'existence de ce compte, deux autres cas peuvent se présenter : si le compte n'existe pas, le système affichera un message d'erreur, sinon l'interface correspondante à l'utilisateur s'affiche.

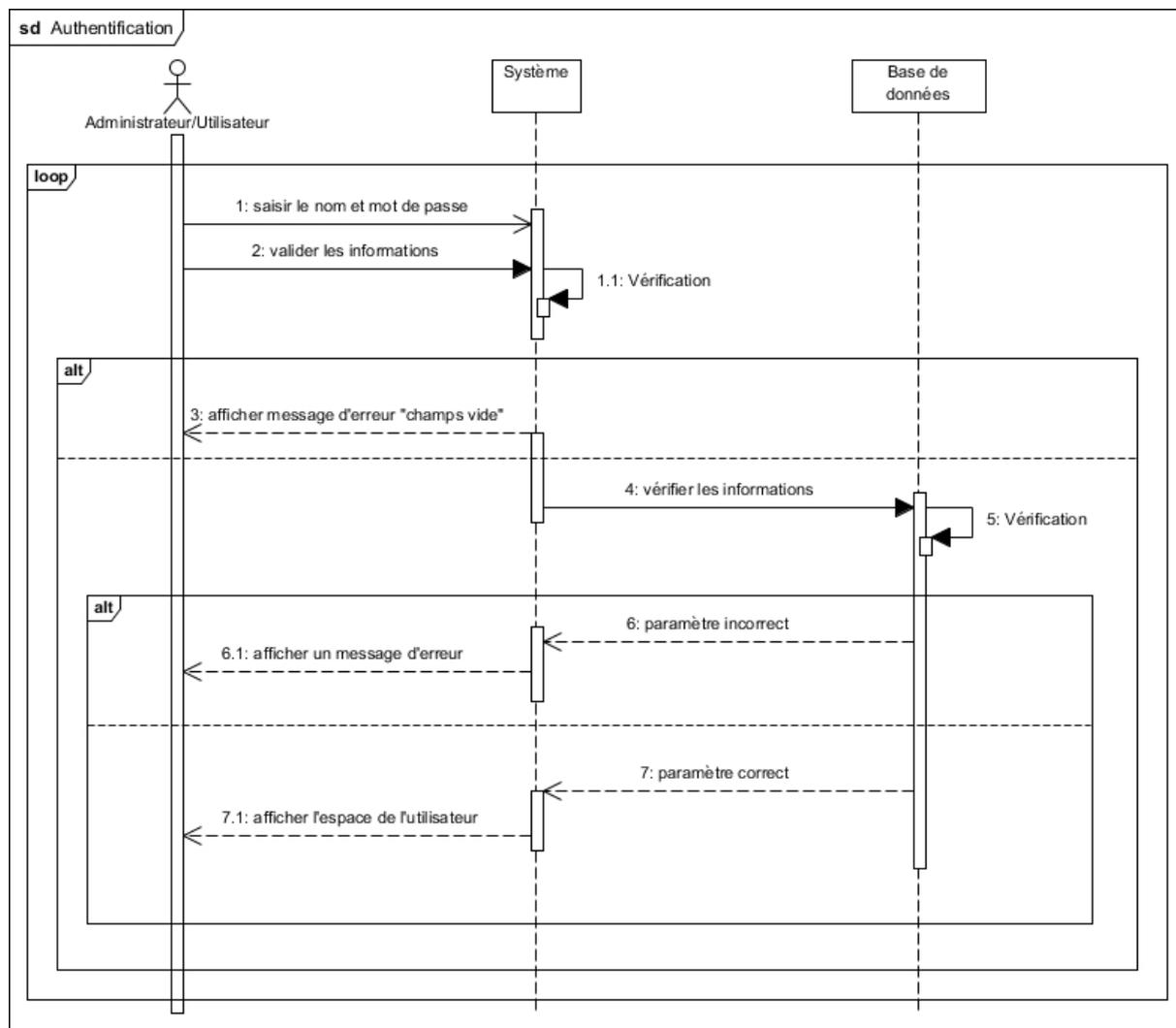


FIGURE 3.1 – Diagramme de séquence système « Authentification »

3.2.2 Diagramme de séquence système « Ajouter un utilisateur »

Après s'être authentifié, l'administrateur pourra accéder à son espace de travail, pour ajouter un utilisateur il doit choisir l'option « ajouter un utilisateur » et remplir le formulaire. Le système commence par vérifier si tous les champs sont bien remplis et bien correctes, si ce n'est pas le cas, il affichera un message d'erreur selon le cas (soit pour remplir les champs vides ou bien pour corriger les erreurs). Si les champs sont bien remplis et les réponses sont correctes, le système effectuera une vérification sur l'existence

de l'utilisateur, s'il existe déjà dans la base de données, il affichera un message d'erreur, sinon l'utilisateur sera ajouté avec succès.

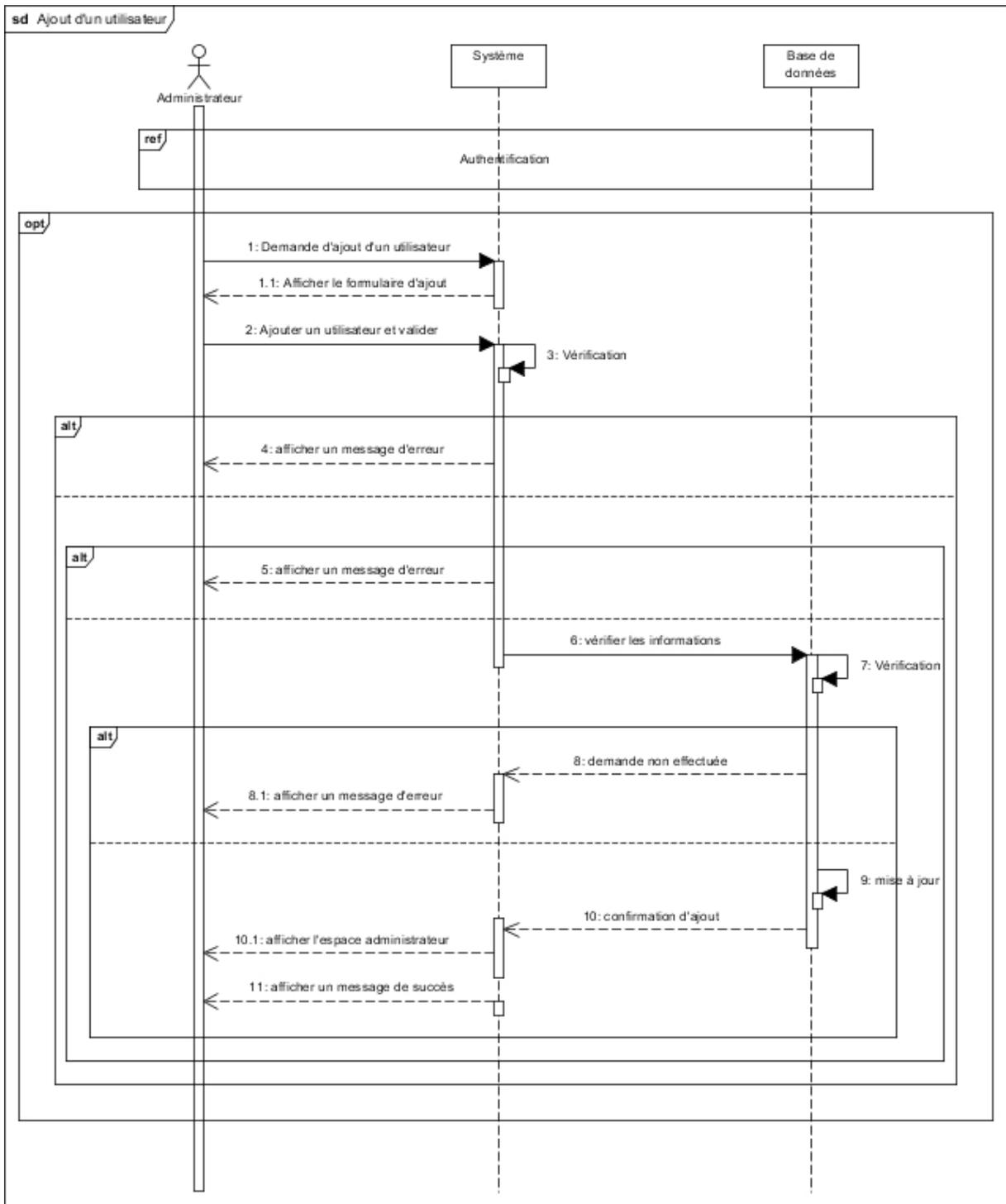


FIGURE 3.2 – Diagramme de séquence système « Ajouter un utilisateur »

3.2.3 Diagramme de séquence système «Rechercher un utilisateur»

Etant données qu'il existe plusieurs employés au niveau de BL, l'administrateur peut rencontrer des difficultés pour rechercher un employé donnée en déroulant toute la liste des utilisateurs. Pour remédier à ce problème nous avons rajouté une barre de recherche.

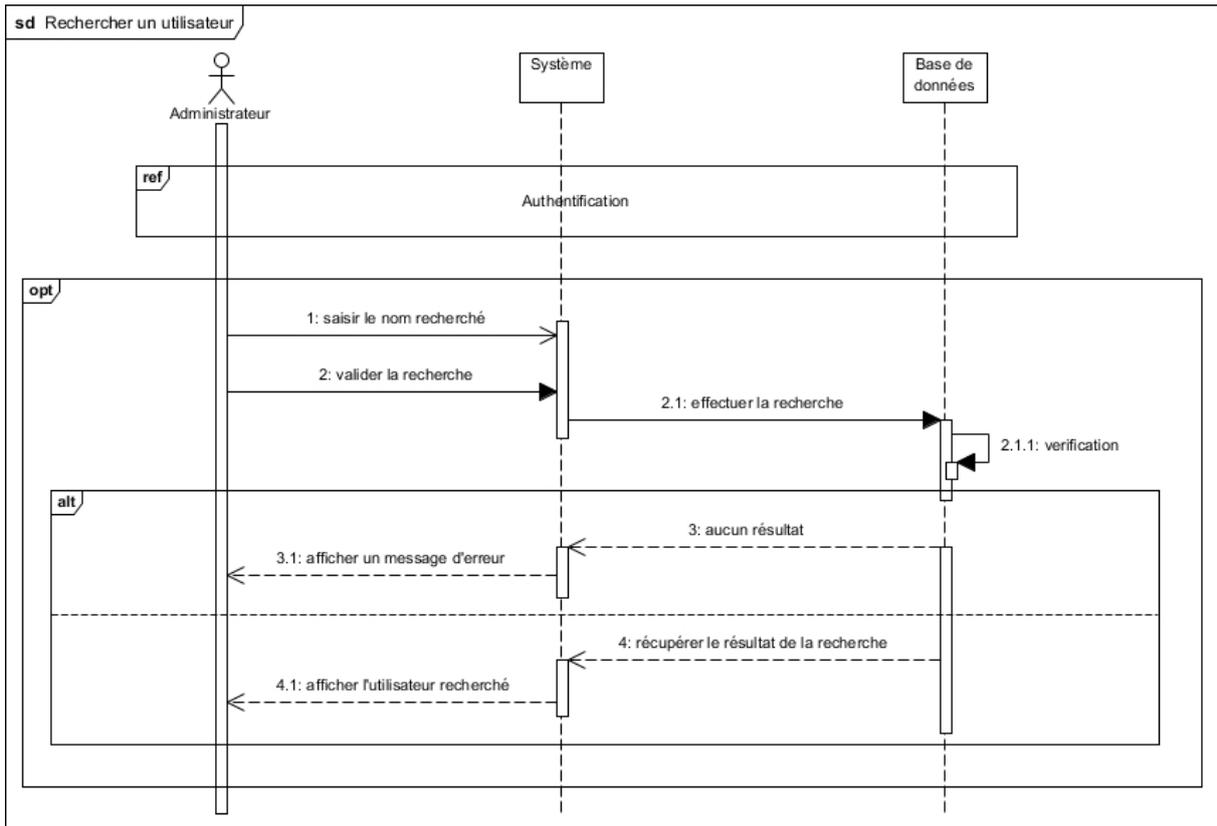


FIGURE 3.3 – Diagramme de séquence système « Rechercher un utilisateur »

3.2.4 Diagramme de séquence système «Modifier un utilisateur

»

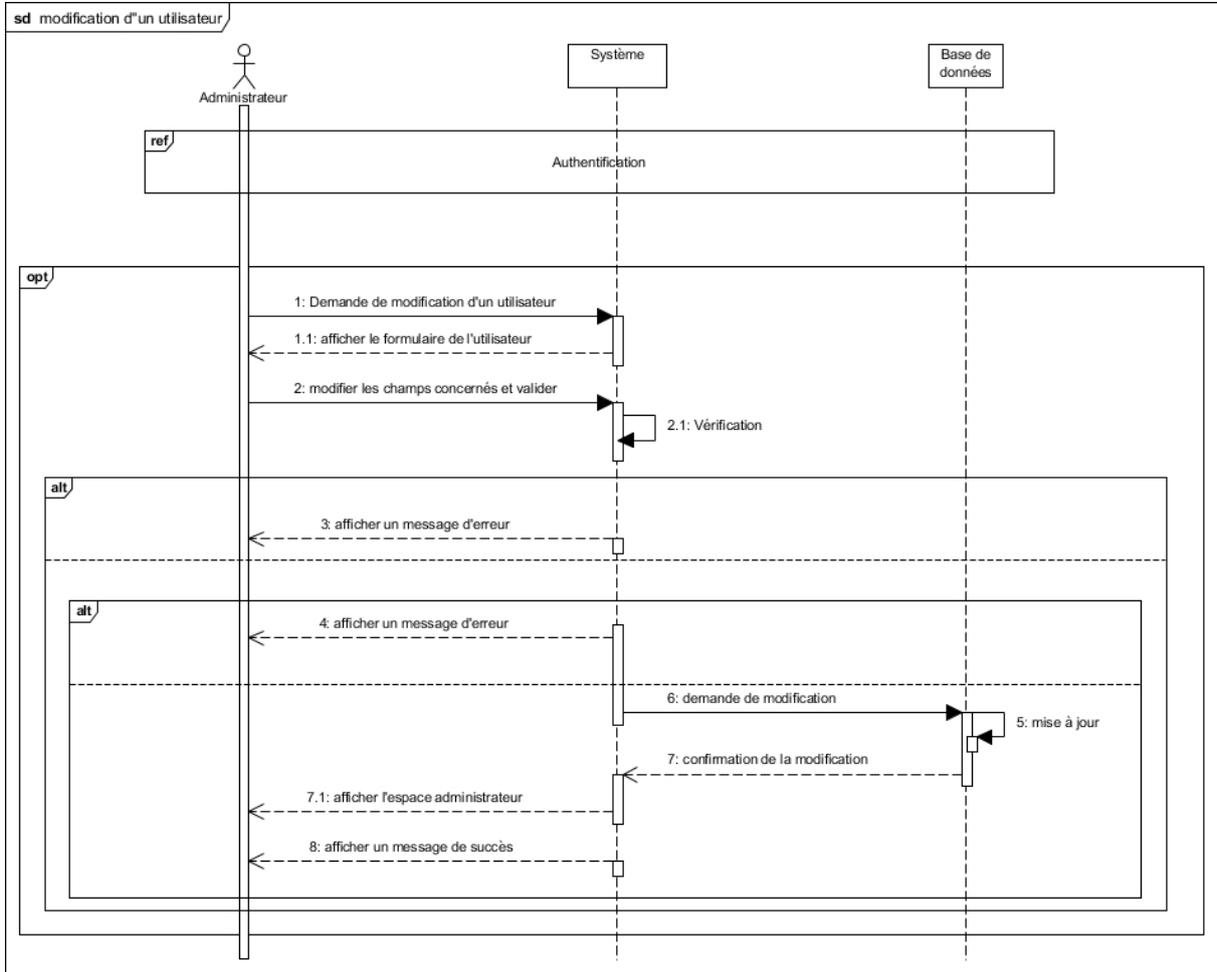


FIGURE 3.4 – Diagramme de séquence système « Modifier un utilisateur »

3.2.5 Diagramme de séquence système «Supprimer un utilisateur»

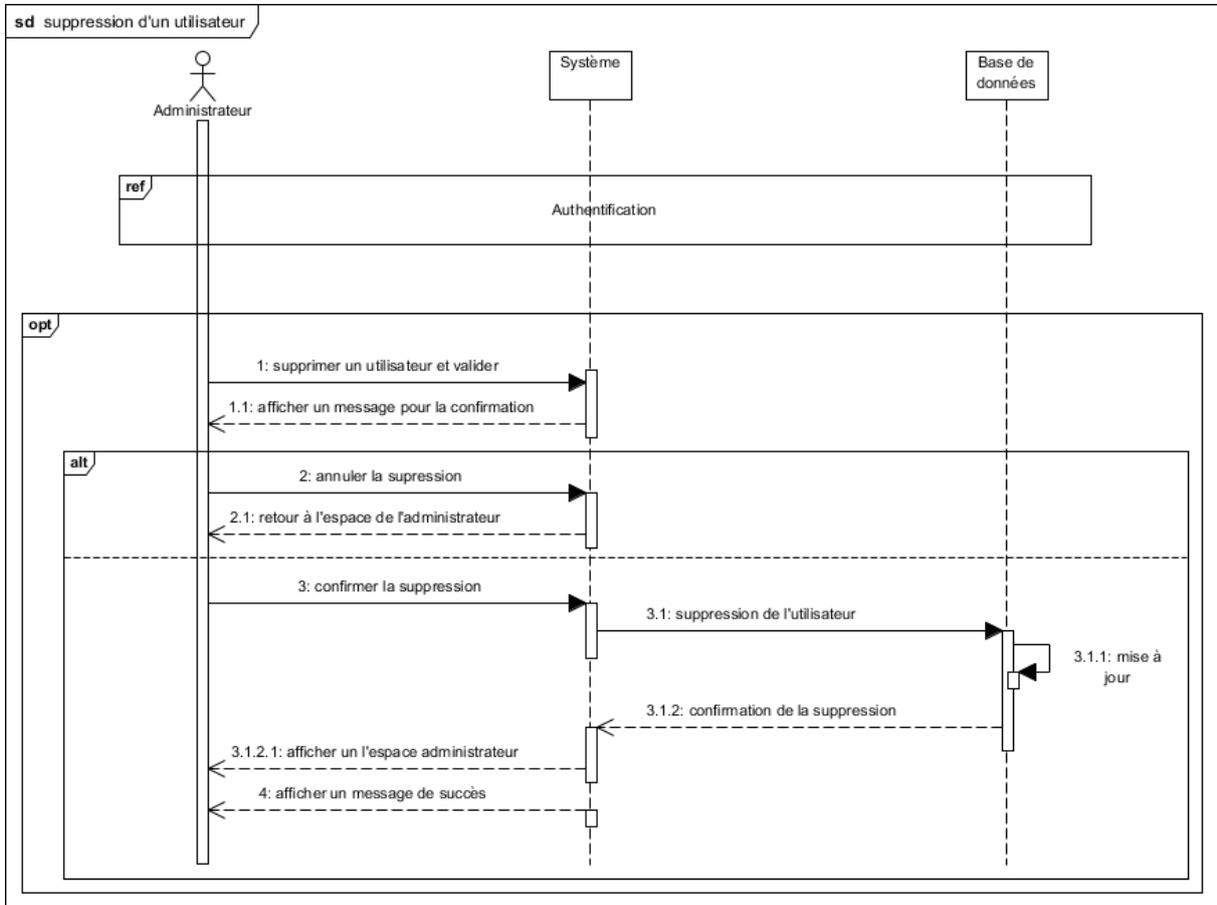


FIGURE 3.5 – Diagramme de séquence système « Supprimer un utilisateur »

3.2.6 Diagramme de séquence système « Planifier les commandes »

»

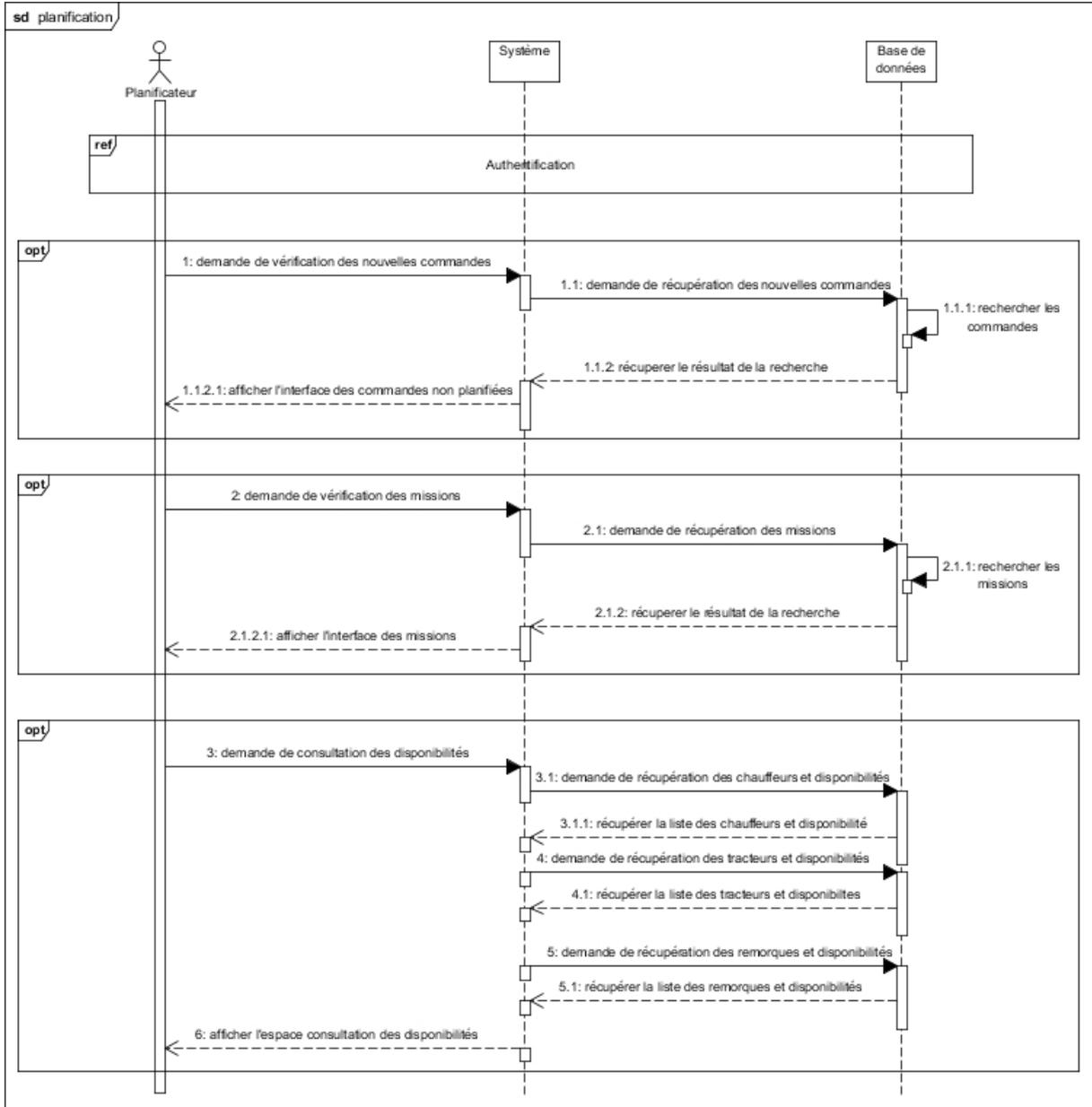


FIGURE 3.6 – Diagramme de séquence système « Planifier les commandes »

3.2.7 Diagramme de séquence système « Vérification des nouvelles commandes »

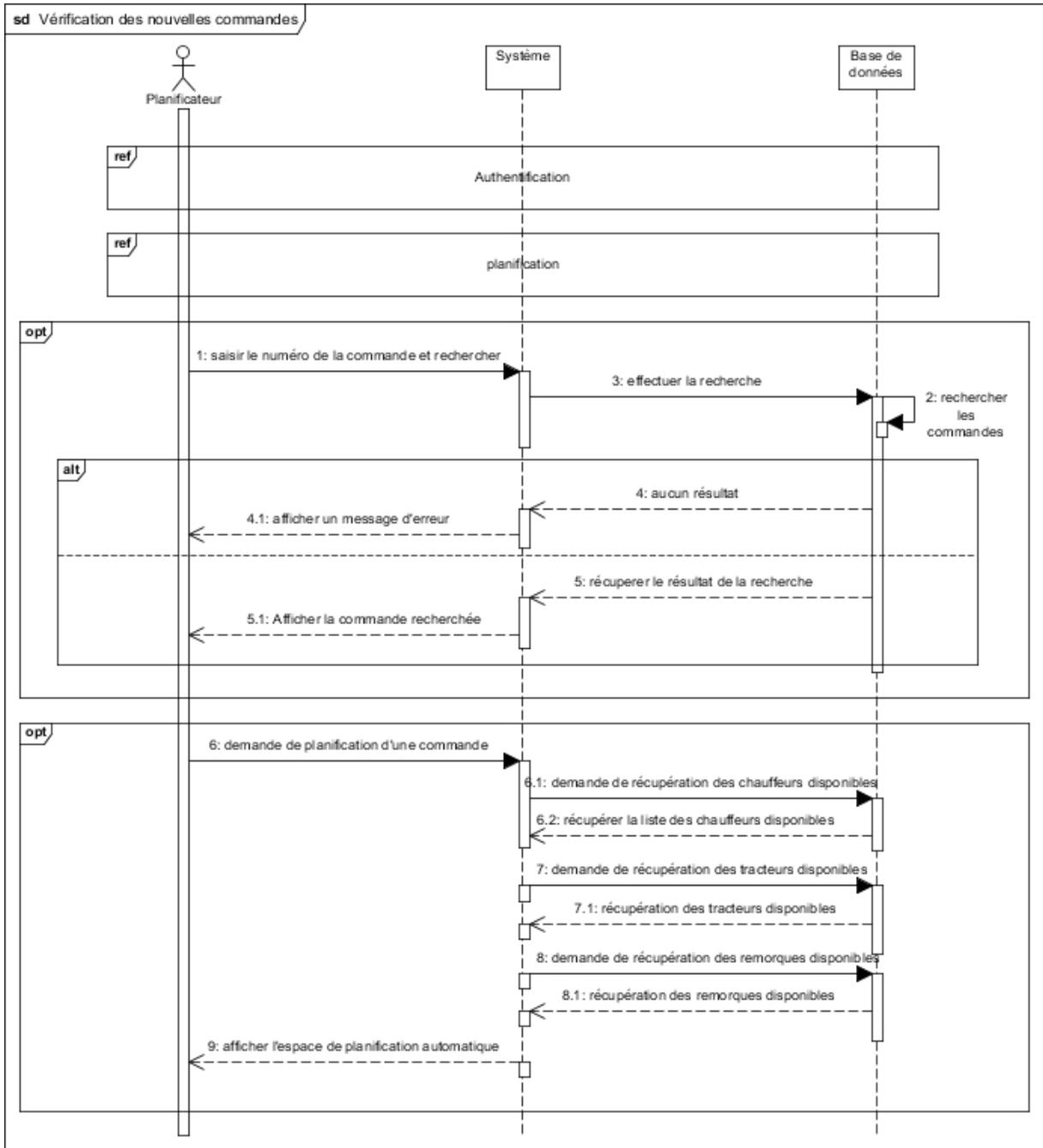


FIGURE 3.7 – Diagramme de séquence système « Vérification des nouvelles commandes »

3.2.8 Diagramme de séquence système « Planification automatique »

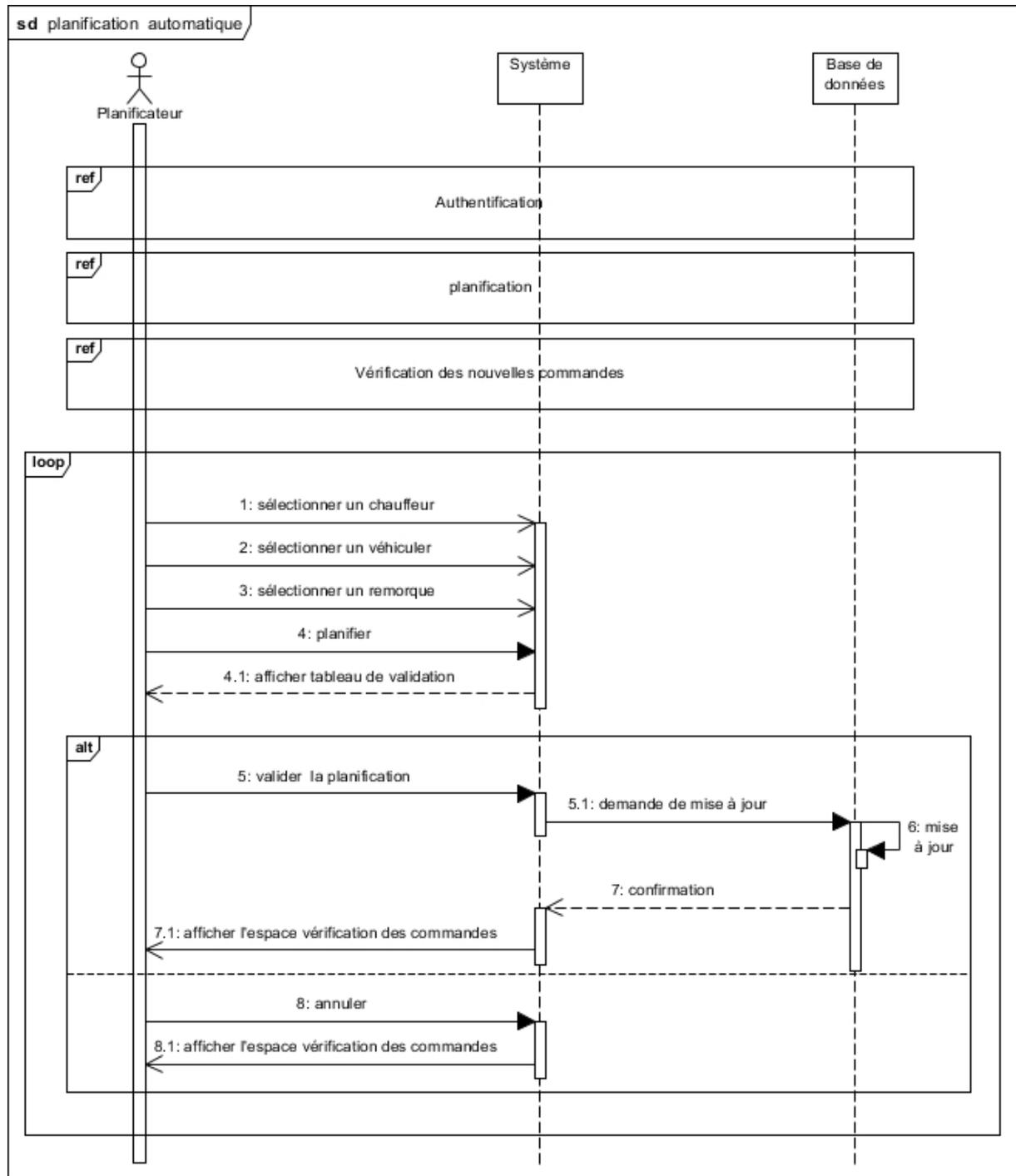


FIGURE 3.8 – Diagramme de séquence système « Planification automatique »

3.4 Modèle relationnel

Dans cette partie, nous réaliserons le modèle logique de données qui est le modèle relationnel, et ce à partir du diagramme de classe. Ce modèle est une manière de modéliser les informations contenues dans une base de données en suivant des principes mathématiques et des règles de passage bien défini.

- **Sous-traitant** (IdSTraitant, Nom, Prenom, Adresse, NumeroTelephone, Email).
- **Client initial** (IdClientInitial, Nom, Prenom, Adresse, NumeroTelephone, Email).
- **Client final** (IdClientFinal, Nom, Prenom, Adresse, NumeroTelephone, Email, #IdClientInitial).
- **Chauffeur** (IdChauffeur, Nom, Prenom, Adresse, NumeroTelephone, Email, CategorieCh).
- **Service client** (IdService, NomService, #IdClientInitial).
- **Véhicule** (IdVehicule, ImmatriculationVE, Categorie, #IdVehiculeType, #IdStraitant).
- **Véhicule interne** (IdVehiculeInterne, ImmatriculationTR, CodeTracteur, Dimension, PoidsMaxRemorqué, PoidsRemorque, PoidsTotalEnCharge, CodeRemorque, DimensionRemorque, CategorieRT).
- **Date CV** (idDateCV, DateDebut, DateFin, #IdChauffeurInterne, #IdVehicule).
- **Date TR** (IdDateTR, DateDebut, DateFin, #IdTracteur, #IdRemorque).
- **Type Véhicule** (IdVehiculeType, NomType).
- **Entête commande** (IdEnteteCommande, Reference, #IdClientInitial, #IdService).
- **Détail commande** (IdDetailC, #IdEnteteCommande, #TypeVehicule, #IdClientFinal, #IdDestination).
- **Destination** (IdDestination, #IdWilaya, #IdDaira).
- **Trajet** (IdTrajet, Distance, Duree, Description, #IdDestination).
- **Wilaya** (IdWilaya, CodeWilaya, NomWilaya).
- **Daira** (IdDaira, NomDaira, #IdWilaya).
- **Type indisponibilité** (IdTypeIndispo, EnPanne, EnMaintenance, EnMission, CongeMaladie, CongeAnnuel, Repos, categorie).

- **Planification** (IdPlanification, DatePlanification, #IdDetailCommande, #IdChauffeurInterne, #Idvehicule).
- **Mission** (IdMission, DateDebut, DateFin, HeureDebut, HeureFin, #IdClientInitial, #IdClientFinal, #IdChauffeurInterne, #IdDestination, #IdFeuilleDeRoute, #IdDetailC, #IdPlanification).
- **Feuille de route** (IdFeuilleDeRoute, DateDepartPrevu, DateDepartReele, DateFinPrevu, DateFinReele, KmDebut, KmFin, #IdChauffeurInterne, #IdSTraitant, #IdVehiculeType, #IdVehicule).
- **Indisponibilité véhicule** (IdInDispoV, DateDebut, DateFin, #IdTypeIndispo, #IdVehicule).
- **Indisponibilité chauffeur** (IdInDispoCh, DateDebut, DateFin, #IdTypeIndispo, #IdChauffeur).

3.5 Conclusion

Ce chapitre nous a permis non seulement de décrire en détail le fonctionnement de notre application, grâce aux différents diagrammes de séquence système. Mais aussi d'illustrer la structure des éléments qui constituent notre base de données, grâce au diagramme de classe et au modèle relationnel. Dans le chapitre suivant nous allons présenter la dernière phase de notre projet qui est bien l'étape de réalisation.

Chapitre 4

Réalisation

4.1 Introduction

Après avoir abordé la partie analyse et conception appropriée à notre projet, nous entamerons la dernière phase qui est la réalisation de notre application.

Tout au long de ce dernier chapitre, nous parlerons de la réalisation de notre application, nous présenterons dans un premier lieu les environnements et les outils de développement comme le Framework Spring boot, le langage Java EE, et nous exposerons par la suite quelques captures d'écran de notre application réalisée.

4.2 Langage et environnements de développement

Dans cette section, nous énumérerons les différentes technologies qui sont utilisées pour développer notre système.

4.2.1 Langages de développement

4.2.1.1 Technologies Back end

1. Langage PHP :

PHP (HyperText Preprocessor) est un langage de script utilisé le plus souvent côté serveur. Sa syntaxe est principalement inspirée du langage C et JAVA. Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le WEB [19].

2. Langage Java :

Java définit à la fois un langage de programmation orienté objet et une plateforme informatique développée par Sun Microsoft, elle permet de créer des applications client-serveur.

Les applications développées en java peuvent fonctionner sur différents systèmes d'exploitation, comme Windows ou Mac OS (Operating System). Nous utilisons ce langage car il offre un ensemble de bibliothèques qui facilitent le développement web et dispose d'une documentation bien fournie [20].

3. Langage SQL :

SQL (Structured Query Language) est un langage standard des systèmes de gestion de bases de données relationnelles. Il permet d'élaborer les requêtes d'interrogation de la base de données. Ce langage est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web [21].

4. J2EE :

J2EE (Java Edition Entreprise) ou Java Entreprise Edition est une technologie développée par SUN puis Oracle qui facilite le développement d'applications d'entreprise distribuées en java, destinées à des plateformes web. C'est une norme décrivant tous les éléments pour le fonctionnement d'une application distribuée. Dans ce cadre, la sécurité n'a pas été négligée. C'est le langage le plus adopté par les développeurs grâce à sa fiabilité et sa performance élevée [22].

4.2.1.2 Technologies Front end

1. HTML :

Le langage HTML (HyperText Markup Language) est un langage de description conçu pour la représentation des pages web. C'est un langage de balise qui fonctionne suivant l'assemblage et la combinaison de balises et qui permet de structurer et de donner l'apparence voulue aux données textes, images et multimédias, hyperliens, . . . etc. Il est souvent lié au CSS (Cascading Style Sheets) [23].

2. CSS :

Aussi appelé « Feuilles de style », c'est un langage de mise en forme d'un document HTML, autrement dit, du contenu des pages web. Le principal enjeu de CSS est

de pouvoir décomposer l'architecture d'un document de sa forme. Il nous permet également de définir le rendu stylistique d'un document selon les préférences de l'utilisateur [24].

3. Langage Java Script :

JavaScript est un langage de script principalement employé dans les pages web interactives, il permet d'améliorer l'ergonomie d'une interface applicative utilisateur et d'intégrer des effets esthétiques.

JavaScript joue un rôle dans l'amélioration de la rapidité de chargement des pages web, c'est un critère d'ergonomie et de référencement de plus en plus important [25].



FIGURE 4.1 – Logo JavaScript

4.2.2 SGBDR (Système de Gestion de Base de Données Relationnelles) et Serveurs web

4.2.2.1 WampServer

Wamp est une des technologies les plus courantes utilisée pour la création des sites dynamiques qui se basent sur le langage de script PHP et sur une base de données MySQL (Structured Query Language), il se base également sur le serveur Apache qui se charge du lancement du serveur web dans Windows.

Ces trois composants étant des composants Open Source peuvent être installés individuellement, mais pour faciliter la tâche au développeur le paquet « WampServer » fournit un moyen convivial de l'installation et de la configuration des composants AMP (Apache, MySQL, PHP) sous Windows [26].

4.2.2.2 Apache Tomcat

Apache est un serveur http (Hyper Text Transfert Protocol) créé et maintenu au sein de la fondation apache, gratuit, extensible, configurable et performant pour des besoins matériels. Il est diffusé en open source sous une licence Apache, il a été écrit en langage JAVA.

Tomcat est un conteneur de Servlet J2EE qui implémente la spécification des Servlets et des JSP (JavaServer Pages) de Sun Microsystems. Il se charge de compiler les pages JSP avec Jasper pour en faire des Servlets.

Apache est une implémentation open source d'un conteneur web qui permet d'exécuter des applications web reposant sur les technologies servlet et JSP [27].

4.2.2.3 MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) open source. Son rôle consiste à stocker et gérer une grande quantité de données en les organisant sous forme de table, et ce, pour améliorer les performances et la durabilité. MySQL est utilisé dans une large gamme d'applications et fonctionne pratiquement sur tous les systèmes d'exploitation y compris Linux et Windows [28].



FIGURE 4.2 – Logo MySQL

4.2.2.4 PhpMyAdmin

PhpMyadmin est une application web de gestion pour les systèmes de gestion de bases de données MySQL, il consiste en un ensemble de scripts PHP rendant plus conviviale l'administration de bases de données MySQL en passant par un navigateur web [29].

4.2.3 Les composants logiciels utilisés

4.2.3.1 Spring boot

Sous projet Spring qui permet non seulement à l'utilisateur de se focaliser uniquement sur les tâches techniques en configurant automatiquement l'application, mais aussi il propose une fonctionnalité importante qui est les Starters. Celle-ci apporte au projet un ensemble de dépendances qui facilite la gestion des versions d'une part, et en d'autre part elle permet de créer un squelette prêt à l'emploi rapidement.

Spring Boot est donc un Framework qui facilite le développement d'applications fondées sur Spring en gagnant du temps et de la productivité [30].



FIGURE 4.3 – Logo SpringBoot

4.2.3.2 Bootstrap

Bootstrap est un Framework CSS créé par Twitter et mis en open source en 2012, fait partie de ces outils qui accélèrent grandement un projet web. Il étend les possibilités du langage CSS en ajoutant plein de fonctionnalités et de composants, permettant de faciliter la création d'applications web aux développeurs [31].

4.2.3.3 Thymeleaf

Thymeleaf est un moteur de Template Java côté serveur pour les environnements web qui fournit un support pour la génération de documents HTML, XHTML, JavaScript, et CSS. Ce dernier permet de créer des Templates respectant le format du document et s'intègre parfaitement dans une application Spring.

Thymeleaf vise principalement à apporter des modèles élégants et naturels au flux de développement (html) qui peut à la fois fonctionner comme prototypes statiques et peut être affiché correctement dans les navigateurs [32].



FIGURE 4.4 – Logo Thymeleaf

4.2.3.4 Hibernate

Hibernate est un Framework permettant de gérer la persistance d'objets dans une BDD (Base De Données), un objet persistant étant un objet destiné à être conservé pour une durée plus longue que celle de l'exécution du programme [33].

4.2.4 Environnement de modélisation et de développement

4.2.4.1 IntelliJ IDEA

C'est un IDE (environnement de développement intégré) développé par JetBrains, qui permet de travailler sur toutes sortes d'applications en Java et dans d'autres langages de la JVM (Java Virtual Machine) tels que Kotlin, Scala et Groovy.

Le principal but d'IntelliJ IDEA est d'améliorer la productivité des développeurs, c'est pourquoi elle se démarque par sa flexibilité, sa simplicité, et sa conception solide [34].

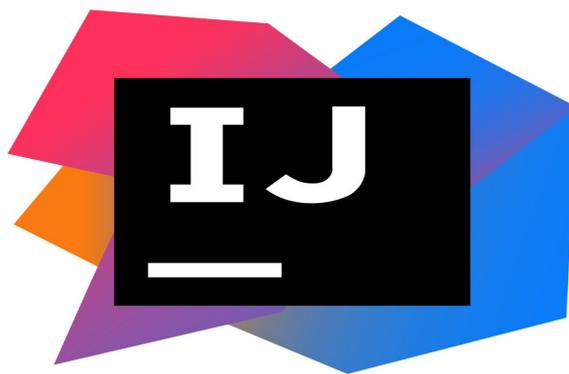


FIGURE 4.5 – Logo IntelliJ IDEA

4.3 Architecture de développement

La conception de l'architecture est une phase particulièrement importante du développement d'un logiciel. Elle assure sa stabilité, son efficacité et sa pérennité. La bonne compréhension du besoin métier et des contraintes fonctionnelles et non fonctionnelles du logiciel aboutit à choisir une architecture adéquate. Parmi les architectures qui existent, on trouve le modèle MVC (Model View Controller) qui est l'architecture suivie tout au long de notre développement [35].

4.3.1 Le Patron Modèle-Vue-Contrôleur

L'architecture MVC consiste à découper son code pour qu'il appartienne à l'une des trois composantes du MVC. Cette architecture est l'une des architectures logicielles les plus utilisées pour les applications Web, elle se compose de trois modules :

- **Modèle** : c'est le noyau de l'application qui contient les données de l'application et la logique métier, permet de récupérer les informations dans la base de données, de les organiser pour qu'elles puissent ensuite être traitées par le contrôleur. La composante modèle n'a aucune connaissance de l'interface graphique [36].
- **Vue** : composant graphique de l'interface qui permet de présenter les données du modèle à l'utilisateur. Contient tout ce qui est visible à l'écran et qui propose une interaction avec l'utilisateur. Par exemple, les boutons, les images, les zones de saisie, . . . etc [36].
- **Contrôleur** : module responsable des prises de décision, gère la logique du code qui prend des décisions, il est l'intermédiaire entre le modèle et la vue [36].

Exemple :

Pour mieux comprendre la manière dans laquelle un utilisateur interagit avec l'application, voici un exemple qui illustre comment un utilisateur répond à une question :

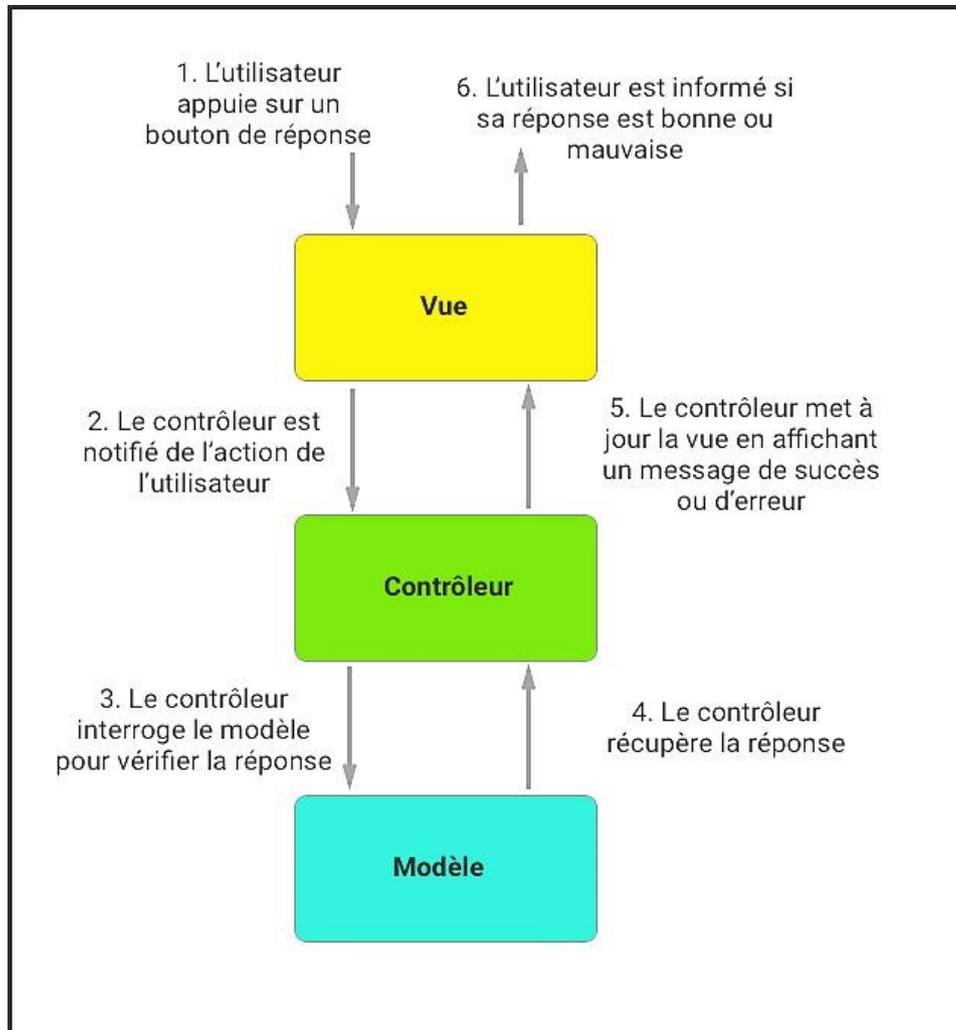


FIGURE 4.6 – Mise en application du MVC

4.3.2 Implémentation d'un projet en J2EE suivant le model MVC

Le modèle MVC est implémenté dans un projet JEE, en utilisant les concepts (ou les composants) suivants :

- **Servlet** : un servlet est une classe Java capable de recevoir une requête HTTP du client (navigateur web), ensuite effectue des traitements et renvoie le résultat [37].
- **JSP** : c'est une technique qui se base sur java et qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de pages web. C'est donc les pages servant à générer l'ensemble du code HTML de l'interface utilisateur [38].
- **Les JavaBean** : un JavaBean désigne tout simplement un composant réutilisable. Autrement dit, ce sont des classes java conçues pour être réutilisables lors des développements en java. Les composants JavaBeans permettent une séparation entre le code Java et la gestion de l'affichage des données. (cad : gestion de l'affichage des données) [39].

La figure suivante illustre les composants de J2EE :

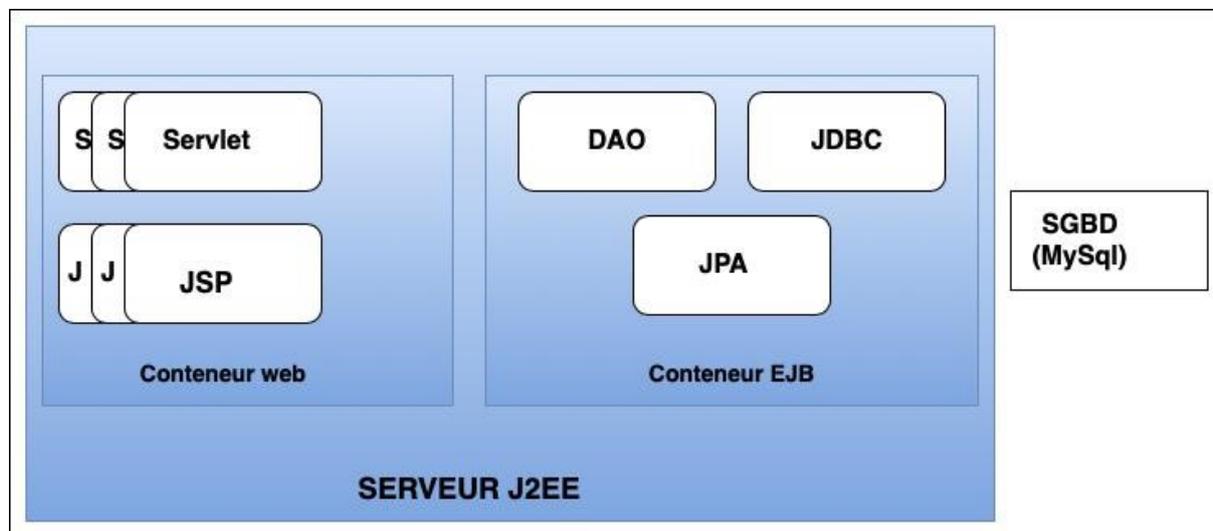


FIGURE 4.7 – Architecture de J2EE

- **Conteneur web** : également connu sous le nom conteneur de servlet, il représente les servlets et les JSP qui constituent les blocs de construction du développement d'application web avec J2EE. Conteneur web est la composante d'un serveur web qui interagit avec les servlets java.

Il est chargé de l'initialisation et de la gestion de la durée de vie des servlets Java et des pages JSP [40].

- **Conteneur EJB (Entreprise Java Bean)** : C'est une architecture de composants logiciels coté serveur pour la plateforme de développement J2EE, il se compose de :
 - o **JPA (Java Persistence Api)** : fournit un langage de requête (également appelé JPQL), qui peut être utilisé pour manipuler des objets sans écrire des requêtes SQL spécifique à la base de donnée concernée [41].
 - o **DAO (Data Access Object)** : Elle est composée d'un ensemble d'interfaces locales et distantes. Les DAO permettent d'accéder aux objets et proposent des méthodes de CRUD (CREATE READ UPDATE DELETE) [42].
 - o **JDBC (Java DataBase Connectivity)** : c'est une technologie qui désigne une API (Application Programming) Interface java pour connecter et exécuter des requêtes sur une base de données [41].
- **SGBD (système de gestion de base de données)** : est un système permettant de gérer et de manipuler la base de données. Il est défini comme un lieu de stockage des données dans des tables séparées [43].

4.4 Captures d'écran de l'application

Cette partie permet de visualiser les conditions réelles d'utilisations de l'application.

- La figure 4.8 représente l'interface qui permet à l'utilisateur de s'authentifier en introduisant un nom d'utilisateur et un mot de passe

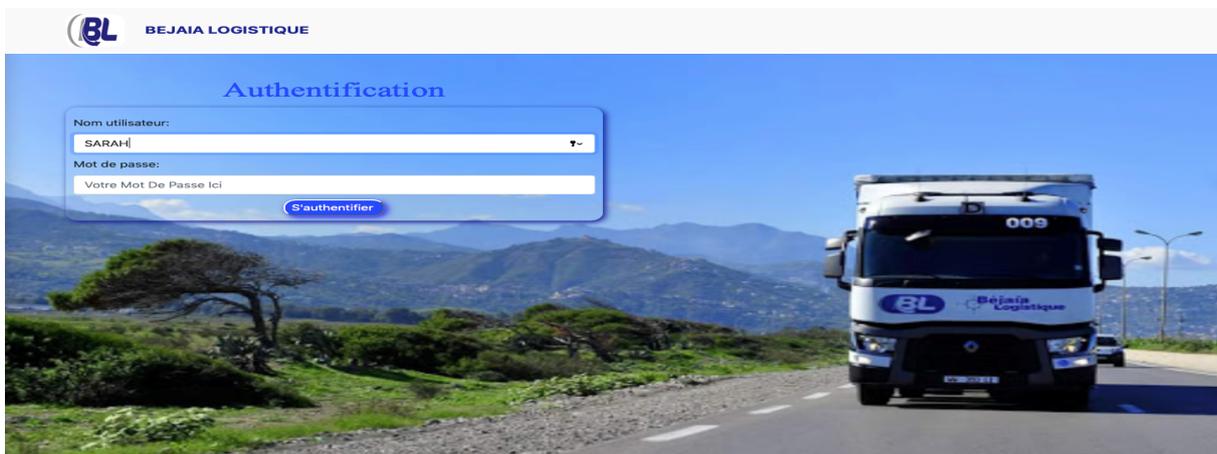


FIGURE 4.8 – Interface d'authentification.

- Après l'authentification, chaque utilisateur se retrouve dans espace de travail. La figure suivante représente l'espace de travail de l'administrateur.

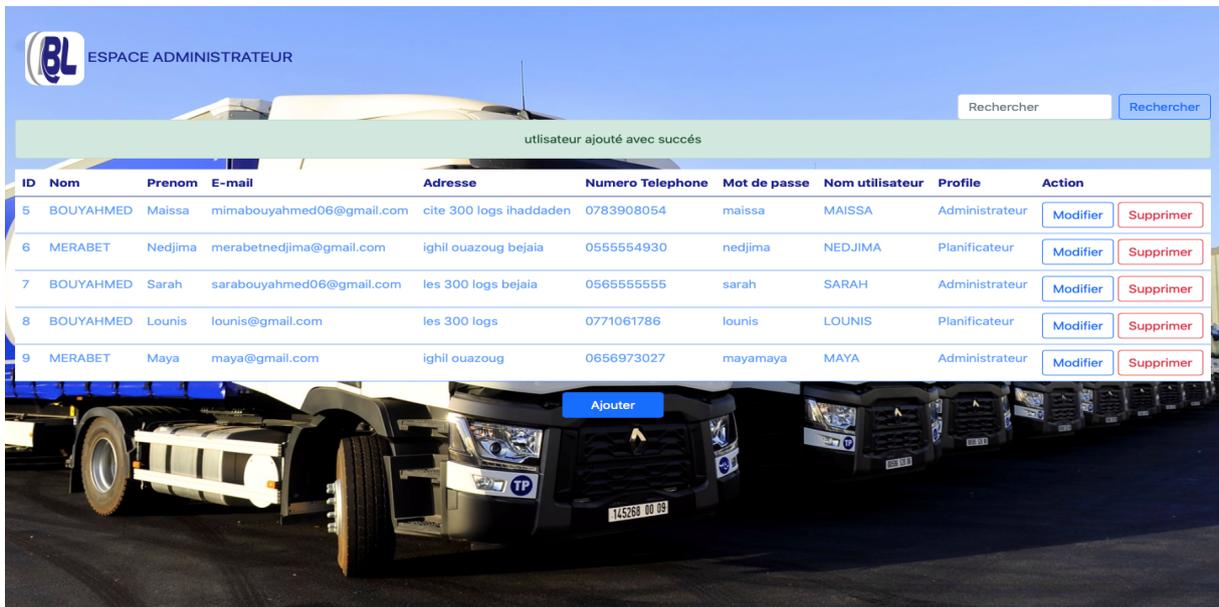


FIGURE 4.9 – Interface liste des utilisateurs.

- La figure 4.10 représente le formulaire d'ajout d'un utilisateur.

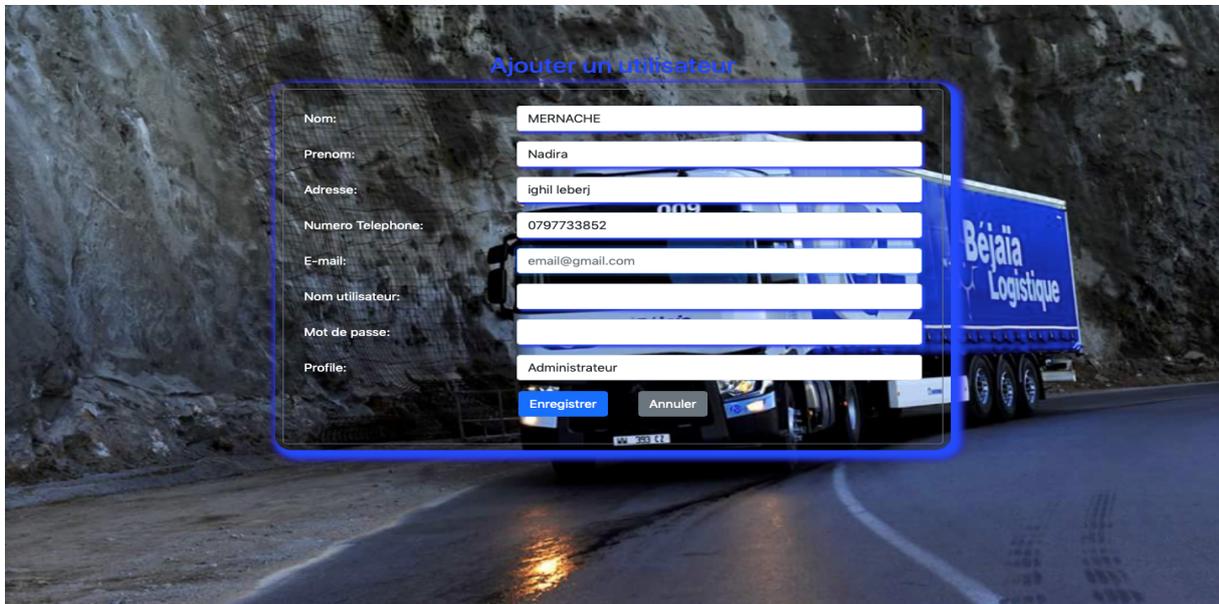


FIGURE 4.10 – Interface formulaire d'ajout.

- La figure 4.11 représente le formulaire qui permet de modifier une information donnée d'un utilisateur déjà existant.

Modifier l'utilisateur (5)

Nom:	<input type="text" value="BOUYAHMED"/>
Prenom:	<input type="text" value="Maïssa"/>
Adresse:	<input type="text" value="cite 300 logs ihaddaden"/>
Numero Telephone:	<input type="text" value="0783908054"/>
E-mail:	<input type="text" value="mimabouyahmed06@gmail.com"/>
Nom utilisateur:	<input type="text" value="MAÏSSA"/>
Mot de passe:	<input type="password" value="*****"/>
Profile:	<input type="text" value="Administrateur"/>

FIGURE 4.11 – Interface de modification.

- La figure suivante représente la première interface affichée au planificateur après son authentification.



FIGURE 4.12 – Interface planificateur.

- La figure ci-dessous représente l'interface associée à la vérification des nouvelles commandes.



FIGURE 4.13 – L’interface de vérification des nouvelles commandes.

- Après avoir choisi de planifier une commande précise, l’interface suivante sera affichée afin de visualiser tous les chauffeurs, tracteurs, et les remorques disponibles.



FIGURE 4.14 – Interface liste des disponibilités.

- La figure suivante représente une interface de validation d’une planification précise après avoir sélectionné un tracteur, un chauffeur, et une remorque et avoir cliqué sur planifier.

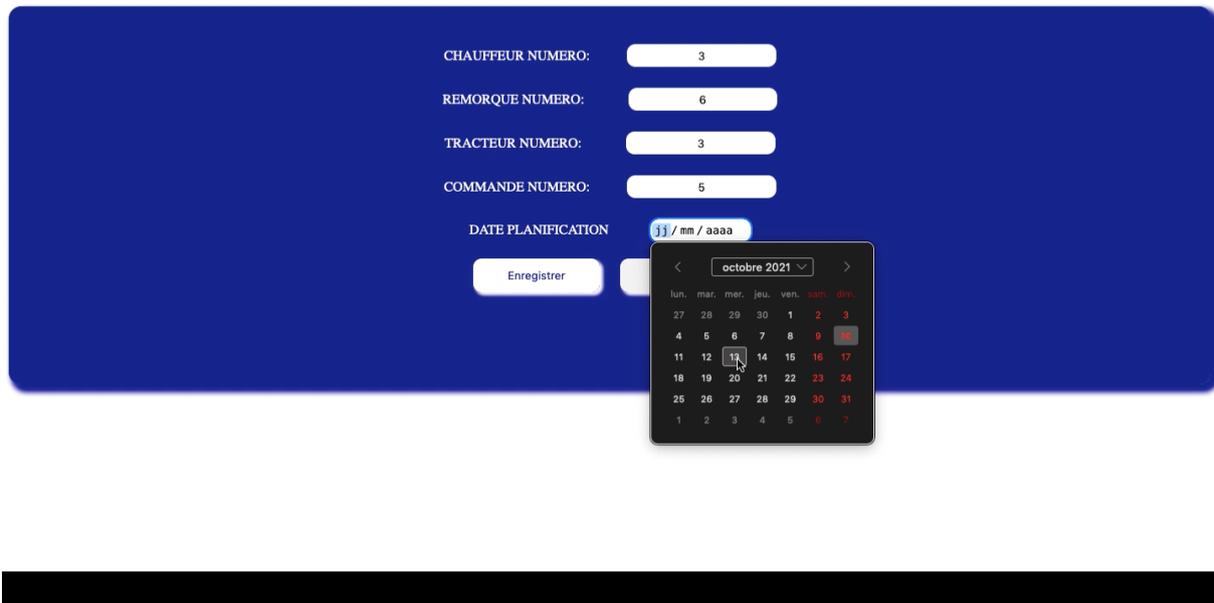


FIGURE 4.15 – Interface de validation d’une planification.

4.5 Conclusion

Dans ce chapitre dédié à la réalisation de notre application, nous avons pu mener à l’existence ce qui était conceptuel. Pour atteindre ce but, nous avons commencé par la présentation des outils et des environnements de développement utilisés, par la suite nous avons présenté quelques prises d’écrans des interfaces de notre application.

Conclusion générale

Nous sommes parvenus grâce à ce projet à concevoir et à mettre en œuvre une application web qui consiste en un module de planification d'un TMS déjà existant destiné à l'entreprise Bejaia logistique qui active dans le domaine du transport routier en Algérie.

Nous avons commencé notre travail par recenser les lacunes et les difficultés rencontrées par cet organisme au quotidien, afin de spécifier les besoins fonctionnel et non fonctionnel d'une part, et en d'autre part pour bien définir les objectifs à atteindre.

Pour la conception de notre application, nous nous sommes appuyés sur le processus de développement UP et le langage UML pour l'élaboration des différents diagrammes à savoir : le diagramme de contexte, les diagrammes de cas d'utilisations, diagrammes de séquences, et le digramme de classe.

Concernant la réalisation de notre application, nous avons utilisé des technologies choisis parmi tant d'autres comme Java EE et SQL et cela en faisant recours au Framework Spring boot.

Ce thème étant très instructif sur le plan pédagogique et très intéressant sur le plan technologique, nous a permis d'avoir une expérience très enrichissante. En effet, il nous a permis non seulement d'appliquer nos connaissances théoriques acquises tout au long de notre cursus universitaire, mais aussi d'accroître nos connaissances et surtout d'avoir un aperçu sur le monde professionnel.

Enfin, comme tout autre projet de développement logiciel non fini, des perspectives d'avenir s'imposent. Dans notre cas nous souhaiterons :

- Finir l'étape de réalisation de l'application en suivant notre conception.
- Améliorer les interfaces pour répondre aux critères de l'ergonomie.
- Intégrer le diagramme de Gantt pour la vérification des disponibilités des chauffeurs, tracteurs, et des remorques.

Bibliographie

- [1] Facon, P. (s.d.). Le Coin des Entrepreneurs. Consulté le 04 18, 2021, sur WWW.LECOINDESENTREPRENEURS.FR/LOGISTIQUE-ENTREPRISE/.
- [2] GAUTHIER, Y. (1991). La logique interne. Paris : Librairie Philosophique J.VRIN.
- [3] Mecalux. (s.d.). Consulté le 04 19, 2021, sur <https://www.mecalux.fr/blog/logistique-interne-definition>.
- [4] Logistique Conseil. (s.d.). Consulté le 04 19, 2021, sur <http://www.logistiqueconseil.org/Articles/Logistique/Logistique-industrielle.htm>.
- [5] Logistique et transport. (s.d.). Consulté le 04 19, 2021, sur <https://lojistike.blogspot.com/2018/02/les-differents-types-de-logistique.html>.
- [6] lotfi, A. S. (2016). la logistique de distribution des produits agroalimentaires. Bejaia : Université abderrahman mira.
- [7] HENSHER.D, B. e. A handbook of logistic and supply chain management. New york : 2001.
- [8] Oracle. (s.d.). Consulté le 04 20, 2021, sur [HTTPS://WWW.ORACLE.COM/CH-FR/SCM/WHAT-IS-TRANSPORTATION-MANAGEMENT-SYSTEM/](https://WWW.ORACLE.COM/CH-FR/SCM/WHAT-IS-TRANSPORTATION-MANAGEMENT-SYSTEM/).
- [9] Bejaia logistique. (s.d.). Consulté le 04 20, 2021, sur <https://www.bejaialogistique.com/presentationmain>.
- [10] CONALLEN, J. (2000). Concevoir des applications web UML. Eyrolles.
- [11] P.Roques, F. (2004). UML en action de l'analyse des desoins à la conception. EYROLLES.
- [12] J.GABAY, D. (2008). UML2 Analyse et conception. DUNOD.
- [13] slideShare. (s.d.). Consulté le 05 02, 2021, sur <https://fr.slideshare.net/oussamabz/pfe-conception-et-developemment>.

- [14] C.Belleil. (2007). Le langage UML 2.0. Université de NANTES.
- [15] C.Saiche, A. (2015). Conception et réalisation d'une application web pour la gestion des étudiants d'une école privée.cas d'étude : "ISA School". Bejaia : Université A/ Mira .
- [16] D. Janiszek. Cours de génie logiciel, Consulté le 26/04/2021.
https://helios2.mi.parisdescartes.fr/~janiszek/_media/enseignement/06_diagramme_cas_d_utilisation.pdf.
- [17] UML les diagrammes de séquence, <http://tvaira.free.fr/dev/uml/UML-DiagrammesDeSequence.pdf>, Consulté le 15/05/2021.
- [18] Cours UML et SysML, Consulté le 04/05/2021.
<https://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/diagramme-de-classe/>.
- [19] E .DASPET ET P .GEYER, PHP 5 AVANCE, 6EME EDITION-EYROLLES, MAI 2012.
- [20] C .DELANNOY, PROGRAMMER EN JAVA, EDITION-EYROLLES, SEPTEMBRE 2004.
- [21] SQL.sh, <https://sql.sh/>, Consulté le 20/06/2021.
- [22] OPENCLASSROUMS, Consulté le 18/06/2021.
<https://openclassrooms.com/fr/courses/2434016-developpez-des-sites-web-avec-java-ee>.
- [23] PIXEES, Consulté le 19/06/2021.
https://pixees.fr/informatiquelycee/n_site/snt_web_html.html.
- [24] MDN Web Docs, Consulté le 19/06/2021.
<https://developer.mozilla.org/fr/docs/Web/CSS>.
- [25] JDN, Consulté le 19/06/2021.
www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/.
- [26] TechLib, Consulté le 20/06/2021.
<https://techlib.fr/definition/wamp.html>.
- [27] Techopedia, Consulté le 20/06/2021.
<https://www.techopedia.com/definition/15735/apache-tomcat>.

- [28] LEMAGIT, Consulté le 22/06/2021.
<https://www.lemagit.fr/definition/MySQL>.
- [29] Consulté le 04/07/2021, sur
<https://www.phpmyadmin.net/>.
- [30] B.NGUIMGO. Spring boot par la pratique. Les éditions du net, 2018.
- [31] JDN, Consulté le 04/07/2021.
www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques//.
- [32] Thymeleaf, Consulté le 18/06/2021. sur
<https://www.thymeleaf.org/>.
- [33] T. HAMON, PROGRAMMATION WEB AVANCEE HIBERNATE, UNIVERSITE PARIS 13, (2012-2013).
- [34] JET BRAINS, Consulté le 04/07/2021.
<https://www.jetbrains.com/help/idea/discover-intellij-idea.html>.
- [35] SoftFluent, Consulté le 16/06/2021.
<https://www.softfluent.fr/blog/architecture-logicielle-pour-application/>.
- [36] Mehdi Bekcaid, Consulté le 16/06/2021.
<https://medium.com/@belcaid.mehdi/larchitecture-logicielle-mvc-1a8bbb5cf6dc>.
- [37] Développons en Java, Consulté le 17/06/2021.
<https://www.jmdoudoux.fr/java/dej/chap-servlets.htm>.
- [38] Développons en Java, Consulté le 17/06/2021.
<https://www.jmdoudoux.fr/java/dej/chap-jsp.htm>.
- [39] OPENCLASSROOMS, Consulté le 17/06/2021.
openclassrooms.com/fr/courses/626954-creez-votre-application-web-avec-java-ee/619935-le-javabean.
- [40] WebDevDesigner, Consulté le 18/06/2021.
webdevdesigner.com/q/difference-between-web-server-web-container-and-application-server-52062/.
- [41] Medium, Consulté le 18/06/2021.
<https://medium.com/code-divoire/les-api-java-ee-642f9ad88d9f>.

[42] DocPlayer, Consulté le 18/06/2021.

<https://docplayer.fr/740129-Creation-d-une-application-jee.html>.

[43] Cours BDD, Consulté le 19/06/2021.

<https://members.loria.fr/AImine/cours/BD.pdf>.