

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 Fin de Cycle

En vue de l'obtention du diplôme Master 2 en informatique

Option : Génie Logiciel

Thème

Systeme d'aide à la décision pour l'optimisation de la planification logistique

Cas d'étude : BEJAIA LOGISTIQUE OUZELLAGUEN wilaya de BEJAIA.

Réalisé par

M^r BELDJOURI Mouloud M^{lle} BOUGHIDEN Nadra

Encadrante :

M^{me} TAHAKOURT Zineb M.Assistante

Jury :

Président du Jury	Examineur	Examineur
M ^{me} ALOUI Soraya	M ^r ACHROUFENE Achour	M ^{me} KHOULALENE Nadjjet
M.C.A	M.C.A	M.C.B

*« Si vous voulez laisser vos empreintes dans le sable du temps, ne traînez pas les
pieds. »*

- Remerciements -

Nous souhaitons exprimer nos sincères remerciements à tous ceux qui ont contribué à la réalisation de ce projet et nous ont permis d'arriver jusqu'à cette étape importante.

Tout d'abord, nous remercions le **Bon Dieu** pour nous avoir accordé la santé et la force nécessaires pour mener à bien ce travail.

Nous adressons également nos remerciements à **notre famille** pour leur soutien, leur encouragement et leur patience tout au long de ce projet.

Nous tenons à remercier notre encadrante, **Madame Tahakourt**, pour ses conseils rigoureux et pertinents qui ont contribué largement à la réalisation de ce projet. Nous avons apprécié nos séances d'encadrement, sans pression ni stress, empreintes d'humour et de beaucoup de travail.

Nous remercions également notre formatrice, **Madame Tamazouzt Kahina**, qui a veillé à ce que nous ayons la motivation nécessaire, l'amour de la programmation, le courage d'affronter les bugs, cette couleur si splendide qui est le #9A1213 et surtout cette extraordinaire joie de vivre qui règne au cœur de son établissement.

Nous tenons à exprimer notre gratitude envers **Monsieur 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. Nous remercions également toute l'équipe des planificateurs pour leur accueil, en particulier **Monsieur Mehidi Mohand**, qui nous a énormément aidé à comprendre le rôle des planificateurs au sein de Bejaia Logistique et pour toute la récolte d'informations qui ont aidé à concevoir le projet final.

Enfin, nous remercions fermement **les membres du jury** pour nous avoir fait le grand honneur d'examiner et de juger ce travail.

Dans l'impossibilité de citer tous les noms, nos sincères remerciements vont à nos amis et celles et ceux qui, de près ou de loin, ont contribué à notre bien-être.

Table des matières

Table des matieres	i
Table des figures	iv
Liste des tableaux	vi
Liste des abréviations	vii
Introduction Générale	1
1 Organisme d'accueil et étude Préalable	3
1.1 Introduction	3
1.2 Présentation de l'organisme d'accueil	3
1.3 Localisation	4
1.4 Missions et activités de Bejaia Logistique	5
1.5 Structure de Bejaia Logistique	5
1.6 Historique et Définition de la logistique	6
1.6.1 Historique	6
1.6.2 Définition	6
1.7 la typologie de la logistique	6
1.7.1 logistique interne	7
1.7.2 Logistique Externe	7
1.7.3 Logistique de production	7
1.7.4 Logistique de distribution	7
1.8 La chaine logistique (Supply Chain)	8
1.9 TMS	10
1.10 Le problème de routage de véhicule (VRP)	11
1.11 Problématique	12
1.12 Objectif de l'étude	13
1.13 Conclusion	13
2 Contexte général du projet et Étude des besoins	14
2.1 Introduction	14

2.2	Démarche de développement	14
2.2.1	Principe de la méthode XP	14
2.2.2	Les Cinq valeurs fondamentales de l'approche XP	15
2.2.3	Les phases de la méthodologie XP	16
2.3	Présentation de la méthode UML	18
2.4	Identification des acteurs	19
2.5	Diagramme de contexte général	20
2.6	Étude et Spécification des besoins	21
2.6.1	Analyse des besoins fonctionnels	21
2.6.2	Analyse des besoins non-fonctionnels	23
2.7	Diagrammes de cas d'utilisation	24
2.7.1	Diagramme de cas d'utilisation global	24
2.7.2	Diagramme de cas d'utilisation de l'acteur « Administrateur »	26
2.7.3	Diagramme de cas d'utilisation de l'acteur « Commercial »	27
2.7.4	Diagramme de cas d'utilisation de l'acteur « Planificateur »	28
2.7.5	Diagramme de cas d'utilisation de l'acteur « Coordinateur logistique »	28
2.7.6	Diagramme de cas d'utilisation de l'acteur « Ressource humaine »	29
2.7.7	Diagramme de cas d'utilisation de l'acteur « GMAO »	29
2.8	Description des cas d'utilisations	30
2.8.1	Description du cas d'utilisation "Authentification"	30
2.8.2	Description du cas d'utilisation "Gestion des comptes utilisateurs et des groupes de rôles"	31
2.8.3	Description du cas d'utilisation "Gestion des paramètres de l'application"	32
2.8.4	Description du cas d'utilisation "Gestion des commandes"	33
2.8.5	Description du cas d'utilisation "Gestion des ClientsBL"	35
2.8.6	Description du cas d'utilisation "Planification des commandes"	36
2.8.7	Description du cas d'utilisation "Gestion des missions"	38
2.8.8	Description du cas d'utilisation "Gestion des trajets BL"	39
2.8.9	Description du cas d'utilisation "Gestion des chauffeurs"	40
2.8.10	Description du cas d'utilisation "Gestion des véhicules"	42
2.9	Diagrammes de séquence	43
2.9.1	Concepts de bases	43
2.9.2	Diagramme de séquence pour le cas d'utilisation « Authentification »	44
2.9.3	Diagramme de séquence du cas d'utilisation « Ajouter une commande »	45
2.9.4	Diagramme de séquence du cas d'utilisation « Modifier une commande »	46
2.9.5	Diagramme de séquence du cas d'utilisation « Supprimer une commande »	47
2.9.6	Diagramme de séquence du cas d'utilisation « Rechercher une commande »	48
2.9.7	Diagramme de séquence du cas d'utilisation « Planifier une commande »	49
2.9.8	Diagramme de séquence du cas d'utilisation « Gérer une mission »	50
2.10	Conclusion	51

3	Analyse et Conception détaillée	52
3.1	Introduction	52
3.2	Modélisation des diagrammes de séquences détaillés	52
3.2.1	Diagramme de séquence détaillé du cas d'utilisation «Authentification»	53
3.2.2	Diagramme de séquence détaillé du cas d'utilisation «Ajouter une commande»	54
3.2.3	Diagramme de séquence détaillé du cas d'utilisation« Modifier une commande »	55
3.2.4	Diagramme de séquence détaillé du cas d'utilisation «Supprimer une commande»	56
3.2.5	Diagramme de séquence détaillé du cas d'utilisation «Rechercher une commande»	57
3.2.6	Diagramme de séquence détaillé du cas d'utilisation «Planifier une commande»	58
3.2.7	Diagramme de séquence détaillé du cas d'utilisation «Gérer une mission»	59
3.3	Diagramme de classe	60
3.3.1	Le dictionnaire de données	62
3.4	Modèle relationnel de données	62
3.4.1	Règles de passage du diagramme de classe vers le modèle relationnel	62
3.4.2	Schéma Relationnel de Notre Base de Données	64
3.5	Conclusion	65
4	Réalisation de l'application	66
4.1	Introduction	66
4.2	Présentation de notre application « TransPlanify »	66
4.3	Environnement du développement de l'application	66
4.4	Imprimés écrans de notre application	74
4.5	Conclusion	80
	Conclusion Générale	81
	Reference	82

Table des figures

1.1	Logo de BL	4
1.2	Plan d'accès vers Bejaia Logistique	4
1.3	Organigramme Générale de Bejaia Logistique	5
1.4	Flux physiques de la supply chain	9
1.5	Flux d'informations de la supply chain	9
1.6	Flux Financier de la supply chain	10
1.7	Systèmes de gestion de transport	11
1.8	Le problème de tournées de véhicules	12
2.1	Les Cinq valeurs de l'approche XP	16
2.2	Les étapes de la méthode XP	18
2.3	Les diagrammes définis par UML	19
2.4	Diagramme de contexte général	21
2.5	Diagramme de cas d'utilisation Global	25
2.6	Diagramme de cas d'utilisation de l'acteur « Administrateur »	26
2.7	Diagramme de cas d'utilisation de l'acteur « Commercial»	27
2.8	Diagramme de cas d'utilisation de l'acteur « Planificateur »	28
2.9	Diagramme de cas d'utilisation de l'acteur « Coordinateur logistique »	28
2.10	Diagramme de cas d'utilisation de l'acteur « Ressource humaine »	29
2.11	Diagramme de cas d'utilisation de l'acteur « GMAO »	29
2.12	Diagramme de cas d'utilisation «Authentification»	30
2.13	Diagramme de cas d'utilisation «Gestion des comptes utilisateurs et des groupes de rôles»	31
2.14	Diagramme du cas d'utilisation «Gestion des paramètres»	32
2.15	Diagramme de cas d'utilisation «Gestion des Commandes»	33
2.16	Diagramme de cas d'utilisation «Gestion d'un clientBL»	35
2.17	Diagramme de cas d'utilisation «Planification des Commandes»	36
2.18	Diagramme de cas d'utilisation «Gestion d'une mission»	38
2.19	Diagramme du cas d'utilisation «Gestion des trajets BL»	39
2.20	Diagramme de cas d'utilisation «Gestion des chauffeurs »	41
2.21	Diagramme du cas d'utilisation «Gestion des camions »	42
2.22	Diagramme de séquence du cas d'utilisation « Authentification »	44

2.23	Diagramme de séquence du cas d'utilisation « Ajouter une commande » .	45
2.24	Diagramme de séquence du cas d'utilisation « Modifier une commande »	46
2.25	Diagramme de séquence du cas d'utilisation « Supprimer une commande »	47
2.26	Diagramme de séquence du cas d'utilisation « Rechercher une commande »	48
2.27	Diagramme de séquence du cas d'utilisation « Planifier une commande »	49
2.28	Diagramme de séquence du cas d'utilisation « Gérer une mission »	50
3.1	Diagramme de séquence détaillé du cas d'utilisation «Authentification» .	53
3.2	Diagramme de séquence détaillé du cas d'utilisation «Ajout d'une commande»	54
3.3	Diagramme de séquence détaillé du cas d'utilisation« Modifier une commande »	55
3.4	Diagramme de séquence détaillé du cas d'utilisation «Suppression d'une commande»	56
3.5	Diagramme de séquence détaillé du cas d'utilisation «Rechercher une commande»	57
3.6	Diagramme de séquence détaillé du cas d'utilisation «Planifier une commande»	58
3.7	Diagramme de séquence détaillé du cas d'utilisation «Gérer une mission»	59
3.8	Diagramme de classe	61
4.1	Interface de Visual Studio Code	68
4.2	Interface d'authentification	74
4.3	La page d'accueil	74
4.4	Interface tableau de bord de l'administrateur	75
4.5	Interface liste des utilisateurs	75
4.6	Interface formulaire d'ajout d'un utilisateur	76
4.7	Interface statistiques	76
4.8	Interface liste des commandes	77
4.9	Interface formulaire d'ajout d'une commande	77
4.10	Interface détails d'une commande	78
4.11	Interface planification d'une commande	78
4.12	Interface création d'une mission	79
4.13	Interface gestion d'une mission	79
4.14	Interface élaboration de la feuille de route d'une mission	80

Liste des tableaux

2.1	Spécification des besoins non-fonctionnels	24
2.2	Description textuelle du cas d'utilisation « Authentification »	31
2.3	Description textuelle du cas d'utilisation « Gestion des comptes utilisateurs et des groupes de rôles »	32
2.4	Description textuelle du cas d'utilisation « Gestion des paramètres généraux de l'application »	33
2.5	Description textuelle du cas d'utilisation « Gestion des Commandes logistique » . .	34
2.6	Description textuelle du cas d'utilisation « gestion des Clients de l'entreprise BL »	36
2.7	Description textuelle du cas d'utilisation «Planification des commandes»	38
2.8	Description textuelle du cas d'utilisation « Gestion d'une mission »	39
2.9	Description textuelle du cas d'utilisation « Gestion des Trajet BL »	40
2.10	Description textuelle de cas d'utilisation « gestion des Chauffeurs BL »	41
2.11	Description textuelle du cas d'utilisation « Gestion des camions »	43
3.1	Description des données liées aux commandes.	62

Liste des abréviations

TMS	T ransport M anagement S ystem
BL	B éjaia L ogistique
PLV	P ublicité L ieu V ente
SARL	S ociété à R esponsabilité L imitée
BTP	B âtiment T ravaux P ublics
PME	P etite M oyenne E ntreprise
ADR	A ccord D angereuses R outes (Accord européen relatif au transport international des marchandises Dangereuses par Route)
VRP	V ehicle R outing P roblem
CVRP	C apacitated V ehicle R outing P roblem
XP	eX tremme P rogramming
UML	U nified M odeling L anguage
RH	R essources H umaines
MVC	M odel V iew C ontroller
MVT	M odel V iew T emplate
PK	P rimary K ey
FK	F oreign K ey
CPU	C entral P rocessing U nit
GHZ	G iga H ertz
RAM	R andom A ccess M emory
GO	G iga O ctets
VSC	V isual S tudio C ode
IDE	I ntegrated D evelopment E nvironment
HTML	H yper T ext M arkup L anguage
CSS	C ascading S tyle S heets
XML	E Xtensible M arkup L anguage
XHTML	E Xtensible H yper T ext M arkup L anguage
JS	J ava S cript
ORM	O bject R elational M apping
URL	U niform R esource L ocator
CSRF	C ross S ite R quest F orgery
SQL	S tructured Q uery L anguage

HTTP	H ypertext T ransfer P rotocol
SGBDR	S ystème G estion B ase D onnées R elationnelles
PHP	P ersonal H ome P age

RÉSUMÉ

Ce mémoire présente une approche agile (XP) pour optimiser la planification logistique d'un TMS grâce à une application web d'aide à la décision. Le système a été modélisé en utilisant UML et implémenté avec Python (framework Django) et MySQL comme base de données.

L'approche agile favorise la flexibilité et la collaboration, tandis que l'utilisation de Python, Django et MySQL garantit une application web robuste et sécurisée. Ce projet vise à améliorer l'efficacité et la rentabilité de l'entreprise Bejaia Logistique dans la gestion de ses opérations logistiques.

Mots clés : XP, TMS, UML, Python, Django, MySQL

ABSTRACT

This thesis presents an agile approach (XP) to optimize the logistics planning of a TMS through a web-based decision support application. The system was modeled using UML and implemented using Python (Django framework) and MySQL as the database.

The agile approach promotes flexibility and collaboration, while the use of Python, Django, and MySQL ensures a robust and secure web application. This project aims to improve the efficiency and profitability of Bejaia Logistique company in managing its logistics operations.

Key words : XP, TMS, UML, Python, Django, MySQL

Introduction Générale

L'augmentation des transactions commerciales internationales est due, d'une part, à l'évolution technologique de tous les moyens du transport et de communication et d'autre part, en vue de satisfaire les divers besoins économiques de différents pays pour le bien-être des citoyens. Cette évolution du commerce et le développement de l'industrie ont contribué au développement des échanges mondiaux avec l'ouverture sur l'extérieur. Le développement international de l'entreprise nécessite de la part de ses managers une réflexion sur ces capacités et ses fonctions principales qui sont : la production, la commercialisation, la gestion financière, les ressources humaines et la logistique.

Cette dernière permet d'organiser les circuits de matières et des produits au moindre coût, autrement dit la logistique est l'art de livrer les bons produits au bon endroit et aux bons moments. Ainsi la logistique au sens large recouvre les activités de transport et qui dit échanges dit capacités à transporter les produits manufacturés sur les lieux de commercialisation.

La dynamique du commerce international est favorisée notamment par l'importance du rôle de la logistique à travers ses différentes activités et intervenants. L'entreprise doit donc effectuer plusieurs choix qui doivent la conduire à la bonne maîtrise de ses flux d'information et de marchandise.

Le transport des marchandises constitue un vecteur indispensable de la chaîne logistique. Il assure la liaison entre les différentes étapes du système logistiques de l'approvisionnement à la distribution (fournisseurs- usines, inter usines, usines- entrepôts et entrepôts –clients). Le transport est un élément majeur dans la qualité des services, puisqu'il est directement lié aux retards, erreurs, pertes, casses, vols, avaries et autres risques de non acheminement de la marchandise à bon port.

Les critères qui favorise le succès des entreprises du transport logistique est exploité à travers le système d'information de l'entreprise avec l'apparition des Nouvelles Technologies de l'Information et de la Communication (NTIC). Ces dernières permettent en effet un meilleur accès aux informations de tout ordre et surtout afin d'exploiter les ressources émises par ces critères qui déterminent l'objectif de l'entreprise, et facilite le partage de l'information, pour une plus grande performance opérationnelle.

Le système d'information est 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 notamment : (la réduction

des couts, la maximisation du profit, 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, nous avons remarqués que le service de planification n'exploite pas assez le TMS (Transport Management System) et ne bénéficie pas totalement de cet atout, car le module qui se charge de la prise de décision des planifications n'est pas automatisé.

L'objectif principale de ce projet de fin de cycle master est de concevoir et réaliser une application web muni d'un système intelligent qui pourra aider à la décision les planificateurs de l'entreprise BL. Ce système intervient à tous les stades du processus décisionnel : la planification d'activité (Commandes, Camions, Chauffeurs), le déclenchement de la circulation, le suivi et le pilotage des mouvements, le contrôle et l'évaluation des opérations dont l'objectif est de maximiser les gains et de réduire les coûts liés à l'entreprise.

Ce manuscrit est structuré comme suit :

- **ORGANISME D'ACCUEIL ET ÉTUDE PRÉALABLE** : Présentation de l'organisation qui accueillera et soutiendra la mise en œuvre du projet, Description succincte de l'étude préalable effectuée pour identifier le contexte du projet.
- **CONTEXTE GÉNÉRAL DU PROJET ET ÉTUDE DES BESOINS** : Ce chapitre est consacré à l'identification et à l'analyse approfondie des besoins spécifiques identifiés lors de l'étude préalable, liés au projet ou à l'initiative en question et une Présentation des facteurs qui jouent un rôle pertinent dans notre projet.
- **ANALYSE ET CONCEPTION DÉTAILLÉE** : ce chapitre se concentre sur la phase de planification approfondie et de conception précise du projet. Il vise à décomposer les besoins identifiés en éléments concrets, à définir les spécifications techniques et à élaborer un plan détaillé pour la mise en œuvre. Cette étape permet de traduire les besoins en solutions concrètes et de fournir une vision claire du projet avant sa réalisation.
- **RÉALISATION DE L'APPLICATION** : description de la mise en œuvre concrète du projet, y compris les étapes de développement, les choix technologiques et les fonctionnalités clés. Présentation synthétique de la réalisation à l'aide d'imprimés écrans.

Nous terminerons par une conclusion générale qui résume le travail effectué avec des perspectives pour des futurs travaux en relation avec ce projet.

Chapitre 1

Organisme d'accueil et étude Préalable

1.1 Introduction

Pour étudier les relations entre système d'information logistique et transport, il est essentiel de comprendre brièvement le concept de logistique. La logistique est une "technologie" qui gère les flux physiques de marchandises, mais également les flux d'information qui les pilotent. Le transport, en tant que partie intégrante de la logistique, joue un rôle clé dans le déplacement des biens et des personnes. Dans ce contexte, nous présenterons la SARL Bejaïa Logistique et nous définirons les types de logistique, tout en énonçant la problématique et l'objectif de notre travail.

1.2 Présentation de l'organisme d'accueil

Fondée en 2008, la SARL BEJAIA LOGISTIQUE (BL) est l'une des références Algériennes dans le domaine du transport routier[1]. Elle bénéficie d'une image de marque de qualité et d'une notoriété nationale, en disposant d'un important parc de transport. En outre, elle propose la location des engins et du matériel pour bâtiment et travaux publics, manutention location de véhicules avec ou sans chauffeur et dans le transport des produits pétroliers. Elle est inscrite au registre de commerce sous le numéro 07B0185663.

Elle est implantée dans la zone industrielle AHRİK, IGHZER AMOKRAN, elle-même située dans la commune d'Ouzellaguen. Ce territoire appartient à la wilaya de Bejaïa et se trouve au nord-est de l'Algérie.

Elle possède un capital de 95 400 000 DA et a réalisé en 2017 un chiffre d'affaire de l'ordre de 1 940 619 000 DA.

BL tente d'apporter des solutions logistiques dans les différents domaines, ce qui traduit la multiplication de ses clients internes et externes[1].

La figure 1.1 correspond au logo de BL



FIGURE 1.1 – Logo de BL

1.3 Localisation

Elle est implantée à la zone industrielle Ahrik Ighzer 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, grâce à leurs capacités, aux compétences de leurs collaborateurs et aux investissements technologiques[1].

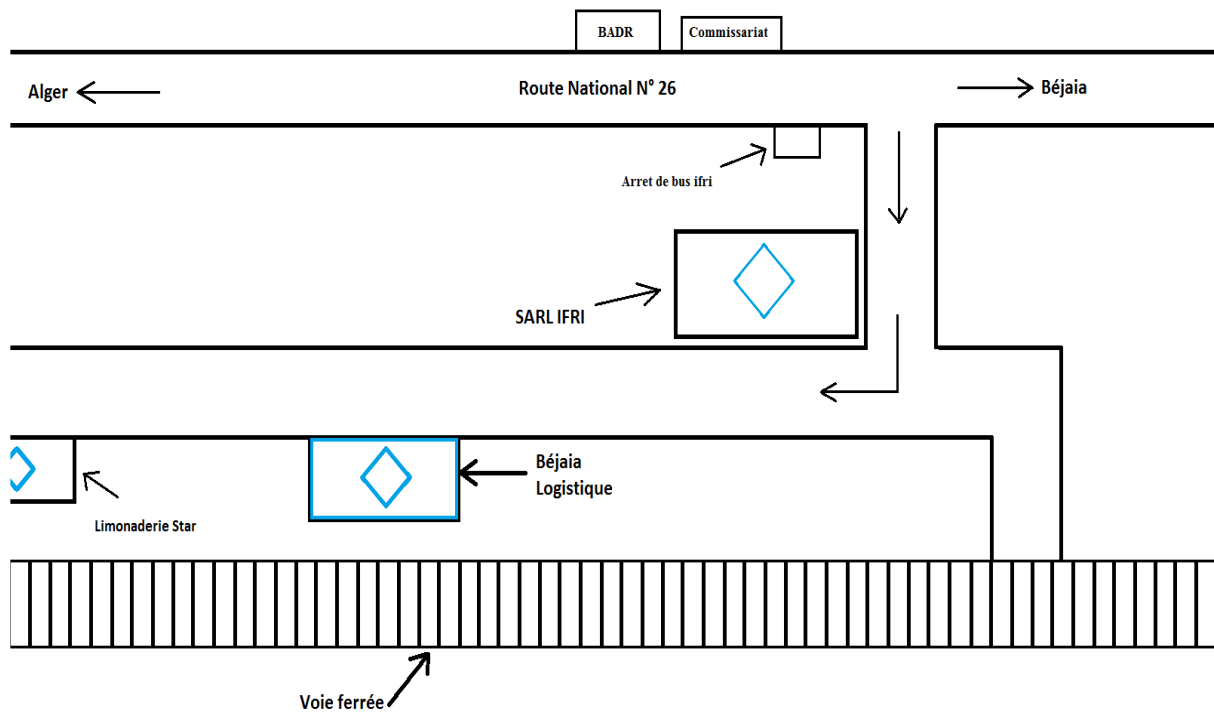


FIGURE 1.2 – Plan d'accès vers Bejaïa Logistique

1.4 Missions et activités de Bejaia Logistique

Les activités de BL sont étendues du transport public de marchandises, Location d'engins et matériels pour bâtiment, travaux publics et manutention, location de véhicules avec ou sans chauffeur et dans le transport des produits pétrolier.

BL a des missions très diverses liées aux activités qu'elle pratique tous les jours comme :

- 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.
- Assurer la sécurité des personnes et les biens en rapport avec les activités de transport.
- Assurer la satisfaction des clients.

1.5 Structure de Bejaia Logistique

La figure 1.3 ci-dessous représente l'organigramme générale de Bejaia Logistique (BL)

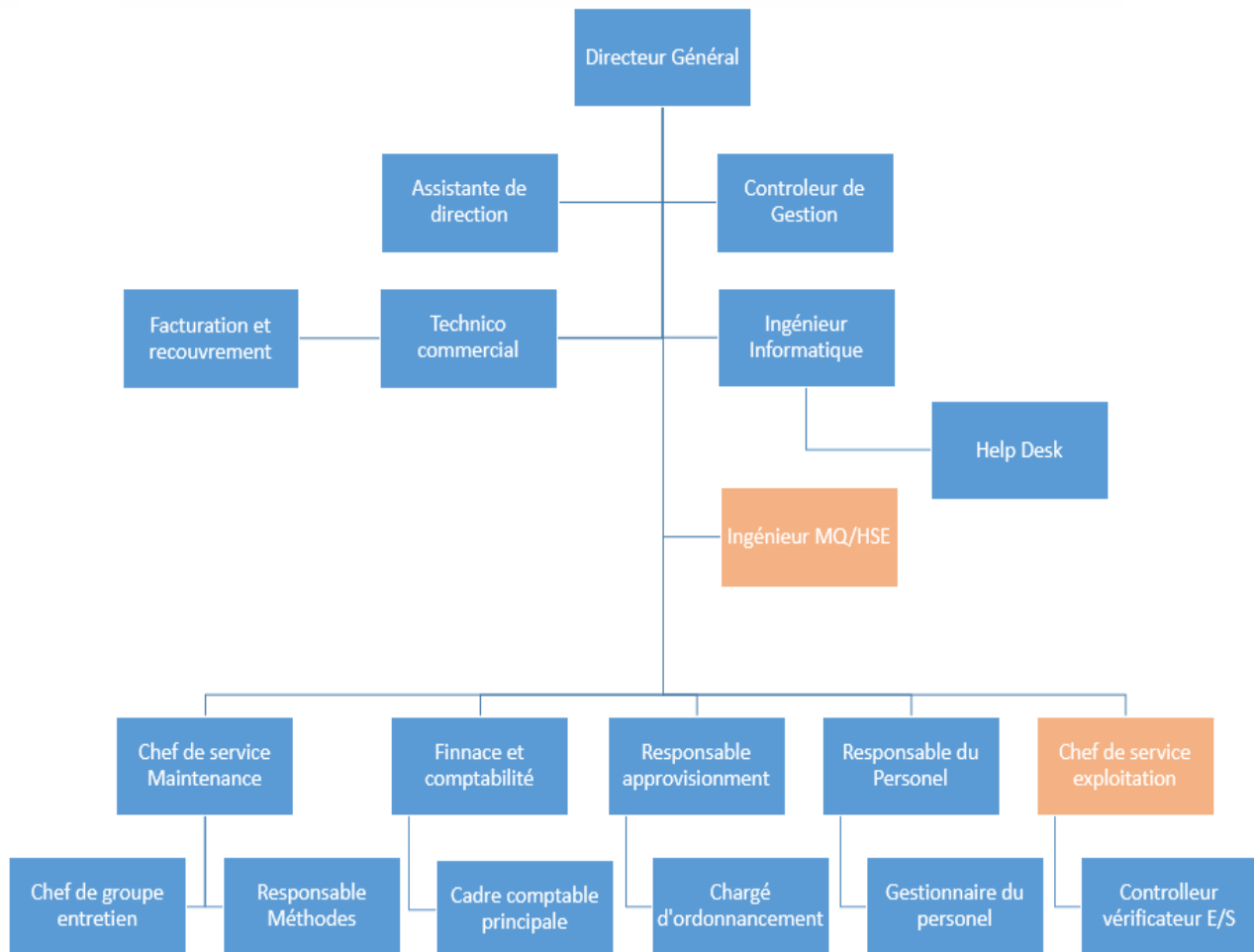


FIGURE 1.3 – Organigramme Générale de Bejaia Logistique

1.6 Historique et Définition de la logistique

1.6.1 Historique

Le dictionnaire de l'académie française donne « la science du calcul » comme premier sens au mot « logistique » étymologiquement, le terme « logistique » provient du grec « LOGISTIKOS³ », ce qui est « relatif à l'art du raisonnement ». Platon est notamment cité comme le premier à avoir utilisé le mot LOGISTIKOS pour opposer le calcul pratique (logistique) à l'arithmétique théorique. La logistique est encore largement empreinte de cette acception puisque de nombreux travaux académiques ainsi que de nombreuses préoccupations des entreprises se centrent sur l'optimisation (de tournées, de chargement de palette, des coûts. . .)

Le terme « logistique » trouve ensuite son origine dans le milieu militaire et provient du grade d'un officier en charge du « logis » des troupes, lors du combat. Napoléon met en place un encadrement de l'approvisionnement en vivres et en munition. Ainsi, le grade de « major général des logis » fut donné à « un officier qui avait la fonction de loger ou de camper les troupes, de diriger les colonnes, de les placer sur le terrain » « (Jomini, 1837) le logisticien militaire avait en charge le transport, le ravitaillement et le campement des troupes. Le terme « logistique » est entré dans le langage courant à partir du début des années quatre-vingt-dix, lorsque la première guerre de Golf a éclaté. A partir de cette période, démocratisé le mot en parlant de « soutien logistique » dans le cadre d'action militaire ou humanitaire.il est classiquement reconnu qu'une des principales organisations logistiques à caractère militaire du XXe siècle fut la coordination du débarquement des troupes alliées en Normandie en juin 1944. Le savoir-faire acquis s'est alors diffusé dans les entreprises, d'abord aux Etats-Unis, ensuite dans les pays européens. Le développement de la fonction logistique au sein des entreprises européennes est également lié à un contexte économique qui en a amené l'émergence[2].

1.6.2 Définition

La logistique est un ensemble de méthodes, fonctions et moyens mis en œuvre par une entreprise en vue de mettre à la disposition du client la marchandise attendue au moindre coût pour cette entreprise, en d'autres termes, cela 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'autres termes gérer tout ce qui concerne le transport et le stockage des produits de l'entreprise[3].

1.7 la typologie de la logistique

La logistique se décline en différentes typologies, chacune apportant des méthodes et des approches spécifiques pour répondre aux besoins de gestion des flux et des ressources.

1.7.1 logistique interne

Sous le nom de logistique intralogistique, fait référence au domaine qui couvre le mouvement des matériaux et les opérations de support qui se produisent au sein d'une entreprise. Il comprend plusieurs processus, tels que l'entreposage, le contrôle des stocks, les systèmes d'automatisation et de stockage, la manutention, l'équipement et les technologies de l'information.

Autrement dit, cela aide à réguler les flux de matériel au sein de l'institution, en veillant à ce que les bons articles soient au bon endroit, en bonne quantité et au bon moment[4] [5].

1.7.2 Logistique Externe

Les opérations qui se déroulent à l'extérieur de l'entrepôt (transport, livraison de la commande, etc...) et se composent de deux principales logistiques :

1. Logistique d'approvisionnement

Elle représente l'organisation et la gestion des flux qui sont généralement des premiers matériaux, de fournisseurs vers l'entreprise.

2. Logistique de distribution

La distribution consiste à écouler les produits finis vers les marchés de consommateurs. Autrement dit de l'entreprise jusqu'aux clients finaux[4].

1.7.3 Logistique de production

La logistique de production ou logistique industrielle couvre et optimise tous les processus logistiques qui ont lieu de l'achat des matières premières à la fabrication du produit[6].

Voici les deux systèmes de fabrication les plus courants :

- La fabrication pour le stock ou système push : les produits sont fabriqués à l'avance, puis stockés dans l'entrepôt. Le département commercial vend uniquement le stock disponible.
- La fabrication sur commande ou système pull : Le produit est fabriqué une fois que la commande du client est validée.

La fonction principale de la logistique de production est de réduire le lead time de production, c'est-à-dire le temps passé entre la génération d'une commande et l'obtention du produit fini[7].

1.7.4 Logistique de distribution

La logistique de distribution, également appelée LOGISTIQUE DE TRANSPORT vise à assurer l'acheminement rapide et efficace d'un produit ou un bien jusqu'au client. Il est inutile d'avoir un produit attrayant, fonctionnel et à un bon prix s'il ne parvient pas au destinataire dans les délais, la forme et les quantités convenues.

La logistique de distribution peut être :

- **Directe** : le fabricant se charge de distribuer directement le produit aux consommateurs finaux.

- **Indirecte** : le produit ne va pas au client final mais aux grossistes ou aux détaillants (qui se chargent de la vente au consommateur final).

La logistique de distribution au sein de l'organisme BL intervient juste après la production du produit, 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 (plate-formes de groupage et de dégroupage) [8].

Avant d'appréhender un quelconque système informatique dans ce domaine nous devons connaître le concept de chaîne logistique ou (SUPPLY CHAIN).

1.8 La chaîne logistique (Supply Chain)

Le SUPPLY CHAIN, ou chaîne d'approvisionnement en anglais est un ensemble d'activités qui a pour objectif d'acheminer un produit depuis sa fabrication jusqu'à sa livraison. Dans le secteur du transport routier, elle autorise à un ou plusieurs systèmes clients ou fournisseurs de prendre en compte et de répondre aux attentes des clients finaux.

Cette pratique permet d'optimiser certaines caractéristiques logistiques pour fournir un service efficace de transport de marchandises, parmi ces caractéristiques, on trouve :

- Délai de transport** : réduire le temps que prend le produit pour arriver dans les mains du clients.
- Fiabilité** : développer une relation de confiance avec le client.
- Disponibilité** : la capacité à livrer les produits dans les délais et les conditions prévus.
- Qualité** : les produits doivent arrivés en bon état lorsque le client les réceptionne.
- Conformité** : le respect des conditions dans lesquels les produits doivent être livrés (respect des normes ADR pour les marchandises dangereuses) [9].

la supply chain est divisée en 3 flux :

A) Le flux physiques de la Supply Chain

Il peut être associé à la logistique, c'est-à-dire le mouvement et le stockage des marchandises. Nous partons d'un site de production, puis nous nous déplaçons probablement vers un ou plusieurs entrepôts, puis vers un magasin ou un client final. Ce flux physique est vraiment basé sur le transport et le stockage du flux de marchandises.

La figure 1.4 ci-dessous représente un schéma explicatif du flux physiques.



FIGURE 1.4 – Flux physiques de la supply chain

B) Le flux d'informations de la Supply Chain

C'est le cerveau qui fera fonctionner le flux physique. C'est comme une gigantesque base de données qui pourrait être résumée par les questions suivantes : (quoi ? où ? comment ? combien ? et quand ?)

On y retrouve par exemple toutes les caractéristiques enregistrées d'un produit, tous les historiques de vente, tous vos indicateurs de performances mais aussi toutes les informations sur vos fournisseurs qui peuvent affecter votre stratégie d'approvisionnement.

La figure 1.5 ci-dessous représente un schéma explicatif du flux d'informations.



FIGURE 1.5 – Flux d'informations de la supply chain

C) Le flux financiers de la Supply Chain

Se définit comme 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.

La figure 1.6 ci-dessous représente un schéma explicatif du flux financiers.



FIGURE 1.6 – Flux Financier de la supply chain

[10]

1.9 TMS

Les TMS (Transport management system) ou systèmes de gestion de transport sont des logiciels d'aide à la décision qui permettent au décisionnaire, de gérer le transport. Dans le détail, les TMS supportent de nombreux aspects liés à la gestion du transport : activité d'expéditions, document de travail, gestion d'une flotte de camions et de chauffeurs, l'organisation des plannings de chargement, étiquetage des produits à transporter, facturation, etc [11].

Les fonctionnalités couvertes par un TMS sont très variables et dépendent de la compagnie qui l'acquiert mais généralement trois fonctions principales :

- **L'analyse / l'aide à la décision.**
- **Le suivi opérationnel des tournées de transport.**
- **Le reporting.**

La figure 1.7 suivante représente les nombreuses fonctionnalités admises par un TMS (Transport management system)

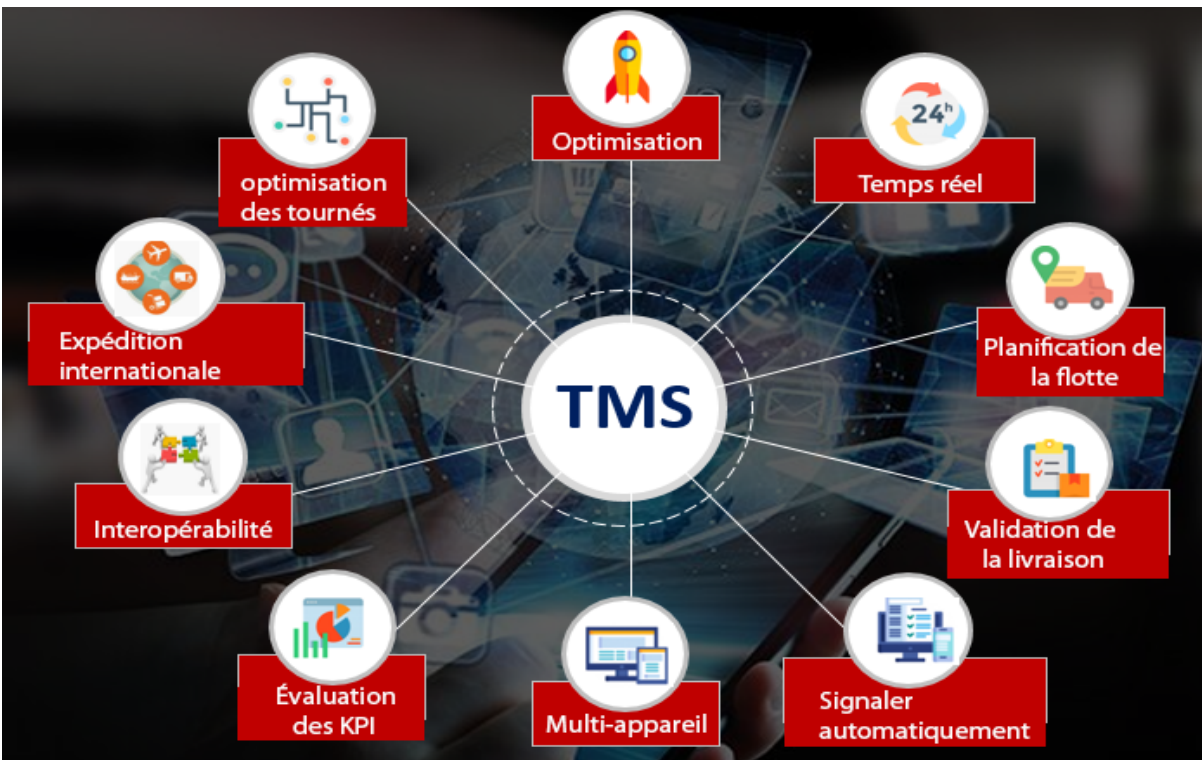


FIGURE 1.7 – Systèmes de gestion de transport

1.10 Le problème de routage de véhicule (VRP)

Une définition générale du VRP (Vehicle Routing Problem) consiste à chercher un itinéraire optimal pour une flotte de véhicules, basée en un ou plusieurs dépôts, afin de desservir un ensemble de clients ayant des commandes connues, et dispersés géographiquement. Une tournée désigne l'ensemble des clients visités par un véhicule qui part et revient au même dépôt.

Le VRP fait partie des problèmes d'optimisation combinatoire et de recherche opérationnelle. La première fois que ce problème a été étudié était en 1959 sous le nom de « The Truck Dispatching Problem » et a depuis fait l'objet de nombreux travaux qui ont donné de nombreuses variantes et différentes méthodes de résolution.

Il modélise un problème de transport avec une contrainte de capacité CVRP (Capacitated Vehicle Routing Problem) qui consiste à livrer des marchandises auprès des clients à l'aide d'une flotte de véhicules à capacité limitée. La résolution consiste à déterminer un ensemble de tournées qui minimise au mieux des objectifs comme le coût total, la distance totale parcourue, la somme des retards des clients pour un nombre minimal de véhicules partant d'un dépôt et y retournant[12].

La figure 1.8 ci-dessous montre une représentation du problème de tournées des véhicules.

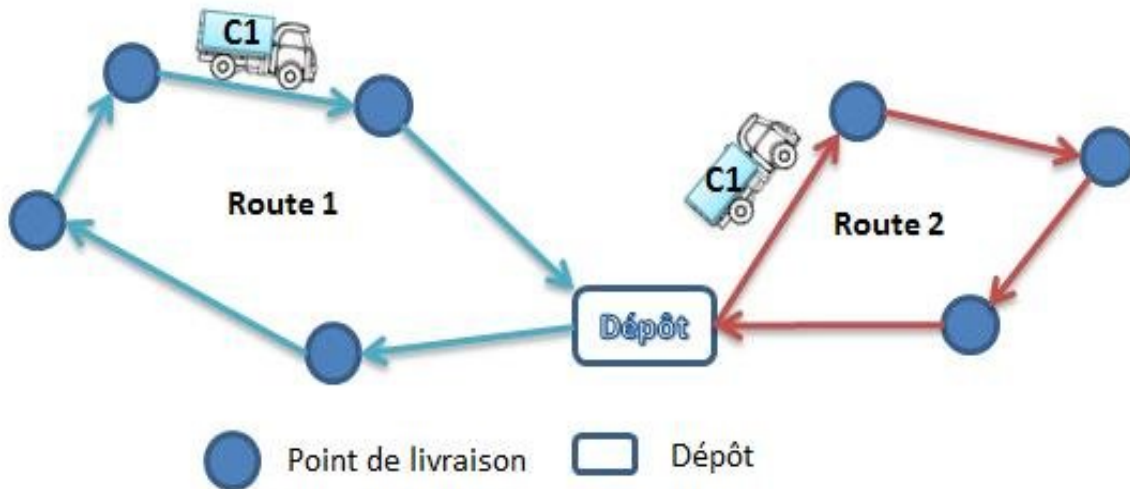


FIGURE 1.8 – Le problème de tournées de véhicules

[13]

1.11 Problématique

L'informatique décisionnelle est un processus nécessaire dans la planification logistique, cela permet une prise de décision imminente basée sur les données collectées par l'entreprise. 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.

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.
- Risque d'erreur lors de la collecte des données.
- Charge de travail énorme pour les planificateurs.
- Difficulté de récupération de l'historique des planifications sur excel.
- Difficulté d'échange d'information entre les différents planificateurs, ce qui provoque une perte de temps.

1.12 Objectif de l'étude

L'objectif de notre étude est de créer une application web d'aide à la décision, doté d'un système d'optimisation au profit du module de la planification, cela facilitera pour les planificateurs la gestion des commandes clients et pour les coordinateurs logistiques le suivi des différentes ressources dont dispose l'entreprise (chauffeurs, tracteur et remorques... etc) et qui permettra à l'entreprise de réduire ses coûts et d'augmenter ses profits.

L'application devra donc répondre aux besoins suivants :

- Avoir un suivi des ressources logistiques.
- la suggestion des ressources adéquates pour chaque commande.
- Propositions de solutions du meilleur itinéraire selon les critères d'une commande logistique.
- Planifications des commandes clients.
- Définir la rentabilité de la commandes(calcul de la fonction objective)
- Élaboration d'une mission à partir des données d'une commande.
- Avoir une meilleure optimisation de la feuille de route de sorte à ordonner les missions pour fournir un meilleur gain pour la société.

1.13 Conclusion

Dans ce chapitre, nous avons présenté l'importance des relations entre système d'information logistique et transport. Nous avons mis en avant les tendances actuelles en matière de logistique et montré comment les flux d'information permettent de piloter les flux physiques. Nous avons également souligné l'étroite interdépendance entre la logistique et le transport, en rappelant que le transport est une partie intégrante de la logistique.

Nous avons ensuite présenté la SARL Bejaïa Logistique et défini le sens de la logistique ainsi que ses différents types. Nous avons posé la problématique de notre travail et expliqué les raisons qui nous ont amenés à mener cette étude.

En somme, ce chapitre nous a permis de mieux comprendre l'importance de la logistique et du transport dans le mouvement des biens et des personnes. Nous avons également montré l'importance des systèmes d'information dans la gestion de la logistique et du transport, en particulier dans le pilotage des flux physiques. Ce chapitre nous a ainsi fourni une base solide pour la suite de notre étude.

Chapitre 2

Contexte général du projet et Étude des besoins

2.1 Introduction

L'étude de l'existant est une étape indispensable dans tout projet informatique. Une bonne conception doit se baser sur une connaissance approfondie du système actuel. Elle correspond à l'analyse des documents, des fichiers, l'enchaînement des procédures et les anomalies du système actuel. Dans ce chapitre nous allons présenter la démarche de développement à suivre pour la réalisation de notre projet, ensuite nous allons procéder à une identification des besoins fonctionnels et non-fonctionnels, les différents acteurs qui vont interagir avec le système, élaborer le diagramme de contexte ainsi que les diagrammes de cas d'utilisation pour spécifier les besoins de notre système. Nous concluons notre étude par l'élaboration des diagrammes de séquences système spécifique à chaque cas d'utilisation.

2.2 Démarche de développement

Dans la réalisation de notre application, nous avons choisi l'approche eXtreme Programming (XP) qui est l'une des méthodes agile. En effet, la méthode XP s'appuie sur un ensemble de pratiques couvrant une grande partie des activités de la réalisation d'un logiciel :

- Planification.
- Organisation de l'équipe de développement.
- Échanges avec le client.
- Programmation.

2.2.1 Principe de la méthode XP

Les principes de la méthode eXtreme Programming ne sont pas nouveaux puisqu'il s'agit de ceux des méthodes Agiles. La différence et l'originalité résident dans le fait qu'ils sont poussés à

l'extrême.

La méthode eXtreme Programming s'appuie sur :

- Une forte réactivité au changement des besoins du client.
- Un travail d'équipe.
- La qualité du travail fourni.
- La qualité des tests effectués au plus tôt[14].

2.2.2 Les Cinq valeurs fondamentales de l'approche XP

XP met en évidence cinq valeurs essentielles au travers de la méthode :

- **Communication** : la communication est considérée comme l'élément clé pour assurer une compréhension commune entre les membres de l'équipe, les clients et les parties prenantes. Cela se fait à travers des interactions fréquentes, des réunions régulières et l'utilisation d'outils de communication efficaces. Une communication ouverte et transparente favorise la collaboration et permet de résoudre rapidement les problèmes ou les malentendus.
- **Simplicité** : La valeur de la simplicité dans XP vise à maintenir les solutions logicielles simples et élégantes. Plutôt que de concevoir des architectures complexes, XP encourage à trouver la solution la plus simple qui répond aux besoins actuels. Cela facilite la maintenance, les tests et les modifications futures. La simplicité permet également une compréhension plus facile du code par tous les membres de l'équipe.
- **Feedback** : le feedback est essentiel pour s'assurer que le projet avance dans la bonne direction et pour permettre des améliorations continues. Le feedback peut provenir des clients, des utilisateurs finaux, des membres de l'équipe ou d'autres parties prenantes. Des cycles de développement courts, des démonstrations régulières du logiciel en cours de développement et des tests fréquents aident à obtenir un feedback rapide et précieux. Ce retour d'information permet d'ajuster et d'optimiser le produit en fonction des besoins réels.
- **Respect** : Le respect est une valeur clé dans XP qui favorise une culture de travail positive et collaborative. Cela implique le respect des idées, des opinions et des contributions de tous les membres de l'équipe. Chaque membre est considéré comme un expert dans son domaine et est encouragé à s'exprimer librement. Le respect mutuel et la confiance favorisent la créativité, l'innovation et l'engagement de chacun dans la réussite collective.
- **Courage** : La valeur du courage dans XP se réfère à la volonté de prendre des décisions difficiles et de faire face aux défis. Cela peut impliquer la remise en question de pratiques établies, la prise de risques calculés et l'adaptation aux changements. Les membres de l'équipe XP sont encouragés à sortir de leur zone de confort et à proposer des améliorations, même si elles semblent initialement difficiles ou risquées. Le courage est essentiel pour maintenir l'agilité et pour résoudre les problèmes de manière proactive [15].

La figure 2.1 recense les valeurs essentielles de l'approche XP.

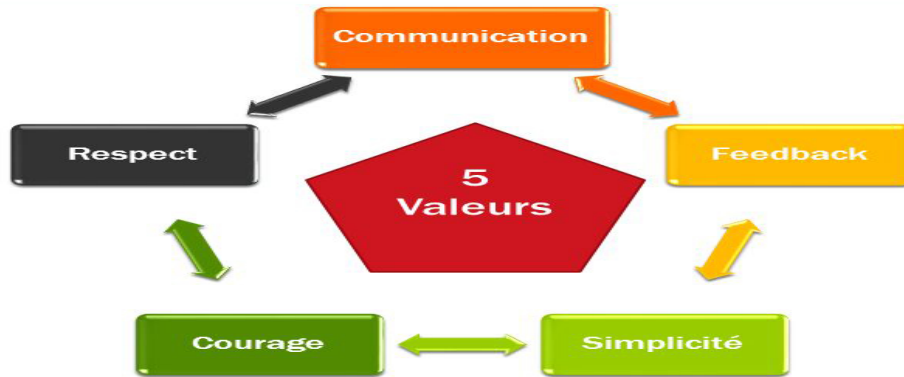


FIGURE 2.1 – Les Cinq valeurs de l'approche XP

2.2.3 Les phases de la méthodologie XP

La méthodologie eXtreme Programming (XP) suit un processus itératif et incrémental, qui comprend plusieurs phases. Voici les principales phases de la méthodologie XP :

- **Exploration**

Au cours de cette phase l'équipe de travail sera focalisée sur des questions d'ordre techniques destinées à la manipulation de l'architecture du système notamment :

- Les différents outils et technologies à utiliser, se pencher sur quel Framework.
- Explorer les différentes possibilités d'architecture pour le système.
- La compréhension approfondie des besoins du client et des objectifs du projet.
- Les itérations dans cette phase permettent de clarifier les exigences en travaillant directement avec le client et les parties prenantes.
- Les itérations aident à affiner les cas d'utilisation, à déterminer les fonctionnalités prioritaires et à établir une base solide pour le reste du projet.

- **Planning**

Au cours d'une réunion dédiée à cette phase, l'équipe de travail sortira avec des directives qui suit :

- Planifier les itérations futures, en définissant les priorités et en estimant la durée des tâches.
- Les itérations dans cette phase permettent d'ajuster et de réviser continuellement le plan du projet en fonction des nouveaux éléments d'information.
- Les itérations de planification facilitent la prise de décision éclairée sur les fonctionnalités à inclure dans chaque itération et aident à éviter le surchargement ou les retards.

- Écritures des tests unitaires.

- **Production de la 1ère release**

Dans cette phase, l'équipe XP se concentre sur la création de la première version fonctionnelle du logiciel.

- Cette phase consiste à développer et à livrer la première version fonctionnelle du logiciel.

- Les itérations dans cette phase permettent de se concentrer sur des fonctionnalités spécifiques et de les développer, les tester et les intégrer dans le produit.

- Les itérations offrent une approche itérative pour s'assurer que chaque fonctionnalité est développée correctement et est prête à être livrée aux utilisateurs.

- **Mise en production**

Une fois que la première version fonctionnelle est prête, la phase de mise en production commence par :

- Impliquer la mise en production du logiciel, rendant les fonctionnalités disponibles aux utilisateurs finaux.

- Les itérations dans cette phase permettent de s'assurer que le processus de déploiement et de mise en production se déroule sans problème.

- Chaque itération produit un sous ensemble de fonctionnalités principales.

- Les itérations aident à valider que les fonctionnalités déployées fonctionnent correctement dans l'environnement de production par l'établissement de tests fonctionnels, cela va aider à prendre des mesures correctives si nécessaire.

- **Maintenance**

La phase de maintenance est dédiée à la gestion continue du logiciel.

- Cette phase implique la maintenance continue du logiciel pour assurer son bon fonctionnement et sa stabilité.

- Les itérations dans cette phase permettent de gérer les correctifs de bogues, d'ajouter de nouvelles fonctionnalités ou d'améliorer les fonctionnalités existantes.

- Les itérations itératives de maintenance garantissent que le logiciel reste réactif aux besoins changeants des utilisateurs et continue de fournir de la valeur.

- **Mort**

La phase finale marque la fin du projet, lorsque le logiciel n'est plus maintenu ou utilisé.

- Quand le client ne parvient plus à spécifier de nouveaux besoins.

- Les itérations dans cette phase peuvent impliquer des tâches de clôture, comme la documentation finale, l'archivage des ressources et la préparation de la transition vers d'autres projets.

La figure 2.2 ci-dessous montre le cycle de vie de la méthodologie XP.

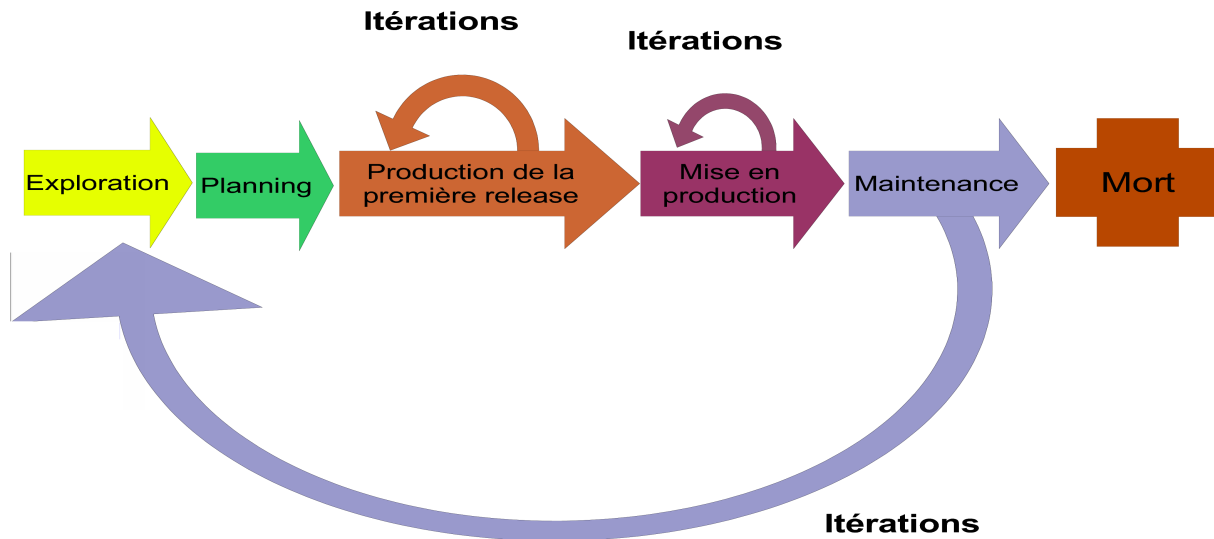


FIGURE 2.2 – Les étapes de la méthode XP

[16]

2.3 Présentation de la méthode UML

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

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[17].

La figure 2.3 ci-dessous représente les différents diagrammes définis par UML.

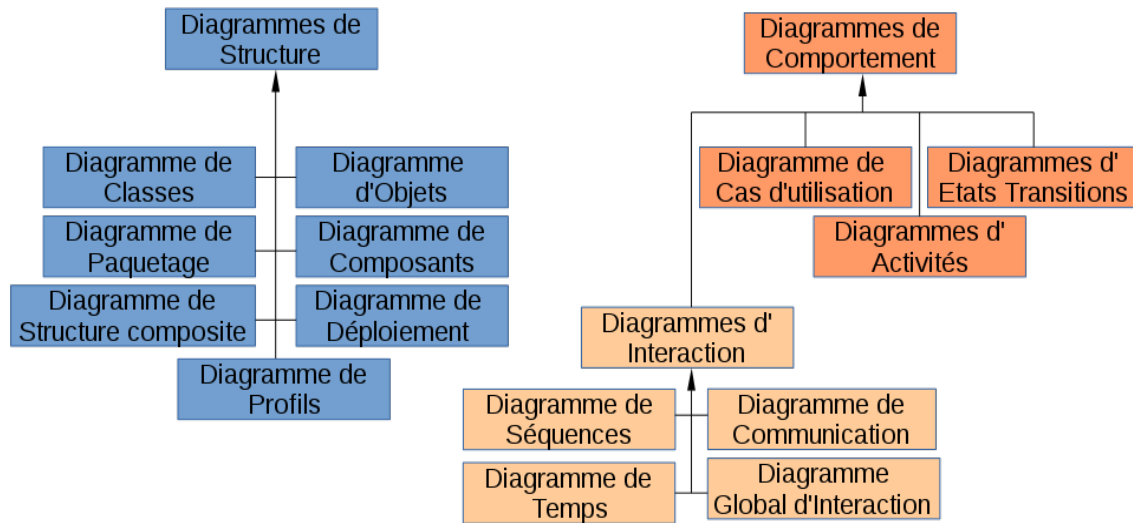


FIGURE 2.3 – Les diagrammes définis par UML

[18]

2.4 Identification des acteurs

Notre système communique avec des acteurs afin de réaliser un ensemble d'opérations. Un acteur est la personne désignée pour interagir avec le système. Il peut être (une personne, un dispositif matériel ou un autre système), dans notre cas et selon l'étude effectuée nous avons identifié les acteurs suivant :

- **L'administrateur (ADMIN)**
 - Il représente le superviseur qui contrôle l'accès aux comptes utilisateurs.
- **Gestionnaire commercial**
 - Réception des commandes clients.
 - Établissement et gestion des commandes.
 - Gestion des clients de l'entreprise BL.
- **planificateur**
 - Planification des commandes logistiques.

- Gestion des missions.
- Élaboration des feuilles de routes.
- Demande d'affrètement en cas de besoin.
- **Coordinateur logistique**
 - La gestion des trajets.
 - Le suivi et localisation des camions en temps réel.
- **Ressource humaine**
 - Gestion des chauffeurs.
 - Déclaration des indisponibilités chauffeurs.
 - Gestion des traitements de pointage.
- **Chargé de la maintenance**
 - Gestion des camions.
 - Déclaration des indisponibilités camions.

2.5 Diagramme de contexte général

Le diagramme de contexte général permet de visualiser les frontières du système, de définir les interactions clés avec son environnement et de clarifier le périmètre du système à l'étude.

Le diagramme choisi pour représenter le contexte général de notre projet, illustré dans la figure 2.4, est le suivant :

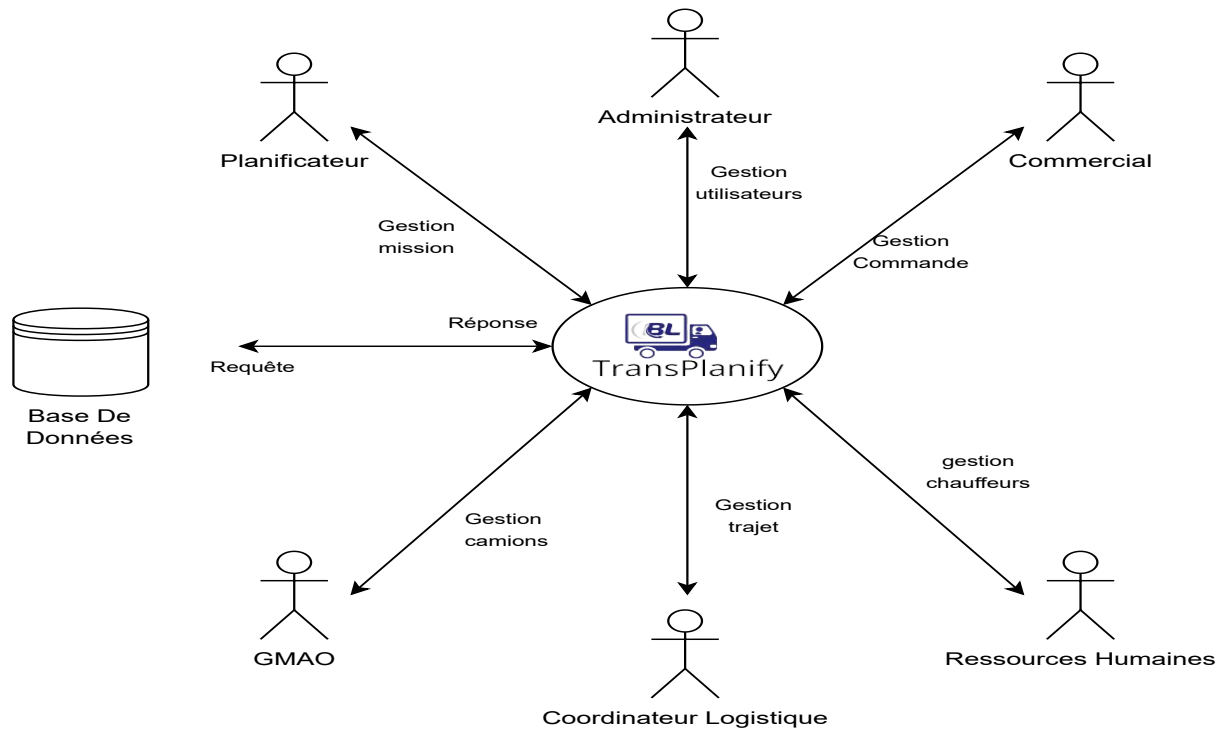


FIGURE 2.4 – Diagramme de contexte général

2.6 Étude et Spécification des besoins

Dans cette section, nous nous intéressons aux besoins des utilisateurs traités dans notre projet. C'est à dire la planification des missions et l'affectation des ressources adéquates.

2.6.1 Analyse des besoins fonctionnels

Dans la suite, nous désignons par centres d'intérêts les différents services offerts par notre application.

Après l'authentification des utilisateurs par un login et un mot de passe, ils seront redirigés vers leurs applications utilisateurs correspondantes.

Dans ce qui suit nous présentons pour chaque utilisateurs ses fonctionnalités.

1. Administrateur

- (a) Gestion des utilisateurs de l'application.
 - Ajout d'un utilisateur.
 - Modification d'un utilisateur.
 - Dés-activation d'un compte utilisateur.
 - Affectation d'un utilisateur à un groupe de rôles spécifique.
- (b) Consultation de la liste des utilisateurs.

- (c) Gestion des rôles utilisateurs.
 - Ajout d'un groupe de rôles.
 - Mise à jour d'un groupe de rôles.
 - Suppression d'un groupe de rôles.
- (d) Gestion des paramètres généraux de l'application.
 - Modification des valeurs d'un paramètre.

2. Commercial

- (a) Gestion des commandes clients.
 - Création d'une commande.
 - Modification d'une commande.
 - Suppression d'une commande.
 - Visualisation des détails d'une commande.
- (b) Gestion des clients.
 - Ajout d'un client.
 - Modification d'un client.
 - Suppression d'un client.

3. Planificateur

- (a) Consultation des commandes planifiées et non-planifiées.
- (b) Planification des commandes.
- (c) Gestion des missions logistiques.
- (d) Élaboration de la feuille de route.
- (e) Avoir une vue sur la disponibilité des ressources adéquates pour chaque commande.
- (f) Possibilité d'avoir une Automatisation du processus de planifications des commande pour une mission donnée.

4. Coordinateur logistique

- (a) Consultation des commandes planifiées.
- (b) Gestion des trajets.
 - Ajout d'un trajet.
 - Mise à jour d'un trajet.
 - Suppression d'un trajet.

5. Ressources Humaines

- (a) Gestion des chauffeurs
 - Ajout d'un chauffeur.

- Mise à jour d'un chauffeur.
- Suppression d'un chauffeur.

(b) Déclaration des indisponibilités chauffeurs.

6. GMAO

(a) Gestion des camions.

- Ajout d'un camion.
- Mise à jour d'un camion.
- Suppression d'un camion.

(b) Déclaration des indisponibilités camions.

2.6.2 Analyse des besoins non-fonctionnels

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

Objectifs Niveau1	Objectifs –Niveau2	Cas d'utilisation
Gestion de la sécurité	Authentification	<ul style="list-style-type: none"> — Vérification des champs. — Notions de sessions et de cookies. — Possibilité de changer de mot de passe, mot de passe oublié.
	Confidentialité	<ul style="list-style-type: none"> — Gestion des rôles utilisateurs.
Gestion des données	Intégrité	<ul style="list-style-type: none"> — Mise à jour simultanées.
	Interopérabilité	<ul style="list-style-type: none"> — Déploiement en réseau.
	Compatibilité	<ul style="list-style-type: none"> — L'éditeur et le langage de programmation ne peut être fiable qu'avec une compatibilité avec tout les navigateurs web.

	Historique	— Affichage des différentes opérations effectuées par chaque exploitant.
Gestion d'aide	Gestion d'erreurs	— Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre site web.
Fiabilité	Satisfaction du client	— L'application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.
Contraintes Ergonomiques	Bonne interface	— L'application doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages, couleurs et mise en textes utilisés.

TABLE 2.1 – Spécification des besoins non-fonctionnels

2.7 Diagrammes de cas d'utilisation

Dans ce qui suit nous définirons qu'est-ce qu'un diagramme de cas d'utilisation et nous allons exposer les diagrammes relatifs à notre étude.

2.7.1 Diagramme de cas d'utilisation global

Un diagramme de cas d'utilisation est un graphe d'acteurs, qui illustre les limites du système, ainsi que les relations de communication entre les acteurs et les cas d'utilisation, et des généralisations de ces cas d'utilisation[19].

La figure2.5 ci-dessous désigne le diagramme de cas d'utilisation global dans lequel nous allons mettre en évidence les fonctionnalités qu'offre notre application.

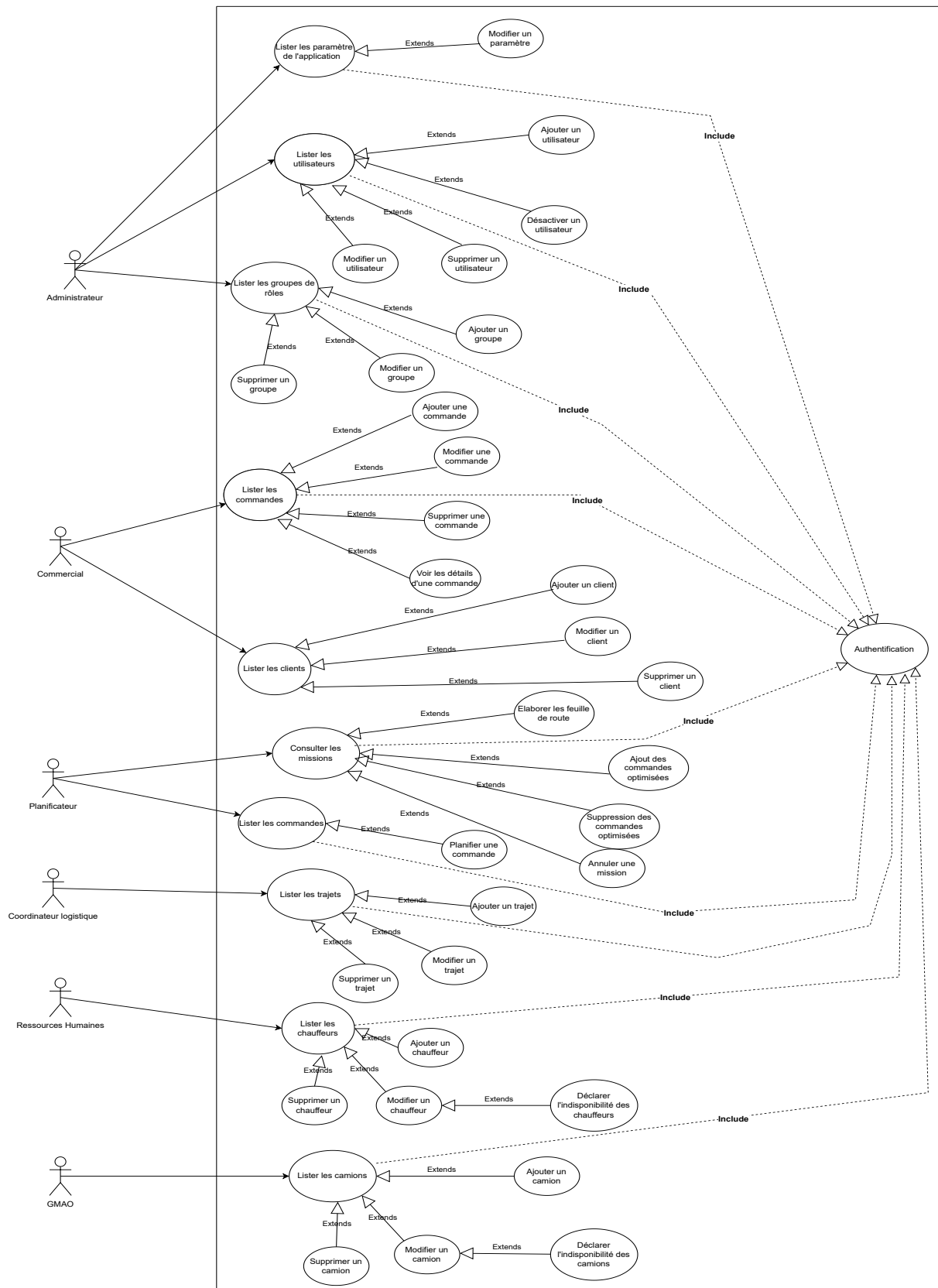


FIGURE 2.5 – Diagramme de cas d'utilisation Global

2.7.2 Diagramme de cas d'utilisation de l'acteur « Administrateur »

La figure 2.6 recense les fonctionnalités de l'administrateur, décrites en diagramme de cas d'utilisation.

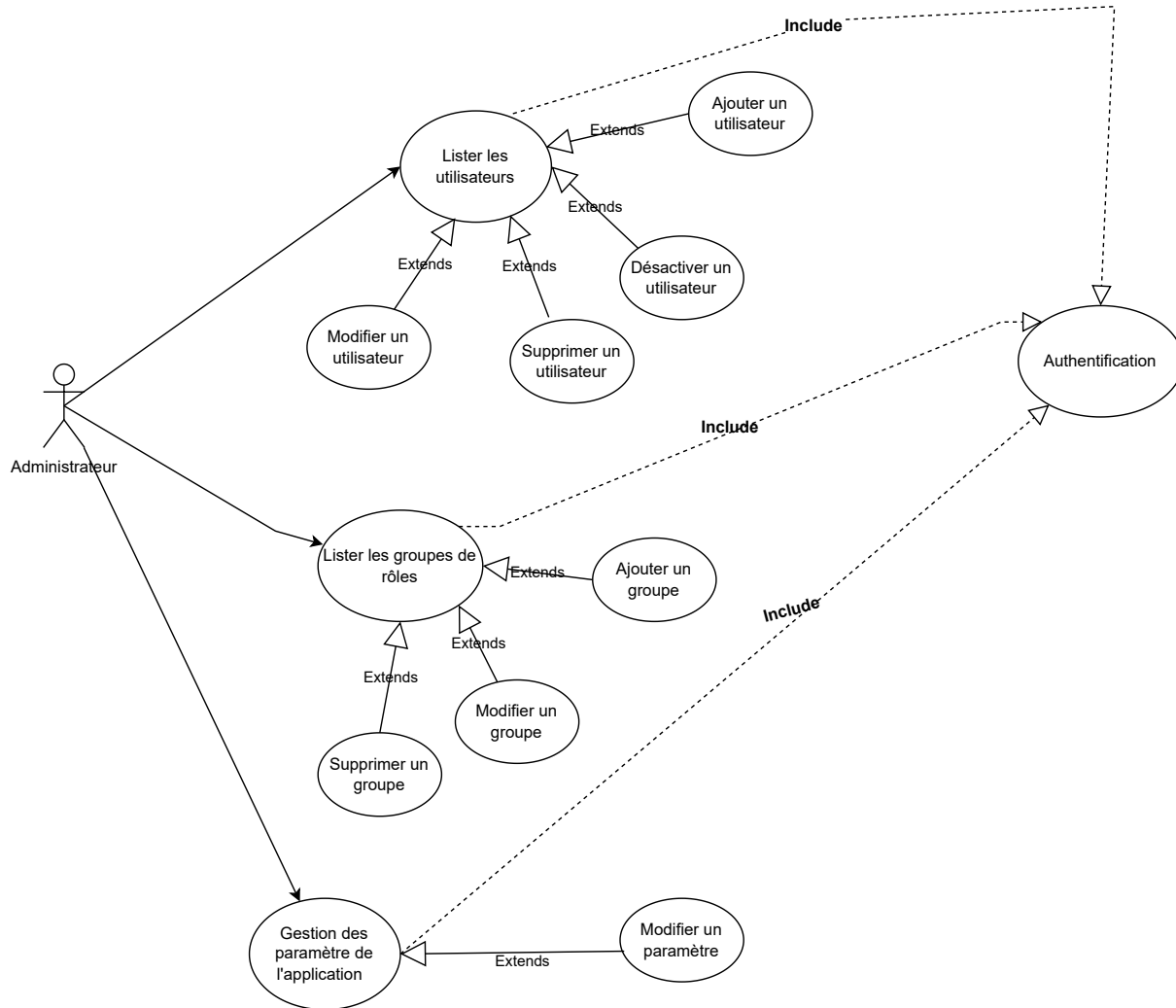


FIGURE 2.6 – Diagramme de cas d'utilisation de l'acteur « Administrateur »

2.7.3 Diagramme de cas d'utilisation de l'acteur « Commercial »

La figure 2.7 recense les fonctionnalités du gestionnaire commercial, décrites en diagramme de cas d'utilisation.

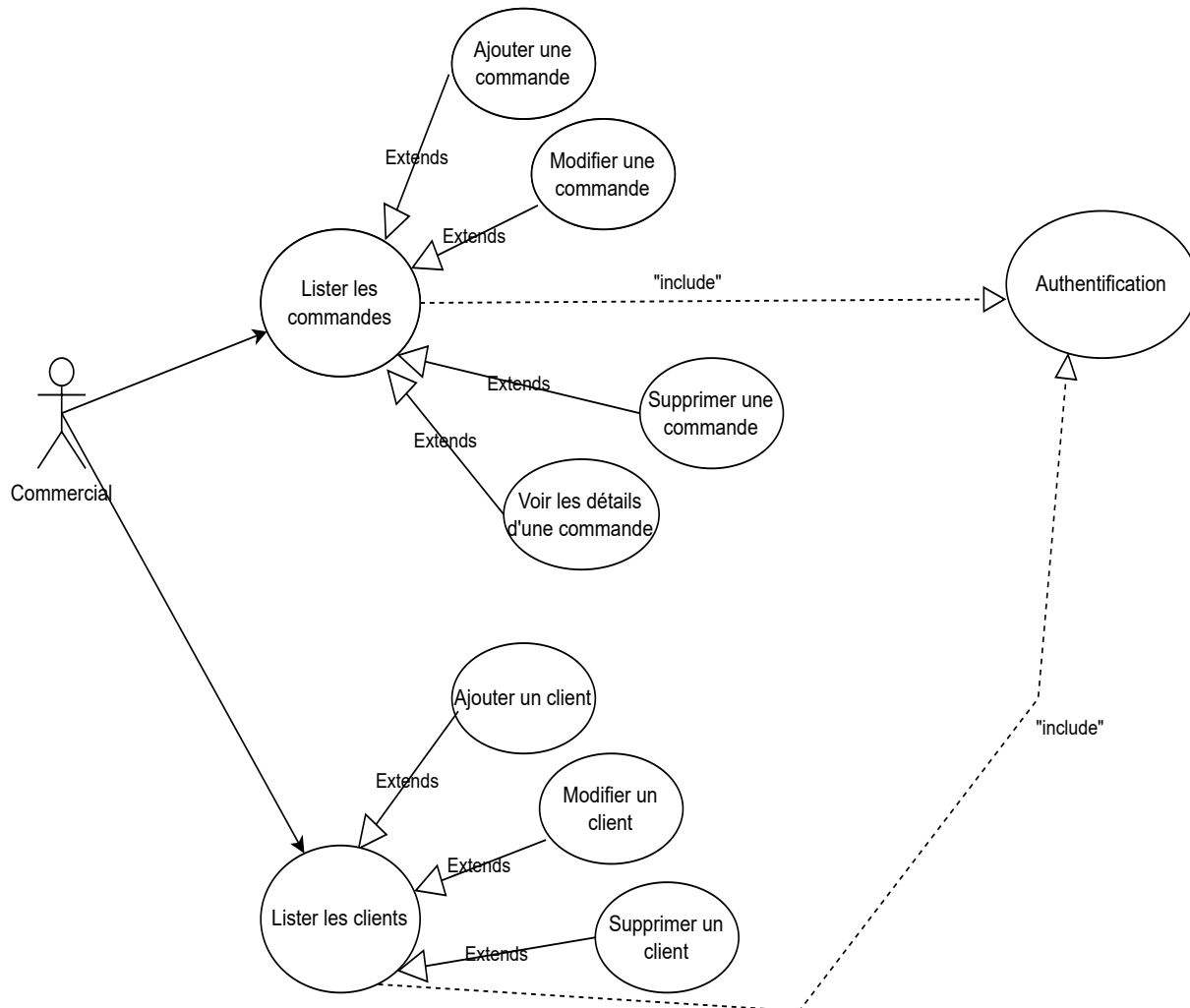


FIGURE 2.7 – Diagramme de cas d'utilisation de l'acteur « Commercial »

2.7.4 Diagramme de cas d'utilisation de l'acteur « Planificateur »

La figure 2.8 recense les fonctionnalités du planificateur, décrites en diagramme de cas d'utilisation.

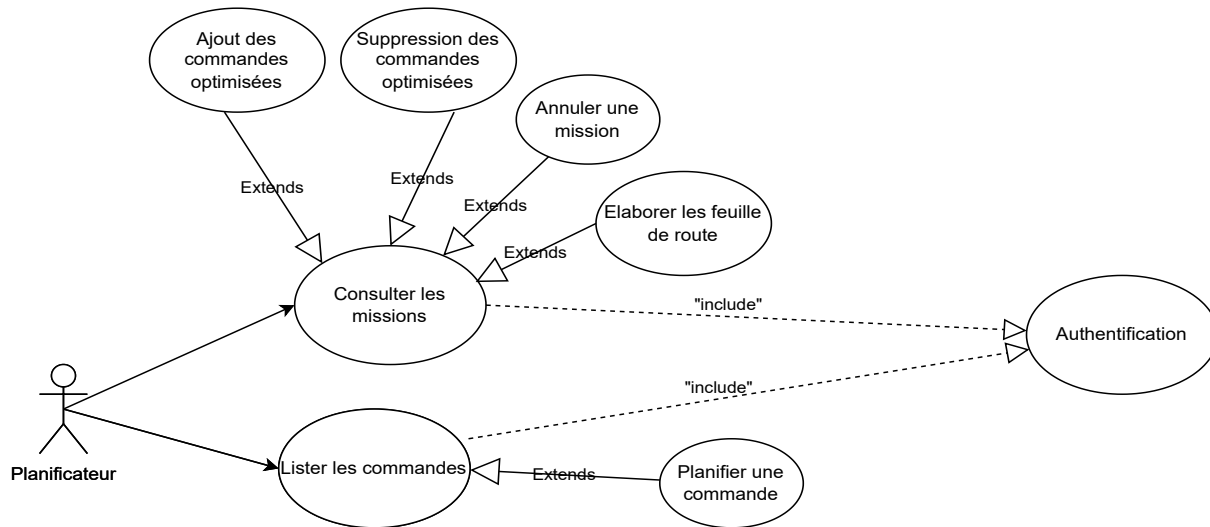


FIGURE 2.8 – Diagramme de cas d'utilisation de l'acteur « Planificateur »

2.7.5 Diagramme de cas d'utilisation de l'acteur « Coordinateur logistique »

La figure 2.9 recense les fonctionnalités du coordinateur logistique, décrites en diagramme de cas d'utilisation.

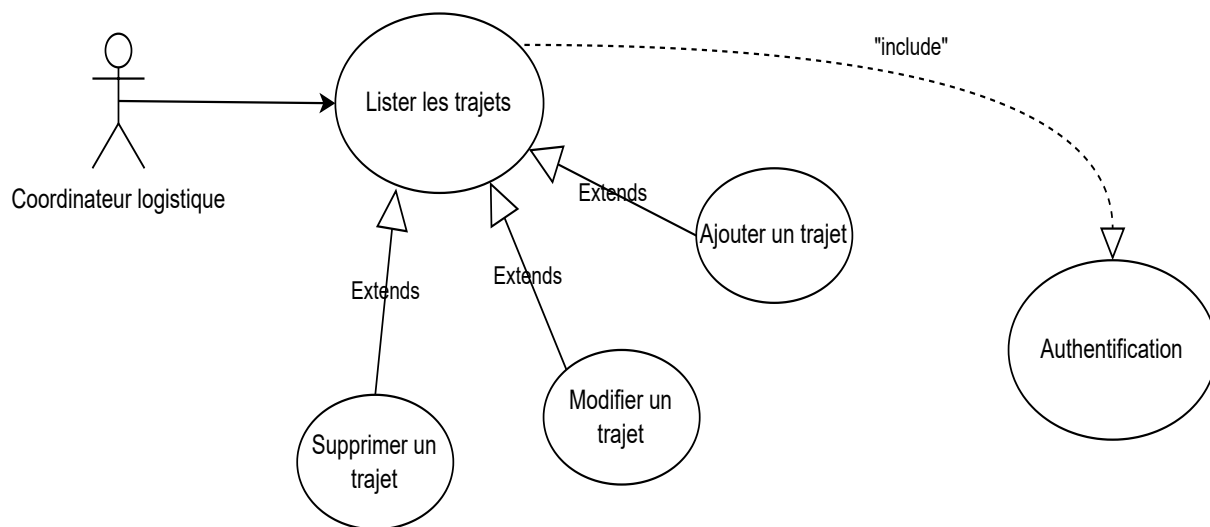


FIGURE 2.9 – Diagramme de cas d'utilisation de l'acteur « Coordinateur logistique »

2.7.6 Diagramme de cas d'utilisation de l'acteur « Ressource humaine »

La figure 2.10 recense les fonctionnalités du ressource humaine, décrites en diagramme de cas d'utilisation.

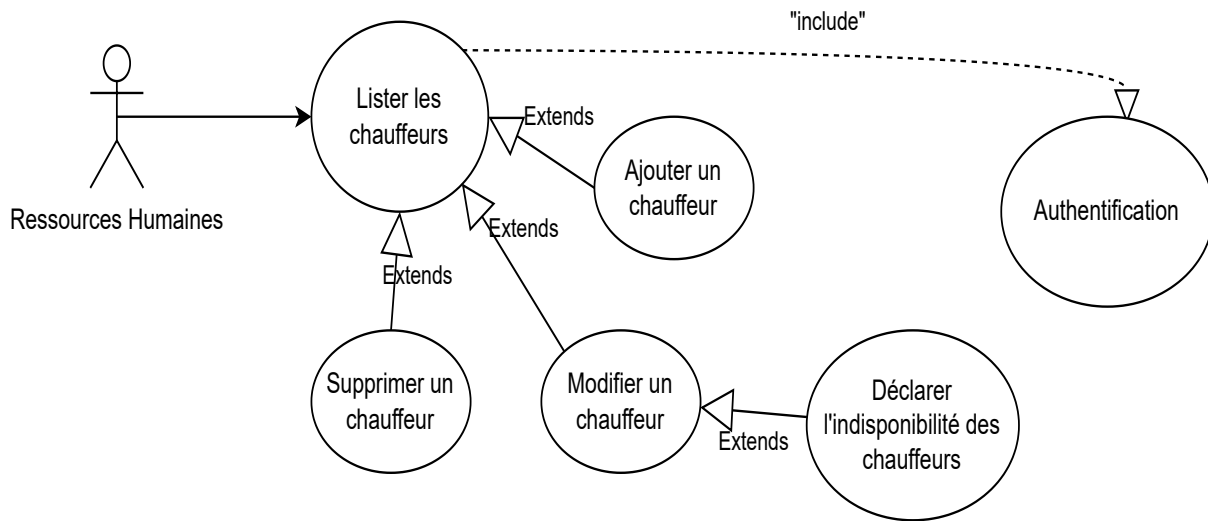


FIGURE 2.10 – Diagramme de cas d'utilisation de l'acteur « Ressource humaine »

2.7.7 Diagramme de cas d'utilisation de l'acteur « GMAO »

La figure 2.11 recense les fonctionnalités du chargé de la maintenance, décrites en diagramme de cas d'utilisation.

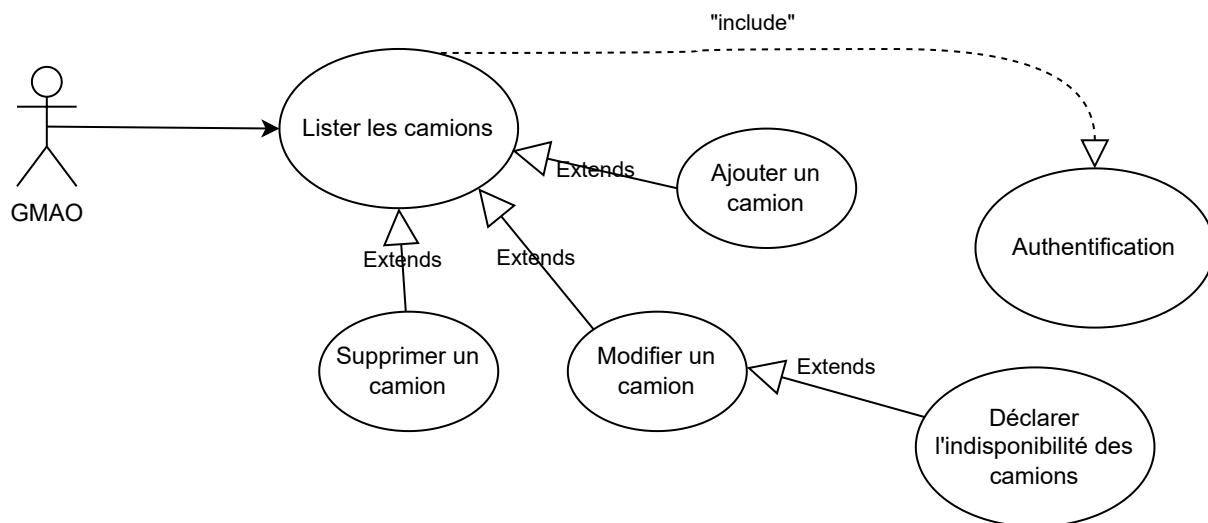


FIGURE 2.11 – Diagramme de cas d'utilisation de l'acteur « GMAO »

2.8 Description des cas d'utilisations

Dans ce qui suit nous allons Décrire les différents cas d'utilisation de notre système.

2.8.1 Description du cas d'utilisation "Authentification"

Le diagramme suivant représente le diagramme de cas d'utilisation de l'authentification effectuer par tout les acteurs.



FIGURE 2.12 – Diagramme de cas d'utilisation «Authentification»

Cas d'utilisation	Authentification
Acteurs	Administrateur, commercial, Planificateur, coordinateur logistique, RH, GMAO
Objectifs	Permet à l'acteur d'accéder à son propre espace
Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • L'acteur doit avoir un compte.
Scénario nominal	<ul style="list-style-type: none"> • Le système invite l'utilisateur à entrer son login et son mot de passe • Scénario nominal. • L'acteur introduit son identifiant et son mot de passe. • Le système vérifie l'existence du compte. • Le système donne l'accès à l'interface correspondante.

Scénario alternatif	<ul style="list-style-type: none"> • Si le login ou le mot de passe sont invalides, le système affiche un message d'erreur. « Votre identifiant ou votre mot de passe est incorrect ».
---------------------	---

TABLE 2.2 – Description textuelle du cas d'utilisation « Authentification »

2.8.2 Description du cas d'utilisation "Gestion des comptes utilisateurs et des groupes de rôles"

Le diagramme suivant représente le diagramme de cas d'utilisation gestion des comptes utilisateurs et des groupes de rôles de notre système.

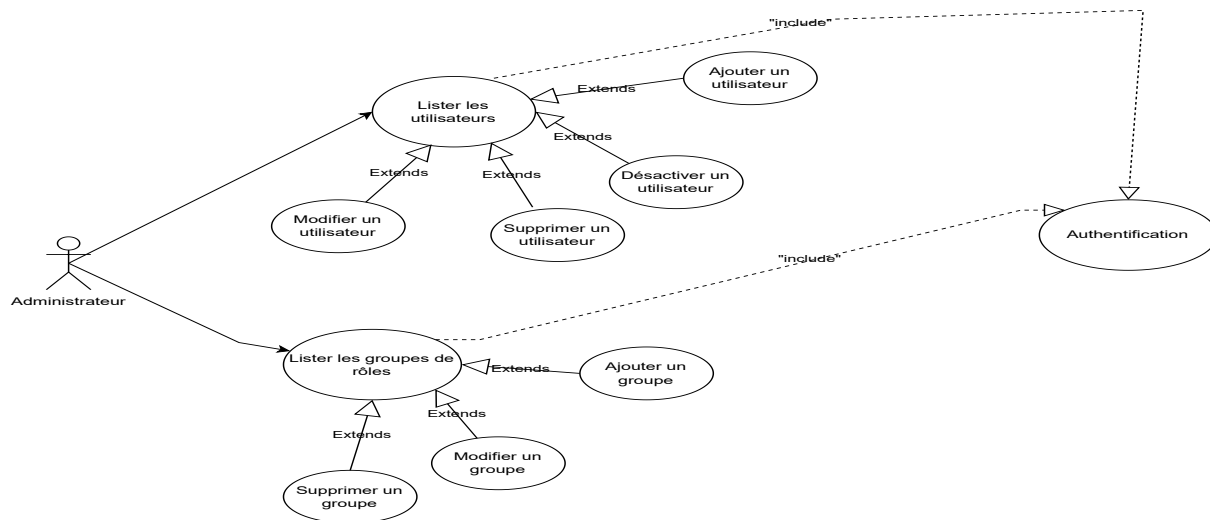


FIGURE 2.13 – Diagramme de cas d'utilisation «Gestion des comptes utilisateurs et des groupes de rôles»

Cas d'utilisation	Gestion des comptes utilisateurs et des groupes de rôles
Acteurs	Administrateur
Objectifs	<p style="text-align: center;">Permet à l'Administrateur de :</p> <ul style="list-style-type: none"> • Visualiser la liste des utilisateurs. • Visualiser la liste des groupes de rôles. • Ajouter, modifier, supprimer et désactiver un utilisateur. • Ajouter, modifier, supprimer et désactiver un groupe de rôles. • Possibilité d'affecter un utilisateurs à un rôle distinct.

Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • S'authentifier en tant qu'administrateur.
Scénario nominal	<ul style="list-style-type: none"> • L'administrateur lance l'application et demande la page regroupant tous les utilisateurs. • Le système l'oblige à passer par l'authentification pour lui autoriser les tâches suivantes : <ol style="list-style-type: none"> 1. La consultation de la liste des utilisateurs ainsi, toutes les manipulations possibles comme l'ajout la suppression la modification et la des-activation de ses derniers. 2. L'administrateur aura également la possibilité de consulter la liste des groupes de rôles avec toutes les options de manipulation citer ci-dessous. 3. Il pourra également affecter un utilisateur pour un ou plusieurs rôles.
Scénario alternatif	Échec d'authentification

TABLE 2.3 – Description textuelle du cas d'utilisation « Gestion des comptes utilisateurs et des groupes de rôles »

2.8.3 Description du cas d'utilisation "Gestion des paramètres de l'application"

Le diagramme suivant représente le diagramme de cas d'utilisation gestion des paramètres de l'application.

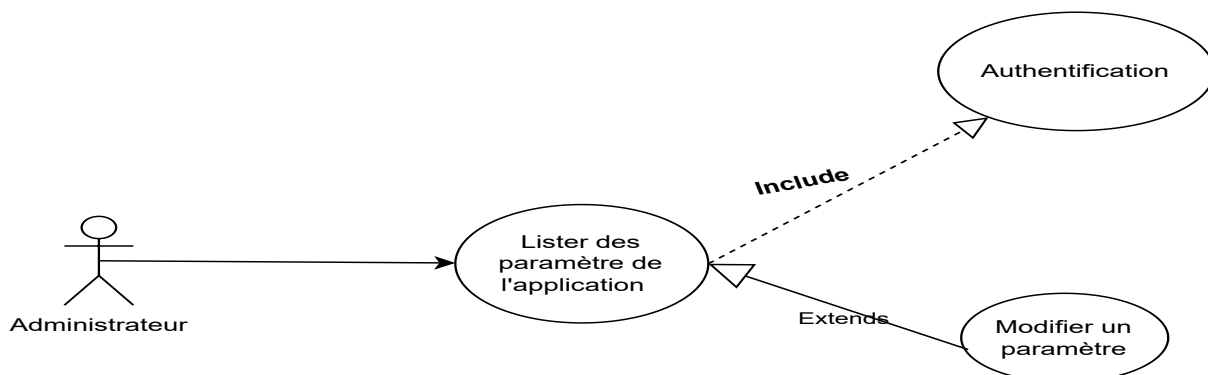


FIGURE 2.14 – Diagramme du cas d'utilisation «Gestion des paramètres»

Cas d'utilisation	Gestion des paramètres généraux de l'application
Acteurs	Administrateur
Objectifs	<p>Permet à l'Administrateur de :</p> <ul style="list-style-type: none"> • Visualiser la liste des paramètres de l'application. • Modifier la valeur des paramètres de l'application.
Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • S'authentifier en tant qu'administrateur.
Scénario nominal	<ul style="list-style-type: none"> • L'administrateur lance l'application et demande la page regroupant tous les paramètres généraux de l'application. • Le système l'oblige à passer vers l'authentification pour lui autoriser les tâches suivantes : <ol style="list-style-type: none"> 1. La consultation de la liste des paramètres et pouvoir en modifier la valeur si nécessaire.
Scénario alternatif	Échec d'authentification

TABLE 2.4 – Description textuelle du cas d'utilisation « Gestion des paramètres généraux de l'application »

2.8.4 Description du cas d'utilisation "Gestion des commandes"

Le diagramme suivant représente le diagramme de cas d'utilisation gestion des commandes logistiques de notre système.

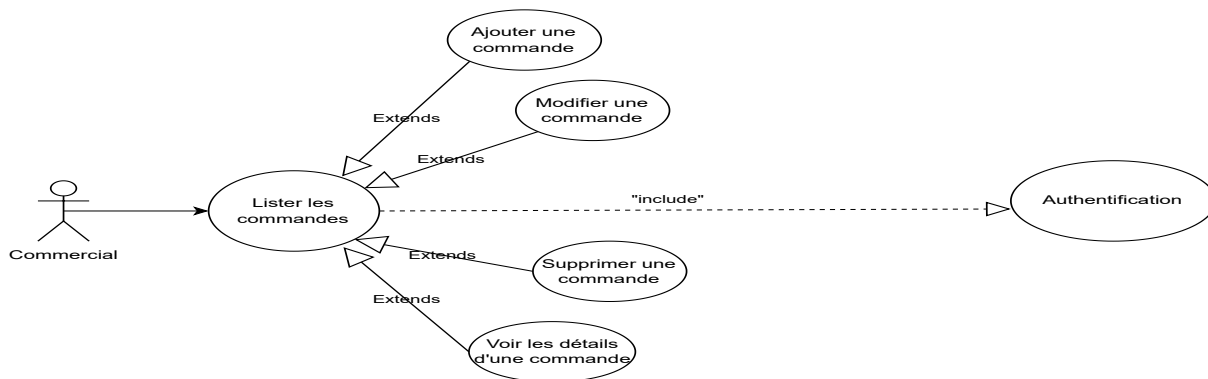


FIGURE 2.15 – Diagramme de cas d'utilisation «Gestion des Commandes»

Cas d'utilisation	Gestion des Commandes logistiques
acteurs	Commercial
objectifs	<p>Permet au commercial de :</p> <ul style="list-style-type: none"> • Visualiser la liste des commandes clients. • Exécuter divers opérations : <ol style="list-style-type: none"> 1. L'ajout d'une commande. 2. La modification d'une commande. 3. La suppression d'une commande. 4. Visualisation des détails d'une commande.
Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • S'authentifier en tant que commercial
Scénario nominal	<ul style="list-style-type: none"> • Le commercial lance l'application et demande à consulter la liste des commandes. • Le système l'oblige à s'authentifier pour lui autoriser un ensemble de fonctionnalités. • L'ajout d'une commande. • La mise à jour d'une commande. • la visualisation des détails d'une commande • La suppression d'une commande.
Scénario alternatif	Échec d'authentification

TABLE 2.5 – Description textuelle du cas d'utilisation « Gestion des Commandes logistique »

2.8.5 Description du cas d'utilisation "Gestion des ClientsBL"

Le diagramme suivant représente le diagramme de cas d'utilisation gestion des ClientBL de notre système.

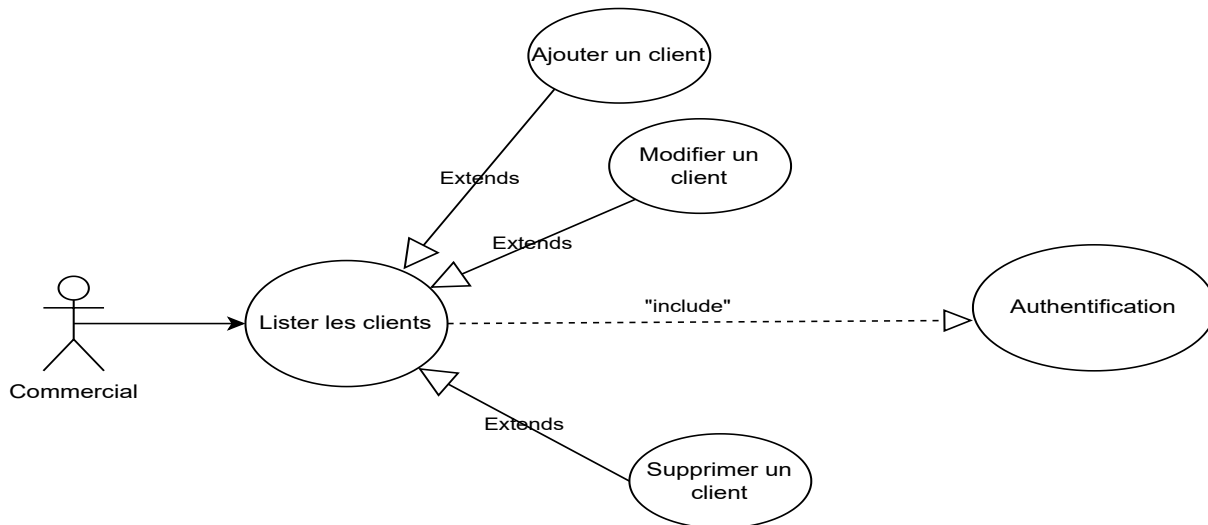


FIGURE 2.16 – Diagramme de cas d'utilisation «Gestion d’un clientBL»

Cas d'utilisation	Gestion des clients de l'entreprise
acteurs	Commercial
objectifs	Permet au commercial de : <ul style="list-style-type: none"> • Visualiser la liste des clients. • Exécuter divers opérations : <ol style="list-style-type: none"> 1. L'ajout d'un client. 2. La modification d'un client. 3. La suppression d'un client.
Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • S'authentifier en tant que commercial

Scénario nominal	<ul style="list-style-type: none"> • Le commercial lance l'application et demande la consultation de la liste des clients. • Le système l'oblige à s'authentifier pour lui autoriser un ensemble de fonctionnalités. • L'ajout d'un client. • La mise à jour d'un client. • La suppression d'un client.
Scénario alternatif	Échec d'authentification

TABLE 2.6 – Description textuelle du cas d'utilisation « gestion des Clients de l'entreprise BL »

2.8.6 Description du cas d'utilisation "Planification des commandes"

Le diagramme suivant représente le diagramme de cas d'utilisation détaillé planification des commandes logistique associé au planificateur.

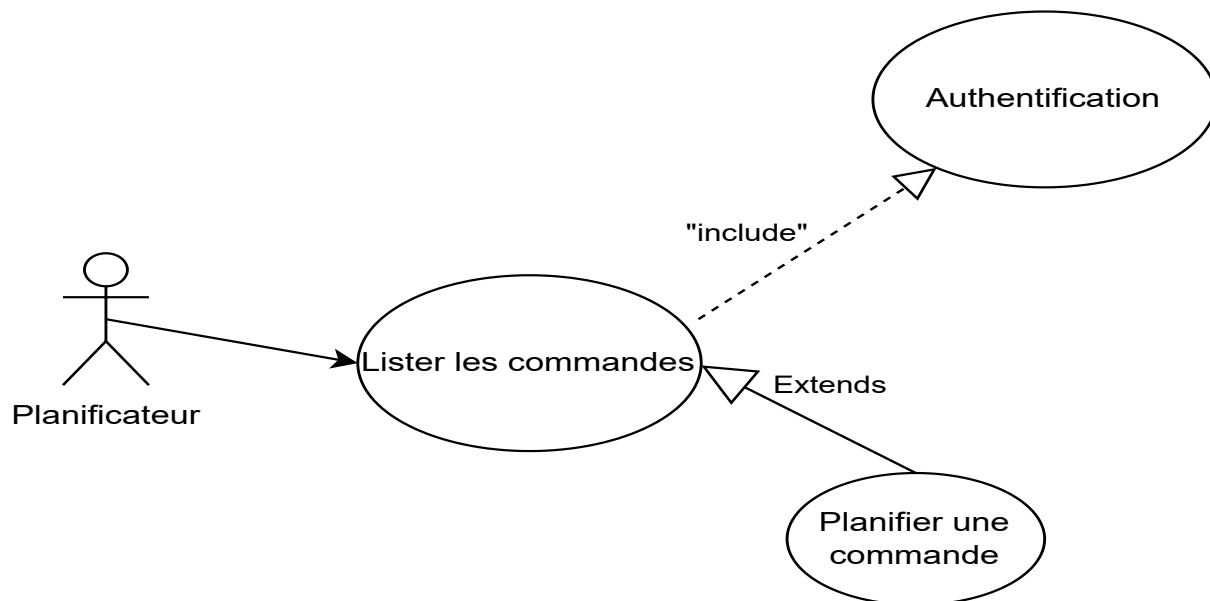


FIGURE 2.17 – Diagramme de cas d'utilisation «Planification des Commandes»

Cas d'utilisation	Planification d'une commande logistique
Acteurs	Planificateur
Objectifs	<p>Permet au planificateur de :</p> <ul style="list-style-type: none"> • Visualiser l'ensemble des Commande clients. • Visualiser le résultat de l'optimisation des ressources (Camions et chauffeurs) pour une commande donnée. • Choisir les ressources adéquates et valider la mission.
Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • S'authentifier en tant que planificateur.
Scénario nominal	<ul style="list-style-type: none"> • Le planificateur lance l'application et demande la liste des commandes clients. • Le système l'oblige à s'authentifier pour lui autoriser un ensemble de fonctionnalités. • L'affichage de la liste des commandes. • Possibilité de voir les détails d'une commande logistique. <ol style="list-style-type: none"> 1. Avoir une vue sur les ressources (Camions, Chauffeurs) adéquat proposé par le système optimisé. 2. Avoir une vue sur les ressources (Camions, Chauffeurs) existants dans la base de données y compris les non-disponibles. 3. Possibilité de planifier une commande en choisissant les ressources voulues. 4. Possibilité de créer une mission en fonction des résultats de la planification à savoir(informations de la commande + ressources adéquates).

TABLE 2.7 – Description textuelle du cas d'utilisation «Planification des commandes»

2.8.7 Description du cas d'utilisation "Gestion des missions"

Le diagramme suivant représente le diagramme de cas d'utilisation détaillé gestion des missions associé au planificateur.

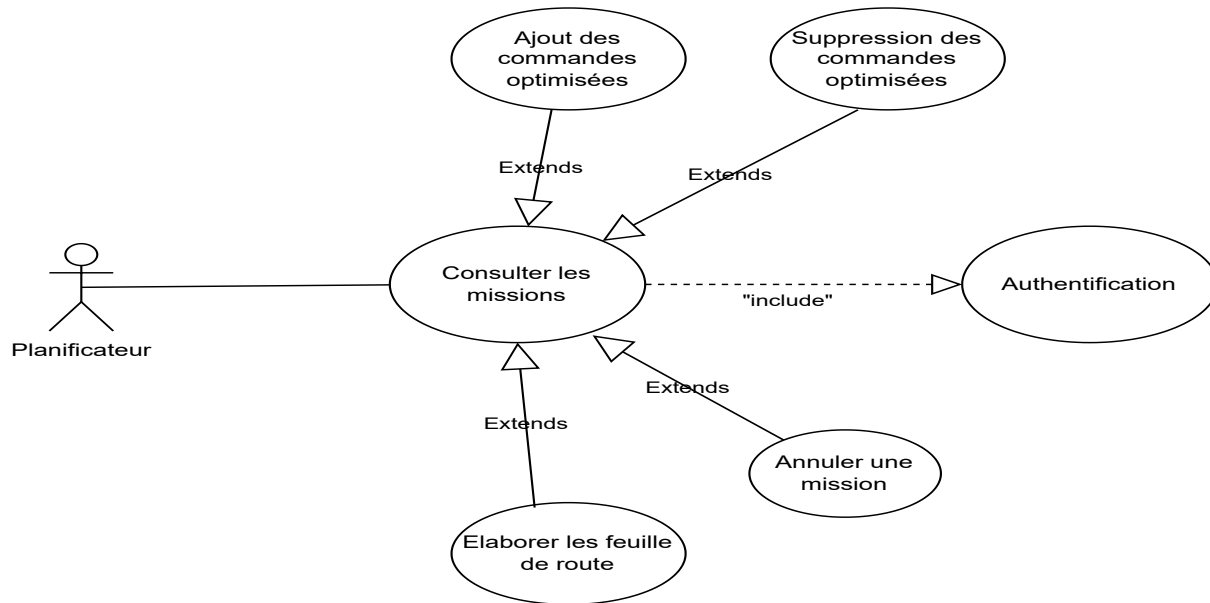


FIGURE 2.18 – Diagramme de cas d'utilisation «Gestion d'une mission»

Cas d'utilisation	Gestion d'une mission
Acteurs	Planificateur
Objectifs	<p>Permet au planificateur de :</p> <ul style="list-style-type: none"> • Gestion des missions. • Avoir une visualisation du résultat de l'optimisation du système quant aux commandes programmées aux alentours du point de déchargement de la mission existante.

Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • S'authentifier en tant que planificateur.
Scénario nominal	<ul style="list-style-type: none"> • Le planificateur lance l'application et demande la consultation de la liste des missions. • Le système l'oblige à passer par l'authentification pour lui autoriser un ensemble de fonctionnalités. <ul style="list-style-type: none"> — Visualisation des détails d'une mission. — Possibilité d'ajouter ou de supprimer des commandes proposer par le système optimisé. — Pouvoir élaborer une feuille de route. — Possibilité d'annuler une mission.
Scénario alternatif	Échec d'authentification

TABLE 2.8 – Description textuelle du cas d'utilisation « Gestion d'une mission »

2.8.8 Description du cas d'utilisation "Gestion des trajets BL"

Le diagramme suivant représente le diagramme de cas d'utilisation Gestion des Trajets associé au coordinateur logistique.

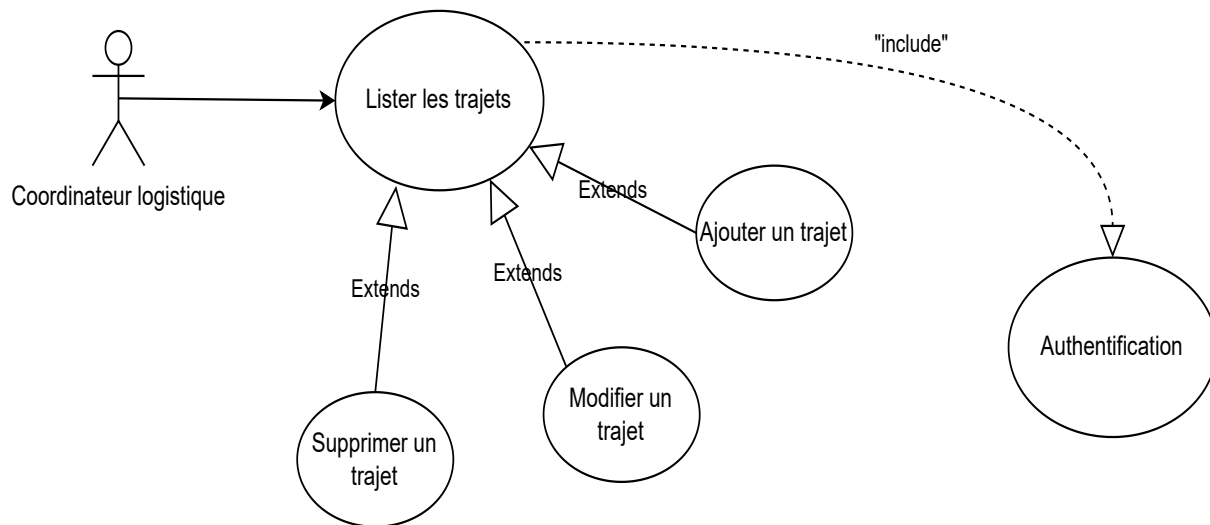


FIGURE 2.19 – Diagramme du cas d'utilisation «Gestion des trajets BL»

Cas d'utilisation	Gestion des Trajets BL
acteurs	Coordinateur logistique
objectifs	<p>Permet au CL de :</p> <ul style="list-style-type: none"> • Visualiser la liste des trajet existant. • Exécuter divers opérations : <ol style="list-style-type: none"> 1. L'ajout d'un trajet. 2. La modification d'un trajet. 3. La suppression d'un trajet.
Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • s'authentifier en tant que coordinateur logistique.
Scénario nominal	<ul style="list-style-type: none"> • Le CL lance l'application et demande la consultation de l'espace trajet. • Le système l'oblige à s'authentifier pour lui autoriser un ensemble de fonctionnalités. <ol style="list-style-type: none"> 1. L'ajout d'un trajet. 2. La mise à jour d'un trajet. 3. La suppression d'un trajet.
Scénario alternatif	Échec d'authentification

TABLE 2.9 – Description textuelle du cas d'utilisation « Gestion des Trajet BL »

2.8.9 Description du cas d'utilisation "Gestion des chauffeurs"

Le diagramme suivant représente le diagramme de cas d'utilisation Gestion des chauffeurs associé au RH.

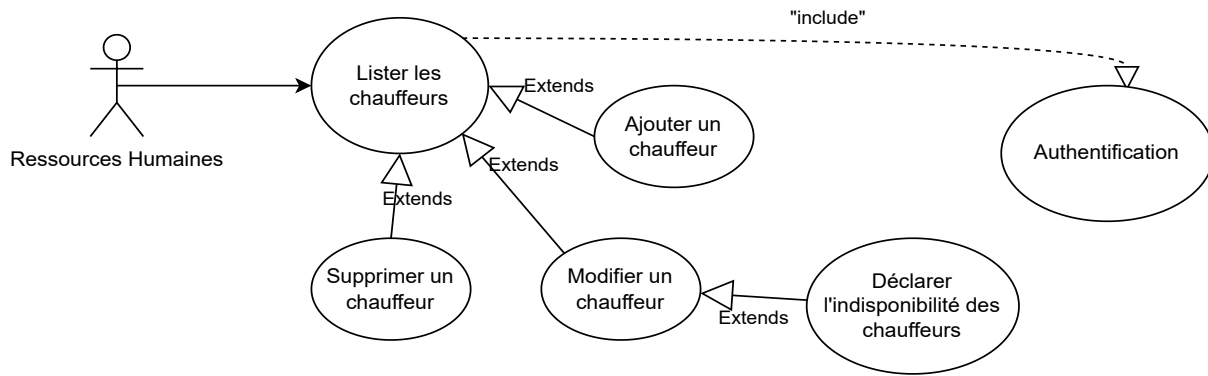


FIGURE 2.20 – Diagramme de cas d'utilisation «Gestion des chauffeurs »

Cas d'utilisation	Gestion des chauffeurs BL
acteurs	Ressources Humaines
objectifs	<p>Permet au RH de :</p> <ul style="list-style-type: none"> • Visualiser la liste des chauffeur existant. • Exécuter divers opérations : <ol style="list-style-type: none"> 1. L'ajout d'un chauffeur. 2. La modification d'un chauffeur. 3. La suppression d'un chauffeur.
Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • S'authentifier en tant que RH.
Scénario nominal	<ul style="list-style-type: none"> • Le RH lance l'application et demande la consultation de l'espace chauffeur. • Le système l'oblige à s'authentifier pour lui autoriser un ensemble de fonctionnalités. <ol style="list-style-type: none"> 1. L'ajout d'un chauffeur. 2. La mise à jour d'un chauffeur. 3. La suppression d'un chauffeur. 4. Déclarer une situation d'indisponibilité d'un chauffeur.
Scénario alternatif	Échec d'authentification

TABLE 2.10 – Description textuelle de cas d'utilisation « gestion des Chauffeurs BL »

2.8.10 Description du cas d'utilisation "Gestion des véhicules"

Le diagramme suivant représente le diagramme de cas d'utilisation Gestion des véhicules associé au GMAO.

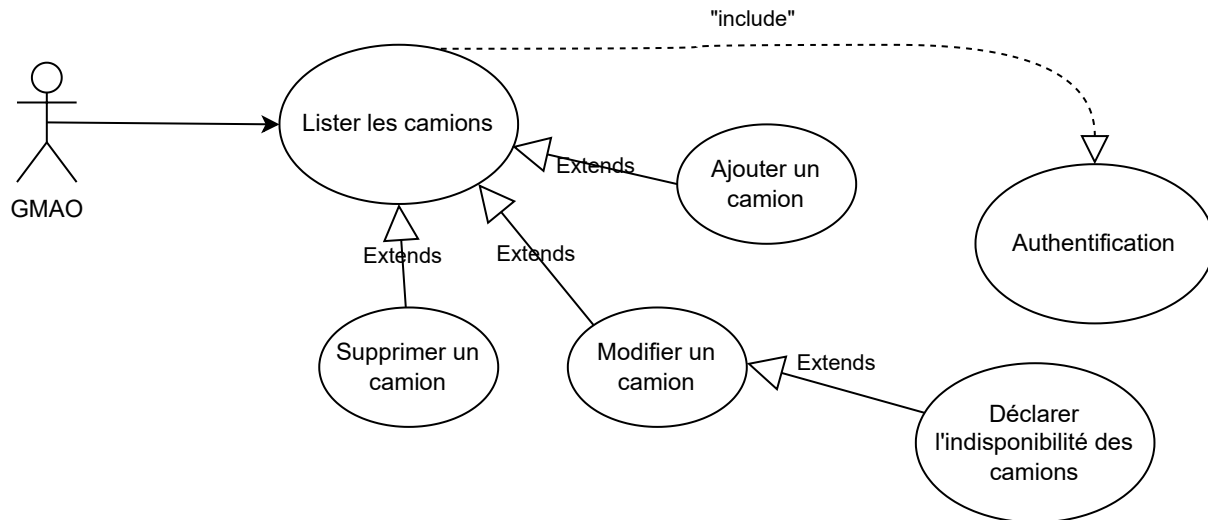


FIGURE 2.21 – Diagramme du cas d'utilisation «Gestion des camions »

Cas d'utilisation	Gestion des camions BL
acteurs	GMAO
objectifs	Permet au GMAO de : <ul style="list-style-type: none"> • Visualiser la liste des camion existant. • Exécuter divers opérations : <ol style="list-style-type: none"> 1. L'ajout d'un camion. 2. La modification d'un camion. 3. La suppression d'un camion.
Prédiction	<ul style="list-style-type: none"> • Être connecté à internet. • Avoir un navigateur web. • S'authentifier en tant que GMAO.

Scénario nominal	<ul style="list-style-type: none"> • Le RH lance l'application et demande la consultation de l'espace camion. • Le système l'oblige à s'authentifier pour lui autoriser un ensemble de fonctionnalités. <ol style="list-style-type: none"> 1. L'ajout d'un camion. 2. La mise à jour d'un camion. 3. La suppression d'un camion. 4. Déclarer une situation d'indisponibilité d'un camion.
Scénario alternatif	Échec d'authentification

TABLE 2.11 – Description textuelle du cas d'utilisation « Gestion des camions »

2.9 Diagrammes de séquence

2.9.1 Concepts de bases

Les diagrammes de séquences sont la représentation graphique des interactions acteurs et système avec un ordre chronologique dans la formulation UML. Ces communications entre les classes sont reconnues comme des messages.

Il modélise l'exécution des différents messages en fonction du temps. Pour réaliser les diagrammes des séquences nous avons utilisé des opérateurs d'interactions. Un opérateur d'interaction dénote le type d'un fragment composé.

Les opérateurs d'interaction que nous avons utilisés dans les diagrammes de séquences sont :

- **Référence (Ref) :**

Cet opérateur désigne que le fragment fait référence à un cas vu précédemment.

- **Alternative(Alt) :**

Cet opérateur désigne que le fragment composé représente un choix de comportement. Un opérande d'interaction au maximum sera choisi. L'opérande choisi doit avoir une expression de garde implicite ou explicite qui a la valeur true à ce point de l'interaction.

- **Loop :**

Cet opérateur désigne que le fragment composé représente une boucle[20].

Les diagrammes de séquences retenus à la fin de la conception sont les suivants :

2.9.2 Diagramme de séquence pour le cas d'utilisation « Authentification »

La figure 2.22 représente le diagramme de séquence du cas d'utilisation « Authentification », retenu à la fin de la conception.

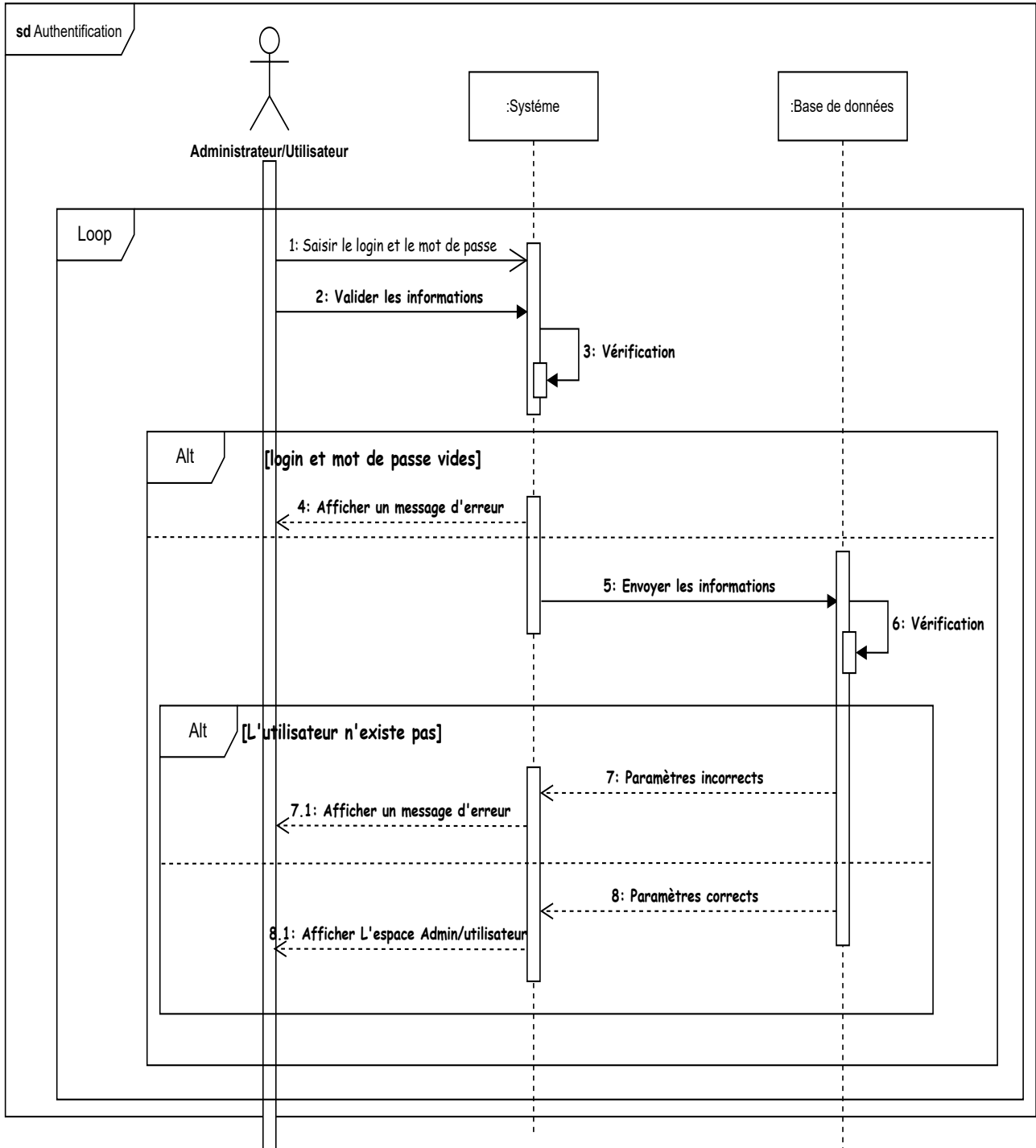


FIGURE 2.22 – Diagramme de séquence du cas d'utilisation « Authentification »

2.9.3 Diagramme de séquence du cas d'utilisation « Ajouter une commande »

La figure 2.23 représente le diagramme de séquence du cas d'utilisation «Ajouter une commande», retenu à la fin de la conception.

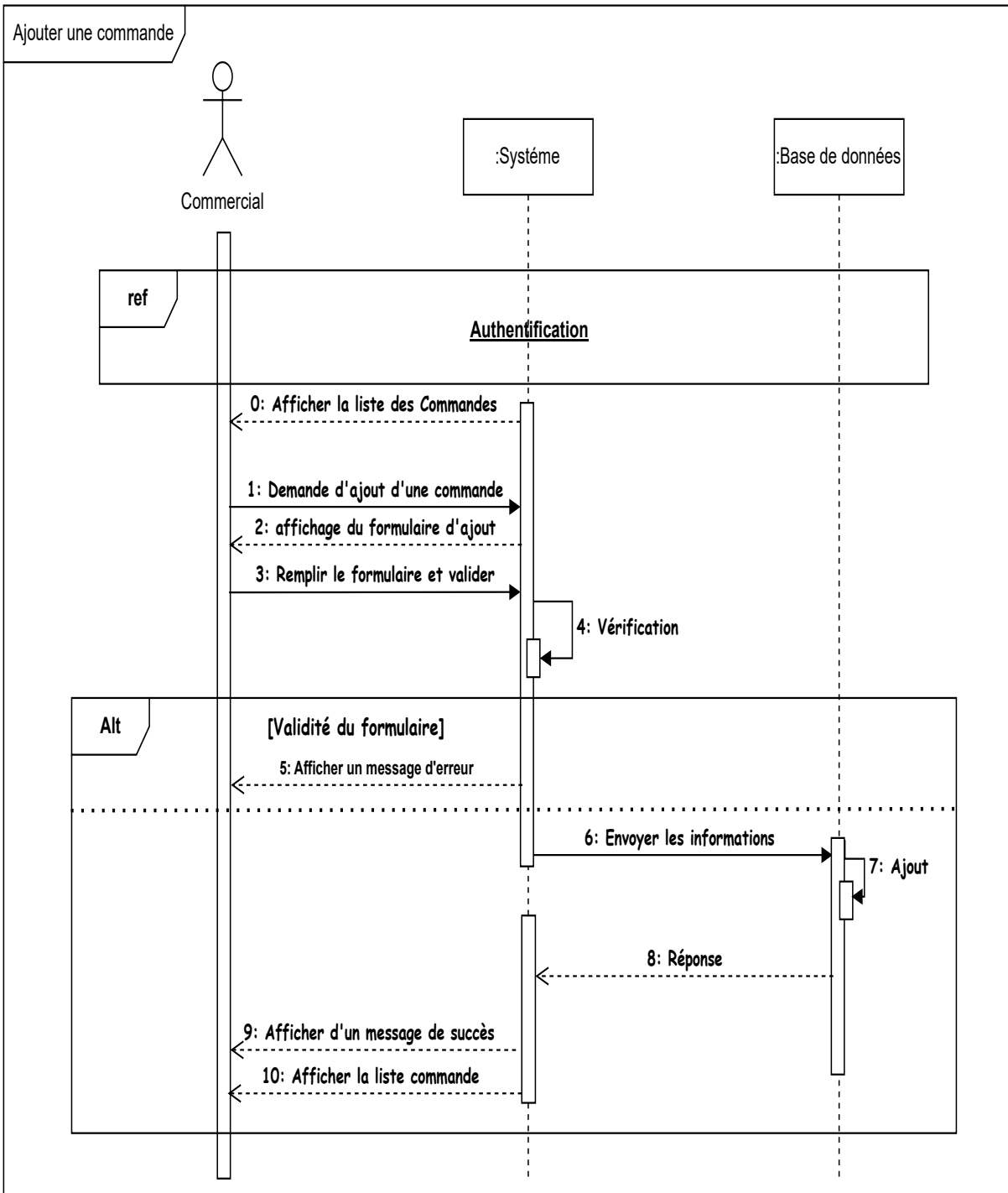


FIGURE 2.23 – Diagramme de séquence du cas d'utilisation « Ajouter une commande »

2.9.4 Diagramme de séquence du cas d'utilisation « Modifier une commande »

La figure 2.24 représente le diagramme de séquence système du cas d'utilisation «Modifier une commande», retenu à la fin de la conception.

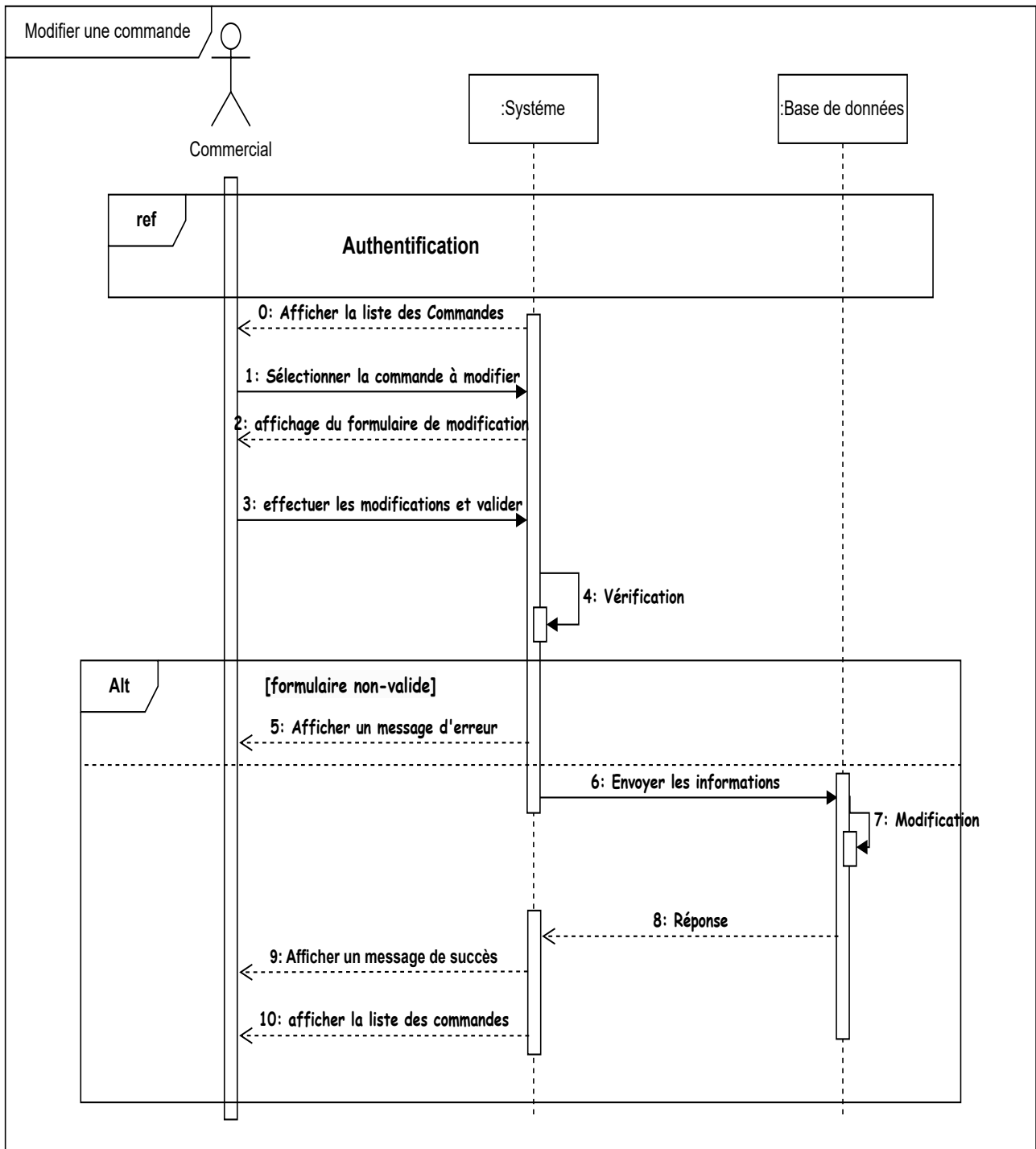


FIGURE 2.24 – Diagramme de séquence du cas d'utilisation « Modifier une commande »

2.9.5 Diagramme de séquence du cas d'utilisation « Supprimer une commande »

La figure 2.25 représente le diagramme de séquence du cas d'utilisation «Supprimer une commande», retenu à la fin de la conception :

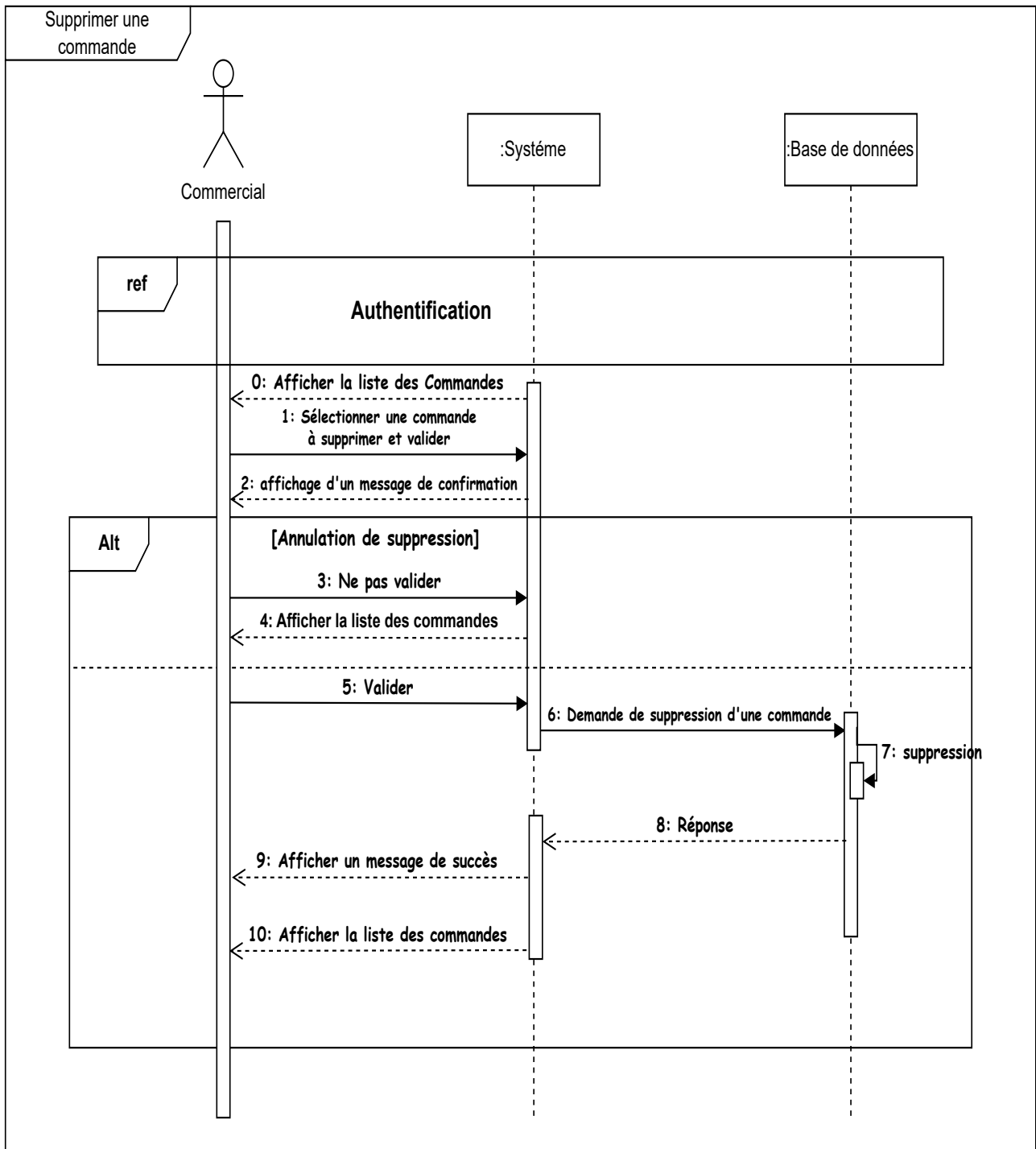


FIGURE 2.25 – Diagramme de séquence du cas d'utilisation « Supprimer une commande »

2.9.6 Diagramme de séquence du cas d'utilisation « Rechercher une commande »

La figure 2.26 représente le diagramme de séquence du cas d'utilisation «Rechercher une commande», retenu à la fin de la conception.

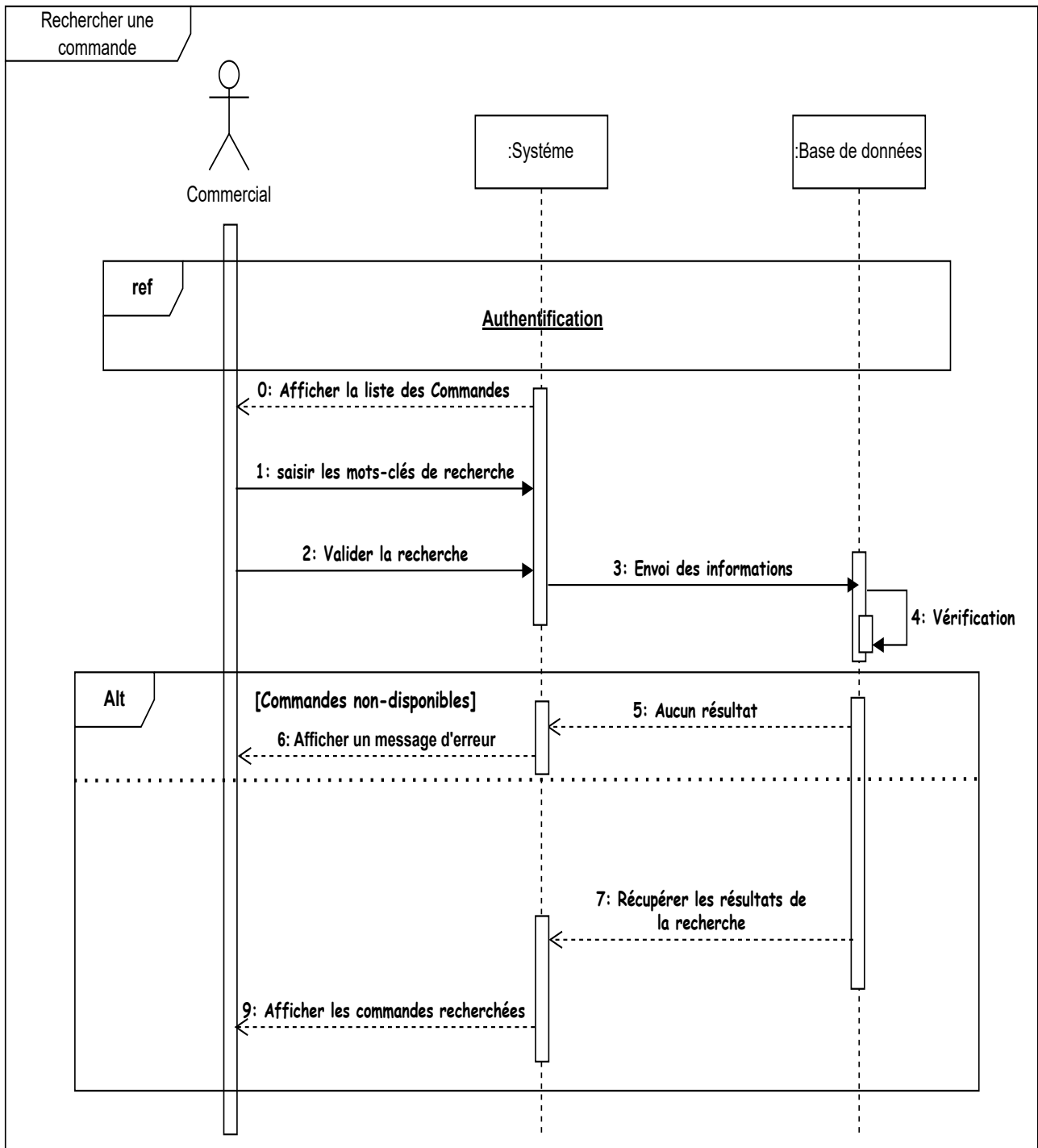


FIGURE 2.26 – Diagramme de séquence du cas d'utilisation « Rechercher une commande »

2.9.7 Diagramme de séquence du cas d'utilisation « Planifier une commande »

La figure 2.27 représente le diagramme de séquence du cas d'utilisation «Planifier une commande», retenu à la fin de la conception :

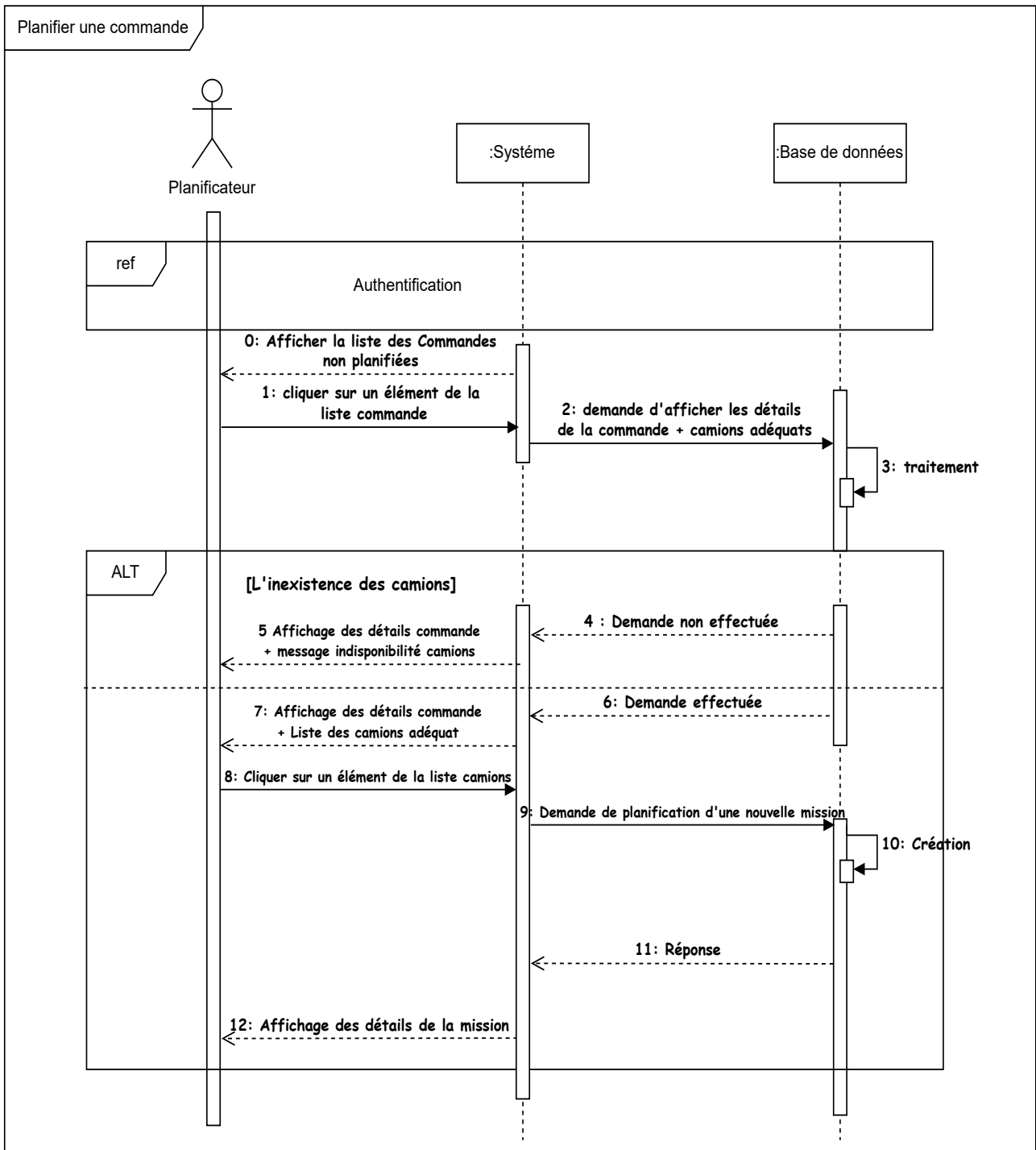


FIGURE 2.27 – Diagramme de séquence du cas d'utilisation « Planifier une commande »

2.9.8 Diagramme de séquence du cas d'utilisation « Gérer une mission »

La figure 2.28 représente le diagramme du cas d'utilisation «Gérer une mission», retenu à la fin de la conception :

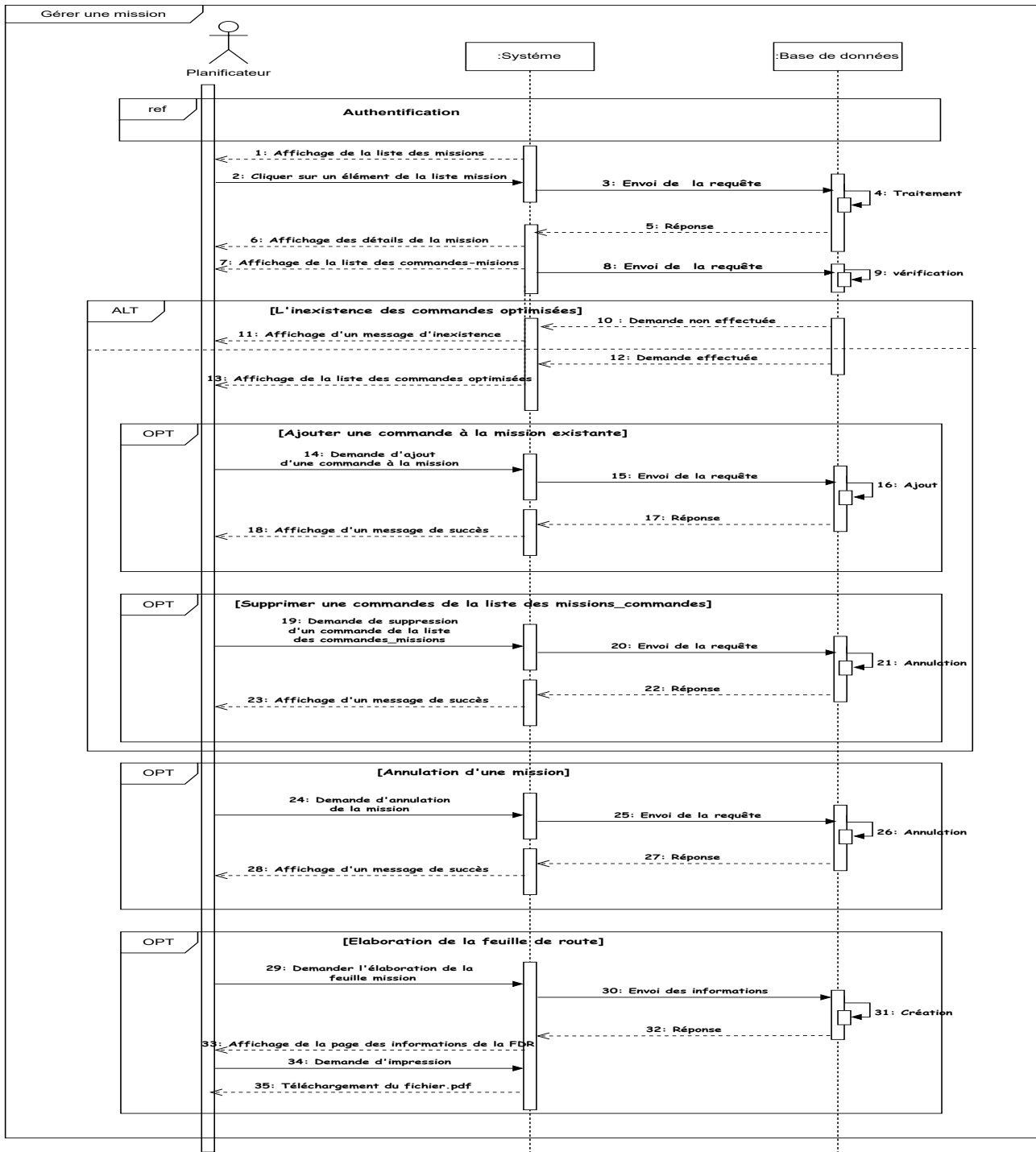


FIGURE 2.28 – Diagramme de séquence du cas d'utilisation « Gérer une mission »

2.10 Conclusion

Dans ce chapitre, nous avons vu l'importance de l'étude de l'existant dans tout projet informatique et comment elle permet d'élaborer une conception solide. Nous avons exposé la démarche de développement suivie pour la réalisation de notre projet, puis nous avons poursuivi par l'identification des besoins fonctionnels et non-fonctionnels, la spécification des différents acteurs qui interagiront avec le système, ainsi que l'élaboration du diagramme de contexte et des diagrammes de cas d'utilisation. Nous avons conclu notre étude en élaborant des diagrammes de séquences système spécifiques à chaque cas d'utilisation.

Cette analyse approfondie de l'existant nous a permis de mieux comprendre les besoins des utilisateurs et de concevoir un système répondant à leurs attentes. En spécifiant les différents cas d'utilisation et en établissant les diagrammes de séquences système, nous avons pu décrire les différentes interactions entre les acteurs et le système, ce qui nous a permis de concevoir un système cohérent et efficace. En somme, ce chapitre nous a permis de mieux comprendre l'importance de l'analyse de l'existant dans tout projet informatique et comment elle peut conduire à une conception solide et pertinente.

Chapitre 3

Analyse et Conception détaillée

3.1 Introduction

La conception de toute solution logicielle doit être traitée avec précision. Précédée d'une analyse profonde et bien réfléchie, cette étape complète les spécifications des diagrammes de l'étape d'analyse et fournit les diagrammes détaillés. Cette dernière constitue le reflet du futur système avant même sa concrétisation. Les diagrammes détaillés offrent une vision claire de la structure et des fonctionnalités du système, permettant ainsi de planifier efficacement les étapes suivantes du développement.

En accordant une attention particulière à la conception, nous nous assurons que la solution logicielle répondra aux attentes et aux exigences spécifiques de nos clients. Cela nous permet également d'identifier les améliorations potentielles et d'anticiper les évolutions futures du système.

3.2 Modélisation des diagrammes de séquences détaillés

Les diagrammes de séquences détaillés sont des représentations graphiques détaillées des interactions selon leur ordre chronologique. Ces diagrammes sont aussi basés sur le modèle MVC (Model, View, Controller), mais en se référant à notre réalisation nous avons utilisé le modèle MVT (model, view, template), donc nous les avons réadaptés à la définition de ce modèle.

Les diagrammes de séquences détaillés retenus à la fin de la conception sont le suivant :

3.2.1 Diagramme de séquence détaillé du cas d'utilisation «Authentification»

La figure 3.1 représente le diagramme de séquence détaillé du cas d'utilisation «Authentification» retenu à la fin de la conception.

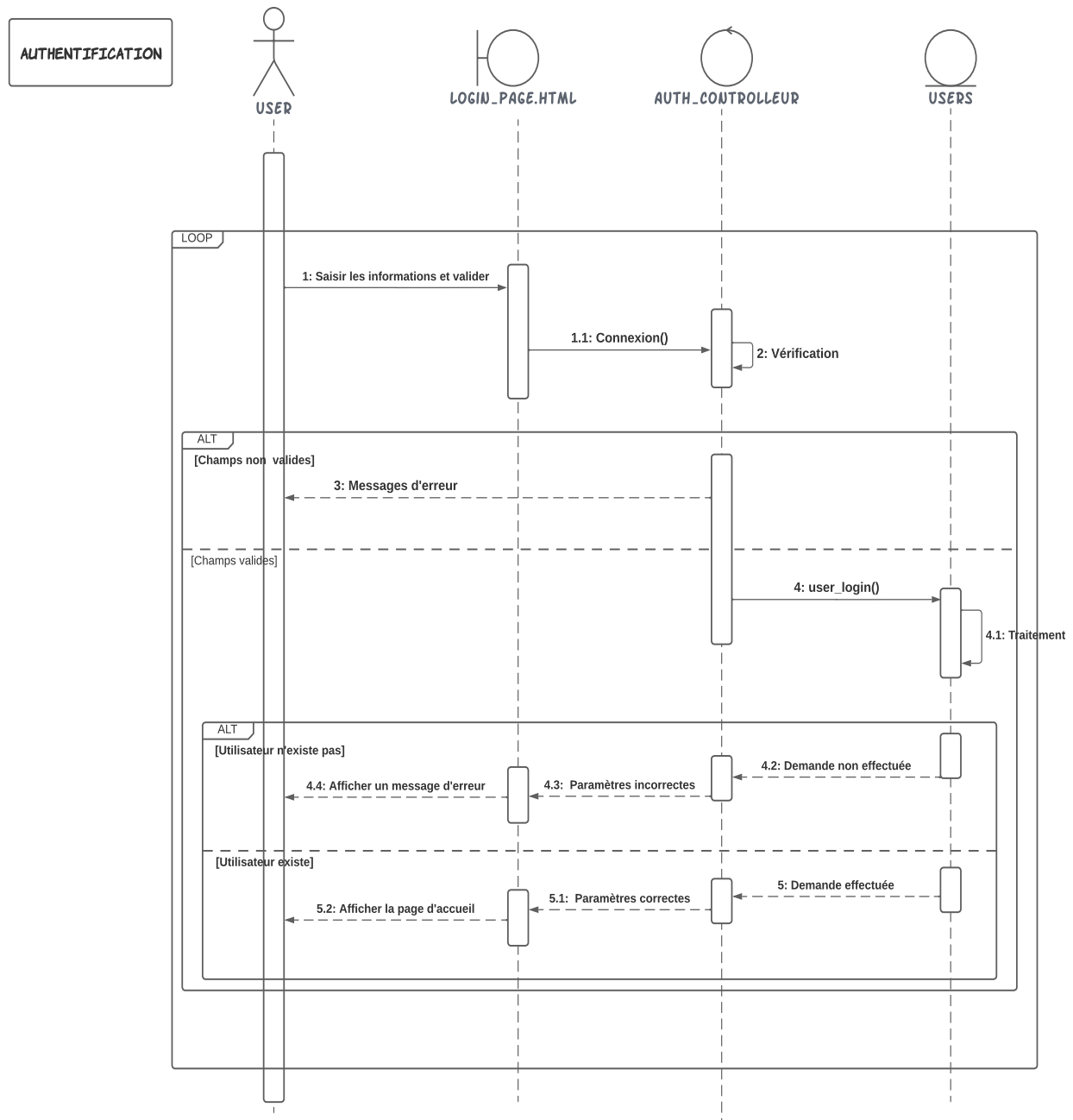


FIGURE 3.1 – Diagramme de séquence détaillé du cas d'utilisation «Authentification»

3.2.2 Diagramme de séquence détaillé du cas d'utilisation «Ajouter une commande»

La figure 3.2 représente le diagramme de séquence détaillé du cas d'utilisation «ajouter une commande» retenu à la fin de la conception.

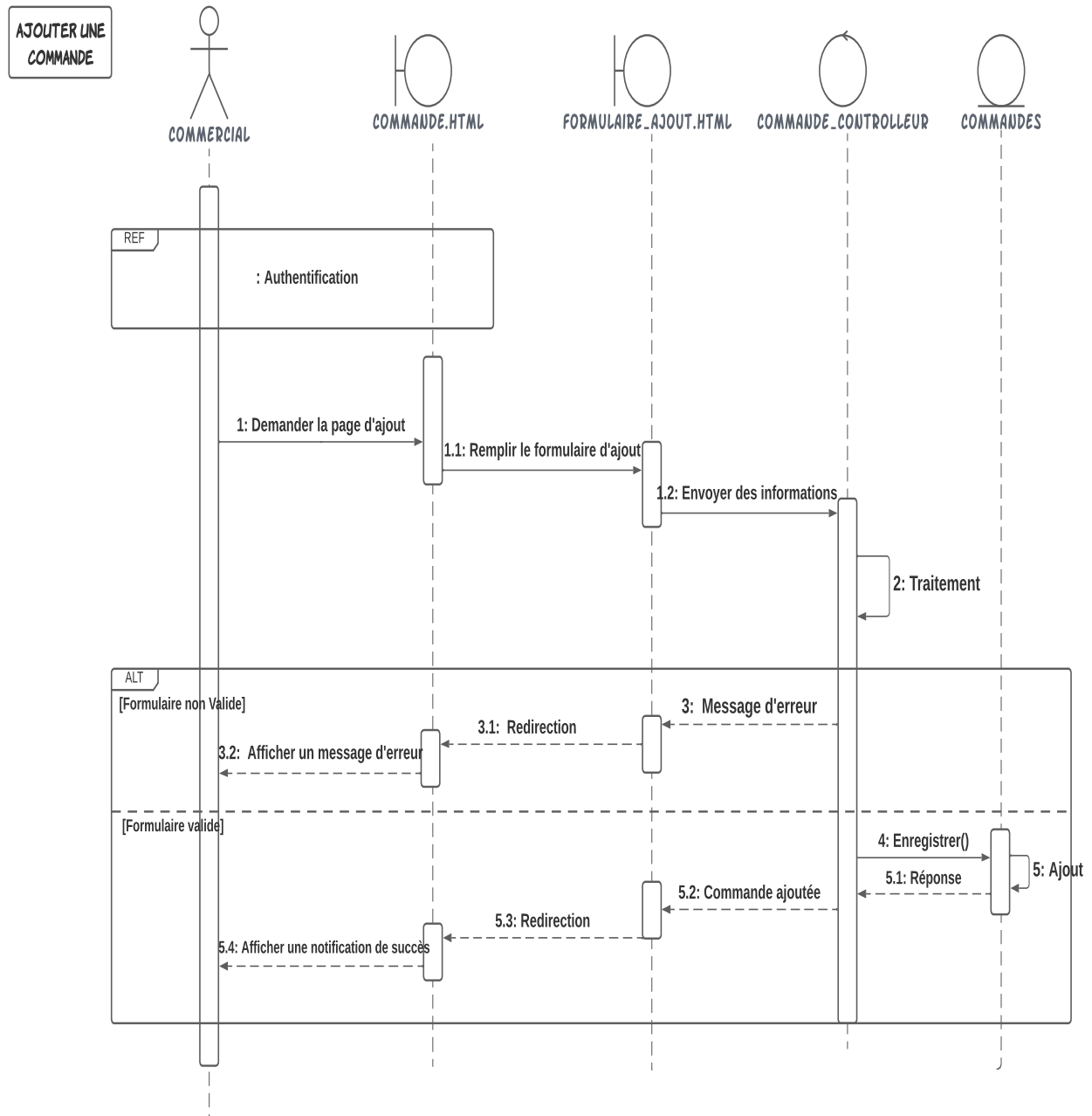


FIGURE 3.2 – Diagramme de séquence détaillé du cas d'utilisation «Ajout d’une commande»

3.2.3 Diagramme de séquence détaillé du cas d'utilisation « Modifier une commande »

La figure 3.3 représente le diagramme de séquence détaillé du cas d'utilisation « modifier une commande » retenu à la fin de la conception.

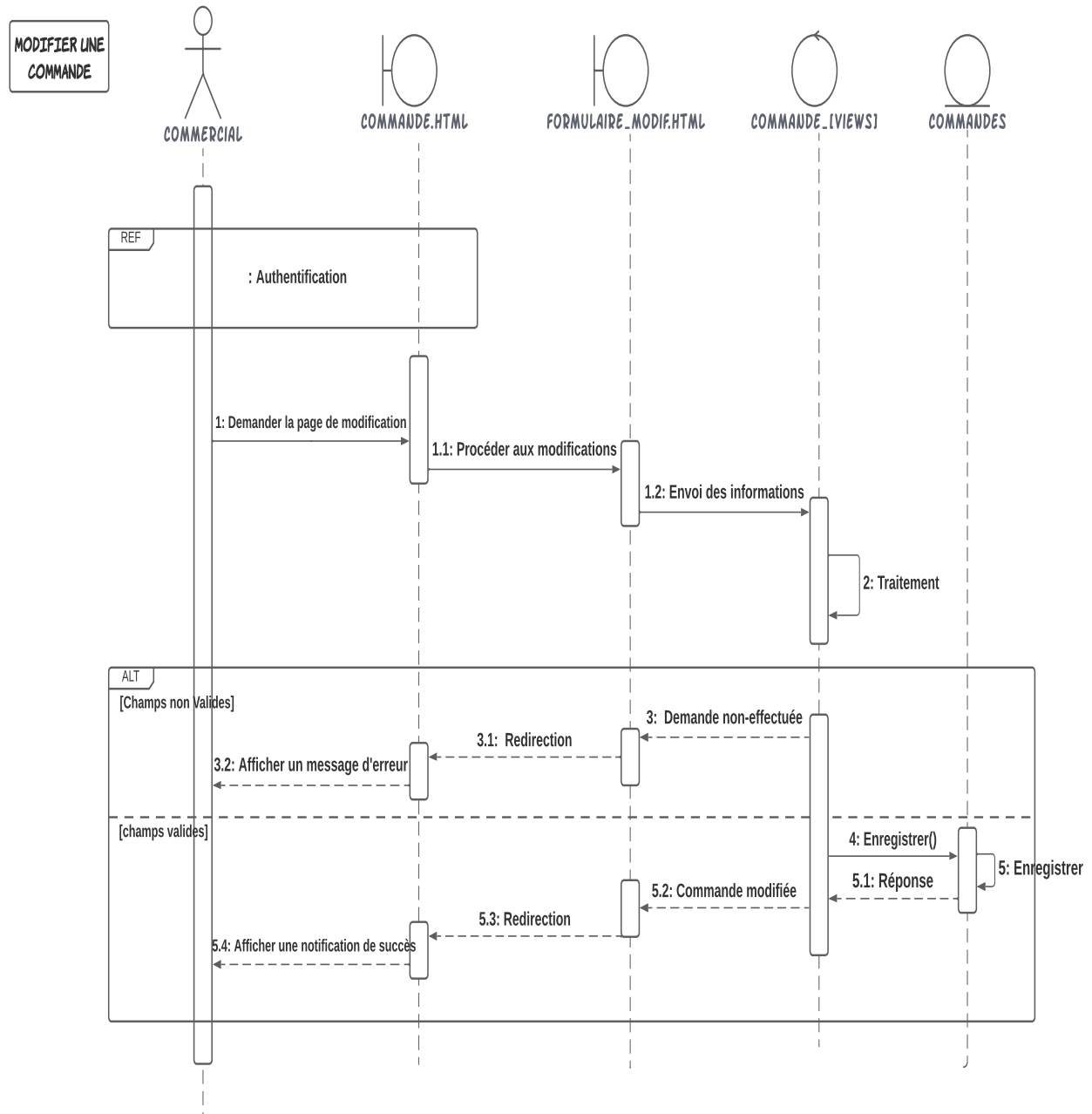


FIGURE 3.3 – Diagramme de séquence détaillé du cas d'utilisation « Modifier une commande »

3.2.4 Diagramme de séquence détaillé du cas d'utilisation «Supprimer une commande»

La figure 3.4 représente le diagramme de séquence détaillé du cas d'utilisation «supprimer une commande» retenu à la fin de la conception.

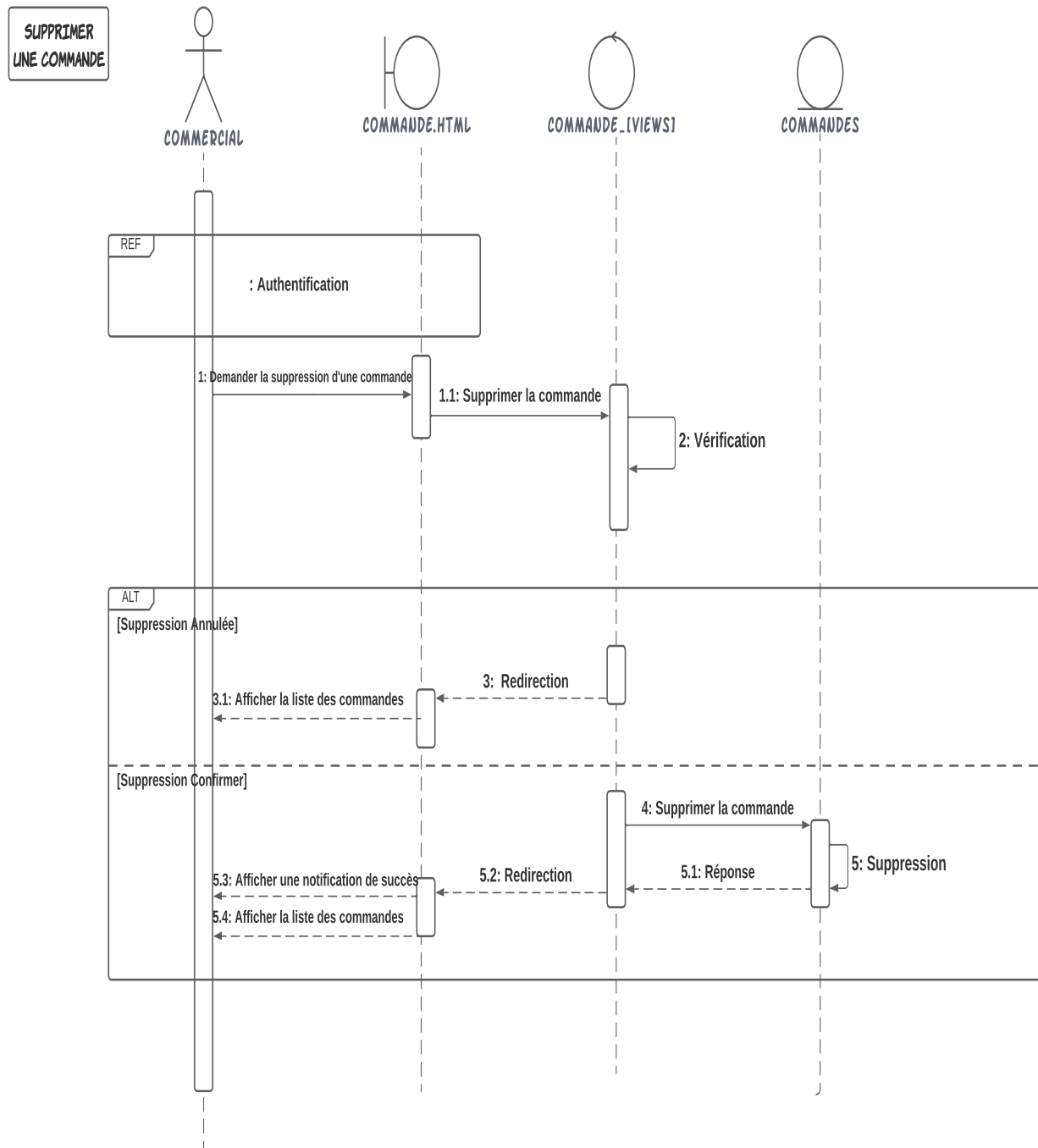


FIGURE 3.4 – Diagramme de séquence détaillé du cas d'utilisation «Suppression d'une commande»

3.2.5 Diagramme de séquence détaillé du cas d'utilisation «Rechercher une commande»

La figure 3.5 représente le diagramme de séquence détaillé du cas d'utilisation «Rechercher une commande» retenu à la fin de la conception.

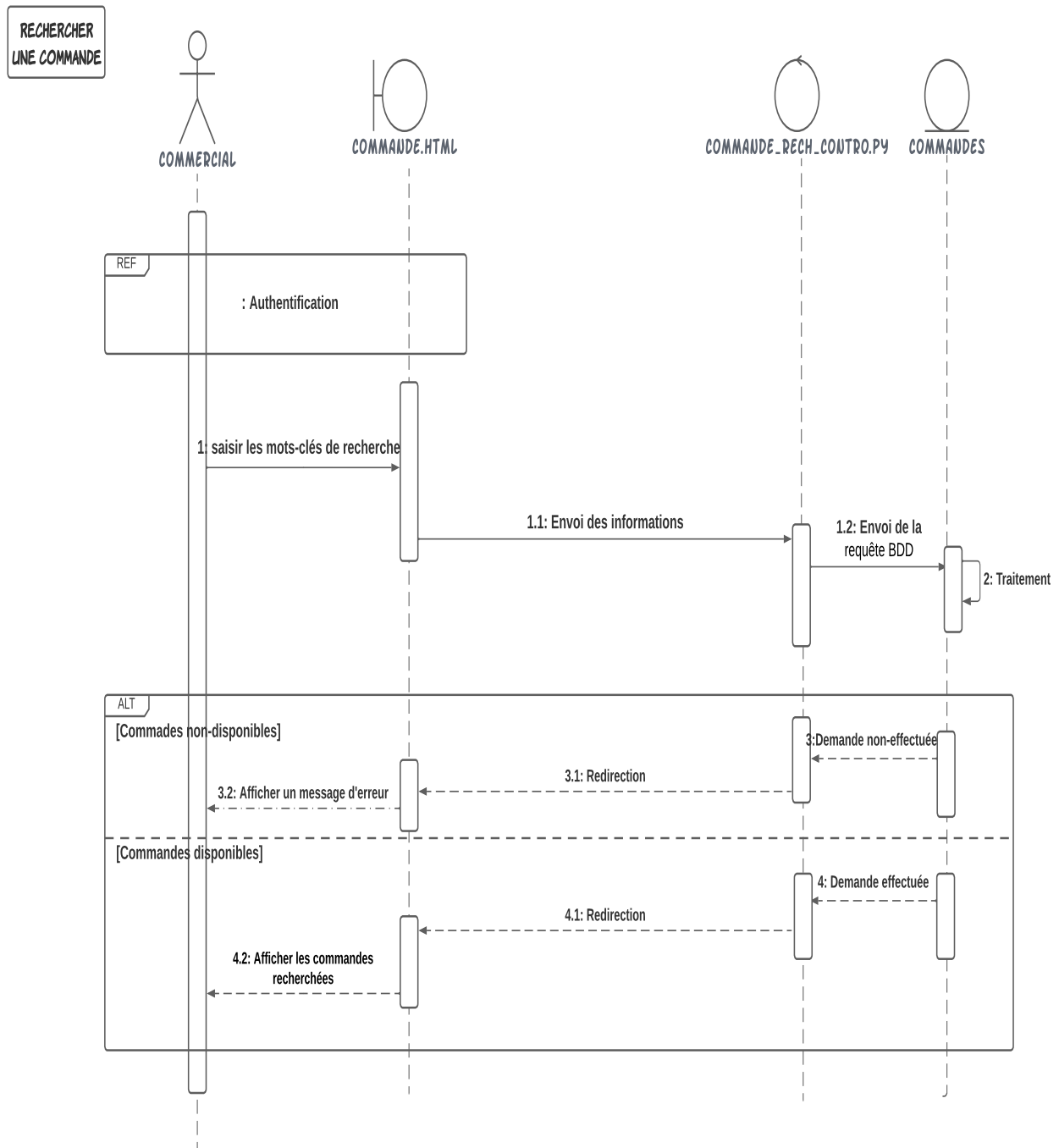


FIGURE 3.5 – Diagramme de séquence détaillé du cas d'utilisation «Rechercher une commande»

3.2.6 Diagramme de séquence détaillé du cas d'utilisation «Planifier une commande»

La figure 3.6 représente le diagramme de séquence détaillé du cas d'utilisation «Planifier une commande» retenu à la fin de la conception.

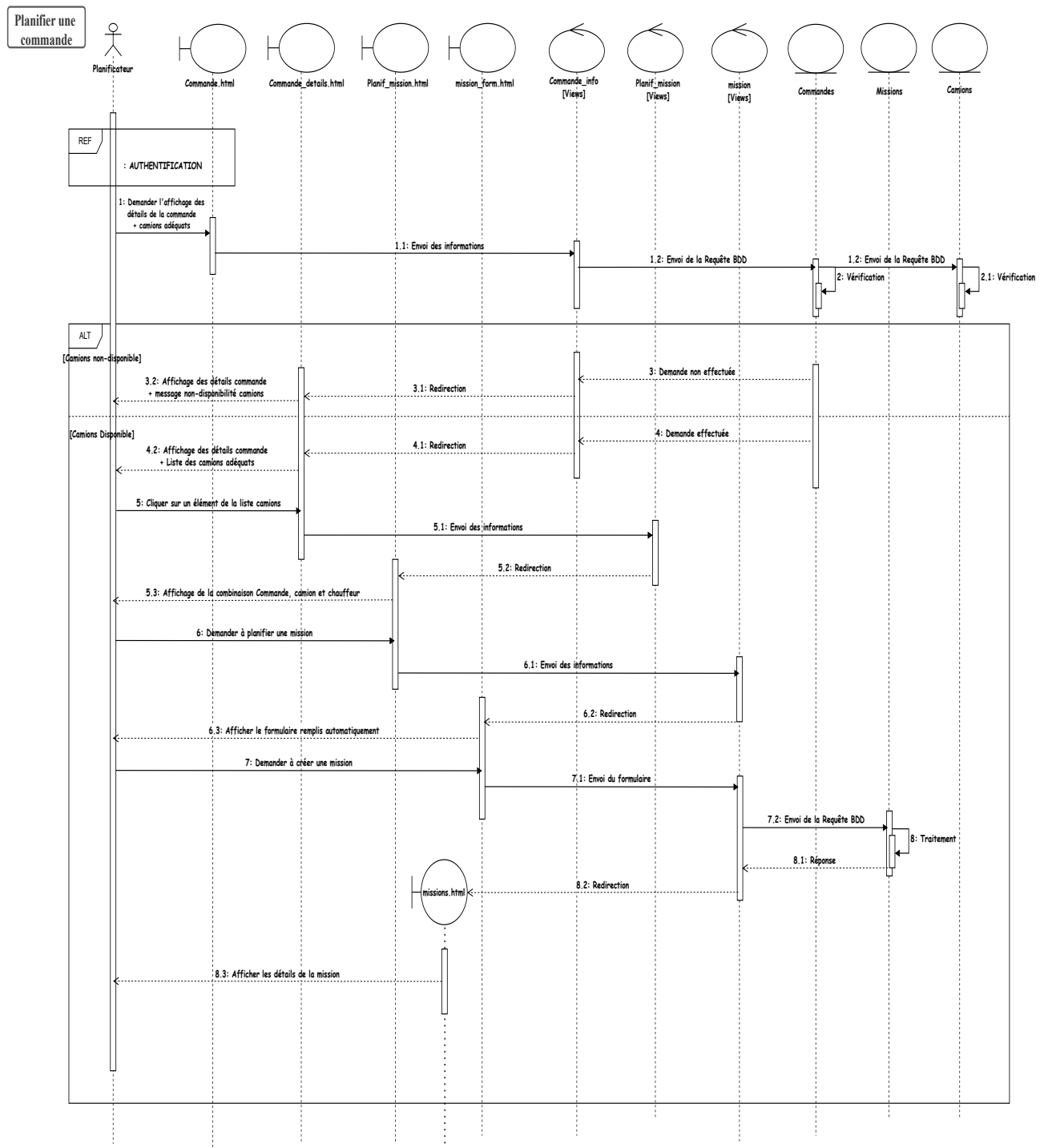


FIGURE 3.6 – Diagramme de séquence détaillé du cas d'utilisation «Planifier une commande»

3.2.7 Diagramme de séquence détaillé du cas d'utilisation «Gérer une mission»

La figure 3.7 représente le diagramme de séquence détaillé du cas d'utilisation «Gérer une mission» retenu à la fin de la conception.

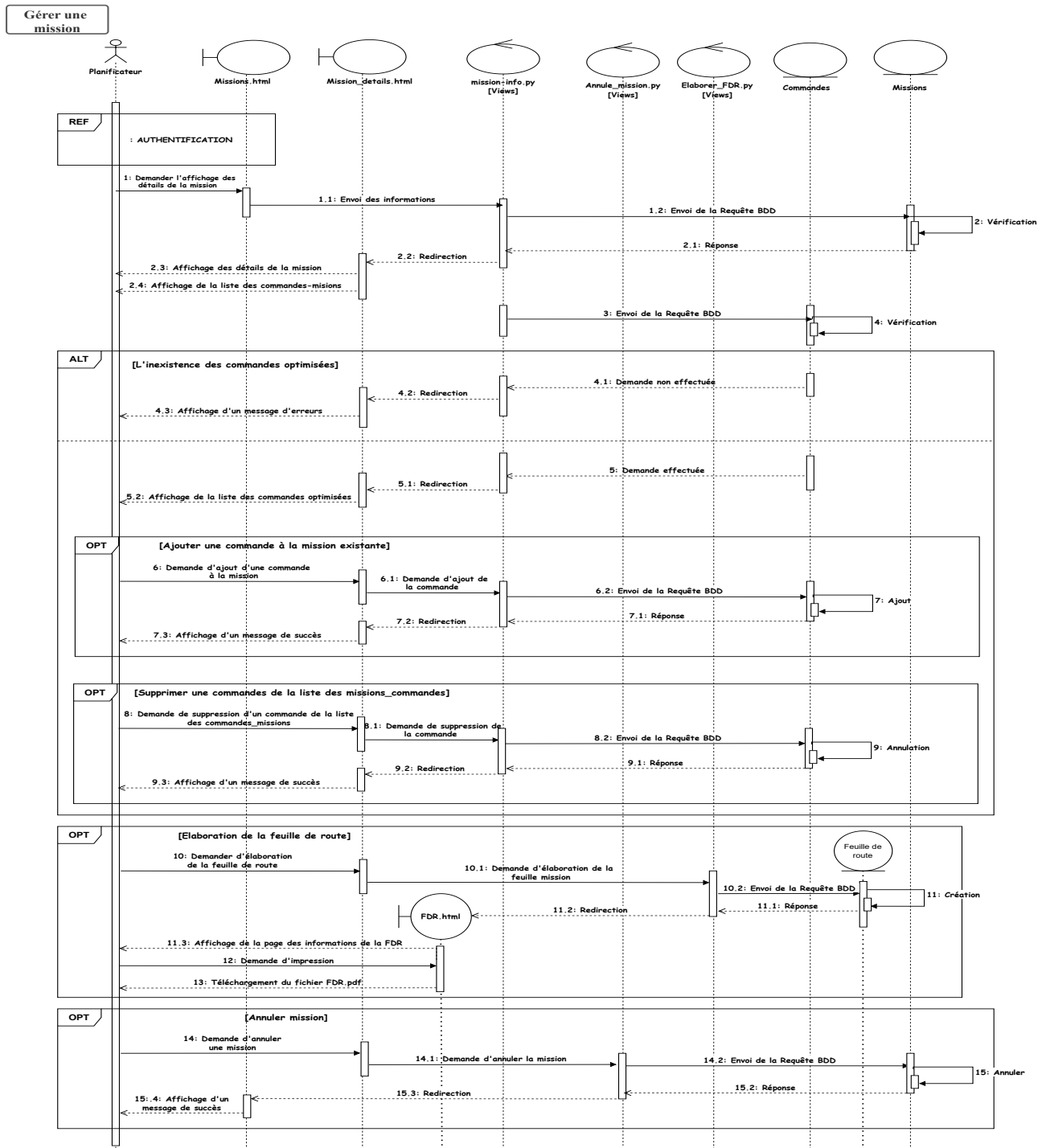


FIGURE 3.7 – Diagramme de séquence détaillé du cas d'utilisation «Gérer une mission»

3.3 Diagramme de classe

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet. Il s'agit d'une vue statique du fait qu'on ne tient pas compte du facteur temporel dans le comportement du système.

Le diagramme de classe montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

L'intérêt du diagramme de classe est de modéliser les entités du système d'information. Il permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont structurées (regroupées dans des classes)[21].

La figure 3.8 représente le diagramme de classe retenu à la fin de la conception.

3.3.1 Le dictionnaire de données

Le dictionnaire de données permet de recenser les informations nécessaires. Il précise le nom de chaque champ, le libellé des données, le type et la dimension utilisées. on va vous présenter l'entité « Commande » accompagnée de ses données.

Voir le tableau 3.1.

1. Informations sur l'entité « Commande »

Nom du champs	Libellé de la propriété	Type	Dimension
id_commande	Numéro d'identifiant de la commande	Entier	AUTO_INCREMENTS
date_commande	date de création de la commande	datetime	/
tarif_commande	tarif de la commande	Nombre réel	10
profit_commande	profit de la commande	Nombre réel	10
Created_by	Entier qui désigne l'utilisateur qui est à l'origine de la création de la commande	Entier	100

TABLE 3.1 – Description des données liées aux commandes.

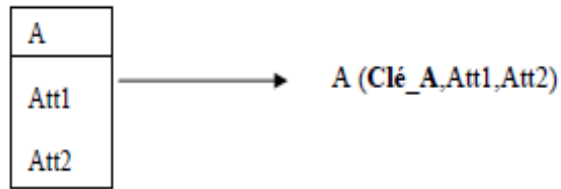
3.4 Modèle relationnel de données

Le modèle relationnel de données est une méthode pour organiser et structurer les informations dans une base de données. Il repose sur le concept de relations, qui sont des tables composées de lignes et de colonnes. Chaque relation représente une entité ou un ensemble d'entités dans le domaine d'application.

3.4.1 Règles de passage du diagramme de classe vers le modèle relationnel

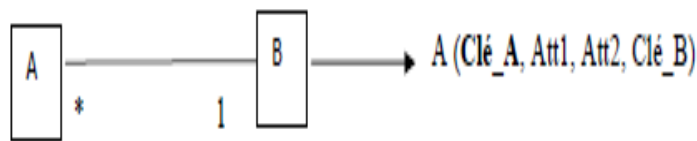
Un schéma relationnel est un ensemble de relations en troisième forme normale. Ce schéma peut être obtenu d'un schéma conceptuel tel qu'un diagramme de classes. Pour cela il faut :

1. Traduire chaque classe par une relation à laquelle on rajoutera une clé.

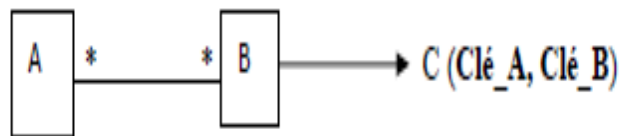


2. Traduire les associations :

- Celles qui ont un bout avec une multiplicité maximum inférieure ou égale à 1.

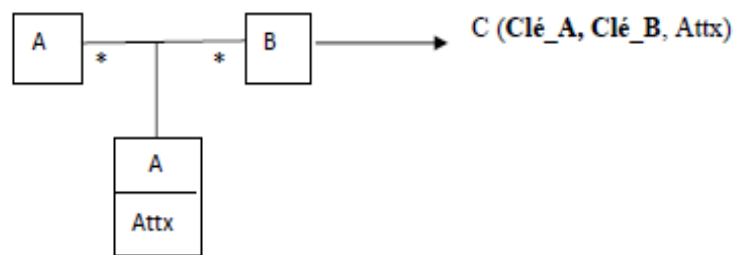


- Celles qui ont les bouts avec une multiplicité maximum *.



Nous ajoutons une relation C ayant une clé composée des clés de A et de B.

- Classes associations.

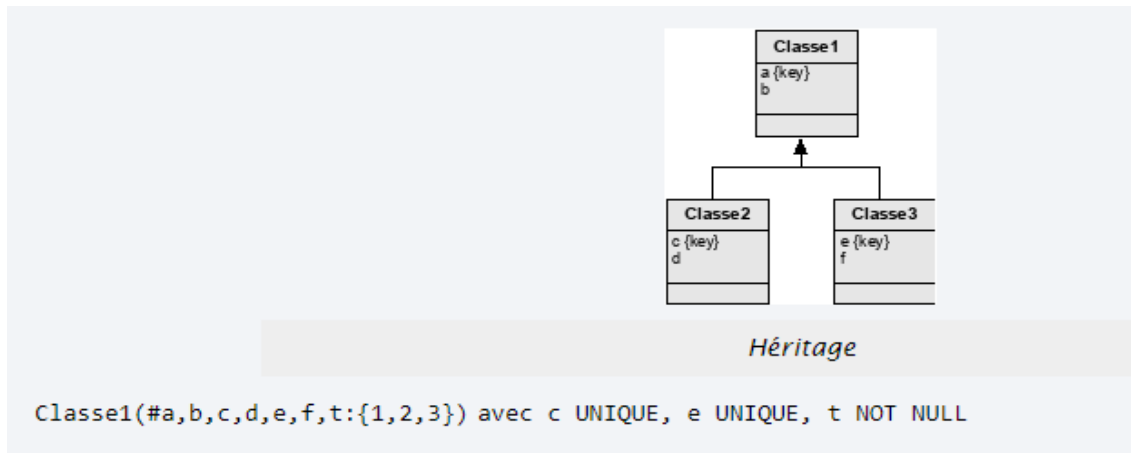


Nous ajoutons une relation C ayant une clé composée des clés A et B, et pour Attributs ceux de la classe association.

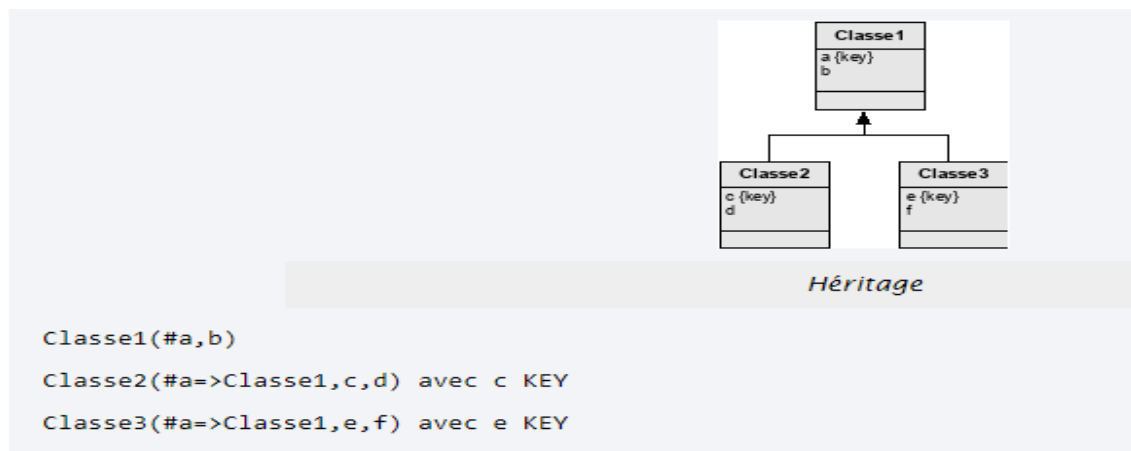
3. Traduction des classes d'héritage

La traduction de la relation d'héritage se fait en propageant la clé de la Classe mère vers les classes filles.

- Seule la classe mère est représentée par une relation (ses classes filles ne sont pas représentées par des relations), Tous les attributs de chaque classe fille sont réintégrés au niveau de la classe mère, La clé primaire de la classe mère est utilisée pour identifier la relation.



- La clé primaire de la classe mère est utilisée pour identifier chacune de ses classes filles : cette clé étant pour chaque classe fille à la fois la clé primaire et une clé étrangère vers la classe mère.



[22] [23]

3.4.2 Schéma Relationnel de Notre Base de Données

Notre schéma relationnel est comme suit :

- Chauffeur (id_chauffeur, nom_chauffeur, prenom_chauffeur, numero_tel, identifiant_permis, date_naissance)
- TypeCamion (idtc, des_tc)
- Roue (idtr, des_tr, capacite)

- Camions (id_camion, num_carte_grise, id_chauffeur_cam#, id_tca#, id_tro#, id_ha#, dispo)
- Habillage (idh, des_h)
- Client (id_client, nom_client, created_by)
- ClientHabillage (id_asso_client# , id_asso_habillage#)
- Ville (id_ville, nom_ville)
- Trajet (id_trajet, id_ville_dep#, id_ville_arr#, distance)
- Mission (id_mission, nom_mission, date_mission, id_dep_mis#, id_arr_mis#, id_camion#, mission_termine, created_by)
- Commande (id_commande, date_c, tarif, profit, nature_dest, id_tc#, id_cl#, ville_dep#, ville_arr#, id_mis#, created_by)
- auth_user (id, password, last_login, is_superuser, username, first_name, last_name, email, is_staff, is_active, date_joined)
- auth_group (id, name)
- auth_user_groups (user_id#, group_id#)
- auth_permission (id, name, content_type_id#, codename)
- auth_user_user_permissions (user_id#, permission_id#)
- auth_group_permissions (group_id# , permission_id#)
- django_content_type (id, app_label, model)

3.5 Conclusion

Tout au long de ce chapitre, nous avons détaillé la conception de notre application à travers les diagrammes de séquences détaillés ainsi que le diagramme de classe associé, afin que la phase de réalisation et la mise en place de l'application soit plus souple et plus aisée. Le chapitre suivant mettra en évidence, le fruit de ce passage et les différents résultats du développement de l'application demandée.

Chapitre 4

Réalisation de l'application

4.1 Introduction

Ce chapitre est consacré à la présentation de l'environnement logiciel utilisé pour le développement de la solution proposée, nous expliquerons éventuellement nos choix techniques relatifs aux langages de programmation et aux outils utilisés. Nous donnons ensuite une description des résultats obtenus par des imprimés écrans.

4.2 Présentation de notre application « TransPlanify »

Nous avons baptisé notre application *TransPlanify*. Elle est destinée à l'entreprise Sarl-BEJAIA LOGISTIQUE en Algérie, entreprise au service de ses clients depuis 2008.

TransPlanify est une application web de gestion et de planification des flux logistiques, doté d'un système d'optimisation intelligent pour une planification optimisée qui traite toute les possibilités pour permettre aux clients de l'entreprise une satisfaction, un gain de temps, d'argent pour l'entreprise et bien sur permettre la réduction de ses coûts et pouvoir augmenter son chiffre d'affaire.

4.3 Environnement du développement de l'application

Nous avons développé une interface graphique simple et pratique pour que Les utilisateurs de BL ne trouvent pas de difficultés pour Manipuler leurs données, l'environnement de développement utilisé :

L'outil de conception

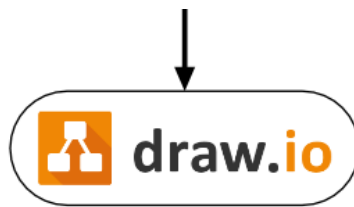
Pour la conception nous avons utilisé :

- **Draw.io**

Draw.io est une application de création de diagrammes en ligne gratuite et open source. Il permet de créer facilement des diagrammes de flux, des organigrammes, des diagrammes

UML, des cartes mentales, des diagrammes de réseaux, des diagrammes de base de données et bien plus encore. Principales fonctionnalités :

1. **Intégration avec d'autres plateformes** : Draw.io peut être intégré à des plateformes populaires telles que Google Drive, Atlassian Confluence, Jira et GitHub, ce qui facilite l'importation et l'exportation de diagrammes.
2. **Édition collaborative** : Draw.io permet aux utilisateurs de collaborer en temps réel sur des diagrammes avec des collègues, des clients et des partenaires, ce qui facilite le travail en équipe.
3. **Bibliothèques de formes personnalisables** : Draw.io offre une large gamme de formes prédéfinies pour les diagrammes, et les utilisateurs peuvent également créer leurs propres formes personnalisées.
4. **Gestion de la version** : Draw.io offre une fonction de gestion de la version qui permet de suivre les modifications apportées aux diagrammes et de revenir à des versions antérieures si nécessaire[24].



□ Visual Studio Code (VS Code)

Éditeur de code open source développé par Microsoft. Il est conçu pour les développeurs qui travaillent sur des projets de toutes tailles et de tous types, des projets Web aux applications mobiles et de bureau.

VS Code est disponible pour Windows, macOS et Linux et offre de nombreuses fonctionnalités pour améliorer la productivité des développeurs. Il dispose d'un environnement de développement intégré (IDE) léger avec une interface utilisateur intuitive et personnalisable, ainsi que d'une large gamme d'extensions pour ajouter des fonctionnalités supplémentaires.

Les fonctionnalités de VS Code incluent :

1. La coloration syntaxique pour une grande variété de langages de programmation.
2. La mise en évidence des erreurs de syntaxe et des avertissements.
3. L'auto-complétion du code.
4. L'intégration avec Git et d'autres systèmes de contrôle de version.
5. Le débogage des applications.
6. La gestion des versions intégrée.
7. L'intégration avec des outils de test et de déploiement[25].

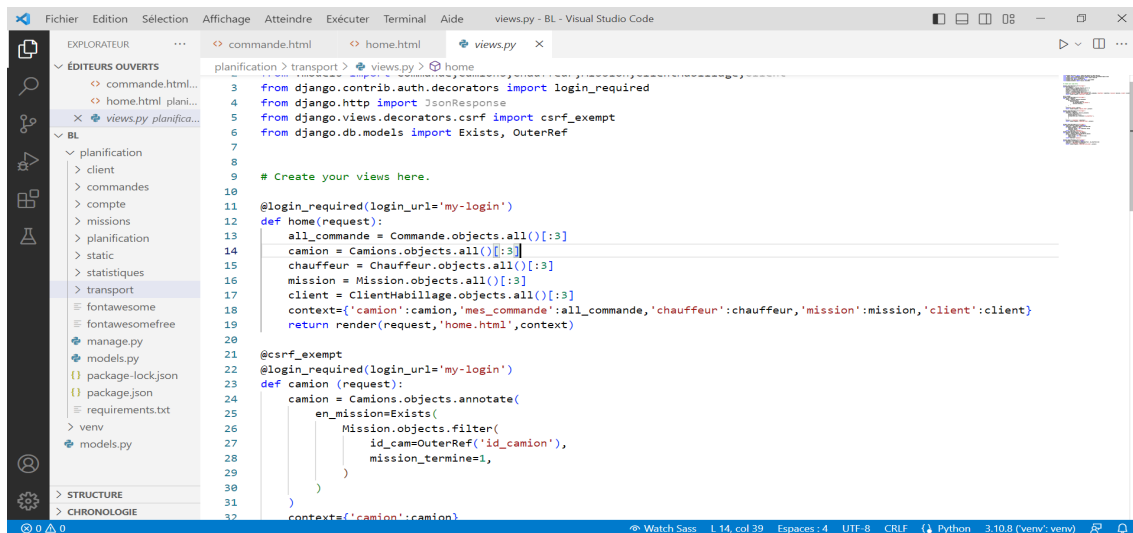


FIGURE 4.1 – Interface de Visual Studio Code

□ Développement Coté Front-end

- **HTML**

(HyperText MarkupLanguage)

C'est le langage universel utilisé sur les pages Web lisibles par tous les Navigateurs Web (Internet Explorer, Google Chrome, Mozilla, etc...). Ce langage fonctionne suivant l'assemblage et la combinaison de balises permettant de structurer et donner l'apparence voulue aux données textes, images et multimédias suivant la mise en page voulue[26].

- **CSS**

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML. Bien que l'HTML puisse être mis en forme à l'aide de balises prévues à cet effet, de nos jours il est plus judicieux d'utiliser le CSS et de n'utiliser le XHTML que pour le contenu. L'avantage de l'utilisation d'un fichier CSS pour la mise en forme d'un site réside dans la possibilité de modifier tous les titres du site en une seule fois en modifiant une seule partie du fichier CSS. Sans ce fichier CSS, il serait nécessaire de modifier chaque titre de chaque page du site (difficilement envisageable pour les énormes sites de plusieurs milliers de pages)[26].

- **JS**

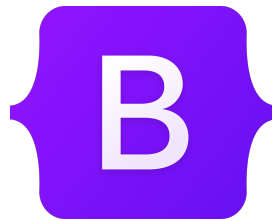
C'est un langage de script incorporé dans un document HTML. C'est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur Web. Il permet d'effectuer des contrôles de saisie pour valider les champs d'un formulaire, d'ouvrir ou fermer des nouvelles fenêtres ou encore de gérer des éléments graphiques[27].



- **Bootstrap**

Bootstrap est un framework de développement front-end gratuit et open source pour la création de sites web et d'applications web. Conçu pour permettre le développement de sites web adaptatifs pour les appareils mobiles en premier, Bootstrap fournit une collection de syntaxes pour les conceptions de modèles.

En tant que framework, Bootstrap comprend les bases du développement web adaptatif, de sorte que les développeurs n'ont besoin que d'insérer le code dans un système de grille prédéfini. Le framework Bootstrap est construit sur le langage de balisage hypertexte (HTML), les feuilles de style en cascade (CSS) et JavaScript. Les développeurs web utilisant Bootstrap peuvent construire des sites web beaucoup plus rapidement sans passer du temps à se soucier des commandes et des fonctions de base[28].



Développement coté back-end

- **Le Framework Django**

Django est un framework python open-source consacré au développement web 2.0 . Les concepteurs de Django lui ont attribué le slogan suivant : " Le framework web pour les perfectionnistes sous pression ". Il est donc clairement orienté pour les développeurs ayant comme besoin de produire un projet solide rapidement et sans surprise.

Comme il est toujours compliqué de partir de rien, meme avec notre expérience dans le langage python, Django nous propose une base de projet solide. Django est donc une belle boîte à outils qui aide et oriente le développement dans la construction de ses projets.

Pour la petite histoire Django a vu le jour en 2003 et a été publié sous licence BSD(licences open-source permet une grande liberté d'utilisation, de modification et de redistribution du code source) en juillet 2005[29].

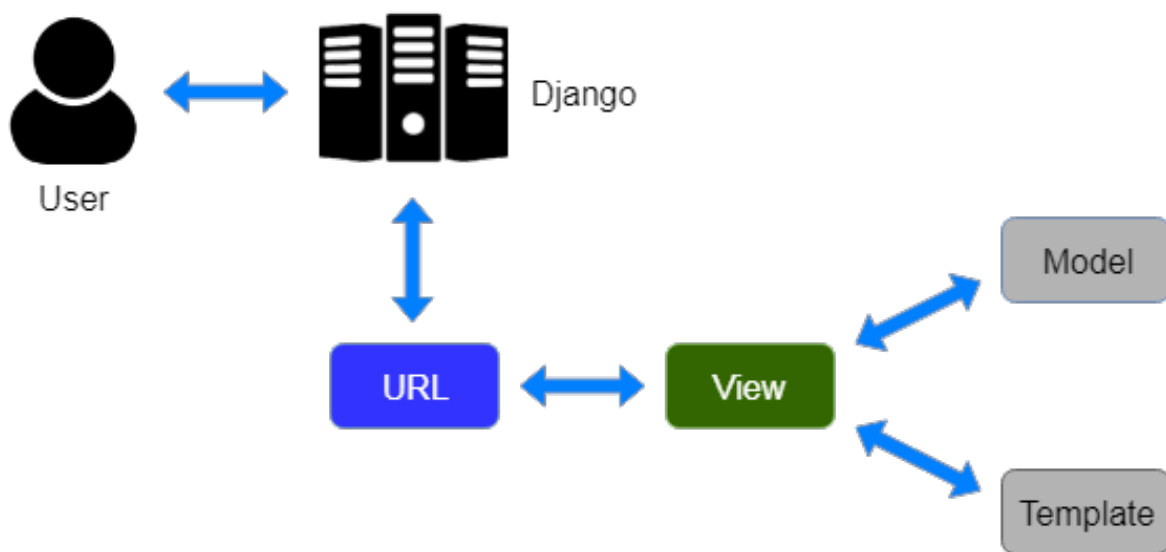
django

Ses fonctionnalités le rendent largement utilisé par les développeurs Web, nous en citons certaines :

- ORM : Django propose une couche d'abstraction de base de données appelée ORM (Object-Relational Mapping) qui permet aux développeurs de travailler avec des bases de données relationnelles de manière plus intuitive en utilisant des objets Python.
- Gestion des URL : Django facilite la gestion des URL en utilisant un système de routage qui permet de mapper des URL à des vues.
- Templates : Django offre un moteur de templates puissant qui permet de créer des vues de manière modulaire et réutilisable.
- Sécurité : Django est conçu pour être sécurisé par défaut, avec des fonctionnalités telles que la protection contre les attaques CSRF (Cross-Site Request Forgery), le hachage de mot de passe sécurisé, la prévention de l'injection SQL, etc[30].

- **Architecture de développement de notre projet**

Nous rappelons que Django s'inspire du modèle MVC (disons plutôt MVT), c'est-à-dire que la structure du framework sépare les données (models) qui sont séparées des traitements (controller) qui sont eux-mêmes séparés de la vue (view / template). ça nous oblige à bien coder, une structure doit être respectée et cela ne peut être que profitable au travail collaboratif ou simplement la cohérence et la communication entre différents projets.



- **L'implémentation d'un projet Django en suivant le modèle MVT**

Le modèle MVT est implémenté dans un projet Django, en utilisant les concepts (ou

les composants) suivants :

1. **Les modèles** : Les modèles sont définis dans le fichier `models.py` de l'application Django. Chaque modèle est une classe Python qui hérite de la classe `Model` fournie par Django. Les champs de chaque modèle sont définis en tant qu'attributs de la classe, en utilisant des classes de champs prédéfinis fournies par Django, telles que `CharField`, `IntegerField`, etc. Les relations entre les modèles sont également définies en utilisant des champs de relation fournis par Django, tels que `ForeignKey`, `ManyToManyField`, etc.
2. **Les vues** : Les vues sont définies dans le fichier `views.py` de l'application Django. Chaque vue est une fonction Python qui prend une requête HTTP en entrée et renvoie une réponse HTTP en sortie. Les vues peuvent récupérer les données nécessaires depuis la base de données en utilisant les modèles, puis les traiter et les transmettre aux templates pour la présentation visuelle. Django fournit également un système de routage pour associer les vues aux URL correspondantes.
3. **Les templates** : Les templates sont définis dans le répertoire `templates` de l'application Django. Les templates sont des fichiers HTML qui contiennent du code HTML et des balises spéciales pour insérer les données dynamiques provenant des vues. Les balises spéciales sont encadrées par des doubles accolades et peuvent contenir des noms de variables Python, des expressions, des filtres et d'autres balises spéciales pour contrôler la logique d'affichage.
4. **Les URL** : Les URL sont définies dans le fichier `urls.py` de l'application Django. Ce fichier contient une liste d'URL qui sont associées à des vues spécifiques. Django utilise des expressions régulières pour définir des URL plus complexes avec des paramètres dynamiques.
5. **Le déploiement** : Django peut être déployé sur différents serveurs web tels que Apache, Nginx ou Gunicorn. Django fournit également son propre serveur de développement intégré, qui est utile pour les tests locaux[31].

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

- **Django development server (runserver)**

Le serveur de développement Django (ou "Django development server") est un serveur Web léger inclus avec le framework Django. Il est destiné à être utilisé uniquement pour le développement et les tests locaux de l'application Django.

Voici quelques caractéristiques et limitations importantes du serveur de développement Django :

1. Il est conçu pour être facile à utiliser et à configurer pour les développeurs qui travaillent localement sur une application Django.
2. Il prend en charge la plupart des fonctionnalités de base d'un serveur Web, telles que l'analyse des requêtes HTTP et la gestion des réponses.

3. Il prend en charge le rechargement automatique de l'application lorsque des modifications sont apportées aux fichiers de code source.
4. Il est écrit en Python et s'exécute dans le même processus que l'application Django.
5. Il n'est pas conçu pour être utilisé en production, car il n'est pas sécurisé et n'est pas optimisé pour gérer des charges de trafic élevées[29].

- **WampServeur**

WampServer est une plateforme de développement web qui permet de créer et de tester des applications web sur un environnement local. Il s'agit d'un logiciel libre et gratuit qui regroupe plusieurs outils tels que le serveur Apache, le langage de programmation PHP et le système de gestion de bases de données MySQL.



WampServer est utilisé par de nombreux développeurs web pour créer et tester des applications web sur leur propre ordinateur, avant de les déployer sur un serveur en ligne. L'avantage de WampServer est qu'il permet de simuler un environnement de production complet sur un ordinateur de développement, ce qui facilite le développement, le test et le débogage des applications[32].

- **MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR) open source. Il est largement utilisé pour stocker, organiser et récupérer des données pour des applications Web, des applications mobiles et des logiciels d'entreprise.

MySQL a été développé par Oracle Corporation et est disponible en deux éditions : la version communautaire gratuite et la version entreprise payante. Il est compatible avec de nombreux systèmes d'exploitation, notamment Windows, macOS, Linux et Unix.

MySQL donne la priorité à l'évolutivité et aux performances



Nous avons optés pour ce SGBD car nous souhaitons maintenir un haut niveau d'intégrité et de personnalisation des données de l'entreprise BL de plus de sa fiabilité, la robustesse de ses fonctionnalités, et parce qu'il fournit des solutions toujours performantes et innovantes[33].

□ GIT

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux. Git est un système de contrôle de version distribué et open source.



- Système de contrôle : cela signifie essentiellement que Git est un outil de suivi de contenu. Git peut donc être utilisé pour stocker du contenu. Il est principalement utilisé pour stocker du code en raison des autres fonctionnalités qu'il fournit
- Système de contrôle de version : Le code qui est stocké dans Git continue de changer à fur et à mesure que l'on en ajoute. De nombreux développeurs peuvent également ajouter du code en parallèle. Ainsi, le système de contrôle de version aide à gérer cela en conservant un historique des changements qui se sont produits.
- Système de contrôle de version distribué : Git possède un référentiel distant qui est stocké sur un serveur et un référentiel local qui est stocké dans l'ordinateur de chaque développeur. Cela signifie que le code n'est pas seulement stocké sur un serveur central, mais la copie complète du code est présente sur tous les ordinateurs des développeurs[34].

4.4 Imprimés écrans de notre application

Cette partie offre une représentation visuelle d'utilisation de notre application.

1. La figure 4.2 représente l'interface d'authentification, l'utilisateur pourra se connecter en introduisant un nom d'utilisateur et un mot de passe.

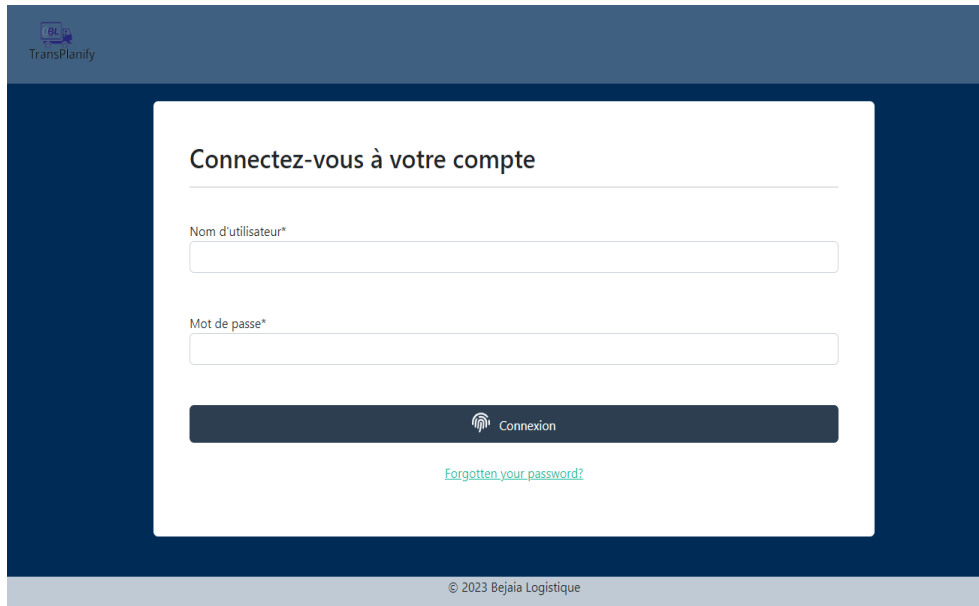


FIGURE 4.2 – Interface d'authentification

2. La figure 4.3 représente l'interface home (accueil) de notre application l'utilisateur après la connexion, il sera rediriger vers son espace de travail selon les accès permis par L'administrateur.



FIGURE 4.3 – La page d'accueil

3. La figure 4.4 montre l'interface Dashboard (tableau de bord) de l'administrateur, il pourra avoir accès aux opérations suivantes :
- La gestion des différentes entités de l'application, la modification des paramètres généraux.
 - Visualiser les statistiques.
 - La gestion des permissions pour les utilisateurs...etc

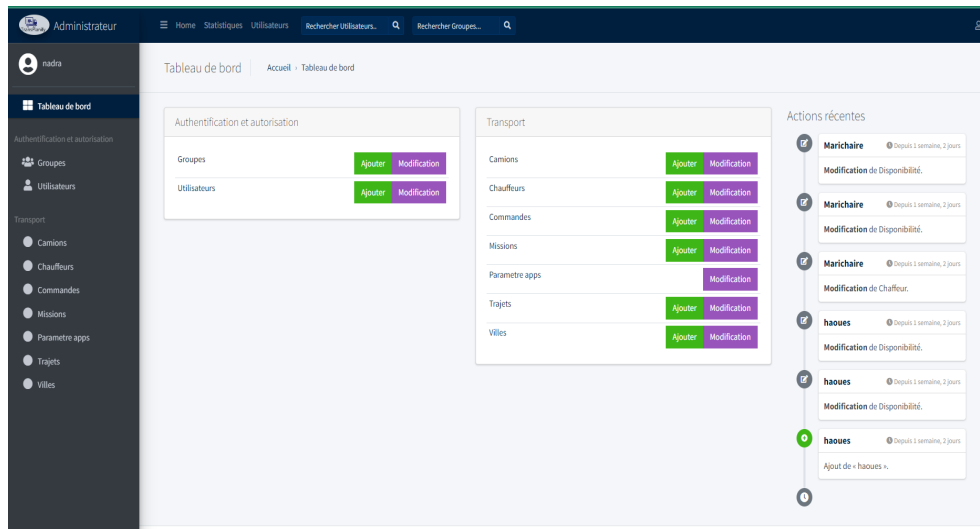


FIGURE 4.4 – Interface tableau de bord de l'administrateur

4. La figure 4.5 représente l'interface de gestion des utilisateurs.

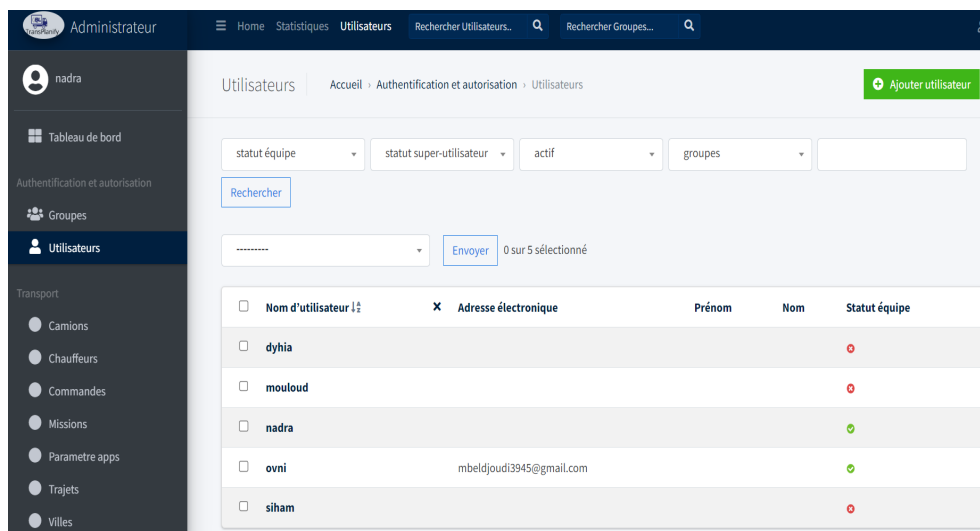


FIGURE 4.5 – Interface liste des utilisateurs

5. La figure 4.6 représente le formulaire d'ajout d'un utilisateur.

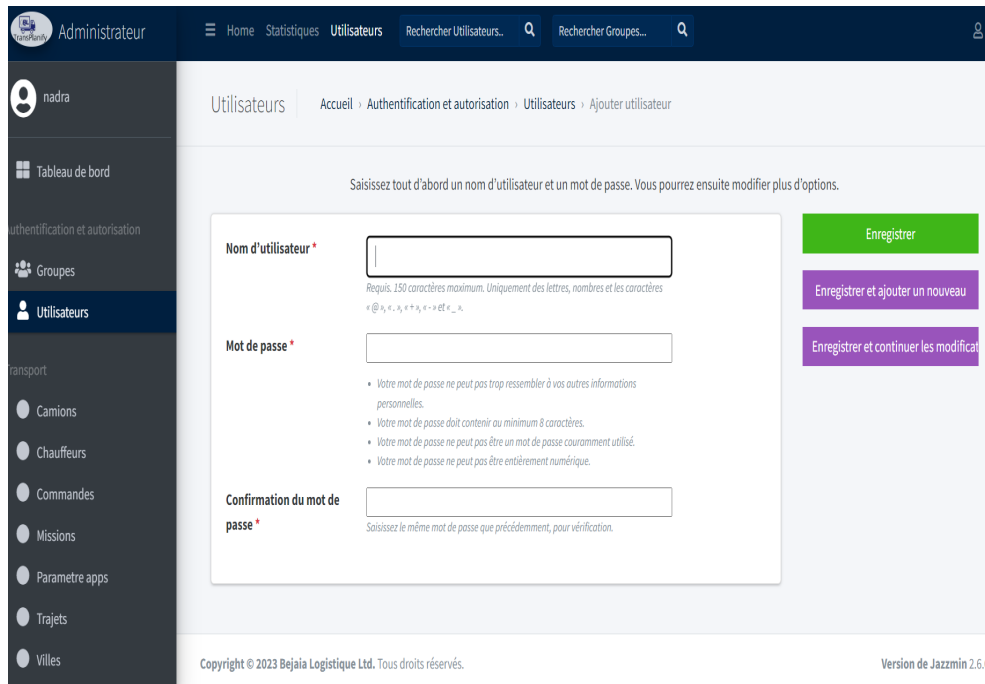


FIGURE 4.6 – Interface formulaire d'ajout d'un utilisateur

6. La figure 4.7 représente l'interface Statistiques, l'administrateur pourra visualiser toutes données liées à la gestion de la logistique.

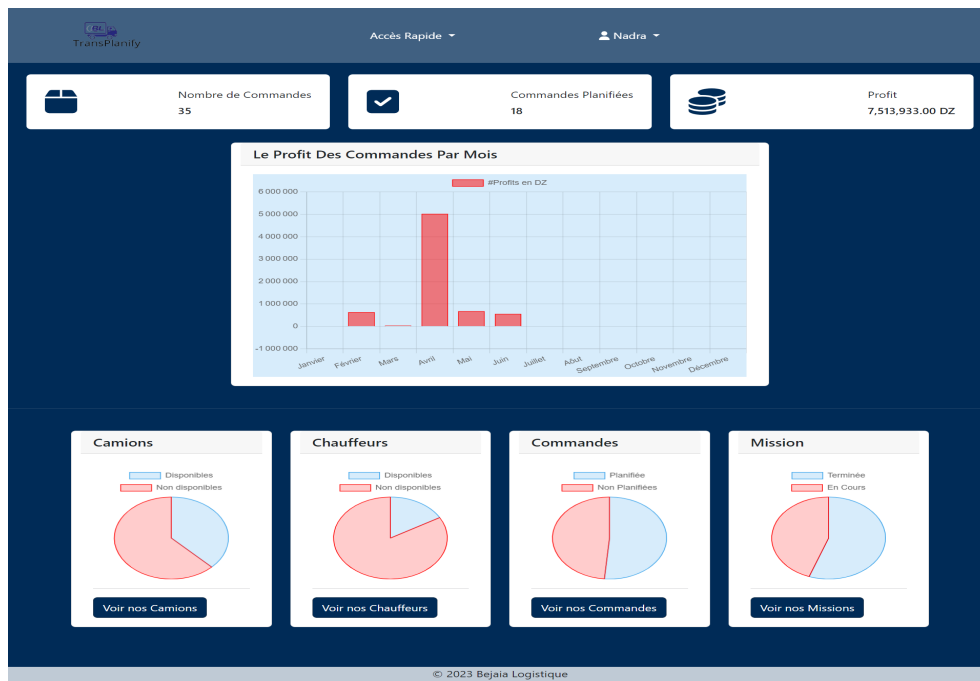


FIGURE 4.7 – Interface statistiques

7. La figure 4.8 représente l'interface consultation des divers commandes, le commercial pourra faire les opérations de gestion sur ces dernières.



FIGURE 4.8 – Interface liste des commandes

8. La figure 4.9 montre le formulaire d'ajout d'une commande.



FIGURE 4.9 – Interface formulaire d'ajout d'une commande

9. La figure 4.10 représente l'interface Détails de la commande, le planificateur pourra choisir les ressources (camions, chauffeurs) adéquats pour la commande voulue planifiée.

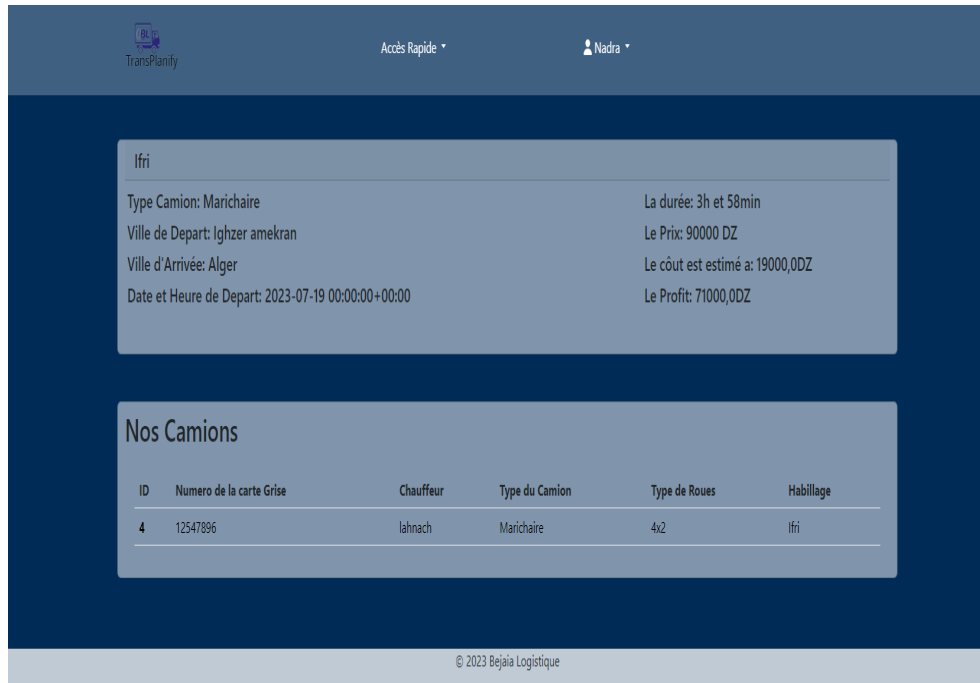


FIGURE 4.10 – Interface détails d'une commande

10. La figure 4.11 montre L'interface regroupement des informations de la combinaison (Com-mande, chauffeur, Camion), le planificateur pourra procéder à la validation de la planification.

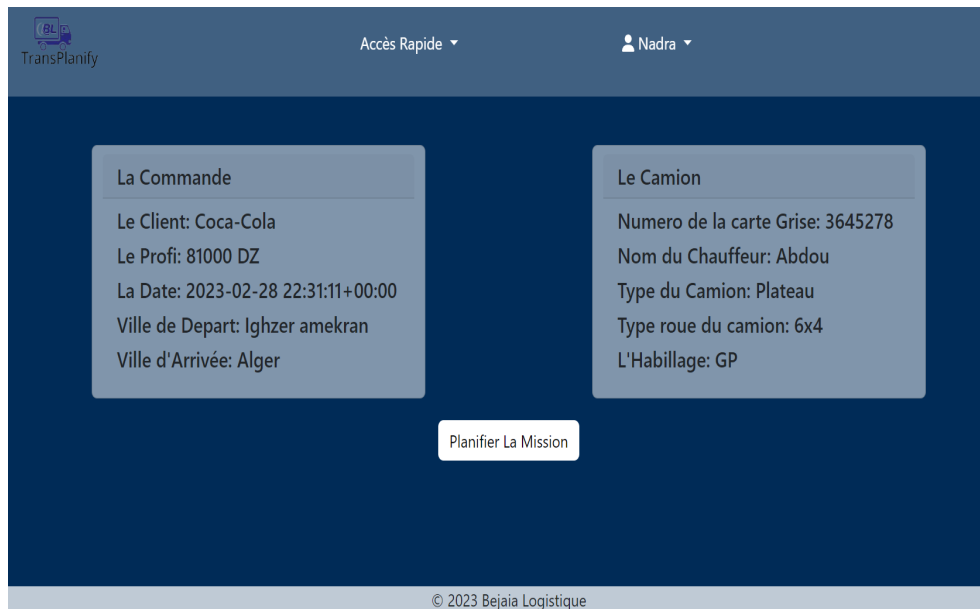


FIGURE 4.11 – Interface planification d'une commande

11. La figure 4.12 montre le formulaire de Création d'une mission préalablement remplis Par l'étape précédente.

Confirmer Votre mission
Enregistrez votre nouvelle mission dans la rubrique qui suit

Nom de la Mission*
mission32

Date de Depart*
24/04/2023 11:25:00

Ville de Depart*
bejaia

Ville d'Arrivée*
Alger

Le Camion
Marichaire

[Planifier la mission](#)

FIGURE 4.12 – Interface création d'une mission

12. Après la création de la mission, le planificateur pourra procéder aux opérations suivantes dans l'interface de gestion de la mission représentée dans la figure 4.13 :

- Visualisation des commandes à rajouter à la mission automatisées par notre système.
- Intégration ou suppression d'une commande dans la fiche mission.
- Annulation d'une mission
- Élaboration de la feuille de route.

Mission32
Profit: 5038066 DZ
Date: 2023-05-03 13:07:00+00:00
Ville de depart: bejaia
Ville d'arrivée: bejaia

Le Camion
Numero de la carte Grise: 123456
Nom du Chauffeur: BELDJOUDI
Type du Camion: Marichaire
Type roue du camion: 4x2
L'Habillage: Ifruit

Les Commandes en mission

ID	date	Client	Profit	Ville de Depart	Ville de Destination	Action
57	3 mai 2023 13:07	KASA	59000	Alger	bejaia	-
32	24 avril 2023 11:25	Coca-Cola	4979066	bejaia	Alger	

Les Commandes a rajouter

ID	date	Client	Profit	Ville de Depart	Ville de Destination	Action
43	6 mai 2023 10:00	ifri	29000	bejaia	Alger	+
55	27 mai 2023 17:58	Coca-Cola	43000	bejaia	Tizi ouazou	+
51	27 mai 2023 15:00	Coca-Cola	59000	bejaia	Alger	+
6	17 juin 2023 22:26	KASA	487007	bejaia	ighzer amekran	+

FIGURE 4.13 – Interface gestion d'une mission

13. Le planificateur peut maintenant élaborer la feuille de route d'une mission, l'interface 4.14 ci-dessous nous montre le regroupement des informations liée à la mission49.

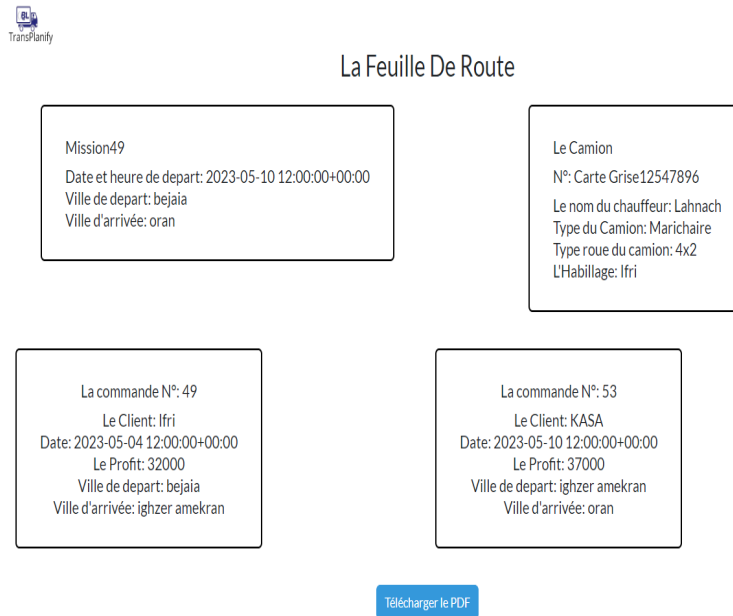


FIGURE 4.14 – Interface élaboration de la feuille de route d'une mission

4.5 Conclusion

Dans ce chapitre, nous avons présenté en détail l'environnement logiciel utilisé pour le développement de notre solution, ainsi que les choix techniques que nous avons effectués en matière de langages de programmation et d'outils. Nous avons également exposé les résultats que nous avons obtenus, en appuyant nos propos avec quelques captures d'écran illustratives. À travers cette présentation, nous avons pu démontrer l'efficacité et la pertinence de notre approche, ainsi que la qualité des résultats obtenus.

Nous sommes également ravis de partager que notre approche a suscité une grande satisfaction au sein de l'organisme d'accueil. Les retours positifs que nous avons reçus soulignent l'adéquation de notre solution avec leurs besoins spécifiques et la valeur ajoutée qu'elle apporte à leurs opérations. La collaboration étroite avec l'organisme d'accueil nous a permis de comprendre leurs attentes et de concevoir une solution sur mesure qui répond à leurs exigences.

En somme, ce chapitre nous a permis de mieux comprendre les étapes nécessaires à la mise en place d'une solution technique, en particulier dans le choix de l'environnement de développement et la sélection des outils appropriés.

Conclusion Générale

En somme, la logistique est un élément essentiel dans le développement international de l'entreprise, qui doit maîtriser ses flux d'information et de marchandise pour répondre aux besoins économiques de différents pays. Le transport de marchandises est un vecteur indispensable de la chaîne logistique, qui assure la liaison entre les différentes étapes du système logistique et est directement lié à la qualité des services. Cependant, pour assurer une performance opérationnelle optimale, la gestion logistique nécessite de nouvelles méthodes et technologies, notamment la réduction des coûts, la maximisation du profit, la traçabilité, la planification et la fiabilité.

Dans ce cadre, notre projet de fin de cycle master a consisté en la conception et la réalisation d'une application web munie d'un système intelligent pour aider à la décision les planificateurs de l'entreprise BL. Grâce à une analyse approfondie des besoins de l'entreprise, nous avons pu cerner les lacunes du service de planification existants et concevoir un système automatisé pour optimiser les gains et réduire les coûts liés à la gestion de l'entreprise. Nous sommes convaincus que notre solution répondra efficacement aux besoins de l'entreprise et permettra d'optimiser la gestion logistique.

En somme, notre projet a permis de mettre en évidence l'importance de la gestion logistique au sein de l'entreprise et la nécessité de recourir à des solutions innovantes pour optimiser les performances opérationnelles.

Ce projet témoigne de l'importance de la logistique et de l'utilisation des technologies pour améliorer la performance de l'entreprise BL dans un contexte de logistique du Transport en constante évolution.

Enfin, Comme de nombreux autres projets de développement logiciel en cours, notre travail est particulièrement en constante évolution. Par conséquent des perspectives d'avenir prometteuses se dessinent. Dans notre cas nous souhaitons :

- Intégrer des algorithmes simples pour accroître l'optimisation vis-à-vis de la planification des Commandes et la vérification des diverses disponibilités.
- Ajout d'un module de géo localisation pour rendre la planification plus fluide.
- Améliorer le design de nos interfaces pour répondre aux critères ergonomiques.
- Application mobile au profit des chauffeurs de l'entreprise pour la prévention des indisponibilités et divers situation sur le chemin.

Bibliographie

- [1] Bejaia logistique. L'histoire de la logistique à travers les âges. <https://www.bejaialogistique.com/presentationmain.>, 04/20/2021.
- [2] VINCENT CRITON. L'histoire de la logistique à travers les âges. <https://logistique-pour-tous.fr/histoire-de-la-logistique/>, 27/08/2012.
- [3] Bensalma et Larbi. Management logistique et la performance de l'entreprise au niveau du groupe le marechal. <https://www.ummtto.dz/dspace/bitstream/handle/ummtto/17143/finale%20%281%20sequence=1isAllowed=y>, 2021.
- [4] supply chain info. Les différents types de flux : logistique interne et logistique externe. <https://www.supplychaininfo.eu/dossier-logistique/quelle-difference-logistique-interne-logistiqueexterne/> : :text=Cette%20logistique%20externe%20d%C3%A9signe%20toutes,'entreprise%20ju 2018.
- [5] Mecalux. Définition et tendances d'optimisation des processus intralogistiques. <https://www.mecalux.fr/blog/logistique-interne-definition.>, 13/09/2019.
- [6] mecalux. Comment optimiser la logistique de production. <https://www.mecalux.fr/blog/logistique-de-production> : :text=Qu'est%20ce%20que%20la,%C3%A0%20 24/01/2020.
- [7] logistique conseil. Logistique industrielle. <http://www.logistiqueconseil.org/Articles/Logistique/Logistique-industrielle.htm.>, 05/2016.
- [8] logistique et transport. Les Différents Types De Logistique. <https://lojistike.blogspot.com/2018/02/les-differents-types-de-logistique.html>, 04/02/2018.
- [9] Samir ; Mazouzi Salem Guerroudj, Rahim ; Bouaissaoui. Le transport routier de marchandises dans la chaine logistique. Mémoire master SEGC Option : logistique des distribution, 2019.
- [10] Kepos Capital Peter Meindl. Supply Chain Management STRATEGY, PLANNING, AND OPERATION. Edition Fifth, 2001.
- [11] faq logistique. Quels gains financiers et RSE attendre de son TMS ? <https://www.faq-logistique.com/TMS.htm>, 2019.
- [12] Alexandre Leuliet. Nouvelles coupes pour le problème de tournées de véhicule avec demandes stochastiques. https://publications.polymtl.ca/1603/1/2014_AlexandreLeuliet.pdf, 2014.
- [13] prasoonsharma. Any R packages to solve Vehicle Routing Problem? <https://www.r-bloggers.com/2010/11/any-r-packages-to-solve-vehicle-routing-problem/>, 09/11/2010.

- [14] L'équipe éditoriale. Tout sur la Méthode Agile XP. <https://www.nutcache.com/fr/blog/lamethode-agile-xp/>, 12/09/2020.
- [15] jean François Le Foll. Developpement agile de logiciels. <https://noti.st/jefflefol/jlZBDg/slides>, 10/2020.
- [16] Marjorie Doucets. Une rétrospective des méthodes agiles. <https://medium.com/@marjoriadoucet/une-rC3A9trospective-des-mC3A9thodes-agilesb379b76a2cde>, 13/09/2015.
- [17] Ed Seidewitz. UML and SysML. <https://www.youtube.com/watch?v=vAHHdnIV8rU>, 09/2018.
- [18] wikiwand. UML (informatique). <https://www.wikiwand.com/fr/UML%28informatique%29>, 12/2017.
- [19] Tutorialspoint. UML Tutorial. https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm, 01/2020.
- [20] Martin Fowler. UML Distilled : A Brief Guide to the Standard Object Modeling Language. Édition Adisson-wesley, 25/09/2003.
- [21] Pascal Roques. les Cahiers du Programmeur UML2 Modéliser une application web. Édition Eyrolles, 2008.
- [22] STÉPHANE CROZAT. Passage UMLRelationnel : classes et associations. <https://stph.scenari-community.org/idl-bd/idl-bd4.pdf>, 02/10/2016.
- [23] Abdelkrim rebai. Passage de diagramme de classe (UML) vers un modèle relationnel. <https://www.youtube.com/watch?v=Oj7geQX8c6A::text=la%20r%C3%A8gle%20de%20base%20c,si%20la>, 10/03/2021.
- [24] David Benson. Bootstrap. <https://desk.draw.io/support/solutions/articles/16000067791-use-the-diagrams-net-editor>, 16/11/2020.
- [25] getCodingKnowledge. Une Introduction à Visual Studio Code pour Débutants. <https://www.youtube.com/watch?v=IpxG-uFq9ac>, 10/06/2018.
- [26] J Duckett. HTML and CSS : Design and Build Websites. Édition Wiley, 2011.
- [27] J Duckett. JavaScript and jQuery : Interactive Front-End Web Development. Édition Wiley, 2014.
- [28] Andrew Zola. Bootstrap. <https://www.techtarget.com/whatis/definition/bootstrap>, 03/08/2022.
- [29] Site officiel de Django. Django en bref. <https://docs.djangoproject.com/>, 07/2005.
- [30] Waldir Pimenta. What is Django ? https://tutorial.djangogirls.org/en/django_start_project/, 26/03/2020.
- [31] Onkar Ruikar. Django Web Framework (Python). <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>, 24/02/2022.
- [32] Romain Bourdon. EXPÉRIMENTER WAMPSEVER. <http://www.wampserver.com/>, 13/02/2006.
- [33] Seyed Tahaghoghi et Hugh Williams. EXPÉRIMENTER WAMPSEVER. Édition OReilly, 28/11/2006.
- [34] Seyed Tahaghoghi et Hugh Williams. Pro Git. Édition Apress, 2014.