

Dédicace

Ce travail est dédié à mon grand père Filali Mohamed, décédé trop tôt. J'espère que, du monde qui est sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'un petit-fils qui a toujours prié pour le salut de son âme. Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde !

Remerciements

*Au terme de ce travail, je tiens à présenter mes remerciements les plus sincères à Monsieur **Mr BOUKHARI KABIL** qui a accepté de m'encadrer et qui m'a fait profiter de ses larges connaissances et ses précieux conseils au cours de mon projet de fin d'études.*

*Je tiens aussi à remercier la société **BON-GEST** de m'avoir permis d'effectuer ce sujet, et je souhaite remercier plus particulièrement le responsable du stage, Monsieur **Mr KHLAYFIYA LAMJED**, qui m'a offert ce stage, et qui a accepté de m'encadrer pendant celui-ci.*

*Je tiens à adresser mes sincères remerciements à tout le personnel de **BON-GEST** qui m'a accueilli chaleureusement durant ces trois mois, contribuant ainsi au bon déroulement de ce stage.*

Il faut souligner que ce travail n'aurait pas pu voir le jour sans les riches connaissances et savoir-faire acquis dans notre honorable faculté "Faculté des Sciences de Monastir".

C'est donc avec une immense fierté, que j'adresse mes remerciements les plus distingués à tous mes professeurs.

Qu'ils trouvent, ici, ainsi que toute personne qui a contribué à l'achèvement de ce projet, directement ou indirectement, l'expression de ma sincère gratitude.

Enfin j'exprime mes remerciements, les plus dévoués, aux membres du jury qui m'honorent en acceptant d'évaluer mon travail.

Table des matières

Table des figures	V
Liste des tableaux	VII
Introduction Générale	1
1 Étude préliminaire	3
1.1 Introduction	4
1.2 Présentation de l'organisme d'accueil	4
1.2.1 Organisation	4
1.2.2 Service	5
1.3 Cadre général du projet	5
1.3.1 Étude et critique de l'existant	5
1.3.2 Problématique	6
1.3.3 Solution proposée et objectifs globaux du projet	7
1.4 Branche Fonctionnelle	7
1.4.1 Besoins fonctionnels	8
1.4.2 Besoins non fonctionnels	9
1.5 Branche technique	10
1.5.1 Architecture logicielle	10
1.5.2 Langage de programmation	12
1.5.3 Serveurs	13
1.6 Conclusion	13
2 Conception	14
2.1 Introduction	15

TABLE DES MATIÈRES

2.2	Méthode de conception	15
2.2.1	Le langage UML	15
2.2.2	Diagrammes d'UML	16
2.2.3	Modèles d'UML utilisés	16
2.3	Conception détaillée	17
2.3.1	Identification des cas d'utilisations	17
2.3.2	Diagramme des classes	23
2.3.3	Diagrammes de séquence système	25
2.4	Conclusion	29
3	Réalisation	30
3.1	Introduction	31
3.2	Environnement et outils de travail	31
3.2.1	Environnement matériel	31
3.2.2	Environnement logiciel	31
3.3	Travail réalisé	33
3.3.1	Page d'authentification	33
3.3.2	Espace Administrateur	33
3.3.3	Espace Magasinier	36
3.3.4	Espace Transporteur	37
3.3.5	Espace Client	39
3.4	Conclusion	40
	Conclusion générale	41
	Bibliographie	43

Table des figures

1.1	Logo Entreprise	4
2.1	Relation entre les acteurs	17
2.2	Diagramme de cas d'utilisation relatif à l'administrateur	18
2.3	Diagramme de cas d'utilisation relatif au magasinier	19
2.4	Diagramme de cas d'utilisation relatif au transporteur	20
2.5	Diagramme de cas d'utilisation relatif au client	21
2.6	Diagramme de cas d'utilisation "Gérer les véhicules"	22
2.7	Diagramme des classes	24
2.8	Diagramme de séquence du scénario "Authentification"	26
2.9	Diagramme de séquence du scénario "Modifier produit"	27
2.10	Diagramme de séquence du scénario "Ajouter Bon de sortie"	28
2.11	Diagramme de séquence du scénario "Passer Commande"	29
3.1	Interface d'authentification	33
3.2	Interface Administrateur	34
3.3	Interface de gestion des utilisateurs	34
3.4	Interface de gestion des factures	35
3.5	Interface d'ajout d'une facture	36
3.6	Interface Magasinier	36
3.7	Interface des produits	37
3.8	Interface Transporteur	37
3.9	Interface d'ajout d'un bon de sortie	38
3.10	Interface d'affectation d'un véhicule	38

TABLE DES FIGURES

3.11 Interface Web	39
3.12 Interface Mobile	40

Liste des tableaux

2.1	Scénario "Authentication"	22
2.2	Scénario "Gérer les véhicules"	23
2.3	Tableau descriptif des classes	25

Introduction Générale

L'information est devenue une capitale qui doit être bien gérée et maîtrisée. A cet égard les entreprises et les administrations nationales, et dans l'objectif de les rendre plus accessible et plus lucratif, elle été obligé d'intégrer la nouvelle technologie au sein de ces organisations.

L'informatique occupe évidemment une grande place dans le domaine commercial. En effet, la gestion commerciale est une tâche capitale qui présente un nombre important de sous tâches. Elle consiste généralement à gérer le stock(les produits entrants et sortants), la gestion des tiers, la gestion des documents(commandes, factures, bons de sortie, bons d'achat) et la gestion des véhicules. En effet, les entreprises cherchent toujours à assurer une bonne gestion commerciale en rendant cette pénible tâche informatisée.

C'est dans ce cadre, que se situe notre projet de fin d'études dont le but est de mettre en place une application Web et Mobile de gestion commerciale pour la société BON-GEST qui vont à leur tour l'utiliser pour une société concerné. Ce logiciel aura pour objectif de simplifier les différentes tâches relatives à la gestion commerciale comme la gestion du stock, des comptes utilisateurs, des documents commerciales des tiers, des véhicules, etc..., qui étaient, jusque là, réalisées soit d'une manière manuelle, soit avec EXCEL.

Nous nous intéressons dans ce rapport à la description de différentes étapes de la réalisation de notre projet. Il est subdivisé en trois principaux chapitres :

Le premier chapitre intitulé "**Étude préliminaire**" nous présentons la société BON-GEST, ensuite nous passerons à l'étude et à la critique de l'existant pour proposer une

solution adéquate. Enfin, nous aborderons la compréhension du contexte du système et cela à travers la définition des besoins fonctionnels, les besoins non fonctionnels et techniques.

Le deuxième chapitre intitulé "**Conception**" nous aborderons la conception détaillée par la présentation de modèle de l'application. Puis, nous détaillons la conception grâce aux diagrammes des cas d'utilisations, des classes et des séquences.

Dans le dernier chapitre intitulé "**Réalisation**", nous présentons l'environnement de développement matériel et logiciel que nous allons adopter. Puis, nous exposerons quelques interfaces des modules réalisés. Nous clôturerons le présent mémoire par une conclusion qui synthétise le travail et dresse les perspectives.

Chapitre 1

Étude préliminaire

1.1 Introduction

Ce premier chapitre définit le contexte général de notre travail, l'étape d'analyse et la spécification des besoins. Dans une première partie nous fournirons une brève présentation de la société BON-GEST. Ensuite, nous décrirons le cadre de notre projet, puis nous allons critiquer l'existant pour dégager leurs insuffisances et proposer les orientations de notre solution future. Après dans la branche fonctionnelle, nous allons définir les besoins fonctionnels et non fonctionnels. Finalement nous clôturons ce chapitre avec la branche technique, d'où seront listés également les principaux besoins.

1.2 Présentation de l'organisme d'accueil

1.2.1 Organisation

BON-GEST est une société de développement informatique basée à Monastir, spécialisée dans la conception et le développement des sites Web et d'applications mobiles Android, multimédia. Elle assure, à travers une équipe de professionnels spécialisés, la création de sites web adaptés à la demande de sa clientèle et faciles à utiliser. Les collaborateurs sont pour la plupart des jeunes informaticiens disciplinés en termes de relation client-service et éprouvant une énergie et ambition illimitées, dotés d'une large expérience dans le domaine de développement web.

Les objectifs de BON-GEST consistent à offrir de meilleurs services de gestion et de sécurité visant à augmenter la productivité et à apporter des solutions innovantes et performantes.

La figure 1.1 présente le logo de la Société BON-GEST.



FIGURE 1.1 – Logo Entreprise

1.2.2 Service

BON-GEST propose un large éventail de services et de solutions :

- création des sites web.
- développement des applications mobiles.
- développements des applications spécifiques.
- création graphique.
- conseil et Accompagnement.
- développer et maintenir des applications qui ont pour fonction d'apporter des solutions innovantes aux clients de l'entreprise dans le contexte de la dématérialisation des documents.

1.3 Cadre général du projet

Le présent projet intitulé "Conception et Développement d'une application Web et Mobile de gestion commerciale " est réalisé dans le cadre de la préparation du Projet de Fin d'études présenté en vue de l'obtention du licence fondamentale en informatique à la Faculté des sciences de Monastir pour l'année universitaire 2018/2019. Il a été réalisé au sein de la société BON-GEST.

1.3.1 Étude et critique de l'existant

Cette application sera exploitée par un entrepôt à Monastir. Les tâches relatives à la gestion commerciale sont réalisées avec Excel. Chaque entrée des produits en stock ou sorties nécessite une saisie des données relatives à la tâche réalisée. Concernant la facturation se fait par des factures pré-imprimées, de même pour les bons de sortie des transporteurs et les commandes des clients .

Cette technique de gestion reflète plusieurs inconvénients dont les plus importants sont :

- Difficulté de partage de l'information : l'une des limites les plus évidente d'Excel est la difficulté à partager le contenu avec les autres ; même en lecture sur un réseau, il faut attendre que l'information soit actualisée par le propriétaire de la feuille pour qu'elle parvienne aux autres.
- Fiabilité des données, et traçabilité : le partage d'un fichier Excel entre les membres d'une équipe implique généralement la création de plusieurs fichiers parallèles, avec des informations mises à jour différemment des 2 côtés, chaque utilisateur ayant la conviction que sa version est bonne. Excel n'offrant aucune possibilité de tracer les données, impossible de savoir qui a utilisé le mauvais fichier.
- Risque de perte ou de dégradation des documents : les factures, les bons de sortie ou les commandes peuvent devenir illisibles.
- Problèmes de portabilité :
 1. Difficultés relatives a la gestion des transporteurs, des véhicules et des bons de sortie.
 2. il n'existe pas des statistiques qui nous aide à avoir une information rapide et claire comme les bons de sortie faite par jours ,mois ,année ou les produits en rupture de stock ou bien les états des commandes etc..
 3. Une grosse perte de temps : chaque tâche relative à la gestion de stock rend la circulation des informations très lente. En effet, certains ordre venant de transporteur ne seront pas transmet a temps réel au magasinier pour les traiter comme la gestion d'une bon de sortie que le transporteur le demande et le Magasinier doit le préparer.
- Une absence de sécurité : il est certes possible d'ajouter un code dans le tableur Excel, mais il est difficile de sécuriser les données qui circulent.

1.3.2 Problématique

La gestion commerciale avec Excel est difficile compte tenu de la diversité des tâches à accomplir et le nombre important des produits entrants et sortants. En fait, parmi ces

tâches, la tâche de facturation qui nécessite beaucoup d'attention, et de temps pour une vérification manuelle. Aussi, l'absence d'une base de données rendent quasiment impossible l'établissement de statistiques fiables.

Donc pour garantir l'automatisation des tâches, une société peut mettre en place une application qui lui permettra de gérer la pression commerciale.

1.3.3 Solution proposée et objectifs globaux du projet

En constatant les insuffisances du système actuel et afin de pallier aux défaillances, ce projet de gestion commerciale doit répondre à plusieurs objectifs qui sont :

- Assurer la simplicité du logiciel tout en gardant la sécurité et la confidentialité des différentes données saisies.
- Concevoir des interfaces claires et simples à manipuler.
- La rapidité, la fiabilité et la facilité des traitements.
- Optimiser au maximum le temps d'accès et de manipulation des données.
- Assurer les différents services de gestion.

La solution que nous avons proposée est de créer une application Web et Mobile de gestion commerciale. Cette application va traiter la gestion des bons d'achat, des bons de sortie, des commandes, des factures, des véhicules, des produits, des catégories ainsi que la gestion des utilisateurs.

Elle nous permet aussi de consulter les alertes générées par le système des produits dont les dates sont expirées ou des véhicules qui sont indisponibles. Donc cette solution sera bien détaillée dans la partie Analyse et spécification des besoins, conception et la partie réalisation.

1.4 Branche Fonctionnelle

L'analyse et la spécification des besoins est une phase cruciale pour comprendre les fonctionnalités de notre système. Ainsi, nous distinguons les besoins fonctionnels et non-fonctionnels. Nous présentons dans ce qui suit une analyse des besoins fonctionnels et

non fonctionnels de notre application et nous proposons, dans ce qui suit, de les spécifier à l'aide de diagrammes des cas d'utilisation.

1.4.1 Besoins fonctionnels

Nous rappelons qu'un besoin fonctionnel est un besoin spécifiant une action qu'un système doit être capable d'effectuer sans considérer aucune contrainte physique. C'est un besoin spécifiant un comportement d'entrée/sortie d'un système.

Notre objectif est de développer un logiciel de gestion commerciale qui facilite la manipulation des données aux administrateurs, magasiniers et transporteurs qui travaillent dans tels entreprise et aussi à leurs clients.

En effet, le logiciel à développer doit assurer les fonctionnalités suivantes :

Si l'acteur est un administrateur il peut :

- S'authentifier.
- Gérer les comptes des utilisateurs.

Si l'acteur est un magasinier il peut :

- S'authentifier.
- Gérer les produits.
- Gérer les catégories.
- Gérer les fournisseurs.
- Gérer les véhicules.
- Gérer les bons de sortie.
- Gérer les bons d'achat.
- Gérer les commandes.
- Gérer les factures.

Si l'acteur est un transporteur il peut :

- S'authentifier.
- créer un bon de sortie.

- visualiser les produits.
- Consulter l'historique des bon de sorties.

Si l'acteur est un client il peut :

- S'authentifier.
- passer des commandes.
- visualiser les produits.
- Consulter l'historique de ses documents.

1.4.2 Besoins non fonctionnels

En plus des besoins fonctionnels cités avant, le projet doit répondre à certaines exigences non fonctionnelles dites besoins non fonctionnels. Ces derniers représentent un ensemble de contraintes à respecter pour garantir l'efficacité du projet.

Dans ce qui suit, nous citons les plus importants besoins non fonctionnels :

- **La sécurité :** l'application devra assurer la sécurité des utilisateurs, d'où la nécessité de procéder l'authentification tout en assurant la confidentialité de leurs données.
- **Contraintes ergonomiques :** l'interface de l'application doit être simple et utilisable afin que l'utilisateur puisse l'exploiter sans se référer à des connaissances particulières, en d'autres termes, notre application doit être lisible et facile à manipuler par n'importe quel utilisateur .
- **Contraintes de disponibilité :** l'application doit être toujours fonctionnelle et disponible en permanence.
- **L'extensibilité :** le système doit être évolutif afin de répondre aux futurs besoins de la société. C'est-à-dire qu'il doit y avoir une possibilité d'ajouter de nouvelles fonctionnalités ou de modifier celles existantes.

1.5 Branche technique

Les besoins fonctionnels de notre système ayant été définis, il est temps de se poser la question de savoir quels outils allons-nous utiliser et surtout sur quelle architecture logicielle notre choix va se porter. C'est ainsi que nous définirons dans cette partie en premier lieu l'architecture logicielle, puis les langages de développement et nous terminerons par les serveurs utilisés.

1.5.1 Architecture logicielle

Nous devons savoir qu'il existe plusieurs types d'architectures. Parmi ces architectures, nous pouvons citer :

L'architecture client-serveur :

C'est la description du fonctionnement coopératif entre le serveur et le client. Les services internet sont conçus selon cette architecture. Ainsi, chaque application est composée de logiciel serveur et logiciel client.

- On appelle logiciel serveur un programme qui offre un service sur le réseau.
- On appelle logiciel client un programme qui utilise le service offert par un serveur.

On reconnaît traditionnellement dans une application 3 modules : Données, Traitement et Présentation.

La répartition de ces 3 modules variera entre le client et le serveur, donc suivant les contraintes d'utilisation ou contraintes techniques on a des différents types d'architectures client-serveur :

- **Architecture 1-tiers** : les 3 couches applicatives s'exécutent sur la même machine, on parle d'informatique centralisé et on a un contexte multi-utilisateurs dans le cadre de site central (mainframe).
- **Architecture 2-tiers** : l'architecture à deux niveaux (aussi appelée architecture 2-tier, tier signifiant rangée en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à

une autre application afin de fournir une partie du service. Pour la présentation et les traitements sont sur le client, les données sur le serveur et contexte multi-utilisateurs avec accées aux données centralisée.

- **Architecture 3-tiers** : dans l'architecture à 3 niveaux (appelée architecture 3-tier) il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :
 - Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation .
 - Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur.
 - Le serveur de données, fournissant au serveur d'application les données dont il a besoin.[9] Concernant le partage des 3 module on distingue que : pour la présentation est sur le client, les traitements sont pris par un serveur intermédiaire et les données sont sur un serveur de données.
- **Architecture n-tiers** : l'architecture n-tiers a été pensée pour pallier aux limitations des architectures trois tiers et concevoir des applications puissantes et simples à maintenir. En fait, l'architecture n-tiers qualifie la distribution d'application entre de multiples services et non la multiplication des niveaux de services.

Choix de l'architecture utilisée

Suite à l'étude qui vient d'être faite, notre choix s'est porté sur l'architecture client-serveur et plus précisément sur l'architecture 3-tiers :

- **Le client** : c'est un ordinateur personnel avec un navigateur web chargé de la représentation des interfaces de l'application qui sont destinées au utilisateurs de la société.
- **Le Serveur d'application (Middleware)** : WAMP joue le rôle d'un intermédiaire entre le client et le serveur de donnée en fournissant les ressources. Il récupère des données à partir du serveur de donnée et traite les requêtes envoyés par le client.

Dans notre cas, tous le traitement est centralisé coté serveur, le client chargé que de la présentation des données.

- **Le serveur de donnée :** MySQL sert à stocker les données et fournit à Apache les données qui a besoin.

Cette architecture offre plusieurs avantages :

Couplage faible : en cas de besoin de modification d'une fonctionnalité, nous ne risquons pas de modifier toute l'application.

Modularité : la réutilisation d'une fonctionnalité déjà développée ce qui nous fait gagner en terme de temps de développement et de maintenance.

1.5.2 Langage de programmation

Dans le développement de notre système, nous avons utilisé plusieurs langages tels que :

- **PHP :** ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que MySQL. Exécuté du côté serveur (l'endroit où est hébergé le site) il n'y a pas besoin aux visiteurs d'avoir des logiciels ou plugins particulier.[1]
- **HTML5 :** HTML5 est une évolution de la norme HTML dont le but avoué est de faciliter le développement d'interfaces utilisateur riches. HTML5 est beaucoup plus orienté applicatif que ses prédécesseurs et veut permettre de s'affranchir de plugins pour utiliser au maximum les technologies Web natives afin de construire une application riche.[2]
- **CSS3 :** un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML ou XML.[3]
- **JS :** JavaScript (qui est souvent abrégé en "JS") est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.

Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que node.js ou Apache CouchDB.[4]

1.5.3 Serveurs

Serveur de base de donnée :

MySQL : est un serveur de base de données relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySql fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment).[5]

Serveur d'application :

WAMP : WAMP est un acronyme informatique signifiant « Windows » « Apache » « MySQL » « PHP » dans la majorité des cas mais aussi parfois, « Perl », ou « Python ». Il s'agit d'un néologisme basé sur LAMP Architecture, il possède 4 composants. Les rôles de ces quatre composants sont les suivants : Apache est le serveur web « frontal » il est « devant » tous les autres et répond directement aux requêtes du client web (navigateur), le langage de script PHP sert la logique, MySQL stocke toutes les données de l'application, Windows assure l'attribution des ressources à ces trois composants.[6]

1.6 Conclusion

Dans ce premier chapitre nous avons présenté l'organisme d'accueil BON-GEST et leurs services ainsi que le cadre général de notre projet. Par ailleurs, ce chapitre nous a permis de définir les différentes fonctionnalités que doit offrir notre application. Le chapitre suivant permet de mettre en évidence la conception.

Chapitre 2

Conception

2.1 Introduction

Avant d'entamer le développement de notre projet, il est indispensable de choisir le formalisme de conception ainsi la détaillée. Nous décrivons, alors, le modèle adoptée pour comprendre comment nous allons diriger la construction des fonctionnalités à développer. À la fin, nous présentons des diagrammes de cas d'utilisations du système, le diagramme de classe et des diagrammes de séquence.

2.2 Méthode de conception

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

2.2.1 Le langage UML

Pour faciliter notre tâche nous avons recours au langage de modélisation unifié (UML : Unified Modelling Language) c'est une notation qui permet de modéliser un problème de façon standard. Ce langage qui est né de la fusion de plusieurs méthodes existantes auparavant est devenu une référence en terme de modélisation objet, UML est caractérisé par :

- C'est un langage formel et normalisé.
- Il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité.
- Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel.
- UML est un support de communication performant.

2.2.2 Diagrammes d'UML

UML s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes [7] .

Sept diagrammes comportementaux :

- Diagramme de cas d'utilisation.
- Diagramme d'états.
- Diagramme de séquence.
- Diagramme de vue d'ensemble des interactions.
- Diagramme de communication.
- Diagramme d'activité.
- Diagramme de temps.

Six diagrammes structurels :

- Diagramme d'objets.
- Diagramme de classes.
- Diagramme de pack-ages.
- Diagramme de structure composite.complexe.
- Diagramme de déploiement.
- Diagramme de composants.

2.2.3 Modèles d'UML utilisés

A cet effet on présente quelques diagrammes de modélisation, qu'on a jugé les plus importants pour la compréhension du fonctionnement du système :

- **Diagramme de cas d'utilisation** : il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.
- **Diagramme de classes** : il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations.

- **Diagramme de séquence** : il montre la séquence verticale des messages passés entre objets au sein d'une interaction.

2.3 Conception détaillée

Dans cette partie, nous présentons la conception de notre application, à travers des diagrammes de cas d'utilisations, le diagramme des classes du modèle de conception et quelques diagramme de séquence.

2.3.1 Identification des cas d'utilisations

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation. Ce diagramme permet d'identifier les possibilités d'interaction entre le système et les acteurs. Il représente toutes les fonctionnalités que le système doit fournir.

Dans notre cas les diagramme des cas d'utilisations cités par la suite décrit ce que peut faire un administrateur, un transporteur, un magasinier et un client lors de l'utilisation de l'application de gestion commerciale MGgest.

2.3.1.1 Diagramme de cas d'utilisation relatif à l'administrateur

Avant de commencer, en tenant compte des fonctionnalités communes entre les quatre acteurs, une relation d'héritage s'impose. Cela est expliqué par la figure 2.1.

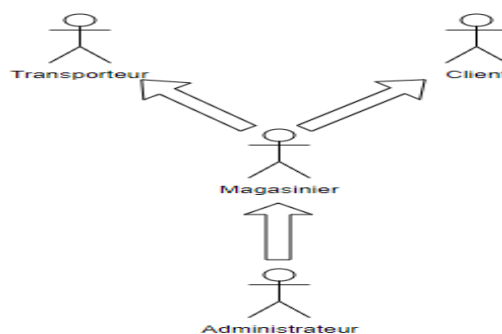


FIGURE 2.1 – Relation entre les acteurs

La figure 2.2 ci-dessous illustre le diagramme de cas d'utilisation relatif à l'administrateur, il nous permet d'obtenir une vision globale du comportement fonctionnel de notre application web coté administrateur .

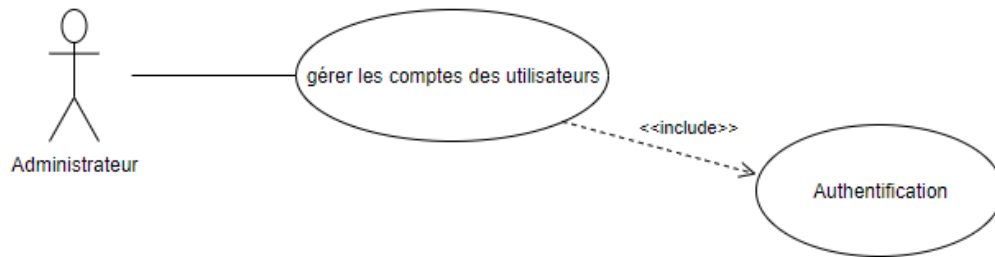


FIGURE 2.2 – Diagramme de cas d'utilisation relatif à l'administrateur

Pré-condition :

nous imposons que l'administrateur n'a le droit d'accéder au système qu'après son authentification.

Le cas d'utilisation :

- **Gérer les comptes des utilisateurs :** c'est une fonction privilégiée à l'administrateur. Il peut ajouter, modifier, rechercher ou supprimer un utilisateur.

2.3.1.2 Diagramme de cas d'utilisation relatif au magasinier

La figure 2.3 ci-dessous illustre le diagramme de cas d'utilisation relatif au magasinier, il nous permet d'obtenir une vision globale du comportement fonctionnel de notre application web coté magasinier .



FIGURE 2.3 – Diagramme de cas d'utilisation relatif au magasinier

Pré-condition :

Nous imposons que le magasinier n'a le droit d'accéder au système qu'après son authentification.

Les cas d'utilisation :

- **Gérer les bons d'achat** : un magasinier peut ajouter des bons d'achat en choisissant la liste des produits entrants, il peut aussi modifier, supprimer ou bien imprimer un bon d'achat.
- **Gérer les véhicules** : le magasinier peut gérer les véhicules, ainsi il peut consulter la liste des véhicules disponibles et ceux non disponibles.
- **Gérer les bons de sortie** : un magasinier peut ajouter des bons de sortie en choi-

sissant un transporteur, un véhicule, et la liste des produits sortants à chaque bon de sortie, il peut aussi supprimer, imprimer un bon de sortie ou bien le modifier.

- **Gérer les commandes** : un magasinier peut ajouter, modifier, supprimer, rechercher ou bien imprimer une commande.
- **Gérer les factures** : le magasinier peut gérer les factures qui seront, par la suite, affectées à un client.
- **Gérer les produits** : le magasinier peut aussi ajouter, modifier, supprimer ou rechercher un produit, il peut aussi Consulter les alertes des produits en rupture de stock.
- **Gérer les fournisseurs** : le magasinier peut gérer les fournisseurs, il peut ajouter, modifier, supprimer ou rechercher un fournisseur. Par ailleurs, il peut ajouter, modifier, supprimer les agents d'un fournisseur.
- **Gérer les approvisionnements** : l'approvisionnement est progressif et varie selon le volume de commandes et les bons de sortie.

2.3.1.3 Diagramme de cas d'utilisation relatif au transporteur

La figure 2.4 ci-dessous illustre le diagramme de cas d'utilisation relatif au transporteur, il nous permet d'obtenir une vision globale du comportement fonctionnel de notre application web coté transporteur .

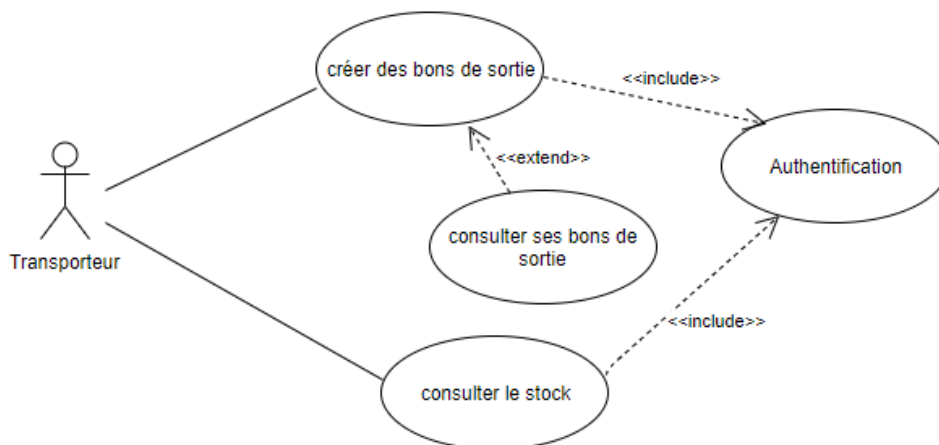


FIGURE 2.4 – Diagramme de cas d'utilisation relatif au transporteur

Pré-condition :

Nous imposons que le transporteur n'a le droit d'accéder au système qu'après son authentification.

Les cas d'utilisation :

- **Créer des bons de sortie** : un transporteur peut créer un bon de sortie en choisissant la liste des produits sortants et le véhicule, il peut aussi consulter l'historique de ses bons de sortie avec le gain pour chaque bon.
- **Consulter le stock** : un transporteur peut consulter le stock avec des alertes pour les produits indisponibles.

2.3.1.4 Diagramme de cas d'utilisation relatif au client

La figure 2.5 ci-dessous illustre le diagramme de cas d'utilisation relatif au client, il nous permet d'obtenir une vision globale du comportement fonctionnel de notre application web/mobile coté client.

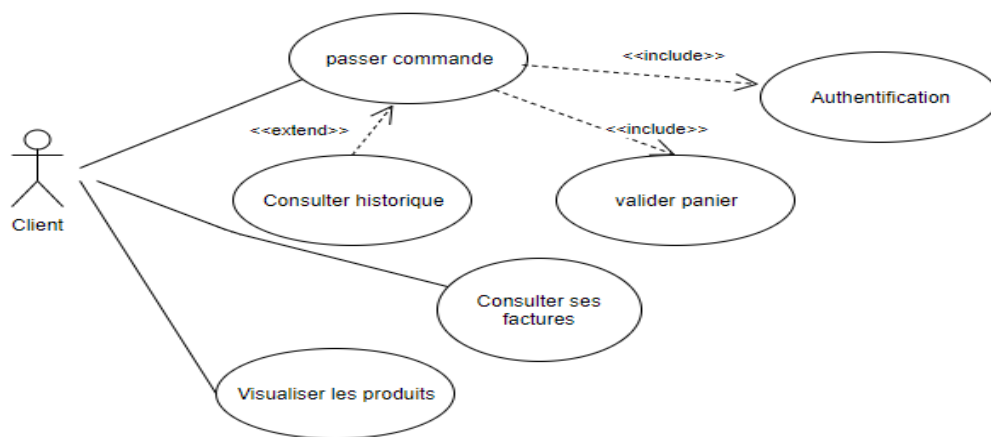


FIGURE 2.5 – Diagramme de cas d'utilisation relatif au client

Les cas d'utilisation :

- **Passer une commande** : un client peut ajouter une commande après son authentification et la validation de son panier, il peut aussi consulter ses commandes.
- **Visualiser les produits** : les produits sont accessibles pour tout visiteurs de site, authentifiés ou pas.

— **Consulter ses factures** : le client peut aussi consulter ses factures.

Dans la suite, nous présentons les raffinements de quelques cas d'utilisation pour les détailler.

2.3.1.5 Diagramme de cas d'utilisation relatif à l'authentification

Le tableau 2.1 illustre la description textuelle du cas d'utilisation "Authentification".

Titre	Authentification
But	Notre système exige la phase d'authentification pour les différents utilisateurs afin de bénéficier des différents services du système. Chaque utilisateur est authentifié par le couple "Login/mot de passe".
Acteurs	Administrateur, magasinier, transporteur, client
Description des enchainements	
Pré-condition	L'utilisateur doit avoir un compte.
Post-condition	L'utilisateur est authentifié.
Scénario principal	
1. L'utilisateur saisit les paramètres de connexion (login/mot de passe)et confirme les paramètres déjà entrés.	
2.Le système vérifie les paramètres login et mot de passe.	
3. Le système affiche l'espace de l'utilisateur authentifié.	
Scénario(s) alternatif(s)	
(1) : Si les données saisies sont invalides, le système affiche un message d'erreur et l'enchaînement reprend au point 1 de la séquence principale.	

TABLE 2.1 – Scénario "Authentification"

2.3.1.6 Diagramme de cas d'utilisation relatif à la gestion du véhicule

La figure 2.6 présente le diagramme de cas d'utilisation "Gérer les véhicules" raffiné :

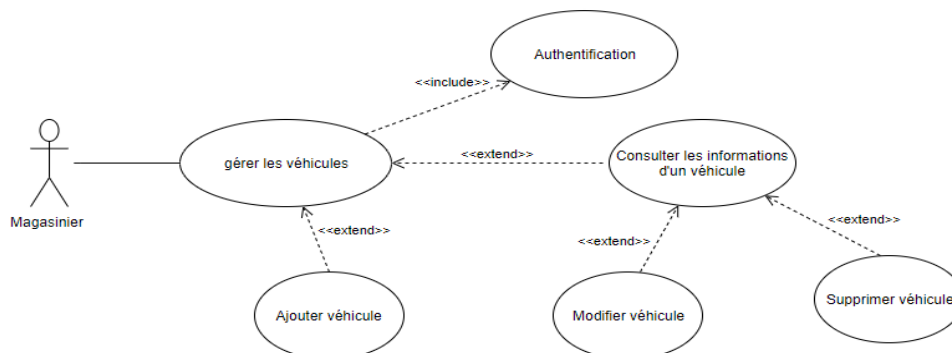


FIGURE 2.6 – Diagramme de cas d'utilisation "Gérer les véhicules"

Le tableau 2.2 présente la description textuelle "Gérer les véhicules".

Titre	Gérer les véhicules
But	Le système doit permettre au responsable d'ajouter d'autres voitures à l'application, de modifier leurs informations , de supprimer des voitures de la base ou de voir les disponibilités de ces derniers.
Acteur	Magasinier
Description des enchainements	
Pré-condition	Le responsable doit être authentifié afin de pouvoir accéder à l'application.
Post-condition	Opération effectuée avec succès.
Scénario principal	
1. Le magasinier choisit l'option "Véhicules" du menu.	
2. Le système affiche tous les véhicules qui existent dans la base.	
3. Le responsable choisit l'action qui veut effectuer, soit d'ajouter un véhicule , supprimer un véhicule , de modifier ces informations.	
4. Si le magasinier choisit d'ajouter un nouveau véhicule ou de modifier un autre existant.	
5.le système affiche un formulaire respectivement vide ou contenant les données du véhicule.	
6. Le magasinier remplit les champs puis il valide l'action.	
7. Le système enregistre les nouvelles données ou les changements entrés et affiche un message de validation.	
8. Si l'utilisateur choisit d'afficher les détails d'un véhicule sélectionné(Immatriculation, Marque, Modèle, carburant, État, Capacité). le système affiche les données relatives à ce dernier.	
9. Pour supprimer un véhicule, l'utilisateur doit choisir l'option "Supprimer".	
Scénario(s) alternatif(s)	
(5) : Si des informations manquent ou ne sont pas valides ,le système affiche un message d'erreur.	
(6)(7)(8) : En cas d'erreur,le système notifie l'utilisateur que la connexion a échoué ou une erreur est survenue sur la base de donnée lors de l'enregistrement des données.	

TABLE 2.2 – Scénario "Gérer les véhicules"

2.3.2 Diagramme des classes

Dans cette partie, nous allons présenter le diagramme de classe. Ce Diagramme est le cœur de la conception, il exprime la structure statique du système en terme de classes et de relations entre ces classes. L'intérêt du diagramme de classes est de modéliser les entités du système d'information.

La figure 2.7 met en relief le diagramme des classes de l'application en précisant dans chaque classe ses attributs et ses méthodes avec les relations qui existent entre elles.

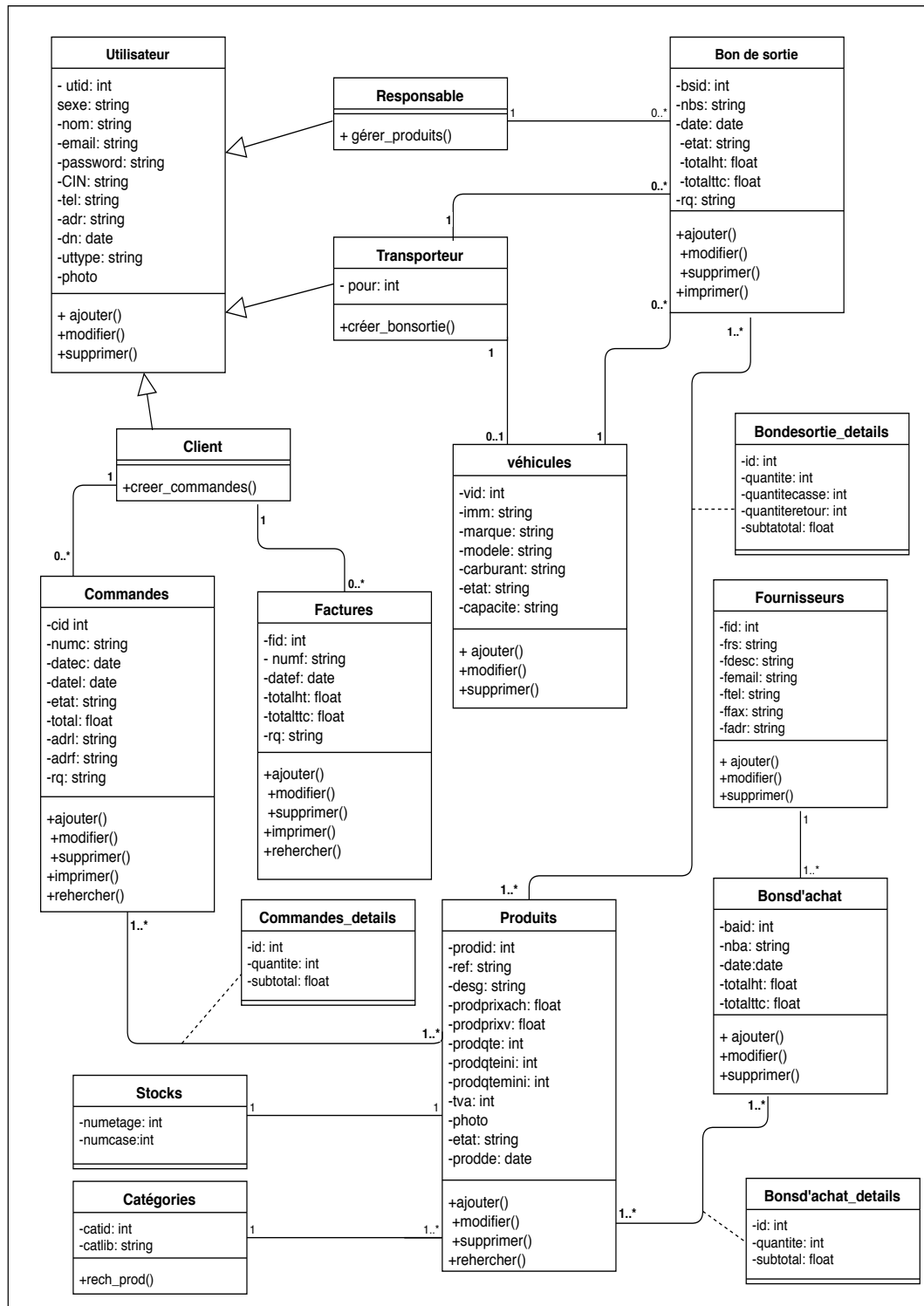


FIGURE 2.7 – Diagramme des classes

Nous présenterons dans ce qui suit un tableau descriptif des classes les plus importantes de la Figure 2.7.

	Description
Utilisateur	C'est l'utilisateur de l'application qui peut être un transporteur, un client ou un responsable(administrateur ou magasinier).
Bons de sortie	Cette classe représente le bon de sortie dont le transporteur va le remplir.Cette classe a une date de sortie, un numéro unique,un état et une total. Cette entité est en relation avec l'entité Produits, c'est à dire les produits sortants en spécifiant à chaque produit la quantité sortante, casse et retour.
Bons d'achat	Cette classe représente le bon d'achat dont le magasinier va le remplir.cette entité a une date d'entrée, un numéro unique et une totale.Cette classe est en relation avec l'entité Produits c'est à dire les produits entrés en stock.
Factures	Cette classe représente la facture affecté à un client. cette entité a une date de facturation , un numéro unique et une totale.
Commandes	Cette classe représente la commande dont le client va la remplir. Cette entité a une date de commande et un numéro unique et une totale. Cette classe est en relation avec Produits c'est à dire les produits commandés.
Véhicules	Chaque véhicule a sa matricule spécifique, une marque, un carburant,un modèle, un état(disponible ou indisponible) et la capacité. Un véhicule peut être affecté a un ou plusieurs bon de sortie.
Produits	Classe Produit représente l'enregistrement dans laquelle on stock les données relatives aux produits en stock. Cette classe contient une quantité minimale,qui à partir de cette entité le système va déclarer automatiquement une alerte lors de la proximité de rupture en stock.
Fournisseurs	cette classe représente l'enregistrement dans laquelle on stock les données relatives aux Fournisseurs. l'entité est en relation avec l'entité Bon d'achat c'est à dire chaque fournisseur a un ou plusieurs bon d'achat.

TABLE 2.3 – Tableau descriptif des classes

2.3.3 Diagrammes de séquence système

Nous rappelons que les diagrammes de séquences peuvent servir à illustrer un cas d'utilisation décrit précédemment. C'est un moyen semi-formel de capturer le comporte-

ment de tous les objets et acteurs impliqués dans un cas d'utilisation. Dans ce qui suit, quelques scénarios de l'application sont présentés.

2.3.3.1 Diagramme de séquence du scénario "Authentification"

la phase d'identification est indispensable pour la suite de l'utilisation de l'outil qui se déroule selon les informations relatives à l'utilisateur(administrateur, transporteur, magasinier ou client). Le diagramme de séquence système dans la figure 2.8 illustre plus en détails le scénario "Authentification".

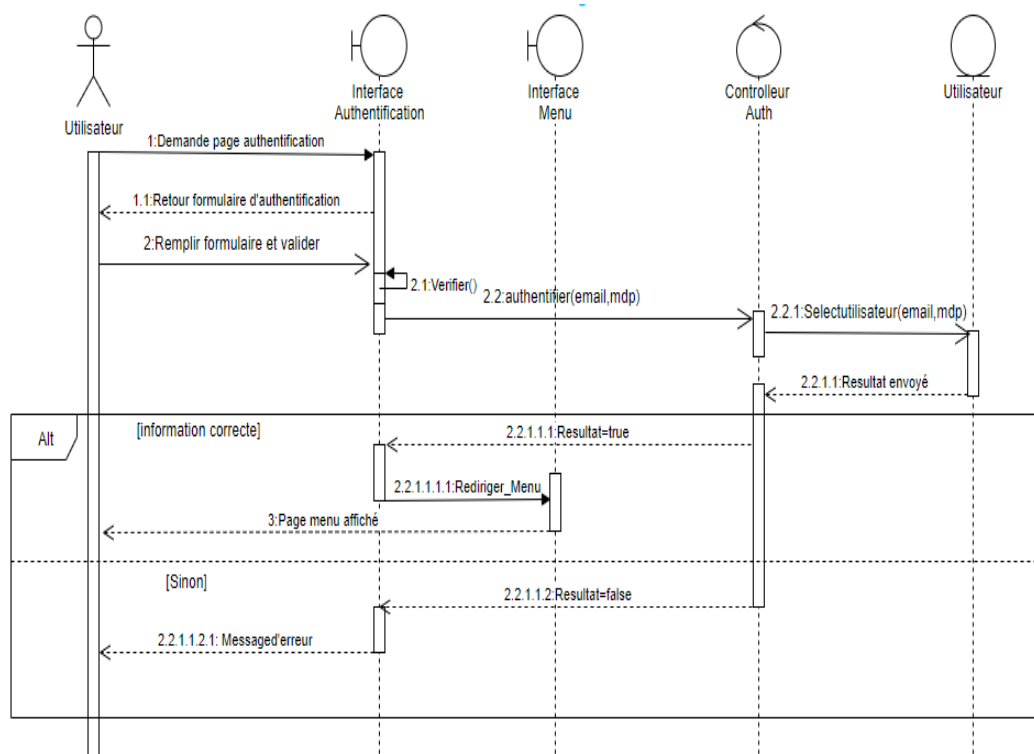


FIGURE 2.8 – Diagramme de séquence du scénario "Authentification"

2.3.3.2 Diagramme de séquence du scénario "Modifier Produit"

Le magasinier demande la liste des produits, le système a son tour affiche la liste des produits demandé. Il va sélectionner le produit à modifier, après la modification le système affiche soit un message de succès ou un message d'échec de modification suite à une erreur de saisie par exemple. Le diagramme de séquence dans la figure 2.9 illustre plus en détails le scénario " Modifier produit".

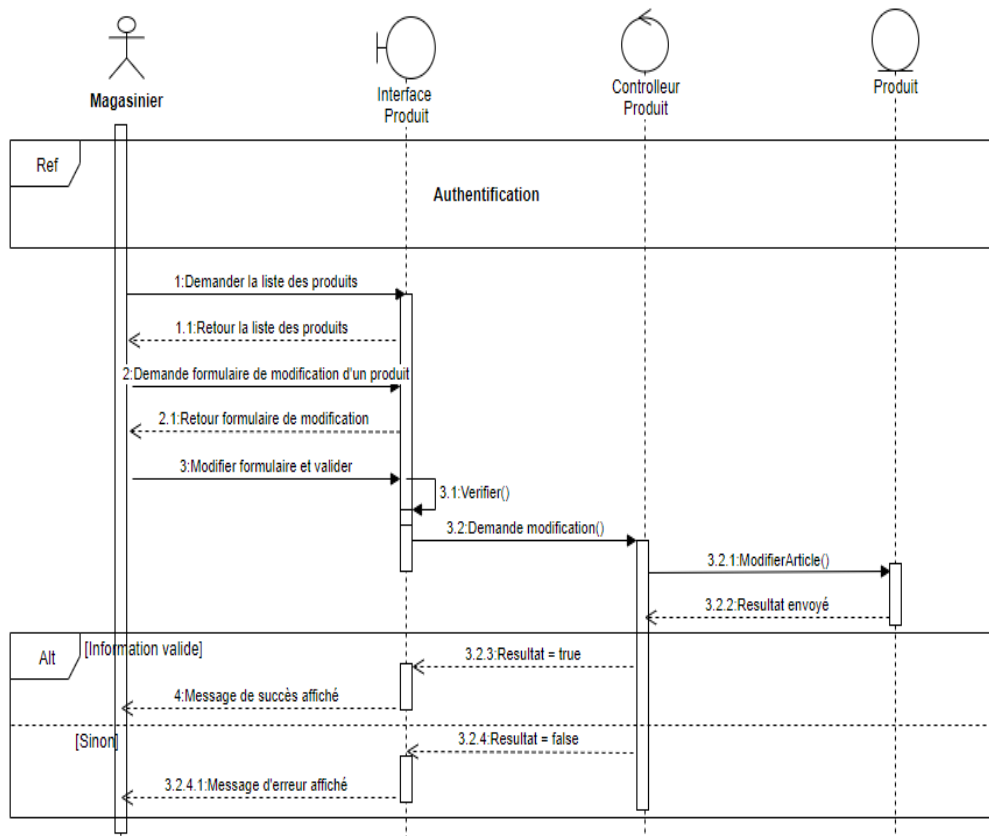


FIGURE 2.9 – Diagramme de séquence du scénario "Modifier produit"

2.3.3.3 Diagramme de séquence du scénario "Ajouter Bon de sortie"

Après authentification, le transporteur accède à la rubrique "Bon de sortie". Une liste des bons de sortie sera affichée, il choisit le bouton ajouter, le système affiche un formulaire à remplir. il remplit les champs du formulaire, la liste des produits et envoie les données pour que le système avec la base de données puisse vérifier la validité des champs, une série de tests doit être réalisée (transporteur existe, produit existe, quantité suffisante, véhicule existe..). Si tous les champs sont correcte alors le système prend en charge les informations introduites et valide le bon de sortie sinon une erreur de saisie va être affichée.

Le diagramme de séquence système dans la figure 2.10 illustre plus en détails le scénario "Ajouter Bon de sortie".

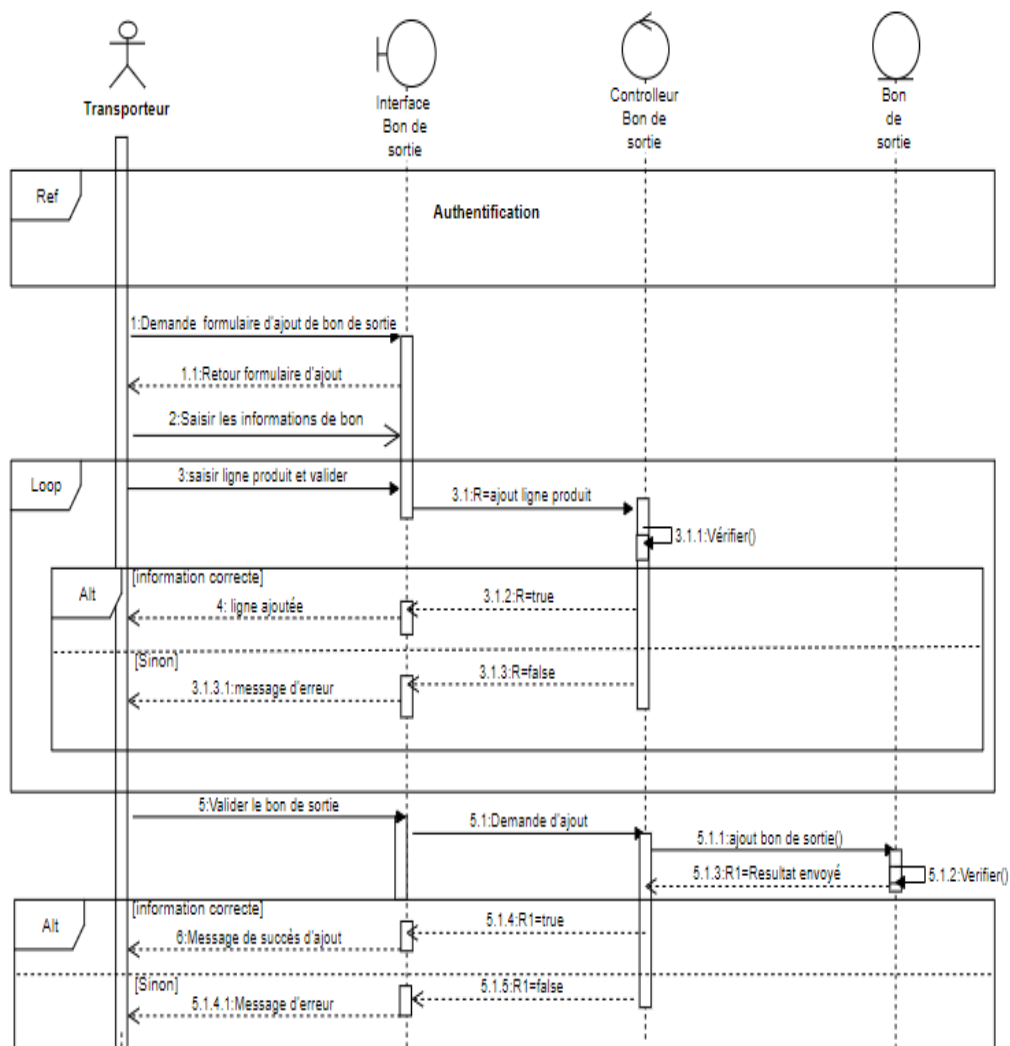


FIGURE 2.10 – Diagramme de séquence du scénario "Ajouter Bon de sortie"

2.3.3.4 Diagramme de séquence du scénario "Passer Commande"

Après authentification, le client accède à la rubrique "Produits". Une liste des produits sera affichée avec le prix et la quantité de chaque produit, il choisit les produits à commander, le système les enregistre dans le panier. il valide le panier pour que le système avec la base de données puisse vérifier la validité des champs, un test doit être réalisé pour les produits. Si tous les champs sont correcte alors le système prend en charge les informations introduites et valide la commande sinon une erreur de saisie va être affichée.

Le diagramme de séquence dans la figure 2.11 illustre plus en détails le scénario "Passer Commande".

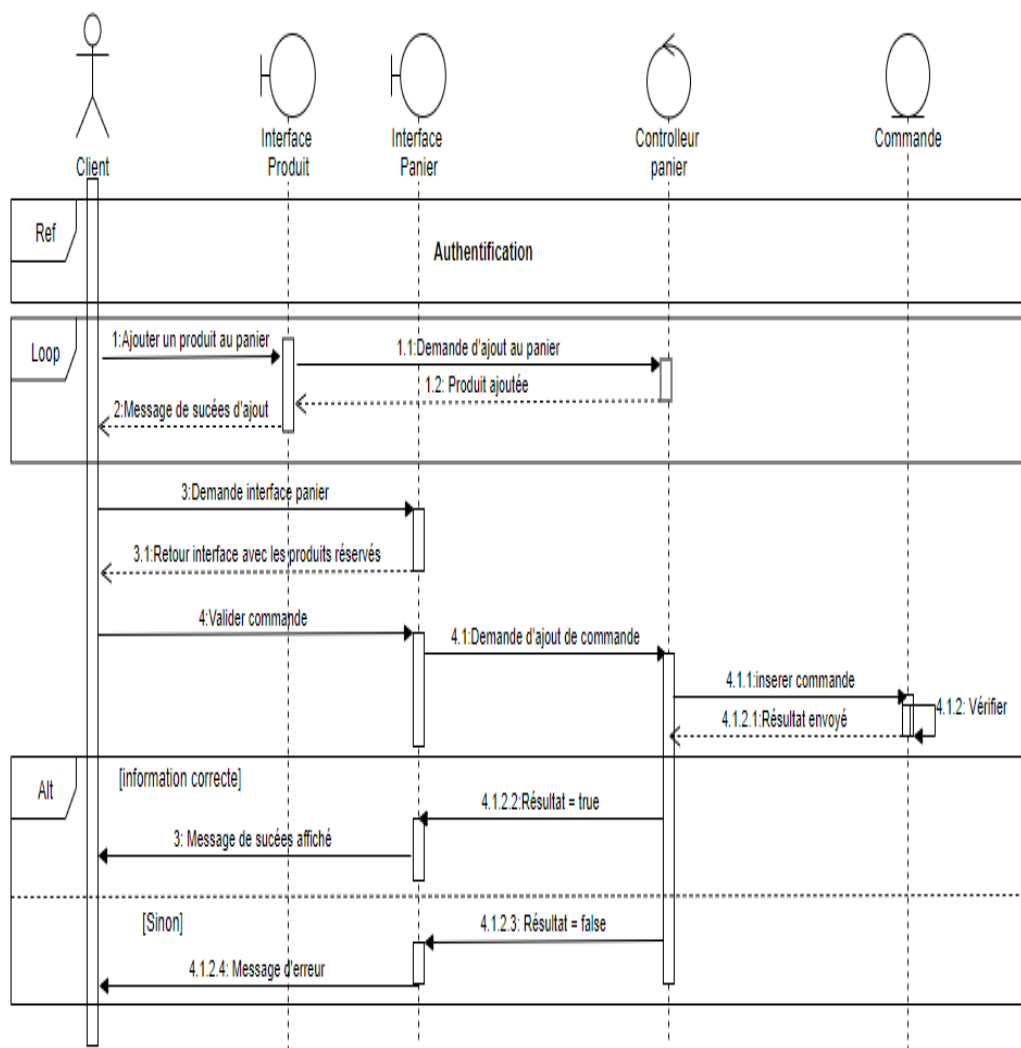


FIGURE 2.11 – Diagramme de séquence du scénario "Passer Commande"

2.4 Conclusion

Ce chapitre nous a permis le formalisme de conception utilisé. Puis, une deuxième phase était consacrée à la conception, et donc l'élaboration de différents diagrammes.

Dans le quatrième chapitre, nous présentons les étapes de réalisation de notre application ainsi que quelques interfaces représentatives de notre travail.

Chapitre 3

Réalisation

3.1 Introduction

L'étape de réalisation et de test représente également une étape fondamentale à entreprendre dans tout projet informatique. Pour cela, nous présentons les différentes interfaces que nous avons réalisées et leurs descriptions seront aussi présentées tout au long de ce chapitre.

3.2 Environnement et outils de travail

3.2.1 Environnement matériel

Pour réaliser notre application, nous avons utilisé un PC ayant les caractéristiques suivantes :

- Micro-ordinateur : ASUS VivoBook X541U (13 pouces).
- RAM : 8 Go de mémoire.
- Processeur : Intel Core i7-7500U à 3,5 GHz.

3.2.2 Environnement logiciel

Au cours de cette section nous décrivons les différents outils/logiciels et langages de programmation qui nous ont permis d'élaborer ce projet :

A) Framework Utilisé :

- **Laravel** : est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.[9]
- **Vue.js** :(aussi appelé plus simplement Vue), est un framework JavaScript open-source utilisé pour construire des interfaces utilisateur. La bibliothèque est notamment utilisée par Adobe, Alibaba et Gitlab. Vue a été créé par Evan You après avoir travaillé pour Google en utilisant AngularJS dans plusieurs projets.[10]
- **Jquery** : ou jQuery, est une bibliothèque JavaScript gratuite, libre et multiplateforme. Compatible avec l'ensemble des navigateurs Web (Internet Explorer, Safari,

Chrome, Firefox, etc.), elle a été conçue et développée en 2006 pour faciliter l'écriture de scripts. Il s'agit du framework JavaScript le plus connu et le plus utilisé. Il permet d'agir sur les codes HTML, CSS, JavaScript et AJAX et s'exécute essentiellement côté client. [11]

- **React Native** : est un framework mobile hybride développé par Facebook depuis début 2015. Il continue d'évoluer avec le soutien de nombreux contributeurs sur Github.[12]
- **CRUDBooster** : est le framework Laravel qui a été modifié pour apporter de nombreuses fonctionnalités permettant de développer une application Web de manière simple. L'une des caractéristiques principales est le générateur "Smart CRUD". C'est pourquoi CRUDBooster créera automatiquement un module incluant la fonction standard créer, lire, mettre à jour et supprimer. CRUDBooster n'est pas un générateur CRUD habituel(la connaissance de laravel est obligatoire).[13]

B) Technologies :

- **Bootstrap** : le cadre Bootstrap est un environnement de conception de site web qui facilite l'utilisation des règles CSS pour concevoir des sites web modernes, attractifs et adaptatifs (ResponsiveWeb Design) rapidement. Il existe actuellement deux versions, nous avons utilisé la version 3.[14]

C) Langage de Modélisation :

- **UML** : "Unified Modeling Language" est un langage de modélisation graphique, il est aujourd'hui utilisé comme standard international, incontournable, stabilisé et industriel.

D) Logiciel de Modélisation :

- **Draw.io** : est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter.[15]

3.3 Travail réalisé

Notre application contient principalement quatre types de comptes : un premier compte pour l'administrateur ,un deuxième pour les magasiniers, un troisième pour les transporteurs et un quatrième pour les clients.

Dans ce qui suit, nous présentons les principales fonctionnalités offertes par notre applications à travers les différentes captures d'écran.

3.3.1 Page d'authentification

C'est la page d'accueil de notre application Web pour l'administrateur, les magasiniers et les transporteurs. Elle est composée principalement de deux espaces de connexion qui donne aux différents acteurs l'accès à l'application. L'opérateur doit saisir son email et son mot de passe pour accéder à son espace. Cette page est illustrée par la figure 3.1. En cas où le mot de passe ou l'email saisie est erronée un message s'affiche .

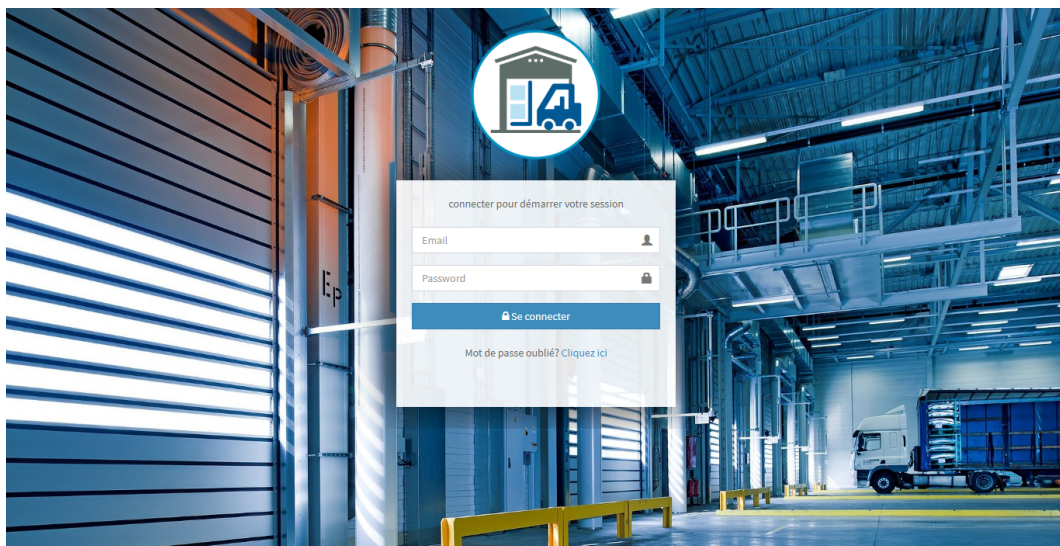


FIGURE 3.1 – Interface d'authentification

3.3.2 Espace Administrateur

La figure 3.2 présente l'interface de l'administrateur qui contient 11 menus : Tableau de bord, Bon d'achat, Bon de sortie, Commandes, Factures, Véhicules, Produits, Catégories, Utilisateurs, Fournisseurs et Stock .

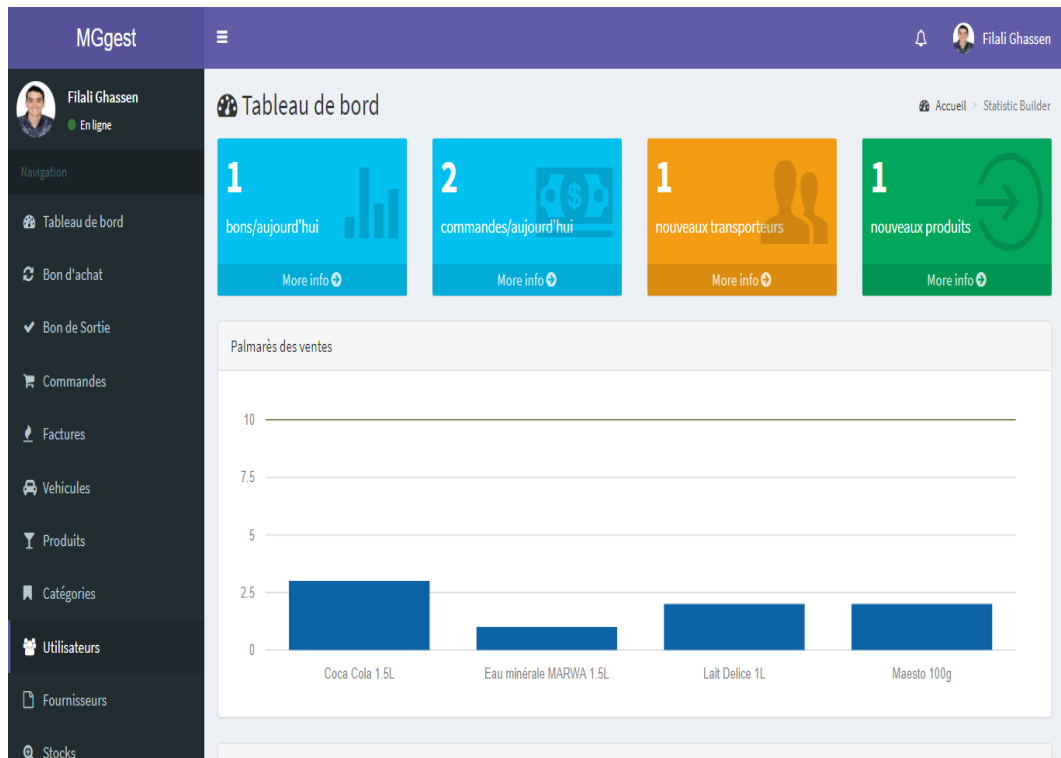


FIGURE 3.2 – Interface Administrateur

3.3.2.1 Menu utilisateurs

Le menu Comptes utilisateurs, permet la gestion des magasiniers, transporteurs et clients à travers les actions suivantes : Ajouter, Modifier, Supprimer, Chercher. La figure 3.3 suivante illustre tous les actions déjà cité.

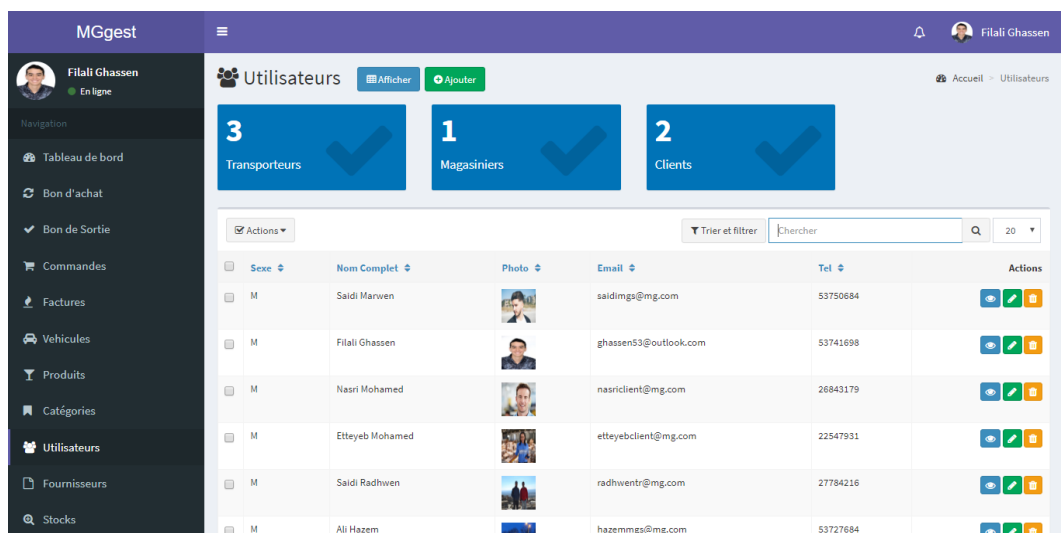


FIGURE 3.3 – Interface de gestion des utilisateurs

3.3.2.2 Gestion des factures

La figure 3.4 aborde l'interface dédiée pour la gestion des factures soit par un administrateur soit par un magasinier. Elle permet de créer une nouvelle facture, modifier, chercher, imprimer et supprimer des factures.

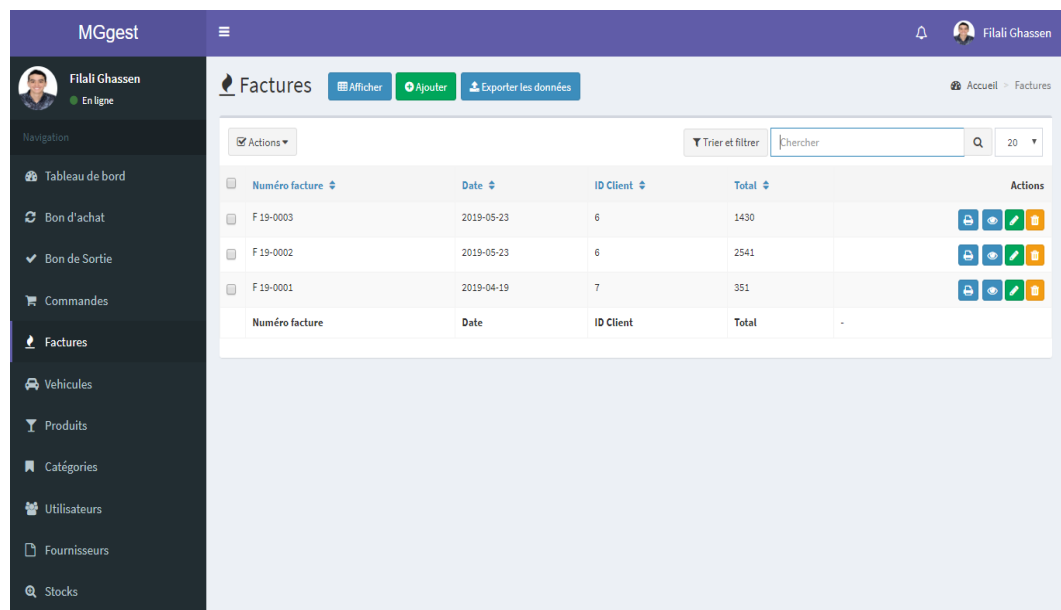


FIGURE 3.4 – Interface de gestion des factures

- Bouton "Ajouter" permet d'ajouter une nouvelle facture.
- L'icône "Stylo" permet d'afficher un formulaire modifiable contenant les informations d'une facture. L'administrateur applique les modifications souhaitées et clique sur le bouton "sauvegarder".
- Pour supprimer une facture il faut cliquer sur l'icône "Poubelle", un message de vérification s'affiche pour valider la suppression. Si nous choisissons "Oui" la facture sera supprimé de la liste, si nous choisissons "Non", l'action sera annulée.
- L'icône "Imprimante" permet d'avoir la version imprimable d'une facture. la figure 4.10 présente le cas d'impression d'une facture.
- **Ajouter une Facture** : l'administrateur ou le magasinier choisit le client et les produits, les autres champs sont remplis automatiquement. La figure 3.5 illustre le déroulement d'ajout d'une facture.

FIGURE 3.5 – Interface d’ajout d’une facture

3.3.3 Espace Magasinier

La figure 3.6 présente l’interface de magasinier qui contient 10 menus : Tableau de bord, Bon d’achat, Bon de sortie, Commandes, Factures, Véhicules, Produits, Catégories, Stocks et Fournisseurs.

Id	Numéro	ID transporteur	ID véhicules	Total	Actions
3	BS 19-0003	3	1	2519	[Icons]
2	BS 19-0002	2	1	1358	[Icons]
1	BS 19-0001	5	4	2276	[Icons]

FIGURE 3.6 – Interface Magasinier

Gestion des produits

La figure 3.7 aborde l’interface dédiée pour la gestion des produits par un magasinier. Elle permet de créer un nouveau produit, modifier, chercher et supprimer des produits ainsi qu’imprimer la liste des produits.

Référence	désignation	Prix d'achat	Quantité	Photo	Actions
JS 52-06	Jus tropico 0.5L	0.8	450		
CH 01	Maestro 100g	24.5	300		
BG 02	Coca Cola 1.5L	12.5	404		
LA 89-74	Lait Delice 1L	6.2	300		
EA 06	Eau minérale Sabrina 1.5l	2.8	80		

FIGURE 3.7 – Interface des produits

- L’icône "Oeil" permet de consulter les détails d’un produit.
- Le bouton "Exporter les données" permet d’imprimer un rapport "PDF" avec les détails de chaque produit.

3.3.4 Espace Transporteur

La figure 3.8 présente l’interface de transporteur qui contient 3 menus : Tableau de bord, Bon de sortie et Produits.

id	Numéro	ID transporteur	ID vehicules	Total	Actions
3	BS 19-0003	3	1	2519	
2	BS 19-0002	3	1	1358	
1	BS 19-0001	3	4	2276	

FIGURE 3.8 – Interface Transporteur

Ajout d'un bon de sortie

Le transporteur choisit les produits et un véhicule pour qu'il puisse passer un bon de sortie. La figure 3.9 illustre le déroulement d'ajout d'un bon de sortie.

The screenshot shows a web form titled "Ajouter Bon de Sortie". It contains several input fields and a radio button group. The fields are: ID (4), Date de sortie (2019-05-24), Numéro (BS 19-0004), Transporteur (Saïdi Radhwen), Téléphone (27784216), pourcentage de transporteur(%) (10), and Immatriculation de vehicules (2145TN145). There are two "Parcourir les données" buttons. At the bottom, there is a radio button group for "Etat" with options: Préparation (selected), en cours, and Confirmé. A sidebar on the left shows a menu with "bon de sorties details".

FIGURE 3.9 – Interface d'ajout d'un bon de sortie

Affecter un véhicule :

La figure 3.10 illustre le déroulement d'affectation d'un véhicule à un bon de sortie.

The screenshot shows a modal window titled "Parcourir les données | Immatriculation de vehicules". It contains a search bar and a table with three columns: ID, imm, and marque. The table has three rows of data. Each row has a "Sélectionner" button. The background shows the "Ajouter Bon de Sortie" form from Figure 3.9.

ID	imm	marque	Sélectionner
4	2145TN145	Renault	Sélectionner
2	1978TN160	Iveco	Sélectionner
1	2242TN142	Citroën	Sélectionner

FIGURE 3.10 – Interface d'affectation d'un véhicule

3.3.5 Espace Client

Puisque le client est le seul acteur qui puisse interagir avec notre système en utilisant l'interface mobile et l'interface web, nous allons diviser cette partie en deux sections.

3.3.5.1 Interface web :

- Le client peut visualiser la liste des produits avec leurs prix.
- Il peut consulter ses commandes et ses factures.
- Une fois le client choisit les produits, il doit valider le panier pour passer le commande.

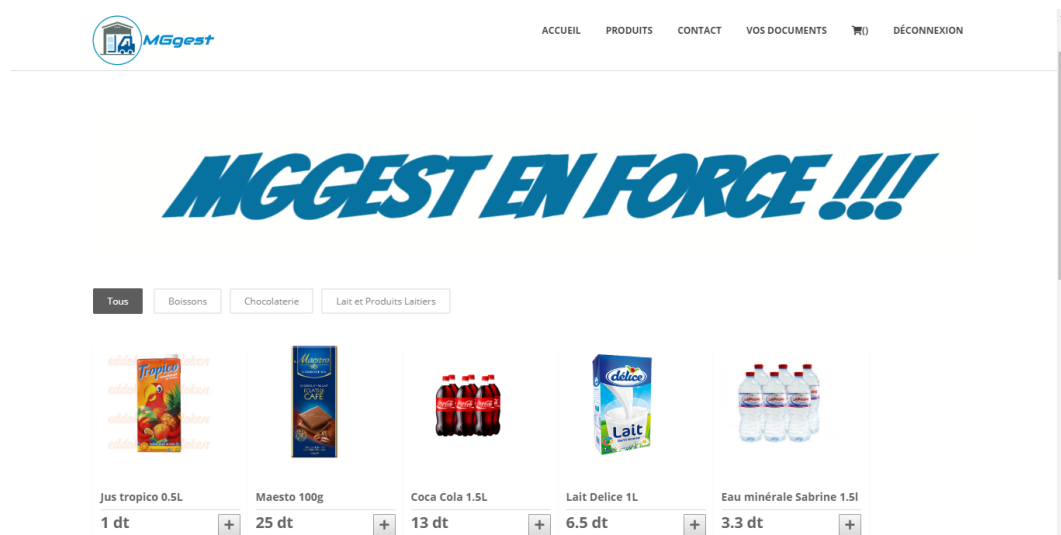


FIGURE 3.11 – Interface Web

3.3.5.2 Interface Mobile :

- Pour que le client puisse passer des commandes, il doit être authentifié.
- Si un nouveau client veut créer un compte pour passer des commandes, un formulaire de saisie s'affiche permettant à l'utilisateur d'entrer ses informations.

La figure 3.12 présente l'interface Mobile.



FIGURE 3.12 – Interface Mobile

3.4 Conclusion

Ce chapitre a été consacré à la description de la partie réalisation de notre application. Quelques interfaces sont exposées pour expliquer le fonctionnement de notre projet. Maintenant après avoir terminé tous les chapitres nous allons résumer notre travail dans la conclusion générale.

Conclusion générale

L'objectif de ce projet est de concevoir et de développer une application web et Mobile pour la gestion commerciale qui va permettre une meilleure gestion des données, et elle va résoudre plusieurs problèmes de la gestion avec Excel qui rend la circulation des informations très lente (la gestion des bons de sortie, des bons d'achat, des commandes...).

Dans ce rapport, nous exposons les différentes étapes suivies pour l'élaboration du projet. Une phase d'étude préliminaire nous a permis de spécifier les besoins fonctionnels ainsi que non fonctionnels et techniques. Ensuite, les différents choix technologiques et techniques adoptés pour la réalisation de ce projet. Par la suite, nous avons présenté une conception détaillée à travers la représentation des vues statiques et dynamiques du système.

Notre projet est clôturé par une phase de réalisation qui comporte une exposition du travail effectué.

Nous tenons à souligner que ce projet a été pour nous une expérience très enrichissante, qui nous a permis d'approfondir les connaissances que nous avons acquises au cours de notre étude. Cette expérience nous a également fait découvrir le monde professionnel et cela nous a été bénéfique autant d'un point de vue professionnel que relationnel en matière de communication et de coopération au sein d'une équipe.

Cependant, des améliorations de ce projet peuvent être envisagées. En effet, l'application peut être étendue en y ajoutant d'autres fonctionnalités comme l'ajout du module de géo-localisation et contrôle d'itinéraire, via navigation GPS par exemple contrôler, suivre

et protéger à distance et en temps réel les transporteurs, ce qui nous permettra de mieux répondre aux besoins des utilisateurs et offrira à ce dernier une plus grande satisfaction.

Enfin, ce stage est une conclusion parfaite au licence fondamentale en sciences de l'informatique et un excellent pas vers le professionnel.

Bibliographie

- [1] PHP [Enligne] URL[[http ://glossaire.infowebmaster.fr/php/](http://glossaire.infowebmaster.fr/php/)]
- [2] HTML5 [Enligne] URL[[http ://www.html5-css3.fr/html5/introduction-html5](http://www.html5-css3.fr/html5/introduction-html5)]
- [3] CSS3 [Enligne] URL[[http ://www.html5-css3.fr/css3/introduction-css3](http://www.html5-css3.fr/css3/introduction-css3)]
- [4] JS[Enligne] URL[[https ://developer.mozilla.org/fr/docs/Web/JavaScript](https://developer.mozilla.org/fr/docs/Web/JavaScript)]
- [5] Mysql [Enligne]URL[[https ://www.futura-sciences.com/tech/definitions/internet-mysql-4640/](https://www.futura-sciences.com/tech/definitions/internet-mysql-4640/)]
- [6] WAMP [Enligne]URL[[http ://dictionnaire.sensagent.leparisien.fr/WAMP/fr-fr/](http://dictionnaire.sensagent.leparisien.fr/WAMP/fr-fr/)]
- [7] Livre : Intitulé : UML 2 en action De l'analyse des besoins à la conception, Auteurs : Pascal Roques et Franck Vallée, 4ième édition, Année de publication : février 2007
- [8] ArchitectureClient/Serveur. [Enligne] URL[www.httr.ups-tlse.fr/pedagogie/cours/internet/services/servclie.html]
- [9] Laravel. [Enligne] URL[[https ://laravel.com/docs/4.2/introduction](https://laravel.com/docs/4.2/introduction)]
- [10] Vue.js [Enligne] URL[[https ://fr.vuejs.org/index.html](https://fr.vuejs.org/index.html)]
- [11] JQuery [Enligne] URL[[https ://jquery.com](https://jquery.com)]
- [12] ReactNative [Enligne]URL[[https ://makina-corpus.com/decouverte-de-react-native](https://makina-corpus.com/decouverte-de-react-native)]
- [13] CRUDBOOSTER [Enligne]URL[[https ://github.com/farnetani/crudbooster](https://github.com/farnetani/crudbooster)]
- [14] Bootstrap [Enligne] URL[[http ://getbootstrap.com/](http://getbootstrap.com/)]
- [15] Draw.io [Enligne] URL[[https ://www.tice-education.fr/index.php/tous-les-articles-et-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne](https://www.tice-education.fr/index.php/tous-les-articles-et-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne)]