

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



Université de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

MEMOIRE DE MASTER PROFESSIONNEL

En
Informatique

Option
Génie Logiciel

Thème

Conception et Réalisation d'une Application Web
de Gestion de la Relation Client (CRM) pour
S.P.A. - Général Emballage

Présenté par :

AMMOUCHE NAIMA
ASSAM MEZIANE

Soutenu le 27/06/2024 devant le jury composé de :

| | | | |
|--------------|-----------------|-----|----------------------|
| Président | MIR Foudil | MCB | Université de Béjaïa |
| Examinatrice | BELKHIRI Louiza | MCB | Université de Béjaïa |
| Promotrice | OUYAHIA Samira | MCB | Université de Béjaïa |

2023-2024.

** Remerciements **

Nous remercions tout d'abord DIEU, le tout puissant de nous avoir donné la force et la connaissance pour la réalisation de ce travail.

Nous tenons à exprimer nos sincères remerciements à notre promotrice Mme « OUYAHIA Samira » d'avoir accepté de nous encadrer, pour son aide, ses conseils, pour la pertinence de ses remarques et la justesse de sa correction.

Nous tenons à remercier les membres du jury d'avoir bien voulu examiner ce travail.

Un remerciement particulier à notre encadreur Mr «KESSOUM Aziz », ingénieur informaticien au sein de l'entreprise Général Emballage, pour son suivi durant toute la durée du travail.

Nos remerciements vont aussi au personnel de Général Emballage pour leur collaboration.

Nos reconnaissances s'adressent également à tous ceux qui de près ou de loin ont soutenu nos efforts et ont contribué à ce que ce travail voit le jour.

Tous les enseignants de l'université de Béjaia plus particulièrement ceux du département informatique, pour leurs apports inestimables durant notre cursus universitaire.

✱ *Dédicaces* ✱

Je dédie ce travail à :

À mes côtés fidèles, à celle dont les prières ont été la raison de ma réussite et le baume de mes blessures, à mon ange dans la vie, **ma mère**, que Dieu prolonge ta vie et accorde à ton cœur tout ce qu'il désire.

À celui dont je porte le nom, **mon père**, merci pour tout ce que tu m'as donné.

À mes chères sœurs Wezna, Yasmine, Mira, Marbouha, et à ma chère sœur Loubna et son mari pour leur soutien moral et leurs précieux conseils tout au long de mes études.

À mes chères grand-mères Tayakout et Halima, à qui je souhaite une bonne santé

À mes amis, pour leur présence et leurs encouragements.

À mes professeurs, pour leur enseignement précieux et leurs conseils avisés.

À mon binôme Meziane, avec qui j'ai eu le plaisir de partager ce projet de fin d'études.

À toute ma famille.

** Dédicaces **

Avec tout mon respect, je dédie ma remise de diplôme à :
À la lumière de ma vie, le fil d'espoir qui éclaire mon chemin, celle qui m'a toujours encouragé à donner le meilleur de moi-même durant toutes les années de mes études : **À ma chère MAMAN.**

À mon support dans la vie, qui a toujours fait des efforts et des sacrifices pour mon éducation et mon bien-être : **À mon cher PAPA.**

À mon cher frère Nadjib et A mes chères sœurs Celia et Ania, qui ont toujours cru en moi et ont été présents à mes côtés pour m'encourager et me soutenir.

À ma binôme Naima, avec qui j'ai eu le plaisir de partager ce projet de fin d'études.

À tous les membres de ma famille, mes amis et mes collègues de promotion, avec qui j'ai partagé les meilleures années d'études.

A .MEZIANE

Table des matières

| | |
|---|-------------|
| Table des matières | i |
| Liste des figures | vi |
| Liste des tableaux | viii |
| Liste des abréviations | x |
| Introduction Générale | xi |
| 1 Présentation de l'organisme d'accueil et étude de l'existant | 1 |
| 1.1 Introduction | 1 |
| 1.2 Présentation générale de l'organisme d'accueil | 1 |
| 1.2.1 Identité de l'entreprise | 2 |
| 1.2.2 Localisation | 2 |
| 1.2.3 Historique de l'entreprise Général Emballage | 3 |
| 1.2.4 Activités et missions de Général Emballage | 4 |
| 1.2.5 Valeurs de Général Emballage | 5 |
| 1.2.6 Organisation de la Spa Général Emballage Unité d'Akbou | 5 |
| 1.3 Présentation du champ d'étude | 7 |
| 1.3.1 Organigramme de la direction commerciale de Général Emballage | 7 |
| 1.3.2 Présentation de la direction commerciale | 8 |
| 1.4 Etude de l'existant | 8 |
| 1.4.1 L'informatique chez Général Emballage | 9 |
| 1.4.2 Flux de données | 10 |
| 1.4.3 Etude des services et postes | 11 |
| 1.4.3.1 Fiche d'analyse du poste de travail N° 1 : | 13 |
| 1.4.3.2 Fiche d'analyse du poste de travail N° 2 : | 14 |
| 1.4.4 Critiques et suggestions | 15 |

| | | |
|----------|--|-----------|
| 1.5 | Introduction aux CRM (Gestion Relation Client) | 16 |
| 1.5.1 | Définition d'un CRM | 16 |
| 1.5.2 | Les objectifs d'un CRM | 16 |
| 1.5.3 | Les types de CRM | 17 |
| 1.5.3.1 | Le CRM opérationnel | 17 |
| 1.5.3.2 | Le CRM analytique | 17 |
| 1.5.3.3 | Le CRM collaboratif ou CRM stratégique | 18 |
| 1.5.4 | Les modèles de CRM disponible | 18 |
| 1.5.4.1 | Logiciels (commerciaux) sous licence | 18 |
| 1.5.4.2 | Programme Open Source | 18 |
| 1.5.4.3 | Offre de Saas (logiciel en tant que service, CRM en ligne) | 19 |
| 1.5.5 | Contraintes du CRM | 20 |
| 1.5.6 | Les avantages du CRM | 20 |
| 1.6 | Démarche de la Solution proposée | 20 |
| 1.7 | Conclusion | 22 |
| 2 | Étude des besoins et Analyse | 23 |
| 2.1 | Introduction | 23 |
| 2.2 | Présentation du langage UML (Unified Modeling Language) | 23 |
| 2.3 | Méthode agile XP | 24 |
| 2.3.1 | Définition de la méthode agile | 24 |
| 2.3.2 | Définition de XP | 24 |
| 2.3.3 | Valeurs de XP | 25 |
| 2.3.4 | Cycle de vie XP | 25 |
| 2.4 | Démarches et Méthodologie XP | 26 |
| 2.5 | Identification des acteurs | 26 |
| 2.5.1 | Définition d'un acteur : | 26 |
| 2.5.2 | Les acteurs de notre système : | 27 |
| 2.6 | Spécification des exigences | 27 |
| 2.6.1 | Exigences fonctionnelles | 27 |
| 2.6.2 | Exigences non fonctionnelles | 28 |
| 2.7 | Diagramme de contexte | 29 |
| 2.8 | Modélisation fonctionnelle | 29 |
| 2.8.1 | Spécification des tâches pour chaque acteur | 30 |
| 2.8.2 | Diagramme de cas d'utilisation | 32 |
| 2.8.3 | Identification des cas d'utilisation | 33 |

| | | |
|----------|--|-----------|
| 2.8.4 | Diagramme de cas d'utilisation de « acteur : Administrateur » | 34 |
| 2.8.5 | Diagramme de cas d'utilisation de « acteur : Commercial » | 35 |
| 2.8.6 | Diagramme de cas d'utilisation de « acteur : Client » | 36 |
| 2.8.7 | Diagramme de cas d'utilisation de « acteur : Directeur commercial » | 37 |
| 2.8.8 | Description textuelle des cas d'utilisation précédents | 38 |
| 2.9 | Conclusion | 48 |
| 3 | Conception | 49 |
| 3.1 | Introduction | 49 |
| 3.2 | Diagramme de séquence | 49 |
| 3.2.1 | Diagramme de séquence du cas d'utilisation « Authentifier » | 50 |
| 3.2.2 | Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur : commercial » | 51 |
| 3.2.3 | Diagramme de séquence du cas d'utilisation « Commander » | 52 |
| 3.2.4 | Diagramme de séquence du cas d'utilisation « Suivre l'état de la commande » | 53 |
| 3.2.5 | Diagramme de séquence du cas d'utilisation « Suivre l'état du produit de chaque commande » | 54 |
| 3.3 | Diagramme de classes | 55 |
| 3.4 | Modélisation de la BDD | 56 |
| 3.4.1 | Modèle relationnel | 56 |
| 3.4.2 | Structure des tables de la BDD | 57 |
| 3.4.2.1 | Utilisateur | 57 |
| 3.4.2.2 | Directeur Commercial | 57 |
| 3.4.2.3 | Client | 58 |
| 3.4.2.4 | Commercial | 58 |
| 3.4.2.5 | Admin | 59 |
| 3.4.2.6 | Catégorie | 59 |
| 3.4.2.7 | Produit | 60 |
| 3.4.2.8 | CommandeClient | 60 |
| 3.4.2.9 | Produit-Commande | 61 |
| 3.4.2.10 | Message | 61 |
| 3.4.2.11 | Réponse | 61 |
| 3.4.2.12 | Sondage | 62 |
| 3.4.2.13 | Notification | 62 |
| 3.5 | Le passage du modèle objet au modèle relationnel | 62 |

| | | |
|----------|---|-----------|
| 3.6 | Conclusion | 64 |
| 4 | Réalisation | 65 |
| 4.1 | Introduction | 65 |
| 4.2 | Langages de développement | 65 |
| 4.2.1 | HTML | 65 |
| 4.2.2 | CSS | 66 |
| 4.2.3 | PHP | 66 |
| 4.2.4 | MySQL | 66 |
| 4.2.5 | Javascript | 66 |
| 4.3 | Outils de développement | 66 |
| 4.3.1 | Laravel | 66 |
| 4.3.2 | Visual Studio Code | 67 |
| 4.3.3 | XAMPP | 67 |
| 4.3.4 | Bootstrap | 68 |
| 4.4 | Outils d'édition UML | 68 |
| 4.4.1 | StarUML | 68 |
| 4.4.2 | Visual Paradigm | 69 |
| 4.5 | Architecture de l'application | 69 |
| 4.6 | Présentation des Interfaces de notre application | 70 |
| 4.6.1 | Page d'authentification | 70 |
| 4.6.2 | Interface Gestion des commerciaux (espace administrateur) | 71 |
| 4.6.3 | Interface Gestion des catégories (espace administrateur) | 71 |
| 4.6.4 | Interface Gestion des client (espace commercial) | 71 |
| 4.6.5 | Interface Gestion des produits (espace commercial) | 72 |
| 4.6.6 | Interface Gestion suivi de commande des clients (espace commercial) | 72 |
| 4.6.7 | Interface Messagerie (espace commercial) | 73 |
| 4.6.8 | Interface Liste des produits pour client (espace client) | 73 |
| 4.6.9 | Interface Liste des commandes clients (espace client) | 74 |
| 4.6.10 | Interface Consulter le retour des clients (espace directeur commercial) | 74 |
| 4.6.11 | Interface Consulter et valider les commandes(espace directeur commercial) | 75 |
| 4.6.12 | Interface Consulter la liste des commerciaux(espace directeur commercial) | 76 |
| 4.6.13 | Interface Consulter la liste des clients(espace directeur commercial) . | 76 |
| 4.6.14 | Interface Tableau de bord directeur commerciale | 76 |

| | | |
|--------|--|-----------|
| 4.6.15 | Interface Tableau de bord commerciale | 77 |
| 4.6.16 | Interface Tableau de bord client | 77 |
| 4.6.17 | Interface Tableau de bord administrateur | 78 |
| 4.7 | Conclusion | 78 |
| | Conclusion Générale | 79 |
| | Bibliographie | 81 |
| | Annexes | 83 |

Table des figures

| | | |
|-----|---|----|
| 1.1 | Logo de Général Emballage | 2 |
| 1.2 | Organigramme général de Général Emballage Unité d'Akbou | 6 |
| 1.3 | Organigramme de la direction commerciale de Général Emballage | 7 |
| 1.4 | ERP VoluPack | 10 |
| 1.5 | Le graphe de flux de la direction commerciale de Général Emballage | 11 |
| 2.1 | Représentation de la Démarches et Méthodologie XP. | 26 |
| 2.2 | Diagramme de contexte | 29 |
| 2.3 | Diagramme de cas d'utilisation de « acteur : Administrateur » | 34 |
| 2.4 | Diagramme de cas d'utilisation de « acteur : Commercial » | 35 |
| 2.5 | Diagramme de cas d'utilisation de « acteur : Client » | 36 |
| 2.6 | Diagramme de cas d'utilisation de « acteur : Directeur commercial » | 37 |
| 3.1 | Diagramme de séquence du cas d'utilisation s'authentifier « acteur : Utilisateur » | 50 |
| 3.2 | Diagramme de séquence du cas d'utilisation ajouter un commercial « acteur : Administrateur » | 51 |
| 3.3 | Diagramme de séquence du cas d'utilisation commander « acteur : Client » . . | 52 |
| 3.4 | Diagramme de séquence du cas d'utilisation suivre l'état de la commande « acteur : Commercial » | 53 |
| 3.5 | Diagramme de séquence du cas d'utilisation suivre l'état du produit de chaque commande « acteur : Commercial » | 54 |
| 3.6 | Diagramme de classes. | 55 |
| 4.1 | Logo de laravel | 67 |
| 4.2 | Logo Visual Studio Code | 67 |
| 4.3 | Logo XMPP | 68 |
| 4.4 | Logo Bootstrap | 68 |
| 4.5 | Logo StarUML | 68 |
| 4.6 | Logo Visual Paradigm | 69 |

| | | |
|------|---|----|
| 4.7 | Architecture MVC | 70 |
| 4.8 | Page d'authentification | 70 |
| 4.9 | Interface Gestion des commerciaux | 71 |
| 4.10 | Gestion des catégories | 71 |
| 4.11 | Gestion des clients | 72 |
| 4.12 | Interface Gestion des produits | 72 |
| 4.13 | Interface Gestion suivi de commande | 73 |
| 4.14 | Interface Messagerie espace commercial | 73 |
| 4.15 | Interface Liste des produits client | 74 |
| 4.16 | Interface Liste des commandes clients | 74 |
| 4.17 | Interface Gestion répondre aux sondage | 75 |
| 4.18 | Interface Consulter et valider les commandes | 75 |
| 4.19 | Interface Consulter la liste des commerciaux | 76 |
| 4.20 | Interface Consulter la liste des clients | 76 |
| 4.21 | Interface Tableau de bord directeur commerciale | 77 |
| 4.22 | Interface Tableau de bord commerciale | 77 |
| 4.23 | Interface Tableau de bord client | 78 |
| 4.24 | Interface Tableau de bord e administrateur | 78 |
| 25 | Structure du dossier projet Laravel | 88 |
| 26 | Fonctionnalités principales de Laravel | 89 |
| 27 | Structure des routes sur Laravel (connexion administrateur). | 90 |
| 28 | Migration de la table « Client » | 90 |
| 29 | Quelques commandes offertes par Artisan | 91 |
| 30 | Requête SQL pour récupérer l'état des produits réalisé avec Query-builder | 91 |
| 31 | Partie de code de la vue « Profil du Client » écrit avec Blade | 92 |
| 32 | Modèle de la table « catégorie » écrit en ORM eloquent | 92 |

Liste des tableaux

| | | |
|------|---|----|
| 1.1 | Flux de données | 11 |
| 1.2 | Fiche d'analyse du poste de travail N° 1. | 13 |
| 1.3 | Fiche d'analyse du poste de travail N° 2. | 14 |
| 2.1 | Spécification des tâches pour chaque acteur du système | 31 |
| 2.2 | Acteurs et cas d'utilisation | 33 |
| 2.3 | Description de cas d'utilisation « Authentification » | 39 |
| 2.4 | Description de cas d'utilisation « Gestion des catégories » | 40 |
| 2.5 | Description de cas d'utilisation « Gestion des clients » | 41 |
| 2.6 | Description de cas d'utilisation « Gestion des produits » | 42 |
| 2.7 | Description du cas d'utilisation « Liste des commandes clients » | 43 |
| 2.8 | Description du cas d'utilisation "Gestion des états des produits commandés" | 44 |
| 2.9 | Description du cas d'utilisation " Répondre au sondage" | 45 |
| 2.10 | Description du cas d'utilisation "Consulter la liste des commerciaux" | 45 |
| 2.11 | Description du cas d'utilisation "Consulter la liste des clients" | 46 |
| 2.12 | Description du cas d'utilisation "Consulter et valider les commandes" | 47 |
| 3.1 | Structure de la table « Utilisateur » de la base de données | 57 |
| 3.2 | Structure de la table « DirecteurCommercial » de la base de données | 57 |
| 3.3 | Structure de la table « Client » de la base de données | 58 |
| 3.4 | Structure de la table « Commercial » de la base de données | 58 |
| 3.5 | Structure de la table « Admin » de la base de données | 59 |
| 3.6 | Structure de la table « Catégorie » de la base de données | 59 |
| 3.7 | Structure de la table « Produit » de la base de données | 60 |
| 3.8 | Structure de la table « Commande Client » de la base de données | 60 |
| 3.9 | Structure de la table « ProduitCommande » de la base de données | 61 |
| 3.10 | Structure de la table « Message » de la base de données | 61 |
| 3.11 | Structure de la table « Réponse » de la base de données | 61 |
| 3.12 | Structure de la table « Sondage » de la base de données | 62 |

3.13 Structure de la table « Notification » de la base de données 62
3.14 Équivalence entre les concepts objet et relationnel 63

Liste des abréviations

API : Application Programming Interface

BDD : Base de données

CRM : Customer Relationship Management

CSS : Cascading Style Sheets

ERP : Enterprise Resource Planning

HTML : Hypertext Markup Language

HTTP : Hypertext Transfer Protocol

JS : JavaScript

MVC : Model View Controller

PHP : Hypertext Preprocessor

SGBD : Système de gestion de base de données

SGBDR : Système de gestion de base de données relationnel

SQL : Structured Query Language

UML : Unified Modeling Language

URL : Uniform Resource Locator

XP : eXtreme Programming

Introduction générale

Aujourd'hui, la concurrence est rude pour les entreprises car les consommateurs sont de plus en plus exigeants et insaisissables, il devient plus difficile d'attirer leur intérêt. Les entreprises sont donc dans l'obligation, pour survivre, de recentrer leurs préoccupations autour du client. Cette préoccupation est devenue très importante aussi bien chez les industriels que dans le marché des services. En quelques années, les entreprises ont dû passer d'une activité centrée autour des produits à une activité centrée autour des clients.

Elles se trouvent dans l'obligation de définir une stratégie bien précise et la suivre pour pouvoir exploiter au maximum les informations à leurs dispositions d'où la notion du CRM (Customer Relationship Management) ou GRC (Gestion Relation Client), qui va leur permettre d'être entièrement orientées vers leurs clientèles et leurs prospects, et cela via l'écoute active de leurs attentes menant à des réponses ou même à des anticipations appropriées pour ainsi offrir une qualité de service irréprochable et d'être performantes dans leurs recherches et le suivi des prospects.

Sans doute, les outils informatiques ont rendu possible, à ce jour, de mettre le client au centre de toutes les activités réalisées par les entreprises. Et c'est grâce à ces outils informatiques qu'elles ont trouvé la clé pour réussir à mettre en place une stratégie CRM (Customer Relationship Management ou Gestion de la Relation Client).

Général Emballage, entreprise fondée en 2000 et opérant dans le secteur de l'impression et de l'édition, est en pleine évolution. Face à cette transformation, l'entreprise se pose la question suivante : comment créer durablement le maximum de valeur à partir de ses clients existants ?

Général Emballage, doit développer et exploiter ses relations car elles constituent la clé du succès pour offrir à ses clients des services nouveaux et complémentaires. Il était l'heure pour elle de mettre en place une stratégie par laquelle elle vise à comprendre, à anticiper et à gérer les besoins de ses clients actuels et potentiels.

Le présent travail tend à la mise en place d'un système, par lequel l'entreprise sera en mesure d'assurer un suivi personnalisé, en offrant la meilleure qualité de service possible. Il s'agit en fait de mettre à la disposition des clients, commerciaux et décideurs une plateforme web unique qui va pouvoir offrir, d'une part, aux clients le moyen de commander des produits, de suivre leur avancement en toute transparence et de communiquer à tout moment leurs réclamations. D'autre part, les commerciaux pourront bénéficier d'une organisation pour pouvoir satisfaire les besoins des clients les plus exigeants. Enfin, les décideurs auront un aperçu de

l'activité commerciale à travers un tableau de bord contenant les indicateurs nécessaires à l'entreprise, pour l'aider à réagir et prendre les bonnes décisions afin de toujours satisfaire ses clients.

Notre mémoire est organisé en quatre (04) chapitres :

Chapitre 01 : Ce chapitre est dédié à la présentation de l'entreprise Général Emballage, lieu de notre stage. Nous y aborderons les différents problèmes rencontrés au sein de l'unité commerciale, ainsi que la définition du concept de CRM, ses fonctionnalités et ses avantages.

Chapitre 02 : Ce chapitre se concentre sur la collecte et l'analyse des exigences des parties prenantes pour le projet CRM web. Il inclut l'identification des besoins, l'analyse des processus existants, et la validation des exigences pour garantir que la solution proposée répondra aux attentes des utilisateurs.

Chapitre 03 : Ce chapitre détaille la transformation des exigences en une architecture technique et fonctionnelle. Il inclut la conception de l'architecture système, la modélisation des données, la conception des interfaces utilisateur, et la spécification des fonctionnalités du CRM.

Chapitre 04 : Ce chapitre présente les outils et technologies choisis pour développer le CRM web. Il décrit les environnements de développement, les systèmes de gestion de bases de données, les langages de programmation, les outils de gestion de projet, ainsi que les outils de test et de déploiement.

Chapitre 1

Présentation de l'organisme d'accueil et étude de l'existant

1.1 Introduction

Ce chapitre est divisé en deux parties essentielles. La première partie se concentre sur la présentation complète de l'organisme d'accueil, Général Emballage, où notre étude est basée sur un stage pratique. Nous examinerons en détail l'organisation de Général Emballage, ses activités principales ainsi que ses objectifs stratégiques.

Le deuxième volet de ce chapitre explore l'analyse du domaine spécifique sur lequel notre projet se concentre, mettant en lumière les principaux axes qui guideront notre travail de recherche et de développement.

1.2 Présentation générale de l'organisme d'accueil

Cette partie présentera l'entreprise sous ses différents aspects. Elle mettra ainsi l'accent sur ses activités, ses missions et son organisation.

1.2.1 Identité de l'entreprise

Général Emballage (G.E) est une entreprise papetière algérienne spécialisée dans la fabrication et la transformation de carton ondulé (son logo est présenté dans la figure 1.1 [1]) . Cette entreprise a été créée en l'an 2000 dans la zone industrielle Taharacht à Akbou, wilaya de Béjaïa, par Remdane Bouctouche. Général Emballage est le plus grand producteur de carton ondulé en Afrique. Aujourd'hui, cette grande entreprise compte parmi ses clients la quasi-totalité des plus grandes entreprises en Algérie, car elle satisfait 80 % des demandes nationales de carton ondulé, dont Cevital, Soummam, Candia, etc. Elle exporte actuellement ses produits vers la Tunisie, la France, le Maroc et certains pays d'Afrique subsaharienne [1].



FIGURE 1.1 – Logo de Général Emballage

1.2.2 Localisation

Son siège social est implanté au niveau de la zone d'activité de Taharacht, située à 2,5 km au nord-est du chef-lieu de la commune d'Akbou. Cette zone représente un véritable carrefour économique en raison du nombre d'unités industrielles qui exercent dans différents domaines sur une superficie totale de 24 hectares [1].

1.2.3 Historique de l'entreprise Général Emballage

Depuis sa création, Général Emballage ne cesse de se développer et de d'augmenter son capital et l'effectif de ses employés, voici son historique [1] :

- **2000, 1er Août** : Création de la SARL Général Emballage avec un capital de 32 millions de dinars dans la Zone d'activités de Taharacht (Akbou, Wilaya de Béjaïa) (décision APSI N°13051 du 06 juin 1998).
- **2002** : Entrée en production de l'usine d'Akbou avec un effectif de 83 employés.
- **2006** : Le capital est porté à 150 millions de dinars et l'entreprise comptait un effectif de 318 employés.
- **2007** : Le capital est porté à 1,23 milliards de dinars, entrée en production de l'usine de Sétif avec un effectif de 425 employés. Général Emballage remporte le Trophée de la Production (Euro-Développement PME).
- **2008** : Début d'exportation vers la Tunisie et entrée en exploitation de l'unité d'Oran.
- **2009** : Augmentation du capital à 2 milliards de DA et entrée de MAGHREB PRIVATE EQUITY FUND II « Cyprus II » (MPEF II) avec une participation de 40%. L'entreprise devient une société de capitaux (Société par actions) avec un effectif de 597 employés.
- **2010** : Avec son chiffre d'affaires, elle parvient à occuper une place parmi les 50 grandes entreprises algériennes avec un effectif de 630 employés.
- **2011** : La capacité totale de production des trois usines d'Akbou, d'Oran et de Sétif était de 130 000 tonnes, soit 80% de la consommation algérienne. L'entreprise comptait un effectif de 699 employés.
- **2012** : Les capacités de production sont portées à 130 000 tonnes. L'usine d'Oran est transférée à la ZI Hassi-Ameur. La production des premiers ouvrages en haute résolution est lancée. Signature d'une Convention cadre de partenariat avec l'Université de Béjaïa. L'effectif était de 830 employés.
- **2013** : Général Emballage obtient la Certification ISO 9001 :2008. Démarrage de la 1ère promotion de Licence en Emballage et Qualité à l'Université de Béjaïa. L'effectif était de 960 employés.
- **2014** : Signature d'un protocole d'accord de recrutement avec l'Agence Nationale de l'Emploi. Début des exportations vers la Libye. L'effectif était de 1005 employés.
- **2015** : Entrée en production de la nouvelle usine de Sétif à ZI Ain Sfiha. Elle remporte le Prix d'encouragement du Trophée Export 2014 (World Trade Center (WTCA)). L'effectif était de 1100 employés.

- **2016** : Première exportation en Espagne. Sortie de Maghreb Private Equity Fund et entrée de Development Partners International (DPI) et de la Deutsche Dation Investitions und Entwicklungsgesellschaft mbH (DEG) à hauteur de 49% du capital social. Première exportation en Mauritanie. L'effectif était de 1170 employés.
- **2019** : L'entreprise est distinguée comme entreprise "inspirante" pour l'Afrique dans le Rapport "Compagnies to inspire Africa 2019" du London Stock Exchange Group (Bourse de Londres). Première expédition vers la Belgique et première exportation vers la France. Elle remporte le Prix spécial du jury du Trophée Export 2018 (World Trade Center (WTCA)). L'effectif était de 1201 employés.
- **2020** : Général Emballage obtient les certifications ISO 14001 :2015 et ISO 45001 :2018. L'effectif était de 1222 employés.

1.2.4 Activités et missions de Général Emballage

Général Emballage a adopté un système de marketing B2B (business to business), c'est-à-dire que la quasi-totalité de ses clients sont des entreprises et des professionnels. Son portefeuille client englobe donc des secteurs d'activités hétérogènes, comptant parmi eux la quasi-totalité des grandes marques nationales. Parmi ses clients locaux, on peut citer : CEVITAL, SOMMAM, IFRI, CANDIA, LA BELLE, SAMSUNG, BIOM, AVEA, SARL ALMAG, ROUIBA, etc.

La mission de Général Emballage est de satisfaire sa clientèle de plus en plus exigeante en matière d'emballage et de plaques en carton ondulé, en fabriquant, à la commande, des plaques double-face (cannelures B, C, E et F) et double-double (BC et BE), des displays, des emballages de tous types comme les PAV (prêts à vendre), les caisses à fond automatique, les caisses télescopiques et les barquettes à découpe spéciale. L'entreprise réalise également des post-impressions en haute résolution, atteignant 6 couleurs avec vernis intégral ou sélectif, ainsi que le paraffinage du carton destiné aux chambres froides et à l'exportation des fruits et légumes. Elle dispose aussi d'un nombre d'équipes maîtrisant l'ensemble des tâches de production : études, prototypage, réalisation de formes de découpe et de films d'impression, fabrication des emballages et des displays, et notamment la livraison.

Général Emballage possède une vaste infrastructure de collecte et de valorisation de papiers et cartons à recycler (PCR) déployée sur l'ensemble du territoire algérien, possédant son propre centre de collecte de déchets à Alger depuis 2020 [1].

1.2.5 Valeurs de Général Emballage

- **Leadership** : Les politiques d'investissement, de recrutement et de formation reposent sur deux principes fondamentaux : satisfaire la demande et anticiper sur les besoins futur du marché. Il en découle une mise à niveau continuelle des compétences humaines et des processus technologiques.
- **Proximité** : Ils entretiennent le rapprochement avec leurs clients pour une meilleure compréhension de leurs besoins et pour réduire les coûts et les délais d'acheminement de leurs produits, garantissant ainsi le meilleur rapport qualité/prix.
- **Citoyenneté** : Général Emballage est une entreprise citoyenne qui inscrit son intérêt dans celui de la société et de l'humanité en général.
- **Développement Durable** : Général Emballage s'engage à :
 1. Recycler l'ensemble de ses déchets de production et de ses rejets industriels.
 2. Ne se fournir qu'auprès d'industries respectant les principes du Développement durable.
 3. Apporter sa contribution aux efforts visant la préservation de l'environnement, notamment aux actions de reforestation [2].

1.2.6 Organisation de la Spa Général Emballage Unité d'Akbou

La démarche commerciale de la société se focalise, en premier lieu, sur la satisfaction de la demande et la fidélisation de la clientèle en recherchant la notoriété de la qualité de ses produits. Cela impose un système de contrôle strict des processus de production, ainsi que la collaboration avec le monde universitaire et la sphère de la recherche scientifique, et l'anticipation sur les besoins futurs du marché, ce qui permet à l'entreprise de développer continuellement sa capacité de leadership.

L'organisation de l'entreprise est présentée selon l'organigramme illustré par la figure 1.2 [2] :

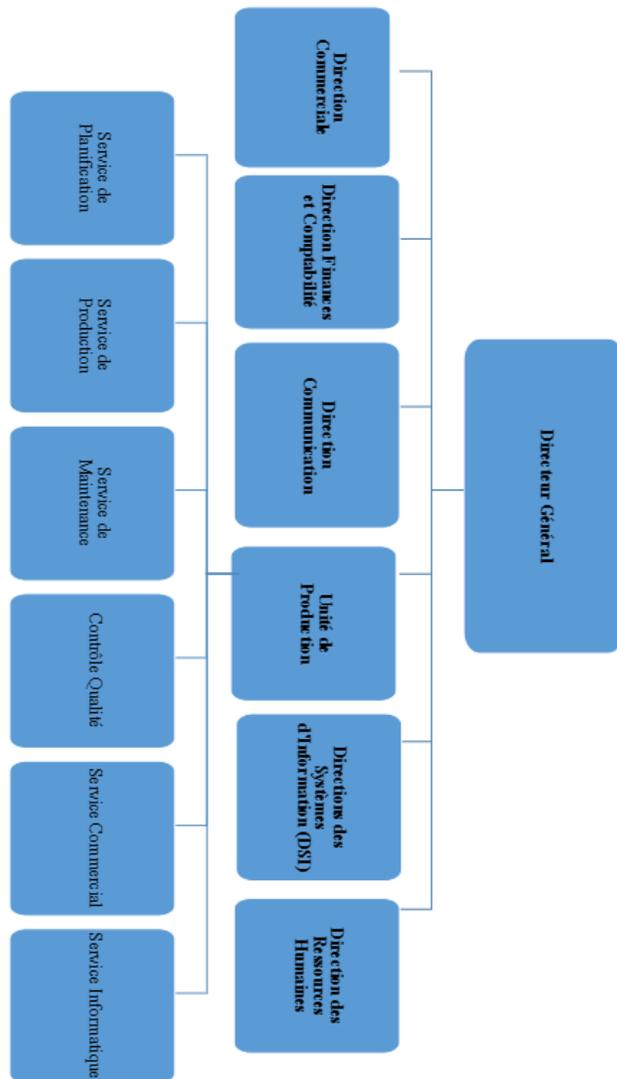


FIGURE 1.2 – Organigramme général de Général Emballage Unité d'Akbou

1.3 Présentation du champ d'étude

Notre travail s'effectuera au sein de la direction commerciale de l'entreprise Général Emballage, plus précisément du côté du suivi des clients, des commerciaux et du recouvrement. Dans cette section, nous allons présenter le champ d'étude de l'organisme d'accueil. Cette partie nous aidera à mieux comprendre le domaine d'étude et à révéler les lacunes du système existant.

1.3.1 Organigramme de la direction commerciale de Général Emballage

Pour mieux situer le champ de notre étude dans l'organisme d'accueil, nous le précisons dans la figure 1.3 de l'organigramme de la direction commerciale de Général Emballage [2].

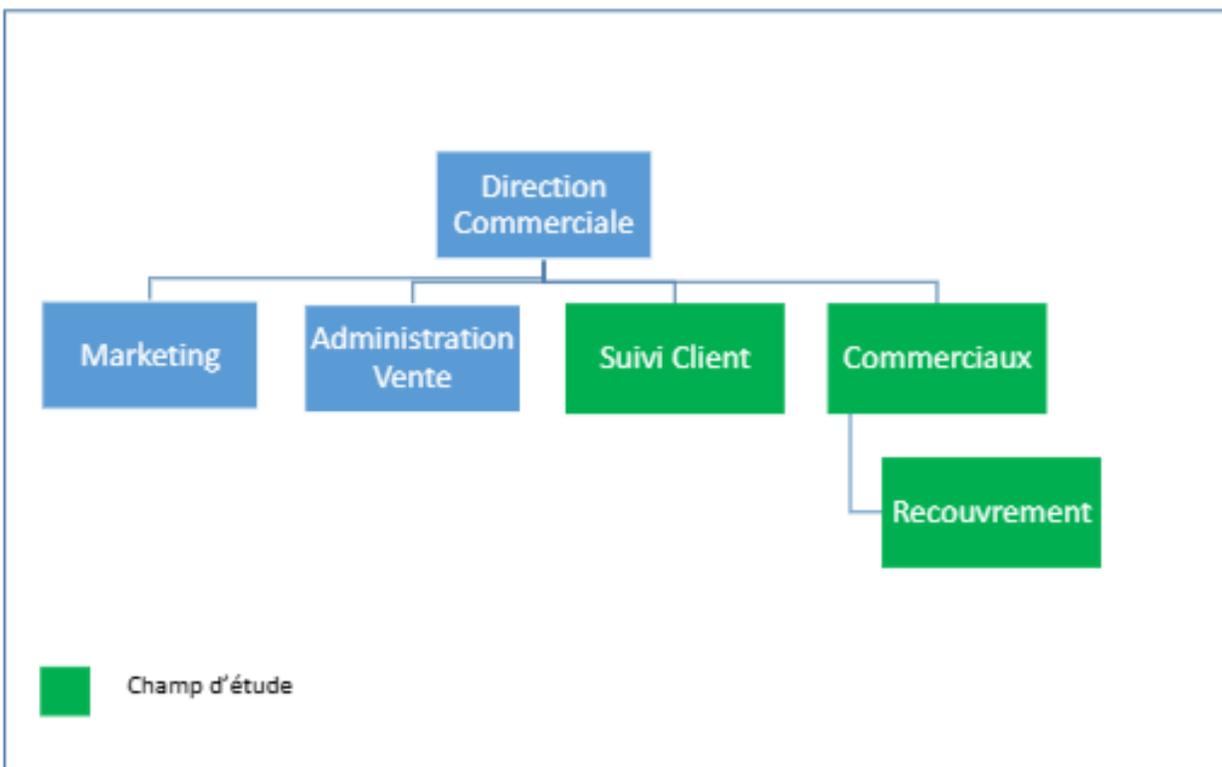


FIGURE 1.3 – Organigramme de la direction commerciale de Général Emballage

1.3.2 Présentation de la direction commerciale

La direction commerciale effectue plusieurs tâches répondant à la fois aux besoins de la clientèle et aux perspectives de l'entreprise. Elle est chargée de la commercialisation des produits finis, et recouvre ainsi tout l'ensemble des activités depuis la prise de la commande jusqu'à la livraison, en étant également en relation avec d'autres services [2].

Service marketing

- Conduit des études de recherches du marché et des produits.
- Initie et suscite les actions d'amélioration de la communication en relation avec l'environnement externe et médiatique de l'entreprise, notamment par les réseaux sociaux et le site web de l'entreprise.
- Participe à la politique de détermination des barèmes de prix.
- Élabore un plan de communication interne et le met en œuvre.

Administration vente

- Organisation et planification de la production.
- Vérification et validation des dossiers de production.

Suivi client

- S'occupe du suivi de la situation financière du client.
- Établissement des factures.

Commerciaux

- Prospection et initiation du contact avec les clients.
- Réception des commandes.
- Suivi de l'état d'avancement des commandes des clients.

Recouvrement

- Gestion des dettes des clients.

1.4 Etude de l'existant

L'étude de l'existant est une étape primordiale dans la compréhension du champ d'étude. Elle nous permet ainsi d'avoir des connaissances détaillées sur le fonctionnement du processus étudié, et d'en dégager d'une manière structurée les informations nécessaires à l'établissement d'un modèle conceptuel cohérent de données. Au cours de cette phase, et suite aux interviews, nous avons recensé les documents utilisés par les différents postes ciblés par notre étude. L'étude des procédures nous a permis de comprendre les flux d'information.

1.4.1 L'informatique chez Général Emballage

Le site web de Général Emballage, www.generalemballages.com, présente l'entreprise, ses services et ses coordonnées.

Général Emballage utilise un système ERP (Enterprise Resource Planning) appelé *Volu-pack*, un logiciel de gestion dédié aux entreprises du packaging, telles que les onduleurs, les cartonneries et les transformateurs. voir figure 1.4

Côté hardware :

La salle serveur est équipée de :

- Un serveur Exchange DELL pour la messagerie interne.
- Un serveur DELL pour l'ERP.
- Un modem D-LINK.
- 8 switches CISCO CATALYST 2960 de 24 ports.
- 2 onduleurs DELL.
- 2 switches CISCO CATALYST 3570.

Chaque poste de travail est équipé de :

- Un ordinateur DELL avec 4 Go de RAM, un processeur de 1,7 GHz et Windows 10.
- Une imprimante Canon MF-420.
- Un télécopieur.

Côté Software :

Tous les processus de vente, y compris les devis, la réalisation des bons de commande, la production, la facturation et la livraison, sont effectués à travers l'ERP *VoluPack*. En parallèle, un logiciel distinct appelé *PCTOP* est dédié à la planification des machines. La figure 1.4 offre une représentation du fonctionnement de ce système ERP [2].

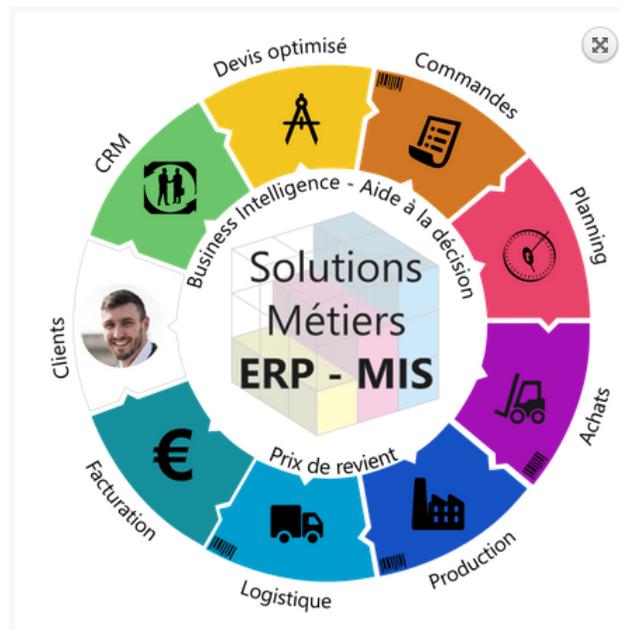


FIGURE 1.4 – ERP VoluPack

1.4.2 Flux de données

Afin de mieux appréhender les besoins et les flux d'informations dans le domaine d'étude, nous avons choisi d'analyser les informations échangées entre divers services et postes, ainsi que les différentes interactions qui les lient.

Le graphe des flux de la figure 1.5 présente un échange d'informations entre deux acteurs. Le flux est représenté par une flèche portant son nom et qui est orientée de l'acteur émetteur vers l'acteur receveur.

Comme nous pouvant le constater à travers notre graphe de flux, nous distinguons quatre acteurs dans notre étude qui échangent différentes informations qui sont données dans le tableau 1.1. Les acteurs sont :

- Le client : acteur externe
- Directeur commercial
- Services commerciaux
- Service Suivi client

Les Flux :

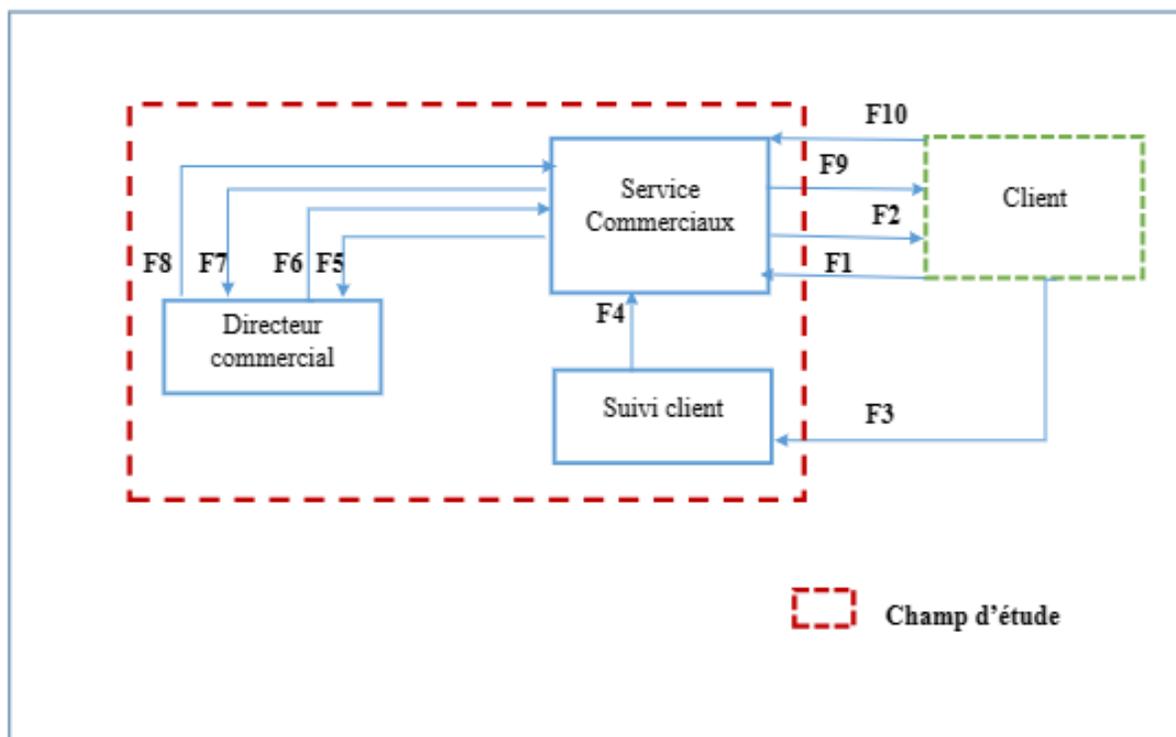


FIGURE 1.5 – Le graphe de flux de la direction commerciale de Général Emballage

| Flux de données | Description |
|-----------------|--|
| F1 | Bon de pré-commande |
| F2 | Offre de devis |
| F3 | Dossier client et le bon de commande |
| F4 | Confirmation de la situation du client et le bon de commande à valider |
| F5 | Bon de commande à valider |
| F6 | Bon de commande validé |
| F7 | Bon de livraison à valider |
| F8 | Bon de livraison validé |
| F9 | Bon de livraison + facture |
| F10 | Chèque |

TABLEAU 1.1 – Flux de données

1.4.3 Etude des services et postes

L'analyse des postes de travail vise à approfondir notre compréhension de l'organisation globale et de la circulation de l'information entre les divers services, ainsi que des procédures administratives en place. Cela nous permettra ensuite d'identifier les postes surchargés, les

lacunes et les anomalies dans la structure organisationnelle actuelle. Les postes examinés dans cette étude comprennent :

- Directeur Commercial.
- Commerciaux.

NB : Tous les postes du service commerciaux et service suivi client sont considérés comme des commerciaux, partageant souvent les mêmes tâches. La différence réside dans le fait que, les commerciaux du service « commerciaux » se déplacent sur le terrain pour la prospection.

1.4.3.1 Fiche d'analyse du poste de travail N° 1 :

| | |
|-------------------------------|---|
| Désignation | Directeur commercial |
| Responsabilité | Superviser l'équipe commerciale |
| Description des tâches | <ul style="list-style-type: none"> — Manager les commerciaux. — Suivi des commandes, du paiement et les réclamations des clients. |
| Documents entrants | <ul style="list-style-type: none"> — Bon de commandes à valider. — Bon de livraison à valider. |
| Provenance | Commercial. |
| Documents sortants | <ul style="list-style-type: none"> — Bon de commande validé. — Bon de livraison validé. |
| Destination | Commercial. |

TABLEAU 1.2 – Fiche d'analyse du poste de travail N° 1.

1.4.3.2 Fiche d'analyse du poste de travail N° 2 :

| Désignation : Commercial | |
|---------------------------------|---|
| Responsabilité | Réception et suivi des commandes |
| Description des tâches | <ul style="list-style-type: none"> — Prospection et initiation du contact avec les clients. — Réception des commandes. — Suivi de l'état d'avancement des commandes des clients. — S'occupe du suivi de la situation financière du client. — Réception des chèques. — Établissement des factures. |
| Documents entrants | <ul style="list-style-type: none"> — Dossier du client. — Chèque de paiement. |
| Provenance | Client. |
| Documents sortants | <ul style="list-style-type: none"> — Bon de commande. — Bon de livraison. |
| Destination | Client. |

TABLEAU 1.3 – Fiche d'analyse du poste de travail N° 2.

1.4.4 Critiques et suggestions

Après avoir examiné attentivement le champ d'étude, plusieurs constats ont été identifiés :

Critique 1 : La prise de commandes se fait actuellement par téléphone ou par email.

Suggestion : Nous proposons la création d'un espace client dédié sur une plateforme web permettant la prise de commandes en ligne et en temps réel, via une application dédiée.

Critique 2 : Le suivi des commandes par les commerciaux est peu organisé, ce qui entraîne un manque d'informations pour les clients concernant l'état d'avancement de leurs commandes. Les clients sont souvent contraints de contacter directement les commerciaux ou le service client pour obtenir des informations.

Suggestion : Pour remédier à cette situation, nous recommandons la mise en place d'un système de suivi détaillé des commandes, avec des notifications automatiques en cas de retard. Cela permettrait d'améliorer la communication avec les clients et d'optimiser la gestion des commandes.

Critique 3 : Les données relatives aux commandes clients et à l'activité des commerciaux sont actuellement mal structurées et difficiles à exploiter pour les cadres dirigeants. Cela entraîne une faible visibilité, des pertes de temps et des retards dans la prise de décision.

Suggestion : Pour résoudre ce problème, nous suggérons la mise en place d'un système décisionnel intégré à l'application web, comprenant un tableau de bord pour suivre efficacement l'activité commerciale de l'entreprise. Cela permettra aux cadres dirigeants d'avoir une vue d'ensemble claire et précise, facilitant ainsi la prise de décision.

En tenant compte des critiques et suggestions émises, notre objectif est d'implémenter un système de gestion de la relation client (CRM) opérationnel en ligne à travers une application web dédiée. Ce CRM est conçu pour maximiser la satisfaction et la fidélisation de la clientèle. Grâce à cette solution, les clients auront la possibilité de passer leurs commandes de manière intuitive, de suivre leur progression, et de soumettre des réclamations en cas de problème ou de manquement. Les commerciaux disposeront des outils nécessaires pour suivre efficacement leurs clients et répondre à leurs demandes de manière proactive.

Il est important de souligner que l'adoption d'une solution CRM représente un véritable investissement pour une entreprise, offrant des avantages à court, moyen et long terme. Cette solution permettra aux dirigeants d'améliorer la fidélisation de leur clientèle et de maintenir une réactivité optimale face à leurs besoins et attentes évolutifs.

1.5 Introduction aux CRM (Gestion Relation Client)

Dans cette section, nous verrons les notions fondamentales concernant les CRM afin de nous familiariser avec cette notion auquel s'intégrera notre futur système.

1.5.1 Définition d'un CRM

La gestion de la relation client, connue sous l'acronyme de CRM (*Customer Relationship Management*) en anglais, combine les technologies informatiques et les stratégies commerciales pour offrir aux clients les produits et les services qu'ils attendent ou qu'ils sont prêts à payer. Ainsi, le CRM est la démarche pour bâtir une relation profitable sur le long terme avec les meilleurs clients en capitalisant sur l'ensemble des points de contact par une allocation optimale des ressources, afin de générer plus de chiffre d'affaires et de bénéfice [3].

Les solutions CRM offrent à tous les interlocuteurs de l'entreprise en contact avec le client des fonctions adéquates et plus particulièrement [4] :

- La fonction marketing (avant-vente) : le CRM permet d'analyser et de tirer profit des flux d'informations tirées des études de marché notamment à travers des outils tels que *Marketing Enterprise Automation*.
- La gestion des ventes : le CRM automatise également la gestion des ventes à travers de nouveaux outils comme *Sale Forces Automation*, qui permettent d'améliorer l'efficacité des vendeurs ou commerciaux.
- La gestion du service (après-vente) : par exemple à travers l'instauration de centres d'appels chargés de l'assistance à distance des clients ou alors les services web. Cela permet à l'entreprise d'être à l'écoute de ses clients et de répondre à leurs besoins.

1.5.2 Les objectifs d'un CRM

Les objectifs du CRM ne sont pas nouveaux, dont les perspectives sont :

- Satisfaire et fidéliser ses clients.
- Attirer de nouveaux clients.
- Accroître le chiffre d'affaires.
- Réduire les coûts de vente et de distribution.

1.5.3 Les types de CRM

Les logiciels de CRM se déclinent principalement en trois types : le CRM opérationnel, le CRM analytique et le CRM collaboratif.

1.5.3.1 Le CRM opérationnel

Le CRM opérationnel rationalise le processus métier qui comprend l'automatisation des ventes, l'automatisation du marketing et l'automatisation des services. Le principal objectif de ce type de CRM est de générer des prospects, de les convertir en contacts, de capturer tous les détails requis et de fournir un service tout au long du cycle de vie du client [5].

- **Automatisation des ventes** : Le but principal de l'automatisation des ventes est d'organiser les informations de manière à ce que l'entreprise puisse répondre aux besoins des clients et augmenter ses ventes plus efficacement. Il comprend divers modules de vente CRM comme la gestion des prospects, la gestion des contacts, etc [5].
- **Automatisation du marketing** : L'objectif principal de l'automatisation du marketing est de trouver la meilleure façon d'offrir des produits et d'approcher des clients potentiels. Le module majeur de l'automatisation du marketing est la gestion des campagnes. Il permet aux entreprises de décider des canaux efficaces (comme les e-mails, les annonces sur les réseaux sociaux, etc.) pour atteindre les clients potentiels [5].
- **Automatisation des services** : L'automatisation des services permet aux entreprises de fidéliser leurs clients en leur offrant la meilleure qualité de service et en établissant des relations solides. Il comprend par exemple la gestion des problèmes pour résoudre les problèmes des clients [5]

1.5.3.2 Le CRM analytique

Le CRM analytique aide le personnel de direction, de marketing, de vente et d'assistance à déterminer la meilleure façon de servir les clients. L'analyse des données est la fonction principale de ce type d'application CRM. Il analyse les données clients, provenant de différents points de contact, pour obtenir de meilleures informations sur l'état actuel d'une organisation. Il aide la direction à prendre de meilleures décisions, les responsables marketing à comprendre l'efficacité de la campagne, les responsables commerciaux à augmenter les ventes et le personnel de support pour améliorer la qualité du support et construire une relation client solide [5].

1.5.3.3 Le CRM collaboratif ou CRM stratégique

Le CRM collaboratif, parfois appelé CRM stratégique, permet à une organisation de partager les informations des clients entre diverses unités commerciales telles que l'équipe de vente, l'équipe marketing, l'équipe technique et de support. Par exemple, les commentaires d'une équipe d'assistance pourraient être utiles pour que l'équipe marketing aborde des clients ciblés avec des produits ou des services spécifiques. Dans le monde réel, chaque unité opérationnelle fonctionne comme un groupe indépendant et partage rarement les données des clients avec d'autres équipes, ce qui entraîne souvent des pertes commerciales. Le CRM collaboratif permet d'unir tous les groupes pour ne viser qu'un seul objectif : utiliser toutes les informations pour améliorer la qualité du service client afin de gagner en fidélité et acquérir de nouveaux clients pour augmenter les ventes [5].

1.5.4 Les modèles de CRM disponible

Les offres de logiciels peuvent être classées en trois catégories principales :

1.5.4.1 Logiciels (commerciaux) sous licence

Il s'agit là de la forme classique des applications logicielles. Les entreprises achètent le logiciel et l'installent sur leurs ordinateurs. Celles dont les salariés ont peu de compétences informatiques et qui ont des exigences spécifiques pour un CRM, devraient opter pour cette variante commerciale classique [6].

- **Avantage** : La société possède le logiciel et peut donc l'adapter à ses besoins. Le fournisseur peut se charger alors de l'installation, de la configuration et de la formation des salariés de l'entreprise. Celle-ci peut ainsi mettre très rapidement en service un logiciel installé de manière optimale [6].
- **Inconvénient** : Les frais engagés sont importants. En effet, l'achat de la licence nécessaire entraîne un paiement unique ou mensuel relativement élevé. La maintenance à intervalles réguliers entraîne également des coûts supplémentaires [6].

1.5.4.2 Programme Open Source

Les logiciels Open Source ne sont pas la propriété d'un seul fabricant et ils ne sont généralement pas vendus. Ils sont le résultat de la coopération de nombreux programmeurs qui développent constamment des applications. Cette variante convient aux entreprises dont

les salariés ont de bonnes connaissances en informatique et peuvent se familiariser seuls avec le nouveau système [6].

- **Avantage** : La majorité des systèmes de ce type sont gratuits et donc moins onéreux. Lorsqu'une solution a une communauté active, elle est constamment améliorée. Les défauts éventuels sont donc rapidement corrigés [6].
- **Inconvénient** : Ces systèmes peuvent également être source de désagréments. En effet, même si l'application proprement dite ne génère aucun coût, les entreprises doivent également investir dans l'installation, qui s'avère plus compliquée que lorsque l'installation est effectuée par le fournisseur d'un programme sous licence, et dans la formation de leurs salariés, ce qui relativise l'avantage financier [6].

1.5.4.3 Offre de Saas (logiciel en tant que service, CRM en ligne)

Dans de nombreux cas, les entreprises n'ont pas besoin de posséder elles-mêmes le logiciel utilisé en interne. L'application est mise à disposition en ligne en tant qu'application dite web, dont le fonctionnement est, en principe, identique à celui du stockage sur le cloud habituel et les utilisateurs y accèdent par le biais d'un compte utilisateur individuel. Cela est intéressant pour les entreprises pour lesquelles il est important que les salariés puissent accéder au CRM sur tous les types d'appareils, à tout moment et à tout endroit. Toutefois, cela n'est intéressant si les économies résultantes du matériel qui n'est pas nécessaire sont conséquentes ou si le CRM est gratuit [6].

- **Avantage** : Le programme proprement dit étant stocké de manière centralisée sur des serveurs externes, le fournisseur peut résoudre les problèmes éventuels sans avoir à intervenir dans l'entreprise. Par ailleurs, il est particulièrement facile pour les salariés en déplacement d'accéder au système partout et à tout moment. Aussi, les solutions SaaS permettent généralement une utilisation hybride, par exemple, que le système de messagerie ne va pas être migré vers le cloud, mais qu'il peut être synchronisé avec le CRM basé sur le cloud [6].
- **Inconvénient** : Un accès internet extrêmement fiable est indispensable, puisque toutes les informations sont sauvegardées en ligne. Cela peut compliquer l'utilisation du logiciel dans de nombreuses régions rurales. Il faut également noter qu'à long terme, les applications SaaS (payantes) sont souvent plus coûteuses qu'une licence individuelle classique, car elles entraînent des frais récurrents. En contrepartie, les utilisateurs bénéficient de mises à jour régulières et d'un support permanent [6].

1.5.5 Contraintes du CRM

Les CRM présentent certaines contraintes lors de leur mise en place, telles que :

- Implication et formation des utilisateurs.
- Grande capacité de changement de l'organisation.
- Ressources financières importantes.
- Délais d'implantation d'un projet CRM considérable [5].

1.5.6 Les avantages du CRM

Les CRM offrent plusieurs avantages dont on peut en tirer plusieurs bénéfices tels que :

- Centraliser l'information.
- Faciliter l'accès à l'information sur les besoins des clients.
- Améliorer la sécurité des données.
- Augmenter la productivité du personnel.
- Gain de temps.
- Maîtriser la relation avec les clients.
- Augmenter les ventes.
- Améliorer l'image de marque de l'entreprise [7].

1.6 Démarche de la Solution proposée

La solution proposée se décompose en trois étapes clés :

1. Collecte d'informations : Dans cette phase, nous avons identifié les parties prenantes de notre futur système CRM, principalement le service de suivi client et les commerciaux. Après des échanges avec les utilisateurs concernés (clients, commerciaux, directeur commercial), nous avons cartographié les flux d'informations et défini leurs besoins. Cette démarche nous a permis de mieux comprendre le processus de commande et d'identifier la problématique principale : comment améliorer le service et le suivi clientèle.

2. Analyse et définition d'objectifs : Nous avons analysé les données collectées lors de la première étape pour élaborer une solution adaptée. La solution retenue consiste à développer une application web de CRM à destination du service client et des commerciaux. Ce CRM permettra de gérer efficacement les commandes et les demandes des clients. Le choix d'une application web de CRM a été motivé par sa capacité à offrir une communication flexible

entre l'entreprise, les clients et les commerciaux, ainsi qu'une accessibilité à tout moment et depuis n'importe quel endroit. Les objectifs principaux de ce CRM sont :

- Faciliter le processus de commande pour les clients.
- Assurer un suivi détaillé des commandes et des paiements, accessible aux clients et aux commerciaux.
- Améliorer la communication entre les clients et leurs commerciaux.

3. Conception et développement du système : Cette étape consiste à concevoir et à mettre en œuvre l'application web de gestion de la relation client. L'application sera déployée sur un extranet et hébergée sur le serveur web de l'entreprise. Chaque utilisateur disposera d'un espace dédié avec une interface conviviale et intuitive, accessible via un navigateur web.

Le développement se déroulera en deux étapes principales :

Étape 1 : Création des fonctions

Cette étape consiste à fournir une plateforme pour les utilisateurs qui pourront se connecter pour effectuer les transactions suivantes :

- Création des comptes des clients/commerciaux/directeur commercial.
- Prise des commandes par les clients, modification ou annulation de celles-ci.
- Suivi des commandes et du paiement des clients par les commerciaux.
- Validation des commandes par le directeur commercial.
- Messagerie entre les clients et les commerciaux.
- Sondage pour le client.
- Réception des notifications signalées par le système.

Étape 2 : Traitement des données

Dans cette étape, nous allons procéder à l'analyse des données générées par les différentes transactions effectuées, notamment :

- Les commandes des clients et leur suivi.
- Le retour des clients à travers les sondages.
- L'activité des commerciaux, c'est-à-dire des informations sur leurs connexions à l'application, sur les opérations de suivi, le nombre de leurs clients, etc.

1.7 Conclusion

Dans ce chapitre, nous avons réalisé une étude approfondie de notre organisation d'accueil, en mettant l'accent particulier sur l'unité commerciale. Nous avons examiné en détail les postes de travail ainsi que les documents en circulation, ce qui nous a permis d'identifier les lacunes et les défis du système actuel. En réponse à ces observations, nous avons élaboré une stratégie visant à résoudre ces problèmes.

Dans le chapitre suivant, nous aborderons la phase d'analyse et d'étude des besoins pour la solution proposée.

Chapitre 2

Étude des besoins et Analyse

2.1 Introduction

Dans ce chapitre, nous allons étudier et analyser les besoins concrets des utilisateurs en utilisant une approche de modélisation pour élaborer notre application, basée sur l'UML. Nous commencerons par la phase d'analyse en identifiant les acteurs et les exigences de chaque utilisateur. Ensuite, nous passerons à la modélisation des diagrammes de contexte et des cas d'utilisation, accompagnée de descriptions textuelles détaillées pour chacun de ces cas.

Par ailleurs, nous avons adopté la méthode agile XP (Extreme Programming) pour garantir une flexibilité et une réactivité optimales tout au long du processus de développement.

2.2 Présentation du langage UML (Unified Modeling Language)

Pour exprimer l'analyse et la conception de notre application on a opté pour UML qui est un langage de modélisation graphique et textuel conçu pour comprendre et définir des exigences, spécifier et documenter des systèmes, dessiner des architectures logicielles, concevoir des solutions et communiquer des idées.

UML modélise toutes les données et tous les processus en élaborant de différents diagrammes. UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles.

1. Les diagrammes structurels

Ces diagrammes, au nombre de six, ont vocation à représenter l'aspect statique d'un système, ces derniers sont diagramme de classe, diagramme d'objet, diagramme de composant, diagramme de déploiement, diagramme de paquetage et diagramme de structure composante.

2. Les diagrammes de comportement

Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par UML, diagramme de cas d'utilisation, diagramme d'état-transition, diagramme d'activités, diagramme de séquence, diagramme de communication, diagramme global d'interaction et diagramme de temps.

Pour la réalisation de notre projet on a opté pour la réalisation des diagrammes de cas d'utilisation, diagrammes de séquence, et les diagrammes de classes.

2.3 Méthode agile XP

Nous avons opté pour la méthode XP car elle met l'accent sur la satisfaction du client, la qualité du code et favorise la programmation en binôme.

2.3.1 Définition de la méthode agile

La méthode agile est une méthodologie de gestion de projet ouverte au changement, qui consiste à décomposer votre projet en une série de petits objectifs réalisables dans le but de développer progressivement un produit de haute qualité. Elle est plus flexible par rapport aux méthodes traditionnelles, centré sur les besoins du client. Cette dernière permet de gérer des projets de développement informatique. La méthode agile devient de plus en plus courante. Ce succès s'explique notamment par sa capacité à s'adapter à divers projets dans tous les secteurs [8, 9].

2.3.2 Définition de XP

L'eXtreme Programming (XP) est une méthode de gestion de projet qui suit les principes de l'agilité à l'extrême, elle vise la rapidité et la simplicité avec des cycles courts de développement et une documentation réduite, elle est centrée sur les besoins du client, elle se divise en sprints et elle a un processus itératif [10, 11].

2.3.3 Valeurs de XP

XP repose sur cinq valeurs fondamentales [10, 12] :

- **Communication** : La communication avec le client et les autres membres de l'équipe du projet permet d'identifier et de corriger les problèmes plus rapidement.
- **SimPLICITÉ** : XP met l'accent sur la simplicité et se concentre uniquement sur le développement des fonctionnalités nécessaires à un moment précis.
- **Feedback** : Les critiques des clients doivent être prises en compte afin d'adapter le processus en conséquence.
- **Respect** : Le respect favorise une bonne entente au sein de l'équipe et améliore sa performance.
- **Courage** : Le courage est une qualité essentielle pour faire face aux défis du changement et pour atteindre les objectifs fixés.

2.3.4 Cycle de vie XP

La méthode de développement logiciel XP suit un cycle de vie itératif et incrémental qui se résume comme suit :

1. **Exploration** : Cette phase consiste à définir les besoins du client et les solutions techniques possibles pour le système. Les développeurs évaluent les user stories et les performances des architectures envisagées [11].
2. **Planning** : Cette phase consiste à planifier le premier release (il est en général programmé pour deux à six mois plus tard) en se concentrant sur les fonctionnalités essentielles [13].
3. **Itérations jusqu'à la 1ère release** : Cette phase consiste à développer la première version de l'application par itérations courtes et testées. La première itération définit l'architecture du système. Les itérations permettent de contrôler le respect du planning et de communiquer sur l'avancement du projet [13].
4. **Mise en production** : Au cours de cette phase, des tests sont effectués pour s'assurer que le logiciel est prêt à être utilisé en production [14, 15].
5. **Maintenance** : Cette phase vise à garantir que le système fonctionne bien et est utilisé correctement et que les fonctionnalités supplémentaires sont développées, sans affecter les anciennes [13].
6. **Mort** : C'est la phase finale de XP, elle survient lorsque les clients ont épuisé toutes les user stories à réaliser et que tous leurs besoins ont été satisfaits [14, 15].

2.4 Démarches et Méthodologie XP

Afin d'aboutir au développement de meilleures applications, il est nécessaire d'avoir une meilleure maîtrise et organisation du travail et donc suivre une démarche méthodologique rigoureuse. Pour cela le choix d'un formalisme de conception est d'une très grande importance. Pour le développement de notre application, nous avons opté pour une démarche de conception orientée objet, en nous basant sur la modélisation UML.

La démarche s'articule autour de deux étapes : l'analyse des besoins et la conception.

- En phase d'analyse des besoins, nous mettons en évidence les exigences et les acteurs du futur système.
- En phase conception, nous présenterons les descriptions détaillées des résultats de l'analyse.

La figure suivante montre la représentation graphique de la démarche de modélisation que nous avons choisie pour concevoir notre application.

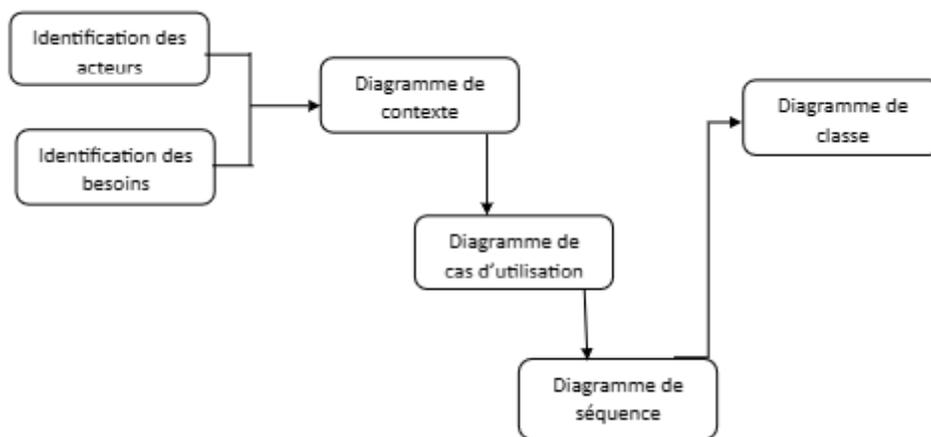


FIGURE 2.1 – Représentation de la Démarches et Méthodologie XP.

2.5 Identification des acteurs

2.5.1 Définition d'un acteur :

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Ainsi, il peut

consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

2.5.2 Les acteurs de notre système :

L'Administrateur : Il est responsable de la gestion des comptes utilisateurs de l'application.

Le Directeur Commercial : Il est chargé de la gestion des commandes dans le système.

Les Commerciaux : Ils sont responsables du portefeuille clients.

Le Client : Il s'agit de l'utilisateur du système qui utilise l'application pour passer des commandes, interagir avec les commerciaux et accéder à d'autres fonctionnalités offertes par le système.

2.6 Spécification des exigences

On trouve deux types d'exigences :

- Les exigences fonctionnelles.
- Les exigences non-fonctionnelles.

2.6.1 Exigences fonctionnelles

Dans cette section, nous présentons l'ensemble des besoins fonctionnels auxquels devrait répondre notre application web. Les exigences et les attentes par rapport à notre application dépendent de la nature de l'acteur. Pour cela, nous avons décrit pour chaque acteur les besoins fonctionnels qui lui sont reliés qui peuvent être résumés comme suit. Le système doit permettre :

Au directeur commercial

- De s'identifier.
- Consulter les notifications signalées par le système (nouvelle commande, retard de suivi).
- Consulter les commandes.
- Valider les commandes.
- Consulter la liste des clients et des commerciaux.

Aux commerciaux

- De s'identifier.
- De faire le suivi des commandes de ses clients, des montants payés et des montants restants.
- D'ajouter/modifier ou supprimer ses clients.
- D'ajouter/modifier ou supprimer les produits de ses clients.
- Annuler/modifier une commande d'un client.
- Recevoir et consulter les notifications signalées par le système (nouvelle commande, nouveau message d'un client).
- De contacter ses clients.

Aux clients

- De s'identifier.
- De passer/modifier ou annuler une commande (modifier ou supprimer avant un délai précis).
- De consulter l'état d'avancement de sa commande et des montants payés et des montants restants.
- Recevoir et consulter les notifications signalées par le système (nouvel état de commande, nouveau message de son commercial).
- Donner son avis à travers un sondage.
- De contacter son commercial.

A l'administrateur

- De s'identifier.
- D'ajouter/Modifier/Supprimer des utilisateurs (commerciaux/directeur commercial).
- D'ajouter/Modifier/Supprimer des catégories de produits.

2.6.2 Exigences non fonctionnelles

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation comme le langage de programmation, le type du SGBD et du système d'exploitation. Le système à réaliser doit respecter les besoins suivants :

Ergonomie de l'interface : L'application doit être facile à utiliser, avec des interfaces utilisateur conviviales, simples, ergonomiques et adaptées à l'utilisateur.

Fiabilité : Les informations fournies par l'application doivent être fiables et sûres.

Disponibilité : L'application doit être disponible à tout moment pour être utilisée par n'importe quel utilisateur.

Sécurité : L'application contient des informations personnelles et sensibles qui ne doivent pas être accessibles à tout le monde. Elle doit donc respecter les règles relatives à la sécurité des systèmes informatiques.

Performance : L'application doit être performante, en répondant de manière optimale à toutes les exigences des utilisateurs à travers ses fonctionnalités.

2.7 Diagramme de contexte

Le diagramme de contexte est un modèle conceptuel qui permet d'avoir une vue globale des interactions entre le système et l'environnement extérieur. Il permet aussi de bien délimiter le domaine d'application. La figure suivante montre le diagramme de contexte de notre application :

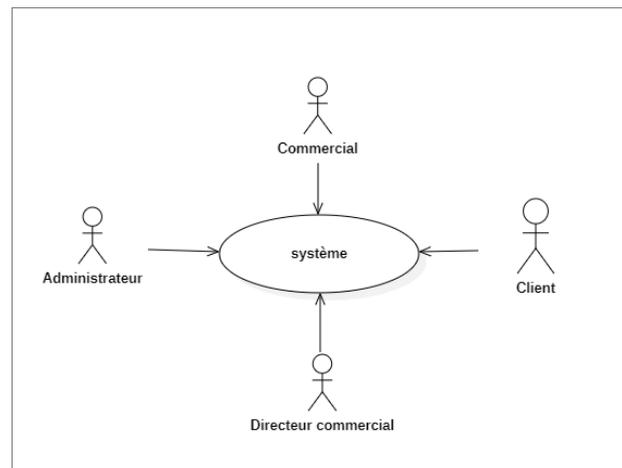


FIGURE 2.2 – Diagramme de contexte

2.8 Modélisation fonctionnelle

Après avoir identifié les acteurs qui interagissent avec le système, nous allons développer un premier modèle UML de haut niveau, pour pouvoir établir précisément les frontières du système. Dans cette optique, nous allons identifier les cas d'utilisation et construire un

diagramme reliant les acteurs et les cas d'utilisation. Ensuite, nous précisons le point de vue fonctionnel détaillé en représentant les interactions entre objets et en indiquant la chronologie des échanges. À cet effet, nous allons établir les diagrammes de séquences.

Pour la modélisation de notre travail, nous avons utilisé trois diagrammes ; le diagramme de classe pour modéliser l'aspect statique et les diagrammes de cas d'utilisation et de séquence pour modéliser l'aspect dynamique.

2.8.1 Spécification des tâches pour chaque acteur

Chaque acteur de notre système joue un rôle particulier et ce tableau ci-dessous nous le montre.

| Acteur | Tâches |
|----------------------|--|
| Directeur commercial | T1 : s'authentifier. T2 : consulter les notifications signalées par le système. T3 : consulter les commandes. T4 : valider les commandes. T5 : consulter la liste des clients et des commerciaux. T6 : se déconnecter. |
| Commercial | T7 : s'authentifier. T8 : gérer les comptes des clients. T9 : gérer les produits. T10 : modifier une commande. T11 : annuler une commande. T12 : faire le suivi des commandes de ses clients et de leurs paiements. T13 : faire le suivi de l'état de chaque produit d'une commande. T14 : consulter les notifications signalées par le système. T15 : contacter ses clients. T16 : se déconnecter. |
| Client | T17 : s'authentifier. T18 : lancer une commande. T19 : modifier une commande. T20 : annuler une commande. T21 : suivre l'état de sa commande et du paiement. T22 : suivre l'état de chaque produit commandé. T23 : donner son avis à travers un sondage. T24 : consulter les notifications signalées par le système. T25 : contacter son commercial. T26 : se déconnecter. |
| Administrateur | T27 : s'authentifier en tant qu'administrateur. T28 : gérer les comptes commerciaux/dirigeants. T29 : gérer les catégories de produits. T30 : se déconnecter. |

TABLEAU 2.1 – Spécification des tâches pour chaque acteur du système

2.8.2 Diagramme de cas d'utilisation

Ce diagramme vise à représenter les besoins des utilisateurs par rapport au système, et il est l'un des diagrammes les plus structurants dans l'analyse d'un système.

Il implique les concepts suivants :

Acteur : Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié ainsi il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Cas d'utilisation (use case) : Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. Il modélise un service rendu par le système en exprimant les interactions entre l'acteur et le système, ajoutant ainsi de la valeur à l'acteur concerné.

Relations entre les acteurs : La seule relation entre les acteurs est la relation de généralisation. Quand un acteur fils hérite d'un acteur père, il hérite en réalité de toutes les associations du père.

Relations entre cas d'utilisation : Pour clarifier un diagramme, UML permet d'établir des relations entre les cas d'utilisation. Il existe principalement deux types de relations : les dépendances stéréotypées et la généralisation/spécialisation.

Les dépendances stéréotypées sont des dépendances dont la portée est explicitée par le nom du stéréotype. Les stéréotypes les plus utilisés sont l'inclusion et l'extension.

- **La relation d'inclusion** : Un cas A est inclus dans un cas B si le comportement décrit par le cas A est inclus dans le comportement du cas B ; on dit alors que le cas B dépend de A. Cette dépendance est symbolisée par le stéréotype *include*.
- **La relation d'extension** : Si le comportement de B peut être étendu par le comportement de A, on dit alors que A étend B. Une extension est souvent soumise à condition. Cette dépendance est symbolisée par le stéréotype *extend*.
- **La relation de généralisation** : Un cas A est une généralisation d'un cas B si B est un cas particulier de A. Cette relation de généralisation/spécialisation est présente dans la plupart des diagrammes UML et se traduit par le concept d'héritage dans les langages orientés objet.

2.8.3 Identification des cas d'utilisation

L'identification des cas d'utilisation consiste à décrire comment un utilisateur peut interagir avec le système pour accomplir un objectif précis, ou détailler les fonctionnalités attendues du système. Au cours de notre analyse, nous avons repéré les actions essentielles pour chaque acteur, comme présenté dans le tableau suivant :

| Acteur | Cas d'utilisation |
|----------------------|---|
| Utilisateurs | Authentification |
| Administrateur | Gestion des utilisateurs (consulter la liste des utilisateurs , chercher, ajouter, supprimer ou modifier un compte) |
| Administrateur | Gestion des catégories (gérer les catégories ajouter, modifier, supprimer) |
| Commercial | Gestion des commandes des clients (consulter les commandes des clients, chercher, modifier, annuler, faire le suivi de chaque commande et de chaque produit commandé) |
| Commercial | Suivi des produits commandés |
| Commercial | Gestion des produits (Ajouter, modifier, supprimer un produit) |
| Commercial | Gestion des clients (Ajouter, modifier, supprimer un client) |
| Client | Liste des commandes clients (suivre l'état d'avancement de chacune de ses commandes) |
| Client | Gestion des états des produits commandés (suivre l'état d'avancement de chaque produit commandé) |
| Client | Répondre au sondage |
| Directeur commercial | Consulter la liste des utilisateurs |
| Directeur commercial | Consulter la liste des clients |
| Directeur commercial | Consulter et valider les commandes |

TABLEAU 2.2 – Acteurs et cas d'utilisation

2.8.4 Diagramme de cas d'utilisation de « acteur : Administrateur »

Le diagramme ci-dessus représente les tâches spécifiées à un administrateur :

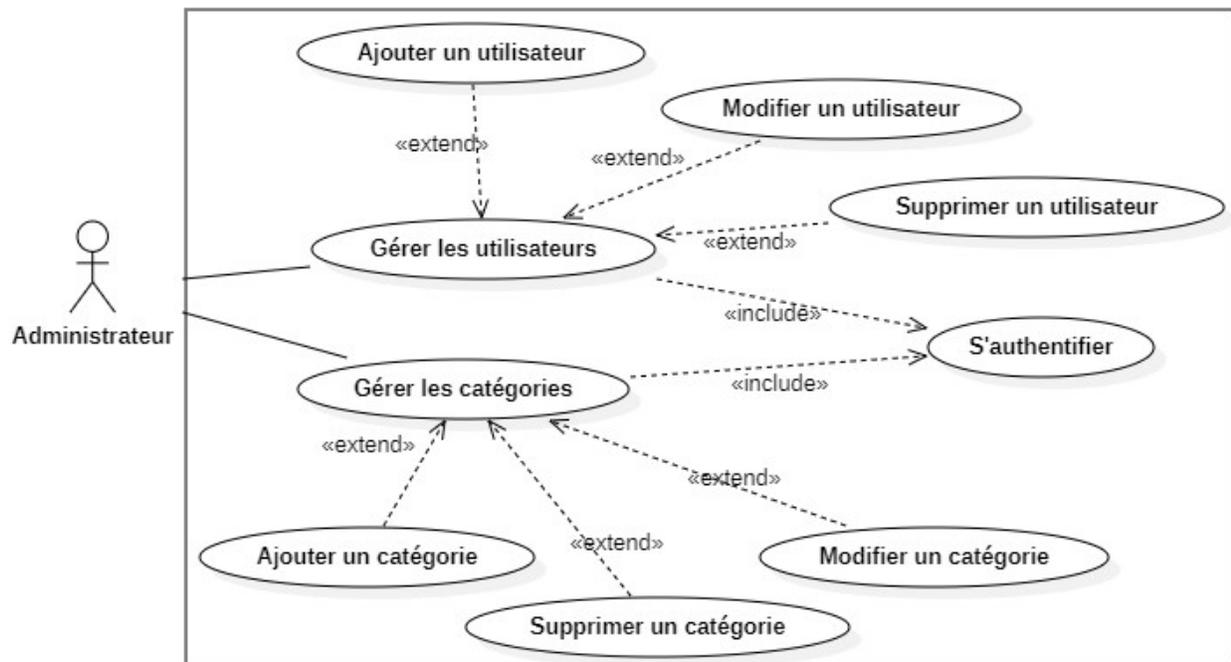


FIGURE 2.3 – Diagramme de cas d'utilisation de « acteur : Administrateur »

2.8.5 Diagramme de cas d'utilisation de « acteur : Commercial »

Le diagramme ci-dessous représente les cas d'utilisation identifiés pour le Commercial :

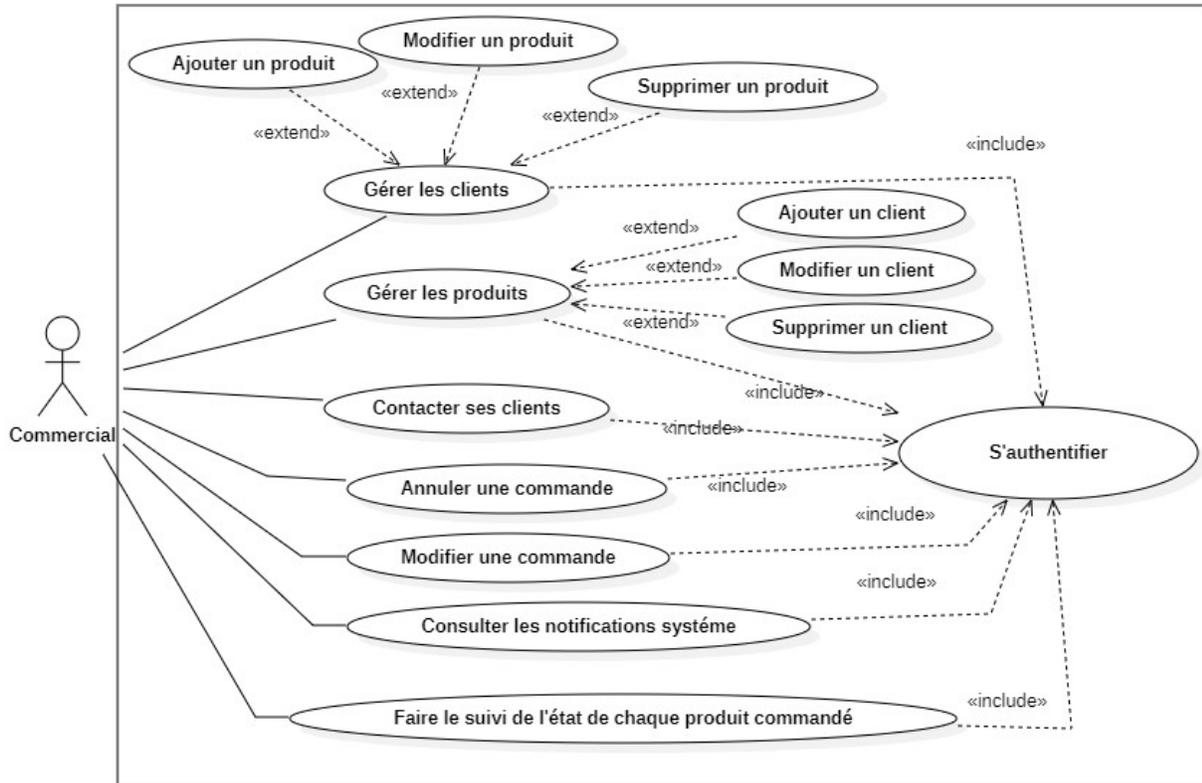


FIGURE 2.4 – Diagramme de cas d'utilisation de « acteur :Commercial»

2.8.6 Diagramme de cas d'utilisation de « acteur : Client »

Le diagramme suivant représente les cas d'utilisation associés à un client. Ce dernier peut réaliser toutes les tâches représentées dans la figure ci-dessous :

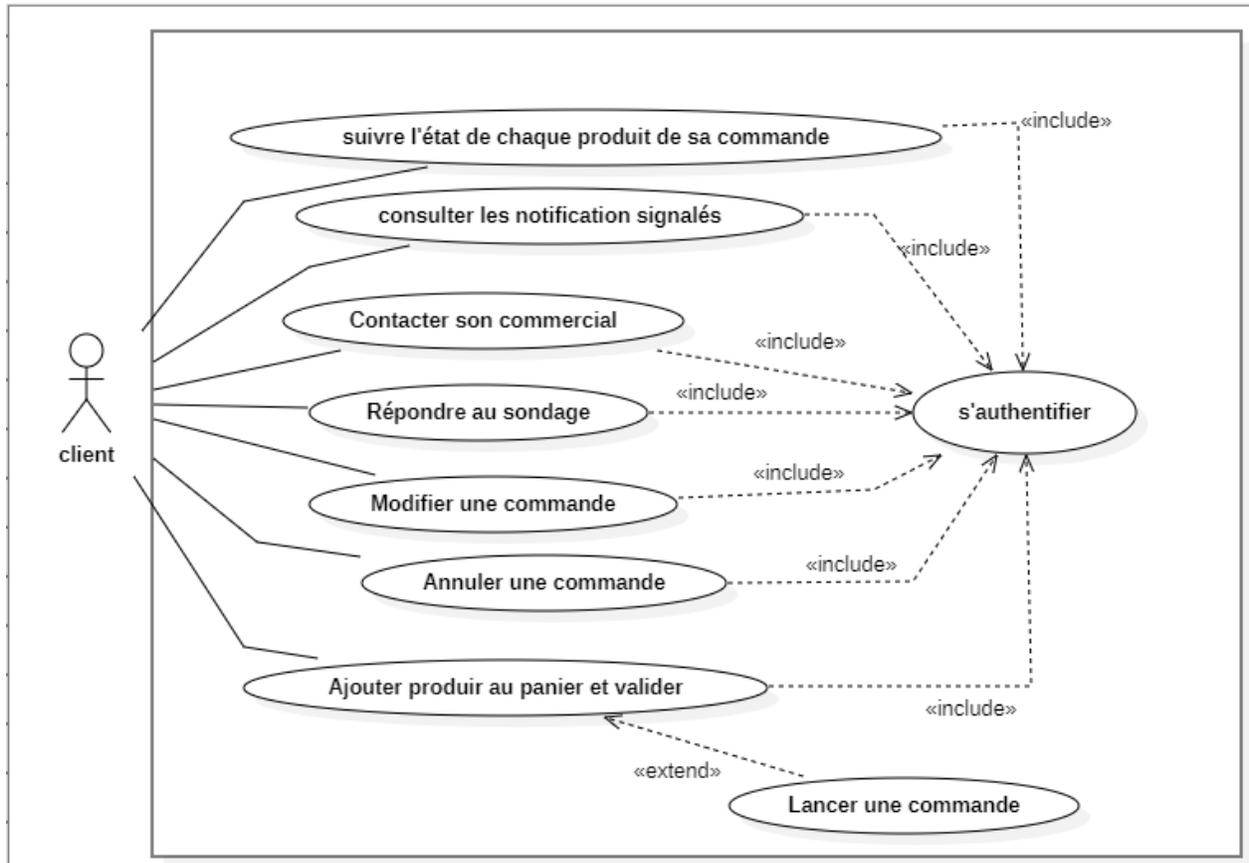


FIGURE 2.5 – Diagramme de cas d'utilisation de « acteur : Client »

2.8.7 Diagramme de cas d'utilisation de « acteur : Directeur commercial »

Le diagramme suivant représente les différentes tâches réalisées par le directeur commercial :

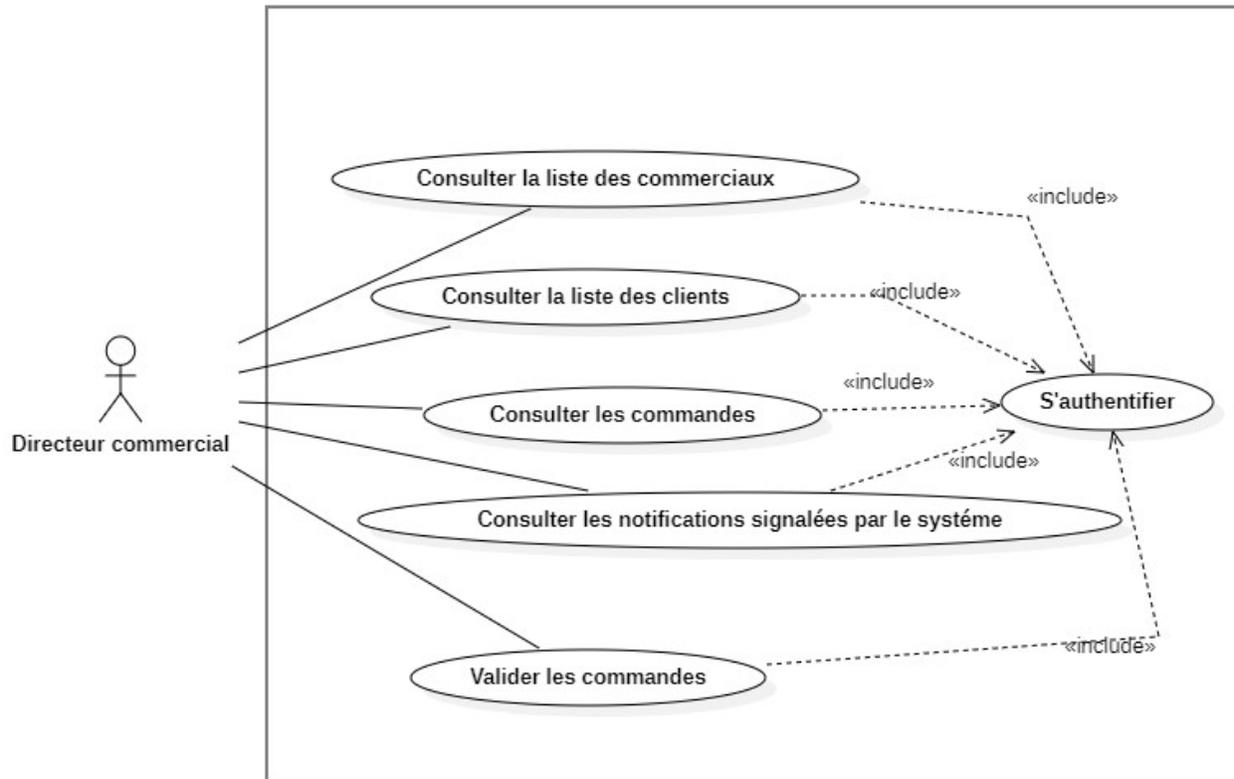


FIGURE 2.6 – Diagramme de cas d'utilisation de « acteur : Directeur commercial »

2.8.8 Description textuelle des cas d'utilisation précédents

À chaque cas d'utilisation doit être associée une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs. La description textuelle d'un cas d'utilisation est articulée en deux parties :

- Première partie permet d'identifier le cas. Elle doit contenir ; le nom du cas, son objectif qui décrit succinctement le contexte et les résultats attendus du cas d'utilisation et les acteurs concernés par le cas en précisant globalement leur rôle.
- La deuxième partie contient la description du fonctionnement du cas sous la forme d'une séquence de messages échangés entre les acteurs et le système. Elle contient une séquence nominale qui correspond au fonctionnement nominal du cas. Cette séquence se développe en trois points :

Pré conditions Si certaines conditions particulières sont requises avant l'exécution du cas, elles sont à exprimer à ce niveau.

Post conditions Par symétrie, si certaines conditions particulières doivent être réunies après l'exécution du cas, elles sont à exprimer à ce niveau.

Scénario nominal Il s'agit là du scénario principal qui doit se dérouler sans incident et qui permet d'aboutir au résultat souhaité.

Scénarios alternatifs Les autres scénarios, secondaires ou correspondant à la résolution d'anomalies, sont à décrire à ce niveau.

Les figures suivantes représentent une description de certains cas d'utilisation de notre système :

Authentification

Ce cas d'utilisation permet à l'utilisateur d'accéder à son compte, et au système de gérer les droits d'accès et d'affichage. Le tableau ci-dessous représente sa description textuelle :

| | |
|----------------------------|---|
| Acteur principal | Administrateur/ Client/Directeur Commercial/ Commercial |
| Objectifs | S'authentifier pour pouvoir accéder à toutes les fonctionnalités de l'application |
| Pré-conditions | L'utilisateur doit avoir un compte. |
| Post-conditions | Accès à la page d'accueil de l'utilisateur. |
| Scénario nominal | <ol style="list-style-type: none"> 1. Demande d'accès au système. 2. Le système répond par l'affichage de la page d'authentification. 3. L'utilisateur saisit son email et son mot de passe. 4. Le système vérifie les champs. 5. Le système vérifie l'existence de l'utilisateur : <ul style="list-style-type: none"> - Si l'utilisateur existe dans la base de données alors il va réussir à s'authentifier, sinon le système renvoie un message d'erreur (l'utilisateur n'existe pas). - Si l'authentification est réussie, il aura accès à la page d'accueil, sinon le système réinitialise la page d'authentification. |
| Scénario alternatif | <ul style="list-style-type: none"> -Champ manquant : Connexion échouée, message d'erreur demandant de compléter les champs requis. -Information incorrecte : Connexion échouée, message d'erreur indiquant des identifiants invalides |

TABLEAU 2.3 – Description de cas d'utilisation « Authentification »

Gestion des catégories

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Gestion des catégories »

| | |
|----------------------------|--|
| Acteur principal | Administrateur |
| Objectifs | Gérer les catégories. |
| Pré-conditions | L'administrateur doit s'authentifier. |
| Post-conditions | -Les modifications apportées par l'administrateur sont enregistrées et la liste des catégories est mise à jour en conséquence. -La base de données reflète les modifications effectuées dans le système. |
| Scénario nominal | <ol style="list-style-type: none"> 1. L'administrateur accède à l'espace "Gestion des catégories". 2. Le système affiche la liste des catégories. 3. L'administrateur peut effectuer les actions suivantes : <ul style="list-style-type: none"> — Ajouter une catégorie : <ol style="list-style-type: none"> (a) L'administrateur clique sur le bouton "Ajouter catégorie". (b) Le système affiche le formulaire d'ajout. (c) L'administrateur remplit le formulaire d'ajout de catégorie. (d) L'administrateur clique sur le bouton "Valider". (e) Le système ajoute la catégorie. (f) Le système affiche la liste des catégories mise à jour. — Modifier une catégorie : <ol style="list-style-type: none"> (a) L'administrateur sélectionne la catégorie qu'il souhaite modifier. (b) Le système affiche le formulaire de modification. (c) L'administrateur remplit le formulaire de modification de catégorie. (d) L'administrateur clique sur le bouton "Valider". (e) Le système modifie la catégorie. (f) Le système affiche la liste des catégories mise à jour. — Supprimer une catégorie : <ol style="list-style-type: none"> (a) L'administrateur sélectionne la catégorie qu'il souhaite supprimer. (b) Le système affiche un message de confirmation de suppression. (c) L'administrateur clique sur le bouton "Confirmer". (d) Le système supprime la catégorie. (e) Le système affiche la liste des catégories mise à jour. |
| Scénario alternatif | Erreur lors de (l'ajout ou modification ou suppression) d'une catégories : Une erreur est survenue lors de (l'ajout ou modification ou suppression) d'une catégories. Le système affiche un message d'erreur indiquant la nature de l'erreur. L'administrateur doit corriger l'erreur et recommencer (l'ajout ou modification ou suppression) d'une catégories. |

TABLEAU 2.4 – Description de cas d'utilisation « Gestion des catégories »

Gestion des clients

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Gestion des clients »

| | |
|----------------------------|--|
| Acteur principal | Commercial |
| Objectifs | Permettre la Gestion des clients . |
| Pré-conditions | Le commercial est authentifié. |
| Post-conditions | <ul style="list-style-type: none"> — La liste des clients est affichée. — Le Commercial a effectué l'action souhaitée (ajouter, modifier, ou supprimer un client). |
| Scénario nominal | <ol style="list-style-type: none"> 1. Le commercial accède à l'espace "Gestion des clients". 2. Le système affiche la liste des clients. 3. Le commercial peut effectuer les actions suivantes : 4. (Ajouter/ modifier/ supprimer) un client : <ol style="list-style-type: none"> (a) Le commercial clique sur le bouton "(Ajouter/ modifier/ supprimer) un client". (b) Le système affiche le formulaire (d'ajout/ modification/ suppression) de client. (c) Le commercial remplit le formulaire (d'ajout/ modification/ suppression) de client. (d) Le commercial clique sur le bouton "Valider". (e) Le système (Ajouter/ modifier/ supprimer) le client. (f) Le système affiche la liste des clients mise à jour. |
| Scénario alternatif | Erreur lors de (l'ajout ou modification ou suppression) d'un client : Une erreur est survenue lors de (l'ajout ou modification ou suppression) d'un client. Le système affiche un message d'erreur indiquant la nature de l'erreur. Le responsable doit corriger l'erreur et recommencer (l'ajout ou modification ou suppression) du client. |

TABLEAU 2.5 – Description de cas d'utilisation « Gestion des clients »

Gestion des produits

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Gestion des produits»

| | |
|----------------------------|--|
| Acteur principal | Commercial |
| Objectifs | Permettre la Gestion des produits . |
| Pré-conditions | Le commercial est authentifié. |
| Post-conditions | <ul style="list-style-type: none"> — La liste des produits est affichée. — Le Commercial a effectué l'action souhaitée (ajouter, modifier, ou supprimer un produit). |
| Scénario nominal | <ol style="list-style-type: none"> 1. Le commercial accède à l'espace "Gestion des produits". 2. Le système affiche la liste des produits. 3. Le commercial peut effectuer les actions suivantes : 4. (Ajouter/ modifier/ supprimer) un produit : <ol style="list-style-type: none"> (a) Le commercial clique sur le bouton "(Ajouter/ modifier/ supprimer) un produit". (b) Le système affiche le formulaire (d'ajout/ modification/ suppression) de produit. (c) Le commercial remplit le formulaire (d'ajout/ modification/ suppression) de produit. (d) Le commercial clique sur le bouton "Valider". (e) Le système (Ajouter/ modifier/ supprimer) le produit. (f) Le système affiche la liste des produits mise à jour. |
| Scénario alternatif | Erreur lors de (l'ajout ou modification ou suppression) d'un produit : Une erreur est survenue lors de (l'ajout ou modification ou suppression) d'un produit. Le système affiche un message d'erreur indiquant la nature de l'erreur. Le responsable doit corriger l'erreur et recommencer (l'ajout ou modification ou suppression) du produit. |

TABLEAU 2.6 – Description de cas d'utilisation « Gestion des produits »

Liste des commandes clients

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Liste des commandes clients ».

| | |
|-------------------------|--|
| Acteur principal | Client |
| Objectifs | Permettre au client de suivre l'état d'avancement de ses commandes, ou les annuler si nécessaire. |
| Pré-conditions | Le client est authentifié et a passé au moins une commande. |
| Post-conditions | <ul style="list-style-type: none"> — Le client a pu suivre l'état d'avancement de ses commandes. — Le client a eu la possibilité d'annuler ses commandes selon les politiques en vigueur. — Le client a pu joindre le bon de commande au besoin. |
| Scénario nominal | <ol style="list-style-type: none"> 1. Le client accède à l'espace "Liste des commandes". 2. Le système affiche la liste des commandes passées par le client, avec leur état d'avancement actuel. 3. Le client peut effectuer les actions suivantes pour chaque commande : <ul style="list-style-type: none"> — Suivre l'état d'avancement de la commande : <ol style="list-style-type: none"> (a) Le client sélectionne la commande qu'il souhaite suivre. (b) Le système affiche les informations concernant l'état actuel de la commande, telles que la date de commande, la date estimée de livraison et l'état de traitement. — Annuler une commande : <ol style="list-style-type: none"> (a) Le client sélectionne la commande qu'il souhaite annuler. (b) Le système affiche un message de confirmation de l'annulation, indiquant les conditions éventuelles liées à l'annulation. (c) Le client confirme l'annulation en cliquant sur le bouton "Annuler". (d) Le système annule la commande et met à jour son statut. |

TABLEAU 2.7 – Description du cas d'utilisation « Liste des commandes clients »

Gestion des états des produits commandés

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Gestion des états des produits commandés ».

| | |
|----------------------------|--|
| Objectif | Permettre aux utilisateurs de suivre l'état d'avancement de chaque produit commandé. |
| Acteur principal | client |
| Pré-conditions | Le client est authentifié . |
| Post-conditions | <ul style="list-style-type: none"> — Le client a pu suivre l'état d'avancement de chaque produit commandé. — Les informations sur l'état des produits ont été mises à jour selon les actions effectuées. |
| Scénario nominal | <ol style="list-style-type: none"> 1. Le client accède à l'interface de gestion des états des produits commandés. 2. Le système affiche la liste des produits commandés, avec leur état d'avancement actuel. 3. Le client peut effectuer les actions suivantes pour chaque produit : <ul style="list-style-type: none"> — Sélectionner un produit pour en afficher les détails. — Vérifier l'état d'avancement du produit, tel que la préparation, l'emballage, l'expédition, etc. — Obtenir des informations supplémentaires sur l'état actuel du produit, telles que la date de livraison prévue, le transporteur, etc. |
| Scénario alternatif | <ul style="list-style-type: none"> — Si un produit est en rupture de stock ou rencontre un problème de disponibilité, le système affiche un message approprié et propose des solutions alternatives si possible. — Si un produit est en cours de préparation ou d'expédition, le système fournit des mises à jour en temps réel sur son statut. |

TABLEAU 2.8 – Description du cas d'utilisation "Gestion des états des produits commandés"

Répondre au sondage

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Gestion Répondre au sondage ».

| | |
|-------------------------|--|
| Objectif | Permettre aux clients de répondre à un sondage pour donner leur avis sur un produit ou un service. |
| Acteur principal | Client |
| Pré-conditions | Le client est authentifiée. |
| Post-conditions | Les réponses du client sont enregistrées dans le système pour analyse ultérieure. |
| Scénario nominal | Le client accède à son compte sur la plateforme en ligne. Le système affiche les sondages disponibles. Le client sélectionne le sondage qu'il souhaite remplir. Le système affiche les questions du sondage. Le client répond à chaque question en fournissant les informations demandées. Une fois toutes les questions répondues, le client soumet ses réponses. Le système enregistre les réponses du client dans la base de données. |

TABLEAU 2.9 – Description du cas d'utilisation " Répondre au sondage"

Consulter la liste des commerciaux

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Consulter la liste des commerciaux ».

| | |
|-------------------------|--|
| Objectif | Permettre au directeur commercial de consulter la liste des commerciaux de l'entreprise. |
| Acteur principal | Directeur commercial |
| Pré-conditions | Le directeur commercial est authentifié . |
| Post-conditions | Le directeur commercial a accès à la liste complète des commerciaux de l'entreprise. |
| Scénario nominal | -Le directeur commercial se connecte au système . -Le système affiche le tableau de bord principal du directeur commercial. -Le directeur commercial accède à la section "Liste des commerciaux" . -Le système affiche la liste complète des commerciaux de l'entreprise, avec leurs informations de contact et éventuellement d'autres détails pertinents comme leur performance ou leur département. -Le directeur commercial peut parcourir la liste des commerciaux, rechercher des commerciaux spécifiques, ou filtrer les résultats en fonction de différents critères. -Le directeur commercial peut consulter les profils individuels des commerciaux pour obtenir plus de détails si nécessaire. |

TABLEAU 2.10 – Description du cas d'utilisation "Consulter la liste des commerciaux"

Consulter la liste des clients

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Consulter la liste des clients ».

| | |
|-------------------------|--|
| Objectif | Permettre au directeur commercial de Consulter la liste des clients de l'entreprise. |
| Acteur principal | Directeur commercial |
| Pré-conditions | Le directeur commercial est authentifié . |
| Post-conditions | Le directeur commercial a accès à la liste complète des commerciaux de l'entreprise. |
| Scénario nominal | <ul style="list-style-type: none"> -le directeur commercial se connecte à l'interface du système de gestion des clients. -Le directeur commercial accède à la section "Liste des clients" . -Le système affiche une liste de tous les clients enregistrés, avec des informations telles que le nom de l'entreprise, le contact principal, les coordonnées, le chiffre d'affaires annuel, etc. -Le directeur commercial peut utiliser des filtres ou des outils de recherche pour affiner la liste en fonction de critères spécifiques, tels que la région géographique, le secteur d'activité, ou le statut de la relation client. -Le directeur commercial sélectionne un client spécifique pour afficher des détails supplémentaires si nécessaire, comme l'historique des achats, les interactions précédentes, ou les notes internes. Après avoir consulté les informations nécessaires, le directeur commercial peut prendre des décisions informées concernant les stratégies de vente, la gestion de la relation client, ou les opportunités de développement. |

TABLEAU 2.11 – Description du cas d'utilisation "Consulter la liste des clients"

Consulter et valider les commandes

Le tableau ci-dessous représente la description textuelle du cas d'utilisation «Consulter et valider les commandes »

| | |
|----------------------------|--|
| Objectif | Permettre au directeur commercial de Consulter et valider les commandes de l'entreprise. |
| Acteur principal | Directeur commercial |
| Pré-conditions | Le directeur commercial est authentifié . |
| Post-conditions | <ul style="list-style-type: none"> — Le directeur commercial a examiné et validé les commandes clients en attente. — Les commandes validées sont transmises pour traitement, tandis que les commandes rejetées ou renvoyées sont gérées conformément aux procédures établies. |
| Scénario nominal | <p>-Le directeur commercial se connecte à l'interface du système de gestion des commandes.</p> <p>-Le directeur commercial accède à la section "Commandes en attente de validation" .</p> <p>-Le système affiche une liste des commandes clients en attente de validation, avec des détails tels que le numéro de commande, la date de création, le client concerné, le montant total, etc.</p> <p>-Le directeur commercial examine les détails de chaque commande, en s'assurant qu'elles respectent les politiques commerciales de l'entreprise et les accords contractuels avec les clients.</p> <p>-Le directeur commercial peut consulter les informations supplémentaires liées à chaque commande, telles que les spécifications des produits, les conditions de paiement, ou les instructions spéciales.</p> <p>-Après avoir examiné les commandes, le directeur commercial peut décider de les valider, de les rejeter, ou de les renvoyer pour des modifications ou des clarifications supplémentaires.</p> <p>-Si une commande est validée, le système met à jour son statut pour indiquer qu'elle a été approuvée, et elle est ensuite transmise au département approprié pour traitement.</p> <p>-Le directeur commercial peut ajouter des commentaires ou des notes internes pour documenter les décisions prises concernant chaque commande.</p> |
| Scénario alternatif | Si le directeur commercial rencontre des problèmes ou des questions concernant une commande particulière, il peut contacter le service clientèle ou le département des ventes pour obtenir des clarifications supplémentaires avant de prendre une décision. |

TABLEAU 2.12 – Description du cas d'utilisation "Consulter et valider les commandes"

2.9 Conclusion

Dans ce chapitre, nous avons exploré la présentation du langage UML et la méthode agile XP pour comprendre et analyser les besoins du système. Nous avons utilisé UML comme langage de modélisation pour représenter les concepts et les relations du système. Ensuite, nous avons examiné la méthode XP, qui met l'accent sur la communication, la simplicité et l'adaptabilité aux changements. Nous avons identifié les besoins fonctionnels et non fonctionnels, les acteurs du système, et utilisé le diagramme de contexte et le diagramme de cas d'utilisation ainsi que la description textuelle de chaque cas d'utilisation pour mieux comprendre les besoins et les processus du système.

Le prochain chapitre se portera sur la phase de conception en établissant les diagrammes de séquences des cas d'utilisation étudiés, puis en élaborant le diagramme de classe et le modèle logique de données associé.

Chapitre 3

Conception

3.1 Introduction

Ce chapitre se concentre sur la phase cruciale de conception de notre projet. Nous complétons les spécifications des diagrammes du chapitre précédent en fournissant des diagrammes d'implémentation. Nous débutons en établissant des diagrammes de séquences pour des cas d'utilisation spécifiques, afin de mieux visualiser les interactions entre les acteurs et le système. Ensuite, nous procédons à une élaboration rigoureuse du diagramme de classe, représentant l'organisation des classes et de leurs relations. De plus, nous définissons le modèle logique de données, transformant ainsi les concepts du diagramme de classe en structures de base de données. Enfin, nous clôturons cette phase de conception par une conclusion.

3.2 Diagramme de séquence

Le diagramme de séquence décrit les interactions entre objets, en montrant, de façon séquentielle, les envois de message qui interviennent entre les objets. Le diagramme peut également montrer les transmissions de données échangées lors des envois de message [16]. Dans ce qui suit nous allons décrire certains cas d'utilisation sous forme de diagrammes de séquence.

3.2.1 Diagramme de séquence du cas d'utilisation « Authentifier »

Un utilisateur a la possibilité de s'authentifier en saisissant un nom d'utilisateur et un mot de passe dans les champs qui leur sont réservés, après validation, l'utilisateur accède à la session qui lui est propre si les données sont correctes et figurent dans la BDD, sinon le système renvoi un message d'erreur.

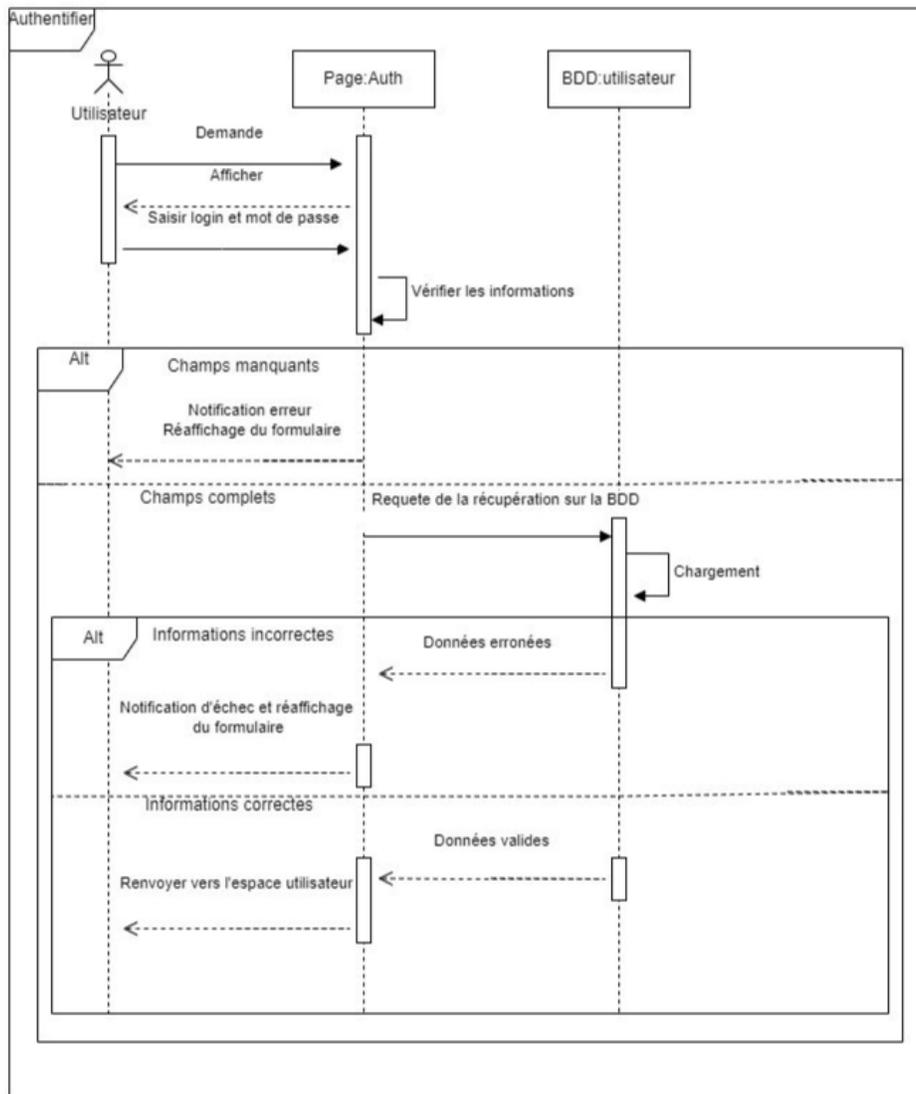


FIGURE 3.1 – Diagramme de séquence du cas d'utilisation s'authentifier « acteur : Utilisateur »

3.2.2 Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur : commercial »

L'administrateur peut gérer les comptes des commerciaux, il a la possibilité d'ajouter un utilisateur en remplissant un formulaire généré par le système, une fois les champs sont remplis il valide, si aucune erreur n'est signalée, le système affiche un message de confirmation.

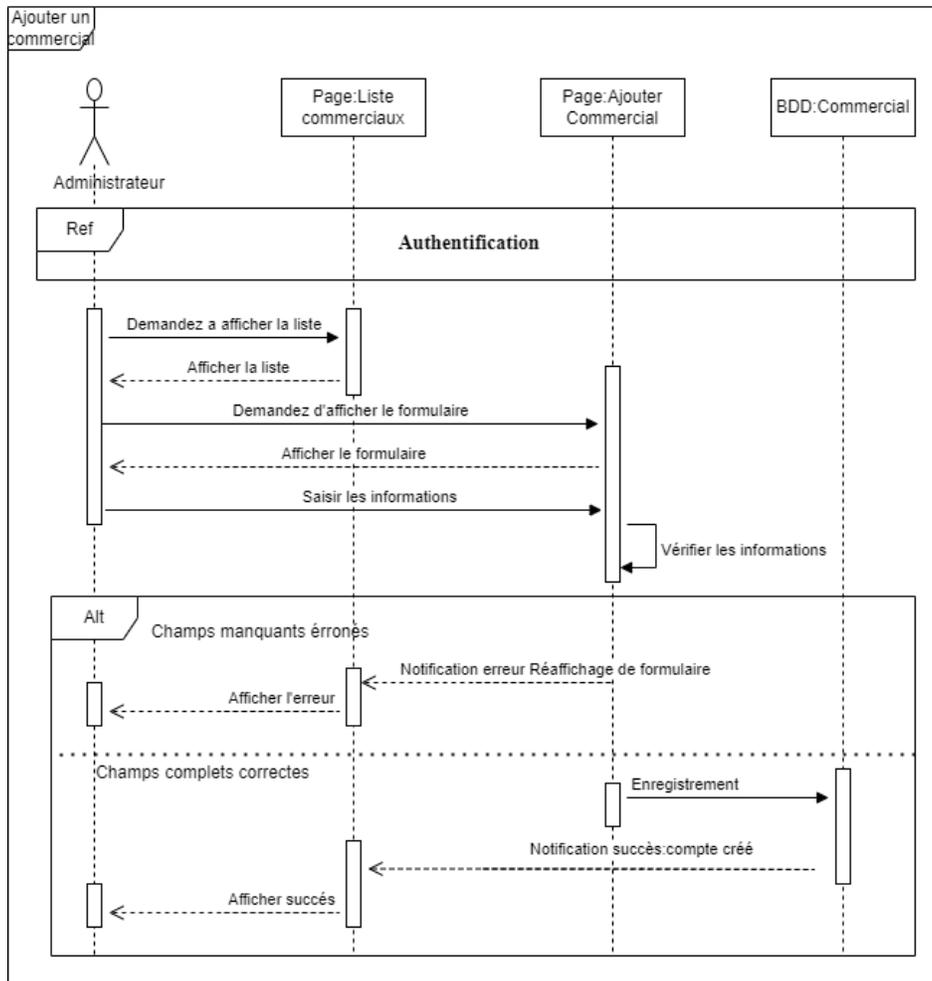


FIGURE 3.2 – Diagramme de séquence du cas d'utilisation ajouter un commercial « acteur : Administrateur ».

3.2.3 Diagramme de séquence du cas d'utilisation « Commander »

Le client peut commander une fois qu'il s'est authentifié. Il va ajouter les produits qu'il souhaite commander. si les données ne comportent pas d'erreurs, la commande sera en attente jusqu'à validation.

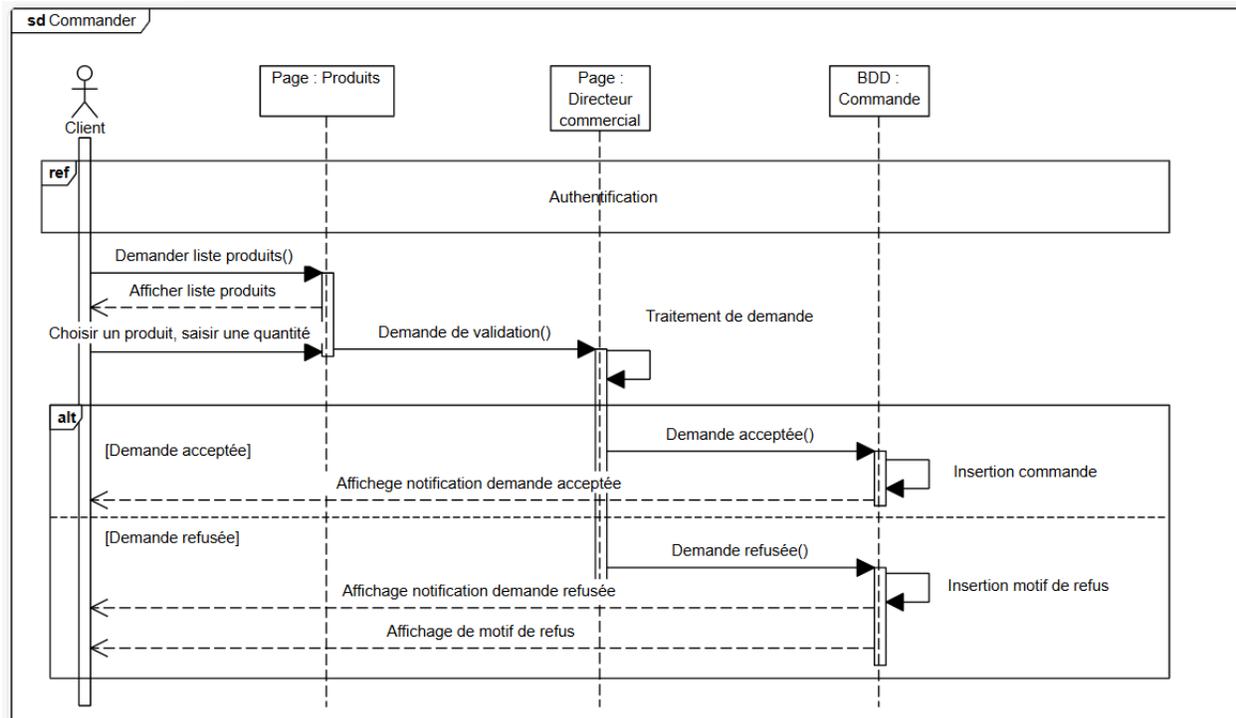


FIGURE 3.3 – Diagramme de séquence du cas d'utilisation commander « acteur : Client ».

3.2.4 Diagramme de séquence du cas d'utilisation « Suivre l'état de la commande »

Le commercial, une fois authentifié, il pourra faire le suivi l'état de chaque commande qu'il lui parvient par son client. Pour cela en choisissant la commande et en cliquant sur le bouton suivre, ensuite un formulaire apparait où il pourra saisir les informations (statut de la commande, son état, le montant, le versement ...etc.).

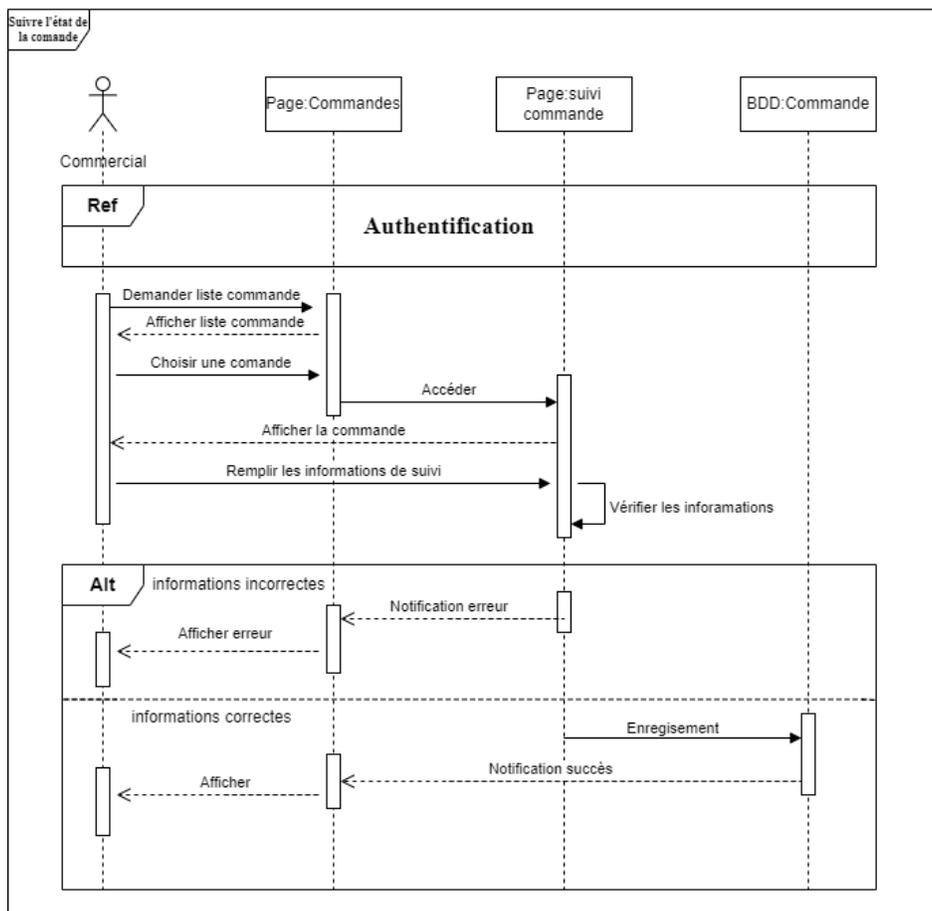


FIGURE 3.4 – Diagramme de séquence du cas d'utilisation suivre l'état de la commande « acteur : Commercial ».

3.2.5 Diagramme de séquence du cas d'utilisation « Suivre l'état du produit de chaque commande »

Le commercial, une fois authentifié, il pourra faire le suivi l'état de chaque produit appartenant à une commande d'un de ses clients. Pour cela en choisissant la commande et le produit, ensuite en cliquant sur le bouton suivre et un formulaire apparaît où il pourra saisir les informations sur le produit (son état, date de livraison...etc.).

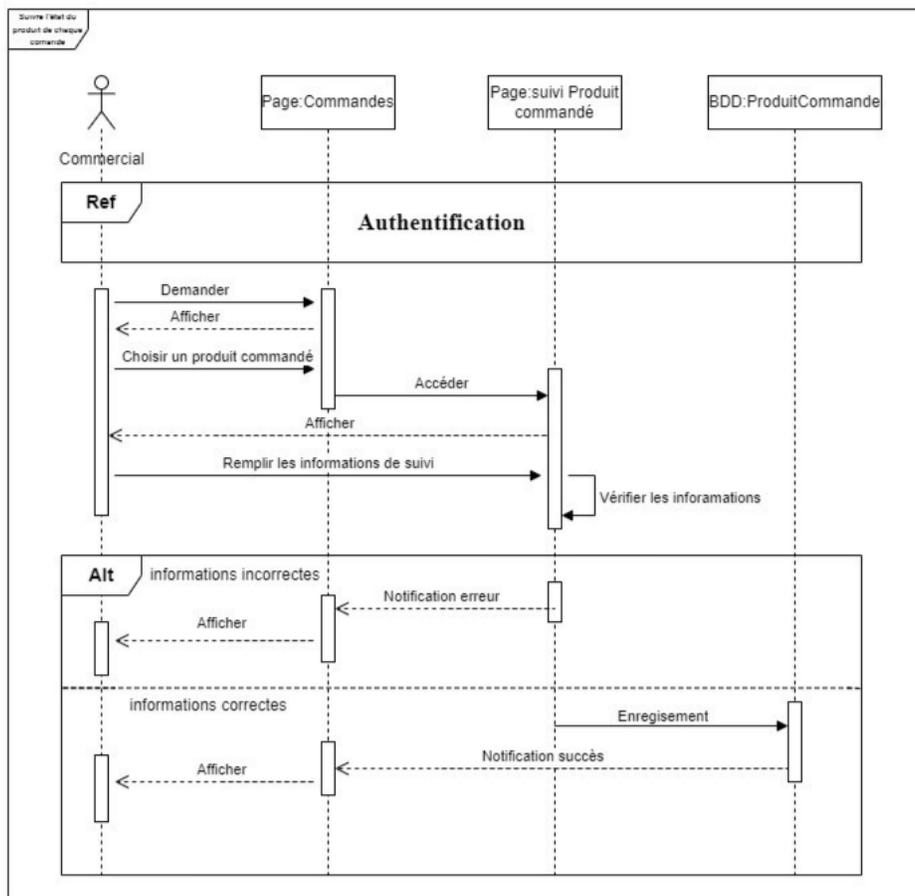


FIGURE 3.5 – Diagramme de séquence du cas d'utilisation suivre l'état du produit de chaque commande « acteur : Commercial ».

3.3 Diagramme de classes

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes [17].

La description de diagramme de classe est fondée sur :

- Le concept d'objet.
- Le concept de classe comprenant les attributs et les opérations.
- Les différents types d'associations entre classes.

Le diagramme ci-dessous représente toutes les classes qui font partie de notre système ainsi que les liens qu'elles ont les unes avec les autres.

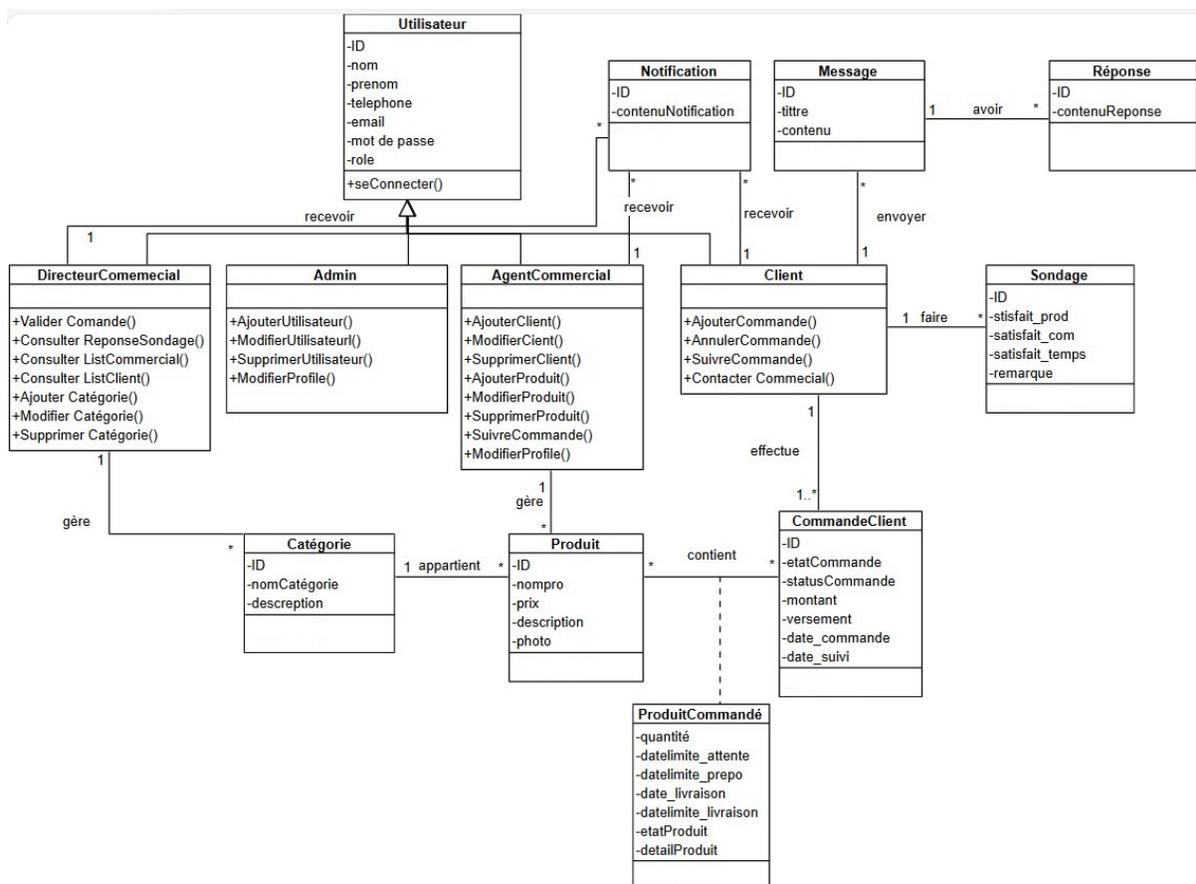


FIGURE 3.6 – Diagramme de classes.

3.4 Modélisation de la BDD

Après avoir construit le diagramme de classes, nous allons établir à ce niveau la base de données lui correspondant en effectuant le passage du diagramme de classes au modèle relationnel.

3.4.1 Modèle relationnel

En respectant les règles de passage qui existent du diagramme de classes au relationnel, nous sommes arrivé au résultat suivant :

- **Utilisateur** (*id_user*, nom_user, prenom_user, tel_mob_user, MDP_user, email_user, role_user).
- **DirecteurCommercial** (*id_dir*, nom_dir, prenom_dir, tel_mob_dir, MDP_dir, email_dir).
- **Client** (*id_soc*, RS_social_soc, adresse_soc, tel_mob_soc, tel_fixe_soc, MDP_soc, email_soc, login_soc).
- **Admin** (*id_admin*, nom_admin, prenom_admin, tel_mob_admin, MDP_admin, email_admin).
- **Commercial** (*id_com*, nom_com, prenom_com, tel_mob_com, MDP_com, email_com).
- **Catégorie** (*id_catg*, nom_catg, desc_catg, #id_admin).
- **Produit** (*id_prd*, nom_prd, desc_prd, prix_unitaire_prd, image, # id_catg, # id_commercial).
- **CommandeClient** (*id_cmd*, etat_cmd, status_cmd, montant_cmd, versement_cmd, date_cmd, datesuivi_cmd, # id_soc).
- **ProduitCommande** (# id_cmd, #id_prd, qtité_prd_cmd, date_limite_attente, date_limite_preparation, date_liv, date_limite_liv, etat_prd, detail_prd).
- **Message** (*id_msg*, titre_msg, contenu_msg, # id_soc).
- **Reponse** (*id_rps*, contenu_rps, # id_msg).
- **Sondage** (*id_sng*, avis_com, avis_produit, avis_temps, remarque, # id_soc).
- **Notification** (*id_notf*, contenu_notf, # id_user).

3.4.2 Structure des tables de la BDD

Le modèle physique de données est l'implantation des données (tables) issues du modèle logique des données en machine afin d'aboutir à la description des fiches de base de données. La structure des tables de la base de données est spécifiée ci-dessous :

3.4.2.1 Utilisateur

| Champ | Type de Donnée | Description | Contrainte |
|--------------|----------------|---|--------------|
| id_user | Int(8) | Identifiant de l'utilisateur | Clé primaire |
| nom_user | Varchar(20) | Nom de l'utilisateur | |
| prenom_user | Varchar(20) | Prénom de l'utilisateur | |
| tel_mob_user | Int(20) | Numéro de téléphone mobile de l'utilisateur | |
| MDP_user | Varchar(20) | Mot de passe de l'utilisateur | |
| email_user | Varchar(50) | Adresse email de l'utilisateur | |
| role_user | Varchar(20) | Rôle de l'utilisateur dans le système | |

TABLEAU 3.1 – Structure de la table « Utilisateur » de la base de données

3.4.2.2 Directeur Commercial

| Champ | Type de Donnée | Description | Contrainte |
|-------------|----------------|--|--------------|
| id_dir | Int(8) | Identifiant du directeur commercial | Clé primaire |
| nom_dir | Varchar(20) | Nom du directeur commercial | |
| prenom_dir | Varchar(20) | Prénom du directeur commercial | |
| tel_mob_dir | Int(20) | Numéro téléphone mobile directeur commercial | |
| MDP_dir | Varchar(20) | Mot de passe du directeur commercial | |
| email_dir | Varchar(50) | Adresse email du directeur commercial | |

TABLEAU 3.2 – Structure de la table « DirecteurCommercial » de la base de données

3.4.2.3 Client

| Champ | Type de Donnée | Description | Contrainte |
|---------------|----------------|--|--------------|
| Id_soc | Int(8) | Identifiant du client (société) | Clé primaire |
| Login_soc | Varchar(20) | Login du client (société) | |
| MDP_soc | Varchar(20) | Mot de passe du client (société) | |
| RS_social_soc | Varchar(20) | Désignation du client (société) | |
| Adresse_soc | Varchar(200) | Adresse du client (société) | |
| Tel_mob_soc | Int(20) | N° de téléphone mobile du client (société) | |
| Tel_fixe_soc | Int(20) | N° du téléphone fixe du client (société) | |
| Email_soc | Varchar(50) | Email du client (société) | |

TABLEAU 3.3 – Structure de la table « Client » de la base de données

3.4.2.4 Commercial

| Champ | Type de Donnée | Description | Contrainte |
|-------------|----------------|--------------------------------------|--------------|
| Id_com | Int(8) | Identifiant du commercial | Clé primaire |
| Login_com | Varchar(20) | Login du commercial | |
| Mdp_com | Varchar(20) | Mot de passe du commercial | |
| Nom_com | Varchar(20) | Nom du commercial | |
| Prenom_com | Varchar(20) | Prénom du commercial | |
| Tel_mob_com | Int(20) | N° de téléphone mobile du commercial | |
| Email_com | Varchar(50) | Email du commercial | |

TABLEAU 3.4 – Structure de la table « Commercial » de la base de données

3.4.2.5 Admin

| Champ | Type de Donnée | Description | Contrainte |
|---------------|----------------|---|--------------|
| id_admin | Int(8) | Identifiant de l'administrateur | Clé primaire |
| nom_admin | Varchar(20) | Nom de l'administrateur | |
| prenom_admin | Varchar(20) | Prénom de l'administrateur | |
| tel_mob_admin | Int(20) | Numéro téléphone mobile de l'administrateur | |
| MDP_admin | Varchar(20) | Mot de passe de l'administrateur | |
| email_admin | Varchar(50) | Adresse email de l'administrateur | |

TABLEAU 3.5 – Structure de la table « Admin » de la base de données

3.4.2.6 Catégorie

| Champ | Type de Donnée | Description | Contrainte |
|-----------|----------------|---|---------------|
| id_catg | Int(8) | Identifiant de la catégorie | Clé primaire |
| nom_catg | Varchar(50) | Nom de la catégorie | |
| desc_catg | Text | Description de la catégorie | |
| id_admin | Int(8) | Identifiant de l'administrateur associé | Clé étrangère |

TABLEAU 3.6 – Structure de la table « Catégorie » de la base de données

3.4.2.7 Produit

| Champ | Type-donnée | Description | Contrainte |
|--------------------|---------------|---|---------------|
| Id_prod | Int(8) | Identifiant du produit | Clé primaire |
| Nom_prod | Varchar(20) | Nom du produit | |
| Desc_prod | Varchar(200) | Description du produit | |
| Prix_unitaire_prod | Decimal(3, 2) | Prix unitaire du produit | |
| Image_prd | Varchar(100) | Image du produit | |
| Id_catg | Int(8) | Identifiant de la catégorie du produit | Clé étrangère |
| Id_soc | Int(8) | Identifiant client auquel appartient le produit | Clé étrangère |

TABLEAU 3.7 – Structure de la table « Produit » de la base de données

3.4.2.8 CommandeClient

| Champ | Type de Donnée | Description | Contrainte |
|---------------|----------------|------------------------------------|---------------|
| id_cmd | Int(8) | Identifiant de la commande | Clé primaire |
| etat_cmd | Varchar(200) | État de la commande | |
| status_cmd | Varchar(200) | Statut de la commande | |
| montant_cmd | Decimal(10,2) | Montant de la commande | |
| versement_cmd | Decimal(10,2) | Versement de la commande | |
| date_cmd | Date-cmd | Date de la commande | |
| datesuivi_cmd | Date-suivi | Date de suivi de la commande | |
| id_soc | Int(8) | Identifiant de la société associée | Clé étrangère |

TABLEAU 3.8 – Structure de la table « Commande Client » de la base de données

3.4.2.9 Produit-Commande

| Champ | Type de Donnée | Description | Contrainte |
|-------------------------|----------------|-------------------------------|---------------|
| id_cmd | Int(8) | Identifiant de la commande | Clé étrangère |
| id_prd | Int(8) | Identifiant du produit | Clé étrangère |
| qtité_prd_cmd | Int(8) | Quantité de produit commandée | |
| date_limite_attente | Date | Date limite d'attente | |
| date_limite_preparation | Date | Date limite de préparation | |
| date_liv | Date | Date de livraison | |
| date_limite_liv | Date | Date limite de livraison | |
| etat_prd | Varchar(20) | État du produit | |
| detail_prd | Text | Détail du produit | |

TABLEAU 3.9 – Structure de la table « ProduitCommande » de la base de données

3.4.2.10 Message

| Champ | Type de Donnée | Description | Contrainte |
|-------------|----------------|------------------------------------|---------------|
| id_msg | Int(8) | Identifiant du message | Clé primaire |
| titre_msg | Varchar(100) | Titre du message | |
| contenu_msg | Text | Contenu du message | |
| id_soc | Int(8) | Identifiant de la société associée | Clé étrangère |

TABLEAU 3.10 – Structure de la table « Message » de la base de données

3.4.2.11 Réponse

| Champ | Type de Donnée | Description | Contrainte |
|-------------|----------------|--------------------------------|---------------|
| id_rps | Int(8) | Identifiant de la réponse | Clé primaire |
| contenu_rps | Text | Contenu de la réponse | |
| id_msg | Int(8) | Identifiant du message associé | Clé étrangère |

TABLEAU 3.11 – Structure de la table « Réponse » de la base de données

3.4.2.12 Sondage

| Champ | Type de Donnée | Description | Contrainte |
|--------------|----------------|------------------------------------|---------------|
| id_sng | Int(8) | Identifiant du sondage | Clé primaire |
| avis_com | Int(1) | Avis sur la communication | |
| avis_produit | Int(1) | Avis sur le produit | |
| avis_temps | Int(1) | Avis sur le temps | |
| remarque | Text | Remarque | |
| id_soc | Int(8) | Identifiant de la société associée | Clé étrangère |

TABLEAU 3.12 – Structure de la table « Sondage » de la base de données

3.4.2.13 Notification

| Champ | Type de Donnée | Description | Contrainte |
|--------------|----------------|--------------------------------------|---------------|
| id_notf | Int(8) | Identifiant de la notification | Clé primaire |
| contenu_notf | Text | Contenu de la notification | |
| id_user | Int(8) | Identifiant de l'utilisateur associé | Clé étrangère |

TABLEAU 3.13 – Structure de la table « Notification » de la base de données

3.5 Le passage du modèle objet au modèle relationnel

Le passage du modèle objet au modèle relationnel est nécessaire lors de l'utilisation d'un SGBDR, ce qui implique un changement de représentation entre la structure des classes et la structure des données relationnelles. Ces deux structures ont des analogies, et les équivalences sont utilisées pour les rapprocher.

Une classe définit une structure de données à laquelle souscrivent des instances, elle correspond donc à une table du modèle relationnel : chaque attribut donne lieu à une colonne, chaque instance stocke ses données dans une ligne (t-uplet) et son OID sert de clé primaire.

Certains attributs de type complexe ne correspondent à aucun des types de SQL. On rencontre fréquemment ce cas pour les attributs représentant une structure de données. Un type complexe peut être conçu : -soit avec plusieurs colonnes, chacune correspondant à un champ de la structure. -soit avec une table spécifique dotée d'une clé étrangère pour relier les instances aux valeurs de leur attribut complexe.

| Modèle Objet | Modèle Relationnel |
|---------------------------|---|
| Classe | Table |
| Attribut de type simple | Colonne |
| Attribut de type composé | Colonne ou Clé étrangère |
| Instance | Tuple |
| OID (Identifiant d'Objet) | Clé primaire |
| Association | Clé étrangère ou Table de liaison |
| Héritage | Clé primaire identique sur plusieurs tables |

TABLEAU 3.14 – Équivalence entre les concepts objet et relationnel

Règle du Passage au modèle relationnel

Nous avons pu passer du diagramme de classes au modèle relationnel en se basant sur les règles suivantes :

- **Règle 1** : Lors de la transformation des classes en tables, chaque classe devient une table. L'identifiant de la classe devient la clé primaire de la table, et chaque attribut de la classe devient un champ de la table.
- **Règle 2** : Dans le cas d'une association plusieurs-à-plusieurs ($?.*$ à $?.*$), trois tables sont créées. La table d'association possède les clés primaires des deux autres tables en tant que clés primaires, ainsi que ses propres attributs si elle est une classe d'association.
- **Règle 3** : Dans le cas d'une association un-à-plusieurs ($?.1$ à $?.*$), une table dérivée de l'entité ayant une cardinalité de (1..1) est créée. La clé primaire de cette entité est ensuite ajoutée comme clé étrangère dans la table dérivée de l'entité ayant une cardinalité de ($?.*$).

3.6 Conclusion

Dans ce chapitre, nous avons abordé les différents diagrammes UML tels que les diagrammes de cas d'utilisation et les diagrammes de séquences permettant de cerner le fonctionnement de notre application web et mettre en évidence les interactions entre les acteurs et le système. Ensuite, On a déterminé les différentes entités et associations qui ont donné lieu d'élaborer le diagramme de classes.

À la fin, nous avons présenté le modèle relationnel construit, structure logique et physique. La prochaine étape consistera donc en la concrétisation du modèle de la solution que nous avons proposé, en d'autres termes, la réalisation des différents objets et des différentes fonctionnalités qui la constituent. Cet aspect fera l'objet de la prochaine étape.

Dans le prochain chapitre, nous passerons à la phase d'implémentation, où nous concrétiserons ces concepts et mettrons en place les fonctionnalités de notre application.

Chapitre 4

Réalisation

4.1 Introduction

Dans le chapitre précédent, nous avons présenté la conception et le fonctionnement global de notre système, nous arrivons dans ce chapitre à la réalisation et à la mise en œuvre de l'application. Ce chapitre sera divisé en trois parties : la première partie sera consacrée à la description de l'environnement de notre application. La deuxième partie traitera en détail des outils de développement pour la réalisation de notre solution. Enfin, la troisième partie contiendra la présentation de la plateforme avec quelques interfaces qui ont été réalisées.

4.2 Langages de développement

Dans cette partie nous allons citer les langages utilisés pour le développement de notre application.

4.2.1 HTML

HTML signifie HyperText Markup Language. C'est un langage informatique qui sert à créer des sites web, il est incontournable et universel aujourd'hui. Il est à la base du fonctionnement de tous les sites web. Son rôle est de gérer et organiser le contenu. C'est donc en HTML que vous écrirez ce qui doit être affiché sur la page [18].

4.2.2 CSS

CSS est un langage de mise en forme des sites web ,il a été lancé au milieu des années 90 et est à présent considéré comme le langage de feuilles de style standard sur le World Wide Web [19].

4.2.3 PHP

PHP est un langage de programmation exécuter par le serveur pour prendre des décisions, il permet de personnaliser la page en fonction du visiteur et de rendre un site dynamique, son rôle est de générer du code HTML. Il est spécialement conçu pour le développement d'applications web [20].

4.2.4 MySQL

MySQL est un serveur de base de données relationnelle, distribué sous double licence, une licence publique générale GNU et une propriétaire. Il est largement exploité comme système de gestion de base de données pour des applications utilisant PHP [21].

4.2.5 Javascript

JavaScript est un langage de programmation interprété créé à l'origine par Netscape, il a été conçu pour traiter localement des évènements provoqué par le lecteur. Il permet de mettre en œuvre des éléments complexes sur des pages Web [22].

4.3 Outils de développement

Dans cette partie nous allons citer les outils utiliser durant le développement de note application.

4.3.1 Laravel

De nombreuses possibilités existent pour réaliser des applications Internet depuis plusieurs années. Des langages ont été créés, des architectures et des environnements de travail ont été conçus pour répondre aux besoins des développeurs et leur faciliter la tâche. En 2011, Taylor OTWELL décide avec une équipe de développeurs de créer Laravel, un framework web open

source écrit en PHP respectant le principe MVC (modèle- vue-contrôleur) et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub. Laravel possède une grande variété de fonctionnalités, parmi elles : (authentification, système de routage, ORM eloquent, etc.) [23].



FIGURE 4.1 – Logo de laravel

4.3.2 Visual Studio Code

Visual Studio Code est un logiciel open-source et gratuit qui permet d'éditer du code sur différentes plateformes. Il est surtout destiné à la création d'application avec JavaScript, TypeScript et Node.js, mais il peut aussi s'adapter à d'autres langages grâce à un système d'extension riche [24].



FIGURE 4.2 – Logo Visual Studio Code

4.3.3 XAMPP

XAMPP est un serveur Web multiplateforme gratuit et open-source créé par Apache Friends qui permet aux programmeurs d'écrire et de tester leur code sur un serveur Web local [25].



FIGURE 4.3 – Logo XAMPP

4.3.4 Bootstrap

Bootstrap est un Framework open source qui permet de créer des sites web responsive avec les langages HTML, CSS et JavaScript, On appelle ce type de framework un "Front-End Framework" [26].



FIGURE 4.4 – Logo Bootstrap

4.4 Outils d'édition UML

4.4.1 StarUML

StarUML est un outil de modélisation UML open source et multiplateforme, compatible avec le métamodèle et les diagrammes standard UML 2.x [27].



FIGURE 4.5 – Logo StarUML

4.4.2 Visual Paradigm

Visual Paradigm est un outil de conception et de gestion puissant, multiplateforme facile à utiliser pour les systèmes informatiques, il prend en charge de nombreux diagrammes (diagramme de classe, diagramme de cas d'utilisation . . .) [28].



FIGURE 4.6 – Logo Visual Paradigm

4.5 Architecture de l'application

Notre application se base sur l'architecture de Laravel, qui comme une grande partie des autres frameworks PHP, a une architecture dite MVC (Model – View – Controller). Chaque action de l'utilisateur passe par le contrôleur (*controller*) qui envoie des demandes de manipulation d'un objet au modèle. Le modèle (*model*) effectue les changements de l'objet et le renvoie au contrôleur qui à son tour le passe à la vue (*view*).

Ainsi, ces trois pôles ont des responsabilités différentes :

- **le *model*** : Ce que nous appelons un modèle est en réalité un fichier PHP qui ne fait que gérer les échanges avec la base de données. Lorsque nous avons besoin de lire ou écrire dans la base de données, nous faisons appel au modèle.
- **le *view*** : Les vues sont principalement des fichiers HTML contenant le code destiné à être transmis au navigateur de l'utilisateur. Nous disposons des données transmises par le contrôleur afin de les intégrer dynamiquement dans nos pages.
- **le *controller*** : Traite les actions utilisateur (via des requêtes), demande au modèle d'effectuer les changements, puis passe les données à la vue. Le contrôleur a donc une place centrale dans cette architecture. Il est le pont entre les interactions utilisateur et les traitements de données. Tout passe par lui afin de pouvoir tout contrôler... d'où son nom.

Enfin, le routage (*routing*) de Laravel est un système simple mais puissant qui va mapper la requête de l'utilisateur avec le bon contrôleur qui traitera cette demande.

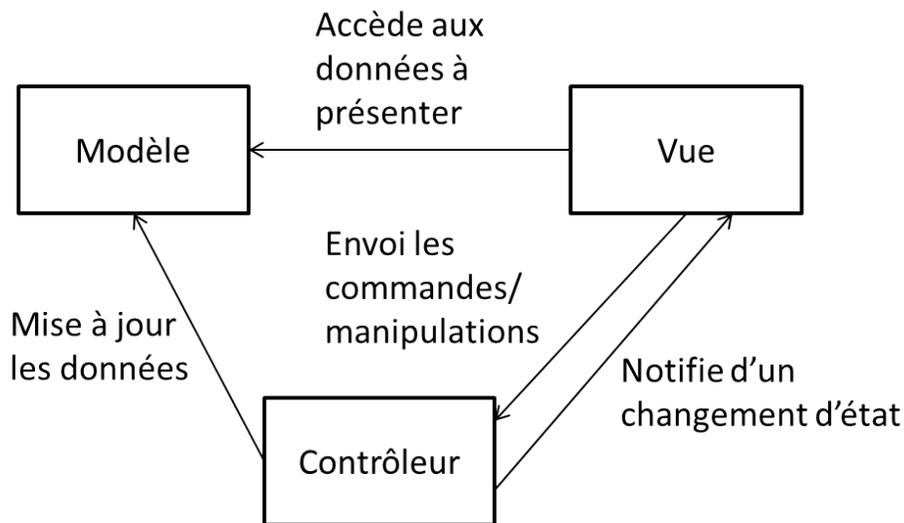


FIGURE 4.7 – Architecture MVC

4.6 Présentation des Interfaces de notre application

Dans cette section, nous allons présenter quelques interfaces de notre application.

4.6.1 Page d'authentification

Il s'agit d'un simple formulaire login qui permet aux utilisateurs de se connecter vers leurs propres espaces.

Se connecter S'inscrire

GENERAL EMBALLAGE

Connexion

Adresse email

Mot de passe

Remember Me

[Mot de passe oublié?](#)

FIGURE 4.8 – Page d'authentification

4.6.2 Interface Gestion des commerciaux (espace administrateur)

Une fois connecté, l'administrateur peut consulter la liste des commerciaux, il peut aussi chercher, ajouter, supprimer ou modifier un compte.

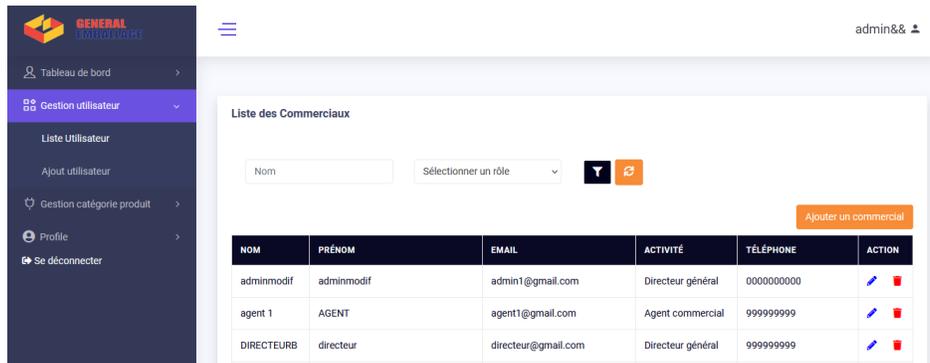


FIGURE 4.9 – Interface Gestion des commerciaux

4.6.3 Interface Gestion des catégories (espace administrateur)

Une fois connecté, l'administrateur peut gérer les catégories (ajouter, modifier, supprimer).

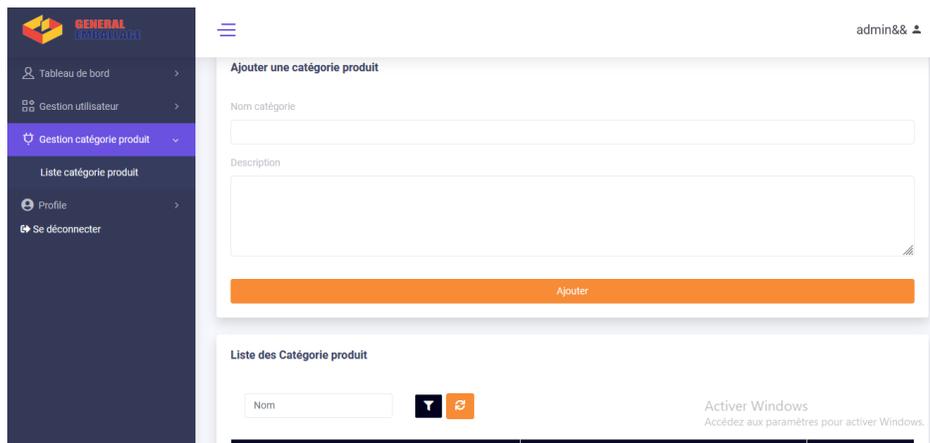


FIGURE 4.10 – Gestion des catégories

4.6.4 Interface Gestion des client (espace commercial)

Une fois connecté, le commercial peut gérer les clients (ajouter, modifier, supprimer).

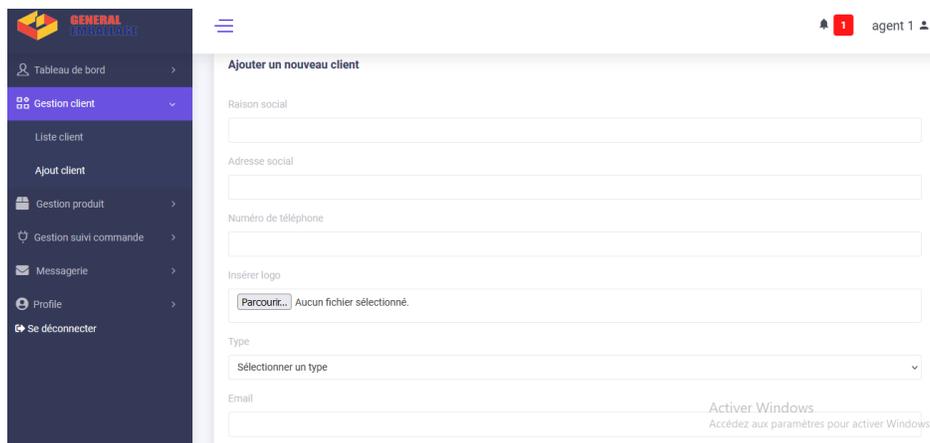


FIGURE 4.11 – Gestion des clients

4.6.5 Interface Gestion des produits (espace commercial)

Une fois connecté, le commercial peut gérer les produits (ajouter, modifier, supprimer).

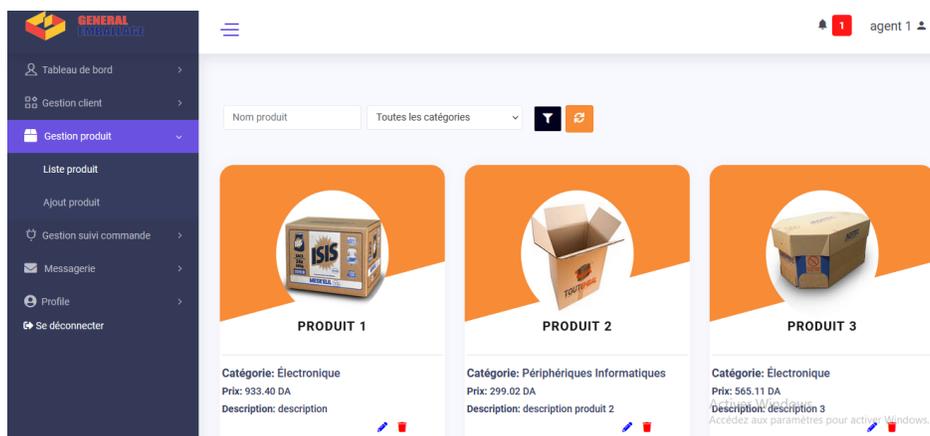


FIGURE 4.12 – Interface Gestion des produits

4.6.6 Interface Gestion suivi de commande des clients (espace commercial)

Une fois connecté, le commercial peut consulter les commandes des clients classés, il peut aussi chercher, modifier, annuler, faire le suivi de chaque commande et de chaque produit commandé.

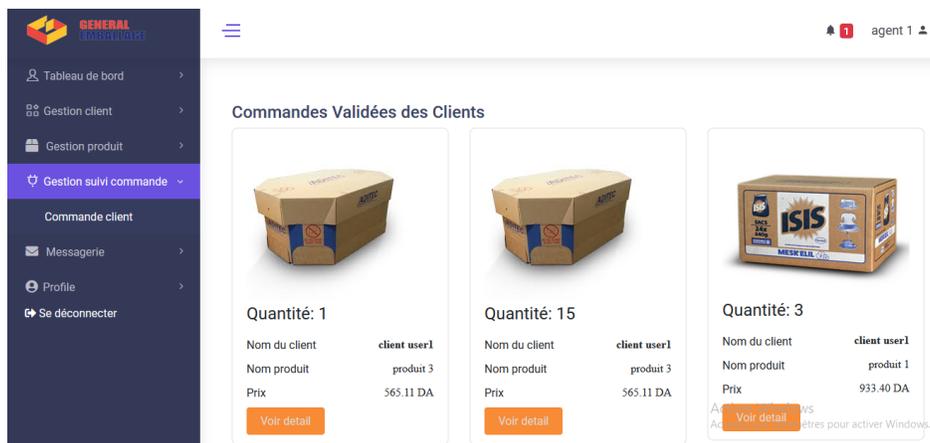


FIGURE 4.13 – Interface Gestion suivi de commande

4.6.7 Interface Messagerie (espace commercial)

Une fois connecté, le commercial peut consulter les messages des clients. Il peut aussi répondre aux clients ainsi que chercher, modifier et supprimer ses réponses.

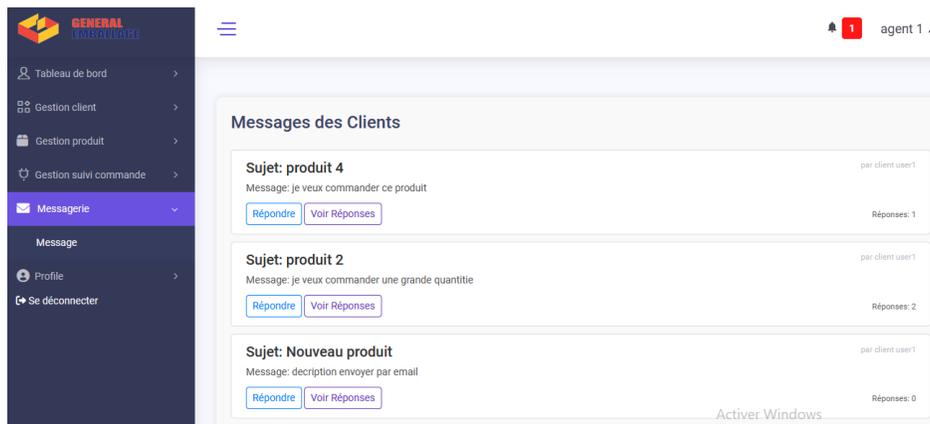


FIGURE 4.14 – Interface Messagerie espace commercial

4.6.8 Interface Liste des produits pour client (espace client)

Une fois connecté, le client peut consulter ses produits.

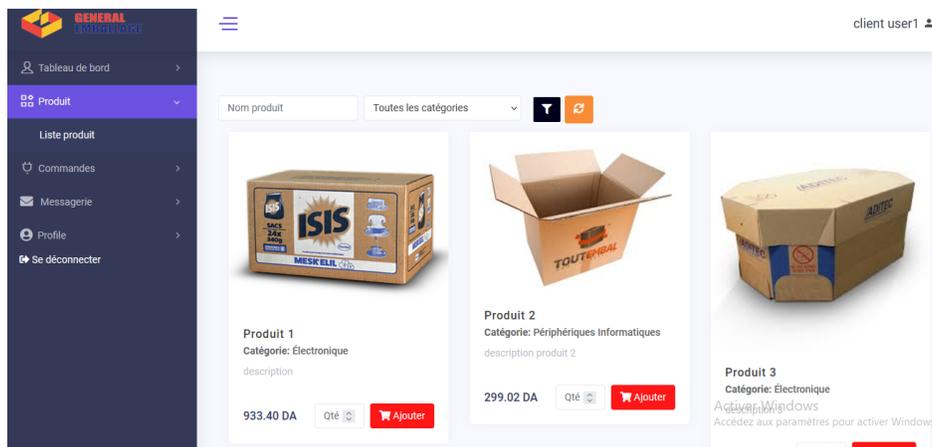


FIGURE 4.15 – Interface Liste des produits client

4.6.9 Interface Liste des commandes clients (espace client)

Le client pourra suivre l'état d'avancement de chacune de ses commandes et l'annuler commande.

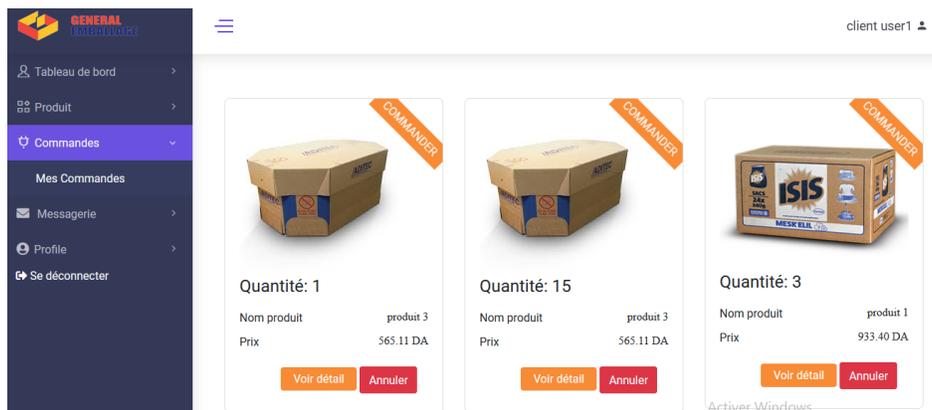
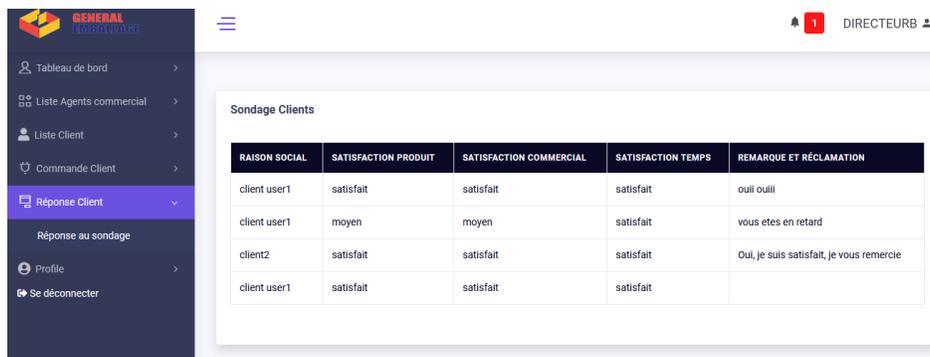


FIGURE 4.16 – Interface Liste des commandes clients

4.6.10 Interface Consulter le retour des clients (espace directeur commercial)

Le directeur commercial peut afficher le retour d'avis des clients saisis en répondant en sondage.

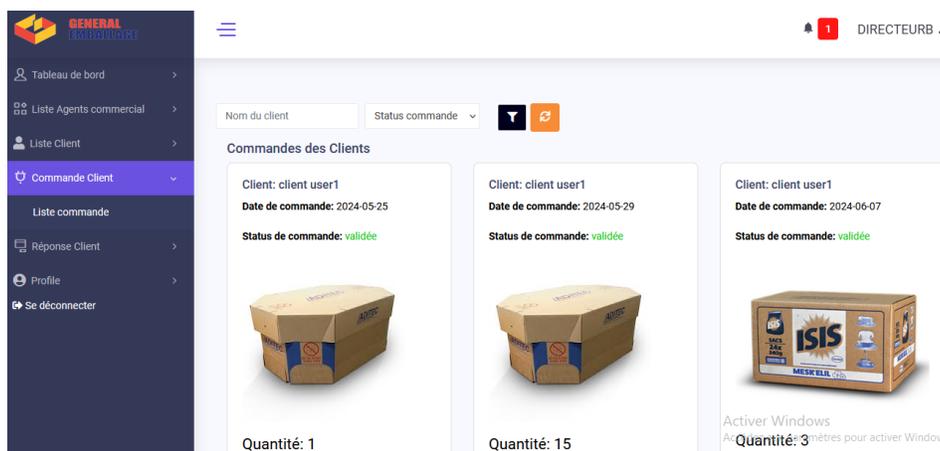


| RAISON SOCIAL | SATISFACTION PRODUIT | SATISFACTION COMMERCIAL | SATISFACTION TEMPS | REMARQUE ET RÉCLAMATION |
|---------------|----------------------|-------------------------|--------------------|--|
| client user1 | satisfait | satisfait | satisfait | ouli ouili |
| client user1 | moyen | moyen | satisfait | vous etes en retard |
| client2 | satisfait | satisfait | satisfait | Oui, Je suis satisfait, je vous remercie |
| client user1 | satisfait | satisfait | satisfait | |

FIGURE 4.17 – Interface Gestion répondre aux sondage

4.6.11 Interface Consulter et valider les commandes (espace directeur commercial)

Une fois connecté, le directeur commercial peut Consulter et valider les commandes des clients.



Nom du client Status commande

Commandes des Clients

Client: client user1
Date de commande: 2024-05-25
Status de commande: validée



Quantité: 1

Client: client user1
Date de commande: 2024-05-29
Status de commande: validée



Quantité: 15

Client: client user1
Date de commande: 2024-06-07
Status de commande: validée



Quantité: 3

FIGURE 4.18 – Interface Consulter et valider les commandes

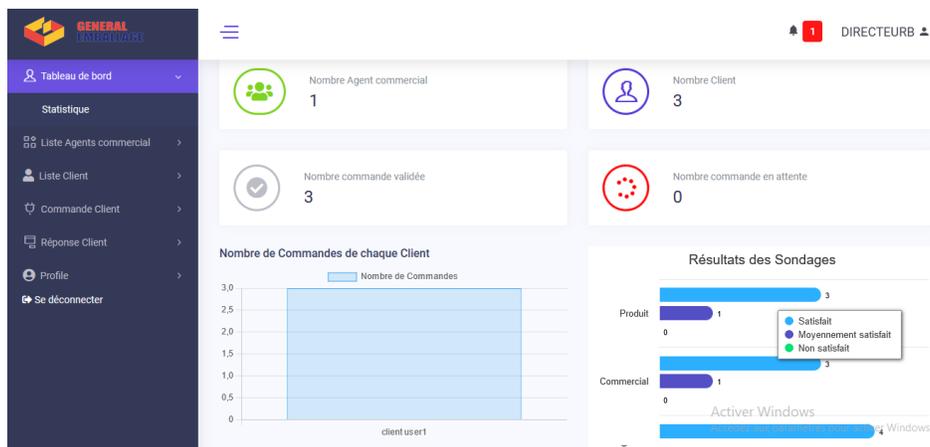


FIGURE 4.21 – Interface Tableau de bord directeur commerciale

4.6.15 Interface Tableau de bord commerciale

Une fois connecté, le commercial peut Consulter leur Tableau de bord .

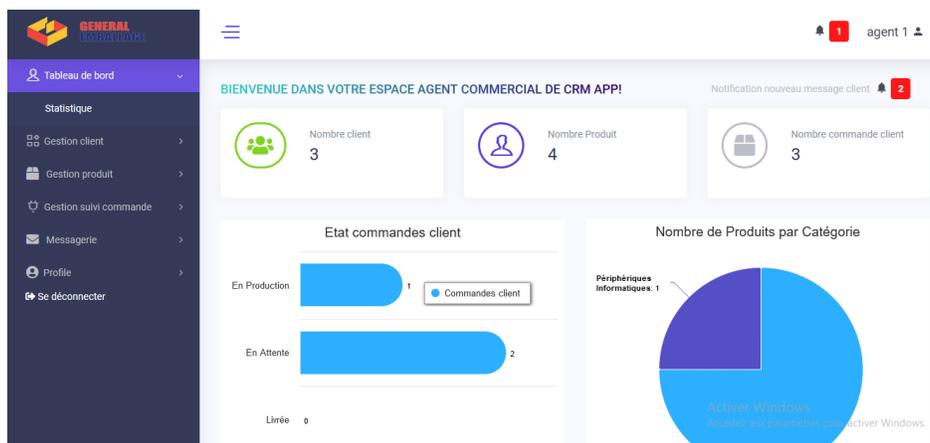


FIGURE 4.22 – Interface Tableau de bord commerciale

4.6.16 Interface Tableau de bord client

Une fois connecté, le client peut Consulter leur Tableau de bord .



FIGURE 4.23 – Interface Tableau de bord client

4.6.17 Interface Tableau de bord administrateur

Une fois connecté, l'administrateur peut Consulter leur Tableau de bord .

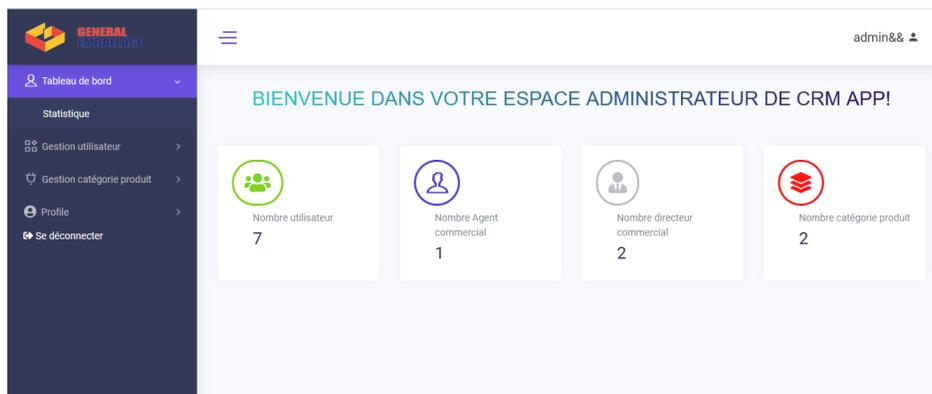


FIGURE 4.24 – Interface Tableau de bord e administrateur

4.7 Conclusion

Dans ce dernier chapitre, nous avons présenté la mise en œuvre de l'application CRM concernant la fonction vente de l'entreprise Général Emballage. En premier lieu, nous avons cité l'environnement et les outils de développement, puis, nous avons présenté quelques interfaces réalisées.

Conclusion générale

Toute entreprise, quel que soit son domaine ou sa vocation, qui veut continuer d'exister sur le marché et réussir à atteindre ses objectifs, est appelée à évoluer et à s'adapter à son environnement. Un environnement de plus en plus exigeant avec une concurrence rude, caractérisé par un développement technologique exponentiel, donc les entreprises se voient dans l'obligation de se doter de nouvelles stratégies et de technologies lui permettant d'être plus performante.

Parmi ces stratégies, le CRM qui répond à ce besoin. Basé sur la personnalisation du service client, le CRM constitue un choix stratégique qui de l'avis de nombreux spécialistes représente une révolution en matière de stratégie d'entreprise, permettant par ses différentes fonctionnalités de mieux gérer l'organisation et la gestion autour du client, toujours dans le but de le satisfaire et ainsi le fidéliser.

Dans ce travail nous avons élaboré une application web CRM pour une entreprise de fabrication et de transformation de carton ondulé Général Emballage, avec laquelle nous avons pu forger une idée plus au moins générale sur le fonctionnement et les caractéristiques de l'entreprise. Cela malgré les difficultés qu'on a eu à cerner les vrais besoins à cause des exigences qui variaient en file du temps, notamment le souci de confidentialité qui s'imposait toujours à nous. Mais nous avons pu, à partir d'une analyse des informations collectées grâce à une série d'entretiens avec les commerciaux et le responsable informatique, de mettre en œuvre une solution CRM web pour le client et l'unité commerciale.

nous avons mené une étude des besoins et une exploration des différentes données collectées de l'entreprise en particulier l'unité commerciale, ce qui nous a permis de relever les manques et donc de proposer et réaliser toutes les fonctionnalités de base que nous avons jugées nécessaires. Tout cela dans le but de permettre :

- Au client de passer ses commandes en ligne, suivre l'historique de l'état d'avancement en toute transparence.
- Aux commerciaux de mieux gérer les produits, les commandes et le suivi de ses clients.

Et pour conclure notre travail, nous pensons qu'il est intéressant de rajouter quelques perspectives et recommandations pour le développement de l'application, telles que :

- Intégrer la facturation et le paiement électronique à la plateforme pour aller encore plus loin dans le processus de prise de commande et le compléter.
- Pour plus de flexibilité et d'efficacité, il est possible de proposer une version mobile de l'application qui sera dans le cadre du CRM mobile, qui est de plus en plus utilisée. Un CRM mobile offre une expérience CRM complète sur les smartphones, tablettes et autres appareils connectés à Internet. Les équipes de vente, de marketing et de service client peuvent ainsi accéder à des informations clés en temps réel, les gérer en toutes circonstances et ainsi transmettre rapidement aux clients les informations fiables dont ils ont besoin.

Bibliographie

- [1] site de "général emballage". <https://www.generalemballage.com>.
- [2] Source entreprise général emballage. Consulté le 12/03/2024.
- [3] René Lefebure and Gilles Venturi. *Gestion de la relation client*. Éditions Eyrolles, 2005.
- [4] H.A. Al-Homer, H. Asharai, and A. Ahmad. The core components and types of crm. *Extrait d'une revue universitaire de Malaisie*, 2019.
- [5] Nicolas Saint Cast. *Organiser sa relation client aujourd'hui : le CRM nouvelle manière*. Éditions Maxima Laurent du Mesnil éditeur, 2003.
- [6] Equipe Hubspot. customer-relationship-management, 2020.
- [7] Yacine Mani and Lydia Siaci. Conception et réalisation d'un système de facturation dédié aux erp/crm. Master's thesis, Université Mouloud Mammeri, Tizi Ouzou, 2015.
- [8] Planzone. Qu'est-ce que la méthode agile ?, s.d. Consulté le 18/04/2024.
- [9] Reussirsprojets. Méthode agile, s.d. Consulté le 18/04/2024.
- [10] A. Raeburn. L'extreme programming (xp), s.d. Consulté le 18/04/2024.
- [11] Planzone. Qu'est-ce que la méthode extreme programming ?, s.d. Consulté le 18/04/2024.
- [12] Wimi-teamwork. Extreme programming : Guide d'introduction pour les débutants, s.d. Consulté le 19/04/2024.
- [13] Alecoledelavie. Le cycle de vie de la méthode xp (extreme programming), s.d. Consulté le 19/04/2024.
- [14] S. O. T. I. MIND. Extreme programming (xp) method of developing i.s., 2013. Consulté le 19/04/2024.

-
- [15] P. Gérard. *Processus de développement logiciel*, 2007.
- [16] Pierre-Alain Muller and Nathalie Gaertner. *Modélisation objet avec UML*. Eyrolles, Paris, 2 edition, 2000. Consulté le 21/04/2024.
- [17] Joseph Gabay and David Gabay. *UML 2 Analyse et Conception*. Dunod, Paris, 2008. Consulté le 24/04/2024.
- [18] Mathieu Nebra. *Réalisez votre site web avec HTML 5 et CSS 3*. Éditions Eyrolles, Paris, 2017.
- [19] D. G. Ionos. *Qu'est-ce que le css?* Consulté le 29/04/2024.
- [20] PHP. *Qu'est-ce que php?*
- [21] Sql.sh. *Mysql*.
- [22] Sql.sh. *Mysql*.
- [23] Matt Stauffer. *Laravel : Up & Running : A Framework for Building Modern PHP Apps*. O'Reilly Media, Inc., 2019.
- [24] Edutechwiki. *Visual_studio_code*.
- [25] EDUCBA. *What is xampp*, consulté le 12 mai 2024.
- [26] Yieldstudio. *Laravel*, consulté le 12 mai 2024.
- [27] StarUML.io. *Staruml*, consulté le 12 mai 2024.
- [28] Visual Paradigm. *Visual paradigm*, consulté le 30 mai 2024.

Annexes

Annexe A

1. Historique de l'UML

Au début des années 90, une cinquantaine de méthodes d'analyse et de conception objet qui existaient, seulement trois d'entre se sont nettement détachées. En effet, la volonté de converger vers une méthode unifiée était déjà bien réelle et c'est pour cette raison que les méthodes OMT, BOOCH et OOSE se sont démarquées des autres.

OMT (Object Modeling Technique) de James Rumbaugh et BOOCH de Grady Booch ont été les deux méthodes les plus diffusées en France durant les années 90. Par ailleurs, OOSE d'Ivar Jacobson s'est aussi imposée dans le monde objet pour la partie formalisation des besoins. Pour aller plus loin dans le rapprochement, James Rumbaugh et Grady Booch se sont retrouvés au sein de la société « Rational Software » et ont été ensuite rejoints par Ivar Jacobson, les trois gourous se mirent d'accord pour définir une méthode commune qui fédérerait leurs apports respectifs (on les surnomme depuis « The Amigos »). UML (Unified Modeling Language) est né de cet effort de convergence. L'adjectif unified est là pour marquer qu'UML unifie, et donc remplace. En fait, et comme son nom l'indique, UML n'a pas l'ambition d'être exactement une méthode : c'est un langage.

L'unification a progressé par étapes. En 1995 Booch et Rumbaugh (et quelques autres) se sont mis d'accord pour construire une méthode unifiée, « Unified Method 0.8 » ; en 1996, Jacobson les a rejoints pour produire UML 0.9 (notez le remplacement du mot méthode par le mot langage, plus modeste). Les acteurs les plus importants dans le monde du logiciel s'associent alors à l'effort (IBM, Microsoft, Oracle, DEC, HP, Rational, Unisys etc.) et UML 1.0 est soumis à l'OMG. La notation UML est adoptée par l'OMG en novembre 1997 dans sa version 1.1 comme langage de modélisation des systèmes d'information à objets. L'OMG crée en son sein une Task Force chargée de l'évolution d'UML.

Depuis cette époque, cette Task Force a mis à jour UML plusieurs fois. En mars 2003, la version 1.5 voit le jour. Cette dernière offre la possibilité de décrire des actions grâce à une extension d'UML appelée Action Semantics ou sémantique des actions. La version d'UML en cours à la fin 2006 est UML 2.0 et les travaux d'amélioration se poursuivent.

UML est donc non seulement un outil intéressant mais une norme qui s'impose en technologie à objets et à laquelle se sont rangés tous les grands acteurs du domaine, acteurs qui ont d'ailleurs contribué à son élaboration.

2. Pourquoi modéliser ?

Modéliser un système avant sa réalisation permet de mieux comprendre le fonctionnement du système. C'est également un bon moyen de maîtriser sa complexité et d'assurer sa cohérence.

Un modèle est un langage commun, précis, qui est connu par tous les membres de l'équipe et il est donc, à ce titre, un vecteur privilégié pour communiquer. Cette communication est essentielle pour aboutir à une compréhension commune aux différentes parties prenantes (notamment entre la maîtrise d'ouvrage et maître d'œuvre informatique) et précise d'un problème donné.

Associé au domaine de l'ingénierie du logiciel, un modèle représente l'ensemble des vues sur une expression de besoin ou sur une solution technique permettant de mieux répartir les tâches et d'automatiser certaines d'entre elles. C'est également un facteur de réduction des coûts et des délais.

Le modèle est enfin indispensable au développement logiciel pour assurer un bon niveau de qualité et une maintenance efficace, car un modèle est prévu pour arriver à anticiper les résultats du codage.

3. Les bases d'UML

UML se définit comme un langage de modélisation graphique et textuelle destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML modélise l'ensemble des données et des traitements en élaborant des différents diagrammes.

UML dans sa version 2 propose treize (13) types de diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles :

3.1. Les diagrammes structurels :

Ces diagrammes représentent l'aspect statique d'un système (classes, objets, composants...) :

- **Diagramme de classes** : ce diagramme représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système.
- **Diagramme d'objet** : le diagramme d'objet permet la représentation d'instances des classes et des liens entre instances.
- **Diagramme de composant** : ce diagramme représente les différents constituants du logiciel au niveau de l'implémentation d'un système.
- **Diagramme de déploiement** : ce diagramme décrit l'architecture technique d'un système avec une vue centrée sur la répartition des composants dans la configuration d'exploitation.
- **Diagramme de paquetage** : ce diagramme donne une vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système (classes, composants...).
- **Diagramme de structure composite** : ce diagramme permet de décrire la structure interne d'un ensemble complexe composé par exemple de classes ou d'objets et de composants techniques. Ce diagramme met aussi l'accent sur les liens entre les sous-ensembles qui collaborent.

3.2. Les diagrammes de comportement :

Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par UML :

- **Diagramme des cas d'utilisation** : ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système.
- **Diagramme d'état-transition** : ce diagramme montre les différents états des objets en réaction aux événements.
- **Diagramme d'activités** : ce diagramme donne une vision des enchaînements des

activités propres à une opération ou à un cas d'utilisation. Il permet aussi de représenter les flots de contrôle et les flots de données.

- **Diagramme de séquence** : ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.
- **Diagramme de communication** : ce diagramme est une autre représentation des scénarios des cas d'utilisation qui met plus l'accent sur les objets et les messages échangés.
- **Diagramme global d'interaction** : ce diagramme fournit une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activités.
- **Diagramme de temps** : ce diagramme permet de représenter les états et les interactions d'objets dans un contexte où le temps a une forte influence sur le comportement du système à gérer.

Annexe B

1. Framework Laravel

Laravel est un framework PHP permettant de développer des applications Web en utilisant les concepts du MVC. En juin 2011, la première édition est sortie. Cependant, il est activement amélioré par le fondateur Taylor Otwell et l'équipage. En raison de ses performances, de ses fonctionnalités et de son évolutivité, Laravel est l'un des frameworks PHP les plus connus au monde, et il fait partie des choix des développeurs professionnels. Laravel est tellement facile à personnaliser qu'il est possible de créer sa propre structure de projet pour répondre aux exigences de son application web. Avant de commencer à parler de Laravel en particulier, il est important de bien comprendre l'utilité d'un framework.

1.1. C'est quoi un Framework ?

Un framework est un ensemble cohérent de composants d'outils et de bibliothèques structurés, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'une application. Autrement dit une base homogène avec des briques toutes prêtes à disposition.

L'utilité d'un framework est d'éviter de passer du temps à développer ce qui a déjà été

fait par d'autres, qui sont souvent plus compétents, et qui a en plus été utilisé et validé par de nombreux utilisateurs. Par exemple je dois faire du routage pour mon site, je prends un composant déjà tout prêt et qui a fait ses preuves et je l'utilise : gain de temps, fiabilité, mise à jour si nécessaire, etc

1.2. Pourquoi Laravel ?

Il est considéré comme l'un des meilleurs frameworks PHP en raison des avantages suivants :

- **Technique d'autorisation** : le framework Laravel rend la mise en œuvre des techniques d'authentification très simple. Laravel fournit un moyen très simple d'organiser la logique d'autorisation et de contrôler l'accès à diverses ressources.
- **Bibliothèque orientée objet** : l'une des bibliothèques préinstallées est la bibliothèque d'authentification. Elle possède de nombreuses fonctionnalités avancées telles que la vérification des utilisateurs actifs, le hachage Bcrypt, la réinitialisation du mot de passe, la protection CSRF et le cryptage.
- **Assistance MVC** : Laravel prend en charge l'architecture MVC, assurant la clarté entre la logique et la présentation. MVC aide à améliorer les performances, permet une meilleure documentation et dispose de plusieurs fonctionnalités intégrées.
- **Sécurité de haute technologie** : Laravel s'occupe de la sécurité, il utilise le mot de passe haché, ce qui signifie que le mot de passe ne sera jamais enregistré sous forme de texte brut dans la base de données. Le framework Laravel utilise des instructions SQL préparées qui rendent les attaques par injection inimaginables.
- **Grands tutoriels (Laracasts)** : les développeurs ont toujours besoin d'en savoir plus pour offrir plus. Laravel propose des fonctionnalités Laracasts qui sont un mélange de didacticiels vidéo gratuits et payants qui vous montrent comment utiliser Laravel pour le développement. Les vidéos sont toutes créées par Jeffery Way, un instructeur expert. La qualité vidéo est élevée et les leçons sont bien pensées et significatives.

1.3. Organisation de Laravel

Voici une copie d'écran de la structure principale par défaut d'un dossier dans lequel Laravel est installé :

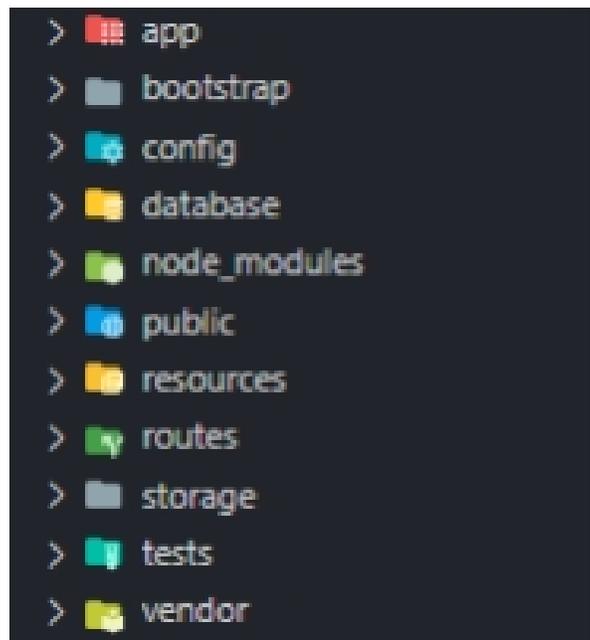


FIGURE 25 – Structure du dossier projet Laravel

- **app** : dossier contenant l'intégralité des fichiers de notre application, c'est-à-dire, tout le code PHP (contrôleurs, modèles, etc.).
- **bootstrap** : il contient le fichier `app.php` qui démarre notre application, ainsi que le cache, pour certaines optimisations.
- **config** : il contient les fichiers de configuration.
- **database** : il permet la gestion de la base de données. Le principal sous-dossier est le sous-dossier `migrations`, qui sont des fichiers permettant de décrire la BDD afin de permettre à Laravel de créer, modifier ou supprimer des tables ou des colonnes.
- **public** : c'est la partie publique du site, il contiendra tout ce qui doit être accessible par le navigateur (JS, CSS, images, vidéo, etc.).
- **resources** : il contiendra les vues qui sont des fichiers majoritairement composés de HTML et sont chargés de la partie affichage de votre site, mais également les dossiers SASS et JS qui sont des fichiers pré-CSS et pré-JS non compilés, et le dossier `lang` qui contient les fichiers de traduction de l'application.
- **routes** : ce dossier contiendra toutes nos définitions de routes de l'application.
- **storage** : il contient les fichiers qui pourraient être générés durant l'utilisation de l'application (logs, fichiers PDF, etc.).
- **tests** : il contient les tests qui sont un moyen rapide et automatisé de vérifier que votre

application fonctionne comme vous le souhaitez. Si votre application commence à grossir, il est important de comprendre comment fonctionnent les tests en programmation.

— **vendor** : dossier contenant toutes les dépendances PHP téléchargées par composer.

1.4. Principales caractéristiques de Laravel

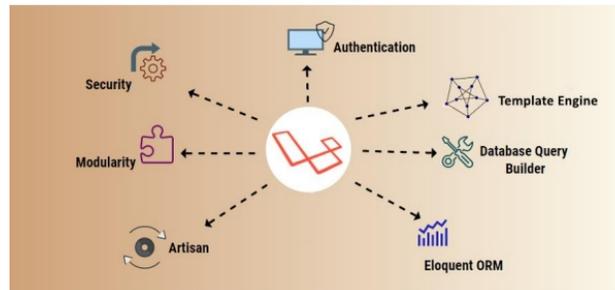


FIGURE 26 – Fonctionnalités principales de Laravel

Le framework de Laravel dispose d'un vaste écosystème avec déploiement immédiat, routage, création de modèles, ORM, requête de base de données, etc. Nous devons voir le nombre de fonctionnalités dans le framework Laravel.

1.4.1. La modularité

La modularité est le degré de séparation et de recombinaison des parties d'une application Web. Avec Laravel on peut facilement créer et concevoir une application d'entreprise de grande envergure à l'aide d'un cadre modulaire. Il propose un guide très simple pour la création dans le Laravel de modules ou de packages.

1.4.2. L'authentification

L'authentification est un composant de chaque application Web. Laravel offre une authentification qui vous permet de créer un schéma d'authentification entièrement fonctionnel à l'aide d'une commande simple. Il fournit également des documents pratiques pour votre propre authentification.

1.4.3. Le routage

Laravel dispose d'un système de cartographie d'itinéraire simple mais puissant entre les méthodes HTTP, les URI d'un côté et les actions (contrôleurs) de l'autre. Il est également possible de séparer les routes en routes API et routes Web.

```
//Accueil*****  
Route::get('AccueilAures', 'AccueilAures@index')-> name('accueil');  
  
// Adminnn *****  
Route::prefix('Admin')->group(function(){  
    Route::get('/Connexion', 'Aures_Admin\AdminConnexion@create') ->name('admin.conx');  
    Route::post('/Connexion', 'Aures_Admin\AdminConnexion@login') ->name('admin.conx.submit');  
    Route::post('/Deconnexion', 'Aures_Admin\AdminConnexion@logoutAdmin') ->name('admin.logout');  
});
```

FIGURE 27 – Structure des routes sur Laravel (connexion administrateur).

1.4.4. La sécurité

Laravel propose des moyens de créer des applications web sûres, tel que l'enregistrement des mots de passe sous forme de hachage en utilisant Bcrypt. Il offre également une sécurité contre les attaques par injection SQL

1.4.5. Le système de migration

Laravel fournit un système de migration pour créer des bases de données. Au lieu d'utiliser SQL, vous pouvez utiliser PHP pour écrire les migrants qui produisent la structure de votre base de données.

```
Schema::create('clients', function (Blueprint $table) {  
    $table->bigIncrements('id_soc');  
  
    $table->unsignedBigInteger('com_id');  
    $table->foreign('com_id')->references('id_com')->on('commerciaux')->onDelete('cascade');  
  
    $table->string('nom_soc')->unique();  
    $table->string('login_soc')->unique();  
    $table->string('password');  
    $table->text('adresse_soc');  
    $table->integer('tel_soc');  
    $table->string('email')->unique();  
    $table->string('logo_soc')->nullable();  
    $table->rememberToken();  
    $table->timestamps();  
});
```

FIGURE 28 – Migration de la table « Client » .

1.4.6. Artisan

Artisan est l'outil de ligne de commande de Laravel. Il comprend plusieurs commandes prédéfinies qui peuvent être utilisées pour effectuer des tâches avec une interface de ligne de commande. Au cours du processus de développement, cet outil peut éviter les tâches les plus répétitives, tels que certaines opérations fonctionnelles importantes, notamment la génération de migrations, la publication d'actifs de packages et de nombreuses tâches similaires.

```
Schema::create('clients', function (Blueprint $table) {
    $table->bigIncrements('id_soc');

    $table->unsignedBigInteger('com_id');
    $table->foreign('com_id')->references('id_com')->on('commerciaux')->onDelete('cascade');

    $table->string('nom_soc')->unique();
    $table->string('login_soc')->unique();
    $table->string('password');
    $table->text('adresse_soc');
    $table->integer('tel_soc');
    $table->string('email')->unique();
    $table->string('logo_soc')->nullable();
    $table->rememberToken();
    $table->timestamps();
});
```

FIGURE 29 – Quelques commandes offertes par Artisan .

1.4.7. Générateur de requêtes de base de données

Le constructeur requêtes de BDD de Laravel offre un moyen simple de créer des requêtes de base de données. Il comprend de nombreuses fonctions d'assistance que vous pouvez utiliser pour filtrer vos données. Les requêtes complexes peuvent facilement être implémentées avec des liens Laravel.

```
Schema::create('clients', function (Blueprint $table) {
    $table->bigIncrements('id_soc');

    $table->unsignedBigInteger('com_id');
    $table->foreign('com_id')->references('id_com')->on('commerciaux')->onDelete('cascade');

    $table->string('nom_soc')->unique();
    $table->string('login_soc')->unique();
    $table->string('password');
    $table->text('adresse_soc');
    $table->integer('tel_soc');
    $table->string('email')->unique();
    $table->string('logo_soc')->nullable();
    $table->rememberToken();
    $table->timestamps();
});
```

FIGURE 30 – Requête SQL pour récupérer l'état des produits réalisé avec Query-builder .

1.4.8. Blade (Template Engine)

Blade est le moteur de modélisation de Laravel pour concevoir un modèle de mise en page. Il propose quelques fonctions d'aide pour formater vos données sur les vues en utilisant également le modèle d'héritage pour créer des mises en page complexes.

```
Schema::create('clients', function (Blueprint $table) {
    $table->bigIncrements('id_soc');

    $table->unsignedBigInteger('com_id');
    $table->foreign('com_id')->references('id_com')->on('commercialis')->onDelete('cascade');

    $table->string('nom_soc')->unique();
    $table->string('login_soc')->unique();
    $table->string('password');
    $table->text('adresse_soc');
    $table->integer('tel_soc');
    $table->string('email')->unique();
    $table->string('logo_soc')->nullable();
    $table->rememberToken();
    $table->timestamps();
});
```

FIGURE 31 – Partie de code de la vue « Profil du Client » écrit avec Blade .

1.4.9. ORM eloquent

Il s'agit d'un outil fourni par Laravel afin que nous puissions utiliser des requêtes simples en remplacement des requêtes SQL. Un exemple d'Eloquent serait le code ci-dessous.

```
Schema::create('clients', function (Blueprint $table) {
    $table->bigIncrements('id_soc');

    $table->unsignedBigInteger('com_id');
    $table->foreign('com_id')->references('id_com')->on('commercialis')->onDelete('cascade');

    $table->string('nom_soc')->unique();
    $table->string('login_soc')->unique();
    $table->string('password');
    $table->text('adresse_soc');
    $table->integer('tel_soc');
    $table->string('email')->unique();
    $table->string('logo_soc')->nullable();
    $table->rememberToken();
    $table->timestamps();
});
```

FIGURE 32 – Modèle de la table « catégorie » écrit en ORM eloquent .

RÉSUMÉ

Ce travail est réalisé en vue de l'obtention du diplôme de master professionnel en génie logiciel.

Il consiste en la conception et la réalisation d'une application web pour la gestion de la relation client.

Notre stage a été effectué au sein de l'entreprise Générale Emballage. L'entreprise rencontre, dans le cadre de son activité commerciale, quelques problèmes de gestion des services liés aux clients. Ces difficultés sont notamment liées à la gestion anarchique des prises de commandes des clients, au suivi aléatoire par les clients et les commerciaux. Ces difficultés ont miné tout effort de satisfaire le client, ainsi que le contrôle des activités commerciales liées au suivi client par les responsables ; c'est pourquoi les dirigeants de l'entreprise aspirent à la mise en place d'un système qui procure aux clients et aux commerciaux un moyen pour faciliter la prise de commande, son suivi et sa gestion. Et aux décideurs les informations nécessaires et fiables, les aidant ainsi à prendre les décisions les plus appropriées.

Dans ce contexte et afin de trouver une solution palliative, nous avons réalisé une application web CRM pour le secteur suivi client au sein de Générale Emballage, facilitant ainsi aux clients et aux commerciaux la réalisation de leurs fonctions respectives et aux dirigeants l'accès et l'exploitation des données, permettant ainsi la prise de bonnes décisions.

Ce travail a été réalisé en utilisant le langage de modélisation Unified Modeling Language (UML), qui permet de schématiser la solution, ainsi que le langage de programmation PHP et MySQL comme système de gestion de base de données (SGBD).

Mots-clés : suivi client, CRM, UML,PHP, MySQL, SGBD

ABSTRACT

This work is carried out for the attainment of the professional master's degree in software engineering.

It involves the design and implementation of a web application for customer relationship management.

Our internship was conducted within the company Générale Emballage. The company faces several issues in managing customer-related services in its commercial activities. These difficulties are particularly related to the disorganized handling of customer orders and the erratic follow-up by both customers and sales representatives. These challenges have undermined all efforts to satisfy the customers as well as to control the commercial activities related to customer follow-up by the managers. Hence, the company's executives aspire to implement a system that provides customers and sales representatives with a means to facilitate order taking, its follow-up, and management, while providing decision-makers with necessary and reliable information, thereby assisting them in making the most appropriate decisions.

In this context, and in order to find a palliative solution, we developed a CRM web application for the customer follow-up sector within Générale Emballage, thus facilitating the execution of their respective functions for both customers and sales representatives, and providing executives with access to and exploitation of data, enabling them to make informed decisions.

This work was carried out using the Unified Modeling Language (UML) for modeling the solution, as well as the PHP programming language and MySQL as the database management system (DBMS).

Keywords : customer follow-up, CRM, UML, PHP, MySQL, DBMS