

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.MIRA-BEJAIA



جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Réf :.....

Faculté Science exacte
Département Informatique

Mémoire

Présenté par
BOUDRAA Fayçal et BOUAZA Mohamed

Pour l'obtention du diplôme de Master

Filière : Informatique

Option : Génie logiciel

Thème

Conception et réalisation d'une application de gestion commerciale d'entreprise

Devant le Jury composé de :

Nom et Prénom

Grade

Mr AMROUN kamal	professeur	Univ. De Bejaia	Président
Mr OUZEGGANE redouane	MAA	Univ. De Bejaia	Examinateur
Mme TAHAKOURT zineb	MAC	Univ. De Bejaia	Encadrante

Année Universitaire : 2021/2022

Remerciements

A l'issue de cette fin de cycle, nous adressons nos remerciements à Dieu tout puissant de nous avoir donné le courage et la volonté de réaliser ce modeste travail.

Et nous tenons à remercier notre honorable promotrice de nous avoir encadrer et orienter durant toute l'année, avec ses compétences, sa patience, ses conseils, sa disponibilité à répondre présente, et surtout pour sa confiance.

Nos remerciements également aux membres du jury d'avoir accepté dévaluer notre travail.

Enfin nos sincères remerciements à toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Dédicaces Fayçal

Je dédie cet humble travail et cet évènement marquant :

À la mémoire de mon exemple éternel, mon cher père

À la lumière de ma vie, ma mère

Mes supports dans la vie qui m'ont tant soutenu et encouragé tout au long de mes études. Et qui ont su m'inculquer le sens de la responsabilité, de l'optimisation et de la confiance en soi. Pour tout l'amour dont vous m'avez entouré, pour vos sacrifices, je ferai de mon mieux pour être un sujet de fierté à vos yeux, et que ce modeste travail soit l'exaucement de vos vœux.

À mon encadrante, je tiens à vous remercier pour votre suivi permanent de mon travail, vos remarques et suggestions qui ont permis à mon travail d'aboutir et de voir le jour.

Boudraa Fayçal

Dédicaces mohamed

C'est avec une très grande joie que je dédie ce travail à mes chers parents pour leurs encouragements, leurs soutiens et leur patience, surtout ma mère qui ne cesse jamais de m'encourager et son soutien qu'elle m'a accordé tout au long de mon chemin.

A mon frère, mes soeurs et toute ma famille.

Et a tous mes amis ou qu'ils soient.

Que DIEU le tout puissant

Soit à vos cotés et vous accorde une meilleure santé

Inchallah.

Bouaza mohamed

LISTE FIGURES

Figure II-1 : Diagramme de contexte	16
Figure II-2 : Cas d'utilisation « S'authentifier ».	21
Figure II-3 : Cas d'utilisation « Gérer les clients ».	22
Figure II-4 : Cas d'utilisation « Changer le profil »	22
Figure II-5 : Diagramme du cas d'utilisation pour responsable.	23
Figure II-6 : Diagramme du cas d'utilisation pour Secrétaire.	24
Figure II-7 : Diagramme du cas d'utilisation pour Chargé clientèle.	24
Figure II-8 : Diagramme du cas d'utilisation pour Chargé approvisionnement.	25
Figure II-9 : Diagramme du cas d'utilisation pour Administrateur.	25
Figure II-10 : Diagramme de séquence système du cas « S'authentifier ».	26
Figure II-11 : Diagramme de séquence système du cas « Ajouter un compte ».	27
Figure II-12 : Diagramme de séquence système du cas « Rechercher un produit ».	28
Figure II-13 : Diagramme de séquence système du cas « Supprimer un produit ».	28
Figure II-14 : Diagramme de séquence système du cas « Modifier un produit ».	29
Figure III-1 : Diagramme d'interaction de cas d'utilisation « S'authentifier ».	32
Figure III-2 : Diagramme d'interaction de cas d'utilisation « Ajouter un produit ».	33
Figure III-3 : Diagramme d'interaction de cas d'utilisation « Modifier un produit ».	34
Figure III-4 : Diagramme d'interaction de cas d'utilisation « Supprimer un produit ».	35
Figure III-5 : Diagramme d'interaction de cas d'utilisation « Rechercher un produit ».	36
Figure III-6 : Diagramme de classe	37
Figure III-7 : Schema de la base données NoSql	38
Figure IV-1 : Schéma illustrant la MERN stack	40
Figure IV-2 : Architecture MVC	42
Figure IV-3 Mappage d'objets entre Node et MongoDB géré via Mongoose	44

Figure IV-4 : Interface login	44
Figure IV-5 : Interfaces articles	45
Figure IV-6 : Interface achats	45
Figure IV-7 : Interface ventes	46
Figure IV-8 : Interface utilisateurs	46
Figure IV-9 : mode sombre	47

LISTE DES TABLEAUX

Tableau II-1 : Identification des messages échangés.	17
Tableau II-2 : Identification des cas d'utilisation	18
Tableau II-3 : Spécification des cas d'utilisation	19

Liste des abréviations

MERN	MongoDB, Express, React, Node
MVC	Model-View-Controller
NoSQL	Not Only SQL
PWD	Password
API	Application Programming Interface
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
JS	Javascript
API	Application Programming Interface
UP	Unified Process
UML	Unified Modeling Language
MEAN	MongoDB, Express, Angular, Node
NPM	Node Package Manager
PDF	Portable Document Format
SQL	Structured Query Language
ORMD	Object Relational Mapping Diagrams
CPU	Central Processing Unit

SOMMAIRE

Remerciement

Dédicaces

Liste des figures

Liste des tableaux

Liste des abréviations

Introduction générale 1

Chapitre I Étude préalable

I.1 Introduction	2
I.2 Entreprise	2
I.2.1 Objectifs	2
I.2.1.1 Les différents métiers dans une entreprise	3
I.2.1.2 La gestion d'une entreprise	4
I.3 Le Commerce	4
I.3.1 La gestion commerciale	4
I.3.1.1 Les enjeux de la gestion commerciale	5
I.3.1.2 Les éléments de la gestion commerciale	5
I.4 Difficultés rencontrées lors de la gestion commerciale d'une entreprise	6
I.5 Solutions	7
I.5.1 Bfgest	8
I.5.2 Applications Web	8
I.5.3 Pourquoi une application web	9
I.6 Cahier des charges	9
I.6.1 Besoins fonctionnels	9
I.6.2 Besoins non-fonctionnels	10

I.7 Processus de développement	10
I.7.1 Processus unifié (Unified Process)	11
I.7.2 UML (Unified Modeling Language)	12
I.8 Conclusion	12

Chapitre II Analyse des besoins

II.1 Introduction	14
II.2 Spécifications des exigences	14
II.3 Identification des acteurs	14
II.4 Interaction entre les acteurs et le système	16
II.5 Identification des messages	17
II.6 Identification des cas d'utilisation	18
II.7 Spécification des cas d'utilisation	19
II.8 Les cas d'utilisation détaillés	21
II.9 Diagramme de cas d'utilisation	23
II.9.1 Diagramme du cas d'utilisation Responsable	23
II.9.2 Diagramme du cas d'utilisation Secrétaire	24
II.9.3 Diagramme du cas d'utilisation Chargé clientèle	24
II.9.4 Diagramme du cas d'utilisation Chargé approvisionnement	25
II.9.5 Diagramme du cas d'utilisation Administrateur	25
II.10 Diagrammes de séquence	26
II.10.1 Diagramme de séquence du cas d'utilisation « S'authentifier »	26
II.10.2 Diagramme de séquence du cas d'utilisation « Ajouter un compte »	27
II.10.3 Diagramme de séquence système du cas d'utilisation « Rechercher un produit »	27
II.10.4 Diagramme de séquence système du cas d'utilisation « Supprimer un produit »	28
II.10.5 Diagramme de séquence système du cas d'utilisation « Modifier un produit »	29
II.11 Conclusion	30

Chapitre III Analyse du domaine et conception

III.1 Introduction	31
III.2 Diagramme d'interactions	31
III.2.1 Les Diagrammes d'interaction des cas d'utilisation	32

III.2.1.1 Diagramme d'interaction de cas d'utilisation « S'authentifier »	32
III.2.1.2 Diagramme d'interaction de cas d'utilisation « Ajouter un utilisateur »	33
III.2.1.3 Diagramme d'interaction de cas d'utilisation « Modifier un produit »	34
III.2.1.4 Diagramme d'interaction de cas d'utilisation « Supprimer un produit »	35
III.2.1.5 Diagramme d'interaction de cas d'utilisation « Rechercher un produit »	36
III.3 Diagramme de classe	37
III.4 Modèle orienté document	37
III.5 Modèle relationnel et modèle orienté document	37
III.6 Conclusion	38

Chapitre IV Réalisation

IV.1 Introduction	39
IV.2 Environnement et outils de développement	39
IV.2.1 Visual Paradigm	39
IV.2.2 Visual Studio Code	39
IV.2.3 MERN stack	39
IV.2.4 Node Package Manager	40
IV.2.5 Front-end	40
IV.2.5.1 React.js	40
IV.2.5.2 Tailwind CSS	40
IV.2.5.3 Formik	41
IV.2.5.4 Axios	41
IV.2.5.5 Vite.JS	41
IV.2.5.6 React-pdf	41
IV.2.6 Back-end	41
IV.2.6.1 Architecture MVC	41
IV.2.6.2 NodeJS	42
IV.2.6.3 ExpressJS	42
IV.2.6.4 PassportJS	42
IV.2.6.5 JOI	43
IV.2.6.6 Dockerfile	43

IV.2.7 Implémentation de la base de données	43
IV.2.7.1 NoSQL	43
IV.2.7.2 MongoDB	43
IV.2.7.3 mongoose	43
IV.3 Principales interfaces	44
IV.4 Conclusion	47
Conclusion générale et perspectives	48
Références bibliographique	
Résumé	

Introduction générale

Aujourd'hui, la gestion des données de manière automatisée a sa place privilégiée dans le monde des entreprises et des organisations. Cette gestion par l'utilisation des technologies de l'information et de la communication apporte beaucoup d'avantages, de recherche et de filtrage des données dans une certaine mesure considérable et la réduction du temps de traitement, permettant aux managers d'améliorer leur performance au travail et éviter également ou du moins réduire les erreurs.

C'est dans ce contexte que plusieurs entreprises tentent de tirer la meilleure partie de ces technologies permettant d'augmenter leur productivité et de faire face à certains problèmes douloureux qui peuvent constituer un obstacle au progrès.

Toute entreprise est prête à investir beaucoup d'argent pour mettre en place une technologie et des logiciels fiables pour améliorer son service, réduire les coûts, augmenter la production et relever les défis du marché. C'est dans ce cadre que s'est inscrit notre projet de fin de cycle qui comprend la conception et réalisation d'une application web de gestion commerciale d'entreprise. Le thème traite de la gestion commerciale et présente diverses fonctionnalités, telles que la gestion du stock, des clients, des fournisseurs, des bon d'achat et de vente, ainsi que l'édition de devis et factures.

Notre système peut constituer une ressource importante pour les outils de reporting et d'analyse décisionnelle pour un meilleur pilotage de l'entreprise

Ce rapport est structuré en 4 chapitres, Le premier chapitre sera consacré à la présentation du contexte général du projet à savoir la gestion commerciale d'entreprise. Le deuxième chapitre portera sur l'analyse des besoins. Le troisième chapitre nous présenterons l'analyse du domaine et conception. Quant au le quatrième chapitre, il sera consacré à la description des différentes étapes de la réalisation de notre travail. Nous terminerons par une conclusion et quelques perspectives.

Chapitre I

Étude préalable

I.1 Introduction

La performance de l'entreprise est la clé de la survie de celle-ci. Son fonctionnement normal nécessite la gestion de l'ensemble de ses ressources humaines, matérielles et logicielles à travers l'ensemble de son système. Nous présenterons dans ce chapitre quelques notions de base de la gestion commerciale et la méthodologie de conception.

I.2 Entreprise

Une entreprise est une organisation dont le but est de produire des biens ou des services afin de les livrer à des clients à des fins d'équilibre financier ou de surplus.

La naissance d'une entreprise est le résultat d'une combinaison de facteurs : une bonne idée, associée à un apport monétaire, humain ou matériel, et beaucoup d'enthousiasme. Les entreprises peuvent exister sous diverses formes publiques ou privées. [1]

I.2.1 Objectifs

L'objectif de l'entreprise est de satisfaire ses clients par des services ce qui permet de créer des emplois. Par le paiement d'impôts et de taxes, l'entreprise participe aussi à la vie collective.

I.2.1.1 Les différents métiers dans une entreprise

Les métiers de domaine de l'administration des entreprises sont nombreux. Voici quelques exemples de professions [2]

➤ Direction

Décider des orientations de l'entreprise, des objectifs à atteindre et de la manière d'y parvenir. Elle joue en quelque sorte un rôle de chef d'orchestre.

➤ Recherche et développement

Étudier, tester et essayer de nouvelles idées, de nouveaux matériaux, de nouvelles formes de production.

➤ Marketing & communication

Définir le produit ou service à vendre, son prix, les moyens de le distribuer et de le faire connaître.

➤ Commercial

Vendre les produits ou services de l'entreprise, trouver les clients et les fidéliser, enregistrer les commandes, argumenter sur la qualité des produits ou services.

➤ Production

Fabriquer un produit, réaliser un service.

➤ Finances & comptabilité

Gérer les factures à payer et à encaisser, effectuer le paiement des charges, des salaires, assurer les relations avec les banques et l'administration.

➤ Service juridique

Rédiger l'ensemble des contrats de l'entreprise, s'assurer de la conformité aux lois et règlements, contrôler le respect des règles commerciales.

➤ Services généraux

Assurer la logistique, acheter les fournitures, garantir la sécurité, effectuer la maintenance des installations, mettre en place et gérer le parc informatique.

➤ Ressources humaines

Organiser et gérer les compétences des hommes et des femmes qui composent l'entreprise, assurer la gestion des contrats de travail, fixer les heures de travail, les congés, prendre en charge le recrutement des salariés, former le personnel, gérer l'évolution des carrières.

I.2.1.2 La gestion d'une entreprise

La gestion correspond, à l'origine, à l'administration des organisations. Elle s'est développée dans les années 1950 pour englober les questions de management et de direction.

La gestion renvoie à la conduite des organisations : c'est l'action ou la manière de gérer, d'administrer, d'organiser quelque chose.

En somme, la gestion c'est l'ensemble des connaissances permettant de conduire une entreprise.

➤ Les fonctions managériales

Ces fonctions de management sont :

- La planification : qui consiste en définition des buts et objectifs de l'organisation ainsi que l'élaboration des plans d'action à suivre.

- L'organisation : sert de la distinction des travaux en tâches et mises en relation des individus.

- Direction : consiste à donner des instructions nécessaires aux membres et leurs motivations.

- Contrôles : Savoir l'exécution des activités selon les plans établis.

I.3 Le Commerce

Le commerce est l'activité de fabrication, de transport et de vente de biens ou de services d'un endroit à un autre en échange de ceux-ci.

I.3.1 La gestion commerciale

La gestion d'entreprise consiste à gérer toutes les opérations d'achat et de vente d'une entreprise. Elle s'appuie sur le traitement et la gestion des produits des données des clients, des fournisseurs et des entreprises. Ce sont donc les services comptabilité, gestion commerciale,

gestion commerciale de l'entreprise qui gèrent les achats, les ventes, les devis, la facturation, la gestion des stocks, la gestion des commandes, etc.

I.3.1.1 Les enjeux de la gestion commerciale

La gestion commerciale revêt une importance significative en assurant plusieurs fonctionnalités

- **Accroissement de la rentabilité**

C'est grâce à une bonne gestion commerciale que les entreprises peuvent honorer les factures, obtenir des paiements des prestataires de services, vendre des produits, effectuer les commandes à temps, etc. la gestion des affaires est un facteur important pour une entreprise, quelles que soient sa taille et son domaine d'activité. Sans une bonne gestion commerciale, les entreprises ne peuvent pas fonctionner correctement, faire des profits et satisfaire leurs clients.

- **Des besoins précis et personnalisés**

Une bonne gestion d'entreprise suppose la prise en compte et le traitement global de toutes les données de l'entreprise. On ne peut donc traiter efficacement la gestion d'entreprise que si l'on dispose des bons outils, totalement intégrés, vraiment complets. L'outil doit permettre de prendre en charge l'ensemble de la structure commerciale de votre entreprise.

I.3.1.2 Les éléments de la gestion commerciale

La gestion commerciale de l'entreprise regroupe différentes actions spécifiques telles que la prévision des ventes, la détermination des prix de vente et le suivi des stocks. Elle permet également un accès rapide aux informations client ou fournisseur. La gestion d'entreprise fournit des indices qui permettent aux dirigeants de faire les bons choix stratégiques. Voici quelques exemples détaillés

- **L'achat**

L'achat ou l'approvisionnement est l'un des piliers de la gestion d'entreprise. Les entreprises doivent avoir une liste de fournisseurs auprès de laquelle s'approvisionner, un inventaire clair et une stratégie de tarification.

C'est le chemin commercial entre l'entreprise et son fournisseur du moment de la passation de commande jusqu'à la livraison.

➤ La vente

C'est l'élément qui repose sur l'entreprise au client et doit donc placer le client au cœur de la stratégie et de la gestion commerciale de l'entreprise. La fonction commerciale ne se limite pas aux ventes et à la livraison dans les délais, il s'agit également d'entretenir de bonnes relations avec les clients et de les fidéliser.

➤ Le stock

Pour qu'une entreprise réussisse, elle doit disposer du bon stock au bon endroit, au bon moment et au meilleur coût, quelle que soit sa taille, une équation incontournable. En fait, tout l'enjeu d'une bonne gestion des stocks est d'éviter les ruptures de stock tout en ayant suffisamment d'espace dans sa zone de stockage pour gagner en efficacité tout en exploitant les surcoûts et dépenses.

➤ La facturation

Dans le cadre d'activités professionnelles, toute vente de biens ou de services doit faire l'objet d'une facturation. Découvrez l'importance des factures La gestion de la facturation est très importante pour les entreprises dans le cadre de la communication avec les clients. En particulier, la facture permet les transactions financières liées aux ventes ou aux services rendus, et elle constitue une pièce justificative comptable et fiscale.

➤ Le bon de commande

Une commande est un document établi pour détailler les articles vendu ou commandé et toutes les conditions utiles à la bonne exécution. Un bon de commande comprend :

- le nom et l'adresse du client ou fournisseur
- la date et le numéro de la commande et mot commande
- le détail et la quantité de l'article commandé
- Prix unitaire et global, conditions de livraison, de paiement et signature de l'acheteur

I.4 Difficultés rencontrées lors de la gestion commerciale d'une entreprise

Pour gérer l'activité globale de la société, chaque département utilise indépendamment des applications informatiques ou autres méthodes. Ceci génère non seulement une difficulté

d'échange de données entre les différents départements, mais également une incohérence de l'information interne. À travers cela on soulève plusieurs problèmes :

➤ Des erreurs dans la saisie des factures

Si l'entreprise ne disposez pas déjà d'un logiciel de gestion commerciale, elle utilise probablement encore des feuilles de calcul pour traiter les devis et les factures. Travailler avec de nombreux fichiers Excel est une tâche ardue qui nécessite un filet de sécurité. Les données doivent être vérifiées régulièrement pour réduire le risque d'erreurs, mais néanmoins, ce type de logiciel est encore trop risqué. Impact sur l'entreprise : perdre beaucoup de temps et les clients sont mécontents.

➤ Perte de clientèle

Une mauvaise gestion des stocks peut entraîner des pertes financières. Sans suivi en temps réel du réapprovisionnement de vos références, l'entreprise ne se rends souvent compte des pénuries que lorsque les clients les demandent. De ce fait, les clients peuvent se tourner vers la concurrence.

A l'inverse, certains des produits peuvent rester en réserve trop longtemps et monopoliser l'espace de stockage pour réduire la rentabilité, sans parler de la trésorerie qui en subit les conséquences.

➤ Une anticipation des besoins difficile

Sans logiciel de gestion d'entreprise, l'entreprise auras peut-être du mal à faire des prévisions. Pourras naviguer à vue sans aucune visibilité sur le développement de l'évènement. Pourtant, l'anticipation et la réactivité sont devenues des compétences essentielles pour reprendre le contrôle de votre entreprise et minimiser les invendus

➤ Des retards de paiement

Ce retard peut engendrer des problèmes de trésorerie. Sans logiciel de gestion commerciale, le délai de traitement des factures est allongé et la trésorerie ne peut pas suivre. On peut difficilement réparer.

I.5 Solutions

➤ Des factures sans erreurs

La facturation permet d'éviter de commettre des erreurs dans l'édition de ses factures et devis afin de se concentrer davantage sur la satisfaction des clients et développer une communication pour vendre plus de produits.

➤ Une gestion des stocks optimisée

Un logiciel de gestion peut prévoir et même automatiser votre réapprovisionnement. Le processus de vente est également optimisé : les bons de commandes, les devis et les factures sont générés automatiquement, et les stocks sont mis à jour en temps réel. Ici aussi l'erreur a disparu. Grâce au logiciel de gestion commerciale, pouvoir limiter la fixation des fonds sans prendre le risque de gérer les ruptures de stock.

➤ Gestion des paiements

En utilisant systématiquement les fonctions liées au paiement, vous pouvez limiter les retards de paiement des clients. L'assurance facture vous permet également d'effectuer plus facilement des transactions.

Pour cela nous supposons la création d'une application qui va répondre à quelques problématiques, nous la nommons « *BFgest* ».

I.5.1 BFgest

BFgest est une application web qui a pour but de créer des solutions plus pertinentes par rapport aux enjeux, et ce afin de faciliter aux entreprises le suivi de leurs achats et vente en toute transparence et faciliter, en garantissant sécurité et convivialité.

I.5.2 Applications Web

En informatique, une application Web (aussi appelée site Web Dynamique ou WebApp) est un logiciel applicatif manipulable grâce à un navigateur Web. De la même manière que les sites web, une application Web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur.

Une application Web est en général basée sur HTML, JavaScript ou CSS. Étant donné qu'une telle application est chargée depuis un serveur et exécutée via un navigateur, il n'y a

donc aucune installation à prendre en charge. Néanmoins l'utilisation d'un signet ou raccourci permet l'accès direct à l'application depuis le bureau ou l'écran d'accueil d'un appareil mobile [3].

I.5.3 Pourquoi une application web

Au cours des dernières années, avec l'amélioration de la sécurité et une technologie de plus en plus flexible, les applications Web se sont développées rapidement. Presque toutes les applications natives peuvent être développées en tant que applications Web et profiter des nombreux avantages offerts par le Web [4] :

- Un accès rapide via divers navigateurs.
- Fonctionne sur tous les systèmes d'exploitation.
- Accessible de partout.
- Travail en simultané dans le « Cloud ».
- Sécurité des contenus.

I.6 Cahier de charges

Nous présentons notre cahier des charges et décrivons les objectifs principaux de l'application que nous comptons développer.

Notre projet porte l'objectif principal qui consiste à concevoir et réaliser une application web d'une plateforme de gestion commerciale. Ce projet vise les entreprises.

I.6.1 Besoins fonctionnels

Les besoins fonctionnels se rapportent aux fonctionnalités que l'application doit offrir pour satisfaire les utilisateurs.

Les fonctionnalités que doit intégrer l'application à développer sont :

- Fidéliser la clientèle
- Gérer la coopération stratégique.
- Mettre en place un suivi commercial efficace.
- Relance, retard ou retard de paiement ou de comptes clients.

- Superviser l'inventaire, les achats et les ventes.
- Problématiques directes liées au financement des entreprises, factures et devis.
- Gérer les différents produits en stock
- Gérer les données des clients, y compris l'historique des relations avec les clients.
- Gérer les relations avec les fournisseurs.

I.6.2 Besoins non-fonctionnels

Les besoins non fonctionnels décrivent les objectifs liés aux performances du système et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d'objectifs spécifiques que doit atteindre le système. Ainsi, notre application doit pouvoir couvrir les besoins non fonctionnels suivants :

- **Maintenabilité** : le code source doit être compréhensible par simple lecture, notamment en respectant les règles de gestion et les normes de développement ;
- **Exploitabilité** : les impacts en termes de performances doivent être pris en compte lors du développement, ainsi que la consommation des ressources (i.e., CPU, mémoire, etc.) qui doit être minimisée ;
- **Convivialité et ergonomie** : le système doit être facilement utilisable et disposer d'interfaces conviviales ; le design doit permettre une identification immédiate de ses différents éléments pour permettre à l'utilisateur d'accéder de manière intuitive à ce qu'il cherche, dès la première utilisation ;
- **Sécurité** : l'obligation pour accéder à l'application de s'authentifier par un nom d'utilisateur et un mot de passe affectés par l'administrateur.

I.7 Processus de développement

Un processus définit une séquence d'étapes, partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objectif d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles. Nous avons choisi de travailler avec le Processus Unifié (UP). En effet, le processus unifié est une méthode de développement logiciel orienté objet adaptée à tout type de projet, et pour la modélisation on a opté pour l'UML qui permet de modéliser les différentes facettes du système durant le développement.

I.7.1 Processus unifié (Unified Process)

Le processus unifié est basé sur l'élargissement et le raffinement d'un système par le biais de multiples itérations, avec une rétroaction et une adaptation cyclique. Le système est développé de manière incrémentale au fil du temps, itération par itération, et cette approche est donc également connue sous le nom de développement logiciel itératif et incrémental. Les itérations sont réparties sur quatre phases, chacune d'entre elles comprenant une ou plusieurs itérations [5] :

- **Inception** : la première et la plus courte des phases du projet. Elle est utilisée pour préparer les bases du projet, y compris la préparation de l'analyse de rentabilisation, l'établissement de la portée du projet et la définition des limites, la description des principales exigences et la solution d'architecture possible ainsi que les compromis de conception, l'identification des risques et le développement du plan de projet initial.

- **Élaboration** : au cours de cette phase, l'équipe de projet doit saisir la majorité des exigences du système (par exemple, sous la forme de cas d'utilisation), effectuer une analyse des risques identifiés et élaborer un plan de gestion des risques afin de réduire ou d'éliminer leur impact sur le calendrier et le produit final, établir la conception et l'architecture (par exemple, en utilisant des diagrammes de classe de base, des diagrammes de paquetage et des diagrammes de déploiement), créer un plan (calendrier, estimations de coûts et étapes réalisables) pour la phase suivante (construction).

- **Construction** : la phase la plus longue et la plus importante du processus unifié. Pendant cette phase, la conception du système est finalisée et affinée et le système est construit en utilisant la base créée pendant la phase d'élaboration. La phase de construction est divisée en plusieurs itérations, chaque itération devant aboutir à une version exécutable du système. L'itération finale de la phase de construction libère un système entièrement terminé qui sera déployé pendant la phase de transition.

- **Transition** : la phase finale du projet qui fournit le nouveau système à ses utilisateurs finaux. La phase de transition comprend également la migration des données des anciens systèmes et la formation des utilisateurs.

Chaque phase et son itération consistent en un ensemble d'activités prédéfinies. Le processus unifié décrit les activités de travail comme des disciplines :

- **Modélisation du business** : modélisation des objets du domaine et modélisation dynamique des processus du business.

- **Exigences** : analyse des exigences du système considéré. Cela comprend des activités telles que la rédaction de cas d'utilisation et l'identification des exigences non fonctionnelles.

- **Analyse et conception** : couvre les aspects de la conception, y compris l'architecture globale.

- **Mise en œuvre** : programmation et construction du système (sauf le déploiement).

- **Test** : comprend les activités de test telles que la planification des tests, le développement de scénarios de test, les tests alpha et bêta, les tests de régression, les tests d'acceptation et le déploiement.

- **Déploiement** : les activités de déploiement du système développé.

I.7.2 UML (Unified Modeling Language)

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

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet [6].

I.8 Conclusion

Une gestion d'entreprise réussie n'est pas seulement une question de chiffres et de besoins financiers. En fait, il est important de ne pas négliger les relations avec les employés, les clients et les fournisseurs. C'est un élément très important qui contribue à renforcer la performance de la stratégie appliquée. Par exemple, des clients insatisfaits sont tombés avec des produits

défectueux. Ce chapitre nous a donné une idée de la gestion commerciale au sein de l'entreprise, ce qui nous aide dans la partie implémentation de l'application.

Chapitre II

Analyse des besoins

II.1 Introduction

L'analyse et la spécification des besoins représentent la première phase du cycle de développement d'un projet informatique. Il identifie les participants réactifs du système et chaque ensemble d'actions réalisées sur eux. Alors, Ce chapitre est consacré à la définition des acteurs et de leurs besoins. Ensuite, nous allons modéliser le diagramme de cas d'utilisation général.

II.2 Spécifications des exigences

L'objectif principal de notre application est de faciliter la tâche aux employés de l'entreprise et de leurs faire gagner beaucoup de temps en leur offrant une interface claire et simple d'utilisation. Pour ce faire, notre système doit :

1. Fournir une interface simple et conviviale pour une meilleure utilisation de l'application.
2. Permettre de gérer différents produits.
3. Permettre de gérer les clients.
4. Permettre de gérer les fournisseurs.
5. Permettre de générer et modifier des devis, des bons de commande et des factures.

II.3 Identification des acteurs

Un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui Interagit avec le système modélisé.

Les acteurs qui interagissent dans notre système sont : le responsable, la secrétaire, le chargé clientèle, le chargé approvisionnement et l'administrateur.

➤ Le responsable : c'est celui qui est responsable des relations avec les clients et les opérations d'achats, son rôle est :

- Authentification.

- Consulter les statistiques.

•Le secrétaire : c'est l'acteur responsable de la gestion des produits, des clients et des fournisseurs, son rôle est :

- Authentification.

- Gérer les articles (ajouter, rechercher, modifier, supprimer).

- Gérer les clients (ajouter, rechercher, modifier, supprimer).

- Gérer les fournisseurs (ajouter, rechercher, modifier, supprimer).

➤ Le chargé clientèle : comme son nom l'indique cet acteur est celui qui prend en charge les commande, les devis et les factures des clients, son rôle est :

- Authentification.

- Etablir des devis clients (ajouter, modifier, supprimer).

- Gérer les commandes des clients (ajouter, modifier, supprimer).

- Gérer les factures des clients (ajouter, modifier, supprimer).

➤ Le chargé approvisionnement : est l'acteur qui s'occupe de l'établissement et la gestion des devis fournisseurs, gère les commandes et les factures de ces fournisseurs, son rôle est :

- Authentification

- Etablir des devis fournisseurs (ajouter, modifier, supprimer).

- Gérer les commandes des fournisseurs (ajouter, modifier, supprimer).

- Gérer les factures des fournisseurs (ajouter, modifier, supprimer).

➤ L'administrateur : c'est celui qui veille sur toutes les mises à jour qui concernent l'application web, son rôle est : son rôle est :

- Authentification.
- Gérer les profils.

II.4 Interaction entre les acteurs et le système

➤ Le diagramme de contexte

La Figure II-1 permet donc de définir les frontières de l'étude et de préciser la phase du cycle de vie dans laquelle on situe l'étude (généralement la phase d'utilisation).

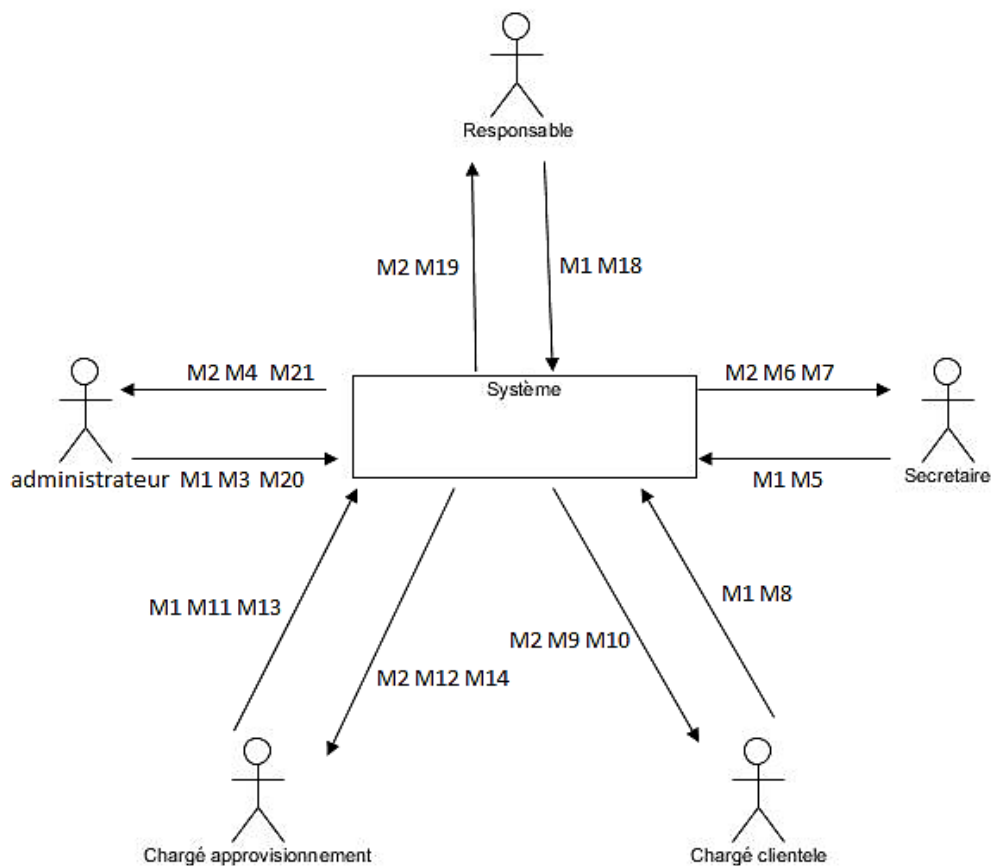


Figure II-1 : Diagramme de contexte

II.5 Identification des messages

Le Tableau II-1, nous présente les différents messages échangés entre le système et les acteurs

N°	Message Acteur->Système	N°	Message Système->Acteur
M1	Demande d'authentification	M2	Affichage de l'interface d'accueil correspondant à chaque acteur
M3	Demande de création de compte utilisateur	M4	Affichage de l'interface de création compte utilisateur
M5	Demande ajout d'un client	M6	Affichage de l'interface d'ajout
		M7	Client ajouté
M8	Demande recherche d'une facture client	M9	Affichage de la facture client recherché
		M10	Affichage des détails de la facture client
M11	Demande de suppression d'une facture fournisseur	M12	Affichage du message de confirmation
M13	Confirmation de la suppression	M14	Notification de suppression
M15	Demande de modification d'un devis fournisseur	M16	Affichage de l'interface de modification
		M17	Devis modifié
M18	Demande de consultation des statistiques	M19	Affichage de l'interface de consultation
M20	Demande de l'interface de la gestion des profils	M21	Affichage de l'interface de la gestion des profils

II.6 Identification des cas d'utilisation

Le Tableau II-2 présentes les différents cas d'utilisation de chaque acteur

Acteur	Tâches
Responsable	T0 : S'authentifier.
	T1 : Consulter les statistiques
Secrétaire	T2 : S'authentifier.
	T3 : Gérer les produits.
	T4 : Gérer les clients.
	T5 : Gérer les fournisseurs.
Chargé clientèle	T6 : S'authentifier
	T7 : Gérer les devis des clients.
	T8 : Gérer les commandes des clients.
	T9 : Gérer les factures des clients.
Chargé approvisionnement	T10 : S'authentifier.
	T11 : Gérer les devis des fournisseurs.
	T12 : Gérer les commandes des fournisseurs.
	T13 : Gérer les factures des fournisseurs.
Administrateur	T14 : S'authentifier.
	T15 : Gérer les profils

II.7 Spécification des cas d'utilisation

Le **Tableau II-3** présentes les différents scénarios de cas d'utilisation de chaque acteur

Acteur	Tache	Scenario
Responsable	T1 : S'authentifier.	S1 : Saisir le login et le mot de passe dans la page d'accueil. S2 : Cliquer sur le bouton 'connexion'.
	T2 : Consulter les statistiques.	S3 : Sélectionner le lien ' Consulter les statistiques'. S4 : Ajouter, modifier, supprimer ou rechercher des statistiques.
Secrétaire	T3 : S'authentifier.	S5 : Saisir le login et le mot de passe dans la page d'accueil. S6 : Cliquer sur le bouton 'connexion'.
	T4 : Gérer les produits.	S7 : Sélectionner le lien ' Gestion des produits '. S8 : Ajouter, modifier, supprimer ou rechercher des produits.
	T5 : Gérer les clients.	S9 : Sélectionner le lien ' Gestion des clients '. S10 : Ajouter, modifier, supprimer ou rechercher des clients.
	T6 : Gérer les fournisseurs.	S11 : Cliquer sur le lien ' Gestion des fournisseurs '. S12 : Ajouter, modifier, supprimer ou rechercher des fournisseurs.
Chargé clientèle	T7 : S'authentifier.	S13 : Saisir le login et le mot de passe dans la page d'accueil.

		S14 : Cliquer sur le bouton ‘connexion’.
	T8 : Gestion des devis client.	S15 : Cliquer sur le lien ‘ Gestion des devis fournisseur’. S16 : Ajouter, modifier, supprimer ou rechercher des devis client.
	T9 : Gestion des commandes client.	S17 : Cliquer sur le lien ‘ Gestion des commandes client ’. S18 : Ajouter, modifier, supprimer ou rechercher des commandes client.
	T110 : Gestion des factures Client.	S17 : Cliquer sur le lien ‘ Gestion des factures client’. S18 : Ajouter, modifier, supprimer ou rechercher des factures client.
Chargé approvisionnement	T11 : S’authentifier	S19 : Saisir le login et le mot de passe dans la page d’accueil. S20 : Cliquer sur le bouton ‘connexion’.
	T12 : Gérer les devis fournisseur.	S19 : Cliquer sur le lien ‘ Gestion des devis fournisseur’. S20 : Ajouter, modifier, supprimer ou rechercher des devis fournisseur.
	T13 : Gérer les commandes fournisseur.	S21 : Cliquer sur le lien ‘ Gestion des commandes fournisseur’. S22 : Ajouter, modifier, supprimer ou rechercher des commandes fournisseur.
	T14 : Gérer les factures fournisseur.	S23 : Cliquer sur le lien ‘ Gestion des factures fournisseur’. S24 : Ajouter, modifier, supprimer ou rechercher des factures fournisseur.
Administrateur	T15 : S’authentifier.	S25 : Saisir le login et le mot de passe dans la page d’accueil.

		S26 : Cliquer sur le bouton 'connexion'.
	T16 : Gérer les profils.	S27 : Cliquer sur le lien ' Gestion des profils ' S28 : Ajouter, modifier, supprimer ou rechercher des profils.

II.8 Les cas d'utilisation détaillés

Les figures suivantes décrivent certains cas d'utilisation de notre système

- Cas d'utilisation « s'authentifier » :

<p>Utilisation : S'authentifier.</p> <p>Scénario : S1, S2.</p> <p>Acteur : Responsable.</p> <p>Résumé : Cette fonctionnalité permet au responsable d'accéder à son espace personnel.</p> <p>Description :</p> <ol style="list-style-type: none"> 1. Accéder à l'application. 2. Le système affiche la page d'accueil. 3. Le responsable clique sur le lien « connexion » 4. Le responsable saisie le login et le mot de passe puis valide en cliquant sur le bouton « Connexion ». 5. Le système vérifie les données, les compare avec celles de la base de données puis affiche l'interface correspondante, ou renvoie un message d'erreur si le login et/ou le mot de passe ne sont pas valides.
--

Figure II-2 : Cas d'utilisation « S'authentifier ».

- Cas d'utilisation « Gérer les clients » :

Utilisation : Gérer les clients.

Scénario : S9, S10.

Acteur : Secrétaire.

Résumé : Cette fonctionnalité permet au secrétaire d'ajouter, modifier, supprimer ou de rechercher des clients.

Description :

1. L'utilisateur accède à son espace par l'authentification.
2. Le système affiche la page d'accueil.
3. Le secrétaire clique sur le lien « Client ».
4. Le système affiche la page correspondante.
5. Le secrétaire ajoute ou modifie les clients puis valide dans les coins correspondants.
6. Le système enregistre le changement et lui redirige vers la liste des clients en tenant compte de la modification

Figure II-3 : Cas d'utilisation « Gérer les clients ».

- Cas d'utilisation « Gérer les profil » :

Utilisation : gérer les profils.

Scénario : S27, S28.

Acteur : Administrateur.

Résumé : Cette fonctionnalité permet à l'utilisateur de gérer ses profils.

Description :

1. L'utilisateur accède à son espace.
2. L'utilisateur clique sur le lien « Profil ».
3. L'utilisateur clique sur le lien « Gérer les profil ».
4. Le système affiche la page correspondante.
5. L'utilisateur choisit les informations à changer et remplit les champs correspondants, puis valide.

Figure II-4 : Cas d'utilisation « Changer le profil »

II.9 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet.

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation [7].

II.9.1 Diagramme du cas d'utilisation Responsable

La Figure II-5 représente le diagramme de cas d'utilisation du responsable

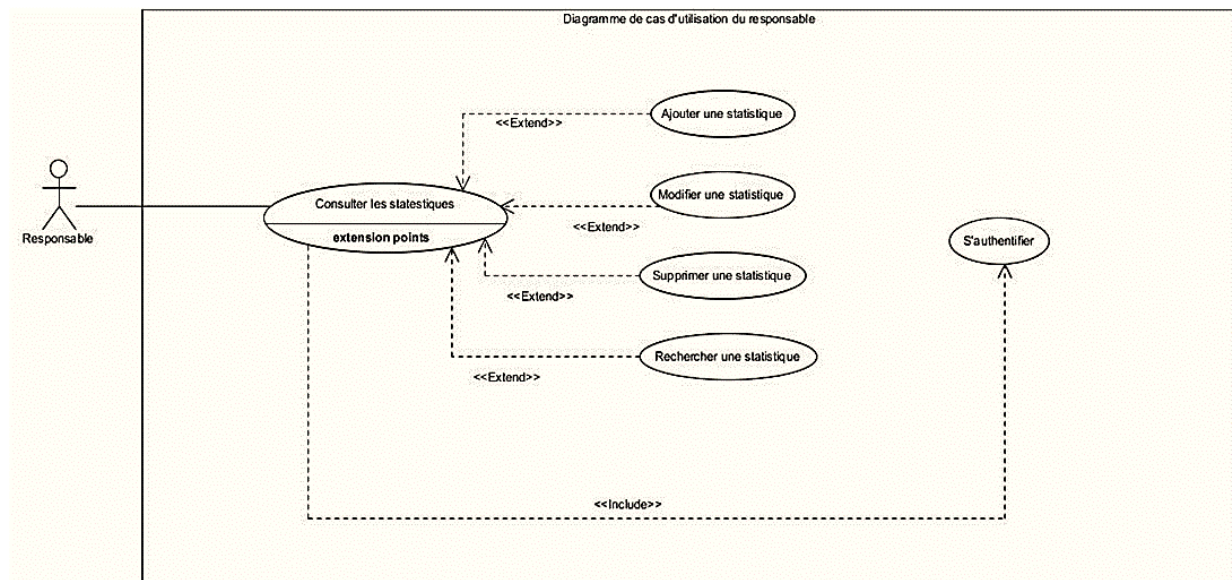


Figure II-5 : Diagramme du cas d'utilisation du « Responsable ».

II.9.2 Diagramme du cas d'utilisation Secrétaire

La Figure II-6 représente le diagramme de cas d'utilisation de la secrétaire

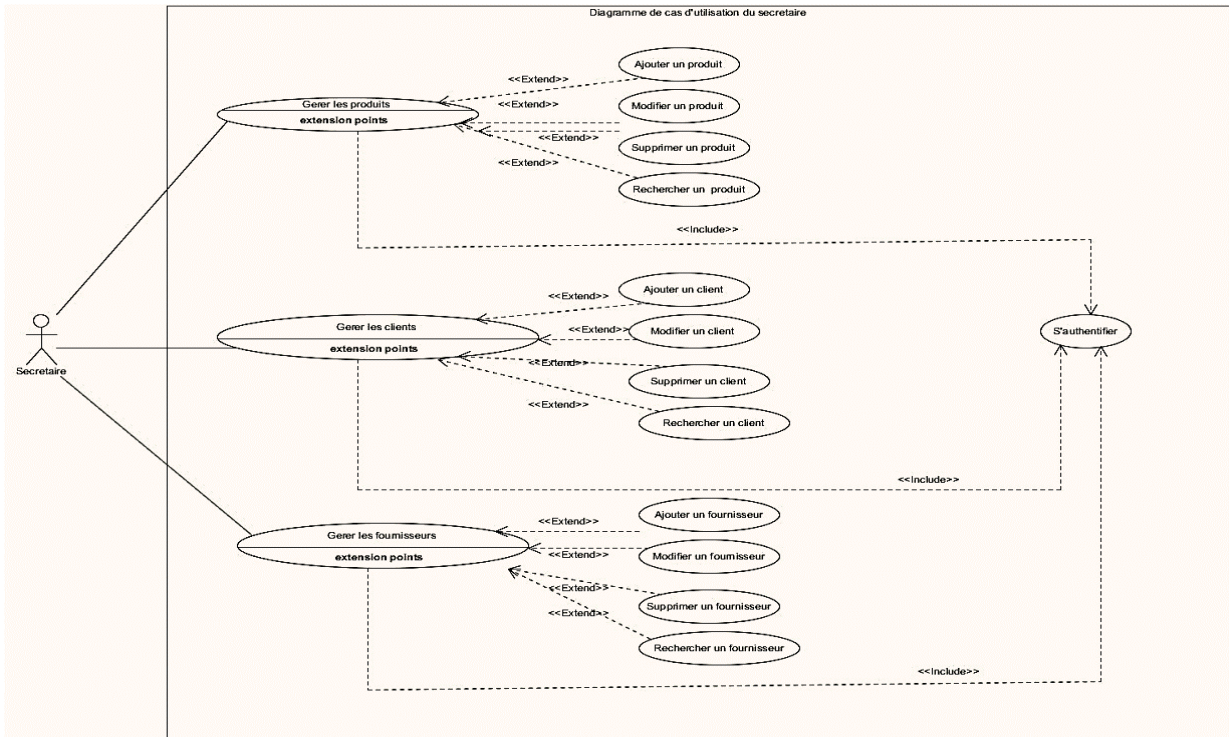


Figure II-6 : Diagramme du cas d'utilisation de « Secrétaire ».

II.9.3 Diagramme du cas d'utilisation Chargé clientèle

La Figure II-7 représente le diagramme de cas d'utilisation du Chargé clientèle

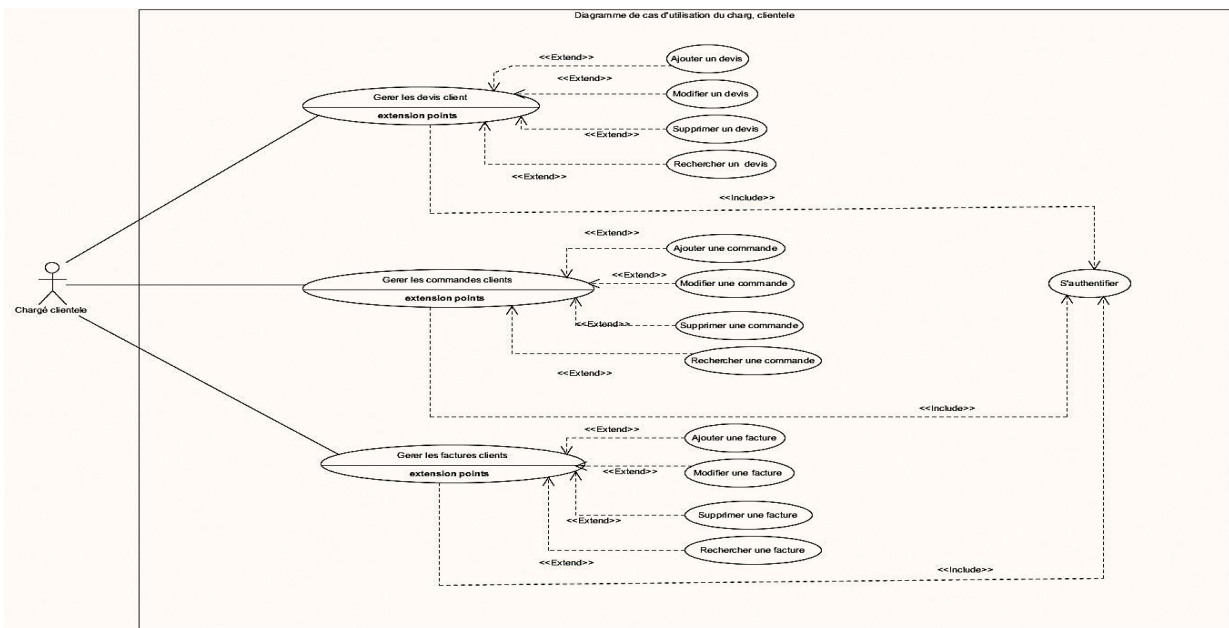


Figure II-7 : Diagramme du cas d'utilisation du « Chargé clientèle ».

II.9.4 Diagramme du cas d'utilisation Chargé approvisionnement

La Figure II-8 représente le diagramme de cas d'utilisation du Chargé approvisionnement

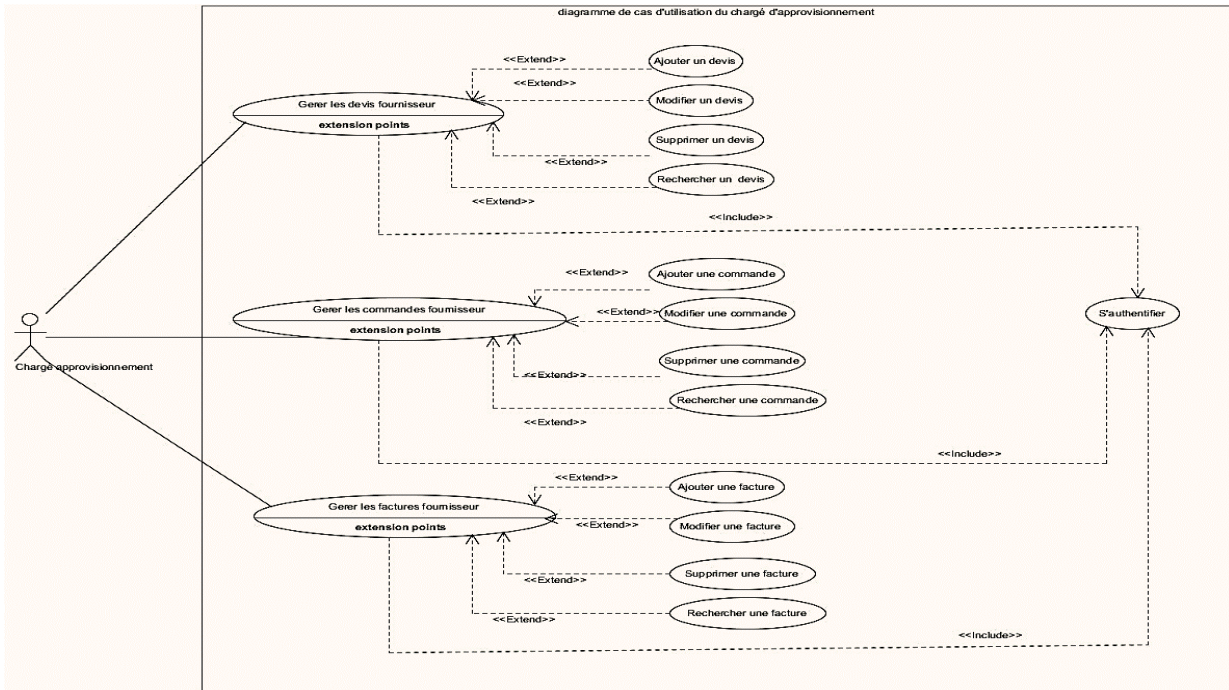


Figure II-8 : Diagramme du cas d'utilisation du « Chargé approvisionnement ».

II.9.5 Diagramme du cas d'utilisation Administrateur

La Figure II-9 représente le diagramme de cas d'utilisation de l'administrateur

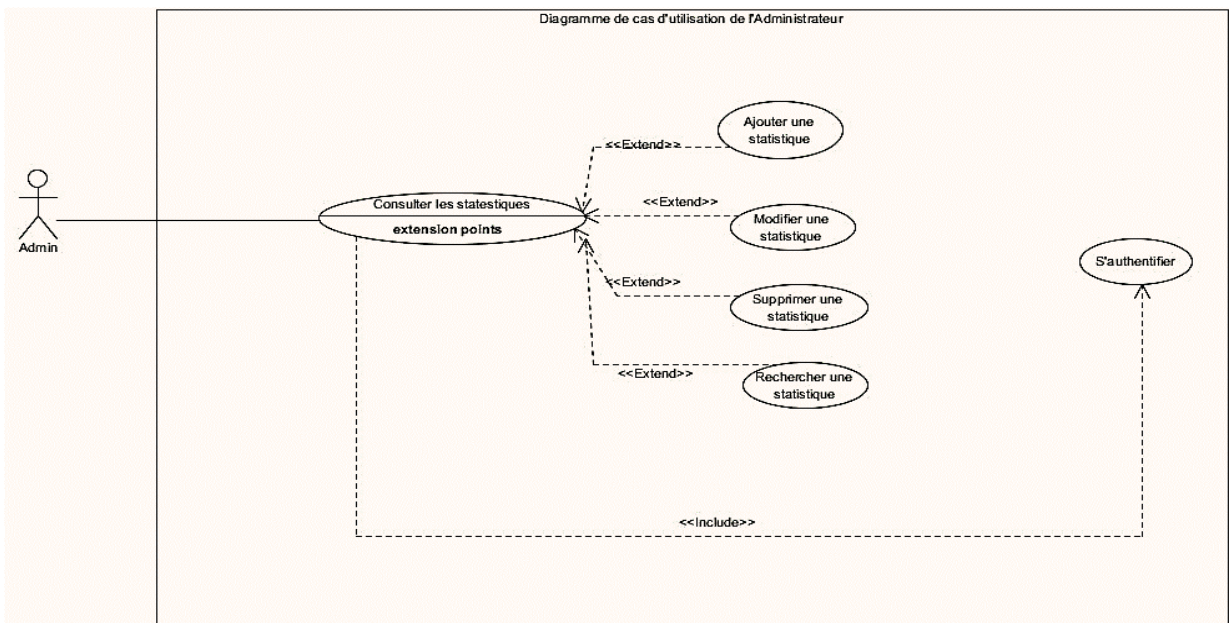


Figure II-9 : Diagramme du cas d'utilisation de « l'administrateur ».

II.10 Diagrammes de séquence

Après avoir décrit les différents cas d'utilisation de notre système avec des mots, nous les présenterons formellement à l'aide de diagrammes de séquence. Les diagrammes de séquence donnent une représentation temporelle des interactions entre objets (on parle aussi de rôles plutôt que d'objets). La formalisation des messages est une étape importante du passage aux objets car elle précise les responsabilités et les interfaces des objets. C'est-à-dire, ce que d'autres objets peuvent leur demander, et quel type de réponse ils promettent de fournir aux requêtes qui leur sont faites. En ce qui suit, nous présenterons quelques diagrammes de séquences relatifs aux cas d'utilisation présentés [8]

II.10.1 Diagramme de séquence du cas d'utilisation « S'authentifier »

Pour accéder à son espace de travail, l'utilisateur saisit son identifiant et son mot de passe et authentifie l'action, le système effectue la vérification. Si l'un des champs est vide ou incorrect, envoyer un message d'erreur, sinon afficher l'interface correspondante pour l'utilisateur.

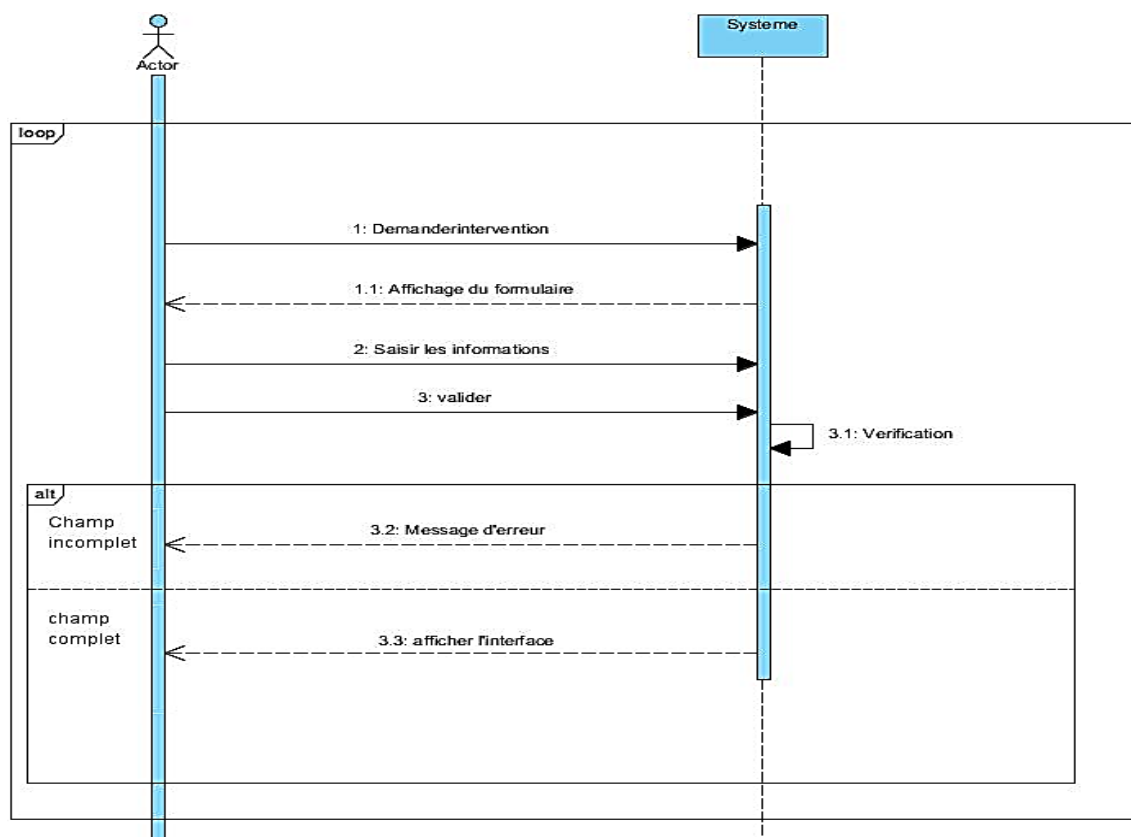


Figure II-10 : Diagramme de séquence système du cas « S'authentifier ».

II.10.2 Diagramme de séquence du cas d'utilisation « Ajouter un compte »

Dans notre système, on a le secrétaire qui gère les produits (ajouter, modifier, supprimer, rechercher). Pour ajouter un compte le secrétaire commence par s’authentifier puis demander il le formulaire d’ajout et saisie les données. Le système renvoi la confirmation d’ajout si les champs sont complètement remplis ou un message d’erreur sinon.

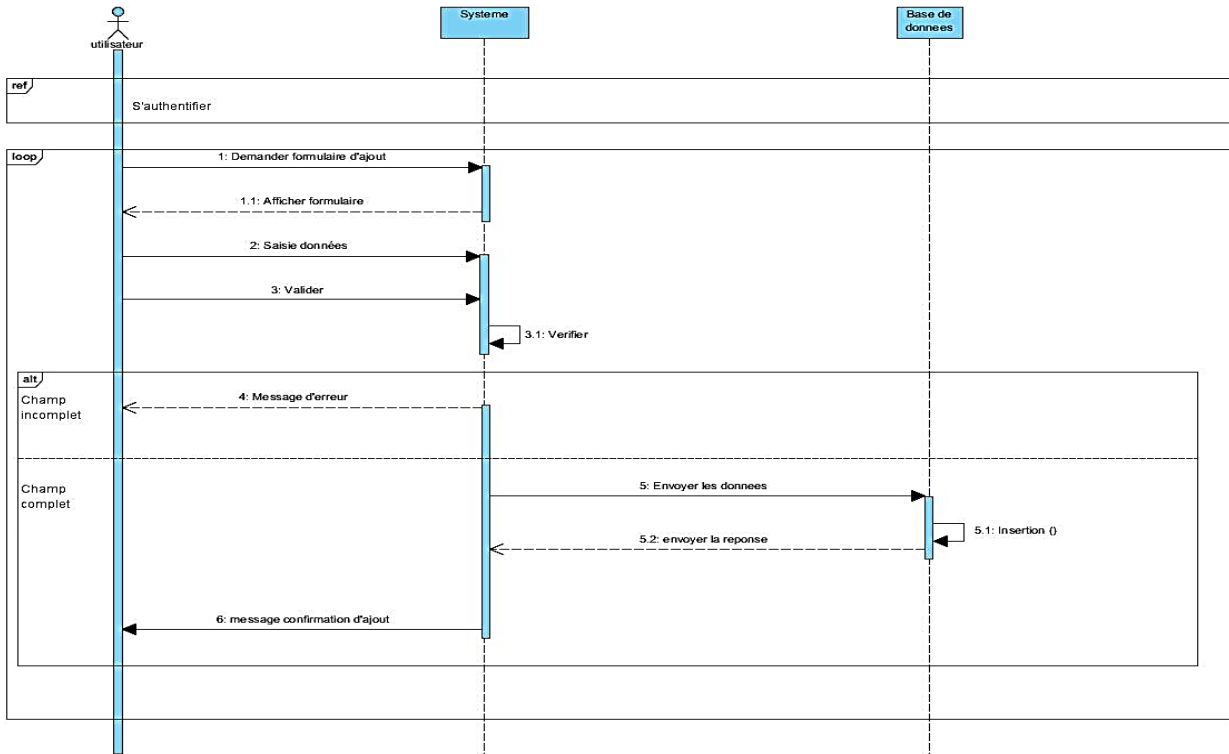


Figure II-11 : Diagramme de séquence système du cas « Ajouter un compte ».

II.10.3 Diagramme de séquence système du cas d'utilisation « Rechercher un produit »

Pour rechercher un compte le secrétaire commence par s’authentifier puis demander la liste des produit, saisie ce dernier recherché, après la vérification le système renvoi la liste désirée ou ‘produit non existant sinon.

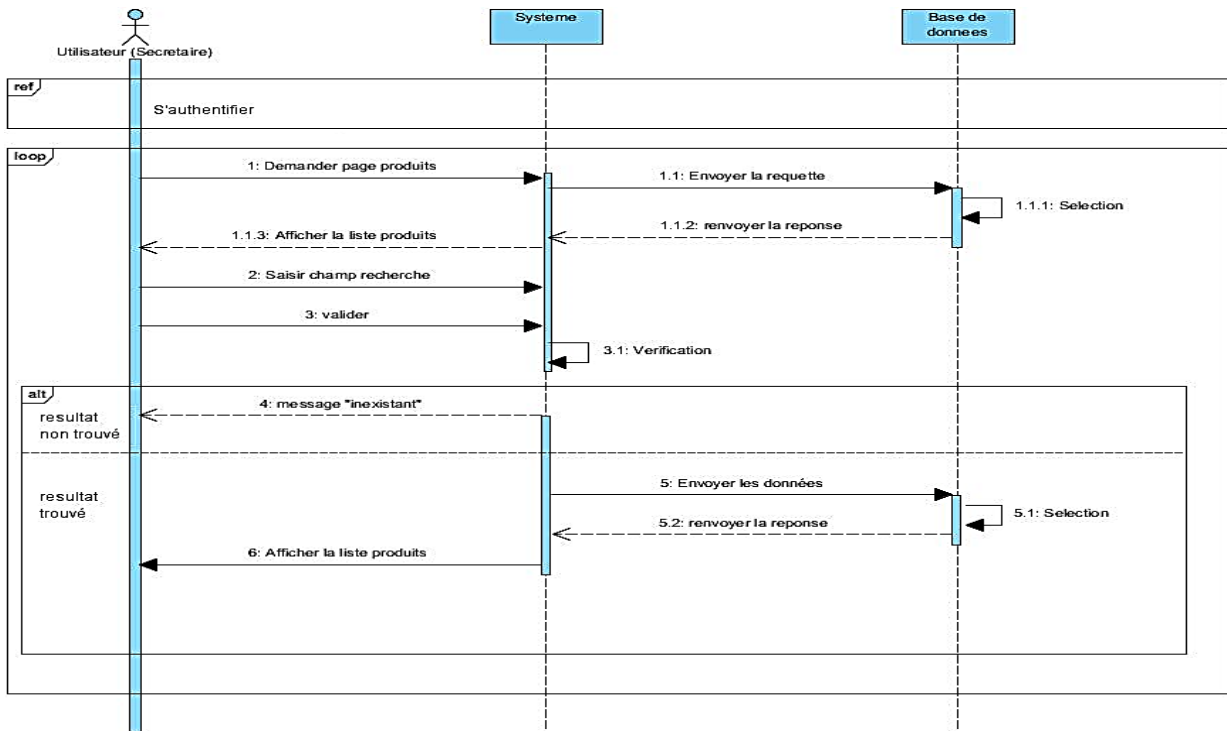


Figure II-12 : Diagramme de séquence système du cas « Rechercher un produit ».

II.10.4 Diagramme de séquence système du cas d’utilisation «Supprimer un produit»

Une opération similaire à la précédente, le secrétaire s’authentifie, recherche et sélectionne ce produit à supprimer, confirme son choix pour que la suppression s’effectue, ou annule la suppression et l’opération s’arrête et le système rafraichie la page.

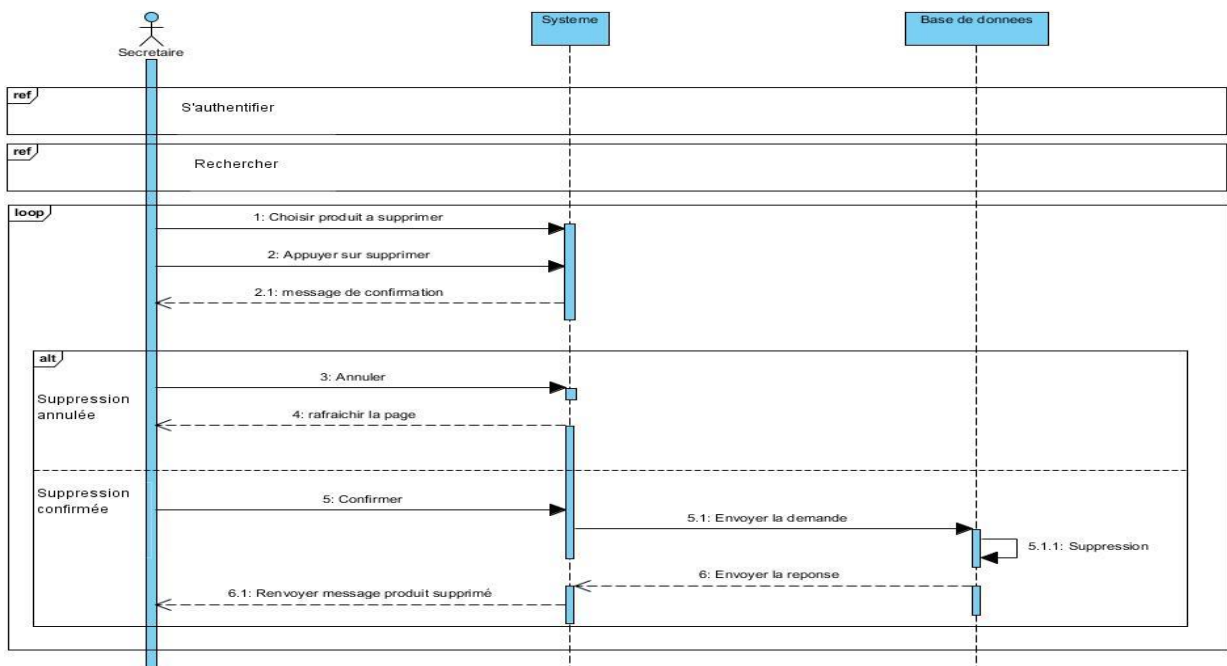


Figure II-13 : Diagramme de séquence système du cas « Supprimer un produit ».

II.10.5 Diagramme de séquence système du cas d'utilisation « Modifier un produit »

Le secrétaire a le privilège de modifier un produit. Commence par s'authentifier, rechercher et sélectionne le produit qu'il désire modifier, le système envoie la requête à notre base de données qui va le sélectionner et renvoie les détails. Le secrétaire modifie et ressaisit les données, le système renvoie le message de confirmation de cette modification si les champs sont complets ou le message d'erreur sinon.

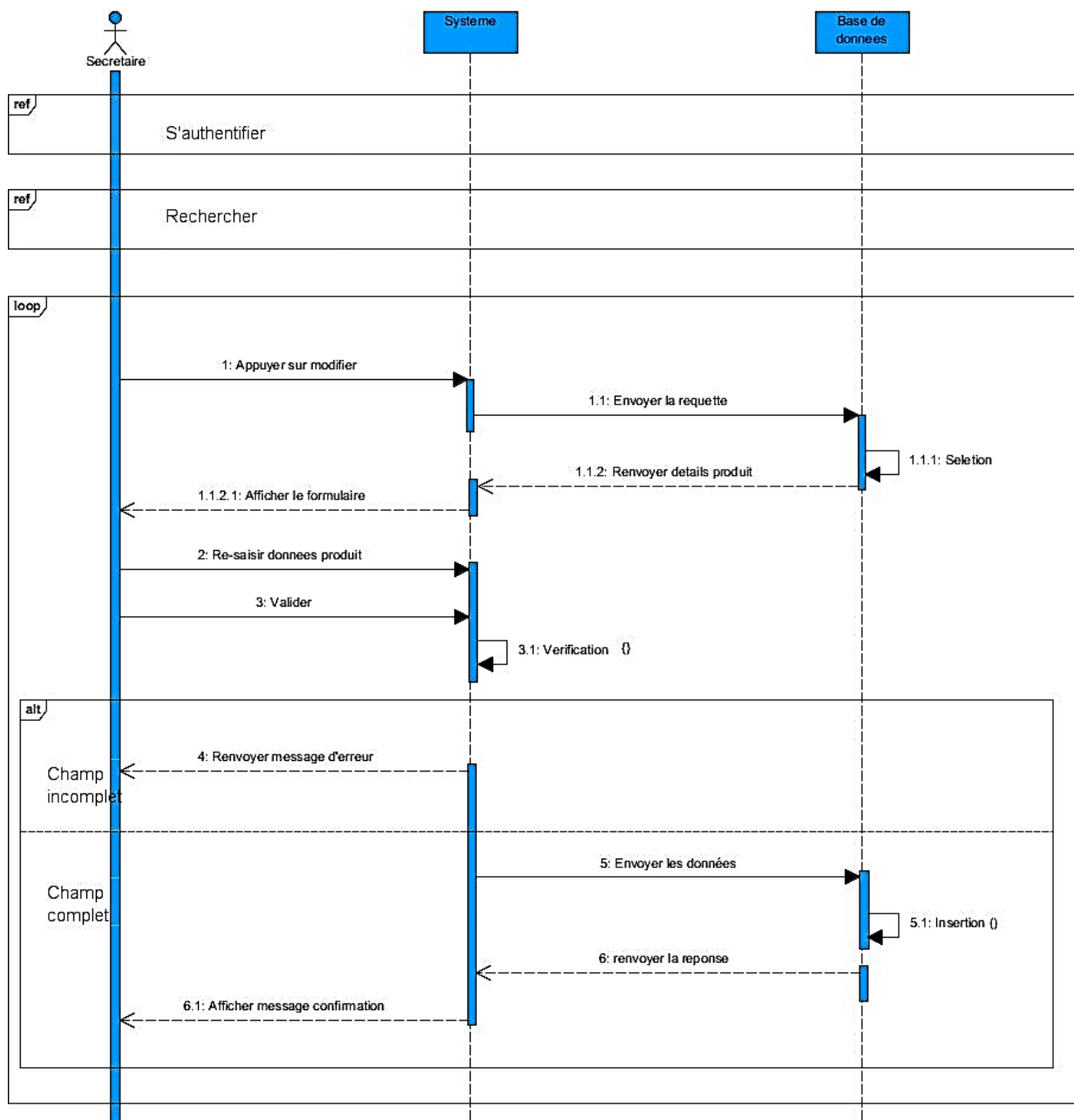


Figure II-14 : Diagramme de séquence système du cas « Modifier un produit ».

II.11 Conclusion

Dans ce chapitre, nous avons analysé les exigences de l'application à l'aide de diagrammes de cas d'utilisation, suivis de diagrammes de séquence du système. Cette phase d'analyse permet de décrire les besoins de l'utilisateur, les fonctionnalités attendues du système de manière globale pour faciliter sa mise en place et sa maintenance. Dans le chapitre suivant, nous commencerons une phase très importante où nous décrirons en détail comment implémenter ces exigences dans notre application.

Chapitre III

Analyse du domaine et conception

III.1 Introduction

Pour assurer la bonne organisation et la maîtrise du travail, tous les processus dans le cycle du développement logiciel ou système informatique doit suivre et se baser sur une démarche ou méthode bien définies, trois phases importantes dont la conception qui est une étape importante de réflexion, après l'analyse et la spécification des besoins.

III.2 Diagramme d'interactions

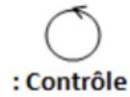
Un diagramme d'interaction donne un aperçu et expose les différents scripts et scenario d'utilisation du système, nous allons édifier un diagramme d'interaction pour chaque diagramme de séquence système déjà établi et qui va inclure des objets représentés par des lignes de vie et messages échangés lors des interactions.

On a trois types de classes dans les diagrammes d'interaction :

- Classes d'interface (boundary) on en trouve au moins une de ces classes dans chaque cas d'utilisation, elle permet l'interaction entre l'utilisateur et l'application. Cette classe est schématisée comme suit :



Classes de Contrôle (Control) ces classes contiennent les traitements et la cinématique de l'application, chargée de transition entre les classes d'interface et les classes entités. Schématisées comme suit



- Classes entités (entity) ça concerne les objets métier qui sont des entités persistantes, et leurs informations (données) seront conservées après l'exécution d'un cas d'utilisation spécifique. Généralement, ils sont conservés dans une base de données. Représenter par ce stéréotype :



III.2.1 Les Diagrammes d'interaction des cas d'utilisation

III.2.1.1 Diagramme d'interaction de cas d'utilisation « S'authentifier »

L'authentification est la première étape pour accéder à l'application, là où l'utilisateur doit saisir son LOGIN et PWD pour profiter des fonctionnalités.

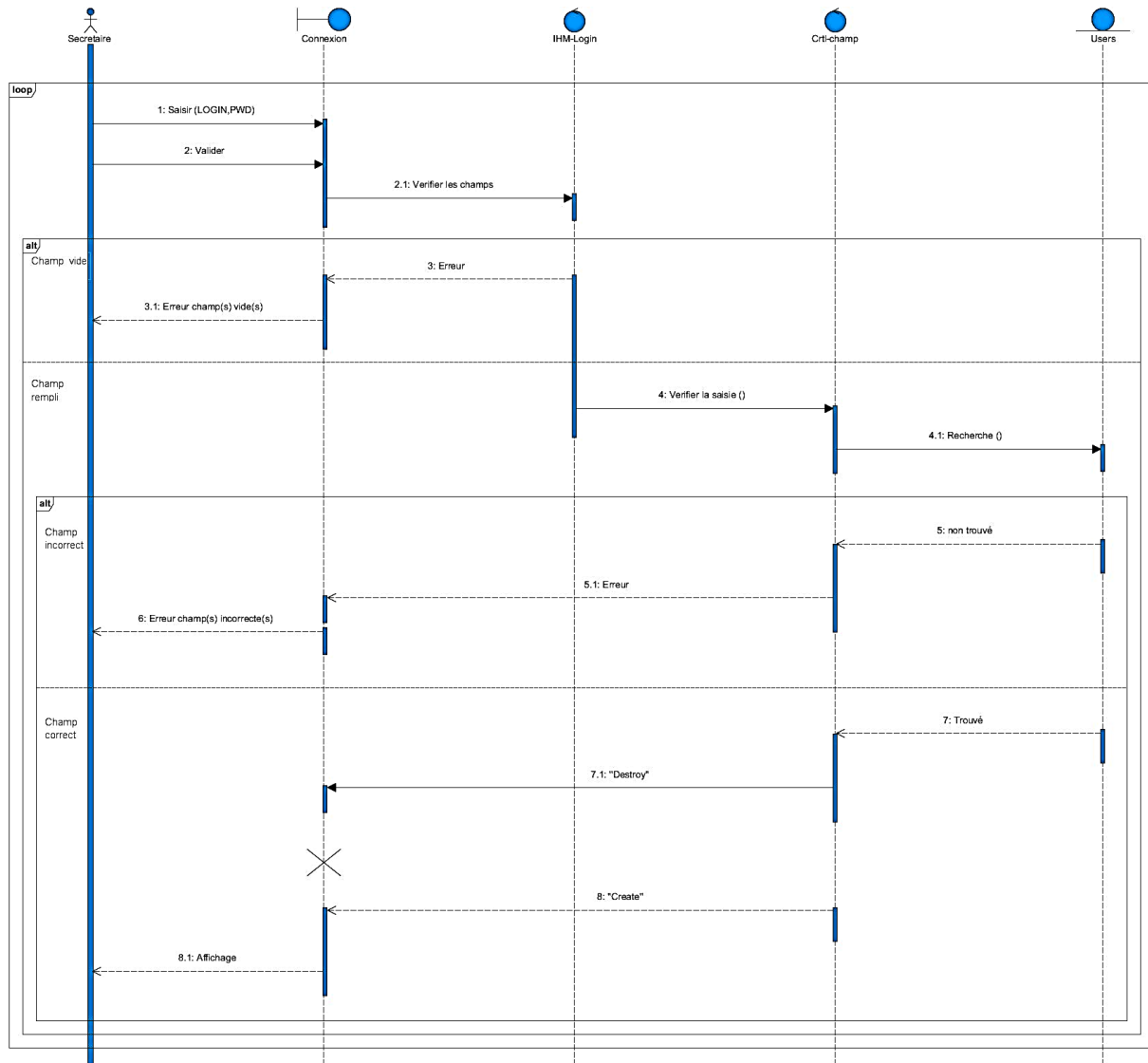


Figure III-1 : Diagramme d'interaction de cas d'utilisation « S'authentifier ».

III.2.1.2 Diagramme d'interaction de cas d'utilisation « Ajouter un utilisateur »

Le Secrétaire qui gère les clients et fournisseurs a comme fonctionnalité l'ajout d'utilisateur qui peut être client ou fournisseur.

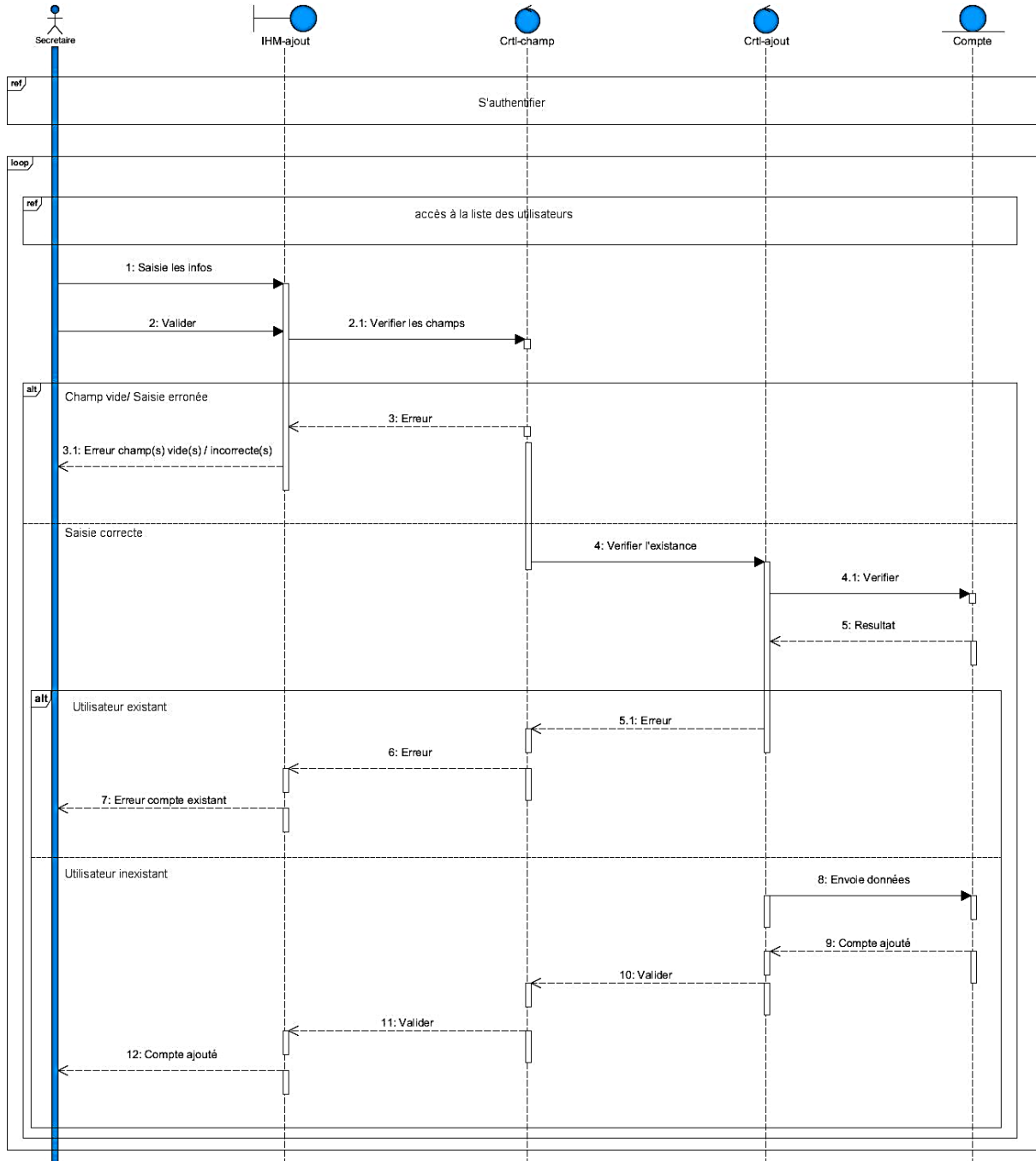


Figure III-2 : Diagramme d'interaction de cas d'utilisation « Ajouter un produit ».

III.2.1.3 Diagramme d'interaction de cas d'utilisation « Modifier un produit »

Dans la gestion des produit, le secrétaire peut modifier un produit déjà existant dans la liste des produits.

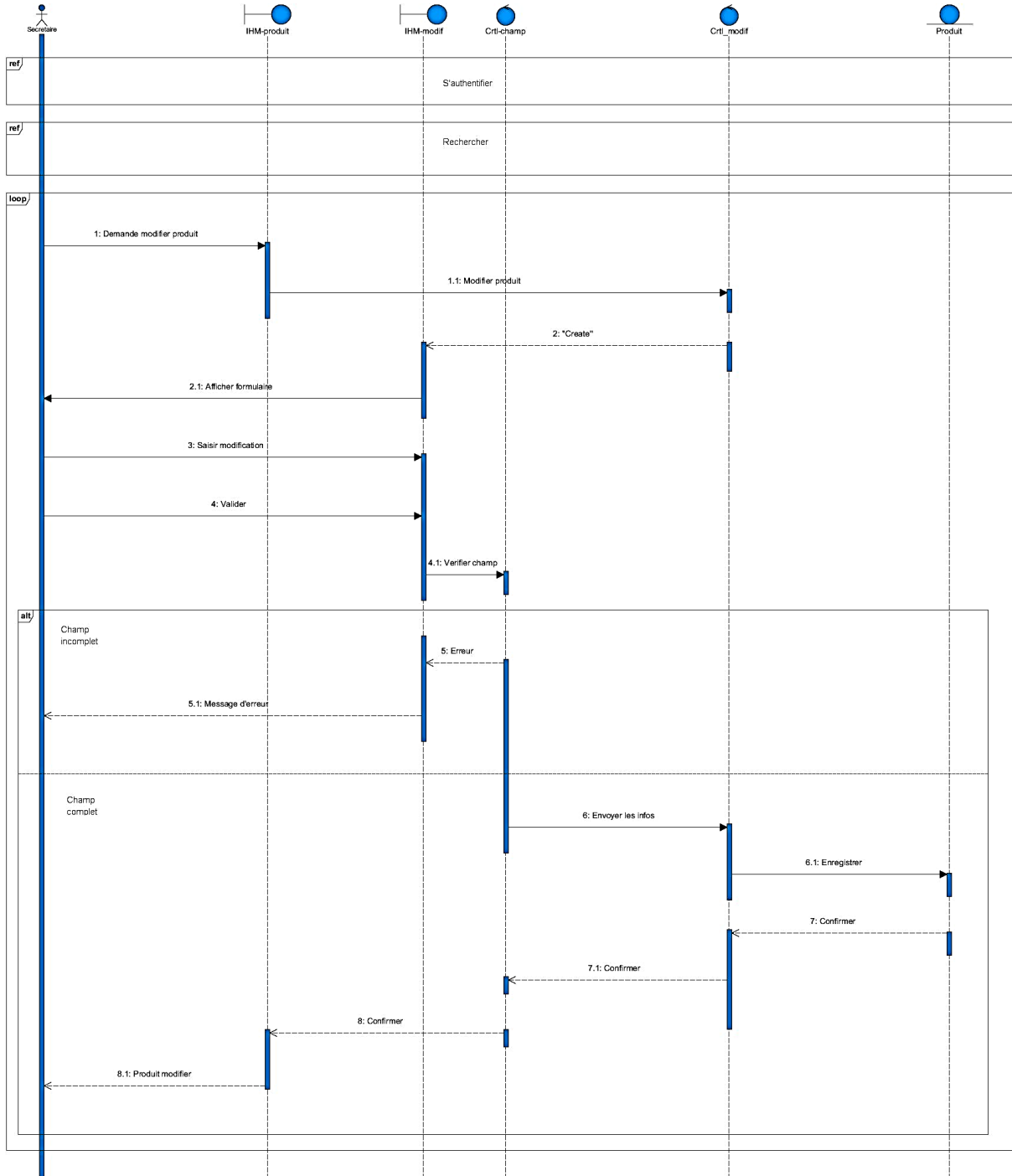


Figure III-3 : Diagramme d'interaction de cas d'utilisation « Modifier un produit ».

III.2.1.4 Diagramme d'interaction de cas d'utilisation « Supprimer un produit »

Comme dans le cas d'utilisation précédent, le secrétaire peut également supprimer carrément un produit existant.

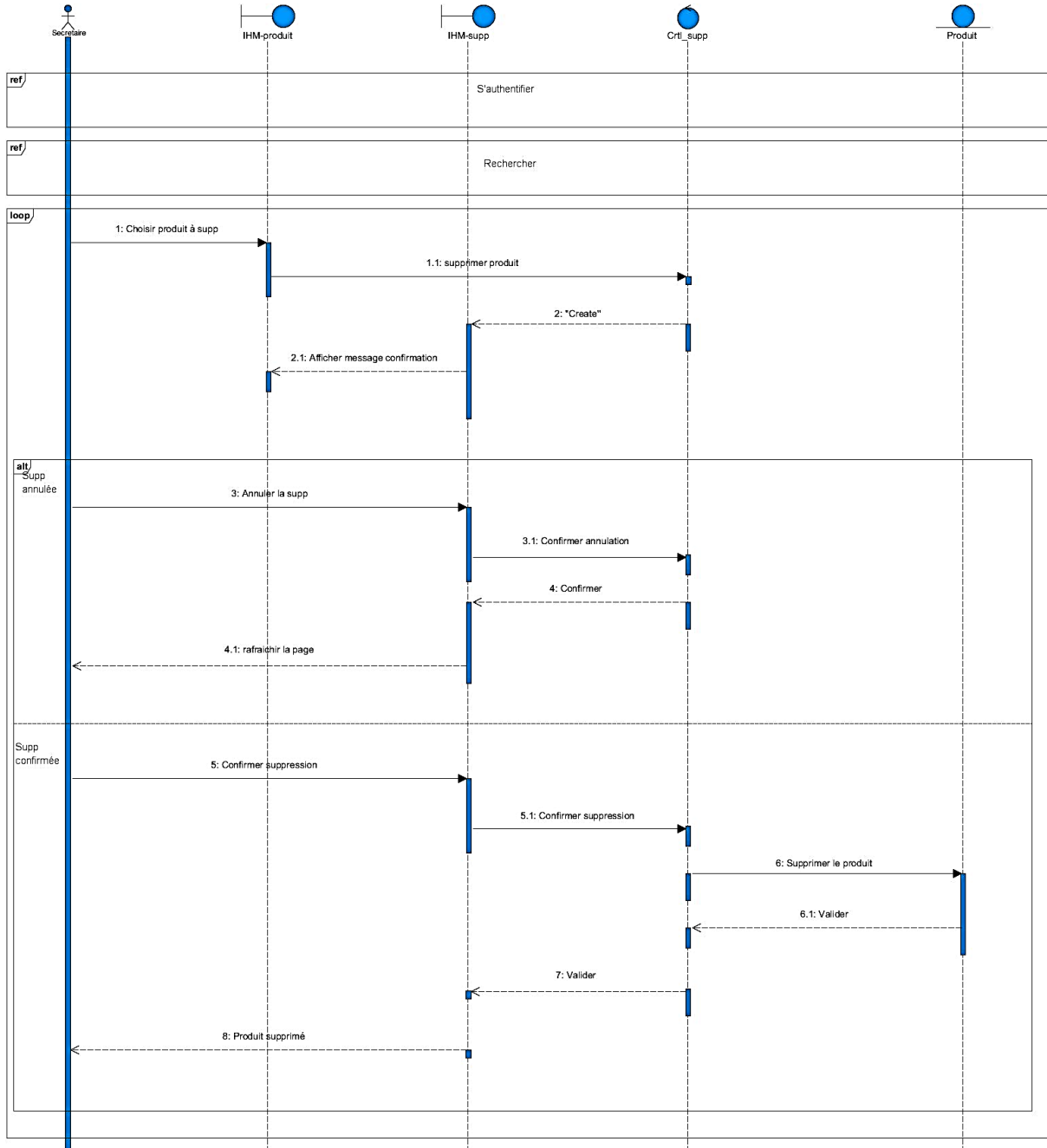


Figure III-4 : Diagramme d'interaction de cas d'utilisation « Supprimer un produit ».

III.2.1.5 Diagramme d'interaction de cas d'utilisation « Rechercher un produit »

En dernier on a pour le secrétaire la fonctionnalité de rechercher un produit dans les produits enregistrés.

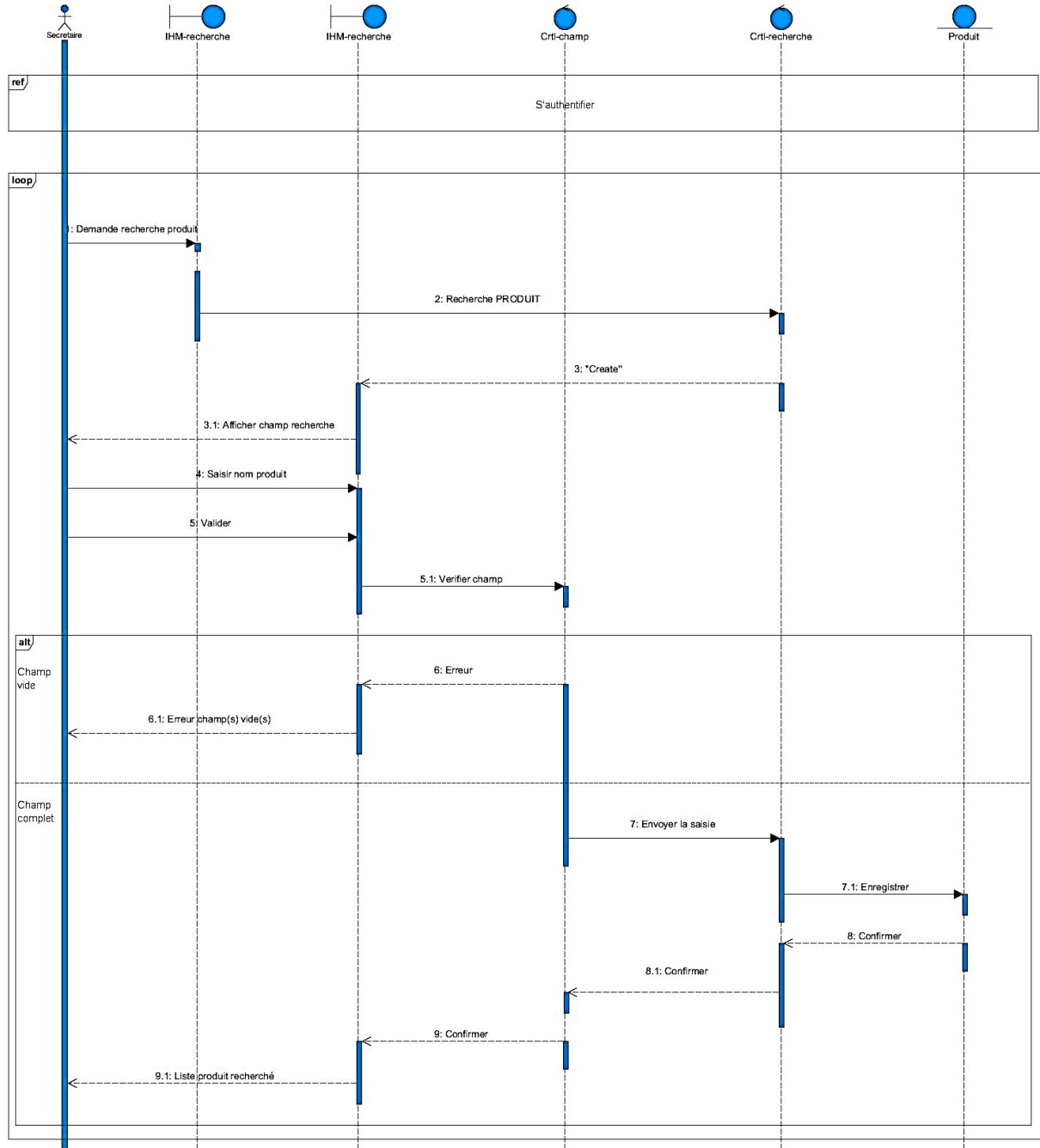


Figure III-5 : Diagramme d'interaction de cas d'utilisation « Rechercher un produit ».

III.3 Diagramme de classe

Le diagramme de classe représente les classes intervenant dans le système. Il décrit clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. La classe est un concept abstrait qui permet de représenter toutes les entités d'un système. Une classe peut donc représenter une voiture, un bouton cliquable, un devis, un utilisateur connecté, une structure de donnée ou tout autre élément devant être modélisé et donnant généralement lieu à la génération d'un code informatique. La classe est définie par son nom, ses attributs et ses opérations [9]

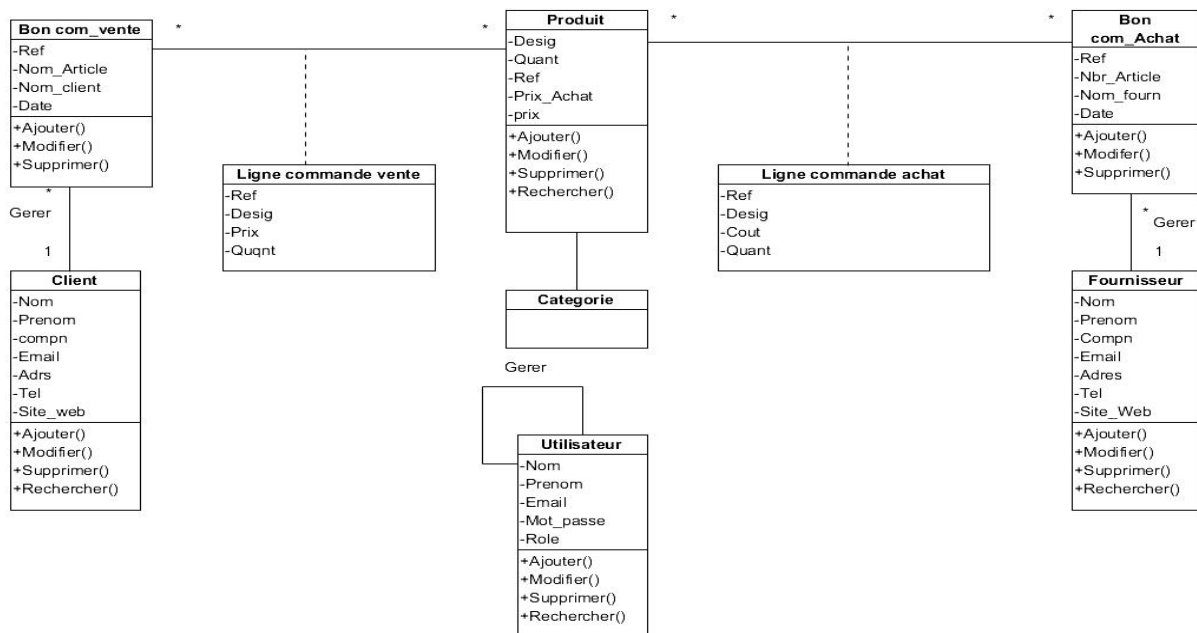


Figure III-6 : Diagramme de classe

III.4 Modèle orienté document

Concernant la structure des données de l'application nous avons choisi de dévier du modèle relationnel, et de nous intéresser au NoSQL, c'est à dire les bases de données dites non relationnelles et plus précisément au modèle orienté document.

III.5 Modèle relationnel et modèle orienté document

Les bases de données SQL sont connues sous le nom de bases de données relationnelles, et ont une structure de données basée sur des tableaux, avec un schéma strict et prédéfini requis. Les bases de données NoSQL, ou bases de données non relationnelles, peuvent être basées sur des documents, des bases de données graphiques, des paires clé valeur ou des magasins à colonnes larges. Les bases de données NoSQL ne nécessitent aucun schéma prédéfini, ce qui

vous permet de travailler plus librement avec des "données non structurées". Les bases de données relationnelles évoluent verticalement, mais généralement plus coûteuses, alors que la nature évolutive horizontale des bases de données NoSQL est plus rentable [10].

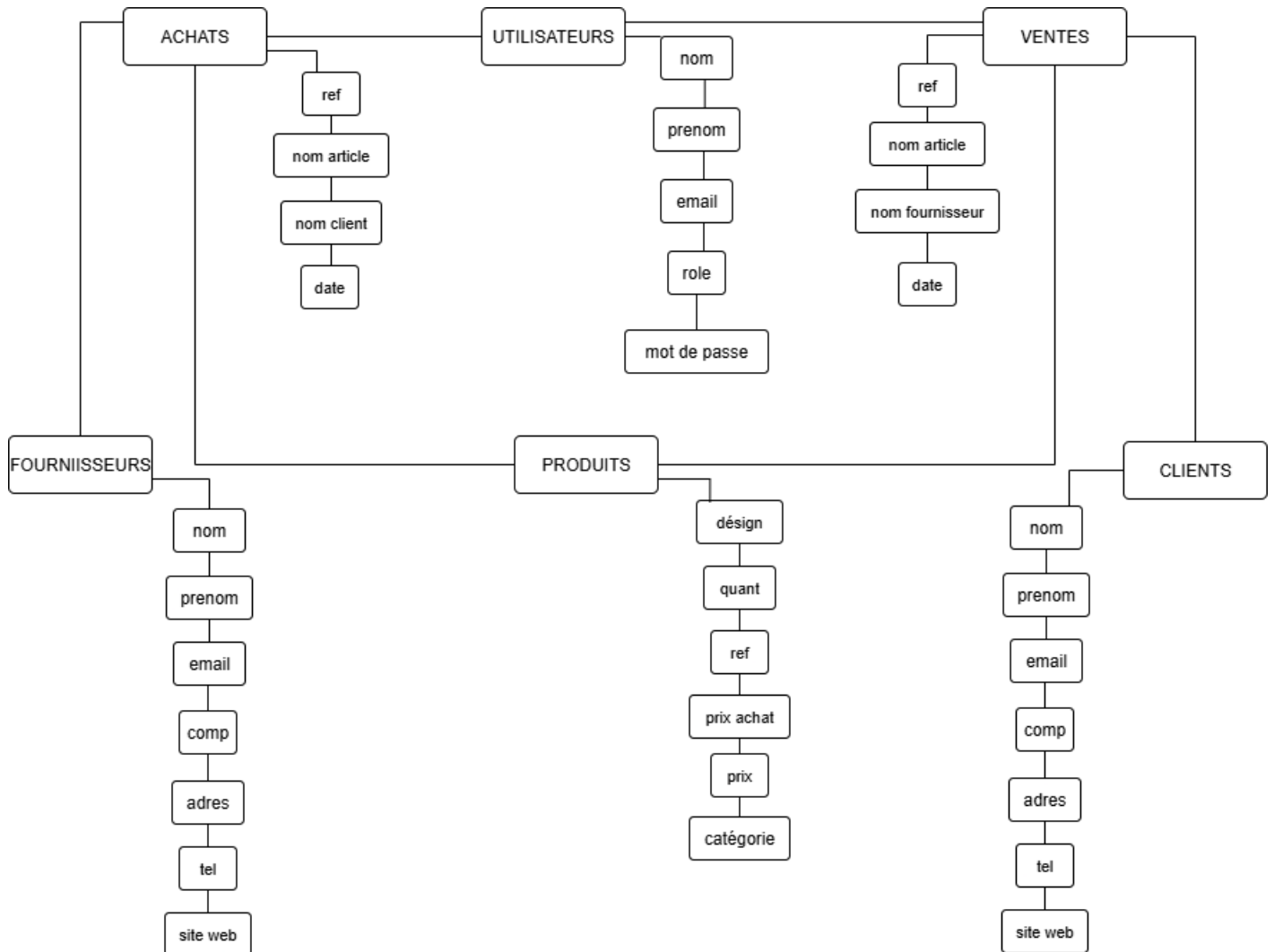


Figure III-7 : Schema de la base donnees NoSql

III.6 Conclusion

Au cours de ce chapitre nous avons en premier lieu modélisé les diagrammes de séquences qui nous ont donné une vue approfondie sur les interactions entre les objets et cela nous a permis de réaliser le diagramme de classe. Nous avons ensuite présenté les modèles orientés documents pour avoir une vue globale de notre base de données. Dans le prochain chapitre nous allons passer à la réalisation de notre application.

Chapitre IV

Réalisation

IV.1 Introduction

Ce chapitre est consacré pour la partie pratique de réalisation de notre application web, dans le but d'expliquer l'environnement et outils utilisés et ainsi la présentation de notre application à travers quelques interfaces homme-machine.

IV.2 Environnement et outils de développement

IV.2.1 Visual Paradigm

Visual Paradigm est une application logicielle conçue pour les équipes de développement de logiciels pour modéliser les systèmes d'information d'entreprise et gérer les processus de développement. En plus de la prise en charge de la modélisation, cette technologie fournit des capacités de génération de rapports et d'ingénierie de code, y compris la génération de code. Cette technologie peut inverser l'ingénierie des diagrammes à partir du code et fournir une ingénierie aller-retour pour divers langages de programmation. Visual Paradigm propose des diagrammes UML (Unified Modeling Language), des utilitaires Entity Relationship Diagram (ERD) et ORMD (Object Relational Mapping Diagrams) essentiels à la conception de systèmes et de bases de données [11][12].

IV.2.2 Visual Studio Code

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git [13].

IV.2.3 MERN stack

La MERN stack (MongoDB, Express, React, Node) représente l'alliance des technologies les plus puissantes que l'on trouve sur le marché. MERN est l'une des nombreuses variantes de la pile MEAN (MongoDB, Express, Angular, Node), toutes basées sur JavaScript, où le Framework front-tal Angular traditionnel est remplacé par React. Il s'agit d'un cadre de développement open source full stack, c'est-à-dire, qu'il fournit des composants de développement front-end et back-end. Cette combinaison va permettre aux développeurs de créer des applications web complètes (back-end et front-end) où on utilise le JavaScript côté client et le Node.js côté [14] [23].

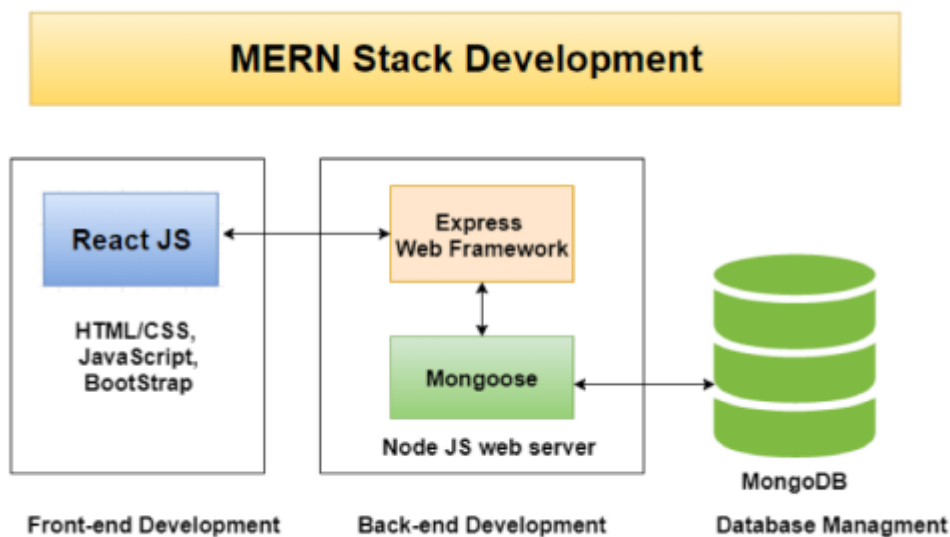


Figure IV-1 : Schéma illustrant la MERN stack

IV.2.4 Node Package Manager

Aujourd'hui, toutes les bibliothèques et frameworks JavaScript s'installent avec NPM (Node Package Manager). Un package manager est un gestionnaire de dépendances, il permet d'installer toutes les bibliothèques, frameworks et outils dont a besoin un projet JavaScript. C'est l'équivalent de composer en PHP, Maven en Java, et NuGet en C#. L'outil s'appelle Node Package Manager car il a été inventé au départ pour Node.js. Aujourd'hui, NPM est devenu le package manager pour tout l'écosystème JavaScript, y compris hors Node.js. Il suffit donc d'installer Node pour installer NPM [15].

IV.2.5 Front-end

IV.2.5.1 React.js

React JS est une bibliothèque basée sur JavaScript développée par Facebook depuis 2013, qui comprend la vitesse de JavaScript et le mécanisme innovant permettant de rendre les sites web extrêmement rapides et réactifs à l'égard des requêtes des utilisateurs. En effet, cette solution a radicalement transformé la méthodologie de Facebook en développement web/mobile. Depuis que React est apparu comme framework open source, deux ans après sa première version, sa diffusion parmi les clients était incroyablement dynamique grâce à son extraordinaire mode de codage de l'interface utilisateur [16].

IV.2.5.2 Tailwind CSS

Tailwind CSS est essentiellement un framework CSS utilitaire pour créer rapidement des interfaces utilisateur personnalisées. Il s'agit d'un cadre CSS personnalisable qui vous donne tous les éléments de base dont vous avez besoin pour créer des conceptions sur mesure [17].

IV.2.5.3 Formik

Formik est une bibliothèque de formulaires React. Il fournit une programmation et une validation de formulaire de base. Il est basé sur des composants contrôlés et réduit considérablement le temps de programmation des formulaires [18].

IV.2.5.4 Axios

Axios est une bibliothèque JavaScript fonctionnant comme un client HTTP. Elle permet de communiquer avec des API en utilisant des requêtes. Comme avec les autres clients HTTP, il est possible de créer des requêtes avec la méthode POST [19].

IV.2.5.5 Vite.JS

Vite approche le problème du développement front-end d'une nouvelle manière. Plutôt que de générer un "bundle" à chaque modification d'un fichier, Vite se repose sur le support des imports natif en créant un serveur personnalisé (basé sur koa) qui va délivrer des fichiers JS compatibles [20].

IV.2.5.6 React-pdf

React-pdf exporte un ensemble de primitives React qui vous permettent de restituer très facilement des éléments dans votre document. Il dispose également d'une API pour les styliser, en utilisant les propriétés CSS et la disposition Flexbox [21].

IV.2.6 Back-end

IV.2.6.1 Architecture MVC

L'architecture MVC (Modèle – Vue – Contrôleur) est un patron utilisé en PHP pour le développement d'applications web. Cette architecture a pour but de séparer les problématiques liées aux différents composants au sein de leur architecture respective. MVC permet de concevoir des applications web de manière claire et efficace grâce à la séparation des intentions. Les opérations de maintenance et de mises à jour sont fortement simplifiées [22] :

- Le modèle permet d'accéder aux données du site web que ce soit dans une base de données à l'aide de classes ou dans des fichiers plats de type XML ;
- La vue traite l'affichage à l'écran dans le navigateur web bien souvent retranscrit grâce à l'HTML et le CSS ;
- Le contrôleur gère les interactions. D'une manière générale, il utilise les données du modèle, les traite en fonction de l'action de l'utilisateur, et les envoie à la vue afin qu'elle les affiche.

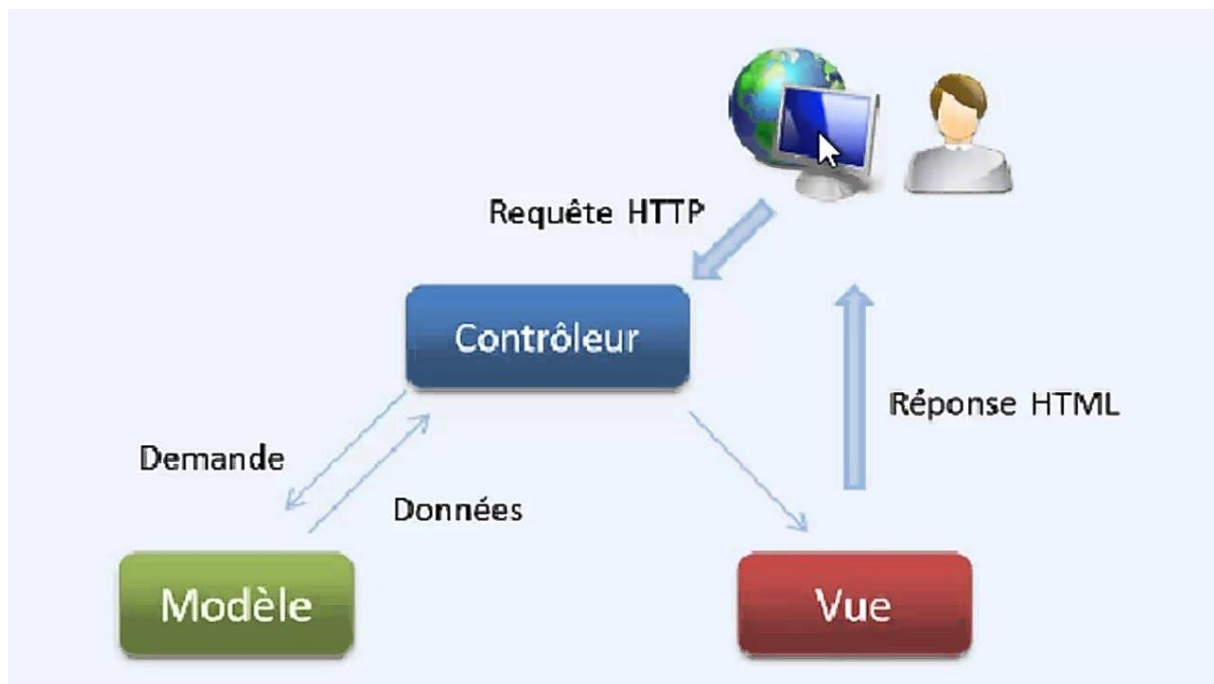


Figure IV-2 : Architecture MVC

IV.2.6.2 NodeJS

Node.js n'est pas un langage de programmation. Ce n'est pas non plus un framework JavaScript. Il s'agit d'un environnement d'exécution JavaScript. C'est une plateforme logicielle avec une architecture orientée événements qui permettent d'utiliser le langage de script JavaScript [24].

IV.2.6.3 Express JS

ExpressJS est une infrastructure d'applications web Node.js minimale et flexible, fournissant un ensemble robuste de fonctionnalités permettant de développer des applications web et mobiles. Il facilite le développement rapide d'applications web basées sur des nœuds [24].

IV.2.6.4 Passport JS

Passport est un middleware d'authentification pour Node.js. Passport est très flexible et modulaire, et peut être soigneusement inséré dans des applications Web basées sur Express. Un ensemble complet de politiques prend en charge l'authentification par nom d'utilisateur et mot de passe [25].

IV.2.6.5 JOI

Joi est Le langage de description de schéma et le validateur de données le plus puissant pour JavaScript. Joi est un Framework centré sur la configuration simple à utiliser avec un support intégré pour la validation des entrées, la mise en cache, l'authentification et d'autres fonctionnalités essentielles pour la création d'applications Web et de services [26].

IV.2.6.6 Dockerfile

Dockerfile est un document texte qui contient toutes les commandes qu'un utilisateur peut appeler sur la ligne de commande pour assembler une image. Les utilisateurs docker build peuvent créer une construction automatisée qui exécute successivement plusieurs instructions de ligne de commande [27].

IV.2.7 Implémentation de la base de données

IV.2.7.1 NoSQL

L'acronyme NoSQL (Not Only SQL) fait référence à des types de bases de données non relationnelles, qui stockent les données dans un format différent des tables relationnelles.

Cependant, les bases de données NoSQL peuvent être interrogées à l'aide d'API de langage idiomatique et de langages de requête structurés déclaratifs [28].

IV.2.7.2 MongoDB

Il s'agit d'une base de données open source centrée sur les documents. En seulement deux ans MongoDB est devenue l'une des bases de données NoSQL les plus populaires [24].

IV.2.7.3 Mongoose

Mongoose dans MongoDB est classée comme une bibliothèque ODM (Object Data Modelling) pour MongoDB et Node.js. Elle aide à gérer les relations entre les données et permet également la validation des schémas. Mongoose est également utilisée pour faire la traduction entre les objets qui sont dans le code et la représentation de ces objets dans MongoDB [29].

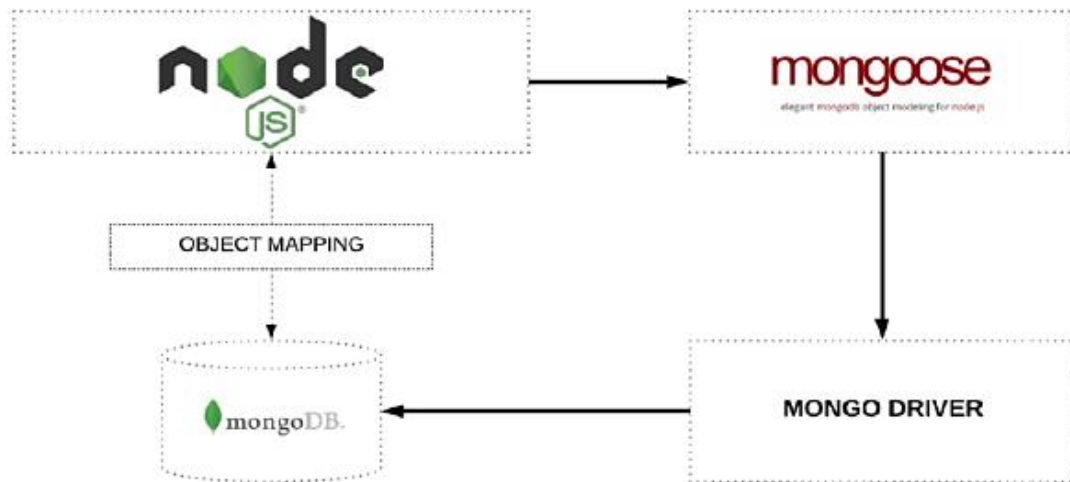


Figure IV-3 Mappage d'objets entre Node et MongoDB géré via Mongoose

IV.3 Principales interfaces

La Figure IV-4 représente l'interface de login pour tous les utilisateurs.

The screenshot shows a login form titled "Connectez vous". Below the title, it says "Connectez vous à votre espace administrateur". There are two input fields: "Email" with the value "admin@admin.fr" and "Mot de passe" with masked characters ".....". A blue button labeled "SE CONNECTER" is at the bottom.

Figure IV-4 : Interface login

La Figure IV-5 représente l'interface liste des produits, où le secrétaire pourra ajouter, supprimer ou modifier ses produits.

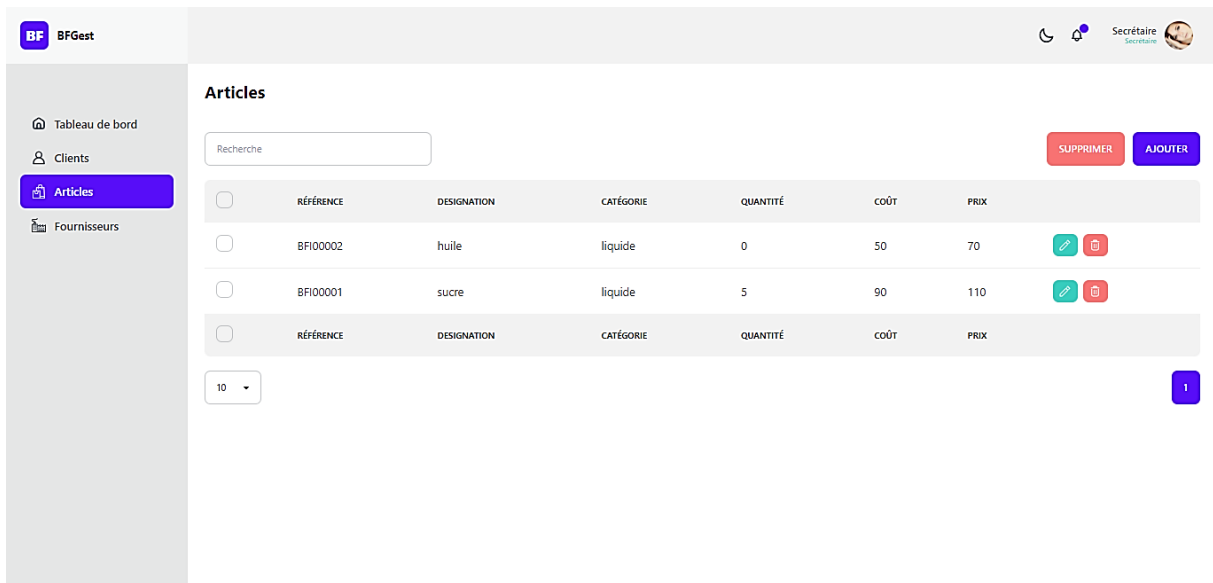


Figure IV-5 : Interfaces articles

La Figure IV-6 représente l'interface des achats, où le chargé approvisionnement pourra modifier, ajouter, supprimer un achat, générer des devis et factures.



Figure IV-6 : Interface achats

La Figure IV-7 représente l'interface ventes, où se trouve les commandes des clients, le chargé clientèle gère les commandes des clients (modifier, ajouter, supprimer) et générer des devis et factures.

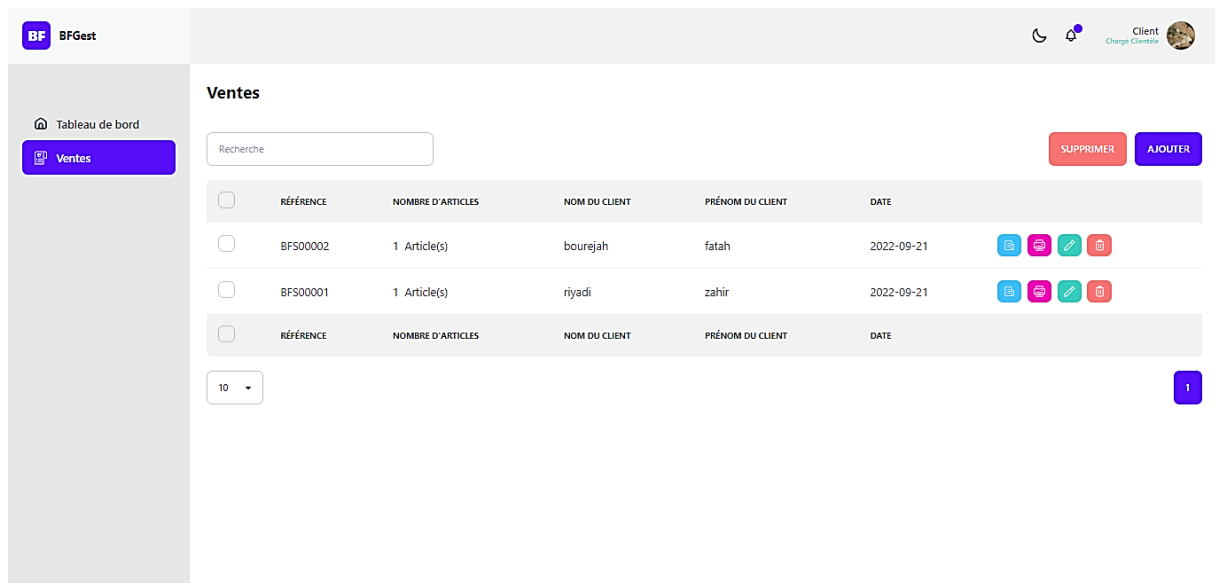


Figure IV-7 : Interface ventes

La Figure IV-8 représente l'interface utilisateurs, où se trouve les informations des utilisateurs, l'administrateur qui gère les rôles des utilisateurs (modifier, ajouter, supprimer).

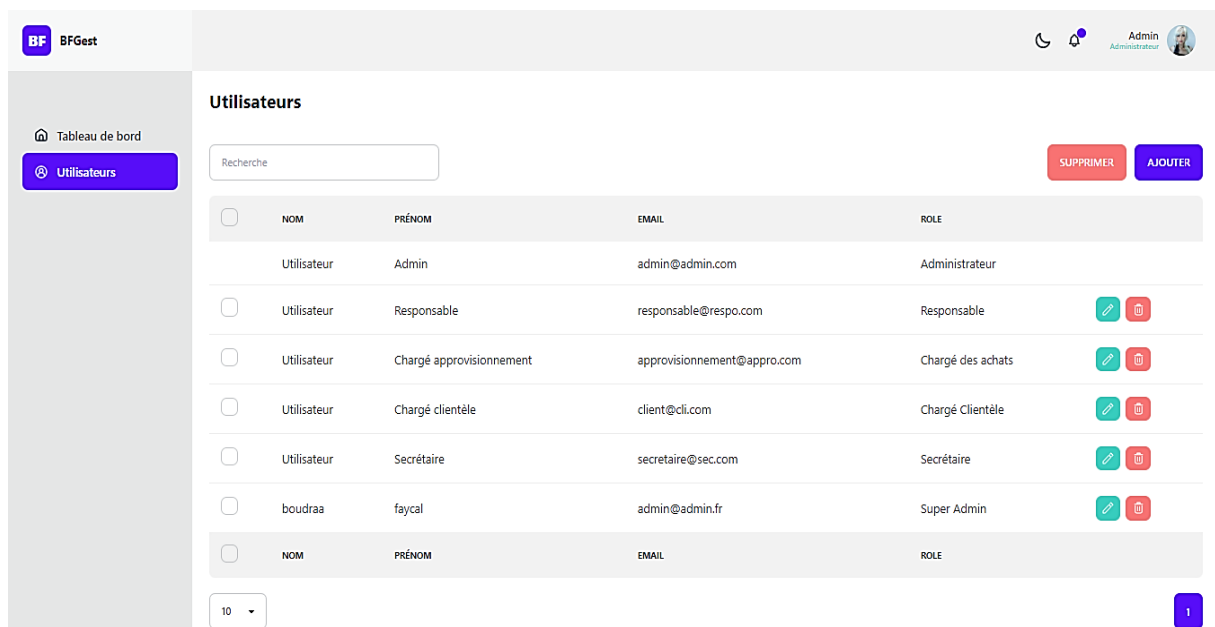


Figure IV-8 : Interface utilisateurs

La Figure IV-9 représente le mode sombre, en cliquant sur ce bouton nous changeons soit en mode normal ou sombre.

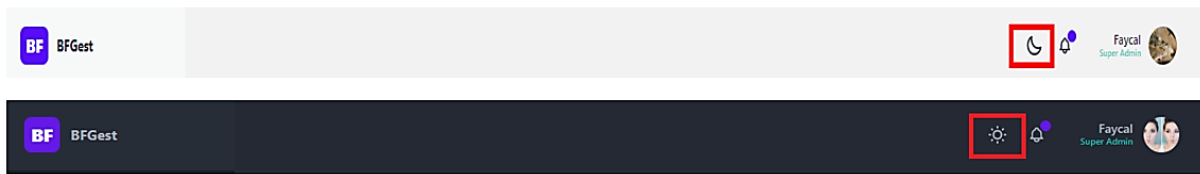


Figure IV-9 : Mode sombre

IV.4 Conclusion

Ce chapitre est dédié à la conception et réalisation de notre application, à savoir notre environnement et outils de développement qui nous ont permis de réaliser cette application, nous avons présenté les principales interfaces qui permettent de montrer l'importance et l'efficacité de notre application.

La gestion commerciale repose sur l'ensemble des tâches liées à l'activité commerciale d'une entreprise l'usage des nouvelles technologies de l'information et de la communication permet d'augmenter de manière rapide le chiffre d'affaires de toute entreprise.

Nous avons traité dans ce projet le problème de gestion commerciale des entreprises, nous avons identifié les problèmes et difficultés rencontrées par les entreprises lors leur gestion, puis proposé une solution qui est de réaliser un système sous forme d'application web, avec différents rôles et profils utilisateurs à savoir le responsable, la secrétaire, le chargé clientèle, le chargé approvisionnement et l'administrateur.

En guise de perspectives, les possibilités d'évolutions de ce projet sont nombreuses comme développer une version mobile permettant aux différents utilisateurs de se connecter même en dehors de l'environnement de travail et aussi inclure les finances et comptabilités, le service après-vente tout en mettant l'accent sur l'aspect coopératif de ces structures.

Références

Bibliographique

- [1] "C'est quoi l'entreprise ?". Source INSEE, SUSE 2004 (www.medef.fr).
- [2] <http://creasoi.fr/monde-lentreprise-ses-metiers-fonctionnement/>
- [3] Application Web-Définition et Explications. <https://www.technoscience.net/glossairedefinition/Application-Web.html>
- [4] Qu'est-ce qu'une application Web ? Aperçu des formats. <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/quest-ce-quuneapplication-web-aperçu-des-formats/>
- [5] Unified Process - an overview. <https://www.sciencedirect.com/topics/computerscience/unified-process>
- [6] Qu'est-ce que le langage UML. <https://www.lucidchart.com/pages/fr/langage-uml>
- [7] Roy GILLES. 'UML2 en action, de l'analyse des besoins en action'. Presses de l'Université de Québec, 2009, première édition.
- [8] ZOURANE Akil, MAMMERI Lydia, 'Conception et Réalisation d'une Application Réseau pour le Système ANDROID', Université de Béjaia ,2013
- [9] Diagramme de classe - UML SysML. <https://www.uml-sysml.org/diagrammes-uml-etsysml/diagramme-uml/diagramme-de-classe/>
- [10] NoSQL vs SQL Databases. <https://www.mongodb.com/nosql-explained/nosql-vs-sql>
- [11] visual paradigm est une application pour modéliser. Récupéré sur <https://www.oit.va.gov/Services/TRM/ToolPage.aspx?tid=10208>
- [12] Site pour essayer visual paradigm. Récupéré sur <https://www.visual-paradigm.com/features/>
- [13] Visual Studio Code. Récupéré sur <https://framalibre.org/content/visual-studio-code/>
- [14] Comprendre la MERN stack. Le blog du codeur. 2019. <https://leblogducodeur.fr/mern-stack/>
- [15] Node Package Manager (NPM). Formations JavaScript. 2021. (s.d.). Récupéré sur <https://formationjavascript.com/guides/npm-node-package-manager/>

- [16] React – Une bibliothèque JavaScript pour créer des interfaces utilisateurs. <https://fr.reactjs.org/>
- [17] Tailwind - un framework css pour créer des interfaces. Récupéré sur <https://tailwindcss.com/docs/installation>
- [18] Formik une bibliothèque de formulaires React. Récupéré sur <https://formik.org/docs/overview>
- [19] La Rédaction. Axios : concevoir une requête Post pour renvoyer les données d'un formulaire. JDN. 2020. <https://www.journaldunet.fr/web-tech/developpement/1441159-axios-concevoir-une-requete-post-pour-renvoyerles-donnees-d-un-formulaire/>
- [20] Vite- outils de construction front-end. Récupéré sur <https://vitejs.dev/guide/why.html>
- [21] React pdf récupéré sur <https://react-pdf.org/>
- [22] Développement de sites web avec l'architecture MVC. Blog Webcky. 2016. Récupéré sur <https://www.webcky.fr/blog/2016/09/16/developpement-site-web-avec-larchitecture-mvc-en-isere/>
- [23] MongoDB. What Is The MERN Stack ? Introduction & Examples. Récupéré sur <https://www.mongodb.com/mern-stack>
- [24] A. Marciano. Votre première application en React, Node, Express & MongoDB. Medium. 2022. Récupéré sur <https://axel-marciano.medium.com/votre-premi%C3%A8re-application-en-react-node-express-mongodb-5ab0dc531091>
- [25] Passport est un middleware d'authentification pour Node.js. Récupéré sur <https://www.passportjs.org/>
- [26] Joi est Le langage de description de schéma et le validateur de données. Récupéré sur <https://joi.dev/api/?v=17.6.0>
- [27] Docker file est un outil de déploiement récupéré sur <https://docs.docker.com/engine/reference/builder/>
- [28] Qu'est-ce que NoSQL ? Oracle. Récupéré sur <https://www.oracle.com/dz/database/nosql/>
- UML, Qu'est-ce que le langage. UML. Récupéré sur <https://www.lucidchart.com/pages/fr/langage-uml>
- [29] Introduction to Mongoose for MongoDB. freeCodeCamp. 2019. Récupéré sur <https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>

Résumé

Ce mémoire a été rédigé en vue de l'obtention du diplôme de fin de cycle Master en Informatique, option Génie Logiciel. Nous avons travaillé sur la conception et la réalisation d'une application web de gestion commerciale dans une entreprise. En effet, la complexité croissante de la gestion commerciale pousse de manière naturelle à la mise en place de systèmes d'information capables d'aider les employés de l'entreprise dans leurs tâches quotidiennes. Afin d'atteindre cet objectif, il nous a été proposé de concevoir et réaliser une application web assurant la gestion des produits, clients, fournisseurs, achats et ventes tout en proposant diverses options et fonctionnalités comme la génération de devis et factures. Ceci permettra d'améliorer la collaboration entre les acteurs dans l'entreprise afin d'offrir un meilleur rendement et de meilleures prestations. Pour ce faire, nous avons choisi de modéliser notre système avec le formalisme UML, et afin de réaliser notre projet informatique, nous avons utilisé la technologie MERN stack qui présente l'alliance des technologies les plus puissantes que l'on trouve sur le marché.

Abstract

This dissertation was written in order to obtain the diploma of end of cycle Master in Computer Science, option Software Engineering. We worked on the design and implementation of a web application for commercial management in a company. Indeed, the increasing complexity of commercial management naturally leads to the implementation of information systems capable of helping the company's employees in their daily tasks. In order to achieve this objective, it was proposed to us to design and implement a web application that manages products, customers, suppliers, purchases and sales, while offering various options and functionalities such as the generation of estimates and invoices. This will improve the collaboration between the actors in the company in order to offer a better performance and better services. To do this, we have chosen to model our system with the UML formalism, and in order to carry out our IT project, we have used the MERN stack technology, which is a combination of the most powerful technologies available on the market.