



DÉPARTEMENT D'INFORMATIQUE

MEMOIRE DE PROJET DE FIN D'ETUDES

Présenté en vue de l'obtention du

**DIPLÔME DE LICENCE FONDAMENTALE EN SCIENCES DE
L'INFORMATIQUE – LMD**

Sujet

**Conception et développement d'une application
gestion commerciale des produits informatiques**

Réalisé par : KARIA KHOULOU

KARIA CHAIMA

Sous la direction de : TABBABI ABID

Soutenu publiquement le 29/06/2018 devant le jury

Président : OTHMENI MOUHAMED

Rapporteur : DHIBI NAZIHA

Année Universitaire 2017/2018

Dédicace

Avec tout respect et amour je dédie ce modeste travail :

A celui qui a été toujours près de moi, pour m'écouter et me soutenir.

Qui était mon super homme et mon support pour toute la vie.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et pour mon bien être. Ce travail exprimer le respect et l'amour que je vous porte

*.... **A mon cher père Ali.***

A celle qui n'a jamais épargné un effort pour m'aider et m'encourager.

Ma raison d'être, ma raison de vivre, la lanterne qui éclaire mon chemin et m'illumine de douceur et d'amour. Veuillez trouver en ce travail la consolation et le témoin de la patience et d'amour.

*.... **A ma chère mère Bouthaina.***

En témoignage de la grande affection et de l'estime que j'ai pour vous.

Que dieu nous protège et nous garde toujours unis, heureux et sincères.

*.... **A mes chères Sœurs Khouloud, Rihab et Raoua***

Je vous dédie cette thèse en témoignage de gratitude d'estime et d'attachement. Puisse dieu vous accorder santé, longue vie et prospérité

*.... **A mes grands-parents Slimen et Brika***

*A celui que j'aime beaucoup et qui m'a soutenue tout au long de ce projet : **Mon fiancé Mouheb***

Mes oncles et leurs épouses.

Mes tantes et leurs maris.

Toutes mes cousines et tous mes cousins que j'aime bien.

Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite. Que Dieu vous protège.

Mon cousine Roua Elhani,

Vous êtes plus que de sœur.

*Veillez trouver, ici, l'expression de mon remerciement les plus sincères et de mon
profonde reconnaissance dont tu as fait montre tout au long de ce travail.*

A tous mes amis, *Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit
possible, je vous dis merci.*

Chaima

Dédicace

« Louange à Allah qu'il m'a guidé à ceci ». Je dédie ce travail

A ma chère mère Bouthaina.

Amon cher père Ali.

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.

Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez.

Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.

Ames chères princesses Chaima, Rihab et Raoua

Aucune dédicace ne saurait exprimer tout l'amour que j'ai pour

Vous, Votre joie et votre gaieté me comblent de bonheur.

Puisse Dieu vous garder, éclairer votre route et vous aider à réaliser votre tour vos vœux les plus chers.

A mes grands-parents Slimen et Brika

Je vous remercie pour votre soutien et vos encouragements me donnent

La force de continuer.

A tous les membres de ma famille, petits et grands

Veillez trouver dans ce modeste travail l'expression de mon affection.

Mon cousine Roua Elhani,

Vous êtes plus que de sœur.

Veillez trouver, ici, l'expression de mon remerciement les plus sincères et de mon profonde reconnaissance dont tu as fait montre tout au long de ce travail.

À tous mes amis

En Souvenir des plus beaux instants qu'on a passés ensemble

Aussi bien à tous ceux qui m'ont aidé

Et je remercie tous ceux qui, de près ou de loin, m'ont aidé à élaborer ce rapport.

Khouloud



Remerciement

C'est un grand plaisir que nous réservons ces lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué à l'élaboration de ce travail.

*Au terme de ce travail je tenais à exprimer toutes mes reconnaissances et remerciements à mon encadrant « **Mr. Tabbabi Abid** » pour son conseil, son orientation ainsi que leur soutien moral et scientifique, j'ai permis de mener à terminer ce projet.*

*Je tenais également à remercier mon encadrant de société « **IB EXTRA SERVICES** » pour l'opportunité qu'il m'a offerte et la patience dont il a fait preuve lors de la réalisation de mon travail.*

Nous exprimons également nos gratitude à l'égard des membres du jury d'avoir accepté jugé ce modeste travail.

Mes vifs remerciements vont aussi à mes enseignants, qui ont fait de leurs mieux pour me donner une formation universitaire



Liste des matières

Introduction générale.....	1
Chapitre I: Présentation du projet	3
I. Etude fonctionnelle :.....	3
I.1. Présentation du projet :	3
I.2. Présentation de l'organisme d'accueil (IB EXTRA SERVICES):	3
I.3. Problématique :	4
I.4. Etude de l'existant:.....	4
I.5. Objectifs à atteindre :	5
I.6. Solution proposée :.....	5
II. Gestion commercial :	6
II.1. Définition de la gestion de commercial :.....	6
II.2. Objectifs de la gestion de commercial :	6
II.3. Différentes tâches de la gestion commerciale :	6
II.4. Flux générés par l'activité commerciale :	7
III. Gestion du projet :	8
III.1. Cycle de vie d'un logiciel :.....	8
III.2. Outils de modélisation :.....	9
III.3. Langage de modélisation :.....	10
Chapitre II: Analyse et spécification des besoins	14
Introduction.....	14
I. Analyse des besoins :.....	14
I.1. Les besoins fonctionnels :	14
I.2. Les Besoins non fonctionnelles :.....	18
II. Analyse et spécificationdu projet:.....	18
II.1. Identification des acteurs :.....	18
II.2. Description textuelles des cas d'utilisations	19
II.3. Diagramme de cas d'utilisation :	20
II.4. Relations entre les cas d'utilisation :	21
II.5. Cas d'utilisations spécifiques :	23
Conclusion :	26
Chapitre III: Conception de l'application	28
Introduction.....	28
I. Description des scénarios, diagrammes de séquence et diagramme d'activité :	28
II. Développement du modèle statique :	41
II.1. Le dictionnaire de données :.....	41
II.2. Représentation des classes :	45
II.3. Diagramme de classes :	52
III. La base de données :	54
III.1. Modèle logique des données :	54

III.2. Modèle physique de données	55
Chapitre IV: Réalisation de l'application.....	54
I. Environnement de travail :.....	58
I.1. Environnement matériel :.....	58
I.2. Environnement logiciel :.....	58
II. Choix langages et techniques :.....	61
III. Présentation des interfaces de l'application :.....	61
III.1. Page d'authentification :.....	61
III.2. Interface d'accueil :.....	62
III.3. Partie back office :.....	63
III.4. Partie front office :	64
Conclusion générale	75
Bibliographie	76
Webographie	76

Liste des figures

Figure 1 : Flux entre le client et la fonction commerciale [2].....	8
Figure 2 : Modèle du cycle de vie en cascade [4]	10
Figure 3: Logo d'UML.....	11
Figure 4:La partie Front office	15
Figure 5: Fonctionnalités du Back office	17
Figure 6: Diagramme de cas d'utilisation global	20
Figure 7: Relation d'association.....	21
Figure 8:Relation d'inclusion	22
Figure 9:Relation de généralisation.....	22
Figure 10: Diagramme de cas d'utilisation « Modules gestion des ventes ».....	23
Figure 11: Diagramme de cas d'utilisation « Modules gestion des stocks »	24
Figure 12: Diagramme de cas d'utilisation « Modules gestion des achats »	25
Figure 13: Diagramme de cas d'utilisation « Modules gestion d'administrateur »	26
Figure 14: Diagramme de séquence de cas d'utilisation « S'authentifier »	30
Figure 15: Diagramme de séquence « Ajouter Produits »	33
Figure 16:Diagramme d'activité « Ajouter Produits »	33
Figure 17: Diagramme de séquence « Supprimer Produit »	35
Figure 18: Diagramme d'activité « Supprimer produit »	35
Figure 19: Diagramme de séquence « Modifier Produits »	37
Figure 20:Diagramme d'activité « Modifier produit ».....	37
Figure 21: Diagramme de séquence « Ajouter commande »	39
Figure 22:Diagramme d'activité «Ajouter commande».....	39
Figure 23: Diagramme de séquence « Ajouter facture ».....	40
Figure 24:Diagramme d'activité «Ajouter commande».....	41
Figure 25:Diagramme de classes.....	53
Figure 26: Microsoft Visual Studio.....	59
Figure 27: Visual C#	59
Figure 28: MySQL Workbench.....	60
Figure 29: StarUML	60
Figure 30: Page d'authentification.....	62
Figure 31: Interface d'accueils employés.....	62
Figure 32: Interface d'accueil Administrateur	63
Figure 33: interface gestion d'employés	64
Figure 34: interface gestion des groupes du travail.....	64
Figure 35: Interface gestion des Clients	65
Figure 36: Interface Supprimer client	66
Figure 37: Interface test ajouté de client	66
Figure 38:Interface gestion des Commandes clients.....	67
Figure 39:Interface imprimer la fiche facture client	68
Figure 40: document PDF "facture client"	69

Figure 41: Interface gestion des Produits	70
Figure 42: Interface gestion des Catégories	70
Figure 43: Interface gestion des fournisseurs	71
Figure 44: Interface Modifier fournisseur	72
Figure 45:Interface gestion des Commande fournisseur	73
Figure 46:Interface gestion des factures fournisseur.....	74

Liste des tableaux

Tableau 1: Les tâches principales de chaque acteur de notre système	19
Tableau 2: Cas d'utilisation « S'authentifier».....	29
Tableau 3:Cas d'utilisation «Ajouter produits»	32
Tableau 4:Cas d'utilisation «Supprimer Produits»	34
Tableau 5:cas d'utilisation « Modifier Produit»	36
Tableau 6:cas d'utilisation «Ajouter Commandes»	38
Tableau 7:cas d'utilisation « Ajouter Facture»	40
Tableau 8:dictionnaire de données	42
Tableau 9:Classe Client.....	45
Tableau 10:Commande Client.....	46
Tableau 11: Classe Ligne De Commande	46
Tableau 12:Classe Facture Client.....	47
Tableau 13 :Classe Fournisseurs	48
Tableau 14:Classe Commande Achat.....	49
Tableau 15 :Classe Ligne D'Achat	49
Tableau 16:Classe Facture Fournisseur.....	50
Tableau 17:Classe Produit.....	51
Tableau 18: Classe Catégories	52
Tableau 19:Configuration Matérielle	58

Liste des acronymes

UML= **U**nified **M**odeling **L**anguage

OMT = **O**bject **M**odeling **T**echnique

OOSE= **O**bject **O**riented **S**oftware **E**ngineering

CRUD = **C**reate **R**ead **U**pdate **D**eleter

COO=Conception **O**rientée **O**bjet

PDF= **P**ortable **D**ocument **F**ormat

MLD= **M**odèle **L**ogique des **D**onnées

SQL =**S**tructured **Q**uery **L**anguage

Introduction générale

De l'âge de la pierre à nos jours, l'esprit perfectionniste de l'homme n'a cessé de lui permettre d'améliorer sa vie quotidienne. Le passage de la mécanique aux domaines d'informatique, d'électronique, d'automatique et de domotique a révolutionné la vie journalière de l'être humain. Les nouvelles technologies de l'information et de communication illustrent ce phénomène.

Aujourd'hui, vu l'intérêt croissant de vouloir gagner en temps, de conserver les données, de limiter le nombre d'employés et pas mal d'autres raisons, ont poussé petites, moyennes et grandes entreprises à chercher des solutions informatiques capables de répondre à leurs besoins.

Dans cette perspective et vue l'importance de la gestion commerciale dans le fonctionnement des entreprises, se présente l'application du notre projet qui permet de gérer tous les éléments concernant les ventes, les achats et les stocks effectuées.

Dans ce cadre s'inscrit notre projet de fin d'études qui consiste à réaliser une application sur mesure de gestion commerciale pour une société de ventes des matériels informatiques.

Le présent rapport présentera les différentes étapes de notre travail qui sera structuré en trois chapitres :

Dans le premier chapitre intitulé «Présentation du projet», nous présentons la description générale de gestion commerciale en déterminant la méthodologie du travail ainsi que le langage de modélisation.

Dans le second chapitre intitulé «Analyse et spécification des besoins », nous identifierons les besoins fonctionnels et non fonctionnels ainsi que la description de la structure de différentes fonctions qui nous allons mettre en place.

Dans le troisième chapitre intitulé « Conception de l'application» est consacré à la partie conception qui consiste à faire l'élaboration des spécifications de l'architecture de notre application en se basant sur le langage UML.

Dans le dernier chapitre intitulé «Réalisation de l'application», nous présenterons l'environnement de développement et les imprimés écrans qui valide notre implémentation. On finit par une conclusion qui synthétise notre travail et présente les perspectives envisageables

Chapitre I: Présentation du projet

Introduction

Dans le présent chapitre, nous entamons la présentation de notre projet. Tout d'abord, nous présentons le cadre de notre travail. Nous enchaînons avec les notions de base de gestion de commercial et la définition des différentes tâches de gestion de commercial. Nous traçons ensuite les objectifs de notre projet. Nous terminons par la méthodologie du travail.

I. Etude fonctionnelle :

I.1. Présentation du projet :

Notre projet de fin d'études concerne la réalisation d'une application pour la société IB EXTRA SERVICES.

En fait, pour assurer la communication de leur équipe, la société a émis le désir de créer une application dans le but de vente des produits informatique et de se faire connaître et attirer des clients ainsi que gérer les différents données, A l'égard de cette difficulté qui provient principalement de leur gestion manuelle (gestion des données), il devient nécessaire de développer une application informatique permettant de résoudre ce problème.

Dans ce cadre s'inscrit notre projet de fin d'études visant la conception et la Développement d'une application qui assure la gestion commerciale des produits informatiques.

I.2. Présentation de l'organisme d'accueil (IB EXTRA SERVICES):

IB EXTRA SERVICES est une société des services de maintenance a été créé au 2 Novembre 2010 sous la registre commerciale ayant de capital 10.000.000 DT et a comme activité principale la maintenance, la réparation, l'assistance et le dépannage des équipements informatiques et des réseaux locaux et internet.

IB EXTRA SERVICES se situe à l'immeuble Chiraz Bureau N°6 ,2ème étage Avenue Farhat Hached, En face de la gare routière 2200 Tozeur.

❖ Les services assurés IB EXTRA SERVICES:

- Installation, configuration et soutien réseau
- Diagnostics, réparation, maintenance et assistance sur site et en atelier.
- Installation et configuration réseaux locaux et internet
- Installation et configuration de logiciels et de systèmes d'exploitation

- Récupération de données
- Suppression de virus, spywares et trojans
- Optimisation des performances du système
- Création d'une application
- Vente des produits informatique, bureautique, des pièces détachées, accessoires et consommables
- ❖ IB EXTRA SERVICES renferme:
 - 1 Gérant.
 - 2 ouvriers.

I.3. Problématique :

Dans le cadre de la gestion de cette société l'administrateur utilise une méthode classique et vu le nombre des employés et des groupes du travail, clients et fournisseurs, cette méthode ne fait qu'influencer négativement sur l'efficacité de la gestion. Pour faciliter et simplifier les tâches à l'administrateur, il nous demande de développer une application qui lui permettra une gestion plus rapide et plus efficace. De même, il n'y a pas une façon de se communiquer avec les clients sans arriver à la société.

I.4. Etude de l'existant:

I.4.1. Présentation et analyse de l'existant:

Jusqu'aujourd'hui, les tâches d'administration au sein de la société IB EXTRA SERVICES sont réalisées manuellement. Actuellement dans l'IB EXTRA SERVICES de Tozeur :

Il n'a aucun moyen qui répond aux besoins de membre public : Le besoin d'une telle information consiste à faire un déplacement au local de la société IB EXTRA SERVICES.

il n'y a pas assez de services offerts aux clients ou de moyen de communication à distance sauf que le téléphone et la page Facebook, c'est pour cela que les clients trouvent certaines difficultés pour avoir des informations concernant la société IB EXTRA SERVICES.

Toutes les données des clients et des fournisseurs de la société IB EXTRA SERVICES sont enregistrés sur des fiches papiers.

Les clients sont obligés à aller à l'IB EXTRA SERVICES pour acheter le produit.

Les fournisseurs sont obligés à aller à l'IB EXTRA SERVICES pour vendre le produit à la société.

I.4.2. Critique de l'existant :

La critique de l'existant doit mettre en évidence les activités fondamentales et les informations associées, ainsi que les dysfonctionnements du système actuel (page Facebook).

Jusqu'aujourd'hui, la gestion des ventes, du stocks et des achats est réalisée manuellement, la IB EXTRA SERVICES utilise une méthode classique au niveau du stockage des données (les informations personnels...). Ce qui implique la perte du temps et la difficulté de cette tâche pour l'administrateur surtout dans le cas où le nombre des clients et des employés.

La société IB EXTRA SERVICES ne possède pas une vitrine sur le web.

Il y a beaucoup d'insuffisance dans la situation actuelle de l'IB EXTRA SERVICES, la page Facebook ne sert pas certains besoins comme l'inscription des clients en ligne.

C'est ce qui l'oblige d'aller au local de l'IB EXTRA SERVICES.

L'IB EXTRA SERVICES ne possède qu'une page Facebook sur laquelle elle partage ses nouveautés, leurs actualités et leurs événements actuels, vous savez tous que la publication des informations personnelles des clients sur Facebook ne respecte pas la vie privée de ces clients. Cet outil est moins sécurisé.

I.5. Objectifs à atteindre :

Notre objectif consiste à réaliser une application qui permet de:

Automatiser le processus de gestion des employés

Automatiser le processus de gestion des clients, fournisseurs.

Améliorer et simplifier le travail de l'administrateur au sein de société.

I.6. Solution proposée :

Après l'étude que nous avons effectué, et selon les besoins de l'IB EXTRA SERVICES, nous obtenons pour la création d'une application qui possède :

- Un espace administrateur.
- Un espace client
- Un espace modules gestion des ventes
- Un espace modules gestion des stocks
- Un espace modules gestion des achats

II. Gestion commercial :

II.1. Définition de la gestion de commercial :

La gestion commerciale concerne toutes les opérations de gestion des achats et des ventes d'une entreprise. Elle s'appuie sur le traitement et la gestion des données clients, fournisseurs et produits de l'entreprise. Elle est un acteur primordial à la bonne santé de l'entreprise en garantissant ses résultats économiques. [1]

II.2. Objectifs de la gestion de commercial :

Le développement d'une entreprise passe avant tout par une gestion informatisée des ventes, des achats et des stocks. Le système de gestion informatisé va permettre :

- La gestion des tiers (prospects, des clients, fournisseurs, salariés)
- La gestion des documents (devis, bons de commandes, bons de livraisons, factures, avoirs financiers,...)
- La gestion des spécificités de l'activité de l'entreprise (négoce, fabrication, prestations de services...)
- La gestion des achats
- La gestion des livraisons
- La gestion du stock
- L'archivage des documents

La gestion commerciale informatisée regroupe toutes les phases de l'activité : de la prospection au service après-vente.

Outre la mise en place d'une organisation structurée, elle permet également de générer un gain de temps et d'éviter les erreurs dues aux éventuelles ressaisies. [1]

II.3. Différentes tâches de la gestion commerciale :

La Gestion commerciale est constituée essentiellement des différentes tâches suivantes :

- **Tâches opérationnelles de vente :** Elle consiste à gérer l'ensemble d'activités de commerciaux et assurer le suivi des commandes des clients tout en gérant les factures et les dossiers des clients. Les utilisateurs sont tenus d'utiliser le système dans la réalisation de leurs tâches qui sont accordées.

- **Tâches opérationnelles d'achat** : c'est l'enregistrement des factures des produits achetés dans la base de données à travers l'application. Des formulaires seront saisis pour collecter des informations sur les produits achetés.
- **Tâches opérationnelles articles** : elle concerne les produits vendus aux clients et les produits approvisionnés chez les fournisseurs
- **Tâches opérationnelles de stock** : ce cas permet le contrôle des produits entrants et sortants et la mise à jour du stock si par exemple un produit est livré il doit être supprimé automatiquement, comme il doit le mettre à jour si un produit est acheté.
- **Tâches opérationnelles des clients et des fournisseurs** : Les fournisseurs et les clients seront rentrés avec leurs coordonnées. Des fiches de contact seront prévues pour rentrer les personnes que l'on doit contacter chez le client ou le fournisseur.
- **Tâches opérationnelles des bons de commandes** : est l'enregistrement d'un document établi par le client dans lequel il détaille les articles commandés et toutes les conditions utiles à la bonne exécution de la commande. A partir de la commande on peut sortir la facture et le bon de livraison.
- **Tâches opérationnelles des fournisseurs** : permet de créer des dossiers des fournisseurs et de les mettre à jour une fois il y a une notification à ajouter et afficher les produits fournis par chaque fournisseur.
- **Tâches opérationnelles de la facturation** : ce cas assure la création et la mise à jour des factures, comme il permet de faire le suivi des factures.
- **Tâches opérationnelles de les produits** : comporte la création des produits ou bien les mettre à jour. Aussi, ce cas d'utilisation permet de visualiser la variation des stocks et de signaler les besoins vis-à-vis les produits matériels non disponibles. [1]

II.4. Flux générés par l'activité commerciale :

Le schéma ci-dessous est un exemple de liaison entre le client et l'activité commerciale d'une entreprise commerciale :

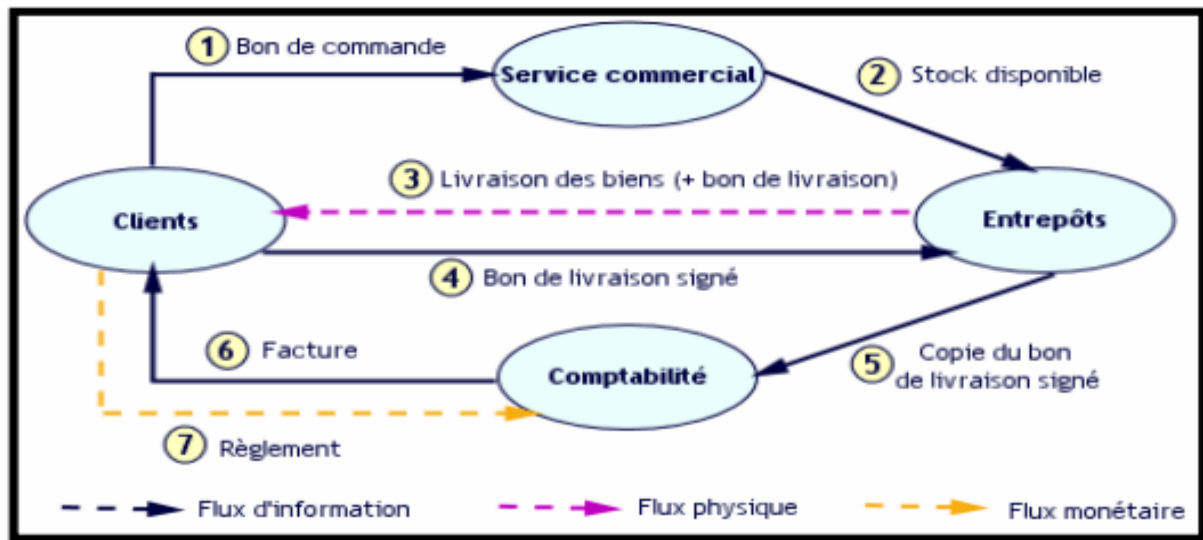


Figure 1 : Flux entre le client et la fonction commerciale [2]

III. Gestion du projet :

La gestion de projet (ou conduite de projet) est une démarche visant à organiser de bout en bout le bon déroulement d'un projet.

III.1. Cycle de vie d'un logiciel :

Le cycle de vie d'un logiciel (en anglais software life cycle), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre. [3]

L'origine de ce découpage provient du constat que les erreurs ont un coût d'autant plus élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés. [3]

Le cycle de vie du logiciel comprend généralement au minimum les étapes suivantes :

- **Définition des objectives :** Cette étape consiste à définir la finalité du projet et son inscription dans une stratégie globale.
- **Analyse des besoins et faisabilité :** C'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes, puis l'estimation de la faisabilité de ces besoins, Spécifications ou conception

générale : Il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.

- **Conception détaillée :** Cette étape consiste à définir précisément chaque sous-ensemble du logiciel.
- **Codage (Implémentation ou programmation):** C'est la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- **Tests unitaires :** Ils permettent de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications.
- **Intégration :** L'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de tests d'intégration consignés dans un document.
- **Qualification (ou recette) :** C'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
- **Documentation :** Elle vise à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- **Mise en production :** C'est le déploiement sur site du logiciel.
- **Maintenance :** Elle comprend toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

La séquence et la présence de chacune de ces activités dans le cycle de vie dépend du choix d'un modèle de cycle de vie entre le client et l'équipe de développement. Le cycle de vie permet de prendre en compte, en plus des aspects techniques, l'organisation et les aspects humains. [3]

Ils existent plusieurs modèles de cycles de vie d'un logiciel tels que : Modèle en cascade, en V, en spiral, par incrément, etc.

III.2. Outils de modélisation :

III.2.1. Modèle de cycle de vie en cascade :

Pour bien conduire notre projet et nous assurer du bon déroulement des différentes phases, nous avons opté cycle de vie en cascade comme une méthodologie de conception et de développement.

Le modèle de cycle de vie en cascade a été mis au point dès 1966, puis formalisé aux alentours de 1970.

Dans ce modèle le principe est très simple : chaque phase se termine à une date précise par la production de certains documents ou logiciels. Les résultats sont définis sur la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s'ils sont jugés satisfaisants. [3]

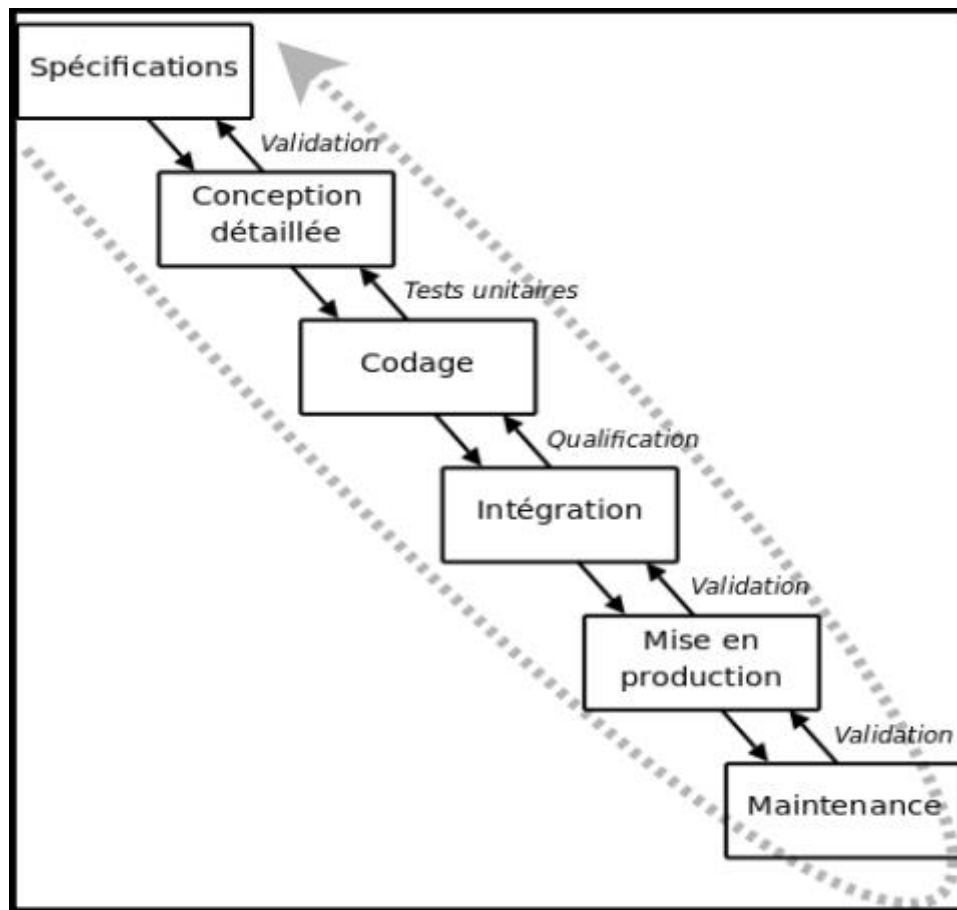


Figure 2 : Modèle du cycle de vie en cascade [4]

III.3. Langage de modélisation :

Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.[5]

L'UML est le résultat de la fusion de précédents langages/techniques de modélisation objet :

- **BOOCH** : Nom tiré de son inventeur (Cree par Grady Booch).
- **OMT** : **O**bject **M**odeling **T**echnique (Cree par James Rumbaugh).
- **OOSE**: **O**bject **O**riented **S**oftware **E**ngineering (Cree par Ivar Jacobson). [5]

⇒ Langage UML est donc la fusion de ces trois méthodes, tout en restant simple et homogène. Ce langage est devenu le standard en termes de modélisation objet.

En effet, UML est un langage de modélisation des systèmes standard et graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

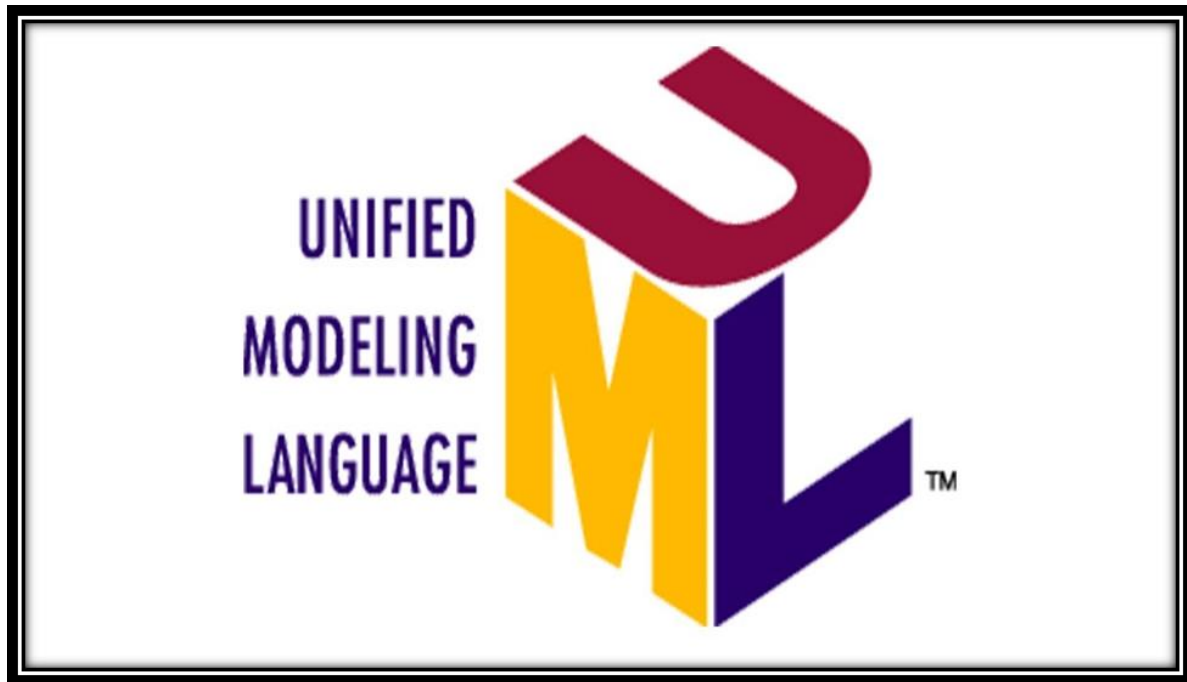


Figure 3: Logo d'UML

Les principaux avantages du langage UML sont :

- UML est un langage formel et normalisé : il représente un gain de précision et de stabilité.
- UML est un support de communication performant : il permet grâce à sa représentation graphique, d'exprimer visuellement une solution objet, de faciliter la comparaison et l'évolution de solution.
- Il est devenu le standard en terme de modélisation objet, universellement reconnu celui-ci est polyvalent et performant.
- Les processus métier sont définis en amont, de façon plus globale, plus proche du métier et donc plus accessible.
- Il est valable pour n'importe quel langage de programmation. [3]

Nous justifions notre choix par le fait que l'UML répond à nos besoins et il est le plus apte à représenter les spécifications et exigences demandées. Il répond aussi aux besoins de notre future Système car il intègre la notion d'objet. Comme il est plus proche du langage de programmation.

Au cours de ce rapport, nous allons établir les diagrammes UML suivants :

- **Le diagramme des cas d'utilisation**
- **Le diagramme des séquences**
- **Le diagramme d'activité**
- **Le diagramme de classes**

Conclusion

Après avoir présenté du projet et le principe de gestion commerciale avec les différentes tâches, la spécification des besoins constitue une partie essentielle du cycle de vie de tout projet informatique, ce qui va être le sujet du chapitre suivant.

Chapitre II: Analyse et spécification des besoins

Introduction

Au cours de ce chapitre, nous allons analyser en premier lieu les besoins fonctionnels et non-fonctionnels auxquels doit répondre de notre futur système.

Dans La deuxième partie, nous identifierons les acteurs ainsi que les différents cas d'utilisations du système.

I. Analyse des besoins :

Dans cette partie, nous exposerons l'ensemble des besoins auxquels doit répondre de système à développer. Nous allons mettre le point sur un ensemble de fonctionnalités dans le cadre de notre projet. L'importance de chaque module lui donne une classification, aussi son exigence permet de réorganiser les besoins afin de juger si un module est fonctionnel ou non fonctionnel.

I.1. Les besoins fonctionnels :

Dans ce qui suit, nous allons énumérer les différents besoins fonctionnels de Notre système issus du cahier des charges du notre projet.

Notre système proposé est divisée en deux parties :

- ❖ **La partie frontale, ou front office**, représente l'interface qui doit être manipulé par les utilisateurs donc c'est la partie visible du système.
- ❖ **La partie back-office** responsable de la mise en œuvre des opérations d'administration.

I.1.1. Partie front office :

La figure 3 résume les fonctionnalités constituant la partie front Office du notre application.

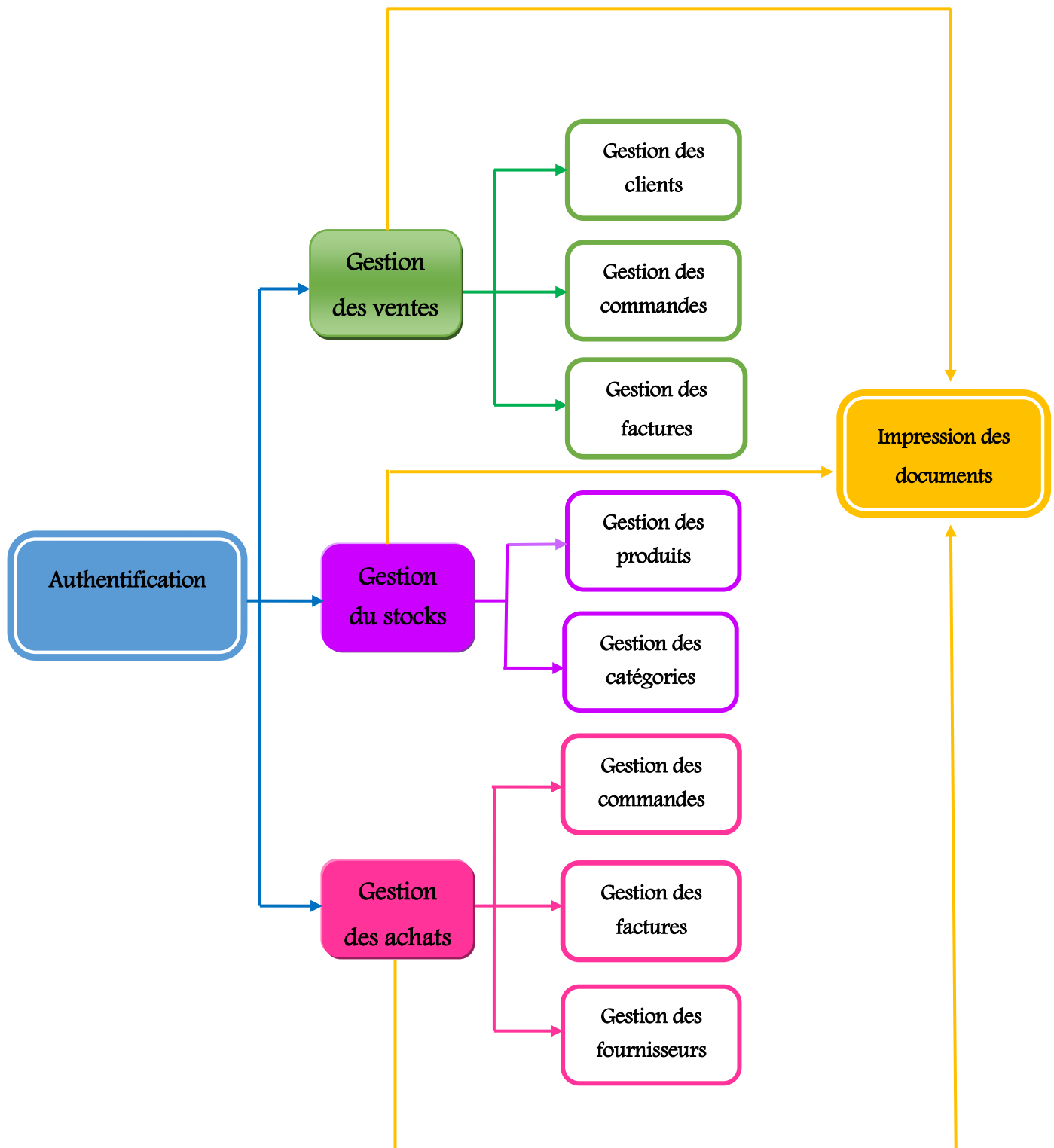


Figure 4:La partie Front office

I.1.1.a. Module gestion des ventes :

Ce module se situe à l'intersection des fonctionnalités de la gestion des clients et des ventes ainsi que la gestion des produits. Et à partir de ces tâches on peut faire les points suivants :

↳ **Gestion des commandes:** est l'enregistrement d'un document établi par le client dans lequel, il détaille les produits informatiques commandés et toutes les conditions utiles à la bonne exécution de la commande. A partir de la commande on peut sortir la facture et le bon de livraison.

↳ **Gestion des clients :** les clients seront identifier leurs coordonnées. Des fiches de contact seront prévues pour rentrer les personnes que nous devons contacter chez le client.

↳ **Gestion des factures :** est le document établi par le vendeur et adressé au client à la suite d'une vente, qui détaille le type de marchandise vendu, la quantité, le décompte du prix et la somme totale due par le client.

↳ **Gestion des livraisons :** est le fait de transporter des biens et d'assurer leur acheminement jusqu'à les clients.

I.1.1.b. Module gestion des stocks :

Ensemble des activités et techniques qui établissent les références à tenir en magasin, les quantités d'articles associés à ces références. Et à partir de ces tâches on peut faire les points suivants :

↳ **Gestion des produits:** elle permet de visualiser très rapidement toutes les informations relatives à un produit (ressources ou services) :

- Caractéristiques générales: prix, catégorie, type, statut (disponible ou non).
- Caractéristiques techniques telles que la version, la marque...
- Un onglet Photos permet d'associer un visuel au produit : photographies du produit.

I.1.1.c. Module gestion des achats :

Il représente tous activités qui sont liés à l'achat de l'entreprise.

↳ **Gestion des commandes:** est l'enregistrement d'un document établi par l'acheteur dans lequel, il détaille les produits informatiques commandés et toutes les conditions utiles à la bonne exécution de la commande. A partir de la commande on peut sortir la facture.

↳ **Gestion des factures** : c'est l'enregistrement des factures des produits achetés dans la base de données à travers le système. Des formulaires seront saisis pour collecter des informations sur les produits achetés.

↳ **Gestion des fournisseurs** : les fournisseurs seront identifier leurs coordonnées. Des fiches de contact seront prévues pour rentrer les personnes que nous devons contacter chez le fournisseur.

I.1.2. Partie back office :

On va s'intéresser à la partie administrative de notre système dont les principales fonctionnalités sont présentées par la figure 4:

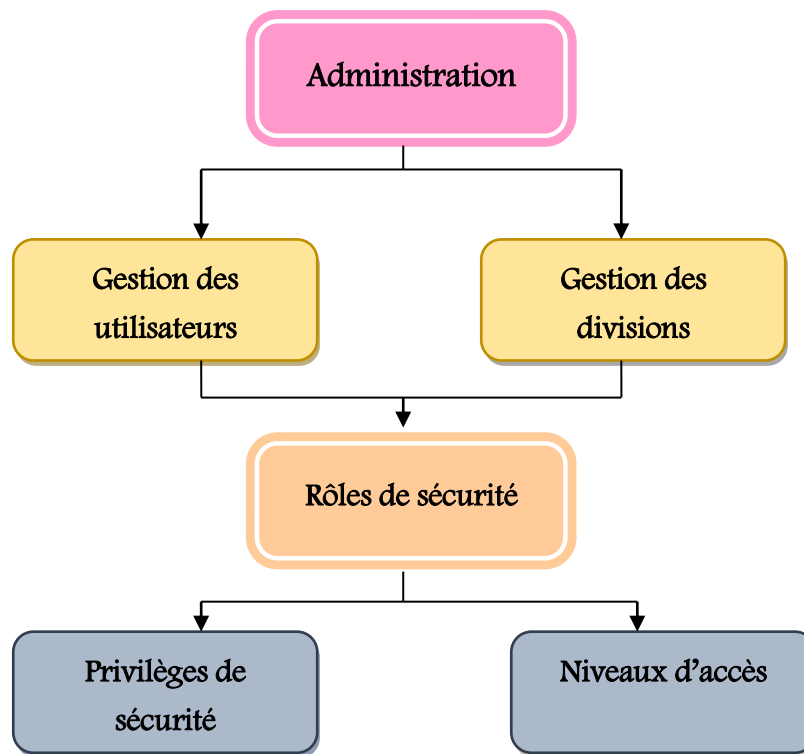


Figure 5: Fonctionnalités du Back office

↳ **Gestion des utilisateurs (membres)** : Un administrateur peut créer des utilisateurs et de définir les rôles de sécurité des groupes du travail.

↳ **Les groupes du travail** : sert à affecter les utilisateurs à un groupe du travail. Ce groupement est utile surtout pour l'affectation des rôles de sécurité.

↳ **Rôles de sécurité** : les groupes du travail peuvent-être associés à un ou plusieurs rôles de sécurité. Ces rôles leur donnent des privilèges et des niveaux d'accès.

Chaque rôle de sécurité englobe un ensemble des privilèges de sécurité (lecture, écriture, suppression, ajout...) pour des niveaux d'accès différents (aucun, tous les groupes).

I.2. Les Besoins non fonctionnelles :

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés, Parmi ces besoins on cite :

- **Sécurité** : L'application doit être accessible à plusieurs utilisateurs à la fois. Chaque utilisateur ne peut avoir accès qu'à certaines informations. L'utilisateur ne peut consulter que l'espace qui lui est réservé.
- **Performance** : L'application doit être rapide, fiable et stable même sous une charge importante.
- **Ergonomie** : Il s'agit d'une application de gestion de la relation client donc l'aspect esthétique ne doit pas être négligé. Ainsi les utilisateurs pourront s'y trouver plus facilement. L'interface doit être adaptée à aux interfaces de terminaux mobiles.
- **Stabilité** : Notre application doit être stable dans son environnement d'exécution et elle ne doit pas contenir des bogues bloquants.

II. Analyse et spécification du projet:

Après avoir déterminé les besoins fonctionnelles et non fonctionnelles de notre futur système, nous identifions dans ce qui suit les acteurs du système et nous détaillons les diagrammes de cas d'utilisation de chacun de ses modules.

II.1. Identification des acteurs :

Un acteur est tout utilisateur du système qui peut être une personne, un système informatique ou un matériel. En se basant sur cette définition, nous avons identifié les acteurs du future système qui sont au nombre de deux :

- ✚ **Administrateur** : Il a tous les droits. Il gère les groupes de travail et les employés ainsi que les rôles de sécurité et se charge de les attribuer à chaque groupe et les différents services.
- ✚ **Employer** : Un employé appartient au personnel de l'application. En fonction de la nature de sa fonction il est classé dans un groupe de travail et se dote d'un ensemble de rôles de sécurité.

Chacun de ces acteurs va utiliser l'application pour réaliser des tâches bien précises et qui lui sont confiées.

II.2. Description textuelles des cas d'utilisations

Les tâches principales de chaque acteur de notre futur système sont présentées dans le tableau suivant :

Tableau 1: Les tâches principales de chaque acteur de notre système

Acteurs	Descriptions
Administrateur	<ul style="list-style-type: none">○ S'authentifier○ Gérer les utilisateurs○ Gérer les groupes de travail○ Gérer les rôles de sécurité
Employer	<ul style="list-style-type: none">○ S'authentifier○ Gérer les ventes<ul style="list-style-type: none">● Gérer les clients● Gérer les commandes● Gérer les livraisons● Gérer les factures○ Gérer les achats<ul style="list-style-type: none">● Gérer les commandes● Gérer les factures● Gérer les fournisseurs○ Gérer les stocks<ul style="list-style-type: none">● Gérer les produits

	<ul style="list-style-type: none"> Gérer les catégories
--	--

II.3. 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. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases).

[6]

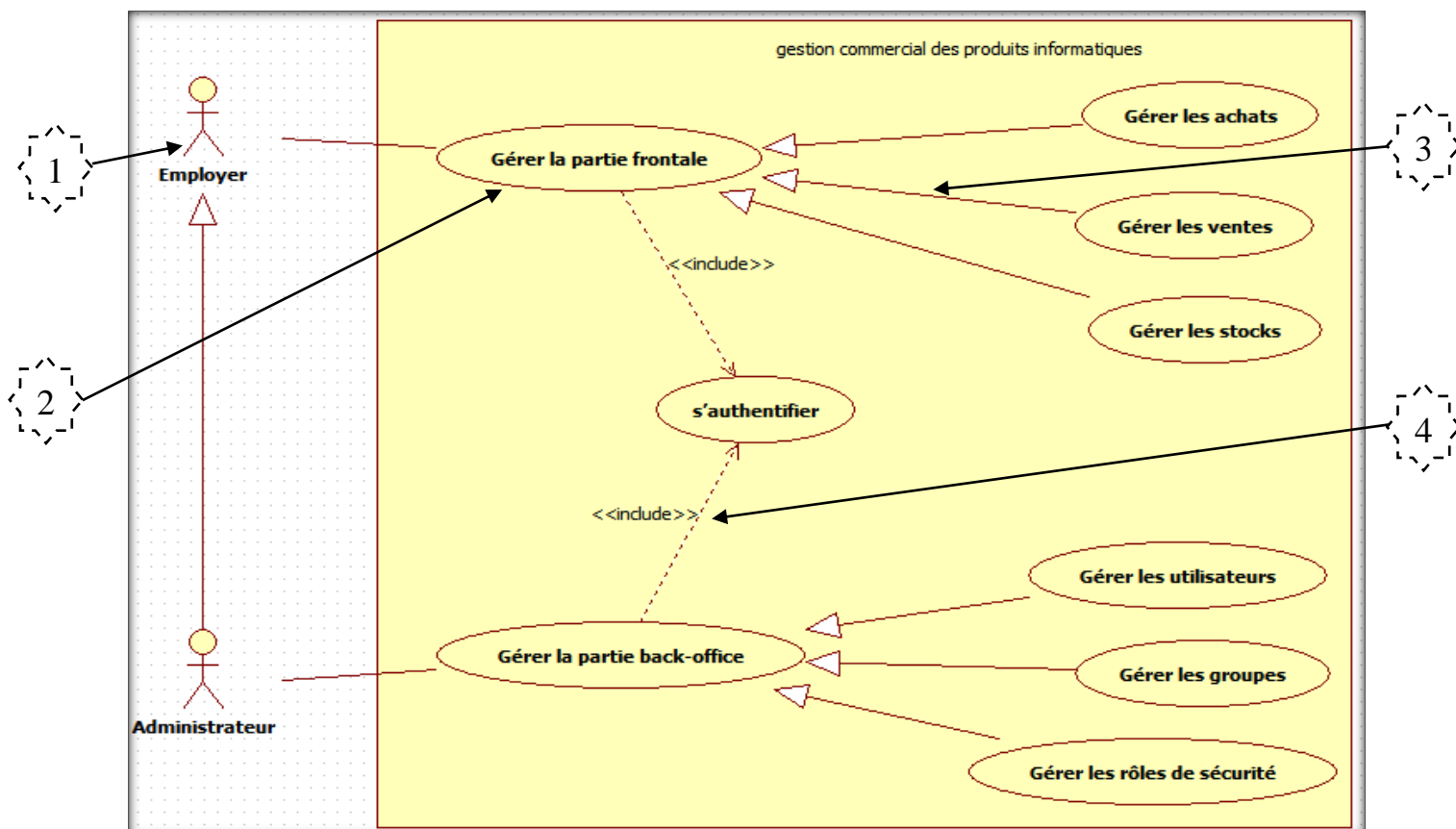


Figure 6: Diagramme de cas d'utilisation global

1 : Acteur 2 : Cas d'utilisation 3 : Relation de généralisation 4 : Relation d'inclusion

La figure (6) ci-dessus représente le diagramme général des cas d'utilisation. Elle explore les différentes fonctionnalités de notre système.

II.4. Relations entre les cas d'utilisation :

Avec UML, il est en effet possible de détailler et d'organiser les cas d'utilisations en ajoutant des relations d'inclusion, d'extension et de généralisation entre cas d'utilisation.

En effet, pour clarifier un diagramme, UML permet d'établir des relations entre les cas d'utilisation. Il existe deux types de relations : les dépendances stéréotypées et les généralisations/spécialisations. Les dépendances stéréotypées les plus utilisées sont l'inclusion et l'extension. [7]

II.4.1. Relation d'association :

Une relation d'association est un lien de communication entre un acteur et un cas d'utilisation. [8]

Dans notre cas :

- Il Une relation d'association entre l'acteur Employer et le cas utilisation Gérer la partie frontale.
- Il Une relation d'association entre l'acteur Administrateur et le cas utilisation Gérer la partie back-office.

Voilà un exemple ci-dessous :

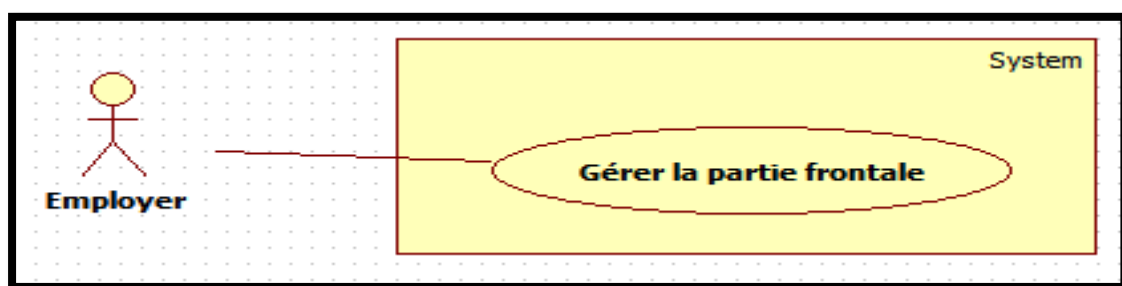


Figure 7: Relation d'association

II.4.2. Relation d'inclusion

Un cas d'utilisation A est inclus dans un cas B si le comportement décrit par le cas 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 ». [7]

Dans notre cas, pour qu'il puisse gérer la partie frontale et gérer la partie back-office, l'administrateur doit nécessairement s'authentifier. Voilà un exemple ce dessous :

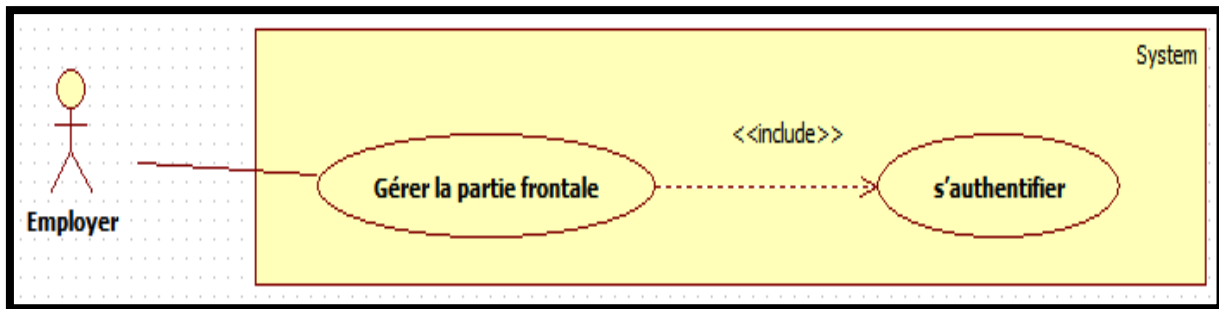


Figure 8:Relation d'inclusion

II.4.3. 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. Graphiquement, la condition est exprimée sous la forme d'une note. [7]

II.4.4. Relation de généralisation

Un cas A est une généralisation d'un cas B si B est un cas particulier de A. [7]

Dans notre cas :

- ✓ La Gestion des achats, des stocks, des ventes sont des cas particuliers du cas gestion de la partie frontale.
- ✓ La Gestion des utilisateurs, des groupes, des rôles de sécurité sont des cas particuliers du cas gestion de la partie back-office.

Voilà un exemple ce dessous :

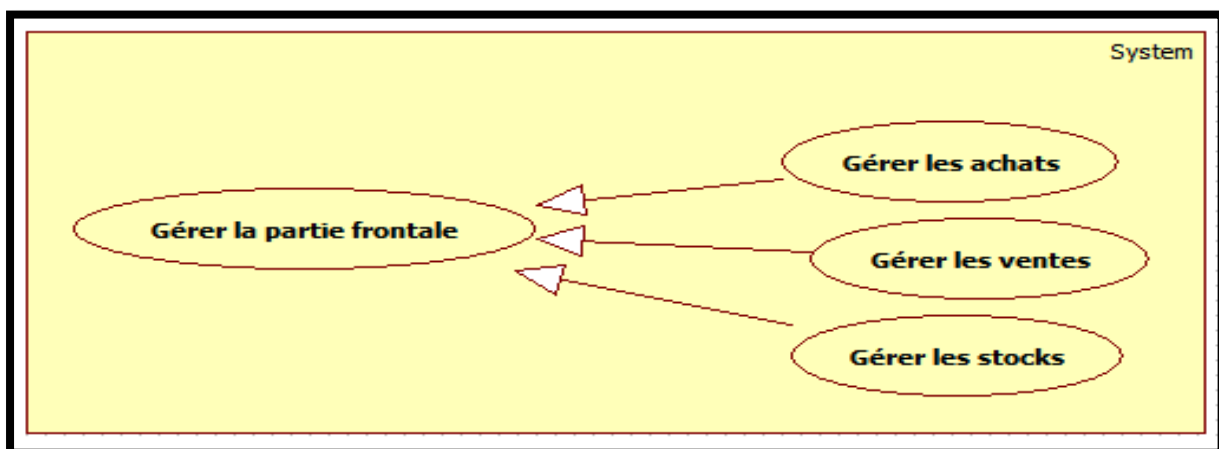


Figure 9:Relation de généralisation

II.5. Cas d'utilisations spécifiques :

Dans cette partie nous entend par « Gestion » les opérations CRUD : création, lecture, mise à jour et suppression des entités en question.

II.5.1. Partie Front office :

Dans la partie qui suit nous citons les différents diagrammes de cas d'utilisation de chaque module.

- *Modules gestion des ventes :*

- Un agent de vente se charge de gérer les commandes d'un client ainsi que l'ajout et la mise à jour des informations d'un client données.
- Le module de vente se charge aussi d'imprimer les documents nécessaires à la commande et aussi les coordonnées ou la liste des clients.
- Un agent de vente se charge de gérer les factures des clients et gérer les livraisons.
- Le diagramme de la figure 9 présente les fonctionnalités offertes pour la gestion des ventes.

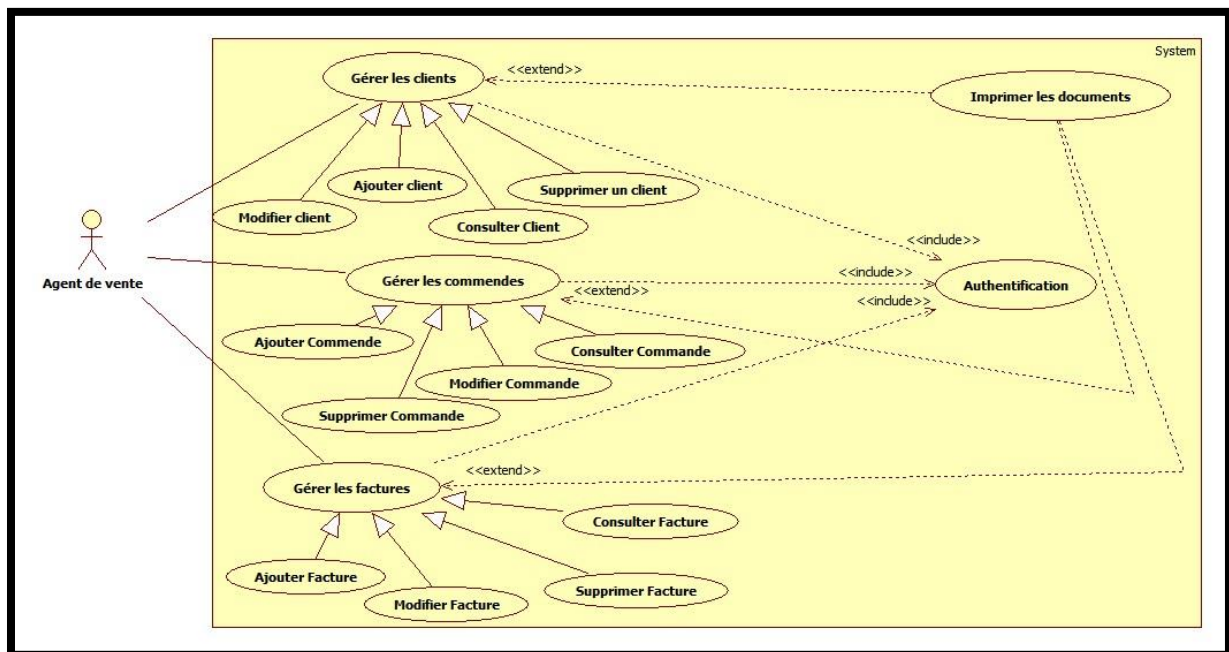


Figure 10: Diagramme de cas d'utilisation « Modules gestion des ventes »

- *Modules gestion des stocks :*

- Le diagramme de la figure 10 présente les fonctionnalités offertes pour la gestion des stocks.
- Ce service permet à un agent de stock de gérer un ensemble des produits informatiques ainsi que la gestion de son catégorie. Il sert aussi d'imprimer les documents liés à un produit.

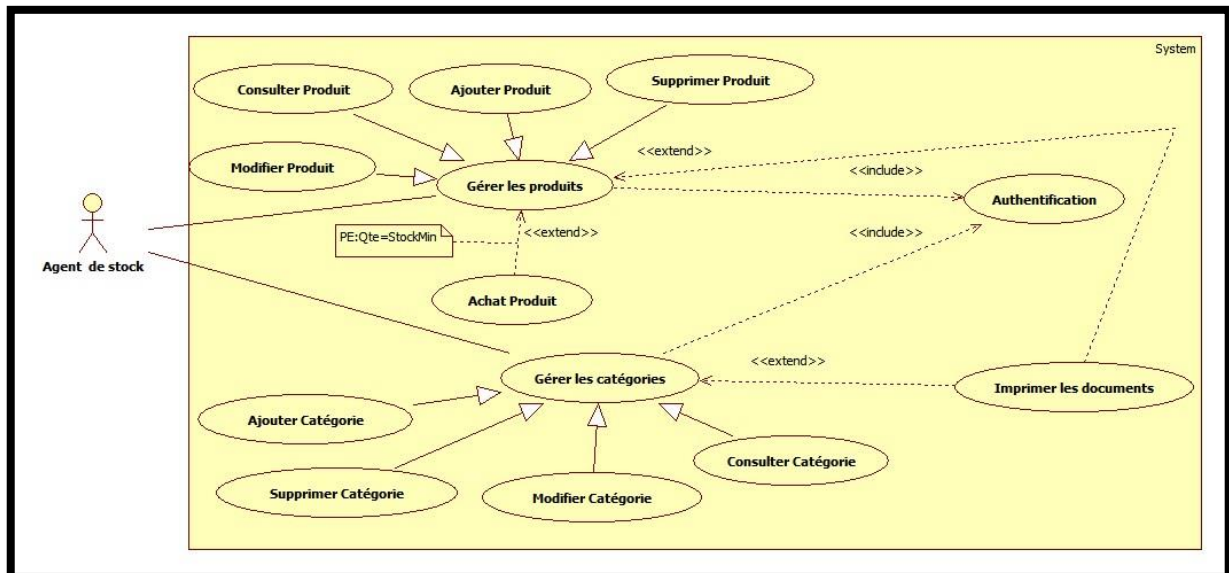


Figure 11: Diagramme de cas d'utilisation « Modules gestion des stocks »

- *Modules gestion des achats :*

- Le diagramme de la figure 11 présente les fonctionnalités offertes par le système pour la gestion des achats.
- Un agent d'achat se charge de gérer les commandes et gérer les factures des fournisseurs.
- Ces fonctionnalités se résument principalement en la gestion des fournisseurs tels que l'ajout des informations des fournisseurs ainsi que la mise à jour de ces informations. Nous avons aussi la gestion des livraisons c'est ce lui qui assurent la livraison des produits informatiques chez le client.

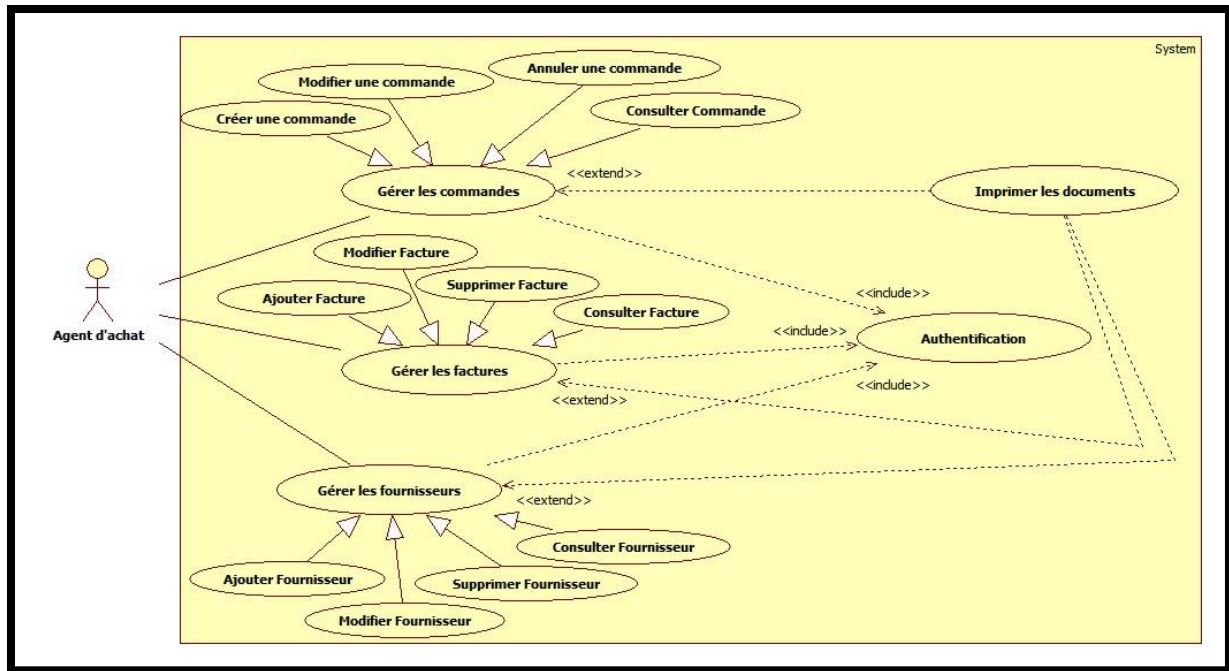


Figure 12: Diagramme de cas d'utilisation « Modules gestion des achats »

II.5.2. Partie back-office :

- L'administrateur de notre application à la possibilité et les privilèges de gérer les utilisateurs (employés) ainsi que les rôles de sécurité affectés aux groupes du travail.
- L'administrateur peut aussi répartir les utilisateurs sur des groupes de travail pour mieux faciliter l'affectation des rôles de sécurité.
- La figure suivante résume les fonctionnalités de la partie back-office.

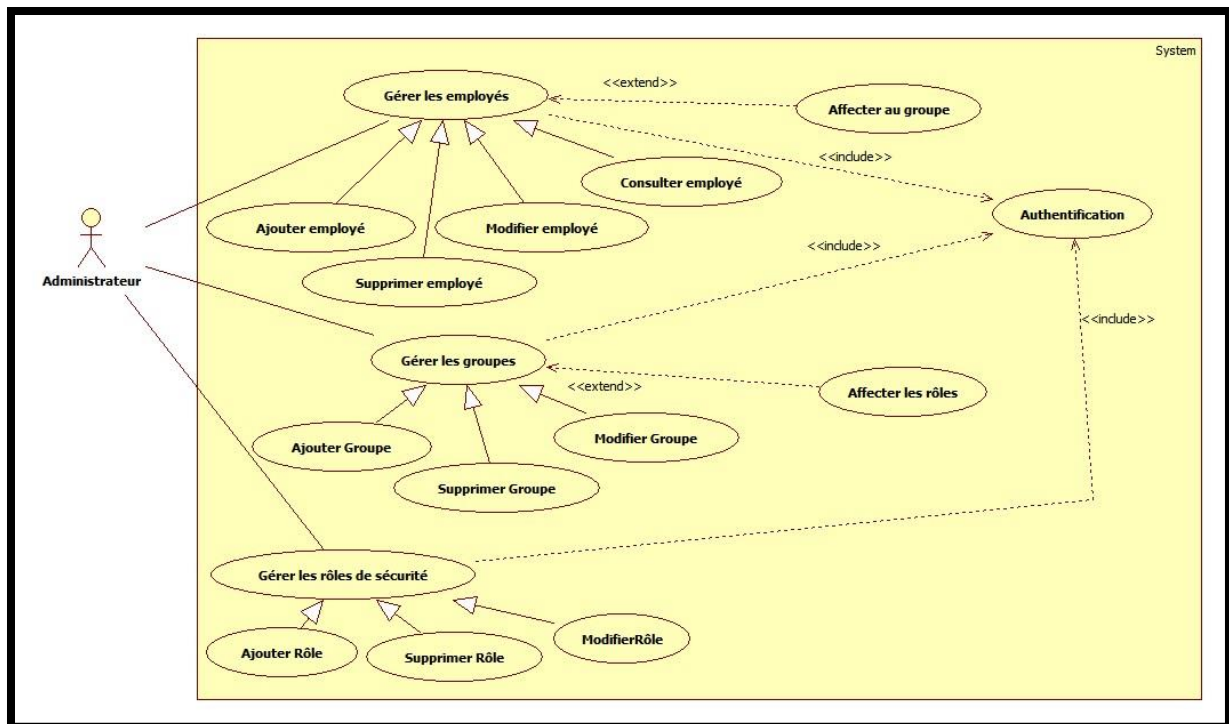


Figure 13: Diagramme de cas d'utilisation « Modules gestion d'administrateur »

Conclusion

La spécification des besoins nous a permis d'avoir une vision plus claire sur le sujet et une compréhension plus profonde des tâches à réaliser. La spécification fonctionnelle et non fonctionnelle des besoins va être la base sur laquelle nous allons réaliser la conception détaillée du système ainsi que notre base de données qui fera l'objet du chapitre suivant.

Chapitre III: Conception de l'application

Introduction

Après la spécification détaillée des principaux besoins, nous nous concentrons sur la phase de conception qui permet de bien mener la phase d'implémentation.

Dans ce chapitre nous allons identifier les composants de l'application ainsi que l'interaction entre eux.

Par suite, on utilise les diagrammes d'UML (Unified Modeling Language) qui sont, les diagrammes de séquence et le diagramme de classes.

I. Description des scénarios, diagrammes de séquence et diagramme d'activité :

Un cas d'utilisation spécifie une séquence d'actions, avec variantes éventuelles, réalisée par le système en interaction du système.

- Les scénarios est une série d'événement ordonnés dans le temps, simulant une exécution particulière du système.

Les scénarios permettent d'expérimenter les exécutions du système, ils sont donc très utiles pour les phases de tests et de maintenance. [7]

- Les diagrammes de séquence : exprime la séquence des interactions entre objets selon un point de vue temporel.

Le diagramme de séquence est construit autour de trois éléments fondamentaux. Les objets, leur ligne de vie et les messages. [7]

- Un diagramme d'activités : permet de spécifier des traitements à priori séquentiels. Ils offrent un pouvoir d'expression très proche de langages de programmation objet : spécification des actions de base (déclaration de variables), structures de contrôle (conditionnels, boucles), ainsi que les instructions particuliers à la programmation orientée objet (appels d'opérations, exceptions.) [7]

Dans cette partie nous devons choisir à présenter l'enchaînement du module gestion du stock puisqu'il représente la même démarche pour le reste des modules.

Authentification

L'Authentification est une étape obligatoire pour tout utilisateur avant d'accéder à sa vue.

❖ Scénario d'exécution du cas d'utilisation «S'authentifier» :

Tableau 2: Cas d'utilisation « S'authentifier»

Cas	S'authentifier
Acteur	Tous les utilisateurs et l'administrateur
Description	Pour l'authentification et l'autorisation d'accès
Pré condition	Système en marche
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur demande l'accès au système 2. Le système affiche le formulaire d'authentification (login et mot de passe) 3. L'utilisateur saisit login et le mot de passe 4. L'utilisateur valide les données saisies 5. Le système vérifie les données saisies 6. Le système affiche l'interface appropriée
Scénario alternatif	Si le formulaire n'est pas bien rempli (un champ obligatoire non valide ou vide) le cas d'utilisation reprend à la 2ème étape du scénario nominal.
Post condition	Authentification réussie

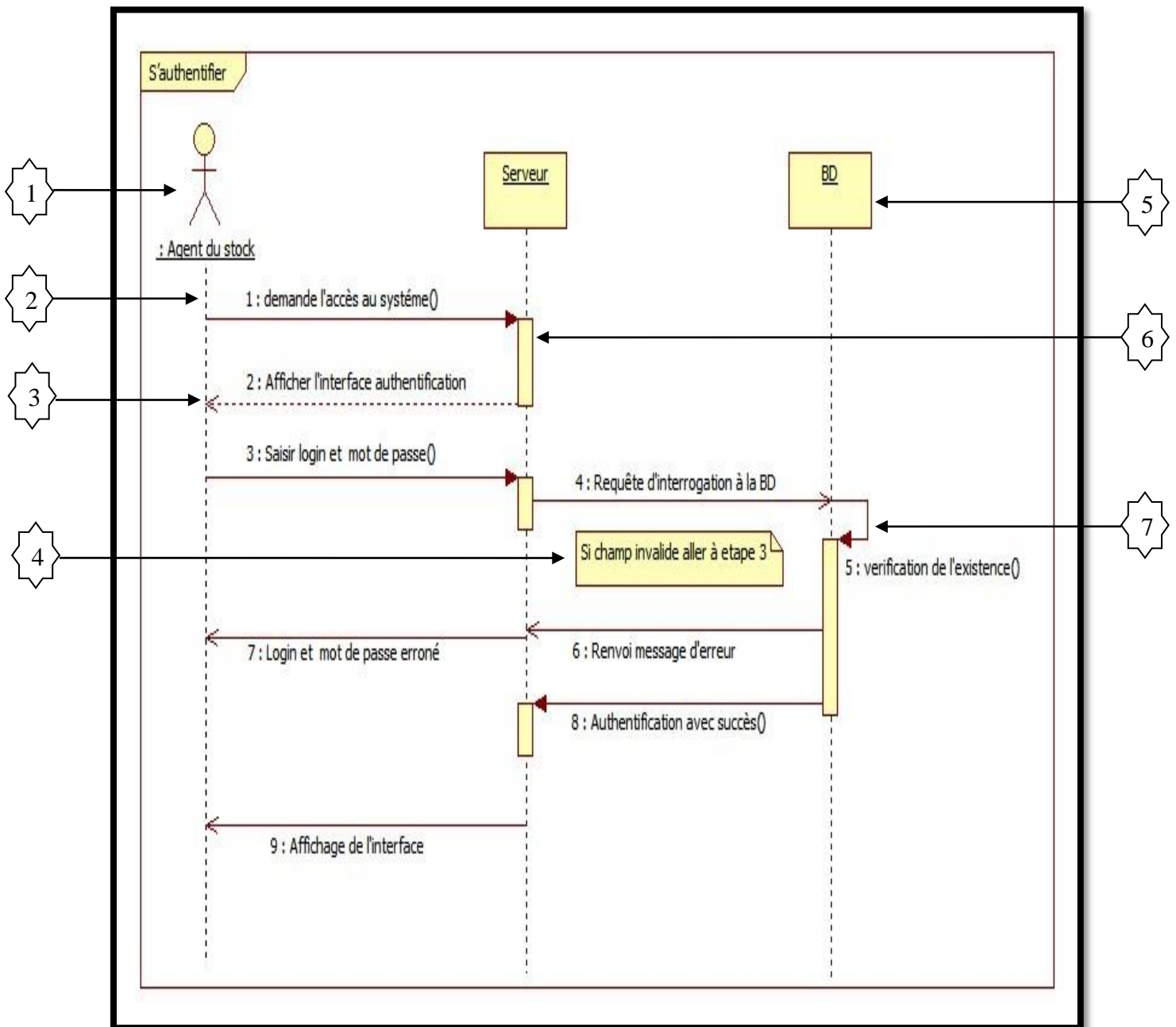
❖ Diagramme de séquence de cas d'utilisation «S'authentifier» :

Figure 14: Diagramme de séquence de cas d'utilisation « S'authentifier »

1 : Acteur 2 : Axe du temps 3 : Message de retour 4 : Note 5 : Objet système
 6 : Période d'activité 7 : Message réflexif

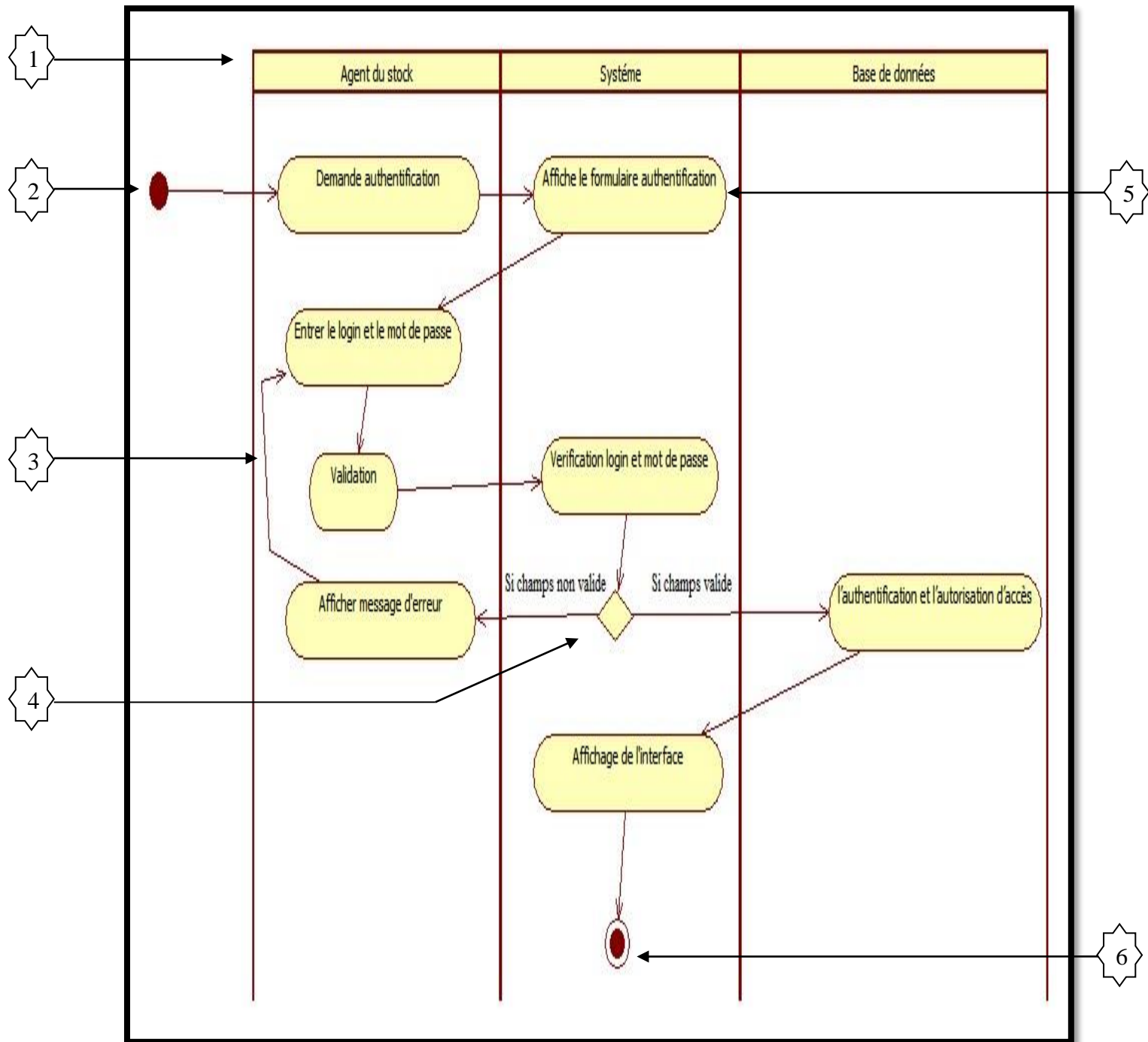
❖ Diagramme d'activité du cas d'utilisation « S'authentifier » :

Figure 2 : Diagramme d'activité de cas d'utilisation « S'authentifier »

1 : Objet 2 : Etat initial 3 : Transition 4 : Condition 5 : Activité 6 : Etat final

Gestion des produits

La Gestion des produits consiste à Ajouter, Modifier et Supprimer des Produits.

❖ Scénario d'exécution de cas d'utilisation «Ajouter produits» :

Tableau 3:Cas d'utilisation «Ajouter produits»

Cas	Ajouter produit
Acteur	Agent du stock
Description	Pour l'ajout d'un produit, l'agent du stock s'identifie et choisit l'opération : ajouter produit.
Pré condition	L'acteur est authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur choisit l'opération Ajout d'un produit 2. Le système affiche le formulaire correspondant 3. L'acteur remplit le formulaire (choix du produit à ajouter) 4. L'acteur valide le formulaire rempli 5. Le système effectue l'opération demandée
Scénario alternatif	Si le formulaire n'est pas bien rempli (un champ obligatoire non valide ou vide) le cas d'utilisation reprend à la 2ème étape du scénario nominal.
Post condition	Produit ajouté avec succès

❖ Diagramme de séquence de cas d'utilisation « Ajouter produits » :

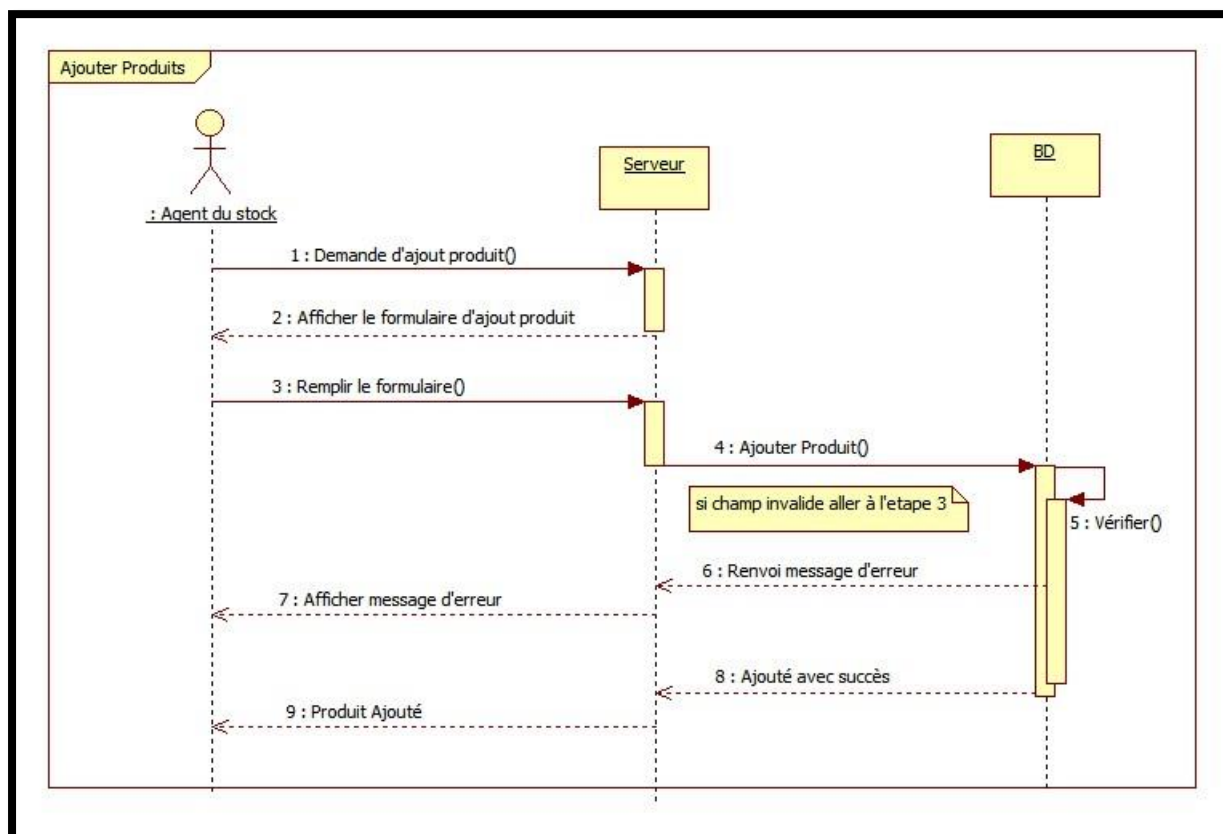


Figure 15: Diagramme de séquence « Ajouter Produits »

❖ **Diagramme d'activité de cas d'utilisation « Ajouter Produits » :**

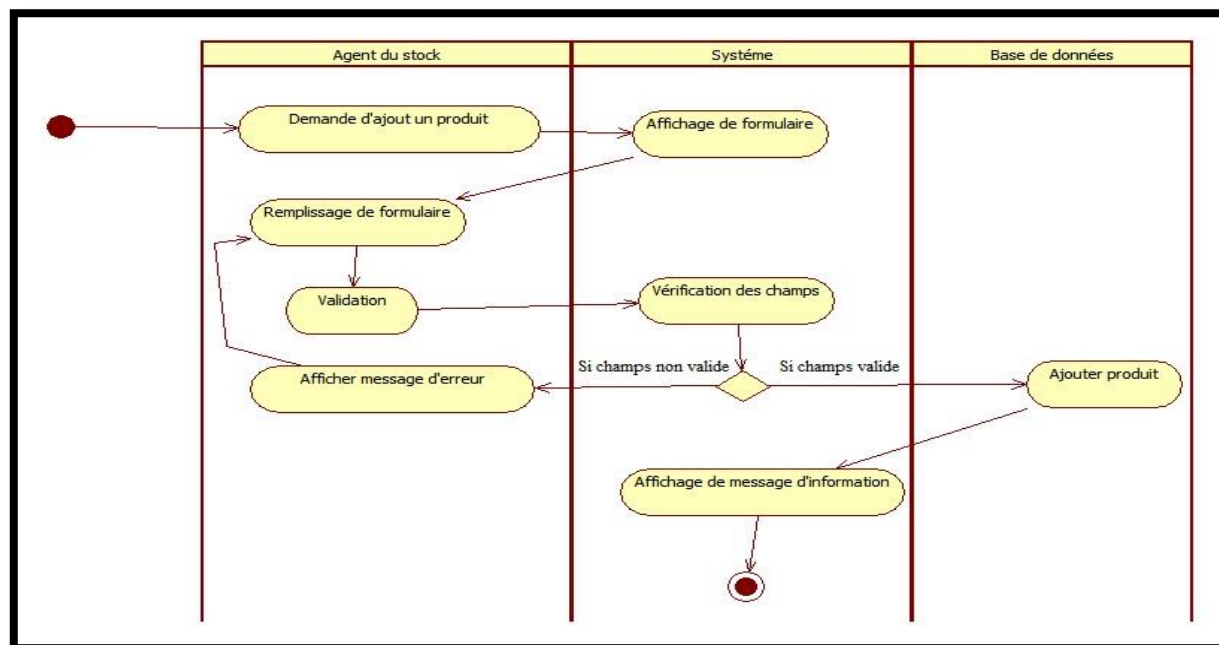


Figure 16:Diagramme d'activité « Ajouter Produits »

❖ **Scénario d'exécution de cas d'utilisation «Supprimer Produits» :****Tableau 4:**Cas d'utilisation «Supprimer Produits»

Cas	Supprimer Produit
Acteur	Agent du stock
Description	Pour la suppression d'un produit, l'Agent du stock s'identifie puis choisit l'opération : supprimer produit.
Pré condition	L'acteur est authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur choisit l'opération Suppression d'un produit 2. Le système affiche le formulaire correspondant 3. L'acteur remplit le formulaire (choix du produit à supprimer) 4. L'acteur valide le formulaire rempli 5. Le système effectue l'opération demandée
Scénario alternatif	Si le formulaire n'est pas bien rempli (un champ obligatoire non valide ou vide) le cas d'utilisation reprend à la 2ème étape de scénario nominale.
Post condition	Produit supprimé

❖ **Diagramme de séquence de cas d'utilisation « Supprimer Produits» :**

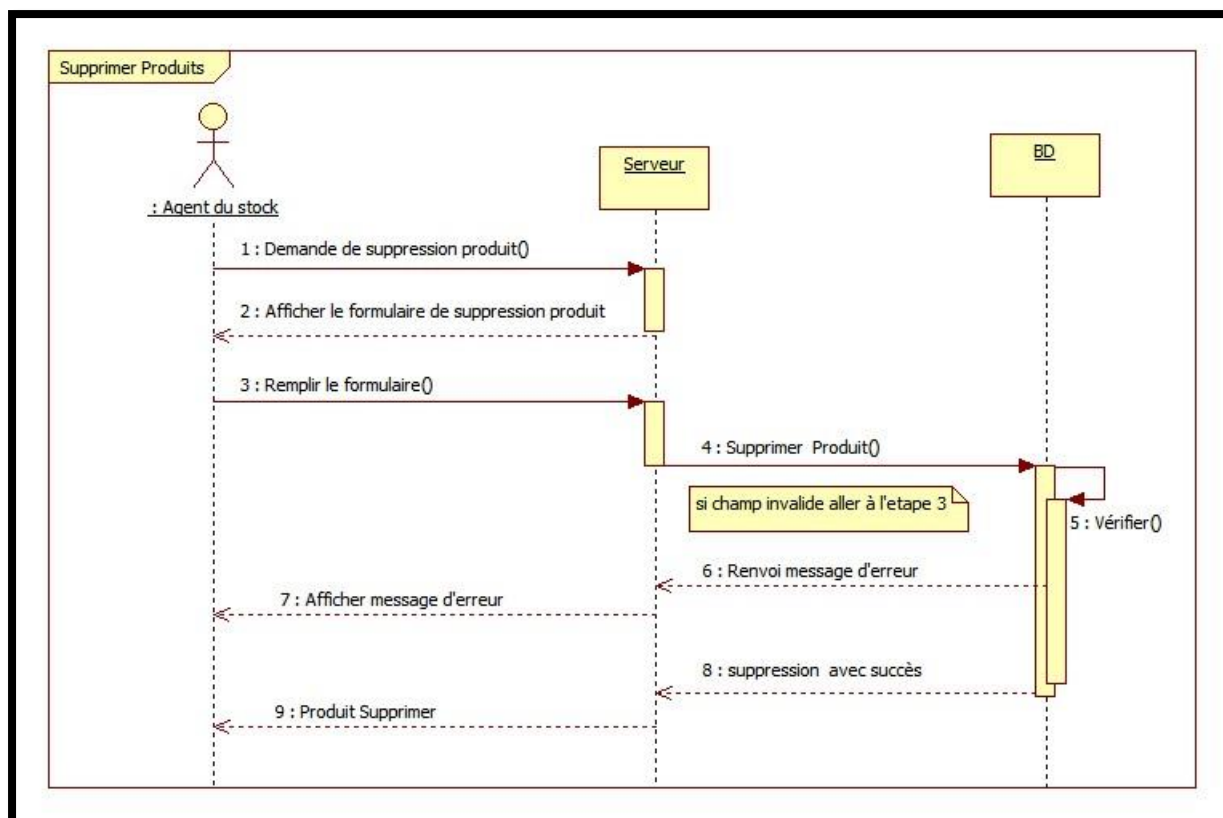


Figure 17: Diagramme de séquence « Supprimer Produit »

❖ **Diagramme d'activité de cas d'utilisation «Supprimer Produits» :**

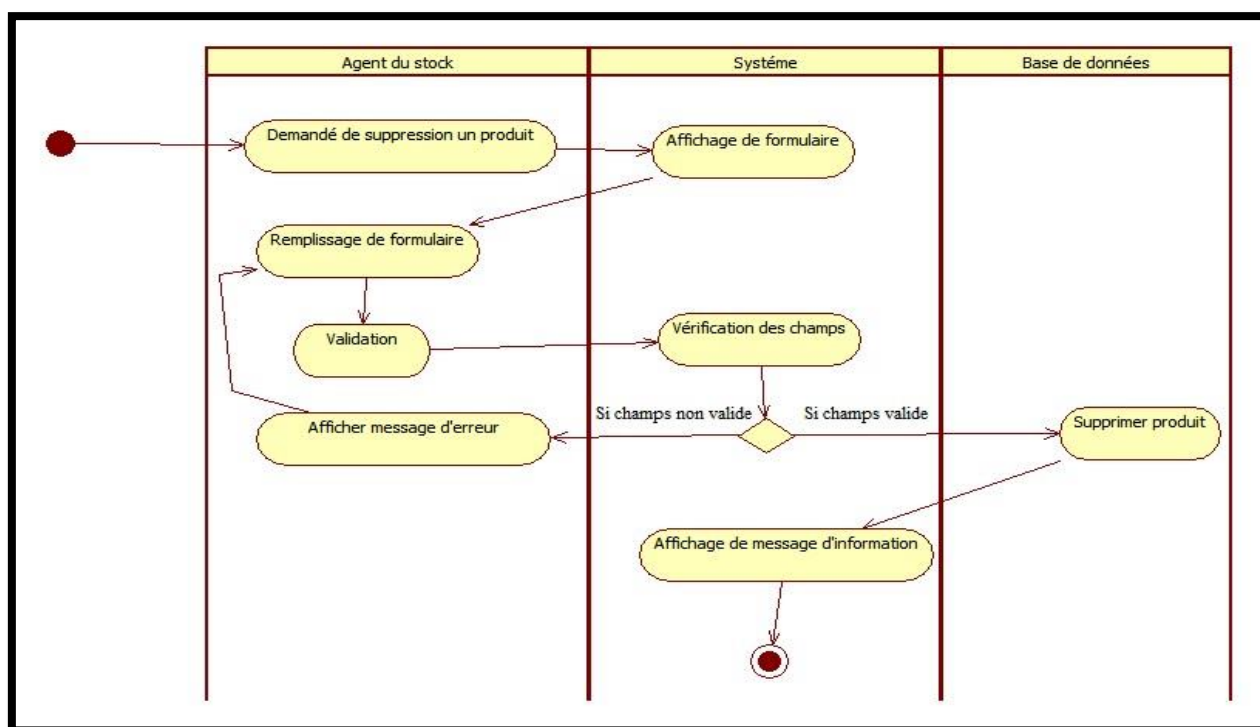


Figure 18: Diagramme d'activité « Supprimer produit »

❖ **Scénario d'exécution de cas d'utilisation « Modifier Produit » :****Tableau 5:**cas d'utilisation « Modifier Produit»

Cas	Modifier Produit
Acteur	Agent du stock
Description	Pour la modification des détails d'un produit, l'Agent du stock s'identifie puis choisit l'opération : Modifier produit.
Pré condition	L'acteur est authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur choisit l'opération de modification d'un produit 2. Le système affiche le formulaire correspondant 3. L'acteur remplit le formulaire (choix du produit à modifier) 4. L'acteur valide le formulaire rempli 5. Le système effectue l'opération demandée
Scénario alternatif	Si le formulaire n'est pas bien rempli (un champ obligatoire non valide ou vide) le cas d'utilisation reprend à la 2ème étape de scénario nominale.
Post condition	Détails du produit choisi modifiés

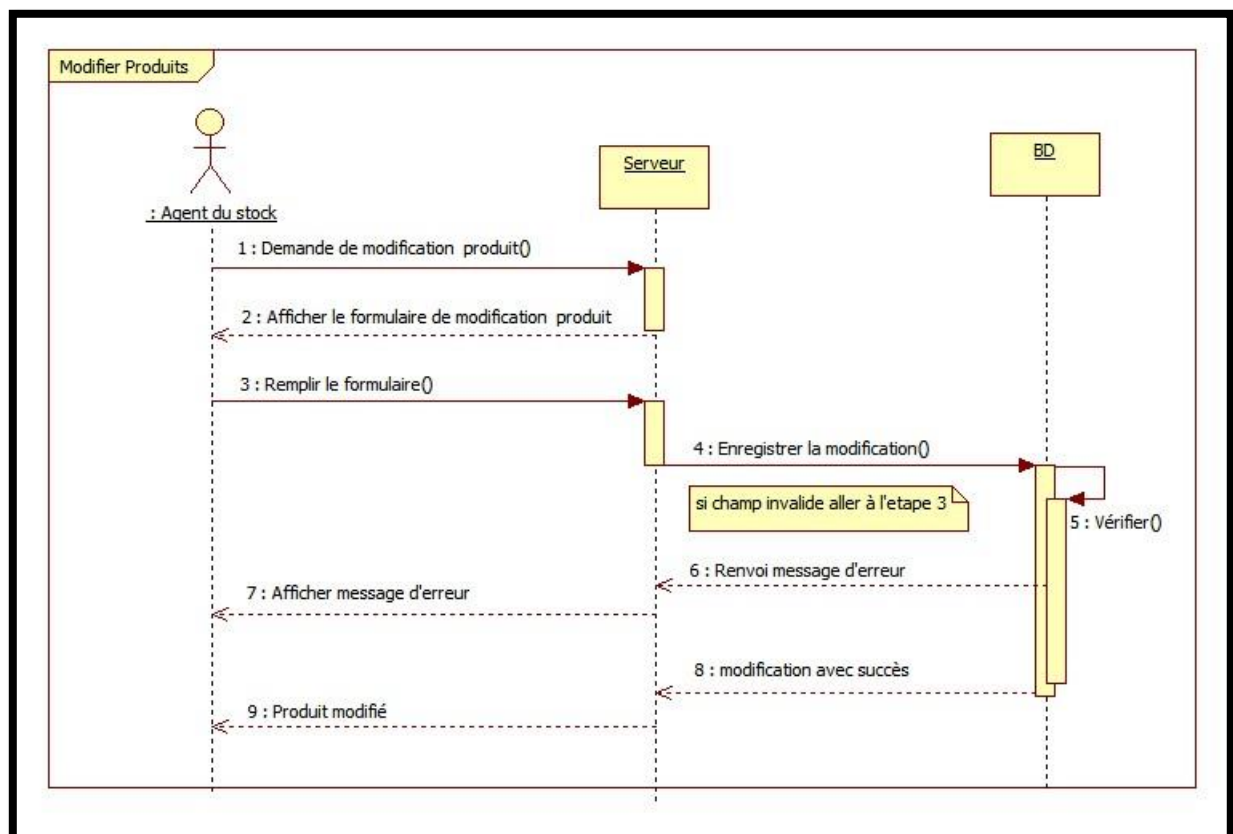
❖ Diagramme de séquence de cas d'utilisation « Modifier Produits » :

Figure 19: Diagramme de séquence « Modifier Produits »

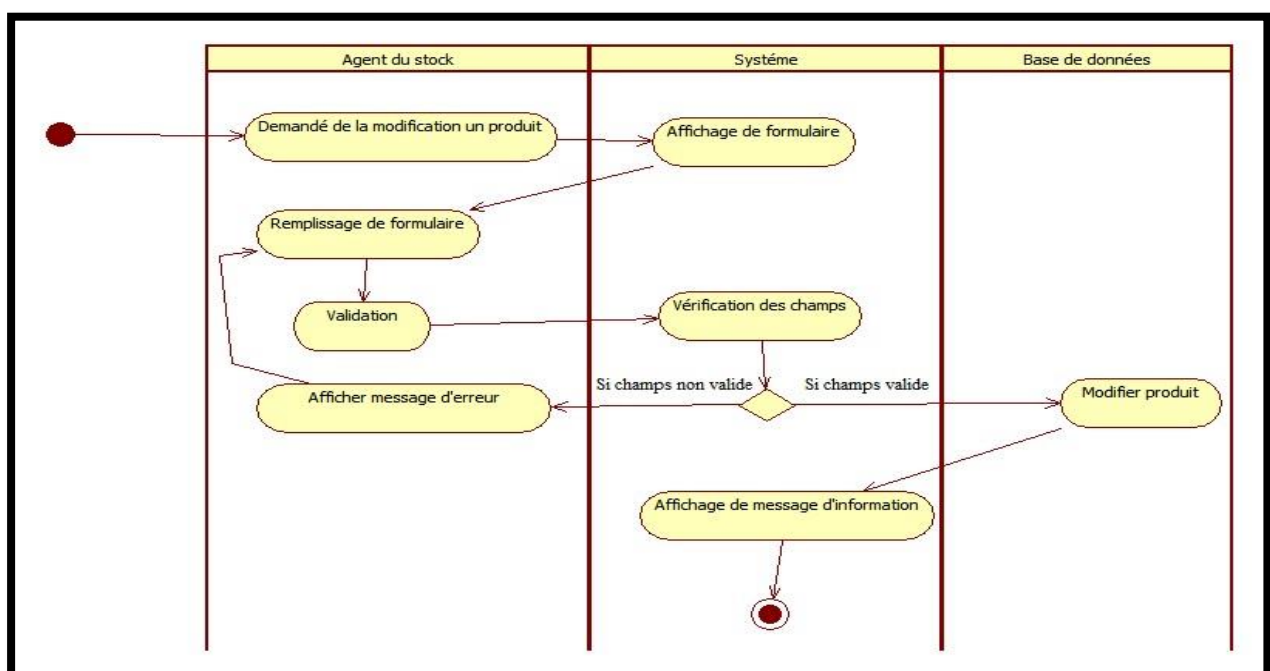
❖ Diagramme d'activité de cas d'utilisation «Modifier Produits» :

Figure 20:Diagramme d'activité « Modifier produit »

Gestion des commandes et factures :

❖ Scénario d'exécution de cas d'utilisation « Ajouter Commande » :

Tableau 6:cas d'utilisation «Ajouter Commandes»

Cas	Ajouter Commande
Acteur	Agent des ventes
Description	Pour l'ajout d'un commande, l'agent de vente s'identifie et choisit l'opération : Ajouter Commandes.
Pré condition	L'acteur est authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur choisit l'opération Ajout d'une commande 2. Le système affiche le formulaire correspondant 3. L'acteur remplit le formulaire (choix du cmd à ajouter) 4. L'acteur valide le formulaire rempli 5. Le système effectue l'opération demandée
Scénario alternatif	Si le formulaire n'est pas bien rempli (un champ obligatoire non valide ou vide) le cas d'utilisation reprend à la 2ème étape du scénario nominal.
Post condition	Commande ajouté avec succès

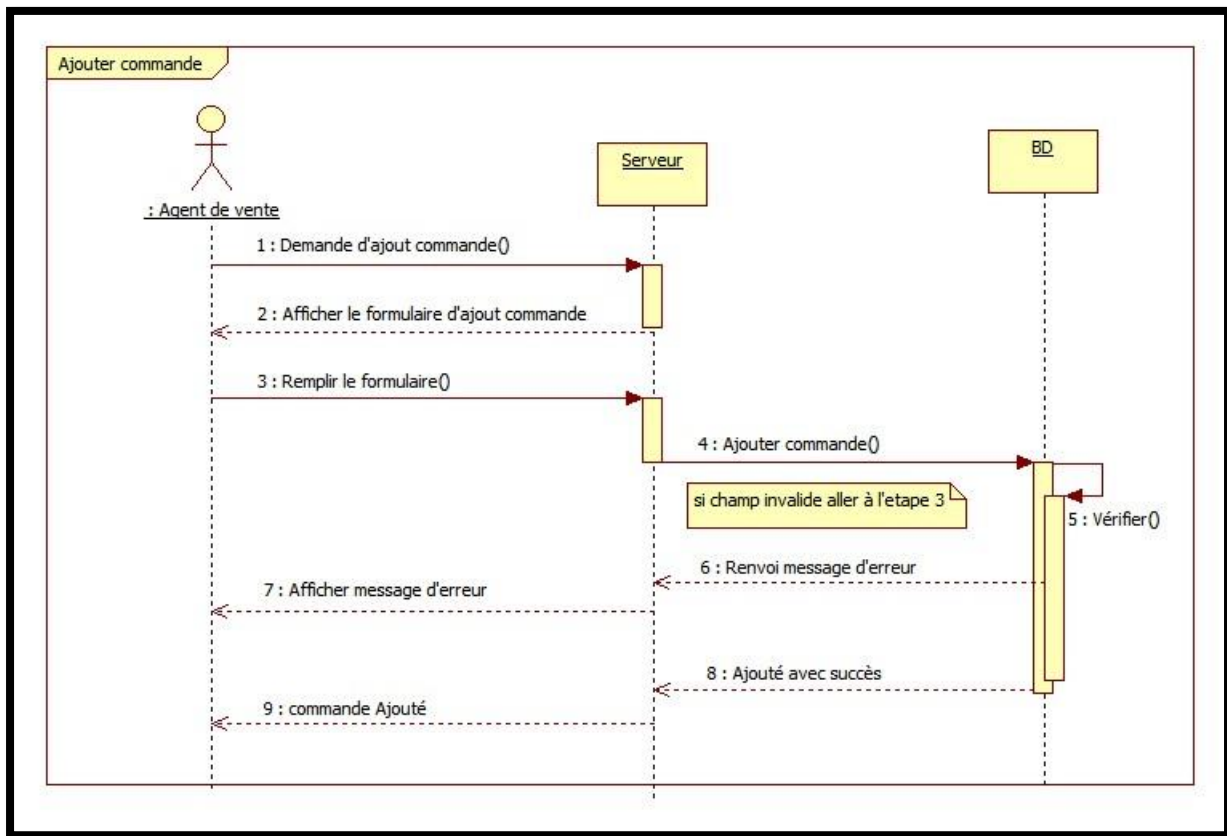
❖ Diagramme de séquence de cas d'utilisation «Ajouter Commande» :

Figure 21: Diagramme de séquence « Ajouter commande »

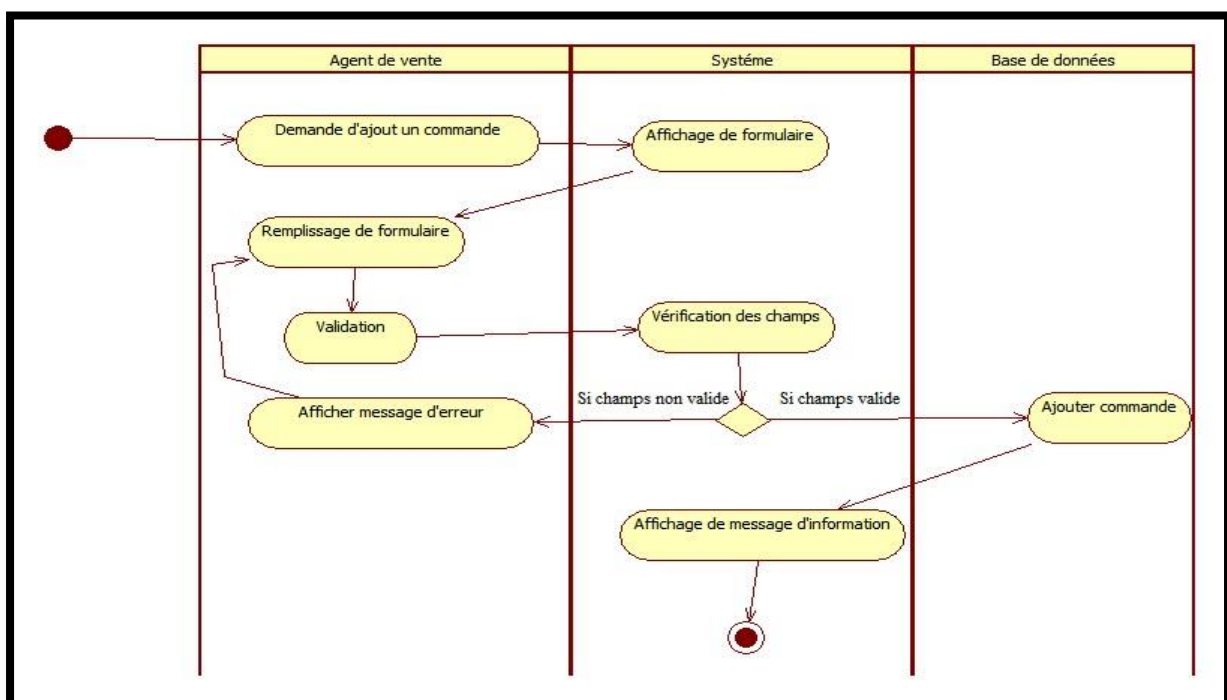
❖ Diagramme d'activité de cas d'utilisation «Ajouter Commande» :

Figure 22:Diagramme d'activité «Ajouter commande»

❖ **Scénario d'exécution de cas d'utilisation «Ajouter Facture» :****Tableau 7:**cas d'utilisation « Ajouter Facture»

Cas	Ajouter Facture
Acteur	Agent des ventes
Description	Pour l'ajout d'un Facture, l'agent de vente s'identifie et choisit l'opération : Ajouter Facture.
Pré condition	L'acteur est authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur choisit l'opération Ajout d'une facture 2. Le système affiche le formulaire correspondant 3. L'acteur remplit le formulaire (choix de la facture à ajouter) 4. L'acteur valide le formulaire rempli 5. Le système effectue l'opération demandée
Scénario alternatif	Si le formulaire n'est pas bien rempli (un champ obligatoire non valide ou vide) le cas d'utilisation reprend à la 2ème étape du scénario nominal.
Post condition	facture ajouté avec succès

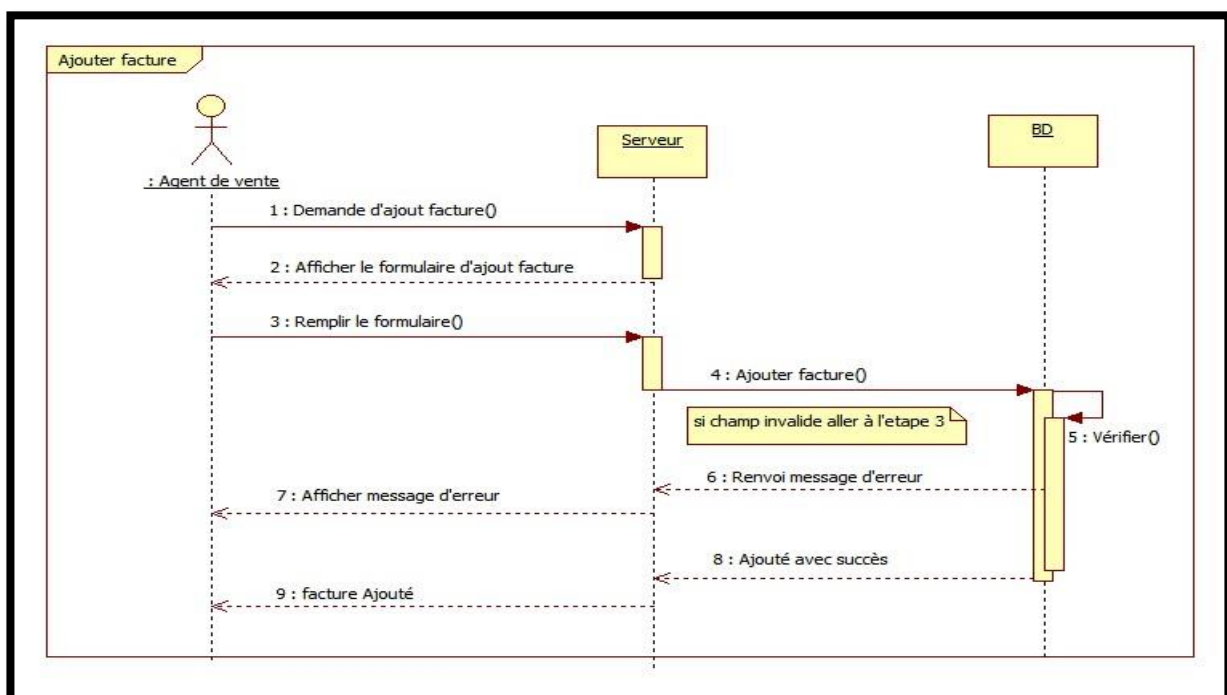
❖ **Diagramme de séquence de cas d'utilisation «Ajouter Factures»:**

Figure 23: Diagramme de séquence « Ajouter facture »

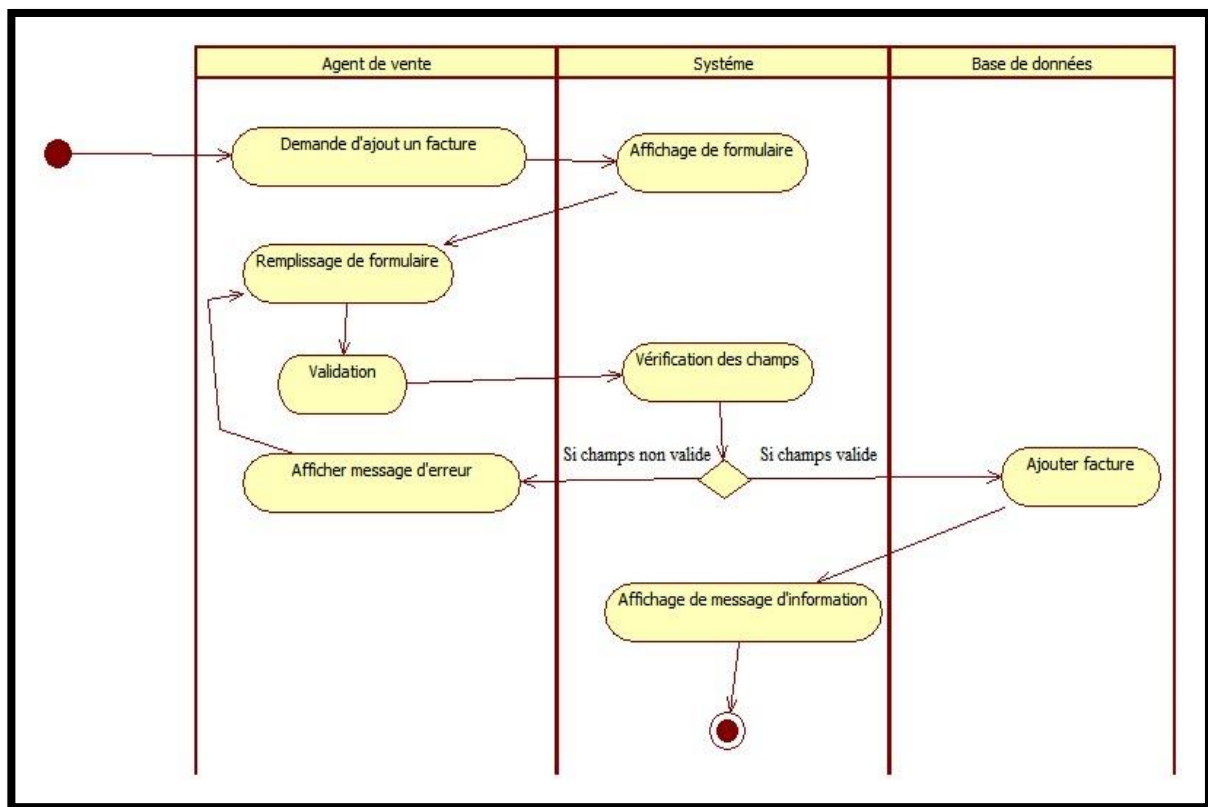
❖ Diagramme d'activité de cas d'utilisation «Ajouter Factures»:

Figure 24:Diagramme d'activité «Ajouter facture»

II. Développement du modèle statique :

Un modèle statique décrit la structure statique du système et les relations entre les composants. Nous allons construire le diagramme de classes qui constituent la forme de la base des données.

II.1. Le dictionnaire de données :

Pour la construction du diagramme de classes, Nous allons créer le dictionnaire de données suivant :

Tableau 8:dictionnaire de données

Numéro	Nom des variables	Désignation
01	Id_clt	L'identité de client
02	Nom_clt	Le nom de client
03	Prénom_clt	Le prénom de client
04	Adresse_clt	L'adresse de client
05	Téléphone_clt	Le téléphone de client
06	Email_clt	L'email de client
07	Ville_clt	La ville de client
08	Région_clt	La région de client
09	Code_Postal_clt	Le code postal de client
10	Pays_clt	Le pays de client
11	Id_Cmd	L'identité de commande du client
12	Date_Cmd	la date de commande du client
13	État_Cmd	L'état de commande du client
14	Date_Livr_clt	Date de livraison du client
15	Délai_Livr_Clt	Délai de livraison du client
16	Qte_Cmd	La quantité de commande du client
17	DescriptionCmd	La description de commande du client
18	Num_Fact_Clt	Le numéro de facture du client
19	Qte_Fact_Clt	La quantité de facture du client

20	prix_Fact_Clt	Le prix de facture du client
21	Date_Fact_Clt	La date de facture du client
22	Date_Échéance	La date d'échéance du client
23	DescriptionClt	La description de facture du client
24	Id_Fr	L'identité de fournisseur
25	Nom_Fr	Le nom de fournisseur
26	Adresse_Fr	L'adresse de fournisseur
27	Téléphone_Fr	Le téléphone de fournisseur
28	Fax_Fr	Le fax de fournisseur
29	Email_Fr	L'email de fournisseur
30	Ville_Fr	La ville de fournisseur
31	Région_Fr	La région de fournisseur
32	Code_Postal_Fr	Le code postal de fournisseur
33	Pays_Fr	Le pays de fournisseur
34	Id_Cmd_Achat	L'identité de commande d'achat
35	Date_ Achat	la date d'achat
36	État_ Achat	L'état d'achat
37	Date_Livr_Fr	Date de livraison du Fournisseur
38	Délai_Livr_ Fr	Délai de livraison du Fournisseur
39	Qte_Achat	La quantité d'achat
40	DescriptionAchat	La description de commande achat

41	Num_Fact_Fr	Le numéro de facture du fournisseur
42	Qte_Fact_Fr	La quantité de facture du fournisseur
43	prix_Fact_Fr	Le prix de facture du fournisseur
44	Date_Fact_Fr	La date de facture du fournisseur
45	Date_Echéance_Fr	La date d'échéance du fournisseur
46	DescriptionFr	La description de facture du fournisseur
47	Id_Pdt	L'identité de Produit
48	Nom_Pdt	Le nom de produit
49	Prix_Unitaire_Vente	Le prix unitaire de vente
50	Prix_Unitaire_Achat	Le prix unitaire de d'achat
51	Qte_Disponible	La quantité disponible de produit
52	Qte_Min	La quantité minimale de produit
53	État	État de produit
52	DescriptionPdt	La description de produit
53	Id_ctg	L'identité de catégorie
54	Nom_ctg	Le nom de catégorie
55	Description	La description de catégorie

II.2. Représentation des classes :

Après la création du dictionnaire des données, nous avons identifié la liste des classes.

➤ *Classe Clients :*

Tableau 9:Classe Client

Attributs	
Nom	Type
Id_clt	Int
Nom_clt	Varchar
Prénom_clt	Varchar
Adresse_clt	Varchar
Téléphone_clt	Varchar
Email_clt	Varchar
Ville_clt	Varchar
Région_clt	Varchar
Code_Postal_clt	Varchar
Pays_clt	Varchar
Méthodes	
Nom	
Ajouter ()	
Modifier ()	
Supprimer ()	
ImprimerPDF ()	

➤ *Classe Commande_Client :*

Tableau 10:Commande Client

Attributs	
Nom	Type
Id_Cmd	Int
Date_Cmd	Date
État_Cmd	Varchar
Date_Livr_clt	Date
Délai_Livr_Clt	Date
Méthodes	
Nom	
Ajouter ()	
Modifier ()	
Supprimer ()	
Imprimer PDF ()	

➤ *Classe Ligne_De_Commande :*

Tableau 11: Classe Ligne De Commande

Attributs	
Nom	Type
Qte_Cmd	Int
DescriptionCmd	Varchar

➤ *Classe Facture_Client :*

Tableau 12:Classe Facture Client

Attributs	
Nom	Type
Num_Fact_Clt	Int
Qte_Fact_Clt	Int
prix_Fact_Clt	Varchar
Date_Fact_Clt	Date
Date_Echéance	Date
DescriptionClt	Varchar
Méthodes	
Nom	
Ajouter ()	
Modifier ()	
Supprimer ()	
ImprimerPDF ()	

➤ *Classe Fournisseurs :*

Tableau 13 : Classe Fournisseurs

Attributs	
Nom	Type
Id_Fr	Int
Nom_Fr	Varchar
Adresse_Fr	Varchar
Téléphone_Fr	Varchar
Fax_Fr	Varchar
Email_Fr	Varchar
Ville_Fr	Varchar
Région_Fr	Varchar
Code_Postal_Fr	Varchar
Pays_Fr	Varchar
Méthodes	
Nom	
Ajouter ()	
Modifier ()	
Supprimer ()	
ImprimerPDF ()	

➤ *Classe Commande_Achat :*

Tableau 14:Classe Commande Achat

Attributs	
Nom	Type
Id_Cmd_Achat	Int
Date_ Achat	Date
État_ Achat	Varchar
Date_Livr_Fr	Date
Délai_Livr_ Fr	Date
Méthodes	
Nom	
Ajouter ()	
Modifier ()	
Supprimer ()	
Imprimer PDF ()	

➤ *Classe Ligne_De_Achat :*

Tableau 15 :Classe Ligne D'Achat

Attributs	
Nom	Type
Qte_Achat	Int
DescriptionAchat	Varchar

➤ *Classe Facture_Fournisseur :*

Tableau 16:Classe Facture Fournisseur

Attributs	
Nom	Type
Num_Fact_Fr	Int
Qte_Fact_Fr	Varchar
prix_Fact_Fr	Varchar
Date_Fact_Fr	Date
Date_Echéance_Fr	Date
DescriptionFr	Varchar
Méthodes	
Nom	
Ajouter ()	
Modifier ()	
Supprimer ()	
ImprimerPDF ()	

➤ **Classe Produit :**

Tableau 17:Classe Produit

Attributs	
Nom	Type
Id_Pdt	Int
Nom_Pdt	Varchar
Prix_Unitaire_Vente	Varchar
Prix_Unitaire_Achat	Varchar
Qte_Disponible	Varchar
Qte_Min	Varchar
État	Varchar
DescriptionPdt	Varchar
Méthodes	
Nom	
Consulter ()	
Ajouter ()	
Supprimer ()	
Imprimer PDF ()	

➤ *Classe Catégories :*

Tableau 18: Classe Catégories

Attributs	
Nom	Type
Id _ctg	Int
Nom_ctg	Varchar
Description	Varchar
Méthodes	
Nom	
Ajouter ()	
Modifier ()	
Supprimer ()	
Imprimer PDF ()	

II.3. Diagramme de classes :

Un diagramme de classes a toujours été le diagramme le plus important dans toutes les méthodes orientées objet.

Un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant le système des classes, leurs attributs, les opérations (ou) les méthodes et les relations entre les classes.[7]

La figure 9 présente le diagramme de classe de notre système :

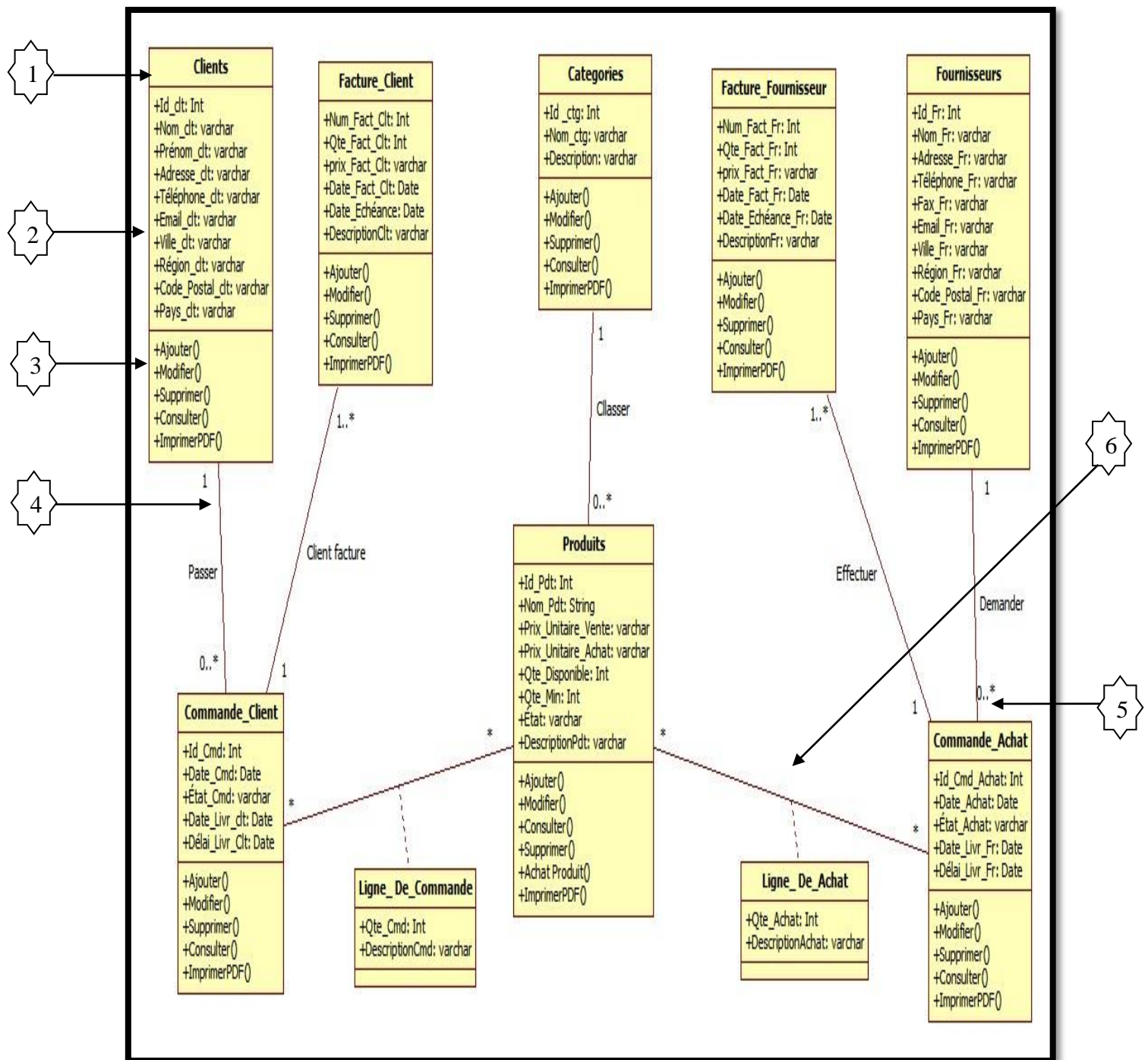


Figure 25:Diagramme de classes

1 : Classe **2 : Attribut** **3 : Opération** **4 : Association** **5 : Cardinalité**
6 : Association porteuse de donnée

III. La base de données :

III.1. Modèle logique des données :

III.1.1. Définition de MLD :

La description conceptuelle par le model objet a permis de représenter le plus fidèlement possible les réalités de l'univers à informatiser. Mais cette représentation ne peut pas être directement manipulée et acceptée par un système informatique. Il est donc nécessaire de passer

D'un modèle objet à un modèle plus proche des capacités des systèmes informatiques
Ce modèle est appelé modèle logique

Le modèle logique permet de présenter les données sous forme de structures basées sur la théorie des relations elle-même construite à partir de la théorie des ensembles. Il fait aujourd'hui autorité dans l'industrie du moment puisqu'il constitue le bas de nombreux systèmes et que les architectures permettant d'accéder depuis une station de travail à des serveurs de données s'appuient en général sur lui.

Pour la conversion du modèle objet (diagramme de classe) en modèle logique (modèle relationnel) il faut appliquer certaines règles de transformation. [9]

III.1.2. Règles de transformations

❖ Règle 1 :

✓ Toute entité devient une relation :

Par définition une classe est traduite en une relation. L'identifiant de la classe devient la clé primaire de la relation. [9]

❖ Règle 2 :

✓ Cardinalité (1..N), (1..1) ou (1..1), (0..1):

Une telle association entraîne l'intégration de l'identifiant de l'entité but de la dépendance fonctionnelle dans la relation associée à l'entité source de la dépendance fonctionnelle. La propriété ainsi dupliquée devient clé étrangère dans la relation source et est marquée par un #.

Une clé étrangère dans une relation est une propriété qui est clé primaire dans une autre relation. [9]

❖ Règle 3 :

✓ Cardinalité (1..N), (1..N):

Toute association se transformera en relation qui hérite des identifiants des entités participants à la relation. [9]

❖ Règle 4 :

✓ Cardinalité (M..N):

Deviens une relation qui hérite des identifiants des entités participants à la relation. Si l'association est porteuse, la relation sera complétée par la liste des propriétés portées. [9]

III.2. Modèle physique de données

✚ **Clients**(Id_{clt},#Id_Cmd,Nom_clt,Prénom_clt,Adresse_clt,Téléphone_clt,Email_clt,Ville_clt,Région_clt,Code_Postal_clt,Pays_clt)

✚ **Commande_Client**(Id_Cmd,#Num_Fact_Clt,Date_Cmd,État_Cmd,Date_Livr_clt,Délai_Livr_Clt)

✚ **Ligne_De_Commande**(#Id_Cmd, #Id_Pdt, Qte_Cmd, DescriptionCmd)

✚ **Facture_Client**(Num_Fact_Clt,Qte_Fact_Clt,prix_Fact_Clt,Date_Fact_Clt,Date_Echéance, DescriptionClt)

✚ **Fournisseurs**(Id_Fr,#Id_Cmd_Achat,Nom_Fr,Adresse_Fr,Téléphone_Fr,Fax_Fr,Email_Fr,Ville_Fr,Région_Fr,Code_Postal_Fr,Pays_Fr)

✚ **Commande_Achat**(Id_Cmd_Achat,#Num_Fact_Fr,Date_Achat,État_Achat,Date_Livr_Fr,Délai_Livr_Fr)

✚ **Ligne_De_Achat** (#Id_Cmd_Achat, #Id_Pd, Qte_Achat,DescriptionAchat)

✚ **Facture_Fournisseur**(Num_Fact_Fr,Qte_Fact_Fr,prix_Fact_Fr,Date_Fact_Fr,Date_Echéance_Fr,DescriptionFr)

✚ **Produit**(Id_Pdt,Nom_Pdt,Prix_Unitaire_Vente,Prix_Unitaire_Achat,État,,Qte_Disponible,Qte_Min, DescriptionPdt)

✚ **Catégories** (Id_ctg,#Id_Pd,Nom_ctg,Description)

Conclusion

Dans ce chapitre, nous avons fait une conception détaillée de notre futur système et nous avons étudié les scénarios possibles et les différents diagrammes (Séquences, Activité et classe) ainsi que notre base de données.

Dans le dernier chapitre nous allons essayer de suivre cette conception dans le but de mettre en place le système souhaité.

Chapitre IV: Réalisation de l'application

Introduction

Ce chapitre représente le dernier volet de ce rapport. Elle a pour objet d'exposer le travail finalisé. Pour le faire, nous allons commencer par la description de l'environnement matériel, logiciel et les techniques utilisées pour notre application. Ensuite nous avons illustré quelques aperçus sur le travail accompli au cours de la période de développement.

I. Environnement de travail :

Dans cette parties nous allons commencer par la description de l'environnement matériel et logiciel de notre application ainsi que les raisons pour lesquelles nous avons opté pour cet environnement.

I.1. Environnement matériel :

Afin de réaliser cette application, il a été mis à notre disposition un ordinateur portable dont les caractéristiques sont détaillées dans ce qui suit:

Tableau 19: Configuration Matérielle

Marque	Dell
Système d'exploitation	Microsoft Windows 7 Édition Intégrale
Processus	Intel® Pentium® 3805U @ 1,90GHz 1,90 GHz
Mémoire installée (RAM)	4,00 Go
Type du système	Système d'exploitation 64 bits
Disque dur	500G

I.2. Environnement logiciel :

Tout le long de la phase de développement, nous nous sommes servis de l'environnement logiciel suivant :

- ❖ **Système d'exploitation:** Le système d'exploitation utilisé est : « Microsoft Windows 7 Edition Intégral Service pack 1 ».
- ❖ **Outil Planification :**

- Microsoft Visual Studio :

Microsoft Visual Studio est une suite de logiciels de développement pour Windows et mac OS conçue par Microsoft. La dernière version s'appelle Visual Studio 2017.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Développeur. [10]



Figure 26: Microsoft Visual Studio

o Visual C# 2010 :

Visual C# est un outil de développement édité par Microsoft, permettant de concevoir des applications articulées autour du langage C#. Visual C# propose les outils pour développer des applications C# qui ciblent la plateforme nouvelle génération de Microsoft pour la programmation distribuée et compatible Internet. [11]



Figure 27: Visual C#

- **MySQL Workbench (anciennement MySQL administrator) :**

MySQL Workbench est un logiciel de gestion et d'administration de données MySQL créé en 2004. Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL. [12]



Figure 28: MySQL Workbench

❖ **Langage de modélisation UML:**

- **StarUML :**

C'est un outil de conception orienté objet UML qui offre un environnement de modélisation pour la représentation de différents diagrammes UML d'une manière simple et facile, tels que la création de diagramme de classes, d'objets, d'activité ou bien de séquences. [13]



Figure 29: StarUML

❖ **Microsoft Word 2013:** a été utilisé pour la rédaction de notre rapport.

II. Choix langages et techniques :

C# « Langage de programmation » :



C# est un langage orienté objet de type sécurisé et élégant qui permet aux développeurs de générer diverses applications sécurisées et fiables qui s'exécutent sur le .NET Framework. Vous pouvez utiliser le langage C# pour créer entre autres des applications clientes Windows, des services Web XML, des composants distribués, des applications client-serveur et des applications de base de données. Visual C# fournit un éditeur de code avancé, des concepteurs d'interfaces utilisateur pratiques, un débogueur intégré et de nombreux autres outils pour faciliter le développement d'applications basées sur le langage C# et .NET Framework. [14]



SQL : « Langage de base de données » :

SQL (sigle de Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles. [15]

III. Présentation des interfaces de l'application :

Notre application est constituée d'un grand nombre d'interfaces et dans ce qui suit nous allons présenter les principaux écrans en décrivant leur fonctionnement :

III.1. Page d'authentification :

Cette interface présente la page d'accueil de notre site web. L'accès aux différents services offerts est établi selon la nature de l'utilisateur : Administrateur ou Employés selon leur activité.



Figure 30: Page d'authentification

III.2. Interface d'accueil :

Elle présente les fonctionnalités de base de notre application: Gestion des clients, gestion des matériels, gestion des employés, gestion des groupes, gestion des fournisseurs et gestion des commandes.



Figure 31: Interface d'accueils employés



Figure 32: Interface d'accueil Administrateur

III.3. Partie back office :

III.3.1. Interfaces gestion d'employés :

C'est l'interface qui permet d'ajouter, chercher, modifier ou supprimer un utilisateur. Comme elle représente aussi la liste des employés en listant les différentes informations.

ID	Nom	Fonction	Type de Compte	Adresse
10	Khouloud	Agent	Agent des Ventes	Tozeur Centre
11	Karia	Administrateur	X	Tunisie

Figure 33: interface gestion d'employés

III.3.2. Interfaces gestion des groupes du travail :

Cette interface permet à l'administrateur de créer, modifier, supprimer ou chercher un groupe du travail selon un service données par exemple la création d'un groupe en vérifiant la saisie d'administrateur pour assurer la gestion de ventes des matériels informatiques comme elle montre la figure suivantes :

The screenshot shows a window titled 'G_Groupes' with two main sections: 'Formulaire' (Form) and 'Outils' (Tools).

Formulaire:

- ID :** A text input field containing '10'.
- Nom :** A text input field containing 'Khouloud'.
- Type de Compte :** A dropdown menu with 'Agent des Ventes' selected.
- Radio buttons:** Three options are listed: 'Groupe [Agent des Ventes]' (selected), 'Groupe [Agent des Achats]', and 'Groupe [Agent du Stocks]'.
- Icon:** A small icon of a pencil and eraser.

Outils:

- Search:** A search bar with a magnifying glass icon.
- Table:** A table with 6 columns: ID, Nom, Fonction, Type de Compte, Adresse, and a blank column. The first row is highlighted in green.

	ID	Nom	Fonction	Type de Compte	Adresse	
▶	10	Khouloud	Agent	Agent des Ventes	Tozeur Centre	

Figure 34: interface gestion des groupes du travail

III.4. Partie front office :

III.4.1. Interfaces Modules gestion des ventes :

Au niveau de cette partie nous devons montrer les interfaces nécessaires pour assurer la gestion du module de vente: gestion des clients, gestion commandes et gestion factures.

❖ Interface Client :

C'est l'interface qui permet à l'agent des ventes de gérer les informations d'un tel client

Tel que : ajouter, modifier et supprimer un client et aussi permet d'imprimer la fiche de "client".

La figure N° 35 montre la gestion des clients d'une société.

The screenshot shows the 'G_Ventes' application window. It has three tabs: 'Clients', 'Commandes', and 'Factures'. The 'Clients' tab is active. Below the tabs is a 'Formulaire' section with input fields for client details: Nom (Chaima), Prénom (Karia), Adresse (Tozeur), Telephone (28523147), Email (chaimakaria@live.fr), Ville (Tozeur), Région (Tozeur), Code Postal (2000), and Pays (Tunis). There are 'Ajouter' and 'Effacer' buttons below the form. Below the form is an 'Outils' section with a search bar and icons for search, print, and delete. At the bottom is a table of clients.

ID	Nom	Prénom	Email	Adresse	Ville	Telephon
4	1	2	5	3	555	4
5	Chaima	Karia	maj@live.fr	1	1	1

Figure 35: Interface gestion des Clients

➤ Interface Supprimer Client:

Pour supprimer un tel client nous allons en premier lieu sélectionner ce dernier et cliquer ensuite sur le bouton supprimer. Un message de confirmation s'affiche avec le message «Tu veux vraiment le supprimer ».

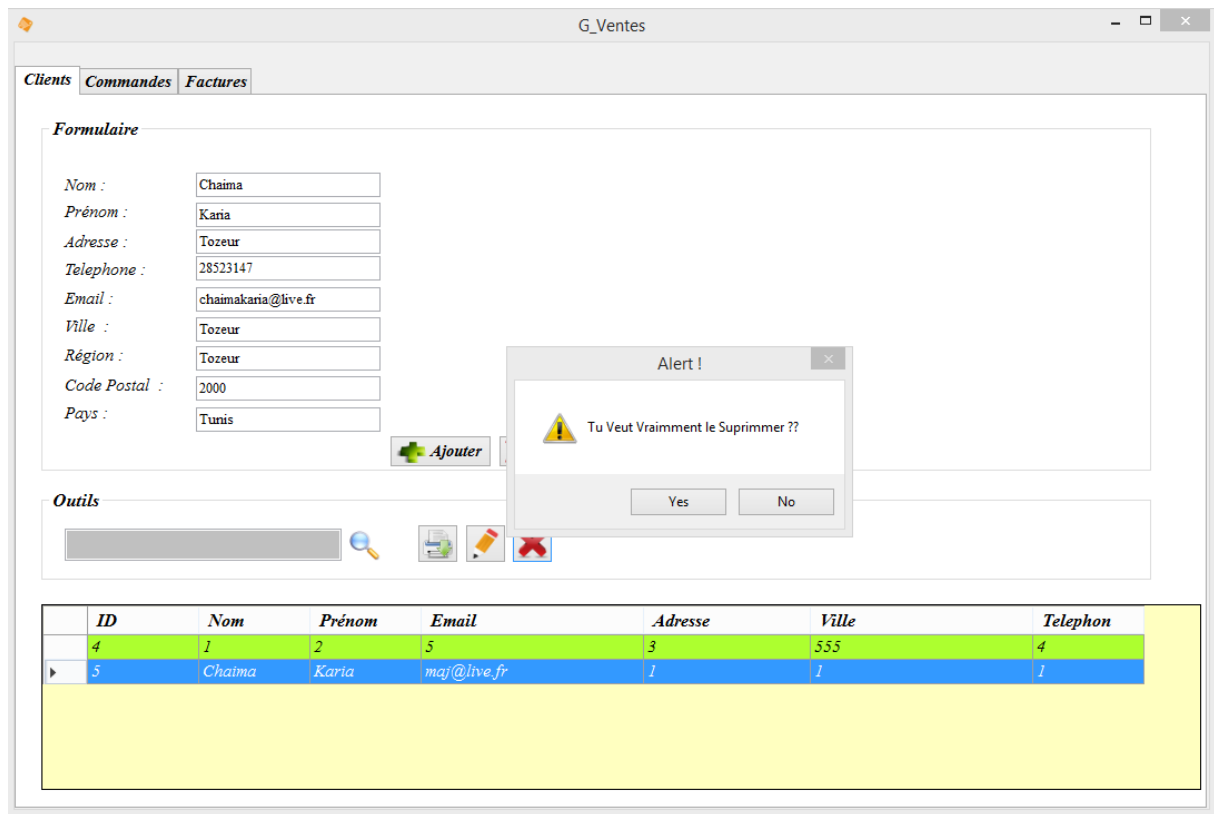


Figure 36: Interface Supprimer client

➤ Interface ajouter client

Lors de l'ajout d'un client Il est à noter que tous les champs doivent être remplis

Un message de confirmation s'affiche avec le message «Ajouter avec succès ».

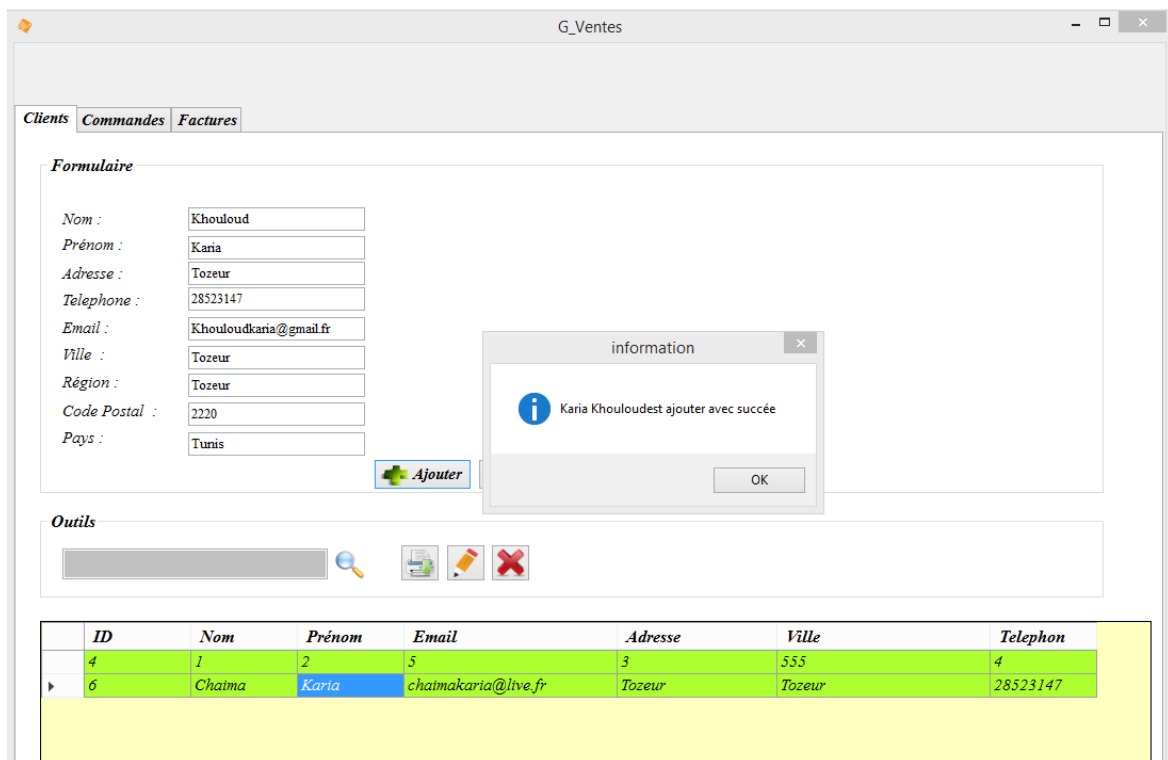


Figure 37: Interface test ajouté de client

❖ Interface Commande client

C'est l'interface qui permet à l'agent des ventes d'ajouter, modifier et supprimer un commande client et aussi permet d'imprimer la fiche de "commande client".

	ID Commande	Num Fact	Date Commande	Etat Commande	Date Livraison	Delai Livraison
	16	2	111	En Stock	1	1
▶	17	1	26/06/2018	En Stock	01/07/2018	30/07/2018

Figure 38:Interface gestion des Commandes clients

❖ Interface facture client

C'est l'interface qui permet à l'agent des ventes d'ajouter, modifier et supprimer une facture client et aussi permet d'imprimer la fiche "facture client".

➤ Interface imprimer la fiche "facture client"

Pour imprimer la fiche "facture client " nous allons en premier lieu sélectionner ce dernier et cliquer ensuite sur le bouton imprimé.

Formulaire

Quantité Facture Client : 2

Prix Facture Client : 20dt

Date Facture Client : 26/06/2018

Date Echéance : 01/01/2022

Description Facture Client : Non payer

Outils

N°	Quantité	Prix	Date Facture	Date Echéance	Description
1	1	1	1	1	1
2	2	1	1032	1	1
1	1	1	1	1	1
2	2	1	1032	1	1
3	2	20dt	26/06/2018	01/01/2022	01/01/2022

Figure 39:Interface imprimer la fiche facture client

Document PDF à l'aide d'une dll qui s'appelle " iTextsharp" qui nous génère un fichier PDF d'une manière plus professionnelle

le 27/06/2018
à 00:08

Fiche de Facture Client Num° 003

Les Information du Facture Client :

Quantité Facture Client :	2
Prix Facture Client :	20dt
Date Facture Client :	26/06/2018
Date Echéance :	01/01/2022
Description Facture Client :	Non payer

Figure 40: document PDF "facture client"

III.4.2. Interfaces Modules gestion des stocks :

La partie gestion du stock se présente sur deux parties : gestion des produits et la gestion de son catégorie.

❖ Interface Produit et Catégorie :

La présente figure montre l'interface qui permet à un agent du stocks de modifier, supprimer, ajouter, consulter ou bien chercher le produit et leur catégories il peut aussi permet d'imprimer la fiche de produit et catégorie.

Produits **Catégories**

Formulaire

Nom Produit : Clé 3G Qte Min Produit : 10

Prix Unitaire vente : 15 DT

Prix Unitaire Achat : 17 DT

Description Produit : Clé LTE - A

Etat Produit : **En Stock** ▼

Qte Dispo. Produit : 100

Outils

	<i>ID Produit</i>	<i>Nom Produit</i>	<i>Prix Unitaire Vente</i>	<i>Prix Unitaire Achat</i>
	1	Tablette	1	1
▶	2	Clé 3G	15 DT	17 DT
	3	Chargeur HTC	4 DT	7 DT

Figure 41: Interface gestion des Produits

Produits **Catégories**

Formulaire

Produit : 1 ▼

Nom Catégories : HP

Description Catégories : PC

Outils

	<i>ID Catégories</i>	<i>ID Produit</i>	<i>Nom Catégories</i>	<i>Description Catégories</i>
	2	3	Des CD	2256
	3	2	Des SFF	dsjfglk
▶	4	1	HP	PC

Figure 42: Interface gestion des Catégories

III.4.3. Interfaces Modules gestion des achats :

Dans cette partie nous devons présenter les différentes interfaces du module d'achats.

Cette interface permet à l'agent d'achat d'ajouter des informations, de modifier les informations, de chercher les informations et supprimer les informations: fournisseur et commandes d'achat et factures d'achat.

❖ Interface gestion des fournisseurs :

La figure N° 43 montre la gestion des fournisseurs d'une société.

The screenshot shows a window titled 'G_Achat' with three tabs: 'Fournisseurs', 'Commandes', and 'Factures'. The 'Fournisseurs' tab is active, displaying a 'Formulaire' (Form) section with the following fields and values:

- Nom : Khouloud
- Adresse : sousse
- Telephone : 23123654
- Fax : 76253589
- Email : Khouloudk@live.fr
- Ville : sousse
- Région : sousse
- Code Postal : 2220
- Pays : tunis

Below the form are two buttons: 'Ajouter' (with a green plus icon) and 'Effacer' (with a red X icon). Under the 'Outils' (Tools) section, there is a search bar and icons for search, print, edit, and delete. At the bottom, a table lists the suppliers:

	ID	Nom	Adresse	Telephon	Fax	Email	Ville
▶	3	Khouloud	sousse	23123654	76253589	Khouloudk...	sousse
	5	Khouloud	adresse_fr	telephone_fr	fax_fr	email_fr	ville_fr

Figure 43: Interface gestion des fournisseurs

➤ Interface Modifier fournisseur :

Pour modifier un tel fournisseur nous allons en premier lieu sélectionner la liste de fournisseur dans la table et cliquer ensuite sur le bouton Modifier. La information de fournisseur importer

automatiquement dans le champ puis en faire la modification par suit on clique sur le bouton Modifier et finalement nous avons obtenir le mise a jour dans le table fournisseur.

Formulaire

Nom : Khouloud

Adresse : sousse

Telephone : 23123654

Fax : 76253589

Email : Khouloudk@live.fr

Ville : sousse

Région : sousse

Code Postal : 2220

Pays : tunis

Ajouter Effacer

Information

Validé

OK

Outils

ID	Nom	Adresse	Telephon	Fax	Email	Ville
3	Chaima	Mosrtan	5455555	6856565	karia@live.fr	tozeur
5	Khouloud	adresse_fr	telephone_fr	fax_fr	email_fr	ville_fr

Figure 44: Interface Modifier fournisseur

❖ Interface gestion des Commandes fournisseur :

C'est l'interface qui permet à l'agent d'achat d'ajouter, modifier et supprimer un commande fournisseur et aussi permet d'imprimer la fiche de "commande fournisseur".

G_Achat

Fournisseurs **Commandes** **Factures**

Formulaire

N Fact Fournisseur : 2

Date Commande Achat : 24/11/2017

Etat Commande Achat : Disponible

Date Livraison Fournisseur : 01/02/2018

Delai Livraison Fournisseur : 30/02/2018

Produit : 1

Qte Produit : 10

Description Ligne : payer

Ajouter **Effacer**

Outils

ID Cmd	Num Fact	Date	Etat	Date Livr	Delai Livr
7	2	1	En Stock	1	1

Figure 45:Interface gestion des Commande fournisseur

❖ Interface gestion des factures fournisseur :

C'est l'interface qui permet à l'agent d'achat d'ajouter, modifier et supprimer une facture fournisseur et aussi permet d'imprimer la fiche "facture fournisseur".

G_Achat

Fournisseurs **Commandes** **Factures**

Formulaire

Quantité Facture Fournisseur : 3

Prix Facture Fournisseur : 40dt

Date Facture Fournisseur : 01/01/2018

Date Echéance Fournisseur : 30/12/2022

Description Facture Fournisseur : payer

Outils

	N°	Quantité	Prix	Date Facture	Date Echéance
	2	1	1	1	1
▶	3	2	40dt	01/01/2018	30/12/2022
	4	2	40dt	01/01/2018	30/12/2022
	5	2	40dt	01/01/2018	30/12/2022
	6	2	40dt	01/01/2018	30/12/2022

Figure 46:Interface gestion des factures fournisseur

Conclusion :

Ce chapitre présenté les détails du déroulement du travail. Nous avons essayé de mettre en évidence, chaque fois, les outils matériels et logiciels les plus importants utilisés dans notre application .La dernière partie a été consacrée, en totalité, pour la visualisation des principales interfaces de travail qui permettent de donner une idée globale sur notre site réalisé.

Conclusion générale

Ce projet de fin d'étude consiste à réaliser une application gestion commerciale des produits informatiques.

Tout au long de ce présent travail, nous avons décrit les étapes de développement de notre application, ainsi que les outils utilisés au cours de ce processus. Il se résume en conception et la réalisation d'une application web. Dans ce contexte, nous avons cherché à développer une application flexible et évolutive permettant son amélioration d'anticiper les changements continus des besoins des utilisateurs.

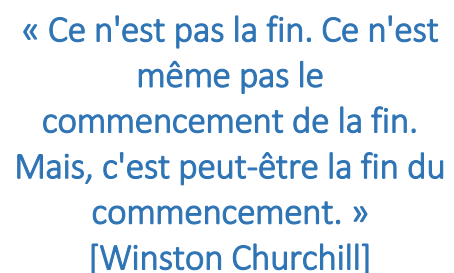
Ce projet nous offre une bonne occasion pour exploiter, pratiquer et approfondir nos connaissances acquises durant les années d'études universitaires.

Il nous a donné l'occasion de profiter d'une bonne expérience professionnelle et d'approfondir nos connaissances de point de vue conception avec UML et développement avec C#.

L'application peut être considérée comme une solution efficace pour dépasser les défauts de la méthode classique de la gestion manuelle des informations.

Nous souhaitons bien que notre produit procure les satisfactions de la Société et qu'il soit dans la mesure de leur souhait.

Cette dernière peut ne pas s'arrêter à ce stade ; c'est une première version dont on peut l'améliorer dans des futurs travaux.



« Ce n'est pas la fin. Ce n'est même pas le commencement de la fin. Mais, c'est peut-être la fin du commencement. »
[Winston Churchill]

Bibliographie

- [7] : Cours conception LFSI2. 2015
- [8] : Cours conception LFSI2. 2016
- [9] : Cours base de données LFSI2. 2015

Webographie

- [1] : <http://assistanceplus.com.tn/vos-besoins/gestion-commerciale/>
- [2] : <https://www.maxicours.com/soutien-scolaire/information-et-gestion/1re-stg/205328.html>
- [3] : <https://laurent-audibert.developpez.com/Cours-UML/?page=introduction-modelisation-objet#L1-2>
- [4] : http://www.uml-sysml.org/images-du-site/uml/cycle-vie-cascade.png/view?set_language=en
- [5] : [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))
- [6] : https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation
- [10]: https://fr.wikipedia.org/wiki/Microsoft_Visual_Studio
- [11]: https://fr.wikipedia.org/wiki/Microsoft_Visual_Studio
- [12] : https://fr.wikipedia.org/wiki/MySQL_Workbench
- [13] : <https://fr.wikipedia.org/wiki/StarUML>
- [14]: [https://msdn.microsoft.com/fr-fr/library/windows/desktop/z1zx9t92\(v=vs.120\).aspx](https://msdn.microsoft.com/fr-fr/library/windows/desktop/z1zx9t92(v=vs.120).aspx)
- [15] : https://fr.wikipedia.org/wiki/Structured_Query_Language

Conception et développement d'une application gestion commerciale des produits informatiques

Karia Chaima&&Karia Khouloud

الخلاصة:

يندرج هذا العمل في إطار مشروع نهاية الدراسة في كلية العلوم بقفصه. وهو يشمل تصميم وتطوير تطبيق الإدارة التجارية لمنتجات تكنولوجيا المعلومات.

ويصف هذا التقرير مختلف مراحل تطوير هذا تطبيق، من العالم الحقيقي إلى التحقيق النهائي للتطبيق المصمم.

المفاتيح: بوابة، تطبيق، الكترونية

Résumé :

Ce travail a été effectué dans le cadre d'un projet de fin d'année réalisé au Faculté des Sciences de Gafsa.

Il englobe la conception et développement d'une application gestion commerciale des produits informatiques. Le présent rapport décrit les différentes étapes de développement de cette application, allant de monde réel à la réalisation finale de l'application conçue.

Mots clés: Conception, Configuration, développement, réalisation, application

Abstract:

This work has been done in the setting of a project of year end achieved to the FSGF.

It covers the conception and development a computers store management application

The present report describes the different stages of development of this site, active from real world to the final realization of the application concue.

Keywords : Conception, Configuration, development, application.