

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

Université A/Mira de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique



Mémoire de Master en Informatique

Option

Génie Logiciel

Thème

**Développement d'une application pour la gestion de stock
de NAFTAL district GPL Bejaia**

Présenté par :

M^{elle} **Soltana** BOUROUINA

M^{elle} **Sarra** BENIMEUR

Devant le jury composé de :

Président : *M^r* Ali LARBI, MAA, université de Bejaia.

Examineur 1 : *M^r* Abdellah AKILAL, MAB, université de Bejaia.

Examineur 2 : *M^r* Khaled BEDJOU, Magister, CERIST, Bejaia.

Encadreur : *M^r* Ali HASSAOUI, Magister, université de Bejaia.

Co-encadreur : *M^r* Mouloud ATMANI, MAB, université de Bejaia.

Promotion 2015/2016

Résumé

L'informatisation des documents de toutes entreprises est devenue indispensable, son objectif principal est de régler les consensus qui étaient inévitable au par avant. Dans le même objectif décrit précédemment, on a proposé une solution pour la gestion du magasin de NAFTAL. Afin de réaliser notre solution d'automatisation de la gestion des stocks du district GPL Bejaia, on s'est basé sur le modèle de conception UML. Notre application est réalisée en utilisant le langage de programmation java en interaction avec une base de données MYSQL et qui a comme objectifs : l'informatisation des entrées sorties du stock, génération automatique des fiches de mouvement de stock, des fiches inventaires et l'implémentation des alertes automatiques des stocks minimum, date péremption des produits et autres.

Mots clés : gestion des stocks, mouvement de stock, inventaire, UML.

Abstract

The computerization of documents of all companies has become indispensable, its main objective is to resolve consensus which were previously inevitable. In the same objective described above, proposed a solution for the management of NAFTAL store. To achieve our automation solution management stocks of GPL district Bejaia, we based on the UML design model. Our application is performed using the Java programming language in interaction with MySQL database and has the following objectives : the computerization of inputs from stock, automatic generation of stock cards movement, inventory records and implementation automatic alerts of minimum stocks, the date of expiration and other products.

Key words : inventory management, inventory movement, inventory, UML.

REMERCIEMENT

Si ce travail a pu voir le jour, c'est certainement grâce à Dieu, au soutien et à l'aide de plusieurs personnes. Nous profitons de cet espace pour les remercier tous.

Au terme de ce travail, on tient à exprimer notre profonde gratitude et nos sincères remerciements à notre encadrant et Co-encadrant M^r Ali HASSAOUI et M^r Mouloud ATMANI pour tout le temps et l'investissement qu'ils nous ont consacré, leurs directives précieuses, et pour la qualité de leurs suivi.

Nous remercions énormément M^r Ali LARBI de nous avoir fait l'honneur de présider notre soutenance. Nous remercions également M^r Abdellah AKILAL et M^r Khaled BEDJOU de nous avoir fait l'honneur en acceptant d'examiner notre travail et faire partie de notre jury.

Nous remercions énormément les membres du personnel du district GPL qui nous ont facilité toutes les procédures de recueil d'information et nous ont aidé à comprendre le système de gestion actuel de l'entreprise.

Nos remerciements seraient sans doute incomplets si nous ne citons pas :

Nos parents, sans qui ce travail n'aurait jamais vu le jour, nous leur sommes infiniment reconnaissant pour leur patience et leur soutien moral, ainsi que pour leurs encouragements. Qu'ils trouvent dans ce mémoire le fruit de leur travail! Nous leur présentons ici de manière spéciale toute notre gratitude.

Nos enseignants, nos amis, ainsi que nos camarades.

Enfin, Toute personne qui nous a aidé de près ou de loin à mener à terme ce modeste travail.

Dédicace

Je dédie ce modeste travail

A mes parents.

A toute ma famille.

Et particulièrement a ma grande tante(Hassiba).

A la mémoire de mes grands parents.

A tous mes amis(es).

Enfin, à vous les lecteurs de ce mémoire.

BOUROUINA Soltana

Dédicace

Je dédie ce modeste travail
A la mémoire de ma mère.
A mon père.
A mes frères et soeurs.
A toute ma famille.
A tous mes amis(es).

Enfin, à vous les lecteurs de ce mémoire.

BENIMEUR Sarra

LISTE DES ACRONYMES

A - **B** - **C**

API
ASCII
BDD
CPLB
CDS

Application **P**rogramming **I**nterface
American **S**tandard **C**ode for **I**nformation **I**nterchange
Base **D**e **D**onnées
Carburant **P**neumatique **L**ubrifiant **B**itumes
Centre **D**e **S**tock

G - **H** - **I** - **J**

GPL
HTML
IHM
IDE
J2ME
J2EE
JMF
J3D

Gaz **P**étrole **L**iquéfié
Hyper **T**ext **M**arkup **L**anguage
Interface **H**omme **M**achine
Integrated **D**evelopment **E**nvironment
Java **M**icro **E**dition
Java **E**ntreprise **E**dition
Java **M**edia **F**ramework
Java **3** **D**imension

P - **S** - **U**

PHP
POI
PDF
SPA
SQL
SGBD
UML
UP
XML

Pre processor **H**ypertext
Poor **O**bfuscation **I**mplementation
Portable **D**ocument **F**ormat
Société **P**ar **A**ction
Structures **Q**uery **L**anguage
Système **G**estion **B**ase **D**e donnée
Unified **M**odeling **L**anguage
Unified **P**rocess
eXtensible **M**arkup **L**anguage

TABLE DES MATIÈRES

Table des Matières	i
Table des Figures	v
Introduction générale	1
1 Etat de l'art	3
1.1 Introduction	4
1.2 Présentation de l'entreprise NAFTAL	4
1.2.1 Organisation de NAFTAL	4
1.2.2 Définition du district GPL	6
1.2.3 Structure organisationnelle du district GPL	6
1.2.4 Mission du district GPL de Bejaia	7
1.2.5 Présentation du service informatique	7
1.3 Généralités sur les stocks	8
1.3.1 Définition d'un stock	8
1.3.2 Typologies	9
1.3.3 Gestion des stocks	9
1.3.4 Rôle de la gestion des stocks dans une entreprise	9
1.3.5 Indicateurs de la gestion de stock	9
1.3.6 Avantages et inconvénients de la tenue d'un stock	10
1.3.7 Documents de la gestion de stock	10
1.3.8 Fonctions des stocks	11
1.3.9 Classification des stocks	12

1.4	Présentation du sujet d'étude	13
1.5	Problématique	13
1.6	Objectifs du projet	14
1.7	Conclusion	14
2	Analyse et conception	15
2.1	Introduction	16
2.2	Cycle de vie	16
2.2.1	Modèles de cycle de vie classique	17
2.2.2	Démarche à suivre	20
2.3	Processus de développement UP	20
2.3.1	Vie du processus unifié	21
2.3.2	Avantage d'UP	21
2.4	Présentation d'UML	21
2.4.1	Définition	21
2.4.2	Modélisation UML	21
2.4.3	Justification de la modélisation UML	22
2.5	Spécifications initiales	22
2.5.1	Expression initiale des besoins	23
2.5.2	Identification des acteurs	23
2.5.3	Diagramme de contexte	25
2.6	Diagramme des cas d'utilisation	25
2.6.1	Identification des cas d'utilisation	25
2.6.2	Description des cas d'utilisation	26
2.6.3	Nos diagrammes de cas d'utilisation	30
2.7	Diagramme de séquence	32
2.7.1	Message synchrone et asynchrone	32
2.7.2	Fragment d'interaction	32
2.7.3	Etude de cas d'utilisation	33
2.8	Glossaire des données	36
2.9	Diagramme de classes	37
2.10	Schéma relationnel	38
2.10.1	Règles de passage	38
2.10.2	Modèle logique de données	39
2.11	Conclusion	40

3 Implémentation	41
3.1 Introduction	42
3.2 Langages de programmation utilisés	42
3.2.1 SQL	42
3.2.2 JAVA	42
3.3 Outils de devloppement	43
3.3.1 WampServer	43
3.3.2 PhpMyAdmin	44
3.3.3 Netbeans	44
3.4 Projection de quelques interfaces de notre application	44
3.4.1 Authentification	45
3.4.2 Gestion des utilisateurs	45
3.4.3 Demande de fourniture	46
3.4.4 Interface pour chef magasin	47
3.4.5 Bon d'entrée	48
3.4.6 Bon de sortie	51
3.4.7 Etat du stock	52
3.4.8 Interface du magasinier	53
3.4.9 Fiche mouvements de stock	56
3.5 Conclusion	57
Conclusion générale	58
Bibliographie	59
Annexe	61

TABLE DES FIGURES

1.1	Organigramme général de NAFTAL [site07].	5
1.2	Organigramme du district GPL [site08].	7
1.3	Organigramme du département informatique	8
2.1	Modèle en cascade	17
2.2	Modèle en V	17
2.3	Modèle en spirale	18
2.4	Modèle incrémental	19
2.5	Prototypage	19
2.6	Processus de développment UP	20
2.7	Trois composantes d'une modélisation UML	22
2.8	Diagramme de contexte.	25
2.9	Diagramme de cas d'utilisation associé à l'administrateur	30
2.10	Diagramme de cas d'utilisation associé au magasinier	31
2.11	Diagramme de cas d'utilisation associé au chef magasin	31
2.12	Diagramme de séquence du cas d'utilisation « Authentification ».	33
2.13	Diagramme de séquence du cas d'utilisation « Gérer un utilisateur ».	34
2.14	Diagramme de séquence du cas d'utilisation « Gérer un article ».	35
2.15	Diagramme de classe.	37
3.1	Interface d'authentification.	45
3.2	Interface de gestion des utilisateurs.	46
3.3	Demande de fourniture.	47
3.4	Interface pour chef magasin.	48

3.5	Bon d'entrée.	49
3.6	Nouvelle désignation.	50
3.7	Nouveau fournisseur.	50
3.8	Bon de sortie.	51
3.9	Alerte Stock minimum.	52
3.10	Etat du stock.	52
3.11	Alerte péremption.	53
3.12	Interface du magasinier	54
3.13	Fiche Manuelle.	55
3.14	Désignation des produits	55
3.15	Impression de la fiche manuelle	56
3.16	Fiche mouvements de stock	56

INTRODUCTION GÉNÉRALE

L Informatique est sans aucun doute le domaine qui a révolutionné le monde scientifique, technique et industriel et cela en s'y imposant dans les différents secteurs de travail mais également social. Elle est devenu primordiale pour chaque entreprise et même pour tout individu. L'informatique a pu relever un déficit en mettant en exergue (pratique) tout son panel d'automatisation de l'information à la disposition de tous ces secteurs.

Notre travail s'intéresse particulièrement à la partie logiciel, qui se présente comme un ensemble de programmes et de procédures nécessaires au fonctionnement d'un système informatique. Dans la famille des logiciels, on s'intéresse plus exactement aux applications qui sont spécifiques à la résolution des problèmes de la vie courante.

Le cas qui sera présenté au court de notre travail est la mise en œuvre d'une application de gestion de stock du magasin de NAFTAL (GPL).

Notre organisme d'accueil a décrit plusieurs lacunes dans sa gestion de stock, nous pouvons citer la difficulté dans la recherche d'un produit, liée à la gestion manuelles, la péremption de plusieurs produits consommables et les différentes nominations pour un même produit, etc

Afin de remédier à ces différents problèmes, nous avons eu comme objectif principal de créer une application de gestion des stocks, qui permet : une saisie des bons d'entrées et de sorties des stocks, de fournir une interface conviviale pour la recherche et choix de produits, de fournir plusieurs alertes automatiques de stock minimum et péremption, de générer les différentes fiches de gestion et suivi des stocks comme fiche inventaire, fiche de mouvements de stock, etc

Introduction générale

Ce mémoire est organisé en trois parties :

La première partie : Cette partie a pour objectif de présenter l'organisme d'accueil et de cerner les problèmes aux quels l'organisme est confronté pour bien définir les objectifs de ce travail.

La deuxième partie : sera consacrée à la spécification des besoins fonctionnels et non fonctionnels du projet, l'identification de ses acteurs, ainsi que les interactions entre les acteurs et le système. Nous proposons une solution conceptuelle formée de plusieurs diagrammes UML.

La troisième et la dernière partie : sera consacrée à la présentation des outils et langages utilisés dans l'implémentation ainsi qu'une présentation de l'application développée, avec quelques prises d'écran des différentes interfaces et fiches PDF générées.

Le mémoire finira par une conclusion générale et des perspectives envisagées.

CHAPITRE 1

ETAT DE L'ART

Sommaire

1.1	Introduction	4
1.2	Présentation de l'entreprise NAFTAL	4
1.3	Généralités sur les stocks	8
1.4	Présentation du sujet d'étude	13
1.5	Problématique	13
1.6	Objectifs du projet	14
1.7	Conclusion	14

1.1 Introduction

Dans ce chapitre, nous allons présenter l'organisme d'accueil à savoir le district GPL de Bejaia. Les membres du personnel de ce district nous ont orienté lors de notre stage pratique. Ils nous ont aidé à comprendre le système de gestion actuel de l'entreprise et de prendre connaissance dans le détail, des domaines dont celle-ci souhaite améliorer le fonctionnement. En plus, nous allons aborder dans la section trois des généralités sur les stocks puis la problématique posée, ainsi que les objectifs de notre étude.

1.2 Présentation de l'entreprise NAFTAL

NAFTAL est une société par actions (SPA) au capital social de 15 650 000 000DA. Fondée en 1982 et filiale à 100% du Groupe SONATRACH, elle est rattachée à l'activité de commercialisation. Elle a pour mission principale, la distribution et la commercialisation des produits pétroliers et dérivés sur le marché national.

NAFTAL se subdivise en deux zones bien distincts : zone GPL (gaz de pétrole liquéfié) et zone CPLB (carburant, pneumatique, lubrifiant et bitumes). Elle intervient également dans le domaine du transport des produits pétroliers. A l'ère de la mondialisation, NAFTAL a jugé indispensable la mise en place d'une nouvelle organisation par ligne de produit (bitumes, lubrifiants, réseaux logistique, GPL, pneumatique, Aviation, Marine). La structure centrale de NAFTAL est située à CHERAGA elle est subdivisé en dix-neuf districts GPL.

1.2.1 Organisation de NAFTAL

NAFTAL SPA (société par action) est constituée de comité exécutif et comité directeur, dans chaque comité nous y trouvons des conseillers et des chefs de projets. Nous distinguons au sein de NAFTAL quatre directions exécutives et quatre directions centrales.

L'organisme de NAFTAL est présenté dans la figure 1.1 ci-après :

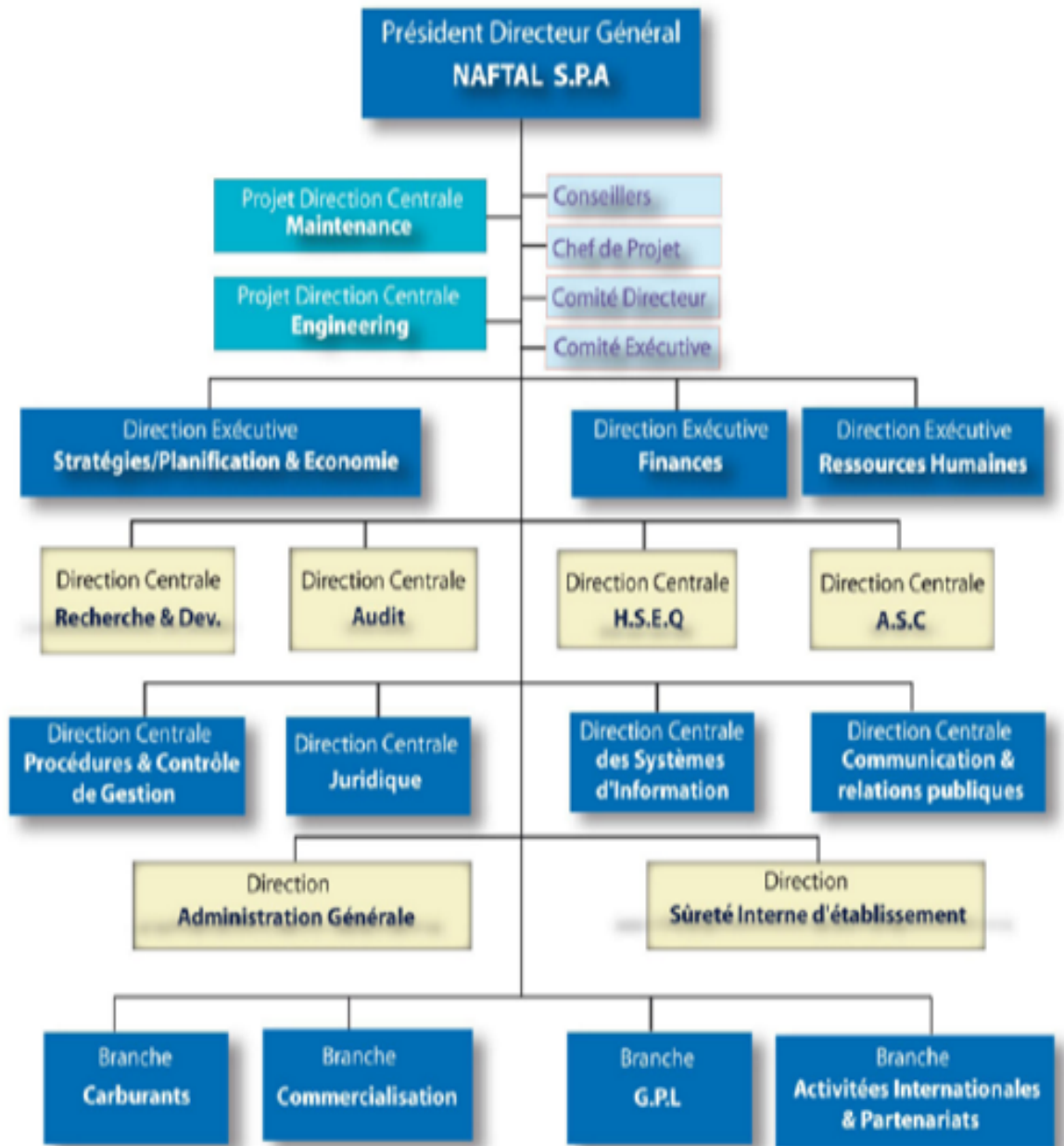


FIGURE 1.1 – Organigramme général de NAFTAL [site07].

1.2.2 Définition du district GPL

La branche GPL de NAFTAL est subdivisée en plusieurs districts GPL dont le district de Bejaia, qui se situe à l'arrière port BP12, cette position est très avantageuse pour faciliter l'approvisionnement direct de la raffinerie vers le port.

1.2.3 Structure organisationnelle du district GPL

Le district compte six (06) départements qui sont gérés par les directions générales, on site :

- Département personnel et moyen communs ;
- Département commercial ;
- Département technique et maintenance ;
- Département exploitation ;
- Département finance et comptabilité ;
- Département informatique.

Nous nous intéressons plus particulièrement au dernier département cité (département informatique), ce dernier se charge d'assurer la coordination de l'activité informatique au niveau de l'entreprise NAFTAL. On trouve aussi au niveau du district les services suivants :

- **Service sureté** : ce service assure la sécurité au sein de l'entreprise NAFTAL.
- **Service sécurité industrielle** : ce service se charge de :
 - La protection et la préservation du personnel ;
 - La préservation et la conservation du personnel ;
 - La protection de l'environnement.
- **Service juriste** : ce service se charge des affaires judiciaires.

L'ensemble de ces services est schématisé par l'organigramme présenté par la figure 1.2 du district GPL (Bejaia).

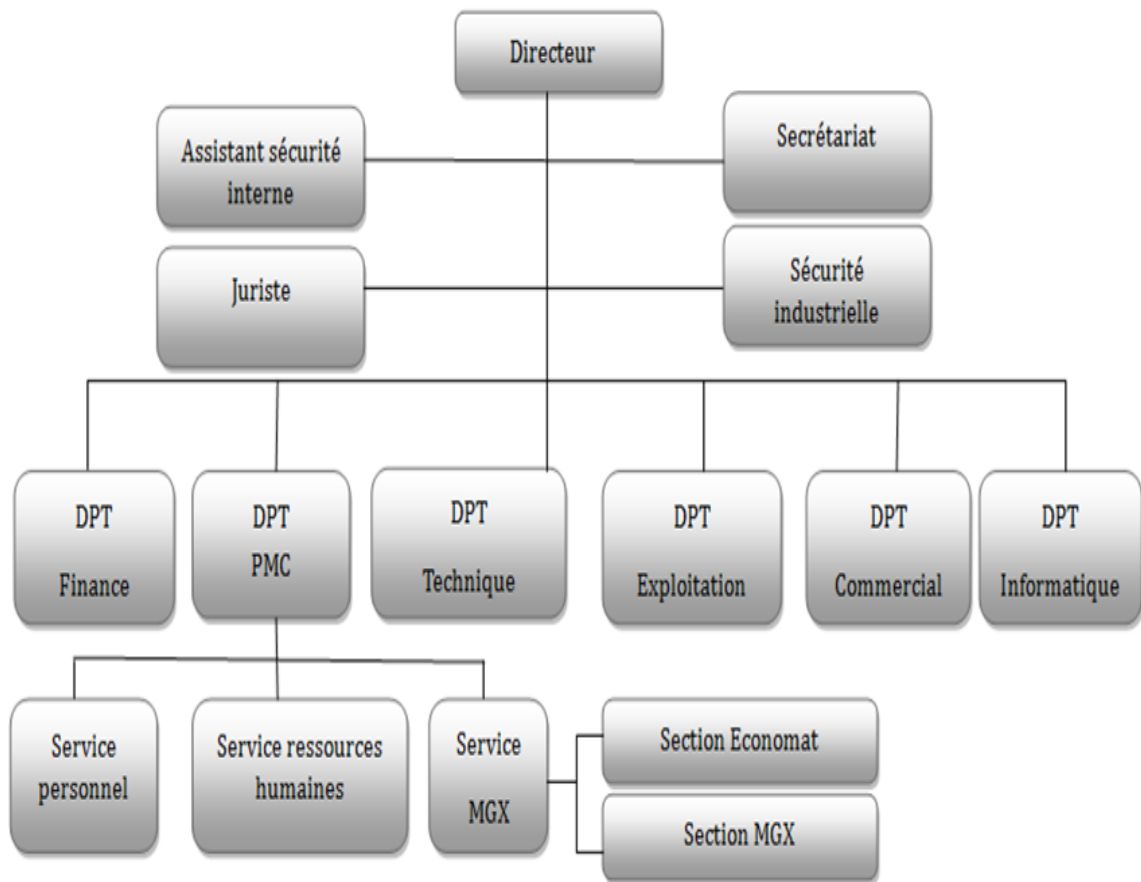


FIGURE 1.2 – Organigramme du district GPL [site08].

1.2.4 Mission du district GPL de Bejaia

L'organisation de NAFTAL est établie pour remplir trois fonctions essentielles :

- Approvisionnement ;
- Stockage des produits ;
- Ventes des produits et prestations de services.

1.2.5 Présentation du service informatique

a) Organigramme du département informatique :

Nous allons à présent illustrer l'organigramme associé à la section informatique illustré par la figure 1.3 afin d'étudier de plus près les deux services qui y contribuent.

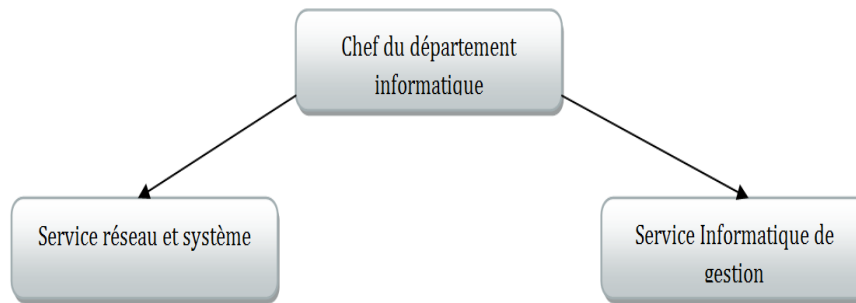


FIGURE 1.3 – Organigramme du département informatique

b) Description et rôle de chaque service au sein du département informatique

Les deux services associés au département informatique ont ce qui suit comme tâches :

- **Service réseau et système assure :**
 - Maintenance de matériels informatique ;
 - Maintenance des logiciels, système et application ;
 - Suivi des différentes activités administrateur réseau.
- **Service informatique de gestion assure :**
 - Recueil de l'information à partir des CDS (centre de stock) ;
 - Analyse les états ;
 - Participe à l'élaboration des plans de production de la zone ;
 - Consolide les plans élaborés par les structures de la zone ;
 - Exécute toute autre tâche, relevant de ces compétences, pouvant lui être confiée.

1.3 Généralités sur les stocks

1.3.1 Définition d'un stock

Il s'agit des biens entrants dans le cycle d'exploitation de l'entreprise pour être vendus en l'état ou après production ou transformation, ou être consommés à la première utilisation. Ces biens doivent appartenir à l'entreprise, et celle-ci doit en être propriétaire au moment de l'inventaire, ce que signifie en particulier que les produits en cours d'acheminement ou reçus doivent être compris dans les stocks, et à l'inverse doivent être exclus les produits qui ont été livrés aux clients. Le stock peut être défini comme l'ensemble des articles détenus par l'entreprise, tout en sachant qu'un article est tout objet manufacturé clairement identifiable dans l'entreprise [1].

1.3.2 Typologies

La qualification de stock pour un bien ne dépend pas de la nature du bien, mais de sa destination : sont considérés comme des stocks les biens destinés à être revendus, et non à être conservés dans l'entreprise pour y être utilisés (immobilisation). Exemple : dans une entreprise d'achat et vente de matériel informatique, les ordinateurs sont des éléments de stocks, sauf ceux qui sont mis en service pour ses propres besoins, qui sont des immobilisations. Les stocks diffèrent selon leurs origines [2] :

- **Stocks matières et fournitures** : matière première, fournitures et emballages.
- **Stocks produits** : produits intermédiaires ou finis.

1.3.3 Gestion des stocks

L'objectif de la gestion des stocks est de réduire les coûts de possession (stockage, gardiennage...) et de passation des commandes, tout en conservant le niveau de stock nécessaire pour éviter toute rupture de stock, pouvant entraîner une perte d'exploitation préjudiciable. Pour cela l'entreprise doit définir des indicateurs précis et contrôler le mieux possible les mouvements des stocks [1].

1.3.4 Rôle de la gestion des stocks dans une entreprise

Une entreprise détient des stocks principalement pour coordonner temporairement ses activités d'achat et de vente. Cette coordination peut-être nécessaire dans un certain nombre de cas [1] :

- Entre le moment de passation d'une commande et sa livraison, s'écoule un temps appelé délai de livraison. En raison de ce délai, un client qui désire acheter un produit ne se trouvant pas en stock ne peut être satisfait immédiatement d'où nécessité d'un certain niveau de stock pour répondre à cette situation.
- Certains produits subissent des fluctuations de leurs prix et l'entreprise peut vouloir profiter de prix bas pour acheter et le prix élevé pour vendre.
- L'offre peut être accentuée à une période donnée de l'année alors que la demande s'étale tout au long de l'année.

1.3.5 Indicateurs de la gestion de stock

Pour une maîtrise de ses stocks, l'entreprise utilise différents indicateurs de gestion des stocks [3] :

- **Stock de sécurité** : c'est la quantité en dessous de laquelle il ne faut pas descendre.
- **Stock d'alerte** : c'est la quantité qui détermine le déclenchement de la commande, elle est calculée en fonction du délai habituel de livraison.
- **Stock minimum** : c'est la quantité correspondant à la consommation pendant le délai de réapprovisionnement, donc $\text{stock minimum} = \text{stock d'alerte} - \text{stock de sécurité}$.

- **Stock maximum** : est le stock plafond, le stock au-delà duquel il y a pléthore de stock.
- **Stock initial** : c'est la quantité des produits ou matières existantes en stocks au début.
- **Stock final** : c'est le stock initial augmenté des mouvements des stocks (entrées et sorties du stock) durant une période déterminée.
- **Stock tampon** : c'est le stock destiné à la consommation courante.

1.3.6 Avantages et inconvénients de la tenue d'un stock

La gestion des stocks a des avantages et des inconvénients qu'on présentera ci-dessous [4] :

Avantages :

- La tenue d'un stock sert d'abord à parer la pénurie ;
- Peut être constitué pour but spéculatif, on achète à bas pris pour revendre à haut prix ;
- Assurer une consommation régulière d'un produit donné, malgré que sa production soit irrégulière ;
- Eviter des dérangements dus à des achats fréquents ;
- Faire face au période d'inflation ;
- Constituer un stock pour faire face aux aléas de consommation, ainsi ceux de livraison ;
- Constitution d'un stock pour profiter de l'avantage d'une réduction sous forme de remises, ristournes, escomptes en achetant par grande quantité.

Inconvénients

- Immobilisation d'un capital onéreux qui ne génère pas de profit et peut être utilisé ailleurs ;
- La rupture de stock est considérée aussi comme un inconvénient parce qu'elle représente l'arrêt ou le ralentissement de la production ;
- Le stock a un caractère périssable de certains produits. En effet certains produits ne se conservent pas très longtemps, ou bien ils se conservent dans des conditions difficiles (température), qui va entraîner une augmentation des couts de stockage.

1.3.7 Documents de la gestion de stock

Pour assurer une bonne gestion des stocks, une entreprise utilise les documents suivants :

- **Fiche de stock** : Elle a pour but de centraliser toutes les informations relatives à la vie d'un produit, Le document est stationnaire et les fiches sont dans des bases pour permettre une facilité de la manipulation. Ce fichier est l'outil qui permet la tenue et la gestion des stocks, généralement les informations sur la fiche de stock sont classées en sections.

- **Bon de réception** : Il est établi sur la base d'un contrôle de conformité d'une livraison par rapport à la commande du point de vue quantitative et qualitative.
- **Bon d'entrée** : il est subdivisé en deux types :
 - Pour les produits acheminés de l'extérieur ;
 - Pour les produits finis provenant de l'entreprise.
- **Bon de sortie** : il est établie pour les produits acheminer vers l'exterieur et les produits finis, il sert à :
 - Tenir à jour le fichier de la gestion de stock ;
 - Permettre à la comptabilité analytique d'exploiter et d'imputer les frais ;
 - Valoriser les interventions de la maintenance et ainsi d'élaborer les parametres qui permettent de determiner le moment de renouvellement ou modification d'un équipement ou d'instalation ;
 - Contrôler le bon de fonctionnement de la gestion des stocks(demande produits satisfaite ou non satisfaite).
- **Bon de réintégration** : il se subdivise en deux types :
 - Pour matière première en excédent ;
 - Pour produit finis renvoyés par le client.
- **Bon de retour** : lorsqu'une partie ou la totalité d'un stock est renvoyé au fournisseur pour des raisons de non respect de specifications.
- **Bon de commande** : C'est un document commercial qui doit être rédigé avec clarté et precision afin éviter toute contestation ultérieure. Il est établi sur la base d'un choix de fournisseur ou de la demande d'achat.

1.3.8 Fonctions des stocks

La gestion de la production «à flux tendus» n'a pas fait disparaître complètement les stocks. Ceux-ci assurent en effet différentes fonctions dans l'entreprise [4].

a) Fonction de régulation

Les délais d'approvisionnement et de production sont, par nature, intermittents ou irréguliers. La constitution d'un stock diminue le risque de rupture d'un programme de fabrication ou évite de manquer une vente, c'est-à-dire augmente la qualité du service rendu par l'entreprise.

b) Fonction économique

Constituer des stocks permet de profiter des remises accordées par les fournisseurs sur les achats en grande quantité. D'autre part, ces mêmes fournisseurs imposent souvent des quantités minimales

de livraison, contraignant ainsi leurs clients à stocker.

c) Fonction de sécurité

Les stocks sont constitués pour permettre à l'entreprise de se protéger contre les variations aléatoires de la demande et les retards de livraison.

d) Fonction d'anticipation

Le stockage permet d'anticiper les hausses de prix des matières ou des produits.

e) Fonction technique

Le stockage préalable de certains produits est parfois nécessaire pour satisfaire les exigences techniques du processus de fabrication (par exemple, le bois doit être séché et les spiritueux doivent vieillir).

1.3.9 Classification des stocks

Grâce à des observations statistiques, les entreprises peuvent classer les stocks en différentes catégories (articles importants ou non) et réserver un contrôle rigoureux et régulier pour les articles de base [site01], ci-après nous présentons une classification selon l'importance des produits :

a) Loi de 20/80

La loi a été mise en jeu à partir de véritables lois établies à la suite de nombreuses observations. On a remarqué qu'environ 20% du nombre des articles consommés par les entreprises représentent environ 80% de la valeur totale de la consommation. Transposée à la gestion des stocks, cette structure veut que 20% d'articles en stock valent 80% du stock.

A partir de cette remarque, il est possible de diviser les articles stockés en deux groupes :

- **Groupe 1 (20/80)** : ceux qui sont plus importants et qu'il faut gérer avec rigueur et au moindre coût. Les économies sur les coûts à cet effet, pourront être élevées.
- **Groupe 2 (80/20)** : ceux qui sont les moins importants et qui nécessitent pas beaucoup de rigueur pour les gérer. Les stocks seront établis avec des marges de sécurité élevées.

b) Méthode ABC

La méthode ABC est une méthode identique à la précédente, mais un peu plus fine dans la mesure où l'on distingue trois groupes des produits :

- **Catégorie A** : il s'agit des articles déterminant pour l'entreprise. Ils représentent environ 10% du nombre d'articles, mais environ 70% de la valeur du stock. Ce sont des articles dits «articles standards». Ils doivent être gérés avec rigueur.
- **Catégorie B** : constitué des articles d'importance moyenne qui peuvent être gérés avec moins de rigueur. Ils représentent environ 20% de la valeur du stock. Ce sont des articles dits «articles courants».
- **Catégorie C** : les articles de faible importance qui peuvent être gérés d'une gestion plus lâche. Ils représentent environ 60% du nombre des articles et environ 10% de la valeur du stock. Ce sont des articles dits «articles exceptionnels».

On trouve aussi dans la littérature une autre méthode de classification présentée ci-après :

Classification selon les fréquences des mouvements

Dans cette classification on mesure trois classes de stocks selon leurs mouvements qui sont :

- **Stock à rotation rapide** : renouvelable au moins vingt fois par an.
- **Stock mort** : où il n'existe pas de renouvellement des stocks.
- **Stock actif** : appelé aussi tournant, qui implique un renouvellement fréquent des stocks.

1.4 Présentation du sujet d'étude

Le sujet qui nous a été présenté par l'organisme d'accueil porte sur le développement d'une application pour la gestion des stocks NAFTAL district GPL, articles de Bureau et Papeterie, documents de Source, équipements individuels, électricité, fournitures informatique, produits d'entretien, produit alimentaires et quincaillerie, etc.

1.5 Problématique

Durant la période de notre stage au sein de district GPL de Bejaia et plus précisément au niveau du magasin, on s'est rendu compte des problèmes suivants :

- Risque de tomber en rupture de stock (difficulté de constater l'arrivée au stock minimum). ;
- Absence d'un moyen efficace de vérification des entrées, sorties du stock ;
- Documents rédigés manuellement.

Tous ces facteurs engendrent la perte de temps et d'erreurs dans les quantités.

1.6 Objectifs du projet

Compte tenu des problèmes déduits de notre analyse et dans le souci de trouver les solutions appropriées, nous avons proposé d'implémenter une application permettant d'automatiser tous les processus de la gestion des stocks et rendre les tâches plus facile à l'utilisateur à savoir :

- Automatiser toutes les procédures de la gestion des entrées et des sorties du stock ;
- Calcul de l'autonomie des produits stockés en fonction de leur consommation ;
- Alerte automatique lorsque le stock atteint le seuil minimum ;
- Alerte automatique lorsque un produit s'approche de sa date de péremption ;
- Génération des documents du processus de gestion des stocks.

1.7 Conclusion

Au cours de ce chapitre, nous avons présenté d'une manière générale le district GPL de Bejaia ainsi que son fonctionnement, nous avons aussi présenté quelques généralités sur la gestion des stocks, puis les problèmes rencontrés au sein du magasin du district GPL Bejaia et nos objectifs afin de leurs fournir un système de gestion des stocks qui répond à leurs attentes. Cette étape sera le point de départ d'une étude détaillée et une conception que nous présenterons dans le prochain chapitre.

CHAPITRE 2

ANALYSE ET CONCEPTION

Sommaire

2.1	Introduction	16
2.2	Cycle de vie	16
2.3	Processus de développement UP	20
2.4	Présentation d'UML	21
2.5	Spécifications initiales	22
2.6	Diagramme des cas d'utilisation	25
2.7	Diagramme de séquence	32
2.8	Glossaire des données	36
2.9	Diagramme de classes	37
2.10	Schéma relationnel	38
2.11	Conclusion	40

2.1 Introduction

Cette partie de notre travail est très importante, elle se subdivise en deux (02) étapes considérées comme dépendantes l'une de l'autre, on parle bien sûr de l'analyse des besoins et de la conception du projet.

La première étape va être déduite de l'ensemble de la collecte d'information de notre organisme d'accueil qui se fera par échange de questions /réponses (ou interview), dans le but de décrire les besoins fonctionnels et non fonctionnels.

Ensuite on se penchera sur la deuxième étape : Conception, qui se base sur UML (*Unified Modeling Language*) qui à son tour, grâce à ses diagrammes, dissipe l'ambiguïté et offre un visuel global sur le projet avant sa réalisation.

2.2 Cycle de vie

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 et la vérification du processus de développement. 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 à temps et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.

Le cycle de vie du logiciel comprend généralement les activités suivantes [5] :

- Définition des objectifs ;
- Analyse des besoins et faisabilité ;
- Conception générale ;
- Conception détaillée ;
- Codage ;
- Tests unitaires ;
- Qualification ;
- Documentation ;
- Mise en production ;
- Maintenance.

2.2.1 Modèles de cycle de vie classique

a) Modèle en cascade (*waterfall model*)

Le modèle de cycle de vie en cascade a été mis au point dès 1966, puis formalisé aux alentours de 1970. Il définit des phases séquentielles, à l'issue de chacune d'elles, des documents sont produits pour en vérifier la conformité avant de passer à la suivante [5], la figure 2.1 ci-après le représente :

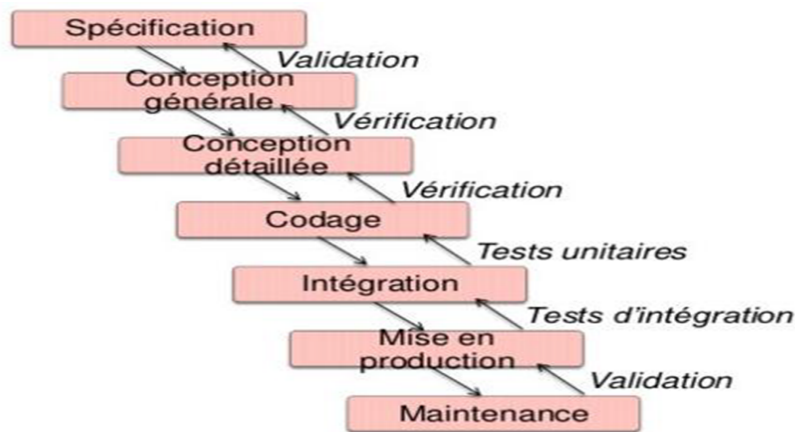


FIGURE 2.1 – Modèle en cascade

b) Modèle en V

Le modèle de cycle de vie en V part du principe que les procédures de vérification de la conformité du logiciel aux spécifications doivent être élaborées dès les phases de conception [6], la figure 2.2 ci-après le représente :

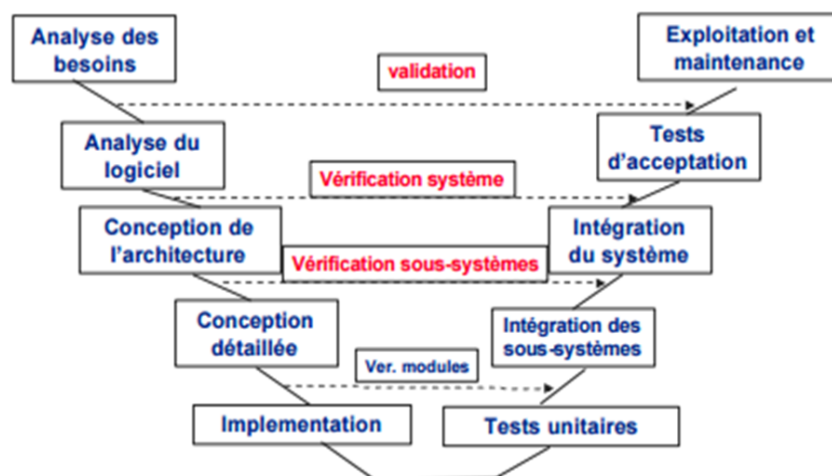


FIGURE 2.2 – Modèle en V

c) Modèle en spirale

Développé par [B. Boehm] en 1988 pour une raison de réduire les risques dans le cycle de vie du logiciel, il combine les éléments du modèle en cascade avec des modèles supplémentaires (voir la figure 2.3 ci-après). L'idée de ce modèle est de fournir le plus rapidement possible un prototype permettant de valider les concepts, et parfois de répondre à un appel d'offre avec un prototype de type «illustratif». Le principe est alors de fournir régulièrement une nouvelle version de l'application en se rapprochant progressivement de l'application finale [6].

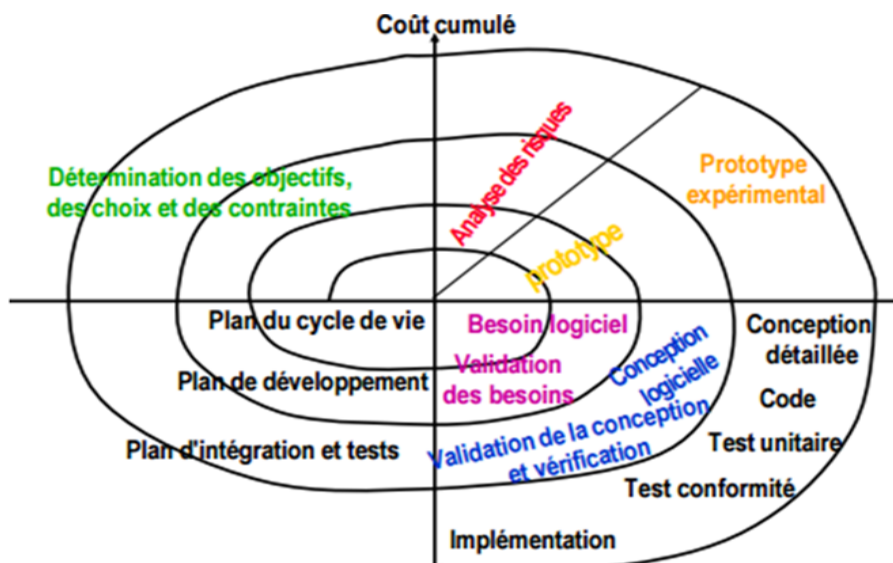


FIGURE 2.3 – Modèle en spirale

d) Modèle incrémental

Dans ce modèle chaque incrément est une construction partielle du logiciel. Trie les spécifications par priorités et les rassembles dans des groupes de spécifications. Chaque incrément implémente un ou plusieurs groupes jusqu'à ce que la totalité du produit soit finie [7], la figure 2.4 ci-après le représente :

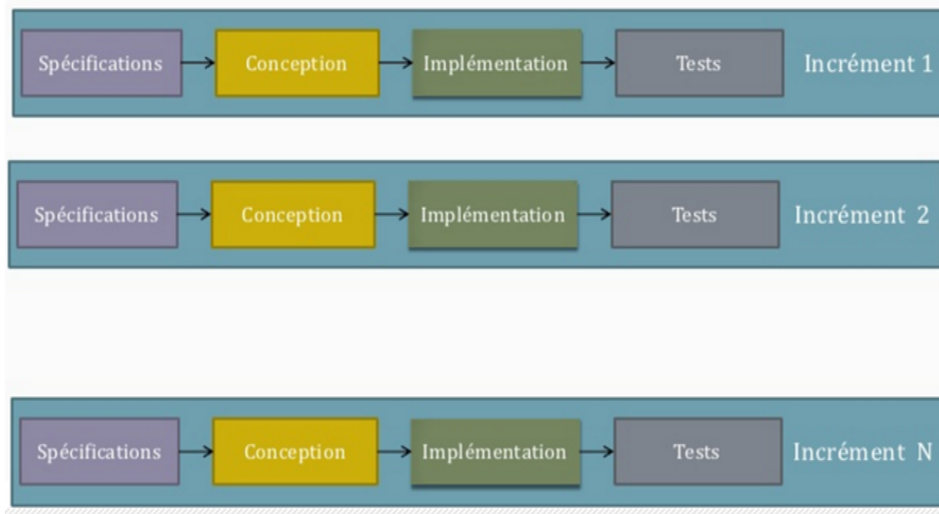


FIGURE 2.4 – Modèle incrémental

e) Prototypage

Le projet se fait sur plusieurs itérations, les développeurs construisent un prototype selon les attentes du client, le prototype est évalué par le client qui donne son feedback après avec quoi les développeurs adaptent le prototype selon les besoins du client. Quand le prototype satisfait le client, le code est normalisé selon les standards et aussi les bonnes pratiques [7], la figure 2.5 ci-après le représente :

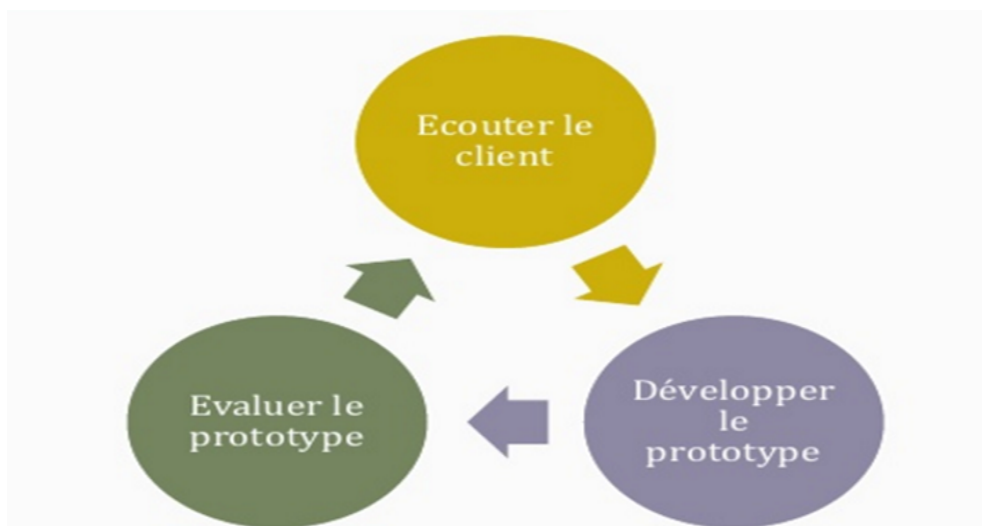


FIGURE 2.5 – Prototypage

2.2.2 Démarche à suivre

Aucun modèle n'est meilleur que l'autre car le choix se fait selon certains critères tel que la nature du projet, sa taille, la nature du client et nos compétences à pouvoir réaliser le projet.

Nous nous basons dans notre travail sur le prototypage. Les raisons pour lesquelles on a choisi ce modèle sont :

- Implication active du client donc moins de risque d'avoir des retours en arrière ;
- Le développeur est en contact direct avec le client ;
- S'adapte rapidement aux changements des besoins ;
- Progrès constant et visible ;
- Une grande interaction avec le produit.

2.3 Processus de développement UP

Un processus unifié est un processus de développement logiciel construit sur UML, il est itératif, incrémental, centré sur une architecture, conduit par les cas d'utilisation et piloté par les risques. La gestion d'un tel processus est organisée suivant les 4 phases suivantes : préétude (inception), élaboration, construction et transition. Ses activités de développement sont définies par 5 disciplines fondamentales qui décrivent la capture des besoins, l'analyse et la conception, l'implémentation, le test et le déploiement comme il est illustré par la figure 2.6 ci-dessous [site02].

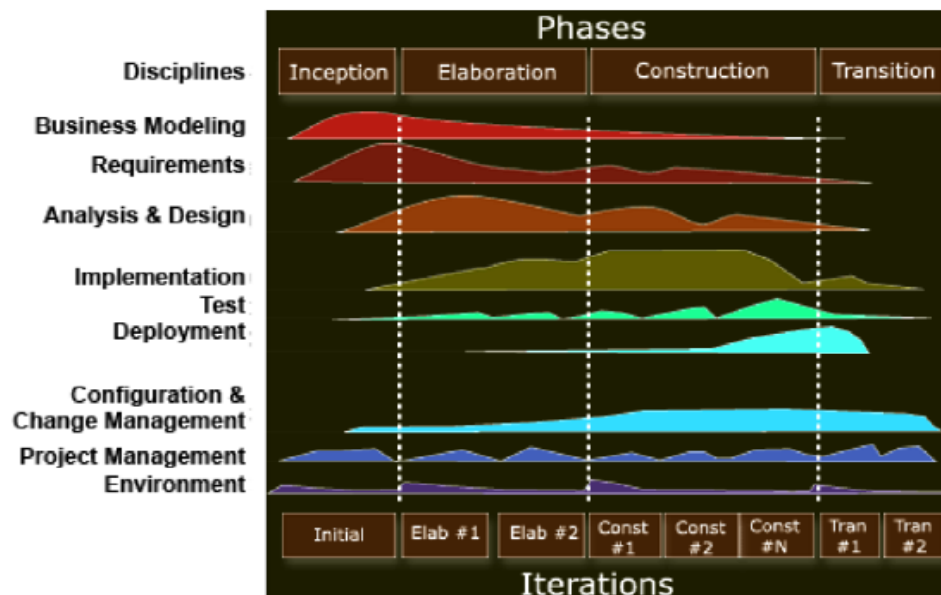


FIGURE 2.6 – Processus de développement UP

2.3.1 Vie du processus unifié

UP est un ensemble de principes génériques adapté en fonctions des spécificités des projets. UP répond aux préoccupations suivantes [site02] :

- **QUI** participe au projet ?
- **QUOI**, qu'est-ce qui est produit durant le projet ?
- **COMMENT** doit-il être réalisé ?
- **QUAND** est réalisé chaque livrable ?

2.3.2 Avantage d'UP

pour le processus de développement UP, on peut cité les avantages suivants :

- Méthodologie complète ;
- Identification rapide des risques ;
- Largement adopté en entreprise ;
- Intégration avec UML.

2.4 Présentation d'UML

2.4.1 Définition

UML est un langage de modélisation très complet, qui couvre de nombreux aspects du développement des logiciels, comme les exigences, l'architecture, les structures et les comportements. UML (*Unified Modeling Language*) est aujourd'hui le langage de modélisation d'applications informatiques le plus important du marché [8].

Ce langage est utilisé pour faire des diagrammes décrivant des programmes informatiques en termes des éléments constitutifs. UML n'est pas limité à l'ingénierie du logiciel, mais est également utilisé dans l'ingénierie des systèmes. La représentation des structures organisationnelles peut être unifiées par la modélisation du langage UML. La mise en place de logiciels de conception intégrée pour un projet peut aussi être à l'acte (diagrammes) d'un code initial du programme.

2.4.2 Modélisation UML

Le méta modèle UML fournit un ensemble d'outils qu'on appelle des diagrammes permettant de représenter l'ensemble des éléments du monde objet (classes, objets, etc.) ainsi que les liens qui les relie. Les 14 diagrammes UML sont dépendants hiérarchiquement et se complètent de façon à

permettre la modélisation d'un projet tout au long de son cycle de vie.

2.4.3 Justification de la modélisation UML

- Obtenir une modélisation de très haut niveau indépendante des langages et des environnements ;
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse ;
- Faire des simulations avant de construire un système ;
- Exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridiques, spécifications etc.

- Documenter un projet ;
- Générer automatiquement la partie logicielle d'un système ;
- valider le modèle vis à vis des clients ;
- spécifier les structures de données et le comportement du système ;
- fournir un guide pour la construction du système ;
- documenter le système et les décisions prises.

La figure 2.7 ci-après représente les trois composantes d'une modélisation UML :

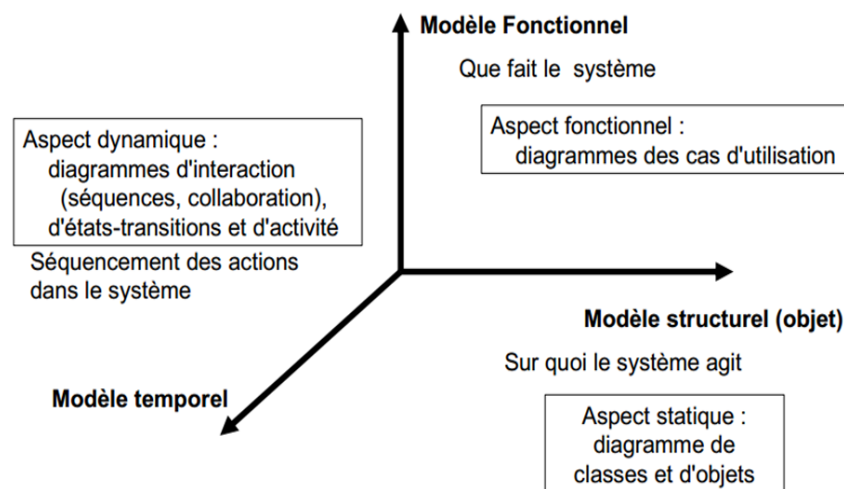


FIGURE 2.7 – Trois composantes d'une modélisation UML

2.5 Spécifications initiales

L'objectif de cette partie est d'analyser et définir les besoins de haut niveau et les caractéristiques du futur logiciel. Il se focalise sur les fonctionnalités requises par les utilisateurs, et sur la raison d'être de ces exigences. Le détail de la description des besoins se trouve dans les spécifications des cas d'utilisations.

2.5.1 Expression initiale des besoins

C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer une application adéquate, sa finalité est la description générale des fonctionnalités du système, en répondant à la question : quelles sont les fonctions du système ?

Notre système doit répondre aux exigences suivantes :

Exigences fonctionnelles

- Le système doit pouvoir récupérer des informations de chaque utilisateur suivant son nom d'utilisateur et son mot de passe, pour lui permettre un accès à l'application ;
- L'insertion des nouveaux produits et leur classement ;
- Modification des informations à propos du produit ou d'utilisateur ;
- La suppression des données (produit ou utilisateur) ;
- Impression des bons ou de fiche de stock ;
- Alerte : si la quantité restante en stock est inférieur à la quantité minimale prévue.

Exigences non-fonctionnels

A part les besoins fondamentaux, notre projet doit répondre aux critères suivants :

- **La rapidité de traitement** : En effet, vu la lenteur de traitement de la gestion des stocks, il faut automatiser quelques tâches, afin de diminuer le temps de traitement des stocks.
- **La performance** : Un logiciel doit être avant tout performant c'est à-dire à travers ses fonctionnalités, répond à toutes les exigences de l'entreprise d'une manière optimale.
- **La convivialité** : Notre future application doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.
- **La sécurité** : Notre future application doit permettre un accès sécurisé aux données (nous distinguons alors l'administrateur qui a le droit de tout faire et qui limite les droits d'accès des autres utilisateurs).
- L'application doit signaler toutes erreurs ou anomalie par des messages d'erreur.

2.5.2 Identification des acteurs

Définition d'un acteur

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec les systèmes étudiés. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [11].

Dans notre système, nous pouvons identifier ces acteurs :

- Chef magasin.
- Magasinier.
- Administrateur.
- Chef d'un service.

On n'oublie pas de mentionner le système externe : un autre district, interagissant au moment du transfert des produits.

Identification des messages échangés

Un message représente la spécification d'une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur [11].

Ci-dessous un résumé des messages échangés entre les acteurs et le système :

- Messages entrants (acteur vers système) :
 - M1 : Demande d'authentification ;
 - M2 : Demande de consultation ;
 - M3 : Etablir un bon d'entrée ;
 - M4 : Etablir un bon de sortie ;
 - M5 : Etablir un bon de consommation ;
 - M6 : Etablir un bon de transfert ;
 - M7 : Etablir une fiche de stock ;
 - M8 : Mettre à jour les utilisateurs ;
 - M9 : Mettre à jour les produits ;
 - M10 : Etablir l'inventaire ;
 - M11 : Etablir la demande de fourniture.
- Messages sortants (système vers acteur) :
 - M12 : Interface d'authentification ;
 - M13 : Interface de consultation adéquate ;
 - M14 : Interface pour établir un bon d'entrée ;
 - M15 : Interface pour établir un bon de sortie ;
 - M16 : Interface pour établir un inventaire ;
 - M17 : Interface de mise à jour adéquate ;
 - M18 : Interface pour remplir la demande de fourniture.

2.5.3 Diagramme de contexte

Diagramme de contexte d'un système est l'ensemble des objets échangés (produits, services et/ ou information) entre un système organisationnel et des acteurs de l'environnement externe du système étudié [11].

Ce diagramme nous permet de représenter et visualiser les composants de l'environnement de notre système à cet effet les différents acteurs interagissant avec le système, la figure ci-après représente le diagramme de contexte de notre système.

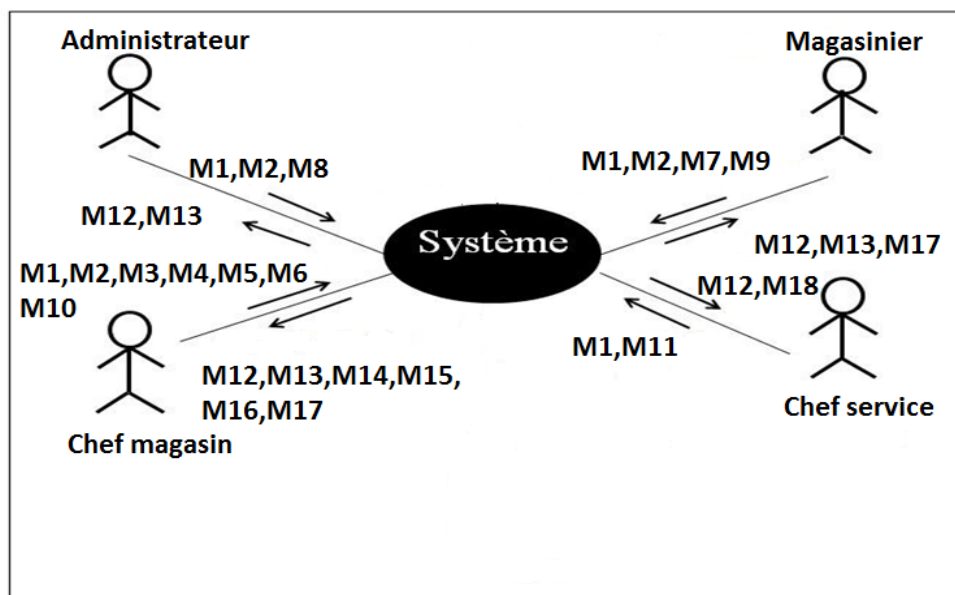


FIGURE 2.8 – Diagramme de contexte.

2.6 Diagramme des cas d'utilisation

Un cas d'utilisation (use case) modélise une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur [10].

2.6.1 Identification des cas d'utilisation

Magasinier :

- S'authentifier ;
- Ajouter un produit ;
- Modifier un produit ;

- Supprimer un produit ;
- Gérer fiche de stock ;
- Consulter le stock ;
- Imprimer fiche de stock.

Chef magasin :

- S'authentifier ;
- Gérer les demandes de fournitures ;
- Gérer les bons d'entrées ;
- Gérer les bons de sorties ;
- Gérer les bons de consommations ;
- Gérer les bons de transfert ;
- Gérer l'inventaire ;
- Consulter le stock ;
- Imprimer les bons.

Administrateur :

- S'authentifier ;
- Ajouter un compte d'utilisateur ;
- Modifier un compte d'utilisateur ;
- Supprimer un compte d'utilisateur.

Chef de service :

- S'authentifier ;
- Etablir une demande de fourniture.

2.6.2 Description des cas d'utilisation

La description textuelle des cas d'utilisation est libre. Cependant, cette description prend souvent une forme rédigée qui convient mieux à la communication avec les utilisateurs. Des règles de structuration doivent être appliquées pour en faciliter l'expression, la compréhension et la cohérence.

Description de cas d'utilisation «S'authentifier» :

S'authentifier est le cas d'utilisation qui permet à l'utilisateur d'accéder à son compte.

- **Cas d'utilisation** : S'authentifier.
- **Acteurs** :Administrateur/ Magasinier /Chef magasin/Chef service.
- **Objectif** :Ce cas d'utilisation permet à un utilisateur de se connecter au système.
- **Scénario nominal** :
 - Se connecter à l'application ;
 - Le système affiche un formulaire d'authentification ;
 - L'utilisateur saisit son nom d'utilisateur et son mot de passe puis valide ;
 - Le système vérifie la conformité des informations saisies[A1] ;
 - Le système donne l'accès à l'interface correspondante.
- **Alternatives [A1]** : Dans le cas où les informations saisies sont fausses, le système affiche un message d'erreur, et il réaffiche le formulaire d'authentification et attend que l'utilisateur ressaisie ses informations.

Description de cas d'utilisation « Gérer un utilisateur » :

Ce cas d'utilisation permet à l'administrateur de mettre à jour la liste des utilisateurs.

- **Cas d'utilisation** : Gérer un utilisateur.
- **Acteurs** : Administrateur.
- **Objectif** : Mettre à jour la liste des utilisateurs.
- **Pré-condition** : S'authentifier.
- **Scénario nominal** :
 - L'administrateur s'authentifie ;
 - L'administrateur demande le formulaire de gestion des utilisateurs (ajouter, modifier ou supprimer) ;
 - Le système affiche le formulaire ;
 - L'utilisateur effectue la modification désirer et valide [A1] ;
 - Le système enregistre le formulaire.
- **Alternatives [A1]** : Dans le cas de l'ajout ou de la modification, si un champ d'information obligatoire est incomplet le système affiche un message d'erreur.

Description de cas d'utilisation « Gérer un produit » :

Ce cas d'utilisation permet au magasinier d'effectuer une mise à jour des produits.

- **Cas d'utilisation** : Gérer un article.
- **Acteurs** : Magasinier.
- **Objectif** : Le magasinier a la possibilité de mettre à jour la liste des produits.
- **Pré-condition** : S'authentifier.
- **Scénario nominal** :
 - Le magasinier s'authentifie ;
 - Si le magasinier veut ajouter un produit, il remplit le formulaire d'ajout, en revanche s'il veut supprimer, il clique sur le bouton de suppression ou encore il remplit le formulaire de modification [A1] ;
 - Le système demande une confirmation ;
 - Le magasinier confirme la modification.
- **Alternatives [A1]** : Si un champ d'information obligatoire est incomplet le système affiche un message d'erreur.

Description de cas d'utilisation « Gérer fiche de stock » :

C'est le cas d'utilisation qui permet au magasinier de gérer la fiche de stock.

- **Cas d'utilisation** : Gérer fiche de stock.
- **Acteurs** : Magasinier.
- **Objectif** : Le magasinier a la possibilité de gérer une fiche de stock.
- **Pré-condition** : S'authentifier.
- **Scénario nominal** :
 - Le magasinier s'authentifie ;
 - Le magasinier demande le formulaire de gestion de fiche de stock ;
 - Le système affiche le formulaire ;
 - il remplit le formulaire de gestion [A1] ;
 - Le système demande une confirmation ;
 - Le magasinier confirme la modification.
- **Alternatives [A1]** : Si un champ d'information obligatoire est incomplet le système affiche un message d'erreur.

Description de cas d'utilisation « Consulter le stock » :

Ce cas utilisation permet à l'utilisateur de consulter le stock.

- **Cas d'utilisation** : Consulter stock.
- **Acteurs** : Administrateur/Magasinier/Chef service.
- **Objectif** : L'utilisateur consulte le stock pour contacter les fournisseurs.
- **Pré-condition** : S'authentifier.
- **Scénario nominal** :
 - L'utilisateur s'authentifie ;
 - L'utilisateur demande la liste des produits. ;
 - Le système affiche la liste des produits.

Description de cas d'utilisation « Etablir un bon d'entrée » :

Ce cas utilisation permet au chef magasin d'établir les bons d'entrée.

- **Cas d'utilisation** :Etablir un bon d'entrée.
- **Acteurs** : Chef magasin.
- **Objectif** : À tout moment, l'économat doit pouvoir accéder au système pour établir un bon d'entrée.
 - **Pré-condition** : S'authentifier.
 - **Scénario nominal** :
 - L'utilisateur s'authentifie ;
 - L'utilisateur demande d'établir un bon d'entée ;
 - Le système affiche un formulaire ;
 - L'utilisateur saisit les informations et valide [A1] ;
 - Le système affiche une confirmation et génère un bon d'entrée.
 - **Alternatives [A1]** :Si un champ d'information obligatoire est incomplet système affiche un message d'erreur.

Description de cas d'utilisation « Etablir un bon de sortie»

Ce cas utilisation permet à l'utilisateur d'établir un bon de sortie.

- **Cas d'utilisation** : Etablir un bon de sortie.
- **Acteurs** : Chef magasin.
- **Objectif** : Mettre à jour le stock.
- **Pré-condition** : S'authentifier.

- **Scénario nominal :**

- L'utilisateur s'authentifie ;
- L'utilisateur demande d'établir un bon d'entrée ;
- Le système affiche un formulaire de sortie ;
- L'utilisateur saisit les informations et valide [A1] ;
- Le système affiche une confirmation et génère un bon de sortie.

- **Alternatives [A1] :** Si un champ d'information obligatoire est incomplet système affiche un message d'erreur.

2.6.3 Nos diagrammes de cas d'utilisation

Les figures ci-après représentent les diagrammes des cas d'utilisation que nous avons réalisés :

Diagramme de cas d'utilisation associé à l'administrateur :

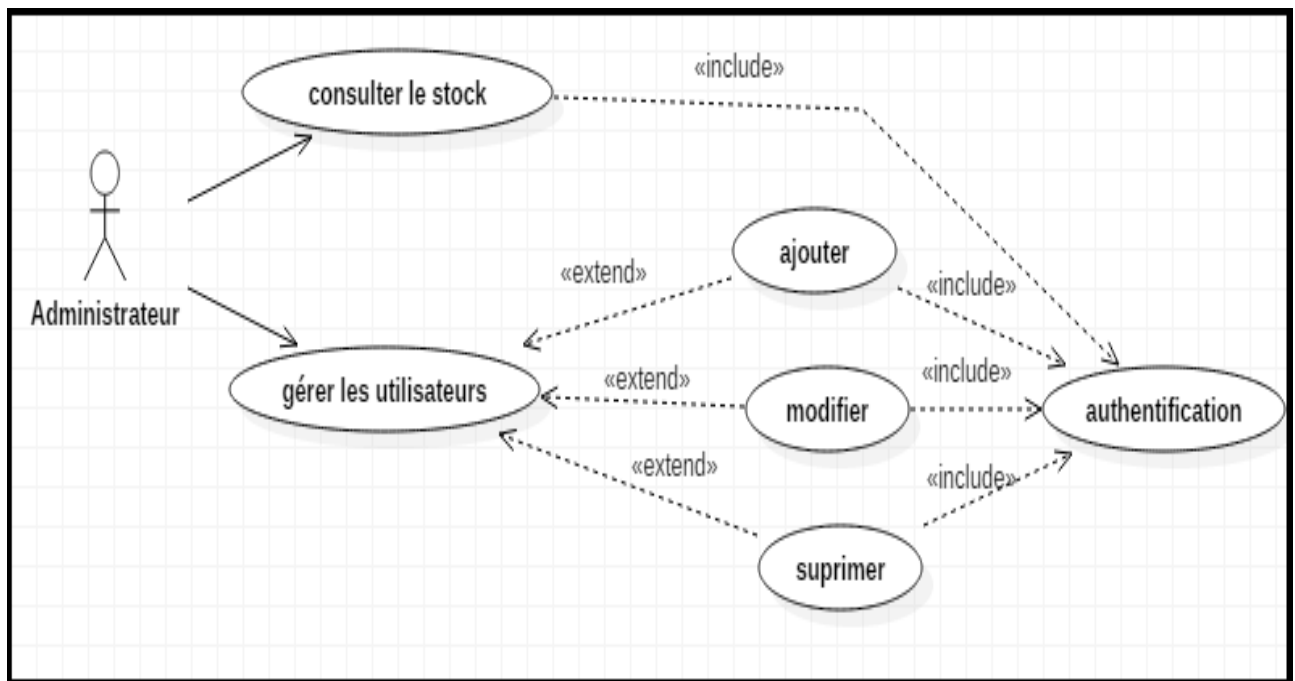


FIGURE 2.9 – Diagramme de cas d'utilisation associé à l'administrateur

Diagramme de cas d'utilisation associé au magasinier :

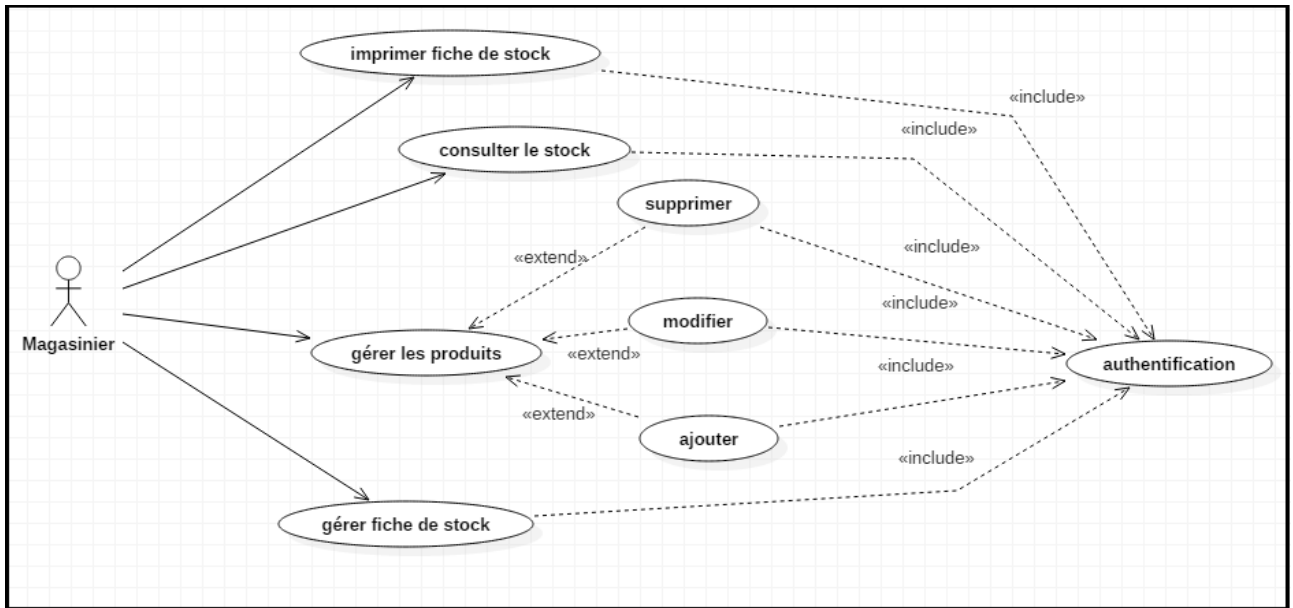


FIGURE 2.10 – Diagramme de cas d'utilisation associé au magasinier

Diagramme de cas d'utilisation associé au chef magasin :

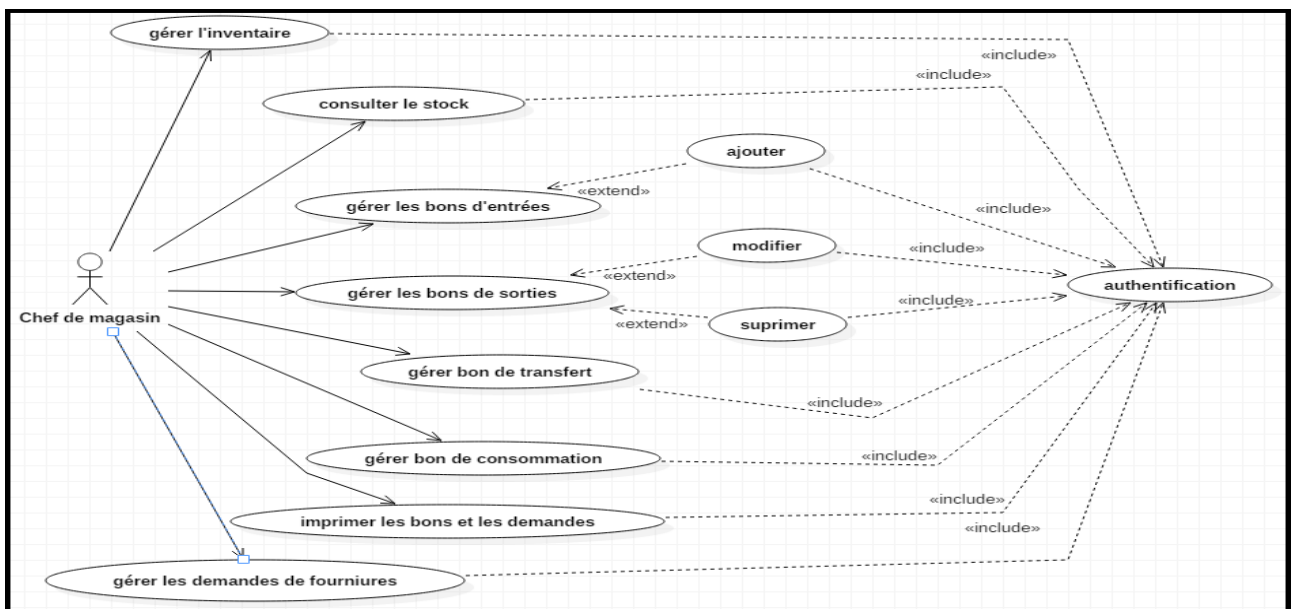


FIGURE 2.11 – Diagramme de cas d'utilisation associé au chef magasin

2.7 Diagramme de séquence

Le diagramme de séquences permet de présenter les interactions entre objets selon un point de vue temporel. L'accent est mis sur la chronologie des envois des messages.

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés [10].

2.7.1 Message synchrone et asynchrone

Dans un diagramme de séquence, deux types de messages peuvent être distingués :

- **Message synchrone** : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message.

- **Message asynchrone** : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.

Dans un diagramme de séquence, il est possible de distinguer des sous-ensembles d'interactions qui constituent des fragments.

2.7.2 Fragment d'interaction

Un fragment d'interaction se représente globalement comme un diagramme de séquence dans un rectangle avec indication dans le coin à gauche du nom du fragment. Un port d'entrée et un port de sortie peuvent être indiqués pour connaître la manière dont ce fragment peut être relié au reste du diagramme. Dans le cas où aucun port n'est indiqué c'est l'ensemble du fragment qui est appelé pour exécution. La liste suivant regroupe les opérateurs d'interaction les plus utilisés :

- **L'opérateur alternative (alt)** : est un opérateur conditionnel possédant plusieurs opérandes, c'est un peu l'équivalent d'une exécution à choix.

- **L'opérateur boucle (loop)** : un fragment combine de ce type possède un sous-fragment et spécifie un compte minimum et maximum (boucle) ainsi de condition de garde.

- **L'opérateur parallèle (par)** : contient au moins deux sous fragments séparés par des pointillés qui s'exécutent simultanément.

- **L'opérateur (ref)** : permet de faire appel à un autre diagramme de séquence décrit par ailleurs.

2.7.3 Etude de cas d'utilisation

Diagramme de séquence du cas d'utilisation « Authentification » :

Un utilisateur a la possibilité de s'authentifier en saisissant ces propres coordonnées (son login et mot de passe), après avoir saisi ses coordonnées, le système procède à la vérification des informations introduites, si l'une des coordonnées est incomplète le système affiche un message d'erreur sinon il envoie une requête de vérification à la base de données, qui procède une vérification de ces données en cas d'erreur le système affiche un message d'erreur sinon il affiche l'interface correspondant.

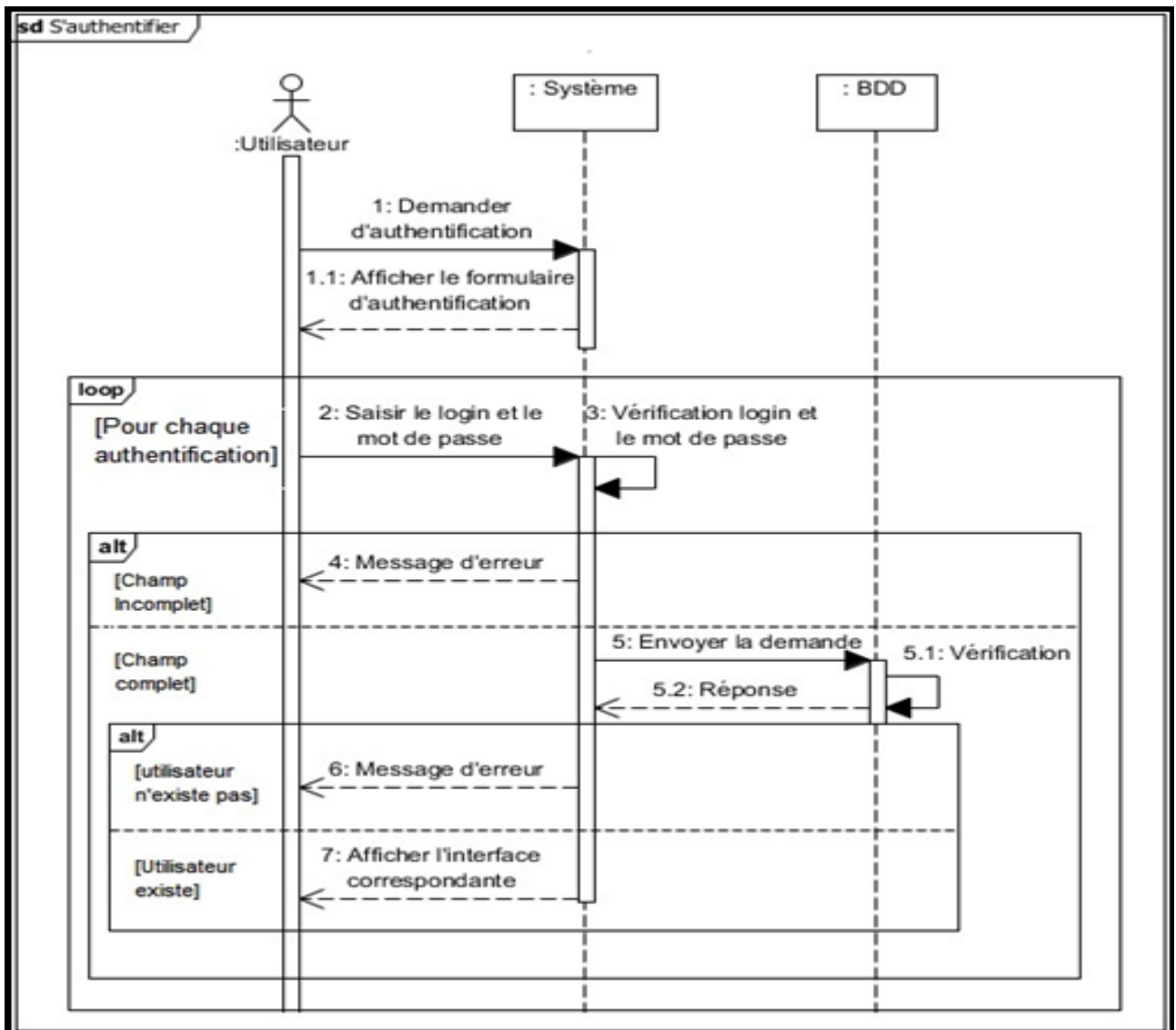


FIGURE 2.12 – Diagramme de séquence du cas d'utilisation « Authentification ».

Diagramme de séquence du cas d'utilisation « Gérer un utilisateur » :

Dans ce cas d'utilisation, l'administrateur doit d'abord s'authentifier, ce dernier peut établir trois cas d'utilisation différents, l'ajout qui consiste à ajouter un utilisateur dans la base de donnée en remplissant un formulaire, la modification qui consiste à apporter un changement sur les informations liées à un utilisateur sélectionné et la suppression qui consiste à effacer un utilisateur sélectionné dans la base de donnée.

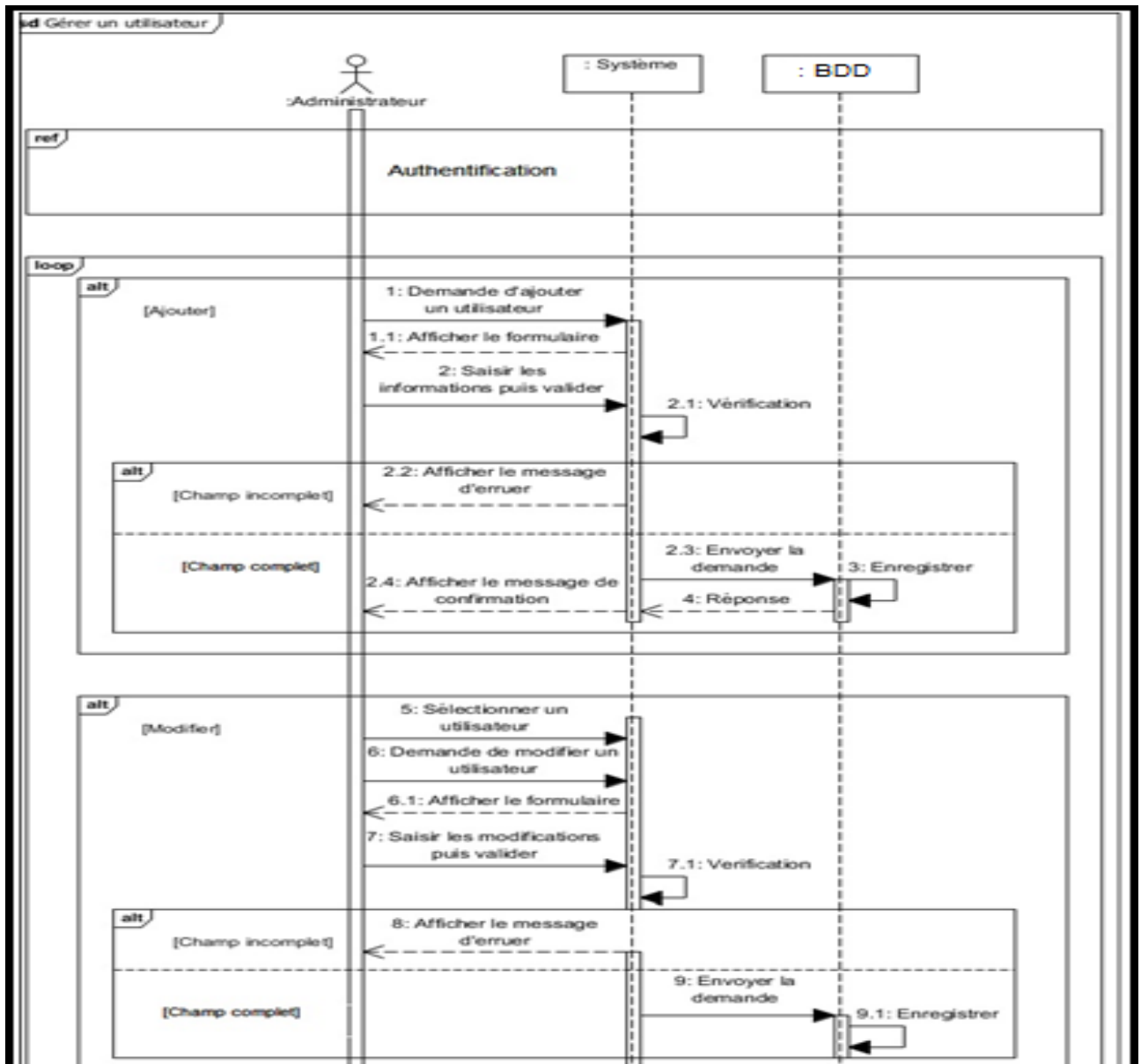


FIGURE 2.13 – Diagramme de séquence du cas d'utilisation « Gérer un utilisateur ».

Diagramme de séquence du cas d'utilisation « Gérer un produit » :

Après l'authentification, le magasinier peut établir des mises à jour sur les différents produits, il est capable d'ajouter, de supprimer ou de modifier un produit, des formulaires différents s'affichent pour chaque un des trois scénarios, le système exécute alors les mises à jour dans la base de donnée.

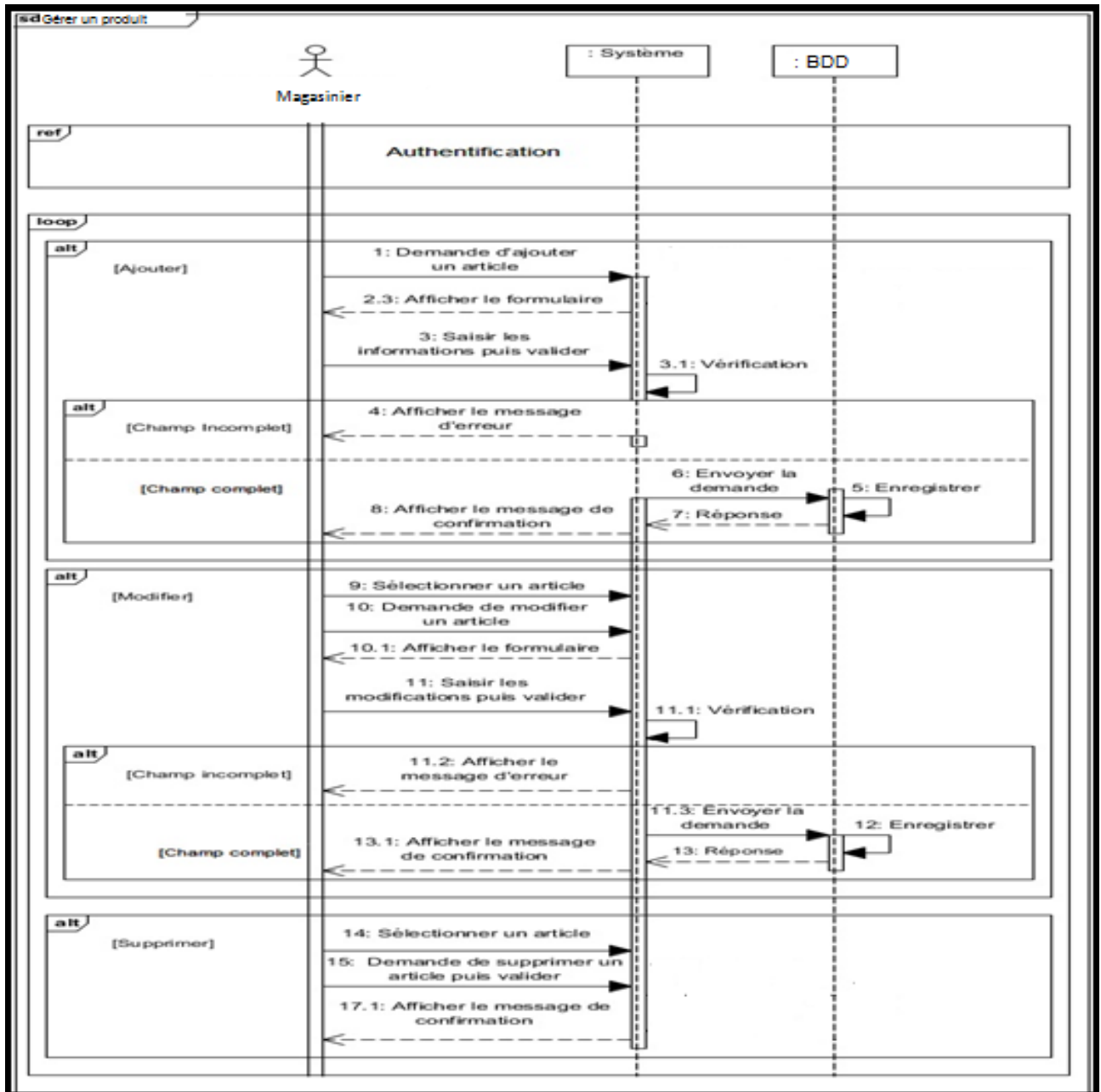


FIGURE 2.14 – Diagramme de séquence du cas d'utilisation « Gérer un article ».

2.8 Glossaire des données

Attributs	Description	Type
id-stock	identificateur stock	Numérique
qutt-stock	la quantité en stock	Numérique
pudadd	prix unitaire en dinars	Numérique
mondadd	montant en dinars	Numérique
u-m	unité de mesure	Alphanumérique
id-produit	identificateur de produit	Numérique
nom-produit	nom du produit	Alphabétique
prix-unitaire	le prix unitaire du produit	Numérique
ref	référence du produit	Alphanumérique
obs	observation	Alphabétique
num-be	le numéro du bon d'entrée	Numérique
qtt-e	la quantité entrée en stock	Numérique
montotda	le montant total en dinars	Numérique
num-bs	le numéro du bon de sortie	Numérique
qtt-s	la quantité sortie du stock	Numérique
qtt-d	la quantité demandée	Numérique
id-famille	identificateur de la famille des produits	Numérique
nom-famille	nom de la famille des produits	Alphabétique
id-fourn	identificateur du fournisseur	Numérique
nom-fourn	nom du fournisseur	Alphabétique
id-fiche	identificateur de la fiche mvt-stock	Numérique
id-demande	identificateur de la demande de fourniture	Numérique
qtt-accordée	la quantité accordée	Numérique
id-utilisateur	identificateur de l'utilisateur	Numérique
login	nom de l'utilisateur	Alphabétique
password	mot de passe de l'utilisateur	Alphanumérique
choix-service	le choix du service	Alphanumérique

TABLE 2.1 – Glossaire des données

2.9 Diagramme de classes

Diagramme de classe modélise la structure statique d'un système, en représentant graphiquement les classes interconnectées par des associations ou des relations de généralisation. Les attributs et les opérations sont mentionnés sur les classes du diagramme de classe [12].

Dans la figure ci-dessous est présenté le diagramme de classe de gestion de stock :

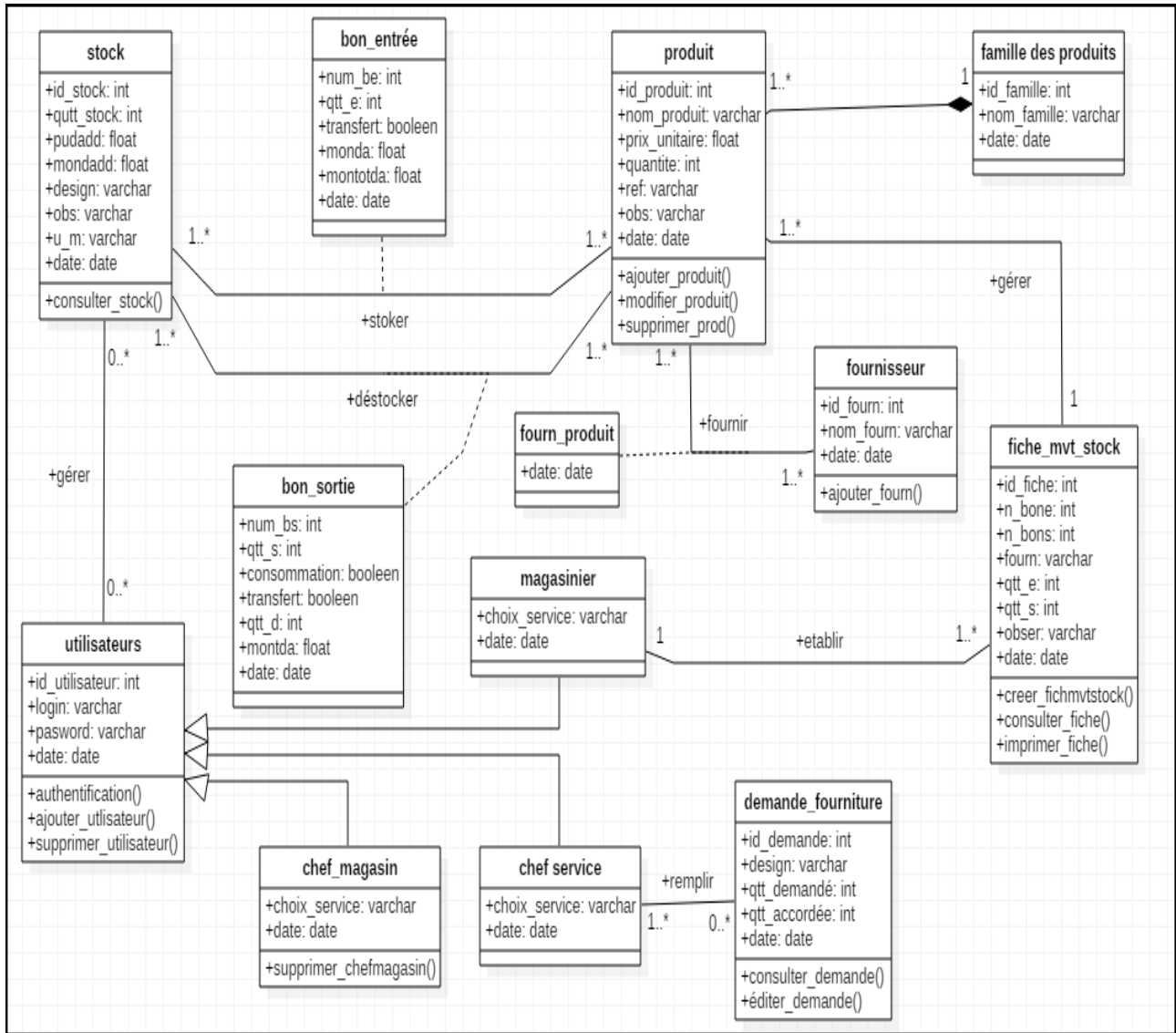


FIGURE 2.15 – Diagramme de classe.

2.10 Schéma relationnel

2.10.1 Règles de passage

Règles de passage du diagramme de classe au modèle relationnel [13] :

- **Règle 1 :**

Toute entité devient une relation ayant pour clé primaire son identifiant. Chaque propriété se transforme en attribut.

- **Règle 2 :**

Toute association hiérarchique (de type $[1, n]$) se traduit par une clé étrangère. La clé primaire correspondant à l'entité père (côté n) migre comme clé étrangère dans la relation correspondant à l'entité fils (côté 1).

- **Règle 3 :**

Toute association non hiérarchique (de type $[n, n]$) devient une relation. La clé primaire est formée par la concaténation (juxtaposition) l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.

- **Règle 4 :**

Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes :

1) Décomposition par distinction : Il faut transformer chaque sous-classe en une relation. La clé primaire de la surclasse migre dans la relation issue de la sous-classe et devient à la fois clé primaire et la clé étrangère.

2) Décomposition descente (push-down) : s'il existe une totalité ou de partition sur l'association d'héritage, il est possible de ne pas traduire la relation issue de la surclasse. Il faut alors faire migrer tous ses attributs dans la relation issue de la sous-classe.

3) Décomposition ascendante (push-up) : il faut supprimer la relation issue de la sous-classe et faire migrer les attributs dans la relation issue de la surclasse.

- **Règle 5 :**

La clé primaire des relations déduites des classes composantes doit contenir l'identifiant de la classe composite (quelque soit les multiplicités).

2.10.2 Modèle logique de données

En appliquant les règles de passages pour notre diagramme de classes, nous aboutissons au schéma relationnel suivant :

- **produit**(**id-produit**, nom-produit, prix-unitaire, quantite, obs, ref, date, #id-famille,

#id-fiche).

- **stock**(**id-stock**, qtt-stock, pudadd, mondadd, design, pbs, u-m, date).
- **fiche-mvt-stock**(**id-fiche**, n-bone, n-bons, fourn, qtt-e, qtt-s, obs, date, #id-utilisateur).
- **famille-produit**(**id-famille**, nom-famille, date).
- **utilisateurs-**(**id-utilisateur**, login, password, date).
- **chef-magasin**(**id-utilisateur**, choix-servise, date).
- **chef-service**(**id-utilisateur**, choix-servise, date).
- **magasinier**(**id-utilisateur**, choix-service, date).
- **demande-fourniture**(**id-demande**, design, qtt-demandé, qtt-accordé, date).
- **bon-entrée**(**id-produit**, **id-stock**, num-be, qtt-e, monda, montotda, transfert, date).
- **bon-sortie**(**id-produit**, **id-stock**, num-bs, qtt-s, consommation, transfert, qtt-d, monda,

date).

- **fournisseur**(**id-fourn**, nom-fourn, date).
- **fourn-produit**(**id-produit**, **id-fourn**, date).

2.11 Conclusion

Dans ce présent chapitre, nous avons proposé une solution conceptuelle formée de plusieurs diagrammes UML, dont le diagramme de classes qui a aidé à déduire le schéma relationnel de notre future base de données. Cela après avoir bien précisé les besoins techniques que le projet doit assurer. Dans le chapitre suivant on présentera l'implémentation de la solution conceptuelle proposée.

CHAPITRE 3

IMPLÉMENTATION

Sommaire

3.1	Introduction	42
3.2	Langages de programmation utilisés	42
3.3	Outils de développement	43
3.4	Projection de quelques interfaces de notre application	44
3.5	Conclusion	57

3.1 Introduction

Ce dernier chapitre représente la phase réalisation, nous allons détailler le processus suivi dans l'implémentation de notre application. On commencera par la présentation des outils et langages utilisés dans l'implémentation puis une présentation de l'application développée avec quelques prises d'écran des différentes interfaces et fiches générées.

3.2 Langages de programmation utilisés

3.2.1 SQL

SQL signifie *Structured Query Language* ou "langage de requêtes structuré". C'est l'un des langages les plus employés par les systèmes de gestion de bases de données relationnelles (SGBDR) pour stocker et récupérer des données. Il est utilisé par des systèmes de bases de données comme MySQL, Oracle, PostgreSQL, Sybase et *Microsoft SQL Server* [14].

MySQL

MySQL est un système de gestion de bases de données relationnelles, robuste et rapide. Une base de données permet de manipuler les informations de manière efficace, de les enregistrer, de les trier, de les lire et d'y effectuer des recherches. Le serveur MySQL contrôle l'accès aux données pour s'assurer que plusieurs utilisateurs peuvent se servir simultanément d'une même base de données pour y accéder rapidement et pour garantir que seuls les utilisateurs autorisés peuvent accéder aux données.



MySQL est donc un serveur multi-utilisateur et multithread. Il utilise SQL comme langage standard des requêtes de bases de données. MySQL est disponible depuis 1996, mais son développement remonte à 1979 [14].

3.2.2 JAVA

Java est un langage de programmation moderne développé par Sun Microsystems (aujourd'hui racheté par Oracle). Il ne faut surtout pas le confondre avec JavaScript (langage de scripts utilisé principalement sur les sites web), car Java n'a rien à voir.

Une de ses plus grandes forces est son excellente portabilité : une fois votre programme créé, il

fonctionnera automatiquement sous Windows, Mac, Linux, etc

On peut faire de nombreuses sortes de programmes avec Java [15] :

- des applications, sous forme de fenêtre ou de console ;
- des applets, qui sont des programmes Java incorporés à des pages web ;
- des applications pour appareils mobiles, avec J2ME ;
- et bien d'autres ! J2EE, JMF, J3D pour la 3D etc

En plus de java, on a eu besoin les plugings suivants :

API d'export en PDF :

iText est une API permettant à partir d'un programme JAVA de générer des documents PDF. iText s'intègre facilement dans un projet JAVA. Ses fonctionnalités et sa simplicité d'utilisation permettent aux développeurs d'un projet JAVA de générer des rapports PDF [site04].

JFreeChart :

JFreeChart est une bibliothèque open source qui permet d'afficher des données statistiques sous la forme de graphiques, donc elle fait une interprétation d'une table de données par un histogramme [site05].

Apache POI :

POI est l'acronyme de *Poor Obfuscation Implementation*. C'est un projet open source dont le but est de permettre la manipulation de fichiers de la suite bureautique Office de Microsoft dans des applications Java [site06].

3.3 Outils de développement

3.3.1 WampServer

Voici comment Wampserver est (très bien) présenté dans Wikipédia :

WampServer 2 (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi qu'une administration pour les deux bases SQL PhpMyAdmin et SQLiteManager. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs à travers d'une *tray icon* (icône près de l'horloge de Windows) [site03].

3.3.2 PhpMyAdmin

PhpMyAdmin est un outil entièrement écrit en PHP qui fournit une interface simple et très complète pour administrer une base de données MySQL. La plupart des commandes de l'utilitaire mysql peuvent s'effectuer par l'intermédiaire de phpMyAdmin, les opérations possibles dépendant bien sûr des droits de l'utilisateur qui se connecte à la base.

Voici une liste des principales possibilités [16] :

1. Créer et détruire des bases de données (sous le compte root de MySQL).
2. Créer, détruire, modifier la description des tables.
3. Consulter le contenu des tables, modifier certaines lignes ou les détruire, etc.
4. Exécuter des requêtes SQL interactivement.
5. Charger des fichiers dans des tables et, réciproquement, récupérer le contenu de tables dans des fichiers ASCII.
6. Administrer MySQL.

3.3.3 Netbeans

NetBeans est un environnement de développement intégré (EDI), placé en open source. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native, ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons [site03]. Il comprend toutes les caractéristiques d'un IDE moderne :

1. configuration et gestion de l'interface graphique des utilisateurs
2. support de différents langages de programmation
3. traitement du code source (édition, navigation, formatage, inspection etc)
4. fonctions d'import/export depuis et vers d'autres IDE, tels qu'Eclipse ou JBuilder.
5. accès et gestion de bases de données, serveurs Web, ressources partagées.
6. gestion de tâches (à faire, suivi etc).
7. documentation intégrée.

3.4 Projection de quelques interfaces de notre application

Dans cette section nous allons présenter le contenu des principales interfaces de notre application développée pour la gestion des stocks NAFTAL district GPL Bejaia.

3.4.1 Authentification

A l'exécution de l'application, la fenêtre ci-dessous apparaît, c'est l'interface d'authentification des utilisateurs, elle propose un choix de statut par un JCombobox (ADMIN, CHEF MAGASIN, MAGASINIER, CHEF DE SERVICE) lié à la table utilisateur dans la BDD, un utilisateur s'authentifie en insérant son nom d'utilisateur et son mot de passe (voir Figure.3.1).

1. propose le menu déroulant de l'interface d'authentification présenté sur 4.



FIGURE 3.1 – Interface d'authentification.

2. c'est le bouton de validation des champs saisis par l'utilisateur sur la fenêtre d'authentification.
3. c'est le bouton permettant de quitter l'application.
4. le menu déroulant des utilisateurs

3.4.2 Gestion des utilisateurs

Elle est autorisée uniquement à l'administrateur, et lui permet l'ajout, modification et la suppression d'utilisateurs. la fenêtre 3.2 apparaît après une authentification de l'administrateur.

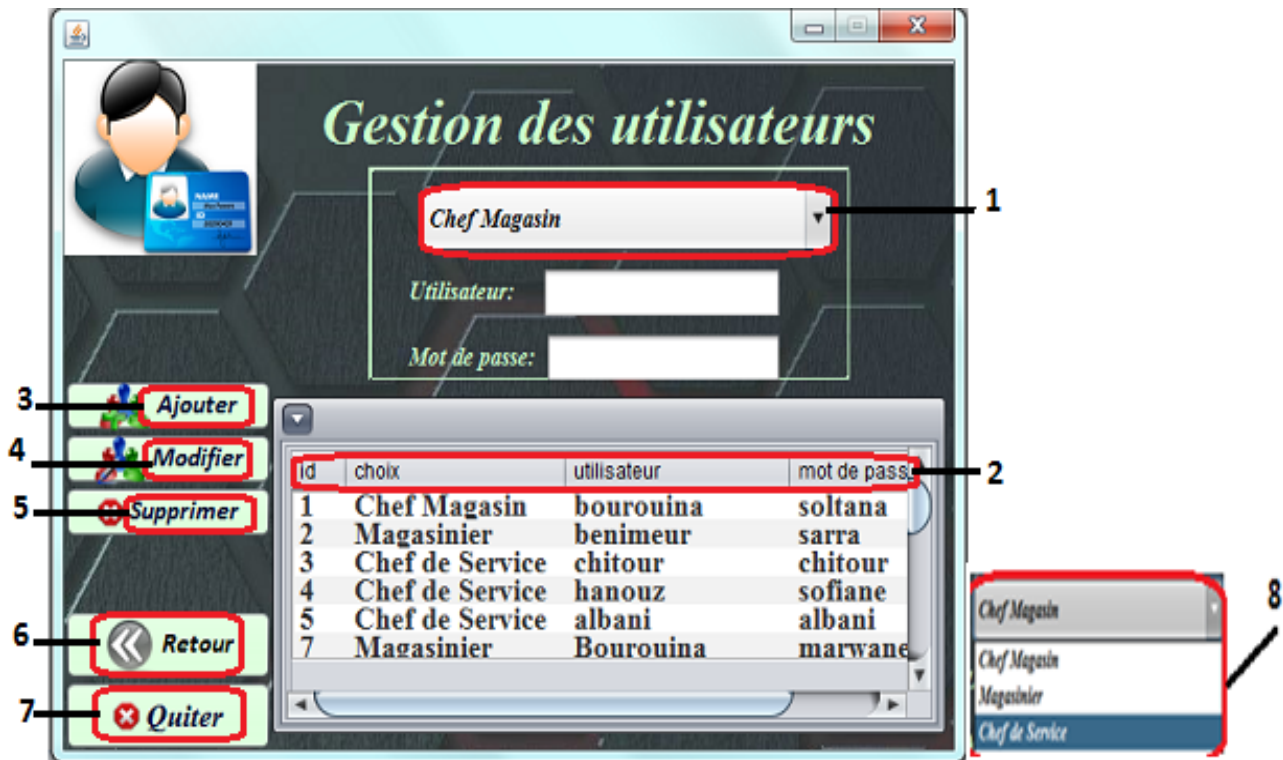


FIGURE 3.2 – Interface de gestion des utilisateurs.

1. offre un menu déroulant présenté sur 8.
2. représente les champs de la table utilisateur (id, choix du statu, le nom de l'utilisateur et aussi son mot de passe) avec les enregistrements appropriés.
3. c'est le bouton qui permet d'ajouter un utilisateur.
4. c'est le bouton qui permet de modifier les informations d'un utilisateur.
5. c'est le bouton qui permet de supprimer un utilisateur.
6. nous permet de faire un retour à la page d'accueil, pour le cas où l'administrateur voudras s'authentifier comme un chef de service, afin d'accéder à une autre interface que celle de l'administrateur.
7. c'est le bouton qui permet à l'administrateur de quitter l'application.
8. un menu déroulant des utilisateurs.

3.4.3 Demande de fourniture

Selon la spécification d'organisme d'accueil, la demande de fourniture est autorisée au chef de service, la fenêtre 3.3 présentée ci-après s'affiche après authentification d'un chef de service.

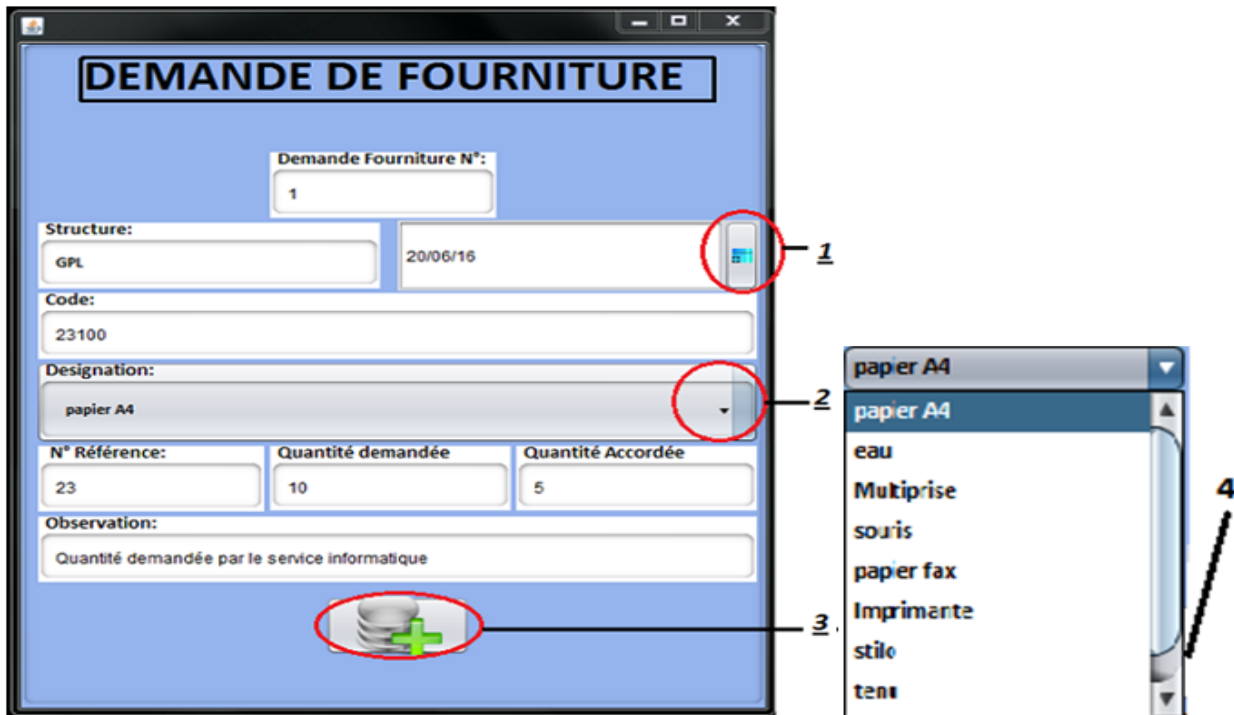


FIGURE 3.3 – Demande de fourniture.

1. nous permet de faire un choix d'une date dans un calendrier.
2. un JComboBox qui nous offre des choix pour des désignations de produits montrés sur 4.
3. nous permet l'enregistrement des champs saisis ou choisis par le demandeur de fourniture dans une table, afin que le chef magasin puisse la traiter, après sa consultation des demandes.
4. c'est un menu déroulant tiré de la table produit.

3.4.4 Interface pour chef magasin

Après l'authentification du chef magasin, l'interface qui lui est appropriée s'affiche, en considérant quatre onglets qui sont : Bon d'entrée, Bon de sortie, Etat du stock et consultation des demandes de fournitures, qui proposent un visuel convivial simple des résultats d'action des magasiniers et chef de service.

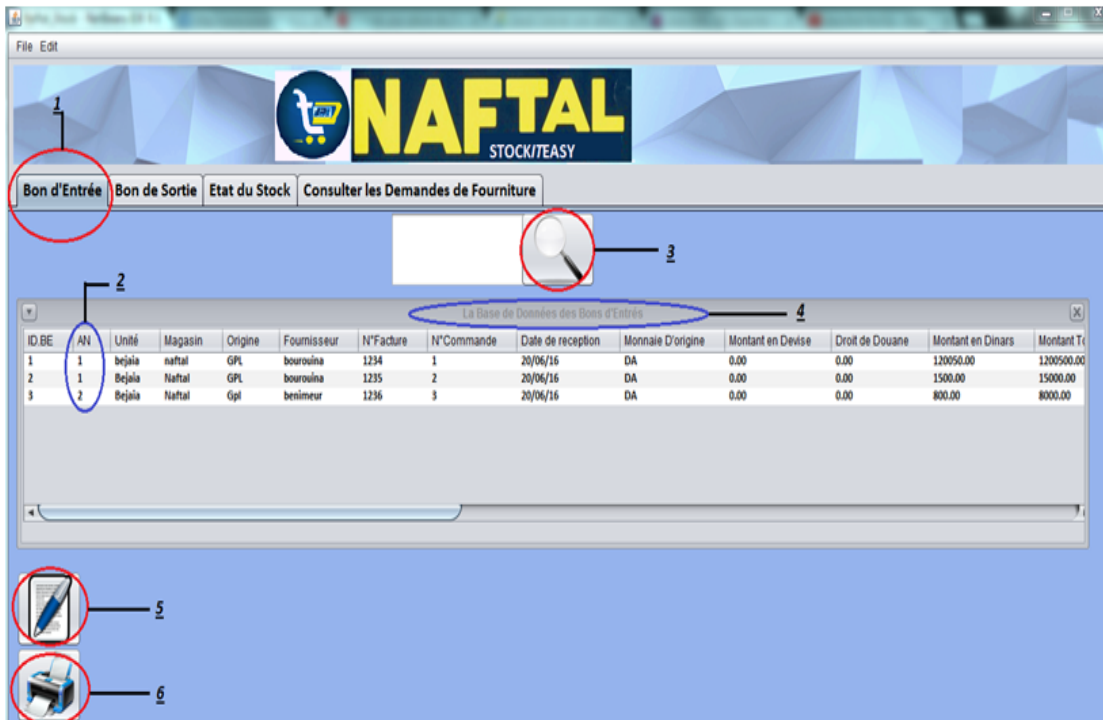


FIGURE 3.4 – Interface pour chef magasin.

1. l'onglet du bon d'entrée.
2. représente le numéro d'un Bon d'Entrée, qui n'est pas un champ unique, c'est-à-dire que plusieurs enregistrements peuvent avoir un même numéro de bon.
3. pour faciliter la recherche dans la table des bons.
4. la table représente les enregistrements de tous les bons d'entrées.
5. c'est un bouton qui nous permet une manipulation efficace du formulaire de bon d'entrée.
6. c'est un bouton permettant l'impression des bon d'entrées en forme de rapport PDF.

3.4.5 Bon d'entrée

Le chef magasin a un accès complet aux différentes manipulations du bon d'entrée à partir du formulaire présenté par la figure 3.5 ci-après comme : ajout et modification des bons, l'impression, l'ajout d'une désignation (Produit) et enfin l'ajout d'un fournisseur etc.

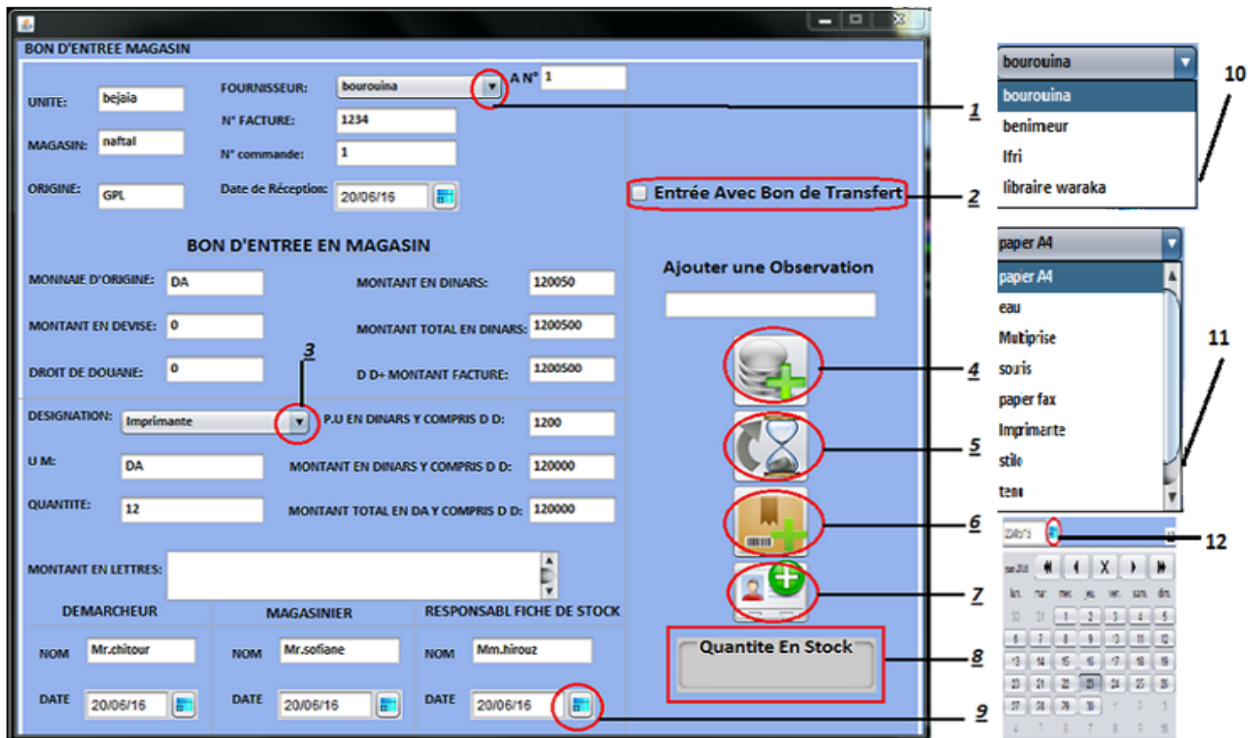


FIGURE 3.5 – Bon d'entrée.

Le formulaire de bon d'entrée est très chargé d'information à remplir, on compte 28 champs de saisi et de choix.

1. présente un jcombobox offrant un menu déroulant présenté en 10.
2. sert à choisir la nature du bon.
3. nous montre un menu déroulant détaillé dans 11.
4. c'est un bouton qui permet l'enregistrement des champs entrés par l'utilisateur dans le formulaire de bon d'entrée, et de les avoir comme des lignes dans la table bon d'entrée.
5. bouton permettant la modification d'une ligne enregistrée précédemment dans la table bon d'entrée, pour ce faire, on doit d'abord sélectionner la ligne à modifier, celle-ci sera chargée sur le formulaire de bon d'entrée ensuite changer les champs qui doivent être modifier puis ensuite cliquer sur le bouton de modification pour la réenregistré dans la table.
6. c'est le bouton permettant l'ajout d'une désignation de produit dans le cas où une désignation d'un produit n'existe pas dans le menu déroulant approprié.
7. c'est le bouton permettant l'ajout d'un fournisseur dans le cas où le fournisseur n'existe pas dans le menu déroulant approprié.
8. permet d'avoir un visuel de la quantité de stock d'un produit si elle existe, sinon il reste vide. Ce phénomène ce produit automatiquement lors d'un choix d'un produit à partir du JCombobox.
9. facilite l'entrée de la date avec un calendrier visible sur 12.

10. un menu déroulant de liste des fournisseurs.
11. un menu déroulant de liste des désignations.
12. calendrier pour l'entrée de la date.

Dans le cas où une désignation d'un produit n'existe pas dans le menu déroulant approprié, le bouton présenté dans 6 permet l'ajout d'une désignation de produit, lors du clic sur ce dernier l'interface présentée par la figure 3.6 apparaît.



id	Famille	Désignation
1	Articles de Bureau e...	papier A4
2	Produit Alimentaires	eau
3	ELECTRECITE	Multiprise
4	Fournitures Informa...	souris

FIGURE 3.6 – Nouvelle désignation.

Dans le cas où un fournisseur n'existe pas dans le menu déroulant approprié, le bouton présenté dans 7 permet l'ajout d'un fournisseur, lors du clic sur ce dernier l'interface présentée par la figure 3.7 apparaît.



Num fournisseur	Nom fournisseur	Contact fournisse...
1	bourouina	2154545
2	benimeur	555555
3	Ifri	23214512
4	libraire waraka	34214511

FIGURE 3.7 – Nouveau fournisseur.

3.4.6 Bon de sortie

Un Bon de sortie n'offre que la manipulation des produits stockés, permettant de choisir le produit et de visualiser sa quantité en stock et son prix, par rapport au bon d'entrée le formulaire de la figure 3.5 ne propose pas l'ajout des produits et fournisseurs.

FIGURE 3.8 – Bon de sortie.

1. c'est un JCheckbox, qui nous propose un choix concernant la nature du bon de sortie par transfert.
2. c'est un JCheckbox, qui nous propose un choix concernant la nature du bon de sortie par consommation.
3. retourne la quantité en stock par rapport au produit sélectionné dans le JComboBox dans ce cas ci en remarque que le produit sélectionné est (papier A4) et sa quantité en stock est de (80).
4. c'est un bouton d'enregistrement du bon de sortie.
5. c'est un bouton de modification dont la manière de fonctionner est identique à celle expliquée au par avant dans la figure 3.5 du formulaire de bon d'entrée.
6. offre un choix d'une date grâce à un calendrier.

Après avoir effectué un bon de sortie d'un produit le système vérifie si sa quantité minimum dans le stock est atteinte, si c'est le cas, une alerte s'affiche automatiquement en demandant un ravitaillement de ce produit, présentée sur la figure 3.9 ci-après :

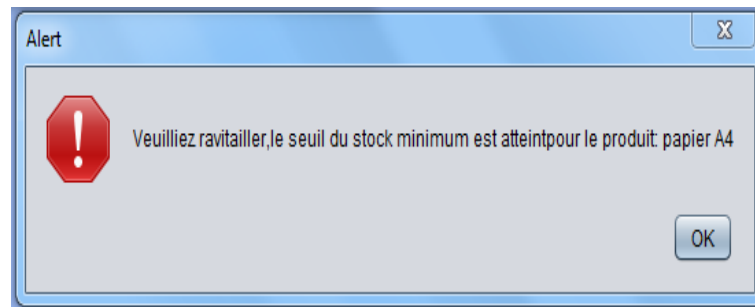


FIGURE 3.9 – Alerte Stock minimum.

3.4.7 Etat du stock

On a proposé un état de stock qui retourne les quantités actuelles en stock de chaque produit, enrichi par un filtre, champ de recherche et représentation graphique des stocks, présentés sur la figure 3.10 ci-après :

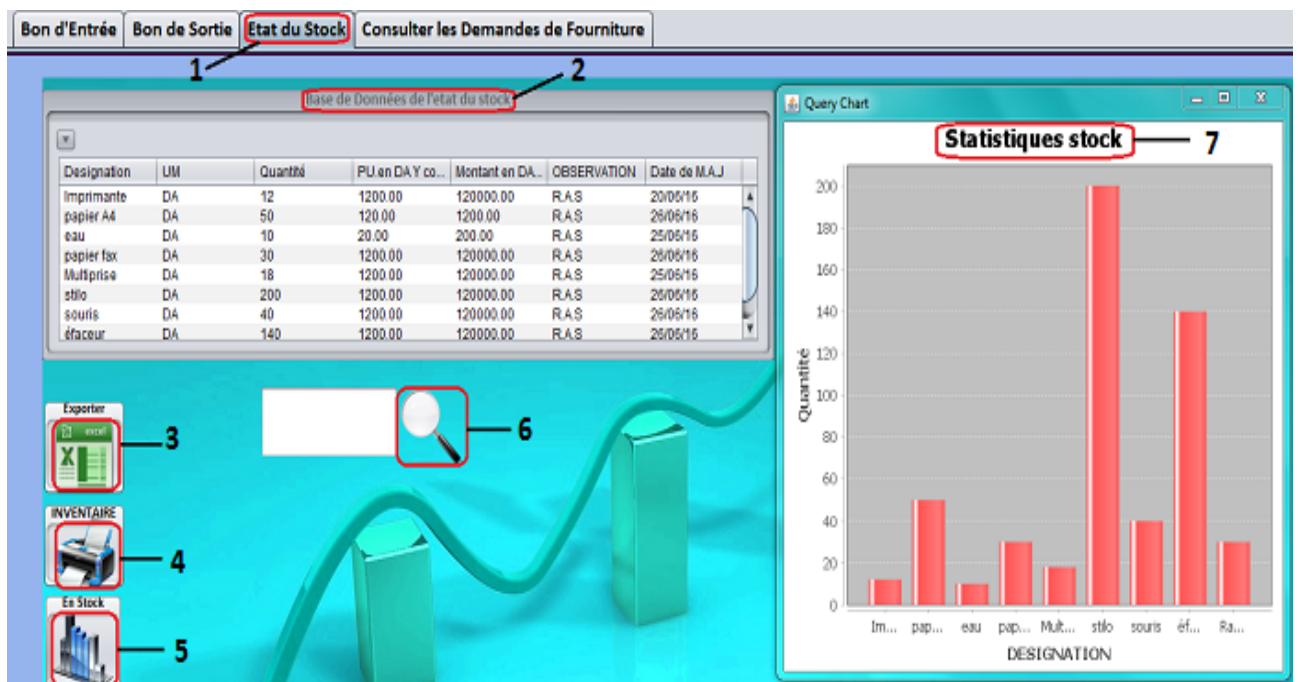


FIGURE 3.10 – Etat du stock.

1. représente l'onglet de l'état du stock.
2. c'est une table qui retourne l'état du stock de chaque produit , elle a été générée grâce à l'interaction entre les entrées et les sorties .
3. c'est un bouton qui permet l'exportation de la table vers un fichier Excel.
4. c'est un bouton pour l'impression de l'inventaire à un moment donné.
5. c'est un bouton qui permet l'interprétation de la table par un graphe (histogramme).
6. effectuer une recherche.
7. une interprétation de la table par un graphe (histogramme).

Dans le cas où un produit contenu dans le stock est arrivé à sa date de péremption l'alerte suivante présentée sur la figure 3.11 s'affiche :

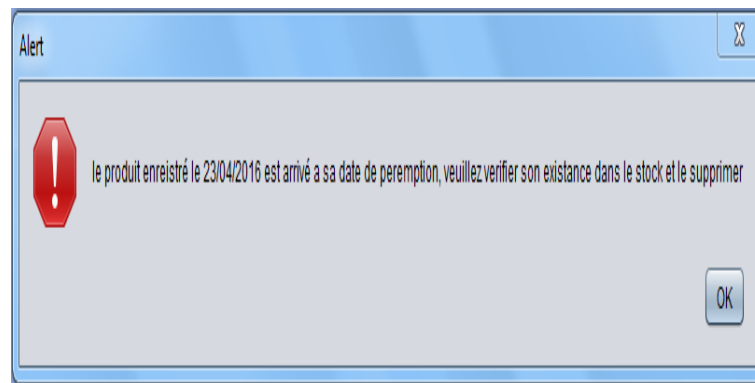


FIGURE 3.11 – Alerte péremption.

3.4.8 Interface du magasinier

La figure qui suit présente l'interface d'accueil dans le cas de l'authentification du magasinier, on a mis en évidence 2 boutons qui sont : le bouton fiche manuelle et le bouton mouvement de stock présenté sur la figure 3.12 ci-après :



FIGURE 3.12 – Interface du magasinier

Le bouton en (1) sur la figure 3.12 nous retourne ce qui suit : La fiche manuelle : est une fiche utilisée par le magasinier afin de garder une trace des mouvements de chaque produit (à chaque produit sa fiche), qui est imprimable en forme de rapport PDF, cette fiche n'est pas connectée aux tables de bons d'entrées ou sorties, et son existence est due à la prévention de l'absence du chef magasin, le mouvement devra être répertorié manuellement jusqu'au retour du chef magasin, cette fiche est présentée par la figure 3.13.

L'index en (2) de la figure 3.13 permet de faire un choix de famille de produit, lors du choix les désignations des produits concernant cette famille apparaissent comme le montre la figure 3.14.

les index sur la figure 3.13 représentent :

1. permet de faire un retour en arrière pour refaire les choix, soit de famille ou de la désignation du produit.
3. c'est un bouton qui permet la validation des choix.
4. c'est un bouton pour une insertion de date grâce à un calendrier.
5. c'est un bouton qui est prévu pour l'ajout d'une désignation de produit.
6. c'est un bouton pour l'ajout d'une ligne dans la table indexée en 11.
7. c'est un bouton pour la modification d'une ligne dans la table indexée en 11.
8. c'est un bouton pour la suppression d'une ligne de la table indexée en 11.
9. nous permet la génération d'un rapport PDF afin de faciliter son impression comme montré dans la figure 3.15 ci-après.
10. nous aide à la recherche dans la table.

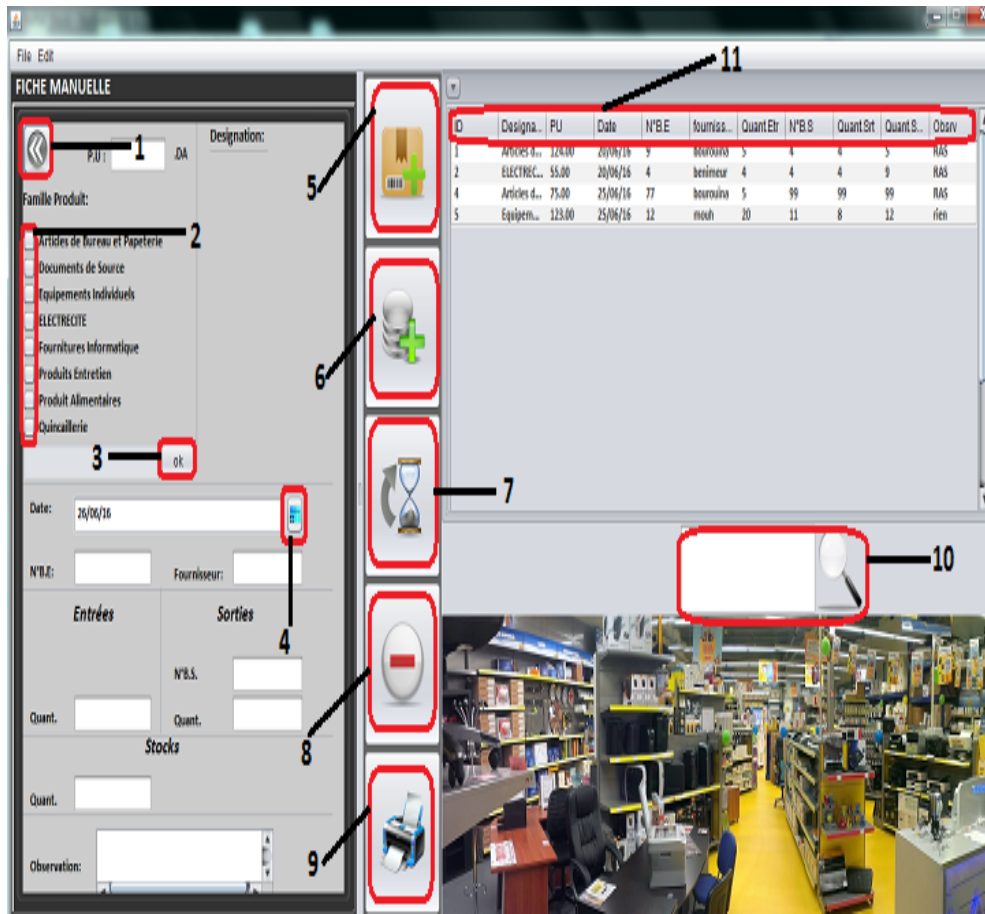


FIGURE 3.13 – Fiche Manuelle.

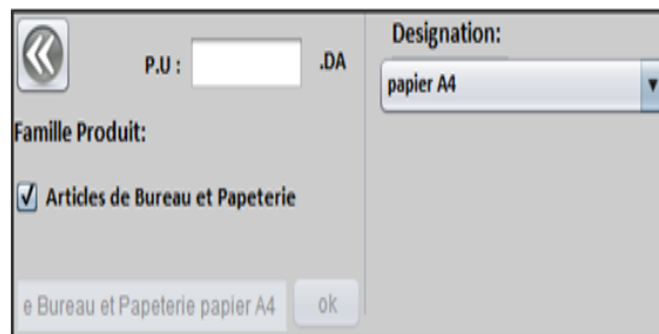


FIGURE 3.14 – Désignation des produits

Date	N°B.E	fournisseur	Q.E	N°B.S	Q.S	STOCK	obsv
20/06/16	9	bourouina	5	4	4	5	RAS
20/08/16	4	benimeur	4	4	4	9	RAS
25/06/16	77	bourouina	5	99	99	99	RAS
25/06/16	12	mouh	20	11	8	12	rien

FIGURE 3.15 – Impression de la fiche manuelle

3.4.9 Fiche mouvements de stock

Si le magasinier choisit le bouton de mouvement lors de l'apparition de sa fenêtre d'accueil, l'interface concernée par le mouvement de stock généré avec les entrées et sorties entre 2 dates apparaît dans la figure ci-dessous, et cela en mettant en évidence les lignes en entrées en bleu, on a également proposé un visuel graphique du mouvement de stock, et une exportation de la table vers un fichier Excel.

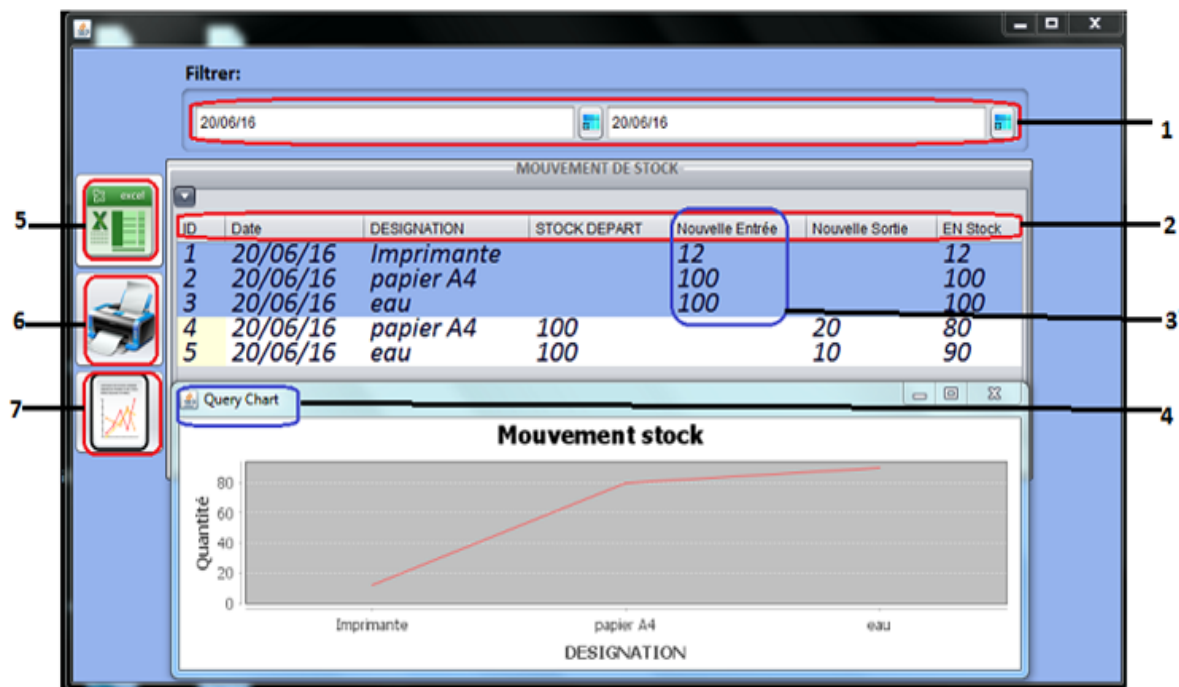


FIGURE 3.16 – Fiche mouvements de stock

1. on propose un filtre avec le choix de la date qui permet de retourner le mouvement entrées et sorties entre ces deux dates.
2. présente la table de mouvement de stock qui est déduite par l'interaction des bons d'entrées et sorties.
3. représente les entrées, on remarque une coloration des lignes des entrées en bleu afin de faciliter la différence entre les entrées et les sorties.
4. représentation du graphe de la table.
5. bouton permettant l'exportation de la table vers Excel.
6. bouton permettant l'impression de la table mouvement.
7. bouton permettant la génération du graphe présenté en 4.

3.5 Conclusion

Dans ce chapitre, nous avons présenté la partie implémentation de notre projet, basée sur la conception présentée dans le chapitre précédent. Nous avons essayer de répondre aux attentes de notre application, on a automatisé les différents bons et fiches. L'application permet aux différents acteurs de l'entreprise (magasinier, chef magasin, chef service et un administrateur) d'avoir un compte et d'intégrer avec cette dernière. Des projections de quelques interfaces de notre application ont été illustré et expliqué.

CONCLUSION GÉNÉRALE

L'objectif de ce travail était de proposer une solution fonctionnelle et adéquate pour la gestion de stock du magasin NAFTAL. La solution proposée devrait répondre aux critères de gestion et faciliter toutes sortes d'interactions entre les entrées et sortie de stock, mais aussi de générer des états imprimables tels que fiche inventaire et fiche de mouvement de stock,etc, ainsi qu'une visualisation globale et évidente de chaque produit.

Pour ce faire, nous avons suivi une démarche conceptuelle qui a été une solution théorique pour le problème, cette solution se démarque grâce aux diagrammes UML, offrant une base solide sur la quelle on s'est appuyé afin de passer à la partie réalisation du projet.

Pour la dernière partie du projet, qui a eu pour but : la réalisation de l'application, on a recours à la programmation avec le langage java, et le SGBD MySql sous l'IDE Netbeans. Ces outils nous ont permis de réaliser facilement la partie IHM de l'application. Notre solution propose la création et les mis à jours des produits et des fournisseurs. Entre autre, notre application donne la possibilité de faire des statistiques sur les différentes produits et leur mouvements de stock. Comme notre application permet aussi de générer automatiquement des alertes sur le stock minimum et les dates de péremption des produits.

Perspective :

On souhaite implémenter des :

- Contrôle sur la saisie des valeurs nulles et négatives.
- Formules de calcul du prix moyen pondéré.
- Alertes sur le rapprochement des dates de péremption des produits.
- Générer automatiquement des bons de commande si un produit atteint le stock d'alert.

BIBLIOGRAPHIE

- [1] Laurena MARQUIS, "Mise en place d'une gestion de stock au sein d'une entreprise d'aménagement paysager", AgroCompus-Ouest, CFR Angers, 12 septembre 2013.
- [2] Mohand Saddek BAHLOUL, "Gestion des stocks et approvisionnements", Maison d'édition pour l'enseignement et la formation Eurl Pages Bleues Internationals, 2011.
- [3] John SNOW, "Indicateurs de suivie et d'évaluation de performance des systèmes logistiques", USAID/Projet Deliver, juillet 2007.
- [4] Abdellatif ABID, "Management des stocks-La tenue de la gestion des stocks", ESCA école de management, Decembre 2009.
- [5] Marlet RENAUD, "Génie logiciel-cycle de vie", LABRI—INRA, Avril 2007.
- [6] Lefrançois ERIC, "Génie logiciel", Ecole d'Ingénieurs de l'Etat de Vaud, Septembre 2006.
- [7] Mohammed Amine MOSTEFAI, Sofiane BATATA, "cours 2 cycle de vie logiciel", ESI, 2003.
- [8] Xavier BLANC, Isabelle MOUNIER, "UML2 pour les développeurs", Eyrolles, 2006.
- [9] Robert OGOR, "Modélisation avec UML", Université ENST Bretagne, 2003.
- [10] Pascal ROQUES, "Les cahiers du programmeur UML2 Modéliser une application web", Eyrolle, 2007.
- [11] Claude BELLEIL, "UML2.0 Diagramme de contexte", université de Nantes, 2003.
- [12] Audibert LAURENT, "UML2 de l'apprentissage à la pratique", Ellipses, 2009.
- [13] Sarah COHEN, "Le modèle relationnel", Université Pris-Sud 11, Orsay, Téléchargé en 2015.
- [14] Luke WELLING, Laura THOMSON, "PHP MySQL", Pearson Education France, 2009.
- [15] Cyrille HERBY, "Apprenez à programmer en Java", site du zéro, juillet2015.
- [16] Philipe RIGAUX, "Pratique de MySQL et PHP", DUNOD, 2009.

Webographie :

- [site01] <http://www.logistiqueconseil.org/> (dernière date de consultation : mai 2016).
- [site02] <http://sabricole.developpez.com/uml/tutoriel/unifiedProcess/> (dernière date de consultation : mai 2016).
- [site03] <https://fr.wikipedia.org/> (dernière date de consultation : mai 2016).
- [site04] <http://www-igm.univ-mlv.fr> (dernière date de consultation : mai 2016).
- [site05] <http://www.jmdoudoux.fr> (dernière date de consultation : mai 2016).
- [site06] <http://jmdoudoux.developpez.com/cours> (dernière date de consultation : mai 2016).
- [site07] <http://www.institut-numerique.org/chapitre-i-presentation-de-naftalspa> (dernière date de consultation : mai 2016).
- [site08] <https://www.google.dz/Organigramme+du+district+GPL> (dernière date de consultation : mai 2016).

ANNEXE

Dans ce qui suit, on présente les documents que NAFTAL nous a légués, afin de les informatiser.

- **Bon d'entrée** : Les informations principales qui le compose sont :
 - La désignation du produit à enregistrer dans le stock.
 - L'unité de mesure.
 - La quantité en entrée.
 - Le prix unitaire du produit en dinars y compris les droits de doine.
 - Le montant en dinars.
- **Demande de fourniture** : C'est la demande rédigée par un chef de service, dans celle-ci, il précise la désignation du produit ainsi que la quantité dont il a besoin.
A la réception de la demande par le chef magasin, il lui attribue un bon de sortie où il précise la quantité fournie dans la demande de fourniture.
- **Bon de sortie** : C'est un document joint pour chaque demande de fourniture, où est spécifiée la désignation du produit et la quantité fournie.
- **Fiche mouvement de stock** : Elle permet au magasinier d'avoir un visuel de l'état du stock sur chaque produit, et ainsi avoir le contrôle sur les mouvements.