

République Algérienne Démocratique et Populaire Ministère de
l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane MIRA Béjaïa



Faculté des Sciences Exactes Département d'Informatique

Mémoire de fin de cycle

En vue de l'obtention du diplôme de MASTER professionnel en

Informatique

Option : Génie logiciel

Thème

**Conception et réalisation d'une solution de gestion
des stocks pour l'école ESTIN**

Réalisé par :

ABBAS Melissa

Devant le jury composé de :

Président :	Mr. ATMANI Mouloud
Examineur :	Mr. DJEBARI Nabil
Encadrant :	Mr. FARAH Zoubeyr
Encadrant d'entreprise :	Mr. OUARET Tawfik

Année 2022/2023

REMERCIEMENT

Je voudrais tout d'abord remercier ALLAH le tout puissant qui m'a donné le courage et la patience pour réaliser ce modeste travail.

J'aimerais exprimer ma reconnaissance envers mon encadrant, Mr. Zoubeyr FARAH, pour ses précieux conseils et son temps précieux qu'il m'a accordé tout au long de cette période.

Je souhaite également exprimer ma gratitude envers mon encadrant de stage, Mr. Tawfik OUERET, pour son accueil, sa confiance, son encadrement et son partage d'expérience durant cette période de stage. Ses conseils éclairés ont été d'une grande aide dans la réalisation de ce projet.

Je souhaite exprimer ma gratitude envers Mr. Abdelkamel TARI pour m'avoir donné l'opportunité de réaliser mon stage au sein de l'école ESTIN. De plus, je tiens à remercier sincèrement Mme Nassima Malki pour sa disponibilité constante et son partage d'informations tout au long de ce projet.

Mes remerciements vont également à l'ensemble du personnel de l'école ESTIN pour leur accueil chaleureux et leur soutien constant.

Je souhaite exprimer ma gratitude envers les membres du jury pour leur intérêt et leur acceptation d'évaluer mon travail.

Enfin, je tiens à remercier toutes les personnes qui ont contribué, de près ou de loin, à l'élaboration de ce mémoire. Leurs aides, leurs conseils et leurs soutiens ont été d'une valeur inestimable, et je leurs suis extrêmement reconnaissante

DEDICACE

Je dédie ce modeste travail à :

À mes parents, les deux êtres les plus chers à mon cœur, qui ont toujours été à mes côtés pour me soutenir et m'encourager à persévérer sans jamais abandonner. Votre amour, votre soutien inconditionnel et vos précieux conseils ont été les fondations de ma réussite. Que Dieu vous protège.

À mes frères, Yazid et Fouad, qui ont été mes compagnons de route, mes alliés et mes sources d'inspirations. Votre présence et votre soutien indéfectibles m'ont donné la force et la motivation nécessaires pour surmonter tous les obstacles.

À mes grands-parents, à qui je rends hommage et dont la mémoire reste gravée dans mon cœur. Leurs foix en moi et leurs enseignements de persévérance et de travail acharné continuent de me guider.

À mes amis, en particulier Fairouz et Nadine, qui ont été à mes côtés tout au long de ce parcours. Vos encouragements, votre soutien moral et nos moments de partage resteront gravés dans ma mémoire.

Et à ma famille ainsi qu'à tous ceux que j'aime profondément et que je n'ai pas cités ici, je vous adresse ma reconnaissance et mon amour sincères.

Que ce travail dédié témoigne de ma reconnaissance envers vous tous.

TABLE DES MATIERES

Table des matières.....	I
Liste des figures.....	VI
Liste des tableaux.....	VII
Liste des abréviations.....	IX
1. Introduction générale.....	1

Chapitre 01 : Généralités sur la gestion de stock

1.1 Introduction.....	3
1.2 Définition des concepts.....	3
1.2.1 Stock.....	3
1.2.2 Niveaux de stock.....	3
1.2.3 Gestion des stocks	4
1.2.4 Objectif de la gestion de stock.....	4
1.2.5 Procédure de la gestion des stocks.....	4
1.2.6 Documents de la gestion des stocks.....	5
1.3 Solutions informatique de gestion des stocks.....	6
1.3.1 Types de solutions.....	6
1.3.2 Objectifs.....	6
1.3.3 Solutions existantes.....	6
1.4 Conclusion	7

Chapitre 02 : Etude de l'existant

2.1 Introduction.....	8
-----------------------	---

2.2	Présentation de l'organisme d'accueil.....	8
2.3	Missions.....	8
2.4	Organigramme de l'école ESTIN.....	9
2.5	Présentation des services d'accueil.....	9
2.5.1	Le service des moyens de l'inventaire et des archives de l'école ESTIN.....	9
2.5.2	Le centre des systèmes et réseaux d'information et de communication, de télé-enseignement et d'enseignement à distance.....	10
2.6	Étude de l'existant informatique.....	10
2.6.1	OpenTalents.....	10
2.6.2	Architecture d'OpenTalents.....	11
2.7	Contexte du système.....	12
2.8	Etude des Procédures de travail liée à la gestion des stocks.....	12
2.8.1	Gestion de stock du magasin principal.....	12
2.8.2	Gestion de stock du magasin Informatique.....	13
2.9	Problématique.....	14
2.10	Solution proposée.....	14
2.11	Conclusion.....	15

Chapitre 03 : Analyse des besoins & Conception

3.1	Introduction.....	16
3.2	Méthodologie de développement.....	16
3.3	Processus Unifié (Unified Process).....	16
3.3.1	Définition.....	16
3.3.2	Caractéristiques du Processus Unifié.....	17
3.3.3	Cycle de vie du Processus unifié.....	18
3.4	Langage UML.....	22
3.4.1	Définition.....	22
3.4.2	Avantages d'UML.....	22

3.4.3 Inconvénients d'UML.....	22
3.4.4 Les diagrammes UML.....	22
3.5 Expression des besoins	26
3.6 Acteurs du système.....	28
3.6.1 Identification des acteurs	28
3.6.2 Héritage des acteurs	29
3.7. Cas d'utilisation du système	29
3.7.1 Identification des cas d'utilisation.....	29
3.7.2 Descriptions textuelles des cas d'utilisation.....	31
3.7.3 Diagrammes des cas d'utilisations.....	36
3.8 Diagrammes de séquence.....	39
3.9 Diagrammes de séquence Détaillés	47
3.10 Diagramme de classes	52
3.11 Glossaire des données.....	53
3.12 Modèle relationnel.....	58
3.12.1 Règles de passage au modèle relationnel	58
3.12.2 Passage au modèle relationnel	59
3.13 Conclusion.....	60

Chapitre 04 : Réalisation

4.1 Introduction	61
4.2 Langages utilisés.....	61
4.2.1 HTML5	61
4.2.2 CSS 3.....	61
4.2.3 JavaScript.....	61
4.2.4 Python	62
4.3. Frameworks.....	62

4.3.1 Bootstrap	62
4.3.2 Django.....	62
4.3.3 JQuery	63
4.4 L'architecture MVT.....	63
4.5 ORM.....	64
4.5.1 SQLite.....	64
4.6. Outils de développement.....	65
4.6.1 Pycharm.....	65
4.6.2 TablePlus	65
4.6.3 Visual Pradigm	65
4.7 Architecture de déploiement de la solution.....	66
4.9 Organisation globale de l'application.....	66
4.10 Présentation de quelques interfaces.....	67
4.10.1 Interface d'accueil	67
4.10.2 Interface ajout d'un produit.....	67
4.10.3 Interface liste des produits.....	68
4.10.4 Interface importer des produits	69
4.10.5 Interface détail d'un produit	69
4.10.6 Interface ajouter un bon d'entrée	70
4.10.7 Interface liste des bons d'entrées	71
4.10.8 Interface détail d'un bon d'entrée	71
4.10.9 Interface établir un bon de commande.....	72
4.10.10 Interface traitement d'un bon de commande.....	73
4.10.11 Interface d'administration.....	74
4.10.12 Interface du Dashboard	75
4.11 Conclusion.....	76

5. Conclusion générale.....	77
6. Bibliographie.....	78

LISTE DES FIGURES

Figure 1 Organigramme de l'école ESTIN	9
Figure 2 Architecture d'Open Talents	12
Figure 3 les 4+1 vues du processus unifié.....	17
Figure 4 Déroulement du Processus UP	18
Figure 5 Cycle de vie du Processus Unifié.....	19
Figure 6 Axes de modélisation.....	23
Figure 7 Héritage des acteurs.....	29
Figure 8 Diagramme de cas d'utilisation associé au chef service.....	36
Figure 9 Diagramme des cas d'utilisation associé à l'administrateur	37
Figure 10 Diagramme des cas d'utilisation associé au superviseur	37
Figure 11 Diagramme des cas d'utilisation associé au magasinier	38
Figure 12 Diagramme de séquence ajouter un produit	39
Figure 13 Diagramme de séquence importer des produits.....	40
Figure 14 Diagramme de séquence modifier un produit	41
Figure 15 Diagramme de séquence supprimer un produit.....	42
Figure 16 Diagramme de séquence gestion des produits	43
Figure 17 Diagramme de séquence ajout d'un bon d'entrée	44
Figure 18 Diagramme de séquence établir un bon de commande	45
Figure 19 Diagramme de séquence traitement d'un bon de commande	46
Figure 20 Diagramme de séquence imprimer une fiche de stock	47
Figure 21 Diagramme de séquence détaillé du cas ajouter un produit.....	48
Figure 22 Diagramme de séquence détaillé du cas ajouter un bon d'entrée	49
Figure 23 Diagramme de séquence détaillé du cas établir un bon de commande	50
Figure 24 Diagramme de séquence détaillé du cas traitement d'un bon de commande..	51
Figure 25 Diagramme de classes de gestion des stocks.	52
Figure 26 Architecture MVT	64
Figure 27 Architecture de déploiement de la solution	66
Figure 28 Architecture globale de l'application	66
Figure 29 Interface d'accueil	67
Figure 30 Interface ajout d'un produit	67
Figure 31 Interface liste des produits	68

Figure 32 Interface Importer des produits	69
Figure 33 Interface détail d'un produit.....	69
Figure 34 Interface ajouter un bon d'entrée.....	70
Figure 35 Interface liste des bons d'entrées	71
Figure 36 Interface détail d'un bon d'entrée	72
Figure 37 Interface établir un bon de commande	73
Figure 38 Interface traitement d'un bon de commande	73
Figure 39Interface d'administration	74
Figure 40 Interface Dashboard	75

LISTE DES TABLEAUX

Tableau 1 Phases et itérations du Processus Unifié	20
Tableau 2 Activités du processus (activités de chaque cycle).....	21
Tableau 3 Concept de base du diagramme de cas d'utilisation	24
Tableau 4 Concept de base du diagramme de classe	25
Tableau 5 Concept de base du diagramme de séquence.....	26
Tableau 6 Ensemble des cas d'utilisation.....	31
Tableau 8 Glossaire des données.....	57

LISTE DES ABREVIATIONS

Abréviation	Signification
ERP	Entreprise Ressource Planning
SaaS	Software as a service
ESTIN	École supérieure en Sciences et Technologies de l'Informatique et du Numérique
ESI	École supérieure d'informatique
WSGI	Web Server Gateway Interface
SGBD	Système de gestion de base de données
UML	Unified Modeling Language (Langage de modélisation unifiée)
UP	Unified Process (Processus unifié)
HTML	HyperText Markup Language
CSS	Cascading Style
XML	Extensible Markup Language
W3C	World Wide Web Consortium
JS	JavaScript
MIT	Massachusetts Institute of Technology

URL	Uniform Resource Locator
ORM	Object Relationnel Mapper
DOM	Document Object Model
AJAX	Asynchronous JavaScript + XML
MVT	Modèle-Vue-Template
SQL	Structured Query Language
ACID	Atomique, cohérente, isolée et durable
SysML	Systems Modeling Language
BPMN	Business Process Modeling Notation
OMG	Object Management Group

Introduction générale

Introduction générale

L'informatique représente indéniablement la révolution la plus significative et novatrice ayant impacté la vie de l'humanité moderne. En effet, les logiciels informatiques offrent désormais des solutions à une multitude de problèmes, que ce soit dans le cadre professionnel ou pour des applications personnelles. De plus, les méthodes de conception et de développement des logiciels ont engendré l'émergence de nombreuses technologies qui facilitent leur déploiement tout en leur conférant des capacités et des fonctionnalités toujours plus étendues.

Effectivement, le système informatique actuel offre des capacités avancées pour gérer et résoudre efficacement les problèmes rencontrés dans les entreprises et les organisations.

Dans le domaine de la gestion des stocks, l'intégration de l'informatique permet d'automatiser les processus de suivi, de contrôle et de gestion des stocks. Les logiciels de gestion des stocks permettent de centraliser les données, d'optimiser les niveaux de stock, d'assurer un suivi précis des entrées et des sorties, de générer des rapports analytiques, et même de mettre en place des alertes en cas de seuils critiques. Grâce à l'informatique, les entreprises peuvent améliorer leurs efficacités opérationnelles, minimiser les erreurs et les coûts liés à la gestion des stocks, et prendre des décisions éclairées basées sur des informations en temps réel.

Dans ce contexte, l'École Supérieure en Science et Technologies de l'Informatique et du Numérique (ESTIN), où j'ai effectué mon stage de fin d'études, m'a confié une mission de conception et de réalisation d'un système de gestion de stock.

Actuellement, l'école dispose d'une plateforme de gestion de pédagogie et de scolarité qui permet de gérer les informations liées à la scolarité, mais elle ne comprend pas de module dédié à la gestion des stocks. Notre objectif est donc d'étendre cette plateforme en ajoutant un module de gestion des stocks qui permettra de centraliser et de structurer les informations relatives aux stocks, tout en les mettant à disposition des utilisateurs de manière automatisée. Ainsi, nous visons à fournir à l'école un outil complet qui facilitera la gestion de ses stocks et contribuera à une meilleure organisation de ses ressources.

Ce mémoire est divisé en quatre chapitres, dans lesquels nous exposons la méthodologie de travail que nous avons suivie pour mener à bien le projet. Nous détaillons les différentes étapes de réalisation ainsi que les résultats que nous avons obtenus. Ce dernier est organisé comme suite :

Introduction générale

Chapitre 1 : Généralités sur la gestion des stocks

Ce chapitre porte sur les généralités de la gestion des stocks. Il englobe la définition des termes et concepts clés liés à ce domaine, ainsi que les objectifs et les procédures de gestion des stocks. Il englobe également la présentation de différentes solutions informatiques couramment utilisées dans ce domaine.

En somme, ce chapitre établit les bases nécessaires pour comprendre les enjeux de la gestion des stocks, il servira de point de départ pour les développements ultérieurs sur la conception et la réalisation d'un système de gestion des stocks pour l'ESTIN.

Chapitre 2 : Etude de l'existant

Le deuxième chapitre de ce mémoire, intitulé "Étude de l'existant", se focalise sur l'étude du processus actuel mis en place par l'école pour détecter les anomalies. Nous analyserons ce processus dans le but de mettre en évidence les éventuelles lacunes ou problèmes qui nécessitent des solutions. Cette étude approfondie nous permettra de proposer des améliorations et des solutions adaptées pour optimiser la gestion des stocks dans l'établissement.

Chapitre 3 : Analyse et conception

Le troisième chapitre, intitulé "Analyse et conception", regroupe l'ensemble des étapes de notre processus de développement. En commençant par présenter la méthodologie et le langage de modélisation que nous avons utilisés. Ensuite, la définition des besoins fonctionnels et non fonctionnels auxquels notre application doit répondre, ainsi que l'identification des acteurs et des différents cas d'utilisation. Nous les décrirons à l'aide des diagrammes de séquences afin d'illustrer leurs fonctionnements. Enfin, nous établirons le diagramme de classe qui représentera la structure et les relations entre les différentes entités de notre application. Cette étape d'analyse et de conception sera essentielle pour guider la réalisation du système.

Chapitre 4 : Réalisation

Le quatrième et dernier chapitre est consacré à la réalisation où nous allons définir tous les outils qui nous ont permis de concevoir notre application, quelques interfaces y seront présentées.

Chapitre 01

Généralités sur la gestion des stocks

1.1 Introduction

Dans ce chapitre, nous introduisons les notions fondamentales de la gestion des stocks, en commençant par définir les termes et concepts essentiels liés à ce domaine, afin d'assurer une compréhension solide de la gestion des stocks et de ses enjeux. Nous allons également présenter les objectifs de la gestion des stocks, ainsi que les différentes procédures impliquées dans cette fonction. Ensuite nous explorons les différents systèmes de gestion des stocks qui existent, en illustrant des exemples de solutions informatiques couramment utilisés dans ce domaine.

1.2 Définition des concepts

1.2.1 Stock

Le stock est un ensemble de marchandises ou d'articles accumulés dans l'attente d'une utilisation ultérieure plus ou moins proche qui permet d'alimenter les utilisateurs au fur et à mesure de leurs besoins, sans imposer les délais et les à-coups d'une fabrication ou d'une livraison par les fournisseurs [1].

1.2.2 Niveaux de stock

Les niveaux de stock sont une mesure importante de la quantité de marchandises ou de produits qu'une entreprise a en stock à un moment donné. Le niveau de stock est généralement mesuré en termes de quantité des produits physiques ou de valeur monétaire de ces produits.

Il existe plusieurs niveaux de stock importants à surveiller lors de la gestion de stock, notamment :

- Stock minimum : c'est la quantité correspondante à la consommation pendant le délai de réapprovisionnement.
- Stock de sécurité : c'est la quantité en dessous de laquelle il ne faut pas descendre. Le stock de sécurité peut aider à éviter les ruptures de stock en cas de demande imprévue ou de délais de livraison plus longs que prévu.
- Stock maximum : c'est le niveau le plus élevé de stock, il est déterminé en fonction de facteurs tels que les coûts de stockage, les coûts de commande et les coûts de détention de stock.
- Stock d'alerte : c'est la quantité qui détermine le déclenchement de la commande, elle est calculée en fonction du délai habituel de livraison [2].

1.2.3 Gestion des stocks

La gestion des stocks est le processus de planification, d'organisation et de contrôle des niveaux de stock pour répondre à la demande des clients, tout en minimisant les coûts de Stockage et de commande [3].

1.2.4 Objectif de la gestion de stock

L'objectif principal de la gestion des stocks est d'assurer la disponibilité des produits souhaités au bon moment et au bon endroit. Pour atteindre cet objectif, il est crucial de disposer d'une visibilité claire sur l'état des stocks, permettant de déterminer le moment opportun pour passer une nouvelle commande, la quantité nécessaire à commander, et/ou stocker les produits de manière efficace [4].

1.2.5 Procédure de la gestion des stocks

La gestion des stocks comprend plusieurs étapes qui permettent un suivi précis des mouvements de stock. Parmi celles-ci, on peut citer :

1. La normalisation des articles

Il s'agit de classer les articles selon le degré de leur utilité, de déterminer les articles qui conviennent le mieux aux besoins de l'entreprise et d'éliminer toutes les variétés superflues. Ce classement a pour avantage la réduction du nombre d'articles stockés et par conséquent du niveau général des stocks et de leur coût.

2. La classification des articles

La classification des articles peut se faire selon un critère déterminé : L'origine du produit, sa nature ou sa destination, ce qui peut faciliter l'identification des produits en magasin.

3. La symbolisation des articles

La symbolisation des articles facilite leur classification, elle consiste à attribuer chaque article un symbole ou un code permettant son identification. La symbolisation peut être alphanumérique ou numérique elle est généralement aussi brève que possible. Quand elle est numérique, la symbolisation permet l'utilisation de l'informatique dans la gestion des stocks.

4. Le contrôle des entrées

Les entrées peuvent provenir soit des fournisseurs, soit des autres services de l'organisation (cas des produits finis). Elles doivent être contrôlées quantitativement et qualitativement par le magasinier. Pour les entrées en provenance des fournisseurs, on établit un bon de réception. Pour les entrées provenant des autres services (les ateliers par exemple), on établit un bon d'entrée selon le même principe que le bon de réception, mais dont l'usage est interne. Pour respecter le principe du contrôle mutuel.

5. Le contrôle des sorties

Les sorties sont destinées soit aux services internes de l'entreprise, soit à l'extérieur (vers les clients). Le contrôle est effectué grâce au bon de sortie.

6. Le contrôle des retours

Bien que ce ne soit pas un mouvement normal, le retour au magasin d'articles déjà sortis et enregistrés peut avoir lieu. C'est le cas d'annulation de commande ou d'ordre de fabrication. Le contrôle des retours est généralement effectué à travers un document appelé « bon de retour » comportant les mêmes informations et ayant la même forme que le bon d'entrée.

7. Le contrôle des existants

La fiche de stock est un document qui permet de suivre l'état des stocks de produits. Cette fiche doit être mise à jour à chaque mouvement de stock (entrée ou sortie de produits), elle doit être en mesure d'informer à tout moment le gestionnaire des stocks sur les quantités stockées. Mais parfois les stocks réels ne se concordent pas avec les stocks indiqués par la fiche. Ce qui cause les ruptures des stocks ou le sur stockage. Pour éviter ces erreurs, le contrôle des existants a lieu à travers le dénombrement effectif des articles stockés ou l'inventaire. Celui-ci est imposé par la loi. Il a lieu soit globalement en fin d'année, soit catégorie par catégorie tout au long de l'année. Dans ce dernier cas l'inventaire est dit tournant [5].

1.2.6 Documents de la gestion des stocks

Les documents qui sont généralement utilisés pour la gestion des stocks sont :

- Fiche de stock : c'est un document qui permet de suivre l'état des stocks des produits dans l'entrepôt. Cette fiche doit être mise à jour à chaque mouvement de stock (entrée ou sortie des produits).
- Bon d'entrée : Ce document précise les références, les quantités des articles reçus et les observations éventuelles. C'est un document interne à l'organisation et sert d'instrument de contrôle pour le gestionnaire.
- Bon de sortie : un document qui enregistre les sorties des produits d'un entrepôt ou d'un magasin. Ce document permet de suivre les quantités des produits sorties, la date de sortie, la destination des produits, ainsi que d'autres informations utiles telles que le nom du responsable de la sortie et le motif de la sortie.
- Bon de commande : C'est un document commercial qui doit être rédigé avec clarté et précision afin d'éviter toute contestation ultérieure, Il est utilisé pour passer une commande de produits, on peut distinguer deux types de bon : bon de commande interne ou externe [5].

1.3 Solutions informatique de gestion des stocks

1.3.1 Types de solutions

- Logiciel de gestion de stock : c'est un logiciel autonome qui est uniquement limité à la gestion des stocks.
- Système de gestion de stock basé sur le Cloud : ce système fait référence à une solution informatique de gestion des stocks, qui est hébergée sur des serveurs distants, qui permet aux entreprises d'accéder à leurs données des stocks en temps réel [6].
- Applications mobiles de gestion des stocks : ces applications permettent aux utilisateurs de gérer leurs stocks à partir de leur téléphone portable ou tablette.
- Intégrations ERP : les solutions ERP (Enterprise Resource Planning) peuvent inclure des modules de gestion des stocks pour les entreprises qui nécessitent une gestion plus globale [7].

1.3.2 Objectifs

Les objectifs visés par l'utilisation d'une application de gestion des stocks sont les suivants :

- Gain de temps
- Assurer le suivi des ventes et des niveaux de stock.
- Diminution des manipulations frauduleuses causées par des actions humaines.
- Le maintien d'un équilibre dans le stock.
- La réduction des pertes de données [7].

1.3.3 Solutions existantes

De nombreuses solutions informatiques de gestion des stocks sont disponibles sur le marché, parmi les plus connues on peut citer :

Zoho inventory : est un logiciel de gestion de stock basé sur le Cloud, développé par la société Zoho Corporation. Il est conçu pour aider les petites et moyennes entreprises à gérer leurs inventaires, leurs commandes, leurs expéditions et leurs factures.

Il offre un large éventail de fonctionnalités, notamment la gestion des stocks en temps réel, la gestion des commandes et des factures, la gestion des fournisseurs, la gestion des expéditions, la gestion des retours, la gestion des niveaux de stock minimum et maximum, la gestion des alertes de stock faible, la synchronisation avec les plateformes de commerce électronique telles que Amazon, eBay et Shopify, ainsi que des rapports et des analyses détaillés [8, 9].

Stocknet : est un logiciel en ligne conçu pour les entreprises de tailles moyenne et grande, cette solution est disponible en mode Saas (software as a service) qui propose une gamme de fonctionnalités pour aider les entreprises à gérer leur stock, notamment la gestion des entrées et sorties, la gestion des inventaires, la gestion des commandes et des achats, la traçabilité des lots, la gestion des alertes, ainsi que la génération des rapports.

Le logiciel est accessible via une interface web intuitive et peut être personnalisé en fonction des besoins spécifiques de chaque entreprise [10].

Odoo : est un logiciel ERP open source qui propose une gamme complète de fonctionnalités de gestion, y compris la gestion de stock. Il permet de suivre les niveaux de stock, de gérer les fournisseurs et les commandes, de réaliser des inventaires et de générer des rapports.

La grande flexibilité et modularité d'Odoo permettent aux utilisateurs de choisir les fonctionnalités dont ils ont besoin pour leur entreprise et d'ajouter des modules supplémentaires au fur et à mesure de la croissance de leur entreprise [11].

1.4 Conclusion

Ce chapitre donne une vue d'ensemble des concepts clés de la gestion des stocks, suivi d'une présentation de certaines solutions informatiques de gestion des stocks disponibles sur le marché. L'objectif de ce projet est d'aider l'école ESTIN à résoudre les problèmes liés à la gestion de ses stocks.

Le prochain chapitre comprendra une analyse du système actuel, dans le but de proposer une solution satisfaisante à ces problèmes.

Chapitre 02
Etude de l'existant

2.1 Introduction

La gestion des stocks au sein de l'école ESTIN d'Amizour est un processus particulier qui mérite d'être perfectionné et analysé soigneusement. Avant de proposer une solution informatique, il est essentiel de présenter l'organisme d'accueil en analysant attentivement les pratiques actuelles concernant la gestion des mouvements de stock.

Cette étape est cruciale pour comprendre les procédures de travail en vigueur dans l'établissement afin d'identifier les points faibles et élaborer un projet adapté aux besoins spécifiques de l'école.

2.2 Présentation de l'organisme d'accueil

L'École supérieure en Sciences et Technologies de l'Informatique et du Numérique de Bejaïa a été créée en août 2019, elle est située au campus d'Amizour à 17 km du siège de la wilaya de Bejaïa. Elle possède une architecture impressionnante et dispose de tous les équipements nécessaires pour offrir une formation de haut niveau. Elle est classée comme l'une des grandes écoles au même titre que l'ESI d'Alger et l'ESI de Sidi Bellabas. L'école est pourvue d'instances administratives et scientifiques qui ont pour rôle d'évaluer les activités pédagogiques et scientifiques [12].

2.3 Missions

L'ESTIN a plusieurs missions :

- Offrir des programmes rigoureux qui préparent les étudiants à entrer directement sur le marché du travail ou à poursuivre des études supérieures.
- Mener l'avancement de l'informatique, des technologies de l'information, du numérique et de la cybersécurité par le biais d'une recherche et d'un enseignement reconnu à l'échelle internationale, ainsi que par le transfert de technologie ;
- Préparer les étudiants à participer pleinement et de manière éthique à une société diversifiée et encourager l'apprentissage tout au long de la vie.
- Former les étudiants aux meilleures pratiques du domaine et intégrer les dernières recherches dans le programme d'études.
- Favoriser le développement des compétences en matière de résolution de problèmes et de communication, qui font partie intégrante de la profession.
- Fournir des expériences d'apprentissage de qualité par le biais de pratiques de classe efficaces, de styles d'enseignement actifs et d'opportunités d'interactions significatives entre les étudiants et le corps enseignant.

- Servir de conseiller les responsables de la wilaya de Bejaia et de l'état algérien en ce qui concerne les impacts des technologies du numérique [12].

2.4 Organigramme de l'école ESTIN

Voici l'organigramme de l'école ESTIN qui représente les divers services et fonctionnalités de l'école (voir Figure 1).

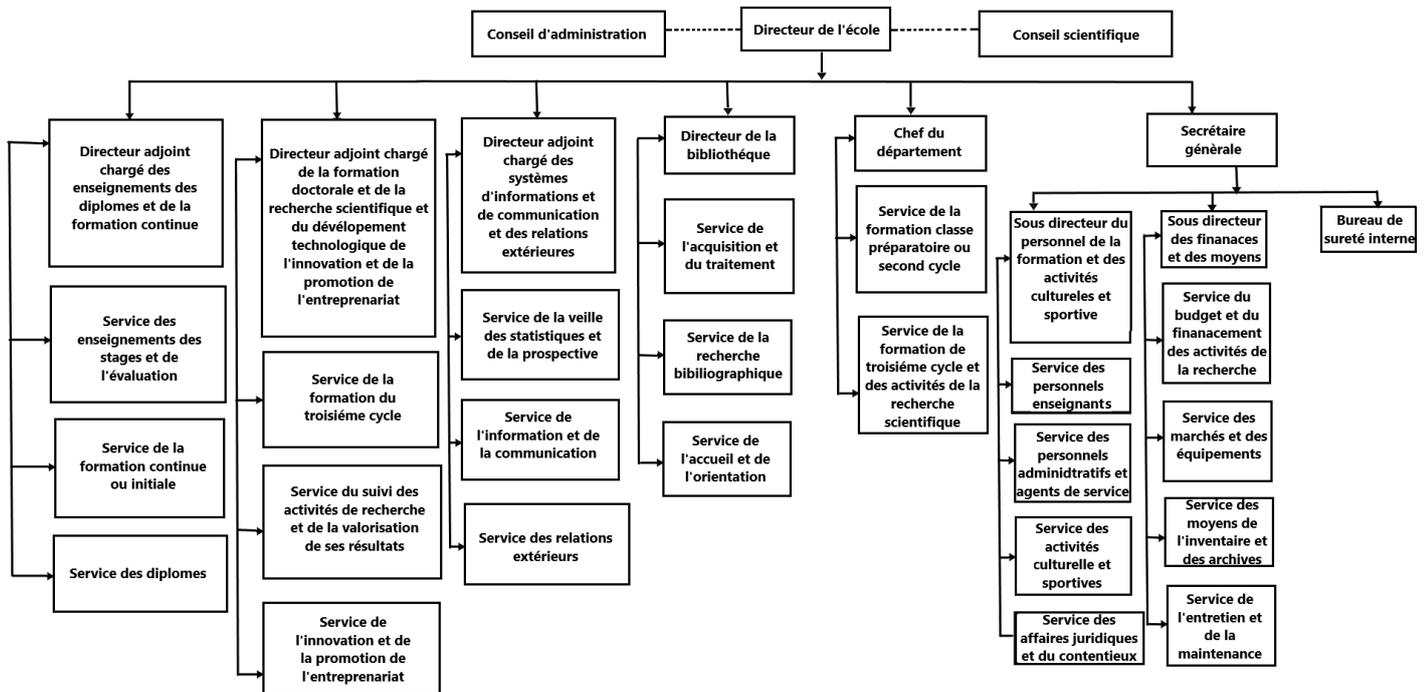


Figure 1 Organigramme de l'école ESTIN

2.5 Présentation des services d'accueil

Les services d'accueil impliqués dans le cadre du projet comprennent, le service des moyens de l'inventaire et des archives et le centre des systèmes et réseaux d'information et de communication, de télé-enseignement et d'enseignement à distance. Ces services jouent un rôle essentiel dans la mise en place du projet en fournissant leurs expertises et leurs ressources dans le domaine.

2.5.1 Le service des moyens de l'inventaire et des archives de l'école ESTIN

Ce service assume la responsabilité de gérer et préserver les ressources et les documents importants de l'établissement. Il est également le service au quel est lié le magasin principal ce dernier est le lieu où sont stockés les biens, les fournitures, les équipements et les matériaux nécessaires au bon fonctionnement de l'école.

Ce service a pour missions :

- Assurer la dotation en moyens de fonctionnement des structures et des services de l'école.
- Assurer la gestion de stocks du magasin principale.
- Veiller à la bonne tenue des registres et des opérations d'inventaires.
- Établir des procédures de gestion des archives pour assurer leur sécurité, leur accessibilité et leur confidentialité [12].

Dans le cadre de ce projet, notre intérêt se concentre sur le magasin principal

2.5.2 Le centre des systèmes et réseaux d'information et de communication, de télé-enseignement et d'enseignement à distance

Etant un service technique de l'école, il comprend trois sections distinctes. Pour ce projet, nous mettrons l'accent sur deux sections spécifiques :

1. section systèmes : Responsable du développement des applications et des programmes informatiques. Son objectif principal est de concevoir, créer et mettre en œuvre des solutions logicielles répondant aux besoins spécifiques des utilisateurs internes. Il travaille en étroite collaboration avec les différents départements pour comprendre leurs exigences et proposer des solutions informatiques adaptées.

2. section réseaux : Est chargé de la gestion de l'ensemble des moyens informatiques liés au réseau. Il a pour mission de garantir le bon fonctionnement des équipements informatiques, d'effectuer des évolutions techniques et de veiller à la sécurité du réseau. Il est également responsable de la mise en place de nouvelles infrastructures réseau, de la maintenance des équipements existants et de la résolution des problèmes liés aux connexions et aux communications. la gestion du magasin informatique fait partie des responsabilités de cette section également, cela implique le traitement des demandes de matériel, la gestion des stocks et l'inventaire des matériels [12].

2.6 Plateforme de l'ESTIN

L'ESTIN dispose de plusieurs plateformes, parmi lesquelles se trouve celle-ci :

2.6.1 OpenTalents

OpenTalents Est une plateforme numérique qui permet la gestion de la pédagogie et de la scolarité dans les établissements d'enseignement. Elle est une évolution de la plateforme ESI Talents utilisée auparavant uniquement par l'École nationale Supérieure d'Informatique (ESI)

d'Alger. L'objectif d'OpenTalents est de permettre le partage et l'utilisation de la solution dans d'autres établissements d'enseignement, en dissociant les règles de gestion de l'ESI de la solution, et en mettant le code d'OpenTalents aux normes de développement open-source. Cela facilitera la contribution et le développement en interne à l'ESI ou par d'autres contributeurs externes, sous une licence appropriée.

Pour faciliter l'utilisation et le développement, de nombreux aspects sont abordés, dont l'automatisation du déploiement, de l'installation, la sauvegarde, la documentation, etc.

Actuellement, la plateforme est déployée avec succès dans plusieurs établissements d'enseignement en Algérie, notamment l'ESI d'Alger, l'ESTIN de Béjaïa et la Faculté de Pharmacie de l'Université Alger 1 [13].

2.6.2 Architecture d'OpenTalents

Une instance d'OpenTalents est constituée de quatre conteneurs :

- Le conteneur de l'application en elle-même :

Il s'agit du conteneur contenant le code de l'application programmée en Python/Django. Ce même conteneur abritera l'exécution des calculs. Une interface WSGI (Web Server Gateway Interface) est exposée à partir de ce conteneur grâce au module Gunicorn, interface qui sera le point d'entrée au conteneur web Nginx décrit par la suite.

- Le conteneur du SGBD (système de gestion de base de données) :

Il s'agit du conteneur du SGBD Postgres. Qui est reconnu performant pour l'exécution des requêtes concurrentes et complexes.

- Le conteneur du serveur web :

Il s'agit du conteneur du serveur Nginx, remplaçant ainsi le serveur web Apache. Ce conteneur permet d'avoir une interface WSGI avec le conteneur de l'application grâce au module Gunicorn. Il s'agit du conteneur dont le port est accessible depuis le serveur physique et donc c'est le point de communication d'une instance OpenTalents avec le serveur physique puis le réseau extérieur.

- Le conteneur du cache :

Ce conteneur permet la cohérence du cache pour certains modules utilisés par l'application, il exploite la technologie Redis [13].

La figure suivante illustre l'ensemble des technologies de la plateforme (voir Figure 2).

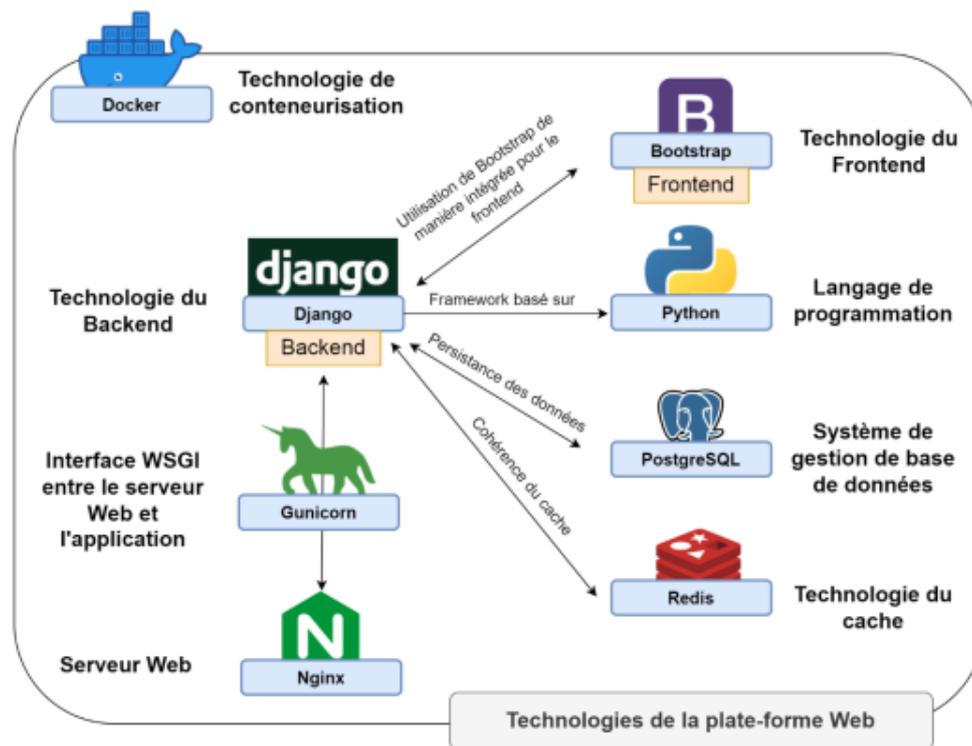


Figure 2 Architecture d'Open Talents

2.7 Contexte du système

Afin de concevoir et mettre en place un système de gestion des stocks pour l'école, il était essentiel pour moi de collecter les informations nécessaires auprès des experts du domaine. Après avoir structuré les informations recueillies, j'ai constaté que la majorité des activités du magasin principal sont centrées sur deux opérations principales à savoir la réception des produits facturés par les fournisseurs et la réception des bons de commande internes établis par les services demandeurs.

Pour le magasin informatique, les activités principales consistent en la réception des produits commandés auprès du magasin principal et la réception des bons de commande internes. Dans la suite de ce projet, nous présenterons en détail la manière dont ces opérations sont traitées par le magasinier, de leur réception à leur achèvement.

2.8 Etude des Procédures de travail liée à la gestion des stocks

2.8.1 Gestion de stock du magasin principal

- A l'arrivée de la facture du fournisseur, ainsi que la marchandise commandée, le magasinier effectue plusieurs vérifications pour s'assurer de leur conformité avec le bon de commande. Ensuite, il procède à l'enregistrement des informations des produits reçus en établissant un bon d'entrée contenant diverses informations telles que son numéro et sa date d'établissement, le numéro de la facture, le nom et prénom du fournisseur. Ensuite il note dans

chaque ligne de bon, la désignation du produit réceptionné, la quantité livrée, le prix unitaire, un numéro d'inventaire s'il s'agit d'un produit non-consommable ou dépasse le prix de 3000DA et le montant total avec toutes Taxes Comprises. Enfin le magasinier appose sa signature sur ce document avant de l'archiver dans un registre dédié.

- Les services peuvent commander des produits au magasin en établissant un bon de commande (interne) qui sera adressé au magasinier, à l'établissement du bon de commande (interne) le demandeur mentionne la date de l'établissement, son nom, prénom, sa fonction ainsi le service demandeur. Ensuite, il mentionne les produits à demander en précisant pour chacun : la désignation du produit et la quantité demandée. Et, il l'envoie au magasinier après que le bon soit signé par le responsable de service.

- A la réception, le magasinier traite la commande selon l'état de stock, et il affecte pour chaque produit demandé une quantité à accorder, la quantité servie et une observation. Ensuite il transmet le bon de commande au secrétaire générale pour le signé.

- A la livraison des produits le magasinier établit un bon de sortie sur lequel il saisit son numéro, sa date d'établissement ainsi le numéro de bon de commande et il note dans chaque ligne de bon la désignation du produit, la quantité servie, un numéro d'inventaire et l'affectation si le produit est non consommable ou dépasse 3000 DA. Enfin le magasinier saisie son nom et prénom ainsi le nom, prénom et la fonction du bénéficiaire et signe le bon, ce dernier est gardé dans un registre et une copie est transférée au secrétaire générale.

- Et pour les suivis de l'état de stock, le magasinier assigne pour chaque produit une fiche de stock.

2.8.2 Gestion de stock du magasin Informatique

Le responsable du magasin établit un bon de commande interne en générale au magasin principal pour recevoir tous le matériel informatique. Le responsable du magasin transfère tous le matériel informatique à ce magasin, Ce dernier a son tour gèrera son propre stock de la même manière que le magasin principal.

2.9 Problématique

Le système d'information actuel de l'école ESTIN ne prend pas encore en charge l'automatisation du système de gestion des stocks par conséquent :

- Les tâches et les procédures administratives liées au contrôle et à la gestion des mouvements de stock sont réalisées manuellement, ce qui peut occasionner l'oubli, la perte de temps.
- Des problèmes liés à l'archivage sur papier sont rencontrés tels que la perte de documents et le manque d'espace de stockage.
- Difficulté pour effectuer des mises à jour des dossiers.
- Difficulté pour effectuer une recherche dans les archives.
- Création d'un environnement de travail perturbé.
- Difficulté de suivi et manque de visibilité des mouvements de stock.
- Manque d'efficacité dans la gestion des inventaires.
- Difficulté à suivre les niveaux de stock en temps réel.
- Risque de rupture de stock.

Ces derniers ont de graves répercussions sur la qualité du travail et peuvent avoir un effet négatif sur les résultats attendus de l'organisation.

2.10 Solution proposée

Pour pallier les problèmes auxquels sont confrontés les magasiniers de l'école ESTIN, la section systèmes liée au centre des systèmes et réseaux d'information et de communication, de télé-enseignement et d'enseignement à distance nous a proposé de développer un module de gestion de stock sur la plateforme Open Talents. Ce module vise à :

- Simplifier les tâches administratives liées au contrôle et à la gestion des mouvements de stock.
- Permettre au magasinier de suivre en temps réel les entrées et les sorties de stock, ainsi que les niveaux de stock actuels.
- Générer des rapports automatisés sur les mouvements de stock, ce qui facilitera la prise de décision.
- Permettre une gestion plus efficace des commandes.
- Faciliter la traçabilité des produits.
- Automatiser les tâches traitées manuellement.

Cette solution sera conçue pour être facile à utiliser et intuitive, afin que les utilisateurs puissent rapidement s'adapter au nouveau système sans avoir besoin d'une formation intensive.

De plus, puisque c'est une plateforme web, elle permettra aux utilisateurs d'y accéder depuis n'importe quel ordinateur connecté à internet, ce qui augmentera leur flexibilité et leur efficacité.

2.11 Conclusion

Ce chapitre nous a permis d'obtenir une vue détaillée sur l'état actuel de l'organisme d'accueil, et de nous familiariser avec les différentes activités et traitements qui se font au sein des magasins. Ce diagnostic a permis d'identifier les insuffisances et les manques résultant des méthodes de travail traditionnelles, qui ont un impact négatif sur le fonctionnement optimal du système.

Suite à l'étude approfondie de l'état actuel, des recommandations ont été émises afin d'améliorer la gestion et l'organisation des magasins en automatisant le système.

Cette étape sera le point de départ d'une étude détaillée et une conception de la solution proposée, qui sera présenté dans le prochain chapitre en essayant de répondre aux objectifs énoncés et ainsi satisfaire les souhaits des utilisateurs.

Chapitre 03
Analyse des besoins & Conception

3.1 Introduction

Ce chapitre met en évidence les étapes fondamentales du processus de développement d'un projet informatique. La première étape est l'analyse des besoins, qui implique l'expression des besoins fonctionnels et non fonctionnels du système, ainsi que la définition du rôle de chaque acteur qui interagit avec le système. Cette étape est suivie de la structuration de ces rôles à l'aide d'un diagramme de cas d'utilisation.

La seconde étape est la conception du projet. Son objectif principal est de traduire les besoins identifiés lors de l'analyse en une solution concrète et efficace. Cette phase est organisée en plusieurs sous-étapes, telles que l'établissement de diagrammes de séquence et d'interaction, ainsi que la création d'un diagramme de classe.

Avant d'entrer dans les détails de ces étapes, il est essentiel de présenter la méthodologie et le langage de modélisation utilisés pour mener à bien le projet. Dans ce contexte, le Processus Unifié se révèle être la solution idéale, avec l'utilisation du langage UML (Unified Modeling Language) en tant qu'élément intégré à ce processus. Ces outils et approches garantissent une gestion efficace du projet et facilitent la communication entre les différents intervenants.

3.2 Méthodologie de développement

Pour répondre aux besoins de notre utilisateur, il est nécessaire de suivre une approche méthodique en utilisant un modèle de gestion de projet par étapes. Il existe plusieurs modèles de ce type tels que le modèle en V, le modèle en Cascade, le modèle UP, le modèle Spirale, etc. Après analyse de la nature de notre projet, nous avons opté pour le processus unifié (UP) car il permet d'accélérer le développement grâce à des objectifs clairs et à court terme. Cette approche permettra une meilleure adaptation aux attentes de notre utilisateur.

3.3 Processus Unifié (Unified Process)

3.3.1 Définition

Le Processus Unifié (PU) est une suite continue d'opérations visant à fabriquer un travail ou un projet, en unifiant les différentes méthodes d'analyse et de conception orientées objet.

Avant l'introduction du Processus Unifié, ces méthodes étaient diverses, mais des experts renommés tels que Rumbaugh, Jacobson et Booch ont eu l'idée de les unifier. [14]

Le Processus Unifié est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise. [15]

Il offre une structure cohérente et un cadre de travail pour le développement logiciel, mettant l'accent sur l'itération, l'incrémentation, la modélisation visuelle et la gestion des risques [14].

3.3.2 Caractéristiques du Processus Unifié

Le Processus Unifié est une méthode de développement de logiciel caractérisée par :

- Un pilotage par les cas d'utilisation

L'objectif d'un système logiciel est de rendre service à ses utilisateurs. Pour réussir la mise au point d'un système, il est important par conséquent, de bien comprendre les désirs et les besoins de ses futurs utilisateurs. Un cas d'utilisation est une fonctionnalité du système produisant un résultat satisfaisant pour l'utilisateur. Les cas d'utilisation saisissent les besoins fonctionnels et leur ensemble forme le modèle des cas d'utilisation qui décrit les fonctionnalités complètes du système [16].

- démarche centrée sur l'architecture

Une architecture adaptée est la clé de voûte du succès d'un développement. Elle décrit des choix stratégiques qui déterminent en grande partie les qualités du logiciel (adaptabilité, performances, fiabilité...).

Ph. Kruchten propose différentes perspectives, indépendantes et complémentaires, qui permettent de définir un modèle d'architecture (publication IEEE, 1995). Cette vue (« 4+1 ») a fortement inspiré UML (voir Figure 3) [17].

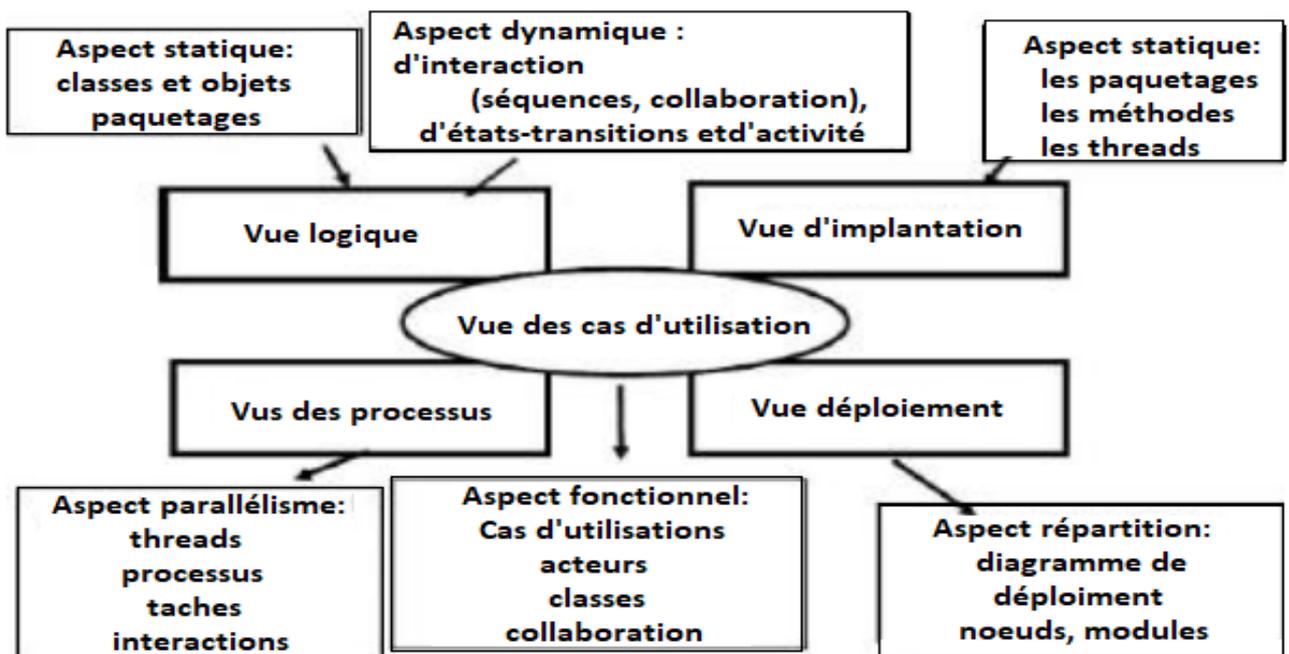


Figure 3 les 4+1 vues du processus unifié

- Une approche itérative et incrémentale

Le projet est découpé en itérations ou étapes de courte durée, ce qui permet de suivre de manière plus précise l'avancement global. À la fin de chaque itération, une partie exécutable du système final est produite de manière incrémentale, c'est-à-dire par ajout progressif. Cette approche itérative et incrémentale permet de livrer des fonctionnalités utilisables à chaque étape du développement, offrant ainsi des bénéfices immédiats et la possibilité d'ajustement et d'amélioration continue [18].

La figure suivante illustre le déroulement du processus unifié (voir Figure 4).

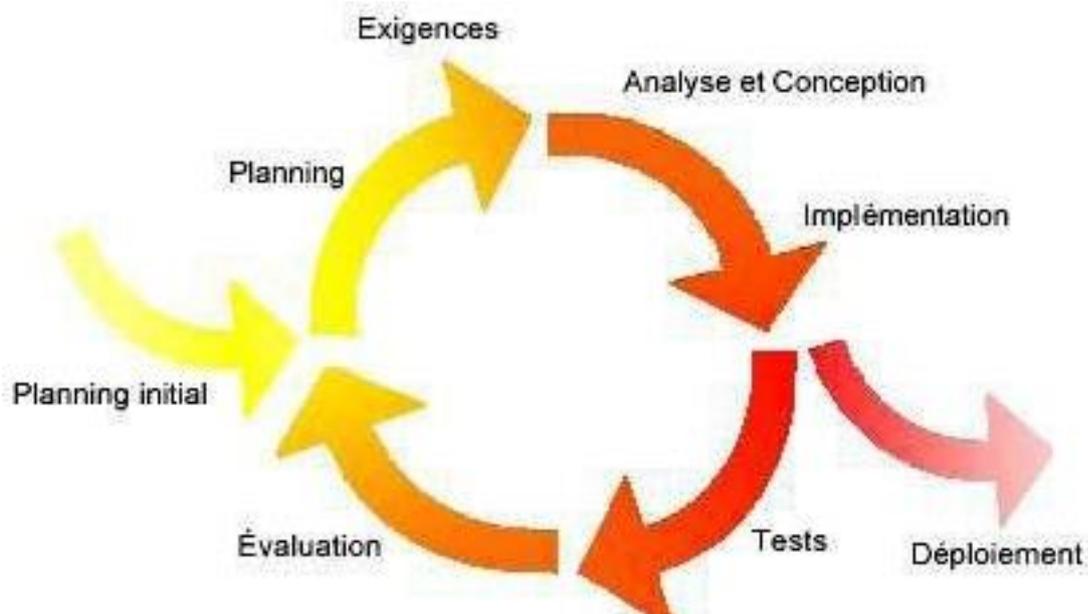


Figure 4 Déroulement du Processus UP

3.3.3 Cycle de vie du Processus unifié

UP gère le processus de développement par deux axes.

- L'axe vertical représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités d'enchaînements, d'artefacts et de travailleurs.
- L'axe horizontal représente le temps et montre le déroulement du cycle de vie du processus. cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'itérations et de jalons [15].

La figure suivante représente le cycle de vie du processus unifié (voir Figure 5).

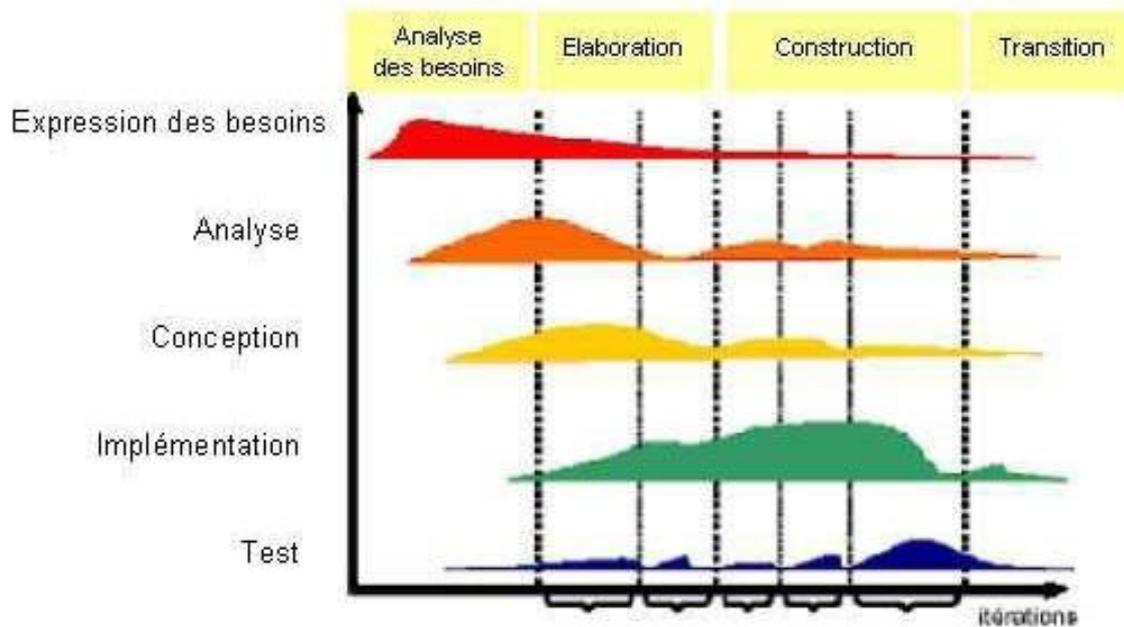


Figure 5 Cycle de vie du Processus Unifié

L'axe horizontal : Phases et itérations du processus

Phase	Description et enchaînement d'activités
Phase de creation	<p>Cette première étape traduit une idée en vision de produit fini et présente une étude de faisabilité et de rentabilité pour ce produit.</p> <ul style="list-style-type: none"> • Que va faire le système pour les utilisateurs ? • A quoi peut ressembler l'architecture d'un tel système ? • Quels sont l'organisation et les coûts du développement de ce produit ?
Phase elaboration	<p>Cette étape permet de préciser la plupart des cas d'utilisation et de concevoir l'architecture du système. Un planning est réalisé pour les phases suivantes du projet en indiquant le nombre d'itérations à réaliser pour les phases de construction. Par exemple : A l'issue de cette phase, le chef de projet doit être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet.</p>

Phase de construction	Durant cette étape, des sous-ensembles du produit final sont exécutables et stables. Cette phase correspond à la production d'une première version du produit. Elle est donc fortement centrée sur les activités de conception, d'implémentation et de test. En effet, les composants et les fonctionnalités non implémentés dans la phase précédente le sont ici. Au cours de cette phase, la gestion et le contrôle des ressources ainsi que l'optimisation des coûts représentent les activités essentielles pour aboutir à la réalisation du produit.
Phase de transition	Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts. Cette phase suppose des activités comme la fabrication, la formation des Langage de modélisation UML 36 utilisateurs clients, la mise en œuvre d'un service d'assistance et la correction des anomalies constatées.

Tableau 1 Phases et itérations du Processus Unifié

L'axe vertical : Activités du processus (activités de chaque cycle)

Activité	Description
Expression des besoins	<p>L'expression des besoins comme son nom l'indique, permet de définir les différents besoins :</p> <ul style="list-style-type: none"> • inventorer les besoins principaux et fournir une liste de leurs fonctions. • recenser les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation. • appréhender les besoins non fonctionnels (techniques) et livrer une liste des exigences. <p>Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.</p>

Analyse	L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution. Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système. Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.
Conception	<p>La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune. Elle constitue un point de départ à l'implémentation :</p> <ul style="list-style-type: none"> • Elle décompose le travail d'implémentation en sous-système • Elle crée une abstraction transparente de l'implémentation
Implémentation	L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutable et d'autres éléments du même type. Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.
Test	Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

Tableau 2 Activités du processus (activités de chaque cycle)

3.4 Langage UML

3.4.1 Définition

UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer les divers aspects d'un système d'information.

UML est en effet une norme maintenue par l'OMG (Object Management Group). Ce langage de modélisation graphique et textuel, issu du développement logiciel, peut également être appliqué à toute science fondée sur la description d'un système [19].

3.4.2 Avantages d'UML

Il permet ainsi :

- Un gain de précision.
- Un gage de stabilité.
- L'utilisation d'outils.
- Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel [20].

3.4.3 Inconvénients d'UML

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- L'intégration d'UML dans un processus déjà existant n'est pas triviale, et améliorer un processus est une tâche complexe et longue [20].

3.4.4 Les diagrammes UML

I. Définition

- Un diagramme UML fournit une représentation visuelle d'un aspect d'un système.
- Les diagrammes UML illustrent les aspects quantifiables d'un système qui peuvent être décrits visuellement, tels que les relations, le comportement, la structure ou la fonctionnalité [21].

II. Les Différents types de diagrammes

UML comporte plusieurs types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Il existe trois axes de modélisation : fonctionnel, statique, dynamique [21].

La figure suivante représente les axes de modélisation (voir Figure 6).

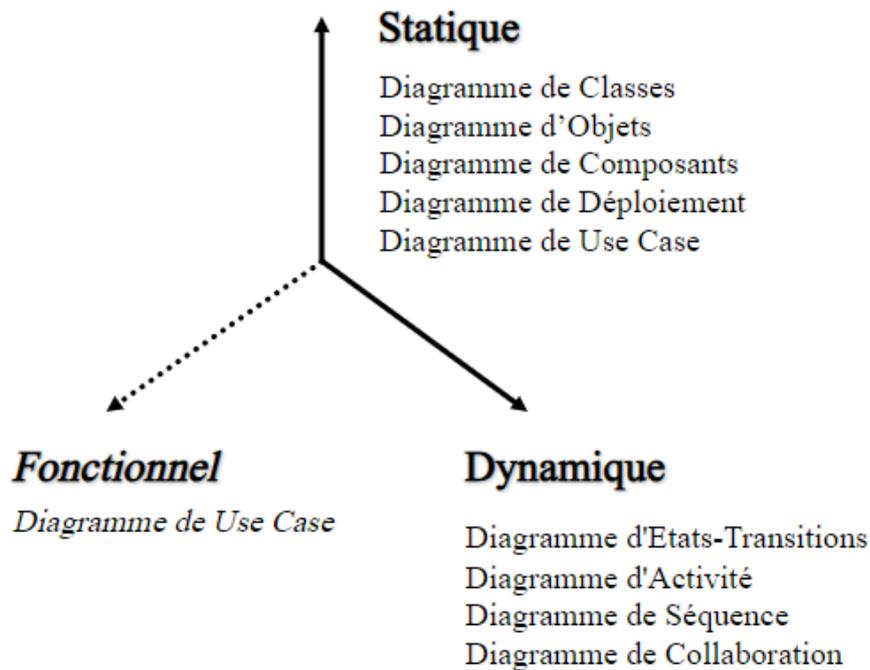


Figure 6 Axes de modélisation

III. Présentation de quelques Diagrammes essentiels

a. Le diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est destiné à représenter les besoins des utilisateurs par rapport au système.

Concepts de base	
Acteur	Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou

	autre système) qui interagit directement avec le système étudié.
Cas d'utilisation	Correspond à un certain nombre d'action que le système devra exécuter en réponse à un besoin d'un acteur
Interaction	Permet de décrire les échanges entre un acteur et un cas d'utilisation
Relation entre cas d'utilisation	
Relation « Inclusion »	Formalisé par le mot clé « include », une instance du cas d'utilisation source comprend également le comportement décrit par le cas d'utilisation destination
Relation « Extension »	Formalisée par le mot clé « extend », le cas d'utilisation source étend le comportement du cas d'utilisation destination.
Relation « Généralisation »	On dit que le cas d'utilisation A est une généralisation de B, si B est un cas particulier de A
Relation entre acteur	
Généralisation	Un acteur A est une généralisation d'un acteur B si l'acteur A peut-être substitué par l'acteur B. Dans ce cas, tous les cas d'utilisation accessibles à A le sont aussi à B, mais L'inverse n'est pas vrai

Tableau 3 Concept de base du diagramme de cas d'utilisation

b. Diagramme de classe :

Le diagramme de classes est le plus important des diagrammes UML, c'est le seul qui soit obligatoire lors de la modélisation objet d'un système. Un diagramme de classes fournit une vue globale d'un système. Il permet de modéliser les classes du système et leurs relations.

Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction [21].

Concepts de base	
Les classes	Une classe est la description d'un ensemble d'objet qui partage les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique. Une classe est symbolisée par un rectangle.
Attribut	Un attribut est une propriété nommée d'une classe qui décrit un ensemble de valeurs que les instances de cette propriété peuvent prendre. Une classe peut ne pas avoir, comme elle peut avoir un ou plusieurs attributs.
Opération	Une opération est une abstraction de ce que peut réaliser un objet et qui est réalisable par tous les objets de la classe. Une classe peut ne pas avoir comme elle peut avoir plusieurs opérations.
Relation entre les classes	
Association	représente une relation sémantique durable entre deux classes.
Agrégation	est un particulier d'association non symétrique exprimant une relation de contenance
Composition	est une agrégation plus forte.

Tableau 4 Concept de base du diagramme de classe

c. Diagramme de séquence

Les diagrammes de séquences permettent de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs :

- Les objets au cœur d'un système interagissent en s'échangeant des messages.
- Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine) [21].

Concepts de base	
Objet	est une manifestation concrète d'une abstraction à laquelle on peut appliquer un ensemble d'opérations et qui possède un état capable de mémoriser les effets de ces opérations. On représente un objet en soulignant son nom
Lien	est une liaison sémantique entre objets, en générale, il s'agit d'une instance d'une association. Chaque fois qu'une classe est reliée à une autre par une association, il peut y avoir un lien entre les instances des deux classes, et chaque fois qu'un lien existe entre deux objets, le premier objet peut envoyer un message au deuxième.
Message	est la spécification d'une communication entre objets, qui transporte des informations et qui s'affiche dans le but de déclencher une activité. La réception d'une instance de message peut être considérée comme une instance d'un événement.

Tableau 5 Concept de base du diagramme de séquence

3.5 Expression des besoins

L'analyse de ce projet a permis d'identifier différents besoins auxquelles notre application doit répondre, ces besoins sont classés en deux catégories distinctes :

➤ **Besoin fonctionnels** : se réfèrent aux fonctionnalités spécifiques et aux actions que le système doit pouvoir exécuter pour satisfaire les attentes des utilisateurs et peuvent être énumérés comme suit :

I. Gestion des produits :

- Ajouter, modifier et supprimer des produits.
- Rechercher et filtrer les produits par nom, catégorie, etc.
- Suivre les informations telles que la quantité en stock, le prix, etc.
- Impression de la liste des produits.

II. Gestion des commandes :

- Passer des commandes de produits.
 - Suivre l'état des commandes (en attente, en cours de traitement, etc.).
 - Traitement des commandes.
 - Impression des bons de commandes.
- III. Gestion des fournisseurs :
- Ajouter, modifier et supprimer des fournisseurs.
 - Suivre les informations sur les fournisseurs (nom, adresse, contact, etc.).
 - Rechercher et filtrer les fournisseurs par nom, entreprise, etc.
- IV. Gestion des bons d'entrée :
- Créer et enregistrer des bons d'entrée pour les nouveaux produits reçus.
 - Associer les bons d'entrée aux fournisseurs et aux factures correspondantes.
 - Suivre les détails des bons d'entrée tels que les dates, les quantités, les prix, etc.
 - Mettre à jour les quantités en stock en fonction des bons d'entrée.
 - Rechercher et filtrer les bons par numéro de bon, date, etc.
 - Impression des bons d'entrée.
- V. Gestion des bons de sortie :
- Générer des bons de sortie pour les produits sortants.
 - Associer les bons de sorties aux commandes correspondants.
 - Suivre les détails des bons de sortie tels que les dates, les quantités, etc.
 - Mettre à jour les quantités en stock en fonction des bons de sortie.
 - Rechercher et filtrer les bons par numéro de bon, date, etc.
 - Impression des bons de sortie.
- VI. Gestion des stocks :
- Suivre les niveaux de stock pour chaque produit.
 - Générer des rapports sur les mouvements de stock (fiche de stock de chaque produit).
- VII. Gestion des utilisateurs :
- Gérer des comptes utilisateur avec des rôles et des autorisations appropriés.
- VIII. Gestion des notifications et des traces :
- Recevoir des notifications en cas de niveaux de stock bas.
 - Recevoir des notifications sur les commandes (nouvelle commande, traitement de commande).
 - Assurer la traçabilité complète des activités et des événements survenus dans le système.

IX. Gestion des années de gestion de stock :

- Créer, modifier et supprimer des années de gestion de stock.
- Associer les opérations de stock (entrées, sorties, commandes, etc.) à des années de gestion spécifiques.

X. Statistique :

- Générer des graphiques pour représenter les données statistiques.

➤ **Besoins non fonctionnels** : décrivent les exigences et les contraintes qui ne sont pas liées aux fonctionnalités spécifiques du système, mais qui affectent plutôt sa qualité, sa performance, sa sécurité, sa convivialité et d'autres aspects, parmi ces besoins on peut citer :

- a. **Extensibilité** : Le système peut être facilement étendu pour ajouter de nouvelles fonctionnalités et s'adapter aux besoins futurs de l'université.
- b. **Sécurité** : Les informations stockées dans le système ne devront pas être accessibles à tout le monde.
- c. **Auditabilité** : Le système permet de suivre et d'enregistrer les actions effectuées par les utilisateurs, facilitant ainsi la traçabilité des opérations et l'identification des responsabilités.
- d. **Performances optimales** : Le système doit offrir un temps de réponse rapides, un chargement efficace des données, des fonctionnalités fluides et des opérations de traitement efficaces pour garantir une expérience utilisateur optimale.
- e. **Convivialité** : offrir une facilité d'utilisation, avec un enchaînement logique entre les pages et un texte clair, visible et lisible. permettant aux utilisateurs de naviguer facilement dans le système.

3.6 Acteurs du système

3.6.1 Identification des acteurs

Les acteurs impliqués dans le système de gestion des stocks sont les suivants :

- **Magasinier** : C'est l'utilisateur principal du système qui interagit avec celui-ci via une interface graphique. Son rôle essentiel est d'introduire toutes les informations relatives aux opérations effectuées dans le magasin. Cela inclut la gestion des produits, la gestion des bons d'entrée et la gestion des fournisseurs...etc.
- **Administrateur** : C'est lui qui contrôle toutes les opérations qui se passent dans l'application. Il se charge de la gestion des utilisateurs, la gestion des autorisations,...etc.
- **Chef de service** : C'est celui qui établit les bons de commandes internes spécifiques à son service.

- **Superviseur** : A pour rôle de consulter l'état et les mouvements des stocks. Il peut accéder aux informations sur les niveaux des stocks actuels, les entrées et sorties de produits et surveiller l'évolution des stocks dans l'ensemble du système.

3.6.2 Héritage des acteurs

La figure suivante illustre la hiérarchie des acteurs système de l'application, mettant en évidence les relations d'héritage entre eux (Voir Figure 7).

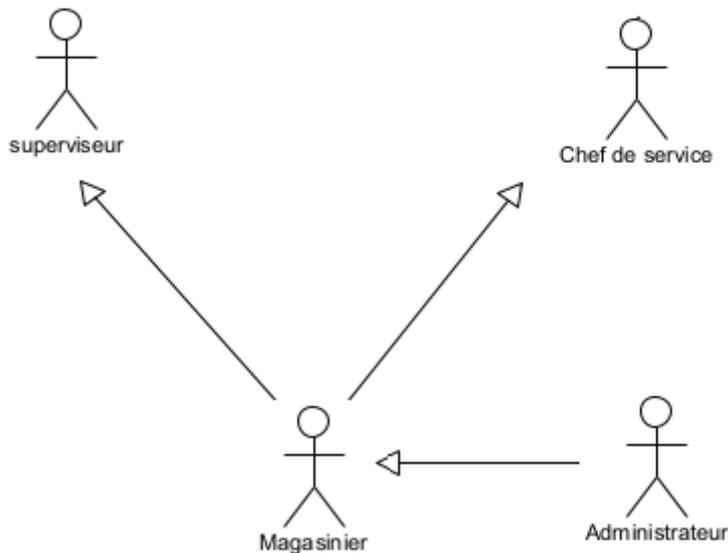


Figure 7 Héritage des acteurs

3.7. Cas d'utilisation du système

3.7.1 Identification des cas d'utilisation

Cas d'utilisation		Acteur
S'authentifier		Magasinier, chef service, administrateur, superviseur
Gestion des produits	Ajouter un produit	Administrateur Magasinier
	Modifier un produit	
	Supprimer un produit	
	Imprimer la liste	

Gestion des fournisseurs	Ajouter un fournisseur	Administrateur Magasinier
	Modifier un fournisseur	
	Supprimer un fournisseur	
Traitement des bons de commandes		Administrateur, Magasinier
Etablir un bon de commande		Administrateur, Magasinier, Chef de service
Impression d'un bon de commande		Administrateur, Magasinier, Chef de service
Gestion des bons d'entrées	Ajouter un bon	Administrateur Magasinier
	Modifier un bon	
	Consulter la liste	
	Imprimer un bon	
Gestion des factures	Ajouter une facture	Administrateur Magasinier
	Supprimer une facture	
	Modifier une facture	
Gestion des années de gestion des stocks	Ajouter une année	Administrateur Magasinier
	Modifier une année	
	Clôturer une année	
Gestion des catégories et sous catégories	Ajouter	Magasinier Administrateur
	Modifier	
	Supprimer	

Consulter les certificats d'affectations	Magasinier, Administrateur
Imprimer les certificats d'affectations	Magasinier, Administrateur
Consulter les bon de sorties	Magasinier, Administrateur
Imprimer les bons de sorties	Magasinier, Administrateur
Consulter les fiches de stock	Magasinier, superviseur, administrateur
Imprimer les fiches de stock	Magasinier, superviseur, administrateur
Consulter l'état du stock	Magasinier, superviseur, administrateur
Gestion des permissions	Administrateur
Consulter les traces	Administrateur

Tableau 6 Ensemble des cas d'utilisation

3.7.2 Descriptions textuelles des cas d'utilisation

La description textuelle d'un cas d'utilisation décrit de manière détaillée les étapes et les interactions entre les acteurs et le système dans le cadre d'un scénario spécifique. Ci-dessous quelques descriptions de cas d'utilisation clés de notre système :

a. Description du cas d'utilisation Ajouter un produit

➤ Identification :

- Nom du cas : Ajouter un produit.
- But : Ce cas d'utilisation permet au magasinier d'ajouter un produit. Dans le but de maintenir à jour la liste des produits dans le stock.
- Acteur principal : Magasinier.
- Acteur secondaire : Néant.

➤ Séquencement :

Ce cas d'utilisation commence lorsque l'administrateur décide d'ajouter un produit.

➤ Précondition :

- Le magasinier doit s'authentifier
- Le magasinier a la permission d'ajouter des produits

➤ Enchaînement nominale :

1. Appel du cas « authentification ».
2. Le magasinier clique sur le bouton d'ajout.
3. Le système affiche l'interface correspondante.
4. Le magasinier saisit les données.

5. Le magasinier choisit l'option « Valider ».
6. Le système enregistre les données et affiche un message de succès.
7. Le système enregistre la trace de l'opération.

➤ Enchaînement alternatif :

[A1] : Si un champ d'information obligatoire est incomplet

L'enchaînement démarre au point 5 de la séquence nominal :

6. Le système affiche un message d'erreur.

b. Description du cas d'utilisation Importer des produits

➤ Identification :

- Nom du cas : Importer des produits.
- But : Ce cas d'utilisation permet au magasinier d'importer une liste de produits à partir d'un fichier Excel dans le système.
- Acteur principal : Magasinier.
- Acteur secondaire : Néant.

➤ Séquencement :

Ce cas d'utilisation commence lorsque le magasinier décide d'importer une liste de produits.

➤ Précondition :

- Le magasinier doit s'authentifier
- Le magasinier a la permission d'importer des produits

➤ Enchaînement nominale :

1. Appel du cas « authentification »
2. Le magasinier clique sur le bouton d'importation.
3. Le système affiche l'interface correspondante.
4. Le magasinier choisit le fichier Excel à importer.
5. Le magasinier confirme l'importation des produits.
6. Le système vérifie le format et la validité du fichier Excel.
7. Le système traite les données du fichier Excel et vérifie chaque produit.
8. Le système enregistre les produits importés dans le stock.
9. Le système enregistre la trace de l'opération d'importation des produits.
10. Le système affiche un message de succès.

➤ Enchaînement alternatif :

[A1] : Si le format du fichier Excel n'est pas valide ou ne correspond pas à la structure attendue.

L'enchaînement démarre au point 7 de la séquence nominal :

8. Le système affiche un message d'erreur.

c. Description du cas d'utilisation Ajouter un bon d'entrée

➤ Identification :

- Nom du cas : Ajouter un bon d'entrée.
- But : Ce cas d'utilisation permet au magasinier d'ajouter un nouveau bon d'entrée.
- Acteur principal : Magasinier.
- Acteur secondaire : Néant.

➤ Séquencement :

- Ce cas d'utilisation commence lorsque le magasinier décide d'ajouter un bon d'entrée.

➤ Précondition :

- Le magasinier doit s'authentifier.
- Le magasinier a la permission d'ajouter un bon d'entrée.

➤ Enchaînement nominale :

1. Appel du cas « authentification ».
2. Le magasinier clique sur le bouton d'ajout.
3. Le système affiche l'interface correspondante.
4. Le magasinier saisit les données sur le bon ainsi sur la facture.
5. Le magasinier choisit l'option « Valider ».
6. Le système enregistre les données et affiche un message de succès.
7. Le système enregistre la trace de l'opération

➤ Enchaînement alternatif :

[A1] : Si un champ d'information obligatoire est incomplet

L'enchaînement démarre au point 5 de la séquence nominal :

5. Le système affiche un message d'erreur.

d. Description du cas d'utilisation établir un bon de commande

➤ Identification :

- Nom du cas : Etablir un bon de commande.
- But : Ce cas d'utilisation permet d'établir un bon de commande.
- Acteur principal : Magasinier, chef service.
- Acteur secondaire : Néant.

➤ Séquencement :

- Ce cas d'utilisation commence lorsque l'utilisateur décide d'établir un bon de commande.
- Précondition :
 - L'utilisateur doit s'authentifier
 - L'utilisateur a la permission d'effectuer une commande
- Enchaînement nominale :
 1. Appel du cas « authentification ».
 2. L'utilisateur clique sur le bouton d'ajout.
 3. Le système affiche l'interface correspondante.
 4. L'utilisateur saisit les données sur le bon de commande.
 5. L'utilisateur choisit l'option « Valider ».
 6. Le système met à jour les données et affiche un message de succès.
 7. Le système envoie une notification au magasinier si la commande est faite par un chef de service.
- Enchaînement alternatif :

[A1] : Si un champ d'information obligatoire est incomplet

L'enchaînement démarre au point 5 de la séquence nominal :

 6. Le système affiche un message d'erreur.

e. Description du cas d'utilisation traitement d'un bon de commande

- Identification :
 - Nom du cas : Traitement d'un bon de commande.
 - But : Ce cas d'utilisation permet de traiter un bon de commande.
 - Acteur principal : Magasinier.
 - Acteur secondaire : Néant.
- Séquencement :
 - Ce cas d'utilisation commence lorsque le magasinier décide de traiter un bon de commande.
- Précondition :
 - Le magasinier doit s'authentifier.
 - Le magasinier a la permission de traiter une commande.
- Enchaînement nominale :
 1. Appel du cas « authentification ».
 2. Le magasinier clique sur le bouton approprié.
 3. Le système affiche l'interface correspondante.

4. Le magasinier saisit les données sur le bon.
 5. Le magasinier choisit l'option « Valider » dans le cas où il a terminé le traitement sinon « enregistrer ».
 6. Le système met à jour les données et affiche un message de succès.
 7. Le système envoie une notification au chef service concerné par la commande.
- Enchaînement alternatif :
- [A1] : Si un champ d'information obligatoire est incomplet ou les quantités des produit ne sont pas disponible
- L'enchaînement démarre au point 5 de la séquence nominal :
6. Le système affiche un message d'erreur.

f. Description du cas d'utilisation gestion des fournisseurs

- Identification :
- Nom du cas : Gérer les fournisseurs.
 - But : Ce cas d'utilisation permet de mettre à jour la liste des fournisseurs.
 - Acteur principal : Magasinier.
 - Acteur secondaire : Néant.
- Séquencement :
- Ce cas d'utilisation commence lorsque le magasinier décide d'ajouter, modifier ou supprimer.
- Précondition :
- Le magasinier doit s'authentifier
 - Le magasinier a la permission d'ajouter, modifier ou supprimer un fournisseur.
- Enchaînement nominale :
1. Appel du cas « authentification ».
 2. Si le magasinier veut ajouter un fournisseur, il clique sur le bouton d'ajout, en revanche, s'il veut supprimer il clique sur le bouton de suppression ou encore en cas de modification il clique sur le bouton éditer.
 3. Le système affiche l'interface correspondante.
 4. Le magasinier saisit les données en cas de modification ou d'ajout.
 5. Le magasinier choisit l'option « Valider ».
 6. Le système met à jour les données et affiche un message de succès.
 7. Le système enregistre la trace de l'opération.
- Enchaînement alternatif :

[A1] : Si un champ d'information obligatoire est incomplet

L'enchaînement démarre au point 5 de la séquence nominal :

5. Le système affiche un message d'erreur

3.7.3 Diagrammes des cas d'utilisations

Les figures suivantes illustrent les diagrammes des cas d'utilisation réalisés :

- **Diagrammes de cas d'utilisation associé au chef de service**

Le diagramme suivant représente les différentes tâches réalisées par le chef service (Voir Figure 8).

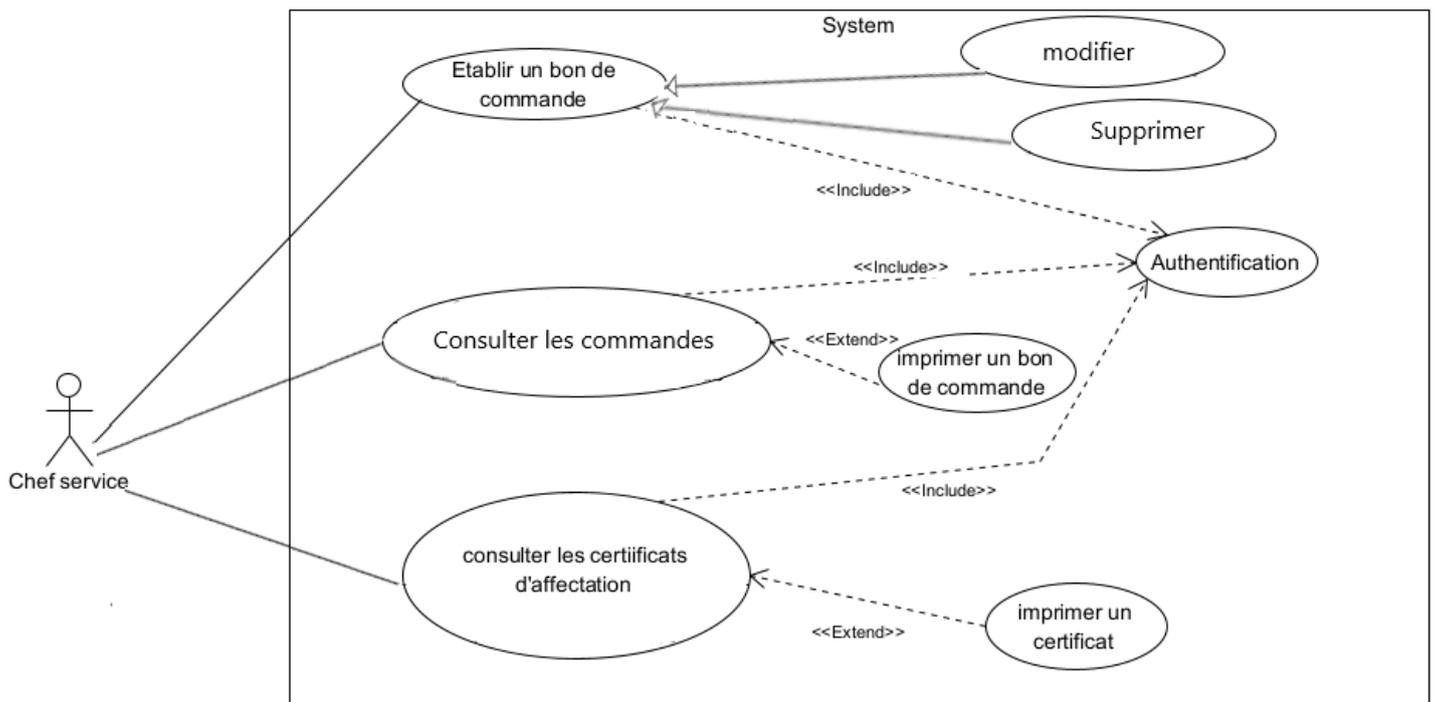


Figure 8 Diagramme de cas d'utilisation associé au chef service

- **Diagrammes des cas d'utilisation associé à l'administrateur**

Le diagramme suivant représente les différentes tâches réalisées par l'administrateur (Voir Figure 9).

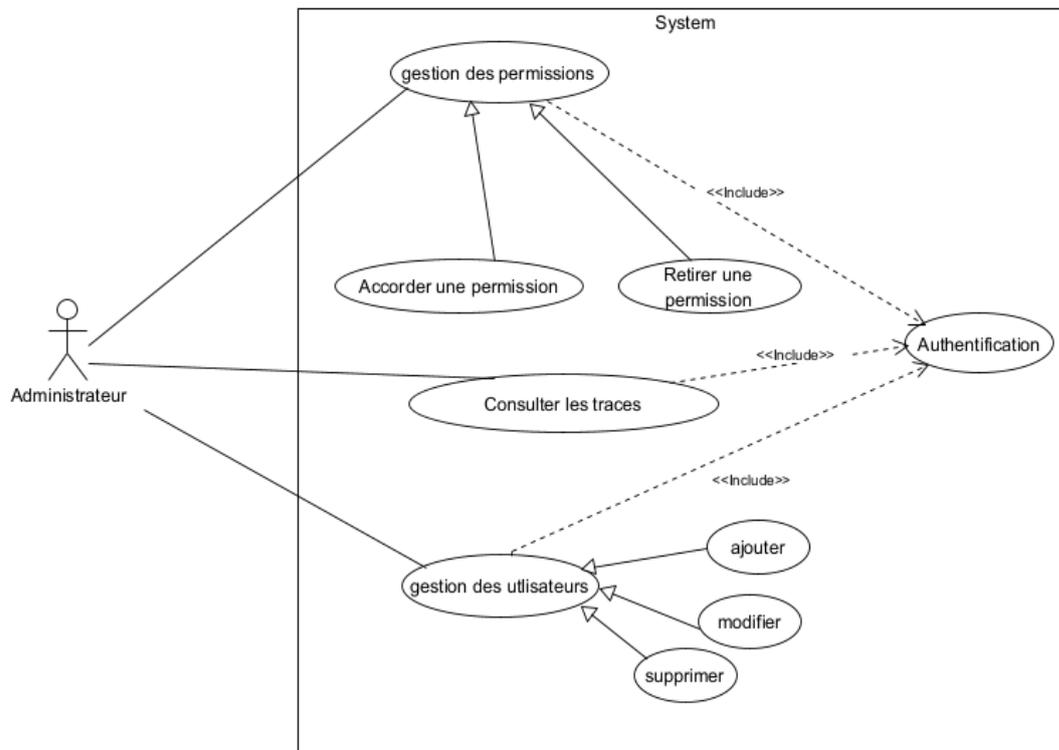


Figure 9 Diagramme des cas d'utilisation associé à l'administrateur

- **Diagrammes des cas d'utilisation associé au superviseur**

Le diagramme suivant représente les différentes tâches réalisées par le superviseur (Voir Figure 10).

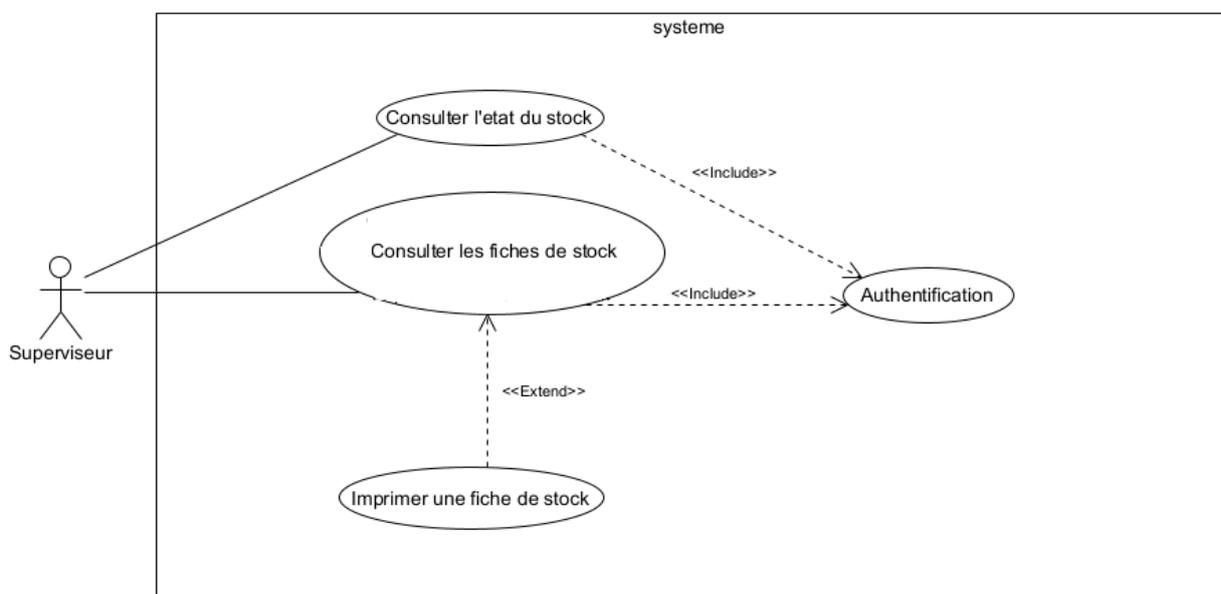


Figure 10 Diagramme des cas d'utilisation associé au superviseur

• Diagramme des cas d'utilisation associé au magasinier

Le diagramme suivant représente les différentes tâches réalisées par le magasinier (Voir Figure 11).



Figure 11 Diagramme des cas d'utilisation associé au magasinier

3.8 Diagrammes de séquence

Ci-dessous quelques diagrammes de séquence de cas d'utilisation clés de notre système :

- **Diagramme de séquence du cas d'utilisation ajouter un produit**

Pour l'ajout d'un produit dans le système, le magasinier remplit un formulaire et valide l'opération. Le système effectue une vérification de la saisie, si tous les champs sont corrects, l'ajout du produit est effectué. Cependant, si des erreurs sont détectées, un message d'erreur est affiché (Voir Figure 12).

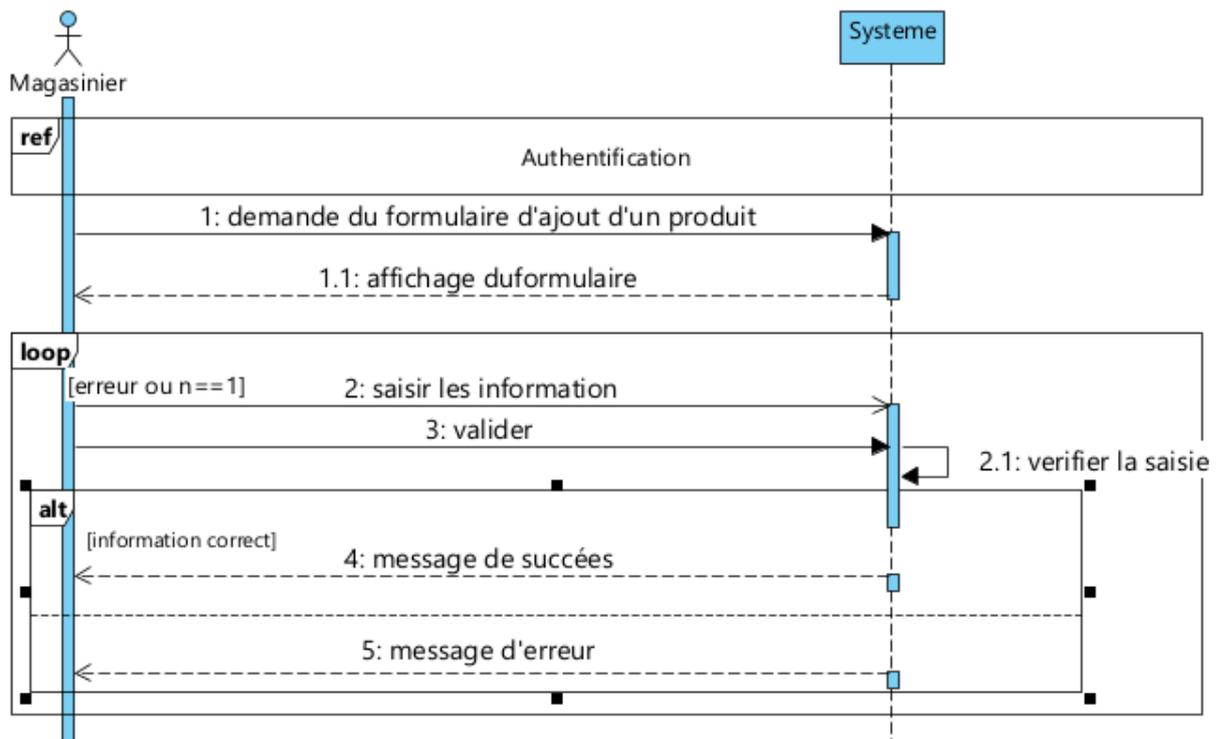


Figure 12 Diagramme de séquence ajouter un produit

- **Diagramme de séquence du cas d'utilisation importer des produits**

Pour l'importation de produits, le magasinier sélectionne un fichier à importer et valide l'opération. Le système procède à une vérification du contenu du fichier, si toutes les données sont correctes, les produits sont importés avec succès. Cependant, si des erreurs sont détectées dans le fichier, un message d'erreur est affiché (Voir Figure 13).

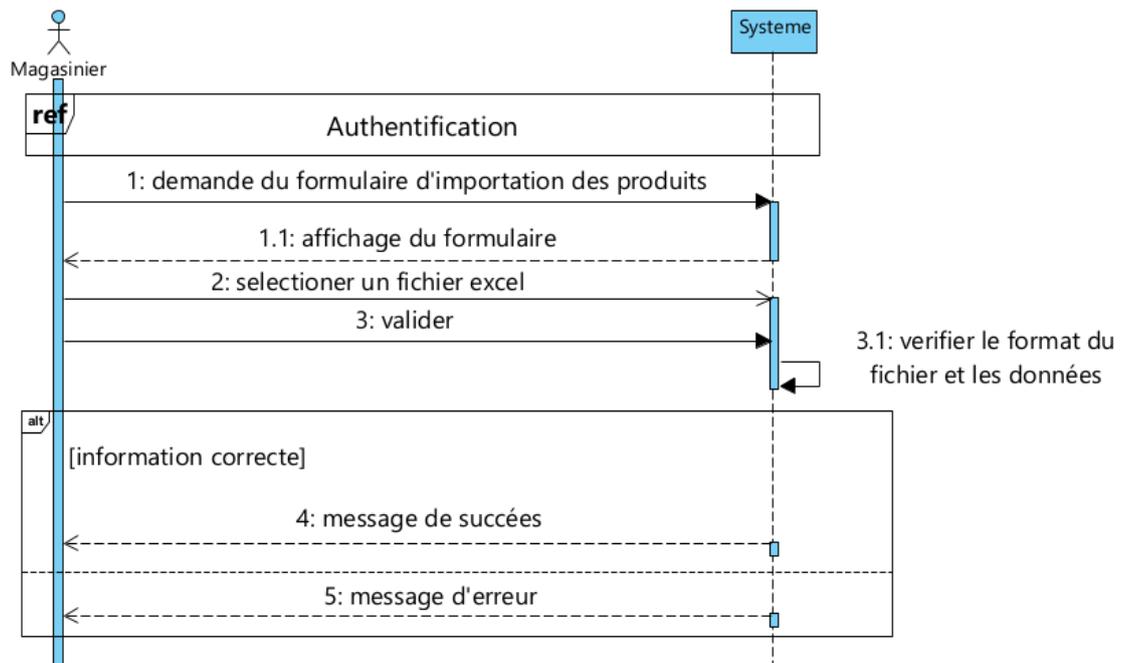


Figure 13 Diagramme de séquence importer des produits

- **Diagramme de séquence du cas d'utilisation modifier un produit**

Pour la modification d'un produit dans le système, le magasinier consulte la liste des produits et sélectionne le produit à modifier et accède au formulaire de modification. Le magasinier effectue les modifications nécessaires dans les champs appropriés et valide l'opération. Le système vérifie la saisie et effectue la mise à jour des informations du produit. En cas de champs incorrects ou manquants, un message d'erreur est affiché pour indiquer les problèmes rencontrés lors de la modification du produit (Voir Figure14).

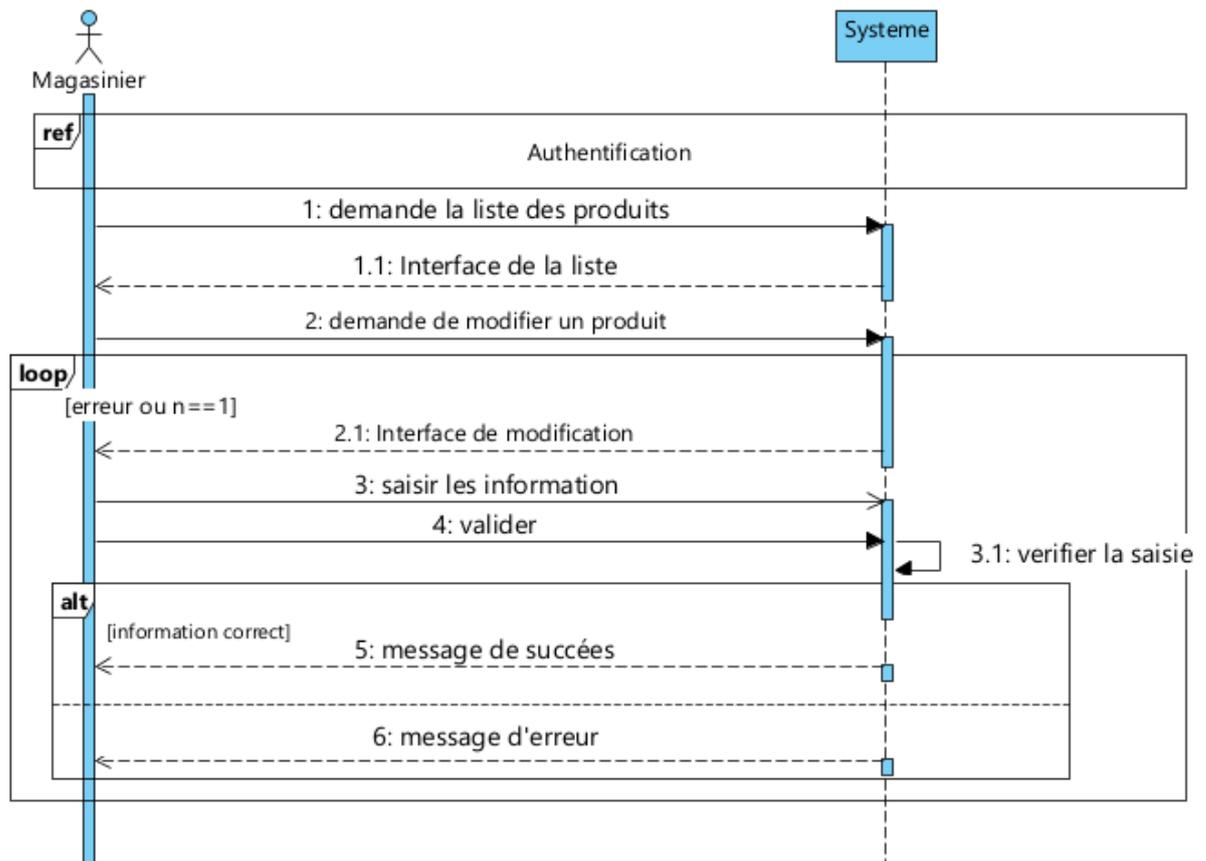


Figure 14 Diagramme de séquence modifier un produit

• Diagramme de séquence du cas d'utilisation supprimer un produit

Pour la suppression d'un produit dans le système, le magasinier accède à la liste des produits et cherche le produit à supprimer, si le produit est déjà utilisé le système n'affichera pas le bouton de suppression dans le cas contraire le magasinier clique sur le bouton de suppression. Le système affiche une confirmation de suppression pour confirmer l'action. Si le magasinier confirme la suppression, le produit est retiré de la liste des produits disponibles et un message de suppression réussie sera affiché. En cas d'annulation, aucune action n'est entreprise et le produit reste inchangé (Voir Figure15).

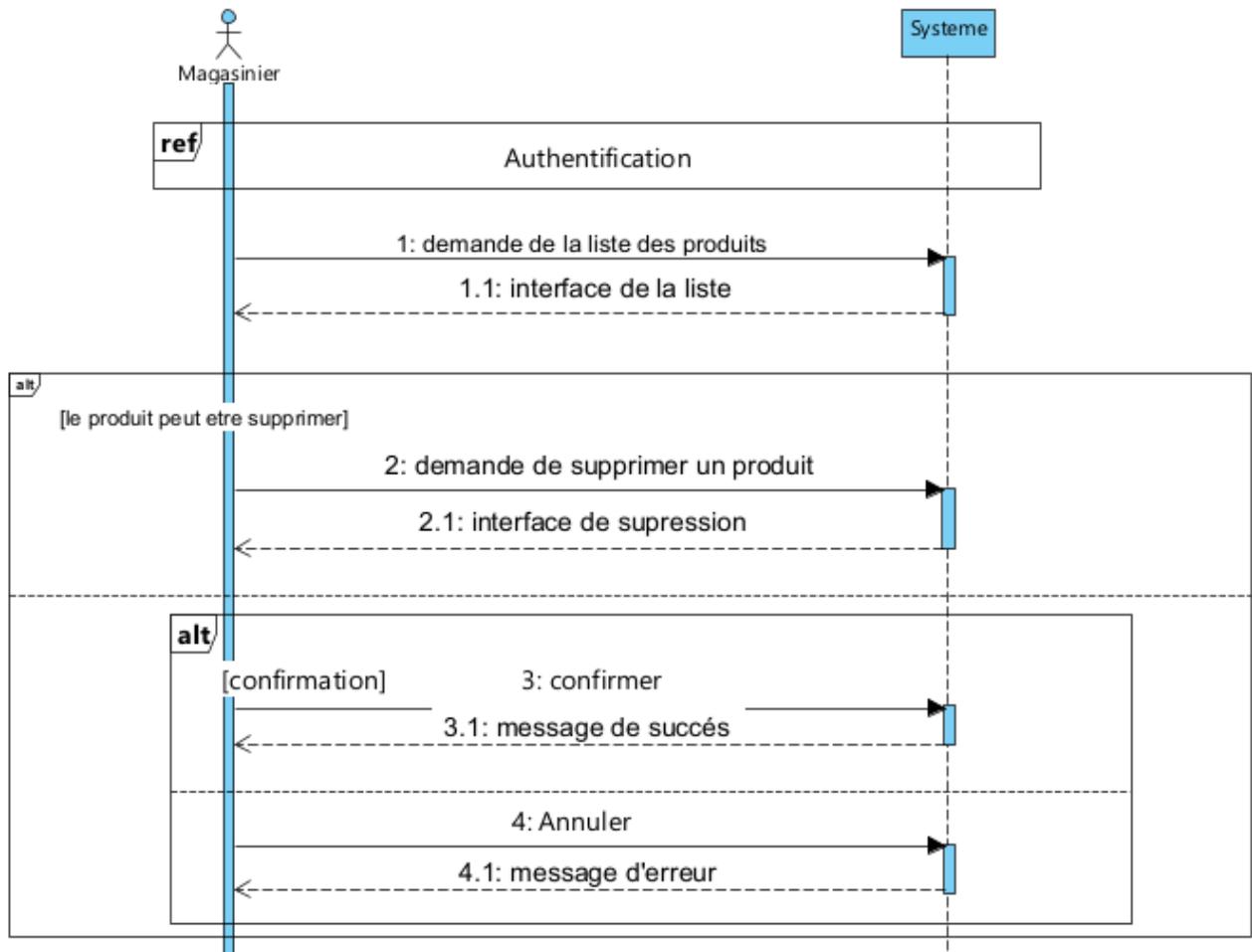


Figure 15 Diagramme de séquence supprimer un produit

- Diagramme de séquence du cas d'utilisation gestion des produits

Ce diagramme englobe toutes les opérations concernant le produit (Voir Figure16).

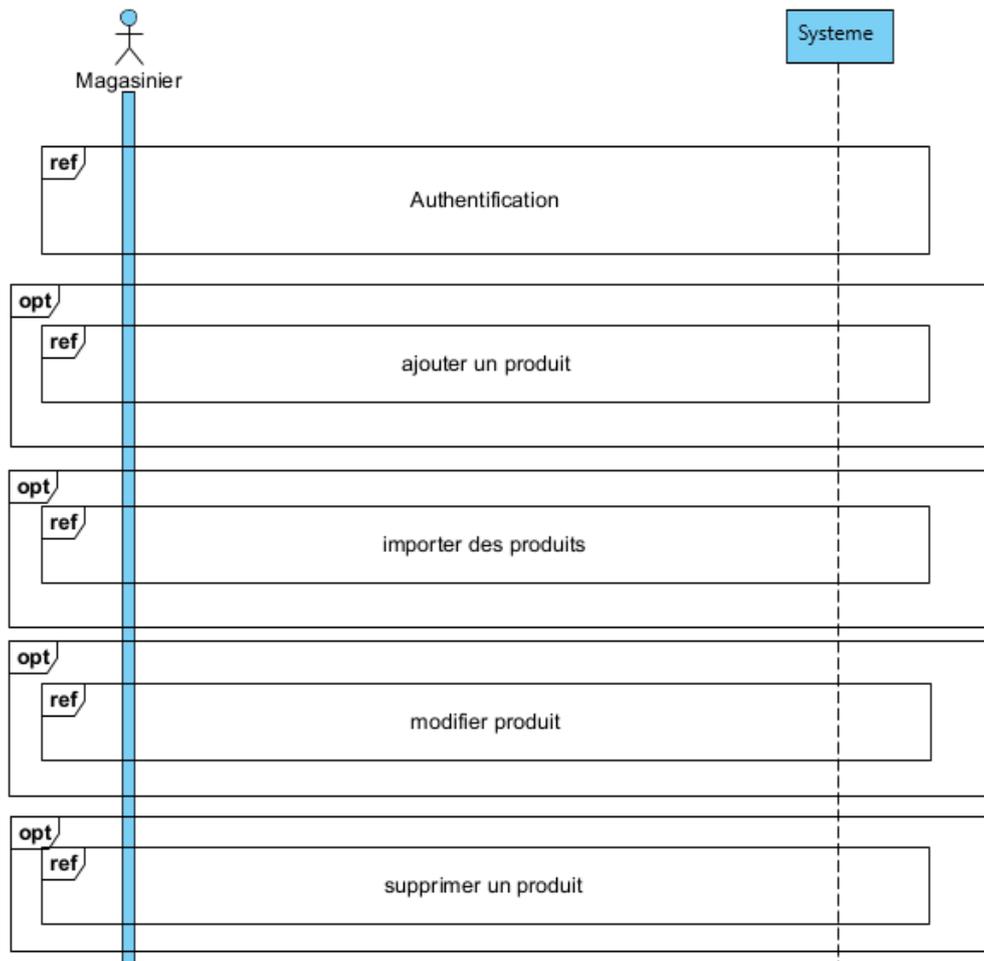


Figure 16 Diagramme de séquence gestion des produits

- **Diagramme de séquence du cas d'utilisation ajout d'un bon d'entrée**

Pour l'ajout d'un bon d'entrée dans le système, le magasinier remplit un formulaire et valide l'opération. Le système effectue une vérification de la saisie, et si tous les champs sont corrects, l'ajout du bon est effectué. Cependant, si des erreurs sont détectées, un message d'erreur est affiché (Voir Figure 16).

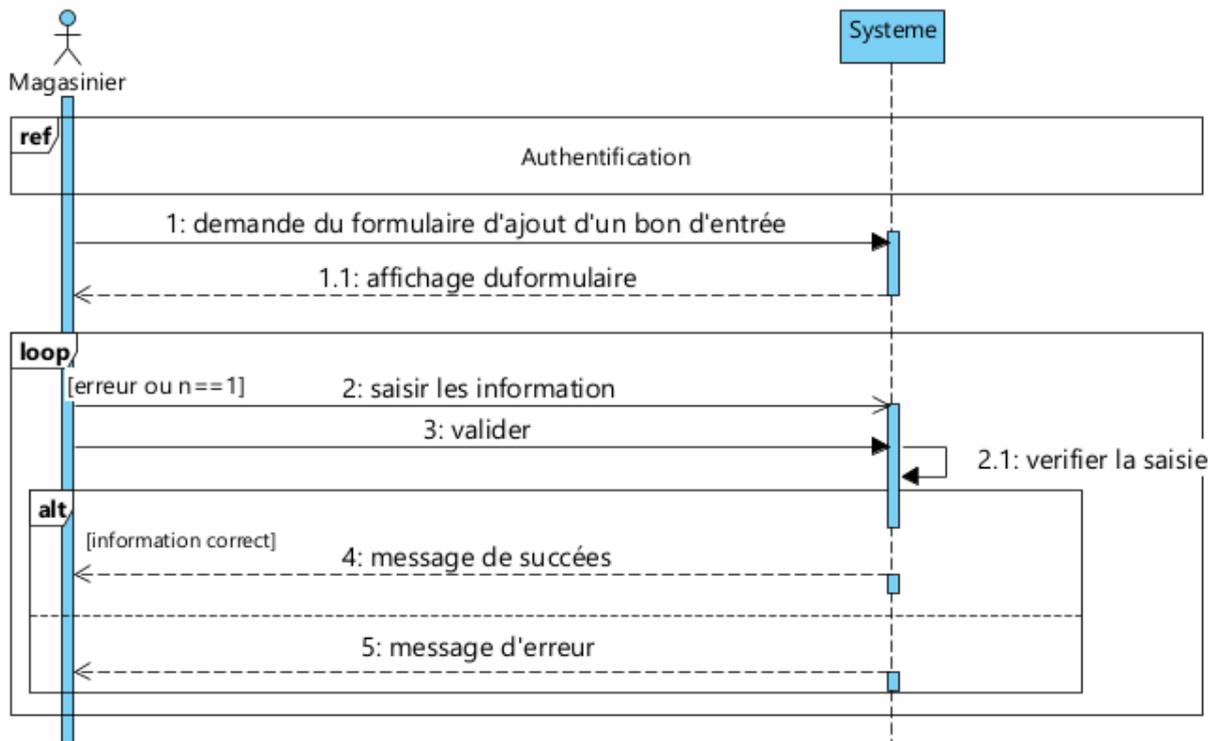


Figure 17 Diagramme de séquence ajout d'un bon d'entrée

- **Diagramme de séquence du cas d'utilisation établir un bon de commande**

Pour établir un bon de commande dans le système, l'utilisateur doit remplir un formulaire et valider l'opération. Le processus diffère selon le rôle de l'utilisateur. Si l'utilisateur est un chef de service, il doit sélectionner le magasin concerné dans le formulaire. En revanche, si l'utilisateur est un magasinier, il peut remplir le formulaire sans avoir à choisir un magasin spécifique. Le système effectue ensuite une vérification des données saisies et affiche un message d'erreur en cas de champs incorrects ou manquants. L'utilisateur est ainsi guidé pour corriger les erreurs et finaliser l'établissement du bon de commande (Voir Figure 18).

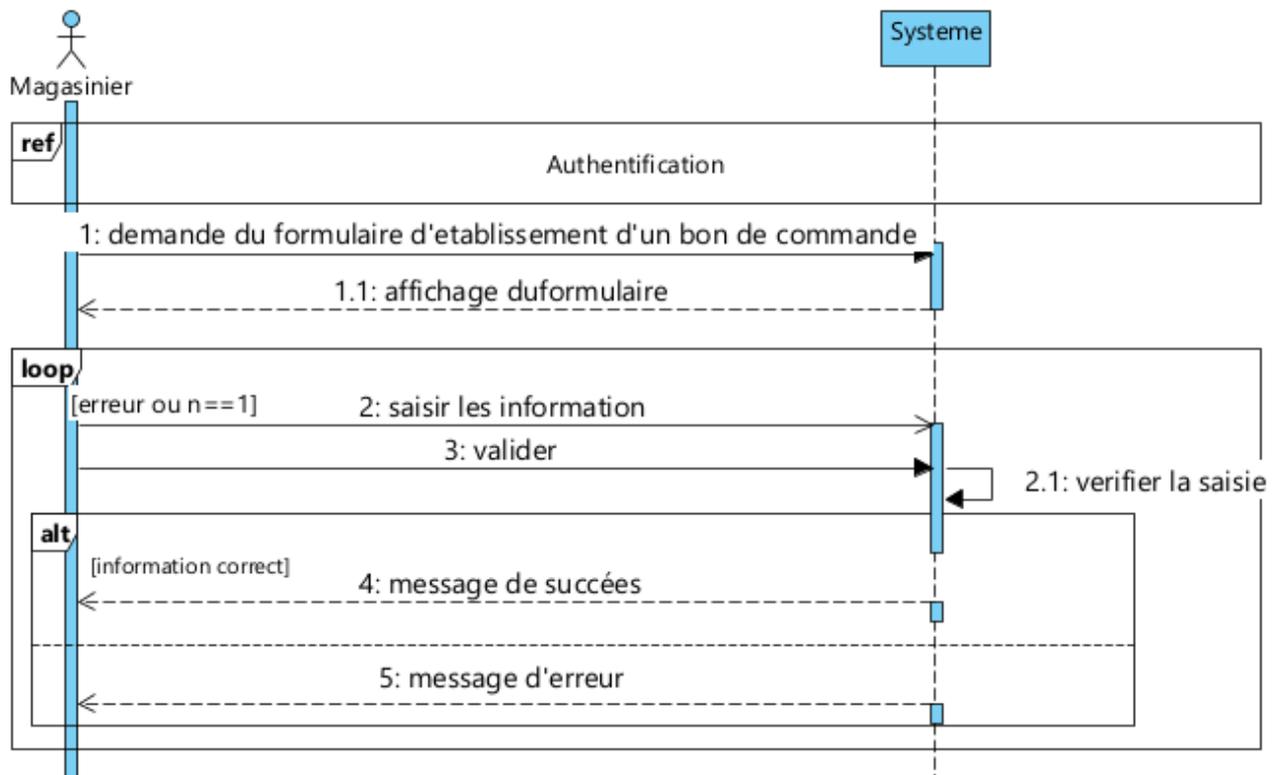


Figure 18 Diagramme de séquence établir un bon de commande

- **Diagramme de séquence du cas d'utilisation traitement d'un bon de commande**

Le traitement d'un bon de commande dans le système commence par la sélection d'un bon de commande existant par le magasinier. Les détails du bon de commande, tels que les produits commandés et les quantités spécifiées, sont affichés. À ce stade, le magasinier a la possibilité de refuser le bon ou de le traiter en attribuant les quantités à servir pour chaque produit sélectionné. Une fois le traitement effectué, le magasinier peut soit valider le traitement, ce qui indique que le traitement du bon est terminé, soit enregistrer le traitement et revenir ultérieurement pour poursuivre le processus.

Le système procède ensuite à une vérification des données saisies, en s'assurant que les quantités sont correctes et que les produits sont disponibles en stock. Si des erreurs sont détectées, un message d'erreur est affiché pour signaler les problèmes spécifiques à l'utilisateur. Si toutes les vérifications sont réussies, le système enregistre les modifications apportées au bon de commande et affiche un message de succès. De plus, une notification est envoyée au bénéficiaire du bon, et un bon de sortie est automatiquement généré (Voir Figure 19).

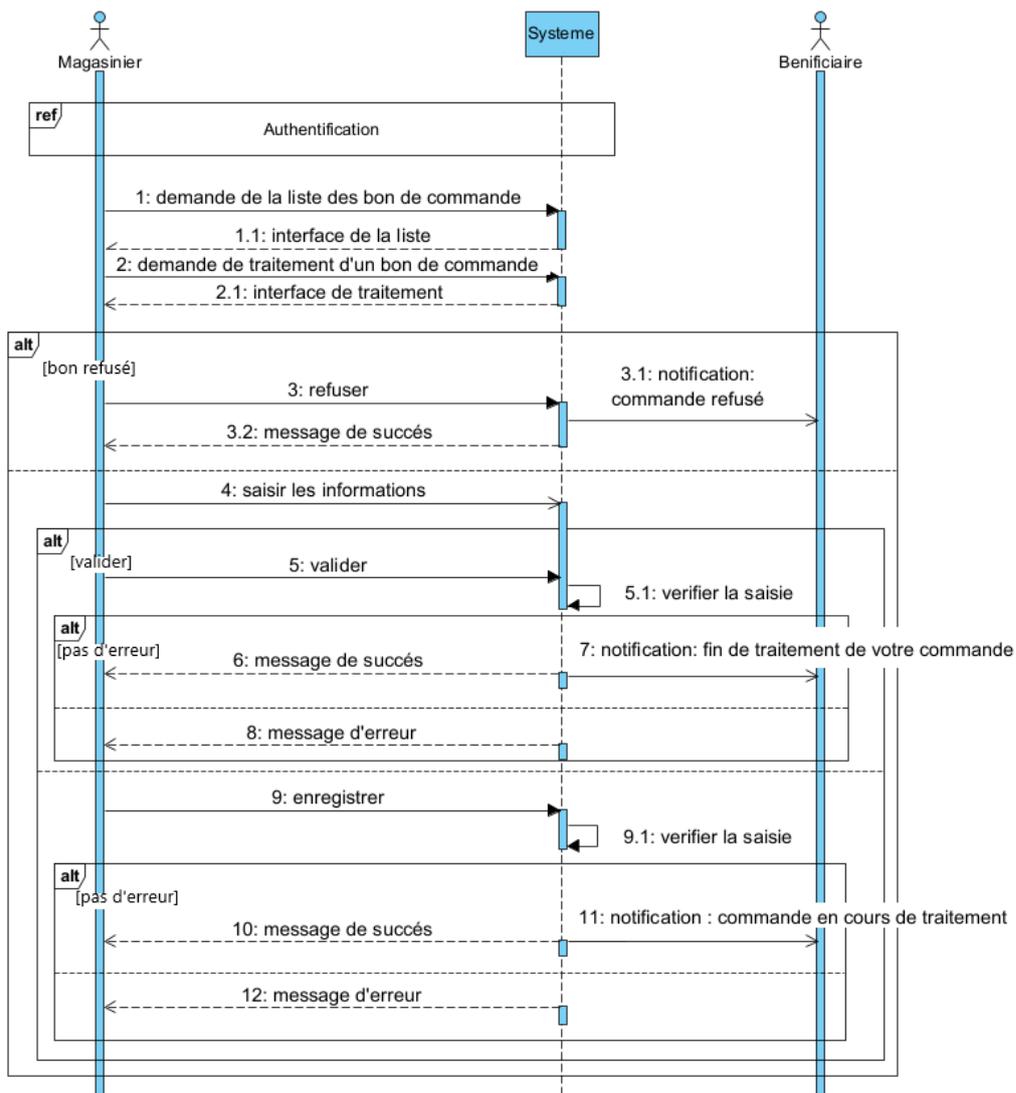


Figure 19 Diagramme de séquence traitement d'un bon de commande

- **Diagramme de séquence du cas d'utilisation imprimer une fiche de stock**

Pour l'impression d'une fiche de stock dans le système, l'utilisateur accède tout d'abord à la liste des fiches de stock disponibles. À partir de cette liste, l'utilisateur sélectionne une fiche spécifique pour laquelle il souhaite obtenir une version imprimée. En accédant à la page de détail de cette fiche, l'utilisateur repère le bouton d'impression dédié. En cliquant sur ce bouton, le système génère automatiquement un fichier PDF (Portable Document Format) contenant toutes les informations et les données de la fiche de stock (Voir Figure 20).

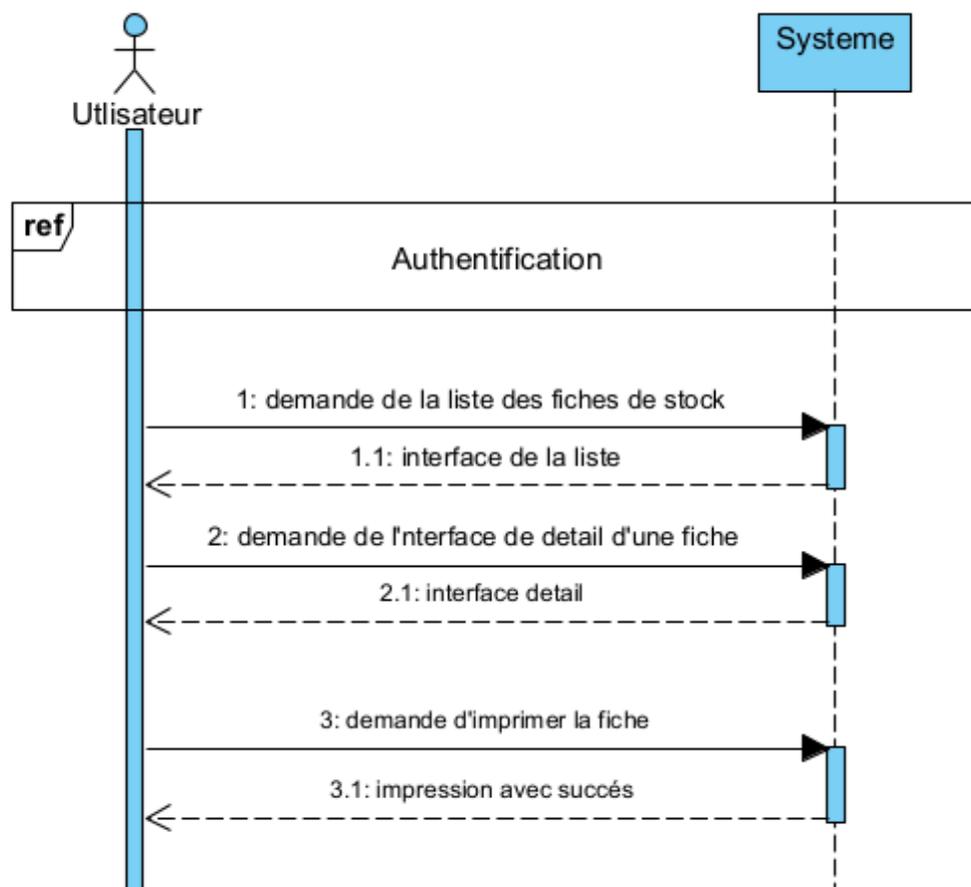


Figure 20 Diagramme de séquence imprimer une fiche de stock

3.9

Diagrammes de séquence Détaillés

Dans le diagramme de séquence détaillé, nous adoptons une approche où le système n'est plus considéré comme une boîte noire, mais plutôt comme un ensemble d'objets interagissant entre eux. Dans ce diagramme, nous utilisons trois types de classes pour représenter ces objets :

- ✓ Classes d'interface (Boundary) : Ces classes facilitent l'interaction entre l'application et les utilisateurs. Chaque cas d'utilisation est associé à au moins une classe d'interface.
- ✓ Classes de contrôle (Control) : Les classes de contrôle exécutent les traitements spécifiques à l'application, en utilisant les données fournies par les classes d'interface et en interagissant avec les classes entités.
- ✓ Classes entités (Entity) : Les classes entités encapsulent les données et le comportement associés à ces objets. Elles sont souvent persistantes, ce qui signifie qu'elles conservent leurs informations même après l'exécution d'un cas d'utilisation spécifique.

• Diagramme de séquence détaillé du cas ajouter un produit

Le figure suivante illustre le diagramme de séquence détaillé du cas ajouter un produit (Voir Figure 21).

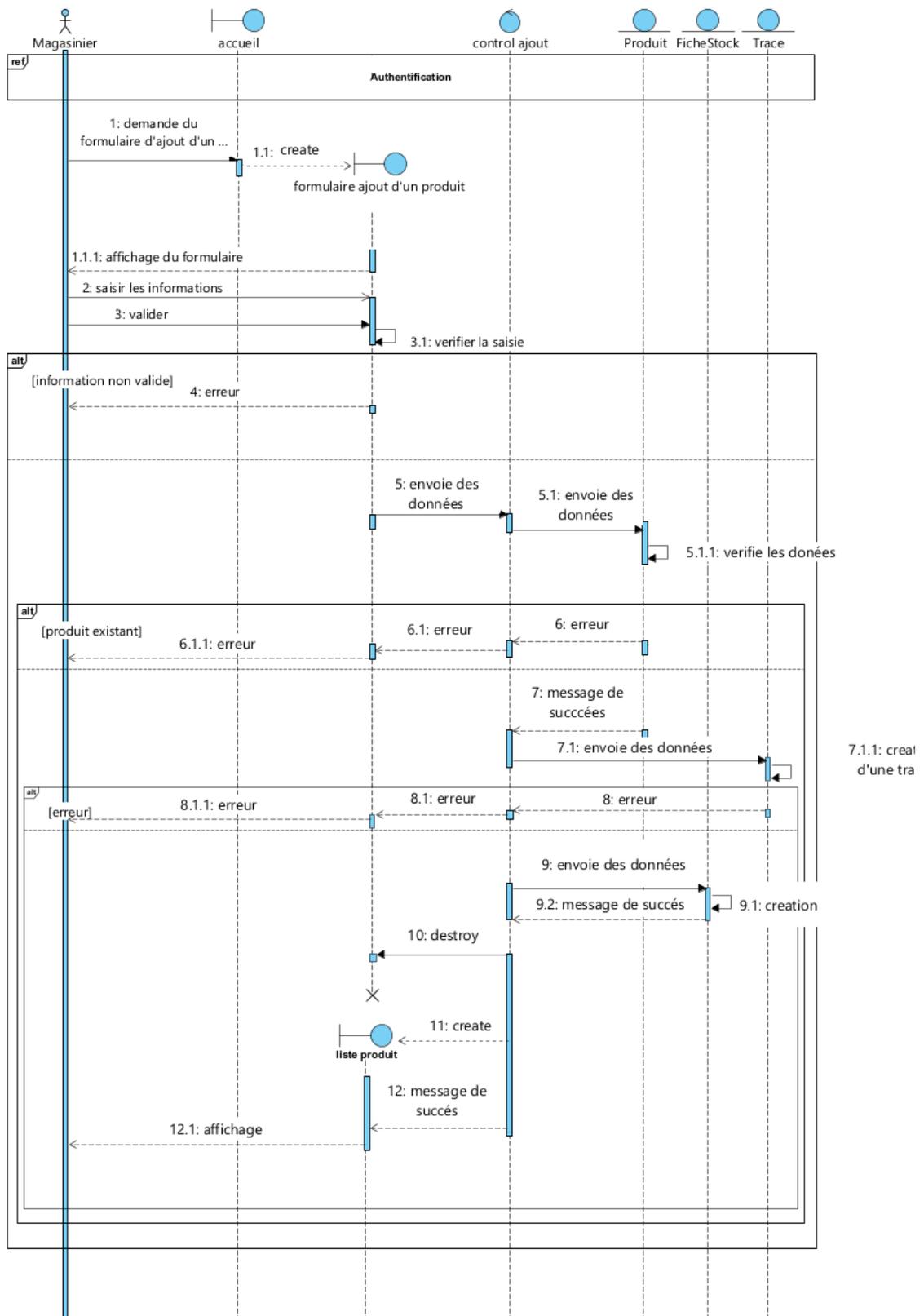


Figure 21 Diagramme de séquence détaillé du cas ajouter un produit

• Diagramme de séquence détaillé du cas ajouter un bon d'entrée

Le figure suivante illustre le diagramme de séquence détaillé du cas ajouter un bon d'entrée (Voir Figure 22).

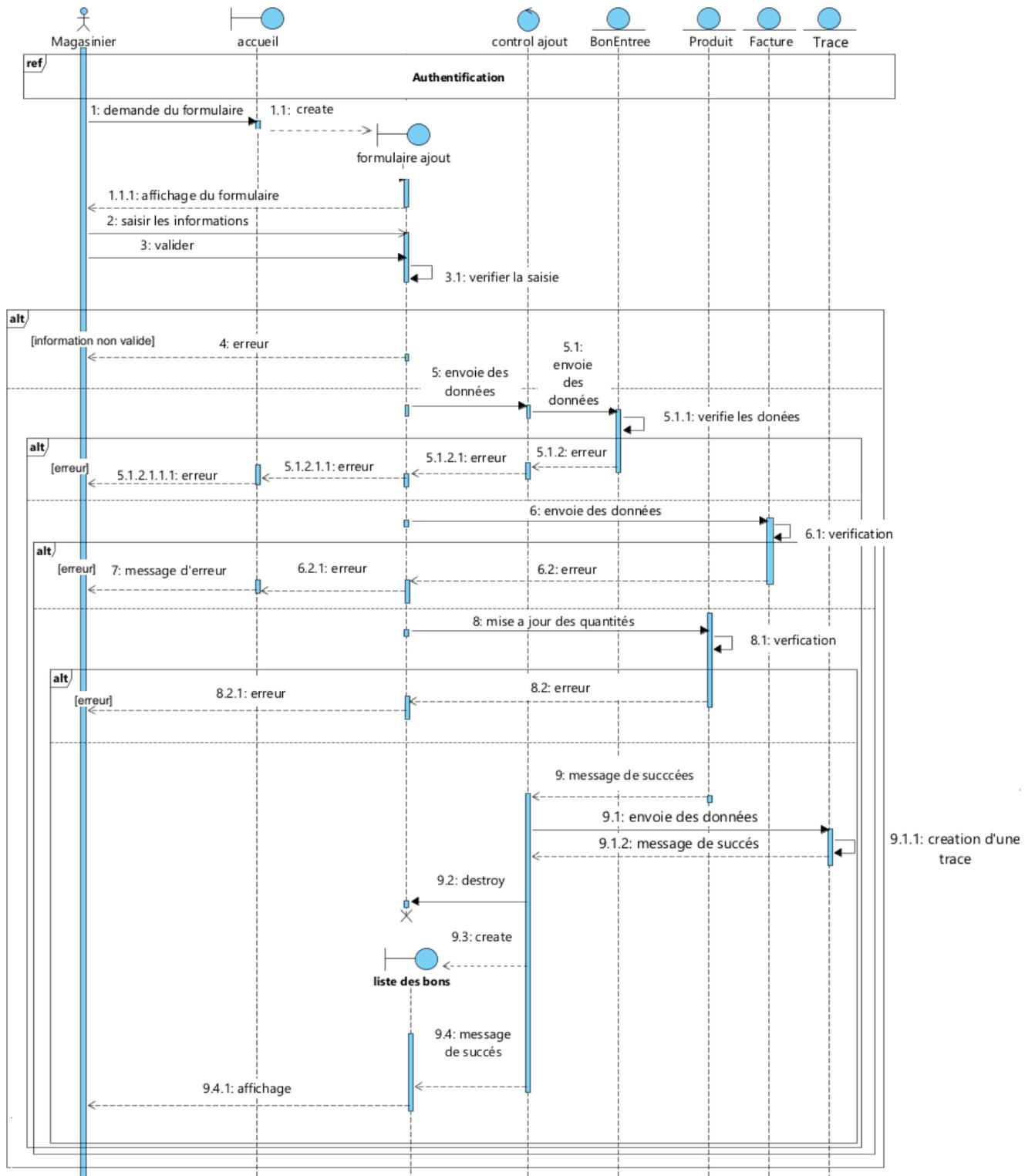


Figure 22 Diagramme de séquence détaillé du cas ajouter un bon d'entrée

• Diagramme de séquence détaillé du cas établir un bon de commande

Le figure suivante illustre le diagramme de séquence détaillé du cas ajouter un bon de commande (Voir Figure 23).

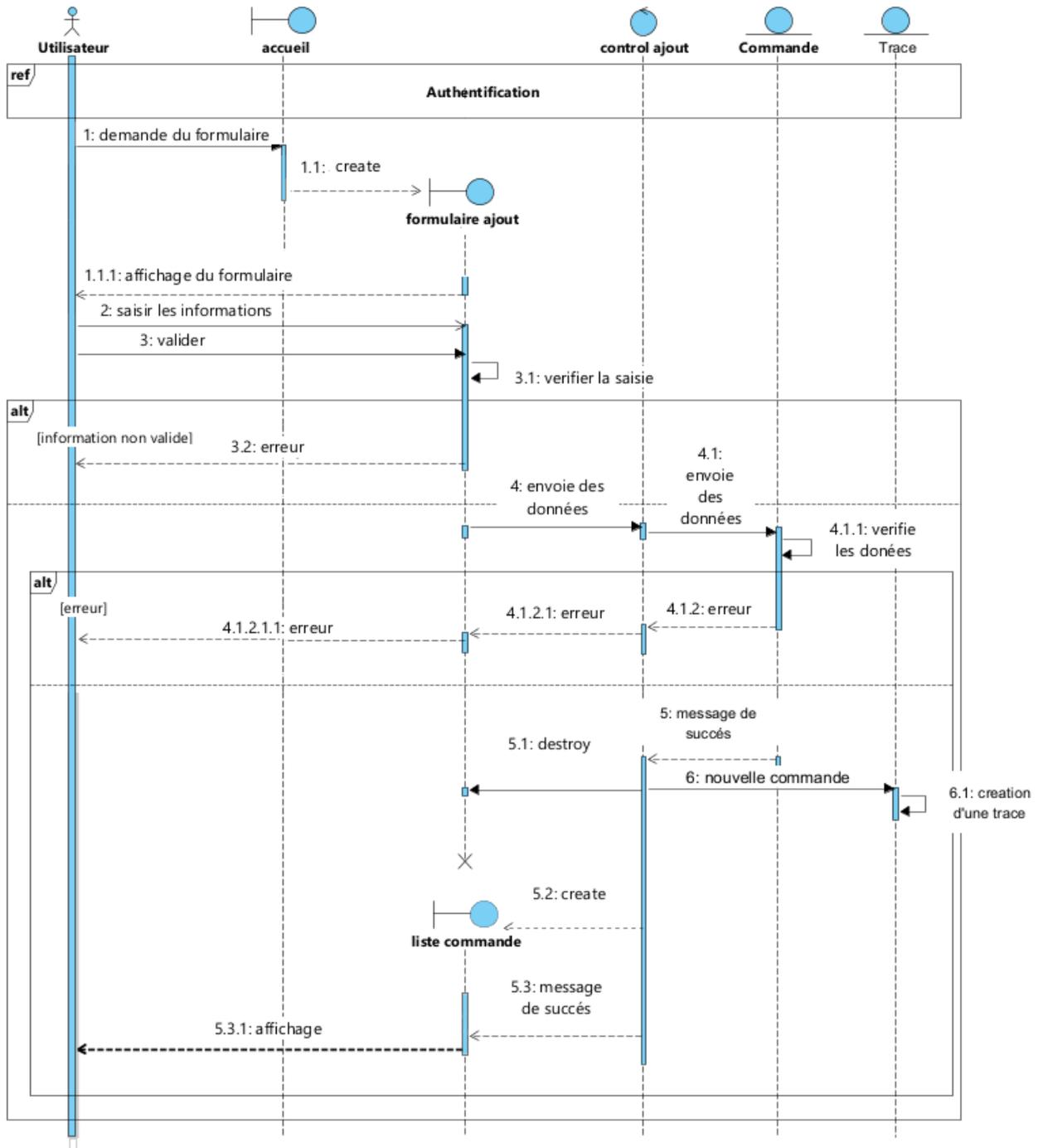


Figure 23 Diagramme de séquence détaillé du cas établir un bon de commande

• Diagramme de séquence détaillé du cas traitement d'un bon de commande

La figure suivante illustre le diagramme de séquence détaillé du cas traitement d'un bon de commande (Voir Figure 24).

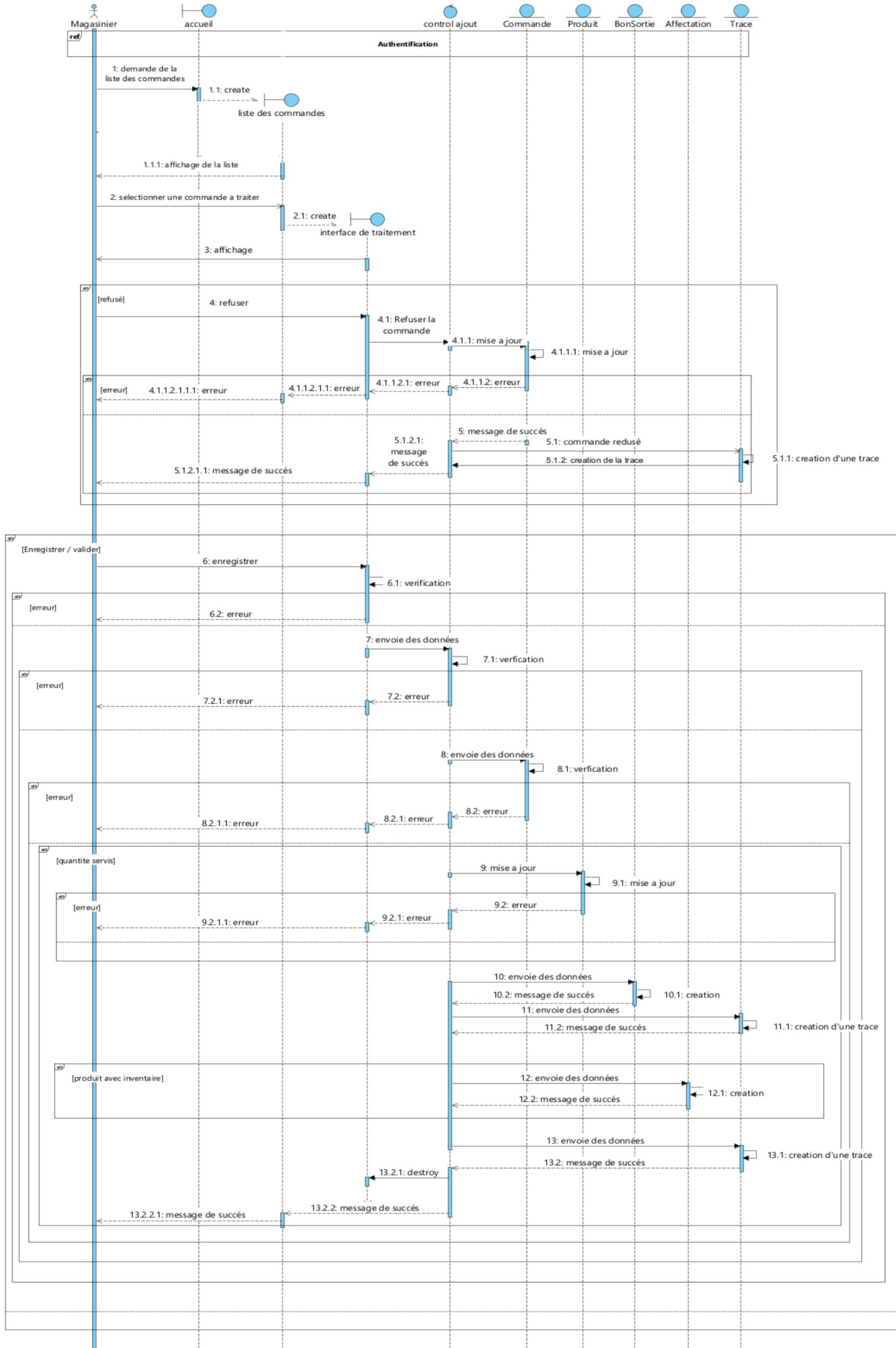


Figure 24 Diagramme de séquence détaillé du cas traitement d'un bon de commande

3.10 Diagramme de classes

Dans la figure ci-dessous est présenté le diagramme de classes de gestion des stocks (Voir Figure 25) :

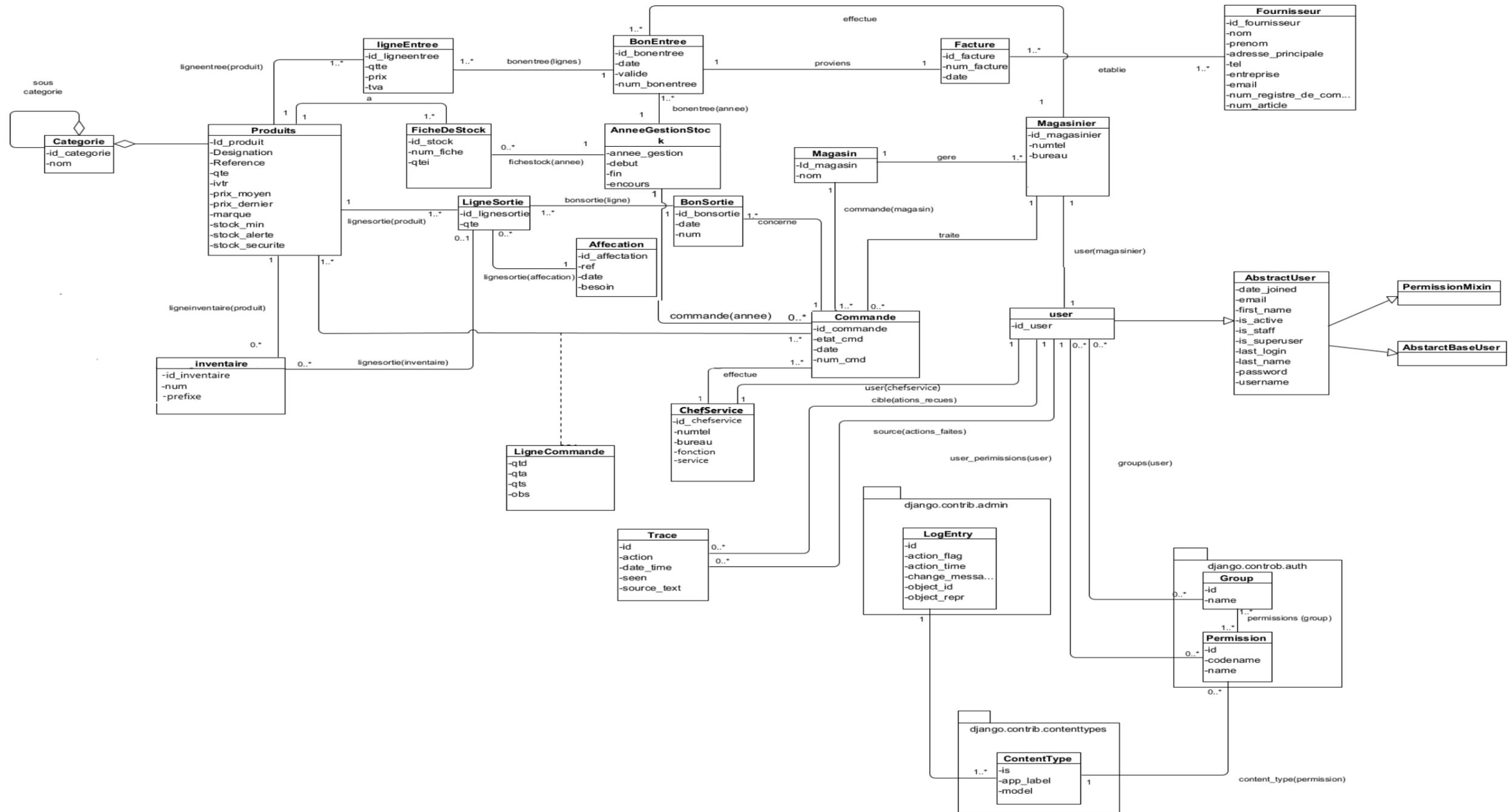


Figure 25 Diagramme de classes de gestion des stocks.

3.11 Glossaire des données

Le tableau suivant représente le dictionnaire des données représenté dans le diagramme de classe précédent :

Nom de la table	Attributs	Description	Type
Produit	Id_produit	Identifiant du produit	Numérique
	Designation	La désignation du produit	Chaîne de caractère
	Reference	La référence du produit	Chaîne de caractère
	Qte	La quantité du produit en stock	Booléen
	Inventaire	Si le produit a un numéro d'inventaire	Numérique
	Prix_moyen	Le prix moyen du produit	Float
	Prix_dernier	Le dernier prix du produit	Float
	Marque	Marque du produit	Chaîne de caractère
	Stock_min	Stock minimum	Numérique
	Stock_alerte	Stock d'alerte	Numérique
Stock_securite	Stock de sécurité	Numérique	
Categorie	Id_categorie	L'identifiant de la catégorie	Chaîne de caractère
	Nom	Le nom de la catégorie	Chaîne de caractère
Fournisseur	Id_fournisseur	Identifiant du fournisseur	Numérique
	Nom	Nom du fournisseur	Chaîne de caractère
	Prénom	Prénom du fournisseur	Chaîne de caractère
	Adresse_pricipale	Adresse de l'entreprise	Chaîne de caractère

	Tel	Numéro de téléphone	Numérique
	Entreprise	Le nom de l'entreprise	Chaîne de caractère
	Email	L'email du fournisseur	Email
	numregistrecommerce	Numéro de registre de commerce du fournisseur	Numérique
	Numarticle	Numéro d'article du fournisseur	Numérique
	Activite	L'activité du fournisseur	Chaîne de caractère
Facture	Id_facture	Identifiant de la facture	Numérique
	Num_facture	Numéro de la facture	Numérique
	Date	La date de l'établissement du bon	Date
BonEntree	Id_bonentree	Identifiant d'un bon d'entrée	Numérique
	Num_bonentree	Numéro du bon d'entrée	Numérique
	Date	La date de l'établissement du bon	Date
	Valide	Si le bon est validé ou non	Booléen
LigneEntree	Id_ligneentree	L'identifiant de ligne entrée	Numérique
	Qtte	Quantité entrante du produit	Numérique
	Prix	Prix unitaire du produit	Float
	Tva	TVA	Float
Inventaire	id_inventaire	Identifiant d'un modèle de numéro d'inventaire	Numérique
	Prefixe	Partie droite d'un numéro d'inventaire	Chaîne de caractère
	Num	Le numéro à gauche d'un numéro d'inventaire	Numérique

Commande	Id_commande	Identifiant d'une commande	Numérique
	Etat_cmd	L'état de la commande	Chaîne de caractère
	Date	Date de la commande	Date
	Num_cmd	Numéro de la commande	Numérique
LigneCommande	Qtd	Quantité demandé du produit par le service demandeur	Numérique
	Qta	Quantité accordé du produit par le magasinier	Numérique
	Qts	Quantité servis du produit par le magasinier	Numérique
	Obs	Observation	Chaîne de caractère
BonSortie	id_bonsortie	Identifiant du bon de sortie	Numérique
	Date	Date d'établissement du bon de sortie	Date
	num	Numéro du bon de sortie	Numérique
LigneSortie	id_lignesortie	Identifiant de la ligne de sortie	Numérique
	Qte	Quantité sortis d'un produit	Numérique
Affectation	id_affectation	Identifiant d'un certificat d'affectation	Numérique
	Ref	Référence d'un certificat d'affectation	Chaîne de caractère
	Date	La date d'établissement du certificat	Date
	Besoin	Pour quel besoin est établi ce certificat	Chaîne de caractère
	id_stock	Identifiant de la fiche de stock	Numérique

FicheDeStock	num_fiche	Numéro de la fiche de stock	Numérique
	Qtei	Quantité initiale du produit	Numérique
AnneeGestionStock	Annee_gestion	L'année de gestion de stock	Numérique
	Debut	Date de début de l'année de gestion de stock	Date
	Fin	Date de fin de l'année de gestion de stock	Date
	Encours	Indique si l'année est encours	Booléen
Magasinier	id_magasinier	Identifiant du magasinier	Numérique
	Numtel	Numéro de téléphone	Numérique
	Bureau	Le numéro du bureau	Numérique
Magasin	id_magasin	L'identifiant du magasin	Numérique
	Nom	Le nom du magasin	Chaîne de caractère
User	id_user	L'identifiant de l'utilisateur	Numérique
AbstractUser	date_joined	La date d'inscription de l'utilisateur	Date
	Email	L'adresse e-mail de l'utilisateur	Email
	first_name	Le prénom de l'utilisateur	Chaîne de caractère
	is_active	Indique si l'utilisateur est actif	Booléen
	is_staff	Indique si l'utilisateur est membre du personnel	Booléen
	is_superuser	Indique si l'utilisateur est un super-utilisateur	Booléen

	last_login	La date et l'heure de la dernière connexion de l'utilisateur	Date
	last_name	Le nom de l'utilisateur	Chaîne de caractère
	Password	Le mot de passe de l'utilisateur	Chaîne de caractère
	Username	Le nom d'utilisateur de l'utilisateur	Chaîne de caractère
Group	Id	Identifiant du groupe	Numérique
	Name	Nom du groupe	Chaîne de caractère
Permission	Id	Identifiant d'une permission	Numérique
	Codename	Le code de la permission	Chaîne de caractère
	Name	Le nom de la permission	
Trace	id_trace	Identifiant d'une trace	Numérique
	Action	L'action effectuée dans la trace	Chaîne de caractère
	date_time	La date et l'heure de la trace	Date
	Seen	Indique si la trace a été vue	Booléen
	source_text	Le texte source de la trace	Chaîne de caractère
ChefService	Id_chefservice	Identifiant du chef de service	Numérique
	service	Indique quel service	Chaîne de caractère
	fonction	Indique quelle fonction	Chaîne de caractère

Tableau 7 Glossaire des données

3.12 Modèle relationnel

À partir du diagramme de classe, nous procéderons à la création du modèle relationnel, qui est le modèle logique des données utilisées pour représenter les informations dans une base de données. Ce modèle offre une description abstraite de la structure des données. Pour décrire une relation, nous utilisons simplement son nom suivi des noms de ses attributs entre parenthèses.

3.12.1 Règles de passage au modèle relationnel

Règle1 : Présence de la multiplicité « * » sur un côté de l'association.

- Chaque classe se transforme en une relation.
- Chaque attribut de classe se transforme en un champ de relation.
- L'identifiant de la classe qui est associée à la multiplicité « 1 » devient une clé étrangère dans l'autre classe

Règle 2 : Présence de la multiplicité « * » sur les deux côtés.

- Chaque classe se transforme en une relation.
- Chaque attribut de classe se transforme en un champ de relation.
- L'association se transforme en une relation qui aura comme champs l'identifiant de chacune des deux classes (plus d'éventuels autres attributs).

Règle 3 : Présence de la multiplicité « 1..1 » « 0..1 » sur les côtés.

- Chaque classe se transforme en une relation.
- Chaque attribut de classe se transforme en un champ de relation.
- L'identifiant de la classe qui est associée à la multiplicité « 1..1 » devient clé étrangère dans l'autre classe (celle qui est associée à la multiplicité « 0..1 »).

Règle 4 : Présence d'une relation de composition.

- Chaque classe se transforme en une relation.
- Chaque attribut de classe se transforme en un champ de relation.
- L'identifiant de la classe qui est associée à la multiplicité « 1 » (classe qui représente le niveau ensemble) devient à la fois une clé étrangère et partie de la clé primaire dans l'autre classe.).

Règle 5 : Présence d'une généralisation-spécialisation

- Push-up : Créer une relation avec tous les attributs des classes et ajouter un attribut pour distinguer les types des objets.
- Push-down : Créer une relation pour chaque sous type et chaque relation se compose des attributs génériques et des attributs spécifiques.

- Distinction : Créer une relation par classe et les relier par des associations [22].

3.12.2 Passage au modèle relationnel

AnneeGestionStock (annee_gestion, debut, fin, encours)

Magasinier (id_magasinier, #user_id, numtel, bureau)

ChefService (id_chefservice, #user_id, numtel, bureau, fonction, service)

Categorie (id_categorie, nom)

SousCategorie (id_souscategorie, nom, #id_categorie)

Magasin (id_magasin, nom)

Produit (id_produit, designation, reference, ivtr, qte, prix_moyen, prix_dernier, #Categorie_id, #souscategorie_id, #id_magasin, marque, stock_min, stock_alerte, stock_securite)

FicheDeStock (id_stock, produit_id, #annee_gestion, num_fiche, qtei)

Fournisseur (id_fournisseur, nom, prenom, adresse_principale, tel, entreprise, email, numregistrecommerce, numarticle, activité)

Facture (id_facture, num_facture, date, #fournisseur_id, #annee_gestion)

BonEntree (id_bonentree, num_bonentree, date, # id _facture, valider, #annee_gestion, #user_id)

LigneEntree (id_ligneentree, #id_bonentree, #id_produit, qtte, prix, tva)

Commande (id_commande, date, _id, etat_cmd, annee_encours_id, , num_cmd, #magasin_id, #chefservice_id)

LigneCommande (#id_produit, #id_commande, qtd, qta, qts, obs)

BonSortie (id_bonsortie, date, num, #id_commande)

LigneSortie (id_lignesorte, #bon_id, # id_produit, qte, id_affectation)

Inventaire (id_inventaire, prefixe, #produit_id,)

LigneInventaire(id_ligneinventaire, num, #id_inventaire, #id_lignesortie)

Affectation (id_affectation, ref, date, besoin)

User (id_user, date_joined, email, first_name, is_active, is_staff, _superuser, last_login, last_name, password, username)

Trace (id_trace, action, date_time, seen, source_text, #cible, #source)

Permission (id_permission, codename, name, #content_type_id)

Group (id_group, name)

ContentType (id_contenttype, app_label, model)

LogEntry (id_logentry, #object_id, #content_type_id, #user_id, action_flag, change_message, action_time)

GroupPermissions(id_groupepermission, #group_id, #permission_id)

UserPermissions(id_userpermission, #user_id, #permission_id)

3.13 Conclusion

Ce chapitre nous a permis de définir une conception solide en utilisant les principes de l'UML. En mettant l'accent sur les diagrammes des cas d'utilisations qui nous ont permis d'identifier les différentes fonctionnalités attendues du système, les diagrammes de séquence qui nous ont permis de visualiser l'interaction entre les acteurs et le système dans différents scénarios. De plus, le diagramme de classes a joué un rôle essentiel en déduisant le schéma relationnel de notre future base de données.

Le chapitre suivant sera consacré à l'implémentation de notre solution. Nous mettrons en œuvre les différents éléments conçus lors de l'analyse et de la conception, en les traduisant en code et en configurant les composants nécessaires. Cela nous permettra de passer de cette phase à la phase de réalisation concrète du projet.

Chapitre 04

Réalisation

4.1 Introduction

Ce chapitre se focalise sur la concrétisation du projet en mettant en place les fonctionnalités identifiées lors de la phase de conception. Il représente une étape cruciale dans le processus de développement, car il permet de transformer les concepts et les spécifications en une application opérationnelle.

Dans cette section, nous aborderons les aspects pratiques de la mise en œuvre du projet en détaillant les choix technologiques, les outils utilisés. De plus, nous présenterons les interfaces de notre application web afin de mettre en évidence leurs praticités et leurs convivialités, qui ont été des objectifs primordiaux pour nous.

4.2 Langages utilisés

4.2.1 HTML5

Le HTML5, pour HyperText Markup Language 5, est un langage de balisage utilisé pour créer des pages Web. Il permet de décrire et de structurer le contenu des documents hypertextes destinés à être affichés sur un navigateur. Le HTML est un langage côté client, tout comme CSS et JavaScript. Il est développé et soutenu par le World Wide Web Consortium (W3C), l'organisme chargé de normaliser les technologies Web [23].



4.2.2 CSS 3

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique qui permet de gérer la présentation d'une page Web (HTML Ou XML). Le langage CSS est une recommandation du World Wide Web Consortium (W3C) [24].



4.2.3 JavaScript

C'est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre il est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web3.

Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript5 pour l'interpréter.

JavaScript est aussi employé pour les serveurs Web avec l'utilisation par exemple de Node.js ou de Deno [25].



4.2.4 Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [26].



4.3. Frameworks

4.3.1 Bootstrap

Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Il est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore [27].



4.3.2 Django

Django est un Framework Python open-source complet et gratuit. Il tente d'inclure toutes les fonctionnalités nécessaires par défaut, au lieu de les proposer sous forme de bibliothèques séparées. Parmi les fonctionnalités de Django, on peut citer l'authentification, le routage d'URL (Uniform Resource Locator), le moteur de modèles, l'ORM (Object Relationnel Mapper) et la migration des schémas de bases de données [28].



4.3.3 JQuery

JQuery est une bibliothèque JavaScript gratuite, libre et multiplateforme. Compatible avec l'ensemble des navigateurs Web (Internet Explorer, Safari, Chrome, Firefox, etc.), elle a été conçue et développée en 2006 pour faciliter l'écriture de scripts. Elle permet, entre autres, de gagner en rapidité dans l'interaction avec le code HTML d'une page Web. Elle propose comme principales fonctionnalités :



- la manipulation du Document Object Model (DOM).
- la gestion des événements (mouvements de souris, clics, etc.) et de l'AJAX (architecture informatique).
- la création d'effets d'animation.
- la manipulation des feuilles de style en cascade.

Et bien d'autres encore [29].

4.4 L'architecture MVT

L'architecture MVT (Modèle-Vue-Template) est une variante de l'architecture MVC (Modèle-Vue-Contrôleur) utilisée par le framework Django. Elle est conçue pour faciliter le développement d'applications web en séparant clairement les responsabilités et en favorisant la réutilisabilité du code.

• **Modèle**

Le modèle représente la structure de données de l'application. Il définit les objets, les attributs et les relations entre eux. En Django, les modèles sont généralement définis sous forme de classes Python, où chaque classe correspond à une table de la base de données. Les modèles sont responsables de la gestion des données, des opérations de lecture/écriture dans la base de données. Ils s'appuient sur un ORM (ObjectRelational Mapping).

• **Vue**

La vue est responsable de la logique de traitement des requêtes et de la génération des réponses. Elle récupère les données nécessaires à partir des modèles, effectue des opérations de traitement si nécessaire, et renvoie une réponse appropriée. En Django, les vues peuvent être définies comme des fonctions ou des classes, et sont associées à des URL spécifiques. Les vues peuvent également gérer les opérations d'authentification, les autorisations d'accès, etc.

• Template

Un template est un fichier HTML, aussi appelé en français « gabarit ». Il sera récupéré par la vue et envoyé au visiteur ; cependant, avant d'être envoyé, il sera analysé et exécuté par le framework, comme s'il s'agissait d'un fichier avec du code. Django fournit un moteur de templates très utile qui permet, dans le code HTML, d'afficher des variables, d'utiliser des structures conditionnelles (if/else) ou encore des boucles (for), etc [30].

La figure suivante représente l'architecture MVT (Voir Figure 26).

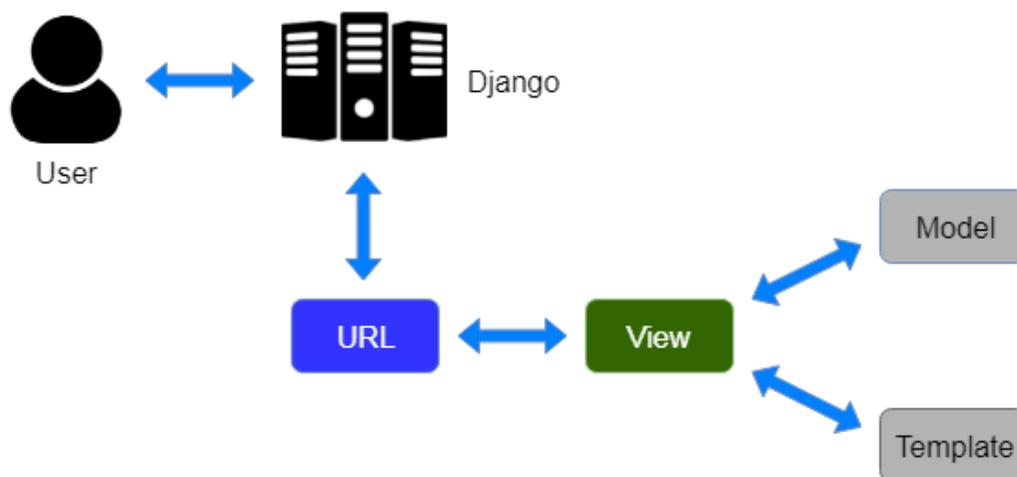


Figure 26 Architecture MVT

4.5 ORM

Le terme ORM vient de l'anglais Object-Relational Mapping. En français, on croise parfois (rarement) le terme mapping objet-relationnel.

Il s'agit d'une technique de programmation informatique qui permet de simplifier l'accès à une base de données en proposant à l'informaticien des « objets » plutôt que d'accéder directement à des données relationnelles [31].

4.5.1 SQLite

Est une bibliothèque écrite en C qui propose un moteur de base de données SQL (Structured Query Language). Contrairement aux serveurs de bases de données comme MySQL ou PostgreSQL, elle ne reproduit pas le schéma habituel client-serveur mais elle est directement intégrée aux programmes en utilisant des fichiers de bases de données.



Elle implémente en grande partie le standard SQL92 et les Propriétés ACID (atomique, cohérente, isolée et durable) [32].

4.6. Outils de développement et de conception

4.6.1 Pycharm

Est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django. &Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et GNU/Linux. Il est décliné en édition professionnelle, diffusé sous licence propriétaire, et en édition communautaire diffusé sous licence Apache [33].



4.6.2 TablePlus

TablePlus est un outil moderne et natif doté d'une interface utilisateur élégante qui permet de gérer simultanément plusieurs bases de données telles que MySQL, PostgreSQL, SQLite, Microsoft SQL Server et bien d'autres.



4.6.3 Visual Pradigm

Visual Paradigm (VP-UML) est un outil de modélisation UML CASE qui prend en charge les normes UML 2, SysML et Business Process Modeling Notation (BPMN) définies par le groupe de gestion d'objets (OMG). Il permet aux utilisateurs de créer des modèles UML et d'autres types de modèles visuels pour



4.7 Architecture de déploiement de la solution

La figure suivante représente l'architecture de déploiement de la solution (Voir Figure 27).

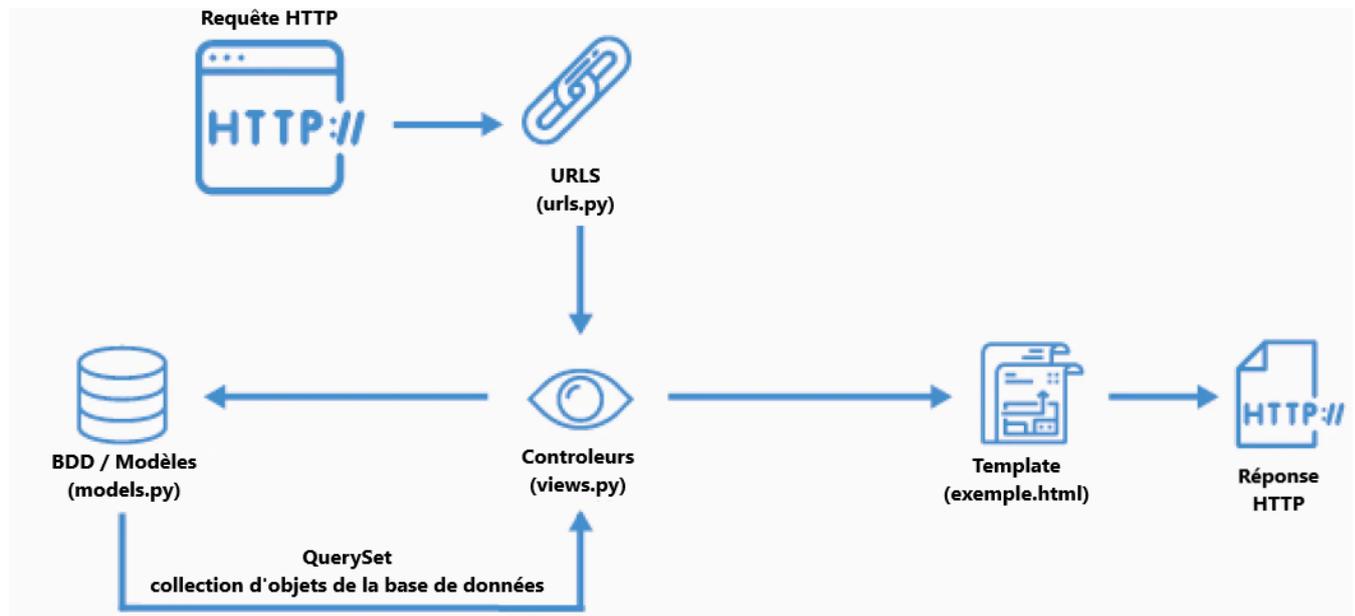


Figure 27 Architecture de déploiement de la solution

4.9 Organisation globale de l'application

La figure suivante représente l'organisation de l'application (Voir Figure 28).

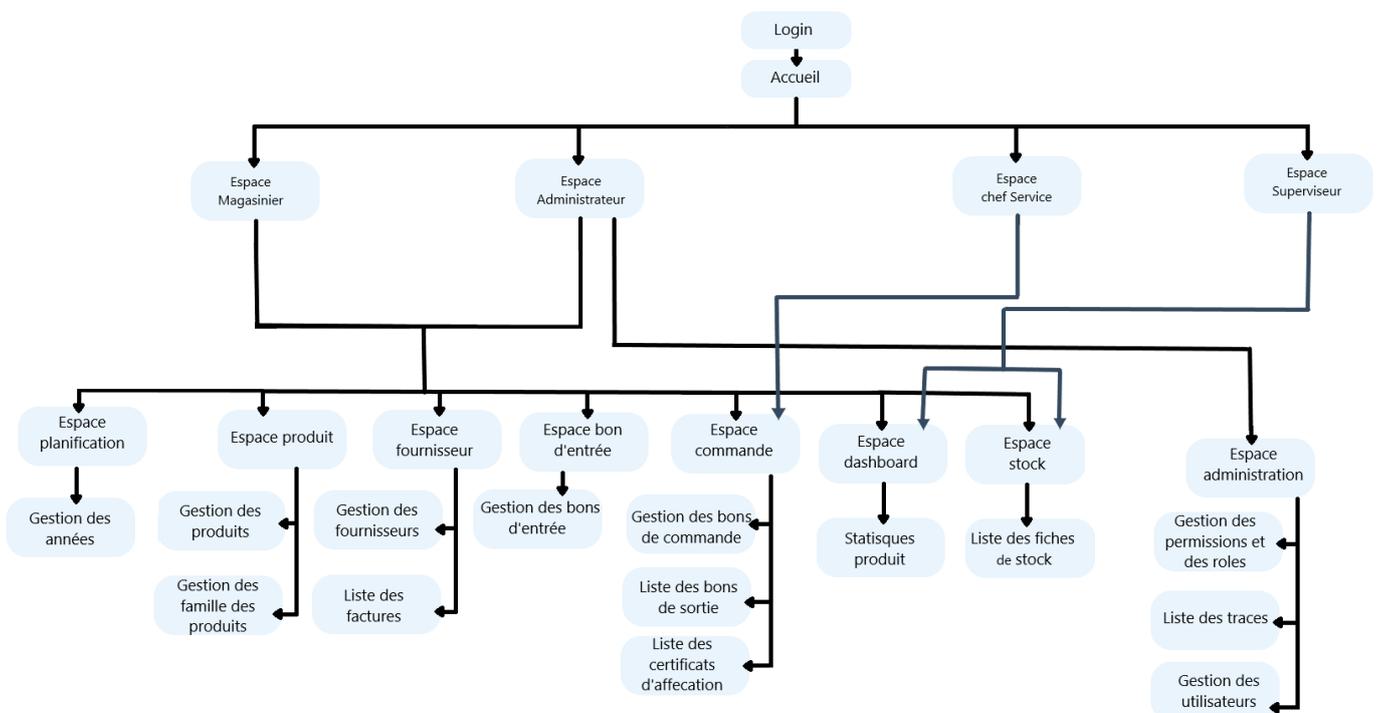


Figure 28 Architecture globale de l'application

4.10 Présentation de quelques interfaces

4.10.1 Interface d'accueil

C'est la première interface graphique après l'authentification d'un utilisateur. Il aura accès à certaines rubriques auxquelles il a la permission (Voir Figure 29).

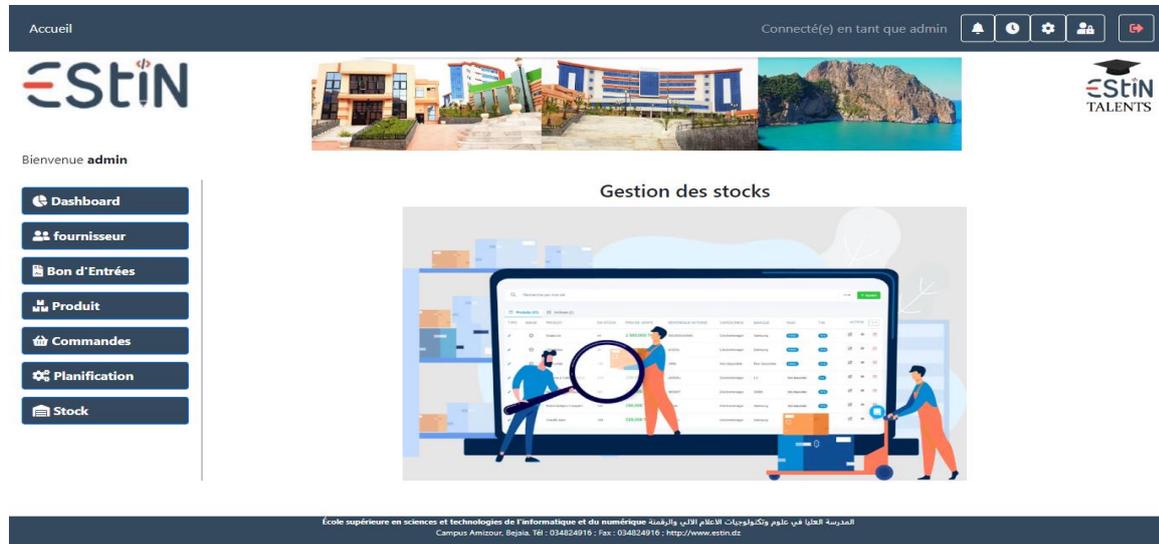


Figure 29 Interface d'accueil

4.10.2 Interface ajout d'un produit

Cette interface permet d'ajouter des nouveaux produits au système (Voir Figure 29).

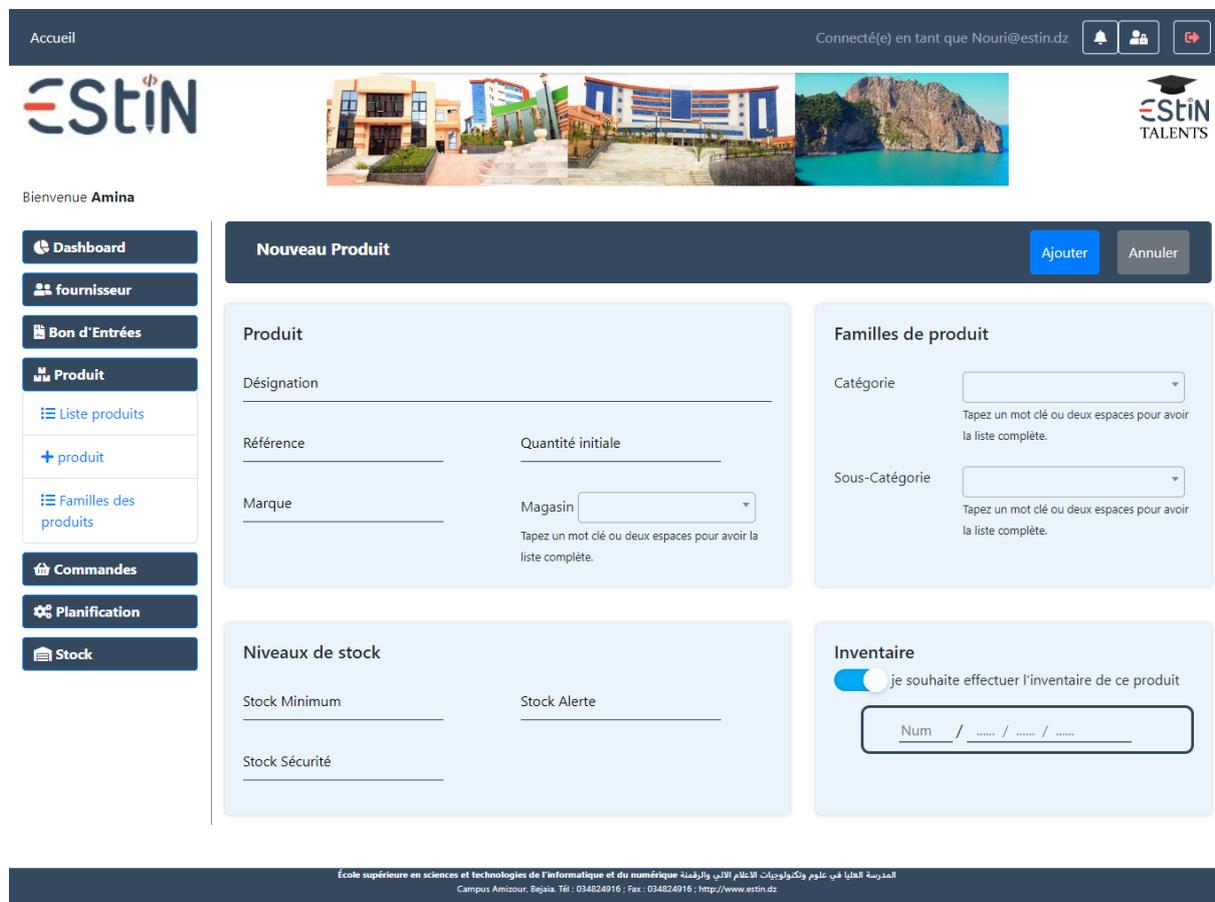


Figure 30 Interface ajout d'un produit

4.10.3 Interface liste des produits

L'interface "Liste des produits" regroupe tous les produits disponibles d'un magasin dans un tableau clair et organisé. Elle offre aux utilisateurs la possibilité de filtrer les produits, d'imprimer la liste, d'accéder aux interfaces d'ajout/modification/suppression/détail d'un produit et l'interface d'importation de produits à partir d'un fichier Excel (Voir Figure 31).

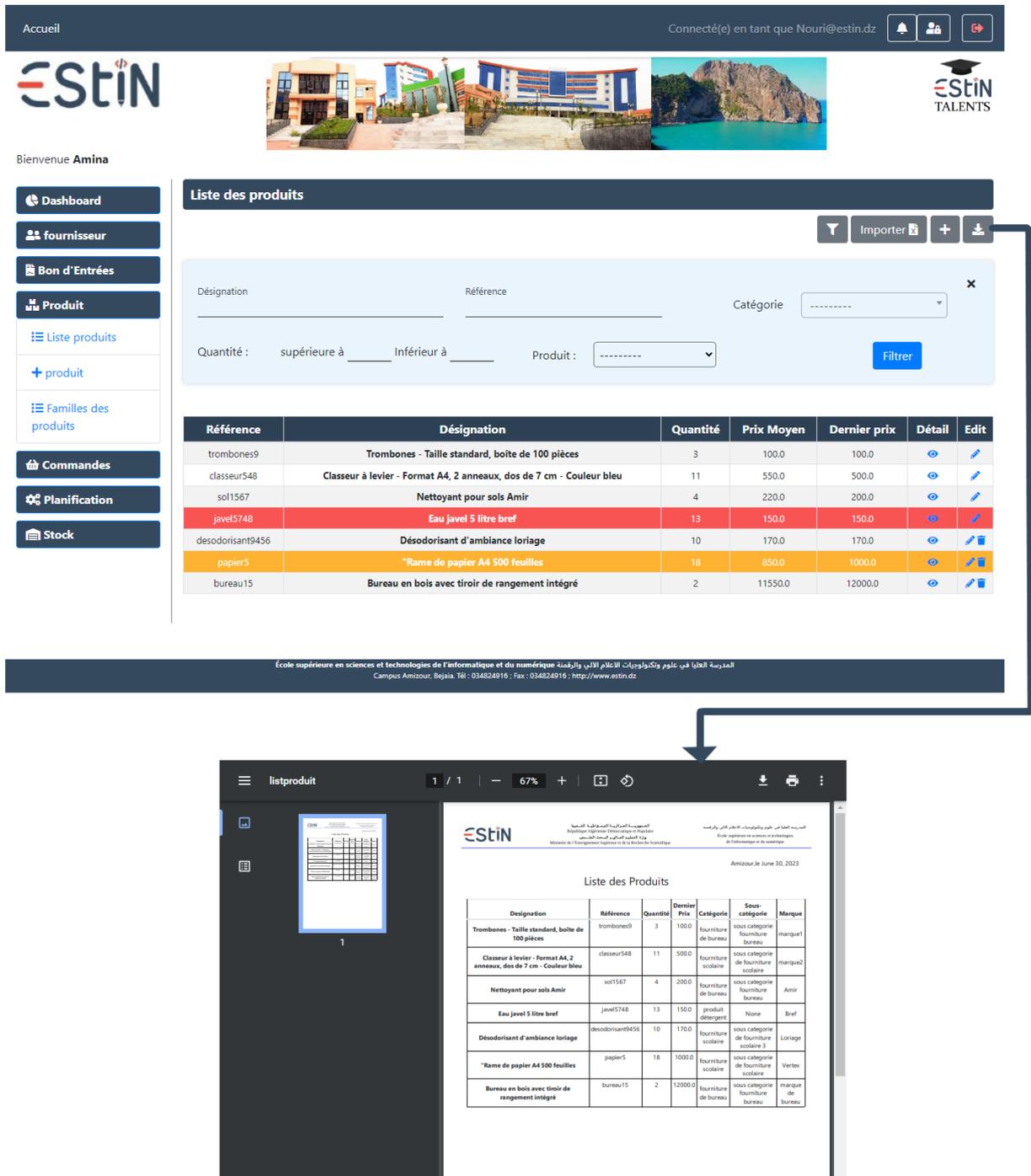


Figure 31 Interface liste des produits

La ligne en rouge représente un produit atteint les niveaux de stock d'alerte.
 La ligne en orange représente un produit atteint le niveau de stock minimum.

4.10.4 Interface importer des produits

Cette Interface permet aux utilisateurs d'importer facilement une liste de produits à partir d'un fichier Excel préalablement préparé, en plus de ça l'utilisateur peut télécharger un exemple de fichier Excel en cliquant sur le bouton de téléchargement (Voir Figure 32).

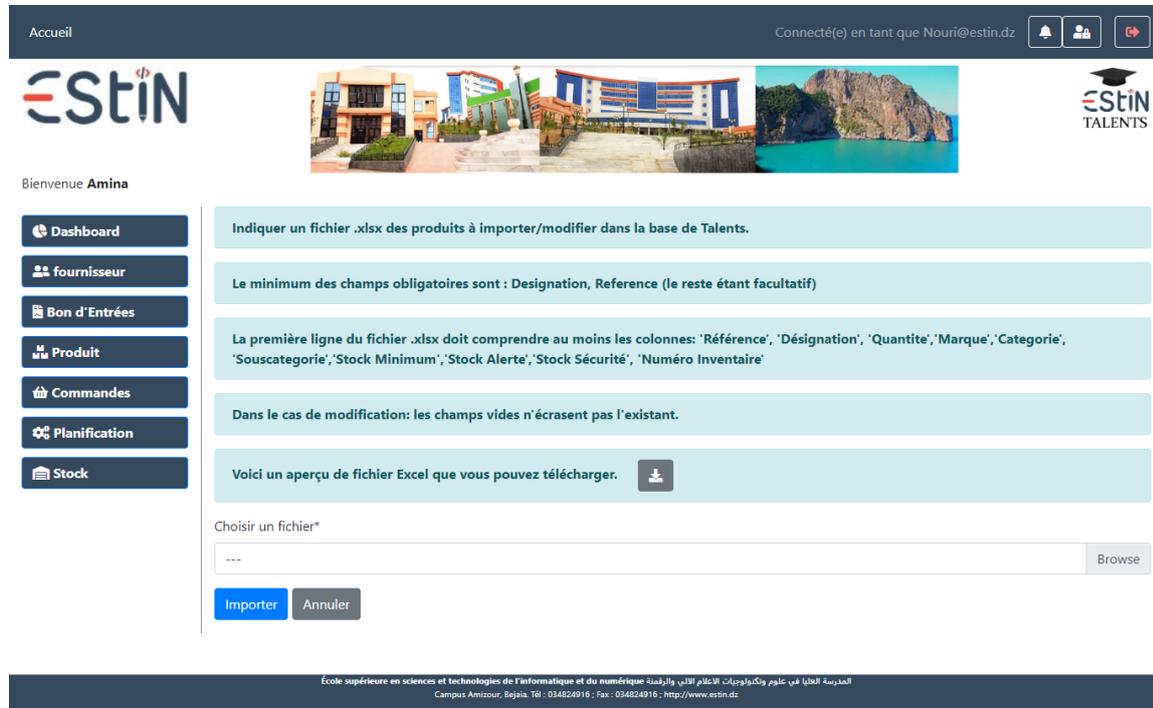


Figure 32 Interface Importer des produits

4.10.5 Interface détail d'un produit

Cette interface permet à l'utilisateur de visualiser les détails d'un produit (Voir Figure 33).

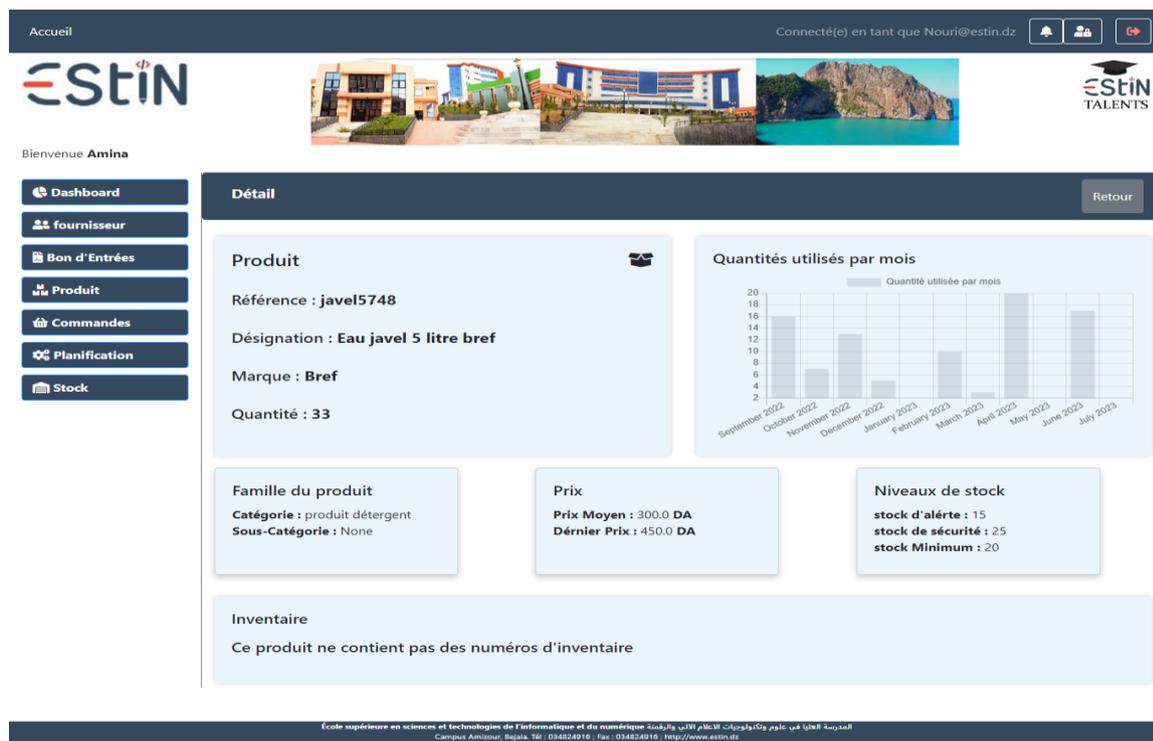


Figure 33 Interface détail d'un produit

4.10.6 Interface ajouter un bon d'entrée

L'interface "Ajouter un bon d'entrée" permet aux utilisateurs d'enregistrer les entrées de produits dans le système de gestion des stocks (Voir Figure 34).

Après avoir saisi les informations nécessaires, l'utilisateur a la possibilité de valider immédiatement le bon d'entrée en cliquant sur le bouton "Valider", ou d'enregistrer les données saisies pour les compléter ultérieurement en cliquant sur le bouton "Enregistrer". Cette flexibilité permet à l'utilisateur de gérer efficacement les entrées de produits selon ses besoins et son emploi du temps.

La facture est créée automatiquement avec le bon d'entrée.

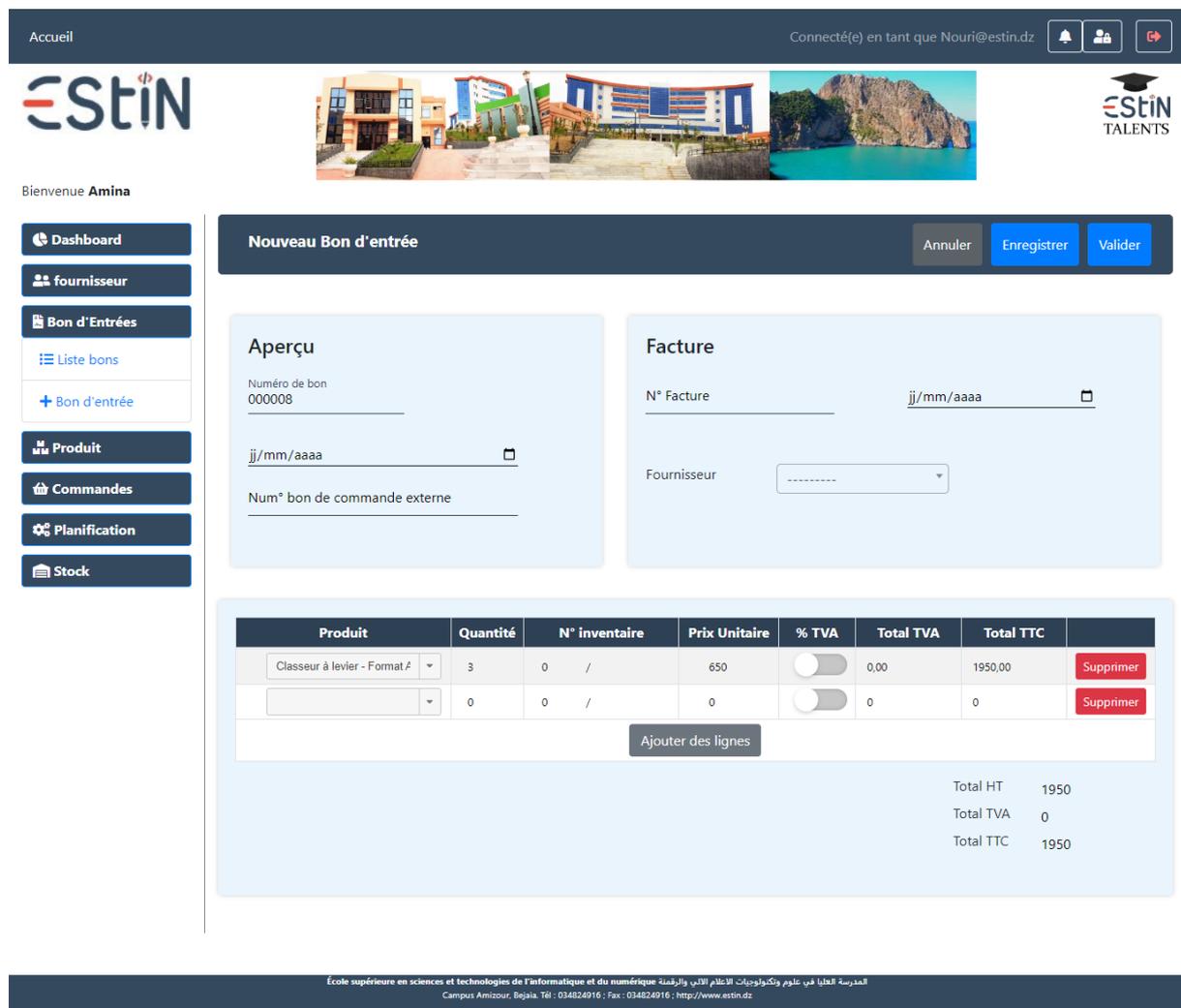


Figure 34 Interface ajouter un bon d'entrée

4.10.7 Interface liste des bons d'entrées

L'interface "Liste des bons d'entrées" regroupe tous les bons. Elle offre aux utilisateurs la possibilité d'accéder aux interfaces d'ajout/modification/suppression/détail d'un bon (Voir Figure 35).

Accueil Connecté(e) en tant que Nouri@estin.dz

Bienvenue Amina

Liste des Bons d'Entrées

chercher un produit +

N° Bon	Date	Facture	Fournisseur	Validé	Détail	Edt
1	Jan. 13, 2023	000001	zitoune yanis	✓	👁	
2	Dec. 7, 2022	02	zitoune yanis	✗	👁	✏ 🗑
3	April 14, 2023	4	zitoune yanis	✓	👁	
4	June 15, 2023	12345	zitoune yanis	✓	👁	
5	Feb. 14, 2023	1234544	Amari abdelhak	✓	👁	
6	Feb. 28, 2023	12345561	Amari abdelhak	✓	👁	
7	Nov. 13, 2022	5742	Amari abdelhak	✗	👁	✏ 🗑
8	Oct. 20, 2022	754	brahami Amine	✗	👁	✏ 🗑

École supérieure en sciences et technologies de l'informatique et du numérique المدرسة العليا في علوم وتكنولوجيا المعلومات والرقمنة
 Campus Amizour, Bejaia, Tél : 034824916 ; Fax : 034824916 ; http://www.estin.dz

Figure 35 Interface liste des bons d'entrées

4.10.8 Interface détail d'un bon d'entrée

Cette interface affiche toutes les informations spécifiques relatives à un bon d'entrée. Elle offre également la possibilité d'imprimer ce bon en cliquant sur un bouton dédié (Voir Figure 36).

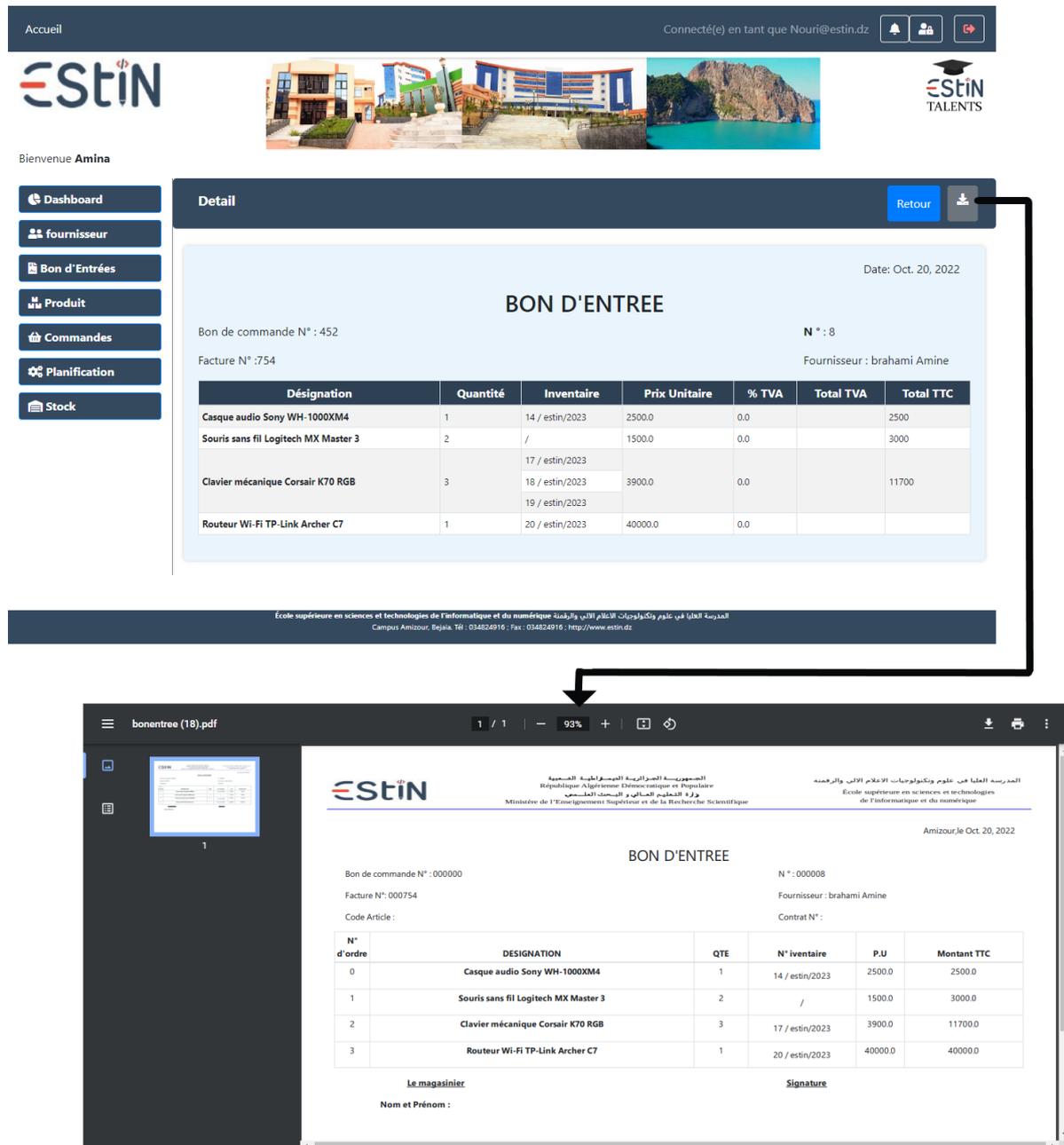


Figure 36 Interface détail d'un bon d'entrée

4.10.9 Interface établir un bon de commande

Cette interface permet aux utilisateurs la possibilité de créer un bon de commande en sélectionnant le magasin auprès duquel les produits seront commandés..

Une fois le magasin choisi, les utilisateurs peuvent sélectionner les produits souhaités à inclure dans la commande (Voir Figure 37).

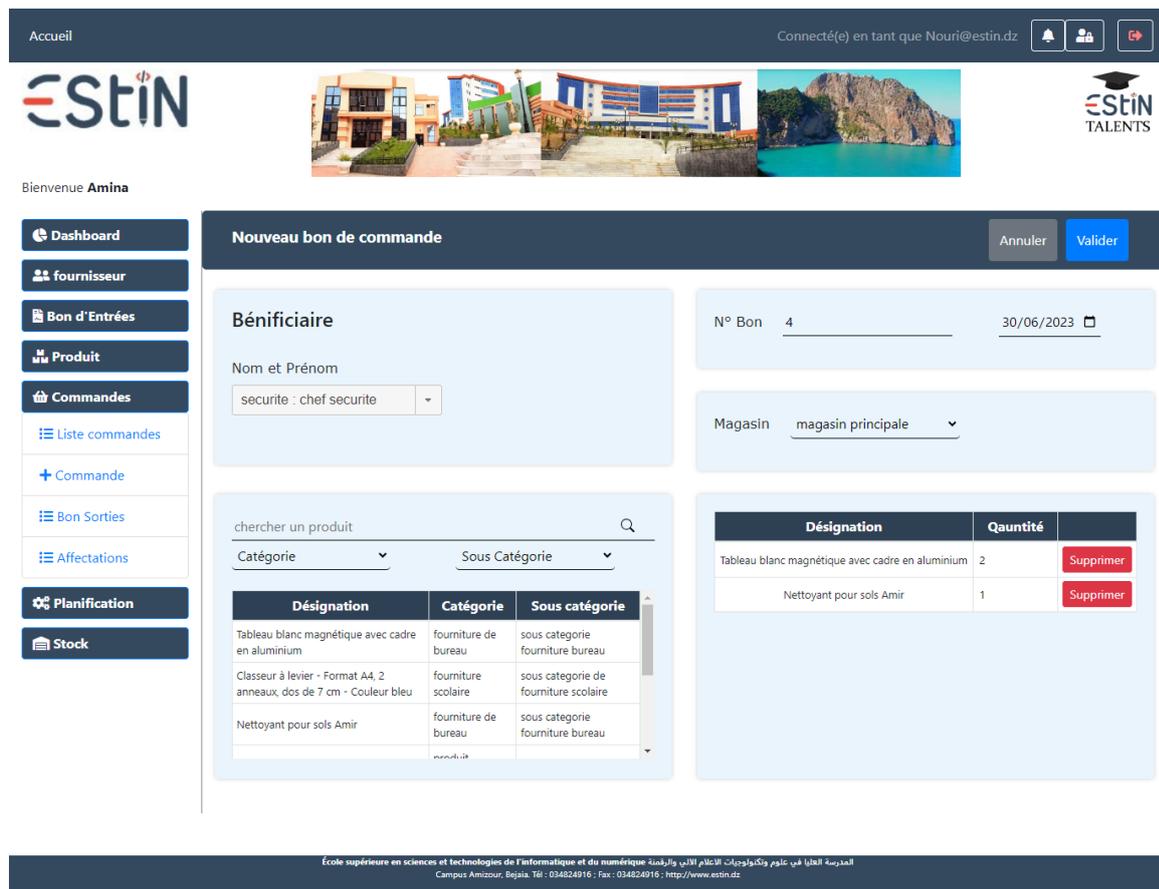


Figure 37 Interface établir un bon de commande

4.10.10 Interface traitement d'un bon de commande

Cette interface permet de traiter un bon de commande par le magasinier (Voir Figure 38).

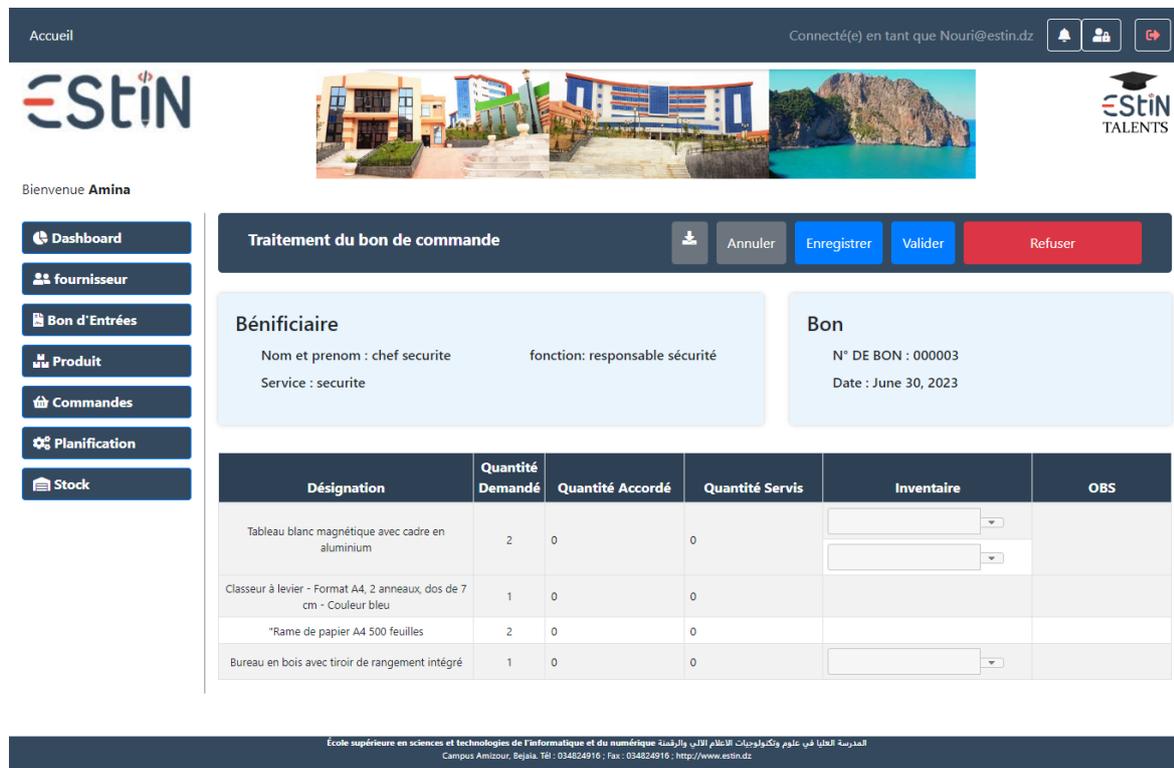


Figure 38 Interface traitement d'un bon de commande

4.10.11 Interface d'administration

Cette interface permet à l'utilisateur de gérer les rôles et les permissions et d'accéder à l'interface de gestion des utilisateurs et des magasins (Voir Figure 39).

The screenshot shows the StIN administration interface. At the top, there is a navigation bar with 'Accueil', 'Connecté(e) en tant que admin', and several utility icons. Below this is a header with the StIN logo, a large image of a building, and the StIN TALENTS logo. A sidebar on the left contains menu items: Dashboard, fournisseur, Bon d'Entrées, Produit, Commandes, Planification, and Stock. The main content area is titled 'Paramètres de Talents' and has tabs for 'Rôles et droits d'accès', 'Utilisateurs', and 'Magasin'. Under 'Rôles et droits d'accès', there is a '+ Rôle' button and a table titled 'Liste des rôles'.

Rôle	Nombre d'utilisateurs	Action
Administrateur	2	Liste des utilisateurs Supprimer
Magasinier	1	Liste des utilisateurs Supprimer
service	1	Liste des utilisateurs Supprimer
superviseur	0	Liste des utilisateurs Supprimer

Below the roles table is a section for 'Liste des droits d'accès' with buttons for 'Modifier les droits d'accès' and 'Rôles et accès supplémentaires par utilisateurs'. This leads to a table showing permissions for different roles.

Liste permissions - 1 permission(s)	Bloc : affectation			
	Administrateur	Magasinier	service	superviseur
Visualiser la liste des certificats d'affectation	✗	✓	✓	✗

Liste permissions - 3 permission(s)	Bloc : bonentree			
	Administrateur	Magasinier	service	superviseur
NAV Visualiser l'espace bon d'entree	✗	✓	✗	✗
Ajouter/Modifier/supprimer les bons d'entree	✗	✓	✗	✗

At the bottom of the page, there is a footer with contact information for the 'École supérieure en sciences et technologies de l'informatique et du numérique'.

Figure 39 Interface d'administration

4.10.12 Interface du Dashboard

Cette interface offre aux utilisateurs une vision globale des statistiques liées aux stocks de l'ESTIN. Elle permet de visualiser de manière synthétique les informations clés et les tendances relatives aux stocks (Voir Figure 40).

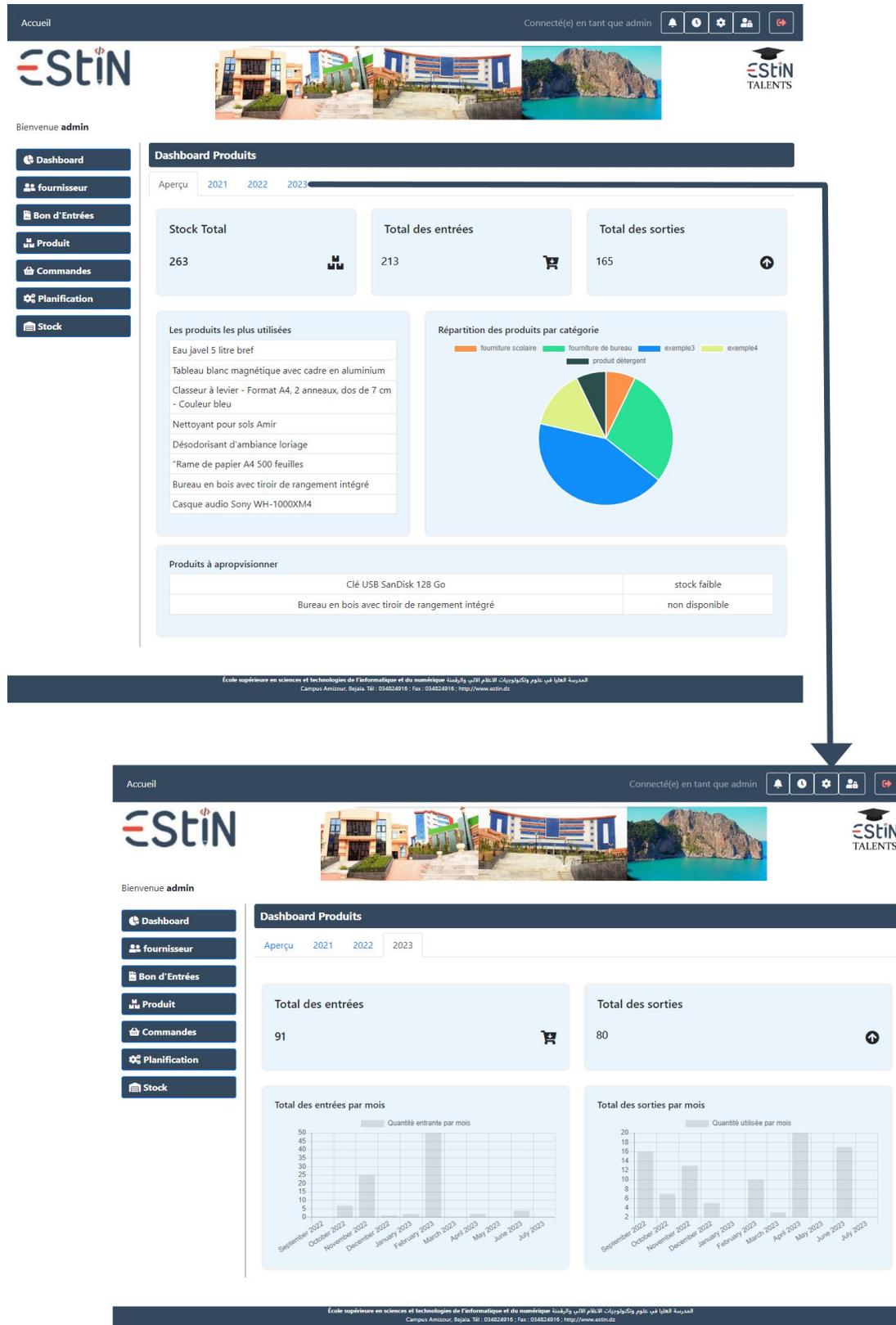


Figure 40 Interface Dashboard

4.11 Conclusion

Dans ce dernier chapitre, nous avons abordé les aspects pratiques de la réalisation de notre module de gestion des stocks de l'école ESTIN. En commençant par la présentation des divers outils de développement que nous avons utilisés, notamment le Framework Django qui nous a offert une solide gestion des données et qui a simplifié la création d'applications grâce à ses fonctionnalités telles que l'ORM, le routage d'URL, le moteur de templates et la sécurité intégrée.

Ensuite, nous avons présenté les résultats de notre application en illustrant les différentes interfaces graphiques qui ont été développées, mettant en évidence l'implémentation de certains cas d'utilisation mentionnés dans le chapitre 3. Il convient de souligner que nous avons tenu en compte des recommandations fournies par l'école, ce qui a grandement amélioré l'application.

Conclusion général

Conclusion général

Ce projet, qui s'inscrit dans le cadre d'un projet de fin de cycle en vue de l'obtention du diplôme de master Génie Logiciel, vise à présenter les différentes étapes de réalisation d'une application pour la gestion des stocks de l'école ESTIN. Au cours de ce mémoire, nous avons abordé en détail les méthodes et les outils utilisés pour concevoir et développer ce système de gestion.

Nous avons débuté par une présentation générale de la gestion des stocks, en définissant les termes et concepts clés, ainsi que les objectifs et les procédures associées. Nous avons également exploré différentes solutions informatiques couramment utilisées dans ce domaine, offrant ainsi un panorama des possibilités existantes.

Ensuite, nous avons effectué une étude approfondie de l'existant au sein de l'école ESTIN, mettant en évidence les lacunes et les problèmes qui nécessitaient des améliorations.

Cette analyse nous a permis de proposer une solution adaptée pour optimiser la gestion des stocks dans l'établissement.

Nous avons ensuite procédé à l'analyse et à la conception de notre système, en définissant les besoins fonctionnels et non fonctionnels, en identifiant les acteurs, en décrivant les différents cas d'utilisation à l'aide de diagrammes de séquences et en élaborant le diagramme de classe. Cette phase d'analyse et de conception a posé les bases solides pour la réalisation du système.

Enfin, nous avons abordé la phase de réalisation, où nous avons détaillé les outils utilisés et présenté quelques interfaces utilisateur développées. Cette étape a permis la concrétisation de notre projet et de mettre en évidence les résultats obtenus grâce à notre travail.

Ce mémoire a permis de proposer une solution complète de gestion des stocks pour l'école ESTIN en utilisant les avancées de l'informatique et en adaptant les meilleures pratiques du domaine. Ce système contribuera à une meilleure organisation des ressources de l'école en améliorant l'efficacité opérationnelle, en minimisant les erreurs et les coûts, et en permettant la prise de décisions éclairées basées sur des informations en temps réel. Ce projet a été une occasion d'approfondir nos connaissances et compétences dans le domaine de la gestion des stocks et de l'informatique, tout en nous confrontant aux défis réels de la conception et de la réalisation d'un système. Nous espérons que ce mémoire servira de référence et d'inspiration pour d'autres projets similaires, contribuant ainsi à l'amélioration continue des processus de gestion des stocks dans les établissements éducatifs et les organisations en général.

Bibliographie

Bibliographie

- [1] J. BELL BELL, Comptabilité Analytique et Contrôle de Gestion, Tome 1, Dunod, Éd., Paris, 1988.
- [2] J. SNOW, Indicateurs de suivi et d'évaluation de performance des systèmes logistiques, U. Deliver, Éd., 2007.
- [3] M. S. BAHLOUL, Gestion des stocks et approvisionnements, l'enseignement et la formation Eurl Pages Bleues Internationals, 2011.
- [4] «Qu'est-ce que la gestion des stocks ?,» [En ligne]. Available: <https://www.ibm.com/fr-fr/topics/inventory-management>. [Accès le 11 04 2023].
- [5] «GESTION DES STOCKS RESUME THEORIQUE,» 2014. [En ligne]. Available: <https://4gestionacademy.com/wp-content/uploads/2014/03/Gestion-de-stocks.pdf>. [Accès le 15 Mars 2023].
- [6] «Qu'est-ce que la technologie et l'informatique basees sur le cloud?,» 24 Mars 2021. [En ligne]. Available: <https://anyconnector.com/fr/cloud-based.html#qu-est-ce-que-la-technologie-et-l-informatique-basees-sur-le-cloud>. [Accès le 20 Mars 2023].
- [7] «Comment bien choisir son Logiciel de Gestion de Stock – ERP,» [En ligne]. Available: <https://abcsupplychain.com/fr/choisir-logiciel-gestion-de-stock-erp/>. [Accès le 21 03 2023].
- [8] «Zoho Inventory : en résumé,» [En ligne]. Available: <https://www.appvizer.fr/operations/gestion-de-stock/zoho-inventory>. [Accès le 21 Mars 2023].
- [9] «zoho,» 1996. [En ligne]. Available: <https://www.zoho.com/fr/inventory>. [Accès le 21 Mars 2023].
- [10] «gestion-stocks,» [En ligne]. Available: <https://www.gestion-stocks.com/notre-solution/>. [Accès le 20 Mars 2023].
- [11] «Odo,» [En ligne]. Available: https://www.odoo.com/fr_FR/app/inventory. [Accès le 25 Mars 2023].
- [12] «ESTIN,» [En ligne]. Available: <https://estin.dz/>. [Accès le 29 Mars 2023].
- [13] S. S. BENBRAHIM Adam, OpenTalents : Plateforme numérique pour la gestion de la pédagogie et la scolarité dans les établissements universitaires, 2021/2022.
- [14] «Processus unifié,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Processus_unifi%C3%A9. [Accès le 2023 mars 16].
- [15] «UP : Unified Process,» [En ligne]. Available: <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/>. [Accès le 9 Mars 2023].
- [16] P.-A. Sunier, «Méthodologies de développement de logiciels de gestion,» [En ligne]. Available: <https://gisnt.org/pdf/m%C3%A9thodologie%20de%20d%C3%A9veloppement%20de%20logiciel%20gestion.pdf>. [Accès le 6 mars 2023].
- [17] «UML, le langage de modélisation objet unifié,» [En ligne]. Available: <https://laurent-piechocki.developpez.com/uml/tutoriel/lp/cours/>. [Accès le 4 Mars 2023].
- [18] «Chapitre I : le langage UML et le processus unifié,» [En ligne]. Available: <http://dSPACE.univ-tlemcen.dz/bitstream/112/5500/5/chapitre1.pdf>. [Accès le 6 mars 2023].

Bibliographie

- [19] «Les processus unifiés et UML : Présentation d'UML,» [En ligne]. Available: <https://wikimemoires.net/2019/12/les-processus-unifies-et-uml-presentation-dum/>. [Accès le 29 Mars 2023].
- [20] «UML en français,» [En ligne]. Available: <http://uml.free.fr/cours/i-p5.html>. [Accès le 1 Avril 2023].
- [21] J.STEFFE, COURS UML, Bordeaux, 2005.
- [22] D. HAMZA et D. MOHAMMEDI, Règles de passage au modèle relationnel, 2020/2021.
- [23] «HTML (HyperText Markup Language),» [En ligne]. Available: <https://developper.mozilla.org/fr/docs/Web/HTML>. [Accès le 16 Avril 2023].
- [24] «CSS,» [En ligne]. Available: <http://glossaire.infowebmaster.fr/css/>. [Accès le 16 Avril 2023].
- [25] «Javascript,» [En ligne]. Available: <https://fr.wikipedia.org/wiki/JavaScript#:~:text=Une%20grande%20majorit%C3%A9%20des%20sites,j%20ou%20de%20Deno..> [Accès le 16 Avril 2023].
- [26] «Python : définition et utilisation de ce langage informatique,» [En ligne]. Available: <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/>. [Accès le 17 Avril 2023].
- [27] «Bootstrap : définition, tutoriels, astuces, pratiques,» [En ligne]. Available: <https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/>. [Accès le 17 Avril 2023].
- [28] «Les 6 meilleurs frameworks Python,» [En ligne]. Available: <https://www.codeur.com/blog/meilleurs-frameworks-python/>. [Accès le 18 Avril 2023].
- [29] «jQuery : définition simple,» [En ligne]. Available: <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203587-jquery-definition/>. [Accès le 17 Avril 2023].
- [30] «Le fonctionnement de Django,» [En ligne]. Available: https://zestedesavoir.com/tutoriels/598/developpez-votre-site-web-avec-le-framework-django/262_presentation-de-django/1517_le-fonctionnement-de-django/. [Accès le 18 Avril 2022].
- [31] «ORM,» [En ligne]. Available: <https://www.base-de-donnees.com/orm/>. [Accès le 10 avril 2023].
- [32] «SQLite : qu'est-ce que c'est ?,» [En ligne]. Available: <https://www.futura-sciences.com/tech/definitions/internet-sqlite-4639/>. [Accès le 10 Avril 2023].
- [33] «PyCharm,» [En ligne]. Available: <https://fr.wikipedia.org/wiki/PyCharm>. [Accès le 22 Avril 2023].
-

Résumé

Ce projet, qui s'inscrit dans le cadre d'un projet de fin de cycle en vue de l'obtention du diplôme de master Génie Logiciel, vise à présenter les différentes étapes de réalisation d'une application pour la gestion des stocks de l'école ESTIN. Cette solution propose des fonctionnalités essentielles telles que la gestion des bons d'entrée, des commandes, des bons de sortie, des certificats d'affectation et la génération de fiches de stock détaillées.

Grâce à l'utilisation d'UML pour modéliser la solution et aux outils de développement tels que Python, JQuery, Bootstrap et Django, nous avons pu réaliser un système complet qui optimise la gestion des approvisionnements et des inventaires, assurant une traçabilité précise et une utilisation efficace des stocks disponibles.

Mots clés : UML, python, gestion des stocks, Bootstrap, Django et JQuery.

Abstract

This project, which is part of a final cycle project for the completion of a Master's degree in Software Engineering, aims to present the different stages of developing an application of a stock management for ESTIN School. This solution offers essential functionalities, including managing entry orders, purchases, exit orders, assignment certificates and generating detailed stock reports.

By utilizing UML for modeling the solution and development tools like Python, JQuery, Bootstrap, and Django, we were able to create a comprehensive system that optimizes supply chain management and inventory control, ensuring accurate traceability and efficient utilization of available stocks.

Keywords : UML, Python, Stock management, Web, Bootstrap framework, Django and JQuery .