

Université mouloud Mammeri de Tizi-Ouzou
Faculté de Génie Electrique et d'Informatique
Département d'Informatique



**Mémoire fin d'études en vue de l'obtention du diplôme de master en
informatique**

Option : Ingénierie des systèmes d'information

THÈME

**Développement d'un système de gestion de stocks et
d'approvisionnements dans le cadre d'un Web ERP**

RÉALISÉ PAR :

MOKHTARI Youcef
NAIT KACI Elyes

ENCADRÉ PAR :

Mr M.SI-MOHAMMED

Promotion 2019/2020

REMERCIEMENTS

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui on voudrait témoigner toute notre gratitude.

On voudrait dans un premier temps adresser toute notre gratitude à notre directeur de mémoire Mr. Si Mohammed, professeur en informatique, pour son encadrement, sa patience et ses conseils, qui ont contribué à alimenter notre réflexion.

On remercie également toute l'équipe pédagogique du département informatique de l'université Mouloud Mammeri Tizi Ouzou, à tous les enseignants, intervenants et toutes les personnes qui par leurs écrits, leurs conseils, leurs paroles et leurs critiques nous ont guidés tout au long de nos études universitaires et dans nos recherches.

On aimerait exprimer notre gratitude à tous les amis et camarades qui nous ont apporté soutien moral et intellectuel tout au long de notre aventure.

Enfin, merci à nos familles qui nous ont soutenu bien au-delà de ce mémoire.

Dédicaces

Je dédié ce travail à :

Mes très chers parents, pour leurs sacrifices, leur amour, leurs conseils précieux, leur soutien et leurs encouragements tout au long de mes études.

Mes très chères sœurs pour leur appui et leur soutien moral.

Toute ma famille.

Mon cher binôme.

Mes amis sans exception.

La promotion ISI 2019/2020.

Toute l'équipe NEWIDEA IT, Pour leur accueil et leurs conseils.

Je vous dédié ce travail en vous adressant toutes mes salutations chaleureuses, je vous souhaite un avenir radieux, je vous remercie tous, votre soutien et vos encouragements me donnent la force de continuer à travailler.

MOKHTARI Youcef

Dédicaces

Je dédie ce travail à :

Mes chers parents, qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de patience et de persévérance. Ceux que personne ne peut compenser les sacrifices qu'ils ont consentis pour mon éducation et mon bien-être tout au long de ma formation.

Mon frère et ma sœur qui m'ont accordé leurs encouragements, leurs soutiens dans les moments difficiles.

Mon très cher binôme et amis ainsi que tous mes camarades de la promotion 2019/2020.

L'équipe NEWIDEA IT, pour leur accueil chaleureux et leurs conseils.

Toute ma famille et toutes les personnes que j'aime.

NAIT KACI ELYES

Sommaire

Introduction Générale.....	1
Chapitre 1.....	2
Solutions web pour les entreprises	2
Introduction	3
I. Concept d'entreprise	3
I.1. Définition de l'entreprise [1].....	3
I.2. Environnement économique [2]	3
I.3. L'évolution des Tic dans les entreprises [3]	4
II. ERP.....	5
II.1. Historique [4]	5
II.2. Définition [5]	5
II.3. Avantages liés à l'intégration d'un ERP [6]	5
II.3.1. Aspect administratif [7]	6
II.3.2. Aspect opérationnel [7].....	6
II.4. Inconvénients.....	7
II.5. Fonctionnalités.....	8
Conclusion	10
CHAPITRE II.....	11
Introduction	12
I. Gestion de stock	12
I.1. La gestion [10]	12
I.2. Le stock [11]	12
I.3. Typologie de stock [12]	12
I.3.1. Types de stocks selon leur fonction	12
I.3.1.1. Stock de sécurité	12
I.3.1.2. Stock d'alerte	13
I.3.1.3. Stock saisonnier	13
I.3.1.4. Stock dormant (ou mort).....	13
I.3.1.5. Stock en transit.....	13
I.3.1.6. Stock cycle	13
I.3.1.7. Stock d'anticipation - spéculation	13
I.3.2. Types de stocks selon l'organisation opérationnelle	13
I.3.2.1. Stock optimal	13
I.3.2.2. Stock physique.....	13
I.3.2.3. Stock net	14
I.3.2.4. Stock disponible.....	14
I.3.2.5. Stock minimum	14
I.3.3. Types de stocks selon la date de péremption.....	14
I.3.3.1. Stock périssable	14
I.3.3.2. Stock non périssable	14

I.3.3.3. Stock avec date de péremption.....	14
I.4. L'utilité des stocks [13]	14
I.5. Gestion de stocks	15
I.5.1. Définition [14]	15
I.5.2. Procédures gestion de stocks [15]	15
I.5.3. Documents gestion de stocks [15]	16
I.5.4. Méthodes de valorisation de stock [16] [17]	17
I.5.4.1. Le coût moyen unitaire pondéré	17
I.5.4.2. La méthode FIFO.....	17
I.5.4.3. La méthode LIFO.....	17
I.5.4.4. La méthode NIFO.....	18
I.5.4.5. L'utilisation des prix par lots.....	18
I.5.5. Méthode de classement des stocks	18
I.5.5.1. La méthode ABC	18
I.5.5.1.1. Principes et intérêt de la méthode [18].....	18
I.5.5.1.2. Les étapes de la méthode [18].....	18
I.5.5.1.3. Limites et précautions d'emploi du classement ABC [19]	19
I.5.5.2. La méthode 20/80 [20]	19
I.5.6. Les enjeux de la gestion des Stocks [21]	19
I.5.6.1. La réduction des coûts :	19
I.5.6.2. L'élimination des ruptures de stock :	19
I.6. Les inventaires [22]	20
I.6.1. L'inventaire permanent	20
I.6.2. L'inventaire annuel.....	20
I.6.3. L'inventaire tournant.....	20
II. Gestion d'approvisionnement [15].....	21
II.1. Objectifs [15]	21
II.2. Tâches [15]	21
II.3. Le processus traditionnel d'approvisionnement [21].....	21
II.3.1. Détermination des besoins	21
II.3.2. Choix des fournisseurs	21
II.3.3. La commande	22
II.3.3.1. La demande d'achat	22
II.3.3.2. La commande	22
II.4. Les Méthodes de réapprovisionnements	22
II.4.1. Approvisionnement à Date et Quantité fixes [23]	22
II.4.2. Approvisionnement à Date fixe et Quantité variable [23].....	23
II.4.3. Approvisionnement à Date variable et Quantité fixe [23].....	23
II.4.4. Approvisionnement à Date et Quantité variables [23]	24
II.4.5. Le drop shipping [24].....	24
III. Logiciels de gestion de stock	24
III.1. Définition.....	24
III.2. Intérêt des logiciels de gestion de stock.....	24
Conclusion	25

CHAPITRE III.....	26
Introduction	27
I. Présentation d'UML	27
II. Analyse et conception	27
II.1. Spécification des besoins	27
II.1.1. Besoins fonctionnels	27
II.1.2. Besoins non fonctionnels	30
II.2. Spécification des acteurs de l'application.....	30
II.2.1. Les acteurs qui interviennent dans notre système	30
II.2.2. Diagramme de contexte	30
II.3. Spécification de tâches	31
II.4. Diagramme de cas d'utilisation	34
II.4.1. Diagramme de cas d'utilisation de l'application	34
II.5. Diagramme de séquence	43
II.6. Diagramme d'activité [25]	48
II.7. Diagramme de classe [26].....	51
II.8. Conception de la base de données	62
II.8.1. Modèle relationnel.....	62
II.8.2. Modèle physique.....	63
Conclusion.....	72
Chapitre IV	73
Introduction	74
I. Présentation de l'environnement de travail	74
I.1. Laravel	74
I.1.2. Présentation des outils de développement	74
I.1.2.1. Git - GitHub	74
I.1.2.2. Laragon	75
I.1.2.3. Apache	75
I.1.2.4. MySQL.....	75
I.1.2.5. HeidiSQL.....	75
I.1.2.6. PHP.....	75
I.1.2. Installation.....	75
I.1.2.1. Installation de Laravel avec Composer.....	75
I.1.2.1.1 Installation de Composer sur MacOS / Linux.....	75
I.1.2.1.2 Installation de Composer sur Windows.....	76
I.1.2.2. Installation de Laravel avec Laragon	76
I.1.2.3. Installation de Git	77
I.1.2.3.1. Installation de Git sur Linux	77
I.1.2.3.2. Installation de Git sur Mac OS / Windows.....	78
I.1.3. Organisation de Laravel [27]	79
I.2. MVC.....	81
I.2.1. Les routes	81
I.2.2. Contrôleurs.....	82

I.2.3. Modèles et migrations	84
I.2.4. Les vues	88
I.3. Livewire [27].....	89
II. Présentation de l'application	89
II.1. Espace administrateur	90
II.2. Espace financier	90
II.3. Espace magasinier.....	91
II.4. Interfaces de l'application.....	92
II.4.1. Gestion des paramètres	92
II.4.2. Produits	96
II.4.3. Devis	98
II.4.4. Bon de commande	98
II.4.5. Facture.....	99
II.4.6. Bon de réception	100
II.4.7. Bon de retour	100
II.4.8. Stock	101
Conclusion	101
Conclusion générale.....	102
Bibliographie et webographie	103

Liste des figures

Figure 1. Taux du dépassement de budget lors de l'implémentation d'un ERP	7
Figure 2. Taux des dépassements des délais lors de l'implémentation d'un ERP.....	8
Figure 3. Exemple de fiche de stock [29].....	16
Figure 4. Diagramme de contexte	30
Figure 5. Diagramme de cas d'utilisation global.....	35
Figure 6. Diagramme cas d'utilisation gestion des articles.....	36
Figure 7. Diagramme cas d'utilisation gestion de commandes d'achats.....	37
Figure 8. Diagramme cas d'utilisation gestion d'entrepôt	38
Figure 9. Diagramme cas d'utilisation gestion de réapprovisionnements	39
Figure 10. Diagramme cas d'utilisation gestion des réceptions	40
Figure 11. Diagramme cas d'utilisation gestion société et utilisateurs	41
Figure 12. Diagramme cas d'utilisation gestion des factures	42
Figure 13. Diagramme cas d'utilisation gestion des fournisseurs	43
Figure 14. Diagramme de séquence pour ajouter un article	44
Figure 15. Diagramme de séquence pour réceptionner ou retourner des articles.....	45
Figure 16. Diagramme de séquence cas confirmer un achat	46
Figure 17. Diagramme de séquence cas planifier un réapprovisionnement	47
Figure 18. Diagramme d'activité ajouter un règlement.....	48
Figure 19. Diagramme d'activité ajouter un article.....	49
Figure 20. Diagramme d'activité retourner des articles.	50
Figure 21. Diagramme d'activité authentification.....	51
Figure 22. Classe catégorie d'articles.....	52
Figure 23. Classe Société.	52
Figure 24. Classe Devises.	53
Figure 25. Classe Factures.....	53
Figure 26. Classe Lignes de facture.	54
Figure 27. Classe Listes de prix.	54
Figure 28. Classe Paiements.....	54
Figure 29. Classe Unités des articles.....	55
Figure 30. Classe Articles.	55
Figure 31. Classe Commandes.	56
Figure 32. Classe Lignes de commande.	56
Figure 33. Classe Réceptions.	57
Figure 34. Classe Lignes de réception.....	57
Figure 35. Classe Réapprovisionnements.....	57
Figure 36. Classe Planifications.	58
Figure 37. Classe Lignes de planification.	58
Figure 38. Classe Taxes.....	58
Figure 39. Classe entrepôts.	59
Figure 40. Classe fournisseurs.....	59
Figure 41. Classe Utilisateurs.....	60
Figure 42. Diagramme de classe global.	61
Figure 43. Commande installation Composer MacOS / Linux	76
Figure 44. Commande configuration PATH.	76
Figure 45. Commande installation Laravel avec Composer.....	76
Figure 46. Installation Laravel avec Laragon.....	77
Figure 47. Installation Laravel avec Laragon.....	77

Figure 48. Commande installation Git sur Debian.	77
Figure 49. Commande installation Git sur Fedora.	78
Figure 50. Commande configuration utilisateur Git.	78
Figure 51. Interface d'un repository vide dans GitHub.	78
Figure 52. Commande d'entrer dans le répertoire du projet.	79
Figure 53. Commande initialisation de Git.	79
Figure 54. Commande chargement du projet dans Git.	79
Figure 55. Commande Git pour créer le commit.	79
Figure 56. Commande Git pour envoyer le commit à l'adresse GitHub.	79
Figure 57. Répertoire Laravel.	80
Figure 58. Répertoire Laravel.	80
Figure 59. Architecture MVC.	81
Figure 60. Méthodes de routes.	82
Figure 61. Commande création contrôleurs.	82
Figure 62. Répertoire contrôleurs.	83
Figure 63. Fichier d'un contrôleur.	84
Figure 64. Commande création modèle et migration.	84
Figure 65. Répertoire modèles.	85
Figure 66. Répertoire migrations.	85
Figure 67. Fichier du modèle « Product ».	86
Figure 68. Fichier d'une migration.	87
Figure 69. Fichier de la migration du modèle « Product ».	87
Figure 70. Commande pour la migration.	88
Figure 71. Table « Products » dans la base de données.	88
Figure 72. Répertoire vue « products ».	89
Figure 73. Espace administrateur.	90
Figure 74. Espace financier.	91
Figure 75. Espace magasinier.	91
Figure 76. Répertoire unités de mesure.	92
Figure 77. Interface unités de mesure.	93
Figure 78. Interface ajouter une unité de mesure.	93
Figure 79. Interface afficher une unité de mesure.	94
Figure 80. Interface modifier une unité de mesure.	94
Figure 81. Interface afficher une unité de mesure.	95
Figure 82. Interface unités de mesure.	95
Figure 83. Interface unités de mesure.	96
Figure 84. Interface ajouter un produit.	96
Figure 85. Interface configurer un réapprovisionnement.	97
Figure 86. Interface ajouter un produit.	97
Figure 87. Interface afficher un devis.	98
Figure 88. Interface afficher un bon de commande.	99
Figure 89. Interface afficher une facture.	99
Figure 90. Interface afficher une réception.	100
Figure 91. Interface afficher un bon de retour.	100
Figure 92. Interface stock.	101

Introduction Générale

De nos jours, nous vivons dans un siècle où l'informatique prend une grande envergure dans n'importe quel domaine. C'est devenu le cœur de toutes les activités journalières de l'homme grâce aux services qu'elle nous offre. Quel que soit le domaine un système d'information intervient toujours pour faciliter la vie, donc c'est devenu un besoin afin d'effectuer des traitements automatiques.

Avant l'utilisation des ordinateurs, les informations sur les entreprises sont saisies manuellement sur des fiches, par la suite ces fiches sont archivées. Alors à chaque besoin le personnel doit chercher dans les archives ce qui n'est pas avantageux pour l'entreprise en terme de temps qui peut être élevé ainsi que la qualité du document qui se dégrade ce qui engendre la perte d'informations.

Avec l'apparition des nouvelles technologies, les entreprises les perçoivent comme étant un moyen pouvant augmenter leurs performances, notamment dans le volet de la gestion où il a connu une grande révolution, ce qui donne à l'entreprise un avantage de consulter sa situation en temps réel à l'aide d'un progiciel de gestion intégrée connu sous l'acronyme ERP.

Entreprise ressource planning (ERP) est un système d'information qui dispose de plusieurs applications (modules) relatives aux plusieurs fonctions de l'entreprise, ces modules sont indépendants entre eux et partagent la même base de données.

C'est dans ce cadre que s'inscrit notre projet fin d'études qui a pour objectif de mettre en place un module de gestion de stocks et d'approvisionnements. Pour cela nous avons organisé notre travail en quatre chapitres :

- Le premier chapitre intitulé « Solutions web pour les entreprises », est dédié à la description du concept de l'entreprise et les problématiques des entreprises et les solutions qu'offre un ERP à celles-ci.

- Le deuxième chapitre intitulé « Gestion de stocks et approvisionnements », est consacré à la description des généralités sur la gestion de stocks et d'approvisionnements.

- Le troisième chapitre intitulé « Analyse et conception », il présente les étapes de la conception et la modélisation de notre projet.

- Le quatrième chapitre intitulé « Réalisation », dans ce chapitre nous présenterons l'environnement et les outils utilisés pour le développement, à la fin nous allons présenter quelques interfaces et le résultat obtenu.

Chapitre 1

Solutions web pour les entreprises

Introduction

L'objectif dans cette partie est de définir l'entreprise, ces interactions avec son environnement, nous aborderons aussi les difficultés que les entreprises doivent faire face au cours de leur développement et les solutions qui s'y offrent.

Ensuite, nous allons détailler une de ces solutions qu'est l'ERP ainsi que son impact sur la performance et les avantages concurrentiel qu'elle offre pour les entreprises qui l'intègrent à leurs systèmes.

I. Concept d'entreprise

I.1. Définition de l'entreprise [1]

L'entreprise de point de vue économique est une entité de productions de produit consommable ou de biens, du point de vue sociologique l'entreprise est composée d'investisseurs, de dirigeants et de salariés.

Chacun avec des objectifs différents, les investisseurs auront tendance à prioriser la finance pour rentabiliser leur placement et ainsi garantir un gain, et avoir le placement le plus rentable possible.

Les dirigeants priorisent la performance et la croissance de l'entreprise, pour garantir leur place et accéder à plus de pouvoir.

Pour les employés leur but est d'avant tout garantir leur emploi et l'épanouissement grâce à leur savoir-faire.

Du point de vue systémique l'entreprise est un système comme les autres avec une ouverture sur son environnement, environnement qui l'oblige à s'adapter pour survivre et prospérer et s'épanouir, l'entreprise en tant que système est un ensemble composé de sous-systèmes et de services qui interagissent entre eux dans un but.

Ce but varie selon l'entreprise ou selon les points de vue des différents services, mais œuvrent uniquement dans l'intérêt de l'entreprise.

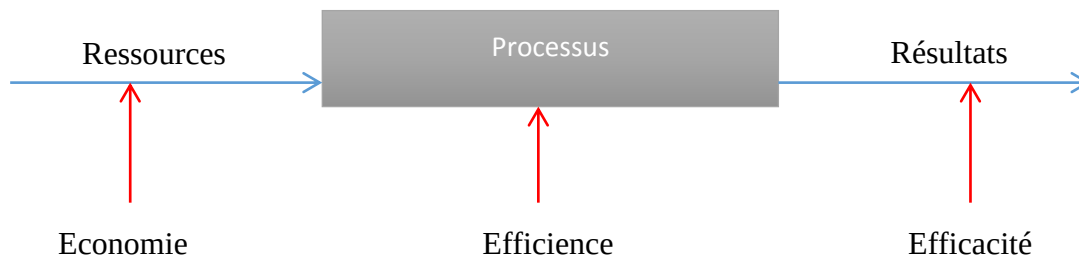
I.2. Environnement économique [2]

L'environnement économique peut être simplement décrit par tous les facteurs qui peuvent affecter l'entreprise et avoir un impact sur son économie.

L'économie d'une entreprise est directement liée à sa performance sur le marché, plus une entreprise est performante plus son économie s'épanouit, et l'inverse est vrai.

Depuis les années 90 les chercheurs se sont attelés à définir la performance (Bouquin 1986, Bourguignon 1995, Bessire 1999...) et à la quantifier par sa dimension financière qui est la rentabilité pour les investisseurs et le chiffre d'affaires de l'entreprise sur le marché.

Selon Bouquin (2004 p63) la performance se joue sur ces trois points.



L'économie est toutes les économies faites lors de la procuration des ressources ainsi une entreprise qui vise à accroître sa performance devra se procurer des ressources au plus faible coût possible.

L'efficience est le fait de créer la plus grande quantité de produit possible avec une quantité de ressources donnée, éventuellement pour éviter le gâchis des ressources qui entraînerait des pertes évitables.

L'efficacité est la réalisation des objectifs attendus.

Une entreprise qui a pour but d'augmenter ces performances devra impérativement optimiser l'une de ces variables, en effet l'économie et l'efficacité peuvent être optimisées par une gestion adéquate, par contre l'efficience dépend de plusieurs facteurs tels que la gestion des ressources humaines, financières et matérielles, l'interaction et la coordination de différents services de l'entreprise ainsi que l'interactions avec les différents acteurs (clients, fournisseurs, sous-traitant..) qui rendent l'optimisation de l'efficience complexe.

L'une des solutions mise en place par les entreprises est de créer des systèmes d'information constitués de systèmes séparés qui communiquent à travers des interfaces afin de coordonner les services (comptabilité, achat, commerce...) mais dans un environnement concurrentiel ou l'obligation de dégager une certaine rentabilité est élevée, ont obligé les entreprises à se tourner vers des solutions axées sur les nouvelles technologies depuis les années 90.

I.3. L'évolution des Tic dans les entreprises [3]

Les technologies de l'information et de la communication abrégée Tic, sont l'ensemble des technologies qui sont utilisés pour stocker et traiter l'information sous forme numérique.

Les premières études sur les effets des Tic sur la performance et la productivité avaient fait leur apparition en 1986 avec Robert Solow qui nota en premier le paradoxe qu'en pratique les Tic n'avait pas d'effet positif sur la productivité contrairement à la théorie qui affirme le contraire.

Une autre analyse menée en 2002 sur l'utilisation des Tic en entreprise, cette fois si à partir de données récentes indique une corrélation forte entre la performance, et la sophistication des équipements.

Ainsi avec l'évolution fulgurante des technologies grâce à la convergence de l'informatique, de la télécommunication et de l'internet. Les Tic ont pris une nouvelle dimension et avec eux la montée en puissance des logiciels de gestion d'entreprises tels que les progiciels de gestion intégrée abrégée PGI ou en anglais ERP.

II. ERP

II.1. Historique [4]

La création de l'ERP revient principalement à Joseph Orlicky qui créa dans les années 1960 le MRP abréviation de Material Requirements Planning ancêtre de l'ERP, le MRP répond essentiellement aux besoins de planification des entreprises.

La notion d'ERP tel que nous la connaissons a fait son apparition dans les années 90, mais n'a connu son essor que dans les années 2000 avec l'arrivée de l'internet, l'utilisation de l'ERP se généralise et évolue jusqu'à arriver à l'ERP tel que nous le connaissons aujourd'hui.

II.2. Définition [5]

Le sigle ERP veut dire Entreprise Resource Planning son semblable en Français est Progiciel de Gestion Intégré abrégé PGI.

Contrairement au MRP qui se contente de la planification des besoins, l'ERP est un logiciel qui permet la gestion de l'ensemble des sous-systèmes d'une entreprise ainsi que la coordination de ceux-ci.

Pour y parvenir l'ERP intègre l'ensemble des fonctions utiles d'une entreprise sous forme de modules qui partagent une base de données unique, ceci permet l'échange d'informations entre les modules, dans ce cas-là on parle de moteur de Workflow.

II.3. Avantages liés à l'intégration d'un ERP [6]

Les bénéfices liés à l'implémentation d'un ERP ont été prouvés par bon nombre de recherches, l'une d'elles menée par le groupe Aberdeen qui ont quantifié et publié les résultats suivants :

- Réduction des coûts d'opérations de 22%
- Réduction des coûts d'administration de 20%
- Réduction d'inventaires de 17%
- Amélioration du temps de livraison de 19%
- Amélioration du respect des délais et des budgets de 17%

Même les entreprises en difficulté ont réalisé des bénéfices grâce à l'intégration de leur ERP, leurs résultats s'élèvent à :

- Réduction des coûts d'opérations de 7%
- Réduction des coûts d'administration de 4%
- Réduction d'inventaires de 9%
- Amélioration du temps de livraison de 11%
- Amélioration du respect des délais et des budgets de 6%

Comme l'étude le souligne le gain en pourcentage ne paraît pas impressionnant, mais pour chaque million de dollars déboursé dans les coûts d'opérations, 70 000 \$ sont économisés.

En effet nous pouvons constater les gains en productivité et en maturité des entreprises, pour y parvenir l'ERP procède à une amélioration sur plusieurs aspects :

II.3.1. Aspect administratif [7]

En fusionnant tous les systèmes de l'entreprise en une seule application, l'installation de l'ERP conduit à une réduction des coûts d'exploitation et de maintenance, et comme l'ERP possède une architecture sous forme modulaire il fournit une infrastructure qui assure une flexibilité en cas de changements futurs, donc offre la possibilité d'implémenter de nouvelles fonctionnalités.

Une seule application donc une seule base de données, cette base de données unique permet un gain de temps. Réduire le volume d'information inutile et d'éviter les saisies multiples, donc l'installation de l'ERP permet la résolution des problèmes d'incohérence des informations et rend les données enregistrées plus fiables.

De plus cela évite les activités manuelles de traitement, de comparaison et de recherche réalisées par les employés dans le cadre de l'interfaçage des différents services. Ce qui conduit à un gain de temps et une croissance de la productivité administrative.

II.3.2. Aspect opérationnel [7]

L'utilisation d'un ERP conduit à la suppression des risques opérationnels et aux risques de pertes liés à des erreurs humaines où des dysfonctionnements dans le contrôle interne, fraudes qui peuvent résulter d'un dysfonctionnement des systèmes d'information déjà en place, l'ERP permet la pertinence des informations partagées et évite ces dysfonctionnements qui peuvent être plus ou moins graves et qui peuvent entraîner des coûts supplémentaires inutiles.

L'ERP permet aussi un suivi au niveau des achats jusqu'aux ventes. En effet, dès la création de la commande, des données telles que le calcul des marges et des crédits est généré automatiquement de façon dynamique réalisant ainsi une intégration financière, grâce à cette fonctionnalité, l'ERP aide les dirigeants dans la planification et la prise de décision, et leur permet d'améliorer la gestion des ressources, et ainsi améliorer les décisions opérationnelles.

De plus la centralisation est très bénéfique pour les services de finance. Car ce logiciel permet également la centralisation des tâches qui permet à une amélioration de la productivité en

réduisant le nombre du personnel qui travaille sur la même tâche, cela permet des économies d'échelles notamment en matière de facturation.

II.4. Inconvénients

L'ERP offre des avantages non négligeables, mais une telle solution doit forcément comporter quelques désavantages.

Les projets ERP conduisent généralement à des coûts lors de la mise en place et de la maintenance. De plus la complexité des programmes utilisés requiert l'utilisation et l'entretien de serveurs puissants. Ce qui implique que le coût est souvent dépassé comme le montre l'étude CXP 2017. [8]

Budget – dépassement ?

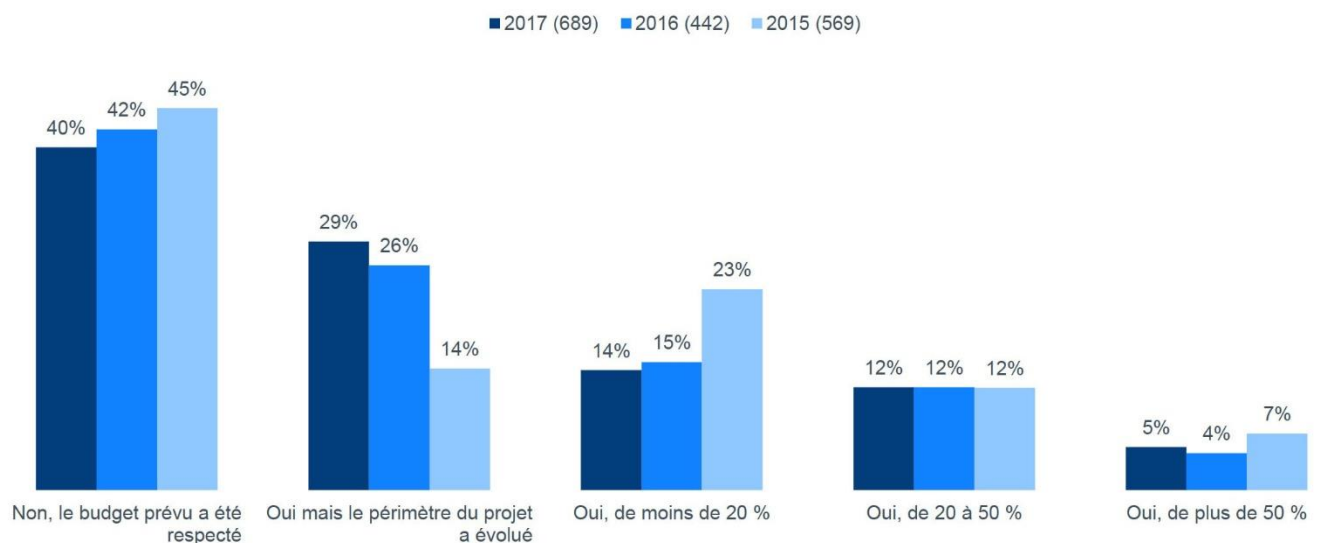


Figure 1. Taux du dépassement de budget lors de l'implémentation d'un ERP.

On peut constater qu'en 2017 plus de 60% des entreprises qui ont implémenté un ERP ont dépassé le budget prévu, 58% en 2016 et 55% en 2015.

En outre du coût, un projet de cette envergure requiert un temps et des ressources qui peuvent dépasser les prévisions, tel que le montre l'étude ERP Report 2010 du cabinet de conseil Panorama Consulting. [9]

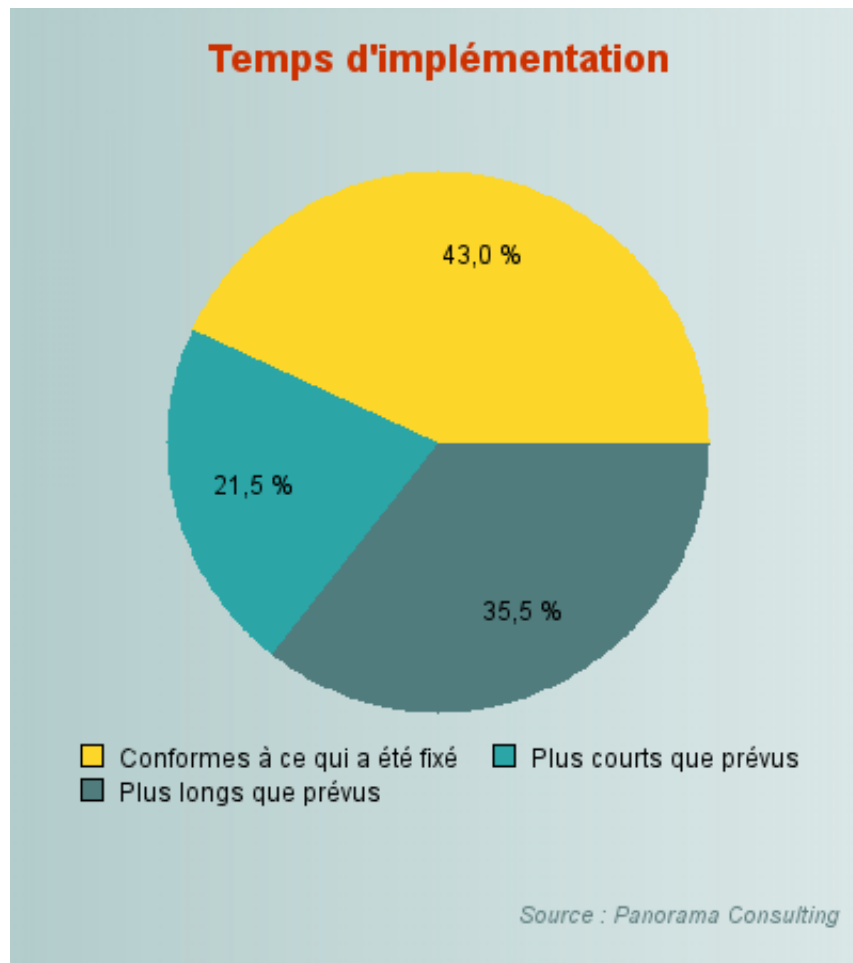


Figure 2. Taux des dépassements des délais lors de l'implémentation d'un ERP.

Cette étude montre que plus de 35.5% des entreprises ayant mis en place un ERP, ont vu le temps prévu pour l'implémentation dépassée, il est aussi à noter que la durée moyenne de la mise en place d'un ERP est de 18,4 mois qui varient d'un éditeur à un autre.

II.5. Fonctionnalités

L'ERP gère et organise les informations de l'ensemble des services de l'entreprise de façon automatique et dynamique, de l'achat des ressources à la vente en passant par la production, ces fonctionnalités sont nombreuses. Les modules plus couramment utilisés sont les suivants :

- Gestion d'achats
- Gestion de la chaîne logistique
- Gestion de stock et d'inventaire
- Gestion de production
- Gestion de projet
- Gestion des ressources humaines
- Gestion comptabilité
- Gestion commerciale

- CRM : Gestion des relations clients

Chaque module couvre des fonctionnalités qui lui sont propres, dans le tableau ci-dessous une présentation de certains modules et des fonctionnalités qu'ils proposent.

Modules	Fonctionnalités
Achats	Gestion de toutes les transactions comptables, telle que les bons de commande pour l'approvisionnement. Etc.
Chaine logistique	Gestion des ressources utilisées pour le pilotage de la chaine d'approvisionnement et de livraison.
Stock	Gestion des mouvements du stock, état du stock, entreposage.
Production	La gestion de la production, permet de réguler l'offre et les besoins en ressources par apport à la demande, impliquent la planification des ordres de fabrication et le contrôle de qualité.
Gestion de projet	Gestion de l'ensemble des projets de l'entreprise, de ces tâches et de ces plannings.
Ressources humaines	Gestion des ressources humaines et l'organisation de la rémunération des employés ainsi que des plannings de travail de ceux-ci.
Comptabilité	Gestion des obligations comptable auxquelles l'entreprise est soumise et suivie en temps réel de la santé financière de celle-ci, ainsi que de la gestion de facturation et des multidevises.
Commerciale	Gestion de l'aspect commerciale de l'entreprise, permet la gestion de l'ensemble des commandes clients et de leur facturation, permet aussi la réalisation de devis rapide et précise.
CRM	Gestion des relations clients, permet de réaliser de meilleurs suivis de l'environnement : clients, fournisseurs, prospects. etc.

Conclusion

Après avoir défini lors de la première partie le concept d'entreprise, l'environnement et les contraintes auxquelles elle fait face. L'obligation de dégager un bénéfice est vitale, mais un certain nombre de points complexifient leurs systèmes d'information et freine la croissance économique de celle-ci, ces mêmes points qui rendent le recours à une technologie de l'information telle que l'ERP presque obligatoire.

Dans le deuxième point nous avons étudié l'ERP qui est au cœur du système d'information et qui permet la gestion de tous les services de l'entreprise. Ainsi que les avantages et les inconvénients à recourir à un tel outil.

Ensuite nous allons approfondir la recherche et étudier lors du deuxième chapitre, la gestion de stocks et d'approvisionnement.

CHAPITRE II

Gestion de stocks et d'approvisionnements

Introduction

Dans ce chapitre, nous allons citer les notions de base de la gestion des stocks. Tout d'abord, nous allons définir les principaux concepts de ce domaine, après nous expliquerons les différentes typologies des stocks et ses utilités, puis, nous allons définir l'objectif de la gestion des stocks et sa procédure, et enfin, nous parlerons sur la notion inventaire et ses types.

Ensuite, nous allons définir la notion gestion d'approvisionnement, ses procédures et ses méthodes. Enfin comme conclusion nous allons donner une petite définition de ce que c'est qu'un logiciel de gestion de stock et quelques intérêts de les utiliser.

I. Gestion de stock

I.1. La gestion [10]

La gestion désigne l'action ou la manière de gérer, d'administrer et d'organiser quelque chose. Le terme est utilisé dans de nombreux domaines comme celui de l'entreprise, de l'administration, de l'immobilier, etc. La gestion se caractérise dans ces situations par le fait de confier à autrui, ou à soi-même, des affaires à gérer.

I.2. Le stock [11]

Désigne l'ensemble de produits stockables appartenant à une entreprise à un moment donné et qui sont dans l'attente de leur utilisation ou bien de leur mise en vente.

Le stock est constitué de matières premières, matières consommables, composants achetés ou fabriqués, sous-ensembles achetés ou fabriqués, articles sous-traités, produits finis, articles de rechange, articles défectueux à retoucher, articles obsolètes, emballages, etc.

Pour chaque entreprise (commerciale ou industrielle), le stock est indispensable afin de bien gérer ses articles dans le but de satisfaire ses clients et assurer ses besoins futurs. Cela implique de connaître gérer son stock afin d'éviter toute rupture et être à la fois optimal et efficace dans la réponse à tout besoin futur sans avoir à supporter les déférents coûts excessifs du stock (notamment dans le cas des dossiers d'importations où les frais d'importations peuvent être énormes...).

I.3. Typologie de stock [12]

I.3.1. Types de stocks selon leur fonction

I.3.1.1. Stock de sécurité

Le stock de sécurité est le niveau de stock qui permet aux entreprises de limiter les ruptures de stock dues aux prévisions non conformes à la demande, délai d'approvisionnement plus long que prévu, etc.

I.3.1.2. Stock d'alerte

Comme indiquer une alerte de stock veut dire que c'est le moment de déclencher la commande de réapprovisionnement pour éviter une rupture de stock.

I.3.1.3. Stock saisonnier

Ce stock permet d'anticiper les périodes de l'année où y'a une forte demande saisonnière, par exemple en été on a une exceptionnelle demande sur les eaux minérales.

I.3.1.4. Stock dormant (ou mort)

Cette catégorie constitue les produits qui ont cessé d'être demandé pour de nombreuses raisons, et que l'entreprise n'a pas pu le temps d'arrêter les commandes. Il fait office du poids mort dans l'entrepôt.

I.3.1.5. Stock en transit

Il s'agit de toutes les marchandises encore présentes dans les processus de production ou de commercialisation : en cours d'acheminement, en cours d'emballage, en cours de fabrication, etc.

I.3.1.6. Stock cycle

Le stock cycle est la quantité de stock nécessaire pour répondre à la demande des clients à partir du moment où on déclenche les nouvelles demandes de réapprovisionnement jusqu'à sa réception.

I.3.1.7. Stock d'anticipation - spéculation

Parfois les entreprises doivent acheter et tenir en stock plus de produits que nécessaire c'est ce qu'on appelle le stock d'anticipation (ou spéculation).

Ce stock nous permet de se protéger de toutes variations et augmentation des prix, une forte demande saisonnière et de se mettre à l'abri des éventuelles grèves.

I.3.2. Types de stocks selon l'organisation opérationnelle**I.3.2.1. Stock optimal**

Il représente le juste équilibre que toute entreprise veut atteindre afin de maximiser sa rentabilité et de minimiser les coûts du stock.

Il a pour but de permettre à toute entreprise de disposer de la quantité exacte de produits et d'éviter les ruptures de stock ou le sur stockage.

I.3.2.2. Stock physique

Comme indiqué, ça représente les quantités physiques des produits stockées de l'entreprise.

I.3.2.3. Stock net

Il s'agit du stock physique moins les commandes clients à traiter c'est-à-dire les commandes confirmées et qui ne sont pas encore livrées.

I.3.2.4. Stock disponible

Ça représente la somme du stock net et des commandes passées auprès de vos fournisseurs qui ne sont pas encore reçus.

I.3.2.5. Stock minimum

C'est la quantité minimale que toute entreprise doit avoir dans son entrepôt, une fois cette quantité est atteinte on doit se réapprovisionner.

I.3.3. Types de stocks selon la date de péremption**I.3.3.1. Stock périssable**

Ce sont tous les produits et marchandises qui se détériorent avec le temps.

Par exemple : les desserts, les viandes rouges, les légumes, etc...

I.3.3.2. Stock non périssable

C'est les produits dont le temps n'a pas d'impact sur eux

Par exemple : l'électroménager, les rasoirs, les ciseaux, etc...

I.3.3.3. Stock avec date de péremption

Ces produits ne pourront plus être vendus une fois la date limite consommation (DLC) est dépassée.

Par exemple : les produits laitiers, les boissons, les chips, etc...

I.4. L'utilité des stocks [13]

Comme on a dit précédemment chaque entreprise (commerciale ou industrielle) doit avoir un stock malgré que ce soit coûteux et que ça représente une grande somme d'argent. Les objectifs sont les suivants :

Objectifs	Exemples
Commercial	• Diminuer les délais de livraison à la clientèle.

Régulation	• Se protéger contre l'arrêt de production. • Répondre à une demande saisonnière.
Technique	• Améliorer la qualité des produits (cas du bois et des boissons alcoolisées)
Sécurité	• Se prémunir contre l'instabilité de l'environnement
Financier	• Protection des hausses des prix et de ses variations.

I.5. Gestion de stocks

I.5.1. Définition [14]

La gestion des stocks regroupe les activités d'achat et de stockage et se propose d'établir les références à tenir en magasin, de déterminer les quantités associées à ces références et de fixer les modes et les échéances de réapprovisionnement en respectant les conditions de coût, de délai et de qualité.

I.5.2. Procédures gestion de stocks [15]

L'un des principaux documents de la gestion de stocks c'est la fiche de stock. Chaque article en possède une qui enregistre tous ses mouvements de stocks.

Ces fiches de stock sont tenues à jour avec les bons d'entrée et les bons de sortie,

- Bon de sortie : à chaque sortie de marchandise du magasin ce document est établi afin de constater et tracer tous les mouvements de stocks.
- Bon d'entrée : après vérification des quantités et références sur le bon de livraison, le magasinier établit un bon d'entrée en magasin. Ce document précise les références, les quantités des articles reçus et les observations éventuelles. C'est un document interne à l'organisation et assure un bon suivi de gestion de stock.
- Fiche de Stock : c'est un document qui vaut de l'or pour le gestionnaire de stock, il lui permet de tracer tous les mouvements de stock d'un article (toutes les entrées et toutes les sorties).

I.5.4. Méthodes de valorisation de stock [16] [17]

I.5.4.1. Le coût moyen unitaire pondéré

Le CUMP ou (CMUP) c'est une technique comptable pour valoriser le stock, elle consiste à calculer un coût moyen après chaque entrée en stock (achat), il est obtenu de cette manière :

$$\text{CUMP} = \frac{(\text{Valeur du stock à l'ancien CUMP avant réapprovisionnement} + \text{Coût du nouveau Approvisionnement})}{(\text{Stock physique avant approvisionnement} + \text{quantité à approvisionner})}$$

I.5.4.2. La méthode FIFO

Aussi connu sous le nom méthode de l'épuisement des lots ce qui signifie le premier entré est le premier sorti (First In First Out).

Généralement elle est utilisée avec les produits périssables qui ont une date limite de consommation (DLC).

I.5.4.3. La méthode LIFO

Il s'agit de la méthode du dernier entré premier sorti (Last In First Out), c'est-à-dire les derniers achetés sont les premiers à sortir, avec cette méthode on indique leur valeur comme étant le prix des derniers entrants.

Elle permet de tenir compte des variations des prix donc elle est recommandée dans les périodes d'inflation afin de se mettre à l'abri de ces variations des prix.

I.5.4.4. La méthode NIFO

Prochain entré premier sorti (Next In First Out), son principe est de retenir une valeur de remplacement qui est un coût théorique au moment de la sortie afin d'obtenir le coût de production ou le coût de revient, il peut être déterminé en se basant sur les prix de la dernière facture ou la prochaine, une estimation du coût de la prochaine commande ou le coût actuel sur le marché.

Cette méthode permet de connaître le coût de reconstitution et valorisation de stock rapidement et éviter un grand nombre de calcul. Mais elle n'est pas reconnue, ni autorisée par les normes **IAS/ISRF**. [13]

I.5.4.5. L'utilisation des prix par lots

Le principe de cette méthode est de séparer et stocker chaque lot avec son prix d'acquisition en mémorisant ce dernier.

Ensuite, chaque sortie le choix va se faire sur un lot bien identifié au choix du magasinier (gestion interne de l'entreprise) avec le prix correspondant à ce lot.

I.5.5. Méthode de classement des stocks

Souvent les entreprises possèdent un grand nombre d'articles à gérer. Pour cela le gestionnaire de stock doit appliquer des méthodes de gestion de stock en tenant compte de l'importance des articles.

I.5.5.1. La méthode ABC

I.5.5.1.1. Principes et intérêt de la méthode [18]

La méthode ABC (Activity Based Costing) a pour principe découper les stocks en trois classes (A, B et C) en fonction de leur valeur. Après découpage on aura :

- Classe A** : les 10% des articles qui représentent environ 75% de la valeur totale du stock, ça représente les produits qui font vivre l'entreprise.

- Classe B** : les 25% des articles suivants qui représentent environ 20% de la valeur totale du stock, ça représente les produits en phase de mutation ou en phase de déclin.

- Classe C** : les 65% des articles restants qui représentent environ 5% de la valeur totale du stock, ça représente les produits en déclin ou en phase de lancement.

Ces pourcentages diffèrent légèrement selon les entreprises.

I.5.5.1.2. Les étapes de la méthode [18]

1-calcul de la consommation annuelle par article (en valeur).

2-classements des articles dans l'ordre des valeurs décroissantes.

3-calculs du pourcentage par rapport au total, et du pourcentage cumulé.

4-définitions des classes A, B, C.

5-représentations graphiques (éventuellement).

I.5.5.1.3. Limites et précautions d'emploi du classement ABC [19]

L'utilisation du classement ABC est recommandée lorsque le nombre d'articles est suffisamment élevé pour espérer avoir des résultats concrets.

Cependant, le classement ABC doit être utilisé avec précaution et il faut avoir une homogénéité réelle des articles classés.

I.5.5.2. La méthode 20/80 [20]

Le principe de cette méthode est de rechercher les 20% des produits qui réalisent 80% du chiffre d'affaires total de l'entreprise et les mettre sous un suivi rigoureux, quant aux restes des articles de moindre valeur seront gérés avec un peu plus de souplesse.

I.5.6. Les enjeux de la gestion des Stocks [21]

Comme on a dit précédemment le stock est indispensable dans une entreprise, il permet de satisfaire les besoins à venir et d'une part honorer les commandes clients au bon moment d'une manière assurant une bonne utilisation de stock tout en évitant les ruptures. Pour cette raison les responsables de la gestion de stock interviennent avec l'art de savoir gérer afin de rechercher cet équilibre juste entre le niveau de satisfactions des clients ainsi que de minimiser les coûts causés par la constitution de stock.

Alors pour éviter toute perte à l'entreprise on distingue deux points majeurs qui sont :

I.5.6.1. La réduction des coûts :

L'un des premiers objectifs de la gestion de stock est de chercher à optimiser et réduire les coûts de stock qui peuvent s'avérer chères et élevés.

I.5.6.2. L'élimination des ruptures de stock :

L'erreur à ne pas faire dans une entreprise c'est d'attendre à ce que son stock soit nul pour passer sa commande auprès du fournisseur car cela peut engendrer un risque de rupture de stock donc ça va mener automatiquement à rater une ou plusieurs ventes. La solution est d'avoir un stock d'alerte qui va déclencher automatiquement la commande d'approvisionnement.

Ce stock d'alerte est calculé de la manière suivante :

Stock d'alerte = stock de sécurité plus le stock consommé avant la prochaine réception.

Où le stock de sécurité est la quantité qui ne doit pas être utilisée sauf en cas de problème de livraison.

I.6. Les inventaires [22]

L'inventaire c'est une liste qui représente l'état et l'évaluation des biens de l'entreprise, il répond à un besoin du chef d'entreprise qui souhaite connaître ces valeurs.

Son but est de faire une comparaison entre le stock théorique (celui qui se trouve sur le logiciel) et le stock physique (magasins et entrepôts) afin de traiter les différences de quantités ou repérer les erreurs d'adressage et les stocks dormants.

On distingue trois types d'inventaires :

I.6.1. L'inventaire permanent

Il est aussi connu sous le nom « inventaire informatique », son principe est de comptabiliser tous les mouvements de stock (entrées et sorties) et ainsi que de connaître le stock actuel, cette méthode d'inventaire est adaptée particulièrement pour les stocks dont les quantités par référence sont petites et les unités stockées sont faciles à compter.

- Lors d'une entrée d'une marchandise : saisie des quantités entrées et saisie de stock final.
- Lors d'une sortie d'une marchandise : saisie des quantités sorties et saisie de stock restant.

Cet inventaire ne provient pas d'un comptage physique mais d'un calcul :

$$\text{ENTREES} - \text{SORTIES} + \text{STOCK INITIAL} = \text{STOCK FINAL}$$

I.6.2. L'inventaire annuel

Il est établi à la fin de chaque exercice comptable, il s'agit d'un comptage de toutes les quantités en stock. L'inventaire annuel est réclamé par les contrôleurs financiers.

Cet inventaire permet Il permet de vérifier la fiabilité de l'inventaire permanent et d'en corriger les erreurs.

I.6.3. L'inventaire tournant

Il s'agit d'un comptage périodique et planifié des stocks durant toute l'année. Cet inventaire nous donne la liste des articles avec ses quantités disponibles dans une période bien définie plusieurs fois tout au long de l'année.

Cette méthode d'inventaire permet également de :

- Alléger l'un inventaire annuel.
- Éviter la rupture des stocks critiques.
- Corriger les erreurs rapidement.

II. Gestion d'approvisionnement [15]

La politique d'approvisionnement est un ensemble d'actions prises par l'entreprise en vue d'acquérir des biens ou de services.

La mission de la fonction « approvisionnement » est de répondre aux besoins de l'entreprise dans les meilleures conditions de qualité, de coûts, de délais et de sécurité en guise de lui bien permettre de continuer son activité.

II.1. Objectifs [15]

Les principaux objectifs de cette fonction sont :

- Assurer la sécurité et la régularité des approvisionnements.
- Améliorer le rapport qualité/prix des produits achetés.
- Limiter et optimiser le niveau des stocks de matières premières et des fournitures.

II.2. Tâches [15]

- Surveiller l'évolution de ses achats et connaître le marché ainsi que les sources d'approvisionnement.
- Collaborer à la définition des caractéristiques et spécifications des produits à acheter.
- Rechercher des fournisseurs et entamer des négociations avec eux.
- Passer des commandes d'achats.
- Suivre les réceptions des commandes.
- Contrôler les réceptions des commandes.
- Analyser l'utilisation des achats dans les différents services.
- Gérer les stocks.
- Fournir les informations dans ce domaine à tous les services utilisateurs.

II.3. Le processus traditionnel d'approvisionnement [21]

II.3.1. Détermination des besoins

La détermination des besoins est une étape essentielle pour établir son programme d'approvisionnement. Cette étape consiste à noter le manque des matières, marchandises, articles.

II.3.2. Choix des fournisseurs

Une fois que les besoins sont déterminés, l'entreprise passe à la prise de décision d'achat en consultant plusieurs fournisseurs afin de choisir le fournisseur le plus avantageux.

Les fournisseurs sont généralement sélectionnés sur les critères suivants :

- Le prix.
- La qualité.

- Le délai de livraison .
- Le mode et les conditions de paiement.
- La réputation.

II.3.3. La commande

Une opération d'achat sur le plan administratif entraîne l'exécution d'un certain nombre de travaux et la création d'un certain nombre de documents :

II.3.3.1. La demande d'achat

C'est document émis par le magasin, il comporte les informations suivantes :

- Numéro de commande ;
- Nom du service utilisateur.
- Référence des articles.
- Signature du demandeur.

Ce document est établi en plusieurs exemplaires :

- 1 doit rester au magasin.
- 3 sont adressés au service des achats.

II.3.3.2. La commande

Elle est rédigée par les services achats aux fournisseurs. Ce document est établi en 5 exemplaires :

- 2 adressés au fournisseur (1 reviendra au service achat signé par le fournisseur).
- 1 adressé au service comptabilité ;
- 2 sont adressés au magasin.

II.3.3.3. La réception et le contrôle

La commande doit être suivie par :

- Les libellés de livraison sont surveillés.
- La réception est organisée et planifiée et contrôlée dans le temps et l'espace.
- Les marchandises, produites ou matières sont contrôlées en quantité et qualité.

Après le contrôle, ils seront stockés au magasin.

II.4. Les Méthodes de réapprovisionnements

II.4.1. Approvisionnement à Date et Quantité fixes [23]

Aussi connue sous le nom de « méthode calendaire », cette méthode s'utilise dans le cadre d'un contrat avec le fournisseur afin de recevoir des quantités fixes de marchandises à une précise et fixe.

Cette méthode est adaptée pour les entreprises qui sont stables et ont une consommation régulière.

Avantages :

- Gestion des stocks simple.
- Une bonne organisation entre l'entreprise et le fournisseur.

Inconvénients :

- Une erreur d'évaluation de stock ou bien une consommation non régulière implique un risque de cumul ou une rupture de stock. Ce qui peut être très coûteux dans le cas où il y a les frais d'importation.

II.4.2. Approvisionnement à Date fixe et Quantité variable [23]

Aussi connues sous le nom de « méthode de reapprovisionnement », les réceptions se font sur des dates prédéterminées avec des quantités variables en fonction du stock afin d'avoir un stock niveau suffisant.

Avantages :

- Simplification de la gestion et maîtrise des immobilisations financières.

Inconvénients :

- Une consommation irrégulière peut entraîner un risque de cumul de stock ou de rupture de stock.

II.4.3. Approvisionnement à Date variable et Quantité fixe [23]

Aussi connue sous le nom de « méthode du point de commande », cette méthode a pour principe de définir un stock minimum en premier lieu qui permet de déclencher des commandes avec des quantités fixes et de couvrir les besoins avant la prochaine réception.

Cette méthode est recommandée pour les articles très coûteux et dont les consommations sont irrégulières.

Avantages :

- Cette méthode permet de faire une meilleure optimisation des approvisionnements.
- Des calculs bien faits évitent le cumul de stock.

Inconvénients :

- Une croissance subite et irrégulière peut entraîner une rupture de stock consommation. Cela impose quelquefois la mise en place d'un stock de sécurité. Ce qui finalement ne résout le problème d'immobilisation financière que dans une moindre mesure.

II.4.4. Approvisionnement à Date et Quantité variables [23]

L'entreprise effectue ses commandes selon ses besoins en estimant les quantités dont elle en a besoin.

Avantages :

- Optimisation des coûts inutiles.

Inconvénients :

- Les fournisseurs ne sont pas très ouverts à cette méthode car ça ne leur permet pas d'avoir une bonne visibilité.
- En cas de commande non prévue l'entreprise peut se retrouver en difficulté de réapprovisionnement si son fournisseur n'est pas dans la mesure de répondre à ses besoins.

II.4.5. Le drop shipping [24]

Lorsque l'entreprise reçoit une commande client, elle récupère le règlement de la facture chez son client. Ensuite elle passe la commande au fournisseur qui va se charger de la préparation et l'expédition du colis.

Avantages :

- Se libérer de la gestion de stock.

Inconvénients :

- C'est une méthode de gestion des stocks plus adaptée pour du E-commerce.
- Elle n'est pas recommandée pour le commerce physique.

III. Logiciels de gestion de stock

III.1. Définition

C'est un système informatique qui assure une bonne maîtrise des stocks de l'entreprise en temps réel, il est indispensable pour limiter les coûts liés aux stocks et satisfaire les demandes clients.

III.2. Intérêt des logiciels de gestion de stock

Les logiciels de gestion de stock sont utilisés pour de nombreuses raisons. Nous pouvons citer :

- Anticiper les commandes.
- Éviter les ruptures de stock et les surstocks.
- Mieux maîtriser les coûts liés à la gestion des stocks.
- Suivre l'évolution des stocks plus rapidement et plus facilement.

- Prévoir des opérations promotionnelles en fonction des stocks disponibles.
- Maîtriser les délais d'approvisionnement.
- Réduction des falsifications humaines.
- Abaissement des pertes de données.
- Limiter les erreurs.

Conclusion

Dans ce chapitre nous avons vu les notions de base sur la gestion de stocks et d'approvisionnements, ce qui nous a permis de bien comprendre la logique métier.

Dans notre prochaine étape, nous allons nous focaliser sur le stock d'alerte et le stock optimal comme typologie de stocks, le coût moyen pondéré pour la valorisation de stock pour définir les besoins de notre projet.

CHAPITRE III

Analyse et conception

Introduction

Dans ce chapitre notre travail consiste à mettre en place la conception et l'analyse des fonctionnalités de notre application.

Pour cela nous allons utiliser une démarche générale, dans cette démarche nous allons illustrer des diagrammes en utilisant le langage UML. Les diagrammes utilisés sont les suivants, diagramme de contexte, diagramme de séquence, diagramme de cas d'utilisations et diagramme de classe globale modélisé grâce à l'outil Astah.

I. Présentation d'UML

UML est un acronyme qui désigne Unified Modeling Language qui veut dire langage de modélisation unifié en français. Les origines de UML remontent aux années 90, à cette époque il y avait plusieurs moyens de représenter et de documenter un logiciel. En conséquence, il se dégage un besoin d'unifier ces systèmes de représentations. Et donc, en 1994-1996 UML a été développé par des ingénieurs qui travaillaient à Rational Software une société de développement logiciel en Amérique. Ce nouveau concept a connu un large succès et a été adopté comme un standard en 1997 et qui maintenant est l'une des techniques de modélisation les plus populaires au monde.

Comme le célèbre proverbe de Confucius qui dit : « une image vaut mille mots ». UML tend à utiliser cette logique en représentant les composants des logiciels sous forme de représentation visuelle.

En général UML était utilisé comme un langage de modélisation à usage général dans le domaine de l'ingénierie des logiciels. Toutefois il a été prouvé son efficacité dans la documentation des processus des entreprises et des Workflows. Par exemple, le diagramme d'activité est un type de diagramme d'UML qui peut être utilisé comme remplace du Flowchart. Remplacement rentable car UML est un meilleur moyen de standardisation et modélisation du Workflow, de plus, il a une plus grande gamme de fonctionnalités qui améliorent la lisibilité et l'efficacité.

II. Analyse et conception

La phase d'analyse débute par la spécification des besoins fonctionnels du système, ensuite définir les acteurs ainsi que leurs différentes tâches. Ceci nous mène vers un résultat qui est un diagramme de cas d'utilisation traduisant la dynamique système qui sera utilisé dans la phase de conception.

II.1. Spécification des besoins

II.1.1. Besoins fonctionnels

C'est une description des caractéristiques du système et de ses fonctionnalités. En partant de ce principe notre application doit permettre :

Aux utilisateurs :

- Gestion des articles :
 - Ajouter un article.
 - Consulter un article.
 - Modifier un article.
 - Supprimer un article.
 - Consulter le stock.
 - Ajouter une catégorie d'article.
 - Modifier une catégorie d'article.
 - Consulter une catégorie d'article.
 - Supprimer une catégorie d'article.
 - Ajouter une unité de produit.
 - Modifier une unité de produit.
 - Consulter une unité de produit.
 - Supprimer une unité de produit.
- Gestion des factures
 - Consulter une facture.
 - Ajouter un règlement.
 - Valider une facture.
 - Supprimer une facture.
 - Annuler une facture.
- Gestion d'entrepôt
 - Ajouter un entrepôt.
 - Modifier un entrepôt.
 - Consulter un entrepôt.
 - supprimer un entrepôt
- Gestion de devis/commandes
 - Ajouter un devis.
 - Modifier un devis.
 - Consulter un devis.
 - Supprimer un devis.
 - Validé un devis.
 - Consulter une commande.
 - Annuler une commande.
- Gestion des réceptions
 - Consulter les réceptions
 - Réceptionner les articles.
 - Retourner des articles.
 - Annuler une réception.

- Gestion des fournisseurs
 - Ajouter un fournisseur.
 - Modifier un fournisseur.
 - Consulter un fournisseur.
 - Supprimer un fournisseur.
- Gestion de taxes
 - Ajouter une taxe.
 - Modifier une taxe.
 - Consulter une taxe.
 - Supprimer une taxe.
- Gestion de liste de prix
 - Ajouter une liste de prix.
 - Modifier une liste de prix.
 - Consulter une liste de prix.
 - Supprimer une liste de prix.
- Gestion des réapprovisionnements
 - Consulter les réapprovisionnements.
 - Modifier et planifier les demandes d'approvisionnement.
 - Lancer les réapprovisionnements manuellement.

Au super administrateur

- Gestion de société et utilisateurs
 - Modifier les informations de la société.
 - Consulter les informations de la société.
 - Ajouter un utilisateur (Financier, Magasinier).
 - Modifier un utilisateur.
 - Consulter un utilisateur.
 - Supprimer un utilisateur.
- Gestion de devise et séquences
 - Ajouter une devise.
 - Modifier une devise.
 - Consulter une devise.
 - Supprimer une devise.
 - Modifier une séquence.
 - Consulter une séquence.

II.1.2. Besoins non fonctionnels

Ça représente des exigences qui ne concernent pas le comportement du système. Elles identifient les contraintes interne et externe du système. Dans notre cas l'application devra respecter les exigences suivantes :

- Performance : temps de réponse petit.
- Maintenabilité : apporter des corrections facilement.
- Fiabilité : précise et correcte.
- Intégrabilité : intégrer de nouvelles fonctionnalités facilement.
- Portabilité : elle peut fonctionner sur plusieurs plateformes.
- Disponibilité : réaliser une fonction requise à tout moment.
- Sécurité : personnaliser les accès selon l'utilisateur.

II.2. Spécification des acteurs de l'application

Un acteur c'est une entité externe qui peut interagir et utiliser les fonctions du système.

II.2.1. Les acteurs qui interviennent dans notre système

- **Le super administrateur** : C'est le cerveau de l'application, il gère toutes les fonctions de l'application ainsi que les utilisateurs et leurs rôles dans l'application.
- **Le financier** : il a pour rôle de gérer les fournisseurs, les commandes.
- **Le magasinier** : il pour rôle de gérer les articles, les entrepôts et la planification des réapprovisionnements.

II.2.2. Diagramme de contexte

Il s'agit d'un modèle conceptuel qui permet d'avoir une vue globale sur le système et les acteurs.

La figure suivante nous présente le diagramme de contexte de notre application :

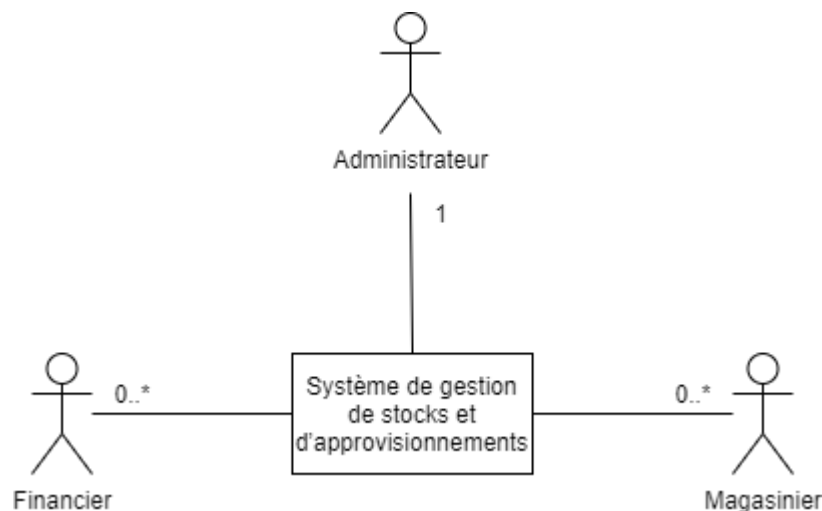


Figure 4. Diagramme de contexte.

II.3. Spécification de tâches

Dans ce point, nous allons spécifier les tâches que chaque acteur peut faire. Le tableau suivant résume ces tâches :

Acteurs	Tâches
Super administrateur	T01 : s'authentifier en tant qu'administrateur. T02 : Gestion des articles : -Ajouter un article. -Consulter un article. -Modifier un article. -Supprimer un article. -Consulter le stock. -Ajouter une catégorie d'article. -Modifier une catégorie d'article. -Consulter une catégorie d'article. -Supprimer une catégorie d'article. -Ajouter une unité de produit. -Modifier une unité de produit. -Consulter une unité de produit. -Supprimer une unité de produit. T03 : Gestion des factures -Consulter une facture. -Ajouter un règlement. -Valider une facture. -Supprimer une facture. -Annuler une facture. -Imprimer une facture. T05 : Gestion d'entrepôt -Ajouter un entrepôt. -Modifier un entrepôt. -Consulter un entrepôt. -supprimer un entrepôt T06 : Gestion de devis/commandes -Ajouter un devis. -Modifier un devis. -Consulter un devis. -Supprimer un devis. -Validé un devis. -Consulter une commande. -Annuler une commande. -Imprimer devis. -Imprimer commande. T07 : Gestion des réceptions -Consulter les réceptions -Réceptionner les articles. -Retourner des articles. -Annuler une réception.

	-Imprimer réception. T08 : Gestion des fournisseurs -Ajouter un fournisseur. -Modifier un fournisseur. -Consulter un fournisseur. -Supprimer un fournisseur. T09 : Gestion de taxes -Ajouter une taxe. -Modifier une taxe. -Consulter une taxe. -Supprimer une taxe. T10 : Gestion de liste de prix -Ajouter une liste de prix. -Modifier une liste de prix. -Consulter une liste de prix. -Supprimer une liste de prix. T11 : Gestion des réapprovisionnements -Consulter les réapprovisionnements. -Modifier et planifier les demandes d'approvisionnement. -Lancer les rapprovisionnements manuellement. T12 : Gestion de société et utilisateurs -Modifier les informations de la société. -Consulter les informations de la société. -Ajouter un utilisateur. -Modifier un utilisateur. -Consulter un utilisateur. -Supprimer un utilisateur. T13 : Gestion de devise et séquences -Ajouter une devise. -Modifier une devise. -Consulter une devise. -Supprimer une devise. -Modifier une séquence. -Consulter une séquence T14 : Se déconnecter
Financier	T15 : S'authentifier en tant que financier T17 : Gestion des factures -Consulter une facture. -Ajouter un règlement. -Valider une facture. -Supprimer une facture. -Annuler une facture. -Imprimer une facture. T18 : Gestion de devis/commandes

	-Ajouter un devis. -Modifier un devis. -Consulter un devis. -Supprimer un devis. -Validé un devis. -Consulter une commande. -Annuler une commande. -Imprimer un devis. -Imprimer une commande. T19 : Gestion des fournisseurs -Ajouter un fournisseur. -Modifier un fournisseur. -Consulter un fournisseur. -Supprimer un fournisseur. T20 : Gestion de taxes -Ajouter une taxe. -Modifier une taxe. -Consulter une taxe. -Supprimer une taxe. T21 : Gestion de liste de prix -Ajouter une liste de prix. -Modifier une liste de prix. -Consulter une liste de prix. -Supprimer une liste de prix. T22 : Se déconnecter
Magasinier	T23 : S'authentifier en tant que magasinier T24 : Gestion des articles : -Ajouter un article. -Consulter un article. -Modifier un article. -Supprimer un article. -Consulter le stock. -Ajouter une catégorie d'article. -Modifier une catégorie d'article. -Consulter une catégorie d'article. -Supprimer une catégorie d'article. -Ajouter une unité de produit. -Modifier une unité de produit. -Consulter une unité de produit. -Supprimer une unité de produit. T25 : Gestion des réapprovisionnements -Consulter les réapprovisionnements. -Modifier et planifier les demandes d'approvisionnement T26 : Gestion des réceptions -Consulter les réceptions -Réceptionner les articles. -Retourner des articles. -Annuler une réception.

	-Imprimer une réception. T27 : Gestion d'entrepôt -Ajouter un entrepôt. -Modifier un entrepôt. -Consulter un entrepôt. -supprimer un entrepôt T28 : Se déconnecter
--	--

II.4. Diagramme de cas d'utilisation

Il répond à la question « comment les acteurs d'un système interagissent avec lui ? ». C'est-à-dire les diagrammes de cas d'utilisation permettent de définir les relations entre le système et les acteurs.

II.4.1. Diagramme de cas d'utilisation de l'application

a) Diagramme de cas d'utilisation global

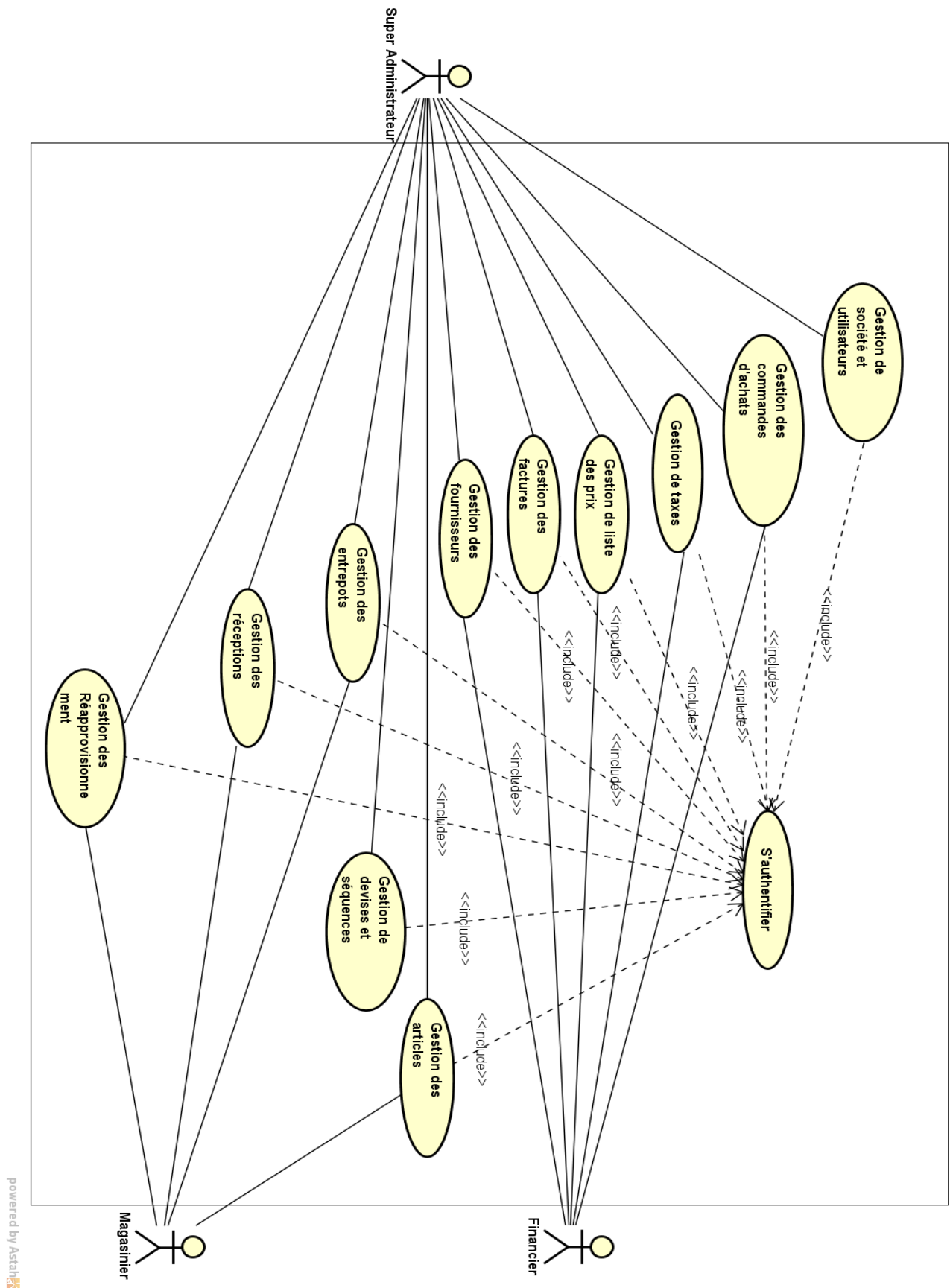
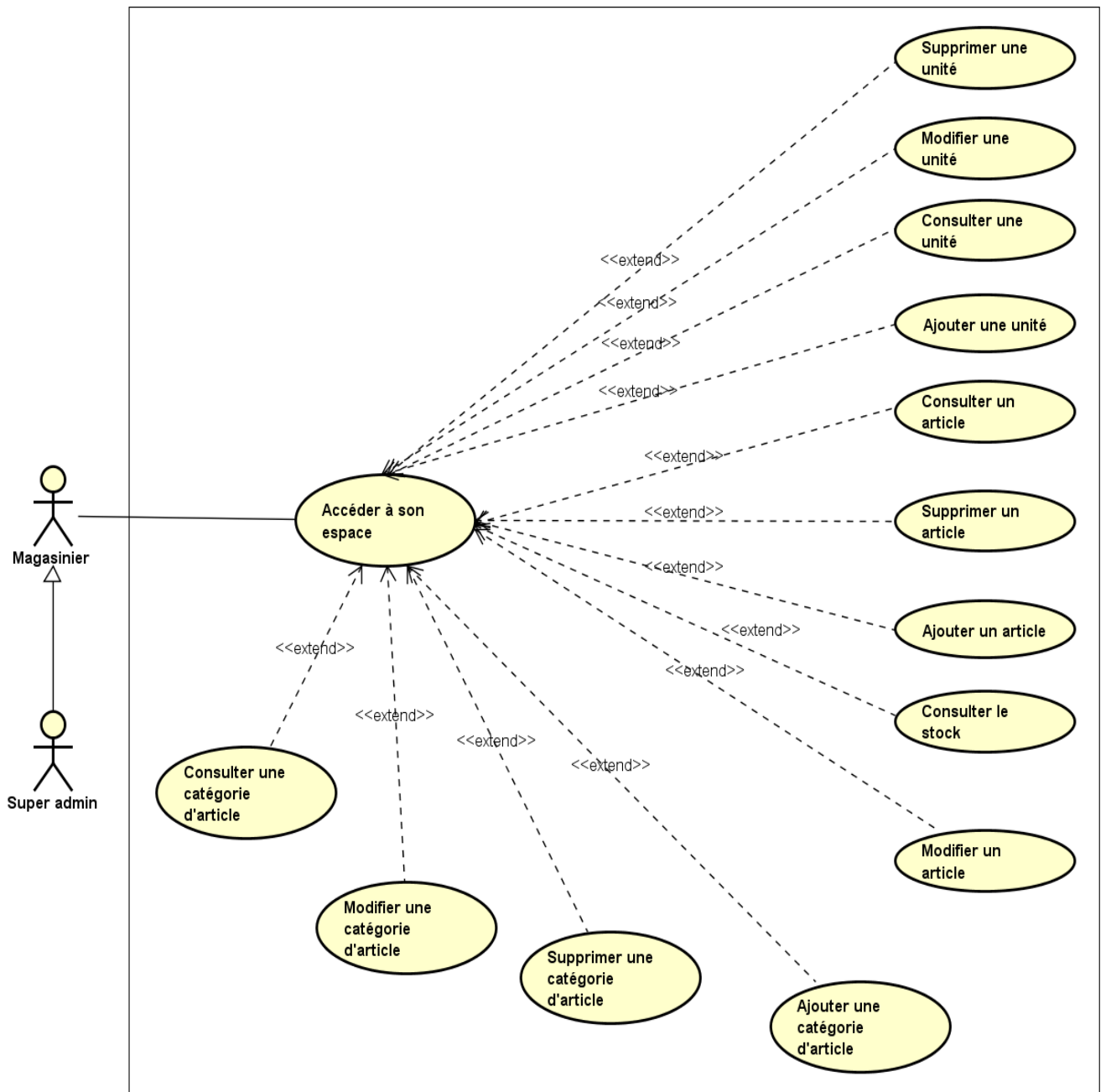


Figure 5. Diagramme de cas d'utilisation global.

b) Diagramme de cas d'utilisation gestion d'articles



powered by Astah

Figure 6. Diagramme cas d'utilisation gestion des articles.

c) Diagramme cas d'utilisation gestion de commandes d'achats

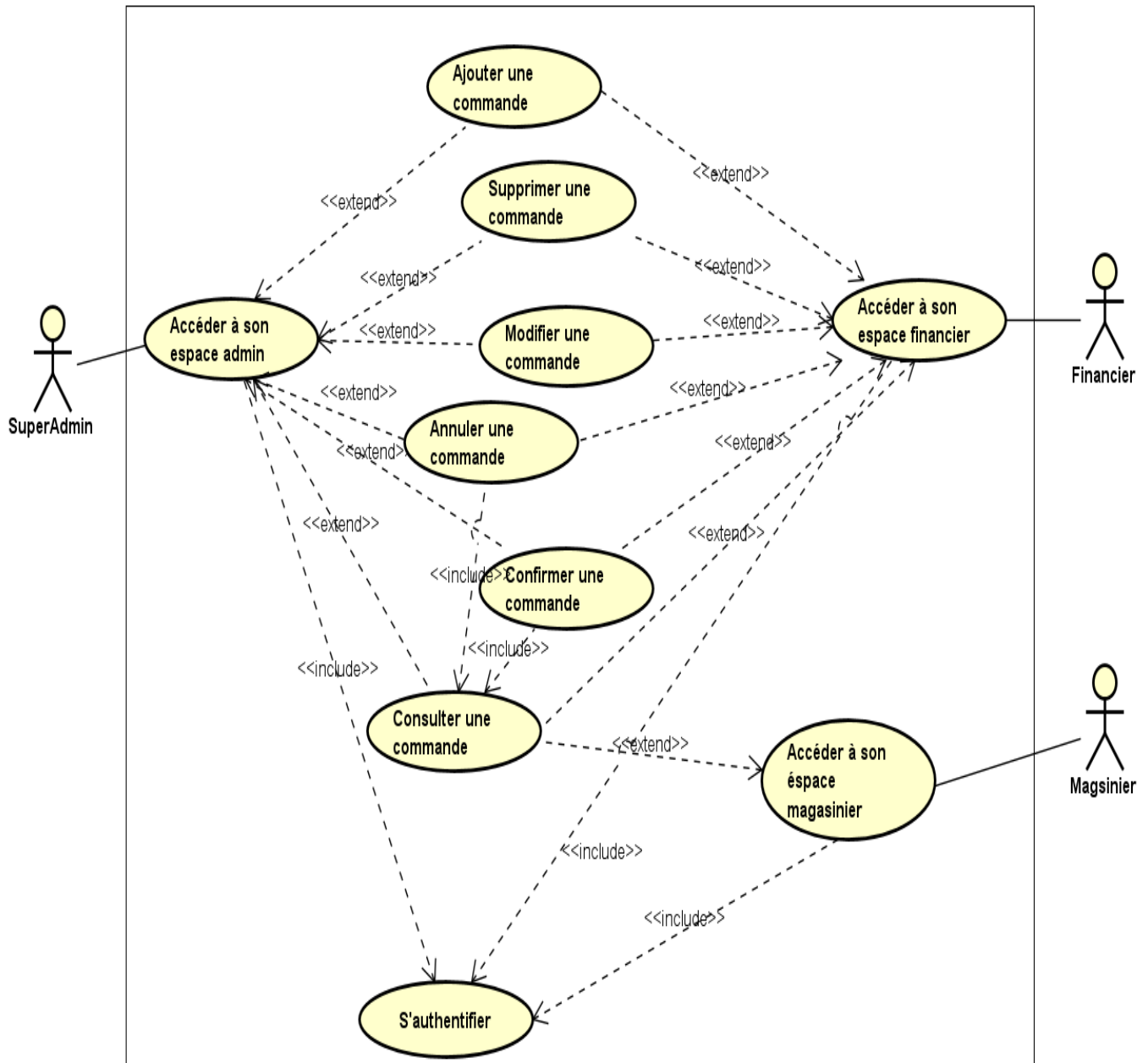


Figure 7. Diagramme cas d'utilisation gestion de commandes d'achats.

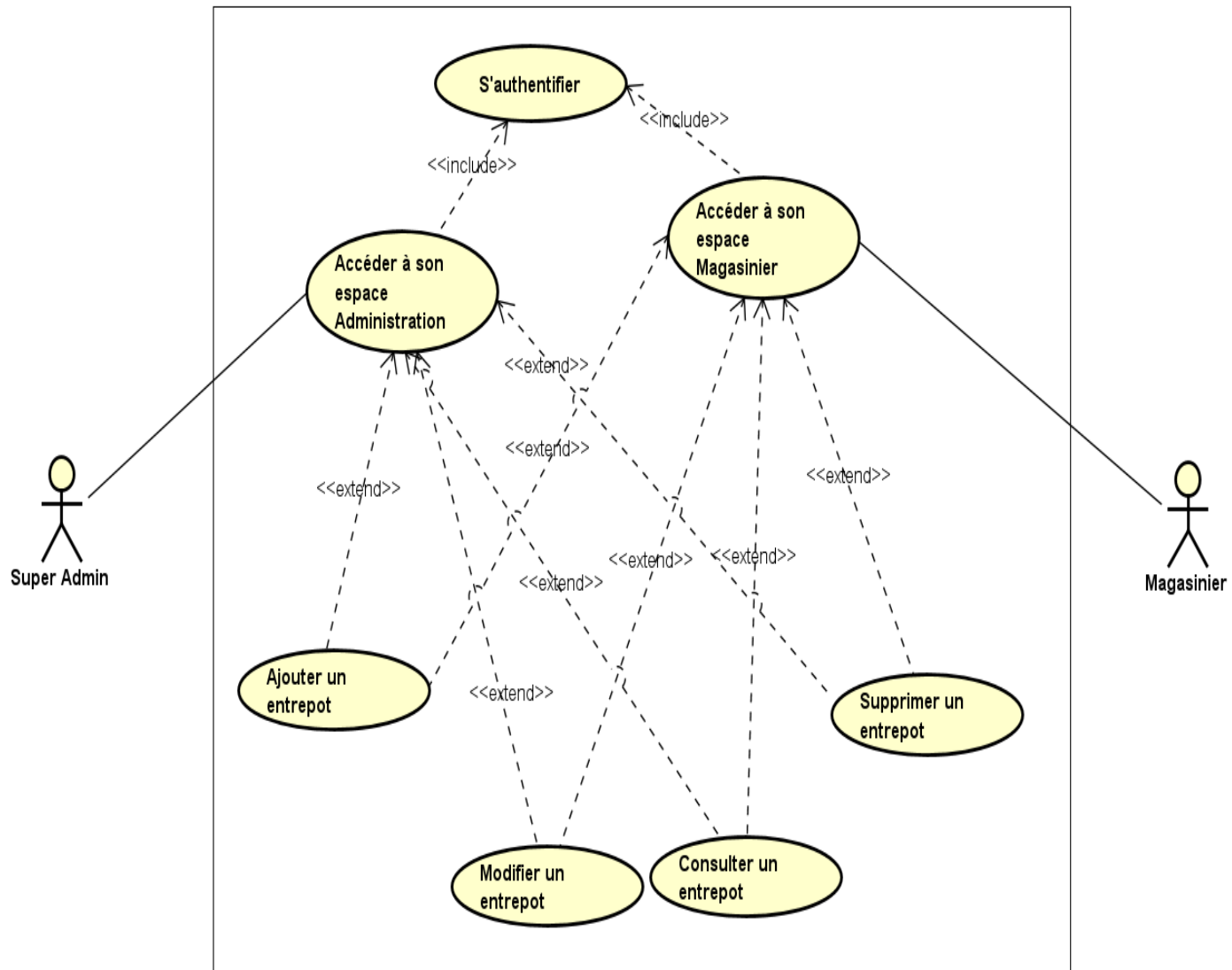
d) Diagramme cas d'utilisation gestion d'entrepôt

Figure 8. Diagramme cas d'utilisation gestion d'entrepôt.

e) Diagramme cas d'utilisation gestion de réapprovisionnements

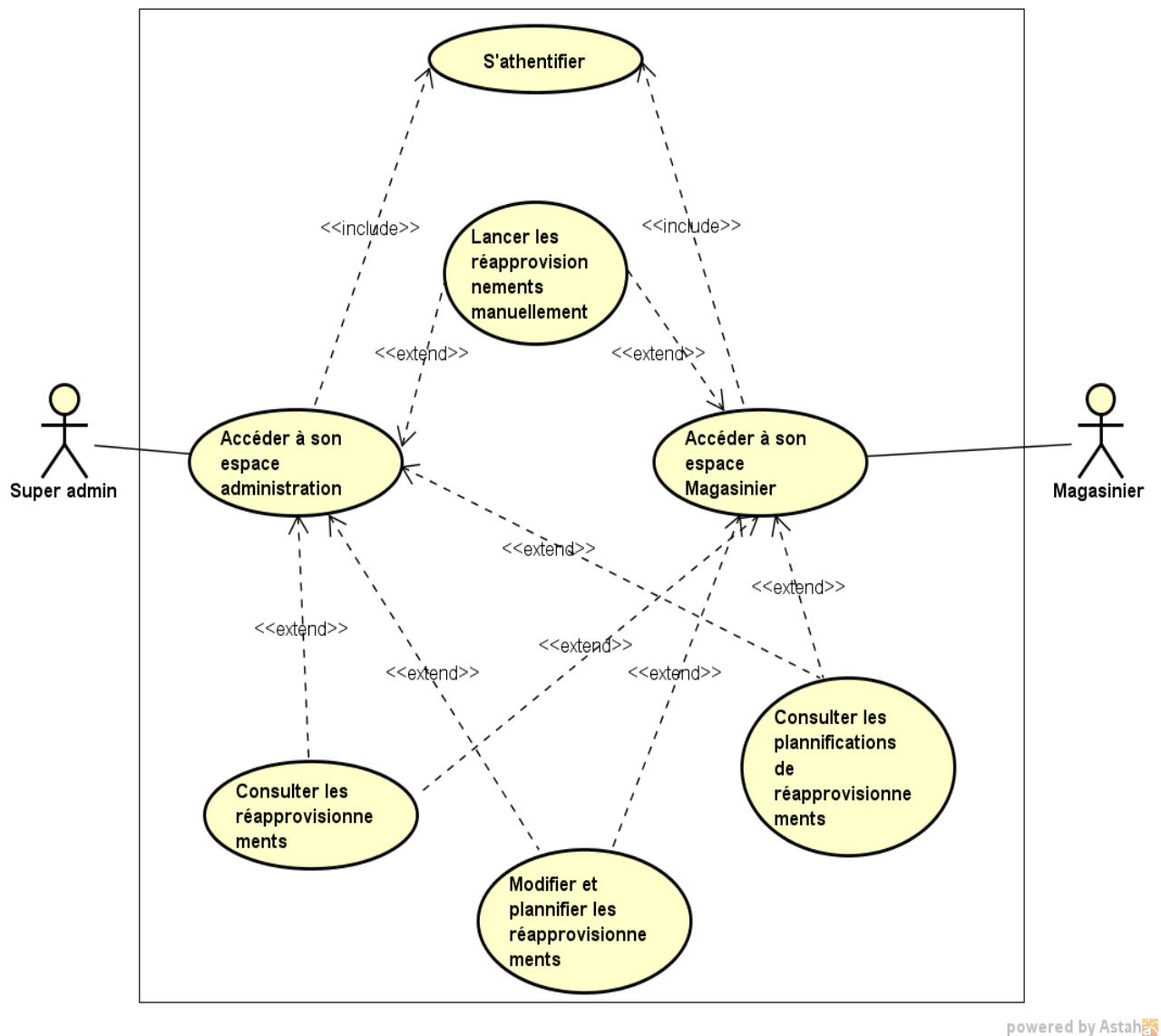
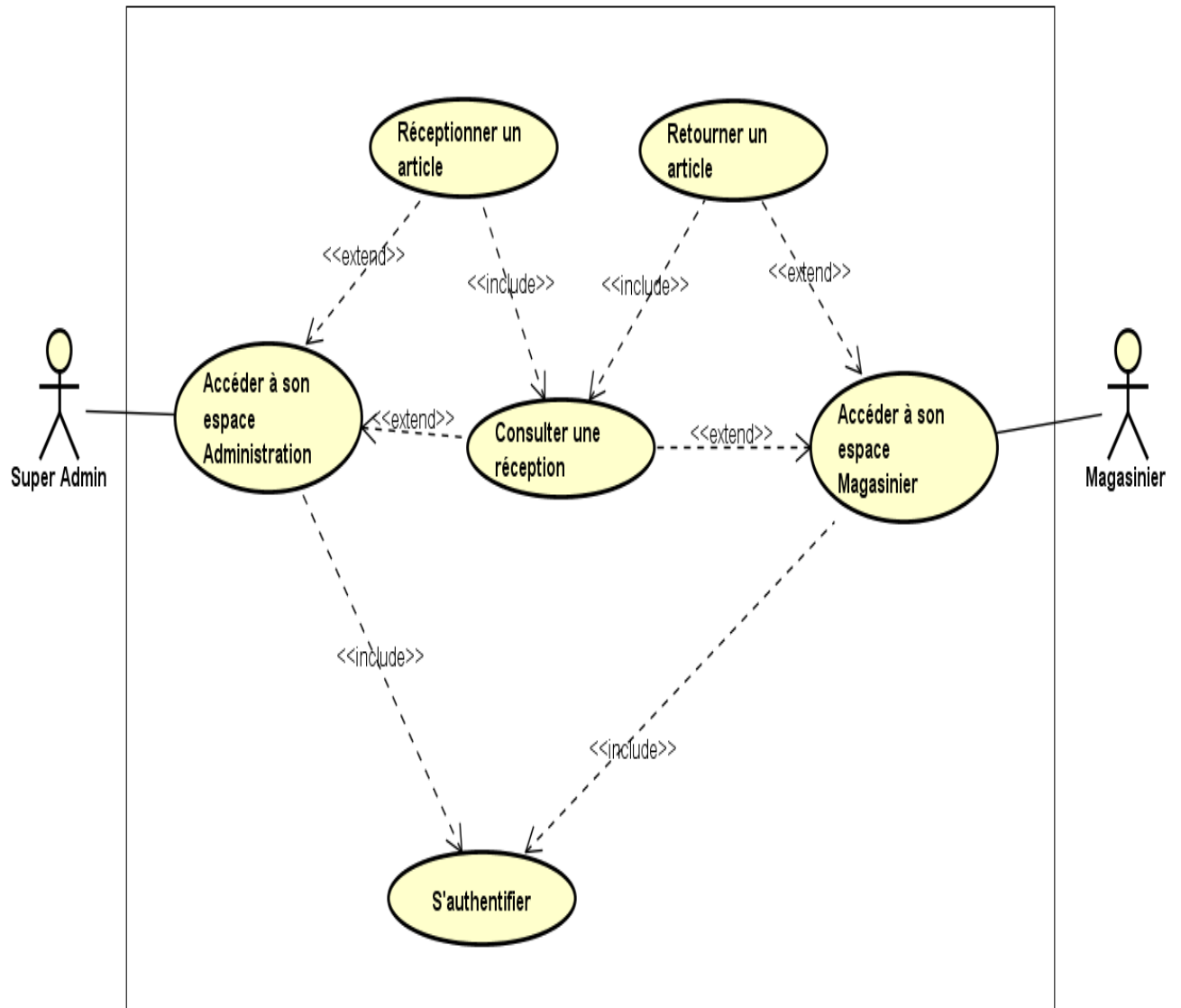


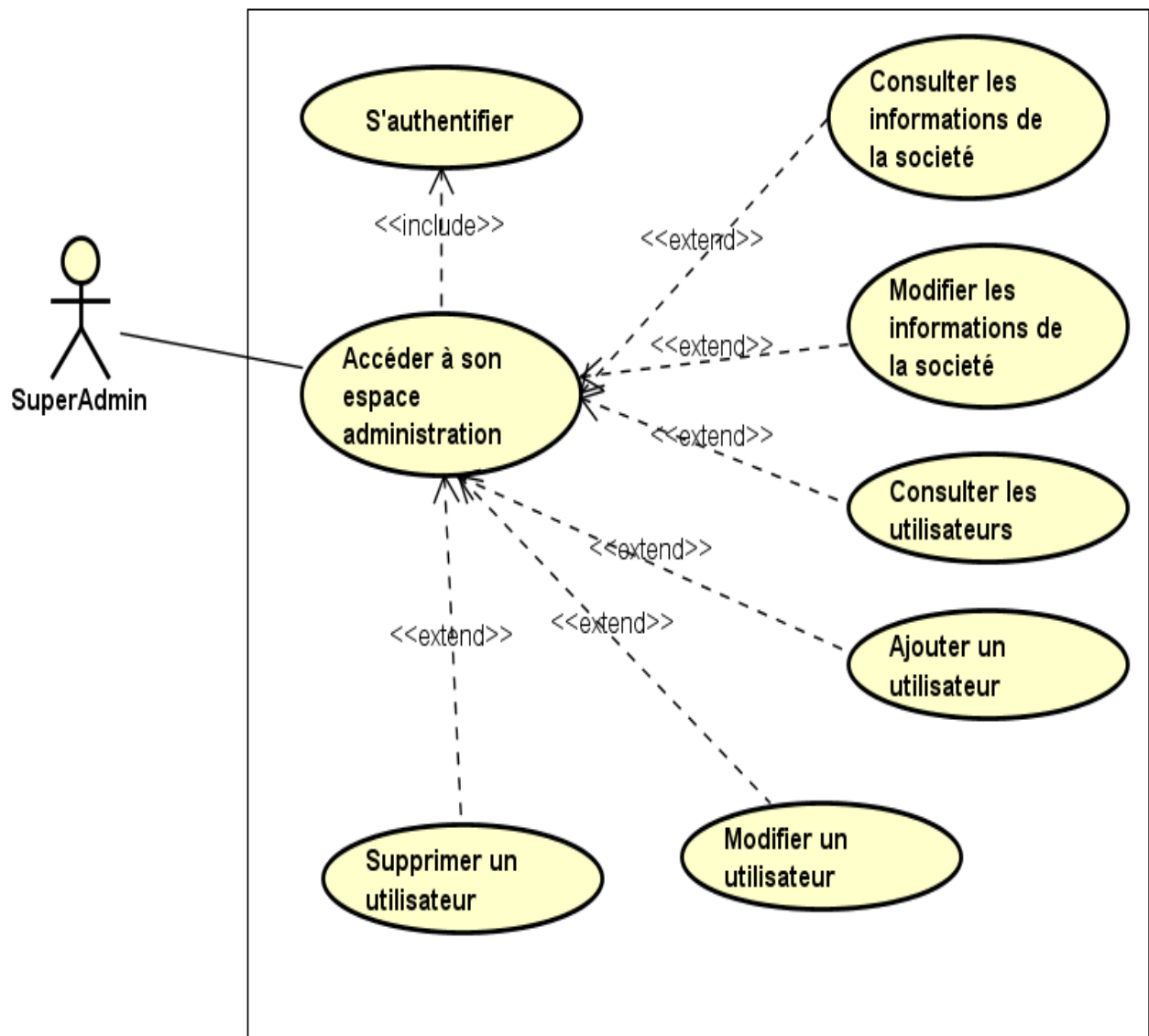
Figure 9. Diagramme cas d'utilisation gestion de réapprovisionnements.

f) Diagramme cas d'utilisation gestion des réceptions

powered by Astah

Figure 10. Diagramme cas d'utilisation gestion des réceptions.

g) Diagramme cas d'utilisation gestion société et utilisateur



powered by Astah

Figure 11. Diagramme cas d'utilisation gestion société et utilisateurs.

h) Diagramme cas d'utilisation Gestion des factures

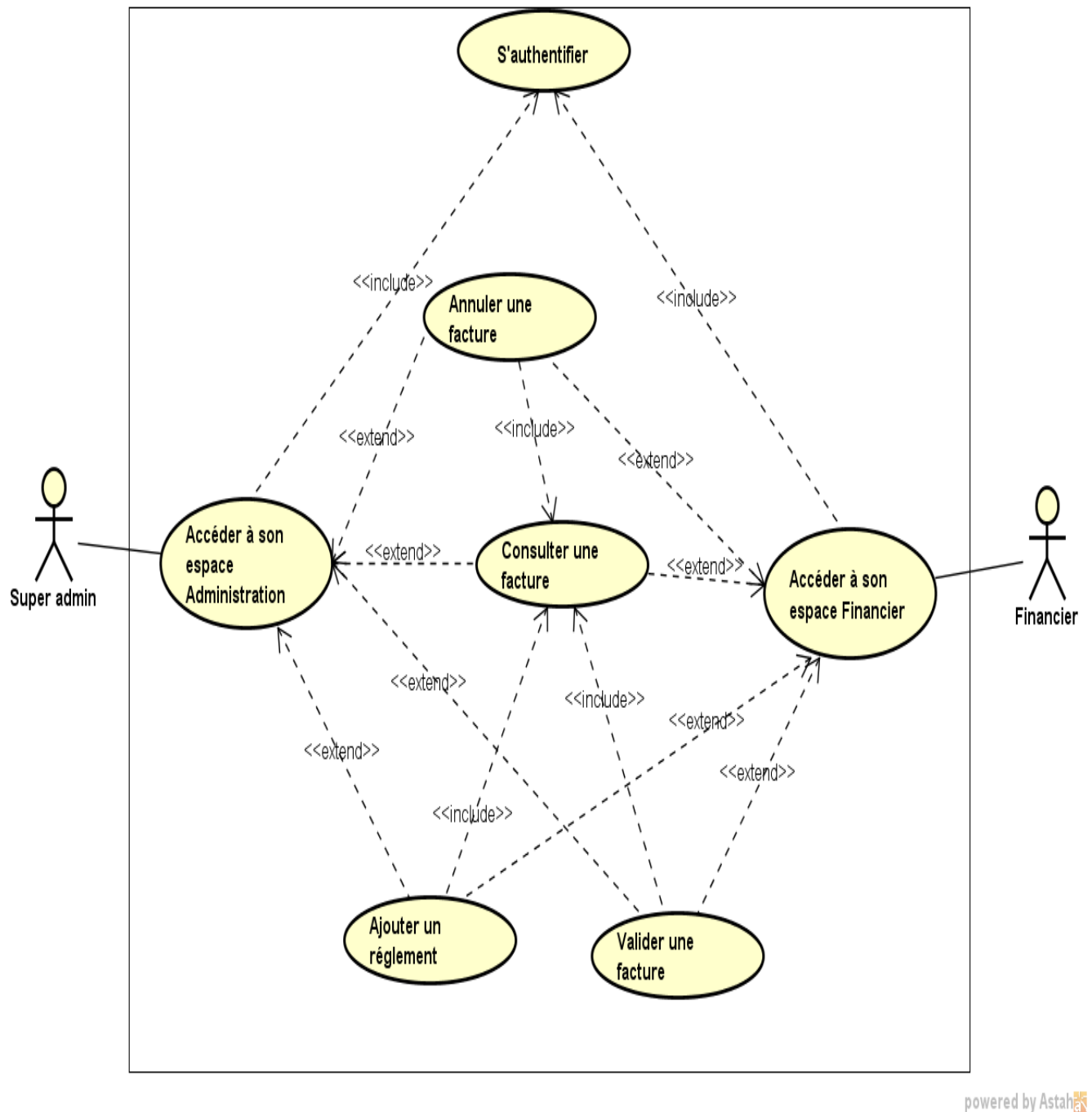


Figure 12. Diagramme cas d'utilisation gestion des factures.

i) Diagramme cas d'utilisation gestion des fournisseurs

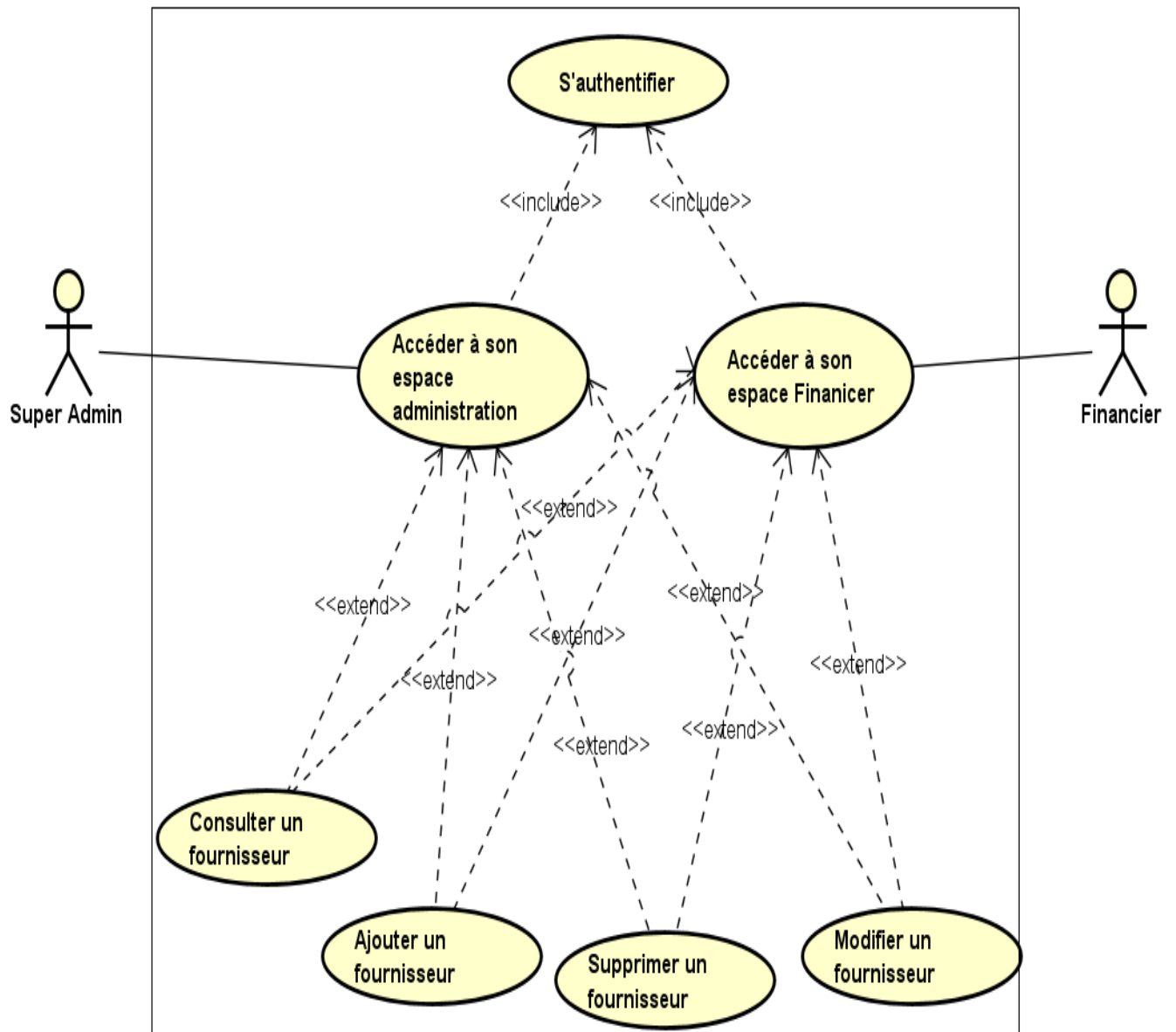


Figure 13. Diagramme cas d'utilisation gestion des fournisseurs.

Après avoir exprimé les fonctionnalités de notre application, nous allons passer à la prochaine étape qui est la conception. Cette dernière consiste à modéliser et détailler la phase d'analyse.

Dans cette étape, nous illustrons les diagrammes de séquences, diagramme d'activité et diagramme de classe.

II.5. Diagramme de séquence

Les diagrammes de séquence décrivent les interactions du logiciel avec son environnement, ils permettent d'identifier et de décrire les fonctionnalités du système.

Remarque :

Vu le nombre important de cas d'utilisation, nous avons choisi de décrire les cas d'utilisation suivants :

- Gestion d'articles (cas ajouter un article).
- Gestion des réceptions (cas réceptionner ou retourner des articles).
- Gestion d'articles (cas ajouter confirmer un achat).
- Gestion d'articles (cas planifier un réapprovisionnement).

a) Diagramme de séquence du cas d'utilisation « Gestion d'articles » exemple « cas Ajouter un article »

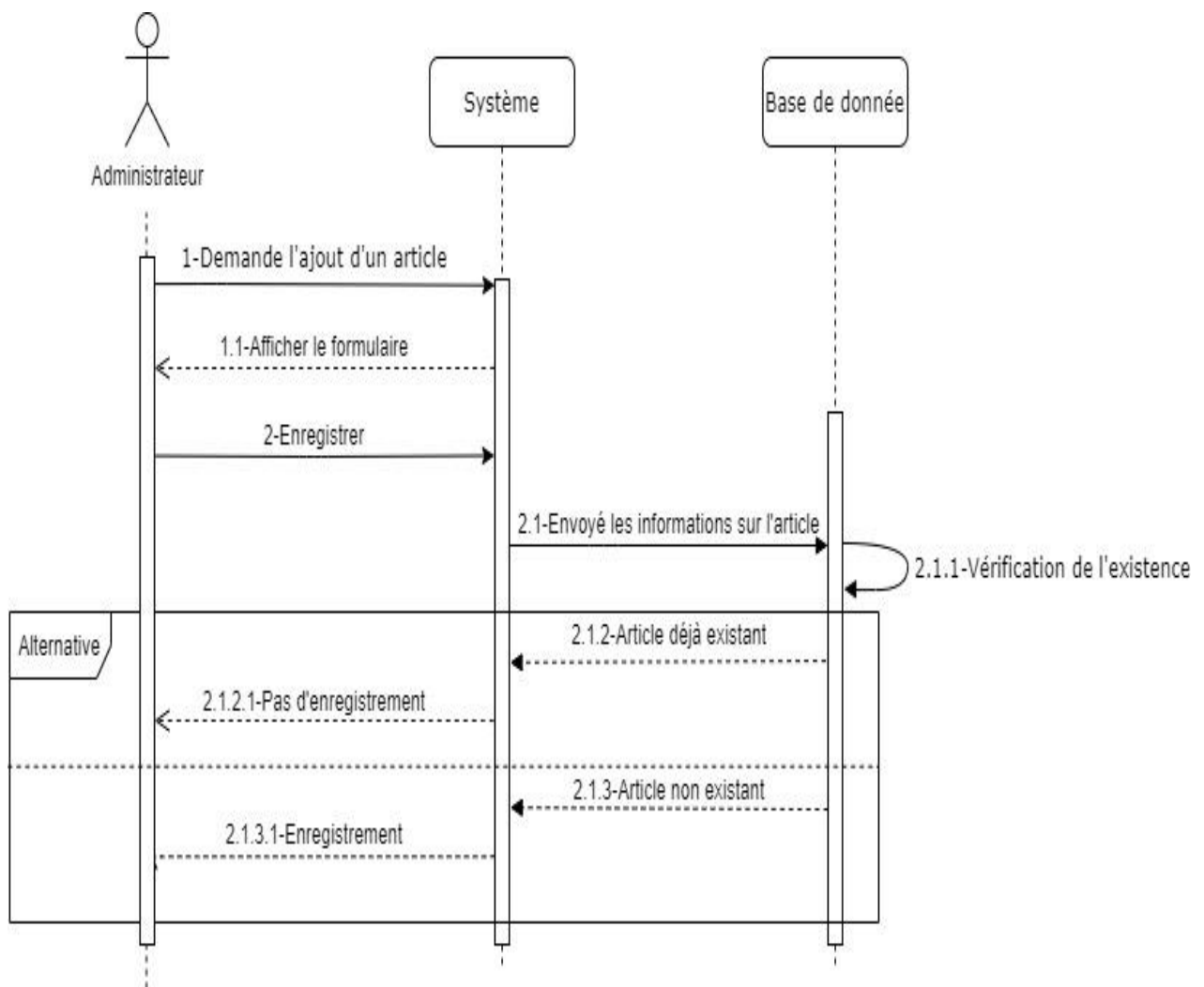


Figure 14. Diagramme de séquence pour ajouter un article.

b) Diagramme de séquence du cas d'utilisation « Gestion des réceptions » exemple « cas réceptionner ou retourner des articles »

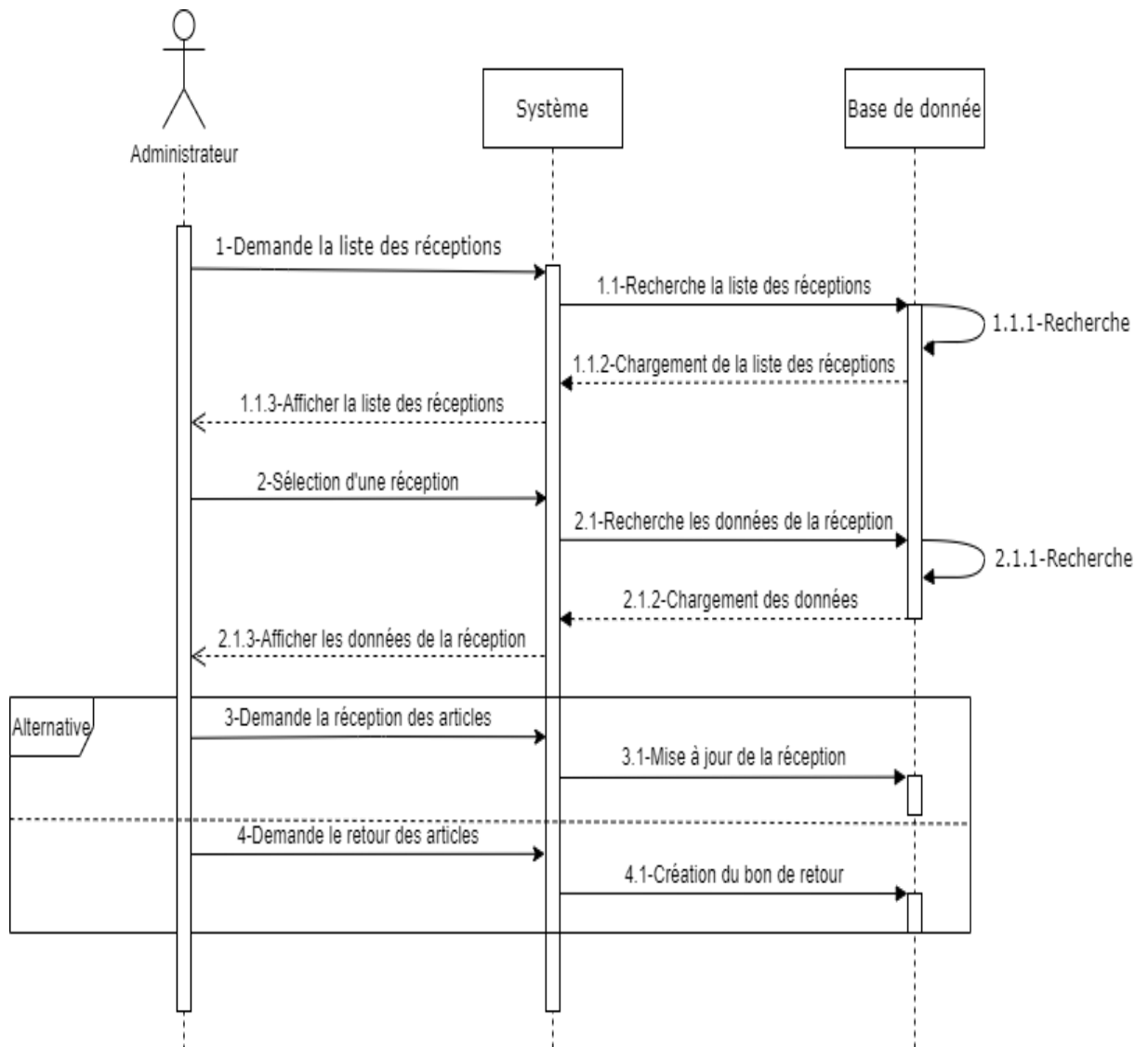


Figure 15. Diagramme de séquence pour réceptionner ou retourner des articles.

c) Diagramme de séquence du cas d'utilisation « Gestion de devis/commandes »
exemple « cas confirmer un achat »

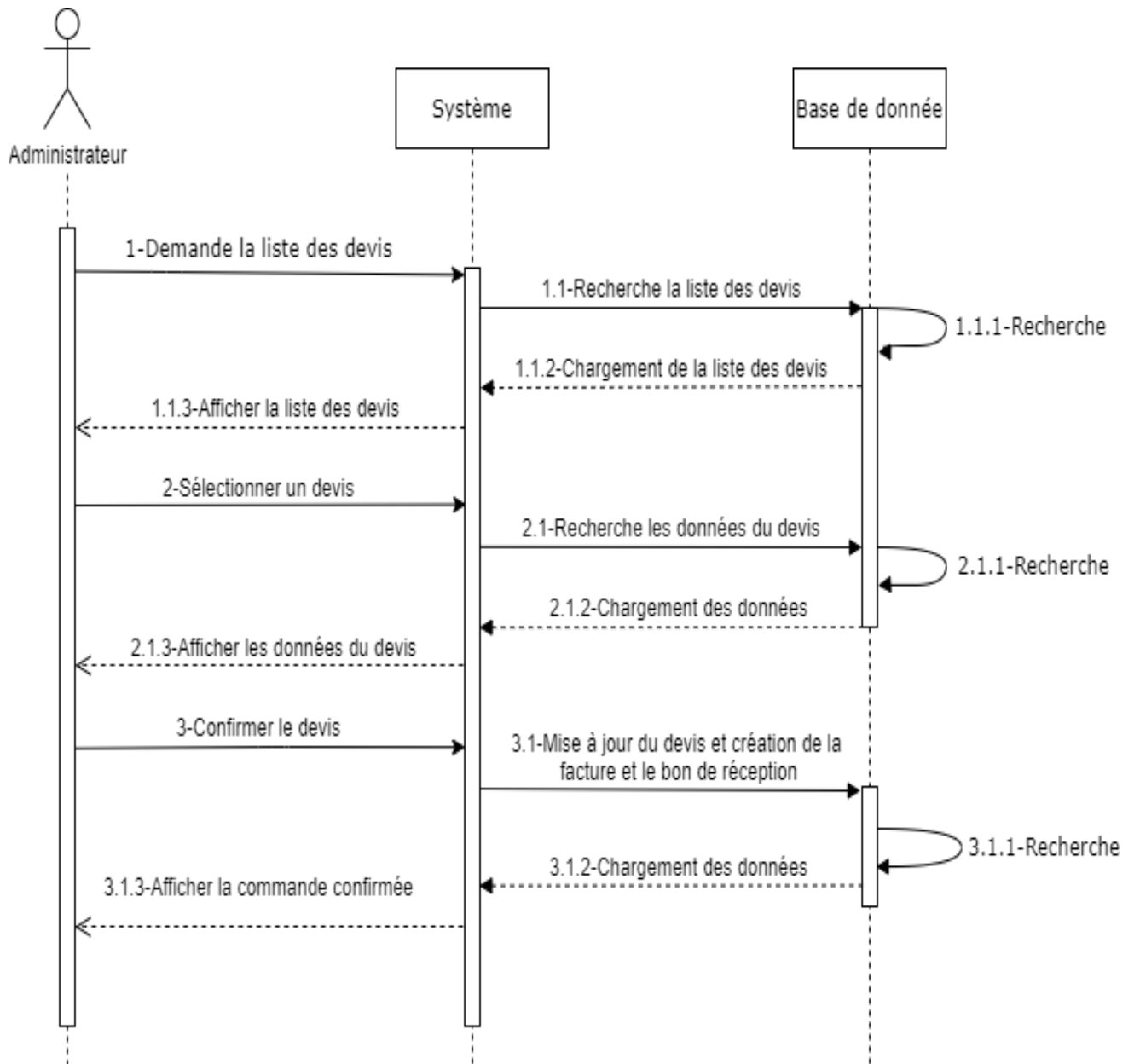


Figure 16. Diagramme de séquence cas confirmer un achat.

**d) Diagramme de séquence du cas d'utilisation « Gestion de réapprovisionnements »
exemple « cas planifier un réapprovisionnement »**

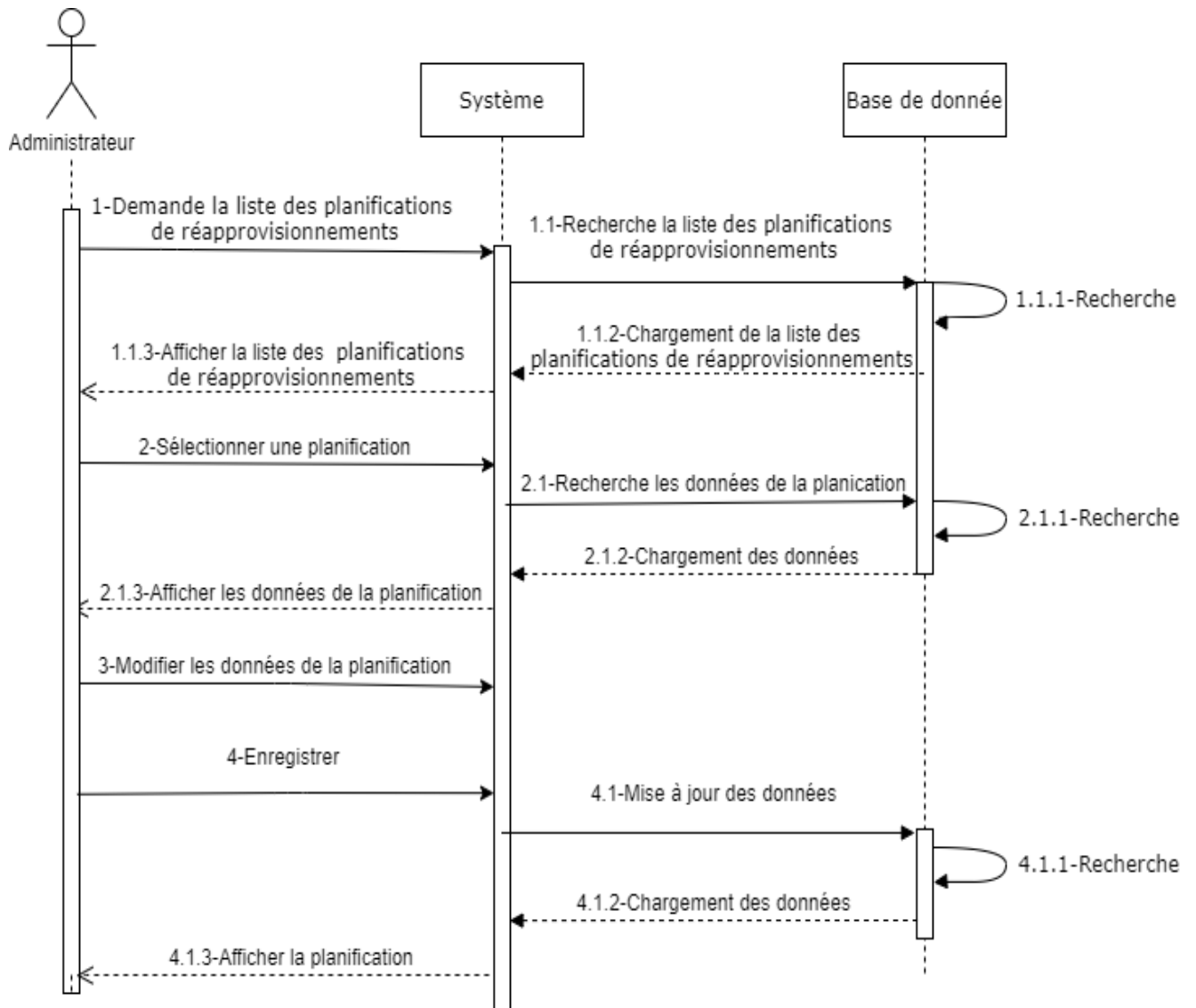


Figure 17. Diagramme de séquence cas planifier un réapprovisionnement.

II.6. Diagramme d'activité [25]

Les diagrammes d'activité permettent de mettre l'accent sur les traitements, ils tendent à exprimer le comportement dynamique à un plus haut niveau que le font les diagrammes de séquences. Par ailleurs, les diagrammes servent à modéliser un déroulement d'un cas d'utilisation.

a) Diagramme d'activité du cas d'utilisation « gestion de factures » exemple « cas ajouter un règlement »

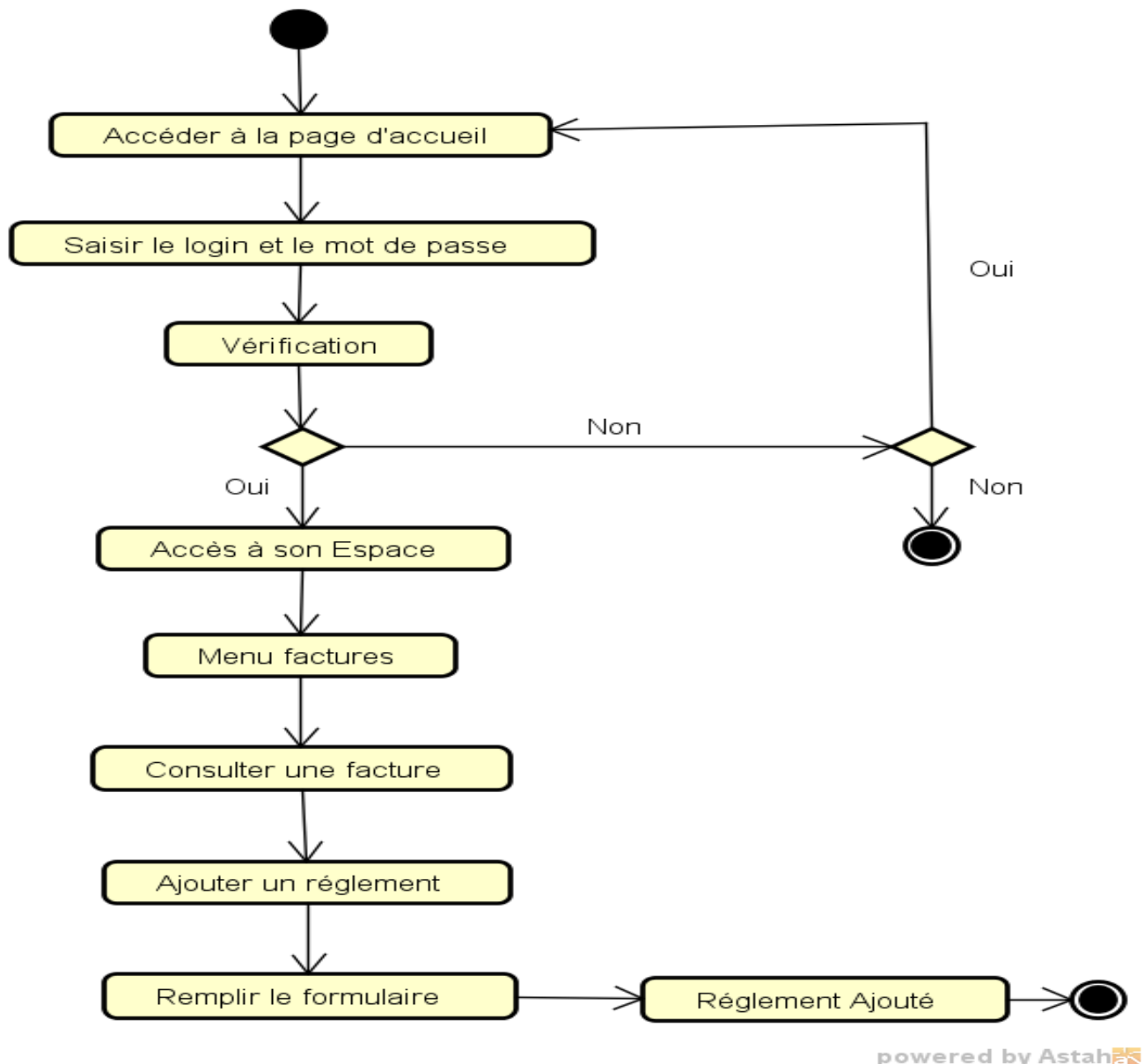


Figure 18. Diagramme d'activité ajouter un règlement.

b) Diagramme d'activité du cas d'utilisation « Gestion d'articles » exemple « cas ajouter un article »

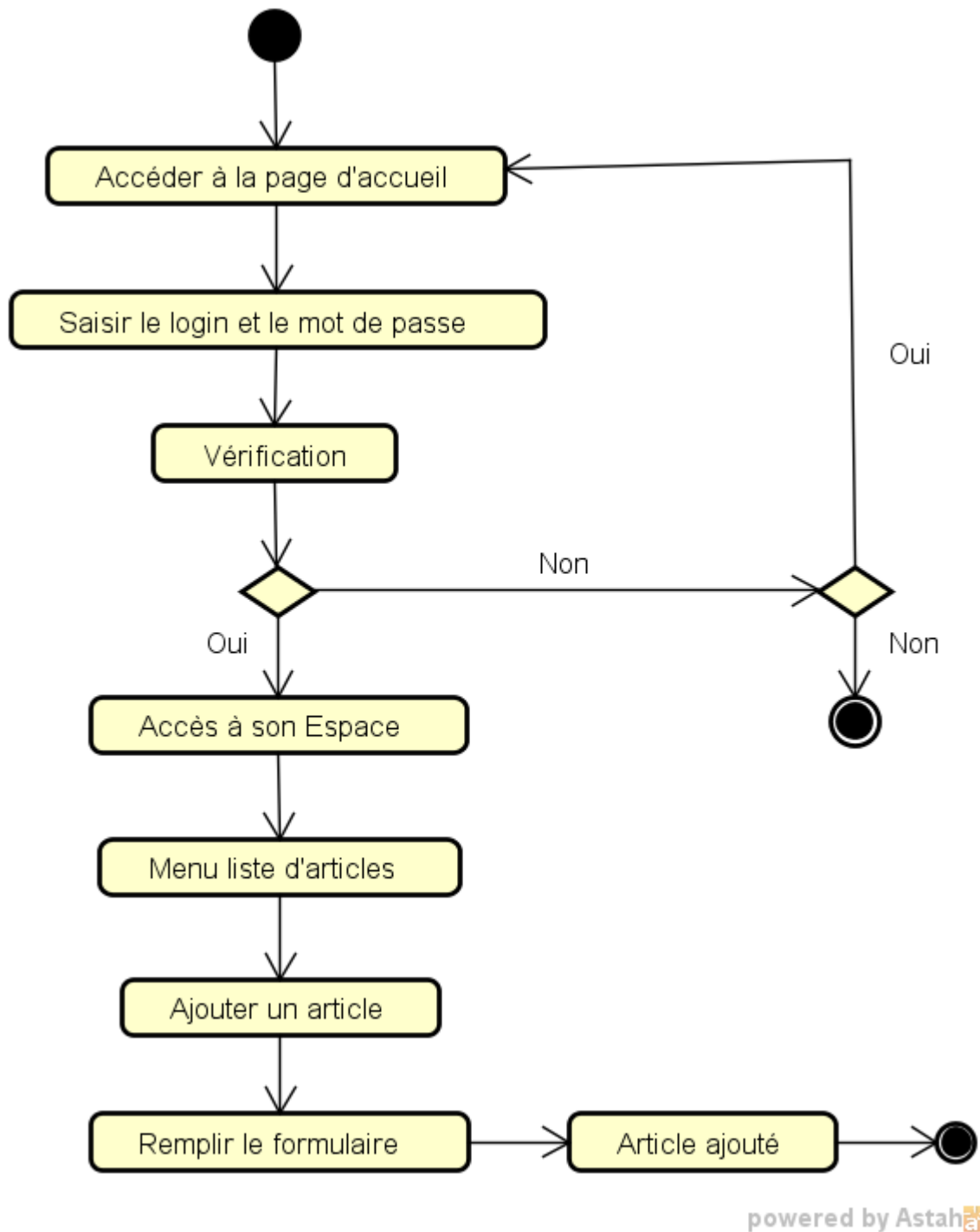
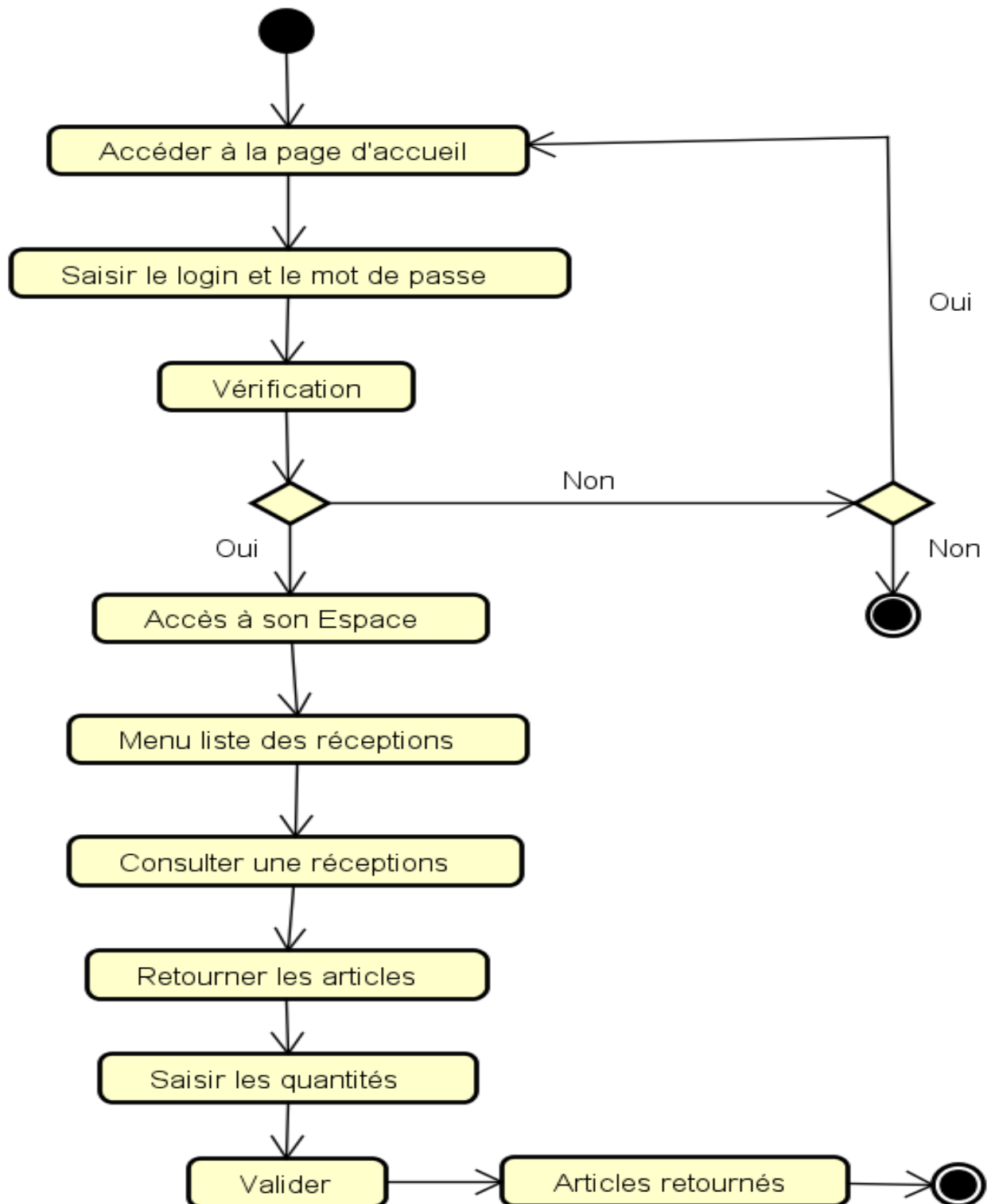


Figure 19. Diagramme d'activité ajouter un article.

c) Diagramme d'activité du cas d'utilisation « Gestion de réceptions » exemple « cas retourner des article »



powered by Astah

Figure 20. Diagramme d'activité retourner des articles.

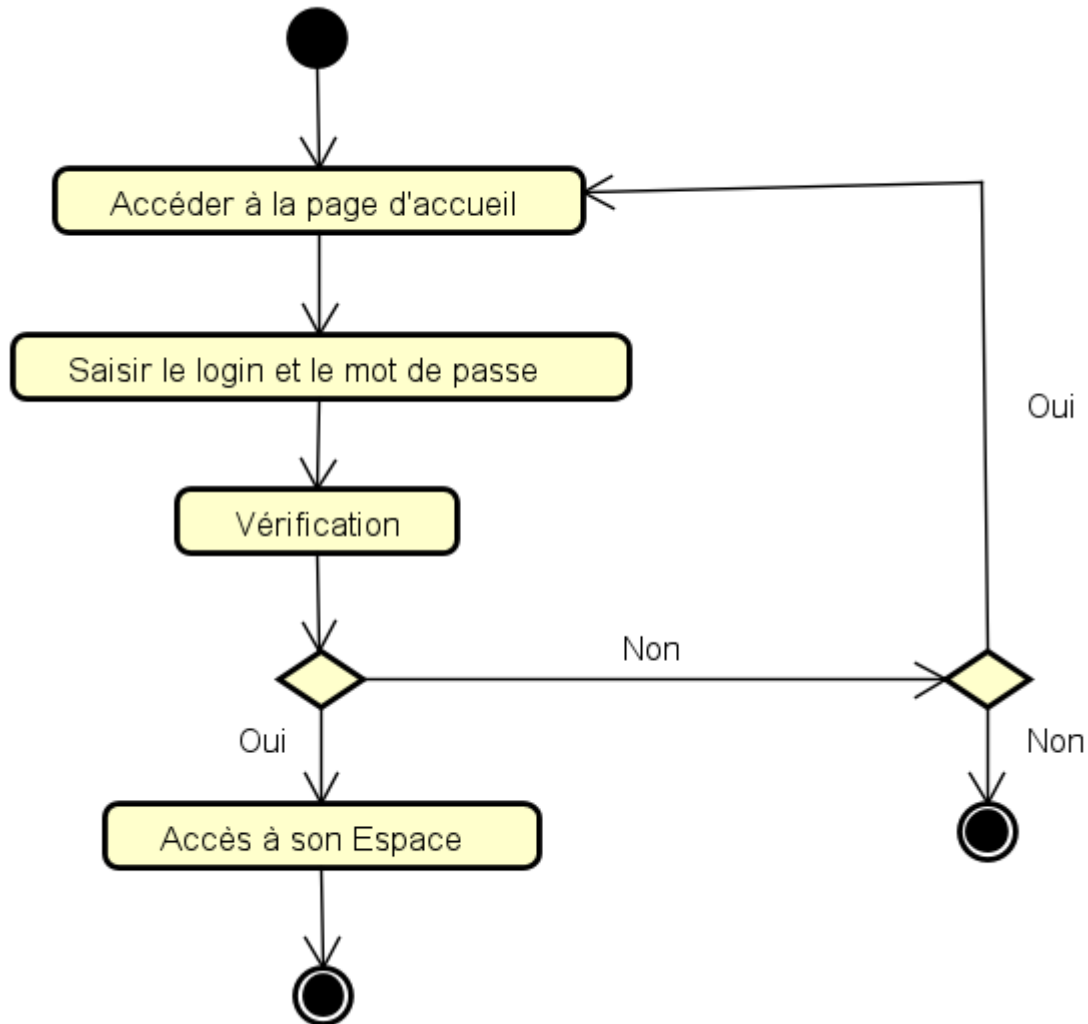
d) Diagramme d'activité authentification

Figure 21. Diagramme d'activité authentification.

II.7. Diagramme de classe [26]

Il s'agit d'une représentation statique des entités qui composent le système et de leurs relations. Son intérêt est de modéliser les classes et présenter l'ensemble des informations gérer par le système.

Remarque :

Pour ne pas encombrer notre diagramme de classe nous allons d'abord définir nos classes avec leurs attributs et méthodes, ensuite nous allons définir les associations et les relations entre ces classes.

Catégorie d'articles
- id : int - name : char
+ ajouter() : void + modifier() : void + consulter() : void + supprimer() : void

Figure 22. Classe catégorie d'articles.

Société
- id : int - name : char - adresse : char - code_postal : int - ville : char - pays : char - phone : int - email : char - web : char - logo : char - pdg : char - capital : double - nif : char - rib : char
+ modifier() : void + consulter() : void

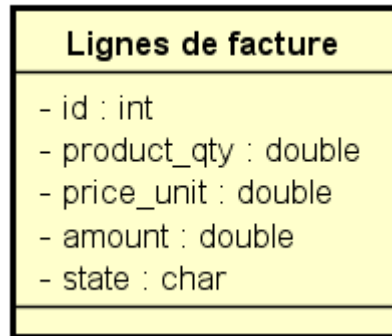
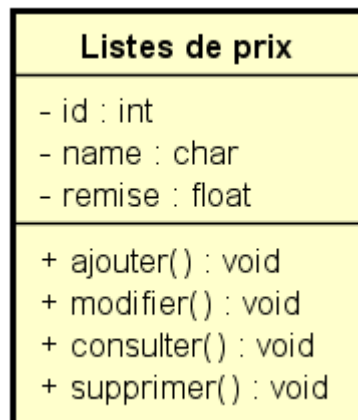
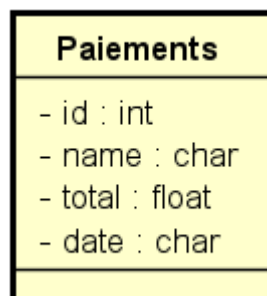
Figure 23. Classe Société.

Devises
- id : int - name : char - symbole : char
+ ajouter() : void + modifier() : void + consulter() : void + supprimer() : void

Figure 24. Classe Devises.

Factures
- id : int - name : char - date : char - date_due : char - state : char - amount_ht : double - remise : double - amount_tax : double - amount_total : double
+ valider() : void + ajouter un règlement() : void + consulter() : void + supprimer() : void + annuler() : void + imprimer() : void

Figure 25. Classe Factures.

*Figure 26. Classe Lignes de facture.**Figure 27. Classe Listes de prix.**Figure 28. Classe Paiements.*

Unités des articles
- id : int - name : char
+ ajouter() : void + modifier() : void + consulter() : void + supprimer() : void

Figure 29. Classe Unités des articles.

Articles
- id : int - name : char - ref : char - sate_price : double - alerte_stock : double - optimal_stock : double - physical_stock : double - virtual_stock : double - procurement_method : char - standard_price : double - cump : double - type : char
+ ajouter() : void + modifier() : void + consulter() : void + supprimer() : void

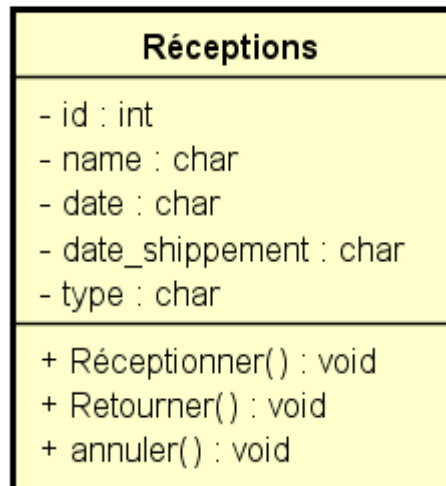
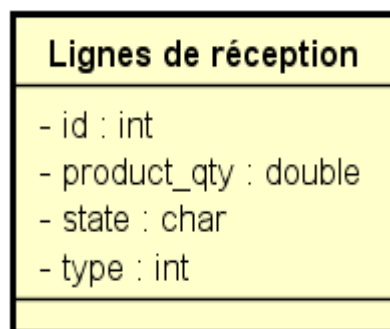
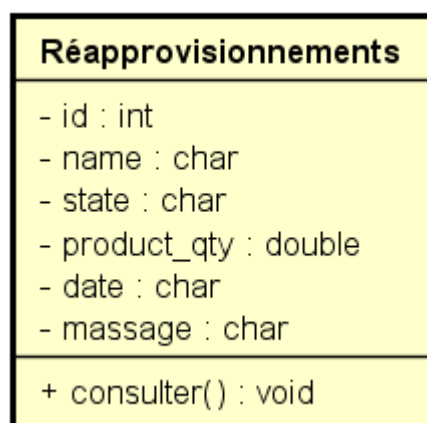
Figure 30. Classe Articles.

Commandes
- id : int - name : char - condition_reglement : int - date : char - state : char - date_shippement : int - amount_ht : double - amount_tax : double - amount_total : double
+ ajouter() : void + modifier() : void + consulter() : void + supprimer() : void + valider() : void + annuler() : void

Figure 31. Classe Commandes.

Lignes de commande
- id : int - product_qty : double - price_unit : double - amount : double - state : char
+ ajouter() : void + modifier() : void + consulter() : void + supprimer() : void

Figure 32. Classe Lignes de commande.

*Figure 33. Classe Réceptions.**Figure 34. Classe Lignes de réception.**Figure 35. Classe Réapprovisionnement.*

Planifications
- id : int - name : char
+ consulter() : void + modifier() : void

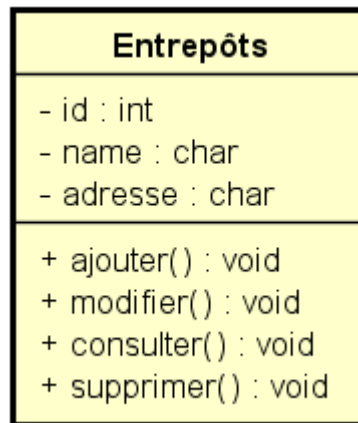
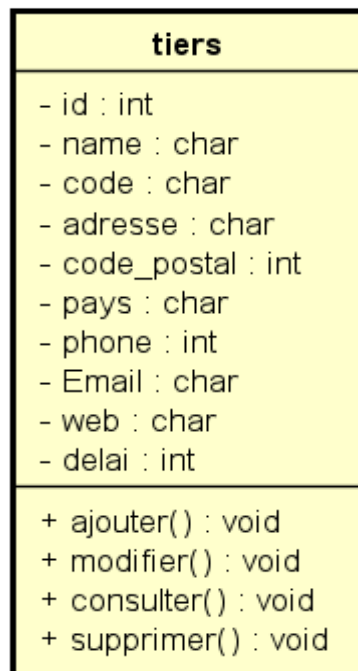
Figure 36. Classe Planifications.

Lignes de planification
- id : int - date : char - product_qty : double - state : char - message : char
+ ajouter() : void + modifier() : void + supprimer() : void

Figure 37. Classe Lignes de planification.

Taxes
- id : int - name : int - taux : double
+ ajouter() : void + modifier() : void + consulter() : void + supprimer() : void

Figure 38. Classe Taxes.

*Figure 39. Classe entrepôts.**Figure 40. Classe Tiers.*

Utilisateurs
- id : int - name : char - email : char - password : char - role : int - images : char
+ ajouter() : void + modifier() : void + consulter() : void + supprimer() : void

Figure 41. Classe Utilisateurs.

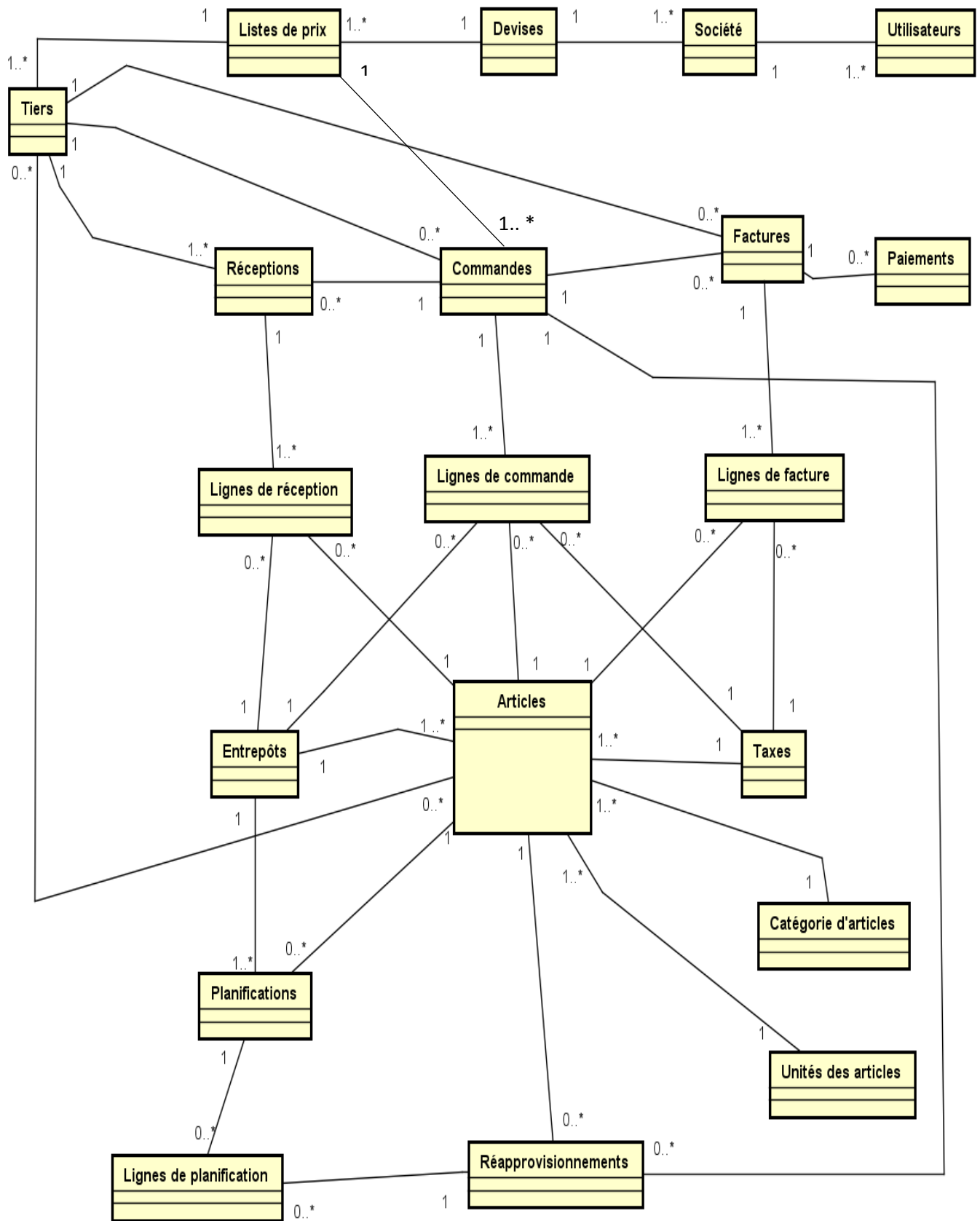


Figure 42. Diagramme de classe global.

II.8. Conception de la base de données

Comme nous avons terminé de modéliser notre application avec les différents diagrammes, nous allons passer à une autre étape qui consiste à mettre en œuvre une base de données sur un serveur web. Celle-ci est nécessaire car elle permet de mettre des données à la disposition des utilisateurs ainsi que de les manipuler selon leurs besoins.

Une base de données c'est un ensemble de données stockées qui servent les besoins d'une ou plusieurs applications.

Pour cela, afin de concevoir notre base de données nous allons procéder par la traduction du diagramme de classe vers un modèle relationnel.

II.8.1. Modèle relationnel

Le modèle relationnel est basé sur une organisation des données sous forme de tables. Le passage du diagramme de classe vers le modèle relationnel consiste à respecter certaines règles de passage.

- **Les règles de passage**

Transformation des classes :

Chaque classe est une relation et pour chaque relation il faut lui définir une clé primaire qui est l'un de ses attributs.

Transformation des associations : On distingue trois types d'associations :

- Association « ?..* ».
- Association « ?..1 ».
- Association « *..* ».

- **Règle 1** : Présence de la cardinalité « ?..1 » d'un côté de l'association * :

L'identifiant de la classe qui est associée à la cardinalité « ?..1 » devient une clé étrangère dans l'autre classe.

- **Règle 2** : Présence de la cardinalité « ?..* » des deux côtés de l'association :

L'association se transforme en une table et cette table a comme identifiant les deux clés primaires des deux classes ainsi qu'éventuellement d'autres attributs.

En se basant sur ces deux règles on arrive à notre modèle relationnel suivant :

Utilisateurs (id, name, email, password, role, image, #company_id).

Société (id, name, adresse, code_postal, ville, pays, phone, email, web, logo, pdg, capital, nif, nis, rib, #currency_id).

Devises (id, name, symbole).

Listes de prix (id, name, remise, #currency_id).

Tiers (id, name, code, adresse, code_postal, pays, phone, Email, web, delai, #list_price_id).

Taxes (id, name, taux).

Entrepôts (id, name).

Unités des articles (id, name).

Catégorie d'articles (id, name).

Articles (id, name, ref, sale_price, alerte_stock, optimal_stock, physical_stock, virtual_stock, procurement_method, standard_price, cump, type, #category_product_id, #taxe_id, #warehouse_id, #product_unity_id).

Commandes (id, name, condition_reglement, date, state, date_shippement, amount_ht, amount_tax, amount_total, #tier_id, #list_price_id).

Lignes de commandes (id, product_qty, price_unit, amount, state, #product_id, #purchase_order_id, #warehouse_id, #taxe_id).

Réceptions (id, name, date, state, date_shippement, type, #purchase_order_id, #tier_id).

Lignes de réception (id, product_qty, state, type, #reception_id, #product_id, #warehouse_id).

Factures (id, name, date, date_due, state, amount_ht, amount_tax, amount_total, #tier_id, #purchase_order_id).

Lignes de facture (id, product_qty, price_unit, amount, state, #product_id, #invoice_id, #taxe_id).

Paielements (id, name, total, date, #invoice_id).

Planifications (id, name, #product_id, #warehouse_id).

Réapprovisionnement (id, name, state, date, product_qty, message, #purchase_order_id, #product_id).

Lignes de planification (id, date, state, message, #replishippement_order_id, #replishippement_rule_id).

Constitution (#product_id, #tier_id, delai, price, qtt_min).

Remarque : Ci-dessous la table de correspondance pour les clés étrangères :

Champs	fait référence à
currency_id	l'attribut « id » dans « Devises »
company_id	l'attribut « id » dans « Société »
list_price_id	l'attribut « id » dans « Listes de prix »
warehouse_id	l'attribut « id » dans « Entrepôts »
taxe_id	l'attribut « id » dans « Taxes »
category_product_id	l'attribut « id » dans « Catégorie d'articles »
product_unity_id	l'attribut « id » dans « Unités des articles »
product_id	l'attribut « id » dans « Articles »

tier_id	l'attribut « id » dans « Tiers »
purchase_order_id	l'attribut « id » dans « Commandes »
reception_id	l'attribut « id » dans « Réceptions »
invoice_id	l'attribut « id » dans « Factures »
replishippement_rule_id	l'attribut « id » dans « Planifications »

II.8.2. Modèle physique

- **Table Utilisateurs**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de l'utilisateur	Primaire
name	Varchar	40	Nom de l'utilisateur	
email	Varchar	60	Email de l'utilisateur	
password	Varchar	30	Mot de passe de l'utilisateur	
role	Varchar	20	Role de l'utilisateur	
image	Varchar	255	Image de l'utilisateur	
comapny_id	int	11	L'identifiant de la compagnie	Etrangère

- **Table Société**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de la compagnie	Primaire
name	Varchar	40	Nom de la compagnie	
email	Varchar	60	Email de la compagnie	
adresse	Varchar	60	adresse de la compagnie	
code_postal	int	10	Code postal de la compagnie	
ville	Varchar	30	Ville où se situe la compagnie	
pays	Varchar	30	pays où se situe la compagnie	
phone	int	10	Téléphone de la compagnie	
web	Varchar	80	Site web de la compagnie	
logo	Varchar	255	Logo de la	

			compagnie	
pdg	Varchar	50	Directeur général de la compagnie	
capital	Double	20	Le capital de la compagnie	
Nif	Varchar	15	Le numéro d'identification fiscal de la compagnie	
rib	Varchar	20	Le relevé d'identification bancaire de la compagnie	
Currency_id	Int	8	L'identifiant de la devise	Etrangère

- **Table Devises**

Nom du champ	Type	Taille	Signification	Clé
id	Int	8	Identifiant de la devise	Primaire
name	Varchar	40	Nom de la devise	
symbole	Varchar	6	symbole de la devise	

- **Table Listes de prix**

Nom du champ	Type	Taille	Signification	Clé
id	Int	8	Identifiant de la liste de prix	Primaire
name	Varchar	40	Nom de la liste de prix	
remise	Double	2,2	taux de remise	
Currency_id	Varchar	8	identifiant de la devise	Etrangère

- **Table Tiers**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant du fournisseur	Primaire
name	Varchar	40	Nom du fournisseur	
Email	Varchar	60	Email du fournisseur	
adresse	Varchar	60	adresse du	

			fournisseur	
code_postal	int	10	Code postal du fournisseur	
delai	int	3	Delai de livraison en jours	
pays	Varchar	30	pays du fournisseur	
phone	int	10	Téléphone du fournisseur	
web	Varchar	80	Site web du fournisseur	
code	Varchar	20	code du fournisseur	
List_price_id	int	8	Identifiant de la liste de prix	Etrangère

- **Table Taxes**

Nom du champ	Type	Taille	Signification	Clé
id	Int	8	Identifiant de la taxe	Primaire
name	Varchar	40	Nom de la taxe	
taux	Double	2,2	Taux de la taxe	

- **Table Warehouses**

Nom du champ	Type	Taille	Signification	Clé
id	Int	8	Identifiant de l'entrepôt	Primaire
name	Varchar	40	Nom de l'entrepôt	
adresse	Varchar	80	Adresse de l'entrepôt	

- **Table Product_unities**

Nom du champ	Type	Taille	Signification	Clé
id	Int	8	Identifiant de l'unité	Primaire
Name	Varchar	40	Nom de l'unité	

- **Table Catégorie d'articles**

Nom du champ	Type	Taille	Signification	Clé
id	Int	8	Identifiant de la catégorie de produits	Primaire
name	Varchar	40	Nom de la catégorie de produits	

- **Table Articles**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant du produit	Primaire
name	Varchar	40	Nom du produit	
ref	Varchar	20	Code du produit	
sale_price	Double	20,4	Prix de vente du produit	
alerte_stock	Double	20,4	Stock d'alerte du produit	
optimal_stock	Double	20,4	Stock optimal du produit	
Physical_stock	Double	20,4	Stock physique du produit	
virtual_stock	Double	20,4	Stock virtuel du produit	
procurement_method	Varchar	20	La méthode d'approvisionnement du produit	
standard_price	Double	20,4	Le prix d'achat du produit	
cump	Double	20,4	le coût moyen pondéré du produit	
type	Varchar	20	le type du produit	
category_product_id	Int	8	identifiant de la catégorie du produit	Etrangère
taxe_id	Int	8	identifiant de la taxe par défaut du produit	Etrangère
warehouse_id	Int	8	Identifiant de l'entrepôt par défaut du produit	Etrangère
product_unity_id	Int	8	Identifiant de l'unité du produit	Etrangère

- **Table Commandes**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de la commande	Primaire
name	Varchar	40	Numéro de séquence de la commande	
condition_reglement	Int	3	Condition de règlement de la commande	
date	Date	AAAA/MM/JJ	Date de la commande	
state	Varchar	12	état de la commande	
date_shippement	Date	AAAA/MM/JJ	Date de réception prévue	
amount_ht	Double	20,4	Montant hors taxes de la commande	
amount_ta	Double	20,4	Montant taxes de la commande	
amount_total	Double	20,4	Montant total de la commande	
tier_id	int	11	Identifiant du fournisseur	Etrangère
list_price_id	int	8	Identifiant de la liste de prix	Etrangère

- **Table Lignes de commandes**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de la ligne de commande	Primaire
product_qty	Double	20,4	Quantité commandée	
price_unit	Double	20,4	Prix unitaire	
amount	Double	20,4	Total	
state	Varchar	12	état	
purchase_order_id	Int	11	Identifiant de la commande	Etrangère
taxe_id	Int	8	identifiant de la taxe	Etrangère

warehouse_id	Int	8	Identifiant de l'entrepôt	Etrangère
product_id	Int	11	Identifiant du produit	Etrangère

- **Table Réceptions**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de la réception	Primaire
name	Varchar	40	Numéro de séquence de la réception	
date_shippement	Date	AAAA/MM/JJ	Date de réception prévue	
date	Date	AAAA/MM/JJ	Date de la commande de la réception	
state	Varchar	12	état de la réception	
type	Varchar	8	Type de la réception	
tier_id	int	11	Identifiant du fournisseur	Etrangère
purchase_order_id	Int	11	Identifiant de la commande	Etrangère

- **Table Lignes de réception**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de la ligne de commande	Primaire
product_qty	Double	20,4	Quantité à réceptionner	
state	Varchar	12	état	
type	Varchar	8	Type de la réception	
reception_id	Int	11	Identifiant de la réception	Etrangère
warehouse_id	Int	8	Identifiant de l'entrepôt de la réception	Etrangère
product_id	Int	11	Identifiant du produit	Etrangère

- **Table Factures**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de la facture	Primaire
name	Varchar	40	Numéro de séquence de la facture	
date	Date	AAAA/MM/JJ	Date de la facture	
state	Varchar	12	état de la facture	
date_due	Date	AAAA/MM/JJ	Date d'échéance de la facture	
amount_ht	Double	20,4	Montant hors taxes de la facture	
remise	Double	20,4	Total de la remise	
amount_tax	Double	20,4	Montant taxes de la facture	
amount_total	Double	20,4	Montant total de la facture	
tier_id	int	11	Identifiant du fournisseur	Etrangère
purchase_order_id	int	11	Identifiant de la commande	Etrangère

- **Table Lignes de facture**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de la ligne de facture	Primaire
product_qty	Double	20,4	Quantité	
price_unit	Double	20,4	Prix unitaire	
amount	Double	20,4	Total	
state	Varchar	12	état	
invoice_id	Int	11	Identifiant de la commande	Etrangère
taxe_id	Int	8	identifiant de la taxe	Etrangère
product_id	Int	11	Identifiant du produit	Etrangère

- **Table Paiements**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant du paiement	Primaire
date	Date	AAAA/MM/JJ	Date du paiement	
name	Varchar	40	Numéro de séquence du paiement	
total	Double	20,4	Montant payé	
invoice_id	Int	11	Identifiant de la facture	Etrangère

- **Table Planifications**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant du réapprovisionnement planifié	Primaire
name	Varchar	40	Nom du réapprovisionnement planifié	
product_id	Int	11	Identifiant du produit	Etrangère
warehouse_id	Int	8	Identifiant de l'entrepôt	Etrangère

- **Table Réapprovisionnements**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant du réapprovisionnement automatique	Primaire
name	Varchar	40	Numéro de séquence du réapprovisionnement automatique	
state	Varchar	12	état du réapprovisionnement automatique	
date	Date	AAAA/MM/JJ	Date du réapprovisionnement automatique	
message	Varchar	120	Message d'erreur	
product_qty	Double	20,4	Quantité à s'approvisionner	
purchase_order_id	Int	11	Identifiant de la commande	Etrangère

			déclenchée	
product_id	Int	11	Identifiant du produit	Etrangère

- **Table Lignes de planification**

Nom du champ	Type	Taille	Signification	Clé
id	Int	11	Identifiant de la ligne du réapprovisionnement planifié	Primaire
date	Date	AAAA/MM/JJ	Date planifiée	
state	Varchar	12	état	
product_qty	Double	20,4	Quantité à s'approvisionner	
message	Varchar	120	Message d'erreur	
replishippement_rule_id	Int	11	Identifiant du réapprovisionnement planifié	Etrangère

- **Table Constitution**

Nom du champ	Type	Taille	Signification	Clé
tier_id	Int	11	Identifiant du fournisseur	Primaire
product_id	Int	11	identifiant du produit	Primaire
delai	Int	3	délai de la réception	
price	Double	20,4	Prix du fournisseur	
qtt_min	Double	20,4	Quantité minimale requise	

Conclusion

Ce chapitre a donné une vision globale de notre travail, nous avons pu donner l'aspect analytique et conceptuel de l'application grâce aux différents diagrammes qui ont été décrit en langage UML.

Pour la phase d'analyse le diagramme de cas d'utilisation et de contexte ont été modélisés, puis le diagramme de séquence ainsi que le diagramme de classe ont été traduits du processus de visualisation des cas d'utilisation dans le cadre de la phase de conception.

Ainsi, nous sommes passés à la conception de la base de données et des tables utilisées.

Le chapitre qui suit sera consacré à la mise en œuvre et à la réalisation de notre application.

Chapitre IV

Réalisation

Introduction

La réalisation est l'étape qui suit la phase d'analyse et conception, c'est l'aboutissement du projet et la raison du projet en lui-même. La réalisation d'application web est principalement réalisée grâce à l'utilisation de plusieurs outils, et chaque outil joue un rôle déterminant pour satisfaire l'une des tâches essentielles.

Ces tâches essentielles sont en nombre de trois, parmi eux la plus importante, l'essentiel du contenu que l'utilisateur vient chercher sur internet : l'information. Donc fournir l'information dont l'utilisateur a besoin est l'objectif principal, c'est l'essence même d'une application web.

Deuxièmement la mise en forme des interfaces, qui est tout aussi importante pour améliorer l'attractivité du contenu.

Troisièmement le comportement de l'application en fonction des actions de l'utilisateur, tel que des animations. Etc.

Et donc dans ce chapitre nous allons présenter l'environnement de travail ainsi que les outils nécessaires pour la réalisation de notre application. Par la suite, nous allons décrire la phase d'implémentation en se basant sur quelques interfaces.

I. Présentation de l'environnement de travail

Pour réaliser notre application, nous avons utilisé le langage de développement PHP ainsi que son framework Laravel.

I.1. Laravel

Laravel est un framework open-source créé par Taylor Orwel qui initie une nouvelle façon de concevoir un framework en utilisant ce qui se fait de mieux dans chaque fonctionnalité, Laravel est caractérisé par une architecture MVC et est entièrement basé sur de la programmation orientée objet.

Afin d'utiliser Laravel nous avons besoin d'utiliser des outils de développement et pour pouvoir utiliser ces outils nous avons opté pour l'environnement de développement local Laragon.

Laragon est un serveur de développement web qui permet de faire fonctionner localement des applications web grâce aux outils qu'il possède.

I.1.2. Présentation des outils de développement

I.1.2.1. Git - GitHub

Git est un outil essentiel pour le travail en équipe ou tout aussi bien en solo. Git permet de suivre les modifications et l'organisation d'un projet grâce à son système qui suit l'évolution des fichiers tout en gardant les anciennes versions de chacune, ce qui permet de retrouver les différentes versions d'un fichier.

GitHub est une plateforme spécialement dédiée aux « codes », c'est une sorte de réseau social pour développeur. GitHub permet de partager ou d'utiliser des codes ou bien même de participer à un projet.

I.1.2.2. Laragon

Laragon est un environnement qui comprend plusieurs outils dont Apache et MySQL, ainsi qu'une administration de base de données HeidiSQL et un interpréteur de scripts PHP et enfin un terminal amélioré ainsi que d'autres outils.

I.1.2.3. Apache

Apache est le serveur web qui permet la gestion des requêtes émises par le navigateur et le renvoi de la page web demandée.

I.1.2.4. MySQL

MySQL est le système de gestion de base de données qui permet le stock et la gestion des informations sous forme de données organisées dans des tables, et permet de manipuler ces données grâce à des requêtes SQL.

I.1.2.5. HeidiSQL

HeidiSQL est un système de gestion de base de données qui possède un éditeur SQL et un constructeur de requête. HeidiSQL est utilisé principalement avec le serveur MySQL.

I.1.2.6. PHP

PHP est un langage script qui permet de créer des pages web dynamique à travers desquelles l'utilisateur peut échanger des informations avec le serveur ce qui permet un affichage dynamique (le contenu affiché va dépendre de variables d'un utilisateur à l'autre), les scripts PHP sont contenu dans un fichier qui porte comme extension « .php » et peuvent contenir du HTML, quand le navigateur demande un fichier « .php » le serveur Apache exécute les instructions PHP et produit une page HTML, une fois la page HTML créée, le serveur la renvoie au navigateur pour l'afficher.

I.1.2. Installation

I.1.2.1. Installation de Laravel avec Composer

Pour installer Laravel avec Composer, il faut d'abord installer Composer pour pouvoir utiliser les commandes « composer » dans l'invite de commande pour Windows ou dans le terminal pour Mac OS / Linux.

I.1.2.1.1 Installation de Composer sur MacOS / Linux

Pour installer Composer sur Mac OS ou Linux, il suffit de taper les commandes suivantes dans le terminal.

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '8a6138e2a05a8c28539c9fb361159823655d7ad2c
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

Figure 43. Commande installation Composer MacOS / Linux.

Quand les commandes sont exécutées, « composer.phar » est installé mais seulement dans le répertoire où les commandes ont été exécutées, pour y remédier et pour pouvoir utiliser les commandes « composer » sur n'importe quel répertoire, il faut l'ajouter au PATH, pour y parvenir il faut exécuter la commande suivante :

```
mv composer.phar /usr/local/bin/composer
```

Figure 44. Commande configuration PATH.

I.1.2.1.2 Installation de Composer sur Windows

Pour installer Composer sur Windows il suffit juste de télécharger « Composer-setup.exe » qui se trouve sur le site officiel de Composer, une fois installé, le PATH est automatiquement configuré, ce qui vous permet d'exécuter des commandes composer sur n'importe quel répertoire ou invite de commande.

Maintenant que composer a été installé, il est temps d'installer Laravel avec la commande ci-dessous.

```
composer create-project --prefer-dist laravel/laravel MONERP
```

Figure 45. Commande installation Laravel avec Composer.

I.1.2.2. Installation de Laravel avec Laragon

Une autre solution pour installer Laravel est Laragon, pour ce faire, il faut d'abord au préalable avoir installé Laragon grâce à l'installateur qui se trouve sur le site officiel de Laragon. Une fois installé, il suffit de cliquer sur « crée un site web rapidement » et ensuite sur « Laravel ».

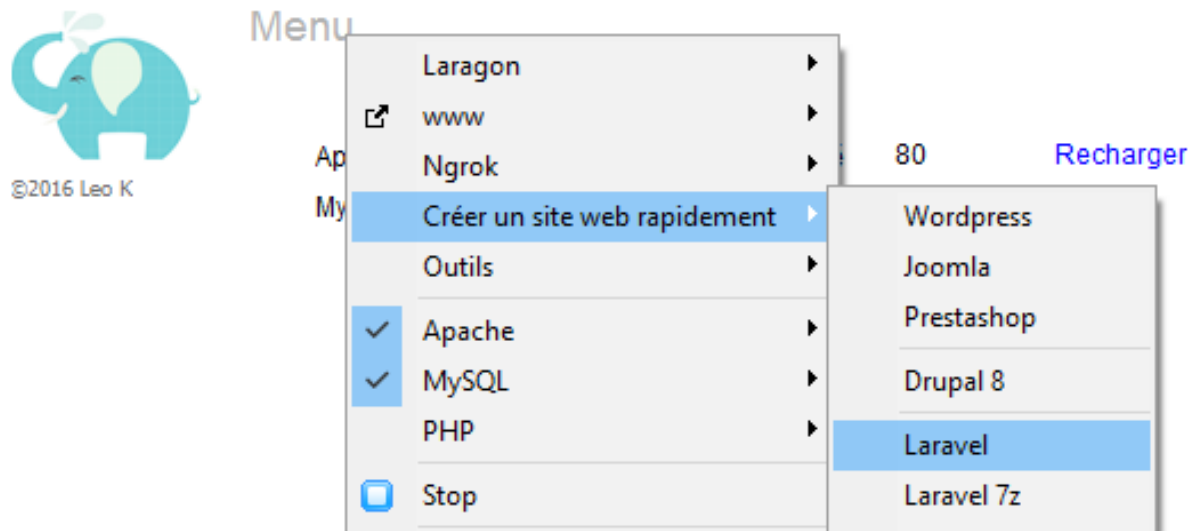


Figure 46. Installation Laravel avec Laragon.

Puis d'entrer le nom du projet et de valider, pour que Laravel soit installé.

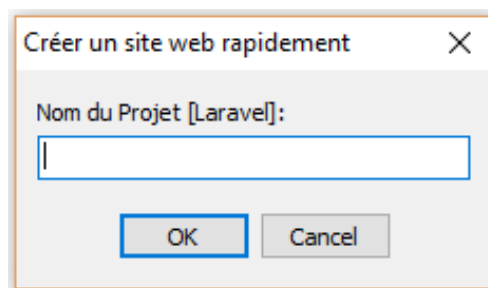


Figure 47. Installation Laravel avec Laragon.

Remarque : Même si l'installation de Laravel a été faite via Composer, l'installation de Laragon est requise pour pouvoir utiliser son environnement de travail.

Remarque 2 : Laragon est uniquement disponible sur Windows, d'autres alternatives s'offrent pour les utilisateurs de Mac OS / Linux, comme Docker, Valet ou Homestead, sans pour autant offrir la possibilité d'installation de Laravel, dans ce cas-là il faudra utiliser Composer.

I.1.2.3. Installation de Git

I.1.2.3.1. Installation de Git sur Linux

Pour installer Git sur Linux, il suffit d'ouvrir le terminal et d'entrer le code ci-dessous si vous êtes sur une distribution Debian, comme Ubuntu.

```
$ apt-get install git-all
```

Figure 48. Commande installation Git sur Debian.

Où le code suivant, si vous êtes sur une distribution Fedora.

```
$ dnf install git-all
```

Figure 49. Commande installation Git sur Fedora.

I.1.2.3.2. Installation de Git sur Mac OS / Windows

Pour Mac OS ou Windows, des installateurs sont disponibles sur le site officiel de Git, il suffit juste de télécharger et d'exécuter le fichier exécutable.

Git étant installé, une des premières choses à faire est de paramétrer le nom de l'utilisateur et l'adresse mail à l'aide de la commande ci-dessous.

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

Figure 50. Commande configuration utilisateur Git.

Maintenant que Git et Laravel sont installés et prêts à l'emploi, il est temps de créer un compte GitHub ce qui est relativement simple puisqu'il faut juste remplir un formulaire.

Le compte créé. Maintenant nous pouvons nous connecter et créer notre repository qui sera le dossier où nous allons garder notre projet, qui est vide pour l'instant.

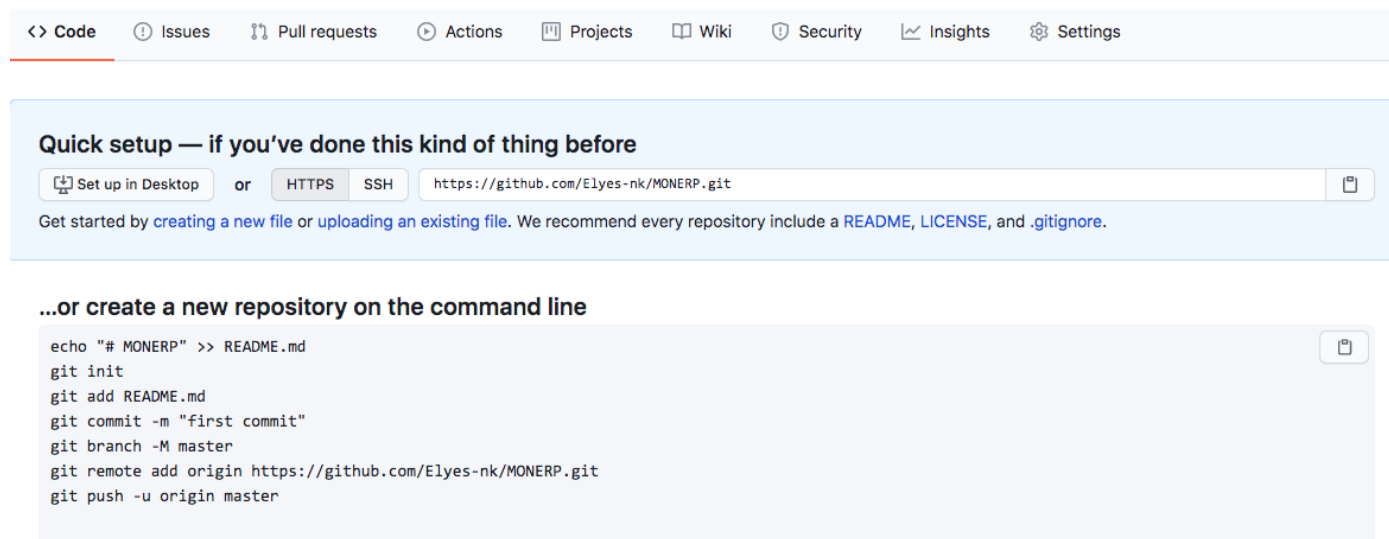


Figure 51. Interface d'un repository vide dans GitHub.

La deuxième étape consiste à y télécharger le projet Laravel que nous avons déjà créé. Pour ce faire, il faut entrer la commande suivante dans l'invite de commande.

```
$ cd\laragon\www\MONERP
```

Figure 52. Commande d'entrer dans le répertoire du projet.

Cette commande permet d'entrer dans le répertoire de notre projet, pour ensuite y exécuter les commandes suivantes.

```
git init
```

Figure 53. Commande initialisation de Git.

```
git add .
```

Figure 54. Commande chargement du projet dans Git.

```
git commit -m "first commit"
```

Figure 55. Commande Git pour créer le commit.

Ainsi que les commandes qui se trouvent dans le repository que nous avons créé.

```
git remote add origin https://github.com/Elyes-nk/MONERP.git  
git branch -M master  
git push -u origin master
```

Figure 56. Commande Git pour envoyer le commit à l'adresse GitHub.

Lorsque les commandes sont exécutées. GitHub demande de se connecter sur le compte où se trouve le repository. Lorsque l'authentification est faite, le téléchargement est finalisé, et nous pourrons accéder à notre projet depuis GitHub.

Ainsi, notre projet Laravel a été téléchargé dans GitHub, nous pouvons dès à présent commencer à développer notre application, mais avant cela nous allons d'abord étudier l'organisation et l'architecture de Laravel.

I.1.3. Organisation de Laravel [27]

Laravel est doté d'une architecture qui est composée de plusieurs dossiers, le premier et l'un des plus importants le dossier « app » qui contient des sous dossiers avec chacun ayant un rôle bien précis :

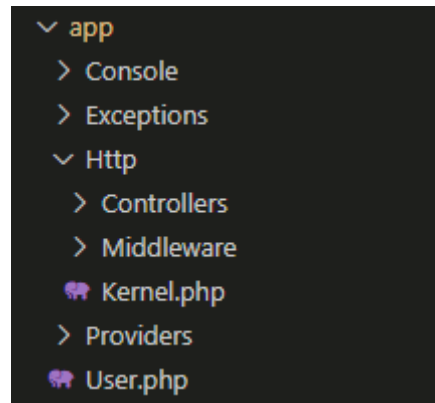


Figure 57. Répertoire Laravel.

- Console : contient toutes les commandes en mode console.
- Http : contient tous les fichiers qui concernent la communication telle que les middlewares qui sont utilisés pour filtrer les requêtes Http, ce dossier contient aussi les contrôleurs détaillés dans le prochain point.
- Providers : contient tous les fournisseurs de services, qui servent à initialiser les composants.
- User : un modèle par défaut pour les utilisateurs.

D'autres dossiers tout aussi importants :

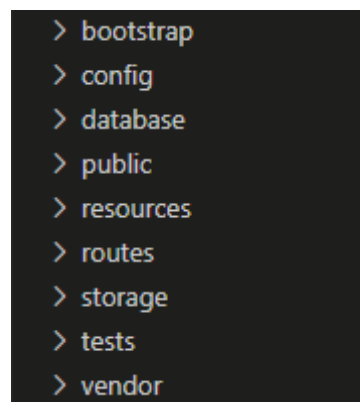


Figure 58. Répertoire Laravel.

- Bootstrap : à ne pas confondre avec le framework CSS, ce dossier contient tous les scripts d'initialisation de Laravel pour le chargement des classes, de l'environnement pour le démarrage de l'application.
- Public : contient tout type de fichiers (images, CSS) qui doivent apparaître en public sur le site.
- Config : contient toutes les configurations de l'application, de l'authentification, du cache, de la base de données.
- Database : contient toutes les migrations détaillées dans le prochain point.
- Ressources : contient les vues, fichiers de langage.
- Routes : contient les fichiers qui servent à la gestion des url.

- Storage : contient des données en caches de l'application : vues compilées, clé de session. Etc.
- Tests : contient des fichiers de tests.
- Vendor : contient tous les composants de Laravel et de ses dépendances.

I.2. MVC

L'architecture de Laravel se base sur une architecture MVC (modèle – vue – contrôleur) qui est un patron utilisé dans le développement d'applications web, cette architecture permet de concevoir des applications claires et efficaces grâce à la séparation des tâches.

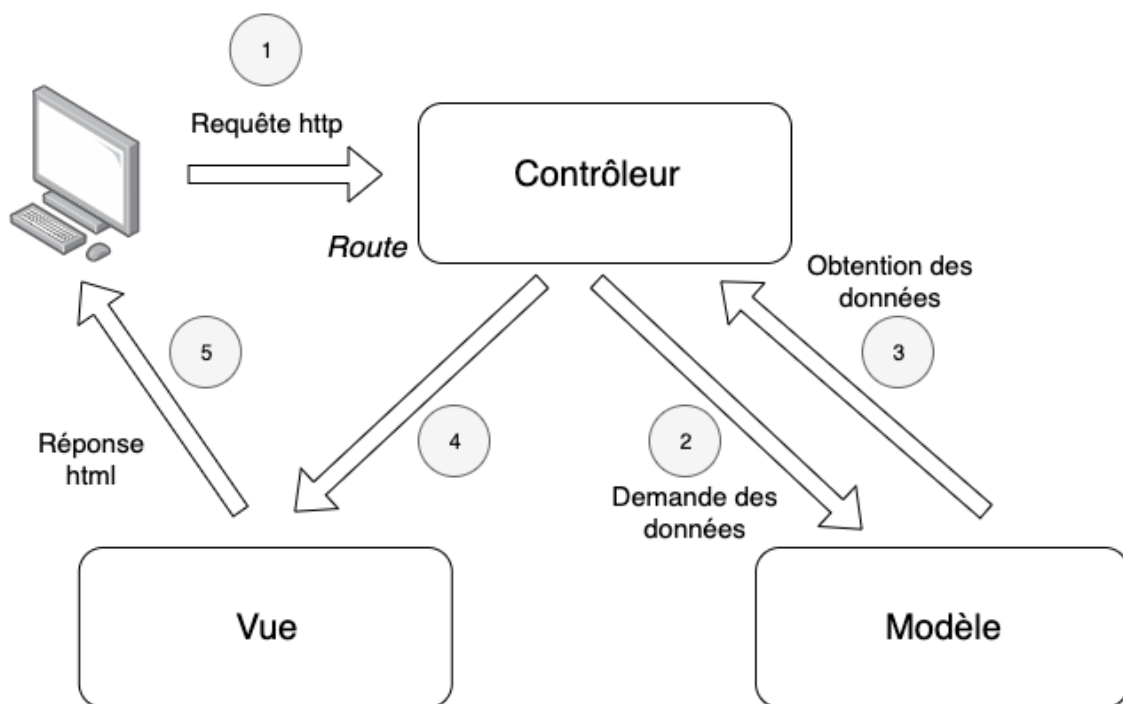


Figure 59. Architecture MVC.

I.2.1. Les routes

Les routes sont indispensables dans la gestion des requêtes HTTP, ce sont les routes qui font le lien entre les vues, les contrôleurs et les modèles.

Les principales requêtes utilisées dans les routes sont :

URI	Name	Action	Before Filters
GET /users	users.index	UsersController@index	auth, checkPermissions
GET /users/create	users.create	UsersController@create	auth, checkPermissions
POST /users	users.store	UsersController@store	auth, checkPermissions
GET /users/{users}	users.show	UsersController@show	auth, checkPermissions
GET /users/{users}/edit	users.edit	UsersController@edit	auth, checkPermissions
PUT /users/{users}	users.update	UsersController@update	auth, checkPermissions
PATCH /users/{users}		UsersController@update	auth, checkPermissions
DELETE /users/{users}	users.destroy	UsersController@destroy	auth, checkPermissions
GET /users/{users}/destroy	users.delete	UsersController@delete	auth, checkPermissions

Figure 60. Méthodes de routes.

- GET : demande une ressource qui ne change pas, généralement utilisé pour demander l’affichage d’un formulaire ou d’une page.
- POST : utilisé pour soumettre des données et les ajouter comme nouvelle ressource dans la base de données.
- PUT/PATCH : utilisé pour la mise à jour d’une ressource.
- DELETE : utilisé pour supprimer une ressource.

I.2.2. Contrôleurs

Les contrôleurs gèrent l’ensemble des interactions, d’une manière générale, ils récupèrent les données des modèles, les traites et les envois aux vues pour être affiché.

Exemple : nous allons créer un contrôleur pour les articles que nous allons appeler « ProductsController », pour cela il suffit d’entrer la commande suivante dans l’invite de commande de Laragon.

```
C:\laragon\www\MONERPs (master)
λ php artisan make:controller ProductsController -r
```

Figure 61. Commande création contrôleurs.

Maintenant que le contrôleur a été créé et se trouve bel et bien dans « app/Http/Controllers »,

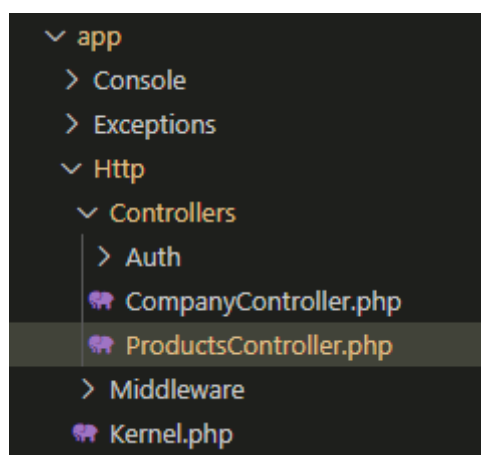


Figure 62. Répertoire contrôleurs.

Il suffit de créer les principales fonctions du contrôleur :

```
<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class ProductsController extends Controller
{
    //fonction pour afficher une page
    public function index()
    {
        /*script */
    }
    //fonction pour afficher un formulaire de création de ressource
    public function create()
    {
        /*script */
    }
    //fonction pour sauvegardé une ressource
    public function store(Request $request)
    {
        /*script */
    }
    //fonction pour afficher une ressource
    public function show($id)
    {
        /*script */
    }
    //fonction afficher un formulaire de modification de ressource
    public function edit($id)
    {
        /*script */
    }
    //fonction pour la mise à jour d'une ressource
    public function update(Request $request, $id)
    {
        /*script */
    }
    //fonction pour supprimer une ressource
    public function destroy($id)
    {
        /*script */
    }
}
```

Figure 63. Fichier d'un contrôleur.

Remarque : Il est impératif que dans chaque contrôleur les modèles utilisés soient appelés.

I.2.3. Modèles et migrations

Les modèles permettent d'accéder aux données de l'application stockée dans une base de données.

Les migrations permettent de mettre en place un schéma de base de données, c'est-à-dire, de créer des tables et de maintenir les tables grâce à un suivi des modifications.

Pour créer un modèle il faut entrer la commande suivante dans le terminal de Laragon.

```
C:\laragon\www\MONERPs (master)
λ php artisan make:model Product -m
```

Figure 64. Commande création modèle et migration.

Remarque : « -m » à la fin de la commande permet de créer la migration qui est conjointe au modèle « Product ».

Ci-dessous nous pouvons constater que le modèle et la migration ont bien été créés, le modèle dans le dossier « app » et la migration dans « database/migrations ».

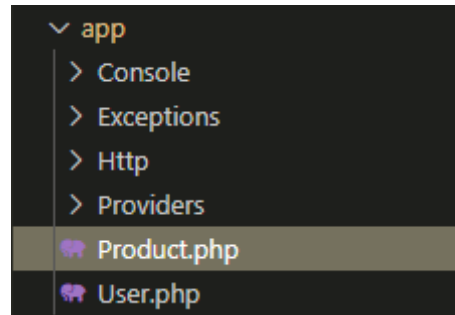


Figure 65. Répertoire modèles.

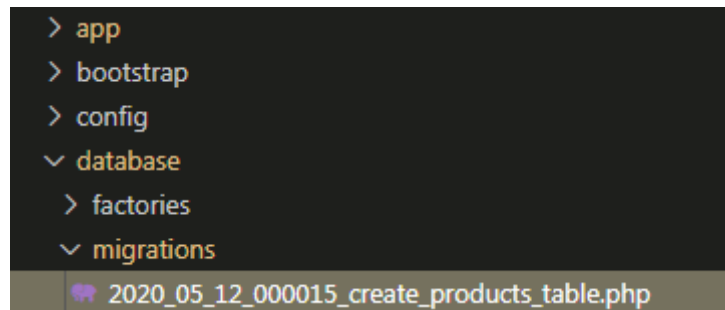


Figure 66. Répertoire migrations.

Le modèle créé aura un code similaire :

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    //sécurité de Laravel lors de l'utilisation de tables en PHP
    //si n'est pas utilisé généré automatiquement une erreur de type "MassAssignmentException"
    protected $guarded = [];

    //fonctions qui retourne d'autres modèles en cas de relations avec celles-ci
    public function taxe()
    {
        return $this->belongsTo('App\Taxe');
    }
    public function warehouse()
    {
        return $this->belongsTo('App\Warehouse');
    }
    public function category()
    {
        return $this->belongsTo('App\CategoryProduct');
    }
}
```

Figure 67. Fichier du modèle « Product ».

Remarque : sachant qu'un modèle et une migration font référence à une seule table, et donc il est impératif d'implémenter une fonction pour chaque relation avec une autre table.

La migration créée aura un code similaire :

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateProductsTable extends Migration
{
    public function up()
    {
        //fonction de creation de la table produits avec les colonnes ci dessous
        Schema::create('products', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('ref');
            $table->float('sale_price');
            $table->float('alerte_stock');
            $table->float('optimal_stock');
            $table->float('physical_stock')->nullable();
            $table->float('virtual_stock')->nullable();
            $table->string('procurement_method');
            $table->float('standard_price');
            $table->float('cump')->nullable();
            $table->string('type');
            $table->foreignId('company_id')->constrained();
            $table->foreignId('user_id')->constrained();
            $table->foreignId('taxe_id')->nullable();
            $table->foreignId('warehouse_id')->nullable();
            $table->foreignId('category_product_id')->nullable();
            $table->foreignId('unity_id')->nullable()->references('id')->on('product_unities');
            $table->timestamps();
        });
    }
}

```

Figure 68. Fichier d'une migration.

```

// fonction de suppression de la table produits
public function down()
{
    Schema::disableForeignKeyConstraints();
    Schema::dropIfExists('products');
}
}

```

Figure 69. Fichier de la migration du modèle « Product ».

Remarque : les colonnes doivent être implémentées exactement comme l'était décrit dans la partie base de données dans le chapitre analyse et conception dans la fonction up.

Remarque 2 : les fonctions up et down font référence aux fonctions de création et de suppression de la table.

Maintenant que la migration a été créée et complétée avec les différentes colonnes, il faut enregistrer la table dans la base de données, pour ce faire, il faut entrer la commande suivante dans le terminal :

```
C:\laragon\www\MONERPs (master)
λ php artisan migrate
```

Figure 70. Commande pour la migration.

Lorsque la commande a été exécutée la table « Products » est créée :

#	Nom	Type de données	Taille/Ensem...	Non signé	NULL autor...	ZERO...	Par défaut	Commentaire	Classe
1	id	BIGINT	20	☒	☐	☐	AUTO_INCRE...		
2	name	VARCHAR	255	☐	☐	☐	Pas de défaut		ut
3	ref	VARCHAR	255	☐	☐	☐	Pas de défaut		ut
4	sale_price	DOUBLE	8,2	☐	☐	☐	Pas de défaut		
5	alerte_stock	DOUBLE	8,2	☐	☐	☐	Pas de défaut		
6	optimal_stock	DOUBLE	8,2	☐	☐	☐	Pas de défaut		
7	physical_stock	DOUBLE	8,2	☐	☒	☐	NULL		
8	virtual_stock	DOUBLE	8,2	☐	☒	☐	NULL		
9	procurement_method	VARCHAR	255	☐	☐	☐	Pas de défaut		ut
10	standard_price	DOUBLE	8,2	☐	☐	☐	Pas de défaut		
11	cump	DOUBLE	8,2	☐	☒	☐	NULL		
12	type	VARCHAR	255	☐	☐	☐	Pas de défaut		ut
13	taxe_id	BIGINT	20	☒	☒	☐	NULL		
14	warehouse_id	BIGINT	20	☒	☒	☐	NULL		
15	category_product_id	BIGINT	20	☒	☒	☐	NULL		
16	unity_id	BIGINT	20	☒	☒	☐	NULL		
17	created at	TIMESTAMP		☐	☒	☐	NULL		

Figure 71. Table « Products » dans la base de données.

Le processus est répété jusqu'à la création de toutes les tables de la base de données que nous avons détaillée lors du chapitre analyse et conception.

I.2.4. Les vues

Une vue est une interface graphique qui affiche le résultat de la requête grâce à de l'HTML et du CSS. La vue a comme extension « blade.php ».

- **HTML** : « Hypertext Markup Language » est le langage de balisage qui a été créé par la World Wide Web Consortium et le Web Hypertext Application Technology Working Group, l'HTML est ce qui permet de créer et de gérer la manière dont un site web est présenté grâce à son système de balisage.
- **CSS** : « Cascading StyleSheets » crée 5 ans après l'HTML dans le but de compléter ce dernier en mettant en forme les différents contenus de l'HTML en leur appliquant des styles.
- **Blade** : est le moteur de template qui est utilisé par Laravel. Blade est simple d'utilisation mais néanmoins très puissant car il permet d'éliminer les parties redondantes de chaque vue, Blade permet aussi d'écrire du script PHP avec une

syntaxe plus simple que celle du PHP brute, tout ceci permet d'améliorer la lisibilité et la maintenabilité.

Exemple : nous allons créer une vue qui va gérer les articles, pour ce faire il faut créer un dossier « products » dans le dossier « ressource » et crée un fichier avec l'extension « blade.php » pour chaque fonction définie dans le contrôleur associé, comme ceci :

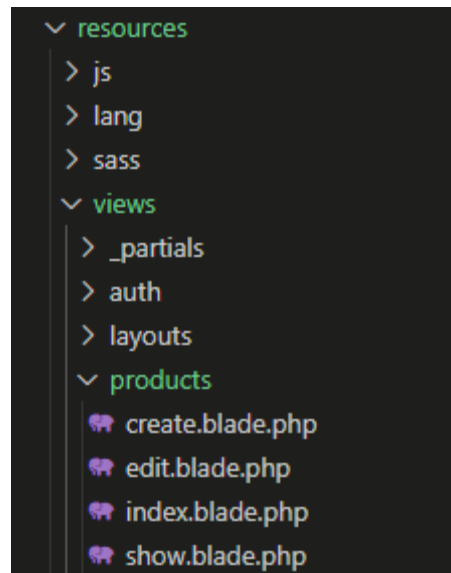


Figure 72. Répertoire vue « products ».

I.3. Livewire [27]

Livewire est un framework full-stack développé par Caleb Porzi. Livewire permet de créer des composants dynamiques avec Laravel sans avoir recours au JavaScript, ce qui permet d'éviter l'installation d'un framework JavaScript ou de transférer les données de PHP à JavaScript.

Ses atouts sont le fait que ces scripts ressemblent beaucoup à Laravel ce qui facilite beaucoup de choses lors du développement, son autre atout est que son fonctionnement est relativement simple, en effet, quand une interaction se produit, Livewire envoie une requête AJAX au serveur avec les nouvelles données, le serveur réaffiche les composants avec l'HTML remis à jour.

II. Présentation de l'application

Dans ce qui suit nous allons présenter le rendu final de notre application ainsi que les interfaces qui la composent.

II.1. Espace administrateur

Si l'utilisateur a un rôle « administrateur » alors cette interface s'affiche, cette interface a pour particularité d'avoir un contrôle total sur l'application.

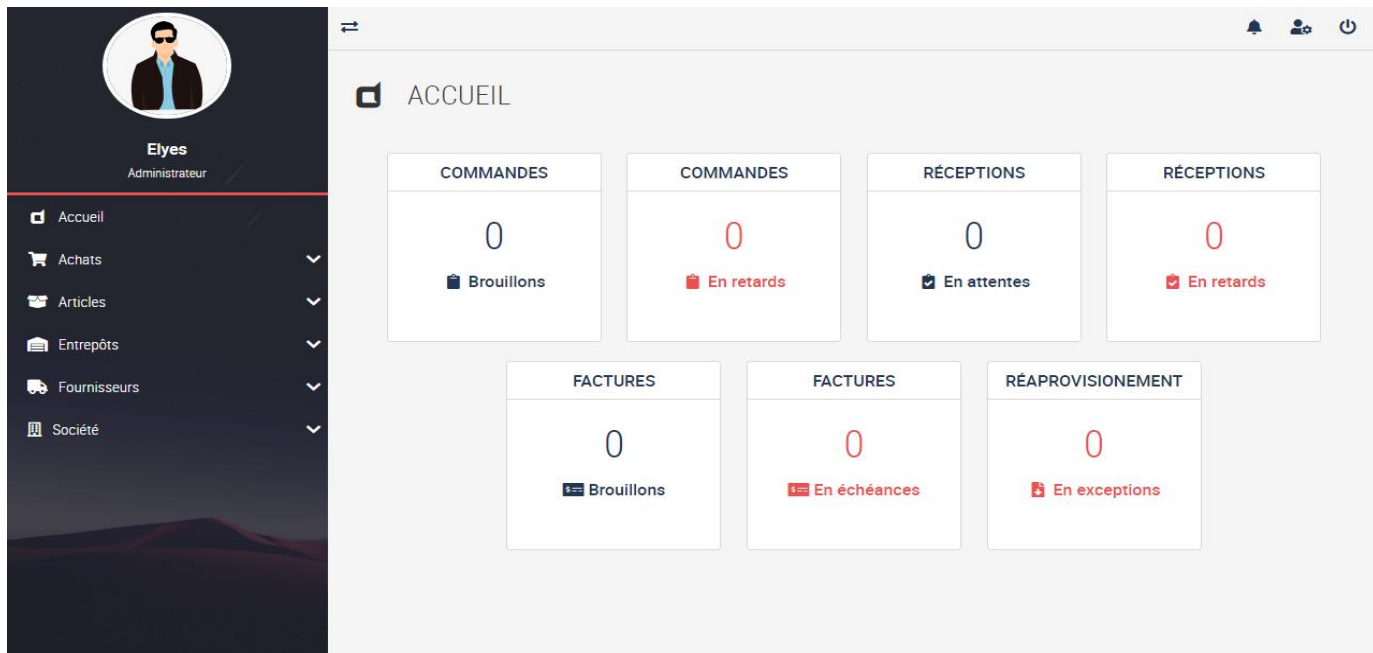


Figure 73. Espace administrateur.

II.2. Espace financier

Si l'utilisateur a un rôle « financier » alors l'interface ci-dessous s'affiche, cette interface a des fonctionnalités limitées par rapport à l'interface administrateur, ces fonctionnalités ont été définies dans le chapitre III analyse et conception.

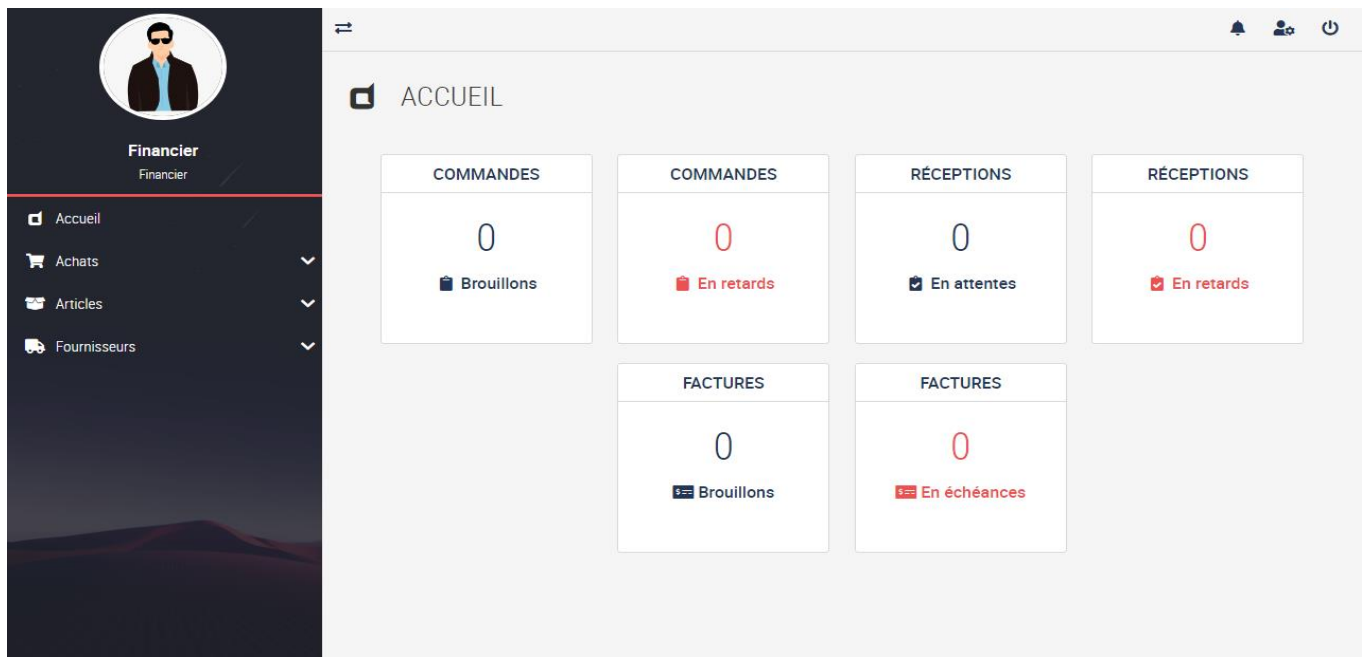


Figure 74. Espace financier.

II.3. Espace magasinier

Si l'utilisateur a un rôle « magasinier » alors cette interface s'affiche, comme pour l'interface « financier » cette interface a des fonctionnalités limitées par rapport à l'interface « administrateur » mais différentes de celle de « financier », ces fonctionnalités ont été définies dans le chapitre III analyse et conception.

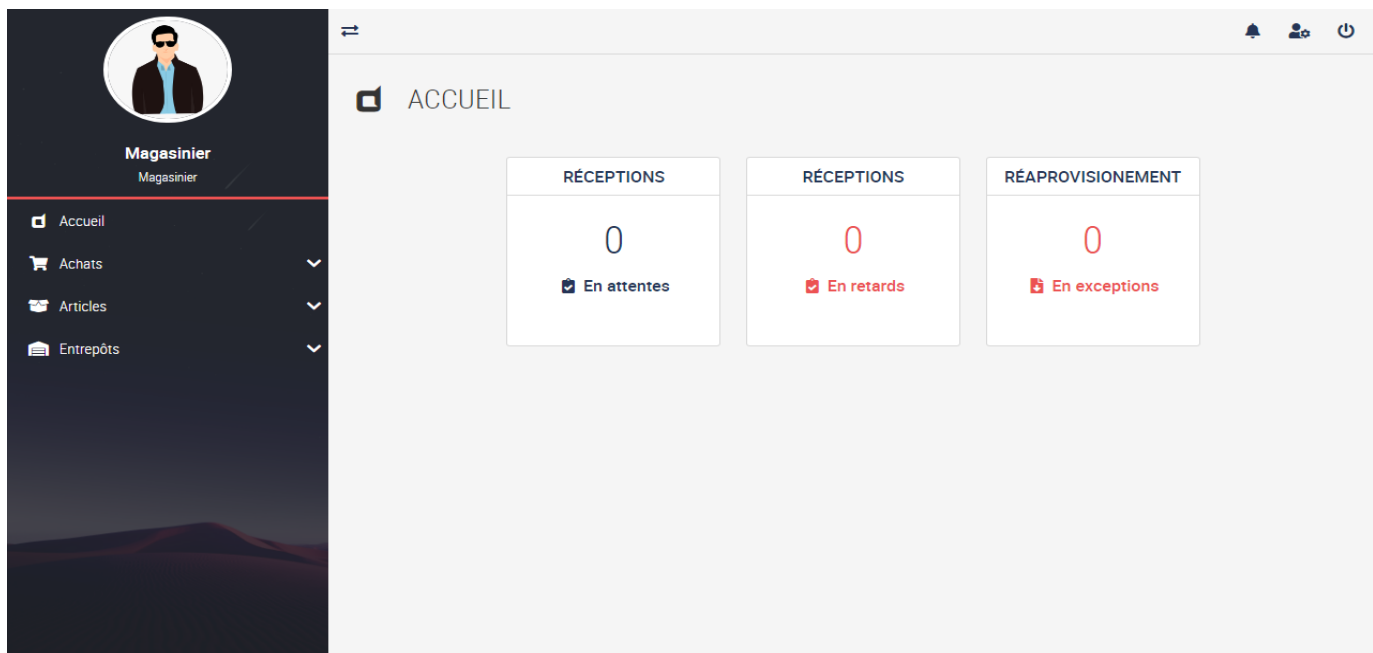


Figure 75. Espace magasinier.

II.4. Interfaces de l'application

Lorsque l'utilisateur authentifié est entré dans son espace il peut accéder aux fonctionnalités qui lui sont accordées.

Pour qu'un utilisateur puisse accéder à certaines fonctionnalités tel que l'ajout de produit ou la création de devis ou de commandes, la présence de certains paramètres est obligatoire, par exemple pour créer un produit il est obligatoire d'avoir une unité de produit et une catégorie de produit déjà créée.

II.4.1. Gestion des paramètres

Exemple : un administrateur veut créer une unité de produit, tout d'abord il doit se rendre vers l'interface « unités de mesure » en cliquant sur le bouton de menu à gauche, comme ci-dessous.

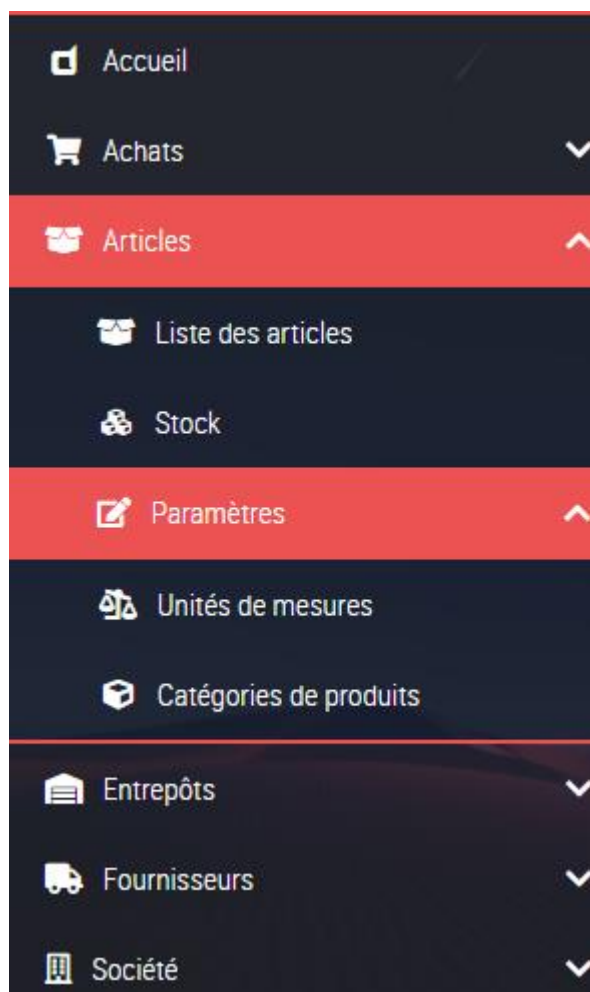


Figure 76. Répertoire unités de mesure.

Puis l'index de la page devis s'affiche avec la liste des devis déjà enregistrés, comme nous n'avons pas créé une unité précédemment il nous affiche une liste vide.

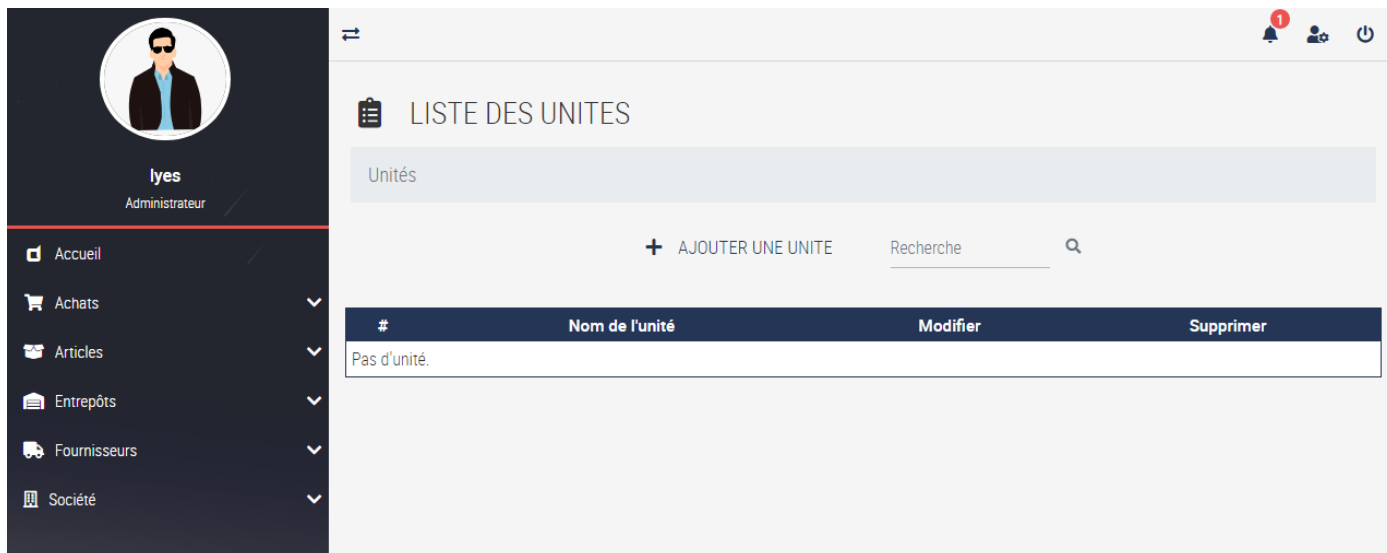


Figure 77. Interface unités de mesure.

Pour créer un devis l'administrateur doit appuyer sur le bouton « créer une unité », et l'application le redirigera vers l'interface « ajouter une unité ».

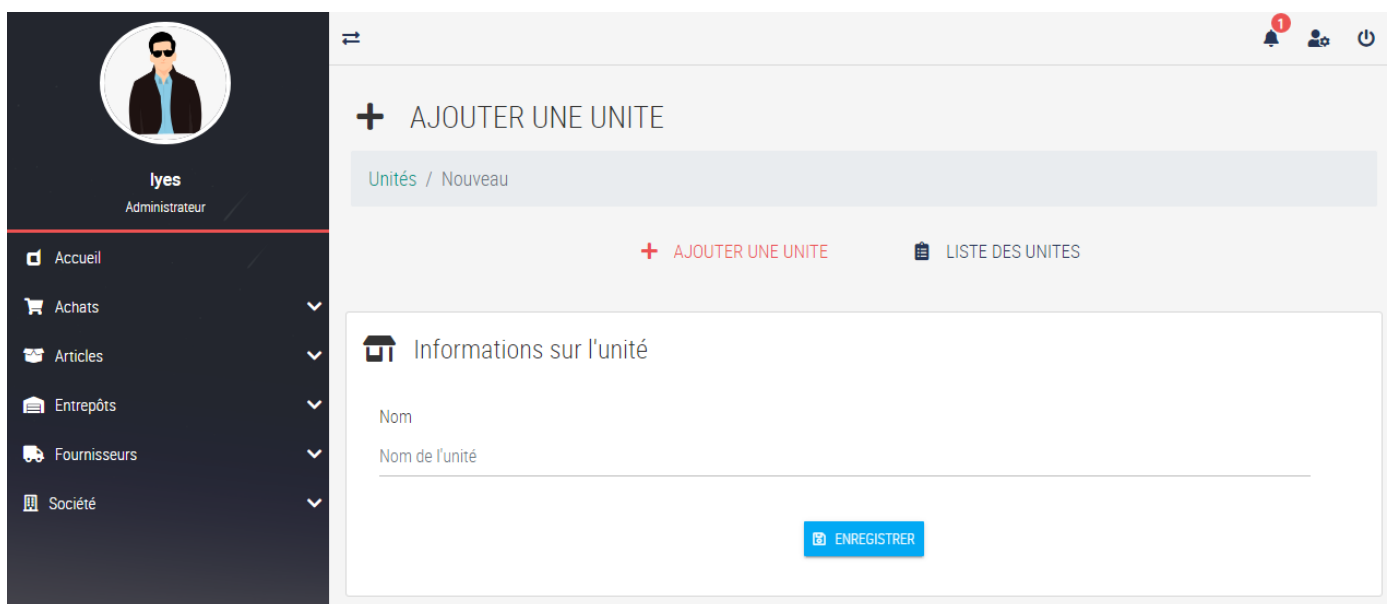


Figure 78. Interface ajouter une unité de mesure.

Où il suffira de remplir le formulaire affiché et de cliquer sur le bouton « enregistrer » pour enregistrer l'unité.

Maintenant que l'unité a été enregistrée, l'application redirige l'administrateur vers l'interface « afficher une unité » tout en affichant que l'unité a bel et bien été enregistrée.

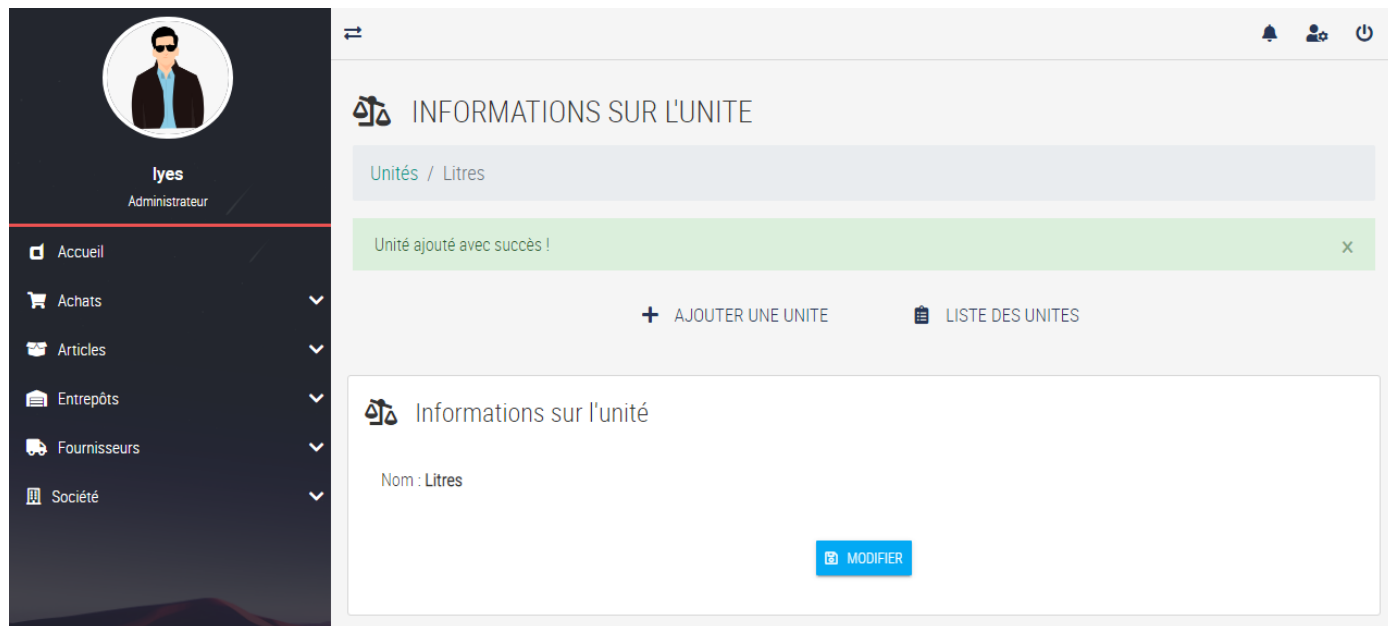


Figure 79. Interface afficher une unité de mesure.

Exemple 2 : l'administrateur veut mettre à jour l'unité qu'il a créée, il lui suffit d'aller sur l'interface « unités de mesure » et de cliquer sur le bouton « modifier » de l'unité qu'il souhaite modifier, ainsi l'interface « modifier une unité » s'affiche comme ci-dessous.

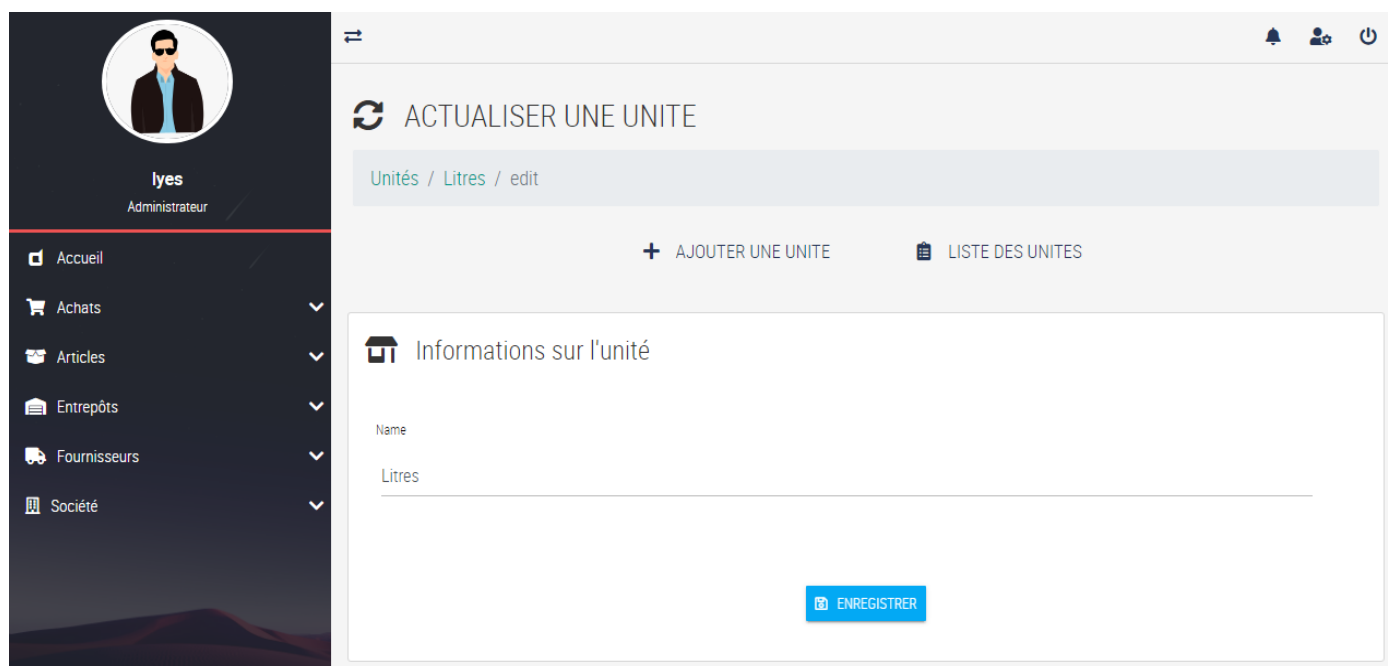


Figure 80. Interface modifier une unité de mesure.

Il reste plus qu'à l'administrateur de modifier les champs qu'il souhaite et de cliquer sur le bouton « Enregistrer » pour enregistrer les modifications, quand cela est fait, l'application le redirigera vers l'interface « afficher une unité » tout en affichant que l'unité a bel et bien été modifiée.

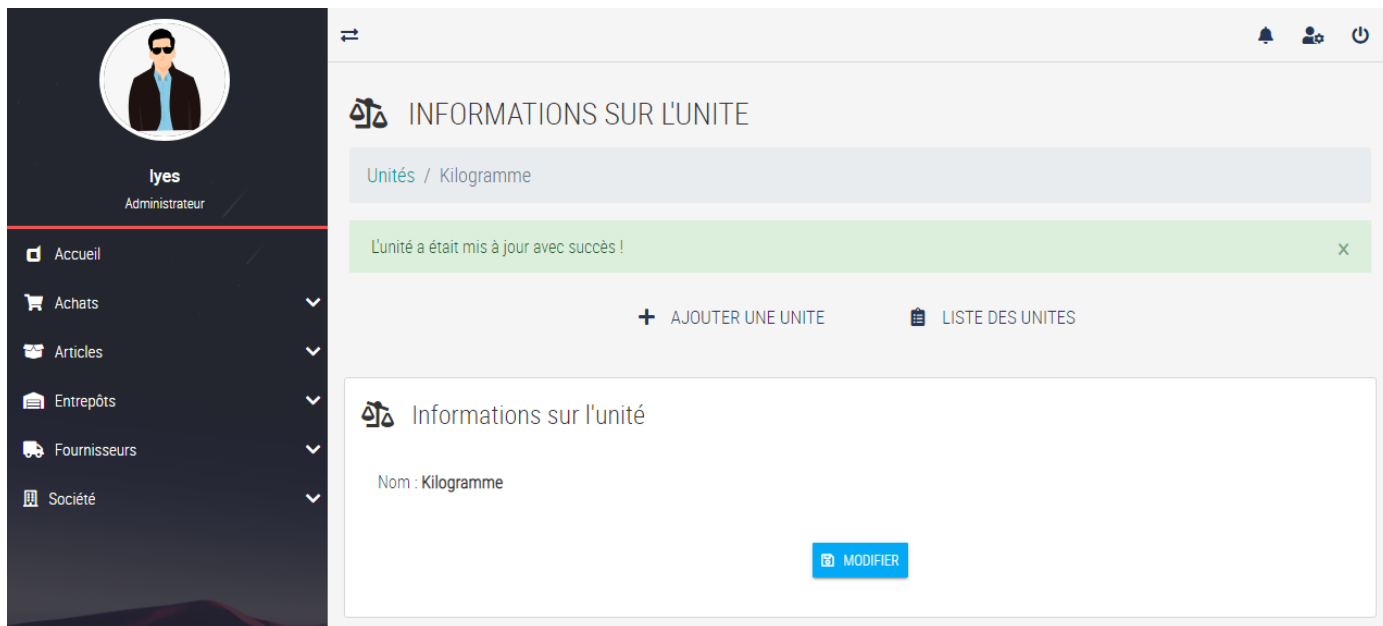


Figure 81. Interface afficher une unité de mesure.

Exemple 3 : l'administrateur veut supprimer l'unité qu'il a créée, il lui suffit d'aller sur l'interface « unités de mesure » et de cliquer sur le bouton « supprimer » de l'unité qu'il souhaite supprimer, ainsi une demande de confirmation de suppression s'affiche comme ci-dessous.

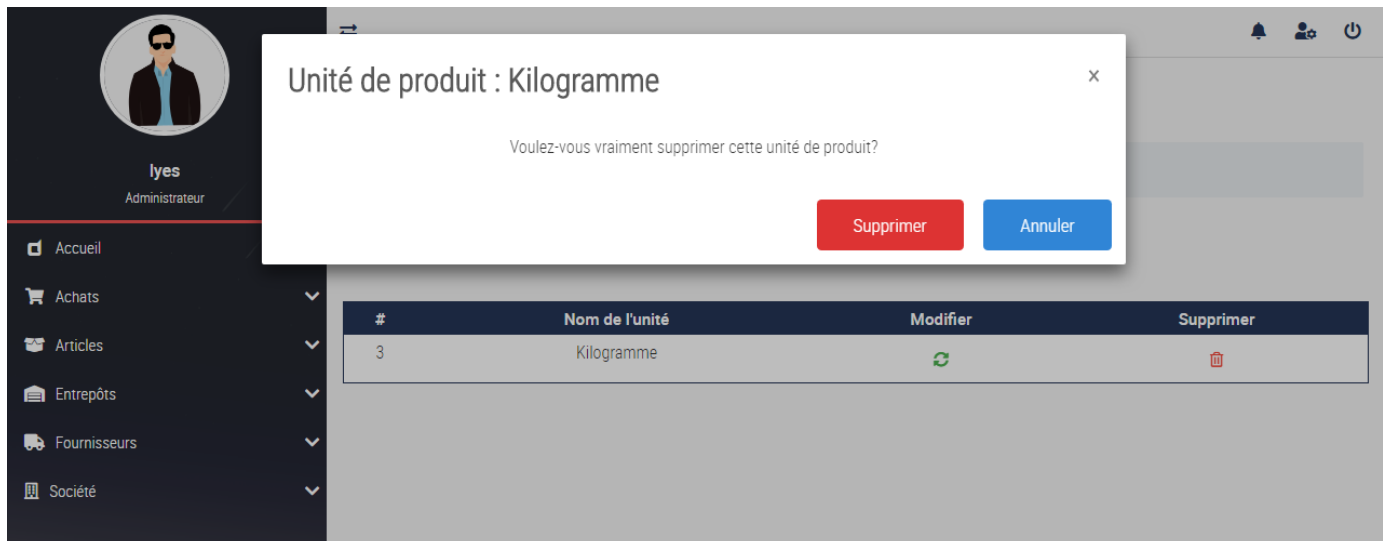


Figure 82. Interface unités de mesure.

Il suffira de confirmer pour que l'unité soit supprimée, quand cela est fait, l'application affichera que l'unité a été supprimée.

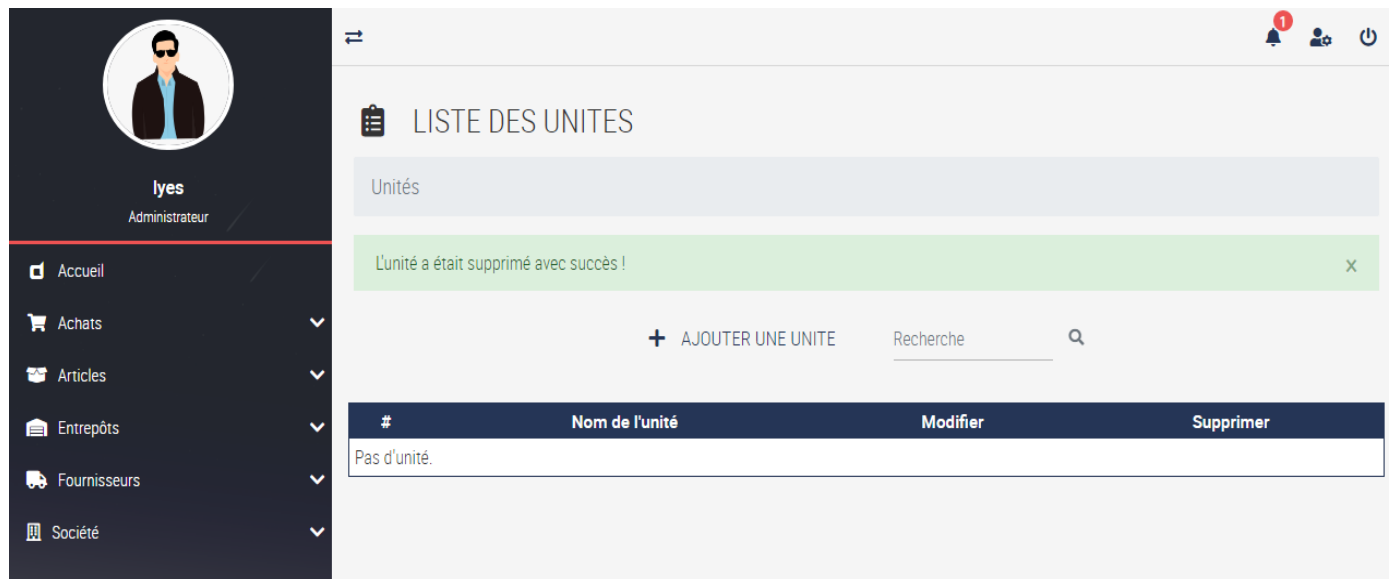


Figure 83. Interface unités de mesure.

II.4.2. Articles

Dans un ERP l'article est considéré comme étant le noyau, une bonne configuration des articles est donc nécessaire pour assurer une bonne gestion de stocks. L'article peut désigner aussi bien un objet matériel qu'un service, et peut être créé par un utilisateur qui peut être soit un administrateur ou soit un magasinier depuis l'interface « crée un article ».

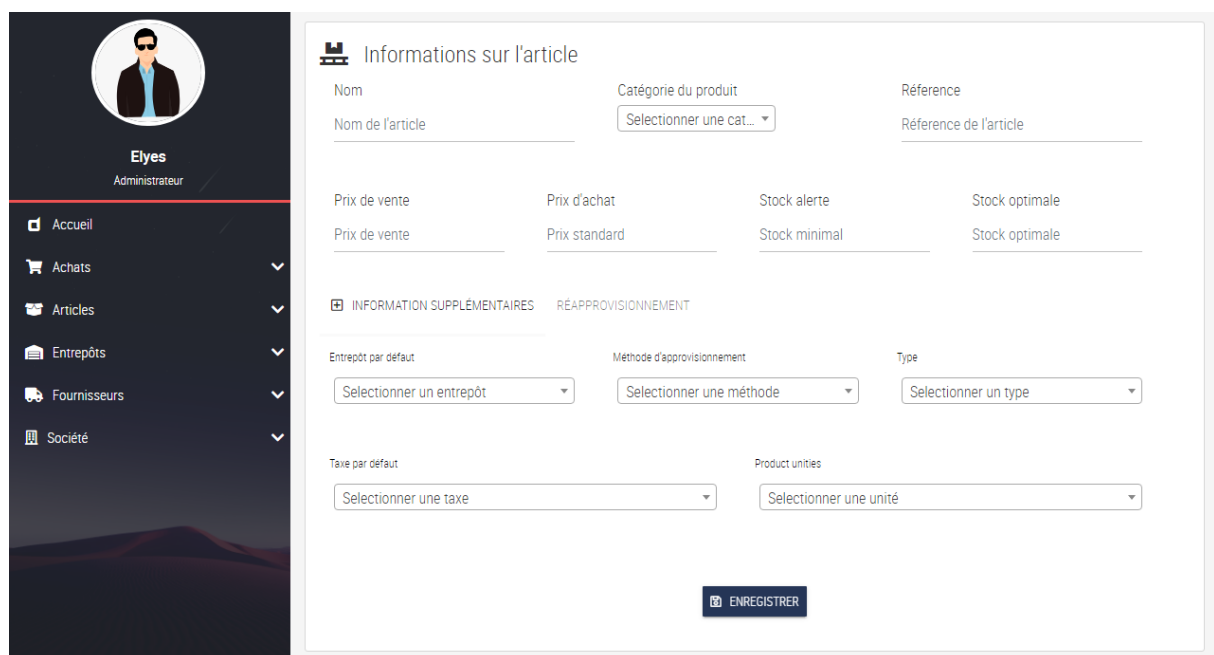


Figure 84. Interface ajouter un produit.

Lors de la création d'un article l'utilisateur doit spécifier la méthode de réapprovisionnement de l'article en question, qui peut être soit « Planifié » soit « Automatique ». La méthode « Planifié » laisse le choix aux utilisateurs quant à la quantité d'articles à réapprovisionner et

à quand celle-ci aura lieu, la configuration de cette planification est faite dans l'interface « Planification de réapprovisionnement »

Figure 85. Interface configurer un réapprovisionnement.

Par contre pour la méthode « Automatique », à chaque fin de journée le système vérifie si la quantité de chaque article ayant cette méthode d'approvisionnement dans le stock physique et le stock virtuel est inférieure au stock d'alerte de ce dernier, une demande de réapprovisionnement sera lancée automatiquement pour cet article, qui par la suite sera confirmé par un administrateur ou un financier.

Pour ce qui est de l'onglet réapprovisionnement, l'utilisateur peut y rentrer les données du fournisseur de l'article créé.

Figure 86. Interface ajouter un produit.

II.4.3. Devis

Un devis est un document qui définit les modalités de la prestation commerciale entre un vendeur et un acheteur, le devis peut soit être généré automatiquement grâce à un réapprovisionnement, ou être créé manuellement depuis l'interface « crée un devis ». Dans les deux cas un devis comme ci-dessous sera créé.

Devis / Achat/002

CONFIRMER L'ACHAT BROUILLON

Nom : Achat/002 Date : 2020-09-30 Condition de règlement : 30 Jours

Fournisseur : FOURNISSEUR 1 Liste de prix : Liste de prix (dzd) date prévue de livraison : 2020-10-20

Article	Code	Unité	Quantité	Prix	Taxe	Entrepot de destination	Total
ARTICLE 1	ART1	Litres	1	100	TVA	Tizi ousou	100

Total hors taxe : 100 dzd
 Total taxes : 12 dzd
 Total remises : 2 dzd
 Total : 110 dzd

MODIFIER IMPRIMER

Figure 87. Interface afficher un devis.

II.4.4. Bon de commande

Un bon de commande est la concrétisation du devis et constitue l'accord entre les deux parties, pour le générer il suffit d'accéder au devis voulu et de cliquer sur le bouton « confirmer l'achat ». Le devis sera confirmé et transformé en bon de commande confirmé comme ci-dessous.

Commandes / Achat/002

LISTE DES COMMANDES

RÉCEPTION DES ARTICLES VOIR FACTURE ANNULER L'ACHAT VOIR LES RÉCEPTIONS CONFIRMER

Nom : Achat/002 Date : 2020-09-30

Fournisseur : FOURNISSEUR 1 Liste de prix : Liste de prix (dzd) Date prévue de livraison : 2020-10-20

Article	Code	Unité	Quantité	Prix	Taxe	Entrepôt de destination	Total
ARTICLE 1	ART1	Litres	1	100	1	Tizi ouzou	100

Total hors taxe : 100 dzd
Total taxes : 12 dzd
Total remises : 2 dzd
Total : 110 dzd

IMPRIMER

Figure 88. Interface afficher un bon de commande.

II.4.5. Facture

Une facture en état « Brouillon » est automatiquement générée lors de la confirmation du devis, cette dernière sera en attente de validation. Puis bascule vers l'état « Ouverte » lorsque celle-ci est validée. La facture sera en attente de règlement jusqu'à sa date d'échéance, si dépassé ce délai la facture passera automatiquement à un état « En échéance ».

Factures / Facture/002

LISTE DES FACTURES

VALIDÉ LA FACTURE ANNULER LA FACTURE BROUILLON

Numéro facture : Facture/002 Date facture: 2020-09-30 Document d'origine: Achat/002

Fournisseur : FOURNISSEUR 1 Liste de prix : Liste de prix (dzd) date d'échéance : 2020-10-30

Article	Code	Prix	Quantité	Taxe	Total
ARTICLE 1	ART1	100	1	TVA	100

Total hors taxe : 100 dzd
Total taxes : 12 dzd
Total remises : 2 dzd
Total : 110 dzd

IMPRIMER

Figure 89. Interface afficher une facture.

II.4.6. Bon de réception

De la même manière un bon de réception est automatiquement généré après la confirmation d'un devis. Celui-ci sera en état « Prêt à recevoir » jusqu'à la réception physique de la totalité de la commande, après cela l'état passera en état « Reçu ».

Figure 90. Interface afficher une réception.

II.4.7. Bon de retour

Le bon de retour peut seulement être générée lorsque la commande a été réceptionnée. Le bon de retour correspond à un retour d'articles dans l'éventualité où un produit ou plusieurs produits ne répondent pas aux qualités attendues.

Figure 91. Interface afficher un bon de retour.

II.4.8. Stock

Le stock est la quantité de marchandise en réserve. Le stock d'un produit donné est quant à lui générer après réception du produit.



Produit	Stock virtuel	Stock physique	Stock alerte	Stock optimal	Cump	Valorisation de stock
ARTICLE 1	2	3	100	200	100	500

Figure 92. Interface stock.

Conclusion

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement, les outils et les langages utilisés pour développer notre système ainsi que leurs installations. Puis nous avons présenté certaines interfaces de notre application ainsi que leurs comportements pour clarifier les étapes de son utilisation. Tout en respectant la conception élaborée lors de l'analyse nous avons pu atteindre les objectifs fixés.

Conclusion générale

L'objectif de notre projet était de concevoir et d'implémenter une plateforme qui fournit une solution web aux entreprises, qui leur permettrait un suivi en temps réel, et une performance accrue grâce à une meilleure maîtrise des mouvements de stock de produits.

Tout au long de ce mémoire nous avons présenté les différentes phases de la réalisation de notre application. Nous avons d'abord commencé par décrire l'environnement des entreprises et de l'apport des ERP sur celles-ci, puis nous avons détaillé le module de gestion de stock et l'approvisionnement. Par la suite nous avons décrit les besoins des futurs utilisateurs en utilisant UML. Ce qui a mené à une analyse approfondie, puis à une conception des fonctionnalités de l'application que nous avons implémentées lors de la réalisation.

Cette dernière est basée sur des technologies fiables et puissantes du web. Le front-end est géré par Livewire qui est un nouveau framework qui ne cesse d'évoluer. Le back-end quant à lui, est implémenté avec Laravel qui offre une architecture qui simplifie grandement l'ajout d'éventuelles nouvelles fonctionnalités, qui viendront s'ajouter à celles déjà élaborées, qui sont la gestion des paramètres de l'entreprise, la gestion des produits et fournisseurs ainsi que la gestion des achats au travers des devis et des factures ainsi que les bons de commande, réception et retour.

Malgré que nous ayons été limité par certaines contraintes, qui étaient un obstacle à l'amélioration de l'application et que nous ayons utilisé des technologies éprouvées. Réaliser une interface simple et ergonomique et des fonctionnalités pratiques restent complexes car nécessite un feedback des utilisateurs régulier.

Cependant, ce projet a de fortes chances d'être amélioré avec l'intégration d'autres modules pour couvrir l'ensemble des besoins de l'entreprise. Ainsi que l'ajout de certaines fonctionnalités, comme l'ajout d'un dossier d'importation ou bien la gestion de la production, la gestion de plusieurs compagnies en même temps ou bien l'ajout d'autres langues.

Commencer un projet à partir de rien était une opportunité que nous voulions saisir. Ceci nous a permis d'approfondir nos connaissances acquises lors de notre formation et nous a permis de maîtriser de nouveaux outils et technologies de développement. Nous sommes donc fiers de notre travail et nous pensons être capable de relever notre prochain challenge.

Bibliographie et webographie

- [1]. MESURE DE LA PERFORMANCE GLOBALE :
https://mozartconsulting.sharepoint.com/Documents/MESURE%20DE%20LA%20PERFORMANCE%20GLOBALE_iae_univ_poitiers.pdf
- [2]. L'entreprise : sa mission, ses objectifs et buts, ses politiques, ses stratégies et ses tactiques par Cédric ISUMO EKEMBE Université de Kinshasa
- [3]. <http://www.epsilon.insee.fr/jspui/bitstream/1/56348/1/4p223.pdf>
- [4]. <https://www.celge.fr/article-conseil/histoire-logiciels-erp>
- [5]. <https://www.choisirmonerp.com/erp/definition-d-un-erp>
- [6]. https://www.bearingpoint.com/files/BearingPoint_SAP_Aberdeen_Group-Too_Small_For_ERP.pdf
- [7]. Impact de l'ERP sur la performance : cas d'IGL par Lilia GHARSALLAH Université de Sfax
- [8]. http://www.erp-infos.com/info_article/m/3203/Étude-erp-du-cxp-2017.html
- [9]. <https://www.journaldunet.com/solutions/reseau-social-d-entreprise/1058737-les-projets-de-deploiement-d-erp-au-peigne-fin/1058740-le-temps-d-implementation>
- [10]. <https://www.journaldunet.fr/business/dictionnaire-economique-et-financier/1199229-gestion-definition-traduction/>
- [11]. <https://www.journaldunet.fr/business/dictionnaire-economique-et-financier/1199051-stock-definition-traduction/>
- [12]. <https://www.mecalux.fr/blog/types-stocks>
- [13].
<https://webcache.googleusercontent.com/search?q=cache:Or1o09QZUqkJ:https://d1n7iqsz6ob2ad.cloudfront.net/document/pdf/5396f6248bccf.pdf+&cd=3&hl=fr&ct=clnk&gl=dz>
- [14]. http://elkadiridriss.magix.net/public/Files/polycopie_gestion_des_stocks.pdf
- [15]. https://www.memoireonline.com/10/13/7575/m_Conception-pour-la-gestion-d-un-systeme-d-information-pour-la-gestion-des-stocks-dans-une-entrepris15.html
- [16]. <http://www.logistiqueconseil.org/Articles/Entrepot-magasin/Valorisation-stocks.htm#CUMP>
- [17]. [https://www.creer-gerer-entreprendre.fr/6-la-gestion-de-lentreprise/6-1-la-gestion-comptable/les-methodes-de-valorisation-des-stocks-cump-fifo-lifo-et-nifo/#:~:text=Le%20NIFO%20\(valeur%20de%20remplacement,selon%20les%20normes%20IAS%20FIFRS.](https://www.creer-gerer-entreprendre.fr/6-la-gestion-de-lentreprise/6-1-la-gestion-comptable/les-methodes-de-valorisation-des-stocks-cump-fifo-lifo-et-nifo/#:~:text=Le%20NIFO%20(valeur%20de%20remplacement,selon%20les%20normes%20IAS%20FIFRS.)

- [18]. <https://www.cours-gratuit.com/cours-gestion/cours-la-gestiion-des-stocks-modele-de-wilson/download?chk=4b91a1801e34ddc3351abca4331b0c63%20Afficher%20le%20dossier>
- [19]. http://www.aunege.org/ressources/module_gestion_stocks/lecon3/courspdf/lecon3.pdf
- [20]. <http://www.logistiqueconseil.org/Articles/Logistique/Methode-pareto-20-80-abc.htm>
- [21]. http://www.mediafire.com/file/5u2p4zvn4hcl4w2/La_gestion_des_approvisionnements_et_de_stock.docx/file
- [22]. <http://www.logistiqueconseil.org/Articles/Entrepot-magasin/Inventaires.htm>
- [23]. <http://www.logistiqueconseil.org/Articles/Entrepot-magasin/Reapprovisionnement-stock.htm>
- [24]. https://agicap.fr/article/methodes-de-gestion-des-stocks/#mcetoc_1cfpucrpqt
- [25]. <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites>
- [26]. <https://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/diagramme-de-classe/>
- [27]. <https://laravel.com/docs/7.x/structure>
- [28]. <https://tsix.be/articles/un-front-de-blog-avec-livewire-et-alpinejs#:~:text=Livewire%20permet%20de%20créer%20des,transparente%2C%20sans%20rechargement%20de%20page!>
- [29] <https://slideplayer.fr/slide/3984697/>