

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique
Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master en Informatique

Spécialité : Génie Logiciel

Conception et réalisation d'une application web et mobile pour la gestion de stock. Cas d'étude : Agrana Fruit-Akbou.

Réalisé par :

Bougheriou Abderrahmane
Djouder Massil

Encadrant :

M. Allem Khaled
Co-Encadrante :
Mme. Bettar Wardia

Soutenu le **17/09/2025**

Devant le jury composé de :

Présidente : Mme. Gadouche Hania
Examinatrice : Mme. Cherifi Ferial
Examinatrice : Mme. Ghanem Souhila
Examinatrice : Mme. Azizou Zahia

Promotion : 2024 / 2025

Remerciements

Tout d'abord, nous exprimons notre profonde gratitude à Allah le Tout-Puissant pour nous avoir accordé le courage et la patience nécessaires pour mener à bien ce travail. Nous tenons à remercier chaleureusement tous les membres du jury pour avoir accepté d'évaluer notre mémoire et pour l'intérêt qu'ils ont porté à notre travail. Nous adressons aussi nos sincères remerciements à nos familles pour leur soutien constant, leurs encouragements et leurs conseils qui nous ont guidés tout au long de ce parcours académique. Nos remerciements les plus sincères vont tout d'abord à notre encadrant, M. Khaled Allem, pour son accompagnement précieux, son suivi attentif, ainsi que ses conseils éclairés qui ont grandement contribué à la réussite de ce travail. Nous souhaitons également exprimer notre profonde gratitude à l'équipe IT de Agrana Fruit, pour leur accueil chaleureux, leur disponibilité et leur précieuse coopération durant notre stage. Nous tenons à exprimer notre profonde gratitude envers toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire, que leur nom soit mentionné ici ou non. Leur collaboration et leurs contributions ont été essentielles et ont enrichi notre travail de manière significative. Enfin, nous remercions également nos camarades pour leur soutien et leur camaraderie tout au long de cette aventure académique.

Dédicace

À nos chers membres de la famille,

« Ce mémoire est dédié à vous, qui avez été une source constante de soutien, d'encouragement et d'amour tout au long de ce parcours académique. Votre présence inébranlable et votre soutien indéfectible a été les fondations sur lesquelles nous avons pu construire et réussir. Que ce travail soit une humble expression de notre gratitude et de notre amour pour vous tous. Merci d'avoir été notre inspiration ».

Table des matières

Remerciements	
Dédicace.....	
Table des matières	
Table des figures.....	
Liste des tableaux.....	
Liste des signes et acronymes	
CHAPITRE 1 : PRESENTATION DU CONTEXTE DU PROJET ET DE L'ORGANISME D'ACCUEIL.....	1
1.1 Introduction	1
1.2 Généralité sur la gestion de stock	1
1.2.1 Définition de la notion de stock.....	1
1.2.2 Définition de la gestion de stock	1
1.3 Les différents aspects de la gestion de stock.....	2
1.3.1 Types de stock	2
1.3.2 Avantages et inconvénients d'avoir un stock.....	3
1.3.3 Risque de la gestion de stock.....	3
1.4 Améliorations de la gestion de stock	4
1.5 Présentation de l'entreprise d'accueil (Agrana fruit).....	5
1.5.1 Gamme de produit.....	5
1.5.2 Historique de l'entreprise	5
1.5.3 Situation géographique.....	6
1.5.4 Fonctionnement général.....	6
1.6 Etude de l'existant	6
1.6.1 Utilisation de SAP B1 comme ERP.....	6
1.7 Limitations du système existant.....	7
1.8 Problématique	7
1.9 Solution envisagée	7
1.10 Objectifs du projet	8
1.11 Définition de quelques concepts métiers	8
1.12 Conclusion.....	9
CHAPITRE 2 : Méthodologie de conception et spécification des besoins.....	10
2.1 Introduction	10
2.2 Méthodologie de conception	10
2.2.1 Méthodes agiles.....	10
2.2.2 Méthode Scrum	10
2.2.3 Cycle de vie de la méthode Scrum.....	11
2.3 Langage de modélisation	13
2.3.1 Définition de UML.....	13
2.4 Architecture MVC.....	13
2.4.1 Organisation du MVC	13
2.5 Pilotage du projet avec Scrum.....	14
2.5.1 Rôle.....	14
2.5.2 Identification des acteurs.....	14
2.5.3 Diagramme de contexte	15
2.5.4 Diagramme de cas d'utilisation global.....	15
2.5.5 Planification des releases.....	16
2.5.6 Backlog-Product (Carnet de produit)	17
2.6 Conclusion.....	19
CHAPITRE 3 : Présentation de premier Release	20
3.1 Introduction	20
3.2 Étude du premier sprint	20

Les fonctionnalités traitées et la durée	20
Diagramme de cas d'utilisation du premier sprint	20
3.3 Les cas d'utilisations du premier sprint.....	21
3.3.1 Cas d'utilisation « s'authentifier ».....	21
3.3.2 Diagramme d'interaction de CU « S'authentifier ».....	23
3.3.3 Cas d'utilisation « Gérer comptes utilisateurs ».....	23
3.3.4 Diagramme de classe du premier sprint	26
3.4 Le schéma relationnel du premier sprint	26
3.4.1 Le dictionnaire de données du premier sprint	27
3.5 Les interfaces graphiques du premier sprint	27
3.6 Étude du deuxième sprint	28
3.6.1 Les fonctionnalités traitées et la durée.....	28
3.6.2 Diagramme de cas d'utilisation du deuxième sprint.....	29
3.7 Les cas d'utilisations du deuxième sprint.....	30
3.7.1 Cas d'utilisation « Gérer articles ».....	30
3.7.2 Diagramme d'interaction de CU « Gérer articles ».....	32
3.7.3 Cas d'utilisation « Gérer clients ».....	32
3.7.4 Diagramme d'interaction de CU « Gérer clients »	35
3.7.5 Diagramme de classe du deuxième sprint	36
3.8 Le schéma relationnel du deuxième sprint.....	36
3.8.1 Le dictionnaire de données du deuxième sprint	36
3.9 les interfaces graphiques du deuxième sprint	37
3.10 Conclusion.....	38

CHAPITRE 4 : Présentation de deuxième Release..... 39

4.1 Introduction	39
4.2 Étude du troisième sprint	39
4.2.1 les fonctionnalités traitées et la durée.....	39
4.2.2 Diagramme de cas d'utilisation de troisième sprint.....	40
4.3 Les cas d'utilisations de troisième sprint.....	40
4.3.1 Cas d'utilisation « Gérer lots ».....	40
4.3.2 Diagramme d'interaction de CU « Gérer lots »	43
4.3.3 Cas d'utilisation « Gérer entrepôts ».....	44
4.3.4 Diagramme d'interaction de CU « Gérer entrepôts ».....	46
4.3.5 Diagramme de classe du troisième sprint	47
4.4 Le schéma relationnel du troisième sprint.....	47
4.4.1 Le dictionnaire de données du troisième sprint.....	48
4.5 les interfaces graphiques du troisième sprint	48
4.6 Étude du quatrième sprint	49
4.6.1 les fonctionnalités traitées et la durée.....	49
4.6.2 Diagramme de cas d'utilisation de quatrième sprint.....	50
4.7 Les cas d'utilisations de quatrième sprint	51
4.7.1 Cas d'utilisation « Gérer mouvements ».....	51
4.7.2 Diagramme de classe du quatrième sprint.....	53
4.8 Le schéma relationnel du quatrième sprint.....	53
4.9.1 Le dictionnaire de données du quatrième sprint.....	53
4.9 les interfaces graphiques du quatrième sprint.....	54
4.10 Conclusion.....	54

Chapitre 5 : Présentation de troisième Release 55

5.1 Introduction	55
5.2 Étude du cinquième sprint.....	55
5.2.1 les fonctionnalités traitées et la durée.....	55
5.2.2 Diagramme de cas d'utilisation de cinquième sprint	56
5.3 Les cas d'utilisations de cinquième sprint	57
5.3.1 Cas d'utilisation « Gérer commande (espace magasinier) ».....	57
5.3.2 Diagramme d'interaction de CU « Gérer commandes(magasinier) ».....	60
5.3.3 Cas d'utilisation « Gérer commande (espace admin) ».....	61
5.3.4 Diagramme d'interaction de CU « Gérer commandes(admin) ».....	63
5.3.5 Diagramme de classe du cinquième sprint.....	64
5.4 Le schéma relationnel du cinquième sprint.....	64

5.4.1	Le dictionnaire de données du cinquième sprint	64
5.5	les interfaces graphiques du cinquième sprint.....	65
5.6	Étude du sixième sprint	66
5.6.1	Les fonctionnalités traitées et la durée.....	66
5.6.2	Diagramme de cas d'utilisation de sixième sprint	67
5.7	Les cas d'utilisations de sixième sprint	67
5.7.1	Cas d'utilisation « Gérer archives ».....	67
5.7.2	Diagramme d'interaction de CU « Gérer archives »	70
5.7.3	Cas d'utilisation « consulter notification ».....	71
5.7.4	Diagramme de classe du sixième sprint.....	73
5.8	Le schéma relationnel du sixième sprint.....	73
5.8.1	Le dictionnaire de données du sixième sprint.....	74
5.9	Les interfaces graphiques du sixième sprint	74
5.10	Outil de développement.....	74
5.11	Conclusion.....	75

Conclusion générale et perspectives.....	77
---	-----------

Bibliographie	78
----------------------------	-----------

Annexes	80
----------------------	-----------

1	GitHub	81
2	SQL.....	81
3	Bootstrap.....	81
4	Metrial_UI.....	81
5	Diagramme de séquence de CU « transférer lot ».....	82

Abstract

Table des figures

2.1	cycle de vie scrum	14
2.2	Diagramme de contexte dynamique.....	17
2.3	Diagramme de cas d'utilisation global.....	18
2.4	Planification des releases.....	19
3.1	Digramme de CU de sprint 1	23
3.2	Diagramme de séquence CU « S'authentifier ».....	24
3.3	Diagramme d'interaction de CU « S'authentifier ».....	25
3.4	Diagramme de séquence de CU « Gérer compte utilisateurs ».....	27
3.5	Diagramme de Classe du premier sprint	28
3.6	Interface d'authentification	29
3.7	Interface de gestion des comptes.....	30
3.8	Use case diagramme sprint 2.....	31
3.9	Diagramme de séquence de CU « Gérer articles ».....	33
3.10	Diagramme d'interaction de CU « Gérer articles ».....	34
3.11	Diagramme de séquence de CU « Gérer clients ».....	36
3.12	Diagramme d'interaction de CU « Gérer clients ».....	37
3.13	Diagramme de Classe du deuxième sprint	38
3.14	Interface de gestion des produits.....	40
3.15	Interface de gestion des clients	40
4.1	Diagramme de cas d'utilisation du sprint 3.....	42
4.2	Diagramme de séquence Gérer lots	44
4.3	Diagramme d'interaction de CU « Gérer lots ».....	45
4.4	Diagramme de séquence de CU « Gérer entrepôts »	47
4.5	Diagramme d'interaction de CU « Gérer entrepôts ».....	48
4.6	Diagramme de classes du troisième sprint	49
4.7	Interface de gestion des lots	51
4.8	Interface de gestion des entrepôts.....	51
4.9	Diagramme de cas d'utilisation du sprint 4.....	52
4.10	Diagramme de séquence de CU « Gérer mouvements »	54
4.11	Diagramme de classes du quatrième sprint.....	55
4.12	Interface de mouvements (E/S).....	56
5.1	Diagramme de cas d'utilisation du sprint 5	58
5.2	Diagramme de séquence du cas d'utilisation « Gérer commandes (magasinier) »	61
5.3	Diagramme d'interaction de CU « Gérer commandes(magasinier) »	62
5.4	Diagramme de séquence du cas d'utilisation « Gérer commandes »	64
5.5	Diagramme d'interaction de CU « Gérer commandes(admin) »	65
5.6	Diagramme de classes du cinquième sprint.....	66
5.7	Diagramme de cas d'utilisation du sprint 6.....	67
5.8	Diagramme de Séquence de CU « Gérer archives »	68
5.9	Diagramme de Séquence de CU « Gérer archives »	69
5.10	Diagramme de Séquence de CU « Consulter notification ».....	71
5.11	Diagramme de classes du sixième sprint	69

Liste des tableaux

1.1	Les aspects de la gestion de stock.....	2
2.1	La répartition des rôles pour la réalisation de projet.....	10
2.2	Interactions entre Administrateur, Système et Magasinier.....	11
2.3	Backlog-Product et User stories.....	14
3.1	Le sprint backlog du premier sprint.....	16
3.2	La description textuelle de CU « s'authentifier ».....	16
3.3	La description textuelle de CU « Gérer compte utilisateur ».....	18
3.4	Le dictionnaire de données du premier sprint.....	21
3.5	Le sprint backlog du deuxième sprint.....	22
3.6	La description textuelle de CU « Gérer articles ».....	24
3.7	La description textuelle de CU « Gérer clients ».....	26
3.8	Le dictionnaire de données de deuxième sprint.....	30
4.1	Le sprint backlog du troisième sprint.....	32
4.2	La description textuelle du CU « Gérer lots ».....	34
4.3	La description textuelle de CU « Gérer entrepôts ».....	37
4.4	Le dictionnaire de données du troisième sprint.....	41
4.5	Le sprint backlog du quatrième sprint.....	42
4.6	La description textuelle de CU « Gérer mouvements ».....	43
4.7	Le dictionnaire de données du quatrième sprint.....	46
5.1	Le sprint backlog du cinquième sprint.....	47
5.2	Description textuelle du cas d'utilisation « Gérer commandes (Magasinier) ».....	49
5.3	Description textuelle du cas d'utilisation « Gérer commandes (espace admin) ».....	52
5.4	Le dictionnaire de données du cinquième sprint.....	56
5.5	Le sprint backlog du sixième sprint.....	57
5.6	Description textuelle du cas d'utilisation « Gérer archives ».....	59
5.7	Description textuelle du cas d'utilisation « consulter notification ».....	62
5.8	Le dictionnaire de données du sixième sprint.....	67

Liste des signes et acronymes

- **AFIM** : *Agrana Fruits Inventory Management*
- **API** : *Application Programming Interface*
- **CU** : *Cas d'Utilisation*
- **ERP** : *Enterprise Resource Planning* (progiciel de gestion intégré)
- **HTML** : *HyperText Markup Language*
- **CSS** : *Cascading Style Sheets*
- **JS / JavaScript** : Langage de programmation côté client
- **MIT** : *Massachusetts Institute of Technology* (Licence MIT pour Bootstrap)
- **MVC** : *Modèle–Vue–Contrôleur*
- **QR Code** : *Quick Response Code*
- **SaaS** : *Software as a Service*
- **SAP B1** : *SAP Business One* (ERP utilisé par Agrana)
- **SQL** : *Structured Query Language*
- **UML** : *Unified Modeling Language*
- **XP** : *Extreme Programming* (méthode agile)
- **Scrum** : Méthodologie agile de gestion de projet, basée sur des itérations appelées *sprints*.

INTRODUCTION GÉNÉRALE

Dans un contexte économique marqué par une concurrence accrue et une exigence croissante en matière de performance, la gestion des stocks occupe une place centrale dans la stratégie des entreprises industrielles. Elle permet non seulement de garantir la disponibilité des produits, mais aussi de réduire les coûts de stockage et d'optimiser les flux logistiques. Une gestion efficace des stocks contribue ainsi directement à la satisfaction des clients et à l'amélioration de la compétitivité de l'entreprise.

Face à ces enjeux, de nombreuses entreprises s'appuient aujourd'hui sur des solutions informatisées pour automatiser le suivi des produits, améliorer la traçabilité et fiabiliser la prise de décision. Cependant, certaines difficultés persistent, notamment en matière de mobilité, de réactivité et d'intégration des données entre les différents services. C'est dans ce cadre que s'inscrit le présent projet, réalisé au sein de l'entreprise **Agrana Fruit Algérie**, spécialisée dans la production de préparations de fruits destinées à l'industrie agroalimentaire.

L'objectif principal de ce travail est de concevoir et de développer une **application web et mobile de gestion de stock**, intuitive et ergonomique, permettant de gérer l'ensemble des entités liées au processus logistique : articles, lots, mouvements, commandes, entrepôts, clients et fournisseurs. Cette solution s'appuie sur une base de données inspirée du système **SAP Business One**, tout en intégrant des fonctionnalités avancées de traçabilité, de suivi des seuils critiques et d'alertes intelligentes liées aux ruptures et aux dates d'expiration.

Pour la réalisation de ce projet, nous avons adopté la **méthodologie agile Scrum**, qui favorise une approche itérative et collaborative. Le langage **UML** a été utilisé pour la modélisation et la structuration du système, tandis que le développement s'est appuyé sur des technologies modernes telles que **React**, **Node.js** et **MySQL**. L'architecture de l'application a été pensée de manière modulaire afin de faciliter, à terme, son intégration complète avec SAP Business One via des services web.

Organisation du mémoire

Ce mémoire est structuré en cinq chapitres principaux, suivis d'une conclusion générale :

- **Chapitre I : Présentation du contexte du projet et de l'organisme d'accueil**

Ce chapitre présente l'entreprise Agrana Fruit Algérie, son domaine d'activité et son organisation interne. Il expose également la problématique identifiée, les objectifs fixés et les outils utilisés, notamment SAP Business One.

- **Chapitre II : Méthodologie de conception et spécification des besoins**

Ce chapitre décrit la méthodologie agile Scrum, les outils de conception et de

développement employés, ainsi que la spécification fonctionnelle et technique du système.

- **Chapitre III : Présentation du premier release**

Il présente le premier incrément du projet, comprenant la mise en place du système d'authentification, la gestion des utilisateurs, et la base du module d'articles et de clients.

- **Chapitre IV : Présentation du deuxième release**

Ce chapitre aborde la gestion des lots, des mouvements de stock et des entrepôts, en insistant sur la traçabilité et la cohérence des données.

- **Chapitre V : Présentation du troisième release**

Ce dernier chapitre détaille la gestion des commandes, la distinction entre les espaces administrateur et magasinier, et la mise en place des notifications automatiques.

Enfin, une **conclusion générale** récapitule les résultats obtenus et propose des perspectives d'évolution pour l'application.

CHAPITRE 1 : PRESENTATION DU CONTEXTE DU PROJET ET DE L'ORGANISME D'ACCUEIL

1.1 Introduction

Ce chapitre présente d'abord les notions fondamentales liées à la gestion des stocks, puis décrit l'organisme d'accueil où se déroule le stage, et il s'achève par une analyse de l'existant permettant de préciser la problématique à laquelle ce travail propose une solution logicielle.

1.2 Généralité sur la gestion de stock

Cette section présente les notions générales liées à la gestion des stocks, en commençant par une définition du stock et de la gestion de stock, avant d'aborder les différents aspects qui y sont associés.

1.2.1 Définition de la notion de stock

Un stock est une quantité de biens ou d'articles mis en réserve pour une utilisation future. Le stock est utilisé soit pour faciliter ou pour assurer la continuité de la production, soit pour satisfaire une demande intérieure formulée par l'un des services de l'entreprise, dans ce cas, on parle de stocks de matières premières, de fournitures ou de produits intermédiaires, ou alors pour assurer une demande extérieure provenant des clients (stocks de produits finis). On peut dire que les stocks sont utiles et nécessaires pour assurer le bon fonctionnement de l'entreprise [1].

Par exemple : On ne peut imaginer un constructeur d'automobile qui ne produit des voitures qu'à la demande et qui commande un à un les pneus nécessaires pour satisfaire la demande. D'un autre côté, un stock étant une ressource qui reste inutilisée pendant un certain temps, constitue un gaspillage, telle est la position des fondateurs de l'approche juste-à-temps qui visent le zéro stock. Ainsi, pour concilier disponibilité des ressources et maîtrise des coûts, la gestion de stock s'impose comme une activité stratégique visant à optimiser les niveaux de stock tout en garantissant la satisfaction des besoins internes et externes de l'entreprise.

1.2.2 Définition de la gestion de stock

La gestion de stock consiste principalement à déterminer à quel moment et en quelle quantité un article devra être renouvelé. Que ce soit pour les matières premières, les pièces de rechange des équipements de production ou pour les produits finis, la gestion des stocks ne prend un sens qu'à partir du moment où l'on connaît les objectifs à atteindre. Or, ces objectifs sont définis par les activités de

planification de la production. Les quantités de matières premières, pièces et composants d'assemblage sont déterminées à partir des besoins de la production.[1]

En résumé La gestion de stock est l'ensemble des méthodes et des outils permettant de suivre, contrôler et optimiser les entrées, les sorties et les niveaux de stock d'une entreprise afin d'éviter les ruptures ou les sur- stocks.

1.3 Les différents aspects de la gestion de stock

Ce tableau ci-dessus montre les activités à synchroniser et les objectifs à atteindre et les contraintes à respecter [1] :

Activités	Objectifs	Contraints
<ul style="list-style-type: none"> – rechercher, organiser et traiter l'information sur les articles en stocks ; – déterminer les quantités à commander, quand commander ; – déterminer les stocks de sécurité ; – organiser et contrôler l'entreposage ; 	<ul style="list-style-type: none"> – minimiser les coûts de maintien en inventaire ; – minimiser les coûts de pénurie ; – minimiser les coûts de com- mande ; 	<ul style="list-style-type: none"> – prévision de la demande et planification de la production ; – capacité de production ; – délais de livraison – personnel disponible ; – infrastructure en place

Table 1.1 – Les aspects de la gestion de stock.

1.3.1 Types de stock

La gestion de stock repose en grande partie sur la bonne classification des articles en fonction de leur nature et de leur rôle dans le processus de production ou de distribution. Voici les principaux types de stock [2],[3].

- **Stock de matières premières** : il représente les produits achetés par l'entreprise, par l'intermédiaire de ses fournisseurs, et destinés à une transformation ultérieure.
- **Stock de produits en cours (ou en cours de fabrication)** : il s'agit de produits non finis qui ne peuvent être mis en vente qu'après avoir subi une certaine transformation, pratiquée par l'entreprise.
- **Stock de produits finis** : ce sont les articles qui ont subi une transformation par l'entreprise afin d'atteindre leur niveau final de fabrication. Ils sont, désormais, prêts à la vente.
- **Stock des marchandises** : on parle, ici, de stocks de commerçants composés de produits sans

valeur ajoutée de transformation de la part de l'entreprise qui les revend à profit.

- **stock d'alerte** : il correspond à la somme du stock minimum et du stock de sécurité. C'est généralement le moment où s'active le point de commande. Le stock doit être renouvelé, et de nouvelles commandes doivent être réalisées auprès des fournisseurs.

1.3.2 *Avantages et inconvénients d'avoir un stock*

Dans cette partie, nous allons voir quels sont les avantages et les inconvénients d'avoir un stock au sein d'une organisation ou une entreprise.

Les avantages

La mise en place d'un système de gestion des stocks présente de nombreux avantages pour l'entreprise.

Parmi les avantages, nous citons :

- Assurer le délai de livraison du client.
- Prévenir des risques de pénurie.
- Répondre à une exigence clients.
- Se protéger des variations de valeur des marchandises.

Les inconvénients

Même si les stocks jouissent de nombreux avantages, cependant d'autres raisons poussent à limiter ses stocks, nous citons à titre d'exemple :

- Immobilisation des moyens financiers importants.
- Immobilisation des surfaces de stockage.
- Potentiel de risque (perte, détérioration, incendie).
- Les coûts engendrés par l'entretien et la protection des stocks.

1.3.3 *Risque de la gestion de stock*

La gestion de stock joue un rôle central dans la performance logistique et financière de l'entreprise. Pourtant, elle comporte de nombreux risques qui, s'ils ne sont pas anticipés, peuvent entraîner des conséquences graves. Voici les principaux risques, illustrés par des exemples réels issus de l'industrie ou du commerce [4].

- **Surstockage** : Avoir trop de stock peut entraîner des coûts de stockage élevés, une immobilisation inutile de capital et des risques accrus de détérioration ou d'obsolescence des

produits.

- **Ruptures de stock** : À l'inverse, ne pas avoir suffisamment de stock peut mener à des ruptures de stock, ce qui perturbe la production, retarde les livraisons et nuit à la satisfaction des clients.
- **Obsolescence** : Produits qui ne se vendent plus parce qu'ils sont dépassés techniquement, par la mode, par de nouveaux modèles, ou parce que la demande a changé. Le stock reste mais ne génère plus de valeur, ou doit être vendu à perte.
- **Données inexactes ou mauvaise traçabilité** : Erreurs dans les enregistrements de stock : mauvais comptage, saisies incorrectes, localisation ou identification erronée des produits. Ces erreurs faussent les décisions (achats, production, etc.).
- **Risques de sécurité et d'organisation** : Un stock mal organisé peut générer des dangers physiques pour les employés. cela provoque des : accidents, incendies, pertes de produits.

1.4 Améliorations de la gestion de stock

Ci-dessus on va présenter quelques pistes d'amélioration pour optimiser la gestion des stocks au sein de l'entreprise.

- **Catégoriser ses produits** : l'étape de la classification est d'une importance capitale car l'objectif est de mieux gérer des articles identiques et d'organiser le suivi de son stock, sinon le risque de confusion de deux références appréhendées identiques risque d'induire des ruptures de stock, d'où l'importance de ne pas négliger cette tâche.
- **Éviter le stock mort** : Le stock mort, aussi nommé stock dormant, consiste en une marchandise qui se vend lentement. Parfois, certains produits ne se vendent jamais. Cette absence de rotation crée des coûts de stockage importants. Tout comme le fait de perdre de l'espace au profit de produits avec une meilleure rotation. Une prévision de la demande erronée est un acteur d'apparition du stock mort mais pas uniquement. Des aléas imprévisibles d'un client peuvent être des éléments déclencheurs comme la crise sanitaire du Covid-19.
- **Mise en œuvre d'un système ERP** : L'ERP (Enterprise Resource Planning) regroupe toutes les fonctions des entreprises (achats, stocks, ventes, finance. . .) sur une même plateforme logicielle, ce qui permet d'avoir un suivi en temps réel des stocks, d'assurer une meilleure coordination entre services, de réaliser des prévisions plus fiables et de réduire le coût du surstock ou du manque à gagner en cas de rupture.
- **Créer un tableau de bord (Dashboard)** : La mise en œuvre d'un tableau de bord contribue à la bonne exploitation des données statistiques relatives à la situation du stock. Toutefois, il

convient de déterminer explicitement les objectifs afin de retenir les indicateurs les plus appropriés.

1.5 Présentation de l'entreprise d'accueil (Agrana fruit)

Agrana (pour Elaboration de fruits) est une société par action (SPA), c'est une société à caractère industriel évoluant dans le domaine agro-alimentaire se spécialise dans la transformation de fruits pour les producteurs nationaux ainsi que les multinationaux, elle vielle pour avoir un produits fini conforme de bonne qualité toute an respectant les normes internationales de fabrication [5].

1.5.1 Gamme de produit

L'entreprise propose une gamme variée de produits issus de matières premières fruitières soigneusement sélectionnées. on cite ici quelques-uns de ces produits :

- Préparations morceau
- Préparation pulpes
- Préparation purées
- Compote
- Sirop
- Compound

Ceux sont des produits alimentaires intermédiaires qui sont destinés à la fabrication des produits laitiers comme les yaourts SOUMMAM, HODNA, DANONE, des boissons comme le jus ROUIBA et le jus RAMDY, des pâtisseries et des glaces, etc.

1.5.2 Historique de l'entreprise

Ici dessus on vous montre l'évolution de l'entreprise Agrana :

- 2006- Création de la société par M. Boussaad Batouche .
- 2007- Partenariat avec FRULACT Portugal.
- 2008 : frulact évolue en ELAFRUIT Algérie.
- 2011-Certification ISO 22000 :2005 HACCP.
- 2013- Rachat de parts portugaises par MONSIEUR BATOUCHE et création d'AGRANA FRUIT.
- 2018 : installation de AGRANA FRUIT par les Autrichiens en 12 juillet 2021-Certification FSSC2200 V 5. 1.44

1.5.3 *Situation géographique*

- Agrana Fruit Algérie est implantée dans une zone industrielle « TAHARACHT », véritable carrefour économique de Bejaia, de quelque 70 unités de production agroalimentaire et en cours d'expansion.
- A deux (02) km d'une grande agglomération (AKBOU), et à quelque dizaine de mètres de la voie ferrée.
- A 60 km de Bejaia, et à 170 km à l'ouest de la capital Alger.
- D'un port à fort tract et un aéroport international reliant diverse destination, (Paris, Marseille, Lyon, st Etienne, et Charleroi).
- Par ailleurs on trouve des acteurs économiques importants tel que : SOUMMAM, RAMDY, DANONE, IFRI, etc [16].

1.5.4 *Fonctionnement général*

L'entreprise fonctionne avec plusieurs services clés :

- **Production** : réception des matières premières, transformation, stockage.
- **Qualité** : contrôle des matières, suivi de traçabilité, respect des normes ISO.
- **Logistique** : gestion des entrepôts, mouvements de stock, expéditions.
- **Informatique (IT)** : gestion des infrastructures techniques, maintenance du système SAP Business One.

1.6 **Etude de l'existant**

Cette section analyse l'existant au sein de l'entreprise, en présentant les outils utilisés pour la gestion des stocks et des opérations. Elle met en évidence leurs atouts et limites afin de dégager les besoins du projet.

1.6.1 *Utilisation de SAP B1 comme ERP*

Agrana Fruits Algérie utilise **SAP Business One (SAP B1)** [6], un ERP reconnu, pour gérer l'ensemble de ses opérations (achats, ventes, gestion des articles, stocks, traçabilité, fournisseurs, etc.). Ce système centralisé permet de garantir la cohérence des données entre les différents services

Durant notre projet, nous avons été confrontés à une contrainte importante : aucune API publique ni

environnement de test n'était mis à disposition pour les stagiaires. Pour des raisons de sécurité, l'accès au système SAP réel était strictement limité, ce qui rendait impossible toute expérimentation directe sur la base de données de production. Afin de contourner cette limitation, nous avons conçu une base de données parallèle simulée, élaborée à partir de l'analyse du modèle SAP et de la documentation interne. Cette approche nous a permis de développer et de tester l'application en toute autonomie, sans compromettre l'intégrité ni la confidentialité des données réelles de l'entreprise.

1.7 Limitations du système existant

Les principales limites observées dans le fonctionnement actuel sont :

- . Complexité de l'interface SAP, absence d'une interface simplifiée et accessible aux utilisateurs non spécialisés.
- Difficultés de consultation rapide des mouvements de stock ou de l'historique des lots.
- Manque d'un outil adapté à une utilisation sur mobile ou en conditions hors-ligne.
- Pas d'interconnexion en temps réel avec des interfaces dédiées aux rôles spécifiques (magasinier, administrateur).

1.8 Problématique

Bien que SAP B1 réponde aux exigences standards d'une entreprise industrielle, plusieurs freins apparaissent lors d'une utilisation terrain, notamment :

- **Manque de mobilité** : les magasiniers n'ont pas de solution mobile simple pour enregistrer des entrées ou consulter des stocks.
- **Interface trop complexe** pour les utilisateurs non familiers avec SAP.
- **Difficulté d'adaptation** de l'interface aux processus locaux ou aux contraintes spécifiques du terrain.

1.9 Solution envisagée

Pour aborder cette problématique, nous avons proposé des outils complémentaires à SAP B1, permettant :

- Facilite le suivi des articles par lot et les mouvements de stock.
- Permettre l'enregistrement des entrées, sorties et transferts rapidement sur le terrain.

- Intégrer la gestion des fournisseurs, clients, entrepôts et commandes.
- Traiter les retours clients avec orientation vers entrepôts de recyclage ou destruction.

Cette application ne remplace pas SAP B1, mais agit comme :

- Un outil de complément terrain pour alléger les tâches opérationnelles.
- Une interface conviviale et intuitive, adaptée aux utilisateurs métiers.
- Une solution modulaire, potentiellement interfaçable par API dans le futur.

1.10 Objectifs du projet

Les objectifs du projet sont les suivants :

- Développer une application Web ou/et mobile ergonomique et intuitive.
- Proposer une base de données parallèle simulée, inspirée de SAP B1.
- Permettre une gestion complète des articles, lots, mouvements, commandes, entrepôts, clients et fournisseurs.
- Offrir des outils de traçabilité des lots, de gestion de seuils critiques et d’alertes (expiration, rupture).
- Structurer l’application de façon modulaire, facilitant son intégration future avec SAP B1 via des APIs ou services tiers.

1.11 Définition de quelques concepts métiers

Terme	Définition
Lot	Un lot correspond à un ensemble homogène de produits fabriqués ou reçus au même moment et partageant les mêmes caractéristiques (date de fabrication, fournisseur, etc.). Il permet une meilleure organisation et un suivi précis des articles [6].
Traçabilité	La traçabilité est l’aptitude à retrouver l’historique, l’utilisation ou la localisation d’un article ou d’une activité semblable au moyen d’une identification enregistrée [7].
Recycler une commande	Le recyclage d’une commande consiste à réutiliser, ajuster ou reprogrammer une commande déjà existante au lieu de la supprimer. Cela permet de corriger des erreurs ou d’optimiser le processus sans perdre les données initiales.

Détruire commande	une	La destruction d'une commande signifie l'annulation définitive de celle-ci, avec suppression de toutes ses informations du système. Elle est utilisée en cas d'erreur irrécupérable, de commande non conforme ou de produit obsolète.
----------------------	-----	---

1.12 Conclusion

Le premier chapitre a permis de présenter le contexte général du projet ainsi que l'organisme d'accueil. Il a également mis en lumière les limites de l'application actuelle de gestion des stocks. ces éléments soulignent la nécessité de concevoir une solution plus appropriée, dont le développement sera abordé dans les chapitres suivants.

CHAPITRE 2 : Méthodologie de conception et spécification des besoins

2.1 Introduction

Dans ce chapitre, nous présentons notre approche méthodologique pour la conception de notre système, ainsi que pour la spécification de ses besoins. La première partie traitera l'utilisation de la méthodologie agile Scrum et l'usage du langage de modélisation UML pour représenter les diverses composantes du système. Ensuite, nous explorerons l'architecture MVC du système et expliquerons comment elle sépare les composants clés. La seconde partie se concentrera sur la spécification des besoins et des exigences. Ce chapitre établira ainsi les bases de notre projet en adoptant une méthodologie de conception agile et en détaillant les besoins et exigences du système.

2.2 Méthodologie de conception

Nous avons suivi une méthode de conception agile et le langage de modélisation unifié UML pour développer notre projet.

2.2.1 Méthodes agiles

Une méthode Agile est une approche itérative et collaborative, capable de prendre en compte les besoins initiaux du client et ceux liés aux évolutions, le principe de base consiste à proposer une version minimale du logiciel puis à intégrer des fonctionnalités supplémentaires à cette base, par processus itératif. Le processus itératif regroupe une séquence d'instructions à répéter autant de fois que possible, selon le besoin. [8]

Nous avons opté pour la méthode Scrum pour la gestion de notre projet, parmi les méthodes agiles populaires telles que Kanban, Extreme Programming (XP) pour sa flexibilité et son approche itérative, permettant des ajustements rapides basés sur les retours des utilisateurs. Contrairement aux autres méthodes, Scrum favorise une collaboration accrue entre les équipes et une planification adaptative

2.2.2 Méthode Scrum

La méthodologie Scrum est une approche agile itérative et collaborative, aujourd'hui « Scrum » est la méthode agile la plus populaire. Ce terme signifie « mêlée » au rugby. La méthode scrum s'appuie sur des « sprints » qui sont des espaces temps assez courts pouvant aller de quelques heures jusqu'à un mois. Généralement et de préférence un sprint s'étend sur deux semaines. À la fin de chaque sprint, l'équipe présente ce qu'elle a ajouté au produit.[6]

Scrum regroupe trois acteurs :

- **Le Product Owner** : (ou « Directeur de produit ») : il communique les objectifs premiers des clients et utilisateurs finaux, coordonne l'implication des utilisateurs et des parties prenantes, et se coordonne lui-même avec les autres product owners pour assurer une cohérence.
- **Le Scrum Master** : membre de l'équipe, il a pour but d'optimiser la capacité de production de l'équipe. Pour se faire, le scrum master aide l'équipe à travailler de façon autonome tout en s'améliorant d'avantage.
- **L'équipe** : opérationnelle (qui regroupe idéalement moins de dix personnes) : la particularité d'une équipe scrum est qu'elle est dépourvue de toute hiérarchie interne. Une équipe scrum est auto organisée [6].

2.2.3 Cycle de vie de la méthode Scrum

Cette section présente le cycle de vie de la méthode Scrum, en décrivant ses principaux concepts, les réunions qui rythment les sprints, ainsi que les étapes clés du déroulement du processus.

Les termes concernant la méthode Scrum

- **Sprints** : Périodes itératives de développement, d'une durée généralement de 2 à 4 semaines.
- **Product backlog** : Liste hiérarchisée des exigences initiales établie en collaboration avec le client.
- **Sprint planning meeting** : Réunion avant chaque sprint pour sélectionner les exigences prioritaires à réaliser.
- **Sprint backlog** : Ensemble des exigences sélectionnées pour le sprint en cours, prêtes à être développées et testées.
- **Daily scrum meeting** : Réunion quotidienne de l'équipe pour discuter des avancées, des obstacles et des tâches à venir.
- **Sprint review meeting** : Réunion à la fin de chaque sprint pour présenter les fonctionnalités développées et recueillir les retours du client.
- **Estimation et planification** : Évaluation des tâches à effectuer, découpage des user stories en tâches et notation de leur complexité.

Les étapes de déroulement de la méthode Scrum

La figure 2.1 représente le cycle de vie de la méthode scrum [17].

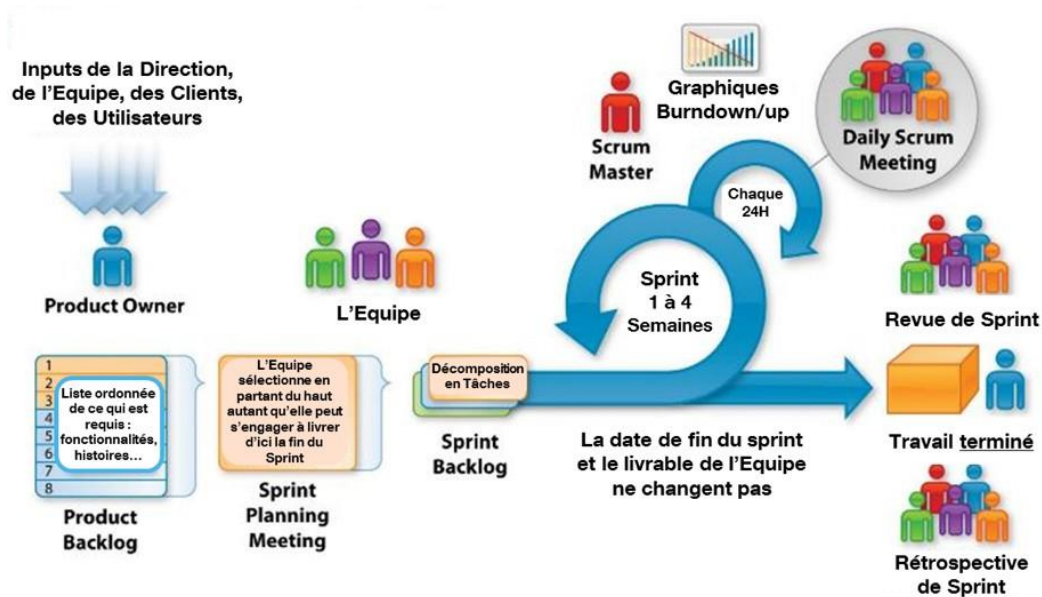


Figure 2.1 – cycle de vie scrum

- **Planification du Sprint** : Avant chaque Sprint, l'équipe se réunit pour sélectionner les exigences prioritaires à partir du Product Backlog et définir l'objectif du Sprint.
- **Développement du Sprint** : Pendant le Sprint, l'équipe travaille sur les éléments du Sprint Backlog, se réunissant quotidiennement lors des Daily Scrums pour suivre les progrès et résoudre les obstacles.
- **Revue de Sprint** : À la fin de chaque Sprint, l'équipe organise une réunion de revue pour présenter les fonctionnalités développées, recueillir les retours et planifier les prochaines étapes.
- **Rétrospectives de Sprint** : Suite à la revue, l'équipe se réunit pour réfléchir à son propre processus de travail, identifier les points forts et les axes d'amélioration, et mettre en place des actions pour le Sprint suivant.
- **Planification du Sprint suivant** : Enfin, l'équipe planifie le prochain Sprint en se basant sur les retours des revues et des rétrospectives, et sélectionne les éléments à inclure dans le nouveau Sprint Backlog.

Ce cycle se répète de manière itérative tout au long du projet, permettant à l'équipe de développer le produit de manière progressive, en s'adaptant aux retours et aux changements.

2.3 Langage de modélisation

Nous avons utilisé UML pour sa capacité à modéliser visuellement les systèmes complexes, facilitant ainsi la compréhension et la communication entre les équipes. De plus, UML aide à documenter les exigences et les structures du projet de manière cohérente et standardisée.

2.3.1 Définition de UML

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

Pour notre projet, nous avons opté pour l'utilisation de l'UML en raison de sa richesse sémantique et syntaxique, permettant une description efficace de la structure et du comportement des systèmes logiciels complexes que nous développons [9].

2.4 Architecture MVC

L'architecture Modèle/View/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface [10].

L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

2.4.1 Organisation du MVC

Examinons de plus près la structure de ce modèle d'architecture en séparant ses trois composantes principales : le modèle, la vue et le contrôleur.[9]

Modèle : cette partie gère ce qu'on appelle la logique métier de votre site. Elle comprend notamment la gestion des données qui sont stockées, mais aussi tout le code qui prend des décisions autour de ces données. Son objectif est de fournir une interface d'action la plus simple possible au contrôleur. On y trouve donc entre autres des algorithmes complexes et des requêtes SQL.

Vue : cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.

Contrôleur : cette partie gère les échanges avec l'utilisateur. C'est en quelque sorte l'intermédiaire entre l'utilisateur, le modèle et la vue. Le contrôleur va recevoir des requêtes de l'utilisateur. Pour chacune, il va demander au modèle d'effectuer certaines actions (lire des articles de blog depuis une base de données, supprimer un commentaire) et de lui renvoyer les résultats (la liste des articles, si la suppression est réussie). Puis il va adapter ce résultat et le donner à la vue. Enfin, il va renvoyer la nouvelle page HTML, générée par la vue, à l'utilisateur.

2.5 Pilotage du projet avec Scrum

Cette section présente le pilotage du projet en s'appuyant sur la méthodologie Scrum, en mettant l'accent sur la planification des sprints et l'organisation des activités de développement.

2.5.1 Rôle

La répartition des rôles dans notre projet est la suivante :

Rôle SCRUM	Personnes affectées
Product Owner	M. LOUABAR Salim, représentant de la SPA AGRANA fruit
Scrum Master	M. ALLEM Khaled
Équipe de développement	M. BOUGHEERIOU Abderrahmane et M. DJOUDER Massil

Table 2.1 – La répartition des rôles pour la réalisation de projet.

2.5.2 Identification des acteurs

Un acteur est une représentation abstraite d'une entité externe qui interagit avec le système en jouant un rôle spécifique. Cela peut être une personne, un processus ou une autre entité externe qui communique avec le système en émettant ou en recevant des messages susceptibles de contenir des données.

Dans le contexte de notre système, nous identifions deux acteurs principaux :

- **Administrateur** : Ce rôle est attribué à une personne qui supervise et gère l'ensemble des activités et fonctions de l'application. L'administrateur prend en charge la gestion des comptes utilisateurs, la définition des droits d'accès et il assure le bon fonctionnement de l'application.
- **Magasinier** : Ce rôle est attribué à un utilisateur chargé de la gestion opérationnelle des stocks au sein de l'entreprise. Le magasinier est responsable de la réception des marchandises, de leur rangement dans les entrepôts, du suivi des mouvements de stock (entrées et sorties), ainsi que de la préparation des commandes. Il veille à la bonne tenue des inventaires, au respect des seuils critiques, et à la traçabilité des lots. Le magasinier utilise l'application pour enregistrer les opérations logistiques quotidiennes de manière fiable et en temps réel.

2.5.3 Diagramme de contexte

La figure 2.2 représente le diagramme de contexte dynamique qui illustre les interactions entre notre système de gestion de stock AFIM (Agrana Fruits Inventory Management) et son environnement.

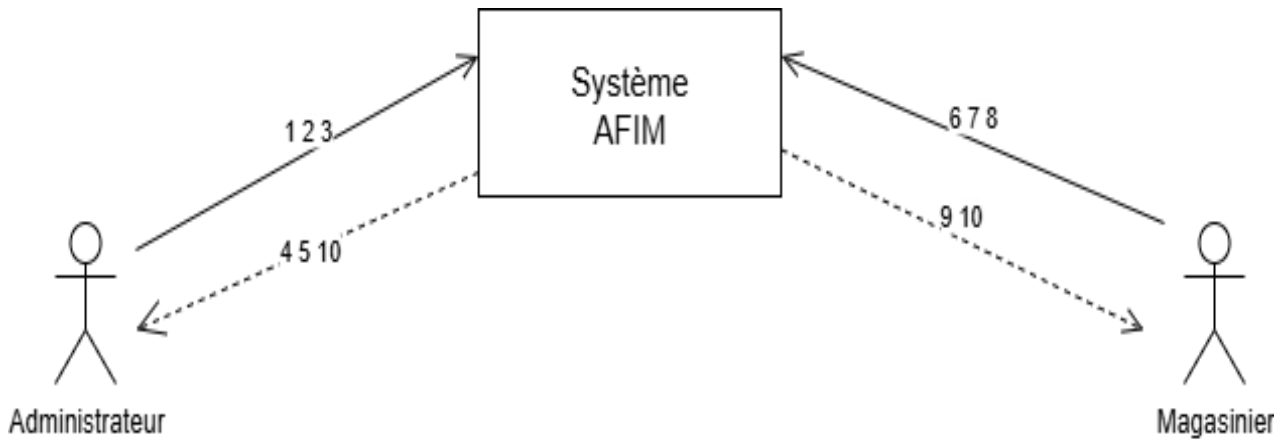


Figure 2.2 – Diagramme de contexte dynamique

Tableau de flux de données

Administrateur	Système de gestion de stock	Magasinier
1. Gérer articles / commandes	4. Générer les statistiques de stock	6. Effectuer entrées/sorties
2. Gérer clients / fournisseurs	5. Afficher les alertes de rupture de stock	7. Créer commandes clients/fournisseurs
3. Créer utilisateurs	9. Confirmation de la disponibilité du stock	8. Saisir retours clients
	10. Générer messages d'erreur / de succès	

Table 2.2 – Interactions entre Administrateur, Système AFIM et Magasinier

2.5.4 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation montre les principales fonctionnalités d'un système du point de vue des acteurs, sans détailler leurs interactions, chaque cas d'utilisation décrit comment un acteur externe interagit avec le système pour bénéficier d'un service complet. Un acteur principal bénéficie directement du service, tandis que les acteurs secondaires peuvent fournir des informations supplémentaires. Un cas d'utilisation peut avoir un acteur principal et des acteurs secondaires, même optionnels. Les acteurs principaux obtiennent des résultats observables, tandis que les acteurs secondaires peuvent être sollicités pour des informations complémentaires.

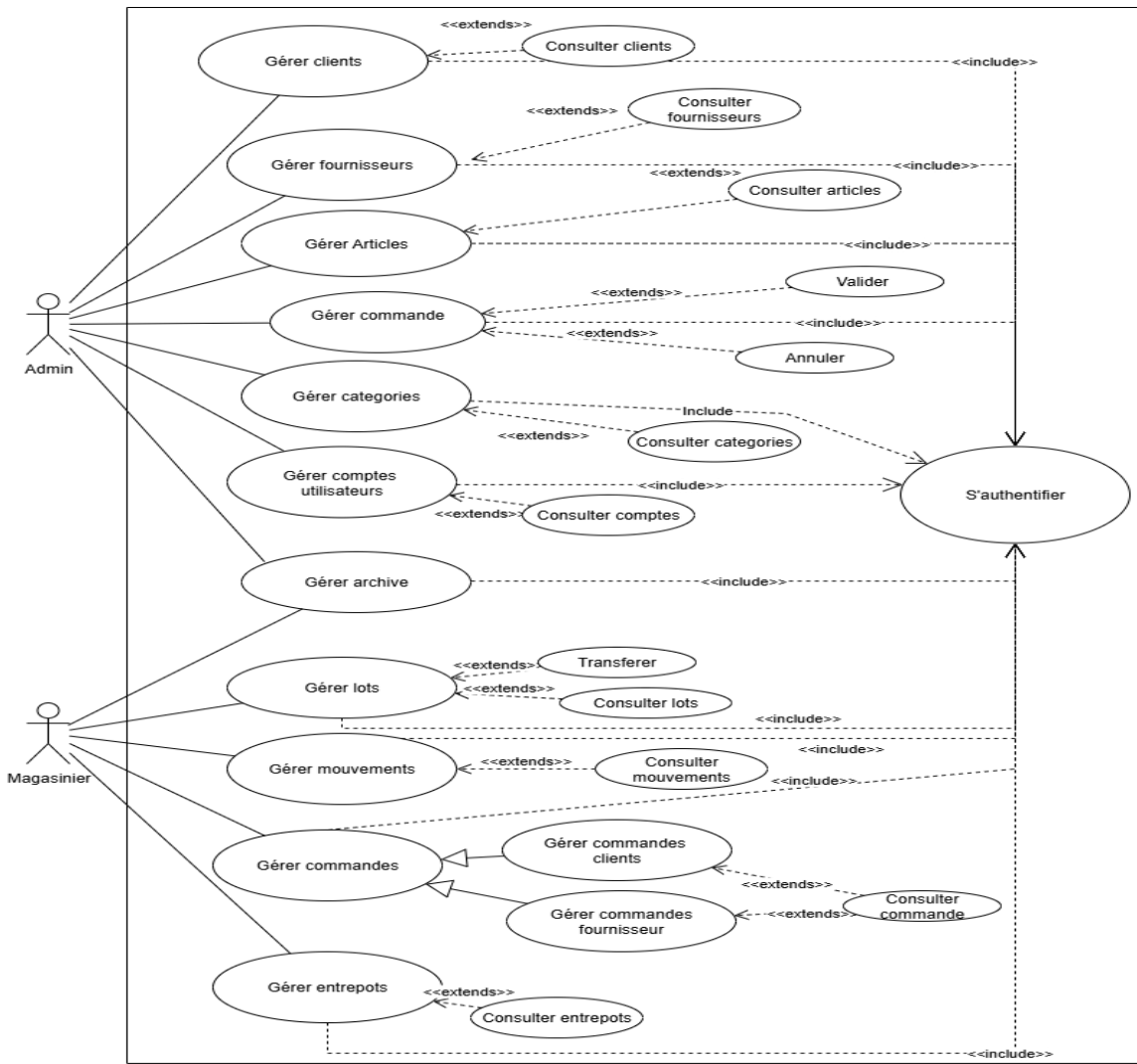


Figure 2.3 – Diagramme de cas d'utilisation global

2.5.5 Planification des releases

La planification des releases dans la méthode Scrum implique la structuration et l'ordonnancement de la livraison des fonctionnalités et des versions du produit, telle que motionné dans la figure 2.4.

La planification est réalisée avec l'outil scrumblr qui est une plateforme de gestion de projet conçue autour de la méthodologie scrum.

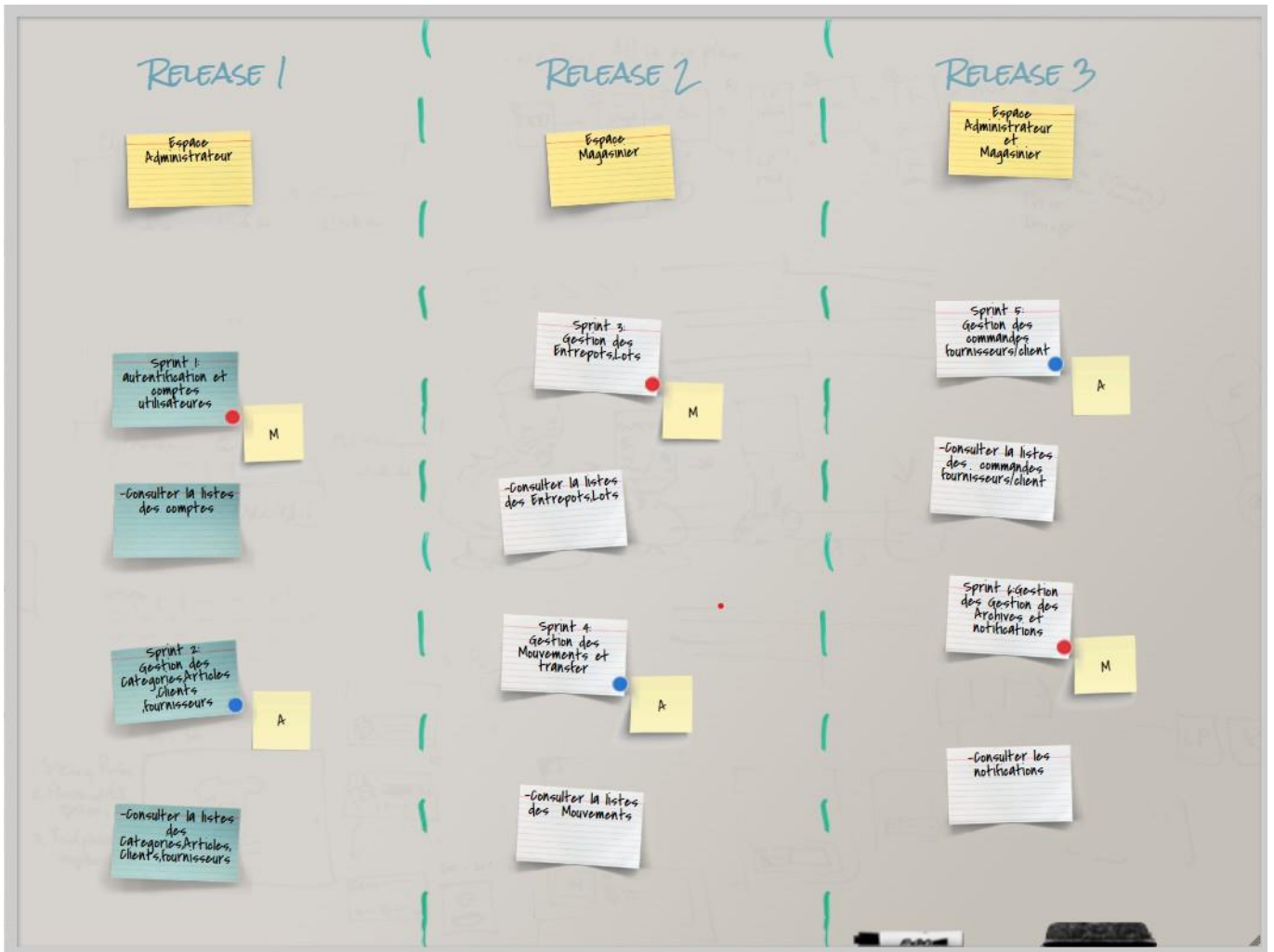


Figure 2.4 – Planification des releases

2.5.6 Backlog-Product (Carnet de produit)

Chaque histoire utilisateur (user story) est identifiée par un ID et est associée à un nom de fonctionnalité, une estimation du temps nécessaire pour sa réalisation, et une priorité définie par le Product Owner. Voici le backlog de notre futur système :

ID	Fonctionnalité	User story	Estimation	Priorité
1	Authentifier les utilisateurs	En tant qu'utilisateur (administrateur ou magasinier), je veux pouvoir m'authentifier sur le système en fournissant mon identifiant et mon mot de passe, afin de sécuriser l'accès et gérer les rôles.	4 Jours	Haute
2	Gérer les articles	En tant qu'administrateur, je veux pouvoir ajouter, modifier, supprimer et consulter des produits afin de maintenir un inventaire à jour.	9 Jours	Haute
3	Gérer les catégories	En tant qu'administrateur, je veux pouvoir organiser les produits par catégories pour faciliter la recherche et la gestion du stock.	8 Jours	Moyenne
4	Gérer les fournisseurs	En tant qu'administrateur, je veux pouvoir enregistrer et mettre à jour les informations des fournisseurs afin de faciliter la gestion des commandes et des réapprovisionnements.	10 jours	Moyenne
5	Gérer les mouvements de stock	En tant que magasinier, je veux enregistrer les entrées et sorties de stock afin d'assurer un suivi précis des produits et d'éviter les ruptures.	10 jours	Haute
6	Gérer les commandes clients	En tant que magasinier, je veux pouvoir créer et gérer les commandes des clients afin de suivre leur état (en attente, validée, livrée).	13 jours	Haute
7	Notifications	En tant qu'utilisateur, je veux recevoir des notifications en cas de rupture de stock, de nouvelle commande ou d'alerte critique afin d'agir rapidement.	5 Jours	Moyenne
8	Consulter Les Statistiques	En tant qu'administrateur, je veux visualiser des statistiques (niveau de stock, produits les plus vendus, commandes en cours) afin de mieux planifier les décisions.	7 Jours	Basse
9	Gestion des archives	En tant qu'administrateur, je veux pouvoir archiver des données (produits périmés, anciennes commandes) afin d'alléger le système tout en conservant l'historique.	7 Jours	Basse

Table 2.3 – Backlog-Product et User stories

2.6 Conclusion

Ce chapitre a établi les bases du projet en introduisant la méthode Scrum, l'utilisation de l'UML ainsi que l'architecture MVC, tout en définissant les besoins et exigences du système, les chapitres suivants seront consacrés à la présentation des différentes releases issues de ce processus de développement

CHAPITRE 3 : Présentation de premier Release

3.1 Introduction

Ce chapitre sera consacré à la présentation de deux sprints. chaque sprint sera détaillé à travers la présentation de la conception réaliser et la logique métier adopter au travers de développement.

3.2 Étude du premier sprint

Cette section présente l'étude du premier sprint, en détaillant les fonctionnalités prioritaires retenues, les tâches réalisées ainsi que les résultats obtenus à l'issue de cette itération.

Les fonctionnalités traitées et la durée

Ce sprint est programmé pour être réalisé dans 13 jours et est composé des fonctionnalités suivantes :

ID	Tâche	Description	Estimé	Responsable
1	Authentification	Connexion des utilisateurs (admin/magasinier) avec identifiant et mot de passe sécurisés.	4 jours	Djouder Massil
2	Ajouter utilisateur	Admin peut ajouter un utilisateur (nom, prénom, email, rôle, mot de passe).	3 jours	Abderrahmane Bougheriou
3	Modifier utilisateur	Admin peut modifier les infos (email, mot de passe, rôle, etc.).	2 jours	Abderrahmane Bougheriou
4	Désactiver utilisateur	Admin peut désactiver un utilisateur sans le supprimer définitivement.	2 jours	Djouder Massil
5	Tests	Vérifier toutes les fonctionnalités et corriger les anomalies détectées.	2 jours	Djouder Massil

Table 3.1 – Le sprint backlog du premier sprint

Diagramme de cas d'utilisation du premier sprint

Le diagramme de cas d'utilisation suivant représente les cas d'utilisations concernant le sprint 1 tel que mentionné dans la figure 3.1 :

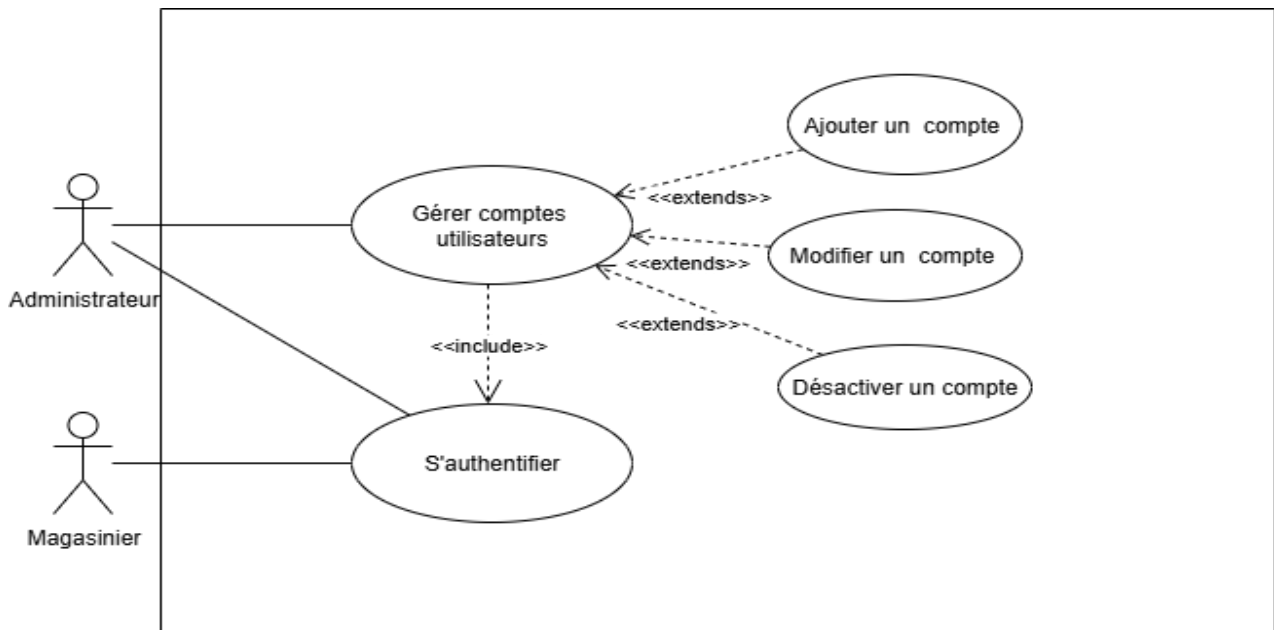


Figure 3.1 – Diagramme de CU de sprint 1

3.3 Les cas d'utilisations du premier sprint

Cette section expose les cas d'utilisation développés lors du premier sprint, en mettant en évidence les interactions principales entre les utilisateurs et le système.

3.3.1 Cas d'utilisation « s'authentifier »

Ce cas d'utilisation décrit le processus d'authentification, permettant à l'utilisateur de se connecter au système de manière sécurisée afin d'accéder à ses fonctionnalités.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « s'authentifier ».

CU :	S'authentifier
BUT :	Permettre à l'utilisateur de s'authentifier pour accéder à son espace personnel.
Acteur principal :	Utilisateur (Administrateur ou Magasinier)
Acteur secondaire :	/
Préconditions :	L'utilisateur dispose d'un compte actif dans le système.
Scénario nominal :	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page de connexion. 2. Il saisit son adresse e-mail et son mot de passe. 3. Le système vérifie les informations saisies. 4. L'utilisateur est redirigé vers son espace personnel (Dashboard).

Enchaînements alternatifs :	<p>3.a : Informations invalides</p> <ul style="list-style-type: none"> – Le système affiche un message d’erreur. – L’utilisateur ressaisit ses identifiants. – La séquence reprend au point 2. <p>3.b : Compte désactivé</p> <ul style="list-style-type: none"> – Le système indique que le compte est désactivé. – L’utilisateur ne peut pas accéder à son espace. – La séquence se termine.
Post-condition :	L’utilisateur est connecté à son espace personnel si l’authentification réussit.

Table 3.2 – La description textuelle de CU « s’authentifier ».

Diagramme de séquence du CU « S’authentifier »

La figure 3.2 représente le diagramme de séquence du CU « S’authentifier » :

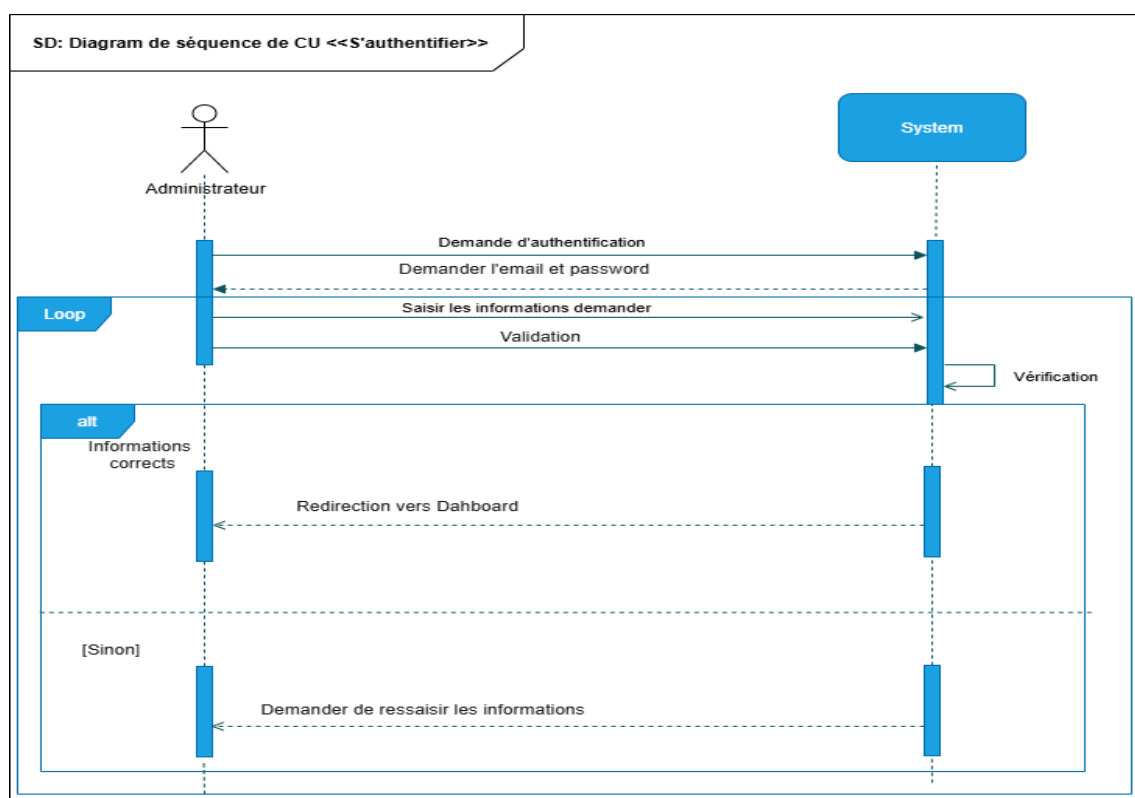


Figure 3.2 – Diagramme de séquence CU « S’authentifier »

3.3.2 Diagramme d'interaction de CU « S'authentifier »

La figure 3.3 représente le diagramme d'interaction du CU « S'authentifier » :

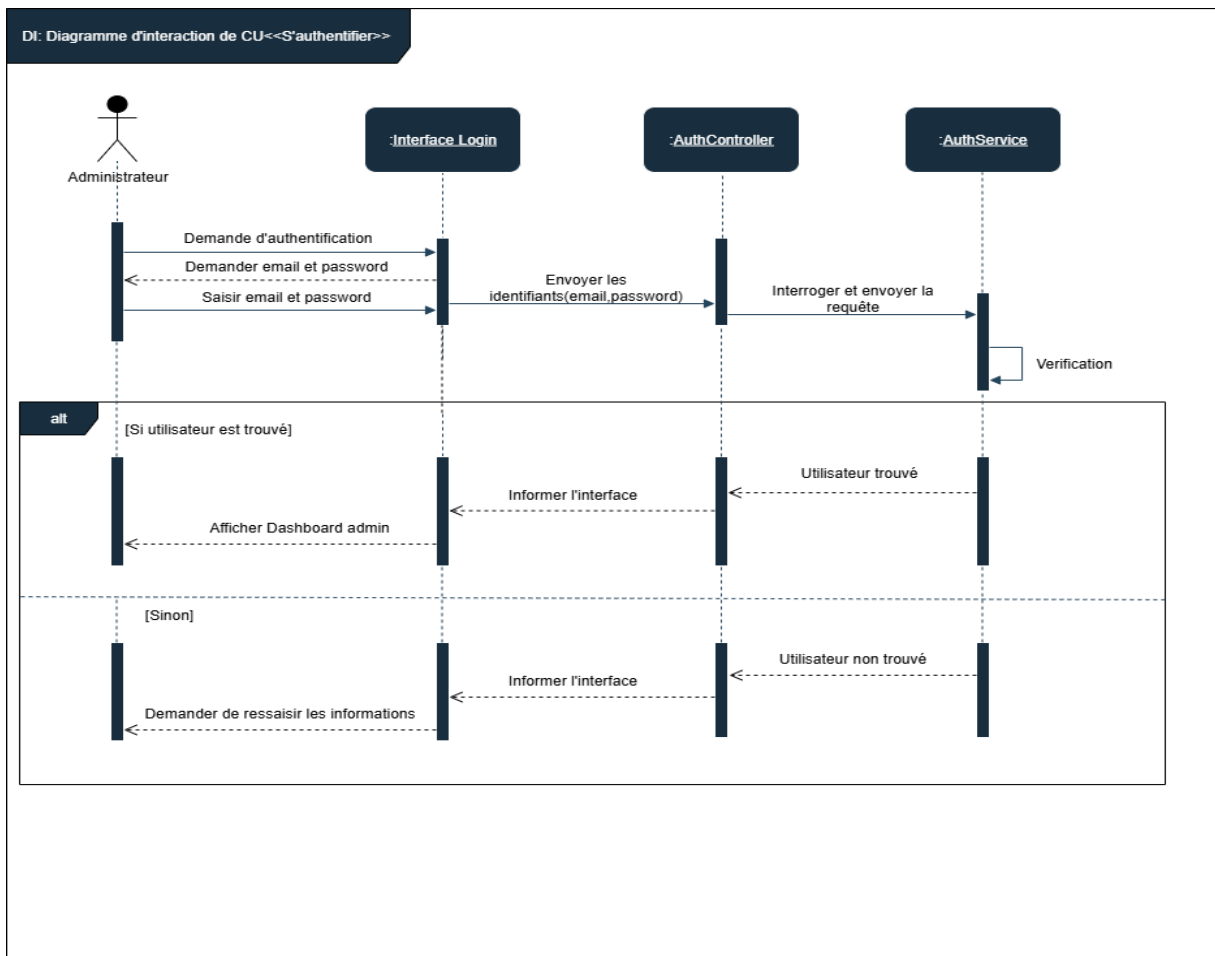


Figure 3.3 – Diagramme d'interaction de CU « S'authentifier »

3.3.3 Cas d'utilisation « Gérer comptes utilisateurs »

Ce cas d'utilisation illustre la gestion des comptes utilisateurs, permettant à l'administrateur de créer, modifier, désactiver ou consulter les informations liées aux utilisateurs du système.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer comptes utilisateurs ».

CU :	Gérer comptes utilisateurs
BUT :	Ce cas permet à l'administrateur d'ajouter, modifier ou désactiver un utilisateur dans le système.
Acteur principal :	Administrateur
Acteur secondaire :	/
Préconditions :	L'administrateur est authentifié.
Scénario nominal :	<p>Ajouter :</p> <ol style="list-style-type: none"> 1. L'administrateur choisit « Ajouter utilisateur ». 2. Il saisit les informations nécessaires. 3. Le système vérifie la validité des données. 4. L'utilisateur est ajouté. <p>Modifier :</p> <ol style="list-style-type: none"> 5. L'administrateur sélectionne un utilisateur existant. 6. Il modifie les informations souhaitées. 7. Le système vérifie les nouvelles données. 8. Les modifications sont enregistrées. <p>Désactiver :</p> <ol style="list-style-type: none"> 9. L'administrateur sélectionne un utilisateur. 10. Il confirme l'action. 11. Le compte est désactivé.
Enchaînements alternatifs :	<p>3.a: Données invalides lors de l'ajout</p> <ul style="list-style-type: none"> - Le système signale que les données sont invalides. - L'administrateur corrige et resoumet. - La séquence reprend au point 4. <p>7.a : Données invalides lors de la modification</p> <ul style="list-style-type: none"> - Le système signale que les données sont invalides. - L'administrateur corrige et resoumet. - La séquence reprend au point 8. <p>10.a : Annulation de la désactivation</p> <ul style="list-style-type: none"> - L'administrateur annule l'opération. - La séquence se termine sans modification.
Post-condition :	Les utilisateurs sont ajoutés, modifiés ou désactivés selon l'action effectuée.

Table 3.3 – La description textuelle de CU « Gérer compte utilisateur ».

Diagramme de séquence du CU « Gérer compte utilisateurs »

La figure 3.4 représente le diagramme de séquence du CU « S'authentifier » :

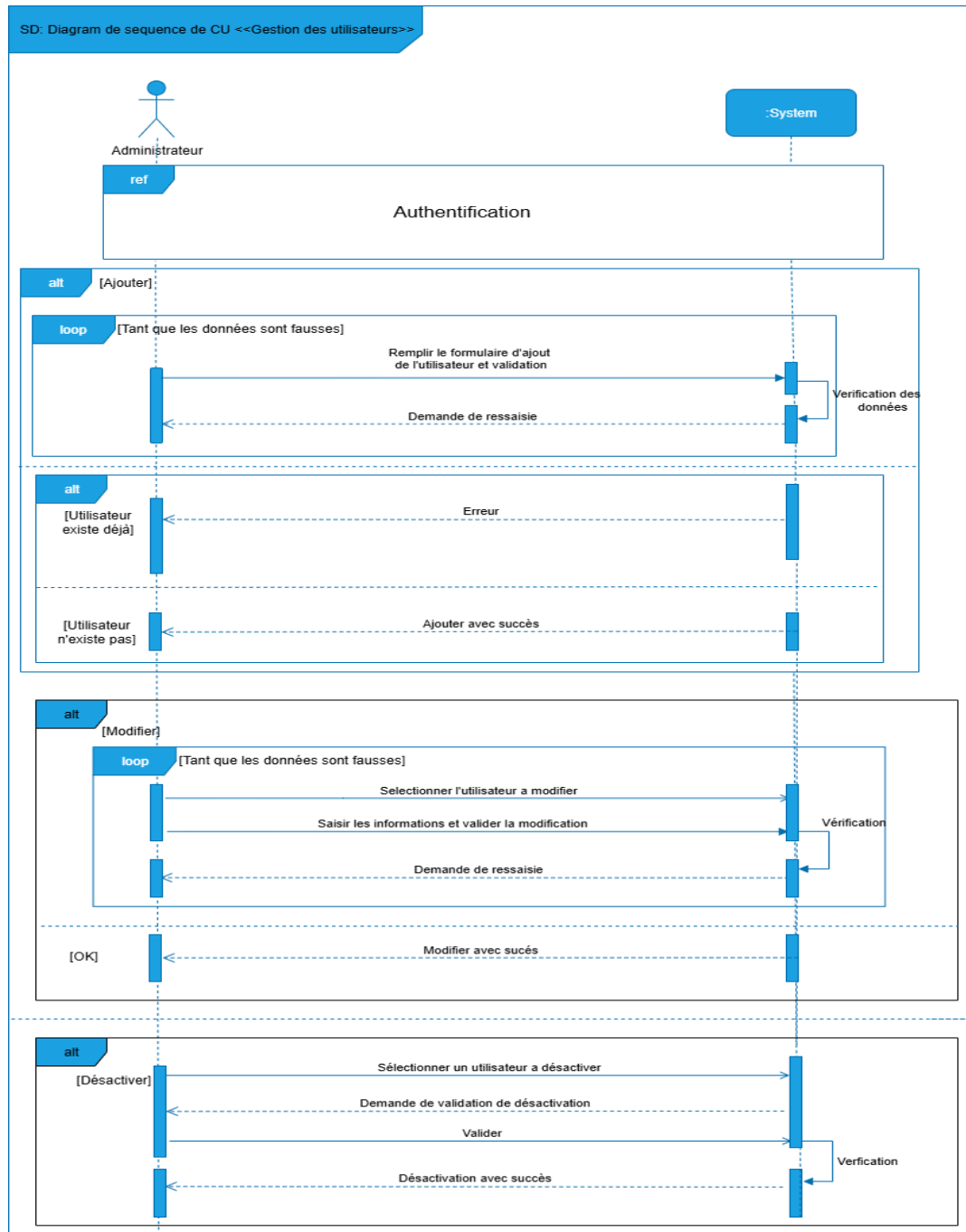


Figure 3.4 – Diagramme de séquence de CU « Gérer compte utilisateurs »

3.3.4 Diagramme de classe du premier sprint

Le diagramme de classe suivant représente les classes qui concernent le premier sprint :

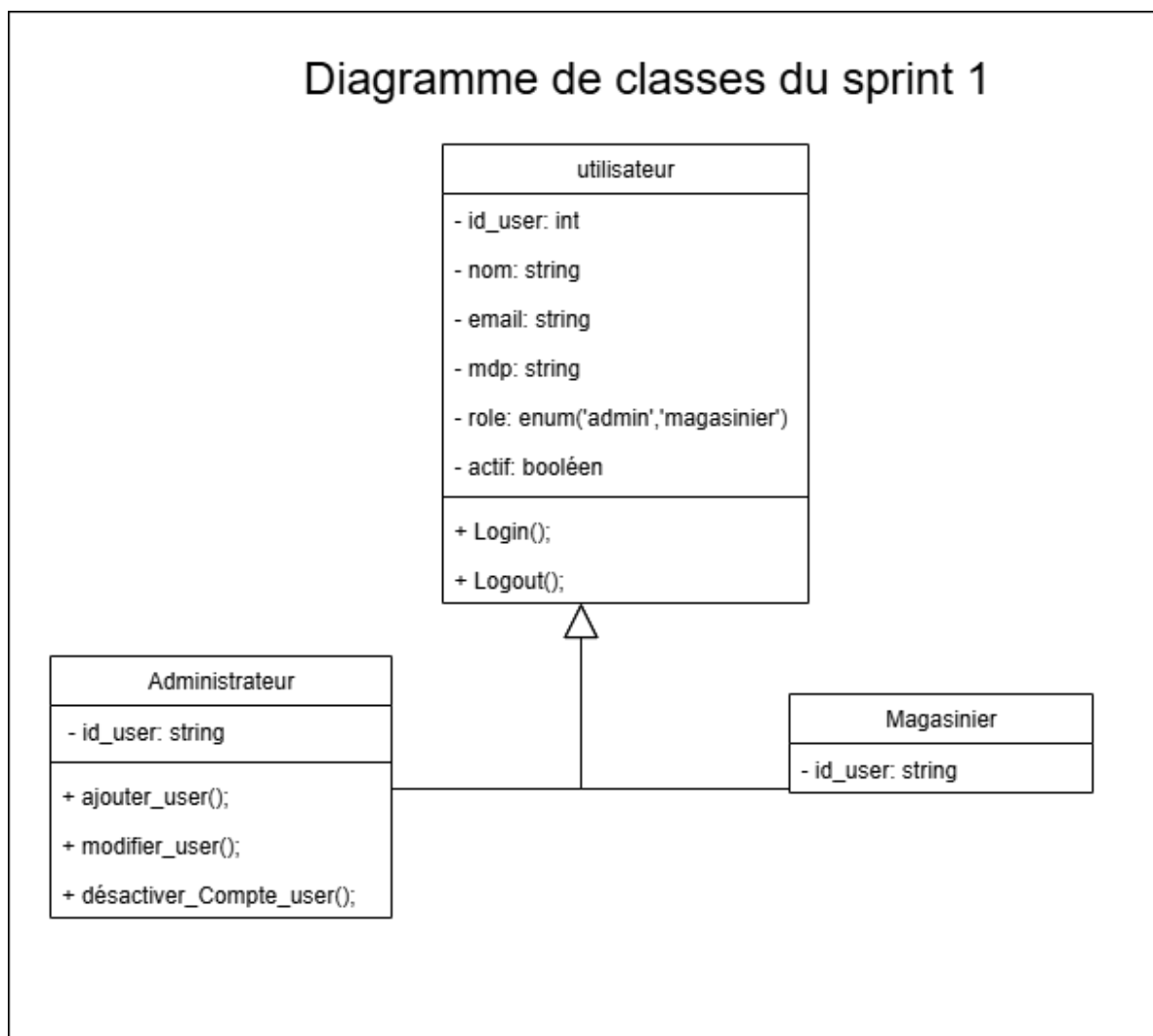


Figure 3.5 – Diagramme de Classe du premier sprint

3.4 Le schéma relationnel du premier sprint

Le schéma relationnel représente la structure logique de la base de données. Il permet de décrire les différentes entités du projet ainsi que les liens qui existent entre elles. Ce modèle met en évidence les tables, leurs attributs, ainsi que les relations établies à travers les clés primaires et étrangères. on le décrit par :

Utilisateur (#id_user, nom, email, mdp, role, actif).

3.4.1 Le dictionnaire de données du premier sprint

Cette section présente le dictionnaire de données élaboré lors du premier sprint, en détaillant la structure des tables, les attributs définis ainsi que leurs relations, afin de garantir une modélisation cohérente du système.

Le dictionnaire de données décrit de manière détaillée les différentes tables de la base, en précisant leurs attributs, leurs types ainsi que leurs rôles (clés primaires, clés étrangères, contraintes, etc.). Il constitue une référence technique essentielle pour la conception et la maintenance du système.

Table : Utilisateur	
#id_user	INT (11), clé primaire
nom	VARCHAR (25)
email	VARCHAR (30)
mdp	VARCHAR (25)
role	ENUM('admin','magasinier')
actif	Booléen

Table 3.4 – Le dictionnaire de données du premier sprint

3.5 Les interfaces graphiques du premier sprint

Les figures ci-dessus représentent les interfaces de premier sprint, tel que la figure (3.6) montre représentée l'interface de l'authentification, et la figure (3.7) représente l'interface de la gestion des utilisateurs

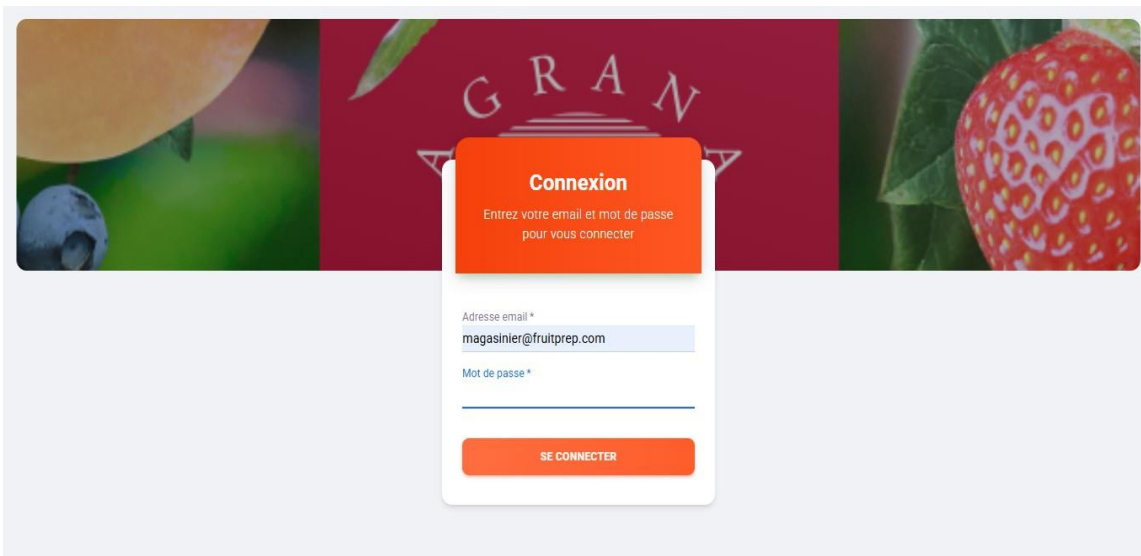


Figure 3.6 – Interface d'authentification

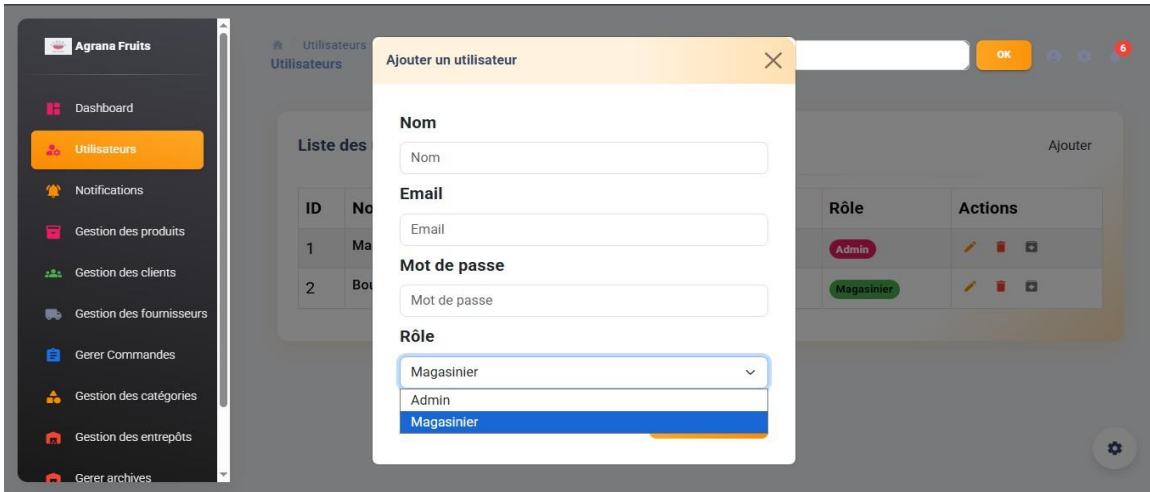


Figure 3.7 – Interface de gestion des comptes

3.6 Étude du deuxième sprint

Cette section présente l'étude du deuxième sprint, en détaillant les fonctionnalités prioritaires retenues, les tâches réalisées ainsi que les résultats obtenus à l'issue de cette itération.

3.6.1 Les fonctionnalités traitées et la durée

Ce sprint est programmé pour être réalisé dans 16 jours et est composé des fonctionnalités suivantes :

ID	Tâche	Description	Estimé	Responsable
1	Ajouter article	Ajouter un nouvel article avec nom, description, catégorie, etc..	3 jours	Abderrahmane Bougheriou
2	Modifier article	Modifier les informations d'un article existant	2 jours	Djouder Massil
3	Supprimer article	Supprimer un article qui n'intervient pas dans une (E/S).	2 jours	Djouder Massil
4	Archiver articles	Archiver des articles périmés ou inutilisés afin de garder l'inventaire organisé.	2 jours	Abderrahmane Bougheriou
5	Ajouter client	Ajouter un nouveau client avec ses informations de contact pour suivre ses commandes	3 jours	Abderrahmane Bougheriou
6	Modifier client	Modifier les informations d'un client (adresse, téléphone, etc.) pour garder une base de données à jour.	2 jours	Abderrahmane Bougheriou

7	Archiver client	Archiver les clients inactifs ou supprimés afin de maintenir une gestion propre des données clients.	2 jours	Abderrahmane Bougheriou
8	Tests	Vérifier toutes les fonctionnalités et corriger les anomalies détectées.	2 jours	Djouder Massil

Table 3.5 – Le sprint backlog du deuxième sprint

3.6.2 Diagramme de cas d'utilisation du deuxième sprint

Le diagramme de cas d'utilisation suivant représente les cas d'utilisations concernant le sprint 2 comme une partie séparée :

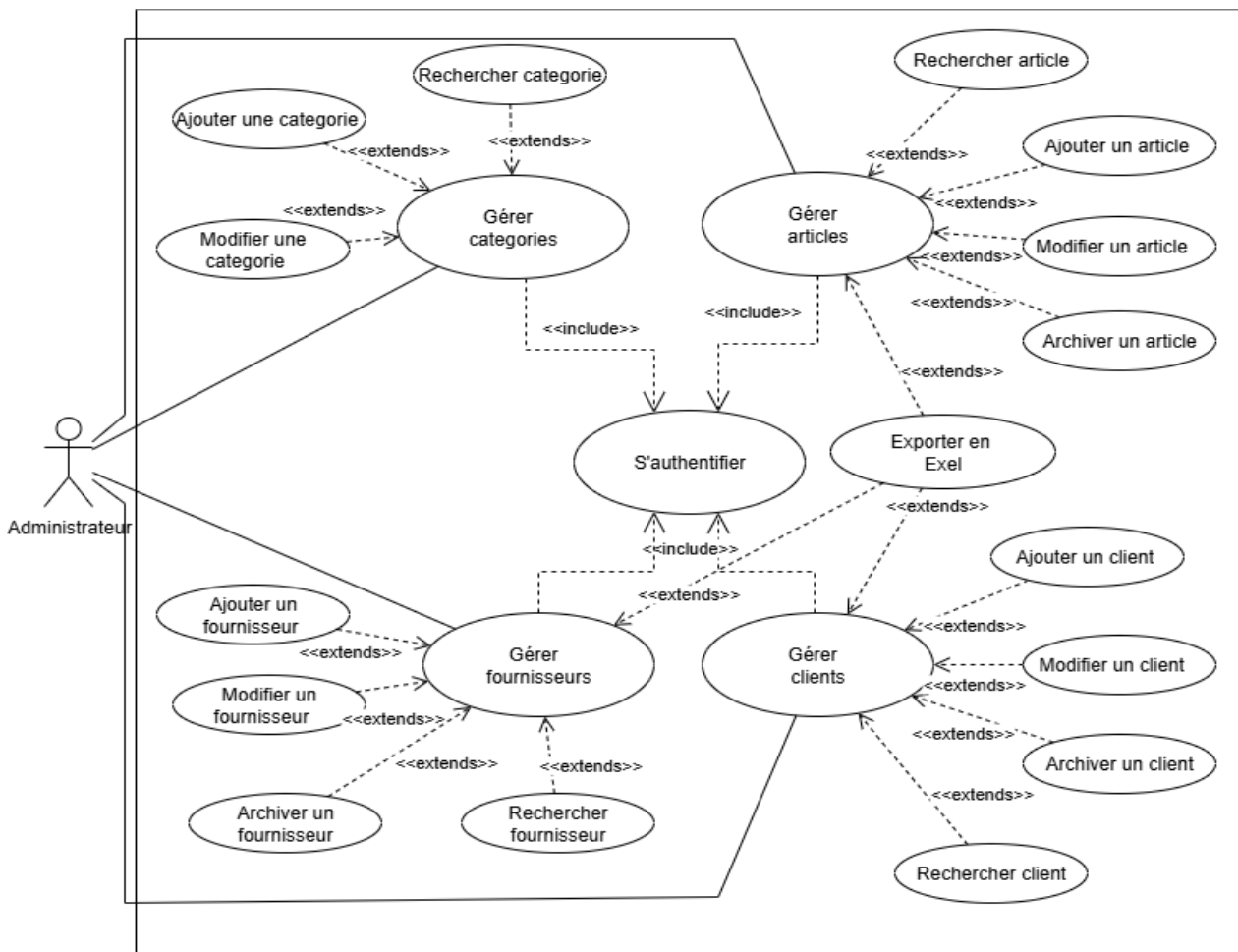


Figure 3.8 – Use case diagramme sprint 2

3.7 Les cas d'utilisations du deuxième sprint

Cette section expose les cas d'utilisation développés lors du deuxième sprint, en mettant en évidence les interactions principales entre les utilisateurs et le système.

3.7.1 Cas d'utilisation « Gérer articles »

Ce cas d'utilisation décrit la gestion des articles, offrant à l'administrateur la possibilité d'ajouter, modifier, supprimer, archiver ou consulter les informations relatives aux articles du stock.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer articles ».

CU :	Gérer articles
BUT :	Permettre à l'administrateur d'ajouter, modifier, supprimer (si jamais utilisé) ou archiver un article.
Acteur principal :	Administrateur
Acteur secondaire :	/
Préconditions :	L'administrateur est authentifié.
Scénario nominal :	<p>Ajouter :</p> <ol style="list-style-type: none"> 1. Choisit « Ajouter article ». 2. Saisit les informations. 3. Le système vérifie. 4. L'article est ajouté. <p>Modifier :</p> <ol style="list-style-type: none"> 5. Sélectionne un article. 6. Modifie les informations. 7. Le système vérifie. 8. Les modifications sont enregistrées. <p>Supprimer :</p> <ol style="list-style-type: none"> 9. Sélectionne un article. 10. Le système vérifie que l'article n'a jamais été utilisé (mouvements E/S). 11. Si condition respectée, l'administrateur confirme. 12. L'article est supprimé. <p>Archiver :</p> <ol style="list-style-type: none"> 13. Sélectionne un article. 14. Confirme l'action. 15. L'article est archivé.
Enchaînements alternatifs :	<p>3.a -Données invalides (ajout)</p> <ul style="list-style-type: none"> – Le système signale l'erreur. – L'administrateur corrige et resoumet. – Reprise au point 4. <p>7.a - Données invalides (modification)</p> <ul style="list-style-type: none"> – Le système signale l'erreur. – L'administrateur corrige et resoumet. – Reprise au point 8.

	<p>10.a - Article déjà utilisé (suppression impossible)</p> <ul style="list-style-type: none"> - Le système indique que la suppression est impossible. - Retour à l'interface sans suppression. <p>14.a - Annulation d'archivage</p> <ul style="list-style-type: none"> - L'administrateur annule. - Fin sans modification.
--	---

Table 3.6 – La description textuelle de CU « Gérer articles ».

Diagramme de séquence de CU « Gérer articles »

La figure ci-après présente le diagramme de séquence correspondant au CU « Gérer articles ».

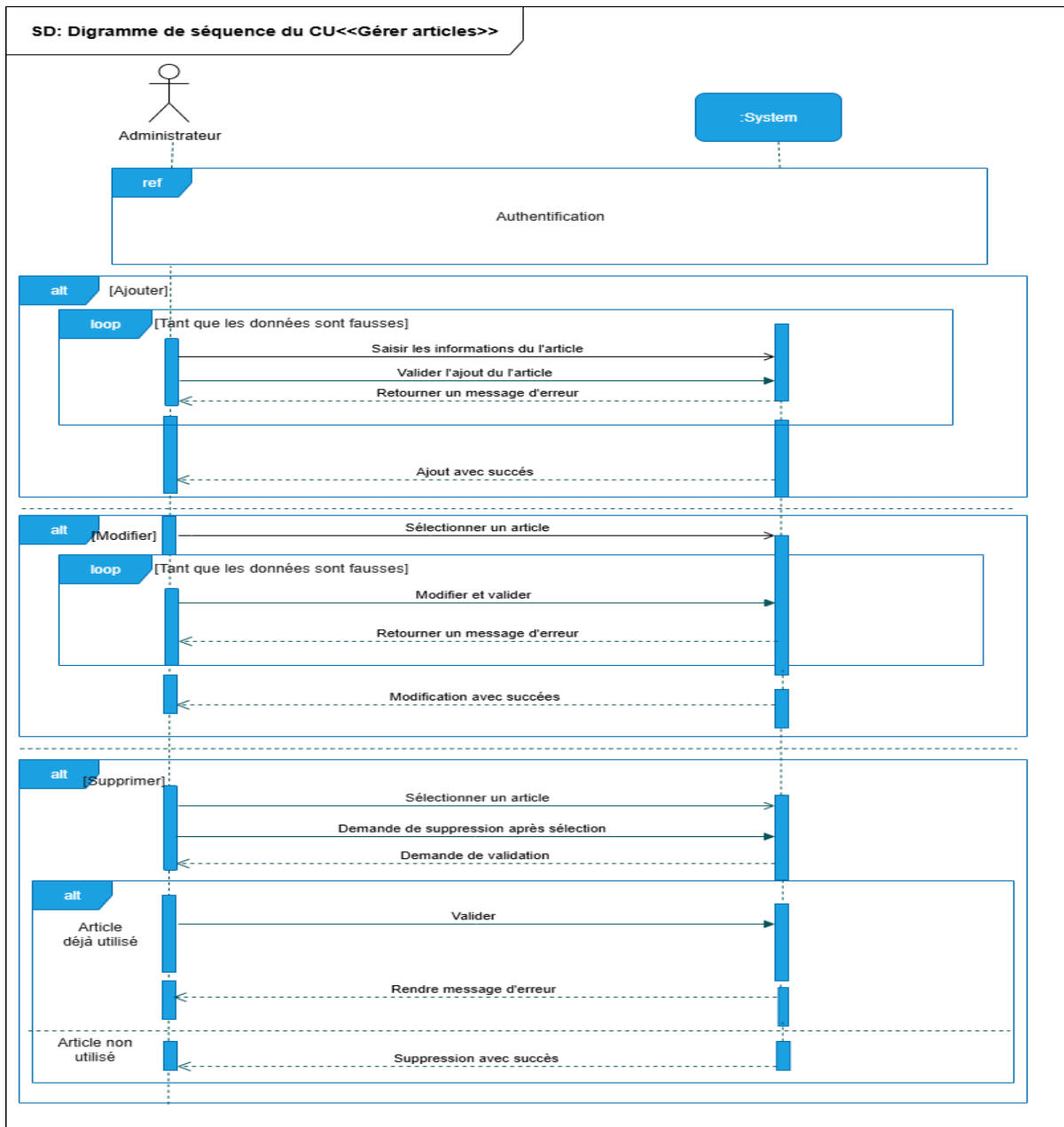


Figure 3.9 – Diagramme de séquence de CU « Gérer articles »

3.7.2 Diagramme d'interaction de CU « Gérer articles »

La figure ci-dessous représente le diagramme d'interaction correspondant au de CU « Gérer articles ».

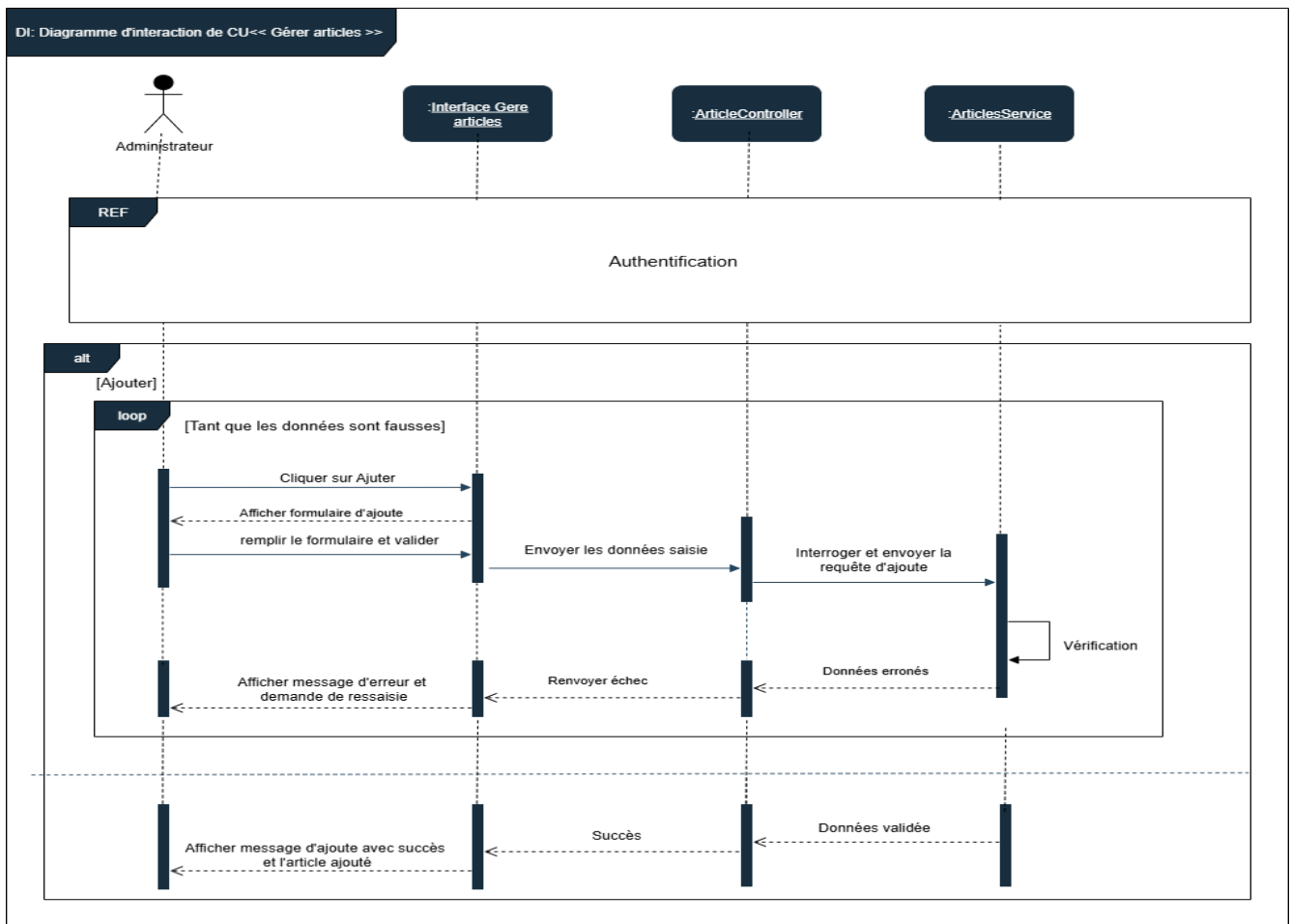


Figure 3.10 – Diagramme d'interaction de CU « Gérer articles »

3.7.3 Cas d'utilisation « Gérer clients »

Ce cas d'utilisation concerne la gestion des clients, permettant à l'utilisateur d'ajouter, modifier, supprimer, archiver ou consulter les informations relatives aux clients de l'entreprise.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer clients ».

CU :	Gérer clients
BUT :	Permettre à l'administrateur d'ajouter, modifier ou désactiver/archiver un client.
Acteur principal :	Administrateur
Acteur secondaire :	/
Préconditions :	L'administrateur est authentifié.

Scénario nominal :	<p>Ajouter :</p> <ol style="list-style-type: none"> 1. Choisit « Ajouter client ». 2. Saisit les informations. 3. Le système vérifie. 4. Le client est ajouté. <p>Modifier :</p> <ol style="list-style-type: none"> 5. Sélectionne un client. 6. Modifie les informations. 7. Le système vérifie. 8. Les modifications sont enregistrées. <p>Désactiver/Archiver :</p> <ol style="list-style-type: none"> 9. Sélectionne un client. 10. Confirme l'action. 11. Le client est archivé.
Enchaînements alternatifs :	<p>3.a : Données invalides lors de l'ajout</p> <ul style="list-style-type: none"> - Le système indique que les données sont invalides. - L'administrateur corrige et resoumet. - La séquence nominale reprend au point 4. <p>7.a : Données invalides lors de la modification</p> <ul style="list-style-type: none"> - Le système indique que les données sont invalides. - L'administrateur corrige et resoumet. - La séquence nominale reprend au point 8. <p>10.a : Annulation de désactivation/archivage</p> <ul style="list-style-type: none"> - L'administrateur annule l'opération. - La séquence se termine sans modification.
Post-condition :	Les informations clients sont mises à jour (ajout, modification ou archivage).

Table 3.7 – La description textuelle de CU « Gérer clients ».

Diagramme de séquence de CU « Gérer clients »

La figure ci-après présente le diagramme de séquence correspondant au CU « Gérer clients ».

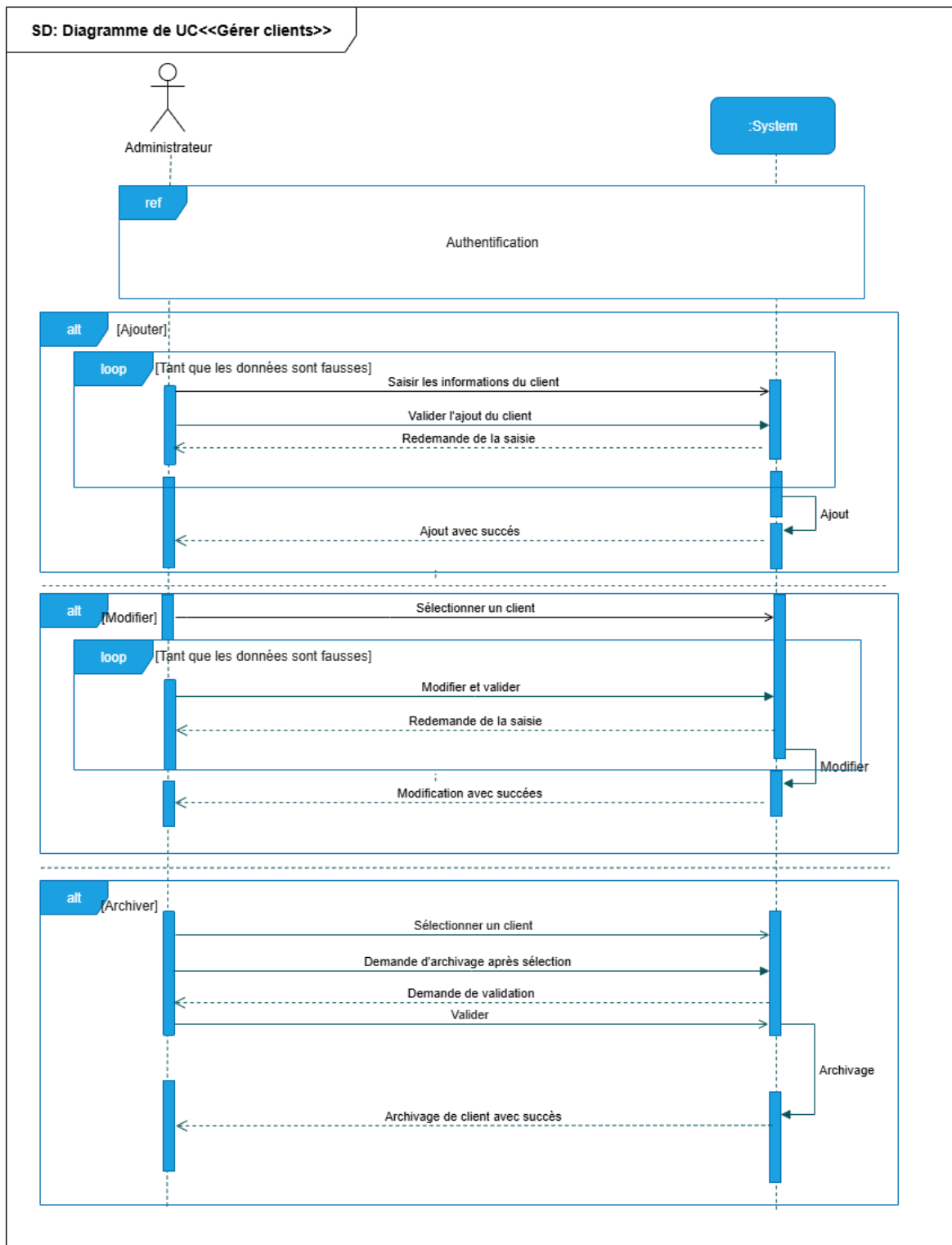


Figure 3.11 – Diagramme de séquence de CU « Gérer clients »

3.7.4 Diagramme d'interaction de CU « Gérer clients »

La figure ci-dessous représente le diagramme d'interaction correspondant au de CU « Gérer clients ».

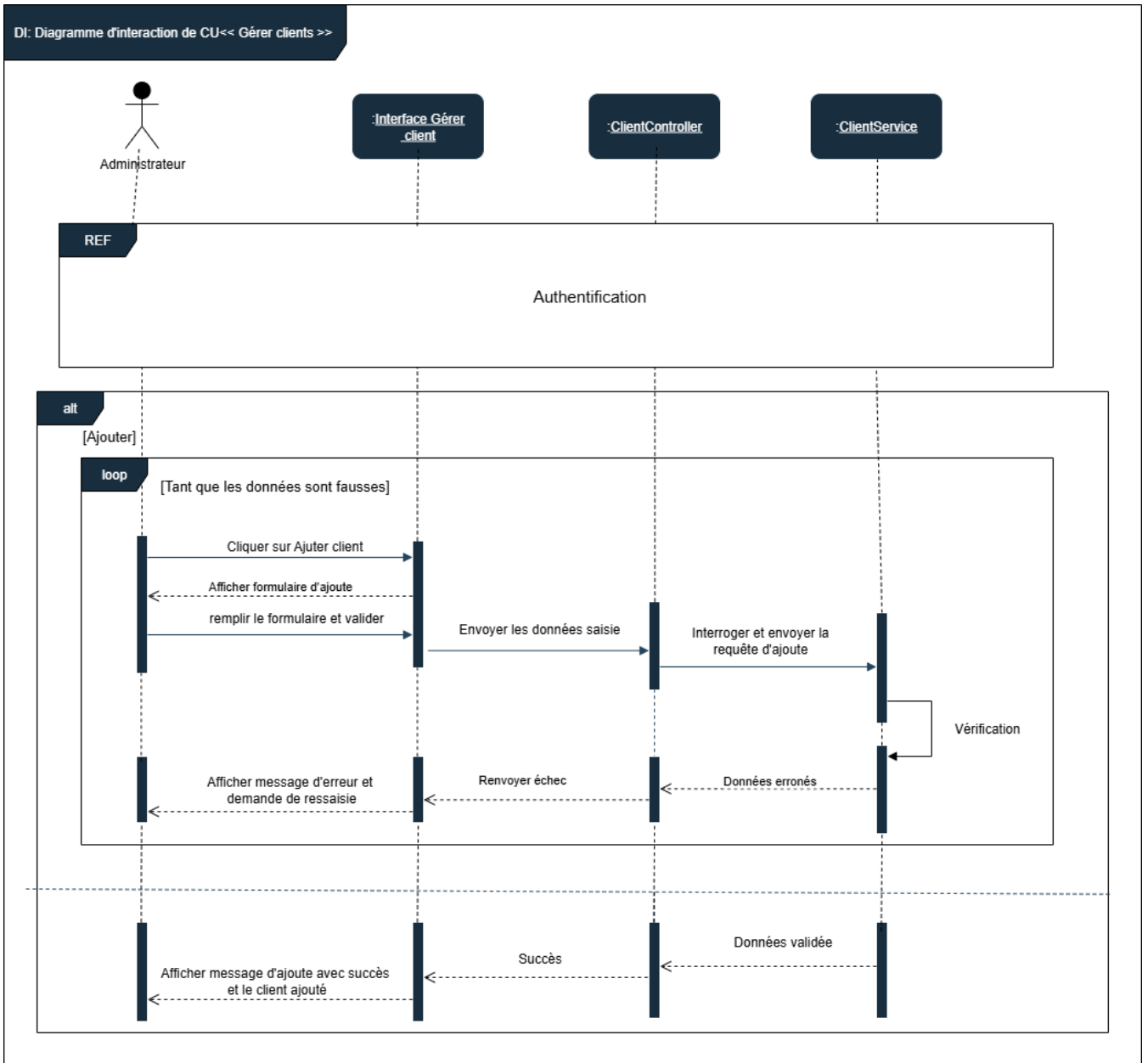


Figure 3.12 – Diagramme d'interaction de CU « Gérer clients »

3.7.5 Diagramme de classe du deuxième sprint

Le diagramme de classe suivant représente les classes qui concernent le sprint 2 :

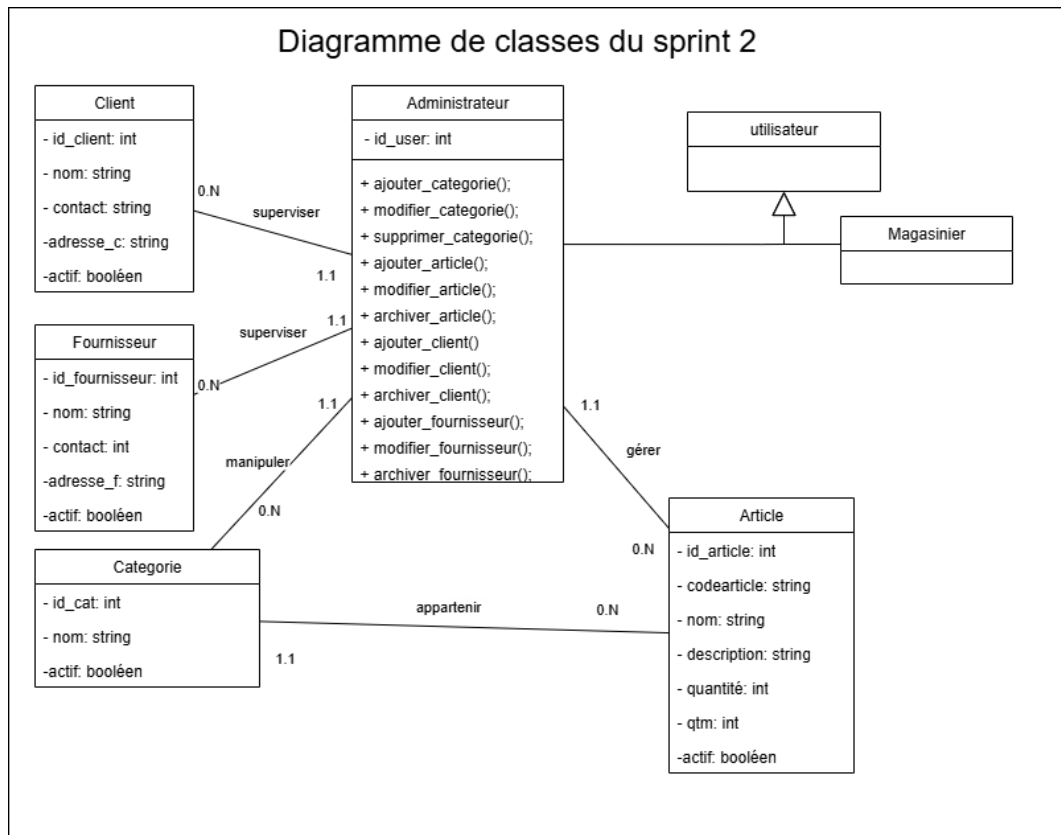


Figure 3.13 – Diagramme de Classe du deuxième sprint

3.8 Le schéma relationnel du deuxième sprint

On le décrit par :

Utilisateur (id_user, nom, email, mdp, role, actif). Article (id_article, CodeArticle, nom, description, quantité, qtm, actif, #*id_cat*).

Client (id_client, nom, contact, adresse_c, actif).

Catégorie (id_cat, nom, actif). Fournisseur (id_fournisseur, nom, contact, adresse_f, actif).

3.8.1 Le dictionnaire de données du deuxième sprint

Cette section présente le dictionnaire de données élaboré lors du deuxième sprint, en détaillant la structure des tables, les attributs définis ainsi que leurs relations, afin de garantir une modélisation cohérente du système.

Table : Utilisateur	
id_user	INT (11), clé primaire
nom	VARCHAR (25)
email	VARCHAR (30)
mdp	VARCHAR (25)
role	ENUM('admin', 'magasinier')

Table : Articles	
id_article	INT (11), clé primaire
CodeArticle	VARCHAR (25)
nom	VARCHAR (25)
description	TEXT
quantite	INT (11)
QTM	INT (11)
actif	BOOLEAN
id_cat	INT (11), clé étrangère vers Categorie

Table : Categorie	
id_cat	INT (11), clé primaire
nom	VARCHAR (25)
actif	BOOLEAN

Table : Client	
id_client	INT (11), clé primaire
nom	VARCHAR (25)
contact	VARCHAR (25)
adresse_c	VARCHAR (25)
actif	BOOLEAN

Table : Fournisseur	
id_fournisseur	INT (11), clé primaire
nom	VARCHAR (25)
contact	VARCHAR (25)
adresse_f	VARCHAR (25)
actif	BOOLEAN

Table 3.8 – Le dictionnaire de données du deuxième sprint

3.9 les interfaces graphiques du deuxième sprint

les figures ci-dessus représentent quelques interfaces de sprint 2, tel que la figure 3.14 montre l'interface de gestion des produits, et la figure 3.15 illustre l'interface de gestion des clients.

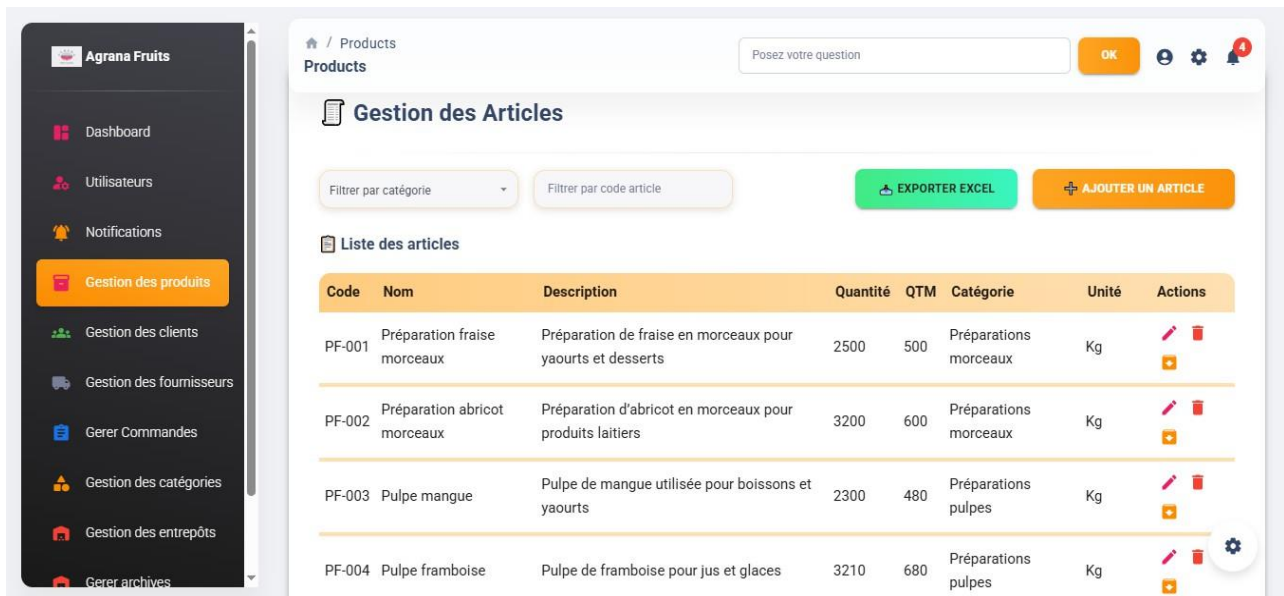


Figure 3.14 – Interface de gestion des produits

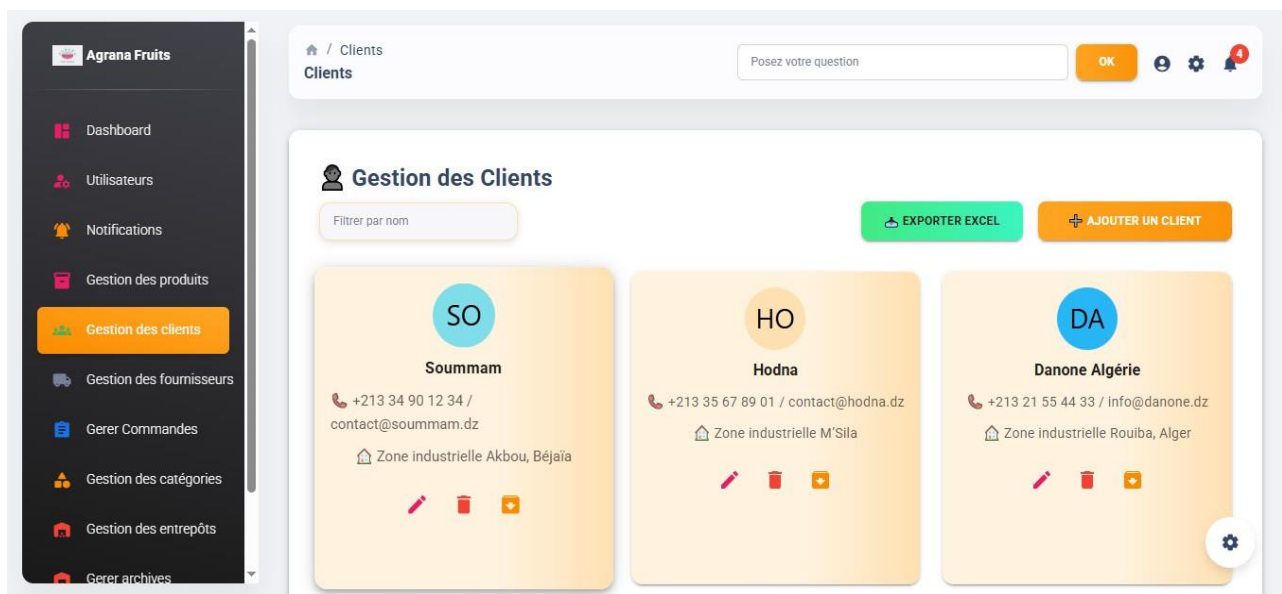


Figure 3.15 – Interface de gestion des clients

3.10 Conclusion

Ce release pose les bases de l'application en permettant de gérer les clients, les articles, les comptes utilisateurs et l'authentification, constituant ainsi ses premières fonctionnalités essentielles.

CHAPITRE 4 : Présentation de deuxième Release

4.1 Introduction

Dans ce chapitre, nous traiterons deux sprints. Pour chacun, nous présenterons la conception et les fonctionnalités réalisées, ainsi que des captures des interfaces développées.

4.2 Étude du troisième sprint

Cette section présente l'étude du troisième sprint, en détaillant les fonctionnalités prioritaires retenues, les tâches réalisées ainsi que les résultats obtenus à l'issue de cette itération.

4.2.1 *les fonctionnalités traitées et la durée*

Ce sprint est programmé pour être réalisé dans 16 jours et est composé des fonctionnalités

suivantes :

I D	Tâche	Description	Estimé	Responsable
1	Ajouter un entrepôt	Ajouter un nouvel entrepôt avec des informations comme le nom, l'emplacement et les capacités.	3 jours	Djouder Massil
2	Modifier un entrepôt	Modifier les informations d'un entrepôt existant pour mettre à jour les détails (nom, emplacement, capacité).	2 jours	Abderrahmane Bougheriou
3	Archiver un entrepôt	Archiver un entrepôt obsolète ou inutilisé pour maintenir la base de données propre.	3 jours	Djouder Massil
4	Ajouter un lot	Ajouter un nouveau lot avec des informations sur les produits, la quantité et les dates.	3 jours	Djouder Massil
5	Modifier un lot	Modifier les informations d'un lot existant, telles que la quantité ou les produits associés.	2 jours	Abderrahmane Bougheriou
6	Archiver un lot	Archiver un lot périmé ou inutilisé pour une gestion plus claire des stocks.	2 jours	Djouder Massil
7	Rechercher un lot	Rechercher un lot spécifique en fonction de critères comme le produit, la date ou la quantité.	2 jours	Abderrahmane Bougheriou
8	Exporter un lot en Excel	Exporter les informations d'un lot vers un fichier Excel pour une analyse ou une gestion externe.	1 jours	Abderrahmane Bougheriou
9	Tests	Vérifier toutes les fonctionnalités et corriger les anomalies détectées.	1 jours	Djouder Massil

Table 4.1 – Le sprint backlog du troisième sprint

4.2.2 Diagramme de cas d'utilisation de troisième sprint

Le diagramme de cas d'utilisation suivant représente les cas d'utilisations concernant le sprint 3 comme une partie séparée :

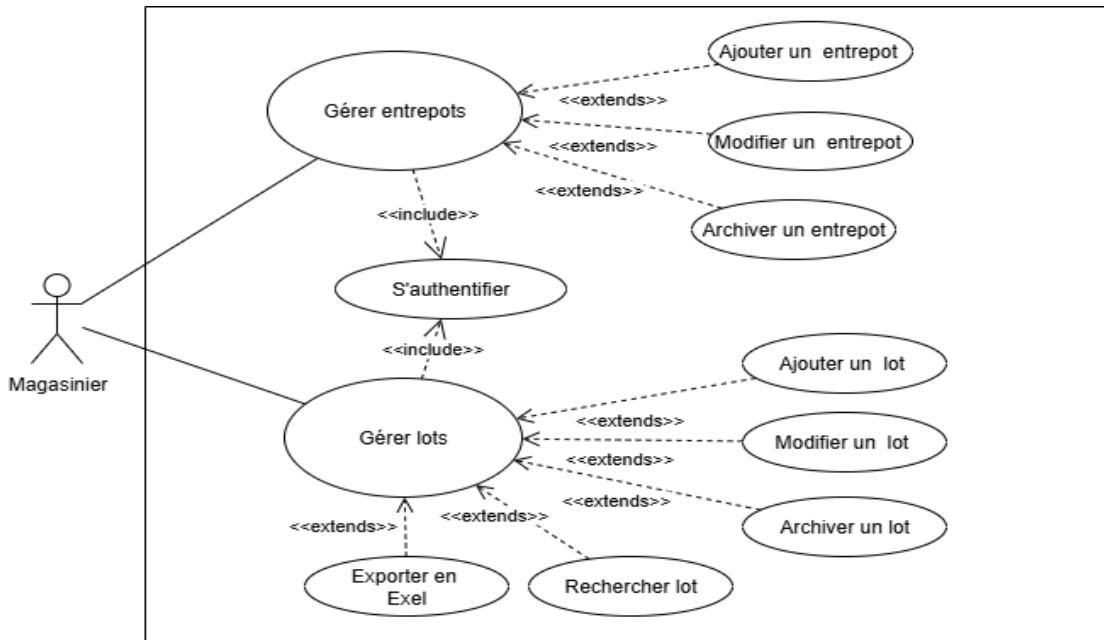


Figure 4.1 – Diagramme de cas d'utilisation du sprint 3

4.3 Les cas d'utilisations de troisième sprint

Cette section expose les cas d'utilisation développés lors du premier sprint, en mettant en évidence les interactions principales entre les utilisateurs et le système.

4.3.1 Cas d'utilisation « Gérer lots »

Ce cas d'utilisation décrit la gestion des lots, permettant d'ajouter, modifier, archiver ou consulter les informations relatives aux lots afin d'assurer la traçabilité des produits dans le stock.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer lots ».

CU :	Gérer lots
BUT :	Permettre au magasinier d'ajouter, modifier ou archiver un lot.
Acteur principal :	Magasinier

Acteur secondaire :	/
Préconditions :	Le magasinier est authentifié.
Scénario nominal :	<p>Ajouter un lot :</p> <ol style="list-style-type: none"> 1. Le magasinier choisit « Ajouter lot ». 2. Il saisit les informations nécessaires. 3. Le système vérifie les données. 4. Le lot est ajouté. <p>Modifier un lot :</p> <ol style="list-style-type: none"> 5. Le magasinier sélectionne un lot existant. 6. Il modifie les informations. 7. Le système vérifie les nouvelles données. 8. Les modifications sont enregistrées. <p>Archiver un lot :</p> <ol style="list-style-type: none"> 9. Le magasinier sélectionne un lot. 10. Il confirme l'action. 11. Le lot est archivé.
Enchaînements alternatifs :	<p>3.a: Données invalides lors de l'ajout</p> <ul style="list-style-type: none"> - Le système indique que les données sont invalides. - Le magasinier corrige et resoumet. - La séquence reprend au point 4. <p>7.a : Données invalides lors de la modification</p> <ul style="list-style-type: none"> - Le système indique que les données sont invalides. - Le magasinier corrige et resoumet. - La séquence reprend au point 8. <p>10.a : Annulation de l'archivage</p> <ul style="list-style-type: none"> - Le magasinier annule l'opération. - La séquence se termine sans modification.
Post-condition :	Les lots sont ajoutés, modifiés ou archivés selon l'action effectuée.

Table 4.2 – La description textuelle du CU « Gérer lots »

Diagramme de séquence de CU « Gérer lots »

La figure ci-après présente le diagramme de séquence correspondant au CU « Gérer lots ».

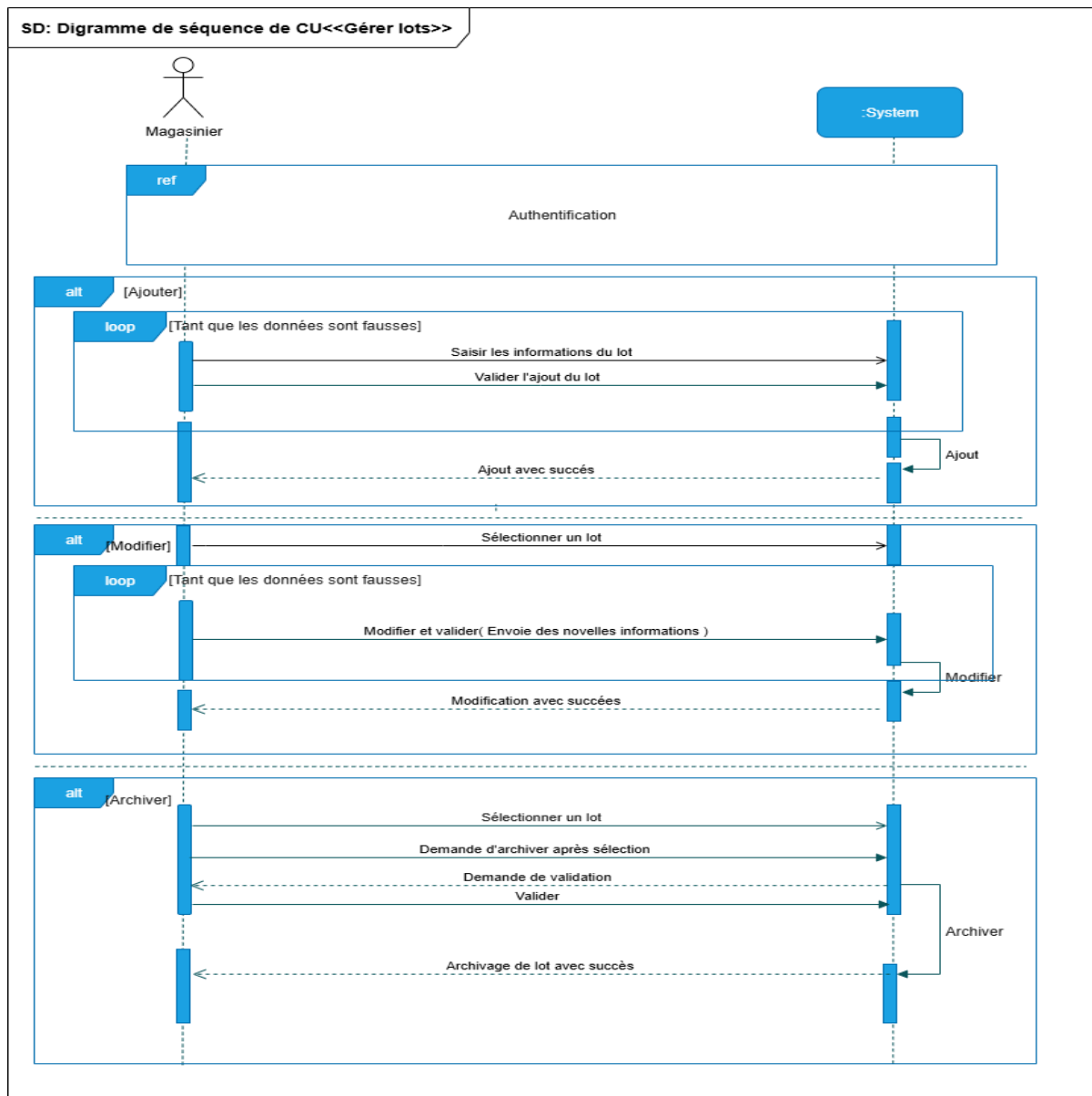


Figure 4.2 – Diagramme de séquence Gérer lots

4.3.2 Diagramme d'interaction de CU « Gérer lots »

La figure ci-dessous représente le diagramme d'interaction correspondant au de CU « Gérer lots ».

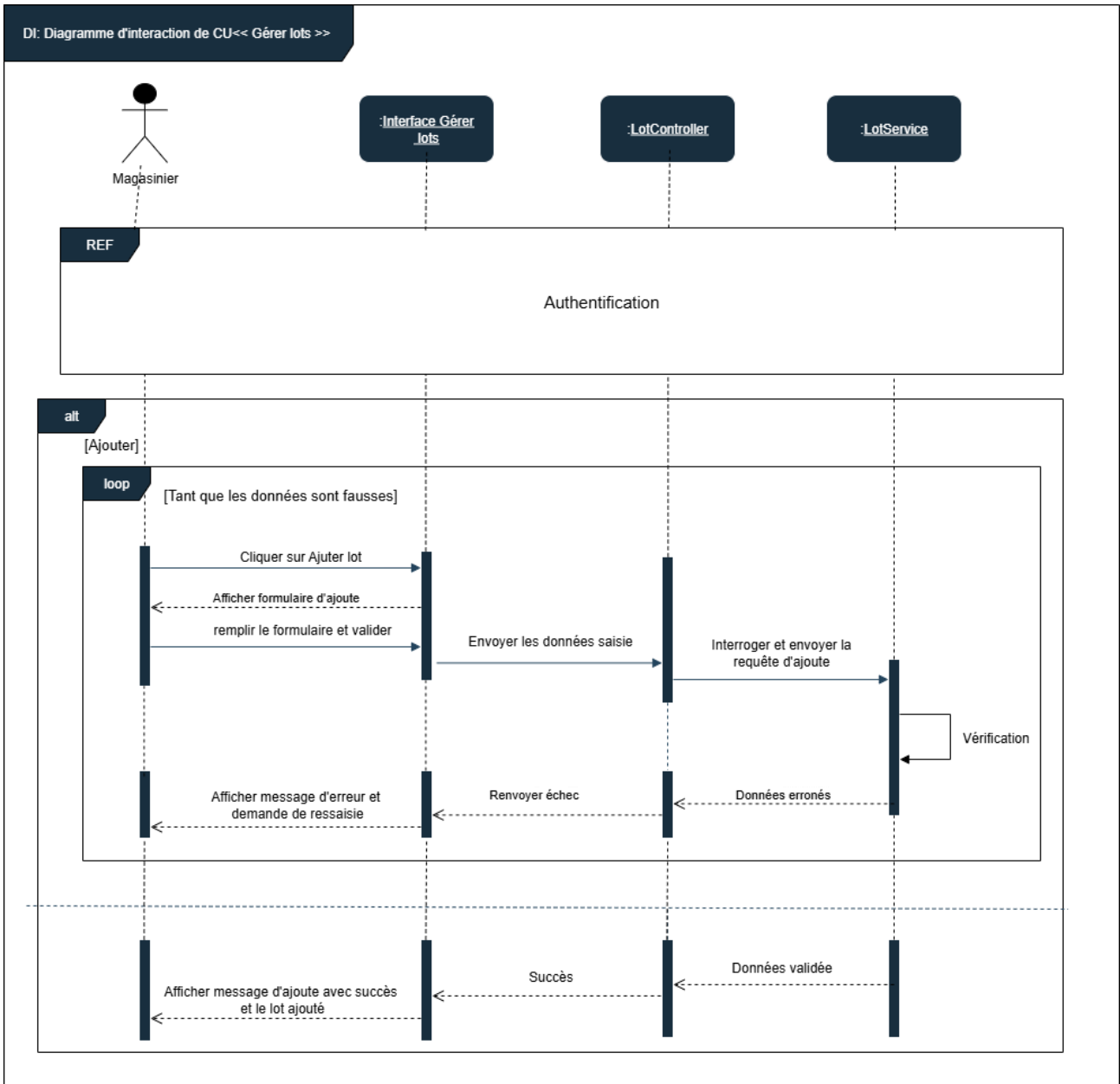


Figure 4.3 – Diagramme d'interaction de CU « Gérer lots »

4.3.3 Cas d'utilisation « Gérer entrepôts »

Ce cas d'utilisation présente la gestion des entrepôts, offrant la possibilité d'ajouter, modifier, supprimer ou consulter les informations relatives aux différents lieux de stockage de l'entreprise.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer entrepôts ».

CU :	Gérer entrepôts
BUT :	Permettre à l'administrateur d'ajouter, modifier ou archiver un entrepôt.
Acteur principal :	Administrateur
Acteur secondaire :	/
Préconditions :	L'administrateur est authentifié.
Scénario nominal :	Ajouter un entrepôt : 1. L'administrateur choisit « Ajouter entrepôt ». 2. Il saisit les informations nécessaires. 3. Le système vérifie les données. 4. L'entrepôt est ajouté. Modifier un entrepôt : 5. L'administrateur sélectionne un entrepôt existant. 6. Il modifie les informations. 7. Le système vérifie les nouvelles données. 8. Les modifications sont enregistrées. Archiver un entrepôt : 9. L'administrateur sélectionne un entrepôt. 10. Il confirme l'action. 11. L'entrepôt est archivé.
Enchaînements alternatifs :	3.a : Données invalides lors de l'ajout - Le système indique que les données sont invalides. - L'administrateur corrige et resoumet. - La séquence reprend au point 4. 7.a : Données invalides lors de la modification - Le système indique que les données sont invalides. - L'administrateur corrige et resoumet. - La séquence reprend au point 8. 10.a : Annulation de l'archivage - L'administrateur annule l'opération. - La séquence se termine sans modification.
Post-condition :	Les entrepôts sont ajoutés, modifiés ou archivés selon l'action effectuée.

Table 4.3 – La description textuelle de CU « Gérer entrepôts ».

Diagramme de séquence de CU « Gérer entrepôts »

La figure ci-après présente le diagramme de séquence correspondant au CU « Gérer entrepôts ».

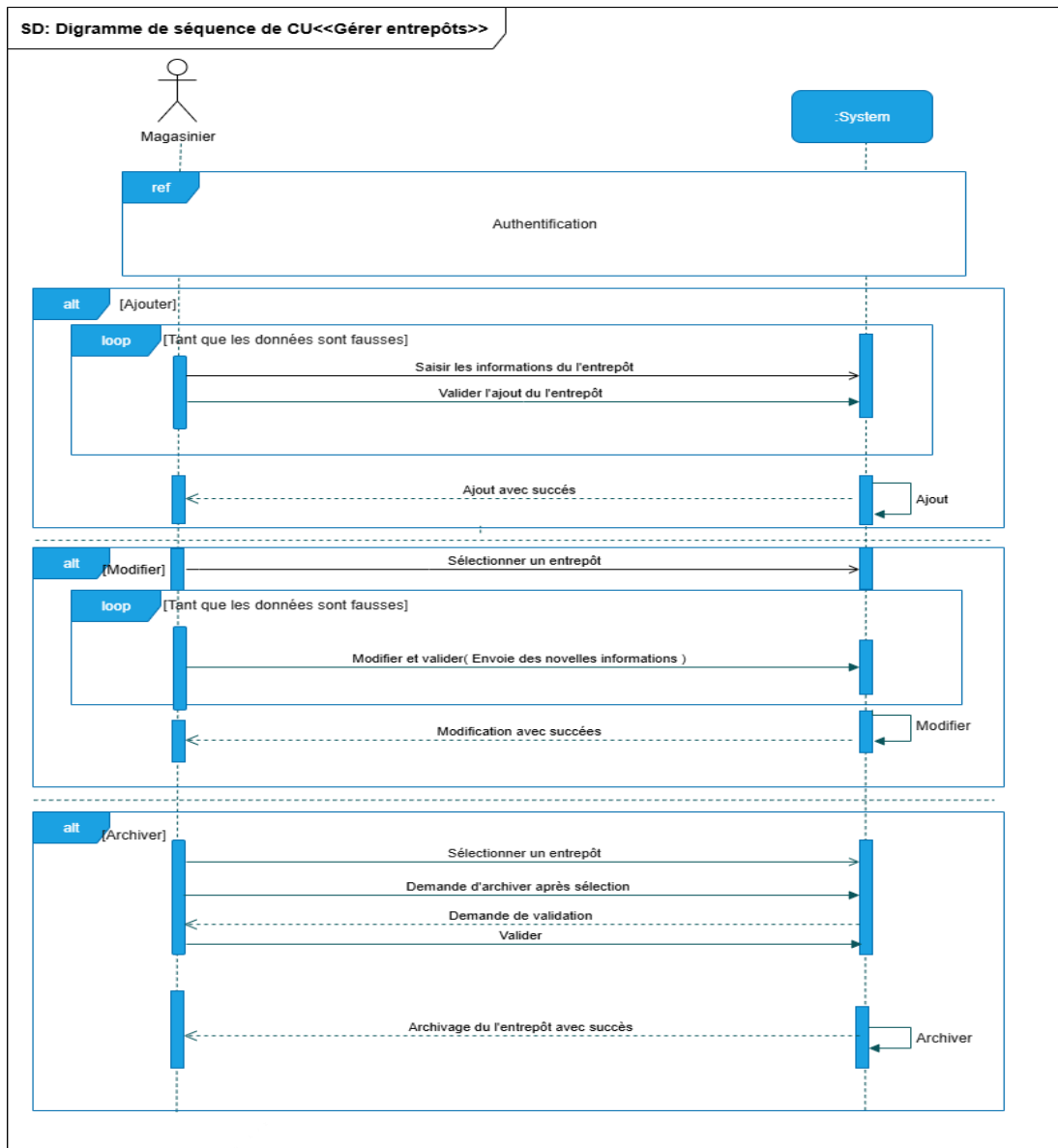


Figure 4.4 – Diagramme de séquence de CU « Gérer entrepôts »

4.3.4 Diagramme d'interaction de CU « Gérer entrepôts »

La figure ci-dessous représente le diagramme d'interaction correspondant au de CU « Gérer entrepôts ».

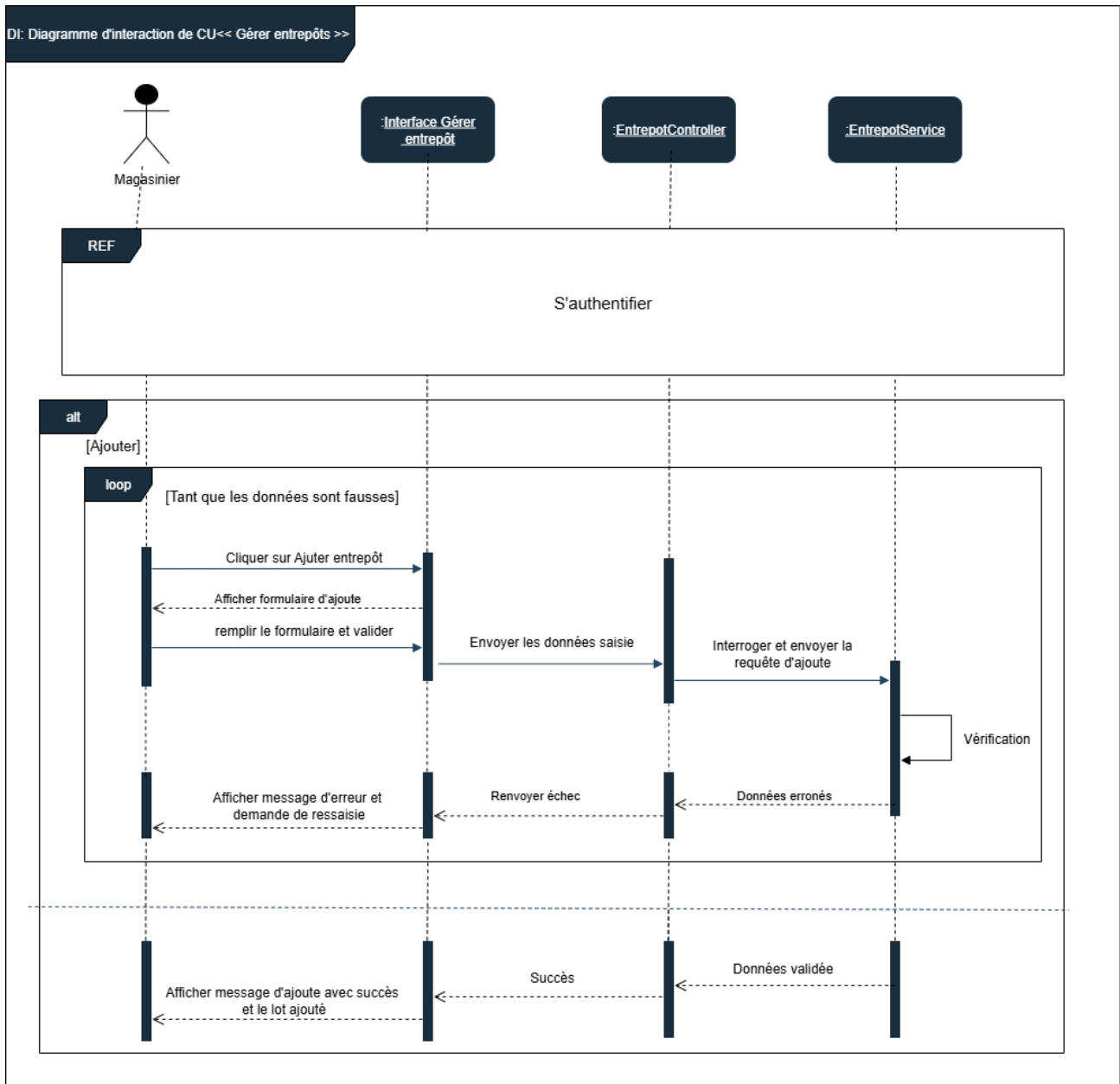


Figure 4.5 – Diagramme d'interaction de CU « Gérer entrepôts »

4.4.1 Le dictionnaire de données du troisième sprint

Cette section présente le dictionnaire de données élaboré lors du troisième sprint, en détaillant la structure des tables, les attributs définis ainsi que leurs relations, afin de garantir une modélisation cohérente du système.

Table : Lot	
id_lot	INT (11), clé primaire
num	INT (11)
quantite	INT (11)
date_fab	DATETIME
date_exp	DATETIME
actif	BOOLEAN
id_user	INT (11), clé étrangère vers Utilisateur
id_entrepot	INT (11), clé étrangère vers Entrepot
id_article	INT (11), clé étrangère vers Article

Table : Entrepot	
#id	INT (11), clé primaire
nom	VARCHAR (25)
localisation	VARCHAR (50)
actif	BOOLEAN

Table 4.4 – Le dictionnaire de données du troisième sprint

4.5 les interfaces graphiques du troisième sprint

les figures ci-dessus représentent quelques interfaces de sprint 3, tel que la figure 4.7 montre l'interface de gestion des lots, et la figure 4.8 illustre l'interface de gestion des entrepôts.

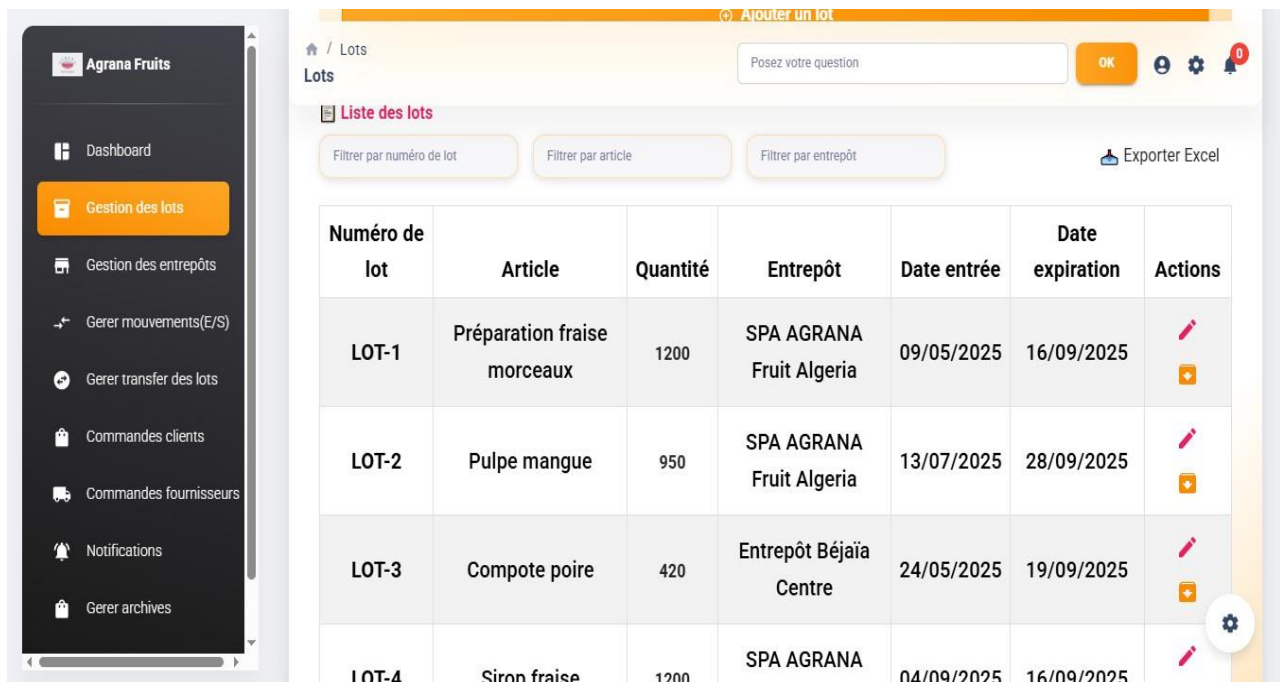


Figure 4.7 – Interface de gestion des lots



Figure 4.8 – Interface de gestion des entrepôts

4.6 Étude du quatrième sprint

Cette section présente l'étude du quatrième sprint, en détaillant les fonctionnalités prioritaires retenues, les tâches réalisées ainsi que les résultats obtenus à l'issue de cette itération.

4.6.1 les fonctionnalités traitées et la durée

Ce sprint est programmé pour être réalisé dans 9 à 10 jours et est composé des fonctionnalités suivantes :

ID	Tâche	Description	Estimé	Responsable
1	Ajouter un mouvement	Ajouter un mouvement de stock pour enregistrer une entrée ou une sortie d'articles.	8 jours	Djouder Massil
2	Exporter un mouvement en Excel	Exporter les mouvements de stock (entrées et sorties) vers un fichier Excel pour un suivi ou une analyse externe.	2 jours	Abderrahmane Bougheriou
3	Tests	Vérifier toutes les fonctionnalités et corriger les anomalies détectées.	2 jours	Abderrahmane Bougheriou

Table 4.5 – Le sprint backlog du quatrième sprint

4.6.2 Diagramme de cas d'utilisation de quatrième sprint

Le diagramme de cas d'utilisation suivant représente les cas d'utilisations concernant le sprint 4 comme une partie séparée :

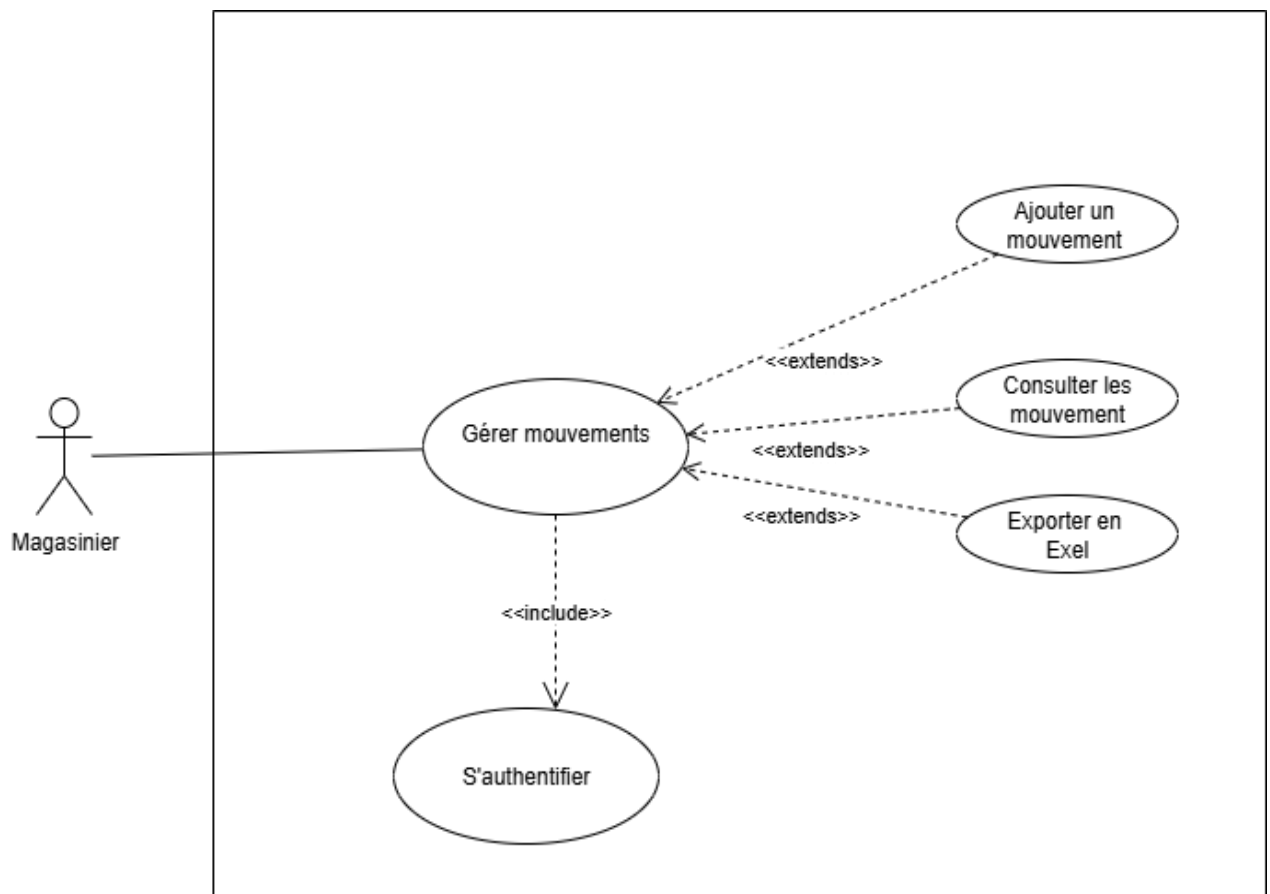


Figure 4.9 – Diagramme de cas d'utilisation du sprint 4

4.7 Les cas d'utilisations de quatrième sprint

Cette section expose les cas d'utilisation développés lors du quatrième sprint, en mettant en évidence les interactions principales entre les utilisateurs et le système.

4.7.1 Cas d'utilisation « Gérer mouvements »

Ce cas d'utilisation décrit la gestion des mouvements de stock, permettant d'enregistrer les entrées et sorties de produits afin d'assurer un suivi précis des flux.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer mouvements ».

CU :	Gérer mouvements
BUT :	Permettre au magasinier de gérer les mouvements de stock (entrées et sorties).
Acteur principal :	Magasinier
Acteur secondaire :	/
Préconditions :	Le magasinier est authentifié.
Scénario nominal :	<p>Créer un mouvement d'entrée :</p> <ol style="list-style-type: none"> 1. Le magasinier sélectionne « Nouveau mouvement d'entrée ». 2. Il choisit le type d'entrée (entrée fournisseur ou retour client). 3. Il sélectionne l'article et le lot, puis saisit la quantité. 4. Le système vérifie les données. 5. Le mouvement est enregistré et les quantités sont mises à jour. <p>Créer un mouvement de sortie :</p> <ol style="list-style-type: none"> 6. Le magasinier sélectionne « Nouveau mouvement de sortie ». 7. Il choisit le type de sortie (sortie client ou consommation interne). 8. Il sélectionne l'article et le lot, puis saisit la quantité. 9. Le système vérifie les données. 10. Le mouvement est enregistré et les quantités sont mises à jour.
Enchaînements alternatifs :	<p>4.a : Données invalides lors d'un mouvement d'entrée</p> <ul style="list-style-type: none"> – Le système signale que les données sont invalides. – Le magasinier corrige et resoumet. – Reprise au point 5. <p>9.a : Données invalides lors d'un mouvement de sortie</p> <ul style="list-style-type: none"> – Le système signale que les données sont invalides. – Le magasinier corrige et resoumet. – Reprise au point 10. <p>9.b : Quantité insuffisante pour une sortie</p> <ul style="list-style-type: none"> – Le système signale que la quantité demandée n'est pas disponible.

	– Le mouvement est annulé.
Post-condition :	Les mouvements (entrées ou sorties) sont enregistrés et les quantités des articles/lots sont mises à jour.

Table 4.6 –La description textuelle de CU « Gérer mouvements »

Diagramme de séquence de CU « Gérer mouvements »

La figure ci-après présente le diagramme de séquence correspondant au CU « Gérer mouvements ».

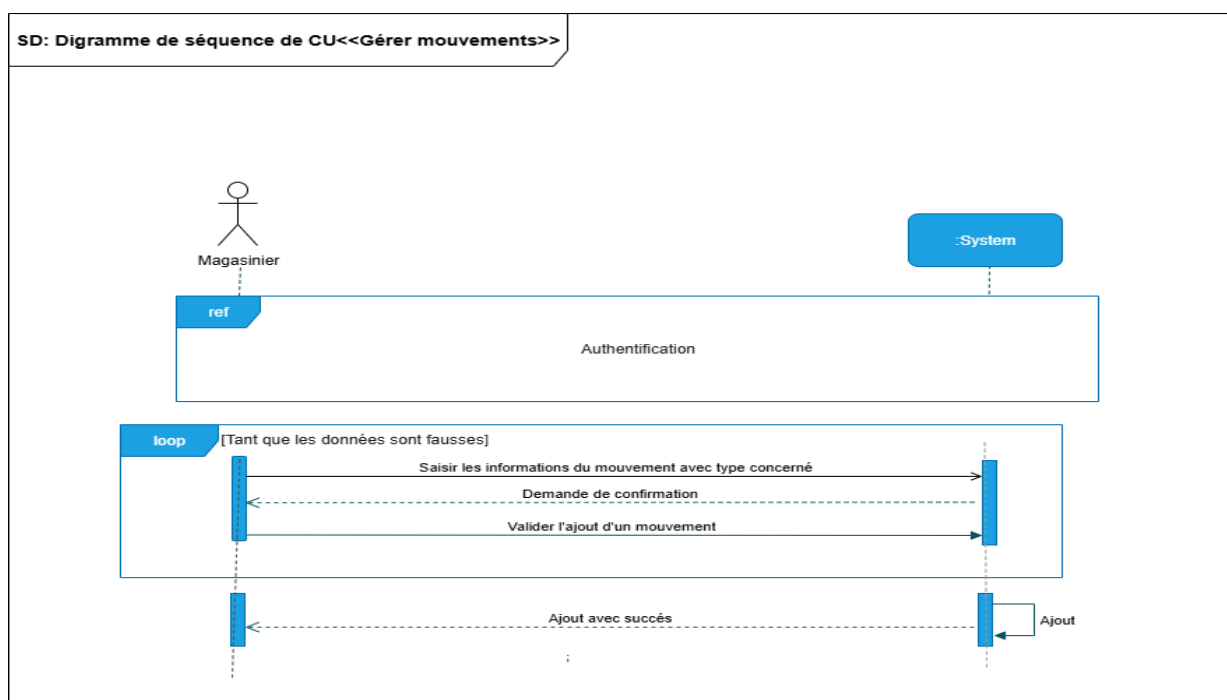


Figure 4.10 – Diagramme de séquence de CU « Gérer mouvements »

4.7.2 Diagramme de classe du quatrième sprint

Le diagramme de classe suivant représente les classes qui concernent le sprint 4 :

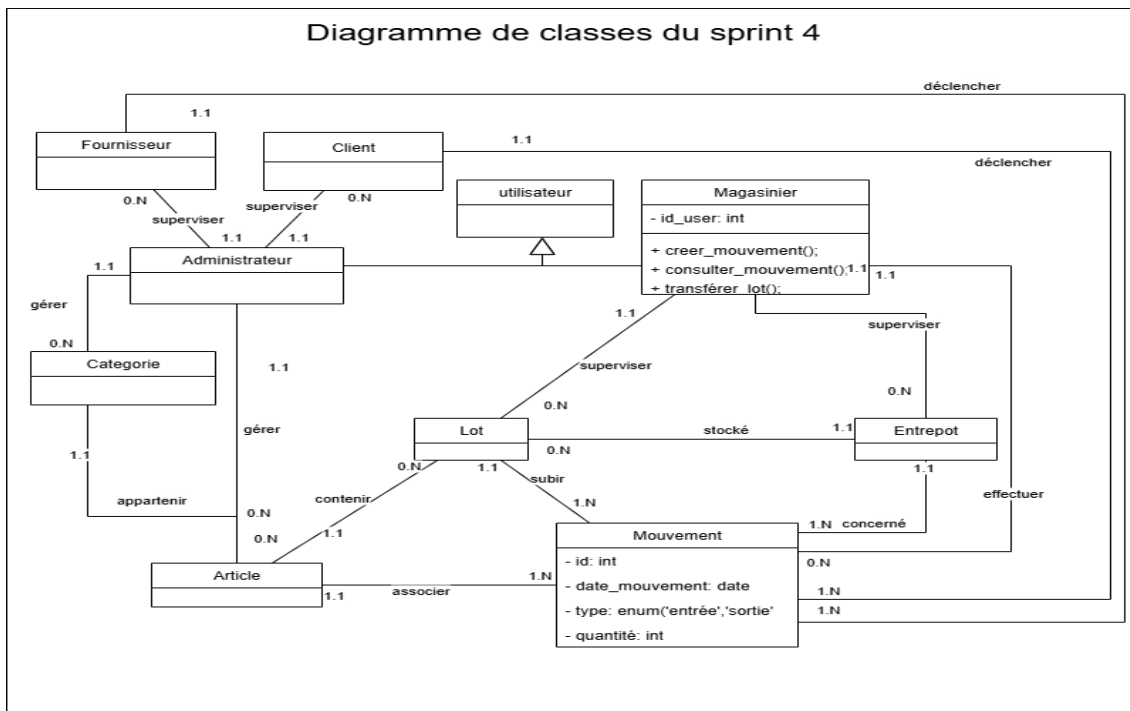


Figure 4.11 – Diagramme de classes du quatrième sprint

4.8 Le schéma relationnel du quatrième sprint

On le décrit par :

Utilisateur (id_user, nom, email, mdp, role, actif)

Article (id_article, CodeArticle, nom, description, quantité, qtm, actif, #id_cat)

Client (id_client, nom, contact, adresse_c, actif)

Categorie (id_cat, nom_cat, actif)

Fournisseur (id_fournisseur, nom, contact, adresse_f, actif)

Lot (id_lot, num, quantité, date_fab, date_exp, actif, #id_user, #id_article, #id_entrepot)

Entrepot (id_entrepot, nom, localisation, actif)

Mouvement (id_mouv, quantité, date_mouvement, actif, #id_user, #id_article, #id_entrepot)

4.9.1 Le dictionnaire de données du quatrième sprint

Cette section présente le dictionnaire de données élaboré lors du quatrième sprint, en détaillant la structure des tables, les attributs définis ainsi que leurs relations, afin de garantir une modélisation cohérente du système.

Table : Mouvement	
id	INT (11), clé primaire
type	ENUM('entrée', 'sortie')
quantite	INT (11)
date_mouvement	DATETIME
id_article	INT (11), clé étrangère vers Article
id_lot	INT (11), clé étrangère vers Lot
id_entrepot	INT (11), clé étrangère vers Entrepot
id_fournisseur	INT (11), clé étrangère vers Fournisseur
id_client	INT (11), clé étrangère vers Client
id_user	INT (11), clé étrangère vers Utilisateur

Table 4.7 – Le dictionnaire de données du quatrième sprint

4.9 les interfaces graphiques du quatrième sprint

La figure 4.12 ci-dessus illustre l'interface des mouvements.

Type	Code Article	Lot	Quantité	Fournisseur	Client	Entrepôt	Date
entrée	MP-004	7	200	Fruitière de Kabylie	-	SPA AGRANA Fruit Algeria	14/09/2025
entrée	MP-005	8	184	Fruitière de Kabylie	-	SPA AGRANA Fruit Algeria	14/09/2025
sortie	PF-008	6	500	-	Pâtisserie Délice	entrepôt de sortie	14/09/2025
sortie	PF-012	5	920	-	Ramdy	entrepôt de sortie	14/09/2025
sortie	PF-003	2	950	-	Soummam	entrepôt de sortie	14/09/2025

Figure 4.12 – Interface de mouvements (E/S).

4.10 Conclusion

Ce release améliore l'application de gestion de stock en permettant aux magasiniers de gérer efficacement les mouvements d'articles, les lots et les entrepôts.

Chapitre 5 : Présentation de troisième Release

5.1 Introduction

Dans ce cinquième chapitre, nous abordons le contexte et les fonctionnalités réalisées dans deux sprints.

5.2 Étude du cinquième sprint

Cette section présente l'étude du cinquième sprint, en détaillant les fonctionnalités prioritaires retenues, les tâches réalisées ainsi que les résultats obtenus à l'issue de cette itération.

5.2.1 les fonctionnalités traitées et la durée

Ce sprint est programmé pour être réalisé dans un mois (31 jours) et est composé des fonctionnalités suivantes :

ID	Tâche	Description	Estimé	Responsable
1	Créer une commande client	Créer une nouvelle commande client avec les informations sur les produits, la quantité et client	4 jours	Djouder Massil
2	Annuler une commande client	Annuler une commande client encore si le client souhaite annuler sa demande.	2 jours	Djouder Massil
3	Modifier une commande client	Modifier une commande client encore pour ajouter, supprimer ou modifier les produits et quantités.	3 jours	Djouder Massil
4	Exporter une commande client en Excel	Exporter les détails d'une commande client vers un fichier Excel pour un suivi ou une gestion externe.	3 jours	Abderrahmane Bougheriou
5	Imprimer et rechercher une commande	Filtrer les commandes et pouvoir les imprimer pour fournir une version papier du détail de la commande.	2 jours	Djouder Massil
6	Livrer une commande client	Gérer la livraison d'une commande client en s'assurant que les produits sont envoyés au client.	3 jours	Abderrahmane Bougheriou
7	Retour client (Recycler/ Détruire)	Gérer les retours de produits clients, soit pour les recycler, soit pour les détruire selon les politiques de l'entreprise.	3 jours	Djouder Massil
8	Créer une commande fournisseur	Créer une nouvelle commande fournisseur pour réapprovisionner les stocks en fonction des besoins.	4 jours	Abderrahmane Bougheriou
9	Modifier une commande fournisseur	Modifier les informations d'une commande fournisseur existante, comme les produits, les quantités ou les prix.	3 jours	Abderrahmane Bougheriou

1 0	Annuler une commande fournisseur	Annuler une commande fournisseur si le besoin change ou si la commande n'est plus nécessaire.	2 jours	Abderrahmane Bougheriou
1 1	Recevoir une commande fournisseur	Recevoir les produits d'une commande fournisseur et les enregistrer dans le système d'inventaire.	3 jours	Abderrahmane Bougheriou
1 2	Valider une commande	Donner la main à l'administrateur de traiter et valider la commande.	2 jours	Djouder Massil
1 3	Tests	Vérifier toutes les fonctionnalités et corriger les anomalies détectées.	2 jours	Djouder Massil

Table 5.1 – Le sprint backlog du cinquième sprint

5.2.2 Diagramme de cas d'utilisation de cinquième sprint

Le diagramme de cas d'utilisation suivant représente les cas d'utilisations concernant le sprint 5 comme une partie séparée :

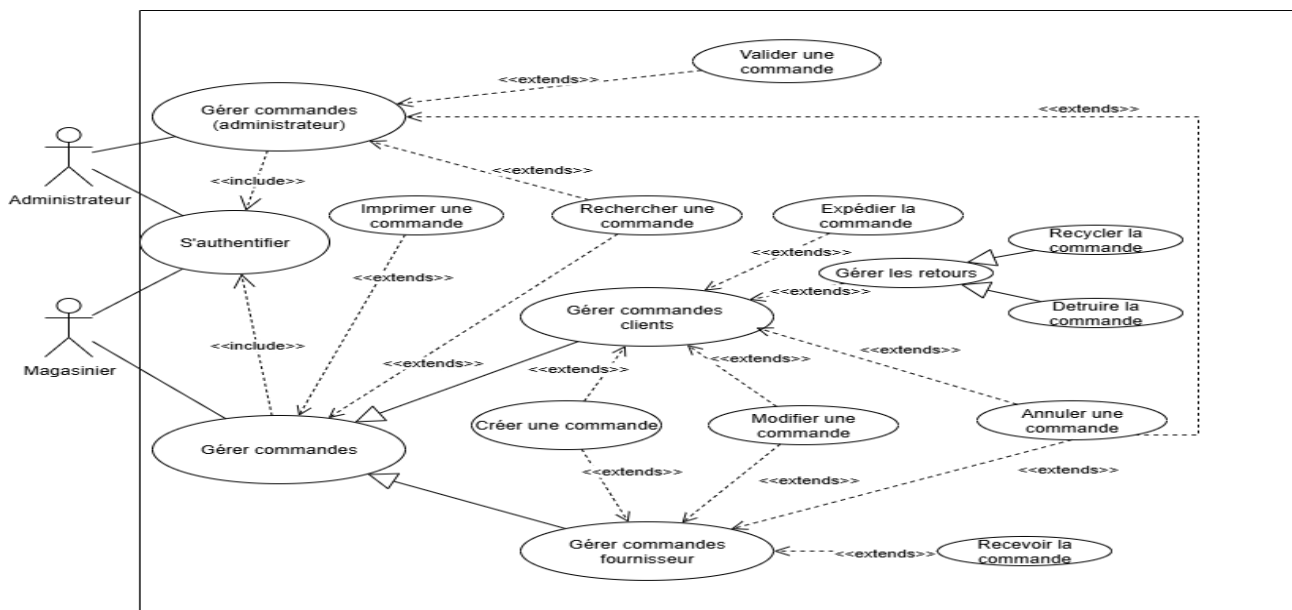


Figure 5.1 – Diagramme de cas d'utilisation du sprint 5

5.3 Les cas d'utilisations de cinquième sprint

Cette section expose les cas d'utilisation développés lors du cinquième sprint, en mettant en évidence les interactions principales entre les utilisateurs et le système.

5.3.1 Cas d'utilisation « Gérer commande (espace magasinier) »

Ce cas d'utilisation illustre la gestion des commandes dans l'espace magasinier, permettant à ce dernier de créer, consulter, recevoir ou annuler des commandes selon leur état d'avancement.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer commande (espace magasinier) ».

CU :	Gérer commandes (Magasinier)
BUT :	Permettre au magasinier de créer, modifier, annuler une commande, gérer les retours clients.
Acteur principal :	Magasinier
Acteur secondaire :	/
Préconditions :	Le magasinier est authentifié.
Scénario nominal :	<p>Créer une commande :</p> <ol style="list-style-type: none">1. Sélectionne « Créer commande ».2. Saisit les informations.3. Le système vérifie la disponibilité.4. La commande est enregistrée. <p>Modifier une commande :</p> <ol style="list-style-type: none">5. Sélectionne une commande encore.6. Modifie les informations.7. Le système vérifie la disponibilité.8. Les modifications sont enregistrées. <p>Annuler une commande :</p> <ol style="list-style-type: none">9. Sélectionne une commande encore.10. Confirme l'annulation.11. Le système marque la commande comme annulée. <p>Gérer les retours clients :</p> <ol style="list-style-type: none">12. Le client retourne des articles.13. Le magasinier réceptionne et contrôle l'état.14. Le magasinier choisit une action : recycler ou détruire.15. Le système met à jour l'état du stock selon l'action choisie.

<p>Enchaînements er- natifs :</p>	<p>alt</p> <p>3.a : Stock insuffisant à la création</p> <ul style="list-style-type: none"> – Le système signale que la quantité demandée dépasse le stock disponible. – Le magasinier corrige la commande. – La séquence reprend au point 4. <p>10.a : Annulation impossible</p> <ul style="list-style-type: none"> – Le système indique que la commande ne peut pas être annulée (déjà expédiée ou traitée). – La séquence se termine sans modification.
<p>Post-condition :</p>	<p>Les commandes sont traitées selon l'action effectuée (création, modification, annulation, retour, recyclage ou destruction).</p>

Table 5.2 – Description textuelle du cas d'utilisation « Gérer commandes (Magasinier) »

Diagramme de séquence du cas d'utilisation « Gérer commandes(magasinier) »

La figure ci-après présente le diagramme de séquence correspondant au CU « Gérer commandes(magasinier) ».

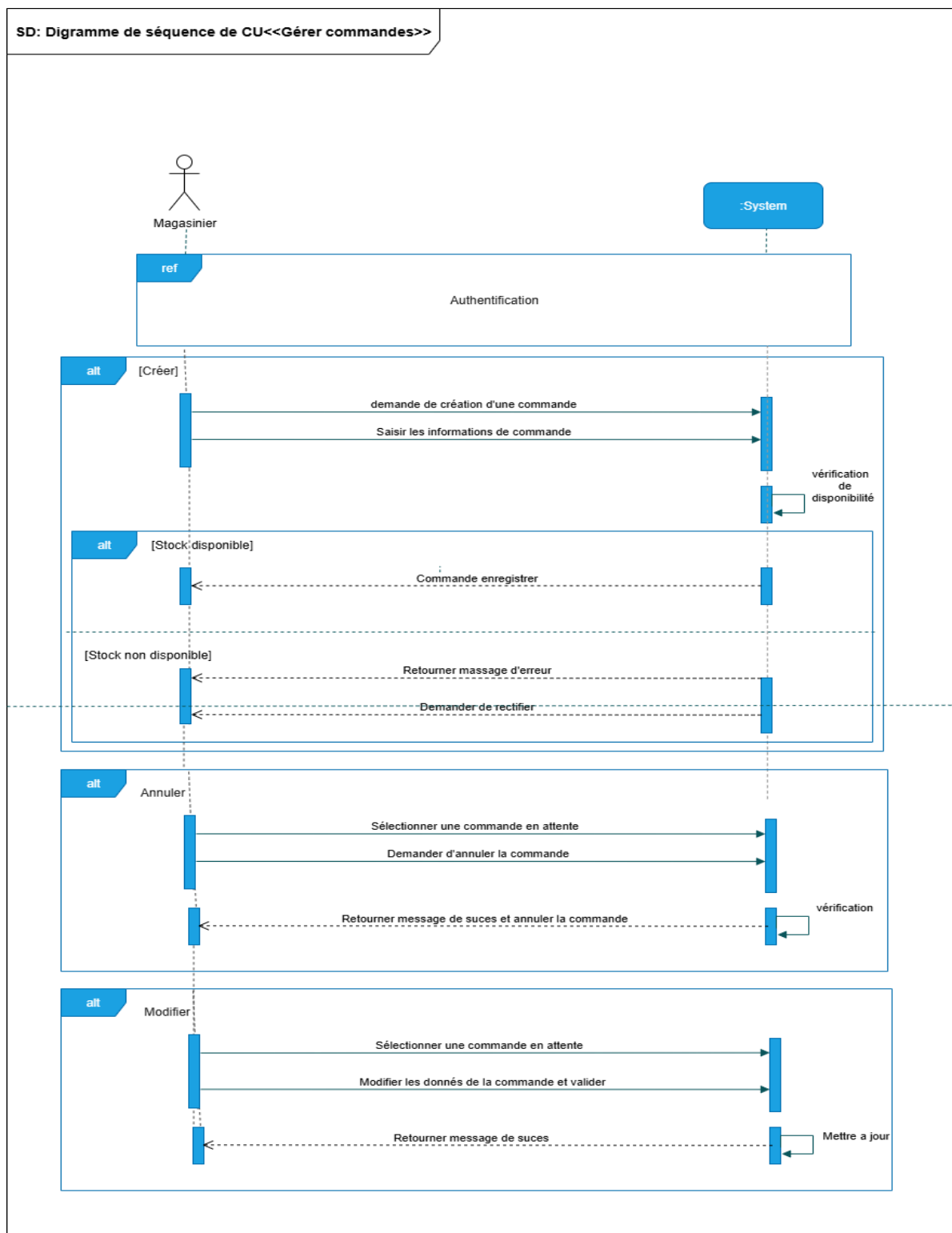


Figure 5.2 – Diagramme de séquence du cas d'utilisation « Gérer commandes (magasinier) »

5.3.2 Diagramme d'interaction de CU « Gérer commandes(magasinier) »

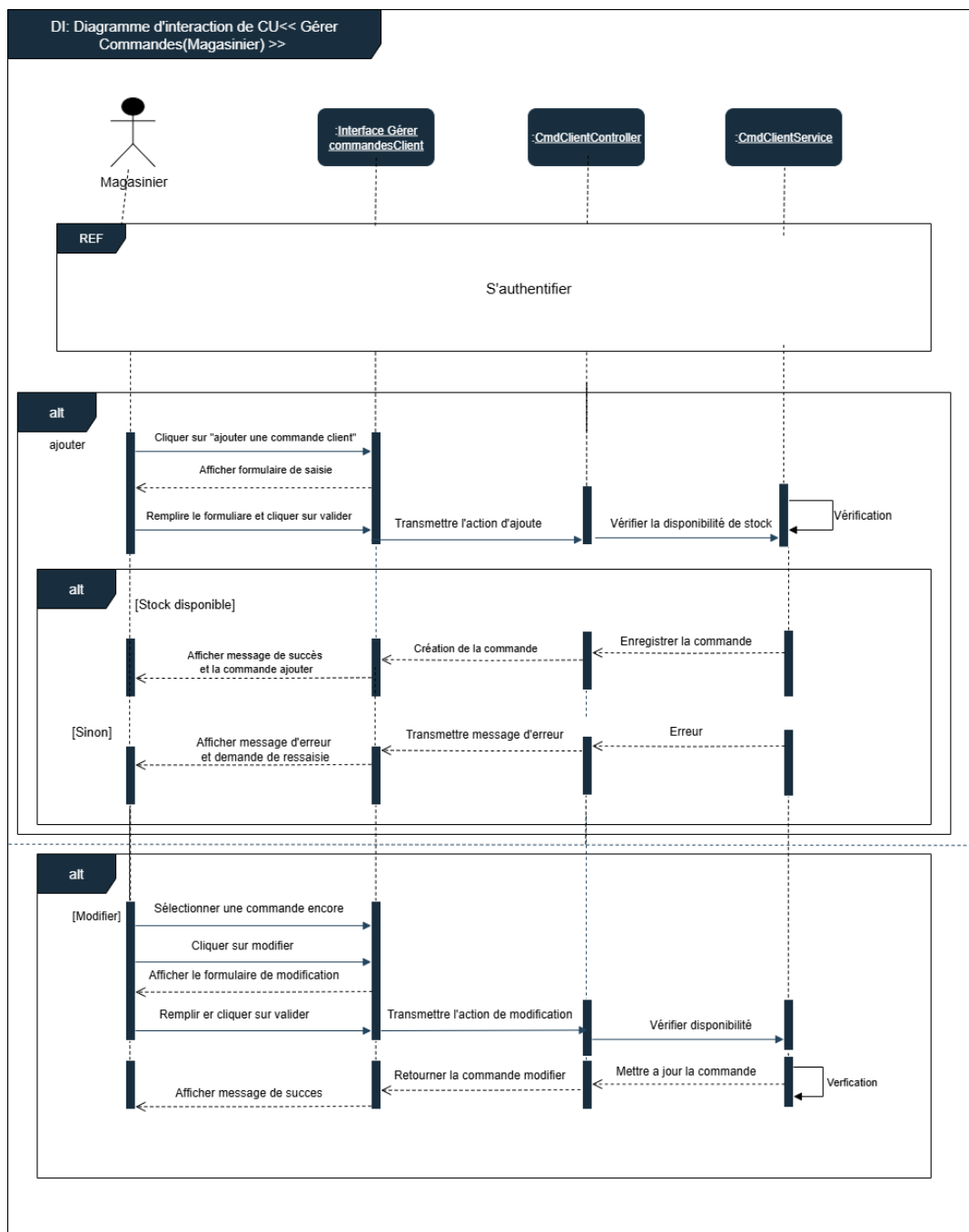


Figure 5.3 – Diagramme d'interaction de CU « Gérer commandes(magasinier) »

5.3.3 Cas d'utilisation « Gérer commande (espace admin) »

Ce cas d'utilisation décrit la gestion des commandes dans l'espace administrateur, offrant la possibilité de valider, annuler et suivre les commandes afin d'assurer un contrôle global du processus.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer commande (espace admin) ».

CU :	Gérer commandes (Administrateur)
BUT :	Permettre à l'administrateur de superviser, valider ou annuler les commandes.
Acteur principal :	Administrateur
Acteur secondaire :	/
Préconditions :	L'administrateur est authentifié.
Scénario nominal :	Superviser les commandes : <ol style="list-style-type: none"> 1. L'administrateur consulte la liste des commandes créées par les magasiniers. 2. Il accède aux détails d'une commande. 3. Le système affiche les informations associées. Valider / Annuler une commande : <ol style="list-style-type: none"> 4. L'administrateur sélectionne une commande en attente. 5. Il valide ou rejette la commande. 6. Le système met à jour l'état de la commande.
Enchaînements alternatifs :	3.a : Informations de commande incomplètes <ul style="list-style-type: none"> – Le système indique que certaines informations sont manquantes. – L'administrateur demande une correction au magasinier. – La séquence reprend au point 3. 5.a : Commande non valide <ul style="list-style-type: none"> – Le système signale une incohérence (stock insuffisant, doublon, etc.). – L'administrateur rejette la commande. – La séquence se termine.
Post-condition :	Les commandes sont supervisées et validées/rejetées par l'administrateur, garantissant la cohérence du stock.

Table 5.3 – Description textuelle du cas d'utilisation « Gérer commandes (espace admin) »

Diagramme de séquence du cas d'utilisation « Gérer commandes(admin) »

La figure ci-après présente le diagramme de séquence correspondant au CU « Gérer commandes(admin) ».

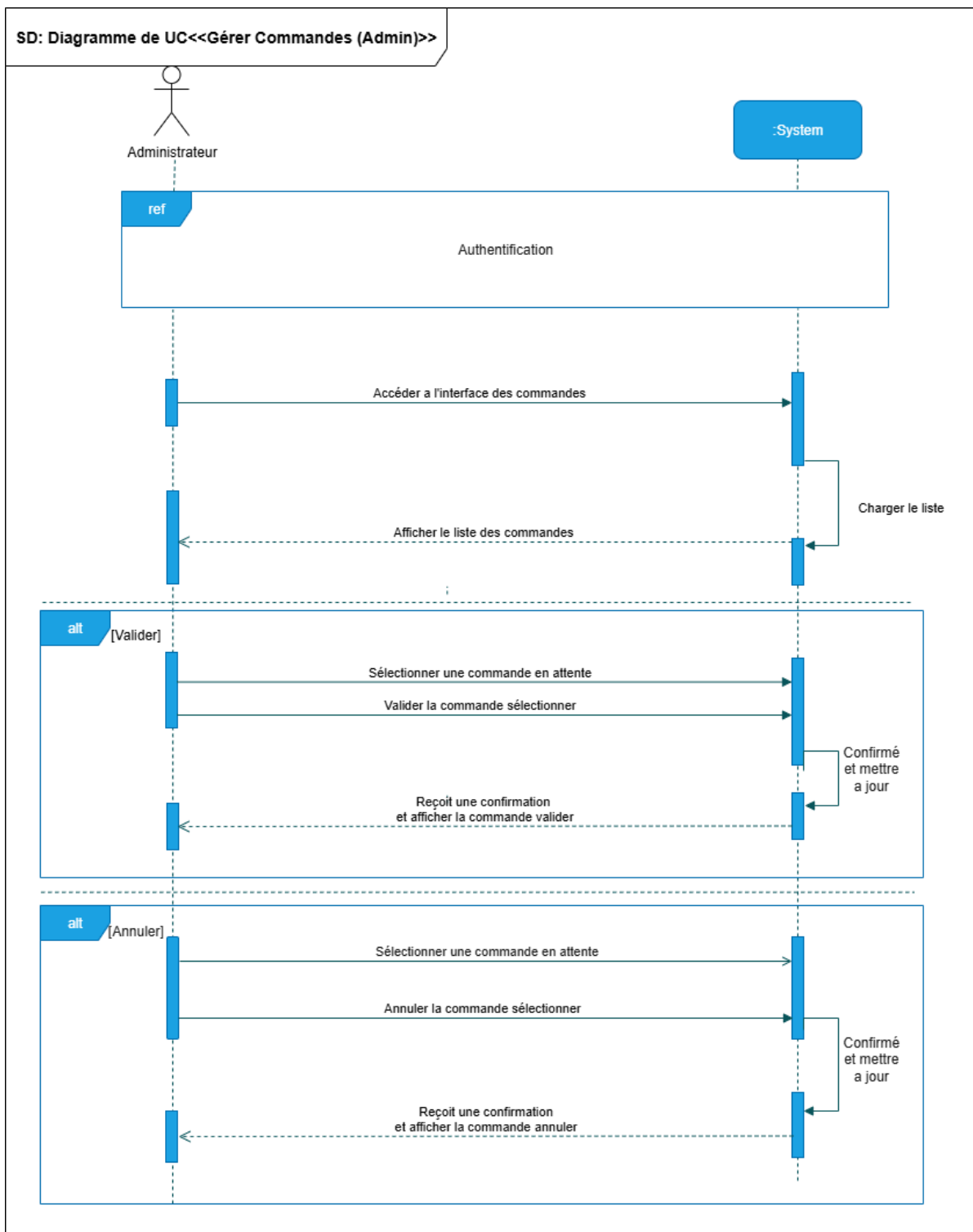


Figure 5.4 – Diagramme de séquence du cas d'utilisation « Gérer commandes »

5.3.4 Diagramme d'interaction de CU « Gérer commandes(admin) »

La figure ci-dessous représente le diagramme d'interaction correspondant au de CU « Gérer commandes(admin) ».

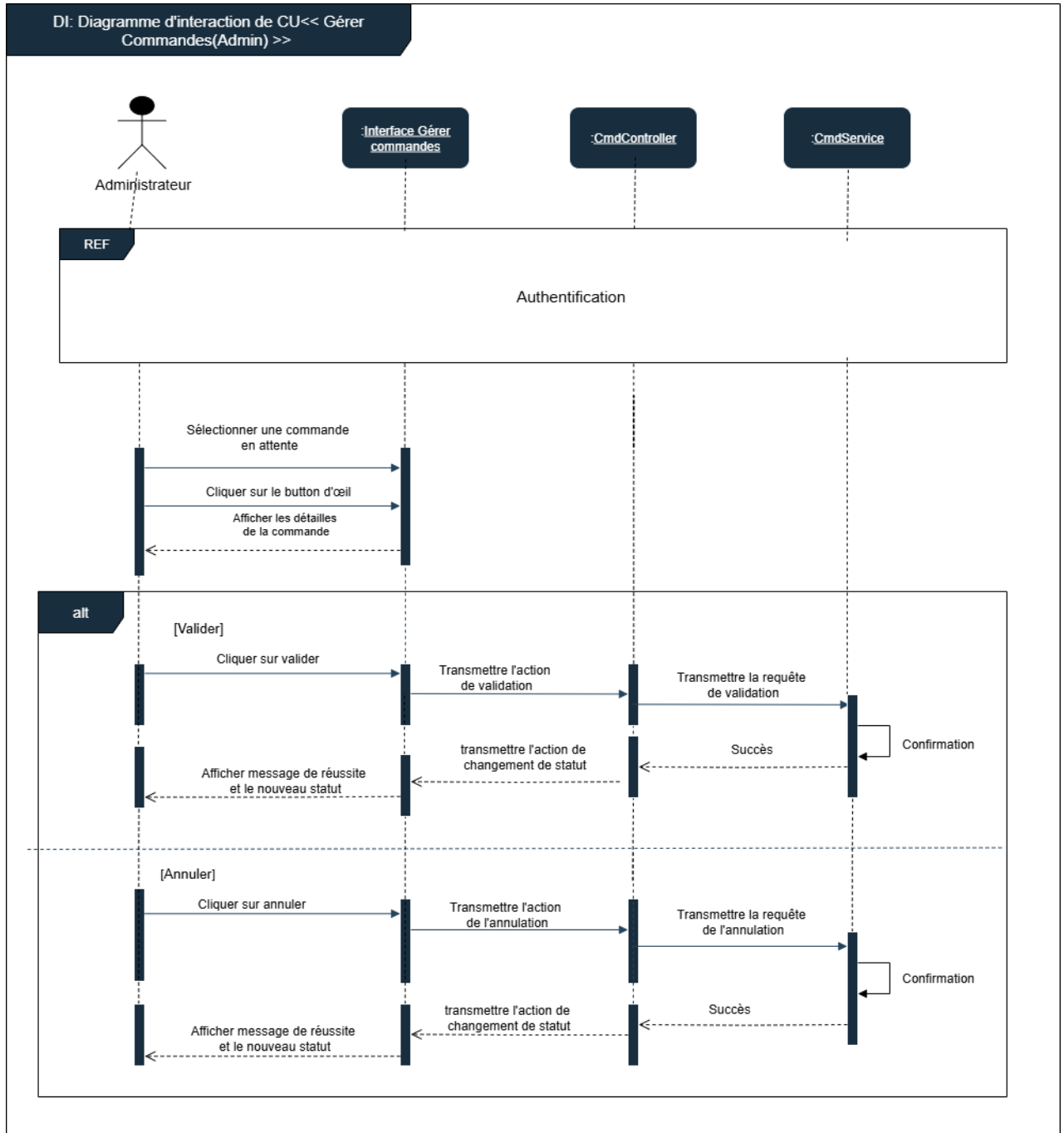


Figure 5.5 – Diagramme d'interaction de CU « Gérer commandes(admin) »

5.3.5 Diagramme de classe du cinquième sprint

Le diagramme de classe suivant représente les classes qui concernent le sprint 5 :

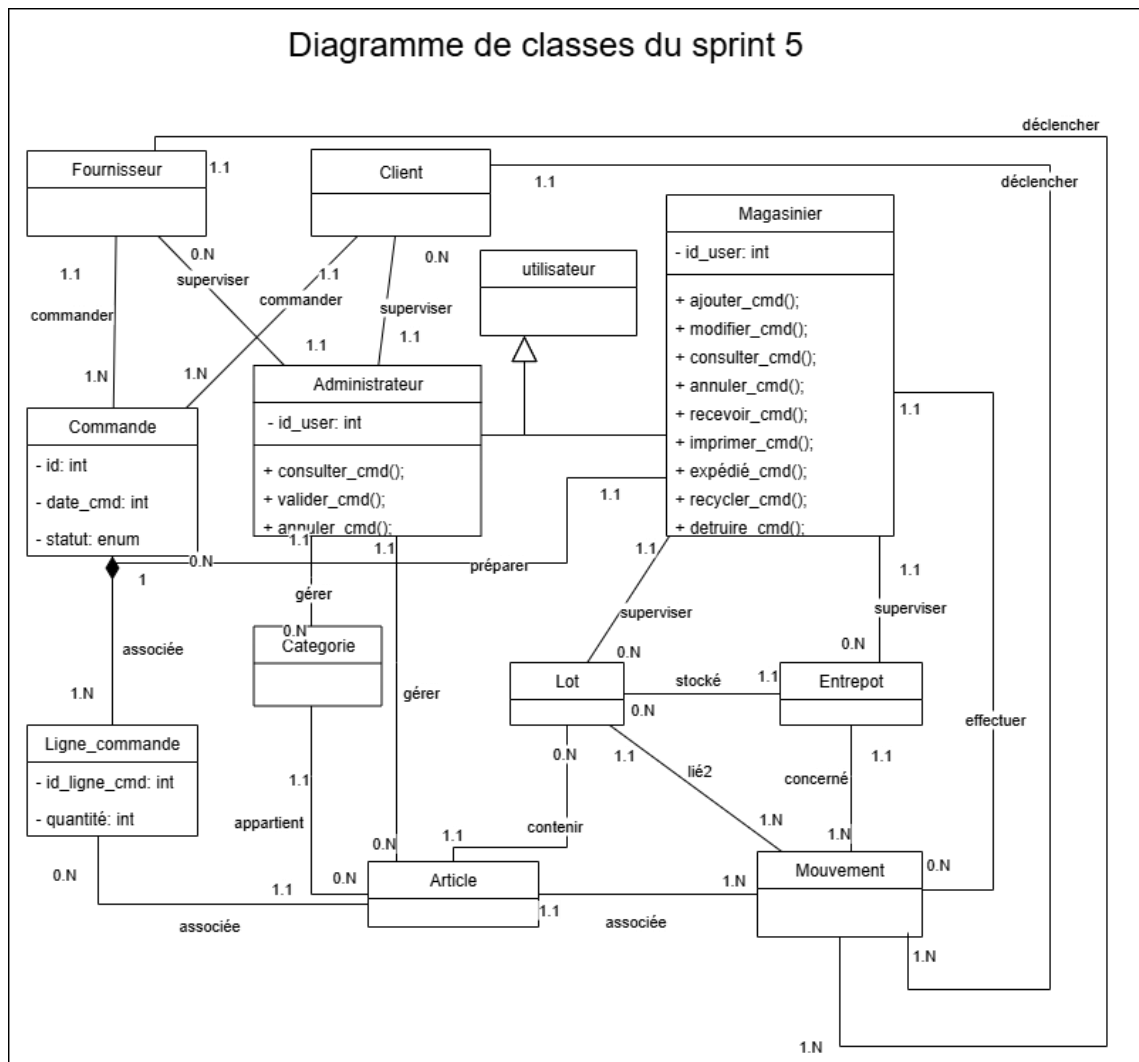


Figure 5.6 – Diagramme de classes du cinquième sprint

5.4 Le schéma relationnel du cinquième sprint

On le décrit par :

Commande (id, statut, type, date_cmd, date_reception, #id_user, #id_fournisseur, #id_client)

Ligne_commande (id_ligne_cmd, quantité, #id_article, #id)

5.4.1 Le dictionnaire de données du cinquième sprint

Cette section présente le dictionnaire de données élaboré lors du cinquième sprint, en détaillant la structure des tables, les attributs définis ainsi que leurs relations, afin de garantir une modélisation cohérente du système.

Table : Commande	
id	INT (11), clé primaire

date_cmd	DATETIME
statut	ENUM ('en attente', 'reçu', 'annulé', 'validé')
type	ENUM('fournisseur', 'client')
date_reception	DATETIME
id_client	INT (11), clé étrangère vers Client
id_fournisseur	INT (11), clé étrangère vers Fournisseur

Table : Ligne_Commande	
id_ligne_cmd	INT (11), clé primaire
quantite	INT (11)
id_commande	INT (11), clé étrangère vers Commande
idArticle	INT (11), clé étrangère vers Article

Table 5.4 – Le dictionnaire de données du cinquième sprint

5.5 les interfaces graphiques du cinquième sprint

La figure 4.12 ci-dessus illustre l'interface des mouvements.

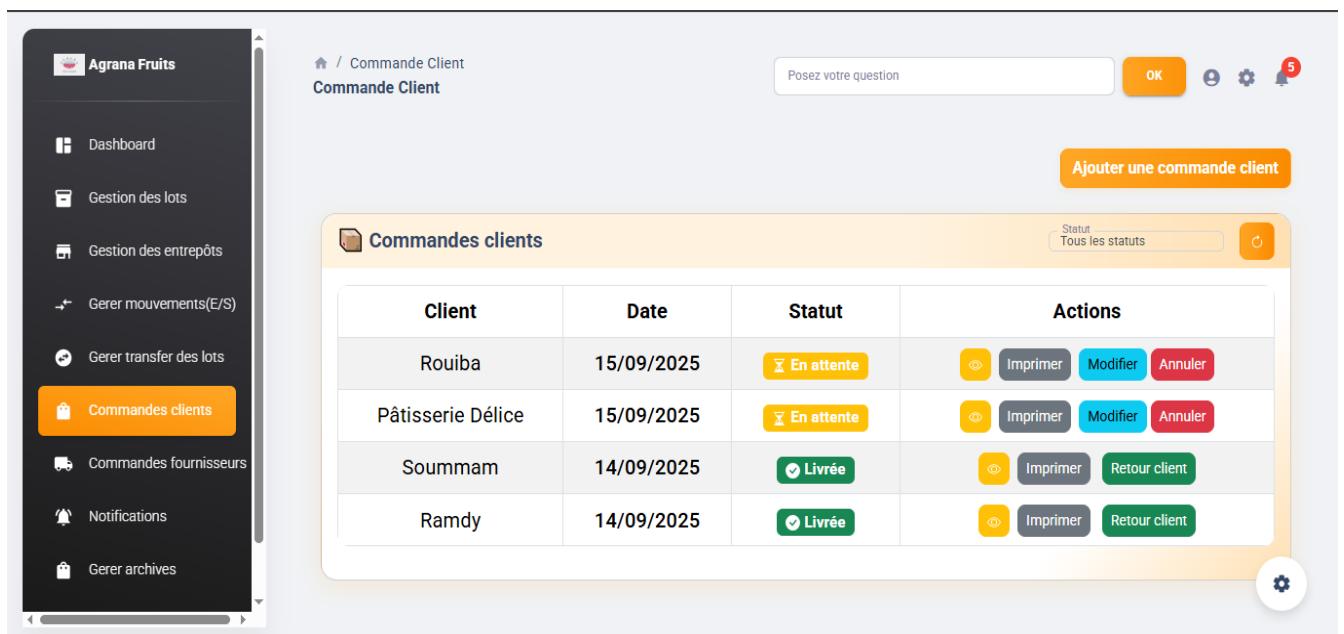


Figure 5.7 – Interface de la gestion des commandes (espace magasinier)

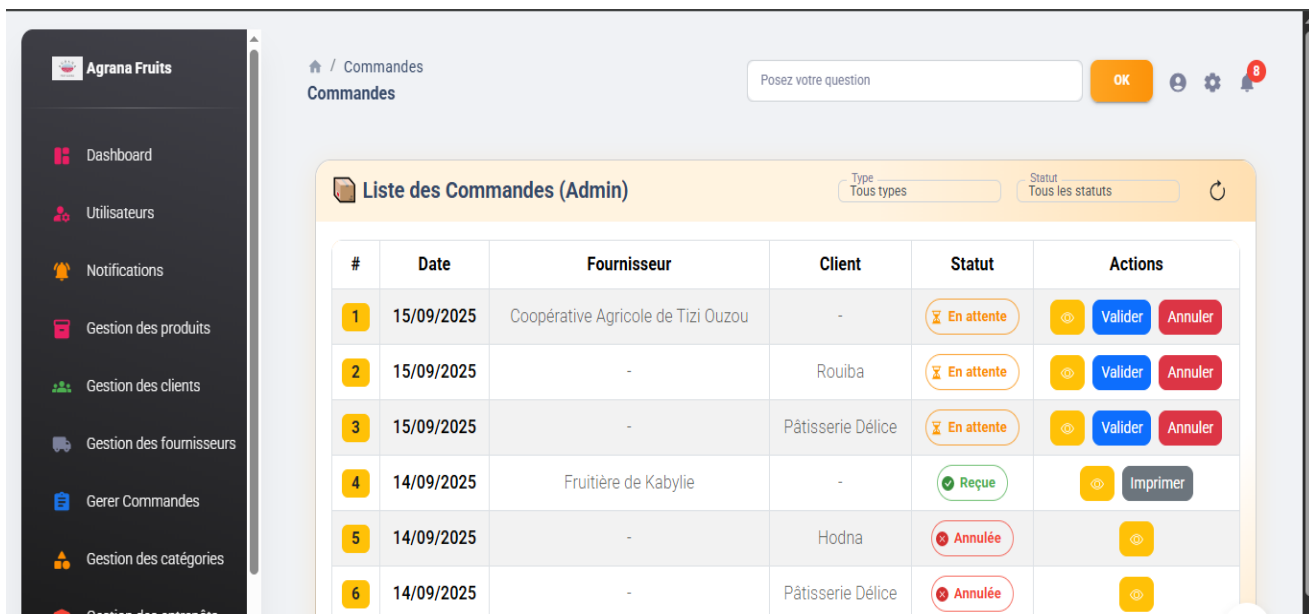


Figure 5.8 – Interface de la gestion des commandes (espace admin)

5.6 Étude du sixième sprint

Cette section présente l'étude du cinquième sprint, en détaillant les fonctionnalités prioritaires retenues, les tâches réalisées ainsi que les résultats obtenus à l'issue de cette itération.

5.6.1 Les fonctionnalités traitées et la durée

Ce sprint est programmé pour être réalisé dans 15 jours et est composé des fonctionnalités suivantes :

ID	Tâche	Description	Estimé	Responsable
1	Activer un client archivé	Réactiver un client précédemment archivé pour lui permettre de passer de nouvelles commandes.	3 jours	Djouder Massil
2	Activer un fournisseur archivé	Réactiver un fournisseur précédemment archivé pour rétablir les commandes et transactions.	2 jours	Abderrahmane Bougheriou
3	Réactiver un compte utilisateur désactivé	Réactiver un compte utilisateur qui a été précédemment désactivé, en cas de besoin.	2 jours	Djouder Massil
4	Notification	Recevoir et consulter une notification pour informer de la réception d'une nouvelle commande.	2 jours	Abderrahmane Bougheriou
5	Notification de rupture de stock (IA)	Recevoir une notification indiquant qu'un produit est en rupture de stock et prendre des mesures.	3 jours	Djouder Massil
6	Notification d'expiration de produit	Recevoir une notification pour alerter qu'un produit est proche de sa date d'expiration.	2 jours	Djouder Massil

7	Supprimer une notification	Permettre la suppression d'une commande déjà lu.	2 jours	Djouder Massil
8	Tests	Vérifier toutes les fonctionnalités et corriger les anomalies détectées.	2 jours	Abderrahmane Bougheriou

Table 5.5 – Le sprint backlog du sixième sprint

5.6.2 Diagramme de cas d'utilisation de sixième sprint

Le diagramme de cas d'utilisation suivant représente les cas d'utilisations concernant le sprint 6 comme une partie séparée :

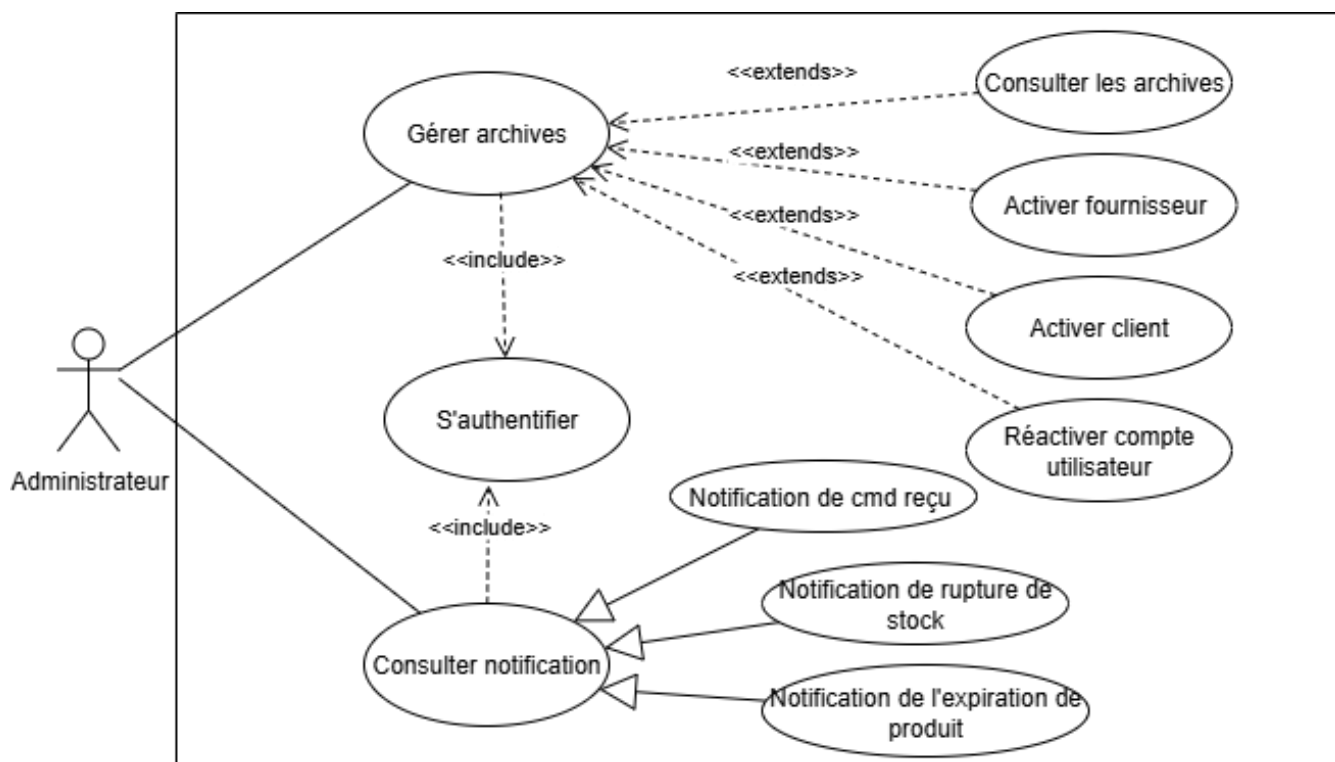


Figure 5.9 – Diagramme de cas d'utilisation du sprint 6

5.7 Les cas d'utilisations de sixième sprint

Cette section expose les cas d'utilisation développés lors du sixième sprint, en mettant en évidence les interactions principales entre les utilisateurs et le système.

5.7.1 Cas d'utilisation « Gérer archives »

Ce cas d'utilisation illustre la gestion des archives, permettant à l'administrateur de consulter, organiser et conserver l'historique des opérations effectuées dans le système.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Gérer archives ».

CU :	Gérer archives
BUT :	Permettre à l'administrateur de consulter les archives des clients, fournisseurs ou comptes utilisateurs désactivés et de les réactiver si nécessaire.
Acteur principal :	Administrateur
Acteur secondaire :	/
Préconditions :	L'administrateur est authentifié.
Scénario nominal :	<p>Consulter les archives :</p> <ol style="list-style-type: none"> 1. L'administrateur choisit « Consulter archives ». 2. Il sélectionne le type d'archive : client, fournisseur ou compte utilisateur désactivé. 3. Le système affiche la liste correspondante des éléments archivés. <p>Réactiver un élément :</p> <ol style="list-style-type: none"> 4. L'administrateur sélectionne un client, fournisseur ou compte utilisateur désactivé. 5. Il clique sur « Activer ». 6. Le système vérifie l'action et réactive l'élément. 7. L'élément réactivé apparaît désormais dans la liste active.
Enchaînements alternatifs :	<p>5.a : Annulation de la réactivation</p> <ul style="list-style-type: none"> – L'administrateur annule l'opération. – La séquence se termine sans modification.
Post-condition :	Les clients, fournisseurs ou comptes utilisateurs sont réactivés ou restent archivés selon l'action effectuée.

Table 5.6 – Description textuelle du cas d'utilisation « Gérer archives »

Diagramme de Séquence de CU « Gérer archives »

La figure ci-après présente le diagramme de séquence correspondant au CU « Gérer archives ».

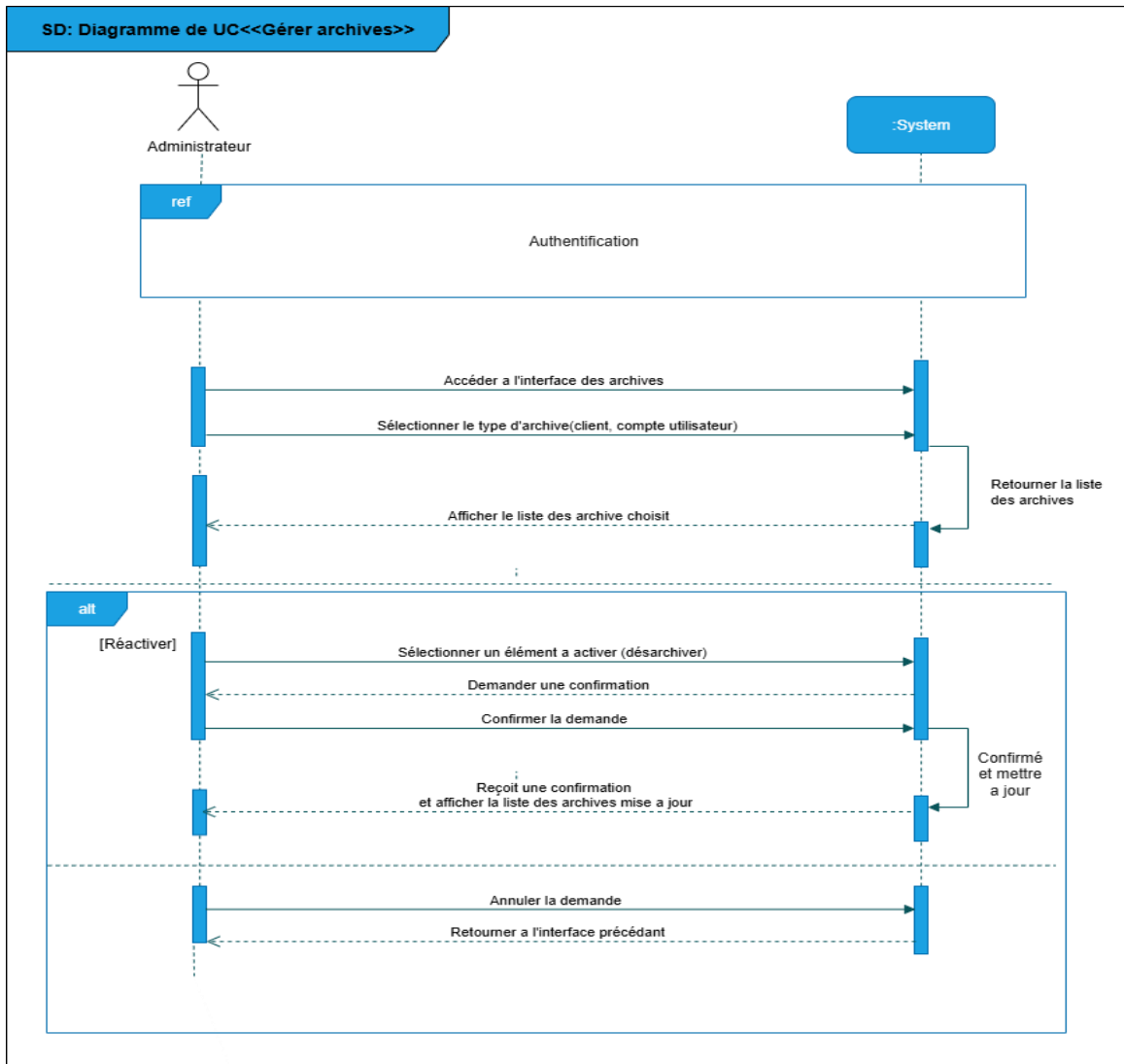


Figure 5.10 – Diagramme de Séquence de CU « Gérer archives »

5.7.2 Diagramme d'interaction de CU « Gérer archives »

La figure ci-dessous représente le diagramme d'interaction correspondant au CU « Gérer archives ».

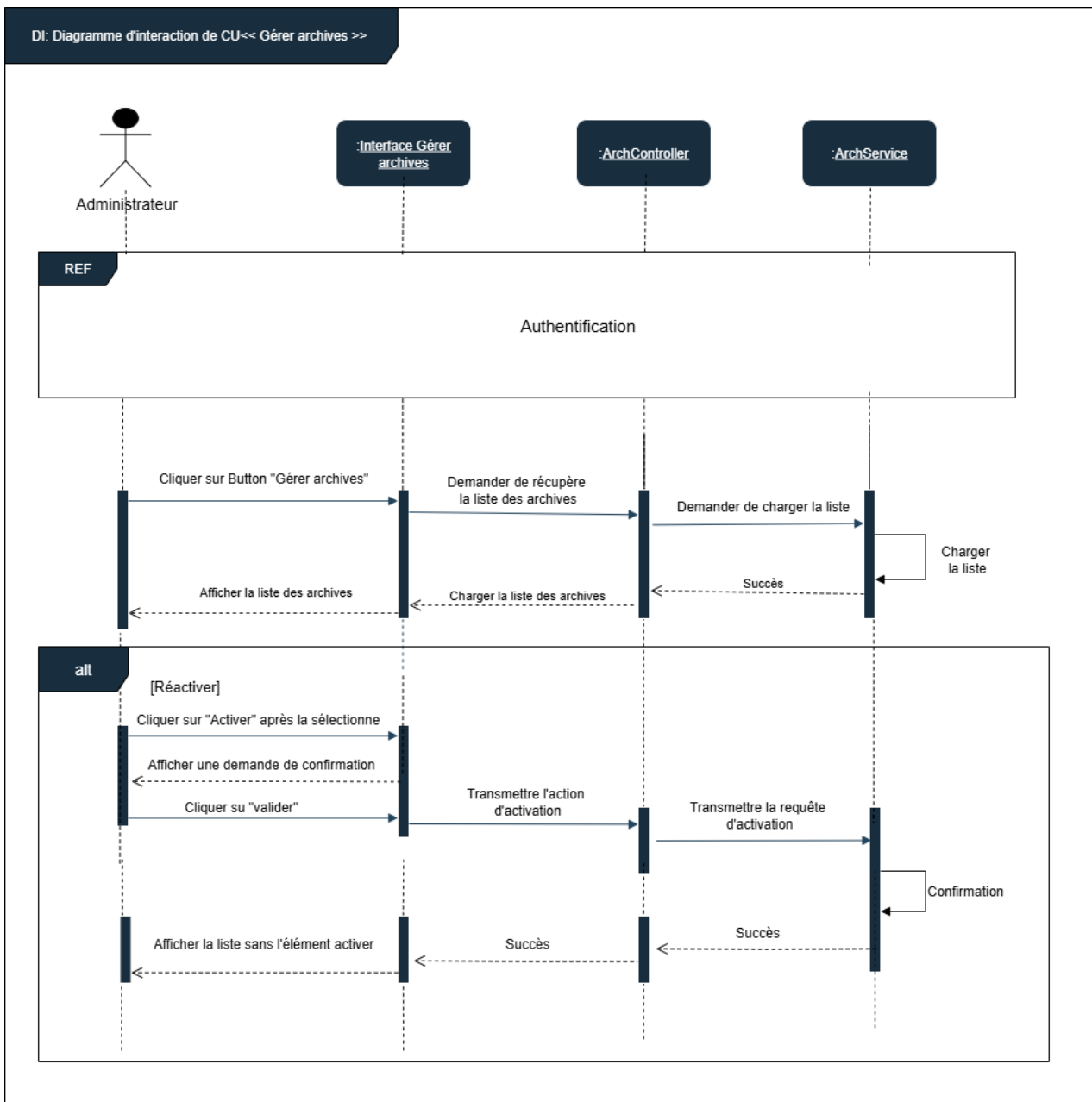


Figure 5.11 – Diagramme de Séquence de CU « Gérer archives »

5.7.3 Cas d'utilisation « consulter notification »

Ce cas d'utilisation décrit le processus par lequel l'utilisateur accède à ses notifications, afin de prendre connaissance des alertes relatives aux stocks, aux commandes ou aux dates d'expiration.

Description textuelle

Le tableau ci-dessous illustre la description textuelle des fonctionnalités de CU « Consulter notification ».

CU :	Consulter notifications
BUT :	Permettre à l'utilisateur (administrateur ou magasinier) de consulter les notifications générées par le système (rupture de stock, commande reçue, validation de commande, livraison fournisseur, etc.).
Acteur principal :	Administrateur
Acteur secondaire :	/
Préconditions :	L'utilisateur est authentifié.
Scénario nominal :	Consulter les notifications : <ol style="list-style-type: none">1. L'utilisateur accède à la section « Notifications ».2. Le système affiche la liste des notifications disponibles (non lues et déjà lues).3. L'utilisateur sélectionne une notification pour en consulter le détail.4. Le système marque la notification comme lue.5. L'utilisateur revient à la liste des notifications.
Enchaînements alternatifs :	2.a : Aucune notification disponible <ul style="list-style-type: none">– Le système affiche un message indiquant qu'il n'y a aucune notification à consulter.– La séquence se termine.
Post-condition :	Les notifications sont consultées et celles ouvertes sont marquées comme lues.

Table 5.7 – Description textuelle du cas d'utilisation « consulter notification »

Diagramme de Séquence de CU « Consulter notification »

La figure ci-après présente le diagramme de séquence correspondant au CU « Consulter notification ».

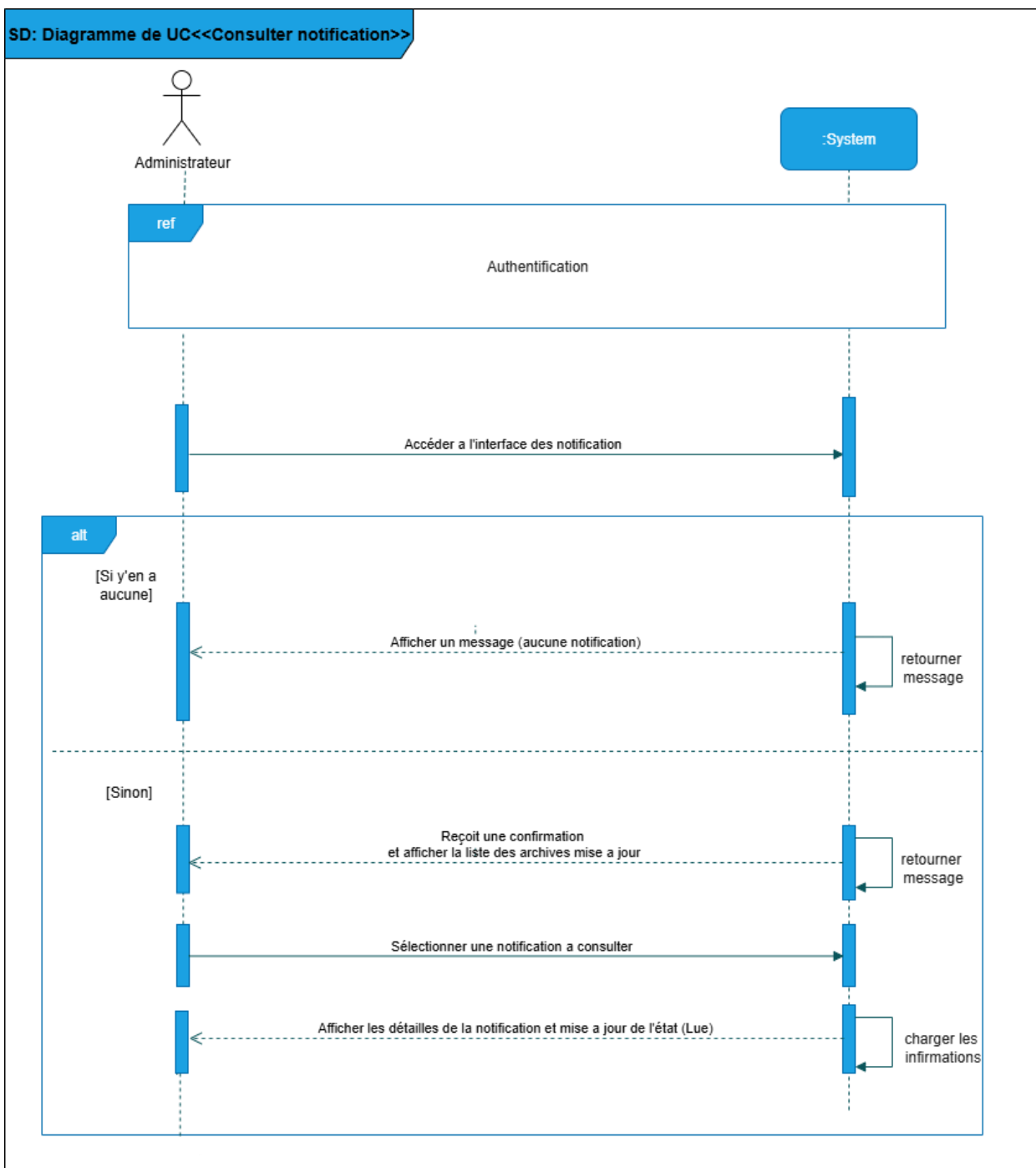


Figure 5.12 – Diagramme de Séquence de CU « Consulter notification »

5.8.1 Le dictionnaire de données du sixième sprint

Cette section présente le dictionnaire de données élaboré lors du sixième sprint, en détaillant la structure des tables, les attributs définis ainsi que leurs relations, afin de garantir une modélisation cohérente du système.

Table : Notifications	
id	INT (11), clé primaire
type	VARCHAR (50)
message	VARCHAR (255)
date_n	DATETIME
lu	BOOLEAN
role_cible	ENUM('admin', 'magasinier')

Table 5.8 – Le dictionnaire de données du sixième sprint

5.9 Les interfaces graphiques du sixième sprint

La figure 5.12 ci-dessus illustre l'interface de la gestion des archives.

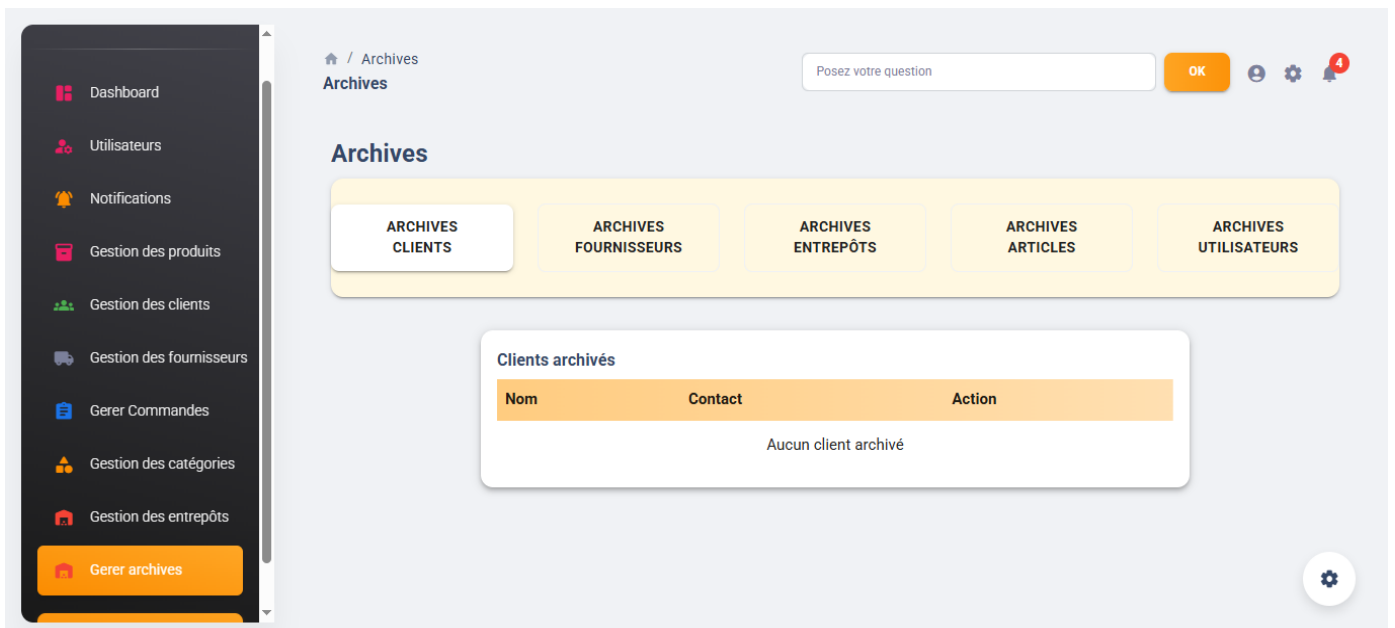


Figure 5.14 – Interface de gestion des archives

5.10 Outil de développement

Nous présentons ci-dessous les outils de développement de notre application.

- **Visual Studio Code** : est un éditeur de code source et un environnement de développement intégré (IDE) de Microsoft. Il est open-source et cross-platform, c'est-à-dire qu'il fonctionne

sur Windows, Linux et Mac. Il a été conçu pour les développeurs web, mais il prend en charge de nombreux autres langages de programmation tels que C++, C, Python, Java, etc. Il offre de nombreuses fonctionnalités comme la coloration syntaxique, l'autocomplétion, la mise en évidence des erreurs, la navigation de code, le débogage, la gestion de versions, l'intégration avec Git, et beaucoup d'autres. Il est également extensible à l'aide d'une grande variété d'extensions développées par la communauté, permettant aux développeurs de personnaliser l'éditeur selon leurs besoins.[10]

- **HTML** : signifie « HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure.[11]
- **CSS** : (Cascading Style Sheets) est un langage de feuille de style utilisé pour décrire la présentation d'un document HTML (HyperText Markup Language). Les feuilles de style CSS permettent de définir la mise en forme des éléments HTML, tels que les polices, les couleurs, les marges et les espacements.[12]
- **JS** : est un langage de programmation qui permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, et tout ce à quoi on peut penser. Bon, peut-être pas tout, mais vous pouvez faire bien des choses avec quelques lignes de JavaScript.[11]
- **Xampp** : est un ensemble de logiciels permettant de facilement créer une interface web interagissant avec une base de données SQL.[14]
- **Node.js** : Node.js est une plateforme de développement Javascript. Ce n'est pas un serveur, ce n'est pas un framework, c'est juste le langage Javascript avec des bibliothèques permettant de réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connections réseau ou encore créer un fichier.[13]
- **React JS** : est une bibliothèque JavaScript open-source qui est utilisée pour construire des interfaces utilisateur spécifiquement pour des applications d'une seule page. Elle est utilisée pour gérer la couche d'affichage des applications web et mobiles.[10]

5.11 Conclusion

Ce Release marque une étape importante dans le développement de notre application, en offrant des outils essentiels permettant aux administrateurs et aux magasiniers de gérer efficacement les commandes, les archives et les notifications. Elle assure une gestion fluide des stocks et des flux de commandes, tout en restant stable et fonctionnelle pour répondre aux besoins évolutifs des utilisateurs.

Conclusion générale et perspectives

Ce travail a permis de répondre à une problématique réelle rencontrée par l'entreprise Agrana Fruit, à savoir l'absence d'un outil simple et efficace pour gérer ses stocks et assurer un meilleur suivi des mouvements et des entrepôts. L'application développée offre une solution web et mobile intuitive, intégrant des fonctionnalités de gestion complète des articles, lots, commandes, clients et fournisseurs, tout en garantissant la traçabilité et la fiabilité des données.

L'adoption de la méthodologie agile *Scrum* et l'utilisation d'UML ont facilité la conception progressive du système et assuré une organisation claire des composants. De plus, la modularité du projet ouvre la voie à une intégration future avec SAP B1 grâce aux APIs et services tiers, ce qui constitue une perspective d'évolution importante.

Perspectives d'amélioration :

- Bien que l'application réponde déjà aux besoins prioritaires de l'entreprise, plusieurs pistes d'évolution peuvent être envisagées afin d'enrichir son efficacité et son ergonomie :
- Intégration d'un module de messagerie interne permettant une communication fluide et rapide entre l'administrateur et les magasiniers.
- Ajout d'un système de lecture de codes-barres ou QR codes afin de faciliter l'enregistrement des entrées/sorties et réduire les erreurs de saisie.
- Évolution vers une solution cloud afin d'assurer une meilleure accessibilité, la synchronisation en temps réel et une montée en charge facilitée.

Réaliser ce projet a été une expérience très enrichissante et motivante pour nous. Il nous a permis de mettre en pratique nos connaissances en conception, en programmation et en méthodes de travail acquises durant notre parcours universitaire. Malgré certaines difficultés rencontrées, ce stage nous a aidés à développer de nouvelles compétences et à gagner en confiance. Cette expérience restera une étape importante qui nous sera certainement utile pour notre avenir professionnel.

Bibliographie

- [1] Chapitre II gestion stock.pdf [En ligne]. Disponible sur : <https://fac.umc.edu.dz/ista/pdf/cours/Chapitre%20II%20gestion%20stock.pdf> [Consulté le 28 février 2025].
- [2] petite-entreprise.net [En ligne]. Disponible sur : <https://www.petite-entreprise.net/P-1338-136-G1-quels-sont-les-differents-types-et-les-finalites-des-stocks.html>[Consulté le 28 février 2025].
- [3] wearephenix.com [En ligne]. Disponible sur : <https://www.wearephenix.com/pro/types-de-stock> [Consulté le 28 février 2025].
- [4] bkssystemes.fr [En ligne]. Disponible sur : <https://www.bkssystemes.fr/blog/gestion-stock-241?utm> [Consulté le 28 février 2025].
- [5] Roques. *UML2 par la pratique, 5ème édition*. Paris : Éditions Eyrolles, 2006.
- [6] bkssystemes.fr [En ligne]. Disponible sur : <https://www.bkssystemes.fr/blog/gestion-stock-241?utm> [Consulté le 2 mars 2025].
- [7] ideematic.com [En ligne]. Disponible sur : <https://www.ideematic.com/actualites/2015/01/methodes-agiles-definition/>
- [8] [Consulté le 5 mars 2025].
- [9] lucidchart.com [En ligne]. Disponible sur : <https://www.lucidchart.com/pages/fr/langage-uml> [Consulté le 6 mars 2025].
- [10] irif.fr [En ligne]. Disponible sur : <https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html>[Consulté le 12 mars 2025].
- [11] openclassrooms.com [En ligne]. Disponible sur : <https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/7847928-decouvrez-comment-fonctionne-une-architecture-mvc> [Consulté le 13 mars 2025].
- [12] bility.fr [En ligne]. Disponible sur : <https://bility.fr/definition-visual-studio-code/> [Consulté le 12 mars 2025].
- [13] developer.mozilla.org [En ligne]. Disponible sur : <https://developer.mozilla.org/>

- [fr/docs/Web/HTML](#)[Consulté le 13 mars 2025].
- [14] bility.fr [En ligne]. Disponible sur : <https://bility.fr/definition-css/>[Consulté le 15 mars 2025].
- [15] makina-corpus.com [En ligne]. Disponible sur : <https://makina-corpus.com/frontend/introduction-nodejs>[Consulté le 15 mars 2025].
- [16] services.montefiore.uliege.be [En ligne]. Disponible sur :https://services.montefiore.uliege.be/verif/cours/bd/repet2014/tp7_slides.pdf [Consulté le 15 mars 2025].
- [17] Bougheriou, Mohand Arezki et Chekar, Billal. Recherche des causes à l'origine des pertes de pression dans des conteneurs de stockage. Mémoire de formation BTS Chimie industrielle. INSFP SAHLI MOHAND CHERIF, sidi aich,2020.
- [18] Agile Thinker Notes [En ligne]. Disponible sur : <https://agilethinkernotes.wordpress.com/2017/11/09/scrum-la-methodes-agile-la-plus-utilisee/>

Annexes

Annexe 1

Langages, environnement et outils de développement

1 GitHub

GitHub est une plateforme open source de gestion de versions et de collaboration destinée aux développeurs de logiciels. Livrée en tant que logiciel à la demande (SaaS, Software as a Service), la solution GitHub a été lancée en 2008. Elle repose sur Git, un système de gestion de code open source créé par Linus Torvalds dans le but d'accélérer le développement logiciel.

GitHub permet aux développeurs de modifier, d'adapter et d'améliorer le logiciel gratuitement à partir de référentiels publics, mais facture les interventions pour les référentiels privés, selon trois formules de paiement. Chaque référentiel public ou privé contient tous les fichiers d'un projet, ainsi que l'historique des révisions de chaque fichier. Plusieurs collaborateurs sont susceptibles de travailler dans les référentiels, qu'ils soient publics ou privés.

2 SQL

Le langage de requête structuré (SQL) est un langage de programmation permettant de stocker et de traiter des informations dans une base de données relationnelle. Une base de données relationnelle stocke les informations sous forme de tableau, avec des lignes et des colonnes représentant différents attributs de données et les diverses relations entre les valeurs de données. Vous pouvez utiliser des instructions SQL pour stocker, mettre à jour, supprimer, rechercher et récupérer des informations de la base de données. Vous pouvez également utiliser le langage SQL pour maintenir et optimiser les performances de la base de données.

3 Bootstrap

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

4 Material_UI

Material-UI est une librairie de composants React qui implémentent les guidelines de Google en terme de Material Design que l'on retrouve dans les interface des applications Google comme Gmail, Google Photos ou encore dans les applications Android.

5 Diagramme de séquence de CU « transférer lot »

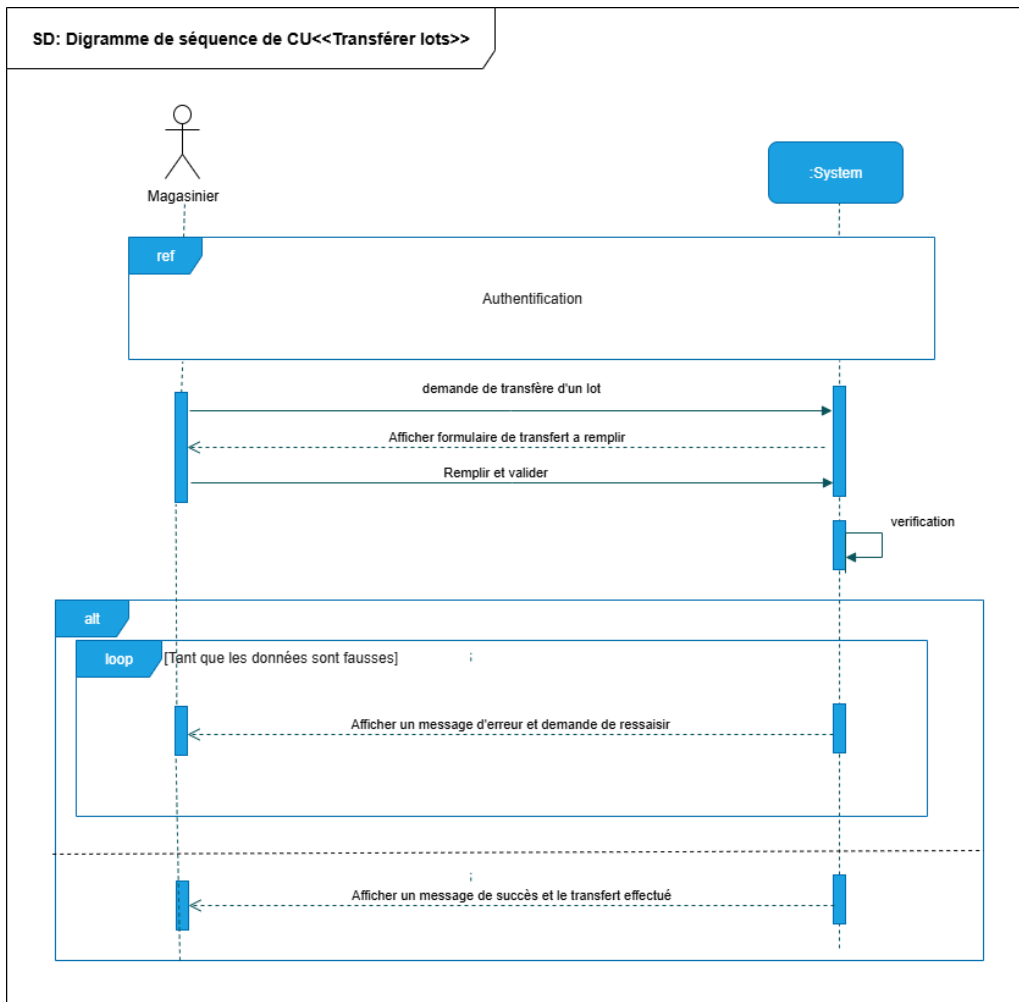


Figure 6.1 – Diagramme de séquence de CU « transférer lot »

6 Digramme d'état transition de CU « Gérer commande (espace ma- gasinier) »

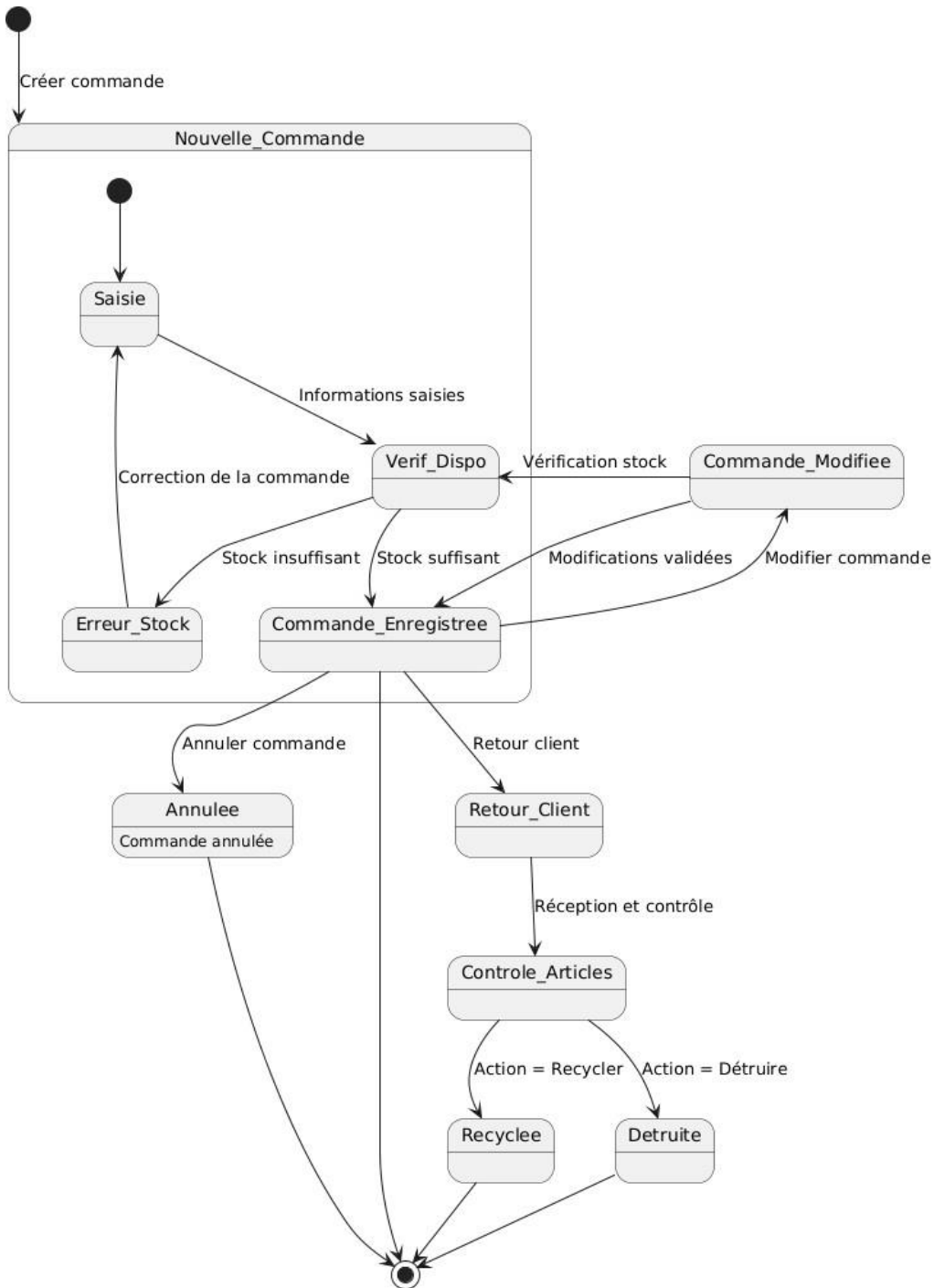


Figure 6.2 – Digramme d'état transition de CU « Gérer commande (espace magasinier)»

Abstract

This project was carried out as part of our final thesis in Software Engineering. Its main objective was to design and develop a stock management application for Agrana Fruit Algérie, a company specialized in the agri-food sector and located in Akbou, Béjaïa. To achieve this, we adopted the Scrum agile methodology and used the Unified Modeling Language (UML) to model and structure the system. The solution is based on a modern architecture with Node.js and Express for the backend, React.js for the frontend, and MySQL as the relational database management system, ensuring reliable and secure handling of stock data. The application provides intuitive and ergonomic interfaces for both administrators and storekeepers, enabling efficient management of articles, lots, stock movements, warehouses, clients, and suppliers. It also integrates advanced features such as lot traceability, monitoring of critical thresholds, and intelligent alerts for stock shortage or product expiration. The aim of this project is to help the company improve its operational efficiency, optimize resource management, and strengthen its competitiveness through a scalable and modular solution that can be easily integrated with enterprise systems such as SAP B1 in the future.

Keywords : Stock management application, Agrana Fruit Algérie, Node.js, Express.js, React.js, MySQL, Scrum methodology, UML, Lot traceability, Critical thresholds, Intelligent alerts, SAP B1.

Résumé

Ce projet a été réalisé dans le cadre de notre mémoire de fin d'études en ingénierie logicielle. Son objectif principal était de concevoir et de développer une application de gestion de stock pour Agrana Fruit Algérie, une entreprise spécialisée dans le secteur agroalimentaire et située à Akbou, Béjaïa. Pour atteindre cet objectif, nous avons adopté la méthodologie agile Scrum et utilisé le langage de modélisation UML afin de modéliser et structurer le système. La solution repose sur une architecture moderne avec Node.js et Express pour le backend, React.js pour le frontend, et MySQL comme système de gestion de base de données relationnelle, garantissant une manipulation fiable et sécurisée des données de stock. L'application offre des interfaces intuitives et ergonomiques destinées aux administrateurs ainsi qu'aux magasiniers, permettant une gestion efficace des articles, des lots, des mouvements de stock, des entrepôts, des clients et des fournisseurs. Elle intègre également des fonctionnalités avancées telles que la traçabilité des lots, le suivi des seuils critiques et des alertes intelligentes en cas de rupture de stock ou d'expiration des produits. L'objectif de ce projet est d'aider l'entreprise à améliorer son efficacité opérationnelle, optimiser la gestion de ses ressources et renforcer sa compétitivité grâce à une solution modulaire et évolutive, pouvant être intégrée ultérieurement avec des systèmes d'entreprise tels que SAP B1.

Mots-clés : Application de gestion de stock, Agrana Fruit Algérie, Node.js, Express.js, React.js, MySQL, Scrum methodology, UML, SAP B1, Traçabilité, alertes intelligentes, SAP B1.

