

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

Université Abderrahmane Mira-Béjaïa

Faculté des Sciences Exactes

Département d'Informatique



Mémoire de fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel

Option : Génie Logiciel

Thème

**Conception et Réalisation d'une Application sous
réseau local pour le suivi des pertes en temps réel pour
l'entreprise DANONE.**



Réalisé Par :

Mlle TIGRINE Hanane

Soutenu devant le jury composé de :

Président: Mr. A.LARBI

Examineur: Mme. MAMMERI

Encadrant : Mr. R.OUZEGGANE

Année 2019/2020

Remerciements

Je remercie en premier lieu Dieu tout puissant qui m'a doté d'une grande volonté, courage et patience pour mener à terme mon projet.

Il m'est particulièrement agréable avant de présenter mon travail, d'exprimer toute ma gratitude envers les personnes qui de près ou de loin m'ont apporté leur soutien.

Ma profonde gratitude et sincères remerciements à mon encadrant **Mr R.OUZEGGANE** qui m'a inculqués une grande confiance et m'a accordé de son temps, ses conseils et m'a orienté dans le bon sens quant à l'élaboration de ce projet.

Je tiens à exprimer toute ma grande gratitude aux membres de jury d'avoir accepté de juger ce travail.

Je remercie également mes chers parents pour tous les sacrifices consentis à mon égard et leur énorme soutien durant ma vie et mon cursus d'études.

Je tiens fortement à exprimer le plaisir que j'ai eu de faire mon stage au sein de l'entreprise **DANONE DJURDJURA Algérie** et je tiens par là à en remercier mon maître de stage **Mr TAMADARTAZA Belaid** et toute l'équipe performance, **Mr Mohammed Berd** et **Mr HAMIMI Khellaf** et **Mme MAAMOURI Hakima** ainsi que tous le personnel de l'entreprise à leur tête le directeur **Mr Djerroud Madjid** pour leur aide précieuse, leur disponibilité et explications quant à la bonne conduite du projet, tout au long de la période du stage.

Mes vifs remerciements s'adressent également à tous mes enseignants de la faculté des sciences exactes de l'université **ABDERRAHMANE MIRA** de **Bejaia** pour la formation qu'ils ont eu le soin de nous apporter le long de notre cursus universitaire

Dédicaces

Je dédie ce travail, à mon papa adoré,

Tu es mon maître et mon idole. J'espère être à la hauteur de tout ce que tu m'apportes dans la vie. Ton soutien sans faille et ta confiance en moi m'ont permis d'arriver là où je suis.

A ma maman chérie,

C'est grâce à ton amour, ton affection, ta protection, et ton soutien que j'ai pu réussir et surmonter les moments difficiles. Que ce travail soit la reconnaissance à tous tes sacrifices.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie.

A ma chère sœur Hakima et ses filles Bylea et Imane,

Merci d'être toujours là pour moi, pour ton amour, ton soutien, tes encouragements et pour l'adorable sœur que tu es. Un avenir radieux à toi et tes filles.

A ma sœur Nawal et son fils Axel,

A mes sœur Rosa et Mouia, à mon frère Toufik et mon petit frère Fares,

Vous êtes le rayon de soleil de ma vie. Je remercie dieu pour la chance que j'ai de vous avoir.

A mon petit amour Dylane et mon cher grand père.

A la mémoire de ma grand-mère, de ma tante Nadia et son mari Bouzid et mon oncle Djamel,

Que dieu vous accueille dans son vaste paradis, et toute la réussite à vos enfants.

A mes ami(e)s Mitchi, Omar, Said, Melissa, Nacera, Cylia et Wiwi,

Merci pour votre présence et pour votre soutien.

A une personne particulière, **Jimmy**,

Tes encouragements et ta présence ont été mon appui. Tu as toujours su me reconforter. Merci pour ton soutien permanent, ta compréhension et ta patience.

Merci pour tout ce que tu m'as apporté.

A mes enseignants.

A tous mes amis et toutes les personnes qui m'ont apporté de l'aide et contribué à la réalisation de ce projet.

Table des Matières

<i>Table des Matière</i>	<i>i</i>
<i>Table des Figures</i>	<i>iv</i>
<i>Liste des Tableaux</i>	<i>vi</i>
<i>Liste des Abréviations</i>	<i>vii</i>
<i>Introduction Générale</i>	<i>1</i>
<i>Chapitre I : Etude préliminaire et Capture des Besoins</i>	<i>3</i>
1. <i>Introduction</i>	<i>3</i>
2. <i>Etude préliminaire</i>	<i>3</i>
2.1. Présentation du projet	<i>3</i>
2.2. Présentation de l'organisme d'accueil	<i>4</i>
2.3. Historique de l'entreprise	<i>4</i>
2.4. Organigramme de l'entreprise <<Danone Djurdjura>>	<i>5</i>
2.5. Description du processus de production	<i>7</i>
2.6. Les différents produits:	<i>7</i>
2.7. Présentation du contexte du projet	<i>8</i>
2.7.1. Situation Informatique	<i>8</i>
2.7.2. Problématiques	<i>8</i>
2.7.3. Objectifs de l'étude	<i>10</i>
2.7.4. Solutions	<i>10</i>
3. <i>L'identification des besoins</i>	<i>11</i>
3.1. Les besoins fonctionnels	<i>11</i>
3.2. Les besoins non fonctionnels	<i>12</i>
4. <i>Conclusions</i>	<i>12</i>
<i>Chapitre II : Analyse des Besoins</i>	<i>14</i>
1. <i>Introduction</i>	<i>14</i>

2. Langage de modélisation et processus de développement	14
3. L'analyse des besoins	15
3.1. Identification des acteurs du système	15
3.2. L'identification des messages	16
3.3. Diagramme de contexte dynamique	19
3.4. Identification des cas d'utilisation	19
3.4.1. Cas d'utilisation par acteur	19
3.4.2. Description textuelle des cas d'utilisation	21
3.4.3. Diagrammes de cas d'utilisation	24
a. Diagramme de cas d'utilisation pour l'acteur « Opérateur process »	24
b. Diagramme de cas d'utilisation pour l'acteur « Opérateur ligne »	25
c. Diagramme de cas d'utilisation pour l'acteur « Administrateur »	25
d. Diagramme de cas d'utilisation global	27
3.5. Développement du modèle dynamique	28
3.5.1. Diagrammes de séquence par scénarios	28
5. Conclusion	34
Chapitre III : Conception et schéma relationnel de données	35
1. Introduction	35
2. Diagrammes de séquence détaillés	35
3. Diagramme de Classes du domaine	42
3.1. Classes, attributs et responsabilités	44
3.2. Passage au modèle relationnel	49
3.2.1. Les règles du passage	49
3.2.2. Le modèle relationnel associé au diagramme de classe de l'application	50
4. Conclusion	51
Chapitre IV : Réalisation	52
1. Introduction	52
2. Langages environnement de développement	52
2.1. Langages de programmation et outils utilisés	52
2.1.1. Java	52
2.1.2. Javafx	53
2.1.3. JFoenix	53

2.1.4.	XAMPP	53
2.1.5.	StarUML	54
2.1.6.	Visual Paradigm	54
2.2.	Environnement de programmation	55
3.	<i>Schéma Physique de la Base de données</i>	56
4.	<i>Architecture de l'application</i>	58
4.1.	Diagramme de déploiement	59
4.2.	Structuration du code source	60
5.	<i>Arborescence de l'application (Diagramme de navigation)</i>	61
6.	<i>Interfaces homme-machines (IHM)</i>	62
7.	<i>Conclusion</i>	69
	<i>Conclusion générale</i>	70
	<i>Annexe A</i>	I
	<i>Annexe B</i>	IV
	<i>Bibliographie et Webographie</i>	XXV

Table des Figures

Figure 1: Organigramme de <<DANONE DJURDJURA>>	6
Figure 2: Diagramme de contexte dynamique	19
Figure 3: Cas d'utilisation "s'authentifier"	22
Figure 4: Diagramme de cas d'utilisation pour l'acteur « Opérateur process »	25
Figure 5: Diagramme de cas d'utilisation pour l'acteur « Opérateur ligne »	25
Figure 6: Diagramme de cas d'utilisation pour l'acteur « Administrateur »	26
Figure 7: Diagramme de cas d'utilisation global	27
Figure 8: Diagramme de séquence "Authentification"	29
Figure 9: Diagramme de séquence du cas d'utilisation « Saisir propriétés lait »	30
Figure 10: Diagramme de séquence du cas d'utilisation « Renseigner dosage »	31
Figure 11: Diagramme séquence du cas d'utilisation « Gestion des utilisateurs »	32
Figure 12: Diagramme séquence du cas d'utilisation « Suivi produit »	33
Figure 13: Diagramme séquence du cas d'utilisation « Suivi pertes »	34
Figure 14: Diagramme séquence d'interaction « S'authentifier »	37
Figure 15: Diagramme séquence d'interaction « Saisir propriétés lait »	38
Figure 16: Diagramme séquence d'interaction « Ajouter utilisateur »	39
Figure 17: Diagramme séquence d'interaction « Modifier utilisateur »	40
Figure 18: Diagramme séquence d'interaction « Supprimer utilisateur »	41
Figure 19: Diagramme séquence d'interaction «Gestion des utilisateurs »	42
Figure 20: Diagramme de classes	43
Figure 21: Schéma physique de la base de données	57
Figure 22: Architecture de l'application	59
Figure 23: Diagramme de déploiement	60
Figure 24: Structure de code source	61
Figure 25: Arborescence de l'application	62
Figure 26: Interface "Authentification "	63
Figure 27: Interface "Lait " de l'opérateur process	64
Figure 28: Interface "Lait " de l'opérateur process	64
Figure 29: Interface "Ecrémage " de l'opérateur process	65
Figure 30: Interface "Ecrémage " de l'opérateur process	65
Figure 31: Interface "Masse Blanche " de l'opérateur process	66
Figure 32: Interface "Gestion des utilisateurs " de l'administrateur	66
Figure 33: Interface "Ajouter utilisateur " de l'administrateur	67
Figure 34: Interface "Modifier utilisateur " de l'administrateur	68
Figure 35: Interface "Gestion des ingrédients " de l'administrateur	68
Figure 36: Les diagrammes d'UML	II
Figure 37: Diagramme de séquence du cas d'utilisation « Saisir informations écrémage »	XIII
Figure 38: Diagramme de séquence du cas d'utilisation « Saisir propriétés masse blanche »	XIV
Figure 39: Diagramme de séquence du cas d'utilisation « Renseigner fiche démarrage ligne »	XV
Figure 40: Diagramme de séquence du cas d'utilisation « Saisir control arôme »	XVI

Figure 41: Diagramme de séquence du cas d'utilisation « Saisir control emballage »	XVII
Figure 42: Diagramme de séquence du cas d'utilisation « Saisir pertes fruit »	XVIII
Figure 43: Diagramme de séquence du cas d'utilisation « Saisir pertes emballage »	XIX
Figure 44: Diagramme de séquence du cas d'utilisation « Saisir pertes arôme »	XX
Figure 45: Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur »	XXI
Figure 46: Diagramme séquence du cas d'utilisation « Modifier un utilisateur »	XXII
Figure 47: Diagramme séquence du cas d'utilisation « Supprimer un utilisateur »	XXIII
Figure 48: Diagramme séquence du cas d'utilisation « Suivi lait »	XXIV

Liste des Tableaux

Tableau 1: Messages échangés entre les acteurs et le système.....	18
Tableau 2: Identification des cas d'utilisation.....	21
Tableau 3: Description de cas d'utilisation "s'authentifier".....	22
Tableau 4: Description de cas d'utilisation "Saisir les propriétés lait".....	23
Tableau 5: Description de cas d'utilisation "Renseigner dosage".....	23
Tableau 6: Description de cas d'utilisation "Gestion des utilisateurs".....	24
Tableau 7: Description de cas d'utilisation "Suivi produits finis".....	24
Tableau 8: Les stéréotypes du Diagramme d'interaction « S'authentifier ».....	38
Tableau 9: Dictionnaire de données.....	48
Tableau 10: Description du cas d'utilisation <<Saisir les informations de l'écrémage>>.....	IV
Tableau 11: Description du cas d'utilisation <<Saisir les informations masse blanche>>.....	V
Tableau 12: Description du cas d'utilisation <<Renseigner fiche démarrage ligne >>.....	V
Tableau 13: Description du cas d'utilisation <<Control arôme >>.....	VI
Tableau 14: Description du cas d'utilisation <<Control emballage>>.....	VI
Tableau 15: Description du cas d'utilisation <<Saisir les pertes en emballage >>.....	VII
Tableau 16: Description du cas d'utilisation << Saisir les pertes en arôme >>.....	VII
Tableau 17: Description du cas d'utilisation <<Saisir les pertes en fruit>>.....	VIII
Tableau 18: Description du cas d'utilisation << Gestion lignes >>.....	VIII
Tableau 19: Description du cas d'utilisation << Gestion produits >>.....	IX
Tableau 20: Description du cas d'utilisation << Gestion emballage >>.....	IX
Tableau 21: Description du cas d'utilisation << Gestion masse blanche >>.....	X
Tableau 22: Description du cas d'utilisation << Gestion ingrédients >>.....	X
Tableau 23: Description du cas d'utilisation << Suivi lait >>.....	XI
Tableau 24: Description du cas d'utilisation << Suivi masse blanche >>.....	XI
Tableau 25: Description du cas d'utilisation << Suivi ingrédients >>.....	XII
Tableau 26: Description du cas d'utilisation << Suivi emballage>>.....	XII

Liste des Abréviations

AGL: Atelier de Génie Logiciel

CRUD: Create-Read-Update-Delete

FXML: FX Markup Language

HTML: HyperText Markup
Language.

HTTP: Hypertext Transfer Protocol

IHM: Interface Homme-Machines.

IDE: Integrated Developement
Environnement

JDK: Java Development Kit

JRE: Java Runtime Environment

OMG: Object Management Group

LAN: Local Area Network

BPS: Bande Polystyrène

MOF: Meta Object Facility

MySQL: Structured Query Language

PHP: Hypertexte PréProcesseur

RUP: Rational Unified Process

SGBD: Système de Gestion de Base
de Données

SPP: Suivi des Pertes de Production

UML: Unified Modeling Language.

UP: Unified Process

XAMPP: cross-platform, Apache,
MySQL, PHP and Perl

XP: eXtreme Programming

2TUP: Two Tracks Unified

Introduction Générale

INTRODUCTION

GENERALE

Introduction Générale

Actuellement, le monde connaît une avancée technologique considérable dans tous les secteurs et cela à l'aide de l'informatique, qui joue un rôle primordial dans le développement des entreprises et organisations.

Vu les avantages d'automatisation des systèmes d'information, l'exploitation des technologies avancées de traitement de l'information est devenue une nécessité dans la gestion des entreprises et cela pour assurer une bonne démarche et une gestion efficace de leurs ressources, par l'utilisation d'un logiciel.

Comme toute entreprise qui consacre de grands efforts pour le perfectionnement de la qualité de ses activités et l'amélioration de son rendu et de sa performance, DANONE DJURDJURA Algérie est une multinationale qui a su assurer sa place dans le marché Algérien et veut encore d'avantage se distinguer en informatisant les tâches qui restent encore manuelles et avoir un système du suivi des pertes de production afin de pouvoir détecter plus rapidement les causes et les résoudre.

Le service performance est le premier concerné, vu qu'un tel système vise à améliorer la performance l'entreprise, vu qu'il aide à la prise de décision grâce aux statistiques et les résultats calculés.

Donc l'informatique doit être capable tout simplement de calculer et afficher les statistiques du suivi de production, en pertes et en consommation, détecter les pannes afin de pouvoir les réparer plus vite et assurer l'objectif zéro papier en permettant la saisie directe des suivi sur l'application.

Pour faire face aux problèmes posés, nous allons proposer une solution informatique que nous avons appelée "*SPP*" (*Système de Pertes de Production*), qui

doit assurer le bon suivi des pertes tout au long du processus de production depuis la réception du lait jusqu'à l'obtention du produit fini.

Pour avoir ce produit logiciel nous nous sommes basés sur un ensemble de méthodes et de langages. Pour la modélisation, nous avons utilisé une méthode qui se situe à mi-chemin entre UP (Unified Process), et la modélisation en UML, ainsi que le langage de programmation JAVA avec le SGBD MariaDB sous XAMPP.

Ce mémoire est organisé en quatre chapitres, comme suit:

- **Chapitre 1:** Étude Préliminaire et Capture des Besoins, qui est dédié pour la présentation de l'organisme d'accueil DANONE DJURDJURA, l'objectif de notre travail, la problématique et la solution ainsi que les besoins fonctionnels et non fonctionnels de notre application.

- **Chapitre 2:** Analyse des besoins, où nous déterminerons les acteurs qui interagissent avec le système à développer, ainsi que les cas d'utilisation correspondant à chaque acteur, qui aboutiront au diagramme de cas d'utilisation. Ces différents cas d'utilisation seront décrit ensuite par des diagrammes de séquences. Nous allons aussi définir la démarche de développement suivi et le langage de modélisation UML.

- **Chapitre 3:** Conception, qui présente la conception qui explique comment commencer à concevoir le système, indépendamment de son contenu fonctionnel et décrit le passage du modèle objet au modèle relationnel en réalisant le diagramme de classe et avant ça les diagrammes de séquence détaillés qui exposent l'interaction utilisateur/différentes parties du systèmes.

- **Chapitre 4:** Réalisation, qui présente les détails relatifs à la réalisation du système en présentant la description du logiciel, les différents outils ainsi que les langages de programmation utilisés.

En plus, nous présenterons le diagramme de déploiement qui montre les différents composants de l'application et leurs relations. Et nous terminons ce chapitre avec l'arborescence de l'application ainsi que quelques interfaces homme-machine.

Et enfin une conclusion générale qui résume l'essentiel du notre travail.

Chapitre I

Etude préliminaire et
Capture des Besoins

Chapitre I

Etude préliminaire et Capture des Besoins

1. Introduction

Dans ce chapitre, nous allons premièrement considérer le système comme une boîte noire à part entière. L'étude préliminaire est la première étape de notre processus, où nous décrirons le contexte du projet, nous poserons la problématique, la solution proposée, et nous exposerons les objectifs de notre travail. L'étape capture des besoins fonctionnels formalise et détaille ce qui a été ébauché au cours de l'étape précédente, enfin la capture des besoins non fonctionnels couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description de métier des utilisateurs, ni de la description applicative.

Notre objectif dans cette étape est donc d'exprimer les besoins attendus du futur système à développer en utilisant le langage naturel.

2. Etude préliminaire

L'étude préliminaire va nous servir à poser les bases de la capture des besoins du système à réaliser. Elle est la toute première étape de notre processus de développement. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels. Elle prépare les étapes les plus formelles de capture des besoins fonctionnels et techniques [1].

2.1. Présentation du projet

Notre projet consiste à développer une application sous réseau local pour le suivi des pertes au niveau de la production de l'entreprise Danone Djurdjura, et ce, dans le but d'améliorer le rendu, faciliter la tâche de saisie des opérateurs et détecter

les zones de pertes et ainsi être plus efficace dans la résolution des problèmes rencontrés et gagner en performance.

2.2. Présentation de l'organisme d'accueil

Danone est une multinationale alimentaire française qui a été fondée par Isaac Carasso en 1919 à Barcelone, en Espagne et dont le siège social est à Paris.

Au fil des années, l'entreprise se décline autour de quatre pôles : produits laitiers frais et d'origine végétale, eaux en bouteille, nutrition médicale et nutrition infantile.

Présente sur plus de 120 marchés, en 2017 ses ventes se sont élevées à €24,68 milliards (\$30,80 milliards), avec plus de la moitié dans des pays émergents, faisant d'elle la 5e entreprise agroalimentaire du monde. Le groupe emploie 100 000 personnes à travers le monde, et compte plus de 180 sites de production.

Danone Djurdjura Algérie est le résultat d'un accord de partenariat entre le leader mondial des produits laitiers frais << Groupe DANONE >> et de la laiterie DJURDJURA, leader du marché Algérien des produits laitiers frais (PLF) en prenant une participation de 51% dans la société << DANONE. JURDJURA ALGERIE SPA (DDA) >>.

DANONE DJURDJURA ALGERIE est implantée : Dans une zone industrielle <<TAHARCHT >> véritable carrefour économique de Bejaia, de quelques 50 unités de productions agroalimentaires et en cours d'expansion. Le directeur d'usine actuel est Mr HAYA Madjid. C'est une société par action au capital de 7900 200 000 DA.

2.3. Historique de l'entreprise

Danone est issue de la fusion, en 1973, entre Danone et Gervais (fusion qui date de 1967) et le groupe français Boussois-Souchon-Neuvesel (connu sous le sigle BSN et issu, à son tour, en 1966, de la fusion de l'entreprise de glaces Boussois et de la verrerie Souchon-Neuvesel). En 1994, il a été décidé de donner au groupe ainsi formé en 1973 le nom de sa marque de produits frais : Danone.

En 1997, le groupe a engagé un important programme de recentrage sur trois métiers prioritaires à vocation mondiale (produits laitiers frais, Boisson et Biscuits, Snacks céréaliers) qui représentent 77% du chiffre d'affaire, le groupe DANONE est le

premier producteur mondial de produits frais, le second producteur mondial de Biscuits et Snacks céréalier et le premier producteur d'eau conditionnée.

En Algérie au terme des accords, le groupe Danone a également conclu un accord de partenariat avec laiterie DJURDJURA, leader du marché des produits laitiers frais (PLF) en prenant une participation de 51% dans la société DANONE DJURDJURA ALGERIE SPA(DDA).

Après l'année 2002 consacrée à rénover le site d'Akbou et à mettre en place des outils industriels nécessaires à l'expansion future, la marque DANONE a été lancée en août 2002.

En 2006 exactement en mois de juillet<< DANONE DJURDJURA>> est devenu << SPA DANONE>> avec 95%.

Les 5% restantes pour la famille Batouche.

2.4. Organigramme de l'entreprise <<Danone Djurdjura>>

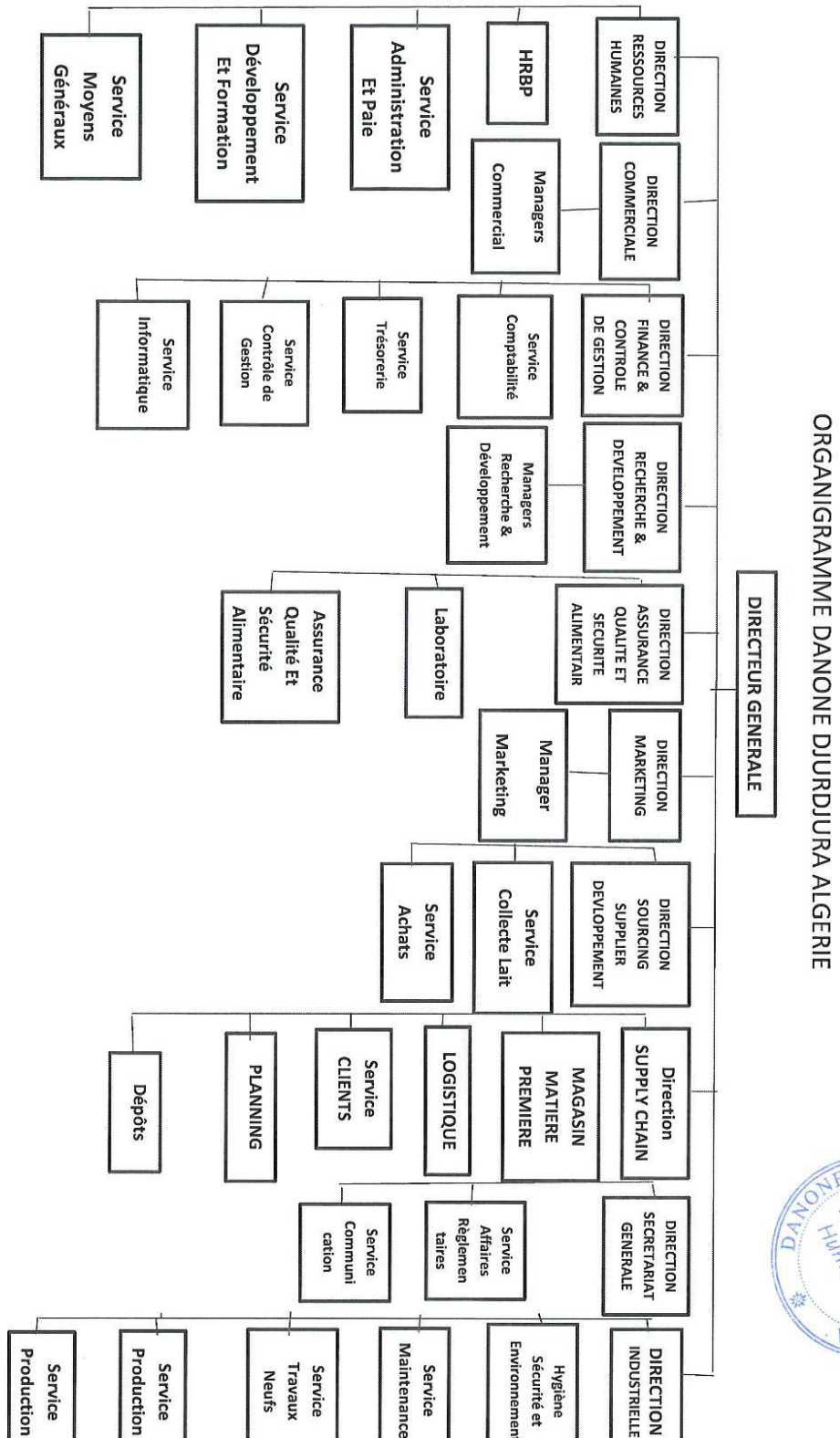


Figure 1: Organigramme de <<DANONE DJURDJURA>>

2.5. Description du processus de production

Un pot de yaourt ou une bouteille de jus ou alors d'eau, est la finalité d'un long processus de production qui passe par plusieurs étapes et cela dépend du type du produit. En effet, les deux phases primordiales de la production restent les mêmes pour, presque, tous les produits : La phase process et la phase conditionnement.

- **La phase process** : La finalité de cette phase est : la masse blanche (un yaourt non aromatisé non fruité). La phase commence dès la réception du lait de vache frais qui sera d'abord testé pour assurer sa conformité. Puis sera stocké, pour passer ensuite à l'écumeuse pour faire l'écémage, où le lait sera séparé de la crème fraîche et stockés séparément. L'étape suivante est celle de la préparation de la masse blanche où tous les ingrédients d'une recette seront mélangés, y compris le lait bien évidemment, pour obtenir une masse blanche d'un produit selon la recette voulu. Ainsi se termine la phase process avec l'envoi de la masse produite vers les ateliers pour le conditionnement.
- **La phase Conditionnement** : Dans cette phase, se fait le packaging, ou encore emballage du produit. La masse blanche, précédemment produite, arrive donc dans les lignes de conditionnement où elle sera mise dans son package, qui est produit en parallèle sur les lignes. Les pots sont donc formés, le décor déposé autour, la masse blanche versée à l'intérieur, l'arôme ou le fruit ajouté, le couvert collé et enfin on obtient le pot fini.

C'est donc ainsi que se termine le processus de production.

2.6. Les différents produits:

L'Unité DANONE DJURDJURA Algérie produit 350 à 400 tonnes/jour.

Ses différents produits sont :

- Yaourt ferme traditionnel.
- Seven bénéfiques.
- Bioactivia aromatisé.
- Bioactivia aux fruits.
- Crème dessert (DANETTE).
- Yaourt fruité (fruits).
- Yaourt à boire (Activia drink, Danino drink).
- Jus (Danao).
- Petit Gervais nature.
- Petit Gervais aux fruits.

2.7. Présentation du contexte du projet

2.7.1. Situation Informatique

Danone Djurdjura est l'une des entreprises les plus avancées dans le domaine informatique parmi ses concurrents. Elle a opté pour le système SAP qui est un progiciel de gestion intégré développé et commercialisé par l'éditeur de ce produit (SAP AG).

Les logiciels SAP reposent aujourd'hui sur une architecture technique commune appelée SAP NetWeaver dont le principal composant est le Web AS (Web Application Server).

Un progiciel de gestion intégré (PGI) peut être défini comme un système dans lequel les différentes fonctions de l'entreprise (comptabilité, finances, production, approvisionnement, marketing, ressources humaines, qualité, maintenance, etc.) sont reliées entre elles par l'utilisation d'un système d'information centralisé sur la base d'une configuration client/serveur.

En effet, son système permet de gérer une grande partie de l'entreprise mais ça devient vite compliqué lorsque ça concerne la saisie de données ou le calcul de pertes et l'affichage de résultat sous des formes graphiques précises. Vu que sur le système, ce n'est pas tout le monde qui a un compte, c'est payant. Du coup, la saisie devient une charge. C'est soit un opérateur qui saisit les données sur papier et un autre possédant un compte les ressaisit dans le système ou alors c'est saisie sur des fichiers Excel accessible uniquement sous le réseau et non intégré dans le système SAP.

En ce qui concerne le matériel informatique, chaque bureau dispose de pc bureau et chaque employé possède son propre pc portable. L'entreprise dispose également d'imprimantes, scanners, photocopieuses et tout le matériel nécessaire.

2.7.2. Problématiques

Le suivi des pertes de production en produit et en emballage est l'un des concepts les plus importants chez Danone. En effet, un suivi non précis ne permet pas de détecter la source ni l'instant de la perte et engendre ainsi des pertes importantes en coût et en temps.

Actuellement, le suivi n'est pas effectué de manière efficace, il n'est ni précis ni global. C'est un suivi en fonction de l'ensemble des fichiers assemblés, qui sont soit des fiches papiers remplis par les opérateurs ou alors des fichiers Excel remplis aussi par les opérateurs que l'administrateur doit utiliser pour calculer le total des pertes manuellement.

L'interrogation de personnel du service performance de Danone Djurdjura nous a permis de mieux appréhender la problématique et de connaître quelques anomalies, ce qui nous a permis d'extraire les points suivants comme problèmes à prendre en considération pour résolution dans le futur logiciel :

- Volume important des informations traitées manuellement, ce qui provoque parfois des erreurs dans l'établissement des documents et des calculs.
- Recherche difficile sur le système en ce qui concerne les pertes, ce qui engendre une perte de temps.
- Utilisation de logiciels à part afin de calculer les pertes et avoir des statistiques et des graphes, ce qui peut rendre des résultats erronés lorsque les données ne sont pas exactes.
- Possibilité d'erreur dans le remplissage des différents documents manuellement ou lors de leur ressaisie sur le système.
- Possibilité d'erreur dans les calculs des statistiques.
- Nombre important des archives qui engendre une difficulté de stockage.
- Détérioration des archives à force de leur utilisation très fréquente.
- La difficulté de la communication entre les différents services Production, Performance, qualité...).
- Difficulté de gérer les pertes par produit, par quart et par type d'emballage, quand c'est important.
- Non existence de notion de temps réel dans le suivi des pertes (On doit attendre la saisie manuelle puis sur machine des données afin de pouvoir les utiliser pour des statistiques).
- Difficultés majeures dans la gestion des comptes utilisateurs (les opérateurs n'ont pas de compte sur le système, du coup, ça fera un double travail, l'opérateur saisie sur papier et un utilisateur ayant un compte va ressaisir sur le système).

Pour ces raisons, nous avons donc décidé d'opter pour une application sous réseau local qui permettra de résoudre tous ces problèmes.

2.7.3. Objectifs de l'étude

Afin de remédier à tous ces problèmes, nous avons assigné à notre étude les objectifs suivants :

- Rapidité dans l'établissement des différents documents.
- Facilité de la recherche et l'accès aux informations.
- Stockage des informations sur des supports informatiques tel que des documents Excel et PDF, ce qui assurera leur sécurité.
- Gain de temps dans les calculs des statistiques.
- Minimiser le temps d'accès, ce qui rend l'information efficace.
- Sécuriser et protéger les informations par l'utilisation des mots de passes.
- Automatiser les tâches qui se traitent manuellement.
- Facilite la communication entre les services en exploitant un réseau local entre eux.
- Automatiser le calcul des pertes grâce aux formules déjà existantes, elles seront donc automatiquement calculée dans l'application.
- Pouvoir calculer les pertes en filtrant les résultats (par produits, par quart, par équipe, par type d'emballage...).
- Projection des statistiques (de l'application) en temps réel sur un écran téléviseur.

2.7.4. Solutions

Pour remédier à ces problèmes soulevés et atteindre les objectifs fixés, il doit y avoir un suivi des pertes par phases et par période de temps le tout intégré dans un même système afin d'avoir un suivi et une gestion performante, facile et efficace.

Nous allons donc réaliser une application sous réseau local, pour le suivi des pertes en temps réel.

L'application permettra de gérer les produits, les utilisateurs et les lignes et fera les calculs de pertes en temps réel à partir des données saisies par les opérateurs pour afficher à l'administrateur les statistiques.

3. L'identification des besoins

3.1. Les besoins fonctionnels

Dans cette section, nous représentons l'ensemble des besoins fonctionnels auxquels devrait répondre notre application.

Les besoins fonctionnels et les attentes par rapport à notre application dépendent de la nature de l'acteur.

Les besoins fonctionnels auxquels notre application doit répondre se résument dans les points suivants :

- L'authentification des utilisateurs par un login et un mot de passe pour accéder aux différentes fonctionnalités.
- Avoir une base de données pour le stockage des produits et des données saisies par les utilisateurs.
- Manipulation et mise à jour de la base de données par l'administrateur de l'application.
- L'administrateur peut ajouter, supprimer, modifier ses données, avoir un suivi de l'évolution de la production en terme de pertes ou de gains que ce soit en masse blanche, bps (Bande Polystyrène), opercule ou décor pour chaque produit, tandis que les opérateurs peuvent uniquement saisir les données (pertes ou propriétés produit) et s'inscrire.
- L'existence de la notion de temps réel (les données saisies par les opérateurs doivent s'afficher sous forme de statistiques pour l'administrateur instantanément).
- Avoir un historique des pertes et pouvoir faire des suivis 'hebdomadaires, mensuels ou annuels).
- Les actions réalisées par les opérateurs seront enregistrées avec le nom de l'utilisateur qui l'a fait.

3.2. Les besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation comme le langage de programmation, le type du SGBD et du système d'Exploitation [2].

La future application doit répondre aux besoins non-fonctionnels suivants :

- **Ergonomie de l'interface** : L'application doit être facile à utiliser, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur afin de mieux gérer son espace de travail.
- **Fiabilité** : Les informations apportées par l'application doivent être fiables et sûres (Il faut garantir la sécurité d'accès à l'espace administrateur afin de protéger les données confidentielles de l'entreprise (recettes+coûts)).
- **Disponibilité** : L'application doit être disponible à tout instant pour être utilisée par n'importe quel utilisateur (L'application doit permettre de gérer les accès des utilisateurs selon un privilège et un état d'activation de chaque compte).
- **Sécurité** : L'application comporte des informations personnelles et sensibles, donc elle doit respecter les règles relatives à la sécurité des systèmes informatiques (Prévenir contre l'accès direct avec les liens URL et définir un délai de temps pour la fermeture de session non active).
- **La performance** : L'application doit être performante c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale.

4. Conclusions

Dans ce chapitre, nous avons décrit notre organisme d'accueil, puis le contexte de notre projet où nous avons posé la problématique et tracé quelques objectifs, et la solution proposée. Ensuite, nous avons établi aussi une étude des systèmes existants qui nous permettent de connaître les fonctionnalités primordiales. Enfin, nous avons capturé les différents besoins fonctionnel et technique de l'application.

Dans le chapitre suivant, nous allons entamer la phase d'analyse des besoins.

Chapitre II

Analyse des Besoins

Chapitre II

Analyse des Besoins

1. Introduction

Depuis quelques années, la modélisation objet avec le langage UML est devenue incontournable pour la plupart des projets informatiques, c'est pourquoi nous l'appliquerons à notre projet. En effet, l'UML nous permet de maîtriser la situation et de la rendre plus confortable avec ses différents diagrammes.

Nous traitons dans ce chapitre l'analyse fonctionnelle de notre projet, où nous utilisons le langage UML. D'abord nous identifions les acteurs impliqués, ensuite nous spécifions les messages échangés dans le système ainsi que le diagramme de contexte. Nous passerons ensuite aux cas d'utilisation et diagrammes de cas d'utilisation pour finir la partie analyse avec les diagrammes de séquence.

2. Langage de modélisation et processus de développement

En ce qui concerne le formalisme et l'enchaînement des étapes d'analyse et de conception que nous allons adopter, nous nous sommes basé sur le langage de modélisation UML et une méthode simple et générique qui se situe à mi-chemin entre UP (Unified Process), qui constitue un cadre général très complet de processus de développement, et XP (eXtreme Programming) qui est une approche minimaliste à la mode centrée sur le code. Cette méthode est issue de celle présentée par Pascal Roques dans son livre « UML - Modéliser un site e-commerce. Les cahiers du programmeur. Eyrolles, 2002 », qui résulte de plusieurs années d'expérience sur de nombreux projets dans des domaines variés. Elle a donc montré son efficacité dans la pratique et est :

- conduite par les cas d'utilisation, comme UP, mais bien plus simple ;
- relativement légère et restreinte, comme XP, mais sans négliger les activités de modélisation en analyse et conception ;
- fondée sur l'utilisation d'un sous-ensemble nécessaire et suffisant du langage UML (modéliser 80 % des problèmes en utilisant 20 % d'UML).

Dans tous les cas, il faut garder à l'esprit qu'une méthode n'est pas une formule magique. Le fait de produire des diagrammes UML selon un ordre établi n'est en aucun cas une garantie de réussite. Une méthode ne sert qu'à canaliser et ordonner les étapes de la modélisation. La valeur n'est pas dans la méthode, mais dans les personnes qui la mettent en œuvre. La démarche est décrite et détaillée dans [2].

Note : Pour plus de détail sur UML voir « AnnexeA ».

3. L'analyse des besoins

Dans cette partie, nous effectuerons une analyse qui se base sur les résultats de l'étude réalisée au cours de notre stage au sein de « Danone Djurdjura ».

3.1. Identification des acteurs du système

Un acteur est un utilisateur (humain, dispositif matériel ou autre système) qui désire effectuer des tâches précises sur l'application.

Les différents acteurs qu'on va trouver dans nos diagrammes de cas d'utilisation sont :

L'administrateur : Acteur principal de notre application, c'est à lui que reviennent les tâches suivantes :

- Gestion des produits (CRUD)
- Gestion des comptes utilisateurs
- Gestions des emballages
- Gestion des lignes
- Gestion des masses blanches
- Gestion des ingrédients
- Visualisation des données saisies par les opérateurs.
- Calcul et affichage des statistiques (à la demande)

- Répondre aux messages des opérateurs.

L'opérateur Process : Utilisateur possédant un compte (login+ mot de passe), il est chargé de la saisie uniquement, dans la phase process, il n'a pas les mêmes droits d'accès que l'administrateur, il est chargé des tâches suivantes :

- Saisir les propriétés et quantités du lait à la réception
- Saisir les quantités de pertes en lait
- Saisir les propriétés et quantités du lait écrémé et de la crème fraîche après écrémage.
- Saisir les quantités et propriétés de masse blanche à produire et après production.

L'opérateur Ligne : Utilisateur possédant un compte (login+ mot de passe), il est chargé de la saisie uniquement, dans la phase conditionnement, il n'a pas les mêmes droits d'accès que l'administrateur, il est chargé des tâches suivantes :

- Effectuer des contrôles sur les arômes, fruits et emballage avant le lancement des productions.
- Saisir les pertes dans les différentes étapes de production (démarrage, changement, arrêt ligne...).
- Saisir la quantité d'arôme, fruit, opercule, décor et BPS avant et après production afin de pouvoir calculer les pertes avec la différence.
- Saisir les poids des pots à la sortie, afin d'avoir les pertes en dosage.
- Signaler les pannes à l'administrateur.

3.2. L'identification des messages

Un message est un moyen de communication entre acteurs. Il définit un événement, c'est-à-dire, une information envoyée à un acteur et provoquant en réponse le commencement d'actions associées à ce dernier.

Les acteurs de notre système, peuvent échanger des messages qu'ils soient entrants ou sortants. En effet, les messages entrants représentent les demandes qu'un acteur effectue tandis que les messages sortants représentent la réponse du système à une demande donnée.

Acteur	A	Messages acteur-système	B	Messages système-acteur
Opérateur Process	1	Demande d'authentification	2	Afficher formulaire d'authentification
	3	Demande de la page Lait	4	Afficher la page lait
	5	Demande de la page écrémage	6	Afficher la page écrémage
	7	Demande de la page masse blanche	8	Afficher a page masse blanche
	9	Demande de validation de la page Lait	10	Valider la page lait
	11	Demande de validation de la page écrémage	12	Valider la page écrémage
	13	Demande de validation de la page masse blanche	14	Valider la page masse blanche
	C		D	
Opérateur Ligne	15	Demande d'authentification	16	Afficher formulaire d'authentification
	17	Demande la page fiche démarrage ligne	18	Afficher la page démarrage ligne
	19	Demande de la page control arôme	20	Afficher la page control arôme
	21	Demande de la page control emballage	22	Afficher la page control emballage
	23	Demande de la page dosage	24	Afficher la page dosage
	25	Demande de la page pertes BPS	26	Afficher la page pertes BPS
	27	Demande de la page pertes arôme	28	Afficher la page pertes arôme
	29	Demande de la page pertes fruit	30	Afficher la page pertes fruit
	31	Demande de validation de la page fiche démarrage ligne	32	Valider la page fiche démarrage ligne
	33	Demande de validation de la page control arôme	34	Valider la page control arôme
	35	Demande de validation de la page control emballage	36	Valider la page control emballage
	37	Demande de validation de la page dosage	38	Valider la page dosage
39	Demande de validation de la page pertes BPS	40	Valider la page pertes BPS	

	41	Demande de validation de la page pertes arôme	42	Valider la page pertes arôme
	43	Demande de validation de la page pertes fruit	44	Valider la page pertes fruit
	E		F	
Administrateur	45	Demande d'authentification	46	Vérification des informations
	47	Demande de gestion des produits	48	Gestion des produits
	49	Demande de gestion des lignes	50	Gestion des lignes
	51	Demande de gestion des emballages	52	Gestion des emballages
	53	Demande de gestion des ingrédients	54	Gestion des ingrédients
	55	Demande de gestion des utilisateurs	56	Gestion des utilisateurs
	57	Demande de suivi lait	58	Suivi lait
	59	Demande de suivi masse blanche	60	Suivi masse blanche
	61	Demande de suivi fruit	62	Suivi fruit
	63	Demande de suivi arôme	64	Suivi arôme
	65	Demande de suivi emballage	66	Suivi emballage
	67	Demande de suivi produit fini	68	Suivi produit fini

Tableau 1: Messages échangés entre les acteurs et le système.

3.3. Diagramme de contexte dynamique

C'est un diagramme représentant les acteurs externes qui interagissent avec le système. Il est considéré comme la plus haute vue de ce système.

Tous les messages (système <-> acteurs) identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme, que nous pouvons qualifier de diagramme de contexte dynamique (figure 2).

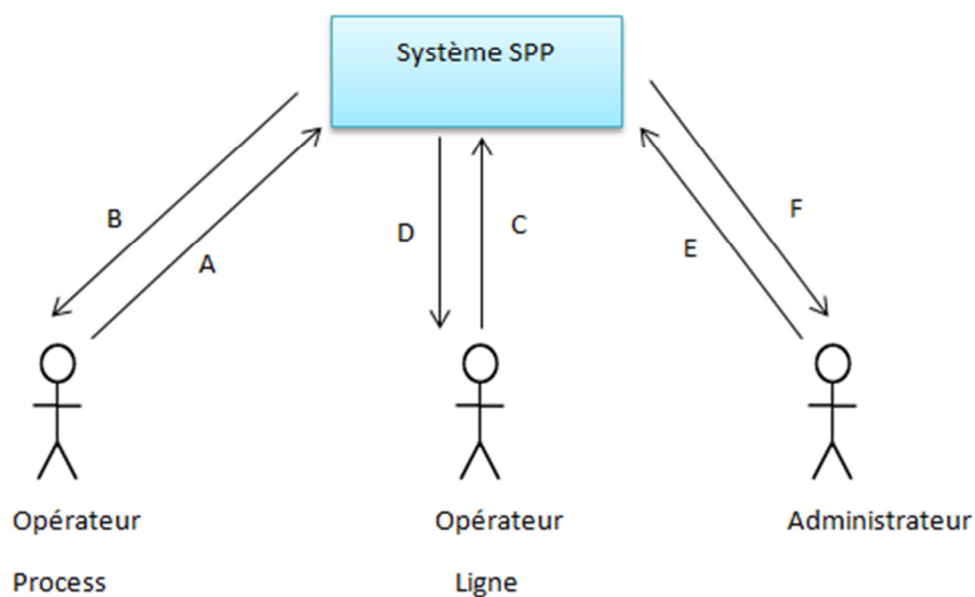


Figure 2: Diagramme de contexte dynamique

3.4. Identification des cas d'utilisation

Le diagramme des cas d'utilisations identifie les fonctionnalités fournies par le système, les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers [5].

Un cas d'utilisation correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur [6].

3.4.1. Cas d'utilisation par acteur

Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur.

Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service [7].

Le tableau suivant décrit les différents cas d'utilisation associés à chaque acteur :

Acteur	No	Cas d'Utilisation	
Opérateur process	1	Authentification	
	2	Saisir propriétés lait	Saisir les propriétés lait
	3	Saisir les informations de l'écémage	Saisir les propriétés lait écémé et crème fraîche
	4	Saisir la quantité et propriétés de la masse	Saisir la quantité et propriété e la masse
Opérateur ligne	5	Authentification	
	6	Renseigner fiche démarrage ligne	
	7	Control arôme	Contrôler les pompes
	8	Control emballage	Contrôler les bobines
	9	Renseigner les dosages	Saisir les poids des pots
	10	Saisir les pertes en emballage	Saisir les pertes en emballage
	11	Saisir les pertes en arôme	Saisir les pertes en arôme
	12	Saisir les pertes en fruit	Saisir les pertes en fruit
Administrateur	13	Authentification	
	14	Gestion utilisateurs	Ajouter utilisateur
			Modifier utilisateur
			Supprimer utilisateur
	15	Gestion lignes	Ajouter ligne
			Modifier ligne
			Supprimer ligne
	16	Gestion produits	Ajouter produit
			Modifier produit
			Supprimer produit
17	Gestion emballages	Ajouter emballage	

	18	Gestion masses blanches	Modifier emballage
			Supprimer emballage
			Ajouter masses blanches
	19	Gestion ingrédients	Modifier masses blanches
			Supprimer masses blanches
			Ajouter ingrédients
	20	Suivi lait	Modifier ingrédients
			Supprimer ingrédients
			Afficher l'évolution de la réception et utilisation du lait
	21	Suivi masse blanche	Suivi de production et consommation de la masse blanche
	22	Suivi ingrédients	Suivi de consommation et pertes ingrédients
	23	Suivi emballage	Suivi de consommation et pertes emballage
	24	Suivi produits finis	Suivi de pertes en produits finis

Tableau 2: Identification des cas d'utilisation

3.4.2. Description textuelle des cas d'utilisation

Elle est organisée de la façon suivante :

- **Un sommaire d'identification:** va résumer les propriétés du cas d'utilisation.
- **Une description des enchaînements:** pré conditions au déclenchement du cas d'utilisation doivent être spécifiées, un scénario nominal décrivant celui-ci additionné à des scénarios alternatifs et d'exceptions [8].
- **Cas d'utilisation <<S'authentifier>> pour tous les utilisateurs**

Sommaire	
Titre	S'authentifier
Résumé	Connecter l'utilisateur à son compte.
Acteurs	Administrateur, Opérateur ligne et Opérateur Process
Description des scénarios	
Pré condition	Accéder au pc serveur ou à l'un des pc clients.

Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur ou l'administrateur introduit son login et son mot de passe. 2. L'utilisateur ou l'administrateur envoie les informations 3. Le système vérifie l'authenticité des informations. 4. Le système donne l'accès à l'interface correspondante.
Alternative	Les informations saisies sont erronées ou champs vide. Le système affiche un message d'erreur (login ou mot de passe erronés ou champs vide).
Arrêt	Utilisateur authentifié.
Post conditions	L'utilisateur accède à sa page d'accueil.

Tableau 3: Description de cas d'utilisation "s'authentifier"



Figure 3: Cas d'utilisation "s'authentifier"

- Cas d'utilisation <<Saisir propriétés lait>> pour l'opérateur process

Sommaire	
Titre	Saisir propriétés lait
Résumé	Saisir les quantités et les informations concernant le lait.
Acteurs	Opérateur Process
Description des scénarios	
Pré condition	Etre connecté et être opérateur Process.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur process remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez que tous les champs sont remplis).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 4: Description de cas d'utilisation "Saisir les propriétés lait"

- Cas d'utilisation << Renseigner les dosages >> pour l'opérateur ligne

Sommaire	
Titre	Saisir les dosages.
Résumé	Saisir les dosages des pots produits (importé directement depuis les balances).
Acteurs	Opérateur Ligne
Description des scénarios	
Pré condition	Etre connecté et être opérateur Ligne.
Scénario nominal	<ol style="list-style-type: none"> 1. Tous les poids doivent être importés depuis les balances vers les cases adéquates. 2. L'opérateur Ligne remplit tous les champs (valider). 3. L'opérateur envoie les informations saisies. 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 5: Description de cas d'utilisation "Renseigner dosage"

- Cas d'utilisation << Gestion utilisateurs >> pour l'administrateur

Sommaire	
Titre	Gestions utilisateurs
Résumé	Ajouter, modifier ou supprimer un utilisateur.
Acteurs	Administrateur
Description des scénarios	
Pré condition	Etre connecté et être Administrateur.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande le formulaire d'ajout, modification ou suppression d'un utilisateur. 2. Le système affiche le formulaire demandé. 3. L'administrateur saisit les données et valide. 4. Le système vérifie les données. 5. Le système donne l'accès à l'interface correspondante.
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies ou des champs vides).

Arrêt	Lorsqu'il valide ou annule.
Post conditions	Informations enregistrées et interface adéquate affichée.

Tableau 6: Description de cas d'utilisation "Gestion des utilisateurs"

- Cas d'utilisation << suivi produits finis >> pour l'administrateur

Sommaire	
Titre	Suivi produits finis.
Résumé	Afficher les suivis produits finis par période ou autres critères selon la volonté de l'administrateur.
Acteurs	Administrateur
Description des scénarios	
Pré condition	Etre connecté et être administrateur.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande à afficher le suivi produit fini selon les critères choisis. 2. Le système affiche le suivi produit fini selon les critères choisis par l'administrateur (suivi mensuel, annuel, à temps réel des pertes)
Alternative	/
Arrêt	Lorsqu'il passe à un autre suivi.
Post conditions	Affichage des statistiques dans l'interface adéquate.

Tableau 7: Description de cas d'utilisation "Suivi produits finis"

Note : Pour le reste des descriptions des cas d'utilisation, voir « Annexe B »

3.4.3. Diagrammes de cas d'utilisation

a. Diagramme de cas d'utilisation pour l'acteur « Opérateur process »

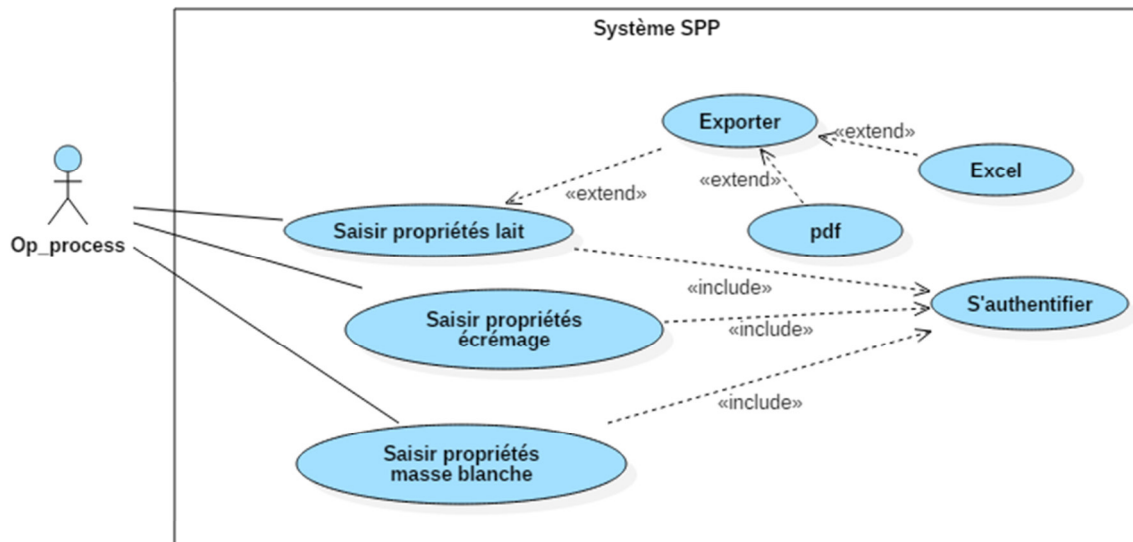


Figure 4: Diagramme de cas d'utilisation pour l'acteur « Opérateur process »

b. Diagramme de cas d'utilisation pour l'acteur « Opérateur ligne »

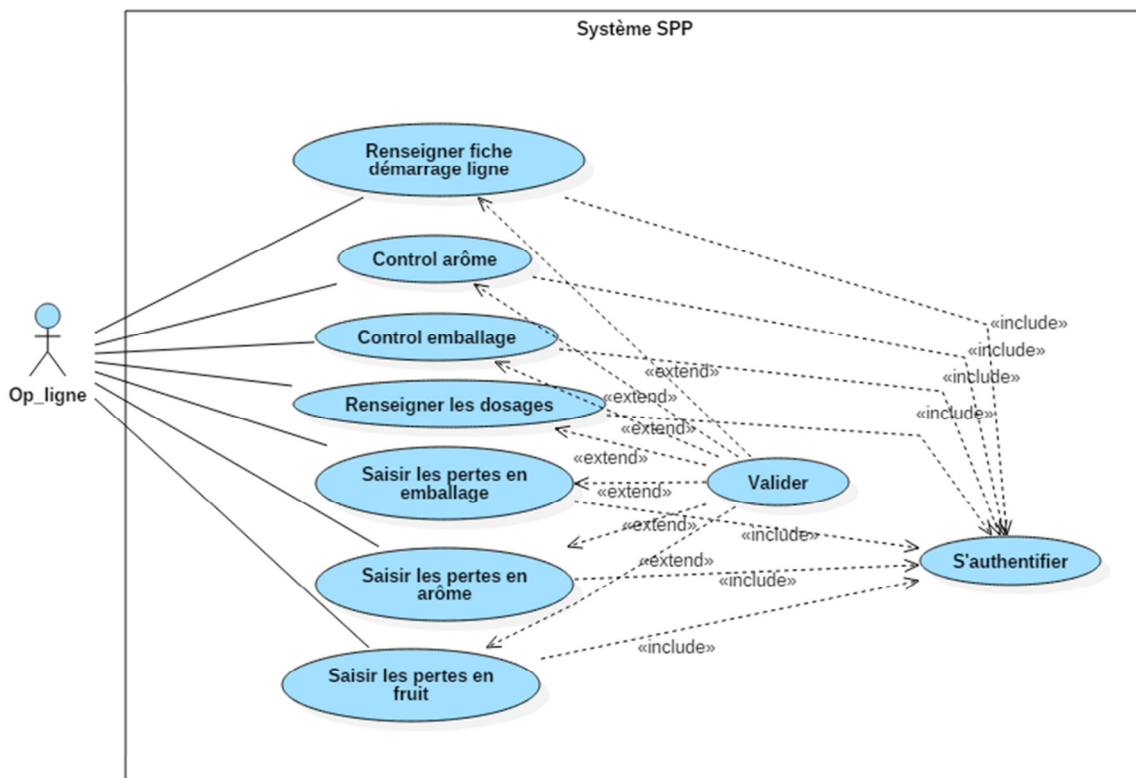


Figure 5: Diagramme de cas d'utilisation pour l'acteur « Opérateur ligne »

c. Diagramme de cas d'utilisation pour l'acteur « Administrateur »

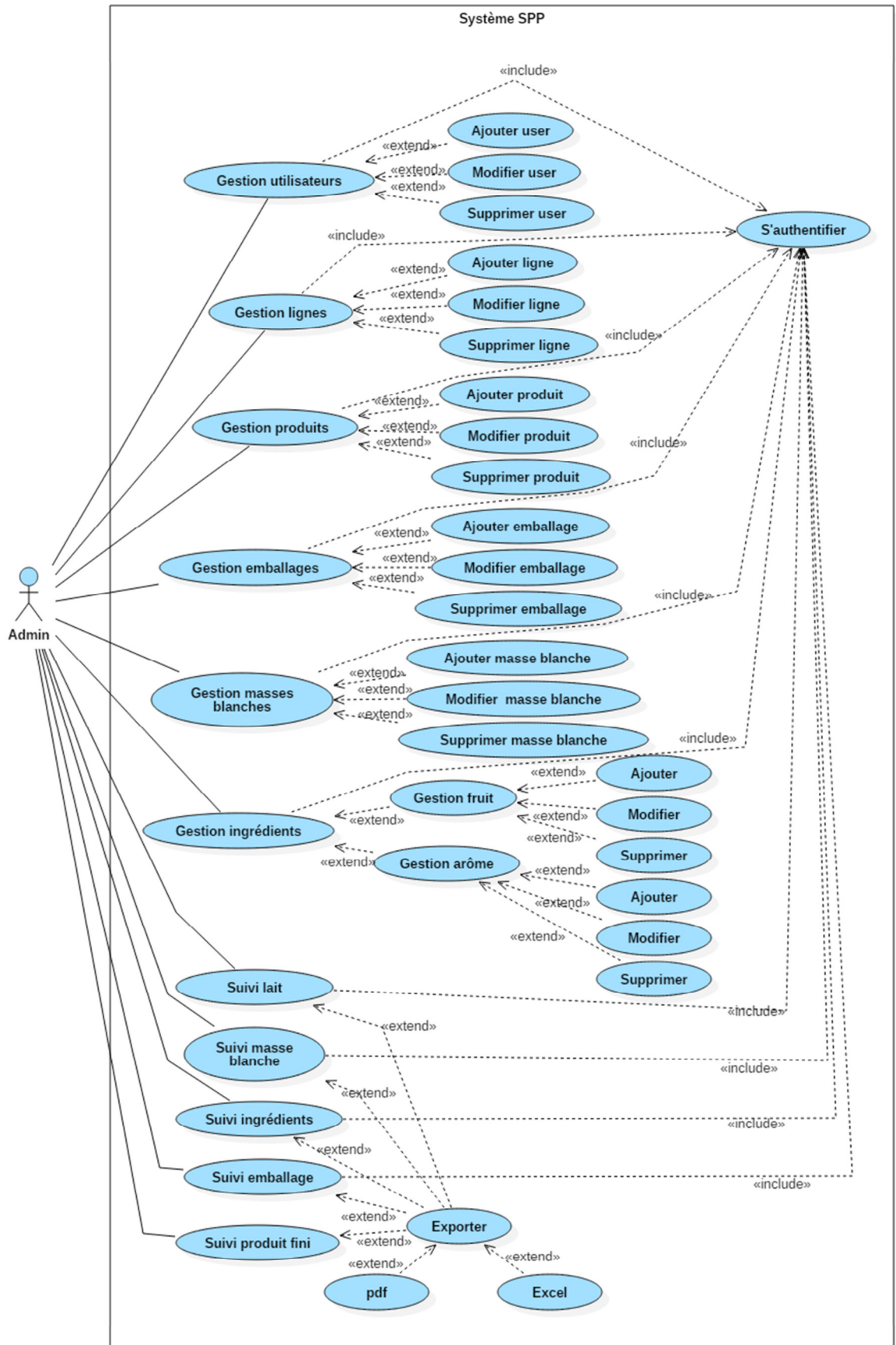


Figure 6: Diagramme de cas d'utilisation pour l'acteur « Administrateur »

d. Diagramme de cas d'utilisation global

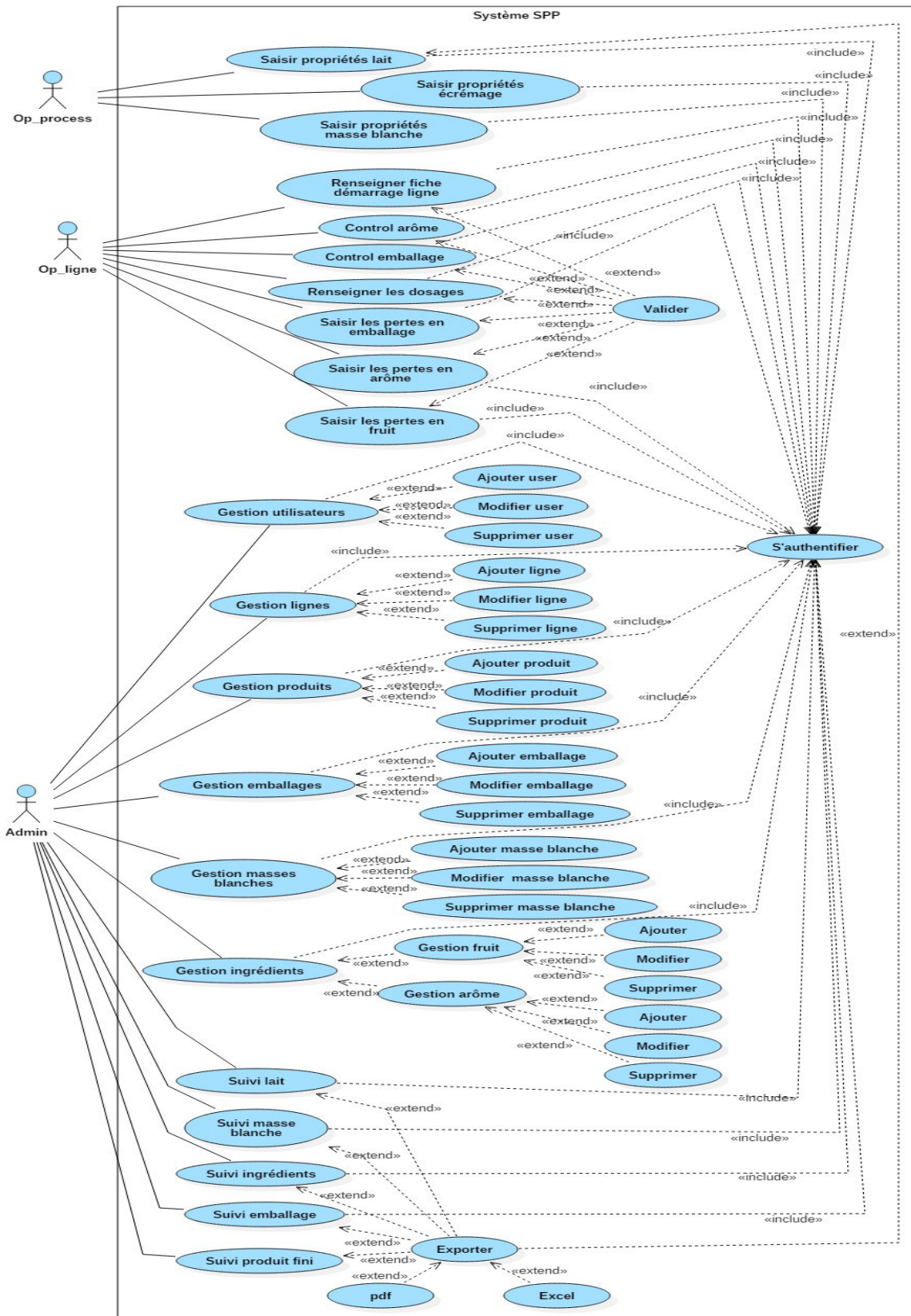


Figure 7: Diagramme de cas d'utilisation global

3.5. Développement du modèle dynamique

Le développement du modèle dynamique va nous permettre d'illustrer l'utilisation des concepts dynamiques d'UML et des diagrammes associés en phase d'analyse. Dans cette étape, nous verrons comment décrire des scénarios mettant en jeu un ensemble d'objets échangeant des messages. Pour ce là nous avons choisi le diagramme de séquence qui met l'accent sur la chronologie des messages.

Les diagrammes de séquences mettent en valeur les échanges de messages (déclenchant des événements) entre acteurs et objets (ou entre objets et objets) de manière chronologique, l'évolution du temps se lisant de haut en bas [9].

Un scénario décrit une exécution particulière d'un cas d'utilisation du début à la fin. Il correspond à une sélection d'enchaînement d'un cas d'utilisation [8].

Les diagrammes de séquences suivants, montrent une vue dynamique sur l'interaction de l'utilisateur, avec le système dans chacun des scénarios possibles, à travers notre travail.

3.5.1. Diagrammes de séquence par scénarios

- **Diagramme de séquence du cas d'utilisation « Authentification »**

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification des informations associées à un utilisateur (généralement un login et un mot de passe). Ces informations sont préétablies dans une base de données. Lors d'une authentification deux cas se présentent : les informations introduites par l'utilisateur sont incomplètes, dans ce cas un message d'erreur s'affiche, ou les informations saisies sont complètes et le système procède à leur vérification. Ceci explique l'utilisation de l'opérateur « alt ». Le même opérateur illustre les deux réactions du système, après la vérification des informations saisies par l'utilisateur, soit par l'affichage d'un message d'erreur, ou de l'interface correspondante.

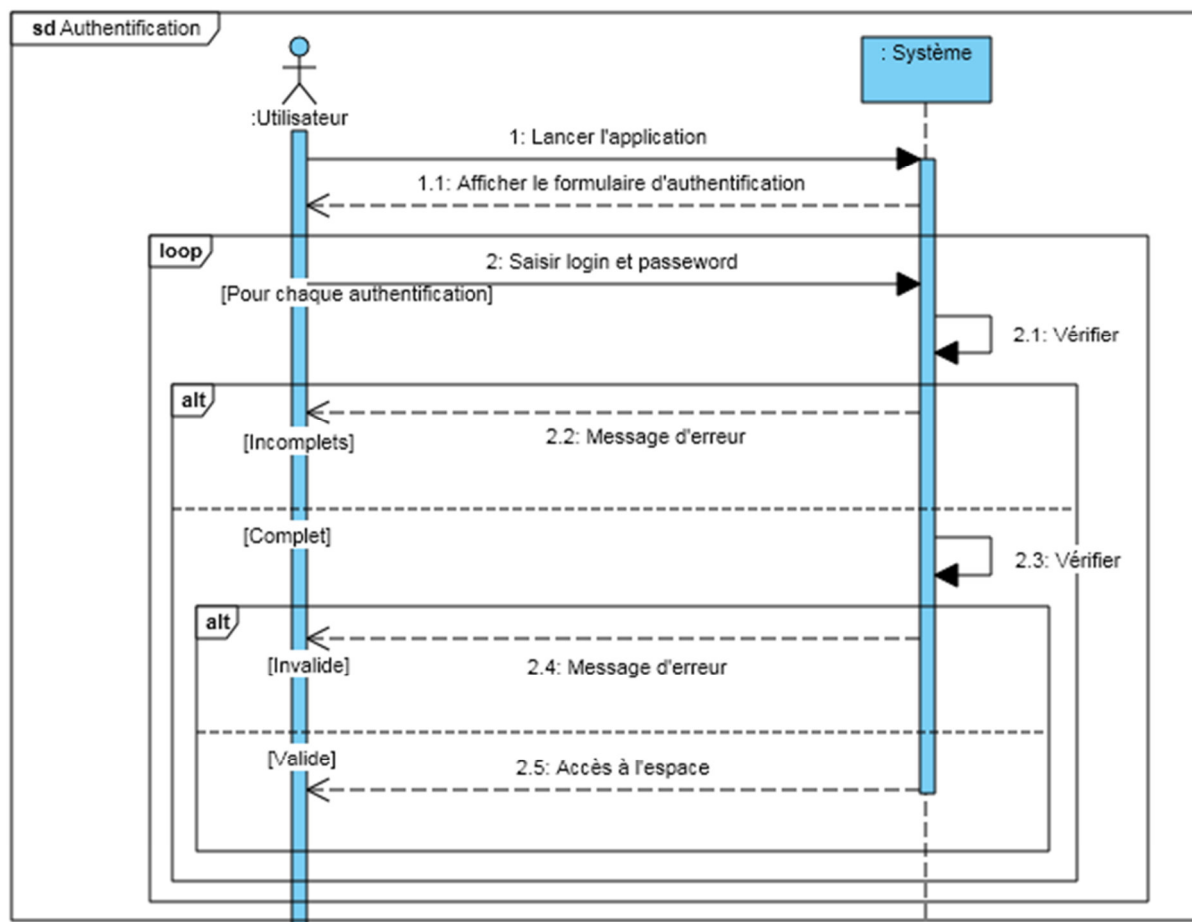


Figure 8: Diagramme de séquence "Authentication"

- Diagramme de séquence du cas d'utilisation « Saisir propriétés lait »

Lorsque l'opérateur ligne envoi une demande de saisie des propriétés lait, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

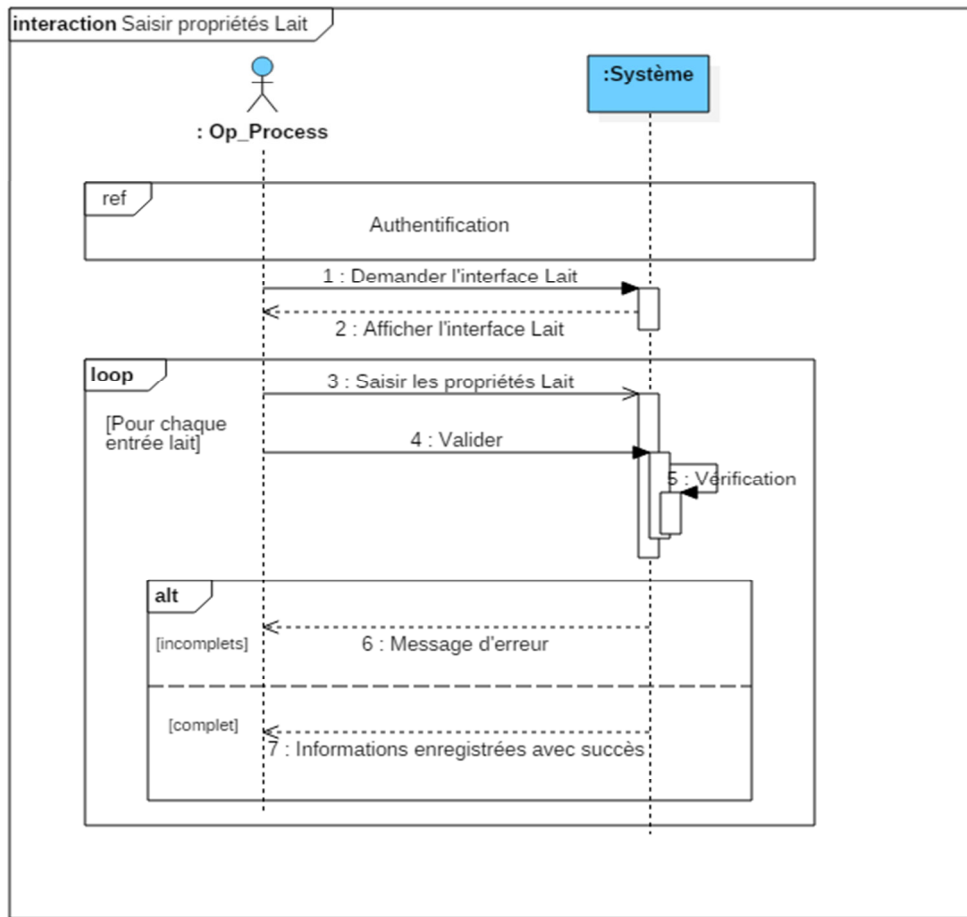


Figure 9: Diagramme de séquence du cas d'utilisation « Saisir propriétés lait »

- **Diagramme de séquence du cas d'utilisation « Renseigner dosage »**

Lorsque l'opérateur ligne envoi une demande de renseigner les dosages, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

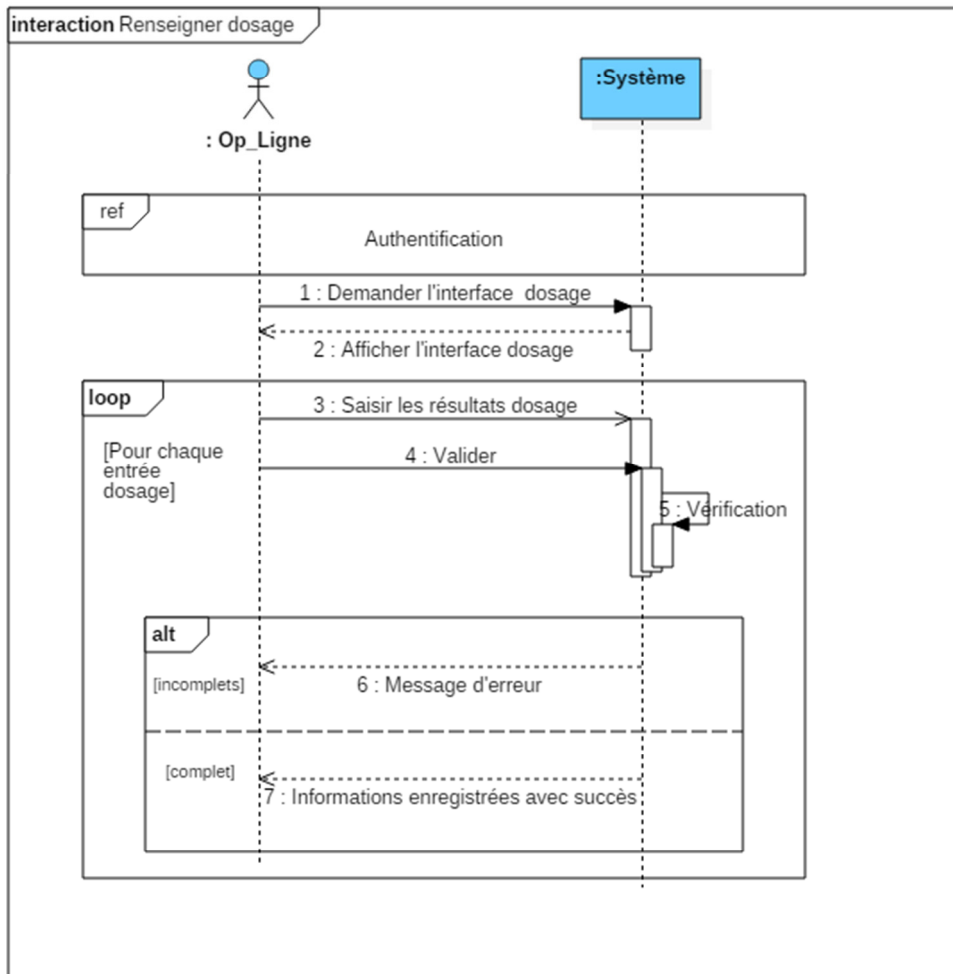


Figure 10: Diagramme de séquence du cas d'utilisation « Renseigner dosage »

- **Diagramme de séquence du cas d'utilisation « Gestion des utilisateurs »**

Le diagramme ci-après décrit le diagramme de séquence général comportant les trois scénarios du cas d'utilisation « ajouter », « modifier » ou « supprimer » un utilisateur.

Chaque scénario est représenté séparément par un diagramme de séquence.

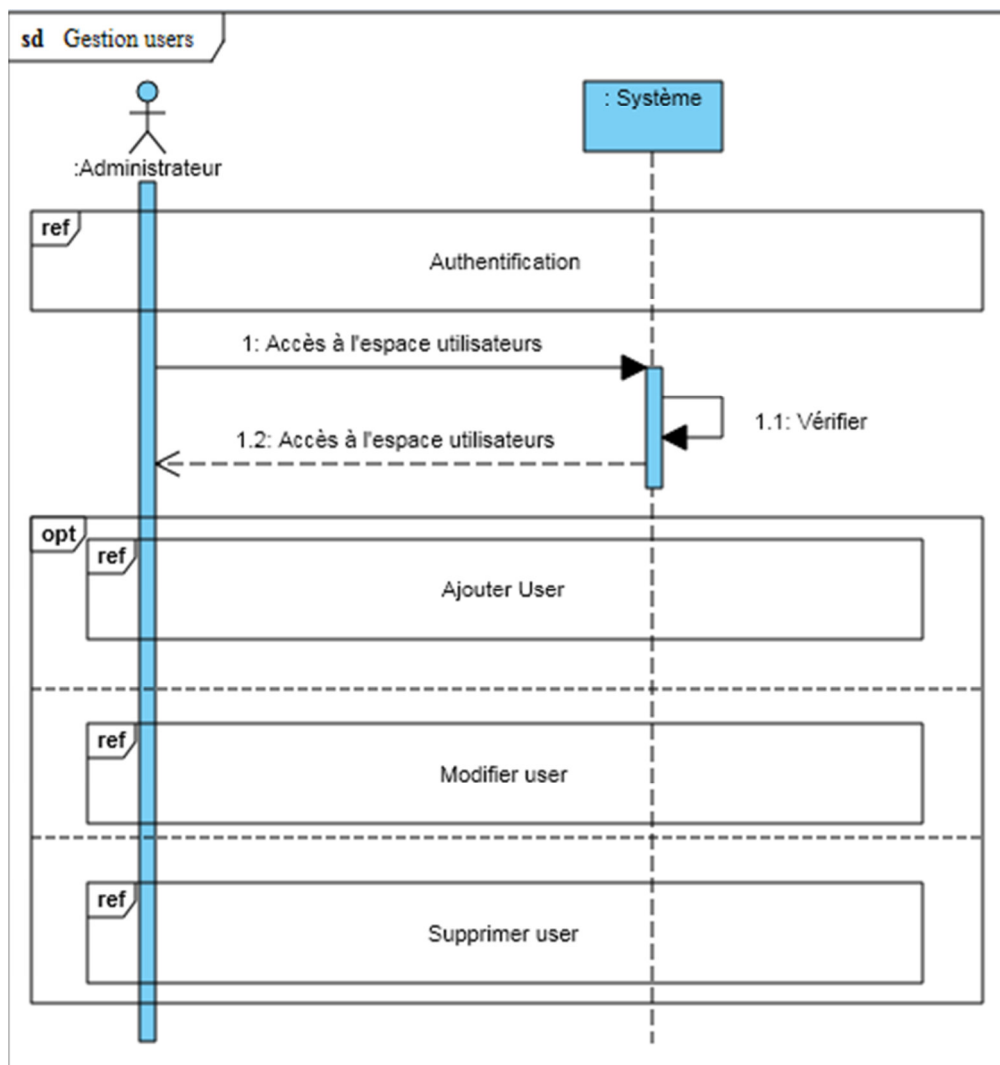


Figure 11: Diagramme séquence du cas d'utilisation « Gestion des utilisateurs »

Note : c'est les mêmes étapes pour les diagrammes de séquence de la gestion des lignes, produits finis, masses blanches, ingrédients et emballages.

- **Diagramme de séquence du cas d'utilisation « Suivi produit fini »**

Après l'envoi de la demande du suivi produit fini par l'administrateur, le système va afficher le suivi produit fini avec les actions possibles à effectuer (suivi à temps réel, par date...).

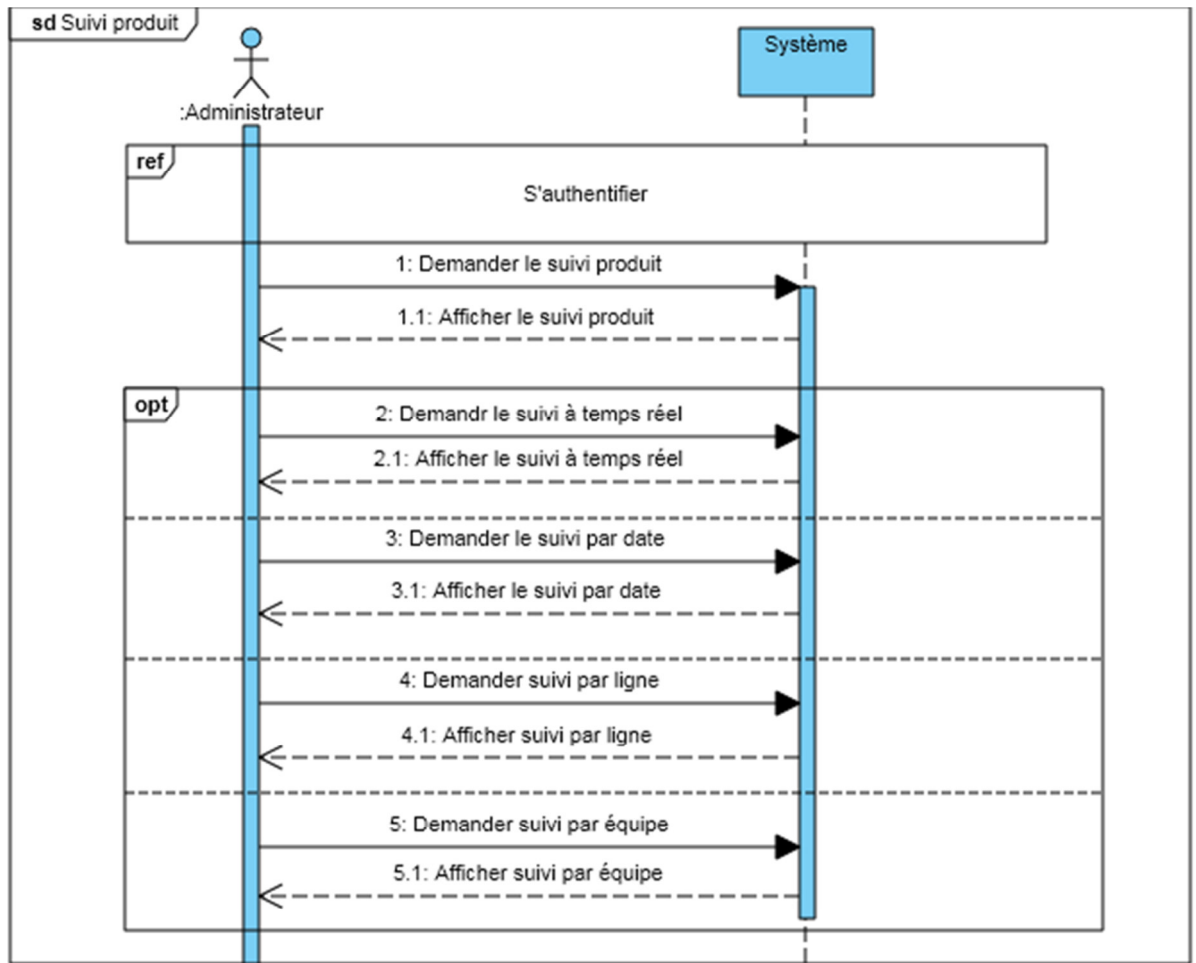


Figure 12: Diagramme séquence du cas d'utilisation « Suivi produit »

Note : c'est les mêmes étapes pour les diagrammes de séquence du suivi ingrédients et suivi emballage.

- **Diagramme de séquence du cas d'utilisation « Suivi pertes »**

Après l'envoi de la demande du suivi pertes par l'administrateur, le système va afficher le suivi pertes avec les actions possibles à effectuer (suivi à temps réel, par date, suivi lait, suivi emballage...).

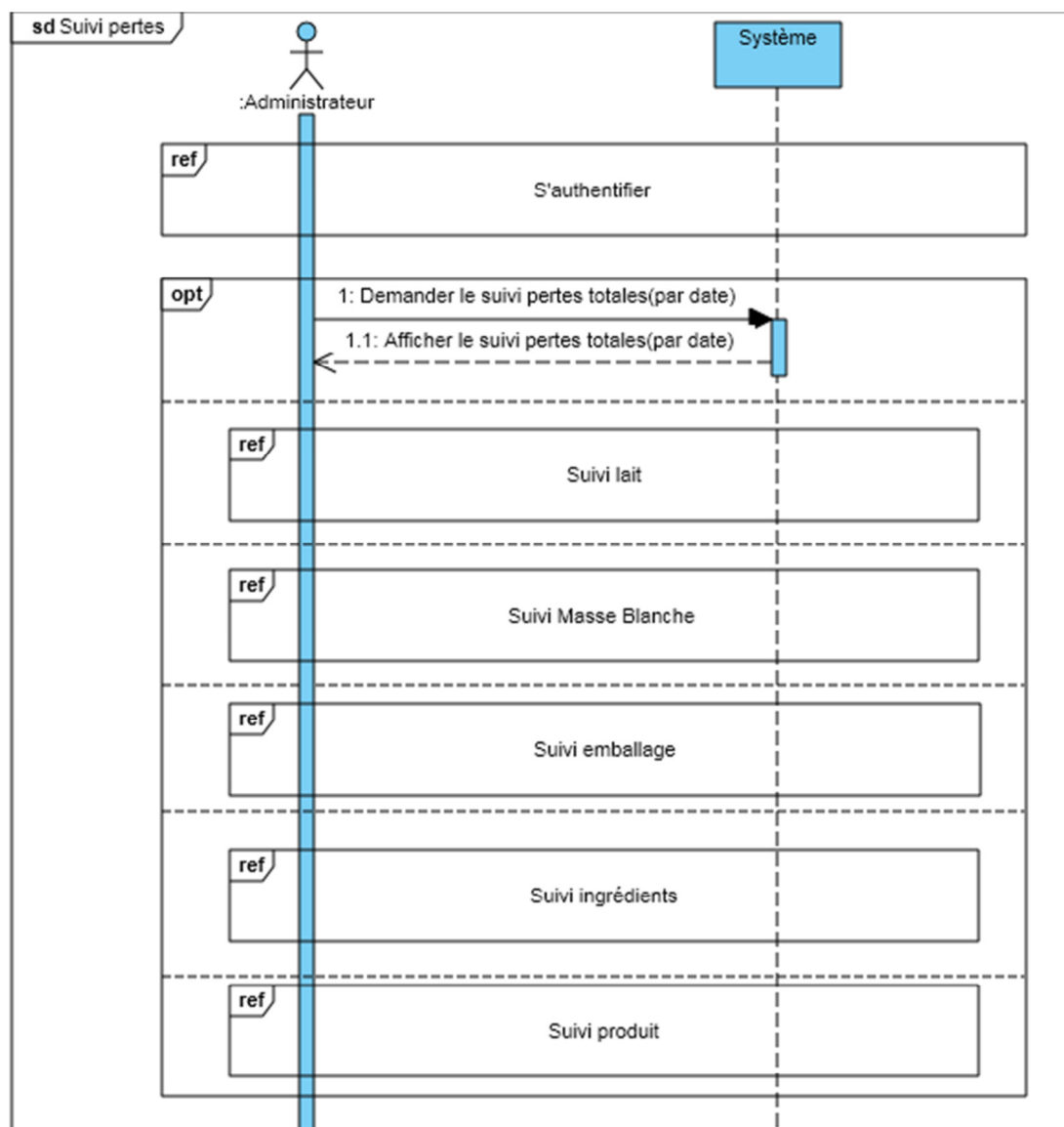


Figure 13: Diagramme séquence du cas d'utilisation « Suivi pertes »

Note : Pour le reste des diagrammes de séquences pour les autres cas d'utilisation, voir « Annexe B »

5. Conclusion

Dans ce chapitre, nous nous sommes intéressé à analyser les besoins de notre application, ce qui nous a permis d'avoir une vision prématurée sur les acteurs qui interagissent avec le système, les relations entre eux et leurs cas d'utilisations, grâce aux diagrammes de contexte, de cas d'utilisation et de séquence système.

Dans le chapitre suivant nous allons entamer la conception de notre système en présentant les diagrammes des interactions et le diagramme des classes.

Chapitre III

Conception et schéma
relationnel de données

Chapitre III

Conception et schéma relationnel de données

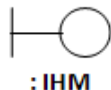
1. Introduction

Nous arrivons maintenant à la phase terminale de modélisation avec UML. Après la modélisation des besoins puis l'organisation de la structure de la solution, la conception consiste à construire et à documenter précisément les classes et les tables. Nous présentons les diagrammes de séquences détaillé correspondant à chaque cas d'utilisation, le diagramme de classe et passer à réaliser le modèle relationnel associé au diagramme de classe de notre application.

2. Diagrammes de séquence détaillés

Pour chaque diagramme de séquence système, nous établirons le diagramme de séquence d'interaction, dans lequel le système est remplacé par les objets qui interviennent pour réaliser le cas d'utilisation concerné. Pour ce type de diagramme, nous avons trois types de classes (voir [10] page 171) :

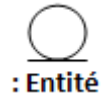
- Classes d'interface (boundary) : des classes qui permettent l'interaction entre l'application et ses utilisateurs. Pour chaque cas d'utilisation, il y a au moins une classe d'interface. Ce type de classe est schématisé comme suit :



- Classes de Contrôle (Control) : Ce sont des classes qui contiennent les traitements et la cinématique de l'application. Elles font la transition entre les classes d'interface et les classes entité. Elles sont schématisées comme suit :



- Classes entité (entity) : Elles représentent les objets métiers, et ce sont très souvent des entités persistantes, c'est-à-dire qui vont garder leurs informations (données) après l'exécution d'un cas d'utilisation particulier. En général, elles sont enregistrées dans une base de données. Leur schématisation se fait grâce à ce stéréotype :



Les figures sises ci-dessous, présenteront respectivement les diagrammes d'interactions s'authentifier, ajouter, modifier et supprimer un utilisateur.

- **Diagramme de séquence d'interaction "S'authentifier"**

Ce diagramme représente le diagramme de séquence détaillé de cas d'utilisation "S'authentifier".

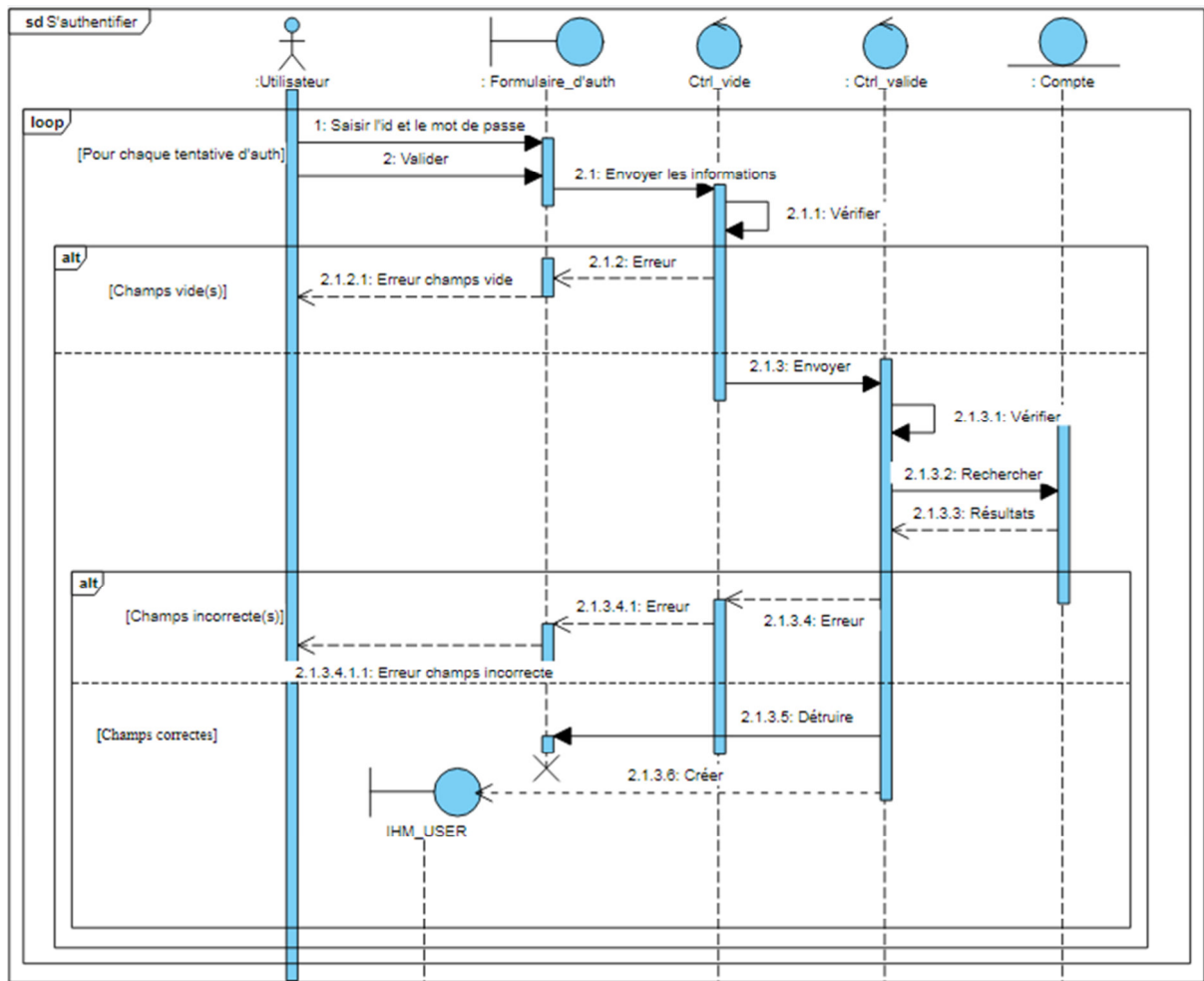
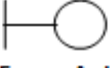




Figure 14: Diagramme séquence d'interaction « S'authentifier »

Le tableau suivant résume les stéréotypes utilisés pour représenter les objets qui interviennent pour réaliser le cas d'utilisation « S'authentifier » :

<i>Stéréotype</i>	<i>Signification</i>
 : Form_Auth	Objet interface pour le formulaire d'authentification. Ce dernier contient deux champs de saisie, à savoir, l'identifiant (login) et le mot de passe, qui permettent à l'utilisateur de renseigner son identifiant et son mot de passe. En plus, il contient un bouton pour envoyer ces informations.
 : Ctrl_vider	Objet contrôle qui permet de vérifier s'il y a un ou plusieurs champs vides. Nous pouvons aussi vérifier dans ce contrôleur la forme des champs, par exemple, le mot de passe doit avoir au moins 7 caractères avec au moins deux caractères numériques, et l'identifiant doit avoir au moins 5 caractères.
 : Ctrl_valide	Deuxième objet contrôle qui permet de vérifier si les champs renseignés par l'utilisateur sont correctes. Cette vérification s'effectue en utilisant l'objet entité de la classe « Compte » (objet



	persistant qui enregistre les comptes des utilisateurs du système).
 : Compte	Objet entité (objet métier) qui contient tous les comptes des utilisateurs du système.
 : IHM_USER	Objet interface qui représente la vue principale de l'utilisateur connecté. Nous pouvons la considérer comme l'espace de travail de l'utilisateur. Tous les autres cas d'utilisation seront accessibles à partir de cette interface.

Tableau 8: Les stéréotypes du Diagramme d'interaction « S'authentifier »

Remarques

- Les diagrammes de séquence d'interaction nous guident dans la phase de codage de l'application, et ceci à travers la séparation entre les parties du code responsable de l'interface, des contrôles, des entités ainsi que l'accès aux données.
- Les différentes classes entités utilisées dans les diagrammes de séquences d'interaction permettent d'établir le diagramme de classe du domaine (classes des objets métiers).

- **Diagramme de séquence d'interaction "Saisir propriétés lait"**

Ce diagramme représente le diagramme de séquence détaillé de cas d'utilisation "Saisir propriétés lait".

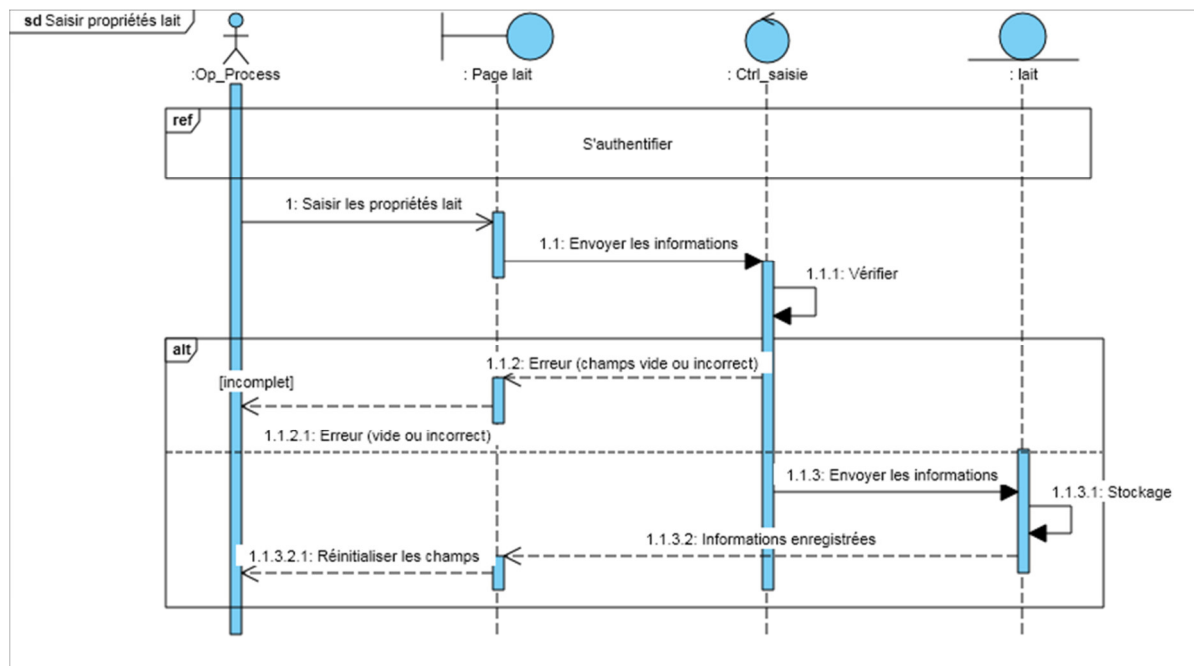


Figure 15: Diagramme séquence d'interaction « Saisir propriétés lait »

Note : c'est les mêmes étapes pour les diagrammes d'interactions des cas : Saisir informations écrémage, Saisir propriétés masse blanche, Renseigner fiche démarrage ligne, Control arôme, Control emballage, Renseigner les dosages, Saisir les pertes en emballage, Saisir les pertes en arôme, Saisir les pertes en fruit.

- **Diagramme de séquence d'interaction "Ajouter un utilisateur"**

Ce diagramme représente le diagramme de séquence détaillé de cas d'utilisation "ajouter un utilisateur".

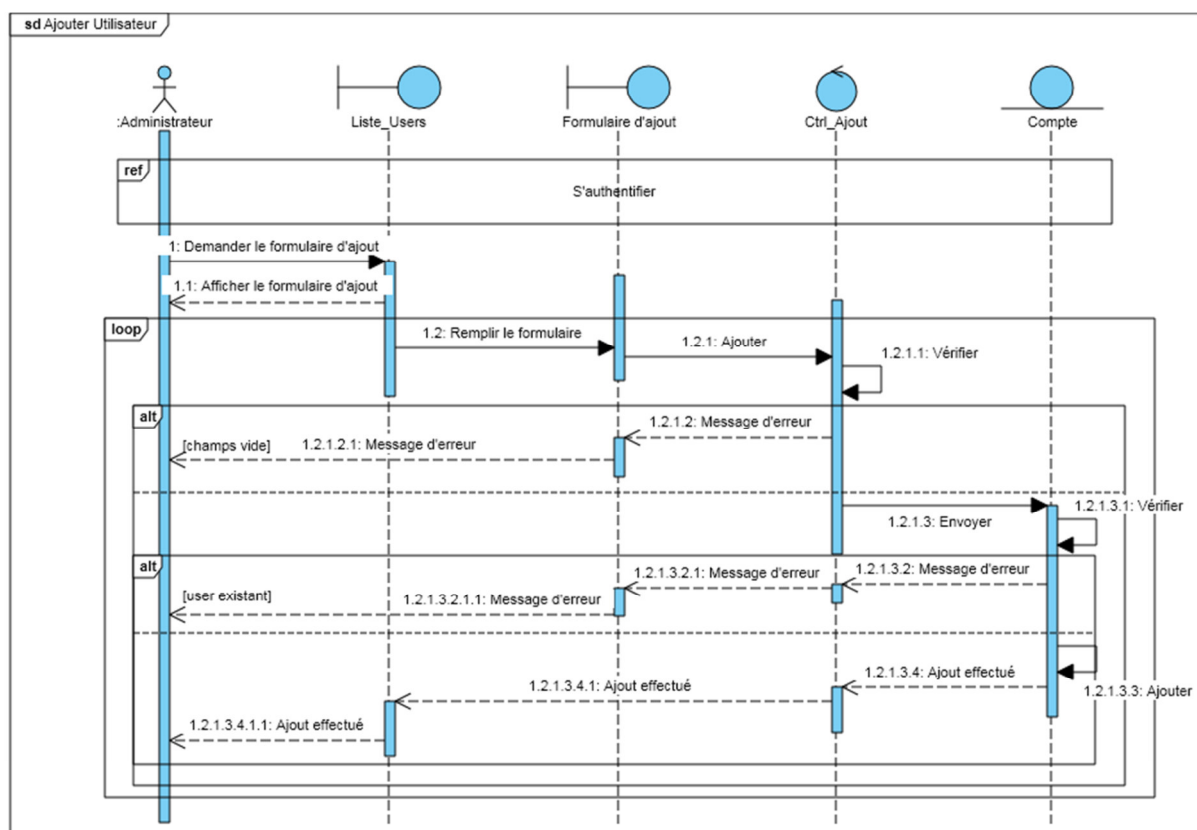


Figure 16: Diagramme séquence d'interaction « Ajouter utilisateur »

- **Diagramme de séquence d'interaction "Modifier un utilisateur"**

Ce diagramme représente le diagramme de séquence détaillé "modifier un utilisateur".

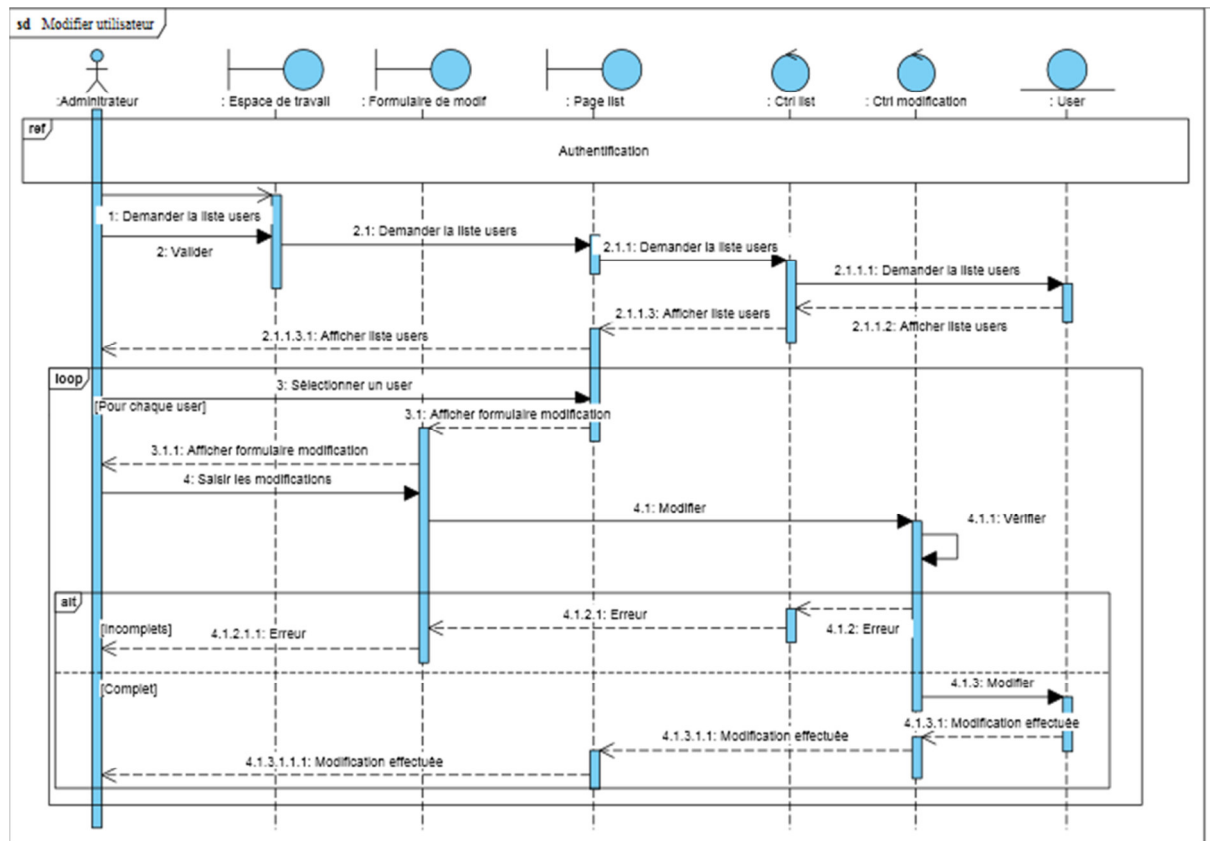


Figure 17: Diagramme séquence d'interaction « Modifier utilisateur »

- **Diagramme de séquence d'interaction "Supprimer un utilisateur"**

Ce diagramme représente le diagramme de séquence détaillé "supprimer un utilisateur".

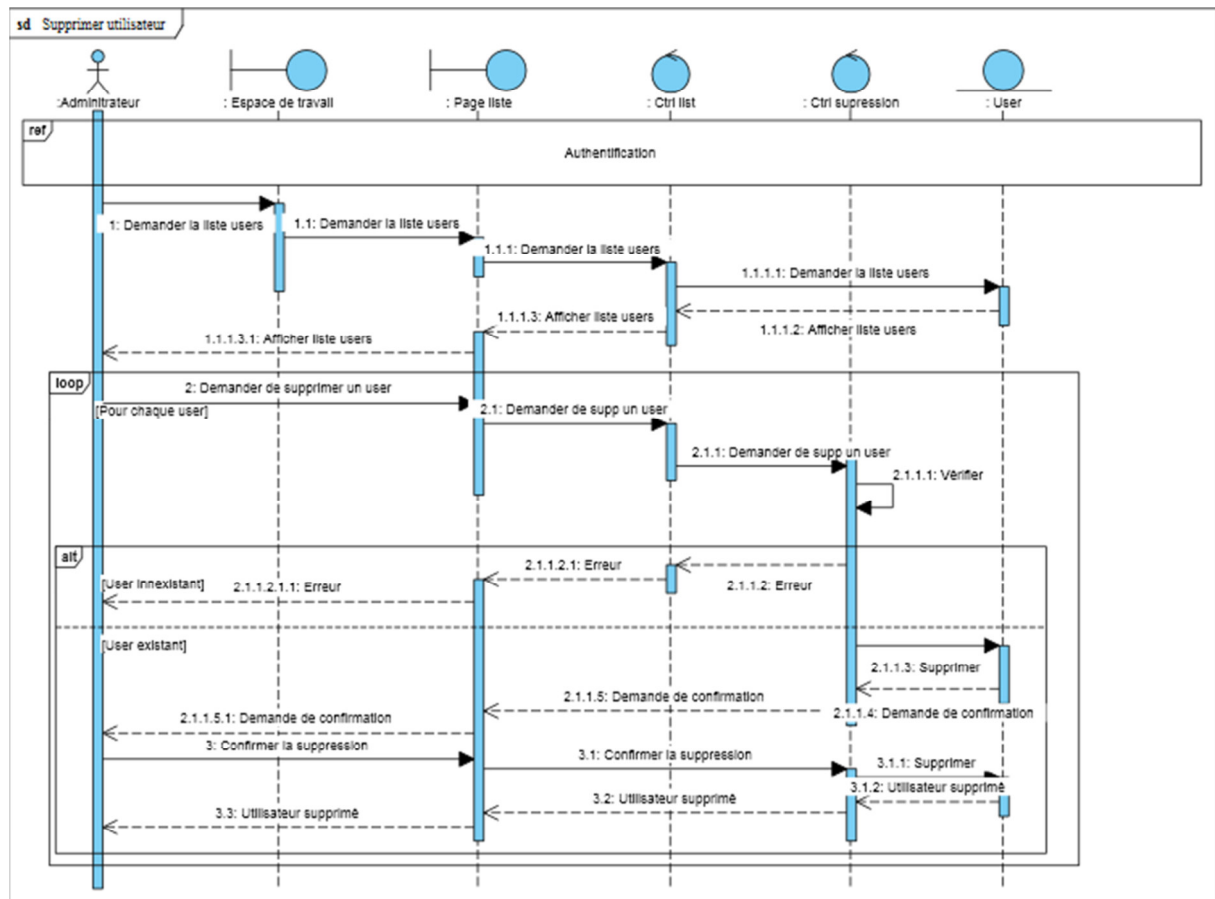


Figure 18: Diagramme séquence d'interaction « Supprimer utilisateur »

- **Diagramme de séquence d'interaction "Gestion des utilisateurs"**

Ce diagramme représente le diagramme de séquence détaillé "Gestion des utilisateurs".

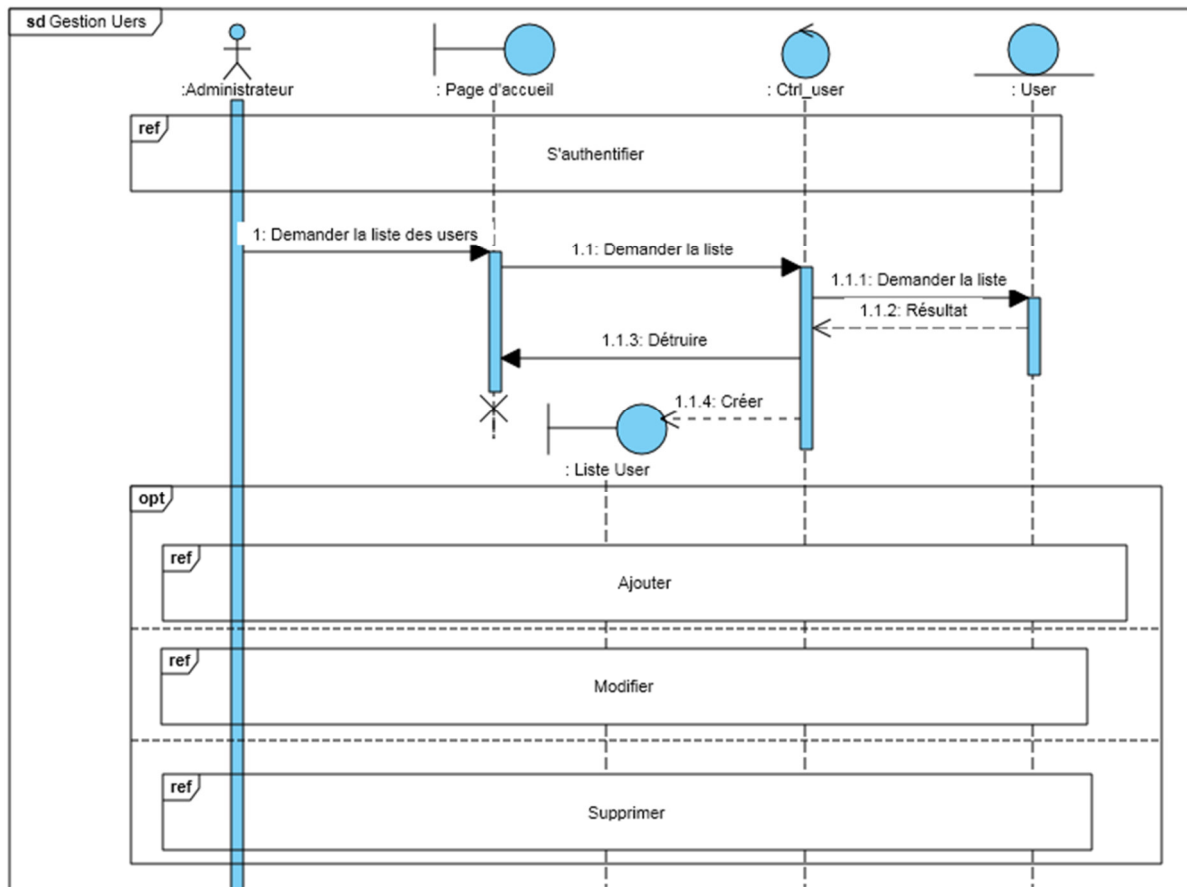


Figure 19: Diagramme séquence d'interaction «Gestion des utilisateurs »

Note : c'est les mêmes étapes pour les diagrammes d'interactions de la gestion des lignes, produits finis, masses blanches, ingrédients et emballages.

3. Diagramme de Classes du domaine

Le diagramme de classe est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens (héritage, agrégation, composition, ...etc.)

La figure ... illustre le diagramme de classe de notre application. Nous nous focalisons sur les classes entité (classes du domaine) qui représente les données de l'application

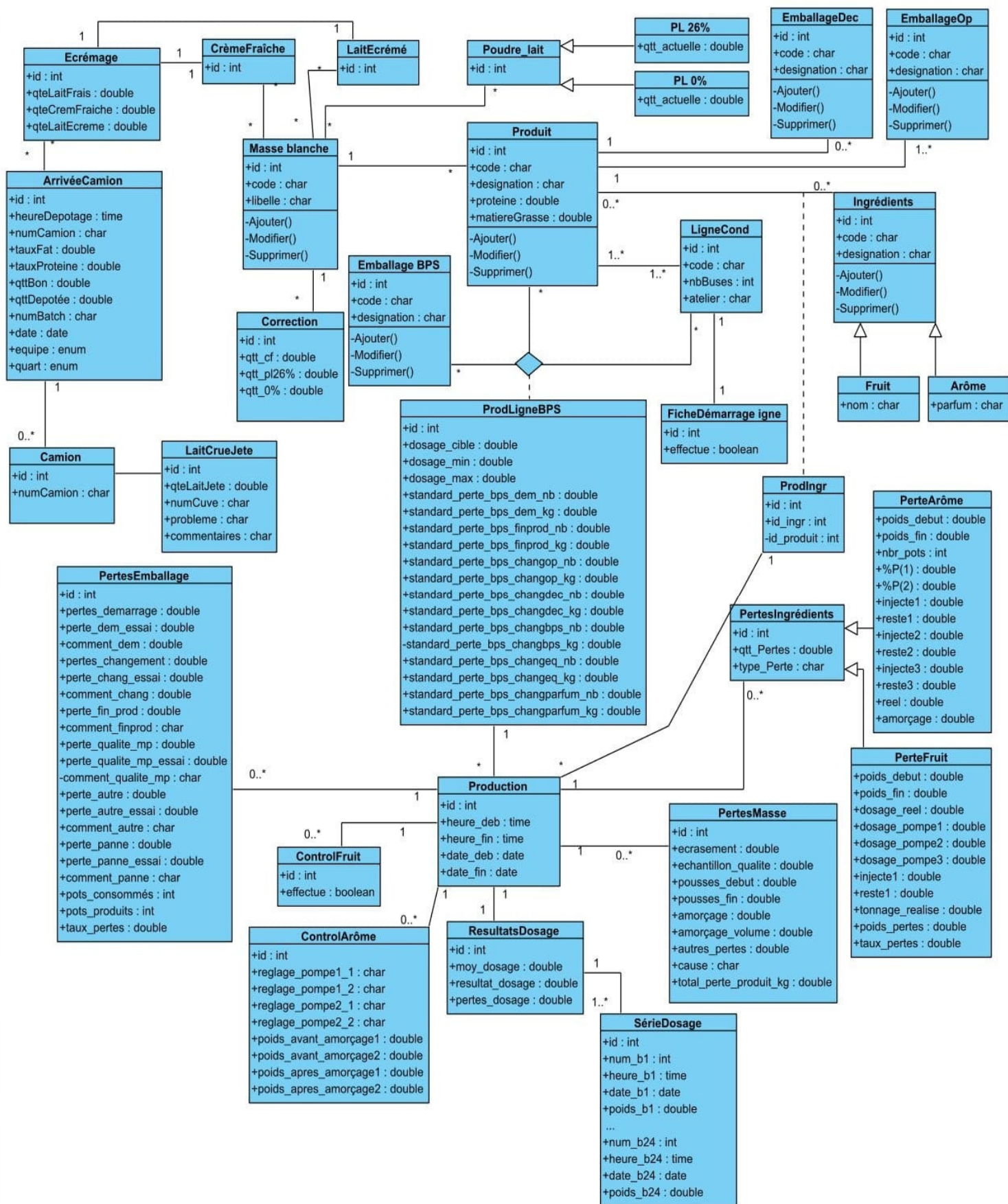


Figure 20: Diagramme de classes

3.1. Classes, attributs et responsabilités

Dans ce qui suit, nous allons décrire les différentes classes schématisées dans la figure Cette description sera présentée sous forme d'un tableau, comme suit :

Classe	Responsabilité	Attributs		
		Désignation	Définition	Type
ArriveeCamion	Classe qui enregistre les informations du lait.	<u>id</u>	Identifiant	Numérique
		heureDepotage	Heure arrivée	time
		numCamion	N° Camion	Alphanumérique
		tauxFat	Taux de FAT	Numérique
		tauxProteine	Taux protéine	Numérique
		qttBon	Quantité Bon	Numérique
		qttDepotée	Quantité Dépotée	Numérique
		numBatch	N° Batch	Alphanumérique
		date	date	Date
		equipe	équipe	Alphanumérique
Ecrémage	Enregistre les informations sur l'écémage.	<u>id</u>	Identifiant	Numérique
		quart	quart	Alphabétique
Crème fraîche	Stocker les informations de crème fraîche.	<u>id</u>	Identifiant	Numérique
Poudre de lait	Stocker les informations de la PL.	<u>id</u>	Identifiant	Numérique
PL 26%	Stocker les informations de la PL26%.	<u>id</u>	Identifiant	Numérique
PL 0%	Stocker les informations de la PL0%.	<u>id</u>	Identifiant	
Lait écrémé	Stocker les informations du lait écrémé.	<u>id</u>	Identifiant	Numérique.
Masse blanche	Stocker les informations de la masse blanche.	<u>id</u>	Identifiant.	Numérique.
		code	Code de la masse	Alphabétique.
		libelle	Libelle de la masse	Alphanumérique.
Correction	Informations de correction	<u>id</u>	Identifiant.	Numérique.
		qtt_cf	Quantité crème fraîche	Numérique
		qtt_pl26%	Quantité poudre de	Numérique

			lait 26%	
		qtt_pl0%	Quantité poudre de lait 0%	Numérique
Produit	Stocks les produits et leurs informations.	<u>id</u>	Identifiant	Numérique
		code	Code produit	Alphanumérique
		designation	Désignation	Alphanumérique
		proteine	Taux protéine	Numérique
		matiereGrasse	Taux matière grasse	Numérique
Emballage BPS	Stock les emballages BPS	<u>id</u>	Identifiant	Numérique
		code	Code BPS	Alphanumérique
		designation	Désignation	Alphanumérique
Emballage Dec	Stock les emballages Décor	<u>id</u>	Identifiant	Numérique
		code	Code Décor	Alphanumérique
		designation	Désignation	Alphanumérique
Emballage Op	Stock les emballages Opercule	<u>id</u>	Identifiant	Numérique
		code	Code Opercule	Alphanumérique
		designation	Désignation	Alphanumérique
Ligne_cond	Les lignes de conditionnement	<u>id</u>	Identifiant	Numérique
		code	Code	Alphanumérique
		nbBuses	Nombre buses	Numérique
		atelier	Atelier	Alphanumérique
Ingrédients	L'id et le code et désignation des ingrédients avec le type (fruit ou arôme)	<u>id</u>	Identifiant	Numérique
		code	Code	Alphanumérique
		designation	Désignation	Alphanumérique
Fruit	Les fruits	nom	Nom	Alphanumérique
Arôme	Les arômes	parfum	Parfum	Alphanumérique
Fiche démarrage	Enregistre si la fiche démarrage a été renseigné ou pas	<u>id</u>	Identifiant	Numérique
		effectue	effectué	Boolean
Prod_ligne_bps	Enregistre les standards des pertes selon la combinaison (produit, ligne, bps)	<u>id</u>	Identifiant	Numérique
		dosage_cible		Numérique
		dosage_min		Numérique
		dosage_max		Numérique
		standard_perte_bps_dem_nb	Standard pertes BPS au démarrage en nbr	Numérique
		standard_perte_bps_dem_kg	En kg	Numérique
		standard_perte_bps_finprod_nb	Fin de prod	Numérique
		standard_perte_bps_finprod_kg		Numérique
		standard_perte_bps_changop_nb	Changement opercule	Numérique

		standard_perte_bps_changop_kg		Numérique
		standard_perte_bps_changdec_nb	Changement décor	Numérique
		standard_perte_bps_changdec_kg		Numérique
		standard_perte_bps_changbps_nb	Changement BPS	Numérique
		standard_perte_bps_changbps_kg		Numérique
		standard_perte_bps_changeq_nb	Changement Equipe	Numérique
		standard_perte_bps_changeq_kg		Numérique
		standard_perte_bps_changparfum_nb	Changement parfum	Numérique
		standard_perte_bps_changparfum_kg		Numérique
Prod_ingr	Les produits et les ingrédients associés	id	Identifiant	Numérique
		id_ingr	Identifiant ingrédient	Numérique
		id_produit	Identifiant produit	Numérique
Production	Les informations de production	id	Identifiant	Numérique
		heure_deb	Heure début	time
		heure_fin	Heure fin	time
		date_deb	Date début	Date
		date_fin	Date fin	Date
Prtes Ingrédients	Les pertes ingrédients	id	Identifiant	Numérique
		qtt_Pertes	Quantité Pertes	Numérique
		type_Perte	Type Perte	Alphanumérique
PerteArôme	Les pertes arôme	poids_debut	Poids début	Numérique
		poids_fin	Poids fin	Numérique
		nbr_pots	Nbr pots	Numérique
		%P(1)	Dosage P1	Numérique
		%P(2)	Dosage P2	Numérique
		injecte1	Injecté	Numérique
		reste1	Reste	Numérique
		injecte2		Numérique
		reste2		Numérique
		injecte3		Numérique
		reste3		Numérique
		reel	Réel	Numérique
PerteFruit	Les pertes fruit	amorçage	Amorçage	Numérique
		poids_debut	Poids début	Numérique
		poids_fin	Poids fin	Numérique
		dosage_reel	Dosage réel	Numérique
		dosage_pompe1	Dosage pompe 1	Numérique
		dosage_pompe2		Numérique
		dosage_pompe3		Numérique
		injecte1		Numérique
		reste1		Numérique
		tonnage_realise	Tonnage réalisé	Numérique
	poids_pertes	Les pertes en	Numérique	

			kg	
		taux_pertes	Le % des pertes	Numérique
Pertes masse	Les pertes masse	id	Identifiant	Numérique
		ecrasement	Pertes écrasement	Numérique
		echantillon_qualite	Pertes échantillons qualité	Numérique
		pousses_debut	Pertes pousses début	Numérique
		pousses_fin	Pertes pousses fin	Numérique
		amorçage	Pertes amorçage	Numérique
		amorçage_volume	Volume pertes amorçage	Numérique
		autres_pertes	Autres pertes	Numérique
		cause	Causes d'autres pertes	Alphanumérique
		total_perte_produit_kg	Total des pertes produit en kg	Numérique
Pertes emballage	Les pertes emballage	id	Identifiant	Numérique
		pertes_demarrage	Pertes au démarrage de la production	Numérique
		perte_dem_essai	Perte au démarrage en essai	Numérique
		comment_dem	commentaire	Alphanumérique
		pertes_changement	Changement	Numérique
		perte_chang_essai		Numérique
		comment_chang		Alphanumérique
		perte_fin_prod	Fin de production	Numérique
		comment_finprod		Alphanumérique
		pertes_qualite_mp	Qualité matière première	Numérique
		perte_qualite_mp_essai		Numérique
		comment_qualite_mp		Alphanumérique
		pertes_autre	Autres pertes	Numérique
		perte_autre_essai		Numérique
		comment_autre		Alphanumérique
		perte_panne	Pertes à cause de pannes	Numérique
		prte_panne_essai		Numérique
comment_panne		Alphanumérique		

		pots_consommés	Nbr pots à produire	Numérique
		pots_produits	Nbr pots à produire	Numérique
		taux_pertes	Nbr pots à produire	Numérique
Control fruit	Le control fruit	id	Identifiant	Numérique
		effectuee	effectué	Alphanumérique
Control arôme	Le control arôme	id	Identifiant	Numérique
		reglage_pompe1_1	Réglage pompe 1	Numérique
		reglage_pompe1_2	Réglage 2	Numérique
		reglage_pompe2_1		Numérique
		reglage_pompe2_2		Numérique
		poids_avant_amorçage1	Poids avant amorçage	Numérique
		poids_avant_amorçage2		Numérique
		poids_apres_amorçage1	Poids après amorçage	Numérique
		poids_apres_amorçage2		Numérique
Série dosage	Les séries de dosage	id	Identifiant	Numérique
		num_b1	N° buse	Numérique
		heure_b1	Heure	time
		date_b1	Date	Date
		poids_b1	dosage	Numérique
			
			
		num_b24	N° buse	
		heure_b24	Heure	
		date_b24	Date	
		poids_b24	dosage	
Resultats_dosage	Les résultats de dosage	id	Identifiant	Numérique
		moy_dosage	La moyenne du dosage	Numérique
		resultat_dosage	Résultat dosage en %	Numérique
		pertes_dosage	Pertes dosage en Kg	Numérique
Camion	Stock les camions	id	Identifiant	Numérique
		numCamion	N°Camion	Alphanumérique
LaitCrueJete	Stock les quantités de lait jetés	id	Identifiant	Numérique
		qteLaitJete	Quantité Lait Jeté	Numérique
		numCuve	N° Cuve	Numérique
		Probleme	Cause non conformité	Alphabétique
		Commentaires	Commentaire	Alphanumérique

Tableau 9: Dictionnaire de données

3.2. Passage au modèle relationnel

3.2.1. Les règles du passage

Règle1 : Transformation des classes

Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle de l'identifiant. Dans le cas où aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire.

Règle2 : Association un-à-plusieurs

Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association.

L'attribut porte le nom de la clé primaire de la relation père de l'association.

Règle3 : Associations plusieurs-à-plusieurs

L'association devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectés à l'association. Les attributs de l'association doivent être ajoutés à la nouvelle relation.

Règle4 : Association un-à-un

Il faut ajouter un attribut clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association.

Si les deux cardinalités (multiplicités) minimales sont à zéro, le choix est donné entre les deux relations dérivées de la règle R1. Si les deux cardinalités minimales sont à un, il est sans doute préférable de fusionner les deux entités (classes) en une seule.

Règle5 : Transformation de l'héritage

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

– Décomposition par distinction : il faut transformer chaque sous-classe en une relation.

La clé primaire de la surclasse, migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.

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

– Décomposition ascendante (push up) : il faut supprimer la (les) relation(s) issue(s) de la (des) sous classe(s) et faire migrer les attributs dans la relation issue de la surclasse. [11]

Règle6 : Agrégation

L'agrégation partagée de UML se traduit au niveau logique comme une simple association, il n'est pas de même pour la composition.

Le modèle relationnel doit refléter le diagramme de classes issu de l'analyse, et donc les éléments présents dans le diagramme des classes (classe, attribut, association, classe association) doivent se retrouver dans le modèle relationnel [12].

3.2.2. Le modèle relationnel associé au diagramme de classe de l'application

Camion (id_camion, heureDepotage, numCamion, id_arriveeLait#)

LaitCrueJete (id_LaitCrueJete, qteLaitCrueJete, numCuve, probleme, commentaires, idCamion#)

Correction (id_Correction, qte_cf, qte_pl26%, qte_pl0%, id_MasseBlanche#)

Produit (id_produit, code, designation, proteine, matiereGrasse, id_masse#)

Prod_Ingr (id_produit#, id_ingredient#)

Correction (id_correction, qtt_cf, qtt_pl26%, qtt_pl0%, id_masse#)

Production (id_production, heure_deb, heure_fin, date_deb, date_fin, id_Prod_Ingr#, id_prod_ligne_bps#)

Pertes_emballage (id_perte_emballage, pertes_demarrage, perte_dem_essai, comment_dem, pertes_changement, perte_chang_essai, comment_chang, perte_fin_prod, comment_finprod, pertes_qualite_mp, perte_qualite_mp_essai, comment_qualite_mp, pertes_autre, perte_autre_essai, comment_autre, perte_panne, prte_panne_essai, comment_panne, id_production#)

Pertes_ingredients (id_ingredient, rqt_Pertes, type_Perte, id_production#)

Pertes_masse (id_panier, ecrasement, echantillon_qualite, pousses_debut, pousses_fin, amorçage, amorçage_volume, autres_pertes, cause, total_perte_produit_kg, id_production#)

Control fruit (id_panier, effectue, id_production#)

Control_arôme (id_panier, reglage_pompe1_1, reglage_pompe1_2, reglage_pompe2_1, reglage_pompe2_2, poids_avant_amorçage1, poids_avant_amorçage2, poids_apres_amorçage1, poids_apres_amorçage2, id_production#)

Série dosage (id_panier, num_b1, heure_b1, date_b1, poids_b1..., num_b1, heure_b1, date_b1, poids_b1, id_resultats_dosage#)

ProduitLigneBPS (id_ProduitLigneBPS, dosage_cible, dosage_min, dosage_max, standard_perte_bps_dem_nb, standard_perte_bps_dem_kg, ..., id_produit#, id_ligne_cond#, id_emballageBPS#)

4. Conclusion

Dans ce chapitre nous avons présenté l'étape conception, où nous avons décrit d'une façon détaillée la modélisation en utilisant le langage de modélisation «UML», ainsi que le processus unifié qui nous a aidé de faire une conception détaillée avec les différents diagrammes.

Nous avons aussi réalisé le diagramme de classe et le schéma relationnel à implémenter en base de données de l'application. On a donc, pu déterminer précisément le fonctionnement de notre application.

Dans le prochain chapitre nous allons présenter la phase de réalisation de notre application.

Chapitre IV

Réalisation

Chapitre IV

Réalisation

1. Introduction

Dans ce chapitre, consacré à la réalisation et la mise en œuvre de notre application, nous allons présenter les outils de développement adoptés ainsi qu'un aperçu général sur la phase pratique qui consiste à décrire les différentes fonctionnalités de notre logiciel et l'architecture de l'application.

2. Langages environnement de développement

Le logiciel que nous avons réalisé est une application Java qui sera déployée sur le réseau local de l'entreprise DANONE DJURDJURA.

2.1. Langages de programmation et outils utilisés

2.1.1. Java

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.

2.1.2. Javafx

JavaFX est une bibliothèque graphique intégrée dans le JRE et le JDK de Java. Oracle la décrit comme « The Rich Client Platform », c'est-à-dire qu'elle permet de réaliser des interfaces graphiques évoluées et modernes grâce à de nombreuses fonctionnalités, telles que les animations, les effets, la 3D, l'audio, la vidéo, etc. Elle a de plus l'avantage d'être dans le langage Java, qui permet de réaliser des architectures avec des paradigmes objet, et aussi de pouvoir utiliser le typage statique [13]

2.1.3. JFoenix

JFoenix est une bibliothèque Java open source, qui implémente Google Material Design à l'aide de composants Java. Pour commencer à utiliser JFoenix, il vous suffit de le télécharger depuis GitHub et de l'intégrer à l'IDE de javafx.

2.1.4. XAMPP

XAMPP est un ensemble de logiciels libres. Le nom est un acronyme venant des initiales de tous les composants de cette suite. Ce dernier réunit donc le serveur Web Apache, la base de données relationnelle et système d'exploitation MySQL ou MariaDB ainsi que les langages scripts Perl et PHP. L'initiale X représente tous les systèmes d'exploitation possibles, à savoir Linux, Windows et Mac OS X.

Apache: le serveur Web open source Apache est utilisé mondialement et permet de délivrer des contenus Web. L'application de serveur est mise à disposition en open source par l'Apache Software Foundation.

MySQL/MariaDB: avec MySQL, XAMPP se compose de l'un des systèmes de gestion de base de données relationnelle les plus populaires au monde. En combinaison avec le serveur Web Apache et le langage script PHP, MySQL sert à l'enregistrement de données pour des services Web. Les versions actuelles de XAMPP favorisaient MariaDB à l'insu de MySQL comme gestionnaire de base de données, marquant un détachement avec ce dernier.

PHP: Il s'agit d'un langage script côté serveur permettant de créer des pages Web ou applications dynamiques. PHP peut être mis en place sur toutes les plateformes possibles et est compatible avec divers systèmes de base de données.

Perl: le langage script Perl est utilisé pour l'administration système, le développement Web et la programmation en réseau. De plus, des applications Web dynamiques peuvent être programmées de la même manière que PHP.

En dehors des composants principaux, la distribution gratuite d'Apache comprend divers outils selon chaque système d'exploitation comme le serveur Mail Mercury, l'application d'administration de base de données PhpMyAdmin, le logiciel d'analyse de données Webalizer, OpenSSL, ApacheTomcat ainsi que FileZilla ou ProFTPD [14].

2.1.5. StarUML

StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux comme Rational Rose ou Together. Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation. Cependant, seule une version Windows est disponible.

Site web du logiciel : <http://staruml.sourceforge.net/en/>

2.1.6. Visual Paradigm

Visual Paradigm est un logiciel de création de diagrammes dans le cadre d'une programmation. Tout en un, il possède plusieurs options permettant une large possibilité de modélisation en UML.

Principales fonctionnalités

- Modélisation : le logiciel **Visual Paradigm** offre de nombreux outils pour créer différents types de schémas comme les diagrammes d'exigences et de cas d'utilisation. Il possède bon nombre de navigateurs permettant de personnaliser chaque élément.

- Analyse et manipulation de codes sources : **Visual Paradigm** permet de générer des codes sources en divers langages comme le Java ou C++ à partir du modèle créé. Inversement, il permet de produire un modèle à partir de codes sources.
- Capture avec la souris : **Visual Paradigm** permet d'utiliser la souris pour manipuler facilement le diagramme à créer. Tous les types d'actions sont pris en charge dont le traçage et la commande ainsi que la connexion des éléments.

2.2. Environnement de programmation

2.2.1. Eclipse IDE

Eclipse est un IDE (un environnement de développement intégré) écrit en Java et conçu avec des fonctionnalités permettant de simplifier le développement d'applications Java. Cet IDE est réputé multi langage, multiplateforme et extensible par des greffons ou plug-ins. Il est avant tout conçu pour le langage Java, mais ses nombreux greffons en font un environnement de développement de choix pour de nombreux autres langages de programmation (C/C++, Python, PHP, Ruby...) [5].

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, reporting... Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.

2.2.2. Scenebuilder

Scene Builder est un outil interactif de conception d'interface graphique pour JavaFX. Il permet de créer des interfaces utilisateurs rapidement et sans avoir besoin de coder: il en résulte des fichiers au format FXML qui sont ensuite chargés par le programme pour afficher une interface graphique à ses utilisateurs.

Développé initialement par Oracle et sous le nom JavaFX Scene Builder⁵, son code source a été publié en open source à partir de sa version 2.0.

3. Schéma Physique de la Base de données

La base de données a été implantée en utilisant l'application Web PhpMyAdmin (outil intégré dans XAMPP), ce dernier nous permet de schématiser les tables et leurs relations (clé étrangère) comme indiqué à la (figure 21)

Dans le schéma physique, nous indiquons les types des attributs de chaque table, les clés primaires, les clés étrangères ainsi que les champs références.

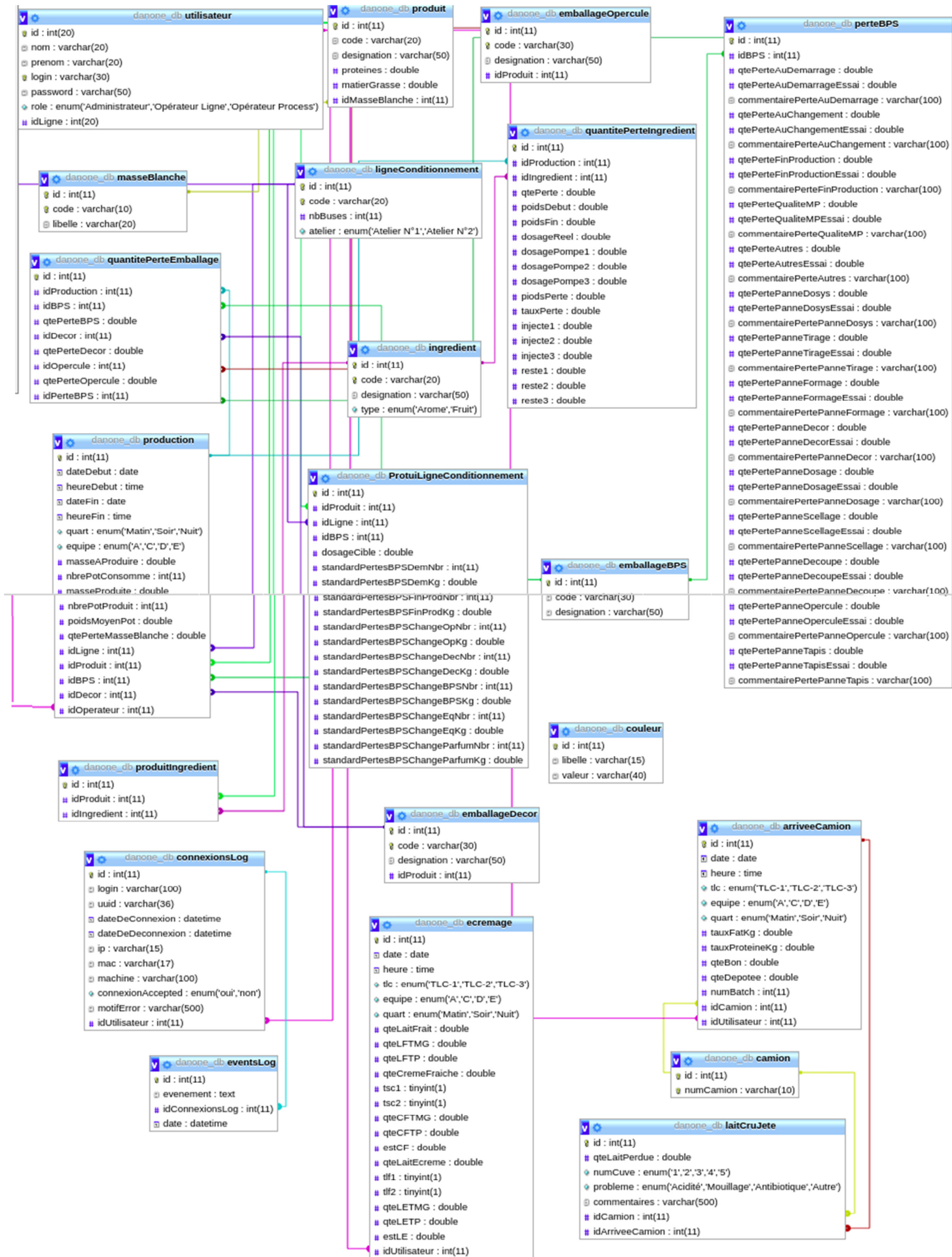


Figure 21: Schéma physique de la base de données

4. Architecture de l'application

Après avoir présenté la partie de données de notre application, nous allons expliquer l'architecture de l'Application en termes de déploiement des modules qui la constituent sur les différentes machines (Serveur, Ordinateur de travail).

L'architecture à deux niveaux est une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client.

Dans l'architecture à trois niveaux par contre, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web/serveur de base de données par exemple). L'architecture à trois niveaux permet :

- Une plus grande flexibilité/souplesse.
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau.
- De meilleures performances, étant donné le partage des tâches entre les différents serveurs.

C'est pour ces raisons donc que nous avons choisi l'architecture 3 tiers pour notre application, c'est la plus appropriée et la plus adaptée aux besoins de l'application.

La figure 22 représente l'architecture de notre application où on remarque une architecture partagée entre :

1. Le client : le demandeur de ressources qui est le Niveau 1 (Application Desktop)
2. Le serveur d'application (appelé aussi middleware) : le serveur chargé de fournir la ressource mais faisant appel à un autre serveur pour certaine demande de ressources (Niveau 2).
3. Le serveur secondaire (généralement un serveur de base de données), fournissant un service (ressources) au premier serveur (Niveau 3). [15]

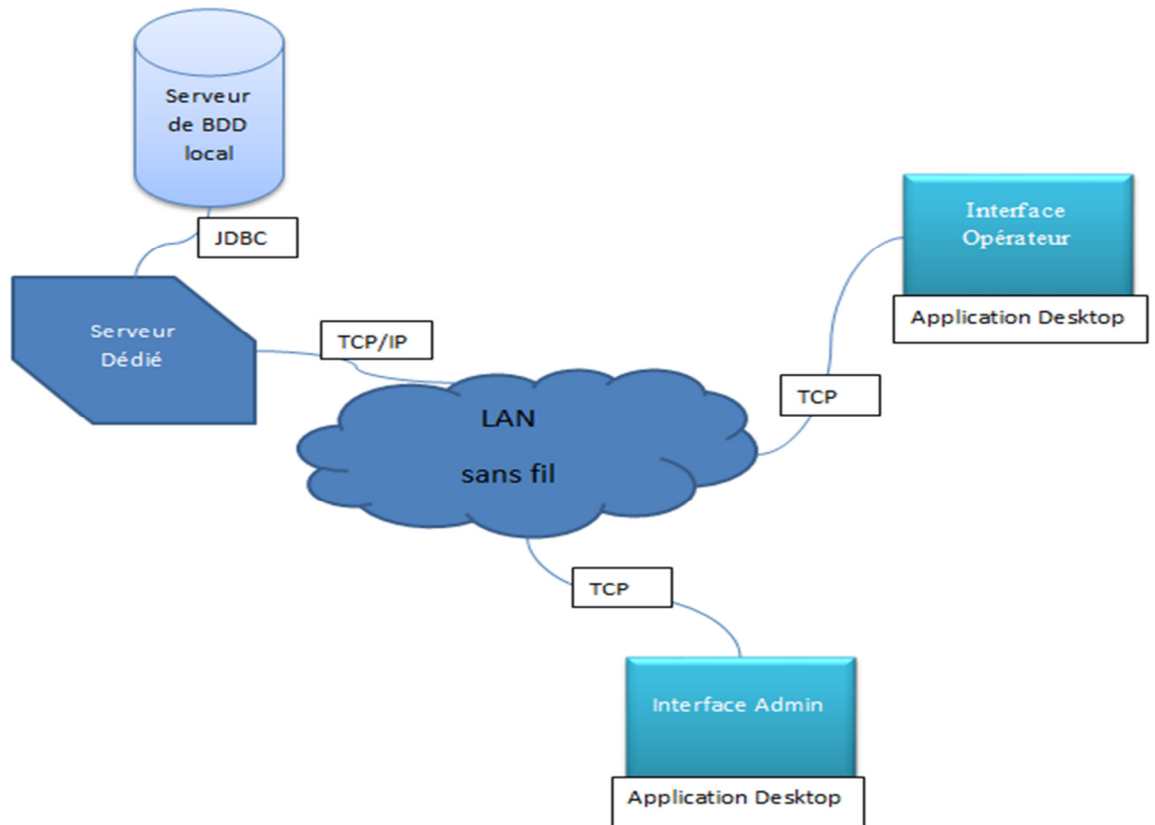


Figure 22: Architecture de l'application

4.1. Diagramme de déploiement

Le modèle de déploiement exprime la répartition physique des fonctions métier de système et permet de justifier la localisation des bases de données et des environnements de travail. Le modèle de déploiement aide à préciser la qualification des postes clients et de leur sécurité [6].

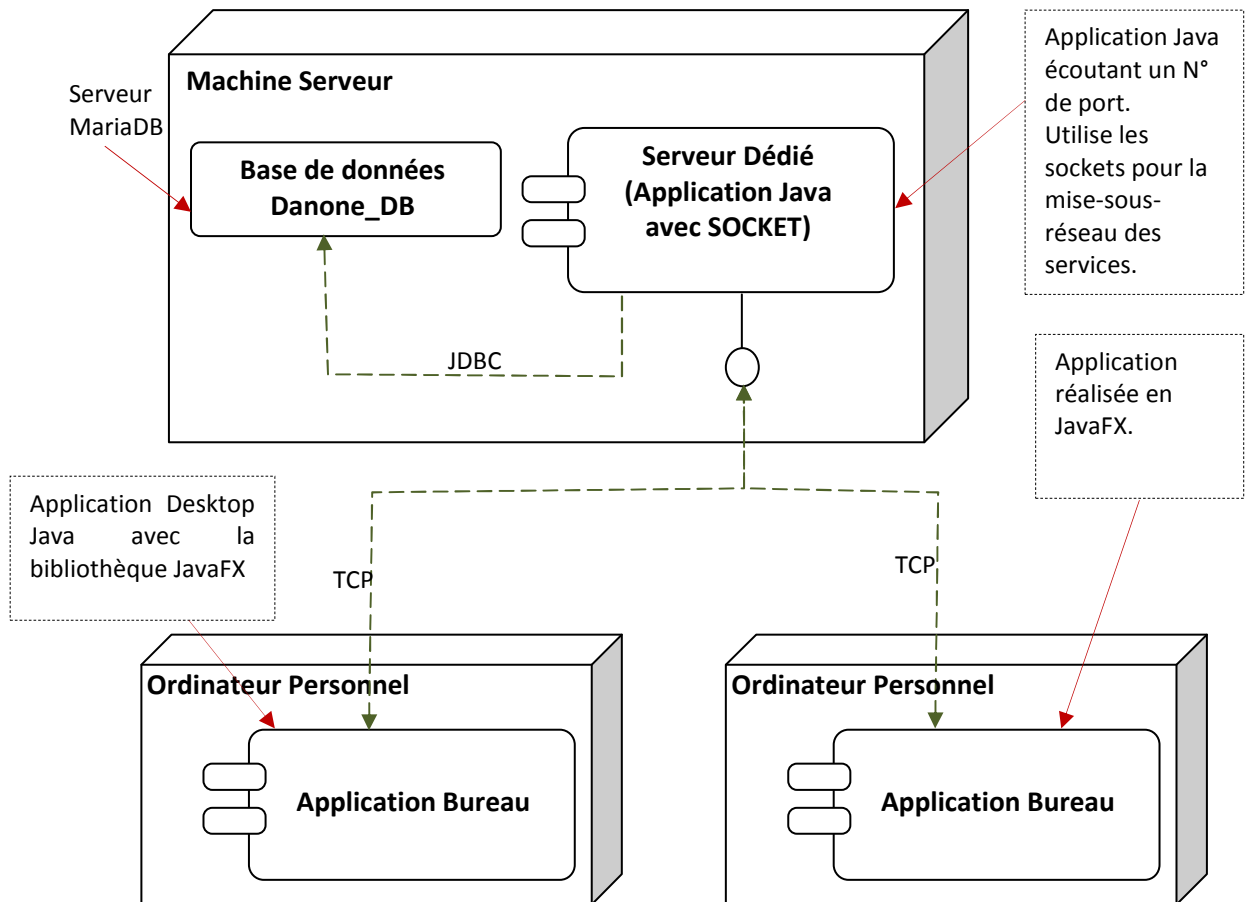


Figure 23: Diagramme de déploiement

Chaque acteur a son propre poste client qui est un " PC " connecté au serveur de l'application, qui est lui-même un client du serveur de base de données (SGBD).

4.2. Structuration du code source

Dans cette partie, nous allons discuter de l'architecture de notre application en termes de structure de code source. Ceci permet d'ajouter un éclaircissement sur le diagramme de déploiement ci-dessus présenté.

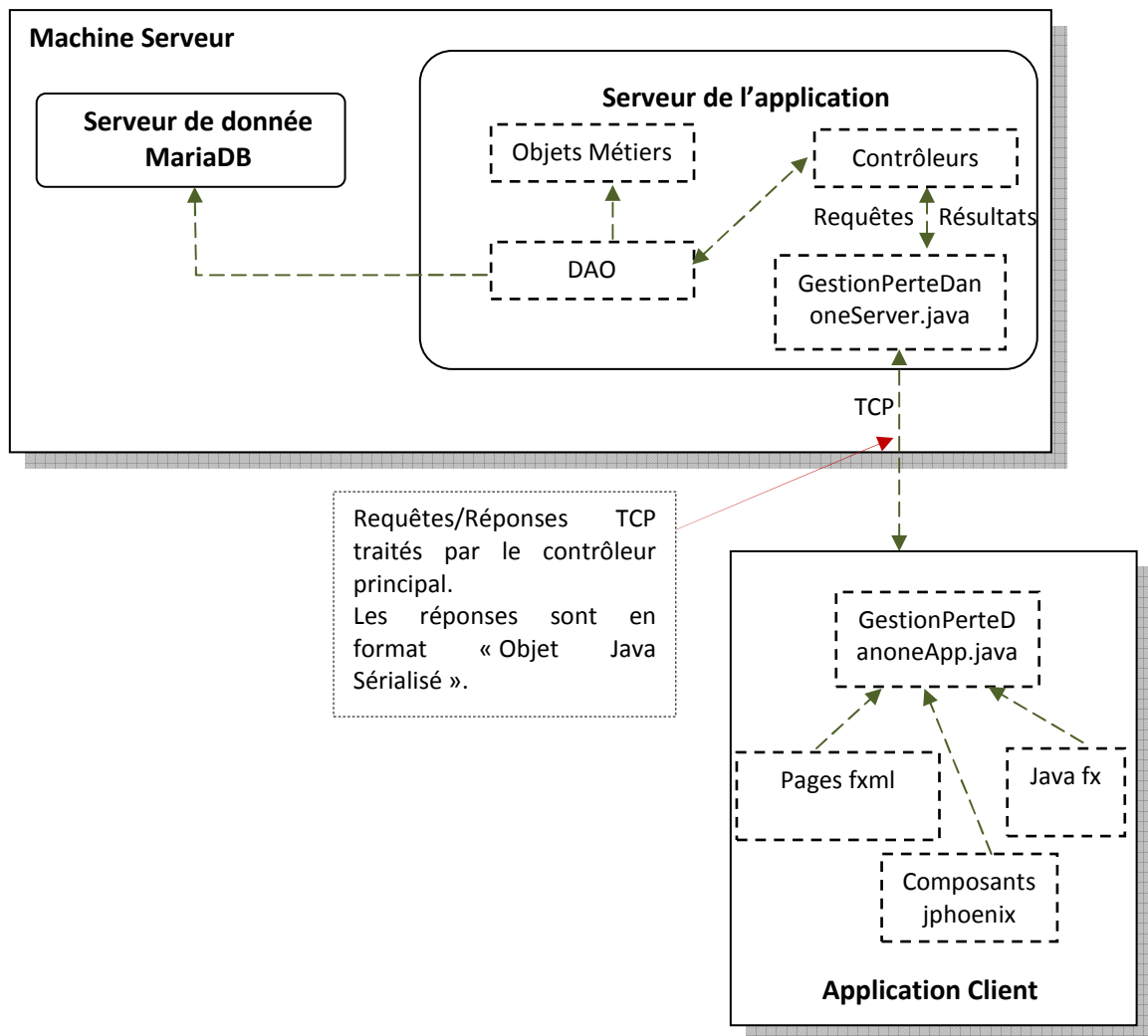


Figure 24: Structure de code source

5. Arborescence de l'application (Diagramme de navigation)

Le schéma ci-après, présenté dans la figure (24) récapitule l'organisation des pages de l'application (Diagramme de navigation)

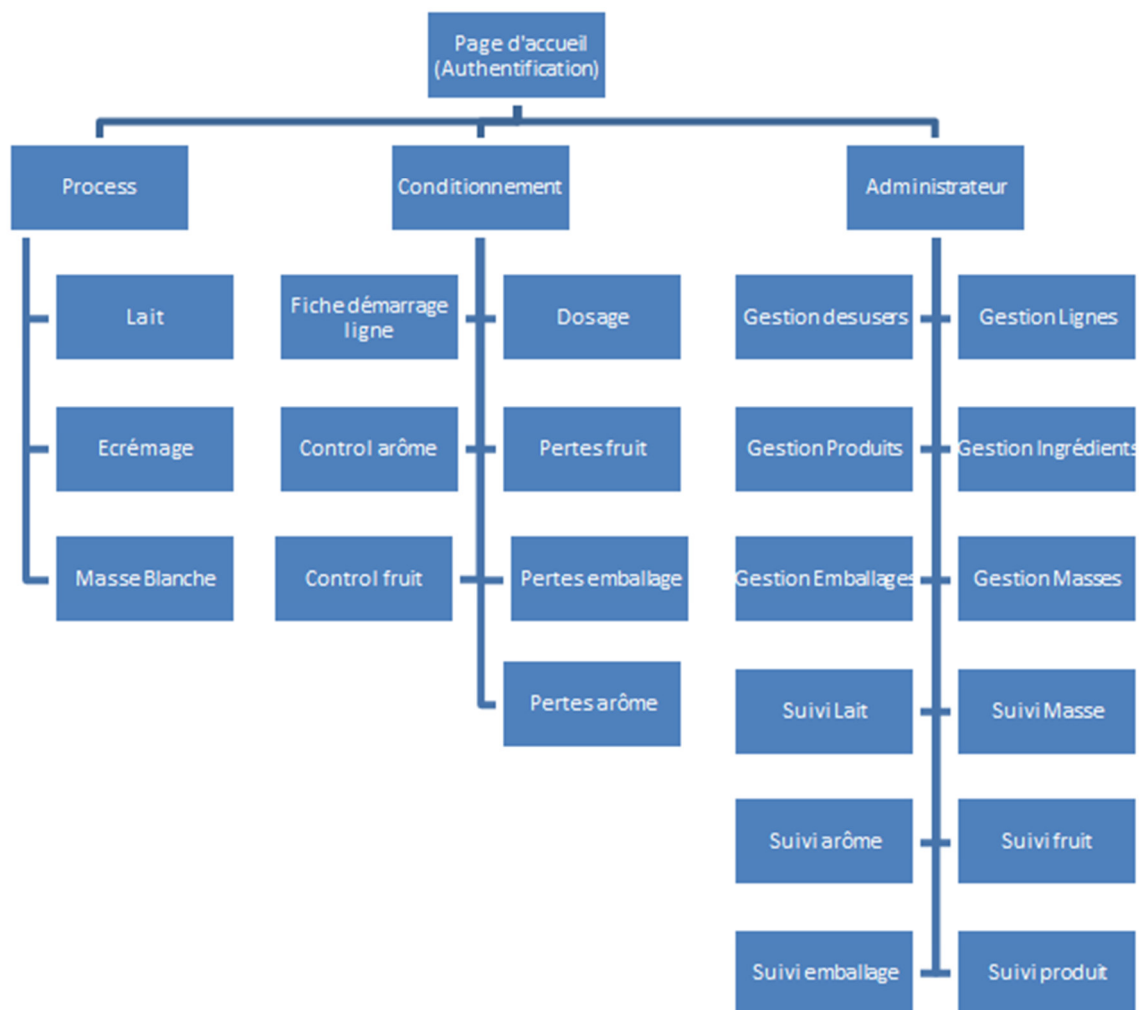


Figure 25: Arborescence de l'application

Note : Toutes les pages de gestion disposent de boutons menant vers les pages : Ajouter, modifier et supprimer.

6. Interfaces homme-machines (IHM)

Nous allons maintenant entamer la phase pratique qui consiste à décrire les différentes fonctionnalités de notre application en présentant les interfaces associées :

- **Interface "Authentification " (Page d'accueil de l'application)**

Cette interface est la page d'accueil de l'application. Elle permet à l'utilisateur de saisir son login et mot de passe afin d'accéder à l'interface associée, selon son rôle.



Figure 26: Interface "Authentification "

- **Interface "Lait " de l'opérateur process**

Après l'authentification, l'opérateur process accède à son espace où il aura la page lait pour saisir toutes les informations concernant la réception du lait puis valider.

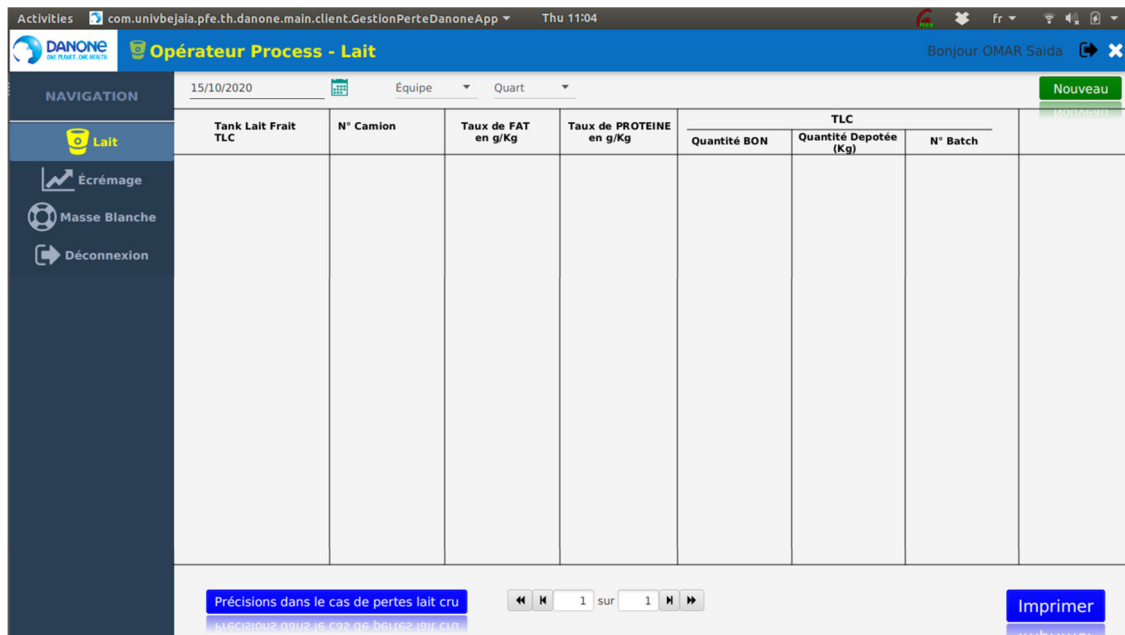


Figure 27: Interface "Lait " de l'opérateur process

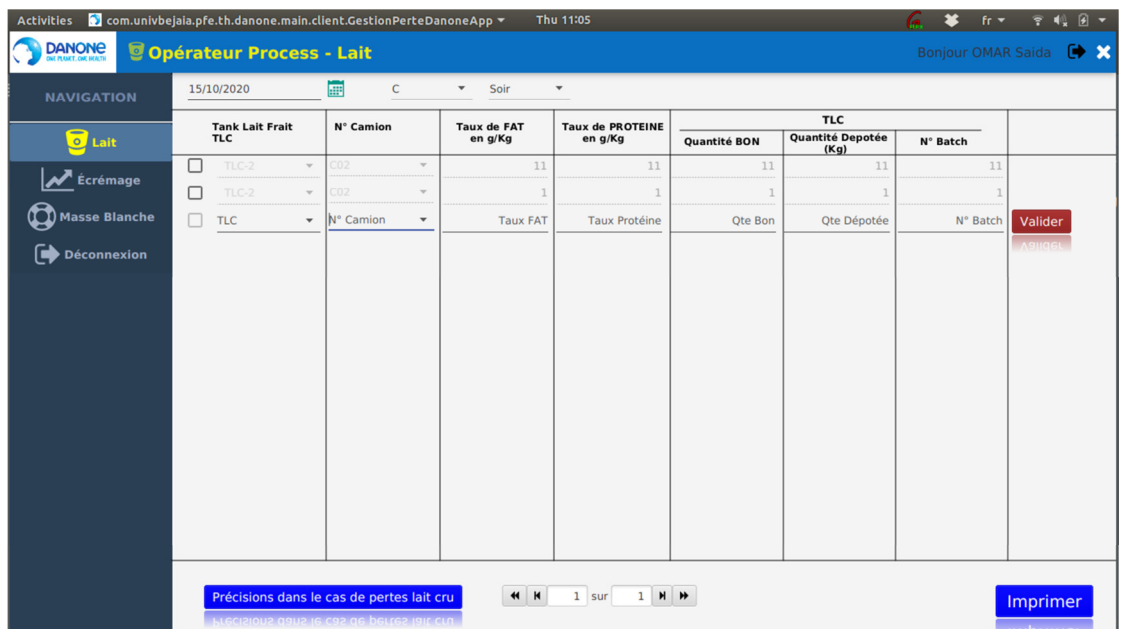


Figure 28: Interface "Lait " de l'opérateur process

- Interface "Ecrémage " de l'opérateur process

L'opérateur process a aussi accès à l'interface écrémage où il va saisir les informations de l'opération d'écrémage (entrée, sortie).

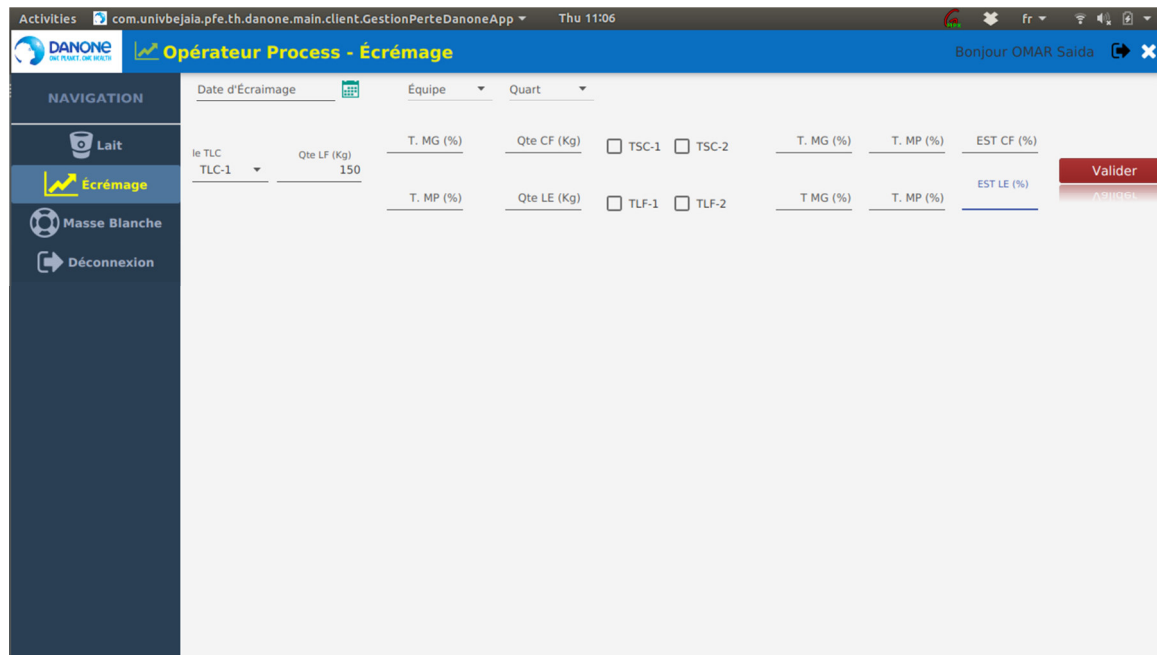


Figure 29: Interface "Écrémage " de l'opérateur process

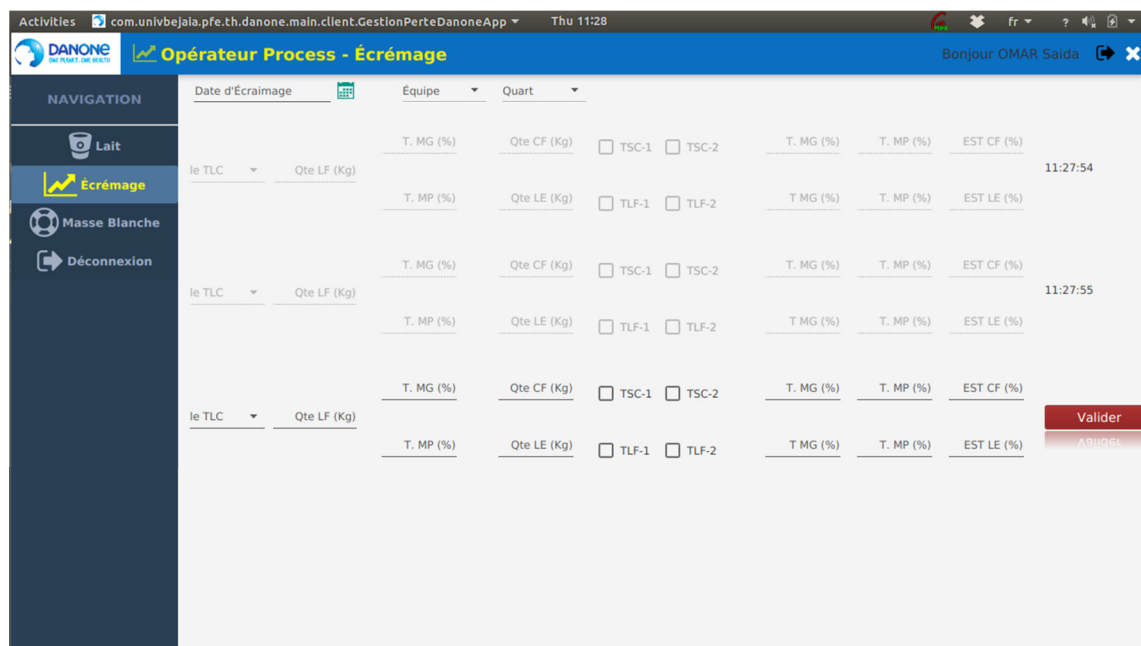


Figure 30: Interface "Écrémage " de l'opérateur process

- Interface "Masse Blanche " de l'opérateur process

L'interface masse blanche permet à l'opérateur process de saisir les informations concernant la production de la masse blanche.

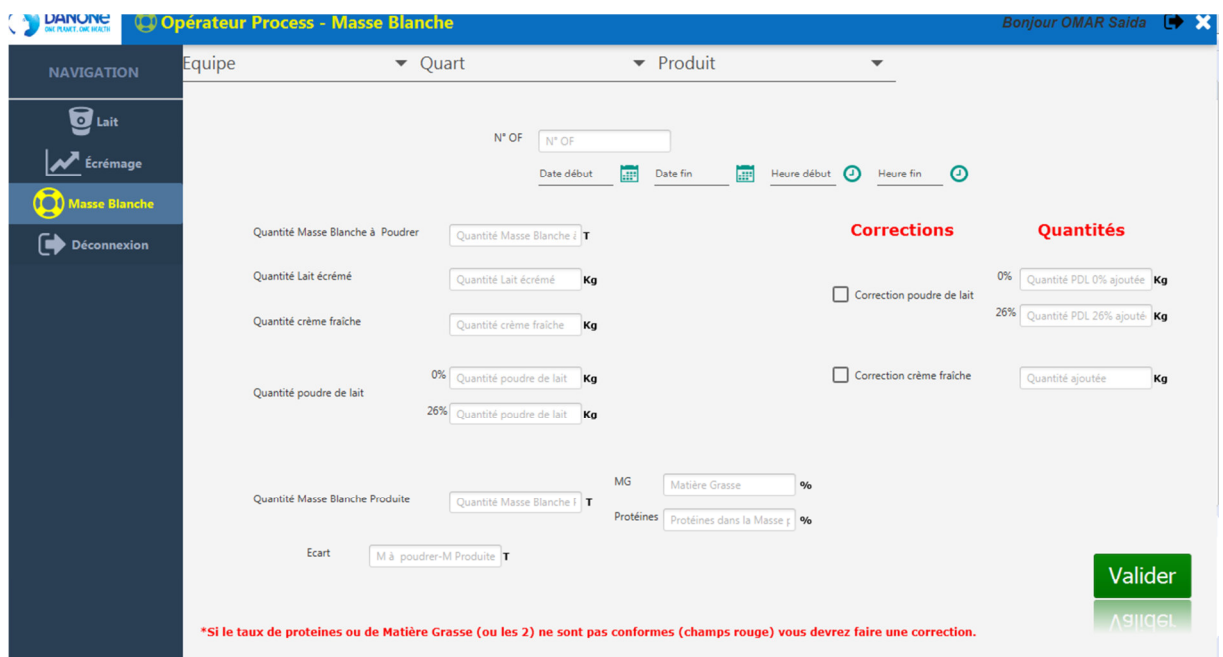


Figure 31: Interface "Masse Blanche " de l'opérateur process

- Interface "Gestion des utilisateurs " de l'administrateur

L'administrateur, après authentification, peut gérer les utilisateurs (Ajout, modification ou suppression) via l'interface Utilisateurs.

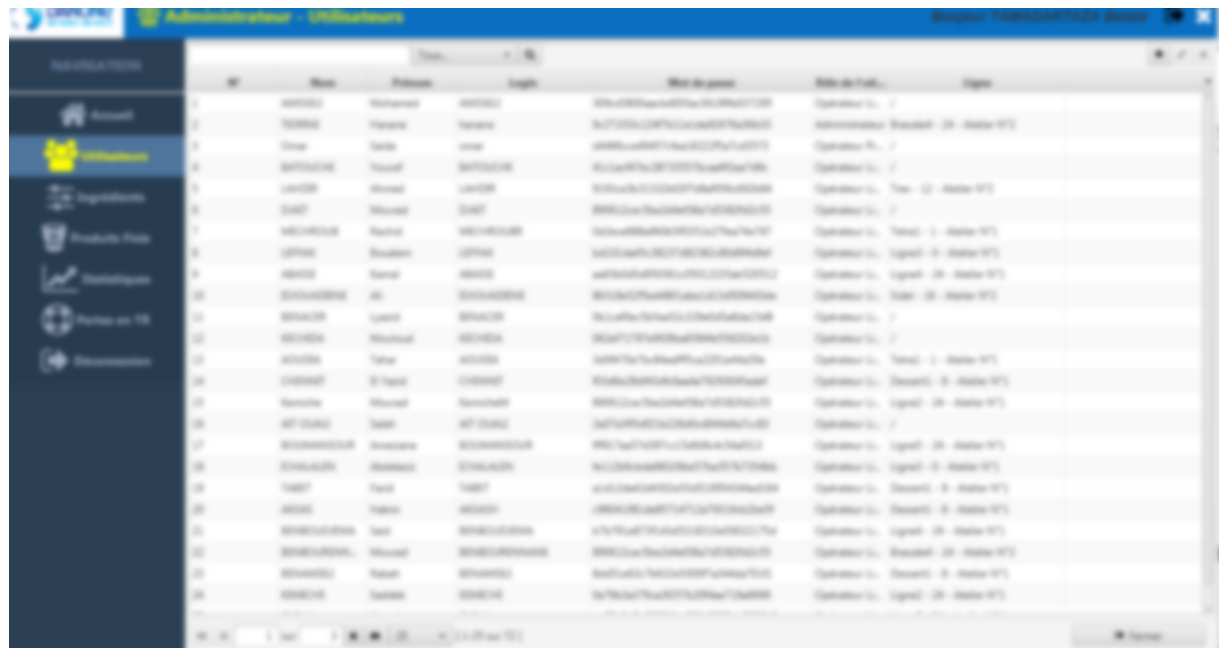


Figure 32: Interface "Gestion des utilisateurs " de l'administrateur

- Interface "Ajouter utilisateur " de l'administrateur

L'administrateur peut ajouter un utilisateur.

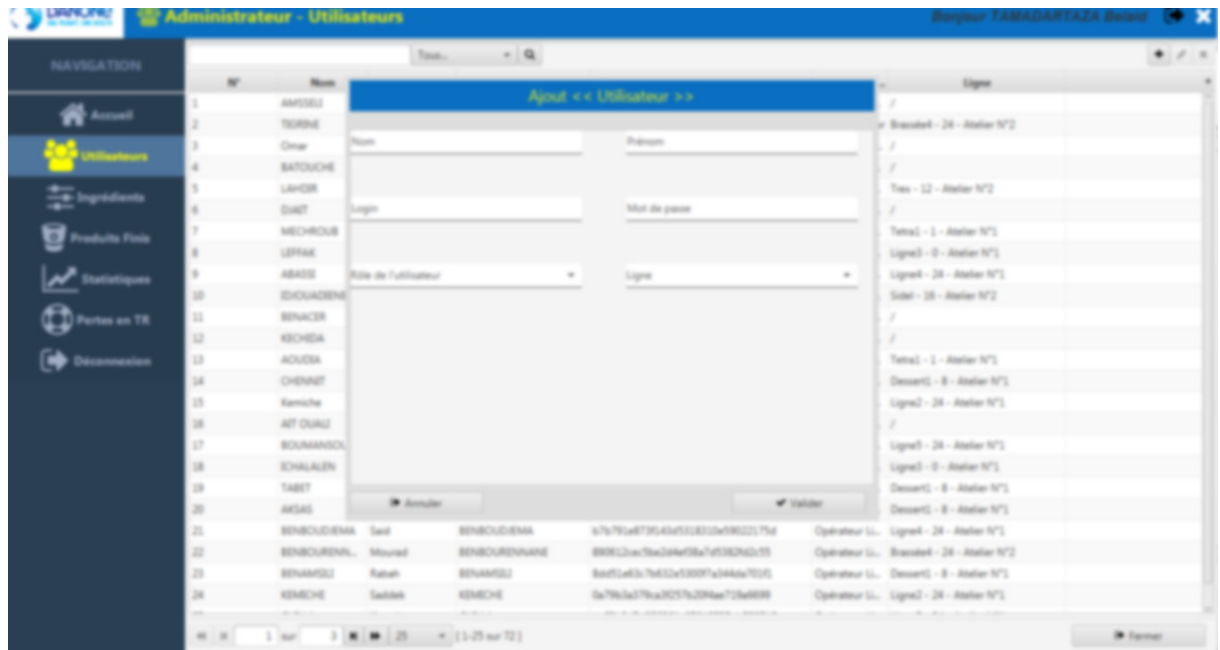


Figure 33: Interface "Ajouter utilisateur " de l'administrateur

- Interface "Modifier utilisateur " de l'administrateur

L'administrateur peut modifier un utilisateur.

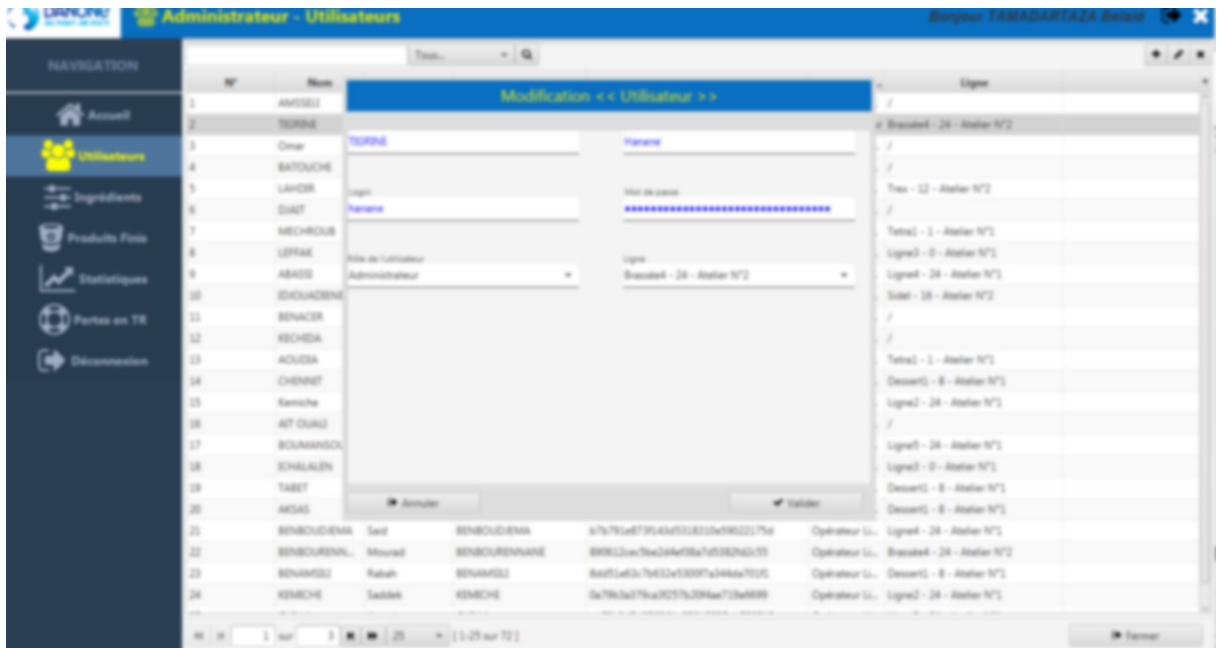


Figure 34: Interface "Modifier utilisateur " de l'administrateur

- Interface "Gestion des ingrédients " de l'administrateur

L'administrateur, après authentification, peut gérer les ingrédients (Ajout, modification ou suppression) via l'interface Ingrédients.

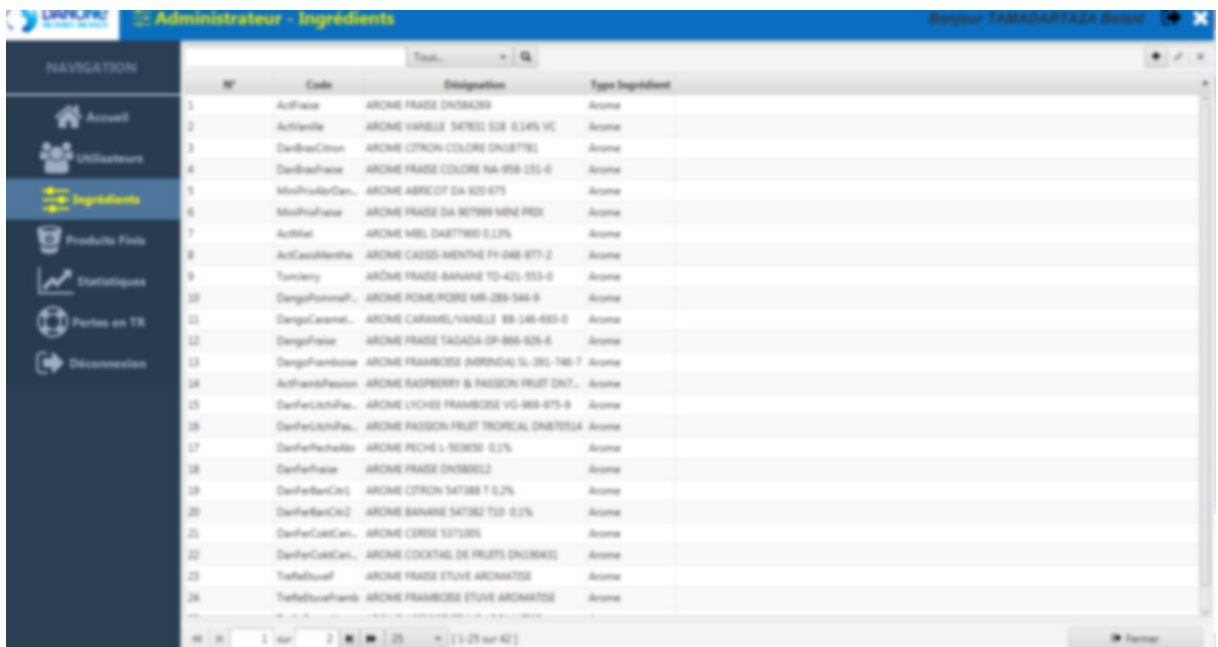


Figure 35: Interface "Gestion des ingrédients " de l'administrateur

7. Conclusion

Dans ce chapitre nous avons présenté les aspects pratiques liés à la réalisation de notre application, à savoir les outils de développement nécessaires pour le fonctionnement de notre application suivi du diagramme de déploiement associé à notre système et le schéma physique associé. En dernier, nous avons illustré quelques interfaces de l'application.

Conclusion

Général

CONCLUSION

GENERALE

Conclusion générale

Notre projet « SPP » consiste à la réalisation d'un système d'information automatique sous réseau local pour le suivi des pertes en temps réel au niveau de la production d'une entreprise de production laitière. L'objectif principal visé par ce projet est de faciliter et d'accélérer toutes les tâches de suivi de production que ce soit en matières premières ou en produits finis, ainsi que l'échange d'informations entre les opérateurs saisissant les pertes et les suivis et le service performance les visualisant .

La contribution de ce modeste travail est présentée en l'application de toutes les étapes de la méthodologie UML dans l'étude avec une démarche simple et générique qui se situe à mi-chemin entre UP (Unified Process), qui constitue un cadre général très complet de processus de développement, et XP (eXtreme Programming) qui est une approche minimaliste à la mode centrée sur le code, et le langage de programmation JAVA et le SGBD MariaDB sous XAMPP.

Nous avons réalisé un système qui permet :

- Une grande souplesse pour l'organisation où un nombre de documents papiers ont été supprimés.
- La représentation des informations d'une manière fiable, efficace et rapide, surtout dans l'établissement des différents documents.
- La possibilité d'établir les statistiques et de donner des résultats exacts et précis en permettant un grand gain de temps.

- Et comme notre logiciel fonctionne sous réseau local, il facilite le transfert de données entre les différents postes des opérateurs et des administrateurs.

Ainsi, le stage que nous avons effectué au sein de l'entreprise DANONE DJURDJURA, service performance, a été pour nous une expérience très enrichissante et très intéressante, dans la mesure où il nous a permis de mettre en pratique et de développer les connaissances que nous avons acquises tout au long de notre cursus universitaire.

Enfin, nous espérons que notre travail puisse répondre aux attentes et aux besoins des clients et en particulier le service performance de l'entreprise DANONE DJURDJURA Algérie.

En guise de perspectives, nous voulons développer une application mobile pour les utilisateurs afin de faciliter la tâche de saisie aux opérateurs dans les ateliers et rendre le suivi (consultation des statistiques) plus souple à l'administrateur, modifier le design pour l'adapter aux petits écrans des Smartphones, ainsi que la projection des statistiques de l'application sur écran téléviseurs dans les ateliers qui serai en quelque sorte, une aide à la décision, pour détecter les failles et les zones de grosses pertes et les résoudre instantanément.

Annexes

Annexe A

3.5. UML

UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue [3].

UML est un langage pseudo-formel

- UML est fondé sur un méta modèle, qui définit :
 - les éléments de modélisation (les concepts manipulés par le langage),
 - la sémantique de ces éléments (leur définition et le sens de leur utilisation).
- Un méta modèle est une description très formelle de tous les concepts d'un langage. Il limite les ambiguïtés et encourage la construction d'outils.
- Le méta modèle d'UML permet de classer les concepts du langage (selon leur niveau d'abstraction ou domaine d'application) et expose sa structure.
- Le méta modèle UML est lui-même décrit par un méta-méta modèle (OMG-MOF).
- UML propose aussi une notation, qui permet de représenter graphiquement les éléments de modélisation du méta modèle.
- Cette notation graphique est le support du langage UML.

UML cadre l'analyse objet, en offrant :

- différentes vues (perspectives) complémentaires d'un système, qui guident l'utilisation des concept objets,
- plusieurs niveaux d'abstraction, qui permettent de mieux contrôler la complexité dans l'expression des solutions objets.

UML est un support de communication

- Sa notation graphique permet d'exprimer visuellement une solution objet.

- L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.
- Son aspect visuel facilite la comparaison et l'évaluation de solutions.
- Son indépendance (par rapport aux langages d'implémentation, domaine d'application, processus...) en font un langage universel.

3.6. Les points forts d'UML

UML est un langage formel et normalisé

- gain de précision
- gage de stabilité
- encourage l'utilisation d'outils

UML est un support de communication performant

- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

3.7. Les diagrammes d'UML

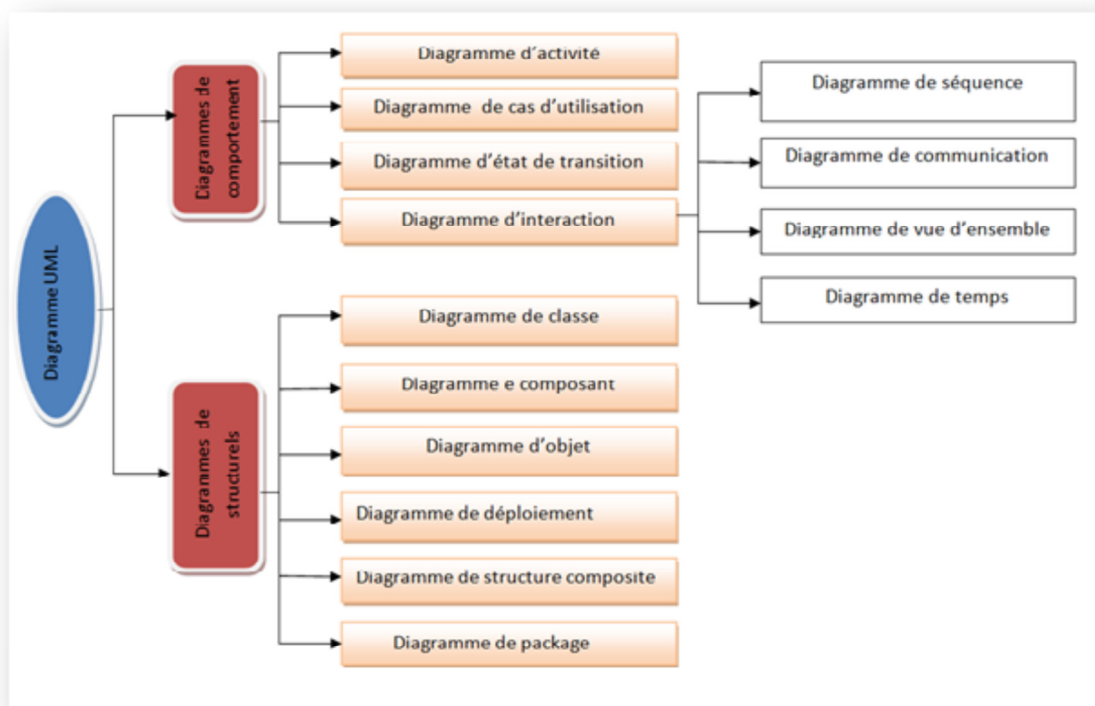


Figure 36: Les diagrammes d'UML

3.8. UML et processus de développement

Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. UML est une sorte de boîte à outils, qui permet d'améliorer progressivement vos méthodes de travail, tout en préservant vos modes de fonctionnement.

Intégrer UML par étapes dans un processus, de manière pragmatique, est tout à fait possible. La faculté d'UML de se fondre dans le processus courant, tout en véhiculant une démarche méthodologique, facilite son intégration et limite de nombreux risques (rejet des utilisateurs, coûts...).

Intégrer UML dans un processus ne signifie donc pas révolutionner ses méthodes de travail, mais cela devrait être l'occasion de se remettre en question, en s'inspirant des meilleures pratiques, capitalisées à travers les processus unifiés (RUP et 2TUP) [4].

Annexe B

Description textuelle des cas d'utilisation

- Cas d'utilisation <<Saisir les informations de l'écémage>> pour l'opérateur process

Sommaire	
Titre	Saisir les informations de l'écémage
Résumé	Saisir les quantités et les informations concernant l'écémage.
Acteurs	Opérateur Process
Description des scénarios	
Pré condition	Etre connecté et être opérateur Process.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur process remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 10: Description du cas d'utilisation <<Saisir les informations de l'écémage>>

- Cas d'utilisation <<Saisir les informations de la masse blanche>> pour l'opérateur process

Cas d'utilisation <<Saisir les informations de la masse blanche>> pour l'opérateur process	
Sammaire	
Titre	Saisir les informations de la masse blanche.
Résumé	Saisir les quantités et les informations concernant la masse blanche.
Acteurs	Opérateur Process
Description des scénarios	
Pré condition	Etre connecté et être opérateur Process.

Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur process remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 11: Description du cas d'utilisation <<Saisir les informations masse blanche>>

- **Cas d'utilisation << Renseigner fiche démarrage ligne>> pour l'opérateur ligne**

Sommaire	
Titre	Saisir les informations de fiche de démarrage de ligne.
Résumé	Saisir les résultats de contrôles effectués sur la ligne avant démarrage.
Acteurs	Opérateur Ligne
Description des scénarios	
Pré condition	Etre connecté et être opérateur Ligne.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur Ligne remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 12: Description du cas d'utilisation <<Renseigner fiche démarrage ligne >>

- **Cas d'utilisation << Control arôme>> pour l'opérateur ligne**

Sommaire	
Titre	Saisir les contrôles arômes.
Résumé	Saisir les informations concernant le contrôle arôme.
Acteurs	Opérateur Ligne
Description des scénarios	
Pré condition	Etre connecté et être opérateur Ligne.

Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur Ligne remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 13: Description du cas d'utilisation <<Control arôme >>

- **Cas d'utilisation << Control emballage >> pour l'opérateur ligne**

Sommaire	
Titre	Saisir les contrôles emballage.
Résumé	Saisir les informations concernant le contrôle emballage.
Acteurs	Opérateur Ligne
Description des scénarios	
Pré condition	Etre connecté et être opérateur Ligne.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur Ligne remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 14: Description du cas d'utilisation <<Control emballage >>

- **Cas d'utilisation << Saisir les pertes en emballage >> pour l'opérateur ligne**

Sommaire	
Titre	Saisir les pertes en emballage.
Résumé	Saisir les quantités de pertes en emballage.
Acteurs	Opérateur Ligne
Description des scénarios	
Pré condition	Etre connecté et être opérateur Ligne.

Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur Ligne remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 15: Description du cas d'utilisation <<Saisir les pertes en emballage >>

- **Cas d'utilisation << Saisir les pertes en arôme>> pour l'opérateur ligne**

Sommaire	
Titre	Saisir les pertes en arôme.
Résumé	Saisir les quantités de pertes en arôme.
Acteurs	Opérateur Ligne
Description des scénarios	
Pré condition	Etre connecté et être opérateur Ligne.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur Ligne remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 16: Description du cas d'utilisation << Saisir les pertes en arôme >>

- **Cas d'utilisation << Saisir les pertes en fruit>> pour l'opérateur ligne**

Sommaire	
Titre	Saisir les pertes en fruit.
Résumé	Saisir les quantités de pertes en fruit.
Acteurs	Opérateur Ligne

Description des scénarios	
Pré condition	Etre connecté et être opérateur Ligne.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur Ligne remplit tous les champs. 2. L'opérateur envoie les informations saisies (valider). 3. Le système vérifie que tous les champs sont remplis. 4. Le système envoie les données vers la base de données et réinitialise les champs (à vide).
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et champs remis à vide.

Tableau 17: Description du cas d'utilisation <<Saisir les pertes en fruit>>

- **Cas d'utilisation << Gestion lignes>> pour l'administrateur**

Sommaire	
Titre	Gestion lignes
Résumé	Ajouter, modifier ou supprimer une ligne.
Acteurs	Administrateur
Description des scénarios	
Pré condition	Etre connecté et être administrateur.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande le formulaire d'ajout, modification ou suppression d'une ligne. 2. Le système affiche le formulaire demandé. 3. L'administrateur saisit les données et valide. 4. Le système vérifie les données. 5. Le système donne l'accès à l'interface correspondante.
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide ou annule.
Post conditions	Informations enregistrées et interface adéquate affichée.

Tableau 18: Description du cas d'utilisation << Gestion lignes >>

- **Cas d'utilisation << Gestion produits>> pour l'administrateur**

Sommaire	
Titre	Gestion produits.
Résumé	Ajouter, modifier ou supprimer un produit.
Acteurs	Administrateur
Description des scénarios	

Pré condition	Etre connecté et être administrateur.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande le formulaire d'ajout, modification ou suppression d'un produit. 2. Le système affiche le formulaire demandé. 3. L'administrateur saisit les données et valide. 4. Le système vérifie les données. 5. Le système donne l'accès à l'interface correspondante.
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide ou annule.
Post conditions	Informations enregistrées et interface adéquate affichée.

Tableau 19: Description du cas d'utilisation << Gestion produits >>

- **Cas d'utilisation << Gestion emballages >> pour l'administrateur**

Sommaire	
Titre	Gestion emballages.
Résumé	Ajouter, modifier ou supprimer un emballage.
Acteurs	Administrateur
Description des scénarios	
Pré condition	Etre connecté et être administrateur.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande le formulaire d'ajout, modification ou suppression d'un emballage. 2. Le système affiche le formulaire demandé. 3. L'administrateur saisit les données et valide. 4. Le système vérifie les données. 5. Le système donne l'accès à l'interface correspondante.
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide ou annule.
Post conditions	Informations enregistrées et interface adéquate affichée.

Tableau 20: Description du cas d'utilisation << Gestion emballage >>

- **Cas d'utilisation << Gestion masses blanches >> pour l'administrateur**

Sommaire	
Titre	Gestion masses blanches.
Résumé	Ajouter, modifier ou supprimer une masse blanche.
Acteurs	Administrateur
Description des scénarios	

Pré condition	Etre connecté et être administrateur.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande le formulaire d'ajout, modification ou suppression d'une masse blanche. 2. Le système affiche le formulaire demandé. 3. L'administrateur saisit les données et valide. 4. Le système vérifie les données. 5. Le système donne l'accès à l'interface correspondante.
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide.
Post conditions	Informations enregistrées et interface adéquate affichée.

Tableau 21: Description du cas d'utilisation << Gestion masse blanche >>

- **Cas d'utilisation << Gestion ingrédients>> pour l'administrateur**

Sommaire	
Titre	Gestion ingrédients (arôme et fruit).
Résumé	Ajouter, modifier ou supprimer un ingrédient.
Acteurs	Administrateur.
Description des scénarios	
Pré condition	Etre connecté et être administrateur.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande le formulaire d'ajout, modification ou suppression d'un ingrédient. 2. Le système affiche le formulaire demandé. 3. L'administrateur saisit les données et valide. 4. Le système vérifie les données. 5. Le système donne l'accès à l'interface correspondante.
Alternative	Les informations saisies sont erronées ou des champs vides. Le système affiche un message d'erreur (vérifiez les informations saisies).
Arrêt	Lorsqu'il valide ou annule.
Post conditions	Informations enregistrées et interface adéquate affichée.

Tableau 22: Description du cas d'utilisation << Gestion ingrédients >>

- **Cas d'utilisation << Suivi lait >> pour l'administrateur**

Sommaire	
Titre	Suivi lait
Résumé	Afficher le suivi des pertes lait par période ou autres critères selon la volonté de l'administrateur.
Acteurs	Administrateur
Description des scénarios	

Pré condition	Etre connecté et être administrateur.
Scénario nominal	1. L'administrateur demande à afficher le suivi lait selon les critères choisis. 2. Le système affiche le suivi lait selon les filtres choisis par l'administrateur (suivi mensuel, annuel, à temps réel...des pertes).
Alternative	/
Arrêt	Lorsqu'il passe à un autre suivi.
Post conditions	Affichage des statistiques dans l'interface adéquate.

Tableau 23: Description du cas d'utilisation << Suivi lait >>

- **Cas d'utilisation << Suivi masse blanche >> pour l'administrateur**

Sommaire	
Titre	Suivi masses blanches.
Résumé	Afficher le suivi des pertes masse blanche par période ou autres critères selon la volonté de l'administrateur.
Acteurs	Administrateur
Description des scénarios	
Pré condition	Etre connecté et être administrateur.
Scénario nominal	1. L'administrateur demande à afficher le suivi masse blanche selon les critères choisis. 2. Le système affiche le suivi masse blanche selon les critères choisis par l'administrateur (suivi mensuel, annuel, à temps réel...des pertes)
Alternative	/
Arrêt	Lorsqu'il passe à un autre suivi.
Post conditions	Affichage des statistiques dans l'interface adéquate.

Tableau 24: Description du cas d'utilisation << Suivi masse blanche >>

- **Cas d'utilisation << Suivi ingrédients >> pour l'administrateur**

Sommaire	
Titre	Suivi ingrédients.
Résumé	Afficher le suivi ingrédients (arôme et fruit) par période ou autres critères selon la volonté de l'administrateur.
Acteurs	Administrateur
Description des scénarios	
Pré condition	Etre connecté et être administrateur.

Scénario nominal	1. L'administrateur demande à afficher le suivi ingrédients (arôme ou fruit) selon les critères choisis. 2. Le système affiche le suivi ingrédients (arôme ou fruit) selon les critères choisis par l'administrateur (suivi mensuel, annuel, à temps réel...des pertes).
Alternative	/
Arrêt	Lorsqu'il passe à un autre suivi.
Post conditions	Affichage des statistiques dans l'interface adéquate.

Tableau 25: Description du cas d'utilisation << Suivi ingrédients >>

- **Cas d'utilisation << Suivi emballage >> pour l'administrateur**

Sommaire	
Titre	Suivi emballage.
Résumé	Afficher le suivi emballage par période ou autres critères selon la volonté de l'administrateur.
Acteurs	Administrateur
Description des scénarios	
Pré condition	Etre connecté et être administrateur.
Scénario nominal	1. L'administrateur demande à afficher le suivi masse blanche selon les critères choisis. 2. Le système affiche le suivi masse blanche selon les critères choisis par l'administrateur (suivi mensuel, annuel, à temps réel...des pertes)
Alternative	/
Arrêt	Lorsqu'il passe à un autre suivi.
Post conditions	Affichage des statistiques dans l'interface adéquate.

Tableau 26: Description du cas d'utilisation << Suivi emballage >>

- **Diagramme de séquence du cas d'utilisation « Saisir informations écrémage »**

Lorsque l'opérateur ligne envoi une demande de saisie des informations écrémage, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

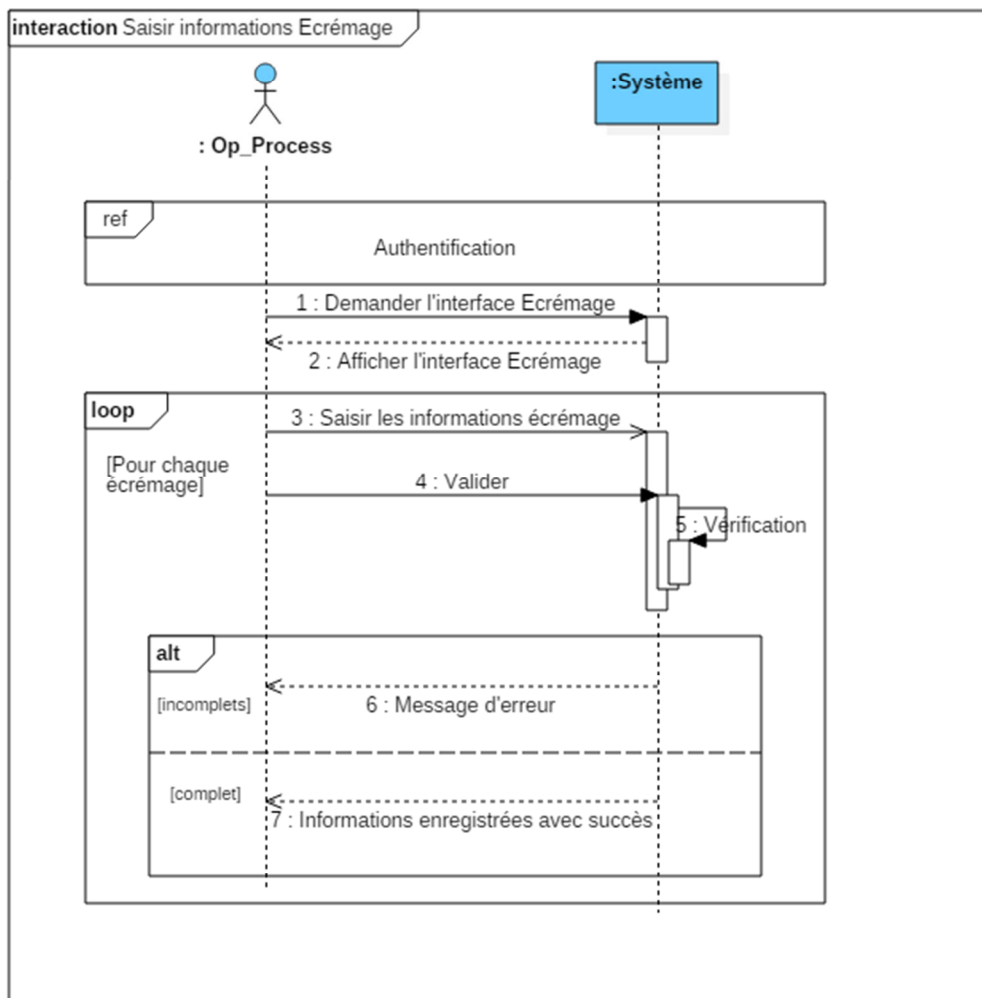


Figure 37: Diagramme de séquence du cas d'utilisation « Saisir informations écrémage »

- **Diagramme de séquence du cas d'utilisation « Saisir propriétés Masse Blanche »**

Lorsque l'opérateur ligne envoi une demande de saisie des propriétés masse blanche, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

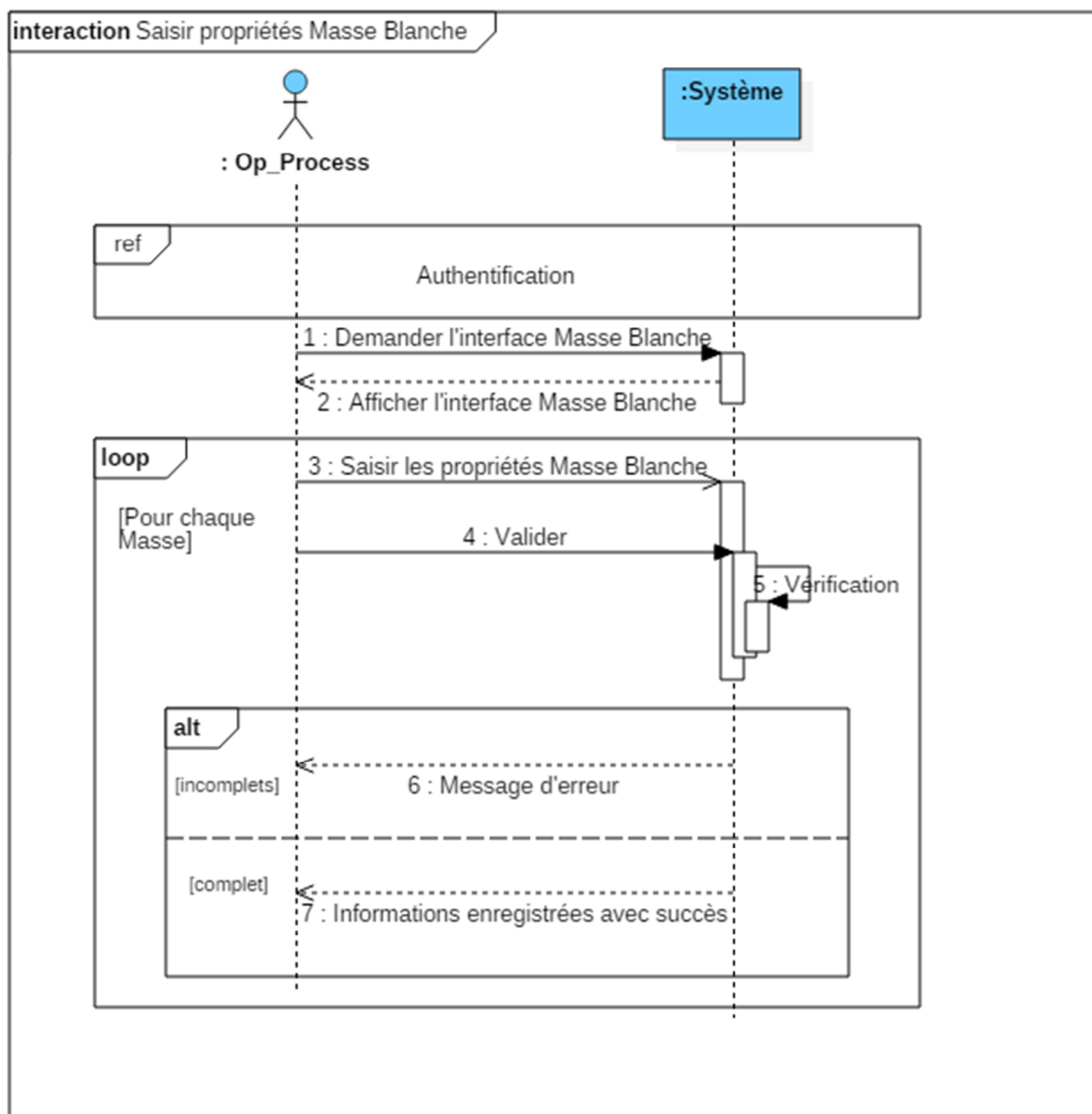


Figure 38: Diagramme de séquence du cas d'utilisation « Saisir propriétés masse blanche »

- Diagramme de séquence du cas d'utilisation « Renseigner fiche démarrage ligne »

Lorsque l'opérateur ligne envoie une demande de renseigner la fiche démarrage ligne, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

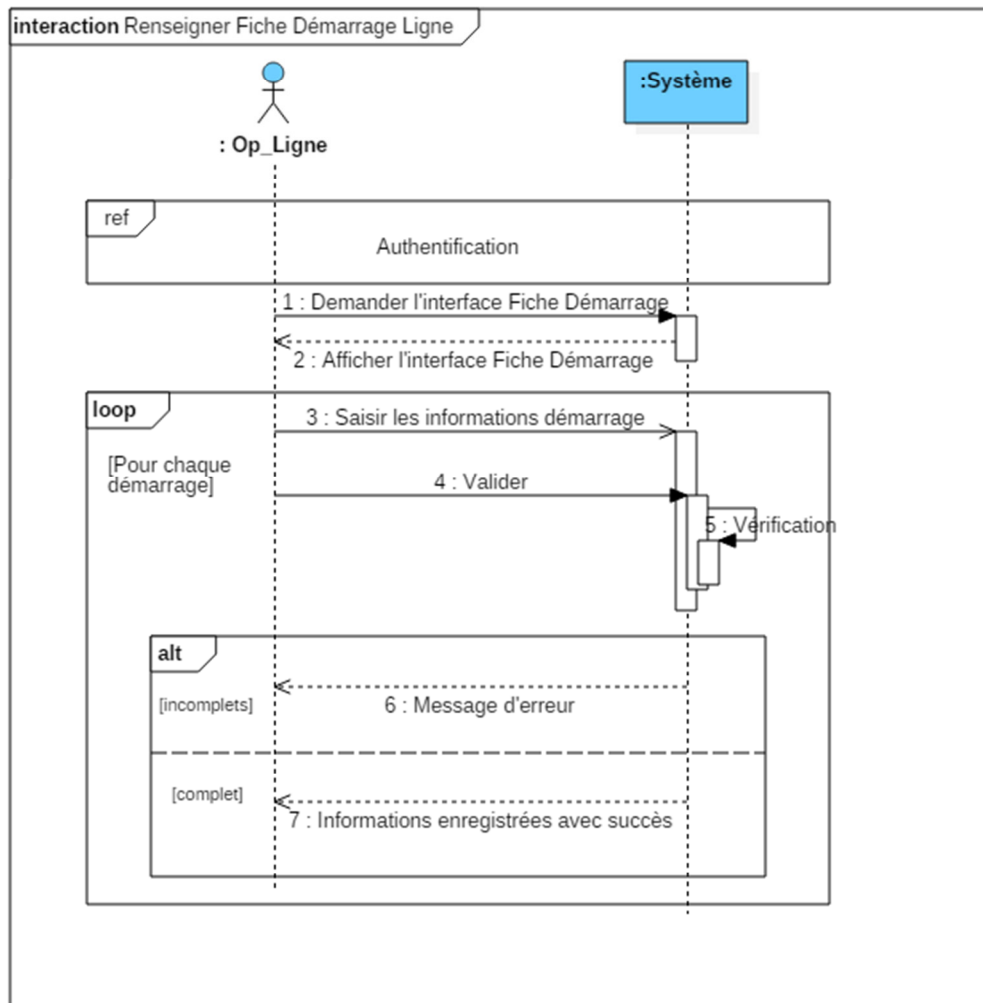


Figure 39: Diagramme de séquence du cas d'utilisation « Renseigner fiche démarrage ligne »

- **Diagramme de séquence du cas d'utilisation « Saisir control arôme »**

Lorsque l'opérateur ligne envoie une demande de saisie du control arôme, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

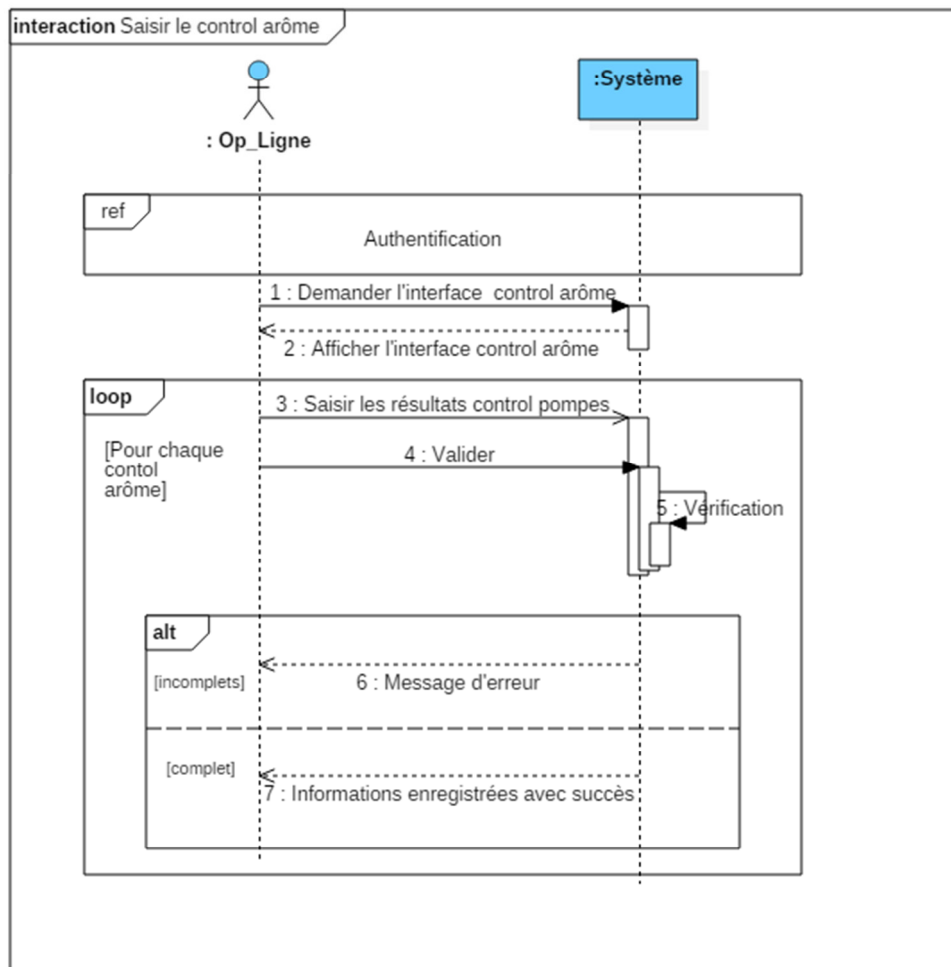


Figure 40: Diagramme de séquence du cas d'utilisation « Saisir control arôme »

- **Diagramme de séquence du cas d'utilisation « Saisir control emballage »**

Lorsque l'opérateur ligne envoie une demande de saisie du control emballage, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

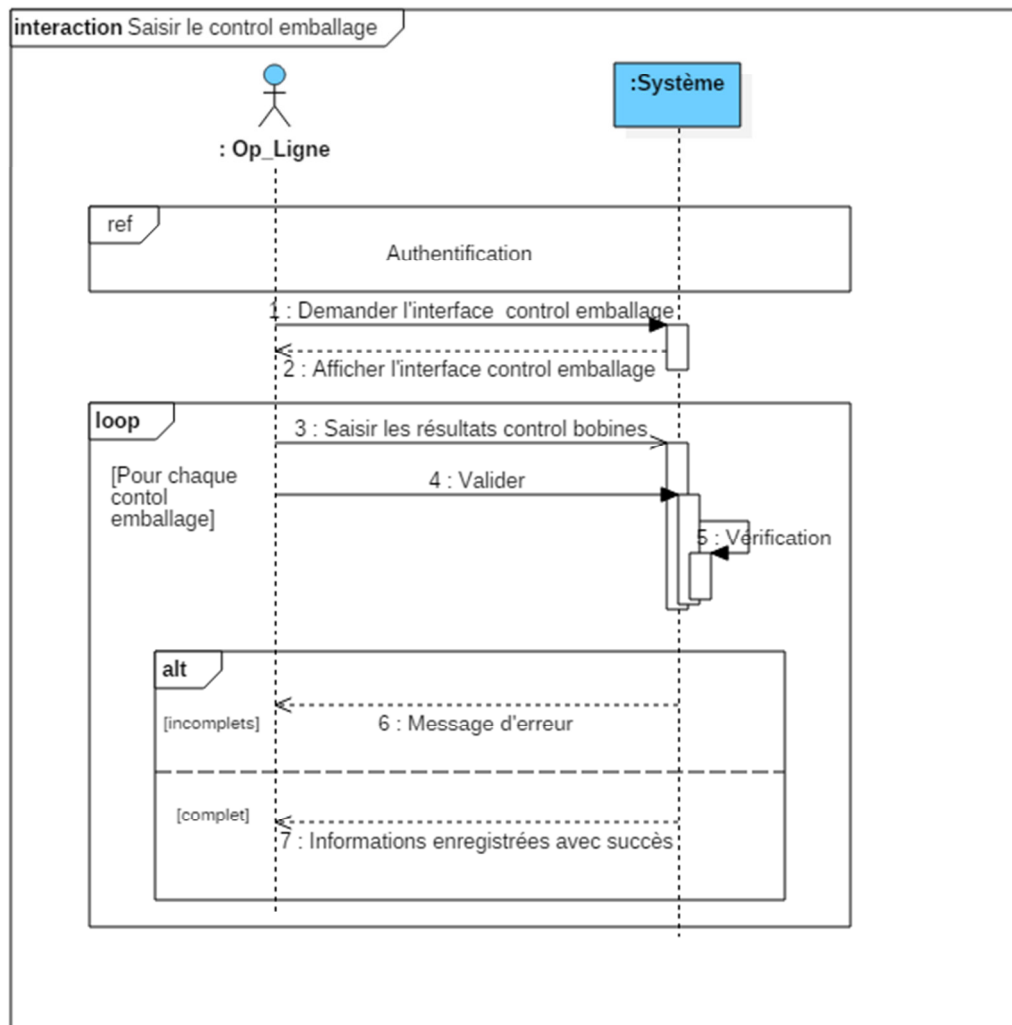


Figure 41: Diagramme de séquence du cas d'utilisation « Saisir control emballage »

- **Diagramme de séquence du cas d'utilisation « Saisir pertes fruit »**

Lorsque l'opérateur ligne envoie une demande de saisie des pertes fruit, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

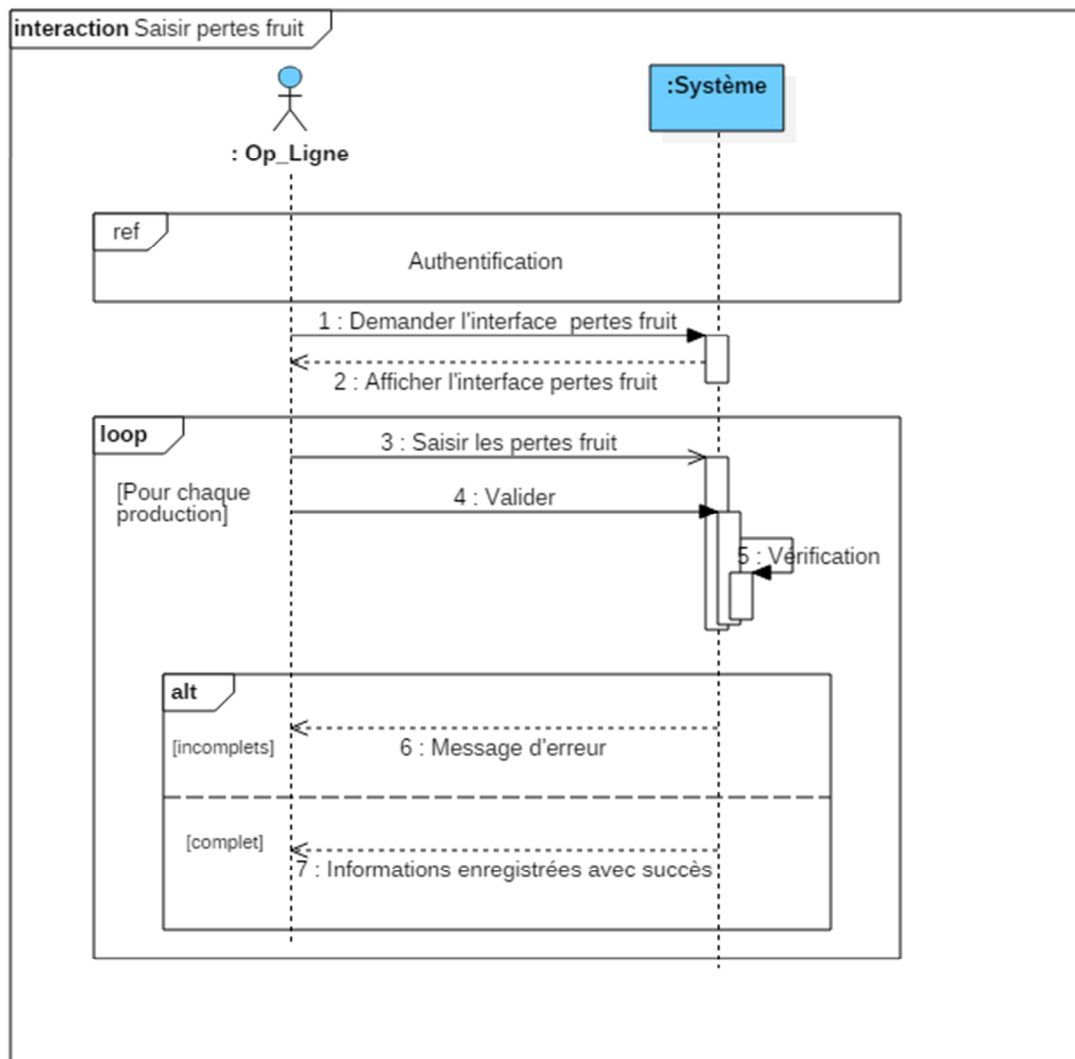


Figure 42: Diagramme de séquence du cas d'utilisation « Saisir pertes fruit »

- **Diagramme de séquence du cas d'utilisation « Saisir pertes emballage »**

Lorsque l'opérateur ligne envoie une demande de saisie des pertes emballage, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

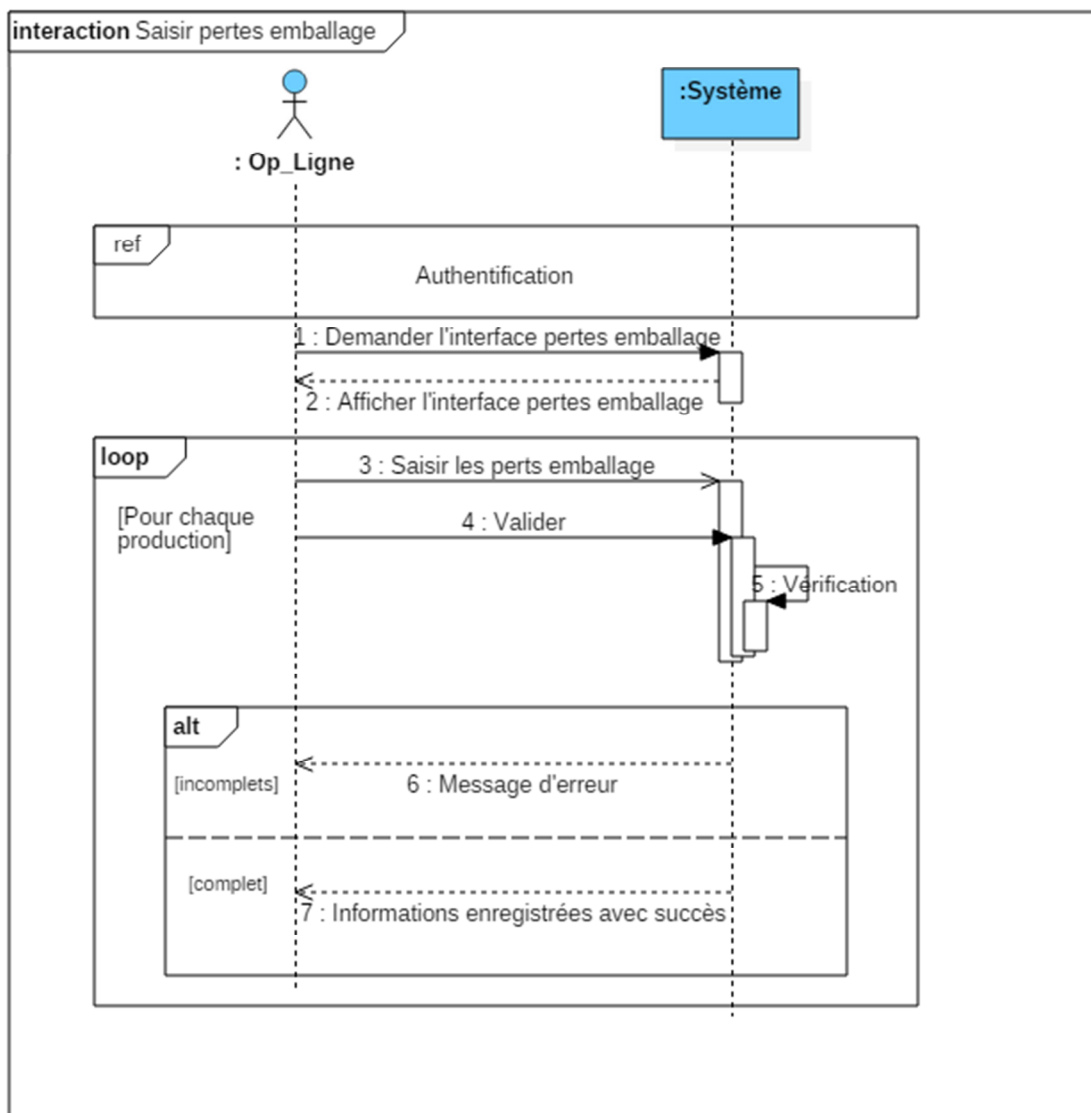


Figure 43: Diagramme de séquence du cas d'utilisation « Saisir pertes emballage »

- Diagramme de séquence du cas d'utilisation « Saisir pertes arôme »

Lorsque l'opérateur ligne envoie une demande de saisie des pertes arôme, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

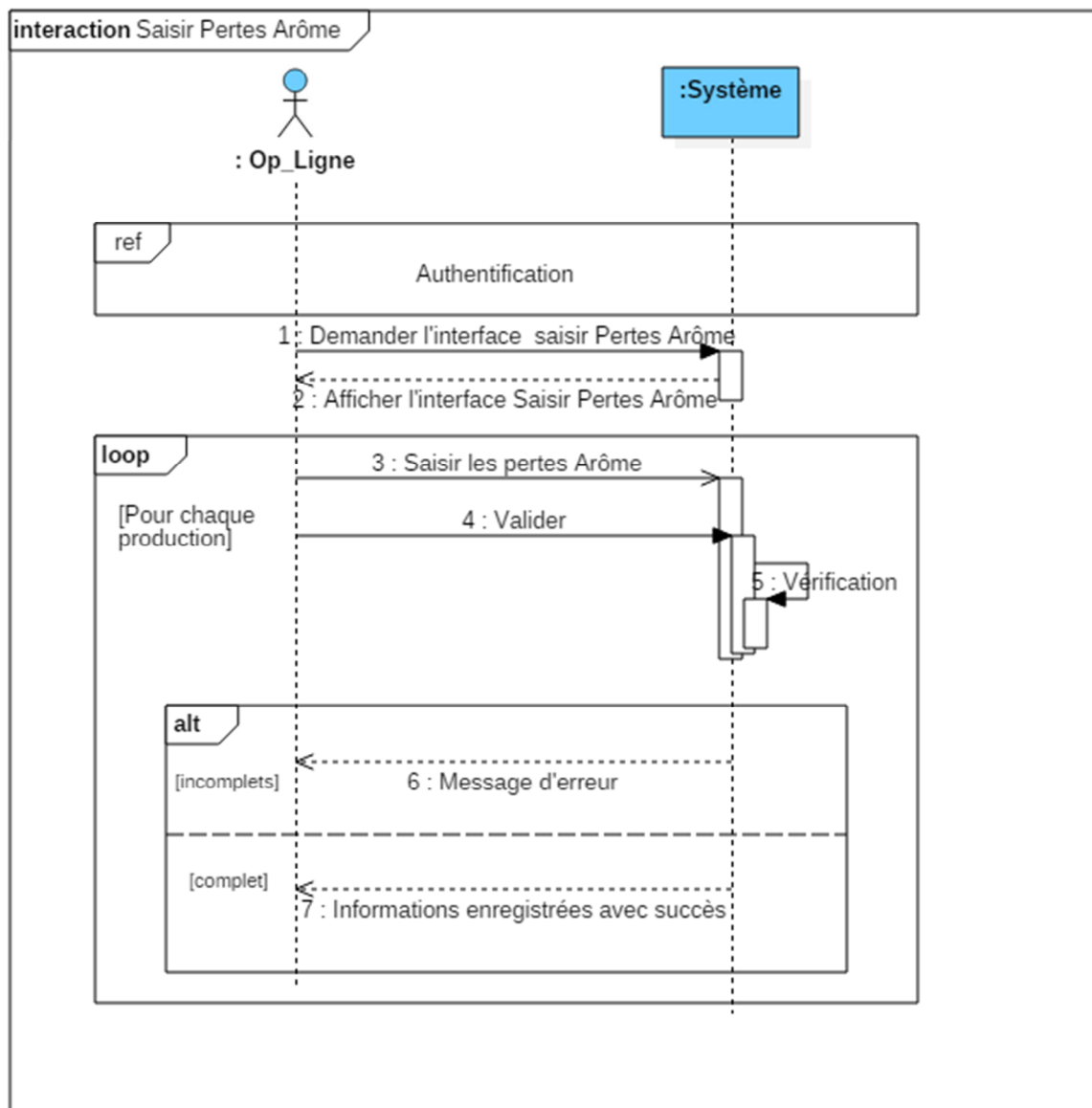


Figure 44: Diagramme de séquence du cas d'utilisation « Saisir pertes arôme »

- Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur »

Lorsque l'administrateur envoie une demande d'ajout d'un utilisateur, le système lui répond par l'affichage d'un formulaire qui sera validé après remplissage.

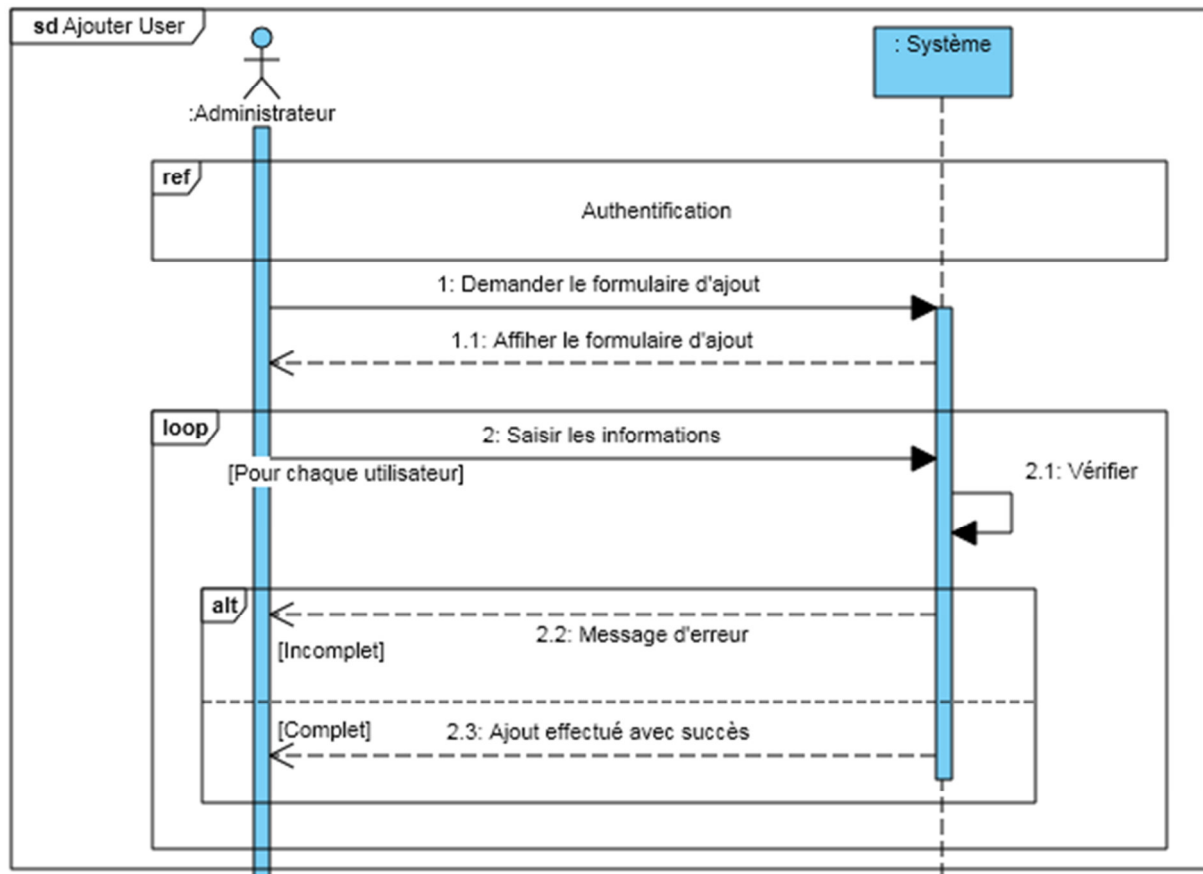


Figure 45: Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur »

- **Diagramme séquence du cas d'utilisation « modifier un utilisateur »**

Lorsque l'administrateur envoie une demande de modification d'un utilisateur en le sélectionnant, le système lui répond en affichant un formulaire afin d'apporter les modifications souhaitées.

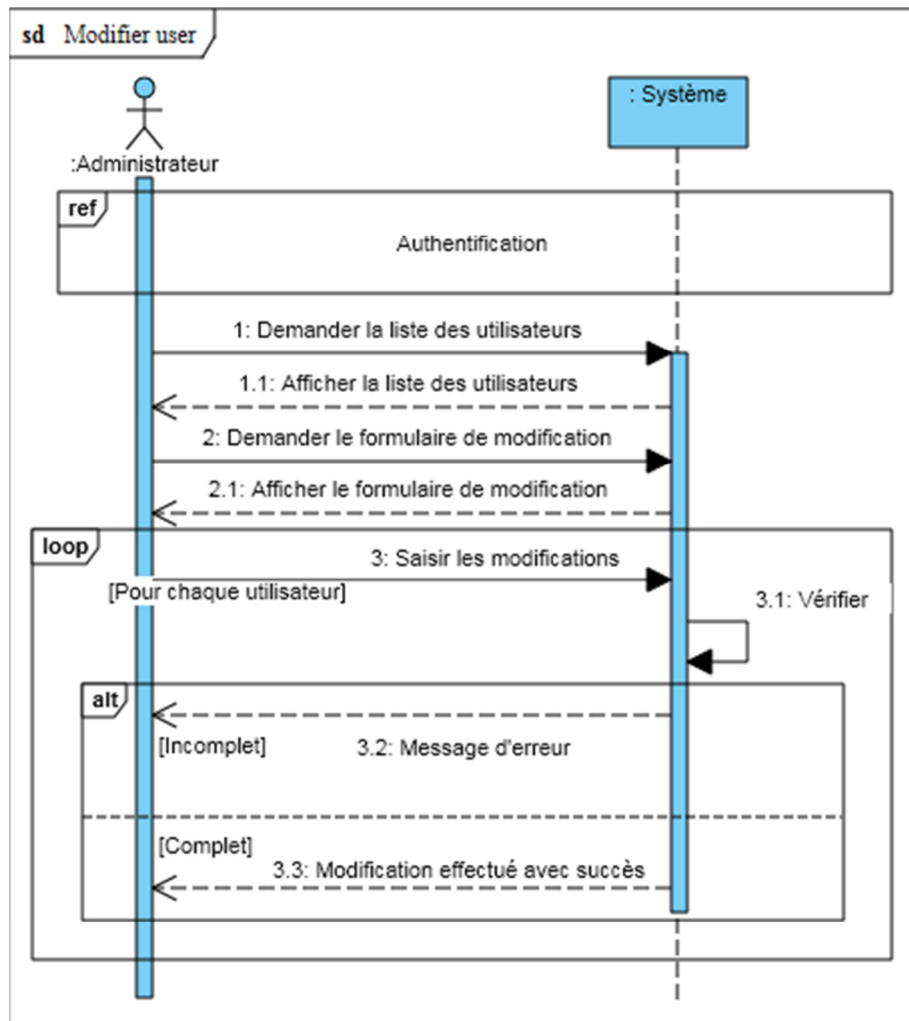


Figure 46: Diagramme séquence du cas d'utilisation « Modifier un utilisateur »

- **Diagramme de séquence du cas d'utilisation : « Supprimer un utilisateur »**

Après l'envoi de la demande de la suppression par l'administrateur, le système va afficher la liste des utilisateurs. Une fois que l'utilisateur à supprimer est choisi, la suppression sera effectuée.

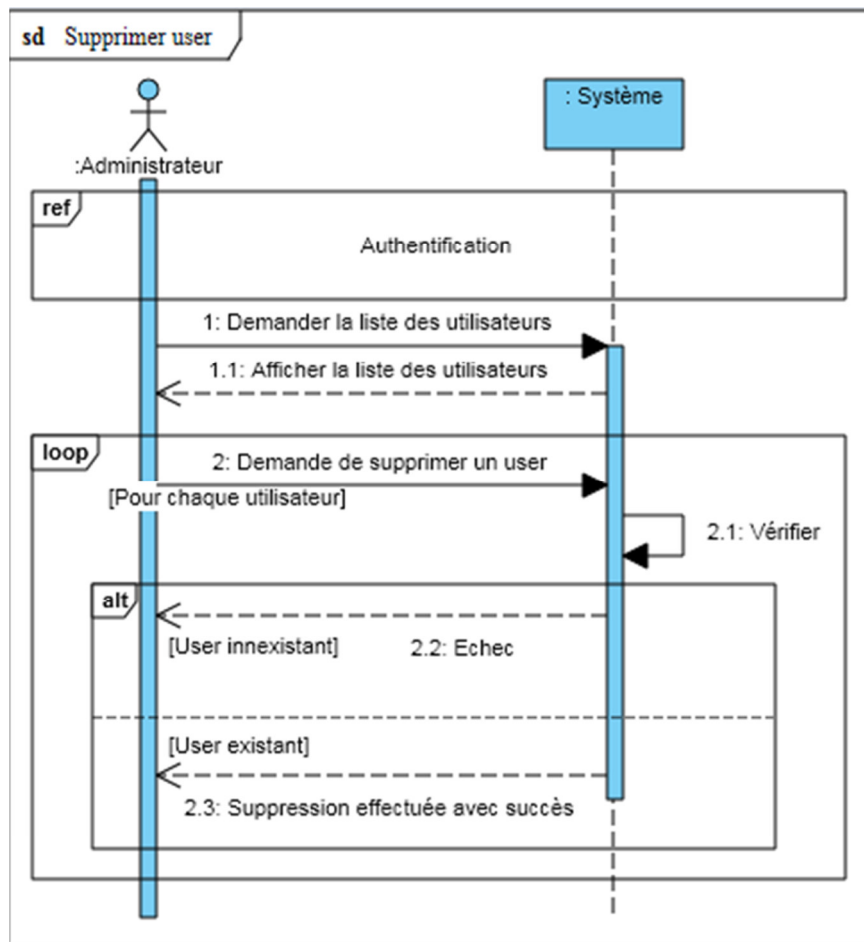


Figure 47: Diagramme séquence du cas d'utilisation « Supprimer un utilisateur »

- **Diagramme de séquence du cas d'utilisation «Suivi lait»**

Après l'envoi de la demande du suivi lait par l'administrateur, le système va afficher le suivi lait avec les actions possibles à effectuer (suivi à temps réel, par date...).

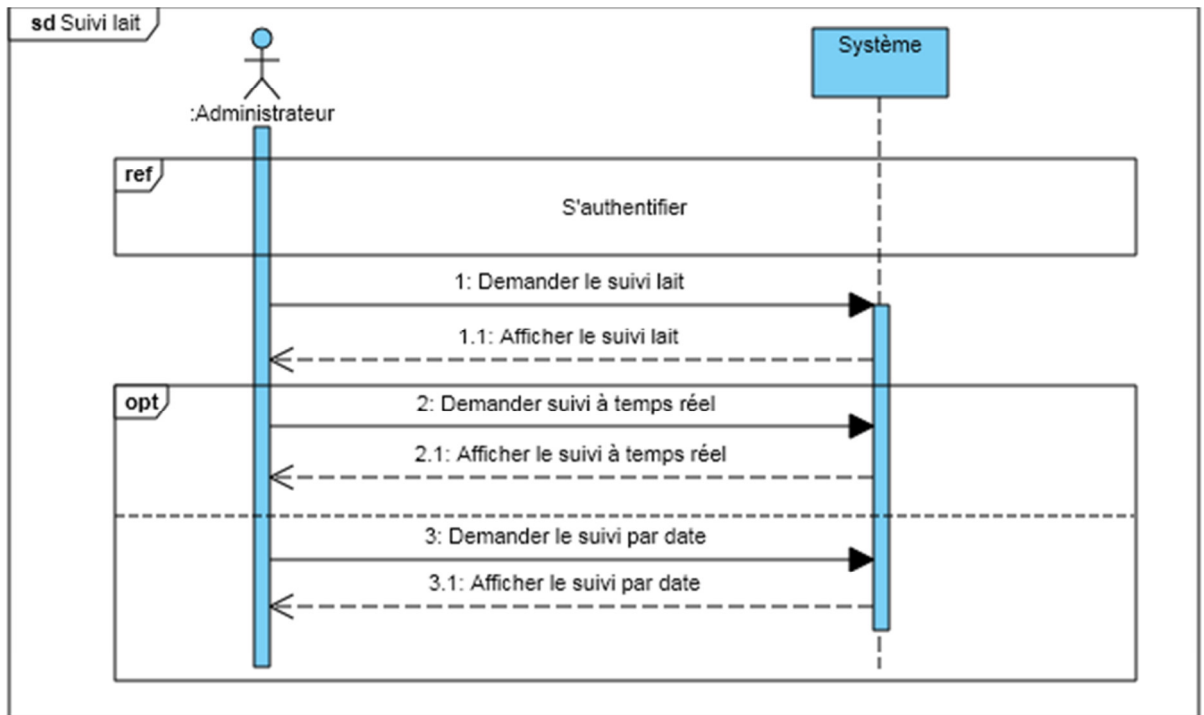


Figure 48: Diagramme séquence du cas d'utilisation « Suivi lait »

Note : Les mêmes étapes seront suivies pour le diagramme de séquence de suivi Masse Blanche.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

Bibliographie et Webographie

[1] Conception et réalisation d'un système d'information sous réseau Intranet pour la gestion de scolarité de la Faculté des Sciences Exactes. Mémoire de fin d'étude- *Promotion 2009*- Université Abderahmane Mira-Béjaïa.

[2] Laurant AUDIBERT. '*UML 2 - De l'apprentissage à la pratique*'. Site web.
URL : <https://laurent-audibert.developpez.com/Cours-UML/?page=mise-en-oeuvre-uml>. Publié le 31 octobre 2006 - Mis à jour le 12 janvier 2009. Date de consultation : 13/09/2020.

[3] P. ROQUES et F.VALLÉE. Les cahiers du programmeur UM2 modélisé une application web.

[4] Laurent Piechocki. Uml en français. Internet. <http://uml.free.fr/>. visité le 13/09/2020

[5] J.CONALLEN. Concevoir des applications web avec UML. Eyrolles, 2000.

[6] Pascal ROQUES Franck VALLEE, UML2 en action,Eyrolles 4 eme edition,2007.

[7] L.Audibert. UML2 De l'apprentissage à la pratique.

[8] Pascal. Roque, Frank. Eyollée « UML en action de l'analyse a la conception en java ».

[9] O. Sigaud. Introduction à la modélisation orientée objets.

[10] Roy GILLES. '*UML2 en action, de l'analyse des besoins en action*'. Presses de l'Université de Québec, 2009, première édition.

[11] Gilles et Roy. UML2 modéliser une application web. EYROLLES. Paris, 4ème édition.2008.

[12] Philippe RIGAUX. Cours de bases de données. juin 2001.

[13] <https://mikarber.developpez.com/tutoriels/java/introduction-javafx/> publié le 01/04/2015 consulté le 13/09/2020

[14] <https://www.ionos.fr/digitalguide/serveur/outils/tutoriel-xampp-creer-un-serveur-de-test-local/> modifié le 06/09/2019 consulté le 13/09/2020

[15] Télécharger Cours. Réseaux & Télécoms. [Http://www.TelechargerCours.com](http://www.TelechargerCours.com), 2012.

Résumé

L'objectif de ce travail est de mettre en place un système d'information automatisé opérationnel sous un réseau local pour le suivi des pertes de production à temps réel pour l'entreprise DANONE DJURDJURA. Notre produit logiciel permet de faciliter les tâches des opérateurs de la saisie des pertes et permet de minimiser l'utilisation papier et à l'administrateur d'avoir le suivi des pertes de production précis et global et prendre des décisions concernant le objectifs de performance selon les résultats. Pour réaliser cette application, nous avons utilisé le langage UML pour la modélisation objet, JAVA comme un langage de programmation et MariaDB comme SGBD.

Mots clés: MariaDB, DANONE, LAN, SPP, SQL, architecture 3 tiers, XAMPP.

Abstract

The objective of this work is to set up an automated information system operational under a local network for the monitoring of production losses in real time for the company DANONE DJURDJURA. Our software product makes it easier for operators to enter waste and helps to minimize paper use and allows the administrator to have precise and comprehensive production loss monitoring and make decisions regarding performance objectives according to the results. To realize this application, we used UML for object modeling, JAVA as a programming language and MariaDB as SGBD.

Key words: MariaDB, DANONE, LAN, SPP, SQL, architecture 3 tiers, XAMPP.