

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



جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Université A.Mira de Béjaïa
Faculté des sciences Exactes
Département d'Informatique

MEMOIRE DE MASTER PROFESSIONNEL

En
Informatique
Option
Génie logiciel

Conception et réalisation d'une application web de
« Gestion de production »
Cas d'étude : Entreprise Cevital

Réalisé par :

Mlle Liza BOUROUINA

Mlle Hanane OURABAH

Soutenu le 01 juillet 2024, devant le jury composé de:

Présidente : Mme KHALED Hayette

Examinatrice : Mme GASMI Badrina

Encadrante : Mme GHANEM Souhila

Co-encadreur : Mr CHIBOUTI Younes

Promotion 2023/2024

REMERCIEMENT

Ce travail n'aurait jamais vu le jour sans l'aide précieuse et le soutien de plusieurs personnes. Nous tenons à profiter de cet espace pour leur exprimer notre gratitude.

En premier lieu, nous adressons nos sincères remerciements à notre encadrant, Mr. Chibouti Younes, ainsi qu'à Mme. Ghanem Souhila. Leur investissement, leurs conseils avisés et la qualité de leur suivi ont été essentiels à la réalisation de ce travail.

Nous tenons à remercier les membres du jury qui nous ont honorés de leur présence, pour leur lecture attentive de ce mémoire, ainsi que pour les remarques qu'ils nous adresseront lors de cette soutenance afin d'améliorer notre travail.

Nous remercions sincèrement les membres du personnel du Cevital qui nous ont facilité toutes les procédures de recueil d'information.

Nous tenons à remercier vivement nos amis et particulièrement nos familles qui ont toujours su nous soutenir invariablement et nous encourager sans relâche tout au long de cette année. Enfin, que tous ceux qui, de près ou de loin, ont participé à la réalisation de ce travail, trouvent ici l'expression de notre sincère gratitude.

Dédicace

”

Je dédie ce travail à mes parents, pour leur amour, leur soutien inconditionnel et leurs encouragements constants qui ont été ma source d'inspiration. À mon frère et mes sœurs, pour leur complicité et leur présence rassurante. À toute ma famille, pour leur affection et leurs précieux conseils. À mes amis, pour leur amitié, leur soutien moral et les moments de joie partagés tout au long de ce parcours. Merci à tous de croire en moi et de m'avoir accompagnée dans cette aventure

”

Liza

Dédicace

”

Je dédie ce travail

À mon très cher père, qui a mis en moi tant d'espoir et de confiance,

À ma très adorée mère, qui a toujours été là pour moi avec un soutien inconditionnel,

À mon très cher grand frère et à ma chère petite sœur, qui ont toujours cru en moi,

À mon cher grand-père et à ma chère grand-mère,

À mes oncles, tantes, cousins, cousines,

À tous mes amis pour leur amitié, leurs encouragements, leur soutien,

À Ma binôme Liza,

À tous ceux qui m'ont aidée et dont je n'ai pas cité le nom.

”

Hanane

Listes des Acronymes

SCRUD : Search, Create, Read, Update, Delete

CRUD : Create, Read, Update, Delete

UML : Unified Modeling Language.

UP : Unified Process

PHP: Hypertext Preprocessor

HTML: Hypertext Markup Language

CSS : Cascading Style Sheets

MVC : Modèle-Vue-Contrôleur

GPAO : Gestion de Production Assistée Ordinateur

Table des matières

REMERCIEMENT	1
Dédicace	1
Listes des Acronymes	1
Introduction générale	1
CHAPITRE 1 : Contexte d'étude et recueil des besoins	3
1.1 Introduction	4
1.2 Présentation de Cevital	4
1.3 Historique	4
1.4 L'organisme agroalimentaire de Bejaia	5
1.5 Situation géographique de l'organisme agroalimentaire de Bejaia	5
1.6 Propriétés et Installations de CEVITAL agroalimentaire	6
1.7 Structure organisationnelle de Cevital Agroalimentaire	7
1.8 Présentation du système d'information de Cevital Agroalimentaire	7
1.9 Discussion	8
1.9.1 Problématique	8
1.9.2 Solution envisagée	8
1.10 Conclusion	9
CHAPITRE 2 : Méthodologie de conception et spécification des besoins	10
2.1 Introduction	11
2.2 Méthodologie de conception	11
2.2.1 Méthodes agiles	11
2.2.2 Langage de modélisation UML	14

2.3 Architecture MVC	15
2.4 Pilotage du projet avec scrum	16
2.4.1 Définition des entités impliquées dans notre projet	16
2.4.2 Identification des besoins fonctionnels et non fonctionnels	17
2.4.3 Rôle et user stories	19
2.4.4 Identification des acteurs	20
2.4.5 Identification des cas d'utilisation	20
2.4.6 Diagramme de contexte	21
2.4.7 Diagramme de cas d'utilisation globale	22
2.4.8 Planification des sprints	23
2.4.9 Backlog-Product (carnet de produit)	24
2.4.10 Charte graphique	26
2.5 Conclusion	27
CHAPITRE 3 : Application de la méthode SCRUM sur le Projet	28
3.1 Introduction	29
3.2 Etude du Sprint 1	29
3.2.1 Diagramme de cas d'utilisation du sprint 1	29
3.2.2 Description textuelle du cas « S'authentifier »	30
3.2.3 Description textuelle du cas « Gestion des comptes utilisateurs »	31
3.2.4 Description textuelle du cas « Gestions des rôles »	32
3.2.5 Diagramme de séquence du cas « s'authentifier »	33
3.2.6 Diagramme de séquence du cas « modifier un compte utilisateur »	34
3.2.7 Diagramme de séquence du cas « ajouter un rôle »	35
3.3 Etude du sprint 2	35
3.3.1 Diagramme de cas d'utilisation du sprint 2	36
3.3.2 Description textuelle du cas « Gérer les ateliers »	37

3.3.3	Description textuelle du cas « Gérer les lignes »	37
3.3.4	Description textuelle du cas « Gérer les catalogues d'arrêts »	38
3.3.5	Description textuelle du cas « Gérer les articles »	39
3.3.6	Description textuelle du cas « Gérer les plannings »	40
3.3.7	Diagramme de séquence du cas : « Ajouter un atelier »	41
3.3.8	Diagramme de séquence du cas : « modifier une ligne »	42
3.3.9	Diagramme de séquence du cas « rechercher un article »	43
3.3.10	Diagramme de séquence du cas « supprimer un catalogue »	44
3.3.11	Diagramme de séquence du cas « supprimer un planning »	45
3.4	Etude du sprint 3	46
3.4.1	Diagramme des cas d'utilisation du sprint 3	46
3.4.2	Description textuelle du cas : « Gérer les recettes »	47
3.4.3	Description textuelle du cas « Gérer les cadences »	48
3.4.4	Diagramme de séquence du cas « ajouter une recette »	49
3.4.5	Diagramme de séquence « modifier une cadence »	50
3.5	Etude du sprint 4	51
3.5.1	Diagramme des cas d'utilisation du sprint 4	51
3.5.2	Description textuelle du cas « Gérer les productions »	52
3.5.3	Description textuelle du cas « Gérer les arrêts de production »	53
3.5.4	Description textuelle du cas « Imprimer un rapport »	54
3.5.5	Diagramme de séquence du cas « Ajouter une production »	55
3.5.6	Diagramme de séquence du cas « Ajouter un arrêt »	56
3.5.7	Diagramme de séquence du cas « Imprimer un rapport »	57
3.6	Conclusion	58
CHAPITRE 4 : Conception de la base de données		59
4.1	Introduction	60

4.2	Règles de gestion	60
4.3	Digramme de classe	61
4.4	Dictionnaire de données	63
4.5	Schéma relationnel	67
4.5.1	Règles de passage du diagramme de classe au modèle relationnel	67
4.5.2	Le model relationnel des données	68
4.6	Conclusion	69
CHAPITRE 5 : Réalisation et implémentation		70
5.1	Introduction	71
5.2	Outils de développement	71
5.2.1	Environnement machine	71
5.2.2	Environnement logiciel	71
5.2.3	Environnement de programmation	72
5.2.4	Frameworks utilisés	74
5.3	Projection de quelques interfaces de notre application	75
5.3.1	Authentification	75
5.3.2	Interface de tableaux de bord	75
5.3.3	Gestion des comptes utilisateurs et attribution des rôles	76
5.3.4	Gestion des ateliers, lignes, catalogues des arrêts	78
5.3.5	Gestion des articles	80
5.3.6	Interface de gestion des recettes	81
5.3.7	Interface de gestion des plannings	82
5.3.8	Interface de gestion des productions	82
5.3.9	Interface d'établir et d'imprimer un rapport	85
5.3.10	Une démonstration de la responsivité	86
5.4	Conclusion	87

Conclusion générale et perspectives	88
Références	90

Listes des Figures

<i>Figure 1:Complexe Cevital Agroalimentaire arrière port de Bejaia</i>	5
<i>Figure 2:Organigramme général du groupe Cevital Agroalimentaire</i>	7
<i>Figure 3:Organigramme DSI de Cevital Agroalimentaire</i>	8
<i>Figure 4:Cycle de vie de SCRUM</i>	12
<i>Figure 5:Differents diagrammes UML [7]</i>	14
<i>Figure 6:Architecture MVC [8]</i>	15
<i>Figure 7:Diagramme de contexte</i>	21
<i>Figure 8:Diagramme de cas d'utilisation globale</i>	23
<i>Figure 9:Logo</i>	26
<i>Figure 10:Diagramme de cas d'utilisation du sprint 1</i>	29
<i>Figure 11:Diagramme de séquence du cas : « S'authentifier »</i>	33
<i>Figure 12:Diagramme de séquence du cas « modifier un compte utilisateur »</i>	34
<i>Figure 13:Diagramme de séquence du cas « ajouter un rôle »</i>	35
<i>Figure 14:Diagramme de cas d'utilisation du sprint 2</i>	36
<i>Figure 15:Diagramme de séquence du cas : « Ajouter un atelier »</i>	41
<i>Figure 16:Diagramme de séquence du cas : « modifier une ligne »</i>	42
<i>Figure 17:Diagramme de séquence du cas « rechercher un article »</i>	43
<i>Figure 18:Diagramme de séquence du cas « supprimer catalogue »</i>	44
<i>Figure 19:Diagramme de séquence du cas « supprimer un planning »</i>	45
<i>Figure 20:Diagramme du cas d'utilisation du sprint 3</i>	46
<i>Figure 21:Diagramme de séquence du cas « ajouter une recette »</i>	49
<i>Figure 22:Diagramme de séquence « modifier une cadence »</i>	50
<i>Figure 23:Diagramme des cas d'utilisation du sprint 4</i>	51
<i>Figure 24:Diagramme de séquence du cas « Ajouter une production »</i>	55
<i>Figure 25:Diagramme de séquence du cas « Ajouter un arrêt »</i>	56
<i>Figure 26:Diagramme de séquence du cas « Imprimer un rapport »</i>	57
<i>Figure 27:Diagramme de Class</i>	62
<i>Figure 28:Interface d'authentification</i>	75
<i>Figure 29:Tableau de bord administrateur</i>	75
<i>Figure 30:tableau de bord du responsable</i>	76
<i>Figure 31:Tableau de bord de l'operateur</i>	76
<i>Figure 32:Interface de création d'un utilisateur</i>	77

<i>Figure 33:Liste des Utilisateurs</i>	77
<i>Figure 34:Interface de modification d'un utilisateur</i>	78
<i>Figure 35:Interface de gestion des ateliers</i>	78
<i>Figure 36:Interface de gestion des catalogues d'arrêts</i>	79
<i>Figure 37:Interface de gestion des ligne</i>	79
<i>Figure 38:Liste des articles</i>	80
<i>Figure 39:Interface d'ajout d'article</i>	80
<i>Figure 40:Liste des recettes</i>	81
<i>Figure 41:Interface d'ajout d'une recette</i>	81
<i>Figure 42:Interface de gestion des plannings</i>	82
<i>Figure 43:Liste des productions</i>	82
<i>Figure 44:Interface d'ajout d'une production</i>	83
<i>Figure 45:Interface de show détails d'une production</i>	83
<i>Figure 46:Interface d'ajout d'arrêt</i>	84
<i>Figure 47:Liste des arrêts déclarés</i>	84
<i>Figure 48:Interface d'établir un rapport</i>	85
<i>Figure 49:Interface d'impression d'un rapport</i>	85
<i>Figure 50:Vue de l'application sur Smartphone</i>	86
<i>Figure 51:Vue de l'application sur une Tablette</i>	86
<i>Figure 52:Vue de l'application sur une Tablette</i>	87

Liste des Tableaux

<i>Tableau 1:Equipe et roles</i>	19
<i>Tableau 2:Identification des cas d'utilisation</i>	20
<i>Tableau 3:Planification des sprints</i>	24
<i>Tableau 4:Product backlog du système</i>	26
<i>Tableau 5:Description textuelle du cas : « s'authentifier »</i>	30
<i>Tableau 6:Description textuelle du cas « Gestion des comptes utilisateurs »</i>	31
<i>Tableau 7:Description textuelle du cas « Gestions des rôles et permissions »</i>	32
<i>Tableau 8:Description textuelle du cas « Gérer les ateliers »</i>	37
<i>Tableau 9:Description textuelle du cas « Gérer les lignes »</i>	38
<i>Tableau 10:Description textuelle du cas « Gérer les catalogues d'arrêts »</i>	39
<i>Tableau 11:Description textuelle du cas « Gérer les articles »</i>	39
<i>Tableau 12:Description textuelle du cas « Gérer les plannings »</i>	40
<i>Tableau 13:Description textuelle du cas : « Gérer les recettes »</i>	47
<i>Tableau 14:Description textuelle du cas « Gérer les cadence »</i>	48
<i>Tableau 15:Description textuelle du cas d'utilisation « Gérer les production »</i>	52
<i>Tableau 16:Description textuelle du cas d'utilisation « Gérer les arrêts de production »</i>	53
<i>Tableau 17:Description textuelle du cas d'utilisation « Imprimer un rapport »</i>	54
<i>Tableau 18:Dictionnaire des données</i>	67

Introduction générale

La gestion de production, dans le contexte moderne, bénéficie grandement de l'intégration des technologies de l'information et de la communication, offrant des solutions innovantes pour optimiser les processus industriels. Notre projet de fin de cycle de master s'inscrit dans cette dynamique, avec la conception et la réalisation d'une application web dédiée à la gestion de production pour Cevital Agroalimentaire de Béjaïa.

Au cours de notre stage au sein de Cevital Agroalimentaire, nous avons eu l'opportunité de recueillir des informations sur les processus de gestion de la production. Après une analyse approfondie avec le responsable de la production, plusieurs lacunes majeures ont été identifiées dans l'application de bureau actuelle. Ces lacunes comprennent l'absence de calcul du Taux de Rendement Synthétique (TRS), limitant l'évaluation de la performance de la production. De plus, il manque une fonction d'impression des rapports de production, entravant la communication et la prise de décisions éclairées. Enfin, l'architecture non web rend l'application moins flexible et moins accessible pour les utilisateurs. En résumé, ces lacunes compromettent l'efficacité et la productivité de la gestion de la production.

Afin de corriger les lacunes de l'application actuelle, nous prévoyons d'ajouter des fonctionnalités telles que l'intégration du calcul du Taux de Rendement Synthétique (TRS) pour évaluer la performance en temps réel et identifier les domaines à améliorer. Nous envisageons également d'inclure une fonctionnalité d'impression de rapports personnalisés, avec des options pour les exporter au format PDF ou Excel. Pour améliorer l'accessibilité, nous migrerons l'application vers une plateforme web, garantissant une interface utilisateur intuitive et réactive accessible depuis n'importe quel appareil. Par ailleurs, des mesures de sécurité robustes seront mises en place pour protéger les données contre les accès non autorisés, assurant ainsi une gestion de production plus efficace et sécurisée.

Le projet vise à développer une application de gestion de production pour Cevital, en intégrant plusieurs aspects cruciaux de la chaîne de production. Cette application permettra la gestion complète des articles, des recettes de production, des ateliers et des arrêts, tout en

incluant la planification de la production. Elle intégrera également le suivi de la cadence de chaque ligne de production, la gestion du personnel, et la gestion de la consommation pour chaque production.

La suite de notre rapport est organisée comme suit : Le premier chapitre, intitulé « Contexte d'étude et recueil des besoins », introduit le projet. Nous y présentons l'organisme d'accueil, sa structure et ses missions, tout en fournissant une description détaillée de l'étude de l'existant et du contexte de travail. Ce chapitre aborde également la problématique et les objectifs du projet.

Le deuxième chapitre, « Méthodologie de conception et spécification des besoins », est consacré à la présentation de la méthode Scrum, du langage de modélisation UML, ainsi que de l'architecture MVC de notre système. Nous y détaillons également la spécification des besoins, en identifiant les acteurs et l'environnement général à l'aide d'un diagramme de contexte. Nous établissons également le diagramme de cas d'utilisation global, accompagné des user stories et du carnet de produit (backlog-product).

Dans le troisième chapitre, nous présentons l'étude des livrables selon les sprints planifiés. Nous y incluons le diagramme de cas d'utilisation, les descriptions textuelles et la réalisation des diagrammes d'interaction pour chaque sprint.

Le quatrième chapitre, « Conception de la base de données », est consacré à la présentation des règles de gestion, du diagramme de classes, du dictionnaire de données et du passage au relationnel.

Dans le cinquième chapitre, nous décrivons l'implémentation de cette application, en présentant les outils et les langages utilisés, ainsi que les différentes interfaces de notre application.

CHAPITRE 1 : Contexte d'étude et recueil des besoins

1.1 Introduction

Ce chapitre est un chapitre introductif dans lequel nous expliquons et décrivons le contexte et les objectifs de notre travail. En présentant dans un premier lieu l'organisme d'accueil (Cevital). En deuxième lieu, nous décrivons Cevital agroalimentaire, sa structure, sa situation géographique, ses propriétés et ses installations. En dernier lieu nous exposons en détail la problématique et les solutions proposées.

1.2 Présentation de Cevital

Le Groupe Cevital est un conglomérat (groupe diversifié) algérien opérant dans divers secteurs tels que l'industrie agroalimentaire, la distribution, la construction et les services. Il occupe une position de premier plan en tant que plus grand groupe privé en Algérie et se classe troisième en termes de chiffre d'affaires. Avec une main-d'œuvre de 18 000 employés [1], Cevital est un leader incontesté dans le secteur agroalimentaire du pays, englobant la production d'huiles végétales, de sucre, de margarines, de boissons, ainsi que la fabrication d'électroménager et de produits verriers. De plus, le groupe a étendu son influence à l'échelle internationale, notamment en Europe, où il a acquis des entreprises dans le secteur de l'électroménager et de la menuiserie.

1.3 Historique

L'histoire de Cevital remonte à 1971, lorsque son fondateur, Issad Rebrab, a créé une petite entreprise de construction en Algérie. Au fil des années, l'entreprise s'est diversifiée et a commencé à investir dans différents secteurs. En 1998, Cevital a été officiellement créée en tant que conglomérat industriel et a rapidement étendu ses activités à travers le pays. Au cours des deux dernières décennies, l'entreprise a continué de croître et d'acquérir de nouvelles filiales, renforçant ainsi sa position sur le marché [1].

1.4 L'organisme agroalimentaire de Bejaia

Créé en 1998 et implanté au sein du port de Bejaia, Cevital Agro-industrie dispose de plusieurs unités de production ultramodernes de sucre, corps gras, eau minérale, boissons et sauces. Elle couvre les besoins nationaux et permet de faire passer l'Algérie du stade d'importateur à celui d'exportateur pour les huiles, les margarines et le sucre. Ses produits se vendent dans plusieurs pays, notamment en Europe, au Maghreb, au Moyen Orient et en Afrique de l'Ouest.

1.5 Situation géographique de l'organisme agroalimentaire de Bejaia

Cevital Agro-industrie est implanté à l'arrière-port de Bejaia, à une distance de 200 mètres du quai et à 3 kilomètres au sud-ouest de la ville, à proximité des routes RN 26 et RN 9. Cette localisation stratégique confère à l'entreprise un avantage en termes de proximité économique. Le complexe s'étend sur une superficie de 45 000 mètres carrés, ce qui en fait le plus grand complexe privé en Algérie. Il dispose d'une capacité de stockage de 182 000 tonnes par an et d'un terminal de déchargement portuaire capable de recevoir jusqu'à 2000 tonnes par jour de matières premières. De plus, l'entreprise bénéficie d'un vaste réseau de distribution comprenant plus de 52 000 points de vente à travers le pays [2].

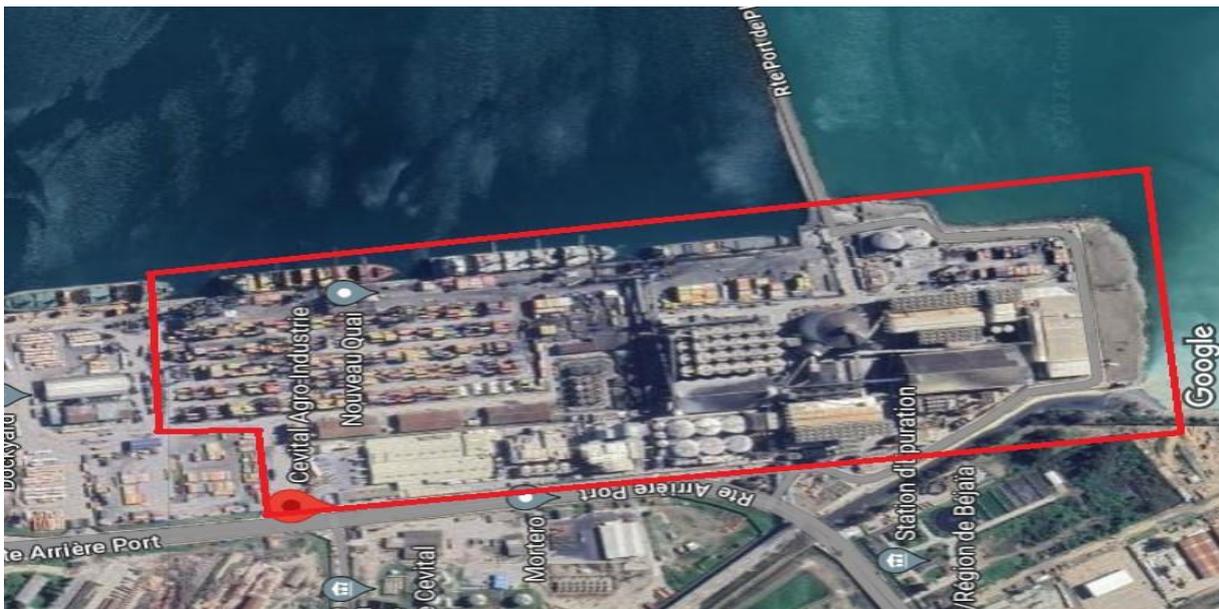


Figure 1: Complexe Cevital Agroalimentaire arrière port de Bejaia

1.6 Propriétés et Installations de CEVITAL agroalimentaire

CEVITAL agroalimentaire est une entreprise diversifiée qui détient un large éventail d'installations et de propriétés industrielles. Parmi ses actifs figurent notamment : [3]

- 2 raffineries de sucre 3000 et 3500 Tn
- 1 unité de sucre liquide et une unité de production de sucre roux
- 2 unités de conditionnement de sucre
- 1 raffinerie d'huile
- 1 unité de conditionnement d'huile
- 1 margarinerie
- 1 unité d'eau minérale et gazéifiée
- 1 unité de fabrication et de conditionnement de boissons fruitées et de production de conserves et confitures
- 1 unité de production des sauces
- 1 unité de fabrication de chaux calcinée et CO2
- Silos portuaires ainsi qu'un terminal de déchargement portuaire d'une capacité de 2000 tonnes/jour

1.7 Structure organisationnelle de Cevital Agroalimentaire

L'entreprise est divisée en différentes divisions, chacune étant responsable d'un domaine d'activité spécifique. Chaque division est dirigée par un responsable qui rend compte directement à la direction générale. Cette structure permet une gestion efficace des différentes activités de l'entreprise et favorise la prise de décision rapide.

L'organigramme ci-dessous illustre la répartition des responsabilités et des relations hiérarchiques au sein de CEVITAL [4].

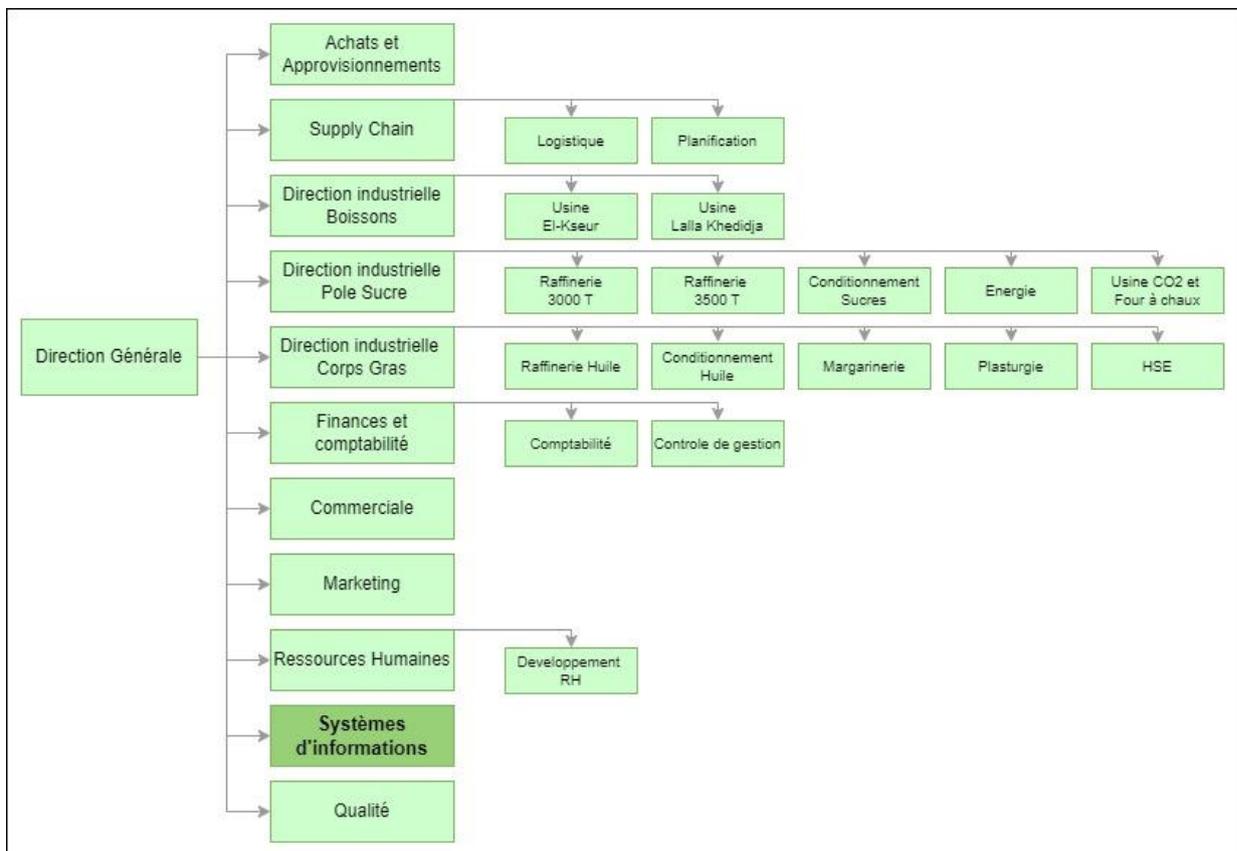


Figure 2: Organigramme général du groupe Cevital Agroalimentaire

1.8 Présentation du système d'information de Cevital Agroalimentaire

La Direction des Systèmes d'Information (DSI) est responsable de la mise en place des outils et des technologies de l'information visant à améliorer l'activité, la stratégie et la performance de l'entreprise. Elle assure la cohérence des moyens informatiques et de communication mis à la disposition des utilisateurs, garantissant leur maîtrise technique ainsi

que leur disponibilité et opérationnalité continues, tout en assurant la sécurité des systèmes [4].

La figure ci-dessous montre l'organigramme du système d'information.

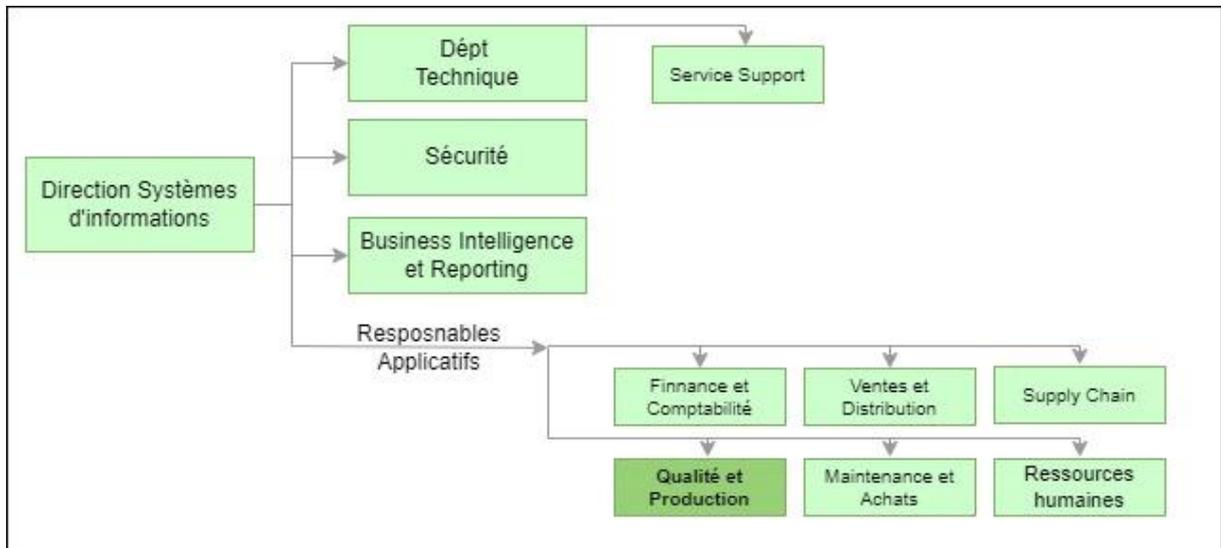


Figure 3: Organigramme DSI de Cevital Agroalimentaire

1.9 Discussion

Dans ce qui suit nous aborderons le contexte du projet en examinant les problématiques et en proposant des solutions

1.9.1 Problématique

Après une analyse approfondie avec le responsable de la production, plusieurs lacunes majeures ont été identifiées dans l'application de bureau actuelle. Ces lacunes comprennent l'absence de calcul du Taux de Rendement Synthétique (TRS), ce qui limite l'évaluation de la performance de la production. De plus, il manque une fonction d'impression des rapports de production, entravant la communication et la prise de décisions éclairées. Enfin, l'architecture non web rend l'application moins flexible et moins accessible pour les utilisateurs. En résumé, ces lacunes compromettent l'efficacité et la productivité de la gestion de la production.

1.9.2 Solution envisagée

Le projet vise à développer une application de gestion de production pour Cevital, une entreprise renommée, en intégrant plusieurs aspects cruciaux de la chaîne de production.

Cette application permettra la gestion complète des articles, des recettes de production, des ateliers et des arrêts, tout en incluant la planification de la production. Elle intégrera également le suivi de la cadence de chaque ligne de production, la gestion du personnel, et la gestion de la consommation pour chaque production. Afin de corriger les lacunes de l'application actuelle, nous prévoyons d'ajouter des fonctionnalités telles que l'intégration du calcul du Taux de Rendement Synthétique (TRS) pour évaluer la performance en temps réel et identifier les domaines à améliorer. Nous envisageons également d'inclure une fonctionnalité d'impression de rapports personnalisés, avec des options pour les exporter au format PDF ou Excel. Pour améliorer l'accessibilité, nous migrerons l'application vers une plateforme web, garantissant une interface utilisateur intuitive et réactive accessible depuis n'importe quel appareil. Par ailleurs, des mesures de sécurité robustes seront mises en place pour protéger les données contre les accès non autorisés, assurant ainsi une gestion de production plus efficace et sécurisée.

1.10 Conclusion

Dans ce chapitre, nous avons présenté le Groupe Cevital. Nous avons exploré son histoire depuis sa fondation en 1971. En outre, nous avons examiné en détail Cevital Agro-industrie, notamment son emplacement stratégique à Bejaia, ses installations, son réseau de distribution et sa structure organisationnelle. Nous avons également identifié la problématique actuelle liée à la gestion de la production et proposé des solutions pour y remédier.

CHAPITRE 2 : Méthodologie de conception et spécification des besoins

2.1 Introduction

Dans ce chapitre, nous présentons la méthodologie de conception et la spécification des besoins pour notre système. Nous commençons par expliquer l'application de la méthode agile Scrum dans notre projet et l'utilisation du langage de modélisation UML pour représenter les différentes composantes de notre système. Ensuite, nous discutons de l'architecture MVC, en expliquant comment cette architecture divise notre système en trois composantes distinctes. La deuxième partie se concentre sur la spécification des besoins et des exigences de notre système, en mettant l'accent sur les avantages de la méthode Scrum en termes de productivité et d'efficacité. Ce chapitre pose ainsi les bases de notre projet en utilisant une méthodologie de conception agile

2.2 Méthodologie de conception

Pour la réalisation de notre projet, nous avons utilisé les méthodes de conception agile et le langage de modélisation unifié UML.

2.2.1 Méthodes agiles

Les méthodes agiles [5] sont souvent différentes des méthodes traditionnelles "cascade" ou "cycle en V". Plus souples et flexibles, elles placent les besoins du client au centre des priorités du projet. A leur création, ces méthodes permettent de gérer des projets de développement web et informatique. Aujourd'hui, les méthodes agiles sont de plus en plus courantes. Ce succès s'explique notamment par leur capacité d'adaptation à des projets variés dans tous les secteurs.

2.2.1.1 Méthode Scrum

Scrum est une méthode de développement agile itérative et incrémentale qui met l'accent sur la gestion de projet, la gestion d'équipe et adaptée pour un projet de moyenne à longue durées. Elle se concentre sur la planification des sprints, des réunions quotidiennes de stand up (Daily Scrum) afin de partager l'état d'avancement, les succès et les obstacles entre les membres et une rétrospective de sprint pour améliorer continuellement le processus de

Chapitre 2 Méthodologie de conception et spécification des besoins

développement. Après plusieurs itérations, une livraison de release qui comporte toutes les fonctionnalités développées et testées est prête pour les livrer au client [6].

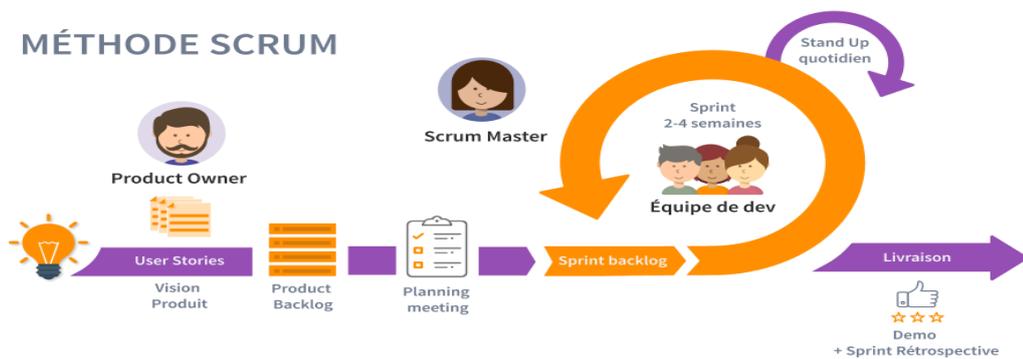


Figure 4: Cycle de vie de SCRUM

La figure ci-dessus présente les principaux composants de la méthode scrum

Les parties prenantes : La méthode scrum regroupe trois acteurs :

Le Product Owner qui représente l'utilisateur ou le client, le Scrum Master qui joue le rôle d'un facilitateur et il a pour mission de tout mettre en œuvre pour que l'équipe travaille dans de bonnes conditions et se concentre sur l'objectif du projet et enfin l'équipe de développement regroupant tous les rôles traditionnels : architecte, développeur, testeur, administrateur, etc.

Product-backlog (carnet de produit) : est une liste priorisée des fonctionnalités ou exigences du produit, sous forme de courtes descriptions appelées "user stories" Il représente l'ensemble des besoins du client ou des utilisateurs finaux pour le produit, et est maintenu et priorisé par le Product Owner.

Sprint : est une période de temps fixe, généralement de 1 à 4 semaines, pendant laquelle l'équipe de développement travaille à la réalisation d'un ensemble de fonctionnalités sélectionnées à partir du product backlog. Les fonctionnalités à réaliser sont définies lors de la planification du sprint, et l'équipe s'engage à les compléter dans la durée du sprint. À la fin du sprint, une version fonctionnelle du produit est prête à être évaluée ou livrée au client.

Justification du choix de la méthode Scrum

Nous avons utilisé la méthode Scrum pour les raisons suivantes :

- **Transparence** : Les tâches et le progrès sont visibles pour toute l'équipe. Chaque membre peut voir l'état d'avancement des différentes tâches grâce à des outils comme le tableau Scrum et les rapports de sprint.
- **Flexibilité** : Scrum permet de s'adapter rapidement aux changements. Les itérations courtes et les revues de sprint permettent d'ajuster les priorités et de répondre rapidement aux besoins changeants du client.
- **Amélioration continue** : Les rétrospectives régulières permettent d'identifier et d'implémenter des améliorations. À la fin de chaque sprint, l'équipe discute des points forts et des points à améliorer, ce qui favorise un apprentissage et un perfectionnement constants.
- **Priorisation efficace** : Les éléments de travail sont constamment priorisés pour maximiser la valeur. Le Product Owner est chargé de prioriser le backlog, garantissant ainsi que l'équipe travaille toujours sur les tâches les plus importantes et à forte valeur ajoutée.

De plus, Scrum permet de livrer des logiciels plus rapidement et de meilleure qualité. Elle offre à l'équipe un contrôle sur la quantité de travail réalisé et s'adapte rapidement aux changements demandés par le client. Les spécifications logicielles sont régulièrement réévaluées par l'équipe et le client, ce qui assure une grande réactivité et une adaptation continue aux besoins évolutifs.

En travaillant par itérations courtes appelées sprints, les équipes peuvent livrer fréquemment des incréments de produit fonctionnels, permettant des ajustements rapides en fonction des retours. Le backlog de produit, maintenu par le Product Owner, permet de prioriser les fonctionnalités selon la valeur ajoutée, tandis que les tâches du sprint backlog sont constamment ajustées par l'équipe en fonction de l'avancement et des imprévus.

Grâce à cette approche, Scrum favorise une collaboration étroite entre tous les membres de l'équipe ainsi qu'avec les parties prenantes, garantissant ainsi une meilleure communication et une compréhension commune des objectifs à atteindre. Cette méthode de

travail itérative et incrémentale contribue à améliorer la transparence et la satisfaction globale des clients et des utilisateurs finaux.

2.2.2 Langage de modélisation UML

Durant notre étude nous avons opté pour le langage UML, qui est un langage de modélisation orienté objet utilisé pour modéliser et représenter visuellement les différents aspects et composants d'un système logiciel, tels que la structure du système, les comportements des objets et les interactions entre eux, les cas d'utilisation, les séquences d'actions, les classes et les composants. UML fournit des éléments visuels et des notations standardisées qui permettent une communication efficace entre les développeurs et concepteurs sur les exigences et les fonctionnalités donc il offre une grande flexibilité [7].

Il existe deux types de vues de système : Les Diagrammes structurels et les diagrammes comportementaux comme montre la figure ci-dessous.

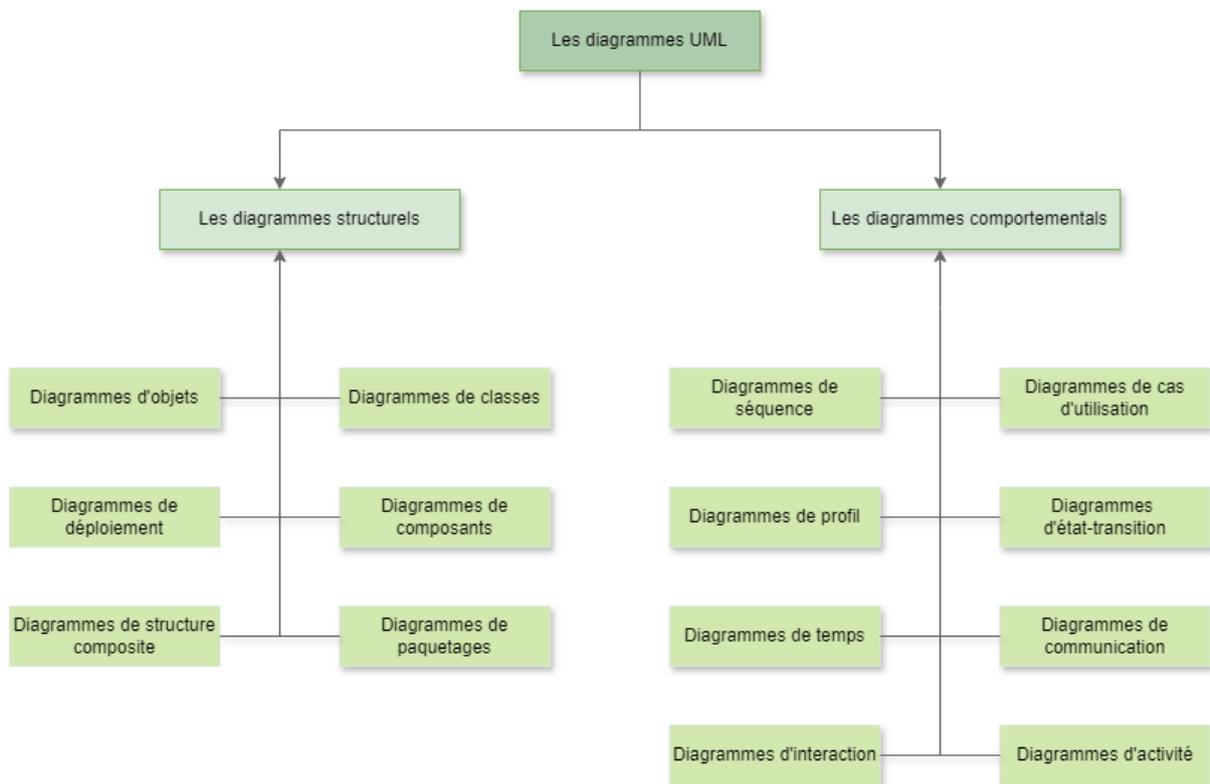


Figure 5: Différents diagrammes UML [7]

2.3 Architecture MVC

Le modèle MVC est un modèle d'architecture utilisé pour créer la structure d'une application qui consiste à séparer l'application en trois couches dépendantes et connectées : couche présentation (UI : User Interface), couche métiers (BLL : Business Logic Layer) et couche d'accès aux données (DAL : Data Access Layer). Cette architecture aide à la réutilisation du code et au développement parallèle ; ainsi les modifications effectuées sur n'importe quelle couche de l'application n'affectent pas les autres couches [8].

Les interactions entre les trois composants (Modèle, Vue et Contrôleur) ainsi qu'avec l'utilisateur sont illustrées dans la figure ci-dessous :

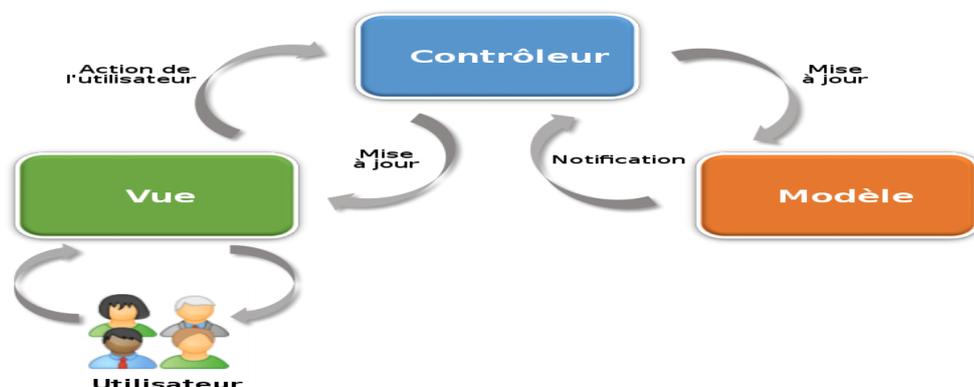


Figure 6:Architecture MVC [8]

-Modèle (Model) :

Définition : Le modèle représente la structure des données et la logique métier de l'application. Il gère la récupération, la manipulation et la mise à jour des données, ainsi que la gestion des règles et des comportements associés à ces données.

Rôle principal : Manipuler les données et répondre aux demandes de récupération ou de mise à jour de la part du contrôleur.

-Vue (View) :

Définition : La vue est responsable de la présentation des données à l'utilisateur. Elle affiche les informations extraites du modèle et rend l'interface utilisateur (UI).

Chapitre 2 Méthodologie de conception et spécification des besoins

Rôle principal : Afficher les données et l'état de l'application de manière cohérente et réactive en fonction des changements dans le modèle.

-Contrôleur (Controller) :

Définition : Le contrôleur agit comme un intermédiaire entre la vue et le modèle. Il reçoit les entrées de l'utilisateur, interprète ces actions, et détermine les modifications nécessaires sur le modèle. Ensuite, il met à jour la vue en conséquence.

Rôle principal : Gérer la logique de l'application, interpréter les commandes de l'utilisateur, et coordonner les mises à jour entre le modèle et la vue.

2.4 Pilotage du projet avec scrum

Avant d'appliquer la méthode scrum a notre projet nous devons identifier les différentes entités impliquées dans notre projet ainsi que les besoins fonctionnels et non fonctionnels

2.4.1 Définition des entités impliquées dans notre projet

Atelier : Il s'agit du lieu de production où les produits sont fabriqués. L'atelier est composé de plusieurs lignes de production.

Ligne : Les lignes sont les sections distinctes de l'atelier où se déroulent les différentes étapes de la production.

Catalogue d'arrêt : Ce catalogue répertorie les arrêts de production et est structuré en plusieurs niveaux, incluant les familles, sous-familles, classes, et l'impact des arrêts. Cela permet une gestion et une maintenance précises et efficaces.

Article : Les articles sont les composants ou produits gérés au sein du système, incluant les matières premières, les produits semi-finis, et les produits finis.

La recette : désigne la liste détaillée des articles, matériaux ou composants nécessaires à la fabrication d'un produit, incluant les éléments de packaging

La cadence : est destinée à chaque ligne de production et représente le rythme ou la vitesse à laquelle les opérations doivent être effectuées pour maintenir un flux de production efficace et répondre à la demande du marché

Chapitre 2 Méthodologie de conception et spécification des besoins

Les arrêts : Les arrêts de production sont des interruptions temporaires ou définitives du processus de fabrication, causées par des pannes, des pénuries, des problèmes de qualité, des grèves ou des maintenances. Ils affectent la productivité et les délais de livraison de l'entreprise.

Quart : Un quart de travail est une période prédéfinie durant laquelle des employés travaillent. Les quarts de travail peuvent varier en durée. Chaque quart est planifié pour assurer une couverture continue des opérations de l'entreprise. Pendant chaque quart, plusieurs équipes peuvent être assignées.

Equipe : Une équipe est un groupe d'employés assignés à travailler ensemble pendant un quart de travail spécifique. Chaque quart de travail peut comprendre plusieurs équipes. Les équipes sont formées pour accomplir des tâches et des responsabilités spécifiques durant leur période de travail.

Consommation IP : La consommation IP désigne les noms et les quantités des matières premières, composants ou sous-assemblages utilisés pour fabriquer un produit fini (PF). Chaque article PF est composé d'un ou plusieurs articles IP, et la consommation IP spécifie les articles spécifiques et leurs quantités nécessaires pour produire une unité du produit fini.

2.4.2 Identification des besoins fonctionnels et non fonctionnels

2.4.2.1 Les besoins fonctionnels :

- **Gestion des ateliers et des lignes de production :**

Créer et gérer les ateliers et les lignes de production.

- **Gestion des recettes de production :**

Créer et gérer les recettes de production pour chaque produit.

Assurer le contrôle des modifications des recettes.

- **Gestion des plannings :**

Créer et gérer les plannings

- **Gestion des utilisateurs et des rôles :**

Créer, modifier, supprimer et rechercher des utilisateurs.

Chapitre 2 Méthodologie de conception et spécification des besoins

Attribuer des rôles spécifiques aux utilisateurs (administrateur, responsable de production, opérateur).

Définir les permissions et les accès pour chaque rôle.

- **Rapports et analyses :**

Générer des rapports détaillés sur les productions.

Analyser les données de rendement pour chaque production.

- **Fournir des tableaux de bord personnalisables pour les différents utilisateurs et rôles.**

- **Gestion des arrêts de production :**

Enregistrer tous les arrêts de production.

- **Catalogues d'arrêts :**

Créer et gérer un catalogue d'arrêts classifié par :

Famille : Catégorie générale de l'arrêt (ex. : technique, qualité, organisationnel ...).

Sous-famille : Sous-catégorie plus spécifique (ex. : grève du personnel, maintenance préventif).

Classe : Classe de l'arrêt selon la nature de la cause (ex. : opération, contrôle de qualité, maintenance).

Impact : Évaluer l'impact de chaque type d'arrêt sur la production.

2.4.2.2 Les besoins non fonctionnels :

- **Accessibilité** : Garantir que l'application est utilisable par tous, y compris les personnes handicapées, en respectant les normes d'accessibilité Web telles que le WCAG (Web Content Accessibility Guidelines).
- **Testabilité** : Assurer que l'application est conçue de manière à être facilement testable, avec des fonctionnalités clairement définies et des cas de test bien documentés pour permettre une validation efficace.

Chapitre 2 Méthodologie de conception et spécification des besoins

- **Sécurité** : Mettre en place des mécanismes robustes pour protéger les données sensibles et prévenir les attaques malveillantes, en utilisant des technologies de chiffrement, des pare-feux et des pratiques de gestion des identités et des accès.
- **Performance et scalabilité** : Veiller à ce que l'application fonctionne de manière rapide et efficace même sous des charges élevées, avec des mécanismes de mise en cache, d'optimisation de requêtes et une architecture capable de s'adapter à une augmentation de la charge utilisateur.

2.4.3 Rôle et user stories

Pour notre projet, les rôles sont répartis comme suit :

Rôles SCRUM	Personnes affectées
Product Owner	L'entreprise CEVITAL
Scrum Master	Mr Chibouti Younes (notre encadrant à Cevital) Mme Ghanem Souhila
Équipes de développement	Mlles Ourabah Hanane et Bourouina Liza

Tableau 1:Equipe et roles

2.4.4 Identification des acteurs

Un acteur est l'idéalisation d'un rôle joué par une personne, un processus ou une entité externe qui interagit avec le système. Autrement dit, un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [9]. Les différents acteurs de notre système :

- Administrateur
- Responsable
- Opérateur

2.4.5 Identification des cas d'utilisation

Acteur	Cas d'utilisation
Administrateur	<ul style="list-style-type: none"> ● S'authentifier ● Gérer les comptes utilisateurs ● Gérer les rôles et des permissions ● Gérer les ateliers ● Gérer les lignes ● Gérer les catalogues des arrêts
Responsable	<ul style="list-style-type: none"> ● S'authentifier ● Gérer les articles ● Gérer les recettes ● Gérer les cadences
Administrateur Responsable Opérateur	<ul style="list-style-type: none"> ● S'authentifier ● Gérer les productions ● Gérer les arrêts de production ● Imprimer un rapport

Tableau 2: Identification des cas d'utilisation

2.4.6 Diagramme de contexte

Il s'agit d'un diagramme statique qui représente l'environnement général dans lequel un système est conçu pour opérer [9].

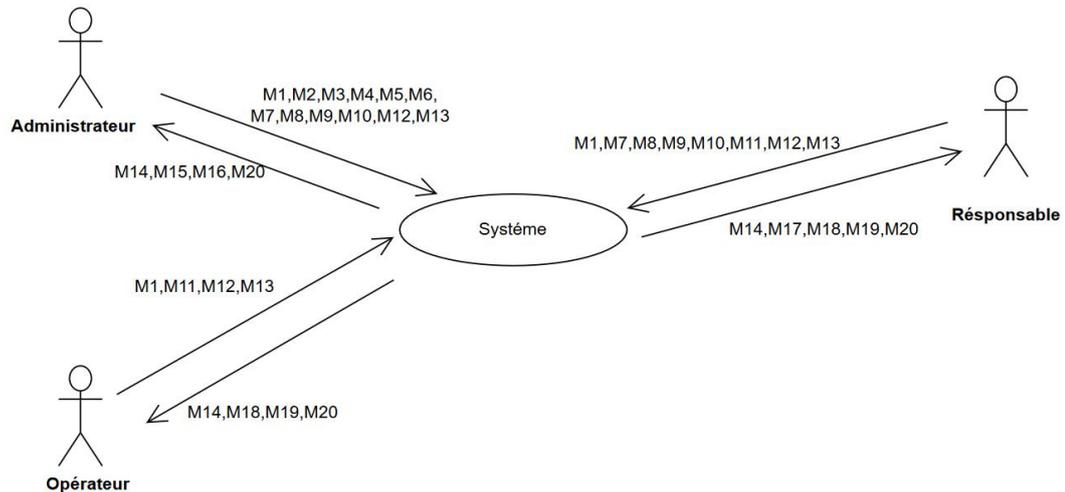


Figure 7: Diagramme de contexte

Identification des messages échangés

Un message est la formalisation d'une communication unidirectionnelle entre des entités, transportant de l'information dans le but de déclencher une activité chez le destinataire. Voici les échanges de messages entre les acteurs et le système [9] :

Messages entrants (acteur vers système)

- M1 : Demande d'authentification
- M2 : Demander la gestion des utilisateurs
- M3 : Demander la gestion des rôles et des permissions
- M4 : Demander la gestion des ateliers
- M5 : Demander la gestion des lignes
- M6 : Demander la gestion des catalogues d'arrêt
- M7 : Demander la gestion des articles
- M8 : Demander la gestion des recettes
- M9 : Demander la gestion des plannings de production
- M10 : Demander la gestion des cadences des lignes de production

Chapitre 2 Méthodologie de conception et spécification des besoins

- M11 : Demander la gestion des productions
- M12 : Déclarer des arrêts de production
- M13 : Imprimer un rapport

Messages sortants (système vers acteur)

- M14 : Interface d'authentification
- M15 : Interface de consultation des listes adéquate
- M16 : Interface pour la gestion de toutes les entités
- M17 : Interface de gestion d'articles, recettes, planning, cadence, arrêts
- M18 : Interface de gestion de production,
- M19 : Interface pour déclarer un arrêt
- M20 : Interface pour imprimer un rapport

2.4.7 Diagramme de cas d'utilisation globale

Le diagramme de cas d'utilisation est un diagramme UML qui représente les interactions entre les acteurs avec le système en question. Les acteurs peuvent être des acteurs internes qui ont un rôle à jouer dans le fonctionnement du système, comme les acteurs cités auparavant, ou des acteurs externes qui se trouvent en dehors du système et qui interagissent avec lui [10].

Le diagramme de cas d'utilisation suivant présente le comportement fonctionnel du système logiciel qui est décomposé en plusieurs sprints constituant le backlog product (carnet de produit).

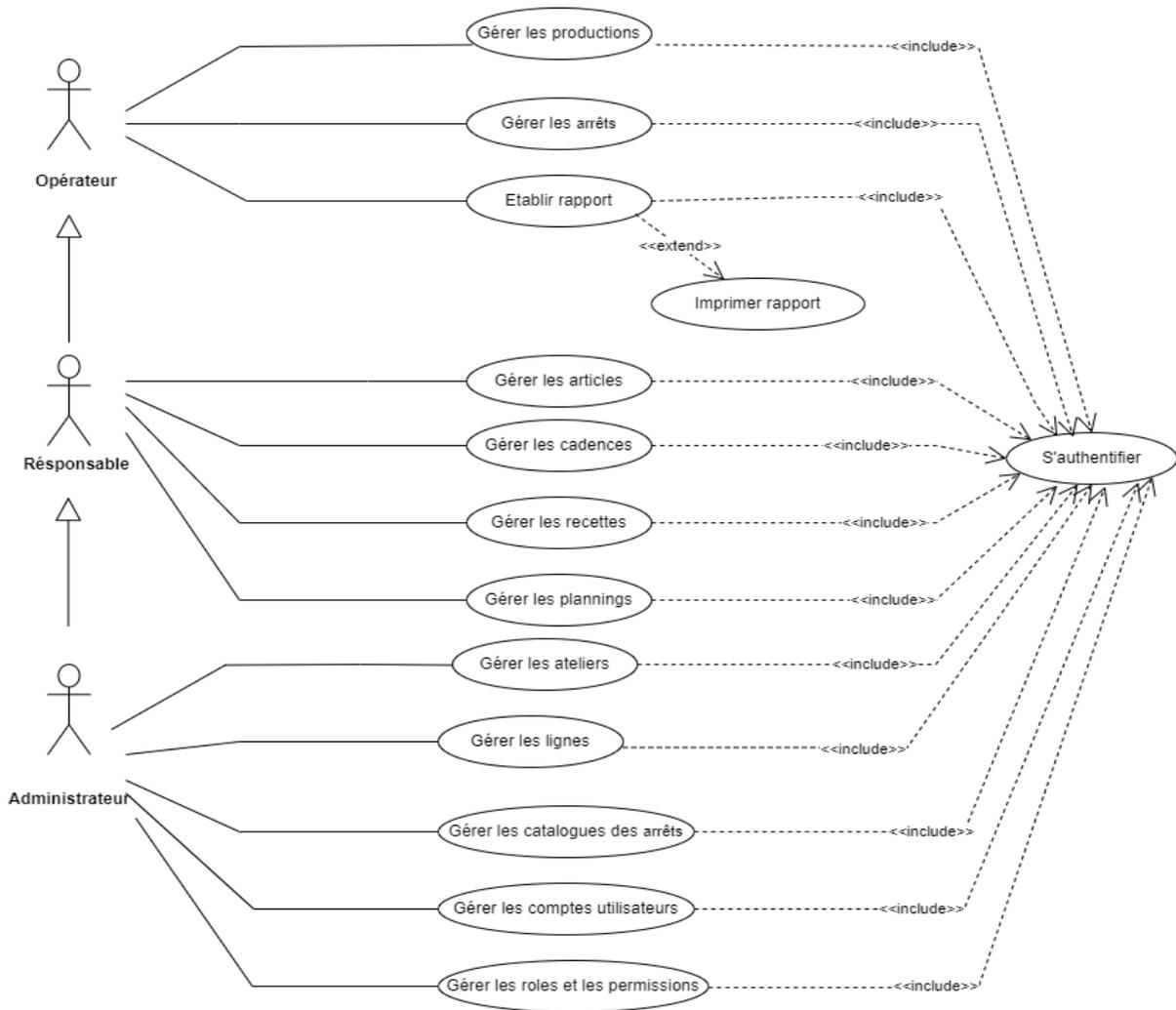


Figure 8: Diagramme de cas d'utilisation globale

2.4.8 Planification des sprints

Dans Scrum, le backlog représente une liste structurée des fonctionnalités à développer. Ces fonctionnalités, décrites sous forme de besoins, sont classées par ordre de priorité par le Product Owner

Sprint	Taches	Durée
Sprint 1	-Authentification.	1 Semaine
	-Gestion des comptes utilisateurs.	

	-Gestion des rôles et des permissions	
Sprint 2	-Gestion des ateliers	4 semaines
	-Gestion des lignes	
	-Gestion des catalogues d'arrêts	
	-Gestion des plannings	
	-Gestion des articles.	
Sprint 3	-Gestion des recettes	1 semaine
	-Gestions des cadences	
Sprint 4	-Gestion des productions	4 semaines
	-Gestions des arrêts d'une production	
	Imprimer un rapport	

Tableau 3:Planification des sprints

2.4.9 Backlog-Product (carnet de produit)

Chaque user-storie (histoire utilisateur) est caractérisée par une priorité définie par le Product Owner. Le backlog de notre futur système est le suivant :

User stories	En tant que	Je veux	Priorité
S'authentifier	Toute personne inscrite	M'authentifier	1
Gestion des utilisateurs	Administrateur	Gérer les comptes utilisateurs pour accéder à l'espace de l'application	2
Gestion des ateliers et lignes	Administrateur	Gérer des ateliers et associer des lignes pour chaque atelier	2
Gestion des plannings	Responsable + Administrateur	Créer, modifier et supprimer des plannings	3
Gestion des recettes	Responsable + Administrateur	Créer, modifier, supprimer des recettes	3
	Operateur	Récupérer cette recette lors de la saisie d'une production	
Gestion des articles	Responsable + Administrateur	Créer, modifier et supprimer un article	3
	Opérateur	Récupérer les articles lors de la saisie d'une production	
Gestion des cadences	Responsable + Administrateur	Gérer la cadence de chaque ligne	3
Gestion des arrêts	Responsable + Administrateur+ l'Opérateur	Déclarer des arrêts, selon leur familles, sous-famille, spécifier l'impact de chaque arrêt	3
Gestion des productions	Responsable + Administrateur+ Opérateur	Créer, modifier, rechercher une	4

Chapitre 2 Méthodologie de conception et spécification des besoins

		production, calculer le temps total, le temps utile et le taux de rendement d'une production	
Imprimer rapport	Administrateur+ Responsable+ Operateur	Etablir le rapport de production et l'imprimer	5

Tableau 4:Product backlog du système

2.4.10 Charte graphique

La charte graphique peut être définie comme étant l'identité visuelle d'une application, d'une entreprise ou d'une marque. C'est un guide qui détermine tous les éléments graphiques (logo, couleur, police, typographie et maquette), leurs utilisations et leurs caractéristiques. Elle permet de donner une cohérence afin de donner à l'utilisateur des repères et lui faciliter sa visite [11].

Logotype

Un logo est une représentation graphique qui a pour but d'identifier de manière unique et directe une marque, une entreprise ou une application.

La figure suivante présente l'apparence du logo de notre application, qui est le logo de Cevital.



Figure 9:Logo

Les couleurs dominantes

Dans notre application, nous avons opté pour une palette de couleurs soigneusement sélectionnées afin de garantir une expérience utilisateur agréable et intuitive. La couleur blanche prédomine l'interface, offrant un design épuré qui facilite la lecture des contenus. Les

Chapitre 2 Méthodologie de conception et spécification des besoins

boutons sont en bleu, attirant l'attention des utilisateurs et indiquant clairement les actions disponibles. Le menu est entièrement gris avec quelques touches de jaune, ajoutant un contraste subtil et une touche de dynamisme sans surcharger l'interface. Ces choix de couleurs contribuent à créer un environnement visuellement cohérent et professionnel.



2.5 Conclusion

Dans ce chapitre, nous avons présenté un aperçu de la méthode Scrum et de ses différents composants. Nous avons également introduit quelques notions du langage de modélisation UML. Ensuite, nous avons spécifié les besoins et les exigences de notre système à travers les diagrammes de contexte et de cas d'utilisation. Cette étape nous a donné l'occasion de nous familiariser avec la méthode Scrum et de décrire de manière exhaustive les besoins de nos utilisateurs dans le backlog product, ainsi que le fonctionnement souhaité du système à développer pour en faciliter la mise en place et la maintenance.

CHAPITRE 3 : Application de la méthode SCRUM sur le Projet

3.1 Introduction

Dans ce chapitre, nous allons appliquer la méthode Scrum à notre projet et étudier tous les sprints réalisés. Nous commencerons par décrire les diagrammes de cas d'utilisation de chaque sprint, suivis des descriptions textuelles et des diagrammes de séquence correspondants. Cette approche nous permettra de structurer le développement du projet de manière itérative et incrémentale, en nous assurant que chaque fonctionnalité est bien définie, développée et testée à chaque étape du processus.

3.2 Etude du Sprint 1

Ce premier sprint s'étale sur une semaine, il comprend trois items qui sont :

- Authentification
- Gestion des comptes utilisateurs
- Gestion des rôles et permissions

3.2.1 Diagramme de cas d'utilisation du sprint 1

Le diagramme de cas d'utilisation suivant représente de manière générale les cas d'utilisation constituant le premier sprint

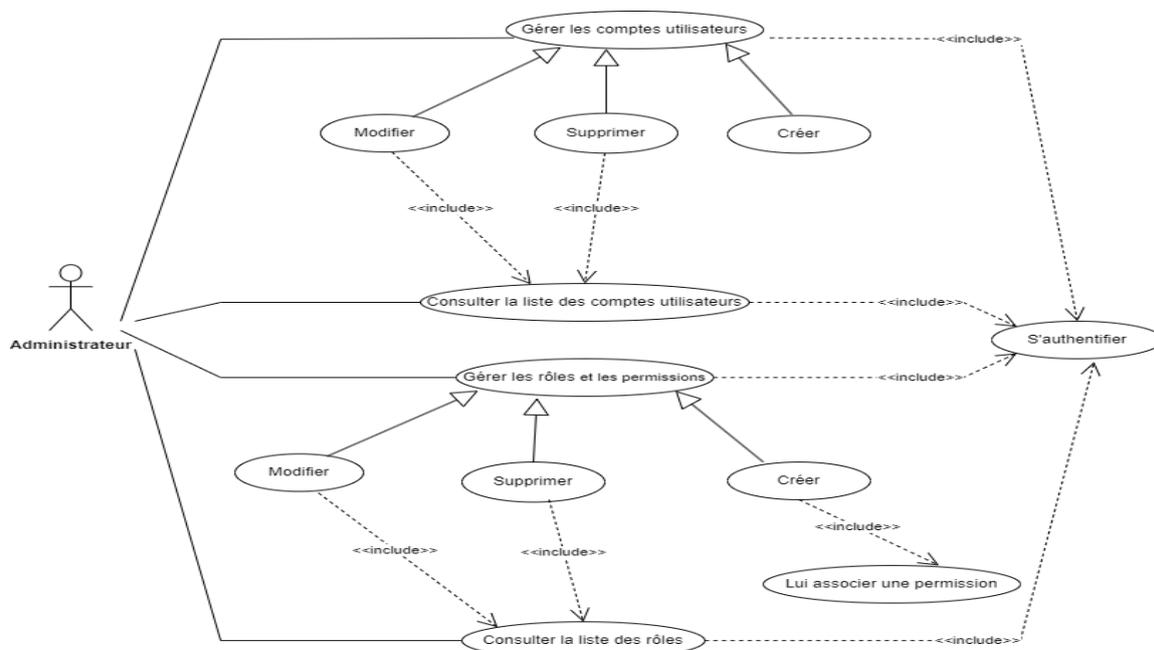


Figure 10: Diagramme de cas d'utilisation du sprint 1

3.2.2 Description textuelle du cas « S’authentifier »

Cas d’utilisation	S’authentifier
Acteur	Administrateur / Opérateur / Responsable
Objectif	Permet à l'utilisateur d'accéder à son compte.
Précondition	Le système fonctionne
Postcondition	L'utilisateur est soit authentifié ou non
Scénario principal	<ul style="list-style-type: none"> -L'utilisateur se connecte à l'application -Le système affiche la page de connexion -L'utilisateur saisit son nom d'utilisateur et son mot de passe puis valide -Le système effectue une vérification des informations saisies. -Le système affiche l'interface correspondante
Exception	<ul style="list-style-type: none"> -Si un champ est vide, le système affiche un message d'erreur. -Si le compte n'existe pas, le système affiche un message d'erreur.

Tableau 5:Description textuelle du cas : « s’authentifier »

3.2.3 Description textuelle du cas « Gestion des comptes utilisateurs »

Cas d'utilisation	Gestions des comptes utilisateurs
Acteur	Administrateur
Objectif	Gérer les comptes des utilisateurs
Précondition	L'utilisateur doit être authentifié en tant qu'admin
Postcondition	Les comptes sont ajoutés, modifiés, supprimés ou recherchés ou consultés avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'admin accède à l'interface de gestion des comptes. 2. L'admin choisit une action (ajouter, supprimer, modifier, rechercher). 3. L'admin entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'admin.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 6:Description textuelle du cas « Gestion des comptes utilisateurs »

3.2.4 Description textuelle du cas « Gestions des rôles »

Cas d'utilisation	Gestions des rôles
Acteur	Administrateur
Objectif	Gérer les rôles et les permissions des utilisateurs afin de contrôler leur accès aux différentes fonctionnalités du système.
Précondition	L'utilisateur doit être authentifié en tant qu'admin
Postcondition	Les rôles et permissions sont ajoutés, modifiés, supprimés ou consultés avec succès.
Scénario principal	<ol style="list-style-type: none"> 1. L'admin accède à l'interface de gestion des rôles et des permissions. 2. L'admin choisit une action (ajouter, modifier, supprimer, consulter un rôle ou une permission). 3. L'admin entre les détails nécessaires pour l'action choisie 4. Le système traite la demande et met à jour les rôles et permissions en conséquence. 5. Le système confirme l'action à l'admin.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 7:Description textuelle du cas « Gestions des rôles et permissions »

Dans ce qui suit nous présentons les diagrammes d'interaction du cas d'utilisation : authentifier et modifier un compte et ajouter un rôle.

3.2.5 Diagramme de séquence du cas « s'authentifier »

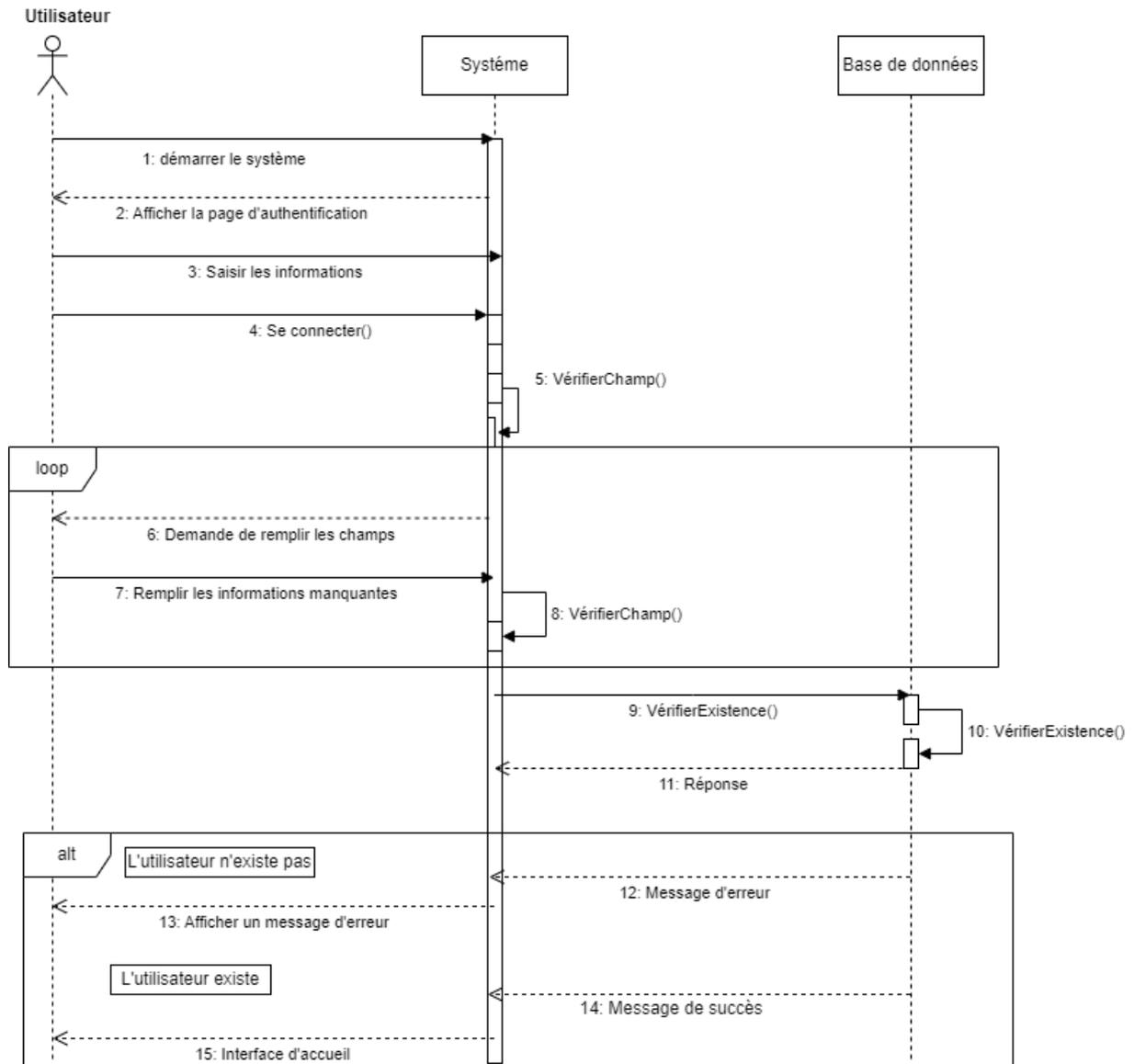


Figure 11: Diagramme de séquence du cas : « S'authentifier »

3.2.6 Diagramme de séquence du cas « modifier un compte utilisateur »

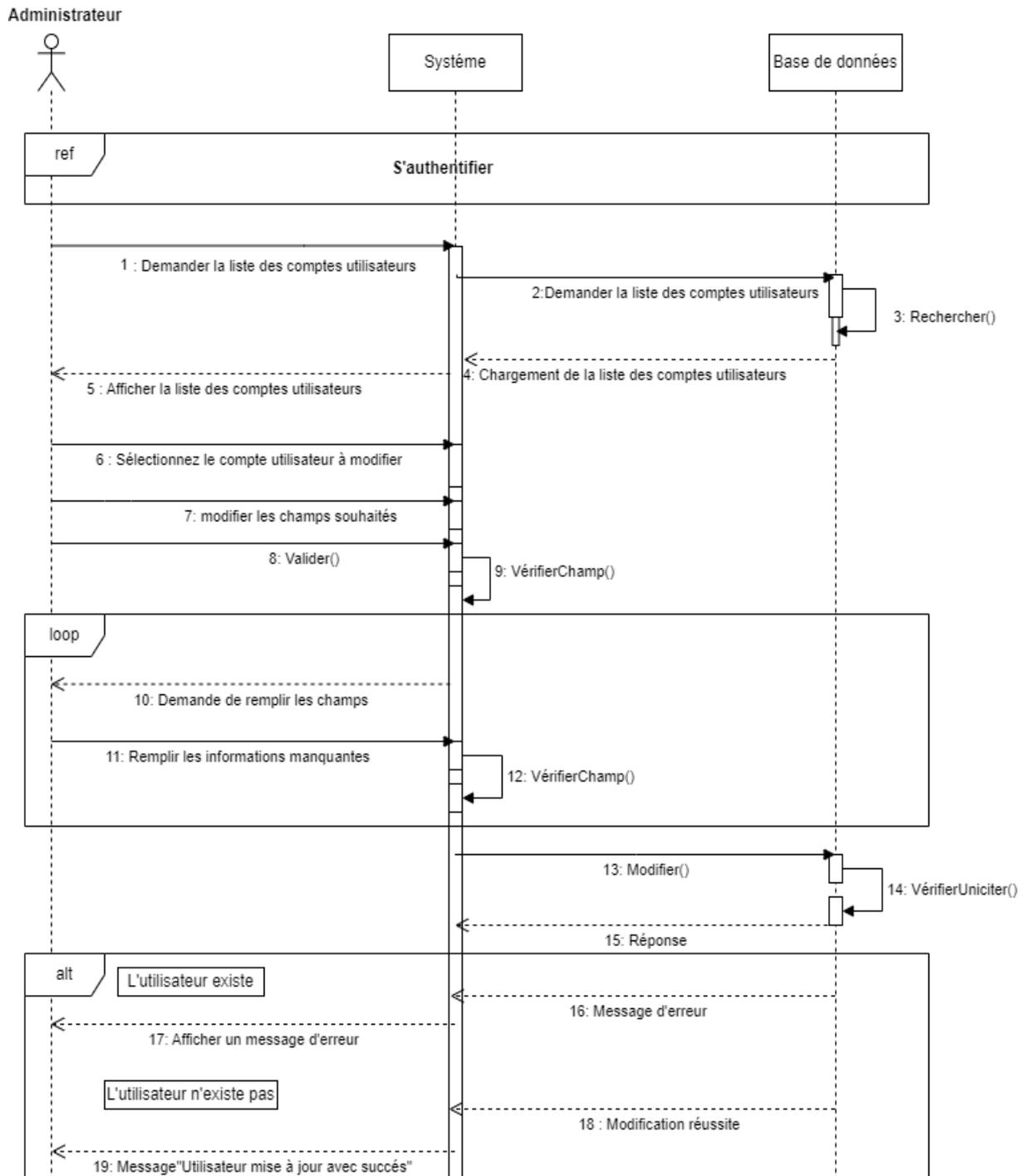


Figure 12: Diagramme de séquence du cas « modifier un compte utilisateur »

3.2.7 Diagramme de séquence du cas « ajouter un rôle »

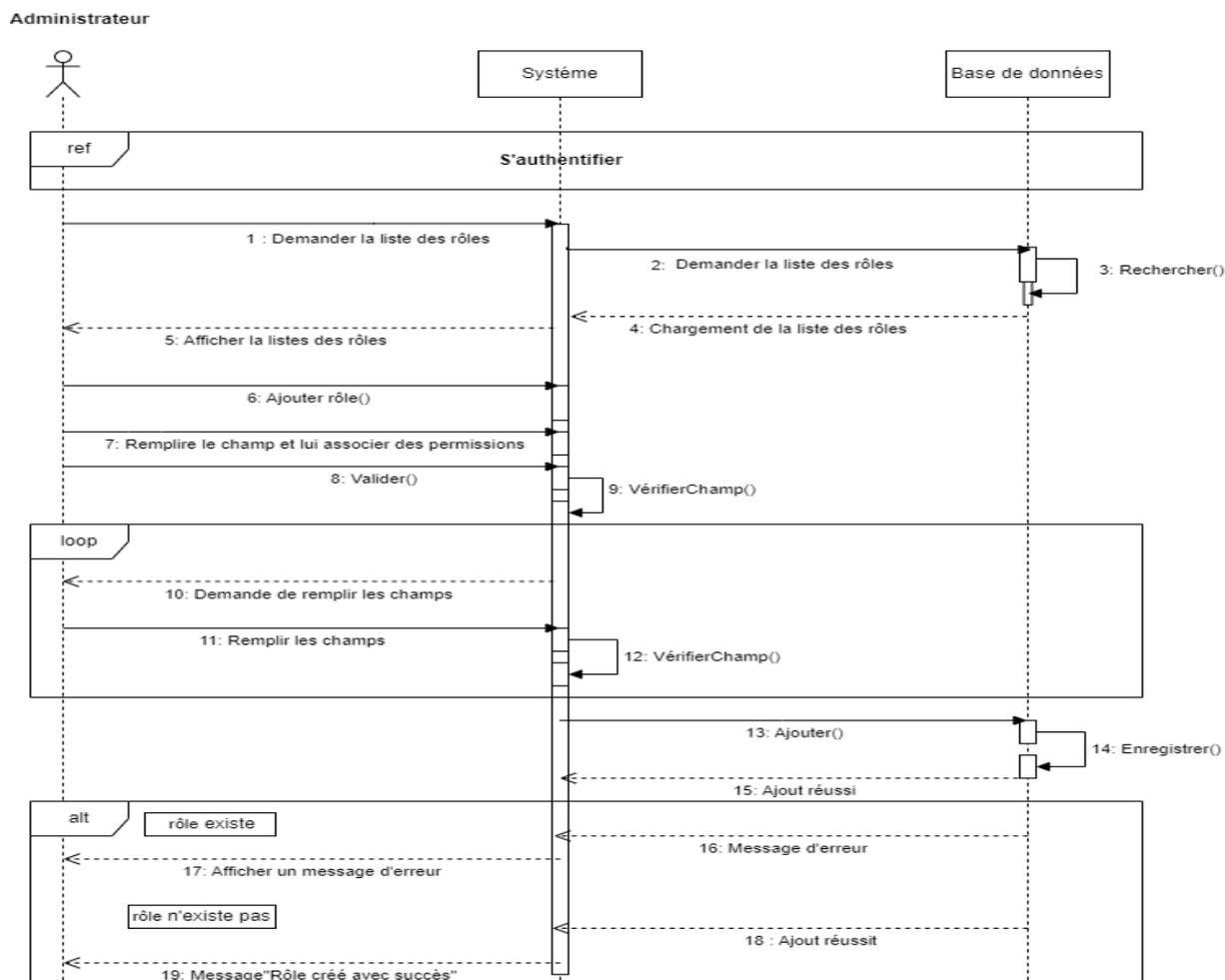


Figure 13: Diagramme de séquence du cas « ajouter un rôle »

3.3 Etude du sprint 2

Ce premier sprint s'étale sur quatre semaines, il comprend cinq items qui sont :

- Gestion des ateliers
- Gestion des lignes
- Gestion des catalogues d'arrêts
- Gestion des plannings
- Gestion des articles

3.3.1 Diagramme de cas d'utilisation du sprint 2

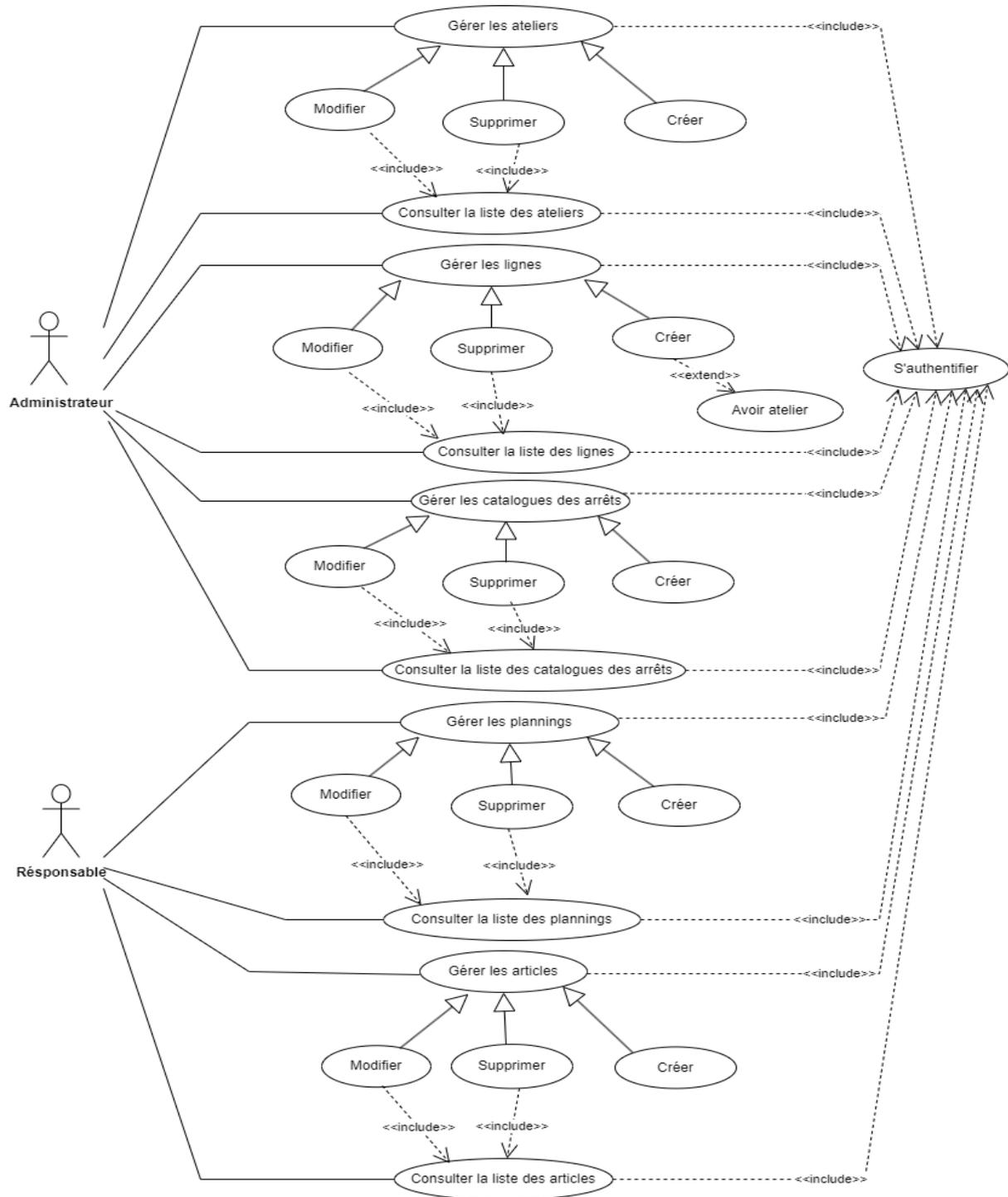


Figure 14: Diagramme de cas d'utilisation du sprint 2

3.3.2 Description textuelle du cas « Gérer les ateliers »

Cas d'utilisation	Gérer les ateliers
Acteur	Administrateur
Objectif	Gérer les ateliers
Précondition	L'utilisateur doit être authentifié en tant qu'admin
Postcondition	Les ateliers sont ajoutés, modifiés, supprimés ou consultés avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'admin accède à l'interface de gestion des ateliers. 2. L'admin choisit une action (ajouter, supprimer, modifier). 3. L'admin entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'admin.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 8:Description textuelle du cas « Gérer les ateliers »

3.3.3 Description textuelle du cas « Gérer les lignes »

Cas d'utilisation	Gérer les lignes
Acteur	Administrateur
Objectif	Gérer les lignes
Précondition	L'utilisateur doit être authentifié en tant qu'admin

Postcondition	Les lignes sont ajoutées, modifiées, supprimés ou consultés avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'admin accède à l'interface de gestion des lignes. 2.L'admin choisit une action (ajouter, supprimer, modifier). 3. L'admin entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'admin.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 9:Description textuelle du cas « Gérer les lignes »

3.3.4 Description textuelle du cas « Gérer les catalogues d’arrêts »

Cas d’utilisation	Gérer les catalogues d’arrêts
Acteur	Administrateur
Objectif	Gérer les catalogues d’arrêts
Précondition	L'utilisateur doit être authentifié en tant qu'admin
Postcondition	Les catalogues d’arrêt sont ajoutés, modifiés, supprimés ou consulter avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'admin accède à l'interface de gestion des catalogues d’arrêt. 2.L'admin choisit une action (ajouter, supprimer, modifier). 3. L'admin entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données.

	5. Le système confirme l'action à l'admin.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 10:Description textuelle du cas « Gérer les catalogues d'arrêts »

3.3.5 Description textuelle du cas « Gérer les articles »

Cas d'utilisation	Gérer les articles
Acteur	Administrateur / responsable
Objectif	Gérer les articles
Précondition	L'utilisateur doit être authentifié en tant qu'admin
Postcondition	Les articles sont ajoutés, modifiés, supprimés, recherchés ou consultés avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'admin accède à l'interface de gestion des articles. 2.L'admin choisit une action (ajouter, supprimer, modifier). 3. L'admin entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'admin.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 11:Description textuelle du cas « Gérer les articles »

3.3.6 Description textuelle du cas « Gérer les plannings »

Cas d'utilisation	Gérer les plannings
Acteur	Administrateur/responsable
Objectif	Gérer les plannings
Précondition	L'utilisateur doit être authentifié en tant qu'admin
Postcondition	Les plannings sont ajoutés, modifiés, supprimés ou consultés avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'admin accède à l'interface de gestion des plannings. 2. L'admin choisit une action (ajouter, supprimer, modifier). 3. L'admin entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'admin.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 12:Description textuelle du cas « Gérer les plannings »

Dans ce qui suit nous présentons les diagrammes d'interaction du cas d'utilisation : ajouter un atelier, modifier une ligne, rechercher un article, supprimer un planning et supprimer un catalogue d'arrêt

3.3.7 Diagramme de séquence du cas : « Ajouter un atelier »

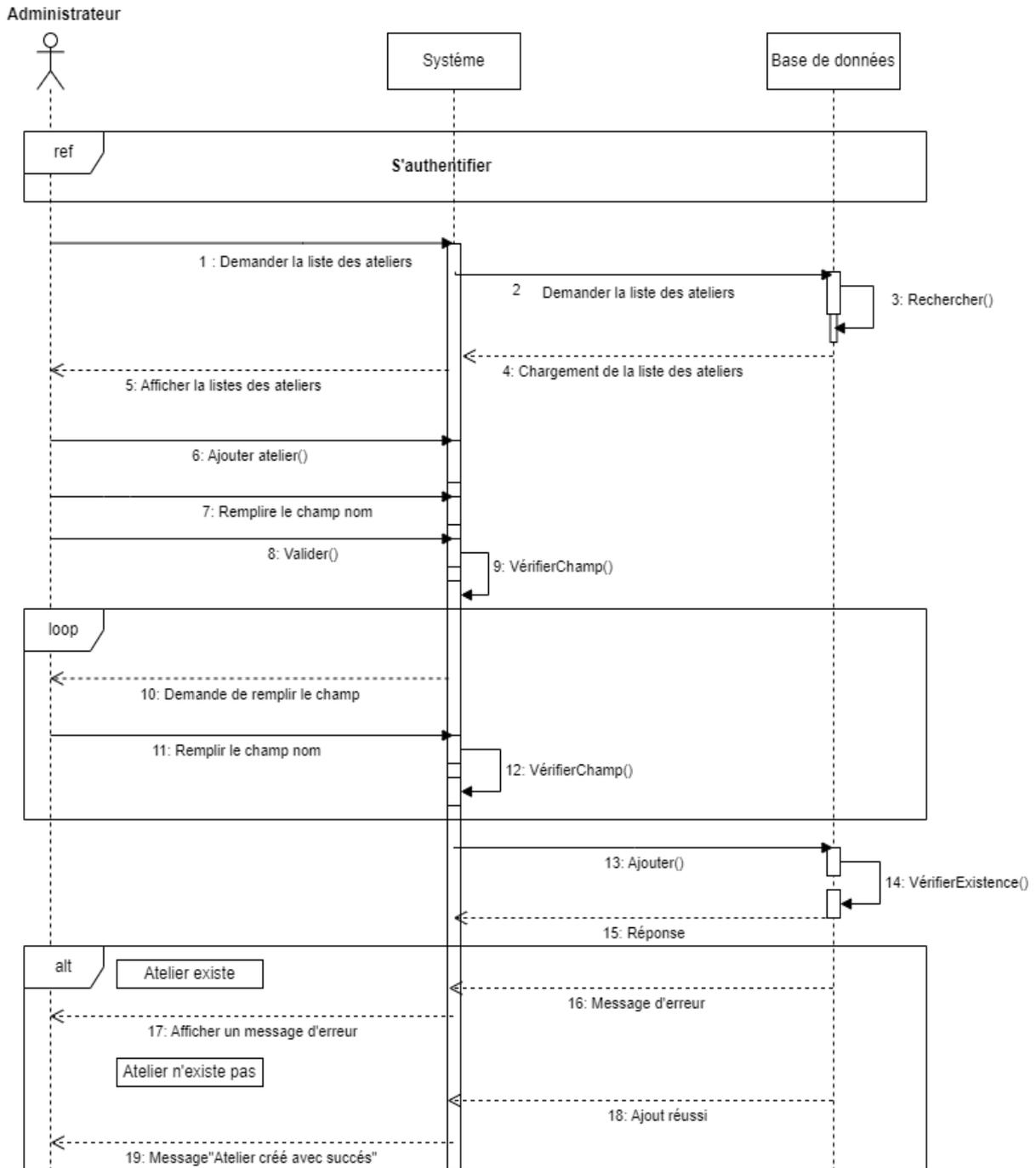


Figure 15: Diagramme de séquence du cas : « Ajouter un atelier »

3.3.8 Diagramme de séquence du cas : « modifier une ligne »

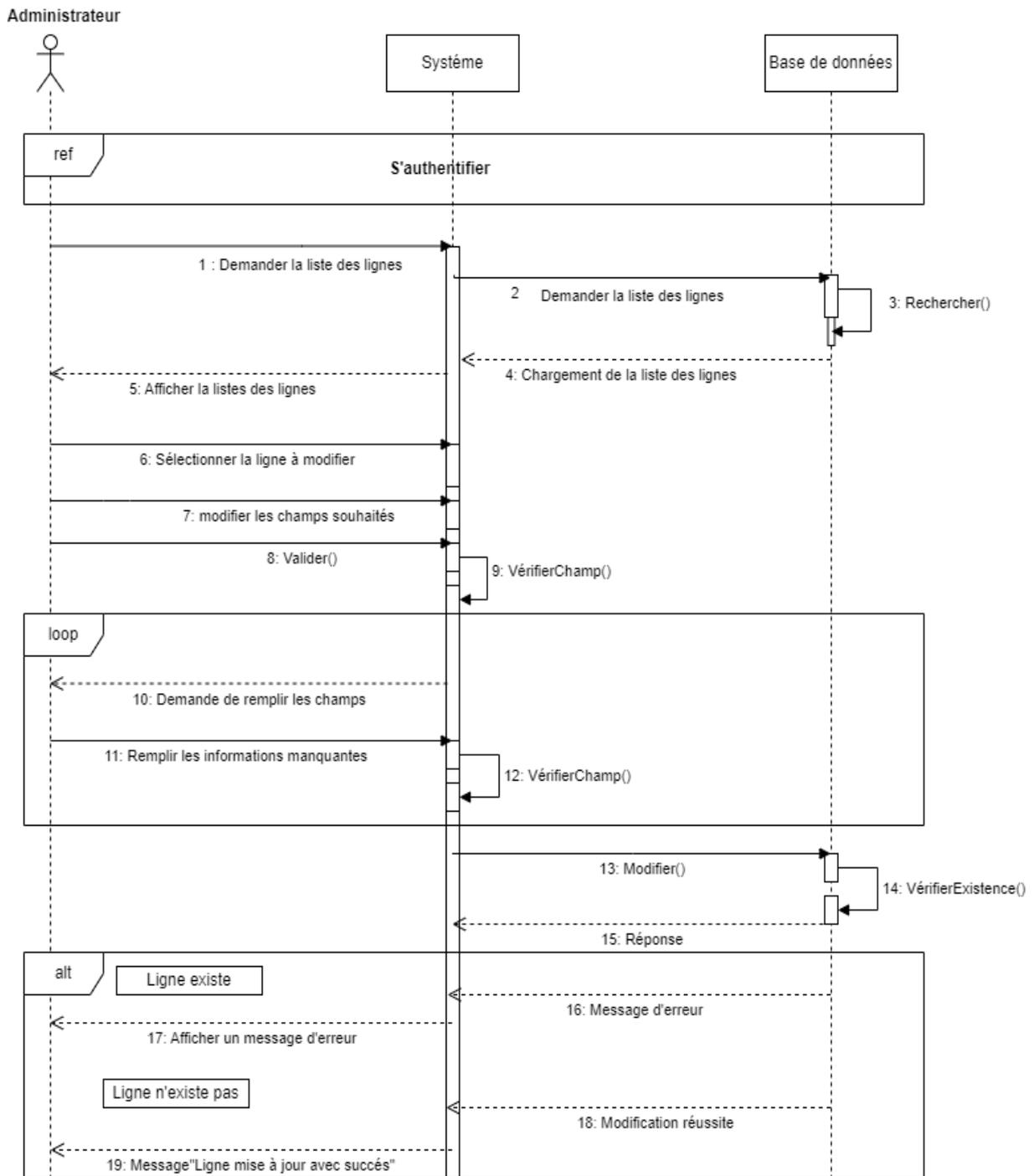


Figure 16: Diagramme de séquence du cas : « modifier une ligne »

3.3.9 Diagramme de séquence du cas « rechercher un article »

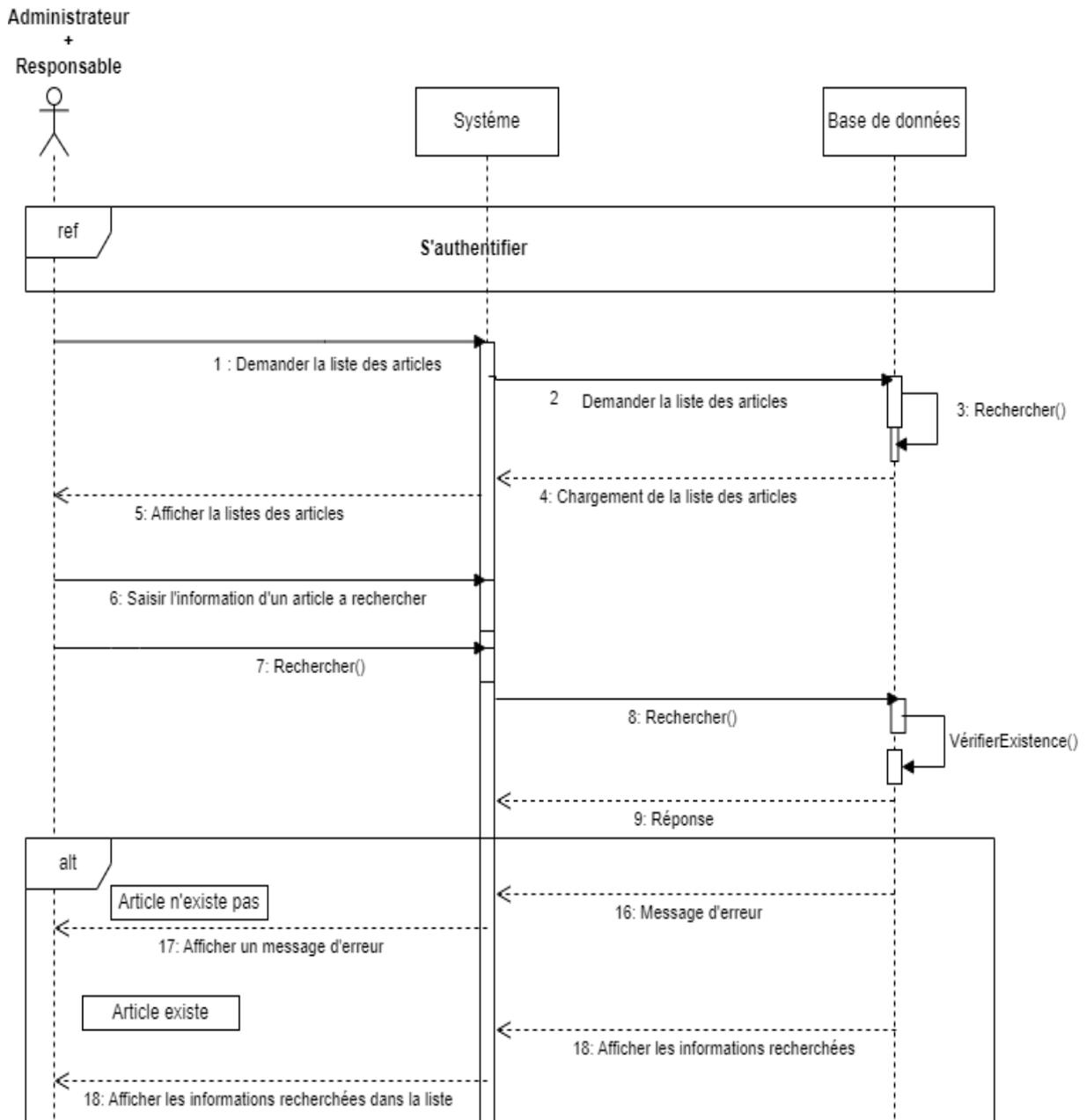


Figure 17:Diagramme de séquence du cas « rechercher un article »

3.3.10 Diagramme de séquence du cas « supprimer un catalogue »

Administrateur

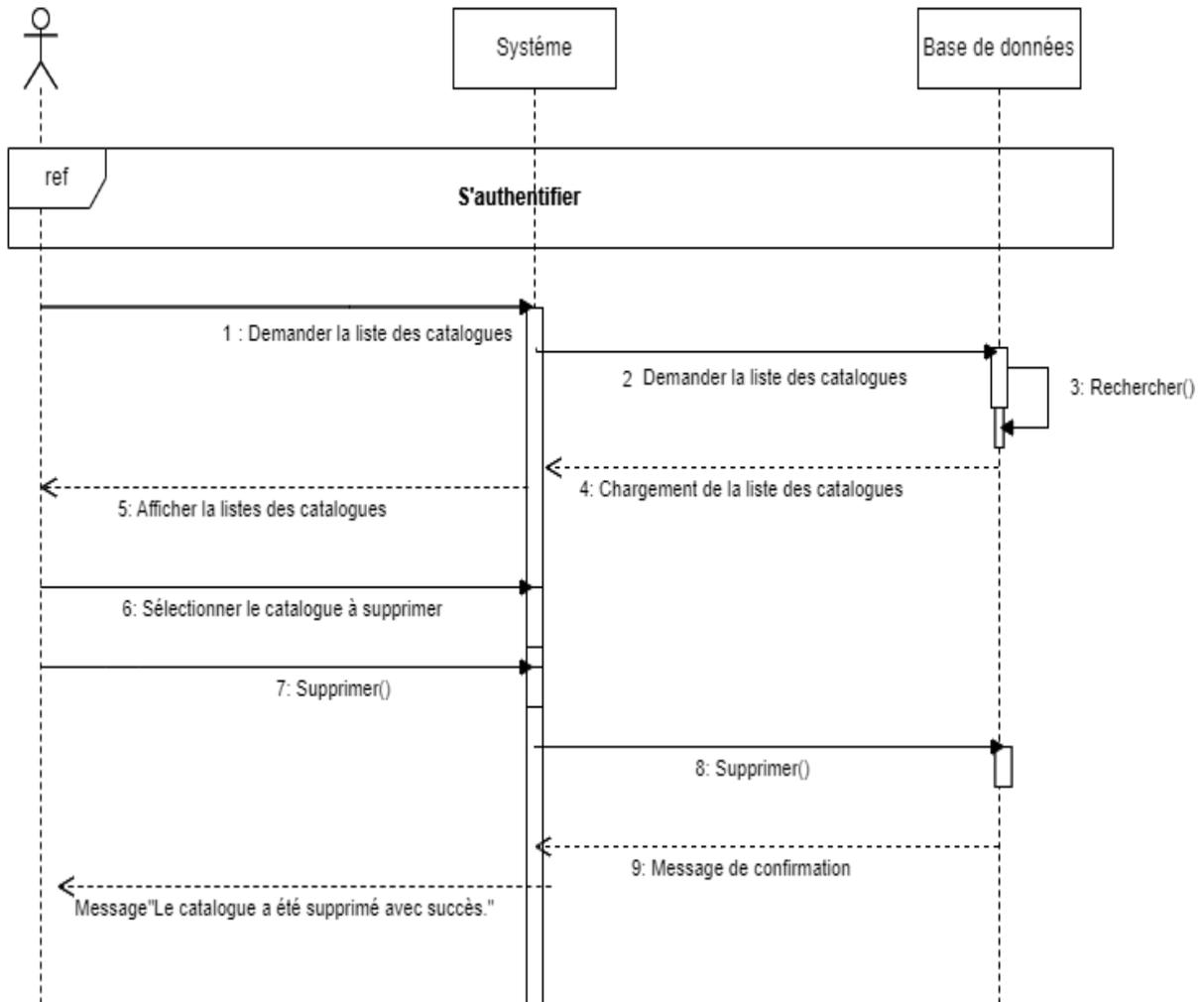


Figure 18:Diagramme de séquence du cas « supprimer catalogue »

3.3.11 Diagramme de séquence du cas « supprimer un planning »

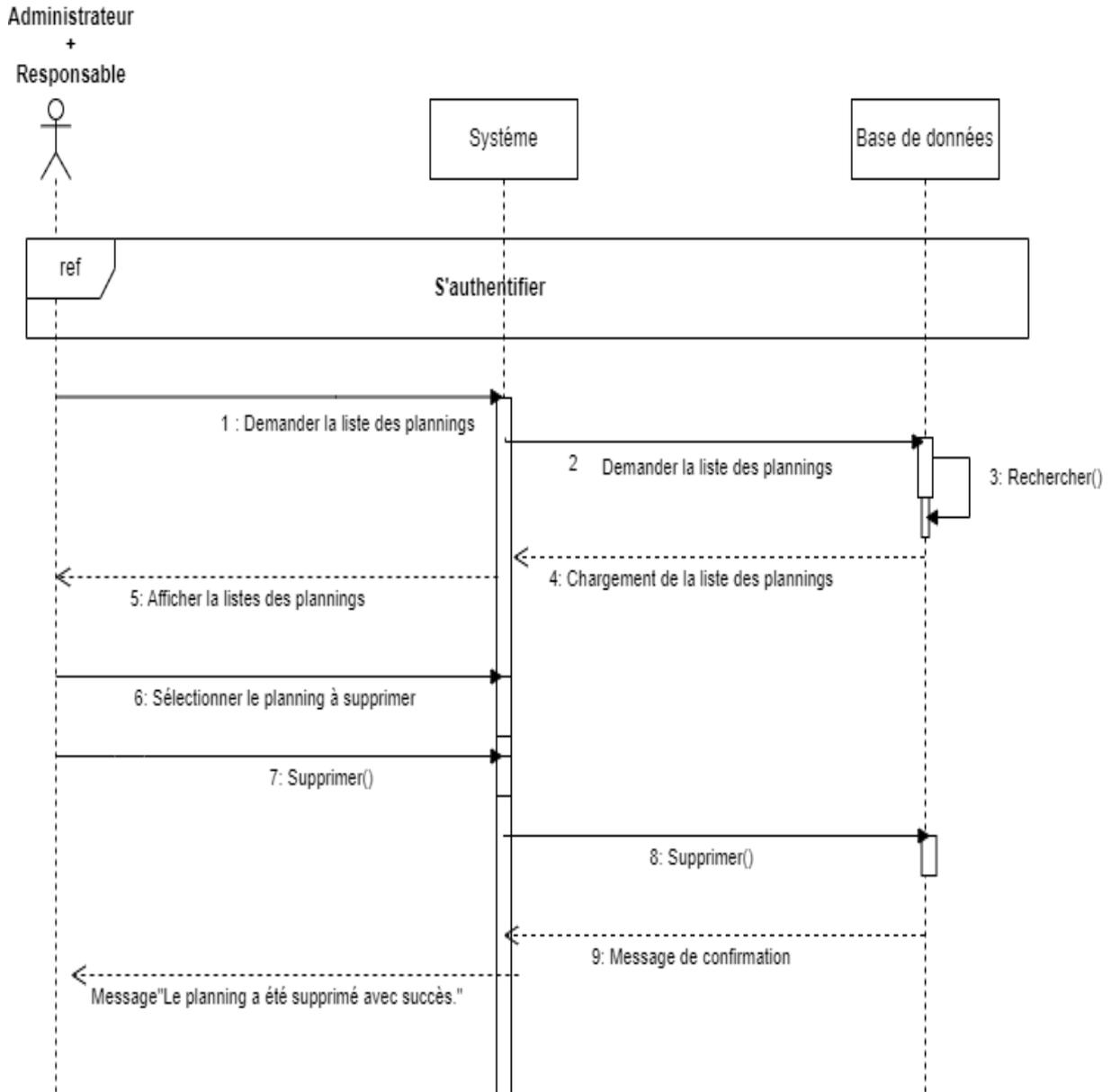


Figure 19: Diagramme de séquence du cas « supprimer un planning »

3.4 Etude du sprint 3

Ce troisième sprint s'étale sur une semaine, il comprend deux items qui sont :

- Gestion des recettes
- Gestion des cadences

3.4.1 Diagramme des cas d'utilisation du sprint 3

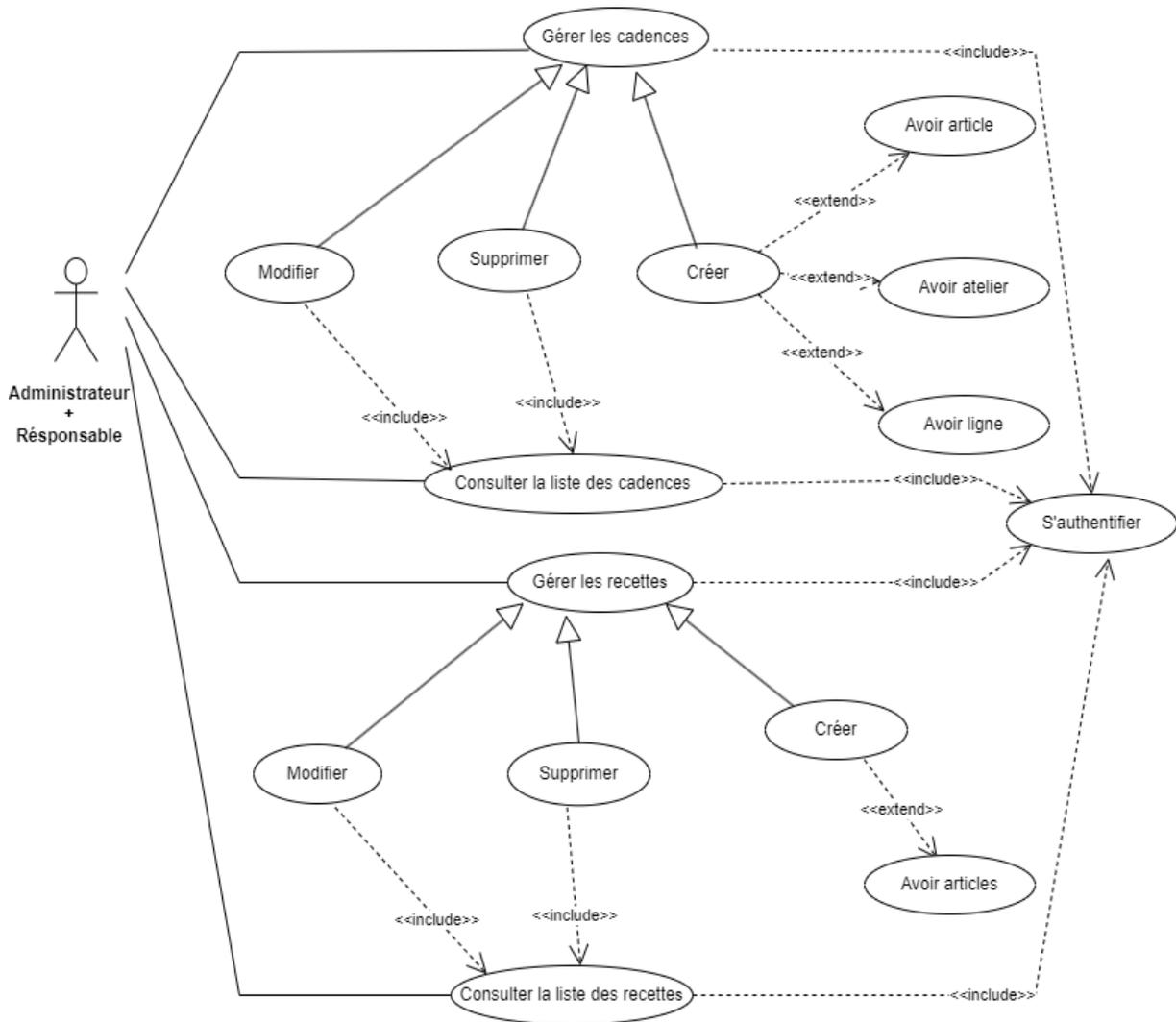


Figure 20: Diagramme du cas d'utilisation du sprint 3

3.4.2 Description textuelle du cas : « Gérer les recettes »

Cas d'utilisation	Gérer les recettes
Acteur	Administrateur/Responsable
Objectif	Gérer les recettes
Précondition	L'utilisateur doit être authentifié en tant qu'administrateur ou responsable
Postcondition	Les recettes sont ajoutées, modifiées, supprimées ou consultées avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de gestion des recettes. 2. L'utilisateur choisit une action (ajouter, supprimer, modifier). 3. L'utilisateur entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'utilisateur.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 13:Description textuelle du cas : « Gérer les recettes »

3.4.3 Description textuelle du cas « Gérer les cadences »

Cas d'utilisation	Gérer les cadences
Acteur	Administrateur/Responsable
Objectif	Gérer les cadences
Précondition	L'utilisateur doit être authentifié en tant qu'administrateur ou responsable
Postcondition	Les cadences sont ajoutées, modifiées, supprimées ou consultées avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de gestion des cadences. 2. L'utilisateur choisit une action (ajouter, supprimer, modifier). 3. L'utilisateur entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'utilisateur.
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 14:Description textuelle du cas « Gérer les cadence »

3.4.4 Diagramme de séquence du cas « ajouter une recette »

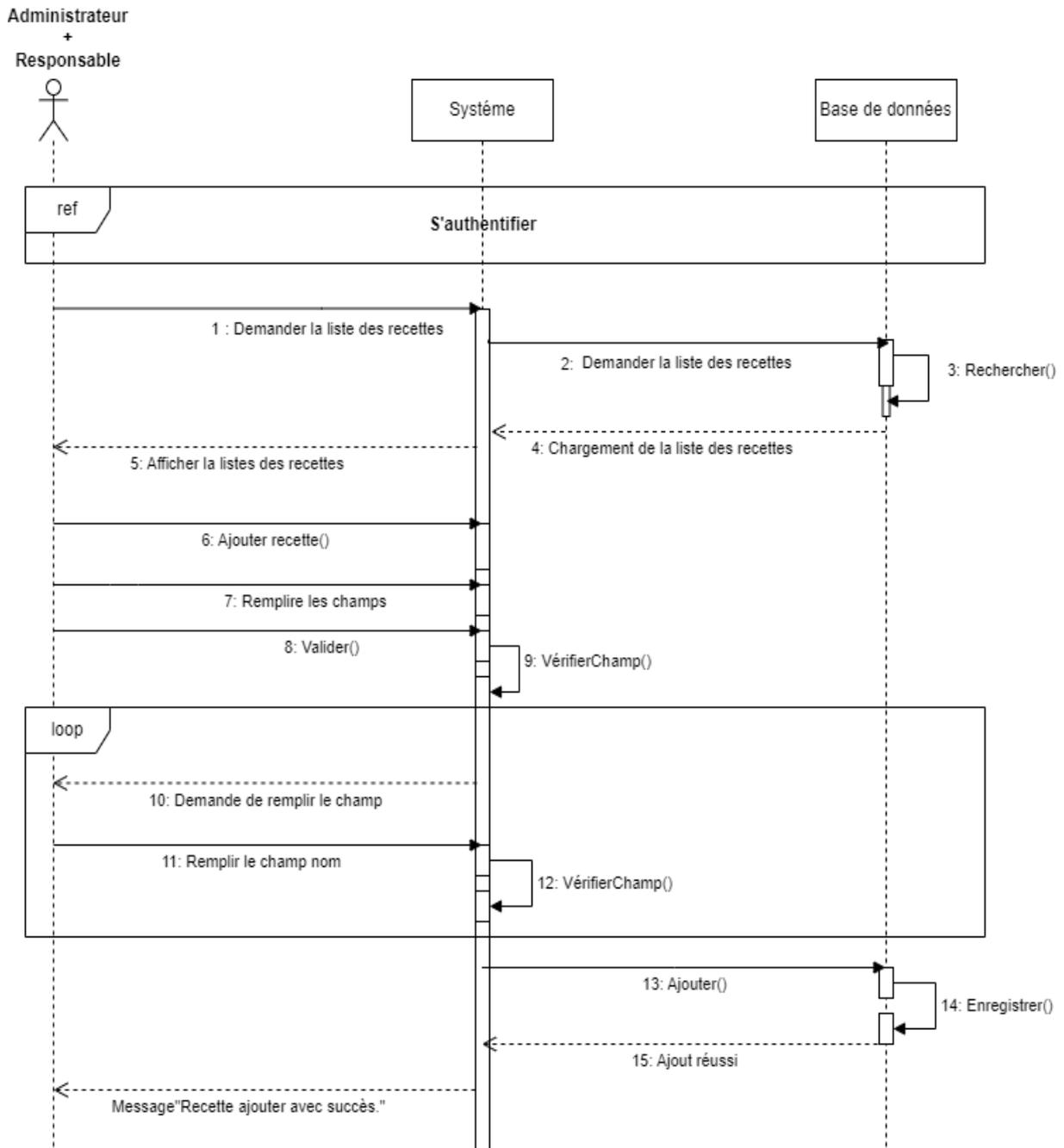


Figure 21:Diagramme de séquence du cas « ajouter une recette »

3.4.5 Diagramme de séquence « modifier une cadence »

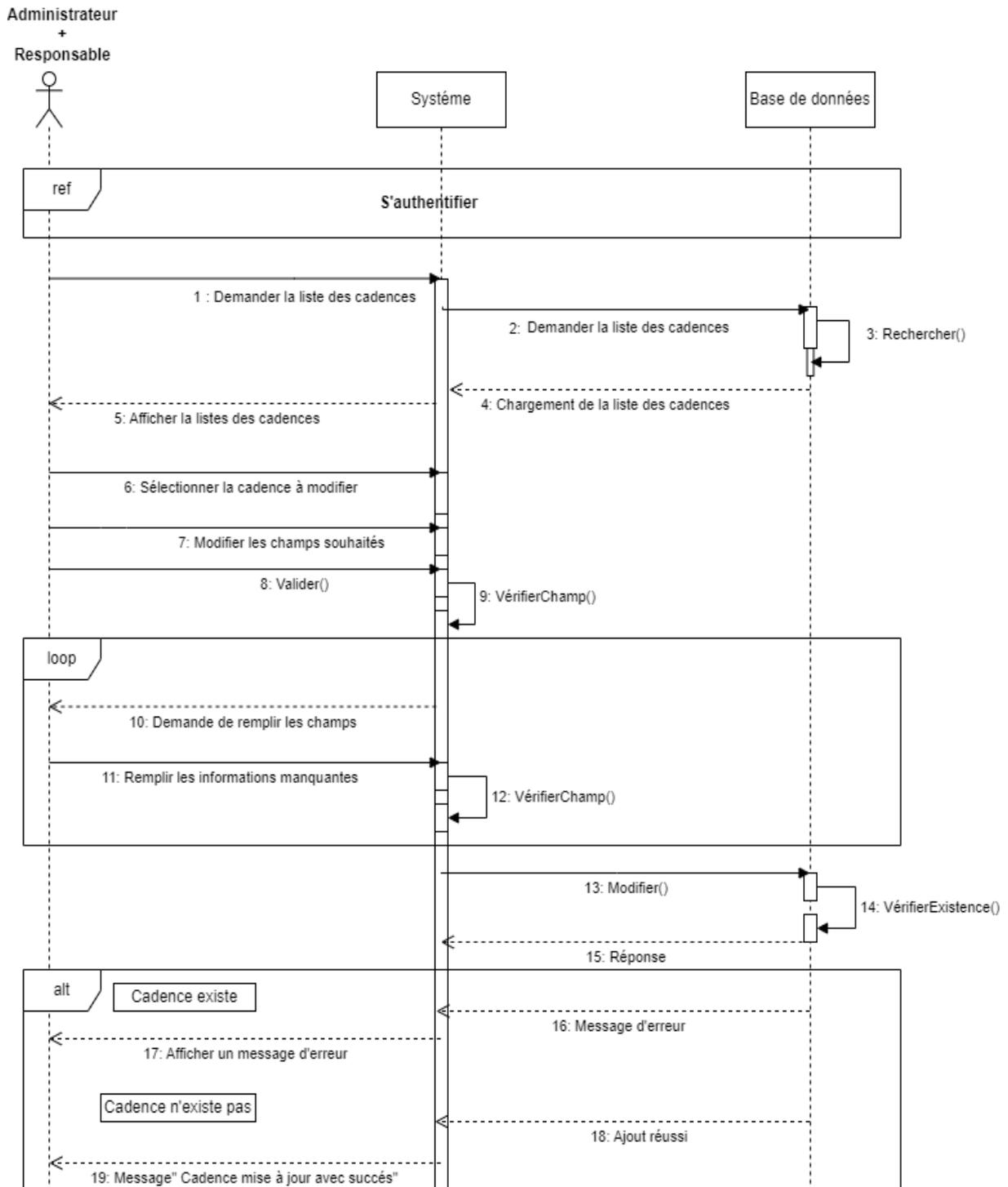


Figure 22:Diagramme de séquence « modifier une cadence »

3.5 Etude du sprint 4

Ce quatrième sprint s'étale sur 4 semaines, il comprend trois items qui sont :

- Gestion des productions
- Gestion des arrêts d'une production
- Imprimer un rapport

3.5.1 Diagramme des cas d'utilisation du sprint 4

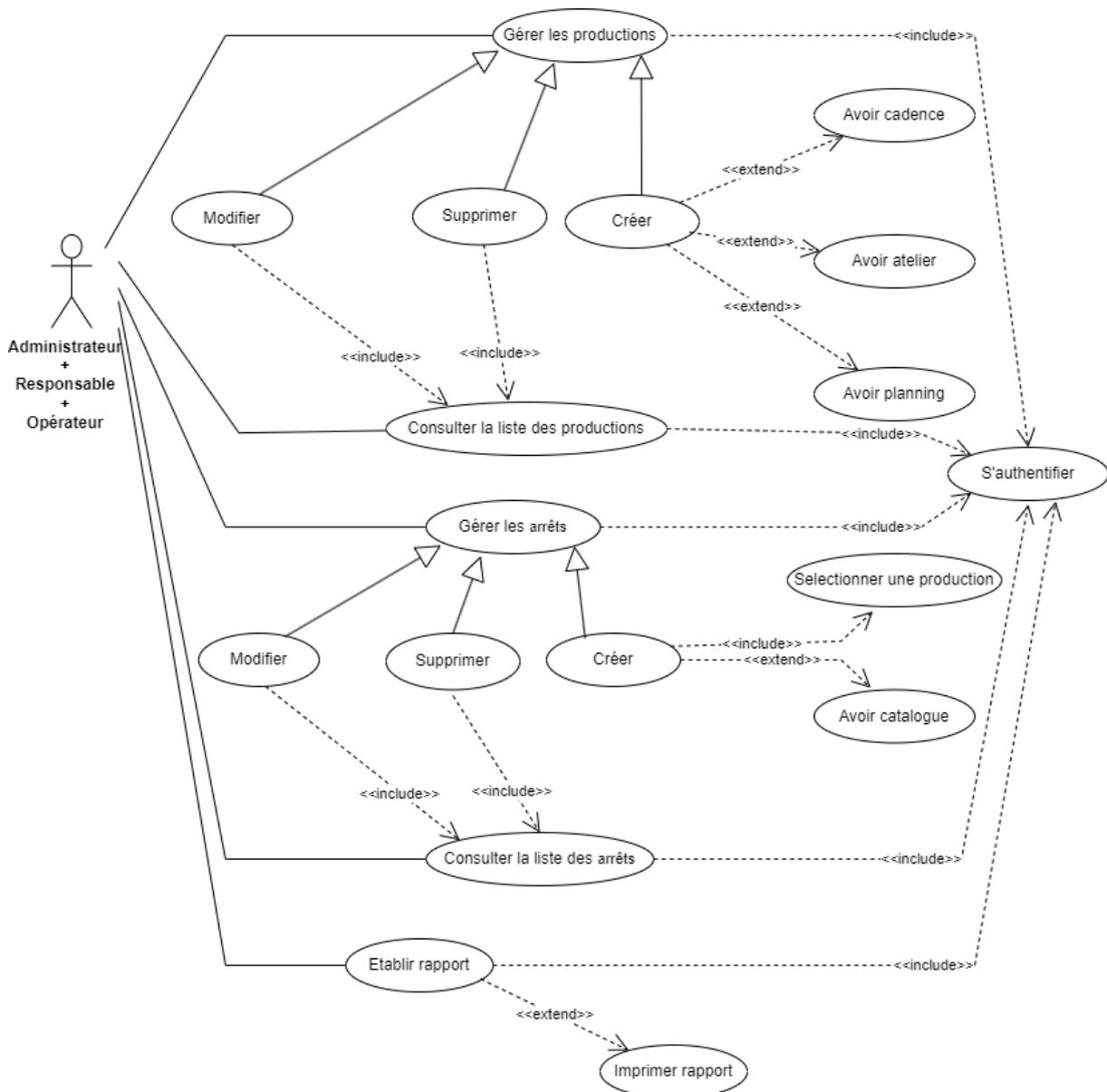


Figure 23: Diagramme des cas d'utilisation du sprint 4

3.5.2 Description textuelle du cas « Gérer les productions »

Cas d'utilisation	Gérer les productions
Acteur	Administrateur/Responsable /opérateur
Objectif	Gérer les productions
Précondition	L'utilisateur doit être authentifié en tant qu'administrateur, responsable ou opérateur
Postcondition	Les productions sont ajoutées, modifiées, supprimées ou consultées avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de gestion des productions. 2. L'utilisateur choisit une action (ajouter, rechercher, supprimer, modifier). 3. L'utilisateur entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'utilisateur
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 15:Description textuelle du cas d'utilisation « Gérer les production »

3.5.3 Description textuelle du cas « Gérer les arrêts de production »

Cas d'utilisation	Gérer les arrêts de production
Acteur	Administrateur/Responsable /Opérateur
Objectif	Gérer les arrêts de production
Précondition	L'utilisateur doit être authentifié en tant qu'administrateur, responsable ou opérateur
Postcondition	Les arrêts sont ajoutés, modifiés, supprimés ou consultés avec succès
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de gestion des arrêts. 2. L'utilisateur choisit une action (déclarer, supprimer, modifier). 3. L'utilisateur entre les détails nécessaires pour l'action choisie. 4. Le système traite la demande et met à jour la base de données. 5. Le système confirme l'action à l'utilisateur
Exception	-Si les informations fournies sont incomplètes ou incorrectes, le système affiche un message d'erreur et demande de corriger les informations.

Tableau 16:Description textuelle du cas d'utilisation « Gérer les arrêts de production »

3.5.4 Description textuelle du cas « Imprimer un rapport »

Cas d'utilisation	Imprimer un rapport
Acteur	Administrateur / Responsable / Opérateur
Objectif	Imprimer un rapport
Précondition	<p>-L'utilisateur doit être authentifié dans le système.</p> <p>-Le système doit avoir accès à une imprimante configurée.</p> <p>-Les données nécessaires pour générer le rapport doivent être disponibles dans le système.</p>
Postcondition	Un rapport papier est généré et disponible pour l'utilisateur.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la fonctionnalité de génération de rapports. 2.L'utilisateur sélectionne la date pour laquelle il souhaite générer le rapport. 3. Le système récupère les données de production pour la date sélectionnée. 4. Le système affiche les données de production à l'utilisateur. 5. L'utilisateur clique sur le bouton 'Imprimer'. 6.Le système envoie la commande d'impression à l'imprimante configurée. 7.L'imprimante reçoit la commande et imprime le rapport. 8.L'utilisateur récupère le rapport imprimé.

Tableau 17:Description textuelle du cas d'utilisation « Imprimer un rapport »

3.5.5 Diagramme de séquence du cas « Ajouter une production »

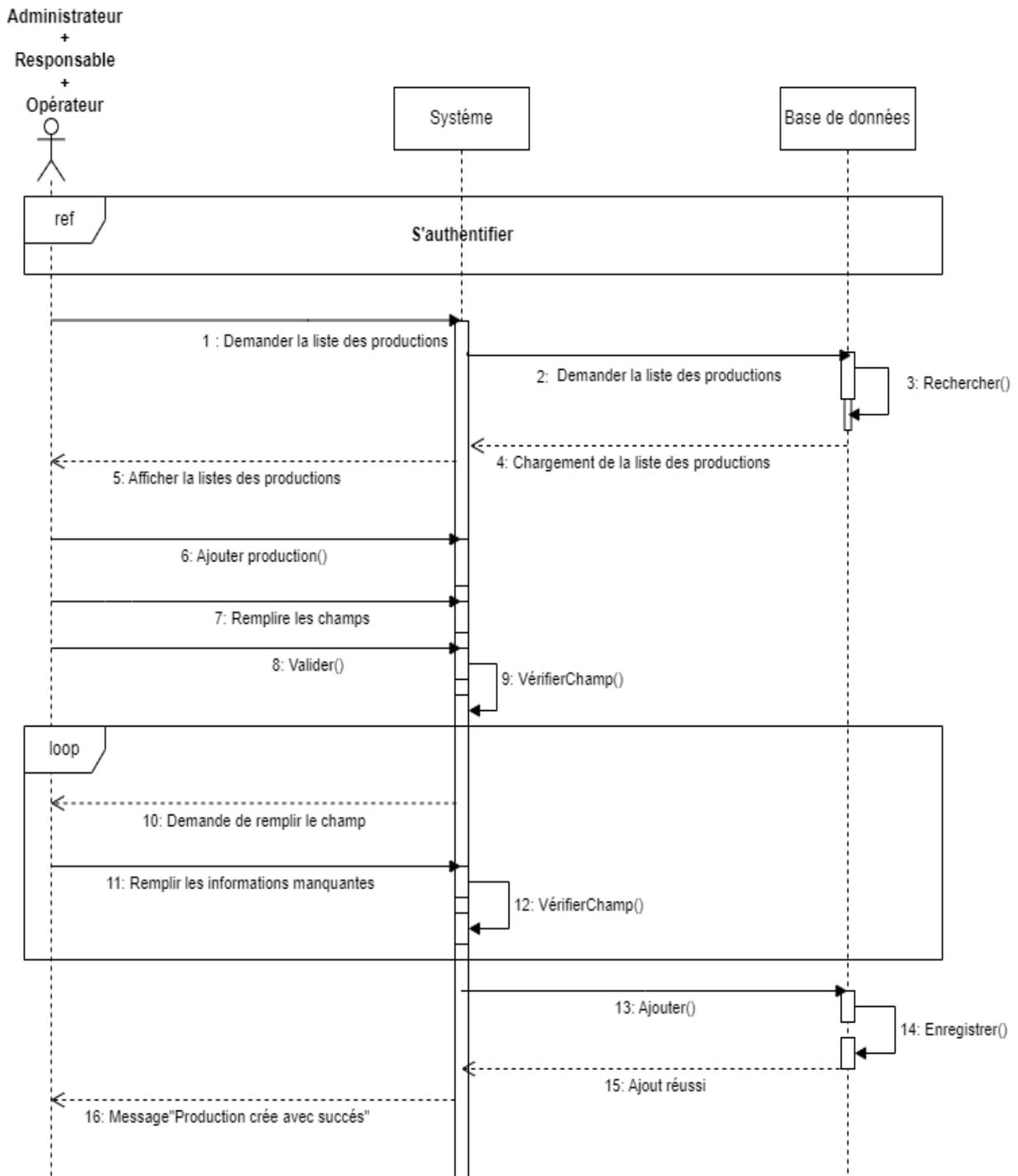


Figure 24:Diagramme de séquence du cas « Ajouter une production »

3.5.6 Diagramme de séquence du cas « Ajouter un arrêt »

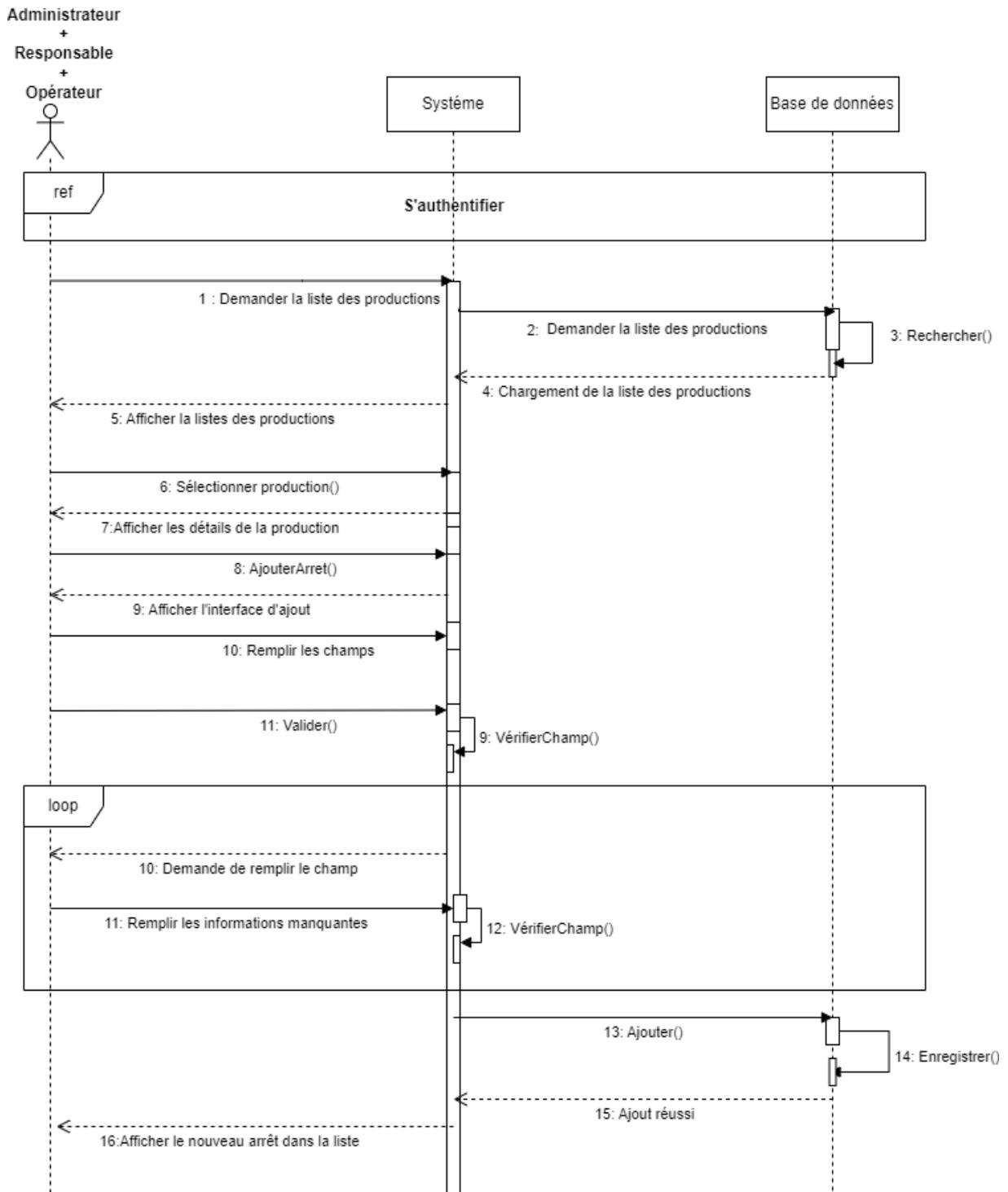


Figure 25:Diagramme de séquence du cas « Ajouter un arrêt »

3.5.7 Diagramme de séquence du cas « Imprimer un rapport »

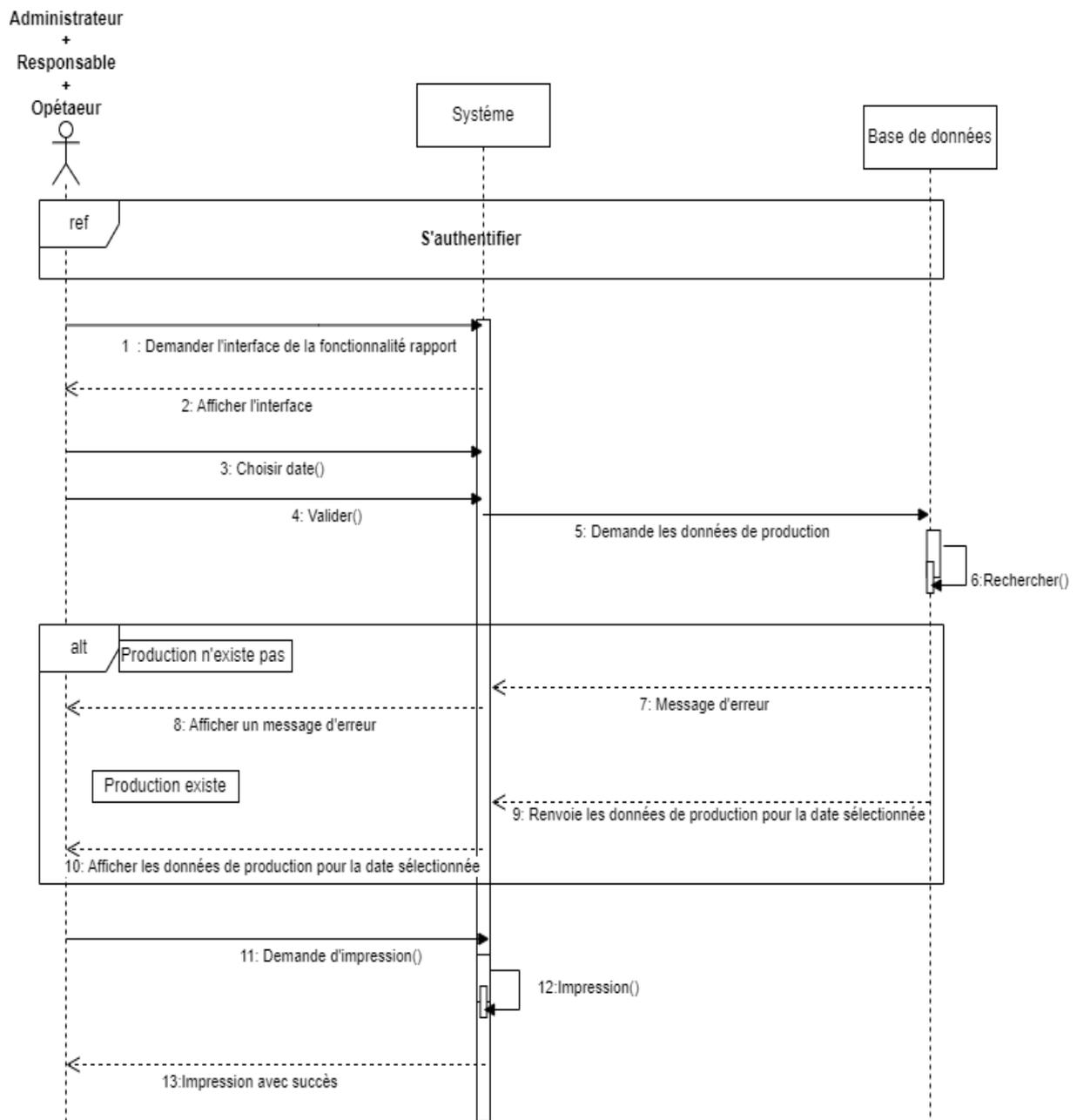


Figure 26: Diagramme de séquence du cas « Imprimer un rapport »

3.6 Conclusion

En conclusion, l'application de la méthode Scrum à notre projet nous a permis de structurer et de gérer efficacement le développement des différentes fonctionnalités. Chaque sprint a été clairement défini à travers des diagrammes de cas d'utilisation, des descriptions textuelles détaillées et des diagrammes de séquence précis, assurant ainsi une compréhension commune des objectifs et des tâches à accomplir.

CHAPITRE 4 : Conception de la base de données

4.1 Introduction

Après avoir mené une analyse approfondie au cours des sprints Scrum précédents, nous consacrons ce chapitre à l'exploration des règles de gestion. Ces règles sont essentielles pour la conception et l'établissement de divers artefacts, notamment le diagramme de classes, le dictionnaire de données et le modèle relationnel.

4.2 Règles de gestion

Voici les règles de gestion qui régissent les interactions et les dépendances entre les différentes entités, permettant ainsi l'élaboration du diagramme de classes de notre projet :

- Un utilisateur peut être l'administrateur, le responsable de production, ou un opérateur
- Un article peut être utilisé dans plusieurs recettes et dans la production.
- Une recette peut utiliser plusieurs articles.
- Une consommation récupère une recette de production.
- Un catalogue peut concerner plusieurs arrêts.
- Une cadence est liée à une ligne de production et à un article.
- Une production est liée à une ligne de production et à un article.
- Une production peut subir un ou plusieurs arrêts.
- Un arrêt est lié à une production et à un catalogue.
- Un atelier peut avoir plusieurs lignes de production.
- Un atelier peut avoir plusieurs plannings.
- Une ligne de production est liée à un atelier.
- Une ligne de production peut avoir plusieurs cadences.
- Un planning est lié à un atelier.

Ces règles de gestion définissent les interactions et les dépendances entre les différentes entités du système, assurant une organisation et une opération cohérentes.

4.3 Diagramme de classe

Un diagramme de classe est un type de diagramme de structure statique qui représente la structure statique d'un système logiciel en utilisant des classes et leurs relations. Il montre les classes du système, les attributs de ces classes, les méthodes qu'elles implémentent, ainsi que les relations entre les classes (comme l'héritage, l'agrégation, l'association, etc.). En bref, un diagramme de classe fournit une vue statique des entités et de leurs interactions dans un système logiciel [7].

Sa composition :

Classe : représente une description abstraite (à l'aide d'un rectangle) d'un groupe d'objets ayant les mêmes caractéristiques, tel que les classes utilisateur, production, atelier...etc.

Attributs : est un type d'information tel que (quantité_p) contenu dans une classe. Chaque attribut est caractérisé par son nom, sa visibilité, son type, sa valeur initiale et ses propriétés [7].

Voici le diagramme de classe de notre projet :

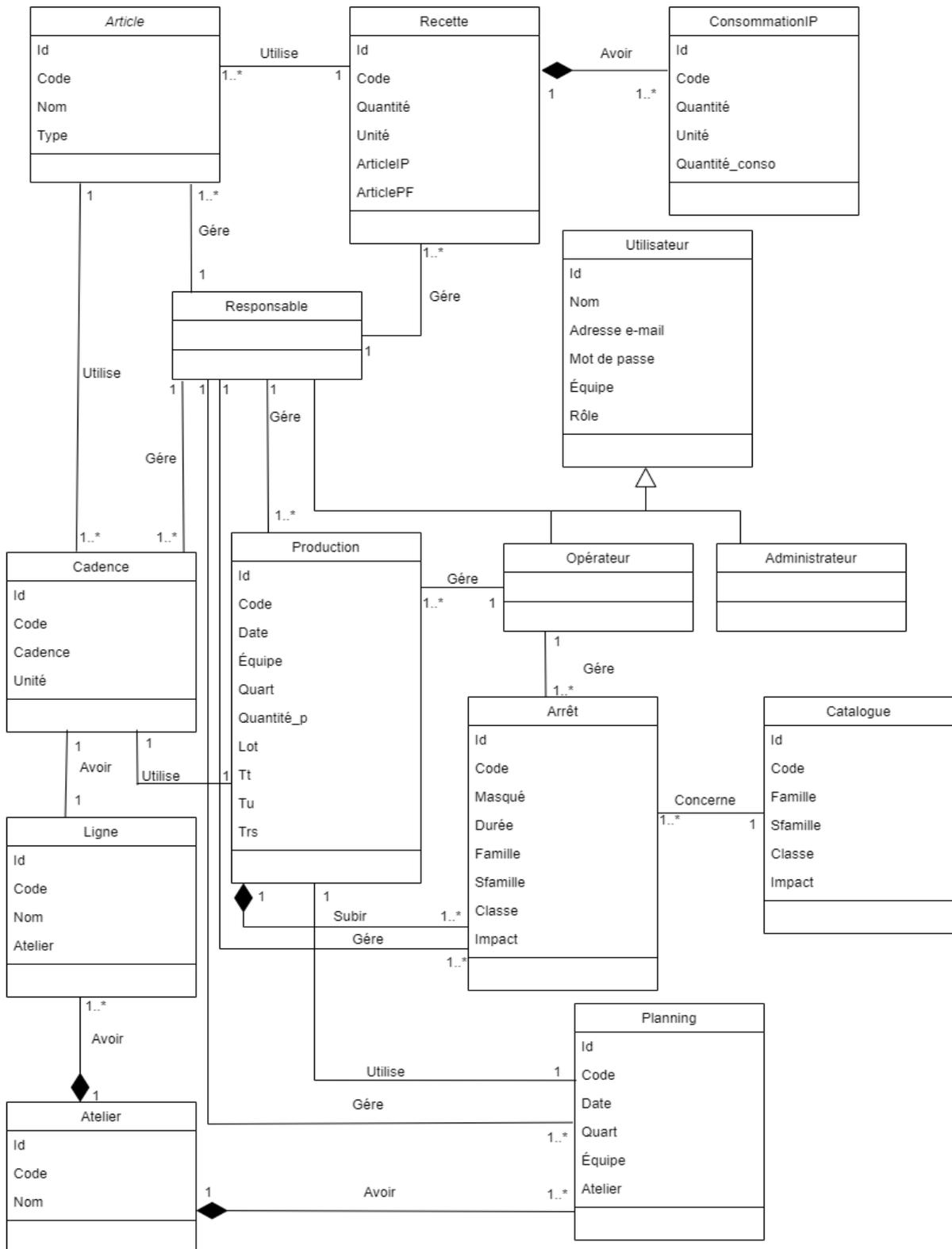


Figure 27:Diagramme de Classe

4.4 Dictionnaire de données

Le dictionnaire de données est un document qui regroupe toutes les données recensées, spécialisées et recueillies à conserver dans notre base de données. Pour chaque donnée [12], il indique :

- La codification : il s'agit d'un libellé désignant une donnée.
- La désignation : il s'agit d'une mention décrivant ce à quoi la donnée correspond.
- Le type de donnée : il s'agit du type de la donnée qui peut être :

Alphabétique (A) : La donnée est uniquement composée de caractères alphabétiques.

Numérique (N) : La donnée est composée uniquement de nombres (entiers ou réels).

Alphanumérique (AN) : La donnée peut être composée à la fois de caractères alphabétiques et numériques.

Date : La donnée est une date.

Booléen : La donnée est un booléen (vrai ou faux).

Classe	Codification	Désignation	Type
Production	id	-Identifiant d'une production	N
	code	-Code d'une production	AN
	date	-Date de la production	Date
	équipe	-L'équipe de l'utilisateur qui a déclaré la production	A

	quart	-Nombre d'heur de travaille	N
	unité	-Unité de production	A
	quantité_p	-Quantité produite	N
	tt	-Temps totale du travaille	N
	tu	-Temps utile	N
	trs	-Taux de rendement s....	N
Article	id	-Identifiant de l'article	N
	code	-Code d'article	AN
	nom	-Nom d'article	A
	type	-Type d'article	A
Atelier	id	-Identifiant de l'atelier	N
	code	-Code d'atelier	AN
	nom	-Nom d'atelier	AN
Ligne	id	-Identifiant de la ligne	N

	code	-Code de la ligne	AN
	nom	-Nom de la ligne	A
Catalog	id	-Identifiant du catalogue	N
	code	-Code du catalogue	AN
	famille	-Famille d'arrêt	A
	sfamille	-Sous famille d'arrêt	A
	class	-Class d'arrêt	A
	impact	-L'impact de l'arrêt	N
Arrêt	id	-Identifiant de l'arrêt	N
	code	-Code d'arrêt	AN
	durée	-Duré d'arrêt	N
	masqué	-Arrêt masqué ou non	BOOLÉEN
	famille	-Famille d'arrêt	A
	sfamille	-Sous famille d'arrêt	A

	class	-Class d'arrêt	A
	impact	-L'impact de l'arrêt	N
Planning	id	-Identifiant du planning	N
	code	-Code du planning	AN
	équipe	-Equipe a qui associer ce planning	A
	quart	-Nombre d'heur de travaille	N
ConsommationIP	id	-Identifiant de la consommationIP	N
	code	-Code de la consommationIP	AN
	quantité	-Quantité unitaire	N
	unité	-Unité	A
	quantité_conso	-Quantité consommer	N
Recette	id	-Identifiant de la recette	N
	code	-Code de la recette	AN
	quantité	-Quantité unitaire	N

	unité	-Unité	A
Utilisateur	id	-Identifiant de l'utilisateur	N
	nom	-Nom d'utilisateur	A
	email	-L'adresse email de l'utilisateur	A
	password	-Mot de passe de l'utilisateur	AN
	équipe	-L'équipe de travail à laquelle il appartient	A

Tableau 18: Dictionnaire des données

4.5 Schéma relationnel

4.5.1 Règles de passage du diagramme de classe au modèle relationnel

Lors du passage du diagramme de classes au modèle relationnel, certaines règles doivent être suivies pour garantir une représentation cohérente et correcte des données. Voici les règles générales à prendre en compte [13]:

- **Classe vers Table** : Chaque classe du diagramme de classes est généralement représentée par une table dans le modèle relationnel. Le nom de la table correspond généralement à celui de la classe.
- **Attributs vers Colonnes** : Chaque attribut d'une classe est généralement représenté par une colonne dans la table correspondante. Le nom de la colonne est généralement le même que celui de l'attribut.
- **Clé primaire** : L'identification de la clé primaire est essentielle pour chaque table. Dans le modèle relationnel, les attributs qui forment la clé primaire de la table sont définis comme clés primaires.

- **Relations un-à-un** : Si une relation un-à-un existe entre deux classes dans le diagramme de classes, elle peut être représentée en incluant la clé primaire d'une classe comme clé étrangère dans l'autre classe. Cela établit une relation entre les deux tables correspondantes.
- **Relations un-à-plusieurs** : Si une relation un-à-plusieurs existe entre deux classes, elle peut être représentée en incluant la clé primaire de la classe "un" comme clé étrangère dans la classe "plusieurs". Cela permet à la classe "plusieurs" de faire référence à la classe "un".
- **Relations plusieurs-à-plusieurs** : Si une relation plusieurs-à-plusieurs existe entre deux classes, une table de jointure supplémentaire est généralement nécessaire. Cette table de jointure contient les clés primaires des deux classes en tant que clés étrangères, ce qui permet d'établir une relation entre les enregistrements des deux tables.

Transformation de l'héritage

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

- **Décomposition par distinction** : Chaque sous-classe est transformée en une relation. La clé primaire de la super-classe migre dans la relation issue de la sous-classe et devient à la fois clé primaire et clé étrangère.
- **Décomposition descendante** : En cas de contrainte de totalité ou de partition sur l'association d'héritage, il est possible de ne pas traduire la relation issue de la super-classe. Tous ses attributs doivent alors migrer dans la ou les relations issues de la ou des sous-classes.
- **Décomposition ascendante** : La relation issue de la sous-classe est supprimée et ses attributs migrent dans la relation issue de la super-classe.

4.5.2 Le model relationnel des données

- **Utilisateur** (idUtilisateur, adresse e-mail, mot de passe, équipe,rôle,)
- **Atelier** (idAtelier,code, nom)
- **Ligne** (idLigne,code, nom ,#idAtelier)

- **Cadence** (idCadence, code, cadence, unité, #idLigne, #idAtelier, #idArticle)
- **Article** (idArticle, code, nom, type)
- **Recette** (idRecette, code, quantité, unité, articlePF_id, articleIP_id, #idArticle)
- **ConsommationIP** (idConsommationIP, code, quantité, unité, quantité_conso, #idRecette, #idArticle)
- **Planning** (idPlanning, code, date, équipe, quart, #idAtelier)
- **Arret** (idArret, code, masqué, duré, Famille, sFamille, classe, impact, #idProduction, #idCatalog)
- **Catalog** (idCatalog, code, Famille, sFamille, classe, impact)
- **Production** (idProduction, code, date, équipe, quart, unité, quantité_p, lot, tt, tu, trs, #idCadence)

4.6 Conclusion

En conclusion, les règles de gestion, le diagramme de classes, le dictionnaire de données et le modèle relationnel ont été des éléments essentiels dans l'élaboration de notre application. Leur utilisation a facilité une conception précise et organisée, assurant ainsi la cohérence et la conformité aux besoins définis.

CHAPITRE 5 : Réalisation et implémentation

5.1 Introduction

Ce dernier chapitre représente la phase de réalisation. Nous allons détailler le processus suivi pour la mise en œuvre de notre application. Nous commencerons par présenter les outils et les langages utilisés pour l'implémentation, puis nous présenterons l'application développée avec quelques captures d'écran des différentes interfaces.

5.2 Outils de développement

5.2.1 Environnement machine

5.2.1.1 ASUSLaptop

- Système d'exploitation : Microsoft Windows 11 Famille Unilingue 64 bits.
- Processeur : AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, 210 MHz.
- Mémoire : 8,00 Go

5.2.1.2 DELL

- Système d'exploitation : 64 bits, processeur x64
- Processeur : Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz 1.90 GHz
- Mémoire : 8,00 Go

5.2.2 Environnement logiciel

Tout projet informatique nécessite l'utilisation de technologies performantes pour garantir une mise en œuvre efficace des exigences définies lors des phases précédentes. Pour le développement de cette application, nous avons utilisé les outils logiciels suivants :

5.2.2.1 Visual Studio Code



Visual Studio Code est un éditeur de code source gratuit et open source, léger et puissant. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et possède un riche écosystème d'extensions pour d'autres langages (tels que C++, C#, Java, Python, PHP, Go et plus encore) et des runtimes (tels que .NET et Unity) [14].

5.2.2.2 Draw.io



draw.io

Draw io est une application en ligne gratuite accessible via un navigateur, permettant de créer des diagrammes et des organigrammes. Cet outil offre la possibilité de concevoir une variété de diagrammes, y compris des dessins vectoriels, de les enregistrer au format XML et de les exporter dans différents formats [15].

5.2.2.3 GitHub



GitHub Desktop est une application de bureau développée par GitHub qui facilite la gestion et la collaboration sur des projets hébergés sur la plateforme GitHub. C'est une interface graphique conviviale qui simplifie les tâches courantes liées à Git, telles que la création, le clonage, la gestion des branches, le suivi des modifications et la gestion des conflits [16].

5.2.3 Environnement de programmation

Pour développer notre application web, nous avons utilisé des différents langages de programmation :

5.2.3.1 SQLite



SQLite est une bibliothèque écrite en langage C qui propose un moteur de base de données relationnelle accessible par le langage SQL. SQLite implémente en grande partie le standard SQL-92 et des propriétés ACID. Contrairement aux serveurs de bases de données traditionnels, tels que MySQL, MariaDB ou PostgreSQL, sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme [17].

5.2.3.2 PHP



PHP est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur web, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook et Wikipédia. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

5.2.3.3 HTML



HTML constitue le langage de base d'un site internet. Ce n'est pas un langage de programmation mais un langage de balisage qui permet d'écrire de l'hypertexte et ainsi de définir la structure sémantique d'une page web. L'hypertexte désigne un ou plusieurs documents qui contiennent des unités d'informations (textes, images, vidéos...) liées ensemble à travers des hyperliens.

5.2.3.4 CSS



CSS est un langage spécifique au web, fréquemment employé comme complément du langage HTML, et dont la fonction est de former des feuilles de styles chargées de la mise en forme des documents web. Il gère l'esthétique (couleur, typographie) et diverses fonctionnalités.

5.2.3.5 JavaScript



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

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

5.2.4 Frameworks utilisés

Un Framework est, comme son nom l'indique en anglais, un cadre de travail. Son objectif principal est de simplifier le travail des développeurs en leur offrant une architecture 'prête à l'emploi'. Dans notre projet, nous avons utilisé les frameworks suivants :

5.2.4.1 Laravel 11



Laravel 11 est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel 11 introduit une structure d'application minimale, utilisant SQLite par défaut [18].

5.2.4.2 Livewire



Livewire est un Framework full-stack pour Laravel qui simplifie la création d'interfaces dynamiques, sans quitter le confort de Laravel [19].

5.2.4.3 Bootstrap



Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

5.3 Projection de quelques interfaces de notre application

Dans cette section nous allons présenter le contenu des principales interfaces de notre application développée pour la gestion de production de Cevital Agroalimentaire

5.3.1 Authentification

La figure suivante représente l'interface d'authentification de notre application web



Figure 28:Interface d'authentification

5.3.2 Interface de tableaux de bord

Si l'utilisateur s'est connecté en tant qu'administrateur la figure ci-dessous apparaîtra

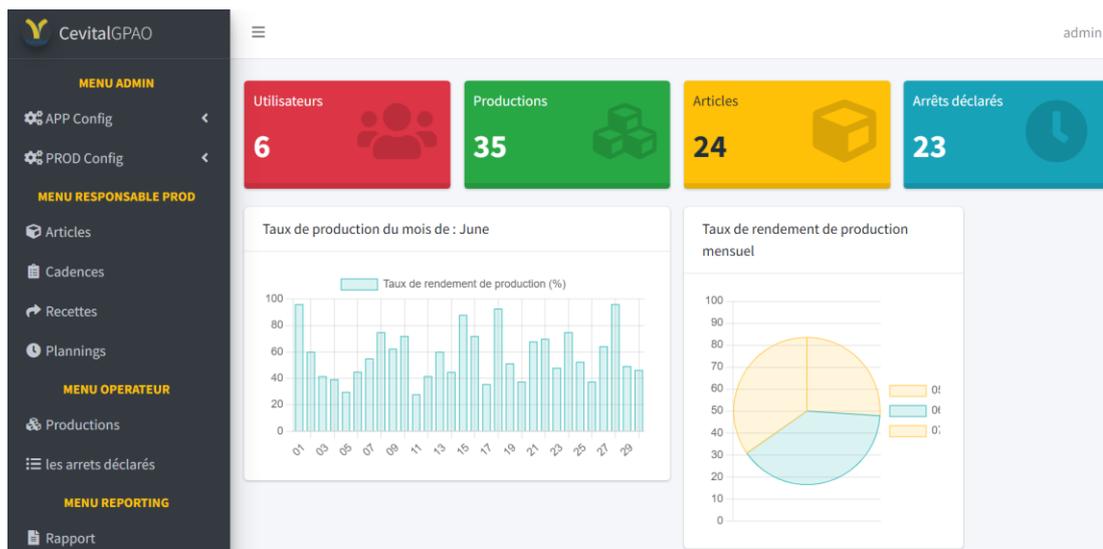


Figure 29:Tableau de bord administrateur

Si l'utilisateur s'est connecté en tant que responsable la figure ci-dessous apparaîtra

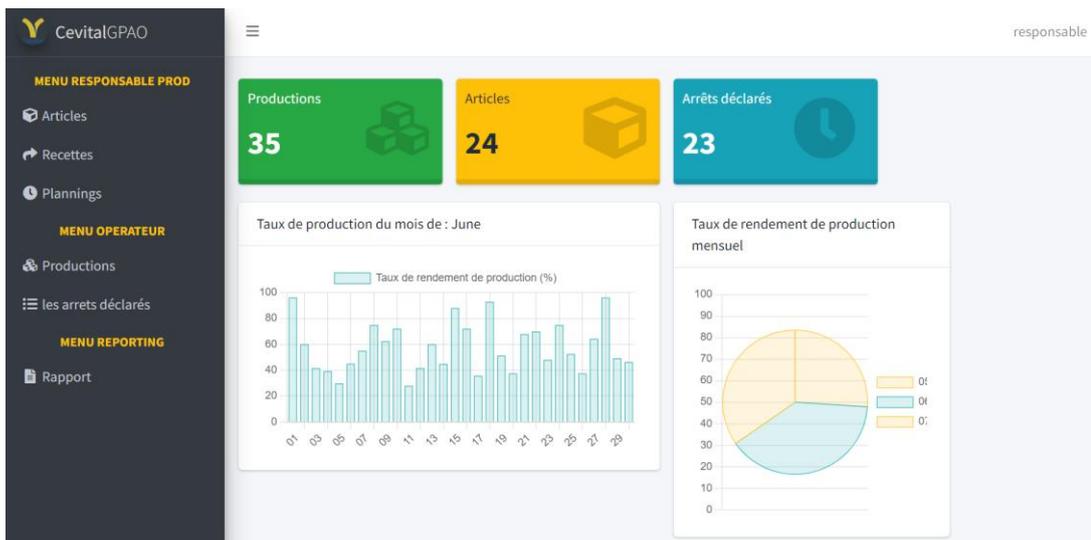


Figure 30:tableau de bord du responsable

Si l'utilisateur s'est connecté en tant qu'opérateur, la figure ci-dessous apparaîtra

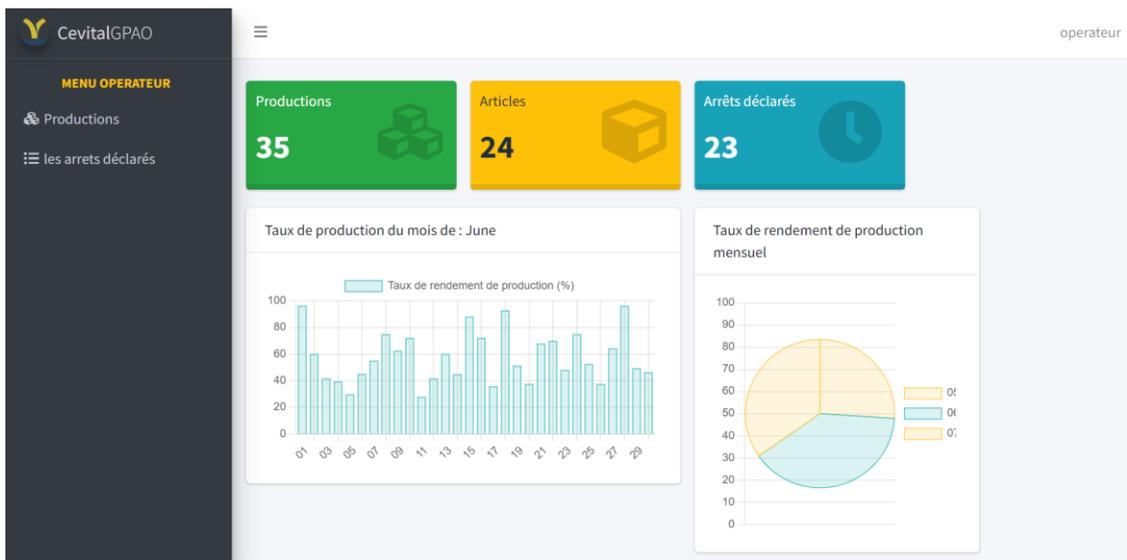


Figure 31:Tableau de bord de l'opérateur

5.3.3 Gestion des comptes utilisateurs et attribution des rôles

Cette figure représente l'interface de création d'un utilisateur avec attribution d'un rôle.

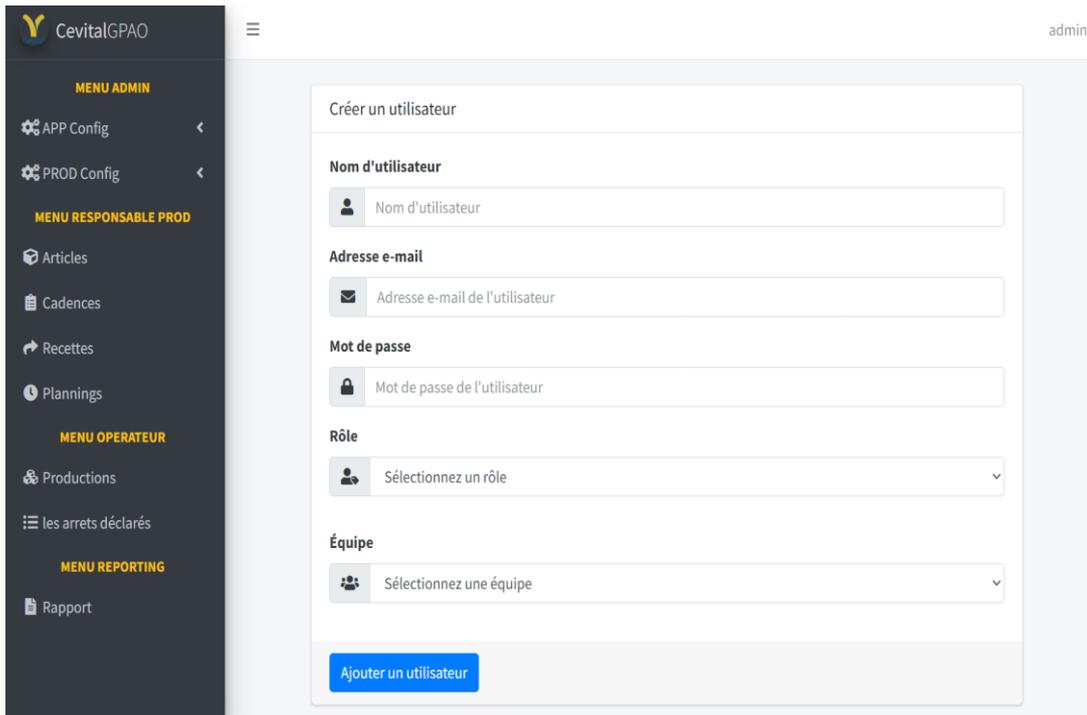


Figure 32: Interface de création d'un utilisateur

Cette figure représente la liste des utilisateurs

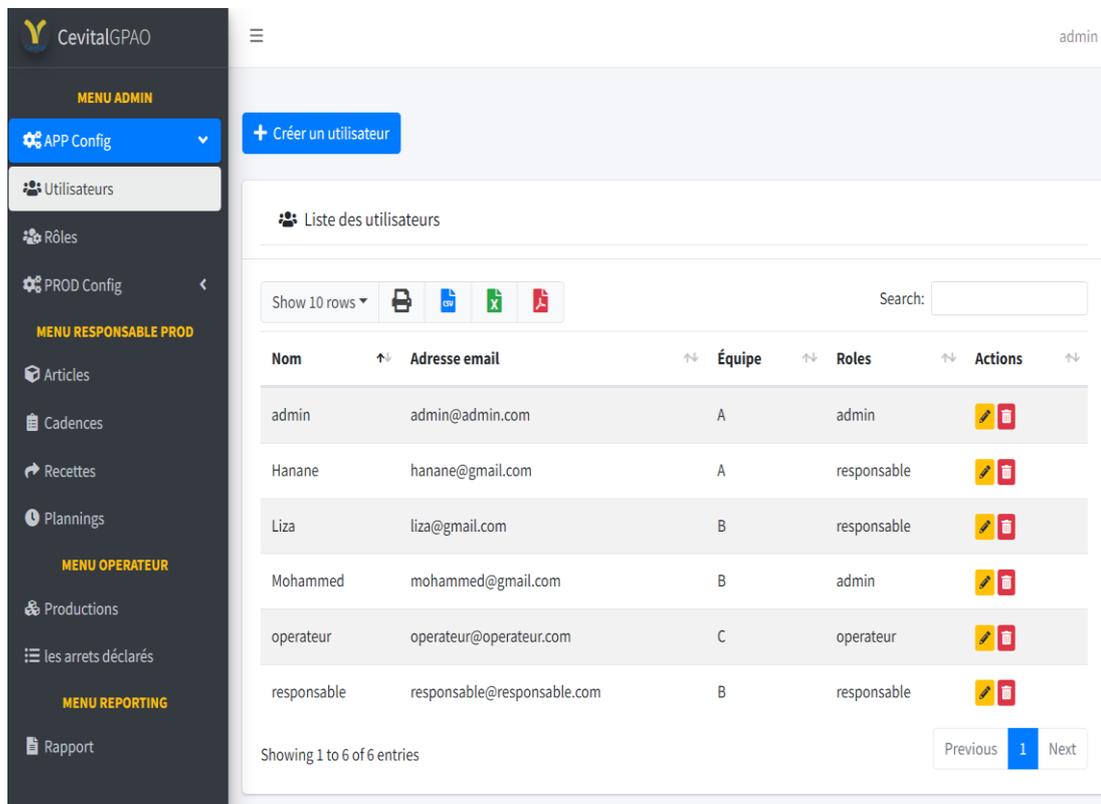


Figure 33: Liste des Utilisateurs

Cette figure représente l'interface de modification d'un utilisateur

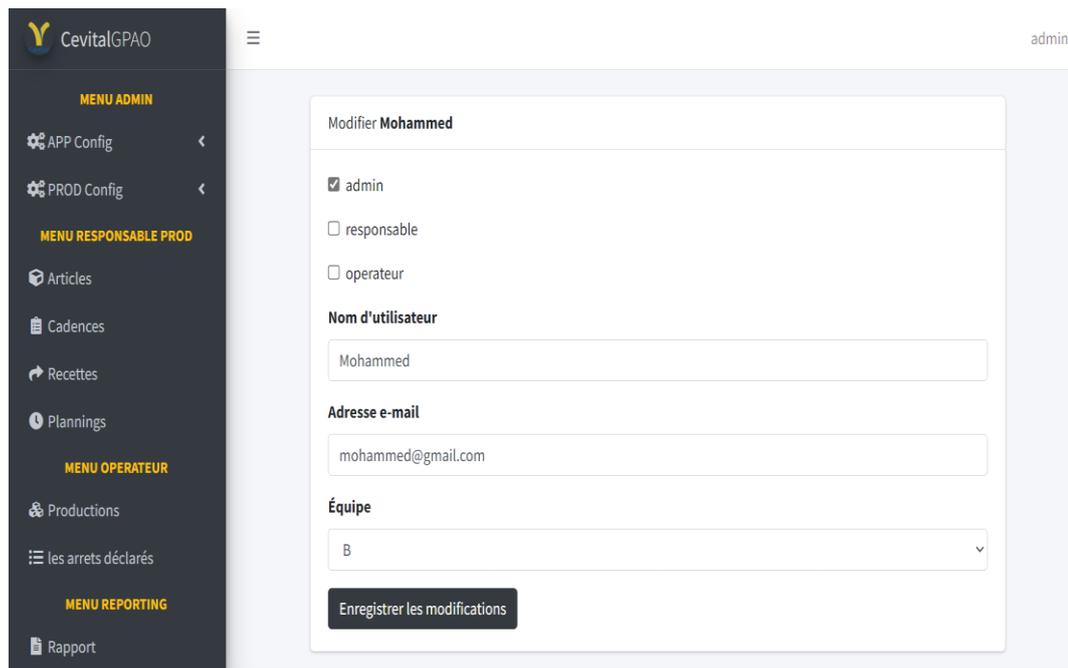


Figure 34: Interface de modification d'un utilisateur

5.3.4 Gestion des ateliers, lignes, catalogues des arrêts

Cette figure représente l'interface de gestion des ateliers.

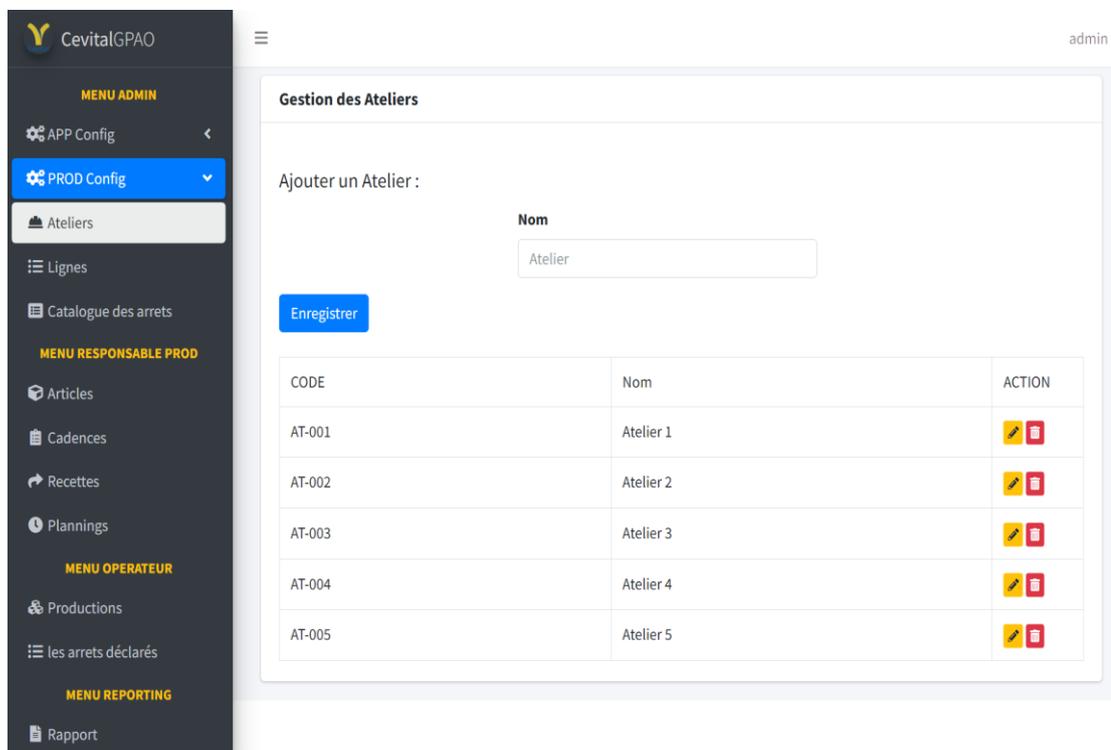


Figure 35: Interface de gestion des ateliers

Cette figure représente l'interface de gestion de catalogue d'arrêts.

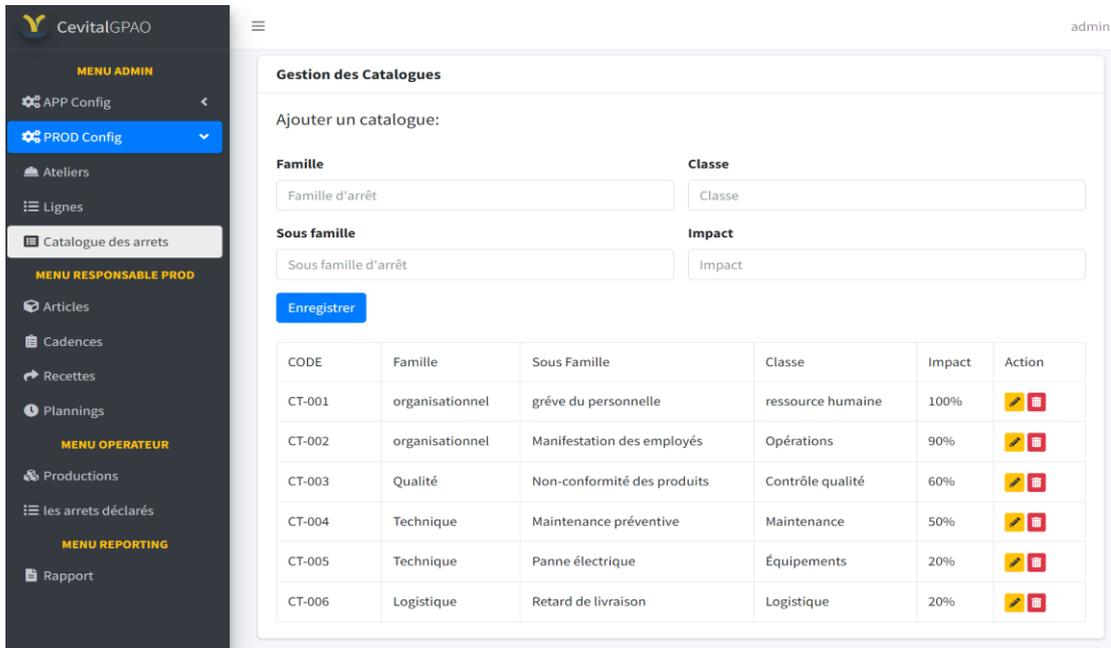


Figure 36: Interface de gestion des catalogues d'arrêts

Cette figure représente l'interface de gestion des lignes.

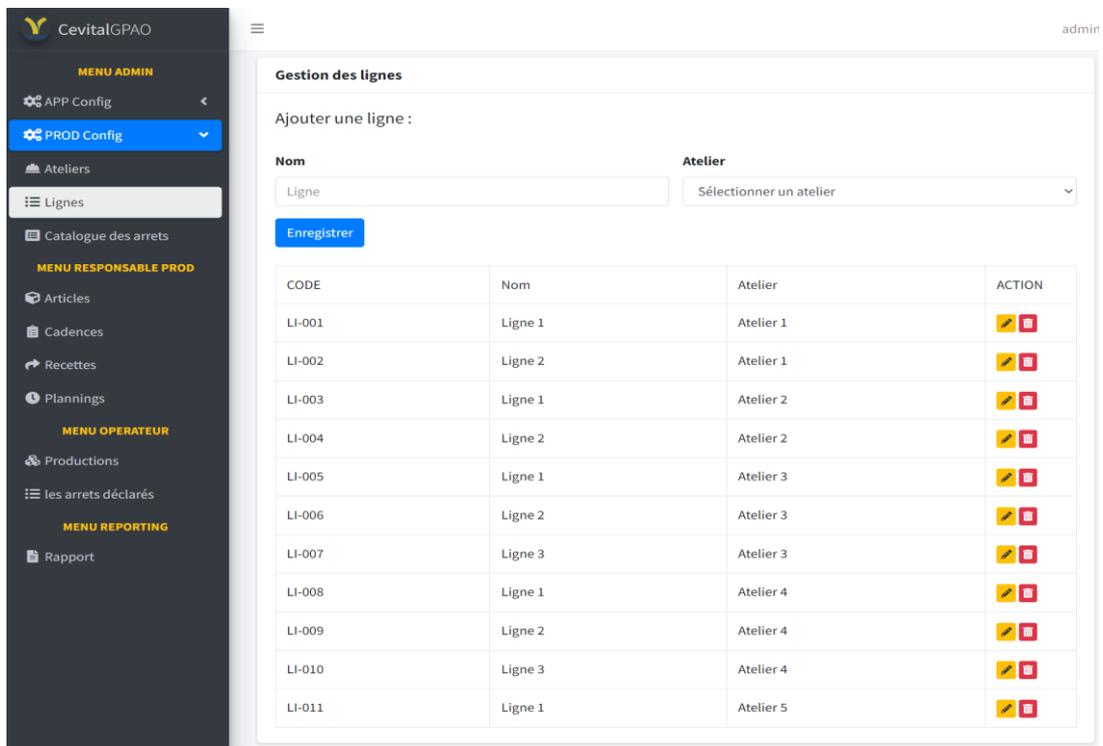


Figure 37: Interface de gestion des ligne

5.3.5 Gestion des articles

Cette figure représente la liste des articles

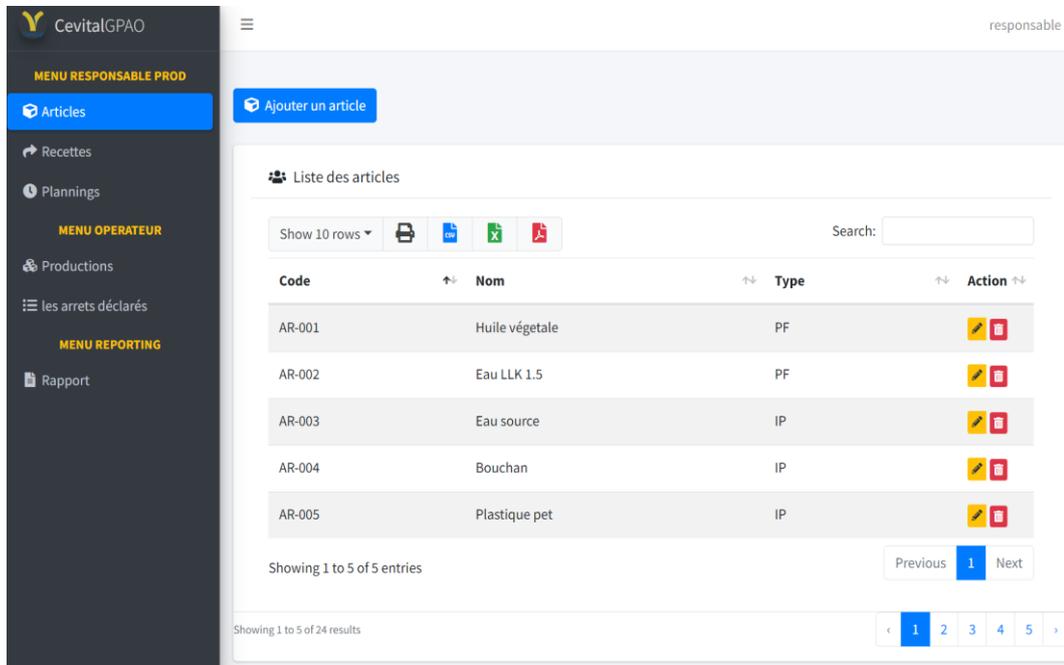


Figure 38: Liste des articles

Cette figure représente l'interface d'ajout d'un article

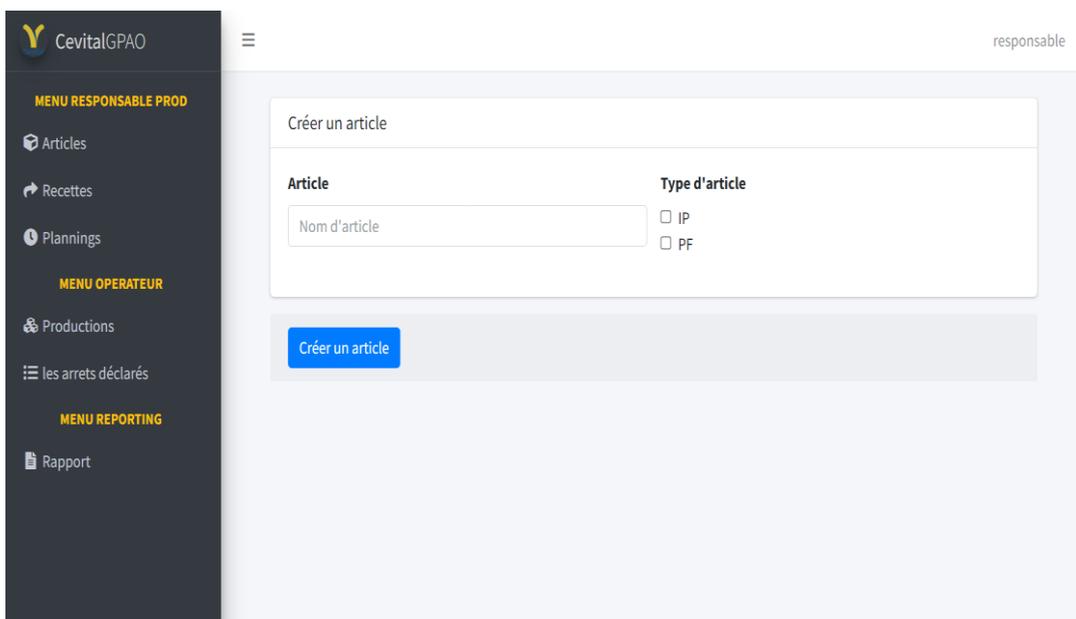


Figure 39: Interface d'ajout d'article

5.3.6 Interface de gestion des recettes

Cette figure représente la liste des recettes

The screenshot displays the 'Liste des recettes' (Recipe List) interface. On the left is a dark sidebar with navigation options: 'Articles', 'Recettes' (highlighted), 'Plannings', 'Productions', 'les arrêts déclarés', and 'Rapport'. The main content area features a top bar with 'Ajouter recette' and 'responsable'. Below is a table with the following data:

Code	Article PF	Article IP	Quantité	Unité	Actions
RC-001	Huile végétale	Bouchan	1	Bouchan	[Edit] [Delete]
RC-002	Huile végétale	Plastique pet	1	Bouteille	[Edit] [Delete]
RC-003	Huile végétale	Etiquette	10	Cm	[Edit] [Delete]
RC-004	Huile végétale	Acide oléique	250	G	[Edit] [Delete]
RC-005	Huile végétale	Acide linoléique	320	G	[Edit] [Delete]
RC-006	Eau LLK 1.5	Eau source	1.5	L	[Edit] [Delete]
RC-007	Eau LLK 1.5	Bouchan	1	Bouchan	[Edit] [Delete]
RC-008	Eau LLK 1.5	Plastique pet	1	Bouteille	[Edit] [Delete]
RC-009	Eau LLK 1.5	Etiquette	10	Cm	[Edit] [Delete]
RC-010	Jus 1L	Eau source	700	MI	[Edit] [Delete]

At the bottom of the table, it says 'Showing 1 to 10 of 25 entries' and includes pagination controls for 'Previous', '1', '2', '3', and 'Next'.

Figure 40: Liste des recettes

Cette figure représente l'interface d'ajout d'une recette

The screenshot shows the 'Créer une recette' (Create a recipe) form. It includes the following fields and controls:

- Article PF:** A dropdown menu with the placeholder text 'Sélectionnez un article PF'.
- Article IP:** A dropdown menu with the placeholder text 'Sélectionnez un article IP'.
- Quantité:** A text input field with the placeholder text 'Quantité'.
- Unité:** A text input field with the placeholder text 'Unité'.
- Enregistrer:** A blue button to save the recipe.

The sidebar and top navigation bar are identical to the previous screenshot.

Figure 41: Interface d'ajout d'une recette

5.3.7 Interface de gestion des plannings

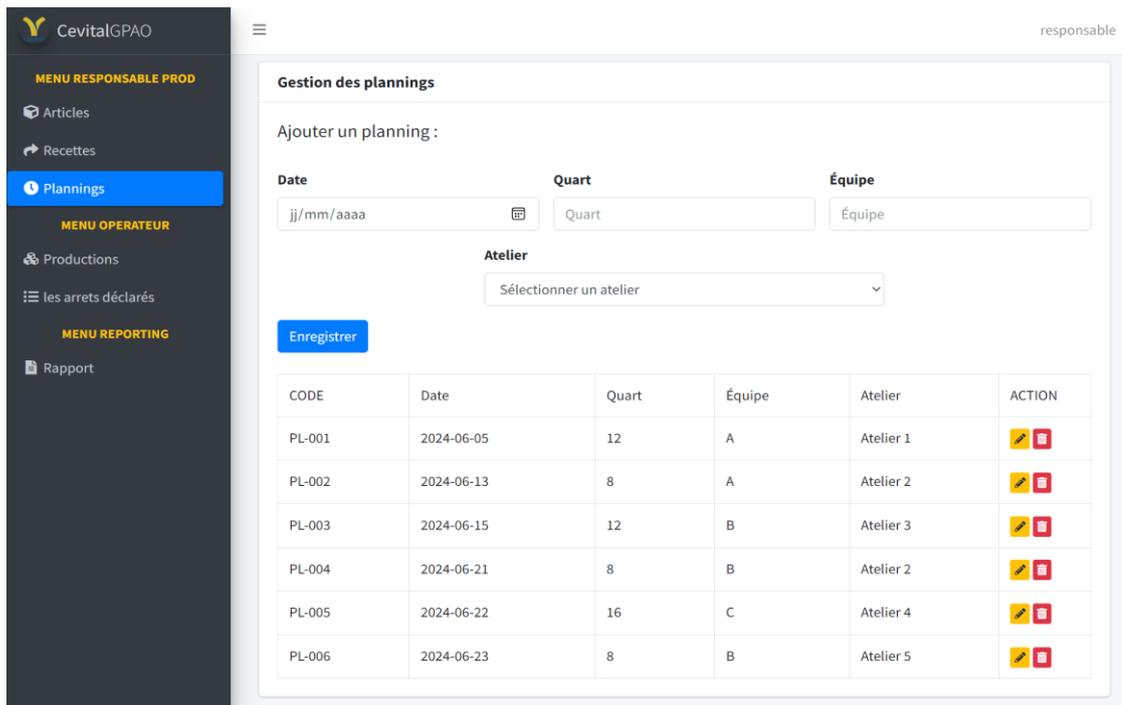


Figure 42: Interface de gestion des plannings

5.3.8 Interface de gestion des productions

Cette figure représente la liste des productions

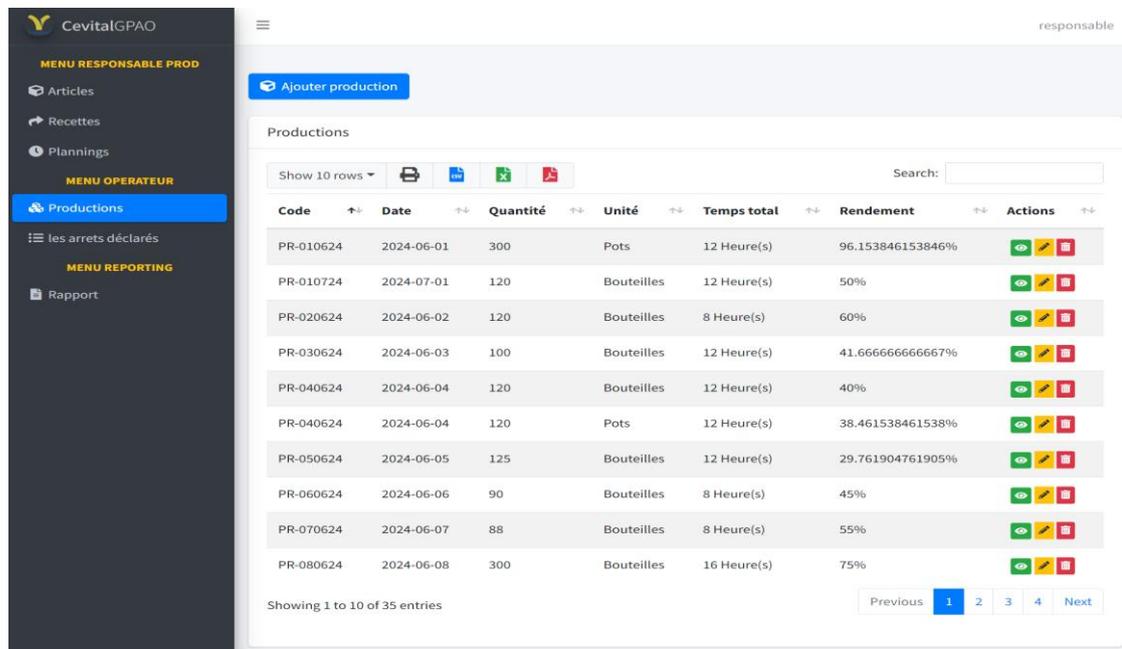


Figure 43: Liste des productions

Cette figure représente l'interface d'ajout d'une production

Créer une production

Date **Quart** **Équipe**

Atelier **Ligne de production** **Article PF**

Quantité **Unité de production** **Lot**

[Créer une production](#)

Figure 44: Interface d'ajout d'une production

Cette figure représente l'interface de show détails d'une production

Détails de la production

Date	2024-06-01	Quantité	300	Unité de production	Pots
Lot	11	Atelier	Atelier 4	Ligne	Ligne 1
Quart	12	Équipe	A	Article	Mayonnaise
Temps total	12	Temps util	11.538461538462	Rendement	96.153846153846

Arrêts

Show 10 rows Search:

Article IP	Quantité unitaire	Quantité consommé	Unité
Conservateur alimentaire	100	30000	G
Etiquette	10	3000	Cm
Moutarde	80	24000	G
Sel	5	1500	G

Showing 1 to 4 of 4 entries

Figure 45: Interface de show détails d'une production

Cette figure représente l'interface d'ajout d'arrêt a une production dans show détails

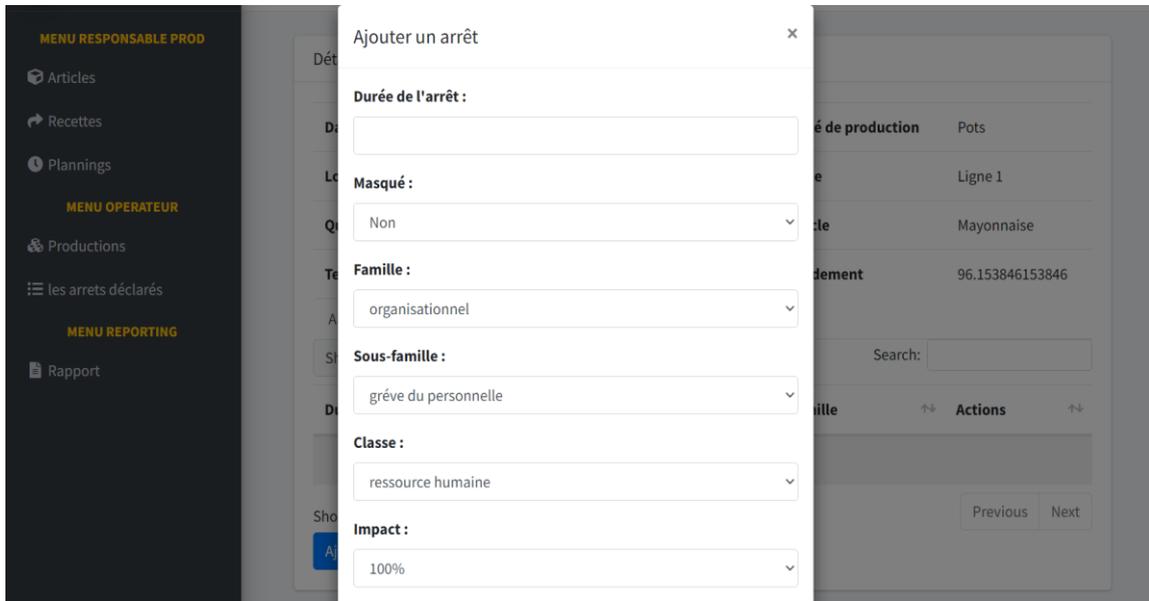


Figure 46:Interface d'ajout d'arrêt

Cette figure représente la liste des arrêts déclarés de toutes les productions qui existe

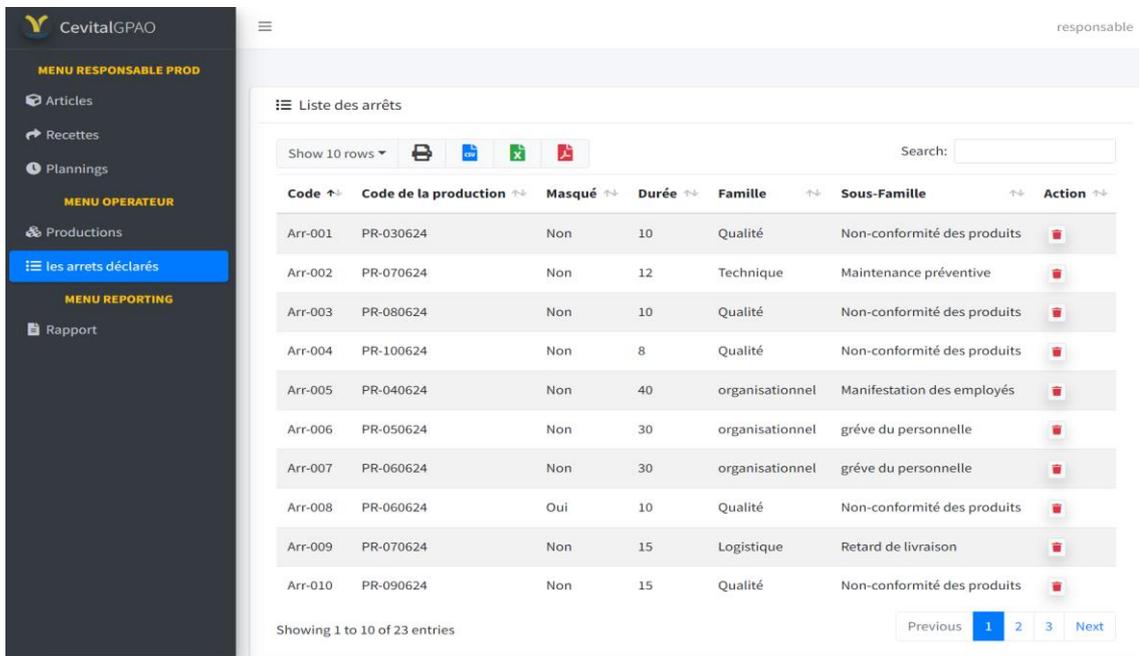


Figure 47:Liste des arrêts déclarés

5.3.9 Interface d'établir et d'imprimer un rapport

Cette figure représente l'interface d'établir un rapport de production

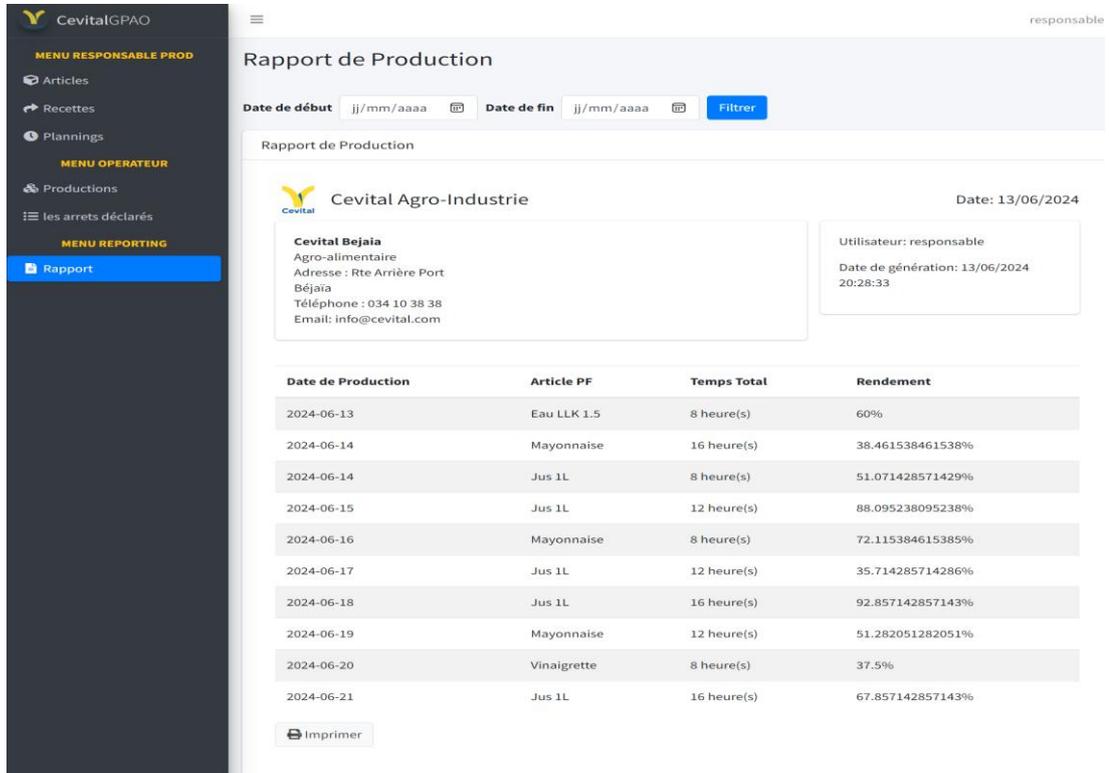


Figure 48: Interface d'établir un rapport

Cette figure représente l'interface d'impression d'un rapport

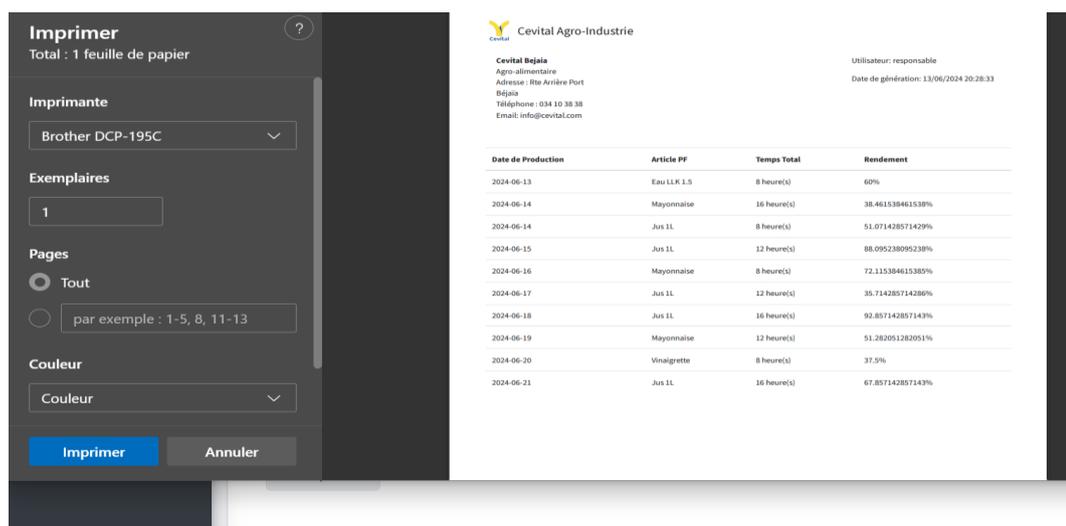


Figure 49: Interface d'impression d'un rapport

5.3.10 Une démonstration de la responsivité

Notre application est conçue pour offrir une expérience utilisateur optimale grâce à sa responsivité. Peu importe l'appareil utilisé, que ce soit un smartphone, une tablette ou un ordinateur, notre interface s'adapte parfaitement à toutes les tailles d'écran. Voici quelques captures d'écran qui illustrent la réactivité et la fluidité de notre application.

5.3.10.1 Smartphone

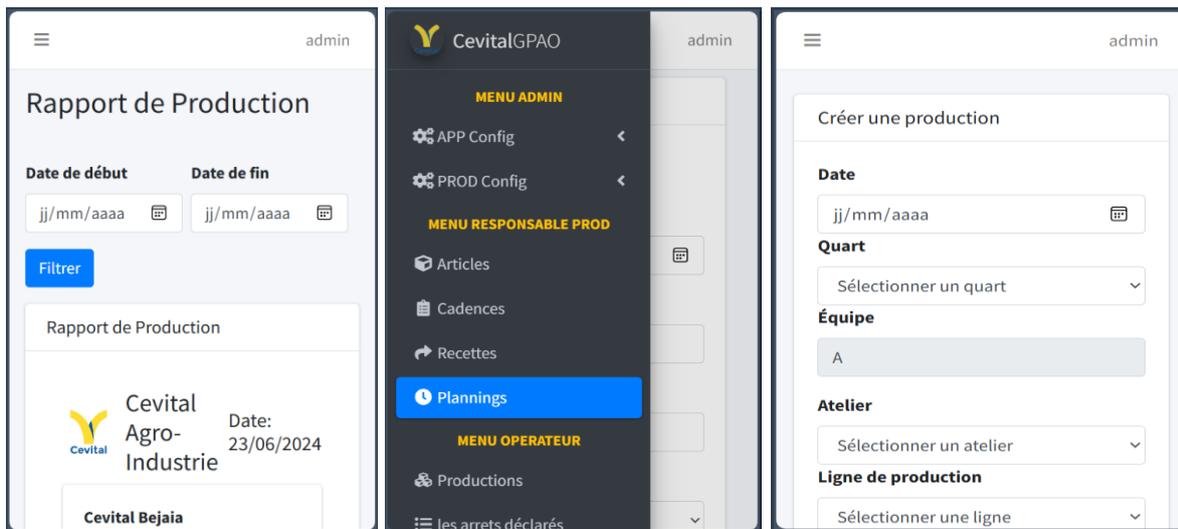


Figure 50: Vue de l'application sur Smartphone

5.3.10.2 Tablette

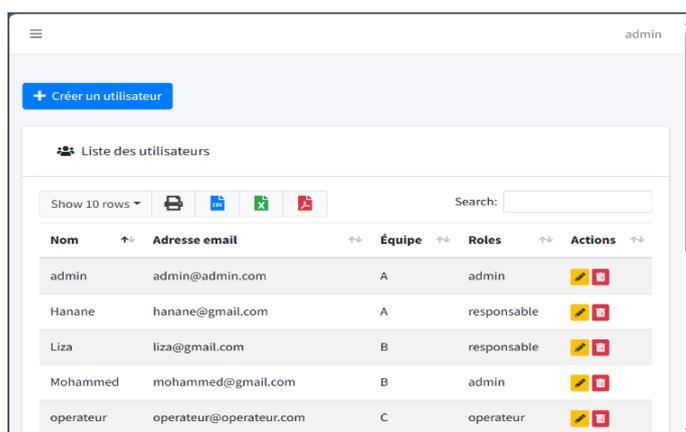


Figure 51: Vue de l'application sur une Tablette

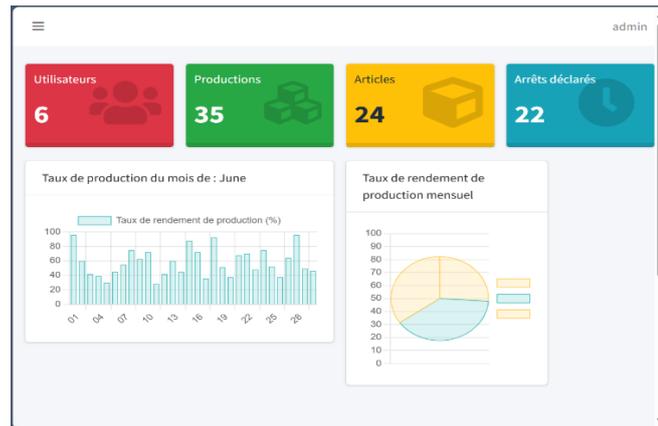


Figure 52: Vue de l'application sur une Tablette

5.4 Conclusion

Dans ce chapitre, nous avons exploré les aspects pratiques liés à la création de notre application web. Nous avons détaillé les outils et les langages de développement indispensables pour assurer le bon fonctionnement de l'application. De plus, nous avons présenté des exemples concrets des interfaces graphiques de notre application "CevitalGPAO", illustrant visuellement les fonctionnalités implémentées.

Conclusion générale et perspectives

Le développement de l'application web de gestion de production pour Cevital Agroalimentaire de Béjaïa représente une avancée significative dans l'optimisation des processus de production de l'entreprise. En identifiant les lacunes majeures de l'application de bureau actuelle, nous avons pu concevoir une solution plus flexible, accessible et performante.

Notre application, nommée **CevitalGPAO**, intègre des fonctionnalités essentielles telles que le calcul du Taux de Rendement Synthétique (TRS), l'impression et l'exportation de rapports de production, ainsi qu'une gestion centralisée et sécurisée des données de production. La migration vers une architecture web assure une accessibilité accrue et une meilleure réactivité, facilitant ainsi l'utilisation par les opérateurs et les responsables de production.

Grâce à l'intégration de la planification de la production, du suivi de la cadence des lignes de production, de la gestion des recettes, des arrêts et du personnel, notre application offre une vue d'ensemble précise et en temps réel des opérations. Cela permet non seulement d'augmenter l'efficacité et la productivité, mais aussi de renforcer la sécurité et la continuité des opérations.

En conclusion, **CevitalGPAO** se positionne comme un outil indispensable pour Cevital Agroalimentaire, répondant aux besoins actuels et futurs de gestion de production. Nous espérons que cette application contribuera de manière significative à la performance et à la compétitivité de l'entreprise, en permettant une gestion de production plus efficace, sécurisée et adaptable.

La réalisation de **CevitalGPAO** a été une expérience très intéressante et enrichissante pour nous. Elle nous a permis d'appliquer les connaissances et compétences que nous avons acquises tout au long de notre parcours, notamment dans les domaines de la conception, de la programmation et de l'application de méthodes de travail. Bien que cela ait parfois été

difficile, nous avons pu apprendre de nouvelles choses qui seront certainement utiles dans notre future carrière professionnelle.

Les perspectives d'amélioration de cette application sont prometteuses. Nous envisageons de renforcer les tableaux de bord et les alertes en temps réel pour améliorer la réactivité des équipes de production dans la prise de décision. Par exemple, en intégrant cette application avec le système de gestion des ressources humaines de l'entreprise, nous visons à augmenter l'efficacité des processus liés à la gestion du personnel, soutenant ainsi les objectifs de performance et de compétitivité de Cevital Agroalimentaire.

Références

[1] <https://www.cevital.com/lhistoire-du-groupe/>. [Accès le 05 Mars 2024].

[2] <https://www.cevital.com/intro-agro-industrie-et-distribution/>. [Accès le 05 Mars 2024].

[3] <https://www.cevital.com/cevital-agro-industrie/>. [Accès le 05 Mars 2024].

[4] <https://fr.scribd.com/document/730676325/organigramme-cevital>. [Accès le 06 Mars 2024].

[5] S. Laoyan, «What is Agile methodology?», Oct 2022. <https://asana.com/fr/resources/agile-methodology>. [Accès le 03 Mai 2024].

[6] J. S. K. Schwaber, The Scrum Guide, Nov 2020.

[7] I. M. Xavier BLANC, UML2 pour les développeurs, Eyrolles, 2006..

[8] R. Hernandez., The Model View Controller Pattern MVC Architecture and Frame, Avr 2021.

[9] C. BELLEIL, UML2.0 Diagramme de contexte, Nantes: université de Nantes, 2003.

[10] P. ROQUES, Les cahiers du programmeur UML2 Modéliser une application web, Eyrolle, 2007.

[11] K. Colard, Une charte graphique : qu'est-ce que c'est et à quoi ça sert?, 2020.

[12]<https://ineumann.developpez.com/tutoriels/merise/initiation-merise/>. [Accès le 25 Mai 2024].

[13] <https://stph.scenari-community.org/idl-bd/idl-bd4.pdf>. [Accès le 13 Mars 2024].

[14] https://fr.wikipedia.org/wiki/Visual_Studio_Code. [Accès le 02 Juin 2024].

[15]<https://www.tice-education.fr/tous-les-articles-et-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne>. [Accès le 02 Juin 2024].

[16] <https://www.lemagit.fr/definition/GitHub>. [Accès le 02 Juin 2024].

[17]<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203607-sqlite-definition/>. [Accès le 02 Juin 2024].

[18]<https://laravel.com/>. [Accès le 02 Juin 2024].

[19] <https://laravel-livewire.com/>. [Accès le 02 Juin 2024].

Résumé

Ce travail a été réalisé dans le cadre d'un projet de fin de cycle pour obtenir un Master professionnel en Génie logiciel. L'objectif était de concevoir et de réaliser une application web nommée CevitalGPAO, destinée à optimiser le suivi et la gestion des productions au sein de la direction des systèmes d'information de l'entreprise Cevital agroalimentaire de Bejaïa, en particulier qualité et production.

Dans ce contexte, nous avons effectué un stage chez Cevital pour analyser de près son activité, identifier les anomalies et proposer des solutions correctives. L'application vise à simplifier les tâches des utilisateurs et à leur faire gagner du temps. Pour son développement, nous avons adopté la méthode SCRUM et utilisé le langage de modélisation UML. Le framework Laravel (PHP) a été choisi pour la programmation, et SQLite a été utilisé comme système de gestion de bases de données relationnelles (SGBDR).

Mots clés : Application web, gestion de production, SCRUM, UML.

Abstract

This work was carried out as part of a final project to obtain a professional Master's degree in Software Engineering. The objective was to design and develop a web application named CevitalGPAO, aimed at optimizing the tracking and management of production within the information systems department of the agro-food company Cevital in Bejaïa, particularly in terms of quality and production.

In this context, we completed an internship at Cevital to closely analyze its activities, identify anomalies, and propose corrective solutions. This application aims to simplify tasks for users and save them time. To implement our solution, we adopted the SCRUM development process and used the Unified Modeling Language (UML) for modeling. We chose to program the application using Laravel (PHP). Regarding data management, we used SQLite as relational database management system (RDBMS).

Keywords : Web application, production management, SCRUM, UML.