

Dédicaces

À la mémoire aimante de mon cher père,

Je dédie humblement ce rapport de mon projet de fin d'études. Chaque page de ce document porte l'empreinte de ton amour inconditionnel, de ta sagesse et de ta bienveillance qui ont façonné l'individu que je suis aujourd'hui. Ta présence, même en ton absence, a été une source inépuisable d'inspiration et de motivation tout au long de mon parcours académique.

Bien que tu ne sois plus physiquement parmi nous, ton héritage se perpétue dans mes actions, dans mes réussites et dans ma vision de l'avenir. Ce rapport est mon humble témoignage de gratitude envers toi, une façon d'immortaliser ton souvenir et de partager nos réalisations communes.

Je voudrais également exprimer ma reconnaissance envers ma mère qui m'a soutenu tout au long de ce parcours. Leur amour inébranlable et leur soutien indéfectible ont été des piliers solides qui m'ont permis d'atteindre mes objectifs.

Papa, ce rapport de projet de fin d'études est bien plus qu'un simple document. C'est un symbole de notre lien éternel, de notre amour inconditionnel et de l'influence profonde que tu as eue sur moi. Chaque victoire, chaque accomplissement, je les dédie à ta mémoire.

Avec tout mon amour et ma reconnaissance éternels, HASSENE.

Je tiens à dédier ce travail à ma mère pour son amour infini, son soutien incomparable, pour sa compréhension qui n'a pas d'équivalent, avec mes sentiments d'amour et de respect les plus chaleureux ; à mon cher père à qui je dois tant et tout, symbole de courage et de sacrifice, sa patience et son aide qui m'ont toujours encouragé et soutenu au cours de la période de mes études. Je souhaite que ce travail soit un témoignage de ma profonde affection et reconnaissance du sacrifice de mes parents.

SENA GUILFO STEEVY.

Remerciements

C'est avec une profonde reconnaissance et considération particulière que je remercie mon encadreur académique Mr Masmoudi Faouzi pour ses conseils, son soutien et ses suggestions judicieuses.

Un merci bien particulier est dédié à Mr Feki Bechir, mon encadreur industriel, pour son accueil, son aide appréciable, sa servabilité et sa disponibilité.

Je tiens à remercier aussi Monsieur le président du jury et tout membres du jury qui ont bien accepté juger ce modeste travail et tout le corps professionnel de

L'institut International de technologie de Sfax et plus particulièrement à celui du département de génie industriel auquel m'a accompagné lors de mon cursus.

MAHFOUDH HASSENE.

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage, qui m'ont accompagné et aidé lors de la rédaction de ce rapport.

Tout d'abord je remercie vivement mon maître de stage, M. Béchir Feki, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions et ainsi développé mes connaissances grâce à son aide précieuse.

Je remercie également mon encadrante académique, Mme Nouha Derbel, pour son accompagnement et sa grande disponibilité dans l'organisation de mon travail tout au long de ce stage et son aide précieuse dans la rédaction de ce rapport.

Je remercie enfin mon collègue de travail, Mahfoudh Hassene, pour son étroite collaboration et son aide indispensable dans les études et les réalisations menées tout au long de ce stage.

SENA GUIFFO STEEVY.

Table des matières

Introduction générale.....	15
Chapitre 1 : Présentation de l'entreprise	16
1. Introduction	16
2. Informations générales	17
3. Engagements et valeurs	17
4. Fiche technique	18
5. Structure hiérarchique et différents services	19
5.1 Structure hiérarchique	19
5.2 Différents services	19
6. Les produits d'Essilor SIVO.....	21
6.1 Les verres organiques	21
6.2 Les verres minéraux	21
7. Description des lignes de production et de traitement verres.....	22
7.1 Atelier de Surfaçage des verres.....	24
7.2 Atelier de coloration des verres	28
7.3 Atelier de traitement resys et anti-reflet.....	29
8. Conclusion.....	30
Chapitre 2 : Processus de Production et Gestion de Production	31
1. Introduction	31
2. La production	31
2.1 Définition	31
2.2 Objectifs associés	32
2.3 Décomposition du système de production	32
3. Gestion de la production	33
3.1 Définition	33
3.2 Rôle et importance	33
3.3 Les contraintes	34
3.4 Les problèmes des systèmes de production	34
4. Conclusion.....	36
Chapitre 3 : Les problèmes d'ordonnancement.....	37
1. Introduction	37
2. Méthodes de résolution des problèmes d'ordonnancement	37

2.1	Méthode de résolution exacte	37
2.2	Outils de suivi de Planification	39
2.3	Méthodes de résolution approchées	40
3.	Conclusion.....	43
Chapitre 4 : description du problème et résolution avec l'algorithme génétique :		44
1	Introduction	44
2	Ordonnancement en industrie sivo	44
3	Type de produit Essilor sivo.....	46
4	Problème survenant dans l'atelier sivo.....	47
4.1	Analyse de l'existant et chronométrage	47
4.2	Analyse du temps de traitement dans l'atelier : Identification du goulot d'étranglement	48
5	Formulation du Problème.....	50
5.1	Présentation de problème à étudier	50
6	Résolution du problème d'ordonnancement en industrie sivo par l'algorithme génétique	51
6.1	Introduction.....	51
6.2	Principe d'algorithme génétique	51
6.3	Phase de test	57
7	Conclusion.....	60
Chapitre 5 : La Simulation du nouveau système d'ordonnancement		61
1.	Contexte général de la simulation	61
1.1	Définition de la simulation.....	61
1.2	Objectifs de la simulation	61
1.3	Les différents logiciels de simulation	62
1.4	Le choix de logiciel de simulation	63
2.	Travail préalable à la simulation	66
2.1	Collecte des données et détermination des paramètres	66
3.	Modélisation sur Arena	69
3.1	Construction du modèle	69
3.2	Présentation du modèle	72
4.	Validation et vérification du modèle	76
4.1	Détermination du nombre des réplications	76
4.2	Détermination de la période de warm-up.....	78

4.3 Validation du modèle	79
5. Conclusion.....	82
Chapitre 6 : Spécification des besoins et analyse.....	83
1. Introduction	83
2. Identification des acteurs.....	83
3. Spécification des besoins	83
3.1. Identification des Besoins fonctionnels	84
3.2. Identification des besoins non fonctionnels	84
4. Choix technologique	85
4.1. Environnement logiciel	85
4.2. Environnement de développement	87
5. Architecture globale de la solution.....	92
5.1. Architecture physique.....	93
5.2. Architecture logique	94
6. Conclusion.....	95
Chapitre 7 : Méthodologie adoptée et planification du projet.....	96
1. Introduction	96
2. Méthodologie adoptée : Kanban	96
3. Backlog du produit	97
4. Planification	98
5. Outil de Planification	98
6. Conclusion.....	99
Chapitre 8 : Mise en place de l'authentification et de la gestion des rôles et utilisateurs.....	100
1. Introduction	100
2. Description des exigences fonctionnelles.....	100
3. Diagramme des cas d'utilisation	101
3.1. Présentation du diagramme de cas d'utilisation	101
3.2. Description textuelle des sous-cas d'utilisation	101
4. Diagrammes de séquence	104
5. Diagramme de classe.....	108
6. Réalisation.....	109
7. Conclusion.....	111
Chapitre 9 : Mise en place de la gestion de la chaîne de production, des traitements et des clients.	112

1. Introduction	112
2. Description des exigences fonctionnelles.....	112
3. Diagramme des cas d'utilisation	113
3.1. Présentation du diagramme des cas d'utilisation	113
3.2. Description textuelle des cas d'utilisation.....	113
4. Diagrammes de séquence	115
5. Diagramme de classe.....	121
6. Réalisation.....	122
7. Conclusion.....	123
Chapitre 10 : Mise en place de la gestion des ordres de fabrication et du stock.....	124
1. Introduction	124
2. Description des exigences fonctionnelles.....	124
3. Diagramme des cas d'utilisation	125
3.1. Présentation du diagramme des cas d'utilisation	125
3.2. Description textuelle des cas d'utilisation	125
4. Diagrammes de séquence	127
5. Diagramme de classe.....	131
6. Réalisation.....	132
7. Conclusion.....	135
Chapitre 11 : Implémentation de l'algorithme de planification optimisée et gestion des plannings	136
1. Introduction	136
2. Description des exigences fonctionnelles.....	136
3. Diagramme des cas d'utilisation	137
3.1. Présentation du diagramme de cas d'utilisation.....	137
3.2. Description textuelle des sous-cas d'utilisation	137
4. Diagrammes de séquences	140
5. Diagramme de classe.....	142
6. Réalisation.....	143
7. Conclusion.....	145
Conclusion générale	146

Liste des figures

Figure 1: logo société Essilor Sivo.....	16
Figure 2: Plan de situation d'Essilor sivo.....	17
Figure 3:Mr Sellami Khaled.....	17
Figure 4: opérateur de contrôle	18
Figure 5: Opérateur machine.....	18
Figure 6: organigramme société	19
Figure 7: flux de production Essilor sivo	23
Figure 8: verre avec film	24
Figure 9: machine préparation film	24
Figure 10: machine glantage	24
Figure 11: machine orbite	25
Figure 12: machine polissage avec 1 seule broche.....	25
Figure 13: machine polissage	25
Figure 14: machine gravure.....	26
Figure 15: machine deglantage	26
Figure 16: bureau de contrôle.....	27
Figure 17: atelier coloration	28
Figure 18: baignoire de coloration	28
Figure 19: machine fast cure	29
Figure 20: machine resys.....	29
Figure 21:Rack des verre.....	29
Figure 22: machine anti reflet	29
Figure 23: plateau pour verre anti reflet	29
Figure 24: système de production	32
Figure 25: la fonction ordonnancement	35
Figure 26: Diagramme Gant.....	39
d'autres. Figure 27: réseau pert.....	39
Figure 28: Fonctionnement d'algorithme génétique.....	41
Figure 29: recherche taboue	42
Figure 30: flux de production atelier sivo	45
Figure 31: cadence des produits sivo	47

Figure 32: cadence des OF et OT	47
Figure 33: temps moy de coloration.....	48
Figure 34: temps moy de resys.....	48
Figure 35: temps moy d'anti reflet.....	49
Figure 36: terminologie d'algorithme génétique	52
Figure 37: exemple de population	52
Figure 38: structure de notre algorithme	53
Figure 39: exemple de sélection.....	55
Figure 40: croisement avec un point	56
Figure 41: mutation uni point.....	56
Figure 42:pourcentage RX 24 avec loterie de biaisée	57
Figure 43:roue du forain.....	57
Figure 44:pourcentage RX 24 avec méthode élitiste.....	58
Figure 45:pourcentage RX 24 avec méthode stochastique	58
Figure 46:pourcentage RX 24 avec méthode de sélection par tournoi.....	59
Figure 47:pourcentage RX 24 avec méthode croisement uniforme	59
Figure 48:pourcentage RX 24 avec méthode de mutation par valeur	60
Figure 49: logiciel anylogic.....	62
Figure 50: logiciel flexim	62
Figure 51: logiciel arena.....	62
Figure 52: environnement arena.....	64
Figure 53: Exemple de modélisation d'un système de production avec Arena	66
Figure 54: temps inter arrivé magasin semi fini	66
Figure 55: temps inter arrivé magasin fi ni	66
Figure 56: détermination du loi par input Analyzer	68
Figure 57: modèle de production société sivo.....	69
Figure 58: liste des ressources.....	70
Figure 59: liste des processus	71
Figure 60: modèle d'atelier de surfaçage	72
Figure 61: modèle des taux de retouche et rebut.....	73
Figure 62: modèle d'atelier de coloration	74
Figure 63: modèle d'atelier de resys	74
Figure 64: modèle e bureau de contrôle qualité	75

Figure 65: modèle d'atelier d'anti reflet	75
Figure 66: détermination de période de warm up.....	78
Figure 67:temps d'attente des process	79
Figure 68:nombres des verres en attente	80
Figure 69:nombres des of et ot pour l'ancien système d'ordonnancement	81
Figure 70:nombres des of et ot pour nouveau système	81
Figure 71:Logo Visual Studio Code.....	85
Figure 72:Logo Spring Tools Suite	85
Figure 73:Logo Xampp	86
Figure 74:Logo MySQL.....	86
Figure 75:Logo Postman	86
Figure 76:Logo Draw.io	87
Figure 77:Logo TypeScript	87
Figure 78:Logo Html 5.....	88
Figure 79:Logo Css 3	88
Figure 80:Logo Spring Security	89
Figure 81:Logo Spring Boot Admin	90
Figure 82:Logo OptaPlanner	90
Figure 83:Logo Angular.....	91
Figure 84:Logo Angular.....	91
Figure 85:Architecture globale.....	93
Figure 86:Architecture physique	94
Figure 87:architecture logistique.....	94
Figure 88:Tableau Kanban	96
Figure 89:Logo Trello	98
Figure 90:Planification réalisée sur Trello	99
Figure 91:Diagramme de cas d'utilisation de la phase 1	101
Figure 92: Diagramme de séquence d'ajout d'un utilisateur.....	104
Figure 93: Diagramme de séquence de modification d'un utilisateur	105
Figure 94: Diagramme de séquence de gestion de profil	106
Figure 95: Diagramme de séquence de suppression d'utilisateur.....	107
Figure 92:Diagramme de classe de la phase 01.	Erreur ! Signet non défini.
Figure 97 : Diagramme de classe de la phase 01	108

Figure 94:Interface de login	109
Figure 95:Echantillon de code du composant LoginPage	109
Figure 96:Interface de la page d'accueil	110
Figure 97:Echantillon du code de configuration Spring Security	110
Figure 98:Interface de gestion d'utilisateurs	111
Figure 99:Diagramme des cas d'utilisation de la phase 02	113
Figure 100:Diagramme de séquence de modification de la chaîne de production.....	115
Figure 101:Diagramme de séquence d'ajout et de suppression de phase	116
Figure 102:Diagramme de séquence de modification de traitement.....	117
Figure 103:Diagramme de séquence d'ajout et de suppression de traitement	118
Figure 104:Diagramme de séquence de modification de client	119
Figure 105:Diagramme de séquence d'ajout et de suppression de client	120
Figure 106:Diagramme de classe de la phase 03.	121
Figure 107:Interface de la gestion de la production	122
Figure 108:Interface du formulaire d'ajout et de modification de phase.....	122
Figure 109:Interface de la gestion des traitements	123
Figure 110:Interface de la gestion des clients	123
Figure 111:Diagramme de cas d'utilisation de la phase 03.	125
Figure 112:Diagramme de séquence d'ajout et de suppression d'un ordre de fabrication. ...	127
Figure 113:Diagramme de séquence de la modification d'un ordre de fabrication.....	128
Figure 114:Diagramme de séquence d'ajout et de suppression d'une ressource.....	129
Figure 115:Diagramme de séquence de modification d'une ressource.....	130
Figure 116;Diagramme de classe de la phase 03.	131
Figure 117:Interface de la page de gestion des ordres de fabrication	132
Figure 118:Interface des détails d'un ordre de fabrication.....	133
Figure 119:Interface de modification.....	133
Figure 120:Interface de gestion des ressources.....	134
Figure 121:Interface d'ajout de ressources en stocks.....	134
Figure 122:Diagramme de cas d'utilisation de la phase 04.	137
Figure 123:Diagramme de séquence du sous-cas « générer un planning »	140
Figure 124:Diagramme de séquence du sous-cas « générer automatiquement des plannings »	141
Figure 125:Diagramme de classe de la phase 04	142
Figure 126:Interface de la page de planification	143

Figure 127:Interface des détails d'une planification	144
Figure 128:Interface du suivi du planning	144
Figure 129:implémentation des contraintes d'Optaplanner	145

Liste des tableaux

Tableau 1:Fiche technique de la société.....	18
Tableau 2: les types d'anti reflet.....	46
Tableau 3 :Les modules fondamentaux du simulateur Arena	65
Tableau 4:Détermination des lois qui décrivent les processus.....	67
Tableau 5:Les taux des déchets et des rebuts	68
Tableau 6:Backlog produit	97
Tableau 7:Planifications des tâches.....	98
Tableau 8:user stories phase 1	100
Tableau 9:Description textuelle « Authentification »	102
Tableau 10:description textuelle « Gérer les utilisateurs et affecter des rôles »	102
Tableau 11:Description textuelle « envoyer une demande utilisateur »	103
Tableau 12:description textuelle « gestion des demandes utilisateur »	103
Tableau 13:User stories de la phase 02	112
Tableau 14:Description textuelle du sous-cas « gérer les phases de production ».....	114
Tableau 15:Description textuelle du sous-cas « gérer les traitements ».....	114
Tableau 16:Description textuelle du sous-cas « gérer les clients ».....	114
Tableau 17:Tableau des user stories de la phase 03.....	124
Tableau 18:Description textuelle du sous-cas « gérer les ordres de fabrication/traitement »	126
Tableau 19:Description textuelle du sous-cas « gérer les ressources en stocks ».....	126
Tableau 20:user stories phase 04.....	136
Tableau 21:Description textuelle « Générer un planning ».....	138
Tableau 22:description textuelle « Génération automatique de plannings ».....	138
Tableau 23:Description textuelle « suivre le planning »	139

Introduction générale

Notre époque est caractérisée par un environnement très concurrentiel qui oblige l'entreprise à tous optimiser pour pouvoir survivre et combattre face à une concurrence agressive qui ne cesse pas de s'accentuer. De ce fait, les entreprises cherchent toujours à bien organiser ses moyens de production pour mieux satisfaire les clients qui sont de plus en plus exigeants. Les clients cherchent la valeur en termes de coût, flexibilité, délai, qualité et innovation.

La production présente la fonction centrale pour toute entreprise industrielle. La compétitivité passe principalement par une gestion de production plus fiable. En effet, l'amélioration de la gestion de production vise à organiser le fonctionnement du système de production, et à bien gérer ses diverses opérations. Ainsi le perfectionnement de processus production contribue forcément à l'amélioration des rendements des postes de travail. Aujourd'hui la prise de décision, en particulier dans la gestion de production, est devenue numérique et informatisée.

Dans cet objectif, ce projet traite le problème d'ordonnancement de production et cela dans le cadre du projet "RX24" qui s'intéresse à l'amélioration de l'ordonnancement des envois de commandes dont les délais ne dépassent pas les 24 heures. Cette étape du processus de production joue un rôle déterminant dans la capacité de l'entreprise à respecter les délais de livraison en passant par une analyse approfondie les flux de production, nous allons mettre en place des stratégies et des algorithmes d'ordonnancement plus efficaces. L'objectif est d'accroître le taux de satisfaction client et à atteindre l'objectif de 100% de commandes livrées dans un délai de 24 heures.

Ce rapport comporte 11 chapitres, le premier chapitre présente le contexte du projet en décrivant le système production. Le deuxième chapitre comportera une description du processus de production et gestion de production. Dans le troisième chapitre, nous allons essayer de citer les problèmes d'ordonnancement. Ensuite on va passer à la description du problème et résolution avec l'algorithme génétique. Et afin de vérifier la fiabilité de notre solution on va passer à la Simulation du nouveau système d'ordonnancement dans le cinquième chapitre. Le sixième chapitre marque le début de l'implémentation de la solution informatique qui met en application les études menées dans les chapitres qui le précédent. Il sera consacré à la spécification des besoins de la solution. Ensuite nous présenterons la méthodologie de travail utilisée pour réaliser cette solution ainsi que la planification qui en découle. Pour terminer, nous allons présenter l'ensemble des travaux réalisés au cours de chaque phase de cette planification.

Chapitre 1 : Présentation de l'entreprise

1. Introduction

Avant de plonger dans notre projet, il est essentiel de bien appréhender la chaîne de production de l'entreprise. Dans ce premier chapitre, nous vous offrons un aperçu global d'Essilor Sivo, mettant en lumière des informations générales, son engagement et ses valeurs, ainsi que des détails techniques. Nous examinerons également la structure hiérarchique et les différents services de l'entreprise, ainsi que les produits qu'elle propose. Enfin, nous décrirons en détail la ligne de production et de traitement, offrant ainsi une compréhension approfondie de son fonctionnement.



Figure 1: logo société Essilor Sivo

2. Informations générales

Essilor SIVO, Société Franco Tunisienne créée en 2012, une joint-venture entre SIVO et Essilor international.

SIVO, connue aussi sous la marque Laboratoires SIVO fut créée en 1969 par son fondateur Mr Abdelaziz SELLAMI, Opticien diplômé de l'école d'optique Bures sur Yvette et installé premier opticien tunisien diplômé, à Sfax en 1964.

Laboratoires SIVO a évolué durant les 40 dernières années de manière fiable et constante en se basant sur un savoir-faire acquis au fil des années, ses ressources humaines qualifiées, et en s'appuyant sur plusieurs partenaires économiques et technologique, dont Essilor International qui fut son partenaire historique depuis sa création en 1969.

Aujourd'hui, grâce à un savoir-faire, accumulé depuis la fondation de l'entreprise, et une technologie de pointe, Laboratoires SIVO est devenu capable de répondre à tous les besoins spécifiques émanant de la part de nos clients partout là où nous sommes installés. Fort de son expérience, et de sa notoriété mondiale reconnue, laboratoires SIVO a su s'imposer sur le devant de la scène en devenant un partenaire incontournable des professionnels de l'optique dans le continent Africain.



Figure 3:Mr Sellami Khaled



Figure 2: Plan de situation d'Essilor sivo

3. Engagements et valeurs

Chez ESSILOR SIVO, l'humanisme, la compétence, la qualité et l'engagement. Favorisent depuis toujours une organisation collective d'entreprise, et ce, à travers notre réseau de filiale et d'agence partout là où nous sommes présents. Ces valeurs trouvaient leur fondement depuis la création de la firme à travers une double mission économique et sociale.

Les valeurs d'Essilor SIVO s'enrichissent au quotidien dans notre approche économique et sociale, que ce soit dans notre management ou à travers notre attitude vis-à-vis de nos clients, partenaires, parties prenantes et institutions financières et publiques.

La consécration de cette approche était par la remise du « Prix Présidentiel Pour le Progrès Social Pour Les Entreprises » le 01 mai 2007 à Tunis. Une récompense qui couronne l'effort social et la contribution au développement durable de la Tunisie fait par ESSILOR SIVO depuis 1970



Figure 5: Opérateur machine



Figure 4: opérateur de contrôle

4. Fiche technique

Dans ce chapitre, nous explorerons en détail la fiche technique d'Essilor Sivo, en mettant l'accent sur la richesse des informations qu'elle contient et sur son importance pour l'entreprise et ses clients

. Tableau 1:Fiche technique de la société

Nom :	Essilor SIVO
Organes de direction :	Mr Alexandre Montagne (président du conseil) Mr Khaled Sallemi (Directeur Général)
Siège :	Route industriel poudrière 1 Sfax 3002
Date de création :	01/11/2012
Forme juridique :	Société anonyme de droit Tunisien
Capital social :	24.000MDT
Nature de la société :	Résidente /partiellement exportatrice
Secteur d'activité :	Industriel
Spécialité :	Surfaçage des verres optiques.
Nombres d'employés :	Plus de 200 employés
Téléphone :	+216 74 151 400
Email :	Http://www.sivo.com.tn

5. Structure hiérarchique et différents services

5.1 Structure hiérarchique

Suite à sa collaboration avec le groupe Essilor la structure hiérarchique de l'entreprise a été un peu modifiée et devenue comme suit :

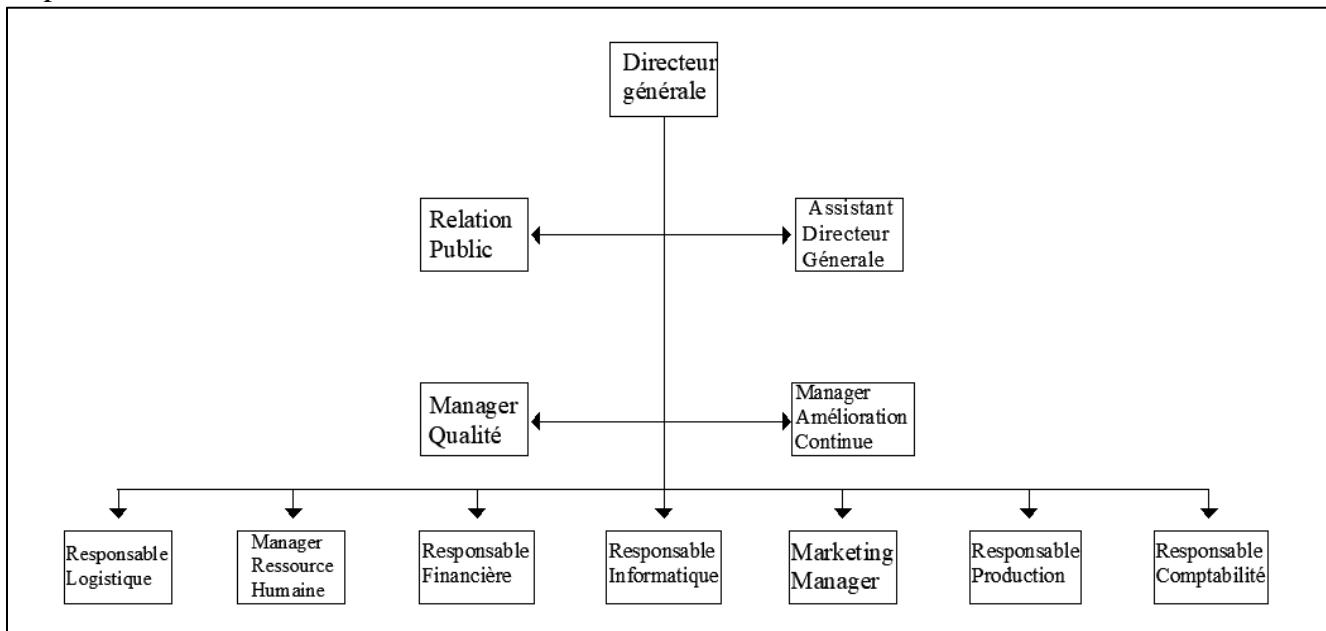


Figure 6: organigramme société

La direction générale a pour rôle de :

- Définir les objectifs et tracer la politique de l'entreprise
- Prévoir les ressources nécessaires pour atteindre les objectifs de la société
- Bien gérer les ressources disponibles

5.2 Différents services

Essilor-SIVO comporte six services :

- *Service assurance qualité :*

Le rôle de ce service est de :

- Assurer le contrôle à la réception de la matière première
- Préparer les audits et les revues de direction
- Fixer les normes de qualité

- *Service production :*

Ce service a pour rôle de :

- Respecter les délais de livraisons préalablement établis
- Respecter les normes établies par le service assurance qualité
- Assurer la production en minimisant les rebuts et les défauts en portant des actions correctives et préventives

- *Service approvisionnement et achat :*

Ce service a pour rôle de :

- Sélectionner les fournisseurs selon leurs aptitudes à satisfaire les besoins et les exigences de la société
- Veiller à ce que les commandes soient claires et contiennent les spécifications exigées : le type, la catégorie, la classe, la norme se rapportant au produit
- S'assurer de la conformité de la marchandise acheté par rapport à la commande demandée

- *Service commercial :*

Ce service doit :

- S'assurer que la commande du client est claire et contient toutes les spécifications nécessaires pour l'exécution de cette commande
- S'assurer que la production est capable d'exécuter cette commande avec les délais et les exigences souhaités par les clients.

A sa création le service commercial de SIVO faisait une part entière de la société. Mais pour assurer une forte pénétration dans le marché tunisien et étranger, le service s'est séparé de l'entreprise pour avoir un nouveau nom : SICOM.

- *Service ressource humaine*

Son rôle est de :

- contrôler les absences et les retards et tout ce qui se rapporte au social des employés
- Etablir les fiches de paye
- Faire les recrutements et les licenciements
- Organiser des programmes de formation professionnelle

- *Service marketing :*

C'est le service le plus récent d'Essilor-SIVO.

Son rôle est de faire connaître les produits de la société auprès des clients. Il représente l'entreprise dans les expositions et il conçoit tout ce qui est prospectus ou brochure.

6. Les produits d'Essilor SIVO

Le verre peut être classé en deux grandes familles : les verres organiques ou plastiques et les verres minéraux

6.1 *Les verres organiques*

La matière qui constitue ces verres est à base de polymère. Cette matière présente l'avantage de grande résistance à la casse et procure plus de sécurité notamment pour les enfants. Cependant, elle est tendre, ce qui la rend vulnérable aux rayures

- *Les verres à simple foyer ou uni-focaux :*

Ils peuvent être :

- Des verres sphériques, c'est-à-dire chaque surface présente les mêmes rayons sur tous les axes.
 - Des verres toriques
- Présentant une sphère et un cylindre destinés aux astigmates

- *Les verres à double foyer ou bifocaux :*

Ils permettent la correction à la fois des. Visions de près et de loin. Pour ce type de verres, il y a présence d'un segment qui permet la vision de près mais qui gêne l'œil lors du passage de la vision de loin à la vision de près.

- *Les verres progressifs :*

Ils ont le même rôle que les doubles foyers mais le segment est invisible, ces verres sont alors plus confortables et plus esthétiques que les doubles foyers.

6.2 *Les verres minéraux*

Les verres minéraux sont réputés pour leur résistance aux rayures et leur durabilité. Ils offrent une excellente clarté optique et sont particulièrement adaptés aux corrections optiques plus élevées. Leur composition minérale leur confère une grande résistance aux produits chimiques, ce qui en fait un choix populaire pour les environnements où une résistance accrue est requise, tels que les laboratoires ou les zones de travail industrielles.

Cependant, en raison de leur composition minérale, les verres minéraux sont généralement plus lourds que les verres organiques, ce qui peut les rendre moins confortables à porter, surtout pour de longues périodes. Ils sont également plus sujets à la casse en cas de choc violent, ce qui peut nécessiter une manipulation plus prudente.

Malgré ces limitations, les verres minéraux restent une option privilégiée pour de nombreux porteurs de lunettes en raison de leur résistance, de leur clarté optique et de leur capacité à corriger des problèmes de vision plus complexes.

7. Description des lignes de production et de traitement verres

L'entreprise est constituée 3 ateliers : Un atelier de surfaçage, un atelier de coloration et un atelier d'anti reflet.

Puisque le type le plus demandé est le Free Forme, on va se focaliser dans notre projet seulement sur ce type de verre.

Plus précisément on va s'intégrer au sein de la société afin de mieux comprendre ses différents départements. Le système de production possède 3 types d'ordre :

- OF : ordre de fabrication
- OT : ordre de traitement
- OS : ordre de stock

Les OF ont la possibilité de passer par les 3 ateliers mais les OT ne peuvent passer que par l'atelier de coloration et l'atelier d'anti reflet.

La quantité des OS est négligeable par rapport à celle des OT et des OF. Pour cela, nous n'allons traiter que les **OF** et les **OT** dans notre projet.

La figure suivante illustre d'une façon globale l'acheminement des verres dans l'**Essilor SIVO**.

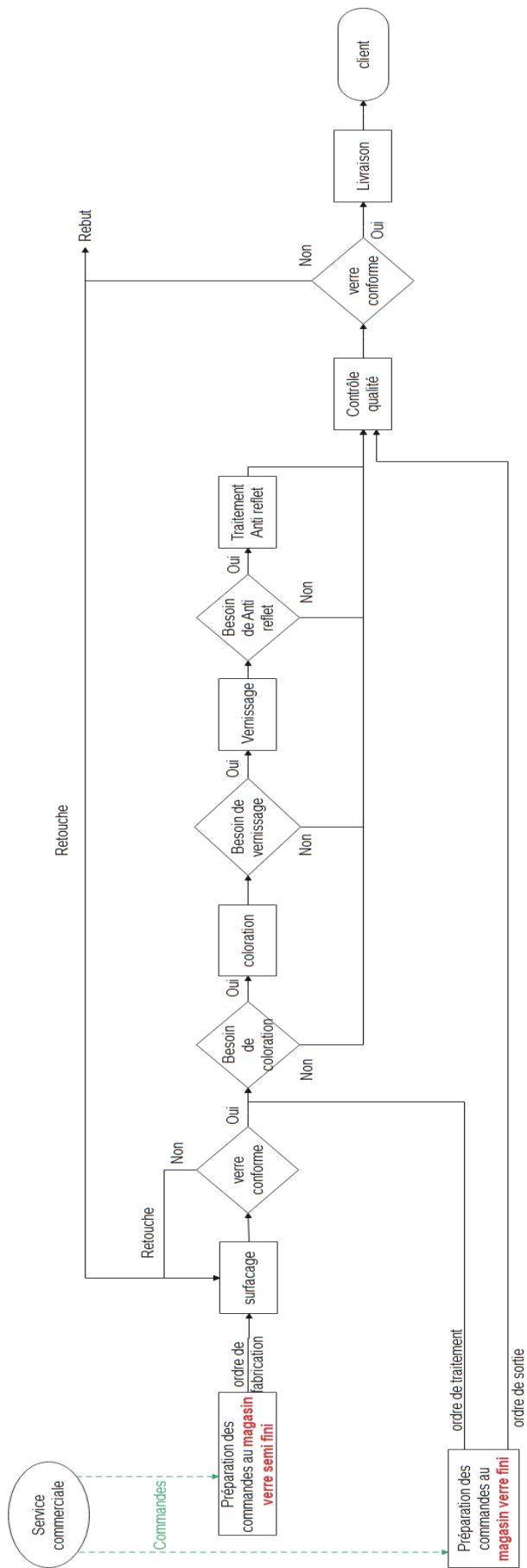


Figure 7: flux de production Essilor sivo

Dans les parties suivantes, on va décrire la chaîne en détails.

7.1 Atelier de Surfaçage des verres

- *Préparation film des verres*

Cette étape consiste à protéger les verres au cours du surfaçage. Les ouvriers couvrent la bosse du verre à l'aide d'un film manuellement ou automatiquement. Ils utilisent la machine manuelle si les verres ont des épaisseurs très petites sinon ils utilisent la machine automatique.



Figure 9: machine préparation film



Figure 8: verre avec film

- *Glantage*

Dans cette étape, un support métallique est ajouté aux verres. Ce support est constitué d'un bouton et d'un alliage de plomb. L'alliage doit être à la température 50 °C et les boutons doivent être appropriés à l'utilisation.



Figure 10: machine glantage

- *Ebauchage*

L'entreprise dispose d'une machine pour faire l'ensemble des opérations de doucissage, d'ébauchage et de détourage (cette étape consiste à diminuer le diamètre du verre). Toutes les 3 informations nécessaires pour la réalisation de ces opérations se trouvent dans le code à barre de la fiche du verre.



Figure 11: machine orbite

- *Polissage*

Cette opération commence par sélectionner l'outil nécessaire à chaque verre. Ensuite, les opérateurs fixent l'outil et le verre dans la machine lance le cycle de polissage. Enfin, ils nettoient le verre. Si le verre est progressif (vision de pré et de loin en même temps) il passe par le poste de gravure sinon on le verre passe directement par le poste de déglantage.



Figure 13: machine polissage



Figure 12: machine polissage avec 1 seule broche

- *Gravure*

Cette étape consiste à tracer l'addition et les cercles (qui sont nécessaires pour le montage) sur le creux du verre. Ensuite, les verres sont envoyés vers le poste de déglantage.



Figure 14: machine gravure

- *Déglantage*

Dans cette étape, le support métallique est enlevé par choc à l'aide de bagues creuses de différents diamètres puis le film est aussi enlevé.



Figure 15: machine deglantage

- *Contrôle de surface*

Le poste du contrôle est le plus intéressant dans la chaîne. Il respecte les exigences du client. Ce poste peut rencontrer plusieurs défauts de fabrication par exemple :

- Puissance très forte : Dans ce cas, on remet le verre dans la chaîne ou on le jette.
- Puissance très faible : On couvre le verre puis on fixe le support métallique. Ensuite, on fait une autre fois le polissage pour diminuer la puissance.
- Fausse gravure : On couvre le verre puis on fixe le support métallique. Ensuite, on fait trois fois le polissage tout en choisissant l'outil afin de ne pas changer la puissance pour effacer la gravure et on répète la gravure.
- Verre inversé : On remet le verre dans la chaîne.

Il y a aussi des défauts qui ne peuvent pas être corrigés. Dans ce cas le verre est écarté comme rebuts.



Figure 16: bureau de contrôle

7.2 Atelier de coloration des verres

La coloration est une étape de traitement du verre au sein de la matière. Elle utilise des produits liquides et des produits poudres.

Les verres qui entrent dans ce poste proviennent du surfaçage à travers le contrôle ou bien à partir du magasin. Les étapes de la coloration sont les suivants :

- Trier les verres dans un raque pour les rendre dans le même ordre que leurs fiches puis les mettre dans une machine afin d'enlever leur vernis ensuite les nettoyer avec l'alcool (si les verres proviennent du magasin).
- Tracer l'axe du verre optique puis marquer avec une lame cet axe (si on a dégradation dans la fiche du verre).
- Mettre les verres dans la machine de coloration selon le traitement demandé.
- Nettoyer après la coloration par l'hexol et l'alcool 90.
- Vérifier la couleur à travers la lumière blanche.

Après la coloration, le verre continue son circuit. Si le verre nécessite de l'anti reflet ou de resys, il passe dans le poste de préparation resys sinon il passe au poste de contrôle.

On peut rencontrer beaucoup de défauts dans le poste de coloration qui sont refusés par le contrôle. Ces défauts peuvent être corrigés. Pour cela on met le verre dans le décolorant afin de relancer les étapes de coloration dès le début ou bien on ajoute d'autre couleur pour le corriger.



Figure 17: atelier coloration



Figure 18: baignoire de coloration

7.3 Atelier de traitement resys et anti-reflet

L'anti-reflet des verres optiques nécessite une couche de vernis. Cette couche peut être faite dans le poste de resys ou bien elle existe déjà dans les verres qui arrivent directement à partir du magasin. Si la couche existe déjà on doit mettre les verres provenant du magasin dans le poste de lavage pour les nettoyer à l'aide des produits différents.

L'opérateur doit mettre les verres de même type de traitement dans le même plateau et dans le même ordre que les fiches puis il le met dans le four.

Avant d'arriver à la machine de resys on doit passer par le poste de préparation de resys. Ce poste prend les verres du bureau d'ordre ou bien du poste de coloration. Au cours de la préparation on met chaque type d'anti reflet dans une péniche et les verres de test dans une péniche puis on nettoie les verres avec l'alcool. Ensuite, on les trie dans des raques. Enfin on les met en position pour lancer la machine de resys.

Après le resys, l'opérateur met les verres de chaque rack dans un plateau suivant le même ordre puis il met ce plateau dans le four pour passer au poste d'anti reflet.

Le poste de l'anti reflet se compose de trois machines automatiques. Une fois le traitement d'anti reflet est terminé, les verres sont transportés au poste de contrôle fini qui est la dernière étape avant le pré livraison aux clients.



Figure 21: Rack des verre



Figure 20: machine resys



Figure 19: machine fast cure



Figure 23: plateau pour verre anti reflet



Figure 22: machine anti reflet

8. Conclusion

Une fois que le processus de fabrication a été pleinement maîtrisé, nous entamerons le chapitre suivant, qui traite du "Processus de Production et Gestion de Production". Ce chapitre se penche sur la manière dont les entreprises gèrent la production de ses produits et les différentes étapes impliquées dans ce processus. Nous commencerons par définir clairement le concept de production et ses objectifs associés. Ensuite, nous examinerons en détail la gestion de la production, en mettant l'accent sur son rôle crucial et son importance pour assurer une production efficace et de qualité. Nous aborderons également les contraintes auxquelles l'entreprise peut être confrontée lors de la gestion de sa production, ainsi que les problèmes potentiels pouvant survenir dans les systèmes de production complexes. Ce chapitre vise à fournir une compréhension approfondie du fonctionnement global de la production dans l'entreprise, en préparant le terrain pour l'analyse ultérieure des problèmes d'ordonnancement et l'application d'algorithmes génétiques pour optimiser ce processus crucial.

Chapitre 2 : Processus de Production et Gestion de Production

1. Introduction

Dans nos jours de globalisation économique et de révolutions technologiques, la fonction de production devient de plus en plus stratégique, dans la mesure, où elle oblige à la maîtrise d'un environnement instable comme elle doit atteindre un objectif mixte de coût-qualité-délais, flexibilité, qui caractérise la meilleure réponse de l'entreprise au consommateur.

Dans ce chapitre introductif on présente les principaux traits de la production, on utilisera trois sections : la première vise à proposer une définition aussi complète que possible de ce qui est la production, la seconde s'intéressera à la gestion de production. Enfin dans la dernière section on va clôturer ce chapitre en démentant les problèmes des systèmes de production qui inclut le problème d'ordonnancement et le problème d'optimisation.

2. La production

2.1 Définition

La production est le processus conduisant à la création de produits par l'utilisation et la transformation de ressources.

Le système de production (SP) représente l'ensemble du processus et des moyens de production que l'entreprise met en place afin de transformer des matières premières ou des produits semi finis, en produits finis prêts à être commercialisés.

Les systèmes de production industrielle se sont considérablement diversifiés et compliqués. En effet, ils peuvent se décomposer en plusieurs sous-systèmes, qui s'intègrent en vue d'assurer la pérennité et la compétitivité de l'entreprise.

Alors la production est la création des produits, le système de production est la séquence des opérations nécessaires pour réaliser des produits finis.

2.2 Objectifs associés

L'objectif principal des systèmes de production est de produire un bien économique. Cependant, la fonction de production se doit de satisfaire d'autres objectifs intermédiaires que sont :

- En termes de quantités produites : Il faut que la fonction de production permette de satisfaire la demande des clients. Pour pouvoir la réaliser, l'entreprise doit mener des actions pour maintenir les capacités productives ou bien mettre au point des plans d'investissement en capacité.
- En termes de qualité : Afin de satisfaire les besoins de la clientèle et s'assurer un niveau de compétitivité, il faudrait que les biens économiques produits soient de bonne qualité
- En termes de coût : Il faut que l'entreprise garantisse sa compétitivité par des coûts de production les plus faibles possibles.
- En termes de délais : Cela consiste à respecter des délais raisonnables, conformes avec le niveau de demande à laquelle doit faire face l'entreprise. Cela permettra d'éviter le stockage des biens finaux et le stockage des produits finis.
- En termes de flexibilité : C'est-à-dire la capacité du Système de Production à pouvoir s'adapter aux variations de la demande.

2.3 Décomposition du système de production

Classiquement, un système de production peut se décomposer en trois sous-systèmes : le système physique de production, le système de décision et le système d'information.

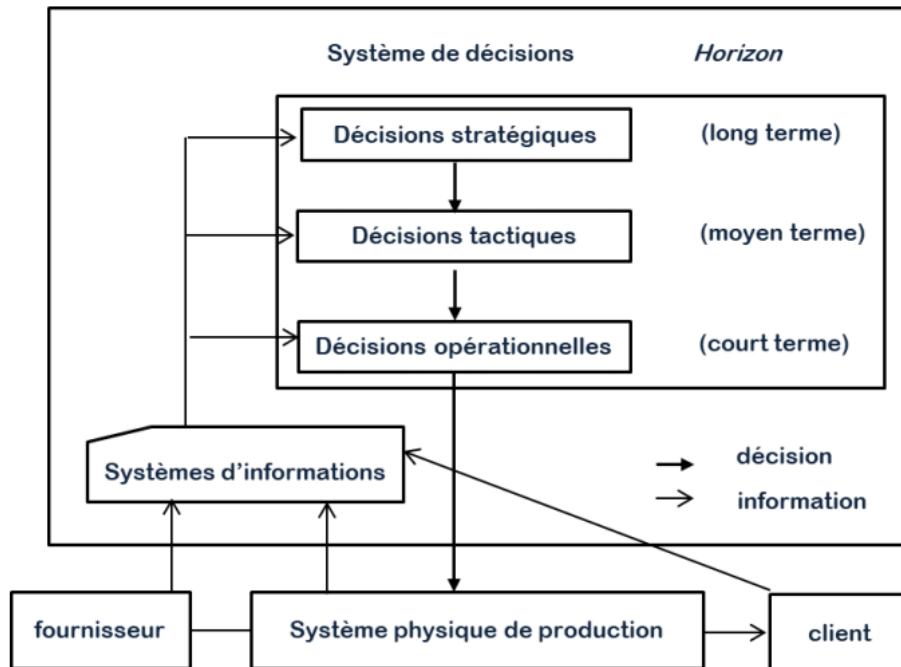


Figure 24: système de production

- *Le système physique de production (le système de fabrication) :*
Transforme les matières premières ou composantes en produits finis. Ce système est composé d'un ensemble des machines à commande numérique capables de faire des changements autonomes d'outils, un système automatisé de transport/manutention/stockage (chariots filoguidés, tapis roulants, robots dédiés au déplacement des pièces...).
- *Le système de décision (le système de contrôle) :*
Contrôle le système physique de production. Il en coordonne et organise les activités en prenant des décisions basées sur les données transmises par le système d'information.
- *Le système d'information :*
Intervient à plusieurs niveaux : à l'interface entre les systèmes de décision et de production ; à l'intérieur du système de décision, pour la gestion des informations utilisées lors de prises de décisions ; et à l'intérieur du système physique de production. Son rôle est de collecter, stocker et transmettre des informations de différents types.

3. Gestion de la production

3.1 Définition

La gestion de la production est « la fonction qui permet de réaliser les opérations de production en respectant les conditions de qualité, délai, cout qui résultent des objectifs de l'entreprise ».

La gestion de la production est la mise en application des méthodes et techniques dans le but d'accomplir la transformation des matières en produits fini. Elle se résume en la combinaison de ressources, parmi lesquelles les moyens matériels (les machines), les moyens humains (le personnel par qualification) et les matières (matières premières, matières consommables) dans un planning pour but d'assurer la fabrication du produit en qualité et en quantité définie.

3.2 Rôle et importance

Le rôle de la gestion de production est, d'organiser et de piloter le fonctionnement des processus physiques mis en œuvre dans l'entreprise, afin d'assurer une meilleure utilisation des moyens humains, physiques et technologiques disponibles, et de satisfaire au mieux l'objectif global de production défini en termes de quantités à fabriquer avec une qualité demandée, et des délais à respecter.

3.3 Les contraintes

Les contraintes rencontrées sont de divers ordres :

- *Financières* (produire à un cout optimal), cout des matières et consommables, cout de stockage des encours et des produits semi ouvres, cout de gestion des magasins, cout des heures de travail supplémentaires, cout des arrêts faisant partie intégrante du cout de revient, maitriser ces derniers est aussi une garantie pour la commercialisation des produits finis.
- *Temporelles* (produire dans les délais, assurer une livraison juste à temps) : éviter les ruptures de stocks, éviter le gonflage des stocks de produits finis. Car cela a une incidence directe sur la satisfaction de la clientèle (pertes de commandes) ou sur le cout de revient du produit finis dû aux couts supplémentaires du stockage.
- *Mécaniques* (maintenance préventives et gestion des temps d'arrêt), anticiper sur les pannes et prévoir des solutions alternatives en cas d'arrêt d'une machine.
- *Qualité* (produire avec le moins de défaut possible, le moins de déchets), un produit de bonne qualité participe à la fidélisation de la clientèle, véhicule l'image de marque de l'entreprise.
- *Planification* : définir un plan de production, définir les gammes opératoires et d'ordonnancer les opérations, et enfin de gérer la répartition des tâches durant tout le processus de fabrication.

3.4 Les problèmes des systèmes de production

L'objectif des systèmes de production (SP) est de satisfaire le client et pour y arriver il est impératif de mieux gérer la production, ce qui implique un ordonnancement pouvant aboutir à des échéances acceptables. De ce fait la complexité des SP peut se résumer en deux principaux problèmes : problème d'**ordonnancement** et problème d'**optimisation**.

- *Problème d'ordonnancement*

La fonction ordonnancement se décompose en trois sous fonctions :

- L'élaboration des ordres de fabrication (OF) : c'est-à-dire transformation des informations du programme directeur de production (suggestion de fabrication) en ordres de fabrication.
- L'élaboration du planning d'atelier : ceci dit, la détermination en fonction des ordres de fabrication et de la disponibilité des ressources le calendrier prévisionnel de fabrication.
- Le lancement et le suivi des opérations de fabrication : les 2 dernières étapes.

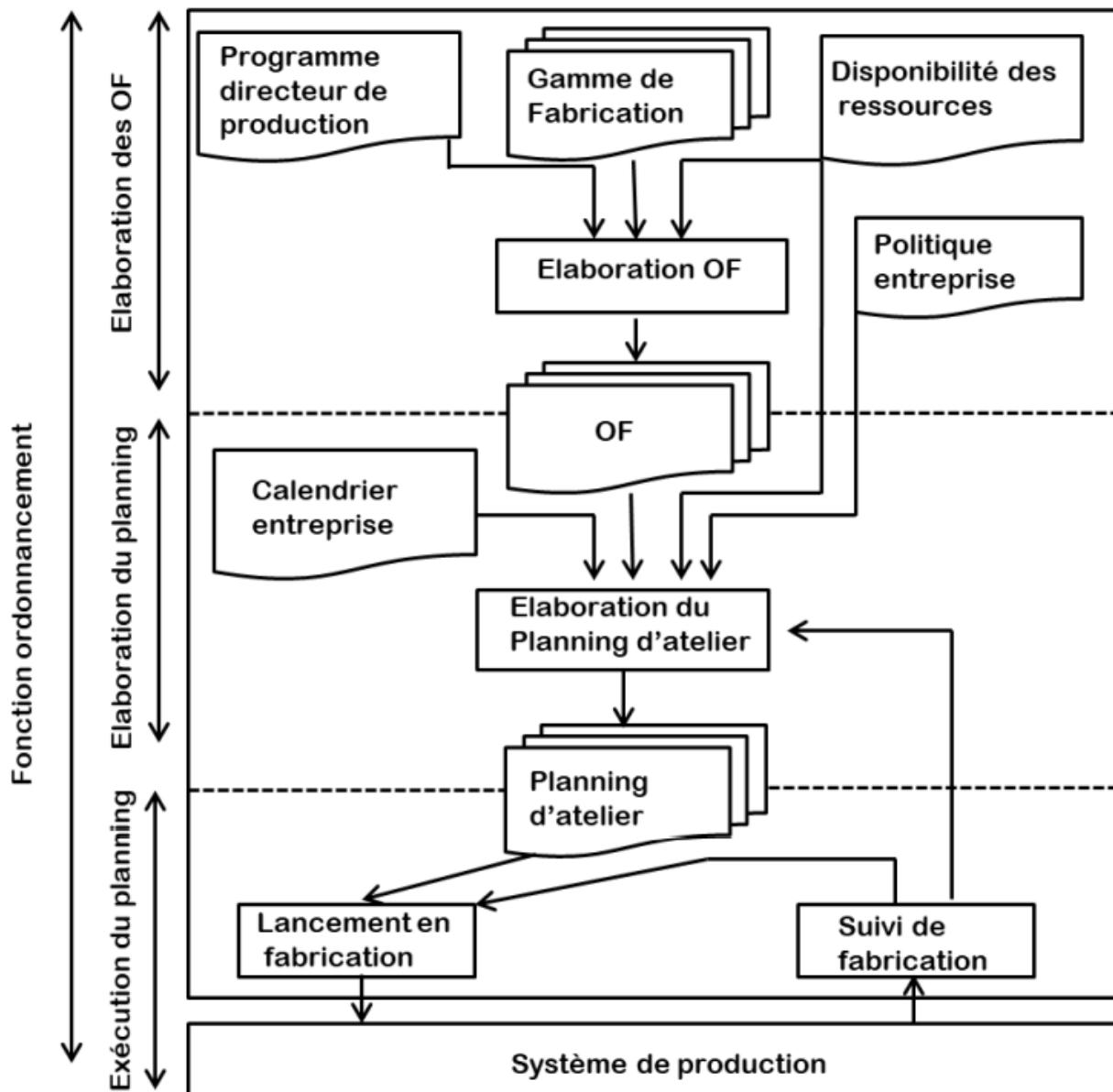


Figure 25: la fonction ordonnancement

- *Problème d'optimisation :*

Optimiser c'est tout d'abord mesurer avant d'agir sur les points critiques. Pour un système de production il faut mettre en place des indicateurs liés à la productivité en fonction des données, aux arrêts machines dont les résultats révèleront les véritables gisements de productivité.

4. Conclusion

Dans ce chapitre nous avons résumé les différentes notions liées aux systèmes de production on a mentionné ses différentes composantes et objectifs, Le deuxième point exposé dans ce chapitre, est la gestion de production et faire valoir l'importance du problème d'ordonnancement au niveau opérationnel. Enfin on a présenté les problèmes des systèmes de production.

Dans le chapitre suivant, nous présentons la notion de l'ordonnancement, et les différents modèles de résolutions des problèmes d'ordonnancement.

Chapitre 3 : Les problèmes d'ordonnancement

1. Introduction

Un problème d'ordonnancement consiste à organiser dans le temps la réalisation d'un ensemble de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînements...) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises

2. Méthodes de résolution des problèmes d'ordonnancement

Les méthodes de résolution des problèmes d'ordonnancement puisent dans toutes les techniques de l'optimisation combinatoire (programmation mathématique, programmation dynamique, procédure par séparation et évaluation, théorie des graphes).

Ces méthodes garantissent en général l'optimisation de la solution fournie. Mais les algorithmes dont la complexité n'est pas polynomiale ne pouvant pas être utilisés pour des problèmes de grande taille, d'où la nécessité de construire des méthodes de résolution approchée, efficaces pour ces problèmes souvent NP-difficiles.

L'analyse d'un problème particulier permet d'obtenir des propriétés sur la structure des solutions optimales, propriétés à partir desquelles on peut ou bien caractériser les solutions optimales ou bien réduire l'espace des solutions à l'explorer (notion de sous-ensemble dominant qui contient au moins une solution optimale).

2.1 Méthode de résolution exacte

Une méthode exacte permet de trouver une solution optimale à un problème donné. Ces méthodes sont généralement utilisées pour résoudre des problèmes de petite taille. Dans ce cas, le nombre de combinaisons possibles est suffisamment faible pour pouvoir explorer l'espace de solutions en un temps raisonnable. On distingue trois sous-classes de méthodes exactes : la procédure de séparation et d'évaluation (Branch and Bound), la programmation dynamique et la programmation linéaire.

- *Branch and bound :*

La procédure de séparation et d'évaluation, également connue sous le nom de "Branch and Bound", est une technique algorithmique de résolution exacte utilisée pour résoudre des problèmes d'optimisation combinatoire. Elle permet de trouver la solution optimale d'un problème en explorant systématiquement l'ensemble de l'espace des solutions possibles.

Voici les étapes principales de la procédure de séparation et d'évaluation (Branch and Bound) :

1. Relaxation continue : On résout une version simplifiée du problème en permettant des solutions fractionnaires.

2 .Évaluation de la solution : On estime la qualité de la solution trouvée en calculant une borne inférieure.

3.Condition d'arrêt : On vérifie si la solution actuelle est la meilleure possible. Si oui, on s'arrête, sinon, on continue.

4.Séparation : On divise l'espace des solutions en sous-ensembles plus petits en ajoutant des contraintes supplémentaires.

5.Branching : On crée de nouveaux sous-problèmes en ajoutant des contraintes, ce qui génère des branches dans l'arbre de recherche.

6.Exploration récursive : On répète les étapes 1 à 5 pour chaque sous-problème créé.

7.Solution optimale : Une fois toutes les branches explorées, on identifie la meilleure solution trouvée jusqu'à présent, qui est la solution optimale.

- *Programmation Dynamique :*

La programmation dynamique est une technique algorithmique utilisée pour résoudre des problèmes d'optimisation en décomposant le problème en sous-problèmes plus petits et en mémorisant les résultats pour éviter de recalculer les mêmes sous-problèmes à plusieurs reprises. Cette approche est souvent utilisée pour résoudre des problèmes de recherche, d'optimisation, et d'analyse de séquences.

Le principe clé de la programmation dynamique est de résoudre les sous-problèmes une seule fois et de stocker leurs solutions dans une table (appelée "tableau de mémoïsation"). Cela permet de réduire considérablement le temps de calcul global et d'éviter des redondances inutiles. La programmation dynamique est particulièrement utile pour les problèmes où les sous-problèmes se recouvrent et peuvent être résolus indépendamment les uns des autres.

- *Programmation Linéaire :*

La programmation linéaire est une méthode mathématique utilisée pour optimiser une fonction linéaire tout en respectant des contraintes linéaires. Elle est utilisée pour résoudre des problèmes d'optimisation où les relations entre les variables et les contraintes peuvent être exprimées sous forme de fonctions linéaires.

Dans un problème de programmation linéaire, on cherche à maximiser ou minimiser une fonction objectif linéaire en ajustant les valeurs des variables tout en satisfaisant un ensemble de contraintes linéaires. Les solutions optimales se trouvent généralement aux sommets d'une région délimitée par les contraintes, ce qui permet de réduire considérablement l'espace de recherche.

2.2 Outils de suivi de Planification

- *Diagramme de Gantt*

Ce type de diagramme a été mis au point par un américain Henry Gantt. On représente au sein d'un tableau en ligne les différentes tâches et en colonne les unités de temps (exprimés en mois, semaines, jours, ...) La durée d'exécution d'une tâche est matérialisée par un trait au sein du diagramme

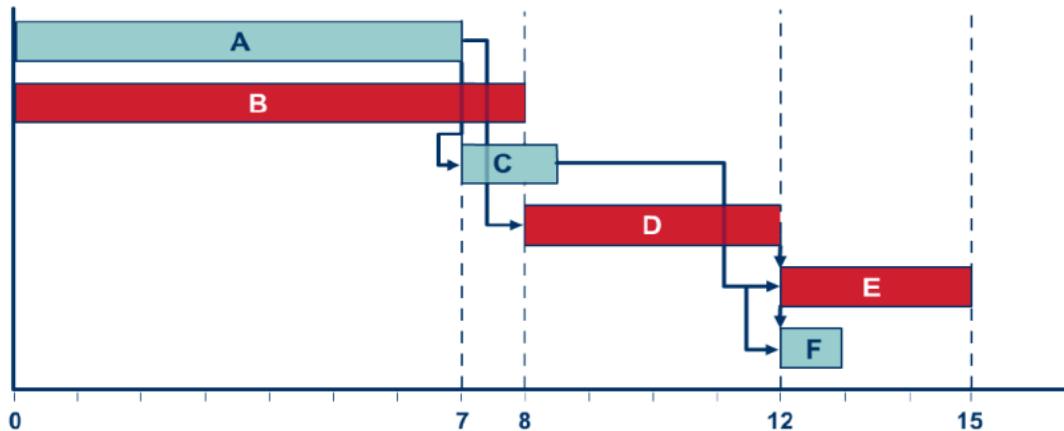
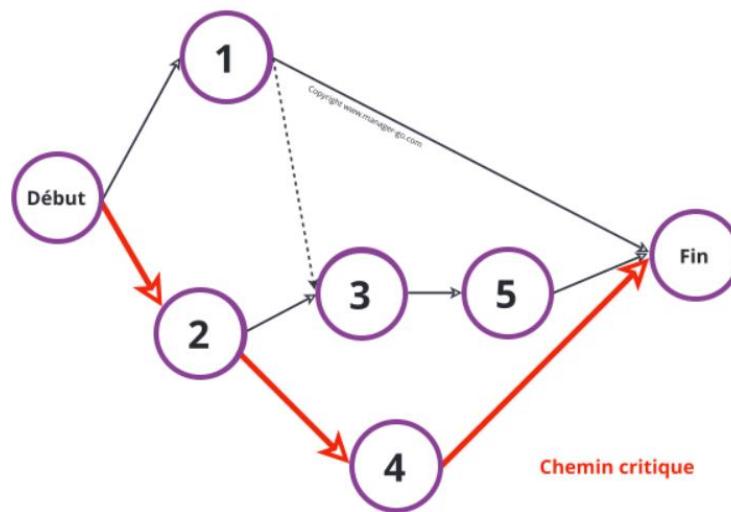


Figure 26: Diagramme Gant

- *Méthode PERT*

C'est une modélisation du problème central de l'ordonnancement par un graphe, elle permet d'évaluer la durée de réalisation d'un projet complexe et de détecter les parties de ce projet ne supportant aucun retard. Ce graphe porte le nom de graphe PERT (Program Evaluation and Review Technique) ou graphe potentiel-étape. Nous donnons ici quelques éléments sur cette modélisation. Dans cette représentation, les arcs sont associés aux tâches ; ils sont valus par la durée des tâches, et les sommets représentent certains événements qui regroupent en général la fin de certaines tâches et le début



d'autres. Figure 27: réseau pert

→ Plusieurs modèles d'ordonnancement et de planification des tâches d'un projet ont été Nous allons aborder dans la suite la résolution par les méthodes approchées qui fournissent Une solution acceptable en temps raisonnable

2.3 Méthodes de résolution approchées

Ces méthodes sacrifient le caractère optimal de la solution pour obtenir, en un temps de calcul raisonnable, des solutions sous-optimales de bonne qualité. Ces méthodes reposent généralement sur un mécanisme de déplacement (aléatoire ou non) dans l'espace des solutions. Elles ne sont pas exactes, mais permettent en général d'obtenir des solutions proches de l'optimum.

- *Les heuristiques :*

Les heuristiques sont des méthodes empiriques basées sur des règles simplifiées pour optimiser un ou plusieurs critères. Le principe général de ces méthodes est d'intégrer des stratégies de décision pour construire une solution proche de l'optimum, tout en essayant de l'obtenir en un temps de calcul raisonnable. On distingue :

- **FIFO (First In First Out)** : la première tâche qui vient est la première tâche ordonnancée,
- **SPT (Shortest Processing Time)** : la tâche ayant le temps opératoire le plus court est traitée en premier lieu
- **LPT (Longest Processing Time)** : la tâche ayant le temps opératoire le plus important est ordonnancée en premier lieu,
- **EDD (Earliest Due Date)** : la tâche ayant la date due la plus petite est la plus prioritaire,
- **SRPT (Shortest Remaining Processing Time)** : cette règle, servant à lancer la tâche ayant la plus courte durée de travail restant à exécuter, est très utilisée pour minimiser les encours et dans le cas des problèmes d'ordonnancement préemptifs
- **ST (Slack Time)** : à chaque point de décision, l'opération ayant la plus petite marge temporelle est prioritaire. Faute de disponibilité des ressources de production, cette marge peut devenir négative.
- **Les algorithmes gloutons** : sont parmi les schémas heuristiques les plus simples et les plus rapides. Ils construisent une solution de manière itérative sans jamais remettre en cause les décisions prises à l'itération antérieure. Ces algorithmes construisent une solution élément par élément. A une itération donnée, on détermine l'élément à inclure dans la solution partielle en évaluant le coût de la nouvelle solution partielle qui inclut cet élément. L'élément engendrant la plus petite augmentation du coût est choisi. Malheureusement, il n'est pas possible de connaître, a priori, l'impact des décisions prises à une itération donnée sur le long terme. En outre, il est possible qu'à une itération donnée aucun élément ne soit insérable

- *Les Méta-Heuristique :*

Les méta-heuristiques sont des stratégies de recherche itératives, destinées à l'exploitation de l'espace de solutions par l'utilisation de différentes techniques. Ces méthodes n'essayeront pas d'examiner toutes les solutions possibles, mais de trouver dans un temps raisonnable une solution satisfaisante par investigation intelligente de l'espace. L'ensemble des méta-heuristiques proposées dans la littérature sont partagées en deux catégories : des méta-heuristiques à base de solution unique, des méta-heuristiques à base de population de solution

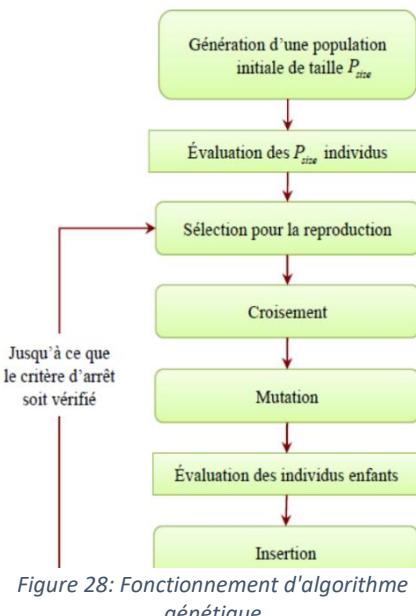
- **Algorithmes génétiques**

Les algorithmes génétiques s'inspirent de la théorie de l'évolution et des règles de la génétique qui expliquent la capacité des espèces vivantes à s'adapter à leur environnement par la combinaison des mécanismes suivants :

- La sélection naturelle : fait que les individus les mieux adaptés à l'environnement tendent à survivre plus longtemps et ont donc une plus grande probabilité de se reproduire
- La reproduction par croisement : fait qu'un individu hérite ses caractéristiques de ses parents, de sorte que le croisement de deux individus bien adaptés à leur environnement aura tendance à créer un nouvel individu bien adapté à l'environnement ;
- La mutation : fait que certaines caractéristiques peuvent apparaître ou disparaître de façon aléatoire, permettant ainsi d'introduire de nouvelles capacités d'adaptation à l'environnement, capacités qui pourront se propager grâce aux mécanismes de sélection et de croisement.

Les algorithmes génétiques reprennent ces mécanismes pour définir une méta-heuristique de résolution de problèmes d'optimisation combinatoire. L'idée est de faire évoluer une population de combinaisons, par sélection, croisement et mutation, la capacité d'adaptation d'une combinaison étant ici évaluée par la fonction objective à optimiser.

La figure présente un schéma de fonctionnement général de l'algorithme génétique.



- **Recherche Taboue**

La recherche taboue est également une des méta-heuristiques les plus connues, elle a été introduite par Glover en 1986. Elle utilise un historique de manière à interdire à l'algorithme de revenir sur ses pas. Cet historique se traduit par la présence d'une liste dite 34 tabou qui garde une trace des dernières solutions visitées, ainsi l'algorithme ne pourra plus explorer ces solutions (du moins à court terme, tout dépend de la taille de la liste tabou). Dans le cas d'un problème d'ordonnancement, la liste taboue peut contenir, par exemple, les informations suivantes : la solution (ordonnancement +affectation) le cout de la solution et les attributs de mouvements utilisés pour aboutir à cette solution. Les systèmes de voisinage utilisés sont le changement de deux pièces contigües, deux pièces quelconque et insertion d'une pièce dans une autre position. L'algorithme fonctionne comme suit. Initialement la liste taboue est vide et on génère une solution S à l'aide d'une heuristique quelconque ; S devient la solution courante. A chaque itération, on choisit le meilleur voisin S0 de S qui n'est pas déjà dans la liste tabou, S0 devient la solution courante S et est ajouté à la liste taboue. Si la taille de la liste taboue dépasse la taille maximale autorisée, on supprime de cette liste l'élément le plus ancien (stratégie FIFO - First In First Out). Notons que la liste taboue permet d'éviter les cycles en interdisant de choisir une solution dans le voisinage de la solution courante qui aurait déjà été explorée. En outre, en pratique, on ne stocke pas les solutions dans leur intégralité (trop coûteux en temps et en espace) mais seulement une signature de ces solutions.

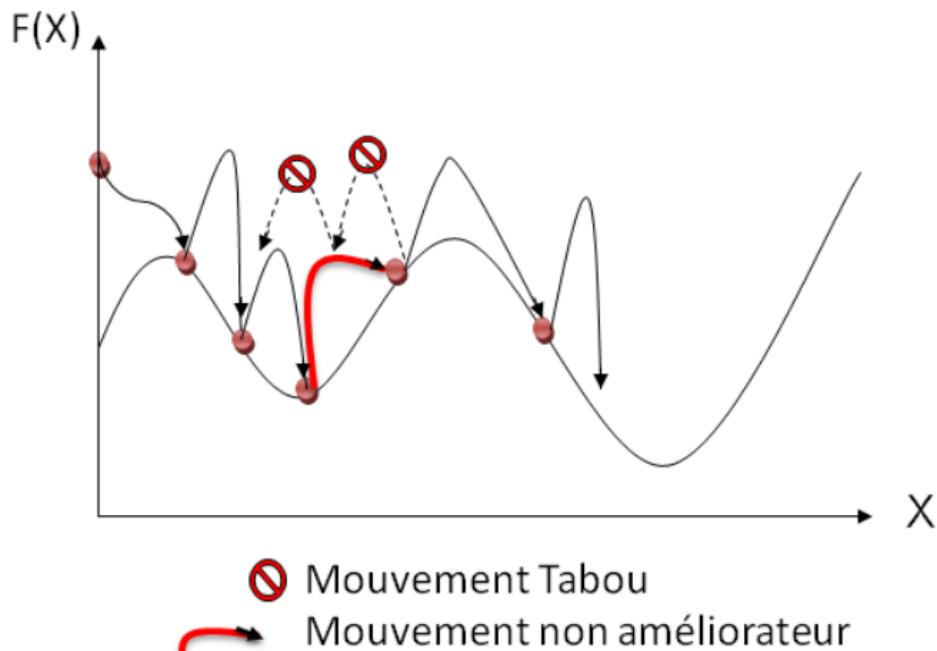


Figure 29: recherche taboue

3. Conclusion

Après la présentation des problèmes d'ordonnancement et de leurs principales caractérisations, différentes méthodes, exactes et approchées, pouvant être utilisées pour la résolution de ces problèmes sont introduites dans ce chapitre. Parmi les méthodes exactes, nous avons distingué le diagramme de gant et la méthode pert. Parmi les méthodes approchées ou métaheuristiques, nous avons présenté les méthodes de recherche locale telles que la méthode de recherche tabou, et les méthodes évolutionnistes, telles que les algorithmes génétiques

Dans le prochain chapitre on va utiliser la méthode génétique afin de résoudre un problème d'ordonnancement dans l'industrie sivo et on va le comparer avec la méthode d'ordonnancement utiliser en ce moment par cette usine.

Chapitre 4 : description du problème et résolution avec l'algorithme génétique :

1 Introduction

La résolution des problèmes est une discipline cruciale pour l'amélioration des processus et des décisions dans de nombreux domaines. Elle repose sur l'utilisation des algorithmes pour trouver la meilleure solution possible parmi un ensemble de possibilités, en respectant des contraintes spécifiques. Les météahéuristiques permettent de trouver une ou plusieurs solutions proches de l'optimum pour des problèmes d'optimisations. Le principe d'une météahéuristique est de trouver un minimum global à un problème de minimisation et de ne pas rester bloqué sur un minimum local. Ainsi, les météahéuristiques ont pour ambition commune de résoudre au mieux les problèmes d'optimisation qualifiés de difficiles.

Dans ce chapitre, un problème d'ordonnancement multi-objectifs d'un atelier de conditionnement de type job-shop flexible en industrie sivo est présenté. Les différentes caractéristiques et opérations le concernant sont introduites puis sa formulation et sa résolution par les algorithmes génétiques proposées.

2 Ordonnancement en industrie sivo

L'industrie construit 3 types de verres : Minéral, Conventionnel et Free Forme. Puisque le type le plus demandé est le Free Forme, on va focaliser dans notre projet seulement sur ce type de verre. Le système de production possède 3 types d'ordre :

- OF : ordre de fabrication
- OT : ordre de traitement
- OS : ordre de stock (La quantité des OS est négligeable par rapport aux OT et les OF)

Les OF ont la possibilité de passer par les 2 ateliers mais les OT ne peuvent passer que par l'atelier de coloration et l'atelier d'anti reflet.

Pour les ordres de fabrication de 8h à 21h, les ordres ayant le score le plus élevé sont traités en premier lieu. Cette mesure vise à donner la priorité aux ordres les plus importants ou urgents, afin de s'assurer qu'ils soient terminés dans les délais requis. Cela permet de maximiser l'efficacité et la satisfaction des clients.

Cependant, une particularité intervient dans le processus d'ordonnancement. En dehors de la plage horaire de 8h à 21h, l'ordre de séquence est inversé. Cette inversion a été modélisée en inversant le signe du score associé aux ordres de fabrication. Ainsi, les ordres qui avaient initialement un score élevé se retrouvent avec un score négatif, tandis que ceux qui avaient un score plus faible obtiennent un score positif. Cette inversion permet de créer un équilibre et de donner une chance équitable à tous les ordres de fabrication, même en dehors des heures de pointe.

En ce qui concerne les ordres de travail (OT), ils peuvent être lancés tout au long de la journée, à l'exception de la période allant de 8h à 14h. Cela peut être dû à des considérations pratiques, à la disponibilité des ressources ou à d'autres facteurs spécifiques à l'atelier. L'objectif est d'optimiser l'utilisation des ressources disponibles et de répartir la charge de travail de manière équilibrée sur la journée.

D'autre part, les (OS) sont lancés à des cadences plus réduites. Cette approche permet de contrôler le flux de production, de s'assurer que les étapes de fabrication sont réalisées de manière précise et de garantir la qualité des produits finis.

En somme, l'ordonnancement en atelier dans ce contexte spécifique implique de prendre en compte les scores des ordres de fabrication, de leur donner la priorité pendant les heures de pointe, tout en inversant leur séquence en dehors de cette période.

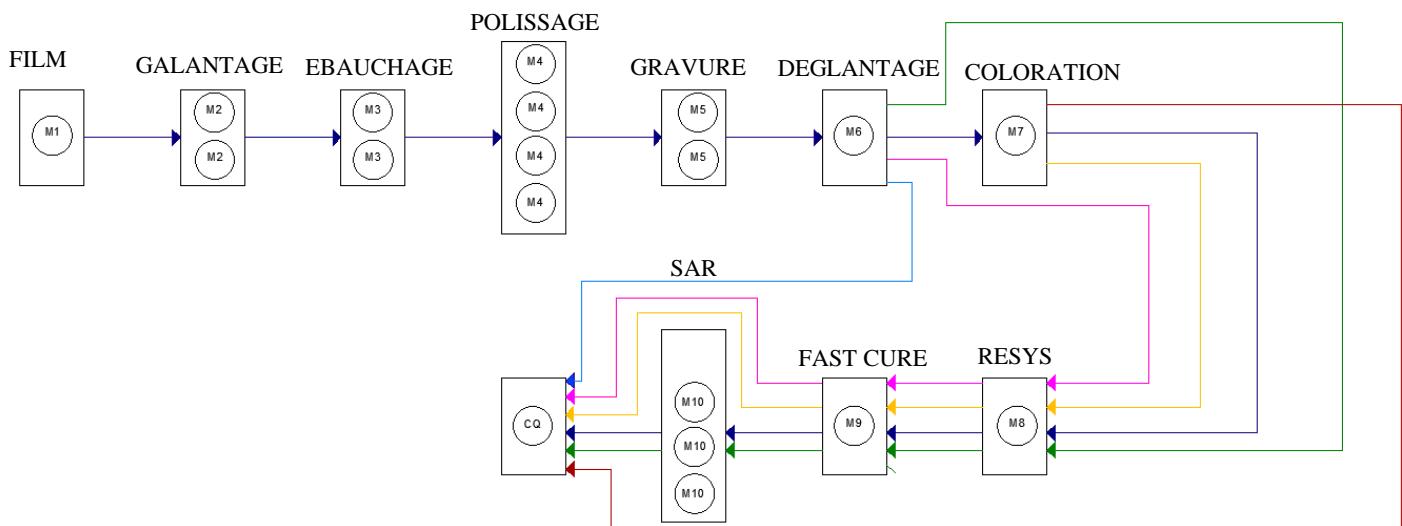


Figure 30: flux de production atelier sivo

3 Type de produit Essilor sivo

Le verre peut être classé en deux grandes familles : les verres organiques ou plastiques et les verres minéraux.

- *Les verres minéraux :*

Ce sont des verres dérivant du sable et contenant des adjuvants pour améliorer leurs performances optiques. C'est une matière d'une et résistance aux rayures mais cassante.

- *Les verres organiques*

La matière qui constitue ces verres est à base de polymère. Cette matière présente l'avantage de grande résistance à la casse et procure plus de sécurité notamment pour les enfants. Cependant, elle est tendre, ce qui la rend vulnérable aux rayures.

➔ Il est important de remarquer que tous ces types de produits passent pratiquement par les mêmes étapes de construction. La seule différence réside dans le type d'anti-reflet utilisé, ce qui entraîne la production de plusieurs types de produits à vendre (rq dans notre étude on va se concentrer aux verres organiques à cause de ça cadences très grande par rapport aux verres minéraux)

Tableau 2: les types d'anti reflet

TYPES D'ANTI REFLET :	
1	prevencia
2	sapphire
3	drive
4	optifog
5	miroir
6	blue
7	Groupe(alize, alizeuv ,premium ,premiumclean ,premium clean uv crizal,crizal uv,crizal easy,crizal kids)

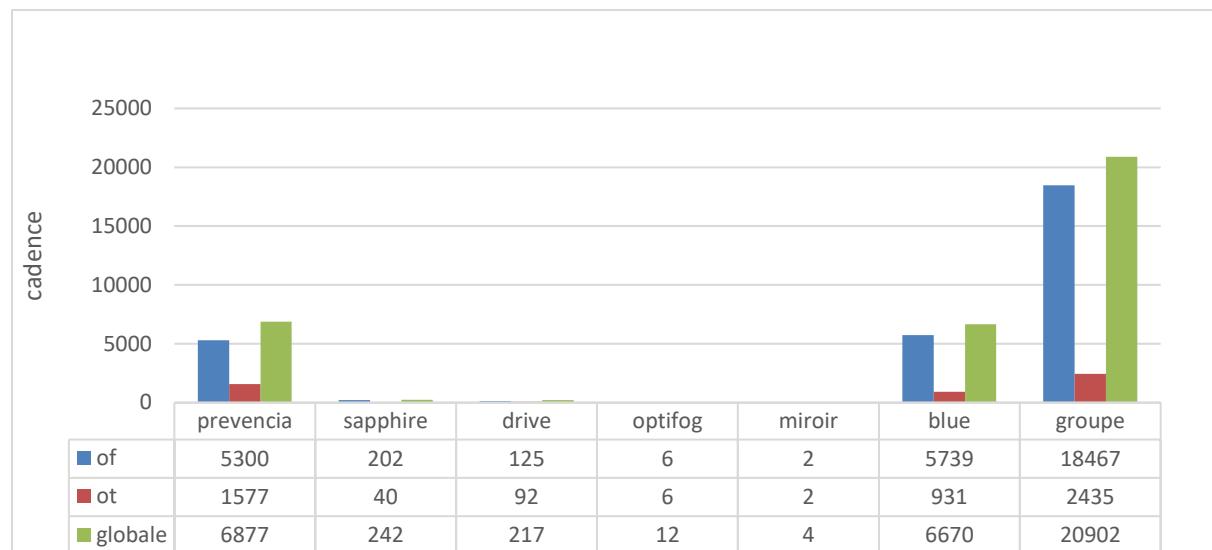


Figure 31: cadence des produits sivo

4 Problème survenant dans l'atelier sivo

4.1 Analyse de l'existant et chronométrage

Nous avons récemment constaté un léger retard dans notre industrie. Par conséquent, nous avons entrepris de nous concentrer sur le problème. Après avoir analysé attentivement la situation actuelle et effectué une phase de chronométrage, nous avons pu identifier l'atelier de traitement comme étant la principale source du problème. Nous avons remarqué un gonflement de la marge de temps allouée à cet atelier par rapport à l'atelier de surfacage. Nous nous efforçons maintenant de résoudre ce problème afin d'améliorer l'efficacité globale de notre processus de production.

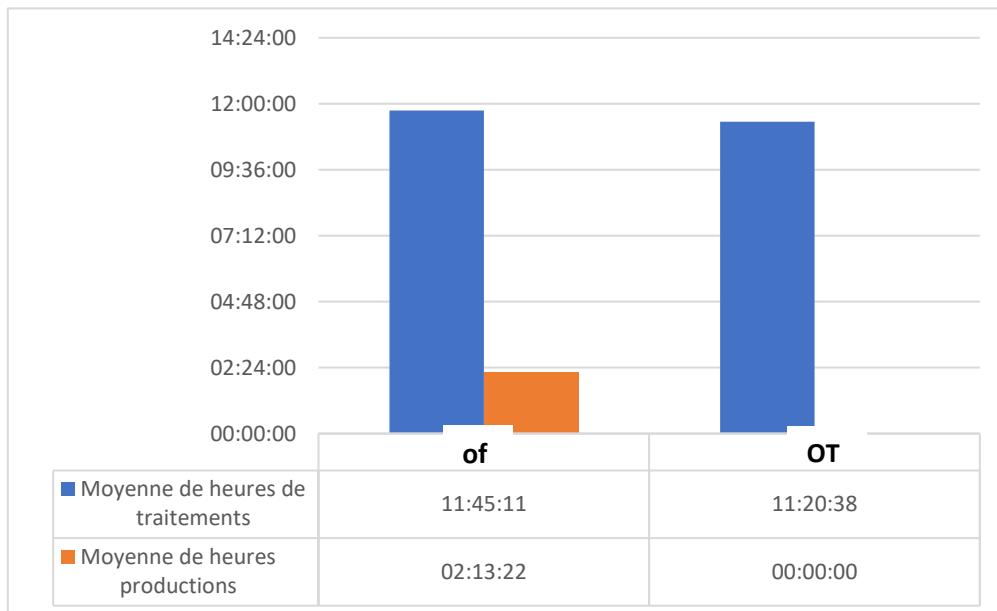


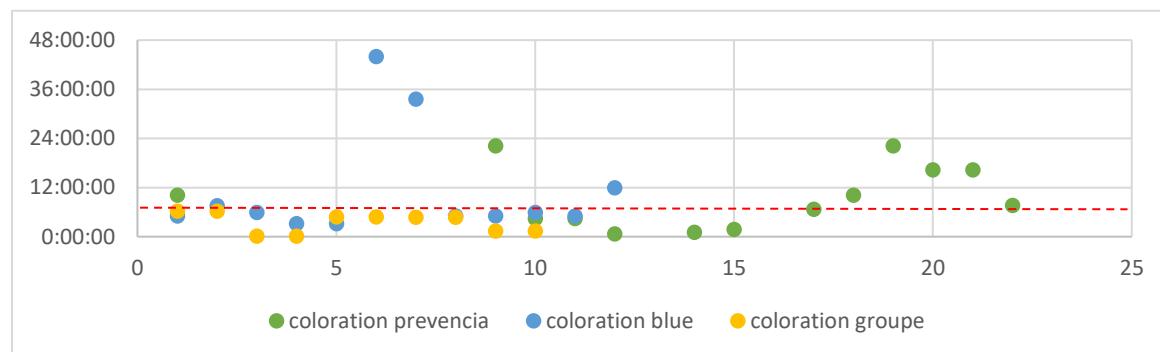
Figure 32: cadence des OF et OT

4.2 Analyse du temps de traitement dans l'atelier : Identification du goulot d'étranglement

Après avoir concentré nos efforts sur l'atelier de traitement, nous avons entrepris d'identifier précisément le problème. Nous avons examiné attentivement les différentes phases de traitement, notamment la coloration, resys et l'anti-reflet. Notre objectif était de déterminer le temps moyen nécessaire pour chacune de ces phases, afin d'identifier le goulot d'étranglement dans notre processus de production.

En effectuant une analyse approfondie, nous avons découvert que le problème résidait principalement dans la phase de traitement anti reflet. Cette phase prenait significativement plus de temps que le resys et la coloration, ce qui provoquait un déséquilibre dans le flux de travail. Cette disparité temporelle constituait un obstacle majeur à une efficacité optimale de l'ensemble de l'atelier de traitement.

À la lumière de ces découvertes, nous sommes déterminés à trouver des solutions pour réduire le temps de traitement dans la phase d'anti reflet. En identifiant le goulot d'étranglement, nous sommes en mesure de concentrer nos efforts sur cette zone critique et d'apporter des améliorations ciblées. Notre objectif final est d'optimiser l'ensemble du processus de production dans l'atelier de traitement, afin de garantir des délais plus courts, une meilleure efficacité et une satisfaction accrue de nos clients.



→ Pour pouvoir faire face à la concurrence, sivo doit maintenir un taux de productivité en constante évolution; des faiblesses peuvent toutefois apparaître au niveau du système de production et plus particulièrement au niveau du poste d'anti reflet et engendrer des coûts de production ainsi que des coûts de non utilisation relativement élevés ,en effet la poste de traitement de l'anti-reflet dispose d'un goulot d'étranglement qui entraîne une perte de temps considérable, car il faut attendre que 55 pièces du même type soient collectées avant de passer à l'étape suivante. Cette étape est nécessaire pour optimiser l'efficacité du processus de traitement et garantir une utilisation efficace des ressources. Cependant, cette attente peut parfois entraîner des délais supplémentaires dans la production globale, ce qui nécessite une planification minutieuse pour assurer une gestion efficace des délais et des commandes clients.

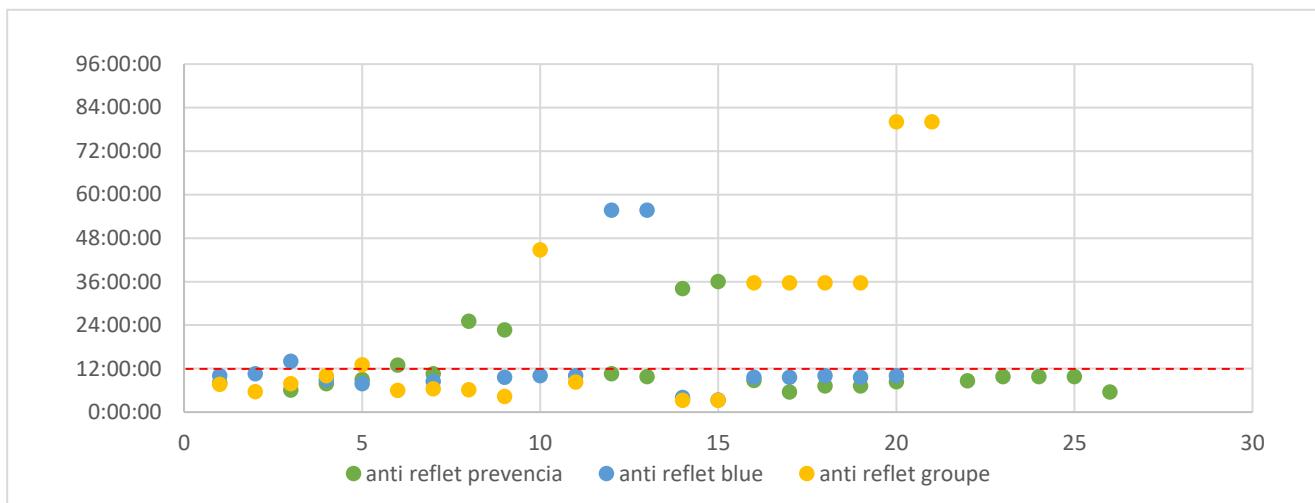


Figure 35: temps moy d'anti reflet

5 Formulation du Problème

5.1 Présentation de problème à étudier

Dans ce travail nous nous sommes intéressés à un problème d'ordonnancement de type job shop où les produits sont transportés dans un sens unidirectionnel.

L'objectif de notre travail est de trouver une solution efficace pour un ordonnancement des produits tout en minimisant la somme des dates de fin d'exécution (ΣC_j) c'est-à-dire de trouver le meilleur ordonnancement faisable qui minimise cette somme et qui respecte les contraintes imposées.

Le problème est défini par Job shop hybride.

Le problème de job shop classique est NP difficile pour cela on peut dire que notre problème est aussi NP difficile puisque ce problème classique peut être classifié en deux larges familles de fonction objective : minimiser le makespan et minimiser la somme des dates de fin d'exécution.

Fonction objective: $[Max(C_1/J_1, C_2/J_2, \dots, C_n/J_n)]$

Sous contraintes :

$$C_{ij} - S_{ij} - P_{ij} = 0 \quad \forall i, j \quad (1)$$

$$S_{ij+1} - C_{ij} \geq 0 \quad (2)$$

$$S_{ij} \geq 0 \quad (3)$$

Signification des équations :

- J : Représente un job (ou tâche) dans le problème d'ordonnancement.
- C_{ij} : Désigne le temps de compléction (ou date de fin) de l'opération O_{ij} du job i sur une machine donnée.
- S_{ij} : Décrit la date de début de l'opération O_{ij} du job i sur une machine donnée.
- P_{ij} : Indique la durée de l'opération O_{ij} du job i sur une machine donnée.

Contrainte 1: contrainte de précédence entre les opérations d'un même job : Une opération ne peut commencer à être exécutée avant la fin du traitement de l'opération précédente du même job

Contrainte 2 : Cette contrainte assure que la valeur du makespan (le temps d'exécution total du processus) doit être supérieure ou égale aux dates de fin des dernières opérations pour tous les jobs. Cela garantit que le processus est terminé après l'achèvement de toutes les opérations

Contrainte 3 : Cette contrainte garantit que la date de début de chaque opération doit être positive ou nulle, ce qui est logique puisque l'opération ne peut pas commencer avant le début du temps.

6 Résolution du problème d'ordonnancement en industrie sivo par l'algorithme génétique

6.1 Introduction

D'après les informations précédemment mentionnées, il semble que notre problème soit lié à l'ordonnancement de l'envoie des commandes dans un environnement de type job shop flexible. Le job shop est un système de production où différentes tâches doivent être exécutées sur différentes ressources dans un ordre spécifique pour atteindre les objectifs de production souhaités. Dans notre cas, l'envoie des différentes commandes selon ces types d'anti reflet nécessitent une planification précise pour optimiser l'efficacité du processus de fabrication, Cette phase est réservée à la présentation AGP et sa résolution. Sa validation à consister à la comparaison des résultats obtenus par l'AGP avec celles obtenues avec l'algorithme d'ordonnancement précédent

6.2 Principe d'algorithme génétique

Pour résoudre le problème d'ordonnancement à ce niveau de poste, une des méthodes d'optimisation possibles est l'utilisation des algorithmes génétiques. Les algorithmes génétiques sont des techniques basées sur la théorie de l'évolution qui cherchent à trouver des solutions optimales en utilisant des processus d'héritage, de mutation et de sélection. En appliquant ces algorithmes à notre problème d'ordonnancement, nous pouvons générer des plans de production plus efficaces en explorant différentes combinaisons de séquences d'envoie des tâches et en évaluant leur performance. Cela permet d'identifier les meilleures solutions possibles pour minimiser les temps d'attente et maximiser l'utilisation des ressources. En utilisant les algorithmes génétiques, nous pouvons ainsi améliorer l'efficacité de notre processus de fabrication et optimiser notre chaîne de production

- *Historique :*

En 1859 les bases de l'évolution étaient posées par C. Darwin avec son idée de la sélection naturelle (« dans tout espèce les meilleurs sont sélectionnés »). En 1901 ceux sont les bases de la génétique qui étaient posées par De-Vries suite à sa théorie du mutationnisme. En 1975 Jhon Holland proposa l'Algorithme Génétique. En 1989 Goldberg exposa les fondements mathématiques des algorithmes génétiques

- *Terminologie*

Les algorithmes génétiques sont basés sur cinq niveaux d'organisation comme présenté sur la figure

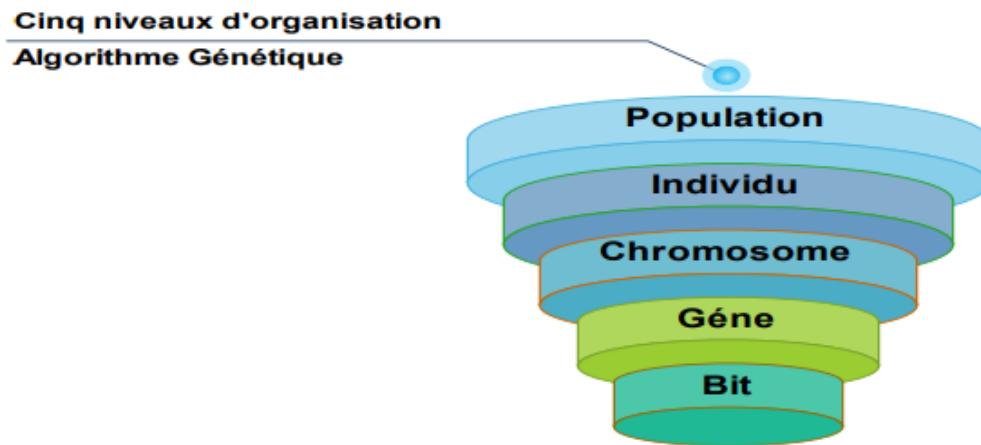


Figure 36: terminologie d'algorithme génétique

- **Bit** : Est la plus petite unité pour présenter une donnée.
- **Gène** : A chaque variable d'optimisation x_i , nous faisons correspondre un gène.
- **Chromosome** : Un chromosome est un ensemble de gènes, (figure 4.2). Chaque dispositif est représenté par un chromosome.
- **Individu** : Un individu est constitué d'un ou plusieurs chromosomes
- **Population** : Une population est un ensemble de N individus qui vont évoluer.

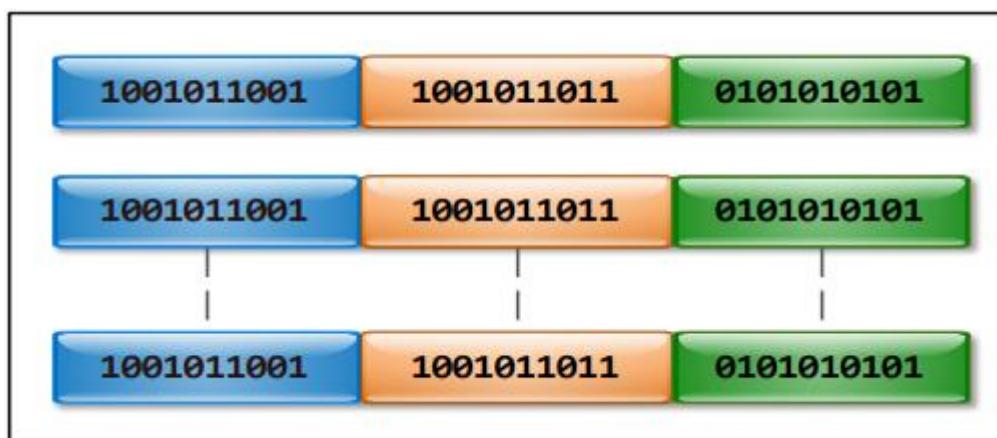


Figure 37: exemple de population

- *Concepts de base :*

Un AG recherche le ou les extrema d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des éléments suivants :

- Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers la solution optimale
- Une fonction à optimiser. Celle-ci est appelée fitness ou fonction d'évaluation de l'individu. Elle est utilisée pour sélectionner et reproduire les meilleurs individus de la population
- Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'état

Des paramètres de dimensionnement : taille de la population, nombre total de générations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation.

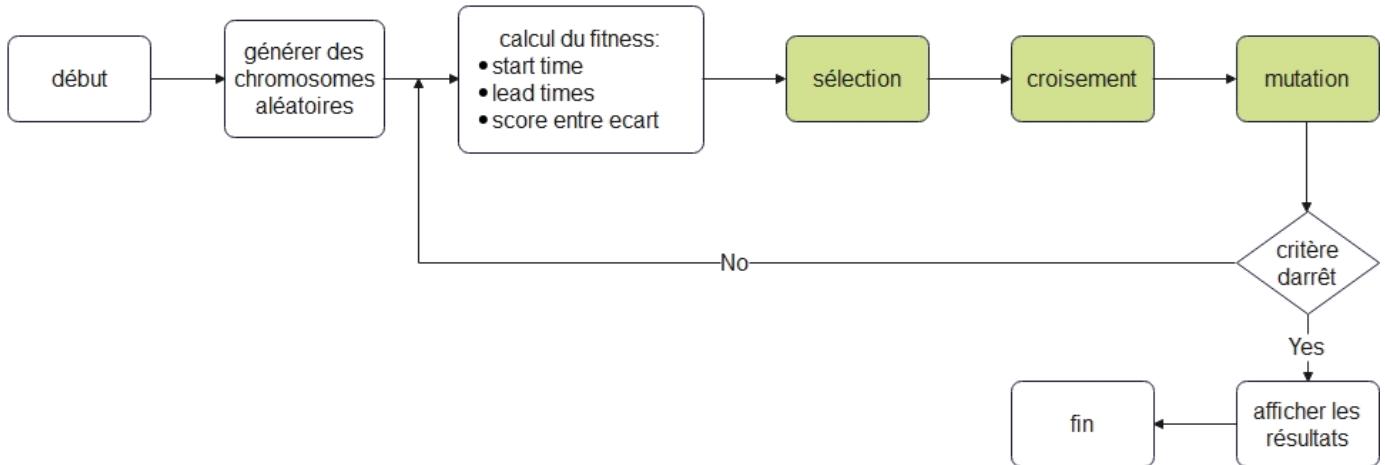


Figure 38: structure de notre algorithme

- *Le codage*

Chaque paramètre d'une solution est assimilé à un gène, toutes les valeurs qu'il peut prendre sont les allèles de ce gène, on doit trouver une manière de coder chaque allèle différent de façon unique (établir une bijection entre l'allèle "réel" et sa représentation codée).

- Codage CLOS proposé :

Le codage pris en compte dans le problème d'ordonnancement en industries sivo, considéré dans cette étude, est le Codage en Liste des Opérations Structuré (CLOS). Dans le cas de plusieurs machines, ce codage consiste à proposer, pour chaque individu, une structure de données contenant le type d'anti reflet des produit traité ainsi que ses dates de début et de fin de fabrication

Produit 1	prevencia	drive	optifog	miroir
Produit 2	blue	miroir	prevencia	optifog
...				
Produit 3	drive	prevencia	miroir	blue
Produit 4	optifog	drive	blue	prevencia
...				
Produit 5	miroir	blue	prevencia	drive

- Calcul de la fonction d'évaluation fitness à optimiser :

Pour évaluer la fonction fitness, nous avons utilisé une approche basée sur le calcul de l'écart entre les différentes commandes. Cet écart est déterminé à l'aide de la formule suivante : "écart = (due date - (current date + lead time)) / priorité". Plus l'écart est petit, plus la commande sera exécutée en priorité par rapport aux autres.

Une fois que nous avons calculé l'écart pour chaque solution, nous attribuons un score à chacune d'entre elles. Ce score est calculé en faisant la somme des différences entre les gènes successifs de la solution : "Score = \sum (différence entre gènes successifs)". Cette formule nous permet de déterminer la meilleure solution, celle qui présente la configuration la plus optimale en termes de séquence de gènes.

Dans une deuxième étape, nous calculons le taux de commandes exécutées dans les 24 heures ($rx24$) pour chaque solution. Cela est réalisé en utilisant la formule suivante : " \sum (date de création + délai < 24H) / nombre total". L'objectif est de sélectionner la meilleure solution qui permettra d'augmenter le taux de commandes exécutées dans les 24 heures.

En combinant ces différentes mesures et en analysant les résultats, nous sommes en mesure d'identifier la solution optimale qui maximisera à la fois la priorité des commandes et le taux d'exécution dans les délais. Cette approche nous aide à prendre des décisions éclairées pour améliorer notre processus de gestion des commandes et garantir une satisfaction client optimale.

- *La sélection :*

Cet opérateur est chargé de définir quels seront les individus d'une population P qui vont être dupliqués dans la nouvelle population P' et vont servir de parents ; ce dernier permet aussi aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale la probabilité de survie d'un individu sera directement reliée à son efficacité relative au sein de la population.

On va essentiellement utiliser le type de **sélection par tournoi** :

La sélection par tournoi est l'une des sélections les plus utilisées dans les algorithmes génétiques. Le principe consiste à choisir un sous-ensemble d'individus (S individus) aléatoirement dans la population, puis à sélectionner le meilleur individu dans ce groupe en fonction de son fitness. Ce processus est répété jusqu'à l'obtention du nombre d'individus requis. Le nombre de participants à un tournoi (S), appelé taille du tournoi, est utilisé pour faire varier la pression de cette sélection. Si ce nombre est grand, alors la pression sera forte et « les faibles individus auront une petite chance d'être choisis. En général, un seul gagnant est choisi parmi les participants à un tournoi. Ce gagnant peut être choisi d'une façon déterministe ou probabiliste. Dans le cas déterministe, qui est pratiquement le plus utilisé le gagnant est l'individu de meilleure qualité (meilleure fitness). Dans le cas probabiliste, chacun des participants peut être choisi en tant que gagnant avec une probabilité proportionnelle à son fitness.

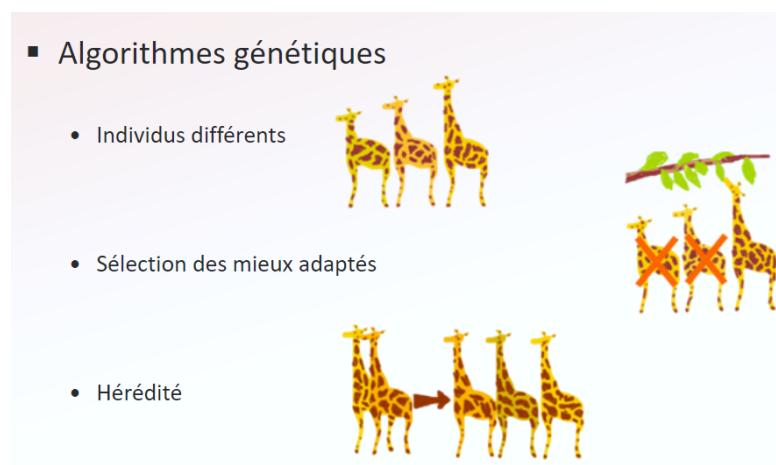


Figure 39: exemple de sélection

- *Le croisement ou crossover avec un point :*

Le croisement utilisé par les algorithmes génétiques est la transposition informatique du mécanisme qui permet, dans la nature, la production de chromosomes qui héritent partiellement des caractéristiques des parents. Son rôle fondamental est de permettre la recombinaison des informations présentes dans le patrimoine génétique de la population. En effet, plus le nombre de points de croisements sera grand et plus la probabilité de croisement sera élevée plus il y aura d'échange de segments, donc d'échange de paramètres, d'information, et plus le nombre de points de croisements sera petit et plus la probabilité de croisement sera faible, moins le croisement apportera de diversité

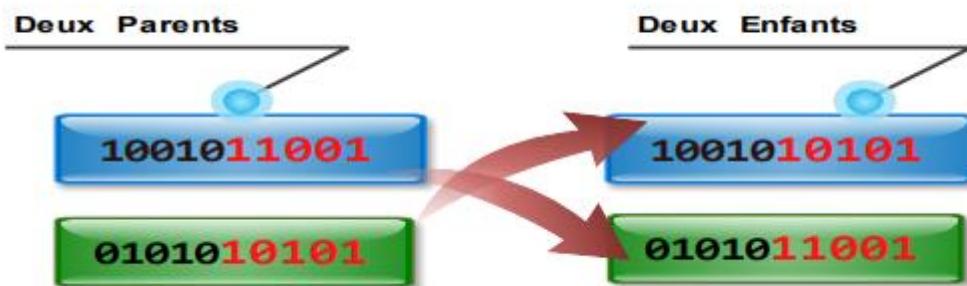


Figure 40: croisement avec un point

- *La mutation uni point :*

Cet opérateur consiste à changer la valeur allélique d'un gène avec une probabilité pm très faible, généralement comprise entre 0.01 et 0.001. Une mutation consiste simplement en l'inversion d'un bit (ou de plusieurs bits, mais vu la probabilité de mutation c'est extrêmement rare) se trouvant en un locus bien particulier et lui aussi déterminé de manière aléatoire ; on peut donc résumer la mutation de la façon suivante :



Figure 41: mutation uni point

6.3 Phase de test

Dans le processus de sélection des méthodes de sélection, de croisement et de mutation pour notre algorithme génétique, nous avons suivi une approche basée sur des expérimentations et des comparaisons. L'objectif principal était de choisir les méthodes qui permettraient d'obtenir les meilleurs résultats en termes d'efficacité et de performance pour résoudre notre problème d'ordonnancement.

Pour commencer, nous avons identifié un ensemble de méthodes de sélection, de croisement et de mutation qui sont couramment utilisées dans les algorithmes génétiques. Nous avons ensuite implémenté ces différentes méthodes dans notre algorithme et effectué une série de tests sur des ensembles de données représentant des instances du problème d'ordonnancement que nous voulions résoudre.

Pour chaque méthode, nous avons exécuté l'algorithme génétique et recueilli les résultats obtenus, notamment les solutions optimales. Ensuite, nous avons comparé les résultats obtenus avec notre objectif principal, c'est-à-dire RX 24

Cette approche itérative de test et de comparaison nous a permis d'identifier les méthodes les plus adaptées à notre problème et d'optimiser ainsi la performance de notre algorithme génétique pour résoudre efficacement les instances du problème d'ordonnancement qui nous intéressaient.

- *Phase de test du Sélection :*

On trouve essentiellement quatre types de méthodes de sélection différentes :

- ***La méthode de la "loterie biaisée" :***

Dans cette méthode chaque individu a une chance d'être sélectionné proportionnelle à sa performance, donc plus les individus sont adaptés au problème, plus ils ont de la chance d'être sélectionnés. Par analogie à la "roue du forain", chaque individu se voit attribué un secteur dont l'angle est proportionnel à son adaptation, son "fitness". On fait tourner la roue, à son arrêt un secteur est désigné par un curseur l'individu choisi et celui correspondant au secteur.

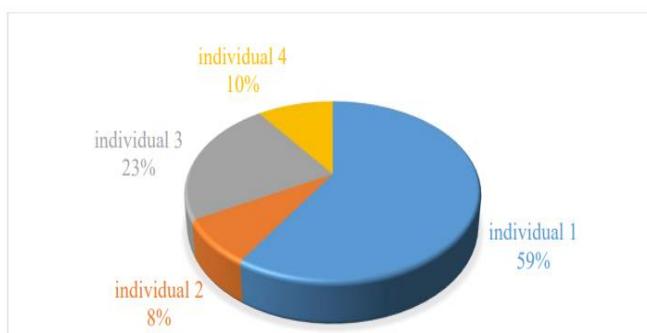


Figure 43:roue du forain

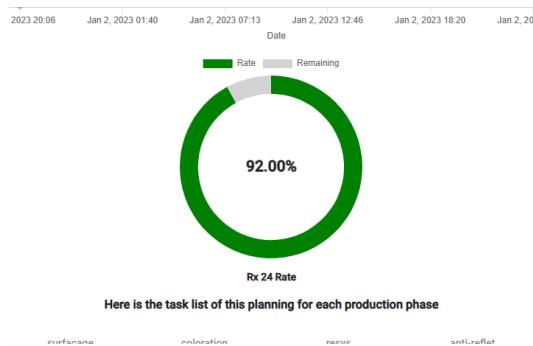


Figure 42:pourcentage RX 24 avec loterie de biaisée

- **La méthode "élitiste" :**

Cette méthode consiste à sélectionner les n individus dont on a besoin pour la nouvelle génération P' en prenant les n meilleurs individus de la population P après les avoir triée de manière décroissante selon la fitness de ses individus. Il est inutile de préciser que cette méthode est encore pire que celle de la loterie biaisée dans le sens où elle amènera à une convergence prématuée encore plus rapidement et surtout de manière encore plus sûre que la méthode de sélection de la loterie biaisée ; en effet, la pression de la sélection est trop forte, la variance nulle et la diversité inexiste

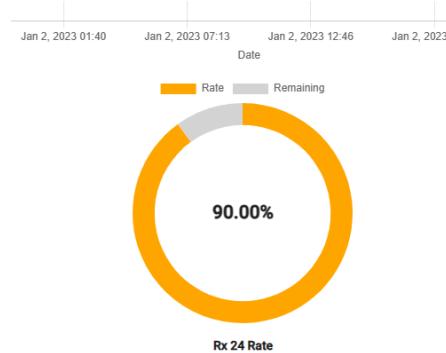


Figure 44: pourcentage RX 24 avec méthode élitiste

- **La sélection universelle stochastique :**

La sélection stochastique universelle, ou SUS, est une méthode de sélection éliminant le biais de la sélection par roulette. A chaque individu de la population est associé un segment d'une longueur égale à son fitness. L'ensemble des segments est placé de manière contiguë pour former une ligne de longueur égale à la somme des fitness, comme dans la sélection par roulette. La différence réside, ici, en l'ajout à intervalles réguliers d'un ensemble de pointeurs situés au-dessus de la concaténation des segments. Le nombre de pointeurs doit être égal au nombre d'individus à sélectionner. La distance les séparant équivaut, ainsi, à l'inverse du nombre d'individus à sélectionner. La position du premier pointeur est donnée par un nombre aléatoire compris entre 0 et 1/N.

→ cette méthode est très efficace au niveau de pourcentage de RX 24 mais il consomme beaucoup de temps



Figure 45: pourcentage RX 24 avec méthode stochastique

- **La sélection par tournois :**

On a expliquer le fonctionnement de cette méthode dans la partie précédente et on guise de conclusion on l'a choisi car il nous a donné la meilleur pourcentage de RX 24 en un temps acceptable par rapport au autre méthode

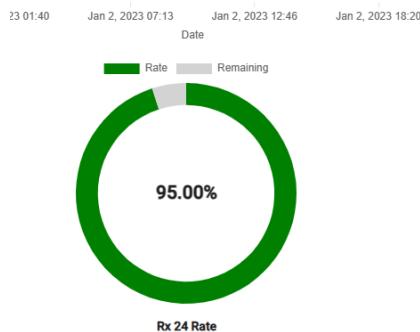
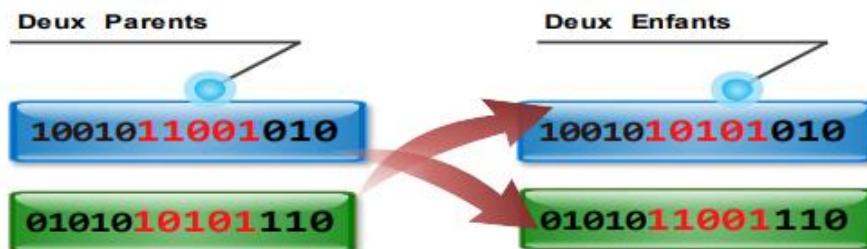


Figure 46: pourcentage RX 24 avec méthode de sélection par tournoi

- **Phase de test du croisement :**

Dans notre algorithme on a choisi le croisement avec un point mais il existe un autre type de croisement appelé **Croisement uniforme**. Ce type de croisement est fondé sur la probabilité. En fait, il permet la génération d'un enfant en échangeant chaque gène des deux parents avec une probabilité égale à 0.5, comme le montre la figure



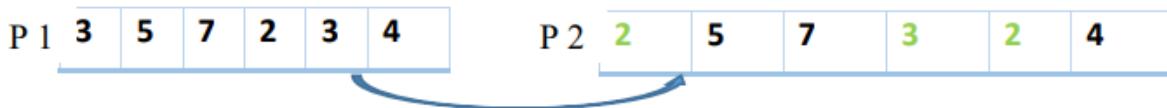
→ après le test de ce type de croisement on a obtenu le même taux de RX24 c'est pour cela on choisit la méthode la plus simple (avec un point)



Figure 47: pourcentage RX 24 avec méthode croisement uniforme

- *Phase de test du mutation :*

Il existe un autre type de mutation intitulé **La mutation par valeur**, il se fait d'une valeur donnée en une autre valeur déterminée, sur tous les gènes du chromosome. Comme le montre la figure



→ Malgré que cette méthode paraît plus efficace mais il a donner un pourcentage plus moins que la méthode de mutation uni point

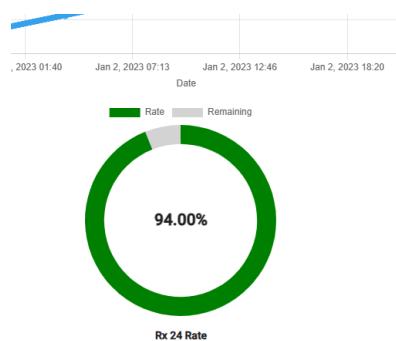


Figure 48: pourcentage RX 24 avec méthode de mutation par valeur

7 Conclusion

Une fois tout le travail accompli, il est essentiel de procéder à une simulation pour valider la solution apportée. Cette simulation permettra de vérifier que tous les efforts déployés sont corrects et qu'ils amélioreront véritablement l'état de l'ordonnancement du processus.

Dans le chapitre suivant, nous aborderons donc cette simulation en détail. Nous utiliserons des outils appropriés pour modéliser le processus de production, en tenant compte des différentes contraintes, des ressources disponibles et des nouvelles stratégies mises en place. Nous simulerons le flux des commandes et évaluerons les performances de notre solution en utilisant des indicateurs clés tels que le temps d'exécution, la productivité et la satisfaction client.

Cette étape de simulation nous permettra de recueillir des données précises sur les résultats attendus de notre solution proposée. Nous pourrons ainsi analyser les performances obtenues, comparer les résultats avec ceux de la situation précédente et déterminer si notre solution répond aux objectifs fixés. En cas de nécessité, nous apporterons les ajustements nécessaires pour parvenir à une solution optimale.

Chapitre 5 : La Simulation du nouveau système d'ordonnancement

1. Contexte général de la simulation

La simulation est largement utilisée dans le monde, la principale raison de son utilisation est la possibilité de la réalisation d'un modèle réaliste, elle est appropriée à l'étude des systèmes complexes et de grande taille, composés de plusieurs éléments en interaction car elle permet de répondre à certains problèmes à chaque fois qu'un modèle mathématique ne peut être trouvé ou que l'expérimentation se révèle impossible ou trop coûteuse. Simulation et modélisation des systèmes de production.

1.1 Définition de la simulation

La simulation est un processus très utile pour bien comprendre le principe de fonctionnement d'un système. C'est une technique, où on peut modéliser des systèmes de production. Elle aide à observer et à comprendre le fonctionnement en temps réel et à analyser plusieurs scénarios du fonctionnement.

La simulation est l'activation du modèle dans le temps, afin de connaître son comportement dynamique et de prédire son comportement futur.

La simulation consiste à :

- Concevoir un modèle du système réel étudié.
- Mener des expérimentations sur ce modèle et non pas des calculs.
- Interpréter les observations fournies par le déroulement du modèle et formuler des décisions relatives au système.

→Le but de la simulation est de comprendre le comportement du système, d'évaluer ses composants et d'optimiser ses performances.

1.2 Objectifs de la simulation

La simulation peut couvrir tous les flux de l'entreprise car elle est capable de représenter : Les flux physiques, les flux informationnels et les flux décisionnels, La simulation aide son utilisateur à :

- Comprendre le fonctionnement du système en apportant des connaissances additionnelles sur son comportement.
- Générer des solutions en fonction des paramètres ou de la structure du modèle étudié.
- Fournir des estimations réalistes (comportement attendu du système, variation à l'intérieur du système).
- Contrôler le système vers un état désiré.

1.3 Les différents logiciels de simulation

Afin de commencer la simulation il faut d'abord choisir le logiciel qui permet de modéliser tout le processus de travail et de générer des interprétations qui permet de résoudre les problèmes rencontrés pour cela il faut faire la comparaison entre ces outils informatiques puissants qui permettent de modéliser et de simuler des systèmes complexes afin de prendre des décisions éclairées, d'optimiser les processus et de prévoir les résultats parmi ces logiciels on peut citer :

- *AnyLogic* :

Il s'agit d'un logiciel de simulation multiméthodes qui permet de modéliser des systèmes dans divers domaines tels que la logistique, la production, les transports, la santé, etc. Il offre une grande flexibilité en permettant d'utiliser plusieurs approches de modélisation, y compris la modélisation basée sur les agents, la dynamique des systèmes et la simulation par événements discrets.



Figure 49: logiciel anylogic

- *Flexim* :

Un logiciel de simulation d'événements discrets. FlexSim comporte une bibliothèque d'objets standards contenant des logiques et des blocs d'activités prédéfinis pour l'optimisation des flux. Ce logiciel de simulation de flux intègre des fonctionnalités en langage C++.



Figure 50: logiciel flexim

- *Arena* :

Est un logiciel de simulation puissant qui permet aux utilisateurs de modéliser, analyser et optimiser des systèmes complexes. Que ce soit pour améliorer la productivité d'une usine, optimiser la chaîne logistique d'une entreprise ou planifier les ressources d'un hôpital, Arena offre une approche efficace pour prendre des décisions éclairées et améliorer les performances opérationnelles.



Figure 51: logiciel arena

1.4 Le choix de logiciel de simulation

- *Pourquoi ARENA ?*

En guise de conclusion nous avons choisi Arena en raison de sa réputation établie, de sa convivialité, de sa capacité de visualisation, de ses fonctionnalités d'analyse avancées et de son soutien. Nous sommes convaincus que ce logiciel nous aidera à atteindre nos objectifs de modélisation, d'analyse et d'optimisation des systèmes de manière efficace et efficiente.

Arena est un environnement facile à utiliser qui offre une bonne visibilité et une bonne compréhension de la modélisation, Arena permet de :

- Reproduire les systèmes réels et analyser le comportement du système modélisé.
- Valider les choix de conception.
- Effectuer des analyses du présent et évaluer les alternatives possibles.
- Identifier les goulots d'étranglements, quantifier les coûts de Process, réduire les temps de cycle.
- Ordonner et allouer les ressources en mode optimum.
- Réaliser d'importantes améliorations des performances (coûts, qualité, service...).

- *Caractéristiques du logiciel Arena*

Arena présente l'avantage de couvrir de larges champs, il possède de nombreuses caractéristiques :

- Plus de 5000 objets d'animation complexes sont inclus dans la bibliothèque d'Arena.
- L'animation peut être également créée par l'utilisateur, Clip Art, des dessins Auto CAD et beaucoup d'autres.
- La compatibilité des données : données d'importation et d'exportation vers un des types de fichiers suivants : Excel, Access, XML, texte,
- Interface simple et intuitive Compatible avec Microsoft Office.
- Modélisation hiérarchique des process.
- Rapports synthétiques et détaillés sur les process, entités, ressources, files d'attentes...
- Simulation et animation 3D exportable.
- Intégration des modèles ARENA avec des systèmes propres à l'entreprise existants.

- *Présentation de l'interface Arena :*

Terminologies utilisées en simulation avec Arena

- **Entité :** Une entité est l'élément isolable composant le flux discret. Il se déplace individuellement à travers le modèle. Exemple : Dans un atelier, les entités sont les pièces à fabriquer.
- **File d'attente ou Stock :** Les stocks sont des éléments qui permettent d'accumuler des entités entre deux machines. Les stocks peuvent être gérés de différentes façons : FIFO, LIFO, avec critère de priorité.
- **Opération ou Machine :** C'est un élément qui traite les entités pendant un temps de cycle qui correspond à un délai fixé dans une unité de temps appropriée. Les articles, stocks et machines constituent l'essentiel des éléments physiques d'un modèle de simulation à événements discrets. Pour faire fonctionner le modèle ou pour prélever des informations, on a aussi besoin d'éléments logiques, qui se décomposent en deux catégories :
- **Les Variables :** Ce sont des valeurs qui caractérisent les éléments du système en entier. On peut y accéder depuis n'importe quel élément physique du modèle. L'analogie avec un langage de programmation structurée est la notion de variable globale.
- **Les Attributs :** Ce sont des variables spécifiques à un article donné et qu'il porte avec lui pendant sa durée de vie dans le modèle. C'est en quelque sorte la carte d'identité d'un article, dans laquelle on pourra mettre par exemple l'heure d'arrivée et l'heure de sortie d'un article, son temps de cycle sur une machine, le nombre de trous à percer sur une autre machine, etc.... Plusieurs articles différents ou identiques peuvent donc porter le même attribut, mais celui-ci pourra avoir des valeurs différentes.

- *Description de l'environnement :*

Dans cette partie, on va expliquer les différents composants de l'interface de logiciel Arena présentée dans la figure ci-dessous :

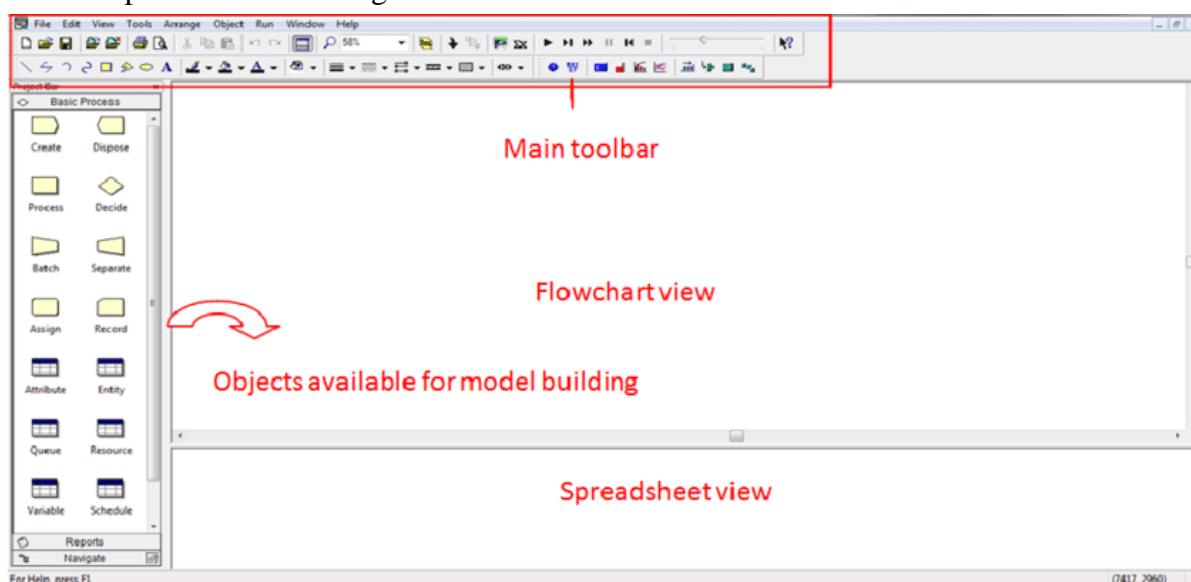


Figure 52: environnement arena

Arena se compose de deux classes : les blocs et les éléments. Les blocs sont des constructions logiques de base qui représentent des opérations ; par exemple, un bloc SEIZE modélise la saisie d'une installation de service par une transaction (désignée dans Arena comme « entité »), tandis qu'un bloc RELEASE libère la fonction pour une utilisation par d'autres transactions. Éléments sont des objets qui représentent des installations, telles que RESSOURCES et QUEUES, ou d'autres composants, tels que DSTATS et TALLIES, utilisés pour la collecte de statistiques. Les composants de modélisation fondamentaux d'Arena, appelés modules, sont sélectionnés à partir de la barre de projet « Project bar », tels que basic process, Advanced processus et Advanced transfer. Chaque bibliothèque contient plusieurs modules qui sont nécessaires à la simulation du processus. Par exemple, un module Process modélise le traitement d'une entité et se compose en interne de blocs tels que : ASSIGN, QUEUE, SAIZE, DELAY et RELEASE. Arena prend également en charge d'autres modules, tels que Statistiques, variables et sorties entre autres.

- *Les modules fondamentaux du simulateur Arena :*

Dans Arena, les modules sont les objets qui définissent le processus à simuler. Toutes les informations nécessaires pour simuler un processus sont stockées dans ses modules.

Tableau 3 :Les modules fondamentaux du simulateur Arena

Create : le début du flux de processus. Les entités entrent ici dans la simulation.	 Create
Dispose : la fin du flux de processus. Les entités sont supprimées de la simulation ici.	 Dispose
Process : une activité, généralement réalisée par une ou plusieurs ressources nécessitant un temps pour être exécuter	 Process
Decide : une branche conditionnelle dans le flux de processus.	 Decide
Batch : collectez un certain nombre d'entités avant qu'elles puissent continuer le traitement.	 Batch
Separate : entités en double pour traitement simultané ou parallèle, ou séparation d'un lot d'entités précédemment établi.	 Separate
Assign : modifie la valeur de certains paramètres (pendant la simulation), tels que le type d'entité ou une variable de modèle.	 Assign
Record : collecte une statistique, comme le nombre d'entités ou le temps de cycle.	 Record

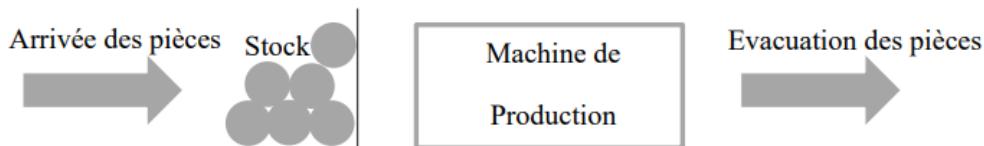


Figure 5 : Système à modéliser

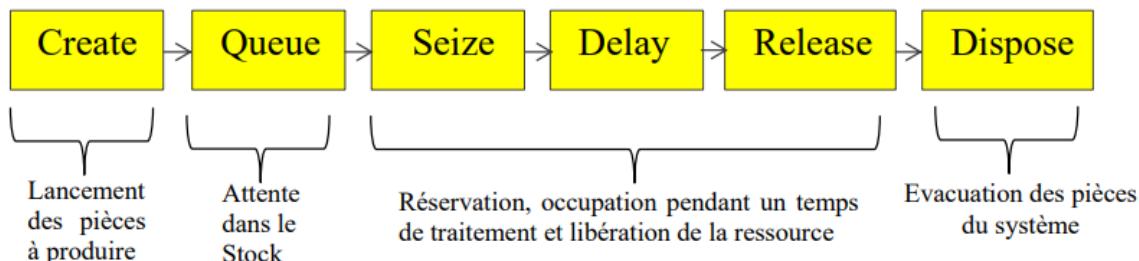


Figure 53: Exemple de modélisation d'un système de production avec Arena

2. Travail préalable à la simulation

2.1 Collecte des données et détermination des paramètres

La modélisation sur les systèmes de simulation nécessite un travail préliminaire qui consiste à la collecte des données et à la génération des lois qui décrivent les processus. Dans cette phase, on va présenter le travail effectué dans ce sens.

- *Le temps inter arrivé :*

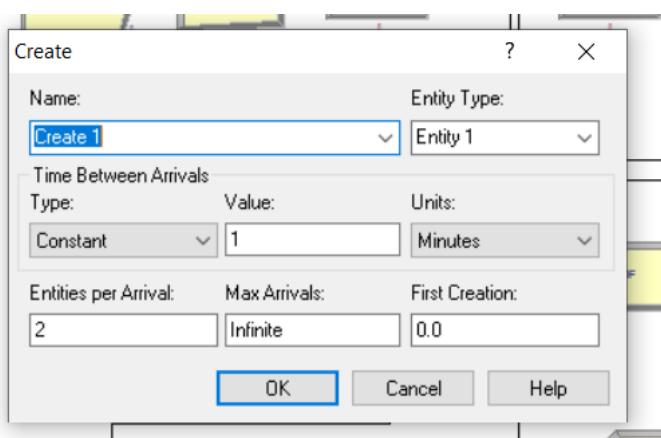


Figure 55: temps inter arrivé magasin fini

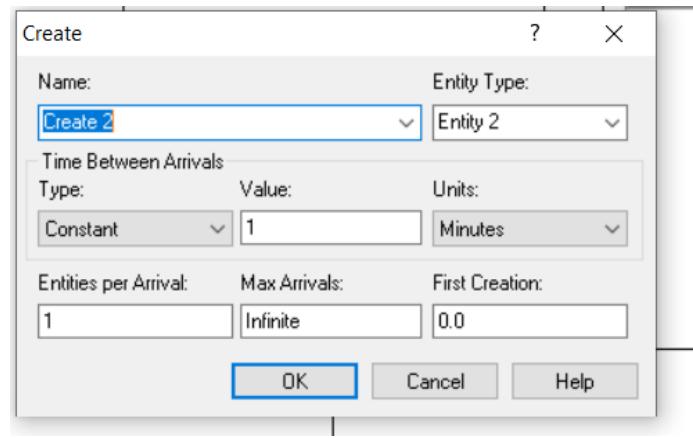


Figure 54: temps inter arrivé magasin semi fini

L'envoie des commandes sera chaque 3.5 heures on va envoyer par des lots de 55 verres pour les OF on va envoyer 4 lots de 55 cela veut dire 2 verres par minute et pour les OT on va envoyer 2 lots de 55 chaque minute cela veut dire 1 verre par minute

- *Détermination des lois qui décrivent les processus :*

Pour chaque processus, on a fait les étapes suivantes :

- Chronométrier le temps nécessaire pour l'achèvement de la tâche.
- Utiliser entre 10 et 100 valeurs à chaque poste pour générer la loi qui décrit le processus à l'aide de l'outil Input Analyser.
- Vérifier la pertinence de la loi à décrire le processus par le test statistique de Kolmogorov.
-

Tableau 4:Détermination des lois qui décrivent les processus

	Processus :	Loi de probabilité (en minutes) :
1	Magasin verre semi fini	WEIB (0.0505, 0.468)
2	Préparation paquet	TRIA (0.04, 0.058, 0.1)
3	Préparation film	Cst (0.08)
4	Glantage	Cst (2)
5	Refroidissement glantage	Cst (10)
6	Ebauchage orbite	1.87+0.45*BETA (1.46, 2.1)
7	Polissage	2.54+1.66*BETA (1.75, 1.61)
8	Gravure	0.36+0.12*BETA (0.743, 1.14)
9	Deglantage	0.09+GAMM (0.0445, 2.84)
10	Bureau contrôle d'ordre	0.14+LOGN (0.254, 0.193)
11	lavage resys	TRIA (0.5, 1, 1.5)
12	Coloration	TRIA (15, 27.6, 57)
13	Magasin fini	WEIB (0.0505, 0.468)
14	Préparation resy	0.28+LOGN (0.231, 0.163)
15	Faste cure	TRIA (40, 45, 50)
16	Contrôle resys	0.05+LOGN (0.0461, 0.0292)
17	Resys	Cst (75)
18	Séchage	CST (10)
19	Anti reflet	CST (45)
20	Contrôle qualité	TRIA (0.5, 1, 1.5)
21	Livraison	WEIB (0.1, 0.5)

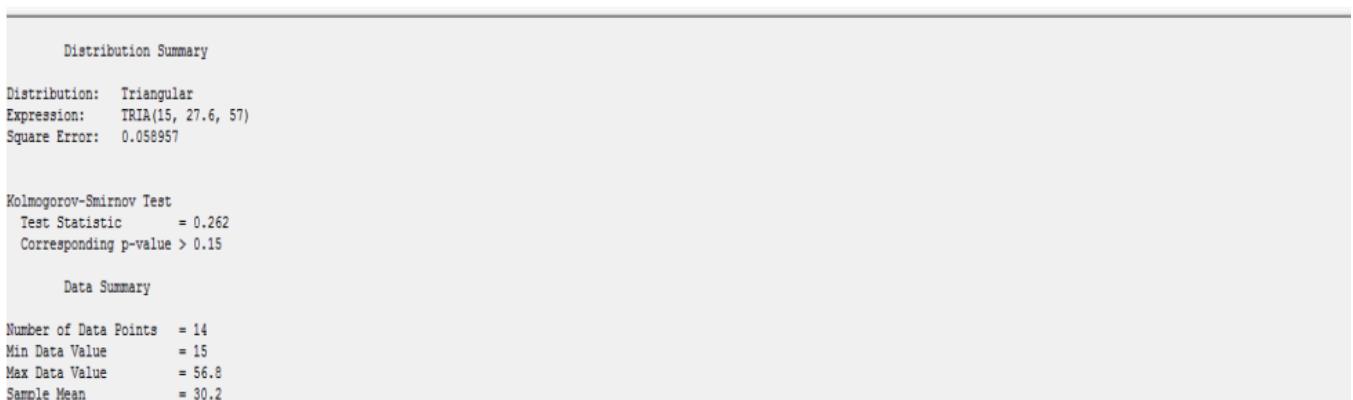
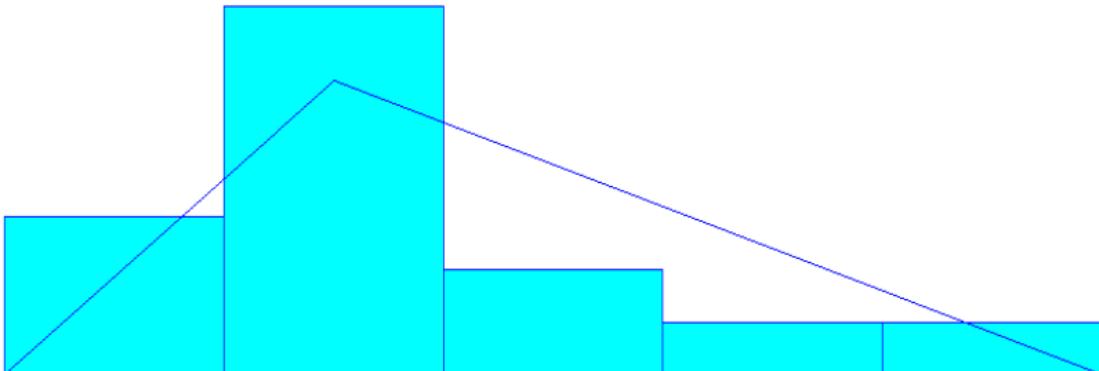


Figure 56: détermination du loi par input Analyzer

- *Les taux du déchet et de rebut :*

Au niveau des postes de contrôle il y a un pourcentage de verres qui revient dans la chaîne pour la retouche, un autre pourcentage est rejeté comme rebut et la quantité restante est envoyée aux postes suivantes. Ces pourcentages ont été déterminés approximativement en se référant au chef de l'atelier de production. L'ensemble est résumé dans le tableau suivant :

Tableau 5: Les taux des déchets et des rebuts

Poste	Verre retouche (%)	Verre rebut (%)
Contrôle surfaçage	2	1
Contrôle coloration	0.5	0.2
Contrôle de traitement Resys	10	0.3
Contrôle de traitement SAR	0.3	0.7

→ Une fois que la phase préliminaire est achevée, on va passer à l'élaboration du modèle sur le logiciel Arena.

3. Modélisation sur Arena

Cette phase sera consacrée à l'explication des étapes suivies pour la modélisation de la ligne de production et celle du traitement du verre dans l'atelier Essilor Sivo dès la présentation des modules utilisés jusqu'à la validation du modèle.

3.1 Construction du modèle

Pour élaborer des modèles de simulation, le concepteur doit construire un réseau basé sur les blocs d'Arena (Create, Decide, Process, Dispose ...) où les entités peuvent circuler. Les blocs utilisés dans un même modèle viennent d'un ou plusieurs bibliothèques [3]. Dans notre modèle de simulation, on a employé deux bibliothèques :

- La Bibliothèque « Basic Process »
- La Bibliothèque « Advanced Process » Dans la partie suivante on va décrire les modules qu'on a utilisés dans notre modèle.

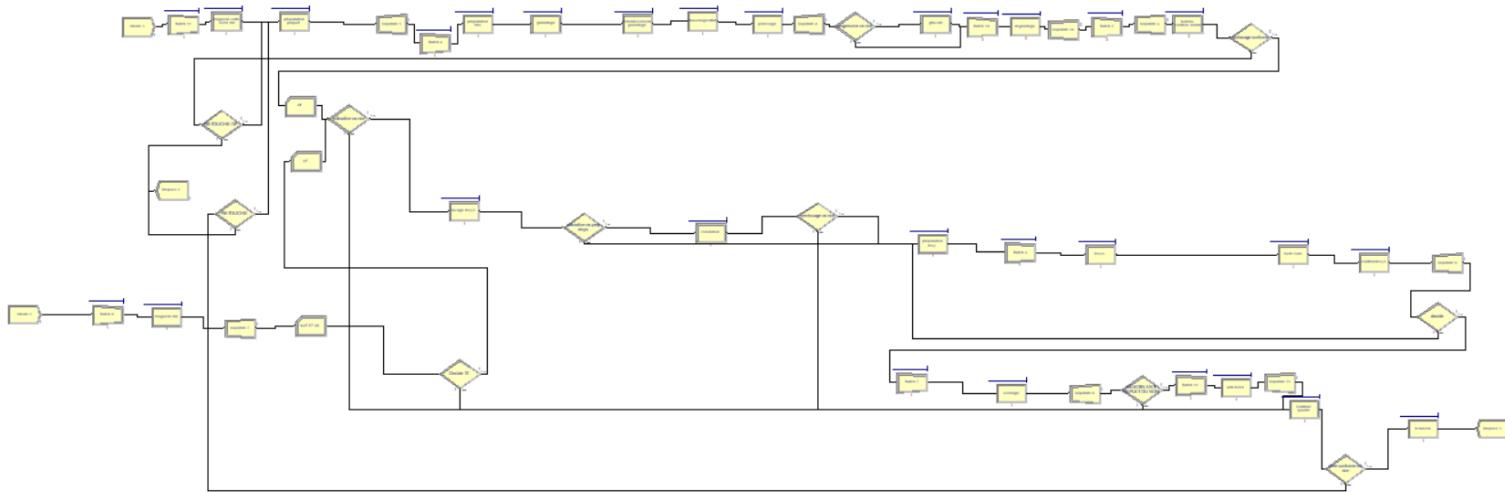


Figure 57: modèle de production société sivo

- *Modélisation des ressources :*

La feuille des données « Ressource » permet de définir les ressources du système y compris la main d'œuvre et les machines. Elle permet également de préciser la disponibilité, les coûts, et les arrêts de ces ressources. Ainsi, nous avons défini toutes les ressources de la partie surfaçage et de traitement comme le montre la figure :

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1 ►	machine galantage1	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
2	machine galantage 2	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
3	machine orbit	Fixed Capacity	2	0.0	0.0	0.0		1 rows	<input checked="" type="checkbox"/>
4	machine polissage 1	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
5	machine polissage 2	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
6	machine polissage 3	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
7	machine gravure 1	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
8	machine gravure 2	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
9	machine deglantage	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
10	operateur controle	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
11	machine coloration 1	Fixed Capacity	4	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
12	machine coloration 2	Fixed Capacity	4	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
13	operateur preparation resys	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
14	machine resys	Fixed Capacity	98	0.0	0.0	0.0		1 rows	<input checked="" type="checkbox"/>
15	machine fast cure	Fixed Capacity	98	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
16	operateur controle resy	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
17	four	Fixed Capacity	Infinite	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
18	machine preparation film	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
19	operateur preparation paquet	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
20	operateur bureau dordre	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
21	operateur lavage resys	Fixed Capacity	28	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
22	operateur controle fini	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
23	bureau dordre	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
24	machine sar 1	Fixed Capacity	55	0.0	0.0	0.0		1 rows	<input checked="" type="checkbox"/>
25	machine sar 2	Fixed Capacity	55	0.0	0.0	0.0		1 rows	<input checked="" type="checkbox"/>
26	machine sar 3	Fixed Capacity	55	0.0	0.0	0.0		1 rows	<input checked="" type="checkbox"/>
27	TAPIS	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
28	REFROIDISSEMENT	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
29	FOUR 2	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Figure 58: liste des ressources

- *Modélisation des processus :*

L'ensemble des processus de transformation doivent être intégrés dans le modèle. Pour simuler ça, on a fait recours au module « Process ».

Ce module issu de la bibliothèque « Basic process » simule le comportement d'une entité en relation avec une ressource. Dans la majorité des modules processus qu'on a utilisés, on a choisi l'action « Seize Delay Release » pour représenter les étapes suivantes :

- L'entité occupe la machine
- La machine reste à la disposition de l'entité tout au long du temps de l'exécution de l'opération donnée
- L'entité libère la ressource Ainsi en utilisant ce module, nous avons défini les processus illustrés dans la figure suivante

Process - Basic Process														
	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Expression	Report Statistics	
1	magasin verre semi fini	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	15	1.5	WEIB(0.0505, 0.468)	✓	
2	preparation paquet	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	1	1.5	TRIA(0.04, 0.058 , 0.1)	✓	
3	preparation film	Standard	Seize Delay Release	Medium(2)	1 rows	Constant	Minutes	Value Added	.5	0.08	1.5	cst .08	✓	
4	galantage	Standard	Seize Delay Release	Medium(2)	2 rows	Constant	Seconds	Value Added	.5	51	1.5	116	✓	
5	refroidissement galantage	Standard	Seize Delay Release	Medium(2)	1 rows	Constant	Minutes	Value Added	.5	1	1.5	cst (10)	✓	
6	ebaughageorbit	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	1	1.5	1.87+0.45*BETA(1.46, 2.1)	✓	
7	polissage	Standard	Seize Delay Release	Medium(2)	3 rows	Constant	Seconds	Value Added	.5	57	1.5	2.45+1.66*BETA(1.75 , 1.61)	✓	
8	gravure	Standard	Seize Delay Release	Medium(2)	2 rows	Expression	Minutes	Value Added	.5	1	1.5	0.36+0.12*BETA(0.743 , 1.14)	✓	
9	deglantage	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	1	1.5	0.09+GAMM(0.0445 , 2.84)	✓	
10	bureau controle dordre	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	1	1.5	0.14+LOGN(0.254 , 0.193)	✓	
11	lavage resys	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Minutes	Value Added	.5	1	1.5	1	✓	
12	coloration	Standard	Seize Delay Release	Medium(2)	2 rows	Expression	Minutes	Value Added	.5	1	1.5	TRIA(15 , 27.6 , 57)	✓	
13	magasin fini	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	1	1.5	WEIB(0.0505, 0.468)	✓	
14	preparation resy	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	1	1.5	0.28+LOGN(0.231 , 0.163)	✓	
15	faste cure	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Minutes	Value Added	40	45	50	1	✓	
16	controleresys	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	1	1.5	0.05+LOGN(0.0461 , 0.0292)	✓	
17	resys	Standard	Seize Delay Release	Medium(2)	1 rows	Constant	Minutes	Value Added	.5	75	1.5	1	✓	
18	sechage	Standard	Seize Delay Release	Medium(2)	2 rows	Constant	Minutes	Value Added	.5	10	1.5	1	✓	
19	anti reflet	Standard	Seize Delay Release	Medium(2)	3 rows	Constant	Minutes	Value Added	.5	45	1.5	EXPO()	✓	
20	controle qualite	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Minutes	Value Added	.5	1	1.5	1	✓	
21	livraison	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	.5	1	1.5	WEIB(0.1, 0.5)	✓	

Figure 59: liste des processus

3.2 Présentation du modèle

Le modèle élaboré sur Arena comporte les processus de surface, et de traitement :

- *Surfaçage :*

Cette partie comporte l'ensemble des processus de préparation du paquet et du film, le glantage, l'ébauchage, le polissage, la gravure et le déglantage. Mise à part ces processus, on a remarqué que les opérateurs déplacent les verres entre les postes par lot. Pour cela on a introduit des modules de « batch » et « separate » pour modéliser ce comportement. Ensuite à partir de contrôle surface, un pourcentage des verres est envoyé au contrôle fini et un autre pourcentage vers le bureau d'ordre. De plus on compte un pourcentage pour le rebut et pour la retouche

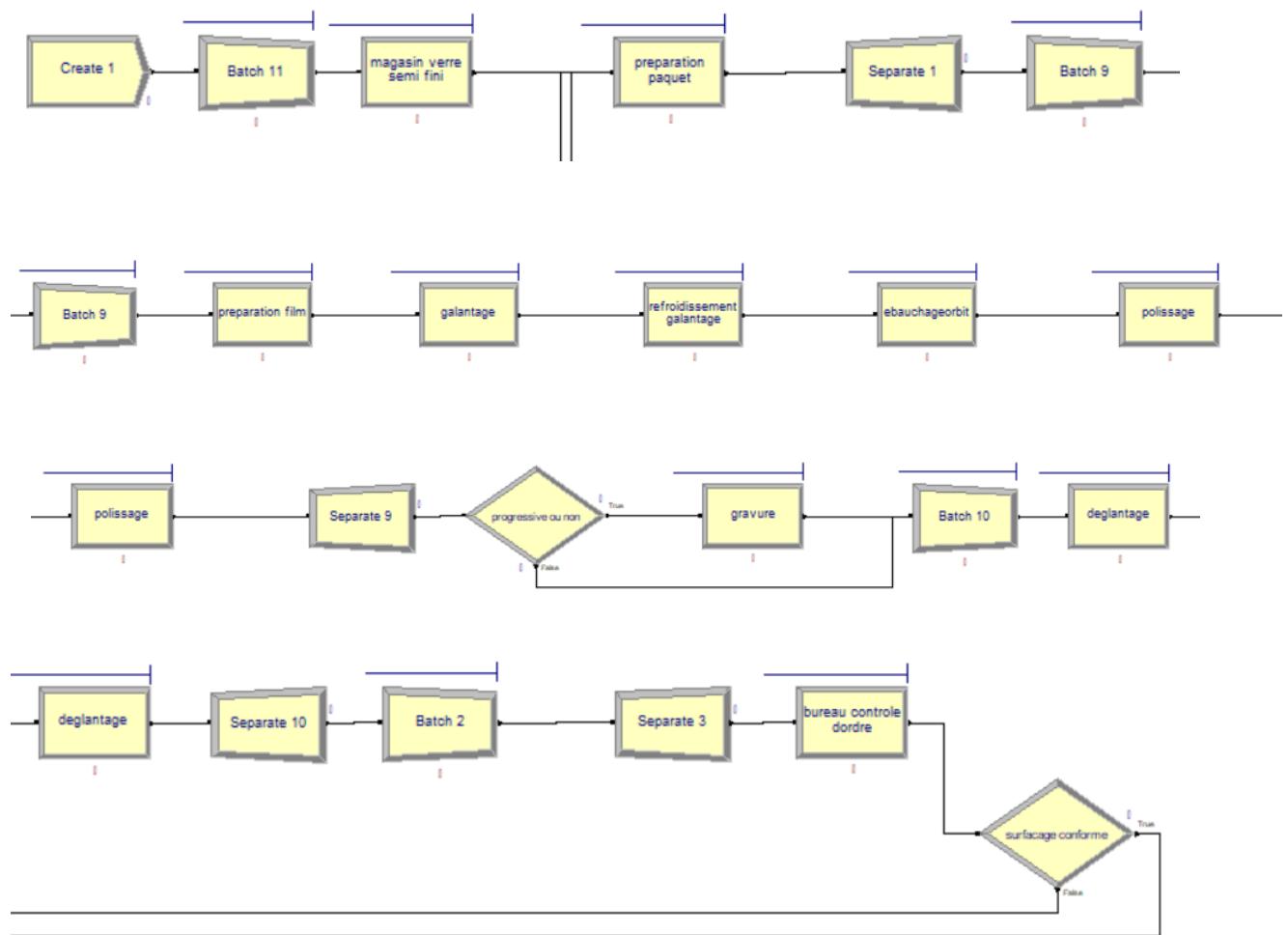


Figure 60: modèle d'atelier de surface

- *Retouche et rebut :*

En prenant en compte les taux de retouche et de rebut mentionnés précédemment, et j'ai utilisé le module "Decide" pour cette phase. Mon objectif était de comprendre et d'optimiser le processus de production en évaluant les impacts des taux de retouche et de rebut sur le rendement global. Grâce au module "Decide" d'Arena, j'ai pu prendre des décisions basées sur des règles spécifiques en fonction des taux de retouche et de rebut.

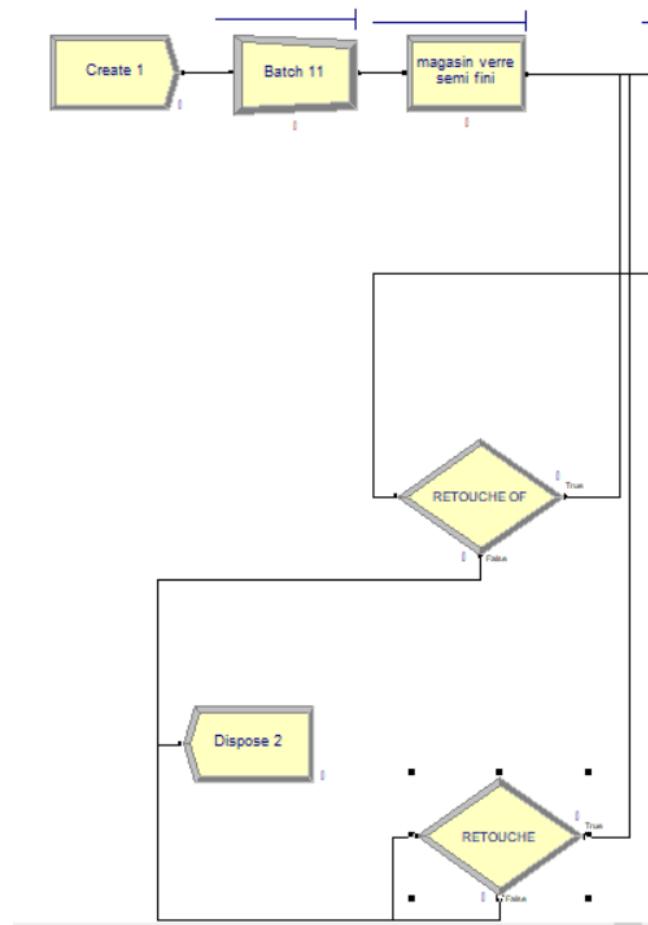


Figure 61: modèle des taux de retouche et rebut

- *Coloration :*

En enregistrant le flux de production avec le module "Record", j'ai pu recueillir des données précises sur le nombre de verres de types "OF" et "OT" qui arrivaient à l'atelier de coloration. Cela m'a permis d'évaluer l'efficacité de la chaîne de production et d'identifier d'éventuels problèmes ou goulets d'étranglement.

En utilisant le module "Decide", j'ai pu définir des règles et des critères pour décider si un verre devait subir une étape de coloration ou non. J'ai également pu déterminer quels verres nécessitaient un lavage "resys" avant l'étape de coloration. Cette fonctionnalité m'a aidé à automatiser le processus de décision, en m'appuyant sur des paramètres prédéfinis pour optimiser le flux de production.

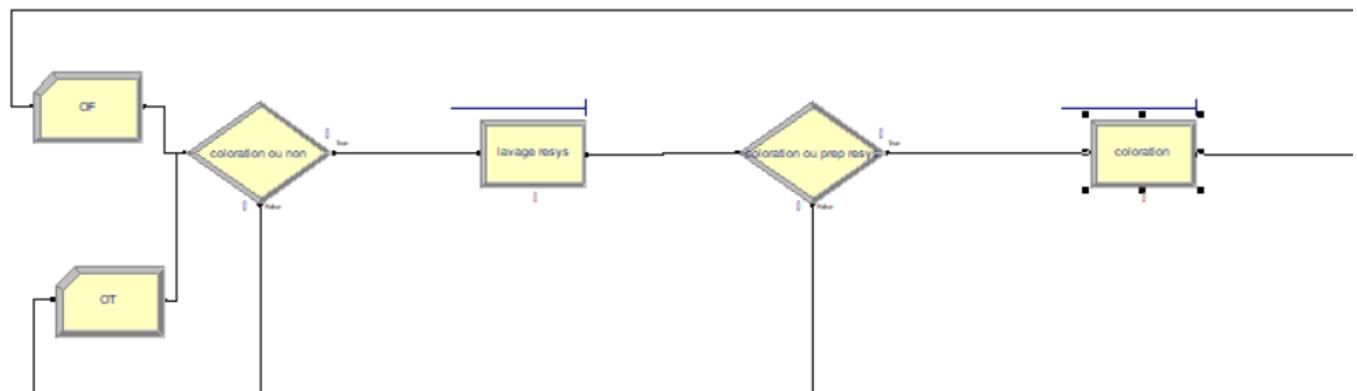


Figure 62: modèle d'atelier de coloration

- *Resys :*

Le traitement resys commence par un processus de préparation de verres. Ensuite un lot de 14 verres est déplacé vers l'atelier de resys. Pour cela on a employé les modules « Batch » et « Separate ». Le contrôle de la qualité de traitement est modélisé par un bloc « Decide ».

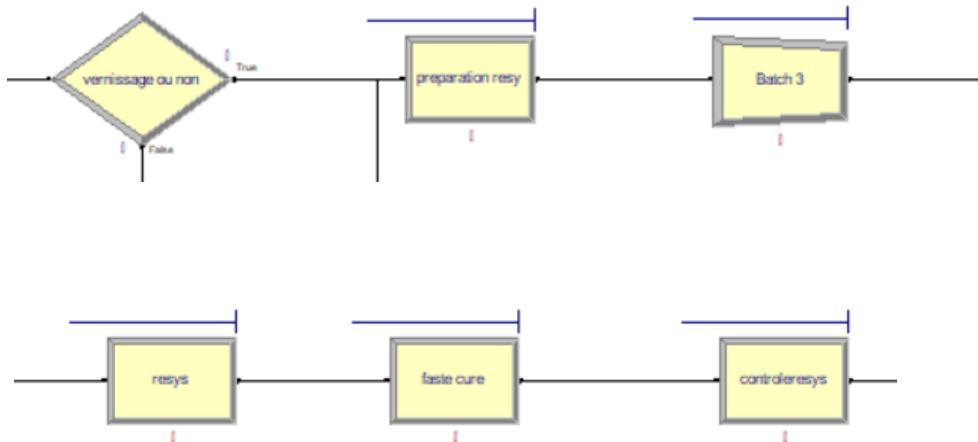


Figure 63: modèle d'atelier de resys

- *Anti reflet et contrôle qualité :*

Le traitement Super anti reflet commence par le séchage des verres mis dans un plateau dans le four. Ensuite, selon le type d'antireflet qu'on a déjà envoyé auparavant par des lots le processus de traitement commence

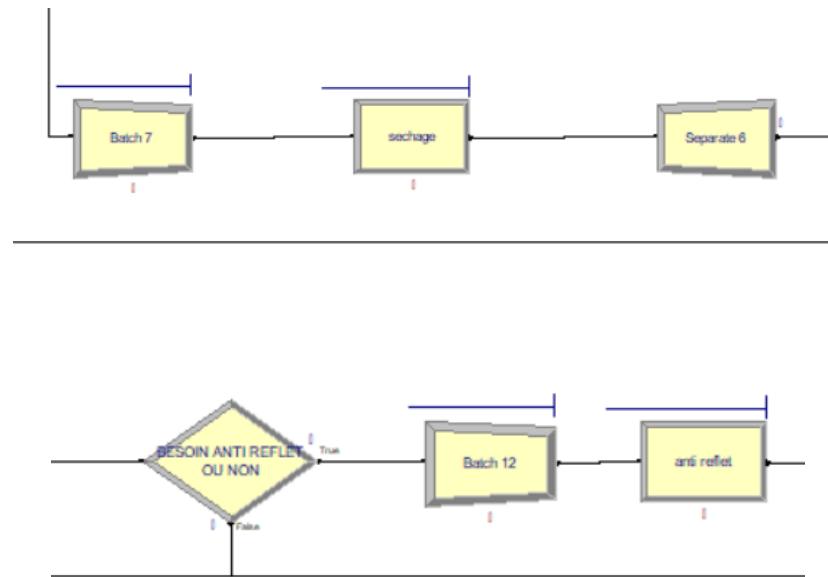


Figure 65: modèle d'atelier d'anti reflet

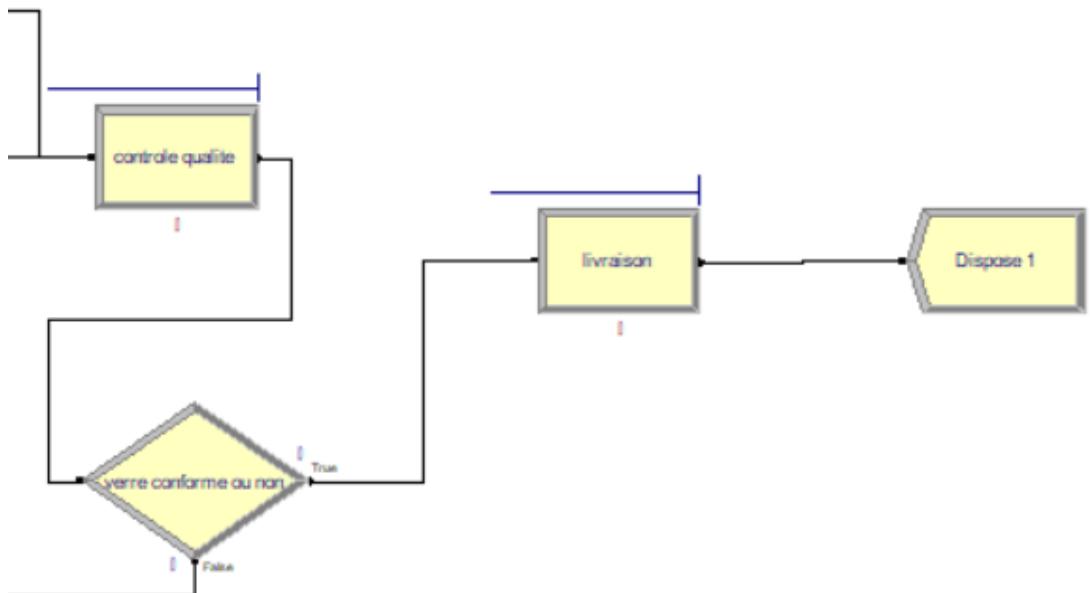


Figure 64: modèle e bureau de contrôle qualité

4. Validation et vérification du modèle

Après avoir établi le modèle de simulation, il faut passer à la phase de validation qui est nécessaire pour confirmer la fiabilité de notre modèle. Pour cela, on va suivre la démarche suivante :

- Déterminer le nombre de réplications nécessaires pour ajouter plus de précision aux résultats.
- Déterminer la période de warm-up de système
- Interpréter les résultats obtenus dans le rapport de la simulation

4.1 Détermination du nombre des réplications

Vu que les données qu'on a intégrées dans le modèle sont de nature probabiliste, on ne peut pas se baser sur les résultats donnés par une seule réplication. Ainsi il faut déterminer le nombre des réplications pour pouvoir proposer des bonnes recommandations aux décideurs

Réplications	Valeurs trouvées
1	8500
2	8521
3	8475
4	8611
5	8652
Moyenne : \bar{x}	8551,5
Ecart type : S	75,95
Moyenne de la population : μ	8457,21
T statique	2,776
Taille n	5

Avec :

- $n = \left(\frac{s * T}{\bar{x} - \mu}\right)^2$
- $Variance (s^2) = \frac{\text{Somme des carrés des écarts}}{n-1}$
- $S = \sqrt{Variance (s^2)}$

Calcule :

- Carré des Ecart =(8500-8551,5)²=2652,25
=(8521-8551,5)²=930,25
=(8475-8551,5)²=5852,25
=(8611-8551,5)²=3540,25
=(8652-8551,5)²=10100,25

- Variance (s^2)= $\frac{2652,25+930,25+5852,25+3540,25+10100,25}{5-1} = 5768,812$

- $S=\sqrt{5768,812} = 75,95$

- μ représente la moyenne théorique ou la moyenne de la population. Il s'agit de la moyenne attendue ou théorique de l'ensemble de la population que vous étudiez. μ est généralement inconnu et doit être estimé à partir d'un échantillon représentatif est pour cela on a utilisé des étude d'autre rapport dans le même domaine afin de fixer $\mu = 8457,21$
- T :niveau de confiance est 95% ($\alpha = 0,05$) et on a 4 degré de liberté($n-1$) alors d'après le table de loi de student $T=2,776$
- $n=(\frac{s*T}{x-\mu})^2 = (\frac{75,95*2,776}{8551,5-8457,21})^2=5$

4.2 Détermination de la période de warm-up

Le début de la simulation représente une phase transitoire où les ressources n'ont pas encore commencé à traiter les entités et où la congestion au niveau des ressources est pratiquement nulle. Cette phase est appelée le « warm-up période ». Le fait d'inclure cette période dans le calcul des indicateurs liés aux performances du système va erroné les résultats. Comme il n'existe pas des méthodes analytiques qui déterminent cette période d'une façon exacte, on va utiliser l'outil « output Analyzer » d'Arena pour évaluer graphiquement la période transitoire sur des courbes de rendement du système en fonction du temps pour 5 réplications

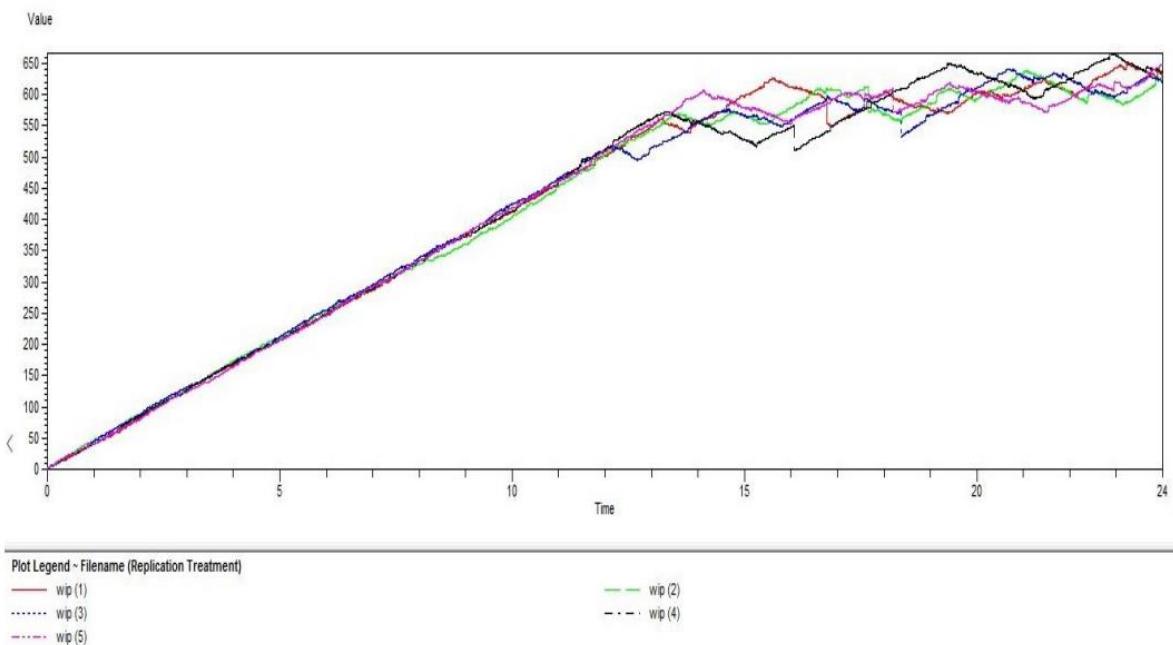


Figure 66: détermination de période de warm up

➔ On remarque sur les courbes que la quantité d'encours reste instable pendant une période transitoire puis il commence à se stabiliser à partir de 16 heures à peu près. Ainsi, on fixe la période de warm-up à 16 heures pour la suite.

4.3 Validation du modèle

Replications:	5	Time Units:	Hours <th data-cs="3" data-kind="parent">Key Performance Indicators</th> <th data-kind="ghost"></th> <th data-kind="ghost"></th>	Key Performance Indicators		
System	Average					
Number Out	8,652					
Time						
Waiting Time	Average	Half Width	Minimum Value	Maximum Value		
anti reflet.Queue	0.00	(Insufficient)	0.00	0.00		
Batch 10.Queue	0.00253411	0,000037500	0.00	0.00799984		
Batch 11.Queue	0.4500	(Correlated)	0.00	0.9000		
Batch 12.Queue	1.3808	0,329086110	0.00	8.8105		
Batch 2.Queue	0.2819	0,005424120	0.00	0.7639		
Batch 3.Queue	2.0064	0,108313359	0.00	4.8112		
Batch 7.Queue	2.4750	0,354811909	0.00	8.8105		
Batch 8.Queue	0.4500	(Correlated)	0.00	0.9000		
Batch 9.Queue	0.00508816	0,000155759	0.00	0.2947		
bureau controle dordre.Queue	0.1769	0,002011703	0.00	0.4428		
coloration.Queue	67.3892	(Correlated)	0.00	135.81		
controle qualite.Queue	29.3841	(Correlated)	0.00	60.4557		
controleresys.Queue	0.00	(Insufficient)	0.00	0.00		
deglantage.Queue	0.00	0,000000000	0.00	0.00		
ebauchageorbit.Queue	0.00780910	0,000392253	0.00	0.03286232		
faste cure.Queue	0.00	(Insufficient)	0.00	0.00		
galantage.Queue	0.05858764	0,001282104	0.00	0.1797		
gravure.Queue	0.00165990	(Correlated)	0.00	0.00799618		
lavage resys.Queue	0.00000566	0,000005696	0.00	0.00994780		
livraison.Queue	0.00023503	0,000100324	0.00	0.1129		
magasin fini.Queue	0.00147704	(Insufficient)	0.00	0.02952796		
magasin verre semi fini.Queue	0.00134308	(Insufficient)	0.00	0.02865578		
polissage.Queue	0.00168014	0,000137738	0.00	0.01582361		
preparation film.Queue	0.00534897	0,000102155	0.00	0.01733333		
preparation paquet.Queue	0.00001140	0,000002591	0.00	0.00160014		
preparation resy.Queue	0.01174465	0,001174379	0.00	0.1439		
refroidissement galantage.Queue	0.1301	0,002351167	0.00	0.3450		
resys.Queue	0.00	(Insufficient)	0.00	0.00		
sechage.Queue	0.00	(Insufficient)	0.00	0.00		

Figure 67:temps d'attente des process

Other

Number Waiting	Average	Half Width	Minimum Value	Maximum Value
anti reflet.Queue	0.00	(Insufficient)	0.00	0.00
Batch 10.Queue	0.2445	0,006784074	0.00	2.0000
Batch 11.Queue	27.0000	0,177660226	0.00	55.0000
Batch 12.Queue	20.3908	(Insufficient)	0.00	55.0000
Batch 2.Queue	27.1799	0,566618686	0.00	55.0000
Batch 3.Queue	48.1857	1,53066	0.00	98.0000
Batch 7.Queue	52.6290	(Insufficient)	0.00	110.00
Batch 8.Queue	27.0000	0,177660226	0.00	55.0000
Batch 9.Queue	0.4924	0,013710781	0.00	2.0000
bureau controle dordre.Queue	17.0494	0,411945120	0.00	54.0000
coloration.Queue	2665.01	(Correlated)	0.00	5354.00
controle qualite.Queue	2799.80	(Correlated)	0.00	5855.00
controleresys.Queue	0.00	(Insufficient)	0.00	0.00
deglantage.Queue	0.00	(Insufficient)	0.00	0.00
ebauchageorbit.Queue	0.3769	0,017766220	0.00	2.0000
faste cure.Queue	0.00	(Insufficient)	0.00	0.00
galantage.Queue	2.8349	0,076952414	0.00	26.0000
gravure.Queue	0.07896294	0,003387441	0.00	1.0000
lavage resys.Queue	0.00031375	(Insufficient)	0.00	2.0000
livraison.Queue	0.01040980	0,002391740	0.00	6.0000
magasin fini.Queue	0.00161132	(Insufficient)	0.00	1.0000
magasin verre semi fini.Queue	0.00146517	(Insufficient)	0.00	1.0000
polissage.Queue	0.08107759	0,007204148	0.00	1.0000
preparation film.Queue	0.2588	0,006605059	0.00	26.0000
preparation paquet.Queue	0.00043182	(Insufficient)	0.00	1.0000
preparation resy.Queue	0.2838	0,038025096	0.00	18.0000
refroidissement	6.2905	0,176305863	0.00	21.0000
galantage.Queue	0.00	(Insufficient)	0.00	0.00
resys.Queue	0.00	(Insufficient)	0.00	0.00
sechage.Queue	0.00	(Insufficient)	0.00	0.00

Figure 68:nombres des verres en attente

➔ Au cours de notre étude, nous avons constaté que le système d'ordonnancement basé sur la métaheuristique génétique a réussi à réduire de manière significative le waiting time global du système. Cette amélioration a été observée dans l'ensemble des processus impliqués, permettant ainsi une meilleure gestion du temps et une utilisation plus optimale des ressources disponibles.

Un résultat particulièrement remarquable est la réduction à zéro du waiting time à la poste goulot, qui était auparavant un point critique dans notre système de production. Cela signifie que notre approche a réussi à éliminer complètement le temps d'attente associé à cette tâche, ce qui a eu un impact considérable sur l'efficacité globale du processus.

Les résultats de cette étude suggèrent que la métaheuristique génétique est une approche prometteuse pour l'optimisation de l'ordonnancement dans des environnements complexes. En permettant une réduction significative du waiting time, notre système d'ordonnancement peut contribuer à améliorer la productivité, réduire les coûts opérationnels et augmenter la satisfaction des clients grâce à des délais de livraison plus courts.

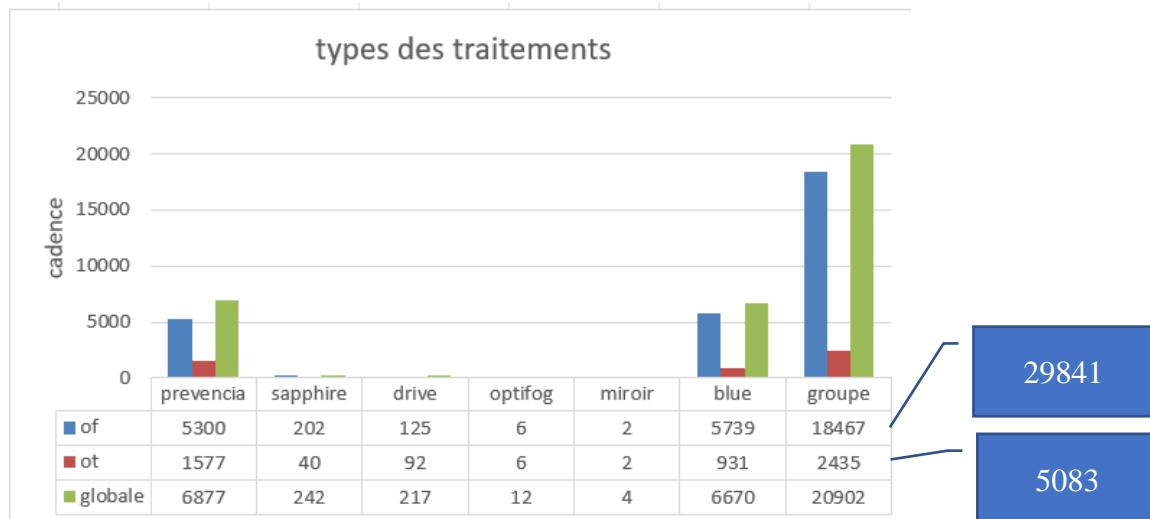


Figure 69:nombres des of et ot pour l'ancien système d'ordonnancement

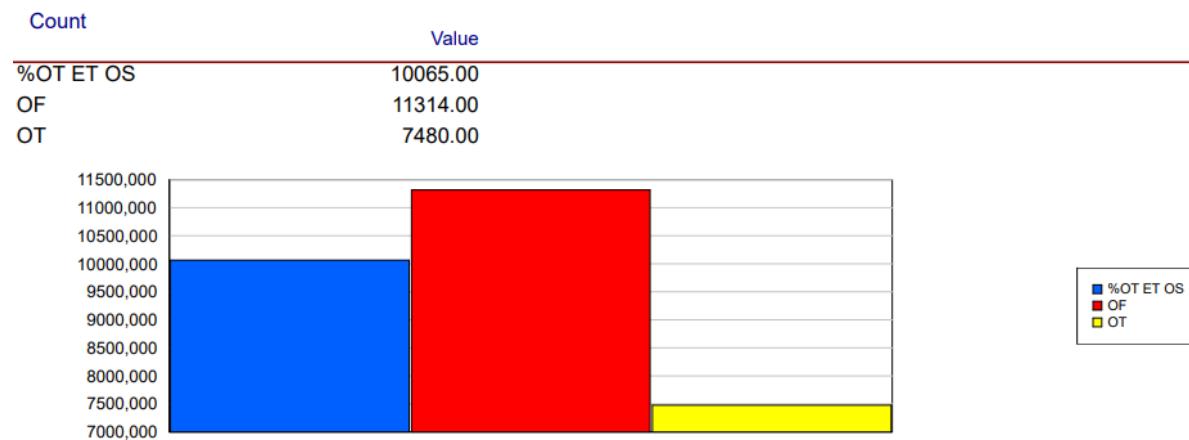


Figure 70:nombres des of et ot pour nouveau système

➔En effectuant une simple comparaison avant et après l'implémentation de notre système d'ordonnancement basé sur la métaheuristique génétique, nous avons pu constater une augmentation significative du nombre de productions des ordres de fabrication (OF) et des ordres de travail (OT). Cette augmentation est indéniablement remarquable et témoigne de l'impact positif de notre approche sur l'efficacité opérationnelle de notre environnement de production.

Avant l'optimisation de l'ordonnancement, le nombre de productions des OF et des OT était souvent limité par des contraintes temporelles et des goulets d'étranglement, entraînant des retards et une utilisation inefficace des ressources. Cependant, avec notre système basé sur la métaheuristique génétique, les OF et les OT ont pu être planifiés de manière plus intelligente et équilibrée, éliminant ainsi les temps d'attente superflus.

5. Conclusion

En conclusion, la comparaison entre les performances avant et après l'application de notre système d'ordonnancement révèle une nette amélioration de l'efficacité opérationnelle, illustrée par une augmentation substantielle du nombre de productions des OF et des OT. Ces résultats confirment l'efficacité de notre approche basée sur la métaheuristique génétique pour l'optimisation de l'ordonnancement dans un environnement de production complexe. Ils soulignent également l'importance d'une planification optimale dans l'amélioration des performances globales du système et ouvrent de nouvelles perspectives pour des améliorations continues et futures.

Une fois qu'on a terminé la phase de modélisation de la chaîne de production et la validation du modèle, on doit passer à la partie informatique

Chapitre 6 : Spécification des besoins et analyse

1. Introduction

Dans ce chapitre, nous présentons tout d'abord, la première phase du cycle de développement du logiciel qui consiste à définir les besoins fonctionnels et non fonctionnels de notre application. Par la suite, nous présentons les technologies utilisées pour la réalisation de notre projet. Nous finissons par la présentation de l'architecture globale de notre application.

2. Identification des acteurs

L'identification des acteurs est la première phase dans l'analyse d'une application. Les acteurs sont des entités externes qui interagissent avec le système pour réaliser des fonctions spécifiques.

Dans cette étape, nous analysons la façon dont notre application interagit avec son environnement externe. Les acteurs qui peuvent interagir avec le système sont entre autres :

- Le chef production ou un planificateur : chargé d'organiser, de consulter ou de lancer des planifications.
- Un agent de production : chargé d'exécuter les tâches de chaque ordre issu de la planification, ainsi que de veiller au suivi de celles-ci.
- Un agent du service client : chargé de gérer les ordres de fabrications et de traitements.
- Un chargé des stocks : responsable de la gestion des stocks.
- Le système ERP : permet un échange de données avec le système de l'application.

3. Spécification des besoins

Il est primordial que l'application réponde aux besoins des clients. Nous citerons dans cette section, les différents besoins fonctionnels et non fonctionnels pour cette application.

3.1. Identification des Besoins fonctionnels

Les besoins fonctionnels d'une application décrivent les fonctionnalités et les capacités spécifiques que l'application doit posséder pour satisfaire les besoins de ses utilisateurs et atteindre les objectifs de l'entreprise. Ces besoins définissent ce que l'application doit pouvoir faire d'un point de vue opérationnel et technique. Ils sont généralement exprimés en termes d'actions, de tâches, de traitements de données et d'interactions avec les utilisateurs. Les principales fonctionnalités que nous avons pu identifier pour cette application se résument comme suit :

- Authentification et gestion des utilisateurs et des permissions : L'application doit permettre aux utilisateurs de s'identifier de manière sécurisée (connexion) de gérer leurs informations de compte (mot de passe, profil, etc.) et de gérer les accès aux fonctionnalités permises à chacun.
- Planification et ordonnancement optimisé des commandes sur plusieurs contraintes.
- Visualisation et historisation des plannings.
- Gestion des données : L'application doit être capable de stocker, récupérer, mettre à jour et supprimer des données spécifiques tels que les informations de clients, de ressources en stocks, de commandes ou de phases de la chaîne de production.

3.2. Identification des besoins non fonctionnels

Les besoins non fonctionnels, également appelés contraintes ou exigences non fonctionnelles, définissent les critères de qualité, les performances et les caractéristiques globales de l'application. Ils se concentrent sur la manière dont l'application doit être, c'est-à-dire les attributs qui définissent sa qualité, sa sécurité, sa fiabilité, sa performance et son comportement global. Ceux-ci doivent être pris en compte, car il nous guide sur la conception de l'architecture du système. Les principales exigences non fonctionnelles auxquelles notre application doit répondre sont :

- Performance : Temps de réponse de l'application, temps d'ouverture et de rafraîchissement de l'écran, etc.
- Sécurité : prévenir les accès non autorisés et garantir la confidentialité des informations.
- Ergonomie : L'application doit être intuitive et facile à utiliser pour les utilisateurs finaux.
- Responsivité et multiplateforme : compatible à toute les plateformes et à toutes les tailles d'écrans.
- Extensibilité : L'application doit être capable de s'adapter à de nouvelles fonctionnalités, de nouvelles technologies ou à une augmentation de la charge de travail.
- Facilité de maintenance : l'application doit être facile à maintenir. Le code doit être lisible, bien structuré et doit suivre les règles de bonne pratique.

4. Choix technologique

Pendant la phase de réalisation de notre projet, nous avons utilisé divers outils logiciels et une variété de langages de développement.

4.1. Environnement logiciel

- Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source léger et puissant développé par Microsoft. Il offre une interface utilisateur moderne et une gamme de fonctionnalités avancées pour faciliter le développement de logiciels dans de nombreux langages de programmation. Doté d'une vaste bibliothèque d'extensions, VS Code permet aux développeurs de personnaliser leur environnement de travail et d'intégrer des outils supplémentaires. Grâce à ses capacités d'auto complétion, de débogage en temps réel et de gestion de contrôle de version, il offre une expérience de développement productive et efficace pour les projets de toutes tailles.

Nous utilisons ce logiciel pour écrire le code source du front-end.

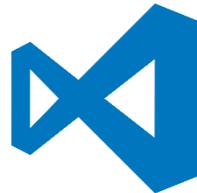


Figure 71:Logo Visual Studio Code

- Spring Tool Suite :

Spring Tool Suite (STS) est un environnement de développement intégré (IDE) basé sur Eclipse, spécialement conçu pour faciliter le développement d'applications Java basées sur le Framework Spring. STS offre un ensemble d'outils puissants pour accélérer le développement, la configuration et le débogage d'applications Spring. Avec une interface utilisateur conviviale et une large gamme d'extensions, STS permet aux développeurs de gagner en productivité tout en garantissant une meilleure qualité de code pour les projets basés sur Spring.

Nous avons utilisé STS pour le développement du back-end de notre application.



Figure 72:Logo Spring Tools Suite

- Xampp

XAMPP est un environnement de développement web qui regroupe plusieurs outils essentiels pour créer et tester des applications web localement. Le nom "XAMPP" est un acronyme qui représente les logiciels inclus dans le package : X (cross-Plateforme), A (Apache Server), M (MySQL), P (PHP language), P (Perl language).



Figure 73:Logo Xampp

- MySQL

MySQL est un système de gestion de base de données relationnelle open source très populaire. Il est utilisé pour stocker, gérer et organiser les données dans de nombreuses applications et sites web. MySQL utilise le langage SQL (Structured Query Language) pour interagir avec la base de données.



Figure 74:Logo MySQL

- Postman

Postman est une plateforme de collaboration pour le développement d'API (Application Programming Interface). C'est un outil puissant qui permet aux développeurs de tester, de déboguer et de documenter les API de manière efficace.



Figure 75:Logo Postman

- Draw.io

Draw.io est une application en ligne de création de diagrammes et de schémas. Elle permet aux utilisateurs de concevoir facilement des organigrammes, des diagrammes de flux, des cartes mentales, des diagrammes UML, des diagrammes de réseau, des diagrammes de processus, et bien d'autres encore. C'est un outil polyvalent et puissant, apprécié aussi bien par les professionnels pour la création de schémas techniques que par les étudiants pour l'organisation d'idées et de concepts. Il a été notamment utilisé pour réaliser l'ensemble des conceptions UML relatifs à ce projet.



Figure 76:Logo Draw.io

4.2. Environnement de développement

- Typescript

TypeScript est un langage de programmation développé par Microsoft, qui étend et améliore le langage JavaScript. En ajoutant des fonctionnalités de typage statique, TypeScript permet aux développeurs de définir les types de données pour les variables, les fonctions et les objets, rendant ainsi le code plus robuste et moins sujet aux erreurs. Il prend également en charge les concepts de programmation orientée objet tels que les classes et les interfaces, facilitant ainsi la structuration et la réutilisation du code. TypeScript est ensuite transcompilé en JavaScript standard, ce qui garantit une large compatibilité avec les navigateurs et les environnements d'exécution.



Figure 77:Logo TypeScript

- **HTML 5**

HTML5 est la dernière version du langage de balisage utilisé pour structurer le contenu des pages web. Il offre des améliorations significatives telles que des balises sémantiques pour une meilleure structuration du contenu, une intégration native des éléments et une compatibilité multiplateforme pour une expérience utilisateur cohérente sur différents appareils et navigateurs. HTML5 est devenu la norme actuelle du développement web, permettant aux développeurs de créer des sites web modernes, interactifs et accessibles aux besoins des utilisateurs.



Figure 78:Logo Html 5

- **CSS**

CSS3, ou Cascading Style Sheets level 3, est la dernière version du langage de feuille de style utilisé pour contrôler l'apparence et la mise en page des pages web. Enrichissant ses prédecesseurs, CSS3 offre des fonctionnalités avancées telles que des sélecteurs améliorés, des transitions, des animations, des ombres, des bordures arrondies, des dégradés et des médias queries pour le rendu responsive. Ces nouvelles fonctionnalités permettent aux développeurs web de créer des designs plus esthétiques, interactifs et adaptatifs, tout en améliorant l'expérience utilisateur sur une variété d'appareils et de navigateurs.



Figure 79:Logo Css 3

- Spring Security

Spring Security est un Framework de sécurité robuste et complet conçu pour les applications Java basées sur Spring. Il offre des fonctionnalités de gestion d'authentification, d'autorisation, de protection contre les attaques courantes (comme les attaques par injection SQL et XSS), ainsi que des mécanismes de gestion des sessions et des rôles. Spring Security facilite l'intégration transparente de la sécurité dans une application web ou une API, en permettant aux développeurs de configurer facilement les stratégies de sécurité et d'appliquer des contrôles d'accès à différentes parties de l'application.



Figure 80:Logo Spring Security

- Spring Cloud : OpenFeign

Spring Cloud OpenFeign est une bibliothèque Java qui simplifie la communication entre les services dans une architecture microservices basée sur Spring Cloud. Elle facilite l'appel de services REST en utilisant une approche déclarative plutôt que de devoir gérer manuellement les appels HTTP.

- Spring Cloud : Eureka

Spring Cloud Eureka est un composant essentiel de l'écosystème Spring Cloud qui fournit un service de découverte de microservices. Grâce à Eureka, les services peuvent s'enregistrer auprès du serveur Eureka lors de leur démarrage, et ce dernier maintient un registre des services disponibles. Les autres services peuvent consulter ce registre pour découvrir dynamiquement les emplacements des services avec lesquels ils souhaitent communiquer, ce qui facilite la communication et l'intégration entre les différents microservices. Cette approche de découverte de services rend les applications basées sur le modèle de microservices résilientes, évolutives et hautement distribuées.

- Spring Cloud : Gateway

Spring Cloud Gateway est un composant de Spring Cloud qui agit en tant que passerelle d'API pour les architectures de microservices. Il permet de router les requêtes entrantes vers les services appropriés en fonction de règles de routage et de filtres définis, simplifiant ainsi la gestion des points d'entrée de l'API. Grâce à son intégration avec d'autres composants de Spring Cloud, il offre une solution complète pour la création de passerelles d'API robustes, améliorant la performance, la sécurité et la scalabilité de l'ensemble du système de microservices.

- Spring Admin

Spring Boot Admin est un outil open-source qui offre une interface utilisateur conviviale pour la gestion et le suivi des applications basées sur Spring Boot. Il permet aux administrateurs du système de surveiller l'état, les métriques, les journaux, les configurations et les informations de santé des différentes applications déployées sur leur infrastructure. Grâce à son tableau de bord centralisé, Spring Boot Admin offre une vue d'ensemble en temps réel de l'état de toutes les applications, ce qui facilite la détection rapide des problèmes potentiels et la prise de décisions éclairées en matière de gestion des applications. Cet outil est particulièrement utile dans les environnements de microservices, car il facilite la gestion et la surveillance centralisées de multiples instances de services déployées sur différents serveurs.



Figure 81:Logo Spring Boot Admin

- OptaPlanner

OptaPlanner est une bibliothèque open-source de planification de contraintes en Java, développée par Red Hat. Elle permet de résoudre des problèmes d'optimisation complexes en trouvant automatiquement les meilleures solutions possibles en respectant un ensemble de contraintes spécifiées. OptaPlanner est conçu pour gérer des problèmes tels que l'ordonnancement, l'affectation de ressources, l'optimisation de tournées, la planification de production et bien d'autres. Grâce à son puissant moteur de planification et à ses algorithmes d'optimisation avancés, OptaPlanner est capable de traiter des problèmes du monde réel avec des milliers de contraintes et de variables. Il peut être intégré facilement dans des applications Java et est particulièrement utile dans les domaines de la logistique, de la production, de la santé et de la distribution, où il contribue à améliorer l'efficacité, la rentabilité et la satisfaction des clients.

Nous utiliserons cet outil pour implémenter notre algorithme de planification optimisée.



Figure 82:Logo OptaPlanner

- Angular

Angular est un Framework front-end open-source développé par Google, basé sur TypeScript. Il permet de créer des applications web modernes et réactives en utilisant des composants réutilisables et une liaison de données bidirectionnelle. Avec des fonctionnalités avancées telles que le routage, la gestion des formulaires et la communication avec les API backend, Angular est devenu l'un des choix préférés des développeurs pour créer des applications web évolutives et performantes, notamment les Single Page Applications (SPA).



Figure 83:Logo Angular

- Angular Material

Angular Material est une bibliothèque d'interface utilisateur (UI) développée par l'équipe Angular de Google. Elle fournit une collection de composants d'interface utilisateur prêts à l'emploi, basés sur le design Material, qui suit les principes du Material Design de Google. Angular Material s'intègre parfaitement avec Angular et offre une solution pratique pour concevoir des interfaces modernes et esthétiques sans avoir à écrire beaucoup de code CSS personnalisé.



Figure 84:Logo Angular

- *Outils de gestion de projet*
 - **GitHub**

GitHub est une plateforme de développement logiciel basée sur le cloud, utilisée pour l'hébergement, le contrôle de version et la collaboration de code source. Elle permet aux développeurs de stocker et de gérer leurs projets de manière décentralisée grâce à Git, un système de contrôle de version distribué.



FIGURE 1.13 – *Logo GitHub*

5. Architecture globale de la solution

Il est indispensable de mener une étude rigoureuse pour concevoir une architecture répondant aux besoins de l'application. Une bonne architecture de l'application est primordiale pour garantir le succès et la pérennité de la solution. Elle permet d'organiser et de structurer de manière globale les différents éléments du système et les interactions entre eux. Dans cette section, nous présentons alors l'architecture globale de notre solution qui est une architecture à 3 niveaux : **architecture 3 tiers**.

- Niveau de présentation (Front-end) : C'est le premier niveau de l'architecture où l'interface utilisateur est développée avec le Framework web **Angular**. Ce dernier est utilisé pour créer une application web réactive et interactive qui sera exécutée dans le navigateur du client. Il s'agit de la partie visible par les utilisateurs finaux et qui leur permet d'interagir avec le système.
- Niveau d'application (Backend) : C'est le deuxième niveau de l'architecture où les traitements métier et les fonctionnalités de l'application sont implémentés. Dans ce cas, **Spring Boot** est utilisé pour créer des microservices, ce qui signifie que l'application est découpée en petits services indépendants qui peuvent être déployés et mis à l'échelle séparément. Chaque microservice a sa propre logique métier spécifique et communique avec les autres.

- Niveau de stockage de données (Base de données) : C'est le troisième niveau de l'architecture où les données sont stockées et gérées. Dans notre cas, une base de données **MySQL** est utilisée pour stocker les données de l'application. Les microservices peuvent communiquer avec la base de données pour lire, écrire et mettre à jour les données nécessaires à l'application.

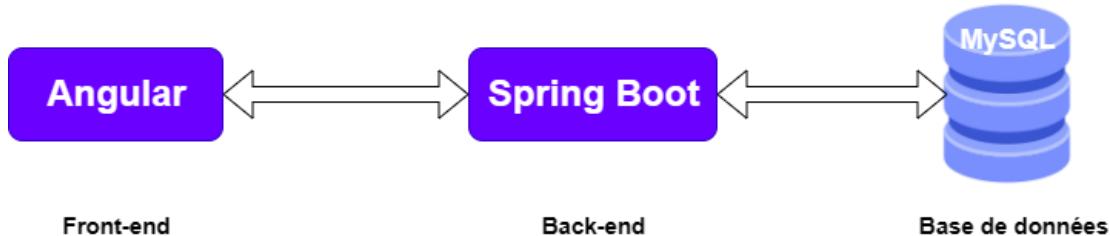


Figure 85: Architecture globale

En résumé, cette architecture à 3 niveaux sépare les préoccupations entre la présentation, les traitements métier et le stockage de données, ce qui rend le système modulaire, évolutif et plus facile à gérer. Chaque niveau fonctionne de manière indépendante et communique avec les autres niveaux via des interfaces définies, généralement des API REST dans le cas de cette architecture.

5.1. Architecture physique

L'architecture physique de notre solution est une architecture 3 tiers composé de 3 couches à savoir :

- Le tiers Client : le navigateur web grâce auquel les utilisateurs et l'application web pourront communiquer. L'application Angular s'exécute sur le navigateur du client et communique avec le backend en utilisant des appels HTTP/HTTPS.
- Le tiers Serveur : responsable du traitement des requêtes et de la logique métier. Il reçoit les requêtes envoyées par le client et renvoie des réponses sous format JSON. Cette communication s'établit par le protocole http.
- Le tiers Base de données : C'est l'espace de stockage dans lequel les données sont stockées et accessibles par le serveur.

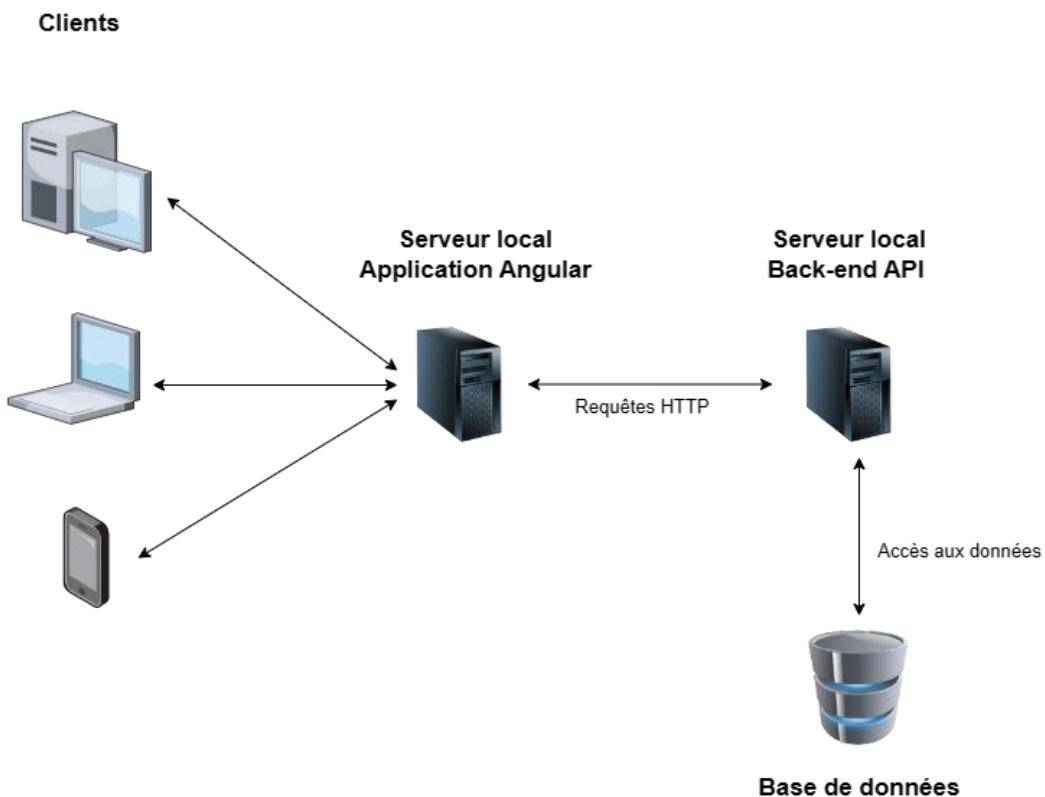


Figure 86:Architecture physique

5.2. Architecture logique

La figure suivante décrit clairement l'architecture logique globale adoptée dans notre projet.

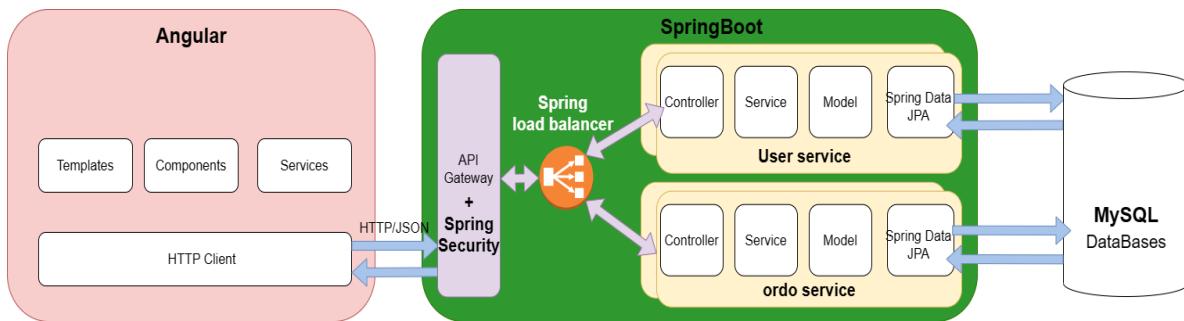


Figure 87:architecture logistique

Dans cette architecture logique :

- Le front-end représente l'application Angular elle-même. Elle gère la présentation et l'interface utilisateur, interagissant avec les microservices via des appels HTTP/HTTPS par l'intermédiaire d'une passerelle (API Gateway) pour récupérer les données ou effectuer des actions.
- Le backend est constitué premièrement d'une passerelle **Spring API Gateway** qui intègre la dépendance **Spring Security** afin de centraliser et de sécuriser les échanges entre la couche supérieure et les couches inférieures de l'application ainsi que les accès aux ressources ; Ensuite, il est composé de microservices qui représentent chacun une partie spécifique de la logique métier. Ils reçoivent les demandes du front-end, effectuent les opérations nécessaires et renvoient les réponses appropriées ; Un équilibrEUR de charge **Spring Load Balancer** est placé entre ceux-ci pour assurer l'équilibre de charge entre les différentes instances des microservices.
- La couche de stockage de données où la base de données **MySQL** est située. Les microservices accèdent à la base de données pour effectuer les opérations de lecture, écriture et mise à jour des données.

La partie Backend a été développée en respectant l'architecture **DDD** (Domain Driven Design). C'est un patron de conception selon lequel la structure et le langage du code logiciel (nom de classes, méthodes, variables) doivent correspondre au domaine métier. Il se base sur deux principes :

- La complexité du domaine métier doit se refléter dans le modèle.
- L'accent doit être mis sur le domaine et la logique associée et non pas sur les contraintes techniques.

6. Conclusion

Dans ce chapitre, nous avons tout d'abord présenter les différents acteurs externes de la solution ainsi que la spécification des besoins de l'application. Nous avons par la suite énuméré les différents outils et langages utilisés pour la réalisation de ce projet. Finalement nous avons essayé de présenter l'architecture adoptée pour notre application. Le prochain chapitre est consacré à la présentation de la démarche de travail et la planification qui en découle.

Chapitre 7 : Méthodologie adoptée et planification du projet

1. Introduction

Afin d'accomplir notre projet de façon rigoureuse et professionnelle, il est indispensable de choisir une méthode de management de projet. Dans la partie qui suit, nous allons parler de la méthodologie adoptée au cours de ce travail ainsi la planification qui en découle.

2. Méthodologie adoptée : Kanban

Le choix d'une méthodologie de travail est fortement corrélé à la nature du projet à accomplir, ainsi que ses contraintes et les conditions de travail dans lesquelles il est réalisé. Choisir la meilleure méthode de travail revient à faciliter grandement la réalisation du projet, ceci de façon structurée et simplifiée.

Pour ce faire nous avons choisi la méthodologie **Agile KANBAN** comme cadre de gestion de projet. C'est une méthodologie de développement logiciel Agile et DevOps. Il repose sur un travail effectué en toute transparence et une communication en temps réel de la capacité. Les tâches sont représentées visuellement sur un tableau Kanban. Ainsi, les membres de l'équipe peuvent voir l'état de chaque tâche à tout moment.

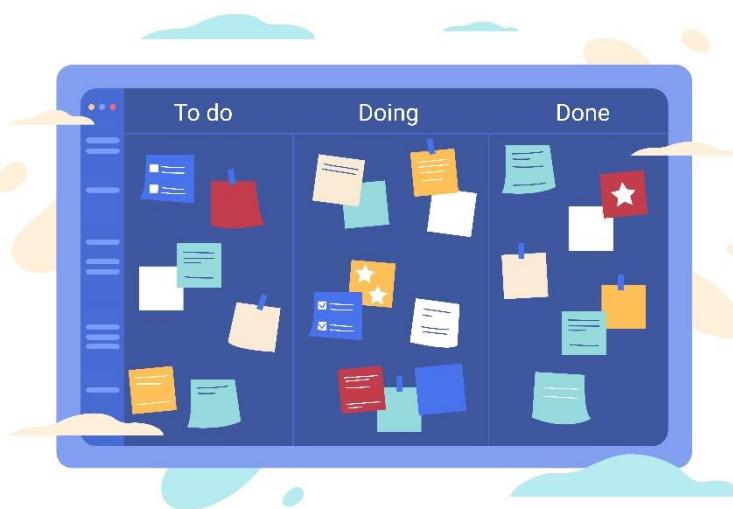


Figure 88:Tableau Kanban

Plusieurs raisons ont justifié ce choix :

- **Flexibilité et agilité** : La méthodologie Kanban est reconnue pour sa flexibilité et sa capacité à s'adapter à des flux de travail variables. Étant donné que notre projet de développement logiciel impliquait des tâches de différentes natures et priorités, Kanban s'est avéré être le choix parfait pour gérer cette diversité de manière agile.
- **Visualisation du flux de travail** : Le développement d'une application web de planification et d'ordonnancement nécessitait une gestion visuelle des tâches et des étapes du projet. Kanban offre un tableau visuel, où chaque étape du flux de travail est représentée par une colonne, permettant ainsi de suivre facilement le progrès de chaque tâche, d'identifier les goulets d'étranglement et de prendre des décisions éclairées.
- **Contrôle de la charge de travail** : Le développement de l'application impliquait une équipe de développement avec diverses compétences. En utilisant des limites de travail en cours pour chaque colonne du tableau Kanban, nous avons pu contrôler la charge de travail, évitant ainsi la surcharge et le risque de sur-promettre tout en garantissant une livraison régulière et prévisible.

3. Backlog du produit

Le backlog du produit est la liste des fonctionnalités attendues d'un produit. Plus exactement, au-delà de cet aspect fonctionnel, il contient tous les éléments nécessaires pour le travail en équipe. Après une compréhension approfondie des User Stories, nous avons pu dégager le backlog produit de notre application qui se résume dans le tableau ci-après.

Tableau 6:Backlog produit

ID	Fonctionnalité	Priorité	Durée
01	Planification et Ordonnancement optimisée des commandes	élevée	20 jours
02	Gestions des utilisateurs , des rôles et des permissions	moyenne	10 jours
03	Gestions des ordres de fabrication et de traitement	moyenne	7 jours
04	Gestion de la chaîne de production	élevée	10 jours
05	Gestion des traitements	moyenne	7 jours
06	Gestion des clients	faible	7 jours
07	Gestion des stocks	moyenne	7 jours

4. Planification

A partir du backlog produit, nous utilisons la méthodologie Kanban pour l'élaboration de la planification de notre travail. Le tableau qui suit présente la planification pour ce projet.

Tableau 7: Planifications des tâches

phase	Objectifs	Durée
01	<ul style="list-style-type: none"> Préparation de l'environnement de travail et des outils utilisés. Authentification & Gestion des rôles et utilisateurs. 	Du 20 /02/2023 Au 31/03/2023
02	<ul style="list-style-type: none"> Gestion de la chaîne de production et des horaires de disponibilité. Gestion des traitements. Gestion des clients 	Du 03 /04/2023 Au 05/05/2023
03	<ul style="list-style-type: none"> Gestion des ordres de fabrication. Gestion du stocks 	Du 08 /05/2023 Au 26/05/2023
04	<ul style="list-style-type: none"> . Développement et Intégration de l'algorithme de planification optimisée. Historisation des planifications. 	Du 29 /05/2023 Au 07/07/2023

5. Outil de Planification

Pour mettre en œuvre notre planification, nous avons choisi **Trello** comme outil de planification. C'est une application de gestion de projet en ligne qui utilise des tableaux, des listes et des cartes pour organiser les tâches et faciliter la collaboration au sein d'une équipe. Simple et intuitive, elle permet de visualiser et de suivre facilement le progrès des projets, attribuer des tâches, définir des échéances et partager des informations, ce qui en fait un outil populaire pour la gestion de projets personnels et professionnels.



Figure 89: Logo Trello

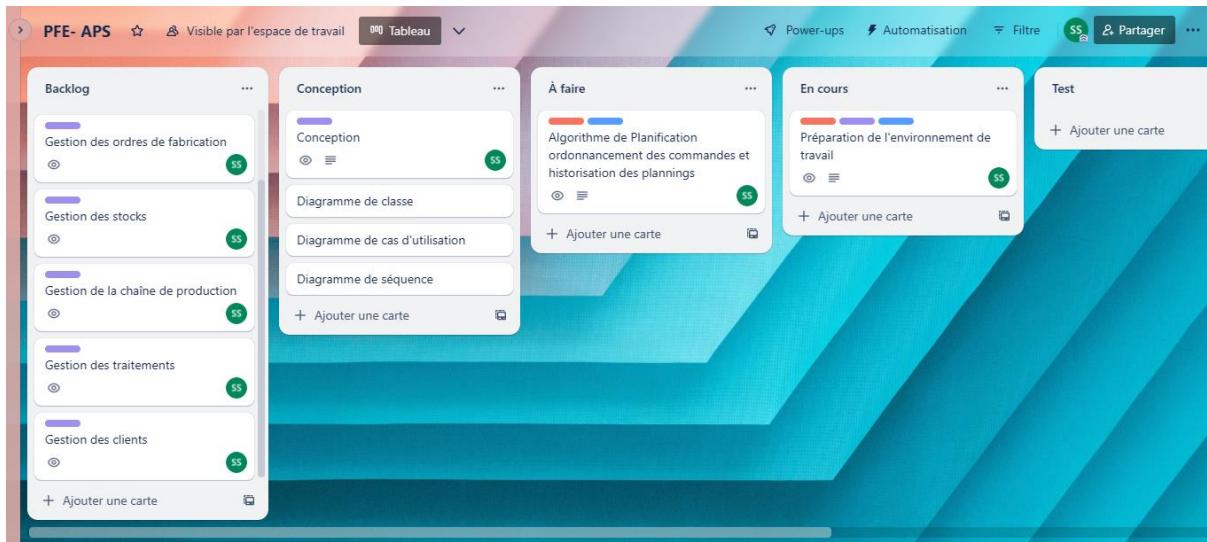


Figure 90: Planification réalisée sur Trello

6. Conclusion

Ce chapitre était consacré en première approche à la description détaillée de la démarche de réalisation du projet. Il s'agissait par la suite de présenter la planification du travail subdivisée en 05 phases. Afin de suivre un avancement logique dans ce rapport, une étude de la première phase de ce planning fera l'objet du prochain chapitre

Chapitre 8 : Mise en place de l'authentification et de la gestion des rôles et utilisateurs

1. Introduction

Dans ce chapitre, nous présentons la conception et la réalisation des fonctionnalités de la première phase de notre projet. Nous commencerons par décrire les exigences fonctionnelles de chaque fonctionnalité, puis nous présenterons les diagrammes relatifs à cette phase. Enfin, nous terminerons par la partie réalisation.

2. Description des exigences fonctionnelles

Nous présentons les user Stories de chaque fonctionnalité dans le tableau

Tableau 8:user stories phase 1

Fonctionnalité	User story	ID
Authentification	<ul style="list-style-type: none"> En tant qu'utilisateur, je veux m'authentifier. 	1
Gestion des utilisateurs	<ul style="list-style-type: none"> En tant qu'utilisateur privilégié, je veux créer, supprimer, modifier et consulter les utilisateurs. 	2
Affectation des rôles et permissions	<ul style="list-style-type: none"> En tant qu'utilisateur privilégié, je veux affecter et modifier les rôles des utilisateurs. 	3
Gestion des demandes utilisateurs	<ul style="list-style-type: none"> En tant qu'utilisateur privilégié, je veux consulter, accepter ou refuser les demandes utilisateur. 	4
Création d'une demande utilisateur	<ul style="list-style-type: none"> En tant qu'employé dans l'entreprise, je veux envoyer une demande d'accès à l'application. 	5
Gestion du profil	<ul style="list-style-type: none"> En tant qu'utilisateur, je veux consulter et modifier mon profil 	6

3. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation illustré , représente les cas d'utilisation de la phase 1.

3.1. Présentation du diagramme de cas d'utilisation

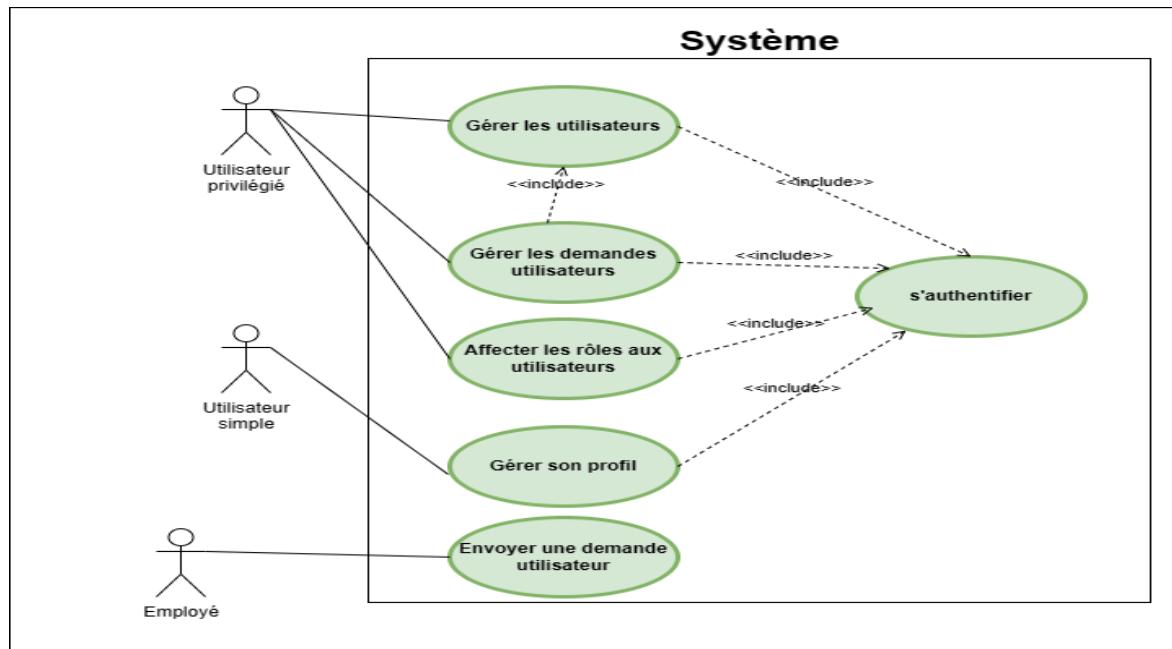


Figure 91:Diagramme de cas d'utilisation de la phase 1

3.2. Description textuelle des sous-cas d'utilisation

Nous allons présenter dans cette partie la description textuelle de chaque sous-cas d'utilisation de cette phase à savoir : l'authentification, la gestion des utilisateurs, l'affectation des rôles, la gestion des demandes utilisateur et l'envoi de demande utilisateur.

Pour avoir accès à l'application, l'utilisateur doit s'authentifier en fournissant ses données personnelles (login et password). Le tableau suivant présente la description textuelle du sous-cas d'utilisation de gestion d'authentification de l'application.

Tableau 9:Description textuelle « Authentification »

Titre	S'authentifier
Acteur	Utilisateur
Précondition	L'utilisateur dispose d'un compte et de ses identifiants sur l'application et accède à la page de connexion.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur doit saisir son email et son mot de passe. 2. Le système charge la page d'accueil. 3. Le système affiche la page d'accueil.
Post-condition	2. Si les identifiants de connexion sont incorrects, un message d'erreur s'affiche à l'écran.

Pour la gestion des utilisateurs et l'affectation des rôles, l'application permet aux utilisateurs dont le rôle permet d'avoir accès à cette fonctionnalité d'ajouter, modifier, supprimer et consulter la liste des données des utilisateurs, tout en permettant d'affecter des rôles aux utilisateurs.

Nous présentons la description textuelle de ces sous-cas d'utilisation dans le tableau suivant.

Tableau 10:description textuelle « Gérer les utilisateurs et affecter des rôles »

Titre	Gérer les utilisateurs et affecter des rôles
Acteur	Utilisateur privilégié
Précondition	L'utilisateur s'est authentifié sur un compte dont le rôle permet l'accès à cette fonctionnalité.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la section « User management » 2. Le système affiche la page relative à la gestion d'utilisateurs. 3. L'utilisateur peut ajouter un nouvel utilisateur. <ol style="list-style-type: none"> 2.3.1. Il remplit le formulaire d'ajout et le valide 2.3.3. Le système vérifie les saisies et valide l'ajout en cas de valeurs correctes. 2.4. Il peut modifier les informations d'un utilisateur 2.5. Il peut supprimer un utilisateur <ol style="list-style-type: none"> 2.5.1. Le système demande une confirmation de suppression. 2.5.2. L'utilisateur confirme la suppression 2.5.3. Le système supprime l'utilisateur et confirme la suppression.
Exception	<ol style="list-style-type: none"> 2.1. Si le compte utilisateur ne dispose pas du rôle nécessaire à l'accès de cette fonctionnalité, le système affiche une page désignant le non accès à celle-ci. 2.3.2. Si les saisies sont incorrectes, un message d'erreur s'affiche

Pour le sous-cas de gestion de profil, l'application permet à chaque utilisateur de modifier les informations relatives à son profile.

Pour le sous-cas de demande utilisateur, le tableau qui suit présente sa description textuelle.

Tableau 11:Description textuelle « envoyer une demande utilisateur »

Titre	Envoyer une demande utilisateur.
Acteur	Employé à SIVO
Précondition	L'employé ne possède pas de compte et n'a donc pas accès à l'application.
Scénario nominal	<ol style="list-style-type: none"> 1. L'employé accède à la page de connexion de l'application. 2. Il clique sur « Register » et remplit le formulaire d'inscription. 3. Il valide le formulaire. 4. Le système vérifie les saisies et confirme l'envoi de la demande utilisateur si les valeurs sont correctes. 5. Le système affiche un message de confirmation d'envoi de la demande.
Post-condition	2.2. Si les informations du formulaire sont incorrectes, un message d'erreur s'affiche à l'écran.

Pour le sous-cas de gestion des demandes utilisateurs, le tableau qui suit présente sa description textuelle.

Tableau 12:description textuelle « gestion des demandes utilisateur »

Titre	Gérer les demandes utilisateurs
Acteur	Utilisateur privilégié
Précondition	L'utilisateur s'est authentifié sur un compte dont le rôle permet l'accès à cette fonctionnalité.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la section « User management » puis « User requests » 2. Le système affiche la page relative à la gestion des demandes utilisateurs. <ol style="list-style-type: none"> 2.1. L'utilisateur peut valider une demande utilisateur. 2.1.1. Le système redirige l'utilisateur vers page de création d'un nouvel utilisateur en renseignant les champs de la demande utilisateur validée. 2.1.2. L'utilisateur valide le formulaire du nouvel utilisateur. 2.1.3. Le système vérifie le formulaire et confirme la création du nouvel utilisateur. 2.1.4. Le système redirige l'utilisateur vers la page de gestion des demandes. 2.2. L'utilisateur peut refuser la demande utilisateur. 2.2.1. Le système demande une confirmation de refus et supprime la demande utilisateur si la confirmation est validée.
Exception	<ol style="list-style-type: none"> 2.1. Si le compte utilisateur ne dispose pas du rôle nécessaire à l'accès de cette fonctionnalité, le système affiche un page désignant le non accès à celle-ci. 2.3.2. Si les saisies sont incorrectes, un message d'erreur s'affiche

4. Diagrammes de séquence

Nous allons présenter dans cette partie les diagrammes de séquence relatifs à chaque fonctionnalité de la phase 1 décrites plus haut.

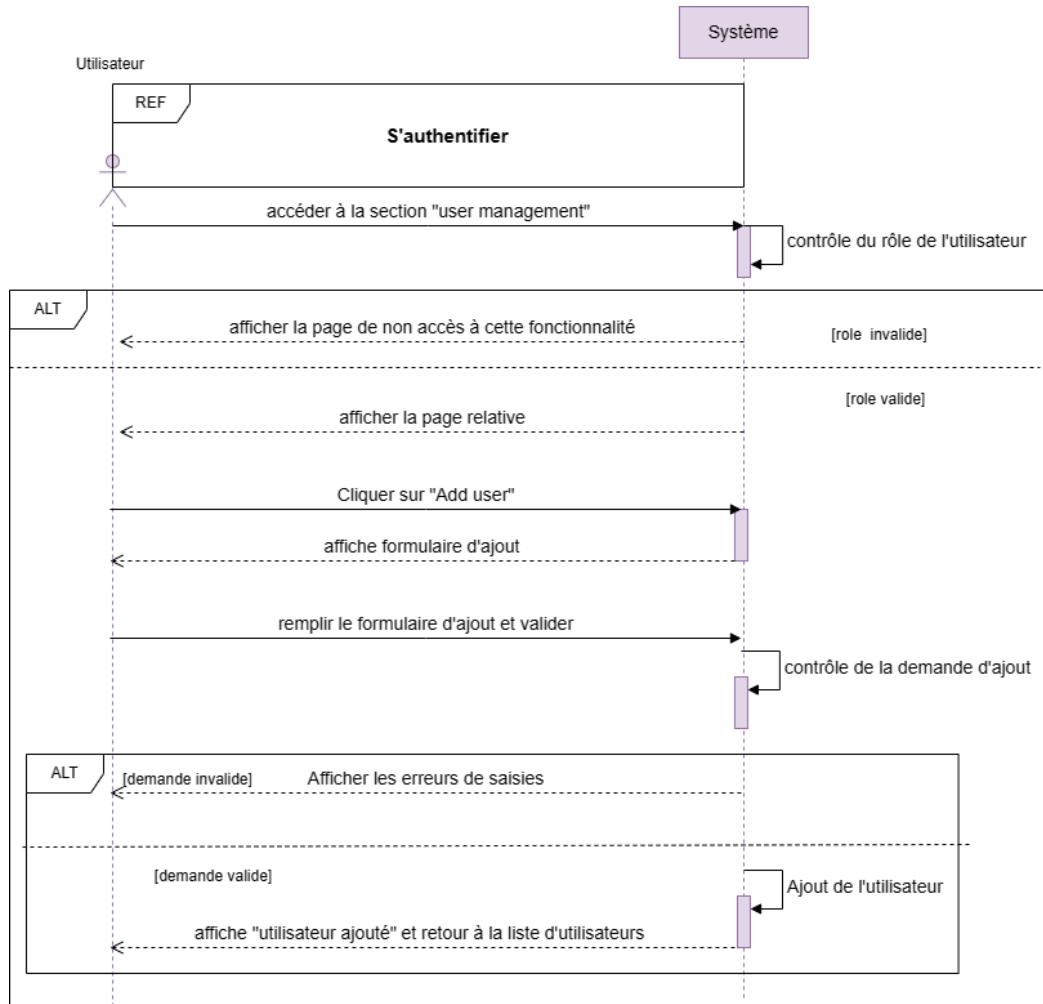


Figure 92: Diagramme de séquence d'ajout d'un utilisateur

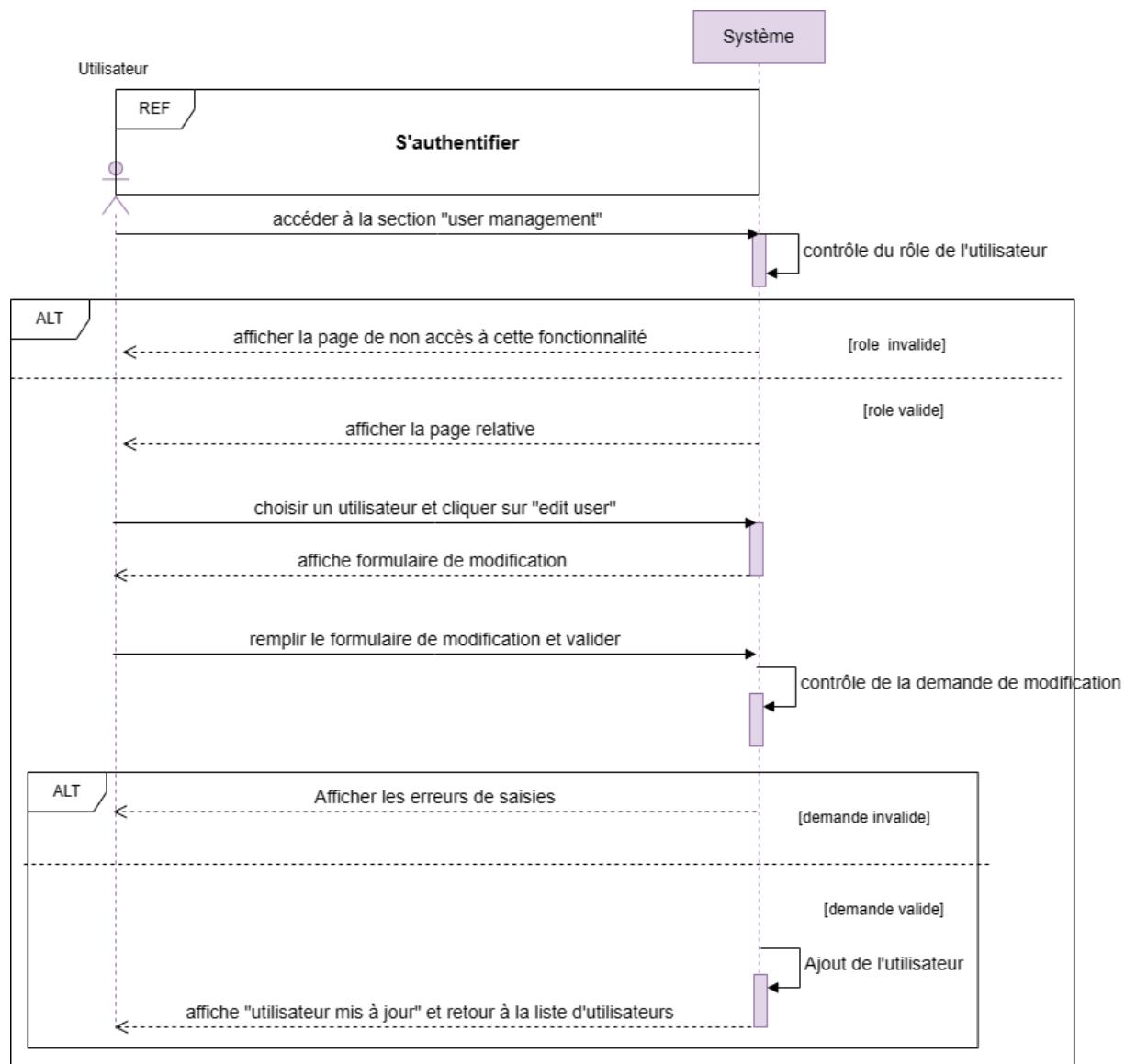


Figure 93: Diagramme de séquence de modification d'un utilisateur

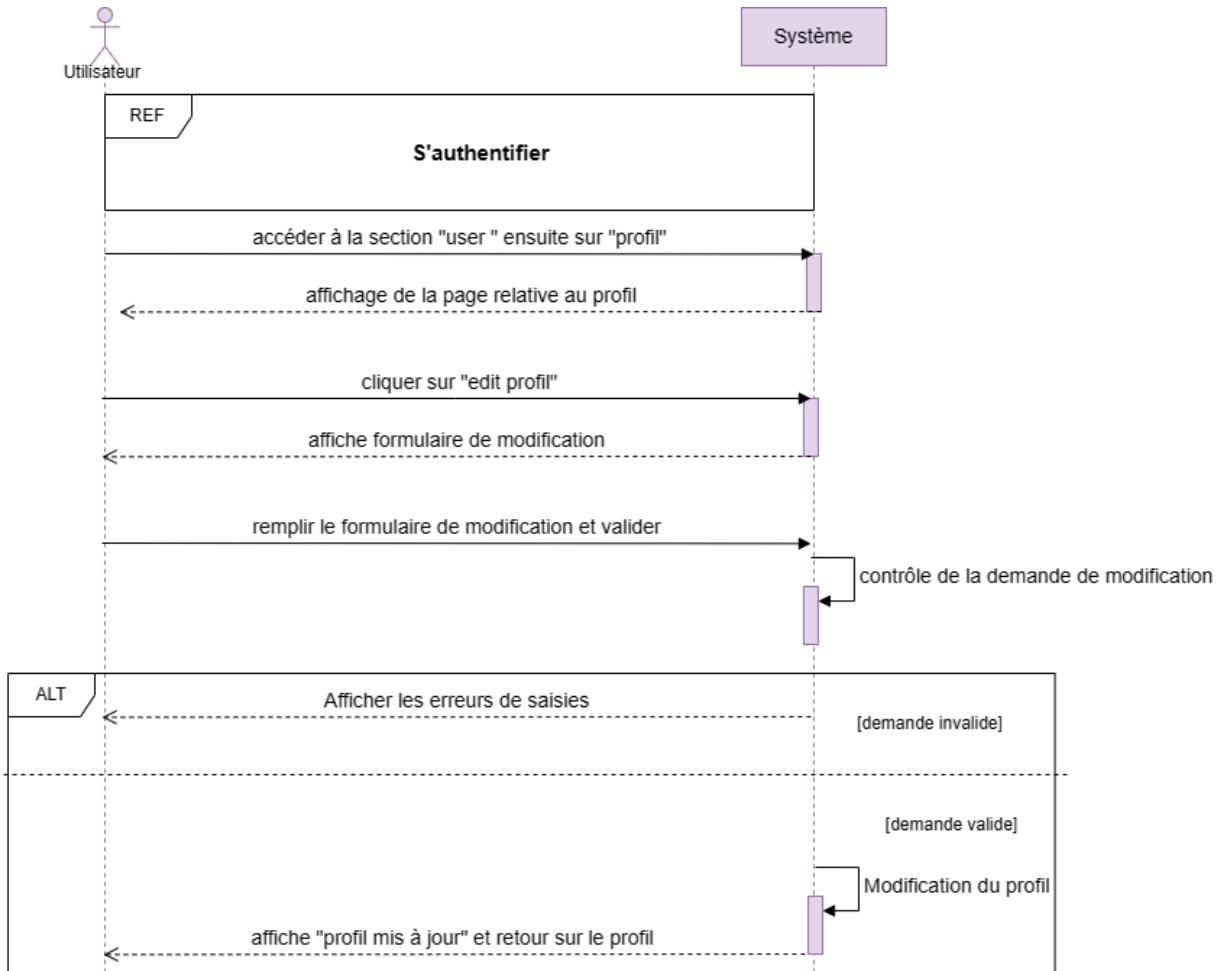


Figure 94: Diagramme de séquence de gestion de profil

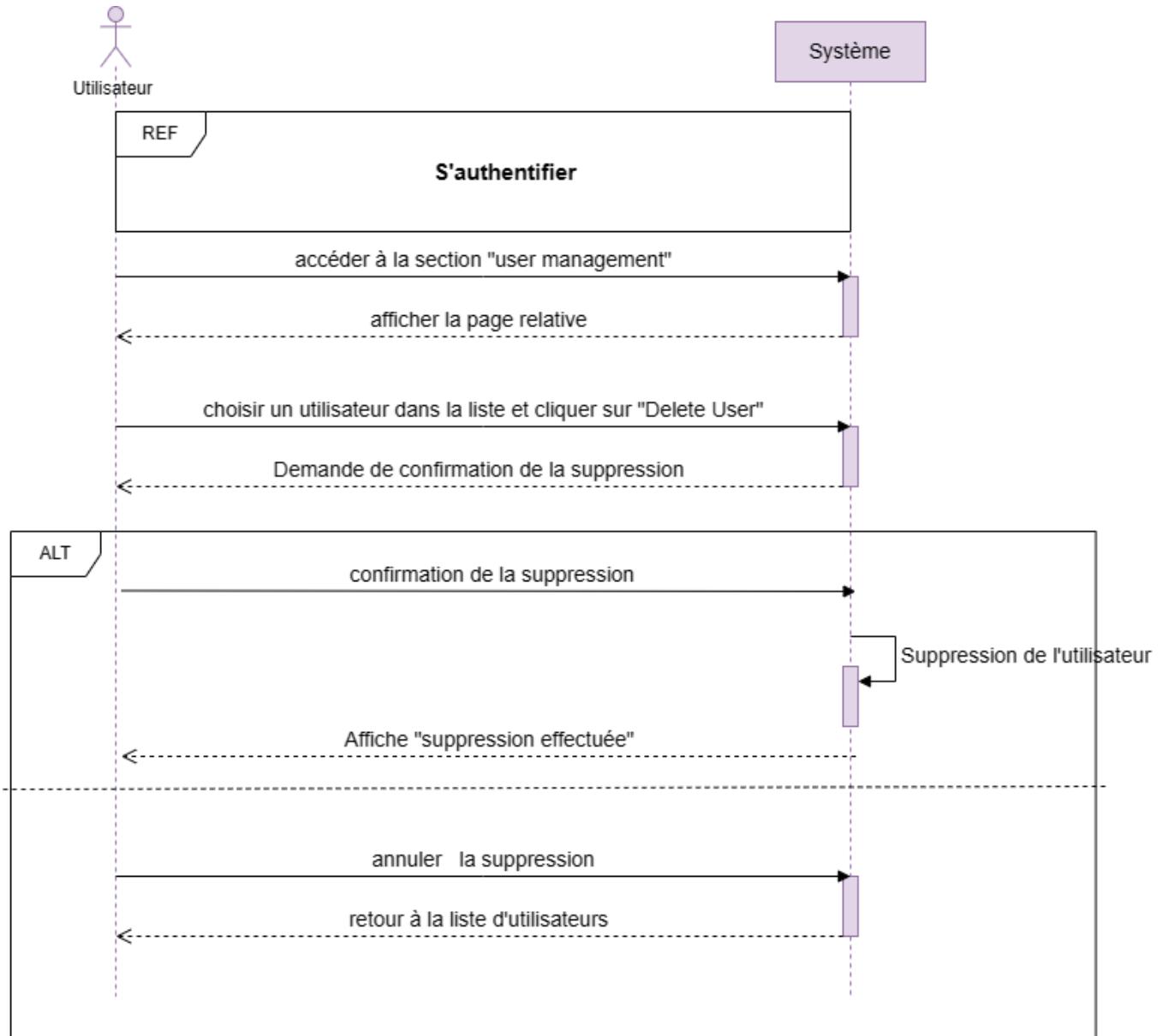


Figure 95: Diagramme de séquence de suppression d'utilisateur

Les diagrammes de séquence relatifs à la fonctionnalité « gestion des demandes utilisateurs » seront présentés dans l'annexe.

Suite à l'observation globale des différentes fonctionnalités de la phase 1 dans son ensemble, nous passons désormais à l'élaboration du diagramme de classe.

5. Diagramme de classe

Nous décrivons dans cette section la structure logique de notre phase en modélisant ses classes, ses différents attributs, ses méthodes et les relations entre ces classes. Le diagramme de classe se présente dans la figure 3.7.



Figure 96 : Diagramme de classe de la phase 01

6. Réalisation

Dans cette partie, nous présentons quelques interfaces réalisées au cours de cette première phase de notre planification.

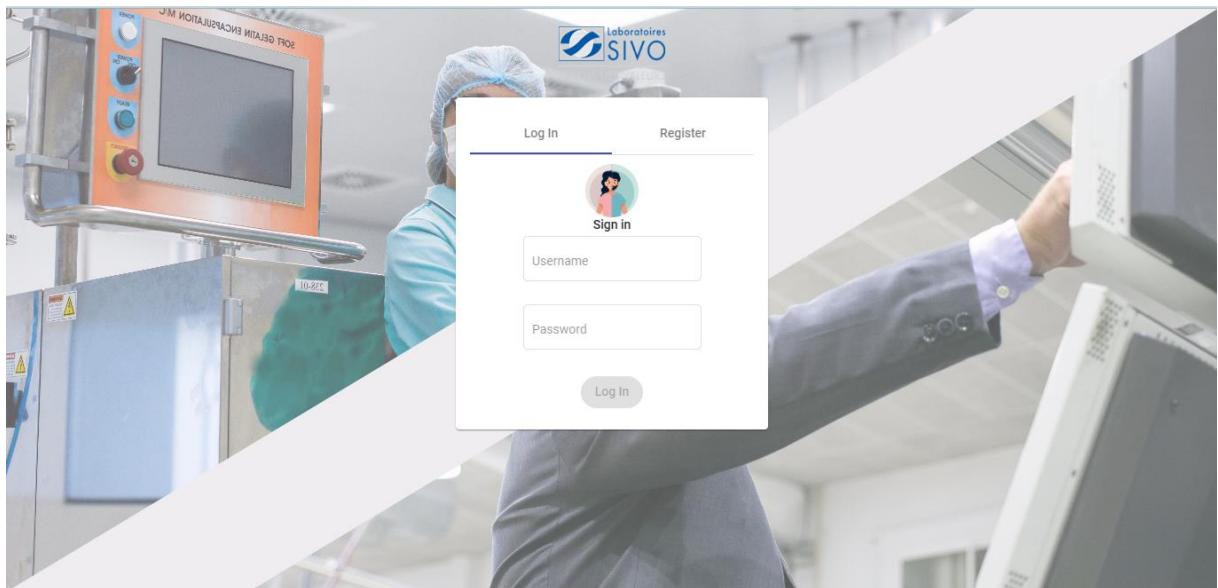


Figure 97:Interface de login

L'échantillon de code présenté ci-après est un bout de code qui permet d'enregistrer les demandes d'inscription des nouveaux utilisateurs.

```
register(){
  if(localStorage.getItem("isRegistered") ==="yes"){
    this.openDialog("Already submitted !", "You already submitted for a registration.")
  }
  else{
    this.userService.newRegisterRequest(this.registerForm.value).subscribe({
      next : (response : any) =>{
        console.log(response)
        let title = "Registration submitted"
        let message = response.firstName +" "+response.lastName+, you have successfully submitted your
        this.openDialog( title, message)
      },
      error : (error: any ) =>{
        console.log(error.message)
        this.openDialog("Oups !", "An error occurred : "+error.message)
      }
    })
  }
}
```

Figure 98:Echantillon de code du composant LoginPage

La page de Login de l'application permet aux utilisateurs de l'application de se connecter et d'accéder ainsi aux fonctionnalités accessibles par leurs rôles. Il permet de sécuriser l'accès à l'application et de contrôler l'identité de ceux qui l'utilise. Lorsque l'utilisateur entre des identifiants corrects (email et mot de passe), il est dirigé vers la page d'accueil de l'application présentée par la figure 3.9.

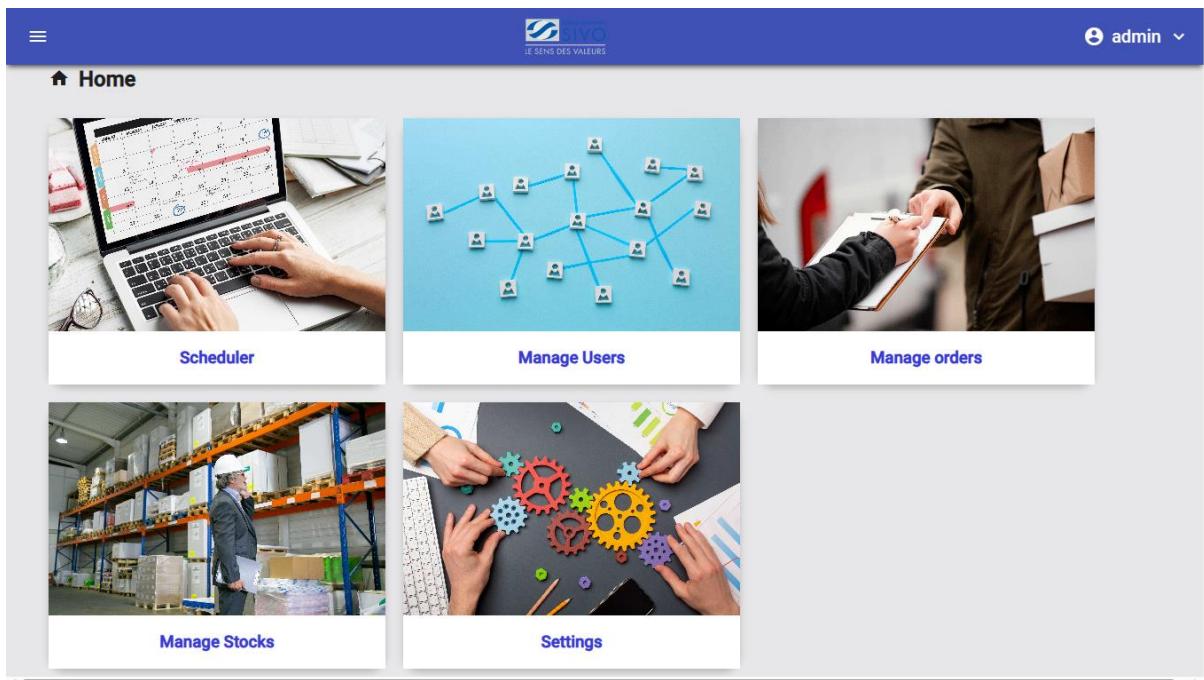


Figure 99:Interface de la page d'accueil

L'accès aux ressources de l'application est sécurisé par l'intégration API Gateway qui implémente Spring Security tel que le montre le bout de code suivant.

```

43     protected void configure( HttpSecurity httpSecurity) throws Exception {
44
45         httpSecurity.cors();
46         httpSecurity.csrf().disable()
47             .authorizeHttpRequests().antMatchers("/api/user/authenticate","/api/user/newRegisterRequest").permitAll()
48             .antMatchers(org.springframework.http.HttpHeaders.ALLOW).permitAll()
49             .anyRequest().authenticated()
50             .and()
51             .exceptionHandling().authenticationEntryPoint(jwtAuthenticationEntryPoint)
52             .and()
53             .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);
54
55         httpSecurity.addFilterBefore(jwtRequestFilter, UsernamePasswordAuthenticationFilter.class);
56     }
57
58     @Bean
59     public PasswordEncoder passwordEncoder() {
60
61         return new BCryptPasswordEncoder();
62     }
63
64
65     public void configureGlobal(AuthenticationManagerBuilder authenticationManagerBuilder) throws Exception {
66         authenticationManagerBuilder.userDetailsService(jwtService).passwordEncoder(passwordEncoder());
67     }

```

Figure 100:Echantillon du code de configuration Spring Security

Ci-dessous l'interface de gestion d'utilisateur, qui permet d'assurer les opérations d'ajout, de suppression, de modification et de consulting des utilisateurs de l'application.

The screenshot shows a user management interface for a system named SIVO. At the top, there's a blue header bar with the SIVO logo and the word 'admin'. Below it, the main title is 'Users Management' with a subtitle 'Users List'. A search bar labeled 'Filter' is followed by a 'New User' button. The main area displays a table with two rows of user data:

Profile	Name	Email	Role
	Mr. admin admin	anesyveets@gmail.com	Admin
	Mr. sena steevy	user1@gmail.com	user

Below the table, there are search filters for 'Gender' (Male), 'First Name' (admin), 'Last Name' (admin), 'Email' (anesyveets@gmail.com), 'post' (Admin), and 'Roles' (Admin). At the bottom right are 'Edit User' and 'Delete User' buttons.

Figure 101:Interface de gestion d'utilisateurs

Cette interface est accessible uniquement par les utilisateurs dont le rôle le permet. On peut également réaliser des opérations de filtre sur la liste des utilisateurs comme le montre la figure

7. Conclusion

Dans ce chapitre, il était question de présenter tout d'abord les différentes conceptions élaborées pour la réalisation de la première phase de notre projet qui est la mise en place de l'authentification et la gestion des utilisateurs et des rôles. Ensuite nous avons exposé de manière générale les réalisations menées à partir de ces conceptions. Le prochain chapitre concerne la deuxième phase de notre planification qui est la mise en place de la gestion de la chaîne de production, la gestion des traitements, ainsi que la gestion des clients

Chapitre 9 : Mise en place de la gestion de la chaîne de production, des traitements et des clients.

1. Introduction

Dans ce chapitre, nous présentons la phase 02 de notre planification qui est la mise en place de la gestion de la chaîne de production, des traitements et des clients. Nous exposerons tout d'abord la description des besoins fonctionnels qui concernent cette phase, l'ensemble des conceptions réalisées pour l'implémentation par la suite, et enfin les réalisations qui en résultent.

2. Description des exigences fonctionnelles

Cette section est consacrée à la description des besoins fonctionnels de la phase 02 à savoir : la gestion de la chaîne de production, la gestion des traitements et la gestion des clients. Le tableau suivant présente les user stories de chaque fonctionnalité.

Tableau 13:User stories de la phase 02

Fonctionnalité	User story	ID
Gestion de la chaîne de production	<ul style="list-style-type: none"> En tant qu'utilisateur responsable de la production, je veux ajouter, modifier, supprimer ou consulter les phases de la chaîne de production. 	1
Gestion des traitements	<ul style="list-style-type: none"> En tant qu'utilisateur privilégié, je veux créer, supprimer, modifier et consulter les traitements de chaque phase de production. 	2
Gestion des clients	<ul style="list-style-type: none"> En tant qu'utilisateur privilégié, je veux créer, modifier, supprimer ou consulter les clients. 	3

3. Diagramme des cas d'utilisation

3.1. Présentation du diagramme des cas d'utilisation

La figure suivante présente le diagramme des cas d'utilisation relatif à la phase 02 du projet.

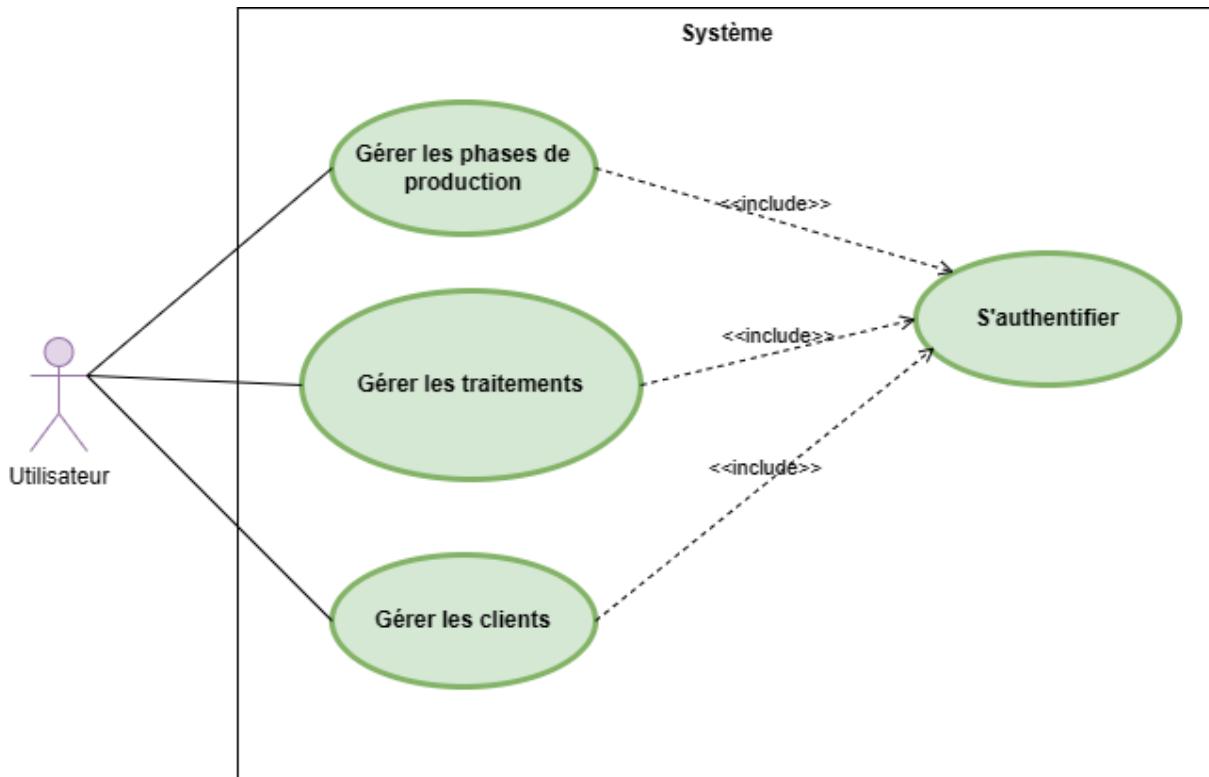


Figure 102: Diagramme des cas d'utilisation de la phase 02.

3.2. Description textuelle des cas d'utilisation

Nous allons présenter dans cette partie la description textuelle de chaque sous-cas d'utilisation de cette phase à savoir : la gestion de la chaîne de production, la gestion des traitements et la gestion des clients.

Pour gérer les phases de production, l'utilisateur doit disposer des permissions nécessaires et accéder à la page relative à celle-ci. Le tableau 4.2 décrit de façon détaillée ce sous-cas d'utilisation.

Tableau 14: Description textuelle du sous-cas « gérer les phases de production »

Titre	Gérer les phases de production
Acteur	Utilisateur
Précondition	L'utilisateur s'authentifie
Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur « Manage production Line » dans le menu de navigation. 2. Le système charge la page relative à la gestion de la chaîne de production. 3. L'utilisateur peut créer, modifier, supprimer et consulter les phases de production.
Exception	<ol style="list-style-type: none"> 1. Si l'utilisateur ne dispose pas des permissions nécessaires, le système redirige vers la page de non accès.

Le prochain tableau présente la description textuelle du sous-cas de gestion de traitements.

Tableau 15: Description textuelle du sous-cas « gérer les traitements »

Titre	Gérer les traitements
Acteur	Utilisateur
Précondition	L'utilisateur s'authentifie
Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur « Manage Treatments » dans le menu de navigation. 2. Le système charge la page relative à la gestion des traitements. 3. L'utilisateur peut créer, modifier, supprimer et consulter les traitements.
Exception	<ol style="list-style-type: none"> 1. Si l'utilisateur ne dispose pas des permissions nécessaires, le système redirige vers la page de non accès.

Le tableau suivant décrit le sous-cas d'utilisation qui concerne la gestion des clients.

Tableau 16: Description textuelle du sous-cas « gérer les clients »

Titre	Gérer les clients
Acteur	Utilisateur
Précondition	L'utilisateur s'authentifie
Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur « Manage Clients » dans le menu de navigation. 2. Le système charge la page relative à la gestion des clients. 3. L'utilisateur peut créer, modifier, supprimer et consulter les phases de production.
Exception	<ol style="list-style-type: none"> 1. Si l'utilisateur ne dispose pas des permissions nécessaires, le système redirige vers la page de non accès.

4. Diagrammes de séquence

Dans cette partie, nous présentons les différents diagrammes de séquence relatifs à chaque sous-cas d'utilisation de cette phase.

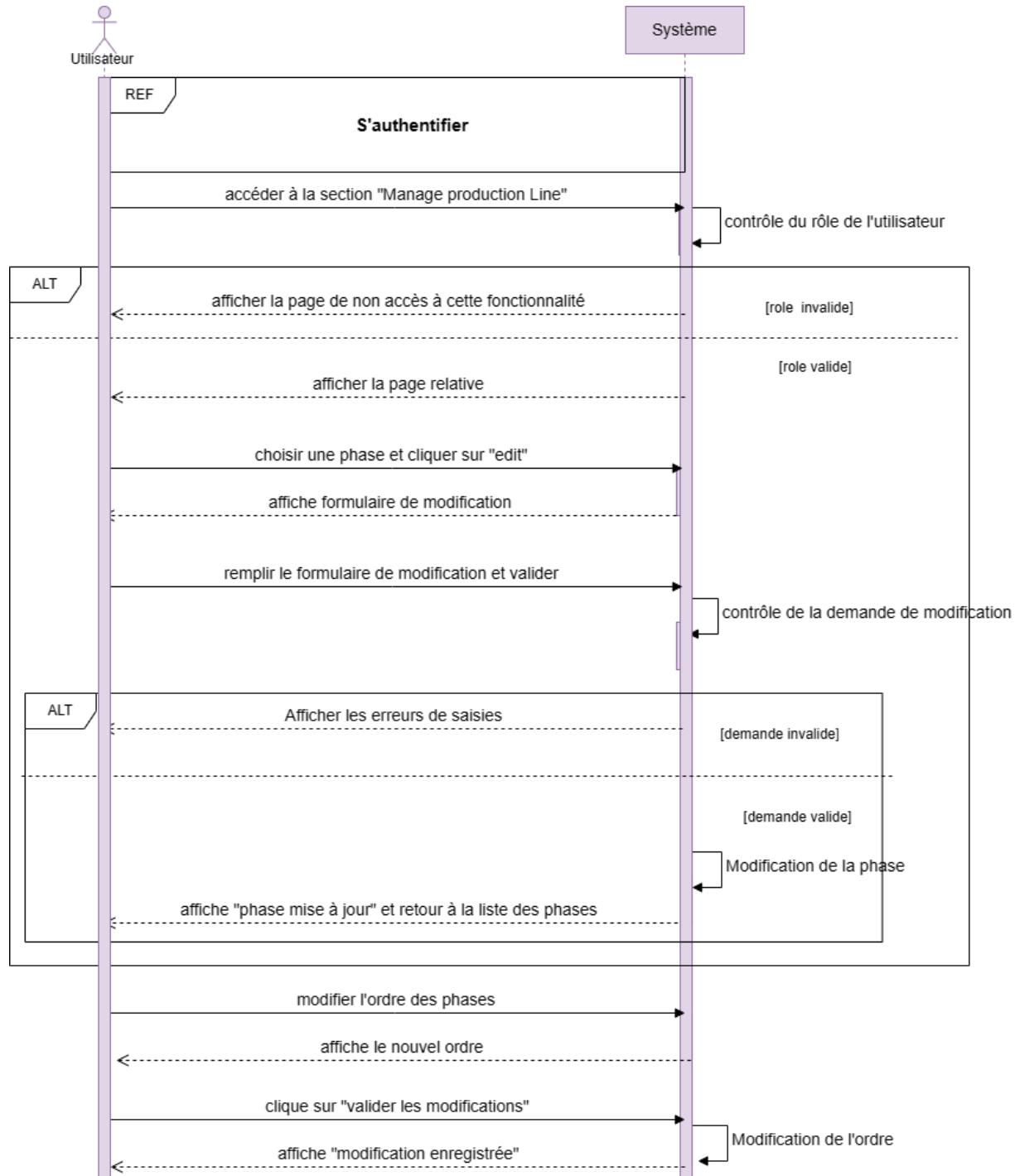


Figure 103:Diagramme de séquence de modification de la chaîne de production.

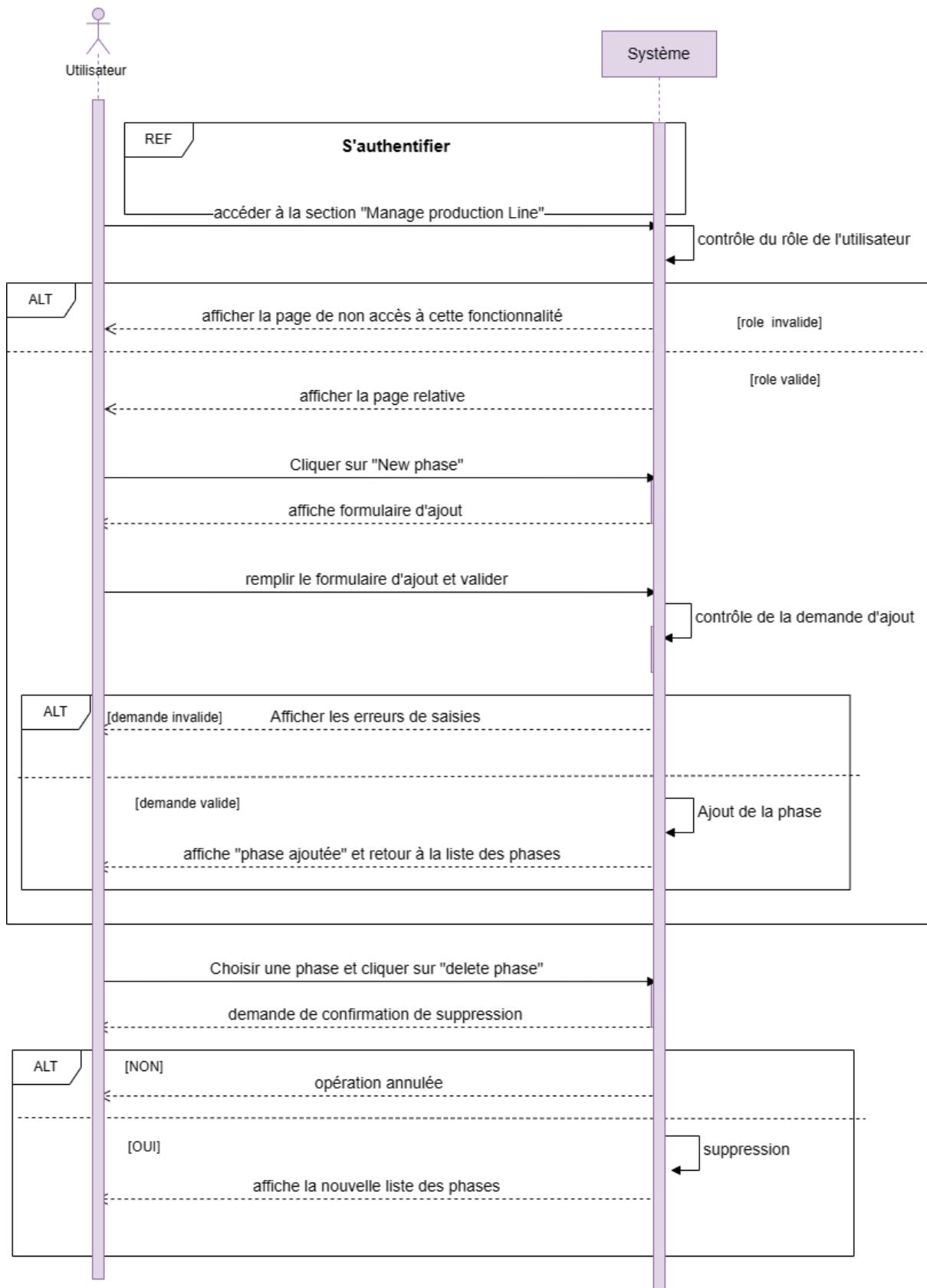


Figure 104: Diagramme de séquence d'ajout et de suppression de phase.

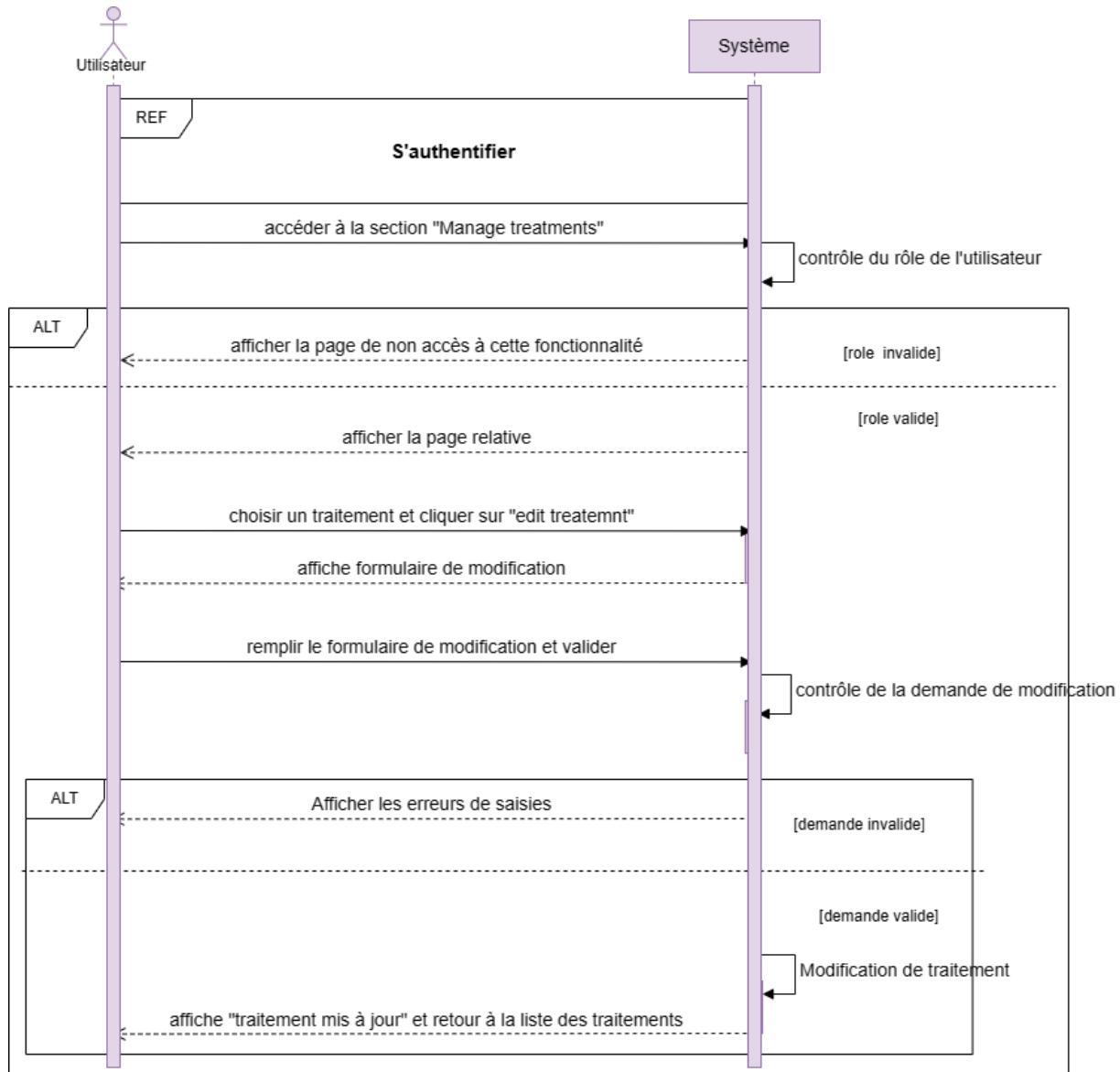


Figure 105: Diagramme de séquence de modification de traitement.

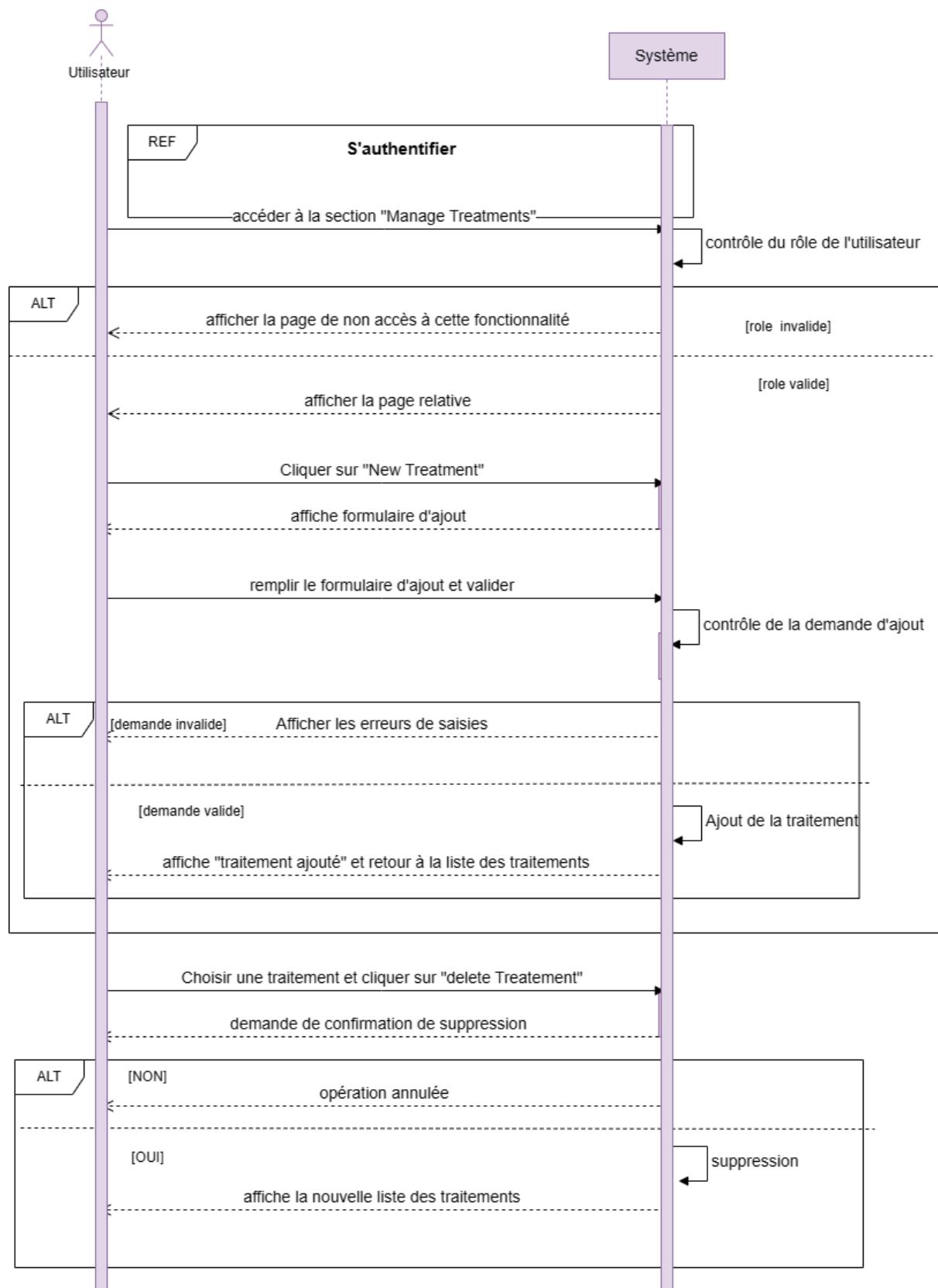


Figure 106: Diagramme de séquence d'ajout et de suppression de traitement.

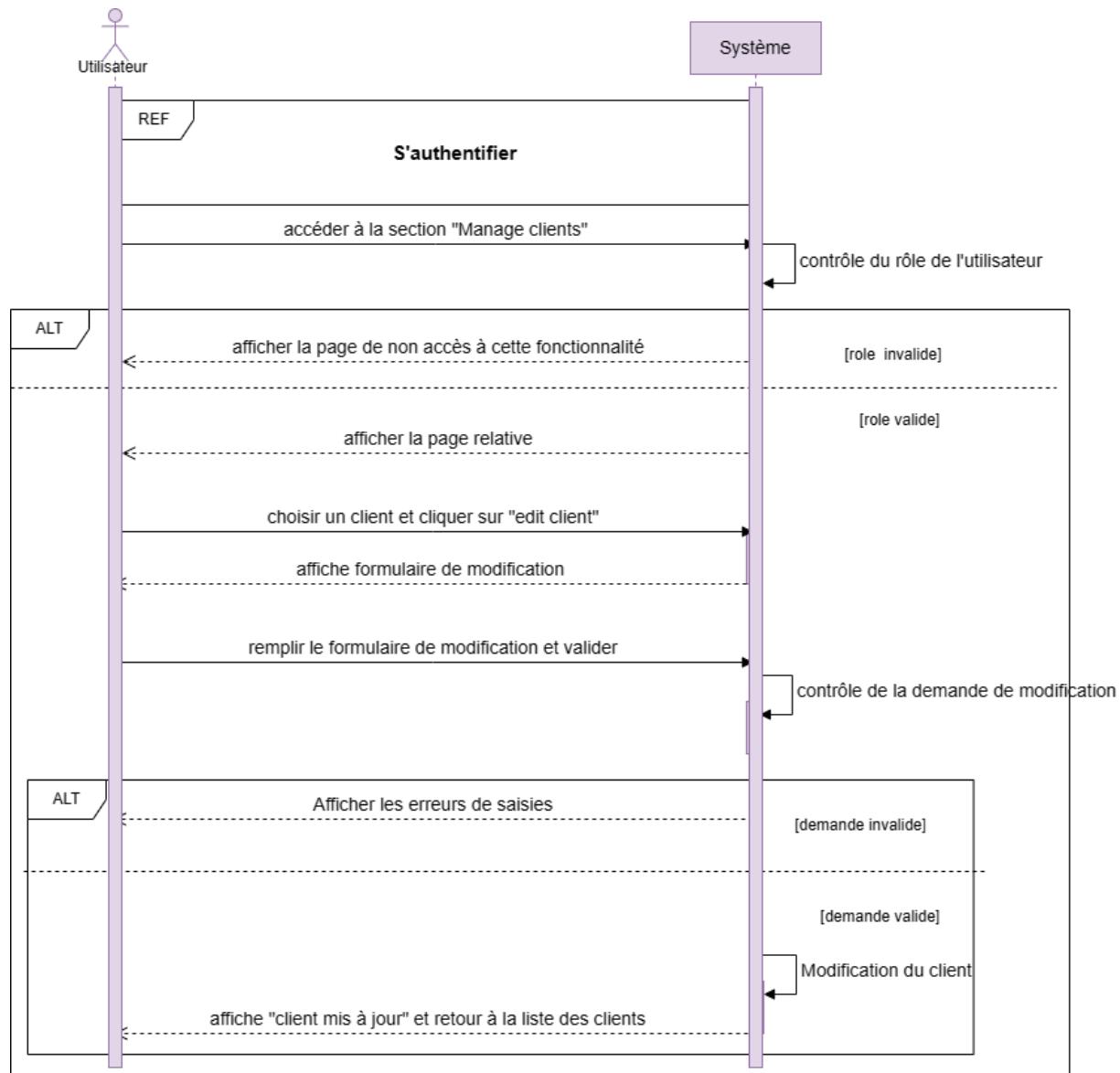


Figure 107: Diagramme de séquence de modification de client.

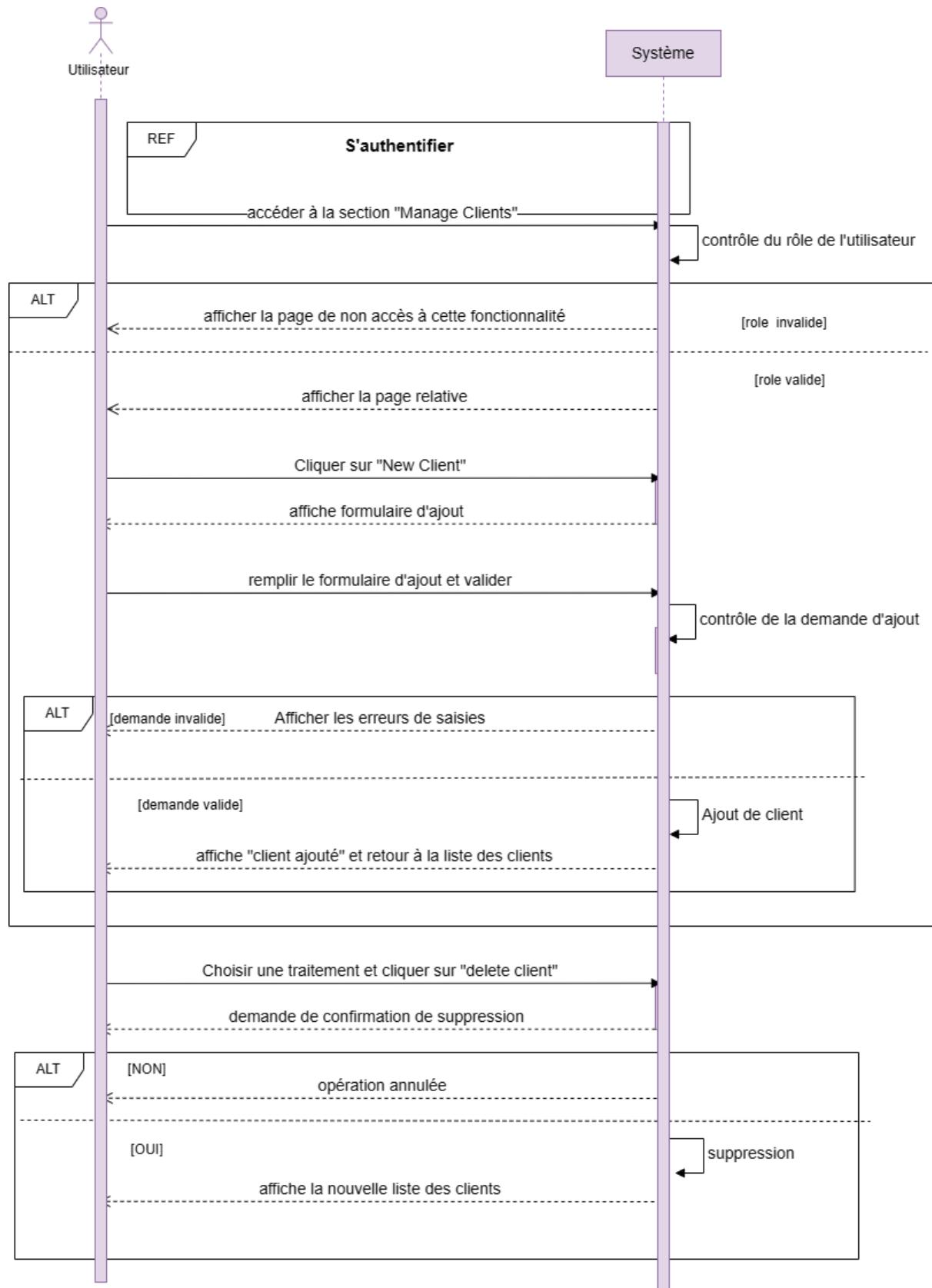


Figure 108: Diagramme de séquence d'ajout et de suppression de client.

5. Diagramme de classe

Suite à l'élaboration des descriptions précédentes, nous présentons dans la figure suivante le diagramme de classe relatif à cette phase.

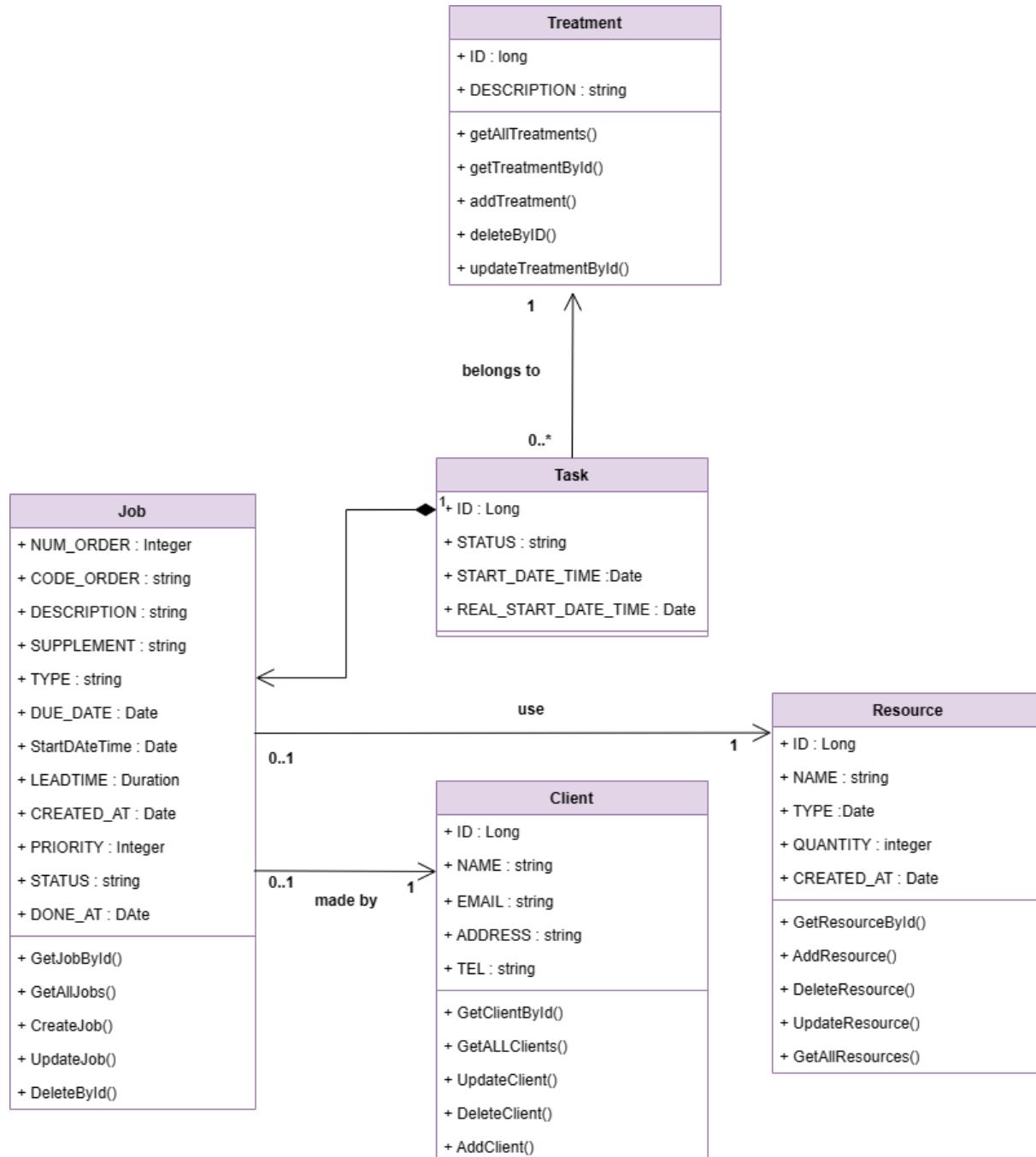


Figure 109:Diagramme de classe de la phase 03.

6. Réalisation

Dans cette partie, nous présentons les différentes réalisations effectuées au cours de cette phase.

Pour la gestion des phases de production, l'application dispose des interfaces permettant de réaliser les manipulations nécessaires. L'utilisateur accède à la page de gestion de la production et visualise l'ensemble des phases ordonnées de la chaîne de production.

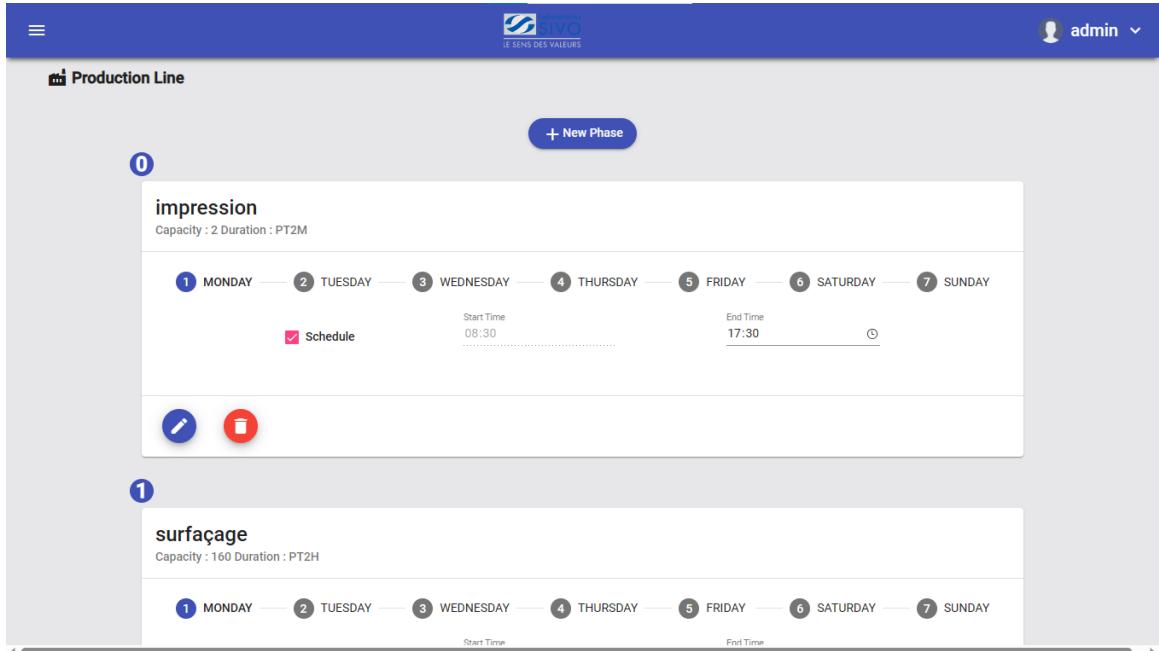


Figure 110:Interface de la gestion de la production.

L'utilisateur peut ainsi créer ou modifier une phase grâce au formulaire d'ajout et de modification de phase.

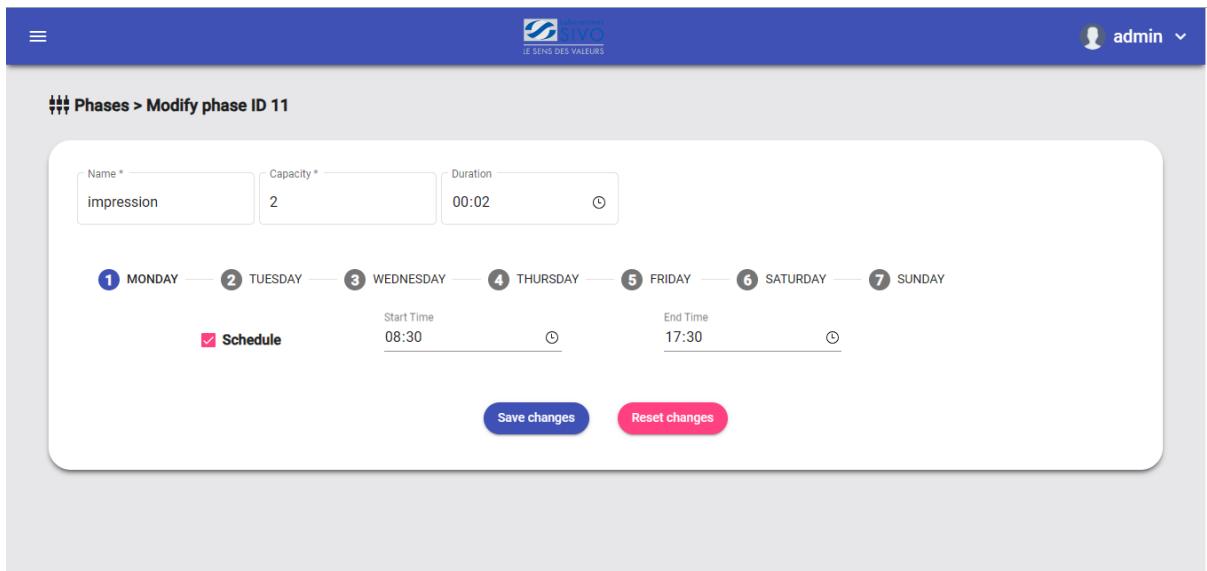
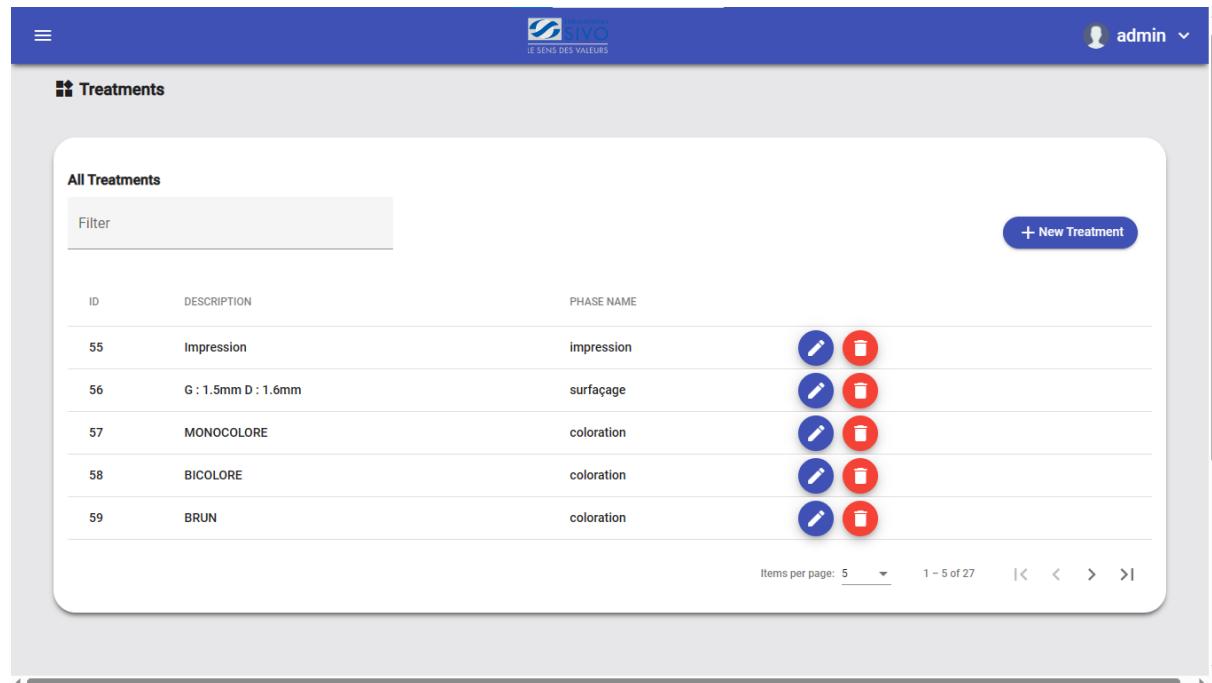


Figure 111:Interface du formulaire d'ajout et de modification de phase.

Pour la gestion des traitements et des clients, l'application fournit les interfaces nécessaires pour leur réalisation.



ID	DESCRIPTION	PHASE NAME
55	Impression	impression
56	G : 1.5mm D : 1.6mm	surfâge
57	MONOCOLORE	coloration
58	BICOLORE	coloration
59	BRUN	coloration



NOM	EMAIL	TEL
Syvets Ndiaye	anesyveets@gmail.com	+216 52000008

Figure 112:Interface de la gestion des traitements.

7. Conclusion

Dans ce chapitre, il était question de présenter les différentes conceptions élaborées pour réaliser la phase 03 de notre planification, ainsi que les réalisations qui en découlent. Dans le prochain chapitre, nous allons nous intéresser à la prochaine phase de notre projet qui traite de la mise en place de la gestion des ordres de fabrication et des ressources en stock.

Chapitre 10 : Mise en place de la gestion des ordres de fabrication et du stock.

1. Introduction

Dans ce chapitre il sera question de mettre en exergue l'ensemble des conceptions élaborées et des réalisations effectuées qui concernent la phase 03 de notre planification qui est l'implémentation de la gestion des ordres de fabrication et du stock.

2. Description des exigences fonctionnelles

Nous décrivons les besoins fonctionnels de chaque fonctionnalité de cette phase qui sont : la gestion des ordres de fabrication/traitement et la gestion des stocks. Le tableau suivant présente les user stories de ces fonctionnalités.

Tableau 17:Tableau des user stories de la phase 03.

Fonctionnalité	User story	ID
Gestion des ordres de fabrication/traitement	<ul style="list-style-type: none"> En tant qu'utilisateur privilégié, je veux ajouter, modifier, supprimer ou consulter les ordres de fabrication/traitement. 	1
Gestion des stocks	<ul style="list-style-type: none"> En tant qu'utilisateur privilégié, je veux créer , supprimer, modifier et consulter les ressources en stocks. 	2

3. Diagramme des cas d'utilisation

3.1. Présentation du diagramme des cas d'utilisation

La figure suivante présente le diagramme de cas d'utilisation de la phase 03 du projet.

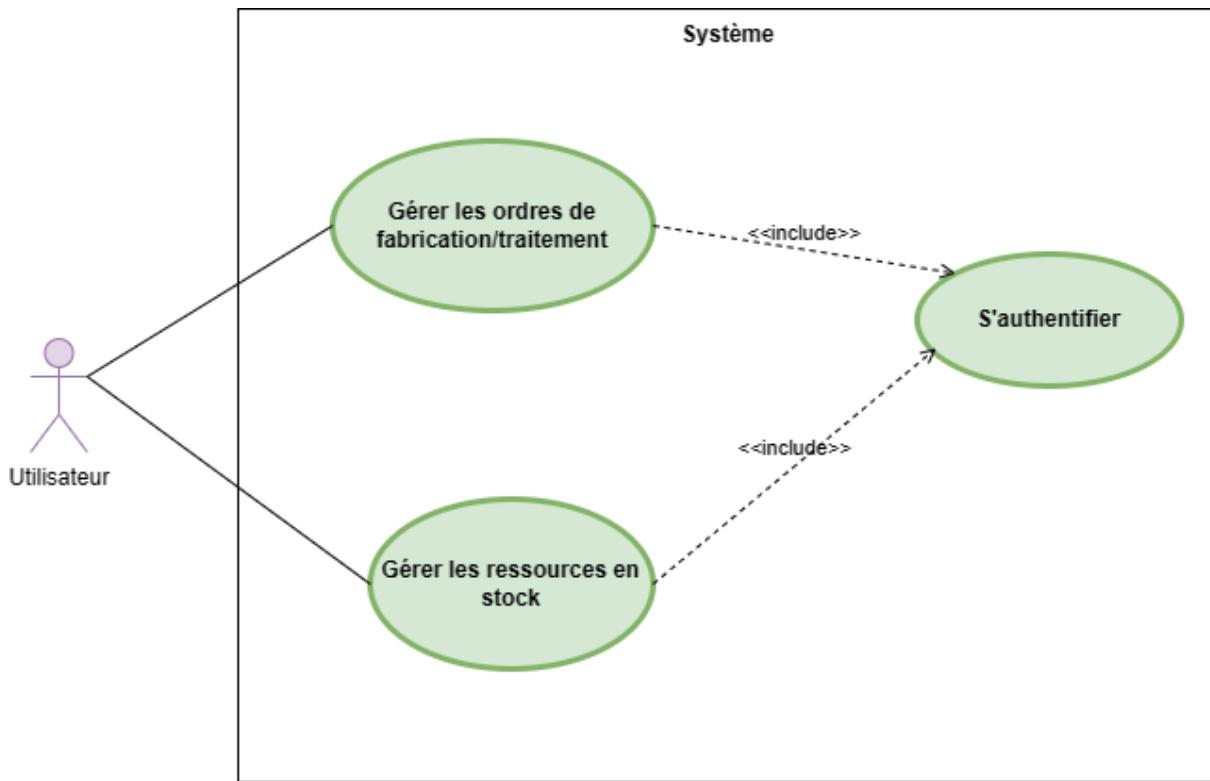


Figure 114: Diagramme de cas d'utilisation de la phase 03.

3.2. Description textuelle des cas d'utilisation

Dans cette section, nous présentons les différents scénarios d'utilisation de chaque sous-cas d'utilisation de la phase 03 de notre planification.

Le tableau 5.2 présente la description textuelle du sous-cas de gestion des ordres de fabrication.

Tableau 18: Description textuelle du sous-cas « gérer les ordres de fabrication/traitement »

Titre	Gérer les ordres de fabrication/traitement
Acteur	Utilisateur
Précondition	L'utilisateur s'authentifie
Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur « Manage Ordres » dans le menu de navigation. 2. Le système charge la page relative à la gestion ordres de fabrication. 3. L'utilisateur peut créer, modifier à l'aide de formulaires, supprimer et consulter des ordres de fabrication.
Exception	<ol style="list-style-type: none"> 1. Si l'utilisateur ne dispose pas des permissions nécessaires, le système redirige vers la page de non accès.

Le prochain tableau décrit celui du sous-cas d'utilisation « gérer les ressources en stock »

Tableau 19: Description textuelle du sous-cas « gérer les ressources en stocks »

Titre	Gérer les ressources en stock
Acteur	Utilisateur
Précondition	L'utilisateur s'authentifie
Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur « Manage Stocks » dans le menu de navigation. 2. Le système charge la page relative à la gestion des stocks. 3. L'utilisateur peut créer, modifier, supprimer et consulter les ressources en stock.
Exception	<ol style="list-style-type: none"> 1. Si l'utilisateur ne dispose pas des permissions nécessaires, le système redirige vers la page de non accès.

4. Diagrammes de séquence

Dans cette partie, nous présentons l'ensemble des diagrammes de séquence de chaque fonctionnalité de la phase 03de notre planification.

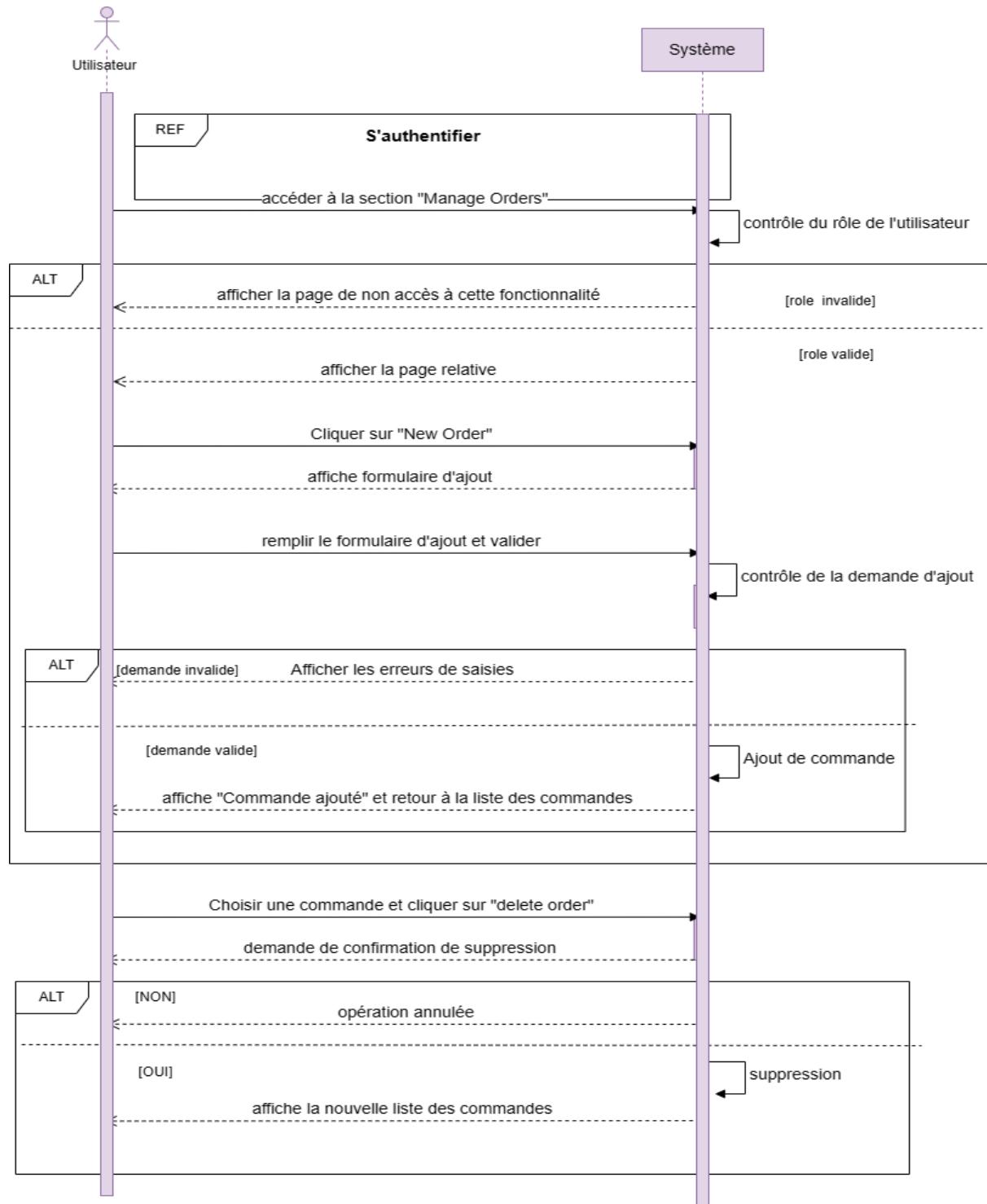


Figure 115:Diagramme de séquence d'ajout et de suppression d'un ordre de fabrication.

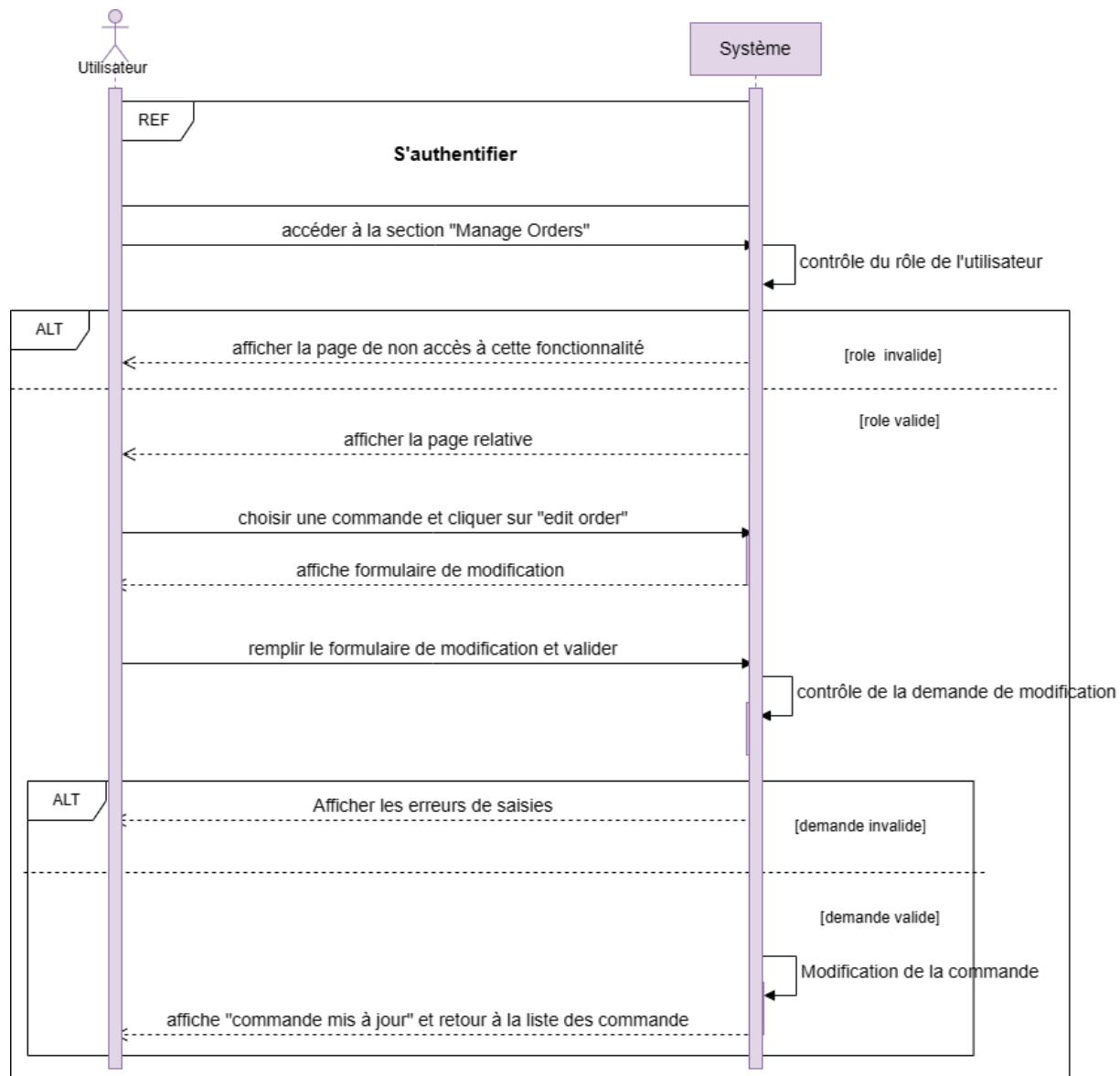


Figure 116: Diagramme de séquence de la modification d'un ordre de fabrication.

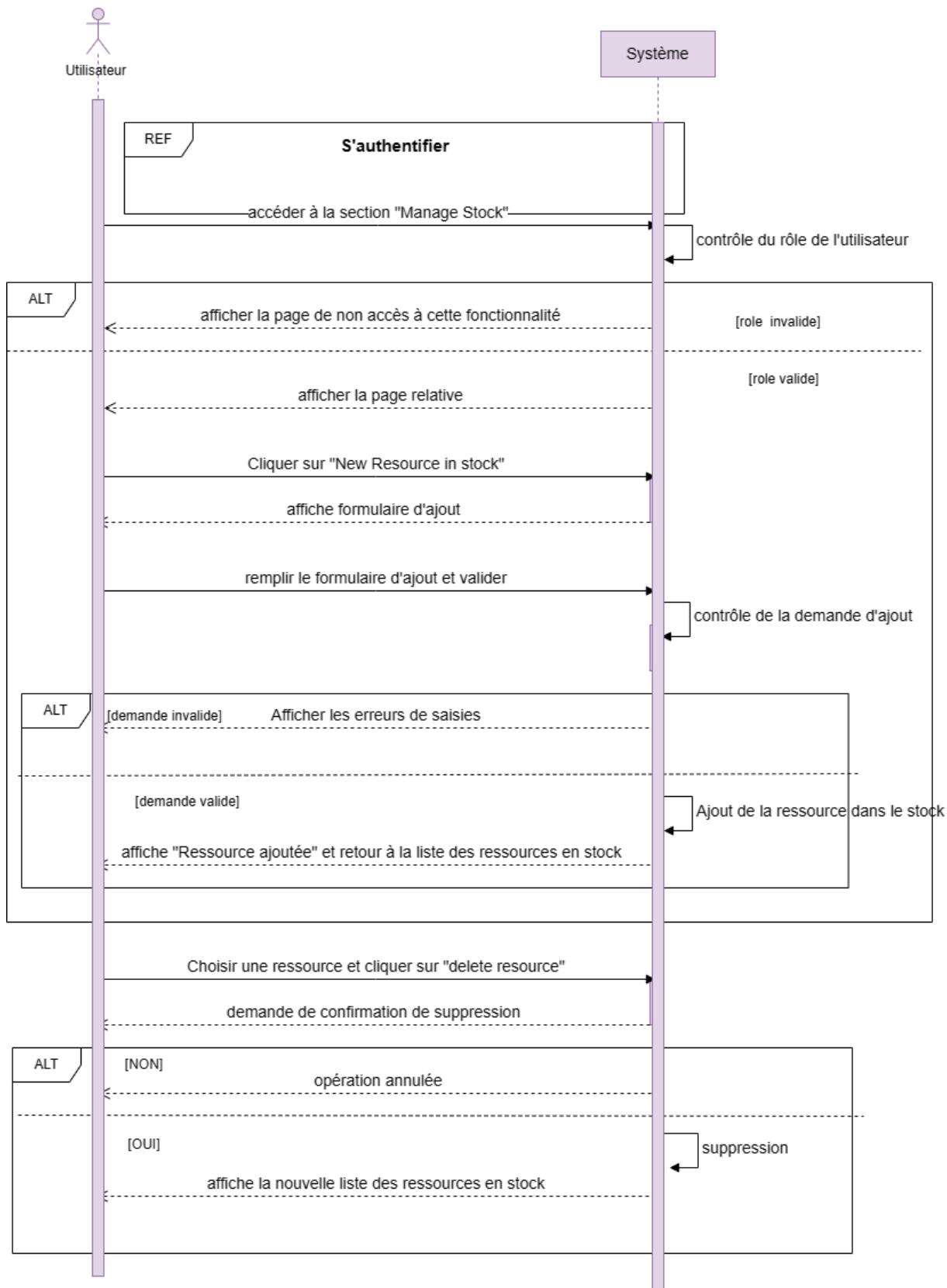


Figure 117: Diagramme de séquence d'ajout et de suppression d'une ressource.

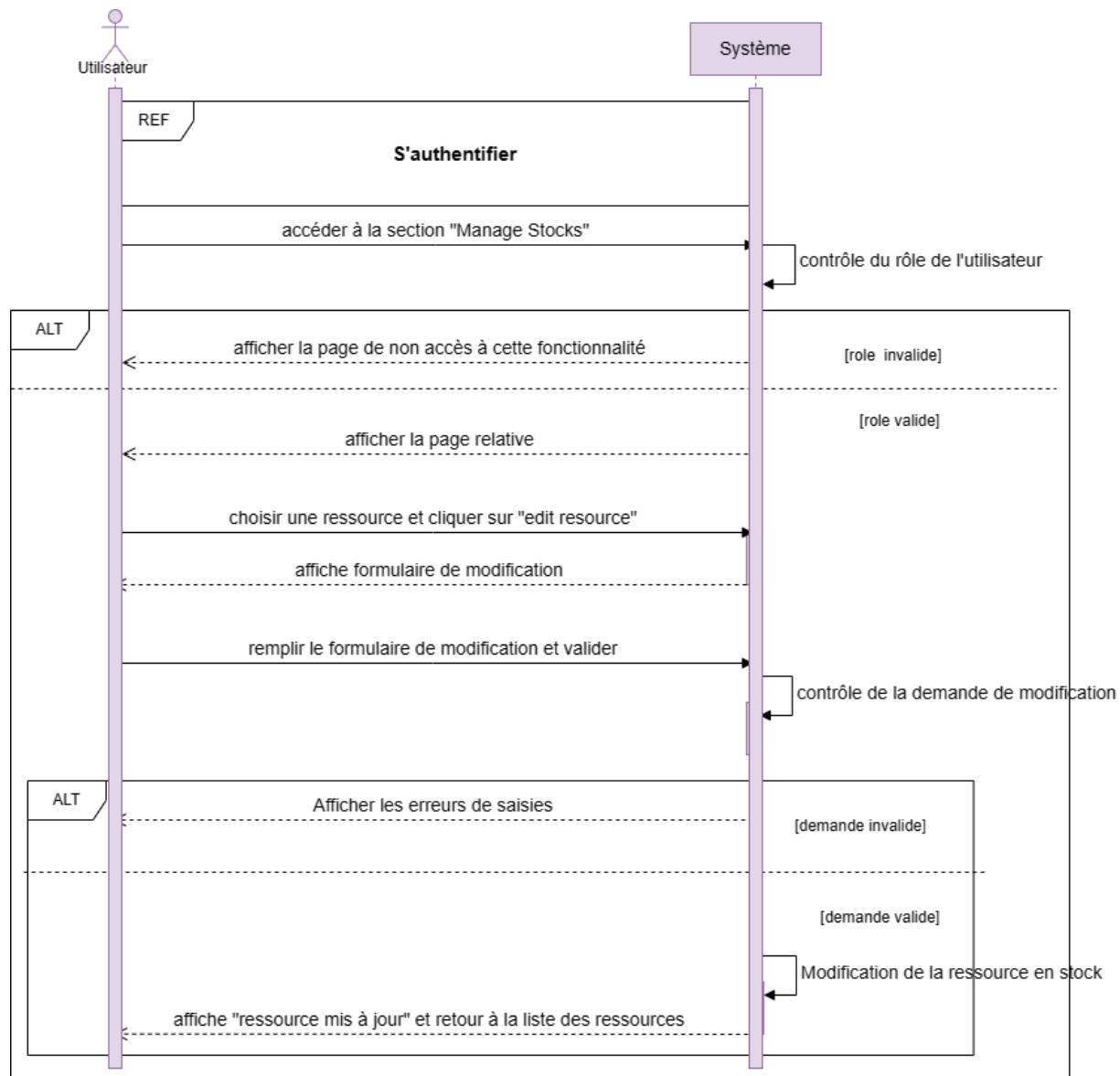


Figure 118: Diagramme de séquence de modification d'une ressource.

5. Diagramme de classe

Après la description des besoins et des scénarios d'utilisations de cette phase, nous déduisons le diagramme de classe de la phase 03 tel que représenté par la figure 5.6.

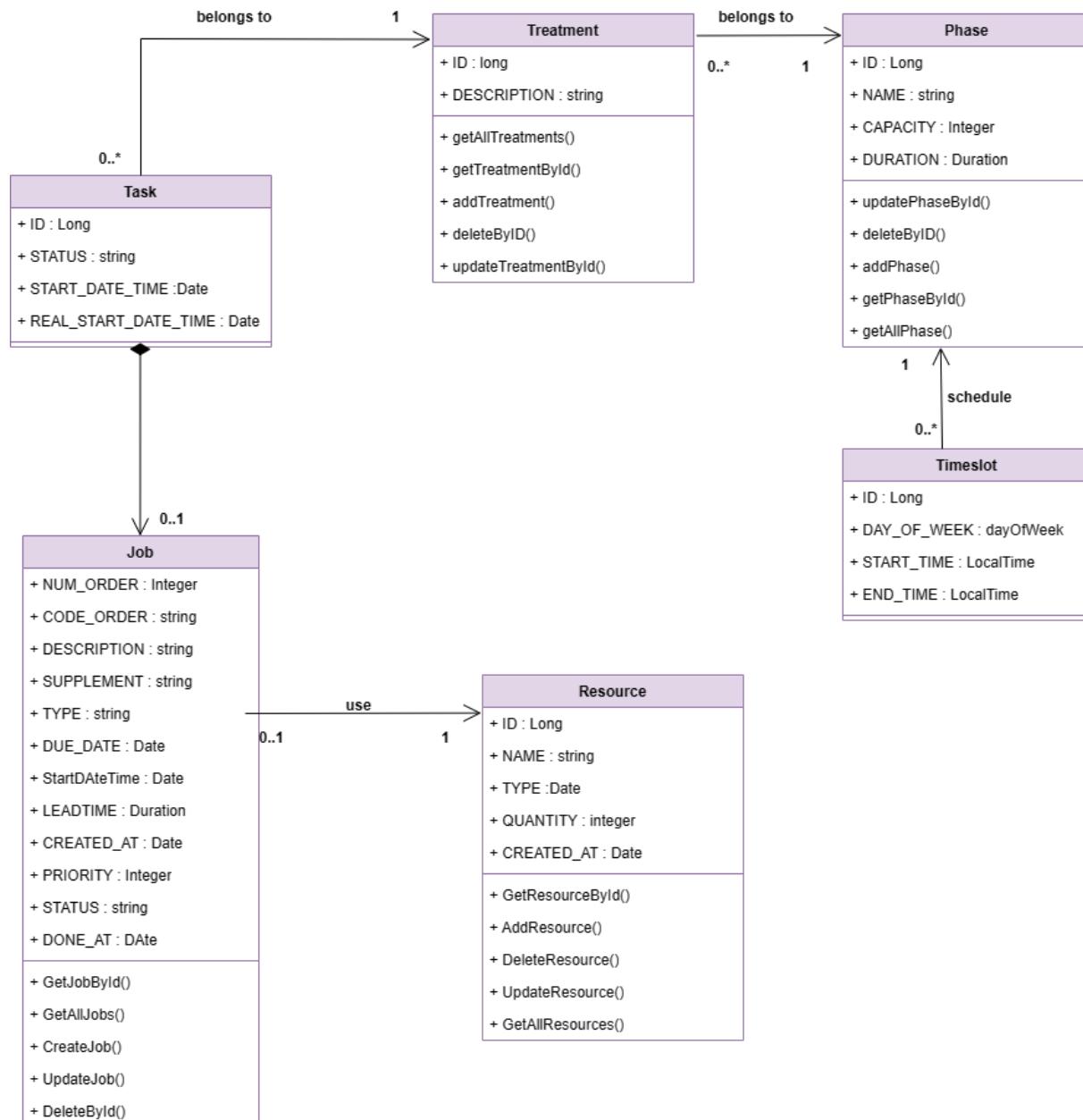


Figure 119;Diagramme de classe de la phase 03.

Les ordres de fabrications sont représentés par la classe d'entité **Job** et tous ses attributs. Chaque ordre possède une liste de tâches (**Task**) à effectuer pour le réaliser. D'autre part, chaque tâche appartient à un type de traitement en particulier dans une phase de la chaîne de production. Les traitements sont représentés par la classe d'entité **Treatment**. Ces derniers appartiennent eux aussi à une phase (**Phase**) précise de la chaîne de production.

Il est important de préciser que chaque phase est réalisée par une machine de la chaîne de production caractérisée par sa durée de traitement, sa capacité et ses horaires de disponibilité. D'où la nécessité de planifier chaque phase par une liste de créneaux horaire(**Timeslot**) de disponibilité.

6. Réalisation

Après avoir élaboré l'ensemble des conceptions que nous avons jugées nécessaires à la réalisation de cette phase, nous passons désormais à l'implémentation réelle de ces fonctionnalités.

Pour gérer les ordres de fabrication et de traitement, l'utilisateur doit disposer des priviléges nécessaires. Dans ce cas la page de gestion de commandes permet de consulter la liste des ordres de fabrication. Il peut ajouter un nouvel ordre en cliquant sur « New Order ». Il peut d'autre part visualiser les statistiques de réalisation de commandes pour chaque mois de l'année.



Figure 120:Interface de la page de gestion des ordres de fabrication.

En cliquant sur un ordre de fabrication, l'utilisateur peut consulter les détails ou le supprimer.

N° ORDER	CLIENT	CREATED DATE	DU DATE	STATUS
6	sena steevy	January 1st 2023, 6:10:03 pm	January 2nd 2023, 6:35:26 pm	UNDONE
7	sena steevy	January 1st 2023, 6:10:04 pm	January 2nd 2023, 6:35:04 pm	UNDONE

Figure 121:Interface des détails d'un ordre de fabrication.

L'utilisateur peut ensuite le modifier s'il le souhaite grâce à l'interface de modification de commande présentée par la figure 5.9.

Figure 122:Interface de modification.

Pour ce qui est des ressources en stocks, les interfaces d'ajout, de modification et de gestion des ressources permettent d'assurer l'ensemble de ces fonctionnalités.

The screenshot shows the SIVO software interface. At the top, there is a blue header bar with the SIVO logo and a user profile for 'admin'. Below the header, the title 'Clients' is displayed. A sub-section titled 'All Resources' is shown. The table lists two resources:

ID	NAME	TYPE	IN STOCK	LAST INCOME DATE
5	paire de verres	simple	1480	2023-07-23T16:26:35
6	paire de verres	semi-finie	525	2023-07-25T01:17:18

Below the table are two red circular icons with edit and delete symbols. At the bottom of the screen, there are navigation buttons for 'Items per page' (set to 10), '1 - 2 of 2', and arrows for navigating through the pages.

Figure 123:Interface de gestion des ressources.

The screenshot shows the SIVO software interface. At the top, there is a blue header bar with the SIVO logo and a user profile for 'admin'. Below the header, the title 'Resources > New Resource' is displayed. In the main area, there are three input fields: 'Name *' (verre A434), 'Type *' (semi-finie), and 'Quantity in Stock *' (200). A modal dialog box titled 'Save New Resource?' is open, asking 'Do you want to Add this resource ?' with 'No' and 'Yes' buttons. At the bottom of the screen, there is a navigation menu with links: Home, Scheduler, Users, Orders, Stocks, and Settings. The status bar at the bottom right shows 'Prés du record' and the date '25/07/2023'.

Figure 124:Interface d'ajout de ressources en stocks.

7. Conclusion

Parvenu au terme de ce chapitre, il était question d'étudier tout d'abord les besoins fonctionnels de la phase 3 de notre planification qui est la mise en place de la gestion des ordres de fabrication et de la gestion des ressources en stock. Ensuite nous avons présenté l'ensemble des conceptions élaborées pour terminer avec les réalisations effectuées. Dans le prochain chapitre, nous allons nous intéresser à la dernière phase de notre projet qui est la mise en place de l'algorithme de planification optimisée et de la gestion des plannings.

Chapitre 11 : Implémentation de l'algorithme de planification optimisée et gestion des plannings

1. Introduction

Ce chapitre est consacré à la phase de développement et d'intégration de l'algorithme d'ordonnancement optimisé, sans oublier l'historisation des plannings. Nous allons commencer par dégager les exigences fonctionnelles de cette phase. Par la suite nous présenterons les différentes conceptions mises sur pied pour la réalisation. Et Enfin nous allons essayer de montrer quelques réalisations au cours de cette phase.

2. Description des exigences fonctionnelles

Le tableau suivant présente la description des user stories des fonctionnalités qui concernent cette phase.

Tableau 20:user stories phase 04

Fonctionnalité	User stories	ID
Génération manuelle et automatique de plannings optimisés	<ul style="list-style-type: none"> • En tant qu'utilisateur responsable de la production, je veux générer manuellement des planifications intelligentes de la production. • En tant qu'utilisateur responsable de la production, je veux planifier des horaires de génération automatique de plannings . 	1
Historisation des plannings	<ul style="list-style-type: none"> • En tant qu'utilisateur, je veux consulter l'historique des plannings générés et visualiser leurs performances. 	3
Suivi de planning	<ul style="list-style-type: none"> • En tant qu'utilisateur contrôlant la production, je veux démarrer, stopper ou recommencer l'exécution des jobs suivant l'ordonnancement du planning en cours d'application. • En tant qu'utilisateur contrôlant la production, je veux suivre l'exécution des jobs du planning en cours d'application. 	4

3. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation illustré par la figure 3.1, représente les cas d'utilisation de la phase 02

3.1. Présentation du diagramme de cas d'utilisation

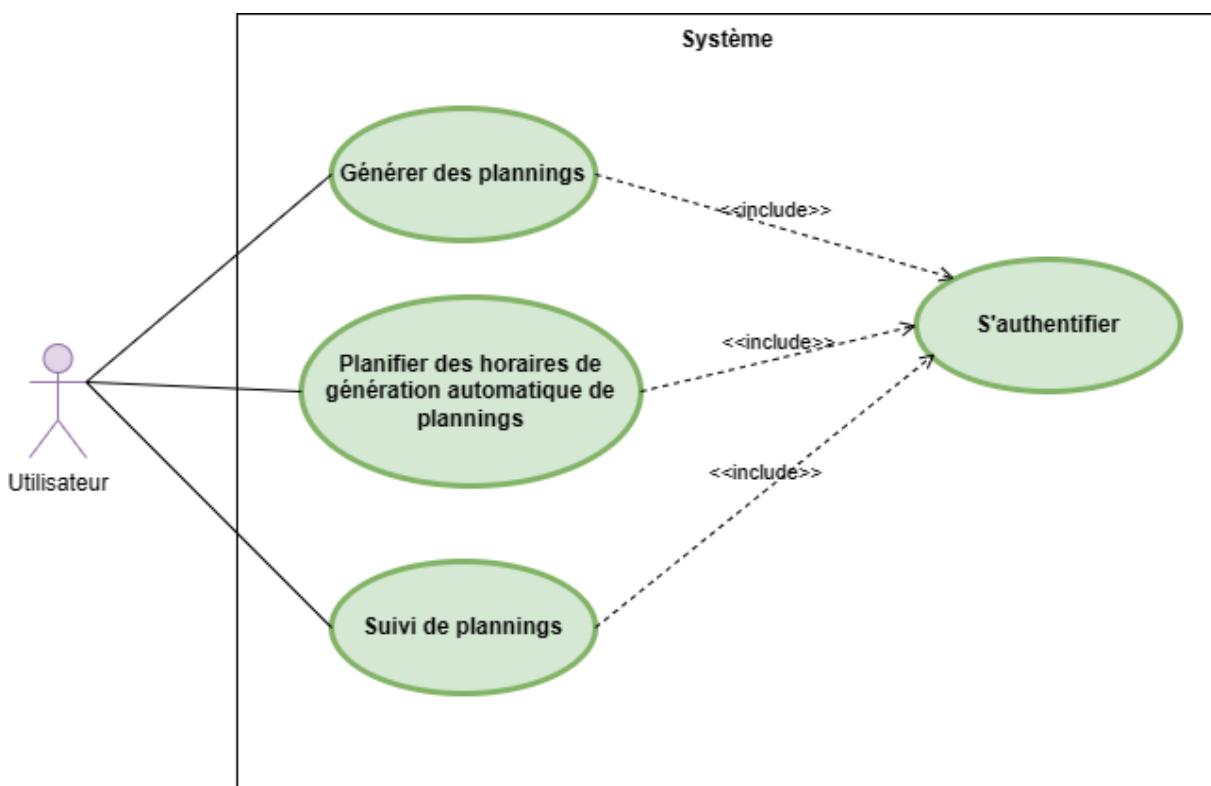


Figure 125:Diagramme de cas d'utilisation de la phase 04.

3.2. Description textuelle des sous-cas d'utilisation

Nous allons présenter dans cette partie la description textuelle de chaque sous-cas d'utilisation de cette phase à savoir : la génération de planning, la planification automatique de génération de plannings et le suivi de planning.

L'application permet de générer une nouvelle planification des commandes à réaliser. Ceci implique l'exécution de l'algorithme d'ordonnancement optimisé développé dans le cadre de ce projet afin de répondre aux besoins spécifiques de la société d'accueil où la solution sera déployée.

L'algorithme d'ordonnancement a été implanté en utilisant l'outil de planification par contraintes Java : **Optaplanner**. Le choix de cet outil se justifie par son caractère open source, sa légèreté et sa facilité d'intégration. Il est compatible avec la programmation

orientée objet (POO) et la saisie des contraintes se fait sous plusieurs formes (équations mathématiques, méthodes Stream, ...).

Pour assurer une meilleure planification, nous allons implémenter également un algorithme génétique afin d'optimiser les solutions issues de l'exécution d'Optaplanner. Ceci entraîne une augmentation importante du temps de résolution de l'algorithme, mais améliore considérablement les performances de la planification qui en résulte. Le tableau suivant présente la description textuelle du sous-cas d'utilisation de génération de planification.

Tableau 21:Description textuelle « Générer un planning »

Titre	Générer un nouveau planning
Acteur	Utilisateur
Précondition	L'utilisateur s'est authentifié sur un compte dont le rôle permet l'accès à cette fonctionnalité.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page de planification 2. L'utilisateur clique sur « Générer un nouveau planning » 3. Le système affiche un message de génération en cours 4. Le système affiche la nouvelle planification
Post-condition	Le système affiche la nouvelle liste des plannings sur la page

L'application permet aussi à l'utilisateur de programmer des générations automatiques de plannings à des horaires définis par celui-ci. Le tableau ci-après présente la description textuelle de ce sous-cas d'utilisation.

Tableau 22:description textuelle « Génération automatique de plannings »

Titre	Générer automatiquement des plannings
Acteur	Utilisateur
Précondition	L'utilisateur s'est authentifié sur un compte dont le rôle permet l'accès à cette fonctionnalité.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la section « horaire de planification auto » 2. L'utilisateur active la planification auto et choisit la cadence horaire 3. L'utilisateur clique sur « Appliquer » 4. Le système affiche un message de validation
Post-condition	2.1. Le système redirige vers la page de non accès à cette fonctionnalité.

Le suivi du planning permet à l'utilisateur de réaliser les commandes dans l'ordre de la planification. Selon la nature de son rôle, il peut démarrer, arrêter ou recommencer l'exécution des commandes dans la chaîne de production et suivre ainsi leur évolution. Le tableau 6.4 présente la description textuelle de ce sous-cas d'utilisation.

Tableau 23:Description textuelle « suivre le planning »

Titre	Suivre le planning
Acteur	Utilisateur privilégié
Précondition	L'utilisateur s'est authentifié sur un compte dont le rôle permet l'accès à cette fonctionnalité.
Scénario nominal	<ol style="list-style-type: none">1. L'utilisateur accède à la page « Plannings » et clique sur le dernier planning2. Le système affiche la courbe et les performances du planning sélectionné ainsi que la liste ordonner des tâches à réaliser.3. L'utilisateur peut suivre modifier l'état des tâches et suivre leur évolution
Exception	<ol style="list-style-type: none">1. S'il n'existe pas de planning pour la date sélectionnée, le système n'affiche aucun planning.2. Si l'utilisateur n'a pas la permission pour réaliser une opération, le système affiche un message de non accès pour cet utilisateur.

4. Diagrammes de séquences

Dans cette partie, nous allons présenter les différents diagrammes de séquence relatifs à cette phase de notre projet.

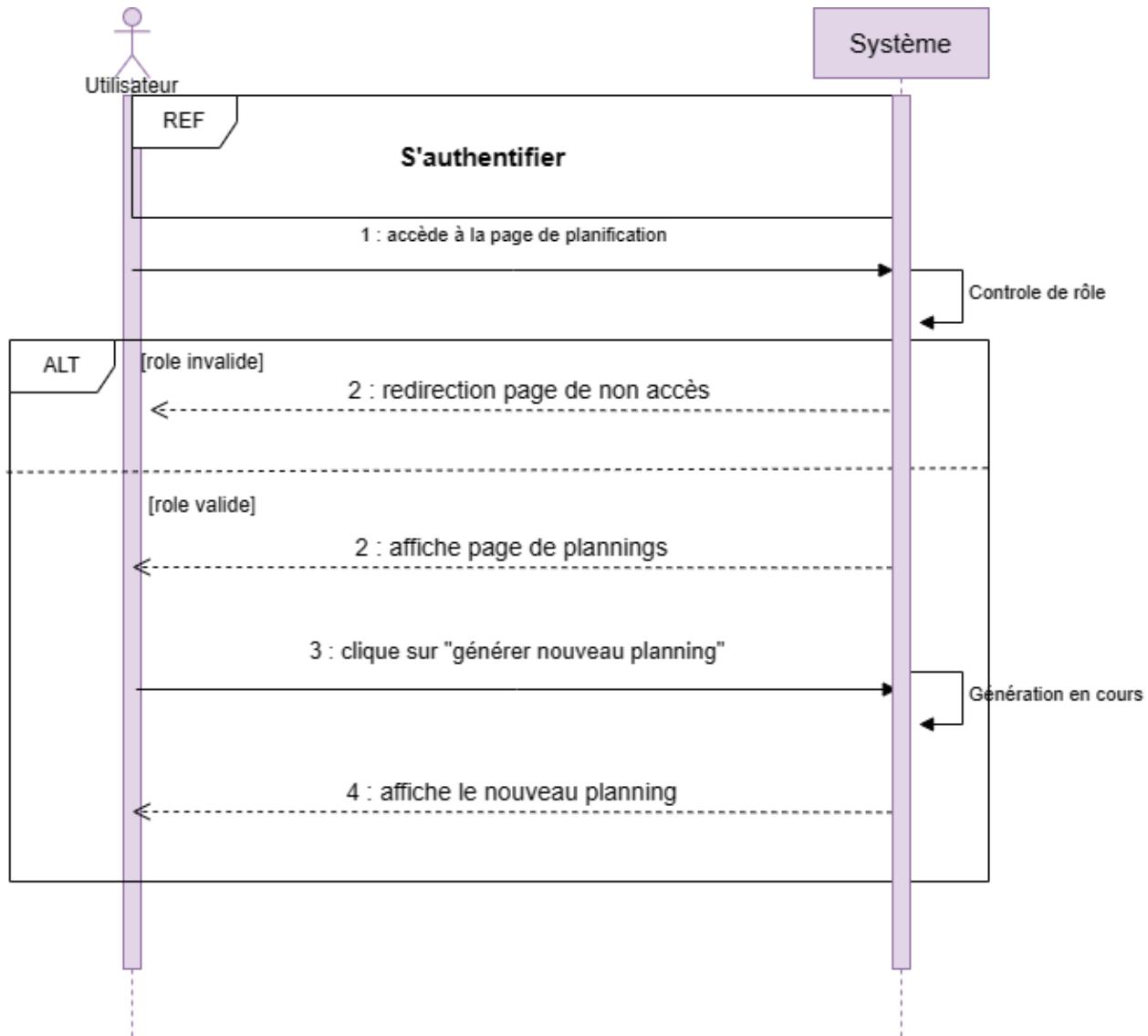


Figure 126:Diagramme de séquence du sous-cas « générer un planning »

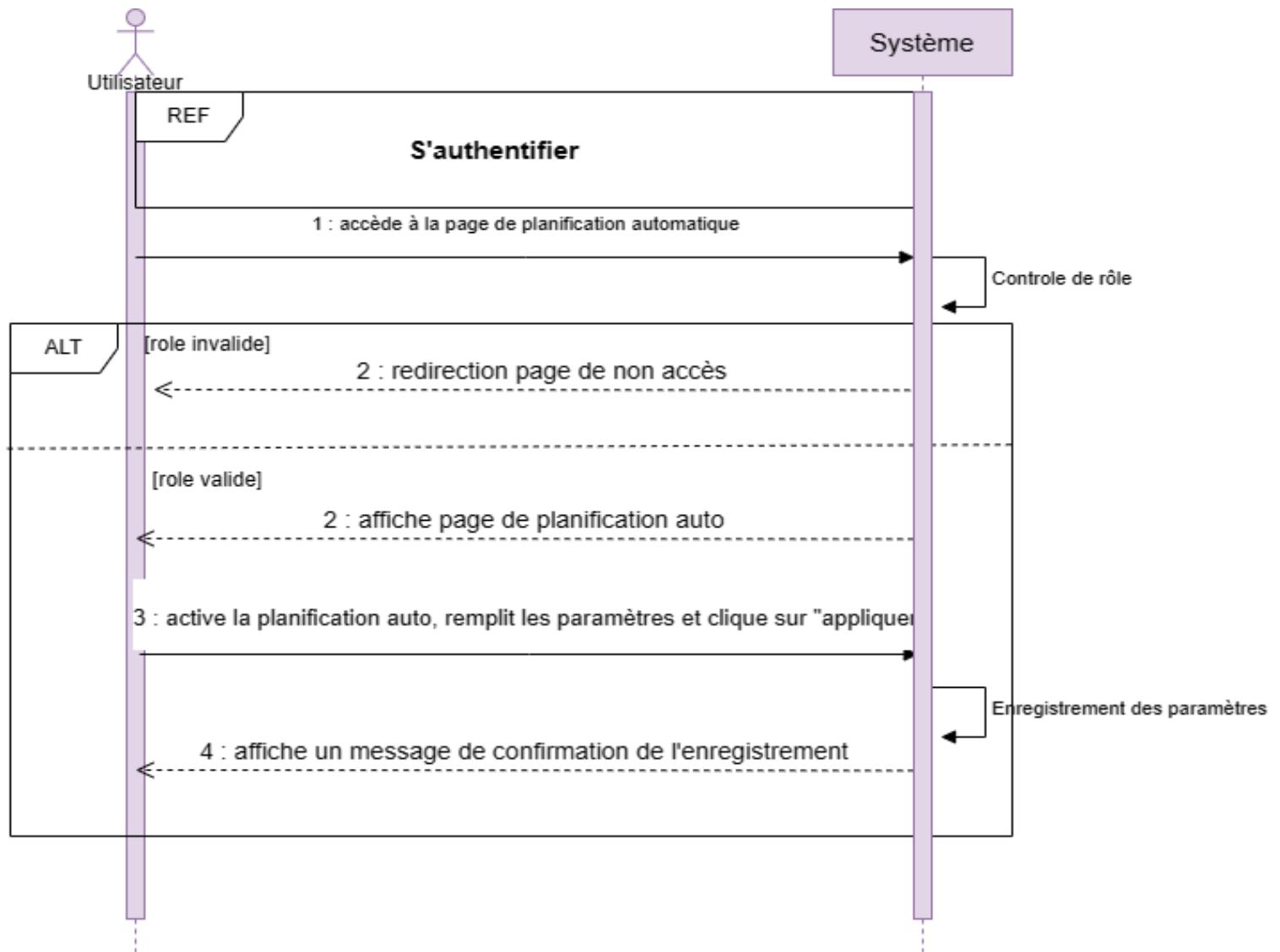


Figure 127:Diagramme de séquence du sous-cas « générer automatiquement des plannings »

Après cette description détaillée de la phase 04 de notre projet, nous passons désormais à l'élaboration du diagramme de classe qui en découle.

5. Diagramme de classe

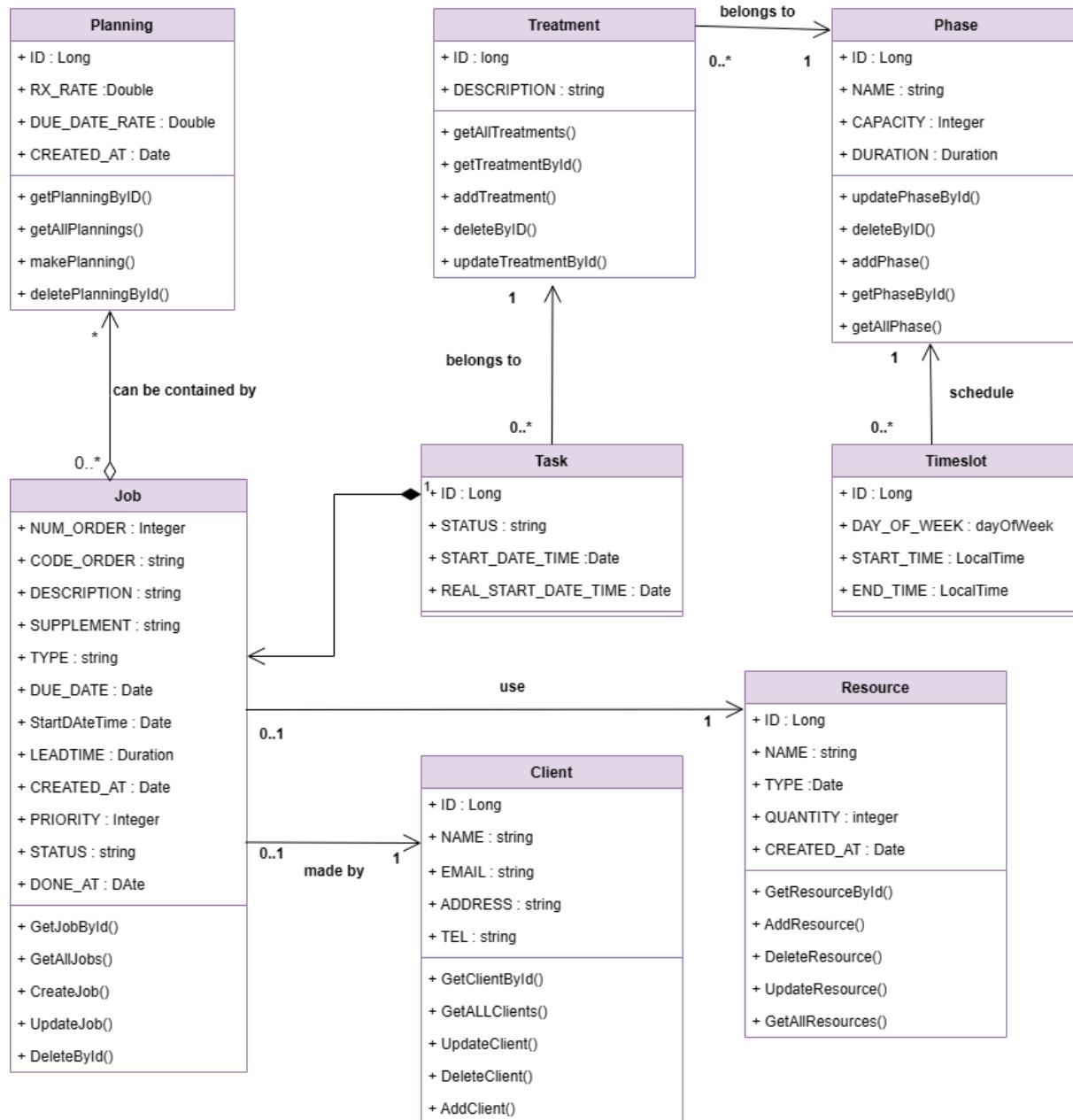


Figure 128:Diagramme de classe de la phase 04

6. Réalisation

Dans cette section, nous présentons l'essentiel des interfaces réalisées durant cette phase de notre planification ainsi que des échantillons de code réalisé.

Afin de permettre à l'utilisateur de visualiser, de générer et de suivre les plannings, nous avons développé l'interface des plannings telle que présentée par la figure suivante.

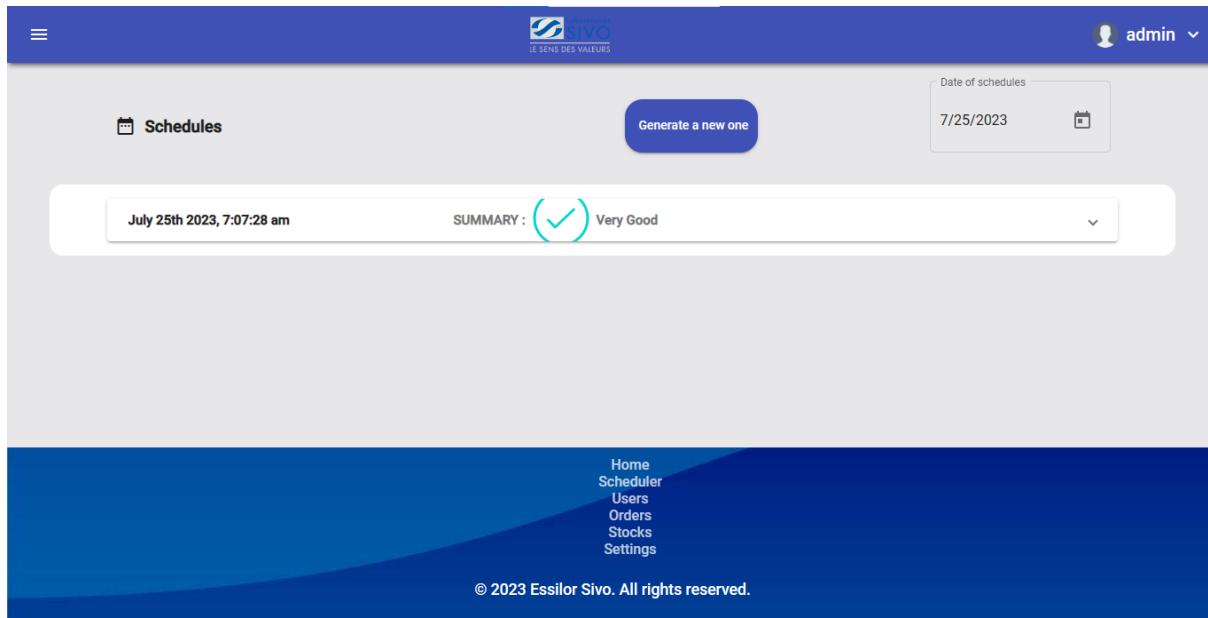


Figure 129:Interface de la page de planification

Ici, l'utilisateur peut générer un nouveau planning de façon manuelle s'il le souhaite. Une fois la génération terminée, le nouveau planning s'affiche dans la liste des plannings pour cette date de création.

L'utilisateur peut ainsi visualiser la courbe et les performances de l'ordonnancement issu de la planification choisie. Il peut également activer la planification automatique des commandes selon une cadence qu'il pourra définir. La figure qui suit présente l'interface développée pour assurer cette fonctionnalité.



Figure 130:Interface des détails d'une planification

L'utilisateur peut ainsi veiller à l'application du planning en modifier les états de chaque phase des commandes suivant l'ordonnancement du planning et suivre leur évolution.

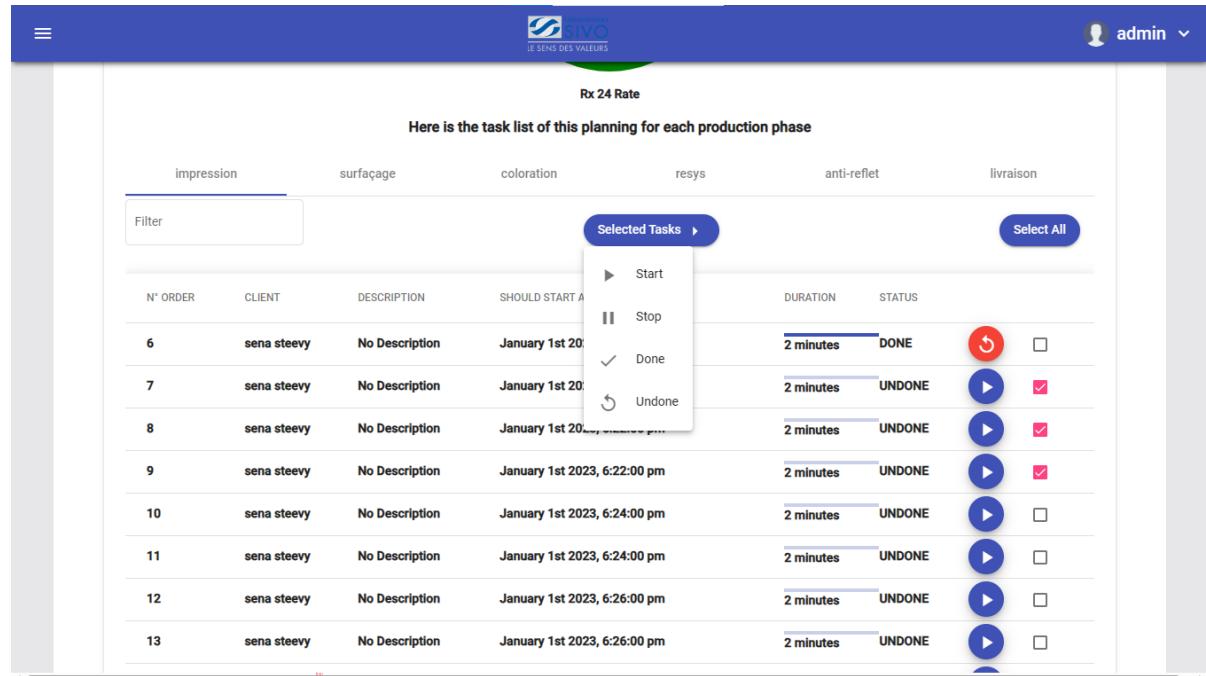


Figure 131:Interface du suivi du planning

La figure suivante est un extrait de l'implémentation des contraintes de l'algorithme d'ordonnancement. Ces contraintes permettent d'évaluer les solutions et de retourner les meilleures qui seront intégrées à l'algorithme génétique d'optimisation.

```
public class Constraints implements ConstraintProvider {  
  
    @Override  
    public Constraint[] defineConstraints(ConstraintFactory constraintFactory) {  
        return new Constraint [] {  
            maxCapacity(constraintFactory),  
            anteriorityTask(constraintFactory),  
            ResourceShouldBeFull(constraintFactory),  
            minimizeTimeGapBetweenTasks(constraintFactory)  
        };  
    }  
  
    private Constraint ResourceShouldBeFull(ConstraintFactory constraintFactory) {  
        return constraintFactory.forEach(Task.class)  
            .groupBy(Task::getStartTime, Task::getPhase, ConstraintCollectors.count())  
            .filter((startTime, phase, count) -> count <= phase.getCapacity())  
            .penalize(HardMediumSoftScore.ONE_MEDIUM, (startTime, phase, count) -> phase.getCapacity()  
            .asConstraint("Resource should be full"));  
    }  
}
```

Figure 132:implémentation des contraintes d'Optaplanner

7. Conclusion

Ce chapitre était consacré à la conception et à la réalisation des différentes fonctionnalités de la dernière phase de notre projet qui est : l'implémentation de l'algorithme de planification et la gestion des plannings. Au terme de celle-ci, nous arrivons à l'achèvement de la réalisation de notre application.

Conclusion générale

La réalisation de ce mémoire nous a permis de plonger dans l'univers de l'entreprise Essilor SIVO, de découvrir ses activités, ses produits et ses procédés de production. Nous avons également exploré les problèmes d'ordonnancement auxquels elle est confrontée et nous avons proposé une solution basée sur l'utilisation d'un algorithme génétique.

Dans le Chapitre 1, nous avons présenté en détail Essilor SIVO, en mettant l'accent sur sa structure hiérarchique, ses différents services et ses produits, notamment les verres organiques et minéraux. Cette présentation a été essentielle pour bien appréhender le contexte de l'entreprise et ses enjeux.

Le Chapitre 2 s'est concentré sur la gestion de la production. Nous avons défini la production, ses objectifs, et abordé les contraintes et les problèmes rencontrés dans les systèmes de production. Une gestion efficace de la production est primordiale pour optimiser les processus et les ressources de l'entreprise.

Dans le Chapitre 3, nous nous sommes intéressés aux problèmes d'ordonnancement, cruciaux dans un environnement industriel comme celui d'Essilor SIVO. Nous avons examiné différentes méthodes de résolution, des approches exactes aux méthodes approchées, en mettant l'accent sur l'algorithme génétique que nous avons utilisé pour résoudre le problème spécifique d'ordonnancement de l'atelier SIVO.

Le Chapitre 4 nous a permis de détailler le problème d'ordonnancement spécifique à l'atelier SIVO et de proposer une formulation claire du problème. En utilisant l'algorithme génétique, nous avons réussi à obtenir des résultats prometteurs pour améliorer l'efficacité de l'atelier.

Ensuite, dans le Chapitre 5, nous avons opté pour la simulation du nouveau système d'ordonnancement afin de vérifier la faisabilité de notre approche. Nous avons choisi le logiciel Arena pour construire notre modèle et nous l'avons validé en ajustant les paramètres pour refléter au mieux la réalité de l'atelier SIVO.

Le Chapitre 6 a été consacré à la spécification des besoins et à l'analyse en vue de la mise en place du nouveau système d'ordonnancement. Nous avons identifié les acteurs et défini les besoins fonctionnels et non fonctionnels, ainsi que les choix technologiques nécessaires à la mise en œuvre réussie de la solution.

Dans le Chapitre 7, nous avons expliqué la méthodologie adoptée pour la gestion de projet, en utilisant le Kanban comme principal outil de planification. Une approche méthodique est cruciale pour mener à bien un projet complexe comme celui-ci.

Les Chapitres 8, 9, 10 et 11 ont été consacrés à la modélisation et à la réalisation du système d'ordonnancement, de la gestion de la chaîne de production et des plannings, en utilisant les diagrammes de cas d'utilisation, de séquence et de classe. Ces chapitres ont détaillé les différentes étapes de la réalisation et ont abouti à une solution fonctionnelle et optimisée.

En conclusion, ce pfe a été l'occasion pour nous d'approfondir nos connaissances sur le fonctionnement d'une entreprise industrielle, ses problématiques de production et

d'ordonnancement. Nous avons pu mettre en pratique nos compétences en modélisation et en résolution de problèmes en utilisant des outils tels que l'algorithme génétique et la simulation. La solution proposée pour l'atelier SIVO devrait permettre d'améliorer significativement son efficacité opérationnelle, contribuant ainsi à renforcer la compétitivité globale d'Essilor SIVO sur le marché.

Nous espérons que ce pfe pourra apporter une contribution utile aux décideurs et aux professionnels du domaine de la production et de l'ordonnancement. Enfin, nous retenons de cette expérience la nécessité d'une approche méthodique et d'une bonne compréhension des enjeux spécifiques de chaque entreprise pour proposer des solutions adaptées et performantes.

Références bibliographiques

Bibliographie:

[1] Mohand Essaid, Modélisation et simulation de la connectivité des flux logistiques dans les réseaux manufacturiers, doctorat, génie industriel, 17 novembre 2008, l'Ecole Nationale Supérieure des Mines de Saint-Etienne.

[2]: Francisco E. Martínez, Application of SIMAN ARENA Discrete Event Simulation Tool in the Operational Planning of a Rail System, rapport d'étude, 31 may 2001, University of Puerto Rico Mayagüez Campus, Departement de genie Civil.

[3]: Fontanili F, Castagna P et Yannou B, La simulation pour la gestion des chaînes logistiques, Paris, Hermes Sciences, Lavoisier, ISBN 9 78-2-7462- 1843-7.

[4]Arena publication, GETTING STARTED WITH ARENA, Janvier 2012,PN-111648. 5 : Jean-Louis Boimond, SIMULATION -SYSTÈMES DE PRODUCTION -RÉSEAUX DE PETRI SIMAN – ARENA, cours.

Web graphie :

[1]www.sivo.com, (Dernier accès juillet 2023).

[2] [Diagramme sans nom - draw.io \(diagrams.net\)](https://draw.io). (Dernier accès juillet 2023).

[3] [Angular Material UI component library](https://material.angular.io/components/component-library). (Dernier accès juillet 2023).

[4] <https://trello.com>. (Dernier accès juillet 2023).

[5] <https://www.atlassian.com/fr/agile/kanban>. (Dernier accès juin 2023).

[6] <https://www.freepik.com/>. (Dernier accès Juillet 2023).

[7] <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>. (Dernier accès Avril 2023).

[8] <https://angular.io/docs>. (Dernier accès Avril 2023).

[9]<https://www.angularjswiki.com/angular/angular-material-icons-list-mat-icon-list/#mat-icon-list-category-navigation>. (Dernier accès juillet 2023).

