

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahman Mira - Béjaia -
Faculté des Sciences Exactes
Département Informatique



Mémoire de fin de cycle en vue de l'obtention d'un Master professionnel
Option : Génie Logiciel

Thème :

Conception et réalisation d'un progiciel de Gestion de
Production Assistée par Ordinateur
(la GPAO)

Réalisé par :

M^{elle} KERKAR Hanane

M^{elle} MEDDOUR Rima

Évalué devant le jury composé de :

Examinatrice 01 :	M ^{me} EL BOUHISSI H.	M.C.A	U.A/Mira Béjaia, Algérie.
Examinatrice 02 :	M ^{me} ZIDANI F.	M.C.B	U.A/Mira Béjaia, Algérie.
Encadrant :	M ^{me} YESSAD S.	M.C.B	U.A/Mira Béjaia, Algérie.

Promotion : 2019 - 2020

Dédicaces

*Je dédicace ce mémoire
En premier lieu
A
Ma chère mère,
Qui a été,
Et qui est toujours le pilier de ma réussite
A
Mon père,
Qui m'a toujours soutenu tout au long de mon parcours scolaire,
A
Ma sœur bien aimée,
A
Mes frères,
A
Mes amies et amies.*

Hanane KERKAR

*Je dédie ce modeste travail
A
Mes chers parents,
A
Mon père,
Que dieu ait pitié de lui,
A
Ma mère,
Que dieu la garde en bonne santé,
Qui m'ont toujours encouragé,
Soutenu que ce soit de loin ou de près et sacrifié
Pour devenir qui je suis
A
Ma famille,
Ma sœur,
Mes frères,
Mes amis et amies,
Toute personne qui m'a apporté de l'aide.*

Rima MEDDOUR

TABLE DES MATIÈRES

TABLE DES MATIERES	I
LISTE DES FIGURES	IV
LISTE DES TABLEAUX	V
LISTE DES ABREVIATIONS	VI
INTRODUCTION GÉNÉRALE	1

Chapitre 1 : L'univers de GPAO

1.1. Introduction	4
1.2. Présentation du produit	4
1.3. Gestion de production	4
1.4. Gestion de Production assistée par ordinateur	5
1.5. Les modules d'un GPAO	6
1.5.1. Gestion des données techniques	7
1.5.2. Un système de gestion de la planification	8
1.5.3. Système de gestion des achats	10
1.5.4. Système de gestion des stocks	10
1.5.5. Gestion des coûts	11
1.5.6. Système d'intégration CIM	11
1.6. Quels avantages à utiliser pour un logiciel GPAO	11
1.7. Les GPAO existants sur le marché	12
1.7.1. SYLOB	12
1.7.2. CLIPPER, Le leader français des progiciels GPAO	12
1.7.3. Sage 100CloudGestion de production	12
1.8. Conclusion	13

Chapitre 2 : Contexte du projet

2.1. Introduction	15
2.2. Présentation du sujet	15
2.3. Présentation du système de gestion de production assistée par ordinateur	15
2.4. Cycle de vie du projet	16
2.4.1. Définition d'un cycle de vie	16

2.4.2. Les étapes du cycle de vie utilisé	16
2.5. Qu'est-ce que UP ?	17
2.5.1. Les principes d'UP	18
2.5.2. Pourquoi UP	19
2.5.3. Le langage de modélisation utilisé par UP	19
2.6. Conclusion	19

Chapitre 3 : Spécification et analyse des besoins

3.1. Introduction	21
3.2. Spécification des besoins	21
3.2.1. Les besoins fonctionnels	21
3.2.2. Les besoins non fonctionnels	22
3.3. Analyse des besoins	22
3.3.1. Identification des acteurs	22
3.3.2. Modélisation du contexte	23
3.3.3. Identification des cas d'utilisation	24
3.3.4. Diagrammes des cas d'utilisation par acteur et leurs descriptions	25
3.4. Conclusion	32

Chapitre 4 : Conception

4.1. Introduction	34
4.2. Diagrammes de séquences des cas d'utilisation de chaque utilisateur	34
4.2.1. Diagramme de séquence associé à tous les utilisateurs	34
4.2.2. Diagramme de séquence associé à l'administrateur	36
4.2.3. Diagramme de séquence associé au « producteur »	48
4.2.4. Diagramme de séquence associé à « Inventaire de production »	40
4.2.5. Diagramme de séquence associé au « Planificateur »	41
4.3. Patrons de conception (Design Pattern)	43
4.3.1. Patron client/serveur à 3-tiers	43
4.3.2. Patron MVT (Modèle-vue- template)	44
4.4. Diagramme de classe	45
4.5. Conclusion	47

Chapitre 5 : Implémentation et Réalisation

5.1. Introduction	51
5.2. Outils de développement	51
5.2.1. Environnement de développement	51
5.2.2. Framework	53
5.2.3. Langages de développement	54
5.2.4. Bibliothèque	55
5.2.5. package NumPy	56
5.2.6. Module	56
5.3. Présentation de l'application	57
5.3.1. Présentation du Logo	57
5.3.2. Présentation des interfaces	57
5.4. Conclusion	61
CONCLUSION GÉNÉRALE	62
RÉFÉRENCE BIBLIOGRAPHIQUE	63

TABLE DES FIGURES

Figure 1.1 : Les modules du GPAO.....	7
Figure 1.2 : Schéma expliquant le plan industriel et commercial.....	8
Figure 1.3 : Schéma expliquant le Plan Directeur de Production.....	9
Figure 1.4 : Schéma expliquant les étapes de pilotage de l'exécution.....	10
Figure 2.1 : Le cycle de vie de notre projet.....	17
Figure 3.1 : Diagramme de contexte.....	24
Figure 3.2 : Diagramme de cas d'utilisation globale.....	25
Figure 3.3 : Diagramme des cas d'utilisations de l'administrateur.....	26
Figure 3.4 : Diagramme de cas d'utilisations de l'inventaire de production.....	27
Figure 3.5 : Diagramme de cas d'utilisations de producteur.....	28
Figure 3.6 : Diagramme des cas d'utilisations du planificateur.....	29
Figure 4.1 : Diagramme de séquence du cas d'utilisation « authentification ».....	35
Figure 4.2 : Diagramme de séquence du cas d'utilisation « Consulter le suivi de fabrication ».....	36
Figure 4.3 : Diagramme de séquence associé à l'« administrateur ».....	38
Figure 4.4 : Diagramme de séquence associé au « producteur ».....	39
Figure 4.5 : Diagramme de séquence associé à « Inventaire de production ».....	40
Figure 4.6 : Diagramme de séquence associé au « planificateur ».....	43
Figure 4.7 : Architecture 3-tiers.....	44
Figure 4.8 : Patron MVT (Modèle, Vue, Template).....	44
Figure 4.9 : Diagramme de classe.....	46
Figure 5.1 : Structure d'un serveur WEB.....	52
Figure 5.2 : Relation d'un mappeur objet-relationnel	56
Figure 5.3 : Logo de notre application.....	57
Figure 5.4 : Interface d'authentification.....	57
Figure 5.5 : Interface de TAKARAGPAO	58
Figure 5.6 : Interface de l'administrateur	59
Figure 5.7 : Interface de création d'un nouvel utilisateur	60

LISTE DES TABLEAUX

Tableau 3.1 : Les fonctionnalités des acteurs de notre système.....	23
Tableau 3.2 : Message échangés entre les acteurs et le système.....	23
Tableau 3.3 : Description du cas d'utilisation « Gérer les comptes ».....	26
Tableau 3.4 : Description du cas d'utilisation « Gérer le stock ».....	27
Tableau 3.5 : Description du cas d'utilisation « Générer les rapports ».....	28
Tableau 3.6 : Description du cas d'utilisation « lancer les ordres de fabrication ».....	29
Tableau 3.7 : Description du cas d'utilisation « Gérer les articles ».....	30
Tableau 3.8 : Description du cas d'utilisation « Gérer des ordres de fabrication ».....	30
Tableau 3.9 : Description du cas d'utilisation « s'authentifier ».....	31
Tableau 3.10 : Description du cas d'utilisation « Consulter le suivi de fabrication »....	32

LISTE DES ABRÉVIATIONS

CIM	Computer Integrated Management.
CSS	Cascading Style Sheets.
GP	Gestion de la Production.
GPAO	Gestion de la Production Assistée par Ordinateur.
HTML	HyperText Markup Language.
http	Hypertext Transfer Protocol.
MRP	Material Requirements Planning.
MVT	Modèle, Vue, Templates.
ORM	Mappeur Objet-Relationnel.
PDP	Plan Directeur de Production.
PHP	PHP Hypertext Preprocessor.
PIC	Plan Industriel et Commercial.
PIP	Pip Installs Packages.
SGBD	Système de Gestion de Base de Données.
SGBDR	Système de Gestion de Bases de Données Relationnelles.
SGML	Standard Generalized Markup Language.
SQL	Structured Query Language.
Ts	Temps de changement de Série.
Tu	Temps de réalisation.
UML	Unified Modeling Language.
UP	Unified Process.
Web	World wide web.
Workflow	Gestion du Flux de Travail.
WSGI	Web Server Gateway Interface.

INTRODUCTION GÉNÉRALE

En ces derniers temps l'informatique occupe de plus en plus une très grande place dans la vie quotidienne, ce domaine est devenu l'outil indispensable de l'homme. En effet, le stockage et le traitement des données sont automatisés dans plusieurs domaines d'applications tels que les administrations, le commerce, mais aussi dans le champ des entreprises quel que soit leurs types.

Pour répondre à ces différents besoins, l'industrie du logiciel a développé un certain nombre d'outils informatiques permettant de mieux gérer la production sous ses divers aspects, par exemple le progiciel de gestion de production assistée par ordinateur où plus brièvement le progiciel GPAO.

Donc, le progiciel GPAO est l'outil qui rend possible la gestion précise de l'ensemble des activités liées à la production. Cette gestion passe notamment par la gestion des stocks et par la gestion de tous les articles qui jouent un rôle dans le processus de production.

Notre projet est réalisé dans le cadre du mémoire de master Génie Logiciel ayant comme objectif principal : la conception et la création d'un progiciel qui met en avant l'interface avec les différents utilisateurs de notre système, afin de pouvoir gérer de façon automatisées toutes les activités liées à la production industrielle.

Le présent document expose les différentes étapes suivies afin de mener ce projet et concevoir le progiciel et ce à travers cinq chapitres.

Nous commençons dans le premier chapitre par introduire les notions plus essentielles dans le domaine du GPAO et détailler les principaux modules de ce dernier.

Le second chapitre est consacré à présenter la démarche utilisée, pour le développement du progiciel, et le langage de modélisation.

Le troisième chapitre se focalise sur la première et la deuxième étape du processus qu'on a choisi pour le développement de notre application qui est la phase de l'expression et l'analyse des besoins. Dans cette partie, nous allons identifier les besoins fonctionnels ou non fonctionnels qui deviendront ensuite des fonctionnalités de notre application, puis définir les acteurs de notre système.

La phase de la conception fait l'objet du quatrième chapitre, dans lequel nous détaillerons les besoins identifiés dans le troisième chapitre. Nous présentons les diagrammes de séquences et le digramme de classes nécessaire à la conception de la solution proposée.

Enfin, le cinquième chapitre, dédié à la réalisation de l'application où nous allons présenter l'environnement de développement, les outils et langages de programmation utilisés dans notre travail, puis montrer quelques interfaces graphiques de notre application réalisée.

Et nous concluons notre travail par une conclusion générale et quelques perspectives.

1

L'UNIVERS DE GPAO

1. Introduction

Les entreprises industrielles de nos jours sont plus complexes et plus vastes qu'avant en ce qui concerne les départements de la production, qui comporte des besoins et des contraintes plus exigeantes et qui demandent plus de temps pour la réalisation. C'est pour cela que le GPAO a été créé en tant que solution pour résoudre ce problème.

Dans ce chapitre, nous allons parler d'une manière globale de la production puis de la gestion de production assistée par ordinateur et de ses principales fonctionnalités et nous terminerons avec les avantages de la GPAO ainsi que les logiciels existants sur le marché.

2. Présentation du produit

Dans un sens général, le produit est ce qui naît d'une activité de la nature ou de l'homme. Un produit est un bien qui résulte d'un processus de production, il peut être un objet matériel, un service, une idée ou une organisation, conçu, créé et offert à la consommation dans le but de satisfaire un besoin identifié des consommateurs.

Les produits permettent de définir le résultat d'une entreprise en créant une augmentation de ce dernier et donc une augmentation de son patrimoine.

3. Gestion de production

La production est le processus conduisant à la création de produits (physiques ou services) par l'utilisation et à la transformation de ressources. Les opérations sont les activités qui le composent [38].

Tandis que la gestion de la production (et des opérations) a comme buts : la conception, la planification et le contrôle des opérations. En gestion de production, on considère, généralement comme données, les caractéristiques du produit suivantes :

- La définition du produit ;
- Le processus de fabrication ;
- La demande à satisfaire.

Les étapes du cycle de production sont au nombre de trois :

- L'ordonnancement : génération et répartition des ordres de fabrication entre les machines, et planification de la production.
- La production : conduite des machines, management des opérateurs et gestion des stocks de matières premières.

- Les contrôles et enregistrements : contrôle de qualité, enregistrement des mouvements de stocks, enregistrement des temps [12].

L'objectif principal de la Gestion de Production est de gérer les flux de matières et d'informations par rapport aux objectifs prioritaires définis par la Direction Générale de l'entreprise.

La GP est difficile à manipuler par rapport aux points suivants :

- a) Incertitude des données : en cas où les machines tombent en pannes, retard d'arrivée de la Matière Première ou ne pas avoir la qualité requise ou bien les ouvriers peuvent tomber malade ou en grève ;
- b) Interconnexion des services : la GP intervient dans tous les services traditionnellement présents dans une entreprise ;
- c) Antagonisme des critères : la maximisation des bénéfices, la minimisation des encours (stocks), la minimisation des retards ;
- d) Complexité des calculs pour obtenir des solutions.

4. Gestion de Production assistée par ordinateur

Un logiciel de GPAO est un programme de gestion informatique qui permet de piloter l'ensemble des activités liées à la production. Son atout principal est de pouvoir suivre et de connaître précisément le coût de revient industriel, de connaître les niveaux de stocks de matières premières, de produits finis et d'encours et d'approvisionner les lignes de production avec le bon composant au bon moment [11].

Il est généralement organisé par deux groupes de modules suivants :

- a) Gestion des données techniques :
 - Gestion des Articles ;
 - Gestion Nomenclatures ;
 - Gestion des Ressources de production ;
 - Gestion des Gammes de fabrication ;
 - Gestion de Stocks.
- b) Gestion de la production :
 - Calcul des besoins nets ;
 - Calcul des charges ;
 - Ordres de fabrication ;

- Ordonnancement d'Ordres de fabrication ;
- Gestion Suivi de production ;
- Calcul des coûts de revient.

5. Les modules d'un GPAO

La GPAO est composé de six modules principaux qui seront détaillés plus loin (la figure 1.1 représente les modules du GPAO générée par nous-même).

Un module est un sous-système du GPAO qui remplit une fonction de haut niveau. Plusieurs modules sont en général mis à disposition et le périmètre fonctionnel couvert par l'application est relatif aux modules mis en service. Les modules peuvent être classés en trois domaines distincts [13] :

- Le domaine métier : est constitué d'une série de modules remplissant les fonctions des processus qui sont situés sur la chaîne de valeur de l'entreprise. C'est le territoire intérieur à l'entreprise où les flux d'information sont directement corrélés aux activités de production et de logistique.
- Le domaine du pilotage : est constitué de modules dédiés à l'analyse et à la synthèse du fonctionnement du système entreprise.
- Le domaine du support : prend en charge l'administration du système informatique. Ce domaine couvre l'ensemble des fonctionnalités permettant d'entretenir la structure de données et de modifier les paramètres de configuration de la solution logicielle.



Figure 1.1 : Les modules du GPAO.

5.1. Gestion des données techniques

Un progiciel de gestion de production est organisé autour de données techniques qui permettent de définir les notions suivantes [37]:

- Les articles correspondent à une entité qui peut être stockée, assemblée, usinée, vendue. Ils peuvent être de la matière première, des composants achetés, des sous-ensembles, des produits finis etc.
- Les nomenclatures permettent de définir l'architecture du produit, des articles fantômes peuvent être créés. Il s'agit en général de sous-ensembles immédiatement incorporés dans le niveau supérieur de la nomenclature.
- Les centres de charge sont la capacité moyenne, les phases et les produits pour lesquels le centre de charge est impliqué, le calendrier d'utilisation, le coût horaire, le type. Elles doivent aussi intégrer les personnels permettant de faire fonctionner ces centres de charge.
- Les gammes de fabrication sont mises en place par le bureau des méthodes (son rôle est de définir les phases, les opérations associées ainsi que leur ordre, il doit préciser les temps de réalisation (Tu) des opérations et des phases ainsi que les temps de changement de série(Ts)). Elles découpent l'ensemble des travaux à réaliser sur le produit en phases, d'où chaque phase est composée d'opérations élémentaires qui sont opérations réalisées sur le même poste ou la même machine sans en changer.

5.2. Un système de gestion de la planification

La gestion de la planification doit permettre de réaliser la planification depuis le long terme jusqu'au pilotage de l'exécution et elle est souvent basée sur une logique MRP.

C'est quoi MRP ?

La planification des besoins en matériaux (MRP) est un système de calcul des matériaux et des composants nécessaires à la fabrication d'un produit. Il se compose de trois étapes principales: l'inventaire des matériaux et des composants disponibles, l'identification de ceux qui sont nécessaires, puis la planification de leur production ou de leur achat [10].

Le système de gestion de planification comporte les niveaux suivants :

- ❖ une planification à long terme permettant de réaliser le Plan industriel et commercial (PIC).

C'est quoi le PIC ?

Le Plan Industriel et Commercial (PIC), généralement élaboré par la direction générale, permet de prévoir et d'organiser les moyens de production et la charge de travail en fonction de ces prévisions (voir figure 1.2) [14]. Il est situé au plus haut niveau de la planification. Son objectif est de définir l'activité de l'entreprise par familles de produits de façon à réaliser l'adéquation entre la charge induite par les besoins commerciaux et la capacité de l'entreprise.

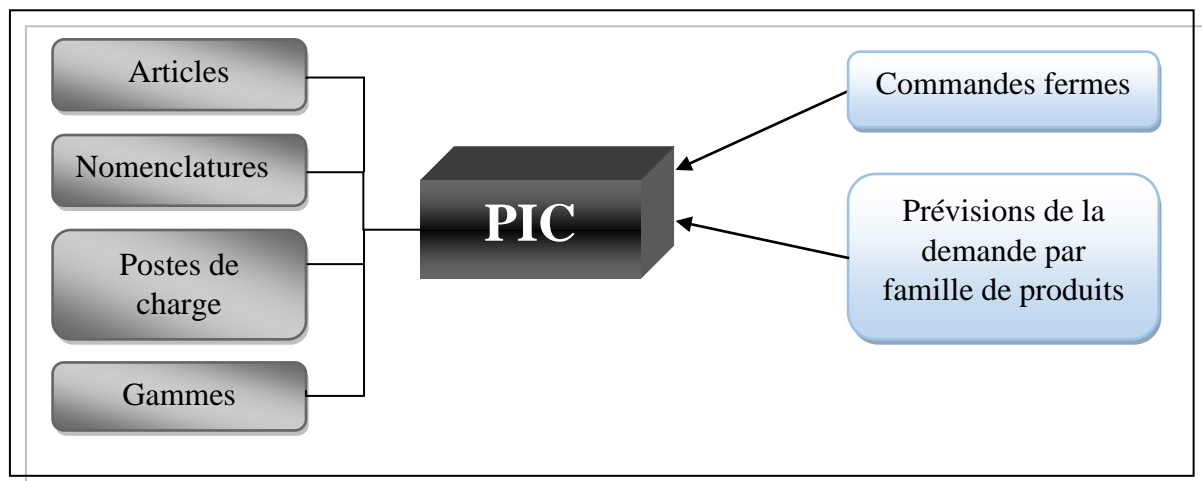


Figure 1.2 : Schéma expliquant le plan industriel et commercial.

- ❖ Une planification plus détaillée et à moyen terme permet de réaliser le Plan Directeur de Production.

Cette planification est réalisée à partir des produits finis qui sont en stock, des résultats du PIC ainsi que des prévisions par produit qui sont schématisés dans la figure 1.3. Un calcul des charges détaillées est aussi réalisé afin de vérifier la capacité de chaque centre de charge.

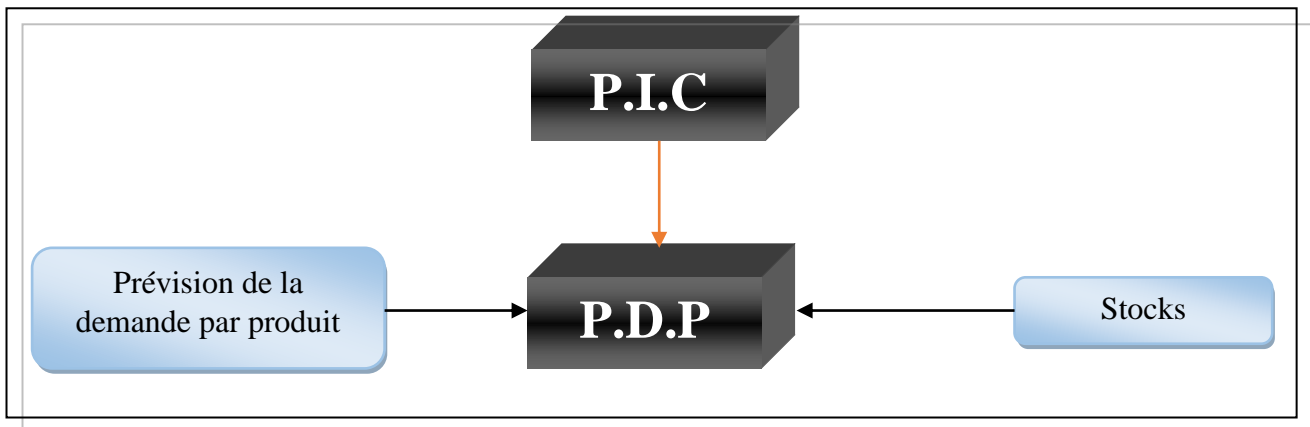


Figure 1.3 : Schéma expliquant le Plan Directeur de Production.

- ❖ Une planification encore plus détaillée après éclatement de la nomenclature par articles grâce au calcul de besoins nets qui permet de calculer les quantités à réaliser ou à commander pour chaque article ainsi que les dates desancements ou des commandes. Et pour cela on a utilisé le pilotage de l'exécution, qui se réalise en plusieurs étapes bien expliqué au-dessous et schématisées juste après dans la figure 1.4 :
 - ✓ L'ordonnancement à partir des ordres de fabrications proposés issus du système MRP. Il permet de transformer les ordres proposés en ordre fermes.
 - ✓ Le lancement permet de transformer les ordres fermes en ordres lancés c'est à dire qui doivent être réalisés par les ateliers.
 - ✓ Le suivi des ordres de fabrication permet d'enregistrer les informations concernant les lots de pièces dans l'atelier.
 - ✓ Les données sont issues des déclarations de la production.

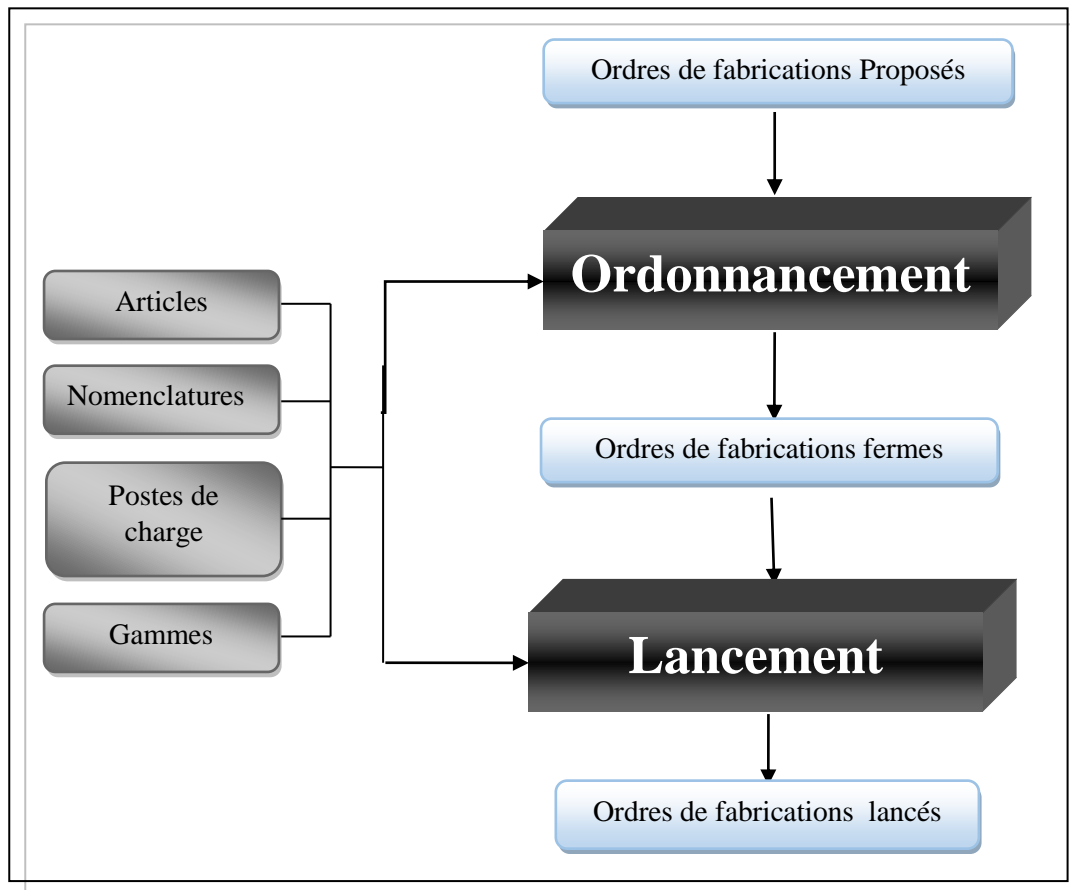


Figure 1.4 : Schéma expliquant les étapes de pilotage de l'exécution.

5.3. Système de gestion des achats

Une des fonctions GPAO est de proposer l'approvisionnement des articles achetés, aussi il est courant de retrouver dans un grand nombre de progiciels du GPAO un module permettant de gérer les achats.

On peut y retrouver les fonctions de : gestion des appels d'offre, gestion des tarifs, achats sur des marchés négociés au préalable, gestion des commandes, gestion des fournisseurs etc.

5.4. Système de gestion des stocks

□ Définition du stock

Le stock est l'ensemble des articles détenus par l'entreprise ; il permet d'assurer la gestion des articles de l'entreprise dans le but de satisfaire, au moment opportun, la disponibilité et la délivrance de ceux-ci pour l'élaboration des produits en évitant absolument les ruptures de stock [9].

❑ La fonction gestion des stocks se compose de 2 sous-fonctions

1. Le suivi des stocks : Cette fonction a pour objectif de connaître à tout moment les articles disponibles dans l'entreprise. Pour cela, elle doit assurer une comptabilité physique et financière des articles.

2. La gestion des stocks : Cette fonction a pour rôle de définir :

- L'optimum d'articles différents à posséder dans l'entreprise en effectuant le plus souvent possible une épuration du stock.
- La politique de réapprovisionnement et de distribution (ou de consommation) la mieux adaptée pour chaque article.

5.5. Gestion des coûts

Lorsque les informations telles que le temps passé, les coûts horaires des centres de charges, les coûts d'achat sont disponibles dans un progiciel, très souvent on retrouve un module de calcul des coûts. Dans ce module on peut retrouver des calculs de coûts : directs et indirects, de fabrication par ordre de fabrication, par articles, des indicateurs de performance etc.

5.6. Système d'intégration CIM

La gestion de production est en liaison avec les autres fonctions de l'entreprise (commercial, comptabilité, finances, bureau d'études....), le besoin de communication est évoqué par le concept de CIM (Computer Integrated Management) dont un des objectifs est de permettre le décloisonnement des fonctions de l'entreprise en permettant l'interaction entre ces fonctions. La création des articles et des nomenclatures est réalisée traditionnellement par le bureau d'études alors que la réalisation des gammes est effectuée au bureau des méthodes, par machine numérique. Les données de coût et celles concernant les stocks peuvent être transmises à la comptabilité générale et analytique lorsque la GPAO possède une interface avec les logiciels comptables utilisés.

6. Quels avantages à utiliser un logiciel GPAO ?

- Un logiciel du GPAO permet à l'entreprise de piloter plus facilement toutes les étapes de la production, en amont et en aval, à un niveau stratégique ou opérationnel.
- Le logiciel du GPAO permet de centraliser l'information dans un référentiel unique et organisé, ce qui constitue un énorme gain de temps pour tous les utilisateurs.

- Le logiciel GPAO permet d'automatiser un grand nombre de processus industriels et d'effectuer des calculs complexes.
- Il apporte un tableau de bord unifié de la production.
- Respect des délais et minimiser les coûts.

7. Les GPAO existants sur le marché

Il existe plusieurs GPAO sur le marché des entreprises, d'où on a cité les plus populaires qui sont :

7.1. SYLOB

Pour des petites aux grandes structures industrielles, ce logiciel a les deux maîtres de mots qui sont :

- L'optimisation de la satisfaction client notamment via la qualité.
- La rentabilité via des outils qui améliorent le pilotage de l'entreprise et donc les prises de décision.

Sylob s'adapte à toute chaîne de production, de plus, les modules relatifs à la gestion commerciale et à la gestion du personnel sont également adaptés aux problématiques du secteur industriel. Des modules de gestion de la qualité, de l'approvisionnement ou encore de la maintenance sont également compris [15].

7.2. CLIPPER, Le leader français des progiciels GPAO

L'un des outils les plus utilisés sur le marché des GPAO, Clipper de Clip Industrie. Le logiciel offre des modules dont l'optimisation de la performance est l'objectif principal.

Clipper est donc dirigé vers les ERP mais plus précisément vers certains métiers précis, l'usinage, la chaudronnerie, le décolletage et la tôlerie/découpe. Les fonctionnalités sont donc d'avance orientée vers ces métiers pour être le plus adapté possible. Ces fonctionnalités ont prioritairement pour objectif de gérer la production, mais des modules complémentaires permettent de gérer tout ce qui a trait au commerce, à la maintenance, à la qualité ou encore à la chaîne d'approvisionnement [15].

7.3. Sage 100Cloud Gestion de production :

Sage 100Cloud Gestion de production est une solution GPAO à destination des PMI assurant une gestion avancée des données et une meilleure planification.

Ce logiciel à interface simple est l'un des leaders du marché qui souhaitent suivre de manière optimale leur activité. Il permet de gérer facilement les fondamentaux de la production [2] [7].

8. Conclusion

Dans ce chapitre, nous avons présenté une vue globale de notre thème qui est la GPAO. Puis nous avons donné quelques généralités de chaque module que compose la GPAO. On a terminé par les principaux avantages du système avec les trois progiciels du GPAO qui sont les plus populaires au monde.

2

CONTEXTE DU PROJET ET MÉTHODOLOGIE DE CONCEPTION

1. Introduction

Dans ce chapitre, nous présenterons une vue globale de notre projet, et de notre système, puis nous définirons tout ce qui concerne le cycle de vie du projet et le processus de développement utilisé, et nous terminerons par le langage de modélisation qui va être notre main droite pour la conception de ce projet.

2. Présentation de notre sujet

Le sujet de notre mémoire consiste à développer un progiciel de gestion de la production assistée par ordinateur GPAO.

On a choisi de faire un progiciel pour réaliser des traitements informatiques standards dont la diffusion revêt un caractère commercial et qu'un usager peut utiliser de façon autonome après une mise en place et une formation limitée, notamment stocker de nombreuses données et les données techniques, donc l'outil informatique permet de réaliser cette fonction de façon rapide et sûre. De plus, ces données peuvent être partagées par les autres services de l'entreprise par l'intermédiaire de réseaux informatiques.

Ce progiciel est conçu pour satisfaire la gestion de production où les produits peuvent être de type assemblage, un assemblage est un procédé permettant de lier entre elles plusieurs pièces pour former un ensemble, il existe différents types d'assemblage en fonction des matériaux utilisés.

3. Présentation du système de gestion de production assistée par ordinateur

Un système GPAO est un logiciel qui permet de piloter globalement l'ensemble des activités de production d'une entreprise, allant de la conception des produits, la planification des ressources nécessaires jusqu'à l'ordonnancement des tâches et des ressources dans l'atelier. Une GPAO est capable de s'adapter à l'utilisateur en proposant une vision stratégique (avec le Plan Industriel et Commercial découlant de la stratégie commerciale de l'entreprise) ou opérationnelle (avec le Plan Directeur de Production et l'ordonnancement) [6].

Sur ce la GPAO permet aux entreprises de connaître les niveaux de stocks (matière première, produits finis et en cours), gérer sur ordinateur de façon automatisée toutes les activités liées à la production industrielle et ainsi qu'optimiser leur productivité et leurs performances métier.

4. Cycle de vie du projet

Chaque projet utilise un type spécifique de cycle de vie, et dans notre projet nous allons utiliser le cycle de vie itératif et incrémental. Dans ce qui suit, nous allons présenter ce cycle de vie qui est destiné pour le processus UP (Unified Process) ainsi que les activités de ces derniers (voir figure 2.1).

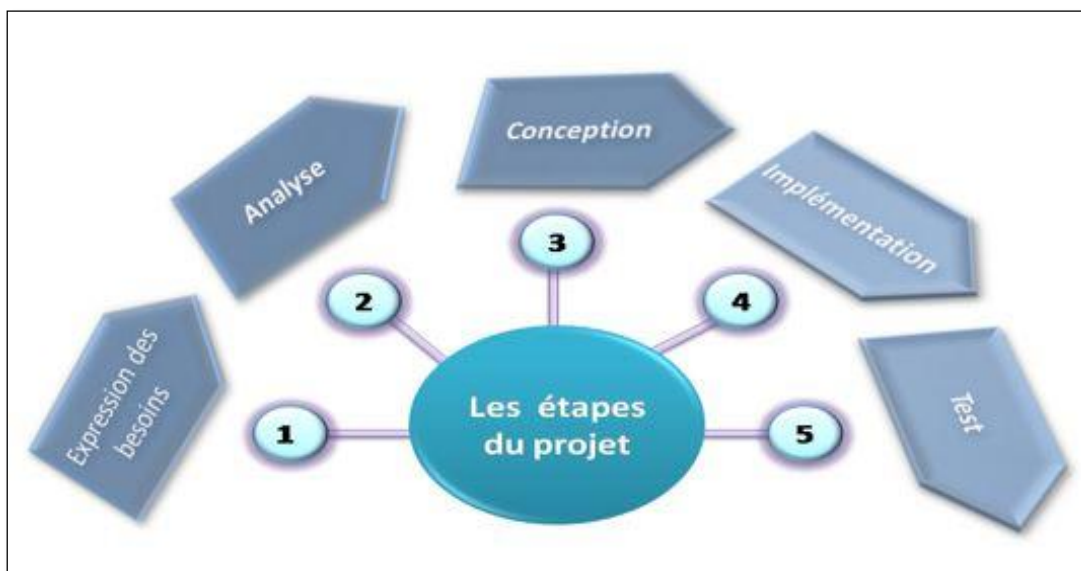


Figure 2.1 : Le cycle de vie du projet.

4.1. Définition d'un cycle de vie

Il s'agit du processus de gestion de projet, composé de ces différentes étapes partantes de l'identification des besoins jusqu'à la clôture du projet [36].

4.2. Les étapes du cycle de vie utilisé

4.2.1. Expression des besoins

L'expression des besoins se fonde sur une bonne compréhension du domaine concerné pour le système à développer et une modélisation des procédures du système existant. Ainsi, on distingue deux types de besoins :

- les besoins fonctionnels qui conduisent à l'élaboration des cas d'utilisation.
- les besoins non fonctionnels (techniques) qui aboutissent à la rédaction d'une matrice des exigences.

4.2.2. Analyse

L'analyse permet la compréhension des besoins et les exigences du client, et il fait donc une livraison de spécification complète des besoins et des structures. Il peut être considéré comme une première ébauche du modèle de conception.

4.2.3. Conception

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune. Elle permet d'étendre la représentation des diagrammes effectuée au niveau de l'analyse en y intégrant les aspects techniques plus proches des préoccupations physiques. Elle constitue un point de départ à l'implémentation.

4.2.4. Implémentation

L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type. Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.

4.2.5. Test

Les tests permettent de vérifier les résultats de l'implémentation en testant la construction grâce au test d'intégration, test de réception, test de performance et test de non-régression.

5. Qu'est-ce que UP ?

C'est un processus de conception et de réalisation de logiciels développés avec des langages de programmation orientés objet, ce processus permet d'éclaircir qui fait quoi, comment et quand les travaux sont réalisés tout au long du cycle de vie de projet et permet à chacun de savoir quelle est sa place dans le processus de production du logiciel [1].

5.1. Les principes d'UP

Le processus de développement UP, associé à UML (Unified Modeling Language), met en œuvre les principes suivants [1] :

5.1.1. Processus guidé par les cas d'utilisation

L'orientation forte donnée ici par UP est de montrer que le système à construire se définit d'abord avec les utilisateurs. Les cas d'utilisation permettent d'exprimer les interactions du système avec les utilisateurs, donc de capturer les besoins.

Une seconde orientation est de montrer comment les cas d'utilisation constituent un vecteur structurant pour le développement et les tests du système. Ainsi, le développement peut se décomposer par cas d'utilisation et la réception du logiciel sera elle aussi articulée par cas d'utilisation.

5.1.2. Processus itératif et incrémental

Ce type de démarche étant relativement connu dans l'approche objet, il paraît naturel qu'UP préconise l'utilisation du principe de développement par itérations successives. Concrètement, la réalisation de maquette et prototype constitue la réponse pratique à ce principe. Le développement progressif, par incrément, est aussi recommandé en s'appuyant sur la décomposition du système en cas d'utilisation.

Les avantages du développement itératif se caractérisent comme suit :

- les risques sont évalués et traités au fur et à mesure des itérations ;
- les premières itérations permettent d'avoir un feed-back des utilisateurs ;
- les tests et l'intégration se font de manière continue ;
- les avancées sont évaluées au fur et à mesure de l'implémentation.

5.1.3. Processus centré sur l'architecture

Les auteurs d'UP mettent en avant la préoccupation de l'architecture du système dès le début des travaux d'analyse et de conception. Il est important de définir le plus tôt possible, même à grandes mailles, l'architecture type qui sera retenue pour le développement, l'implémentation et ensuite le déploiement du système. Le vecteur des cas d'utilisation peut aussi être utilisé pour la description de l'architecture.

5.1.4. Processus orienté par la réduction des risques

L'analyse des risques doit être présente à tous les stades de développement d'un système. Il est important de bien évaluer les risques des développements afin d'aider à la bonne prise de décision. Du fait de l'application du processus itératif, UP contribue à la diminution des risques au fur et à mesure du déroulement des itérations successives [1].

5.2. Pourquoi UP ?

C'est un processus de développement moderne qui répond bien aux exigences de l'utilisateur. L'UP facilite la compréhension et l'adoption d'une méthodologie orientée objet et fournit une méthodologie et des notations standard qui aident à concevoir des logiciels de qualité.

5.3. Le langage de modélisation utilisé par UP

UML (**Unified Modeling Language**) que l'on peut traduire par "*langage de modélisation unifié*") est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

Ses points forts est que l'UML est un langage formel, normalisé et aussi c'est un support de communication performant mais nécessite un apprentissage et passe par une période d'adaptation.

6. Conclusion

Ce chapitre est consacré à la présentation de notre sujet en détaillant les caractéristiques du cycle de vie, le processus et le langage de modélisation qui va être utilisé lors du développement et la conception de notre projet.

Dans le prochain chapitre, nous exprimerons les besoins du système et représenterons des notions qui sont liées à la GPAO.

3

SPÉCIFICATION ET ANALYSE DES BESOINS

1. Introduction

Dans ce chapitre, nous allons commencer par exprimer les besoins fonctionnels et non fonctionnels. Ensuite on va faire l'analyse des besoins : ce qui va contenir l'identification des acteurs du système, le diagramme de contexte, diagramme de cas d'utilisation globale et enfin les diagrammes des cas d'utilisation plus détaillés ainsi que leurs descriptions.

2. Spécification des besoins

La gestion de production a besoin de stocker de nombreuses données, notamment les données techniques et l'outil informatique permet de réaliser cette fonction de façon rapide et sûre. De plus, ces données peuvent être partagées par les autres services de l'entreprise par l'intermédiaire de réseaux informatiques.

2.1. Les besoins fonctionnels

Dans cette section, nous allons représenter l'ensemble des besoins fonctionnels aux quels devrait répondre notre application et se résumant dans les points suivants :

- ❑ **Gestion des comptes :** créer, mettre à jour et supprimer les comptes des utilisateurs ainsi que l'attribution des droits d'accès aux utilisateurs.
- ❑ **Gestion des articles:** l'ajout, la mise à jour et la suppression des articles quel que soit les nomenclatures ou les gammes.
- ❑ **Gestion de stock :** c'est la gestion des entrants/sortants dans le stock de production ainsi que la consultation des besoins de la production.
- ❑ **Gestion des ordres de fabrication :** donner une autorisation d'ordonnancement à l'atelier de fabrication pour la production d'un produit, ou de pièces, ce document regroupe l'ensemble des données nécessaires au produit, et précise les tâches et les opérations à effectuer.
- ❑ **Génération de rapport :** rédaction des rapports va être générée par le producteur qui concernera que la production.
- ❑ **Consulter le suivi de fabrication.**
- ❑ **Lancement des ordres de fabrication :** c'est le fait de réaliser ou lancer les ordres déjà donnés.
- ❑ **Gestion des ordres de fabrication :** planification des opérations.

2.2. Les besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation comme le langage de programmation, le type du SGBD et du système d'exploitation [7].

Le progiciel à réaliser doit respecter les besoins suivants :

- Fonctionnement : L'application fonctionne sous réseau ;
- Ergonomie : cohérence de l'application et interfaces simples et faciles à manipuler par les utilisateurs;
- Sécurité et confidentialité : l'accès aux informations est sécurisé par une authentification obligatoire de chaque rôle et restreint aux opérations nécessaires.

3. Analyse des besoins

Dans cette partie, nous allons identifier les acteurs, modéliser le contexte du projet et le diagramme de cas d'utilisation associé aux acteurs du système ainsi que leurs descriptions.

3.1. Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités qui interagissent directement avec le système étudié, il peut être principal ou secondaire [2].

Dans le tableau 3.1 qui suit, nous allons définir les différents acteurs et leurs fonctionnalités :

Acteur	Fonctionnalités
Administrateur	<ul style="list-style-type: none"> ▪ Gestion des comptes ; ▪ Consultation du suivi de fabrication.
Inventaire de production	<ul style="list-style-type: none"> ▪ Gestion de stock : <ul style="list-style-type: none"> ▫ Gestion des entrants et des sortants dans le stock de production ; ▫ Consultation des besoins de la production ; ▪ Consultation du suivi de fabrication.
Producteur	<ul style="list-style-type: none"> ▪ Lancement des ordres de fabrication ; ▪ Consultation du suivi de fabrication ; ▪ Génération de rapport ;

Planificateur	<ul style="list-style-type: none"> ▪ Gestion des articles : <ul style="list-style-type: none"> ▫ Gestion des nomenclatures de production ; ▫ Gestion des gammes de production ; ▪ Gestion des ordres de fabrication : <ul style="list-style-type: none"> ▫ Gestion de numéro de lots ; ▫ Ordonnancement de la production ; ▪ Consultation du suivi de fabrication.
---------------	---

Tableau 3.1 : Les fonctionnalités des acteurs de notre système.

3.2. Modélisation du contexte

Le diagramme de contexte permet la représentation des mouvements de données à l'intérieur d'un système d'information, le tableau 3.2 illustre les différents messages échangés entre les acteurs de notre système et le système.

<i>ID</i>	Messages acteur-système	<i>ID</i>	Messages système-acteur
<i>M1</i>	Demande de l'authentification.	<i>M1'</i>	Interface Authentification.
<i>M2</i>	Demande de gestion des comptes.	<i>M2'</i>	Interface des comptes.
<i>M3</i>	Demande de gestion des articles.	<i>M3'</i>	Interface des articles.
<i>M4</i>	Demande de gestion de stock.	<i>M4'</i>	Interface du stock.
<i>M5</i>	Demande de lancement des ordres de fabrication.	<i>M5'</i>	Interface de lancement des ordres de fabrication.
<i>M6</i>	Demande de la consultation du planning et le suivi de fabrication.	<i>M6'</i>	Interface de la consultation du planning et le suivi de fabrication.
<i>M7</i>	Demande de génération de rapports.	<i>M7'</i>	Interface de génération de rapports.
<i>M8</i>	Demande de gestion des ordres de fabrication.	<i>M8'</i>	Interface de gestion des ordres de fabrication.

Tableau 3.2 : Message échangés entre les acteurs et le système.

Le diagramme de la figure 3.1 représente les différents acteurs du système, ainsi que les messages échangés entre ces derniers.

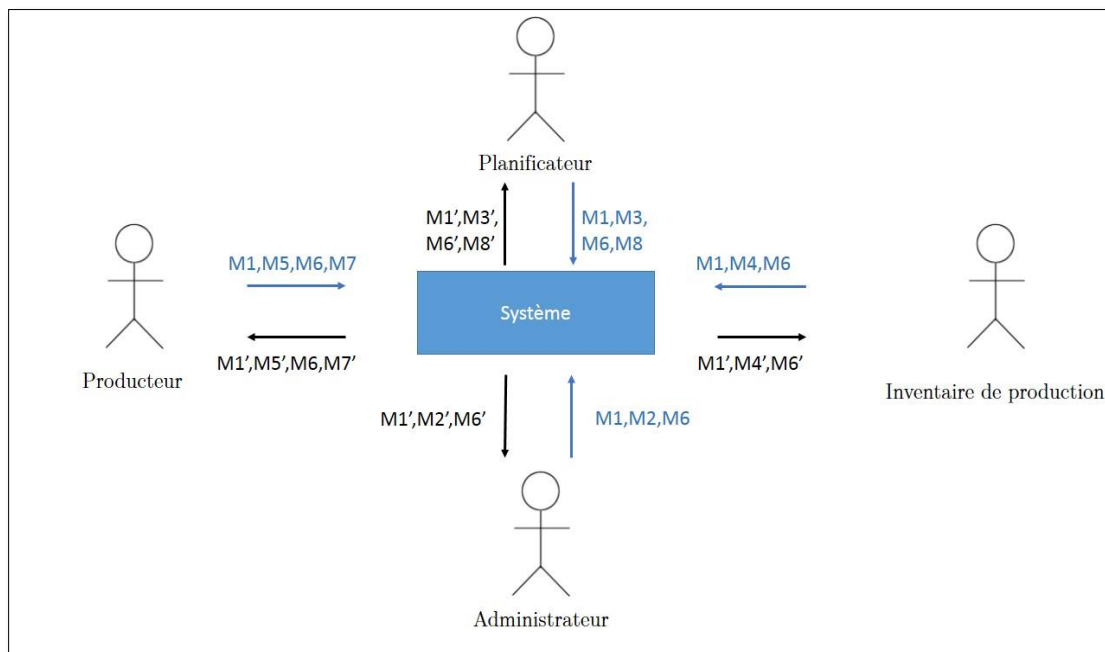


Figure 3.1 : Diagramme de contexte.

3.3. Identification des cas d'utilisation

Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

Un cas d'utilisation se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype [16].

Les cas d'utilisations que nous avons identifiées pour notre système sont représentés dans la figure 3.2 et ils seront détaillés plus loin.

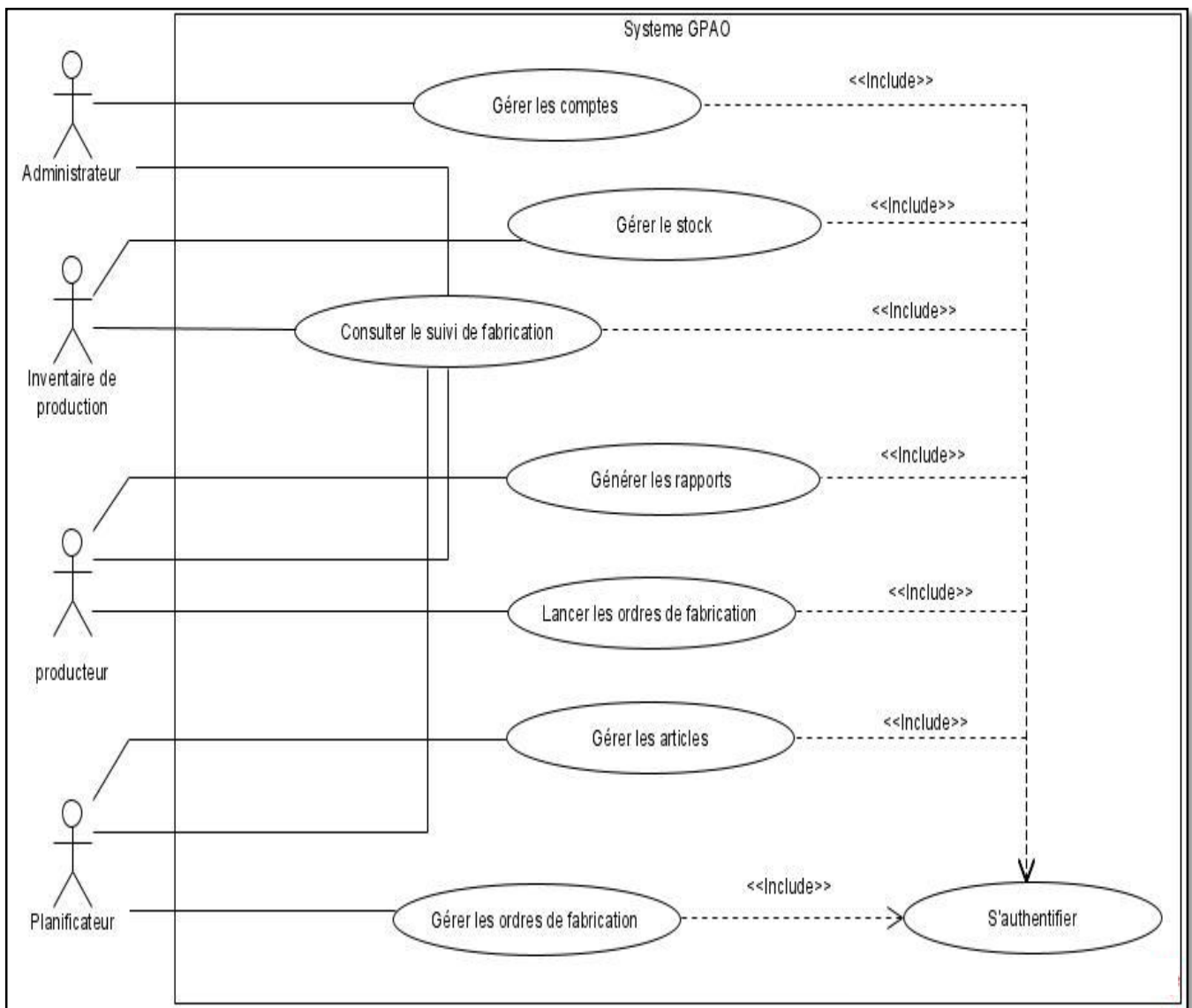


Figure 3.2 : Diagramme de cas d'utilisation globale.

3.4. Diagrammes des cas d'utilisation par acteur et leurs descriptions

Dans ce qui suit nous allons définir les cas d'utilisations associés à leurs acteurs avec des tableaux descriptifs de chaque cas.

3.4.1. Coté « Administrateur »

Le diagramme de cas d'utilisation dans la figure 3.3 présente les fonctionnalités de l'administrateur.

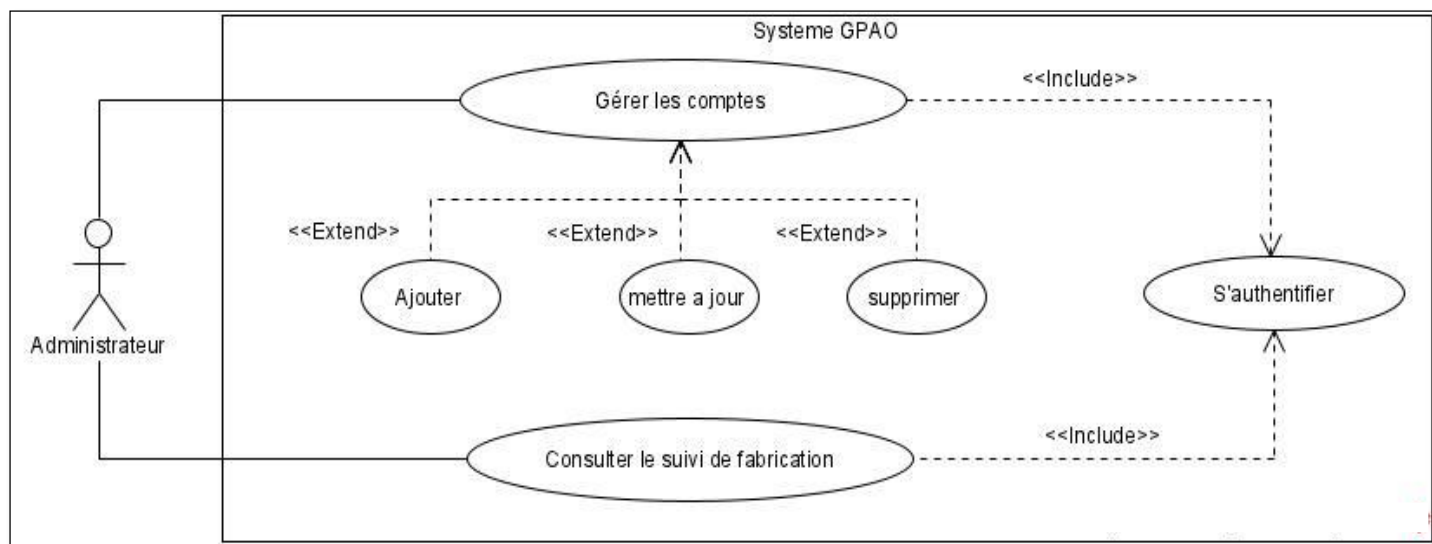


Figure 3.3 : Diagramme des cas d'utilisations de l'administrateur.

❖ Description du cas d'utilisation « Gérer les comptes »

Cas d'utilisation	Gérer les comptes.
Acteur	Administrateur.
Objectif	L'ajout, la mise à jour et la suppression des comptes.
Pré-conditions	Authentification, le système affiche le menu principal.
Scénario nominale	- Le système affiche l'interface de gestion de comptes - L'utilisateur effectue la mise à jour et valide. - Le système enregistre les données
Alternative	Si le champ d'information n'est pas valide le système affiche un message d'erreur et réaffiche l'interface de gestion de comptes.

Tableau 3.3 : Description du cas d'utilisation « Gérer les comptes ».

3.4.2. Coté « Inventaire de production »

Le diagramme de cas d'utilisation suivant présente les fonctionnalités de l'inventaire de production :

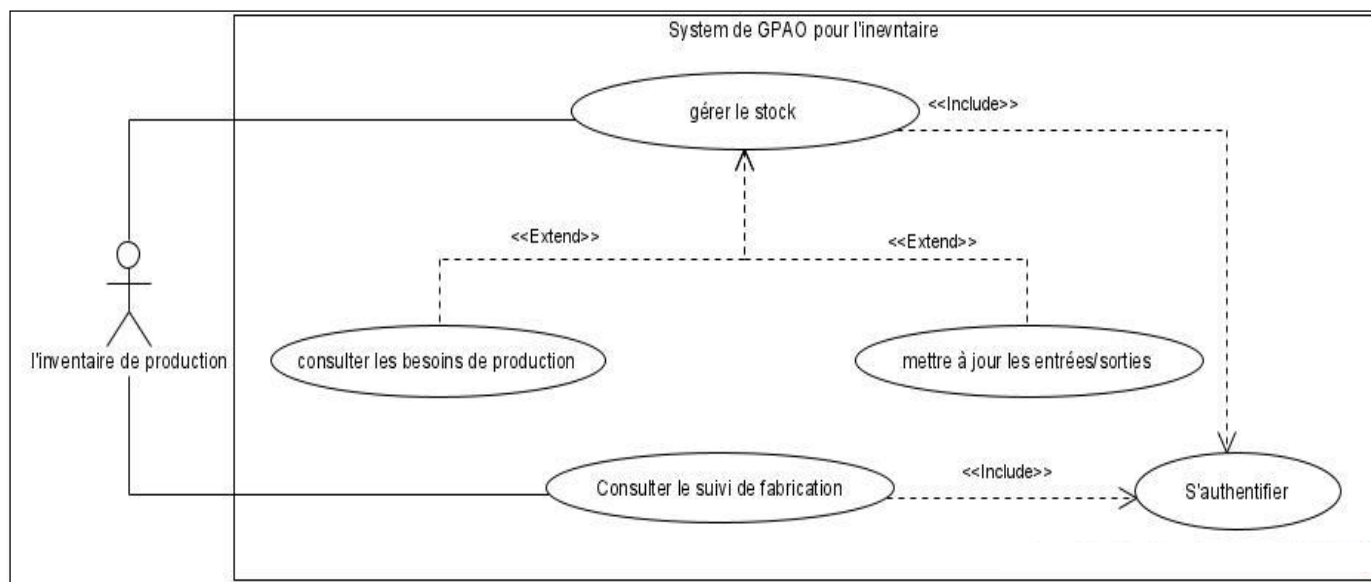


Figure 3.4: Diagramme de cas d'utilisations de l'inventaire de production.

❖ Description du cas d'utilisation « Gérer le stock »

Cas d'utilisation	Gérer le stock.
Acteur	Inventaire de production.
Objectif	L'ajout, la mise à jour, la suppression des entrants/sortants et la consultation des besoins de production.
Pré-conditions	Authentification, le système affiche le menu principal.
Scénario nominale	<ul style="list-style-type: none"> - L'utilisateur sélectionne l'onglet «Stock». - Le système affiche l'onglet (sous menu) relatif aux stocks. - L'utilisateur gère les entrants/sortants ou il peut consulter les besoins de production. - Le système enregistre les modifications effectuées dans la base de données.

Tableau 3.4 : Description du cas d'utilisation « Gérer le stock ».

3.4.3. Coté « Producteur »

Le diagramme de cas d'utilisation suivants présente les fonctionnalités du producteur dans notre système :

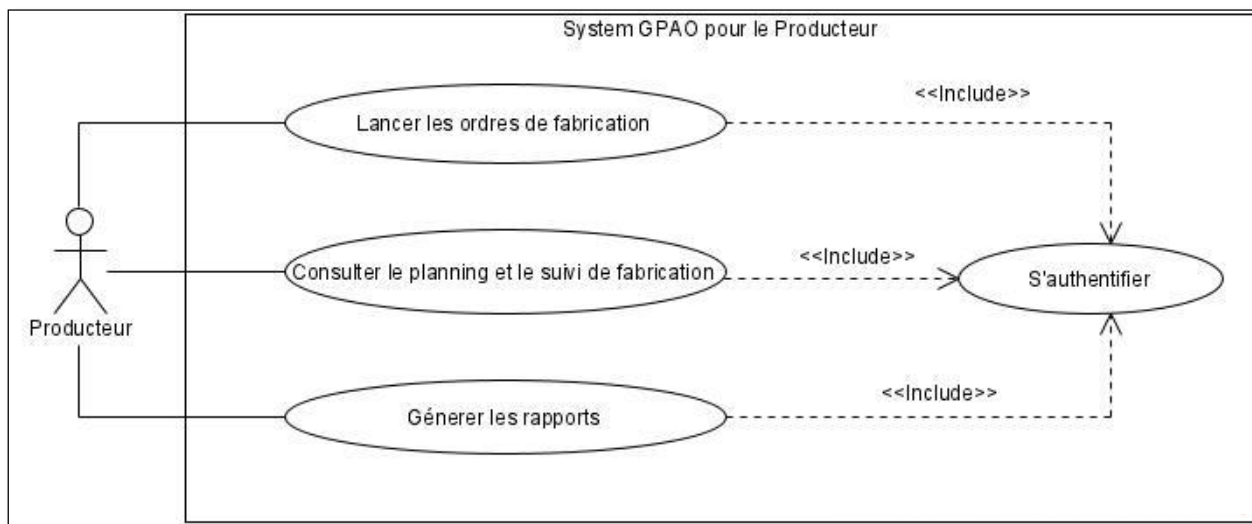


Figure 3.5: Diagramme de cas d'utilisations de producteur.

❖ Description du cas d'utilisation « Générer les rapports »

Cas d'utilisation	Générer les rapports.
Acteur	Producteur.
Objectif	Consultation d'article et du stock afin de générer des rapports.
Pré-conditions	Authentification, le système affiche le menu principal.
Scénario nominale	<ul style="list-style-type: none"> - Le système affiche l'interface de génération de rapports ; - L'utilisateur demande de générer un rapport au système ; - le système affiche l'interface des rapports ; - L'utilisateur choisit la génération de rapport ; - Le système génère, enregistre le rapport et l'envoie au système concerné.
Alternative	S'il n'y a pas du changement à propos du stock ou article l'utilisateur ne pourra pas générer de rapport dans ce cas le système affiche un message « Pas de mise à jour » et réaffiche l'interface de génération des rapports.

Tableau 3.5 : Description du cas d'utilisation « Générer les rapports ».

❖ Description du cas d'utilisation « Lancer les ordres de fabrication »

Cas d'utilisation	Lancer les ordres de fabrication.
Acteur	Producteur.
Objectif	Lancement des ordres de fabrication afin de fournir le produit final.
Pré-conditions	Authentification, le système affiche le menu principal.
Scénario nominale	<ul style="list-style-type: none"> - L'utilisateur sélectionne l'onglet «Lancement de fabrication». - Le système affiche l'onglet (sous menu) relatif au lancement des ordres de fabrications. - L'utilisateur choisit de gérer la production. - Le système lance les ordres de fabrication.

Tableau 3.6 : Description du cas d'utilisation « lancer les ordres de fabrication ».

3.4.4. Coté " Planificateur "

Le diagramme de cas d'utilisation dans la figure 3.6 représente les fonctionnalités du planificateur dans notre système :

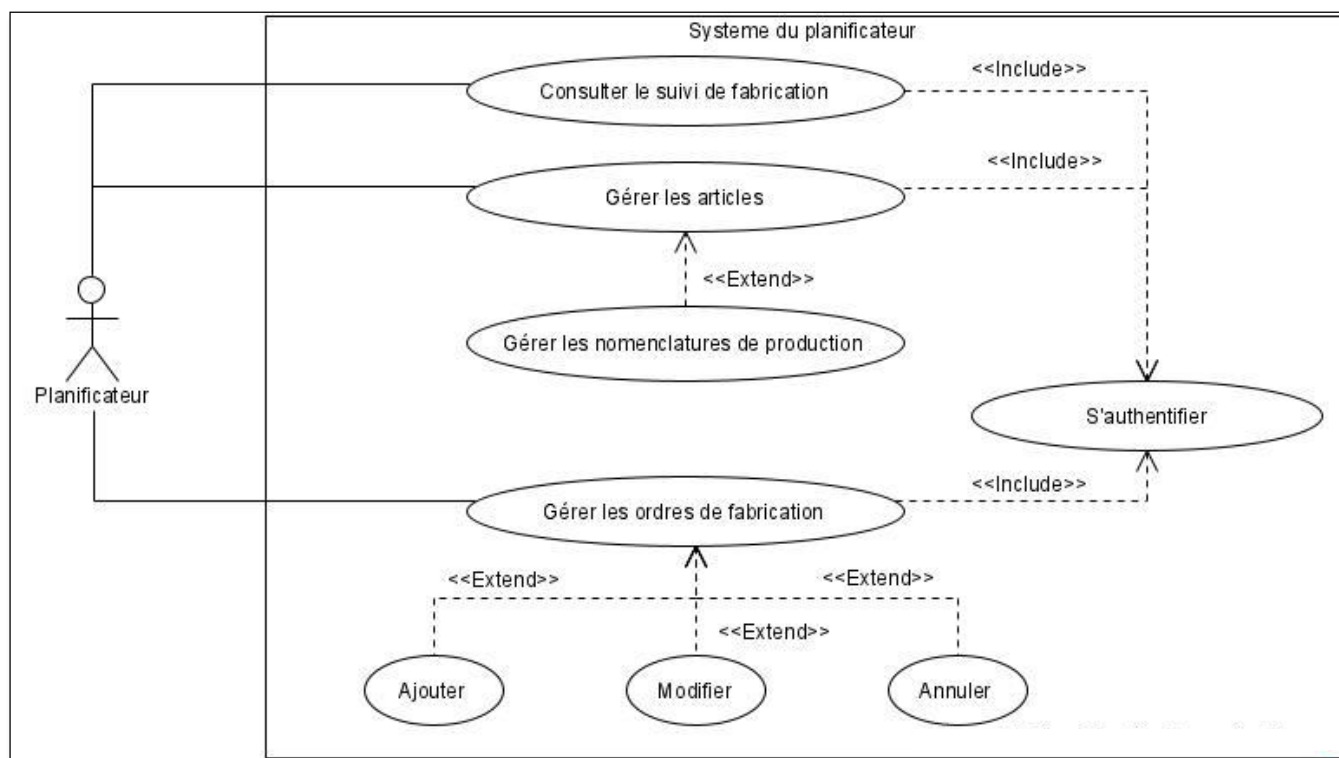


Figure 3.6 : Diagramme des cas d'utilisations du planificateur.

❖ Description du cas d'utilisation « Gérer les articles »

Cas d'utilisation	Gérer les articles.
Acteur	Planificateur.
Objectif	L'ajout, la mise à jour et la suppression des articles quelques soit les nomenclatures.
Pré-conditions	Authentification, le système affiche le menu principal.
Scénario nominale	<ul style="list-style-type: none"> - L'utilisateur sélectionne le menu «Nomenclatures» - Le système affiche l'onglet (sous menu) relatif aux Nomenclatures. - L'utilisateur gère les nomenclatures en ayant la possibilité de les ajouter, modifier ou supprimer des articles. - Le système enregistre les modifications effectuées dans la base de données.

Tableau 3.7 : Description du cas d'utilisation « Gérer les articles ».

❖ Description du cas d'utilisation « Gérer des ordres de fabrication »

Cas d'utilisation	Gérer des ordres de fabrication.
Acteur	Planificateur.
Objectif	Gestion de numéro de lots et l'ordonnancement de la production.
Pré-conditions	Authentification, le système affiche le menu principal.
Scénario nominale	<ul style="list-style-type: none"> - L'utilisateur sélectionne l'onglet «Ordre de fabrication». - Le système affiche l'onglet (sous menu) relatif aux ordres de fabrications. - L'utilisateur gère les ordres de fabrication en ayant la possibilité de gérer le numéro de lots ainsi que l'ordonnancement de la production. - Le système enregistre les modifications effectuées dans la base de données.

Tableau 3.8 : Description du cas d'utilisation « Gérer des ordres de fabrication ».

3.4.5. Digrammes de cas d'utilisation en commun

Les deux descriptions suivantes expliquent les deux cas d'utilisations qui sont en communs entre tous les utilisateurs.

❖ Description du cas d'utilisation « s'authentifier » qui est commun pour tous les utilisateurs du système

Cas d'utilisation	S'authentifier
Acteur	Administrateur Inventaire de production Producteur Planificateur
Objectif	Vérifier que cet acteur existe dans la base de données du système.
Scénario nominale	- L'utilisateur demande l'accès à l'application - Le système affiche l'interface Authentification - L'utilisateur introduit son identifiant et le mot de passe - Le système vérifie l'existence de l'utilisateur. - Si l'utilisateur existe, le système donne l'accès à l'interface correspondante
Alternative	Si le champ d'information n'est pas valide le système affiche un message d'erreur et réaffiche l'interface de l'authentification.

Tableau 3.9 : Description du cas d'utilisation « s'authentifier ».

❖ Description de cas d'utilisation « Consulter le suivi de fabrication »

Cas d'utilisation	Consulter le suivi de fabrication.
Acteur	L'administrateur Inventaire de production Producteur Planificateur
Objectif	Consultation de suivi de fabrication
Pré-conditions	Authentification, le système affiche le menu principal.

Scénario nominale	<ul style="list-style-type: none"> - L'utilisateur sélectionne «Ordre de fabrication» sur le menu « Fabrication ». - Le système affiche le menu «Ordre de fabrication». - Le système affiche les ordres de travail appartenant aux ordres de fabrication en cours.
Post-conditions	Les ordres de travail appartenant aux ordres de fabrication en cours seront affichés.

Tableau 3.10 : Description du cas d'utilisation « Consulter le suivi de fabrication ».

4. Conclusion

Dans ce chapitre, nous avons identifié les besoins fonctionnels et non fonctionnels, puis nous avons identifié les acteurs de notre système et leurs différentes fonctionnalités, nous les avons ensuite exprimés en forme de diagramme de cas d'utilisations.

L'étape de la conception est la phase la plus détaillée dans notre projet, dont on va en parler dans le chapitre qui suit.

4

LA CONCEPTION

1. Introduction

Dans ce chapitre, nous étendrons la représentation des diagrammes de cas d'utilisation effectués au niveau de spécification et analyse des besoins. Nous commencerons par établir les diagrammes de séquences des cas d'utilisation, puis nous définirons les patrons de conception qui seront utilisés lors de l'implémentation. Enfin la phase de conception s'achèvera par l'élaboration du diagramme de classes conceptuel ainsi que le modèle logique de données.

2. Diagrammes de séquences des cas d'utilisation pour chaque utilisateur

2.1. Diagramme de séquence associé à tous les utilisateurs

Les utilisateurs de notre système ont en commun deux cas d'utilisation (voir les figures 4.1 et 4.2) qui sont « s'authentifier » et « consulter le suivi de fabrication ».

Alors nous commencerons ici par présenter les diagrammes de séquences associés à ces deux cas d'utilisation.

2.1.1. Diagramme de séquence du cas d'utilisation « s'authentifier »

Un utilisateur doit s'authentifier en saisissant ses propres coordonnées (identifiant et mot de passe), puis le système procède à la vérification des informations introduites pour les comparer avec les données stockées dans la base de données, si l'une des coordonnées est incomplète, login ou mot de passe est incorrect le système affiche un message d'erreur sinon l'accès est autorisé.

Le diagramme de séquence dans la figure 4.1 représente l'enchaînement des actions dans notre système lors de l'authentification, qui est commune pour tous les utilisateurs.

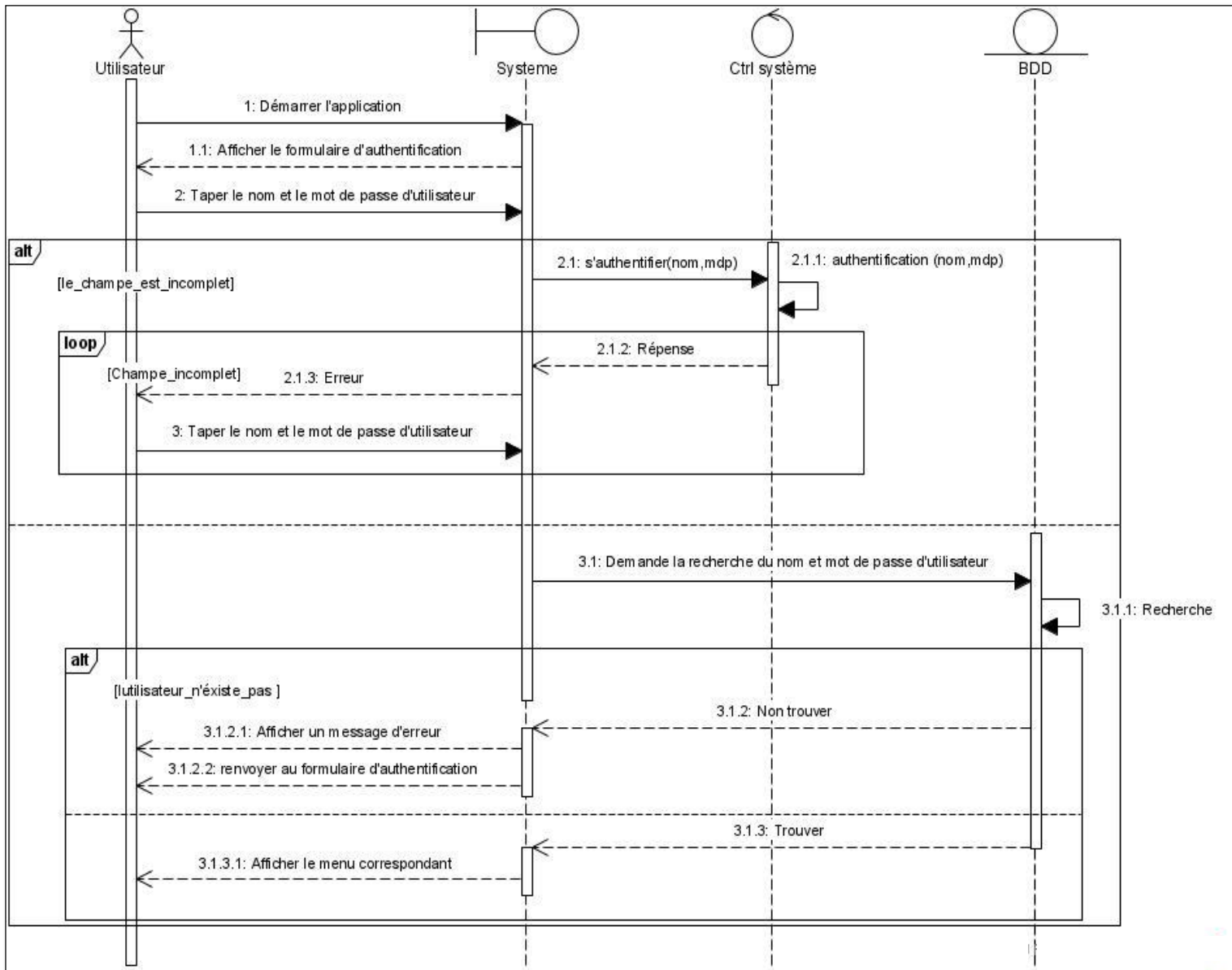


Figure 4.1 : Diagramme de séquence du cas d'utilisation « s'authentifier ».

2.1.2. Diagramme de séquence du cas d'utilisation « Consulter le suivi de fabrication »

Dans ce diagramme de séquence, un utilisateur doit tout d'abord s'authentifier puis demander l'interface associée à ce cas d'utilisation pour qu'il consulte tout le suivi de fabrication.

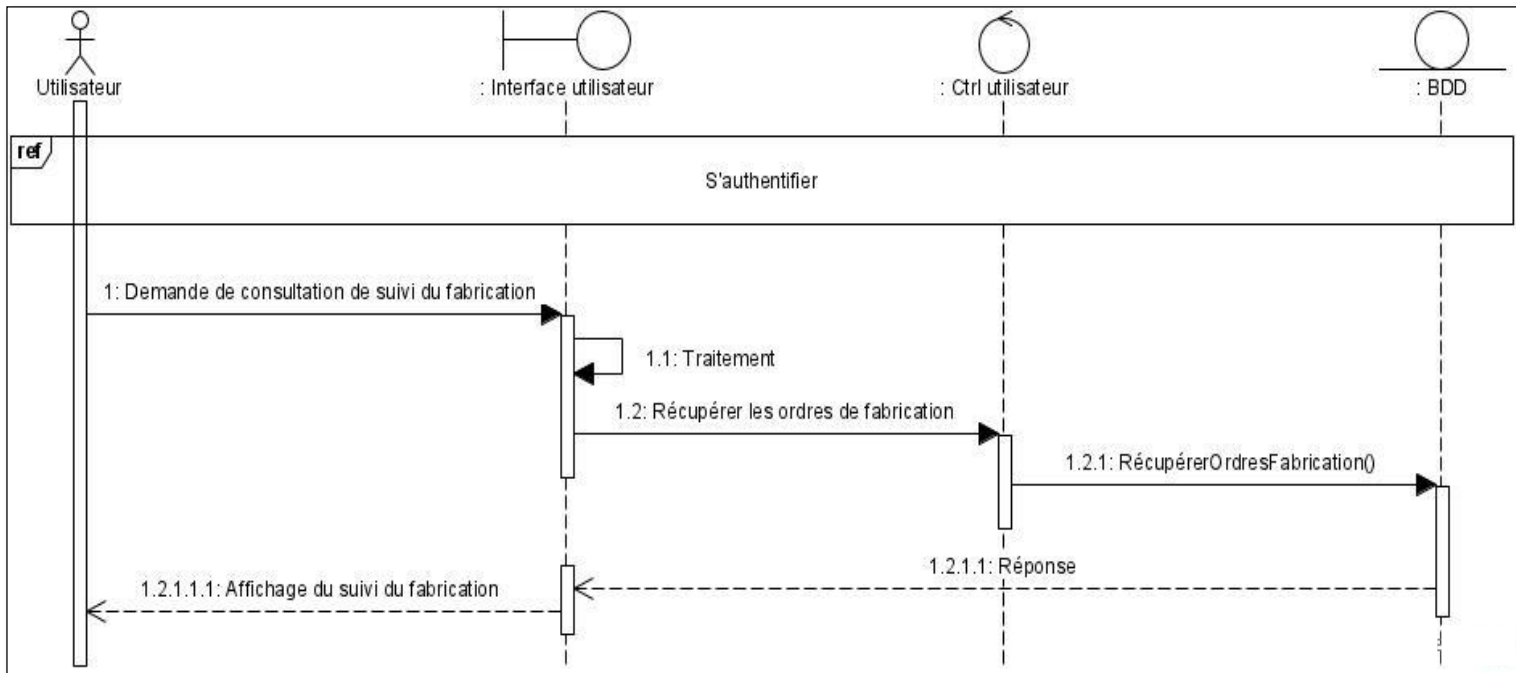
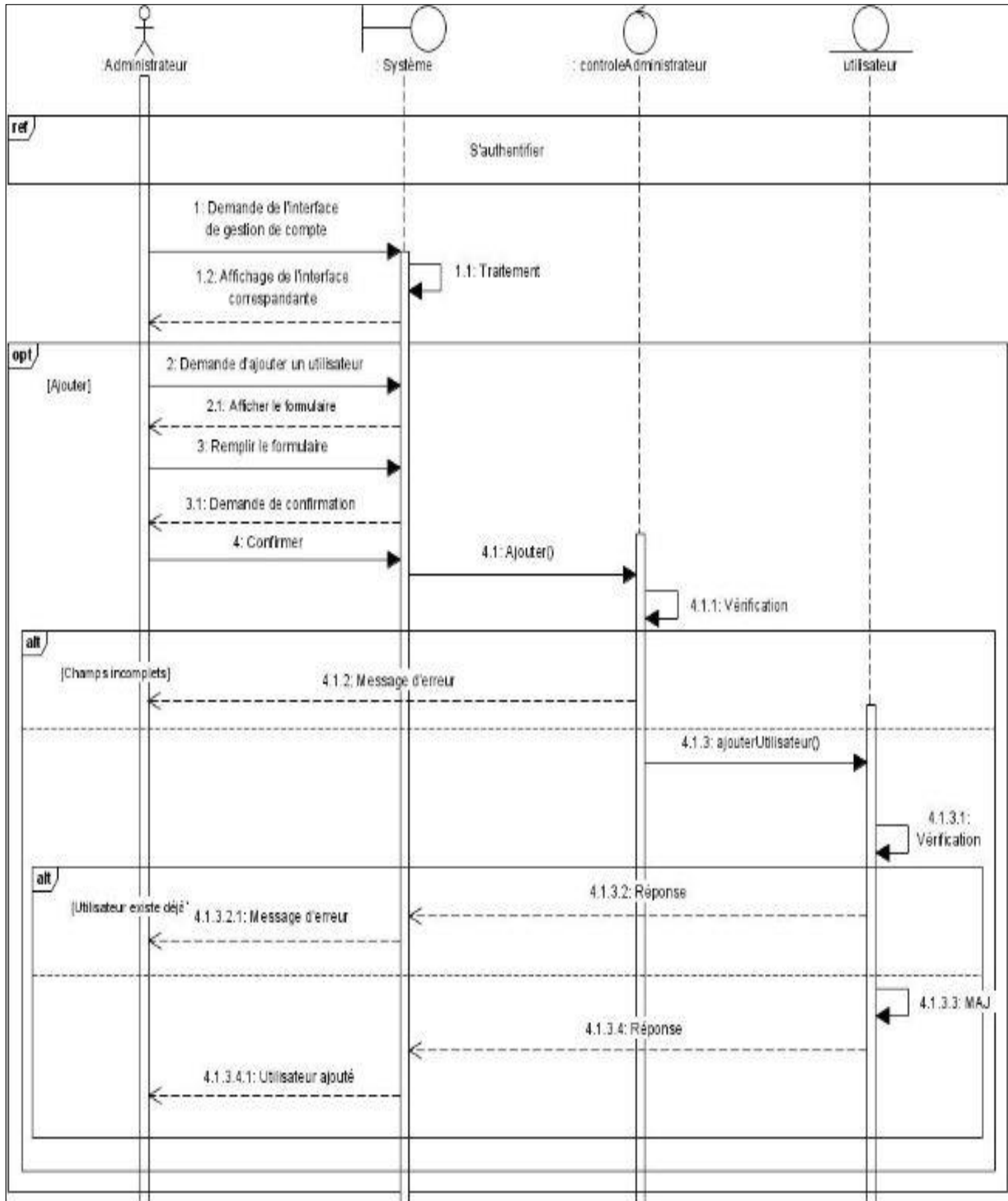


Figure 4.2 : Diagramme de séquence du cas d'utilisation « Consulter le suivi de fabrication ».

2.2. Diagramme de séquence associé à l'« administrateur »

Dans ce diagramme de séquence, l'administrateur doit d'abord s'authentifier, ce dernier peut établir deux cas d'utilisation différents, le premier cas est la gestion des comptes dont il peut aussi effectuer trois options différentes, l'ajout d'un utilisateur dans la base de données en remplissant un formulaire, la modification ou la suppression d'un utilisateur sélectionné sur l'interface dans la base de données, et le deuxième cas est le suivi de fabrication.



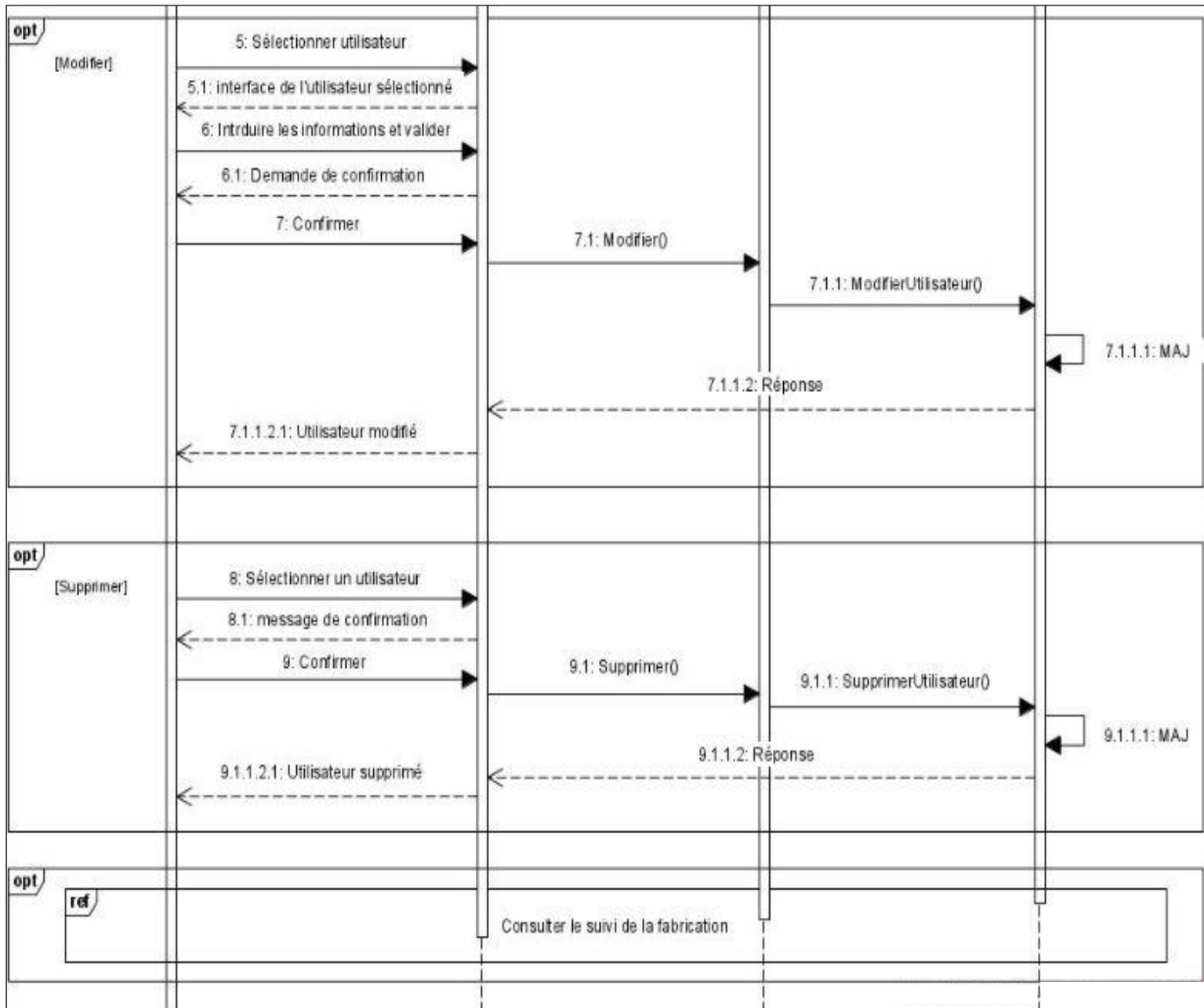


Figure 4.3 : Diagramme de séquence associé à l'« administrateur ».

2.3. Diagramme de séquence associé au « producteur »

Dans ce diagramme de séquence, le producteur doit tout d'abord s'authentifier, puis il pourra établir trois cas d'utilisation différents qui sont la consultation du suivi de la fabrication, la génération de rapport et la consultation des ordres de la fabrication :

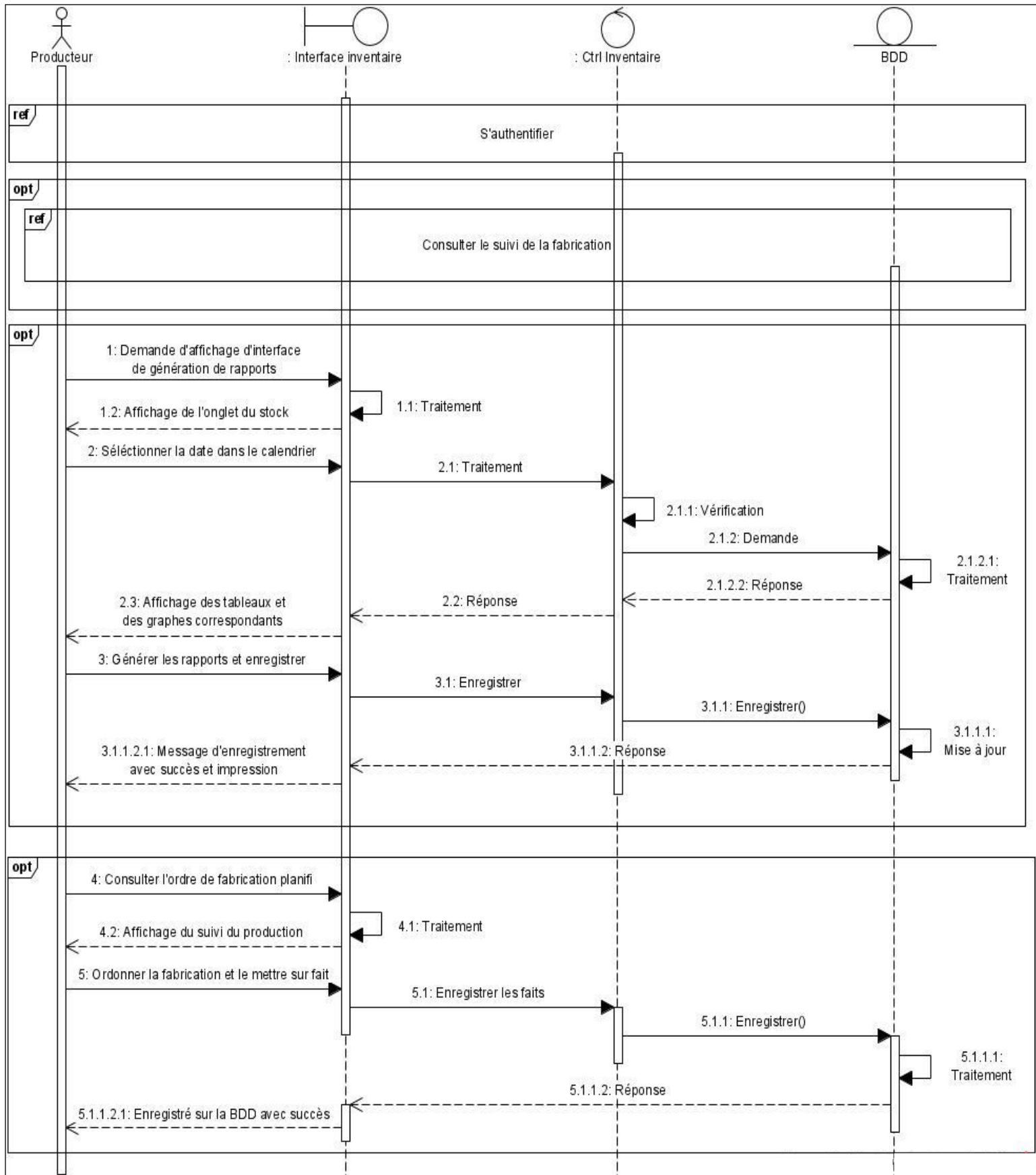


Figure 4.4 : Diagramme de séquence associé au « producteur ».

2.4. Diagramme de séquence associé à « Inventaire de production »

Dans ce diagramme de séquence, l'inventaire de production doit d'abord s'authentifier, ce dernier peut établir trois cas d'utilisation différents, la gestion de stock, consultation des besoins de la production, et la consultation du suivi de fabrication :

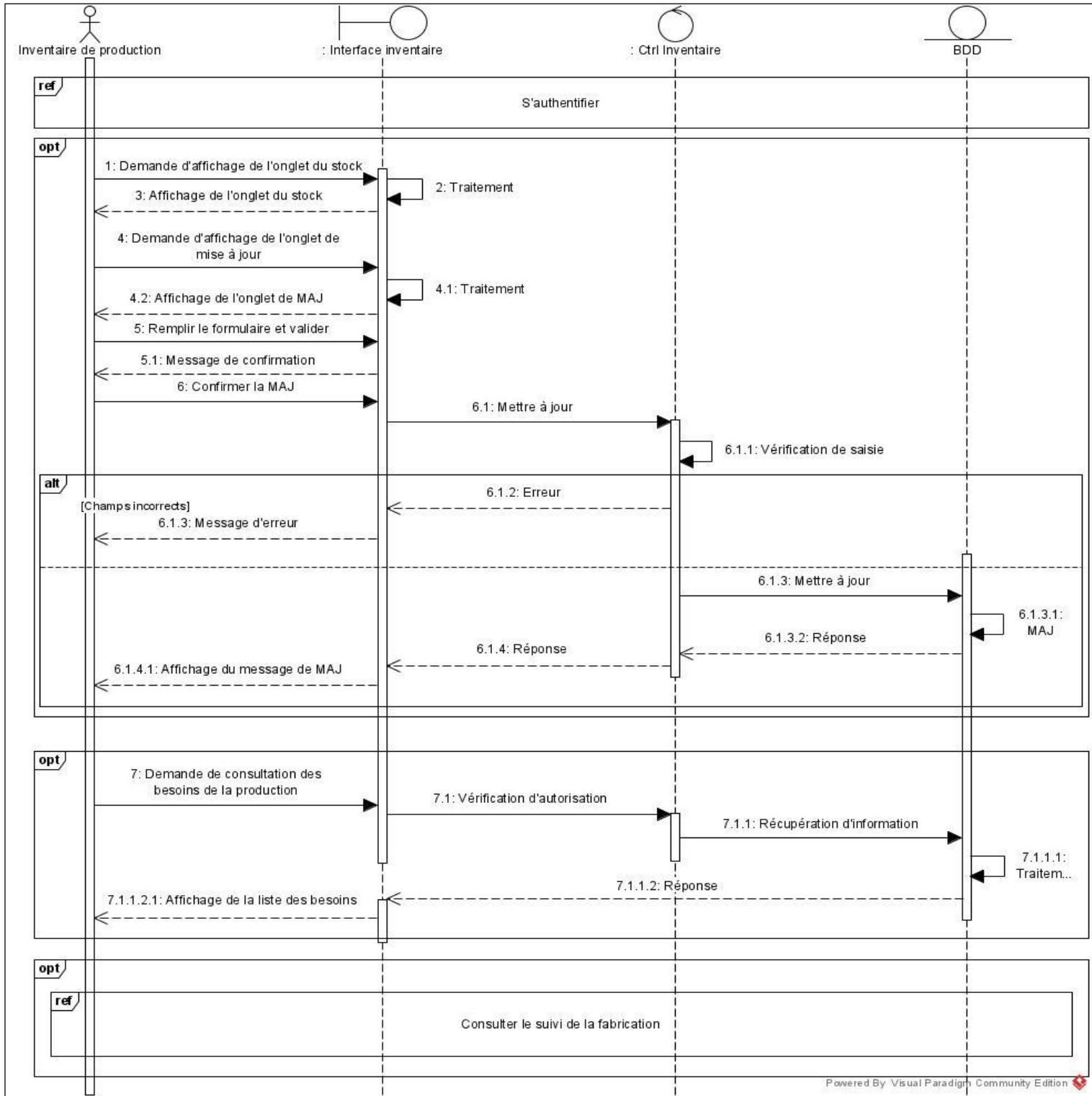
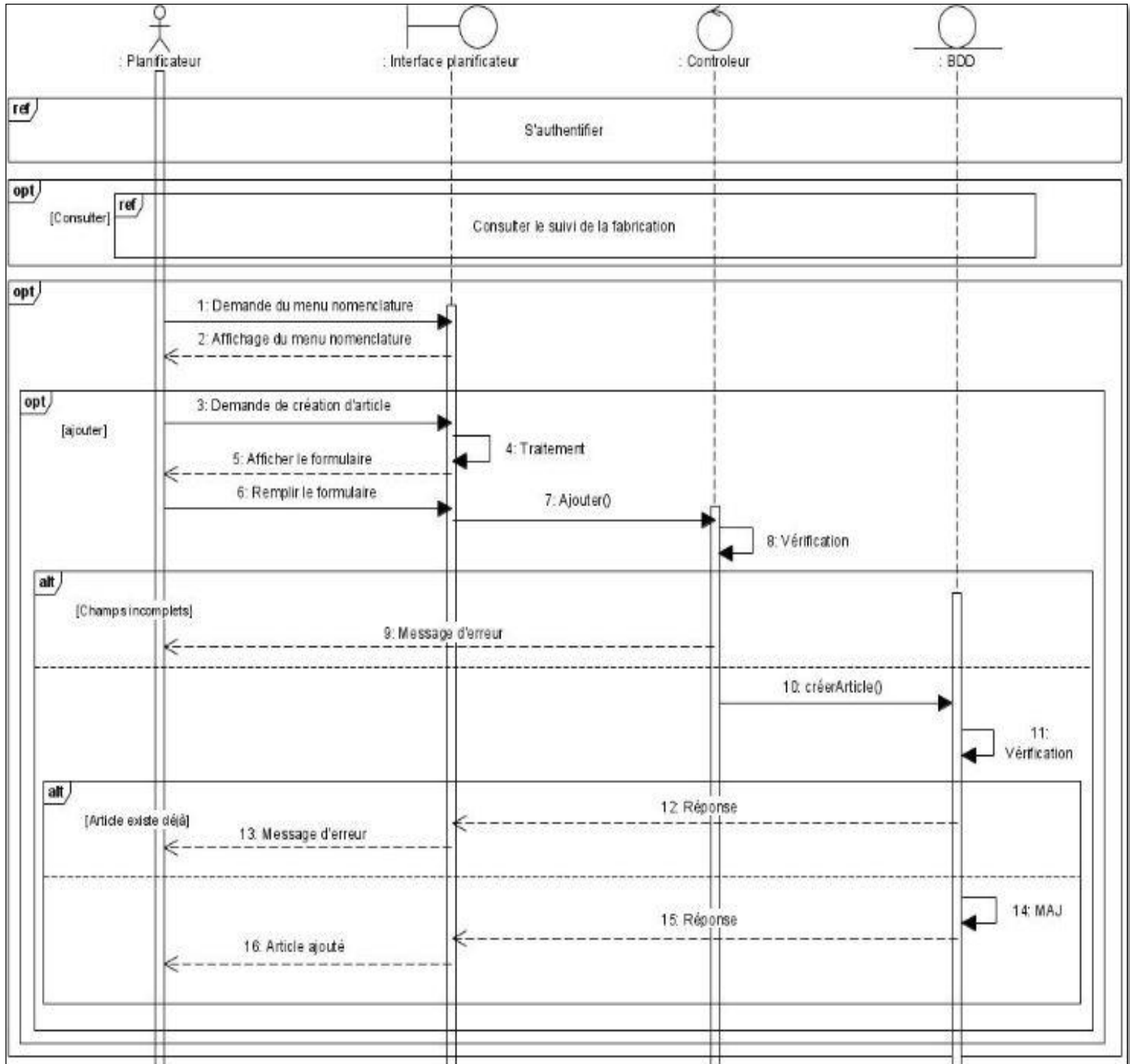
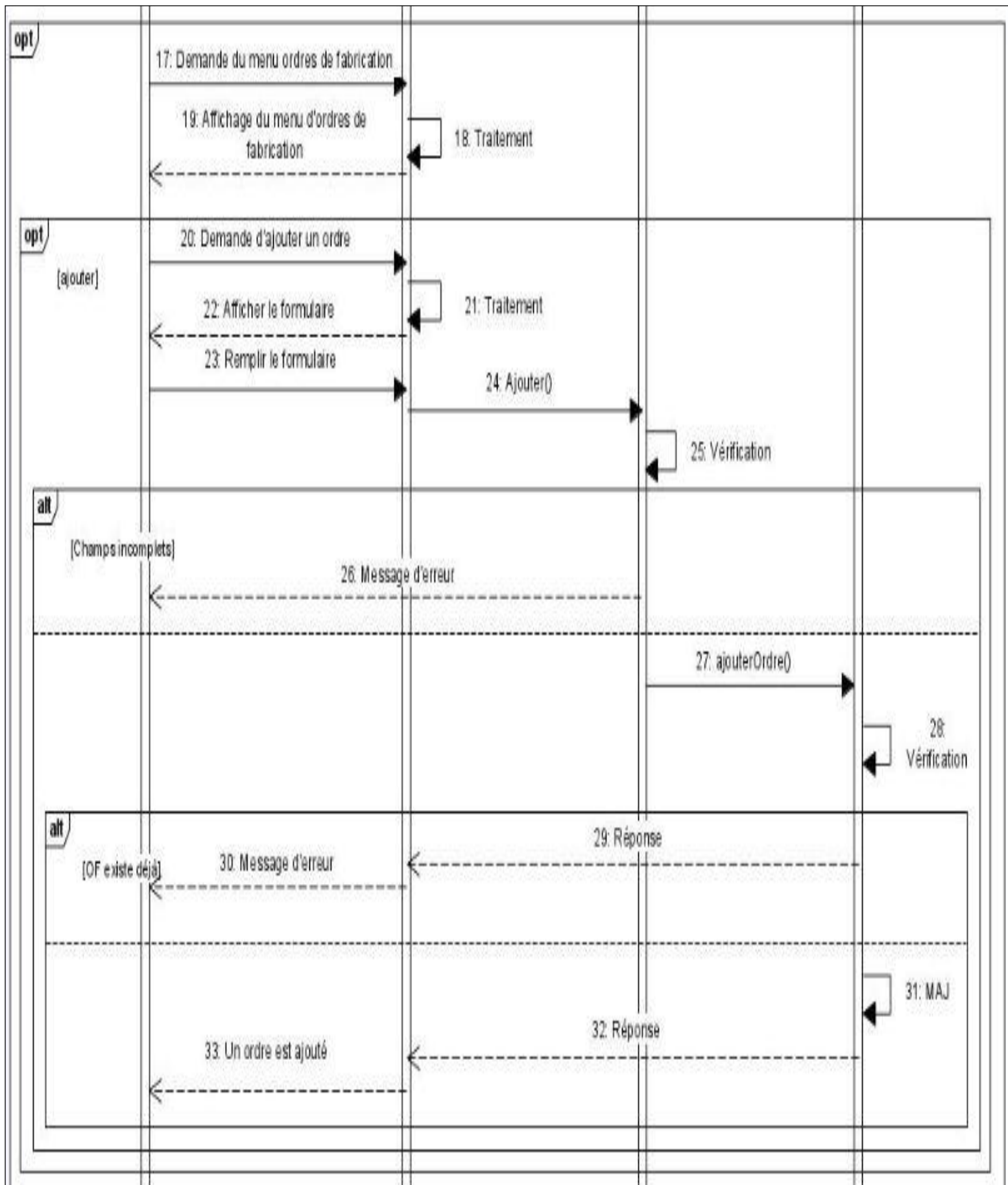


Figure 4.5 : Diagramme de séquence associé à « Inventaire de production ».

2.5. Diagramme de séquence associé au « Planificateur »

Dans ce diagramme de séquence, le planificateur doit d'abord s'authentifier, ce dernier peut établir trois cas d'utilisation différents, le premier cas d'utilisation est le suivi de fabrication, le deuxième est le cas d'utilisation d'ajout des articles dans l'ongle de la nomenclature, en dernier le cas d'utilisation de gestion des ordres de fabrication qui contient à son tour trois autres cas d'utilisation différents qui sont : l'ajout, la modification et l'annulation.





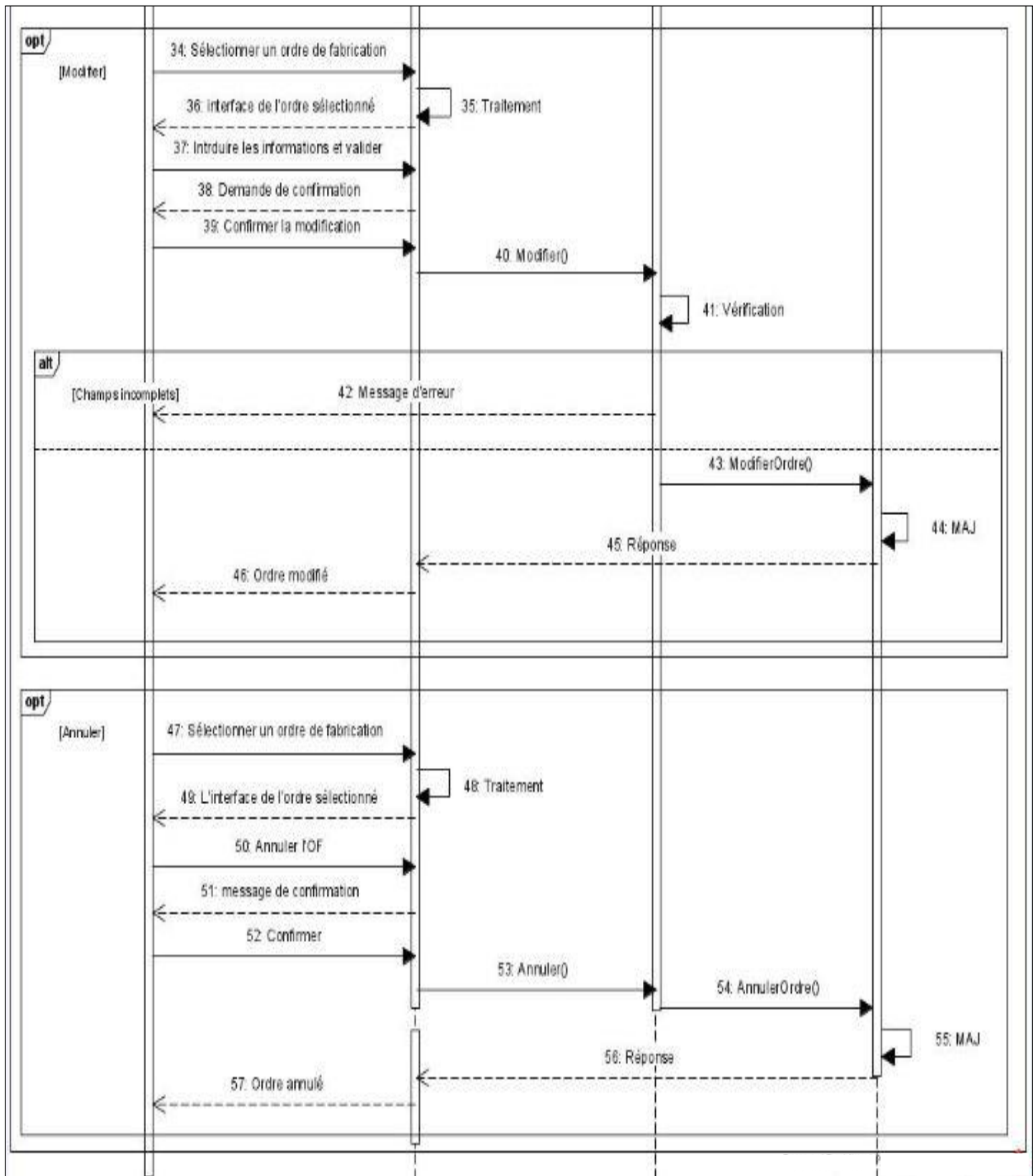


Figure 4.6 : Diagramme de séquence associé au « planificateur ».

3. Patrons de conception (Design Pattern)

Un patron de conception (plus souvent appelé design pattern) est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différents logiciels [8].

Les patrons utilisés dans notre projet sont :

3.1. Patron client/serveur à 3-tiers

Dans l'architecture à 3 niveaux, appelée architecture 3-tiers (illustrée dans la figure 4.7), il existe un niveau intermédiaire, c'est une architecture partagée entre :

- Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipé d'une interface utilisateur (généralement un navigateur web);
- Le serveur d'application (ou middleware), chargé de fournir les ressources mais faisant appel à un autre serveur;
- Le serveur de données (ou database server) qui va fournir au serveur d'application les données dont il a besoin.

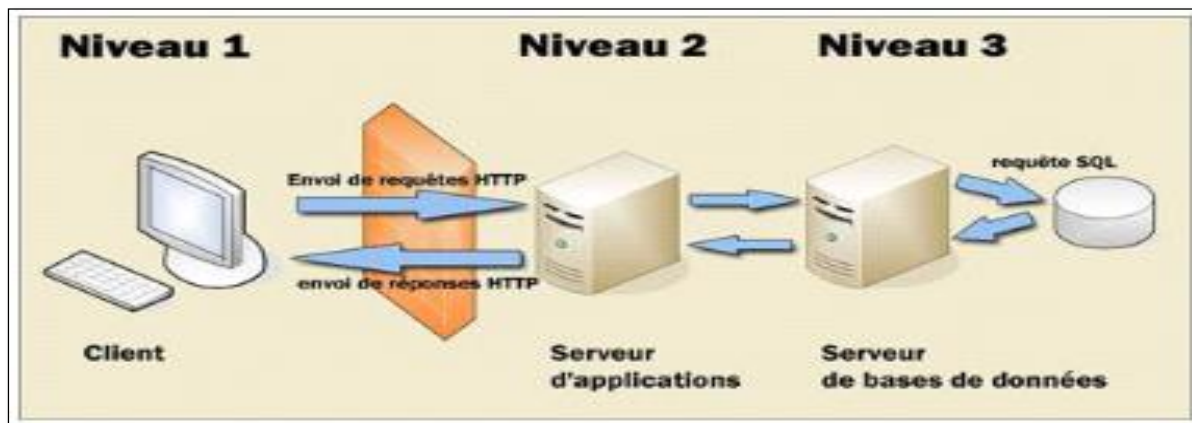


Figure 4.7 : Architecture 3-tiers.

3.2. Patron MVT (Modèle-Vue- Template)

Le MVT (Model View Template) est un modèle de conception logicielle. Il s'agit d'une collection de trois composants importants, la vue, le modèle et la Template. Le modèle aide à gérer la base de données. Il s'agit d'une couche d'accès aux données qui gère les données [29].

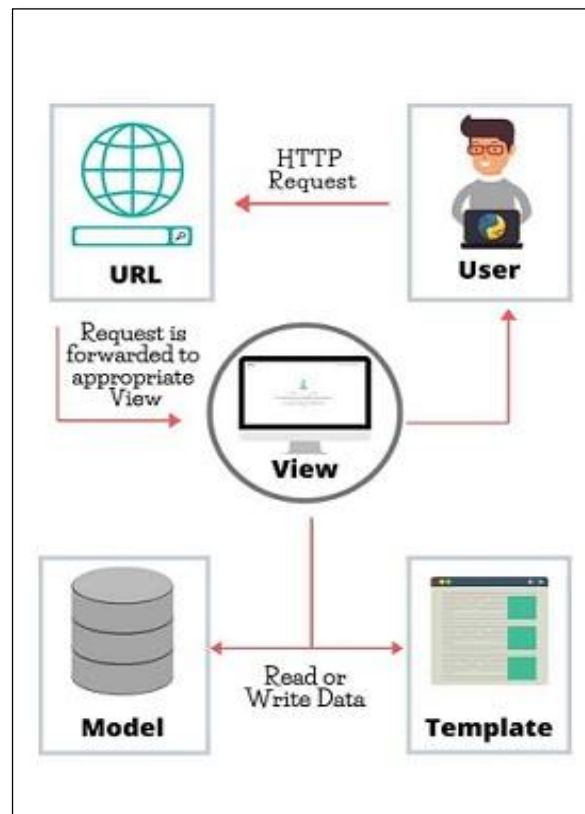


Figure 4.8 : Patron MVT (Modèle, Vue, Templates).

Ce modèle de conception impose donc une séparation en 3 couches :

- Le modèle : Le modèle agit comme les interfaces de notre système. Il est responsable de la maintenance des données. C'est la structure de données logique derrière l'ensemble de l'application et est représentée par une base de données
- La vue : La vue est l'interface utilisateur ce que vous voyez dans votre navigateur lorsque vous affichez un site Web. Il est représenté par des fichiers HTML / CSS / Javascript et Django.
- La Template : Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante [30].

4. Diagramme de classe

La figure ci-dessous nous permet de voir les classes principales dans notre système, et décrit la structure générale de ce dernier.

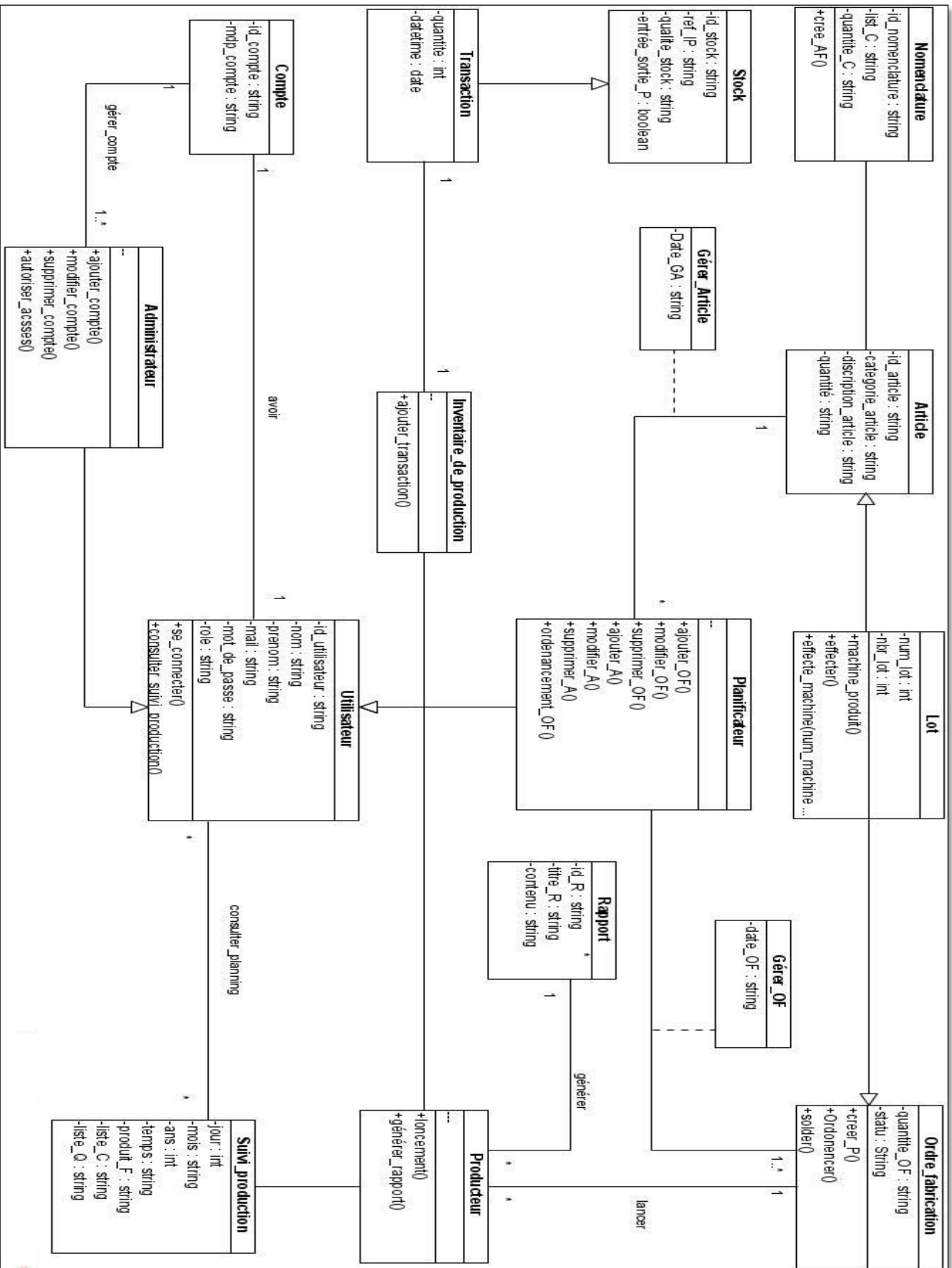


Figure 4.9 : Diagramme de classe.

4.1. Passage au modèle relationnel

Le modèle relationnel est le modèle logique de données qui correspond à l'organisation des données dans les bases de données relationnelles. Un modèle relationnel est composé de relations, appelée table. Ces tables sont décrites par des attributs ou champs. Pour décrire une relation, on indique tout simplement son nom, suivi du nom de ses attributs entre parenthèses. L'identifiant d'une relation est composé d'un ou plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée [5].

4.1.1. Règles de passage au modèle relationnel

Les règles de passage sont :

Règle 1 : Toute entité devient une relation ayant pour clé primaire son identifiant.

Règle 2 : Toute association hiérarchique (de type $[1, n]$) se traduit par une clé étrangère.

La clé primaire correspondant à l'entité père (côté n) migre comme clé étrangère dans la relation correspondant à l'entité fils (côté 1).

Règle 3 : Toute association non hiérarchique (de type $[n, n]$) devient une relation.

La clé primaire est formée par la concaténation (juxtaposition) de l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.

Règle 4 : Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes :

- a) **Décomposition par distinction :** Il faut transformer chaque sous-classe en une relation. La clé primaire de la surclasse migre dans la relation issue de la sous-classe et devient à la fois clé primaire et la clé étrangère.
- b) **Décomposition descente (push-down) :** Il faut faire migrer tous ses attributs dans la relation issue de la sous-classe.
- c) **Décomposition ascendante (push-up) :** Il faut supprimer la relation issue de la sous-classe et faire migrer les attributs dans la relation issue de la surclasse.

Règle 5 : La clé primaire des relations déduites des classes composantes doit contenir l'identifiant de la classe composite (quelles que soit les multiplicités).

4.1.2. Modèle logique des données

Voici le modèle logique des données :

Nomenclature (id_nomenclature , liste_C , quantite_C) ;
 Article(id_article , categorie_C , discription_article , quantie_A , #id_nomenclature , #id_planificateur) ;
 Lot (num_lot , nbr_lot) ;
 Ordre_fabrication (id_OF , quantite_OF , statu , #id_Producteur) ;
 Producteur (id_Producteur , nom , prenom , mail , mot_de_passe , role) ;
 Rapport (id_R , titre_R , contenu) ;
 Planificateur (id_planificateur , nom , prenom , mail , mot_de_passe , role , #id_OF) ;
 Utilisateur (id_Utilisateur , nom , prenom , mail , mot_de_passe , role , #id_compte) ;
 Suivi_production(id_SP , jour , mois , ans , temps , produit_F , liste_C , liste_Q) ;
 Machine(num_machine) ;
 Ouvrier(non_ouvrier) ;
 Administrateur(id_admin , nom , prenom , mail , mot_de_passe , role) ;
 Compte(id_compte , mdp_compte , #id_admin) ;
 Inventaire_de_production(id_IP , nom , prenom , mail , mot_de_passe , role , #id_Transaction) ;
 Transaction(id_transaction , quantite_t , datetime , ref_IP , quantite_stock , entre_sortie_P) ;
 Stock(id_stock , ref_IP , quantite_stock , entre_sortie_P) ;
 Gérer_OF(#id_Planificateur , #id_OF , date_OF) ;
 Gérer_Article(#id_article , #id_Planificateur , date_GA) ;
 consulter_palnning(id_SP, id_utilisateur , jour , mois , ans , temps , produit_F , liste_C , liste_Q , nom , prenom , mail , mot_de_passe , role) ;
 consulter_O(id_SP, nom_ouvrier , jour , mois , ans , temps , produit_F , liste_C , liste_Q) ;
 consulter_M(id_SP, num_machine , jour , mois , ans , temps , produit_F , liste_C , liste_Q) ;
 a_la_compétence(nom_ouvrier, num_machine) ;

5. Conclusion

Nous avons présenté dans ce chapitre la phase de conception via les diagrammes de séquence, qui nous ont permis de décrire de manière globale et détaillée, le fonctionnement du système afin d'en faciliter la réalisation et la maintenance. Ces derniers ont mis en avant les interactions entre les différents objets constituant notre application. Puis nous avons défini les patrons de conception qui seront utilisés lors de l'implémentation. Ensuite, nous avons conçu le

diagramme de classes donnant ainsi une vue plus structurée des éléments qui formeront la base de données liée à notre application. Dans le chapitre suivant, nous entamerons la phase d'implémentation et réalisation.

5

IMPLÉMENTATION ET RÉALISATION

1. Introduction

Nous entamons dans ce chapitre l'implémentation et la réalisation du progiciel, après l'étape de conception définie au préalable. Nous commencerons par citer et définir les différents outils de développement, ensuite, nous présenterons les différentes interfaces de notre application.

2. Outils de développement

2.1. Environnement de développement

2.1.1. Anaconda



Anaconda est une distribution Python. A son installation, Anaconda installera Python ainsi qu'une multitude de packages. Cela nous évite de nous ruiner dans les problèmes d'incompatibilités entre les différents packages.

Anaconda propose un outil de gestion de packages appelé Conda. Ce dernier permettra de mettre à jour et installer facilement les librairies dont on aura besoin pour nos développements [21].

2.1.2. Adobe PhotoShop CC



Adobe Photoshop CC est un logiciel de retouche photo, image et design conçu pour les designers professionnels, les photographes et les artistes. L'application d'imagerie et de conception fait partie du service Adobe Creative Cloud et permet aux utilisateurs de concevoir des sites Web, des applications mobiles, des affiches, des bannières et des icônes à l'aide de ses modèles faciles à

utiliser et de ses outils intuitifs. Il fournit également des outils de photographie professionnels qui aident à créer et à améliorer des images, en les transformant en œuvres d'art étonnantes [23].

2.1.3. Visual studio code

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau, disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et possède un riche écosystème d'extensions pour d'autres langages (tels que C ++, C #, Java, Python, PHP, Go) et des runtimes (tels que .NET et Unity) [22].

2.1.4. TeamViewer



TeamViewer est un Logiciel permettant, entre autres, la prise de contrôle à distance d'un ordinateur et le transfert de fichiers [17]. Nous utilisons cet outil pour enchaîner le travail pendant la conception, et pour partager les données de notre travail.

2.1.5. Visual Paradigme



Visual Paradigme For UML : permet la création des diagrammes UML et des modèles qui en sont à l'origine. Ceux-ci peuvent alors générer du code dans un langage de programmation déterminé. Il propose également la création d'autres types de diagrammes, comme celui qui permet la modélisation des bases de données pouvant, lui aussi générer des canevas d'applications basés sur des Framework et Pattern mais en plus, générer du code SQL qu'il peut ensuite déployer automatiquement dans différents environnements [4].

2.1.6. Serveurs WSGI

Un serveur WSGI (Web Server Gateway Interface) implémente le côté serveur Web de l'interface pour exécuter les applications Web Python, donc WSGI est l'interface de passerelle du serveur Web. Il s'agit d'une spécification qui décrit la manière dont un serveur Web communique avec les applications Web et comment les applications Web peuvent être chaînées pour traiter une demande [31] [32].

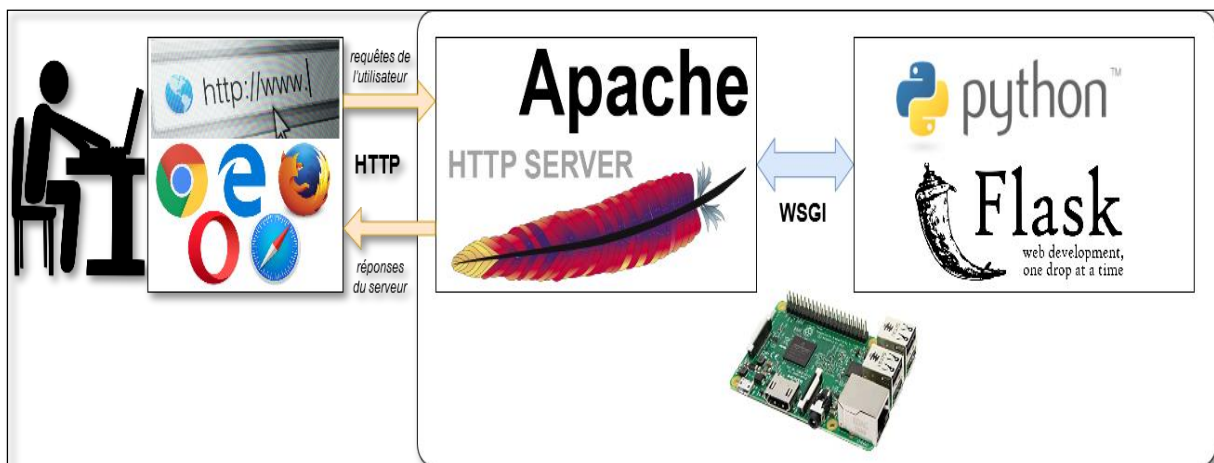


Figure 5.1 : Structure d'un serveur WEB.

2.2. Framework

2.2.1. Django



Django est un framework Web Python de haut niveau qui encourage un développement rapide et une conception propre et pragmatique [25].

Django permet de créer rapidement une application Web sans tous les problèmes d'installation ou de dépendance que vous rencontrez normalement avec d'autres frameworks. Lorsque nous créons un site Web, nous avons toujours besoin d'un ensemble similaire de composants: un moyen de gérer l'authentification des utilisateurs (inscription, connexion, déconnexion), un panneau de gestion de notre site Web, des formulaires, un moyen de télécharger des fichiers, Django vous propose des composants prêts à l'emploi [35].

Un répertoire créé par la requête « `django-admin startproject mysite` » (`mysite` est le nom du projet) vas vous générer ces fichiers :

- Le premier répertoire racine **mysite/** (le nom données à notre projet) est un contenant pour votre projet. Son nom n'a pas d'importance pour Django.
- **manage.py** : un utilitaire en ligne de commande qui nous permet d'interagir avec ce projet Django de différentes façons.
- Le sous-répertoire **mysite/** correspond au paquet Python effectif de notre projet. C'est le nom du paquet Python que nous devons utiliser pour importer ce qu'il contient (par ex. **mysite.urls**).
- **mysite/__init__.py** : un fichier vide qui indique à Python que ce répertoire doit être considéré comme un paquet.
- **mysite/settings.py** : réglages et configuration de ce projet Django.
- **mysite/urls.py** : les déclarations des URL de ce projet Django, une sorte de « table des matières » de notre site Django.
- **mysite/wsgi.py** : un point d'entrée pour les serveurs Web compatibles WSGI pour déployer votre projet.

2.2.2. Bootstrap



Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source, ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils et un design responsive pour créer un site facilement qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il

fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de framework un "Front-End Framework" [39].

2.3. Langages de développement

2.3.1. JavaScript



JavaScript désigne un langage de développement informatique, et plus précisément un langage de script orienté objet. On le retrouve principalement dans les pages Internet. Il permet, entre autres, d'introduire sur une page web ou HTML des petites animations ou des effets. Le langage JavaScript se distingue des langages serveurs par le fait que l'exécution des tâches est opérée par le navigateur lui-même, sur l'ordinateur de l'utilisateur, et non sur le serveur web. Il s'active donc généralement sur le poste client plutôt que côté serveur [20].

2.3.2. HTML



L'HTML est un langage informatique utilisé sur l'Internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie HyperText Markup Language, ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage en cascade (CSS). HTML est initialement dérivé du Standard Generalized Markup Language (SGML) [18].

2.3.3. CSS



CSS est le diminutif de Cascading Style Sheets, ou feuilles de styles en cascade. Le CSS a été créé en 1996 et a pour rôle de mettre en forme du contenu en lui appliquant ce qu'on appelle des styles [19].

2.3.4. Python



Python est un langage portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) une approche modulaire et orientée objet de la programmation. Python est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles [3].

2.4. Bibliothèque

2.4.1. SQLite



SQLite est une bibliothèque en langage C qui implémente un petit moteur de base de données SQL complet, rapide, autonome, haute fiabilité et complet. SQLite est le moteur de base de données le plus utilisé au monde. SQLite est intégré à tous les téléphones mobiles et à la plupart des ordinateurs et est intégré à d'innombrables autres applications que les gens utilisent tous les jours [24].

2.4.2. Pandas



Pandas est une bibliothèque open source conçue principalement pour travailler facilement et intuitivement avec des données relationnelles ou étiquetées. Il fournit diverses structures de données et opérations pour manipuler des données numériques et des séries chronologiques. Cette bibliothèque est construite au-dessus de la bibliothèque NumPy. Pandas est rapide et offre des performances et une productivité élevées pour les utilisateurs [27].

2.4.3. jQuery



jQuery est une bibliothèque JavaScript rapide, petite et riche en fonctionnalités. Il simplifie considérablement la traversée et la manipulation de documents HTML, la gestion des événements ainsi que l'animation. jQuery simplifie considérablement la programmation JavaScript [33].

2.4.4. Mappers objet-relationnel (ORM)

Un mappeur objet-relationnel (ORM) est une bibliothèque de codes qui automatise le transfert des données stockées dans les tables de bases de données relationnelles vers des objets plus couramment utilisés dans le code d'application.

Les ORM fournissent une abstraction de haut niveau sur une base de données relationnelle qui permet à un développeur d'écrire du code Python au lieu de SQL pour créer, lire, mettre à jour et supprimer des données et des schémas dans leur base de données. Les développeurs peuvent utiliser le langage de programmation avec lequel ils sont à l'aise pour travailler avec une base de données au lieu d'écrire des instructions SQL ou des procédures stockées [35].

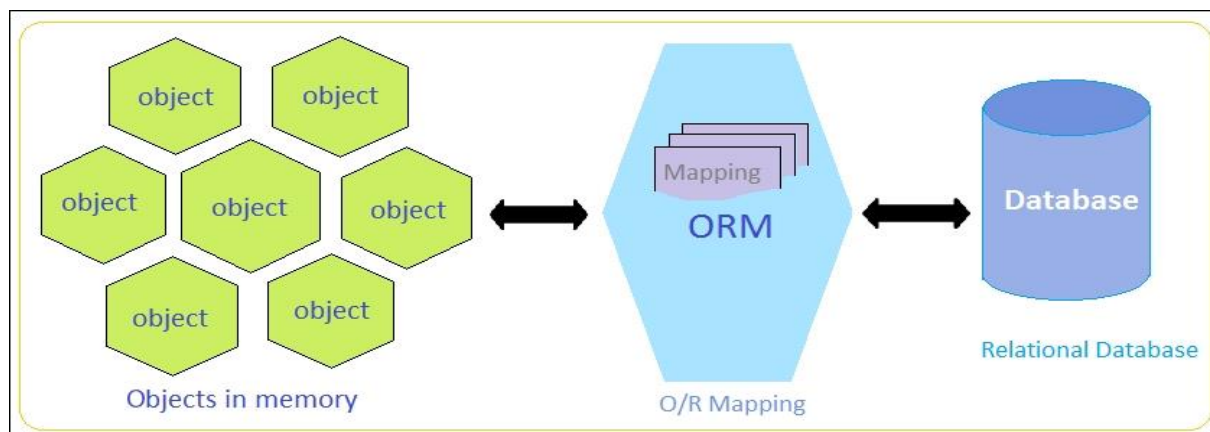


Figure 5.2 : Relation d'un mappeur objet-relationnel.

2.5. Package NumPy



NumPy est un package de traitement de tableau à usage général. Il fournit un objet de tableau multidimensionnel de hautes performances et des outils pour travailler avec ces tableaux. C'est le package fondamental pour le calcul scientifique avec Python. Il peut également être utilisé comme un conteneur multidimensionnel efficace de données génériques [34].

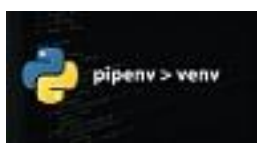
2.6. Module

2.6.1. PIP



PIP est un système de gestion de packages utilisé pour installer et gérer des packages logiciels écrits en Python. Il signifie «programme d'installation préféré» ou «Pip Installs Packages». PIP pour Python est un utilitaire pour gérer les installations de packages PyPI à partir de la ligne de commande [28].

2.6.2. venv



Le module venv permet de créer des "environnements virtuels" légers avec leurs propres dossiers site, optionnellement isolés des dossiers site système. Chaque environnement virtuel a son propre binaire Python (qui correspond à la version du binaire qui a été utilisée pour créer cet environnement) et peut avoir sa propre liste de paquets Python installés dans ses propres dossiers site [26].

3. Présentation de l'application

3.1. Présentation du Logo

Nous avons choisi le nom TAKARA GPAO pour l'application, TAKARA veut dire trésor en langue japonaise.

Et pour le slogan, nous avons préféré la phrase suivante « la puissance et la simplicité aux services de la production » parce que chaque GPAO est considéré puissant et simple et il est mis aux services de la production.



Figure 5.3 : Logo de notre application.

3.2. Présentation des interfaces

3.2.1. Interface d'authentification

Ceci est l'aperçu de l'interface d'authentification permettant aux différents utilisateurs d'avoir accès au contenu de l'application et ce selon la fonction occupée dans le service.



Figure 5.4 : Interface d'authentification.

3.2.2. Interface de TAKARAGPAO :

La figure 5.5 représente l'interface d'accueil de notre progiciel dont tous les utilisateurs accèdent selon leurs rôles dans l'entreprise.

The screenshot shows the TAKARAGPAO dashboard. On the left is a dark sidebar with the following menu items: GENERAL (Tableau de bord), MÉTHODES (Nomenclatures, Ordonnancement), INVENTAIRE (Gestion de l'inventaire), and OPÉRATIONS (Lancement, Suivi de production). At the bottom of the sidebar, it says 'Connecté en tant que: {{django.contrib.auth.username}}'. The main content area is titled 'Tableau de bord' and contains several sections: a 'Planning' section, three colored cards for 'Ofs planifiés' (blue), 'Ofs en instance' (red), and 'Ofs faits' (green), and a 'Statistiques mensuelles' section. Below these is a 'Suivi de l'inventaire' section with a search bar, a table with columns 'Ref article', 'Description', 'quantité', and 'Besoin', and a pagination control showing 'entrées 1 à 10 de {datatable.lenght}'. The footer of the dashboard says 'Copyright © Takara 2020'.

Figure 5.5 : Interface de TAKARAGPAO.

3.2.3. Interface administrateur

Cette interface est dédiée uniquement à l'administrateur afin de gérer les utilisateurs, ce dernier peut ajouter, modifier ou supprimer un utilisateur, il peut aussi bloquer ou autoriser un utilisateur déjà bloqué, ainsi pour effectuer ses différentes opérations il peut retrouver un utilisateur donné en effectuant une recherche.

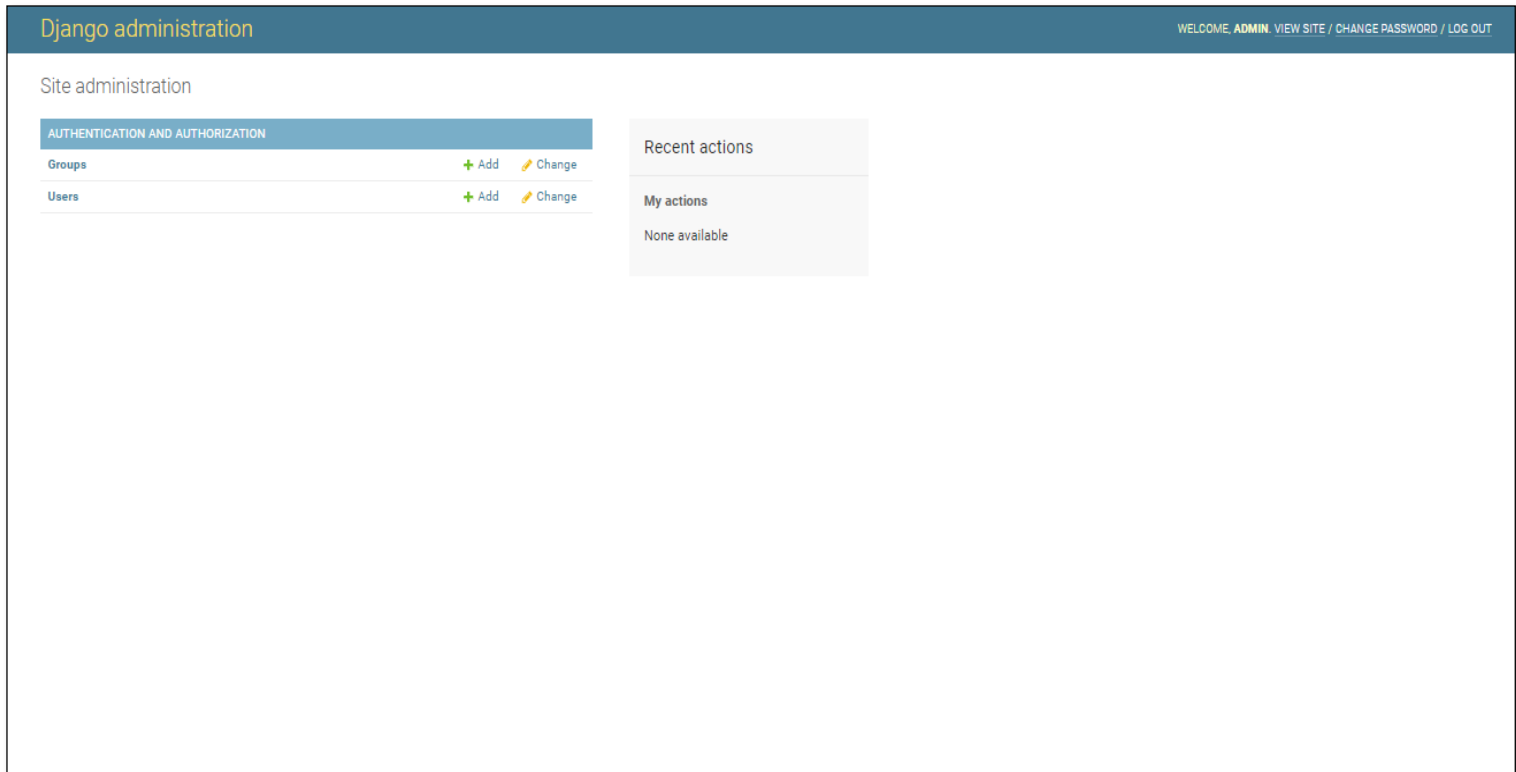


Figure 5.6 : Interface de l'administrateur.

3.2.4. L'interface de création d'un nouvel utilisateur

En cliquant sur le bouton ajouter, l'administrateur pourra ajouter un nouvel utilisateur, en remplissant et validant les champs de l'interface ci-dessous.

Django administration WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Authentication and Authorization > Users > Add user

Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username:
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:
Your password can't be too similar to your other personal information.
 Your password must contain at least 8 characters.
 Your password can't be a commonly used password.
 Your password can't be entirely numeric.

Password confirmation:
Enter the same password as before, for verification.

Figure 5.7 : Interface de création d'un nouvel utilisateur.

4. Conclusion

Dans le présent chapitre nous avons abouti à un système qui prend en charge d'une façon générale la majorité des fonctions définies. Nous avons présenté l'environnement de développement de notre application TAKARAGPAO. Nous avons terminé par la présentation des fonctionnalités de notre application à travers ses différentes interfaces.

CONCLUSION GÉNÉRALE

En guise de conclusion, le travail réalisé, et qui consiste à réaliser un système de gestion de production assistée par ordinateur, à contribuer à améliorer nos compétences dans divers domaines tout en enrichissant nos connaissances en conception, programmation et modélisation, au travers de l'utilisation de nouvelles technologies et de nouveaux outils, environnements de développement telles Anaconda, le serveur WSGI, les bibliothèques Pandas, jquery et mappers objet-relationnel, le framework Django et bootstrap, le package Numty, les modules PIP et venv.

Cette expérience nous a permis de découvrir le domaine de la GPAO totalement inconnu pour nous auparavant.

La mise en pratique de nos compétences et connaissances communes nous a permis d'implémenter cette application d'une manière simple et compréhensible, cette dernière peut être améliorée en ajoutant des fonctionnalités pour avoir un système d'approvisionnement et une PLM (Product lifecycle management) plus grande que celle qu'on a fait avec des gammes de production, des bons de commandes, et une maintenance de gestion de panne ressource consommé, ou encore analyser des modules de défaillance et de causes et faire un Reporting c'est-à-dire le calcul TRS (Time Reporting System).

En effet, ce travail étant un essai, n'est donc pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à toutes les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer davantage cette initiative.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] J. Gabay, D. Gabay, Uml 2 analyse et conception, dunod, paris, 2008.
- [2] R. P. Franck Vallée, UML 2 en action, de l'analyse des besoins à la conception. 4^{ème} édition, eyrolles, 2007.
- [3] G. Swinnen, apprendre à programmer avec python 3, eyrolles 2012.
- [4] L. Debrouwer et Fien Ven Der Heyde, UML 2 modélisation des objets, ENI 2010.
- [5] R. Pascal and V. Franck, UML2 en action, de l'analyse des besoins en action, 4^{ème} édition.
- [6] « <https://www.clipindustrie.com/clipper-erp-gpao/gpao> » consulté le 19/06/2020.
- [7] « <https://www.cci-numerique-moselle.fr/la-gestion-de-production-assistee-par-ordinateur-gpao/> » consulté le 22/06/2020.
- [8] « <https://docplayer.fr/489716-Lecon-11-gpao-l-objectif-principal-de-la-lecon-est-de-connaître-et-de-comprendre-les-principales-fonctionnalités-d-un-progiciel-de-gpao-d-un-erp.html> » consulté le 22/06/2020.
- [9] « <https://fr.slideshare.net/arou85/parte1-gpao>. » consulté le 22/06/2020.
- [10] « http://www.synaxiom.com/solutions_pour_les_entreprises/solutions_d_applications/erp.html » consulté le 23/06/2020.
- [11] « <https://www.aloer.fr/glossary/gpao> » consulté le 22/06/2020.
- [12] « <https://www.sage.com/fr-fr/blog/glossaire/gestion-de-la-production-gpao-definition> » consulté le 22/06/2020.
- [13] « <http://docplayer.fr/632257-These-proposition-d-un-cadre-de-reference-pour-la-conception-et-l-exploitation-d-un-progiciel-de-gestion-integre.html>. » consulté le 22/06/2020.
- [14] « <https://www.clipindustrie.com/clipper-erp-gpao/gpao>. » consulté le 23/06/2020.
- [15] « <https://lebonlogiciel.com/articles/top-5-des-logiciels-gpao-sur-le-marche?id=72>. » consulté le 23/06/2020.
- [16] « <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation#L2-2-2>. » consulté le 23/06/2020.
- [17] <http://www.infonitec.com/definition-informatique-telecom/> definition-informatique-telecom.php?id=1883. » consulté le 01/07/2020.
- [18] « <http://glossaire.infowebmaster.fr/html>. » consulté le 01/07/2020.
- [19] « <http://pierre-giraud.com/html-css/courscomplet/html-css-definition-role.php> »

- consulté le 01/07/2020.
- [20] « <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering.1203585-javascript>. » consulté le 25/07/2020.
- [21] « <https://mrmint.fr/installer-environnement-python-machine-learning-anaconda>. » consulté le 01/07/2020.
- [22] « <https://code.visualstudio.com/Docs>. » consulté le 01/07/2020.
- [23] « <https://reviews.financesonline.com/p/adobe-photoshop-cc/#:~:text=Adobe%20Photoshop%20CC%20is%20a%20photo%2C%20image%2C%20and,icons%20through%20the%20aid%20of%20its%20easy-to-use%20templates>. » consulté le 01/07/2020.
- [24] « <https://www.sqlite.org/index.html>. » consulté le 01/07/2020.
- [25] « <https://www.djangoproject.com>. » consulté le 01/07/2020.
- [26] « <https://docs.python.org/fr/3/library/venv.html>. » consulté le 25/07/2020.
- [27] « <https://www.geeksforgeeks.org/introduction-to-pandas-in-python/?ref=lbp>. » consulté le 18/07/2020.
- [28] « <https://phoenixnap.com/kb/install-pip-windows>. » consulté le 18/07/2020.
- [29] « <https://www.javatpoint.com/django-mvt>. » consulté le 22/07/2020.
- [30] « <https://www.geeksforgeeks.org/django-project-mvt-structure/?ref=lbp>. » consulté le 22/07/2020.
- [31] « <https://wsgi.readthedocs.io/en/latest/what.html>. » consulté le 24/07/2020.
- [32] « <https://www.fullstackpython.com/wsgi-servers.html>. » consulté le 24/07/2020.
- [33] « <https://jquery.com>. » consulté le 25/07/2020.
- [34] « <https://www.geeksforgeeks.org/python-numpy/?ref=lbp>. » consulté le 20/07/2020.
- [35] « <https://www.fullstackpython.com/object-relational-mappers-orms.html>. » consulté le 26/07/2020.
- [36] « <http://users.polytech.unice.fr/~hugues/GL/chapitre2.pdf> », consulté le 02/07/2020.
- [37] « <https://www.google.com/url?q=http://gestion-production.univ-lyon1.fr/webapp/website/website.html?id%3D3301179%26pageId%3D220666&sa=D&source=hangouts&ust=1601843024282000&usg=AFQjCNFZrCz2wOOrsiuFTn4QxZy5EZpA> », consulté le 27/07/2020.
- [38] « <http://www.toupie.org/Dictionnaire/Production.htm> » consulté le 29/07/2020.
- [39] « <https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/> » consulté le 26/07/2020.

RÉSUMÉ

Au cours de ce mémoire, nous avons conçu et réalisé une application web dédiée à la gestion de production assistée par ordinateur de qualité (GPAO). Cette application web permet de faciliter le travail du gérant en automatisant la gestion du stock, la gestion des articles et les ordres de fabrication. Elle permet aussi la création un planning pour ordonnancer et lancer les ordres de fabrication.

Ce travail a été réalisé en utilisant le processus de développement “Processus Unifier” (UP) et le langage de modélisation Unified Modeling Language (UML) afin de schématiser la solution.

Nous avons choisi de programmer l’application avec le langage HTML5, CSS3, JavaScript et Python à l’aide du framework Django. La conception de la base de données est faite avec les serveurs WSGI.

La couche sécurité de l’application a été exposée par un certain nombre de mécanismes permettant de garantir l’atteinte des objectifs de sécurité.

Mots clés : Application Web, Django, GPAO, HTML5, CSS3, Python, UP, serveur WSGI, UML.

ABSTRACT

During this report, we designed and produced a web application dedicated to quality computer-assisted production management (CAPM). This web application facilitates the work of the manager by automating stock management, item management and production orders. It also allows the creation of a schedule to schedule and launch production orders.

This work was carried out using the “Processus Unifier” (UP) development process and the Unified Modeling Language (UML) in order to schematize the solution. We have chosen to program the application with HTML5, CSS3, JavaScript and Python using the Django framework. The design of the database is made with the WSGI servers.

The security layer of the application has been exposed by a number of mechanisms to ensure that security objectives are met.

Keywords: Django, HTML5, CSS3, Python, WSGI Server, UP, UML, web application.