

# Dossier de conception

**Sujet : Gestion de Stock-Pharma**

**Réaliser par : FADIL Ghizlane**

**ECH CHARFAOUY Abdelouahad**

Encadré par :

**M. LACHGAR Mohammed**

**Cadre réservé à l'encadrant :**

Code d'identification du Candidat : .....	
<i>Nom des Validateurs</i>	Commentaires :

## TABLE DES MATIERES

<b>I. PERIMETRE DU PROJET .....</b>	<b>1</b>
1. INTRODUCTION.....	1
2. ENJEUX ET VISION DU PROJET .....	1
3. REFERENTIEL DES EXIGENCES.....	2
1.1.1. <i>Les contraintes de conception :</i> .....	2
<b>II. PLAN DE MISE EN ŒUVRE .....</b>	<b>3</b>
1. CYCLE DE VIE(MODELE EN CASCADE).....	3
<b>III. ANALYSE ET CONCEPTION .....</b>	<b>4</b>
2. IDENTIFICATION DES ACTEURS .....	4
3. DIAGRAMME DE CAS D'UTILISATION :.....	5
a. <i>Définition :</i> .....	5
4. DIAGRAMME DE CLASSE .....	13
a. <i>Définition :</i> .....	13
5. DIAGRAMME DE SEQUANCE : .....	14
a. <i>Définition :</i> .....	14
6. DIAGRAMME D'ETAT TRANSITION : .....	17
a. <i>Définition :</i> .....	17
7. DIAGRAMME D'ACTIVITE: .....	18
a. <i>Définition :</i> .....	18
8. DIAGRAMME DE DEPLOIEMENT : .....	19
a. <i>Définition :</i> .....	19
<b>IV. ARCHITECTURE APPLICATIVE.....</b>	<b>20</b>
1. CHOIX TECHNOLOGIQUES .....	21
2. OUTILS : .....	21



# I. PERIMETRE DU PROJET

## 1. INTRODUCTION

Pour notre projet nous avons opté pour le thème Gestion de Pharmacie. Notre choix a été motivé par plusieurs points et spécialement le fait d'avoir un client pour notre logiciel ce qui nous permet de nous initier à l'approche client développeur et au cycle de vie du logiciel, et nous engage à concevoir un produit fiable, robuste et répondant complètement aux besoins du client. Ce choix a été influé aussi par les outils informatiques que nous allions mettre en œuvre pour ce logiciel à savoir la conception d'une base de données, une Interface utilisateur graphique(GUI) et de toute la programmation qu'il y a derrière afin de satisfaire le cahier de charge du client et aboutir à une application simple, utile, performante, ergonomique et fiable. La conception et la mise en œuvre des bases de données constituent un volet très important de l'informatique car elles sont aujourd'hui au cœur des applications quotidiennes et du système d'information des entreprises.

## 2. ENJEUX ET VISION DU PROJET

Comme de nombreuses autres entreprises, les pharmacies sont contraintes de constituer les stocks car elles commercialisent des produits. De ce fait, afin de satisfaire au mieux la clientèle en quantité, qualité, délais,... et de réduire son coût d'exploitation, les stocks constituent un maillon essentiel pour le bon fonctionnement de la structure. D'où l'incontournable problème de gestion de stocks.

En effet, la gestion de stocks a pour but de maintenir à un seuil acceptable le niveau de service pour lequel le stock considéré existe ceci à des coûts relativement faibles. En effet, la gestion de stocks n'a pas d'objectifs absolus valables pour toutes les entreprises, pour tous les produits, pour toutes les catégories de stocks. L'objectif correspondra toujours à un contexte particulier, de plus il ne sera pas figé mais évoluera dans le temps. Dans le cas de la pharmacie Santé et Nature, elle a pour objectif la satisfaction continue de la demande de sa clientèle en produits pharmaceutiques, cela à moindre coût.

### 3. REFERENTIEL DES EXIGENCES

1.1.1. Les contraintes de conception :

- **Gestion de la vente des médicaments :**

Lorsqu'un client lui présente une ordonnance ou veut acheter un médicament précis, le pharmacien ou un de ses employés doit vérifier la disponibilité de chaque médicament dans le stock, le logiciel doit être capable d'indiquer au pharmacien la quantité de stock restante dans le stock pour chaque médicament demandé par le client ainsi que leur date de péremption. Ce n'est qu'après cette étape de vérification que le pharmacien peut effectuer la vente en saisissant dans le logiciel le nom et la quantité de médicament vendu, ce dernier doit lui retrouver le prix de chaque médicament ainsi que le montant total qui sera encaissé par le pharmacien.

- **Gestion stock :**

Lorsqu'un fournisseur lui livre les médicaments commandés, le propriétaire de la pharmacie (et non ses employés) doit implémenter la base de donnée du logiciel par le nombre de médicaments livrés, leur nom, leur type et leur date de péremption.

## II. PLAN DE MISE EN ŒUVRE

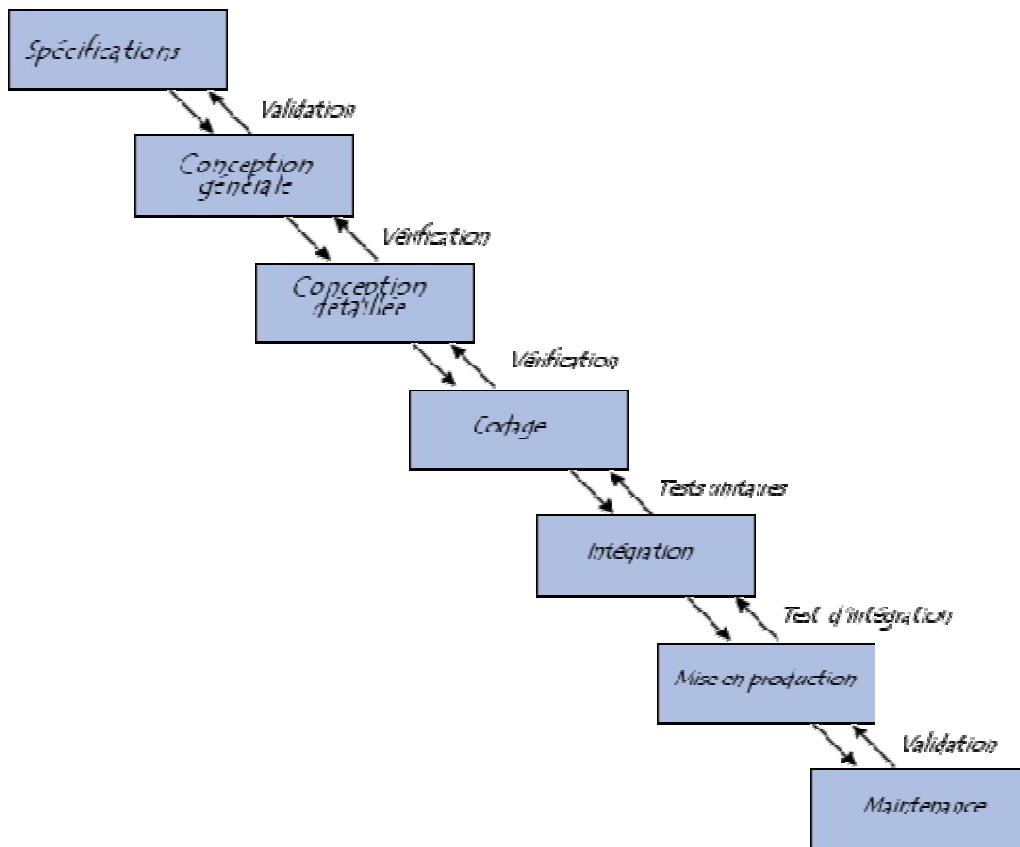
### 1. CYCLE DE VIE(MODELE EN CASCADE)

Le modèle en cascade1 est hérité de l'industrie du BTP. Ce modèle repose sur les hypothèses suivantes :

- on ne peut pas construire la toiture avant les fondations ;
- les conséquences d'une modification en amont du cycle ont un impact majeur sur les coûts en aval (on peut imaginer la fabrication d'un moule dans l'industrie du plastique).

Les phases traditionnelles de développement sont effectuées simplement les unes après les autres, avec un retour sur les précédentes, voire au tout début du cycle. Le processus de développement utilisant un cycle en cascade exécute des phases qui ont pour caractéristiques :

- de produire des livrables définis au préalable ;
- de se terminer à une date précise ;
- de ne se terminer que lorsque les livrables sont jugés satisfaisants lors d'une étape de validation-vérification.



## III. ANALYSE ET CONCEPTION

### 2. IDENTIFICATION DES ACTEURS

Acteurs	Description
Pharmacien	Possède des droits sur la gestion de stock pharmaceutique. Possède des droits sur la gestion des médicaments. Possède des droits sur la gestion des commandes et les fournisseurs. Possède des droit sur la gestion des clients et les ventes.

#### Description détaillée des cas d'utilisation

Les cas d'utilisation du système sont organisés en cinq packages :

- [1] Gestion de stock.
- [2] Gestion des médicaments.
- [3] Gestion des fournisseurs.
- [4] Gestion des commandes.
- [5]Gestion des clients et des ventes

### 3. DIAGRAMME DE CAS D'UTILISATION :

#### a. Définition :

Ils sont développés par Ivar Jacobson et permettent de modéliser des processus métiers en les découplant en cas d'utilisation.

Ce diagramme permet de représenter les fonctionnalités d'un système. Il se compose :

- d'acteurs : ce sont des entités qui utilisent le système à représenter
- les cas d'utilisation : ce sont des fonctionnalités proposées par le système

#### UC1 : Gestion de stock .

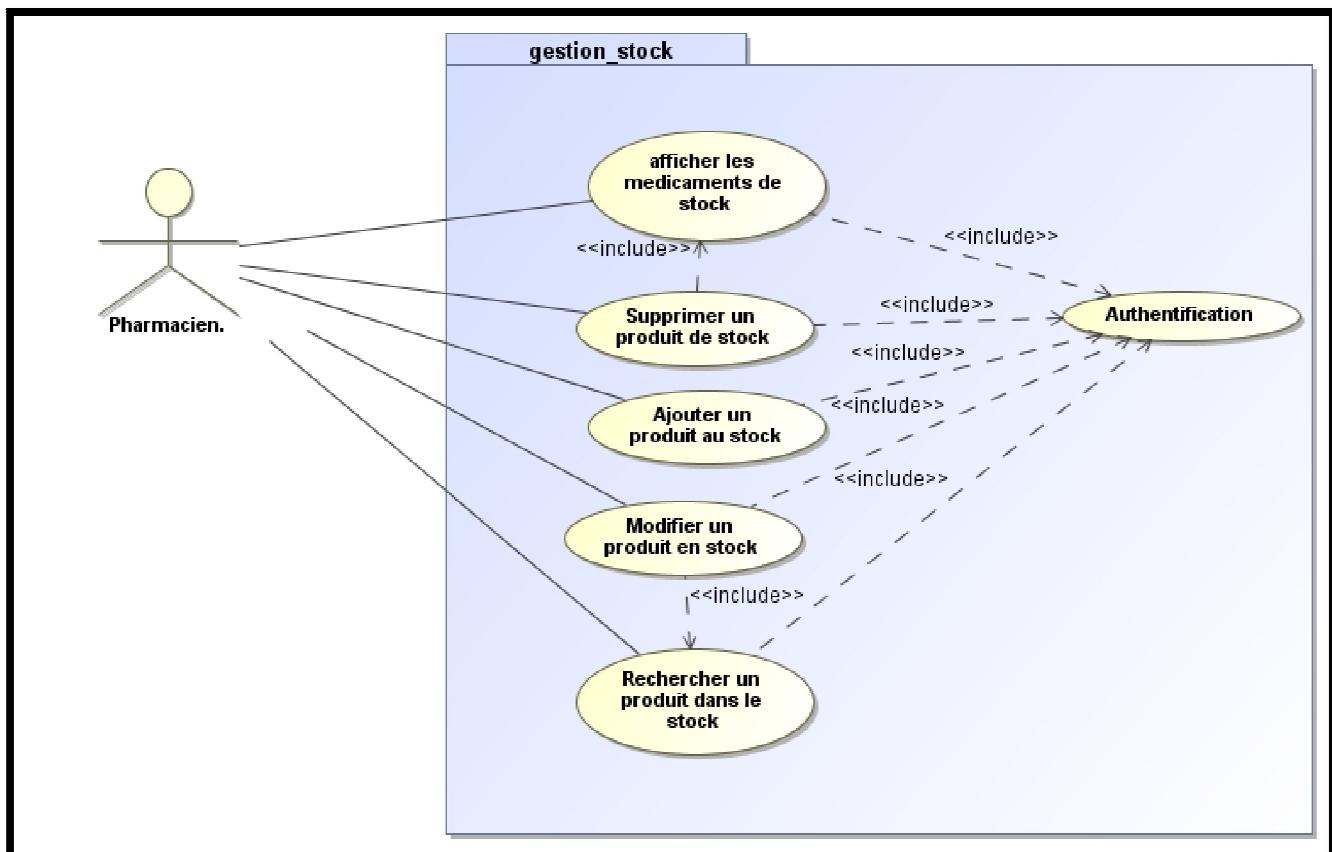


Figure 1:Use Case Gestion stock

#### Diagramme :

#### Sommaire d'identification :

**Titre : Organisation**

**But :** Ajouter, modifier ou supprimer un médicament dans le stock.

**Résumé :** Cette fonctionnalité permet au responsable de stock d'ajouter un produit au stock, de consulter la listes des produits existe en stock, de les modifier, de les chercher.

**Acteur :** Responsable de stock.

**Description Détailier :**

☞ **PRE CONDITIONS : LE RESPONSABLE DE STOCK S'AUTHENTIFIEE SUR LE SYSTEME**

☞ **DESCRIPTION DU TRAITEMENT NOMINAL : L'ACTEUR PEUT :**

1-Ajouter un produit au stock.

2-Modifier un produit en stock.

3-Supprimer un produit de stock.

4-Chercher un produit en stock.

**Exceptions :**

[Exception 1 : ChampsObligatoires] : Message d'erreur si l'un des champs obligatoires n'est pas rempli.

## UC2 : Gestion des medicaments.

Diagramme :

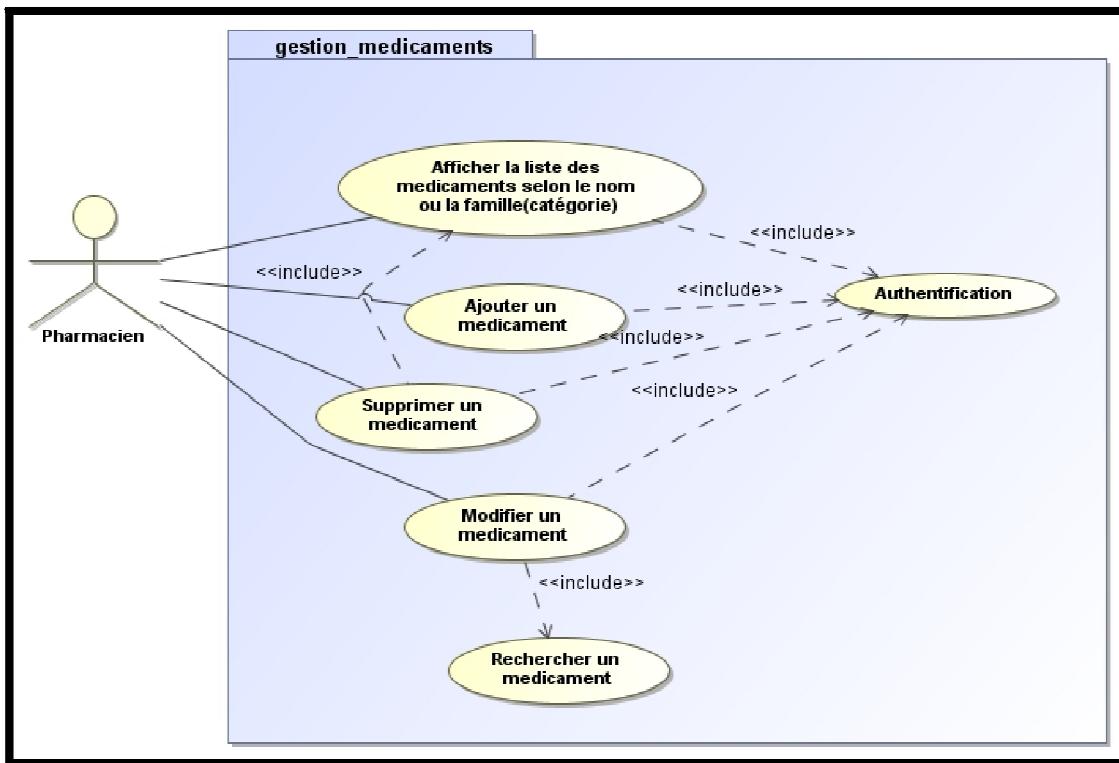


Figure 2:Use Case Gestion des médicaments

### Sommaire d'identification :

**Titre :** Gestion des médicaments

**But :** Ajouter, modifier ou supprimer un médicament.

**Résumé :** Cette fonctionnalité permet au pharmacien d'ajouter un Nouveau médicament, de consulter la listes des produits selon la catégorie de médicament, de les modifier, de les supprimer, de les chercher.

**Acteur :** pharmacien.

### Description Détailier

☞ **PRE CONDITIONS : LE PHARMACIEN S'AUTHENTIFIE SUR LE SYSTEME**

☞ **DESCRIPTION DU TRAITEMENT NOMINAL : L'ACTEUR PEUT :**

1-Ajouter un médicament.

- 2-Modifier un médicament.
- 3-Supprimer un médicament.
- 4-Chercher un médicament.
- 5-Afficher les informations d'un médicament.

**Exceptions :**

**[Exception 1 : ChampsObligatoires]** : Message d'erreur si l'un des champs obligatoires n'est pas rempli.

### UC3 : Gestion des fournisseurs.

**Diagramme :**

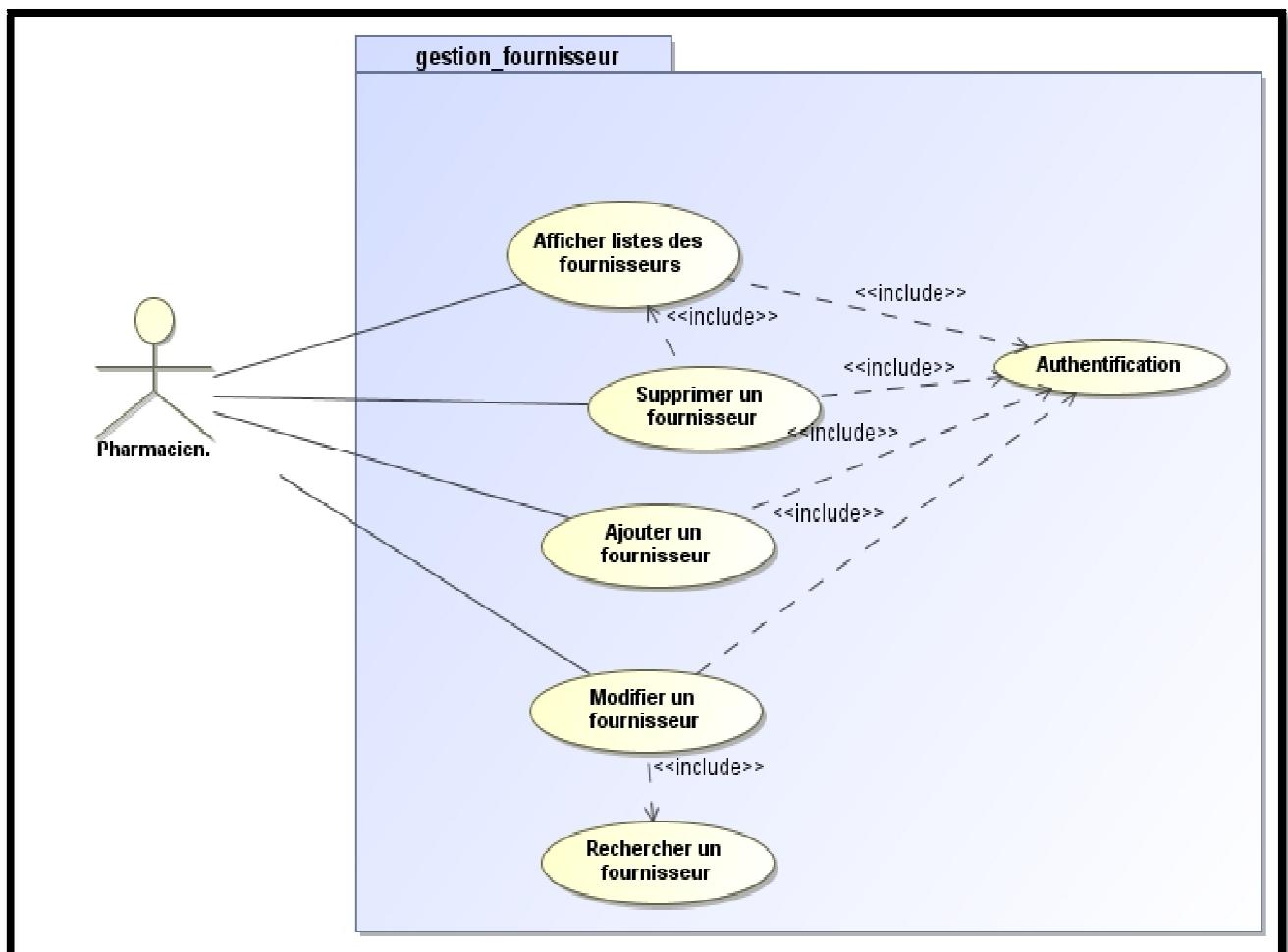


Figure 3:Use Case Gestion des fournisseurs

**Sommaire d'identification :****Titre : Gestion des fournisseur**

**But :** Ajouter, modifier ou supprimer un fournisseur.

**Résumé :** Cette fonctionnalité permet au agent de commandes d'ajouter un Nouveau fournisseur, de consulter la listes des fournisseurs, de les modifier, de les supprimer, de les chercher.

**Acteur :** Agent de commandes.

**Description Détailler**

☞ **PRE CONDITIONS : L'AGENT DE COMMANDES S'AUTHENTIFIEE SUR LE SYSTEME**

☞ **DESCRIPTION DU TRAITEMENT NOMINAL : L'ACTEUR PEUT :**

- 1-Ajouter un fournisseur.
- 2-Modifier un fournisseur.
- 3-Supprimer un fournisseur.
- 4-Chercher un fournisseur.
- 5-Afficher les informations d'un fournisseur.

☞ **EXCEPTIONS :**

**[Exception 1 : ChampsObligatoires]** : Message d'erreur si l'un des champs obligatoires n'est pas rempli.

**UC4 : Gestion des commandes.**

### Diagramme :

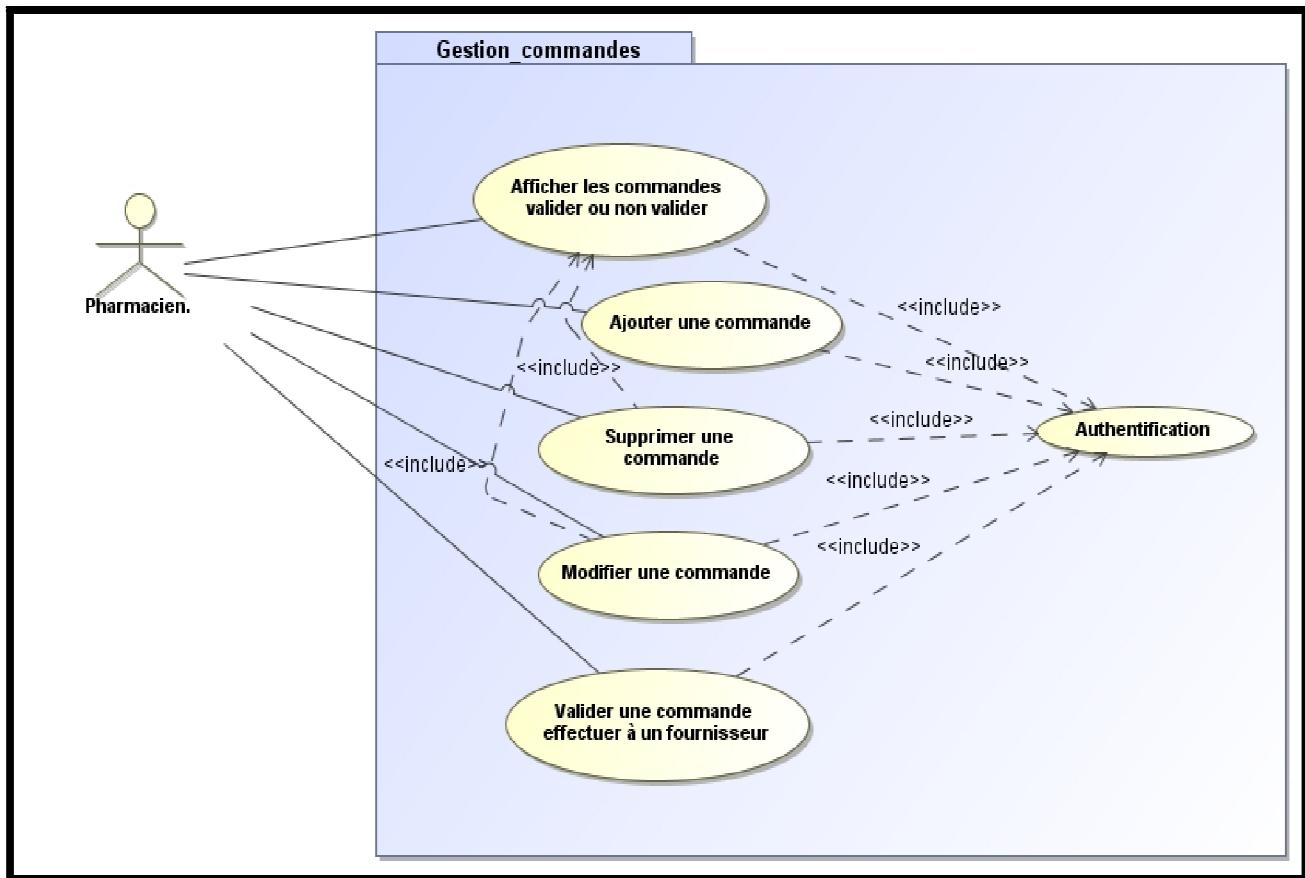


Figure 4 Use Case Gestion des commandes

### Sommaire d'identification :

**Titre :** Gestion des commandes

**But :** Ajouter, modifier ou supprimer une commande.

**Résumé :** Cette fonctionnalité permet au agent de commandes d'ajouter une commande, de consulter la liste des commandes validées et non validées, de les modifier, de les supprimer, de les chercher.

**Acteur :** Agent de commandes.

[Description Détailier](#)

☞ **PRE CONDITIONS : L'AGENT DE COMMANDES S'AUTHENTIFIEE SUR LE SYSTEME**

☞ **DESCRIPTION DU TRAITEMENT NOMINAL : L'ACTEUR PEUT :**

- 1-Ajouter une commande.
- 2-Modifier une commande.
- 3-Supprimer une commande.
- 4-Consulter les commandes valide ou non valide.

☞ **EXCEPTIONS :**

**[Exception 1 : ChampsObligatoires]** : Message d'erreur si l'un des champs obligatoires n'est pas rempli.

### UC5 : Gestion de clients et vente.

Diagramme :

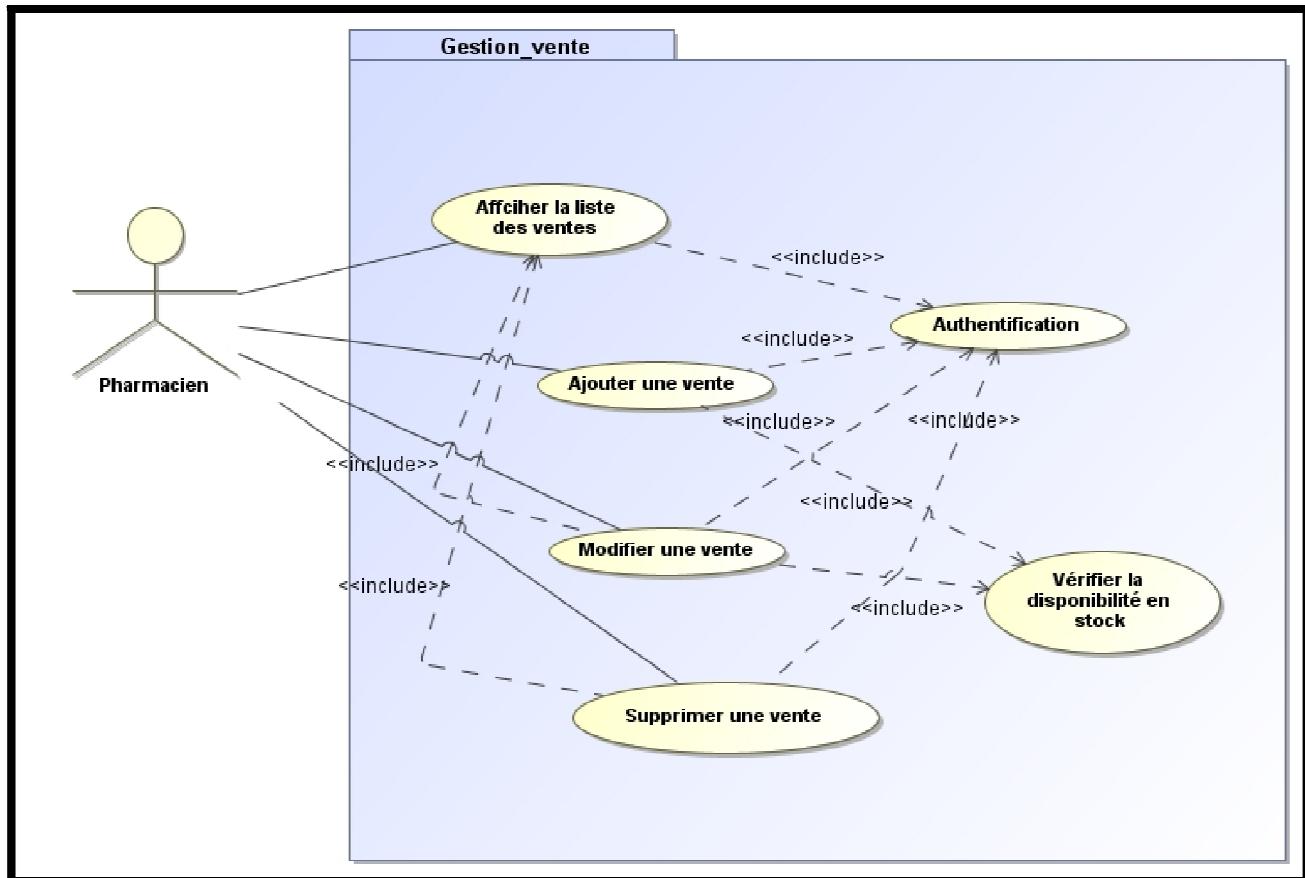


Figure 5:Use Case Gestion de vente

**Sommaire d'identification :****Titre : Gestion des ventes****But :** mettre à jour les ventes.**Résumé :** Cette fonctionnalité permet au agent commerciale d'ajouter une vente, de consulter la liste des vente, de les modifier, de les supprimer, de les chercher.**Acteur :** Agent commerciale.**Description Détailler****☞ PRE CONDITIONS : L'AGENT COMMERCIALE S'AUTHENTIFIEE SUR LE SYSTEME****☞ DESCRIPTION DU TRAITEMENT NOMINAL : L'ACTEUR PEUT :**

- 1-Ajouter vente pour un client.
- 2-Modifier une vente.
- 3-Supprimer une vente.
- 4-consulter la liste des ventes effectuées.

**☞ EXCEPTIONS :****[Exception 1 : ChampsObligatoires]** : Message d'erreur si l'un des champs obligatoires n'est pas rempli.**[Exception 2 : disponibilité]** : Message d'erreur si la quantité demander non pas disponible dans le stock.

## 4. DIAGRAMME DE CLASSE

### a. Définition :

Ce schéma représente les différentes classes : il détaille le contenu de chaque classe mais aussi les relations qui peuvent exister entre les différentes classes.

Une classe est représentée par un rectangle séparée en trois parties :

- la première partie contient le nom de la classe
- la seconde contient les attributs de la classe
- la dernière contient les méthodes de la classe

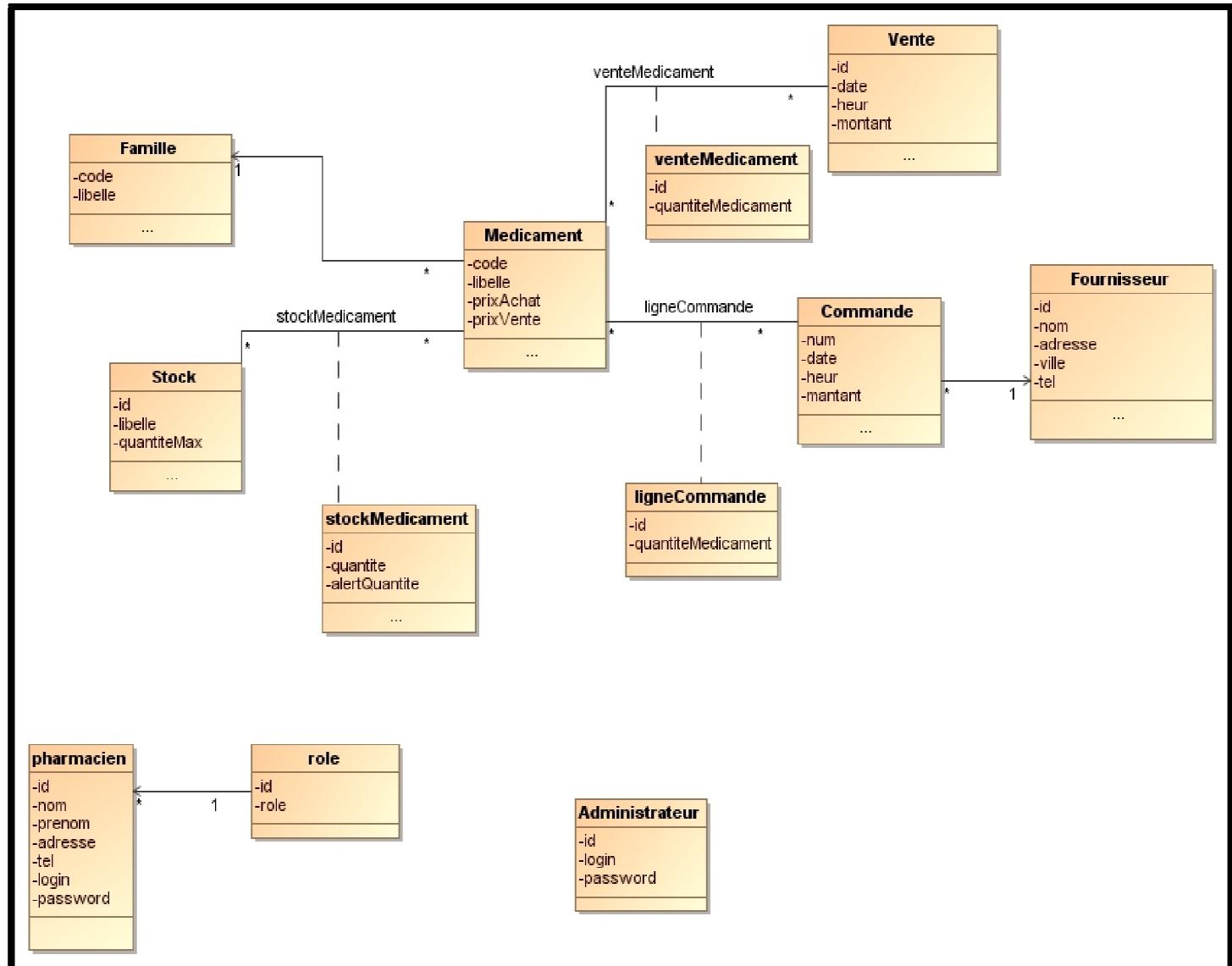


Figure 6:Diagramme de classe

## 5. DIAGRAMME DE SEQUANECE :

### a. Définition :

Le diagramme de séquence est une représentation intuitive qu'on a utilisée pour concrétiser des interactions entre nos entités.

Le diagramme de séquence décrit un scénario qui modélise une exécution particulière d'un cas d'utilisation du début jusqu'à la fin. Il correspond à une sélection d'enchaînements du cas d'utilisation.

### Diagramme de séquences dynamique de traitement de vente :

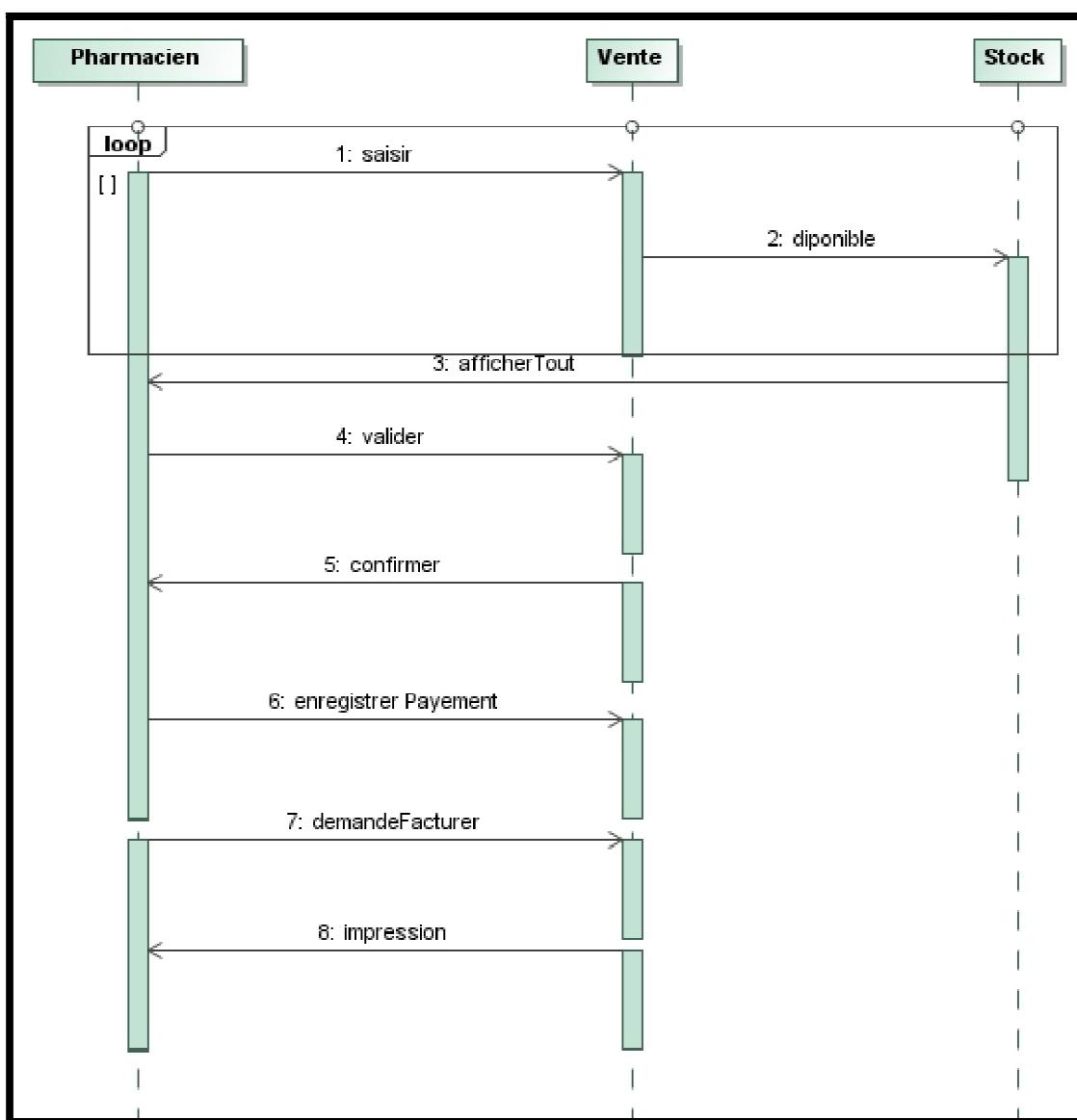


Figure 7:Diagramme de séquence d'effectuer une vente

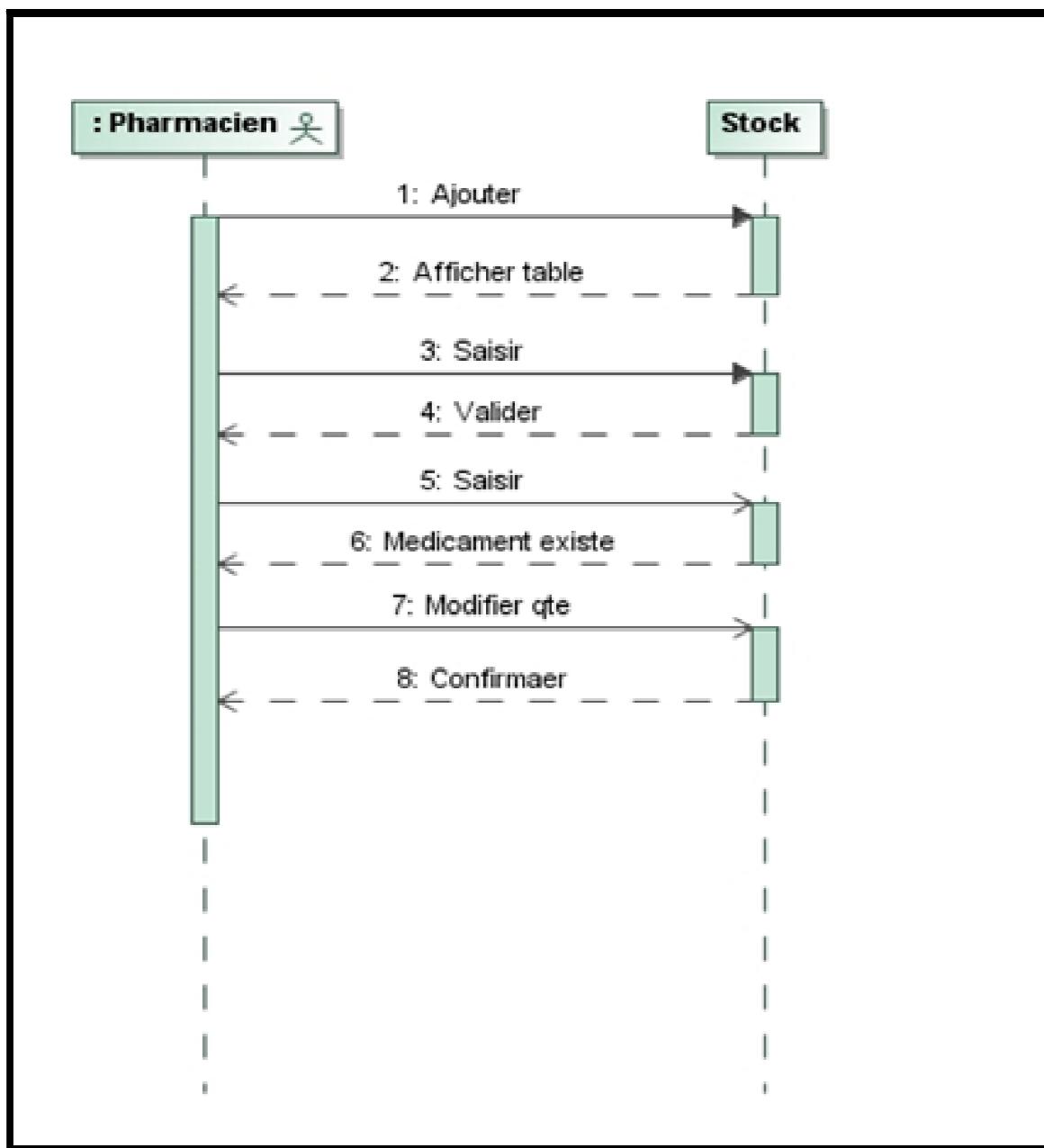
**Diagramme de séquences dynamique d'ajout dun produit en stock :**

Figure 8:Diagramme de séquence ajouter un produit au stock

**Diagramme de séquence (Commande):**

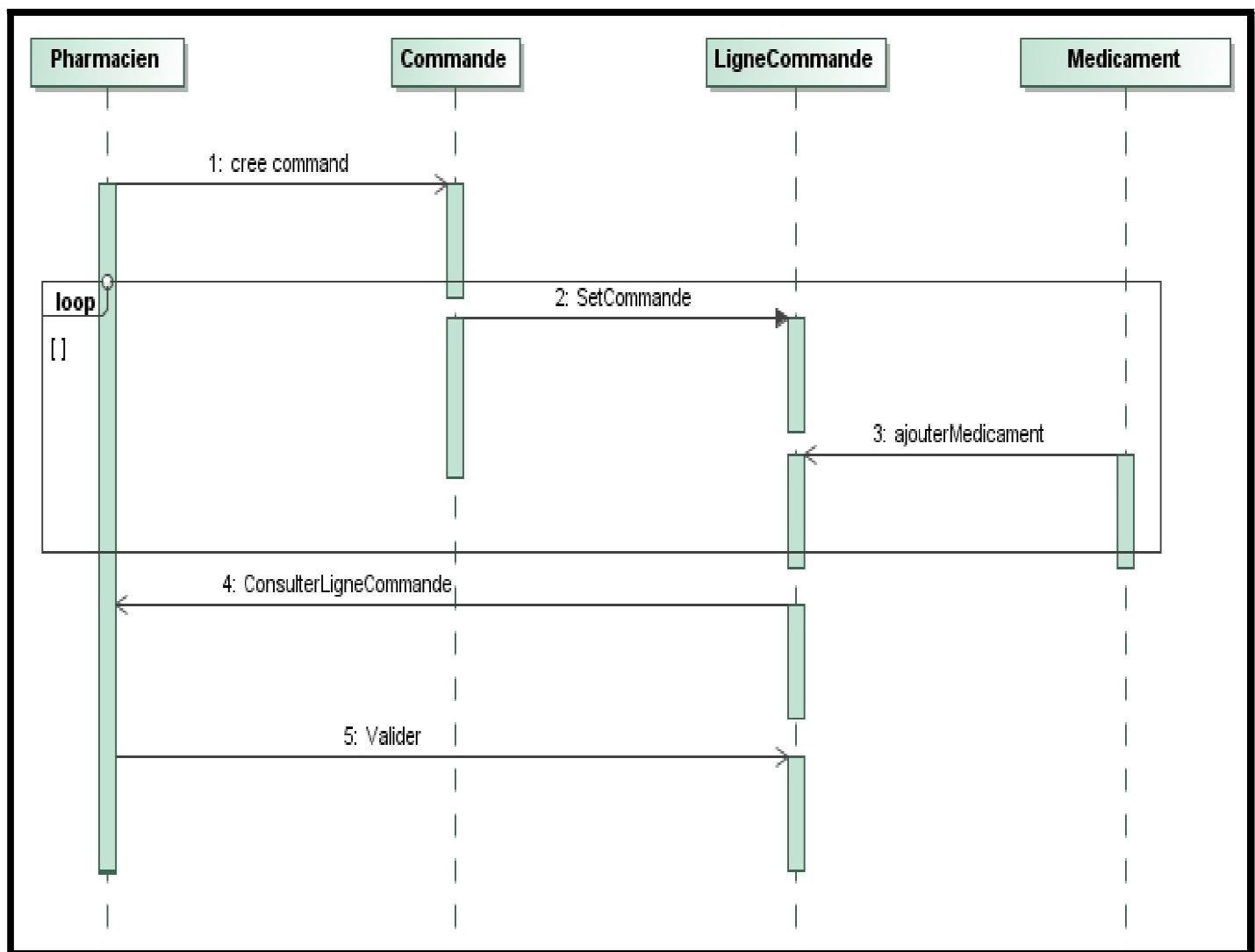


Figure 9:Diagramme de séquence commande

## 6. DIAGRAMME D'ETAT TRANSITION :

### a. Définition :

Un diagramme états-transitions est un schéma utilisé en génie logiciel pour représenter des automates déterministes. Il fait partie du modèle UML et s'inspire principalement du formalisme des statecharts et rappelle les grafcets des automates. S'ils ne permettent pas de comprendre globalement le fonctionnement du système, ils sont directement transposables en algorithme. Tous les automates d'un système s'exécutent parallèlement et peuvent donc changer d'état de façon indépendante.

#### Diagramme d'états Transition « Médicament » :

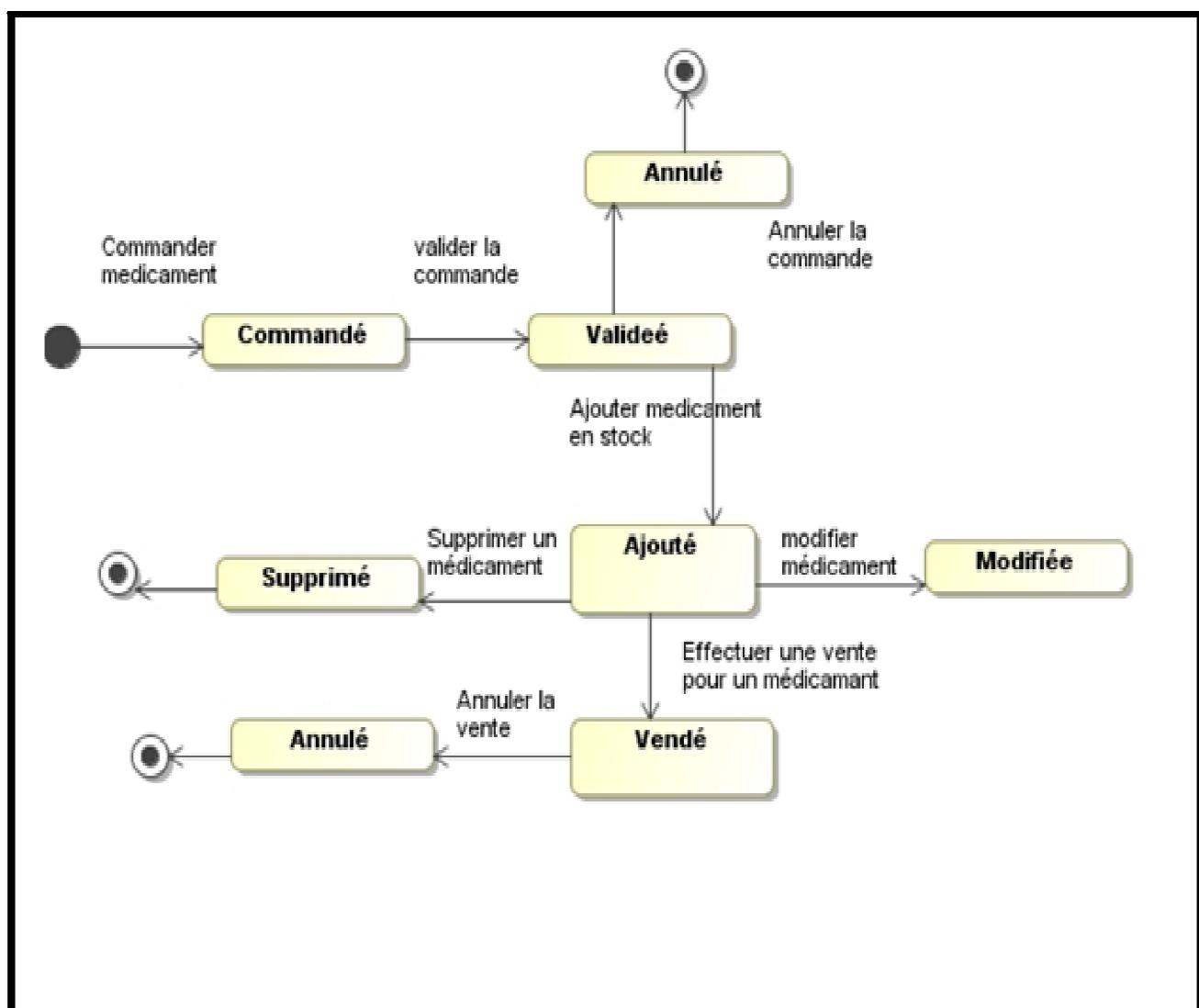


Figure 10: Diagramme d'états Transition "Médicament"

## 7. DIAGRAMME D'ACTIVITE:

### a. Définition :

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Il est recommandable pour exprimer une dimension temporelle sur une partie du modèle, à partir de diagrammes de classes ou de cas d'utilisation, par exemple.

Le diagramme d'activité est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Le diagramme d'activité est sémantiquement proche des diagrammes de communication (appelés diagramme de collaboration en UML 1), ou d'état-transitions, ces derniers offrant une vision microscopique des objets du système.

### Diagramme d'activité (Passer une commande) :

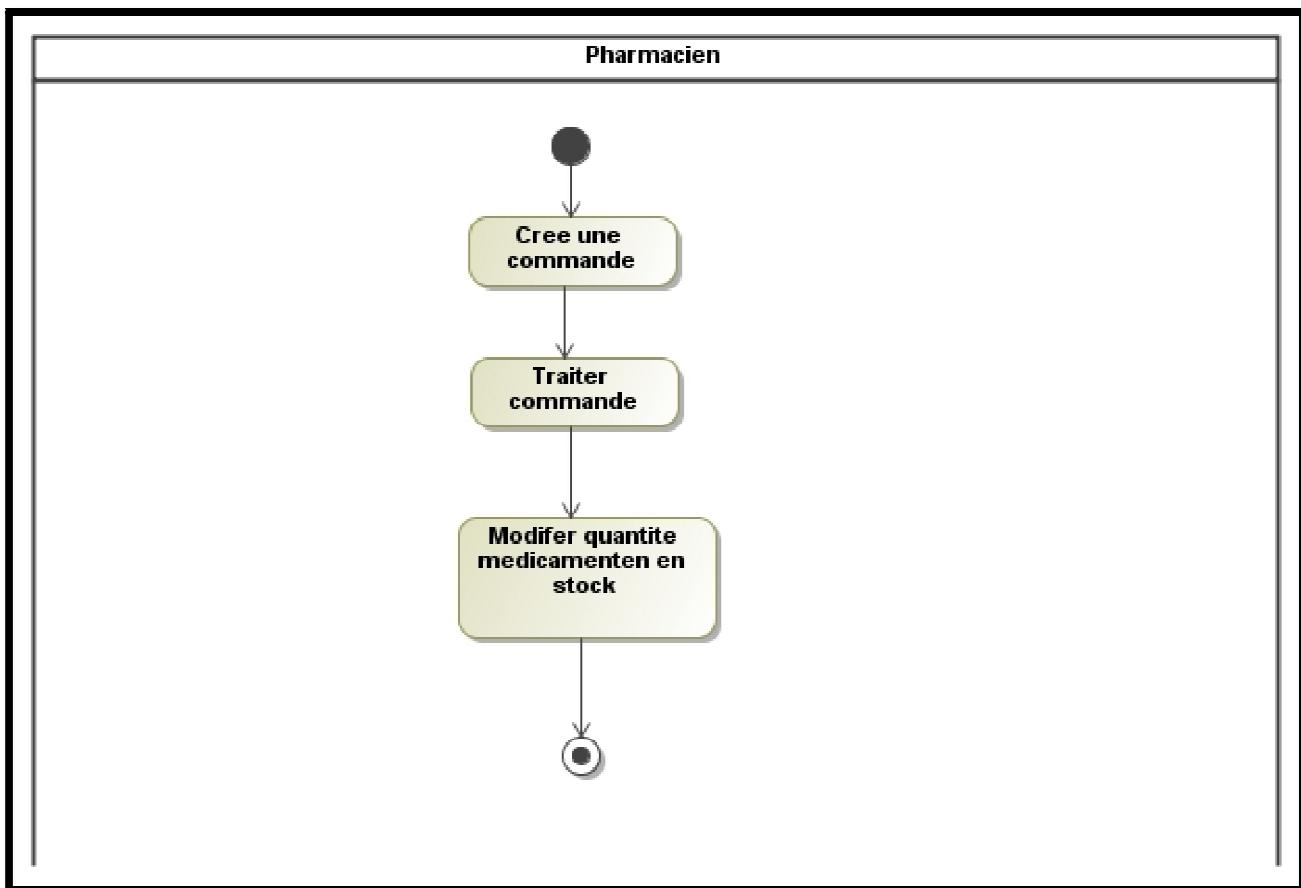


Figure 11:Diagramme d'activité "passer une commande"

## 8. DIAGRAMME DE DEPLOIEMENT :

### a. Définition :

Un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les noeuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype.

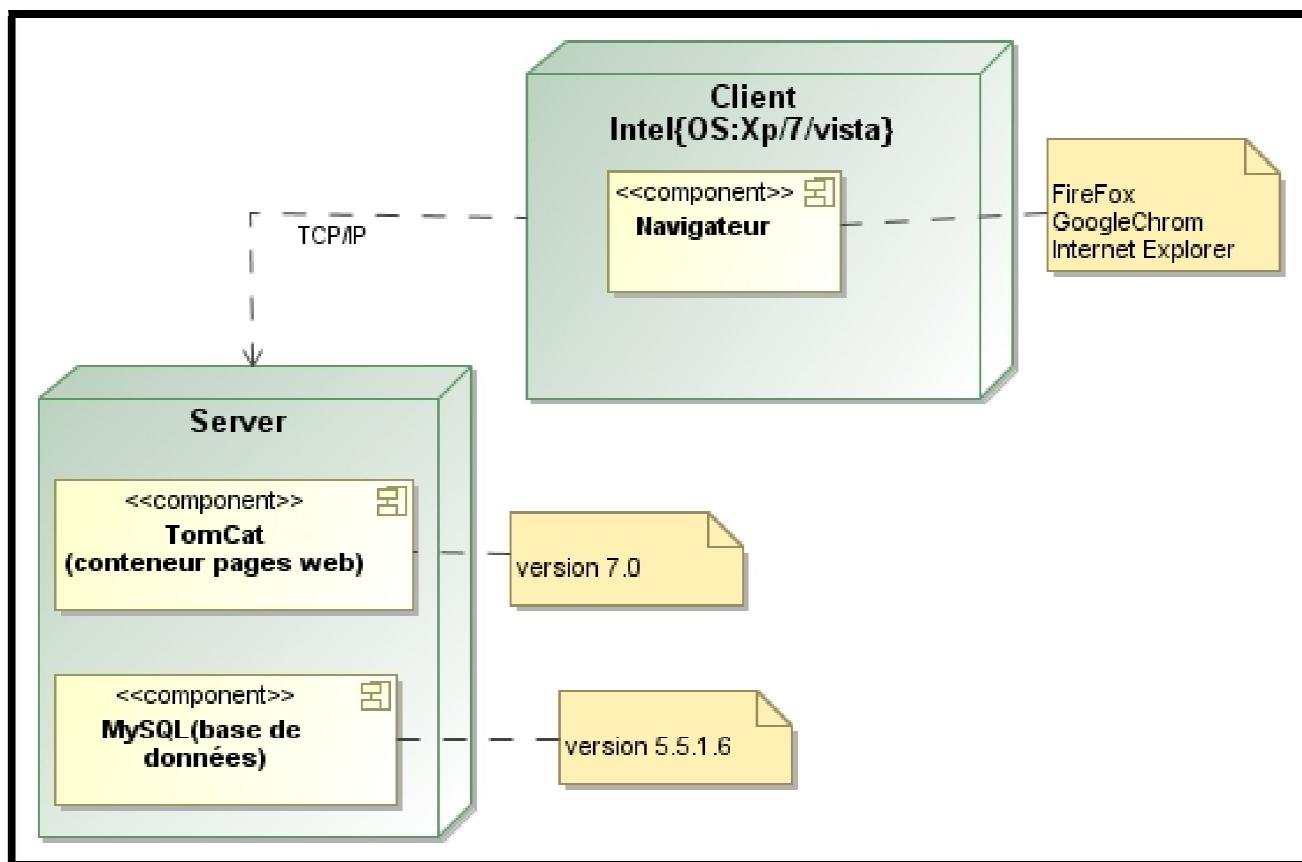
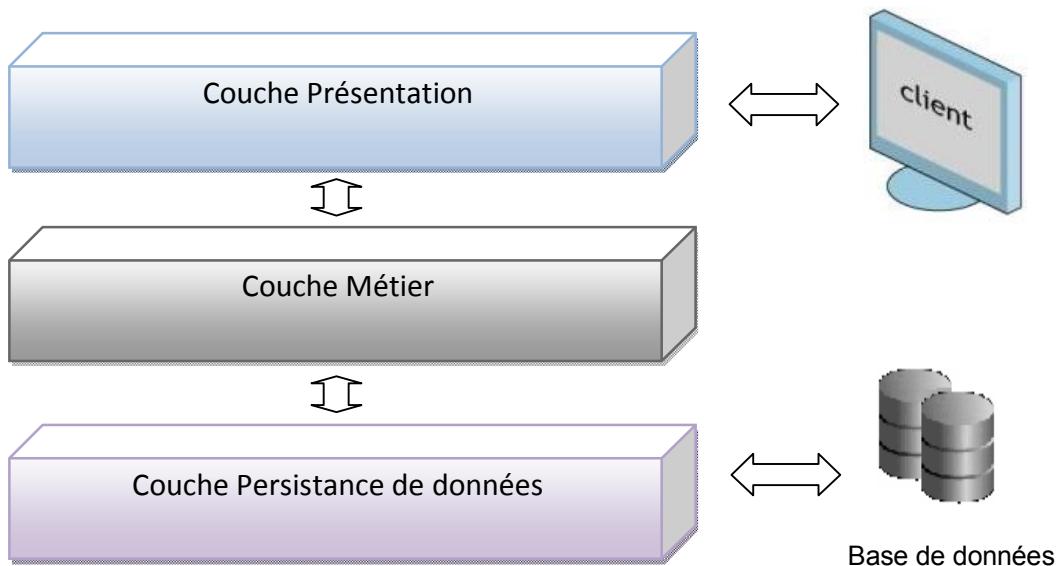


Figure 12:Diagramme de déploiement

## IV.ARCHITECTURE APPLICATIVE

Notre application est constituée de trois couches. Ces trois couches se conforment à l'architecture de couches fermées «Closed layer architecture» (une couche peut communiquer seulement avec la couche qui lui est adjacente).

La figure suivante présente l'architecture Java EE d'une application web en trois couches :



Le concept de partitionner une application en couches et de garder toute la logique de l'application dans ces couches distinctes et séparées, a été introduite bien avant l'approche orientée objet. Ainsi une application est divisée en trois couches logiques, chacune traitant des fonctions spécifiques :

- ➔ Présentation : interface usager et présentation.
- ➔ Logique du logiciel à produire (besoins, services de l'entreprise) : les règlements de l'entreprise et la logique de l'application.
- ➔ Logique des données : Base de données et intégration des services de l'entreprise.

Ce concept nous permet de créer des composants indépendants et de les déployer sur des plates-formes différentes. En fait, ce concept est très utilisé dans le développement des applications multi-tier. Plus tard, il fut adapté au modèle de conception Model-View-Controller (MVC) qui est un modèle très commun pour développer des applications distribuées et multi-tier.

## 1. CHOIX TECHNOLOGIQUES

### → HIBERNATE HIBERNATE

Hibernate est un Framework open source gérant la persistance des objets en base de données relationnelle

### → JSP :

Les JSP (Java Server Pages) sont une technologie Java qui permettent la génération de pages web dynamiques.

La technologie JSP permet de séparer la présentation sous forme de code HTML et les traitements sous formes de classes Java définissant un bean ou une servlet. Ceci est d'autant plus facile que les JSP définissent une syntaxe particulière permettant d'appeler un bean et d'insérer le résultat de son traitement dans la page HTML dynamiquement.

## 2. OUTILS :



### TOMCAT

L'utilisation d'un serveur Java EE est obligatoire pour le développement de pages Web dynamiques en Java EE. Un serveur HTTP classique reçoit des requêtes HTTP et renvoie des réponses mais il ne connaît pas les Servlets, les JSP... Il est donc essentiel d'utiliser un programme appelé moteur de Servlets qui est contenu dans le serveur Java EE et qui permet de pallier ce manque.



### ECLIPSE

Eclipse est l'environnement de développement (spécialisé pour le langage Java) qui sera utilisé dans ce projet. Le choix d'Eclipse repose essentiellement sur sa gratuité, sa facilité d'utilisation, sa puissance de développement et surtout ses nombreux plugins (bibliothèques additives).



### APACHE

Apache est le serveur Web le plus utilisé sur Internet. Dans une architecture en production, il est recommandé d'utiliser un serveur Web en frontal d'un serveur d'applications. Ces recommandations sont également appliquées

dans le cas de l'utilisation d'un conteneur Web comme Tomcat.

L'utilisation d'un serveur Web en frontal est nécessaire dans ce projet pour des raisons de performance, de sécurité et de flexibilité.



### Coupler Tomcat et le serveur web Apache

L'intégration d'un serveur Tomcat avec un serveur Web se fait au travers d'un connecteur configuré au sein de Tomcat et d'une extension ajoutée au serveur Web. Un connecteur Tomcat est une classe Java qui supporte un protocole réseau spécifique et propriétaire. La librairie d'extension du serveur Web permet un dialogue entre les deux serveurs.



### MySQL

Le SGBD MySQL est supporté par un large éventail d'outils. MySQL est surtout installé pour les applications Web, ce SGBD est solide et utilisé par de grands groupes spécialisés dans l'Internet. Plusieurs pilotes natifs de type 4 sont disponibles pour MySQL et sont conseillés pour une utilisation en Java.