

TABLES DES MATIERES

DEDICACE.....	2
REMERCIEMENT.....	3
INTRODUCTION GENERALE.....	7
CHAPITRE 1 : PRESENTATION DU PROJET.....	9
1. INTRODUCTION	9
2. PRESENTATION DE LA PHARMACIE.....	9
3. ENJEUX ET VISION DU PROJET.....	10
4. PRESENTATION DE LA MISSION.....	11
5. DESCRIPTION DE L'EXISTANT	11
5.1. Etude de l'existant.....	11
5.2. Présentation de l'application.....	11
5.3. Critique de l'existant.....	12
5.4. Orientations(Solutions)	12
6. METHODOLOGIE ADOPTEE.....	13
6.1. Processus Unifié	13
6.2. Spécification des besoins.....	14
6.3. Identification des besoins.....	14
6.3.1. Les besoins fonctionnels	14
6.3.2. Les besoins non fonctionnels.....	15

7. LES CAS D'UTILISATION	16
7.1. Définition.....	16
7.2. Identification des acteurs du système	16
8. CONCLUSION.....	17
CHAPITRE 2 : CONCEPTION	18
1. INTRODUCTION.....	18
2. DESCRIPTION DU MODELE DE CAS D'UTILISATION	18
3. DIAGRAMME DE CAS D'UTILISATION :.....	20
3.1. Diagramme de cas d'utilisation authentification :.....	20
3.2. Diagramme de cas d'utilisation gestion pharmacie :	20
4. RAFFINEMENT DES CAS D'UTILISATIONS :.....	21
4.1. Gestion de stock :.....	21
4.2. Gestion des ventes :	21
4.3. Gestion des médicaments :.....	22
5. DESCRIPTION TEXTUELLE DES CAS D'UTILISATION :.....	22
5.1. Description textuelle pour le cas d'utilisation « s'authentifier » :.....	23
5.2. Description textuelle du cas d'utilisation<<gestion des utilisateurs>>	24
5.2.1. Description textuelle du sous cas <<Ajouter un utilisateur>>.....	24
5.2.2. Description textuelle du sous cas << Modifier un utilisateur>>.....	26
5.2.3. Description textuelle du sous cas << Supprimer un utilisateur>>	26
.....Error! Bookmark not defined.	
5.3. Description textuelle du cas <<gestion des médicaments>>	28
5.3.1. Description textuelle du sous cas <<Ajouter un médicament>>.....	28
5.3.2. Description textuelle du sous cas << Modifier un médicament>>	29
5.3.3. Description textuelle du sous cas <<Supprimer un médicament>>	30
5.4. Description textuelle du cas << gestion des ventes>>	31
5.5. Description textuelle du cas << gestion de stock>>	32

6. DIAGRAMME DE SEQUENCE	33
6.1. Diagramme de séquence d'authentification :	33
6.2. Diagramme de séquence de traitement de vente :	34
6.3. Diagramme de séquence pour ajouter un nouveau produit	35
6.4. Diagramme de séquence de la suppression de médicament :	36
7. CONSTRUCTION DU DIAGRAMME DE CLASSES :	37
7.1. La liste des supports d'information :	37
7.2. Description des classes :	38
7.3. Description des associations :	39
8. CONCLUSION.....	40
CHAPITRE 3: ETUDE TECHNIQUE ET IMPLEMENTATION.....	41
1. INTRODUCTION	41
2. OUTILS ET LANGAGE UTILISE :	41
2.1. Environnement de réalisation :	41
2.1.1. L'environnement matériel :	41
2.1.2. L'environnement logiciel :	42
3. CONCEPTION DE LA BASE DE DONNEES :	44
4. REALISATION DU PROJET :	46
4.1. Interface d'authentification	47
4.2. Menu principale	48
4.3. Les Fenêtres Gestion des Produits :	48
d) Consulter le stock :	51
5. CONCLUSION :	54
CONCLUSION GENERALE ET PERSPECTIVE	55

INTRODUCTION

GENERALE

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de l'entreprise et d'autres établissements.

Avant l'invention de l'ordinateur, on enregistrait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de données des entreprises, ce qui est la partie essentielle dans leur développement aujourd'hui.

Les pharmacies hospitalières et celles des dispensaires publiques font partie intégrante des établissements que l'informatique pourra beaucoup aider. En effet, la croissance du nombre des médicaments hospitaliers nécessite la mise en place d'une gestion rationnelle prise et rapide, or et jusqu'à ce jour, la manière de gérer manuellement est encore dominante.

On remarque ainsi la mauvaise organisation du travail dans la pharmacie lors de la recherche d'une information ainsi lors de la création des statistiques l'information n'est pas toujours précise ni disponible d'où la nécessité d'introduire l'informatique dans les pharmacies hospitalières.

Vu cet état de fait, mon projet fin d'études a pour objectif de concevoir et mettre en œuvre une application web interactive, fiable, conviviale et facile à intégrer dans l'environnement de travail des pharmacies assurant la gestion de ces dernières et de suivre les ordonnances en prenant en considération le type des médicaments sortis à chaque ordonnance et l'état de stock des médicaments, en essayant de trouver les solutions aux problèmes rencontrés lors de l'exécution.

Cette application vise essentiellement à diminuer la complexité des traitements ainsi que le temps perdu lors de la gestion de stock, en particulier

Dans ce travail, on a commencé par faire une présentation de projet, ainsi le choix méthodologique suivi.

On a spécifié par la suite les besoins fonctionnels de l'application, le diagramme du cas d'utilisation général du système.

Par la suite on a étudié la partie conception qui contiendra une description détaillée de cas d'utilisation, les diagrammes de séquence ainsi que le diagramme de classe détaillé.

Après avoir achevés la partie conception, on a définis dans le dernier chapitre l'étude technique en précisant l'outil de travail et l'implémentation du projet. On a aussi illustré quelques interfaces qui donnent une idée sur les fonctionnalités de l'application.

CHAPITRE 1 :

1. Introduction :

De nos jours, il n'est plus aisé de définir les enjeux des nouvelles technologies de la gestion dans la professionnalisation de pharmacologie. En effet, elles représentent un véritable aide dans le travail du pharmacien et elles assurent la communication et la collaboration avec ses correspondants et patients. Ce chapitre va présenter le cahier des charges du projet. Dans lequel on va effectuer une étude de l'existant du projet et les solutions proposées et on précisera les besoins fonctionnels et non fonctionnels.

2. Présentation de la pharmacie :

Dans l'exercice de la pharmacie, tout pharmacien, en fonction du poste qu'il occupe, peut être appelé à utiliser des notions de base en gestion des ressources humaines, financières et matérielles. Planifier, organiser, budgéter, diriger, gérer, contrôler et évaluer représentent des fonctions de gestion nécessaires au maintien d'un environnement professionnel. Ceci permet d'assurer aux patients l'accès aux services et aux produits requis pour répondre à leurs besoins.

L'application, dans son milieu d'exercice, des principes de base et des habiletés de gestion permet au pharmacien d'optimiser les soins pharmaceutiques et les relations interprofessionnelles. Elle contribue également à assurer et à maintenir l'hygiène, la sécurité, l'efficacité et l'efficience du milieu de travail.



Figure I.1 : pharmacie

3. Enjeux et vision du projet

Comme de nombreuses autres entreprises, les pharmacies sont en face de la contrainte de gestion des stocks. 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 est fait grâce 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 ou 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, elle a pour objectif la satisfaction continue de la demande de sa clientèle en produits pharmaceutiques, cela à moindre coût.

4. Présentation de la mission :

Dans cette partie je voudrai présenter les principes de fonctionnement du système utilisé. Je commencerai par une description de l'existant puis déterminer les besoins fonctionnels et non fonctionnels du système ensuite définir les acteurs qui interagissent avec le système.

5. Description de l'existant :

5.1. Etude de l'existant

Ce projet consiste à créer une application permettant de gérer une pharmacie, il s'agit de définir les responsabilités de la gestion, mettre à jour les données, organiser des données collectées, de renforcer le contrôle et la confrontation, assurer une meilleure gestion et une cohérence de l'information et enfin faciliter le travail des responsables.

5.2. Présentation de l'application

Cette application que nous nous sommes proposé de développer pour l'automatisation de la gestion de la majorité des activités existantes dans la pharmacie, doit permettre de répondre aux exigences de cette dernière. Pour le développement de cette application, nous avons jugé nécessaire d'utiliser les différents outils et méthodes qui sont les suivants :

- Pour la Programmation, nous utilisons **Netbeans7.3 langage java**.
- Pour le traitement de texte, nous travaillons avec **Microsoft Office**.
- Nous utilisons la base de données **SQLite manager**

5.3. Critique de l'existant

L'analyse de l'existant met l'accent sur plusieurs difficultés telles que :

1. Le travail de certaines pharmacies publiques se fait encore manuellement.
2. Négligence du facteur temps : le facteur temps est un facteur fondamental pour toutes activités dans le centre médical et vue que les tâches destinées au responsable de pharmacie, pour bien gérer le stock des médicaments, il sera difficile de réussir cette tâche manuellement, aussi bien pour les différentes ordonnances que pour les statistiques qui lui sont associées.
3. Mal organisation du travail dans la pharmacie.
4. Les documents (fiche de produit, le stock, bon de livraison, etc.) ne sont pas bien détaillés.
5. Volume important des informations traitées manuellement, ce qui provoque parfois des erreurs dans l'établissement des documents.
6. Recherche difficile sur les registres qui engendre une perte de temps
7. Insécurité des informations.
8. Possibilité d'erreur dans le remplissage des différents documents et registres.
9. Possibilité d'erreur dans les calculs des statistiques.
10. Nombre important des archives qui engendre une difficulté de stockage.
(Déterioration des archives à force de leur utilisation trop fréquente.
11. Mauvaise codification sur quelques objets dans la gestion d'information.

5.4. Orientations(Solutions) :

Afin de corriger les problèmes présentés ci-dessus, je suis appelé à réaliser cette application qui assure les points suivants :

1. Automatiser les tâches qui se traitent manuellement.
2. Faciliter la recherche et l'accès aux informations.

3. Sauvegarder toutes les données relatives à la gestion des ordonnances sur des supports informatiques ce qui assurera leur sécurité.
4. Minimiser les supports papiers utilisés.
5. Faire toute modification (ajout, suppression, modification) automatiquement.
6. Plus d'organisation dans le travail du responsable de pharmacie.
7. Faciliter la recherche de l'information.
8. Rapidité dans l'établissement des différents documents.
9. Gain de temps dans les calculs des statistiques.¹ Proposer une bonne codification.

6. Méthodologie adoptée :

L'objectif de toute approche de conduite de projet est d'obtenir des résultats fiables. En fait, la fiabilité d'un système dépend de l'approche utilisée. Nous avons adopté pour un processus de développement logiciel appelé **Processus Unifié**.

6.1. Processus Unifié :

Le processus unifié est un processus de développement logiciel qui regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. Mais c'est plus qu'un simple processus. C'est un framework de processus générique pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétence et à différentes tailles de projets [1].

Ce processus simplifié aux caractéristiques suivantes :

- Piloté par les cas d'utilisation d'UML
- Ne néglige pas l'analyse et la conception
- Utilise 20% d'UML pour modéliser 80% du système

Le processus simplifié est composé des phases suivantes :

- Étude des besoins (qui est détaillé en ce qui va suivre)
- Analyse
- Conception· Implémentation

6.2. Spécification des besoins

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques les plus critiques et d'identifier les cas d'utilisation initiaux

6.3. Identification des besoins

6.3.1. Les besoins fonctionnels :

Les besoins fonctionnels se rapportent aux fonctionnalités que l'application en question doit offrir pour satisfaire les utilisateurs.

Les fonctionnalités que doit intégrer l'application à développer peuvent être décrites comme suit :

- Gestion des sécurités** : Le Système permet de gérer les droits d'accès de chaque utilisateur en associant un ID et un mot de passe pour chaque utilisateur (pharmacien ou employé).
- Gestion des médicaments** : Cette opération consiste à ajouter un nouveau médicament dans le stock, modifier ou supprimer un médicament qui existe déjà stock à savoir les mouvements réalisés sur le stock (entrée /sortie de médicament, quantité des médicaments dans le stock).
- Gestion des ventes** : 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.. L'utilisateur doit créer une facture correspondant à la liste des médicaments prescrits dans les ordonnances.

- d. Gestion stock** : Lorsqu'un fournisseur lui livre les médicaments commandés, le propriétaire de la pharmacie doit implémenter la base de données du logiciel par le nombre de médicaments livrés, et vérifier leurs dates de péremption.
- e. Facturation** : cette fonction permettra de suivre les différentes statistiques et informations possibles (vendeur, montant, date...) pour chaque facture selon son numéro.

6.3.2. Les besoins non fonctionnels

Les besoins non fonctionnels sont indispensables et permettent l'amélioration de la qualité logiciel de notre système. Ils agissent comme des contraintes sur les solutions, mais leur prise en considération fait éviter plusieurs incohérences dans le système. Ce dernier doit répondre aux exigences suivantes :

- a. Authentification** : le système doit permettre à l'utilisateur de saisir son login et son mot de passe pour accéder au système. Cette opération assure la sécurité du système et limite le nombre des utilisateurs.
- b. Ergonomie** : le système devra offrir aux utilisateurs une interface qui soit le plus riche possible afin de limiter le nombre d'écrans. Par ailleurs, l'interactivité devra être adaptée (usage du clavier, menu, etc..).

7. Les cas d'utilisation

7.1. Définition

Les cas d'utilisation représentent un élément essentiel de la modélisation orientée objet : ils doivent en principe permettre de concevoir et de construire un système adapté aux besoins de l'utilisateur.

Un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. D'après Bittner et Spence, « Un cas d'utilisation, défini simplement, permet de décrire une séquence d'événements qui, pris tous ensemble, définissent un système faisant quelque chose d'utile »[2].

Chaque cas d'utilisation contient un ou plusieurs scénarios qui définissent comment le système devrait interagir avec les utilisateurs (appelés acteurs) pour atteindre un but ou une fonction spécifique d'un travail.

7.2. Identification des acteurs du système

Un acteur représente un rôle joué par une personne ou une chose qui interagit avec un système. En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin.

Les différents acteurs définis pour notre système sont les suivants :

- **Pharmacien (*principal*)** : Il s'occupe à la fois de la partie de la gestion des médicaments, de la gestion des ventes.
- **Gestion médicaments** : Il a pour rôle d'effectuer le traitement qui touche directement au stock après réception des ordonnances provenant des clients : demandes des produits, suivi des mouvements des médicaments et l'état du stock.
- **Gestion des ventes** : pour déterminer la quantité de médicament, une facture de vente est préparée.
- **Employé(secondaire)**

8. Conclusion

On a réussi à d'introduire ce projet, de préciser le travail demandé et de définir la méthodologie de travail, ainsi d'identifier les acteurs du système. Ce qui va permettre la bonne implémentation du projet dans les chapitres qui suivent.

CHAPITRE 2 : CONCEPTION

1. Introduction

Dans ce chapitre, le diagramme global des cas d'utilisation du système ainsi que les autres cas d'utilisation en particulier seront présentés. Cela est fait en utilisant les acteurs qui interagissent avec le système. On couvre une vue dynamique du système à travers les différents diagrammes de séquences relatifs aux cas d'utilisations. Enfin, on dégage les différentes tables de la base de données via le diagramme de classe.

2. Description du modèle de cas d'utilisation

Dans un diagramme de cas d'utilisation (figure II.1), elles peuvent être reliées par une relation de généralisation ou de dépendance (inclusion ou extension).

La relation de généralisation entre cas d'utilisation signifie que le cas d'utilisation père peut être substitué par le cas d'utilisation fils. La relation de dépendance d'inclusion est utilisée pour éviter de décrire plusieurs fois les mêmes collaborations d'objets.

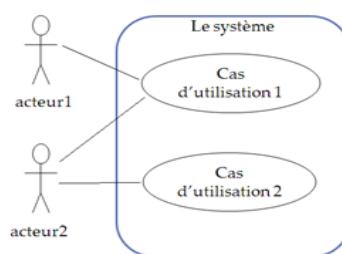


Figure II.1 : exemple de diagramme de cas d'utilisation

Cette relation peut aussi être vue comme un exemple de délégation, c'est à dire que le cas d'utilisation qui est inclus représente une responsabilité particulière dans le S.I. qui pourra être utilisée par les autres cas d'utilisation qui en auront besoin.

La relation de dépendance d'extension entre cas d'utilisation indique que le cas d'utilisation de base va éventuellement être étendu par le cas d'utilisation à la base de la relation. Ce cas d'utilisation peut correspondre à la description du traitement d'une exception.[3]

L'acteur ici est le pharmacien, le diagramme des cas d'utilisation est montrée par le tableau des cas d'utilisation (figure II.2)

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 sécurités
	Possède des droits sur la gestion des ventes.

Figure II.2 : tableau des cas d'utilisation

Le nombre des acteurs et des cas change selon les besoins de chaque projet. La représentation de cas d'utilisation en diagrammes est décrite par les figures ci-dessous :

3. Diagramme de cas d'utilisation :

3.1. Diagramme de cas d'utilisation authentification :

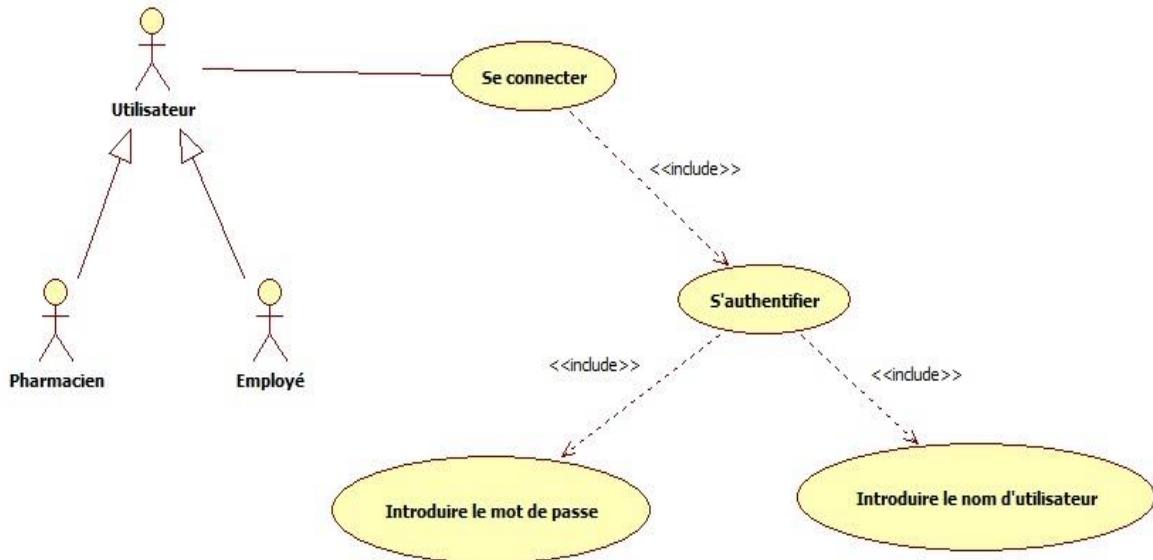


Figure II.3 : Diagramme de cas d'utilisation 'S'authentifier'

3.2. Diagramme de cas d'utilisation gestion pharmacie :

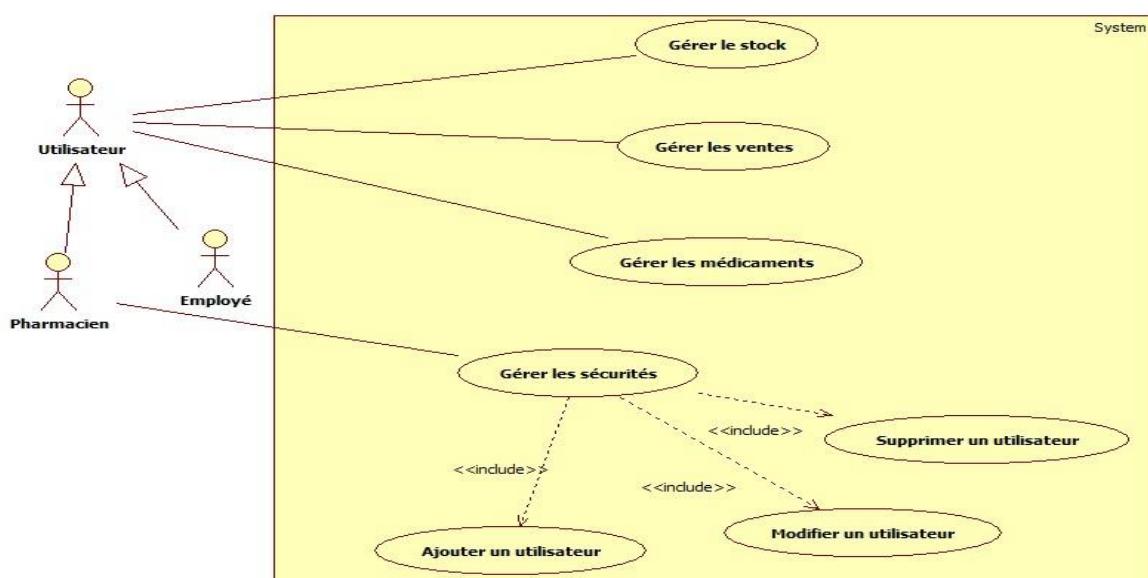


Figure II.4 : Diagramme global des cas d'utilisation

4. Raffinement des cas d'utilisations :

4.1. Gestion de stock :

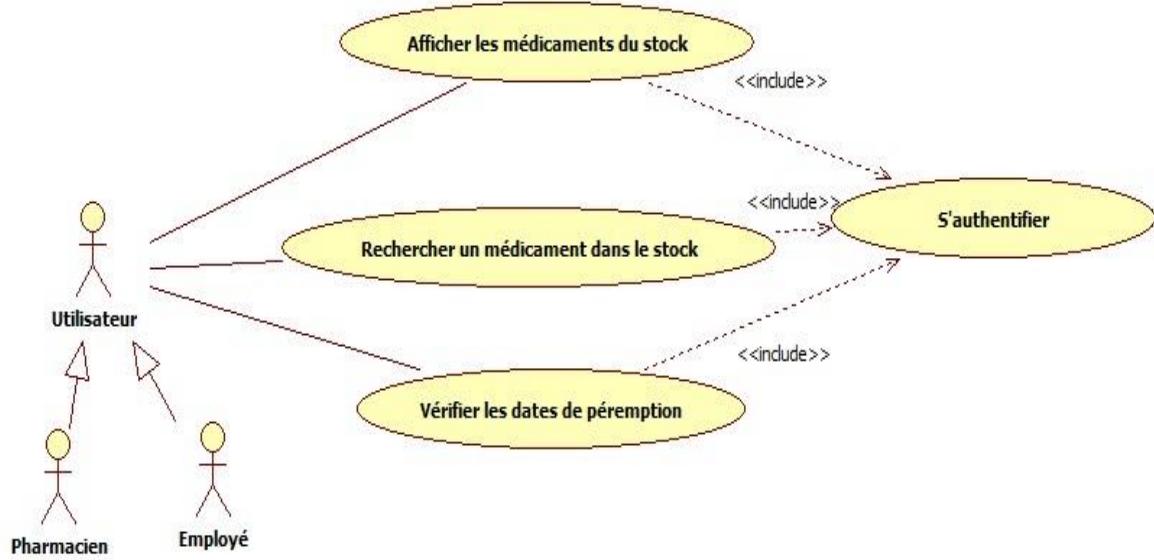


Figure II.5 : Diagramme de cas d'utilisation gestion de stock

4.2. Gestion des ventes :

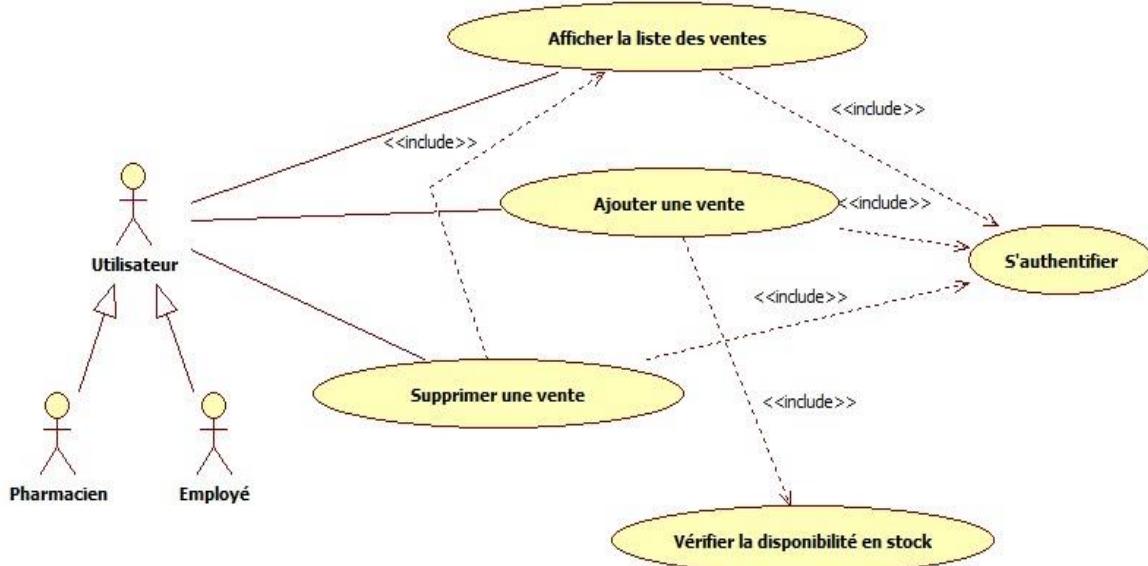


Figure II.6 : Diagramme de cas d'utilisation gestion des ventes

4.3. Gestion des médicaments :

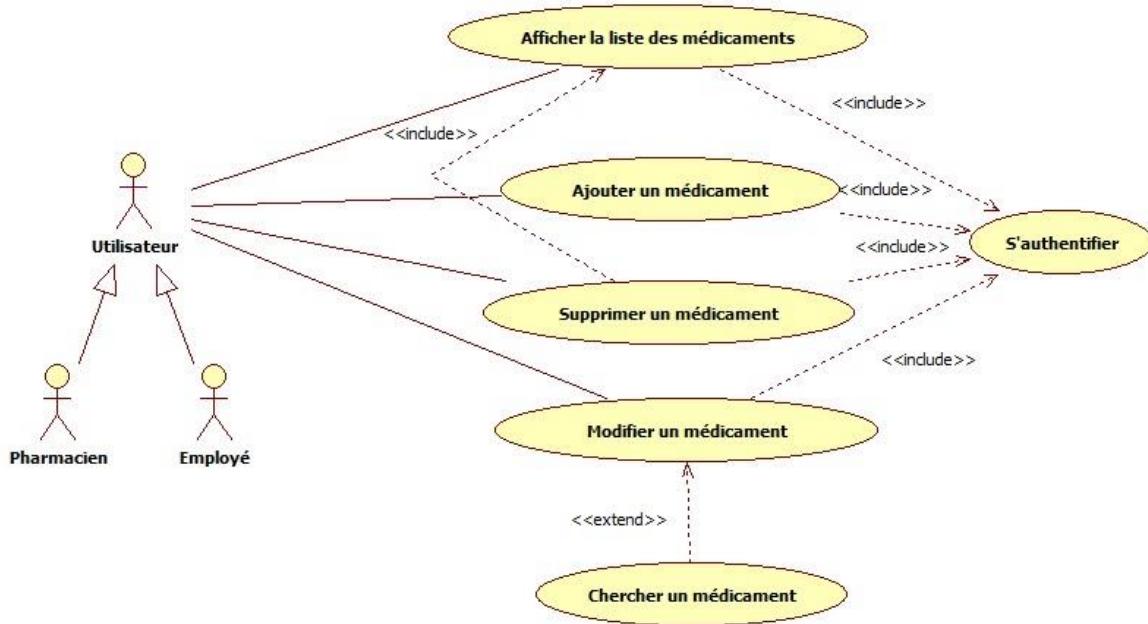


Figure II.7 : Diagramme de cas d'utilisation gestion des médicaments

5. Description textuelle des cas d'utilisation :

Dans cette section, chaque cas d'utilisation sera décrit de façon exhaustive suivant le format présenté dans les cas d'utilisation précédents.

Pour la documentation des cas d'utilisation (les scenarios), et aussi la description textuelle est indispensable, car elle seule permet de communiquer facilement et précisément avec les utilisateurs. Elle est l'occasion d'identifier le contexte d'exécution de l'un ou de l'autre des enchainements.

Cette partie est composée de deux :

1- **Sommaire d'identification** :(titre, but, résumé, acteur).

- **Titre** : Cas d'utilisation concerné.
- **But** : l'objectif de ce cas d'utilisation dans le système.
- **Résumé** : c'est le résumé du contenu textuel

2- **Description de l'enchaînement** :(pré-condition, post-condition, scenario nominale, scénario alternative).

- **Pré-condition** : ce sont les conditions nécessaires pour déclencher les enchainements.
- **Post-condition** : représente l'événement futur.
- **Scenario nominale** : représente les événements produits par l'acteur et le système de la façon sans échec (sans erreur).
- **Scenario alternative** : représente les événements après les erreurs produits par l'acteur et le système.

5.1. Description textuelle pour le cas d'utilisation « s'authentifier » :

Sommaire d'identification
<p>Titre : s'authentifier</p> <p>But : Permet à l'utilisateur d'établir son propre nom d'utilisateur et son mot de passe pour accéder à l'interface.</p> <p>Résumé : chaque utilisateur doit taper son propre nom d'utilisateur et son mot de passe pour accéder à l'interface qui le concerne.</p> <p>Acteur : Pharmacien, employé.</p>
Description de l'enchaînement
<p>Pré conditions : l'utilisateur n'est pas authentifié donc il ne peut pas accéder à l'application.</p> <p>Post conditions : l'utilisateur est authentifié, il accède à l'interface qui le concerne.</p>

Scénario nominal :

1. L'utilisateur tape son propre nom d'utilisateur et son mot de passe.
2. L'utilisateur clique sur le bouton pour y accéder.
3. Le système vérifie la validité des coordonnées de l'utilisateur.
4. L'utilisateur va accéder à la session qui le concerne.

Scénario alternatif :

A1 : l'enchaînement démarre à l'étape 3 du scénario nominale. Le système indique le non validité des coordonnées de l'utilisateur, alors il ne peut pas accéder à la session. L'enchaînement reprend à l'étape 1 du scénario nominal

Figure II.8 : Table de description textuelle pour cas d'utilisation « s'authentifier ».

5.2. Description textuelle du cas d'utilisation <<gestion des utilisateurs>>

5.2.1. Description textuelle du sous cas <<Ajouter un utilisateur>>

Sommaire d'identification

Titre : Créer un nouvel utilisateur

But : Le système permet de créer un nouveau Id et un mot de passe.

Résumé : chaque utilisateur doit avoir son propre nom d'utilisateur et son mot de passe

pour accéder à l'interface qui le concerne.

Acteur : Pharmacien

Description de l'enchaînement

Pré conditions : Pharmacien connecté, l'utilisateur n'est pas créé

Post conditions : un nouvel utilisateur et un mot de passe sont créé.

Scénario nominal :

1. Le cas d'utilisation commence lorsque le pharmacien demande à créer un nouveau Id et un mot de passe
2. Le système affiche l'interface appropriée.
3. Le pharmacien effectue les opérations de création.
4. le système vérifie que l'Id n'existe pas déjà.
5. Le système enregistre l'opération effectuée.

Scénario alternatif :

A1 : l'enchaînement démarre à l'étape 4 du scénario nominale. Le système indique que l'Id

existe déjà, alors il faut rentrer un nouveau Id.

L'enchaînement reprend à l'étape 3 du scénario nominal

Figure II.9 : Table de sous cas « ajouter un utilisateur».

5.2.2. Description textuelle du sous cas << Modifier un Utilisateur>>

Sommaire d'identification
Titre : Modifier le mot de passe utilisateur
But : Le système permet de modifier un ancien mot de passe utilisateur.
Résumé : le pharmacien peut réinitialiser les mots de passe des utilisateurs.
Acteur : Pharmacien
Description de l'enchaînement
Pré conditions : mot de passe actuel
Post conditions : le mot de passe de l'utilisateur est modifié
Scénario nominal :
<ol style="list-style-type: none">1. Le cas d'utilisation commence lorsque l'utilisateur demande de modifier les mots de passe des utilisateurs.2. Le système affiche l'interface appropriée.3. L'utilisateur saisie le nom de l'utilisateur à modifier son mot de passe.4. Le pharmacien saisit l'ancien mot de passe.5. Le pharmacien saisit le nouveau mot de passe.6. Le pharmacien confirme le nouveau mot de passe.7. Le système valide et enregistre les modifications.
Scénario alternatif :
A1 : l'enchaînement démarre à l'étape 7 du scénario nominale. Le système indique que l'utilisateur n'existe pas. L'enchaînement reprend à l'étape 3 du scénario nominal.

Figure II.10 : Table de sous cas « modifier un utilisateur».

5.2.3. Description textuelle du sous cas << Supprimer un utilisateur>>

Sommaire d'identification
Titre : Supprimer un utilisateur
But : Le système permet de supprimer un utilisateur.
Résumé : le pharmacien gère les utilisateurs de l'application.
Acteur : Pharmacien
Description de l'enchaînement
Pré conditions : Utilisateur actuel
Post conditions : L'utilisateur est supprimé
Scénario nominal :
<ol style="list-style-type: none">1. Le cas d'utilisation commence lorsque l'utilisateur demande de supprimer un utilisateur2. Le système affiche l'interface appropriée.3. Le pharmacien saisit le nom de l'utilisateur à supprimer.4. Le système valide et enregistre la suppression.

Figure II.11 : Table de sous cas « supprimer un utilisateur».

5.3. Description textuelle du cas <<gestion des médicaments>>

5.3.1. Description textuelle du sous cas <<Ajouter un médicament>>

Sommaire d'identification
<p>Titre : Ajouter un nouveau médicament</p> <p>But : Le système permet de créer un nouveau médicament.</p> <p>Résumé : l'utilisateur gère la liste des médicaments en ajoutant de nouveau qui n'existent pas déjà dans le stock</p> <p>Acteur : Utilisateur</p>
Description de l'enchaînement
<p>Pré conditions : Le médicament n'existe pas dans le stock</p> <p>Post conditions : Un nouveau médicament sera ajouté</p> <p>Scénario nominal :</p> <ol style="list-style-type: none">1. Le cas d'utilisation commence lorsque l'utilisateur demande à créer un nouveau médicament.2. Le système affiche l'interface appropriée.3. L'utilisateur effectue les opérations de création.4. Le système vérifie que le médicament n'existe pas déjà dans le stock5. Le système enregistre l'opération effectuée.6. Le système affiche un message de confirmation. <p>Scénario alternatif :</p> <p>A1 : l'enchaînement démarre à l'étape 4 du scénario nominale. Le système indique que le médicament existe déjà dans le stock et affiche son code.</p> <p>L'enchaînement reprend à l'étape 3 du scénario nominal.</p>

Figure II.12 : Table de sous cas « ajouter un médicament ».

5.3.2. Description textuelle du sous cas << Modifier un médicament>>

Sommaire d'identification
<p>Titre : Modification</p> <p>But : Le système permettra à l'utilisateur de modifier les données d'un médicament.</p> <p>Résumé : L'utilisateur peut modifier les informations concernant le médicament sélectionné</p> <p>Acteur : L'utilisateur</p>
Description de l'enchaînement
<p>Pré conditions : Le médicament existe déjà dans le stock</p> <p>Post conditions : médicament modifié.</p> <p>Scénario nominal :</p> <ol style="list-style-type: none">1. Le cas d'utilisation commence lorsque l'utilisateur demande de modifier les données d'un médicament.2. Le système affiche l'interface appropriée.3. L'utilisateur sélectionne le médicament à modifier.4. L'utilisateur saisit ensuite les nouveaux paramètres et valide.5. Le système enregistre les modifications.6. Le système affiche un message de confirmation.

Figure II.13 : Table de sous cas « modifier un médicament».

5.3.3. Description textuelle du sous cas <<Supprimer un médicament>>

Sommaire d'identification

Titre : Suppression

But : Le système permettra à l'utilisateur de supprimer un médicament de la liste.

Résumé : L'utilisateur peut supprimer un médicament sélectionné

Acteur : L'utilisateur

Description de l'enchaînement

Pré conditions : Le médicament existe déjà dans le stock

Post conditions : médicament supprimé.

Scénario nominal :

1. Le cas d'utilisation commence lorsque l'utilisateur demande la suppression d'un médicament.
2. Le système affiche l'interface appropriée.
3. L'utilisateur sélectionne le médicament à supprimer et valide.
4. Le système supprime le médicament sélectionné
- 6- Le système affiche un message de confirmation.

Figure II.14: Table de sous cas « supprimer un médicament».

5.4. Description textuelle du cas << gestion des ventes>>

Sommaire d'identification
Titre : Gérer les ventes
But : Mettre à jour les ventes.
Résumé : Cette fonctionnalité permet à l'utilisateur d'ajouter une vente, de consulter la liste des ventes, de les supprimer, de les chercher.
Acteur : L'utilisateur
Description de l'enchaînement
Pré conditions : Le médicament existe déjà dans le stock
Post conditions : médicament supprimé.
Scénario nominal :
<ol style="list-style-type: none">1. Le cas d'utilisation commence lorsque l'utilisateur demande de créer une vente.2. L'utilisateur sélectionne le médicament et saisit la quantité à vendre.3. Le système vérifie la disponibilité de la quantité du médicament à vendre.4. Le médicament s'affiche automatiquement dans l'interface de vente5. L'utilisateur valide et imprime la facture de vente
Scénario alternatif :
A1 : l'enchaînement démarre à l'étape 3 du scénario nominal. Le système indique par un message d'erreur que la quantité du médicament n'existe pas dans le stock.
L'enchaînement reprend à l'étape 2 du scénario nominal.

Figure II.15 : Table de cas « gestion de vente ».

5.5. Description textuelle du cas << gestion de stock>>

Sommaire d'identification
<p>Titre : Gérer le stock</p> <p>But : consulter le stock.</p> <p>Résumé : Cette fonctionnalité permet à l'utilisateur de consulter la listes des produits qui existent en stock, de les chercher et de vérifier les dates de péremptions.</p> <p>Acteur : L'utilisateur</p>
Description de l'enchaînement
<p>Pré conditions : l'utilisateur s'authentifie sur le système</p> <p>Post conditions : mettre à jour le stock.</p> <p>Scénario nominal :</p> <ol style="list-style-type: none">1. Le cas d'utilisation commence lorsque l'utilisateur demande d'accéder au stock2. Le système affiche la liste des produits présents dans le stock3. L'utilisateur vérifie la quantité et la date de péremption des produits.4. Stock vérifié avec succès

Figure II.16 : Table de cas « gestion de stock».

6. Diagramme de séquence

Le diagramme de séquence est une description graphique des opérations d'un système sous un angle chronologique. C'est une vue dynamique qui contient les symboles d'objets (instances de classe), d'acteurs et de messages qu'ils échangent. La dimension verticale est l'axe temporel : les messages y sont représentés par ordre chronologique. La dimension horizontale montre des objets et des acteurs qui échangent des informations, le diagramme de séquence permet de représenter les échanges entre les composants et les objets du système, dans le cadre d'exécution des cas d'utilisation, de point de vue temporel. [4]

Dans ce qui suit, nous présentons quelques fonctions de l'utilisateur à travers des différents diagrammes de séquences :

6.1. Diagramme de séquence d'authentification :

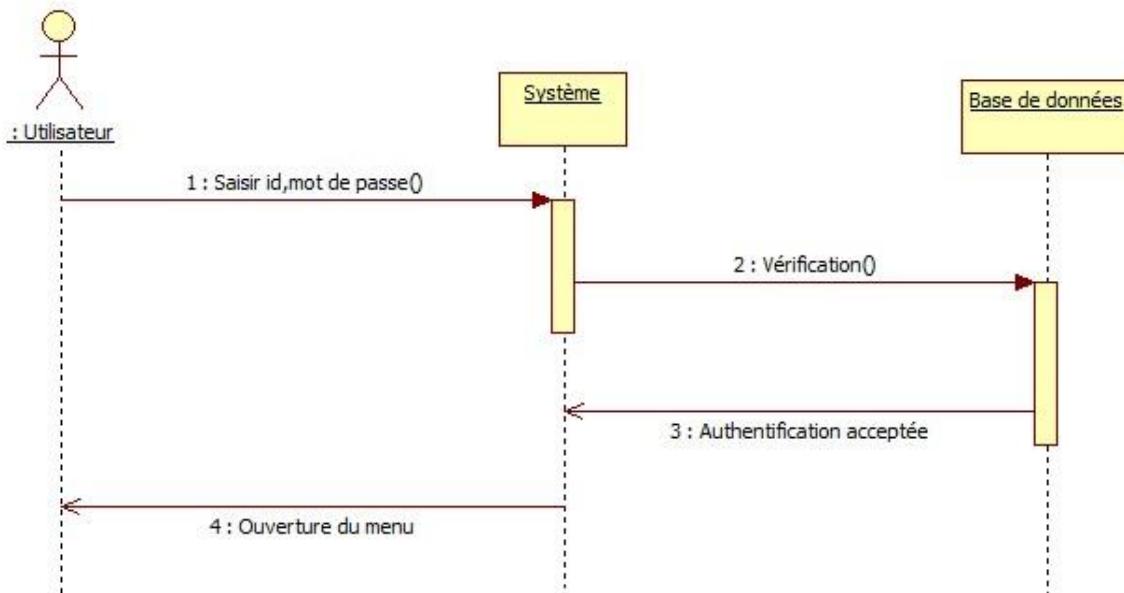


Figure II.17 : Diagramme de séquence de l'authentification

Description textuelle : L'utilisateur qui veut y accéder tape son id et son mot de passe, le système vérifie la concordance des données et confirme l'authentification, l'affichage de l'interface principale se suit directement.

6.2. Diagramme de séquence de traitement de vente :

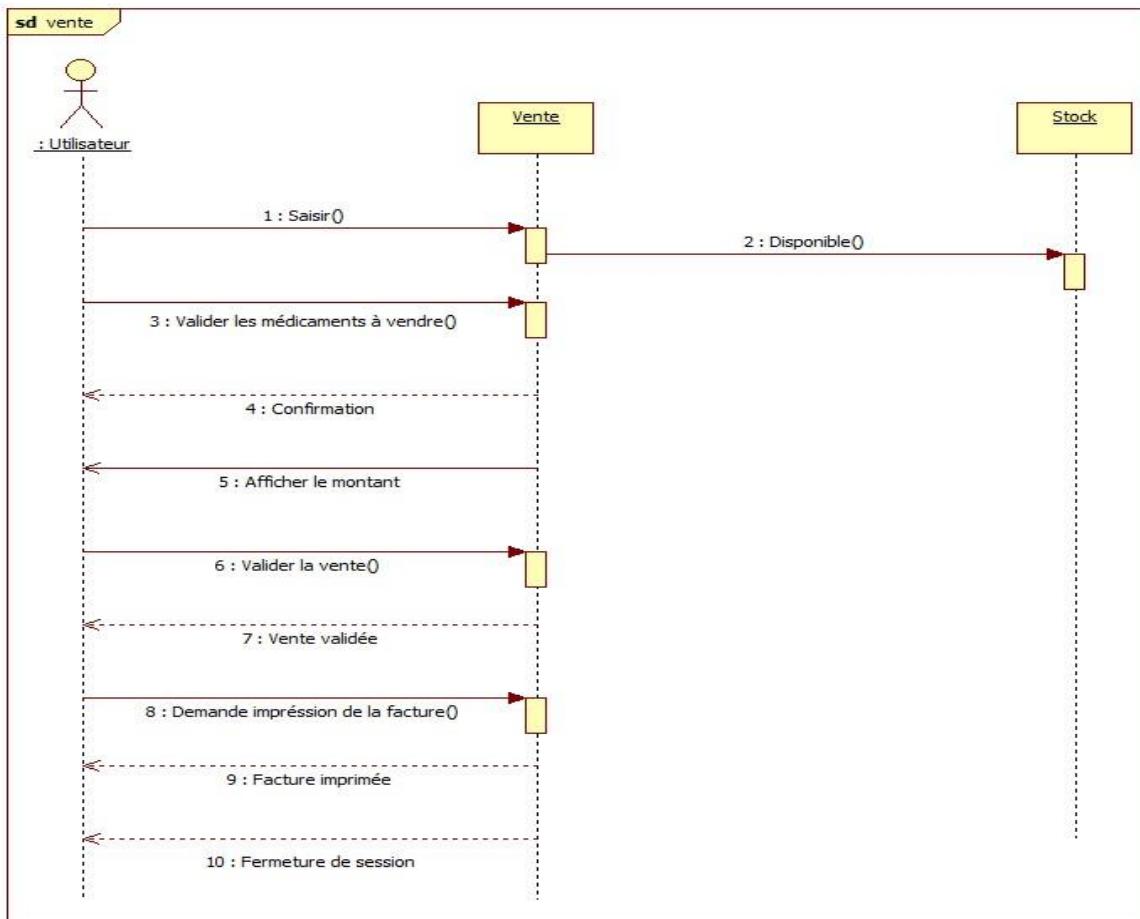


Figure II.18 : Diagramme de séquence de la gestion de vente

Description textuelle : L'utilisateur saisit les médicaments à vendre à l'aide de l'ordonnance, en même temps le système vérifie la disponibilité en termes de produit et de quantité, l'utilisateur valide la liste des produits et confirme la vente, le montant s'affiche, après l'effectue de la vente une facture sera imprimée.

6.3. Diagramme de séquence pour ajouter un nouveau produit

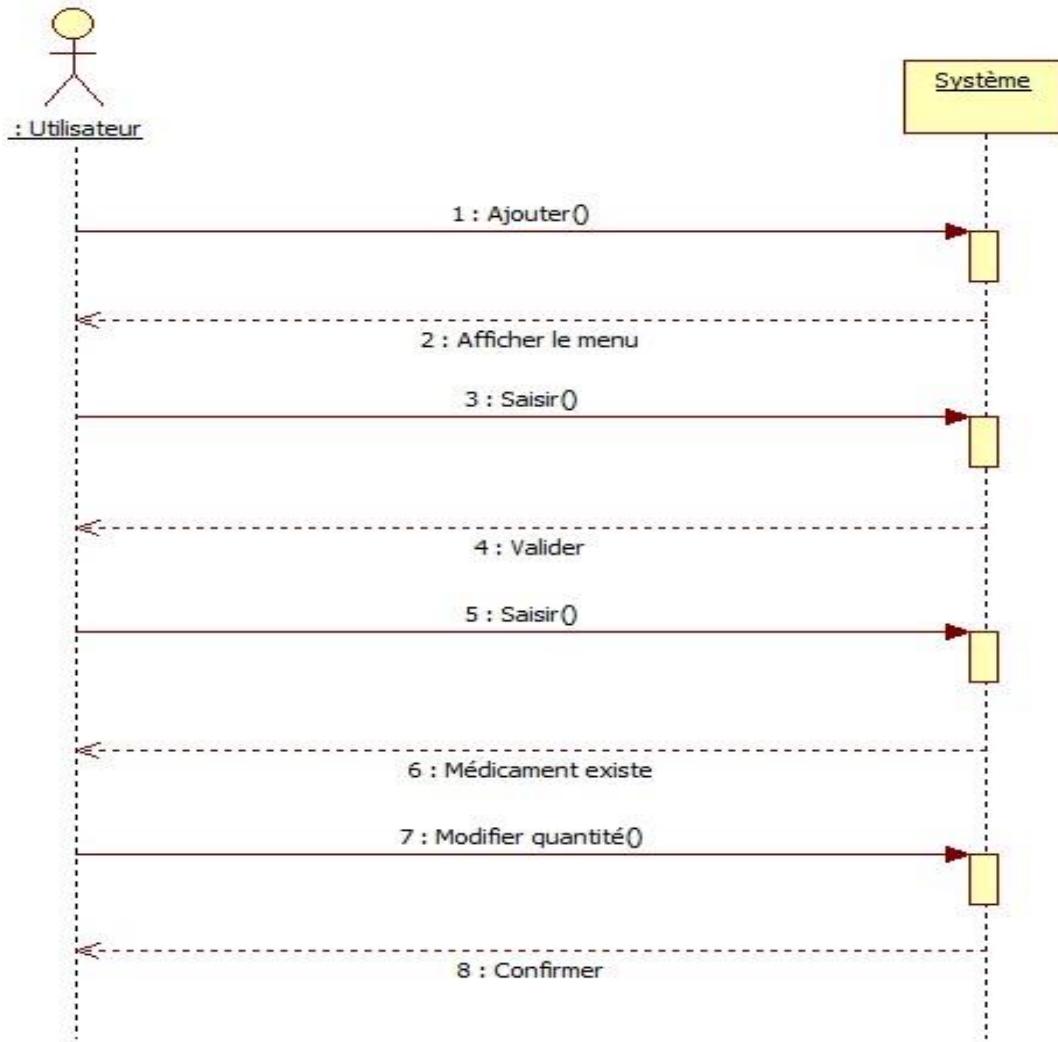


Figure II.19 : Diagramme de séquence de la mise à jour du stock (l'ajout)

Description textuelle : L'utilisateur mentionne les informations du nouveau médicament, le système valide, sinon l'utilisateur modifie seulement la quantité si le produit existe déjà dans le stock, le système confirme.

6.4. Diagramme de séquence de la suppression de médicament :

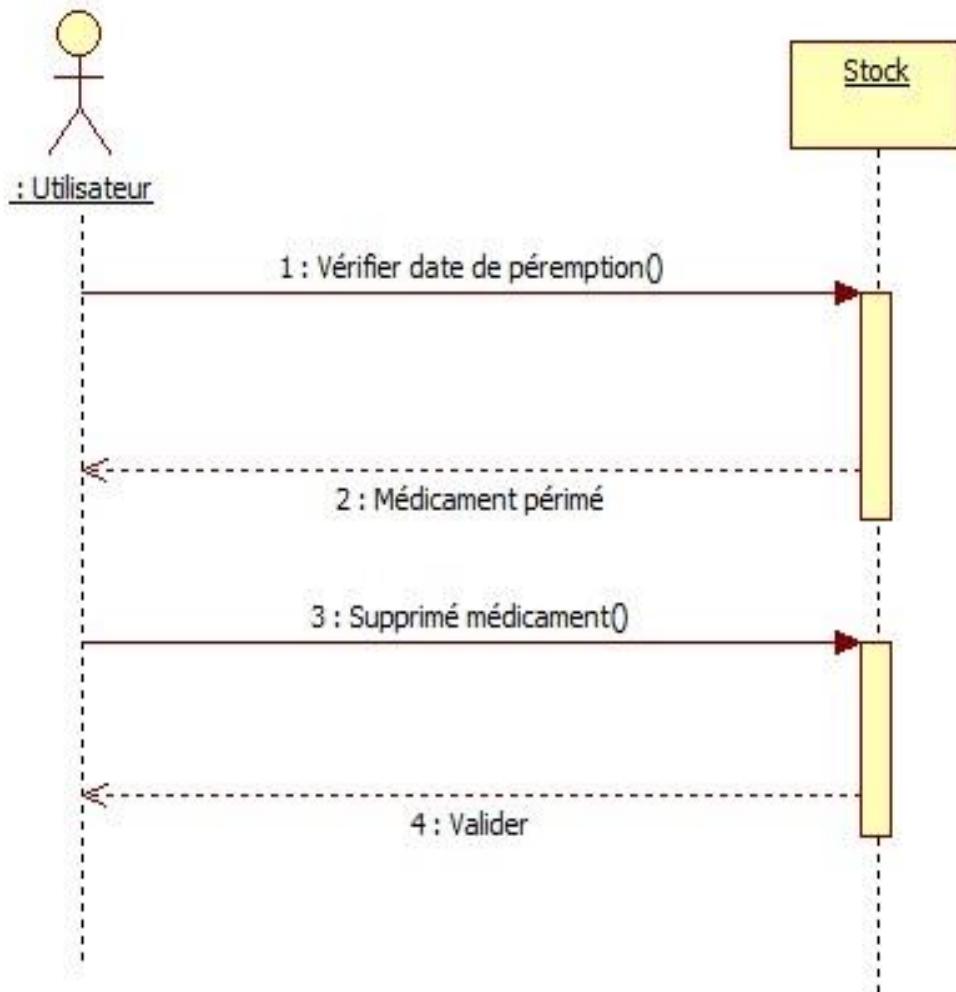


Figure II.20 : Diagramme de séquence de la suppression d'un médicament périmé

Description textuelle : L'utilisateur vérifie l'état du stock, notamment les dates de péremption, le stock retourne la présence des médicaments périmés, l'utilisateur supprime ces derniers et l'action sera validé.

7. Construction du diagramme de classes :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.[4]

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.[4]

7.1. La liste des supports d'information :

Support	Informations
Médicament	Code Quantité Nom Prix Taux Date péremption
Stock	Médicament

	Quantité Code Prix Date per taux
Facture	Numéro de facture Date Total montant Quantité
Vente	Numéro de vente Quantité disponible Quantité à vendre Date

7.2. Description des classes :

Après une analyse de l'existant, nous avons dégagé les classes nécessaires pour une bonne gestion de la pharmacie.

N	Nom de classe
1	Médicament
2	Stock
3	Facture

4	Vente
---	-------

7.3. Description des associations :

Numéro	Association	Classes participantes	Description
o1	Vendre médicament	Médicament, Vente	Un médicament peut être vendu plusieurs fois
o2	Appartient	Stock, Médicament	Le stock contient plusieurs médicaments
o3	Etablie	Vente, Facture	Une facture est établie après chaque vente

Voici le diagramme de classes :

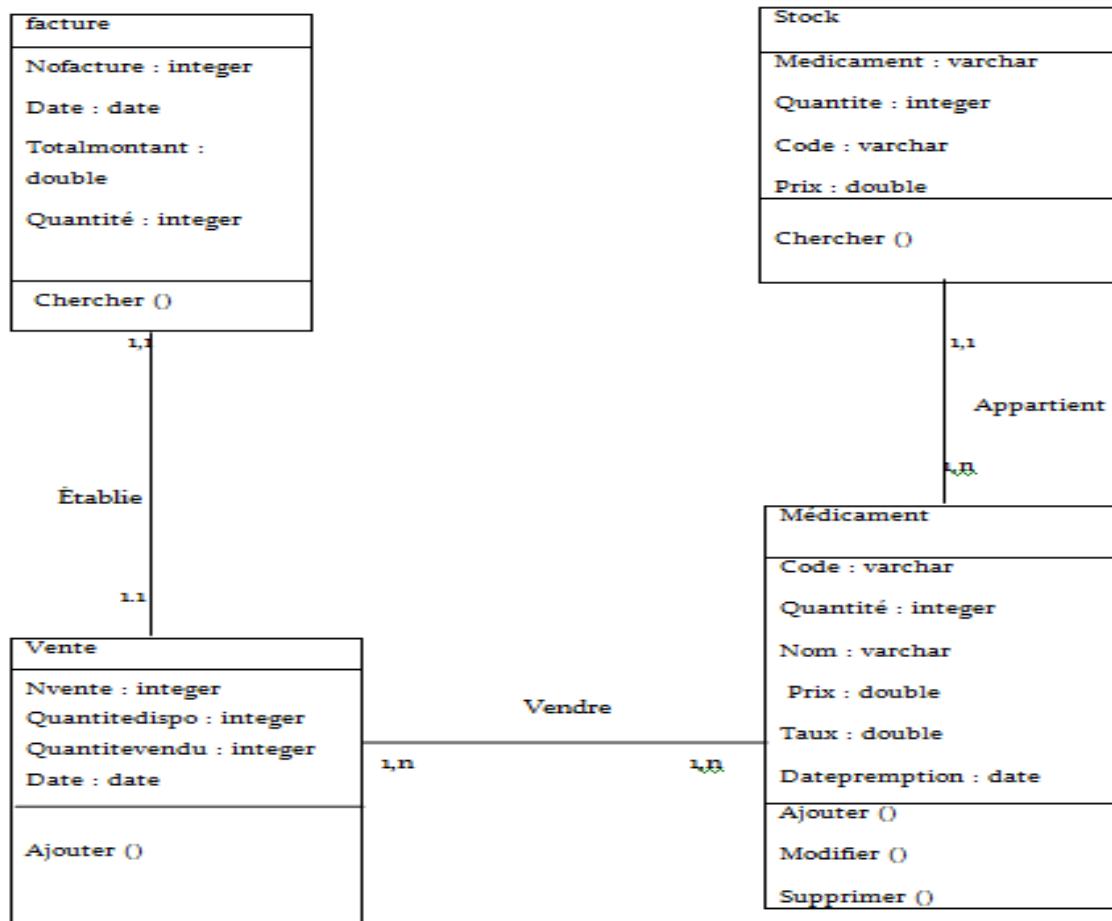


Figure II.21 : Le diagramme de classe

8. Conclusion

Durant cette phase, j'ai achevé l'analyse de tous les cas d'utilisations via la description des différents diagrammes de séquences, de cas d'utilisation et l'exposition du diagramme de classe. J'ai entamerai dans la prochaine phase, la construction du système en présentant l'environnement technique, et une description des différentes interfaces de l'application.

CHAPITRE 3: ETUDE TECHNIQUE ET IMPLEMENTATION

1. Introduction

Dans ce chapitre on va s'intéresser à la présentation de l'environnement matériel et logiciel utilisés pour la réalisation de l'application.

Il s'agit en plus de décrire les étapes de mise en œuvre de l'application ainsi que les différentes interfaces permettant l'interaction entre l'utilisateur et le système à développer et décrivant les différentes phases suivies pour la réalisation.

2. Outils et langage utilisé :

2.1. Environnement de réalisation :

Dans cette partie, on va présenter :

- L'environnement matériel
- L'environnement logiciel

2.1.1. L'environnement matériel :

Pendant la phase de documentation, de spécification des besoins, de conception et de développement, on a utilisé un PC ayant les caractéristiques suivantes :

1. Processeur Intel® Pentium® i3 CPU .

2. 4Gb MB de mémoire vive.
 3. Disque dur de capacité 500 Go .
 4. Système d'exploitation Microsoft Windows 8 Professionnel.
 5. Ecran 17 pouce .

2.1.2. L'environnement logiciel :

a) NetBeans 7.3 :

Pour la réalisation de notre application JAVA en utilisant l'environnement de développement NetBeans 7.3. NetBeans est un environnement de développement intégré (IDE) pour JAVA, placé en Open Source par Sun en juin 2000 sous licence CDDL et GPLv2 et qui offre de nombreuses fonctionnalités. Il comporte notamment une série d'outils permettant la conception, le développement, l'exécution, le débogage et le déploiement des applications Java. NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, PHP, HTML [5].

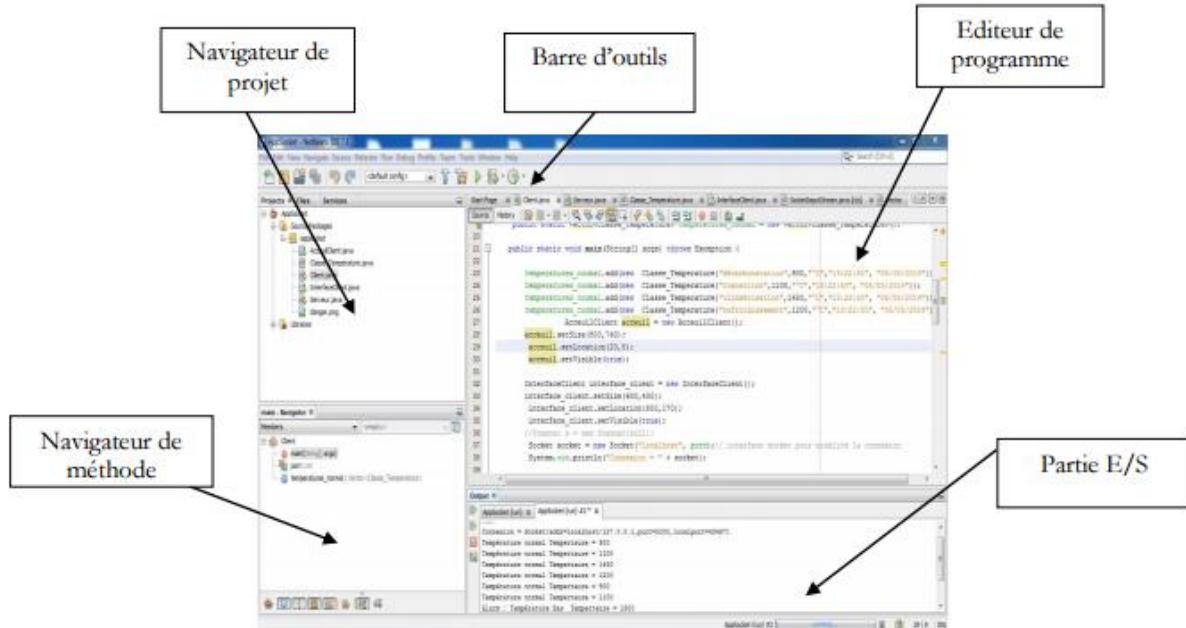


Figure III.1 : Présentation générale du logiciel de programmation NetBeans IDE 7.3

b) Le langage Java :

Java est un langage de programmation informatique orienté objet, la particularité et l'objectif central de java est que les logiciels écrit dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation. Java a donné naissance à un système d'exploitation (Java OS), a des environnements de développements (JDK), des machines virtuelles (MSJVM, JRE) applications multiplateformes (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes(Oracle SQL...), des technologies web(applets, servlets) et une déclinaison pour l'entreprise (La J2EE) .Le point essentiel en java c'est la portabilité de ses applications , elles peuvent être exécuté sur n'importe quelle machine [6].

c) Base de données :

Une base de données (en anglais : *database*) est un outil permettant de stocker et de retrouver l'intégralité de données brutes ou d'informations en rapport avec un thème ou une activité ; celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles [7].

Dans la très grande majorité des cas, ces informations sont très structurées, et la base est localisée dans un même lieu et sur un même support. Ce dernier est généralement informatisé.

La base de données est au centre des dispositifs informatiques de collecte, mise en forme, stockage, et utilisation d'informations. Le dispositif comporte un système de gestion de base de données (abr. SGBD) : un logiciel moteur qui manipule la base de données et dirige l'accès à son contenu. De tels dispositifs — souvent appelés base de données — comportent également des logiciels applicatifs, et un ensemble de règles relatives à l'accès et l'utilisation des informations [7].

d) Système de Gestion de Bases de Données :

SQLite est le moteur de base de données le plus distribué au monde, grâce à son utilisation dans de nombreux logiciels grand public comme Firefox, Skype, Google Gears, dans certains produits d'Apple, d'Adobe et de McAfee et dans les bibliothèques standards de nombreux langages comme PHP ou Python. De par son extrême légèreté (moins de 300 Kio)[8].

e) Langage de Manipulation de Bases de Données :

Le langage de requête spatiale a été conçue comme une extension minimale à la partie interrogative de SQL et distingue des extensions précédemment conçus SQL par: la préservation des concepts SQL; le traitement de haut niveau des objets spatiaux; et l'intégration des opérations spatiales et les relations. Il se compose de deux éléments, un langage de requête pour décrire les informations à récupérer et une langue de présentation pour spécifier comment afficher les résultats de la requête. Les utilisateurs peuvent demander des requêtes SQL standard pour récupérer des données non spatiales sur la base de contraintes non spatiales [9].

3. Conception de la base de données :

Notre base de données est composée de 8 tables, et voici les listes des tables de la base de données :

Tables (4)
▷ facture
▷ médicaments
▷ seconnecter
▷ utilisateur

Figure III.2-Liste des tables de la base de données.

*La table « facture » : contient les informations sur la facture de vente.

Elle contient aussi les champs suivants :

- « No. facture » : un entier représentant le numéro de la facture ;
- « Code» : une chaîne de caractère pour le code du médicament ;
- « Nom» : une chaîne de caractère pour le nom du médicament ;
- « Quantité » : un entier représentant la quantité du produit;
- « Prix » : un entier pour le prix du produit ;
- « Taux » : de type double pour le la quantité minimale;
- « Date pér » : de type date pour enregistrer la date de péremption ;
- « Date» : de type date pour la date de vente des produits ;
- « Montant total» :de type double pour le prix total des produits ;

*La table « médicaments »: contient les informations sur les médicaments présents dans le stock :

Elle contient aussi les champs suivants :

- « Code » : une chaîne de caractère pour le code de médicament;
- « Quantité» : un entier pour la quantité disponible des produits ;
- « Nom» :Une chaîne de caractère pour le prix total des produits ;
- « Prix » : un entier pour le prix du produit ;
- « Taux» : de type double pour le la quantité minimale;
- « Date per» : de type date pour enregistrer la date de péremption ;

*La table « seconnecter » : contient les informations d'authentification sur les utilisateurs de l'application :

Elle contient aussi les champs suivants :

« Id» : une chaine de caractère pour l'identifiant de l'utilisateur;

« motdepasse» : une chaine de caractère pour le mot de passe de l'utilisateur;

*La table « nom_util » : contient les noms d'utilisateur de l'application

Elle contient aussi les champs suivants :

« Id» : une chaine de caractère pour le nom d'utilisateur;

4. Réalisation du projet :

Notre application de gestion de pharmacie permet de gérer les produits, la vente, etc.... et d'offrir à l'utilisateur quelques accessoires à savoir la date actuelle.

La multitude des tâches que notre application est capable de faire engendrer un grand nombre de fenêtres.

On va essayer de sélectionner quelques fenêtres qui nous paraissent importantes pour les intégrer dans le présent mémoire.

4.1. Interface d'authentification

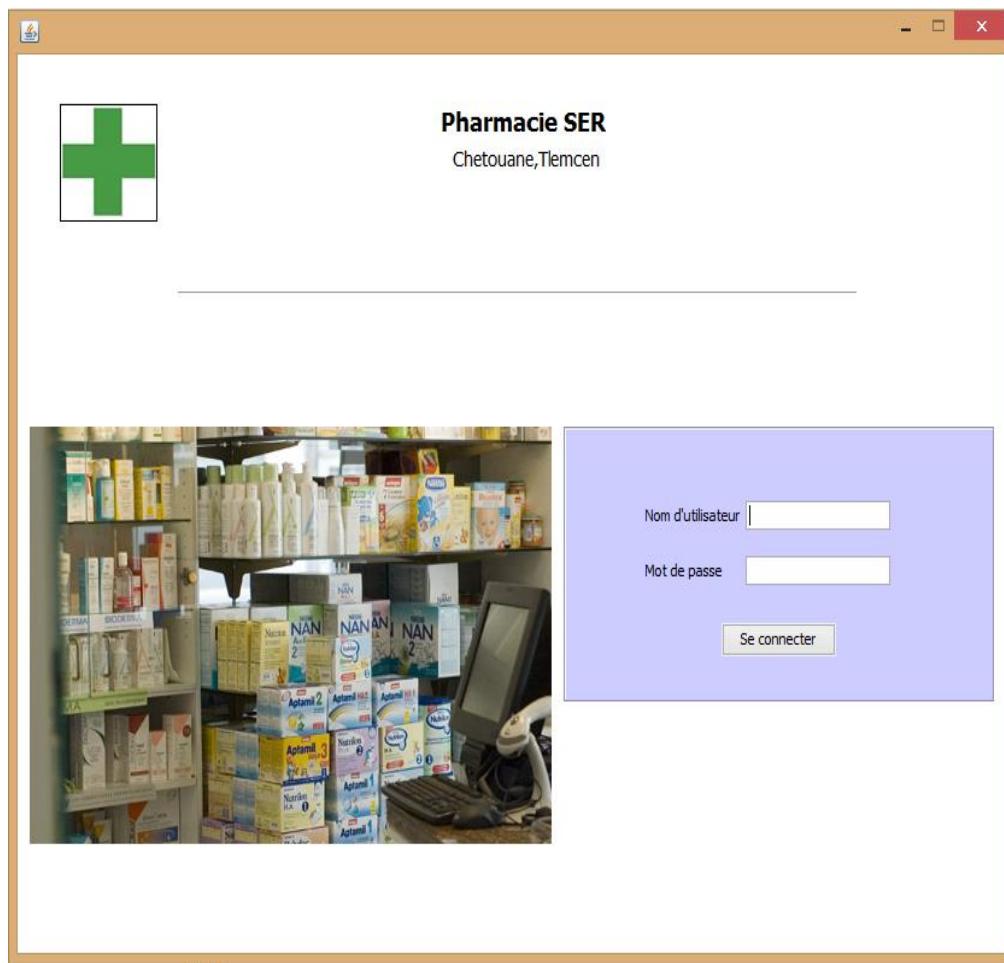


Figure III.3 : L'interface d'authentification

Cette première capture présente l'interface d'authentification dans laquelle on doit choisir le nom d'utilisateur et entrer le mot de passe pour commencer à utiliser notre application. À travers cette fenêtre l'utilisateur s'authentifie pour utiliser l'application. Cette étape met en valeur l'aspect sécurité.

Si ces données sont fausses un message d'erreur s'affiche :

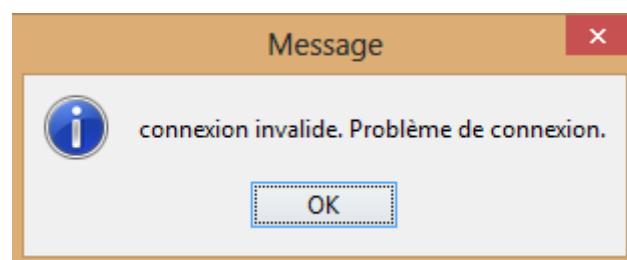


Figure III.4 : message d'erreur d'authentification

4.2. Menu principale

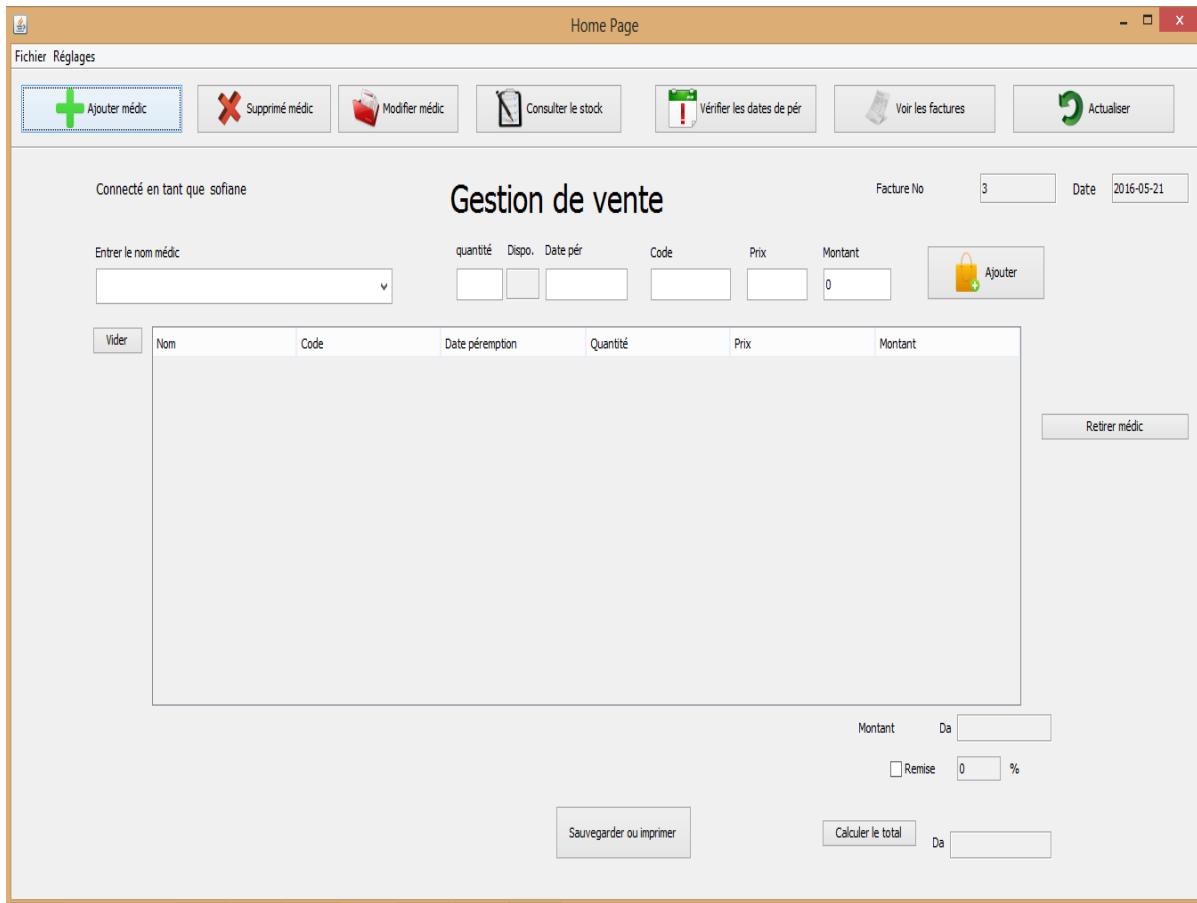


Figure III.5 : interface du menu principale

Cette interface représente la page d'accueil de notre application offrant à l'utilisateur de multiples choix. J'ai veillé à ce que les interfaces soient assez conviviales: le choix des couleurs, l'ergonomie et la clarté du contenu permettent de faciliter son exploitation par l'utilisateur.

4.3. Les Fenêtres Gestion des Produits :

a) Ajouter un médicament :

Cette interface (figure III.6) permet à l'utilisateur d'ajouter un nouveau médicament avec la saisie de toutes les informations qui lui sont relatives.

Fichier Modifier

La saisie des données

Ajouter un nouveau médicament

Si le code apparaît, le médicament est déjà dans la bdd

Nom de médicament Code Trouver les détails

Date de péremption Janvier 2016

Stock actuel:

Quantité Taux Prix

 Sauvegarder et quitter

 Retour

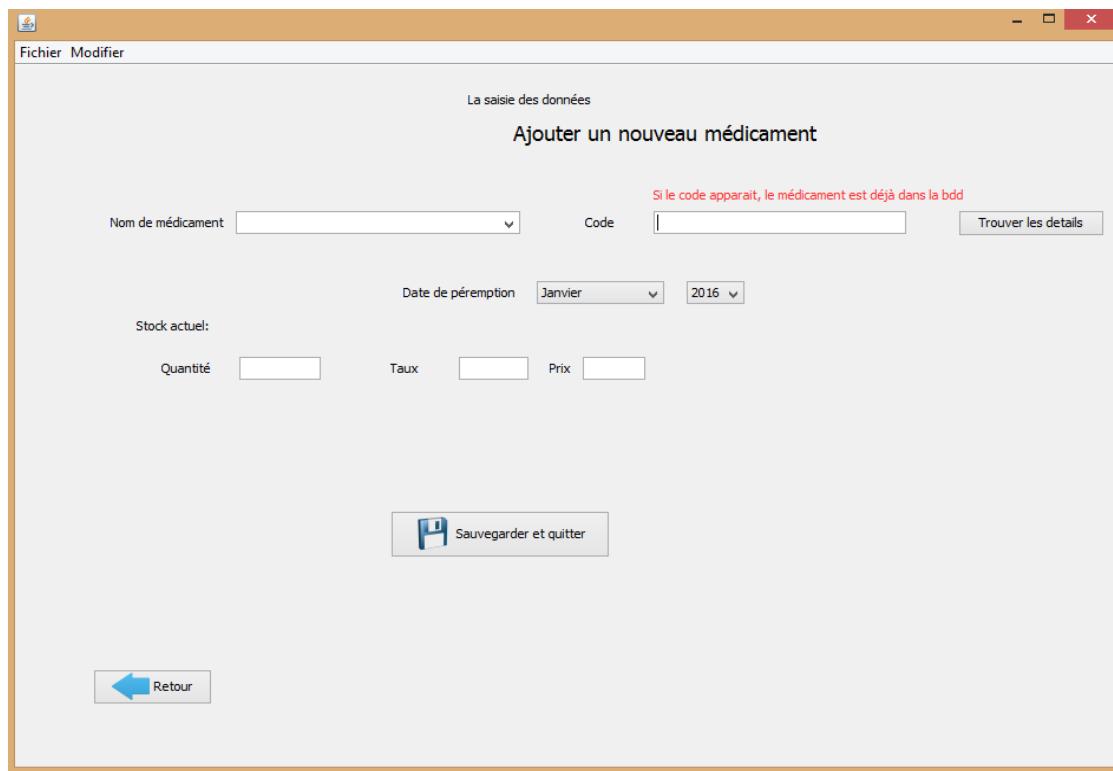


Figure III.6 : interface d'ajout des médicaments

b) Modifier un médicament :

Saisie de données

Modifier Médic/Stock

Actualiser

Chercher le nom médic Chercher

Nouveau nom de médic Code

Stock actuel aaaa-mm-jj

Quantité Date pér Taux Prix

 Sauvegarder

 Retour

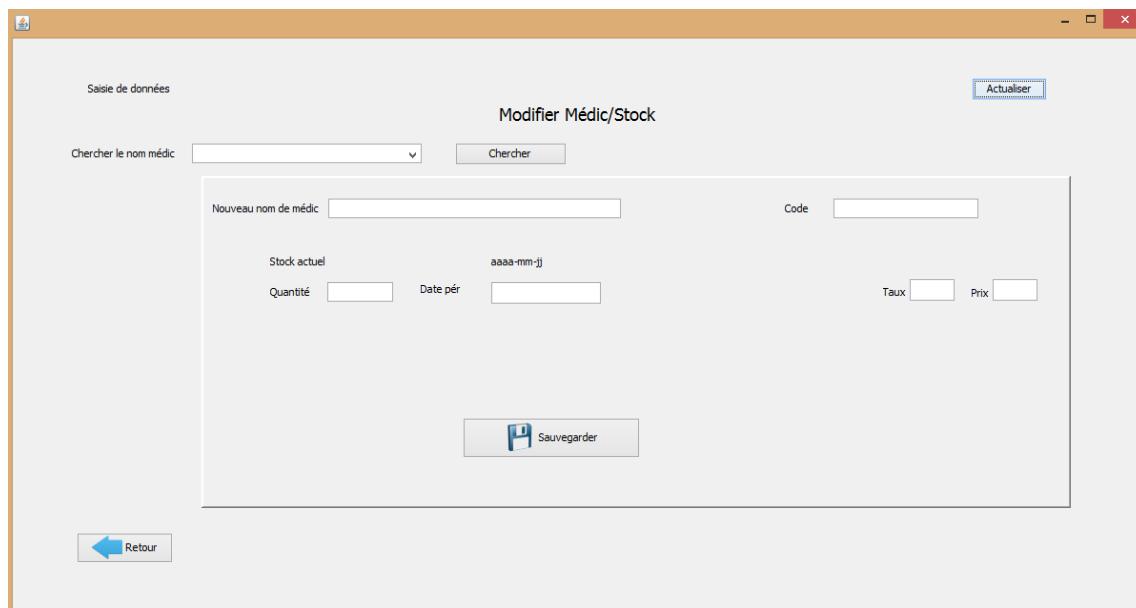


Figure III.7: interface de modification des médicaments

L'interface précédente permet de modifier toutes les données d'un médicament après la recherche de ce dernier.

c) Supprimer un médicament :

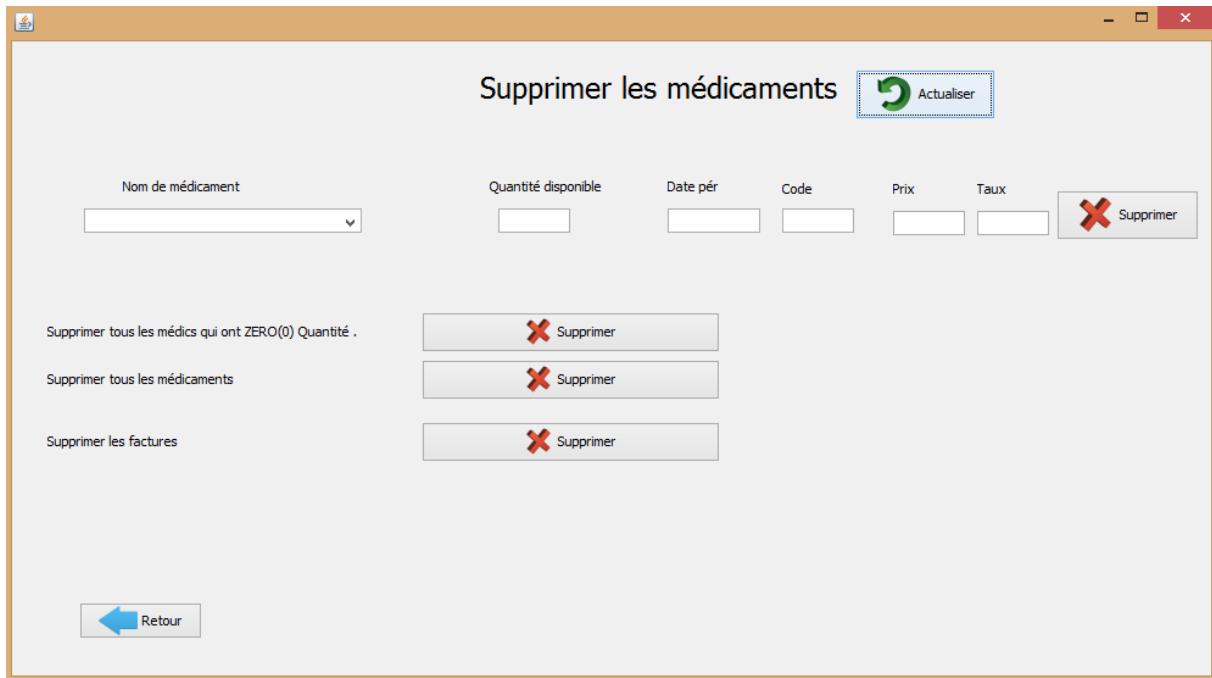


Figure III.8: interface de suppression

Cette fenêtre offre de multiple choix de suppression :

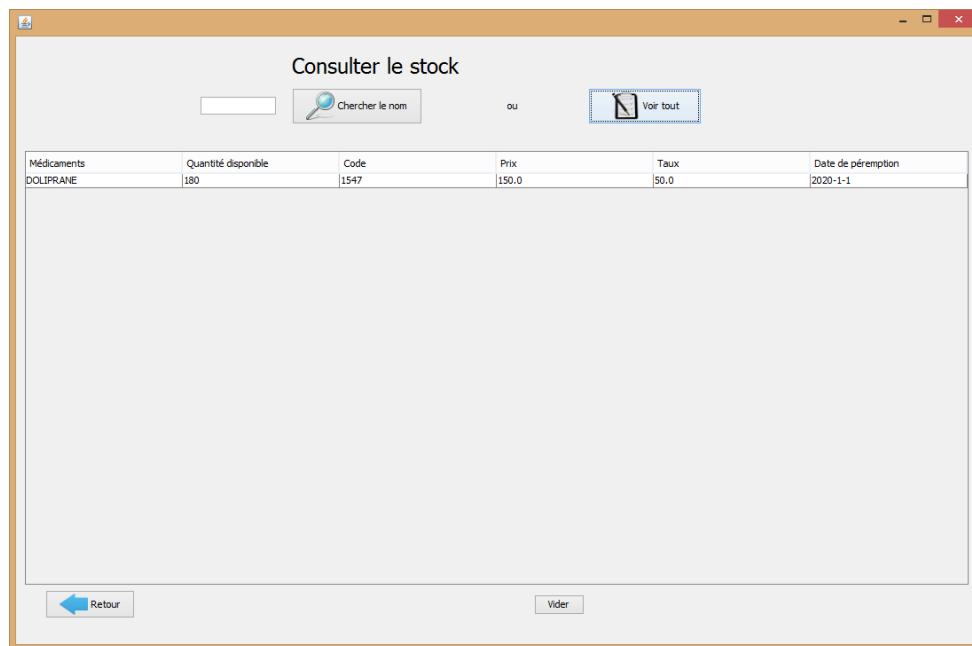
*Suppression du médicament à l'aide de son nom.

*Suppression des médicaments qui ont zéro quantité.

*Suppression de tous les médicaments.

*Suppression de toutes les factures.

d) Consulter le stock :

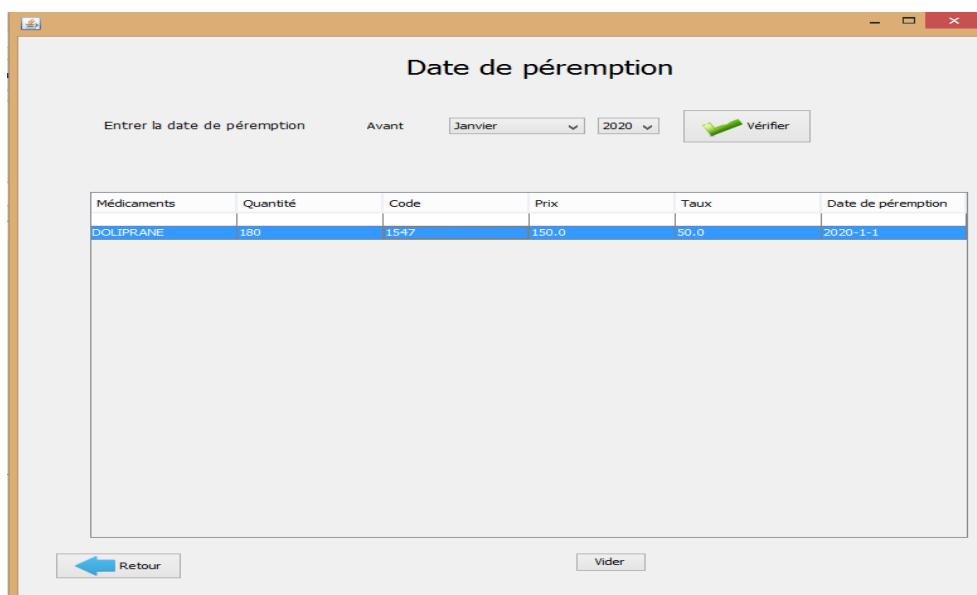


Médicaments	Quantité disponible	Code	Prix	Taux	Date de péremption
DOLIPRANE	180	1547	150.0	50.0	2020-1-1

Retour Vider

Figure III.9: interface de consultation du stock

e) Vérifier les dates de péremption :



Médicaments	Quantité	Code	Prix	Taux	Date de péremption
DOLIPRANE	180	1547	150.0	50.0	2020-1-1

Retour Vider

Figure III.10: interface de vérification des dates de péremption

L'utilisateur fait entrer une date en choisissant le mois et l'année, le produit qui sera périmé dans cette date s'affiche dans le tableau sinon le tableau reste vide.

f) Les Fenêtres de réglages :

Cette interface contient boutons qui sont les suivants :

Bouton changer mot de passe :

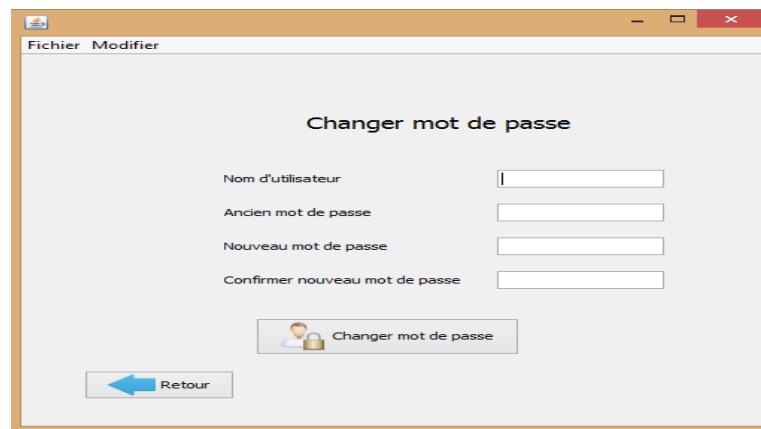


Figure III.11: interface de changement de mot de passe

Bouton changer mot de passe :

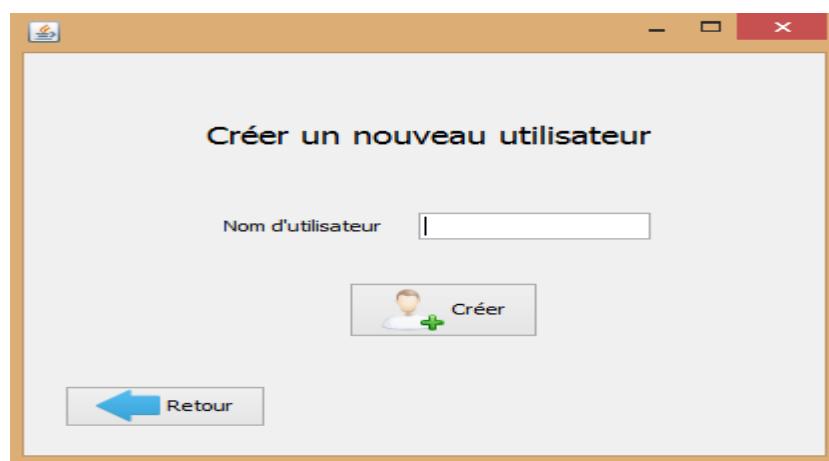


Figure III.12: fenêtre de création des utilisateurs

Bouton de réinitialisation :

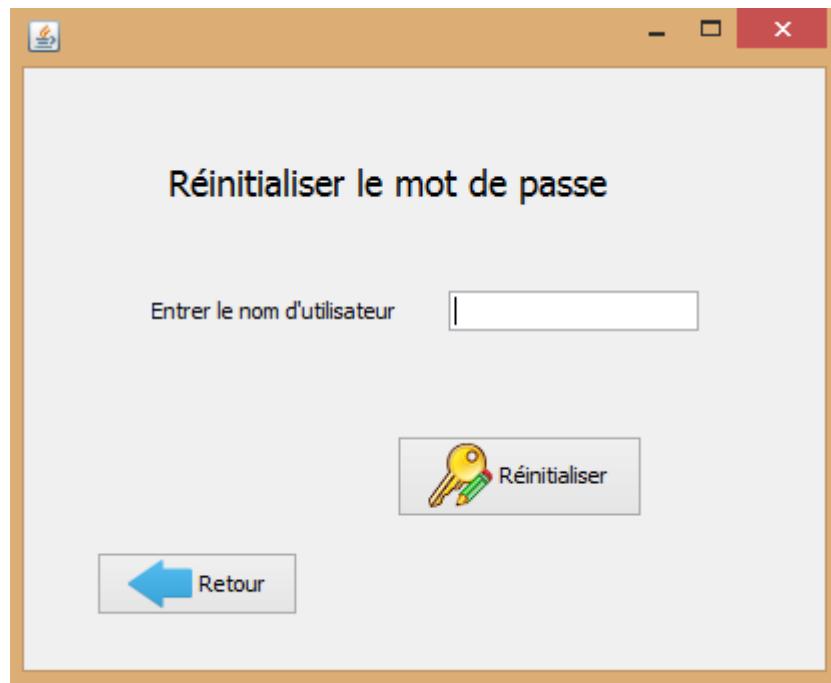


Figure III.13: fenêtre de création des utilisateurs

Bouton supprimer un utilisateur :

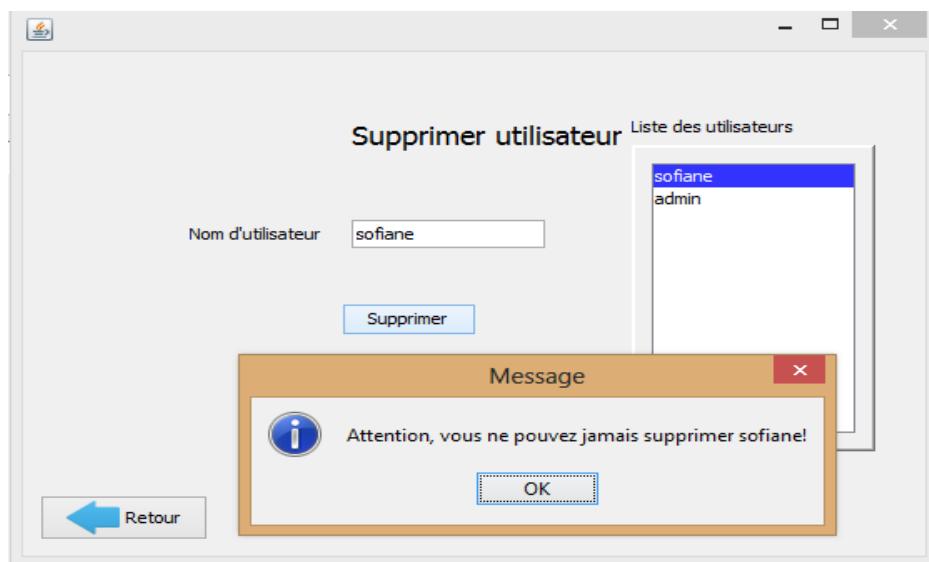


Figure III.14: fenêtre de suppression des utilisateurs

sofiane est défini comme un administrateur (pharmacien), il ne peut pas être supprimé

5. Conclusion :

Dans ce chapitre, nous avons présentés notre environnement de travail matériel et logiciel, les différentes techniques utilisées pour implémenter le système ainsi que les principales interfaces de notre application avec leurs descriptions.

CONCLUSION GENERALE ET PERSPECTIVE

Rappelons que l'objectif de ce travail était d'informatiser l'activité de gestion du système d'informations des pharmacies hospitalières et des dispensaires publiques. Pour cela, on a réalisé une application interactive permettant de gérer les différents traitements de cette activité et de satisfaire les besoins des différents utilisateurs impliqués dans ce processus de gestion.

Le travail est débuté par la compréhension du contexte de mon projet. Ensuite, on a réalisé une étude de l'existant concernant les applications de gestion des activités de la pharmacie, ce qui nous a permis de fixer les anomalies à éviter et les objectifs à réaliser pour avoir un système satisfaisant. Puis, on a passé à la l'étude conceptuelle de mon application selon une approche orientée objet tout en se basant sur le langage UML. Par la suite, on a effectué le codage et l'implémentation de l'application. Enfin on a effectué les tests nécessaires pour valider l'application.

Ce projet a été très bénéfique pour nous car il nous a permis de renforcer et enrichir mes connaissances théoriques dans le domaine de la conception, et de mettre en application mes connaissances acquises le long de mes études. Il nous a encore donné l'occasion de maîtriser le langage de programmation Java, le système de gestion de la base de donnée Sqlite manager et de me familiariser avec la conduite des projets informatiques.

En plus, ce projet était une bonne occasion pour réaliser un travail très concret, avec des objectifs clairs et bien définis et de se familiariser avec l'environnement du travail et de la vie professionnelle.

En perspective, mon application peut être améliorée en ajoutant d'autres fonctionnalités comme **(Gestion des fournisseurs, des commandes, gestion des ordonnances...).**

LISTE DES FIGURES :

Chapitre I : Présentation du projet

Figure I.1 : pharmacie.....10

Chapitre II : Conception

Figure II.1 : exemple de diagramme de cas d'utilisation.....18

Figure II.2 : tableau des cas d'utilisation.....19

Figure II.3 : Diagramme de cas d'utilisation ‘S’authentifier ‘20

Figure II.4 : Diagramme global des cas d'utilisation.....20

Figure II.5 : Diagramme de cas d'utilisation gestion de stock.....21

Figure II.6 : Diagramme de cas d'utilisation gestion des ventes.....21

Figure II.7 : Diagramme de cas d'utilisation gestion des médicaments.....22

Figure II.8 : Table de description textuelle pour cas d'utilisation « s’authentifier »....24

Figure II.9 : Table de sous cas « ajouter un utilisateur».....25

Figure II.10 : Table de sous cas « modifier un utilisateur».....26

Figure II.11 : Table de sous cas « supprimer un utilisateur».....27

Figure II.12 : Table de sous cas « ajouter un médicament».....28

Figure II.13 : Table de sous cas « modifier un médicament».....29

Figure II.14: Table de sous cas « supprimer un médicament».....30

Figure II.15 : Table de cas « gestion de vente».....31

Figure II.16 : Table de cas « gestion de stock».....32

Figure II.17 : Diagramme de séquence de l’authentification.....33

Figure II.18 : Diagramme de séquence de la gestion de vente.....34

Figure II.19 : Diagramme de séquence de la mise à jour du stock (l’ajout)35

Figure II.20 : Diagramme de séquence de la suppression d’un médicament périmé....36

Figure II.21 : Le diagramme de classe.....40

Chapitre III : Etude technique et implémentation

Figure III.1 : Présentation générale du logiciel de programmation NetBeans IDE 7.3..42
Figure III.2 -Liste des tables de la base de données.....44
Figure III.3 : L'interface d'authentification.....47
Figure III.4 : message d'erreur d'authentification.....47
Figure III.5 : interface du menu principale.....48
Figure III.6 : interface d'ajout des médicaments.....49
Figure III.7 : interface de modification des médicaments.....49
Figure III.8 : interface de suppression50
Figure III.9 : interface de consultation du stock.....51
Figure III.10 : interface de vérification des dates de péremption51
Figure III.11 : interface de changement de mot de passe.....52
Figure III.12 : fenêtre de création des utilisateurs.....52
Figure III.13 : fenêtre de création des utilisateurs.....53
Figure III.14 : fenêtre de suppression des utilisateurs.....53

BIBLIOGRAPHIE :

- [1] Grady Booch et James Rumbaugh Ivar Jacobson. Le processus unifié de développement logiciel. Eyrolles, 2000
- [2] Ivar Jacobson, Kurt Bittner, Ian Spence, *Use Case Modeling*, Addison Wesley Professional, 2002 (ISBN 0-201-70913-9)
- [3] Goncalves, G., & Hémery, F. (2000, May). Des cas d'utilisation en UML à la gestion de rôles dans un système d'information. In *INFORSID* (pp. 367-379).
- [4] Muller, P. A., & Gaertner, N. (2000). *Modélisation objet avec UML* (Vol. 514). Paris: Eyrolles.
- [5] <http://www.softonic.com/s/netbeans-ide-7.3>
- [6] <http://www.jmDoudoux.com/java>
- [7] (Carlos Coronel, Steven Morris et Peter Rob, *Database Systems: Design, Implementation, and Management*, Cengage Learning - 2012, (ISBN 9781111969608))
- [8] <http://www.sqlite.org/about.html>
- [9] Egenhofer, M. J. (1994). Spatial SQL: A query and presentation language. *Knowledge and Data Engineering, IEEE Transactions on*, 6(1), 86-95.

RESUME

Depuis le début de construction des pharmacies, les pharmaciens s'orientaient à un objectif identique, celui-ci se résumait à comment arriver à gérer la pharmacie à l'aide de moyens facile et rapides. Mais malgré tous les efforts faites par les pharmaciens, ils restent des difficultés. Dans ce but qu'on a développé une application qui permet au pharmacien de gérer le stock des médicaments, la facturation, les utilisateurs ainsi que le processus de vente. L'environnement de programmation utilisé est Netbeans 7.3. On a eu recours, le long de ce projet au concept UML de conception. Le système a été développé à l'aide du langage Java, utilisant comme système de gestion de base de données SQLite.

SUMMARY

Since the beginning of construction of pharmacies, pharmacists were moving in the same objective, it boiled down to how to get a handle pharmacy with easy and fast ways. But despite all the efforts made by pharmacists. For this purpose we have developed an application allowing the pharmacy to manage the stock of medicines, billing, users and the sales process. The environment of my project is Netbeans 7.3. I had recourse, along this project to the UML design concept. The system was developed using the Java language, using as SQLite database management system.

ملخص

منذ بداية إنشاء الصيدليات، والصيادلة تتحرك في نفس الهدف يتمثل ذلك في كيفية تسخير الصيدلية بالاعتماد على طرق سهلة وسريعة. ولكن على الرغم من كل الجهود التي بذلت من قبل الصيادلة، لا تزال توجد صعوبات

ولهذا الغرض قمنا بتطوير تطبيق يسمح للصيدلية لإدارة مخزون الأدوية، والفوافير والمستخدمين وعملية البيع

بيئة المشروع هي نتبيانس 7.3. كان لي اللجوء، إلى جانب هذا المشروع إلى مفهوم التصميم 1 او الـ

وقد تم تطوير هذا النظام باستخدام لغة جافا، وذلك باستخدام نظام إدارة قاعدة البيانات اسكلبتي

