



APPLICATION PHARMACIE JAVA

REALISER PAR :

abdessamad idboussadel

jamal jabbour

LAYINDE Akanni Saobane

ENCADRER PAR

Prof. Walid

Le contenu du rapport :

- I. Introduction et définition du problème
 - Problématique
 - Contexte du projet
 - Objectifs de l'étude
- II. Objectif du projet
 - Objectif principal
 - Objectifs spécifiques
- III. Description fonctionnelle des besoins
 - Analyse des besoins
 - Spécifications fonctionnelles
- IV. Partie conception UML
 - Diagramme de cas d'utilisation
 - Diagramme de classes
 - Diagramme de séquence d'analyse
 - Diagrammes de séquence de conception
- V. Résultats et démonstration de notre application
 - Architecture de l'application
 - Présentation de l'interface utilisateur
- VI. Bilan et difficultés rencontrées et les points intéressants
 - Bilan qualitatif du travail
 - Difficultés rencontrées
 - Points intéressants et apprentissages
- VII. L'organisation et la répartition des tâches au sein du groupe durant la durée du projet
- VIII. Une conclusion sur l'apport du projet



I. Introduction et définition du problème :

L'invention de l'ordinateur a résolu les problèmes liés à la gestion manuelle des informations sur support papier. Aujourd'hui, l'informatisation des systèmes de gestion de données est essentielle au développement des entreprises, y compris les pharmacies hospitalières et les dispensaires publics. L'ordinateur offre une solution fiable et efficace pour traiter et sauvegarder les informations de manière optimisée.

La croissance du nombre de médicaments hospitaliers nécessite une gestion efficace et rapide, mais la gestion manuelle reste dominante. Cela entraîne une mauvaise organisation du travail, des informations imprécises et une difficulté dans la recherche d'informations et la création de statistiques. Pour remédier à cela, notre projet vise à concevoir et mettre en œuvre une application de bureau fiable et facilement intégrable dans l'environnement de travail des pharmacies.

Notre projet est divisé en trois parties distinctes :

La première partie concerne la description fonctionnelle des besoins, nous permettant ainsi de définir clairement notre problématique et les objectifs à atteindre. La deuxième partie se concentre sur la conception, regroupant toutes les étapes du processus de développement à l'aide du langage de modélisation UML.

Enfin, la dernière partie est dédiée à la réalisation, où nous détaillerons les outils utilisés pour la conception de notre application, en présentant également quelques interfaces. Nous concluons notre travail par une synthèse générale.

II. Objectif du projet :

L'objectif principal de notre projet est de concevoir et développer une application de gestion complète pour une pharmacie. Cette application aura pour but de faciliter la gestion des ventes, du stock et des fournisseurs, en automatisant les processus et en offrant une interface conviviale.

Nous visons à fournir à la pharmacie un outil fiable et efficace pour optimiser ses opérations quotidiennes. Pour atteindre cet objectif, nous utiliserons le langage de programmation Java, en exploitant ses fonctionnalités avancées pour créer une application robuste et performante.

III. Description fonctionnelle des besoins :

1) Page d'authentification :

L'application doit fournir un système d'authentification pour sécuriser l'accès aux fonctionnalités. L'utilisateur (pharmacien) doit s'authentifier en utilisant un nom d'utilisateur et un mot de passe.

2) Gestion des ventes :

a. Ajouter une vente :

L'application doit permettre d'ajouter une vente en spécifiant le numéro de vente, le code du médicament vendu et la quantité vendue.

Avant d'ajouter la vente, l'application doit vérifier si le médicament existe dans la base de données et si la quantité vendue est disponible dans le stock.

b. Annuler une vente :

Il doit être possible d'annuler une vente existante en fournissant le numéro de vente correspondant. Lors de l'annulation d'une vente, la quantité vendue doit être restaurée dans le stock du médicament associé.

c. Modifier une vente :

L'application doit permettre de modifier une vente existante en modifiant la quantité vendue. La modification de la vente doit respecter les mêmes conditions que lors de l'ajout d'une vente (vérification du médicament et de la disponibilité de la quantité).

3) Gestion du stock :

a. Ajouter un médicament :

L'application doit permettre d'ajouter un médicament avec les informations suivantes : code du médicament, libellé, prix unitaire et quantité en stock. Le code du médicament doit être unique dans la base de données.

b. Supprimer un médicament :

Il doit être possible de supprimer un médicament existant en fournissant son code.

c. Modifier un médicament :

L'application doit permettre de modifier les informations d'un médicament existant, telles que le libellé ou le prix unitaire.

d. Rechercher un médicament :

Il doit être possible de rechercher un médicament en fournissant son code. L'application doit afficher toutes les informations sur le médicament recherché, y compris le libellé, le prix unitaire et la quantité en stock.

4) Gestion des fournisseurs :

a. Ajouter un fournisseur :

L'application doit permettre d'ajouter un fournisseur avec les informations suivantes : code du fournisseur, prénom, nom, âge, numéro de téléphone et adresse. Le code du fournisseur doit être unique dans la base de données.

b. Supprimer un fournisseur :

Il doit être possible de supprimer un fournisseur existant en fournissant son code.

c. Rechercher un fournisseur :

Il doit être possible de rechercher un fournisseur en fournissant son code. L'application doit afficher toutes les informations sur le fournisseur recherché, y compris le prénom, le nom, l'âge, le numéro de téléphone et l'adresse.

5) Gestion des erreurs et des exceptions :

L'application doit être capable de gérer les erreurs et les exceptions de manière appropriée, en fournissant des messages d'erreur clairs et en assurant la cohérence des données.

6) Sécurité :

L'application doit permettre à un pharmacien de s'authentifier avec un nom d'utilisateur et un mot de passe pour accéder aux fonctionnalités de gestion des ventes, du stock et des fournisseurs. Seuls les utilisateurs authentifiés peuvent accéder à ces fonctionnalités.

7) Technologies utilisées :

- **Environnement de développement intégré (IDE) :**

NetBeans est un IDE populaire pour le développement d'applications Java. Il offre un large éventail de fonctionnalités telles que l'éditeur de code avancé, la compilation et le débogage, la gestion de projet, l'achèvement automatique du code, l'intégration avec les systèmes de contrôle de version, etc.

NetBeans facilite le processus de développement en fournissant un environnement convivial et intuitif.

- **Langage de programmation :**

Java est un langage de programmation orienté objet largement utilisé. Il offre une syntaxe claire et compréhensible, ce qui facilite la création et la maintenance de code. Java est apprécié pour sa portabilité, ce qui signifie que les applications développées en Java peuvent fonctionner sur différents systèmes d'exploitation sans nécessiter de modifications majeures. De plus, Java dispose d'une vaste bibliothèque standard qui offre des fonctionnalités prêtes à l'emploi pour faciliter le développement d'applications.

- **Framework :**

JavaFX est un framework graphique utilisé pour développer des interfaces utilisateur (UI) riches et interactives. Il fournit des fonctionnalités avancées telles que des contrôles personnalisables, des effets visuels, des animations, des graphiques, etc. JavaFX est intégré à la plateforme Java standard à partir de Java 8, ce qui facilite son utilisation dans les applications Java. Il permet de créer des interfaces utilisateur modernes et attrayantes pour les applications de bureau.

- **Base de données :**

MySQL est un système de gestion de base de données relationnelle open source largement utilisé. Il est réputé pour sa facilité d'utilisation, sa stabilité et ses performances élevées. MySQL utilise le langage SQL (Structured Query Language) pour gérer les données stockées dans la base de données. Wamp et Xamp sont des packages logiciels qui combinent un serveur web Apache, une base de données MySQL et un interpréteur PHP. Ils facilitent l'installation et la configuration d'un environnement de développement local pour le développement d'applications web utilisant MySQL.

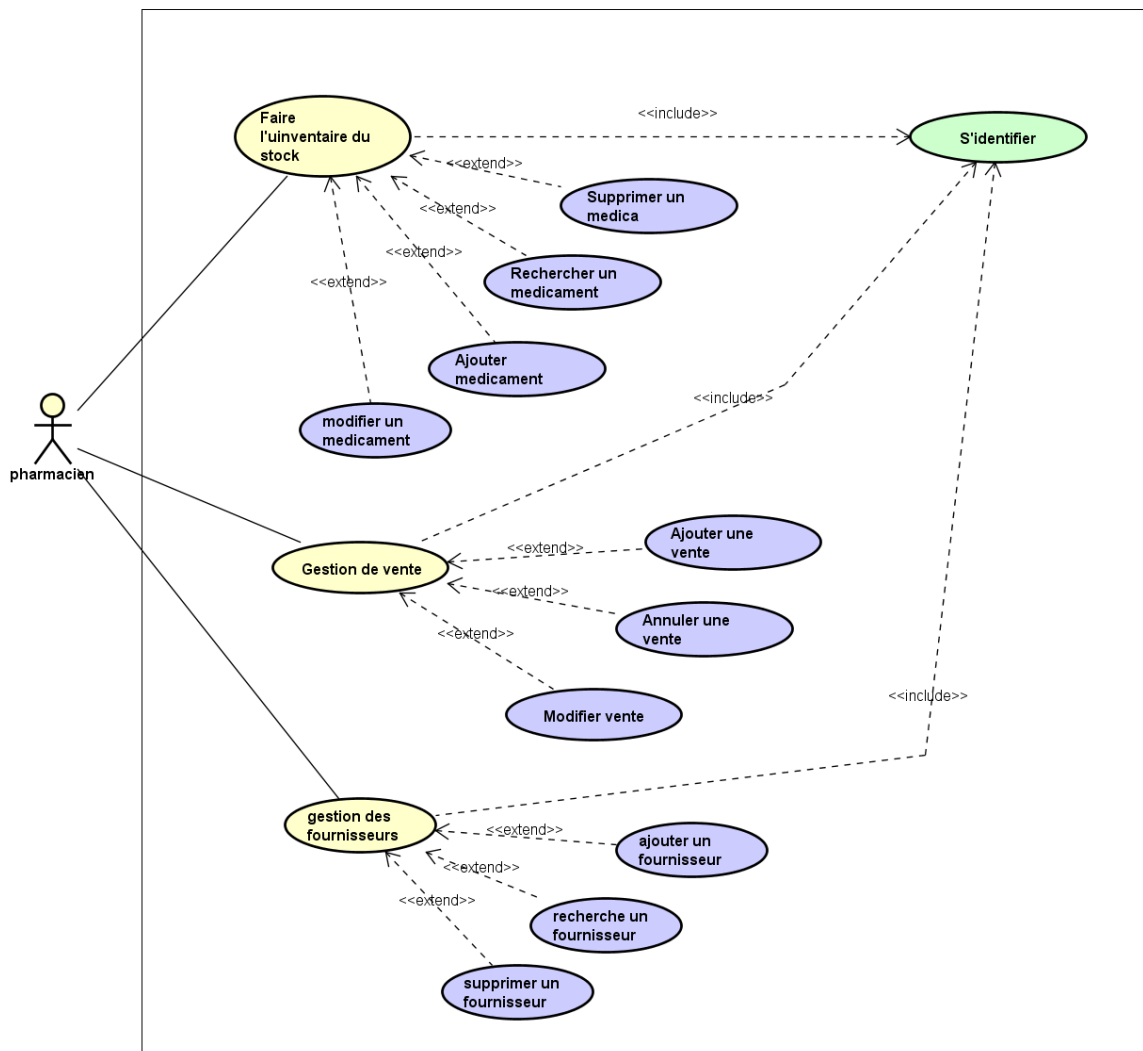
- **Connexion à la base de données :**

JDBC (Java Database Connectivity) est une API standard de Java pour interagir avec des bases de données relationnelles. Elle fournit des classes et des méthodes pour établir une connexion avec une base de données, exécuter des requêtes SQL, récupérer et mettre à jour des données. JDBC permet de gérer les opérations courantes sur la base de données, telles que l'extraction de données pour les afficher à l'utilisateur, l'insertion de nouvelles données et la mise à jour ou la suppression des données existantes. Dans le cas de l'application de gestion de pharmacie, JDBC sera utilisé pour se connecter à la base de données MySQL et exécuter des opérations de gestion des stocks, des ventes, des fournisseurs, etc.

IV. Partie conception UML :

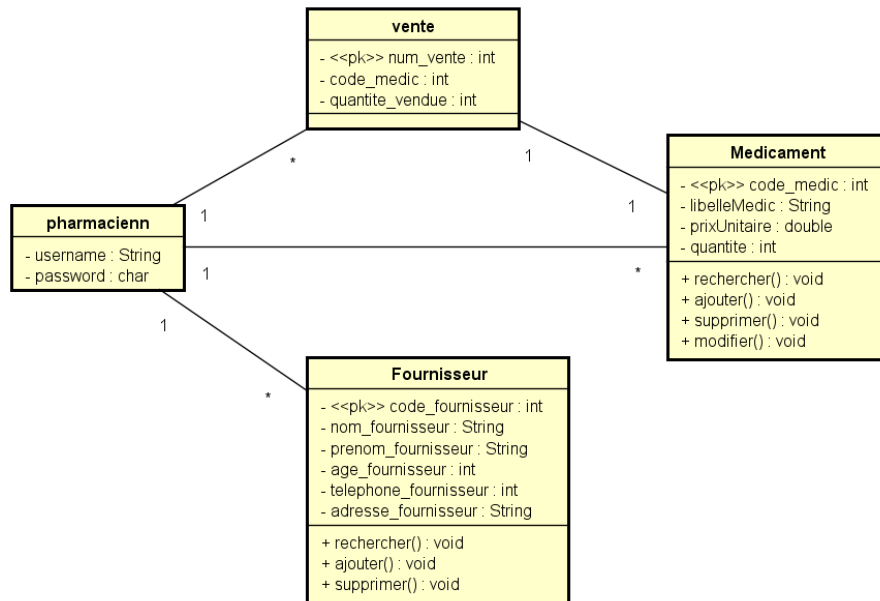
1) Diagramme des cas d'utilisation :

Un diagramme des cas d'utilisation est un outil de modélisation utilisé en ingénierie logicielle pour représenter les interactions entre les acteurs (utilisateurs, systèmes externes) et le système en cours de développement. Il met en évidence les fonctionnalités principales du système du point de vue de l'utilisateur et permet de décrire les différents scénarios d'utilisation.



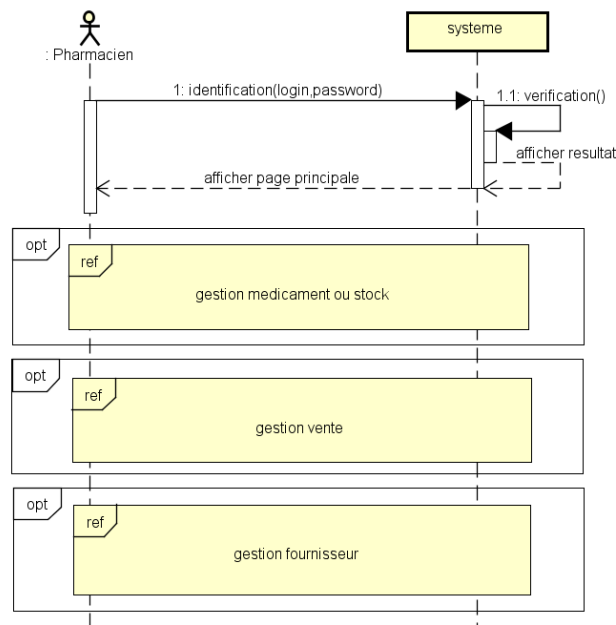
2) Diagramme des classes :

Un diagramme de classe est un type de diagramme de modélisation statique utilisé en ingénierie logicielle pour représenter la structure et les relations entre les classes d'un système. Il offre une vue conceptuelle des entités principales du système, de leurs attributs et de leurs relations. Le diagramme de classe est utilisé pour visualiser la structure du système, les relations entre les classes et les attributs et méthodes de chaque classe. Il aide les développeurs à comprendre la conception du système, à identifier les classes et les relations nécessaires, et à faciliter la communication entre les membres de l'équipe de développement. Le diagramme de classe est souvent utilisé comme point de départ pour la conception et l'implémentation du système. Les attributs d'une classe sont représentés avec leur nom et leur type de données, tandis que les méthodes sont indiquées avec leur nom, leurs paramètres et leur type de retour.



3) Diagramme de séquence d'analyse :

Un diagramme de séquence d'analyse, également appelé diagramme de séquence de système, est un type de diagramme de séquence utilisé en ingénierie logicielle pour modéliser la séquence des interactions entre les acteurs et le système pendant une fonctionnalité ou un scénario spécifique. Le diagramme de séquence d'analyse met l'accent sur la manière dont les objets interagissent entre eux pour réaliser un objectif ou accomplir une tâche spécifique. Il montre la chronologie des messages échangés entre les objets pendant l'exécution du scénario, en mettant en évidence les appels de méthode, les réponses et les échanges de données.

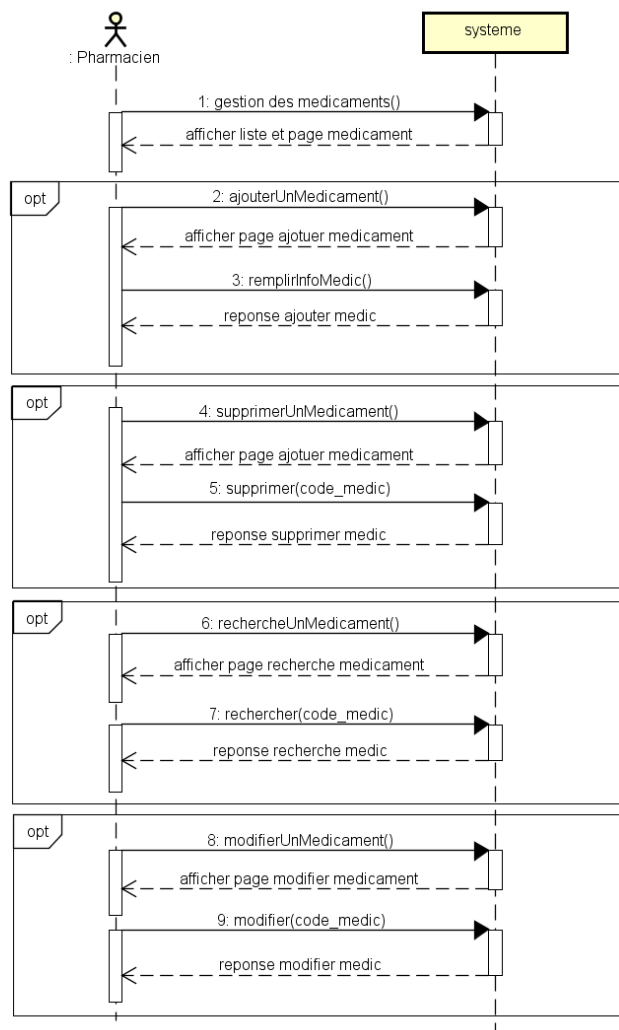


4) Diagrammes de séquence de conception :

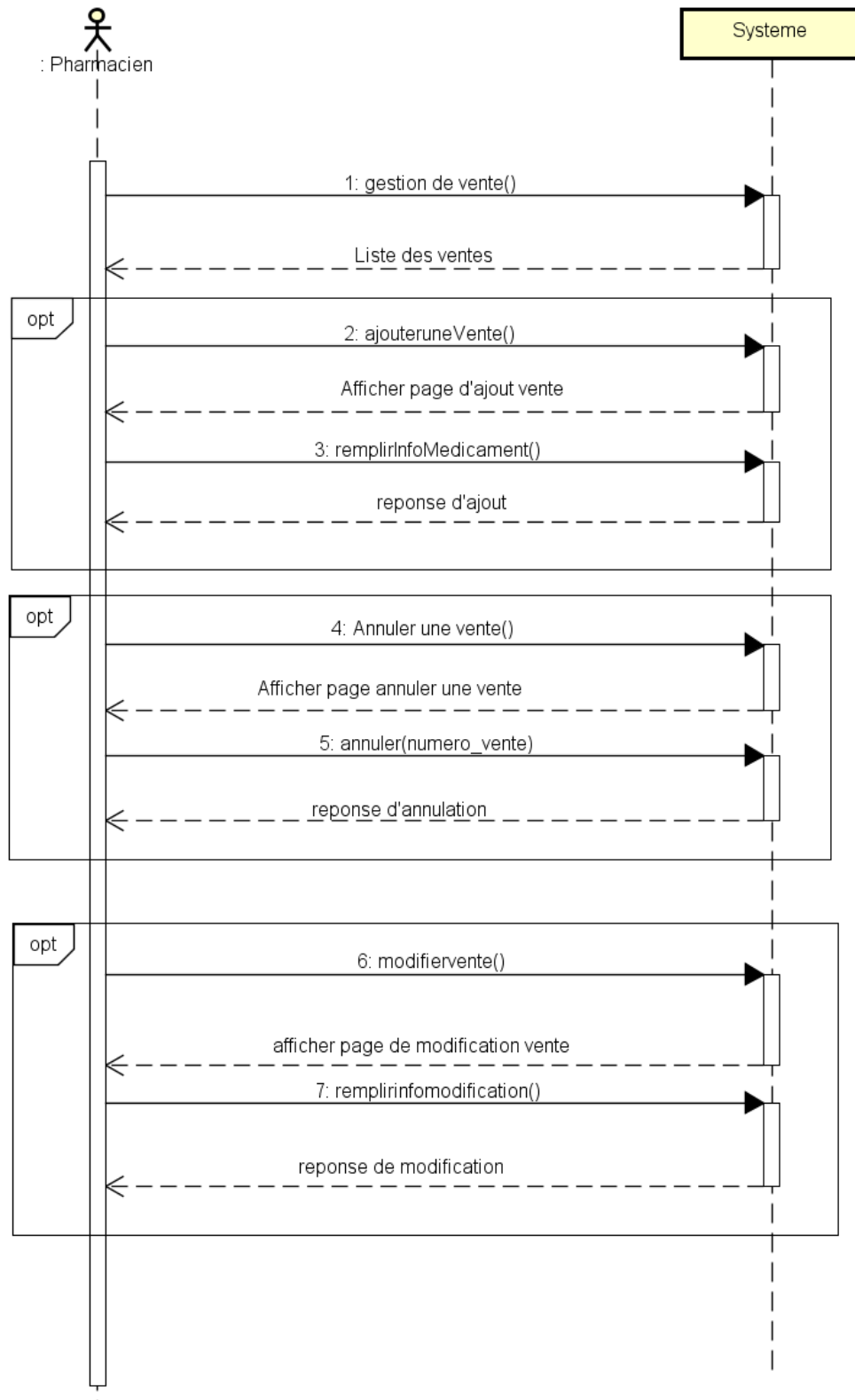
Un diagramme de classe de conception est un type de diagramme de modélisation statique utilisé en ingénierie logicielle pour représenter la structure et les relations entre les classes d'un système lors de la phase de conception.

Le diagramme de classe de conception est plus détaillé que le diagramme de classe initial et il se concentre sur les détails de mise en œuvre du système. Il inclut généralement des informations telles que les attributs, les méthodes, les visibilité, les types de données, les associations, les héritages, les interfaces, les packages, etc. Il fournit une vue plus précise des classes et de leurs interactions.

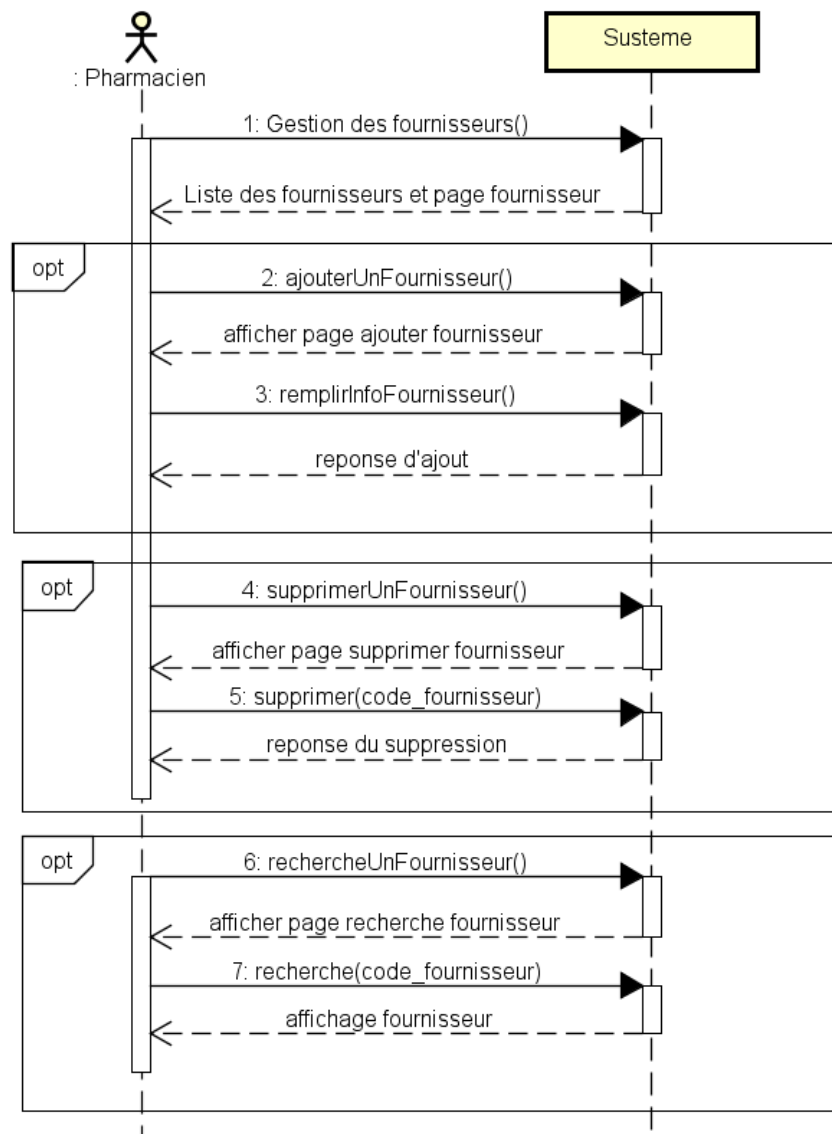
- D. de gestion médicament :



- D. de gestion vente :



- **D. de gestion fournisseur :**



V. L'organisation et la répartition des tâches au sein du groupe durant la durée du projet :

Nous avons utilisé une approche collaborative, en tenant compte des objectifs du projet et des délais à respecter. Des réunions régulières ont été organisées pour discuter de l'avancement du projet, partager les idées et prendre des décisions collectives. Chaque membre était responsable d'une partie spécifique du projet, mais nous avons également travaillé en équipe pour résoudre les problèmes et soutenir les uns les autres lorsque cela était nécessaire.

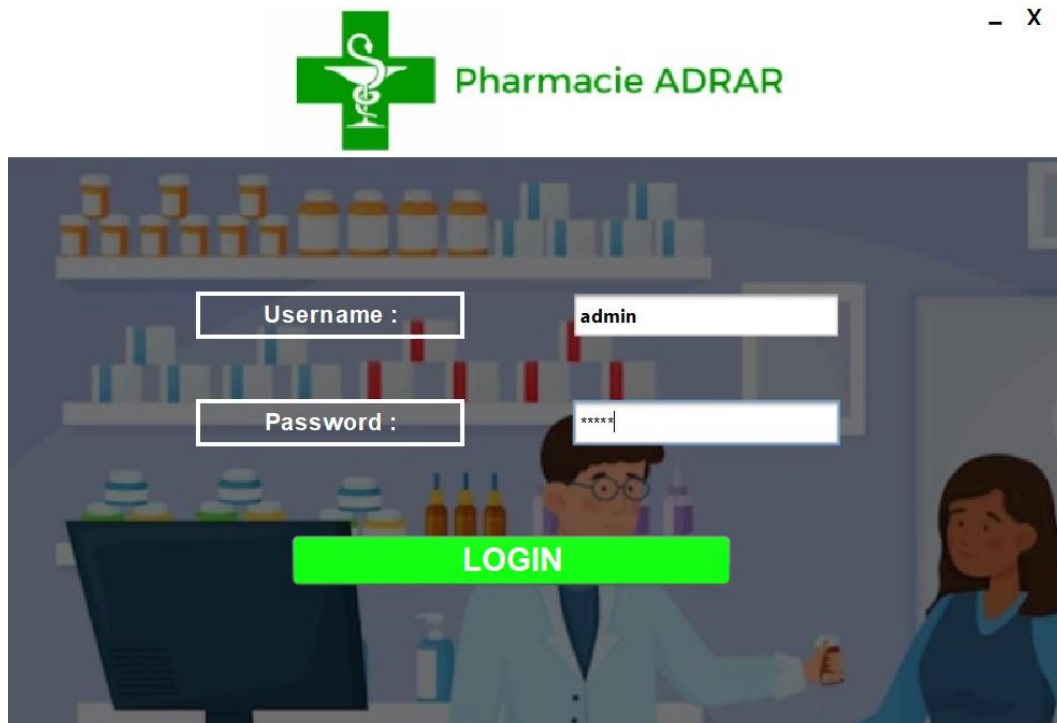
Durant la durée du projet ont suivi les étapes suivantes :

- Élaboration du cahier des charges.
- Conception et modélisation de l'application.
- Développement de la partie frontend.
- Développement de la partie backend.

- Intégration et tests de l'application.
- Correction des anomalies et améliorations.
- Finalisation et déploiement de l'application.

VI. Résultats et démonstration de notre application :

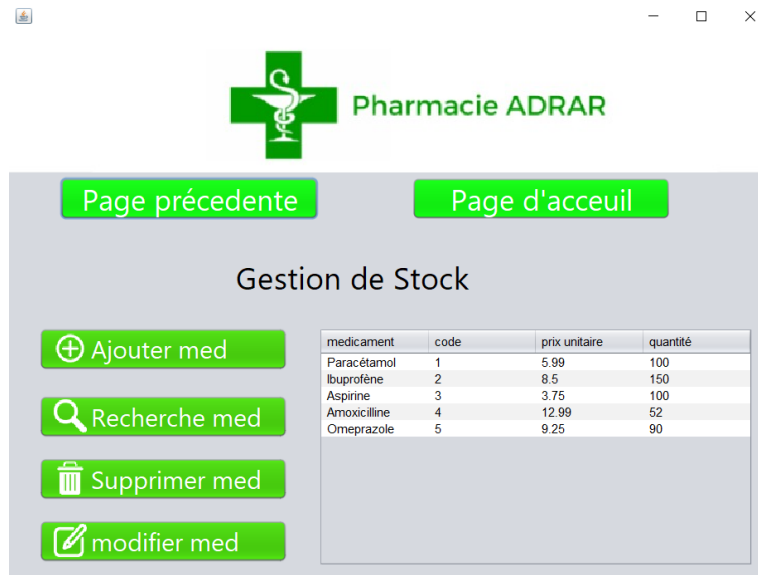
- Page login :



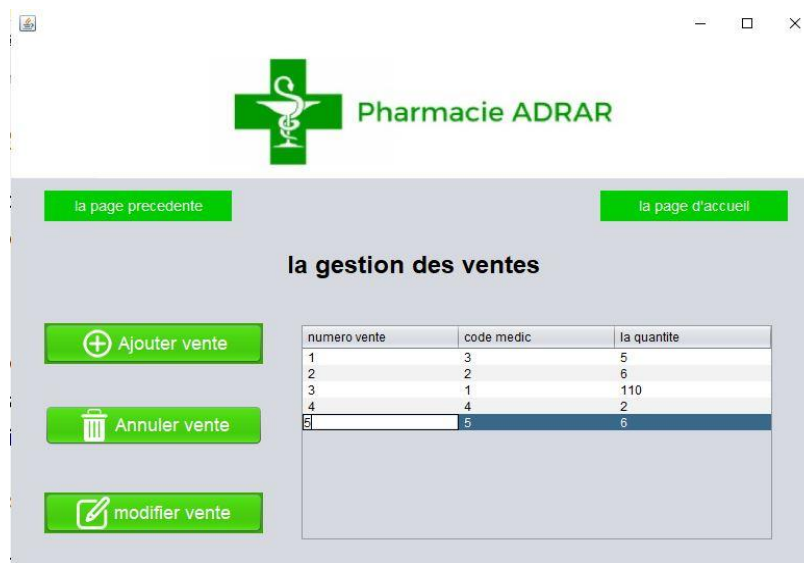
- Page principale :



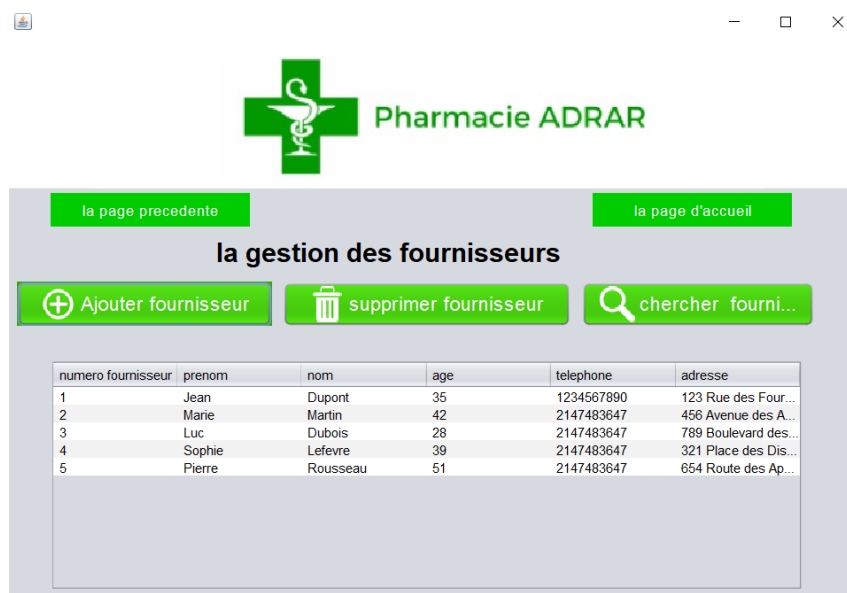
- Page gestion médicament :



- Page gestion vente :



- Page gestion fournisseur :



VII. Bilan et difficultés rencontrées et les points intéressants :

1. Bilan qualitatif du travail :

Notre travail s'est révélé satisfaisant dans l'ensemble, avec une application fonctionnelle répondant aux exigences du client. Nous avons maintenu une rigueur de qualité tout au long du développement, en suivant les meilleures pratiques et en effectuant des tests réguliers.

Un soin particulier a été apporté à la conception de l'interface utilisateur, visant à offrir aux utilisateurs une expérience agréable et intuitive. Nous avons également cherché à créer un design moderne, simple et professionnel, afin de rendre l'application attirante.

2. Difficultés rencontrées :

- Nous avons été confrontés à des défis techniques, tels que l'intégration de différentes technologies (Java, JavaFX, MySQL, NetBeans).
- La coordination de l'équipe a présenté des défis, notamment en ce qui concerne la répartition des tâches et la communication entre les membres. En raison de notre manque d'expérience individuelle.

3. Points de vue intéressants :

Découverte de nouvelles technologies : Le projet nous a permis d'explorer et d'apprendre de nouvelles technologies telles que Java, JavaFX et MySQL, enrichissant ainsi notre bagage technique.

Impact sur notre compréhension du domaine : Le projet nous a donné une meilleure compréhension des défis et des exigences du domaine informatique et pharmaceutique, renforçant ainsi nos connaissances acquises en classe grâce à une expérience pratique concrète.

VIII. Une conclusion sur l'apport du projet :

1. Aspect technique :

- Ce projet a permis d'appliquer les concepts et les techniques de programmation en utilisant Java et JavaFX. Des fonctionnalités spécifiques ont été développées, telles que la gestion des médicaments, l'ajout, la modification et la suppression des données dans la base de données.
- Utilisation d'une base de données : on a acquis des compétences dans la gestion de la persistance des données en travaillant avec MySQL (via phpMyAdmin). Cela inclut la création de tables, l'exécution de requêtes SQL et la mise à jour des données. Avec ça on a renforcé nos compétences techniques dans la gestion des données et la manipulation des systèmes de gestion de bases de données.
- Utilisation d'outils de développement : L'utilisation de l'environnement de développement intégré (IDE) NetBeans a facilité le processus de développement, ce qui nous a permis de travailler plus efficacement et de bénéficier de fonctionnalités intégrées pour le développement logiciel.

2. Apport scientifique :

- L'utilisation d'UML pour la modélisation de notre application a été extrêmement bénéfique. Grâce à cette approche, on a amélioré la clarté et la structure de notre conception. Cela a facilité la collaboration entre l'équipe de développement en fournissant une représentation visuelle des composants, classes et relations. En analysant les diagrammes UML, nous avons pris des décisions plus informées et anticipé les impacts des modifications, favorisant ainsi un développement professionnel et efficace.

3. Aspect humain :

- Travail d'équipe : La réalisation du projet en équipe vous a permis de collaborer avec d'autres membres, de partager les tâches, de résoudre des problèmes ensemble et d'apprendre à travailler en harmonie.
- Communication et gestion du projet : on a organisé des réunions, défini des objectifs, géré les délais et communiqué régulièrement pour assurer la progression du projet. Cela nous a permis d'acquérir des compétences en gestion de projet et en communication interpersonnelle.
- Résolution de problèmes : Tout au long du projet, nous avons été confrontés à des défis et des problèmes techniques. Cette expérience nous a permis d'améliorer notre capacité à identifier, analyser et résoudre efficacement les problèmes, ce qui est une compétence essentielle dans le domaine de l'ingénierie.

Conclusion :

En conclusion, ce projet de développement d'une application de gestion de pharmacie a été une expérience enrichissante. Nous avons pu mettre en pratique nos connaissances en programmation, en conception logicielle et en gestion de bases de données. Nous avons également acquis de nouvelles compétences techniques et renforcé notre capacité à résoudre des problèmes et à travailler en équipe. L'application développée répond aux besoins fonctionnels et offre une interface conviviale pour les utilisateurs. Nous sommes fiers des résultats obtenus et sommes convaincus que cette expérience nous servira dans notre parcours professionnel. Nous tenons à remercier notre encadrant, nos collègues et tous ceux qui ont contribué à la réalisation de ce projet.