

المدرسة الوطنية للعلوم التطبيقية - أكادير
+ⵍⵎⵓⵏⵉ ⵜⴰⵎⴰⵏⵏⵉⵜ ⵜⴰⵏⵓⵙⵏⵉⵜ ⵜⴰⵏⵓⵙⵏⵉⵜ - ⵎⴰⵎⴰⵔⵓⵏⵉ.
ÉCOLE NATIONALE DES SCIENCES APPLIQUÉES - AGADIR



Projet Java

Application desktop de la gestion d'une pharmacie

Première année cycle d'ingénieur

Génie informatique

Présenté par :

AIT LAKHLIFT Tariq

TALEB Nour Eddine

TALEB Amine

Encadré par :

Mr A. Walid

Année universitaire 2022-2023

Table des matières

| | |
|--|----|
| Table des matières | 2 |
| Table des figures..... | 3 |
| Liste des tableaux | 4 |
| 1 Introduction..... | 5 |
| 2 Présentation du projet | 6 |
| 2.1 Objectif du projet..... | 6 |
| 2.2 Contexte et définition du projet..... | 6 |
| 3 Cahier des charges | 7 |
| 3.1 Introduction | 7 |
| 3.2 Spécifications de l'application..... | 7 |
| 3.3 Fonctionnalités..... | 7 |
| 3.4 Conclusion | 9 |
| 4 Conception | 10 |
| 4.1 Introduction | 10 |
| 4.2 Répartition des tâches à réaliser | 10 |
| 4.3 Diagramme de cas d'utilisation | 11 |
| 4.4 Diagramme de classe | 12 |
| 4.5 Diagramme de séquence | 12 |
| 5 Implémentation | 18 |
| 5.1 Introduction | 18 |
| 5.2 Outils de développement | 18 |
| 5.3 Organisation du code | 19 |
| 5.4 Structure de la base de données | 20 |
| 5.5 Interface utilisateur | 21 |
| 5.6 Difficultés rencontrées et améliorations | 23 |
| 6 Installation..... | 24 |
| 6.1 Introduction | 24 |
| 6.2 Outil utiliser pour création du JAR et spécifications..... | 24 |
| 7 Conclusion | 26 |

Table des figures

| | |
|--|----|
| Figure 1 - Diagramme de cas d'utilisation..... | 11 |
| Figure 2 - Diagramme de classe | 12 |
| Figure 3 - Diagramme de séquence d'authentification | 13 |
| Figure 4 - Diagramme de séquence d'ajout de fournisseur..... | 13 |
| Figure 5 - Diagramme de séquence de modification de fournisseur | 14 |
| Figure 6 - Diagramme de séquence de suppression de fournisseur..... | 14 |
| Figure 7 - Diagramme de séquence d'ajout de produit | 15 |
| Figure 8 - Diagramme de séquence de suppression de produit | 15 |
| Figure 9 - Diagramme de séquence d'ajout de produit | 16 |
| Figure 10 - Diagramme de séquence d'ajout de vente | 17 |
| Figure 11 - Diagramme de séquence d'impression de facture totale | 17 |
| Figure 12 - Quelques commits sur Github | 19 |
| Figure 13 - Structure du code source..... | 19 |
| Figure 14 - Les différentes tables de la base de données | 20 |
| Figure 15 - Icône de l'application | 21 |
| Figure 16 - Interface d'inscription | 21 |
| Figure 17 - Interface d'authentification | 22 |
| Figure 18 - Interface d'accueil | 22 |
| Figure 19 - Spécification sur JAR et JRE et l'icône utilisée..... | 24 |
| Figure 20 - Spécification sur JDK et les arguments de la machine virtuelle | 25 |

Liste des tableaux

| | |
|--|----|
| Tableau 1 - Grille de la fonction gestion des produits..... | 8 |
| Tableau 2 - Grille de la fonction gestion des ventes | 8 |
| Tableau 3 - Grille de la fonction gestion des fournisseurs | 9 |
| Tableau 4 - Liste des JARs utilisés..... | 20 |

1 Introduction

Etant en première année de cycle d'ingénieur, spécialité génie informatique, dans un contexte d'études nous avons choisi comme thème pour notre projet la gestion d'une pharmacie. Ce projet représente une occasion précieuse pour nous d'acquérir une expérience et de mettre en pratique nos connaissances dans le domaine du développement logiciel.

Le but de ce rapport est de décrire les différentes étapes suivies tout au long de la réalisation de notre projet. Nous aborderons les différents axes sur lesquels nous nous sommes concentrés, en détaillant chacun d'entre eux pour donner une vue d'ensemble complète de notre démarche.

Dans un premier temps, nous présenterons le projet dans son ensemble, en fournissant une introduction détaillée et des informations sur les objectifs que nous nous sommes fixés. Cette section permettra de mieux comprendre le contexte dans lequel notre application de gestion de pharmacie a été développée. Ensuite, nous aborderons le cahier des charges de notre application. Nous détaillerons les fonctionnalités attendues, les contraintes techniques et les exigences spécifiques auxquelles nous avons dû répondre. Cette partie du rapport fournira un aperçu clair des attentes et des spécifications du projet. Puis, Les deux derniers principaux chapitres concernent la conception et l'implémentation de notre application desktop de gestion de pharmacie. Nous expliquerons en détail les différentes décisions de conception que nous avons prises, en mettant l'accent sur les diagrammes de cas d'utilisation, de classes et de séquence, mais avant cela on va préciser l'organisation et la répartition des tâches au sein du groupe lors de la réalisation du projet. Nous illustrerons également les fonctionnalités clés de notre application et les choix techniques que nous avons effectués lors de sa mise en œuvre. Enfin, nous concluons ce rapport en dressant un bilan de notre expérience et en soulignant les principaux enseignements que nous avons tirés de ce projet. Nous discuterons des difficultés rencontrées, des solutions trouvées et des améliorations possibles pour l'avenir.

2 Présentation du projet

2.1 Objectif du projet

Dans le cadre de ce projet, on aura l'opportunité de développer l'expertise en matière de développement logiciel à travers une expérience concrète et atteindre certains objectifs.

Premièrement, ce projet est conçu pour nous permettre de maîtriser la chaîne de développement logiciel en acquérant une expérience pratique, ce qui donne un sens concret à l'apprentissage théorique. **Deuxièmement**, développer le savoir-faire et les compétences techniques en mettant en pratique les concepts appris en classe, tels que la programmation orientée objet et la gestion des bases de données. **Troisièmement**, renforcer la capacité d'autoformation et l'autonomie en explorant de nouvelles technologies, en recherchant des solutions et en acquérant de nouvelles compétences de manière indépendante. **Quatrièmement**, améliorer les compétences de communication et de collaboration en travaillant en équipe sur ce projet collectif, en échangeant des idées, en partageant des connaissances et en résolvant des problèmes ensemble. **Cinquièmement**, vivre des situations de réalisation les plus proches possible de la réalité professionnelle en développant un produit destiné à des clients spécifiques, ce qui leur permettra de comprendre les attentes et les exigences des utilisateurs finaux. **Sixièmement**, apprendre d'après les erreurs commises et gérer les imprévus, ce qui favorisera la progression en tant que développeurs et apprendre à résoudre les problèmes et à faire face aux défis. **Finalement**, livrer un produit final de qualité à des clients spécifiques, ce qui donnera une expérience concrète de la satisfaction client et aidera à prendre en compte les retours d'utilisateurs pour améliorer leur produit.

2.2 Contexte et définition du projet

Dans un cadre où la gestion efficace des pharmacies est devenue essentielle, notre projet vise à développer une application desktop de gestion d'une pharmacie. Cette application fournira aux pharmaciens et au personnel médical un outil complet et convivial pour gérer les stocks, les ventes, les clients et les fournisseurs.

Notre objectif principal est de fournir une interface utilisateur intuitive et conviviale qui permettra aux utilisateurs, même non techniciens, de naviguer facilement dans l'application et d'effectuer leurs tâches de gestion de manière efficace.

L'application que nous développerons fournira des fonctionnalités telles que la gestion des stocks, permettant de suivre les médicaments disponibles, les dates d'expiration et les quantités en temps réel. Elle permettra également de gérer les fournisseurs, les ventes et d'établir des factures.

Grâce à ce projet, nous visons à offrir aux pharmaciens une solution logicielle moderne qui les aidera à améliorer leur productivité, à réduire les erreurs et à offrir un service de qualité à leurs patients.

3 Cahier des charges

3.1 Introduction

Le projet vise à développer une application desktop de gestion de pharmacie. Ce chapitre est consacré à la description des spécifications de l'application et les technologies utilisées. En effet, il existe certains besoins qui doivent être satisfaits pour l'application. Par conséquent, il est nécessaire de choisir les bonnes technologies/langages après l'étude des spécifications.

3.2 Spécifications de l'application

Le but principal est d'avoir une application pouvant gérer les ventes, les produits et les fournisseurs en respectant les exigences suivantes :

Chaque produit sera identifié par code. De plus, on enregistre le nom, le prix en DH, la quantité, la catégorie (spécialité médicament, parapharmaceutique, Diététique), la forme (orale, injectable, dermique, inhalée, rectale), la date de fabrication et la date d'expédition.

Chaque fournisseur à son propre identifiant. De plus on connaît le nom, numéro de téléphone, email, type de paiement effectué, cash ou par virement bancaire. Dans le cas de paiement par banque on doit ajouter son RIB.

Pour chaque vente on connaît le nom du client, son CIN, les produits achetés, la quantité achetée (pour chaque produit on insère la quantité demandé).

On cherche aussi à ce que pour chaque employé dans la pharmacie de pouvoir créer son propre compte afin d'utiliser l'application et de réaliser les différentes tâches.

L'application sera développée en utilisant les technologies suivantes :

- Langage de programmation : Java
- Framework graphique : JavaFX
- Base de données : MySQL

3.3 Fonctionnalités

Les fonctionnalités à offrir à ce système sont les suivants :

Gestion des produits

- Ajouter un produit
- Modifier les informations d'un produit
- Supprimer un produit

| | |
|--|--|
| Fonction : Gestion de stock des produits | |
| Objectif | Manipuler le stock des produits, accéder facilement à l'information lié au produit. |
| Description | À travers l'application on peut enregistrer un nouveau produit en cliquant sur le bouton « ajouter » dans la rubrique « produits ». On peut consulter la liste complètes des produits présentent dans le stock en cliquant sur le bouton « liste des produits », qui comprend les informations suivantes sur le produit : code, nom, catégorie, forme, prix, fournisseur, date fabrication et date expédition. |
| Contraintes/règles de gestion | Avoir les informations sur le fournisseur en avance |
| Niveau de priorité | Haute priorité |

Tableau 1 - Grille de la fonction gestion des produits

Gestion des ventes

- Ajouter une nouvelle vente
- Imprimer facture de client
- Imprimer facture totale des ventes

| | |
|---|---|
| Fonction : Control des factures et enregistrer les ventes | |
| Objectif | Enregistrer les différentes opérations de vente, et accéder facilement à leur historique. |
| Description | À travers l'application on peut enregistrer des opérations de vente et imprimer les factures en cliquant sur le bouton « imprimer » dans rubrique « vente ». On peut aussi afficher l'historique des ventes en cliquant sur bouton « liste des vente », qui comprend les informations suivantes : nom et CNI du client, date de vente et prix total en plus d'imprimer la facture totale des ventes de tous les clients en cliquant sur « imprimer ». |
| Contraintes/règles de gestion | Présence de la quantité demandée du produit dans la base de données/stock. |
| Niveau de priorité | Haute priorité |

Tableau 2 - Grille de la fonction gestion des ventes

Gestion des fournisseurs

- Ajouter un fournisseur
- Modifier les informations d'un fournisseur
- Supprimer un fournisseur

| | |
|--|--|
| Fonction : Enregistrer le contact du fournisseur | |
| Objectif | Accéder facilement aux informations du fournisseur |
| Description | Affichage de la liste des fournisseurs à travers l'application en cliquant sur bouton « fournisseur ». Il comprend les informations suivantes : Identifiant, nom et email du fournisseur et le type de paiement. |
| Contraintes/règles de gestion | Aucune |
| Niveau de priorité | Priorité haute |

Tableau 3 - Grille de la fonction gestion des fournisseurs

3.4 Conclusion

Dans ce chapitre, on a spécifié les besoins auxquels l'application doit satisfaire et les différentes fonctionnalités. En les prenant en compte, on a pu proposer une solution optimale avec l'outil de conception d'interface graphique « JavaFX Scene Builder », car il permet de concevoir rapidement les interfaces utilisateur de l'application JavaFX, sans avoir besoin d'écrire le code. Il suffit de faire glisser et déposer (drag and drop) les composants de l'interface d'utilisateur à une zone de travail et donc réduire le temps de développement.

4 Conception

4.1 Introduction

La conception est une étape cruciale dans le développement des applications. Elle permet de transformer les besoins fonctionnels en une structure logique et cohérente, en utilisant divers diagrammes et modèles. Dans ce chapitre, nous allons tout d'abord présenter l'organisation et la répartition des tâches au sein du groupe durant la durée du projet. Ensuite, nous aborderons les différents diagrammes utilisés dans la conception de notre application.

4.2 Répartition des tâches à réaliser

Une fois que la planification est établie et l'identification des tâches est réalisée, il fallait répartir ces tâches. Nous avons attribué ces dernières en fonction de préférence et des compétences de chacun pour que chaque membre de l'équipe soit compétent dans celles qui lui ont été imputées.

Tâches confiées à Tariq (chargé de programmation, gestion contrôleur et BDD)

Implémentation des différents contrôleurs et établir la connexion à la base de données (à l'aide de WAMP et PHPMyAdmin) :

- La connexion à la base de données pour manipuler les différentes tâches (CRUD) en temps réel et assurer la synchronisation et l'authentification.
- Contrôleur pour l'authentification.
- Contrôleur pour gestion des ventes.
- Contrôleur pour gestion des produits/stock.
- Contrôleur pour gestion des fournisseurs.

Tâches confiées à Nour Eddine (chargé d'interface graphique)

Création des différentes interfaces (à l'aide de SceneBuilder) comme suit :

- Interface d'accueil, authentification.
- Interface pour la gestion des produits/stock.
- Interface pour gestion des ventes.
- Interface pour gestion des fournisseurs.

Tâches confiées à Amine (chargé communication)

Communiquer avec le client et relever ses besoins afin de rédiger un cahier des charges complet.

Tâches communes :

- Rédaction du cahier des charges.
- Elaboration des différents diagrammes.
- Tâches supplémentaires/auxiliaires.

4.3 Diagramme de cas d'utilisation

Après avoir étudié le cahier des charges et analysé les différents besoins, nous avons établi ce diagramme de cas d'utilisation qui décrit les fonctions générales et la portée du système, identifier les différents acteur (gérant, fournisseur, client) et leurs interactions avec le système.

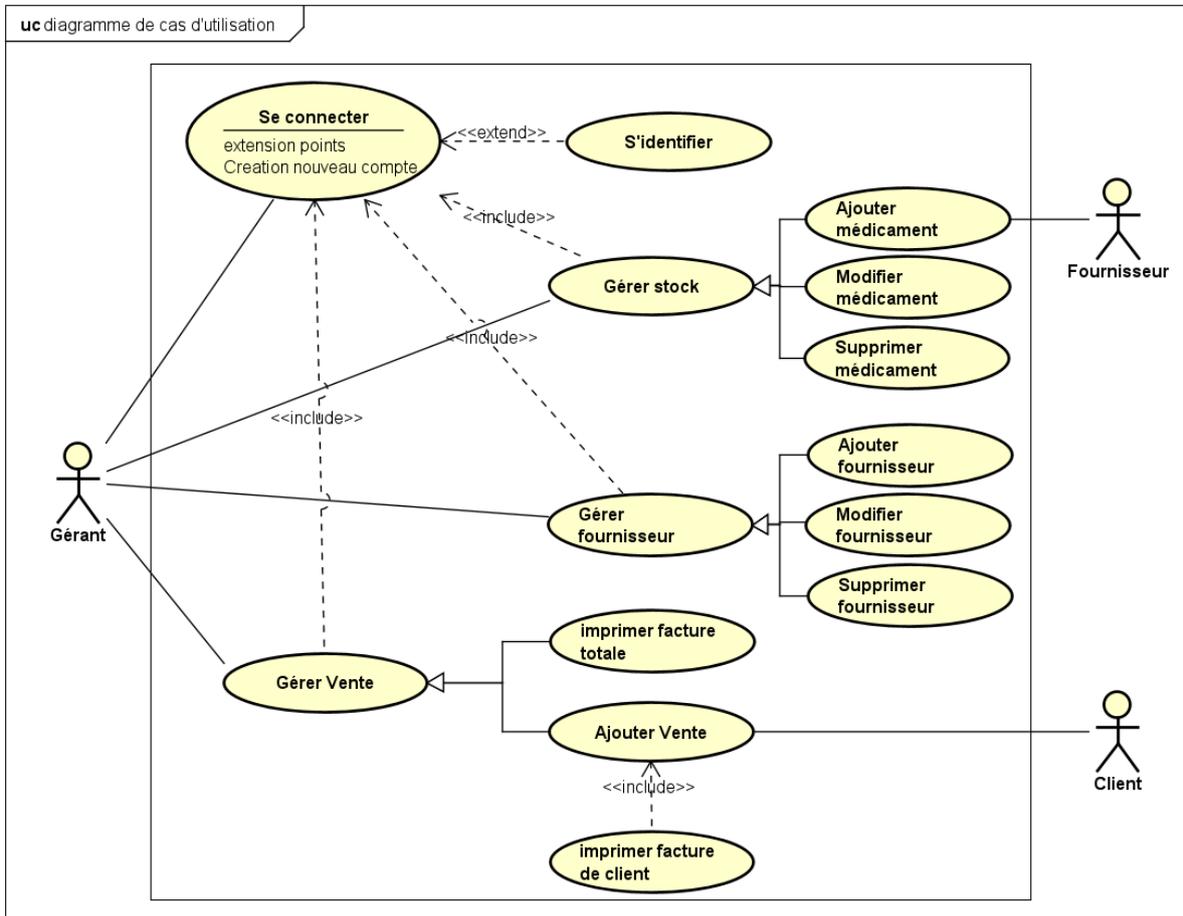


Figure 1 - Diagramme de cas d'utilisation

4.4 Diagramme de classe

Après avoir étudié le cahier des charges et analysé les différents besoins, nous avons établi ce diagramme de classe qui représente les différentes classes, leurs relations et leurs attributs.

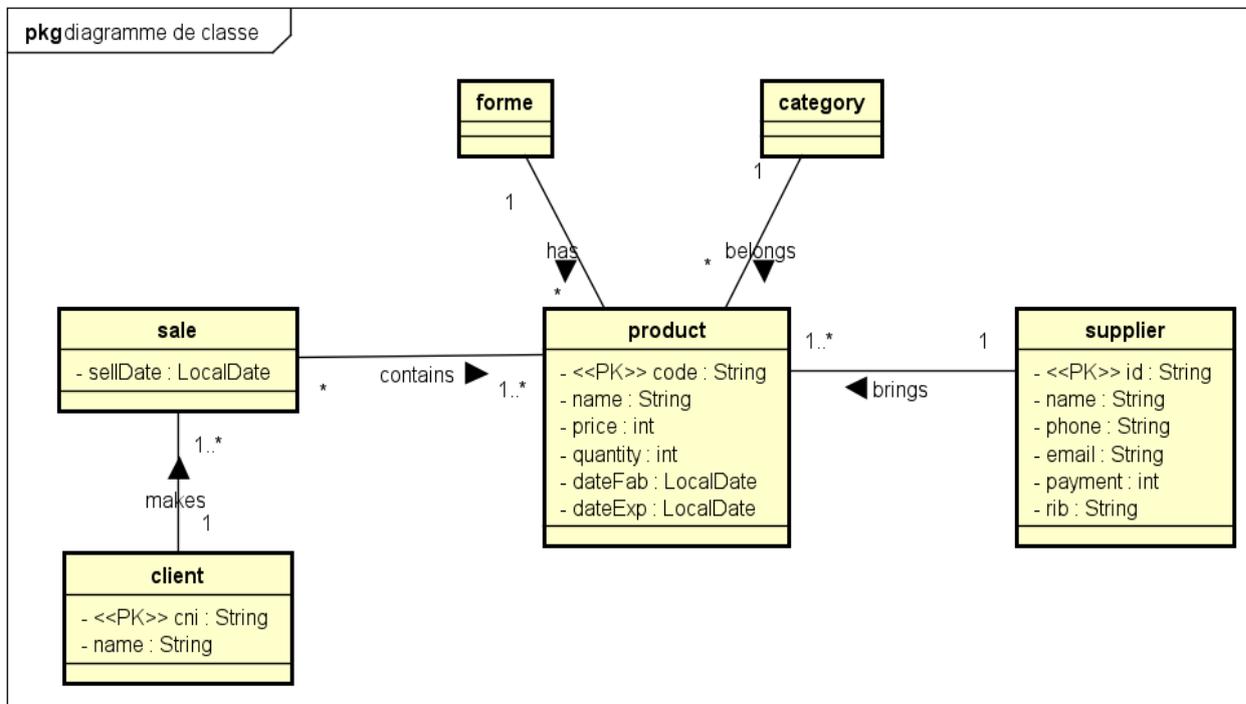


Figure 2 - Diagramme de classe

4.5 Diagramme de séquence

Après avoir étudié le cahier des charges et analysé les différents besoins, nous avons établi les différents diagrammes de séquence qui décrivent comment et dans quel ordre plusieurs objets fonctionnent ensemble. On peut diviser ces diagrammes en quatre catégories : l'authentification, gestion produits, gestions fournisseurs et gestion de ventes.

Diagramme de séquence pour Authentification

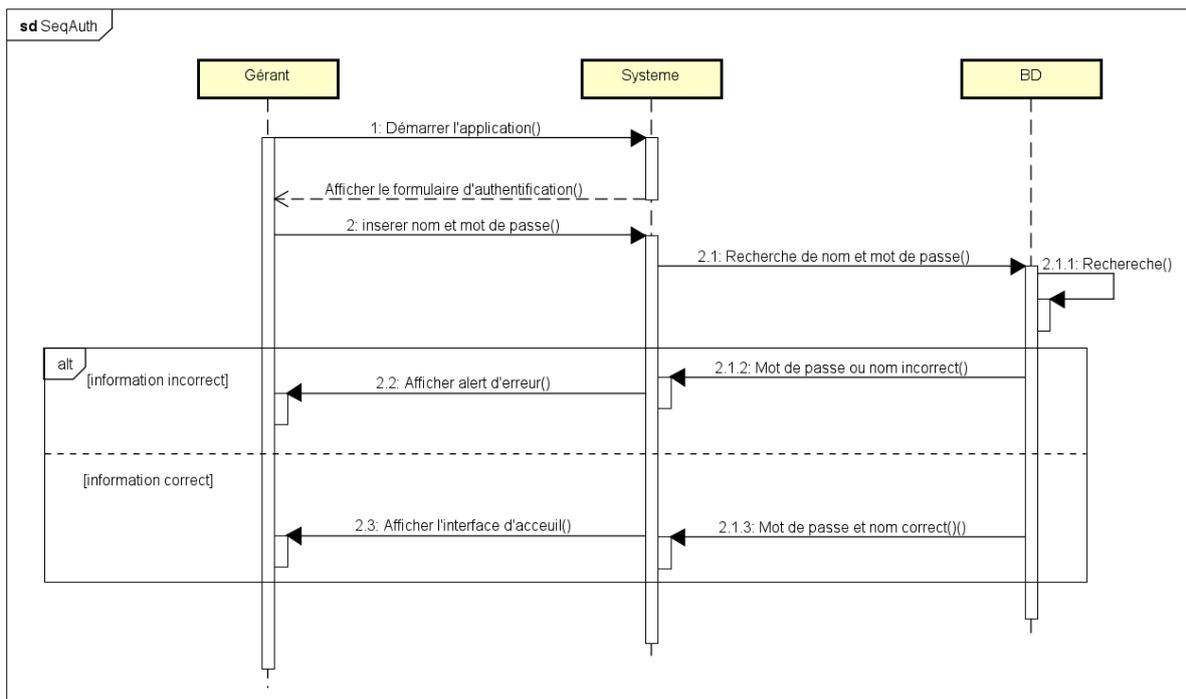


Figure 3 - Diagramme de séquence d'authentification

Diagrammes de séquence pour gestion des fournisseurs

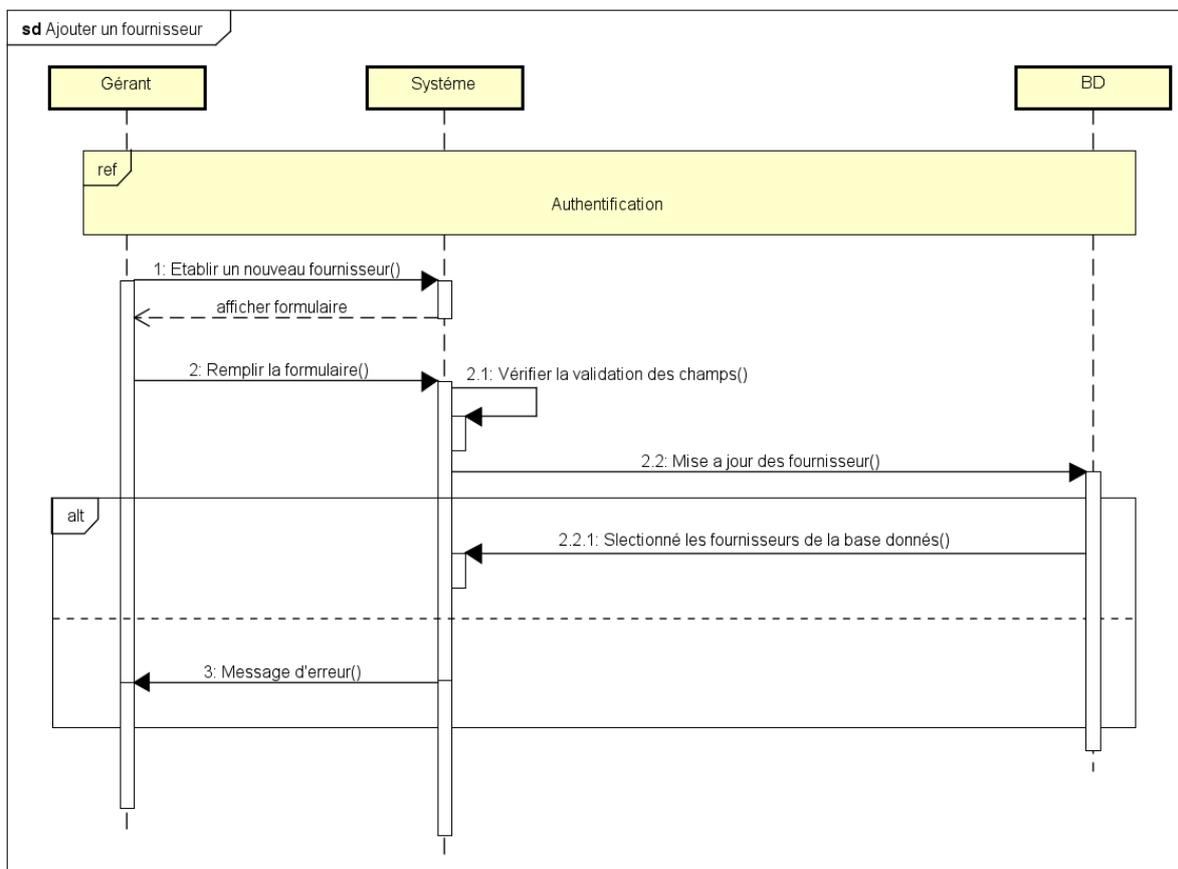


Figure 4 - Diagramme de séquence d'ajout de fournisseur

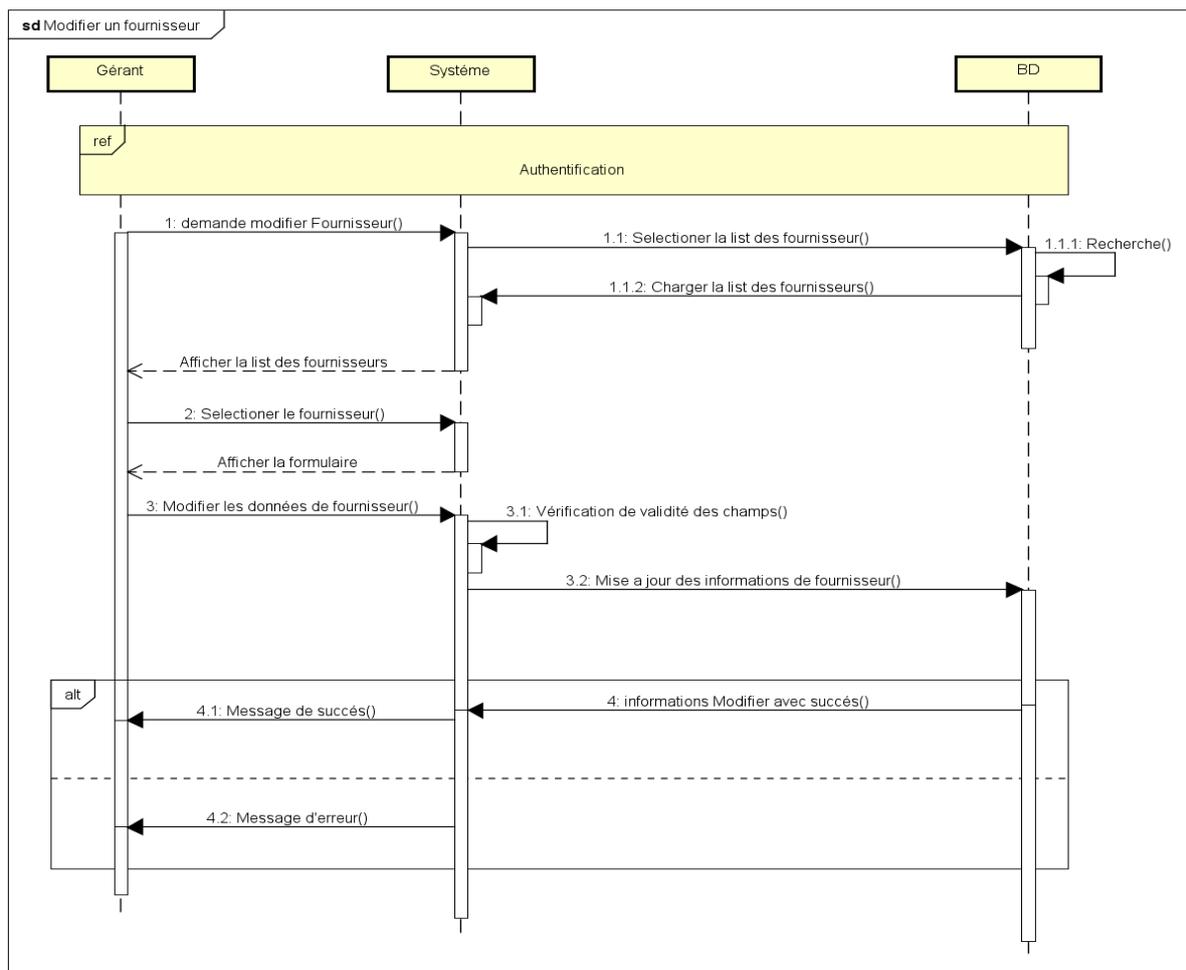


Figure 5 - Diagramme de séquence de modification de fournisseur

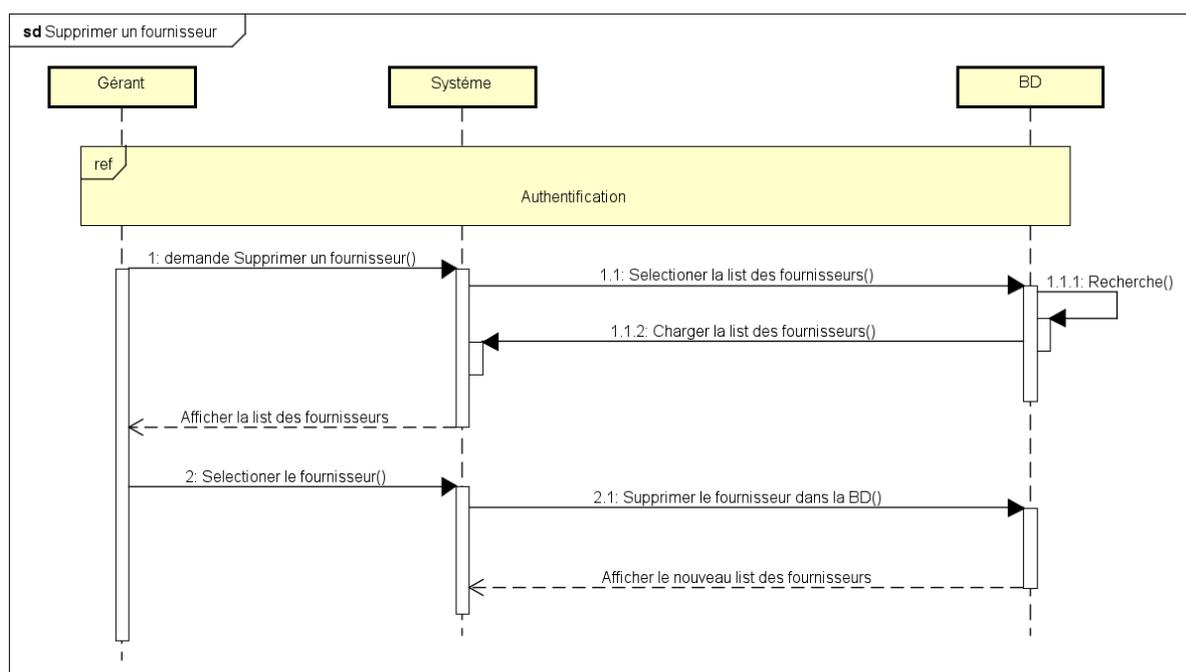


Figure 6 - Diagramme de séquence de suppression de fournisseur

Diagrammes de séquence pour gestion des produits

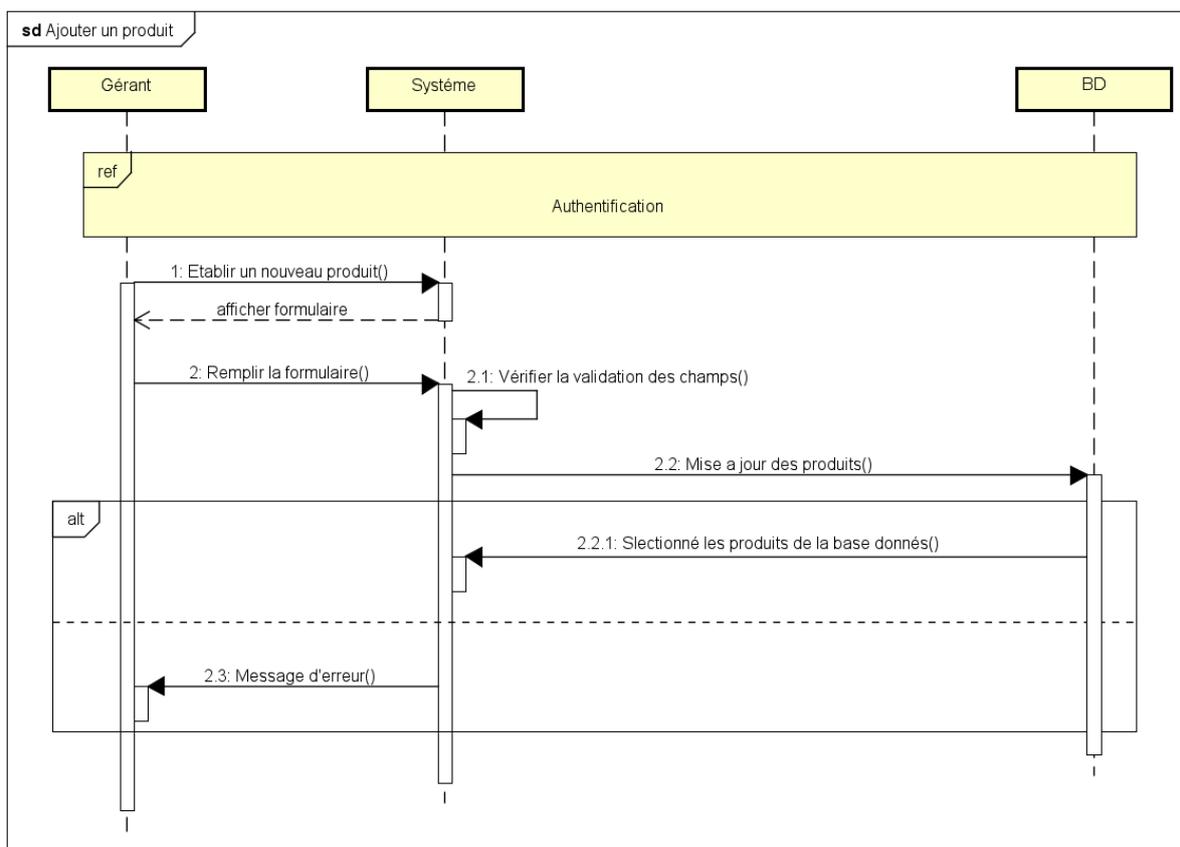


Figure 7 - Diagramme de séquence d'ajout de produit

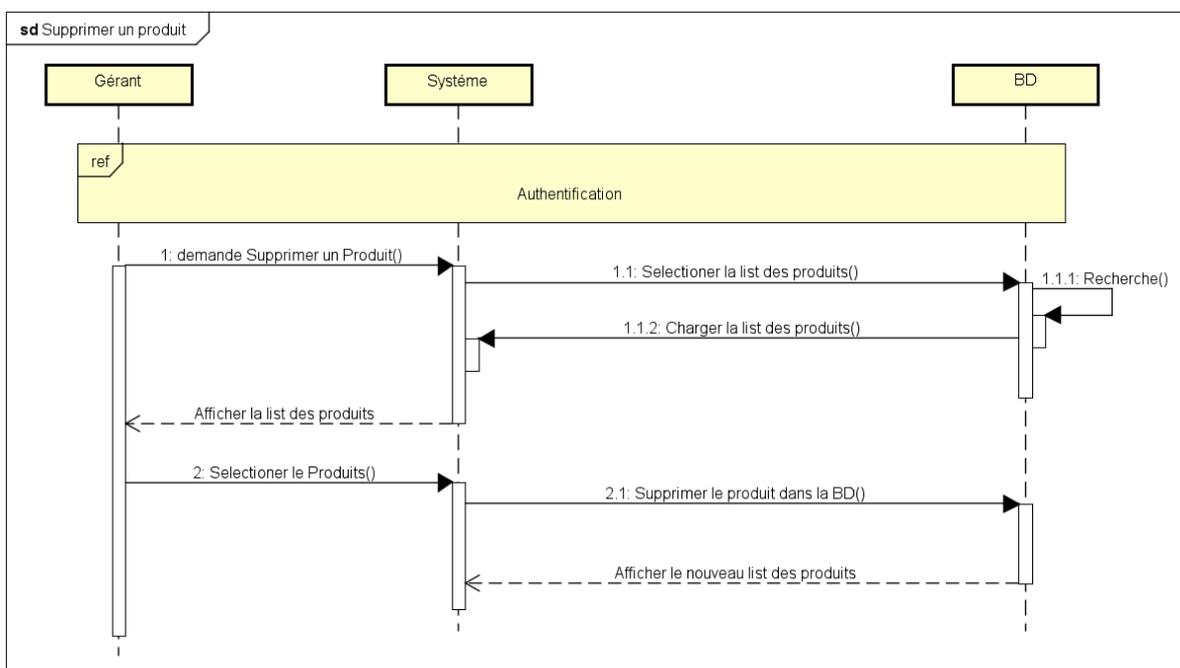


Figure 8 - Diagramme de séquence de suppression de produit

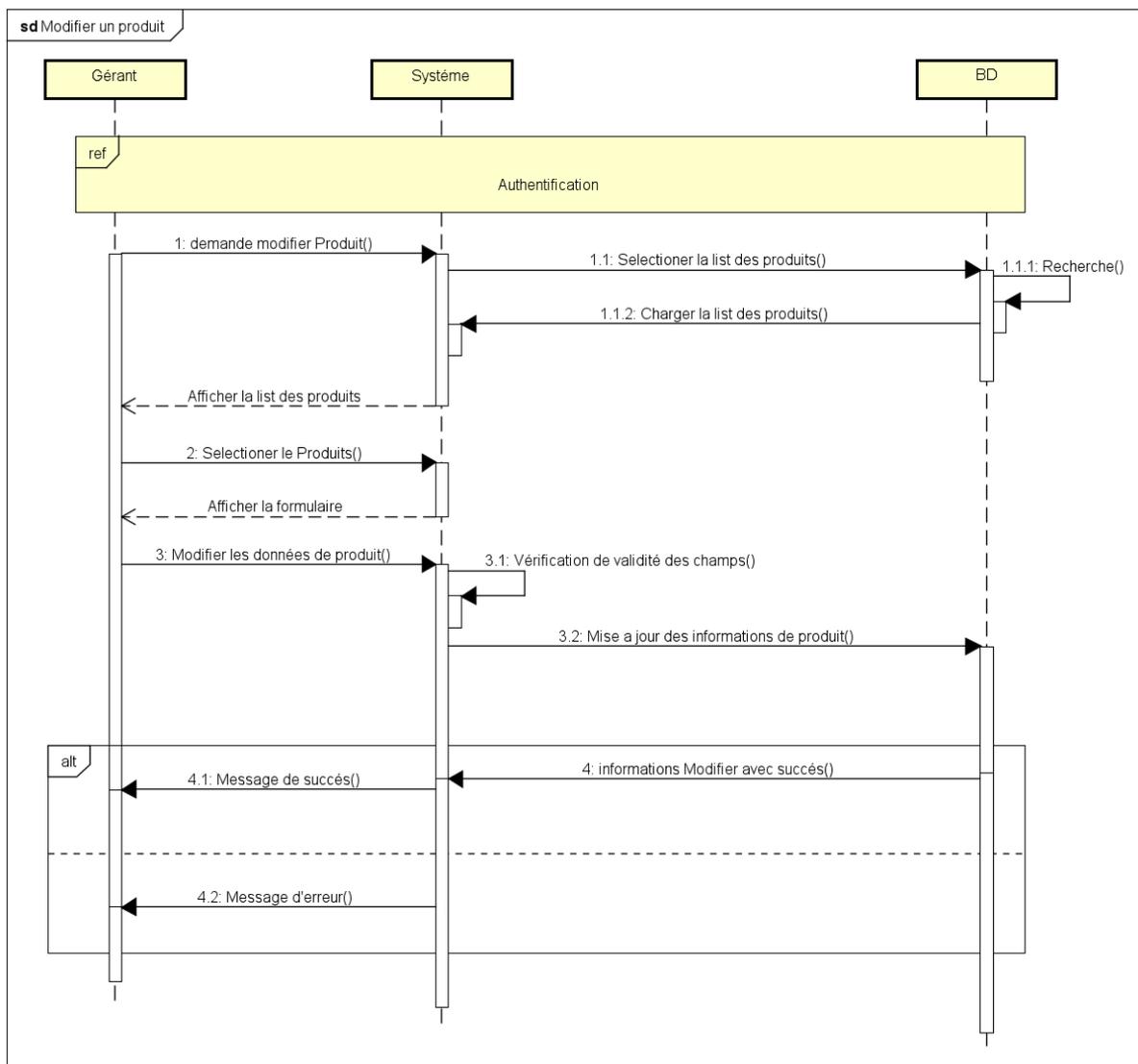


Figure 9 - Diagramme de séquence d'ajout de produit

Diagrammes de séquence pour gestion des ventes

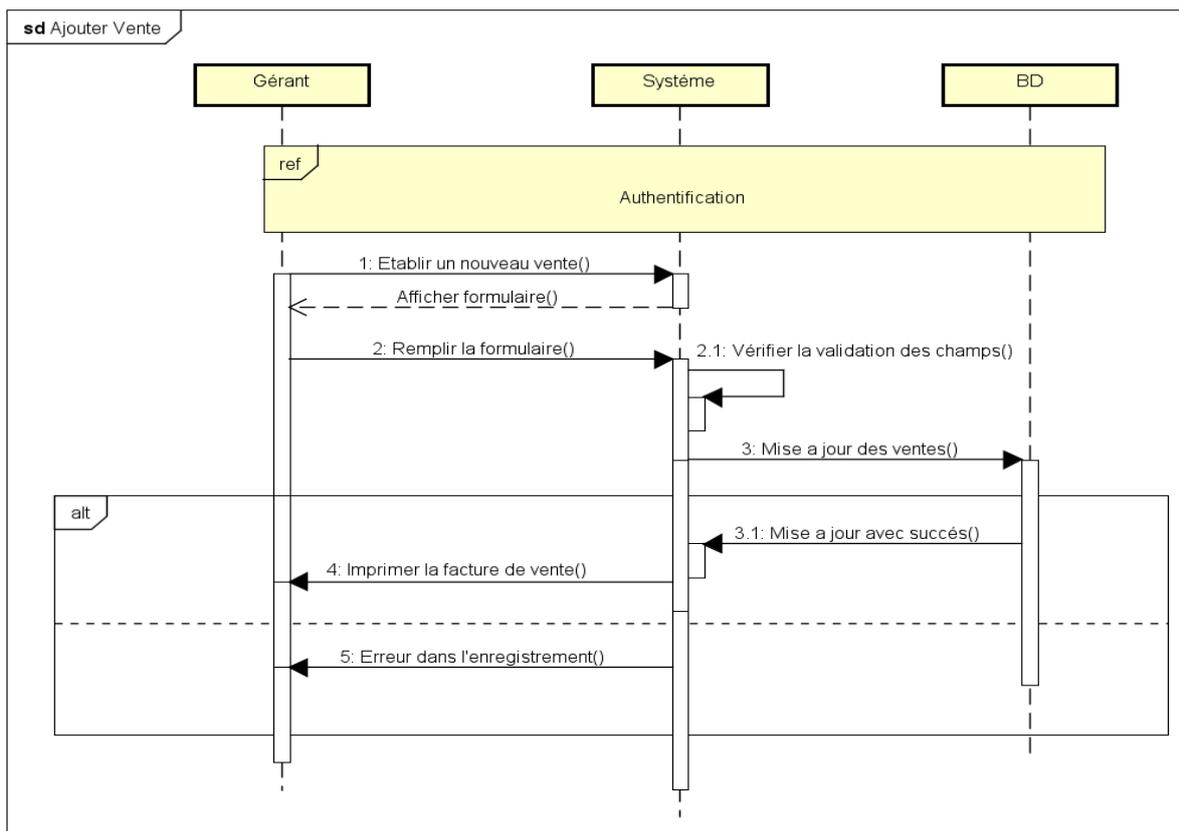


Figure 10 - Diagramme de séquence d'ajout de vente

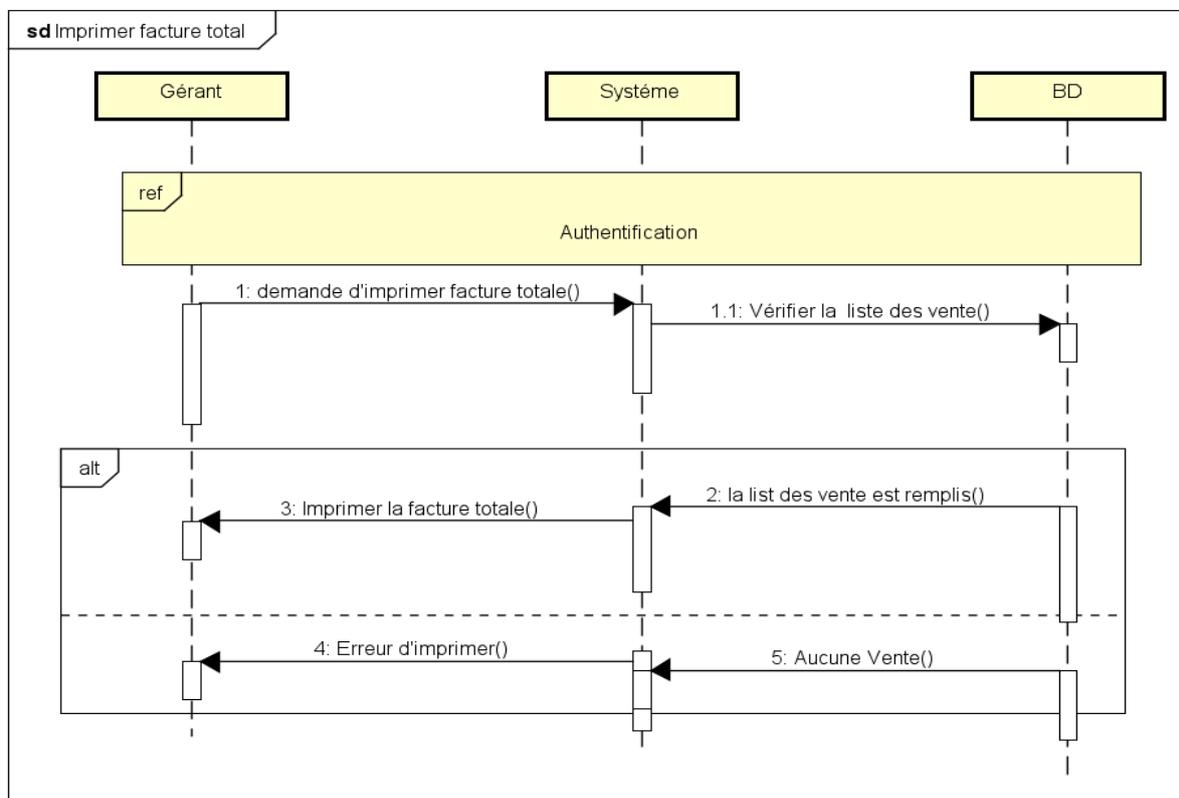


Figure 11 - Diagramme de séquence d'impression de facture totale

5 Implémentation

5.1 Introduction

Après avoir étudié les besoins, les spécifications de l'application et élaboré les différents diagrammes, on a trouvé une technologie adaptée pour aboutir au projet et pour satisfaire au cahier des charges.

Tout d'abord, dans ce chapitre, les outils de développement utilisés seront spécifiés. Puis, on précisera la manière dont le code a été organisé et comment l'interface graphique a été définie. Enfin, les difficultés rencontrées au cours du projet seront citées.

5.2 Outils de développement

Java (JDK v.17.0.6)

D'après le site officiel, Java est un langage de programmation et une plate-forme de calcul lancé par Sun Microsystems en 1995. Depuis ses débuts modestes, Java a beaucoup évolué. À l'heure actuelle, une grande partie du monde numérique dépend de Java : de nombreux services et applications reposent sur cette plate-forme fiable. De même, de nouveaux produits et services numériques innovants et tournés vers l'avenir dépendent aussi de Java.

JavaFX (SDK v.20)

JavaFX est une plateforme logicielle basée sur Java, permettant aux développeurs de créer des interfaces utilisateur graphiques interactives pour les applications de bureau et les applications Internet riches. Elle offre un large éventail de fonctionnalités, de contrôles et de graphiques, et peut être utilisée pour créer des applications multiplateformes avec une expérience utilisateur cohérente. JavaFX a été conçu pour remplacer Swing et est largement utilisé dans le développement d'applications Java modernes.

Css

CSS (Cascading Style Sheets) est un langage de feuilles de style utilisé pour décrire la présentation et le style d'un document HTML (ou d'autres types de documents XML). Il permet de contrôler l'apparence des éléments d'une page web, tels que les couleurs, les polices, les marges, les positions, les arrière-plans, les bordures, etc.

Scene Builder (v.20)

Scene Builder est un outil de conception visuelle d'interfaces utilisateur pour les applications JavaFX. Il permet de créer des interfaces en glissant-déposant des éléments graphiques et en configurant leurs propriétés, sans avoir à écrire de code JavaFX manuellement.

Eclipse (R2022)

Eclipse est un environnement de développement intégré (IDE) polyvalent et largement utilisé pour la programmation dans différents langages. Il offre une interface conviviale et des fonctionnalités avancées pour le développement d'applications

Wamp (MySQL version 5.6.17)

WAMP est un ensemble de logiciels (Windows, Apache, MySQL, PHP) qui permet de créer facilement un environnement de développement web sur Windows. Il inclut Apache comme serveur web, MySQL pour la base de données, et PHP pour la programmation côté serveur.

Github

GitHub est un service cloud pour les développeurs, permettant de stocker et gérer leur code. Il offre un historique des modifications, utile pour résoudre les bugs et erreurs introduits lors des modifications. L'utilisation de Git permet de contrôler les versions et d'identifier la source des problèmes.

Lien vers dépôt git : https://github.com/ELEF-TQ/Pharmacy_Management

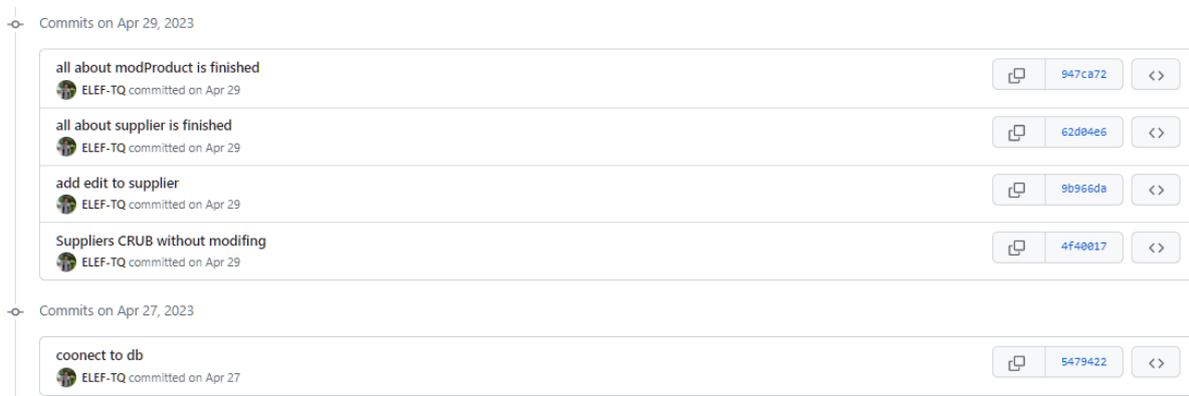


Figure 12 - Quelques commits sur Github

5.3 Organisation du code

Le projet a été découpé en plusieurs packages qu'on peut regrouper en deux parties majeures. La première est la mise en place de l'interface graphique. La deuxième partie est la gestion des interactions avec l'interface graphique ou l'application.

Les fichiers de code source sont organisés de la manière présentée dans la figure suivante « Figure 13 - Structure du code source. ».

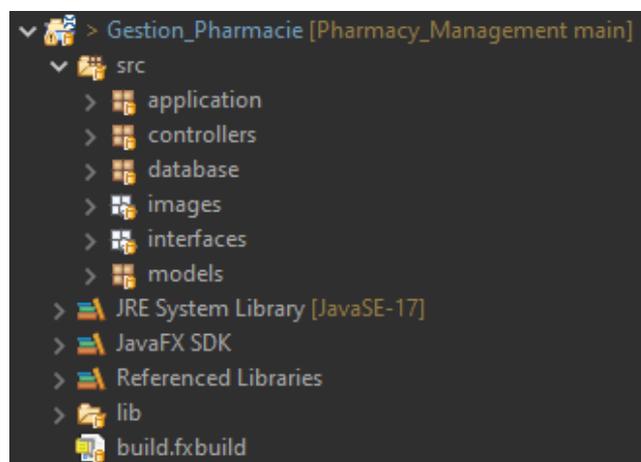


Figure 13 - Structure du code source.

- Le package « application » : contient la classe main exécutable et le fichier CSS des interfaces.
- Le package « controllers » : contient les contrôleurs des interfaces graphiques.
- Le package « database » : contient la classe qui assure la connexion à la base de données.
- Le package « images » : contient les images utilisées dans les interfaces.
- Le package « interfaces » : contient les fichiers FXML des interfaces graphiques.
- Le package « models » : contient les classes utilisées (produit, fournisseur, client).

Il existe certaines fonctionnalités qui ne sont pas disponibles dans la librairie standard de Java. Il est alors plus pratique d'utiliser des JARs externes pour répondre à ces besoins. Le tableau « Tableau 4 - Liste des JARs utilisés. » représente les principaux JARs importés dans l'application ainsi que leurs utilités.

| JARs | Utilité |
|---|--|
| <i>mysql-connector-java-5.1.49</i> | Etablir la connexion à la base de données |
| <i>fontawesomefx-8.2</i> <i>jfoenix-8.0.10</i> | Fournir les icônes et contrôleurs des interfaces |
| <i>fontbox-2.0.28</i> <i>pdfbox-2.0.28</i> <i>commons-logging-1.2</i> | Impression d'un fichier PDF |

Tableau 4 - Liste des JARs utilisés

5.4 Structure de la base de données

Afin de gérer, stocker et manipuler les données, on propose une architecture avec 5 tables qui répondent à tous les besoins de ce projet. On y trouve les différentes tables pour gérer les produits, les ventes, fournisseurs et les clients. La structure de la base de données suivante réalisée avec phpMyAdmin.

| Table | Action | Rows | Type | Collation | Size | Overhead |
|---|--|------|--------|-------------------|--------|----------|
| <input type="checkbox"/> products | Browse Structure Search Insert Empty Drop | ~20 | InnoDB | latin1_swedish_ci | 16 KiB | - |
| <input type="checkbox"/> sales | Browse Structure Search Insert Empty Drop | ~0 | InnoDB | latin1_swedish_ci | 16 KiB | - |
| <input type="checkbox"/> sales_products | Browse Structure Search Insert Empty Drop | ~0 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| <input type="checkbox"/> suppliers | Browse Structure Search Insert Empty Drop | ~20 | InnoDB | latin1_swedish_ci | 16 KiB | - |
| <input type="checkbox"/> users | Browse Structure Search Insert Empty Drop | ~2 | InnoDB | latin1_swedish_ci | 16 KiB | - |
| 5 tables | Sum | 42 | InnoDB | latin1_swedish_ci | 96 KiB | 0 B |

Figure 14 - Les différentes tables de la base de données

5.5 Interface utilisateur

Après l'installation de l'application sur l'ordinateur, celle-ci se présente comme indiqué dans la figure « Figure 6 - Icône de l'application ».



Figure 15 - Icône de l'application

Interface d'inscription

Cette interface correspond à la première interface affichée lors de l'ouverture de l'application. Elle sert à créer un nouveau compte pour le gérant.

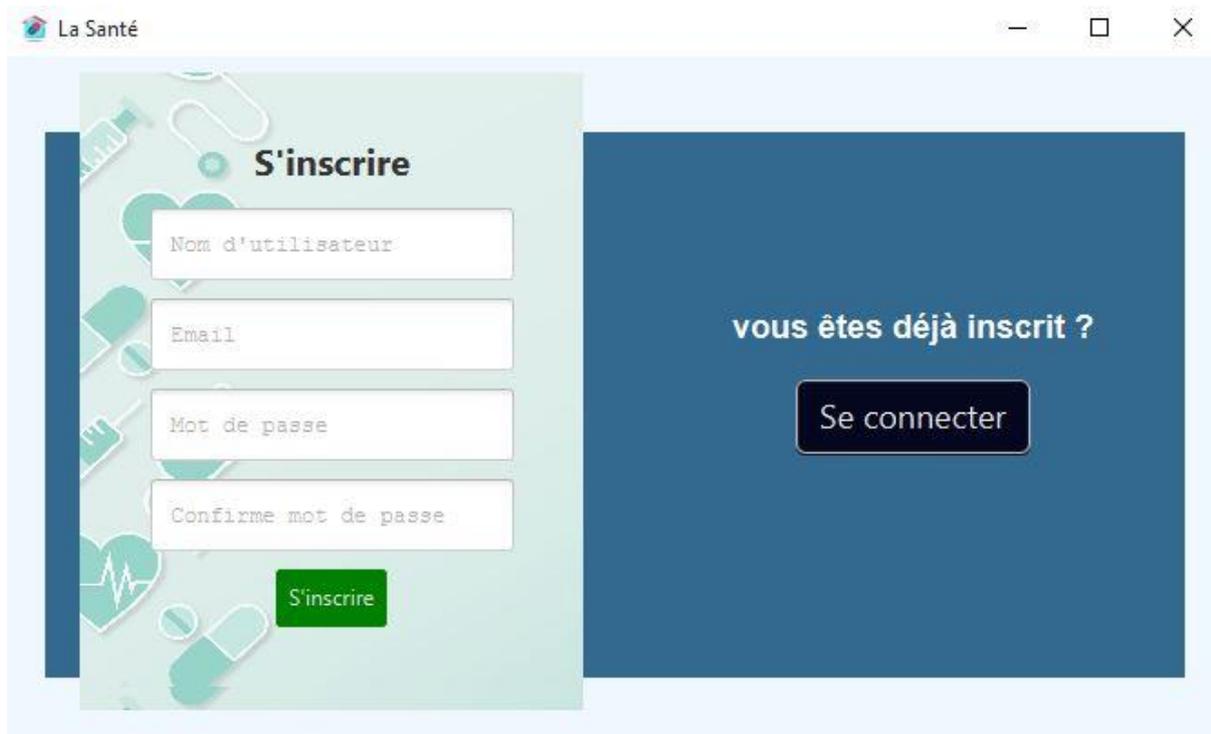


Figure 16 - Interface d'inscription

Interface d'authentification

Cette interface sert à s'identifier pour accéder à l'application et se connecter avec la base de données.

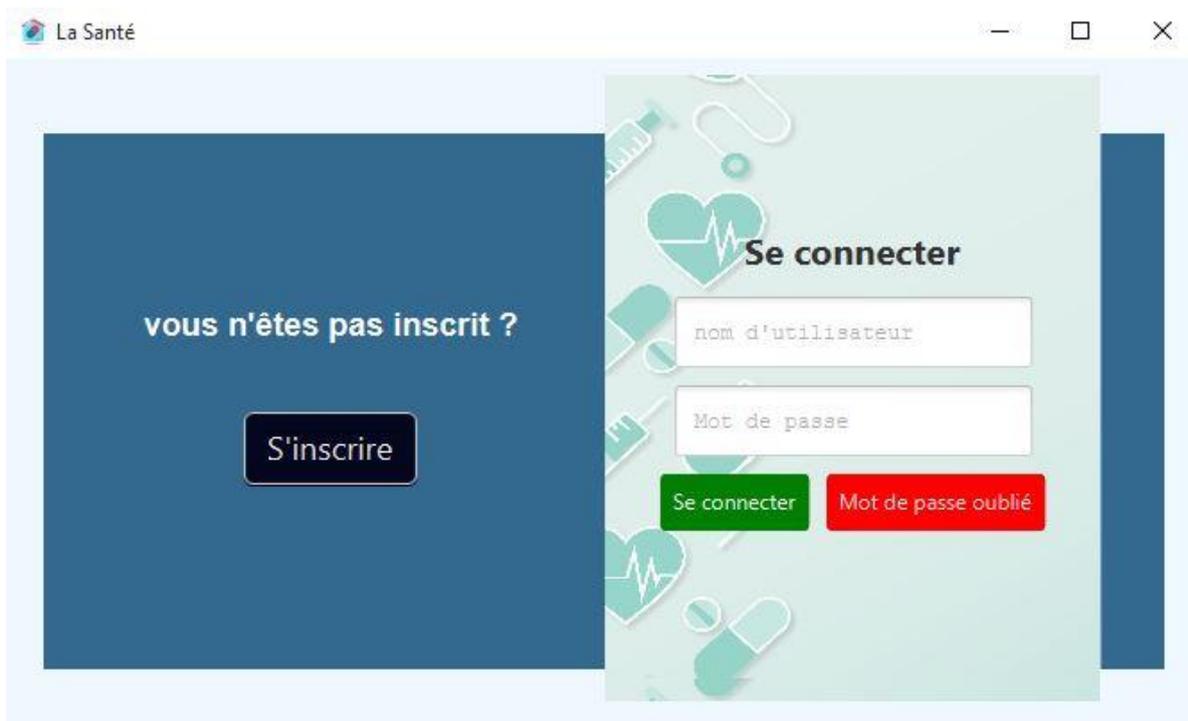


Figure 17 - Interface d'authentification

Interface d'accueil

Cette interface représente un bilan. On y trouve le total des médicaments, total des fournisseurs et total des ventes. Dans la barre à gauche, on peut accéder aux différentes interfaces qui permettent de gérer les produits, les fournisseurs et les ventes, avec les différentes fonctionnalités indiquées dans le cahier des charges.



Figure 18 - Interface d'accueil

5.6 Difficultés rencontrées et améliorations

Lors du développement, plusieurs difficultés ont été rencontrées. D'abord, le premier obstacle était de trouver un client qui accepte de nous fournir un cahier des charges complet. Cela est dû à la disponibilité des applications similaires à notre type sur le marché, ce qui pose un défi majeur. De plus, nous avons rencontré des difficultés lors de l'utilisation de Git à savoir des problèmes d'importation des bibliothèques nécessaires et autre. Nous avons également dû effectuer des recherches pour trouver des versions compatibles entre JavaFX, JDK, Scene Builder et Eclipse.

Pour améliorer l'application, nous envisageons les actions suivantes :

D'une part, concernant la sécurité l'application, nous prévoyons de créer un compte spécial pour l'administrateur, qui aura des privilèges pour gérer les comptes utilisateurs (création, suppression, etc.). Cette approche permettra de restreindre l'accès et de garantir une utilisation appropriée de l'application, contrairement à la version actuelle où tous les utilisateurs peuvent créer un compte sans restriction. **D'autre part**, quant à l'annulation d'une vente, nous constatons la nécessité de pouvoir annuler une vente dans le cas où un client retourne un produit. Pour cela, nous devons ajouter une nouvelle propriété aux ventes pour indiquer leur état (annulée, validée, etc.). De plus, il sera nécessaire de remettre le produit retourné dans le stock. Cette fonctionnalité est actuellement absente de notre application.

En prenant en compte ces améliorations, nous pourrions renforcer la sécurité de l'application et offrir une meilleure gestion des ventes, en permettant l'annulation des ventes et la mise à jour du stock en conséquence. Cela améliorera l'expérience utilisateur et la fiabilité globale de l'application de gestion de pharmacie.

6 Installation

6.1 Introduction

Dans ce chapitre nous allons indiquer les éléments essentiels et les spécifications de l'installation afin que l'utilisateur puisse utiliser l'application dans sa propre machine et l'outil de création du fichier JAR.

6.2 Outil utiliser pour création du JAR et spécifications

Launch4j

Launch4j est un outil multiplateforme permettant d'encapsuler des applications Java distribuées sous forme de fichiers jar dans des exécutables natifs Windows légers. L'exécutable peut être configuré pour rechercher une certaine version de JRE ou utiliser une version groupée, et il est possible de définir des options d'exécution, comme la taille de tas initiale/max. Le wrapper offre également une meilleure expérience utilisateur grâce à une icône d'application.

Spécification de de JAR et JRE et l'icône utilisée :

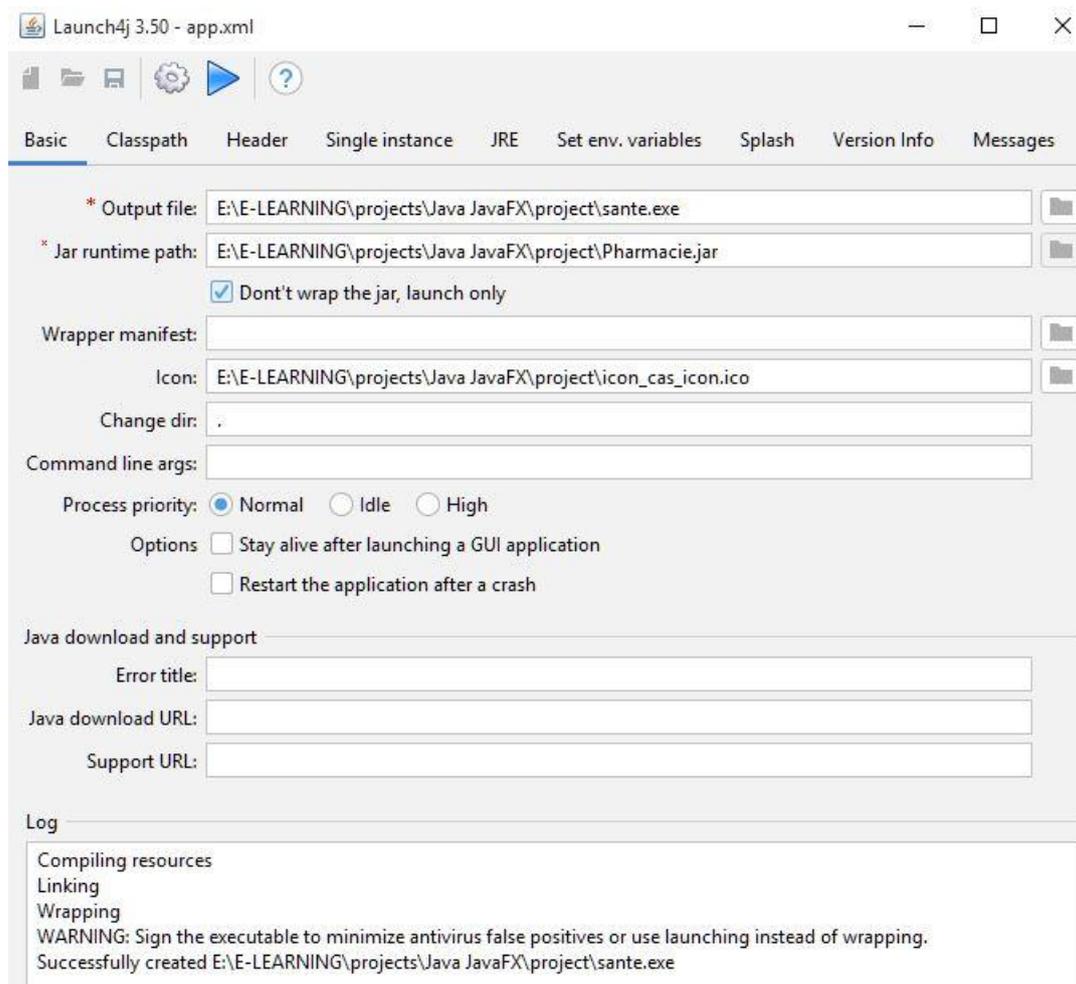


Figure 19 - Spécification sur JAR et JRE et l'icône utilisée

Spécification de JDK et les arguments de la machine virtuelle

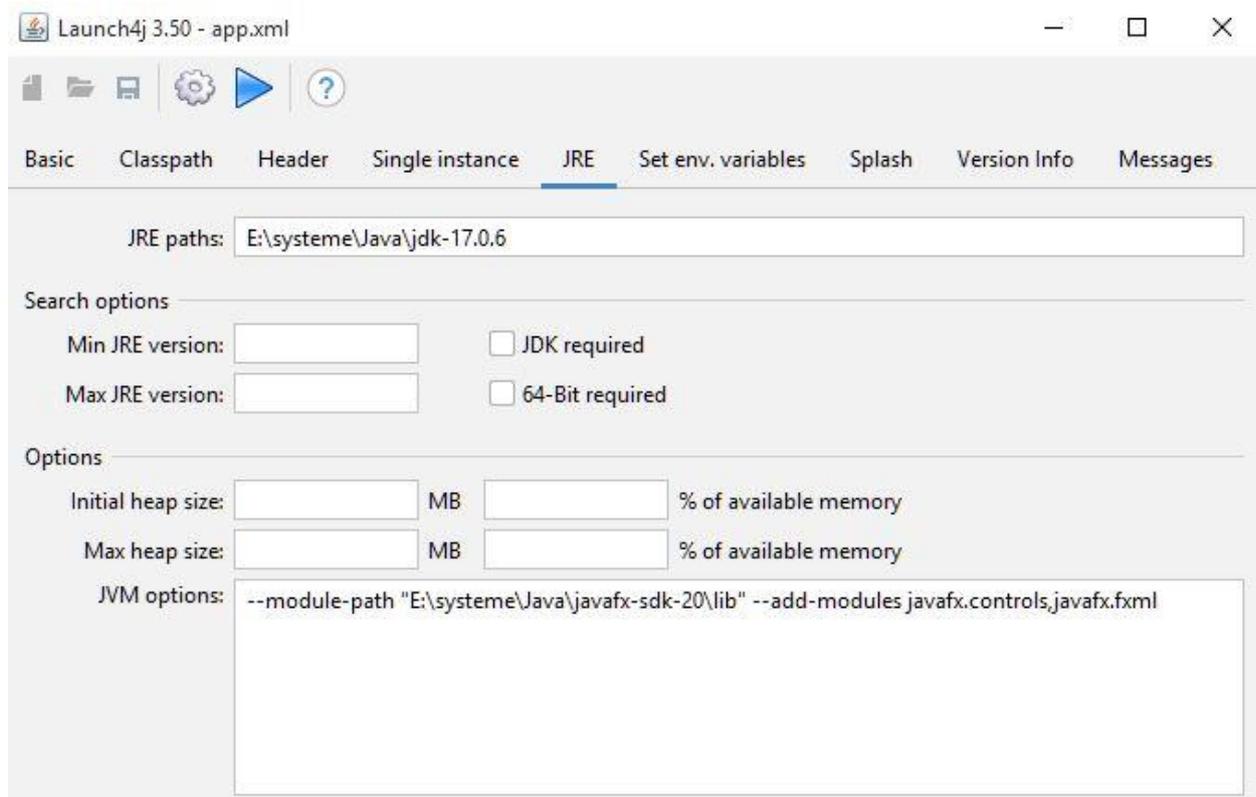


Figure 20 - Spécification sur JDK et les arguments de la machine virtuelle

7 Conclusion

En guise de conclusion, le projet a pour objectif la conception d'une application desktop visant à faciliter la gestion d'une pharmacie. L'application réalisée répond de manière satisfaisante au cahier des charges initial, offrant ainsi une solution fonctionnelle et adaptée aux besoins de gestion de la pharmacie. Néanmoins, il est important de souligner que des améliorations peuvent encore être apportées pour optimiser davantage l'application.

En ce qui concerne les améliorations potentielles, il serait envisageable d'ajouter des fonctionnalités telles l'annulation de vente, et améliorer la sécurité en ajoutant un compte unique et spécial d'administrateur permettant de gérer tous les comptes dans notre application. Ces améliorations permettraient d'accroître l'efficacité et la convivialité de l'application, offrant ainsi une expérience utilisateur encore plus agréable en toute sécurité.

À l'issue de ce projet, nous avons pu acquérir de nouvelles connaissances et compétences précieuses. Malgré les difficultés rencontrées lors du développement, celles-ci nous ont permis de découvrir de nouvelles notions et de relever des défis techniques. Nous avons ainsi renforcé notre compréhension en programmation et en développement logiciel de manière générale.

En conclusion, ce projet de développement d'une application de gestion de pharmacie a été une expérience formatrice et enrichissante. Nous avons pu mettre en pratique nos connaissances théoriques, tout en faisant face à des problématiques concrètes. Cela nous a permis d'acquérir de nouvelles compétences et de consolider notre expertise dans le domaine du développement logiciel. Nous sommes fiers du résultat obtenu et sommes convaincus que cette expérience nous servira dans nos futurs projets professionnels.