



IIR

INGÉNIERIE
INFORMATIQUE
& RÉSEAUX

ANNÉE UNIVERSITAIRE

2024-2025

OPTION

MÉTHODES INFORMATIQUES APPLIQUÉES
À LA GESTION DES ENTREPRISES (MIAGE)

MINI PROJET

THÈME

*Conception et développement d'un Système de
gestion des pharmacies avec alerte de stock
et expiration des médicaments*

RÉALISÉ PAR

- MECHACH Marouane
- IDRISSE Mohamed
- BAANNI Zaineb

ENCADRÉ PAR

- ESSABBAR Driss
- BOUSQAOU Halima
- NADIRI Abdeljalil
- CHAREF Ayoub

RENDU LE

12 mai 2025

Dédicace

Nous souhaitons dédier ce travail à nos familles, qui nous ont encouragés et soutenus tout au long de notre parcours académique. Leur sagesse et leur amour inconditionnel nous ont aidés à surmonter les obstacles et à poursuivre nos rêves.

À nos professeurs et encadrants, pour leur expertise, leur patience et leur précieuse guidance tout au long de ce projet. Leurs conseils avisés et leur passion pour l'enseignement ont été pour nous une source d'inspiration constante.

À nos amis et camarades de promotion, qui ont partagé avec nous cette expérience et qui nous ont offert un environnement positif et stimulant pour la réalisation de ce travail. Leurs encouragements, leur collaboration et leur amitié ont été un véritable soutien.

Enfin, cette dédicace est le témoignage de notre profonde gratitude envers toutes les personnes qui ont compté pour nous durant notre parcours académique. Votre soutien indéfectible a été la clé de notre réussite.

Remerciement

En préambule à ce projet, nous souhaitons exprimer notre profonde reconnaissance à toutes les personnes qui, de près ou de loin, ont contribué à l'aboutissement de ce travail ainsi qu'à la réussite de notre formation.

Nous tenons à remercier tout particulièrement Monsieur **ESSABBAR Driss** pour les bases solides en conception logicielle qu'il nous a transmises, et qui nous ont été d'une grande utilité tout au long de ce projet.

Nos remerciements vont également à Monsieur **NADIRI Abdeljalil**, dont les enseignements nous ont permis de mieux comprendre les pratiques modernes de gestion des cycles de développement et de déploiement.

Nous exprimons notre gratitude à Madame **BOUSQAoui Halima** pour l'approche rigoureuse et structurée qu'elle nous a inculquée dans le développement d'applications d'entreprise.

Enfin, nous remercions Monsieur **CHAREF Ayoub** pour nous avoir initiés aux méthodes de développement mobile, un domaine essentiel dans l'univers technologique actuel.

Que toutes les personnes ayant contribué, de près ou de loin, à la réussite de ce projet trouvent ici l'expression de notre sincère reconnaissance.

Résumé

Ce rapport présente le projet « Système de gestion des pharmacies avec alerte de stock et expiration des médicaments », visant à moderniser la gestion officinale en remplaçant les méthodes manuelles encore largement utilisées. Ces méthodes montrent des limites en matière de suivi en temps réel, de contrôle des dates de péremption et de prévention des ruptures de stock. La solution développée comprend une interface de gestion intégrant des alertes automatiques, un module de suivi des ordonnances, une planification des rendez-vous, ainsi qu'un tableau de bord décisionnel. Elle s'appuie également sur un modèle d'intelligence artificielle permettant de prédire les besoins en stock, afin d'anticiper les commandes et d'optimiser les approvisionnements. Les résultats attendus incluent une gestion plus fiable, la réduction des pertes de médicaments, un gain de temps pour le personnel, et une amélioration globale du service. Ce système constitue une avancée vers une organisation plus intelligente, autonome et durable des pharmacies.

Afin de mener à bien ce projet, nous avons adopté la méthode agile SCRUM, permettant un développement itératif et collaboratif. Une étude des besoins fonctionnels et conception des diagrammes UML ont d'abord été réalisées pour bien cerner les attentes du système. Nous avons ensuite utilisé Next.js pour le développement du front-end et Spring Boot pour le backend. Enfin, des tests fonctionnels ont été effectués pour valider la performance et la fiabilité du système.

Mots clés : Intelligence artificielle, Prédiction de stock, SCRUM, UML

Abstract

This report presents the project "Pharmacy Management System with Stock and Expiry Alerts," aimed at modernizing pharmacy operations by replacing the still widely used manual methods. These traditional methods show limitations in real-time tracking, expiration date control, and stockout prevention. The developed solution includes a management interface with automated alerts, a prescription tracking module, appointment scheduling, and a decision-making dashboard. It also leverages an artificial intelligence model to predict stock needs, allowing for order anticipation and optimized supply management. The expected outcomes include more reliable management, reduced medication waste, time savings for staff, and an overall improvement in service. This system represents a step toward smarter, more autonomous, and sustainable pharmacy operations.

In order to carry out this project, we adopted the agile SCRUM methodology, enabling iterative and collaborative development. A study of functional requirements and the design of UML diagrams were first conducted to clearly define the system's expectations. We then used Next.js for front-end development and Spring Boot for the back-end. Finally, functional testing was carried out to validate the system's performance and reliability.

Keywords: Artificial Intelligence, Stock Prediction, SCRUM, UML

Liste des figures

Figure 1 Processus SCRUM	5
Figure 2 Diagramme de Gantt.....	7
Figure 3Diagramme de cas d'utilisation général	10
Figure 4 DS : Prédire les besoins en stock avec IA	12
Figure 5 DS : Créer des commandes de réapprovisionnement	14
Figure 6 DS : Gérer les dates de péremption	15
Figure 7 Diagramme de classe	16
Figure 8 logo de spring	19
Figure 9 logo de next.js.....	19
Figure 10 logo de tailwind	20
Figure 11logo de HTML5	20
Figure 12 logo de docker	20
Figure 13 logo de flask	21
Figure 14 logo du SGBD MySQL	21
Figure 15 logo de VS Code.....	22
Figure 16 logo d'Eclipse.....	22
Figure 17 logo d'Android studio	22
Figure 18 logo de GanttProject	23
Figure 19 logo d'Azure Devops	23
Figure 20 Structure du projet	25
Figure 21 Interface de login	26
Figure 22 Interface du register	26
Figure 23 Interface du tableau de bord	27
Figure 24 Interface de gestion des produits	27
Figure 25 Interface de gestion des ventes	28
Figure 26 Interface de gestion des clients.....	28
Figure 27 Écran de connexion mobile /Figure 28 Écran d'accueil mobile.....	29
Figure 29 Écran mobile de gestion des clients /Figure 30 Écran mobile de gestion de l'inventaire ...	30

Glossaire

API: Application Programming Interface

CSS: Cascading Style Sheets

EMSI: École Marocaine des Sciences de l'Ingénieur

HTML: HyperText Markup Language

IA: Intelligence Artificielle

IBM: International Business Machines

IDE: Integrated Development Environment

IDEA: Integrated Development Environment for Advanced programming

J2EE: Java 2 Platform, Enterprise Edition

JS: JavaScript

JSON: JavaScript Object Notation

SDK: Software Development Kit

SGBDR: Système de Gestion de Base de Données Relationnelle

SQL: Structured Query Language

SSG: Static Site Generator

SSR: Server-Side Rendering

SWT: Standard Widget Toolkit

UML: Unified Modeling Language

Table des matières

Dédicace	II
Remerciement.....	III
Résumé	IV
Abstract.....	V
Liste des figures.....	VI
Glossaire.....	VII
Introduction générale	1
CHAPITRE 1 :CONTEXTE GENERALE DU PROJET.....	2
1.Introduction	3
2. Présentation du projet	3
2.1 Problématique.....	3
2.2 Objectifs du projet.....	4
2.3 Solution proposée.....	4
2.4 Démarche et planification	4
2.4.1 La méthode SCRUM	4
2.4.2 Pourquoi SCRUM ?.....	5
2.4.3 L'équipe et rôles.....	6
2.5 Identification des sprints	6
CHAPITRE 2 :ANALYSE ET CONCEPTION	8
1. Introduction	9
2. Langage de modélisation.....	9
3. Diagramme de cas d'utilisation général.....	10
4. Diagrammes de séquence	12
4.1DS pour le cas d'utilisation : Prédire les besoins en stock avec IA	12
4.2DS pour le cas d'utilisation : Créer des commandes de réapprovisionnement	14

4.3DS pour le cas d'utilisation : Gérer les dates de péremption	15
5. Diagramme de classes.....	16
6. Conclusion	17
CHAPITRE 3 :TECHNOLOGIES ET OUTILS UTILISES	18
1. Introduction.....	19
2. Technologies utilisées.....	19
3.Outils de base de données : MySQL.....	21
4. Environnement de travail.....	22
5. Outils de gestion de projet.....	23
6. Conclusion	23
CHAPITRE 4 :IMPLEMENTATION ET MISE EN OEUVRE.....	24
1. Introduction.....	25
2. Architecture logiciel.....	25
3.1 Les interfaces web	26
3.2 Les interfaces mobiles	29
Conclusion générale	31
Références	32

Introduction générale

Dans un contexte où la gestion des pharmacies devient de plus en plus complexe, l'adoption de technologies logicielles performantes s'avère essentielle pour améliorer l'efficacité des opérations, réduire les coûts, gagner du temps et éviter les erreurs humaines. En effet, la gestion manuelle des stocks et des dates de péremption des médicaments, bien que courante dans de nombreuses pharmacies, présente des limites qui entravent la performance opérationnelle et la qualité du service.

Pour répondre à ces défis, notre projet vise à concevoir et développer une application web et mobile de gestion des pharmacies avec alerte de stock et expiration des médicaments, qui permettra d'automatiser et d'optimiser la gestion des stocks, d'assurer un suivi rigoureux des dates de péremption, et de faciliter la gestion des ordonnances et des rendez-vous. Ce système s'appuie sur des technologies modernes telles que **Next.js** pour le frontend et **Spring Boot** pour le backend, avec l'intégration d'un modèle d'intelligence artificielle pour la prédiction des besoins en stock.

Ce rapport représente en grande partie mon travail architecturé suivant quatre parties faisant chacune l'objet d'un chapitre :

- Le premier chapitre présente le contexte du projet, ainsi que la problématique à laquelle il répond et la solution proposée.
- Le deuxième chapitre aborde l'analyse fonctionnelle et la conception du système, en utilisant des diagrammes UML pour modéliser les différents modules.
- Le troisième chapitre détaille les technologies utilisées et les outils techniques mis en œuvre.
- Enfin, le quatrième chapitre présente la réalisation du système, sa mise en œuvre, suivis d'une conclusion générale qui proposera des améliorations potentielles pour l'avenir.

CHAPITRE 1 :

CONTEXTE GENERALE DU PROJET

1.Introduction

Ce chapitre présente le contexte dans lequel s'inscrit le projet, en détaillant sa problématique et ses objectifs. Il met en évidence la solution développée, ainsi que le processus de développement et la méthodologie adoptée pour mener à bien ce projet.

2. Présentation du projet

2.1 Problématique

Actuellement, dans la pharmacie, la gestion du stock et du suivi des médicaments repose principalement sur des procédures manuelles ou sur des outils bureautiques simples comme Microsoft Excel ou des registres papier. Voici comment se déroule généralement la gestion au quotidien :

- **Réception des médicaments** : Lors de la livraison par les fournisseurs, les médicaments sont enregistrés manuellement dans un cahier ou une feuille de calcul.
- **Rangement en stock** : Les boîtes sont rangées dans des étagères ou armoires selon la logique du pharmacien (par ordre alphabétique ou catégorie thérapeutique).
- **Suivi des ventes** : Chaque vente est notée ou passée via une caisse simple, mais il n'y a pas toujours de lien automatique avec la sortie de stock.
- **Vérification des dates de péremption** : Elle se fait de manière visuelle et ponctuelle, souvent une fois par mois ou selon la disponibilité du personnel.
- **Commande de nouveaux médicaments** : Elle est déclenchée lorsque le pharmacien s'aperçoit que le stock est faible, ou lorsque des ruptures sont constatées au comptoir.
- **Alertes et suivi** : Il n'existe aucun système d'alerte automatisé concernant les ruptures de stock ou les médicaments proches de l'expiration.

Cependant, ce mode de fonctionnement, basé sur des outils manuels ou semi-numériques, révèle progressivement un certain nombre de faiblesses. L'absence de suivi automatisé complique la gestion des stocks en temps réel, ralentit la détection des produits proches de leur date de péremption, et augmente le risque de ruptures inattendues. De plus, la charge de travail liée aux vérifications régulières, au contrôle des quantités disponibles et au passage des commandes pèse sur le personnel et multiplie les risques d'erreurs humaines. Ces limites rendent difficile une organisation optimale et soulignent la nécessité de mettre en

place un système numérique capable de fiabiliser, simplifier et sécuriser la gestion quotidienne de la pharmacie.

2.2 Objectifs du projet

Ce projet vise à améliorer l'efficacité opérationnelle de la pharmacie, en optimisant l'utilisation des ressources, en réduisant les pertes liées aux médicaments expirés, et en favorisant une meilleure organisation interne. Il permet également d'atteindre une gestion plus transparente et plus économique, tant en termes de coûts que de temps, tout en renforçant l'autonomie et la responsabilité de l'équipe officinale à l'échelle locale.

Le projet consiste en la mise en place d'un système qui permettra de :

- Améliorer la gestion des stocks et prévenir les ruptures avec des alertes,
- Assurer un suivi rigoureux des dates d'expiration des médicaments,
- Faciliter la gestion des clients et de leurs ordonnances,
- Optimiser la planification des rendez-vous,
- Fournir des analyses et statistiques pertinentes pour la prise de décision.

2.3 Solution proposée

La solution développée est une application web intelligente dédiée à la gestion des pharmacies. Elle permet d'automatiser plusieurs tâches critiques liées à la gestion quotidienne, notamment l'enregistrement des médicaments, le suivi des ventes, la gestion des stocks et la surveillance des dates de péremption. Le système génère automatiquement des alertes en cas de niveau de stock critique ou d'approche de la date d'expiration, évitant ainsi les ruptures et les pertes. Un module de gestion des clients et des ordonnances facilite la traçabilité des traitements, tandis qu'un calendrier permet de planifier les rendez-vous avec les patients. Par ailleurs, la plateforme intègre un tableau de bord décisionnel affichant des statistiques sur les ventes, les commandes, et les stocks. Pour renforcer l'aspect prédictif de la gestion, un modèle d'intelligence artificielle a été intégré afin d'anticiper les besoins en approvisionnement, en se basant sur l'historique des ventes et des consommations.

2.4 Démarche et planification

2.4.1 La méthode SCRUM

La méthode Scrum est une méthode agile de gestion de projets informatiques privilégiant la communication, et facilitant les réorientations opportunes. C'est désormais la

méthode privilégiée pour les démarches dites "agiles". Fort de son succès dans l'univers informatique, elle est maintenant déployée en entreprise comme nouvelle organisation du fonctionnement en "mode projet".

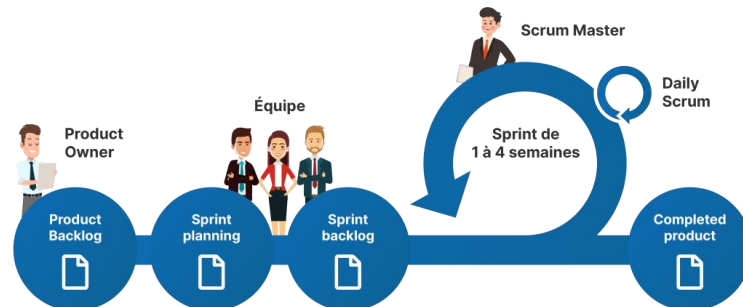


Figure 1 Processus SCRUM

2.4.2 Pourquoi SCRUM ?

Scrum est de très loin la méthodologie la plus utilisée parmi les méthodes Agile existantes. Elle est donc la plus éprouvée, documentée et supportée. Livres, blogs, formations, vidéos, 11 associations, conférences traitant de Scrum ne manquent pas et bon nombre de ces ressources sont accessibles gratuitement. On pourrait pratiquement parler d'un standard Agile. Un autre atout important : Scrum est simple à comprendre. Sa maîtrise est en revanche difficile. [1] Les experts de Scrum, même ses fondateurs, le décrivent comme un « cadre de travail permettant de répondre à des problèmes complexes et changeants tout en livrant de manière productive et créative des produits de la plus grande valeur possible » Scrum propose un modèle de contrôle de processus basé sur l'empirisme. Il s'appuie sur trois piliers.

- **La transparence** : Scrum met l'accent sur le fait d'avoir un langage commun entre l'équipe et le management, qui doit permettre à tout observateur d'obtenir rapidement une bonne compréhension du projet.
- **L'inspection** : Scrum propose de faire le point sur les différents artefacts produits à intervalle régulier, afin de détecter toute variation indésirable.
- **L'adaptation** : Si une dérive est constatée pendant l'inspection, le processus doit alors être adapté. Scrum fournit des rituels, durant lesquels cette adaptation est possible. Il s'agit de sprint planning, de daily scrum, et de sprint review qu'on va détailler dans la section suivante.

2.4.3 L'équipe et rôles

Les rôles durant le projet ont été répartis comme suit :

- **Product Owner** : Mr : ESSEABBAR Driss, NADIRI Abdeljalil, BOUSQAOU Halima, CHAREF Ayoub
- **Scrum Master** : IDRISSE Mohamed
- **Membre de l'équipe de développement** : MECHACH Marouane, IDRISSE Mohamed, BAANNI Zaineb

2.5 Identification des sprints

Numéro du sprint	Titre du sprint
Sprint 1	Authentification et Gestion de base
Sprint 2	Gestion des stocks et Alertes
Sprint 3	Planification et rendez-vous clients
Sprint 4	Suivi clients et ordonnances
Sprint 5	Tableau de bord et analyse
Sprint 6	Notification et Gestion des alertes

- ✓ *Sprint 1 : Authentification & Gestion de Base* : Ce sprint permet aux utilisateurs (pharmaciens, techniciens, administrateurs) de s'authentifier dans le système via un identifiant et un mot de passe. Il inclut également la mise en place des fonctionnalités de base telles que l'enregistrement et la gestion des médicaments avec leurs caractéristiques, la classification par catégories, la création de fiches clients complètes, ainsi que l'enregistrement et le suivi des ordonnances.
- ✓ *Sprint 2 : Gestion des Stocks & Alertes* : Ce sprint se concentre sur la gestion efficace des stocks, notamment la définition d'alertes pour les niveaux de stock bas ou les dates de péremption proches. Les utilisateurs peuvent visualiser les produits expirants bientôt, effectuer un suivi historique des mouvements de stock, planifier les livraisons fournisseurs et générer automatiquement des demandes de réapprovisionnement.
- ✓ *Sprint 3 : Planification & Rendez-vous Clients* : Ce sprint permet aux utilisateurs de planifier des rendez-vous clients pour les retraits ou consultations médicales, de recevoir des rappels automatiques pour les événements importants, et de consulter ces informations dans un calendrier configurable par jour, semaine ou mois.
- ✓ *Sprint 4 : Suivi Client & Ordonnances* : Ce sprint renforce la relation client en permettant au pharmacien de consulter l'historique des achats et ordonnances d'un client, de classer

les clients selon leur fréquence d'achat, et de suivre les traitements chroniques pour un accompagnement personnalisé.

- ✓ **Sprint 5 : Tableaux de Bord & Analyse** : Ce sprint est dédié aux outils d'aide à la décision : visualisation des KPI principaux, graphiques d'évolution des ventes, identification des produits les plus vendus, analyse des tendances saisonnières, rapports financiers détaillés, et modèles prédictifs de demande pour anticiper les besoins en stock.
- ✓ **Sprint 6 : Notifications & Gestion des Alertes** : Le dernier sprint centralise toutes les notifications du système dans une interface unique. Il permet de classer les alertes par niveau de priorité, de les marquer comme traitées ou à ignorer, et donne accès à l'historique des alertes pour audit et analyse.

2.6 Planification du projet

La planification est l'une des phases préliminaires les plus essentielles du projet. Elle consiste à anticiper l'enchaînement des étapes et à estimer leur durée respective. Pour modéliser l'avancement de mon projet, nous avons utilisé un diagramme de Gantt, un outil efficace pour organiser et ordonnancer les tâches sur une période définie. Grâce à sa visualisation claire et pratique, nous avons pu suivre et contrôler le déroulement de chaque phase du projet. Cet outil nous a également permis de représenter graphiquement les tâches, leur durée, leur séquence, ainsi que le calendrier global du projet.

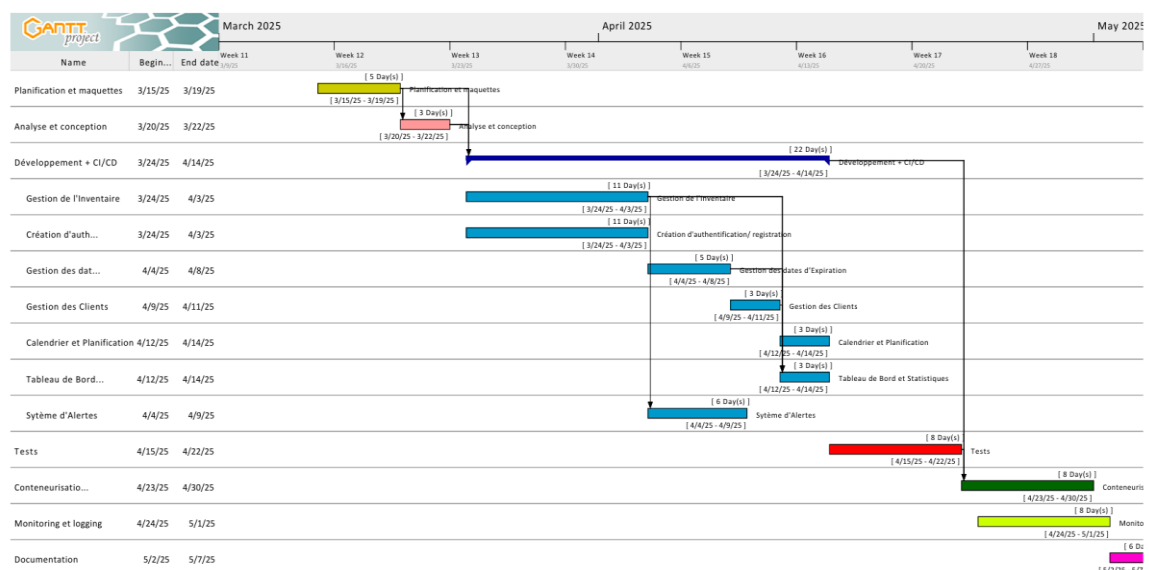


Figure 2 Diagramme de Gantt

3. Conclusion

Ce chapitre a été le point de départ pour l'élaboration du projet, dans la mesure où il décrivait le contexte général du projet, en présentant successivement la problématique, les objectifs généraux à atteindre, ainsi que la démarche et les étapes de sa mise en œuvre.

CHAPITRE 2 :

ANALYSE ET CONCEPTION

1. Introduction

Le deuxième chapitre s'articule autour de deux principaux volets : un premier où nous présenterons le langage UML, un deuxième où nous exposerons les diagrammes de cas d'utilisation, les diagrammes de séquence, et le diagramme de classes.

2. Langage de modélisation

2.1 Diagrammes utilisés

- Diagramme de cas d'utilisation : Un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. 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. Un acteur d'un cas d'utilisation peut être un humain ou un autre système externe à celui que l'on tente de définir. Les cas d'utilisation tentent d'éviter tout jargon technique et essaient au contraire d'adopter le langage de l'utilisateur final ou de l'expert du domaine.[2]
- Un diagramme de classe montre la structure statique du modèle d'information, particulièrement les choses qui existent, leur structure interne, et leurs relations aux autres choses. Un diagramme de classe ne doit présenter aucune information de nature temporelle. Contenu : classes, sous-classes, attributs et valeurs, méthodes, liens (multiplicité, généralisation, composition), catégories et dépendance. [3]
- Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont exécutées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquences sont organisés en fonction du temps qui s'écoule au fur et à mesure que nous parcourons la page. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence.[4]

2.2 Identification des principaux acteurs

Dans cette partie, nous procédons à l'identification des acteurs clés intervenant dans l'exploitation de notre système d'information. Un acteur se définit comme une entité externe – généralement un utilisateur – assumant un rôle déterminé et interagissant directement avec le système en cours d'étude.

- **Administrateur** : L'administrateur gère les utilisateurs (création, suppression, attribution des rôles) et configure les paramètres généraux du système (seuils d'alerte de stock, gestion des catégories, indicateurs KPI). Il a un accès complet à

toutes les fonctionnalités et peut exporter tous les rapports financiers et statistiques.

- **Pharmacien** : Le pharmacien est chargé de la gestion quotidienne des médicaments, des stocks, des clients et des ordonnances. Il crée et valide les commandes fournisseurs, surveille les dates d'expiration, ajuste les stocks après inventaire, et suit les alertes. Il peut aussi consulter les statistiques de vente et utiliser les prédictions de stock basées sur l'intelligence artificielle.
- **Technicien** : assiste dans la gestion des mouvements de stock (entrées/sorties), signale les anomalies, vérifie quotidiennement les alertes liées aux péremptions, et retire les produits expirés. Il peut créer des fiches clients basiques et enregistrer des ordonnances initiales, mais il n'a pas accès aux données sensibles ni à la modification des paramètres système.

3. Diagramme de cas d'utilisation général

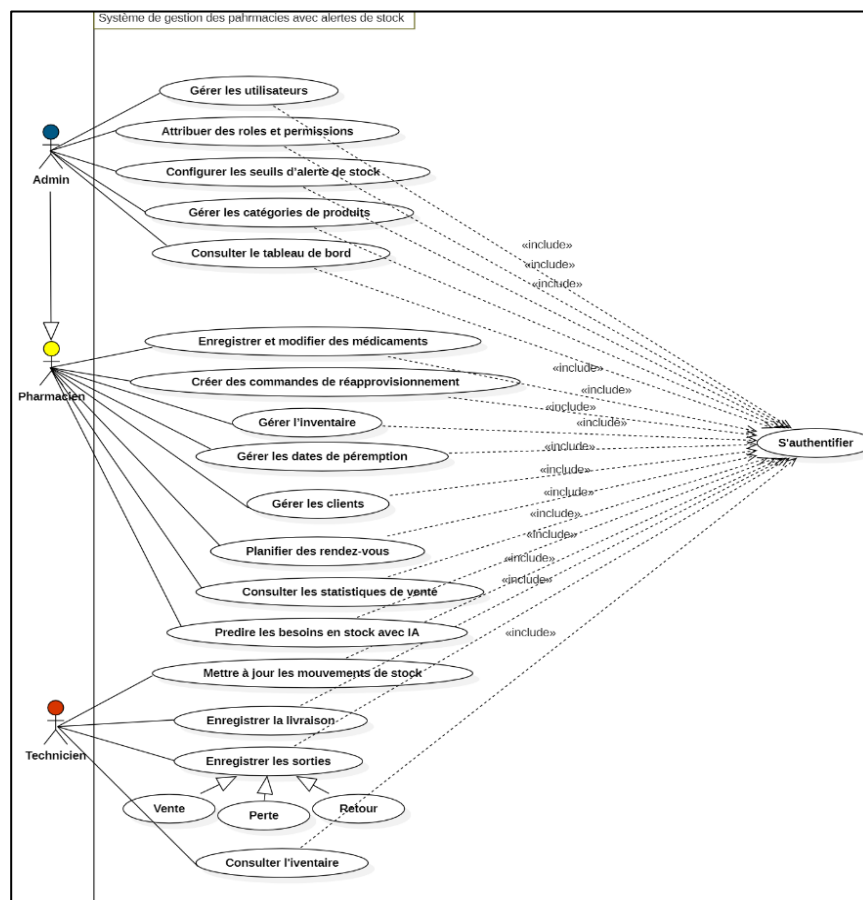


Figure 3 Diagramme de cas d'utilisation général

1. Les cas d'utilisation de l'administrateur (Hérite de pharmacien):

- **Gérer les utilisateurs** : L'administrateur peut créer, modifier, désactiver ou supprimer des comptes utilisateurs dans le système.
- **Attribuer des rôles et permissions** : Il définit les accès de chaque utilisateur selon son profil (ex : pharmacien, technicien).
- **Configurer les seuils d'alerte de stock** : Il paramètre les niveaux minimaux de stock pour chaque produit afin de déclencher les alertes de réapprovisionnement.
- **Gérer les catégories de produits** : Il organise les médicaments en catégories (antibiotiques, vitamines, etc.) pour faciliter la gestion et l'analyse.
- **Consulter le tableau de bord** : Il accède à une vue synthétique des indicateurs clés (état du stock, alertes, statistiques).

2. Les cas d'utilisation du pharmacien :

- **Enregistrer et modifier des médicaments** : Le pharmacien peut ajouter de nouveaux produits dans le système et mettre à jour leurs informations.
- **Créer des commandes de réapprovisionnement** : Il génère manuellement ou automatiquement des demandes de commande en fonction du stock.
- **Gérer l'inventaire** : Il consulte et ajuste le stock disponible, en s'assurant de sa fiabilité.
- **Gérer les dates de péremption** : Il surveille les produits proches de leur date d'expiration et suit les alertes.
- **Gérer les clients** : Il enregistre les clients, leurs ordonnances et suit les traitements chroniques.
- **Planifier des rendez-vous** : Il fixe des créneaux pour des consultations, retraits ou autres services.
- **Consulter les statistiques de vente** : Il visualise les ventes et les produits les plus demandés pour optimiser les approvisionnements.
- **Prédire les besoins en stock avec IA** : Permet d'utiliser l'intelligence artificielle pour analyser les données de vente et générer automatiquement des prévisions de besoins en stock, afin d'anticiper les ruptures et optimiser les commandes.

3. Les cas d'utilisation du technicien :

- **Mettre à jour les mouvements de stock** : Le technicien enregistre toutes les entrées et sorties de produits.
- **Enregistrer la livraison** : Il saisit les produits reçus lors d'une livraison fournisseur.
- **Enregistrer les sorties** : Il note les produits sortis du stock pour diverses raisons:
 - ✓ Vente : produit vendu.
 - ✓ Perte : produit perdu ou inutilisable.
 - ✓ Retour : produit retourné au fournisseur.
- **Consulter l'inventaire** : Il accède à la liste des produits disponibles et à leur état de stock.

4. Diagrammes de séquence

4.1 DS pour le cas d'utilisation : Prédire les besoins en stock avec IA

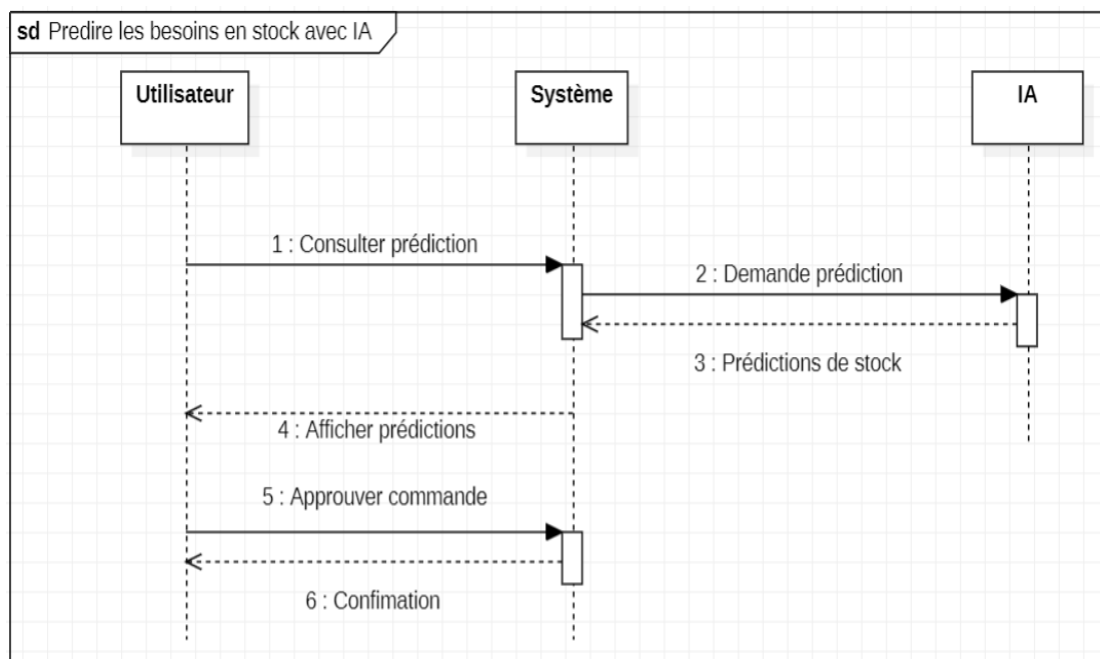


Figure 4 DS : Prédire les besoins en stock avec IA

- **Consultation de la prédiction par l'utilisateur** : L'utilisateur (Administrateur ou pharmacien), initie le processus en demandant au système de consulter les prédictions relatives aux besoins en stock. Cette demande peut être effectuée via une interface utilisateur intuitive, telle qu'un tableau de bord ou une application dédiée.

- **Demande de prédiction par le système à l'IA** : Une fois que l'utilisateur a initié la consultation, le système pharmaceutique transmet une requête à l'Intelligence Artificielle (IA). Cette IA est conçue pour analyser diverses données historiques et contextuelles, telles que :
 - ✓ Les ventes passées des médicaments.
 - ✓ Les tendances saisonnières ou cycliques.
 - ✓ Les événements externes (par exemple : épidémies, campagne des vaccinations).
 - ✓ Les niveaux actuels des stocks disponibles.
- **Prédiction de stock par l'IA** : L'IA traite les données reçues et génère des prédictions sur les besoins futurs en stock pour chaque médicament. Ces prédictions sont basées sur des algorithmes d'apprentissage automatique et de modélisation statistique, permettant de prendre en compte les variations complexes et les corrélations entre différents facteurs. Les résultats obtenus sont ensuite renvoyés au système central.
- **Affichage des prédictions à l'utilisateur** : Le système récupère les prédictions générées par l'IA et les affiche à l'utilisateur sous forme de rapports clairs et compréhensibles. Ces rapports peuvent inclure :
 - ✓ Une liste des médicaments dont les stocks doivent être réapprovisionnés.
 - ✓ Les quantités recommandées à commander pour chaque produit.
 - ✓ Des alertes spéciales pour les produits critiques ou ceux approchant leur date d'expiration.
 - ✓ Des graphiques ou visualisations facilitant la compréhension des tendances.
 L'utilisateur peut ainsi avoir une vue d'ensemble complète des besoins futurs et ajuster les commandes si nécessaire.
- **Approuver la commande** : Après avoir examiné les prédictions fournies par le système, l'utilisateur prend une décision concernant les commandes à passer. Il approuve les quantités recommandées ou, s'il le juge nécessaire, apporte des modifications manuelles. Par exemple, il peut augmenter ou diminuer les quantités en fonction de circonstances particulières non prises en compte par

l'IA (comme une augmentation imprévue de la demande due à un événement local). Cette étape garantit que les décisions finales tiennent compte à la fois des prévisions automatiques et de l'expertise humaine.

- **Confirmation de la commande par le système** : Une fois que l'utilisateur a approuvé les commandes, le système enregistre les informations et confirme la validation des actions. Le système peut alors générer automatiquement les documents nécessaires pour la commande auprès des fournisseurs, envoyer des notifications aux responsables logistiques, et mettre à jour les stocks prévus dans la base de données.

4.2 DS pour le cas d'utilisation : Créer des commandes de réapprovisionnement

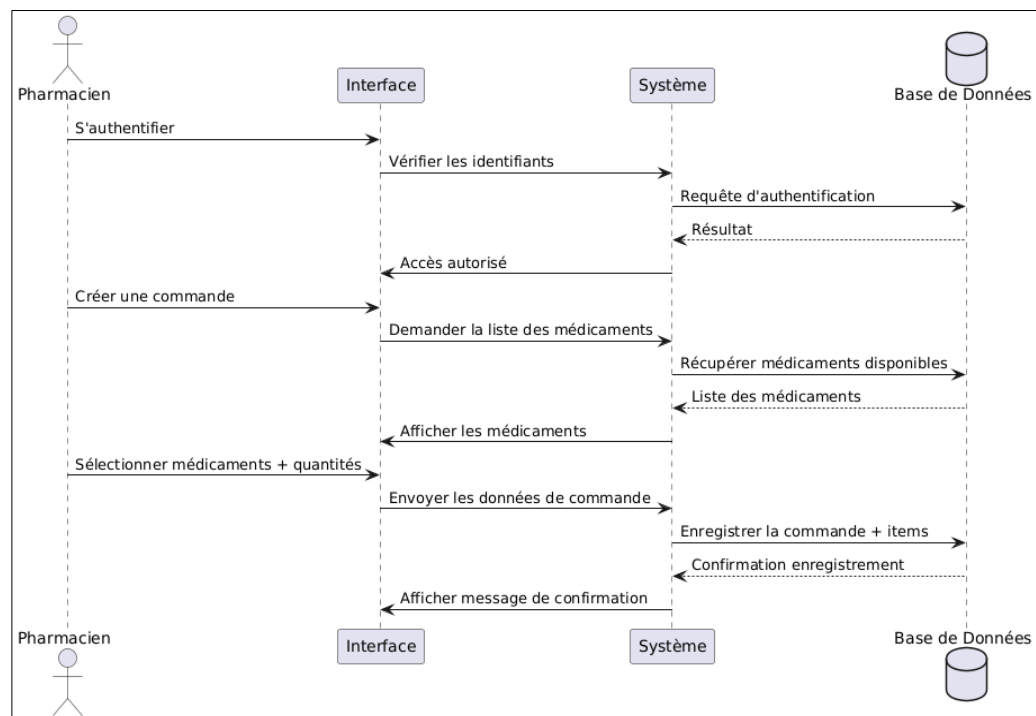


Figure 5 DS : Créer des commandes de réapprovisionnement

- **Authentification du pharmacien** : Le pharmacien commence par se connecter à l'application via l'interface.
 - ✓ L'interface envoie ses identifiants au système pour vérification.
 - ✓ Le système interroge la base de données pour vérifier la validité des identifiants.
 - ✓ Une réponse est renvoyée (accès autorisé ou refusé).

- **Accès au module de commande** : Une fois authentifié, le pharmacien accède à la section dédiée à la création de commandes.
- **Demande de la liste des médicaments disponibles** : L'interface demande au système les produits qui peuvent être commandés.
 - ✓ Le système consulte la base de données pour obtenir la liste complète des médicaments avec leurs quantités actuelles.
 - ✓ Cette liste est renvoyée à l'interface et affichée au pharmacien.
- **Sélection des médicaments et des quantités** : Le pharmacien sélectionne les médicaments à commander ainsi que les quantités nécessaires, selon l'état du stock ou les besoins prévisionnels.
- **Soumission de la commande** : Une fois la commande finalisée, les informations sont envoyées au système.
 - ✓ Le système enregistre la commande et ses détails dans la base de données.
 - ✓ Une confirmation d'enregistrement est retournée.
- **Retour visuel à l'utilisateur** : L'interface affiche un message de confirmation signalant que la commande a été correctement créée et enregistrée.

4.3 DS pour le cas d'utilisation : Gérer les dates de péremption

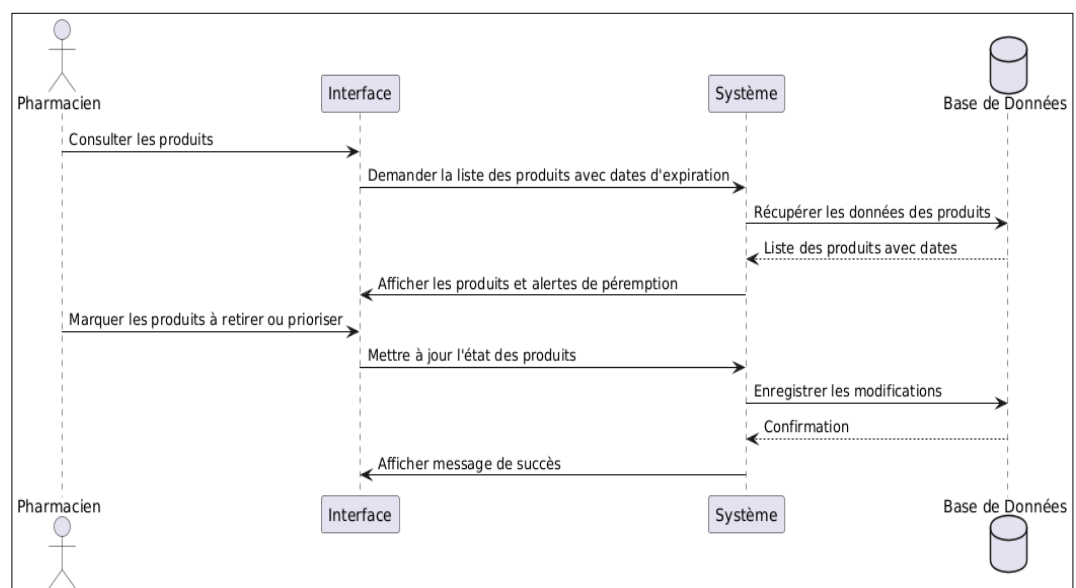


Figure 6 DS : Gérer les dates de péremption

- **Consultation des produits** : Le pharmacien accède à l'interface pour consulter la liste des médicaments et produits enregistrés dans le système.
- **Récupération des données** : Le système interroge la base de données pour extraire tous les produits avec leurs dates de péremption.
- **Analyse automatique** : Le système compare les dates actuelles aux dates d'expiration de chaque produit, et signale ceux qui sont :
 - ✓ Déjà périmés
 - ✓ Proches de péremption (selon un seuil défini, ex. 30 jours) Ces produits sont mis en alerte et affichés en priorité à l'écran.
- **Affichage des alertes** : L'interface affiche la liste des produits concernés, souvent avec une couleur ou une icône d'alerte pour faciliter l'identification.
- **Action du pharmacien** : Le pharmacien prend des mesures :
 - ✓ Marquer un produit à retirer du stock
 - ✓ Prioriser sa vente ou son écoulement si encore utilisable
 - ✓ Notifier une destruction ou un retour fournisseur
- **Mise à jour du statut** : Les actions sont envoyées au système, qui enregistre les modifications dans la base de données.
- **Confirmation du traitement** : Une notification confirme au pharmacien que les statuts des produits ont bien été mis à jour.

5. Diagramme de classes

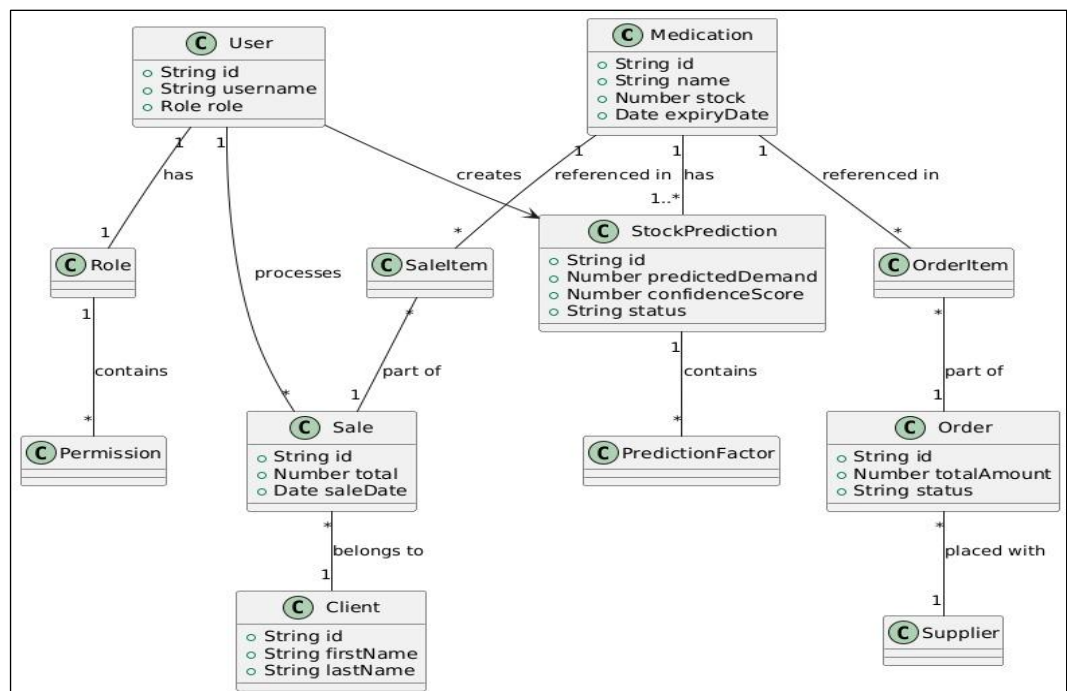


Figure 7 Diagramme de classe

- **User** est associé à un **Rôle**, qui lui donne accès à certaines fonctionnalités via les Permissions.
- Un **Rôle** contient plusieurs **Permissions**, ce qui permet de définir les droits d'accès de façon détaillée.
- Un **User** peut effectuer des ventes (classe **Sale**) qui sont liées à des **clients**.
- Une **vente** est composée de plusieurs éléments de vente (**SaleItem**), chacun correspondant à un **médicament** spécifique.
- Chaque **SaleItem** est lié à un seul **médicament**.
- Les **médicaments** peuvent être utilisés pour générer des **prédictions de stock** (StockPrediction), qui anticipent la demande.
- Une **prédiction de stock** est basée sur un ou plusieurs facteurs de prédiction (**PredictionFactor**).
- Les **médicaments** sont également référencés dans des éléments de commande (**OrderItem**), qui font partie d'une commande (**Order**).
- Une **commande** est passée à un fournisseur (**Supplier**) et regroupe plusieurs **OrderItem**.

6. Conclusion

En conclusion, nous avons passé en revue en détail les divers aspects conceptuels de notre projet, nous avons établi une base solide pour le développement de notre solution de gestion des pharmacies. En définissant clairement les interactions utilisateur-système à l'aide du diagramme de cas d'utilisation et en structurant la conception du système à l'aide du diagramme de classe, nous avons établi une vision cohérente de notre solution.

CHAPITRE 3 :

TECHNOLOGIES ET OUTILS UTILISES

1. Introduction

Dans ce chapitre, nous aborderons le cadre technique de notre application en détaillant les différents outils et technologies utilisés tout au long de sa réalisation. Nous explorerons les choix technologiques adoptés pour répondre aux besoins définis.

2. Technologies utilisées



Figure 8 logo de spring

En informatique, Spring est un framework open source pour construire et définir l'infrastructure d'une application Java3, dont il facilite le développement et les tests. En 2004, Rod Johnson a écrit le livre *Expert One-on-One J2EE Design and Development*4 qui explique les raisons de la création de Spring.[5]

Le choix de Spring Boot pour le backend s'explique par sa capacité à simplifier le développement d'API REST robustes et bien structurées, essentielles pour une application accessible à la fois sur le web et le mobile comme celle de notre projet. Grâce à sa compatibilité avec les bases de données, sa gestion automatique des dépendances et son intégration facile avec des services comme la sécurité et le déploiement cloud, Spring Boot permet de créer un backend performant, évolutif et maintenable.



Figure 9 logo de next.js

Next.js est un framework React qui vous permet de créer des sites web statiques et des applications web surpuissantes, adaptés au référencement et extrêmement conviviaux à l'aide du framework React. Next.js est connu pour offrir la meilleure expérience aux développeurs lors de la création d'applications prêtes pour la production avec toutes les fonctionnalités dont vous avez besoin.[6]

Le choix de Next.js permet de bénéficier des avantages de React tout en offrant un rendu côté serveur (SSR) et une génération statique (SSG), améliorant les performances et le référencement. Sa structure simple, son routage automatique et son intégration fluide avec les

API en font une solution idéale pour développer des interfaces web rapides et évolutives. Il dispose d'une prise en charge de TypeScript, d'un regroupement intelligent, d'une pré-lecture de route, et plus encore, sans aucune configuration supplémentaire.



Figure 10 logo de tailwind

Tailwind CSS est un framework CSS open source. La fonctionnalité principale de cette bibliothèque est, contrairement à d'autres frameworks CSS comme Bootstrap, qu'elle ne procure pas une série de classes prédéfinies pour des éléments tels que des boutons ou des tables. À la place, Tailwind crée une liste de classes CSS « utilitaires » pouvant être utilisés pour ajouter un style à chaque élément en les mélangeant et en les agençant.[7]



Figure 11 logo de HTML5

Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour écrire les pages web. Il s'agit d'un format ouvert très utilisé en informatique.[8]



Figure 12 logo de docker

Docker est une plateforme de conteneurisation lancée en 2013. Elle permet d'emballer des applications et leurs dépendances dans des conteneurs isolés, qui peuvent être exécutés sur n'importe quel serveur. Contrairement à la virtualisation, Docker utilise la conteneurisation, une approche plus légère qui s'appuie sur certaines parties du système d'exploitation hôte.[9]



Figure 13 logo de flask

Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires.[10]

Le choix de **Flask** s'explique par sa légèreté, sa simplicité d'utilisation et sa flexibilité, ce qui en fait une solution idéale pour créer rapidement des API RESTful. Dans le cadre de ce projet de prédiction de stock pharmaceutique, Flask permet de structurer et d'exposer efficacement les fonctionnalités du modèle d'intelligence artificielle (comme l'entraînement, la prédiction ou l'évaluation) via des endpoints accessibles à d'autres systèmes ou interfaces. De plus, grâce à sa large communauté et sa compatibilité avec de nombreuses bibliothèques Python que nous avons utilisés dans notre modèle de prédiction (Pandas, NumPy, TensorFlow, etc.), Flask facilite l'intégration d'éléments de data science et de machine learning dans une application web fonctionnelle et évolutive.

3.Outils de base de données : MySQL



Figure 14 logo du SGBD MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde⁴, autant par le grand public que par des professionnels, avec Oracle, PostgreSQL et Microsoft SQL Server.en concurrence Il est développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multi-utilisateur.[11]

4. Environnement de travail

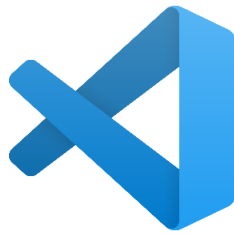


Figure 15 logo de VS Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. [12]



Figure 16 logo d'Eclipse

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.[13]



Figure 17 logo d'Android studio

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Avant Android Studio, de 2009 à 2014, Google propose comme environnement de développement officiel une distribution spécifique de l'environnement Eclipse, contenant notamment le SDK d'Android.[14]

5. Outils de gestion de projet



Figure 18 logo de GanttProject

GanttProject est un logiciel libre de gestion de projet écrit en Java, ce qui permet de l'utiliser sur divers systèmes d'exploitation (Windows, Linux, MacOS). Il permet de planifier un projet à travers la réalisation de diagrammes de Gantt ainsi que des diagrammes de ressources et des réseaux PERT.[15]



Figure 19 logo d'Azure DevOps

Azure DevOps est une plateforme de Microsoft qui fournit un ensemble d'outils de développement intégrés pour gérer tout le cycle de vie des applications, de la planification à la livraison. Elle inclut des services comme Azure Repos (contrôle de version), Azure Pipelines (intégration et déploiement continu), Azure Boards (gestion de projet), Azure Test Plans (tests) et Azure Artifacts (gestion des paquets). Accessible via le cloud ou en version locale (Azure DevOps Server), elle facilite la collaboration entre les équipes, l'automatisation des processus et l'amélioration continue du développement logiciel.[16]

6. Conclusion

Après avoir décrit les différentes technologies qu'on a utilisées le chapitre suivant sera consacré à la mise en œuvre de l'application.

CHAPITRE 4 :

IMPLEMENTATION ET MISE EN OEUVRE

1. Introduction

Dans cette section, nous explorerons les étapes pratiques de développement, de codage de notre système. À travers des descriptions, des exemples de code et des captures d'écran, nous démontrerons le processus de création des composants principaux de l'application.

2. Architecture logiciel

- Type d'application : L'application sera développée sous forme d'une plateforme web ou hybride (web + mobile), en s'appuyant sur une architecture client-serveur robuste avec une API RESTful pour gérer efficacement les interactions entre l'interface utilisateur (frontend) et la logique métier (backend)
- Backend : Utilisation de Java avec le framework Spring Boot. La gestion des sessions et l'authentification des utilisateurs seront assurées par des mécanismes sécurisés tels que JWT (JSON Web Tokens) ou OAuth2.
- Frontend : Utilisation de Next.js et une conception responsive avec des frameworks CSS comme Bootstrap ou Tailwind CSS.
- Base de données : SGBD relationnel pour stocker les données structurées (MYSQL)
- Hébergement : Conteneurs Docker pour une gestion simplifiée des environnements de développement et de production.

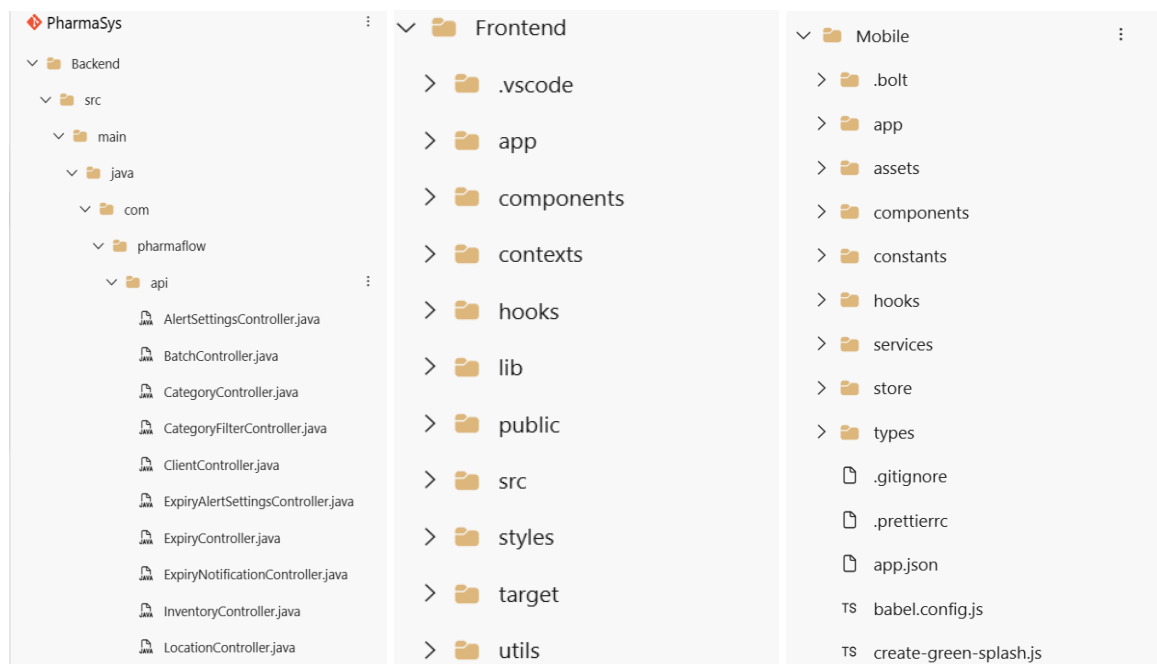


Figure 20 Structure du projet

3. Interfaces graphiques

3.1 Les interfaces web

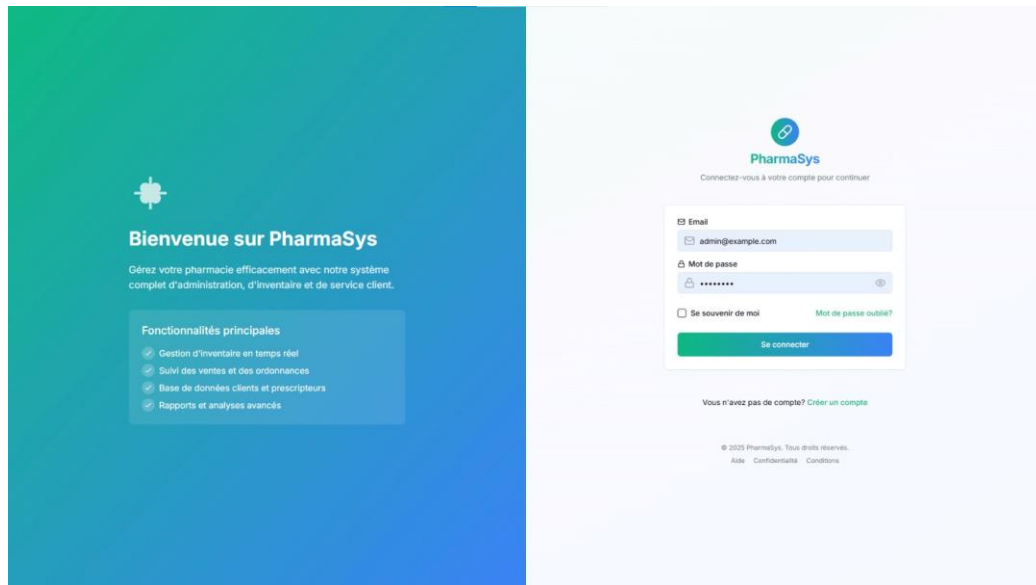


Figure 21 Interface de login

La figure représente la page de connexion au système PharmaSys, permettant aux utilisateurs existants de se connecter à leur compte en saisissant leur email et leur mot de passe.

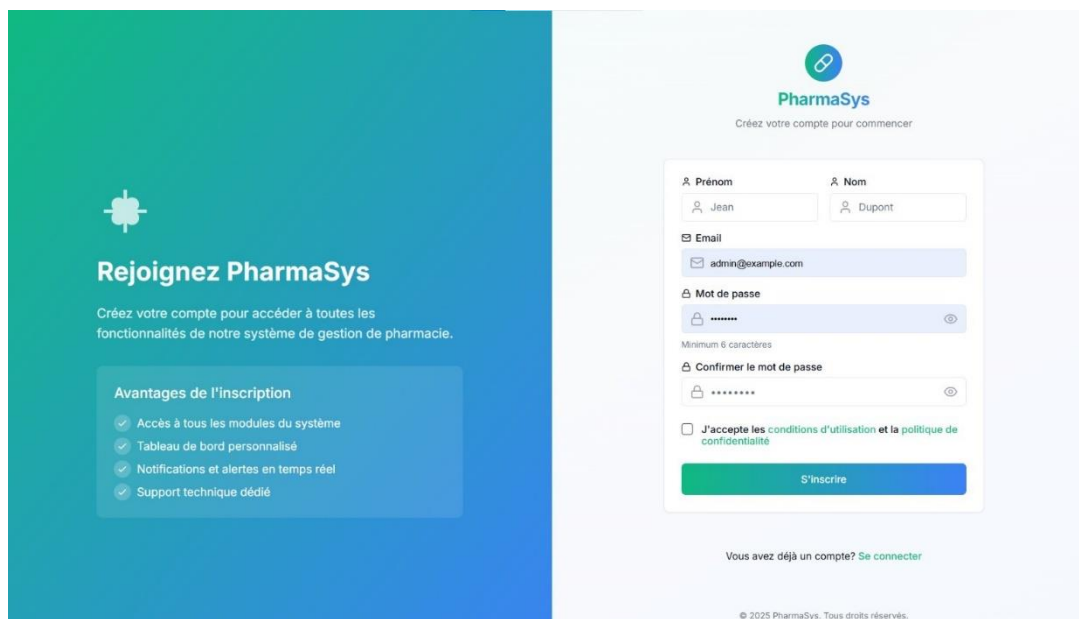


Figure 22 Interface du register

Cette figure montre la page d'enregistrement, où l'utilisateur peut créer un compte en fournissant des informations telles que le prénom, le nom, l'email et le mot de passe.

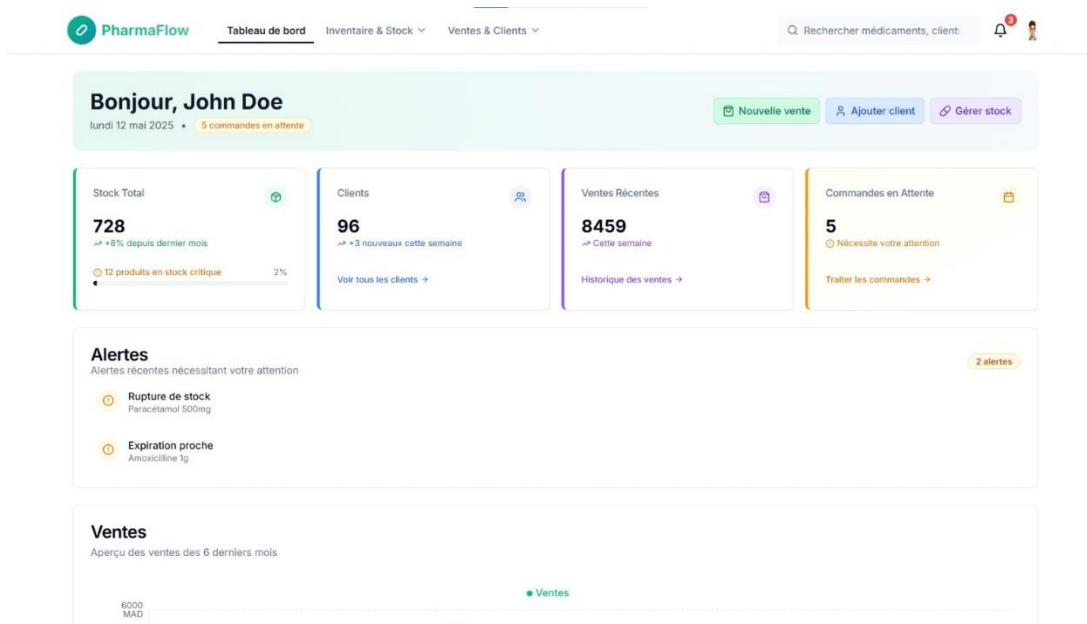


Figure 23 Interface du tableau de bord

La figure représente l'interface du Tableau de bord de notre application, qui sert de point d'entrée principal pour les utilisateurs. Cette page fournit un aperçu global des principales métriques et alertes liées à la gestion de pharmacie, permettant aux utilisateurs de surveiller l'état de leur activité commerciale et logistique.

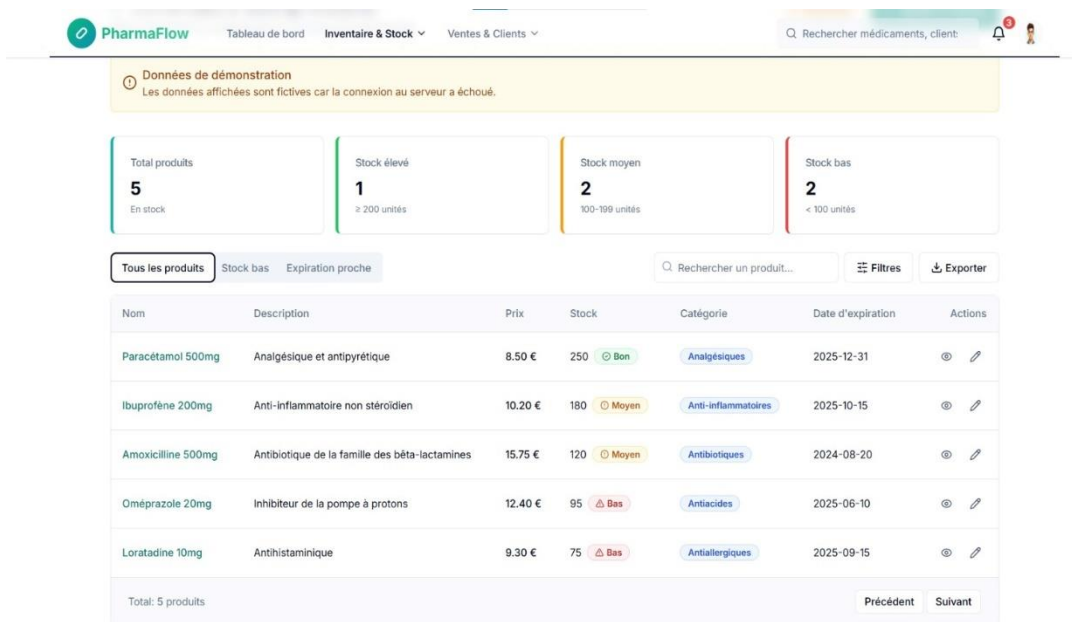


Figure 24 Interface de gestion des produits

Cette interface de gestion des produits est conçue pour offrir une vue claire et structurée des stocks de médicaments, avec des indicateurs visuels pour identifier rapidement les produits critiques (stocks bas ou expirations proches). Les filtres et la barre de recherche facilitent la

localisation rapide des informations nécessaires, tandis que les actions directes simplifient la gestion quotidienne des stocks.

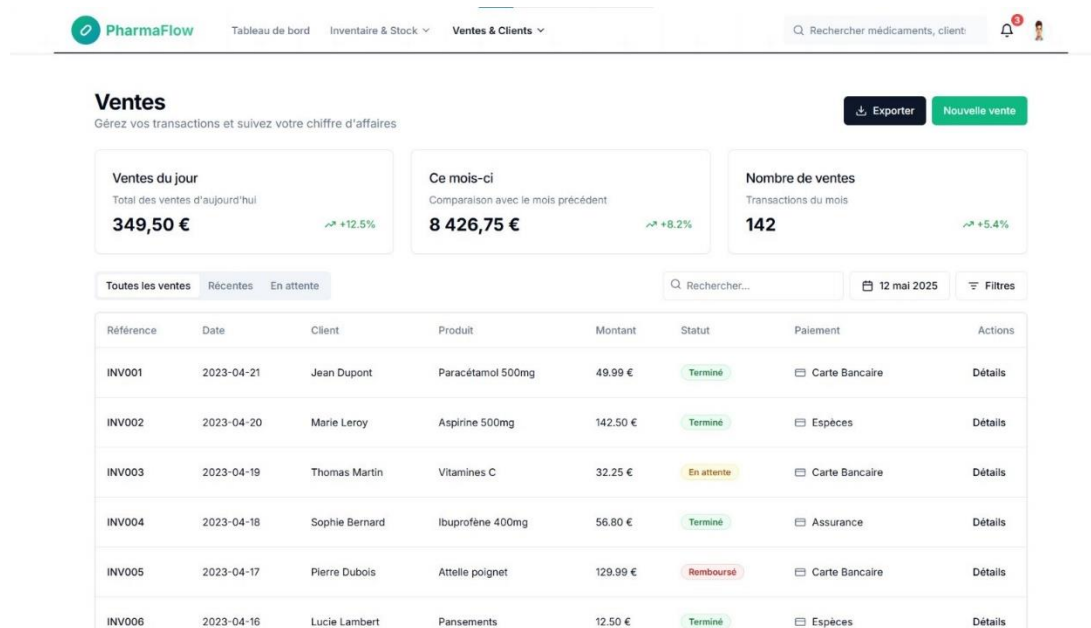


Figure 25 Interface de gestion des ventes

Cette interface, située sous l'onglet Ventes & Clients, permet aux utilisateurs de gérer et suivre leurs transactions commerciales. Elle fournit un aperçu global des ventes du jour, du mois en cours par rapport au mois précédent, ainsi que le nombre total de transactions pour le mois. Des boutons pratiques, tels qu'Exporter et Nouvelle vente, facilitent la gestion et la création de nouvelles transactions.

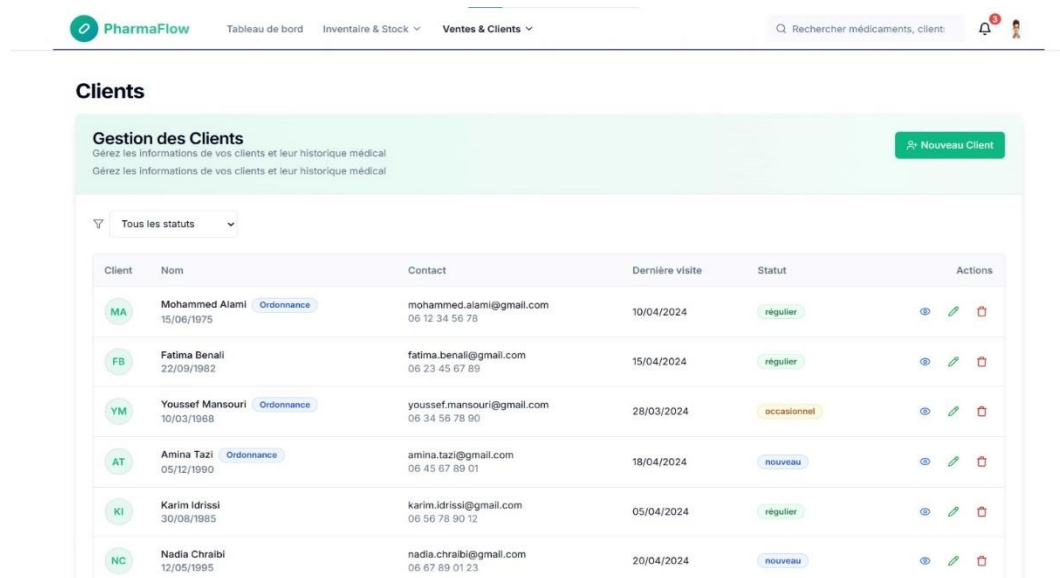


Figure 26 Interface de gestion des clients

3.2 Les interfaces mobiles

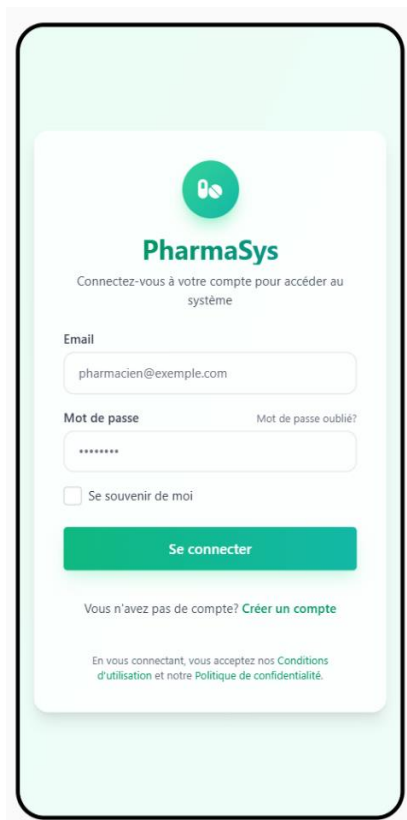


Figure 27 Écran de connexion mobile

La figure représente l'écran de connexion de l'application mobile PharmaSys. Cette interface simple et intuitive permet aux utilisateurs de s'authentifier en saisissant leur adresse e-mail et leur mot de passe. Elle inclut également une option pour rester connecté et un lien vers la création d'un compte pour les nouveaux utilisateurs.

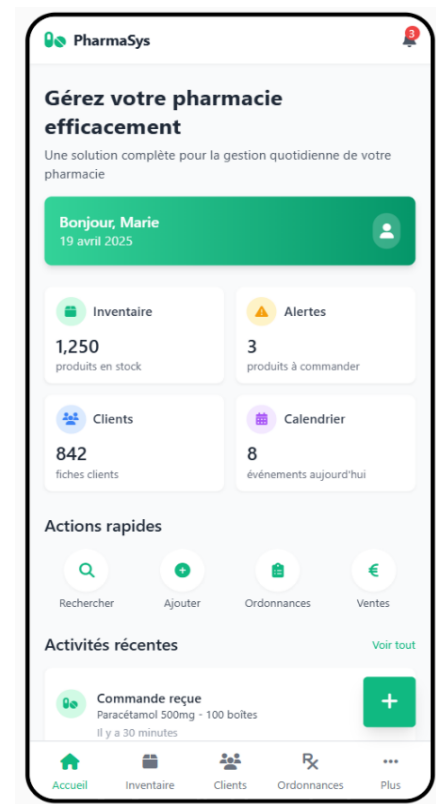


Figure 28 Écran d'accueil mobile

Cette interface mobile, accessible depuis l'écran d'accueil de PharmaSys, offre une vue d'ensemble des principales métriques et fonctionnalités liées à la gestion quotidienne d'une pharmacie. En bas de l'écran, une section Actions rapides propose des raccourcis vers les fonctions essentielles.

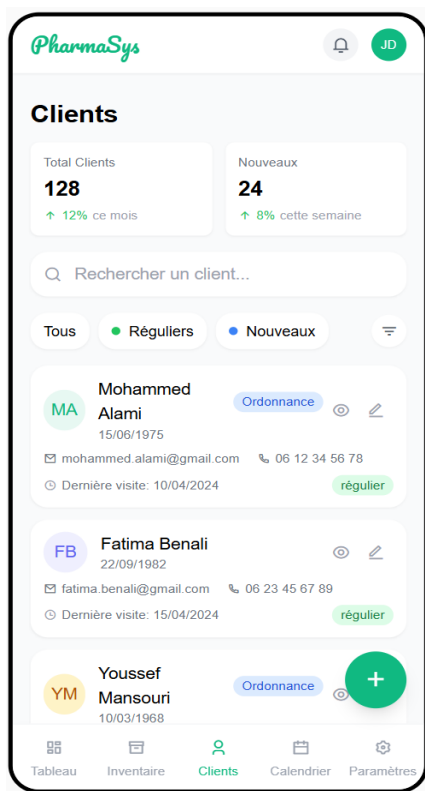


Figure 29 Écran mobile de gestion des clients

Cette interface mobile, accessible depuis l'onglet Clients, permet aux utilisateurs de gérer leur base de données clientèle. L'écran affiche un résumé global avec le nombre total de clients et les nouveaux clients ajoutés cette semaine. Les filtres rapides facilitent la segmentation des clients. Un bouton vert en bas à droite (+) permet d'ajouter rapidement un nouveau client.

4. Conclusion

Les interfaces web et mobile de l'application présentées dans ce chapitre ont été conçues pour offrir une expérience utilisateur fluide, intuitive et fonctionnelle. Les différentes pages présentées illustrent une organisation claire des informations, des outils de gestion performants et une navigation simplifiée, adaptée aux besoins quotidiens d'une pharmacie.

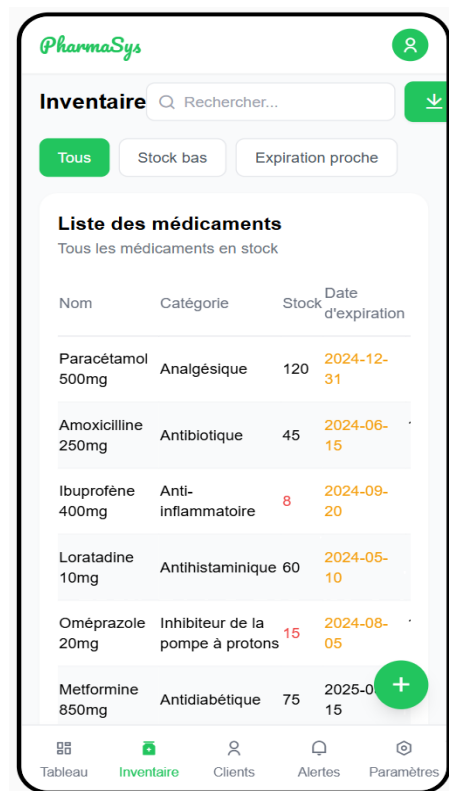


Figure 30 Écran mobile de gestion de l'inventaire

Cette interface mobile, accessible depuis l'onglet Inventaire, permet aux utilisateurs de gérer leur stock de médicaments. L'écran affiche une liste détaillée des produits en stock, avec des filtres rapides pour trier les médicaments par catégorie. Un bouton vert en bas à droite (+) facilite l'ajout de nouveaux médicaments au stock. La barre de navigation en bas offre un accès direct aux autres sections de l'application.

Conclusion générale

En conclusion, ce projet de développement d'une application web et mobile dédiée à la gestion des pharmacies répond efficacement aux besoins croissants de digitalisation dans le secteur pharmaceutique. Il apporte une solution innovante aux limites de la gestion manuelle en automatisant le suivi des stocks et des dates de péremption, tout en facilitant la gestion des ordonnances et des rendez-vous. L'architecture adoptée, reposant sur des technologies robustes et évolutives telles que Next.js, Spring Boot, et l'intégration de l'intelligence artificielle, permet d'assurer à la fois performance, sécurité, et facilité d'utilisation.

Ce travail a permis non seulement de mettre en œuvre des compétences techniques variées, mais aussi de répondre à une problématique réelle par une solution concrète et fonctionnelle. Le développement de ce système marque une étape importante vers la modernisation de la gestion pharmaceutique, tout en assurant un service plus fiable et plus réactif pour les professionnels de santé comme pour les patients.

Perspectives

Plusieurs pistes d'amélioration peuvent être envisagées pour faire évoluer cette application :

- Intégration d'un module de reconnaissance d'ordonnances basé sur la vision par ordinateur pour faciliter la saisie des prescriptions.
- Ajout d'un tableau de bord analytique permettant aux pharmaciens de suivre les tendances de consommation et les performances de gestion.
- Déploiement d'un système de notifications multicanal (SMS, email, notifications push) pour alerter les utilisateurs des renouvellements de stock, de rendez-vous ou d'expirations imminentes.
- Connexion à des bases de données médicales officielles pour enrichir les informations sur les médicaments.
- Extension du système à une échelle régionale ou nationale, avec une gestion multi-pharmacies et une interopérabilité avec les plateformes de santé.

Ces évolutions visent à faire de cette application non seulement un outil de gestion, mais aussi un véritable assistant intelligent pour les pharmaciens, contribuant ainsi à une meilleure qualité des soins et à une gestion optimisée des ressources médicales

Références

- [1] Agiliste. "Introduction aux méthodes agiles." Agiliste, <https://agiliste.fr/introduction-methodes-agiles/>. Consulté le 07 mai 2025.
- [2] Thésaurus des activités gouvernementales du Québec. "Modèle conceptuel." <https://www.thesaurus.gouv.qc.ca/tag/terme.do?id=MDL416>. Consulté le 07 mai 2025.
- [3] "Wikipedia." *Cas d'utilisation*. Wikipédia, https://fr.wikipedia.org/wiki/Cas_d%27utilisation. Consulté le 07 mai 2025.
- [4] "Wikipedia." *Diagramme d'activité*. Wikipédia, https://fr.wikipedia.org/wiki/Diagramme_d%27activit%C3%A9. Consulté le 07 mai 2025.
- [5] "Wikipedia." *Spring (framework)*. Wikipédia, [https://fr.wikipedia.org/wiki/Spring_\(framework\)](https://fr.wikipedia.org/wiki/Spring_(framework)). Consulté le 07 mai 2025.
- [6] Kinsta. "Qu'est-ce que Next.js ? Un regard sur le framework JavaScript populaire." *Kinsta*, 3 décembre 2024, <https://kinsta.com/fr/base-de-connaissances/next-js/>. Consulté le 07 mai 2025.
- [7] "Wikipedia." *Tailwind CSS*. Wikipédia, https://fr.wikipedia.org/wiki/Tailwind_CSS. Consulté le 07 mai 2025.
- [8] "Wikipedia." *Hypertext Markup Language*. Wikipédia, https://fr.wikipedia.org/wiki/Hypertext_Markup_Language. Consulté le 07 mai 2025.
- [9] "Wikipedia." *Docker (logiciel)*. Wikipédia, [https://fr.wikipedia.org/wiki/Docker_\(logiciel\)](https://fr.wikipedia.org/wiki/Docker_(logiciel)). Consulté le 07 mai 2025.
- [10] "Wikipedia." *Flask (framework)*. Wikipédia, [https://fr.wikipedia.org/wiki/Flask_\(framework\)](https://fr.wikipedia.org/wiki/Flask_(framework)). Consulté le 07 mai 2025.
- [11] "Wikipedia." *MySQL*. Wikipédia, <https://fr.wikipedia.org/wiki/MySQL>. Consulté le 07 mai 2025.
- [12] Développez.com. "L'extension C++ pour Visual Studio Code passe en version 1.0 et apporte un riche ensemble de fonctionnalités de productivité." *Développez.com*, 10 mai 2024, <https://visualstudio.developpez.com/actu/308869/L-extension-Cplusplus-pour-Visual-Studio-Code-passe-en-version-1-0-et-apporte-un-riche-ensemble-de-fonctionnalites-de-productivite-adaptables-a-diverses-plateformes-et-architectures/#:~:text=Visual%20Studio%20Code%20est%20un,du%20code%20et%20Git%20int%C3%A9gr%C3%A9>. Consulté le 12 mai 2025.
- [13] Techno-Science.net. "Eclipse (logiciel)." *Techno-Science.net*, <https://www.techno-science.net/glossaire-definition/Eclipse-logiciel.html>. Consulté le 07 mai 2025.

- [14] Android Developers. "Introduction à Android Studio." *Android Developers*, <https://developer.android.com/studio/intro?hl=fr>. Consulté le 07 mai 2025.
- [15] "Wikipedia." *GanttProject*. Wikipédia, <https://fr.wikipedia.org/wiki/GanttProject>. Consulté le 07 mai 2025.
- [16] Microsoft. "Azure DevOps Services." *Microsoft Azure*, <https://azure.microsoft.com/fr-fr/products/devops/?msockid=1801e521c947624f247ff094c8916373>. Consulté le 07 mai 2025.