

3ème année ingénieur informatique

Spécialité : Génie Logiciel

Module : Génie Logiciel

Rapport de Mini projet

Conception et réalisation d'une application de gestion des livraisons de médicaments d'une pharmacie centrale

Réalisé par :

SAHI Nour El Imene
MAOUCHÉ Lina

Groupe : 2

Chargée de cours :

Mme HAMZA L.

Chargées de TP :

Mme BAIR N.
Mme BOUSSOUFA S.

Remerciements

Tout d'abord, nous remercions **Allah le Tout-Puissant** de nous avoir donné le courage et la patience nécessaires pour mener ce travail à son terme.

Nous tenons à exprimer notre profonde gratitude à **Madame Hamza**, notre chargée de cours de *Génie Logiciel*, pour la qualité de son enseignement, la richesse de ses cours et l'inspiration qu'elle nous a transmise tout au long du semestre. Ses explications claires et sa rigueur nous ont permis d'acquérir les notions essentielles à une bonne conception logicielle.

Nous adressons également nos sincères remerciements à nos **chargées de TP, Madame Bair et Madame Boussoufa**, pour leur accompagnement, leurs conseils précieux et pour nous avoir proposé un projet qui nous a permis de mettre en œuvre concrètement nos connaissances théoriques. Ce mini-projet a représenté une expérience très enrichissante, tant sur le plan technique que personnel.

Enfin, nous remercions toutes les personnes, de près ou de loin, qui ont contribué à la réalisation de ce travail, ainsi que l'ensemble du corps enseignant du département d'informatique pour leurs efforts constants dans la formation des étudiants.

Table des matières

Table des matières	2
Table des figures	5
Introduction générale	6
1 Élaboration des besoins fonctionnels	8
1.1 Page d'accueil	8
1.2 Authentification	8
1.3 Tableau de bord (Dashboard)	8
1.4 Consultation globale	9
1.5 Gestion des clients	9
1.6 Gestion des médicaments	10
1.7 Gestion des livraisons	10
1.8 Gestion des utilisateurs (Admin uniquement)	11
2 Règles métier et contraintes	12
2.1 Principe général	12
2.2 Règles d'unicité et d'identification	12
2.3 Contraintes sur les utilisateurs et les clients	12
2.4 Contraintes sur les médicaments	13
2.5 Contraintes sur les livraisons	13
2.6 Contraintes sur les opérations et la cohérence des données	13
2.7 Sécurité et intégrité	13

2.8	Synthèse	14
3	Modélisation UML	15
3.1	Diagramme des cas d'utilisation	15
3.2	Description textuelle des cas d'utilisation	16
3.2.1	Cas d'utilisation : S'authentifier	16
3.2.2	Cas d'utilisation : Gérer les clients	17
3.2.3	Cas d'utilisation : Gérer les médicaments	18
3.2.4	Cas d'utilisation : Gérer les livraisons	19
3.2.5	Cas d'utilisation : Gérer les utilisateurs	20
3.2.6	Cas d'utilisation : Consulter les statistiques	20
3.3	Diagrammes de séquence	21
3.4	Diagramme de classes	24
4	Modèle Relationnel	25
4.1	Tables principales du système	25
4.2	Remarques sur le modèle relationnel	25
5	Script SQL	26
5.1	Création de la base et des tables	26
5.2	Insertion de données d'exemple	27
5.3	Affichage formaté des identifiants	28
6	Interfaces de l'application	30
6.1	Page d'accueil	30
6.2	Fonctionnalités supplémentaires	31
6.3	Authentification	31
6.4	Inscription d'un utilisateur	31
6.5	Tableau de bord	32
6.6	Consultation globale	32
6.7	Gestion des clients	33

6.8	Gestion des médicaments	33
6.9	Gestion des livraisons	33
6.10	QR Code pour la livraison	34
6.11	Ajouter une livraison	34
6.12	Suppression d'une livraison	34
6.13	Gestion des utilisateurs (admin)	35
	Conclusion générale	36

Table des figures

3.1	Diagramme des cas d'utilisation	16
3.2	Diagramme de séquence pour le cas d'utilisation « S'authentifier »	22
3.3	Diagramme de séquence pour le cas d'utilisation « Gérer les livraisons » .	23
3.4	Diagramme de classes	24
6.1	Page d'accueil avec navigation principale.	30
6.2	Exemples de fonctionnalités accessibles depuis le tableau de bord.	31
6.3	Page de connexion à l'application.	31
6.4	Formulaire d'inscription pour un nouvel utilisateur.	31
6.5	Tableau de bord montrant les statistiques et indicateurs clés.	32
6.6	Autre vue du tableau de bord.	32
6.7	Vue globale des livraisons et des stocks.	32
6.8	Interface de gestion des clients.	33
6.9	Interface de gestion des médicaments.	33
6.10	Interface de gestion des livraisons.	33
6.11	QR Code généré pour une livraison spécifique.	34
6.12	Interface pour ajouter une nouvelle livraison.	34
6.13	Interface pour supprimer une livraison.	34
6.14	Interface de gestion des utilisateurs	35
6.15	QR code du dépôt GitHub du projet PharmaGo.	37

Introduction générale

Introduction

Dans un contexte où la digitalisation touche tous les secteurs, la gestion des livraisons pharmaceutiques nécessite aujourd’hui des outils modernes, rapides et fiables. Les processus manuels utilisés dans de nombreuses pharmacies peuvent engendrer des erreurs, des retards ou un manque de visibilité sur les opérations. C'est dans cette optique que s'inscrit le projet **PharmaGo**, une application desktop visant à faciliter et automatiser la gestion interne des livraisons de médicaments au sein d'une pharmacie centrale.

Description du projet

Le projet **PharmaGo** a pour objectif de fournir une solution logicielle complète pour la gestion interne des livraisons de médicaments. Conçue pour les pharmaciens et le personnel d'une pharmacie centrale, l'application permet de suivre les livraisons, de gérer les clients, les médicaments et les utilisateurs, tout en offrant une vision globale de l'activité grâce à un tableau de bord interactif et des statistiques visuelles.

Elle se distingue par son interface intuitive, sa gestion multi-utilisateurs (administrateur et personnel), et sa capacité à générer automatiquement des tickets de livraison avec un **QR code**, assurant ainsi une meilleure traçabilité et une organisation optimisée.

Objectifs du projet

L'objectif principal de ce projet est de concevoir une solution informatique fiable et ergonomique répondant aux besoins spécifiques d'une pharmacie centrale. Plus précisément, il s'agit de :

- Automatiser la gestion des livraisons pour réduire les erreurs humaines ;
- Offrir une visibilité en temps réel sur les stocks, les clients et l'état des livraisons ;
- Simplifier la recherche et la consultation d'informations grâce à une interface claire ;
- Permettre aux administrateurs de gérer efficacement les utilisateurs et les données du système ;
- Fournir des statistiques visuelles pour soutenir la prise de décision.

Technologies utilisées

Pour le développement de l'application **PharmaGo**, les technologies suivantes ont été employées :

- **JavaFX** : pour la conception d'une interface graphique moderne et dynamique ;
- **PHPMyAdmin (WAMP)** : pour la gestion de la base de données MySQL ;
- **Architecture MVC (Model–View–Controller)** : pour une meilleure organisation du code et une séparation claire entre la logique métier, la vue et le contrôle.

Méthodologie de développement

Pour la conception de **PharmaGo**, nous avons adopté le **modèle de cycle de vie en prototypage**.

Le développement a ainsi débuté par la réalisation d'un **prototype d'interface** sur l'outil **Figma**, permettant de visualiser la structure générale de l'application, d'ajuster les choix esthétiques et de définir les interactions principales avant la phase de codage. Ce prototype a servi de base à la construction progressive de l'application finale, tout en facilitant les retours et les ajustements nécessaires au fur et à mesure de l'avancement du projet.

Afin aussi de mieux travailler en binôme et de faciliter la collaboration, tout le code source de l'application a été centralisé sur un dépôt **GitHub**. Cela a permis de partager facilement les modifications, de gérer les versions, et d'assurer une traçabilité complète de l'évolution du projet. Chaque membre a pu contribuer de manière organisée, avec la possibilité de revenir sur d'anciennes versions si nécessaire.

En somme, **PharmaGo** vise à moderniser la gestion pharmaceutique interne tout en offrant une expérience fluide, performante et adaptée aux besoins des utilisateurs. Ce rapport présentera les différentes étapes de sa conception, de l'analyse à la réalisation, jusqu'aux tests finaux.

Chapitre 1

Élaboration des besoins fonctionnels

Ce chapitre présente l'ensemble des besoins fonctionnels de l'application **Pharma-Go**. Ces besoins décrivent les principales fonctionnalités que le système doit offrir afin d'assurer une gestion fluide et efficace des opérations au sein de la pharmacie.

1.1 Page d'accueil

- L'utilisateur arrive sur la page d'accueil de l'application.
- La page affiche une image descriptive, le logo et le slogan.
- La page propose un bouton « **Commencer** » pour accéder au tableau de bord.
- La page contient également une section « **À propos** » et « **Aide** » pour guider l'utilisateur.

1.2 Authentification

- L'utilisateur saisit son email et son mot de passe pour accéder à l'application.
- Le système vérifie les informations et détermine le rôle de l'utilisateur (administrateur ou personnel de la pharmacie).
- L'utilisateur reçoit un message d'erreur en cas d'identifiants incorrects.
- Un utilisateur ayant le rôle **admin** dispose de droits supplémentaires dans l'application.

1.3 Tableau de bord (Dashboard)

- L'utilisateur arrive sur le tableau de bord après authentification.
- Le tableau de bord contient un menu vertical avec :
 - Logo

- Tableau de bord
- Consultation globale
- Gestion des clients
- Gestion des médicaments
- Gestion des livraisons
- (Pour l'administrateur) Gestion des utilisateurs
- Cette page offre un aperçu général des fonctionnalités et des statistiques de l'application.

1.4 Consultation globale

- L'utilisateur visualise les statistiques de la pharmacie sous forme de graphiques :
 - Nombre total de clients, de médicaments et de livraisons.
 - Types de livraisons les plus convoités.
 - Clients les plus actifs.
 - Médicaments les plus livrés.

1.5 Gestion des clients

Ajouter un client

- L'utilisateur saisit les informations nécessaires (nom, prénom, code client, adresse, téléphone, etc.).
- Le système vérifie les informations saisies et confirme l'ajout.

Modifier un client

- L'utilisateur peut modifier les informations d'un client existant.
- Le système enregistre les modifications et confirme la mise à jour.

Supprimer un client

- L'utilisateur peut supprimer un client.
- Le système demande une confirmation avant la suppression.

Recherche d'un client

- L'utilisateur peut rechercher un client par son code ou par son nom ou prénom.
- Le système affiche les informations correspondantes.

Affichage de la liste des clients

- L'utilisateur peut consulter la liste complète des clients.

1.6 Gestion des médicaments

Ajouter un médicament

- L'utilisateur saisit les informations nécessaires (nom, code, quantité, prix, etc.).
- Le système vérifie les informations et confirme l'ajout.

Modifier un médicament

- L'utilisateur peut modifier les informations d'un médicament existant.
- Le système enregistre les modifications et confirme la mise à jour.

Supprimer un médicament

- L'utilisateur peut supprimer un médicament.
- Le système demande une confirmation avant la suppression.

Recherche d'un médicament

- L'utilisateur peut rechercher un médicament par son nom ou son code.
- Le système affiche les informations correspondantes.

Affichage de la liste des médicaments

- L'utilisateur peut consulter la liste complète des médicaments disponibles.

1.7 Gestion des livraisons

Créer une livraison

- L'utilisateur saisit les informations nécessaires (client, date, type de livraison, médicaments à livrer, quantité, coût, etc.).
- Le système génère un ticket pour la livraison avec un **QR code**.

Ajouter des médicaments à une livraison

- L'utilisateur sélectionne les médicaments et leurs quantités.
- Le système calcule le coût total et met à jour le ticket.

Modifier ou annuler une livraison

- L'utilisateur peut modifier les informations d'une livraison existante.
- L'utilisateur peut annuler une livraison, avec confirmation du système.

Recherche d'une livraison

- L'utilisateur peut rechercher une livraison par numéro ou par client.
- Le système affiche les détails de la livraison : médicaments, client, type et coût.

Filtrage des livraisons

- L'utilisateur peut filtrer les livraisons par type ou par statut (en attente, livrée, annulée , en cours , urgente ..ect).

1.8 Gestion des utilisateurs (Admin uniquement)

Ajouter un utilisateur

- L'administrateur saisit les informations nécessaires (nom, prénom, mail, mot de passe).
- Le système vérifie et confirme l'ajout.

Modifier un utilisateur

- L'administrateur peut modifier les informations d'un utilisateur existant.
- Le système enregistre les modifications et confirme la mise à jour.

Supprimer un utilisateur

- L'administrateur peut supprimer un utilisateur, avec confirmation du système.

Chapitre 2

Règles métier et contraintes

Ce chapitre présente les différentes règles métier et contraintes à respecter dans le cadre du développement de l'application **PharmaGo**. Ces règles garantissent la cohérence, la fiabilité et la sécurité des données manipulées par le système.

2.1 Principe général

Les règles métier sont des conditions imposées par la logique fonctionnelle de l'application. Elles permettent d'assurer la validité des informations saisies et la bonne exécution des traitements internes. Les contraintes, quant à elles, sont d'ordre technique et visent à préserver l'intégrité des données au niveau de la base de données et du code applicatif.

2.2 Règles d'unicité et d'identification

- Tous les identifiants (`IdUser`, `CodeClt`, `IdMed`, `numLiv`) sont **uniques** et **générés automatiquement** (auto-incrémentés).
- Aucun doublon ne doit être présent dans les tables de la base de données.

2.3 Contraintes sur les utilisateurs et les clients

- L'adresse e-mail doit contenir le caractère @ et être conforme au format standard (ex : `nom@example.com`).
- Le mot de passe doit contenir **au minimum 10 caractères** (idéalement combiner lettres, chiffres et caractères spéciaux).
- Les noms et prénoms ne doivent contenir que des lettres (pas de chiffres ni de caractères spéciaux).
- Le numéro de téléphone doit comporter exactement **10 chiffres**.
- Les champs obligatoires (nom, prénom, e-mail, mot de passe, téléphone..ect) doivent être remplis avant validation.

2.4 Contraintes sur les médicaments

- La **quantité disponible** doit toujours être supérieure ou égale à **0**.
- Le **prix unitaire** doit toujours être supérieure à **0**.
- La **taxe** appliquées doit être supérieur à **0**.
- Le **nom du médicament** ne doit pas être vide.

2.5 Contraintes sur les livraisons

- La **date de livraison** doit être supérieure ou égale à la date actuelle.
- Chaque livraison doit être associée à un **client existant** et à au moins un médicament.
- Le **coût total** de la livraison est calculé automatiquement à partir des quantités et des prix unitaires.
- En cas d'annulation, la livraison doit être marquée comme **Annulée**, mais conservée dans l'historique pour traçabilité.
- Le **QR code** associé à chaque livraison doit être unique.

2.6 Contraintes sur les opérations et la cohérence des données

- Les opérations de suppression sont soumises à confirmation afin d'éviter toute perte accidentelle de données.
- En cas de suppression d'un client, ses livraisons doivent être conservées pour assurer la traçabilité (suppression logique).
- Les champs numériques ne doivent pas contenir de texte (et inversement).
- Les dates doivent respecter le format standard **AAAA-MM-JJ**.

2.7 Sécurité et intégrité

- Les mots de passe doivent être stockés sous forme **hachée** (non en clair) dans la base de données.
- L'accès aux fonctionnalités de gestion des utilisateurs est réservé aux utilisateurs ayant le rôle **admin**.
- Les formulaires doivent être protégés contre les entrées invalides.

2.8 Synthèse

Ces règles métier et contraintes assurent :

- la cohérence et la fiabilité des données,
- la sécurité du système et la protection des utilisateurs,
- la conformité de l'application aux bonnes pratiques de développement.

Chapitre 3

Modélisation UML

Ce chapitre présente la modélisation UML du système afin d'illustrer les interactions entre les acteurs et la structure générale des fonctionnalités.

3.1 Diagramme des cas d'utilisation

Le diagramme de cas d'utilisation, présenté à la figure 3.1, illustre les principales fonctionnalités du système et les interactions avec les différents acteurs.



FIG. 3.1: Diagramme des cas d'utilisation

3.2 Description textuelle des cas d'utilisation

3.2.1 Cas d'utilisation : S'authentifier

Identification

Nom du cas : « S'authentifier ».

But : Permettre à l'utilisateur d'accéder à l'application selon son rôle (personnel ou administrateur).

Acteur principal : Utilisateur (personnel ou administrateur).

Acteur secondaire : Néant.

Date de création : 07/11/2025.

Responsable : Équipe de développement PharmaGo (Mlle Maouche , Mlle Sahi).

Version : 1.0.

Séquencement

Le cas d'utilisation commence lorsqu'un utilisateur saisit ses identifiants de connexion.

Pré-conditions

L'utilisateur possède un compte valide enregistré dans le système.

Enchaînement nominal

1. L'utilisateur saisit son e-mail et son mot de passe.
2. Le système vérifie la validité des identifiants.
3. Si les informations sont correctes, l'utilisateur est redirigé vers le tableau.

Enchaînements alternatifs

A1 : Si les identifiants sont incorrects, le système affiche un message d'erreur et propose une nouvelle tentative.

A2 : Si l'utilisateur ne possède pas encore de compte, le système lui propose de créer un compte en saisissant son nom, prénom, adresse e-mail et mot de passe. L'adresse e-mail doit contenir le caractère « @ » et le mot de passe doit comporter au minimum 10 caractères, conformément aux contraintes définies dans le chapitre 2.

Post-conditions

L'utilisateur est authentifié et peut accéder aux fonctionnalités correspondant à son rôle.

3.2.2 Cas d'utilisation : Gérer les clients

Identification

Nom du cas : « Gérer les clients ».

But : Permettre au personnel de gérer les informations relatives aux clients.

Acteur principal : Personnel de la pharmacie.

Acteur secondaire : Néant.

Date de création : 07/11/2025.

Responsable : Équipe de développement PharmaGo (Mlle Maouche , Mlle Sahi).

Version : 1.0.

Pré-conditions

L'utilisateur doit être authentifié.

Enchaînement nominal

1. L'utilisateur sélectionne l'option « Gestion des clients » depuis le menu.
2. Le système affiche la liste des clients existants.
3. L'utilisateur peut :
 - Ajouter un nouveau client.
 - Modifier les informations d'un client.
 - Supprimer un client.
 - Rechercher un client par nom ou code.
 - Consulter les informations détaillées d'un client.

Enchaînements alternatifs

A1 : Si les informations saisies sont incomplètes, le système affiche un message d'erreur.

A2 : Si le client recherché n'existe pas, le système affiche un message « Aucun résultat ».

Post-conditions

Les informations des clients sont mises à jour dans la base de données.

3.2.3 Cas d'utilisation : Gérer les médicaments

Identification

Nom du cas : « Gérer les médicaments ».

But : Permettre au personnel de gérer le stock et les informations des médicaments.

Acteur principal : Personnel de la pharmacie.

Acteur secondaire : Néant.

Date de création : 07/11/2025.

Responsable : Équipe de développement PharmaGo (Mlle Maouche , Mlle Sahi).

Version : 1.0.

Pré-conditions

L'utilisateur doit être authentifié.

Enchaînement nominal

1. L'utilisateur accède à la section « Gestion des médicaments ».
2. Le système affiche la liste des médicaments disponibles.

3. L'utilisateur peut :

- Ajouter un nouveau médicament.
- Modifier les informations d'un médicament existant.
- Supprimer un médicament.
- Rechercher un médicament par nom ou code.
- Consulter la liste complète des médicaments.

Enchaînements alternatifs

A1 : Si la quantité saisie est négative, le système rejette la modification.

Post-conditions

Le stock et les informations des médicaments sont mis à jour.

3.2.4 Cas d'utilisation : Gérer les livraisons

Identification

Nom du cas : « Gérer les livraisons ».

But : Permettre au personnel de créer, modifier, supprimer et consulter les livraisons.

Acteur principal : Personnel de la pharmacie.

Acteur secondaire : Néant.

Date de création : 07/11/2025.

Responsable : Équipe de développement PharmaGo (Mlle Maouche , Mlle Sahi).

Version : 1.0.

Pré-conditions

L'utilisateur doit être connecté et avoir accès aux données clients et médicaments.

Enchaînement nominal

1. L'utilisateur sélectionne « Gestion des livraisons ».
2. Le système affiche la liste des livraisons existantes.
3. L'utilisateur peut :
 - Créer une nouvelle livraison.
 - Ajouter un ou plusieurs médicaments à la livraison.
 - Modifier ou annuler une livraison.
 - Consulter les détails d'une livraison (médicaments, client, coût, statut).
 - Rechercher une livraison par numéro ou client.
 - Filtrer par type ou statut.

Enchaînements alternatifs

A1 : Si la date de livraison saisie est antérieure à la date actuelle, le système affiche une erreur.

A2 : Si le client n'existe pas, le système empêche la création de la livraison.

Post-conditions

Les informations sur la livraison sont enregistrées et un QR code est généré.

3.2.5 Cas d'utilisation : Gérer les utilisateurs

Identification

Nom du cas : « Gérer les utilisateurs ».

But : Permettre à l'administrateur de gérer les comptes du personnel.

Acteur principal : Administrateur.

Acteur secondaire : Néant.

Date de création : 07/11/2025.

Responsable : Équipe de développement PharmaGo (Mlle Maouche , Mlle Sahi).

Version : 1.0.

Pré-conditions

L'administrateur doit être connecté au système.

Enchaînement nominal

1. L'administrateur accède à la section « Gestion des utilisateurs ».
2. Le système affiche la liste des comptes existants.
3. L'administrateur peut :
 - Ajouter un nouvel utilisateur.
 - Modifier les informations d'un utilisateur.
 - Supprimer un utilisateur.
 - Rechercher un utilisateur par nom ou identifiant.
 - Consulter les informations d'un utilisateur.

Post-conditions

Les informations des utilisateurs sont mises à jour dans la base de données.

3.2.6 Cas d'utilisation : Consulter les statistiques

Identification

Nom du cas : « Consulter les statistiques ».

But : Permettre à l'utilisateur de visualiser des statistiques globales sur les livraisons, les

clients et les médicaments.

Acteur principal : Personnel de la pharmacie.

Acteur secondaire : Néant.

Date de création : 07/11/2025.

Responsable : Équipe de développement PharmaGo (Mlle Maouche , Mlle Sahi).

Version : 1.0.

Pré-conditions

L'utilisateur doit être connecté.

Enchaînement nominal

1. L'utilisateur accède à la page tableau de bord ou consultation globale.
2. Le système affiche les statistiques :
 - Nombre de livraisons effectuées par jour ou mois.
 - Médicaments les plus livrés.
 - Clients les plus fréquents.
3. Les statistiques sont représentées sous forme de graphiques.

Post-conditions

Les statistiques sont consultées sans modification de données.

3.3 Diagrammes de séquence

Dans le cadre de notre modélisation, nous avons choisi de représenter deux cas d'utilisation : *S'authentifier* et *Gérer les livraisons*, étant donné que le but principal de l'application est la gestion des livraisons pharmaceutiques. C'est dans cette optique que nous présentons deux diagrammes de séquence pour illustrer chacun de ces cas.

Pour ces diagrammes, nous utilisons l'acteur générique *Utilisateur*, regroupant à la fois le personnel de la pharmacie et l'administrateur, puisque les deux peuvent effectuer les opérations liées à la gestion des livraisons.

Dans le diagramme de séquence de *Gérer les livraisons*, nous avons utilisé un **loop** pour représenter la répétition des tâches de gestion. Cela signifie que l'utilisateur peut effectuer autant de fois que nécessaire les opérations d'ajout, modification, suppression ou recherche de livraisons. Le **loop** permet donc d'indiquer visuellement dans le diagramme que ces actions sont répétitives et ne sont pas limitées à une seule occurrence.

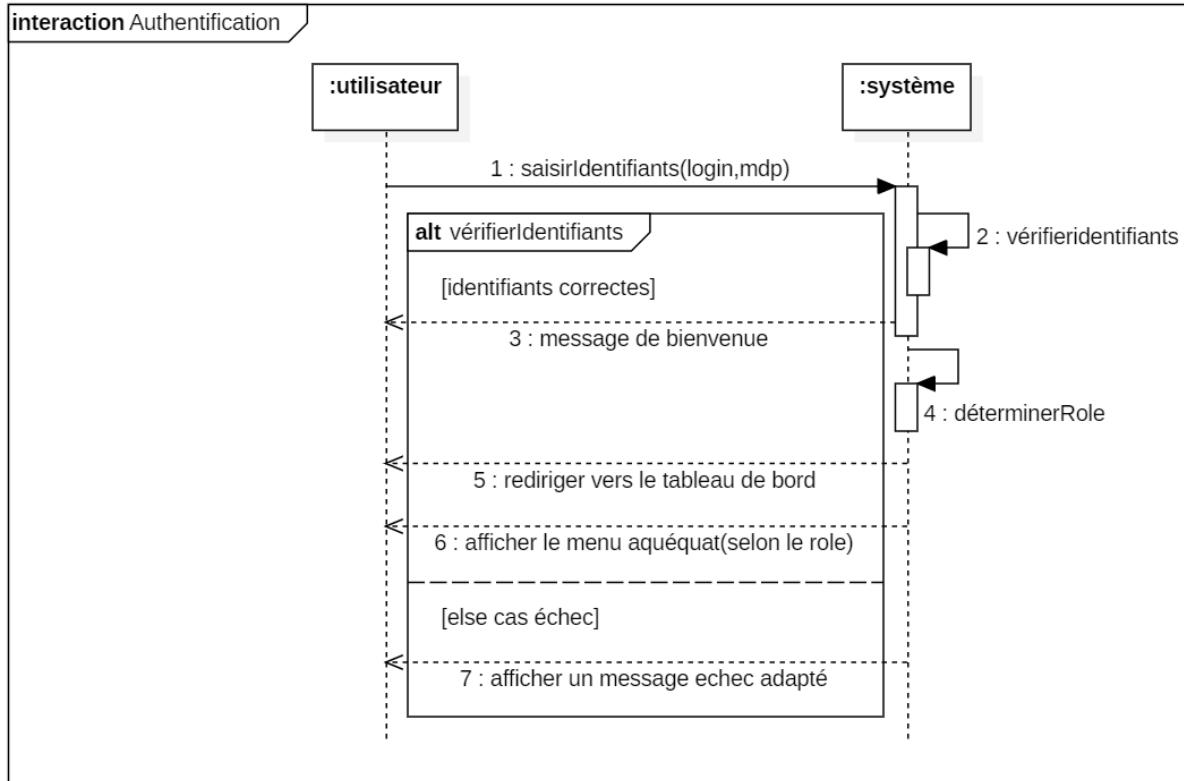


FIG. 3.2: Diagramme de séquence pour le cas d'utilisation « S'authentifier »

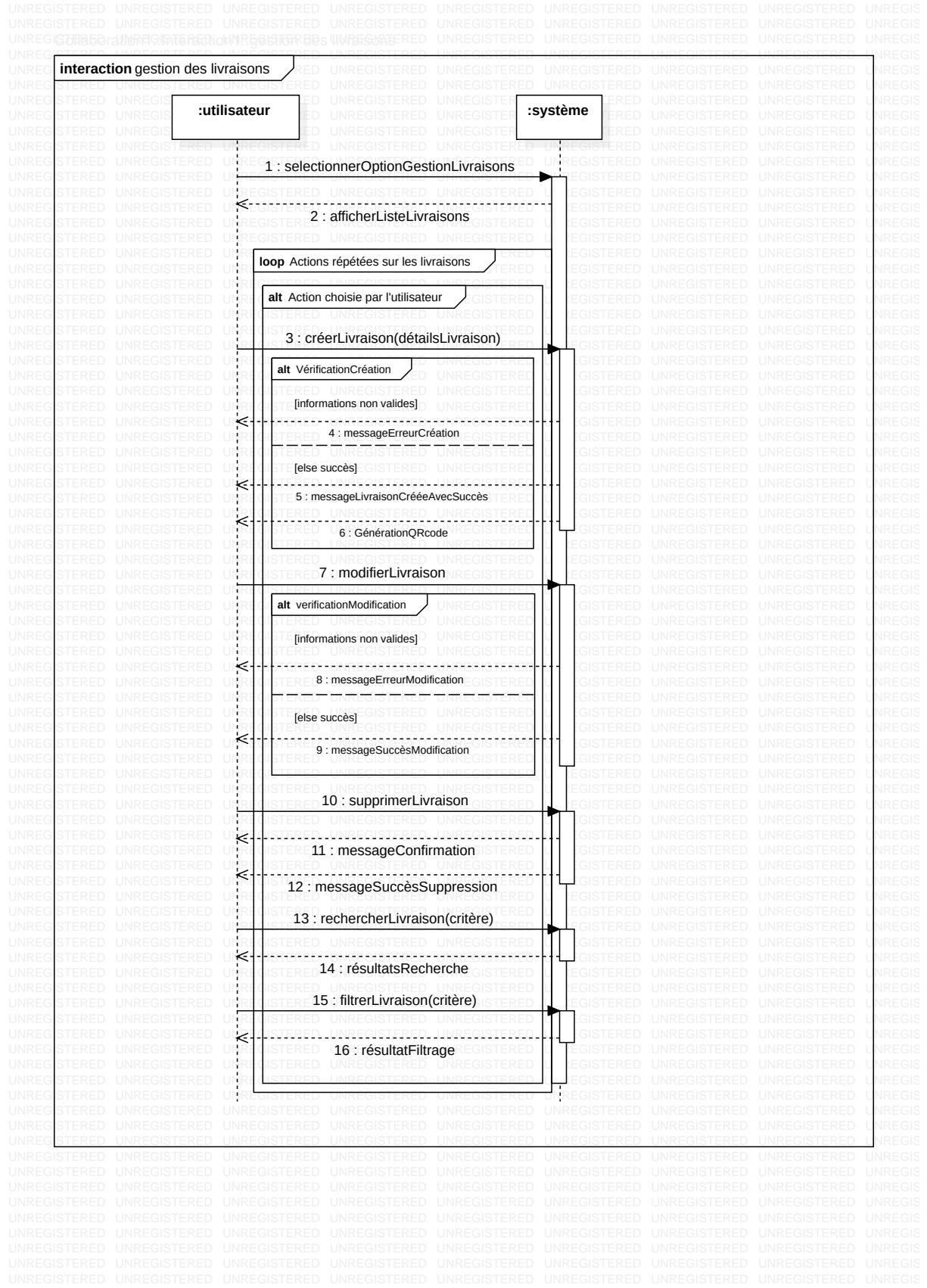


FIG. 3.3: Diagramme de séquence pour le cas d'utilisation « Gérer les livraisons »

3.4 Diagramme de classes

Le diagramme de classes présenté à la figure 3.4 illustre la structure du système, les classes principales et leurs relations.

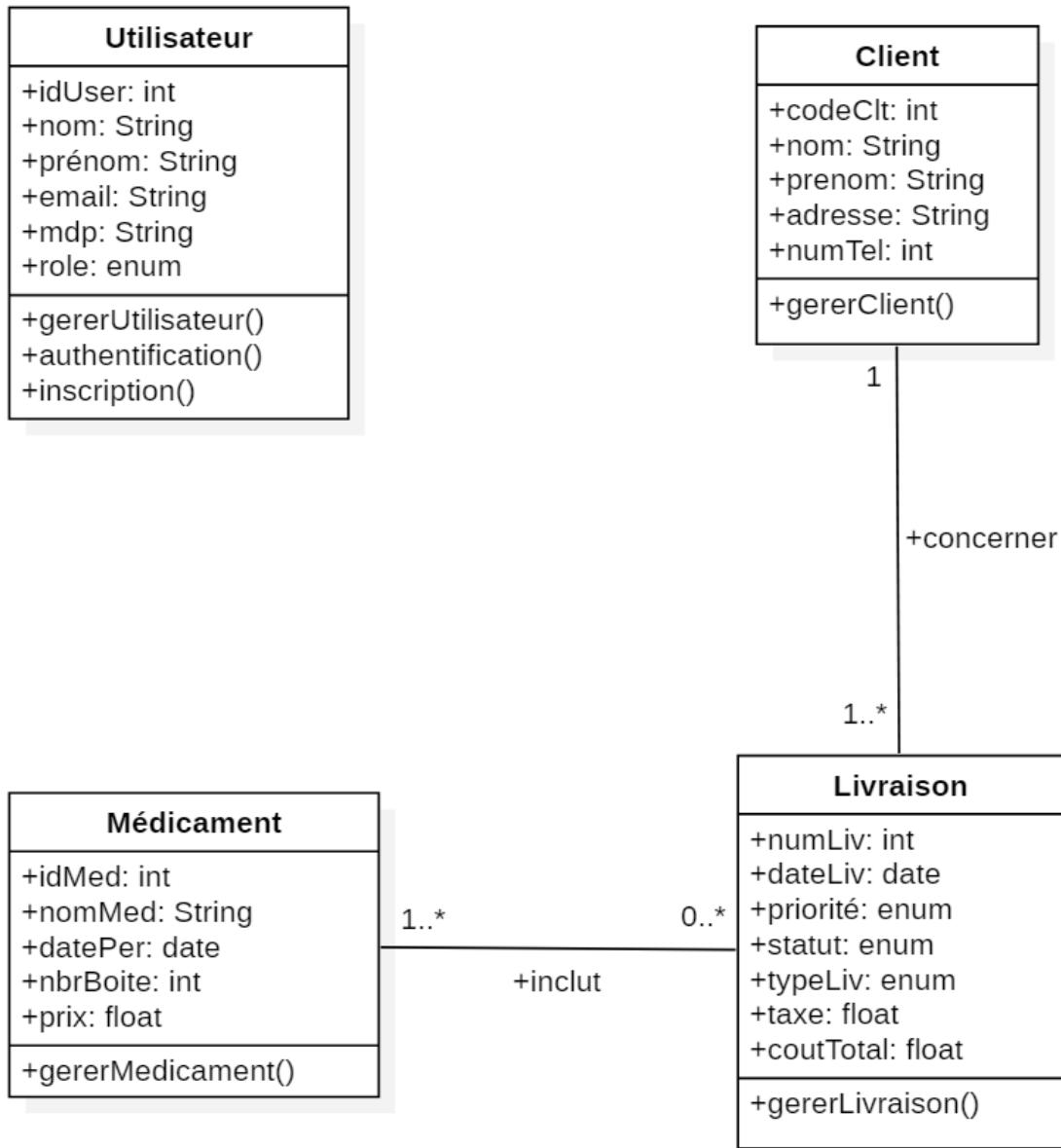


FIG. 3.4: Diagramme de classes

Chapitre 4

Modèle Relationnel

Ce chapitre présente le modèle relationnel du système, qui traduit les classes et associations du diagramme de classes en tables de base de données. Chaque classe devient une relation, et les associations sont transformées selon des règles de passage du diagramme de classes vers le modèle relationnel.

4.1 Tables principales du système

À partir des classes du diagramme de classes et des associations, nous obtenons le modèle relationnel suivant :

- Client(codeClt, nom, prenom, adresse, numTel)
- Livraison(numLiv, dateLiv, priorité, statut, taxe, typeLiv, #codeClt)
(Règle 1 : multiplicité 1-* entre Client et Livraison)
- Medicament(idMed, nomMed, datePer, nbrBoite, prixMed)
- Inclure(#idMed, #numLiv)
(Règle 2 : multiplicité *-* entre Livraison et Medicament)
- Utilisateur(idUser, nom, prenom, role, mdp, mail)

4.2 Remarques sur le modèle relationnel

Ce modèle relationnel permet de :

- Garantir la cohérence des données via les clés primaires et étrangères.
- Reproduire les relations définies dans le diagramme de classes.
- Constituer la base pour la création des tables dans le SGBD.

Chapitre 5

Script SQL

Ce chapitre présente le script SQL utilisé pour la création de la base de données **PHARMAGO**, la définition de ses tables et contraintes, ainsi que quelques insertions d'exemple illustrant la structure de la base.

5.1 Crédation de la base et des tables

```
1 CREATE DATABASE PHARMAGO;
2
3 CREATE TABLE Client (
4     codeClt INT AUTO_INCREMENT PRIMARY KEY,
5     nom VARCHAR(20),
6     prenom VARCHAR(20),
7     adresse VARCHAR(30),
8     numTel CHAR(10) CHECK (numTel REGEXP '^[0-9]{10}$')
9 );
10
11 CREATE TABLE Livraison (
12     numLiv INT AUTO_INCREMENT PRIMARY KEY,
13     codeClt INT,
14     dateLiv DATE,
15     priorite ENUM('urgent','normal'),
16     statut ENUM('en_attente','livrée','annulée','en_cours'),
17     type_liv ENUM('sous_chaine_du_froid','sous_congélation','
18         dangereuse','normale'),
19     taxe FLOAT CHECK (taxe >= 0),
20     cout FLOAT CHECK (cout >= 0),
21     FOREIGN KEY (codeClt) REFERENCES Client(codeClt)
22 );
23
24 CREATE TABLE Medicament (
25     idMed INT AUTO_INCREMENT PRIMARY KEY,
26     nomMed VARCHAR(20),
27     datePer DATE,
28     nbrBoite INT CHECK (nbrBoite >= 0),
29     prixMed INT CHECK (prixMed >= 0)
30 );
```

```

31 CREATE TABLE Inclure (
32     idMed INT,
33     numLiv INT,
34     PRIMARY KEY (idMed, numLiv),
35     FOREIGN KEY (idMed) REFERENCES Medicament(idMed),
36     FOREIGN KEY (numLiv) REFERENCES Livraison(numLiv)
37 );
38
39 CREATE TABLE Utilisateur (
40     idUser INT AUTO_INCREMENT PRIMARY KEY,
41     nom VARCHAR(20),
42     prenom VARCHAR(20),
43     role ENUM('personnel','admin') DEFAULT 'personnel',
44     mdp VARCHAR(255),
45     mail VARCHAR(100)
46 );

```

Listing 5.1: Création de la base de données et des tables

5.2 Insertion de données d'exemple

Seules quelques insertions d'exemple sont présentées ici pour illustrer la structure de la base.

```

1 -- Clients
2 INSERT INTO Client(nom, prenom, adresse, numTel) VALUES ("Sahi", "Imene", "Sidi\u00e2Ahmed", "0793714172");
3 INSERT INTO Client(nom, prenom, adresse, numTel) VALUES ("Salimi", "Amina", "Cartier\u00e2Sghir", "0786545445");
4 INSERT INTO Client(nom, prenom, adresse, numTel) VALUES ("Tabet", "Bilal", "Tala\u00e2Hamza", "0786545444");
5 INSERT INTO Client(nom, prenom, adresse, numTel) VALUES ("Maouche", "Lina", "C\u00f4t\u00e9\u00e2Aouchiche", "0784453246");
6 INSERT INTO Client(nom, prenom, adresse, numTel) VALUES ("Saïdi", "Fateh", "Edimco", "0567345340");
7
8 -- Livraisons
9 INSERT INTO Livraison (codeClt,dateLiv,priorite,statut,type_liv,taxe
10 ,cout)
11 VALUES (3,'2026-03-03','urgent','en_attente','normale',250,2250);
12 INSERT INTO Livraison (codeClt,dateLiv,priorite,statut,type_liv,taxe
13 ,cout)
14 VALUES (4,'2025-11-01','normal','annul\u00e9e','sous_chaine_du_froid'
15 ,650,1500);
16 INSERT INTO Livraison (codeClt,dateLiv,priorite,statut,type_liv,taxe
17 ,cout)
18 VALUES (5,'2026-01-04','urgent','en_attente','dangereuse',200,1750);
19 INSERT INTO Livraison (codeClt,dateLiv,priorite,statut,type_liv,taxe
20 ,cout)
21 VALUES (6,'2026-01-12','urgent','en_cours','normale',450,3330);
22 INSERT INTO Livraison (codeClt,dateLiv,priorite,statut,type_liv,taxe
23 ,cout)
24 VALUES (7,'2026-02-08','normal','livr\u00e9e','sous_cong\u00e9lation',150,240)

```

```

19 ;
20
-- Médicaments
21 INSERT INTO Medicament (nomMed,datePer,nbrBoite,prixMed) VALUES ("Paracétamol_500g",'2028-07-06',15,150);
22 INSERT INTO Medicament (nomMed,datePer,nbrBoite,prixMed) VALUES ("Ibuprofène",'2023-02-06',27,750);
23 INSERT INTO Medicament (nomMed,datePer,nbrBoite,prixMed) VALUES ("Colchimax_5mg",'2029-09-04',9,800);
24 INSERT INTO Medicament (nomMed,datePer,nbrBoite,prixMed) VALUES ("Clamoxyl_500mg",'2026-07-22',0,1150);
25 INSERT INTO Medicament (nomMed,datePer,nbrBoite,prixMed) VALUES ("Levothyrox_150",'2027-01-05',22,950);

26
-- Association Inclure
27 INSERT INTO Inclure VALUES (1,6);
28 INSERT INTO Inclure VALUES (3,8);
29 INSERT INTO Inclure VALUES (5,10);
30 INSERT INTO Inclure VALUES (8,12);
31 INSERT INTO Inclure VALUES (9,14);

32
-- Utilisateurs
33 INSERT INTO Utilisateur (nom,prenom,role,mdp,mail)
34 VALUES ('Sahi','Nour_EL_Imene','admin','123456',
35 'nourelimenesahi@gmail.com');
36 INSERT INTO Utilisateur (nom,prenom,role,mdp,mail)
37 VALUES ('Maouche','Lina','admin','123456','linamaouche@gmail.com');
38 INSERT INTO Utilisateur (nom,prenom,role,mdp,mail)
39 VALUES ('Salmi','Warda','personnel','123456','leswinxthebest@gmail.
40 com');
41 INSERT INTO Utilisateur (nom,prenom,role,mdp,mail)
42 VALUES ('Amer','Fahem','personnel','123456','fahemfahem123@gmail.com
43 ');
44 INSERT INTO Utilisateur (nom,prenom,role,mdp,mail)
VALUES ('Debbag','Djamel','personnel','123456','djimmyy76@hotmail.fr
');
```

Listing 5.2: Insertions d'exemple dans les différentes tables

5.3 Affichage formaté des identifiants

Pour un affichage plus lisible des identifiants, nous avons choisis de les présenter avec un préfixe spécifique suivi d'un numéro à trois chiffres. Par exemple, pour les clients :

```

1 SELECT CONCAT('C', LPAD(codeClt, 3, '0')) AS codeClt, nom, prenom,
2 numTel
FROM Client;
```

Listing 5.3: Affichage formaté des identifiants

De la même manière, on peut utiliser :

- 'C' pour les clients,

- 'L' pour les livraisons,
- 'M' pour les médicaments,
- 'U' pour les utilisateurs.

Chapitre 6

Interfaces de l'application

Ce chapitre présente les différentes interfaces graphiques de l'application **PHARMA-GO** à travers des captures d'écran, illustrant les fonctionnalités principales accessibles aux utilisateurs et aux administrateurs.

6.1 Page d'accueil



FIG. 6.1: Page d'accueil avec navigation principale.

6.2 Fonctionnalités supplémentaires



FIG. 6.2: Exemples de fonctionnalités accessibles depuis le tableau de bord.

6.3 Authentification

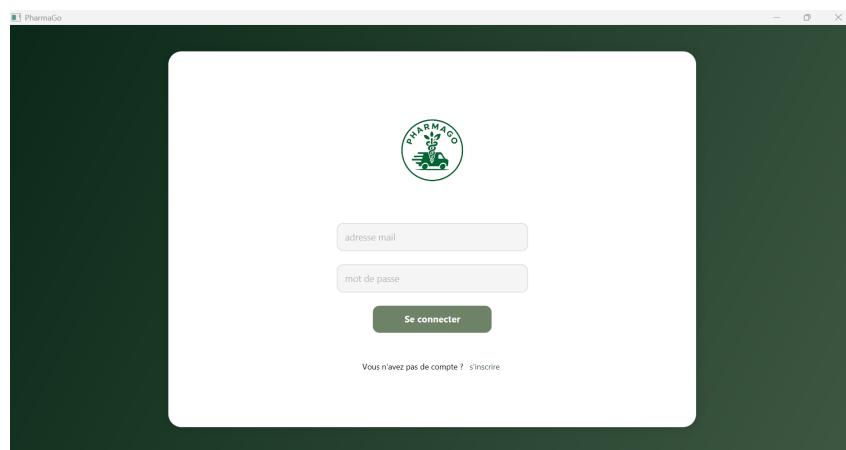


FIG. 6.3: Page de connexion à l'application.

6.4 Inscription d'un utilisateur

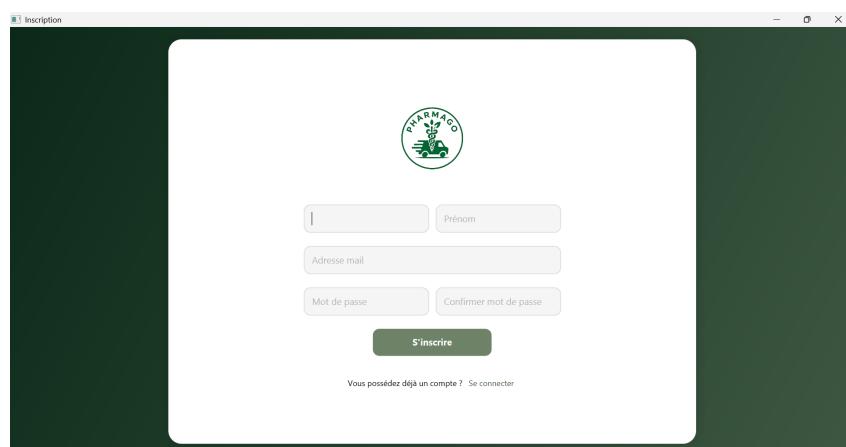


FIG. 6.4: Formulaire d'inscription pour un nouvel utilisateur.

6.5 Tableau de bord

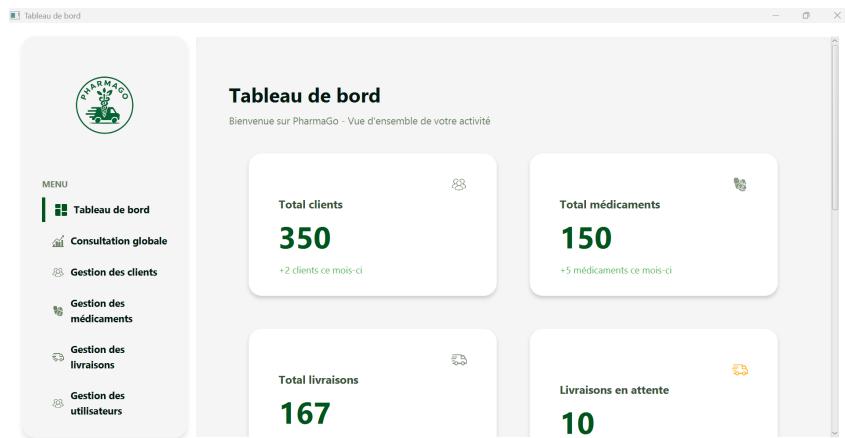


FIG. 6.5: Tableau de bord montrant les statistiques et indicateurs clés.

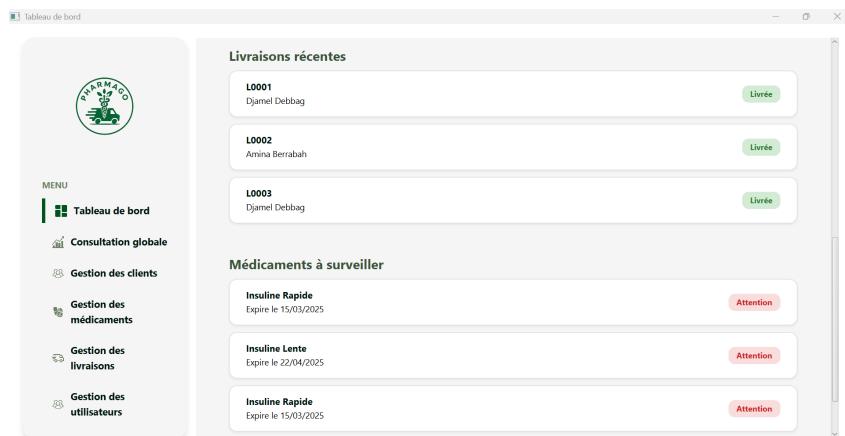


FIG. 6.6: Autre vue du tableau de bord.

6.6 Consultation globale

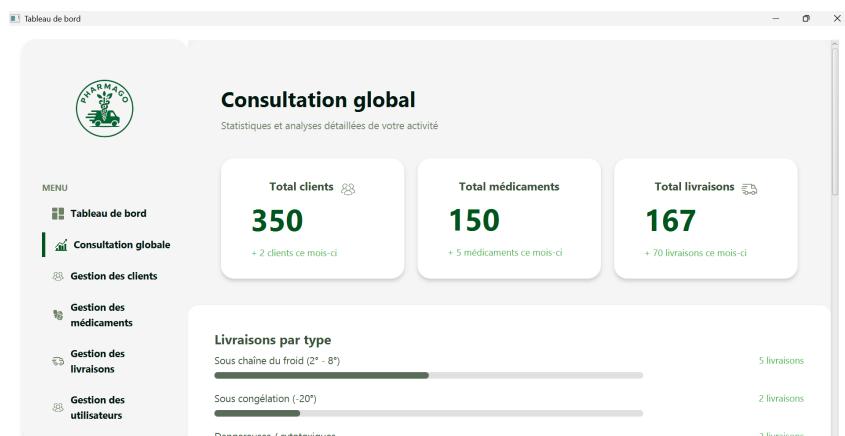


FIG. 6.7: Vue globale des livraisons et des stocks.

6.7 Gestion des clients

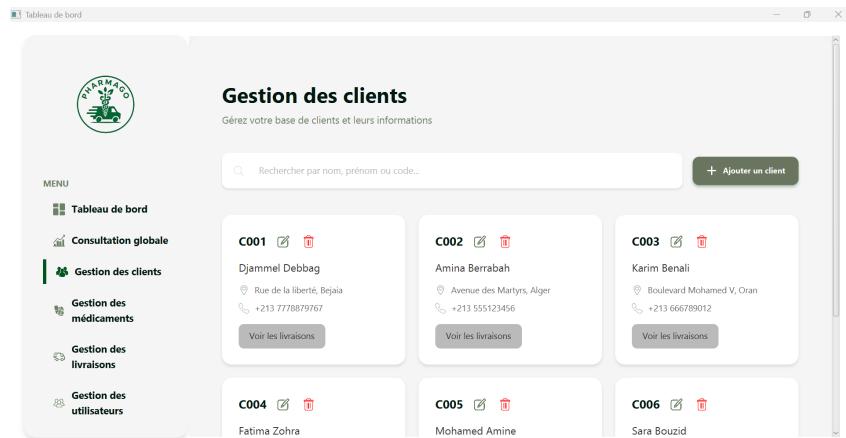


FIG. 6.8: Interface de gestion des clients.

6.8 Gestion des médicaments

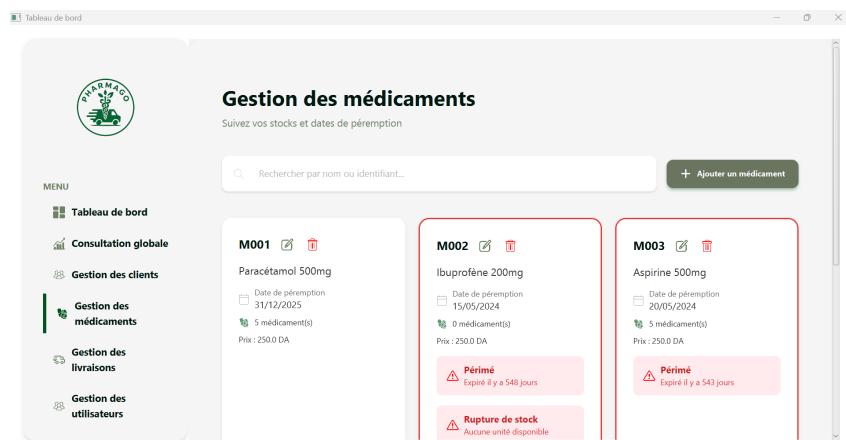


FIG. 6.9: Interface de gestion des médicaments.

6.9 Gestion des livraisons

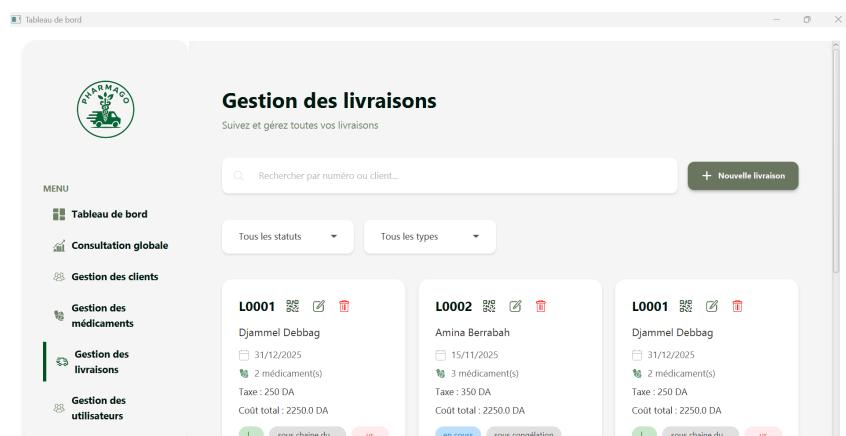


FIG. 6.10: Interface de gestion des livraisons.

6.10 QR Code pour la livraison

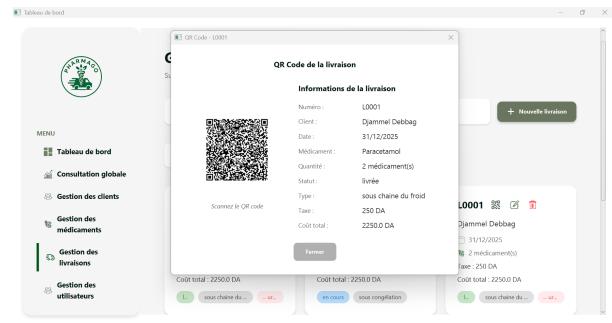


FIG. 6.11: QR Code généré pour une livraison spécifique.

6.11 Ajouter une livraison

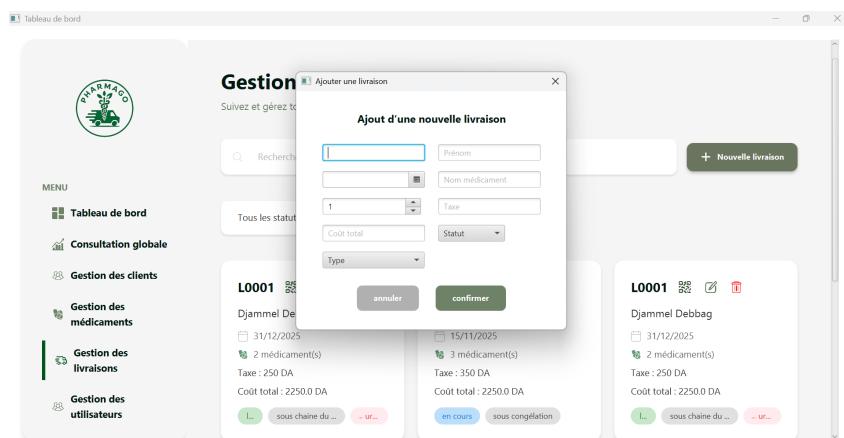


FIG. 6.12: Interface pour ajouter une nouvelle livraison.

6.12 Suppression d'une livraison

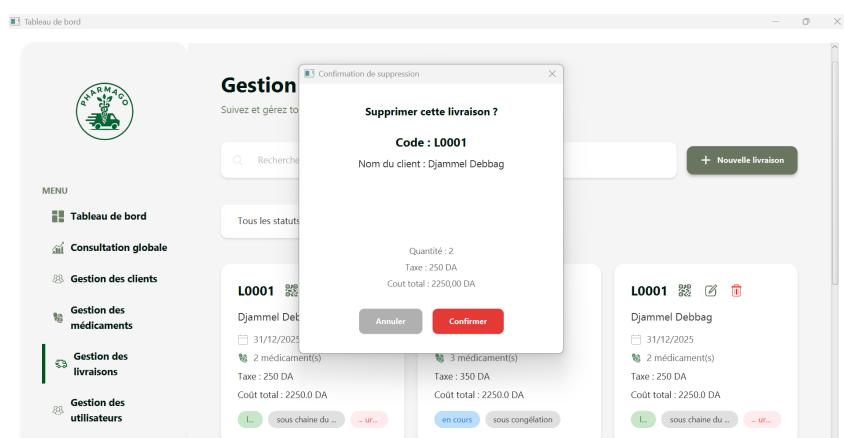


FIG. 6.13: Interface pour supprimer une livraison.

6.13 Gestion des utilisateurs (admin)

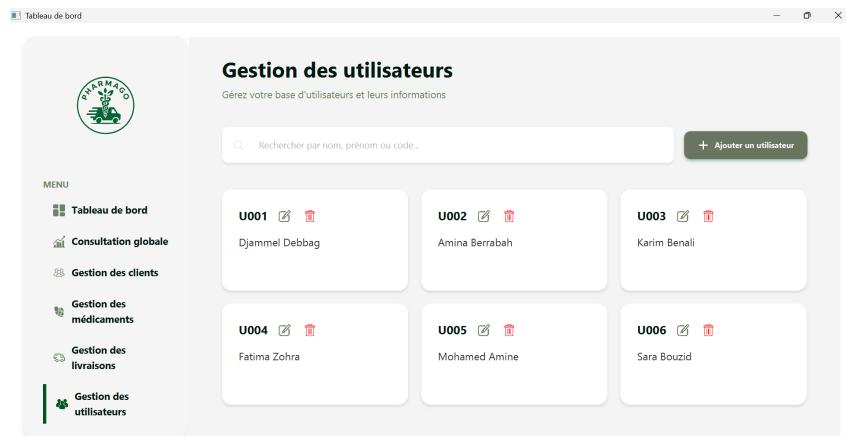


FIG. 6.14: Interface de gestion des utilisateurs

Conclusion générale

Le projet **PharmaGo** a été conçu dans le but de moderniser et d'automatiser la gestion des livraisons au sein d'une pharmacie centrale. À travers le développement de cette application, nous avons pu répondre à un besoin réel de digitalisation, en offrant une solution fiable, intuitive et adaptée aux différents profils d'utilisateurs, qu'ils soient administrateurs ou personnel de la pharmacie.

Bilan du projet

Le bilan de ce projet est globalement positif. PharmaGo permet désormais :

- une gestion efficace et sécurisée des livraisons, avec génération automatique de tickets et QR codes pour une meilleure traçabilité ;
- le suivi et la consultation des clients, médicaments et livraisons de manière centralisée ;
- une visualisation claire des statistiques et indicateurs clés grâce au tableau de bord interactif ;
- une distinction claire des rôles utilisateurs, offrant des droits spécifiques aux administrateurs ;
- une interface moderne et ergonomique, facilitant l'utilisation quotidienne par le personnel.

Compétences développées

La réalisation de ce projet nous a permis de développer et renforcer plusieurs compétences, tant techniques qu'organisationnelles :

- **Techniques** : maîtrise de JavaFX pour le développement d'interfaces graphiques, gestion de bases de données MySQL avec PHPMyAdmin, mise en place de l'architecture MVC, et utilisation des outils de versioning via GitHub ;
- **Méthodologiques** : application du cycle de vie en prototypage, capacité à analyser les besoins utilisateurs, modélisation UML, et rédaction d'un rapport structuré ;

- **Collaboration** : travail en binôme avec gestion de version et coordination sur le dépôt GitHub, échanges réguliers pour résoudre les problèmes et ajuster le prototype ;
- **Gestion de projet** : planification des tâches, suivi de l'avancement, respect des contraintes et des délais.

Perspectives d'évolution

Bien que PharmaGo soit fonctionnel et opérationnel, plusieurs pistes d'évolution peuvent être envisagées pour enrichir l'application et répondre à de futurs besoins :

- intégration d'un module de notification pour informer les utilisateurs des livraisons en temps réel ;
- mise en place d'un système de rapports exportables (PDF ou Excel) pour faciliter le suivi et l'analyse des données ;
- évolution vers une application web ou mobile pour offrir un accès distant sécurisé aux informations ;
- développement d'une interface client permettant de passer des commandes directement depuis l'application, offrant ainsi une interaction directe entre les clients et la pharmacie ;
- amélioration continue de l'interface utilisateur et ajout de fonctionnalités basées sur les retours des utilisateurs.

En conclusion, le projet PharmaGo a non seulement permis de répondre à un besoin concret de la pharmacie centrale, mais a également constitué une expérience formatrice, tant sur le plan technique que sur le plan méthodologique et collaboratif. Les perspectives d'évolution proposées ouvrent la voie à un développement futur permettant d'améliorer encore l'efficacité et l'ergonomie de l'application.

Accès au code source

Si vous souhaitez consulter le code source de l'application et découvrir son organisation interne, nous vous invitons à scanner le QR code ci-dessous, qui redirige vers le dépôt GitHub du projet :



FIG. 6.15: QR code du dépôt GitHub du projet PharmaGo.