

Table des matières

INTRODUCTION	2
DESCRIPTION DU PROJET	3
L'ORGANISATION CLIENTE	3
LE PRESTATAIRE INFORMATIQUE	3
L'EXPRESSION DE BESOIN	3
CALENDRIER DE REALISATION	4
CONCEPTION	4
DIAGRAMME DES CLASSES UML	4
BASE DE DONNEES	5
DEVELOPPEMENT	6
OUTILS ET LANGUAGES UTILISÉS	6
LES PRODUCTIONS REALISÉS	7
LES SCHEMAS EXPLICATIFS	7
MODALITES D'ACCES	14
DEPLOIEMENT EN LOCAL	14
HEBERGEMENT DE LA BASE DE DONNEES EN LIGNE	15
CONCLUSION	16

INTRODUCTION

Dans le cadre de notre BTS SIO option SLAM, nous participons à un projet professionnel majeur, PHARMAGEST, qui mobilise l'ensemble des compétences développées au cours de notre formation. Inspiré d'un contexte réel et conçu pour simuler une commande d'entreprise, ce projet constitue une opportunité unique d'appliquer nos savoir-faire en conception et développement d'applications.

Dans ce cadre, nous intervenons pour le compte de la pharmacie PHARMAGEST, située à Port-Louis, qui a sollicité un prestataire informatique externe pour la création d'une application de gestion de stock, de ventes et de finances. Notre mission est de répondre aux besoins exprimés dans le cahier des charges en respectant les normes professionnelles et les contraintes spécifiques du domaine pharmaceutique.

Ce projet nous plonge dans une situation professionnelle exigeante, où nous devons mobiliser nos compétences en développement logiciel, gestion de bases de données et conduite de projet, afin de proposer une solution efficace et conforme aux attentes du client.

Dans ce rapport, nous présenterons le contexte du projet, ses objectifs et ses enjeux. Nous mettrons également en avant le travail réalisé ainsi que nos principales réalisations au cours de ce projet.

DESCRIPTION DU PROJET

1. L'ORGANISATION CLIENTE :

PHARMAGEST est une pharmacie implantée à Port-Louis, sur l'île Maurice, depuis 2020. Elle est spécialisée dans la vente de produits pharmaceutiques destinés à un large public. Actuellement, PHARMAGEST utilise un système de gestion traditionnel, principalement manuel, pour ses activités de vente et d'approvisionnement. Face à l'augmentation du volume de ses opérations et aux défis liés à la gestion efficace de son stock, l'entreprise a décidé de moderniser son fonctionnement en investissant dans une solution informatique adaptée.

2. LE PRESTATAIRE INFORMATIQUE :

PHARMAGEST a confié son projet d'informatisation à une société de services du numérique (SSII), spécialisée dans le développement de solutions adaptées aux besoins des entreprises. Ce prestataire informatique a pour mission d'analyser les besoins de la pharmacie, de proposer une solution ergonomique et efficace, et de réaliser une application desktop conforme aux attentes. Il intervient sur l'ensemble du processus de développement, de l'étude des besoins jusqu'à la livraison de l'application. Maîtrisant des technologies telles que Java, JavaFX pour l'interface graphique, et PostgreSQL pour la gestion de bases de données, ce prestataire s'engage à fournir une solution fiable, performante et respectueuse des contraintes légales et métiers propres au secteur pharmaceutique.

3. L'EXPRESSION DE BESOIN :

Le prestataire informatique a pour mission de concevoir et de développer l'application desktop "PHARMAGEST" en conformité avec les exigences et les spécifications définies par la pharmacie PHARMAGEST.

Ses missions incluent le développement de l'application en utilisant les technologies requises, à savoir Java/JavaFX pour l'interface et PostgreSQL pour la base de données.

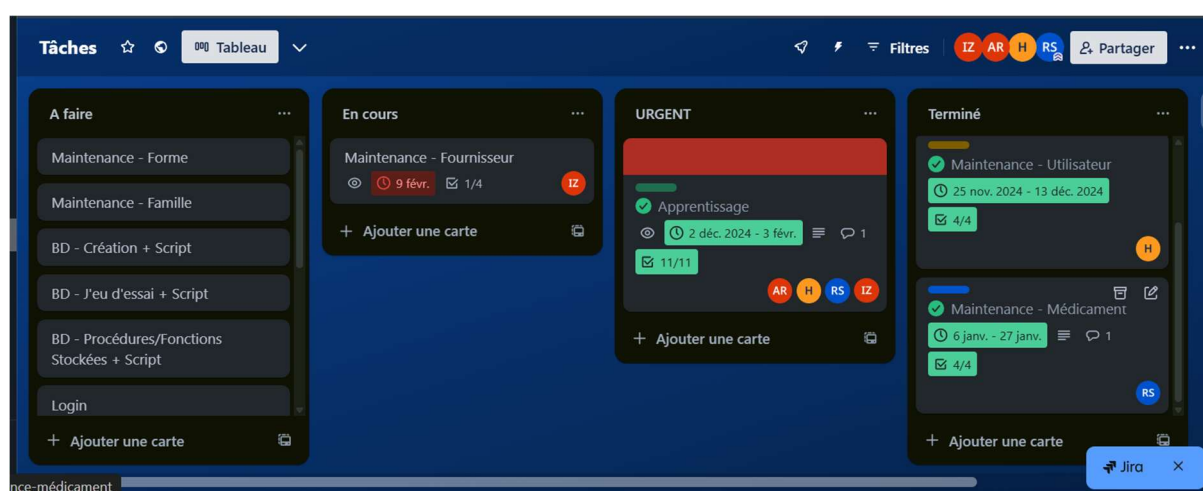
Le prestataire est également chargé de créer une interface utilisateur conviviale permettant au personnel de PHARMAGEST de gérer efficacement les stocks, les ventes, les finances et l'approvisionnement automatique.

Après le développement, le prestataire doit livrer une solution fonctionnelle, stable et conforme aux attentes, tout en assurant la compatibilité avec les exigences techniques fixées et en respectant les normes de sécurité et de fiabilité attendues dans le secteur pharmaceutique.

CALENDRIER DE REALISATION

Pour une gestion optimale du projet PHARMAGEST, nous avons établi :

Un tableau Trello qui est utilisé pour un suivi collaboratif et dynamique en ligne. Les tâches sont organisées en colonnes À faire, En cours, Urgent et Terminé, facilitant le suivi en temps réel.



CONCEPTION

1. DIAGRAMME DE CLASSES UML

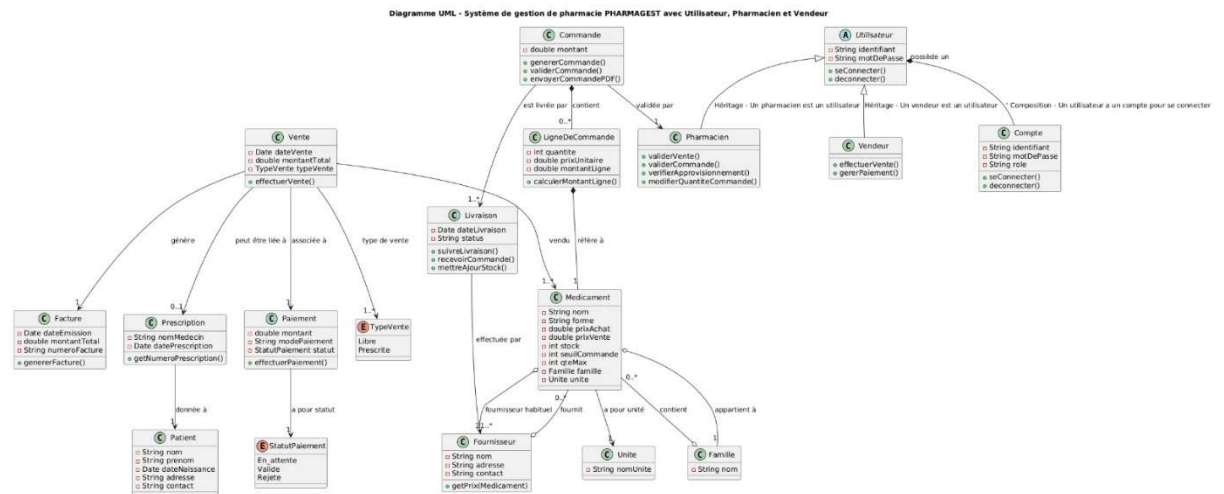
Le diagramme de classes UML du projet PHARMAGEST a pour objectif de modéliser les principales classes du système et de définir les relations entre elles.

Il représente la structure du système de gestion de la pharmacie de manière orientée objet, en identifiant les classes principales telles que **Utilisateur**, **Pharmacien**, **Vendeur**, **Médicament**, **Commande**, **LigneDeCommande**, **Livraison**, **Vente**, **Paiement**, **Fournisseur**, **Famille**, etc .

Chaque classe possède ses propres attributs et méthodes. Par exemple, la classe **Médicament** comprend des attributs tels que **nom**, **forme**, **prix d'achat**, **prix de vente**, **stock disponible**, **seuil de commande** et **quantité maximale**, et est associée à une **famille**.

Les relations entre les classes permettent de modéliser les processus métier clés, notamment la vente de médicaments, la gestion des stocks, l'approvisionnement auprès des fournisseurs .

Ce diagramme UML constitue une base essentielle pour la conception technique et assure une vision cohérente du fonctionnement global du système PHARMAGEST.



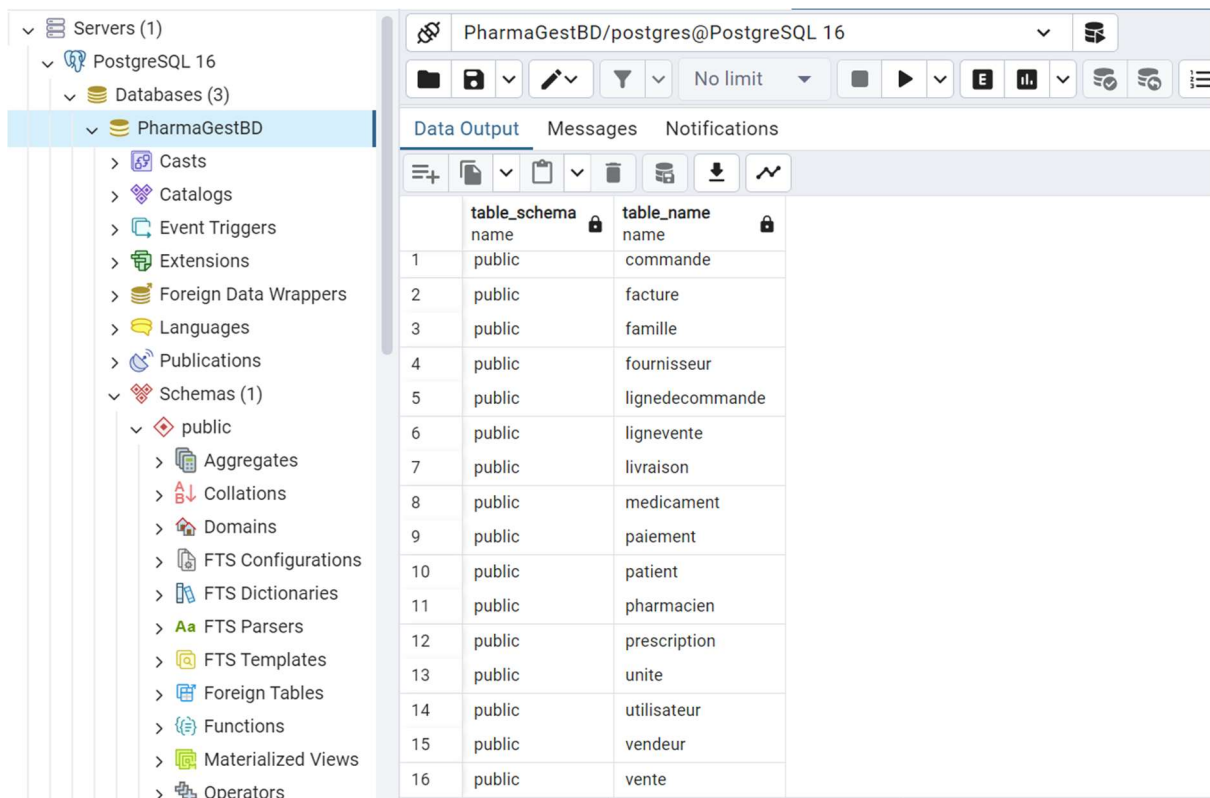
2. BASE DE DONNEE

La base de données est un élément central du projet PharmaGest, car elle permet de stocker et de gérer toutes les informations nécessaires au bon fonctionnement de l'application. Nous avons opté pour PostgreSQL en tant que système de gestion de base de données en raison de sa robustesse et de sa flexibilité.

La base de données est structurée autour de plusieurs tables clés qui représentent les différentes entités de notre application. Voici un aperçu des principales tables :

- **Table médicament** : Contient des informations sur les médicaments disponibles, incluant des détails tels que leur nom, leur prix et leur stock.
- **Table patient** : Enregistre les données des patients, incluant le nom, les coordonnées et des informations médicales pertinentes.
- **Table commande** : Gère les commandes effectuées, en stockant des informations sur les médicaments commandés, les quantités et les dates.
- **Table vente** : Suit les ventes réalisées, en liant les informations sur les patients, les pharmaciens et les médicaments vendus.
- **Table utilisateur** : Stocke les informations des utilisateurs ayant accès à l'application, incluant leur rôle et leurs identifiants.

Ci-dessous, nous pouvons visualiser les tables complètes de notre base de données :



DEVELOPPEMENT

1. OUTILS ET LANGAGES UTILISÉS

Java : Langage de programmation principal utilisé pour développer l'application desktop PharmaGest. Java offre robustesse, portabilité et un riche écosystème de bibliothèques adaptées au développement d'applications professionnelles.

JavaFX : Framework utilisé pour créer l'interface graphique (GUI) de l'application. JavaFX permet de concevoir des interfaces modernes, ergonomiques et interactives adaptées à un environnement de bureau.

Scene Builder : Outil graphique utilisé pour construire les interfaces utilisateur en JavaFX sans avoir besoin de coder manuellement les fichiers FXML. Il permet de créer des fenêtres et des éléments d'interface de manière visuelle et intuitive.

PostgreSQL : Système de gestion de base de données relationnelles utilisé pour gérer toutes les données de l'application. PostgreSQL est reconnu pour sa stabilité, sa puissance et ses fonctionnalités avancées, parfaitement adaptées aux besoins de PharmaGest.

pgAdmin : Outil graphique d'administration pour PostgreSQL. Utilisé pour créer, modifier et administrer la base de données PharmaGest de manière intuitive.

IntelliJ IDEA : Éditeur de code utilisé pour le développement Java. Il offre un environnement complet avec de nombreuses extensions facilitant le codage, le débogage et la gestion de projets.

GitHub : Outils utilisés pour la gestion du code source et la collaboration entre les membres de l'équipe, permettant un suivi des modifications et une meilleure organisation du travail.

2. LES PRODUCTIONS REALISÉS

Dans le cadre du développement du projet PharmaGest, plusieurs modules fonctionnels ont été réalisés pour couvrir l'ensemble des besoins de la pharmacie. Voici un aperçu des principales fonctionnalités mises en place :

Module Maintenance : Ce module permet d'assurer la gestion des informations de base de l'application, telles que la gestion des médicaments et des utilisateurs. Il garantit la bonne organisation des données essentielles au fonctionnement de la pharmacie.

Module Vente : Ce module permet d'enregistrer les ventes de médicaments au comptoir, avec ou sans ordonnance. Il intègre la gestion automatique des prescriptions médicales et génère un reçu pour chaque transaction, conformément aux obligations légales.

Module Caisse : Il permet de finaliser les transactions en réceptionnant les paiements. Le caissier valide chaque vente après paiement.

Module Fournisseur : Ce module assure la gestion des fournisseurs de médicaments..

Module Approvisionnement : Ce module gère tout le processus d'approvisionnement en médicaments. Il facilite la création des commandes fournisseurs, et permet la réception et la mise à jour automatique des stocks.

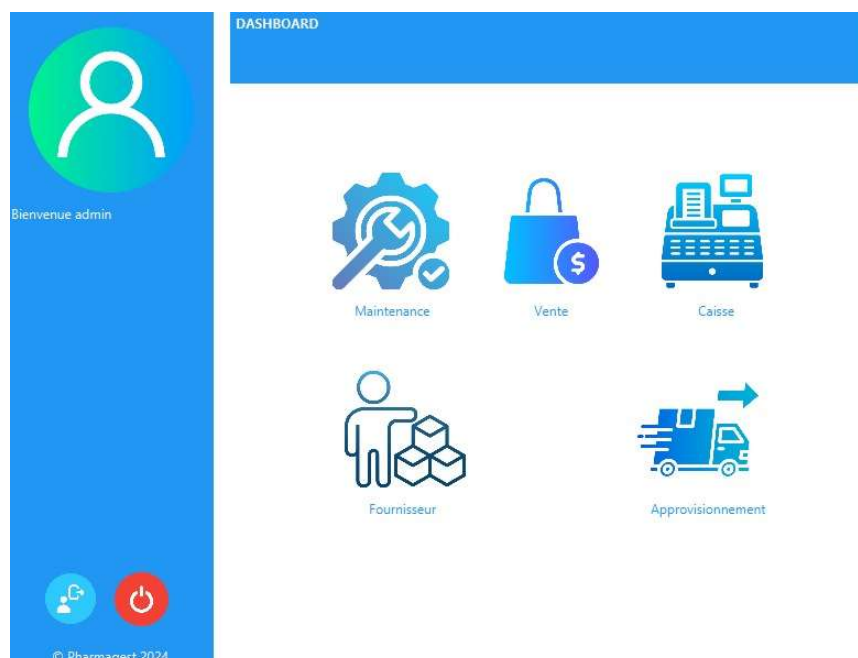
3. LES SCHEMAS EXPLICATIFS

Dans cette section, nous présentons des schémas illustrant les interfaces utilisateurs du projet **PharmaGest**. Ces interfaces permettent de visualiser les étapes essentielles de l'utilisation de l'application par un pharmacien, un vendeur ou un administrateur.

Les écrans sélectionnés mettent en avant les fonctionnalités clés suivantes :

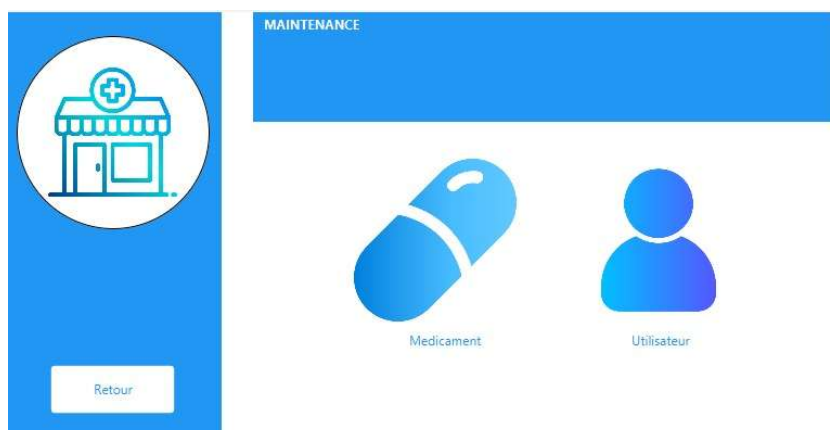
- **Interface de connexion** : Elle permet aux utilisateurs de s'authentifier de manière sécurisée afin d'accéder à l'application PHARMAGEST.

- **Dashboard** : Interface principale après la connexion. Elle offre un accès rapide aux modules Maintenance, Vente, Caisse, Fournisseur et Approvisionnement via des icônes claires et intuitives.



- **Maintenance** : L'interface Maintenance permet d'accéder aux fonctionnalités essentielles liées à la gestion des utilisateurs et des médicaments. Elle se présente sous forme de deux grandes icônes : l'une

dédiée à la gestion des médicaments et l'autre à la gestion des comptes utilisateurs.



- **Médicament :**
L'interface Médicament permet de gérer l'ensemble du stock de médicaments de la pharmacie. Elle offre plusieurs fonctionnalités essentielles : l'ajout de nouveaux médicaments, la modification des informations existantes, la suppression, ainsi que la consultation détaillée de la liste des médicaments enregistrés.

MÉDICAMENT										Retour

ainsi que la suppression de comptes.

The screenshot shows a web interface titled 'UTILISATEUR'. At the top right is a 'Retour' button. Below the title bar, there are three buttons: 'Tous les utilisateurs', a search input field labeled 'Rechercher...', and a 'Rechercher' button. Below these are three more buttons: 'Ajouter utilisateur', 'Supprimer', and 'Modifier'. The main part of the interface is a table with two columns: 'Utilisateur' and 'Role'. The table contains three rows of data:

Utilisateur	Role
admin	PHARMAC...
user1	PHARMAC...
user2	PHARMAC...

- **Interface de Vente** : L'interface Vente permet de réaliser les opérations de vente de médicaments de manière rapide et efficace. L'utilisateur peut rechercher un médicament, en vérifier la disponibilité en stock, sélectionner la quantité à vendre, et enregistrer la transaction. L'interface calcule automatiquement le montant total à payer, facilitant ainsi le passage en caisse.

The screenshot shows a web interface titled 'Création de VENTE(Pharmacien)'. It has a 'Retour' button at the top right. The interface is divided into several sections:

- Type de Vente**: A dropdown menu currently set to 'PRESCRITE'.
- Informations du Patient**: Includes a search input 'Rechercher un patient...', a 'Rechercher' button, and fields for 'Nom', 'Prénom', 'Date de naissance', 'Adresse', and 'Contact'.
- Informations de la Prescription**: Includes fields for 'Nom du médecin' and 'Date de prescription'.
- Médicaments Disponibles**: A section with a search input 'Rechercher un médicament...' and a table showing available medications:

Nom	Forme	Stock Actuel	Prix Vente
Paracetamol 500mg	Comprimé	50	20.0
Doliprane 1000 mg	Comprimé	90	40.0

Below the table is a 'Quantité' input field and an 'Ajouter à la vente' button.

- Lignes de Vente en Cours**: A table with columns 'Médicament', 'Quantité', 'Prix Unitaire', and 'Sous-total'. It currently shows 'Aucun contenu dans la table'.

At the bottom right, there is a red button 'X Supprimer Ligne(s)', a text label 'Montant Total : 0.00 €', and a blue button 'Valider la Vente'.

- **Interface Caisse** : L'interface Caisse est dédiée à la gestion des transactions financières effectuées dans la pharmacie. Elle permet de suivre en temps réel les ventes journalières, d'enregistrer les paiements reçus, et de générer des rapports de caisse en fin de journée.

Caisse - Vente en attente

Ventes à encaisser

Date	Montant Total	Facture	Type
Aucun contenu dans la table			

Détails de la vente

Date :

Montant total :

Type de vente :

Monnaie à rendre :

Médicaments associés

Médicament	Quantité	Prix unitaire
Aucun contenu dans la table		

Montant reçu :

Valider le paiement

Pavé numérique

7	8	9
4	5	6
1	2	3
0	.	C

- Ticket : Lorsqu'une vente est effectuée, l'application génère un ticket de manière automatique. Ce ticket contient toutes les informations nécessaires, telles que les articles vendus, le montant total et la date de la transaction.

```

=====
PHARMACIE PHARMAGEST
=====

Date: 18/04/2025 09:40
Facture N°: FAC-16-1744954844035
Type: LIBRE

-----
PRODUIT      QTÉ   PRIX   TOTAL
-----
Doliprane 1000 mg    20   40,00€  800,00€
Paracetamol 500mg    20   20,00€  400,00€
-----
TOTAL: 1200,00€
-----

Merci pour votre visite!
Conservez ce ticket pour tout échange.
=====

```

- **Interface Fournisseur** : Cette interface permet à l'utilisateur d'ajouter, modifier, consulter ou supprimer les informations liées aux fournisseurs.

FOURNISSEUR
Retour

Tous les fournisseurs
Rechercher...
Rechercher

Ajouter fournisseur
Modifier
Supprimer

ID	Nom	Adresse	Contact	Email	
1	Chan	louis	5554470	sensiro456...	
3	ff	ff	7777777	kkkk@gma...	

- Interface Approvisionnement** : L'interface d'approvisionnement permet à l'utilisateur de gérer tout le processus de commande et de réception des médicaments, depuis la demande jusqu'à la livraison. Elle est divisée en trois sections principales accessibles via trois boutons clairs et distincts.

APPROVISIONNEMENT
Retour

Liste des commandes

Livraisons

Gestion des commandes

- Liste des commandes** : Ce bouton redirige vers une page affichant l'historique complet des commandes effectuées auprès des fournisseurs. Chaque commande est présentée avec des détails tels que la date, le fournisseur concerné, le statut (en attente, validée, livrée), et le montant total.

Une fois toutes les configurations terminées, le lancement de l'application se fait localement en exécutant le fichier qui ouvre l'interface utilisateur principale (dashboard).

Cela nous a permis de tester et d'améliorer toutes les fonctionnalités en environnement local.

6. HEBERGEMENT DE LA BASE DE DONNEES EN LIGNE

Pour rendre notre application **PharmaGest** plus accessible et simuler un environnement proche de la production, nous avons procédé à **l'hébergement en ligne de la base de données**.

Nous avons utilisé la plateforme **AlwaysData** pour héberger la base **PostgreSQL** créée initialement en local.

La procédure mise en œuvre a été la suivante :

- Exportation de la base de données locale au format SQL.
- Création d'une base de données PostgreSQL sur AlwaysData.
- Importation des données dans la base distante via l'interface d'administration AlwaysData.
- Modification des paramètres de connexion dans l'application Java afin d'établir la connexion avec la base hébergée (en indiquant l'adresse du serveur AlwaysData, le port, le nom de la base, l'utilisateur et le mot de passe).

Grâce à cette solution, nous avons pu simuler une utilisation professionnelle de l'application, en séparant la couche applicative (Java/JavaFX) et la couche base de données (PostgreSQL hébergé en ligne).

CONCLUSION

Le projet **PharmaGest** a constitué une étape essentielle dans notre parcours de formation.

Il nous a permis de mettre en pratique nos connaissances théoriques et de développer des compétences clés telles que le travail en équipe, la résolution de problèmes et la gestion de projet.

À travers ce projet, nous avons démontré notre capacité à collaborer efficacement, à communiquer clairement et à répartir les tâches de manière équilibrée entre les membres de l'équipe.

Sur le plan technique, nous avons renforcé nos compétences en développement d'applications desktop avec **Java/JavaFX**, en conception d'interfaces utilisateur avec **Scene Builder**, ainsi qu'en gestion de bases de données relationnelles avec **PostgreSQL**.

La mise en œuvre de l'application et l'hébergement de la base de données en ligne sur **AlwaysData** nous ont permis de nous rapprocher des conditions réelles d'un projet informatique en entreprise.

Ce projet fut ainsi une expérience enrichissante, aussi bien sur le plan professionnel que personnel, et représente une base solide pour nos futures réalisations dans le domaine du développement d'applications.