

<b>DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE</b>		<b>N° réalisation : RP2</b>
<b>Nom, prénom : RAZAFITSIAROVANA Ialitiana Manoa</b>		<b>N° candidat : 02344518736</b>
<b>Épreuve ponctuelle</b> <input checked="" type="checkbox"/>	<b>Contrôle en cours de formation</b> <input type="checkbox"/>	<b>Date : 19 / 04 /2024</b>
<b>Organisation support de la réalisation professionnelle :</b> Conception d'une application qui permettra d'informatiser la gestion d'une pharmacie avec les interfaces de vente, caisse, approvisionnement.		
<b>Intitulé de la réalisation professionnelle : PHARMAGEST</b> Développement d'une application qui va permettre la facilitation de la gestion de l'approvisionnement, la vente au comptoir, gérer la caisse et la partie finance.		
<b>Période de réalisation : Lieu : MCCI Business School(CCI de Maurice) .....</b> <b>Modalité : <input type="checkbox"/> Seul(e) <input checked="" type="checkbox"/> En équipe</b>		
<b>Compétences travaillées</b> <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données		
<b>Conditions de réalisation<sup>1</sup> (ressources fournies, résultats attendus)</b> Ressources fournies : contexte (cahier de charge), accès à internet Résultats attendus : -une application qui va gérer permettre d'informatiser la gestion du stock de médicament -une interface de vente pour gérer la vente au comptoir en respectant les impératifs légaux autour des ordonnances -informatiser la gestion de la caisse et faciliter la gestion des finances		
<b>Description des ressources documentaires, matérielles et logicielles utilisées<sup>2</sup></b> Outils collaboratifs et suivis du projet : Microsoft planner, Microsoft teams, GitHub IDE : intellij IDEA Langage de développement : Java, Java Fx SGBD : PostgreSQL, pgadmin Analyse : UML		
<b>Modalités d'accès aux productions<sup>3</sup> et à leur documentation<sup>4</sup></b> Lien GitHub : <a href="https://github.com/Angelin57/PHARMAGEST.git">https://github.com/Angelin57/PHARMAGEST.git</a> Lien documentation : <a href="https://github.com/Manoa-123/RP2-DOCUMENTATION.git">https://github.com/Manoa-123/RP2-DOCUMENTATION.git</a>		

<sup>1</sup> En référence aux conditions de réalisation et ressources nécessaires du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

<sup>2</sup> Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

<sup>3</sup> Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

<sup>4</sup> Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

## Épreuve E5 - Conception et développement d'applications (option SLAM)

**ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle  
(verso, éventuellement pages suivantes)****Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs**

Le projet PHARMAGEST a pour objectif d'informatiser les opérations de la pharmacie PHARMAGEST qui emploie un système traditionnel pour vendre ces produits pharmaceutiques. Elle veut une application qui permet de gérer efficacement les stocks de médicament et améliorer la partie vente.

Pour développer PHARMAGEST nous avons utilisé les outils suivants :

Pour faciliter le développement en groupe nous avons utilisé GitHub, Microsoft planner, Microsoft teams.

Voici une description des outils et environnement de développement de PHARMAGEST :

- Java /java fx/Scene Builder : Java est un langage de programmation populaire et polyvalent pour développer une grande variété d'application comme PHARMAGEST. Il est apprécié pour sa portabilité et sa robustesse, Java suis le paradigme de la programmation orienté objet (POO) quant à JavaFX c'est une bibliothèque graphique intégrée dans le JRE et le JDK de Java, elle permet de créer des interfaces graphiques évolués et modernes. Scene Builder est un outil visuel qui facilite la conception d'interfaces graphiques pour les applications JavaFX.
- (IDE)Intellij IDEA : on a utilisé Intellij IDEA (Community Edition) comme IDE. Intellij IDEA est un environnement de développement intégré puissant et polyvalent principalement utilisé pour la programmation Java.
- SGBD : PostgreSQL/pgmyadmin : PostgreSQL est une base de données relationnelle open source puissante et robuste quant à pgAdmin c'est une plateforme d'administration et de développement pour PostgreSQL.

Il y aura donc dans l'application les interfaces suivantes :

**L'interface de connexion** : l'interface pour permettre à l'utilisateur de s'authentifier, il faut prendre en compte les fonctions de chaque utilisateur (seul le pharmacien qui est l'administrateur a le droit d'accès à toutes les interfaces, le vendeur n'a le droit d'accès qu'à l'interface de la vente et le caissier, n'a accès qu'à l'interface caisse)

**Les interfaces qui permettent de gérer la famille des médicaments, les médicaments** : pour la gestion des médicaments, chaque médicament est classé en famille de produits (ex. psychotrope, analgésique, anti-inflammatoire, etc...). Ces interfaces permettent de rechercher, d'ajouter, de mettre à jour et de supprimer les familles des médicaments et les médicaments.

**L'interface profile** : c'est l'interface qui permet de gérer les profils des utilisateurs de l'application et seul l'administrateur a le droit d'y accéder.

**L'interface vente** : c'est l'interface principale qui permet de gérer la vente au comptoir. Le vendeur doit y préciser s'il s'agit d'une vente avec ordonnance ou une vente sans ordonnance.

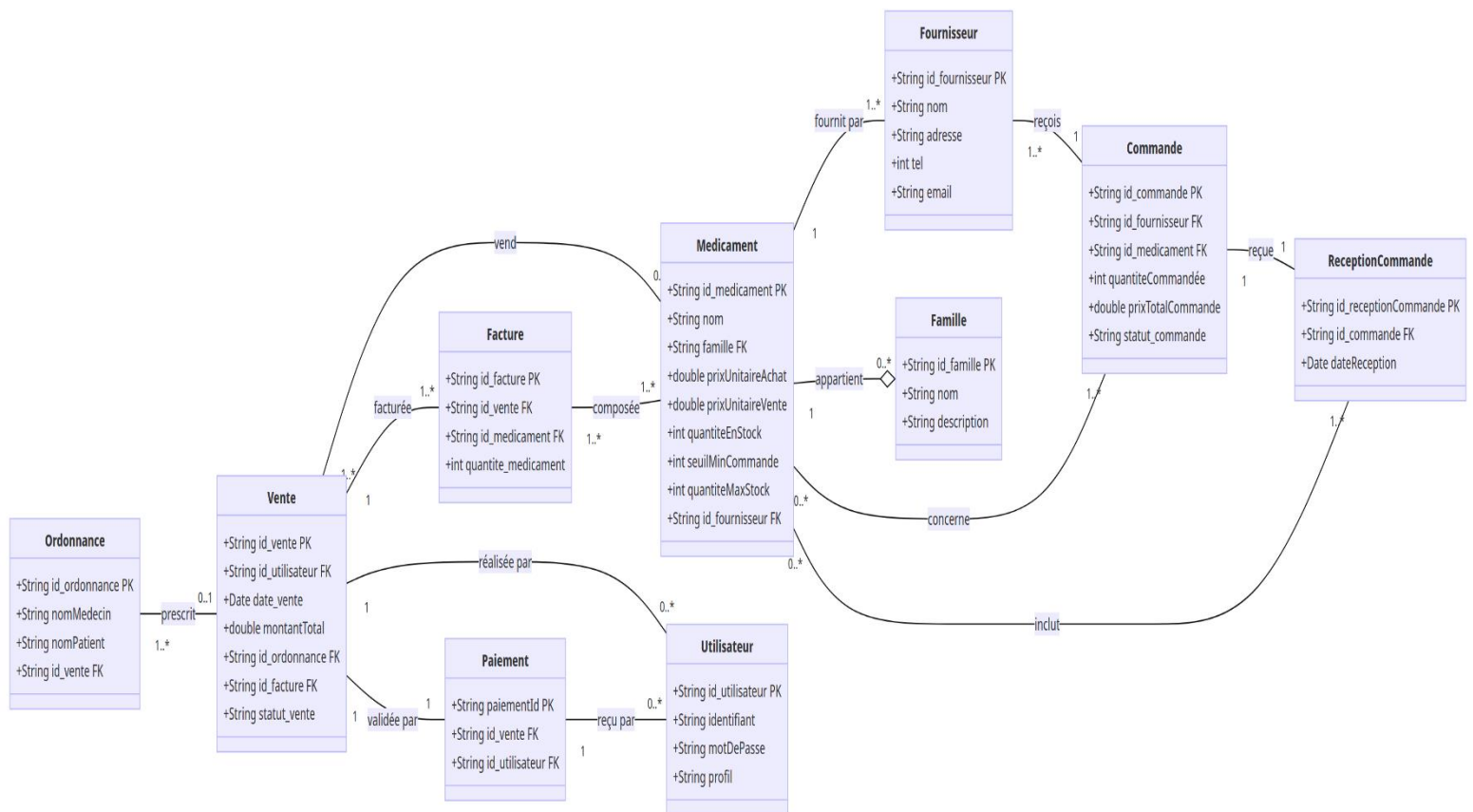
**L'interface caisse** : c'est l'interface qui va permettre de gérer la caisse. De l'interface vente le vendeur va envoyer à la caisse la facture qui montre les médicaments commandés par le client. Dans la caisse il va juste avoir dans un tableau la liste de toutes les factures non payées. Quand il va cliquer sur une facture, la liste des médicaments dans la facture vont s'afficher sur une autre table. Elle permet aussi d'automatiser les opérations à la caisse lors du paiement (elle va afficher le montant à payer, le caissier va saisir le montant reçu et elle va afficher automatiquement le montant à rendre)

**L'interface approvisionnement** : c'est l'interface où on gère l'approvisionnement et qui permet de gérer automatiquement le stock pour éviter la rupture.

**L'interface réception commande** : l'interface montre un tableau qui afficher les listes des médicaments reçu après livraison et le nouveau stock de médicaments

## PRESENTATION DE L'APPLICATION (AVEC CAPTURES D'ECRAN)

### 1. UML du site :



### 2.Interface de connexion



Username

Password

☐ Admin ☐ Caisse ☐ Vente

Connecter

Annuler

Préciser la fonction de l'utilisateur

## 2.Interface vente :

PharmaGest 19/04/2024 00:34:30 angelin

**Médicaments**

Nom\_médicament :

Famille :

Prix unitaire :

Quantité achetée :

Avec Ordonnance Sans Ordonnance

Client :

Médecin :

Ajouter Annuler

Code_Médicament	Nom_médicament	Prix_Unit	Quantité_achetée	Prix_Total
Aucun contenu dans la table				

Envoyer à la caisse

Tableau qui va afficher la liste des commandes à envoyer à la caisse

Vente avec ou sans ordonnance

## 3.Interface caisse



5.Interface fournisseur :

PharmaGest

19/04/2024 00:35:17

angelin

Dashboard

Vente

Caisse

Medicament

Fournisseur

Famille

Profile

Approvisionnement

Reception commnde

Ajout fournisseur

QUICK ACCESS

Fournisseur ID

ID

Nom

Tel

Email

Rechercher

Mise a jour

Supprimer

Tous les fournisseurs

Nom

Telephone

Tel

Email

Ajouter

id	nom	telephone	email
3	fournisseur2	25555	fournisseur2@gmail.com
4	Fournisseur3	55555	Fournisseur3@gmail.com

6.Interface famille : (pour gérer la famille des médicaments)

PharmaGest

19/04/2024 00:35:40

angelin

Dashboard

Vente

Caisse

Medicament

Fournisseur

Famille

Profile

Approvisionnement

Reception commnde

Ajout Famille

ID Famille :

ID

Nom\_famille :

Nom Famille

Rechercher

Mise a jour

Supprimer

Voir toutes les familles

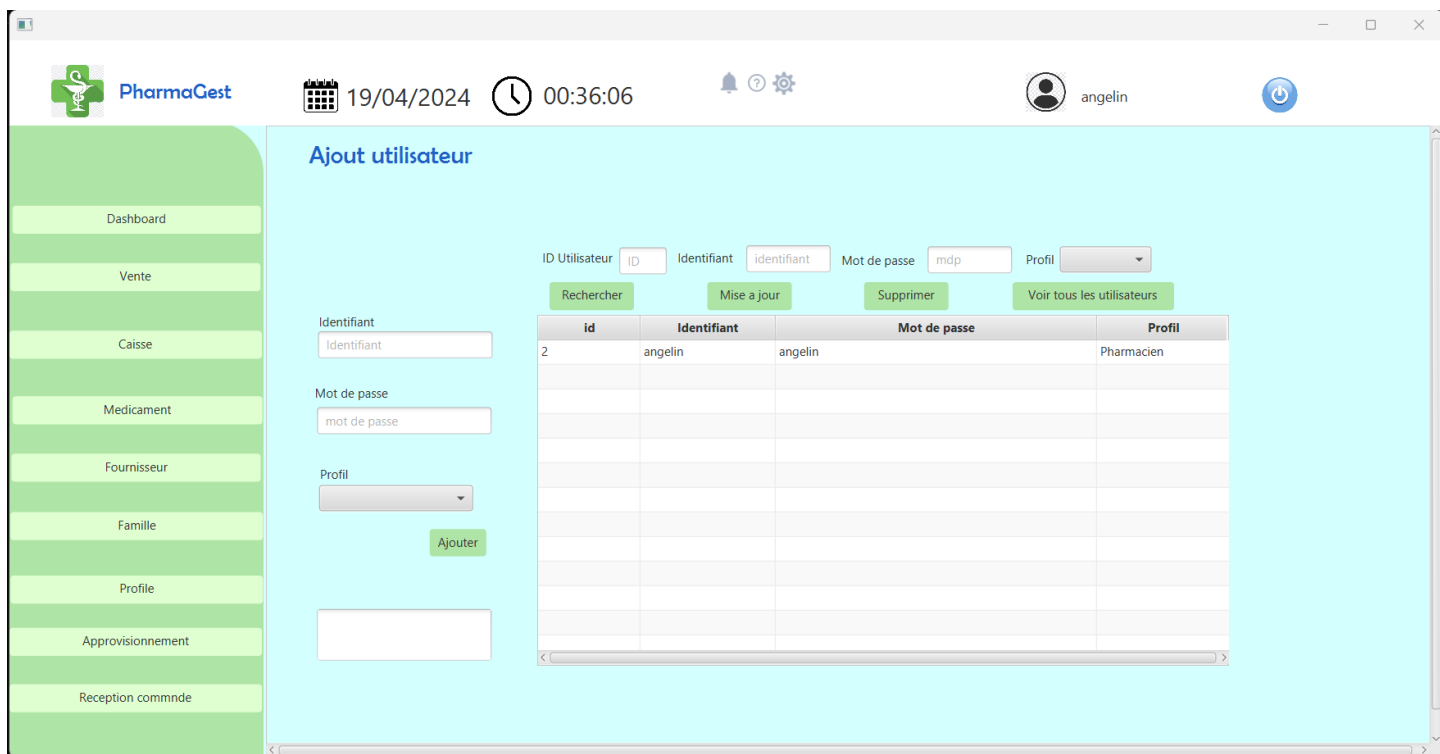
Nom

Nom

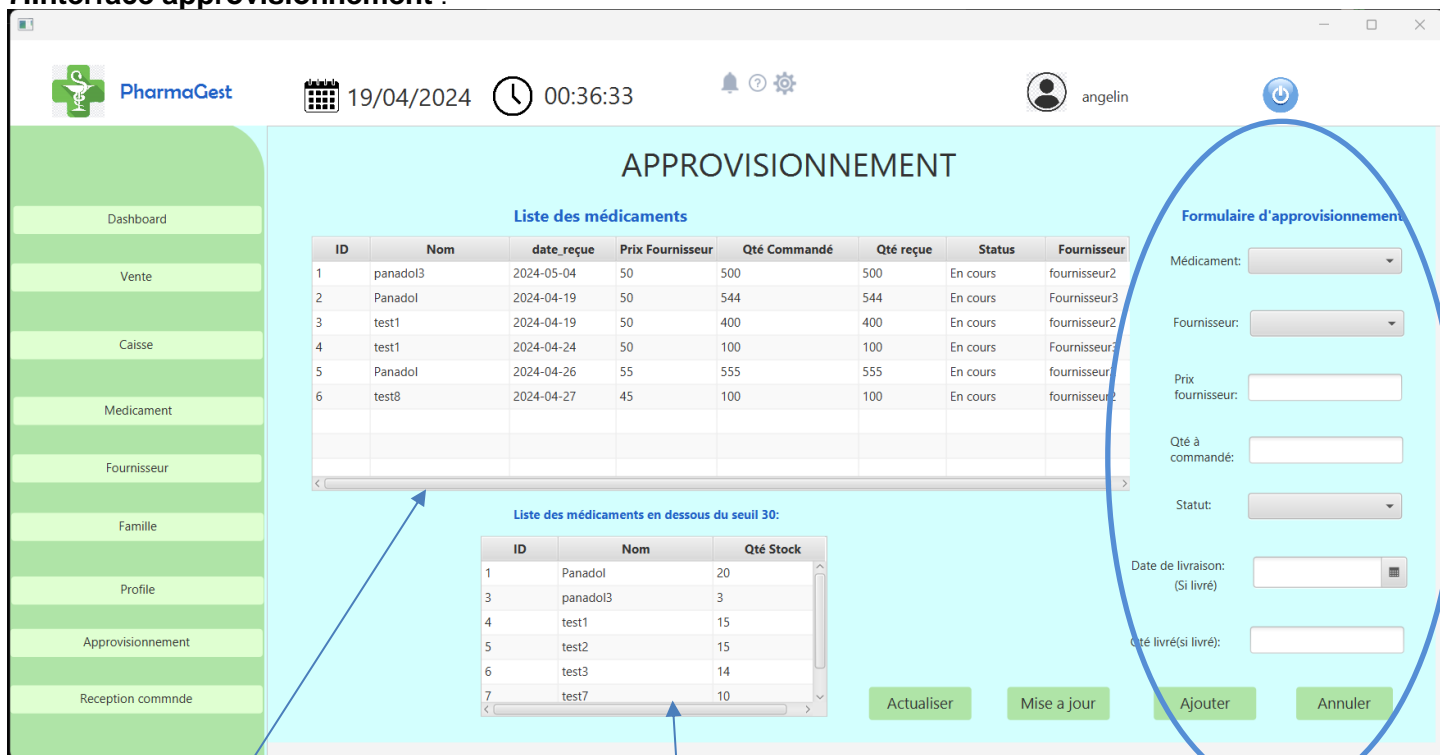
Ajouter

id	nom_famille
1	Pilules
2	effervesence
4	test
6	test5

Interface utilisateur (pour gérer les profils d'utilisateurs dans l'application)



## 7.Interface approvisionnement :



La liste de médicament dont le seuil dont la quantité en stock est inférieure à 30(mise à jour recommandée)

La liste des médicaments commandés et reçus (avec statut en cours), le stock est mis à jour si le statut est terminé

Formulaire d'approvisionnement pour mettre à jour le stock

## 8.Interface réception commande :

le tableau montre la liste des médicaments reçus et affiche les anciennes et les nouvelles quantités de médicaments



en stock

## 9.Trigger :

On a créer une table user\_login pour enregistre la date et l'heure de connexion des utilisateurs

```
CREATE OR REPLACE FUNCTION update_login_trigger()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.identifiant IN (SELECT identifiant FROM utilisateur) THEN
        -- L'utilisateur existe déjà, donc mettre à jour l'heure et la date de connexion
        UPDATE user_logins
        SET login_time = NOW()
        WHERE identifiant = NEW.identifiant;
    ELSE
        -- L'utilisateur n'existe pas encore, donc insérez un nouvel enregistrement
        INSERT INTO user_logins (identifiant, login_time)
        VALUES (NEW.identifiant, NOW());
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER login_trigger
AFTER INSERT ON utilisateur
FOR EACH ROW
EXECUTE FUNCTION update_login_trigger();
```



Ce déclencheur est conçu pour mettre à jour ou insérer des enregistrements dans une table `user\_logins` en fonction des insertions dans la table `utilisateur`. Voici une explication détaillée du fonctionnement de ce trigger :

Fonction `update_login_trigger()` :

- Elle prend en paramètre un nouvel enregistrement qui représente la ligne insérée dans la table utilisateur.
- À l'intérieur de la fonction, il y a une structure conditionnelle pour vérifier si l'identifiant de l'enregistrement nouvellement inséré existe déjà dans la table utilisateur.
- Si l'identifiant existe, la fonction effectue une mise à jour (`UPDATE`) de l'heure de connexion pour cet identifiant
- Sinon, si l'identifiant n'existe pas encore, la fonction effectue une insertion d'un nouvel enregistrement avec l'identifiant et l'heure de connexion actuelle.

Création du déclencheur :

- Un déclencheur (`trigger`) nommé `login_trigger` est créé après chaque insertion dans la table utilisateur
- L'action du déclencheur est d'exécuter la fonction `update_login_trigger()` lorsqu'un nouvel enregistrement est inséré dans la table utilisateur.

Ce déclencheur assure que chaque fois qu'un nouvel utilisateur est inséré dans la table utilisateur, il met à jour l'heure de connexion dans la table `user\_logins` si l'utilisateur existe déjà, sinon il insère un nouvel enregistrement avec l'heure de connexion actuelle. Cela permet de garder une trace des heures de connexion des utilisateurs de manière automatique.