



PHARMAGEST.

1. Classe Utilisateur

- **Description** : Classe abstraite qui regroupe les informations d'identification et les méthodes communes à tous les utilisateurs du système (pharmaciens et vendeurs).
- **Attributs** :
 - `identifiant` : Nom d'utilisateur ou email pour la connexion.
 - `motDePasse` : Mot de passe de l'utilisateur.
- **Méthodes** :
 - `seConnecter()` : Méthode pour authentifier l'utilisateur.
 - `deconnecter()` : Méthode pour se déconnecter du système.
- **Héritage** :
 - **Pharmacien** et **Vendeur** héritent de cette classe.
- **Exemple** : Un pharmacien avec l'identifiant `pharma123` et le mot de passe `securePass`.

2. Classe Pharmacien

- **Description** : Spécialisation de la classe `Utilisateur`. Les pharmaciens ont des responsabilités spécifiques comme la gestion des ventes, des commandes, et des stocks.
- **Méthodes** :
 - `validerVente()` : Le pharmacien valide une vente effectuée par le vendeur.
 - `validerCommande()` : Le pharmacien valide une commande d'approvisionnement.
 - `verifierApprovisionnement()` : Vérification des niveaux de stock pour lancer une commande.
 - `modifierQuantiteCommande()` : Ajuster la quantité commandée avant

validation.

- **Exemple** : Le pharmacien **Jean Dupuis** valide une commande de réapprovisionnement pour 100 boîtes de "Paracétamol".
-

3. Classe vendeur

- **Description** : Spécialisation de la classe `Utilisateur`. Les vendeurs sont chargés de la gestion des ventes au comptoir et des paiements.
 - **Méthodes** :
 - `effectuerVente()` : Enregistre une vente.
 - `gererPaiement()` : Gère le paiement des médicaments par le client.
 - **Exemple** : Le vendeur **Pierre Martin** enregistre une vente de 5 boîtes de "Ibuprofène" et gère le paiement en espèces.
-

4. Classe `Compte`

- **Description** : Représente le compte d'un utilisateur (pharmacien ou vendeur). Contient les informations nécessaires à la connexion et à l'authentification.
 - **Attributs** :
 - `identifiant` : Nom d'utilisateur pour se connecter.
 - `motDePasse` : Mot de passe pour la connexion.
 - `role` : Rôle de l'utilisateur (Pharmacien ou Vendeur).
 - **Méthodes** :
 - `seConnecter()` et `deconnecter()` : Pour l'authentification et la déconnexion.
 - **Composition** :
 - Un **Utilisateur** possède un **Compte** (1..1).
 - **Exemple** : Le compte de **Jean Dupuis** utilise l'identifiant `jdupuis` et le mot de passe `hashedpassword`.
-

5. Classe `Medicament`

- **Description** : Représente un médicament dans la pharmacie, avec ses informations de stock, prix, etc.
 - **Attributs** :
 - `nom` : Nom du médicament (ex: "Paracétamol").
 - `forme` : Forme du médicament (ex: "Comprimé", "Sirop").
 - `prixAchat` et `prixVente` : Prix d'achat et de vente du médicament.
 - `stock` : Quantité en stock.
 - `seuilCommande` : Seuil de stock pour déclencher une commande.
 - `qteMax` : Quantité maximale de stock.
 - **Relations** :
 - **1 Medicament appartient à 1 Famille** (o-- 1).
 - **1 Medicament est fourni par 1 Fournisseur habituel** (o-- 1).
 - **1 Medicament est inclus dans plusieurs LigneDeCommande** (*-- 1).
 - **1 Medicament a une Unite spécifique** (--> 1).
 - **Exemple** : Le médicament **Ibuprofène** est en stock avec 50 unités et a un seuil de commande de 10 unités.
-

6. Classe `Famille`

- **Description** : Catégorie à laquelle un médicament appartient (ex: Antibiotiques, Analgésiques).
- **Attribut** :
 - `nom` : Nom de la famille.
- **Relations** :
 - **1 Famille contient plusieurs Medicament** (0..*).
- **Exemple** : La famille **Anti-inflammatoires** contient des médicaments comme **Ibuprofène**.

7. Classe Fournisseur

- **Description** : Représente un fournisseur de médicaments pour la pharmacie.
 - **Attributs** :
 - nom : Nom du fournisseur.
 - adresse, contact.
 - **Relations** :
 - **1 Fournisseur fournit plusieurs Medicament** (0..*).
 - **Exemple** : Le fournisseur **PharmaDistributeur** fournit des médicaments comme **Ibuprofène** et **Paracétamol**.
-

8. Classe Commande

- **Description** : Représente une commande passée par la pharmacie pour réapprovisionner les stocks.
 - **Attribut** :
 - montant : Montant total de la commande.
 - **Méthodes** :
 - genererCommande(), validerCommande(), envoyerCommandePDF().
 - **Relations** :
 - **1 Commande contient plusieurs LigneDeCommande** (*-- 0..*).
 - **1 Commande est validée par 1 Pharmacien** (--> 1).
 - **1 Commande est livrée par plusieurs Livraison** (1..*).
 - **Exemple** : La pharmacie passe une commande de 100 boîtes de "Paracétamol" pour un montant total de 500 €.
-

9. Classe LigneDeCommande

- **Description** : Représente un médicament spécifique dans une commande.
 - **Attributs** :
 - quantite : Quantité commandée du médicament.
 - prixUnitaire : Prix unitaire du médicament dans la commande.
 - montantLigne : Montant total pour cette ligne de commande.
 - **Relations** :
 - **1 LigneDeCommande réfère à 1 Medicament** (*-- 1).
 - **Exemple** : Une ligne de commande pour 50 unités de **Ibuprofène** à 2 € l'unité, montant total de 100 €.
-

10. Classe Vente

- **Description** : Représente une vente effectuée au comptoir.

- **Attributs :**
 - `dateVente`, `montantTotal`, `typeVente` (Libre ou Prescrite).
 - **Méthodes :**
 - `effectuerVente()`.
 - **Relations :**
 - **1 Vente vend plusieurs Medicament** (1..*).
 - **1 Vente peut être liée à 1 Prescription** (0..1).
 - **1 Vente génère 1 Facture** (1).
 - **1 Vente est associée à 1 Paiement** (1).
 - **Exemple :** Vente de 5 boîtes de **Ibuprofène** pour un montant de 20 €.
-

11. Classe `Facture`

- **Description :** Représente une facture générée après une vente.
 - **Attributs :**
 - `dateEmission`, `montantTotal`, `numeroFacture`.
 - **Méthode :**
 - `genererFacture()`.
 - **Relations :**
 - **1 Vente génère 1 Facture** (1).
 - **Exemple :** La facture n°**1234** est générée pour une vente de 50 €.
-

12. Classe `Paiement`

- **Description :** Représente le paiement d'une vente.
 - **Attributs :**
 - `montant`, `modePaiement`, `statut`.
 - **Méthodes :**
 - `effectuerPaiement()`.
 - **Relations :**
 - **1 Vente est associée à 1 Paiement** (1).
 - **1 Paiement a 1 StatutPaiement** (1).
 - **Exemple :** Paiement de 20 € en espèces pour une vente de 5 boîtes de "Ibuprofène".
-

13. Classe `Prescription`

- **Description :** Représente une ordonnance médicale pour un médicament.
- **Attributs :**
 - `nomMedecin`, `datePrescription`.
- **Méthodes :**
 - `getNumeroPrescription()`.
- **Relations :**
 - **1 Prescription est donnée à 1 Patient** (1).

- **1 Vente peut être liée à 1 Prescription** (0..1).
 - **Exemple** : La prescription du **Dr Dupont** est donnée à **Pierre** pour acheter des **Antibiotiques**.
-

14. Classe Patient

- **Description** : Représente le patient pour lequel une ordonnance a été émise.
 - **Attributs** :
 - nom, prenom, dateNaissance, adresse, contact.
 - **Relations** :
 - **1 Prescription est donnée à 1 Patient** (1).
 - **Exemple** : Le patient **Pierre Dupuis** a une prescription pour des **Antibiotiques**.
-

15. Classe Livraison

- **Description** : Représente une livraison de médicaments commandés à un fournisseur.
 - **Attributs** :
 - dateLivraison, status.
 - **Méthodes** :
 - suivreLivraison(), recevoirCommande(), mettreAJourStock().
 - **Relations** :
 - **1 Commande est livrée par plusieurs Livraison** (1..*).
 - **1 Livraison est effectuée par plusieurs Fournisseur** (1..*).
 - **Exemple** : La livraison des médicaments commandés a été effectuée le **10 septembre 2023** par le fournisseur **PharmaDistributeur**.
-

16. Enums TypeVente et StatutPaieement

- **TypeVente** : Indique si la vente est **Libre** ou **Prescrite**.
- **StatutPaieement** : Indique si le paiement est **En attente**, **Validé** ou **Rejeté**.

Conclusion :

Ce modèle représente toutes les classes du système PHARMAGEST, avec leurs relations, cardinalités, et des exemples pratiques de leur utilisation. Ce modèle est conçu pour gérer efficacement la vente, la prescription, l'approvisionnement, et le suivi des médicaments dans une pharmacie.