**Cahier des Charges : Gestion de Stock pour une Pharmacie.**

**1. Introduction**

Le présent cahier des charges vise à définir les fonctionnalités, les exigences techniques, et les contraintes du projet de gestion de stock pour une pharmacie. Ce système devra permettre aux différents acteurs de gérer efficacement les stocks de médicaments, les commandes clients et les interactions avec les pharmacies.

**2. Objectifs**

* Développer une application accessible pour les pharmacies, les clients et l'administration pour gérer les stocks, les commandes, et les comptes utilisateurs.
* Assurer l'intégration avec des outils de géolocalisation, des technologies IA pour la lecture des ordonnances et l’analyse des stocks.
* Offrir un service sécurisé et convivial pour chaque acteur impliqué.
* Un Chat entre pharmacie et client.

**3. Acteurs du Système**

1. **Admin** : Supervise les pharmacies, gère les comptes, et génère des rapports.
2. **Pharmacie** : Gère le stock, les commandes, et l'état des stocks.
3. **Client** : Passe des commandes, consulte les médicaments et contacte les pharmacies.

**4. Cas d’utilisation**

**4.1 Authentification & Gestion des Comptes**

* **Acteurs concernés** : Admin, Pharmacie, Client
* **Fonctionnalité** : Authentification des utilisateurs et gestion de leurs profils.
  + Inscription, connexion, récupération de mot de passe.
  + Gestion des comptes pharmacies et clients par l'Admin.

**4.2 Gestion du Stock (Pharmacie)**

* **Acteur concerné** : Pharmacie
* **Fonctionnalités** :
  + Ajouter, modifier, et supprimer des médicaments.
  + Suivi de l'état du stock.
  + Réception des alertes de rupture de stock.
  + Gestion des bons de commande (émission et validation).
  + Localisation de la pharmacie via MapBox.

**4.3 Supervision (Admin)**

* **Acteur concerné** : Admin
* **Fonctionnalités** :
  + Validation des nouvelles pharmacies inscrites.
  + Gestion des alertes (rupture de stock, commandes à valider).
  + Reporting et statistiques des ventes et ruptures de stock.

**4.4 Machine Learning (Admin)**

* **Acteur concerné** : Admin
* **Fonctionnalités** :
  + Prédiction des besoins en stock.
  + Détection des interactions médicamenteuses.
  + Génération de rapports basés sur l’analyse de données (ventes, ruptures de stock).

**4.5 Création de Compte et Commandes (Client)**

* **Acteur concerné** : Client
* **Fonctionnalités** :
  + Création et gestion du compte client.
  + Passer des commandes, rechercher des médicaments.
  + Ajouter des médicaments au panier et finaliser l’achat.
  + Contacter une pharmacie via message ou appel.
  + Affichage des pharmacies proches via géolocalisation.

**4.6 Fonctionnalités Optionnelle**

* **Lecture d'ordonnance** : Scanner une ordonnance pour détecter automatiquement les médicaments via l'IA.
* **Chatbot IA** : Recommander des médicaments en fonction des symptômes et des besoins.

**5. Exigences Fonctionnelles**

**5.1 Interface Utilisateur**

* **Frontend Web** : Application responsive utilisant React et TailwindCSS.
* **Mobile** : Application Android native développée avec Java dans Android Studio.
* **Accessibilité** : Interface facile d’utilisation pour chaque acteur.

**5.2 Backend**

* **Spring Boot** : REST API pour la gestion des données et des interactions entre le frontend et la base de données.
* **Base de données** : PostgreSQL pour les données relationnelles et MongoDB pour le stockage des données non structurées.

**5.3 Technologies Avancées**

* **Machine Learning** : Python, TensorFlow pour la prédiction des stocks et l’analyse des interactions médicamenteuses.
* **Lecture d’ordonnance** : OpenCV et TensorFlow pour la détection des médicaments.
* **Géolocalisation** : Intégration avec Google Maps API ou OpenStreetMap pour localiser les pharmacies proches.

**5.4 Architecture**

* **Microservices** : Architecture modulaire pour une meilleure gestion des fonctionnalités indépendantes.
* **DevOps** : Utilisation de Docker, Git, et CI/CD pour les processus de déploiement et de mise à jour continue.