

Cahier des Charges : Plateforme de Santé Personnalisée avec IA pour la Prévention et Suivi Médical (Microservices)

1. Contexte et Objectif du Projet

La santé est un domaine crucial où la technologie peut apporter des solutions innovantes pour améliorer la qualité de vie des individus. Le projet proposé vise à développer une **plateforme de santé personnalisée** qui intègre des **microservices**, **l'intelligence artificielle (IA)** et la **téléconsultation** pour offrir un suivi proactif et des recommandations médicales en temps réel. Cette plateforme sera accessible via une interface web moderne basée sur **Angular** et communiquera avec un backend basé sur **Spring Boot** pour gérer les différents services.

Objectif principal :

Développer une plateforme permettant de surveiller la santé des utilisateurs en collectant des données via des appareils connectés, d'utiliser l'IA pour fournir des recommandations personnalisées, et d'offrir des services de téléconsultation pour un suivi médical de proximité.

2. Fonctionnalités du Projet

2.1. Gestion des utilisateurs

- **Inscription et authentification des utilisateurs** via un système de gestion d'identité (ex : **Keycloak**).
- Gestion des profils utilisateurs, y compris la saisie des données personnelles et médicales.
- Authentification sécurisée avec des **tokens JWT** et gestion des rôles d'utilisateur (ex : patient, médecin).

2.2. Suivi des paramètres de santé

- **Collecte en temps réel** des données des appareils connectés (montres intelligentes, capteurs de santé, etc.) : fréquence cardiaque, activité physique, qualité du sommeil, etc.
- **Affichage des statistiques** des paramètres de santé dans un tableau de bord utilisateur.

2.3. Intelligence Artificielle et Recommandations Personnalisées

- **Analyse des données** de santé pour fournir des recommandations personnalisées basées sur l'historique de santé de l'utilisateur.
- **Prévisions de risques médicaux** pour identifier des maladies potentielles à partir des habitudes de vie et des données collectées.
- Algorithmes de machine learning pour détecter des anomalies dans les données de santé (ex : risques de maladies cardiaques, diabète).

2.4. Téléconsultation

- **Réservation et gestion des consultations** avec des médecins à distance.
- **Consultations vidéo et audio** avec des médecins via une intégration de services tels que **WebRTC** ou **Zoom API**.
- Accès aux données de santé des utilisateurs pour les médecins pendant les consultations.

2.5. Gestion des prescriptions et rappels

- Gestion des **prescriptions médicales** envoyées par les médecins et affichage dans le tableau de bord.
- **Rappels automatisés** pour la prise de médicaments, les consultations et les suivis médicaux.

2.6. Alertes en Temps Réel

- **Alertes en temps réel** en cas d'anomalie détectée dans les données de santé des utilisateurs (ex : fréquence cardiaque anormale).
- Notifications push via **Firebase Cloud Messaging (FCM)** ou autre service.

2.7. Sécurité et Confidentialité des Données

- **Chiffrement des données sensibles** pour garantir la confidentialité des informations médicales des utilisateurs.
- Respect des normes de sécurité comme **OAuth 2.0** et **GDPR** pour la protection des données personnelles.

3. Architecture Technique du Système

3.1. Architecture Microservices

Le projet sera basé sur une architecture de microservices afin d'assurer la scalabilité et la flexibilité. Les services seront découplés et communiqueront entre eux via des API REST ou des événements asynchrones.

- **Microservice d'authentification** (Keycloak) pour la gestion des utilisateurs et de l'authentification.
- **Microservice de gestion des données de santé** pour collecter et stocker les informations de santé.
- **Microservice de recommandations** utilisant l'IA pour fournir des recommandations et prédictions sur la santé.
- **Microservice de téléconsultation** pour gérer les rendez-vous, les consultations en ligne, et l'intégration avec des services tiers comme **WebRTC** ou **Zoom API**.
- **Microservice des prescriptions et rappels** pour gérer les prescriptions des médecins et envoyer des notifications.
- **API Gateway** pour centraliser l'accès aux services backend.

3.2. Base de Données

- **MongoDB** pour la gestion des données utilisateur, des historiques de santé et des informations non structurées.
- **Redis** pour le cache et les sessions en temps réel.
- **PostgreSQL** ou **MySQL** pour la gestion des données de santé structurées si nécessaire.

3.3. Sécurité

- **OAuth 2.0** pour la gestion sécurisée des sessions utilisateurs.
- **JWT** pour l'authentification des utilisateurs et la sécurisation des API.
- **SSL/TLS** pour assurer la sécurité des échanges de données sur le réseau.

3.4. Technologies Utilisées

- **Backend :**
 - **Spring Boot** pour le développement des microservices.
 - **Spring Security** pour la gestion des rôles et de l'authentification.
 - **Spring Cloud** pour la gestion des microservices distribués.
 - **TensorFlow** ou **PyTorch** pour les algorithmes d'IA.
 - **Kafka** ou **RabbitMQ** pour la gestion des événements entre services.
 - **Frontend :**
 - **Angular** pour l'interface utilisateur.
 - **Angular Material** pour les composants UI modernes et responsive.
 - **Leaflet** ou **Google Maps API** pour afficher des données de localisation, si nécessaire.
 - **DevOps :**
 - **Docker** pour la conteneurisation des services.
 - **Kubernetes** pour l'orchestration des microservices.
 - **Jenkins** ou **GitLab CI** pour l'intégration continue et le déploiement automatisé.
-

4. Planning du Projet

4.1. Phase de Conception (2 mois)

- **Réalisation du cahier des charges détaillé.**
- **Conception de l'architecture du système.**
- **Définition des technologies et outils à utiliser.**
- **Conception de la base de données et des API.**
- **Prototype de l'interface utilisateur.**

4.2. Phase de Développement (4-6 mois)

- **Développement des microservices.**
- **Intégration des modèles d'IA** pour les recommandations.
- **Développement de l'interface utilisateur avec Angular.**
- **Intégration de la sécurité avec Keycloak et OAuth 2.0.**
- **Tests unitaires et d'intégration.**

4.3. Phase de Test (2 mois)

- **Tests fonctionnels** sur toutes les fonctionnalités.
- **Tests de performance** pour garantir la scalabilité du système.
- **Tests de sécurité** pour s'assurer de la confidentialité des données.

4.4. Phase de Déploiement et Maintenance

- **Déploiement en production** sur des environnements cloud (ex : AWS, GCP).
 - **Suivi et maintenance** du système pour garantir son bon fonctionnement et résoudre les bugs éventuels.
-

5. Équipe du Projet

- **Chef de projet** : Responsable de la gestion de l'équipe, des délais et du budget.
 - **Développeurs Backend** : Spécialistes en **Spring Boot, microservices, IA**.
 - **Développeurs Frontend** : Spécialistes en **Angular, UI/UX design, responsive design**.
 - **Spécialistes en sécurité** : Gestion de l'authentification, des données sensibles et des normes de sécurité.
 - **Testeurs QA** : Tests de validation des fonctionnalités, performance et sécurité.
 - **DevOps** : Gestion de l'infrastructure, **Docker, Kubernetes**.
-

6. Budget et Ressources

- **Infrastructure cloud** : Coût lié au déploiement sur des services cloud comme **AWS, Google Cloud, ou Azure**.
 - **Licences** : Certains outils comme **Zoom API** ou **WebRTC** peuvent nécessiter une licence.
 - **Outils de développement** : Outils comme **JetBrains, Postman**, ou des services de cloud computing.
-

Ce cahier des charges définit clairement les exigences et la méthodologie de développement de la plateforme de santé personnalisée. Ce projet combinant microservices, IA, et téléconsultation résout un problème de santé majeur en offrant une solution moderne, scalable, et accessible à tous les utilisateurs.