

Cahier de charge

Cahier de charge du Projet iot Pellulier intelligente

Encadré par :

Mr.CHAHBOUNI

Réalisé par :

TAOUFIQ YASSAMINA

MOUNIR LOUBNA

Année universitaire : 2024/2025

Introduction

La gestion des prescriptions médicales constitue un enjeu majeur pour les professionnels de santé, en particulier lorsqu'il s'agit de patients polymédiqués ou atteints de maladies chroniques. Ces derniers rencontrent souvent des difficultés à respecter les horaires de prise, les dosages, ou à distinguer les différents traitements à suivre, ce qui peut entraîner une inefficacité du traitement, voire des complications médicales.

Ce projet s'inscrit dans le cadre d'une démarche d'innovation technologique appliquée à la santé. Il vise à concevoir une solution intelligente, accessible et connectée permettant d'assister les médecins dans la prescription, de guider les patients dans leur traitement, et de suivre en temps réel la prise effective des médicaments à l'aide d'un système embarqué basé sur l'ESP32.

Le présent cahier des charges définit les besoins fonctionnels et techniques du projet, les technologies utilisées, l'architecture globale du système, ainsi que les objectifs de développement et de déploiement.

Département Informatique, Réseaux et Télécommunications (IRT)

1. Problématique :

De nombreux patients, en particulier ceux atteints de maladies chroniques ou polymédiqués (nécessitant la prise de plusieurs médicaments différents), rencontrent des difficultés à suivre correctement leur traitement. Les oublis de prise de médicaments, les erreurs de dosage ou de timing peuvent compromettre l'efficacité du traitement et mettre la santé du patient en danger.

Les médecins, quant à eux, manquent souvent de moyens simples et efficaces pour assurer un suivi en temps réel de l'observance thérapeutique de leurs patients.

2. Contexte:

Dans un contexte où la médecine connectée prend une place de plus en plus importante, il devient essentiel de fournir aux professionnels de santé et aux patients des outils numériques performants pour optimiser la gestion des traitements médicamenteux. Les défis liés à l'observance thérapeutique, notamment chez les patients polymédiqués ou atteints de maladies chroniques, nécessitent des solutions innovantes, accessibles et automatisées.

Le projet consiste à concevoir et développer une **application mobile multiplateforme**, basée sur **Flutter**, destinée à faciliter l'interaction entre médecins et patients autour des prescriptions médicales. L'application est connectée à la **plateforme cloud Firebase**, qui assure la gestion sécurisée des utilisateurs, le stockage centralisé des prescriptions, l'envoi de notifications de rappel, ainsi que la synchronisation en temps réel des données.

Par ailleurs, le système intègre une carte **ESP32** simulée (via Wokwi) permettant de **confirmer physiquement la prise des médicaments** par les patients à l'aide de boutons représentant des compartiments. Ces actions sont automatiquement enregistrées dans la base de données Firebase, ce qui permet un suivi fiable et en temps réel par les médecins.

Ce projet s'inscrit ainsi dans une logique d'amélioration de l'observance des traitements, de digitalisation de la prescription, et de renforcement du lien entre les acteurs du parcours de soins à travers une solution mobile, connectée et intelligente.

3. Objectifs :

- Faciliter la prescription de médicaments par les médecins.
- Permettre aux patients de recevoir des rappels de prise de médicaments.
- Assurer le suivi en temps réel de la prise grâce à une carte ESP32 connectée.
- Centraliser les données médicales dans Firebase pour accessibilité et synchronisation.

3. Fonctionnalités

Espace Médecin

- Connexion / Inscription via Firebase Auth.
- Liste de patients associés.
- Création de prescriptions (médicaments, horaires, durée, compartiment).
- Modification / Suppression de prescriptions.
- Suivi de l'historique de prise des médicaments par patient.

Espace Patient

- Connexion / Inscription via Firebase Auth.
- Consultation des prescriptions reçues.
- Réception de notifications push 10 minutes avant chaque prise.
- Accusé de réception automatique via ESP32.
- Statut de prise : en temps / en retard / oubliée.

Notifications

- Notifications locales programmées avec flutter_local_notifications.
- Activation automatique après récupération des prescriptions.
- Annulation des notifications si prescription modifiée ou supprimée.

Département Informatique, Réseaux et Télécommunications (IRT)**ESP32 – Interaction physique**

- Connexion à Firebase (Realtime Database).
- Envoi d'événements lorsque boutons (A, B, C, D) sont pressés.
- Mise à jour en temps réel dans l'application patient.

4. Architecture Technique

composant	Technologie utilisé
Frontend	Flutter
Backend	Firebase (Auth, Firestore, Realtime Database, Cloud Messaging)
Matériel connecté	ESP32
Languages	Dart (Flutter), C++ (ESP32 - Arduino)
Notifications	flutter_local_notifications, Firebase Messaging

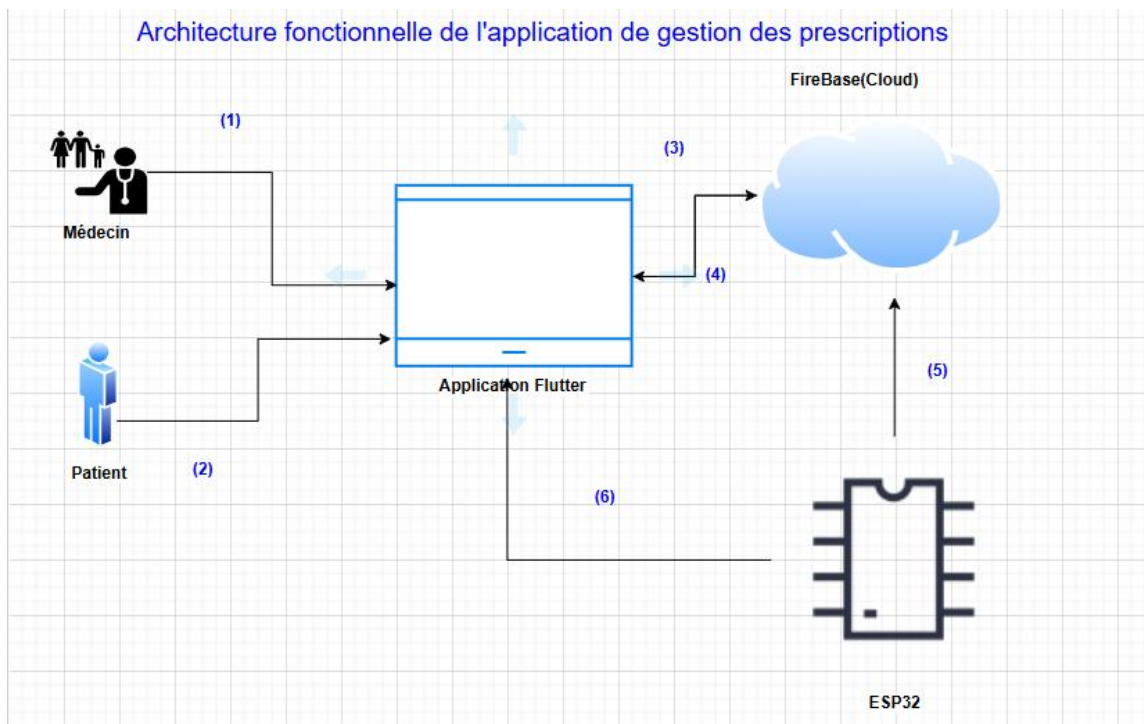
5. Réseau & Cloud**Connexion réseau**

- **ESP32** : Connexion Wi-Fi stable requise pour interagir avec Firebase.
- **Application mobile** : Fonctionne en mode connecté/déconnecté grâce au cache local et à la synchronisation automatique.

Cloud & Stockage

- **Firebase Firestore** : Stockage des prescriptions.
- **Firebase Realtime Database** : Stockage des événements de prise des médicaments.
- **Firebase Authentication** : Gestion sécurisée des utilisateurs.
- **Firebase Cloud Messaging (FCM)** : Envoi des notifications push.

6. Architecture du système :



- (1) Le médecin se connecte à l'application Flutter et crée une prescription.
- (2) Le patient consulte ses prescriptions via l'application mobile.
- (3) L'application Flutter envoie les données de prescription vers Firebase Firestore pour stockage.
- (4) L'application synchronise automatiquement les prescriptions et les notifications depuis Firebase.
- (5) Lors de la prise du médicament, l'ESP32 envoie un signal à Firebase Realtime Database (ex : bouton appuyé).
- (6) L'application lit la base en temps réel et met à jour l'interface patient (état : pris / en retard / oublié)

7. Modèle de Données

Firestore:

```
prescriptions {  
  id: string,  
  patientId: string,  
  medecinId: string,  
  medicament: string,  
  horaires: [ "08:00", "14:00", "20:00" ],  
  compartiment: "A",  
  dateDebut: timestamp,  
  dateFin: timestamp  
}
```

Realtime Database:

```
prises {  
  patientId: {  
    "2025-04-13": {  
      "08:00": { compartiment: "A", pris: true }  
    }  
  }  
}
```

8. Contraintes Techniques

- Compatible Android (minSdk 23+).
- Prise en charge des notifications en arrière-plan.
- Doit fonctionner en mode hors-ligne (cache local + synchro dès que connecté).
- ESP32 doit avoir une connexion stable à Internet pour échanger avec Firebase.

9. Sécurité & Confidentialité

- Authentification via email/mot de passe ou Google.
- Règles Firebase sécurisées par rôles (médecin/patient).
- Données personnelles chiffrées côté client et protégées dans Firebase.

10. Planification

Semaine	Tâches
S1	Maquettes UI + Structure Flutter
S2	Authentification + rôles
S3	Prescription & base de données
S4	Notifications
S5	Intégration ESP32
S6	Tests, débogage et déploiement

Conclusion

Le système proposé vise à offrir une solution numérique complète, connectée et sécurisée pour la gestion des prescriptions médicales, en s'appuyant sur une application mobile Flutter, l'infrastructure cloud Firebase, et un dispositif matériel ESP32. Il répond aux besoins croissants de suivi thérapeutique à distance, de personnalisation des rappels de prise, et d'implication active du patient dans son traitement.

Grâce à une architecture flexible, une prise en charge du mode hors-ligne, des notifications intelligentes et un suivi automatisé, ce projet s'intègre parfaitement dans le contexte de la santé connectée. Il constitue une avancée à la fois technique et sociale en améliorant l'observance thérapeutique et la communication médecin-patient.

Les prochaines étapes consisteront à développer, tester et déployer la solution dans un environnement réel, avec une attention particulière portée à la robustesse, la sécurité des données et l'expérience utilisateur.