

# Documentation Application MEDI-ALERT

Cours : Développement mobile

Etudiant : Nguento Nouba-asra Emmanuel

Master 1 Isoc 2024-2025

## **1. Instructions d'installation et d'exécution**

### **1.1 Prérequis**

Pour exécuter l'application MediAlert, il faut :

- Un smartphone Android (Android 8.0 / API 26 ou supérieur).
- Activer l'installation des applications depuis des sources externes.
- Connexion Internet (pour la première synchronisation avec Firebase).

### **1.2 Installation de l'application (APK)**

1. Récupérer le fichier MediAlert.apk fourni dans le livrable.
2. Copier le fichier sur le smartphone (ou le télécharger directement).
3. Ouvrir le fichier APK.
4. Accepter l'installation.
5. Si un message d'avertissement apparaît, activer :
  - **Paramètres → Sécurité → Sources inconnues / Installer des APK.**

### **1.3 Première exécution**

1. Ouvrir l'application MediAlert depuis le menu d'applications.
2. Accepter les autorisations suivantes :
  - Notifications (obligatoire pour les rappels, surtout sur Android 13+).
  - Appareil photo si vous souhaitez ajouter une photo du médicament.
3. L'écran d'accueil apparaît avec une liste vide de médicaments.

### **1.4 Test rapide de fonctionnement**

1. Appuyer sur le bouton + (Ajouter).
2. Renseigner :
  - Nom : Test Médicament
  - Dose : 1 comprimé
  - Heure : choisir une heure dans 20 à 30 secondes.
3. Enregistrer.
4. Fermer complètement l'application.
5. Attendre la notification.
6. Une alerte doit apparaître même si l'application est fermée.

## **2. Description des fonctionnalités principales**

L'application MediAlert propose plusieurs fonctionnalités essentielles pour le suivi médical quotidien :

### **2.1 Ajout d'un médicament**

L'utilisateur peut enregistrer un médicament en indiquant :

- un nom,
- une dose,
- une heure de prise,
- une photo optionnelle.

### **2.2 Liste des médicaments enregistrés**

La page d'accueil affiche tous les médicaments ajoutés, avec :

- nom,
- dose,
- heure du rappel.

### **2.3 Notifications programmées (fonctionnalité native Android)**

MediAlert utilise des notifications locales programmées, permettant de :

- recevoir un rappel à l'heure exacte,
- déclencher l'alerte même si l'application est fermée,
- rediriger vers la fiche du médicament en cliquant sur la notification.
- Grâce au mode alarmClock, les rappels sont fiables et prioritaires.

### **2.4 Synchronisation en ligne (service externe Firebase)**

Chaque médicament est automatiquement synchronisé dans Firebase Firestore :

- sauvegarde dans le cloud,
- protection des données,

### **2.5 Suppression**

L'utilisateur peut :

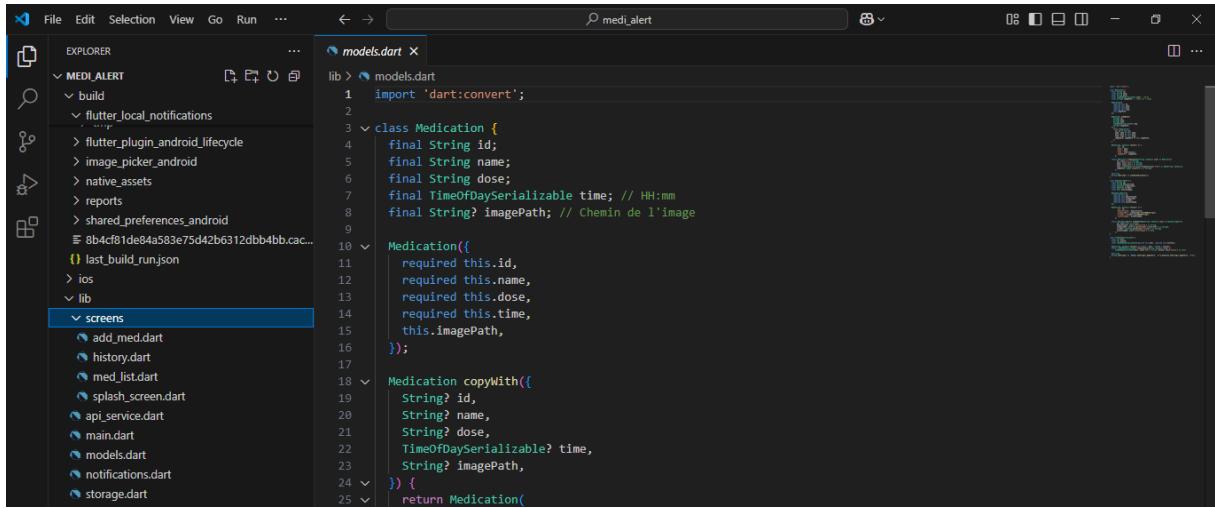
- supprimer un médicament.

### 3. Architecture technique et bibliothèques utilisées

#### 3.1 Architecture technique

L'application est construite avec Flutter et suit une architecture modulaire organisée.

Structure du projet



The screenshot shows a code editor with the following details:

- File Explorer:** Shows the project structure under "MEDI\_ALERT". It includes "build", "lib" (which contains "flutter\_local\_notifications", "image\_picker\_android", "native\_assets", "reports", "shared\_preferences\_android", and "models.dart"), and "screens" (which contains "add\_med.dart", "history.dart", "med\_list.dart", "splash\_screen.dart", "api\_service.dart", "main.dart", "models.dart", "notifications.dart", and "storage.dart").
- Code Editor:** Displays the content of "models.dart". The code defines a class "Medication" with properties: id, name, dose, time, and imagePath. It also includes a copyWith method.

```
lib > models.dart
1 import 'dart:convert';
2
3 <> class Medication {
4   final String id;
5   final String name;
6   final String dose;
7   final TimeOfDaySerializable time; // HH:mm
8   final String? imagePath; // Chemin de l'image
9
10 Medication({
11   required this.id,
12   required this.name,
13   required this.dose,
14   required this.time,
15   this.imagePath,
16 });
17
18 Medication copyWith({
19   String? id,
20   String? name,
21   String? dose,
22   TimeOfDaySerializable? time,
23   String? imagePath,
24 }) {
25   return Medication(
```

#### 3.2 Bibliothèques utilisées

Flutter (Dart)

Framework principal pour l'interface et la logique métier.

Notifications locales

flutter\_local\_notifications

timezone

Utilisés pour :

initialiser les canaux de notifications,

programmer les rappels,

faire fonctionner les notifications même si l'application est fermée.

Service de stockage : SharedPreferences

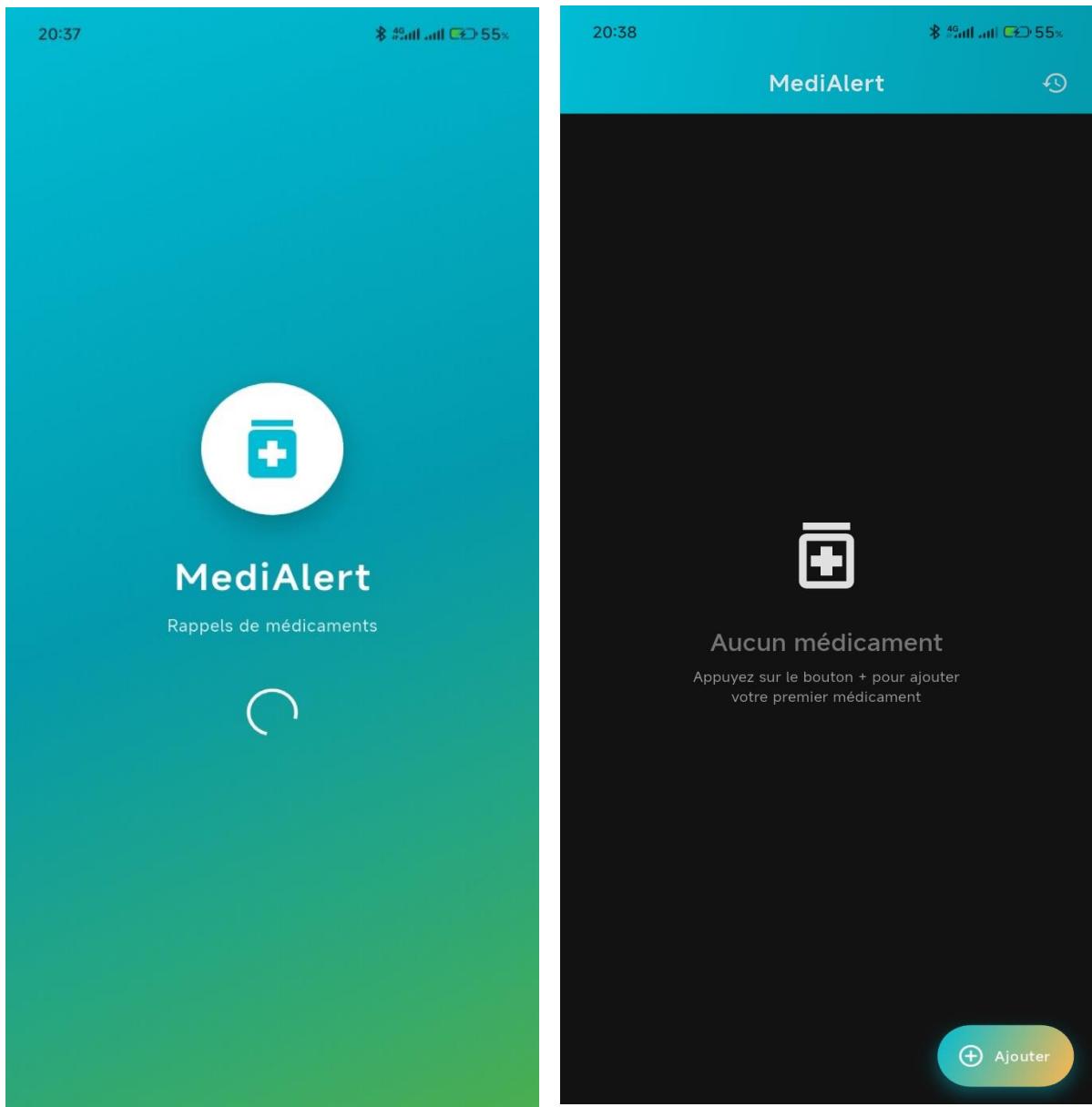
Utilisés pour :

stocker les médicaments.

Autres dépendances potentielles

image\_picker : pour prendre une photo du médicament

#### 4. Aperçu de l'interface de l'application :



**Nouveau médicament**

21:56 62%

Nom du médicament  
Ex: Paracétamol

Dose  
Ex: 500mg, 1 comprimé

Photo du médicament  
Appuyez pour prendre une photo

Heure du rappel  
08:00 Matin >

✓ Enregistrer

20:38 55%

**MediAlert**

2 Médicaments | 2 Rappels actifs

Paracétamol  
20:35 500mg

Doliprane  
20:37 500mg

Ajouter

20:38

Bluetooth 4G 55%



## Historique



### Aucun rappel

L'historique de vos rappels  
apparaîtra ici