

Voici une liste de logiciels gratuits que vous pouvez utiliser pour simuler la détection de la prise de médicament dans votre projet sans avoir à acheter de capteurs physiques. Ces outils vous permettront de créer des interactions logicielles dans l'application pour simuler la prise de médicament :

1. Flutter (pour l'application mobile)

- **Utilisation** : Développer l'application mobile qui permet de simuler la prise de médicaments via des interactions de boutons et de notifications.
- **Fonctionnalité** : Vous pouvez utiliser des boutons et des fonctionnalités de gestion d'état dans Flutter pour simuler l'action de retrait d'un médicament.
- **Outils spécifiques** :
 - **flutter_local_notifications** : Pour afficher des notifications de rappels.
 - **flutter_bloc** ou **provider** : Pour gérer l'état de l'application et simuler les interactions de l'utilisateur.
- **Lien** : [Flutter](#)

2. Firebase (base de données et notifications)

- **Utilisation** : Firebase peut être utilisé pour stocker les données des prises et envoyer des notifications en fonction de l'interaction du patient.
- **Fonctionnalité** : Vous pouvez enregistrer chaque action du patient dans Firebase et utiliser Firebase Cloud Messaging pour envoyer des rappels.
- **Outils spécifiques** :
 - **Firestore** : Pour stocker l'historique des prises.
 - **Firebase Cloud Messaging** : Pour envoyer des rappels de prise.
- **Lien** : [Firebase](#)

3. Figma (prototypage UI/UX)

- **Utilisation** : Pour créer des maquettes interactives de l'interface utilisateur, y compris les boutons de prise de médicament.

- **Fonctionnalité** : Figma permet de simuler des interactions utilisateur sans avoir besoin de coder. Vous pouvez créer des prototypes cliquables pour simuler la détection de la prise.
- **Outils spécifiques** : Créer des boutons qui, lorsqu'ils sont cliqués, simulent la prise du médicament.
- **Lien** : [Figma](#)

4. MIT App Inventor (développement d'applications)

- **Utilisation** : Si vous cherchez une solution plus simple pour créer l'application sans coder, MIT App Inventor permet de développer des applications mobiles avec une interface graphique.
- **Fonctionnalité** : Vous pouvez créer une application avec des boutons pour simuler la prise de médicaments et enregistrer les actions des patients.
- **Outils spécifiques** : Interface drag-and-drop pour créer facilement des boutons et gérer les interactions.
- **Lien** : [MIT App Inventor](#)

5. Thymeleaf (pour générer des interfaces Web)

- **Utilisation** : Si vous préférez développer une application web pour le suivi des médicaments, Thymeleaf est un moteur de templates pour Java permettant de générer dynamiquement des pages HTML.
- **Fonctionnalité** : Vous pouvez utiliser Thymeleaf pour créer une interface web simple avec des boutons interactifs pour simuler la prise de médicament.
- **Outils spécifiques** : Vous pouvez intégrer Thymeleaf avec Spring Boot pour créer un backend léger.
- **Lien** : [Thymeleaf](#)

6. Blynk (IoT sans matériel)

- **Utilisation** : Bien que Blynk soit conçu pour l'IoT, il permet aussi de simuler des capteurs sans avoir besoin de matériel physique. Vous pouvez créer une interface où un bouton simule l'action de prendre un médicament.

- **Fonctionnalité** : L'application Blynk peut être utilisée pour créer des boutons virtuels, ce qui vous permet de simuler la détection d'une prise de médicament.
- **Outils spécifiques** : Utilisation de l'interface de contrôle et de simulation dans Blynk.
- **Lien** : [Blynk](#)

7. Scratch (pour des prototypes simples)

- **Utilisation** : Scratch est un outil visuel qui permet de créer des prototypes simples. Bien qu'il soit principalement destiné aux enfants, il peut être utilisé pour créer des prototypes d'interactions simples sans coder.
- **Fonctionnalité** : Créez une simulation de prise de médicament via des événements et des actions, sans nécessiter de codage complexe.
- **Lien** : [Scratch](#)

Comment les utiliser dans votre projet ?

1. **Créer l'interface** : Utilisez **Flutter** pour créer une application mobile où l'utilisateur pourra appuyer sur un bouton pour simuler la prise du médicament.
2. **Simuler la détection** : À chaque fois qu'un bouton est pressé dans l'application, utilisez **Firebase** pour stocker cette action et mettre à jour l'historique des prises.
3. **Envoyer des rappels** : Utilisez **Firebase Cloud Messaging** pour envoyer des notifications de rappels de prise, ou utilisez **flutter_local_notifications** pour envoyer des rappels dans l'application.
4. **Prototypage d'interfaces** : Si nécessaire, utilisez **Figma** pour créer des prototypes interactifs et tester le flux de l'application avant de passer au développement réel.

Ces outils gratuits vous permettront de créer une simulation efficace de la prise de médicament, tout en permettant un suivi et une gestion des données sans nécessiter d'équipement coûteux.

Vous avez dit :

c'est bon redonnez moi le rapport

ChatGPT a dit :

Voici la version mise à jour du rapport, incluant les outils gratuits et la simulation de la détection de prise de médicament :

Rapport de Projet : Boîte de Médicaments Intelligente avec Application Flutter, IoT et Stockage de Données

1. Introduction

Les personnes âgées et les patients chroniques oublient souvent de prendre leurs médicaments, ce qui peut entraîner des complications de santé graves. Ce projet propose une solution innovante en combinant une boîte de médicaments intelligente divisée en 7 compartiments (un pour chaque jour de la semaine), une application mobile Flutter, l'Internet des Objets (IoT), un système de stockage et traitement de données en ligne, et un suivi personnalisé par le médecin.

2. Objectifs du projet

- Aider les patients à suivre leurs prescriptions médicales grâce à des rappels automatisés.
- Permettre aux médecins de suivre à distance l'adhérence du patient au traitement.
- Offrir un système de collecte, traitement et visualisation des données.
- Donner la possibilité aux médecins de faire des recommandations personnalisées à partir des données enregistrées.
- Permettre aux médecins d'ajouter et de gérer le traitement de leurs patients via l'application.
- Détecter automatiquement la prise de médicament grâce à la boîte intelligente sans interaction manuelle du patient.

3. Architecture générale du système

- **Application Flutter** : interface patient et interface médecin.

- **Firestore** : Authentification, Firestore (base de données temps réel), stockage des historiques, notifications via Firebase Cloud Messaging.
 - **Backend Python (Flask/FastAPI)** : pour le traitement des données et gestion des recommandations.
 - **Recommandations médicales** : suggestions manuelles par le médecin selon les données collectées.
 - **Simulation de détection de prise** : prototypage logiciel pour compenser l'absence de composants physiques. La version finale prévoit un capteur (ex. : capteur de poids ou capteur infrarouge) dans chaque compartiment pour détecter automatiquement si la pilule a été retirée.
-

4. Fonctionnalités principales

Côté Patient :

- Réception du traitement prescrit par le médecin (ajouté via l'application médecin).
- Alarmes sonores et notifications pour chaque prise.
- Détection automatique de la prise via la boîte intelligente (pas besoin de confirmation manuelle).
- Historique des prises visible par le patient.

Côté Médecin :

- Ajout du traitement à suivre pour chaque patient : nom du médicament, dose, heure, durée.
 - Accès à l'historique des prises de chaque patient.
 - Alertes si le patient oublie plusieurs fois ses médicaments.
 - Possibilité de faire des recommandations personnalisées en fonction de l'historique.
-

5. Détails techniques

Application Flutter

- **Packages utilisés :** `flutter_local_notifications` , `cloud_firestore` , `firebase_auth` , `firebase_messaging`
- Fonction de planification des notifications pour chaque médicament quotidiennement.
- Deux interfaces : médecin et patient.

Backend Python

- **Librairies :** Flask, pandas
- Fonctionnalités :
 - Traitement des historiques
 - Interface médecin pour recommandations
 - Envoi d'alertes médecin
 - API sécurisée pour la gestion des traitements

Simulation de détection de prise

- **Prototypage logiciel :** Plutôt que d'utiliser des capteurs physiques coûteux, nous simulons la détection via des interactions dans l'application. Par exemple, lorsqu'un patient "retire" un médicament, un bouton dans l'application permet de simuler la détection de la prise.
 - **Outils recommandés :**
 - **Flutter** pour la simulation d'interactions dans l'application.
 - **Firestore** pour enregistrer les actions dans la base de données en temps réel.
 - **Figma** pour prototyper les interfaces et simuler les interactions avant la programmation.

6. Scénario d'utilisation

1. Le médecin se connecte à l'application et ajoute le traitement pour son patient.
2. Le patient reçoit automatiquement le planning de prise dans son interface.
3. L'application programme les alarmes pour chaque médicament.

4. Le jour J, l'alarme se déclenche : la boîte détecte automatiquement si le médicament a été retiré.
 5. Les données sont envoyées au backend et stockées dans Firebase.
 6. Le médecin peut visualiser les oublis et ajouter des recommandations personnalisées.
-

7. Avantages du projet

- Amélioration de la santé des patients.
 - Réduction des hospitalisations dues aux oublis.
 - Détection automatique sans interaction manuelle, adaptée aux personnes analphabètes.
 - Suivi médical intelligent sans matériel coûteux dans la version prototype.
 - Facilite le travail des médecins avec un outil de prescription et de suivi.
 - Facilement extensible avec des dispositifs IoT physiques plus tard.
-

8. Limitations et perspectives

- Pas de matériel IoT réel dans cette version (possibilité future : ESP32 + capteurs).
 - Recommandations dépendantes de l'implication du médecin.
 - Interface médecin améliorable (dashboard web par exemple).
-

9. Conclusion

Ce projet propose une solution digitale innovante et accessible pour améliorer la prise de médicaments, en utilisant intelligemment des technologies modernes : Flutter pour l'application, Firebase pour le stockage cloud, et le traitement des données pour le suivi. Le médecin joue un rôle central en prescrivant les traitements et en suivant les patients, ce qui garantit un accompagnement personnalisé. Le système peut facilement être amélioré par l'ajout de capteurs IoT permettant de détecter automatiquement les prises, rendant le dispositif encore plus adapté aux personnes âgées ou analphabètes.

10. Annexes

- Schémas de la base de données Firestore
- Exemple de données historiques (CSV)
- Capture d'écran de l'application Flutter
- Exemple de script Python pour gestion des données