



Université Cadi Ayyad  
École Supérieure De Technologie-Safi  
Département : Informatique  
Option : Ingénierie des Systèmes d'information et Réseaux

Rapport de Stage de Fin d'Études

---

# Application de Suivi de la Tension Artérielle avec Intégration de l'IA.

---

Réalisé par :  
REKHLA Zakaria

Encadré par :  
M.EL ABDELLAOUI SAID  
Mme. KACHBAL ILHAM



# Dédicaces

*Je dédie ce projet :*

À mes très chers parents, pour leur amour, leurs sacrifices et leur soutien inconditionnel.

À mes adorables frères et sœurs, pour leurs encouragements et leurs conseils.

À mes proches amis et à toute notre grande famille, pour leur aide.

À toutes les personnes que je connais de près ou de loin.

# **Remerciements**

Avant de commencer mon travail, je tiens à exprimer ma profonde gratitude envers les personnes qui nous ont apporté leur soutien durant cette période. Je tiens également à remercier tout particulièrement. Monsieur ELABDELLAOUI Said, mon professeur et mon encadrant, pour son accompagnement, son soutien, les précieux renseignements qu'il nous a fournis et son encouragement tout au long de ce projet. Madame Kachbal Ilham, qui n'a cessé de m'encourager pendant la durée du projet, ainsi que pour sa générosité en matière de formation et d'encadrement.

Merci à tout le corps enseignant du département Informatique de l'Ecole Supérieure de Technologie de SAFI.

# Résumé

Notre projet vise à développer une application mobile avancée pour le suivi de la tension artérielle, intégrant des technologies modernes et l'intelligence artificielle pour une surveillance optimale de la santé. L'application mobile, développée avec Flutter, permettra aux patients de suivre leur tension artérielle, d'enregistrer les mesures, de visualiser l'évolution des données, et de recevoir des alertes et des recommandations personnalisées basées sur des prédictions d'IA. En parallèle, l'application web développée avec Next.js offrira aux médecins un accès au back-office pour gérer les données des patients, analyser les résultats grâce à l'IA, et établir des diagnostics et recommandations de traitement. Ce projet ambitionne de révolutionner la gestion de la tension artérielle en fournissant des outils précis et des recommandations personnalisées, améliorant ainsi la qualité de vie des utilisateurs.

# **Abstract**

Our project aims to develop an advanced mobile application for blood pressure monitoring, integrating modern technologies and artificial intelligence for optimal health surveillance. The mobile application, developed with Flutter, will enable patients to track their blood pressure, record measurements, view data trends, and receive alerts and personalized recommendations based on AI predictions. Concurrently, the web application developed with Next.js will provide doctors with access to a back-office to manage patient data, analyze results using AI, and establish diagnoses and treatment recommendations. This project aims to revolutionize blood pressure management by providing precise tools and personalized recommendations, thereby enhancing the quality of life for users.

## ملخص

يهدف مشروعنا إلى تطوير تطبيق جوال متقدم لمراقبة ضغط الدم، باستخدام تقنيات حديثة والذكاء الاصطناعي لتحقيق مراقبة صحيحة مثل. يمكن تطبيق الجوال، الذي يتم تطويره باستخدام Flutter، من متابعة ضغط الدم، تسجيل القياسات، عرض تطور البيانات، وتلقي التنبهات والتوصيات الشخصية بناءً على تنبؤات الذكاء الاصطناعي. في الوقت نفسه، سيوفر تطبيق الويب المطور باستخدام Next.js للأطباء إمكانية الوصول إلى مكتب خلفي لإدارة بيانات المرضى، تحليل النتائج باستخدام الذكاء الاصطناعي، وإصدار التشخيصات والتوصيات العلاجية. يهدف هذا المشروع إلى إحداث ثورة في إدارة ضغط الدم من خلال توفير أدوات دقيقة وتوصيات مخصصة، مما يعزز جودة حياة المستخدمين.

# Table des matières

<b>Dédicaces</b>	I
<b>Remerciements</b>	II
<b>Résumé</b>	III
<b>Abstract</b>	IV
<b>ملخص</b>	V
<b>Introduction Générale</b>	1
<b>1 Présentation de l'organisme d'accueil :</b>	2
1.1 Introduction : . . . . .	3
1.2 Choix de l'entreprise : . . . . .	3
1.3 Présentation de l'Entreprise Norsys Afrique . . . . .	3
1.3.1 Identité de Norsys Afrique : . . . . .	3
1.3.2 Activité de Norsys Afrique : . . . . .	4
1.3.3 Métiers de Norsys Afrique : . . . . .	4
1.3.4 Répartition des collaborateurs : . . . . .	4
1.4 Conclusion : . . . . .	5
<b>2 Cahier des Charges et Méthodologie suivie</b>	6
2.1 Introduction : . . . . .	7
2.2 Description du Projet : . . . . .	7
2.2.1 Problématique . . . . .	7
2.2.2 Objectif . . . . .	7
2.3 Cahier de Charge . . . . .	7
2.3.1 Besoins Fonctionnels : . . . . .	7

2.3.2 Besoins Non Fonctionnels : . . . . .	9
2.4 Méthodologie suivi : . . . . .	9
2.4.1 Méthode Agile : . . . . .	9
2.4.2 Méthode SCRUM : . . . . .	9
2.5 Conclusion : . . . . .	10
<b>3 Etude Conceptuel</b>	<b>11</b>
3.1 Introduction : . . . . .	12
3.2 Conception UML : . . . . .	12
3.3 Diagramme de cas d'utilisation : . . . . .	12
3.4 Diagramme de Classe : . . . . .	15
3.5 BPMN (Business Process Model and Notation) : . . . . .	16
3.6 Conclusion : . . . . .	17
<b>4 L'Architecture Globale du Système</b>	<b>18</b>
4.1 Introduction . . . . .	19
4.2 Description de mon Système . . . . .	19
4.2.1 Architecture de Back-end . . . . .	21
4.2.2 Architecture de Front-end . . . . .	21
4.3 Intégration de Firebase . . . . .	22
4.4 Modèle OCR de Hugging Face : . . . . .	23
4.4.1 OCR : . . . . .	23
4.4.2 Fonctionnement du Modèle OCR de Hugging Face . . . . .	23
4.4.3 Description du Modèle OCR-Donut-CORD . . . . .	23
4.4.4 Intégration du Script Python avec Laravel en utilisant LaravelPython .	24
4.5 Conclusion . . . . .	24
<b>5 Technologie et les Outils Utilisés</b>	<b>25</b>
5.1 Introduction : . . . . .	26
5.1.1 Technologies et Outils Utilisés . . . . .	26
5.2 Conclusion . . . . .	30
<b>6 Réalisation du projet :</b>	<b>31</b>
6.1 Introduction : . . . . .	32
6.2 Développement de l'application mobile . . . . .	32
6.2.1 Système de Connexion . . . . .	32

6.2.2	Page d'Accueil . . . . .	33
6.3	Développement de l'application Web . . . . .	38
6.3.1	Page d'Accueil . . . . .	38
6.3.2	Page Patient . . . . .	39
6.3.3	Page de Chat . . . . .	41
6.3.4	Rendez-vous . . . . .	41
6.4	Conclusion . . . . .	42
<b>Conclusion</b>		<b>43</b>
<b>Reference</b>		<b>44</b>

# Table des figures

1.1	Logo officiel de l'Entreprise Norsys Afrique . . . . .	3
2.1	SCRUM . . . . .	9
2.2	KanBan . . . . .	10
3.1	UML Logo . . . . .	12
3.2	– Diagramme de cas d utilisation. . . . .	13
3.3	– Diagramme de classe pour MongoDB. . . . .	15
3.4	– Diagramme BPMN de processus d'authentification . . . . .	16
3.5	– Diagramme BPMN de model OCR. . . . .	17
4.1	Architecture globale du système. . . . .	20
4.2	Architecture de système. . . . .	20
4.3	Architecture de Back-End. . . . .	21
4.4	Architecture de Front-end. . . . .	22
4.5	Donut Architecture. . . . .	24
5.1	Atlas MongoDB . . . . .	26
5.2	Laravel . . . . .	26
5.3	Next.js . . . . .	26
5.4	Flutter . . . . .	27
5.5	Tailwind CSS. . . . .	27
5.6	TypeScript . . . . .	27
5.7	HTML . . . . .	27
5.8	CSS . . . . .	27
5.9	JWT (JSON Web Tokens) . . . . .	28
5.10	Postman . . . . .	28
5.11	VSCode . . . . .	28

---

5.12 Émulateur Android . . . . .	28
5.13 Hugging Face . . . . .	29
5.14 OpenAI . . . . .	29
5.15 Astah UML . . . . .	29
5.16 Notion . . . . .	29
5.17 Git . . . . .	30
5.18 Github . . . . .	30
6.1 – Screenshots of login-related functionalities. . . . .	32
6.2 – Page d'Accueil. . . . .	33
6.3 – Gestion des Rendez-Vous. . . . .	34
6.4 – Notifications. . . . .	35
6.5 – Screenshots of Blood Pressure interface. . . . .	35
6.6 – BMI. . . . .	36
6.7 – Chat. . . . .	37
6.8 – Screenshots of Medicaments interface. . . . .	38
6.9 – Page d'Accueil de l'application web. . . . .	39
6.10 – Liste des Patients de l'application web. . . . .	40
6.11 – Ajout de Patient. . . . .	40
6.12 – Page de Chat de l'application web. . . . .	41
6.13 – Page de Rendez-vous de l'application web. . . . .	42

# Liste des sigles et acronymes

<b>IA</b>	<i>Internet Protocol</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>OCR</b>	<i>Optical character recognition or optical character reader</i>
<b>JSON</b>	<i>JavaScript Object Notation</i>
<b>REST</b>	<i>Representational State Transfer</i>
<b>HTTP</b>	<i>HyperText Transfer Protocol</i>
<b>UML</b>	<i>Unified Modeling Language</i>
<b>BPMN</b>	<i>Business Process Model and Notation</i>

# Introduction Générale

Notre projet vise à développer une application mobile avancée pour le suivi de la tension artérielle, intégrant des technologies modernes et l'intelligence artificielle pour une surveillance optimale de la santé. Pour ce faire, nous avons choisi d'utiliser des technologies de pointe, notamment Flutter pour le développement de l'application mobile et Next.js pour l'application web, sans recourir à Laravel pour le backend. L'objectif principal est de fournir une solution complète et intuitive permettant aux utilisateurs de surveiller leur tension artérielle, de prédire les fluctuations potentielles et de recevoir des recommandations personnalisées basées sur les données recueillies.

Le projet se divise en deux volets principaux pour répondre aux besoins des patients et des médecins. Pour les patients, l'application mobile développée avec Flutter fournira une interface conviviale pour le suivi de la tension artérielle, permettant l'enregistrement des mesures, la visualisation de l'évolution des données et la réception d'alertes et de recommandations personnalisées basées sur des prédictions d'IA. Pour les médecins, le back-office accessible via l'application web développée avec Next.js permettra de gérer les données des patients, d'accéder à des analyses détaillées fournies par l'IA et d'utiliser des outils avancés pour établir des diagnostics et recommander des traitements appropriés.

En conclusion, ce projet ambitionne de révolutionner la manière dont les utilisateurs surveillent et gèrent leur tension artérielle en intégrant des technologies avancées et une interface utilisateur optimisée. En offrant des outils précis et des recommandations personnalisées, nous espérons améliorer significativement la qualité de vie des utilisateurs en leur permettant de prendre en charge leur santé de manière autonome et éclairée.

## **Section 1:**

**Présentation de l'organisme d'accueil :**

## 1.1 Introduction :

Ce chapitre sera consacré à la présentation de l'entreprise **Norsys Afrique**, où j'ai effectué mon stage. J'y détaillerai l'identité de l'entreprise, ses métiers et ses activités principales. Cette présentation fournira un contexte essentiel pour comprendre l'environnement dans lequel j'ai travaillé ainsi que les contributions spécifiques que j'ai pu apporter au cours de mon stage.

## 1.2 Choix de l'entreprise :

J'ai choisi **Norsys Afrique** pour mon stage de fin d'études en raison de son influence dans des secteurs qui m'attirent vivement. En tant que filiale du groupe Norsys, cette grande entreprise se distingue par ses projets de développement pour de grands clients. Je trouve fascinant le rôle crucial que jouent ces entreprises en concevant des sites web et en fournissant des solutions informatiques.

Cette dimension de **Norsys Afrique** suscite mon intérêt et je suis impatient d'en apprendre davantage sur la manière dont elle intègre la technologie pour répondre aux besoins de ses clients.

## 1.3 Présentation de l'Entreprise Norsys Afrique

Créée en 1996 à Marrakech, **Norsys Afrique**, filiale du groupe **Norsys**, s'est spécialisée dans la réalisation de développements orientés sur les nouvelles technologies. **Norsys Afrique** est ainsi devenue un centre de développement du groupe Norsys.

Cette entreprise assume autant de responsabilités économiques que sociales et sociétales. Elle est reconnue pour son dynamisme et ses actions opérationnelles dans le domaine sociétal et humanitaire, visant à favoriser l'épanouissement professionnel et personnel de ses salariés.



FIG. 1.1 – Logo officiel de l'Entreprise Norsys Afrique

### 1.3.1 Identité de Norsys Afrique :

**Norsys Afrique** est une ESN engagée, comptant plus de 600 collaborateurs (appelés easymarkers), créée en 1994. Elle est spécialisée dans le conseil en assistance à maîtrise d'ouvrage

et l'ingénierie informatique sur mesure, utilisant les dernières technologies (JAVA JEE, PHP, Angular, Nodejs...).

En 2019, **Norsys** a été certifiée B Corp, ce qui démontre la réalité de son engagement sur les plans social et environnemental, et intègre le cercle pionnier d'entreprises internationales très impliquées.

### 1.3.2 Activité de Norsys Afrique :

- **JAVA JEE en chiffres** : 19 projets, 33 collaborateurs au sein de la communauté JAVA, 36 % de nos projets sont en Angular et 80 % des projets suivent une approche DevOps.
- **PHP en chiffres** : 13 projets, 15 collaborateurs dont 3 pilotes, 50 % de nos projets sont en Drupal, 24 % en Magento et 15 % en Symfony.

### 1.3.3 Métiers de Norsys Afrique :

- **Consultant AMOA H/F** : Assiste à la maîtrise d'ouvrage en définissant les besoins et en pilotant les projets pour s'assurer de leur bonne réalisation.
- **Pilote de projet H/F** : Coordonne et gère les différentes étapes d'un projet pour garantir le respect des délais, du budget et de la qualité.
- **Développeur JEE / PHP / FRONT/ .NET / FULL STACK H/F** : Conçoit, développe et maintient des applications en utilisant diverses technologies et frameworks.
- **Expert Technique H/F** : Fournit une expertise approfondie sur des technologies spécifiques et aide à résoudre des problèmes techniques complexes.
- **Ingénieur d'affaire H/F** : Gère les relations commerciales, identifie les opportunités d'affaires et s'assure de la satisfaction des clients.
- **PIRATE H/F** : Accompagne les collaborateurs dans leurs projets en agissant comme un mentor ou un coach technique.
- **Product Owner H/F** : Définit et priorise les fonctionnalités d'un produit pour maximiser la valeur apportée aux utilisateurs et aux parties prenantes.

### 1.3.4 Répartition des collaborateurs :

La maîtrise des technologies PHP, JavaScript, jQuery, Bootstrap et des méthodes DevOps est essentielle pour tous les collaborateurs du pôle PHP.

Les collaborateurs se spécialisent ensuite dans les technologies suivantes : Magento, Drupal ou Symfony, ce dernier représentant le framework de base de ces deux CMS.

## 1.4 Conclusion :

Dans ce chapitre introductif, nous avons présenté l'entreprise **Norsys Afrique**, ce qui nous a offert une vision détaillée du cadre général de l'entreprise, de son identité, de ses métiers et de ses activités. Cette compréhension contextuelle est essentielle pour appréhender le cadre de mon stage et les contributions spécifiques que j'ai pu y apporter. Dans la partie suivante, nous allons nous concentrer sur la présentation du projet, en détaillant ses objectifs, ses enjeux et les différentes étapes de sa réalisation.

## **Section 2:**

# **Cahier des Charges et Méthodologie**

## **suivi**

## 2.1 Introduction :

Ce cahier des charges vise à définir les spécifications pour le développement d'une application mobile de suivi de la tension artérielle avec intégration de l'intelligence artificielle. L'objectif est de créer une application qui permettra aux utilisateurs de surveiller leur tension artérielle de manière efficace, de prédire les fluctuations et de fournir des recommandations personnalisées basées sur les données recueillies.

## 2.2 Description du Projet :

### 2.2.1 Problématique

La gestion de la tension artérielle est essentielle pour éviter des complications graves comme les maladies cardiaques et les accidents vasculaires cérébraux. La tension artérielle est souvent appelée la "maladie silencieuse" car elle peut ne montrer aucun symptôme tout en causant des dommages importants. Beaucoup de gens ont du mal à surveiller leur tension régulièrement et à comprendre les variations de leurs mesures. Les méthodes traditionnelles de suivi peuvent être décourageantes, et les patients manquent souvent de conseils personnalisés adaptés à leurs données spécifiques. En outre, certaines personnes ne savent pas comment interpréter les chiffres de leur tension artérielle, ce qui les rend inconscients des risques potentiels liés à des valeurs anormales.

### 2.2.2 Objectif

L'objectif de ce projet est de développer une application mobile intuitive et fonctionnelle pour le suivi de la tension artérielle, intégrant des fonctionnalités d'intelligence artificielle (IA) pour analyser les données et prédire les épisodes de tension élevée ou basse. En plus de fournir une plateforme de suivi, l'application offrira des recommandations personnalisées basées sur les prédictions de l'IA, aidant les utilisateurs à maintenir une tension artérielle saine grâce à des conseils adaptés à leur profil et historique de santé. Nous visons à permettre aux utilisateurs de gérer leur santé de manière proactive en comprenant les facteurs influençant leur tension artérielle et en prenant des mesures préventives. Le projet se divise en deux parties : une interface conviviale pour les patients pour enregistrer et suivre leurs données et recevoir des recommandations, et un back-office pour les médecins pour gérer les patients, accéder à l'historique des tensions artérielles, analyser les données avec l'IA, et recommander des traitements appropriés.

## 2.3 Cahier de Charge

### 2.3.1 Besoins Fonctionnels :

#### Enregistrement de la Tension Artérielle :

- Permettre aux utilisateurs d'enregistrer leur tension artérielle en saisissant manuellement les valeurs de pression systolique et diastolique.
- Autoriser la prise de photo des résultats de tension artérielle par les tensiomètres pour extraire les valeurs directement dans l'application en utilisant l'intelligence artificielle.
- Afficher les données de tension artérielle sous forme de graphiques pour visualiser les tendances au fil du temps.

### **Suivi de Dossier Médical**

- Offrir aux utilisateurs la possibilité de stocker leur historique de tension artérielle et d'autres informations médicales pertinentes.
- Permettre aux utilisateurs d'ajouter des notes ou des commentaires à leurs enregistrements de tension artérielle.

### **Gestion de Médicaments**

- Permettre aux utilisateurs d'enregistrer les médicaments qu'ils prennent pour contrôler leur tension artérielle.
- Programmer des rappels pour les prises de médicaments. L'intelligence artificielle recommandera des rappels aux moments les plus efficaces pour contrôler la tension artérielle.

### **Suivi de l'Activité Physique et de l'Alimentation**

- Permettre aux utilisateurs d'enregistrer leur activité physique et leur alimentation.
- Intégrer l'analyse de l'impact de l'activité physique et de l'alimentation sur la tension artérielle.

### **Analyse de l'Intelligence Artificielle**

- Développer un modèle prédictif pour anticiper les épisodes de tension artérielle élevée ou basse.
- Mettre en œuvre des recommandations personnalisées basées sur les prédictions de l'intelligence artificielle pour aider les utilisateurs à maintenir une tension artérielle saine.
- Adapter les recommandations en fonction des préférences de l'utilisateur et des données de suivi.

### **Analyse des Données et Diagnostic**

- Utiliser les outils d'analyse de l'application pour interpréter les données de tension artérielle.

### **Communication avec les Patients**

- Communiquer avec les patients pour discuter de leurs résultats de tension artérielle et de leur état de santé général.
- Fournir des conseils et des recommandations personnalisées pour améliorer le contrôle de la tension artérielle.
- Envoyer des rappels de rendez-vous et des notifications de suivi aux patients.

### **Gestion des Rendez-vous et des Dossiers Médicaux**

- Planifier des rendez-vous avec les patients pour des consultations et des suivis.
- Enregistrer les informations des patients, y compris leurs antécédents médicaux, leurs allergies et leurs médicaments.
- Mettre à jour les dossiers médicaux des patients avec les résultats des consultations et des examens.

## 2.3.2 Besoins Non Fonctionnels :

- **Performance** : L'application doit offrir des temps de réponse rapides, même lorsqu'elle gère un grand nombre de requêtes simultanées.
- **Sécurité** : Toutes les transactions financières doivent être sécurisées, en utilisant des méthodes de chiffrement.
- **Évolutivité** : Le système doit être conçu de manière à pouvoir évoluer facilement pour gérer une croissance future, que ce soit en termes de nombre d'utilisateurs, de données ou de fonctionnalités.
- **Soutenabilité** : Le code source et l'architecture doivent être conçus de manière à être maintenables et évolutifs dans le temps.

## 2.4 Méthodologie suivie :

### 2.4.1 Méthode Agile :

La méthode AGILE est une méthodologie de gestion de projet. Il existe en réalité plusieurs méthodes qui ont toutes un point commun : elles découlent du Manifeste Agile. Édité en 2001, le Manifeste Agile a été développé par plusieurs développeurs de logiciels. Son but est d'améliorer leurs processus et de réduire leur taux d'échec. Pour cela, ils placent le client au cœur du projet et s'adaptent tout au long de son déroulement. C'est donc une toute nouvelle façon de voir les choses et d'aborder le développement d'un produit, d'un service ou d'un projet. Parmi les méthodes agiles les plus usuelles, on peut citer la méthode SCRUM.

### 2.4.2 Méthode SCRUM :

Scrum n'est pas une méthode, mais un cadre de travail (framework) permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible. Les cinq valeurs de Scrum sont : Focus, Ouverture, Respect, Courage et Engagement (FORCE). L'objectif de ces valeurs est d'aider les équipes à créer et renforcer un mindset agile.

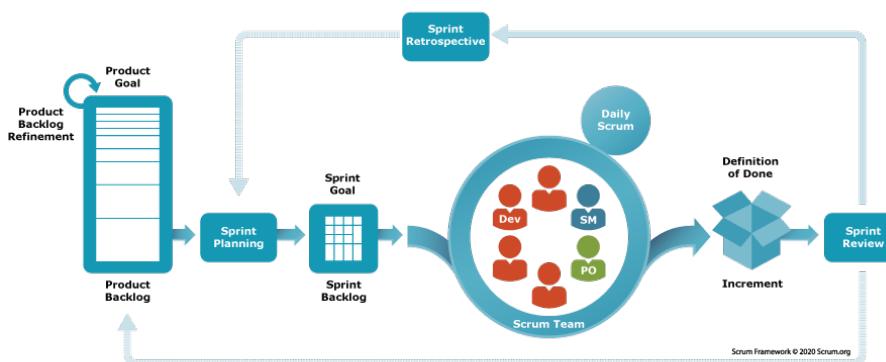


FIG. 2.1 – SCRUM

J'assiste à des réunions avec mon encadrante tous les jours. Lors de ces rencontres, je présente brièvement ce que j'ai accompli la veille, ce que je compte faire aujourd'hui, ainsi que les éventuels problèmes ou obstacles que j'ai rencontrés. Cette routine me permet de rapporter régulièrement mes progrès et de demander de l'aide à mon encadrante si j'ai un problème.

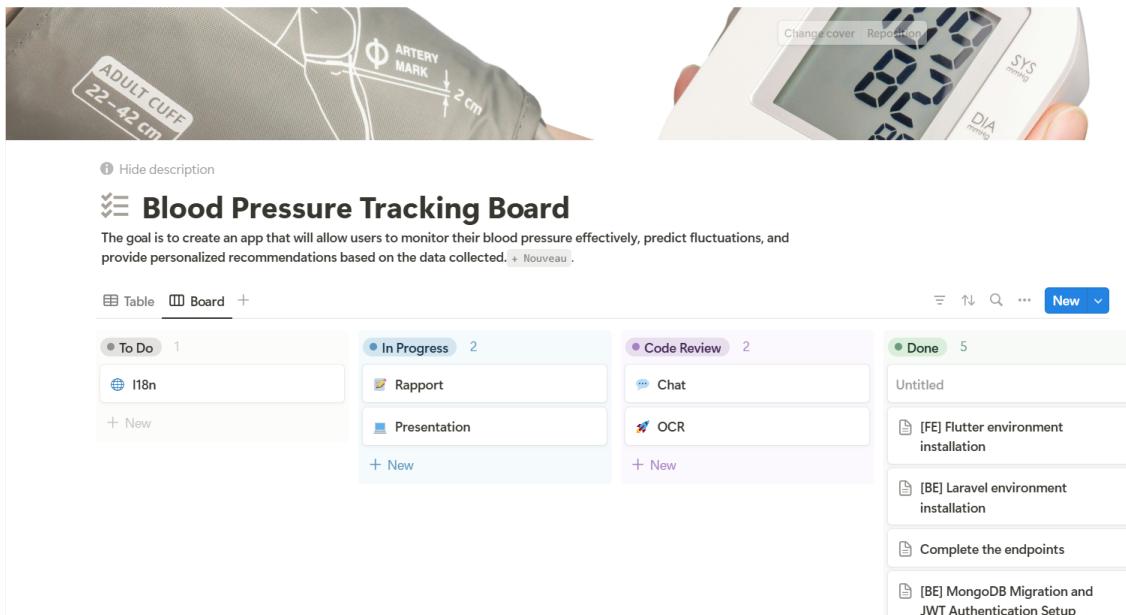


FIG. 2.2 – KanBan

Ce tableau Kanban est composé de trois phases : "To Do" (À faire), "In Progress" (En cours), "Done" (Terminé). La première phase, "To Do", représente les tâches ou les éléments qui doivent encore être accomplis (Frontend, template). La deuxième phase, "In Progress", indique que les tâches sont en cours d'exécution (Autoformation (Angular)). Et la dernière phase, "Done", représente les tâches ou les éléments qui ont été complétés avec succès (Autoformation (Django), cahier des charges, backend).

## 2.5 Conclusion :

Dans ce chapitre, j'ai établi un cadre complet pour le travail à venir. J'ai commencé par décrire le contexte et les objectifs du projet, offrant ainsi une vision claire des défis à relever et des objectifs à atteindre. Ensuite, j'ai présenté ma solution, mettant en lumière les méthodologies de développement qui guideront la réalisation du projet. Cette étape préliminaire fournit une base solide pour la suite du travail. La compréhension de cette étude sera enrichie davantage lors de la phase d'étude conceptuelle dans le prochain chapitre.

# **Section 3:**

## **Etude Conceptuel**

### 3.1 Introduction :

Dans ce chapitre, après avoir identifié les problématiques et défini les objectifs du projet de Norsys, nous plongeons dans une étude conceptuelle approfondie. Cette phase vise à comprendre les besoins du système et des utilisateurs. Nous débutons par le diagramme de cas d'utilisation pour une vision globale des interactions, suivi du diagramme de classe décrivant la structure statique du système. Ces outils jetteront des bases solides pour la conception et le développement de la solution.

### 3.2 Conception UML :

UML (Unified Modeling Language) a été conçu comme un langage de modélisation visuel commun, sémantiquement et syntaxiquement riche. Il est utilisé pour l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes en termes de structure et de comportement. UML a des applications potentielles au-delà du développement de logiciels, en particulier pour les flux de processus industriels.



FIG. 3.1 – UML Logo .

### 3.3 Diagramme de cas d'utilisation :

Un diagramme de cas d'utilisation est une représentation graphique qui illustre les interactions entre les acteurs (utilisateurs ou systèmes externes) et les différents cas d'utilisation d'un système.

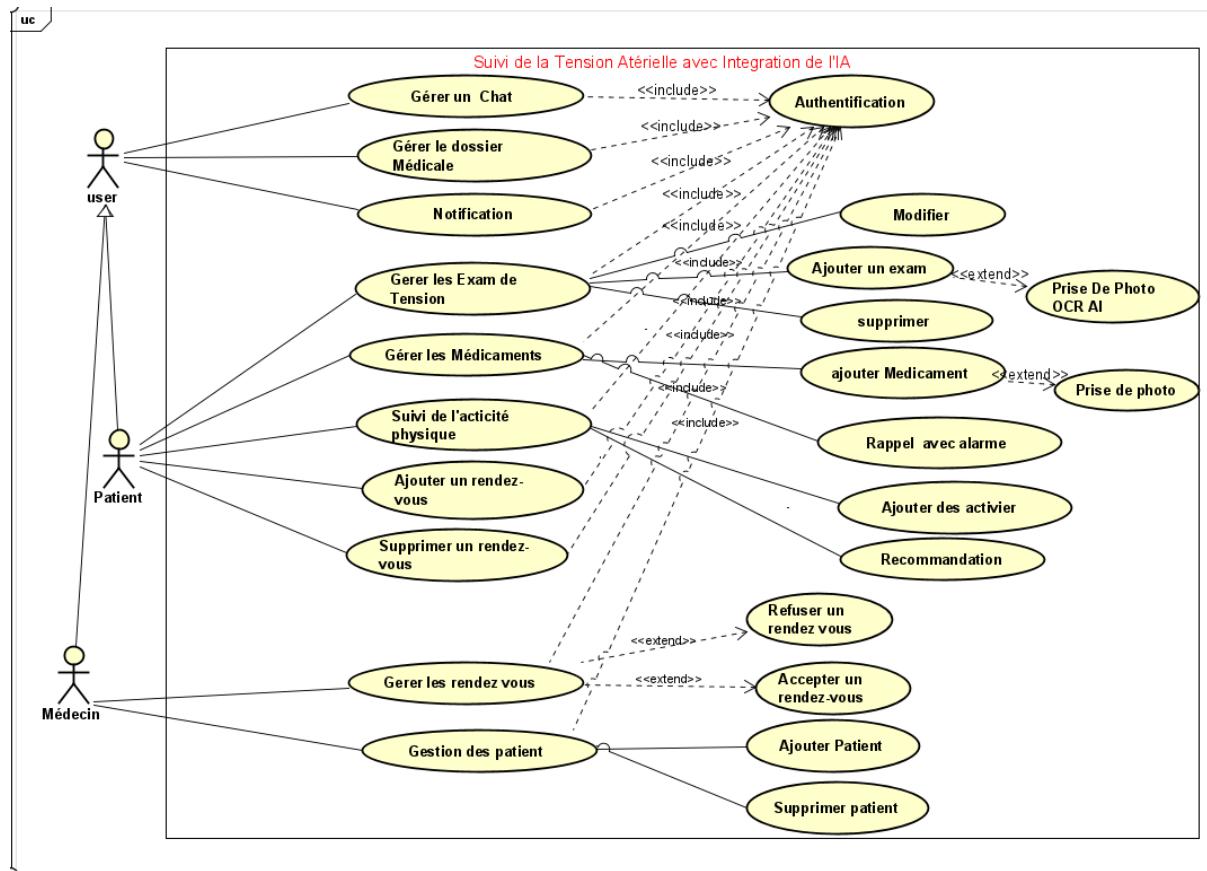


FIG. 3.2 – Diagramme de cas d'utilisation.

Le diagramme de cas d'utilisation ci-dessus décrit les principales fonctionnalités de l'application de suivi de la tension artérielle avec intégration de l'intelligence artificielle (IA). Voici une description des différents cas d'utilisation et des acteurs impliqués :

## Acteurs

- **User** : Représente tout utilisateur de l'application.
- **Patient** : Utilisateur principal de l'application qui surveille sa tension artérielle.
- **Médecin** : Professionnel de santé qui utilise l'application pour gérer les données des patients et fournir des recommandations.

## Cas d'utilisation

- **Gérer un Chat**
  - L'utilisateur peut gérer un chat pour communiquer avec d'autres utilisateurs ou professionnels de santé.
  - Inclut l'authentification pour sécuriser les échanges.
- **Gérer le dossier Médicale**
  - Les utilisateurs peuvent gérer leur dossier médical, ajoutant et modifiant des informations.
  - Inclut l'authentification pour garantir la confidentialité des données.
- **Notification**

- L'application envoie des notifications aux utilisateurs pour divers événements (rappels, résultats, etc.).
- Inclut l'authentification pour l'accès sécurisé aux notifications.
- **Gérer les Exam de Tension**
  - Les patients peuvent ajouter, modifier ou supprimer des examens de tension artérielle.
  - L'ajout d'examens peut inclure la prise de photo avec OCR AI pour l'analyse des résultats.
- **Gérer les Médicaments**
  - Les patients peuvent ajouter et gérer leurs médicaments, y compris des rappels avec alarme pour la prise des médicaments.
  - Inclut l'authentification pour l'accès sécurisé à cette fonctionnalité.
- **Suivi de l'activité physique**
  - Les patients peuvent suivre leur activité physique.
  - Inclut l'authentification pour garantir la sécurité des données.
- **Ajouter un rendez-vous**
  - Les patients peuvent ajouter des rendez-vous avec des professionnels de santé.
  - Inclut l'authentification pour un accès sécurisé.
- **Supprimer un rendez-vous**
  - Les patients peuvent supprimer des rendez-vous.
  - Inclut l'authentification pour garantir que seules les personnes autorisées peuvent effectuer cette action.
- **Gérer les rendez-vous**
  - Les médecins peuvent gérer les rendez-vous, incluant accepter ou refuser des rendez-vous.
  - Les extensions montrent les différentes actions possibles comme accepter ou refuser un rendez-vous.
- **Gestion des patients**
  - Les médecins peuvent gérer les informations des patients, y compris ajouter ou supprimer des patients.

## Extensions

- **Prise de Photo OCR AI** : Extension du cas d'utilisation "Ajouter un exam de Tension" permettant d'utiliser la reconnaissance optique de caractères pour analyser les résultats.
- **Prise de photo** : Extension du cas d'utilisation "Ajouter Medicament" pour enregistrer des images de médicaments.
- **Rappel avec alarme** : Extension du cas d'utilisation "Gérer les Médicaments" pour configurer des rappels.

## Sécurité

- **Authentification** : Presque tous les cas d'utilisation incluent l'authentification pour s'assurer que les utilisateurs accèdent uniquement aux fonctionnalités et données autorisées.

### 3.4 Diagramme de Classe :

Un diagramme de classe est une représentation visuelle qui illustre la structure et les relations entre les classes dans un système logiciel. Les classes sont des éléments fondamentaux de la programmation orientée objet, et elles regroupent des données (attributs) et des méthodes (fonctions ou opérations) liées.

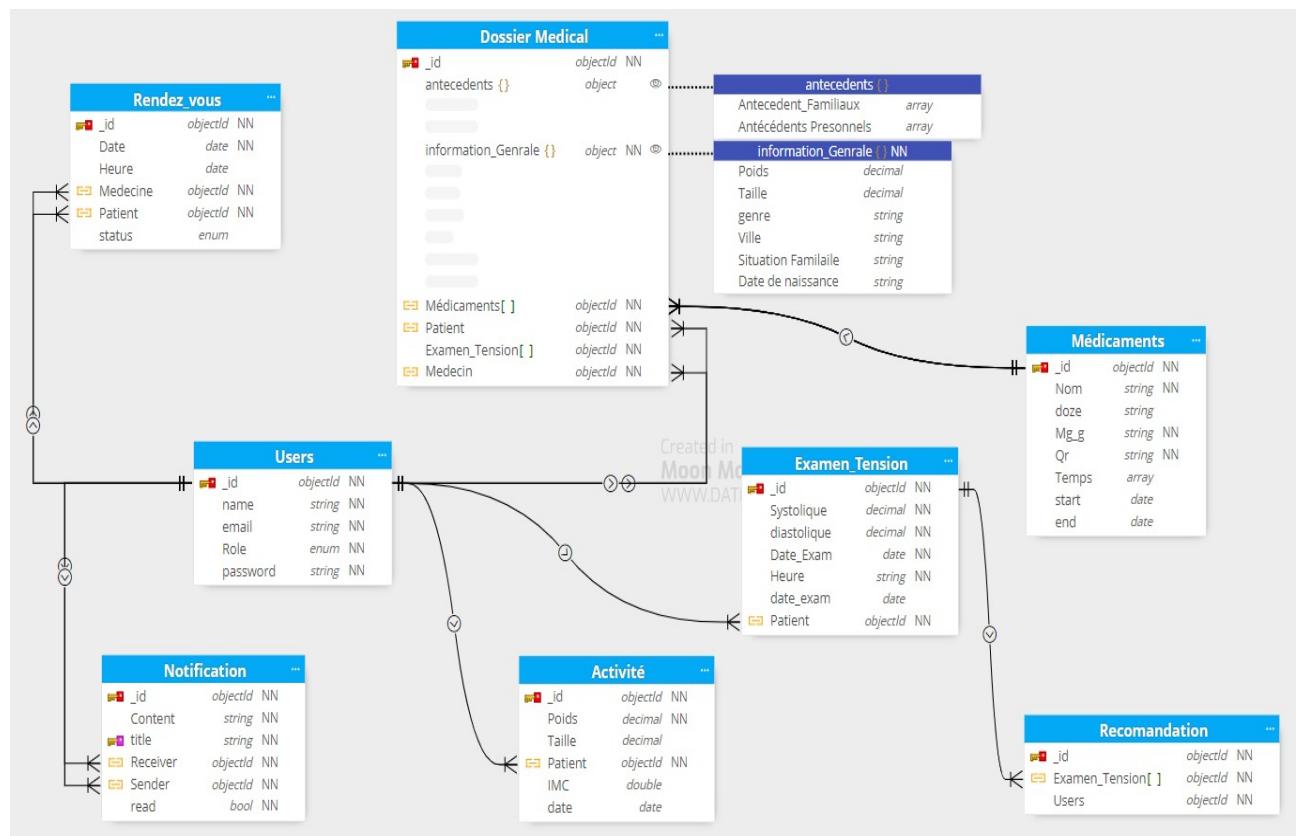


FIG. 3.3 – Diagramme de classe pour MongoDB.

- **Utilisateurs** : Cette collection centralise les informations des utilisateurs, incluant des identifiants uniques pour chaque utilisateur.
- **Notifications** : Cette collection relie les utilisateurs pour envoyer et recevoir des notifications. Elle contient des références aux utilisateurs pour les expéditeurs et les destinataires.
- **Rendez-vous** : Cette collection gère les rendez-vous médicaux, incluant des références aux patients et aux informations pertinentes sur les rendez-vous.
- **Activités** : Cette collection enregistre les activités des utilisateurs, avec des relations vers la collection des utilisateurs.
- **Examen de Tension** : Cette collection stocke les résultats des examens de tension, avec des relations vers les patients et les examens correspondants.

- **Recommandations** : Cette collection contient des recommandations médicales, faisant référence aux examens de tension et aux utilisateurs concernés.
- **Médicaments** : Cette collection gère les informations sur les médicaments, avec des détails spécifiques sur chaque médicament.
- **Dossier Médical** : Collection centrale pour le suivi médical des patients, incluant des antécédents médicaux et des informations générales. Elle est reliée aux examens de tension, aux médicaments et aux patients.

### 3.5 BPMN (Business Process Model and Notation) :

Un diagramme BPMN (Business Process Model and Notation) est une représentation graphique d'un processus métier. Il utilise des symboles standardisés pour décrire les différentes étapes, les flux de travail et les interactions entre les participants au sein d'un processus. Le BPMN est conçu pour être facilement compris par tous les intervenants, des analystes métier aux développeurs techniques, facilitant ainsi la communication et la collaboration dans l'amélioration et l'automatisation des processus métier.

- Le diagramme ci-dessous illustre le processus d'authentification d'application mobile.

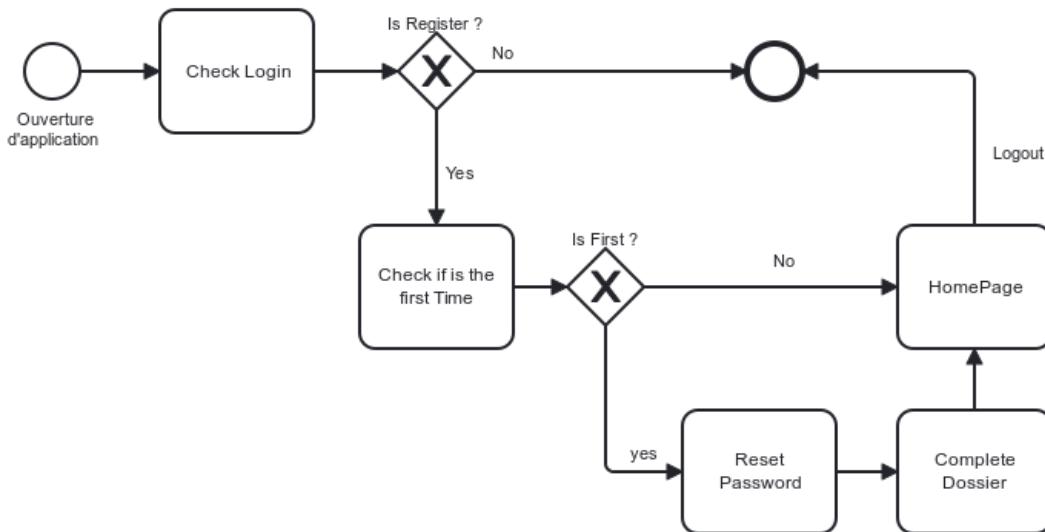


FIG. 3.4 – Diagramme BPMN de processus d'authentification .

Le processus commence par l'ouverture de l'application. Ensuite, l'utilisateur tente de se connecter. Si l'utilisateur n'est pas enregistré, il est redirigé vers la page d'accueil. En cas de première connexion, l'utilisateur doit réinitialiser son mot de passe. Après cette réinitialisation, l'utilisateur doit compléter son dossier. Enfin, une fois ces étapes complétées, l'utilisateur accède à la page d'accueil de l'application. Ce processus garantit que chaque utilisateur est authentifié correctement et que ses informations sont à jour avant d'accéder aux fonctionnalités principales de l'application.

- Le diagramme ci-dessous illustre le processus d'authentification d'une application.

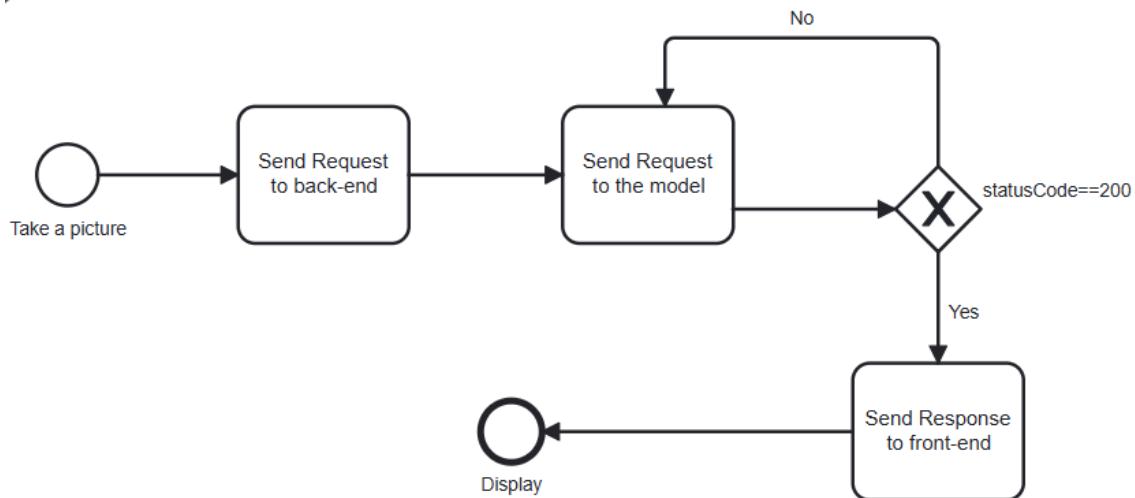


FIG. 3.5 – Diagramme BPMN de model OCR.

Le processus commence par la prise d'une photo par l'utilisateur. Ensuite, la photo est envoyée au back-end de l'application. Le back-end transmet ensuite la photo au modèle OCR pour analyse. Si le code de statut retourné par le modèle OCR n'est pas 200, la requête est renvoyée au modèle pour une nouvelle tentative. Si le code de statut est 200, le résultat de l'analyse est envoyé au front-end. Enfin, le résultat de l'analyse OCR est affiché à l'utilisateur. Ce processus garantit que l'image est correctement analysée par le modèle OCR et que les résultats sont affichés de manière fiable à l'utilisateur.

### 3.6 Conclusion :

En conclusion, ce chapitre a permis de définir les principaux aspects de la conception de notre application en se focalisant sur les diagrammes UML et BPMN. Ces diagrammes ont été essentiels pour modéliser les processus métier et les interactions au sein du système, assurant ainsi une compréhension claire des flux de travail et des rôles de chaque composant. Nous avons détaillé les différentes interactions et processus critiques pour le bon fonctionnement de l'application, établissant ainsi une base solide pour la mise en œuvre et le développement de notre projet. Le chapitre suivant se concentrera sur l'architecture globale du système, détaillant les différentes couches du backend et leur interaction avec les applications front-end destinées aux patients et aux médecins.

## **Section 4:**

# **L'Architecture Globale du Système**

## 4.1 Introduction

L'architecture globale du système joue un rôle crucial dans la réalisation des objectifs de notre projet. Elle est conçue pour offrir une performance élevée, une sécurité renforcée, une évolutivité facilitée et une soutenabilité à long terme. Cette section présente une vue d'ensemble des composants principaux de l'architecture, décrivant comment ils interagissent pour fournir une solution complète et intégrée pour le suivi de la tension artérielle. Grâce à l'utilisation de technologies modernes comme Flutter pour l'application mobile, Next.js pour l'application web, une API REST robuste pour le backend, et Firebase pour la gestion des chats en temps réel et l'authentification, nous assurons une expérience utilisateur fluide et efficace, tout en garantissant la confidentialité et la sécurité des données médicales des utilisateurs.

## 4.2 Description de mon Système

Le système global est conçu pour fournir une solution intégrée et efficace pour le suivi de la tension artérielle. Il se compose de cinq parties principales : le backend, l'application pour les patients, l'application pour les médecins, Firebase pour les chats en temps réel et l'authentification secondaire, et le modèle OCR de Hugging Face pour la reconnaissance de texte.

- **Backend** : Point central pour la gestion des données et des opérations métier, communiquant avec la base de données pour stocker et récupérer les informations, et exposant des endpoints sécurisés pour les applications frontales. L'authentification principale est gérée par le backend Laravel.
- **Application pour les Patients** : Fournit une interface conviviale pour enregistrer et suivre les mesures de tension artérielle des patients, visualiser les tendances et recevoir des recommandations personnalisées, tout en synchronisant les données avec le backend.
- **Application pour les Médecins** : Offre des outils avancés pour analyser les données des patients, établir des diagnostics et recommander des traitements, permettant aux médecins d'accéder à l'historique des mesures et de communiquer directement avec les patients via Firebase, tout en se synchronisant avec le backend.
- **Firebase** : Utilisé pour la gestion des chats en temps réel entre les patients et les médecins et pour l'authentification secondaire. Firebase Auth simplifie l'intégration de l'authentification sécurisée dans nos applications.
- **Modèle OCR de Hugging Face** : Intègre le modèle OCR-Donut-CORD pour la reconnaissance optique de caractères. Cela permet de transformer les données textuelles issues de documents papier ou d'images en données numériques exploitables au sein du système.

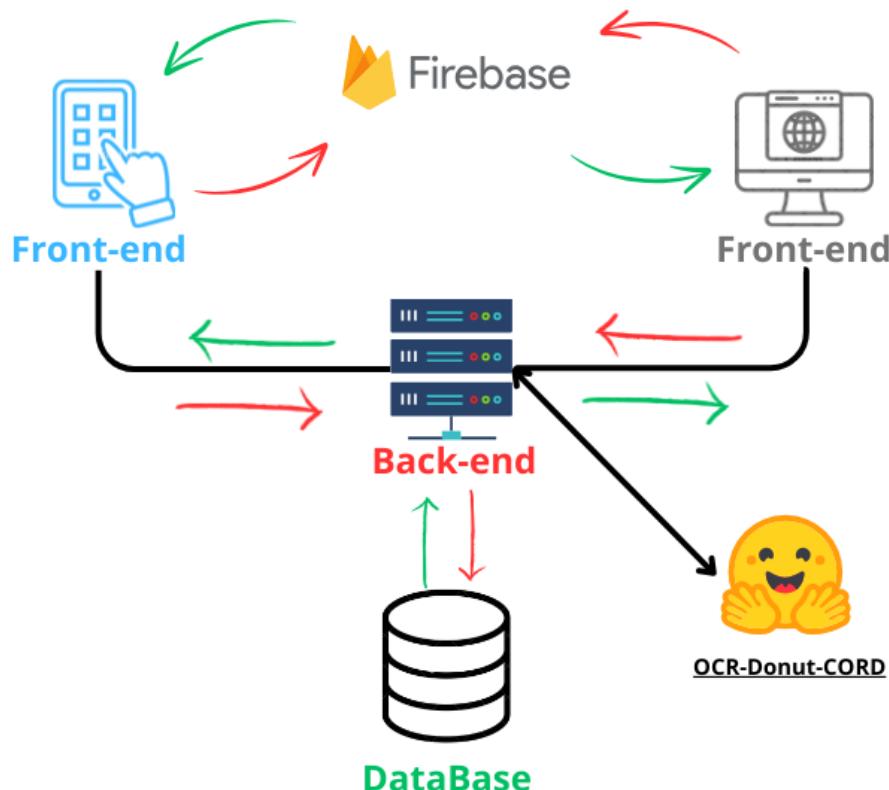


FIG. 4.1 – Architecture globale du système.

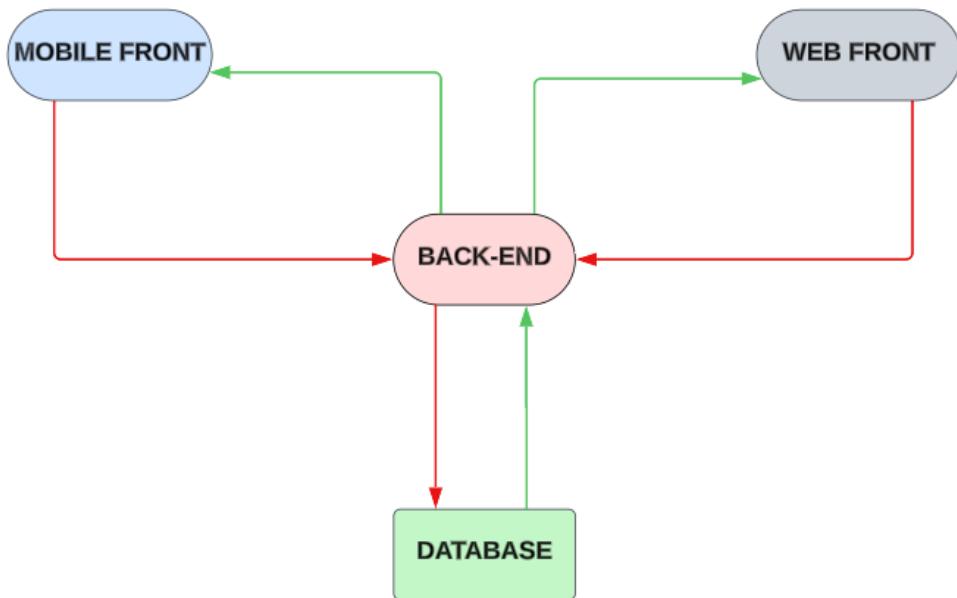


FIG. 4.2 – Architecture de système.

Cette architecture assure une interaction fluide et sécurisée entre les composants, garantissant une gestion optimale des données de santé et une expérience utilisateur harmonieuse pour les patients et les médecins.

#### 4.2.1 Architecture de Back-end

Dans le backend, nous avons suivi le design pattern Repository, qui se compose de plusieurs couches interconnectées : Interface, Repository, Service et Controller. Ce design pattern est utilisé pour isoler le domaine et les couches de logique métier de l'application, tout en facilitant la gestion des données et les interactions avec la base de données.

- **Interface** : Les interfaces définissent les contrats que les repositories doivent implémenter. Elles spécifient les méthodes pour accéder et manipuler les données sans exposer les détails d'implémentation. Cela permet une flexibilité et une facilité de maintenance, car les implémentations peuvent être changées sans affecter les consommateurs de ces interfaces.
- **Repository** : Les repositories implémentent les interfaces et contiennent la logique pour interagir avec la base de données. Ils encapsulent les détails d'accès aux données et fournissent des méthodes pour récupérer, ajouter, mettre à jour et supprimer des données. Cela permet de centraliser la logique d'accès aux données en un seul endroit.
- **Service** : La couche Service contient la logique métier de l'application. Elle utilise les repositories pour effectuer des opérations sur les données et appliquer les règles métier. En séparant la logique métier des contrôleurs et des repositories, nous assurons une meilleure organisation du code et une réutilisabilité accrue.
- **Controller** : Les contrôleurs gèrent les requêtes entrantes de l'application front-end et appellent les services pour effectuer les opérations nécessaires. Ils sont responsables de la gestion du cycle de vie des requêtes HTTP et de la conversion des réponses en format approprié pour le client.

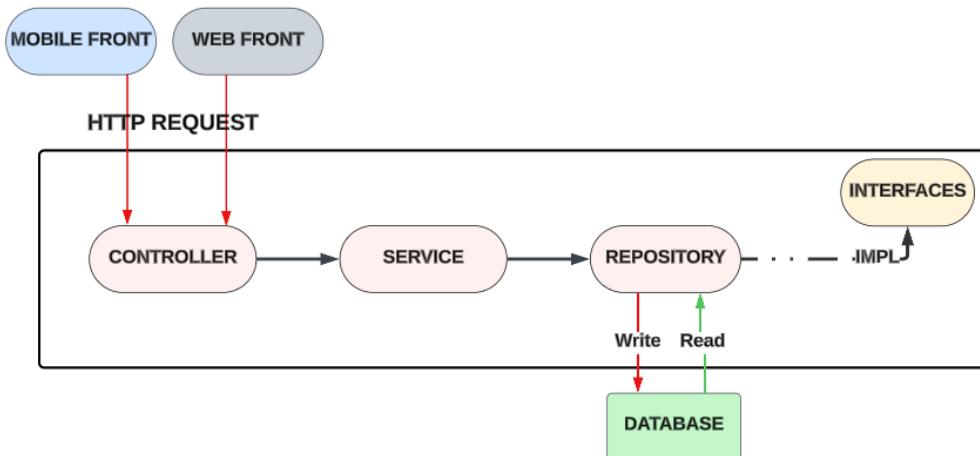


FIG. 4.3 – Architecture de Back-End.

Cette architecture en couches offre plusieurs avantages, notamment une meilleure organisation du code, une séparation claire des responsabilités, une facilité de test et une maintenance simplifiée. En isolant la logique d'accès aux données, la logique métier et les contrôleurs, nous pouvons évoluer et maintenir le système plus efficacement.

#### 4.2.2 Architecture de Front-end

Dans le front-end, nous avons suivi une architecture en couches pour structurer notre code de manière organisée et modulaire. Cette architecture comprend plusieurs composants inter-

connectés : **Api**, **Service**, **i18n**, **Class**, **Utils**, et **Screens**. Cette approche permet d'isoler les différentes responsabilités de l'application, tout en facilitant la gestion des données et les interactions avec l'interface utilisateur.

- **Api** : Définit les appels et interactions avec les services backend, permettant une flexibilité et une facilité de maintenance.
- **Service** : Contient la logique métier, utilisant les APIs pour appliquer les règles métier et assurant une meilleure organisation du code.
- **i18n** : Gère la traduction et l'internationalisation, permettant l'adaptation de l'application à différents marchés linguistiques.
- **Class** : Contient les modèles de données principaux, encapsulant la logique de validation et manipulation des données.
- **Utils** : Regroupe des fonctions d'assistance, incluant le formatage de date, la manipulation de chaînes de caractères, et la gestion des préférences utilisateur.
- **Screens** : Contient les différentes pages de l'application, chaque écran représentant une interface utilisateur distincte.

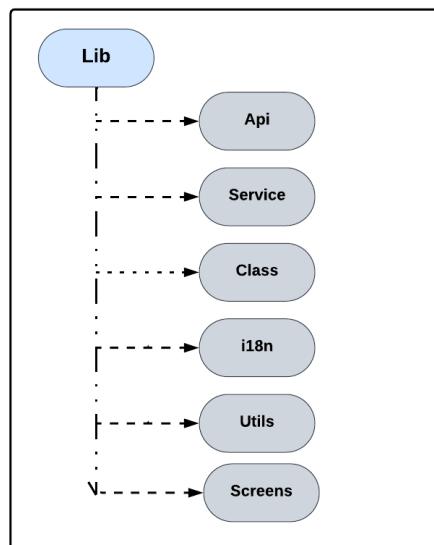


FIG. 4.4 – Architecture de Front-end.

### 4.3 Intégration de Firebase

Firebase est intégré dans notre architecture pour fournir des fonctionnalités de chat en temps réel et une authentification secondaire. Cette intégration est essentielle pour assurer une communication fluide et sécurisée entre les patients et les médecins, ainsi que pour offrir une couche de sécurité supplémentaire pour l'accès aux données sensibles.

- **Chat en temps réel** : Firebase Realtime Database est utilisé pour la gestion des messages instantanés entre les patients et les médecins. Cette base de données permet une synchronisation en temps réel des messages, offrant une expérience de chat fluide et réactive.

- **Authentification Firebase** : Firebase Authentication est utilisé pour gérer l'authentification secondaire des utilisateurs. Cela permet de renforcer la sécurité de l'application en ajoutant une couche supplémentaire de vérification des utilisateurs.

## 4.4 Modèle OCR de Hugging Face :

### 4.4.1 OCR :

La reconnaissance optique de caractères (OCR) est une technologie qui permet de convertir différents types de documents contenant du texte en format image, tels que des documents papier numérisés, des fichiers PDF ou des images capturées par un appareil photo numérique, en texte numérique éditable. Cette conversion est essentielle pour extraire des informations textuelles à partir de sources non numériques, facilitant ainsi la manipulation, l'analyse et la recherche de données textuelles.

### 4.4.2 Fonctionnement du Modèle OCR de Hugging Face

- **Acquisition de l'Image** : L'image contenant le texte est d'abord fournie au modèle. Cela peut être une photo, un scan d'un document ou toute autre image contenant du texte.
- **Prétraitement de l'Image** : Le modèle utilise des techniques de traitement d'image pour améliorer la qualité et la lisibilité de l'image. Cela peut inclure la correction de l'angle de l'image, l'amélioration de la clarté, et la réduction du bruit.
- **Détection et Segmentation du Texte** : Le modèle localise les zones de l'image qui contiennent du texte. Cela inclut la segmentation en lignes, mots, et caractères, permettant ainsi une analyse plus fine.
- **Reconnaissance des Caractères** : Chaque caractère ou groupe de caractères est identifié à l'aide d'algorithmes de reconnaissance de motifs basés sur des réseaux neuronaux. Le modèle compare les caractères détectés avec une base de données de caractères connus pour déterminer le texte.
- **Post-traitement** : Les résultats initiaux de la reconnaissance sont affinés pour corriger les erreurs et assurer la cohérence du texte. Cela peut inclure la correction orthographique et la mise en forme du texte.
- **Sortie du Texte** : Le texte extrait est présenté sous forme numérique, prêt à être utilisé dans des applications ou des systèmes d'analyse de données.

### 4.4.3 Description du Modèle OCR-Donut-CORD

Donut se compose d'un encodeur de vision (Swin Transformer) et d'un décodeur de texte (BART). Étant donné une image, l'encodeur commence par encoder l'image en un tenseur d'embeddings (de forme batch\_size, seq\_len, hidden\_size), après quoi le décodeur génère du texte de manière autorégressive, conditionné par l'encodage produit par l'encodeur.

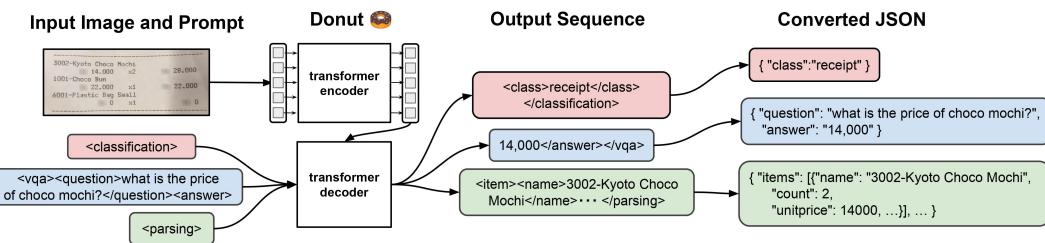


FIG. 4.5 – Donut Architecture.

#### 4.4.4 Intégration du Script Python avec Laravel en utilisant LaravelPython

Pour intégrer un script Python dans un projet Laravel en utilisant le package LaravelPython, suivez les étapes générales suivantes :

- **Installation de LaravelPython** : Commencez par installer le package LaravelPython via Composer. Cette installation permet à Laravel de communiquer avec des scripts Python.
- **Configuration de LaravelPython** : Une fois installé, publions le fichier de configuration pour LaravelPython. Cela nous permettra de définir les paramètres nécessaires pour l'exécution des scripts Python dans un fichier de configuration dédié.
- **Création du Script Python** : Écrivons notre script Python et plaçons-le dans un répertoire spécifique de notre projet Laravel. Ce script contiendra la logique que nous souhaitons exécuter depuis Laravel.
- **Appel du Script Python depuis Laravel** : Utilisons LaravelPython pour appeler et exécuter le script Python à partir de nos routes ou contrôleurs Laravel. Cela se fait en configurant les routes ou les contrôleurs pour qu'ils déclenchent l'exécution du script et retournent les résultats nécessaires.
- **Test et Validation** : Accédons aux routes ou utilisons des outils comme Postman pour tester que le script Python s'exécute correctement et que les résultats sont retournés comme prévu.

## 4.5 Conclusion

En conclusion de ce chapitre, nous avons clairement défini les objectifs et les principaux aspects de l'architecture de notre application mobile et web. Nous avons examiné en détail les différentes couches du backend et leur interaction avec les applications front-end destinées aux patients et aux médecins. L'intégration de Firebase pour la gestion des chats en temps réel et une authentification renforcée renforcent la sécurité et la fluidité des échanges au sein du système. Cette architecture garantit une gestion optimale des données de santé et vise à offrir une expérience utilisateur harmonieuse pour toutes les parties impliquées. Ces fondations solides posent les bases essentielles pour la mise en œuvre et le développement réussi de notre projet.

## **Section 5:**

# **Technologie et les Outils Utilisés**

## 5.1 Introduction :

Dans ce chapitre, nous explorerons les technologies et outils utilisés pour développer notre application de suivi de la tension artérielle, axés sur la performance, la sécurité, l'évolutivité et la facilité de maintenance. Nous couvrirons le développement du backend, des applications front-end pour les patients et les médecins, ainsi que les technologies pour l'analyse des données et l'intelligence artificielle, soulignant nos choix technologiques pour une expérience utilisateur avancée.

### 5.1.1 Technologies et Outils Utilisés

Dans cette sous-section, nous détaillerons les différentes technologies et outils utilisés dans le cadre du développement de notre application de suivi de la tension artérielle, en mettant en avant leur rôle et leur contribution à notre projet :



FIG. 5.1 – Atlas MongoDB .

- **Atlas MongoDB et MongoCompass** : Atlas MongoDB est une plateforme de base de données cloud qui offre une gestion efficace et évolutive des données. MongoCompass est un outil de visualisation et d'administration pour MongoDB, permettant une exploration intuitive de nos données

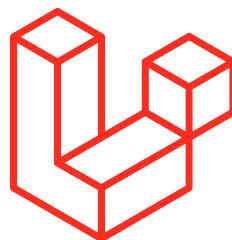


FIG. 5.2 – Laravel .

- **Laravel** : Laravel est un framework PHP moderne et robuste utilisé pour le développement du backend de notre application. Il offre une structure MVC claire, ainsi que des fonctionnalités avancées pour la gestion des requêtes HTTP, la validation des données et la sécurité.



FIG. 5.3 – Next.js .

- **Next.js** : Next.js est un framework React utilisé pour le développement d'applications web. Il offre des fonctionnalités avancées telles que le rendu côté serveur et la génération de pages statiques, garantissant une expérience utilisateur fluide et rapide.



FIG. 5.4 – Flutter .

- **Flutter** : Flutter est un framework open-source développé par Google, utilisé pour la création d'applications mobiles multiplateformes. Grâce à sa performance élevée et sa facilité de développement, nous avons choisi Flutter pour développer l'interface utilisateur de notre application mobile.



FIG. 5.5 – Tailwind CSS.

- **Tailwind CSS** : Tailwind CSS est un framework CSS utilisé pour styliser l'interface utilisateur de notre application web. Il offre une approche basée sur les classes pour la conception des composants, ce qui permet un développement rapide et flexible.



FIG. 5.6 – TypeScript .

- **TypeScript** : TypeScript est un langage de programmation open-source qui étend JavaScript en lui ajoutant des fonctionnalités de typage statique. Nous avons utilisé TypeScript pour améliorer la robustesse et la maintenabilité de notre code frontend.



FIG. 5.7 – HTML



FIG. 5.8 – CSS

- **HTML et CSS** : HTML (HyperText Markup Language) et CSS (Cascading Style Sheets) sont les langages de base utilisés pour créer la structure et le style de notre application web. Ils fournissent les fondations de l'interface utilisateur et définissent son apparence et son comportement.



FIG. 5.9 – JWT (JSON Web Tokens) .

- **JWT (JSON Web Tokens)** : JWT est un standard ouvert qui définit un moyen compact et autonome pour sécuriser les échanges d'informations entre parties, notamment pour l'authentification et l'autorisation dans les applications web et mobiles.



FIG. 5.10 – Postman .

- **Postman** : Postman est un outil de développement d'API utilisé pour tester, déboguer et documenter les API. Il offre une interface conviviale pour envoyer des requêtes HTTP et inspecter les réponses, facilitant ainsi le développement et le débogage de nos services backend.



FIG. 5.11 – VSCode .

- **VSCode** : Visual Studio Code est un éditeur de code source léger mais puissant, utilisé pour le développement de notre application. Il offre des fonctionnalités avancées telles que la coloration syntaxique, l'achèvement automatique et la débogage intégré, ce qui en fait un outil de choix pour les développeurs.



FIG. 5.12 – Émulateur Android .

- **Émulateur Android** : L'émulateur Android est un outil fourni par Google pour tester des applications Android sur un ordinateur. Il nous permet de simuler différents appareils et configurations Android pour vérifier le bon fonctionnement de notre application mobile.



FIG. 5.13 – Hugging Face .

- **Hugging Face** : Hugging Face est une plateforme open-source offrant des modèles de traitement du langage naturel (NLP) pré-entraînés et des outils pour leur utilisation. Nous avons utilisé les modèles de Hugging Face pour l'analyse des données textuelles dans notre application.



FIG. 5.14 – OpenAI .

- **OpenAI** : OpenAI est une entreprise de recherche en intelligence artificielle, fournissant des API et des modèles pour développer des applications d'IA avancées. Nous avons intégré les capacités de traitement du langage naturel d'OpenAI dans notre application pour améliorer l'expérience utilisateur.



FIG. 5.15 – Astah UML .

- **Astah UML** : Astah UML est un outil de modélisation UML utilisé pour concevoir et visualiser la structure et le comportement de notre application, facilitant ainsi la communication et la collaboration entre les membres de l'équipe de développement.

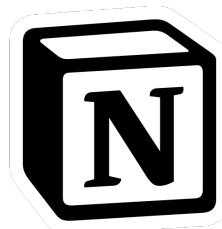


FIG. 5.16 – Notion .

- **Notion** : Notion est une plateforme de travail collaboratif qui offre des fonctionnalités de gestion de projet, de documentation et de prise de notes. Nous avons utilisé Notion pour organiser et partager des informations sur le projet, ainsi que pour suivre les tâches et les progrès.



FIG. 5.17 – Git



FIG. 5.18 – Github

— **Git et GitHub :** Git est un système de contrôle de version utilisé pour gérer le code source de notre application, permettant le suivi des modifications, la collaboration et la gestion des branches de développement. GitHub est une plateforme d'hébergement de code qui offre des fonctionnalités supplémentaires telles que le suivi des problèmes, les demandes de tirage et l'intégration continue.

## 5.2 Conclusion

En somme, ce chapitre a détaillé les technologies et outils essentiels au développement de notre application de suivi de la tension artérielle. Nous avons abordé les composants du backend, les applications front-end pour les patients et les médecins, ainsi que les technologies d'analyse des données et d'intelligence artificielle. Ces choix technologiques sont motivés par notre engagement à offrir une performance optimale, une sécurité robuste, une évolutivité sans faille, et une facilité de maintenance. Dans le prochain chapitre, nous nous concentrerons sur la réalisation du projet, en détaillant les étapes de développement.

## **Section 6:**

### **Réalisation du projet :**

## 6.1 Introduction :

Dans ce chapitre, nous aborderons le processus de développement d'une application de suivi de la tension artérielle, disponible sur les plateformes web et mobile, intégrant des fonctionnalités avancées basées sur l'intelligence artificielle. Nous détaillerons l'implémentation et du déploiement des deux versions de l'application, en mettant l'accent sur l'apport des technologies d'IA pour améliorer la précision du suivi et fournir des recommandations personnalisées aux utilisateurs.

## 6.2 Développement de l'application mobile

### 6.2.1 Système de Connexion

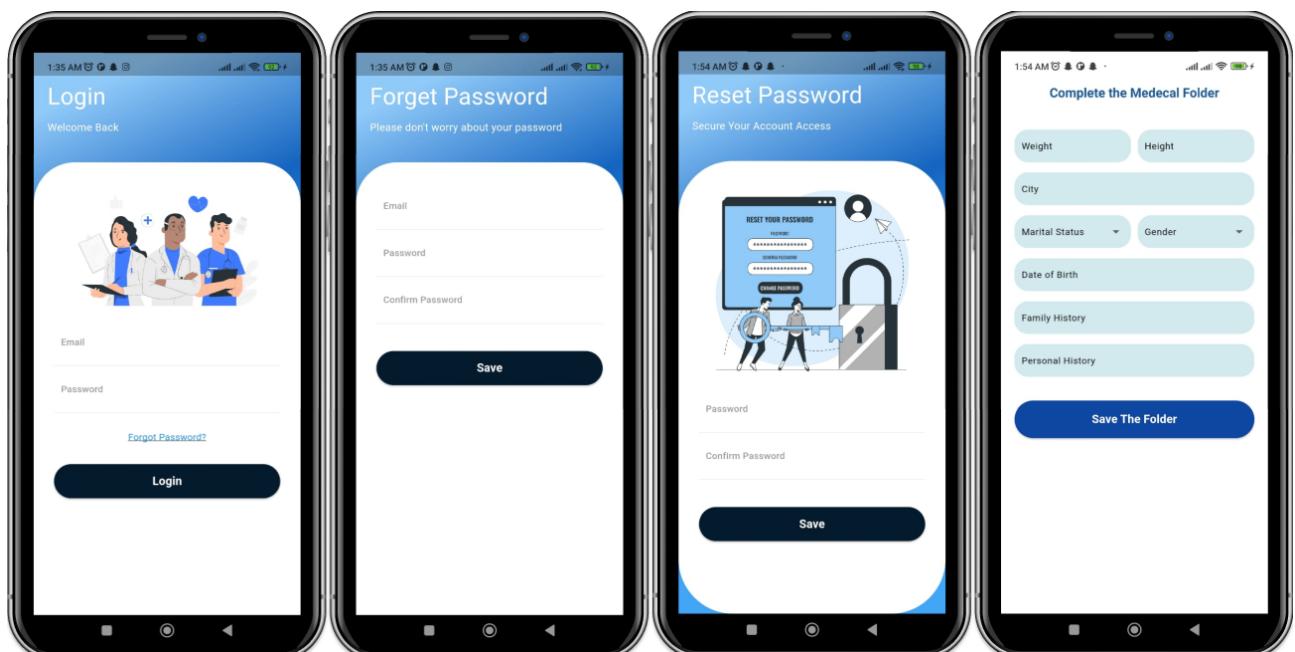


FIG. 6.1 – Screenshots of login-related functionalities.

Dans la figure 6.1, nous présentons plusieurs fonctionnalités liées à la connexion dans notre application de suivi de la tension artérielle :

- **Login ( ?? )** : Cet écran permet aux utilisateurs de se connecter à l'application en entrant leurs identifiants.
- **Forget Password ( ?? )** : Cette fonctionnalité permet aux utilisateurs de récupérer leur mot de passe en cas d'oubli en suivant un processus de récupération.
- **Reset Password ( ?? )** : Ici, les utilisateurs peuvent réinitialiser leur mot de passe .
- **Complete Dossier ( ?? )** : Ce formulaire permet aux utilisateurs de compléter leur profil avec des informations personnelles supplémentaires après leur première connexion.

Ces fonctionnalités sont essentielles pour assurer une expérience utilisateur fluide et sécurisée lors de l'utilisation de notre application.

## 6.2.2 Page d'Accueil

La page d'accueil de notre application représente le hub central à partir duquel les utilisateurs peuvent accéder à diverses fonctionnalités et informations importantes. Voici un aperçu des boutons principaux disponibles sur la page d'accueil :

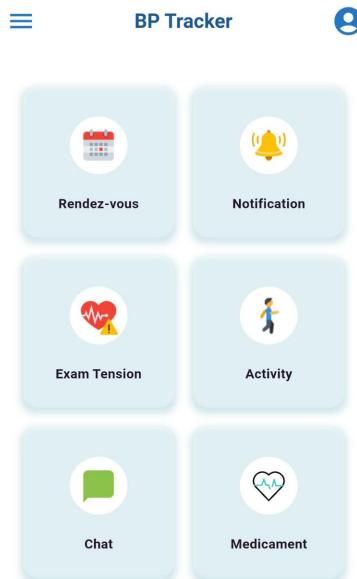


FIG. 6.2 – Page d'Accueil.

- **Rendez-vous** : Permet aux utilisateurs de gérer leurs rendez-vous médicaux et de consulter leur agenda.
- **Notifications** : Affiche les notifications concernant l'état des rendez-vous, comme l'acceptation ou le rejet d'une demande de rendez-vous.
- **Notifications** : Affiche les notifications importantes liées à la santé des utilisateurs, telles que les rappels de prise de médicaments ou les résultats d'examens, y compris ceux liés aux rendez-vous.
- **Exam Tension** : Fournit des informations sur la tension artérielle de l'utilisateur à différents moments de la journée ou en réponse à des activités spécifiques.
- **Activité** : Permet de suivre les niveaux d'activité physique de l'utilisateur, intégrant ces données dans l'analyse de la santé globale.
- **Chat** : Offre la possibilité aux utilisateurs de communiquer avec leur médecin pour des conseils ou du soutien.
- **Médicament** : Facilite la gestion des médicaments , avec des rappels.

Chaque bouton dans la page d'accueil représente un accès rapide à une fonctionnalité cruciale de l'application, visant à simplifier la gestion quotidienne de la santé et à fournir un soutien continu aux utilisateurs dans leur suivi de la tension artérielle.

### 1- Rendez-vous :

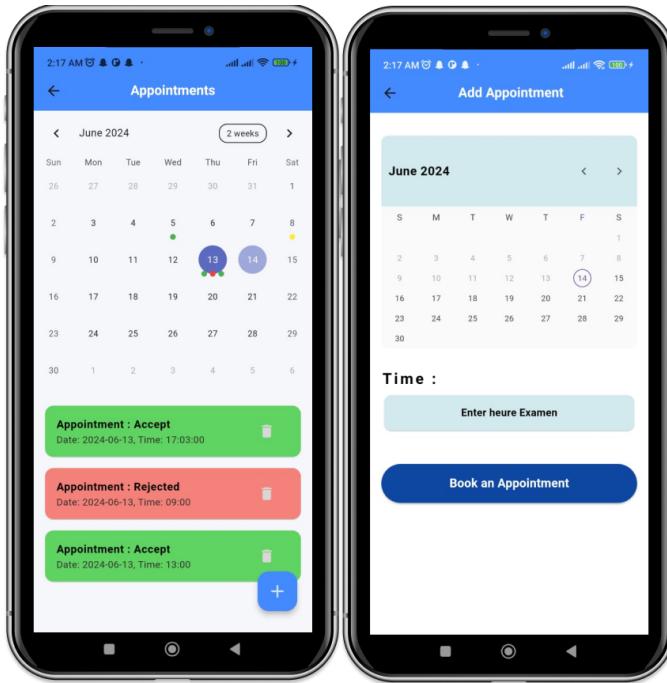


FIG. 6.3 – Gestion des Rendez-Vous.

La figure 6.3 présente deux écrans principaux de la fonctionnalité de gestion des rendez-vous dans l'application. À gauche, l'écran affiche la liste des rendez-vous avec des indications de leur statut, comme accepté ou rejeté, permettant ainsi à l'utilisateur de voir facilement l'état de ses demandes. À droite, l'écran montre le processus d'ajout d'un nouveau rendez-vous, où l'utilisateur peut sélectionner une date et une heure pour son rendez-vous médical. Cette interface simplifie la gestion des rendez-vous médicaux et aide les utilisateurs à garder un suivi précis et organisé de leurs consultations.

## 2- Fonctionnalité notification :

La figure 6.4 présente l'écran de notifications de l'application. Cet écran affiche la liste des notifications, indiquant si un rendez-vous a été accepté ou rejeté, ainsi que la date et l'heure correspondantes. Cela permet à l'utilisateur de suivre facilement l'état de ses demandes et d'être informé en temps réel des changements apportés à ses rendez-vous. Cette interface aide les utilisateurs à rester informés et organisés concernant leurs consultations médicales.

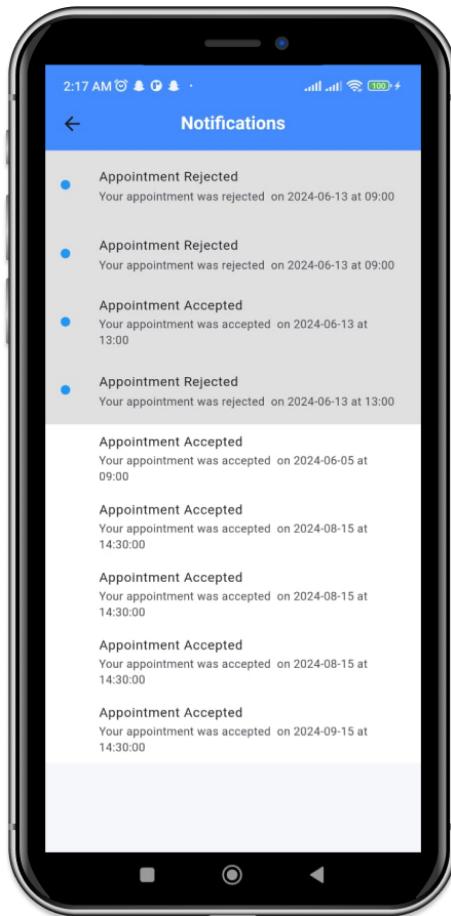


FIG. 6.4 – Notifications.

### 3- Exam Tension :

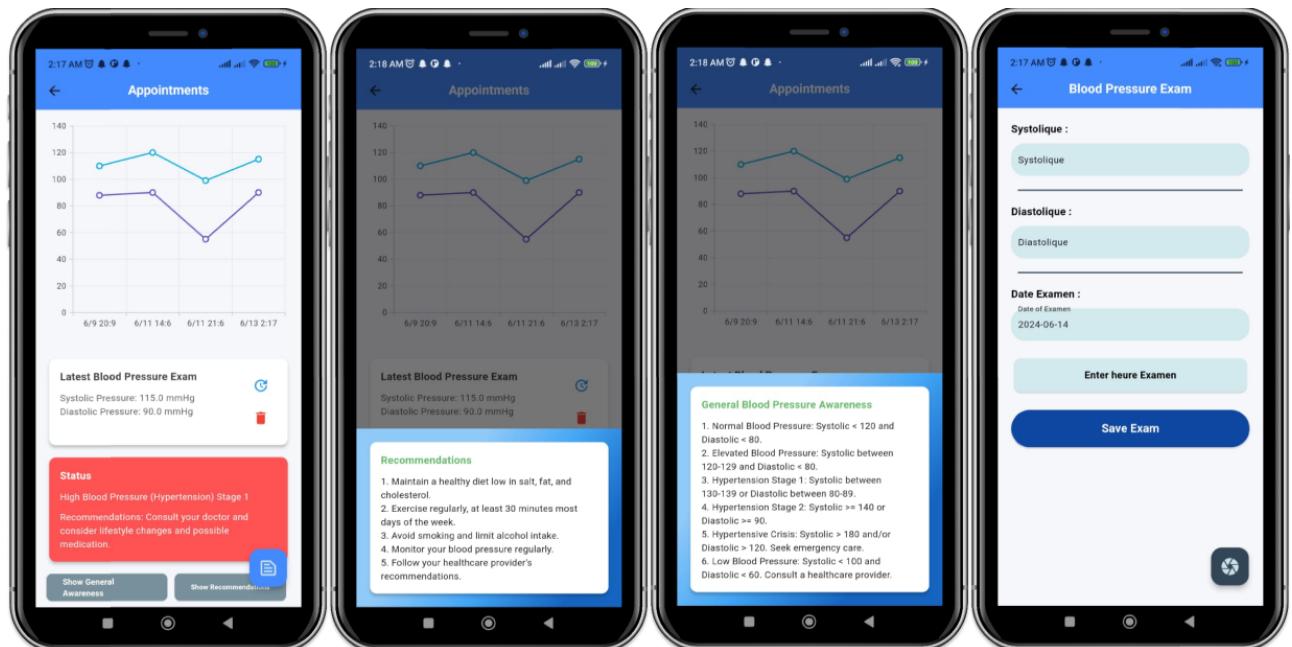


FIG. 6.5 – Screenshots of Blood Pressure interface.

La figure 6.5 présente plusieurs écrans de la fonctionnalité de gestion des examens de tension artérielle dans l'application. Les trois premiers écrans affichent les tendances de la tension artérielle systolique et diastolique au fil du temps, ainsi que des recommandations et des informations sur l'état actuel de l'utilisateur. Le quatrième écran montre le processus d'ajout d'un nouvel examen de tension artérielle.

Sur l'écran de l'ajout d'un nouvel examen, un bouton contenant une icône de caméra est utilisé pour traiter une image de l'appareil de mesure de la tension artérielle en utilisant un modèle **OCR (Reconnaissance Optique de Caractères)**. L'OCR est une technologie qui permet de convertir différents types de documents, tels que des images de texte manuscrit ou imprimé, en données textuelles éditables. Après le traitement de l'image, les champs de l'examen sont remplis automatiquement pour que le patient puisse vérifier les données récupérées. Ce modèle OCR est trouvé sur **Hugging Face** et nommé **jinyhybr/OCR-Donut-CORD** <https://huggingface.co/jinyhybr/OCR-Donut-CORD>. Cette fonctionnalité permet à l'utilisateur de capturer facilement les résultats de son examen de tension artérielle directement depuis l'appareil, facilitant ainsi l'enregistrement des données dans l'application.

#### 4- Gestion des activités :

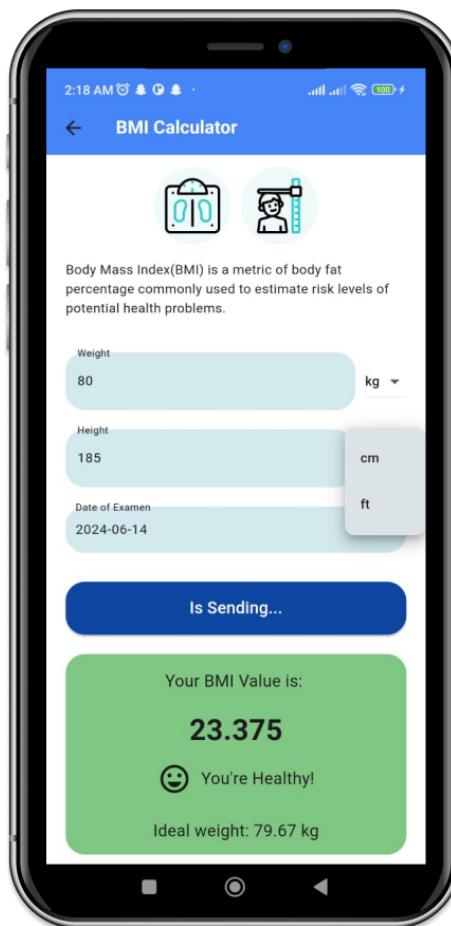


FIG. 6.6 – – BMI.

La figure 6.6 présente l'écran de l'outil de calcul de l'Indice de Masse Corporelle (IMC) dans l'application. Cet écran permet à l'utilisateur de saisir son poids, sa taille et la date de l'examen. L'utilisateur peut entrer la taille en centimètres ou en pieds en utilisant le menu déroulant.

### 5- Fonctionnalité chat :

L'outil de discussion de l'application utilise **Firebase** pour permettre une communication en temps réel entre les utilisateurs. Cette intégration garantit que les messages sont instantanément envoyés et reçus, offrant ainsi une expérience de messagerie fluide et réactive.



FIG. 6.7 – Chat.

La figure 6.7 présente l'écran de l'outil de discussion dans l'application. Cet écran permet à l'utilisateur de communiquer en temps réel avec d'autres utilisateurs. L'utilisateur peut voir les messages entrants et répondre en saisissant du texte dans le champ de saisie en bas de l'écran.

Après avoir tapé un message et cliqué sur le bouton d'envoi, le message apparaît dans le fil de discussion, horodaté pour indiquer l'heure d'envoi. Dans cet exemple, les messages affichés montrent une conversation continue entre deux utilisateurs, démontrant la fonctionnalité de messagerie instantanée de l'application.

### 6- Gestion des médicaments :

La figure 6.8 présente l'écran de gestion des médicaments dans l'application. Ces écrans permettent à l'utilisateur de visualiser la liste des médicaments (figure 6.8), d'accéder aux détails d'un médicament spécifique (figure 6.8) et d'ajouter un nouveau médicament (figure 6.8).

L'utilisateur peut entrer le nom du médicament, la dose, le type de médicament (comprimé, capsule, liquide), et les horaires de prise. Il est également possible de prendre une photo ou scanner un code QR du médicament pour une identification plus facile. Après avoir saisi les

informations nécessaires et cliqué sur le bouton "Save Medication", le médicament est ajouté à la liste et peut être géré par la suite.

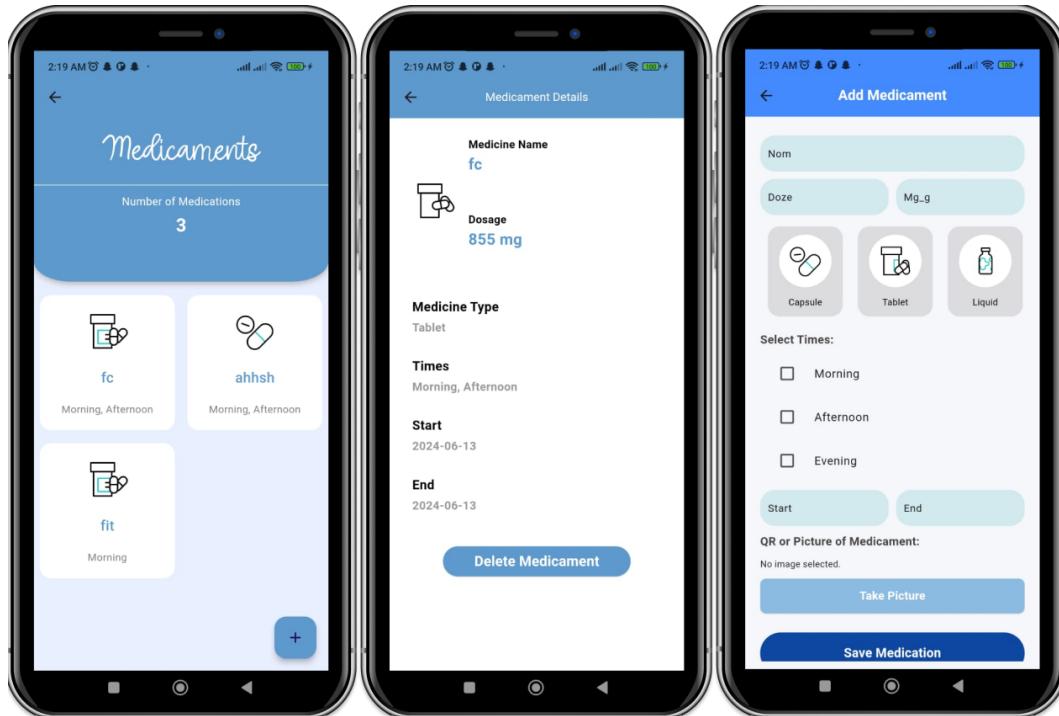


FIG. 6.8 – Screenshots of Medicaments interface.

## 6.3 Développement de l'application Web

### 6.3.1 Page d'Accueil

La page d'accueil de notre application web représente le hub central à partir duquel les utilisateurs peuvent accéder à diverses fonctionnalités et informations importantes. Elle affiche également des statistiques cruciales pour le médecin concernant la santé des patients, notamment leur pression artérielle.

- **Patient :** Affiche le nombre total de patients enregistrés. Par exemple, ici, il y a 12 patients enregistrés.
- **Rendez-vous par jours :** Indique le nombre de rendez-vous programmés pour la journée en cours. Par exemple, ici, il y a 3 rendez-vous dont 2 sont déjà passés.
- **Notifications :** Montre le nombre total de notifications, avec le nombre de notifications déjà lues. Par exemple, ici, il y a 12 notifications dont 10 ont déjà été lues.
- **Message :** Affiche le nombre total de messages reçus, avec les nouveaux messages reçus depuis la dernière heure. Par exemple, ici, il y a 9 messages dont 2 nouveaux depuis la dernière heure.
- **Statistiques des Patients :**
  - **Pression artérielle élevée :** Indique le nombre de patients ayant une pression artérielle élevée. Par exemple, ici, il y a 2 patients avec une pression artérielle élevée.

- **Pression artérielle basse :** Indique le nombre de patients ayant une pression artérielle basse. Par exemple, ici, il y a 1 patient avec une pression artérielle basse.
- **Liste des Patients :** Affiche les patients avec leurs informations comme le nom, l'email, la date de naissance, le statut des tâches et la pression artérielle. Par exemple, ici, patient2 a une pression artérielle élevée et la tâche est complétée, patient6 a une pression artérielle élevée et la tâche est en cours, et patient3 a une pression artérielle basse et la tâche est complétée.
- **Position du Médecin :** Affiche la position actuelle du médecin pour faciliter le suivi et la gestion des rendez-vous et des visites à domicile.

Voici un aperçu des éléments principaux disponibles sur la page d'accueil :

Name	Email	Date de naissance	Status	Pressure
patient2	patient2@gmail.com	2018-05-24	Completed	↑ High
patient6	patient6@gmail.com		In progress	↑ High
patient3	patient3@gmail.com	2018-05-24	Completed	↓ Low

FIG. 6.9 – Page d'Accueil de l'application web.

### 6.3.2 Page Patient

#### 1- Liste des Patients :

La liste des patients offre une vue d'ensemble des informations essentielles pour chaque patient suivi par le médecin, ainsi que leur état de santé, notamment en ce qui concerne leur pression artérielle. Cette interface permet également d'effectuer diverses actions de gestion des patients. Cette interface affiche les patients et leur état de santé actuel, avec des indicateurs pour la pression artérielle comme "High", "Medium" ou "Low". En outre, elle permet de réaliser les actions suivantes :

- **More Info :** Accéder à des informations détaillées sur le patient pour un suivi plus approfondi.
- **Delete Patient :** Supprimer un patient de la liste, ce qui peut être nécessaire en cas de doublon ou de fin de suivi médical.
- **Complete Dossier Médicale :** Compléter le dossier médical du patient avec des informations supplémentaires ou mises à jour.
- **Add Patient :** Accéder à l'interface permettant d'ajouter un nouveau patient à la liste.

Voici un aperçu des fonctionnalités principales :

Name	Email	Date de naissance	Status	Pressure
patient1	patient1@gmail.com	2024-05-19	Completed	...
patient2	patient2@gmail.com	2018-05-24	Completed	↑ High
patient3	patient3@gmail.com	2018-05-24	Completed	↓ Low
patient5	patient5@gmail.com	1984-06-12	Completed	→ Medium

FIG. 6.10 – Liste des Patients de l'application web.

## 2- Ajout de patient :

L'interface d'ajout de patient permet aux utilisateurs d'ajouter de nouveaux patients au système. Les informations de base telles que le nom d'utilisateur, l'email, et le mot de passe du patient sont collectées via un formulaire simple et intuitif.

FIG. 6.11 – Ajout de Patient.

Cette interface inclut les champs suivants :

- **Username** : Le nom du patient.
- **Email** : L'email du patient.
- **Password** : Le mot de passe pour le compte du patient.

### 6.3.3 Page de Chat

La page de chat de notre application web permet une communication en temps réel entre le médecin et les patients. Cette fonctionnalité utilise Firebase pour gérer et synchroniser les messages instantanément. Les utilisateurs peuvent voir la liste des patients disponibles sur la gauche et, à droite, la fenêtre de chat avec l'historique des messages échangés.

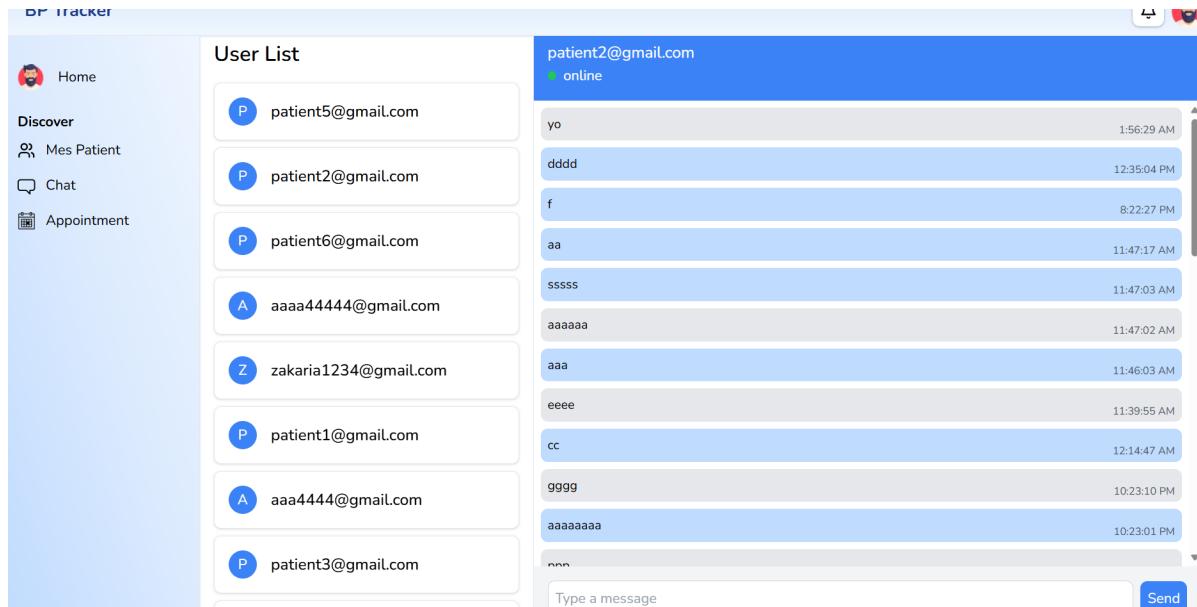


FIG. 6.12 – Page de Chat de l'application web.

Figure 6.12 montre l'interface de la page de chat, où l'utilisateur peut sélectionner un patient de la liste et communiquer via des messages texte. L'intégration de Firebase permet une mise à jour instantanée des conversations, assurant ainsi une communication fluide et efficace entre le médecin et les patients.

### 6.3.4 Rendez-vous

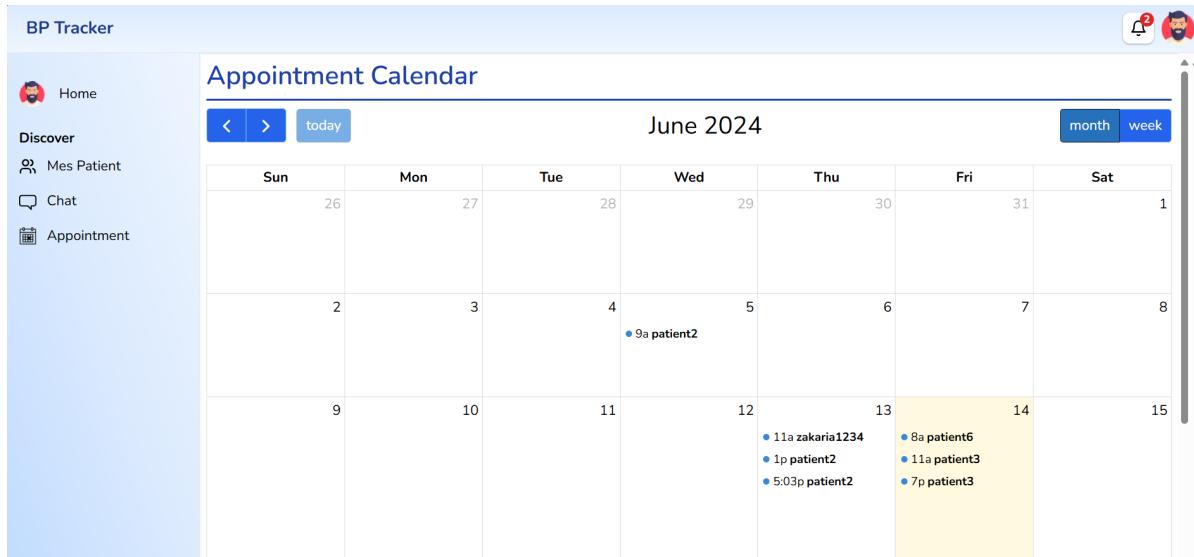


FIG. 6.13 – – Page de Rendez-vous de l'application web.

La page de rendez-vous de notre application web permet aux médecins de visualiser et de gérer les rendez-vous avec leurs patients. Le calendrier affiche les rendez-vous programmés pour le mois en cours, offrant une vue claire et organisée des consultations à venir. Cette fonctionnalité est rendue possible grâce à l'utilisation du plugin FullCalendar, disponible à l'adresse suivante : <https://fullcalendar.io/>.

Figure 6.13 montre l'interface du calendrier de rendez-vous, où chaque date peut contenir plusieurs consultations planifiées. Les utilisateurs peuvent naviguer entre les vues mensuelles et hebdomadaires pour une gestion plus détaillée de leur emploi du temps.

## 6.4 Conclusion

Dans ce chapitre, nous avons exploré le processus de développement d'une application de suivi de la tension artérielle, conçue pour être accessible sur les plateformes web et mobile. Nous avons discuté en détail de l'implémentation des fonctionnalités avancées, l'intelligence artificielle qui jouent un rôle crucial dans l'amélioration de la précision du suivi et dans la fourniture de recommandations personnalisées aux utilisateurs.

En démontrant l'interface utilisateur de nos applications frontales pour les plateformes web et mobile, nous avons mis en lumière l'importance de l'expérience utilisateur dans le succès de notre application. Ces interfaces, que nous allons présenter en détail dans le prochain chapitre, illustrent comment les utilisateurs peuvent interagir de manière intuitive et efficace avec les différentes fonctionnalités de l'application.

Dans le prochain chapitre, nous aborderons le déploiement de notre backend développé en Laravel. Nous détaillerons les étapes et les outils utilisés pour assurer une intégration transparente et une performance optimale de l'application dans un environnement de production.

# Conclusion

Notre projet vise à développer une application mobile avancée pour le suivi de la tension artérielle, intégrant des technologies modernes et l'intelligence artificielle pour une surveillance optimale de la santé. Pour ce faire, nous avons choisi d'utiliser des technologies de pointe, notamment Flutter pour le développement de l'application mobile et Next.js pour l'application web, sans recourir à Laravel pour le backend. L'objectif principal est de fournir une solution complète et intuitive permettant aux utilisateurs de surveiller leur tension artérielle, de prédire les fluctuations potentielles et de recevoir des recommandations personnalisées basées sur les données recueillies.

Le projet se divise en deux volets principaux pour répondre aux besoins des patients et des médecins. Pour les patients, l'application mobile développée avec Flutter fournira une interface conviviale pour le suivi de la tension artérielle, permettant l'enregistrement des mesures, la visualisation de l'évolution des données et la réception d'alertes et de recommandations personnalisées basées sur des prédictions d'IA. Pour les médecins, le back-office accessible via l'application web développée avec Next.js permettra de gérer les données des patients, d'accéder à des analyses détaillées fournies par l'IA et d'utiliser des outils avancés pour établir des diagnostics et recommander des traitements appropriés.

En conclusion, ce projet ambitionne de révolutionner la manière dont les utilisateurs surveillent et gèrent leur tension artérielle en intégrant des technologies avancées et une interface utilisateur optimisée. En offrant des outils précis et des recommandations personnalisées, nous espérons améliorer significativement la qualité de vie des utilisateurs en leur permettant de prendre en charge leur santé de manière autonome et éclairée.

Pour la suite, notre perspective est de déployer l'application mobile sur les plateformes de distribution telles que l'App Store et le Google Play Store, rendant ainsi notre solution accessible à un large public. En ce qui concerne l'application web, nous envisageons de la déployer sur Vercel, ce qui nous permettra de bénéficier d'une infrastructure fiable et scalable pour offrir une expérience utilisateur fluide et performante.

En intégrant ces technologies et en suivant ces étapes de déploiement, nous sommes convaincus que notre application fournira une solution robuste et efficace pour le suivi de la tension artérielle, répondant ainsi aux attentes et aux besoins des utilisateurs.

# Reference

Lien	Date de consultation
NextJS The React Framework for the Web	25/04/2024
TailwindCss	25/04/2024
Shadcn component library	25/04/2024
Flutter	01/05/2024
Laravel	30/04/2024