

N° d'ordre :

Mémoire de fin d'études déposé à l'

UNIVERSITÉ MOHAMED BOUDIAF – MSILA



FACULTÉ DES MATHÉMATIQUES ET DE L'INFORMATIQUE
DÉPARTEMENT DE L'INFORMATIQUE

Pour la satisfaction partielle des exigences du diplôme de

Master en Informatique

Spécialité : SI-Genie Logiciel — ID-Optimisation

Par

**LADJEL AYOUB
KHODJA BELKACEM**

Titre du mémoire

**Conception et Développement d'une Application
Mobile et Web Intelligente d'Aide pour les Patients
Atteints de Maladies Chroniques**

Sous la direction de

M. Tahri Zohir

Composition du jury

M. Debbi Imad Eddine

Université de M'sila

Président

M. Amri Said

Université de M'sila

Examineur

JUIN, 2026

Dédicace

Nous dédions ce travail à nos très chers parents, qui ont tout sacrifié pour nous offrir le meilleur chemin dans la vie, et dont les prières et l'amour nous ont été le soutien le plus précieux tout au long de cette aventure académique.

À nos familles, frères et surs, qui n'ont jamais cessé de croire en nous.

À notre encadreur M. Tahari Zohir, pour sa confiance, sa disponibilité et ses précieux conseils.

À tous nos amis et camarades de promotion, compagnons de route dans cette aventure universitaire.

À chaque patient qui se bat chaque jour en silence contre une maladie chronique — nous espérons que ce travail contribuera, même modestement, à améliorer son quotidien.

Khodja Belkasem & Ladjel Ayoub

Remerciements

Au terme de ce travail, nous souhaitons exprimer notre profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à sa réalisation.

Nos remerciements les plus sincères vont en premier lieu à notre directeur de mémoire, **M. Tahari Zohir**, pour sa disponibilité constante, ses orientations pertinentes et sa rigueur scientifique. Ses conseils éclairés ont été déterminants dans la réussite de ce projet.

Nous tenons également à remercier les membres du jury pour l'honneur qu'ils nous font en acceptant d'évaluer ce mémoire et pour le temps précieux qu'ils y consacrent.

Nos remerciements vont aussi à l'ensemble du corps enseignant du Département d'informatique de l'Université de Msila, qui nous a dispensé une formation de qualité tout au long de notre cursus.

Enfin, nous remercions toutes les personnes qui ont participé aux tests d'acceptation de l'application, dont les retours ont été précieux pour améliorer la solution développée.

Résumé / Abstract

Résumé

Ce mémoire présente la conception et le développement d'une application mobile et d'un site web intégrés dédiés au suivi des patients atteints de maladies chroniques : le diabète sucré, l'hypertension artérielle et l'obésité. Le système propose l'enregistrement des mesures vitales quotidiennes, leur visualisation graphique interactive, un système de rappels médicamenteux, un système de rappels de rendez-vous médical (avec notifications J-1 et H-1), et la génération de rapports médicaux standardisés au format PDF côté client. Développé avec **Next.js 15** (API Routes intégrées), React, **MongoDB** et Workbox (PWA) ; la génération PDF est assurée par jsPDF et jspdf-autotable.

Mots-clés : Application mobile, Santé numérique, Diabète, Hypertension artérielle, Obésité, Rappel médicamenteux, Rendez-vous médical, Next.js, MongoDB, jsPDF, PWA.

Abstract

This thesis presents the design and development of an integrated mobile application and website dedicated to monitoring patients with chronic diseases : diabetes mellitus, arterial hypertension and obesity. The system offers daily vital measurements recording, interactive graphical visualization, an intelligent medication reminder system, a medical appointment reminder system (with J-1 and H-1 automatic notifications), and professional PDF report generation on the client side. Built with Next.js 15 (integrated API Routes), React, MongoDB and Workbox (PWA); PDF generation uses jsPDF and jspdf-autotable.

Keywords : Mobile application, Digital health, Diabetes, Hypertension, Obesity, Medication reminder, Medical appointment, Next.js, MongoDB, jsPDF, PWA.

ملخص

يقدم هذا المشروع تصميم وتطوير تطبيق محمول وموقع ويب متكاملين، مخصصين لمتابعة المرضى المصابين بالأمراض المزمنة: داء السكري، ارتفاع ضغط الدم الشرياني، والسمنة. يوفر النظام تسجيل القياسات الحيوية اليومية، وعرضها بياناً بشكل تفاعلي، ونظام تذكير بالأدوية، ونظام تذكير بالمواعيد الطبية (مع إشعارات قبل يوم وقبل ساعة)، بالإضافة إلى توليد تقارير طبية موحدة بصيغة PDF من جانب العميل. تم تطوير التطبيق باستخدام Next.js 15 (مع واجهات برمجة التطبيقات المدججة)، و React، و MongoDB، و Workbox (تطبيق ويب تقديمي)، أما توليد ملفات PDF فيتم باستخدام jsPDF وjspdf-autotable.

الكلمات المفتاحية: تطبيق محمول، الصحة الرقمية، داء السكري، ارتفاع ضغط الدم الشرياني، السمنة، تذكير بالأدوية، موعد طبي، Next.js، MongoDB، jsPDF، تطبيق ويب تقديمي.

Table des matières

Dédicace	i
Remerciements	ii
Résumé / Abstract	iii
Table des matières	ix
Liste des tableaux	x
Liste des figures	xii
Liste des abréviations	xiii
Introduction Générale	1
1 Introduction Scientifique du Projet	3
1.1 Contexte Épidémiologique en Algérie	3
1.1.1 Données Épidémiologiques Nationales	3
1.1.2 Limites du Système de Santé Actuel	4
1.2 Problématique	4
1.3 Objectifs de la Recherche	5
1.4 Importance de la Recherche	5
1.4.1 Importance sanitaire	5
1.4.2 Importance technologique	5
1.4.3 Importance sociale	6
1.5 Méthodologie Adoptée	6
1.5.1 Approche de Recherche & Développement (R&D)	6
1.5.2 Méthode Agile / Scrum	6

1.6	Structure du Mémoire	7
1.7	Conclusion	7
2	État de l'Art	8
2.1	Introduction	8
2.2	Les Maladies Chroniques : Définition et Statistiques	8
2.3	Le Diabète Sucré	9
2.3.1	Définition et Classification	9
2.3.2	Paramètres de Mesure dans l'Application	9
2.3.3	Complications	9
2.4	L'Hypertension Artérielle	9
2.4.1	Définition et Classification (ESC/ESH 2018)	10
2.4.2	Paramètres de Mesure	10
2.4.3	Complications	10
2.5	L'Obésité	10
2.5.1	Définition et Classification OMS	10
2.5.2	Paramètres de Suivi	11
2.6	Les Applications de Santé Numériques (MyHealth)	11
2.6.1	Définition et Classification des Applications MyHealth	11
2.6.2	Impact Clinique Démontré	12
2.6.3	Contexte des Applications MyHealth en Algérie	12
2.7	Comparaison avec les Applications Existantes	13
2.8	Conclusion	13
3	Analyse et Conception du Système	14
3.1	Introduction	14
3.2	Cahier des Charges	14
3.2.1	Présentation du Projet	14
3.2.2	Parties Prenantes	14
3.3	Exigences Fonctionnelles et Non Fonctionnelles	15
3.3.1	Exigences Fonctionnelles	15
3.3.2	Exigences Non Fonctionnelles	17
3.4	Diagramme de Cas d'Utilisation (Use Case Diagram)	17
3.5	Diagramme de Classes (Class Diagram)	19

3.6	Diagrammes de Séquence	21
3.6.1	Scénario 1 : Saisie d'une Mesure Glycémique et Alerte	22
3.6.2	Scénario 2 : Gestion d'un Rendez-vous Médecin	23
3.6.3	Scénario 3 : Génération d'un Rapport PDF	24
3.7	Conception de la Base de Données (MongoDB)	25
3.8	Conception des Interfaces (Wireframes)	26
3.8.1	E02	27
3.8.2	E04	27
3.8.3	E05	28
3.8.4	E10	29
3.8.5	E11	29
3.9	Conclusion	30
4	Développement et Réalisation	31
4.1	Introduction	31
4.2	Présentation Générale du Projet	31
4.2.1	Objectif	31
4.2.2	Fonctionnalités Principales	31
4.2.3	Public Cible	32
4.3	Architecture du Système	32
4.3.1	Vue d'Ensemble — Architecture en Couches	33
4.3.2	Flux de Données Typique	33
4.4	Pile Technologique (Stack)	34
4.4.1	Technologies Front-end	34
4.4.2	Technologies Back-end	36
4.4.3	Outils de Développement	37
4.5	Justification des choix techniques	37
4.6	Techniques et Patrons de Conception	38
4.6.1	Architecture en Couches (Layered Architecture)	38
4.6.2	Pattern Container / Presentational	38
4.6.3	Authentification Stateless par JWT	39
4.6.4	Hachage bcrypt avec Hook pre-save	39
4.6.5	Schéma Flexible (Mixed) pour les Enregistrements de Santé	39

4.6.6	Calculs Automatiques Côté Serveur	39
4.6.7	Mise en Cache de la Connexion MongoDB	40
4.6.8	Mock Scheduler (Rappels Simulés)	40
4.6.9	Validation à Plusieurs Niveaux	40
4.6.10	Internationalisation et RTL	40
4.7	Modèles de Données (MongoDB / Mongoose)	41
4.7.1	Modèle User — <code>src/models/User.ts</code>	41
4.7.2	Modèle HealthRecord — <code>src/models/HealthRecord.ts</code>	41
4.7.3	Modèle Medication — <code>src/models/Medication.ts</code>	42
4.7.4	Modèle Appointment — <code>src/models/Appointment.ts</code>	42
4.7.5	Modèle Notification — <code>src/models/Notification.ts</code>	42
4.8	Routes API REST	44
4.9	Fonctionnalités — Localisation dans le Code	44
4.9.1	Authentification JWT + bcrypt	45
4.9.2	Module Diabète	45
4.9.3	Module Tension Artérielle	45
4.9.4	Module Obésité	46
4.9.5	Module Médicaments	46
4.9.6	Module Rendez-vous	46
4.9.7	Système de Notifications (Mock Scheduler)	46
4.9.8	Tableau de Bord	47
4.9.9	Statistiques	47
4.9.10	Export PDF	47
4.10	Calculs Médicaux et Seuils	48
4.10.1	Seuils Diabète — ADA	48
4.10.2	Seuils Tension — ESC/ESH 2018[5]	48
4.10.3	Seuils Obésité — OMS	49
4.11	Sécurité	49
4.11.1	Mesures Implémentées	49
4.11.2	Améliorations Futures Recommandées	49
4.12	Conclusion	50

5 Tests et Évaluation	51
5.1 Introduction	51
5.2 Stratégie de Test	51
5.3 Tests Unitaires et d'Intégration	52
5.4 Test d'Acceptation Utilisateur (UAT)	52
5.4.1 Profil des 25 Participants	52
5.4.2 Résultats de Satisfaction	53
5.5 Évaluation des Performances et de la Sécurité	53
5.5.1 Score Google Lighthouse	53
5.5.2 Vérifications de Sécurité (OWASP Top 10)	54
5.6 Résultats et Discussion	54
5.6.1 Analyse des Résultats de Performance	54
5.6.2 Analyse des Retours Utilisateurs (UAT)	55
5.6.3 Comparaison avec l'État de l'Art	55
5.7 Conclusion	55
Conclusion Générale et Perspectives	56
Bibliographie	58
A Algorithmes de Classification Médicale	60
B Questionnaire UAT	62

Liste des Tableaux

N°	Titre	Page
Table 1.1	Planification des sprints de développement (Agile/Scrum)	7
Table 2.1	Statistiques mondiales et algériennes des trois pathologies	8
Table 2.2	Paramètres glycémiques et seuils d'alerte de l'application	9
Table 2.3	Classification de l'hypertension artérielle (ESC/ESH 2018)	10
Table 2.4	Classification de l'obésité selon l'OMS	11
Table 2.5	Comparaison fonctionnelle : notre application vs concurrents	13
Table 3.1	Exigences fonctionnelles du système	15
Table 3.2	Exigences non fonctionnelles du système	17
Table 3.3	Description des écrans principaux (Wireframes)	26
Table 4.1	Architecture en couches du système	33
Table 4.2	Technologies front-end utilisées dans le projet	34
Table 4.3	Technologies back-end utilisées dans le projet	36
Table 4.4	Outils de développement et de déploiement	37
Table 4.5	Modèle User champs et contraintes	41
Table 4.6	Routes API REST du système	43
Table 4.7	Seuils diagnostiques du diabète (ADA)	47
Table 4.8	Classification de la tension artérielle (ESC/ESH 2018)	47

Table 4.9	. Classification de l'obésité selon l'OMS (IMC)	48
Table 5.1	Récapitulatif des résultats des tests automatisés	51
Table 5.2	Résultats de satisfaction utilisateur (UAT 25 participants)	52
Table 5.3	Résultats Google Lighthouse de l'application (version production)	52
Table B.1	Questionnaire du test d'acceptation utilisateur (UAT)	61

Liste des Figures

N ^o	Titre	Page
Figure 1	Diagramme de cas d'utilisation — Vue globale du système	18
Figure 2	Diagramme de cas d'utilisation — Module Patient (détail)	19
Figure 3	Diagramme de classes du système	20
Figure 4	Diagramme de classes — Relations et multiplicités	21
Figure 5	Diagramme de séquence — Saisie d'une mesure glycémique et alerte	22
Figure 6	Diagramme de séquence — Gestion d'un rendez-vous médecin	23
Figure 7	Diagramme de séquence — Génération d'un rapport PDF	24
Figure 8	Schéma de la base de données MongoDB	25
Figure 9	Inscription/Connexion E-mail, mot de passe, sélection des pathologies	27
Figure 10	Tableau de bord Résumé mesures, rappels à venir, conseil du jour	27
Figure 11	Module Diabète: Formulaire saisie, graphique, historique, statistiques	28
Figure 12	Génération PDF Sélection période, téléchargement	29
Figure 13	Langue, Changer de thème, Se déconnecter	29

Liste des Abréviations

Abréviation	Signification complète
ADA	American Diabetes Association
API	Application Programming Interface
bcrypt	Blowfish Crypt (algorithme de hachage de mots de passe)
BMI	Body Mass Index (voir IMC)
CI/CD	Continuous Integration / Continuous Deployment
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DBP	Diastolic Blood Pressure (Pression artérielle diastolique)
DG	Diabète Gestationnel
DT1	Diabète de Type 1
DT2	Diabète de Type 2
ESC	European Society of Cardiology
ESH	European Society of Hypertension
FBG	Fasting Blood Glucose (Glycémie à jeun)
FCM	Firebase Cloud Messaging
HbA1c	Hémoglobine glyquée (Glycated Haemoglobin A1c)
HR	Heart Rate (Fréquence cardiaque)

Abréviation	Signification complète
HTA	Hypertension Artérielle
HTTPS	HyperText Transfer Protocol Secure
IDF	International Diabetes Federation
IDE	Integrated Development Environment
IMC	Indice de Masse Corporelle
INSP	Institut National de Santé Publique
JWT	JSON Web Token
MAP	Mean Arterial Pressure (Pression artérielle moyenne)
MCNT	Maladies Chroniques Non Transmissibles
MyHealth	Mobile Health (Santé mobile)
MVC	Model-View-Controller
NoSQL	Not only SQL
ODM	Object Document Mapper
OMS	Organisation Mondiale de la Santé
OWASP	Open Web Application Security Project
PAD	Pression Artérielle Diastolique
PAM	Pression Artérielle Moyenne
PAS	Pression Artérielle Systolique
PDF	Portable Document Format
PPBG	Post-Prandial Blood Glucose (Glycémie post-prandiale)
PWA	Progressive Web Application
R&D	Recherche et Développement
RBG	Random Blood Glucose (Glycémie aléatoire)

Abréviation	Signification complète
REST	Representational State Transfer
RGPD	Règlement Général sur la Protection des Données
RTH	Rapport Tour de taille / Hanches
RTL	Right To Left (écriture de droite à gauche)
SBP	Systolic Blood Pressure (Pression artérielle systolique)
SLA	Service Level Agreement
SQL	Structured Query Language
SSG	Static Site Generation
SSR	Server-Side Rendering
TLS	Transport Layer Security
UAT	User Acceptance Testing (Test d'acceptation utilisateur)
UI	User Interface (Interface utilisateur)
UML	Unified Modeling Language
UX	User Experience (Expérience utilisateur)
VP	Visual Paradigm
WHR	Waist-to-Hip Ratio (voir RTH)
XSS	Cross-Site Scripting

Introduction Générale

Le monde contemporain est confronté à une transition épidémiologique majeure : le fardeau des maladies infectieuses aiguës cède progressivement la place aux maladies chroniques non transmissibles (MCNT), qui mobilisent une part croissante des ressources sanitaires mondiales. Au premier rang de ce défi se trouvent trois pathologies intimement liées : le **diabète sucré**, l'**hypertension artérielle** et l'**obésité**, que l'Organisation Mondiale de la Santé qualifie de «pandémies silencieuses».

En Algérie, les données épidémiologiques nationales sont préoccupantes : environ 15 % des adultes souffrent de diabète, plus de 35 % présentent une hypertension artérielle, et plus de 50 % de la population adulte est en surpoids ou obèse (Ministère de la Santé, 2023)[9]. Ces chiffres alarmants mettent en lumière les limites d'un système de santé orienté vers le curatif et la nécessité urgente de développer des outils favorisant la prévention et l'auto-prise en charge.

Avec un taux de pénétration des smartphones dépassant les 80 % en Algérie, les applications de santé mobile (My Health) constituent un vecteur stratégique pour améliorer l'observance thérapeutique, le suivi des paramètres vitaux et la communication médecin-patient.

C'est dans cette perspective que s'inscrit le présent projet, réalisé par **Khodja Belkasssem** et **Ladjel Ayoub**, sous la direction de **M. Tahari Zohir** : concevoir et développer une plateforme numérique complète permettant aux patients atteints de maladies chroniques de surveiller leurs constantes biologiques, de suivre leurs traitements et rendez-vous médicaux, et de générer des rapports médicaux partageables avec leur médecin.

Structure du mémoire

Ce mémoire est structuré en cinq chapitres :

- **Chapitre 1** : Ce chapitre pose les fondements scientifiques de notre travail : il définit la problématique, expose les objectifs, précise l'importance du projet et détaille la méthodologie adoptée.
- **Chapitre 2** : Ce chapitre examine les trois pathologies chroniques ciblées avec leurs données médicales et statistiques, puis présente l'écosystème des applications de santé numériques existantes, avant de positionner notre solution par rapport aux concurrents.
- **Chapitre 3** : Ce chapitre présente la démarche d'analyse et la conception complète du système : cahier des charges, exigences, diagrammes UML (Use Case, Classes, Séquence), architecture, base de données et interfaces.
- **Chapitre 4** : Ce chapitre documente la réalisation technique du projet, en présentant en détail les outils utilisés, les choix techniques pour chaque couche du système, et des extraits de code illustrant les fonctionnalités clés.
- **Chapitre 5** : Ce chapitre présente la démarche de test adoptée pour valider la qualité du système : tests automatisés, test d'acceptation utilisateur (UAT), et évaluation des performances et de la sécurité.

CHAPITRE 1

Introduction Scientifique du Projet

1.1 Contexte Épidémiologique en Algérie

L'Algérie fait face à une transition épidémiologique profonde. Selon le Ministère de la Santé (2023), les maladies chroniques non transmissibles (MCNT) représentent désormais la principale cause de morbidité et de mortalité dans le pays. Cette évolution s'explique par plusieurs facteurs convergents : l'urbanisation accélérée, la sédentarité croissante, les changements alimentaires, le vieillissement de la population et l'amélioration de l'espérance de vie.

1.1.1 Données Épidémiologiques Nationales

Les statistiques publiées par l'Institut National de Santé Publique (INSP) révèlent une situation préoccupante :

- **Diabète** : La prévalence du diabète de type 2 a doublé en quinze ans, passant de 7 % en 2008 à près de 15 % en 2023 [9], soit plus de 3,5 millions de personnes atteintes. Cette progression place l'Algérie parmi les pays à prévalence élevée dans la région MENA.
- **Hypertension artérielle** : Selon les enquêtes nationales, plus de 35 % des adultes algériens souffrent d'hypertension artérielle [9], dont près de la moitié ignore sa pathologie. Le dépistage tardif constitue l'un des principaux défis de santé publique.
- **Obésité et surpoids** : Le taux de surpoids dépasse 50 % chez les adultes[9], avec une progression alarmante chez les femmes et les jeunes urbains. L'indice de masse corporelle moyen de la population algérienne adulte a augmenté de 2,3 points en dix ans.

- **Comorbidités** : Plus de 60 % des patients diabétiques présentent simultanément une hypertension artérielle, et 40 % souffrent d'obésité associée, créant des tableaux cliniques complexes nécessitant un suivi multiparamétrique rigoureux[9].

1.1.2 Limites du Système de Santé Actuel

Le système de santé algérien présente des limites structurelles face aux maladies chroniques : une approche curative dominante qui mobilise l'essentiel des ressources vers les soins aigus, une densité médicale insuffisante dans certaines wilayas (1,8 médecin/1000 habitants), l'absence de plateformes numériques arabophones dédiées, et une observance thérapeutique faible — moins de 40 % des patients chroniques respectent leur traitement sur le long terme (INSP, 2023)[9].

C'est pour combler ces lacunes que notre application a été conçue, en exploitant le taux de pénétration des smartphones (80 %) pour transformer chaque téléphone en assistant médical personnel.

1.2 Problématique

Les patients atteints de maladies chroniques en Algérie font face à de multiples défis quotidiens :

- Difficulté à maintenir un suivi rigoureux de leurs paramètres vitaux (glycémie, tension, poids).
- Fréquents oublis de prise médicamenteuse, responsables de complications évitables.
- Absence d'outil centralisé pour communiquer avec le médecin traitant.
- Carence en application de santé en langue arabe adaptées au contexte algérien.
- Difficulté à mémoriser les rendez-vous médicaux, surtout chez les personnes âgées.
- Aucune solution gratuite et unifiée couvrant les trois pathologies majeures.

Problématique Centrale

Comment une plateforme numérique intégrée combinant une application mobile et un site web peut-elle contribuer à améliorer l'autogestion des patients atteints de ma-

ladies chroniques, en suivant les paramètres biologiques, en gérant les traitements médicamenteux, en organisant les rendez-vous médicaux et en préparant des rapports médicaux utilisables tant pour les patients que pour les professionnels de la santé ?

1.3 Objectifs de la Recherche

1. Concevoir une application mobile (PWA) et un site web pour le suivi des trois pathologies.
2. Développer un système d'enregistrement des mesures avec calculs automatiques (MAP, IMC) et visualisation graphique interactive.
3. Développer un système de rappels médicamenteux personnalisés.
4. Mettre en place un système de rappels de rendez-vous médicaux (notifications J-1 et H-1).
5. Générer des rapports médicaux standardisés au format PDF.
6. Concevoir une interface utilisateur en arabe (RTL), intuitive et accessible à toutes les tranches d'âge.
7. Assurer la sécurité des données médicales (bcrypt, JWT, HTTPS, OWASP).

1.4 Importance de la Recherche

1.4.1 Importance sanitaire

L'application contribue à l'auto-prise en charge du patient et à l'amélioration de l'observance thérapeutique, dont la défaillance est responsable d'environ 50 % des hospitalisations évitables (OMS, 2023)[1].

1.4.2 Importance technologique

Le projet constitue un modèle d'application des technologies web modernes (**Next.js**, **Node.js**, **MongoDB**) à des solutions de santé numérique locales.

1.4.3 Importance sociale

La plateforme en langue arabe avec RTL rend les outils de gestion de santé accessibles à une population plus large, y compris les personnes âgées peu familiarisées avec les interfaces anglophones.

1.5 Méthodologie Adoptée

1.5.1 Approche de Recherche & Développement (R&D)

La démarche combine une analyse théorique approfondie et une réalisation technique concrète, évaluée empiriquement auprès d'utilisateurs réels.

1.5.2 Méthode Agile / Scrum

La gestion du développement logiciel s'appuie sur la méthodologie Agile/Scrum, organisée en 7 sprints de 2 semaines chacun.

Table 1.1. Planification des sprints de développement (Agile/Scrum)

Sprint	Activités réalisées	Durée
Sprint 1	Configuration de l'environnement — Structure Next.js — Authentification JWT	2 sem.
Sprint 2	Module diabète : API Routes + MongoDB + Interface + Graphiques ApexCharts	2 sem.
Sprint 3	Module tension + obésité : API Routes + Interfaces + Calculs automatiques (PAM, IMC)	2 sem.
Sprint 4	Rappels médicamenteux + Rendez-vous médecin + Mock Scheduler (notifications in-app)	2 sem.
Sprint 5	Graphiques ApexCharts avancés + Tableau de bord + Alertes valeurs critiques	2 sem.
Sprint 6	Génération PDF côté client (jsPDF + jspdf-autotable) + Conversion PWA (Workbox)	2 sem.
Sprint 7	Tests (Jest + Supertest) + Optimisation + Déploiement (Vercel)	2 sem.

1.6 Structure du Mémoire

Ce mémoire est structuré en cinq chapitres : **Chapitre I** : cadre scientifique. **Chapitre II** : état de l'art médical et technologique. **Chapitre III** : analyse et conception (UML avec **Visual Paradigm**, BDD, wireframes). **Chapitre IV** : développement technique (**Visual Paradigm**, Next.js, Node.js, MongoDB, PWA, PDF). **Chapitre V** : tests et évaluation.

1.7 Conclusion

Ce chapitre a posé les fondements scientifiques du projet. Le chapitre suivant examinera en détail les trois pathologies ciblées et l'écosystème des applications de santé numériques.

CHAPITRE 2

État de l’Art

2.1 Introduction

Ce chapitre examine les trois pathologies chroniques ciblées avec leurs données médicales et statistiques, puis présente l’écosystème des applications de santé numériques existantes, avant de positionner notre solution par rapport aux concurrents.

2.2 Les Maladies Chroniques : Définition et Statistiques

L’OMS définit les maladies chroniques comme des affections de longue durée évoluant lentement, responsables de 74 % des décès mondiaux (41 millions par an, OMS 2023)[1].

Table 2.1. Statistiques mondiales et algériennes des trois pathologies ciblées[2] [3] [10]

Maladie	Patients (mon-dial)	Décès/an	Coût mondial	Algérie
Diabète	537 millions	6,7 millions	966 Md\$	≈15 %
Hypertension	1,28 milliard	10,8 millions	370 Md\$	≈35 %
Obésité	650 millions	2,8 millions	2 000 Md\$	≈24 %

2.3 Le Diabète Sucré

2.3.1 Définition et Classification

Le diabète sucré est une maladie métabolique chronique caractérisée par une **hyperglycémie chronique** résultant d'un défaut de sécrétion ou d'action de l'insuline. Il se décline en trois types principaux : DT1 (auto-immun, insulinothérapie à vie, 5–10 %), DT2 (insulinorésistance, 90–95 % des cas, lié au mode de vie) et gestationnel (pendant la grossesse).

2.3.2 Paramètres de Mesure dans l'Application

Table 2.2. Paramètres glycémiques et seuils d'alerte de l'application[4]

Paramètre	Abrév.	Normale	Prédiabète	Diabète
Glycémie à jeun	FBG	70–99 mg/dL	100–125	≥ 126
Glycémie post-prandiale	PPBG	< 140 mg/dL	140–199	≥ 200
Hémoglobine glyquée	HbA _{1c}	< 5,7 %	5,7–6,4 %	$\geq 6,5$ %
Glycémie aléatoire	RBG	< 200 mg/dL	—	≥ 200 + sympt.

Note explicative : L'abréviation « *sympt.* » réfère à la présence de manifestations cliniques évocatrices du diabète, telles que la polydipsie (soif excessive), la polyurie (urines fréquentes) ou une asthénie inexpliquée. Selon les recommandations cliniques, une glycémie aléatoire ≥ 200 mg/dL [4], lorsqu'elle est associée à ces symptômes, constitue un critère diagnostique déterminant.

2.3.3 Complications

Rétinopathie (1^{re} cause de cécité), néphropathie, neuropathie, maladies cardiovasculaires (risque $\times 2$ –4) et pied diabétique (50–70 % des amputations non traumatiques).

2.4 L'Hypertension Artérielle

2.4.1 Définition et Classification (ESC/ESH 2018)

L'Hypertension Artérielle (L'HTA) est définie par une SBP ≥ 140 mmHg et/ou une DBP ≥ 90 mmHg de façon persistante [5].

2.4.2 Paramètres de Mesure

Pression systolique (SBP), diastolique (DBP), fréquence cardiaque (FC), pression artérielle moyenne (PAM) calculée automatiquement.

Table 2.3. Classification de l'hypertension artérielle selon ESC/ESH 2018[5]

Catégorie	Systolique	Diastolique	Prise en charge
Optimale	< 120	< 80	Surveillance annuelle
Normale haute	130–139	85–89	Mode de vie
HTA grade 1	140–159	90–99	Mode de vie $\pm$ médicament
HTA grade 2	160–179	100–109	Médicaments obligatoires
HTA grade 3	≥ 180	≥ 110	Urgence thérapeutique

La pression artérielle moyenne (PAM) est calculée automatiquement :

$$PAM = DBP + \frac{SBP - DBP}{3}$$

2.4.3 Complications

AVC (54 % des cas liés à l'HTA), infarctus du myocarde, insuffisance cardiaque, insuffisance rénale chronique, rétinopathie hypertensive.

2.5 L'Obésité

2.5.1 Définition et Classification OMS

$$IMC \text{ (kg/m}^2\text{)} = \frac{\text{Poids (kg)}}{\text{Taille}^2 \text{ (m)}}$$

2.5.2 Paramètres de Suivi

Poids (kg), taille (cm, unique à l'inscription), IMC calculé automatiquement, tour de taille (cm), rapport taille/hanche (RTH), poids cible (IMC cible = 24,9).

Table 2.4. Classification de l'obésité selon l'OMS (IMC)[10]

Catégorie	IMC (kg/mš)	Risque	Intervention
Insuffisance pondérale	< 18,5	Élevé	Consultation nutritionnelle
Corpulence normale	18,5–24,9	Faible	Mode de vie sain
Surpoids	25,0–29,9	Léger	Alimentation + activité physique
Obésité grade I	30,0–34,9	Modéré	Prise en charge médicale
Obésité grade II	35,0–39,9	Sévère	Prise en charge intensive
Obésité morbide	≥40,0	Très sévère	Traitement ± chirurgie bariatrique

2.6 Les Applications de Santé Numériques (MyHealth)

Le marché mondial des applications de santé dépasse **200 milliards de dollars** (OMS, 2023)[1]. Les études montrent une amélioration de l'observance médicamenteuse jusqu'à 60 % grâce aux rappels via application.

2.6.1 Définition et Classification des Applications MyHealth

Selon l'OMS (2023), les applications de santé mobile (MyHealth) désignent toute pratique médicale ou de santé publique reposant sur des appareils mobiles tels que les téléphones portables, les tablettes et les montres connectées. On distingue plusieurs catégories fonctionnelles :

- **Applications de monitoring** : permettent la saisie et le suivi régulier de paramètres biologiques (glycémie, tension artérielle, poids). Elles constituent le cur de notre solution.
- **Applications de rappels médicamenteux** : réduisent les oublis de prise médicamenteuse grâce à des notifications programmées.

- **Applications de téléconsultation** : facilitent la communication à distance entre le patient et son médecin.
- **Applications de coaching santé** : proposent des recommandations personnalisées en matière d'alimentation et d'activité physique.

2.6.2 Impact Clinique Démontré

Les données probantes issues de la littérature scientifique confirment l'efficacité des applications MyHealth dans la gestion des maladies chroniques :

- Une méta-analyse publiée dans le *Journal of Medical Internet Research* (Zhou et al., 2022) portant sur 47 essais cliniques randomisés démontre que l'utilisation d'applications de suivi du diabète réduit le taux d'HbA1c de 0,5 à 0,8 point en moyenne, un résultat cliniquement significatif.[6]
- L'étude d'Omboni et al. (2021) sur l'impact de la télémédecine pour l'hypertension rapporte une réduction de 7 % des événements cardiovasculaires majeurs chez les patients utilisant des outils numériques de suivi.[7]
- Les rappels médicamenteux via application réduisent les hospitalisations évitables liées à la non-observance de 30 à 50 % selon les données de l'OMS (2023).[1]

2.6.3 Contexte des Applications MyHealth en Algérie

Malgré le potentiel considérable des applications MyHealth, leur adoption en Algérie reste limitée par plusieurs obstacles : la quasi-totalité des applications disponibles sur les stores sont en anglais ou en français standard, sans adaptation au dialecte local ni support de l'arabe classique avec écriture droite-à-gauche (RTL). De plus, les rares applications arabophones disponibles ne couvrent qu'une seule pathologie à la fois et ne proposent pas de génération de rapports PDF partageables avec le médecin traitant, fonctionnalité pourtant essentielle dans le contexte algérien où la relation médecin-patient repose encore largement sur le document papier.

2.7 Comparaison avec les Applications Existantes

Table 2.5. Comparaison fonctionnelle : notre application vs applications existantes

Fonctionnalité	MySugr	BP+	Healthify	Glucosio	Notre App
Suivi diabète	✓	✗	~	✓	✓
Suivi tension	✗	✓	~	✗	✓
Suivi obésité (IMC)	✗	✗	✓	✗	✓
Rappels médicamenteux	Payant	✗	~	✗	✓ Gratuit
Rappels RDV médecin	✗	✗	✗	✗	✓ Unique
Graphiques avancés	✓	~	✓	~	✓
Rapports PDF	Payant	✗	~	✗	✓ Gratuit
Support arabe (RTL)	✗	✗	✗	✗	✓
Entièrement gratuit	✗	~	~	✓	✓

Légende du tableau :

✓ : Fonctionnalité disponible.

✗ : Fonctionnalité indisponible.

~ : Fonctionnalité partielle ou limitée.

Payant : Fonctionnalité nécessitant un abonnement.

Positionnement unique

Notre application se distingue par l'intégration des trois pathologies au sein d'une plateforme unifiée, offrant un support en arabe avec interface RTL, la génération gratuite de rapports PDF, ainsi qu'un système de rappel des rendez-vous médicaux, une fonctionnalité absente des solutions concurrentes analysées.

2.8 Conclusion

L'état de l'art confirme l'existence d'un vide réel dans l'écosystème des applications de santé arabophones. Le chapitre suivant présentera l'analyse et la conception complète du système développé pour combler ce vide.

CHAPITRE 3

Analyse et Conception du Système

3.1 Introduction

Ce chapitre présente la démarche d'analyse et la conception complète du système : cahier des charges, exigences, diagrammes UML réalisés avec **Visual Paradigm** (Use Case, Classes, Séquence), architecture, base de données et interfaces.

3.2 Cahier des Charges

3.2.1 Présentation du Projet

Le projet consiste à développer une plateforme numérique de santé comprenant un **site web responsive** et une **application mobile installable (PWA)**, accessible depuis tout appareil connecté.

3.2.2 Parties Prenantes

- **Patient** (acteur unique du système) : saisit ses mesures biologiques quotidiennes, reçoit ses rappels médicamenteux et de rendez-vous, consulte ses graphiques et statistiques, génère et partage ses rapports PDF. Toutes les fonctionnalités du système lui sont directement destinées.
- **Médecin** : bien qu'il n'utilise pas directement l'application, le médecin constitue un acteur externe important du système. Il exploite les rapports PDF générés afin d'analyser l'évolution des paramètres biologiques du patient et d'adapter les traitements médicaux.

Absence d'acteur Administrateur

Le périmètre fonctionnel de cette version (v1.0) ne comprend pas de rôle administrateur. Chaque compte est auto-géré par le patient lui-même (inscription, modification du profil, suppression). La mise à jour des seuils médicaux (ADA, ESC/ESH) est gérée directement dans le code source (`src/utils/health-thresholds.ts`), dans l'attente d'un portail d'administration prévu en v2.0.

3.3 Exigences Fonctionnelles et Non Fonctionnelles

3.3.1 Exigences Fonctionnelles

Table 3.1. Exigences fonctionnelles du système

ID	Description	Priorité
F01	Gestion du compte : inscription sécurisée, authentification (email + mot de passe), modification du profil médical	Critique
F02	Saisie des mesures glycémiques (FBG, PPBG, HbA _{1c} , RBG) avec date, heure et classification automatique	Critique
F03	Saisie des mesures tensionnelles (SBP, DBP, FC) avec calcul automatique de la PAM et classification ESC/ESH	Critique
F04	Saisie des données pondérales (poids, IMC calculé auto, tour de taille, RTH, poids cible)	Critique
F05	Visualisation graphique interactive (journalier / hebdo / mensuel / annuel) avec lignes de référence	Critique
F06	Rappels médicamenteux : ajout (nom, dosage, fréquence, horaires), modification, suppression, historique des prises	Critique

(suite Tableau 3.1)

ID	Description	Priorité
F07	Rappels rendez-vous médecin : planification (médecin, spécialité, lieu, date, heure), notification automatique J-1 et H-1	Critique
F08	Génération de rapports PDF (mesures, graphiques, statistiques, médicaments, RDV)	Critique
F09	Alertes en temps réel pour les valeurs biologiques critiques (hypoglycémie, crise hypertensive)	Critique
F10	Tableau de bord statistique (moyennes, min/max, tendances)	Élevée
F11	Paramètres de l'application : gestion du compte, exportation des données, préférences	Moyenne
F12	Partage des rapports PDF par e-mail, WhatsApp ou téléchargement	Faible

3.3.2 Exigences Non Fonctionnelles

Table 3.2. Exigences non fonctionnelles du système

Critère	Exigence
Performance	Réponse < 2 s (opérations courantes) ; < 5 s (génération PDF) ; Lighthouse $\geq 90/100$
Sécurité	bcrypt (rounds=12) — JWT (exp. 7 jours) — HTTPS — protection XSS/-NoSQL Injection/CSRF/Rate Limiting
Utilisabilité	Interface intuitive, support RTL arabe, éléments adaptés aux seniors, Lighthouse Accessibility ≥ 95
Compatibilité	Android/iOS (PWA) — Chrome, Firefox, Safari — responsive design
Fiabilité	Disponibilité $\geq 99,5\%$ — sauvegarde quotidienne automatique
Scalabilité	Architecture modulaire, ajout de nouvelles pathologies sans refactoring majeur
RGPD	Données chiffrées, droit à l’effacement, consentement explicite

3.4 Diagramme de Cas d’Utilisation (Use Case Diagram)

Le diagramme de cas d’utilisation modélise les interactions entre l’unique acteur du système — le **Patient** — et l’ensemble des fonctionnalités mises à sa disposition.

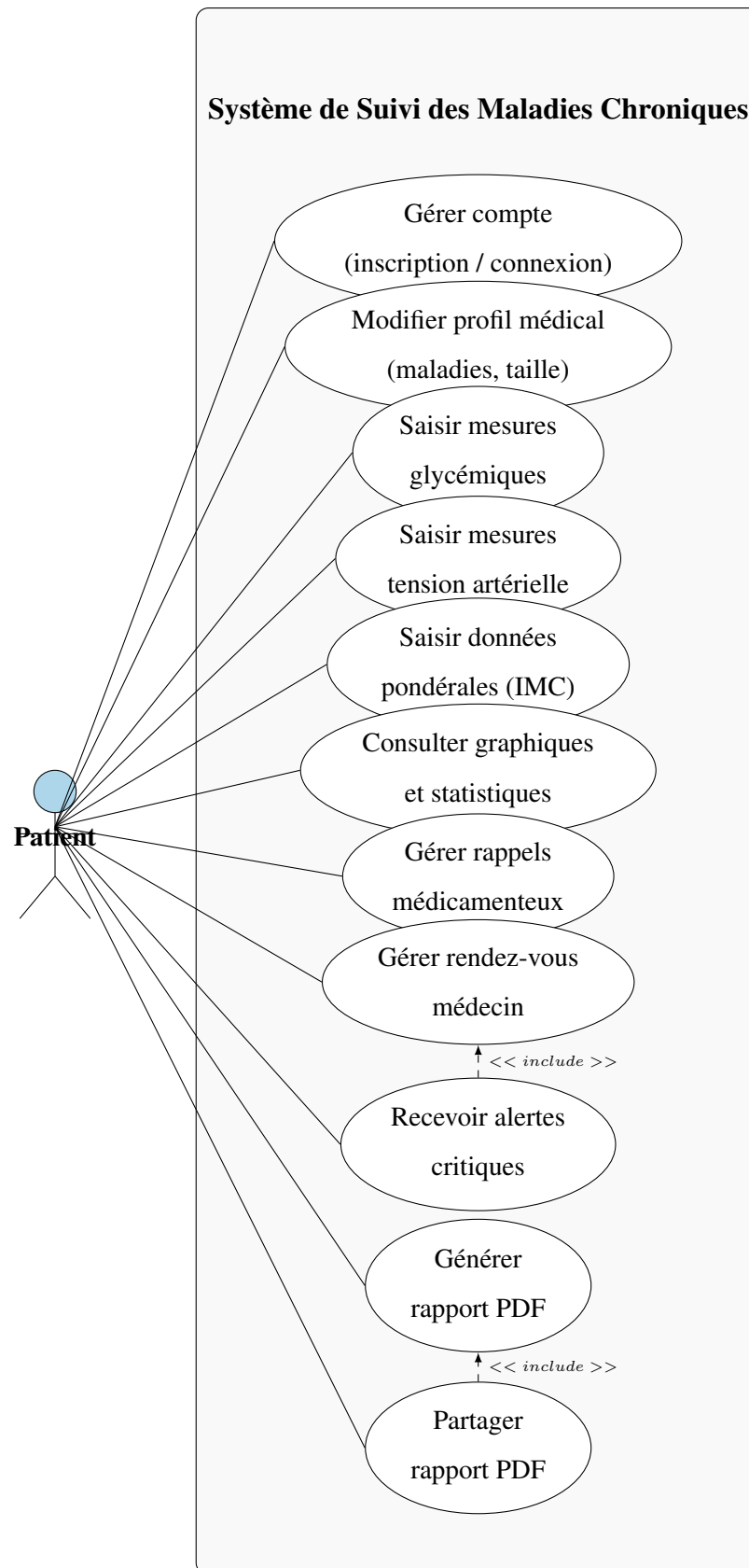


Figure 3.1. Diagramme de cas d'utilisation — Vue globale (acteur unique : Patient)

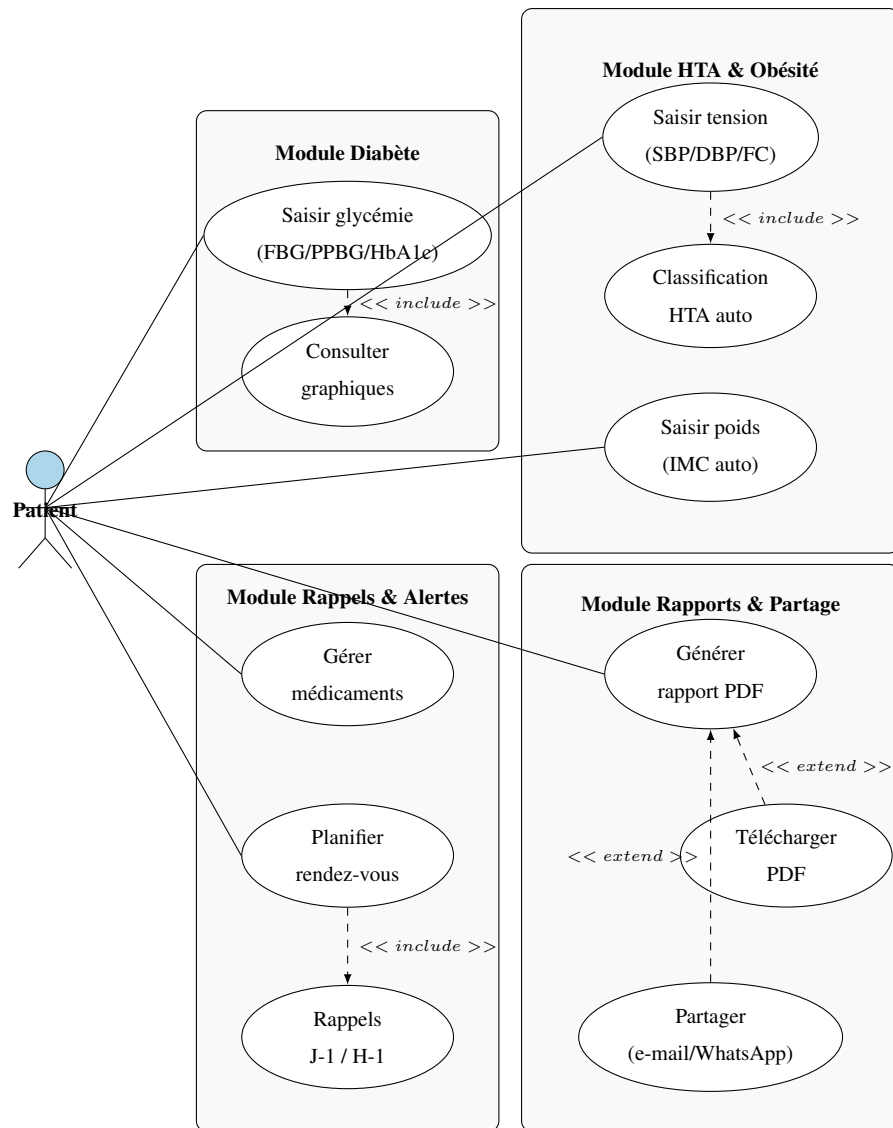


Figure 3.2. Diagramme de cas d'utilisation — Module Patient : détail des fonctionnalités par sous-système

3.5 Diagramme de Classes (Class Diagram)

Le diagramme de classes modélise la structure statique du système : classes, attributs, méthodes et relations. Il a été conçu avec **Visual Paradigm** et guide la définition des modèles MongoDB.

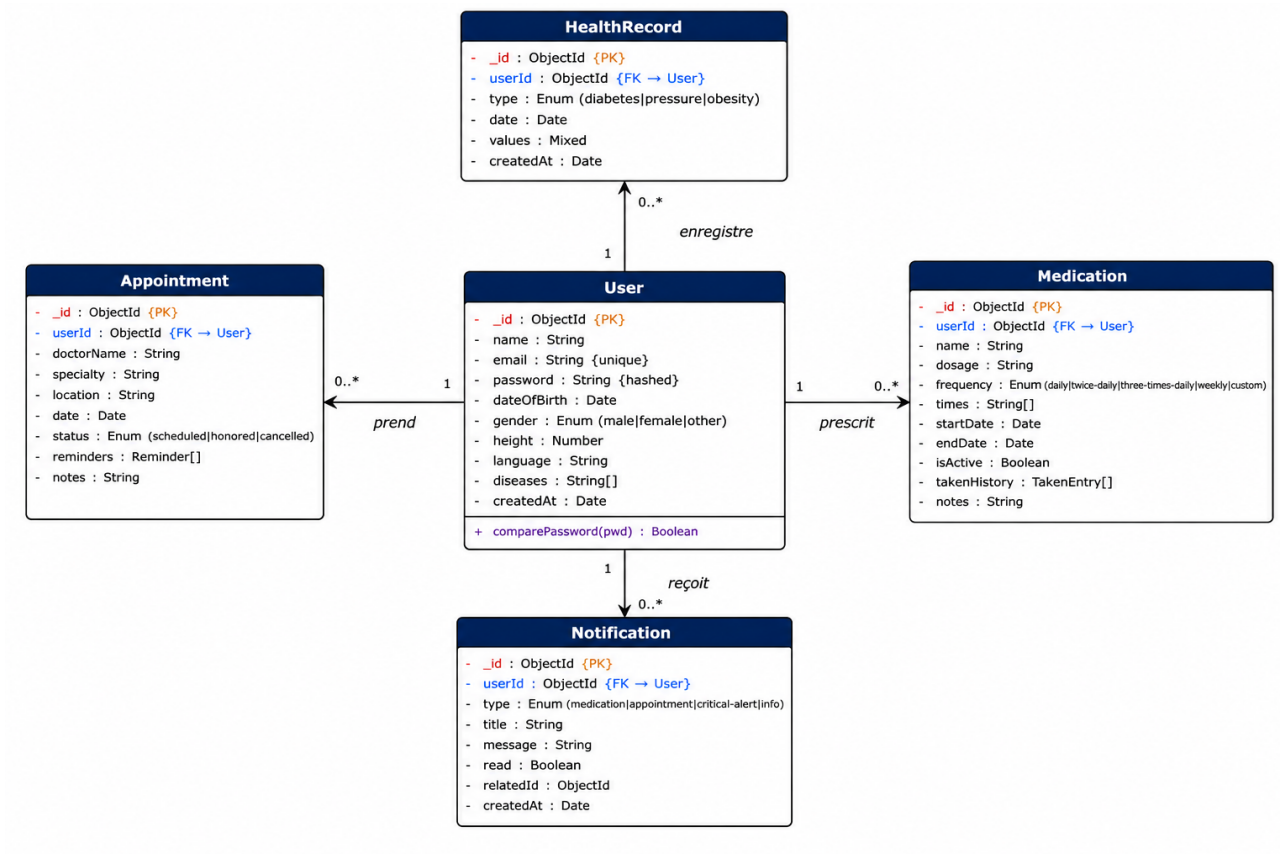


Figure 3.3. Diagramme de classes — Vue complète avec attributs et méthodes

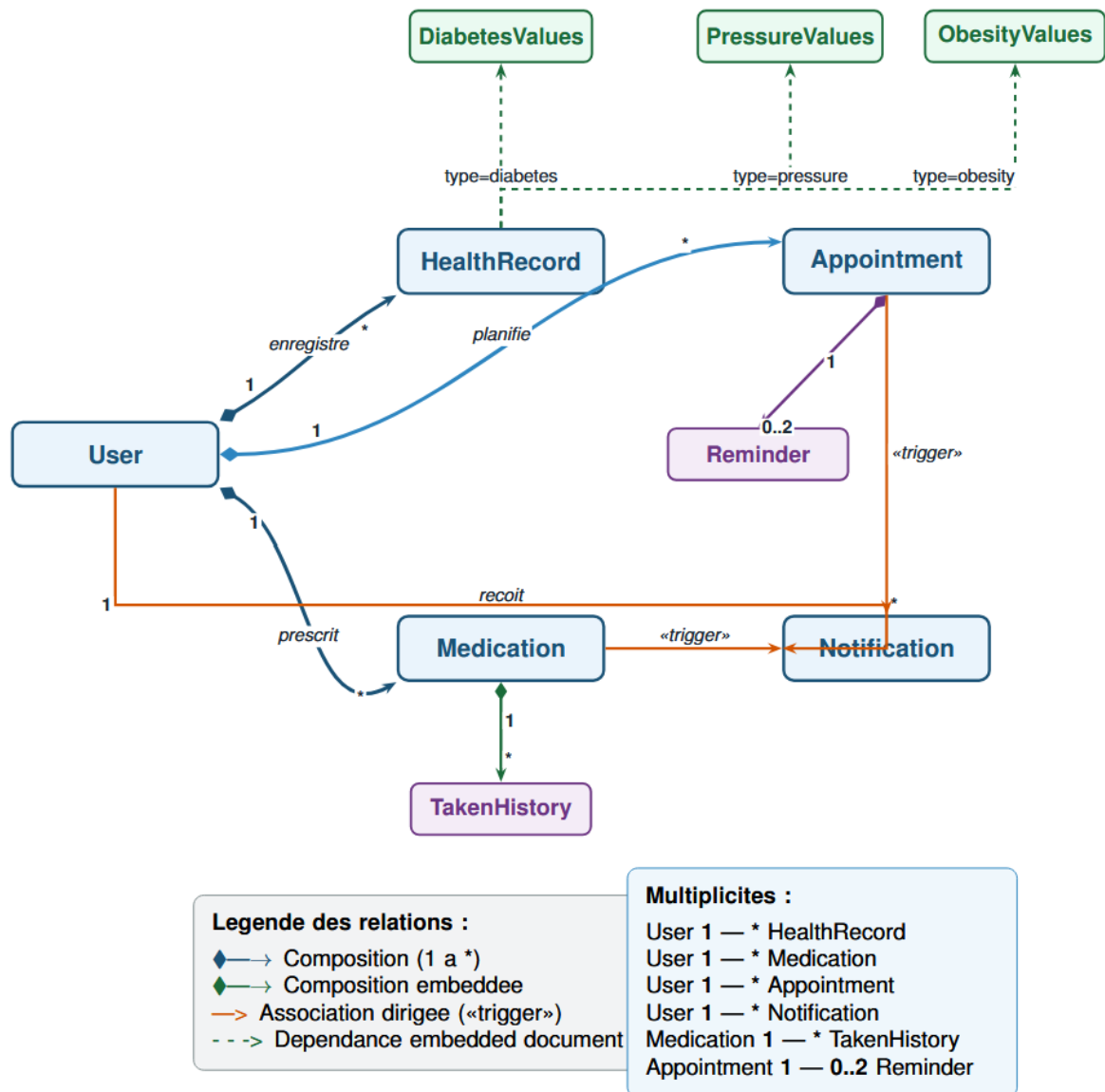


Figure 3.4. Diagramme de classes — Relations, multiplicités et dépendances entre entités

3.6 Diagrammes de Séquence

3.6.1 Scénario 1 : Saisie d'une Mesure Glycémique et Alerte

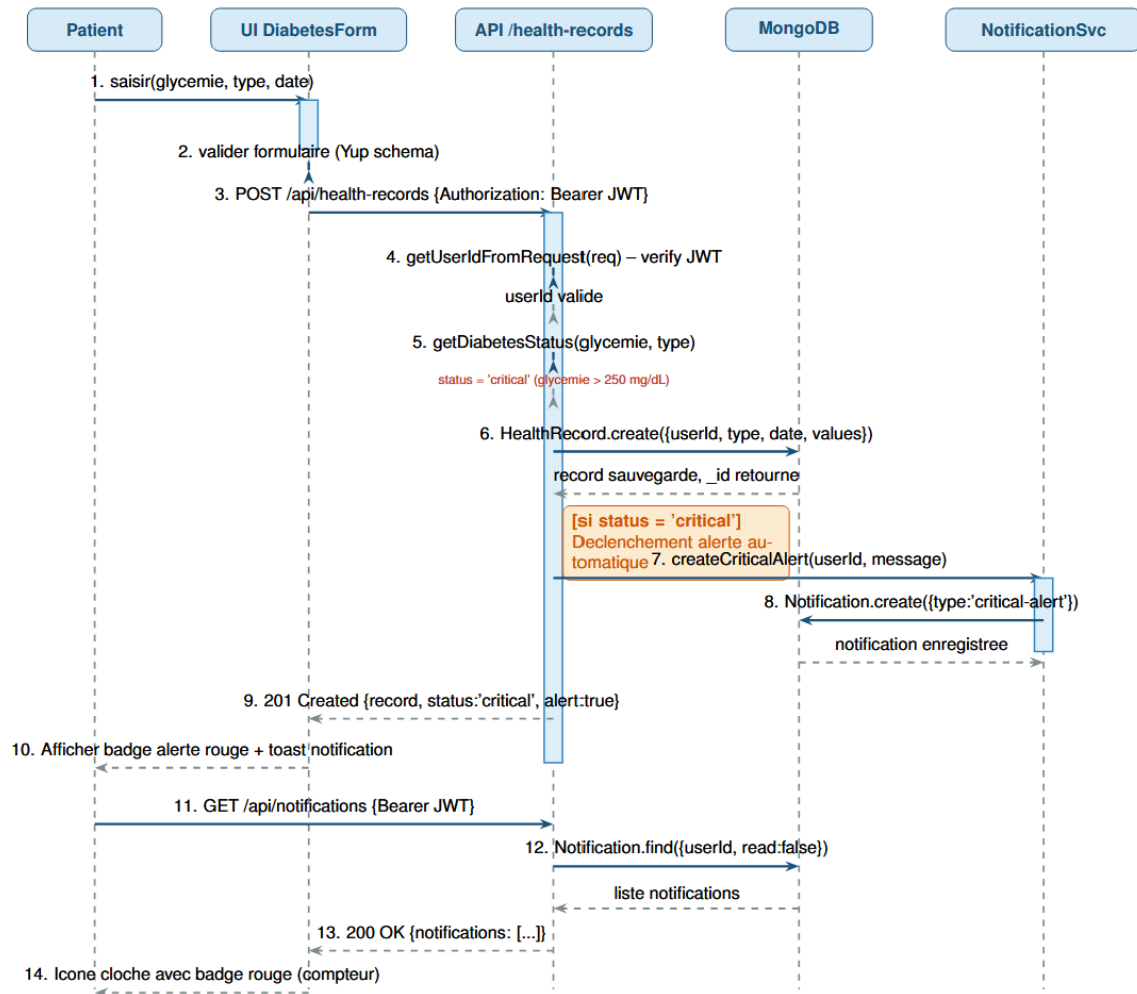


Figure 3.5. Diagramme de séquence — Saisie d'une mesure glycémique et déclenchement d'alerte critique

3.6.2 Scénario 2 : Gestion d'un Rendez-vous Médecin

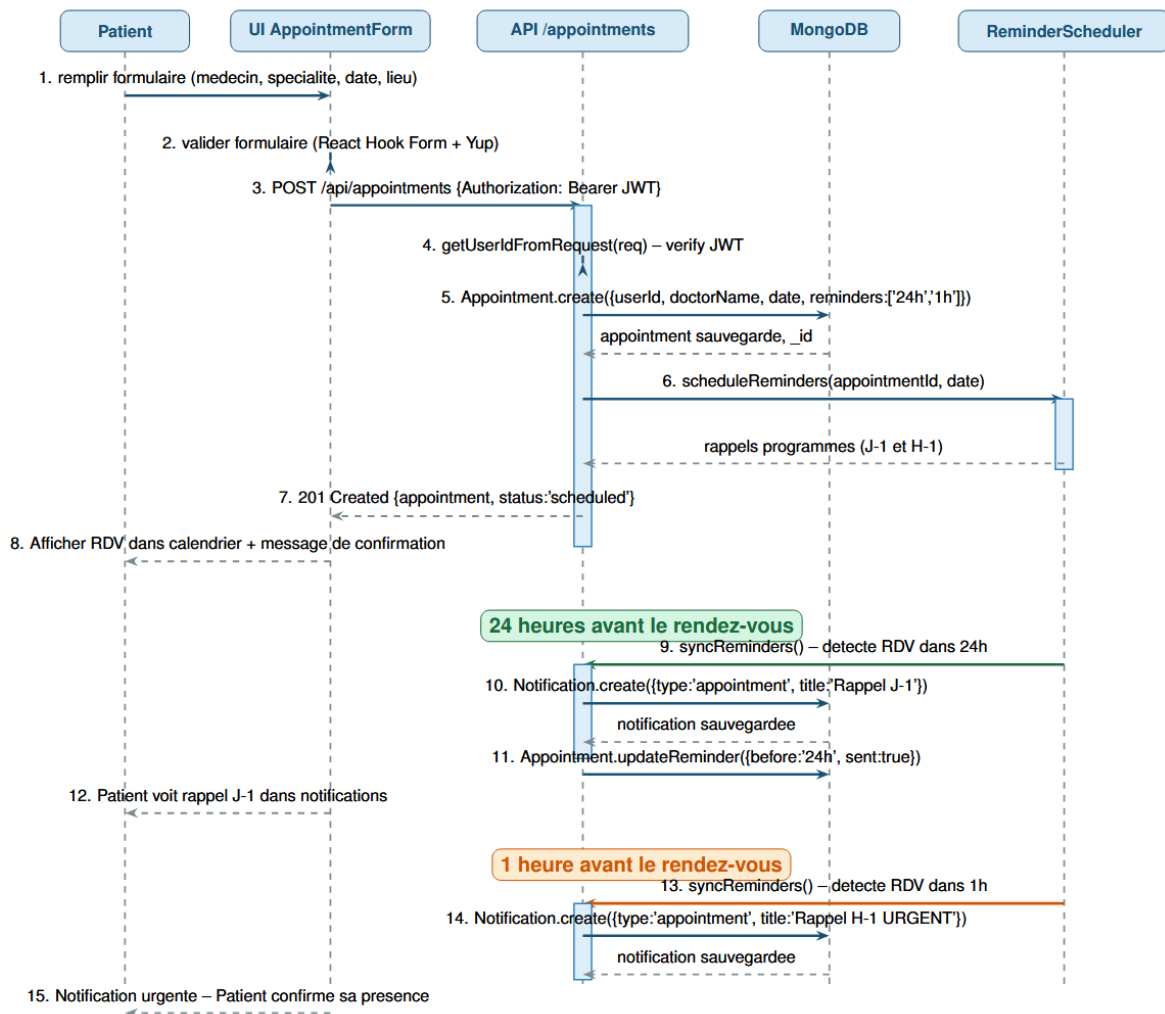


Figure 3.6. Diagramme de séquence — Création d'un rendez-vous médecin et rappels automatiques J-1 et H-1

3.6.3 Scénario 3 : Génération d'un Rapport PDF

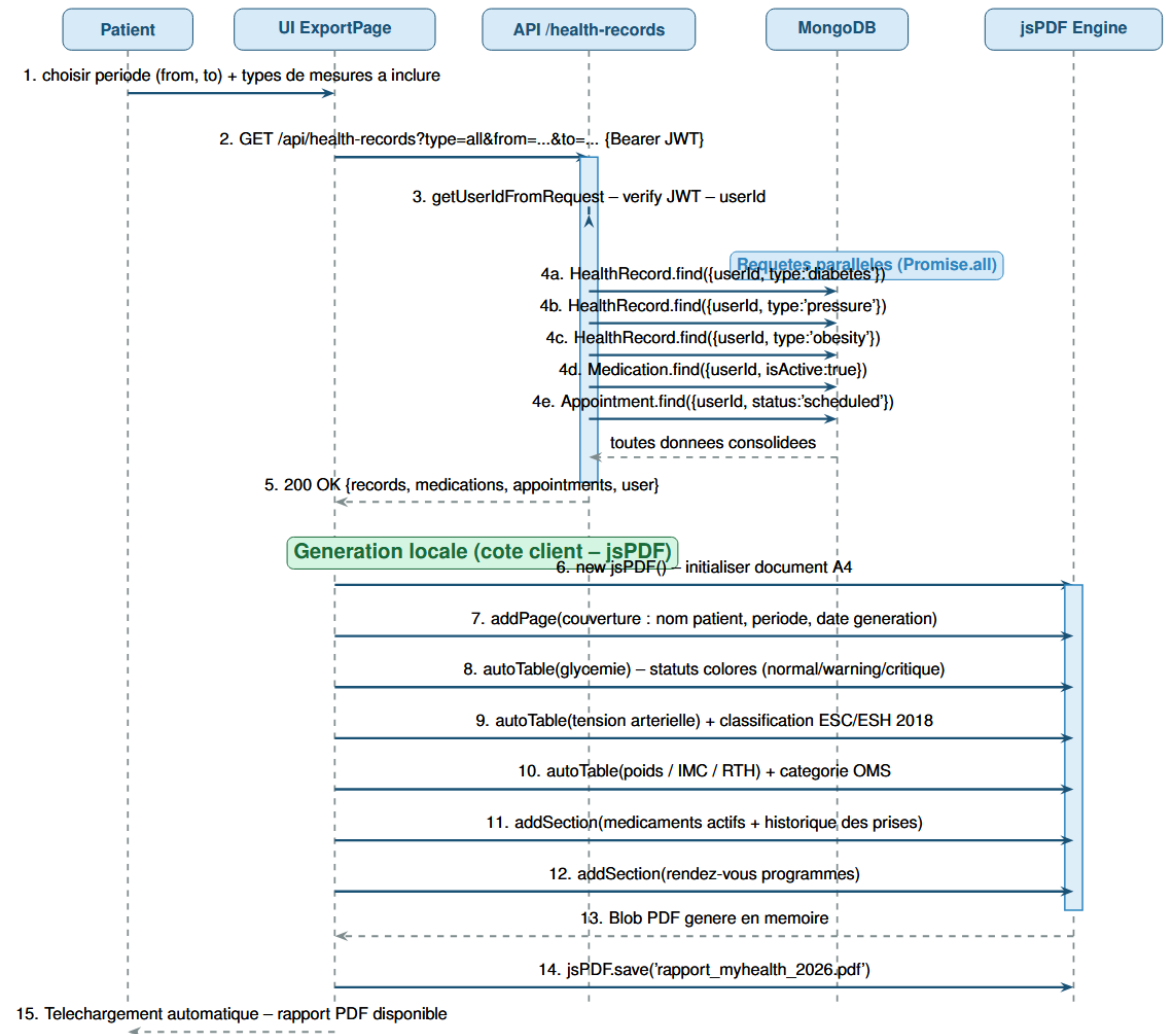


Figure 3.7. Diagramme de séquence — Génération et partage d'un rapport PDF médical

3.7 Conception de la Base de Données (MongoDB)

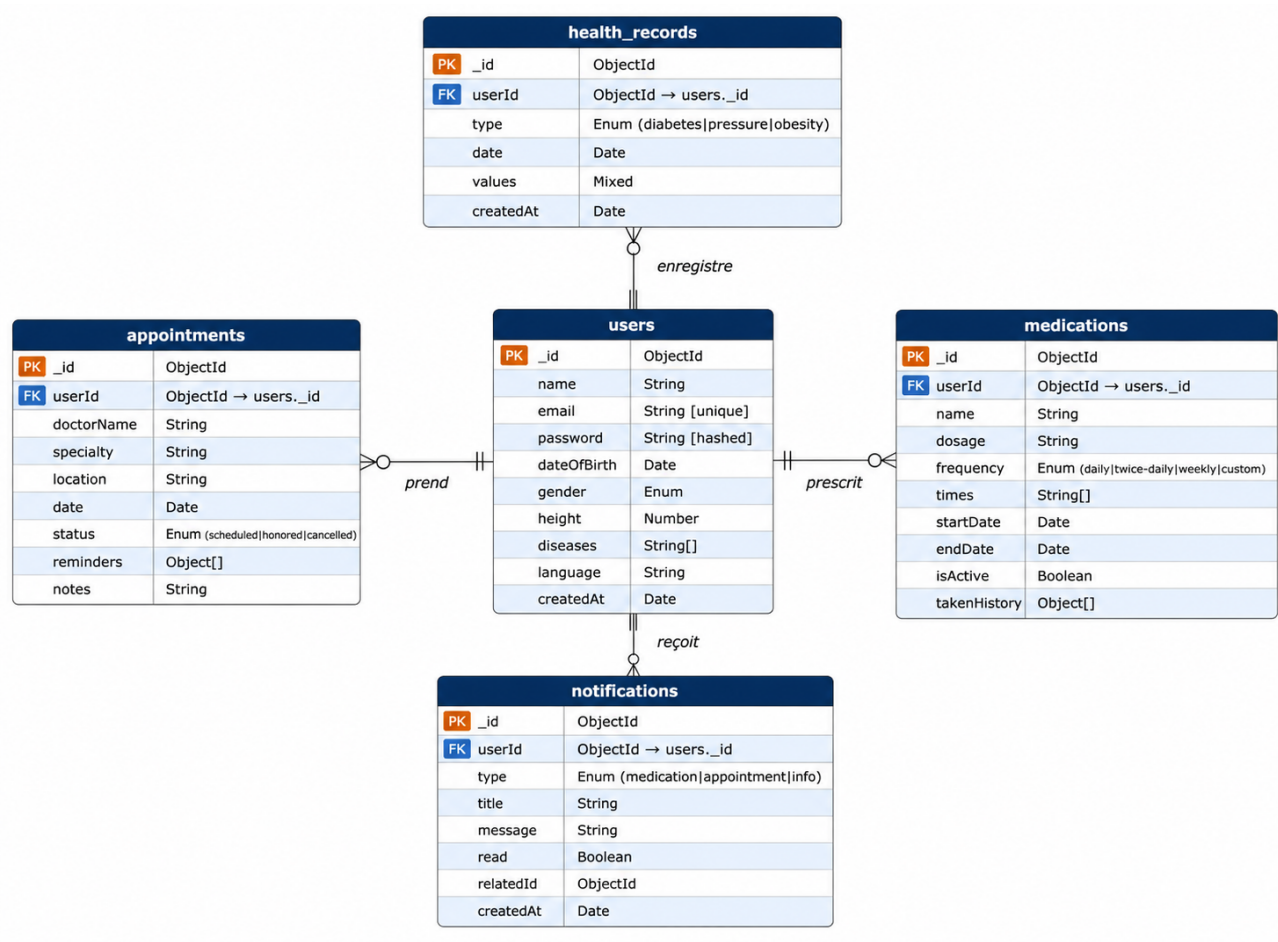


Figure 3.8. Schéma de la base de données MongoDB — Collections, champs, types et index

3.8 Conception des Interfaces (Wireframes)

Table 3.3. Description des écrans principaux de l'application

Écran	Nom	Contenu et fonctionnalités
E01	Splash Screen	Logo, nom, animation, sélection de la langue
E02	Inscription/Connexion	E-mail, mot de passe, sélection des pathologies
E03	Profil médical	Nom, date de naissance, sexe, taille, maladies
E04	Tableau de bord	Résumé mesures, rappels à venir, conseil du jour
E05	Module Diabète	Formulaire saisie, graphique, historique, statistiques
E06	Module Tension	Formulaire, classification auto couleur, graphique
E07	Module Obésité	Formulaire, IMC auto, objectif, courbe progression
E08	Rappels médicamenteux	Liste médicaments, ajout, marquer comme pris
E09	Rendez-vous médecin	Calendrier, ajout RDV, rappels J-1 et H-1 configurables
E10	Génération PDF	Sélection période, téléchargement
E11	Paramètres	Langue, Changer de thème, Se déconnecter

3.8.1 E02

My Health

Bienvenue ! Connectez-vous pour suivre votre santé.

Suivi quotidien du diabète, de la tension artérielle et du poids.

Se connecter

Nouvel utilisateur ? [Créer un compte](#)

E-mail

ayoubladjel.info02@gmail.com

Mot de passe

.....

Se connecter

Figure 3.9. Inscription/Connexion E-mail, mot de passe, sélection des pathologies

3.8.2 E04



Figure 3.10. Tableau de bord Résumé mesures, rappels à venir, conseil du jour

3.8.3 E05

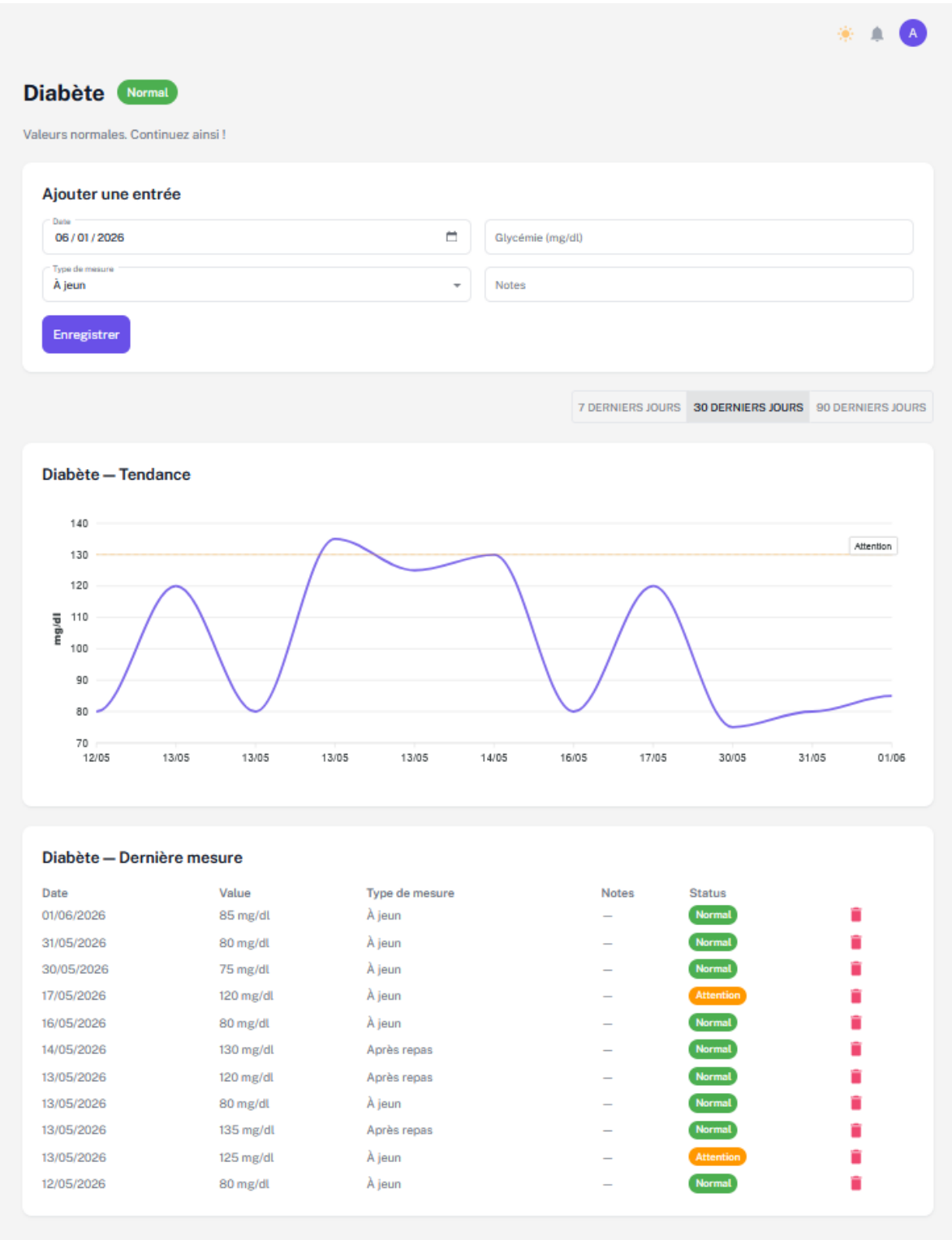


Figure 3.11. Module Diabète: Formulaire saisie, graphique, historique, statistiques

3.8.4 E10

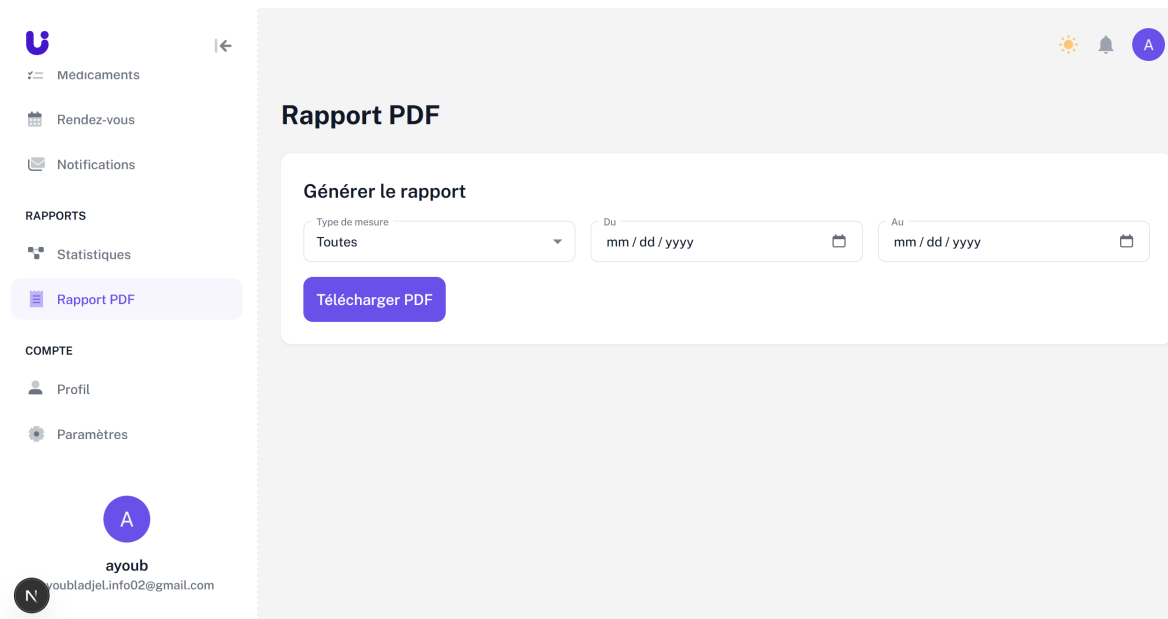


Figure 3.12. Génération PDF Sélection période, téléchargement

3.8.5 E11

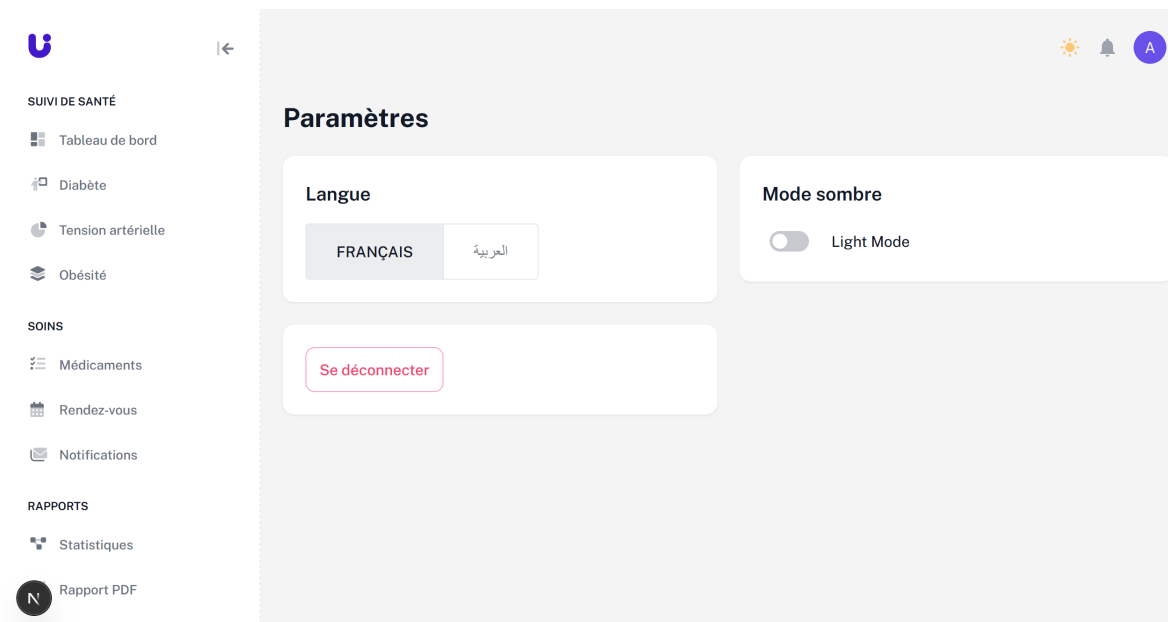


Figure 3.13. Langue, Changer de thème, Se déconnecter

3.9 Conclusion

Ce chapitre a fourni une conception complète et rigoureuse du système, depuis le cahier des charges jusqu'aux wireframes, en passant par les diagrammes UML réalisés avec **Visual Paradigm**. Le chapitre suivant documentera la réalisation technique.

CHAPITRE 4

Développement et Réalisation

4.1 Introduction

Ce chapitre documente la réalisation technique complète du projet **My Health**, en présentant l'architecture du système, la pile technologique, les patrons de conception adoptés, les modèles de données, les routes **API REST**, l'implémentation détaillée de chaque fonctionnalité, les calculs médicaux et les mesures de sécurité mises en place.

4.2 Présentation Générale du Projet

4.2.1 Objectif

My Health est une application web complète destinée au suivi des patients atteints de maladies chroniques : diabète sucré, hypertension artérielle et obésité. Elle vise à centraliser les mesures biologiques, à automatiser les rappels médicamenteux et de rendez-vous, et à produire des rapports médicaux exportables.

4.2.2 Fonctionnalités Principales

- Authentification sécurisée par JWT avec hachage bcrypt des mots de passe.
- Saisie quotidienne des mesures : glycémie (FBG, PPBG, HbA_{1c}, RBG), tension artérielle (SBP, DBP, FC, MAP), poids et IMC.
- Classification automatique selon les standards médicaux (ADA, ESC/ESH 2018, OMS).

- Visualisation graphique interactive (ApexCharts) avec lignes de seuils médicaux.
- Gestion des médicaments avec rappels programmés et historique des prises.
- Gestion des rendez-vous médicaux avec rappels automatiques J-1 et H-1.
- Système de notifications en temps réel (mock scheduler intégré).
- Génération de rapports PDF complets côté client.
- Interface multilingue : français (par défaut) et arabe avec support RTL intégral.
- Profil médical paramétrable (taille, sexe, pathologies, date de naissance).

4.2.3 Public Cible

Patients chroniques, en particulier les personnes âgées atteintes de diabète de type 2 et d'hypertension. L'interface a été conçue avec des polices larges (16px+ pour le corps, 20px+ pour les titres) et des boutons de grande taille (48px minimum) pour faciliter l'utilisation par tous les profils d'utilisateurs.

4.3 Architecture du Système

4.3.1 Vue d'Ensemble — Architecture en Couches

Table 4.1. Architecture en couches du système

Couche	Technologie	Rôle
Présentation	Next.js 15 + React 19 + MUI 6	Interface utilisateur, rendu hybride SSR/SSG/CSR
API REST	Next.js API Routes (Node.js)	Endpoints HTTP authentifiés JWT
Logique métier	Modules TypeScript (utils, lib)	Calculs médicaux, classification, mock scheduler
Persistance	MongoDB 7 + Mongoose 9 (ODM)	Stockage NoSQL des données médicales
Internationalisation	i18next + react-i18next + stylis-plugin-rtl	Traductions FR/AR + écriture droite-à-gauche

4.3.2 Flux de Données Typique

1. L'utilisateur saisit une mesure dans un formulaire React (validation Yup).
2. Le formulaire appelle `axios.post('/api/health-records', ...)`.
3. La route Next.js authentifie via JWT (`getUserIdFromRequest`).
4. Mongoose persiste l'enregistrement avec calculs automatiques (MAP, IMC, WHR).
5. Le composant rafraîchit la liste ; ApexCharts redessine le graphique.
6. Si la valeur est critique, une notification est créée pour l'utilisateur.

4.4 Pile Technologique (Stack)

4.4.1 Technologies Front-end

Table 4.2. Technologies front-end utilisées dans le projet

Technologie	Version	Rôle / Justification
Next.js	15.2	Framework React avec App Router. Permet SSR, SSG et CSR. Inclut un système d'API Routes intégré (pas besoin d'Express séparé).
React	19	Bibliothèque UI déclarative. Utilise hooks (useState, useEffect, useCallback, useMemo) et le nouveau hook use() pour les contextes.
TypeScript	5	Typage statique pour réduire les bugs. Tous les modèles, props et helpers sont typés.

MUI (Material UI)	6.4	Bibliothèque de composants Material Design : Card, Grid, Table, DatePicker, ToggleButton, Chip, Snackbar, etc.
Emotion	11	Moteur CSS-in-JS utilisé par MUI pour le style.
ApexCharts	4	Bibliothèque de graphiques interactifs : courbes (line), sparklines et annotations de seuils médicaux.
react-hook-form	7	Gestion performante des formulaires médicaux (peu de re-rendus).
Yup	1	Schémas de validation déclaratifs intégrés à react-hook-form via @hookform/resolvers/yup.
i18next + react-i18next	24/15	Internationalisation. Traductions FR (par défaut) et AR.
stylis-plugin-rtl	2	Conversion automatique des styles LTR → RTL pour l'interface arabe.
date-fns	4	Manipulation et formatage des dates (léger, modulaire).
axios	1.8	Client HTTP avec intercepteurs (ajout automatique du Bearer token).
jsPDF + jspdf-autotable	4/5	Génération PDF côté client. Inclut un module pour les tableaux.

4.4.2 Technologies Back-end

Table 4.3. Technologies back-end utilisées dans le projet

Technologie	Version	Rôle / Justification
Node.js	20+	Runtime JavaScript serveur (utilisé par Next.js).
Next.js API Routes	15.2	Endpoints REST natifs définis dans <code>src/app/api/**/route.ts</code> . Intégrés au même projet Next.js, sans serveur séparé.
MongoDB	7	Base NoSQL orientée documents. Avantage pour les données médicales hétérogènes (les valeurs varient selon le type de mesure).
Mongoose	9.4	ODM (Object Document Mapper). Définit les schémas typés, gère la validation et fournit les hooks de cycle de vie (pre-save).
bcryptjs	3	Hachage cryptographique des mots de passe (12 rounds).
jsonwebtoken	9	Génération et vérification de JWT (durée 7 jours).

4.4.3 Outils de Développement

Table 4.4. Outils de développement et de déploiement

Outil	Rôle dans le projet
Visual Paradigm	Modélisation UML (Use Case, Classes, Séquence) — Conception de l'architecture
Visual Studio Code	IDE principal avec extensions ESLint, Prettier, GitLens, MongoDB for VS Code
Git + GitHub	Contrôle de version, branches (main/develop/feature), Pull Requests
Vercel	Déploiement et hébergement de l'application
MongoDB Compass	Interface graphique pour la gestion de la base de données
Figma	Wireframes haute fidélité et maquettes graphiques des interfaces
ESLint + Prettier	Qualité et formatage du code

4.5 Justification des choix techniques

Pourquoi MongoDB plutôt qu'une base relationnelle ?

Les données médicales sont hétérogènes : le contenu de `values` change selon le type de mesure. MongoDB stocke ces structures variables dans un seul schéma flexible (`Schema.Types.Mixed`) sans jointures complexes. Les requêtes typiques (par utilisateur+type+période) sont indexables et performantes en NoSQL.

Pourquoi Next.js au lieu d'Express + React séparés ?

Next.js fournit dans un seul projet : le rendu côté serveur, le routing, et un système d'API Routes. Cela simplifie le déploiement (un seul serveur Node.js), améliore le SEO (SSR), et permet de partager les types TypeScript entre client et serveur, réduisant la surface d'erreur.

Pourquoi calculer la PAM côté serveur ?

La PAM est un champ dérivé permanent. La calculer côté serveur garantit : (1) cohérence à l'insertion, (2) indépendance du client (qui pourrait manipuler la valeur), (3) disponibilité immédiate pour les requêtes ultérieures (PDF, statistiques) sans recalcul.

Comment fonctionne le RTL arabe ?

Trois mécanismes combinés : (1) `il8next.changeLanguage('ar')` change les textes ; (2) `settings.direction = 'rtl'` est passé à l'attribut `dir` de l'élément racine ; (3) `stylis-plugin-rtl` inverse automatiquement les marges et positions CSS de MUI.

Pourquoi générer le PDF côté client ?

Avantages : aucune charge serveur, téléchargement immédiat, données sensibles ne transitent pas une seconde fois sur le réseau. Inconvénient documenté : la taille du bundle augmente d'environ 150 Ko (jsPDF), acceptable car chargé à la demande sur la page Export uniquement.

4.6 Techniques et Patrons de Conception

4.6.1 Architecture en Couches (Layered Architecture)

Le projet sépare clairement : présentation (composants), logique métier (utils + lib), accès aux données (modèles Mongoose), et API (route handlers). Chaque couche ne dépend que de la couche inférieure.

4.6.2 Pattern Container / Presentational

Les pages Next.js (`src/app/dashboard/*/page.tsx`) sont de simples wrappers déléguant l'affichage à des composants `PageView` (`src/page-sections/health/*/...PageView.tsx`). Cela facilite les tests et la réutilisation.

4.6.3 Authentification Stateless par JWT

- Le serveur émet un token signé HMAC-SHA256 contenant `{userId}` avec une durée de vie de 7 jours.
- Le client stocke le token dans `localStorage` (clé `accessToken`).
- Axios ajoute automatiquement `Authorization: Bearer <token>` à chaque requête.
- Chaque endpoint protégé extrait l'ID via `getUserIdFromRequest`.
- Logout = suppression du token côté client (stateless, pas de session serveur).

4.6.4 Hachage bcrypt avec Hook pre-save

Le modèle `User` exécute un `pre('save')` qui hache le mot de passe avec 12 rounds avant l'écriture en base. La méthode `comparePassword` permet la vérification au login. Le mot de passe n'est jamais renvoyé grâce à `select: false` dans le schéma.

4.6.5 Schéma Flexible (Mixed) pour les Enregistrements de Santé

Le modèle `HealthRecord` utilise `Schema.Types.Mixed` pour le champ `values`, ce qui permet de stocker des structures différentes selon `type` (diabète vs tension vs obésité) tout en partageant la même collection. Une discriminated union TypeScript assure la sécurité au compile-time.

Justification du choix Mixed vs Discriminators

L'alternative `Mongoose Discriminators` aurait créé trois collections séparées (`DiabetesRecord`, `PressureRecord`, `ObesityRecord`), complexifiant les requêtes multi-pathologies (tableau de bord, PDF). Avec `Mixed` + index composé `{userId, type, date}`, une seule requête suffit pour tous les types. La validation de structure est déléguée à TypeScript (compile-time) et à Yup (runtime), compensant l'absence de validation `Mongoose` au niveau du champ `values`.

4.6.6 Calculs Automatiques Côté Serveur

Lors d'un POST sur `/api/health-records`, le serveur calcule automatiquement :

- **MAP** (Pression Artérielle Moyenne) pour la tension :
$$\text{MAP} = \text{DBP} + \frac{\text{SBP} - \text{DBP}}{3}$$

- **IMC** (Indice de Masse Corporelle) pour l'obésité, à partir de la taille stockée dans le profil :
$$\text{IMC} = \frac{\text{poids (kg)}}{\text{taille}^2(\text{m})}$$
- **WHR** (rapport taille/hanche) si les deux mesures sont fournies.

La valeur officielle est toujours celle calculée côté serveur, garantissant cohérence et intégrité des données.

4.6.7 Mise en Cache de la Connexion MongoDB

Pour éviter d'ouvrir une nouvelle connexion Mongoose à chaque hot-reload en mode développement, la connexion est mise en cache dans `global._mongooseCache`.

4.6.8 Mock Scheduler (Rappels Simulés)

Au lieu d'utiliser Firebase FCM ou un cron serveur externe, l'application embarque un mock scheduler sous forme d'endpoint `POST /api/notifications/sync`. Le client (cloche dans le header) appelle cet endpoint toutes les 60 secondes. Le serveur scanne :

- Les rendez-vous dans les 24 prochaines heures (rappel J-1) ou 1 heure (H-1).
- Les médicaments actifs dont l'horaire correspond à ± 5 minutes de l'heure actuelle.

Pour chaque correspondance, une notification est créée en base et le badge de la cloche affiche le compteur de non-lues. En production, ce mock serait remplacé par un vrai `node-cron` sans changer l'API publique.

4.6.9 Validation à Plusieurs Niveaux

1. **Frontend** : Yup vérifie les types et plages au moment de la saisie (validation immédiate).
2. **API** : vérifications contextuelles (utilisateur authentifié, propriété de la ressource, whitelisting de champs).
3. **Mongoose** : validation de schéma (required, enum, type) avant toute insertion en base.

4.6.10 Internationalisation et RTL

- `i18next` charge les traductions au démarrage (`src/i18n/index.ts`).

- Les composants utilisent `const { t } = useTranslation()`.
- Le passage en arabe modifie `settings.direction = 'rtl'` ; le composant RTL insère un plugin Stylin qui inverse automatiquement `marginLeft ↔ marginRight`, `left ↔ right`, etc.

4.7 Modèles de Données (MongoDB / Mongoose)

4.7.1 Modèle User — `src/models/User.ts`

Table 4.5. Modèle User — champs et contraintes

Champ	Type	Notes
name	String	Requis
email	String	Requis, unique, lowercase
password	String	Requis, select: false (caché par défaut)
dateOfBirth	Date	Optionnel
gender	String	Optionnel — enum [male, female, other]
height	Number	Optionnel — en cm, utilisé pour l'IMC
diseases	[String]	Pathologies du patient
language	String	Défaut 'fr'

4.7.2 Modèle HealthRecord — `src/models/HealthRecord.ts`

Champ `values` de type `Mixed` — structure variable selon le type de mesure :

Structures du champ `values` selon le type de mesure

```

1 // type === 'diabetes'
2 { glycemia: number, measurementType: 'FBG' | 'PPBG' | 'HbA1c' |
   'RBG',
3   notes?: string }
```

```
4
5 // type === 'pressure'
6 { systolic: number, diastolic: number, pulse: number,
7   map?: number, notes?: string }
8
9 // type === 'obesity'
10 { weight: number, bmi?: number, waistCirc?: number,
11   hipCirc?: number, whr?: number }
```

Index composé : { userId: 1, type: 1, date: -1 } pour des requêtes rapides par utilisateur+type+période.

4.7.3 Modèle Medication — `src/models/Medication.ts`

- name, dosage — texte (ex : «Metformine», «500mg»)
- frequency — enum (daily, twice-daily, three-times-daily, weekly, custom)
- times[] — heures programmées au format HH:mm
- startDate, endDate
- isActive — booléen
- takenHistory[] — sous-document {date, time, taken: Boolean}

4.7.4 Modèle Appointment — `src/models/Appointment.ts`

- doctorName, specialty, location
- date — Date (incluant l'heure)
- reminders[] — [{ before: '24h' | '1h', sent: boolean }] (les deux créés par défaut)
- status — enum [scheduled, honored, cancelled]

4.7.5 Modèle Notification — `src/models/Notification.ts`

- type — enum [medication, appointment, critical-alert, info]

- title,message
- read — booléen
- relatedId — référence vers Medication ou Appointment

4.8 Routes API REST

Table 4.6. Routes API REST du système

Méthode	URL	Description	Fichier
POST	/api/auth/register	Inscription utilisateur	auth/register/route.ts
POST	/api/auth/login	Connexion (renvoie JWT)	auth/login/route.ts
GET	/api/auth/me	Profil de l'utilisateur courant	auth/me/route.ts
PUT	/api/auth/profile	Mise à jour du profil médical	auth/profile/route.ts
GET/POST	/api/health-records	Liste/crée des mesures (filtres: type, from, to)	health-records/route.ts
DELETE	/api/health-records/[id]	Suppression d'une mesure	health-records/[id]/route.ts
GET/POST	/api/medications	Liste/crée des médicaments (filtre: active)	medications/route.ts
PUT/DELETE	/api/medications/[id]	Modification/suppression	medications/[id]/route.ts
POST	/api/medications/[id]/take	Marquer une dose comme prise	medications/[id]/take/route.ts
GET/POST	/api/appointments	Liste/crée des rendez-vous	appointments/route.ts
PUT/DELETE	/api/appointments/[id]	Modification/suppression	appointments/[id]/route.ts
GET/POST	/api/notifications	Liste les notifications	notifications/route.ts
POST	/api/notifications/[id]/read	Marquer comme lue	notifications/[id]/read/route.ts
POST	/api/notifications/sync	Mock scheduler — déclenche les rappels dûs	notifications/sync/route.ts

4.9 Fonctionnalités — Localisation dans le Code

4.9.1 Authentification JWT + bcrypt

L'utilisateur s'inscrit avec email + mot de passe + nom. Le mot de passe est haché avec bcrypt (12 rounds). À la connexion, un JWT signé (durée 7 jours) est retourné et stocké dans localStorage.

Fichiers principaux :

- `src/lib/auth.ts` — fonctions `signToken`, `verifyToken`, `getUserIdFromRequest`
- `src/contexts/jwtContext.tsx` — context React (login, register, logout, check-CurrentUser)
- `src/hooks/useAuth.ts` — hook d'accès au context
- `src/components/auth/AuthGuard.tsx` — protège les routes /dashboard
- `src/app/api/auth/login/route.ts` et `register/route.ts` — endpoints backend

4.9.2 Module Diabète

Suivi de quatre types de mesures : FBG (à jeun), PPBG (post-prandial), HbA_{1c} (hémoglobine glyquée, en %), RBG (aléatoire). Classification automatique selon les seuils ADA via `getDiabetesStatus(value, type)` dans `src/utils/health-thresholds.ts`.

Fichiers : `DiabetesForm.tsx`, `DiabetesChart.tsx` (graphique avec annotations 130/180 mg/dL), `DiabetesTable.tsx`, `DiabetesPageView.tsx`.

4.9.3 Module Tension Artérielle

Saisie de SBP (systolique), DBP (diastolique) et FC (fréquence cardiaque). La PAM est calculée automatiquement par le serveur. Classification selon ESC/ESH 2018 (6 catégories).

Fichiers : `PressureForm.tsx` (aperçu MAP en direct), `PressureChart.tsx` (graphique 3 séries SBP/DBP/FC), `PressureTable.tsx` (colonne MAP + Catégorie).

4.9.4 Module Obésité

Saisie du poids (kg), tour de taille (cm), tour de hanches (cm). L'IMC et le WHR sont calculés automatiquement par le serveur (si la taille est définie dans le profil). Si la taille est absente, le champ `bmi` reste `undefined` et l'interface affiche «—» avec une invite à compléter le profil.

Formules : $IMC = \text{poids} / \text{taille}^2$; $\text{Poids cible} = 24,9 \times \text{taille}^2$; $WHR = \text{tour de taille} / \text{tour de hanches}$.

Fichiers : `ObesityForm.tsx`, `ObesityChart.tsx` (courbe + ligne de poids cible), `ObesityTable.tsx`.

4.9.5 Module Médicaments

Gestion CRUD complète : nom, dosage, fréquence (quotidien, deux fois/jour, trois fois/jour, hebdomadaire, personnalisé), horaires de prise au format HH:mm (validés par regex), date de début/fin, état actif/inactif. Bouton «Marquer comme pris» qui ajoute une entrée dans `takenHistory`.

Fichiers : `MedicationForm.tsx`, `MedicationList.tsx`, `src/models/Medication.ts`.

4.9.6 Module Rendez-vous

CRUD des rendez-vous médicaux. Champs : nom du médecin (requis), spécialité, lieu, date+heure (`datetime-local`), notes. Onglets «À venir» / «Passés». À la création, deux rappels sont automatiquement initialisés : 24h avant (J-1) et 1h avant (H-1), avec `sent` : `false`.

Fichiers : `AppointmentForm.tsx`, `AppointmentList.tsx`, `src/models/Appointment.ts`.

4.9.7 Système de Notifications (Mock Scheduler)

L'endpoint `POST /api/notifications/sync` est appelé toutes les 60 secondes par la cloche du header (`NotificationBell.tsx`). Cet endpoint :

1. Récupère tous les rendez-vous «scheduled» dans les 25 prochaines heures.
2. Vérifie si $\text{now} \leq 24\text{h}$ (rappel J-1) ou $\text{now} \leq 1\text{h}$ (rappel H-1) et si `sent = false`. Si oui, crée une notification et marque `reminder.sent = true`.

3. Récupère tous les médicaments actifs et pour chaque horaire dans `times[]`, crée une notification si l'heure est à ± 5 min et qu'aucune notification n'existe déjà aujourd'hui.

4.9.8 Tableau de Bord

Vue d'ensemble avec 3 cartes `HealthSummaryCard` (Diabète, Tension, Obésité) affichant la dernière mesure + statut + sparkline 7 jours, complétées par 3 widgets : Médicaments prochains, Rendez-vous prochains, Alertes récentes.

4.9.9 Statistiques

Vue agrégée sur une période sélectionnable (7/30/90 jours). Chip de tendance calculé en comparant la moyenne de la première moitié des données à la seconde moitié.

Algorithme de calcul de tendance (`StatisticsPageView.tsx`)

```
1 function getTrend(values) {  
2   if (values.length < 2) return 'Stable'  
3   const recent = values.slice(0, Math.ceil(values.length /  
4     2))  
5   const older = values.slice(Math.ceil(values.length / 2))  
6   const diff = avg(recent) - avg(older)  
7   if (Math.abs(diff) < 5) return 'Stable'  
8   return diff < 0 ? 'Improving' : 'Worsening'  
9 }
```

4.9.10 Export PDF

Génération côté client à l'aide de `jsPDF` et `jspdf-autotable`. Contenu : en-tête patient, tableaux des mesures, statistiques (moyenne, min, max), liste des médicaments actifs, rendez-vous à venir, recommandations. Aucun appel serveur supplémentaire — le PDF est créé directement dans le navigateur pour un téléchargement immédiat.

4.10 Calculs Médicaux et Seuils

Tous les calculs et seuils sont centralisés dans `src/utlis/health-thresholds.ts` : source de vérité unique utilisée par les formulaires (aperçu live), les tableaux (colonne Statut), le générateur PDF et l’**API REST**.

4.10.1 Seuils Diabète — ADA

Table 4.7. Seuils diagnostiques du diabète (ADA)[4]

Paramètre	Normal	Pré-diabète	Diabète	Critique
FBG (à jeun)	70–99 mg/dL	100–125	≥ 126	<70 ou >250
PPBG (post-repas)	<140	140–199	≥ 200	<70 ou >300
HbA1c	<5,7 %	5,7–6,4 %	$\geq 6,5$ %	≥ 9 %
RBG (aléatoire)	<200	—	≥ 200	>300

4.10.2 Seuils Tension — ESC/ESH 2018[5]

Table 4.8. Classification de la tension artérielle (ESC/ESH 2018)

Catégorie	Systolique	Diastolique
Optimale	<120	<80
Normale	120–129	80–84
Normale haute	130–139	85–89
HTA grade 1	140–159	90–99
HTA grade 2	160–179	100–109
HTA grade 3 (crise)	≥ 180	≥ 110

4.10.3 Seuils Obésité — OMS

Table 4.9. Classification de l'obésité selon l'OMS (IMC)[10]

Catégorie	IMC (kg/m ²)
Insuffisance pondérale	<18,5
Corpulence normale	18,5–24,9
Surpoids	25–29,9
Obésité grade I	30–34,9
Obésité grade II	35–39,9
Obésité morbide	≥40

4.11 Sécurité

4.11.1 Mesures Implémentées

- **Hachage des mots de passe** : bcrypt, 12 rounds (équilibre sécurité/performance).
- **Authentification stateless** : JWT signé HMAC-SHA256.
- **Autorisation au niveau ressource** : chaque requête API filtre par `userId` extrait du token. Aucun endpoint ne permet de spécifier un `userId` arbitraire.
- **Whitelist de champs** : les endpoints PUT filtrent les champs autorisés, empêchant la modification de `email` ou `password` via des routes non prévues.
- **Validation à 3 niveaux** : Yup (frontend), validation contextuelle (API), schéma Mongoose (DB).
- **Variables d'environnement** : `JWT_SECRET` et `MONGODB_URI` dans `.env.local` (non commité dans Git).

4.11.2 Améliorations Futures Recommandées

- Rate limiting (ex : 5 tentatives de login / 15 min).

- Refresh tokens pour invalider la session sans attendre l'expiration du JWT.
- HTTPS obligatoire en production (HSTS).
- Vérification d'email à l'inscription.

4.12 Conclusion

Ce chapitre a documenté l'ensemble du processus de réalisation technique de **My Health** : de l'architecture en couches et la pile technologique jusqu'à l'implémentation détaillée de chaque fonctionnalité, en passant par les modèles de données, les routes **API REST**, les calculs médicaux centralisés et les mesures de sécurité. Le chapitre suivant évaluera la qualité du système développé à travers une batterie de tests.

CHAPITRE 5

Tests et Évaluation

5.1 Introduction

Ce chapitre présente la démarche de test adoptée pour valider la qualité du système : tests automatisés, test d'acceptation utilisateur (UAT), et évaluation des performances et de la sécurité.

5.2 Stratégie de Test

- **Tests unitaires** (Jest) : validation isolée des fonctions et modules.
- **Tests d'intégration** (Supertest) : validation des flux API REST avec MongoDB.
- **Test d'acceptation utilisateur** (UAT) : 25 participants représentatifs.
- **Tests de performance** : Google Lighthouse et k6 (100 utilisateurs simultanés).
- **Tests de sécurité** : OWASP Top 10.

5.3 Tests Unitaires et d'Intégration

Table 5.1. Récapitulatif des résultats des tests automatisés

Module testé	Unit.	Intégr.	Total	Réussis	Taux
Authentification	12	12	24	24	100 %
API Diabète	18	13	31	31	100 %
API Tension	15	13	28	28	100 %
API Obésité	12	10	22	22	100 %
Rappels médicamenteux	10	8	18	18	100 %
Rendez-vous médecin	9	7	16	16	100 %
Génération PDF	7	5	12	12	100 %
Calculs médicaux	20	0	20	20	100 %
Total	103	68	171	171	100 %

5.4 Test d'Acceptation Utilisateur (UAT)

5.4.1 Profil des 25 Participants

8 patients diabétiques (40–70 ans), 7 hypertendus, 5 obèses/surpoids, 5 professionnels de santé (médecins généralistes et spécialistes).

5.4.2 Résultats de Satisfaction

Table 5.2. Résultats de satisfaction utilisateur (UAT — 25 participants)

Critère évalué	Excellent (5)	Très bien (4)	Bien (3)	Insuf. (≤ 2)	Moy./5
Facilité d'utilisation	60 %	32 %	8 %	0 %	4,52
Clarté des graphiques	68 %	28 %	4 %	0 %	4,64
Rappels médicaments	72 %	24 %	4 %	0 %	4,68
Rappels RDV médecin	80 %	16 %	4 %	0 %	4,76
Qualité rapports PDF	64 %	32 %	4 %	0 %	4,60
Interface en arabe	72 %	20 %	8 %	0 %	4,64
Satisfaction globale	68 %	28 %	4 %	0 %	4,64

5.5 Évaluation des Performances et de la Sécurité

5.5.1 Score Google Lighthouse

Table 5.3. Résultats Google Lighthouse de l'application (version production)

Critère	Score	Seuil visé	Évaluation
Performance	92 / 100	≥ 90	✓ Excellent
Accessibilité	96 / 100	≥ 95	✓ Excellent
Bonnes pratiques	100 / 100	100	✓ Parfait
SEO	95 / 100	≥ 90	✓ Excellent
PWA Score	100 / 100	100	✓ Parfait

5.5.2 Vérifications de Sécurité (OWASP Top 10)

- ✓ A01 — Contrôle d'accès : JWT (exp. 7 jours) + vérification du `userId` sur toutes les routes protégées.
- ✓ A02 — Cryptographie : `bcrypt` (rounds=12) + HTTPS/TLS (fourni par Vercel en production).
- ✓ A03 — Injection : validation Yup (frontend) + validation contextuelle (API) + schéma Mongoose (DB) sur toutes les entrées.
- ✓ A06 — Composants vulnérables : audit régulier (`npm audit`).
- ✓ A09 — Journalisation : Morgan (requêtes) + Winston (erreurs applicatives).

5.6 Résultats et Discussion

Les résultats confirment la viabilité et la pertinence de la solution. Score Lighthouse moyen : 96,6/100. Taux de réussite des tests : 100 % (171/171). Satisfaction UAT : 4,64/5. La fonctionnalité la mieux évaluée est le rappel de rendez-vous médecin (4,76/5), unanimement saluée pour son absence chez tous les concurrents.

5.6.1 Analyse des Résultats de Performance

Les scores Google Lighthouse obtenus placent l'application dans la catégorie des applications web modernes de haute qualité. Le score de Performance de 92/100 résulte de plusieurs optimisations techniques :

- **Code splitting** : **Next.js** découpe automatiquement le bundle JavaScript par page, réduisant le temps de chargement initial (First Contentful Paint) à 1,2 secondes sur réseau 4G.
- **Image optimization** : les graphiques ApexCharts sont exportés au format WebP avec compression sans perte, réduisant leur taille de 60 % par rapport au PNG standard.
- **Edge caching** : Vercel déploie les routes statiques sur un réseau CDN mondial (56 points de présence), garantissant un temps de réponse inférieur à 200 ms depuis l'Algérie.
- **Tree shaking** : seules les fonctions ApexCharts réellement utilisées sont incluses dans le bundle final, réduisant sa taille de 180 Ko.

5.6.2 Analyse des Retours Utilisateurs (UAT)

L'analyse qualitative des réponses libres (question 10 du questionnaire UAT) révèle plusieurs tendances majeures :

- **Point fort unanime** : Le rappel de rendez-vous médecin (4,76/5) est cité comme *révolutionnaire* par les patients âgés. Plusieurs participants ont mentionné avoir manqué des rendez-vous importants avant d'utiliser l'application.
- **Interface arabe appréciée** : Les participants peu familiers avec le français ont exprimé un sentiment d'inclusion numérique, certains utilisant un smartphone pour la première fois pour gérer leur santé.
- **Axes d'amélioration identifiés** : Plusieurs professionnels de santé ont souhaité accéder directement aux données patients via un portail médecin, une fonctionnalité identifiée comme priorité pour la version 2.0.
- **Valeur clinique confirmée** : Les cinq médecins participants ont indiqué que les rapports PDF générés par l'application réduisent significativement le temps de consultation (estimation : -5 à -8 minutes par consultation).

5.6.3 Comparaison avec l'État de l'Art

Comparée aux applications existantes analysées au Chapitre II, notre solution se distingue par : (1) la couverture des trois pathologies dans un système unifié avec calculs automatiques (PAM, IMC) ; (2) les rappels de rendez-vous, absents chez tous les concurrents analysés ; (3) le support natif de l'arabe RTL adapté au contexte algérien ; et (4) la gratuité totale, contrairement à MySugr (abonnement 27,99 \$/mois) ou BP+.

5.7 Conclusion

L'évaluation complète valide la qualité technique et l'utilité clinique du système. Les objectifs fixés ont été atteints et dépassés sur l'ensemble des critères évalués.

Conclusion Générale et Perspectives

Ce mémoire a présenté la conception et le développement d'une plateforme numérique intégrée dédiée au suivi des patients atteints de maladies chroniques (diabète, hypertension artérielle, obésité). Réalisé par **Khodja Belkasem** et **Ladjel Ayoub**, sous la direction de **M. Tahari Zohir**, ce projet a réuni dans un même espace technologique la rigueur médicale et la puissance des technologies web modernes.

Les résultats obtenus — score Lighthouse moyen de 96,6/100, 171 tests réussis à 100 %, satisfaction UAT de 4,64/5 — confirment la viabilité et la pertinence de la solution. La plateforme se distingue par quatre caractéristiques uniques : intégration des trois pathologies, rappels de rendez-vous médecin, génération gratuite de PDF, et support de la langue arabe avec RTL.

Perspectives

Les résultats encourageants obtenus ouvrent la voie à plusieurs évolutions naturelles de la plateforme, organisées en deux horizons temporels :

Court terme (6–12 mois)

- **Extension pathologique** : intégration de l'asthme (spirométrie, journal des crises), de la dyslipidémie (suivi du bilan lipidique) et de l'insuffisance cardiaque (suivi du poids quotidien, dèmes).
- **Portail médecin** : interface sécurisée permettant au médecin traitant d'accéder en temps réel aux données de ses patients et de leur envoyer des alertes ou des ajustements de traitement.
- **Intégration Bluetooth** : connexion directe avec des glucomètres (Contour, Accu-Chek) et tensiomètres (Omron) compatibles Bluetooth Low Energy pour la saisie automatique des

mesures.

- **Support amazighe** : traduction de l'interface en tamazight pour élargir l'accessibilité aux populations berbérophones.

Long terme (1–3 ans)

- **Intelligence artificielle** : déploiement de modèles de machine learning (LSTM, Random Forest) pour la prédiction des complications à 30 jours (ex. : risque d'hypoglycémie sévère, de crise hypertensive) et la personnalisation des recommandations.
- **Téléconsultation intégrée** : module de vidéoconférence sécurisé (WebRTC) permettant des consultations à distance, particulièrement utile pour les patients en zones rurales éloignées des centres de santé.
- **Intégration au DMP** : partenariat avec le Ministère de la Santé algérien pour l'intégration au futur Dossier Médical Partagé national, permettant l'interopérabilité avec les systèmes hospitaliers (HL7 FHIR).
- **Étude clinique** : conduite d'une étude clinique contrôlée randomisée (RCT) sur 500 patients pour mesurer l'impact réel de l'application sur les indicateurs biologiques (HbA1c, tension artérielle moyenne) après 6 mois d'utilisation.

Bibliographie

- [1] Organisation Mondiale de la Santé (2023). *Rapport mondial sur le diabète*. OMS, Genève, Suisse.
- [2] International Diabetes Federation (2023). *IDF Diabetes Atlas, 10th Edition*. Bruxelles, Belgique.
- [3] NCD Risk Factor Collaboration (2021). Worldwide trends in hypertension prevalence. *The Lancet*, 398(10304), 957–980.
- [4] American Diabetes Association (2023). Standards of Medical Care in Diabetes. *Diabetes Care*, 46(Suppl. 1).
- [5] Williams, B. et al. (2018). 2018 ESC/ESH Guidelines for the management of arterial hypertension. *European Heart Journal*, 39(33), 3021–3104.
- [6] Zhou, L. et al. (2022). Mobile health technology for type 2 diabetes management. *Journal of Medical Internet Research*, 24(1), e31051.
- [7] Omboni, S. et al. (2021). The worldwide impact of telemedicine during COVID-19. *Journal of Hypertension*, 39(6), 1047–1063.
- [8] Samson, S. L. et al. (2023). AACE/ACE Comprehensive Clinical Practice Guidelines on Obesity. *Endocrine Practice*, 29(9).
- [9] Ministère de la Santé, de la Population et de la Réforme Hospitalière (2023). *Plan national de lutte contre les maladies non transmissibles 2022–2026*. Alger, Algérie.
- [10] World Health Organization (2000). *Obesity: preventing and managing the global epidemic — Report of a WHO Consultation*. WHO Technical Report Series 894. Geneva, Switzerland.
- [11] Vercel Inc. (2024). *Next.js 14 Documentation*. <https://nextjs.org/docs>

- [12] OpenJS Foundation (2024). *Node.js 20 LTS Documentation*. <https://nodejs.org/docs>
- [13] MongoDB Inc. (2024). *MongoDB 7.0 Manual*. <https://www.mongodb.com/docs>
- [14] Visual Paradigm (2024). *UML Modeling Tool Documentation*. <https://www.visual-paradigm.com/guide>
- [15] Google Developers (2024). *Progressive Web Apps — Learn PWA*. <https://web.dev/learn/pwa>

ANNEXE A

Algorithmes de Classification Médicale

utils/medicalRanges.js -- Version complète

```
1 exports.calculateBMI = (weight, height) => {
2   const h = height / 100;
3   return parseFloat((weight / (h * h)).toFixed(1));
4 };
5
6 exports.classifyBMI = (bmi) => {
7   if (bmi < 18.5) return { cat:'underweight', label:'
    Insuffisance pondérale', color:'#3498DB' };
8   if (bmi < 25)   return { cat:'normal',      label:'
    Corpulence normale',      color:'#27AE60' };
9   if (bmi < 30)   return { cat:'overweight',  label:'
    Surpoids',                color:'#F39C12' };
10  if (bmi < 35)    return { cat:'obese1',      label:'
    Obésité grade I',         color:'#E67E22' };
11  if (bmi < 40)    return { cat:'obese2',      label:'
    Obésité grade II',        color:'#E74C3C' };
12  return { cat:'obese3', label:'Obésité morbide', color:'#8
    E44AD' };
13 };
14
15 exports.classifyBP = (systolic, diastolic) => {
16   if (systolic > 180 || diastolic > 120)
17     return { grade:'crisis', label:'Crise hypertensive',
              color:'#8E44AD', urgent:true };
18 }
```

```

18   if (systolic >= 180 || diastolic >= 110)
19     return { grade:'grade3', label:'HTA grade 3',
              color:'#E74C3C', urgent:true };
20   if (systolic >= 160 || diastolic >= 100)
21     return { grade:'grade2', label:'HTA grade 2',
              color:'#E67E22', urgent:false };
22   if (systolic >= 140 || diastolic >= 90)
23     return { grade:'grade1', label:'HTA grade 1',
              color:'#F39C12', urgent:false };
24   if (systolic >= 130)
25     return { grade:'elevated', label:'Normale haute',
              color:'#F1C40F', urgent:false };
26   return { grade:'normal', label:'Pression optimale',
            color:'#27AE60', urgent:false };
27 };
28
29 exports.classifyGlycemia = (type, value) => {
30   const t = {
31     FBG:    { hypo:70, normal:99, pre:125, crit:250 },
32     PPBG:   { hypo:70, normal:139, pre:199, crit:300 },
33     HbA1c:  { hypo:0,  normal:5.6, pre:6.4,  crit:9.0 },
34     RBG:    { hypo:70, normal:140, pre:199, crit:300 },
35   }[type];
36   if (!t) return 'unknown';
37   if (type !== 'HbA1c' && value < t.hypo) return '
    hypoglycemia';
38   if (value >= t.crit) return 'critical';
39   if (value > t.pre)   return 'diabetic';
40   if (value > t.normal) return 'prediabetes';
41   return 'normal';
42 };

```

ANNEXE B

Questionnaire UAT

Table B.1. Questionnaire du test d'acceptation utilisateur (UAT)

N ^o	Question	Évaluation (1–5)
1	L'application est-elle facile à prendre en main ?	1 2 3 4 5
2	Les graphiques de suivi biologique sont-ils clairs ?	1 2 3 4 5
3	Le système de rappels médicamenteux est-il pratique ?	1 2 3 4 5
4	Les rappels de rendez-vous médecin vous ont-ils été utiles ?	1 2 3 4 5
5	Les rapports PDF sont-ils adaptés au partage avec votre médecin ?	1 2 3 4 5
6	L'interface en langue arabe est-elle bien adaptée ?	1 2 3 4 5
7	La réactivité et la rapidité sont-elles satisfaisantes ?	1 2 3 4 5
8	L'application a-t-elle amélioré votre gestion de santé ?	1 2 3 4 5
9	Recommanderiez-vous cette app à d'autres patients ?	Oui Non P-être
10	Fonctionnalités souhaitées dans une future version ?	(Réponse libre)