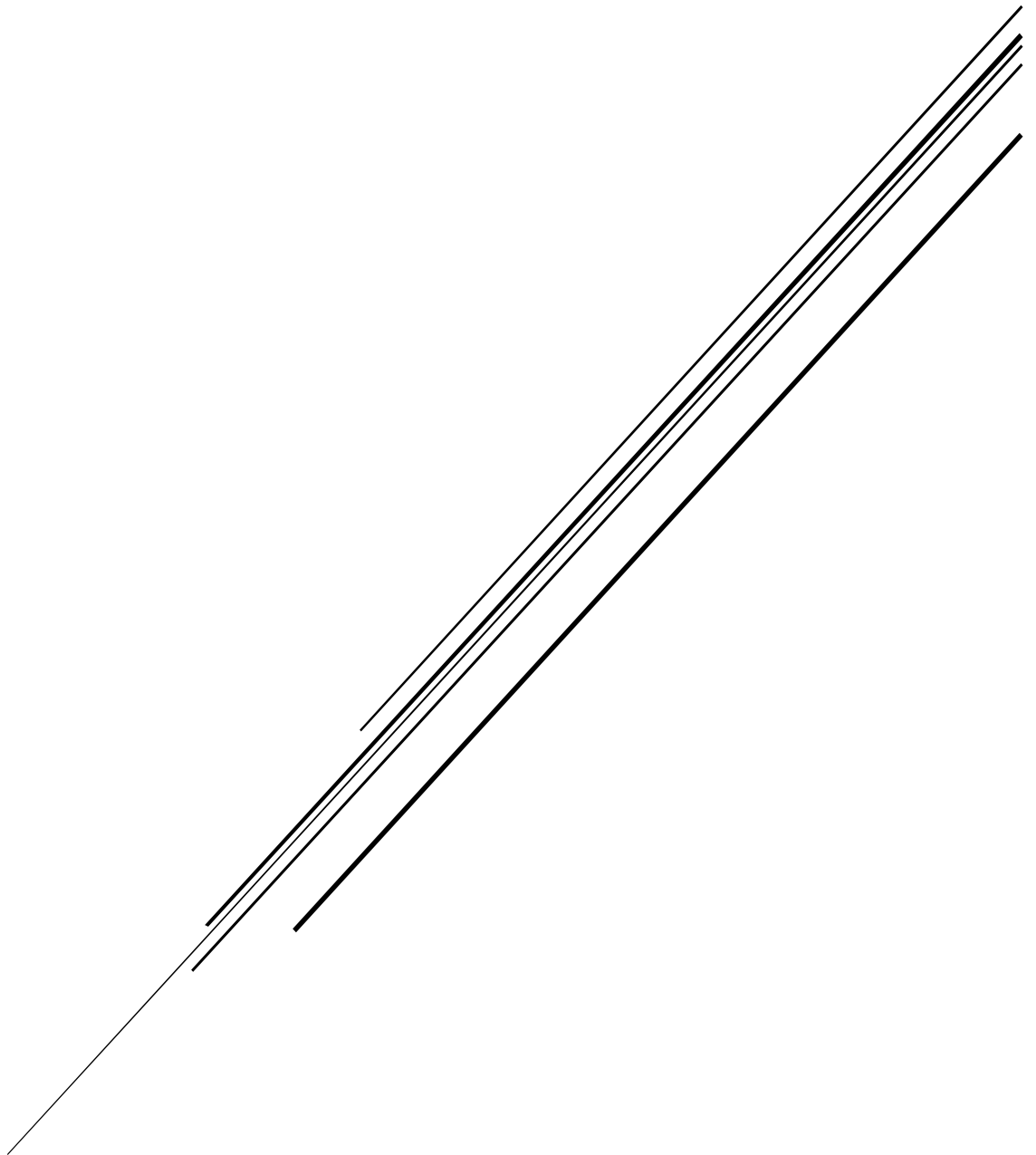


CARNET DE SANTE VIRTUEL

Cahier de charges



AJVDEC

Par Julien KOLOU – Analyste programmeur

Table des matières

I.	Contexte général	2
II.	Objectifs du projet.....	2
	Objectif principal	2
	Sous-objectifs	2
III.	Fonctionnalités principales.....	3
	1. Gestion de profil utilisateur	3
	2. Gestion des dossiers médicaux.....	3
	3. Gestion des rendez-vous médicaux	3
	4. Partage d'informations médicales.....	3
	5. Alertes et notifications	3
	6. Fonctionnalités avancées	3
IV.	Outils et méthodologie recommandés	4
	1. Technologies recommandées	4
	2. Méthodologie de travail	4
V.	Critères de performance et de succès.....	5
	1. Performance	5
	2. Compatibilité et accessibilité	5
	3. Sécurité	5
	4. Documentation	5
VI.	Phases de développement	5
	Phase 1 : Recherche et design	5
	Phase 2 : Développement	5
	Phase 3 : Tests	6
	Phase 4 : Lancement et suivi	6

I. Contexte général

La gestion des informations médicales est complexe pour les patients et les professionnels de santé. Le **Carnet de Santé Virtuel** vise à centraliser toutes les informations médicales d'un individu afin de faciliter le suivi, l'accès, et la gestion des données de santé de manière sécurisée. Ce projet inclut le développement d'une **plateforme web** et d'une **application mobile** permettant aux utilisateurs de :

- Consulter leur historique médical (consultations, vaccins, examens).
- Gérer leurs rendez-vous médicaux.
- Partager leurs informations avec les professionnels de santé de manière sécurisée.

Le projet s'inscrit dans une démarche de **digitalisation du domaine médical**, et son succès dépendra de la capacité à offrir une solution intuitive, sécurisée et performante.

II. Objectifs du projet

Objectif principal

- Créer un **Carnet de Santé Virtuel** pour permettre une **gestion sécurisée des informations médicales** personnelles et faciliter la communication entre patients et médecins.

Sous-objectifs

- Simplifier l'accès et la gestion des informations médicales (consultations, vaccins, allergies).
- Faciliter la **communication** entre les patients et les professionnels de santé grâce à un **partage sécurisé** des données.
- Mettre en place un **suivi personnalisé** avec des notifications et alertes (prises de médicaments, rappels de rendez-vous).
- Garantir un **accès multi-plateforme** (web et mobile).

III. Fonctionnalités principales

1. Gestion de profil utilisateur

- **Enregistrement des informations personnelles** : nom, âge, sexe, groupe sanguin, allergies, antécédents médicaux.
- **Mise à jour des informations** par l'utilisateur.

2. Gestion des dossiers médicaux

- **Stockage sécurisé** des documents médicaux (ordonnances, analyses, radios).
- Affichage **chronologique des consultations** et rapports médicaux.
- Gestion de l'**historique des vaccinations** et rappels.

3. Gestion des rendez-vous médicaux

- Prise de **rendez-vous en ligne** avec des médecins.
- **Notifications de rappel** pour les rendez-vous à venir.

4. Partage d'informations médicales

- Partage des informations via des **accès temporaires** pour les médecins.
- **Gestion des autorisations en temps réel.**

5. Alertes et notifications

- **Rappels personnalisés** pour les prises de médicaments, examens à venir, vaccinations.

6. Fonctionnalités avancées

a. *Suivi de la santé en temps réel*

- **Intégration avec des dispositifs de santé connectés** (montres, bracelets) pour surveiller la fréquence cardiaque, la tension, etc.

b. Intelligence Artificielle

- **Recommandations personnalisées** basées sur l'historique médical (bilans de santé, examens).

c. Système d'alertes d'urgence

- En cas d'urgence (accident, perte de conscience), un **SMS d'alerte** peut être envoyé à des contacts avec des informations critiques.

IV. Outils et méthodologie recommandés

1. Technologies recommandées

- **Frontend :**
 - **HTML, CSS, JavaScript** pour la base du front.
 - **React.js** pour la partie dynamique de l'application web.
 - **Next.js** pour gérer les aspects server-side rendering et le SEO.
 - **Apps Script** pour l'automatisation de certaines tâches administratives (e.g., gestion des formulaires).
- **Backend :**
 - **Node.js** (Express) pour la création d'une API REST.
 - **PostgreSQL** pour la gestion des données relationnelles de manière sécurisée.
- **Mobile :** Pour l'extension mobile, l'utilisation de **React Native** peut être envisagée.
- **Authentification :** Utilisation d'**OAuth 2.0** pour sécuriser les accès utilisateurs (patients et médecins).

2. Méthodologie de travail

- Travailler en **méthode Agile** avec des **sprints** de développement courts (1 à 2 semaines).
- Utilisation d'**outils de gestion de version** (comme **Git**) pour faciliter la collaboration.

V. Critères de performance et de succès

1. Performance

- **Temps de réponse API** : Moins de 300ms.
- **Disponibilité** : 99.9% de disponibilité pour garantir un accès constant aux données médicales.

2. Compatibilité et accessibilité

- **Multi-plateforme** : Accessible sur **Android**, **iOS**, et via le web sur tous les navigateurs récents.
- **Normes d'accessibilité** : Respecter les normes pour permettre l'accès aux personnes en situation de handicap.

3. Sécurité

- Mise en place de **chiffrement des données** (SSL/TLS).
- **Protection contre les attaques** (injections SQL, CSRF).

4. Documentation

- Fournir une **documentation claire** pour chaque composant (API, interfaces utilisateur) afin de faciliter la maintenance future.

VI. Phases de développement

Phase 1 : Recherche et design

- **Recherche sur les utilisateurs** cibles et leurs besoins.
- Création des **maquettes de l'application** et validation des interfaces utilisateur.

Phase 2 : Développement

- **Mise en place de la structure back-end** (API, gestion des bases de données).

- Développement du **front-end** et des **applications mobiles**.
- Intégration des **fonctions de sécurité** (authentification, gestion des accès).

Phase 3 : Tests

- **Tests unitaires** pour s'assurer que chaque fonctionnalité fonctionne correctement.
- **Tests d'intégration** pour vérifier l'interaction entre les différents modules.

Phase 4 : Lancement et suivi

- **Lancement** de l'application en production.
- **Support et maintenance** pour les corrections de bugs et les améliorations.