

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAÏA
FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT D'INFORMATIQUE



Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel en Informatique

Option : Génie Logiciel

Thème

Conception et réalisation d'une plateforme de e-santé

Réalisé par

M. BENNAI Farouk

M. MAMMASSE Amine

Soutenu le 07/07/2022 devant le jury composé de :

	Nom & Prénom	Grade	Établissement
Président	M. DJEBARI Nabil	MCB	Université de Bejaia
Examineur	M. BEDJOU Khaled	MAA	Université de Bejaia
Encadrant	M. AKILAL Abdellah	MAA	Université de Bejaia
Co-encadrante	Mme BOUCHELAGHEM Siham	MCB	Université de Bejaia

Année Universitaire : 2021 — 2022

Remerciements

Nous tenons à la fin de ce travail à remercier ALLAH le tout puissant de nous avoir donné la santé et de nous avoir permis de mener à terme ce projet.

En premier lieu, nous tenons à remercier nos encadrants, M. AKILAL Abdellah et Mme BOUCHELAGHEM Siham, pour leur disponibilité, leur patience et leur précieux suivi tout au long de la réalisation de ce travail.

Nous tenons également à remercier les membres du jury d'avoir consacré une partie de leur temps à examiner ce mémoire, pour l'intérêt qu'ils ont porté à notre travail et pour leurs contributions à l'enrichir.

Nous remercions également notre enseignant M. ACHROUFENE Achour qui n'a malheureusement pas pu assister à notre soutenance.

Enfin, nous tenons à exprimer nos sentiments de reconnaissance à toutes les personnes qui ont participé à ce projet, qui nous ont appris une infinité de choses et qui nous ont aidés, conseillés et soutenus à tout moment afin de réaliser ce travail dans les meilleures conditions.

Dédicaces

À NOS CHERS PARENTS

À nos frères, nos soeurs et nos familles

Aucune dédicace ne saurait exprimer notre respect, notre amour éternel et notre considération pour les sacrifices que vous avez consenti pour notre instruction et notre bien-être. On vous remercie pour tout le soutien et l'amour que vous nous avez portez depuis notre enfance et on espère que votre bénédiction nous accompagne toujours. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien qu'on ne vous en acquitterai jamais assez.

Puisse ALLAH, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais on ne vous déçoivent.

M. BENNAI Farouk
M. MAMMASSE Amine

Table des matières

Table des figures	v
Liste des tableaux	vi
Liste des abréviations	vii
Introduction générale	1
1 Généralités sur la e-santé	3
1.1 Introduction	3
1.2 Définition de la e-santé	3
1.3 Dossier Médical Informatisé	3
1.3.1 Caractéristiques du DMI	4
1.3.2 Intérêt du DMI	4
1.3.3 Dossier Médical Partagé	5
1.4 Télémédecine	5
1.4.1 Prise de rendez-vous en ligne	6
1.4.2 Santé mobile	6
1.4.3 Intérêt des applications dans le secteur médical	7
1.5 Obstacles et challenges	8
1.6 Conclusion	8
2 Étude préliminaire et analyse des besoins	9
2.1 Introduction	9
2.2 Présentation de l'organisme d'accueil	9
2.3 Problématique	10
2.4 Étude de marché	10
2.4.1 Application web établis en Algérie	10
2.4.2 Sites internationaux	13
2.5 Discussion et critiques	17
2.6 Solution proposée	18
2.7 Conclusion	18

3	Conception	19
3.1	Introduction	19
3.2	Cahier des charges	19
3.2.1	Contexte et objectifs	19
3.2.2	Besoins fonctionnels	20
3.2.3	Besoins non fonctionnels	21
3.2.4	Délais de réalisation	21
3.3	Processus de développement	21
3.3.1	Présentation de la méthode SCRUM	21
3.3.2	Étapes de la méthode SCRUM	22
3.4	Identification des acteurs	24
3.5	Diagrammes de cas d'utilisation	25
3.5.1	Diagramme de cas d'utilisation du « Médecin »	25
3.5.2	Diagramme de cas d'utilisation du « Patient »	31
3.5.3	Diagramme de cas d'utilisation de l'« Administrateur »	35
3.6	Diagrammes de séquence	37
3.6.1	Diagramme de séquence « Inscription d'un médecin »	37
3.6.2	Diagramme de séquence « Générer un lien de téléconsultation »	38
3.6.3	Diagramme de séquence « Partager des documents avec un patient »	39
3.6.4	Diagramme de séquence « Prendre un rendez-vous »	40
3.7	Diagramme de classes	40
3.8	Dictionnaire de données	42
3.9	MongoDB Schema	44
3.10	Conclusion	45
4	Réalisation	46
4.1	Introduction	46
4.2	Environnement et outils de développement	46
4.2.1	PlantUML	46
4.2.2	Outil de collaboration GitLab	46
4.2.3	Visual Studio Code	47
4.2.4	Android Studio	47
4.2.5	MERN stack	47
4.2.6	Node Package Manager	47
4.3	Front-end	48
4.3.1	React JS/Native	48
4.3.2	Material UI	48
4.3.3	Chakra UI	48
4.3.4	Redux	49

4.3.5	Axios	49
4.3.6	MomentJS	49
4.3.7	React Icons	49
4.3.8	React-file-base64	49
4.4	Back-end	50
4.4.1	Architecture MVC	50
4.4.2	Node.js	51
4.4.3	ExpressJS	51
4.4.4	NodeMailer	51
4.5	Implémentation de la base de données	51
4.5.1	NoSQL	51
4.5.2	MongoDB	52
4.5.3	Mongoose	52
4.5.4	Choix de la pile MERN	52
4.6	Implémentation de la téléconsultation	53
4.6.1	Agora	53
4.6.2	Web Real-Time Communication	53
4.7	Outils de déploiement	54
4.7.1	Heroku	54
4.7.2	Netlify	54
4.8	Sitemaps	54
4.9	Principales interfaces	57
4.9.1	Interfaces de l'application web	57
4.9.2	Interfaces de l'application mobile	69
4.10	Conclusion	73
	Conclusion générale	74
	Bibliographie	76
	A PV des tests sur l'application UrDocto	79

Table des figures

1.1	Terminologie de la e-santé [8].	7
2.1	Tableau de bord d'un patient sur eSiha.	11
2.2	Tableau de bord d'un patient sur DZDOC.	12
2.3	Interfaces de l'application mobile DZDOC.	13
2.4	Interface patient pour l'ajout de documents sur Doctolib.	14
2.5	Interfaces de l'application mobile Doctolib.	15
2.6	Interface d'accueil sur Livi.	16
2.7	Interfaces de l'application mobile Livi.	17
3.1	Méthode de gestion de projet SCRUM [15].	23
3.2	User stories de notre application.	23
3.3	Planification des releases.	24
3.4	Diagramme de contexte statique de l'application à réaliser.	25
3.5	Diagramme de cas d'utilisation « Gestion des patients ».	26
3.6	Diagramme de cas d'utilisation « Gestion du dossier médical patient ».	27
3.7	Diagramme de cas d'utilisation « Gestion des rendez-vous ».	28
3.8	Diagramme de cas d'utilisation « Gestion des collaborateurs ».	29
3.9	Diagramme de cas d'utilisation global associé au médecin.	30
3.10	Diagramme de cas d'utilisation « Gestion du profil ».	31
3.11	Diagramme de cas d'utilisation « Gestion des rendez-vous ».	32
3.12	Diagramme de cas d'utilisation « Gestion du dossier médical ».	33
3.13	Diagramme de cas d'utilisation global associé au patient.	34
3.14	Diagramme de cas d'utilisation global associé à l'administrateur.	36
3.15	Diagramme de séquence « Inscription d'un médecin ».	37
3.16	Diagramme de séquence « Générer un lien de téléconsultation ».	38
3.17	Diagramme de séquence « Partager des documents avec un patient ».	39
3.18	Diagramme de séquence « Prendre un rendez-vous ».	40
3.19	Diagramme de classes.	41
3.20	MongoDB Schema.	45
4.1	Schéma illustrant la MERN stack [21].	48

4.2	Fichier package.json de notre projet.	50
4.3	Architecture MVC [30].	51
4.4	Mappage d'objets entre Node et MongoDB géré via Mongoose [33].	52
4.5	Architecture de WebRTC.	54
4.6	Map de l'application web - Patient.	55
4.7	Map de l'application web - Médecin.	55
4.8	Map de l'application web - Administrateur.	56
4.9	Map de l'application mobile - Patient.	56
4.10	Interface dashboard de l'administrateur.	57
4.11	Interface administrateur sidebar.	57
4.12	Interface administrateur gestion des blogs.	58
4.13	Interface administrateur gestion des médecins.	58
4.14	Interface d'accueil.	59
4.15	Interface d'inscription du patient.	60
4.16	Interface de connexion du patient.	60
4.17	Interface dashboard du patient.	61
4.18	Interface de recherche d'un médecin.	61
4.19	Interface de prise de rendez-vous.	62
4.20	Interface de gestion du dossier médical.	62
4.21	Interface de profil du patient.	63
4.22	Interface du blog.	63
4.23	Interface de connexion du médecin.	64
4.24	Interface de l'agenda des rendez-vous du médecin.	64
4.25	Interface détails d'un rendez-vous.	65
4.26	Interface de génération d'un lien de téléconsultation.	65
4.27	Interface de téléconsultation.	66
4.28	Interface de gestion des patients.	66
4.29	Interface de gestion du dossier médical d'un patient côté médecin.	67
4.30	Interface de recherche d'un collaborateur.	67
4.31	Interface liste des collaborateurs.	68
4.32	Interface liste des documents envoyés.	68
4.33	Interface de l'application mobile pour la recherche d'un médecin.	69
4.34	Interface de l'application mobile des résultats de recherche d'un médecin.	70
4.35	Interface de l'application mobile de la liste des rendez-vous.	71
4.36	Interface de l'application mobile pour le carnet de santé.	72
4.37	Interface de l'application mobile du compte d'un patient.	73

Liste des tableaux

2.1	Tableaux comparatif.	18
3.1	Présentation de l'équipe SCRUM.	22
3.2	Dictionnaire de données.	44

Liste des abréviations

API	Application Programming Interface
CSS	Cascading Style Sheet
DMI	Dossier Médical Informatisé
DMP	Dossier Médical Partagé
DPI	Dossier Patient Informatisé
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IT	Information Technology
JSON	JavaScript Object Notation
JS	JavaScript
MEAN	MongoDB, Express, Angular, Node
MERN	MongoDB, Express, React, Node
MVC	Model-View-Controller
NoSQL	Not Only SQL
NPM	Node Package Manager
SCRUM	Systematic Customer Resolution Unraveling Meeting
TIC	Technologie de l'Information et de la Communication
UML	Unified Modeling Language
WebRTC	Web Real-Time Communication
YAML	Yet Another Markup Language

Introduction générale

La digitalisation et l'innovation sont au cœur de nos activités quotidiennes. Le secteur de la santé est autant concerné par cette transformation digitale qu'est la e-santé. Bien que ce concept ne soit pas nouveau, l'Algérie n'est pas en avance dans ce domaine. Cependant, ces dernières années ont été marquées par la volonté politique d'adopter de nouvelles dispositions pour son développement [1].

Dans un environnement dynamique et concurrentiel, les innovations en santé croissent de jour en jour. Les technologies numériques bouleversent les pratiques médicales actuelles. Sous ses différents domaines ; santé mobile, télémédecine, systèmes d'information en santé, la e-santé est aujourd'hui en plein essor. Il y a ainsi convergence de deux phénomènes. D'une part, les professionnels de santé s'emparent des outils et des services numériques pour par exemple pouvoir échanger de manière instantanée et sécurisée avec d'autres professionnels de santé autour d'un même patient, bénéficier d'outils numériques pour optimiser la gestion et fluidifier le parcours de soin ou pour soigner les patients à distance. D'autre part, les citoyens-patients cherchent à prendre la main sur leur état de santé comme sur d'autres domaines. La santé connectée et mobile permet en effet d'être plus efficace pour prendre un rendez-vous médical, pour effectuer des démarches administratives, pour consulter à distance et être suivi en continu en dehors des consultations... permettant ainsi d'améliorer la qualité de la prise en charge des patients. L'enjeu est maintenant de guider les différents professionnels de santé dans le choix et l'utilisation de ces nouveaux outils afin qu'ils les recommandent à leurs patients.

Dans ce contexte, nous sommes appelés à concevoir, développer et mettre en œuvre, un site web et une application mobile d'une plateforme de e-santé afin de faciliter la tâche au patient et au médecin, améliorer la qualité des soins de santé délivrés aux patients, augmenter le rendement du personnel médical et faciliter les tâches administratives.

Pour mener à bien notre travail, nous avons opté pour une démarche qui s'étale sur les quatre (04) chapitres suivants.

Dans le premier chapitre intitulé "*Généralités sur la e-santé*", nous présentons les concepts généraux de la e-santé en définissant le dossier médical, la télémédecine, la prise de rendez-vous en ligne et la santé mobile, ainsi que l'intérêt des applications dans le secteur médical tout en abordant les obstacles et challenges que rencontre ce domaine,

principalement en Algérie.

Le deuxième chapitre intitulé “*Étude préliminaire et analyse des besoins*” est consacré à l’étude de marché où nous avons sélectionné des applications existantes dans le domaine de la e-santé pour analyser et étudier les différentes fonctionnalités qu’elles offrent. Cette étude nous a permis d’envisager une solution à la problématique définie au début de ce chapitre.

Dans le troisième chapitre intitulé “*Conception*”, nous commençons par établir le cahier des charges afin de déterminer les acteurs qui interagiront avec le futur système à développer, ainsi que les cas d’utilisation correspondant à chaque acteur. Ces différents cas d’utilisation sont ensuite décrits par des diagrammes de séquence, ce qui nous permettra finalement d’établir notre diagramme de classes.

Le quatrième et dernier chapitre est consacré à la phase de “*Réalisation*”, nous y présentons l’environnement de développement, les langages et les outils utilisés, ainsi que les interfaces principales de notre plateforme.

Nous achevons ce mémoire par une conclusion générale résumant les points essentiels de notre travail et illustrant les perspectives d’amélioration de notre plateforme.

Chapitre 1

Généralités sur la e-santé

1.1 Introduction

Depuis plusieurs années, Internet et la téléphonie mobile ont profondément transformé nos modes de communication. Le secteur de la santé s'est saisi de ces nouvelles technologies tant du côté des professionnels de santé, que de celui des usagers. La téléphonie mobile via les applications et les objets connectés permettent de mieux se suivre et de partager facilement ses données. La santé, autrefois réservée au milieu médical, devient grâce au numérique accessible au grand public. Dans ce chapitre, nous définissons la e-santé, le dossier médical informatisé, le concept de dossier médical partagé, ses caractéristiques et son intérêt, ainsi que la télémédecine.

1.2 Définition de la e-santé

La notion d'e-santé date du début des années 2000 et représente un domaine émergent à la croisée de l'informatique médicale, de la santé publique et du monde de l'entreprise. Elle fait référence aux services de santé fournis grâce à Internet ou à des technologies numériques [2] à la fois localement et à distance. Sa définition est désormais plus large ; elle correspond à l'application des technologies de l'information et de la communication (TIC) au domaine de la santé et du bien-être. Cela correspond à du contenu numérique lié à la santé, appelé également la santé électronique ou e-santé. Elle est constituée de trois grands domaines : les systèmes d'information en santé, la télémédecine et la santé mobile (m-santé).

1.3 Dossier Médical Informatisé

Le dossier patient informatisé (DPI) est un dossier médical électronique constitué de toutes les données relatives à un patient donné, qu'elles soient de nature administrative

ou médicale. Le dossier médical informatisé (DMI) inclue les éléments d'information de prévention, de diagnostic, de traitement et de suivi du patient, ainsi que les échanges dématérialisés entre les professionnels de santé et de l'équipe de soins [3].

1.3.1 Caractéristiques du DMI

Toute conception d'un dossier personnel doit être guidée par les caractéristiques suivantes [4] :

- Être précis, concis et logique ;
- Garantir une rapidité d'accès selon les besoins ;
- Assurer la sécurité et le respect de la confidentialité ainsi que le secret médical ;
- Désigner les acteurs qui sont amenés à y porter des modification ou le consulter grâce a des droit d'accès.

1.3.2 Intérêt du DMI

L'informatisation du dossier médical permet d'apporter les avantages suivants [5] :

- Amélioration de la qualité des soins ;
- Amélioration de la lisibilité des informations ;
- Faciliter la coordination des soins entre les différents professionnels de la santé. Le dossier de santé doit permettre une prise en charge partagée des patients au sein des différentes structures de soins du réseau ;
- Faciliter la pratique professionnelle au quotidien en proposant des outils de classification permettant de retrouver rapidement les informations selon plusieurs critères ; chronologiquement, par type de données (i.e., cliniques, biologiques ou d'imagerie), par nom, par âge, par lieu de résidence ou par type de maladie ;
- Apporter une aide à la décision, l'évaluation et la recherche clinique en permettant l'utilisation de protocoles de soins prédéfinis établis à partir de référentiels de pratique. Ces accords comprendront des formulaires de saisie correspondant à des données structurées permettant d'évaluer la qualité des soins dispensés dans le réseau, des études cliniques collaboratives régionales, des études épidémiologiques et la traçabilité des parcours des patients dans le système de soins ;
- En plus de ces services, les dossiers de santé sur Internet offrent aux patients la possibilité d'accéder à leur dossier de n'importe où. De plus, il améliore la sensibilisation des patients et la gestion de leur santé grâce à la mise en place de messages d'alertes automatiques tels que les rappels de vaccination obligatoire, les examens complémentaires lors des consultations annuelles, etc.

1.3.3 Dossier Médical Partagé

Le dossier médical partagé (DMP) est un dossier médical numérique conçu pour favoriser la prévention, la qualité, la continuité et la coordination des soins apportés ou patients. Les dossiers des patients ne doivent être partagés entre professionnels de santé que si les règles déontologiques et la législation en vigueur sont pleinement respectées. En effet, toute information extraite du DMP ne doit pas être utilisée à des fins commerciales directes ou indirectes. Et, lorsqu'un médecin utilise ses observations médicales pour des publications scientifiques, il doit s'assurer que les patients ne peuvent pas être identifiés [4].

1.4 Télémedecine

La télémedecine peut prendre plusieurs formes, y compris la téléconsultation, qui permet à un professionnel de la santé de consulter un patient à distance, et la télé-expertise, qui permet à un professionnel de la santé de demander conseil à un autre professionnel de la santé sur la base d'informations médicales liées à la prise en charge d'un patient [3]. La télémedecine possède bien sûr certaines limites, comme tous les personnes ne sont pas équipés pour cette pratique en effet pour certains d'entre eux, il est difficile d'y avoir accès comme les personnes âgées ne disposant pas d'un smartphone, d'une tablette ou d'un ordinateur doivent faire appel à une tierce personne et les pratiques médicales sont davantage limitées dans des cas où l'examen physique d'un patient est nécessaire. Ce dernier devra alors inévitablement se présenter à une clinique ou se rendre à l'urgence afin de voir en personne un professionnel de la santé. La télémedecine, malgré quelques inconvénients, gagne en popularité et mérite une place au sein d'un système de santé. elle offre des avantages telle que [6] :

- Éviter la propagation de microbes : dans les salles d'attente ou cabinets médicaux. Les patients consultent de chez eux, ils ne véhiculent pas d'éventuels virus ou microbes en se déplaçant ;
- Plus rapide et réactive : la télé-consultation médicale fait gagner un temps précieux aux médecins, professionnels de santé et patients. Elle permet de partager plus rapidement et efficacement les informations entre les professionnels de santé ;
- Elle permet également de désengorger les cabinets médicaux pour des pathologies bénignes, pour communiquer des résultats d'analyse ou encore pour de simples renouvellements d'ordonnances ;
- Pas de contraintes de déplacement : Si la télé-consultation médicale permet un gain de temps précieux, c'est avant tout parce qu'elle annule toute contrainte de déplacement. Elle entre ainsi en cohérence avec l'avènement du télétravail et la

volonté de réduire les déplacements inutiles, notamment pour des raisons environnementales. C'est également une solution alternative pour les personnes à mobilité réduite ou qui n'ont pas de moyen de transport ;

- La solution aux déserts médicaux : la télé-consultation médicale est sans doute la solution la plus crédible au problème des déserts médicaux. En effet, certaines régions isolées manquent cruellement de médecins généralistes et il faut parfois faire des dizaines de kilomètres pour consulter. En Algérie, le Sahara est particulièrement concerné ;
- La télémédecine permet également d'obtenir plus facilement un rendez-vous chez un spécialiste alors que les délais sont parfois de plusieurs semaines ou mois ;
- Un meilleur suivi des patients : la télé-consultation médicale permet enfin une meilleure coordination des professionnels de santé dans le suivi des patients, notamment grâce au DMP. Ce carnet de santé numérique conserve et sécurise toutes les informations de santé du patient (e.g., traitements, résultats d'examen, allergies, etc.), qu'il peut partager avec les professionnels de santé de son choix ;

1.4.1 Prise de rendez-vous en ligne

Les rendez-vous en ligne sont une solution qui permet aux patients de prendre ou de modifier des rendez-vous sans interagir avec le personnel administratif. La plupart des systèmes de prise de rendez-vous envoient des confirmations de rendez-vous et des rappels par voie électronique. Ces systèmes peuvent également gérer les heures de rendez-vous en clinique ou les heures de consultation sur rendez-vous. Ils peuvent également être configurés pour que les médecins puissent réserver des heures de rendez-vous pour leurs propres patients. Selon les médecins utilisant le système de rendez-vous en ligne, les avantages sont principalement les suivants [9] :

- Une augmentation de l'efficacité du personnel administratif ;
- Une satisfaction accrue des patients, qui peuvent prendre rendez-vous en tout temps et sans attente ;
- Une réduction du nombre de rendez-vous manqués par les patients grâce à l'utilisation d'alertes et de rappels ;
- Une plus grande satisfaction du personnel administratif qui peut se consacrer à des tâches plus gratifiantes auprès des patients.

1.4.2 Santé mobile

Selon l'Organisation Mondiale de la Santé, la santé mobile ou m-santé recouvre *"les pratiques médicales et de santé publique reposant sur des dispositifs mobiles, tels que les*

téléphones portables, les systèmes de surveillance des patients, les assistants numériques personnels et d'autres appareils sans fil". Cela englobe également les applications concernant le mode de vie et le bien-être qui peuvent se connecter à des dispositifs médicaux ou capteurs (e.g., bracelets ou montres), ainsi que les systèmes de conseils personnalisés, les informations de santé et rappels de prise de médicaments envoyés par SMS et la télémedecine pratiquée par communication sans fil [7]. La Figure 1.1 regroupe la terminologie de la e-santé.

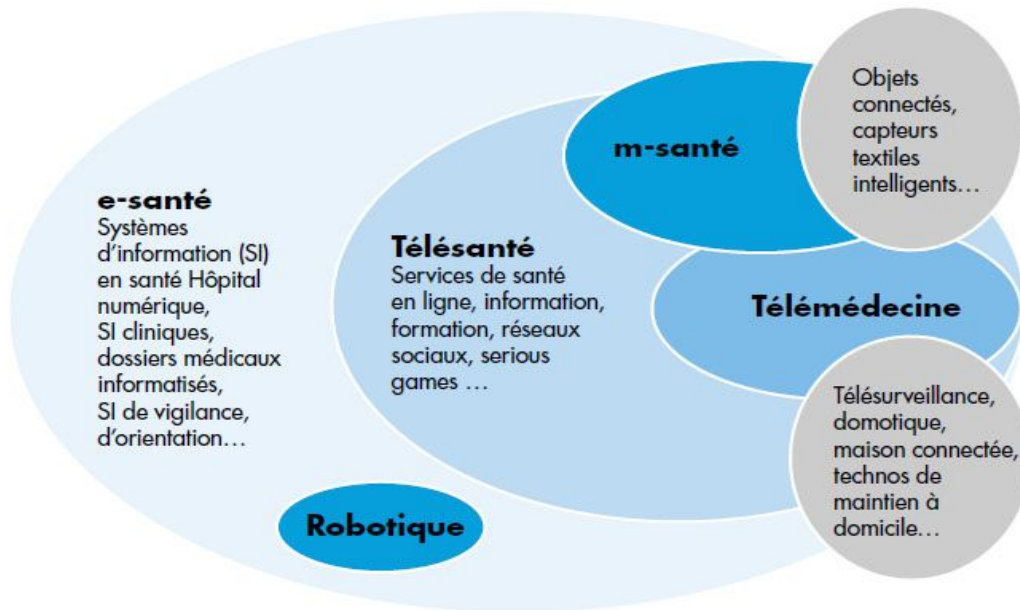


FIGURE 1.1 – Terminologie de la e-santé [8].

1.4.3 Intérêt des applications dans le secteur médical

Étant donné que le personnel médical est appelé à être la majorité de son temps de service en mouvement dans les hôpitaux ou dans les cabinets médicaux, les applications mobiles implémentant un système qui répond aux besoins des professionnels de santé représente le meilleur moyen pour faciliter leurs tâches. Ainsi cela permet aux personnels médicaux d'optimiser leur temps de service et d'améliorer leur rendement.

Pour mener à bien ce projet, nous comptons utiliser les ressources dont le web et les téléphones mobiles disposent, afin d'implémenter des fonctionnalités très pratiques pour le domaine médical, par exemple, le fait de gérer le planning des rendez-vous efficacement, pouvoir prendre un rendez-vous chez son médecin traitant sans avoir à se déplacer, etc.

1.5 Obstacles et challenges

Le marché de la e-santé est prometteur et répond aux besoins actuels du patient et du professionnel de santé, la santé connectée et mobile est une véritable opportunité pour l'optimisation de la prise en charge des patients. Cependant de nombreux challenges sont encore à relever pour que la e-santé entre dans la politique de santé globale. Ces enjeux sont liés à l'organisation du système de santé actuel, à l'accès à ces dispositifs, ou encore à des thématiques techniques et juridiques.

En outre, il n'existe pas encore de définition communément admise de la santé numérique. De manière extensive, on considère qu'elle regroupe les activités, services et systèmes appliqués aux domaines de la santé, du médico-social et du social, pratiqués à distance au moyen des TIC. Les outils numériques sont ainsi à la fois une solution et une difficulté pour l'organisation du système de santé. Cette double réalité est d'autant plus notable dans la médecine de premiers recours, qui ne bénéficie pas de la culture de l'innovation technologique des établissements sanitaires, et repose sur des formes d'organisation beaucoup plus lâches.

L'enjeu est maintenant de guider les différents professionnels de santé dans le choix et l'utilisation de ces nouveaux outils afin qu'ils les recommandent à leurs patients. Pour que les patients aient confiance en ce système, il est impératif de bien gérer la confidentialité, la sécurité et la protection des données médicales d'un patient, le secret médical étant obligatoire dans beaucoup de pays.

Aussi, parmi les challenges que nous rencontrons, la concurrence des applications existantes sur le marché et la diffusion de notre application en particulier pour les personnes âgées, pour qui il faut trouver un moyen de faciliter l'utilisation de ces nouveaux outils.

1.6 Conclusion

Le monde de la santé se lie de plus en plus au numérique, et depuis quelques années, nous constatons de plus en plus d'applications dans ce domaine. De la e-santé à la m-santé, des outils voient le jour pour les professionnels de santé et les patients. Dans ce chapitre, nous avons défini en premier lieu la e-santé. Nous avons par la suite décrit le DMI, le DMP et la télémédecine. Le chapitre suivant sera consacré à l'étude de l'existant afin de cerner le marché et pouvoir définir la structure de notre application.

Chapitre 2

Étude préliminaire et analyse des besoins

2.1 Introduction

L'étude de marché est une étape essentielle de la réussite d'un projet. Elle désigne la collecte et l'analyse d'informations relatives à un marché spécifique. L'objectif est d'en connaître toutes les caractéristiques. En effet, il ne suffit pas d'une bonne idée de départ. Il faut également connaître le marché sur lequel on va s'implanter. Dans ce chapitre, nous réalisons l'étude de marche de notre projet en analysant certaines entreprises algériennes et internationales populaires dans le domaine de la e-santé.

2.2 Présentation de l'organisme d'accueil

Du 26 Octobre 2021 au 01 Juillet 2022, nous avons effectué un stage au sein de BigNova. Le stage, c'est d'abord vivre une expérience : être en situation d'autonomie, en dehors de l'établissement universitaire, et découvrir le monde du travail. En ce sens, le stage aide à la construction de l'identité et plus globalement à la connaissance de soi. Plus largement, ce stage a été l'opportunité pour nous d'obtenir l'expérience nécessaire mais aussi l'occasion de nous sensibiliser aux questions relatives au travail en équipe.

BigNova est une entreprise algérienne, basée à Bejaia, de prestation de service et de consulting dans le domaine de l'IT (*Information Technology*) créée fin 2018. Elle s'associe aux entreprises nationales et internationales pour transformer et gérer leurs activités en libérant la valeur de la technologie. Quatre domaines prioritaires sont définis par BigNova dans le cadre de ses activités :

- Inclusion numérique ;
- Diversité et inclusion ;
- Transport et logistique ;

- Distribution et hôtellerie.

Dans le cadre du projet UrDocto, BigNova s'est associée avec Bicom qui exerce dans le secteur industriel des réseaux et de la vente, et qui veut se pencher sur le domaine de la e-santé. Cette dernière existe sur le marché Algérien depuis 2015.

Nous avons choisi d'intégrer une entreprise de petite taille telle que BigNova pour plusieurs considérations ; la première est l'objectif de pouvoir découvrir toutes les facettes du milieu plus facilement, son attachement à la spécialisation que nous étudions, le Génie Logiciel, et l'adéquation avec notre projet professionnel, mais aussi le fait qu'elle soit nouvelle, ce qui facilite les opportunités d'emploi.

2.3 Problématique

Non seulement nous sommes en phase de passage d'une période dominée par des maladies transmissibles, mais plusieurs problèmes d'ordre, stratégiques, techniques et organisationnels mettent en péril le système de santé de notre pays. Les principales entraves dans un système de santé est la surcharge des patients, la difficulté d'accès à l'information et la perte de documents médicaux. Ainsi, la santé faisant face à de multiples freins, les TIC sont de plus en plus présentes dans les systèmes de santé des pays développés. Au cours des dernières années, la croissance des réseaux de santé s'est en effet accélérée.

Aussi, choisir aujourd'hui des solutions informatiques dans le domaine de la santé en Algérie nécessite d'avoir l'esprit tourné vers l'avenir. Dans ce contexte, nous analysons les solutions existantes en Algérie, puis à l'international où ce domaine est bien plus évolué, pour déterminer la meilleure approche concernant la solution que nous envisageons.

2.4 Étude de marché

Dans cette section, nous analysons certaines des plateformes médicales les plus connues et les plus utilisées, en Algérie et à l'international.

2.4.1 Application web établis en Algérie

Nous avons choisi d'étudier les plateformes eSiha et DZDOC présentées ci-après.

2.4.1.1 eSiha

C'est la première plateforme médicale multi-services en Algérie : recherche de médecins et prise d'informations, dossier médical électronique, diagnostic à distance, et gestion médicale électronique de l'établissement de santé [11].

- **Adresse du site** : <https://www.esiha.net/>
- **Technologies utilisées** : TypeScript et Angular
- **Année de création** : 2019
- **Nombre de praticiens** : 10 850 à ce jour
- **Nombre d'utilisateurs** : 77 000 (le 06 mars 2022)
- **Connexion quotidienne** : 4 000 (le 06 mars 2022)
- **Application mobile** : Plus de 50 000 téléchargements.
- **Fonctionnalités pour le patient** :
 - Inscription/connexion en tant que patient ;
 - Rechercher un médecin par spécialité et région ;
 - Prendre un rendez-vous en ligne pour une consultation en présentiel ou pour une téléconsultation ;
 - Dossier et historique médical et suivi continu avec des courbes statistiques.
- **Fonctionnalités pour le médecin** :
 - Inscription/connexion en tant que médecin ;
 - Planification des rendez-vous dans un agenda.

La figure 2.1 représente le dashboard patient de eSiha

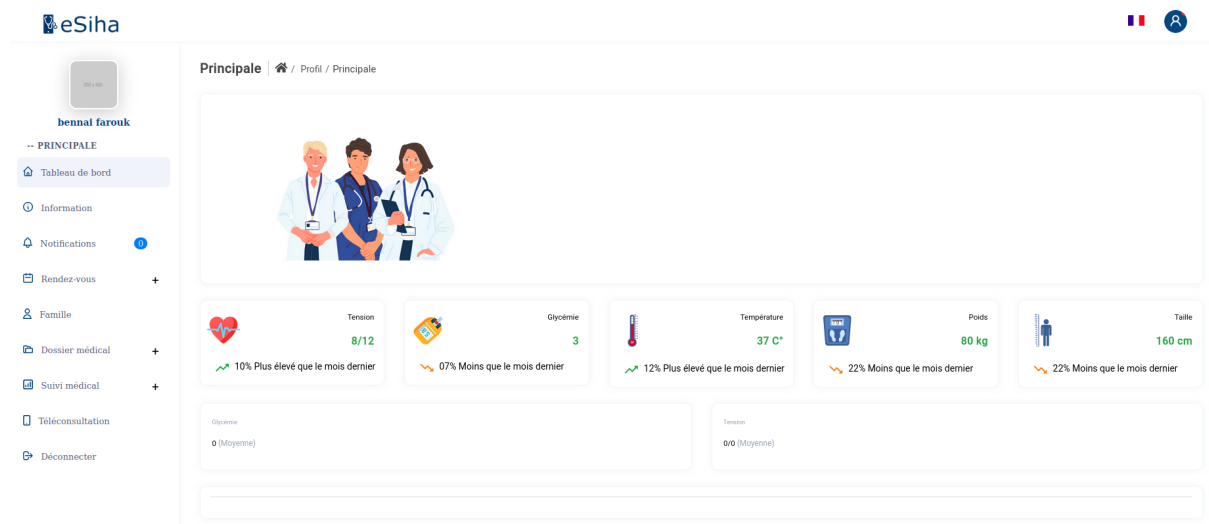


FIGURE 2.1 – Tableau de bord d'un patient sur eSiha.

2.4.1.2 DZDOC

C'est une plateforme médicale multi-services créée par Khidma Tech, une société de service en ingénierie informatique qui bénéficie d'un large savoir-faire dans le domaine des

technologies de l'information. En Mai 2015, DZDOC lance le premier service de prise de rendez-vous en ligne en Algérie afin d'offrir aux médecins et aux patients algériens une solution simple, efficace et confortable pour la gestion de leurs rendez-vous médicaux [10].

- **Adresse du site** : <https://dzdoc.com/>
- **Technologies utilisées** : PHP et JQuery
- **Année de création** : 2015
- **Application mobile** : Plus de 1000 téléchargements
- **Fonctionnalités pour le patient** :
 - Inscription/connexion en tant que patient ;
 - Rechercher un médecin par spécialité et région ;
 - Rechercher une clinique par nom, wilaya et commune ;
 - Prendre un rendez-vous en ligne uniquement pour des téléconsultations ;
 - Des rappels de rendez-vous par e-mail ou par SMS ;
 - Une liste des médecins favoris ;
 - Un espace conseils santé (i.e., un blog).
- **Fonctionnalités pour le médecin** :
 - Inscription/connexion en tant que médecin ;
 - Planification des rendez-vous dans un agenda.

La figure 2.2 représente le dashboard patient de DZDOC

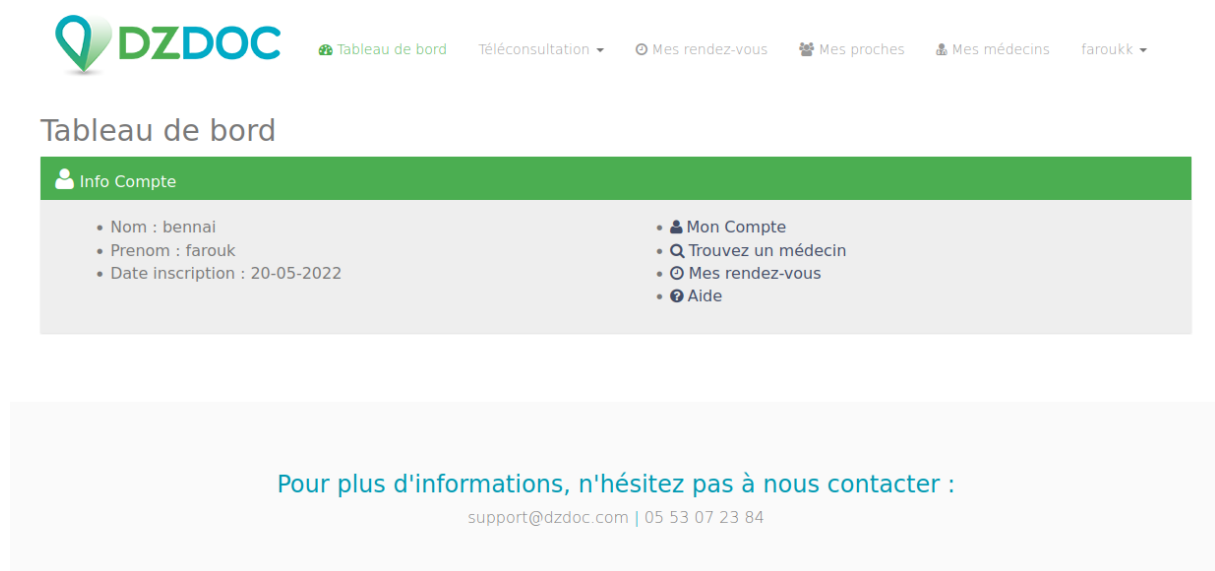


FIGURE 2.2 – Tableau de bord d'un patient sur DZDOC.

La figure 2.3 représente quelques interfaces de l'application mobile patient de DZDOC

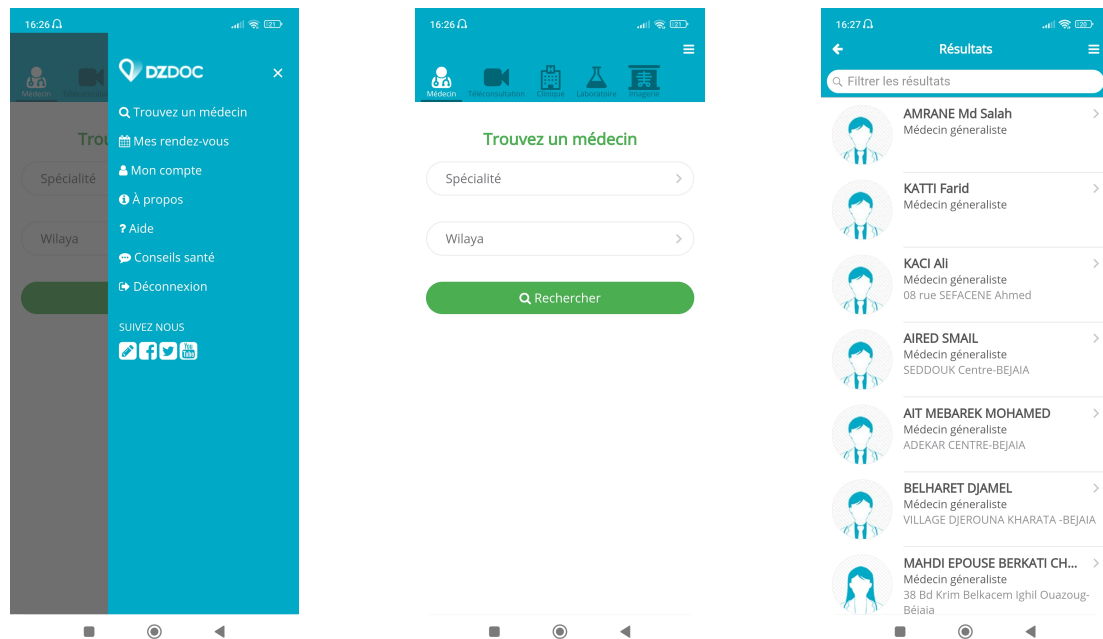


FIGURE 2.3 – Interfaces de l'application mobile DZDOC.

2.4.2 Sites internationaux

Nous avons choisi d'étudier les plateformes Doctolib et Livi présentées ci-après.

2.4.2.1 Doctolib

Doctolib est une entreprise française fondée en 2013 par Stanislas Niox-Chateau qui propose un logiciel de gestion des rendez-vous et des services aux patients. Pour les professionnels de santé, Doctolib a créé le premier service complet en France de gestion des rendez-vous médicaux et de développement de patientèle, en France, en Italie et en Allemagne [12].

- **Adresse du site** : <https://www.doctolib.fr/>
- **Technologies utilisées** : Ruby et React
- **Année de création** : 2013
- **Nombre de praticiens** : 300 000
- **Nombre d'utilisateurs** : 60 millions
- **Consultations vidéo** : 14 millions
- **Application mobile** : Plus de 10 millions de téléchargements
- **Classement** : 1^{ère} place en France
- **Fonctionnalités pour le patient** :
 - Inscription/connexion en tant que patient ;
 - Rechercher un médecin par spécialité et région ;

- Prise de rendez-vous en ligne ;
- Consultation vidéo avec remboursements ;
- Partage des ordonnances en pharmacie ;
- Gestion du dossier médical ;

• **Fonctionnalités pour le médecin :**

- Un logiciel médical avec gestion documentaire, partage de documents aux patients, dossier patient clair et complet, gestion optimisée de la consultation, facturation et télétransmission accélérées ;
- Un logiciel de gestion des rendez-vous et des services aux patients avec prise de rendez-vous en ligne ;
- Agenda et partage en ligne des ordonnances et de documents sur un compte patient sécurisé ;
- Messagerie instantanée sur ordinateur et sur mobile, gratuite et sécurisée, des professionnels de santé et annuaire de santé ;
- Partage sécurisé de documents, discussions collaboratives autour d'un cas patient, adressage en ligne des patients.

La figure 2.4 représente l'interface gestion des documents patient de Doctolib

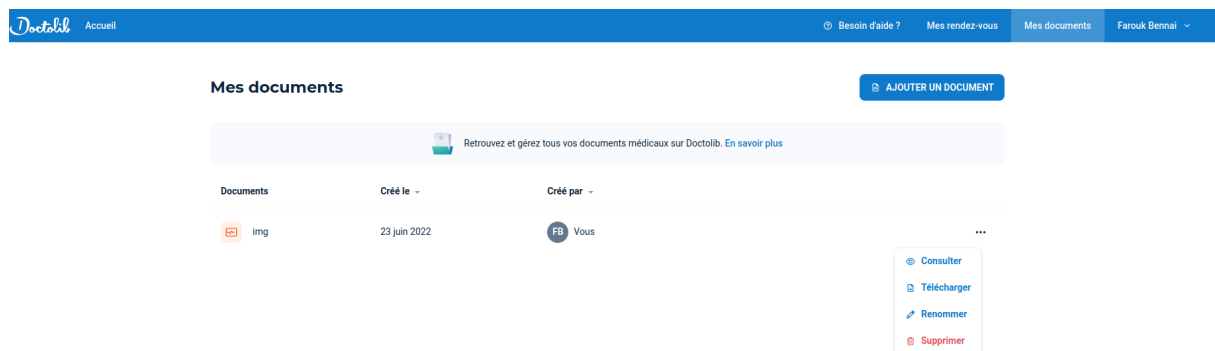


FIGURE 2.4 – Interface patient pour l'ajout de documents sur Doctolib.

La figure 2.5 représente quelques interfaces de l'application mobile de Doctolib.

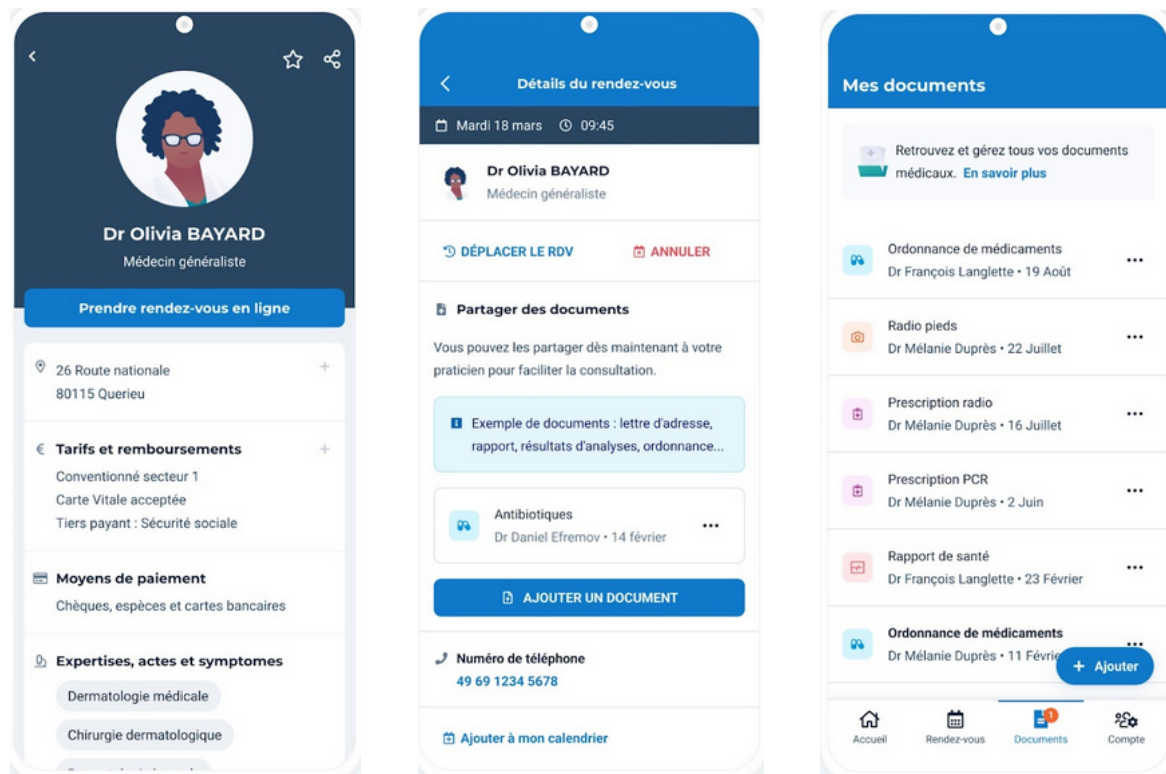


FIGURE 2.5 – Interfaces de l'application mobile Doctolib.

2.4.2.2 Livi

Lancée en 2015 en Suède et en Septembre 2018 en France, l'équipe médicale est composée de professionnels de santé diplômés et inscrits à l'Ordre National des Médecins, tous engagés dans la révolution de la consultation en ligne. Livi est un offereur de soins de santé qui propose des consultations médicales à distance auprès de praticiens enregistrés et exerçant en France, et formés à la télémédecine. En effet, il offre un service de consultation entièrement en ligne dédié à la médecine généraliste et aux spécialistes de toutes catégories [13].

- **Adresse du site** : <https://www.livi.fr/>
- **Technologies utilisées** : Nextjs et Nodejs
- **Année de création** : 2015
- **Application mobile** : Plus de 1 million de téléchargements
- **Classement** : 61^{ème} place en France
- **Fonctionnalités pour le patient** :
 - Inscription/connexion en tant que patient ;
 - Rechercher un médecin par spécialité et région ;
 - Prise de rendez-vous en ligne ;
 - Ordonnances électroniques valables en pharmacie ;

- Consultations médicales remboursables comme en cabinet ;
- Téléconsultations réalisées sur Livi remboursables par l'assurance maladie ;
- Blog santé.

- **Fonctionnalités pour le médecin :**

- Inscription/connexion en tant que médecin ;
- Planification des rendez-vous dans un agenda.

La figure 2.6 représente l'accueil de Livi.



FIGURE 2.6 – Interface d'accueil sur Livi.

La figure 2.7 représente quelque interfaces de l'application mobile de Livi.

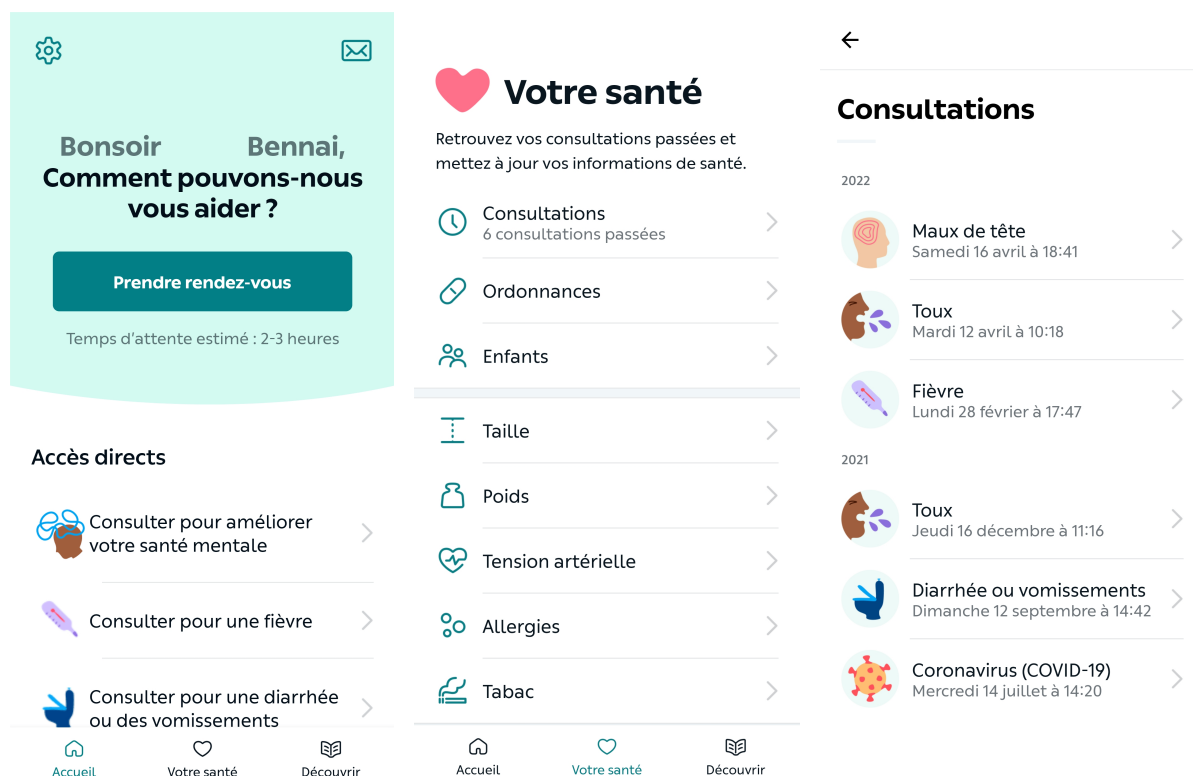


FIGURE 2.7 – Interfaces de l'application mobile Livi.

2.5 Discussion et critiques

Après avoir consulté chaque site web et application mobile précédemment présentés et testé les différentes fonctionnalités proposées, nous avons constaté quelques insuffisances qui auraient besoin d'être améliorées :

- Des interfaces peu conviviales (DZDOC) ;
- Les rendez-vous en ligne sont proposés uniquement pour les téléconsultations ;
- Certains médecins inscrits sur la plateforme ne sont pas joignable ;
- Peu de services sont proposés aux médecins et aux patients (DZDOC et eSiha).

La table 2.1 représente une comparaison entre les différentes applications étudiées ci-dessus

	eSiha	DZDOC	Doctolib	Livi
Gestion du dossier médical	Disponible	Non disponible	Disponible	Non disponible
Pris de rendez vous en ligne	Disponible pour téléconsultation	Disponible pour téléconsultation	Disponible pour téléconsultation et présenteielle	Disponible pour téléconsultation
Téléconsultation	Disponible	Disponible	Disponible	Disponible
Partage de documents	Non disponible	Non disponible	Disponible entre médecin et patient	Non disponible
Recherche de médecin	Disponible	Disponible	Disponible	Disponible
Collaboration entre médecin	Non disponible	Non disponible	Disponible	Non disponible
Gratuité	Gratuit	Gratuit	Gratuit pour patient et payante pour les médecin	Gratuit

TABLE 2.1 – Tableaux comparatif.

2.6 Solution proposée

En tenant compte des critiques, et en réponse aux objectifs de notre travail, nous envisageons de développer une application web et mobile pour la gestion du dossier médical des patients et la prise de rendez-vous chez un médecin. Notre application doit garantir la disponibilité des informations sur les médecins et les patients, ainsi que la sécurité des données confidentielles des patients tout en respectant certaines règles d'utilisabilité telles que la simplicité de navigation web et une bonne ergonomie.

2.7 Conclusion

Dans ce chapitre, nous avons étudié certaines applications de e-santé en analysant les différentes fonctionnalités proposées afin de cerner les objectifs de notre futur système. Nous avons ensuite exposé la solution envisagée à la problématique énoncée au début de ce chapitre. Pour implémenter notre système, nous devons mener une étude conceptuelle prenant en compte les acteurs et les cas d'utilisation identifiés. C'est justement le sujet du prochain chapitre.

Chapitre 3

Conception

3.1 Introduction

La première étape de la conception consiste à analyser la situation pour tenir compte des contraintes, des risques et de tout autre élément pertinent afin de développer un système répondant aux besoins du client sous forme d'un cahier des charges. Le présent chapitre, nous permet d'identifier toutes les fonctionnalités de notre future application pour chaque type d'utilisateurs en recensant les besoins fonctionnels, et permet d'établir la liste des exigences traduites par les besoins non fonctionnels. Ceci se fera par l'identification des acteurs et la définition de tous les besoins qui seront par la suite modélisés par des diagrammes de cas d'utilisation, suivi des diagrammes de séquence, et enfin du diagramme de classes.

3.2 Cahier des charges

Dans cette section, nous présentons notre cahier des charges et décrivons les objectifs principaux de l'application que nous comptons développer.

3.2.1 Contexte et objectifs

Notre projet porte sur la réalisation d'une application web et mobile de santé. L'objectif principal consiste à concevoir et réaliser une application web et mobile d'une plateforme de e-santé. Une telle application devrait offrir l'intégrité, la sécurité et la confidentialité des données, ainsi que l'accès à l'information au moment voulu, en faisant en sorte de :

- Perfectionner les solutions de gestion pour aider le médecin à mieux exploiter les systèmes informatiques, accroître son rendement et participer ainsi au progrès informatique ;

- Améliorer et simplifier le travail administratif en confiant à une machine les procédures lourdes et répétitives de l'organisation, comme la gestion des rendez-vous ;
- Proposer une meilleure expérience aux patients en améliorant l'accès aux soins ;
- Gagner du temps pour le patient en prenant un rendez-vous en ligne et ainsi éviter l'attente.

Ce projet vise les clients algériens ; médecins et patients.

3.2.2 Besoins fonctionnels

Notre projet consiste en la réalisation de trois (03) applications ; une pour le médecin, une pour le patient, et une autre pour l'administrateur.

- **Pour le médecin :** nous devons regrouper toutes les fonctionnalités nécessaires de gestion pour faciliter et organiser le travail des médecins. Parmi ces fonctionnalités, on retrouve :
 - Possibilité de créer un compte avec confirmation par l'administrateur ;
 - Avoir un planning où le médecin pourra organiser ses rendez-vous ;
 - Possibilité d'offrir au patient le service de prise de rendez-vous en ligne ;
 - Possibilité d'organiser des téléconsultations ;
 - Gestion des patients (i.e., consulter, ajouter, modifier et supprimer) ;
 - Gestion du dossier médical d'un patient (i.e., consulter, ajouter, modifier et supprimer) ;
 - Partager des documents avec les patients ;
 - Partager des documents avec d'autres médecins.
- **Pour le patient :**
 - Possibilité de créer un compte ;
 - Gestion du dossier médical (i.e., consulter, ajouter, modifier et supprimer) ;
 - Partager des documents avec un médecin ;
 - Prise de rendez-vous en ligne avec choix entre présentiel ou téléconsultation ;
 - Recherche d'un médecin par spécialité, adresse et nom ;
 - Avoir un blog de conseils.
- **Pour l'administrateur :**
 - Confirmation des comptes des médecins ;
 - Gestion des médecins ;
 - Gestion des utilisateurs ;
 - Gestion du blog.

3.2.3 Besoins non fonctionnels

Les besoins non fonctionnels décrivent les objectifs liés aux performances du système et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d'objectifs spécifiques que doit atteindre le système. Ainsi, notre application doit pouvoir couvrir les besoins non fonctionnels suivants :

- **Maintenabilité** : le code source doit être compréhensible par simple lecture, notamment en respectant les règles de gestion et les normes de développement ;
- **Exploitabilité** : les impacts en termes de performances doivent être pris en compte lors du développement, ainsi que la consommation des ressources (i.e., CPU, mémoire, etc.) qui doit être minimisée ;
- **Convivialité et ergonomie** : le système doit être facilement utilisable et disposer d'interfaces conviviales ; le design doit permettre une identification immédiate de ses différents éléments pour permettre à l'utilisateur d'accéder de manière intuitive à ce qu'il cherche, dès la première utilisation ;
- **Sécurité** : l'obligation pour accéder à l'application de s'authentifier par un nom d'utilisateur et un mot de passe affectés par l'administrateur.

3.2.4 Délais de réalisation

Délai de mise en test souhaité : finalisation des tâches prioritaires avant le 01/07/2022.

Délai de mise en production souhaité : non défini.

3.3 Processus de développement

Pour satisfaire les exigences du client, au moindre coût et dans les moindres délais, et mener à bien notre projet, nous avons opté pour la méthode agile "SCRUM".

3.3.1 Présentation de la méthode SCRUM

SCRUM (Systematic Customer Resolution Unraveling Meeting) est une méthode agile consacrée à la gestion de projet qui s'adapte aux projets Web et qui permet aux développeurs d'être au plus près des besoins du marché, en leur évitant de développer des systèmes qui ne seront pas utilisés. Son objectif phare est d'améliorer la productivité des équipes, tout en permettant une optimisation du produit grâce à des feedbacks réguliers du marché. En parallèle, la méthode SCRUM permet d'avoir une vue d'ensemble du projet pour chacune des parties prenantes, de réduire les bugs et d'avoir une mise à jour régulière des priorités [14].

L'équipe de la méthode SCRUM est composée de :

1. **Scrum Master** : c'est le guide de l'avancement du projet, celui qui s'assure que les principes et les valeurs du Scrum sont respectés, et qui vérifie la communication entre les équipes. Il améliore aussi la productivité et il lève les obstacles.
2. **Product Owner** : c'est l'expert qui collabore avec le client. Souvent le fondateur ou le chef d'équipe, il définit à la suite des feedbacks clients les spécificités fonctionnelles du produit, puis les priorise avec l'équipe et valide les fonctionnalités développées.
3. **Équipe de développement** : dans la méthode SCRUM, il n'est pas censé y avoir de hiérarchie entre les membres de l'équipe, quand bien même leurs savoir-faire et compétences seraient différents.

Pour notre projet, les rôles sont repartis comme représenté dans la table 3.1 et concernât nous les développeur dans la cadre de ce projet notre rôle n'est pas limité ou développement, on n'a aussi états dans les autres rôles :

Rôles SCRUM	Personne affectées
Scrum Master	M. AKILAL Abdellah
Product Owner	Responsable de stage
Équipe de développement	M. BENNAI Farouk et M. MAMMASSE AMINE

TABLE 3.1 – Présentation de l'équipe SCRUM.

3.3.2 Étapes de la méthode SCRUM

Trois (03) grandes étapes constituent la méthode SCRUM :

- **Étape 1 : Product Backlog**

Durant cette phase, le Product Owner rencontre le client et analyse ses besoins. Il identifie toutes les fonctionnalités dont le produit devra être composé (les *user stories*), dans ce qui s'appelle le *Product Backlog*. Ce cahier des charges n'est pas fixé pour toujours, et pourra évoluer en fonction des besoins du client et l'avancement du projet. L'équipe décide de ce qu'elle peut faire et dans quel ordre le faire.

- **Étape 2 : Sprint**

La méthode SCRUM se caractérise par une répartition de chacune des tâches à faire. L'équipe trie les fonctionnalités et les tâches, qu'elle répartit dans des *sprints* (durée de cycle de deux semaines). Pendant ce cycle, l'équipe s'occupera par exemple uniquement de coder une fonctionnalité du produit qu'elle devra livrer à la fin de cette phase. Avant chaque sprint, une réunion de planification est organisée, c'est le *sprint planning meeting*.

- **Étape 3 : Sprint Review**

À chaque fois qu'un sprint est terminé, on teste les bénéfices de la fonctionnalité avec le Product Owner. On fait une démonstration de ce qui a été créé, puis le destinataire de la fonctionnalité (e.g., le client) confirme ou non si la fonctionnalité marche comme il le souhaitait.

La Figure 3.1 illustre le déroulement de la méthode SCRUM.

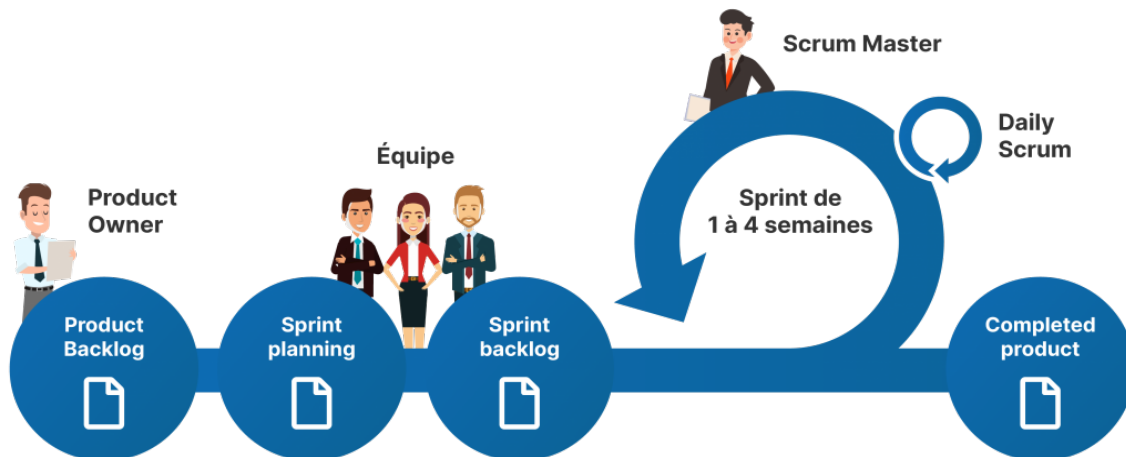


FIGURE 3.1 – Méthode de gestion de projet SCRUM [15].

Pour la réalisation de notre projet et après avoir rencontré le client et analysé ses besoins, nous avons défini les fonctionnalités à développer sous forme de *user stories*. Ces dernières, sont des phrases simples dans le langage de tous les jours permettant de décrire une fonctionnalité, comme illustré par la Figure 3.2.

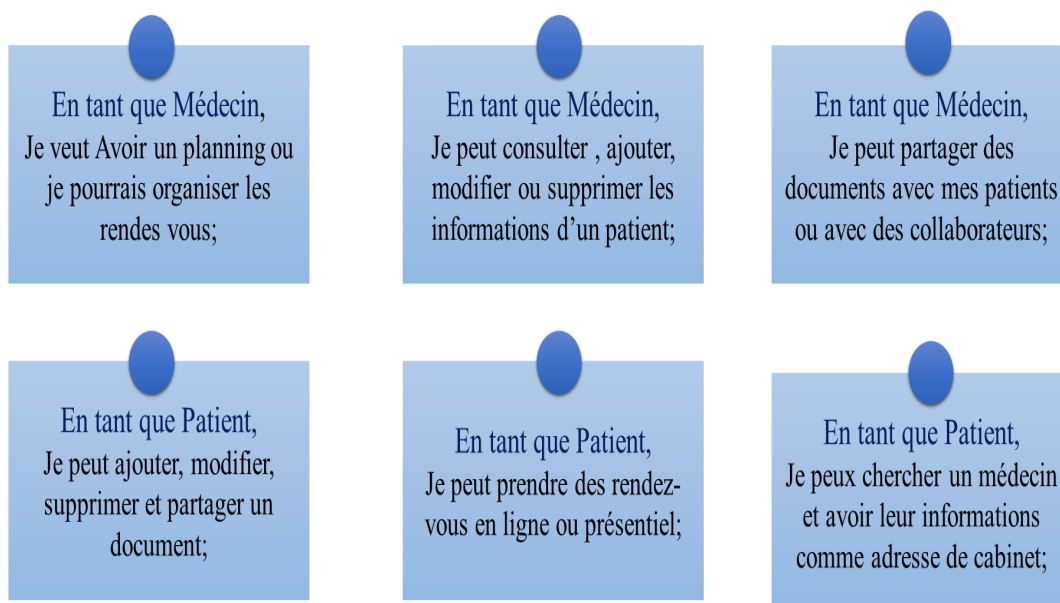


FIGURE 3.2 – User stories de notre application.

Après avoir identifié les user stories, nous avons réparti les tâches selon des sprints d'une durée limite de réalisation (en fonction de la tâche à réaliser) et nous avons créé un plan de releases illustré par la Figure 3.3. Avant chaque sprint, nous avons effectué un meeting pour confirmer la tâche précédente et évoquer la tâche suivante.

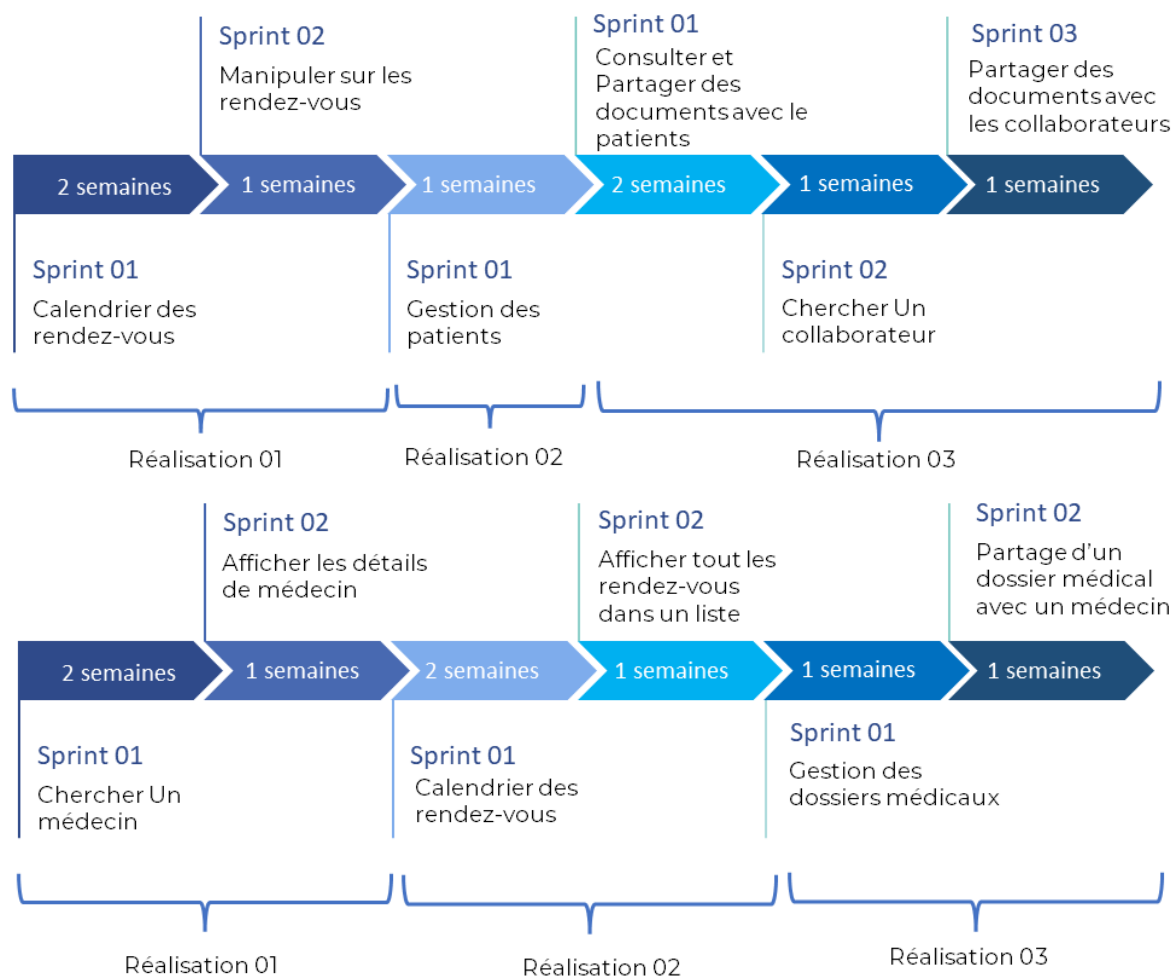


FIGURE 3.3 – Planification des releases.

3.4 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données. Dans le cas de notre système, nous avons identifié principalement trois (03) acteurs en interaction avec celui-ci :

- **Médecin** : il désigne la personne habilitée à pratiquer la médecine et chargée d'examiner les patients, de diagnostiquer les maladies, lésions et pathologies afin

de les traiter, et de gérer les rendez-vous et les tâches administratives.

- **Patient** : il désigne la personne examinée par le médecin et qui se voit administrer un traitement.
- **Administrateur** : c'est la personne chargée de la maintenance de l'application et de la gestion des comptes des utilisateurs. Il veille au bon fonctionnement du serveur de données et à sa sécurité.

La Figure 3.4 représente le diagramme de contexte statique de l'application à développer.

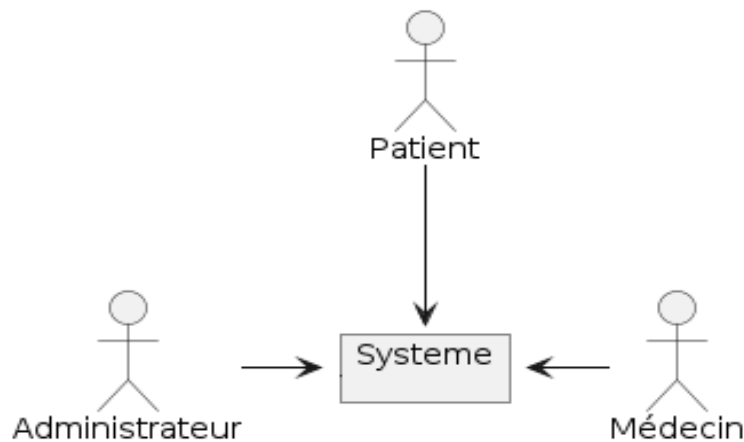


FIGURE 3.4 – Diagramme de contexte statique de l'application à réaliser.

3.5 Diagrammes de cas d'utilisation

Un cas d'utilisation (*use case*) représente un ensemble de séquences d'action qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

Reprenons un à un les trois acteurs et listons les différentes façons qu'ils ont d'utiliser le futur système.

3.5.1 Diagramme de cas d'utilisation du « Médecin »

- Inscription/connexion en tant que médecin (il faut attendre la confirmation du compte de la part de l'administrateur).
- Gestion des patients :
 - Consulter les informations d'un patient ;
 - Ajouter un patient à sa liste ;

- Modifier les informations d'un patient ;
- Supprimer un patient de sa liste ;
- Rechercher un patient ;
- Consulter la liste des patients (avec filtrage par nom, prénom, date de naissance, numéro de téléphone et e-mail).

L'ensemble de ces cas d'utilisation sont représentés sur la Figure 3.5.

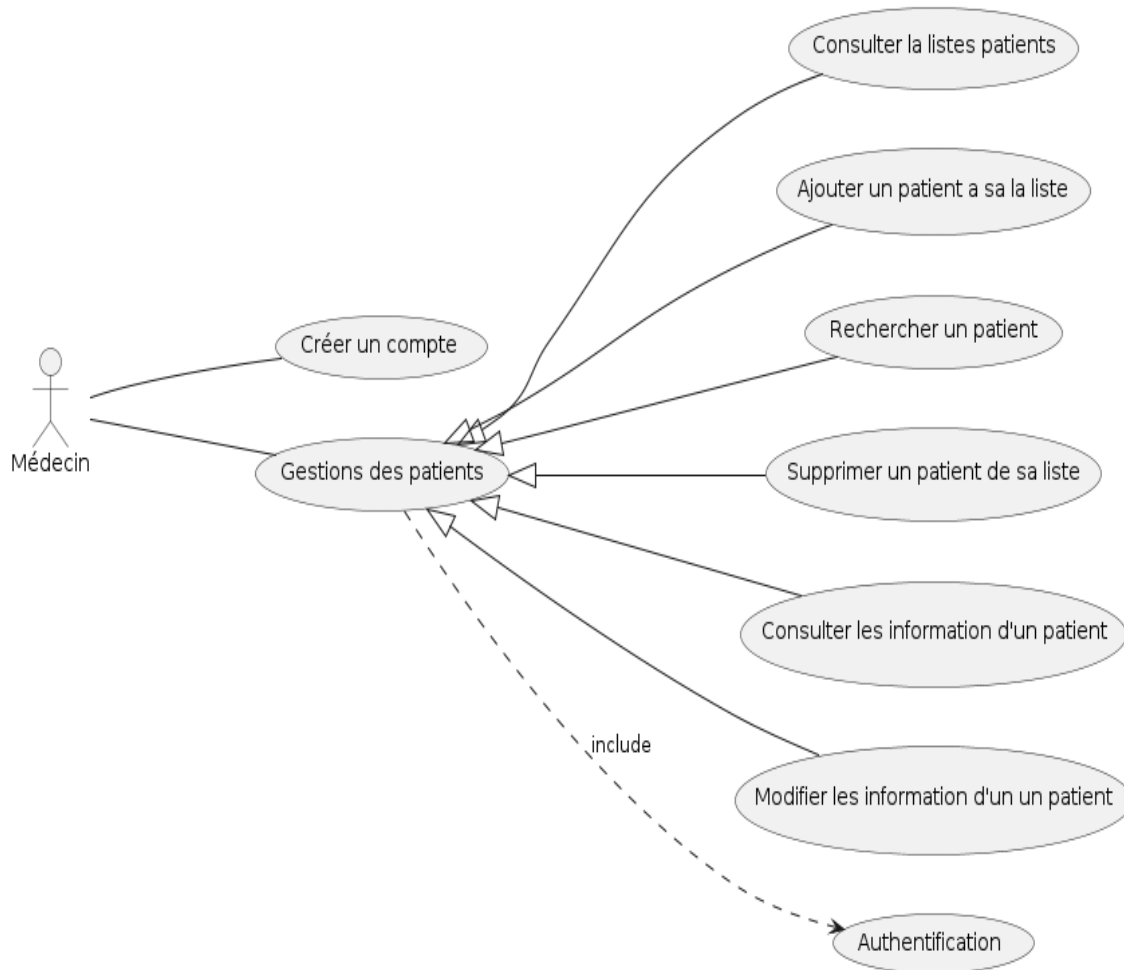


FIGURE 3.5 – Diagramme de cas d'utilisation « Gestion des patients ».

- Gestion du dossier médical patient :
 - Consulter le dossier médical d'un patient (avec filtrage global) ;
 - Ajouter un document ou dossier médical du patient ;
 - Modifier le dossier médical d'un patient ;
 - Supprimer un document du dossier médical d'un patient ;
 - Partager des documents avec un patient ;
 - Télécharger un document du dossier médical d'un patient.

La Figure 3.6 représente l'ensemble des cas d'utilisation associé à la gestion du DMP.

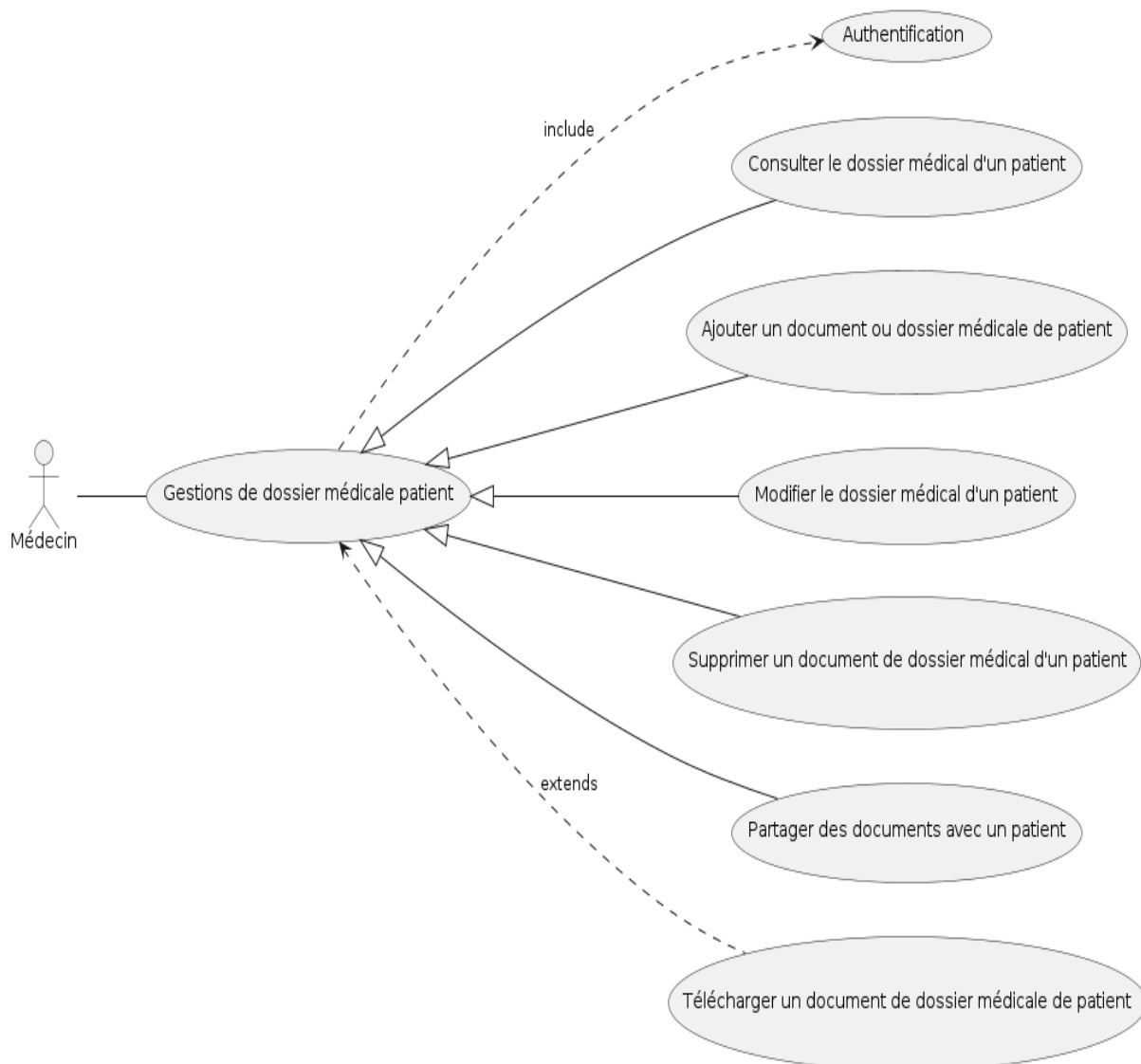


FIGURE 3.6 – Diagramme de cas d'utilisation « Gestion du dossier médical patient ».

- Consulter le planning des rendez-vous (par liste, jour, semaine et mois).
- Gestion des rendez-vous :
 - Indiquer, modifier et supprimer la disponibilité ;
 - Indiquer la durée d'une consultation ;
 - Modifier un rendez vous ;
 - Générer un lien de téléconsultation (si le rendez-vous est en ligne).

L'ensemble de ces cas d'utilisation sont représentés sur la Figure 3.7.

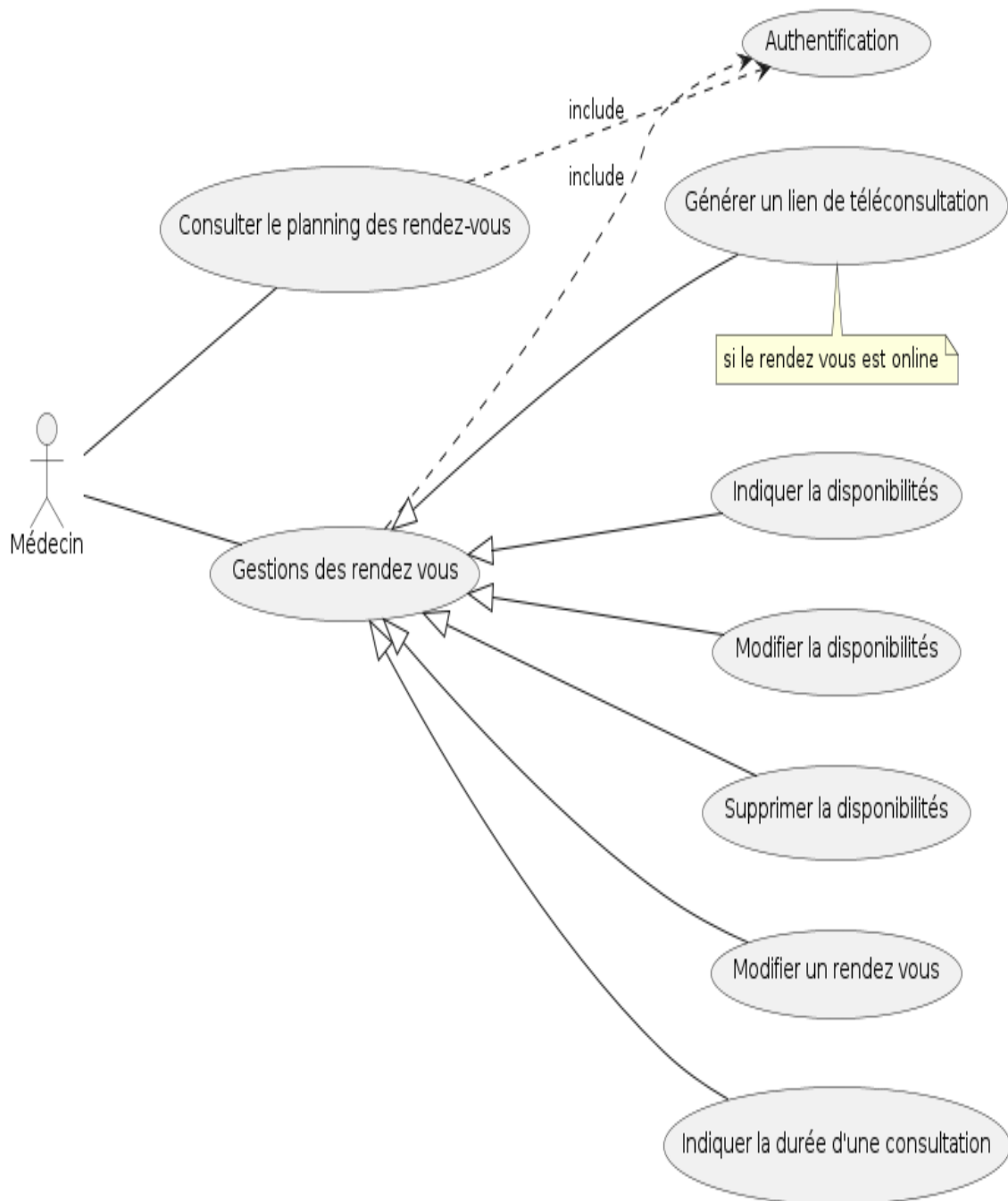


FIGURE 3.7 – Diagramme de cas d'utilisation « Gestion des rendez-vous ».

- Gestion des collaborateurs :
 - Consulter la liste des collaborateurs ;
 - Rechercher un collaborateur ;
 - Ajouter un collaborateur à la liste ;
 - Supprimer un collaborateur de la liste ;
 - Partager des documents avec un collaborateur.

La Figure 3.8 illustre les cas d'utilisation associés à la gestion des collaborations.

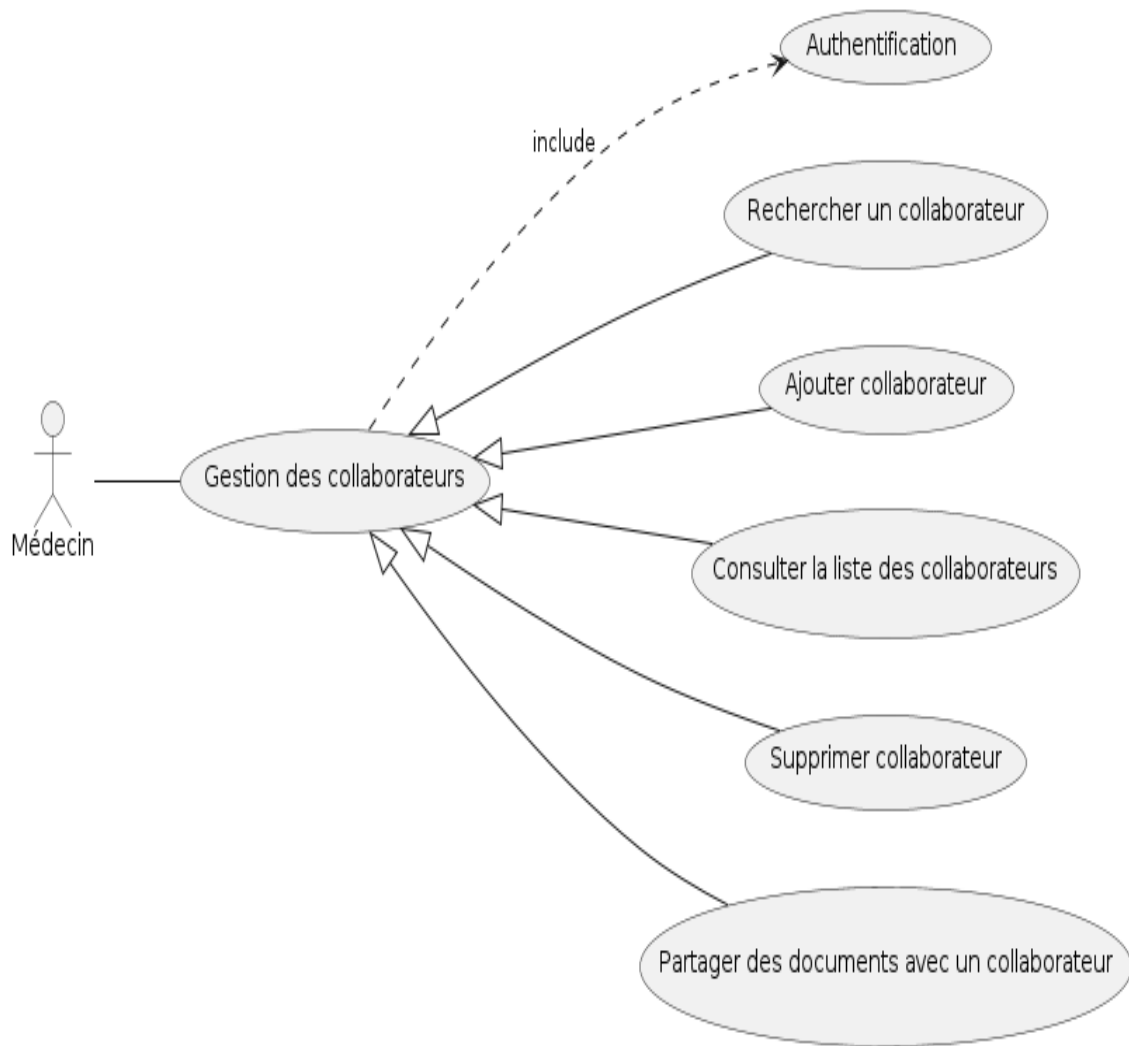


FIGURE 3.8 – Diagramme de cas d'utilisation « Gestion des collaborateurs ».

La Figure 3.9 représente le diagramme de cas d'utilisation global associé au médecin.

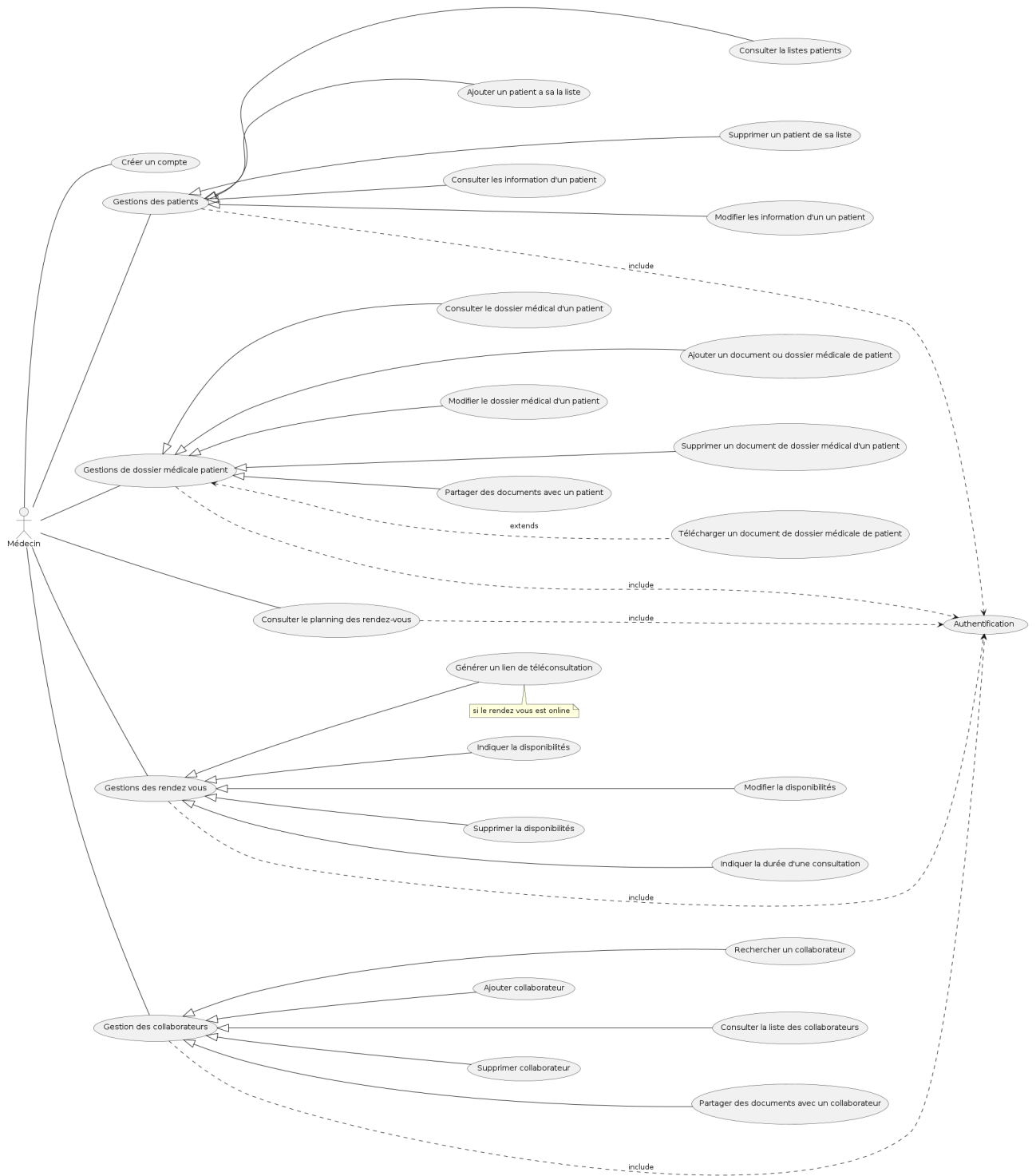


FIGURE 3.9 – Diagramme de cas d'utilisation global associé au médecin.

3.5.2 Diagramme de cas d'utilisation du « Patient »

- Inscription/connexion en tant que patient :
 - Validation par e-mail lors de l'inscription ; un lien est envoyé à l'adresse mail du patient lors de son inscription ;
 - Créer un nouveau mot de passe en cas de mot de passe oublié.
- Gestion du profil :
 - Consulter les informations du profil ;
 - Modifier les informations du profil ;
 - Ajouter une personne à contacter en cas d'urgence ;
 - Ajouter/supprimer des antécédents ;
 - Ajouter/supprimer des traitements.
- Consulter le blog et ajouter un commentaire.

L'ensemble de ces cas d'utilisation sont représentés sur la Figure 3.10.



FIGURE 3.10 – Diagramme de cas d'utilisation « Gestion du profil ».

- Prendre un rendez-vous en ligne ou en présentiel :
 - Rechercher un médecin selon la spécialité, l'adresse et le nom ;
 - Consulter les détails d'un médecin ;
 - Consulter la disponibilité d'un médecin ;
 - Participer à la téléconsultation avec le médecin (si le rendez-vous est en ligne).
- Consulter la liste des rendez-vous.
 - Annuler un rendez vous.
- Créer une liste des médecins favoris :
 - Ajouter un médecin ;
 - Supprimer un médecin.

La Figure 3.11 représente l'ensemble des cas d'utilisation associés à la gestion des rendez-vous.

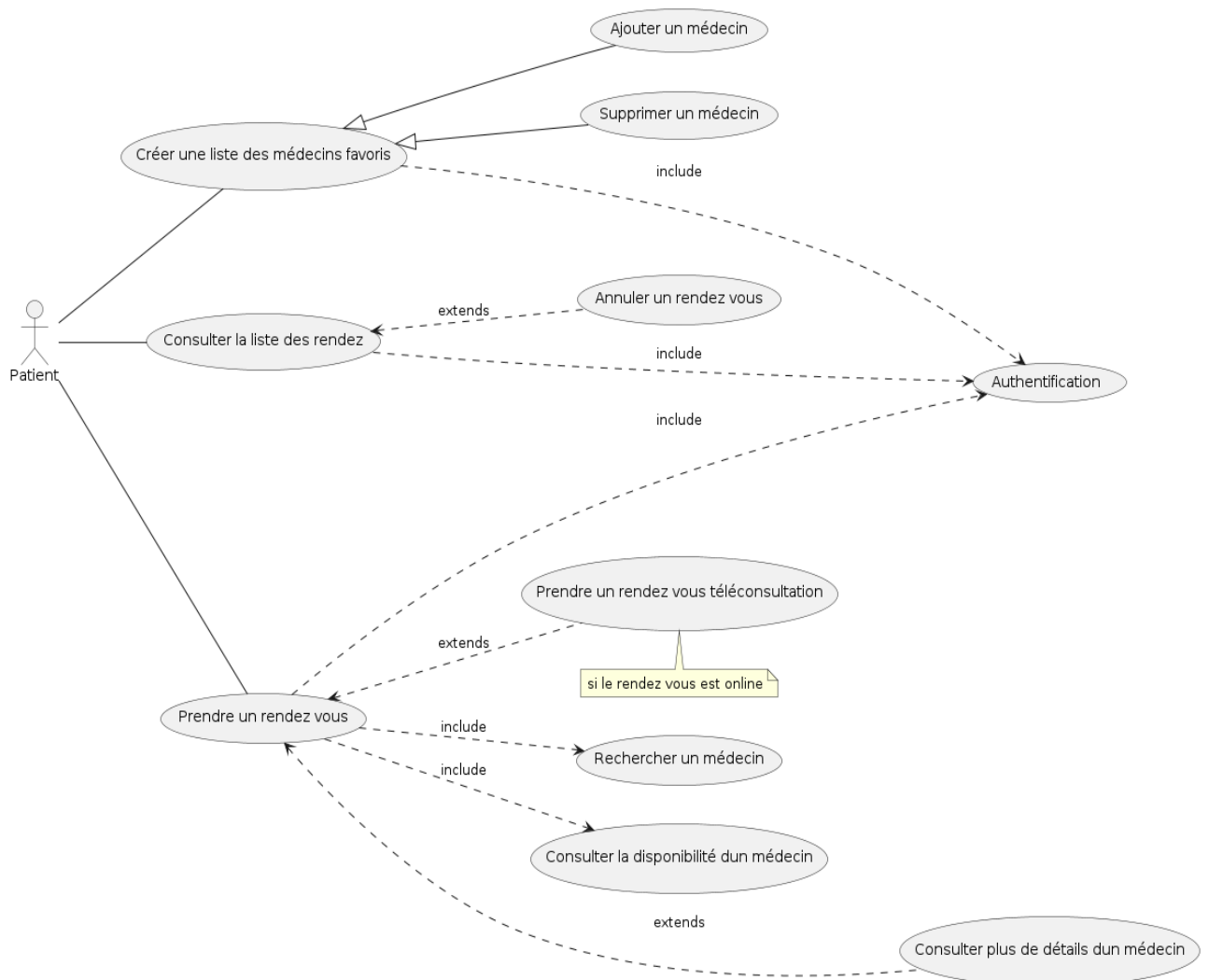


FIGURE 3.11 – Diagramme de cas d'utilisation « Gestion des rendez-vous ».

- Gestion du dossier médical :
 - Ajouter un document au dossier médical ;
 - Supprimer un document ;
 - Consulter les informations d'un document ;
 - Modifier les informations d'un document ;
 - Télécharger ou consulter un document ;
 - Partager un ou plusieurs documents en offrant des vues sur les documents sélectionnés au médecin choisi.

L'ensemble de ces cas d'utilisation sont représentés sur la Figure 3.12.

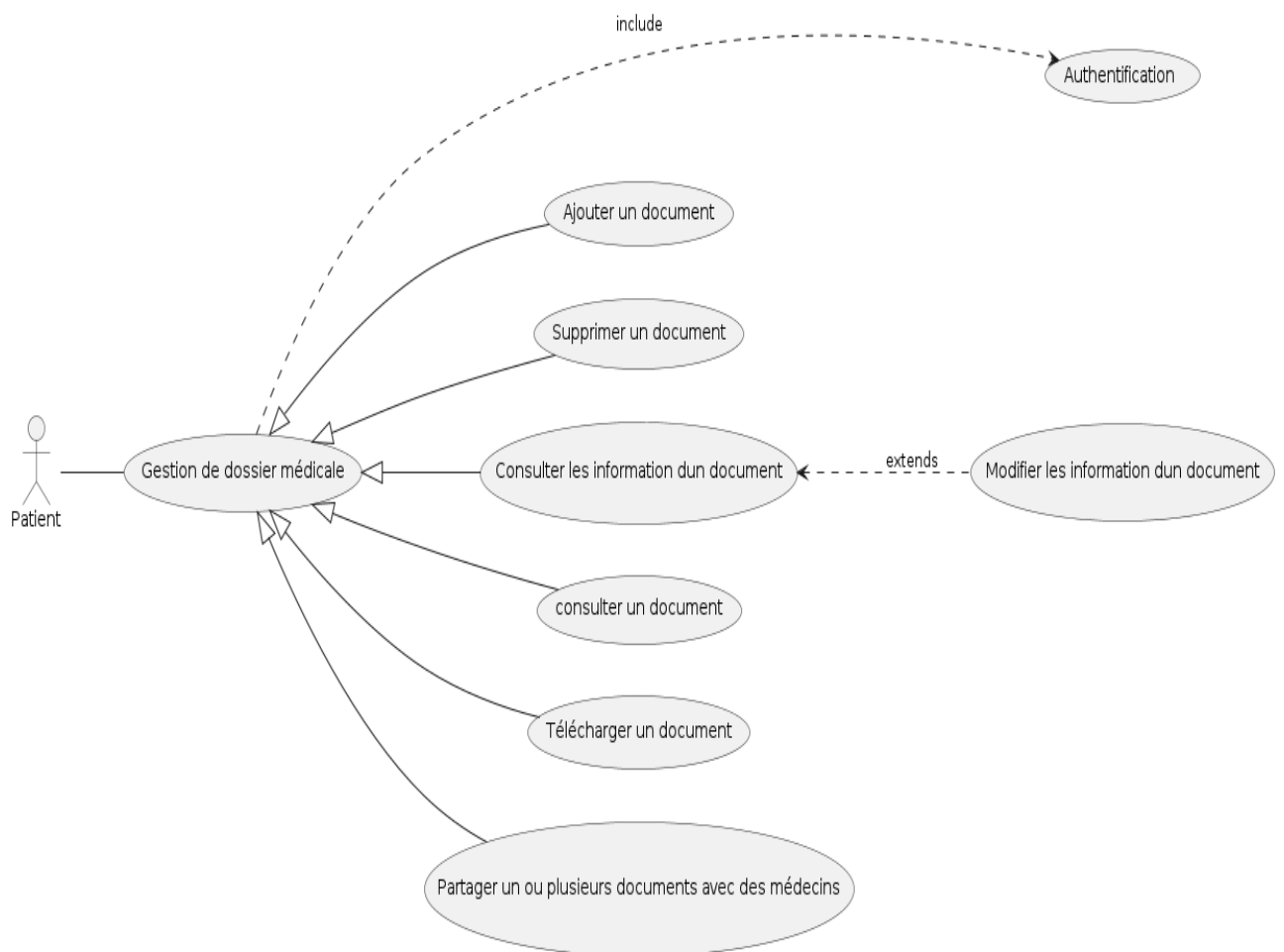


FIGURE 3.12 – Diagramme de cas d'utilisation « Gestion du dossier médical ».

La Figure 3.13 représente le diagramme de cas d'utilisation global associé au patient.

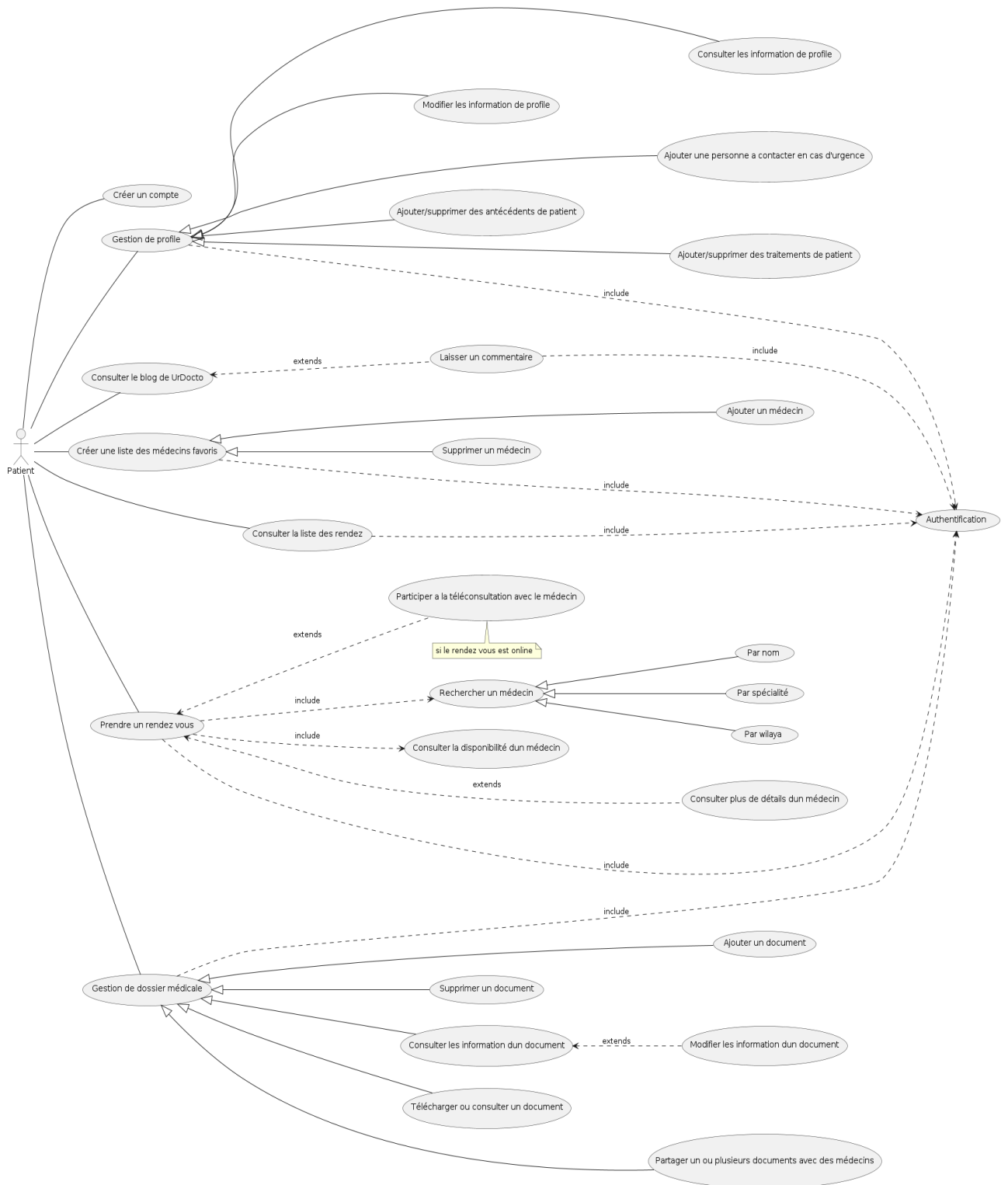


FIGURE 3.13 – Diagramme de cas d'utilisation global associé au patient.

3.5.3 Diagramme de cas d'utilisation de l'« Administrateur »

- Gestion des médecins :
 - Confirmer, refuser ou bloquer le compte d'un médecin (dans le cas où l'abonnement n'a pas été renouvelé) ;
 - Supprimer le compte d'un médecin uniquement dans le cas où l'état de son compte est "refusé" ;
 - Consulter la liste des médecins ;
 - Consulter les informations d'un médecin.
- Gestion des blogs :
 - Ajouter un blog ;
 - Modifier un blog ;
 - Supprimer un blog.
- Consulter la liste des utilisateurs.
- Consulter les statistiques.
- Gestion des administrateurs :
 - Consulter la liste des administrateurs ;
 - Ajouter un administrateur ;
 - Supprimer un administrateur.

La Figure 3.14 représente le diagramme de cas d'utilisation global associé à l'administrateur.

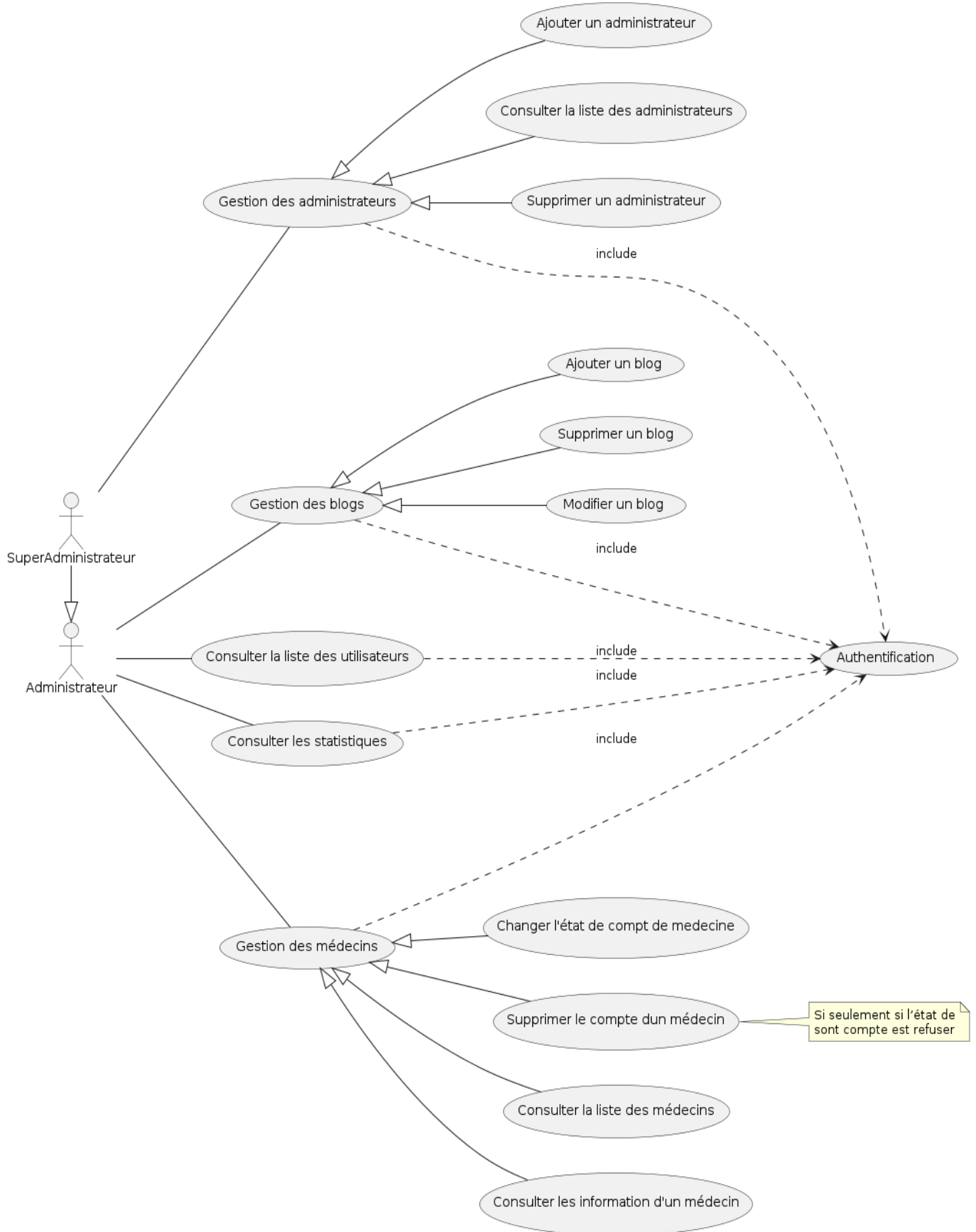


FIGURE 3.14 – Diagramme de cas d'utilisation global associé à l'administrateur.

3.6 Diagrammes de séquence

L'objectif des diagrammes de séquence est de représenter les interactions entre les objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés. Dans ce qui suit, nous représentons le diagramme de séquence d'un scénario représentatif de chacun des cas d'utilisation décrits précédemment.

3.6.1 Diagramme de séquence « Inscription d'un médecin »

La Figure 3.15 représente le diagramme de séquence du cas d'utilisation « Inscription d'un médecin ».

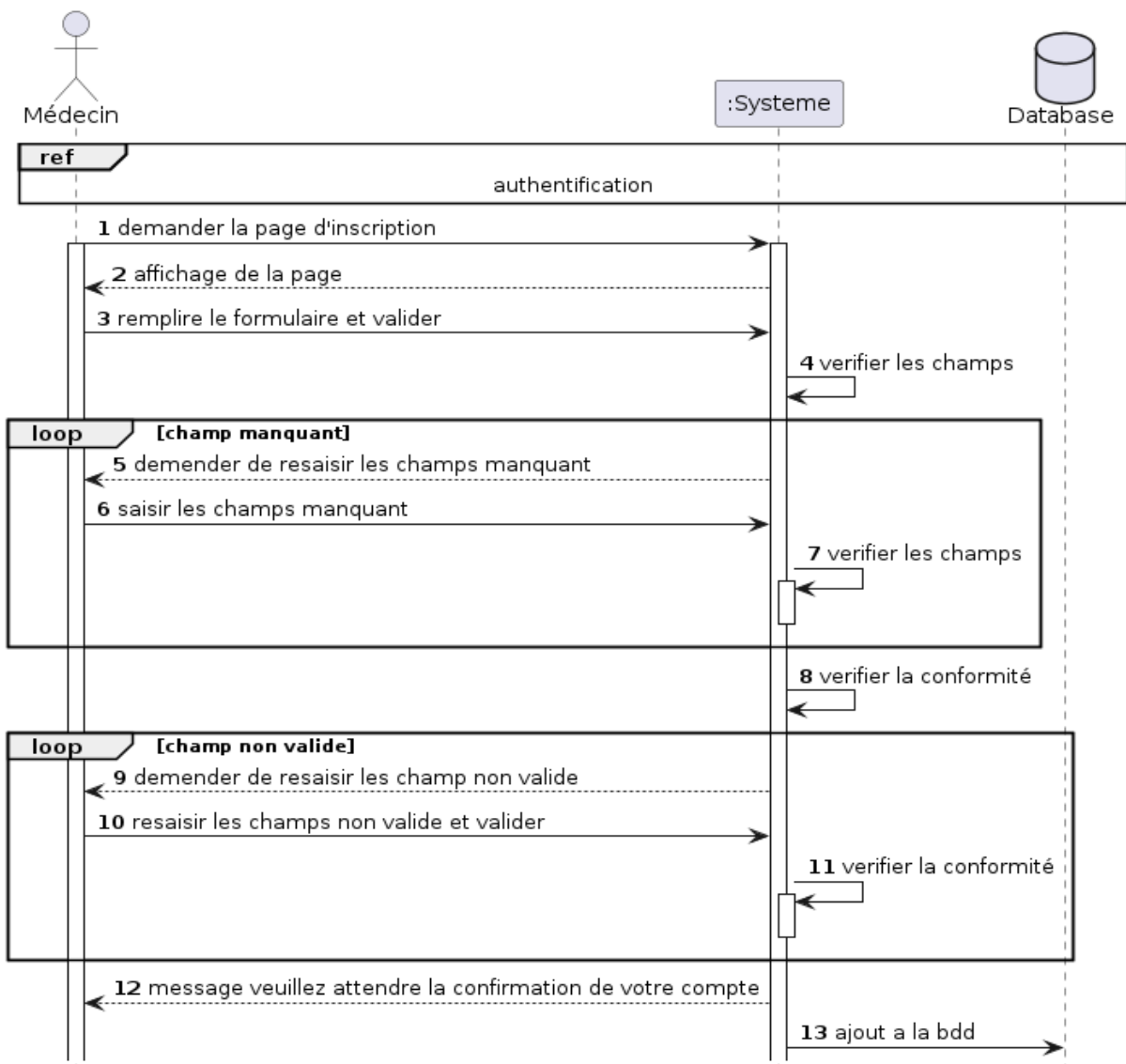


FIGURE 3.15 – Diagramme de séquence « Inscription d'un médecin ».

3.6.2 Diagramme de séquence « Générer un lien de téléconsultation »

La Figure 3.16 représente le diagramme de séquence du cas d'utilisation « Générer un lien de téléconsultation ».

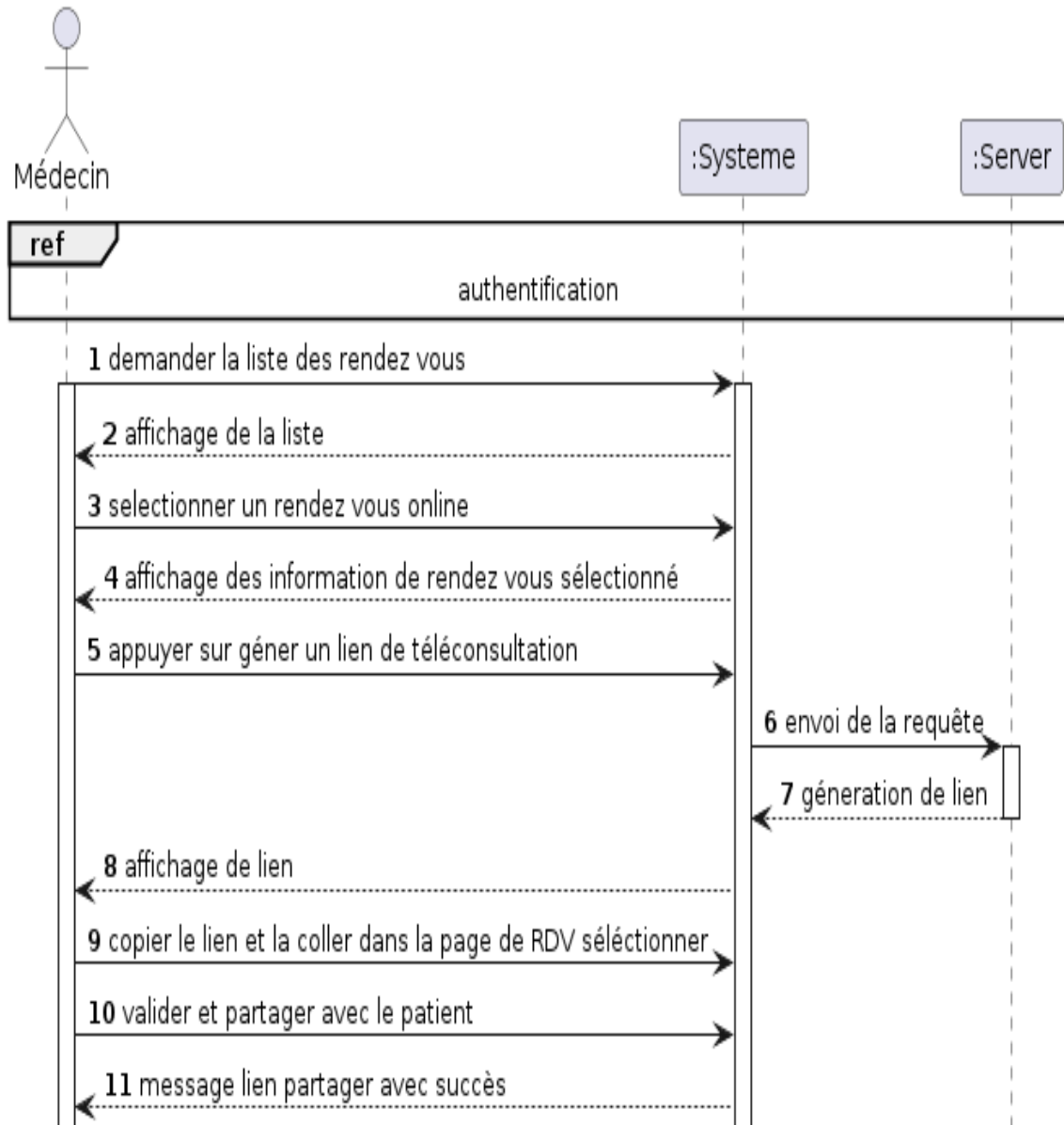


FIGURE 3.16 – Diagramme de séquence « Générer un lien de téléconsultation ».

3.6.3 Diagramme de séquence « Partager des documents avec un patient »

La Figure 3.17 représente le diagramme de séquence du cas d'utilisation « Partager des documents avec un patient ».

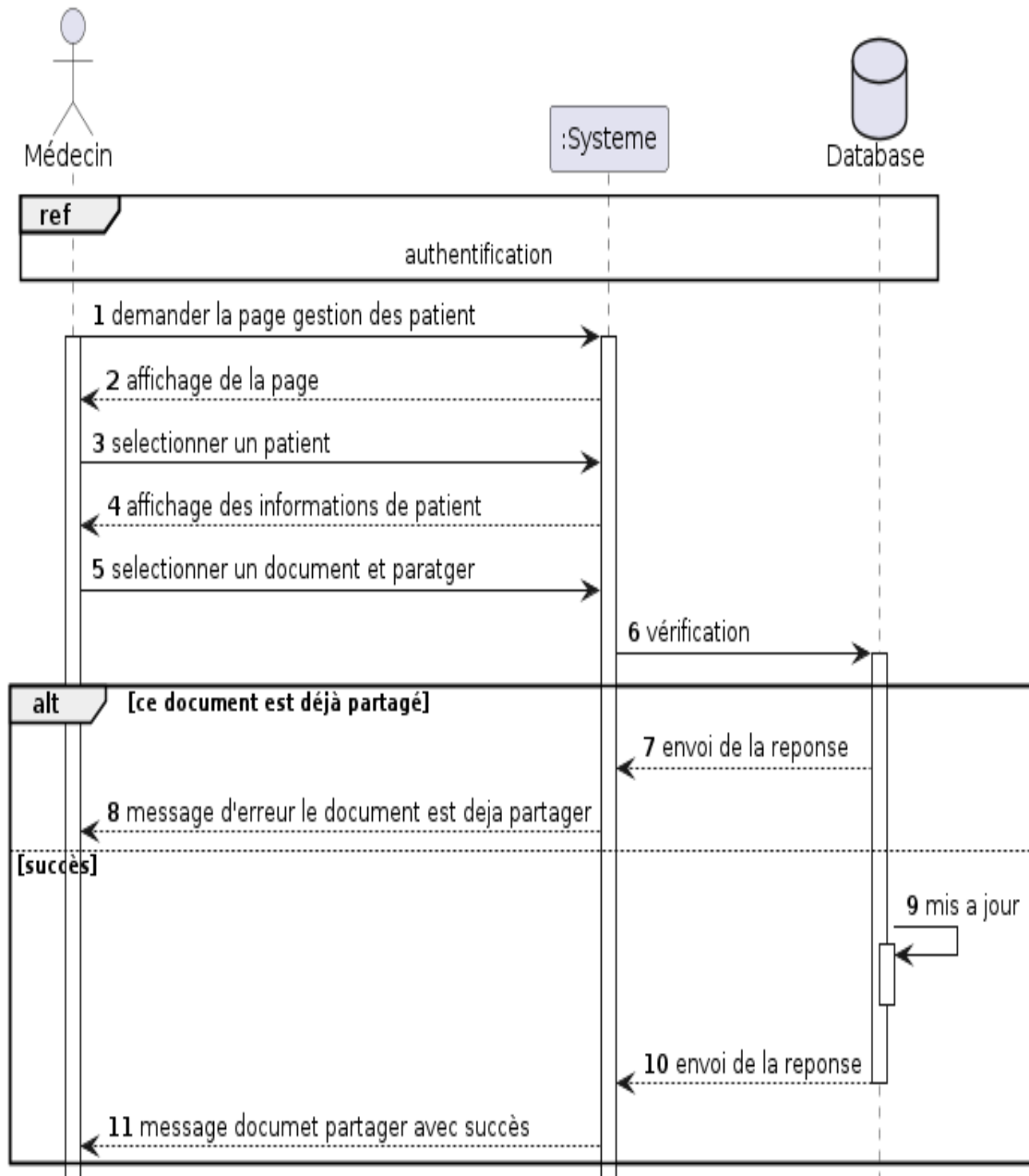


FIGURE 3.17 – Diagramme de séquence « Partager des documents avec un patient ».

3.6.4 Diagramme de séquence « Prendre un rendez-vous »

La Figure 3.18 représente le diagramme de séquence du cas d'utilisation « Prendre un rendez-vous ».

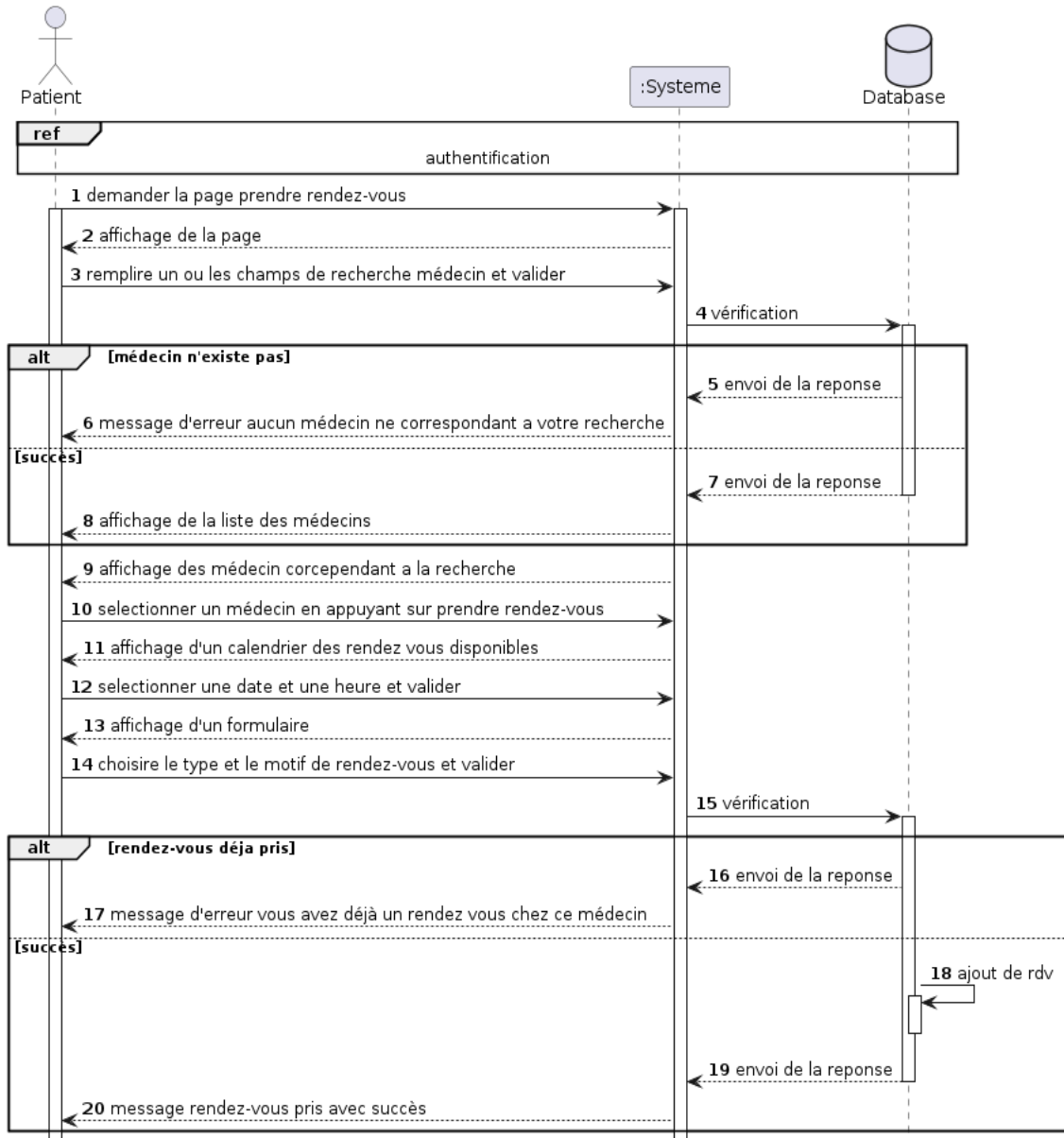


FIGURE 3.18 – Diagramme de séquence « Prendre un rendez-vous ».

3.7 Diagramme de classes

Un diagramme de classes se définit comme étant un ensemble de classes contenant des attributs et des opérations, reliées les unes aux autres par des relations et ceci en ayant des conditions de participation (cardinalités). Il s'agit de la version UML de la base de données.

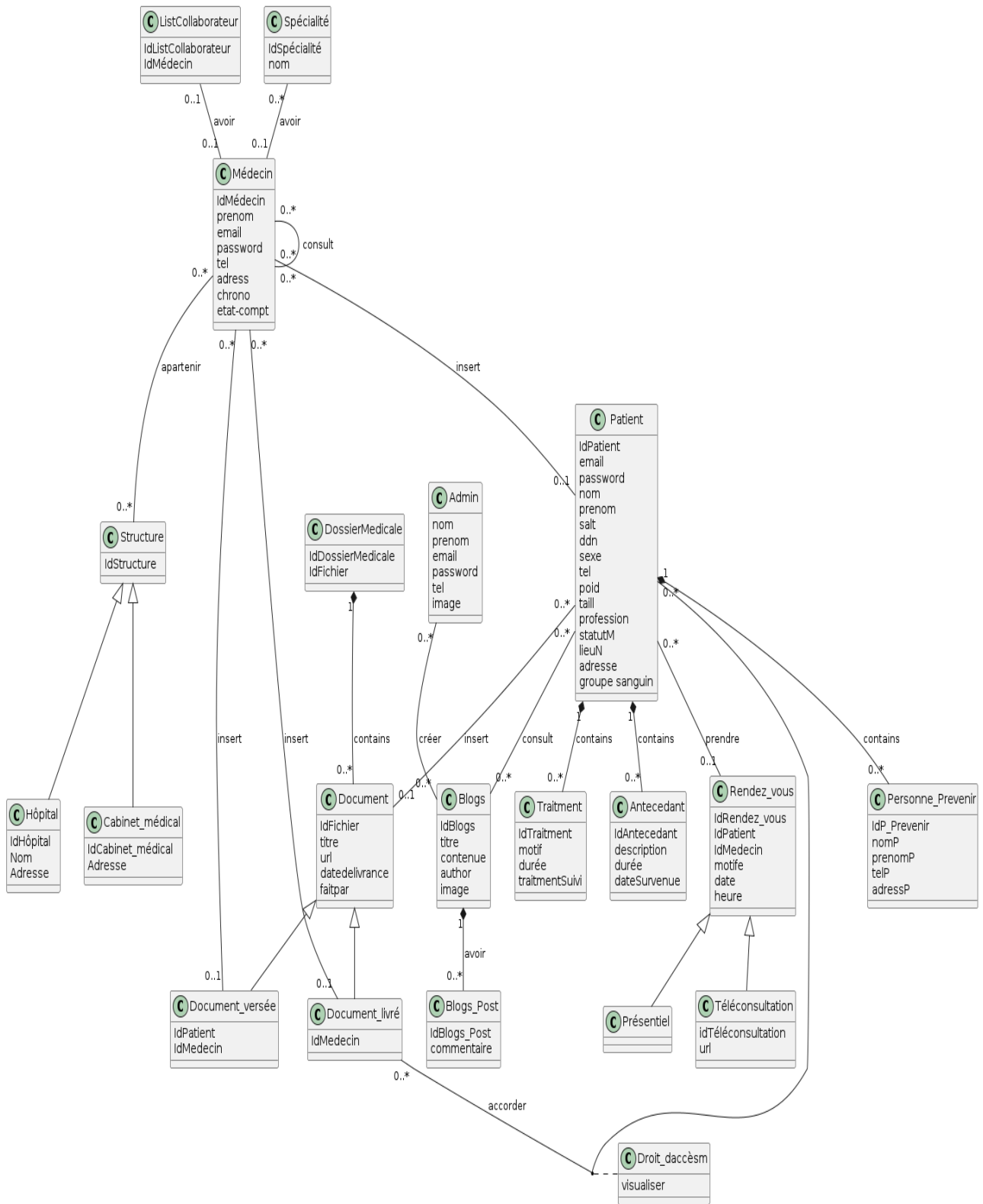


FIGURE 3.19 – Diagramme de classes.

3.8 Dictionnaire de données

La Table 3.2 représente le dictionnaire de données associé à notre diagramme de classes.

Classe	Attributs	Désignation	Type
Patient	idPatient	Identifiant du patient	ObjectId
	nom	Nom du patient	String
	prenom	Prénom du patient	String
	ddn	Date de naissance	date
	sexe	Genre du patient	String
	tel	Numéro de téléphone	String
	poid	Poids	Number
	taill	Taille	Number
	profession	Métier du patient	String
	StatuM	Statut familial	String
	lieuN	Lieu de naissance	String
	adresse	Adresse du patient	String
Médecin	idMédecin	Identifiant du médecin	ObjectId
	nom	Nom du médecin	String
	prenom	Prénom du médecin	String
	email	E-mail du médecin	String
	Password	Mot de passe du médecin	String
	tel	Numéro de téléphone	String
	adress	Adresse du médecin	String
	chrono	Durée de consultation	Number
Admin	idAdmin	Identifiant de l'administrateur	ObjectId
	nom	Nom de l'administrateur	String
	prenom	Prénom de l'administrateur	String
	email	E-mail de l'administrateur	String
	password	Mot de passe de l'administrateur	String
	tel	Numéro de téléphone de l'administrateur	String
	image	Image de profil de l'administrateur	String

Rendez-vous	idRendez-vous	Identifiant du rendez-vous	ObjectId
	date	Date du rendez-vous	Date
	heure	Heure du rendez-vous	Time
Téléconsultation	idtéléconsultation	Identifiant de la téléconsultation	ObjectId
	url	Lien de la téléconsultation	String
Dossier Médicale	idDossierMedicale	Identifiant du dossier médical	ObjectId
	idFichier	Identifiant du fichier médical	array of ObjectId
Document	idFichier	Identifiant du fichier médical	ObjectId
	titre	Type de fichier	String
	url	Lien du fichier médical	String
	datedelivrance	Date de délivrance du fichier	String
	faitpar	Rédacteur du fichier médical	String
Droit-daccés	Visualiser	Consultation du fichier médical	Boolean
Etat-compt	accepted	Compte accepté	Boolean
	rejected	Compte refusé	Boolean
	blocked	Compte bloqué	Boolean
	waiting	Compte mis en attente	Boolean
Blogs	idBlogs	Identifiant du blog	ObjectId
	titre	Titre du blog	String
	contenue	Contenu du blog	String
	author	Auteur du blog	ObjectId
	image	Image du blog	String
Antécédent	idAntecedent	Identifiant de l'antécédent	ObjectId
	description	Description de l'antécédent	String
	durée	Durée de l'antécédent	String
	datesurvenue	Date de survenue de l'antécédent	String
Traitement	idTraitement	Identifiant du traitement	ObjectId
	motif	Motif pour suivre ce traitement	String
	durée	Durée du traitement	String
	traitementSuivi	Nom du traitement suivi	String

Personne-Prevenir	idP-Prevenir	Identifiant de la personne à prévenir	ObjectId
	nomP	Nom de la personne	String
	prenomP	Prénom de la personne	String
	telP	Numéro de téléphone	String
	adressP	Adresse de la personne	String

TABLE 3.2 – Dictionnaire de données.

3.9 MongoDB Schema

La figure 3.9 représente le schéma des collection que nous avons utilisée dans notre base de donnée

Medecin	
_id	ObjectId
Nom	String
Prenom	String
E-mail	String
Mots de passe	String
Téléphone	String
Spécialités	ObjectId
Adresse	String
Confirmé {	Object
Accepté	Boolean
Rejeter	Boolean
En_attend	Boolean
Bloqué }	Boolean
Tarif [{	Array of Objects
Nom	String
Prix}]	Number
Collaborateur [{	Array of Objects
idcollaborateur}]	ObjectId

Spécialités	
_id	ObjectId
Nom	String

Blog	
_id	ObjectId
Titre	String
Contenu	String
Auteur	ObjectId
Figure	String

Patient	
_id	ObjectId
Nom	String
Prenom	String
E-mail	String
Mots de passe	String
Téléphone	String
Salt	String
Ddn	Date
LieuN	String
sexe	String
poids	Number
Taille	Number
StatutM	String
Profession	String
Traitement [{	Array of Objects
Motif	String
Durée	String
TraitementSuivi}]	String
Antecedent[{	Array of Objects
Description	String
Durée	String
DateSurvenue}]	String
Prevenire({	Object
Nom	String
Prenom	String
Téléphone	String
Adresse})	String

Admin	
_id	ObjectId
Noim	String
Prenom	Type
E-mail	Type
Mots de passe	Type
Téléphone	Type
Image	Type

DocumentMedeical	
_id	ObjectId
idPatient	ObjectId
Titre	String
Description	String
URL	String
DateDelivrance	Date
FailPar	String
AccèsMedecin [{	Array of Objects
idMedecin	ObjectId
Droit {	Object
Visualiser }]}	Boolean
AccèsPatient	Boolean

Rendez-vous	
_id	ObjectId
Motif	String
Debut	Datetime
idMedecin	ObjectId
idPatient	ObjectId
Tout_la_journée	Boolean
Type	Boolean
TeleconsultationURL	String

FIGURE 3.20 – MongoDB Schema.

3.10 Conclusion

Dans ce chapitre, nous avons recensé les cas d'utilisation pour chaque acteur, que nous avons modélisés par une représentation graphique UML : les diagrammes de cas d'utilisation complétés par les diagrammes de séquence. Par la suite, en définissant les relations entre les entités, nous sommes parvenus à concevoir le diagramme de classes donnant ainsi une vue plus structurée des éléments qui formeront la base de données liée à notre application. Ainsi, nous pouvons entamer la phase de réalisation de notre application.

Chapitre 4

Réalisation

4.1 Introduction

Ce chapitre est consacré à la partie pratique de la réalisation de notre application web et mobile. Ce dernier comporte une description des outils de développement utilisés ainsi qu'une présentation des deux applications. Le choix de nos outils de développement s'est fondé principalement sur leur nouveauté, gratuité et la liberté d'accès (*open source*). En implémentant une solution qui se base sur des technologies récentes, gratuites et open source, nous pourrions éviter toutes sortes de problèmes liés aux licences, contrats, etc., mais aussi réduire les coûts.

4.2 Environnement et outils de développement

4.2.1 PlantUML

PlantUML est un outil open source permettant aux utilisateurs de créer des diagrammes à partir d'un langage de texte brut. Outre divers diagrammes UML, PlantUML prend en charge divers autres formats liés au développement de logiciels, ainsi que la visualisation des fichiers JSON (JavaScript Object Notation) et YAML (Yet Another Markup Language) [19].

4.2.2 Outil de collaboration GitLab

C'est un outil open-source basé sur Git, permettant de gérer les dépôts de code source, leur versioning, et qui propose tout un tas de fonctionnalités autour des projets de développement. Il peut être utilisé par une seule personne comme par une équipe, voire une organisation ou une grande entreprise, pour gérer des projets de développement. Concurrent direct du connu et reconnu GitHub, GitLab lui est souvent préféré [16]. Parmi les fonctionnalités proposées par GitLab :

- Système de suivi de bugs (ou *issues*) ;
- Une gestion d'équipe (membres et rôles) ;
- Un tableau de bord interactif ;
- Une intégration et livraison continue ;
- Un hébergement de pages web statiques (HTML/CSS/JavaScript).

4.2.3 Visual Studio Code

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git [17].

4.2.4 Android Studio

Android Studio est un environnement de développement d'applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous tous les systèmes d'exploitation [18].

4.2.5 MERN stack

La MERN stack (MongoDB, Express, React, Node) représente l'alliance des technologies les plus puissantes que l'on trouve sur le marché. MERN est l'une des nombreuses variantes de la pile MEAN (MongoDB, Express, Angular, Node), toutes basées sur JavaScript, où le Framework frontal Angular traditionnel est remplacé par React. Il s'agit d'un cadre de développement open source full stack, c'est-à-dire, qu'il fournit des composants de développement front-end et back-end. Cette combinaison va permettre aux développeurs de créer des applications web complètes (back-end et front-end) où on utilise le JavaScript côté client et le Node.js côté serveur [20]. La Figure 4.1 illustre la MERN stack.

4.2.6 Node Package Manager

Aujourd'hui, toutes les bibliothèques et frameworks JavaScript s'installent avec NPM (Node Package Manager). Un package manager est un gestionnaire de dépendances, il permet d'installer toutes les bibliothèques, frameworks et outils dont a besoin un projet JavaScript. C'est l'équivalent de composer en PHP, Maven en Java, et NuGet en C#. L'outil s'appelle Node Package Manager car il a été inventé au départ pour Node.js. Aujourd'hui, NPM est devenu le package manager pour tout l'écosystème JavaScript, y compris hors Node.js. Il suffit donc d'installer Node pour installer NPM [22].

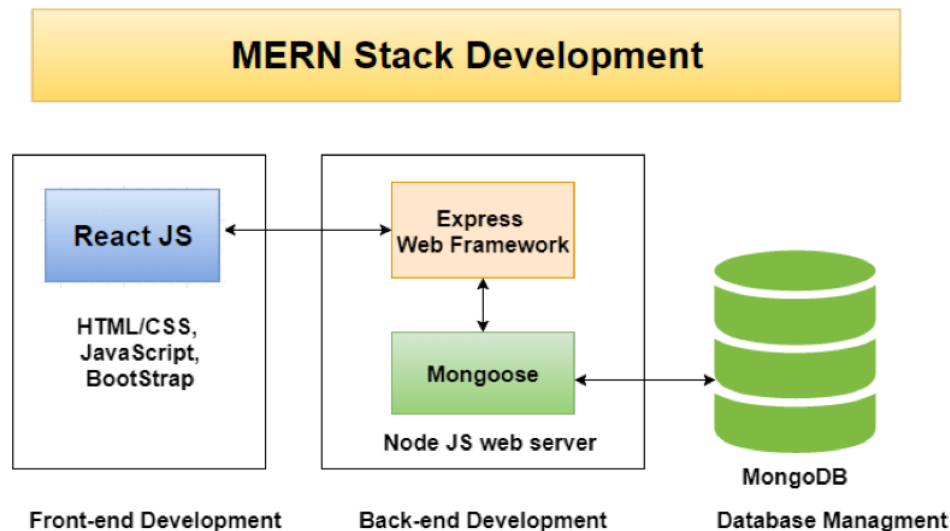


FIGURE 4.1 – Schéma illustrant la MERN stack [21].

4.3 Front-end

4.3.1 React JS/Native

React JS/Native est une bibliothèque basée sur JavaScript développée par Facebook depuis 2013, qui comprend la vitesse de JavaScript et le mécanisme innovant permettant de rendre les sites web/applications mobiles extrêmement rapides et réactifs à l'égard des requêtes des utilisateurs. En effet, cette solution a radicalement transformé la méthodologie de Facebook en développement web/mobile. Depuis que React est apparu comme framework open source, deux ans après sa première version, sa diffusion parmi les clients était incroyablement dynamique grâce à son extraordinaire mode de codage de l'interface utilisateur [23].

4.3.2 Material UI

Material UI est un framework React UI qui adhère aux concepts de Material design. Il est construit sur le framework React de Facebook et comprend des composants qui adhèrent aux normes Material [25].

4.3.3 Chakra UI

Chakra UI est une bibliothèque de composants simple, modulaire et accessible qui donne les éléments de base dont a besoin un développeur pour créer des applications React [26].

4.3.4 Redux

Redux est une bibliothèque open-source JavaScript de gestion d'état pour applications web. Elle est plus couramment utilisée avec des bibliothèques comme React ou Angular pour la construction d'interfaces utilisateur [27].

4.3.5 Axios

Axios est une bibliothèque JavaScript fonctionnant comme un client HTTP (Hypertext Transfer Protocol). Elle permet de communiquer avec des API (Application Programming Interface) en utilisant des requêtes [28].

4.3.6 MomentJS

MomentJS est une bibliothèque JavaScript qui aide à analyser, valider, manipuler et afficher la date/heure en JavaScript de manière très simple. Nous l'avons utilisée car la date récupérée par les inputs est en format ISO (International Organization for Standardization), et pour la lisibilité de la date dans sa forme DD/MM/YYYY [29].

4.3.7 React Icons

React Icons est une bibliothèque qui rassemble toutes les icônes des autres packages de React, ce qui permet de ne pas choisir une bibliothèque précise [29].

4.3.8 React-file-base64

Base64 est un groupe de schémas pour encoder des données binaires sous forme d'un texte au format ASCII grâce à la représentation de ces données en base 64. Le terme base64 vient à l'origine de l'encodage. Nous l'avons utilisé pour sauvegarder les fichiers sans avoir un dossier pour enregistrer ces derniers [24].

Nous avons utilisé d'autres packages et bibliothèques, la Figure 4.2 illustre le fichier *package.json* présent à la racine de notre projet où tous les frameworks, packages et bibliothèques sont présents.

```
{ } package.json > { } dependencies
1  {
2    "name": "utilisateur",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@chakra-ui/react": "^1.8.6",
7      "@emotion/react": "11",
8      "@emotion/styled": "11",
9      "@testing-library/jest-dom": "^5.14.1",
10     "@testing-library/react": "^12.0.0",
11     "@testing-library/user-event": "^13.2.1",
12     "axios": "^0.26.1",
13     "formik": "^2.2.9",
14     "framer-motion": "6",
15     "i18next": "^21.6.14",
16     "i18next-browser-languagedetector": "^6.1.3",
17     "i18next-http-backend": "^1.4.0",
18     "moment": "^2.29.1",
19     "react": "^17.0.2",
20     "react-alice-carousel": "^2.6.0",
21     "react-dom": "^17.0.2",
22     "react-i18next": "^11.15.7",
23     "react-icons": "^4.3.1",
24     "react-query": "^3.34.19",
25     "react-router-dom": "^6.2.2",
26     "react-scripts": "5.0.0",
27     "react-spinners": "^0.11.0",
28     "web-vitals": "^2.1.0",
29     "yup": "^0.32.11"
30   },
```

FIGURE 4.2 – Fichier package.json de notre projet.

4.4 Back-end

4.4.1 Architecture MVC

L'architecture MVC (Modèle – Vue – Contrôleur) est un patron utilisé en PHP pour le développement d'applications web. Cette architecture a pour but de séparer les problématiques liées aux différents composants au sein de leur architecture respective. MVC permet de concevoir des applications web de manière claire et efficace grâce à la séparation des intentions. Les opérations de maintenance et de mises à jour sont fortement simplifiées [30] :

- Le modèle permet d'accéder aux données du site web que ce soit dans une base de données à l'aide de classes ou dans des fichiers plats de type XML ;
- La vue traite l'affichage à l'écran dans le navigateur web bien souvent retranscrit grâce à l'HTML et le CSS ;
- Le contrôleur gère les interactions. D'une manière générale, il utilise les données du modèle, les traite en fonction de l'action de l'utilisateur, et les envoie à la vue afin qu'elle les affiche.

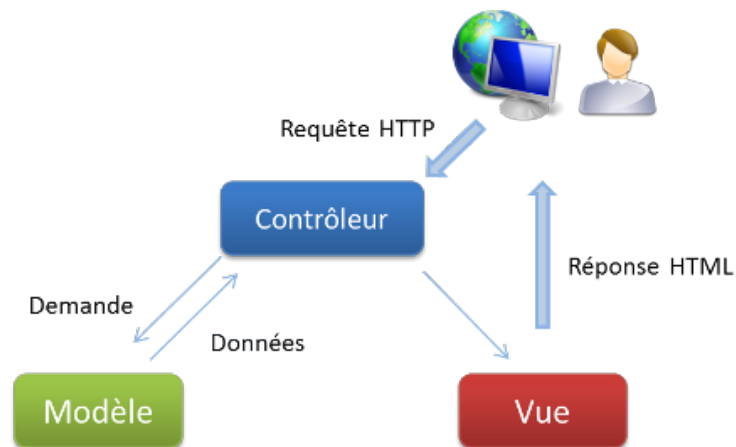


FIGURE 4.3 – Architecture MVC [30].

4.4.2 Node.js

Node.js n'est pas un langage de programmation. Ce n'est pas non plus un framework JavaScript. Il s'agit d'un environnement d'exécution JavaScript. C'est une plateforme logicielle avec une architecture orientée événements qui permettent d'utiliser le langage de script JavaScript [31].

4.4.3 ExpressJS

ExpressJS est une infrastructure d'applications web Node.js minimale et flexible, fournissant un ensemble robuste de fonctionnalités permettant de développer des applications web et mobiles. Il facilite le développement rapide d'applications web basées sur des nœuds [31].

4.4.4 NodeMailer

NodeMailer est un module pour Node.js permettant l'envoi d'e-mails en toute simplicité. Nous l'avons utilisé pour les cas de confirmation des e-mails et d'envoi de rappels.

4.5 Implémentation de la base de données

4.5.1 NoSQL

Le terme NoSQL (Not Only SQL) fait référence à des types de bases de données non relationnelles, qui stockent les données dans un format différent des tables relationnelles. Cependant, les bases de données NoSQL peuvent être interrogées à l'aide d'API de langage idiomatique et de langages de requête structurés déclaratifs [32].

4.5.2 MongoDB

C'est une base de données open source, centrée sur les documents. MongoDB a été mis sur le marché en l'espace de deux ans seulement et il aura fallu peu de temps pour que celui-ci devienne une des bases de données NoSQL les plus populaires [31].

4.5.3 Mongoose

Mongoose est une bibliothèque ODM (Object Data Modeling) pour MongoDB et Node.js. Elle fournit une solution simple basée sur un schéma pour modéliser les données d'application. Il comprend le casting de type intégré, la validation, la création de requêtes et plus encore [33].

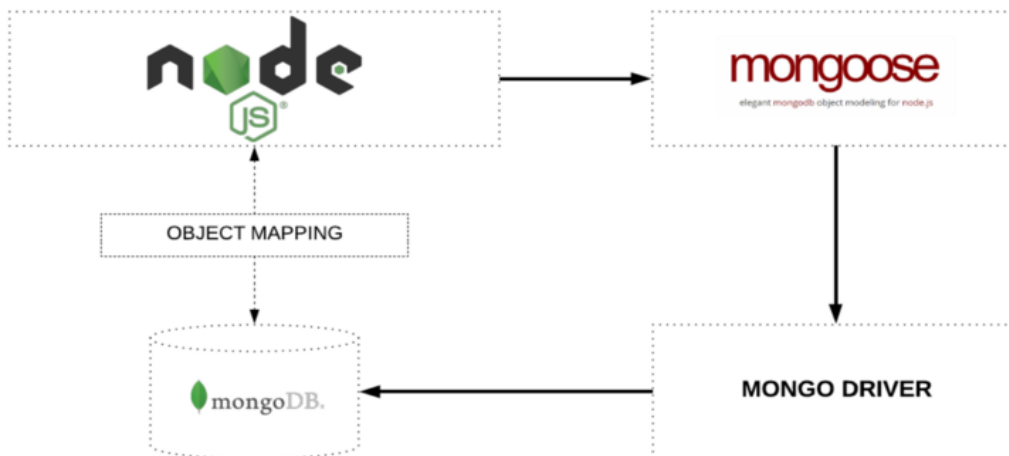


FIGURE 4.4 – Mappage d'objets entre Node et MongoDB géré via Mongoose [33].

4.5.4 Choix de la pile MERN

MongoDB, la base de données de documents à la racine de la pile MERN a été conçue pour stocker les données JSON de manière native (elle utilise techniquement une version binaire de JSON appelée BSON), et tout, de son interface de ligne de commande à son langage de requête (MQL - Mongo DB Query Language), sont construits sur JSON et JavaScript.

MongoDB fonctionne extrêmement bien avec Node.js, et facilite le stockage, la manipulation et la représentation des données JSON à chaque niveau de l'application. Pour les applications Cloud-natives, MongoDB Atlas facilite encore plus la tâche en offrant un cluster MongoDB mis à l'échelle automatique sur le fournisseur Cloud choisi, en quelques clics.

Express.js et React.js rendent l'application JavaScript/JSON MERN complète. La combinaison signifie que les données JSON circulent naturellement de l'avant vers l'arrière,

ce qui les rend rapides à construire et raisonnablement simples à déboguer. De plus, nous n'avons besoin que d'un seul langage de programmation et de la structure du document JSON pour comprendre l'ensemble du système [20].

4.6 Implémentation de la téléconsultation

4.6.1 Agora

Agora est une plateforme de diffusion vidéo, vocale et interactive en direct de premier plan, qui aide les développeurs à offrir des expériences intégrées riches, y compris le chat vocal et vidéo intégré, l'enregistrement en temps réel, la diffusion interactive en direct et la messagerie en temps réel. Avec plus de 200 centres de données dans le monde, le réseau en temps réel défini par logiciel (SD-RTN™) d'Agora est fiable et largement utilisé par les plus grandes marques du monde et offre une facilité d'évolution pour jusqu'à 1 million d'utilisateurs simultanés (PCU) de pointe dans une seule session avec une qualité d'expérience inégalée. Agora prend entièrement en charge une gamme d'environnements de développement, ce qui facilite l'intégration profonde d'appels vidéo de haute qualité et à très faible latence sur toutes les plateformes et tous les canaux [34].

4.6.2 Web Real-Time Communication

WebRTC, littéralement communication en temps réel pour le Web, est une interface de programmation JavaScript développée au sein du W3C et de l'IETF. C'est aussi un canevas logiciel avec des implémentations précoces dans différents navigateurs web pour permettre une communication en temps réel [35]. WebRTC permet notamment de réaliser des conférences audios ou vidéos directement depuis son navigateur, sans télécharger un plugin ou une extension particulière.

1. Le client A veut communiquer avec le client B (le client A ne connaît pas les informations de ce dernier).
2. Il envoie au serveur ses informations ainsi que l'identité du client avec qui il veut communiquer.
3. Le serveur avec ses voisins cherche le client correspondant aux informations envoyées.
4. Lorsque toutes les informations sont envoyées sur les deux navigateurs un lien est créé entre les deux.
5. Les deux clients peuvent communiquer à travers le navigateur sans avoir besoin du serveur.

La Figure 4.5 résume la manière dans cette communication est effectuée.

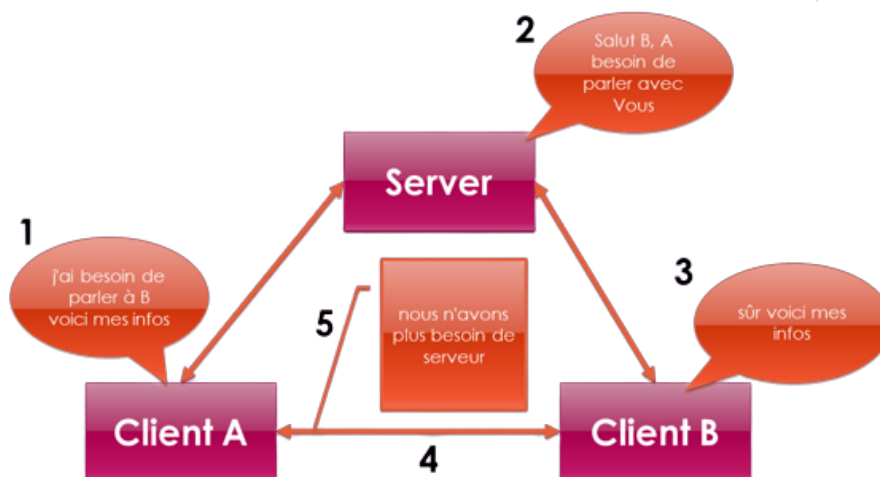


FIGURE 4.5 – Architecture de WebRTC.

4.7 Outils de déploiement

Pour le déploiement de notre application, nous avons utilisé Heroku pour le back-end et Netlify pour le front-end.

4.7.1 Heroku

Heroku est une plateforme cloud en tant que service (PaaS - Platform as a Service) prenant en charge plusieurs langages de programmation. Étant l'une des premières plateformes cloud, Heroku est en développement depuis Juin 2007. Elle ne prenait en charge que le langage de programmation Ruby, mais prend désormais en charge Java, Node.js, Scala, Clojure, Python, PHP et Go [37].

4.7.2 Netlify

Netlify était à la base principalement une solution dédiée aux sites et applications web statiques, celle-ci est désormais connectée à un écosystème de services web modernes (Heroku) permettant de créer des sites et applications dynamiques, ou même des API personnalisées [36].

4.8 Sitemaps

Notre projet est composé d'une application web, divisée en trois applications, et d'une application mobile. Nous présentons ici les navigations possibles entre les interfaces.

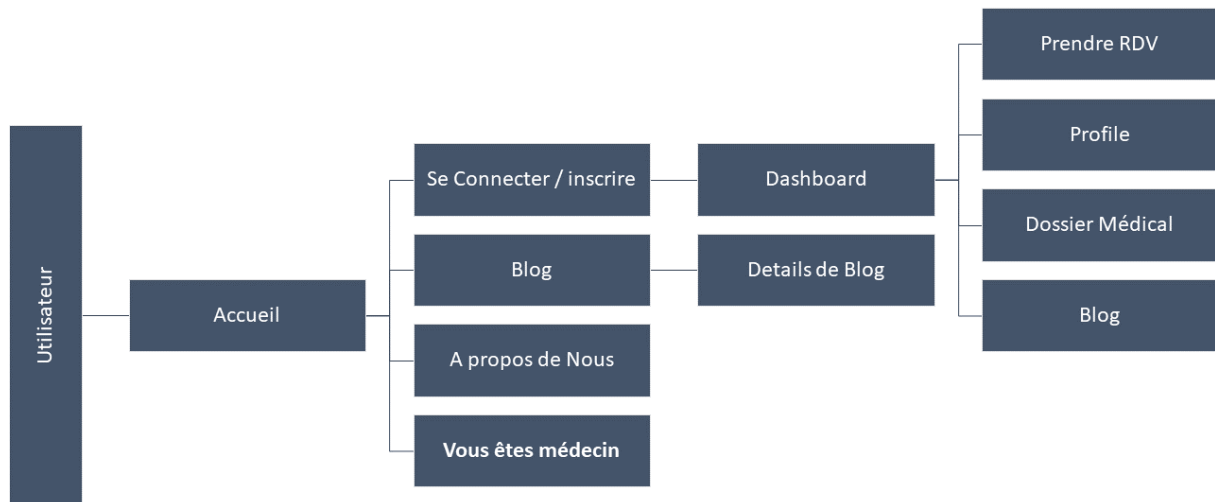


FIGURE 4.6 – Map de l'application web - Patient.

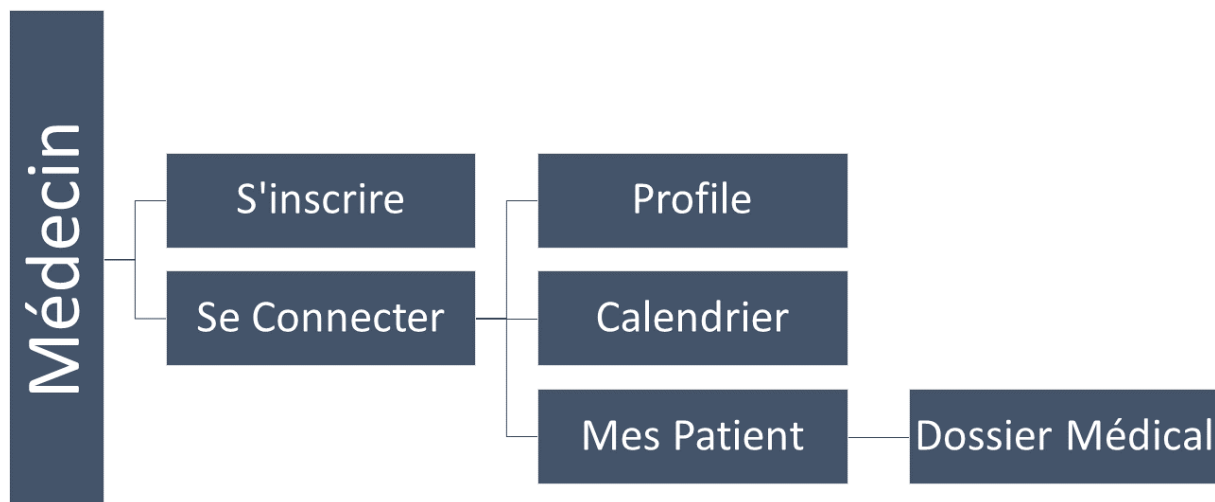


FIGURE 4.7 – Map de l'application web - Médecin.

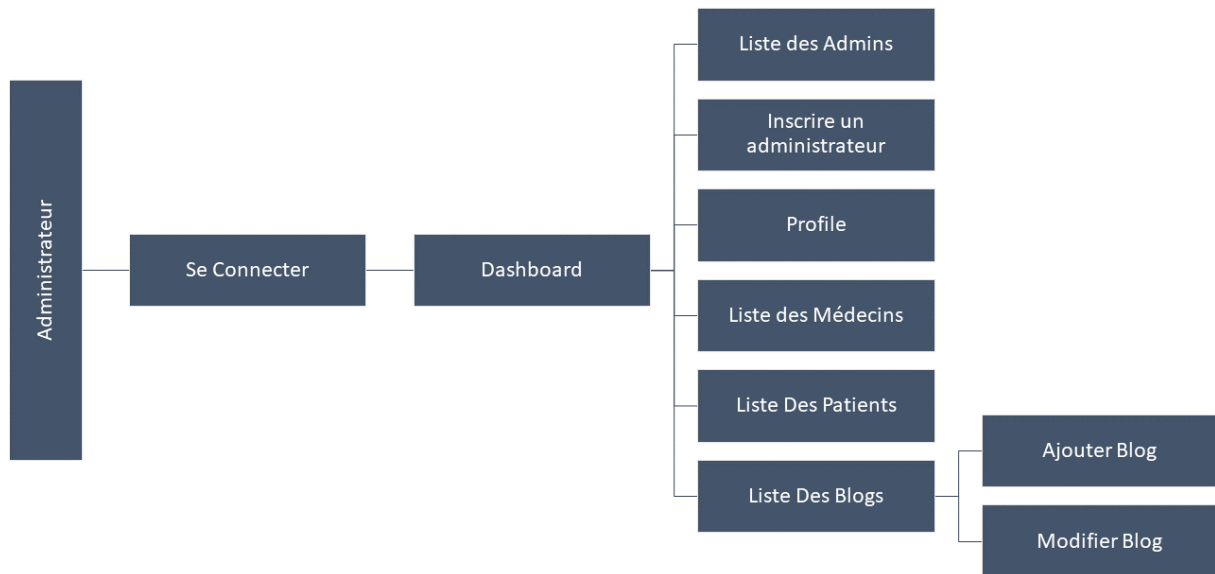


FIGURE 4.8 – Map de l’application web - Administrateur.

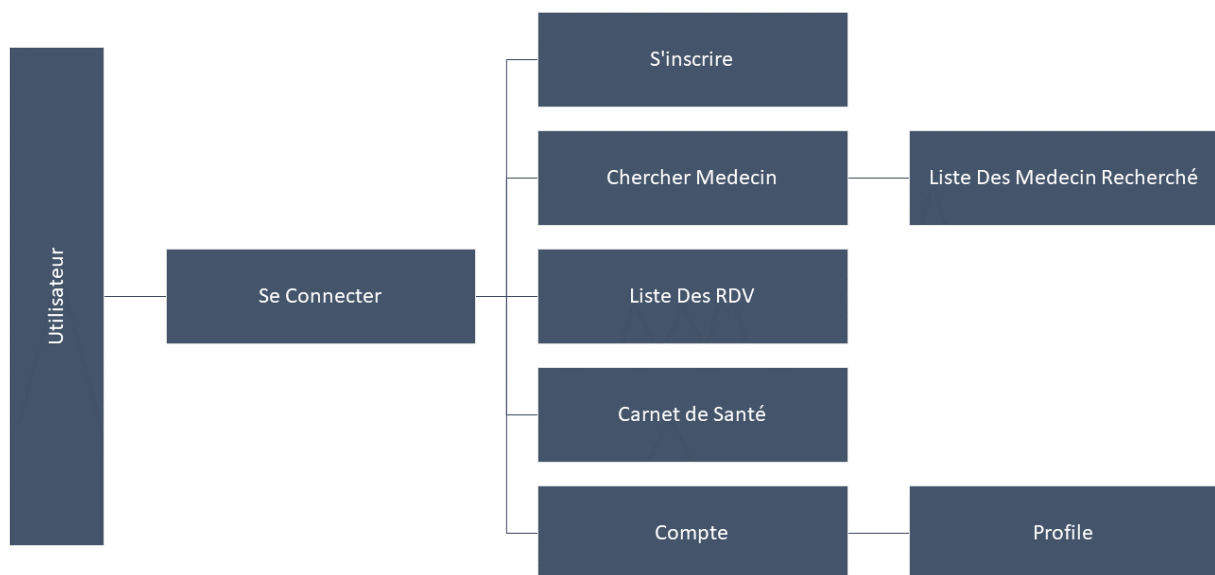


FIGURE 4.9 – Map de l’application mobile - Patient.

4.9 Principales interfaces

4.9.1 Interfaces de l'application web

La Figure 4.10 représente l'interface dashboard de l'administrateur.



FIGURE 4.10 – Interface dashboard de l'administrateur.

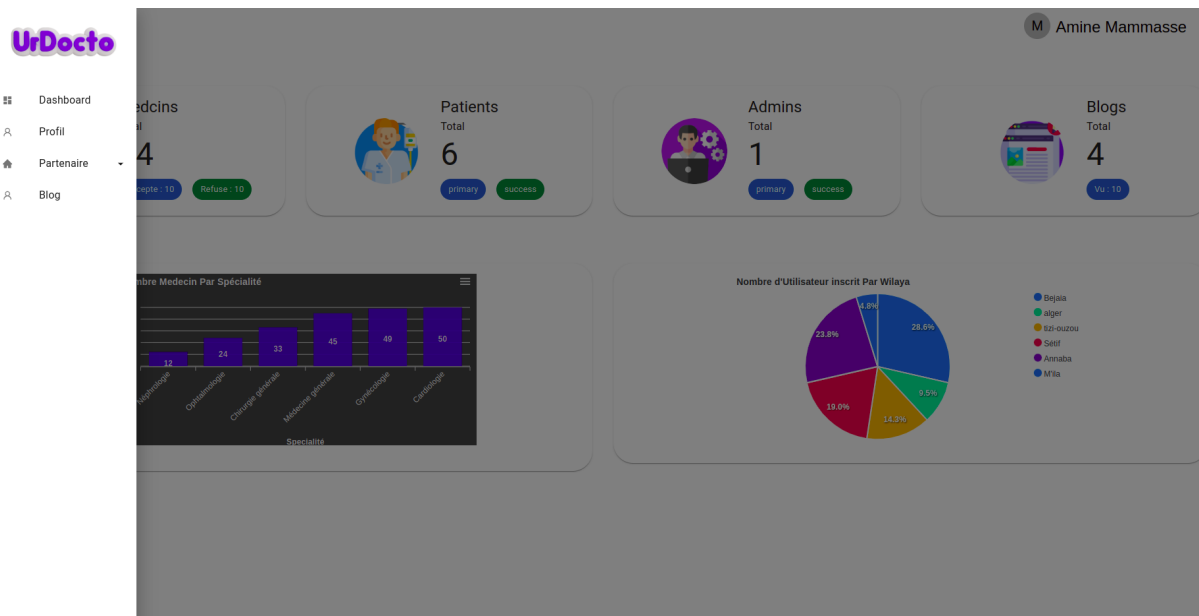


FIGURE 4.11 – Interface administrateur sidebar.

La Figure 4.12 représente l'interface gestion des blogs où l'administrateur pourra créer des blogs, les modifier, et les supprimer.

The screenshot shows the 'Gestion des Blogs' interface. At the top right, there is a user profile 'M Amine Mammasse'. Below the header, there is a table with the following data:

Titre	Editeur	Date	SUPPRIMER	MODIFIER
CYCLE DU SOMMEIL : LES DIFFERENTES PHASES À CONNAÎTRE	Mammasse Amine	2022-06-07T22:09:20.024Z	[SUPPRIMER]	[MODIFIER]
CYCLE DU SOMMEIL : LES DIFFERENTES PHASES À CONNAÎTRE	Mammasse Amine	2022-06-07T22:09:21.767Z	[SUPPRIMER]	[MODIFIER]
CYCLE DU SOMMEIL : LES DIFFERENTES PHASES À CONNAÎTRE	Mammasse Amine	2022-06-07T22:09:22.329Z	[SUPPRIMER]	[MODIFIER]
Test N°55	Mammasse Amine	2022-06-13T19:38:43.718Z	[SUPPRIMER]	[MODIFIER]

At the bottom of the table, there are navigation buttons 'PRÉCÉDENT' and 'SUIVANTE', and a 'page 1 sur 1' indicator. A 'AJOUTER UN BLOG' button is located at the top right of the table area.

FIGURE 4.12 – Interface administrateur gestion des blogs.

La Figure 4.13 représente l'interface gestion des médecins où l'administrateur pourra changer l'état du compte d'un médecin.

The screenshot shows the 'Gestion des medecins' interface. At the top right, there is a user profile 'M Amine Mammasse'. Below the header, there is a table with the following data:

Nom	Prénom	E-mail	Numéro de Téléphone	Spécialité	Confirmé	Voir Plus
Mohamed	Belhocine	belhocine.mohamed@gmail.com	0525060035	Néphrologie	Accepté	[🔍]
Daoud	Kamel	daoud.kamel@gmail.com	0513521001	Ophthalmologie	Accepté	[🔍]
Hasbellaoui	Mokhtar	hasbellaoui.mokhtar@gmail.com	0623224451	Chirurgie générale	Accepté	[🔍]
Mustapha	Lailam	mustaphalal@gmail.com	0520323003	Médecine générale	Accepté	[🔍]
Zerhouni	Elias	zerhounills@gmail.com	0536634117	Gynécologie	Bloqué	[🔍]
Saliha	Adwan	adwan.saliha@gmail.com	0566887710	Cardiologie	En Attend	[🔍]
Layla	Kadri	kadriya@gmail.com	0782399382	Cardiologie	En Attend	[🔍]
Aamer	Mouna	aamer.mouna@gmail.com	0697462241	Gynécologie	En Attend	[🔍]
Nassim	Baghdadi	baghdadi.nassim@gmail.com	0562154899	Médecine générale	En Attend	[🔍]
Srour	Zahir	zahirmedecin@gmail.com	0534201215	Cardiologie	Refusé	[🔍]

At the bottom of the table, there are navigation buttons 'PRÉCÉDENT' and 'SUIVANTE', and a 'page 1 sur 2' indicator. A 'MASQUER LES FILTRES TELECHARGER EN XLS' button is located at the top left of the table area. Search filters for 'Médecin', 'Spécialité', and 'Region' are located at the top of the table area.

FIGURE 4.13 – Interface administrateur gestion des médecins.

La Figure 4.14 représente l'interface d'accueil de notre application UrDocto.



FIGURE 4.14 – Interface d'accueil.

La Figure 4.15 représente l'interface d'inscription du patient.

Se connecter | Inscription

Créer un compte
Vous avez déjà un compte?

Nom Prénom

Date de naissance

Numero de telephone

Email

Mot de passe

Confirmation de mot de passe

Inscription

Disponible sur

App store | Play store

Consultez la disponibilité de votre médecin et réservez le rendez-vous qui vous convient le plus.

FIGURE 4.15 – Interface d'inscription du patient.

La Figure 4.16 représente l'interface de connexion du patient, avec un lien en cas de mot de passe oublié.

Se connecter | Inscription

Se connecter a votre compte
Inscrivez vous si vous n'avez pas de compte?

Email

Mot de passe

Mot de passe oublié?

Connexion

Disponible sur

App store | Play store

Consultez la disponibilité de votre médecin et réservez le rendez-vous qui vous convient le plus.

FIGURE 4.16 – Interface de connexion du patient.

La Figure 4.17 représente l'interface dashboard du patient, où il retrouvera la liste des rendez-vous actuels et la liste de ses médecins favoris.

FIGURE 4.17 – Interface dashboard du patient.

La Figure 4.18 représente l'interface de recherche d'un médecin par spécialité, wilaya ou nom. C'est la première étape de la prise de rendez-vous, le patient peut aussi consulter les informations d'un médecin ou l'ajouter à sa liste de favoris.

FIGURE 4.18 – Interface de recherche d'un médecin.

La Figure 4.19 représente l'interface de la prise de rendez-vous qui consiste à sélectionner une date et une heure, et à renseigner le motif et le type de la consultation.

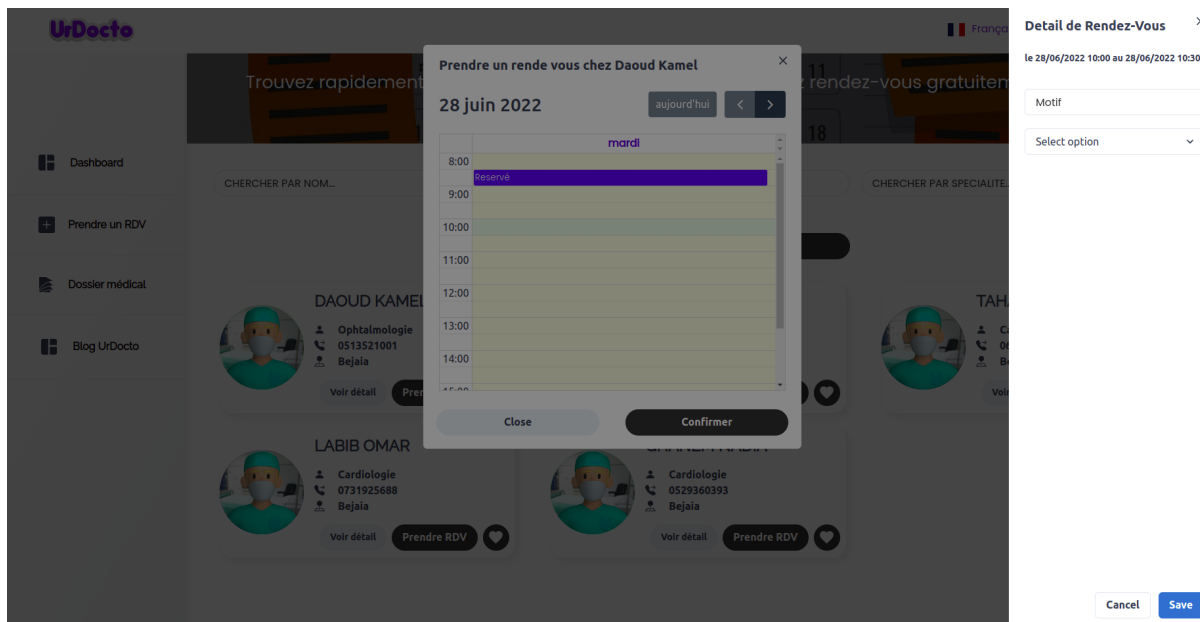


FIGURE 4.19 – Interface de prise de rendez-vous.

La Figure 4.20 représente l'interface de gestion du dossier médical d'un patient, où il pourra ajouter, modifier, supprimer, consulter et partager des documents.

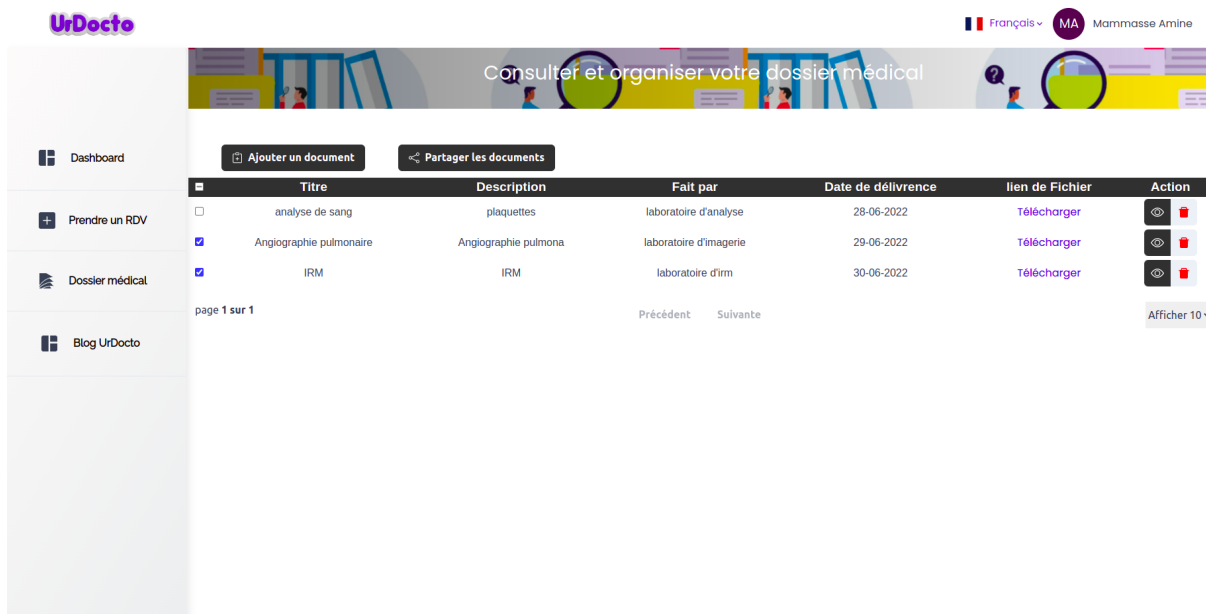


FIGURE 4.20 – Interface de gestion du dossier médical.

La Figure 4.21 représente l'interface de profil du patient, où il pourra modifier ses informations personnelles, ajouter, modifier ou supprimer les informations : d'une personne à contacter en cas d'urgence, d'un antécédent ou d'un traitement.

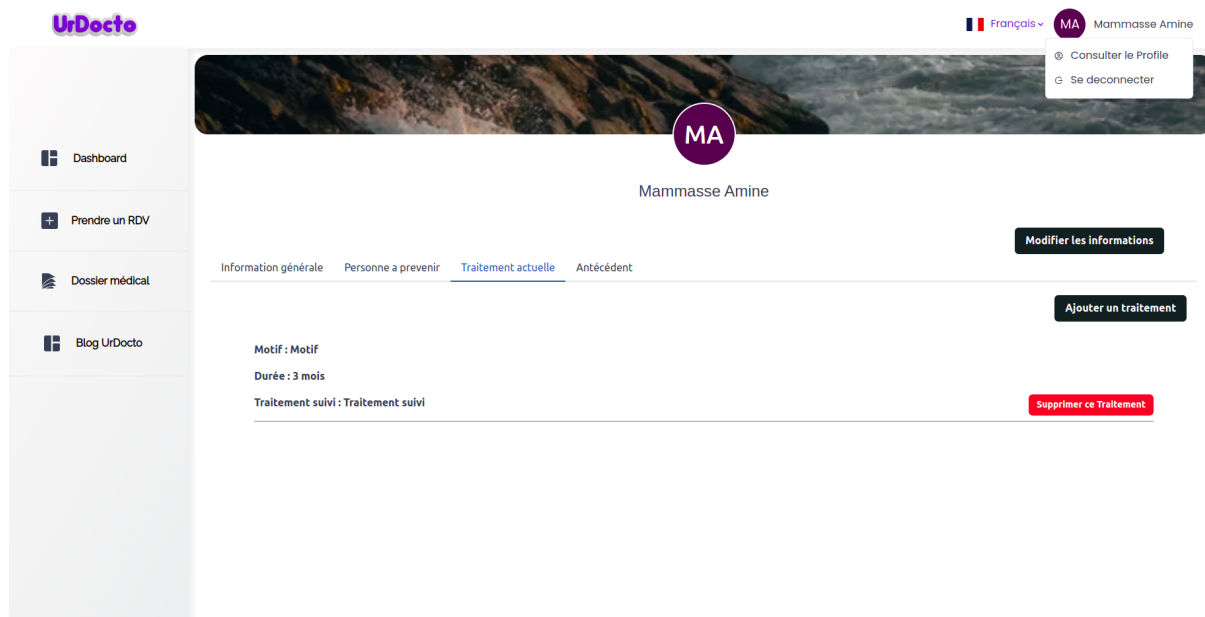


FIGURE 4.21 – Interface de profil du patient.

La Figure 4.22 représente le blog de notre application web.

UrDocto Blog

08 juin 2022

CYCLE DU SOMMEIL : LES DIFFÉRENTES PHASES À CONNAÎTRE



[Lire La Suite](#)



08 juin 2022

CYCLE DU SOMMEIL : LES DIFFÉRENTES PHASES À CONNAÎTRE

Qui n'a jamais rêvé de dormir comme un bébé ? Si ce dicton semble tout droit sorti d'un conte pour enfants, il est loin d'être le quotidien pour 70 % de Français qui souffrent de troubles du sommeil. Considérés comme un enjeu majeur de santé publique, plusieurs...

[Lire La Suite](#)

Overtime:

The Dribbble Podcast

The Power of play

All Categories

Teleconsultation

E-sante

Telemedecine

Votre pratique medicale

Actus secteur sante

Tout voir

FIGURE 4.22 – Interface du blog.

La Figure 4.23 représente l'interface de connexion du médecin.

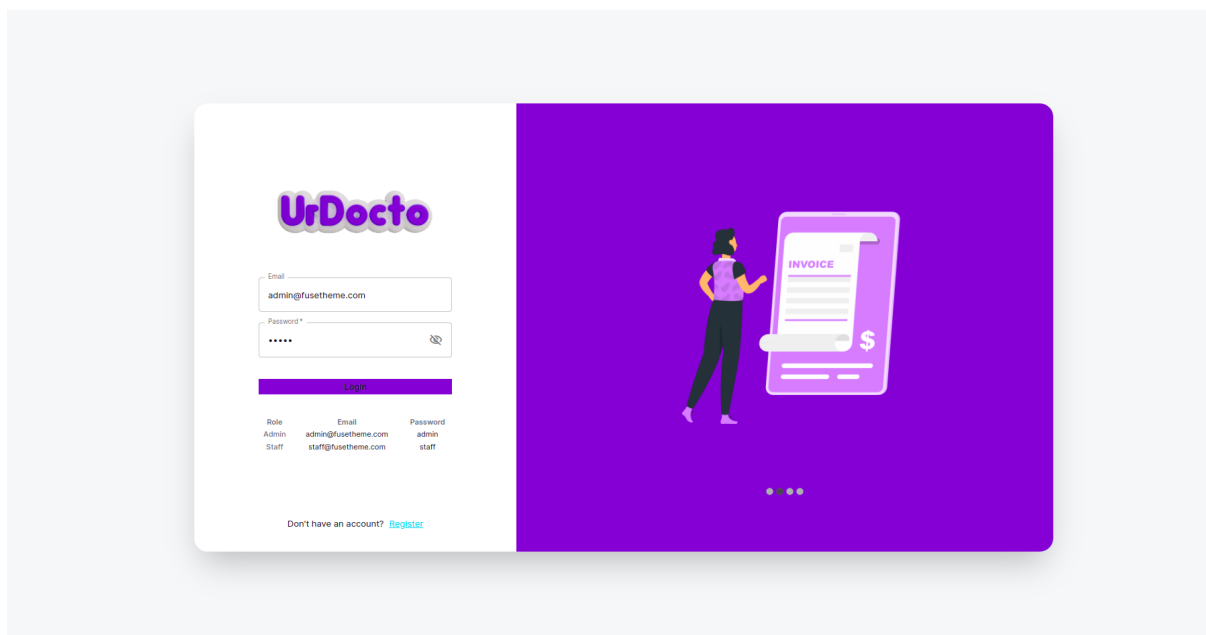


FIGURE 4.23 – Interface de connexion du médecin.

La Figure 4.24 représente l'interface où le médecin pourra consulter la liste de ses rendez-vous.

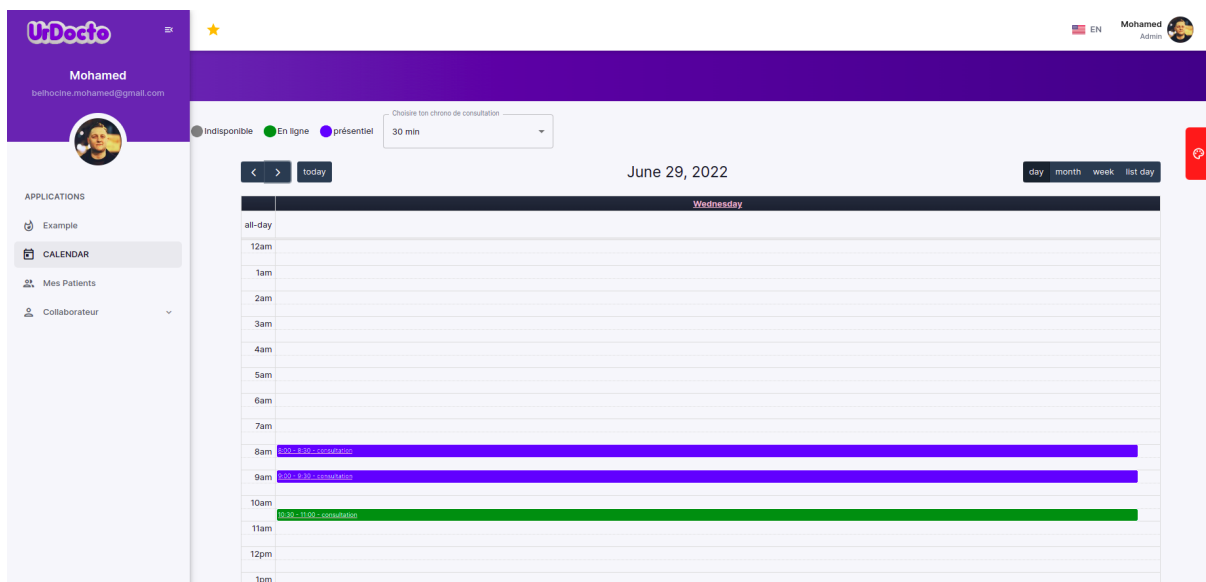


FIGURE 4.24 – Interface de l'agenda des rendez-vous du médecin.

La Figure 4.25 représente l'interface où le médecin pourra consulter les détails concernant un rendez-vous, et confirmer l'ajout d'un patient à sa liste.

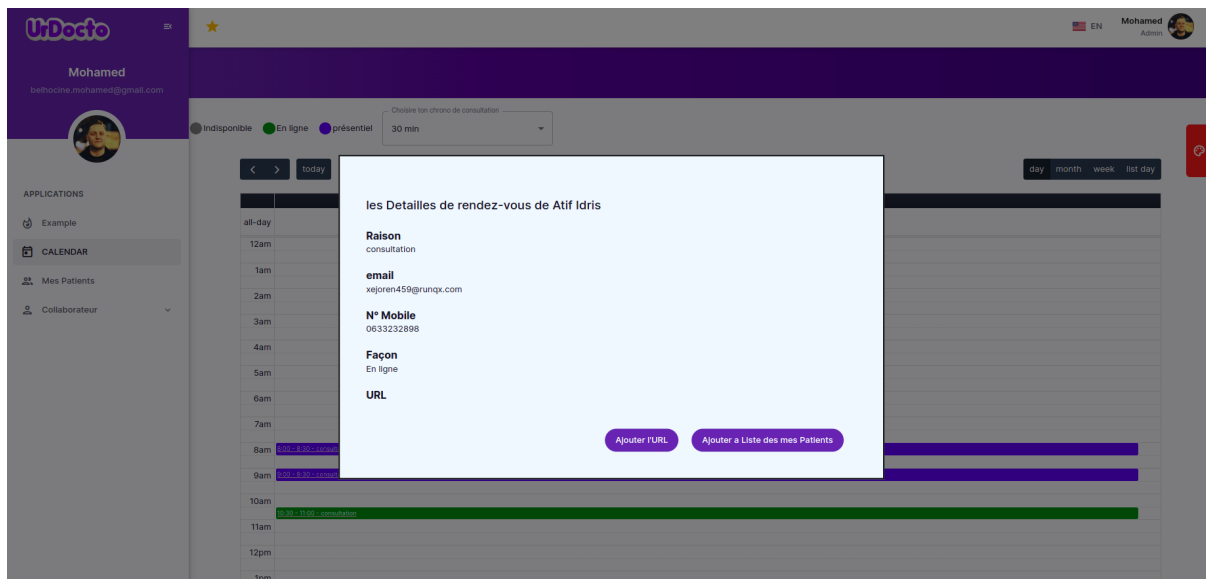


FIGURE 4.25 – Interface détails d'un rendez-vous.

La Figure 4.26 représente l'interface de génération d'un lien de téléconsultation.

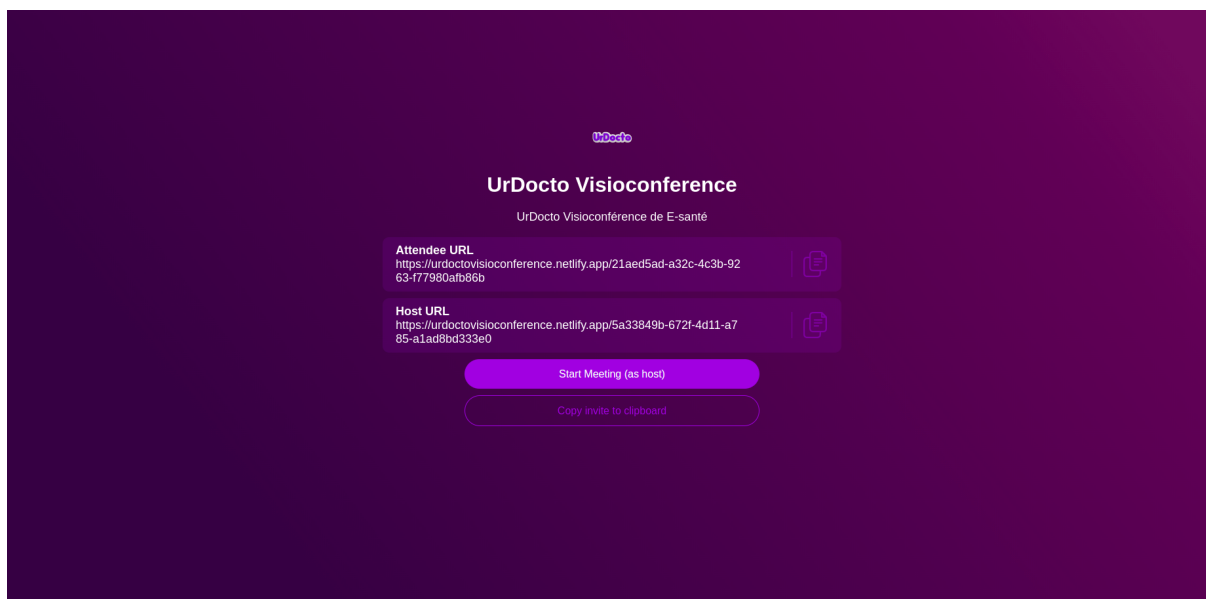


FIGURE 4.26 – Interface de génération d'un lien de téléconsultation.

La Figure 4.27 représente l'interface lorsque le médecin et le patient participent à une téléconsultation.

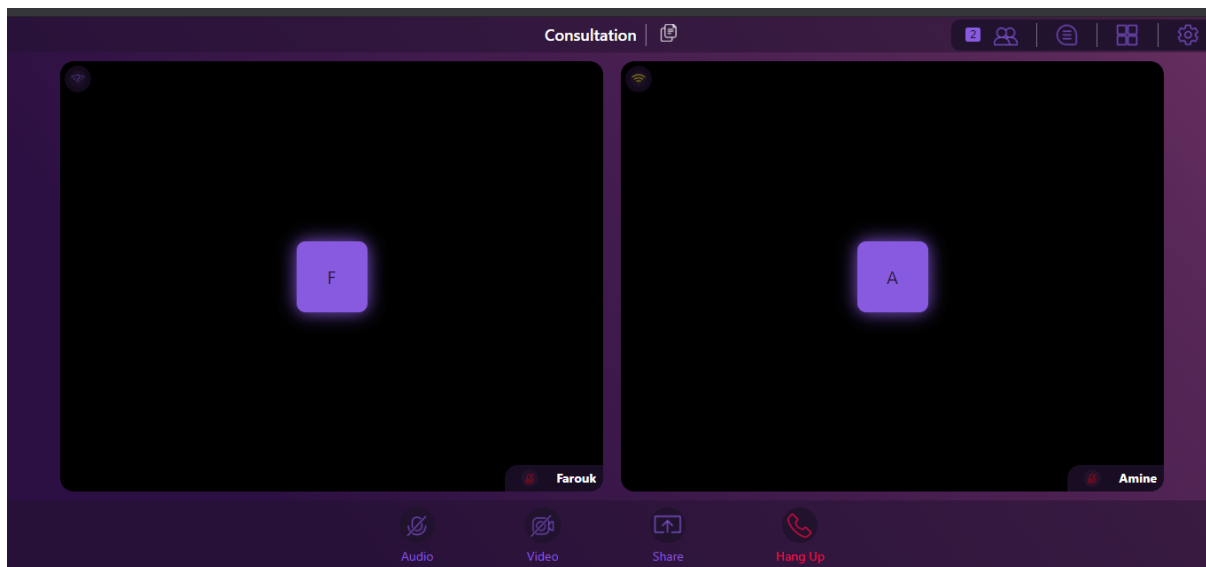


FIGURE 4.27 – Interface de téléconsultation.

La Figure 4.28 représente l'interface de gestion des patients, où le médecin pourra consulter, ajouter, supprimer et rechercher un patient.

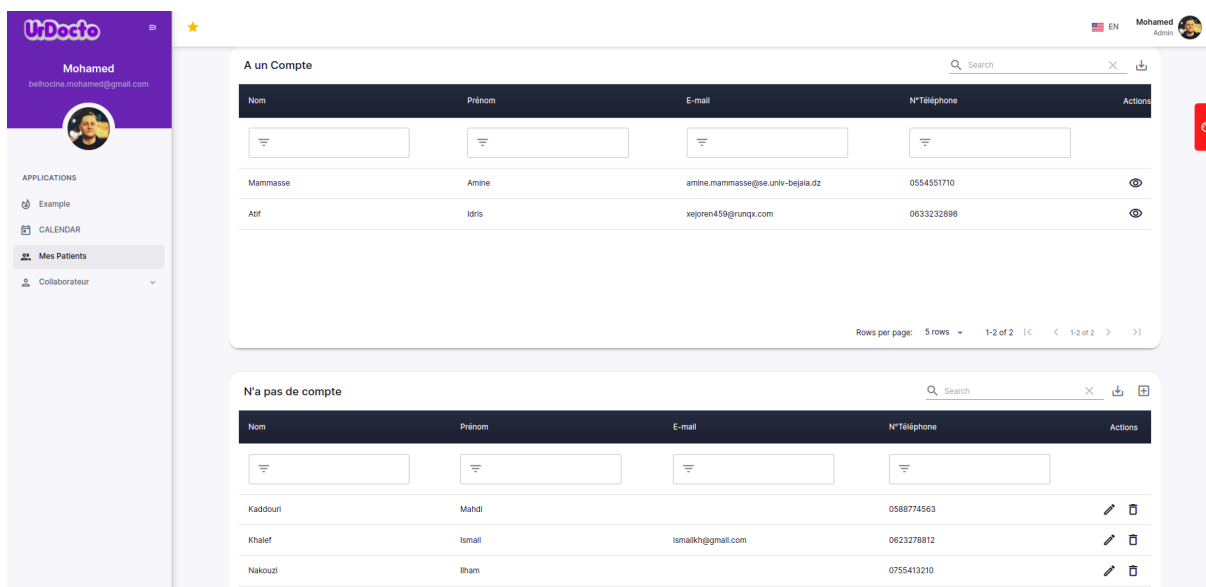


FIGURE 4.28 – Interface de gestion des patients.

La Figure 4.29 représente l'interface de gestion du dossier médical d'un patient, où le médecin pourra consulter, partager, ou ajouter un document.

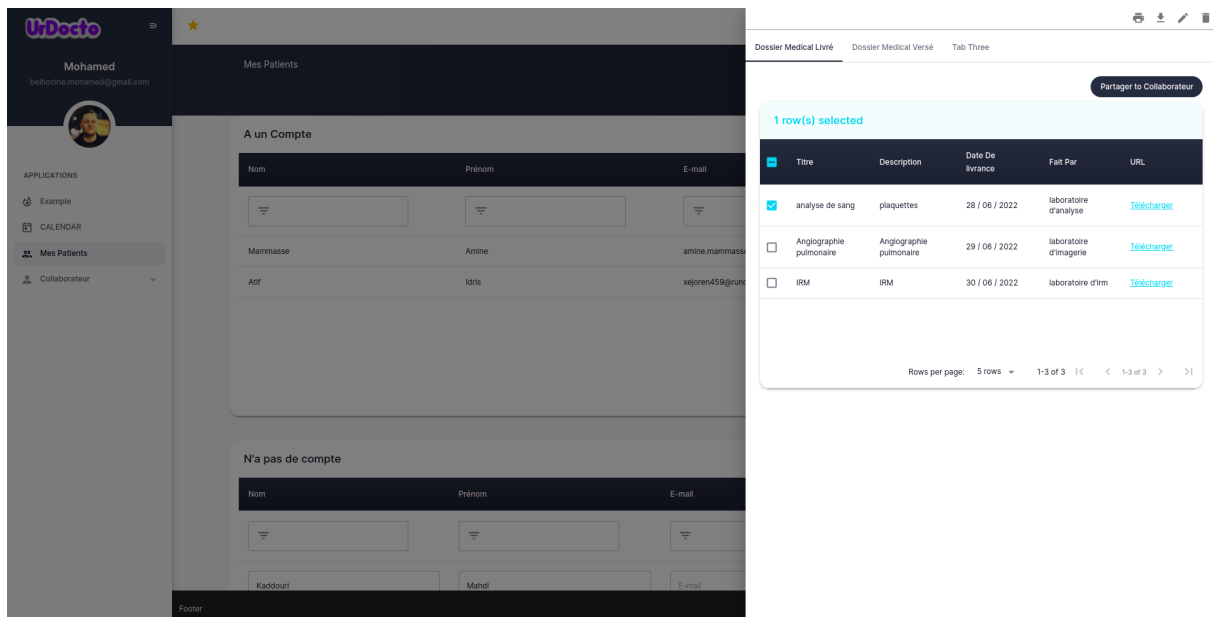


FIGURE 4.29 – Interface de gestion du dossier médical d'un patient côté médecin.

La Figure 4.30 représente l'interface où le médecin pourra rechercher des collaborateurs pour les ajouter à sa liste et partager des documents avec ces derniers.

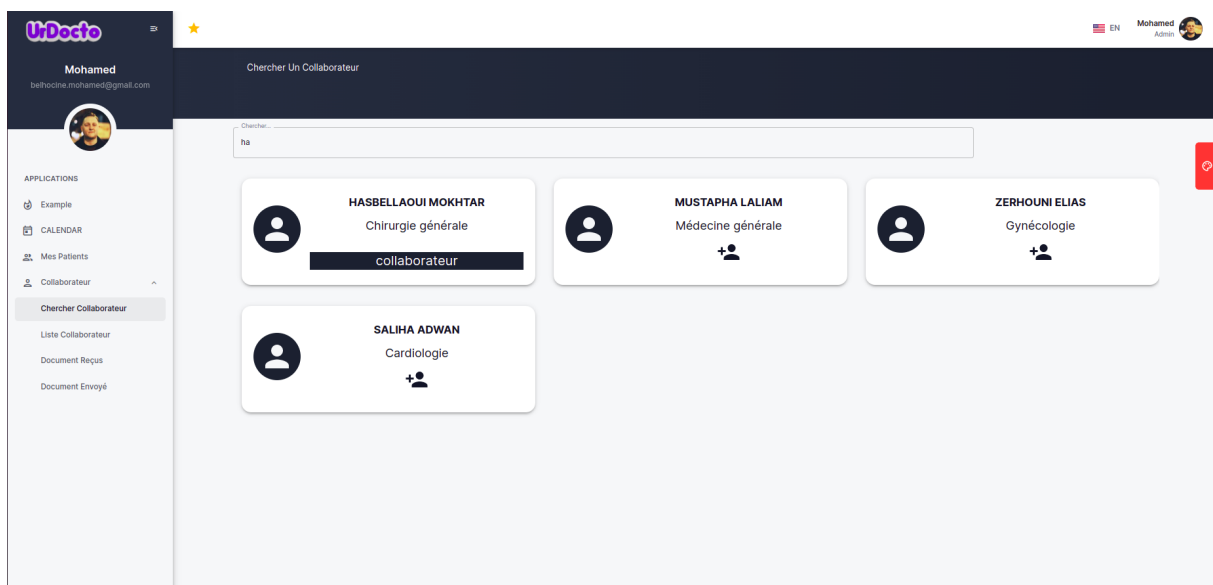


FIGURE 4.30 – Interface de recherche d'un collaborateur.

La Figure 4.31 représente l'interface où le médecin pourra consulter la liste de ses collaborateurs et supprimer un de ces derniers.

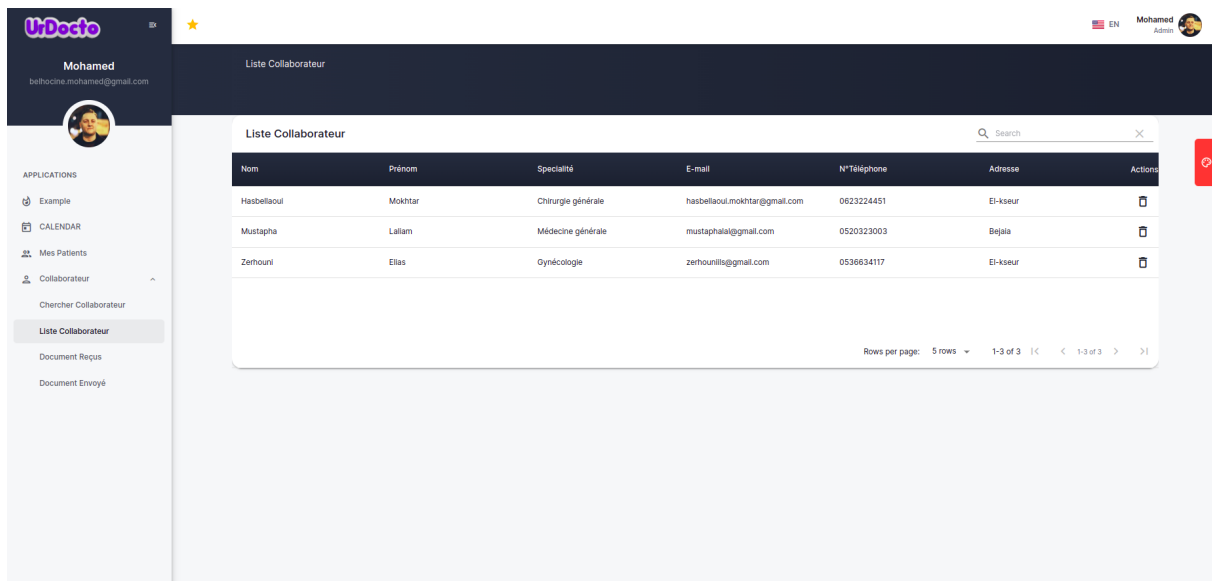


FIGURE 4.31 – Interface liste des collaborateurs.

La Figure 4.32 représente l'interface où le médecin pourra consulter les documents envoyés à ses collaborateurs, il pourra aussi annuler la visualisations sur un document.

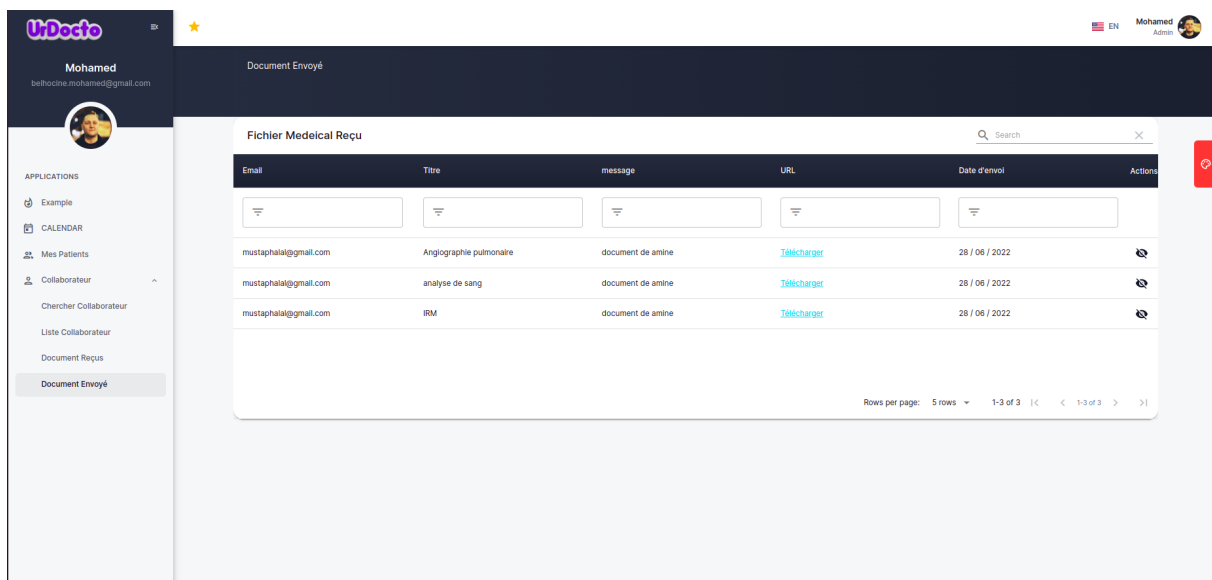


FIGURE 4.32 – Interface liste des documents envoyés.

4.9.2 Interfaces de l'application mobile

La Figure 4.33 représente l'interface où le patient pourra rechercher un médecin.



FIGURE 4.33 – Interface de l'application mobile pour la recherche d'un médecin.

La Figure 4.34 représente l'interface des résultats de recherche d'un médecin. Le patient pourra ainsi prendre un rendez-vous.

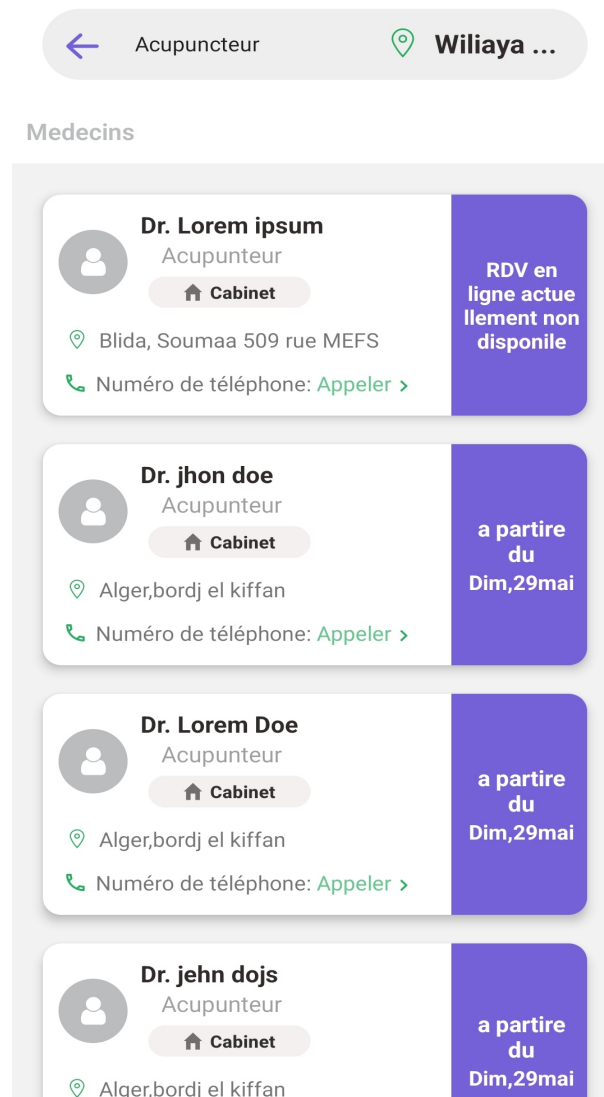


FIGURE 4.34 – Interface de l'application mobile des résultats de recherche d'un médecin.

La Figure 4.35 représente l'interface où le patient pourra consulter la liste de ses rendez-vous.

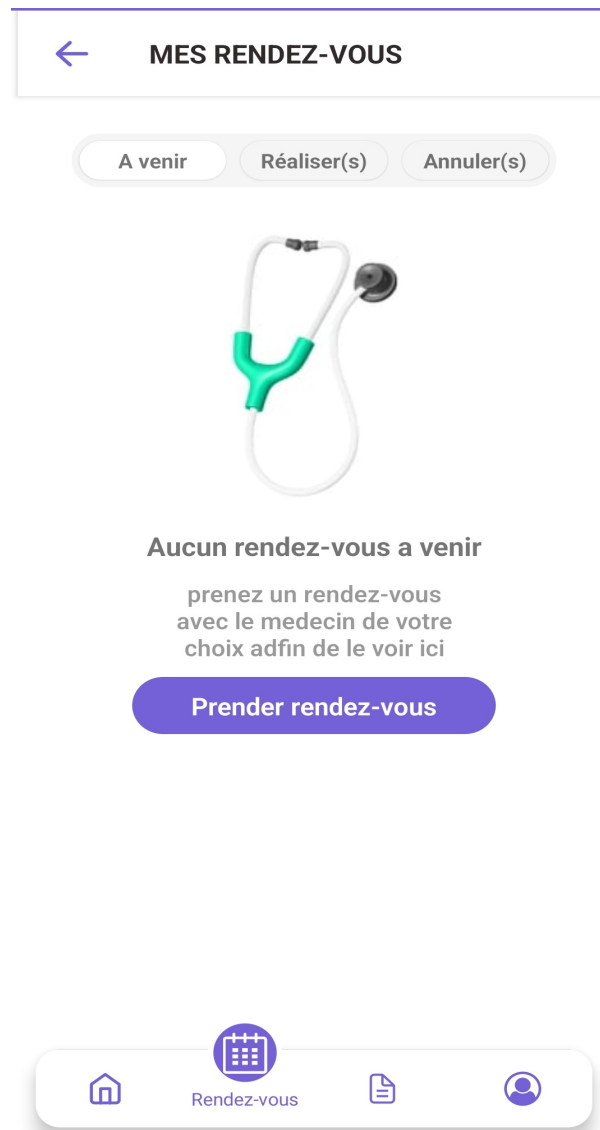


FIGURE 4.35 – Interface de l'application mobile de la liste des rendez-vous.

La Figure 4.36 représente l'interface où le patient pourra consulter son carnet de santé.

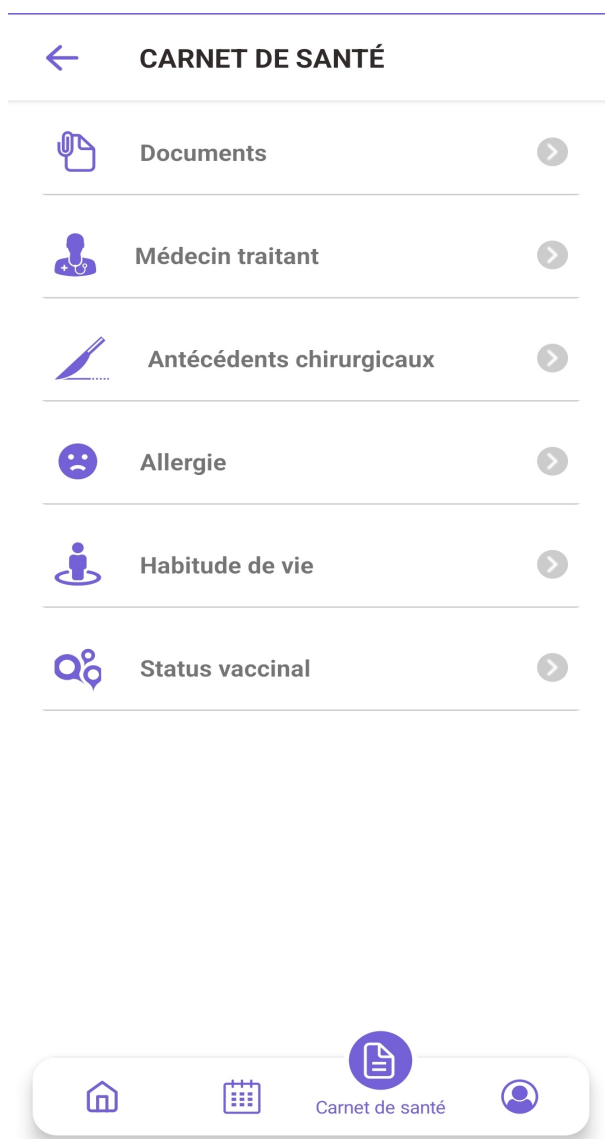


FIGURE 4.36 – Interface de l'application mobile pour le carnet de santé.

La Figure 4.37 représente l'interface du compte d'un patient.

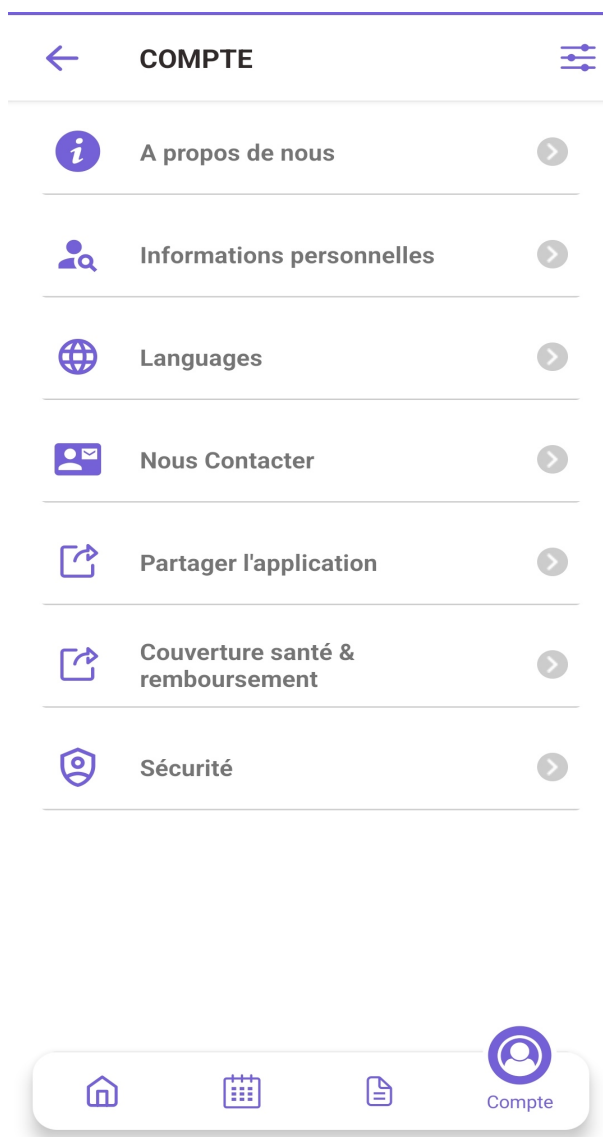


FIGURE 4.37 – Interface de l'application mobile du compte d'un patient.

4.10 Conclusion

Cette dernière partie de notre travail a été dédiée à l'implémentation, qui nous a permis d'atteindre les objectifs décrits dans le cahier des charges. Nous avons présenté l'environnement de développement et les principaux outils utilisés pour concevoir nos deux applications. Nous avons présenté également, les deux applications réalisées à travers une arborescence des vues, ainsi que les principales interfaces pour bien illustrer le travail qui a été fait.

Conclusion générale

En guise de projet de fin du cycle Master, option Génie Logiciel, nous avons travaillé sur la conception et la réalisation d'une plateforme web et d'une application mobile de santé nommée UrDocto.

L'objectif de ce projet était de régler certains problèmes liés à l'organisation et la couverture des zones isolées dans le secteur médical. Nous voulions, à travers ce travail, établir un moyen de communication entre les différents acteurs de la santé en Algérie. Ces acteurs peuvent être des médecins, comme ils peuvent être des patients. Notre plateforme leur offrira une opportunité d'exercer la médecine en utilisant les TIC, une meilleure organisation du dossier médical, une facilité à prendre un rendez-vous en ligne, ou tout simplement à consulter un médecin de chez soi.

Pour ce faire, nous avons commencé, dans un premier temps, par comprendre le contexte général de notre application en définissant les concepts essentiels de la e-santé, puis nous avons réalisé une étude de marché ce qui nous a permis d'identifier les différents objectifs et exigences à satisfaire, et de réaliser un cahier des charges qui contient les différentes fonctionnalités souhaitées. Afin d'assurer le bon déroulement du projet, nous avons opté pour la méthode SCRUM qui est la plus utilisée actuellement. Ces fonctionnalités, que nous avons nommées *sprints*, ont été traduites en différents diagrammes UML. Ce dernier est considéré comme l'un des langages de modélisation les plus utilisés pour illustrer la démarche de conception. L'étape suivante a fait figure d'inventaire de l'environnement et des outils de développement utilisés. Nous avons ainsi choisi la technologie MERN stack qui représente l'alliance des technologies les plus puissantes que l'on trouve actuellement sur le marché; à savoir, Mongo DB, Express, Reactjs et Node.js. Enfin, nous avons présenté les interfaces principales de notre plateforme UrDocto.

Des perspectives d'amélioration de notre application restent toutefois indispensables. Nous envisageons ainsi d'ajouter de nouvelles fonctionnalités telles que :

- Inclure d'autre organismes à notre application (e.g., les pharmacies, les laboratoires d'analyses médicales, etc.);
- Ajout d'une application mobile pour les médecin;
- La gestion de la comptabilité;

- L'ajout d'un module de géo-localisation qui va permettre de localiser les médecins à proximité;
- Objets connectés comme pour les applications de m-santé, on retrouvera des objets dédiés au bien-être tels que les bracelets connectés, et des objets permettant de mesurer spécifiquement une constante à valeur médicale (e.g., tensiomètre connecté, pilulier connecté pour l'observance, etc.);
- Big Data santé, utilisé dans le domaine du marketing, le big data est l'ensemble des systèmes permettant de collecter, traiter et analyser des volumes très importants de données, ayant souvent à voir avec les comportements des individus. Dans le domaine de la santé, devant l'explosion du volume de données de santé, le big data offre des perspectives uniques pour anticiper et prédire des phénomènes de santé majeurs, par exemple dans le domaine de l'épidémiologie, ou encore mesurer en temps réel l'efficacité de mesures de santé publiques ou de nouveaux dispositifs[38].

Bibliographie

- [1] Thèse C. Genève. État des lieux de la E-santé en 2020, étude d'une application mobile de santé. Sciences pharmaceutiques. 2020.
- [2] M. Dubreuil. E-santé : décryptage des pratiques et des enjeux. Observatoire régional de santé Île-de-France. 2019.
- [3] E-santé : les principaux outils numériques sont utilisés par 80% des médecins généralistes de moins de 50 ans. Direction de la recherche, des études de l'évaluation et des statistiques DREES. 2020. <https://drees.solidarites-sante.gouv.fr/sites/default/files/er1139.pdf>
- [4] A. Meziane. Le dossier médical partagé. Bulletin d'information trimestriel du CERIST, pp. 10-19, 2012.
- [5] Le dossier médical informatisé. CADUCEE. 2021. <https://www.caducee.net/DossierSpecialises/systeme-information-sante/dmi.asp>
- [6] N. Ho-Pun-Cheung. La télémédecine : les avantages et inconvénients. Le Blog du Groupe CRC. 2022. <https://blog.groupecrc.com/telemedecine-avantages-inconvenients> [Consulté le 29 Mai 2022]
- [7] L. Vassieux. Applications mobiles, objets connectés et promotion de la santé. Dossier Technique N°9. Ireps Bourgogne Franche-Comté. 2017. https://ireps-bfc.org/sites/ireps-bfc.org/files/dt_ocs_promotion_sante_version_finale.pdf[Consulté le 20 juin 2022]
- [8] K. Meziane. Santé 3.0, l'e-santé et la santé connectée nous tendent les bras. Journal du Community Manager. 2021. <https://www.journalducmm.com/sante-connectee-e-sante/> [Consulté le 29 Mai 2022]
- [9] C. Larouche. La prise de rendez-vous électronique. Le Médecin du Québec. 2014. <https://lemedecinduquebec.org/archives/2015/1/la-prise-de-rendez-vous-electronique-points-de-reflexion/> [Consulté le 23 juin 2022]
- [10] DZDOC - Trouver un médecin en Algérie et prenez rendez-vous en ligne. <https://dzdoc.com/> [Consulté le 10 janvier 2022]
- [11] eSiha - Trouver un médecin, centre d'analyse ou radiologie et paramédical en Algérie et prendre un rendez-vous. <https://www.esiha.net/> [Consulté le 10 janvier 2022]

-
- [12] Doctolib : Prenez rendez-vous en ligne chez un professionnel de santé. <https://www.doctolib.fr/> [Consulté le 10 janvier 2022]
- [13] Livi - Consultation médicale en ligne – 7 jours sur 7. <https://www.livi.fr/> [Consulté le 10 janvier 2022]
- [14] La méthode SCRUM pour les nuls. Ignition program. <https://ignition-program.com/tuto/la-methode-scrum-pour-les-nuls?locale=fr> [Consulté le 10 juin 2022]
- [15] J. Beutler. Scrum, une méthode de développement agile. 2021. <https://www.cfi.ch/scrum-une-methode-de-developpement-agile/>[Consulté le 10 juin 2022]
- [16] A. Grisey. Qu'est ce que GitLab? Définition et utilisation. 2021. https://talks.freelancerepublik.com/quest-ce-que-gitlab-definition-et-utilisation/#GitLab_la-vraie_definition[Consulté le 12 mars 2022]
- [17] Visual Studio Code. <https://framalibre.org/content/visual-studio-code/>[Consulté le 4 mai 2022]
- [18] Agence Nationale de la Sécurité Informatique - Android Studio. ANSI. <https://www.ansi.tn/fr/assistance/outils-de-securite/android-studio>[Consulté le 4 mai 2022]
- [19] PlantUML. Outil open-source qui utilise des descriptions textuelles simples pour dessiner des diagrammes UML. <https://plantuml.com/fr/>[Consulté le 4 mai 2022]
- [20] MongoDB. What Is The MERN Stack? Introduction & Examples. <https://www.mongodb.com/mern-stack>[Consulté le 4 mai 2022]
- [21] Comprendre la MERN stack. Le blog du codeur. 2019. <https://leblogducodeur.fr/mern-stack/>[Consulté le 4 mai 2022]
- [22] Node Package Manager (NPM). Formations JavaScript. 2021. <https://formationjavascript.com/guides/npm-node-package-manager/>[Consulté le 4 mai 2022]
- [23] React – Une bibliothèque JavaScript pour créer des interfaces utilisateurs. <https://fr.reactjs.org/>[Consulté le 7 mai 2022]
- [24] B. Manseur. Tutoriel E-Learning Base 64. LearnCodiz. <https://www.learncodiz.com/Cours/Base-64/70/9>[Consulté le 9 mai 2022]
- [25] Bootstrap vs Material UI : Quel framework choisir? Mobiskill. 2021. <https://mobiskill.fr/blog/conseils-emploi-tech/bootstrap-vs-material-ui-quel-framework-choisir/>[Consulté le 4 janvier 2022]
- [26] Release de la V1 de Chakra UI. Premier Octet. 2021. <https://www.premieroctet.com/blog/chakraui-v1>[Consulté le 4 janvier 2022]

-
- [27] Redux (bibliothèque JavaScript). Wikiwand. [https://www.wikiwand.com/fr/Redux_\(biblioth%C3%A8que_JavaScript\)](https://www.wikiwand.com/fr/Redux_(biblioth%C3%A8que_JavaScript))[Consulté le 4 janvier 2022]
- [28] La Rédaction. Axios : concevoir une requête Post pour renvoyer les données d'un formulaire. JDN. 2020. <https://www.journaldunet.fr/web-tech/developpement/1441159-axios-concevoir-une-requete-post-pour-renvoyer-les-donnees-d-un-formulaire/>[Consulté le 4 janvier 2022]
- [29] Gestion des dates et des heures à l'aide de Moment.js. Blog ARC Optimizer. 2020. <https://blog.arcoptimizer.com/gestion-des-dates-et-des-heures-a-laide-de-moment-js>[Consulté le 13 avril 2022]
- [30] Développement de sites web avec l'architecture MVC. Blog Webcky. 2016. <https://www.webcky.fr/blog/2016/09/16/developpement-site-web-avec-larchitecture-mvc-en-isere/>[Consulté le 29 juin 2022]
- [31] A. Marciano. Votre première application en React, Node, Express & MongoDB. Medium. 2022. <https://axel-marciano.medium.com/votre-premi%C3%A8re-application-en-react-node-express-mongodb-5ab0dc531091>[Consulté le 4 janvier 2022]
- [32] Qu'est-ce que NoSQL? Oracle. <https://www.oracle.com/dz/database/nosql/what-is-nosql/>[Consulté le 16 janvier 2022]
- [33] Introduction to Mongoose for MongoDB. freeCodeCamp. 2019. <https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>[Consulté le 16 janvier 2022]
- [34] J. Lamprea & Hermesf. Agora Video for WordPress. WordPress.org Français. 2022. <https://fr.wordpress.org/plugins/wp-agora-io/>[Consulté le 20 juin 2022]
- [35] Wazo. Différence entre WebRTC et SIP | Guide VoIP & WebRTC. 2020. <https://wazo.io/guide-voip-webrtc/webrtc/difference-webrtc-sip/>[Consulté le 20 juin 2022]
- [36] La Rédaction. Netlify : utiliser ce CMS pour plus de performance web. JDN. 2022. <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1511305-netlify-utiliser-c>[Consulté le 29 juin 2022]
- [37] La Rédaction. Heroku : la plateforme cloud de Salesforce. JDN. 2022. <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1511295-heroku-la-plateforme-cloud-de-salesforce/>[Consulté le 29 juin 2022]
- [38] L. Lichy. Les mots de la e-Santé : 5 minutes pour tout comprendre. Pharmagest. 2017. <https://pharmagest.com/mots-de-e-sante-5-minutes-comprendre/>[Consulté le 29 juin 2022]

Annexe A

PV des tests sur l'application UrDocto



PV des tests sur l'application UrDocto

Nous l'équipe chef de projets de **BigNova** sous la direction de **Mr Reda BEKKA** et suite aux tests concluants effectués ce jour le **28/06/2022** sur l'application web réalisé par nos deux stagiaires **Mr Amine MAMMASSE** et **Mr Farouk BENNAI** qui ont travaillé sur la partie backend et frontend ainsi que l'aspect UI/UX du projet.

Nous avons jugé que l'application est exploitable, par conséquent nous avons décider de lancer la partie production du projet et la proposer au marché national.

Amine et **Farouk** se sont engagé à continuer l'aventure avec nous en tant qu'ingénieurs d'études et de développement et de travailler principalement sur ce projet pour toute mise à jour et maintenance de l'application.

Nous tenons à remercier les deux stagiaires pour leurs assiduité, sérieux et efficacité pendant la période de stage.

BIGNOVA
Programmation Informatique
Tel: 0782 20 50 66
Cité 612 logts Bt D N°04 Sidi Ahmed
Bejaïa - RC N°21/A/5810966-00/06

Site web : <https://bignovacompany.com>

N° Téléphone : (+213) 782205066

Adresse email : contact@bignovacompany.com



Résumé

Ce mémoire a été rédigé en vue de l'obtention du diplôme de fin de cycle Master en Informatique, option Génie Logiciel. Nous avons travaillé sur la conception et la réalisation d'une application web et mobile de santé. Le système de santé Algérien se modernise de jour en jour et se retrouve dans la nécessité d'informatiser l'information médicale. En effet, la complexité croissante de la médecine occidentale pousse de manière naturelle à la mise en place de systèmes d'information capables d'aider le praticien et le patient dans leurs tâches quotidiennes. Afin d'atteindre cet objectif, il nous a été proposé de concevoir et réaliser une application web et mobile de santé assurant la gestion et le suivi des patients tout en proposant diverses options et fonctionnalités comme la prise de rendez-vous en ligne, la gestion du dossier médical et la téléconsultation. Ceci permettra d'améliorer la collaboration entre les acteurs dans le secteur médical afin d'offrir un meilleur rendement et de meilleurs prestations. Pour ce faire, nous avons choisi de modéliser notre système avec le formalisme UML, afin d'assurer le bon déroulement du projet, nous avons opté pour la méthode SCRUM, et pour la réalisation nous avons utilisé la technologie MERN.

Mots clés : E-santé, M-santé, Dossier médical, Téléconsultation, SCRUM, MERN.

Abstract

This thesis was written for the end of the Master's degree in Computer Science, Software Engineering option. We worked on the design and implementation of a web and mobile health application. The Algerian health system is modernizing day by day and finds itself in the need to computerize medical information. Indeed, the increasing complexity of Western medicine naturally pushes to the implementation of information systems capable of helping the practitioner and the patient in their daily tasks. In order to achieve this objective, we have been proposed to design and implement a web and mobile health application that will ensure the management and follow-up of patients while offering various options and functionalities such as online appointment booking, medical record management and teleconsultation. This will improve the collaboration between the actors in the medical sector in order to offer a better performance and better services. To do this, we chose to model our system with the UML formalism, to ensure the smooth running of the project, we opted for the SCRUM method, and for the realization we used the MERN stack technology.

Keywords : E-health, M-health, Medical record, Teleconsultation, SCRUM, MERN.