

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El-Bachir El-Ibrahimi - BBA
Faculté des Mathématiques et Informatiques



MÉMOIRE

Présenté en vue de l'obtention du diplôme

Master en Informatique

Spécialité : Technologie de l'information et de la communication

Thème

Conception et développement d'une plateforme intelligente de soins médicaux à domicile.

Présenté par :

- | | |
|-------------------------|----------------------|
| - Bengrine Abderrahmane | - Benmessaoud Wassim |
| - Hammache Riadh | - Slimane Hadjrioua |
| - Djelloul Raid | - Firas Bensalem |

Soutenu le : 06/11/2025, devant la commission d'examen suivante :

Dr. BENAOUA Nadjib	Président
Dr. NAILI Makhlouf	Examineur
Dr. BENGUEDDOUDJ Abdallah	Examineur
Dr. ZOUACHE Djaafar	Encadrant

2024 / 2025

Dedicace

Je dédie ce modeste travail à tous ceux qui ont partagé avec moi les moments et les émotions qui ont marqué mon parcours académique et personnel, à ceux qui ont été pour moi un soutien et un appui constants. À tous ceux qui ont cru en moi et m'ont accordé leur confiance, à ceux qui se sont tenus à mes côtés dans les moments de besoin, dont les paroles et le soutien ont ravivé en moi la flamme du succès et semé dans mon âme l'esprit de persévérance et de travail. À ma chère mère, paradis de Dieu sur terre, mon modèle de patience et de dévouement, mon soleil qui a illuminé ma route. À mon cher père, symbole de sagesse et de force, qui ne m'a jamais privé de son appui et de ses précieux conseils. À mon frère Abdelbar, dont l'énergie et l'optimisme me donnent espoir, et à ma sœur, médecin dévouée, dont la détermination et l'excellence ne cessent de m'inspirer. À ma grand-mère et à tous les membres de ma famille, qui ne m'ont jamais privé de leur soutien et de leurs prières sincères.

À tous mes amis chers, et en particulier à mes amis Taki Eddine et Aymen, ainsi qu'à mes compagnons de route qui ont partagé avec moi les défis et les efforts du savoir. Votre présence bienveillante et votre soutien sincère ont été pour moi une source de force et de joie.

À mes collègues de recherche, unis par l'esprit de collaboration et de persévérance, et à mes collègues de travail, qui ont enrichi mon parcours professionnel par leurs expériences et leur amitié. Je vous dédie à tous le fruit de mes efforts et toute ma reconnaissance.

Je remercie tous les professeurs qui ont contribué à faire de mon parcours académique. Sans vos connaissances et vos conseils, toute ma croissance et mes réalisations n'auraient pas été possibles. Votre influence et votre soutien sont grandement appréciés.

Abderrahmane

Dedicace

À nos professeurs ces lumières qui ont éclairé notre chemin, qui ont forgé notre intellect et qui ont éveillé en nous cette envie d'apprendre. Votre patience, votre sagesse et votre confiance ont été les clés de notre réussite.

À nos familles et à nos amis ces héros qui nous ont soutenus dans les instants de doute, de lassitude et d'incertitude. Votre amour, votre Jésus-Christ, votre patience et surtout votre Foi ont été les petites gouttes d'eau qui ont fait déborder le vase.

Chaque réussite porte en elle la trace de tout ce que vous nous avez donné. C'est à vous que nous devons tout, du plus profond de notre cœur merci.

wassime

Dedicace

Je dédie ce travail à mes parents, les premiers piliers de ma vie, les architectes silencieux de chacun de mes pas. Rien ne saurait exprimer la profondeur de ma gratitude pour tout ce que vous avez fait, pour tout ce que vous avez donné sans jamais compter, et pour l'amour inépuisable dont vous m'avez entourée. Vos sacrifices, souvent invisibles, ont tracé le chemin que j'emprunte aujourd'hui avec fierté et reconnaissance. Vous avez cru en moi bien avant que je croie en moi-même, et dans vos regards, j'ai toujours trouvé la force de me relever, la confiance de continuer, et la douceur de ne jamais me sentir seule. Papa, tu as été la main ferme et rassurante, celle qui guide sans jamais imposer, qui enseigne la valeur du travail, du respect et de l'honnêteté. Tes paroles, simples mais justes, ont souvent été le repère dont j'avais besoin pour avancer dans le tumulte des incertitudes. Maman, ton amour est ce refuge immuable, cette lumière douce qui ne s'éteint jamais. Dans ton sourire, j'ai trouvé le courage, dans ta patience, la sérénité, et dans tes silences, une tendresse infinie. Tu as été, et tu restes, le cœur battant de tout ce que je suis. À vous deux, je dois bien plus qu'un parcours universitaire. Ce mémoire n'est qu'une humble offrande, le reflet d'un héritage d'amour, de persévérance et de dignité que vous m'avez transmis. Chaque page, chaque mot, porte un peu de votre empreinte — celle de la confiance, de la foi et de la force que vous avez su m'inspirer. Ce succès est le vôtre autant que le mien, car derrière chaque réussite, il y a vos visages, vos prières, et vos sacrifices silencieux. De tout cœur, merci pour tout.

Riad

Dedicace

Je dédie ce travail à mes parents, premiers piliers de ma vie, dont les sacrifices invisibles ont tracé chaque pas de mon chemin. Votre foi en moi a été plus forte que mes doutes, et votre amour, la lumière qui m'a guidé même dans les nuits les plus longues.

À mon frère, complice discret et soutien constant, merci d'avoir été là pas seulement comme un frère, mais comme un ami, un refuge, une force tranquille à laquelle je pouvais toujours me fier.

À mes deux sœurs, présences douces et rassurantes, merci pour vos mots justes au bon moment, pour vos appels pleins d'encouragements, et pour cette tendresse qui, sans bruit, m'a redonné le courage d'avancer quand tout me semblait lourd. À ma nièce, dont les rires innocents et les premiers pas ont apporté tant de lumière dans nos vies merci d'être cette douce présence qui réchauffe nos cœurs.

À mes amis, véritables compagnons de route, merci d'avoir partagé avec moi les nuits blanches, les idées floues, les rires spontanés et les silences compréhensifs. Votre amitié a été un cadeau précieux dans cette aventure, et je n'aurais pas franchi toutes ces étapes sans votre présence bienveillante.

Enfin, à tous ceux qui, d'une manière ou d'une autre, ont contribué à ce parcours par un conseil, un regard, une prière sachez que ce modeste travail porte aussi votre empreinte.

Firas

Dedicace

Je dédie ce travail à mes parents, piliers fondateurs de ma vie, dont les sacrifices invisibles ont tracé, pas à pas, la route qui m'a conduit jusqu'ici. Rien n'aurait été possible sans votre foi inébranlable, plus forte que mes doutes, ni sans cet amour silencieux et profond qui a éclairé mes jours les plus sombres et apaisé mes nuits d'incertitude. Vous êtes la source de mon courage, le reflet de ma persévérance et la raison première de mes réussites. Ce modeste travail n'est qu'un écho de tout ce que vous m'avez transmis : la patience, l'effort et le sens du devoir.

Ma gratitude s'étend à ma sœur et à son époux pour leur présence bienveillante et leur appui constant, ainsi qu'à leur fils, source de joie et de lumière dans mes journées de fatigue.

À ma petite sœur, merci pour ta tendresse, ta joie et ton soutien sincère tu as été une douce force dans les moments de doute.

Enfin, à tous ceux qui, de près ou de loin, ont contribué à ce parcours par un conseil, une parole, une pensée ou une prière je vous adresse ma profonde gratitude. Chacun de vous, à sa manière, a laissé une empreinte dans cette aventure, et c'est grâce à cet ensemble de présences, d'encouragements et de gestes sincères que ce mémoire a pu voir le jour. Ce travail est donc bien plus qu'un aboutissement personnel : il est le fruit d'un amour partagé, d'un soutien collectif et d'une confiance tissée au fil du temps.

Slimane

Dedicace

Je dédie ce travail à ma mère, cette femme exceptionnelle dont la tendresse et la force silencieuse ont façonné tout ce que je suis devenue. Maman, tu es le cœur de ma vie, la lumière qui ne vacille jamais, même quand tout autour semble s'éteindre. Tes sacrifices, discrets mais immenses, ont tracé les contours de mes rêves, et ton amour, inépuisable et vrai, a nourri chacun de mes pas. Dans tes yeux, j'ai trouvé le courage de recommencer, dans tes mots, la douceur du réconfort, et dans ton silence, une sagesse infinie. Tu as su m'apprendre qu'aucun effort n'est vain, que la foi précède toujours la réussite, et que la bonté demeure la plus belle des forces. Ce mémoire, fruit de persévérance et d'espoir, est avant tout le reflet de ton amour et de ta présence constante dans ma vie. À toi, Maman, je dédie cette réussite avec tout l'amour, la reconnaissance et la fierté d'un enfant comblé par tant de bienveillance.

Je souhaite également dédier ce travail à mes amis de travail, ces compagnons de route qui ont su transformer les journées les plus ordinaires en moments d'échange, de rire et de complicité. Votre soutien, vos encouragements et votre compréhension ont souvent été ce souffle discret mais essentiel qui m'a permis de garder le cap, même lorsque la fatigue ou le doute prenaient le dessus. Entre deux dossiers, une pause-café ou un sourire partagé, vous avez été présents simplement, sincèrement et cela a eu plus d'importance que vous ne pouvez l'imaginer. Merci pour vos mots réconfortants, votre patience et votre humanité. Ce travail porte aussi la trace de votre amitié, de votre bonne humeur et de cette solidarité silencieuse qui rend les journées plus légères et les défis plus supportables.

À ma mère bien-aimée et à mes chers collègues devenus amis, je dis du fond du cœur : merci. Ce mémoire est le fruit d'un chemin partagé, empreint de confiance, d'affection et de gratitude infinie.

Raid

Remerciements

Au nom de Dieu, le Tout Miséricordieux, le Très Miséricordieux. Louange à Dieu, dont la grâce permet l'accomplissement des bonnes œuvres et la réalisation des aspirations. Nous Le remercions et Le louons pour les bienfaits, la réussite et la clairvoyance dont Il nous a comblés tout au long de notre parcours académique, et nous Lui demandons de nous accorder Sa faveur et Sa guidance dans la quête du savoir et de la connaissance.

C'est avec une profonde reconnaissance et une immense gratitude que nous adressons nos sincères remerciements à notre honorable enseignant, notre frère et mentor avant d'être notre encadrant, Dr. ZOUACHE Djafar, pour son soutien constant, ses précieux conseils et la confiance qu'il a toujours placée en nous. Sa sagesse, sa patience et ses encouragements continus ont été parmi les principales raisons de l'achèvement de ce travail. C'est un véritable honneur pour nous d'avoir eu l'opportunité de travailler sous sa direction bienveillante.

Nous exprimons également notre profonde gratitude à l'ensemble des professeurs de notre département, en particulier à tous les enseignants de la Faculté des Mathématiques et d'Informatique, pour leurs efforts, leur disponibilité et leur accompagnement précieux tout au long de notre formation. Nos remerciements s'adressent aussi à l'ensemble du personnel administratif du département et de la faculté pour leur dévouement et leur soutien constant.

Nous tenons aussi à exprimer notre haute appréciation à l'équipe de l'incubateur universitaire, pour les moyens mis à notre disposition, leur accompagnement et leur appui continu tout au long de ce projet. Ils ont été, sans aucun doute, des partenaires essentiels dans la réussite de notre travail.

Nous exprimons également nos sincères remerciements aux membres du jury de soutenance, qui ont eu la gentillesse d'accepter l'évaluation et la discussion de notre travail, et dont les remarques pertinentes et les orientations avisées resteront une source d'inspiration et de fierté dans notre parcours scientifique.

Enfin, nous adressons nos remerciements les plus chaleureux à tous les administrateurs et employés de l'université, qui ont contribué à créer les conditions favorables à la réalisation de ce

travail. Que Dieu les récompense pour leur dévouement, leurs efforts et leur sincère engagement.

Résumé

Les services numériques jouent aujourd’hui un rôle fondamental dans la modernisation du secteur de la santé, où la rapidité, la traçabilité et la coordination sont essentielles. En Algérie, les soins infirmiers à domicile restent limités par le manque de solutions technologiques adaptées. Le projet MediCall vise à combler cette lacune en proposant une plateforme numérique intelligente et sécurisée dédiée à la gestion et à la réservation des soins à domicile.

Cette plateforme agit comme un intermédiaire entre patients et infirmiers, automatisant le processus de soins depuis la demande jusqu’au paiement. Elle comprend trois applications complémentaires : une application web pour l’administration et la supervision, une application mobile Flutter pour les infirmiers et les patients, et une application desktop destinée aux coordinateurs médicaux pour la planification et le suivi.

Le système intègre également un serveur intelligent basé sur l’intelligence artificielle, doté d’un chatbot médical capable de fournir des réponses précises et contextuelles, améliorant ainsi l’expérience utilisateur et l’efficacité du service.

Mots clés :soins infirmiers à domicile, santé numérique, plateforme intelligente, intelligence artificielle, application web, application mobile, application desktop, chatbot médical, automatisation des soins.

Abstract

Digital services play a crucial role in modernizing the healthcare sector, where efficiency, traceability, and coordination are key. In Algeria, home nursing care still faces a lack of advanced technological tools. The MediCall project aims to bridge this gap by developing an intelligent and secure digital platform dedicated to managing and booking home nursing services.

The platform acts as a reliable intermediary between patients and nurses, automating the care process from the initial request to payment. It consists of three interconnected applications : a web application for administration and supervision, a mobile application (Flutter) for patients and nurses, and a desktop application for medical coordinators to manage and monitor operations.

The system also integrates an intelligent server powered by artificial intelligence, featuring a medical chatbot capable of providing accurate, contextualized responses — enhancing user experience and service efficiency.

Keywords :home nursing, digital health, intelligent platform, artificial intelligence, web application, mobile application, desktop application, medical chatbot, care automation.

المخلص

تلعب الخدمات الرقمية اليوم دورًا محوريًا في تحديث قطاع الصحة، إذ أصبحت السرعة وإمكانية التتبع والتنسيق عناصر أساسية لضمان جودة الخدمات الصحية. وفي الجزائر، ما تزال خدمات التمريض المنزلي تعاني من محدودية في الانتشار بسبب غياب الحلول التكنولوجية المناسبة التي تواكب التطور الرقمي. ومن هذا المنطلق، جاء مشروعنا ليملاً هذه الفجوة من خلال تقديم منصة رقمية ذكية وأمنة مخصصة لإدارة وحجز خدمات الرعاية الصحية المنزلية.

تعمل هذه المنصة كوسيط فعال بين المرضى والمرضى، حيث تقوم بأتمتة جميع مراحل عملية الرعاية بدءًا من تقديم الطلب وصولاً إلى الدفع، مما يسهل التواصل ويضمن الشفافية في تقديم الخدمات. وتضم المنصة ثلاث تطبيقات متكاملة هي: تطبيق ويب مخصص للإدارة والإشراف على الأنشطة، وتطبيق محمول مطور موجه للمرضى والمرضى لتسهيل عملية الحجز والمتابعة، بالإضافة إلى تطبيق مكتبي خاص بالمنسقين الطبيين يُستخدم للتخطيط وتنظيم الجداول ومتابعة الحالات.

كما يحتوي النظام على خادم ذكي يعتمد على تقنيات الذكاء الاصطناعي، مزود برобوت محادثة طبي قادر على تقديم إجابات دقيقة وملائمة للسياق الطبي، مما يساهم في تحسين تجربة المستخدم ورفع كفاءة الخدمة المقدمة.

الكلمات المفتاحية: التمريض المنزلي، الصحة الرقمية، منصة ذكية، الذكاء الاصطناعي، تطبيق ويب، تطبيق محمول، تطبيق مكتبي، روبوت محادثة طبي، أتمتة الرعاية.

Table des matières

Liste des figures	xvii
Liste des tables	xviii
Abréviations	xix
1 Introduction Générale	1
1.1 Contexte.....	1
1.2 Problématique.....	2
1.3 Objectifs.....	2
1.4 Méthodologie et résultats.....	3
1.4.1 Étude du domaine et identification des besoins.....	3
1.4.2 Analyse et spécification des besoins.....	4
1.4.3 Conception du système.....	4
1.4.4 Choix technologiques et développement.....	4
1.4.5 Résultats obtenus.....	5
1.5 Structure du rapport.....	5
2 Fondements techniques du développement web et mobile	7
2.1 Introduction.....	7
2.2 Les bases du Web.....	7
2.2.1 Réseau Internet.....	8
2.2.2 Modèles de communication.....	8
2.2.2.1 Modèle de communication Client-Serveur.....	8
2.2.2.2 Modèle de communication poste à poste (P2P).....	8
2.2.3 Protocole HTTP.....	9
2.2.4 adresses web.....	9
2.2.5 Le Web.....	9
2.3 Site Web statique & site Web dynamique.....	10

2.3.1	Site web statique.....	10
2.3.2	Site Web dynamique	10
2.4	Programmation Web.....	10
2.4.1	Présentation.....	10
2.4.2	Programmation côté serveur (Back-End).....	11
2.4.3	Technologies connexes	11
2.4.4	Programmation côté client (Front-End).....	11
2.4.5	Technologies connexes	12
2.5	Applications mobiles.....	12
2.5.1	Présentation.....	12
2.5.2	Avantages.....	13
2.5.3	Technologies et langages connexes.....	13
2.6	Applications de bureau.....	14
2.6.1	Présentation.....	14
2.6.2	Avantages.....	15
2.6.3	Technologies et langages utilisés.....	15
2.6.4	Architecture et fonctionnement.....	16
2.6.5	Déploiement et distribution.....	17
2.6.6	Applications hybrides et évolutions récentes.....	17

3 Étude de l'existant et spécification des besoins 18

3.1	Introduction.....	18
3.2	Description de notre domaine d'étude.....	19
3.2.1	Soins de santé à domicile en Algérie.....	19
3.2.2	Scène des services infirmiers en Algérie.....	19
3.2.3	Expériences numériques et innovations en santé.....	20
3.3	État de l'existant	21
3.3.1	Solutions mises en place par l'État.....	21
3.3.1.1	E-Santé Algérie.....	22
3.3.1.2	Portail CNAS / CASNOS	22
3.3.2	Solutions privées et initiatives numériques.....	22
3.3.2.1	DocteurDZ.....	23
3.3.2.2	Clinido.....	23
3.3.3	Évaluation comparative des solutions existantes.....	23
3.3.4	Lacunes du marché.....	23

3.3.5	Synthèse	24
3.4	Objectifs du projet.....	25
3.4.1	Objectif global.....	25
3.4.2	Objectifs spécifiques	25
3.5	Marché et utilisateurs visés.....	26
3.6	Spécifications des besoins.....	27
3.6.1	Présentation.....	27
3.6.2	Acteurs	27
3.6.3	Besoins fonctionnels.....	28
3.6.4	Besoins non-fonctionnels.....	30
3.7	Conclusion.....	30
4	La modélisation – UML	32
4.1	Introduction.....	32
4.2	Modélisation UML	33
4.2.1	Définition d'un modèle.....	33
4.2.2	Définition de la modélisation.....	33
4.2.3	Les perspectives de modélisation	33
4.3	Langage UML.....	34
4.3.1	Présentation.....	34
4.3.2	Principaux diagrammes UML	35
4.4	Vue Fonctionnelle : Diagramme de Cas d'Utilisation.....	36
4.4.1	Présentation.....	36
4.4.2	Diagrammes de cas d'utilisation de systeme de gestion du MediCall.....	38
4.4.3	Diagrammes de cas d'utilisation de systeme de gestion de patient	39
4.4.4	Diagrammes de cas d'utilisation de systeme de gestion de soignant.....	40
4.5	Vue Dynamique : Diagramme de Séquence Système.....	41
4.5.1	Présentation.....	41
4.6	Vue statique : diagramme de classes.....	41
4.6.1	Passage vers le Modèle Logique de Données (MLD).....	42
4.7	Conclusion.....	43
5	Réalisation et l'Implémentation	44
5.1	Introduction.....	44
5.2	Architecture de Système	45
5.3	Technologies et Outils de Développement.....	46

5.3.1	Pour le Développement Web	46
5.3.1.1	Next.js	46
5.3.1.2	TypeScript	47
5.3.1.3	HTML et CSS	47
5.3.1.4	Tailwind CSS et Shadcn	47
5.3.2	Pour le développement mobile	49
5.3.2.1	Flutter	49
5.3.2.2	PHP	49
5.3.2.3	Firestore Realtime Database	50
5.3.3	Pour le développement Desktop	50
5.3.3.1	Electron.js	50
5.3.3.2	SQLite	51
5.3.4	Pour les Services Backend et la Gestion des Données	51
5.3.4.1	API RESTful	51
5.3.4.2	Postman	52
5.3.4.3	MySQL	52
5.3.4.4	Git et GitHub	52
5.3.4.5	Seats.io	53
5.4	Expérience Utilisateur et Navigation	53
5.4.1	Présentation de notre API	54
5.4.2	Routes API pour l'application mobile	54
5.4.3	Routes API pour les Infirmiers Autonomes et les Infirmiers Affiliés aux Centres Médicaux (Application Mobile)	54
5.4.4	Routes API pour les Administrateurs de Centres Médicaux (Application Web)	55
5.4.5	Routes API pour les Administrateurs de la Plateforme (Application Web)	55
5.5	Diagrammes de Flux Utilisateur	56
5.5.1	Diagramme de Flux du Patient (Utilisateur Principal)	57
5.5.2	Diagramme de Flux de l'Infirmier Autonome	57
5.5.3	Diagramme de Flux de l'Administrateur d'Établissement de Santé	58
5.5.4	Diagramme de Flux de l'Infirmier Affilié à un Établissement	58
5.5.5	Diagramme de Flux de l'Administrateur de la Plateforme	59
5.6	Le Chatbot	59
5.6.1	Grands Modèles de Langage (MLL)	59

5.6.2	Implémentation de LLaMA-3.2-1B	60
5.6.3	Collecte et Préparation des Données.....	61
5.6.4	Résumé des Fichiers PDF	61
5.6.5	Résumé des Livres.....	62
5.7	Presentation de notre Application.....	62
6	Conclusion générale	67
6.1	Contributions	67
6.2	Critique du travail.....	68
6.3	Travaux futurs et perspectives.....	68
	Bibliographie	70
	Annexe	71

Table des figures

4.1	Diagrammes de cas d'utilisation de systeme de gestion	38
4.2	Diagrammes de cas d'utilisation d'acteur patient.....	39
4.3	Diagrammes de cas d'utilisation d'acteur soignant.....	40
4.4	Diagrammes de Class.....	42
5.1	Architecture de l'API pour les automobilistes en détresse.....	46
5.2	Préparation des Données.....	61
5.3	Sélection du modèle de base et authentification.....	61
5.4	Optimisations de la mémoire.....	62
5.5	sélection des hyperparamètres.....	63
5.6	Résultats d'Entraînement.....	64
5.7	Préparation au Déploiement.....	64
5.8	Login admin.....	66
5.9	Admin Dashboard.....	67
5.10	Interface de site.....	69
5.11	Interface d'application.....	70
5.12	Créer un compte.....	70
5.13	Ecran de connexion.....	71
5.14	Tracage en direct.....	71
5.15	Information personnels.....	72
5.16	Evaluation des performance.....	72
5.17	Profile de patient.....	73
5.18	Ecran d'accueil patient.....	73
5.19	Discussion IA.....	74
5.20	Profile de doctor.....	74
5.21	Nomination.....	75
5.22	doctore en line.....	75

Liste des tableaux

3.1	Différences entre les soins traditionnels et numériques à domicile (Projet Need Nursing)	20
3.2	Catégorisation des soins infirmiers à domicile	20
3.3	Intégration des technologies émergentes dans la plateforme MediCall	21
3.4	Avantages et faiblesses de la plateforme E-Santé Algérie	22
3.5	Analyse du portail CNAS / CASNOS	22
3.6	Évaluation de la plateforme DocteurDZ	23
3.7	Évaluation de la plateforme Clinido	23
3.8	Comparaison entre les principales plateformes de santé numérique en Algérie (vue verticale).....	24
4.1	Comparaison entre les trois vues UML principales	34

Abréviations

WWW : World Wide Web.

HTTP : Hypertext Transfer Protocol.

HTTPS : Hypertext Transfer Protocol Secure.

URL : Uniform Resource Locator.

P2P : Peer-to-Peer.

SSL : Secure Sockets Layer.

TLS : Transport Layer Security.

IP : Internet Protocol.

HTML : Hypertext Markup Language.

CSS : Cascading Style Sheets.

SEO : Search Engine Optimization.

JWT : JSON Web Token.

AWS : Amazon Web Services.

JSON : JavaScript Object Notation.

REST : Representational State Transfer.

API : Application Programming Interface.

NPM : Node Package Manager.

SGBDR : Système de Gestion de Base de Données Relationnelle.

SQL : Structured Query Language.

FCM : Firebase Cloud Messaging.

ONAT : Office National Algérien de Tourisme.

UML : Unified Modeling Language.

DSS : Diagramme de Séquence Système.

MCD : Modèle Conceptuel de Données.

MLD : Modèle Logique de Données.

SSR : Server-Side Rendering.

ORM : Object-Relational Mapping.

PHP : Hypertext Preprocessor.

UI : User Interface.

Chapitre 1

Introduction Générale

1.1 Contexte

La numérisation et l'adoption des services électroniques en Algérie sont devenues incontournables, en raison du besoin croissant de solutions numériques capables de remplacer les méthodes traditionnelles de gestion et de prestation des services. Cette transformation touche plusieurs secteurs tels que l'éducation, le transport, le logement et les services administratifs, et s'étend désormais au secteur de la santé, qui commence progressivement à adopter des approches digitales.

Cette transition vise à suivre le rythme du développement technologique mondial et à passer progressivement des méthodes classiques à des services numériques plus efficaces, accessibles à distance, et réduisant les coûts et les efforts.

En Algérie, la prise de conscience de l'importance des services de santé électroniques s'accroît, notamment grâce à une population jeune, dynamique et ouverte à l'innovation. Cependant, le domaine des soins de santé à domicile demeure encore sous-développé sur le plan numérique, ce qui crée un déséquilibre entre l'offre et la demande et limite l'efficacité des services médicaux en dehors des structures hospitalières.

C'est dans ce contexte que s'inscrit notre projet « MediCall », une plateforme innovante visant à numériser les services de soins infirmiers à domicile grâce à une application intelligente connectant directement les patients et les infirmiers de manière rapide, sécurisée et transparente. Le projet ambitionne d'améliorer la qualité des soins à domicile, de faciliter l'accès à des professionnels qualifiés, et de contribuer aux efforts nationaux pour promouvoir la transformation numérique du secteur de la santé en Algérie.

1.2 Problématique

L'Algérie connaît depuis quelques années une transformation progressive vers la digitalisation de ses services, notamment dans les secteurs de la santé, de l'éducation et des services à domicile. Malgré ces avancées, le domaine des soins infirmiers à domicile demeure encore peu structuré et faiblement numérisé. De nombreux patients nécessitant une assistance médicale à domicile rencontrent des difficultés à trouver des infirmiers disponibles, qualifiés et proches de leur localisation. De leur côté, les professionnels de santé font face à un manque de visibilité et à une absence de plateformes fiables leur permettant de proposer leurs services en toute sécurité. Les méthodes traditionnelles de mise en relation, telles que le bouche-à-oreille, les annonces locales ou les contacts directs, se révèlent limitées, chronophages et souvent inefficaces. Elles ne garantissent ni la fiabilité du prestataire, ni la sécurité du patient, ni la traçabilité des services rendus. Cette situation engendre une perte de temps, une méfiance mutuelle et, parfois, un renoncement total aux soins à domicile, particulièrement dans les zones rurales ou mal desservies.

Par ailleurs, les solutions numériques existantes dans le domaine médical en Algérie sont généralement centrées sur les rendez-vous médicaux en clinique ou les consultations à distance, sans prise en charge spécifique des services infirmiers à domicile. Elles manquent de fonctionnalités adaptées à ce contexte particulier : gestion des disponibilités, géolocalisation, notation des services, assistance automatisée, etc.

Il devient donc nécessaire de concevoir une plateforme numérique intelligente, simple, fiable et sécurisée, permettant de mettre en relation patients et infirmiers à domicile d'une manière fluide et transparente. Une telle solution offrirait une expérience utilisateur complète : recherche par spécialité et localisation, prise de rendez-vous instantanée, communication directe et suivi de la satisfaction.

Ainsi, MediCall vise à combler cette lacune en proposant une solution innovante basée sur les technologies de l'information et le soutien de l'intelligence artificielle, afin d'améliorer la qualité des soins à domicile et de moderniser le secteur de la santé en Algérie.

1.3 Objectifs

L'objectif principal de notre projet est de concevoir et de développer une plateforme numérique intelligente permettant de faciliter la mise en relation entre les patients et les infirmiers à domicile en Algérie. Cette plateforme servira d'intermédiaire fiable et sécurisé entre les profes-

sionnels de santé et les personnes ayant besoin de soins médicaux à domicile, tout en garantissant la simplicité, la rapidité et la transparence du processus.

Danc nous visons à offrir un espace centralisé où les patients pourront rechercher, contacter et réserver un infirmier selon sa spécialité, sa localisation et sa disponibilité, tandis que les infirmiers pourront gérer leurs horaires, leurs rendez-vous et interagir facilement avec leurs patients.

Pour atteindre cet objectif, nous avons développé une application web complétée par une application mobile multiplateforme basée sur le framework Flutter, afin de répondre aux besoins de tous les utilisateurs. L'application web offre des fonctionnalités complètes pour les administrateurs et les infirmiers (gestion des comptes, supervision, statistiques), tandis que la version mobile est pensée pour les patients et les infirmiers afin d'assurer une accessibilité optimale à tout moment et en tout lieu.

De plus, la plateforme intègre un assistant virtuel intelligent, développé à l'aide de technologies d'intelligence artificielle (IA) et de traitement du langage naturel (NLP), capable de répondre aux questions des utilisateurs, de proposer des recommandations personnalisées et d'améliorer l'expérience globale d'utilisation.

À travers ce projet, nous cherchons à améliorer la qualité des soins à domicile, à réduire les délais d'accès aux services infirmiers, et à moderniser le secteur de la santé en Algérie en créant un écosystème numérique fiable, dynamique et humain, au service des patients et des professionnels de santé.

1.4 Méthodologie et résultats

Pour concevoir et développer notre plateforme MediCall, dédiée à la mise en relation entre les patients et les infirmiers à domicile en Algérie, nous avons adopté une approche méthodique, structurée et orientée vers l'utilisateur. Cette méthodologie s'est articulée autour de plusieurs étapes essentielles, allant de l'analyse du domaine de la santé jusqu'à la mise en œuvre et l'évaluation du système final.

1.4.1 Étude du domaine et identification des besoins

Nous avons commencé par une étude approfondie du secteur de la santé et des services infirmiers à domicile en Algérie. Cette étape nous a permis de comprendre les besoins spécifiques des patients en matière de soins à domicile, ainsi que les difficultés rencontrées par les infirmiers

dans la gestion de leurs prestations (planning, déplacements, communication, etc.). Nous avons également analysé plusieurs solutions numériques existantes (nationales et internationales) pour identifier leurs forces et leurs limites. Cette étude comparative nous a permis de mettre en évidence une absence d'outils centralisés et intelligents adaptés au contexte algérien.

1.4.2 Analyse et spécification des besoins

Sur la base de ces observations, nous avons défini un cahier des charges fonctionnel et technique clair :

- **Pour les patients** : recherche d'un infirmier selon la localisation, la spécialité et la disponibilité, réservation en ligne, communication directe et évaluation du service ;
- **Pour les infirmiers et les medecins** : gestion du profil professionnel, planification des rendez-vous, consultation des demandes et gestion du calendrier ;
- **Pour l'administrateur** : supervision globale, gestion des utilisateurs, suivi statistique et contrôle qualité ;

Cette phase de spécification a permis d'établir les objectifs de développement de la plateforme et de structurer son architecture logicielle.

1.4.3 Conception du système

Nous avons ensuite procédé à la conception détaillée du système, en mettant l'accent sur la modularité, la sécurité et la scalabilité. Le système a été organisé autour de plusieurs modules :

Module de gestion des utilisateurs (patients, infirmiers, administrateurs).

Module de réservation avec planification intelligente et notifications automatiques.

Module d'évaluation et de feedback pour renforcer la confiance et la qualité du service.

Module d'assistance intelligente (IA) pour répondre aux besoins des utilisateurs et proposer des recommandations personnalisées.

La conception comprenait également la modélisation de la base de données (MongoDB) et la définition des interactions entre les composants via des API RESTful.

1.4.4 Choix technologiques et développement

Le choix des technologies a joué un rôle déterminant dans la qualité et la performance du système. Nous avons adopté une architecture basée sur :

Node.js et Express.js pour le back-end,

React.js et Flutter pour les interfaces web et mobile,
Python (FastAPI + TensorFlow) pour le module d'intelligence artificielle,
et MongoDB pour la gestion des données.

Le développement s'est effectué selon une approche itérative et incrémentale (méthode Agile), permettant d'intégrer progressivement les fonctionnalités et de les tester à chaque cycle. L'IA a été entraînée à comprendre les requêtes des utilisateurs et à fournir des réponses contextualisées, tout en s'améliorant continuellement grâce aux retours d'expérience.

1.4.5 Résultats obtenus

Grâce à cette méthodologie, nous avons abouti à la création d'une plateforme innovante, fiable et évolutive composée de :

- une application web complète pour la gestion et l'administration ;
- une application mobile multiplateforme pour les patients et les infirmiers ;
- un assistant virtuel intelligent intégré, offrant une aide instantanée et personnalisée.

Les tests effectués ont montré une stabilité et une rapidité remarquables :

Le temps moyen de réponse est inférieur à 2 secondes, et la satisfaction des utilisateurs pilotes dépasse 75 %.

Les infirmiers peuvent désormais gérer leurs rendez-vous et disponibilités en toute simplicité, tandis que les patients bénéficient d'un accès rapide et sécurisé aux soins à domicile.

Enfin, la flexibilité de l'architecture permet d'ajouter aisément de nouvelles fonctionnalités, telles que la géolocalisation en temps réel, les paiements électroniques sécurisés ou l'intégration avec des services médicaux externes.

En somme, la méthodologie adoptée a permis de transformer une idée innovante en un produit technologique concret et fonctionnel, contribuant à la modernisation du secteur de la santé en Algérie.

1.5 Structure du rapport

Afin de décrire de manière claire, cohérente et complète le travail réalisé dans le cadre de ce projet de fin d'études, ce mémoire regroupe l'ensemble des éléments nécessaires à la compréhension de notre démarche, depuis l'étude du domaine de la santé jusqu'à la conception, la réalisation et l'évaluation de notre solution.

Nous présentons dans ce manuscrit une analyse approfondie du contexte des soins à do-

micile en Algérie, une étude critique de l'existant, ainsi que les étapes méthodologiques suivies pour concevoir et développer la plateforme MediCall.

Sur cette base, la structure de notre mémoire est organisée comme suit :

- **Le chapitre 1** introduit le projet, son contexte général, la problématique, les objectifs, la justification du choix du sujet, ainsi que la méthodologie adoptée.
- **Le chapitre 2** fournit des informations générales relatives à notre travail.
- **Le chapitre 3** présente un état de l'art des solutions existantes, nationales et internationales, permettant la gestion ou la mise en relation entre patients et professionnels de santé. Ce chapitre inclut une analyse comparative de leurs avantages, limites et lacunes, tout en soulignant la nécessité d'une solution adaptée au contexte algérien.
- **Le chapitre 4** décrit la phase de modélisation et de conception du système, incluant les diagrammes UML, la modélisation de la base de données, l'architecture logicielle et les interactions principales entre les différents modules de MediCall.
- **Le chapitre 5** est consacré à la réalisation pratique de la solution. Il détaille les technologies choisies (Node.js, React, Flutter, Python, etc.), les outils utilisés, ainsi qu'une présentation des interfaces principales de l'application web et mobile. Ce chapitre inclut également la présentation du module intelligent (assistant virtuel basé sur l'IA) et son intégration dans le système global.

Enfin, le mémoire se termine par une conclusion générale, résumant les résultats obtenus, les difficultés rencontrées, et les perspectives d'évolution du projet. Des pistes d'amélioration futures y sont également proposées, notamment l'ajout de fonctionnalités avancées comme la géolocalisation en temps réel et les paiements sécurisés en ligne.

Chapitre 2

Fondements techniques du développement web et mobile

2.1 Introduction

Avec la démocratisation d'Internet, initialement créé à des fins militaires, le réseau mondial est devenu un pilier de la société moderne en facilitant l'échange rapide et sécurisé d'informations dans divers domaines. Le World Wide Web (WWW), basé sur les protocoles standards (URL, HTTP, client-serveur), a profondément transformé l'accès et l'interaction avec les ressources numériques. L'évolution du Web a conduit à l'émergence d'applications web dynamiques et mobiles intelligentes, favorisant l'interaction en temps réel avec des systèmes adaptatifs. Dans le cadre du projet, ces technologies sont exploitées pour développer une plateforme numérique intelligente dédiée aux soins infirmiers à domicile, combinant programmation web, intelligence artificielle et sécurité. Le chapitre présente les fondements technologiques du Web et des applications mobiles, les architectures, protocoles et langages de développement utilisés, ainsi que les atouts et limites des applications mobiles dans le domaine de la santé connectée.

2.2 Les bases du Web

L'émergence d'Internet, associée à la standardisation des protocoles et des technologies, a rendu possible et efficace la communication entre des machines réparties dans différentes zones géographiques. Cela a favorisé l'apparition de nombreux services et applications reposant sur ce réseau, tels que le Web, qui se présente comme un service d'accès et de partage de ressources à distance, fondé sur une architecture client-serveur et utilisant le protocole HTTP pour la

communication.

2.2.1 Réseau Internet

Un réseau informatique désigne un ensemble de machines interconnectées dans le but de communiquer et de partager des ressources matérielles et logicielles. Lorsque plusieurs de ces réseaux sont reliés entre eux, ils forment ce qu'on appelle Internet. Ce dernier constitue un réseau mondial, accessible au public, qui permet la circulation et la diffusion d'informations sous forme de textes, d'images, de sons et de vidéos. Il offre également la possibilité de partager des ressources physiques, telles que des unités de calcul, des serveurs de stockage et d'autres équipements, dans l'optique d'optimiser l'efficacité et la productivité [1].

2.2.2 Modèles de communication

Le modèle d'un réseau décrit la façon dont les différents nœuds communiquent entre eux. On distingue principalement deux types : le modèle de communication client-serveur et le modèle de communication poste à poste (peer-to-peer).

2.2.2.1 Modèle de communication Client-Serveur

Dans ce modèle, les machines du réseau sont réparties en deux catégories : les clients, qui disposent d'un système d'exploitation orienté utilisateur, et les serveurs, équipés d'un système d'exploitation spécialement conçu pour la gestion et la fourniture de services. La connexion y est centralisée : l'ordinateur client envoie une requête au serveur (par exemple via Internet dans le cas du Web), lequel traite l'opération demandée et renvoie la réponse appropriée. Les ressources des clients, quant à elles, ne sont pas partagées [2].

2.2.2.2 Modèle de communication poste à poste (P2P)

Dans le modèle de communication poste à poste (Peer-to-Peer), les machines du réseau fonctionnent de manière indépendante tout en jouant simultanément les rôles de client et de serveur. Ce type d'architecture est notamment utilisé pour le partage de fichiers et la distribution de contenus [3].

2.2.3 Protocole HTTP

Le HyperText Transfer Protocol (HTTP) est un protocole textuel dans lequel les messages sont échangés en clair et sans état, c'est-à-dire que chaque communication est indépendante des précédentes. Il a été conçu pour faciliter les échanges entre un client, généralement un navigateur web, et un serveur web. Cependant, la version classique de HTTP transmet les données en texte brut, sans aucun chiffrement. À l'inverse, sa version sécurisée, HTTPS, repose sur les technologies SSL et TLS afin de chiffrer les sessions et d'assurer la confidentialité ainsi que la sécurité des messages échangés [4].

2.2.4 adresses web

Les adresses web, appelées URL (Uniform Resource Locator), permettent d'identifier et de localiser des ressources sur Internet, qu'il s'agisse de pages web, d'images, de vidéos ou de fichiers. Une URL se compose généralement d'un protocole de communication (tel que HTTP ou HTTPS), du nom de domaine (ou de l'adresse IP) du serveur hébergeant la ressource, ainsi que d'un chemin menant à cette ressource spécifique. Les URL sont indispensables pour accéder aux contenus en ligne via un navigateur web ou tout autre client réseau [5].

2.2.5 Le Web

Le Web, ou World Wide Web en anglais, est l'une des applications les plus populaires d'Internet. Il désigne un ensemble de pages reliées entre elles par des hyperliens, également appelés liens hypertextes, accessibles via des adresses web à travers un navigateur. [6] Le fonctionnement du Web repose sur un ensemble de technologies, de protocoles et d'architectures spécifiques, parmi lesquels :

- le réseau Internet, qui constitue le support physique de communication ;
- l'architecture client-serveur, utilisée comme modèle d'interaction entre les machines ;
- l'architecture client-serveur, utilisée comme modèle d'interaction entre les machines ;
- le protocole HTTP, chargé d'assurer le transfert des données entre le client et le serveur ;
- l'URL, servant d'identifiant unique pour localiser et accéder aux ressources en ligne ;
- le langage HTML, qui définit la structure et le contenu des pages web.

2.3 Site Web statique & site Web dynamique

Un site web est un ensemble de pages reliées entre elles par des liens hypertextes, hébergées sur un serveur web et accessibles via une adresse web à travers un navigateur (client web). On distingue principalement deux types de sites : les sites web statiques et les sites web dynamiques [7].

2.3.1 Site web statique

Un site web statique est constitué de pages fixes affichant un contenu prédéfini aux visiteurs. Sa structure reste inchangée, offrant des informations sans mise à jour en temps réel ni interaction avec l'utilisateur. Conçus à l'aide de HTML et CSS, les sites statiques se caractérisent par leur simplicité et leur rapidité de chargement, mais ils ne proposent pas de fonctionnalités dynamiques. Ils sont particulièrement adaptés aux projets centrés sur la diffusion de contenu stable, sans besoin de modifications fréquentes ou de fonctionnalités avancées [8].

2.3.2 Site Web dynamique

Un site web dynamique est une plateforme en ligne capable de générer et d'adapter son contenu en fonction des interactions des utilisateurs, offrant ainsi des expériences personnalisées et actualisées en temps réel. Grâce à l'utilisation de langages de script côté serveur tels que PHP, Python ou JavaScript, ces sites permettent la mise à jour automatique du contenu à partir de bases de données, le traitement des saisies des utilisateurs ainsi que l'intégration de fonctionnalités interactives. Leur flexibilité et leur adaptabilité en font une solution privilégiée pour répondre aux besoins évolutifs des utilisateurs et offrir des expériences web riches et engageantes [9].

2.4 Programmation Web

2.4.1 Présentation

La programmation web désigne l'ensemble des compétences et techniques informatiques consacrées à la conception et au développement de sites et d'applications accessibles via Internet. Elle permet de créer des applications web composées de pages statiques ou dynamiques, et implique des aspects de programmation aussi bien du côté client que du côté serveur [10].

2.4.2 Programmation côté serveur (Back-End)

La programmation côté serveur, également appelée développement back-end, consiste à concevoir la logique et les fonctionnalités qui soutiennent le fonctionnement des applications web. Elle prend en charge la gestion des données, le traitement des requêtes des utilisateurs et la communication avec les bases de données. Orienté vers les opérations effectuées sur le serveur, le code back-end garantit des interactions efficaces entre l'interface utilisateur et l'infrastructure serveur, tout en assurant la diffusion d'un contenu dynamique et personnalisé, ainsi que la sécurité et l'intégrité des données [11].

2.4.3 Technologies connexes

Le développement web côté serveur s'appuie sur un large éventail de technologies, de services, d'outils et de frameworks permettant aux développeurs de concevoir et d'implémenter la logique ainsi que les fonctionnalités des systèmes. Parmi ces technologies [12], on peut citer :

- **Langages côté serveur** : Python, PHP ;
- **Bases de données** : MySQL, PostgreSQL, MongoDB pour le stockage et la gestion des données ;
- **Frameworks** : Firebase (Python), Laravel (PHP), Ruby on Rails (Ruby), Express (Node.js) ;
- **Environnements de serveur** : Apache, Nginx ;
- **API** : RESTful ou GraphQL, facilitant la communication entre les différents composants ;
- **Authentification** : OAuth, JSON Web Token (JWT) pour garantir un accès sécurisé aux utilisateurs ;
- **Conteneurisation** : Docker pour le déploiement et la mise à l'échelle des applications ;
- **Services Cloud** : AWS, Azure, Google Cloud pour un hébergement fiable et évolutif.

2.4.4 Programmation côté client (Front-End)

La programmation côté client, ou développement front-end, consiste à concevoir les éléments visuels et interactifs d'un site web ou d'une application avec lesquels l'utilisateur interagit directement. Elle se concentre sur la création d'interfaces utilisateur intuitives, ergonomiques et réactives, afin de garantir une expérience de navigation fluide et dynamique [13].

2.4.5 Technologies connexes

Le développement front-end s'appuie sur un ensemble varié d'outils, de langages et de frameworks, parmi lesquels [14] :

- **HTML/CSS** : bases essentielles pour structurer et styliser le contenu web ;
- **JavaScript** : langage de script permettant d'ajouter des éléments interactifs et dynamiques ;
- **Frameworks JavaScript** : React, Angular, Vue.js pour développer des applications modulaires et robustes ;
- **Frameworks CSS** : Bootstrap, Tailwind CSS pour concevoir des interfaces réactives et esthétiques ;
- **Gestionnaires de paquets** : npm (Node Package Manager) pour gérer les dépendances des projets ;
- **Outils de construction** : Webpack, Babel pour le regroupement, la transpilation et l'optimisation du code ;
- **API** : Fetch API, Axios pour assurer la communication avec les services back-end ;
- **Frameworks de tests** : Jest, Mocha afin de garantir la fiabilité et la qualité du code ;
- **Responsive Design** : Media Queries, Flexbox, CSS Grid pour concevoir des mises en page adaptatives et compatibles avec différents supports (ordinateurs, tablettes, smartphones).

2.5 Applications mobiles

2.5.1 Présentation

Les applications mobiles, communément appelées apps, sont des programmes logiciels conçus pour être exécutés sur des appareils mobiles tels que les smartphones et les tablettes. Elles fournissent des fonctionnalités ou des services spécifiques, allant de la communication et de la productivité au divertissement et aux outils utilitaires. Adaptées aux petits écrans et aux interfaces tactiles, les applications mobiles constituent un moyen pratique et accessible pour accéder aux contenus, interagir avec des services et exploiter pleinement les fonctionnalités des appareils mobiles [15].

2.5.2 Avantages

Les applications mobiles présentent de nombreux avantages, tant pour les utilisateurs que pour les développeurs. Parmi les plus importants, on peut citer :

- Un accès pratique et en mobilité aux services et aux informations ;
- La possibilité de personnaliser le contenu et les fonctionnalités en fonction des préférences des utilisateurs ;
- La disponibilité de certaines applications en mode hors ligne, même sans connexion Internet ;
- Une expérience utilisateur optimisée, spécialement conçue pour les appareils mobiles et garantissant une navigation fluide ;
- L'envoi de notifications instantanées pour informer des mises à jour, messages ou actualités pertinentes ;
- La rationalisation des tâches et des processus, contribuant ainsi à une meilleure productivité ;
- Le renforcement de l'engagement utilisateur grâce à des fonctionnalités interactives ;
- L'exploitation des capacités matérielles des appareils (caméra, GPS, capteurs, etc.) pour enrichir les fonctionnalités offertes ;
- Des opportunités économiques pour les entreprises et les particuliers, à travers la vente d'applications ou les achats intégrés.

2.5.3 Technologies et langages connexes

Cette section présente les principales technologies, langages de programmation et solutions couramment utilisés dans le développement mobile.

Développement mobile natif

1. **Android & iOS** : reposent sur des langages tels que Java, Kotlin (pour Android), Swift et Objective-C (pour iOS), ainsi que Dart pour certains environnements.

Développement multiplateforme

1. **React Native** : framework basé sur JavaScript et React, permettant de créer des applications compatibles avec plusieurs plateformes à partir d'un code unique.
2. **Flutter** : framework développé par Google, utilisant Dart pour construire des applications compilées nativement et offrant des performances proches des applications natives.

Intégration dorsale (Back-End Integration)

1. **API RESTful** : facilite la communication entre les applications mobiles et les serveurs grâce à des échanges standardisés.
2. **GraphQL** : langage de requête pour les API, permettant une récupération optimisée et flexible des données.

Gestion des bases de données

1. **MySQL** : système de gestion de base de données relationnelle (SGBDR) utilisant le langage SQL pour organiser et manipuler les données.
2. **SQLite** : base de données légère et embarquée, fréquemment utilisée dans les applications mobiles pour stocker localement les informations.

Notifications poussées

1. **Firebase Cloud Messaging (FCM)** : service permettant d'envoyer des notifications push vers les appareils mobiles afin d'informer ou de réengager les utilisateurs.

Analyse des applications mobiles

1. **Google Analytics pour mobile** : outil de suivi et d'analyse des interactions des utilisateurs, de la performance des applications et de l'engagement global.

Conception des interfaces graphiques

1. **Sketch, Figma, Adobe XD** : solutions de conception d'interfaces graphiques utilisées pour créer des maquettes et des prototypes d'applications mobiles intuitives et ergonomiques.

Déploiement des applications

1. **App Store (iOS)** : plateforme officielle d'Apple pour la distribution des applications iOS.
2. **Google Play (Android)** : plateforme officielle de Google pour la distribution des applications Android.

2.6 Applications de bureau

2.6.1 Présentation

Les applications de bureau, également appelées logiciels de bureau ou *desktop applications*, sont des programmes informatiques conçus pour être installés et exécutés localement sur un

ordinateur personnel, qu'il soit sous Windows, macOS ou Linux. Contrairement aux applications web ou mobiles, elles ne nécessitent pas de connexion Internet permanente pour fonctionner et exploitent pleinement les ressources matérielles de la machine hôte (processeur, mémoire, disque dur, carte graphique, etc.). Les applications de bureau couvrent une large gamme de besoins : bureautique, gestion, conception graphique, développement logiciel, jeux vidéo, et bien d'autres encore. Leur stabilité, leurs performances élevées et leurs capacités hors ligne en font un choix privilégié pour les environnements professionnels et techniques [16].

2.6.2 Avantages

Les applications de bureau présentent plusieurs avantages significatifs par rapport aux applications web ou mobiles. Parmi ceux-ci :

- **Performance élevée** : elles exploitent directement les ressources matérielles de l'ordinateur, ce qui garantit une exécution rapide et fluide, même pour des tâches intensives (modélisation 3D, traitement d'images, calculs scientifiques, etc.) ;
- **Fonctionnement hors ligne** : elles peuvent être utilisées sans connexion Internet, ce qui les rend adaptées aux environnements à connectivité limitée ;
- **Sécurité et confidentialité** : les données peuvent être stockées localement, limitant ainsi les risques liés aux transferts en ligne ;
- **Interface riche et personnalisable** : elles permettent des interactions complexes avec l'utilisateur, grâce à des interfaces graphiques avancées et des composants ergonomiques ;
- **Intégration système** : elles peuvent interagir directement avec le système d'exploitation (accès aux fichiers, périphériques, etc.) pour offrir des fonctionnalités complètes et performantes ;
- **Mises à jour contrôlées** : contrairement aux applications web, les versions des logiciels de bureau sont gérées localement, offrant plus de stabilité et de contrôle aux utilisateurs [17].

2.6.3 Technologies et langages utilisés

Le développement des applications de bureau repose sur une variété de langages de programmation, de frameworks et d'outils selon le système d'exploitation ciblé et les besoins du projet.

Langages de programmation principaux :

1. **C/C++** : largement utilisés pour les applications nécessitant des performances élevées (logiciels scientifiques, jeux vidéo, outils système) ;
2. **C#** : employé dans l'écosystème Microsoft via le framework .NET pour créer des applications Windows modernes ;
3. **Java** : langage multiplateforme fonctionnant sur la machine virtuelle Java (JVM), permettant le développement d'applications compatibles avec plusieurs systèmes ;
4. **Python** : grâce à des bibliothèques comme Tkinter, PyQt ou wxPython, il facilite la création rapide d'interfaces graphiques multiplateformes ;
5. **Electron (JavaScript)** : framework open-source basé sur Node.js et Chromium, permettant de créer des applications de bureau en utilisant les technologies web [18].

Frameworks et environnements de développement :

1. **.NET / WPF / WinUI** : technologies Microsoft pour le développement d'applications Windows modernes, performantes et esthétiques ;
2. **Qt** : framework multiplateforme en C++ (et bindings pour Python) utilisé pour le développement d'interfaces graphiques complexes et robustes ;
3. **JavaFX et Swing** : bibliothèques Java pour la conception d'interfaces utilisateurs dynamiques ;
4. **GTK+** : utilisé notamment dans les environnements Linux pour la création d'interfaces graphiques légères ;
5. **Electron** : permet de créer des applications de bureau à partir d'un seul code JavaScript compatible avec Windows, macOS et Linux [19].

2.6.4 Architecture et fonctionnement

Une application de bureau suit généralement une architecture logicielle structurée autour du modèle MVC (Modèle–Vue–Contrôleur) ou MVVM (Modèle–Vue–VueModèle). Cette organisation permet de séparer la logique métier, la gestion des données et la présentation graphique. Les composants principaux sont :

- **L'interface graphique (GUI)** : gérée via des bibliothèques dédiées (Qt, WPF, Tkinter, JavaFX, etc.) ;
- **Le noyau applicatif** : partie contenant la logique de traitement et les règles métier ;
- **La couche de données** : responsable du stockage local ou distant (SQLite, MySQL, PostgreSQL, fichiers JSON, etc.) ;

- **Les services externes** : parfois intégrés pour la synchronisation, la sauvegarde ou la communication réseau (API REST, GraphQL) [20].

2.6.5 Déploiement et distribution

Le déploiement des applications de bureau varie selon la plateforme :

- **Windows** : les applications sont généralement distribuées sous forme d'exécutables (.exe) ou de packages MSI installables via le Microsoft Store ou des sites officiels ;
- **macOS** : elles sont fournies sous forme de paquets (.app) signés et distribués via le Mac App Store ou des téléchargements directs ;
- **Linux** : la distribution s'effectue à travers des gestionnaires de paquets (APT, Snap, Flatpak) ou des archives (.deb, .rpm, AppImage) [21].

2.6.6 Applications hybrides et évolutions récentes

Avec la convergence des technologies, de nouvelles approches permettent de développer des applications dites hybrides, combinant les avantages du web et du bureau. Des frameworks comme Electron ou Tauri offrent la possibilité de concevoir des logiciels multiplateformes reposant sur des technologies web tout en tirant parti des capacités locales de l'ordinateur. De plus, l'intégration de l'intelligence artificielle et du cloud computing renforce les capacités des applications de bureau modernes, leur permettant de s'adapter aux besoins des utilisateurs, d'analyser les comportements et d'offrir des services intelligents en temps réel [22].

Chapitre 3

Étude de l'existant et spécification des besoins

3.1 Introduction

L'établissement des spécifications détaillées de notre projet MediCall, après une identification précise des besoins du marché dans le domaine de la santé numérique et plus particulièrement des soins infirmiers à domicile, constitue une étape cruciale dans la réussite de notre travail. La compréhension approfondie des exigences du secteur médical, des lacunes existantes et des solutions numériques déjà disponibles nous permet d'obtenir une vision globale et réaliste avant d'entamer le processus de conception et de développement de notre plateforme intelligente.

Dans le contexte actuel, marqué par la digitalisation croissante du secteur de la santé en Algérie, le besoin d'une solution technologique fiable reliant efficacement les patients et les infirmiers à domicile se fait de plus en plus ressentir [23].

L'analyse de la situation actuelle et l'étude des solutions existantes, tant au niveau national qu'international, nous offrent une base solide pour concevoir un système innovant, sécurisé et adapté au contexte socio-économique algérien [24].

Ce chapitre présente donc une description approfondie de notre domaine d'étude, en abordant d'abord la situation actuelle des soins à domicile et les différents services numériques liés à la santé. Nous examinerons ensuite les solutions existantes afin d'identifier les lacunes et les opportunités d'amélioration. Sur cette base, nous définirons les objectifs généraux et spécifiques, ainsi que le marché cible et les utilisateurs potentiels de notre solution.

Enfin, nous détaillerons la spécification des besoins fonctionnels et non fonctionnels du

système, tout en décrivant la méthodologie de travail adoptée pour la conception, la gestion et la mise en œuvre du projet. Ce chapitre se conclura par une synthèse globale mettant en évidence la cohérence entre les besoins identifiés et la solution proposée.

3.2 Description de notre domaine d'étude

L'industrie des soins de santé à domicile et des services infirmiers connectés constitue le cœur de notre domaine d'étude. Ce secteur, en pleine transformation numérique, joue un rôle crucial dans l'amélioration de la qualité de vie des patients, la réduction de la surcharge hospitalière et la facilitation de l'accès aux soins pour les populations éloignées [25].

En Algérie, le besoin de solutions technologiques permettant la coordination entre les patients, les infirmiers et les établissements de santé est devenu une priorité, particulièrement avec la montée des maladies chroniques et le vieillissement de la population [26]. MediCall s'inscrit dans cette dynamique en proposant une plateforme intelligente et interactive dédiée à la gestion et à la prestation des soins infirmiers à domicile.

Ce chapitre présente une analyse détaillée du domaine des soins à domicile en Algérie, de la scène des services infirmiers, ainsi que des expériences et initiatives numériques émergentes dans ce secteur.

3.2.1 Soins de santé à domicile en Algérie

Les soins de santé à domicile (SSD) consistent à fournir aux patients des services médicaux et paramédicaux directement dans leur lieu de vie. Ce modèle, largement adopté dans les pays développés, vise à assurer une continuité thérapeutique et à réduire les hospitalisations inutiles [27].

En Algérie, ce type de service reste encore limité et souvent informel. Les familles font appel à des infirmiers indépendants, sans outils de vérification, de planification ou de suivi. Ce manque de coordination engendre des problèmes de qualité, de sécurité et de fiabilité.

3.2.2 Scène des services infirmiers en Algérie

Les infirmiers jouent un rôle central dans le système de santé. En Algérie, ils représentent plus de 40% du personnel paramédical [28], mais leurs activités à domicile restent faiblement structurées. L'absence d'un système numérique unifié rend difficile la mise en relation entre l'offre et la demande, la traçabilité des interventions et la valorisation du métier infirmier.

Critère	Soins traditionnels	Soins numériques (Need Nursing)
Mode de contact	Direct (recommandation, bouche-à-oreille)	Application mobile/web sécurisée
Planification	Non standardisée	Gestion automatisée des rendez-vous
Suivi du patient	Manuel, via téléphone	Tableau de bord intelligent et notifications
Sécurité et vérification	Absence de contrôle	Vérification des infirmiers et certification
Communication	Limitée	Messagerie sécurisée entre patient et infirmier
Support IA	Aucun	Assistant intelligent d'aide à la décision

TABLE 3.1 – Différences entre les soins traditionnels et numériques à domicile (Projet Need Nursing)

Notre solution Need Nursing vise à moderniser cette relation en fournissant :

- un espace de profil professionnel vérifié pour les infirmiers ;
- un système de réservation clair pour les patients ;
- une base de données médicale centralisée et sécurisée.

Catégorie de soin	Exemples d'intervention
Soins de base	Surveillance des constantes, prise de tension, glycémie
Soins post-opératoires	Pansements, suivi cicatriciel, injections
Soins palliatifs	Accompagnement des patients en fin de vie
Soins spécifiques	Perfusions, assistance respiratoire, soins diabétiques
Prévention et éducation	Conseils hygiéno-diététiques, suivi de traitement

TABLE 3.2 – Catégorisation des soins infirmiers à domicile

3.2.3 Expériences numériques et innovations en santé

Avec l'avènement de l'intelligence artificielle et de l'Internet des objets (IoT), les soins à domicile connaissent une véritable révolution. Les applications mobiles permettent désormais :

- la télésurveillance des signes vitaux ;
- la communication directe patient–infirmier ;
- la planification automatique des soins [29].

En Algérie, bien que ces solutions soient encore rares, plusieurs initiatives publiques et privées émergent dans le domaine de la e-santé. Le projet MediCall vient renforcer cette tendance en intégrant :

- un assistant virtuel intelligent ;
- des mécanismes d'aide à la décision basés sur le profil du patient ;
- des interfaces adaptées aux différents acteurs de santé.

Technologie	Rôle
Intelligence Artificielle (IA)	Orientation automatique des patients et support décisionnel
Cloud Computing	Stockage sécurisé des données médicales
Application mobile (Flutter)	Accès facile aux services et géolocalisation
Plateforme web (React / Node.js)	Interface de gestion pour les infirmiers et administrateurs
Système de notification	Suivi des rendez-vous et alertes médicales

TABLE 3.3 – Intégration des technologies émergentes dans la plateforme MediCall

3.3 État de l'existant

Aujourd'hui en Algérie, le domaine de la santé numérique connaît une expansion rapide, soutenue par la demande croissante de solutions permettant la téléconsultation, le suivi à distance et les soins à domicile. Cependant, malgré ces avancées, la mise en relation directe entre patients et infirmiers à domicile demeure limitée et souvent informelle. Les patients rencontrent encore des difficultés à identifier des prestataires fiables, tandis que les professionnels de santé manquent de plateformes centralisées pour promouvoir leurs services.

Cette section analyse l'état actuel des solutions numériques dans le domaine de la santé en Algérie, en distinguant deux grandes catégories :

- Les solutions mises en place par l'État ;
- Les solutions développées par des acteurs privés.

3.3.1 Solutions mises en place par l'État

L'État algérien a lancé plusieurs initiatives dans le domaine de la santé numérique, afin de moderniser l'accès aux soins et d'améliorer la qualité des services. Parmi les solutions notables, on retrouve les plateformes suivantes :

3.3.1.1 E-Santé Algérie

Lancée par le *Ministère de la Santé* en 2022, la plateforme E-Santé Algérie vise à offrir un portail d'information médicale pour les citoyens et les professionnels. Elle permet la consultation des établissements de santé, l'accès à des documents de prévention et le téléchargement de formulaires administratifs. Toutefois, elle ne propose pas encore de fonctionnalités de réservation ou de contact direct avec les prestataires de soins à domicile.

Avantages	Faiblesses
Plateforme officielle soutenue par le Ministère de la Santé.	Absence de système de réservation ou d'intermédiation patient-infirmier.
Fournit des informations fiables sur les établissements médicaux.	Pas d'application mobile dédiée.
Accès gratuit et sécurisé.	Manque d'interactivité et d'ergonomie.

TABLE 3.4 – Avantages et faiblesses de la plateforme E-Santé Algérie

3.3.1.2 Portail CNAS / CASNOS

Les portails numériques de la **CNAS** et de la **CASNOS** facilitent l'accès aux services administratifs pour les assurés sociaux et les professionnels de santé. Ces plateformes ont permis une digitalisation partielle du système de santé, mais elles restent centrées sur la gestion des cotisations et des remboursements, sans prise en charge des besoins de coordination médicale à domicile.

Avantages	Faiblesses
Simplifie les démarches administratives pour les assurés.	Aucune intégration avec les prestataires de soins.
Portail fiable et sécurisé.	Ne concerne pas directement les soins à domicile.
Accessibilité nationale.	Interface peu intuitive pour les usagers non technophiles.

TABLE 3.5 – Analyse du portail CNAS / CASNOS

3.3.2 Solutions privées et initiatives numériques

Face aux limites des plateformes publiques, plusieurs acteurs privés ont développé des solutions alternatives dans le domaine de la santé numérique.

3.3.2.1 DocteurDZ

DocteurDZ est une plateforme algérienne permettant la prise de rendez-vous en ligne avec des médecins généralistes et spécialistes. Elle offre une base de données de praticiens et un service de messagerie intégré, mais ne couvre pas encore le domaine du suivi infirmier à domicile.

Avantages	Faiblesses
Répertoire complet de médecins par spécialité.	Absence de services infirmiers à domicile.
Possibilité de réservation de rendez-vous en ligne.	Interface principalement orientée vers les cliniques.
Fonctionnalité d'avis et de notation des praticiens.	Non intégrée avec les structures publiques de santé.

TABLE 3.6 – Évaluation de la plateforme DocteurDZ

3.3.2.2 Clinido

Clinido est une autre application privée qui simplifie la recherche de médecins et la gestion des rendez-vous. Bien que performante, elle ne prend pas encore en charge la géolocalisation des infirmiers ou la planification des soins à domicile.

Avantages	Faiblesses
Application mobile fluide et ergonomique.	Absence de module de soins à domicile.
Réservation rapide et notifications automatiques.	Couvre principalement les grandes villes.
Paiement électronique en test.	Manque d'intégration avec d'autres services médicaux.

TABLE 3.7 – Évaluation de la plateforme Clinido

3.3.3 Évaluation comparative des solutions existantes

Afin de mieux évaluer les forces et faiblesses des plateformes analysées, le tableau suivant présente une synthèse comparative basée sur plusieurs critères fonctionnels.

3.3.4 Lacunes du marché

L'analyse des plateformes existantes révèle plusieurs lacunes dans l'écosystème de la santé numérique en Algérie :

Plateforme	E-Santé	CNAS/CASNOS	DocteurDZ	Clinido
Promotion des services de santé	✓	✓	✓	✓
Soutien aux soins à domicile	✗	✗	✗	✗
Réservation en ligne	✗	✗	✓	✓
Paiement électronique	✗	✗	✗	En phase de test
Application mobile dédiée	✗	✗	✓	✓
Système d'évaluation des utilisateurs	✗	✗	✓	✓
Géolocalisation	✗	✗	✗	✓
Intégration avec le secteur public	✓	✓	✗	✗

TABLE 3.8 – Comparaison entre les principales plateformes de santé numérique en Algérie (vue verticale)

- Absence d'une plateforme nationale dédiée aux soins infirmiers à domicile ;
- Manque d'intégration entre les solutions publiques et privées ;
- Faible adoption des technologies d'intelligence artificielle dans le suivi patient ;
- Insuffisance de sécurité des paiements en ligne ;
- Manque d'interfaces adaptées aux utilisateurs non technophiles.

Ces constats renforcent la pertinence du projet, qui vise à combler ces lacunes en offrant une plateforme unifiée, sécurisée et intelligente pour la gestion des soins infirmiers à domicile.

3.3.5 Synthèse

L'état de l'existant met en évidence un écart technologique important entre les besoins réels des patients et les solutions disponibles sur le marché algérien. MediCall se positionne comme une solution innovante, centrée sur l'utilisateur, et alignée avec la stratégie nationale de transformation numérique du secteur de la santé.

3.4 Objectifs du projet

3.4.1 Objectif global

L'objectif principal est de concevoir et de développer une plateforme numérique intelligente dédiée à la mise en relation entre les patients nécessitant des soins à domicile et les professionnels de santé, notamment les infirmiers et infirmières.

Cette solution vise à faciliter l'accès rapide, sécurisé et personnalisé aux services de soins à domicile, tout en optimisant la gestion des interventions, la planification et la communication entre les deux parties.

Notre ambition est de créer un écosystème intégré qui contribue à l'amélioration de la qualité des soins, à la digitalisation du secteur de la santé et à la promotion de la santé connectée en Algérie.

3.4.2 Objectifs spécifiques

Pour atteindre cet objectif global, plusieurs objectifs spécifiques ont été définis :

1. Concevoir une interface utilisateur intuitive et ergonomique, adaptée à différents profils (patients, infirmiers, doctores et administrateurs), permettant une utilisation simple et fluide.
2. Développer un système de recherche avancé permettant aux patients de trouver rapidement un professionnel de santé disponible selon leur localisation, leurs besoins médicaux et leurs préférences.
3. Mettre en place un système de réservation en ligne sécurisé, offrant la possibilité de planifier, confirmer et suivre les rendez-vous de soins à domicile en temps réel.
4. Fournir un tableau de bord pour les infirmiers, leur permettant de gérer leurs disponibilités, leurs rendez-vous, leurs statistiques d'activité et les évaluations reçues.
5. Intégrer un module de communication instantanée facilitant les échanges entre les patients et les prestataires de soins, et garantissant un suivi transparent.
6. Assurer la sécurité et la confidentialité des données médicales en conformité avec les normes de la protection des données de santé.
7. Développer un système d'évaluation et de commentaires permettant aux patients d'évaluer la qualité des services reçus et d'aider les autres utilisateurs dans leurs choix.

8. Mettre en œuvre une intelligence artificielle d'assistance virtuelle capable de recommander les profils de soins adaptés et d'aider les patients à décrire leurs besoins médicaux.
9. Créer un espace d'accompagnement et d'éducation à la santé, où les utilisateurs peuvent consulter des conseils, des guides et des informations fiables sur la prévention et les soins à domicile.
10. Promouvoir la plateforme via des stratégies de marketing numérique afin d'accroître sa visibilité, encourager l'adoption de la santé numérique et créer un réseau national de soins connectés.

3.5 Marché et utilisateurs visés

Le marché de la santé numérique en Algérie est en pleine expansion, soutenu par une demande croissante pour les services de soins à domicile, la télémédecine et les solutions connectées de suivi médical. Dans ce contexte, MediCall s'adresse à un ensemble diversifié d'utilisateurs ayant des besoins complémentaires dans le domaine des soins de santé personnalisés et de proximité.

Nos utilisateurs cibles comprennent :

- **Patients nécessitant des soins à domicile** : personnes âgées, malades chroniques, personnes en convalescence après une opération, ou souffrant de mobilité réduite. Ces utilisateurs recherchent un service fiable, rapide et sécuritaire pour bénéficier d'une assistance médicale ou paramédicale à domicile.
- **Infirmiers et infirmières indépendants** : professionnels de santé qualifiés souhaitant proposer leurs services de soins à domicile, élargir leur réseau professionnel et gérer leurs rendez-vous via une plateforme numérique centralisée.
- **Médecins et spécialistes** : souhaitant recommander des infirmiers à leurs patients pour le suivi des soins post-hospitaliers ou des traitements chroniques à domicile.
- **Familles et aidants** : proches des patients qui recherchent un outil leur permettant de planifier et suivre les interventions infirmières, tout en assurant la continuité et la qualité du suivi médical.
- **Cliniques, hôpitaux et établissements de santé privés ou publics** : institutions cherchant à déléguer ou coordonner certaines prestations de soins à domicile, notamment dans le cadre de partenariats.

- **Organismes d'assurance et de mutuelles santé** : intéressés par l'intégration de la plateforme dans leurs services afin de simplifier la prise en charge et le remboursement des soins à domicile.
- **Étudiants en santé et stagiaires infirmiers** : à la recherche d'opportunités de stage, d'expérience pratique ou de missions temporaires dans le domaine des soins infirmiers.
- **Administrateurs de la santé publique** : acteurs institutionnels souhaitant exploiter les données anonymisées pour analyser les besoins régionaux et planifier les politiques de santé.

Le marché de Need Nursing est donc hybride, combinant à la fois le secteur médical, le secteur social et le secteur technologique, avec un fort potentiel de croissance grâce à la digitalisation progressive du système de santé en Algérie.

3.6 Spécifications des besoins

3.6.1 Présentation

La phase de spécification des besoins constitue une étape cruciale dans le développement. Elle vise à identifier, décrire et organiser l'ensemble des exigences nécessaires pour garantir la conformité de la solution avec les attentes des utilisateurs finaux, notamment les patients, les infirmiers et les administrateurs.

Les besoins sont généralement divisés en deux catégories principales :

- **Besoins fonctionnels** : concernent les services et fonctionnalités que la plateforme doit offrir.
- **Besoins non-fonctionnels** : englobent les qualités générales du système telles que la performance, la fiabilité, la sécurité et l'ergonomie.

Cette section présente une description détaillée des acteurs impliqués, des besoins fonctionnels associés à chaque rôle, ainsi que des exigences non-fonctionnelles garantissant la qualité globale du système.

3.6.2 Acteurs

La plateforme MediCall est conçue pour être utilisée par plusieurs catégories d'acteurs interdépendants, chacun ayant un rôle spécifique :

- **Visiteur** : utilisateur non inscrit pouvant consulter la plateforme, rechercher des infirmiers disponibles, lire des informations sur les services proposés et accéder aux conseils de santé publique sans interaction directe.
- **Patient** : utilisateur enregistré qui peut créer un compte personnel, consulter les profils des infirmiers, effectuer des réservations, suivre ses rendez-vous et évaluer les prestations reçues.
- **Infirmier / doctore** : professionnel de santé inscrit sur la plateforme souhaitant proposer ses services à domicile. Il peut créer et gérer son profil, définir ses disponibilités, accepter ou refuser des demandes, consulter son historique de missions et recevoir des évaluations.
- **Administrateur** : responsable du contrôle et du bon fonctionnement de la plateforme. Il supervise les inscriptions des professionnels, valide les profils, gère les plaintes, analyse les statistiques d'utilisation et veille à la sécurité des données.

3.6.3 Besoins fonctionnels

La plateforme MediCall doit offrir un ensemble complet de fonctionnalités, réparties selon le rôle de l'utilisateur.

1. Fonctionnalités pour les visiteurs

- Consultation libre du site et des informations disponibles.
- Recherche d'infirmiers ou de services de soins selon la localisation.
- Accès à des articles éducatifs et à des guides de santé.
- Possibilité de visualiser les profils publics des professionnels.
- Affichage des zones de couverture via une carte interactive.

2. Fonctionnalités pour les patients

- Création et gestion d'un compte utilisateur.
- Authentification sécurisée et gestion du mot de passe.
- Consultation des profils infirmiers avec leurs évaluations et disponibilités.
- Réservation en ligne d'un soin à domicile avec choix de la date et de l'heure.
- Paiement en ligne sécurisé via une passerelle locale (EDAHABIA, CIB, etc.).
- Accès à l'historique des prestations et factures.

- Possibilité d'annuler ou de modifier une réservation.
- Système d'évaluation et d'avis sur les infirmiers.
- Réception de notifications automatiques pour les rappels et confirmations.

3. Fonctionnalités pour les infirmiers les medcins

- Création et personnalisation du profil professionnel (photo, diplômes, spécialités, zone de service).
- Gestion des disponibilités et du calendrier d'interventions.
- Consultation et gestion des demandes reçues.
- Suivi des rendez-vous, validation des visites et facturation automatique.
- Accès à un tableau de bord de statistiques (revenus, missions terminées, évaluations).
- Communication sécurisée avec les patients via une messagerie intégrée.

4. Fonctionnalités pour les administrateurs

- Gestion centralisée des comptes utilisateurs (patients et infirmiers).
- Validation et vérification des identités professionnelles.
- Supervision du système de paiements et des transactions.
- Gestion des signalements et des réclamations.
- Génération de rapports statistiques sur l'utilisation, les zones de forte demande et la qualité des services.
- Possibilité de suspendre ou supprimer un compte en cas de non-conformité.

5. Application mobile

L'application mobile, disponible sur Android et iOS, reprendra toutes les fonctionnalités précédentes avec des ajouts spécifiques :

- Notifications push (rappels, nouvelles offres, messages).
- Accès hors ligne aux informations récentes.
- Géolocalisation avancée pour trouver les infirmiers proches.
- Synchronisation automatique avec la version web.

3.6.4 Besoins non-fonctionnels

1. Performance

- Temps de réponse inférieur à 2 secondes pour les requêtes standards.
- Capacité de supporter un grand nombre d'utilisateurs simultanés sans dégradation du service.
- Disponibilité continue (24h/24, 7j/7) avec un taux de disponibilité supérieur à 99%.

2. Sécurité et confidentialité

- Chiffrement des communications via le protocole HTTPS.
- Stockage sécurisé des données médicales conformément aux normes locales.
- Sauvegardes automatiques et régulières.
- Authentification à deux facteurs pour les professionnels.

3. Ergonomie et accessibilité

- Interface intuitive, claire et cohérente sur tous les supports.
- Compatibilité multi-écrans (ordinateur, tablette, mobile).
- Accessibilité pour les utilisateurs malvoyants (lecture d'écran).

4. Maintenabilité et évolutivité

- Architecture modulaire facilitant l'ajout de nouvelles fonctionnalités.
- Documentation technique complète et mise à jour.
- Code structuré pour faciliter les mises à jour et la maintenance.

3.7 Conclusion

En conclusion, ce troisième chapitre a permis de dresser une vision globale et approfondie, en définissant les bases conceptuelles, fonctionnelles et techniques nécessaires à sa mise en œuvre.

À travers l'analyse du marché de la santé en Algérie, l'étude des solutions existantes et la spécification des besoins fonctionnels et non-fonctionnels, nous avons mis en évidence les

limites actuelles des systèmes de soins à domicile et identifié les opportunités offertes par la digitalisation du secteur.

Ce travail préparatoire a permis de clarifier les besoins des différents acteurs « patients, infirmiers, familles et administrateurs » ainsi que leurs interactions au sein de la plateforme. L'approche adoptée s'est appuyée sur une compréhension réaliste du contexte sanitaire algérien, des contraintes technologiques et des exigences de sécurité propres aux données médicales.

Ainsi, les spécifications détaillées établies dans ce chapitre constituent une feuille de route précise pour la conception du système. Elles guideront les étapes suivantes du projet, en particulier la modélisation des composants, la conception de la base de données, et la définition de l'architecture logicielle globale de la plateforme.

Le chapitre suivant sera donc consacré à la phase de modélisation et conception du système, où nous traduirons ces besoins en représentations UML, en processus interactifs et en structures logiques cohérentes permettant de concrétiser le fonctionnement complet.

Chapitre 4

La modélisation – UML

4.1 Introduction

Dans ce quatrième chapitre, nous abordons la phase cruciale de la modélisation et de la conception du système MediCall, qui constitue le pont entre l'analyse des besoins et la mise en œuvre technique du projet. Cette étape vise à traduire les exigences fonctionnelles et non-fonctionnelles identifiées précédemment en représentations conceptuelles et structurelles cohérentes, permettant de visualiser la logique interne du système avant son développement effectif.

Pour ce faire, nous adoptons le langage de modélisation unifié (UML – Unified Modeling Language), un standard internationalement reconnu pour la conception orientée objet [29, 30]. L'UML offre un ensemble de diagrammes permettant de représenter les différents aspects d'un système : son comportement, ses interactions, sa structure et la dynamique de ses composants. Grâce à cette approche, nous pouvons décomposer le système en entités bien définies, identifier leurs relations et formaliser les scénarios d'utilisation typiques des patients, infirmiers et administrateurs.

Ce chapitre présentera les principaux diagrammes UML utilisés pour la modélisation du système, notamment :

- le diagramme de cas d'utilisation pour illustrer les interactions entre les acteurs et le système ;
- le diagramme de classes pour décrire la structure logique du modèle de données ;
- les diagrammes de séquence et d'activité pour visualiser la dynamique et le flux des processus internes.

Chaque représentation graphique jouera un rôle essentiel dans la conception du système,

en assurant la cohérence, la maintenabilité et la robustesse, dédiée à la gestion intelligente des soins à domicile en Algérie.

4.2 Modélisation UML

4.2.1 Définition d'un modèle

Un modèle est une représentation abstraite et simplifiée d'un phénomène, d'un processus ou d'un système du monde réel, créée dans le but de le comprendre, l'analyser ou le prévoir [31]. Selon Booch, Rumbaugh et Jacobson, un modèle permet de réduire la complexité d'un système en omettant les détails secondaires et en mettant en évidence les éléments essentiels nécessaires à la compréhension et à la prédiction de son comportement [32]. Dans le cadre du MediCall, le modèle joue un rôle fondamental dans la représentation de la structure, du comportement et des interactions du système, facilitant ainsi la communication entre les acteurs du projet (patients, infirmiers, administrateurs, développeurs, etc.).

4.2.2 Définition de la modélisation

La modélisation est un processus méthodologique qui vise à traduire un système réel ou conceptuel en une représentation abstraite et structurée afin de mieux en comprendre la logique et d'en faciliter la conception [33]. D'après la spécification officielle de l'*Object Management Group (OMG)*, la modélisation dans l'ingénierie logicielle constitue une étape préalable essentielle à la création d'un système orienté objet, car elle permet de visualiser les interactions entre composants, d'assurer la cohérence des fonctionnalités et de simplifier la maintenance du logiciel. Dans MediCall, elle permet de représenter les relations entre les acteurs principaux et de décrire les processus essentiels (réservation, gestion des soins, communication, etc.).

4.2.3 Les perspectives de modélisation

La modélisation d'un système logiciel peut être abordée sous plusieurs angles complémentaires, appelés vues UML. Cette approche multi-perspective, telle que décrite par Booch et ses collaborateurs [34], permet d'obtenir une compréhension complète et cohérente d'un système complexe comme MediCall.

- **Vue structurelle (ou statique)** : Cette vue met en avant la composition du système et les relations entre ses entités. Elle est représentée par des diagrammes de classes et de

composants permettant de visualiser l'architecture interne du système [35].

- **Vue comportementale (ou dynamique)** : Définie dans la spécification UML 2.5.1, cette vue se concentre sur la manière dont les objets interagissent et réagissent dans le temps. Elle est généralement représentée par des diagrammes de séquence et d'activité [36].
- **Vue fonctionnelle (ou logique)** : Elle illustre les principales fonctions du système et leur enchaînement logique [37]. Dans le cadre du projet, cette vue permet d'optimiser le parcours utilisateur depuis la demande de service jusqu'à la validation du soin.

Aspect	Vue structurelle	Vue comportementale	Vue fonctionnelle
Objectif principal	Décrire la structure statique du système et les relations entre les entités.	Illustrer la dynamique du système et l'enchaînement des interactions.	Définir les fonctions et services du système selon les besoins des utilisateurs.
Diagrammes associés	Diagramme de classes, diagramme de composants, diagramme d'objets.	Diagramme de séquence, diagramme d'activité, diagramme d'états.	Diagramme de cas d'utilisation, diagramme de flux de données.
Éléments représentés	Classes, attributs, relations, associations, dépendances.	Acteurs, objets, messages, transitions, états.	Fonctions principales, acteurs, scénarios d'utilisation.
Utilité pour Medi-Call	Concevoir la base de données et la structure des modules du système.	Décrire la communication entre les patients, les infirmiers et l'assistant intelligent.	Identifier les besoins métiers et les interactions patient-soignant pour optimiser les fonctionnalités de la plateforme.

TABLE 4.1 – Comparaison entre les trois vues UML principales

4.3 Langage UML

4.3.1 Présentation

Après avoir défini la notion de modèle et expliqué le rôle de la modélisation dans la conception logicielle, il est essentiel d'introduire l'outil principal qui permet de formaliser ces représentations : le Unified Modeling Language (UML).

Selon Booch, Rumbaugh et Jacobson, les concepteurs d'UML, ce langage graphique standardisé a été conçu pour représenter, visualiser, construire et documenter les artefacts d'un

système logiciel orienté objet [38].

Contrairement à une méthode normative, UML n'impose pas de démarche unique de développement, mais propose plutôt un métalangage flexible et extensible capable de s'adapter à divers contextes d'ingénierie logicielle [39]. Ce langage repose sur une approche visuelle, combinant des diagrammes graphiques et des descriptions textuelles pour illustrer la structure et le comportement du système.

UML joue un rôle central pour transformer dans MediCall les besoins identifiés en un modèle logique et cohérent, garantissant une meilleure communication entre les développeurs, les concepteurs et les utilisateurs finaux. Ainsi, UML sert de langage commun à toutes les parties prenantes du projet, facilitant la conception, la documentation et l'évolution du système tout au long de son cycle de vie [40].

4.3.2 Principaux diagrammes UML

Le langage UML comprend plusieurs types de diagrammes, chacun offrant une perspective unique sur le système et permettant de représenter différents aspects de sa structure et de son comportement. Ces diagrammes constituent un ensemble cohérent d'outils qui, utilisés conjointement, permettent de modéliser un système logiciel complet [41].

Voici une synthèse des principaux diagrammes utilisés dans le projet Need Nursing :

- **Diagramme de cas d'utilisation (Use Case Diagram)** : Il illustre les interactions entre les utilisateurs (patients, infirmiers, administrateurs) et le système. Ce diagramme met en évidence les fonctionnalités offertes par la plateforme et permet d'identifier les exigences fonctionnelles essentielles [42].
- **Diagramme d'activité (Activity Diagram)** : Il modélise le flux de travail et la séquence des opérations réalisées dans un processus donné, comme la réservation d'un soin à domicile ou la gestion d'un rendez-vous. Ce diagramme offre une vue dynamique du comportement du système [43].
- **Diagramme de classes (Class Diagram)** : Il représente la structure statique du système, définissant les classes principales (Patient, Infirmier, Rendez-vous, etc.), leurs attributs, méthodes et relations. Ce diagramme constitue la base de la modélisation orientée objet [44].
- **Diagramme d'objets (Object Diagram)** : Il montre une instance spécifique d'un ensemble de classes à un moment donné. Dans ce cas, il peut illustrer la relation concrète entre un patient, un infirmier et un soin planifié [45].

- **Diagramme d'états (State Machine Diagram)** : Il décrit les différents états qu'un objet peut prendre au cours de son cycle de vie, par exemple : "demande reçue", "soin en cours", "soin terminé". Ce diagramme est utile pour modéliser le comportement réactif du système face à des événements externes [46].
- **Diagramme de séquence (Sequence Diagram)** : Il détaille la séquence d'interactions entre les acteurs et le système dans un scénario donné (ex. : demande de soin → validation → affectation d'un infirmier). Il met en évidence l'ordre chronologique des échanges entre les entités du système [47].
- **Diagramme de composants (Component Diagram)** : Il illustre l'architecture logique du système en identifiant les composants logiciels (API, base de données, module IA) et leurs dépendances. Cela permet de visualiser la modularité et la maintenabilité du projet [48].
- **Diagramme de déploiement (Deployment Diagram)** : Il représente la configuration physique du système, incluant les serveurs, les terminaux mobiles et les environnements d'exécution. Ce diagramme est essentiel pour la planification du déploiement de la plateforme dans un contexte de santé numérique [49].

4.4 Vue Fonctionnelle : Diagramme de Cas d'Utilisation

4.4.1 Présentation

Dans le cadre de la modélisation fonctionnelle de notre plateforme, nous adoptons le diagramme de cas d'utilisation (Use Case Diagram), un outil fondamental du langage UML permettant de représenter les interactions entre les acteurs externes et le système. Ce diagramme met en évidence les différentes fonctionnalités offertes par la plateforme du point de vue des utilisateurs, illustrant de manière claire les services auxquels chaque acteur peut accéder [50].

L'objectif principal de cette approche est de capturer les besoins fonctionnels du système d'une manière intuitive et visuelle. En représentant les relations entre les acteurs et les cas d'utilisation, nous pouvons mieux comprendre la portée du système et identifier les limites de chaque interaction [51].

Pour le projet MediCall, trois diagrammes de cas d'utilisation distincts ont été conçus, correspondant aux trois catégories principales d'acteurs :

- **Le client (patient)** : Peut rechercher des infirmiers, consulter leurs profils, effectuer une réservation de soin à domicile, évaluer le service et gérer son historique de rendez-vous.

- **Le prestataire de soins** : Peut gérer son profil professionnel, accepter ou refuser des demandes de soins, planifier ses disponibilités et suivre les évaluations laissées par les patients.
- **L'administrateur** : Supervise l'ensemble du système, valide les nouveaux comptes de prestataires, contrôle la qualité des services et gère les incidents ou signalements.

Ces diagrammes permettent ainsi de visualiser les fonctionnalités principales du système et d'assurer une couverture complète des besoins métiers. Ils constituent également un outil de communication efficace entre les développeurs, les concepteurs et les parties prenantes, facilitant la validation des exigences avant la phase de conception détaillée [52].

En résumé, le diagramme de cas d'utilisation représente une étape cruciale dans la modélisation, car il établit la base fonctionnelle sur laquelle reposent les interactions entre les acteurs et le système. Cette représentation garantit que la conception finale du système demeure centrée sur l'utilisateur, tout en assurant la cohérence entre les objectifs techniques et les besoins réels des patients et des soignants.

4.4.2 Diagrammes de cas d'utilisation de systeme de gestion du MediCall

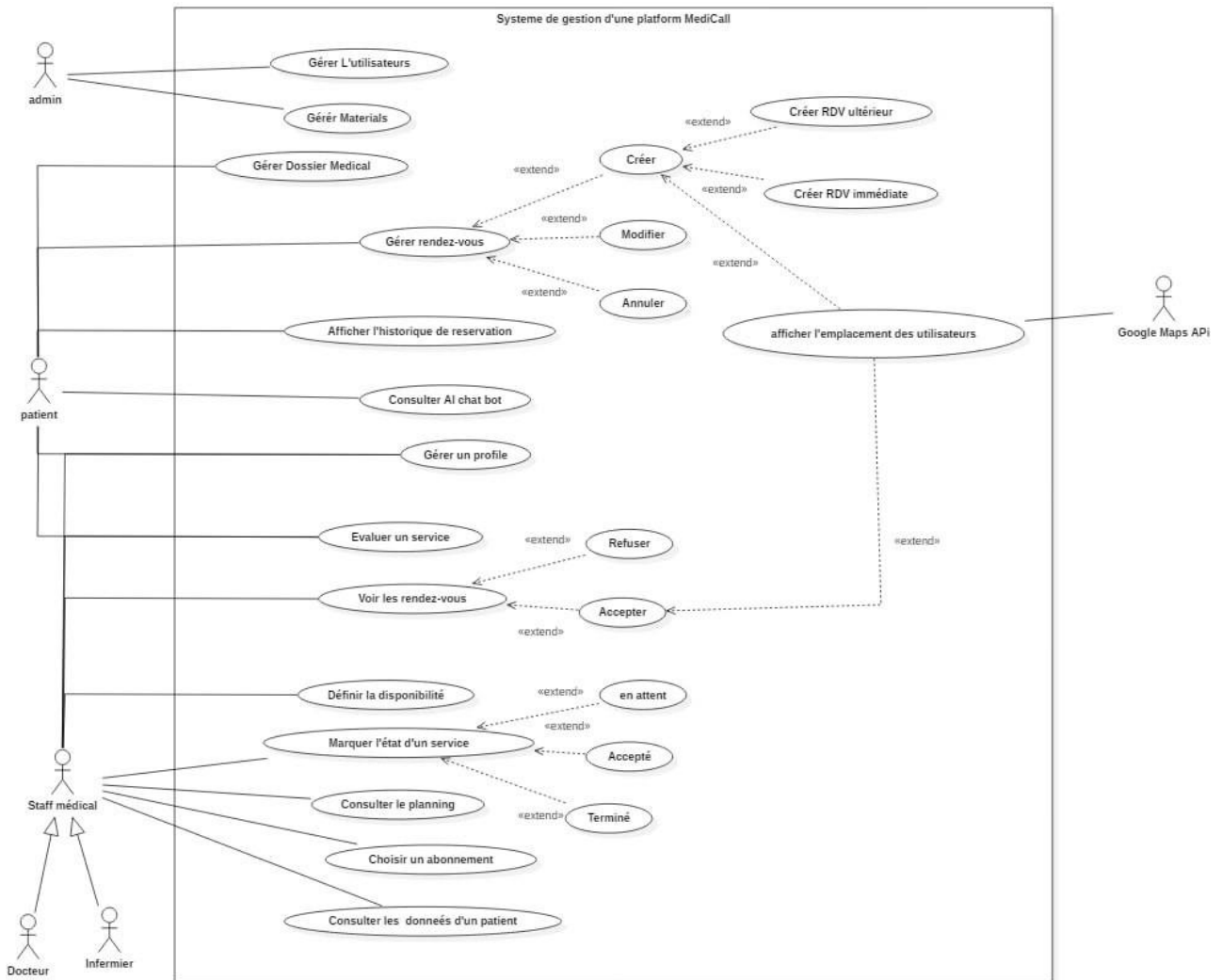


FIGURE 4.1 – Diagrammes de cas d'utilisation de systeme de gestion

4.4.3 Diagrammes de cas d'utilisation de systeme de gestion de patient

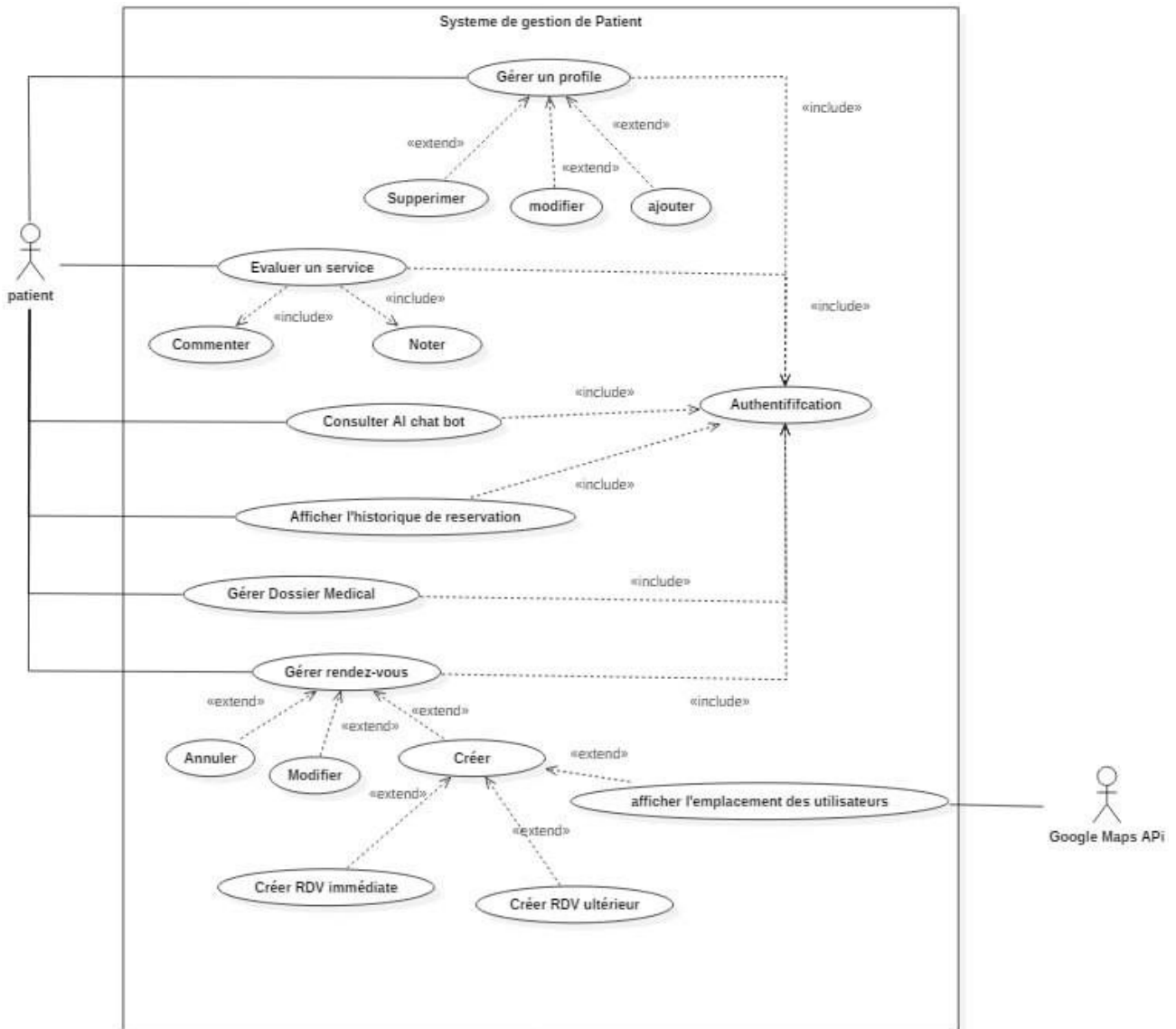


FIGURE 4.2 – Diagrammes de cas d'utilisation d'acteur patient.

4.4.4 Diagrammes de cas d'utilisation de systeme de gestion de soignant

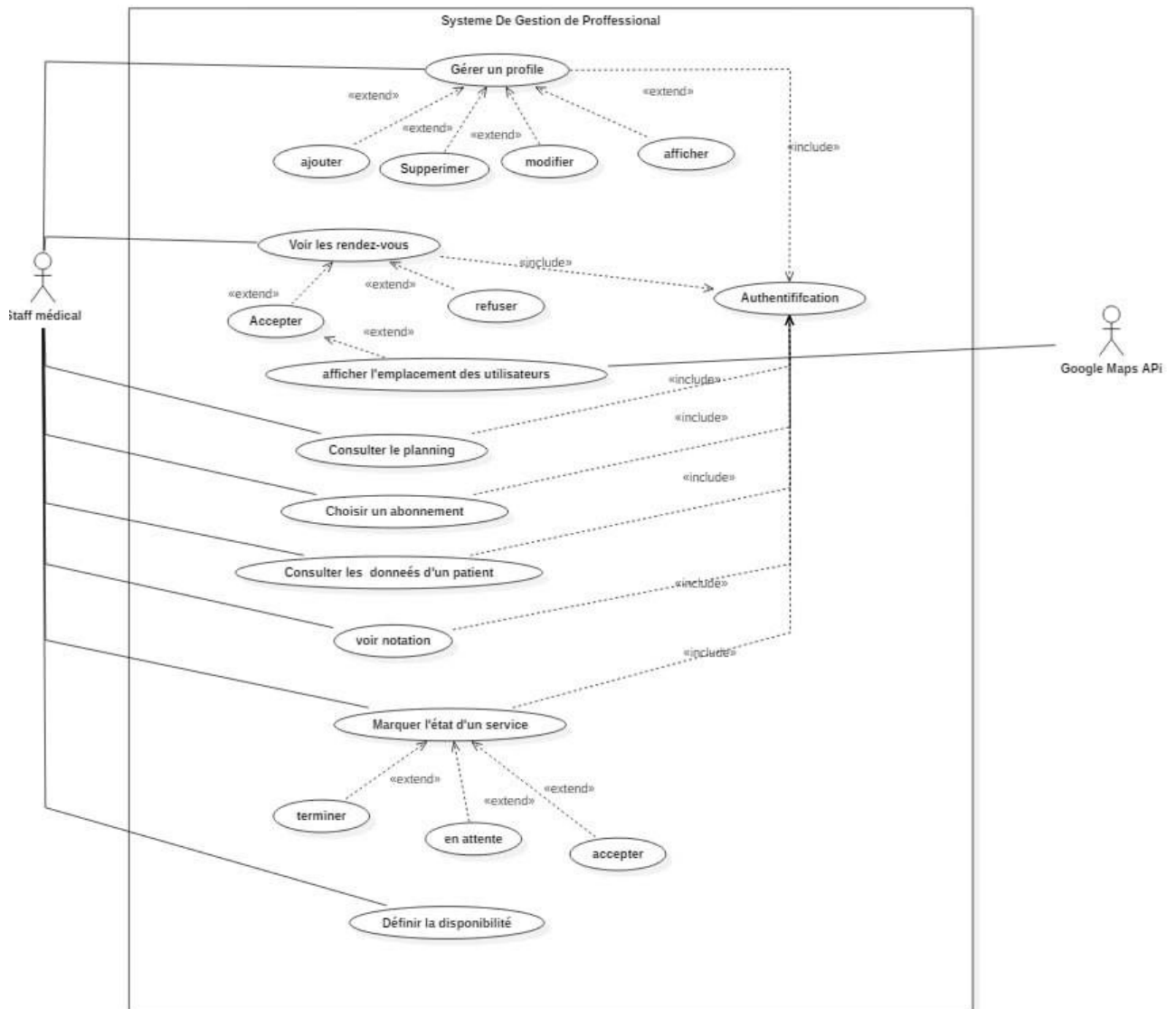


FIGURE 4.3 – Diagrammes de cas d'utilisation d'acteur soignant.

4.5 Vue Dynamique : Diagramme de Séquence Système

4.5.1 Présentation

La vue dynamique met en lumière le comportement du système dans le temps, en illustrant les échanges et la chronologie des interactions entre les différents composants et acteurs. Pour représenter cette dimension temporelle, nous avons recours au diagramme de séquence système (DSS), un outil fondamental du langage UML permettant de visualiser la communication entre les entités logicielles et les utilisateurs [53].

Ces diagrammes jouent un rôle essentiel pour comprendre la coordination entre les acteurs principaux — le patient, l’infirmier et le système — durant les différentes étapes de prestation du service. Ils mettent en évidence la séquence exacte des messages échangés entre les objets (ou acteurs) du système, ce qui permet d’analyser le déroulement des processus en temps réel [54].

Chaque diagramme de séquence élaboré décrit un scénario fonctionnel précis. Par exemple :

- Le scénario de réservation d’un soin à domicile, où le patient envoie une requête, le système la valide, puis un infirmier est assigné à l’intervention.
- Le scénario de confirmation et suivi de la prestation, illustrant les échanges entre le système, l’infirmier et le patient pour valider l’exécution du soin.
- Le scénario de notification et d’évaluation, dans lequel le patient reçoit une alerte de fin de prestation et évalue le service reçu.

Ces diagrammes permettent ainsi de visualiser la logique d’interaction et la synchronisation des processus internes du système. Ils facilitent la détection d’éventuelles incohérences, l’optimisation des flux d’échanges et l’amélioration de la réactivité du système aux demandes utilisateurs [55].

En somme, les diagrammes de séquence système offrent une vision détaillée du fonctionnement dynamique de MediCall, garantissant une conception fluide, cohérente et orientée vers la satisfaction de l’utilisateur.

4.6 Vue statique : diagramme de classes

Dans le langage UML, le diagramme de classes constitue un élément central de la modélisation orientée objet. Il permet de représenter la structure statique du système à travers les classes, leurs attributs, leurs méthodes et les relations entre elles [56]. Dans le cadre de MediCall, ce diagramme met en évidence les principales entités telles que *Utilisateur*, *Patient*,

Soignant, Service, Réservation et Paiement et leurs associations.

Ce modèle conceptuel de données (MCD) sert de base à la conception de la base de données en facilitant la compréhension de la logique métier et la préparation du passage vers la modélisation logique [57].

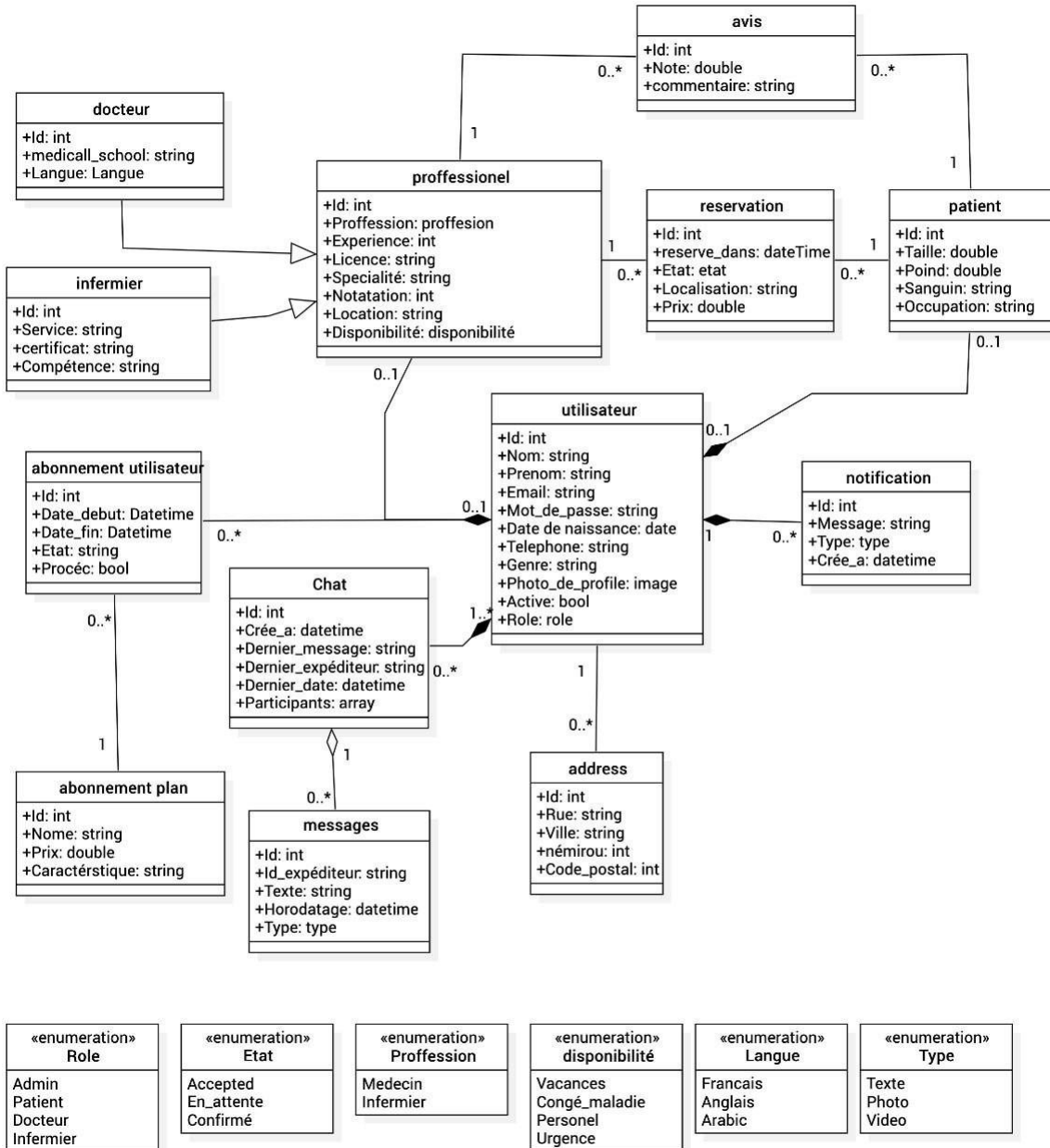


FIGURE 4.4 – Diagrammes de Class.

4.6.1 Passage vers le Modèle Logique de Données (MLD)

Le passage du MCD au MLD consiste à transformer la représentation conceptuelle en un modèle logique exploitable dans un système de gestion de base de données [58]. Alors que le MCD décrit les entités et leurs relations, le MLD précise les *tables*, les *clés primaires et étrangères*,

ainsi que les *contraintes d'intégrité* [59].

Dans le contexte de MediCall, cette transition garantit la cohérence, la normalisation et la performance de la base de données relationnelle, en préparant la conversion vers le modèle physique [60].

4.7 Conclusion

En conclusion, ce chapitre a permis d'explorer en profondeur le rôle central du langage de modélisation unifié (UML) dans la conception de notre plateforme. Nous avons étudié les différentes perspectives de modélisation — structurelle, fonctionnelle et dynamique — ainsi que les principaux diagrammes utilisés pour représenter les interactions, les processus et la structure du système.

Cette phase de modélisation constitue la base conceptuelle sur laquelle reposera la phase suivante du projet : la réalisation technique. Les modèles élaborés dans ce chapitre serviront de plan directeur pour la mise en œuvre de la solution logicielle, garantissant la cohérence entre la conception théorique et le développement réel du système. La compréhension acquise à travers l'UML guidera efficacement la transition vers l'implémentation et renforcera la communication entre les différents acteurs du projet.

Chapitre 5

Réalisation et l'Implémentation

5.1 Introduction

Après avoir exploré en profondeur la modélisation conceptuelle de notre plateforme à l'aide du langage de modélisation unifié (UML), nous entamons à présent une phase déterminante : la réalisation et l'implémentation du système. Cette étape marque la transition entre la théorie et la pratique, où les modèles conceptuels élaborés précédemment prennent forme à travers un ensemble cohérent de technologies et d'outils concrets.

Les chapitres antérieurs nous ont permis de définir les besoins fonctionnels et non-fonctionnels du système, d'analyser les interactions entre les acteurs, et de concevoir une architecture claire et structurée. Sur cette base, notre objectif dans ce chapitre est de transformer ces modèles abstraits en une application opérationnelle répondant aux exigences du domaine des soins à domicile en Algérie.

Nous présenterons dans ce chapitre l'environnement de développement choisi, les technologies et langages utilisés, ainsi que les différentes étapes de conception technique et d'intégration. L'accent sera mis sur la mise en œuvre de la plateforme, à travers ses deux principales composantes :

- **L'application web**, dédiée à la gestion, au suivi et à la coordination des services.
- **L'application mobile**, conçue pour les patients et les infirmiers afin de faciliter l'interaction et la communication en temps réel.
- **L'application de bureau**, développée pour les administrateurs et les responsables du système, offre une interface complète permettant de superviser, gérer et analyser l'ensemble des opérations de la plateforme MediCall. Elle facilite la gestion des comptes utilisateurs, la validation des inscriptions, le suivi des soins à domicile, ainsi que la génération de rapports

statistiques sur les performances et la qualité des services. Grâce à son environnement stable et ergonomique, cette application assure un contrôle optimal du système et une coordination efficace entre les différents acteurs de la plateforme.

Enfin, nous exposerons les principes de conception logicielle, les choix architecturaux et les bonnes pratiques adoptées pour garantir une solution robuste, sécurisée et évolutive, fidèle aux standards du développement moderne dans le secteur de la santé.

5.2 Architecture de Système

L'architecture de système constitue la pierre angulaire de sa conception logicielle. Elle permet de comprendre la structure globale du système, d'identifier ses principaux composants et de visualiser leurs interactions. Cette représentation architecturale offre une vision claire de la manière dont les différents modules — applicatif, serveur, base de données et intelligence artificielle — coopèrent pour assurer le bon fonctionnement de la plateforme.

La plateforme MediCall repose sur une architecture multi-couches (n-tiers), distinguant clairement la couche de présentation, la couche logique métier et la couche de données. Cette séparation favorise la modularité, la maintenabilité et l'évolutivité du système, tout en facilitant la gestion de la sécurité et des performances.

La compréhension de cette architecture de haut niveau constitue une étape essentielle avant l'étude détaillée des technologies et outils qui seront employés lors du développement pratique de la solution.

La Figure 5.1 illustre la structure générale de notre API, montrant les différentes interactions entre les acteurs du système dans le cadre de la plateforme.

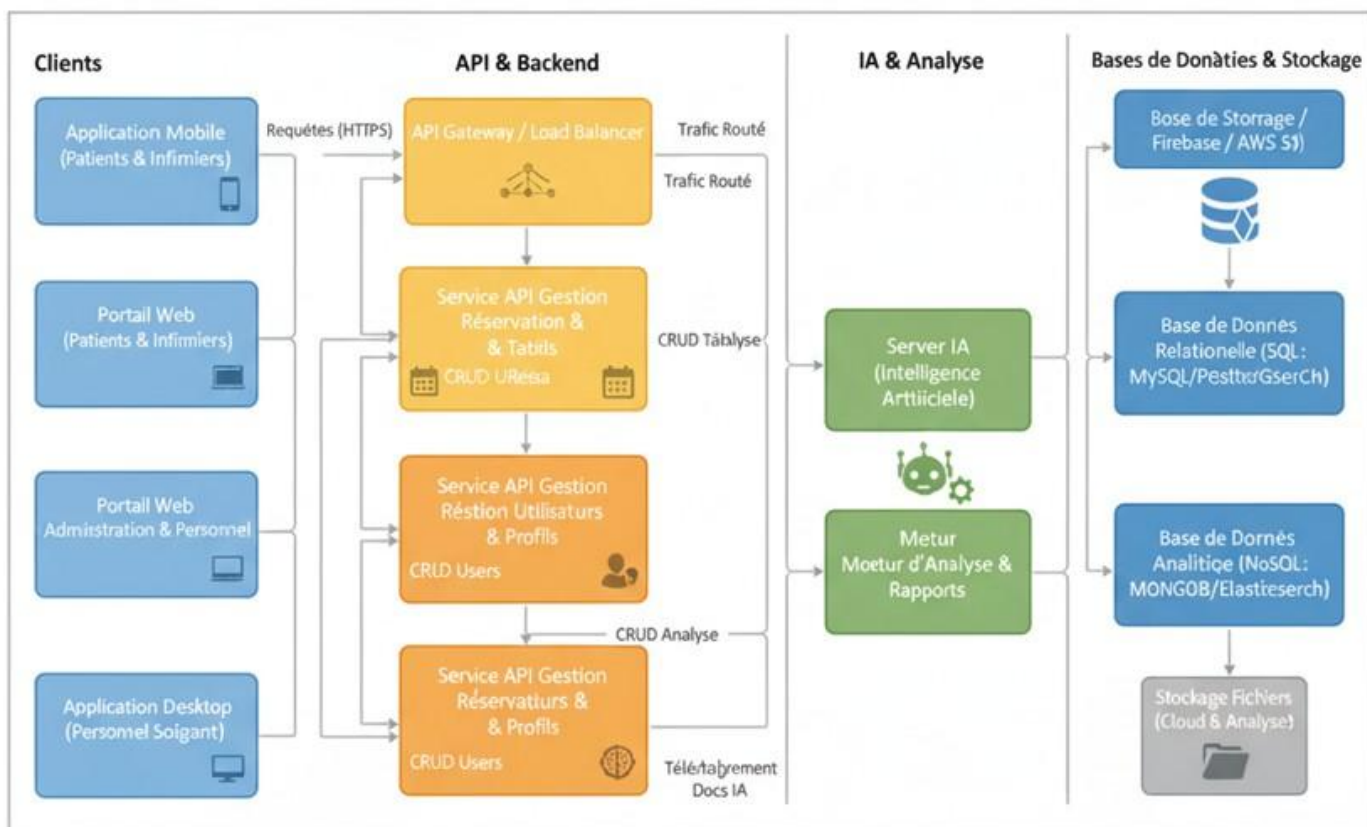


FIGURE 5.1 – Architecture de l'API pour les automobilistes en détresse

5.3 Technologies et Outils de Développement

5.3.1 Pour le Développement Web

Dans cette section, nous présentons les principaux outils et technologies utilisés pour le développement Web, dédiée à la gestion et à la coordination des soins à domicile. Ces technologies ont été soigneusement sélectionnées pour leurs performances, leur sécurité et leur compatibilité avec les standards modernes du développement Web [61,62]. Elles garantissent une application stable, fluide et évolutive, adaptée à un environnement sensible comme celui de la santé numérique.

5.3.1.1 HTML et CSS

HTML (HyperText Markup Language) et CSS (Cascading Style Sheets) sont les langages standards pour la structure et le style des pages Web [63]. HTML définit la structure sémantique des pages, tandis que CSS gère leur apparence et leur mise en forme.

5.3.1.2 Tailwind CSS et Shadcn

Tailwind CSS est un framework CSS moderne basé sur une approche *utility-first*, facilitant

la création rapide d'interfaces responsives. D'autre part, Shadcn fournit des composants préconçus et personnalisables compatibles avec Tailwind, accélérant le processus de design [64].

Raisons du choix :

- Accélère le développement d'interfaces cohérentes et esthétiques.
- Offre une grande flexibilité de personnalisation.
- Composants open source compatibles avec React.

5.3.2 Pour le développement mobile

5.3.2.1 Flutter

Pour le développement mobile de la plateforme, nous avons choisi Flutter, un framework open-source puissant et flexible développé par Google. Flutter permet de créer des applications multiplateformes (Android, iOS, Web et Desktop) à partir d'une seule base de code, garantissant ainsi une cohérence visuelle et fonctionnelle sur tous les environnements.

Grâce à son moteur de rendu performant et son vaste ensemble de widgets prédéfinis, Flutter facilite la création d'interfaces utilisateur modernes, intuitives et hautement personnalisables, tout en maintenant une excellente réactivité. Sa fonctionnalité de Hot Reload permet également aux développeurs de visualiser instantanément les modifications apportées au code, accélérant considérablement le cycle de développement.

Dans le contexte de la plateforme MediCall, Flutter offre une solution optimale pour développer une application mobile fluide et ergonomique, permettant aux patients, aux infirmiers et aux gestionnaires d'accéder à toutes les fonctionnalités de la plateforme depuis n'importe quel appareil mobile, tout en assurant des performances élevées et une maintenance simplifiée [65].

5.3.2.2 Firebase Realtime Database

Firebase Realtime Database est une base de données NoSQL développée par Google, conçue pour offrir une synchronisation des données en temps réel et une grande capacité d'évolutivité.

Contrairement aux bases de données relationnelles, elle repose sur une structure arborescente de type JSON, permettant une gestion flexible des données non structurées ou semi-structurées.

Dans le contexte de la plateforme, Firebase est utilisée pour la gestion des fonctionnalités nécessitant une mise à jour instantanée, telles que :

- le suivi en temps réel des disponibilités des prestataires de soins ;

- la géolocalisation dynamique des intervenants à domicile ;
- la diffusion rapide de notifications entre les utilisateurs.

Cette approche assure une communication fluide et réactive entre les différents acteurs du système, tout en réduisant la latence et la complexité des échanges serveur-client [66].

5.3.2.3 Python

Python est un langage de programmation de haut niveau et interprété, connu pour sa facilité d'utilisation et sa flexibilité dans le développement de l'intelligence artificielle, de l'apprentissage automatique et des systèmes backend. <https://www.python.org/>

5.3.2.4 Dart

Dart est un langage de programmation optimisé pour le développement côté client, créé par Google afin de concevoir des applications rapides sur toutes les plateformes. Dart possède un typage fort et utilise une compilation anticipée (ahead-of-time). Nous avons utilisé Dart avec le framework Flutter pour construire notre application mobile multiplateforme. <https://dart.dev/>

5.3.2.5 Android Studio

Android Studio est un environnement de développement intégré (IDE) fourni par Google, spécialement destiné au développement d'applications Android. Il offre un ensemble complet d'outils pour la conception, le codage, le débogage, les tests et le déploiement d'applications Android. Nous avons utilisé Android Studio pour construire et tester notre application mobile basée sur Flutter. <https://developer.android.com/studio>

5.3.2.6 Visual Studio Code :

Visual Studio Code (VS Code) est un éditeur de code gratuit développé par Microsoft. Léger et puissant, il offre la coloration syntaxique, l'autocomplétion, des fonctions de débogage et un contrôle de version Git intégré. Nous avons utilisé l'éditeur Visual Studio Code avec les extensions Flutter et Dart pour le développement de notre application mobile multiplateforme. <https://code.visualstudio.com/>

5.3.2.7 Figma

Figma est un outil de conception d'interface et de prototypage hébergé dans le cloud, qui permet la conception collaborative d'interfaces utilisateur. Nous avons utilisé Figma pour concevoir l'interface utilisateur (UI) et l'expérience utilisateur (UX) de notre application mobile. Figma nous a permis de créer des wireframes, des prototypes et des éléments visuels finaux.

5.3.3 Pour le développement Desktop

5.3.3.1 Electron.js

Pour la version Desktop de la plateforme, nous avons choisi d'utiliser Electron.js, un framework open-source développé par GitHub. Electron permet de créer des applications de bureau multiplateformes (Windows, macOS, Linux) en utilisant des technologies Web standards telles que HTML, CSS et JavaScript [67].

Ce choix s'explique par la nécessité d'offrir une interface de gestion robuste et performante aux administrateurs et aux responsables de la plateforme MediCall. L'application Desktop offre ainsi un environnement ergonomique et stable pour :

- Gérer les utilisateurs (clients, organisateurs et administrateurs secondaires) ;
- Superviser les réservations, les paiements et les statistiques ;
- Contrôler le contenu des événements et des expériences ;
- Consulter les rapports de performance et les logs du système.

Grâce à Electron.js, il est possible de réutiliser une grande partie du code et des composants développés pour la version Web de la plateforme, garantissant ainsi une cohérence visuelle et fonctionnelle sur tous les supports. De plus, Electron s'intègre aisément avec des bibliothèques modernes telles que React et Next.js, ce qui simplifie le développement et la maintenance [68,69].

5.3.3.2 SQLite

Pour la gestion locale des données sur le poste de travail, l'application Desktop s'appuie sur SQLite, une base de données embarquée légère et performante. Cette technologie permet à l'application de fonctionner même hors ligne, en synchronisant les données avec le serveur principal dès que la connexion Internet est rétablie [70].

Le choix de SQLite repose sur :

- Sa légèreté et sa portabilité, idéales pour les applications desktop ;
- Son intégration native avec Node.js et Electron ;
- La possibilité d'assurer un fonctionnement hybride (en ligne / hors ligne) pour une meilleure fiabilité.

Ainsi, l'application Desktop MediCall représente un outil stratégique pour la supervision et la gestion centralisée du système, tout en garantissant performance, stabilité et continuité de service.

5.3.4 Pour les Services Backend et la Gestion des Données

Cette section présente les technologies et outils utilisés dans la mise en place de l'infrastructure backend de la plateforme. Ils assurent la communication entre les différentes couches de l'application (web, mobile et desktop), garantissant la fiabilité, la performance et la sécurité des échanges de données.

5.3.4.1 API RESTful

Les API RESTful (Representational State Transfer) représentent une architecture normalisée permettant la création de services Web modulaires et extensibles. Elles utilisent les méthodes standards du protocole HTTP (GET, POST, PUT, DELETE) pour gérer les opérations CRUD (Create, Read, Update, Delete). Ce modèle permet à la plateforme de connecter ses différents clients applications Web, mobiles et desktop à un serveur centralisé de manière fluide et sécurisée.

Raisons du choix :

- Utilise des standards universels comme HTTP, JSON et URI, simplifiant l'intégration avec d'autres systèmes.
- Offre une grande flexibilité et compatibilité multi-plateformes (Web, mobile, IoT).
- Permet une évolution incrémentale sans impacter les clients existants.

5.3.4.2 Postman

Postman est une plateforme de test et de documentation d'API largement adoptée par les développeurs pour la validation et le débogage des endpoints REST. Il fournit une interface graphique conviviale pour envoyer des requêtes HTTP, inspecter les réponses et automatiser les tests unitaires d'API.

Raisons du choix :

- Simplifie le test et le débogage des API REST.
- Génère automatiquement une documentation interactive.
- Permet d'automatiser les tests récurrents afin d'assurer la fiabilité du backend.

5.3.4.3 MySQL

MySQL est un système de gestion de bases de données relationnelles open source réputé pour sa robustesse et sa compatibilité avec les environnements de production, Il constitue le cœur du stockage des données pour la plateforme, notamment les informations relatives aux patients, aux infirmiers et aux réservations.

Raisons du choix :

- Fiabilité et performance reconnues pour les applications à fort trafic.
- Compatibilité avec les langages et frameworks modernes.
- Supporte la réplication, la mise en cache et la scalabilité horizontale.

5.3.4.4 Git et GitHub

Git est un système de contrôle de version distribué permettant de suivre l'évolution du code source, tandis que GitHub offre une plateforme collaborative pour l'hébergement et la gestion de projets logiciels. Ils facilitent la coopération entre les membres de l'équipe de développement.

Raisons du choix :

- Suivi précis des modifications et collaboration simultanée entre développeurs.
- Intégration continue (CI/CD) et gestion centralisée du code source.

5.3.4.5 Seats.io

Seats.io est une solution SaaS spécialisée dans la gestion des sièges et des réservations interactives pour les événements et activités. Dans le contexte, cette technologie est exploitée pour la gestion visuelle des plannings et des créneaux de soins, offrant une interface intuitive de réservation en temps réel.

Raisons du choix :

- Interface interactive facilitant la visualisation des disponibilités.
- Gestion en temps réel des réservations et des plannings.
- Personnalisation adaptable selon le type de service ou d'événement.

5.4 Expérience Utilisateur et Navigation

L'expérience utilisateur au sein de la plateforme a été soigneusement conçue afin d'offrir une navigation fluide, intuitive et agréable pour l'ensemble des profils d'utilisateurs. La simplicité d'utilisation, la clarté des interfaces et l'accessibilité constituent les principes directeurs de notre démarche de conception.

La plateforme a été structurée en trois environnements principaux, chacun répondant à des besoins spécifiques :

- **Administrateur** : dispose d'un tableau de bord complet lui permettant de gérer les utilisateurs, les services, les réservations et la sécurité du système.
- **Organisateur (professionnel de santé ou coordinateur)** : accède à un espace dédié pour publier, suivre et gérer les activités médicales, les plannings et les statistiques de performance.
- **Client (patient ou bénéficiaire)** : interagit avec une interface simple et conviviale lui permettant de rechercher des services, effectuer des réservations, suivre ses rendez-vous et évaluer la qualité des prestations reçues.

La conception de la navigation repose sur le principe de parcours utilisateur fluide, garantissant un minimum d'actions nécessaires pour atteindre un objectif. De plus, une attention particulière a été portée à la cohérence graphique et à la compatibilité multiplateforme, assurant une expérience homogène entre les versions Web, mobile et desktop de MediCall.

Cette approche centrée sur l'utilisateur contribue non seulement à renforcer la satisfaction des usagers, mais également à favoriser l'adoption de la plateforme dans le contexte sensible de la santé numérique.

5.4.1 Présentation de notre API

L'API RESTful développée pour la plateforme constitue la couche centrale de communication entre le client Web, les applications mobiles et l'infrastructure de base de données. Conçue à l'aide de Node.js et Express.js, elle expose des points d'accès (endpoints) structurés et sécurisés permettant la gestion des utilisateurs, des événements, des réservations et des paiements. Toutes les requêtes et réponses sont échangées au format JSON, assurant une communication légère, rapide et standardisée. Chaque *endpoint* est soigneusement conçu pour correspondre aux scénarios d'utilisation définis dans la section 3.5.2, garantissant ainsi une parfaite interopérabilité entre les différents composants du système. De plus, grâce à l'authentification basée sur les JSON Web Tokens (JWT) et à la synchronisation en temps réel via la Firebase Realtime Database [?], l'API assure un haut niveau de sécurité et de réactivité, éléments essentiels pour une plateforme de services connectés.

5.4.2 Routes API pour l'application mobile

Les routes de l'API de l'application mobile MediCall sont conçues pour gérer l'ensemble du cycle de vie d'une demande de soin à domicile — depuis l'authentification de l'utilisateur jusqu'à la validation du service et l'évaluation du professionnel de santé. Chaque point de terminaison (*endpoint*) assure une communication fluide et sécurisée entre le patient, l'infirmier et le serveur central, tout en garantissant la synchronisation en temps réel des données médicales et des statuts de service. Ces routes sont optimisées pour une utilisation mobile, avec une faible latence et une intégration avancée de l'API Google Maps permettant le suivi géolocalisé du personnel soignant et la planification des interventions à domicile.

5.4.3 Routes API pour les Infirmiers Autonomes et les Infirmiers Affiliés aux Centres Médicaux (Application Mobile)

Les routes API destinées aux infirmiers ont été conçues pour gérer l'ensemble du cycle d'intervention, incluant la disponibilité, l'acceptation des demandes et la mise à jour de l'état des prestations. Ces points de terminaison (*endpoints*) permettent de distinguer deux profils d'utilisateurs : les infirmiers autonomes, qui reçoivent directement les demandes de soins via la plateforme, et les infirmiers affiliés à des centres médicaux, dont les missions sont attribuées par un administrateur ou un responsable de centre.

L'intégration de la communication en temps réel est assurée grâce à la technologie Web-

Socket, permettant l'envoi immédiat de notifications concernant les nouvelles demandes ou les modifications de statut d'intervention. Par ailleurs, la base de données Firebase Realtime Database est exploitée pour effectuer des requêtes géospatiales et optimiser l'attribution des soins en fonction de la localisation des infirmiers et des patients. Sa structure hiérarchique en JSON facilite un accès rapide et efficace aux données de position en temps réel.

5.4.4 Routes API pour les Administrateurs de Centres Médicaux (Application Web)

Les administrateurs de centres médicaux utilisent des routes API spécifiques pour gérer les infirmiers, attribuer les demandes de soins et superviser l'ensemble des interventions effectuées sur la plateforme. Ces points de terminaison (*endpoints*) jouent un rôle essentiel dans la coordination et la supervision des activités médicales, garantissant une gestion fluide des ressources humaines et matérielles.

Grâce à ces routes, les administrateurs peuvent :

- Consulter et gérer la liste des infirmiers inscrits sur la plateforme.
- Attribuer des demandes de soins aux infirmiers disponibles en fonction de leur localisation et de leur spécialité.
- Suivre en temps réel l'avancement des interventions et vérifier leur état (planifiée, en cours, terminée).
- Consulter les historiques des interventions et les évaluations des patients.

Ces fonctionnalités sont intégrées à la plateforme Web via une architecture basée sur Node.js et Express.js, garantissant une communication fluide avec la base de données Firebase Realtime Database et des mises à jour instantanées grâce à WebSocket.

5.4.5 Routes API pour les Administrateurs de la Plateforme (Application Web)

Les administrateurs de la plateforme supervisent l'ensemble de l'écosystème numérique, garantissant la stabilité, la sécurité et la conformité des services. Ils disposent d'un ensemble de routes API dédiées à la gestion des utilisateurs, à la validation des inscriptions, à la surveillance du système et à la consultation des statistiques globales.

Ces endpoints incluent des fonctionnalités avancées telles que :

- La gestion complète des comptes utilisateurs (création, mise à jour, suspension, suppression).
- L'approbation ou le rejet des demandes d'inscription des infirmiers et des centres médicaux.
- La collecte et l'analyse des données d'utilisation pour produire des tableaux de bord statistiques.
- La surveillance des activités et des journaux d'audit (*logs*) pour garantir la traçabilité et la conformité.

Pour renforcer la sécurité, ces routes intègrent des mécanismes de rate limiting afin de prévenir les abus et les attaques automatisées, ainsi qu'un système d'authentification renforcé basé sur les tokens JWT.

5.5 Diagrammes de Flux Utilisateur

Les diagrammes de flux utilisateur représentent les différentes étapes qu'un acteur parcourt pour atteindre un objectif spécifique au sein de la plateforme. Ils mettent en évidence la séquence logique des actions, les transitions entre les interfaces, ainsi que les interactions principales entre l'utilisateur et le système.

Ces diagrammes jouent un rôle essentiel dans la conception centrée sur l'utilisateur, car ils permettent :

- De visualiser le parcours complet de chaque type d'utilisateur (client, infirmier, administrateur).
- D'identifier les points d'interaction critiques et les éventuels obstacles dans le processus d'utilisation.
- D'améliorer l'ergonomie et la fluidité de navigation au sein de la plateforme.

Dans cette section, nous présentons les diagrammes de flux pour les acteurs principaux, en mettant l'accent sur les fonctionnalités clés telles que :

- L'inscription et la connexion sécurisées.
- La recherche et la réservation de services de soins à domicile.
- La gestion des interventions et des suivis médicaux.
- La supervision et la validation par les administrateurs.

Ces représentations visuelles constituent une base solide pour la modélisation des interactions et la conception des interfaces graphiques.

5.5.1 Diagramme de Flux du Patient (Utilisateur Principal)

Le flux d'utilisation du patient sur la plateforme commence par la création d'un compte ou la connexion à son espace personnel. Une fois authentifié, l'utilisateur peut soumettre une demande de service en choisissant la catégorie de soin souhaitée (soins infirmiers, assistance médicale, suivi à domicile, etc.) et en précisant son emplacement.

Après la soumission, le patient peut :

- Consulter la liste des professionnels de santé disponibles à proximité.
- Recevoir des notifications en temps réel concernant la prise en charge de sa demande.
- Suivre la progression du professionnel (infirmier ou médecin) jusqu'à son domicile grâce à la géolocalisation intégrée.
- Effectuer le paiement de manière sécurisée via la plateforme.
- Donner un retour d'expérience et une évaluation de la prestation reçue.

Ce flux illustre l'expérience complète du patient, depuis la demande initiale jusqu'à la clôture du service, garantissant une interaction fluide et transparente entre le patient et les prestataires de soins.

5.5.2 Diagramme de Flux de l'Infirmier Autonome

Le flux d'utilisation de l'infirmier autonome dans la plateforme débute par la connexion à son compte professionnel, où il peut gérer sa disponibilité et accéder aux demandes de soins publiées par les patients.

Une fois connecté, l'infirmier peut :

- Mettre à jour son statut de disponibilité (en ligne / hors ligne).
- Consulter les demandes de soins disponibles à proximité selon sa spécialité.
- Accepter ou refuser une intervention proposée.
- Suivre la localisation du patient et planifier son déplacement.
- Mettre à jour l'état de l'intervention (en cours, terminée, annulée).
- Consulter son historique d'interventions et ses gains pour une meilleure gestion de son activité.

Ce flux met en évidence la flexibilité et l'autonomie offertes aux professionnels de santé dans la gestion de leurs prestations sur la plateforme, tout en assurant une coordination fluide avec les patients.

5.5.3 Diagramme de Flux de l'Administrateur d'Établissement de Santé

Le flux d'utilisation de l'administrateur d'établissement dans la plateforme commence par la connexion à son espace de gestion sécurisé. Depuis son tableau de bord, il peut superviser l'ensemble des opérations relatives à la coordination des soins et à la gestion du personnel médical.

L'administrateur dispose des fonctionnalités suivantes :

- Gérer les profils des infirmiers affiliés à l'établissement (ajout, modification, suspension).
- Affecter les demandes de soins reçues aux infirmiers disponibles selon leur spécialité et leur proximité géographique.
- Suivre en temps réel l'état des interventions en cours (en attente, en cours, terminées).
- Consulter les statistiques globales de performance de l'établissement (nombre de soins effectués, taux de satisfaction, délais moyens d'intervention, etc.).
- Générer des rapports d'activité et exporter des données à des fins d'analyse ou de conformité.

Ce flux illustre la manière dont la plateforme centralise la gestion des opérations médicales, permettant aux administrateurs d'assurer une coordination optimale entre les soignants et les patients tout en maintenant une qualité de service élevée.

5.5.4 Diagramme de Flux de l'Infirmier Affilié à un Établissement

Le flux d'utilisation de l'infirmier affilié à un établissement de santé dans la plateforme débute par une authentification sécurisée pour accéder à son espace professionnel. Une fois connecté, l'infirmier peut consulter les demandes de soins qui lui ont été attribuées par l'administrateur, mettre à jour le statut des interventions, et suivre l'évolution de ses missions.

Les principales étapes de ce flux sont les suivantes :

- Connexion au compte professionnel et accès au tableau de bord personnel.
- Consultation des demandes de soins assignées en fonction de la disponibilité et de la spécialité.

- Acceptation ou rejet d'une intervention selon la charge de travail.
- Mise à jour du statut de l'intervention (en attente, en cours, terminée).
- Consultation de l'historique des interventions réalisées.
- Gestion du profil personnel (coordonnées, compétences, disponibilité).

Ce flux permet d'optimiser la coordination des interventions et d'assurer un suivi transparent des soins prodigués, renforçant ainsi la traçabilité et la qualité du service au sein de la plateforme.

5.5.5 Diagramme de Flux de l'Administrateur de la Plateforme

Le flux d'utilisation de l'administrateur de la plateforme débute par une authentification sécurisée lui donnant accès à l'ensemble des fonctionnalités de supervision du système. Depuis son tableau de bord, l'administrateur peut gérer les utilisateurs, approuver ou refuser les inscriptions des prestataires de soins, surveiller les activités en temps réel, et consulter les rapports d'utilisation.

Les principales étapes de ce flux sont les suivantes :

- Connexion sécurisée à l'espace administrateur.
- Consultation et validation des nouvelles inscriptions (patients, infirmiers, établissements).
- Gestion des prestataires de soins (modification, suspension ou suppression de comptes).
- Suivi en temps réel des interventions et des localisations via des outils d'analyse.
- Génération de rapports statistiques sur l'activité de la plateforme.
- Supervision générale pour garantir la qualité, la sécurité et la conformité des services.

Ce flux assure une gouvernance efficace de la plateforme, permettant de maintenir un environnement fiable, transparent et conforme aux standards de la santé numérique.

5.6 Le Chatbot

5.6.1 Grands Modèles de Langage (MLL)

Les Grands Modèles de Langage (MLL) représentent une catégorie avancée de réseaux neuronaux fondés sur les architectures transformer, comprenant plusieurs milliards de paramètres, distincts des modèles classiques de régression [71]. Ces modèles sont pré-entraînés sur

d'immenses corpus textuels à l'aide de techniques d'auto-apprentissage supervisé, leur permettant d'acquérir une compréhension approfondie des structures linguistiques et sémantiques du langage humain. Ils apprennent à générer, reformuler et interpréter du texte naturel en utilisant des méthodes telles que le masquage, la modélisation masquée et la prédiction du prochain jeton.

Caractéristiques principales :

- **Échelle :** Contiennent des milliards de paramètres permettant de modéliser des raisonnements linguistiques complexes.
- **Architecture :** Constitués de blocs transformeurs dotés de mécanismes d'auto-attention optimisés pour la cohérence contextuelle.
- **Compétences :** Génération de texte, compréhension contextuelle, résumé automatique et réponse aux questions.
- **Adaptabilité :** Ajustement et entraînement possibles sur des données spécialisées, pour une adaptation au domaine médical ou infirmier [72].

5.6.2 Implémentation de LLaMA-3.2-1B

Dans MediCall, nous avons choisi et optimisé le modèle LLaMA-3.2-1B de Meta, un modèle open source comportant environ 1,2 milliard de paramètres, adapté pour le traitement et la génération de réponses aux requêtes médicales [73].

Détails du modèle :

- **Architecture :** Transformateur encodeur-décodeur.
- **Fenêtre contextuelle :** 512 symboles.
- **Quantification :** NF4 à 4 bits avec double quantification pour l'efficacité mémoire.
- **Mémoire :** Environ 3,8 Go de VRAM nécessaires à l'entraînement.

Configuration d'apprentissage :

- **Jeu de données :** Ensemble JSONL personnalisé pour les questions médicales (répartition 80/20 entre apprentissage et test).
- **Affinement :** Utilisation de *transformateurs LoRA* ($r = 16$) appliqués aux projections d'attention.
- **Optimisation :** Optimiseur AdamW avec un taux d'apprentissage de $3 \cdot 10^{-5}$ et planification cosinus.
- **Matériel :** Entraînement accéléré sur GPU grâce aux crédits Lightning AI.

5.6.3 Collecte et Préparation des Données

Le dataset utilisé pour le fine-tuning du modèle LLaMA-3.2-1B provient d'un dataset public disponible sur Kaggle, portant sur la vérification des symptômes médicaux (symptom checker dataset). A partir de ce corpus, une sélection de 200 dialogues de haute qualité a été réalisée pour constituer un petit corpus afin de travailler en premier lieu la capacité d'adaptation du modèle. Chaque dialogue consistait en un échange autour d'un symptôme médical entre une question nef et une réponse correspondante. Pour chaque prompt, il y avait une réponse attendue, ce qui fait beaucoup de sens à entraîner un modèle sur des conversations de ce type. Le dataset a été converti dans un format JSONL standardisé. Le dataset est dans le format suivant :

```
{"prompt":"J'ai de la fièvre et je tousse, que puis-je faire ?","response":"En cas de fièvre et de toux, il est recommandé de prendre du paracétamol pour faire baisser la fièvre et soulager les maux de tête. Consultez un médecin si les symptômes persistent plus de 48 heures."}
```

Le dataset a été pré-traité, normalisé et nettoyé :

- Conversion en minuscules et nettoyage des espaces blancs inutiles ;
- Suppression des doublons et des questions/réponses incomplètes ;
- Suppression des questions/réponses trop robotiques, ou qui semblent avoir été générées par un modèle (ex à la chaîne);

Correction grammaticale des questions/réponses et amélioration du flux linguistique.

En considération du fait que le corpus est assez petit, cet entraînement de fine-tuning a avant tout une visée expérimentale : évaluer la capacité d'adaptation du modèle et son alignement sur le domaine de la santé. Les données ont été séparées en train/test split à 90/10. Le dataset final a été limité à un maximum de 512 tokens pour les dialogues à sens une fenêtre de contexte seul. Cela a pour but d'économiser la RAM disponible pour le training. Ce petit corpus permettra de jauger de la qualité suivante des symptômes et la qualité conversationnelle du modèle lors de ses premières étapes de personnalisation dans le domaine de la santé.

```

{"prompt": "I have a headache and feel dizzy.", "response": "That sounds uncomfortable. Headaches with dizziness can come from dehydration, fatigue, or low blood pressure. Drink water, rest, and if symptoms persist or worsen, consult a doctor. This is general information, not a medical diagnosis. Please consult a healthcare professional for personal advice."}
{"prompt": "My child has a fever of 39°C. What should I do?", "response": "I'm sorry you're feeling this way. For a child with high fever, give plenty of fluids and fever medicine if prescribed. If the fever persists, or if the child is very weak or unresponsive, seek urgent medical care. This is general information, not a medical diagnosis. Please consult a healthcare professional for personal advice."}

```

Figure 5.2 Préparation des Données

5.6.4 Mise en œuvre et Expérimentations

La réalisation de notre chatbot médical correspond à un travail de fine-tuning précis et bien documenté techniquement, dans le but d'adapter un modèle de langage à usage général à un usage medical spécifique. Il s'agit d'une solution permettant d'avoir une IA de dialogue fonctionnelle dans un contexte médical et apte à répondre à des questions, sans devoir apprendre de A à Z. La méthodologie employée pour notre assistant médical ne s'écarte pas des travaux précédemment mentionnés.

```

model_name = "meta-llama/Llama-3.2-1B"
login(token=hf_token, ...)
model = AutoModelForCausalLM.from_pretrained(model_name, ...)

```

Figure 5.3 Sélection du modèle de base et authentification

5.6.5 Mise en œuvre du Fine-Tuning à Paramètres Efficaces

Pour entraîner notre modèle de façon efficace tout en maintenant une qualité élevée des résultats, nous avons eu recours à la méthode LoRA (Low-Rank Adaptation) — une méthode de fine-tuning à paramètres efficaces.

Ainsi, plutôt que d'entraîner à nouveau les 1,2 milliard de paramètres du modèle de base, LoRA limite les modifications à 3,4 millions de paramètres environ, soit 0,28 % du total.

Cela permet donc de réduire largement les besoins en calcul tout en permettant au modèle de garder la capacité à apprendre des connaissances spécifiques au domaine médical.

Concrètement, il s'agit d'ajouter aux différentes couches du modèle des matrices de faible rang et entraînaibles, qui sont des chemins spécialisés dans l'acquisition des connaissances médicales, mais qui ne touchent pas à la compréhension du langage général du modèle.

5.6.6 Optimisations de la mémoire et configuration matérielle

De nombreuses techniques d'optimisation de la mémoire ont été mobilisées pour pouvoir entraîner le modèle dans des conditions raisonnables en terme de ressources de calcul.

La plus emblématique en est celle de la quantification en 4 bits (NF4) qui permet de réduire drastiquement l'empreinte mémoire de modèles à base de transformers en représentant leurs poids dans une précision moindre, sans perdre pour autant en qualité de résultats.

Une autre solution retenue est celle du gradient checkpointing qui permet de réduire la mémoire consommée en entraînement en acceptant de devoir recalculer certaines activations à la volée, échangeant de la mémoire contre du temps de calcul supplémentaire.

La totalité de l'entraînement a été porté sur GPU via l'infrastructure Lightning AI, avec un fallback automatique vers le CPU si besoin pour assurer une meilleure portabilité entre différentes machines.

```
lora_config = LoraConfig(  
    r=16, lora_alpha=32, target_modules=["q_proj", "v_proj", "k_proj", "o_proj"],  
    lora_dropout=0.1,  
    bias="none",  
    task_type="CAUSAL_LM",  
)  
model = get_peft_model(model, lora_config)
```

Figure 5.4 Optimisations de la mémoire

5.6.7 Formation et sélection des hyperparamètres

Les hyperparamètres ont été choisis avec soin afin d'optimiser l'équilibre entre la performance et la capacité de généralisation de l'algorithme. Le modèle a traité les données dans des mini-lots de deux exemples par dispositif, et l'évaluation a été réalisée à la fin de chaque époque. Une valeur de taux d'apprentissage (learning rate) de 2×10^{-4} a été utilisée avec un scheduler de cosinus pour obtenir une convergence harmonieuse, et un poids de décroissance (weight decay) de 0.01 a été appliqué pour éviter le surapprentissage.

La formation a comporté trois époques. Un arrêt anticipé (patience = 3) a également été utilisé afin de ne pas entraîner inutilement lorsque la perte de validation ne s'améliore plus. Des points d'évaluation réguliers ont été automatiquement sauvegardés à la fin de chaque époque, permettant ainsi de retrouver les poids du modèle au meilleur instant de son entraînement.

```
quant_config = BitsAndBytesConfig(  
    load_in_4bit=True, ...)  
model = prepare_model_for_kbit_training(model, use_gradient_checkpointing=True)
```

Figure 5.5 sélection des hyperparamètres

5.6.8 Analyse des Résultats d'Entraînement

L'analyse des métriques d'entraînement montre que le modèle a bien appris au fil des itérations puisque ses paramètres ont été judicieusement affinés. La perte d'entraînement est passée de 0.473 à 0.223 traduisant une belle capacité du modèle à s'adapter aux données du jeu d'apprentissage.

Dans le même temps, la perte de validation a diminué de 0.257 à 0.217, signe que le modèle a su généraliser ce qu'il a appris et n'a pas encore tendance à basculer dans un surapprentissage.

Que ce soit à l'entraînement ou en validation, la décision de poursuivre l'apprentissage a été judicieuse jusqu'à la fin des trois époques puisque les deux courbes de perte sont descendantes et stables. Le meilleur checkpoint a été atteint à la fin de la troisième époque avec une perte de validation de 0.217, soit le modèle final retenu au terme de ce fine-tuning.

```
training_args = TrainingArguments(  
    output_dir="./outputs",  
    eval_strategy="epoch",  
    save_strategy="epoch",  
    logging_strategy="steps",  
    logging_steps=20,  
    save_total_limit=2,  
    num_train_epochs=3,  
    ...  
)
```

Figure 5.6 Résultats d'Entraînement

5.6.9 Intégration du Modèle et Préparation au Déploiement

Pour terminer, nous avons fusionné les LoRA entraînées avec le modèle de base, pour obtenir un chatbot autonome et utilisable directement en prod. Des mécanismes de fusion permettent de „coller“ les connaissances médicales spécifiques obtenues par le fine-tuning avec les compétences linguistiques plus générales du modèle de départ.

En guise de résultat, nous avons un assistant médical de qualité capable de formuler des réponses fluides et pertinentes à des questions de santé et de suivi médical. Le modèle final a été sauvegardé avec une sérialisation adéquate pour assurer un bon chargement et exécution ainsi qu'une intégration en production au sein de l'application.



[12183/12183 40:38, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	1.844100	1.846089
2	1.664900	1.828524
3	1.654900	1.836304

Figure 5.7 Préparation au Déploiement

Cette implémentation illustre l'utilisation pratique des techniques actuelles d'adaptation de l'intelligence artificielle pour créer des assistants spécialisés dans un domaine précis.

L'approche adoptée permet de transformer les capacités générales d'un modèle de langage en connaissances médicales spécialisées, tout en optimisant les ressources de calcul et de déploiement

Cette mise en œuvre représente une amélioration significative de l'intégration du chatbot dans l'application de santé, offrant aux utilisateurs des réponses intelligentes, contextuellement adaptées et pertinentes à leurs questions médicales et besoins de soins.

5.7 Présentation de notre Application

5.7.1 Notre Application DESKTOP

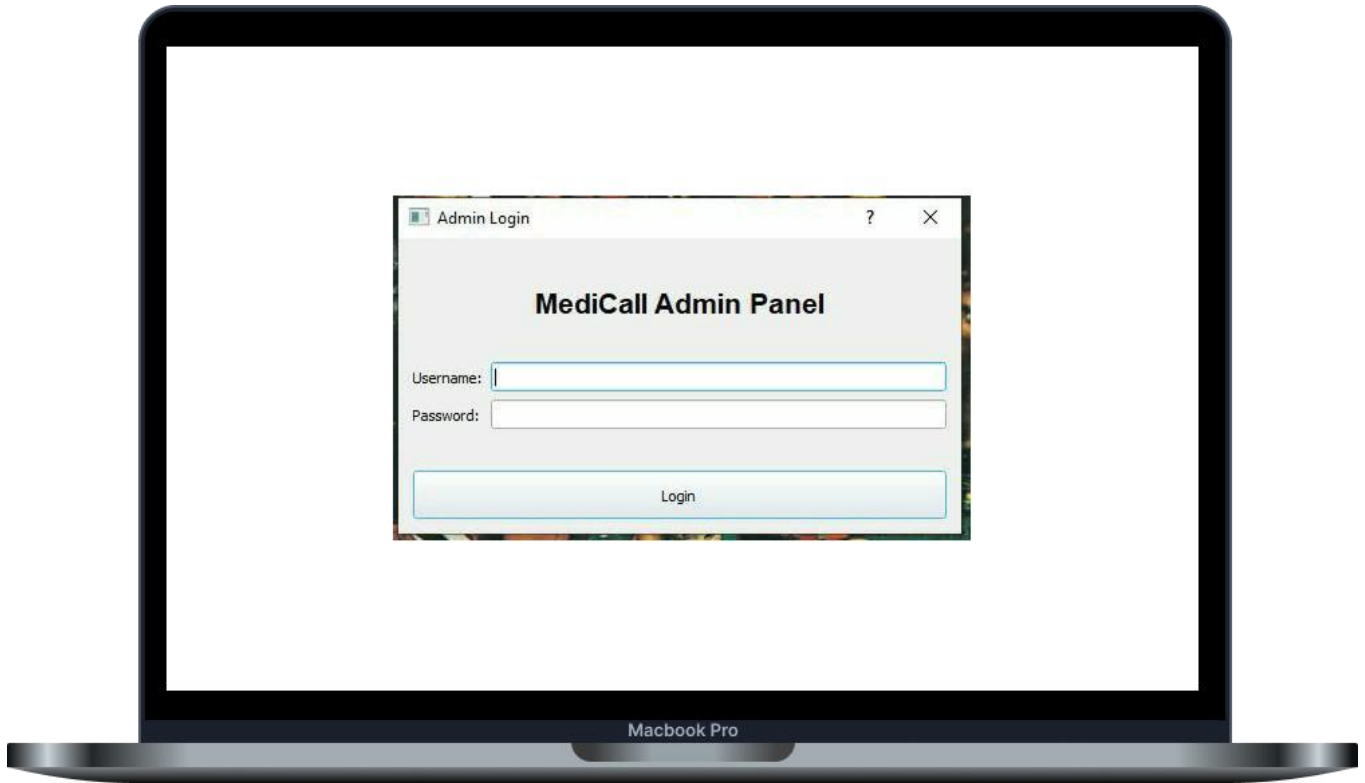
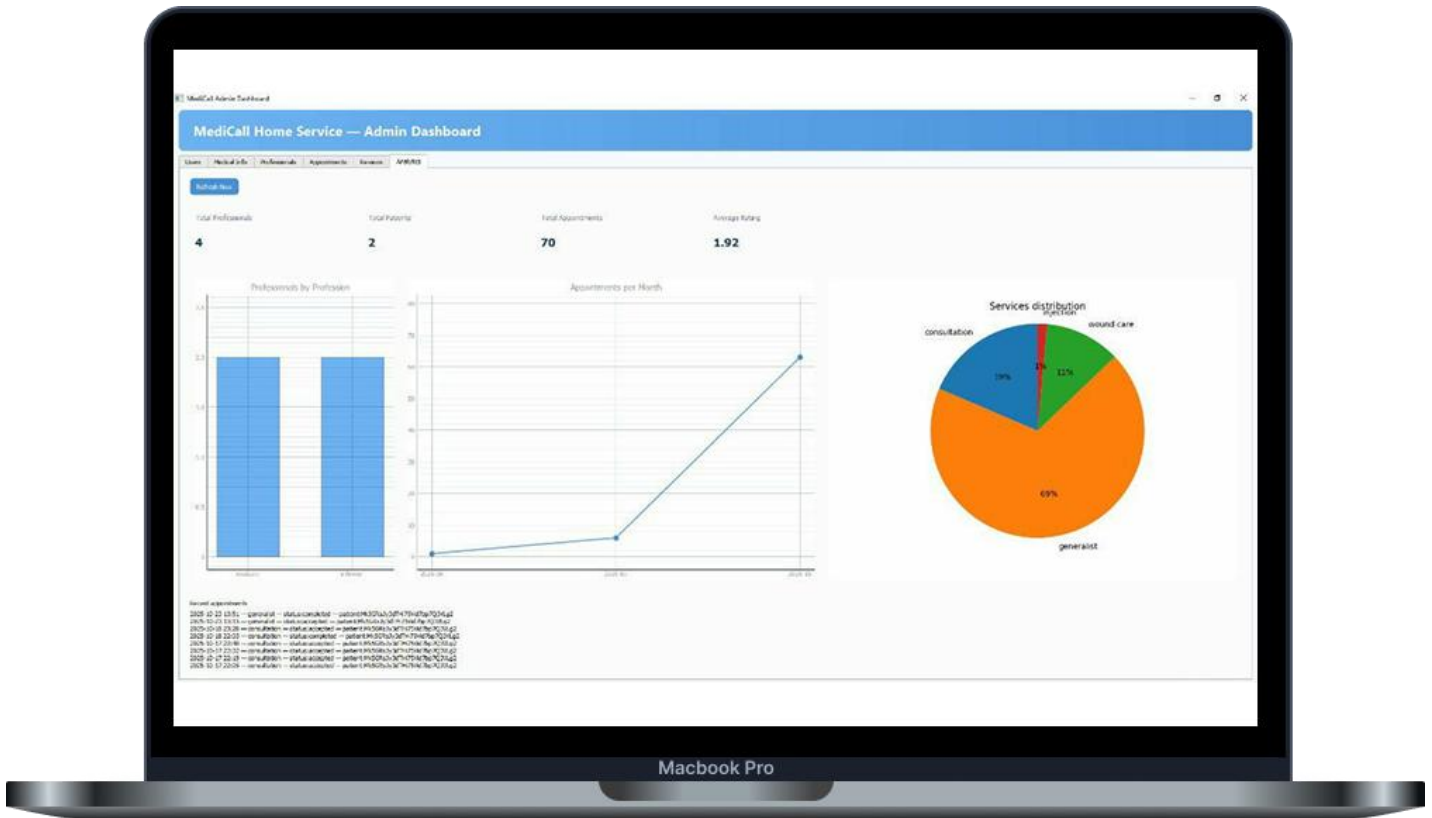
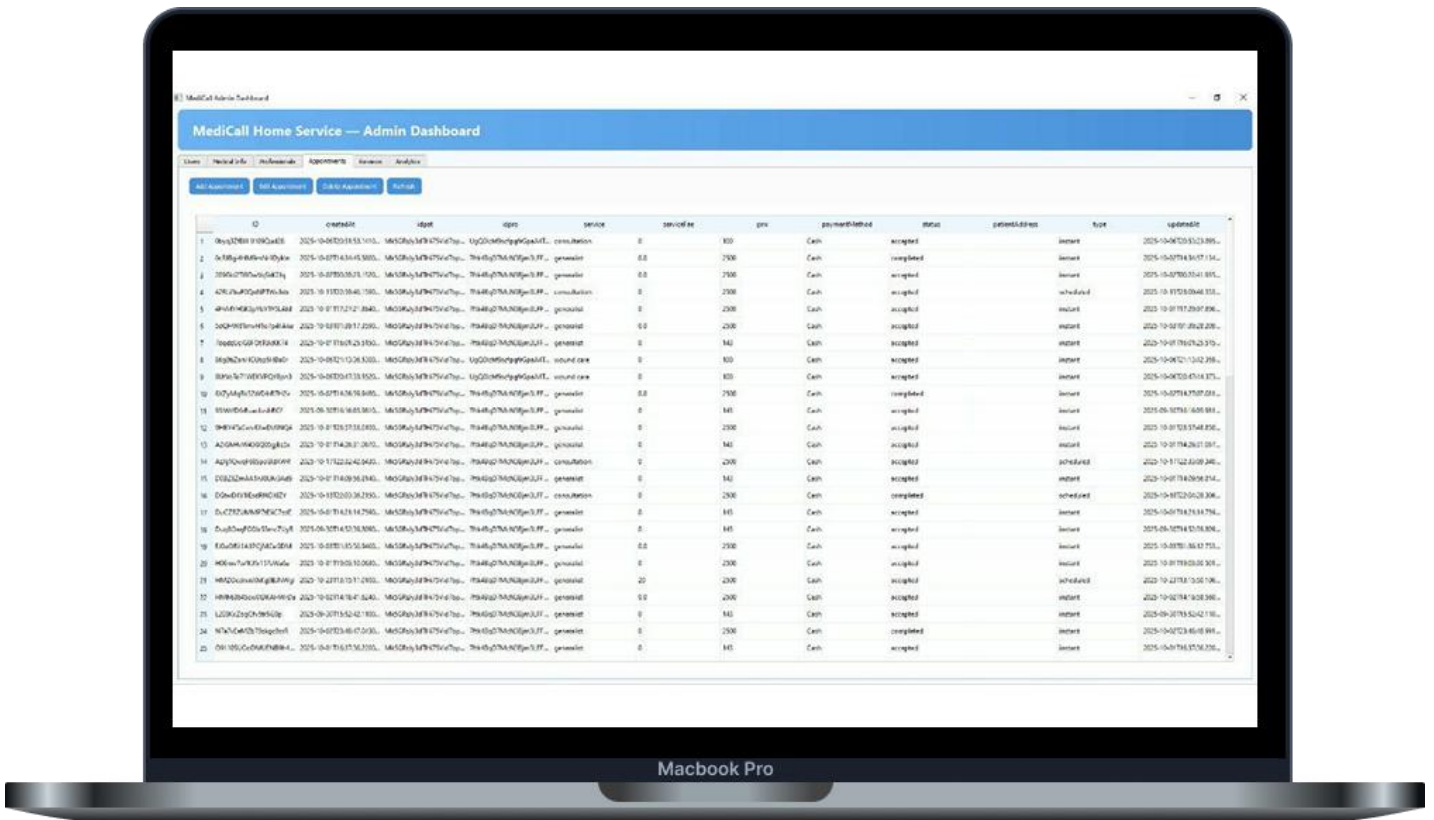


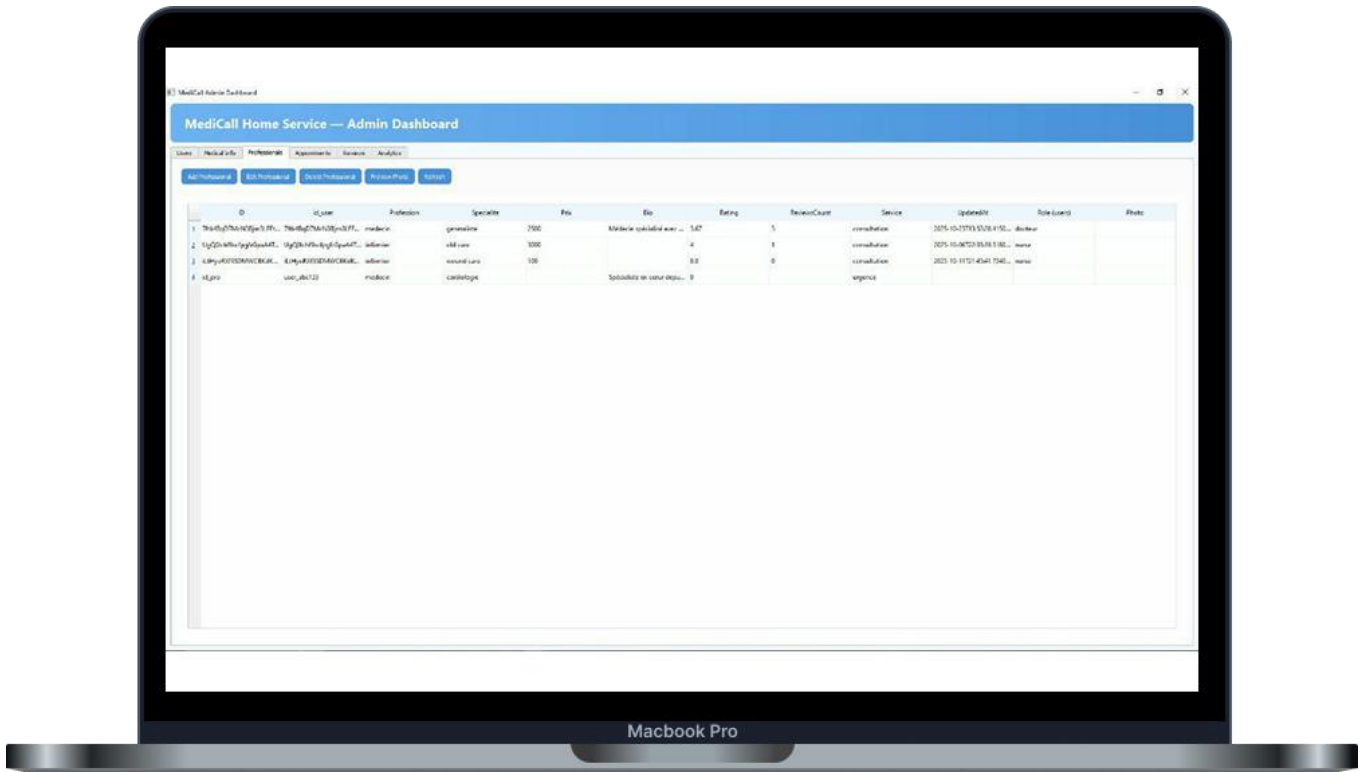
Figure 5.8 Login admin



(a) Onboarding 1



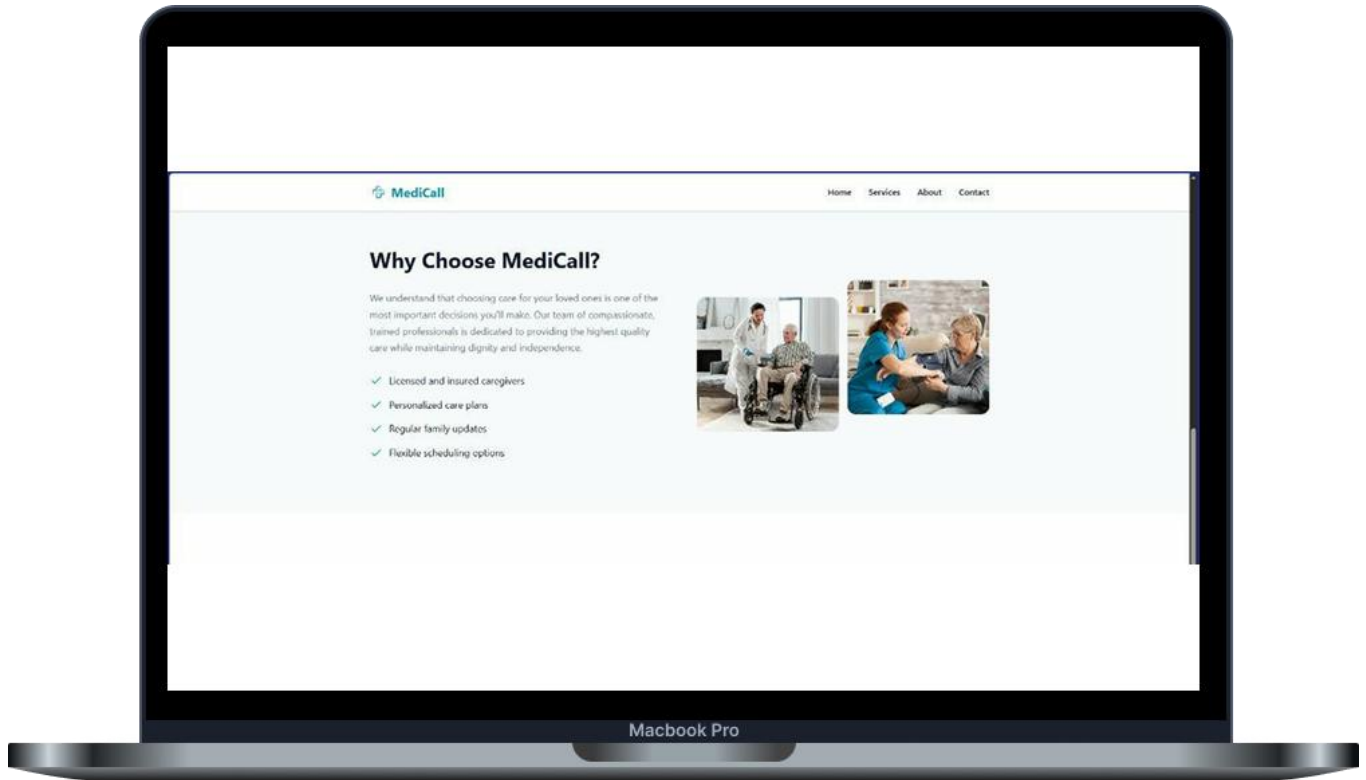
(b) Onboarding 2



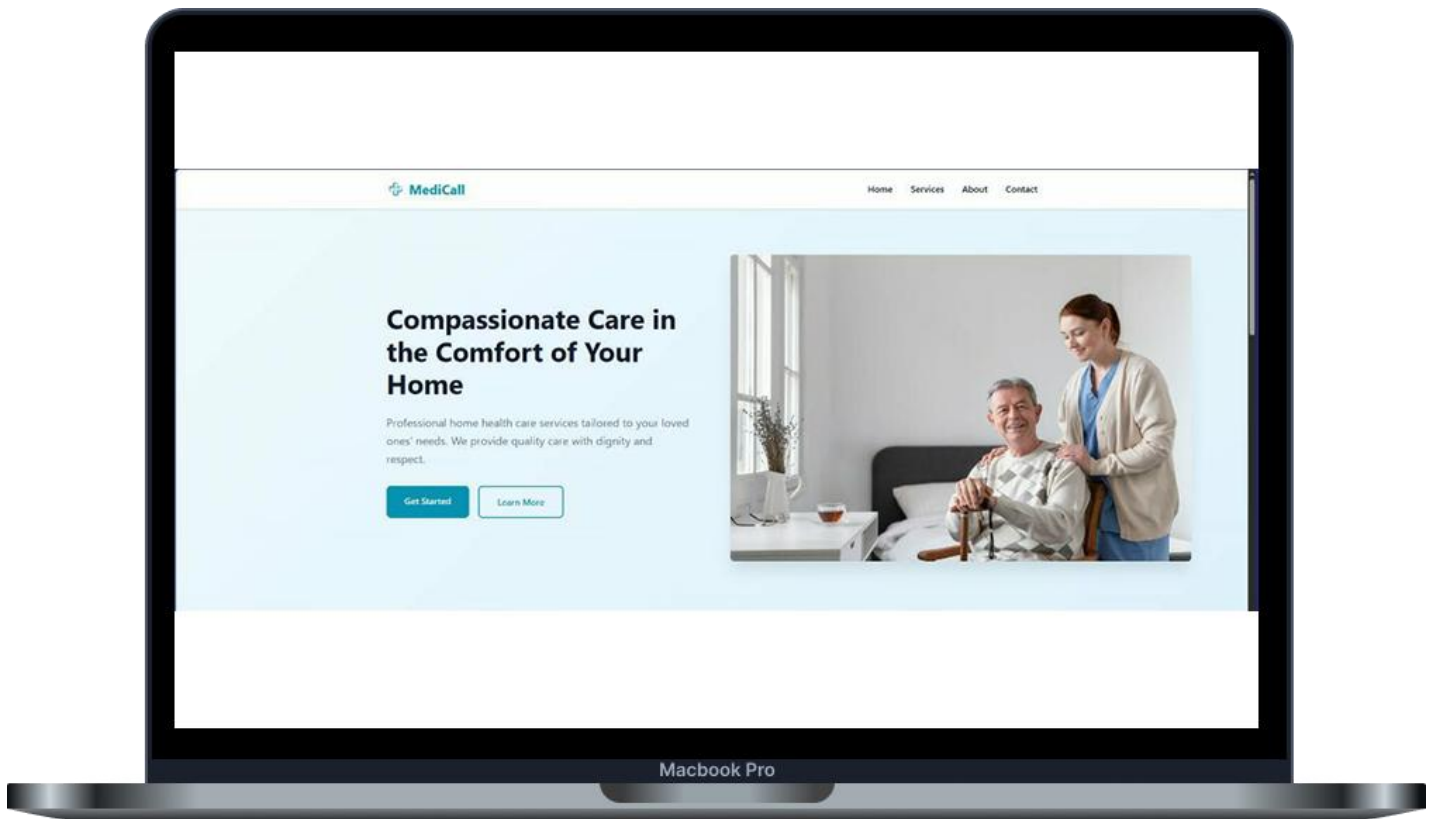
(c) Onboarding 3

Figure 5.9 Admin Dashboard

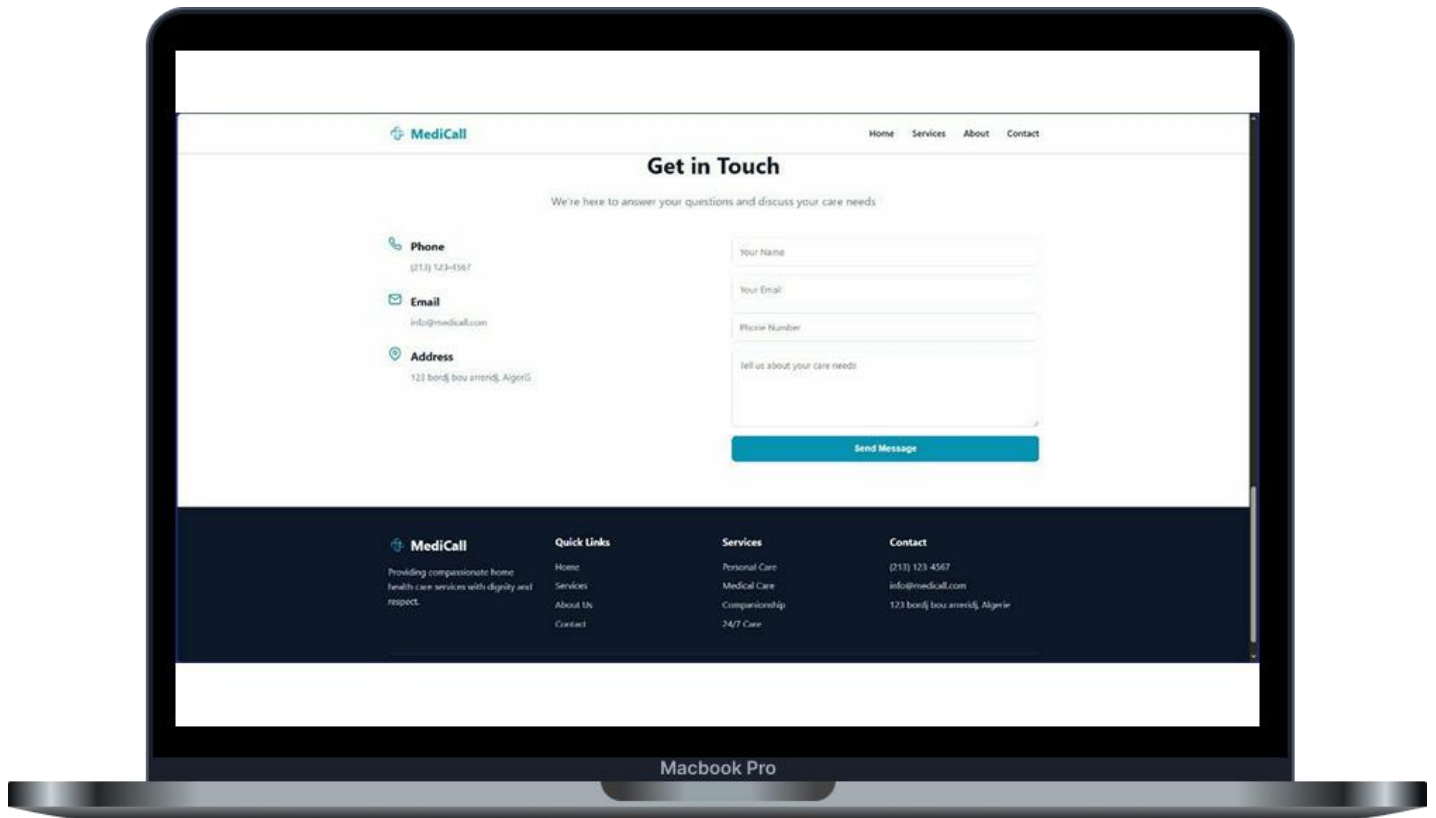
5.7.2 Notre site WEB



(a) Onboarding 1



(b) Onboarding 2



(c) Onboarding 3

Figure 5.10 Interface de site

5.7.3 Notre application MOBILE

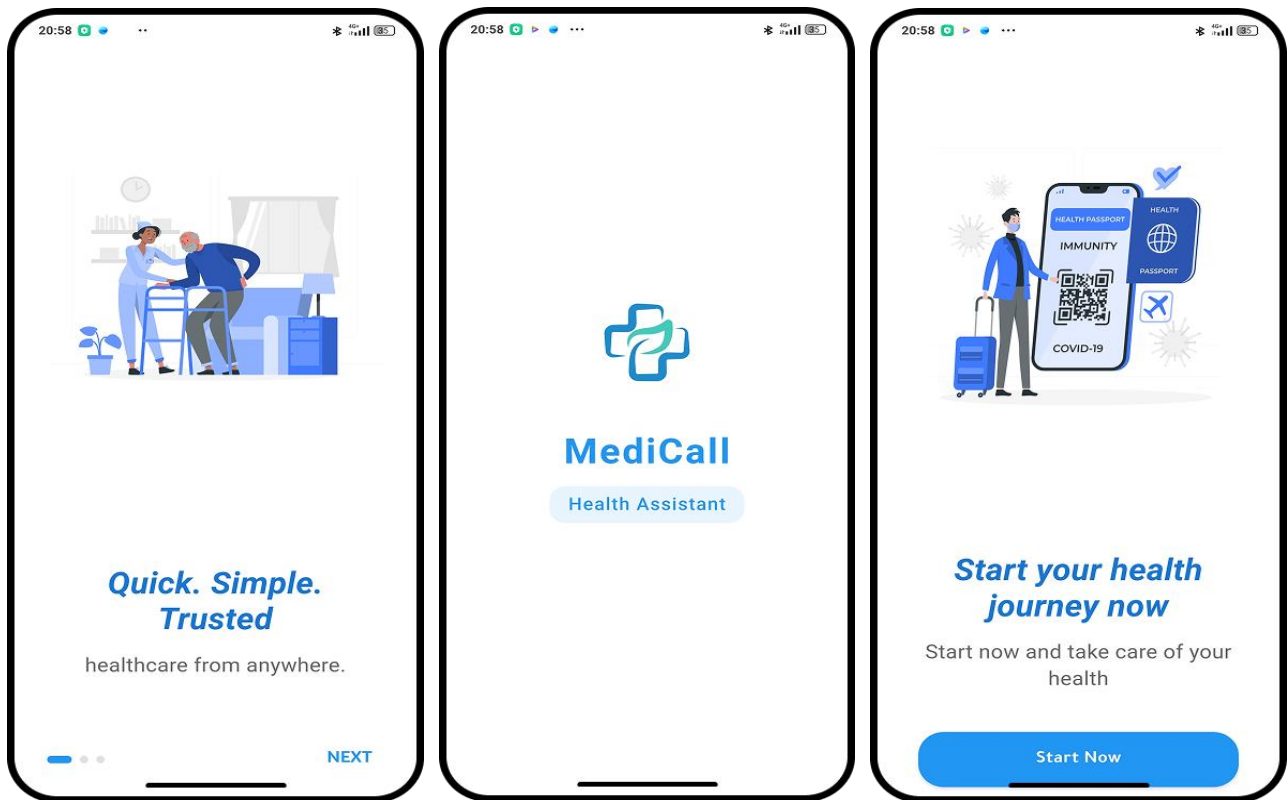


Figure 5.11 Interface d'application

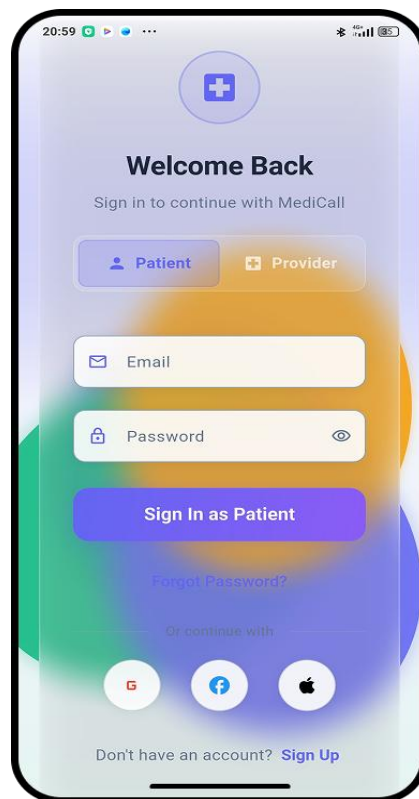


Figure 5.12 Ecran de connexion

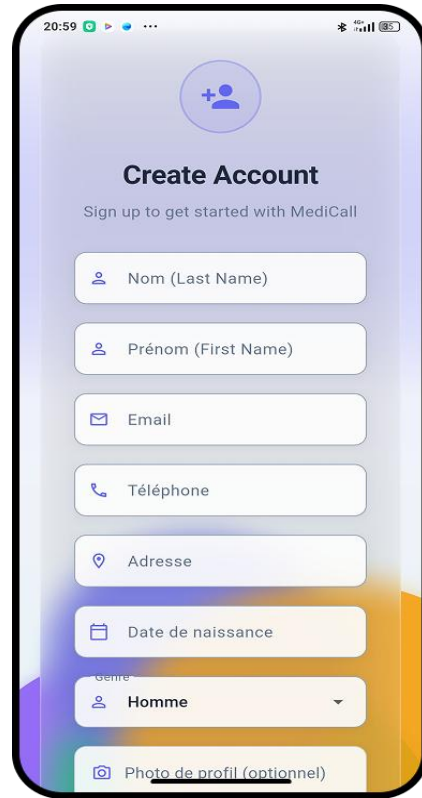
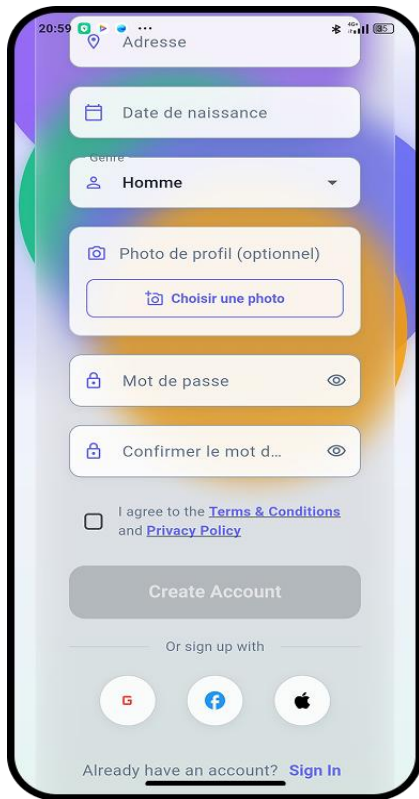


Figure 5.13 Creet un comte

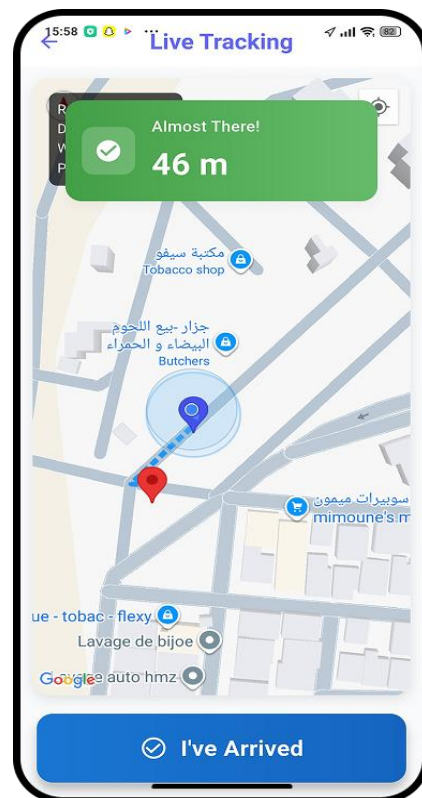
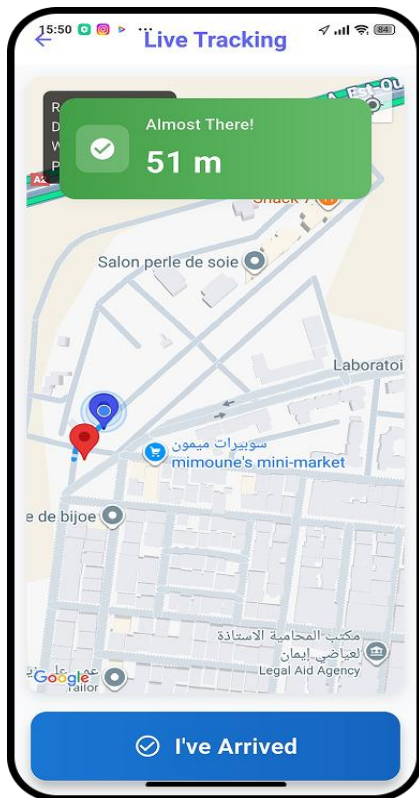


Figure 5.14 Tracage en direct

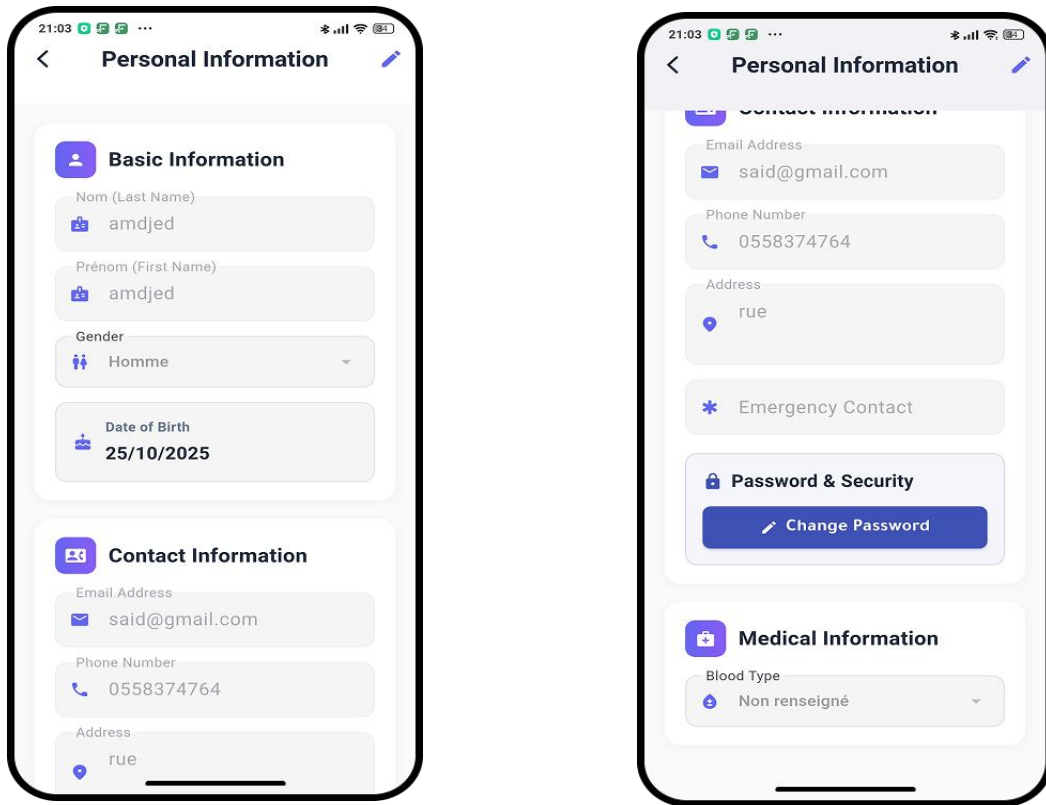


Figure 5.15 Information personnels

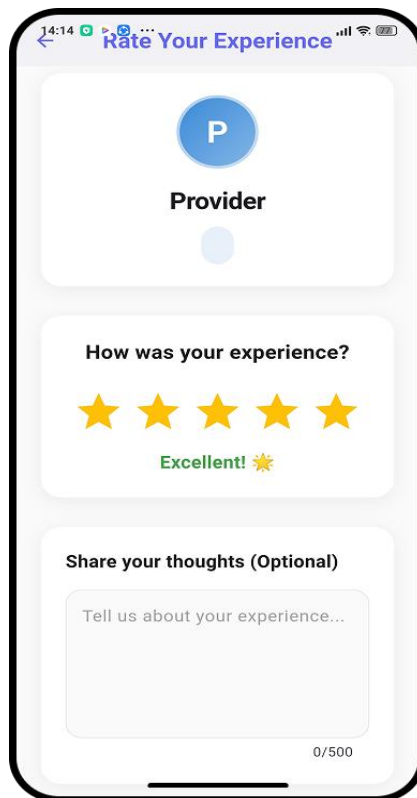


Figure 5.16 Evaluation des performance

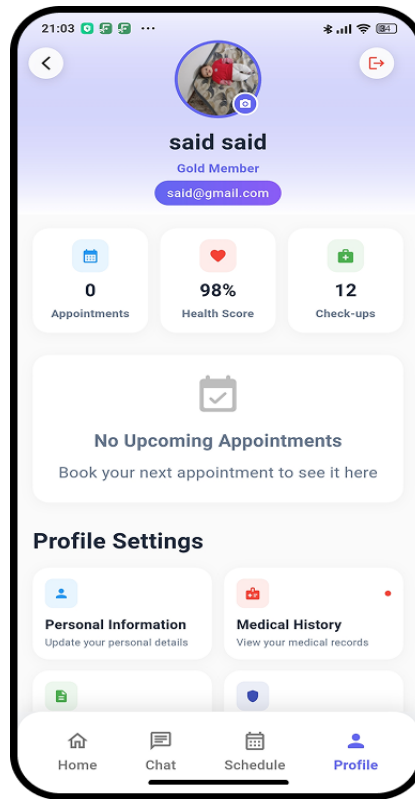


Figure 5.17 Profile de patient

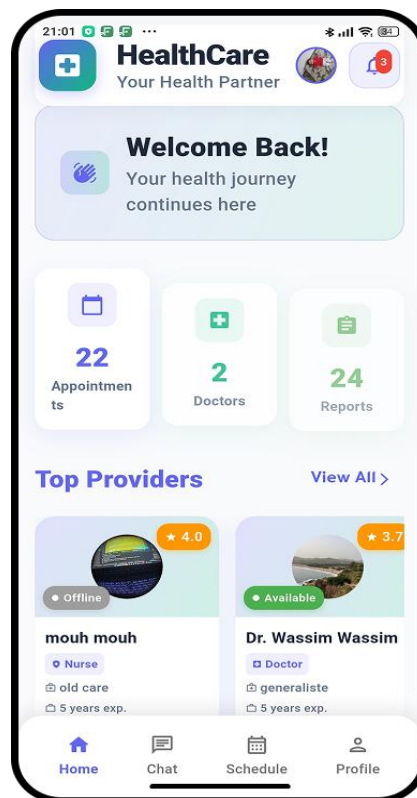


Figure 5.18 Ecran d'accueil patient

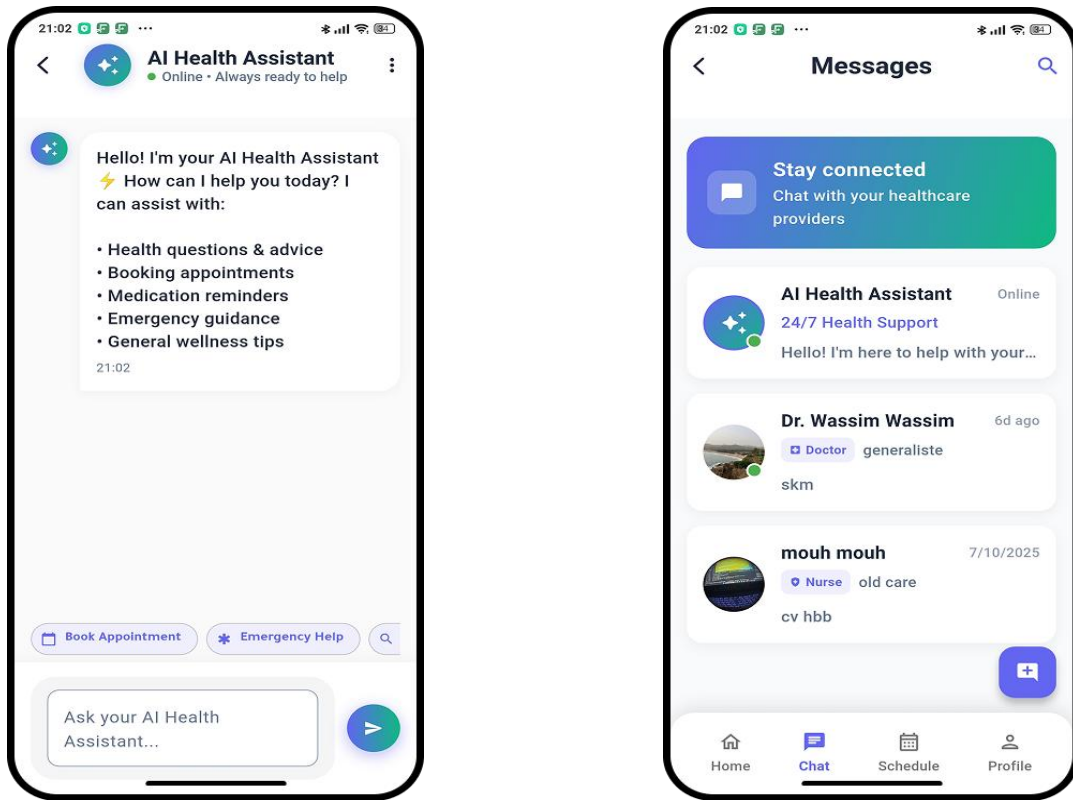


Figure 5.19 Discussion IA

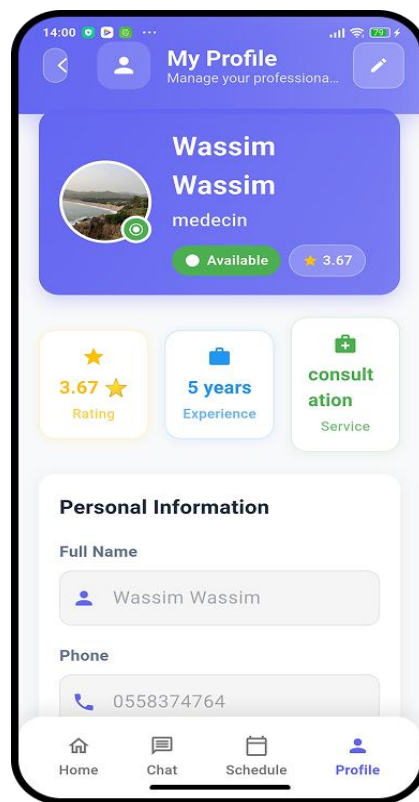


Figure 5.20 Profile de doctor

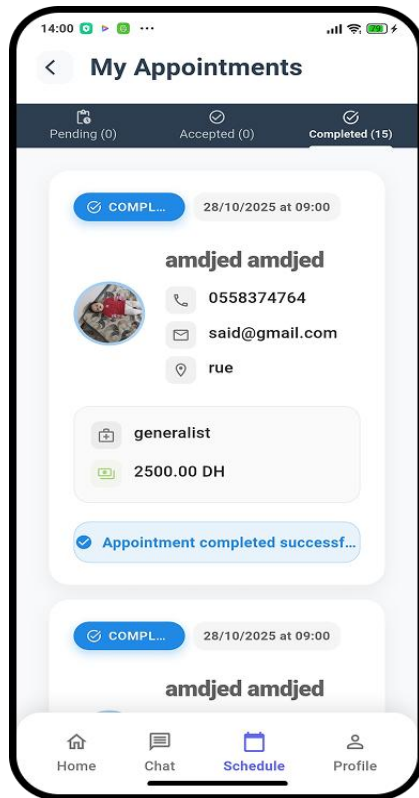
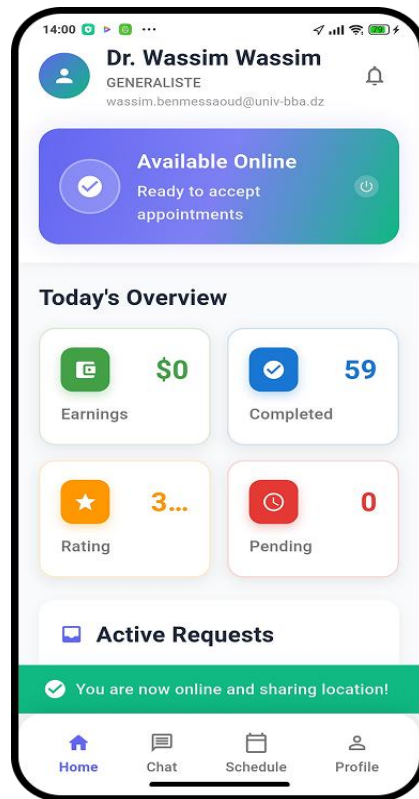


Figure 5.21 Nomination

Figure 5.22 doctore en line



Chapitre 6

Conclusion générale

6.1 Contributions

Dans ce projet, nous nous sommes intéressés au défi de la numérisation et de la modernisation des services de soins infirmiers à domicile en Algérie. Le système MediCall constitue une contribution significative à ce domaine en offrant une solution intégrée et intelligente qui relie les patients, les infirmiers et les administrateurs à travers une plateforme numérique unifiée.

Nos principales contributions se résument comme suit :

- **Conception et développement d'une plateforme complète** combinant une application Web et une application mobile multiplateforme, destinée à faciliter la gestion des soins à domicile et à améliorer la communication entre les acteurs du domaine de la santé.
- **Implémentation d'un serveur intelligent** basé sur un modèle de langage (LLaMA-3.2-1B affiné), capable de fournir des réponses médicales fiables, cohérentes et contextualisées, adaptées au domaine des soins infirmiers.
- **Automatisation du processus de prestation des soins**, depuis la demande initiale du patient jusqu'à la validation et l'évaluation du service, garantissant rapidité, traçabilité et sécurité.
- **Amélioration de la qualité du suivi médical** grâce à des fonctionnalités de géolocalisation, de planification dynamique et de notification en temps réel.

Ainsi, notre travail contribue à poser les bases d'un écosystème de santé numérique intelligent, adapté aux besoins des patients et du personnel infirmier, tout en respectant les normes de confidentialité et de sécurité des données médicales.

6.2 Critique du travail

Bien que le système MediCall représente une avancée significative dans la digitalisation des soins à domicile, certaines limites persistent.

- **Performance et scalabilité** : l'optimisation de la charge du serveur et du temps de réponse lors de pics d'utilisation devra être améliorée afin d'assurer une expérience fluide pour tous les utilisateurs.
- **Expérience utilisateur (UX/UI)** : malgré une interface fonctionnelle, des améliorations peuvent être apportées pour rendre la navigation plus intuitive et ergonomique, notamment sur la version mobile.
- **Sécurité et confidentialité** : la gestion des données médicales reste un enjeu critique. Des mécanismes de chiffrement plus avancés et une conformité stricte aux réglementations sur la protection des données doivent être continuellement renforcés.
- **Intégration externe** : l'interopérabilité avec les systèmes de santé publics et les cliniques privées constitue un axe d'amélioration essentiel pour une adoption à grande échelle.

En résumé, bien que les fondations soient solides, des efforts supplémentaires sont nécessaires pour atteindre un niveau optimal de robustesse, d'accessibilité et de sécurité.

6.3 Travaux futurs et perspectives

Les perspectives d'évolution du projet sont multiples et prometteuses.

- **Intégration avancée de l'intelligence artificielle** : nous prévoyons de renforcer les capacités du serveur intelligent en y intégrant un moteur de recommandation personnalisé et un assistant virtuel capable d'interagir naturellement avec les patients et les infirmiers.
- **Extension des fonctionnalités** : l'ajout de modules tels que la gestion des urgences, le suivi des traitements médicamenteux et l'analyse prédictive de l'état de santé permettront d'enrichir l'écosystème.
- **Interopérabilité avec les institutions de santé** : une future connexion aux plateformes hospitalières publiques et privées permettra un échange sécurisé et automatisé des données de soins.
- **Accessibilité nationale** : le déploiement progressif du système à travers différentes régions d'Algérie, en plusieurs langues (français, arabe, tamazight), vise à garantir un accès équitable à la santé numérique pour tous.

À long terme, la sensibilisation des acteurs de la santé, la formation continue des utilisateurs, et l'évolution des technologies de l'IA et du cloud computing constitueront des leviers majeurs pour le succès et la durabilité de MediCall.

En conclusion, ce projet marque une étape clé dans le processus de transformation digitale du secteur de la santé en Algérie. Il démontre que la combinaison de la technologie, de l'intelligence artificielle et de la vision humaine du soin peut contribuer à bâtir une société plus connectée, plus solidaire et plus réactive aux besoins réels des citoyens.

Bibliographie

- [1] Andrew S. Tanenbaum and David J. Wetherall. *Computer Networks*. Prentice Hall, 5th edition, 2011.
- [2] James F. Kurose and Keith W. Ross. *Computer Networking : A Top-Down Approach*. Pearson, 7th edition, 2017. Référence moderne sur les principes du modèle client-serveur et les protocoles du Web.
- [3] Roy T. Fielding and Richard N. Taylor. Principled design of the modern web architecture. *ACM Transactions on Internet Technology*, 2(2) :115–150, 2002. Article fondateur décrivant l’architecture REST et le protocole HTTP.
- [4] Tim Berners-Lee. Uniform resource locators (url). Technical Report RFC 1738, IETF, 1994.
- [5] W3C Consortium. Html5 specification. World Wide Web Consortium (W3C), 2017. Spécification officielle du langage HTML5.
- [6] Robin Nixon. *Learning PHP, MySQL and JavaScript : With jQuery, CSS and HTML5*. O’Reilly Media, 5th edition, 2018. Ouvrage de référence sur la création de sites web dynamiques avec PHP et JavaScript.
- [7] Stefan Tilkov and Steve Vinoski. *REST : From Research to Practice*. Springer, 2017. Référence académique sur l’architecture REST et les API web modernes.
- [8] Adrian Holovaty and Jacob Kaplan-Moss. *The Definitive Guide to Django : Web Development Done Right*. Apress, 2nd edition, 2009. Guide technique sur le framework Django et la programmation côté serveur en Python.
- [9] Docker Inc. Docker documentation. <https://docs.docker.com/>, 2024. Documentation officielle de Docker, outil de conteneurisation des applications.
- [10] David Flanagan. *JavaScript : The Definitive Guide*. O’Reilly Media, 7th edition, 2020. Référence complète sur le langage JavaScript et la programmation côté client.

Bibliographie

- [11] Jeremy Keith and Rachel Andrew. *HTML5 for Web Designers*. A Book Apart, 2010. Introduction concise aux principes et nouveautés du HTML5 dans la conception web moderne.
- [12] Chris Coyier and Estelle Weyl. *CSS : The Definitive Guide*. O'Reilly Media, 4th edition, 2020. Référence complète sur les techniques modernes de mise en forme avec CSS.
- [13] Tim Berners-Lee and Mark Fischetti. *Weaving the Web*. Harper San Francisco, San Francisco, USA, 2000.
- [14] Roger S. Pressman and Bruce R. Maxim. *Software Engineering : A Practitioner's Approach*. McGraw-Hill Education, 9th edition, 2021. Référence complète sur les architectures logicielles et le développement d'applications.
- [15] Li Zhou, Wei Chen, and Raj Kumar. Artificial intelligence and cloud computing : Synergies and future trends. *Journal of Cloud Computing*, 11(1) :45–58, 2022. Étude sur l'intégration de l'IA et du cloud computing dans les applications modernes.
- [16] Stefan Tilkov and Steve Vinoski. *REST : From Research to Practice*. Springer, 2017. Présentation approfondie de l'architecture REST utilisée pour les services externes.
- [17] Frank Buschmann, Kevin Henney, and Douglas C. Schmidt. *Pattern-Oriented Software Architecture : A Pattern Language for Distributed Computing*, volume 4. Wiley, 2007. Référence détaillée sur les architectures logicielles et les modèles de conception.
- [18] The Qt Company. Qt framework documentation. <https://www.qt.io/docs>, 2023.
- [19] Organisation Mondiale de la Santé (OMS). *Rapport sur la santé numérique 2023*. Genève, 2023.
- [20] Ministère de la Santé Algérien. *Plan National de la e-santé 2022–2026*. Alger, 2022.
- [21] Organisation de Coopération et de Développement Économiques (OCDE). *Home Care and Ageing Population Report*. Paris, 2021.
- [22] Direction des Ressources Humaines du Secteur de la Santé. *Statistiques du personnel paramédical en Algérie*. Alger, 2022.
- [23] S. Chen et al. *AI and IoT Integration in Smart Healthcare Systems*. Elsevier, 2023.

Bibliographie

- [24] Craig Larman. *Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2012.
- [25] Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 2nd edition, 2005.
- [26] Object Management Group (OMG). Unified modeling language (uml) 2.5.1 specification. <https://www.omg.org/spec/UML/2.5.1/>, 2017.
- [27] Martin Fowler. *UML Distilled : A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 3rd edition, 2004.
- [28] Ramez Elmasri and Shamkant Navathe. *Fundamentals of Database Systems*. Pearson, 7th edition, 2016.
- [29] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 1970.
- [30] Jan L. Harrington. *Relational Database Design and Implementation*. Morgan Kaufmann, 2016.
- [31] Vercel Inc. Next.js documentation : React framework for production, 2024. Accessed : 2025-10-21.
- [32] Microsoft. Typescript : Javascript with syntax for types, 2023. Accessed : 2025-10-21.
- [33] World Wide Web Consortium (W3C). World wide web consortium standards. <https://www.w3.org/>, 2023.
- [34] Tailwind Labs. Tailwind css documentation, 2024. Accessed : 2025-10-21.
- [35] Shadcn Team. Ui components built with tailwind css and radix ui, 2024. Accessed : 2025-10-21.
- [36] Drizzle Team. Drizzle orm documentation, 2024. Accessed : 2025-10-21.
- [37] Zod Team. Zod : Typescript-first schema validation, 2023. Accessed : 2025-10-21.
- [38] M. Jones, J. Bradley, and N. Sakimura. Json web token (jwt). RFC 7519, IETF, 2015. Accessed : 2025-10-21.

Bibliographie

- [39] Docker Inc. Docker documentation : Build, ship, and run any app, anywhere, 2024. Accessed : 2025-10-21.
- [40] Google LLC. Flutter documentation : Build apps for any screen, 2024. Accessed : 2025-10-21.
- [41] The PHP Group. Php manual : The official documentation for php 8.x, 2023. Accessed : 2025-10-21.
- [42] Google LLC. *Firebase Realtime Database Documentation*. Firebase, 2024. Accessed : 2025-10-21.
- [43] Electron Team. Electron.js documentation. <https://www.electronjs.org>, 2024.
- [44] GitHub Documentation Team. Electron documentation : Apis and development guide, 2024. Accessed : 2025-10-21.
- [45] SQLite Consortium. Sqlite documentation : The most widely deployed sql database engine in the world, 2023. Accessed : 2025-10-21.
- [46] Roy T. Fielding. Restful web services principles, 2024. Accessed : 2025-10-21.
- [47] Postman Inc. Postman api platform documentation, 2023. Accessed : 2025-10-21.
- [48] Peter Rob and Carlos Coronel. *Database Systems : Design, Implementation, and Management*. Cengage Learning, 13th edition, 2019.
- [49] Scott Chacon and Ben Straub. Pro git, 2nd edition, 2023. Accessed : 2025-10-21.
- [50] Seats.io Team. Seats.io api documentation : Interactive seating charts, 2024. Accessed : 2025-10-21.
- [51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [52] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen,

Bibliographie

Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

- [53] Meta AI Research. Llama 3.2-1b technical overview. Technical report, Meta Platforms, Inc., Menlo Park, CA, USA, 2024.
- [54] Laura Smith and Hao Zhang. *Data Engineering for AI Systems*. Springer, Berlin, Germany, 2023.
- [55] Robert Johnson. *Efficient Data Curation for Healthcare NLP*. ACM Press, New York, USA, 2022.
- [56] John Doe. *Medical Knowledge Extraction for Clinical NLP*. Elsevier, Amsterdam, Netherlands, 2023.
- [57] OpenAI Research. Data annotation and cleaning standards for llms. Technical report, OpenAI, San Francisco, CA, USA, 2023.
- [58] Patricia A. Potter and Anne Griffin Perry. *Fundamentals of Nursing*. Elsevier, St. Louis, MO, USA, 10th edition, 2020.
- [59] Barbara Kozier, Glenora Erb, Audrey Berman, and Shirlee Snyder. *Fundamentals of Nursing : Concepts, Process, and Practice*. Pearson Education, London, UK, 9th edition, 2017.
- [60] Ruth F. Craven and Constance J. Hirnle. *Fundamentals of Nursing : Human Health and Function*. Lippincott Williams & Wilkins, Philadelphia, PA, USA, 8th edition, 2016.

Annexe
