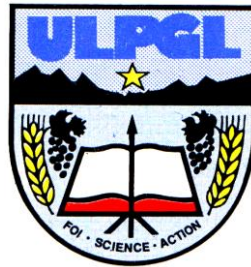


UNIVERSITE LIBRE DES PAYS DES GRANDS LACS
FACULTE DE SCIENCES ET TECHNOLOGIES
APPLIQUÉES

DEPARTEMENT DE GENIE ELECTRIQUE ET INFORMATIQUE



BP. 368 GOMA

www.ulpgl.net

CONCEPTION ET IMPLEMENTATION
D'UNE PLATE-FORME DE
CONSULTATION MÉDICALE EN LIGNE

Par **KOLIAMA ASSUMANI Dieudonné**

Nathanaël

Travail présenté en vue de l'obtention du Diplôme de
License en Sciences de l'Ingénieur

Option : Génie Informatique

Directeur : MSc. Ir. NZANZU VINGI Patrick

Encadreur : MSc. Ir. KRAME KADURHA

ANNÉE ACADÉMIQUE 2023- 2024

Epigraphe

« La technologie ne remplace pas les médecins, mais elle leur donne les moyens d'être plus proches, plus efficaces et plus humains. »

Eric Topol

Dédicace

À nos chers parents AWEWE ANYAMUSIGIO BAPIANI Augustin et OTONO KIBAYA Jolith.

KOLIAMA ASSUMANI Dieudonné Nathanaël

Remerciements

Notre profonde gratitude s'adresse avant tout à Dieu, Maître des temps et des circonstances, pour ce travail et tous les bienfaits qu'Il nous accorde gratuitement.

Nos vifs remerciements vont également au Master Ir. NZANZU VINGI Patrick, qui, malgré ses nombreuses responsabilités, a accepté de diriger ce travail avec bienveillance.

Nous exprimons aussi toute notre reconnaissance à notre Encadreur, le Master Ir. KRAME KADURHA David, pour ses orientations précieuses, son accompagnement attentif et ses remarques constructives qui ont enrichi ce travail.

Nos sincères remerciements à l'ULPGL/Goma, à travers la Faculté des sciences et technologies, ainsi qu'à ses autorités académiques et administratives pour l'environnement de travail et les ressources mises à notre disposition tout au long de notre formation.

Nous exprimons nos remerciements sincères à nos chers parents, AWEWE ANYAMUSIGIO BAPIANI Augustin et OTONO KIBAYA Jolith, pour leurs sacrifices constants durant notre parcours, ainsi qu'à toute notre famille, notamment ma grand-mère WAWINA ADOLPHE et nos frères, pour leur soutien moral, leur amour et leurs encouragements inestimables.

Nous remercions chaleureusement notre famille et nos connaissances, notamment KARUME Jean-Paul, AMISI Rebecca, Tantine Théthé, EYENGA Virginie, ILAKA John, Mopipi Féli, ainsi que tous ceux dont les noms n'ont pas été mentionnés, pour leur aide précieuse, qu'elle soit financière, morale, par des conseils avisés ou par des encouragements constants. Leur générosité, leur soutien et leur bienveillance ont été d'un grand secours tout au long de notre parcours.

Nos remerciements s'adressent aussi à nos amis et camarades qui, d'une manière ou d'une autre, nous ont soutenu et accompagné dans la réalisation de ce travail.

KOLIAMA ASSUMANI Dieudonné Nathanaël

Résumé

Face aux difficultés d'accès aux soins spécialisés à Goma, ce travail part de l'hypothèse qu'une plateforme de consultation médicale en ligne, assistée par l'intelligence artificielle, peut améliorer la prise en charge préliminaire des patients et optimiser l'orientation vers des spécialistes. Ce projet propose ainsi la conception d'une plateforme intégrant un système de diagnostic préliminaire basé sur le modèle de langage LLaMA. Le chatbot médical analyse les symptômes décrits par les patients pour fournir un diagnostic initial et orienter vers des spécialistes. La plateforme inclut également des modules d'authentification sécurisée, de gestion des rendez-vous, d'historiques médicaux, d'ordonnances électroniques et d'administration. Une API a été développée pour faciliter l'intégration du système par d'autres services médicaux. Les résultats obtenus démontrent la viabilité de la solution et ouvrent de nouvelles perspectives pour la digitalisation de la santé dans des régions sous-desservies.

Mots-clés : Intelligence artificielle, Chatbot médical, LLaMA, Diagnostic automatisé, Santé numérique, API.

Abstract

Faced with the difficulties of accessing specialized healthcare in Goma, this work is based on the hypothesis that an online medical consultation platform, assisted by artificial intelligence, can improve the preliminary management of patients and optimize their referral to specialists. This project thus proposes the design of a platform integrating a preliminary diagnostic system based on the LLaMA language model. The medical chatbot analyzes the symptoms described by patients to provide an initial diagnosis and guide them toward appropriate specialists. The platform also includes modules for secure authentication, appointment management, medical history tracking, electronic prescriptions, and administration. An API has been developed to facilitate the integration of the system by other medical services. The results obtained demonstrate the viability of the solution and open new perspectives for the digitalization of healthcare in underserved regions.

Keywords: Artificial Intelligence, Medical Chatbot, LLaMA, Automated Diagnosis, Digital Health, API.

Table des matières

Epigraphe.....	i
Dédicace	ii
Remerciements	iii
Résumé	iv
Table des matières	vi
Sigles et abréviations.....	ix
Liste des tableaux	Error! Bookmark not defined.
Liste des figures.....	xi
0. Introduction générale.....	1
0.1. Contexte/Généralités sur le thème.....	1
0.2. Identification et formulation du problème	2
0.3. Questions de recherche.....	2
0.4. Formulation des hypothèses	3
0.5. Justification du choix du sujet et motivations	3
0.6. Énoncé des objectifs de recherche	4
0.6.1. L'objectif général	4
0.7. Méthodologie et délimitation du travail	5
0.8. Structure du mémoire/ Subdivision du travail.....	5
Chapitre 1 Fondements théoriques et contexte de la télémédecine à Goma.....	7
1.1 Introduction partielle	7
1.2. Etat de l'art	7
1.3. Pratiques traditionnelles d'accès aux soins médicaux.....	8

1.3.1.	Consultation de Premier niveau :	8
1.3.2.	Référencement vers des Spécialistes :	8
1.3.3.	Gestion Manuelle des Rendez-vous et des Dossiers Médicaux:	8
1.4.	Défis d'accès aux soins spécialisés à Goma	9
1.5.	Besoin d'une solution numérique pour améliorer l'accès aux soins	10
1.5.1.	Concepts de base de la télémédecine et des systèmes de consultation en ligne.....	10
1.6.	Chatbot et Intelligence Artificielle dans le Diagnostic Médical	12
1.7.	Utilisation de Llama pour le Diagnostic Médical	19
1.7.1.	Qu'est-ce que Llama ?.....	19
1.7.2.	Caractéristiques de Llama	19
1.8.	Conclusion Partielle	20
Chapitre 2 Analyse et Conception du Système		21
2.1.	Introduction	21
2.2.	Analyse des besoins	21
2.2.1.	Besoins fonctionnels	21
2.2.2.	Besoins non fonctionnels	23
2.3.	Diagramme de cas d'utilisation.....	23
2.4.	Documentation de différents cas d'utilisation.....	26
2.5.	Diagramme de classes	36
2.5.1.	Structure des classes	36
2.6.	Diagramme de séquences	40
2.6.1.	Authetification des utilisateurs.....	40
2.6.2.	Diagnostic préliminaire automatisé.....	42
2.6.3.	Mise à jour du diagnostic par le médecin.....	42
2.6.4.	Ordonnance électronique.....	43
2.6.5.	Supervision et génération de rapports	44
2.7.	Conclusion Partielle	45
Chapitre 3 : Réalisation de l'Application Web.....		46
3.1.	Introduction	46

3.2. Technologies utilisées	46
3.3. Architecture du système	48
3.4. Méthodologie de réalisation	49
3.5. Évaluation des performances	50
3.6. Présentation du fonctionnement de l'application	52
3.6.1. Interface Patient.....	53
3.6.2. Espace Médecin.....	59
3.6.3. Espace admin.....	62
3.7 Conclusion partielle.....	65
Conclusion générale	66
Bibliographie	68
ANNEXES	71

Sigles et abréviations

API: Application Programming Interface

BART: Bidirectional and Auto-Regressive Transformers

BERT: Bidirectional Encoder Representations from Transformers

CSS: Cascading Style Sheets

CU : Cas d'Utilisation

DALL-E : Ce nom est un jeu de mots basé sur Salvador Dalí (le peintre surréaliste) et WALL·E (le robot du film de Pixar).

DB: Data Base

GPT: Generative Pre-trained Transformer

HTTP : Hyper Text Transfer Protocol

IA : Intelligence Artificielle

JSON : JavaScript Object Notation

LLaMA: Large Language Model Meta AI

LLM: Large Language Model

ML: Machine Learning

NLP: Natural Language Processing

NoSQL: Not Only SQL

OMS : Organisation mondiale de la santé

REST: REpresentational State Transfer

RLHF: Reinforcement Learning with Human Feedback

RoBERTa: Robustly Optimized BERT Approach

T5: Text-To-Text Transfer Transformer

UML: Unified Modeling Languag

Liste des tableaux

<i>Tableau 2-0-1 : Documentation de cas d'utilisation : « Créer un compte »</i>	27
<i>Tableau 2-0-2: Documentation de cas d'utilisation : « Gérer mot de passe »</i>	28
<i>Tableau 2-0-3: Documentation de cas d'utilisation : « Diagnostic préliminaire »</i>	29
<i>Tableau 2-0-4: Documentation de cas d'utilisation : « Prendre un rendez-vous »</i>	30

Liste des figures

<i>Figure 2-1: Diagramme de cas d'utilisation de patient</i>	24
<i>Figure 2-2: Diagramme de cas d'utilisation de Médecin</i>	25
<i>Figure 2-3: Diagramme de cas d'utilisation d'Administrateur</i>	26
<i>Figure 2-4 : Diagramme de classe du système</i>	40
<i>Figure 2-5 : Diagramme de séquence d'authentification</i>	41
<i>Figure 2-6 : Diagramme de séquence de diagnostic préliminaire</i>	42
<i>Figure 2-7 : Diagramme de séquence de mise à jour de diagnostic</i>	43
<i>Figure 2-8 : Diagramme de séquence d'ordonnance électronique</i>	44
<i>Figure 2-9 : Diagramme de séquence de supervision et rapport</i>	45
<i>Figure 3-1: Section Héros de la page d'accueil (Landing Page)</i>	53
<i>Figure 3-2: Section des services proposés sur la page d'accueil</i>	53
<i>Figure 3-3: Section "À propos" de la page d'accueil</i>	54
<i>Figure 3-4 Présentation des médecins sur la page d'accueils</i>	54
<i>Figure 3-5: Section des témoignages utilisateurs</i>	54
<i>Figure 3-6: Interface de connexion des utilisateurs</i>	55
<i>Figure 3-7 : Interface d'inscription des utilisateurs</i>	56
<i>Figure 3-8 : Interface de discussion entre le patient et le chatbot médical</i>	56
<i>Figure 3-9: Affichage des résultats du diagnostic préliminaire</i>	57
<i>Figure 3-10: Bouton de recherche de médecin disponible</i>	57
<i>Figure 3-11: Confirmation de l'attribution d'un médecin spécialiste</i>	58
<i>Figure 3-12: Liste des rendez-vous du patient</i>	58
<i>Figure 3-13: Interface de consultation de l'historique médical</i>	59
<i>Figure 3-14: Liste des diagnostics en attente de validation</i>	59
<i>Figure 3-15: Information du patient avant l'envoi d'ordonnance</i>	60
<i>Figure 3-16: Formulaire de prescription d'une ordonnance électronique</i>	60
<i>Figure 3-17 Interface du profil du médecin</i>	61
<i>Figure 3-18: Interface de gestion des utilisateurs</i>	62

<i>Figure 3-19: Formulaire de création d'un compte médecin</i>	62
<i>Figure 3-20: Interface de génération automatique des rapports</i>	64

0. Introduction générale

0.1. Contexte/Généralités sur le thème

La santé publique est un secteur vital pour le développement et le bien-être des sociétés. Toutefois, dans de nombreuses régions du monde, notamment dans les pays en développement, l'accès aux soins de santé demeure un défi majeur. Les infrastructures médicales limitées, le manque de spécialistes et les longues distances à parcourir pour obtenir des soins médicaux contribuent à aggraver cette situation [1]. À Goma, comme dans plusieurs autres villes africaines, ces problèmes sont exacerbés par des conditions économiques et sociales difficiles, rendant difficile l'accès rapide à des consultations médicales appropriées.

Dans ce contexte, l'avènement des technologies numériques offre une opportunité unique pour améliorer l'accès aux soins médicaux. Le développement d'un système de consultation médicale en ligne vise à combler le fossé entre les besoins des patients et la disponibilité des professionnels de santé. En intégrant l'intelligence artificielle, il est désormais possible d'automatiser certaines étapes du diagnostic médical et de faciliter la mise en relation des patients avec des médecins spécialisés. Un tel système permettrait non seulement de répondre aux besoins immédiats des patients, mais aussi de désengorger les structures médicales surchargées.

Ce projet de mémoire s'inscrit donc dans une démarche visant à développer un système de consultation médicale en ligne qui utilise un chatbot capable de fournir des diagnostics préliminaires à partir des symptômes décrits par l'utilisateur, tout en permettant une validation ou modification par des médecins.

0.2. Identification et formulation du problème

Cette problématique s'articule autour de l'absence de systèmes de consultation médicale en ligne dans des villes comme Goma limite l'accès rapide à des soins médicaux spécialisés et génère une surcharge des médecins généralistes. Cette situation crée une insatisfaction notable chez les patients, qui se retrouvent confrontés à des difficultés pour obtenir un diagnostic rapide et des soins appropriés.

L'exploration empirique de ce problème est nécessaire car elle permettrait non seulement de répondre aux besoins de la population en facilitant l'accès aux soins via une solution numérique, mais aussi d'optimiser l'allocation des ressources médicales. En particulier, un système de diagnostic préliminaire automatisé, intégré à un réseau de spécialistes accessible à distance, pourrait aider à réduire la charge de travail des médecins généralistes et à fournir des soins plus ciblés.

0.3. Questions de recherche

Face aux difficultés d'accès aux soins spécialisés dans la ville de Goma, une question fondamentale se pose :

Comment tirer parti des technologies numériques pour améliorer l'accès, la qualité et l'efficacité des consultations médicales, tout en assurant la confidentialité et la sécurité des données des patients ?

Cette question principale soulève plusieurs interrogations complémentaires :

- Quelles fonctionnalités essentielles devraient être incluses pour répondre aux besoins des utilisateurs (patients et médecins) ?
- Dans quelle mesure une plateforme numérique de consultation médicale permet-elle de réduire les délais de prise en charge, d'accroître la satisfaction des utilisateurs et d'améliorer les résultats de santé ?
- Quels mécanismes mettre en place pour garantir la confidentialité, l'authenticité et la sécurité des échanges entre patients et professionnels de santé ?

0.4. Formulation des hypothèses

En réponse aux questions soulevées précédemment, nous formulons les hypothèses suivantes, qui devront être testées et validées au cours de ce travail de recherche :

- Les fonctionnalités telles que les chatbots pour les diagnostics préliminaires, la prise de rendez-vous en ligne et les ordonnances numériques sont des éléments clés pour répondre aux besoins spécifiques des utilisateurs, qu'ils soient patients ou professionnels de santé.
- Une plateforme numérique de consultation médicale contribuerait à réduire les délais de prise en charge, à accroître la satisfaction des utilisateurs, et à améliorer les résultats de santé en optimisant l'allocation des ressources médicales et en simplifiant le parcours patient.
- La mise en œuvre de mécanismes de protection tels que les authentifications sécurisées, et la gestion stricte des accès aux données médicales garantirait la confidentialité, l'authenticité, et la sécurité des échanges entre patients et professionnels de santé.

Ces hypothèses orientent notre travail vers la vérification de la faisabilité technique, la pertinence médicale, et l'acceptabilité sociale d'un tel système dans un environnement où l'accès aux soins reste un défi majeur.

0.5. Justification du choix du sujet et motivations

Nous avons décidé de concevoir et de mettre en place un système de consultation médicale en ligne, en observant les défis auxquels sont confrontées les populations de Goma pour accéder aux soins de santé spécialisés. Le manque de médecins qualifiés et l'encombrement des structures médicales locales sont des problèmes récurrents, créant des retards dans les diagnostics et les soins, en particulier pour les cas nécessitant une attention spécialisée. Ce projet vise à fournir une solution technologique pour faciliter l'accès rapide à des diagnostics de base et à des médecins spécialistes, contribuant ainsi à une meilleure gestion des flux de patients et à une amélioration de la qualité des soins.

Cette thématique s'inscrit dans le cadre des préoccupations contemporaines concernant l'utilisation des technologies numériques pour améliorer l'efficacité des systèmes de santé, en particulier dans les régions à ressources limitées. En intégrant un chatbot pour le diagnostic préliminaire et la recommandation de spécialistes, ce projet représente une approche novatrice pour désengorger les structures médicales locales. De plus, il répond aux préoccupations des décideurs de la santé publique en proposant une solution accessible et évolutive, susceptible de réduire les délais de prise en charge des patients, tout en optimisant l'allocation des ressources médicales.

0.6. Énoncé des objectifs de recherche

0.6.1. L'objectif général

L'objectif principal de cette recherche est de concevoir et de mettre en œuvre un système de consultation médicale en ligne qui facilite l'accès aux soins spécialisés pour les patients de la ville de Goma. Ce système intégrera un chatbot capable de fournir des diagnostics préliminaires et de recommander des médecins en fonction des symptômes rapportés par les utilisateurs, avec l'intention d'améliorer la rapidité et la qualité des soins.

0.6.2. Les objectifs opérationnels/spécifiques

Pour atteindre cet objectif général, plusieurs objectifs spécifiques seront poursuivis :

- Identifier les besoins des patients en matière de consultation médicale en ligne afin de concevoir des fonctionnalités adaptées à leurs attentes.
- Étudier les meilleures pratiques en matière de développement de chatbots pour le diagnostic médical, en se basant sur des recherches antérieures et des systèmes existants.
- Développer un prototype de chatbot capable d'analyser les symptômes et de fournir des recommandations précises concernant les médecins spécialistes.
- Analyser les données recueillies afin d'optimiser le fonctionnement du système et d'améliorer l'expérience utilisateur dans le processus de consultation.

Ces objectifs opérationnels permettront de structurer la recherche et de garantir l'atteinte de l'objectif général.

0.7. Méthodologie et délimitation du travail

Dans le cadre de la conception de notre système de consultation médicale en ligne, nous avons prévu d'utiliser des méthodes analytiques en complément de techniques d'observation en situation et d'entretiens de recherche. Ces méthodes visent à vérifier si l'application en développement répond efficacement aux défis rencontrés dans l'accès aux soins spécialisés à Goma. L'analyse documentaire s'avère également cruciale pour obtenir une vision approfondie des techniques existantes ainsi que des améliorations potentielles pour mieux adapter la solution aux besoins locaux.

Cette étude s'inscrit dans le cadre de l'année académique en cours et vise à développer une plateforme numérique de consultation médicale en ligne, répondant aux besoins spécifiques de la ville de Goma. Elle prend en compte les défis majeurs liés à l'accès aux soins, notamment la surcharge des centres de santé et le manque de spécialistes. Le projet repose sur l'intégration d'un chatbot pour le diagnostic préliminaire, un système de recommandation de médecins adaptés aux symptômes des patients, et un module de prise de rendez-vous en ligne. L'objectif est de désengorger les structures médicales, d'améliorer l'accès aux soins de qualité et de rationaliser la gestion des flux de patients. En s'appuyant sur ces innovations numériques, cette initiative cherche également à transformer l'expérience des soins pour les patients tout en répondant aux spécificités locales.

0.8. Structure du mémoire/ Subdivision du travail

En dehors de l'introduction et de la conclusion générale, ce travail se structure comme suit :

1. Chapitre 1 : Fondements théoriques et contexte de la télémédecine à Goma
Ce chapitre couvre les concepts et fondements théoriques nécessaires pour comprendre les défis de la télémédecine dans des régions sous-desservies, en particulier les

technologies numériques pour les consultations médicales en ligne et les systèmes de diagnostic automatisés.

2. Chapitre 2 : Analyse et Conception du Système

Dans ce chapitre, nous détaillons l'analyse des besoins ainsi que la conception et la modélisation de notre système de consultation médicale en ligne. Cette section aborde également les méthodologies adoptées pour structurer le développement de l'application.

3. Chapitre 3 : Réalisation de l'Application Web

Ce dernier chapitre se concentre sur la mise en œuvre pratique de notre plateforme, correspondant à la phase finale de développement. Nous y décrivons les étapes de réalisation, les outils utilisés et les résultats obtenus à chaque phase du processus de création de l'application.

Ces sections forment un ensemble cohérent visant à documenter l'ensemble du processus, depuis les fondements théoriques jusqu'à la concrétisation de la solution.

Chapitre 1

Fondements théoriques et contexte de la télémédecine à Goma

1.1 Introduction partielle

Dans ce chapitre, nous posons les bases théoriques et contextuelles nécessaires pour comprendre les enjeux et les opportunités de la télémédecine. Nous abordons d'abord le contexte général de la santé à Goma et les défis rencontrés pour l'accès aux soins spécialisés. Ensuite, nous explorons les concepts clés de la télémédecine et de la consultation en ligne, en mettant l'accent sur le rôle des chatbots et de l'intelligence artificielle dans le diagnostic médical. Nous terminerons par une analyse des technologies utilisées pour le développement de notre plateforme.

1.2. Etat de l'art

Dans des régions comme la ville de Goma, où l'accès aux soins de santé spécialisés reste un défi majeur, la télémédecine se présente comme une solution innovante et nécessaire. Les centres de santé de la région sont souvent surchargés, et le manque de médecins spécialistes entraînent des retards dans la prise en charge des patients, particulièrement pour les diagnostics et soins nécessitant une expertise spécifique. Face à ces obstacles, le développement de technologies de consultation médicale en ligne pourrait non seulement désengorger les structures de santé, mais aussi améliorer l'accès aux soins pour les patients qui en ont le plus besoin.

Ce projet de mémoire vise à concevoir un système de consultation médicale en ligne intégrant un chatbot pour le diagnostic préliminaire et l'orientation des patients vers des spécialistes adaptés. Ce système, pensé pour le contexte de Goma, se propose de répondre aux besoins de la population en fournissant des diagnostics rapides et en optimisant la gestion des flux de patients dans les centres de santé. Une telle solution pourrait avoir un impact significatif en facilitant l'accès aux soins spécialisés, en particulier les temps d'attente, et en maximisant les ressources médicales disponibles.

1.3. Pratiques traditionnelles d'accès aux soins médicaux

Dans la ville de Goma, l'accès aux soins de santé spécialisés repose traditionnellement sur des consultations physiques dans les établissements de santé locaux, qu'ils soient publics ou privés. Ce modèle implique plusieurs étapes que les patients doivent suivre pour obtenir un diagnostic et des soins spécialisés.

1.3.1. Consultation de Premier niveau :

La plupart des patients commencent par consulter un centre de santé local, souvent surchargé et limité en ressources. Ces établissements de soins primaires disposent principalement de médecins généralistes, capables de diagnostiquer et de traiter les Infections courantes, mais non spécialisés pour les cas complexes. Si le médecin soupçonne une condition qui requiert une expertise plus poussée, il oriente le patient vers un hôpital ou un spécialiste. Cette étape initiale crée souvent un premier niveau de délai dans le processus de prise en charge.

1.3.2. Référencement vers des Spécialistes :

Une fois le besoin de soins spécialisés identifié, les patients sont dirigés vers des centres hospitaliers disposant de médecins spécialistes. Cependant, à Goma, le nombre de spécialistes est restreint, ce qui entraîne des temps d'attente importants avant de pouvoir accéder à une consultation spécialisée. Le manque de coordination entre les différents niveaux de soins (centre de santé, hôpital général, spécialiste) allonge le parcours du patient, augmentant les risques de complications en raison des retards.

1.3.3. Gestion Manuelle des Rendez-vous et des Dossiers Médicaux:

Actuellement, la gestion des rendez-vous et des dossiers médicaux se fait souvent de manière manuelle dans les établissements de santé de Goma. Les patients doivent physiquement se déplacer pour prendre un rendez-vous, et il est courant qu'ils apportent leurs dossiers médicaux sous forme de documents papier, risquant ainsi la perte ou la dégradation d'informations cruciales pour le suivi de leur état de santé.

Déplacements Fréquents et Coûts Associés :

Pour chaque étape de ce processus, les patients doivent se déplacer, ce qui entraîne des coûts de transport et un temps de déplacement conséquent. Les patients résidant dans les zones rurales ou en périphérie de la ville sont particulièrement touchés, car les infrastructures de transport peuvent être limitées ou peu fiables. Cela rend les consultations de suivi, souvent nécessaires dans les cas chroniques ou complexes, particulièrement difficiles à réaliser.

1.4. Défis d'accès aux soins spécialisés à Goma

Dans la ville de Goma, l'accès aux soins de santé spécialisés est entravé par plusieurs facteurs, parmi lesquels le manque de spécialistes, la surcharge des centres de santé et des infrastructures de transport limitées.

- **Manque de spécialistes** : La ville de Goma dispose d'un nombre insuffisant de médecins spécialisés dans des domaines essentiels, ce qui rend l'accès aux soins spécialisés difficiles pour les patients. Cette pénurie de spécialistes entraîne des délais d'attente prolongés, pouvant aggraver les conditions de santé des patients en retardant leur prise en charge. Les consultations médicales nécessitant une expertise spécifique sont souvent limitées ou signalées, ce qui crée une insatisfaction parmi les patients et diminue la qualité des soins.
- **Surcharge des centres de santé** : Les établissements de santé de Goma font face à une forte demande de services médicaux, ce qui provoque une surcharge importante dans les centres de soins et les hôpitaux. Ce supplément allonge encore les délais pour obtenir un rendez-vous, en particulier pour les consultations spécialisées. Par conséquent, le personnel médical est souvent submergé, ce qui réduit l'efficacité et la rapidité des soins prodigués.
- **Difficultés de transport et d'accessibilité géographique** : En dehors de la pénurie de spécialistes et de la surcharge des centres de santé, les infrastructures de transport jouent également un rôle dans l'accès limité aux soins de santé. Les patients résidant en périphérie de Goma ou dans des zones rurales rencontrent des obstacles géographiques importants

pour se rendre dans les centres de santé. Cela complique l'accès aux consultations, en particulier pour les cas nécessitant un suivi régulier ou une prise en charge urgente.

1.5. Besoin d'une solution numérique pour améliorer l'accès aux soins

Face aux défis rencontrés à Goma en matière d'accès aux soins spécialisés, la mise en œuvre d'une solution numérique telle que la télémédecine se présente comme une réponse pertinente et efficace. En exploitant les outils numériques, la télémédecine peut non seulement désengorger les structures de santé en réalisant une évaluation initiale à distance, mais aussi prioriser les patients nécessitant une prise en charge urgente. Ce dispositif offre également une alternative précieuse pour les personnes confrontées à des difficultés de déplacement, leur permettant d'accéder à des consultations sans contrainte géographique. Plus spécifiquement, l'introduction d'un système de consultation médicale en ligne intégrant un chatbot de diagnostic préliminaire pourrait transformer l'accès aux soins pour la population locale. En offrant un diagnostic initial et en orientant les patients vers les spécialistes adéquats, cette solution contribuerait à rationaliser l'utilisation des ressources médicales tout en réduisant les délais de prise en charge. Ainsi, un tel système représente une opportunité majeure pour améliorer l'efficacité des soins et répondre aux besoins spécifiques des habitants de Goma.

1.5.1. Concepts de base de la télémédecine et des systèmes de consultation en ligne

Cette section vise à introduire les concepts fondamentaux de la télémédecine et de la consultation en ligne.

1.5.1.1. Définition et types de télémédecine [2]

La télémédecine est une pratique médicale à distance mobilisant des technologies de l'information et de la communication. Elle permet de répondre aux difficultés démographiques, épidémiologiques et organisationnelles. La télémédecine met en rapport entre eux, soit le patient et un ou plusieurs professionnels de santé ; soit plusieurs professionnels de santé.

Elle permet d'établir un diagnostic, d'assurer, pour un patient à risque, un suivi dans le cadre de la prévention ou un suivi post-thérapeutique, de requérir un avis spécialisé, de préparer une décision thérapeutique, de prescrire des produits, de prescrire ou de réaliser des prestations ou des actes, ainsi que d'effectuer une surveillance de l'état des patients.

Selon l'Organisation mondiale de la santé (OMS) [3], les types de télémédecine comprennent :

1. Téléconsultation : Consultation médicale à distance entre un patient et un professionnel de santé.
2. Télésurveillance : Surveillance de la santé d'un patient à distance, souvent pour des maladies chroniques.
3. Téléassistance : Assistance médicale entre professionnels de santé, par exemple pour une chirurgie assistée à distance.
4. Télé-expertise : Consultation entre professionnels de santé pour obtenir l'avis d'un spécialiste.
5. La régulation médicale : La régulation médicale réalisée par les SAMU-centres 15 permet d'apporter à distance une réponse médicale et d'orienter le patient dans son parcours de soins.

Ces différentes formes de télémédecine permettent de pallier les difficultés d'accès aux soins dans les zones isolées en réduisant les déplacements et les délais d'attente pour des soins spécialisés.

1.5.1.2. Défis et Limites des Systèmes de Consultation en Ligne

Malgré leurs avantages, les systèmes de consultation en ligne présentent certains défis :

- Précision du diagnostic automatisé : Les chatbots et les systèmes d'IA peuvent commettre des erreurs, en particulier pour les cas médicaux complexes. Par conséquent, il est essentiel que les diagnostics automatiques soient validés par un professionnel de santé. [4]

- Confidentialité et sécurité des données : La consultation en ligne nécessite le traitement de données médicales sensibles. La confidentialité et la sécurité des informations patient doivent donc être strictement garanties.
- Adoption par les utilisateurs : Dans certaines régions, les barrières culturelles et sociales ou technologiques peuvent freiner l'adoption de la télémédecine. Il est donc crucial de sensibiliser les patients et les professionnels de santé pour garantir l'efficacité de ces solutions. [5]

1.6. Chatbot et Intelligence Artificielle dans le Diagnostic Médical

Dans cette section, nous examinerons les principales composantes de l'IA dans les chatbots médicaux, notamment le traitement automatique du langage naturel (NLP), les modèles de langage larges (LLM), et les modèles de classification. Nous explorerons également l'intégration de Llama, un modèle de langage avancé, pour fournir un diagnostic adapté aux besoins de ce projet.

a) L'intelligence artificielle (IA) :

L'intelligence artificielle (IA) est une technologie qui permet aux ordinateurs et aux machines de simuler l'apprentissage, la compréhension, la résolution de problèmes, la prise de décision, la créativité et l'autonomie de l'être humain.

Les applications et les appareils équipés d'une IA peuvent voir et identifier les objets. Ils sont capables de comprendre le langage humain et d'y répondre. Ils peuvent tirer des enseignements de nouvelles informations et de nouvelles expériences. Ils peuvent formuler des recommandations détaillées aux utilisateurs et aux experts. Ils peuvent agir indépendamment, remplaçant ainsi l'intelligence ou l'intervention humaine [6].

b) Machine Learning

Le machine learning est la science du développement d'algorithmes et de modèles statistiques que les systèmes informatiques utilisent pour effectuer des tâches sans instructions explicites, en s'appuyant sur des modèles et des déductions. Les systèmes informatiques utilisent des algorithmes

de machine learning pour traiter de grandes quantités de données et identifier des modèles de données. Cela leur permet de prédire les résultats avec davantage de précision à partir d'un jeu de données d'entrée donné. Par exemple, les scientifiques des données pourraient entraîner une application médicale en vue de diagnostiquer le cancer à partir d'images radiographiques en stockant des millions d'images numérisées et les diagnostics correspondants [7].

c) Deep Learning

Le deep learning est un type de technique de machine learning qui s'inspire du cerveau humain. En effet, les algorithmes de deep learning analysent les données avec une structure logique similaire à celle utilisée par les humains. Le deep learning utilise des systèmes intelligents appelés réseaux neuronaux artificiels pour traiter les informations par couches. Ainsi, les données circulent de la couche d'entrée à travers plusieurs couches de réseaux neuronaux cachés « profonds » avant d'arriver à la couche de sortie. Les couches cachées supplémentaires permettent un apprentissage bien plus performant que celui des modèles de machine learning standard [7].

d) Chatbot :

Un chatbot est un programme informatique qui simule une conversation humaine avec un utilisateur final. Tous les chatbots ne s'appuient pas sur l'intelligence artificielle (IA), mais les chatbots modernes sont de plus en plus nombreux à utiliser des techniques d'IA conversationnelle telles que le traitement automatique du langage naturel (NLP) pour comprendre les questions des utilisateurs et automatiser les réponses qu'ils leur donnent [8].

L'intelligence artificielle (IA) joue un rôle clé dans l'amélioration des systèmes de consultation en ligne, notamment par le biais des chatbots médicaux. Ces chatbots utilisent des algorithmes d'apprentissage automatique pour fournir un diagnostic préliminaire basé sur les symptômes décrits par le patient. En analysant ces symptômes, le chatbot peut :

- **Évaluer l'urgence :** En fonction des informations fournies, le chatbot peut déterminer si le patient a besoin d'une prise en charge immédiate ou si une consultation différée est suffisante.

- Orienter vers les spécialistes : Le chatbot peut recommander des médecins spécialistes en fonction des symptômes analysés, ce qui aide à diriger les patients vers le professionnel le plus adapté à leur situation.

L'intelligence artificielle permet aux chatbots médicaux de comprendre, analyser et interpréter les symptômes décrits par les patients, même lorsque ceux-ci utilisent un langage non spécialisé. Les principales techniques d'IA appliquées dans les chatbots médicaux incluent le NLP, les LLM, et les modèles de classification pour le diagnostic.

Des plateformes telles que LangChain offrent des outils pour intégrer des capacités de diagnostic dans les systèmes de télémédecine, permettant ainsi aux chatbots de tirer parti de bases de données médicales pour améliorer la précision des recommandations. Cependant, il est important de noter que ces systèmes nécessitent une validation médicale, car l'IA peut parfois présenter des limites dans l'interprétation des symptômes complexes.

e) Traitement Automatique du Langage Naturel (NLP)

Le NLP pour Natural Language Processing ou Traitement du Langage Naturel est une discipline qui porte essentiellement sur la compréhension, la manipulation et la génération du langage naturel par les machines. Ainsi, le NLP est réellement à l'interface entre la science informatique et la linguistique (car il permet aux machines de comprendre, traiter et générer du langage humain). Il porte donc sur la capacité de la machine à interagir directement avec l'humain [9].

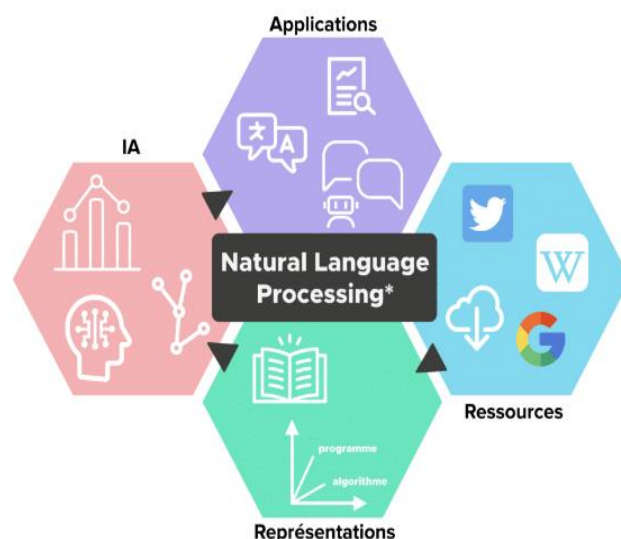


Figure 1-1 : Domaines d'applications du NLP

Dans le contexte des chatbots médicaux, le NLP joue un rôle essentiel en permettant au chatbot de poser des questions ciblées et d'interpréter les descriptions de symptômes en langage naturel. En analysant le texte fourni par les patients, le NLP permet une interaction plus fluide et une collecte d'informations précises, ce qui améliore la qualité du diagnostic préliminaire

f) Modèles de Langage Larges (LLM)

Un grand modèle linguistique (Large Language Model, LLM) est un type de programme d'intelligence artificielle (IA) capable, entre autres tâches, de reconnaître et de générer du texte. Les LLM sont entraînés sur de vastes ensembles de données, d'où l'emploi du terme « large » (grand) dans la dénomination anglaise. Ils s'appuient sur l'apprentissage automatique (Machine Learning, ML) et plus spécifiquement sur un type de réseau neuronal dénommé « modèle transformateur » [10].

En termes plus simples, un LLM est un programme informatique nourri de suffisamment d'exemples pour être capable de reconnaître et d'interpréter le langage humain ou d'autres types de données complexes. De nombreux LLM sont entraînés à l'aide de données recueillies sur Internet,

soit des milliers ou des millions de gigaoctet de texte. Toutefois, la qualité des échantillons a une incidence sur l'apprentissage du langage naturel par les LLM. Les programmeurs d'un LLM auront donc tendance à utiliser un ensemble de données mieux structuré [10].

Les LLM se déclinent en plusieurs types, chacun étant adapté à des tâches spécifiques ou offrant des caractéristiques distinctes :

a) Modèles Autoregressifs [11]

Les modèles autorégressifs sont une classe de modèles de machine learning (ML) qui prédisent automatiquement le composant suivant d'une séquence en prenant des mesures à partir des entrées précédentes de la séquence. On peut citer la génération de texte, la complétion de texte, la réponse aux questions, etc.

Exemples : GPT (Generative Pre-trained Transformer) de OpenAI, LLaMA de Meta, Gopher de DeepMind.

b) Modèles Autoencoders [12]

Les auto encodeurs sont des réseaux de neurones un peu particuliers qui possèdent exactement le **même nombre de neurones sur leur couche d'entrée et leur couche de sortie**. Le but pour un auto encodeur est d'avoir une sortie la plus proche de l'entrée. L'apprentissage est donc « auto-supervisé » car l'erreur à minimiser correspond à la différence entre la sortie générée et l'entrée initiale. Les données n'ont donc pas à être étiquetées, car elles sont leurs propres étiquettes, ce qui fait donc de ce modèle un modèle non supervisé. Ce modèle est utilisé dans la classification de texte, l'analyse de sentiment, l'extraction d'entités nommées, etc.

Exemples : BERT (Bidirectional Encoder Representations from Transformers) de Google, RoBERTa de Facebook, DistilBERT.

c) Modèles Seq2Seq (Sequence-to-Sequence)

Les réseaux Seq2Seq sont des modèles d'apprentissage automatique puissants qui transforment une séquence d'entrée en une séquence de sortie, même lorsque les longueurs de ces séquences

sont différentes [13]. Ce modèle est utilisé dans la traduction automatique, le résumé de texte, la génération de réponses, etc.

Exemples : T5 (Text-To-Text Transfer Transformer), BART (Bidirectional and Auto-Regressive Transformers).

d) Modèles de Type Diffusion ou Multimodaux [14]

Le Multimodal Learning est une évolution du Machine Learning, consistant à utiliser simultanément plusieurs sources de données comme le texte, l'image et l'audio pour résoudre des tâches beaucoup plus complexes. Ce modèle est utilisé dans la génération d'images à partir de texte, l'analyse d'images avec contexte textuel, etc.

Exemples : DALL-E (qui crée des images à partir de prompts textuels), GPT-4 Multimodal (intègre à la fois le texte, la parole et l'image pour comprendre et répondre aux requêtes de manière contextuelle et personnalisée).

e) Modèles Basés sur des Graphes [15]

Les graphes sont une structure de données omniprésente et un langage universel pour décrire des systèmes complexes. Dans la vision la plus générale, un graphe est simplement une collection d'objets (c'est-à-dire des nœuds), ainsi qu'un ensemble d'interactions (c'est-à-dire des arêtes) entre des paires de ces objets.

Par exemple, pour coder un réseau social sous forme de graphe, nous pourrions utiliser des nœuds pour représenter des individus et utiliser des arêtes pour représenter que deux individus sont amis. Dans le domaine biologique, nous pourrions utiliser les nœuds d'un graphe pour représenter des protéines et utiliser les arêtes pour représenter diverses interactions biologiques, telles que les interactions cinétiques entre protéines.

f) Modèles Pré-entraînés et Ajustés par RLHF (Rétroaction Humaine)

L'apprentissage par renforcement à partir du feedback humain (RLHF) est une technique de machine learning (ML) qui utilise le feedback humain pour optimiser les modèles ML afin

d'améliorer l'efficacité de l'autoapprentissage. Les techniques d'apprentissage par renforcement (RL) entraînent les logiciels à prendre des décisions qui maximisent les récompenses et rendent leurs résultats plus précis [16].

Exemples : InstructGPT, GPT-4 (ajusté par rétroaction humaine).

Les modèles de langage de grande envergure (LLM) sont des outils avancés en intelligence artificielle, capables de comprendre des dialogues complexes et de traiter des descriptions nuancées de symptômes. En exploitant le traitement du langage naturel (NLP), ils interprètent le contexte, comportent des questions de suivi et génèrent des réponses adaptées pour affiner les diagnostics. Complétant les LLM, les modèles de machine learning comme les réseaux neuronaux et les arbres de décision classifient les symptômes en catégories de maladies probables, identifiant des schémas pour prioriser les cas critiques et orienter les patients nécessaires.

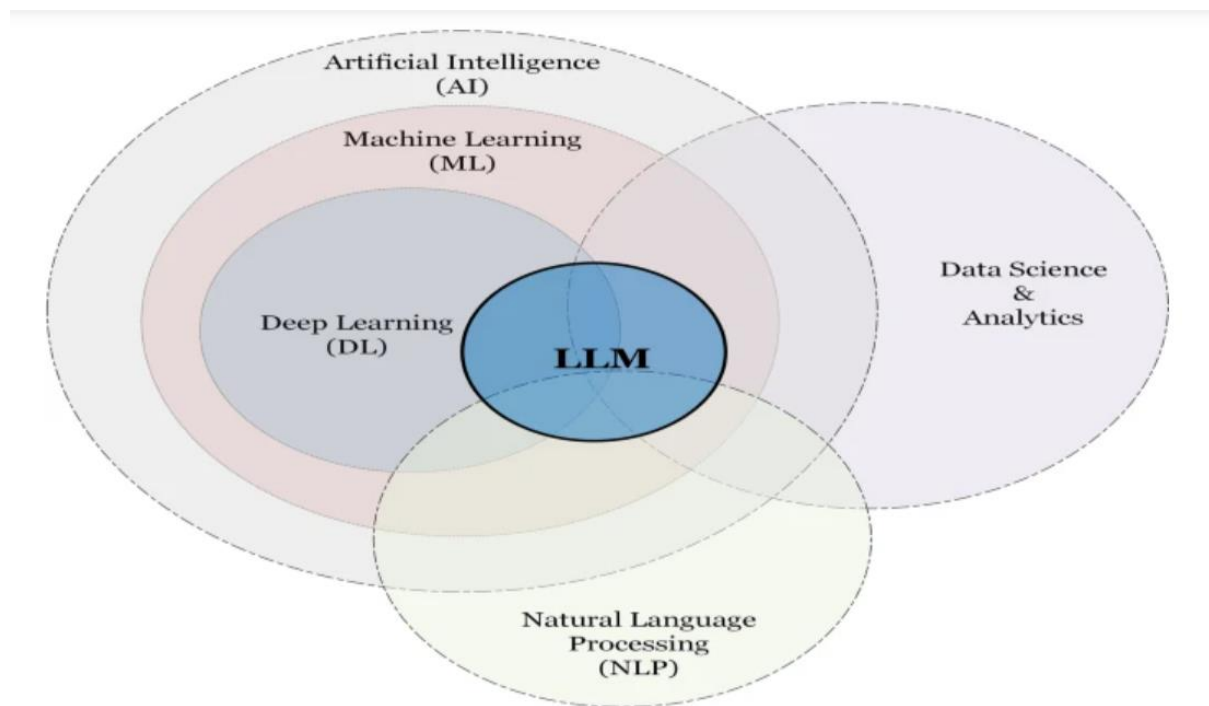


Figure 1-2 : Une illustration soulignant la pertinence et la position du LLM dans le domaine de l'IA, de la science des données et du PNL au sens large

1.7. Utilisation de Llama pour le Diagnostic Médical

Dans le cadre de ce projet, nous avons choisi d'intégrer Llama, un modèle de langage naturel développé par Meta.

1.7.1. Qu'est-ce que Llama ?

LLama, pour Large Language Model Meta AI, est une collection de modèles de langage allant de 7B à 65B paramètres (7 milliards à 65 milliards de paramètres). Les modèles sont entraînés sur des trillions de tokens en utilisant uniquement des ensembles de données accessibles au public [17].

Selon Meta, les modèles Llama s'appuient sur des techniques avancées, notamment l'apprentissage par renforcement avec des retours humains (RLHF), pour garantir des réponses utiles et sûres. Leur entraînement repose sur des données publiques et des sources autorisées, tout en excluant des informations sensibles ou privées. Cette approche permet à Llama d'offrir des résultats performants dans des cas d'utilisation spécifiques comme les chatbots ou la rédaction assistée [18].

1.7.2. Caractéristiques de Llama

Llama présente plusieurs caractéristiques techniques et pratiques qui en font un choix pertinent pour ce projet :

1. **Précision et Fiabilité** : Entraîné sur un large ensemble de données textuelles, Llama se distingue par sa capacité à produire des analyses précises, même face à des descriptions complexes ou variées. Grâce à son architecture avancée, il minimise les erreurs, une qualité essentielle dans des domaines critiques tels que la santé.
2. **Flexibilité et Adaptabilité** : Llama peut être ajusté pour répondre aux besoins spécifiques d'applications telles que le diagnostic médical, grâce à sa capacité d'adaptation à différents contextes linguistiques, une compétence essentielle pour comprendre les descriptions variées et parfois imprécises des symptômes des patients.
3. **Efficacité dans le Traitement Contextuel** : Llama excelle dans l'analyse contextuelle des informations, en identifiant, par exemple, les symptômes rapportés par un patient et en

posant des questions pertinentes pour approfondir l'analyse. Cette capacité contextuelle améliore considérablement l'expérience utilisateur, en la rendant plus fluide et personnalisée.

4. **Facilité d'Intégration :** En tant que modèle avancé, Llama est compatible avec des frameworks comme LangChain, ce qui facilite son intégration dans un système de consultation en ligne tout en permettant d'exploiter des bases de données médicales existantes pour fournir des recommandations précises et adaptées.

1.8. Conclusion Partielle

Ce chapitre a posé les bases théoriques et contextuelles nécessaires pour comprendre l'importance de la télémédecine à Goma. Il a mis en évidence les défis majeurs liés à l'accès aux soins spécialisés, tels que la pénurie de médecins, la surcharge des centres de santé et les obstacles géographiques.

Les limites des approches traditionnelles ont souligné la pertinence de la télémédecine comme solution innovante pour améliorer l'accès aux soins, réduire les délais d'attente et optimiser les ressources médicales. Nous avons également exploré les concepts fondamentaux de la télémédecine, les technologies associées, et le rôle crucial de l'intelligence artificielle et des modèles comme **Llama** dans les diagnostics médicaux.

Ces éléments théoriques constituent une base solide pour le développement d'un système de consultation en ligne, adapté aux besoins spécifiques de Goma et visant à offrir des soins plus rapides et accessibles.

Chapitre 2

Analyse et Conception du Système

2.1. Introduction

Ce chapitre explore les fondements nécessaires à la mise en œuvre d'un système de consultation médicale en ligne adapté au contexte de Goma. Il couvre l'analyse des besoins, les choix méthodologiques et la modélisation du système, tout en mettant en avant les solutions envisagées pour répondre aux défis identifiés.

2.2. Analyse des besoins

Pour concevoir une solution de consultation médicale en ligne qui réponde efficacement aux défis spécifiques de Goma, il est indispensable de commencer par identifier les besoins essentiels des principaux utilisateurs, à savoir les patients et les médecins. Ces besoins se fondent sur la nécessité d'améliorer l'accès aux soins spécialisés, de réduire les délais de prise en charge et de garantir la sécurité des informations médicales. Une analyse approfondie permet ainsi de définir les fonctionnalités qui constitueront le cœur du système.

2.2.1. Besoins fonctionnels

Pour concevoir une plateforme de consultation médicale en ligne à Goma, il est essentiel de définir les fonctionnalités clés permettant d'améliorer l'accès aux soins tout en garantissant une expérience utilisateur de qualité. Les principales fonctionnalités sont les suivantes :

❖ Authentification des utilisateurs

Pour garantir la sécurité des données, chaque utilisateur (patient, médecin, administrateur) doit se connecter avec un identifiant unique (à l'instar de son adresse mail) et un mot de passe, avec une option de récupération de ce dernier en cas d'oubli.

❖ **Diagnostic préliminaire automatisé**

Le diagnostic préliminaire représente une étape cruciale pour réduire les délais d'attente et faciliter l'orientation des patients. Pour y parvenir, le système intègre un chatbot basé sur l'intelligence artificielle, conçu pour recueillir les symptômes des patients de manière interactive. Le chatbot analyse ces informations et génère automatiquement un diagnostic préliminaire, accompagné de suggestions de médecins spécialistes adaptés à la situation. Cette fonctionnalité permet d'alléger la charge des centres médicaux en traitant en première instance les cas non urgents.

❖ **Gestion des rendez-vous**

Les patients peuvent demander un rendez-vous, et les médecins ont la possibilité d'accepter, modifier ou annuler ces rendez-vous. Le système envoie des notifications automatiques aux patients et aux médecins pour confirmer et rappeler les rendez-vous.

❖ **Historiques médicaux**

Les patients ont accès à l'ensemble de leur historique (diagnostics, prescriptions, rendez-vous), tandis que les médecins peuvent consulter les dossiers de leurs patients pour assurer un suivi personnalisé.

❖ **Gestion des utilisateurs**

Le système offre à l'administrateur la possibilité d'ajouter, de modifier ou de supprimer des comptes utilisateurs, assurant ainsi une gestion centralisée et sécurisée des informations. L'administrateur peut mettre à jour les informations des médecins, telles que leurs spécialités, afin de garantir que les recommandations soient toujours pertinentes.

❖ **Superviser les données**

Une fonctionnalité de supervision par l'administrateur permet de générer des rapports détaillés (PDF), pour évaluer et améliorer les performances du système.

❖ **Ordonnances électroniques**

Les médecins ont la possibilité de créer, modifier, partager, supprimer et archiver des ordonnances électroniques via la plateforme. Une fois une ordonnance générée, elle est envoyée au patient sous forme de notification et stockée dans son historique médical pour un accès ultérieur. Ce système facilite le suivi des traitements et assure une meilleure traçabilité des prescriptions.

2.2.2 Besoins non fonctionnels

❖ **Accessibilité**

L'application doit fonctionner sur une large gamme d'appareils, notamment les smartphones, tablettes et ordinateurs, et son interface doit être conçue de manière intuitive afin de faciliter son utilisation par des utilisateurs novices en technologie.

❖ **Performance**

Le système doit assurer des temps de réponse rapides lors des opérations d'authentification et de déconnexion, garantissant ainsi une expérience fluide pour l'utilisateur.

❖ **Sécurité et confidentialité**

La protection des données sensibles est primordiale. L'application doit mettre en œuvre des mécanismes robustes pour sécuriser l'accès aux informations, notamment par l'utilisation d'une authentification sécurisée, ainsi que la création de sessions protégées avec un suivi régulier pour prévenir les accès non autorisés. Par ailleurs, une déconnexion automatique après une période d'inactivité doit être instaurée pour assurer une protection continue des données.

2.3. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisations (DCU) sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel [19].

- a. Diagramme de cas d'utilisation de Patient

La figure 2.1 représente les besoins fonctionnels du patient

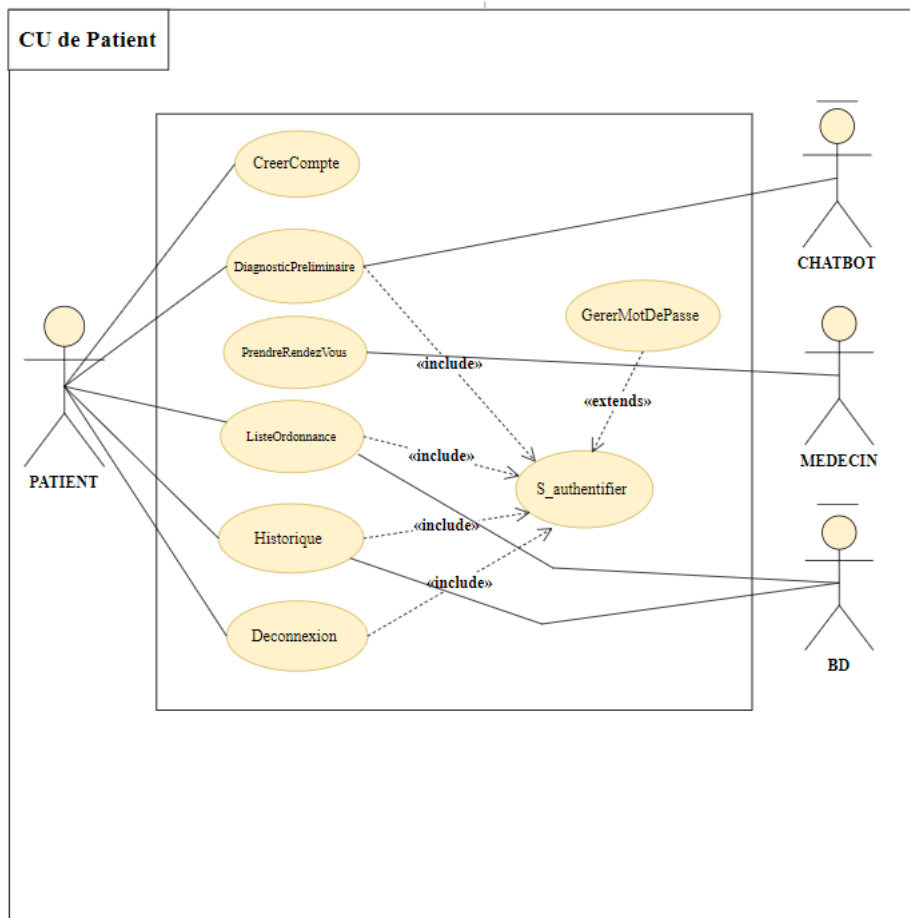


Figure 2-1: Diagramme de cas d'utilisation de patient

b. Diagramme de cas d'utilisation de Médecin

La figure 2.2 représente les besoins fonctionnels du Médecin

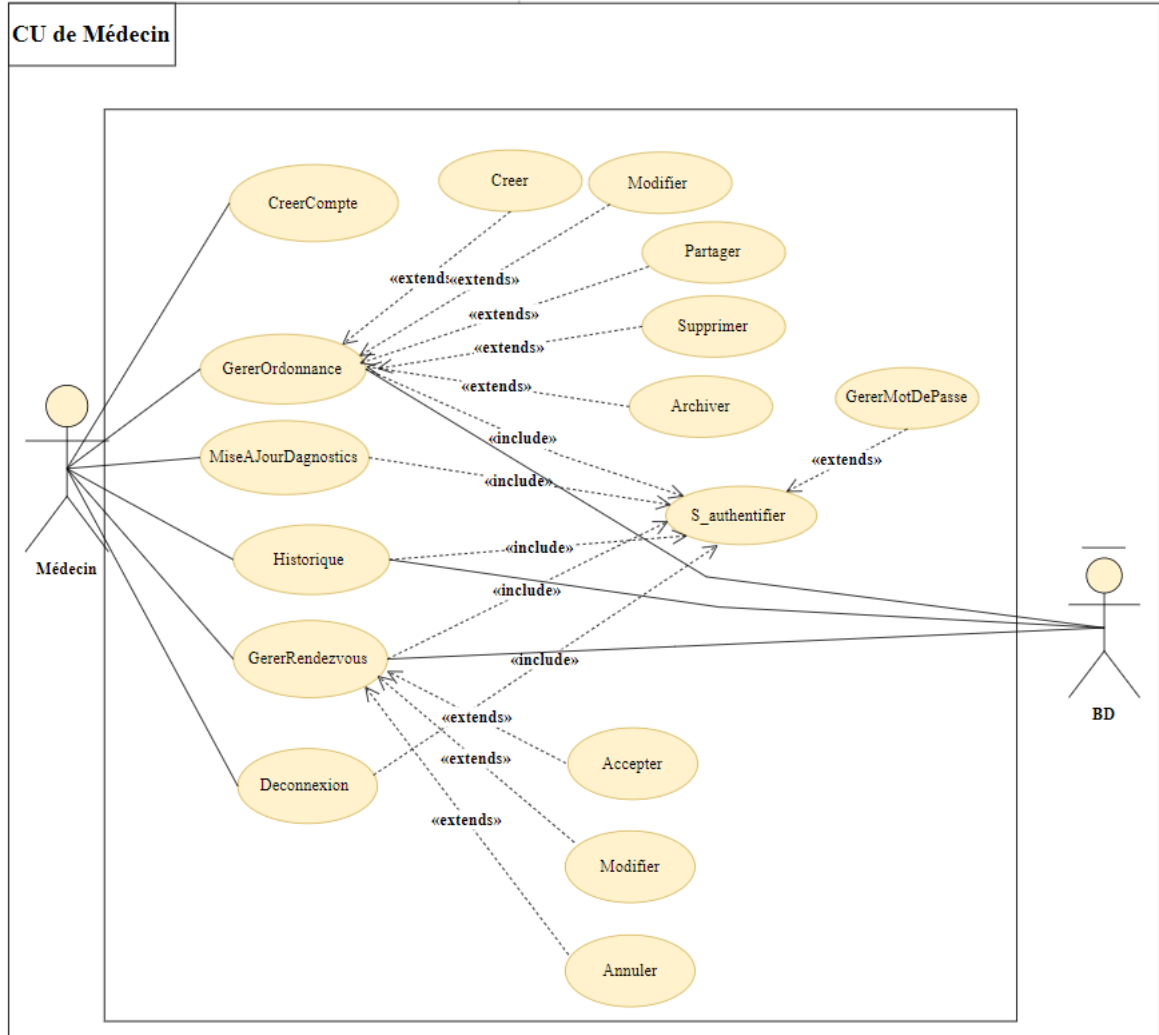


Figure 2-2: Diagramme de cas d'utilisation de Médecin

c. Diagramme de cas d'utilisation d'Administrateur

La figure 2.3 représente les besoins fonctionnels de l'administrateur

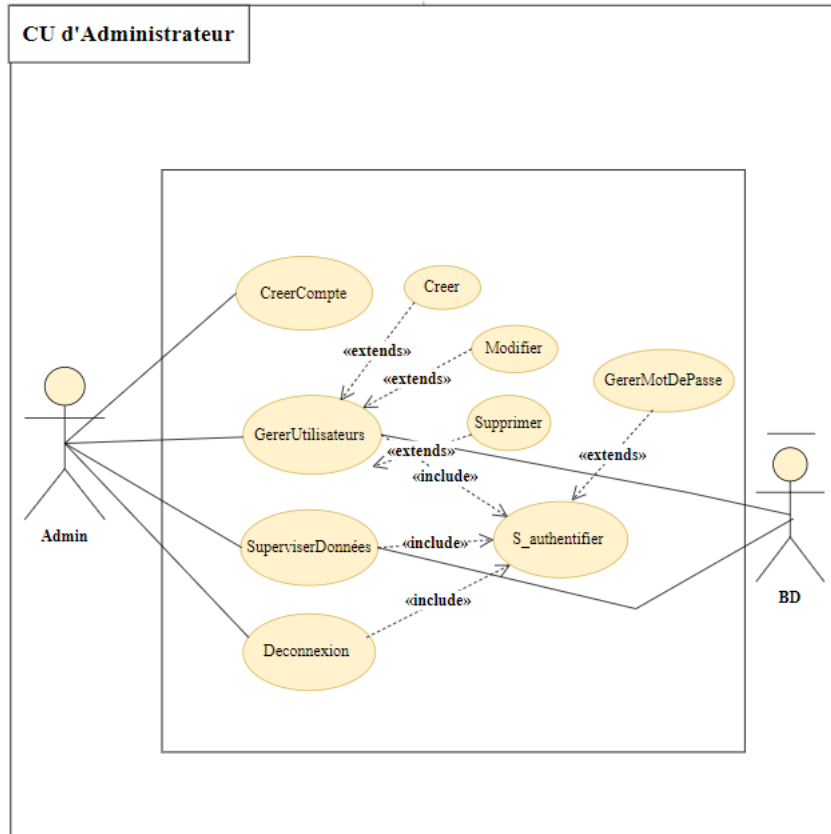


Figure 2-3: Diagramme de cas d'utilisation d'Administrateur

2.4. Documentation de différents cas d'utilisation

Le cas d'utilisation correspond à un ensemble d'actions réalisées par le système en interaction avec les acteurs en vue d'une finalité. L'ensemble des cas d'utilisation permet ainsi de décrire les exigences fonctionnelles d'un système en adoptant le point de vue et le langage de l'utilisateur final [20].

a. Cas d'utilisation « **Créer un compte** »

Tableau 2-1 : Documentation de cas d'utilisation : « Créer un compte »

ID	1
Description	Permet aux nouveaux utilisateurs de s'inscrire sur la plateforme.
Acteur principal	Les internautes cherchant à avoir un compte sur la plateforme.
Acteur secondaire	Aucun
Préconditions	L'utilisateur doit avoir accès à la page d'inscription.
Enchaînement principal	<ol style="list-style-type: none"> 1. Le CU démarre lorsque l'internaute clique sur le bouton s'inscrire. 2. Il remplit le formulaire d'inscription. 3. Le système valide les informations et crée un nouveau compte.
Post condition	Le compte est créé et l'utilisateur peut se connecter.
Alternatif	Si les informations sont invalides, un message d'erreur est affiché.

b. Cas d'utilisation « **Gérer Mot de passe** »

Tableau 2-2: *Documentation de cas d'utilisation : « Gérer mot de passe »*

ID	2
Description	Permet aux utilisateurs de réinitialiser leur mot de passe en cas d'oubli.
Acteur principal	Tous les Utilisateurs
Acteur secondaire	Aucun
Préconditions	L'utilisateur doit avoir déjà créé un compte auparavant
Enchaînement principal	<ol style="list-style-type: none"> 1. Le CU démarre lorsque l'utilisateur clique sur le bouton mot de passe oublié 2. L'utilisateur accède à la section "Mot de passe oublié". 3. Il entre son e-mail pour recevoir un lien de réinitialisation. 4. Le système envoie un e-mail avec les instructions. 5. L'utilisateur clique sur le lien et entre un nouveau mot de passe. 6. Le système valide et met à jour le mot de passe.
Post condition	Le mot de passe est réinitialisé et l'utilisateur peut se connecter avec ses nouvelles informations.
Alternatif	Si l'e-mail est invalide ou non trouvé, un message d'erreur est affiché.

c. Cas d'utilisation « **Diagnostic préliminaire** »

Tableau 2-3: Documentation de cas d'utilisation : « Diagnostic préliminaire »

ID	4
Description	Permet au patient d'obtenir un diagnostic initial basé sur ses symptômes.
Acteur principal	Patient
Acteur secondaire	Chatbot (LLM)
Préconditions	Le Patient doit être connecté.
Enchaînement principal	<ol style="list-style-type: none"> 1. Le CU démarre lorsque le Patient accède à la section " Diagnostic" 2. Le patient lance le chatbot. 3. Le chatbot pose des questions sur les symptômes du patient. 4. Le système analyse les réponses et propose un diagnostic préliminaire. 5. Le système propose un médecin spécialiste en fonction du diagnostic préliminaire établi.
Post condition	Le patient reçoit un diagnostic préliminaire et des recommandations.
Alternatif	Si les réponses sont insuffisantes, le chatbot demande des informations supplémentaires.

d. Cas d'utilisation « **Prendre un rendez-vous** »

Tableau 2-4: Documentation de cas d'utilisation : « Prendre un rendez-vous »

ID	5
Description	Permet au patient de réserver un créneau horaire avec un médecin.
Acteur principal	Patient
Acteur secondaire	Médecin
Préconditions	L'utilisateur doit être connecté et faire un diagnostic
Enchaînement principal	<ol style="list-style-type: none"> 1. Le CU démarre lorsque l'utilisateur clique sur le bouton prendre un rendez-vous 2. Le patient fait un diagnostic et Le système propose un médecin spécialiste en fonction du diagnostic préliminaire établi. 3. Le patient sélectionne un médecin. 4. Le système affiche les créneaux disponibles. 5. Le patient choisit un créneau. 6. Une notification est envoyée au médecin
Post condition	Le rendez-vous est confirmé et ajouté à l'agenda des deux parties.
Alternatif	Si aucun créneau n'est disponible, le patient est informé et peut choisir un autre médecin.

e. Cas d'utilisation « **Consulter l'historique médical (Patient)** »

Tableau 2-5 : Documentation de cas d'utilisation « Consulter l'historique médical (Patient) »

ID	7
Description	Permet au patient de visualiser ses diagnostics, ordonnances et rendez-vous précédents.
Acteur principal	Patient
Acteur secondaire	Aucun
Préconditions	L'utilisateur doit être connecté.
Enchaînement principal	1. Le CU démarre lorsque le patient accède à la section "Historique" depuis son tableau de bord. 2. Le système affiche les données stockées.
Post condition	L'utilisateur visualise les informations demandées.
Alternatif	Si aucune donnée n'est disponible, un message informatif est affiché.

f. Cas d'utilisation « **Mise à jour des diagnostics** »

Tableau 2-7 : Documentation de cas d'utilisation : « Mise à jour des diagnostics »

ID	8
Description	Permet au médecin de valider ou modifier les diagnostics préliminaires.
Acteur primaire	Médecin
Acteur secondaire	Aucun
Préconditions	Le médecin doit être connecté.
Enchaînement principal	1. Le CU démarre lorsque le médecin accède à la section "Diagnostic" 2. Le médecin accède au diagnostic préliminaire d'un patient. 3. Il valide ou apporte des modifications.

	4. Le système enregistre les modifications.
Post condition	Le diagnostic est mis à jour.
Alternatif	Si le médecin ne trouve pas le diagnostic, il peut demander au patient plus d'informations.

g. Cas d'utilisation « **Émettre des ordonnances** »

Tableau 2-8 : Documentation de cas d'utilisation : « **Émettre des ordonnances** »

ID	9
Description	Permet au médecin de créer et partager des ordonnances électroniques.
Acteur primaire	Médecin
Acteur secondaire	Aucun
Préconditions	Le médecin doit être connecté et un diagnostic préliminaire doit avoir été effectué par le chatbot. Avant de générer une ordonnance, le médecin doit valider ou mettre à jour le diagnostic afin de garantir que la prescription repose sur une évaluation médicale confirmée.
Enchaînement principal	1. Le CU démarre lorsque Le médecin accède au dossier d'un patient. 2. Il génère une ordonnance et la partage avec le patient.
Post condition	L'ordonnance est enregistrée et disponible pour le patient.
Alternatif	Si le système rencontre une erreur on affiche un message d'erreur

h. Cas d'utilisation « **Gestion des rendez-vous** »

Tableau 2-9 : Documentation de cas d'utilisation : « Gestion des rendez-vous »

ID	10
Description	Permet au médecin de gérer les rendez-vous programmés avec ses patients.
Acteur primaire	Médecin
Acteur secondaire	Aucun
Préconditions	Le médecin doit être connecté.
Enchaînement principal	<ol style="list-style-type: none"> 1. Le CU démarre lorsque le médecin consulte la liste des rendez-vous. 2. Il peut faire la consultation. 3. Le système envoie une notification au patient.
Post condition	L'ordonnance est enregistrée et disponible pour le patient.
Alternatif	Si le médecin ne peut pas faire la consultation, il peut contacter l'admin.

i. Cas d'utilisation « **Consulter l'historique médical (Médecin)** »

Tableau 2-10 : Documentation de cas d'utilisation : « Consulter l'historique médical (Médecin) »

ID	11
Description	Permet au médecin de consulter l'historique des diagnostics et rendez-vous pour un suivi optimal des patients.
Acteur principal	Médecin
Acteur secondaire	Aucun
Préconditions	Le médecin doit être connecté.

Enchaînement principal	<ol style="list-style-type: none"> 1. Le CU démarre lorsque le médecin accède à la section "Historique" depuis son tableau de bord. 2. Le système affiche les dossiers des patients liés à ce médecin.
Post condition	Le médecin peut consulter les informations nécessaires pour un meilleur suivi des patients.
Alternatif	Si aucune donnée n'est disponible, un message informatif est affiché.

j. Cas d'utilisation « **Gérer les utilisateurs** »

Tableau 2-11 : Documentation de cas d'utilisation : « Gérer les utilisateurs »

ID	12
Description	Permet à l'administrateur de gérer les comptes des utilisateurs (patients et médecins).
Acteur primaire	Administrateur
Acteur secondaire	Aucun
Préconditions	L'administrateur doit être connecté.
Enchaînement principal	<ol style="list-style-type: none"> 1. Le CU démarre lorsque l'administrateur accède à la section "Gestion des utilisateurs". 2. Il consulte la liste des utilisateurs. 3. Il peut ajouter, modifier ou supprimer un utilisateur et pour les médecins modifier le profil. 4. Les modifications sont sauvegardées dans la base de données.
Post condition	Les informations des utilisateurs sont mises à jour dans la base de données.
Alternatif	Si une modification échoue, un message d'erreur est affiché.

k. Cas d'utilisation « **Superviser les données** »

Tableau 2-12 : Documentation de cas d'utilisation : « Superviser les données »

ID	13
Description	Permet à l'administrateur de surveiller les activités sur la plateforme et générer des rapports.
Acteur primaire	Administrateur
Acteur secondaire	Aucun
Préconditions	L'administrateur doit être connecté.
Enchaînement principal	<ol style="list-style-type: none"> 1. Le CU démarre lorsque l'administrateur accède à la section "Supervision". 2. Il choisit les données à analyser (utilisateurs actifs, diagnostics réalisés, rendez-vous pris, ordonnance réalisée). 3. Le système affiche les statistiques correspondantes. 4. L'administrateur génère un rapport en format PDF.
Post condition	Un rapport est généré et disponible pour consultation ou partage
Alternatif	Si aucune donnée n'est disponible, un message informatif est affiché.

2.5. Diagramme de classes

Le diagramme de classes représente la structure statique du système en définissant les différentes classes, leurs attributs, méthodes et les relations entre elles. Il permet de modéliser les interactions entre les utilisateurs (patients, médecins, administrateurs) et les fonctionnalités clés du système, comme la gestion des rendez-vous, l'analyse des symptômes et l'émission d'ordonnances.

2.5.1. Structure des classes

Dans ce point, nous allons détailler comment chaque classe de notre système est structurée, en décrivant ses attributs et méthodes.

1. Utilisateurs

❖ Attributs :

- id : Identifiant unique de l'utilisateur.
- noms : Nom complet de l'utilisateur.
- sexe : Sexe de l'utilisateur.
- email : Adresse e-mail de l'utilisateur.
- adresse : Adresse physique de l'utilisateur.

❖ Méthodes :

- seConnecte() : Permet à l'utilisateur de se connecter au système.
- seDeconnecte() : Permet à l'utilisateur de se déconnecter du système.
- GererMotDePasse() : Permet de réinitialiser le mot de passe en cas d'oubli.

2. Admin (hérite de Utilisateurs)

❖ Méthodes :

- `gererUtilisateurs()` : Permet de gérer les comptes des utilisateurs.
- `genererRapport()` : Génère des rapports sur les activités du système.

3. Patient (hérite de Utilisateurs)

❖ Attributs :

- `age` : Âge du patient.

❖ Méthodes :

- `demanderDiagnostic()` : Permet au patient de soumettre ses symptômes pour analyse.
- `consulterHistorique()` : Permet au patient d'accéder à son historique médical (ordonnances , diagnostics, rendez-vous).

4. Médecin (hérite de Utilisateurs)

❖ Attributs :

- `specialite` : Spécialité du médecin.
- `description` : description précise sur la spécialité du médecin

❖ Méthodes :

- `modifierRendezVous()` : Permet de modifier un rendez-vous existant.
- `consulterHistorique()` : Accède à l'historique médical des patients (ordonnances , diagnostics, rendez-vous).
- `mettreAJourDiagnostic()` : Met à jour le diagnostic du patient.

5. Chatbot

❖ **Attributs :**

- contenuChatbot : Contenu des réponses fournies par le chatbot.

❖ **Méthodes :**

- analyserSymptomes() : Diagnostic : Analyse les symptômes du patient et fournit un diagnostic préliminaire.
- recommanderMedecin() : Médecin : Propose les médecins spécialistes en fonction du diagnostic.

6. Rapport

❖ **Attributs :**

- contenuRapport : Contenu du rapport généré.
- dateGeneration : Date de création du rapport.

❖ **Méthodes :**

- genererRapport() : Génère un rapport d'analyse ou d'activité.

7. Historique

❖ **Attributs :**

- diagnostics : Liste des diagnostics médicaux du patient.
- prescription : Liste des prescriptions médicales.
- rendezvous : Historique des rendez-vous du patient.

❖ **Méthodes :**

- consulterHistorique() : Permet d'accéder aux données médicales passées.

8. RendezVous

❖ Attributs :

- id : Identifiant unique du rendez-vous.
- dateRendezvous : Date et heure du rendez-vous.
- statut : Statut du rendez-vous (confirmé, annulé, en attente).

❖ Méthodes :

- reserver() : Permet de réserver un rendez-vous.
- modifier() : Modifie les détails du rendez-vous.
- annuler() : Annule un rendez-vous.

9. Ordonnance

❖ Attributs :

- id : Identifiant unique de l'ordonnance.
- dateEmission : Date de création de l'ordonnance.
- contenuOrdonnance : Détails de la prescription médicale.

❖ Méthodes :

- genererOrdonnance() : Génère une nouvelle ordonnance pour un patient.

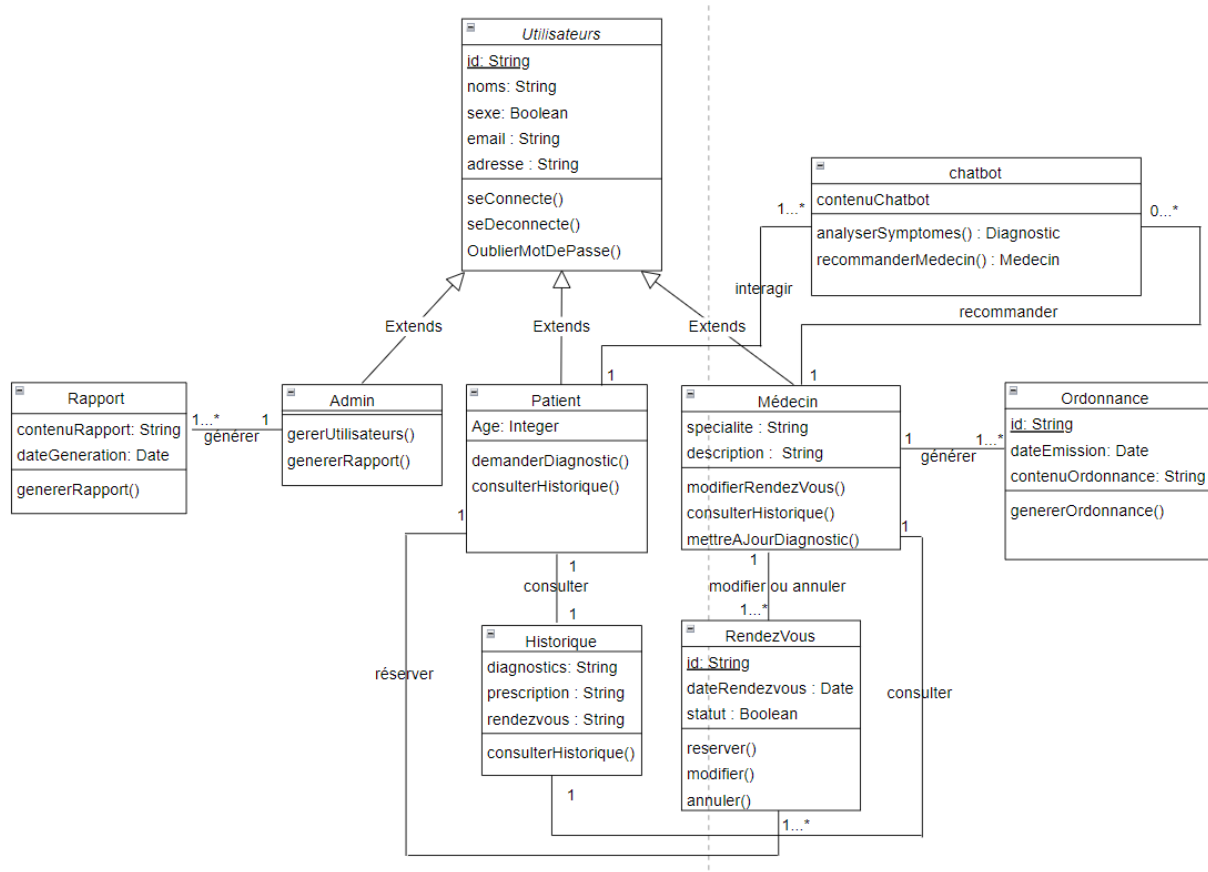


Figure 2-0-4 : Diagramme de classe du système

2.6. Diagramme de séquences

Les diagrammes de séquences illustrent les interactions entre les utilisateurs et le système pour différentes fonctionnalités essentielles.

2.6.1. Authentification des utilisateurs

Pour l'authentification des utilisateurs, le diagramme décrit comment un utilisateur peut créer un compte, se connecter et récupérer son mot de passe. Le système vérifie l'unicité de l'email lors de la création, valide les identifiants lors de la connexion et, en cas de demande, envoie un lien de réinitialisation pour modifier le mot de passe

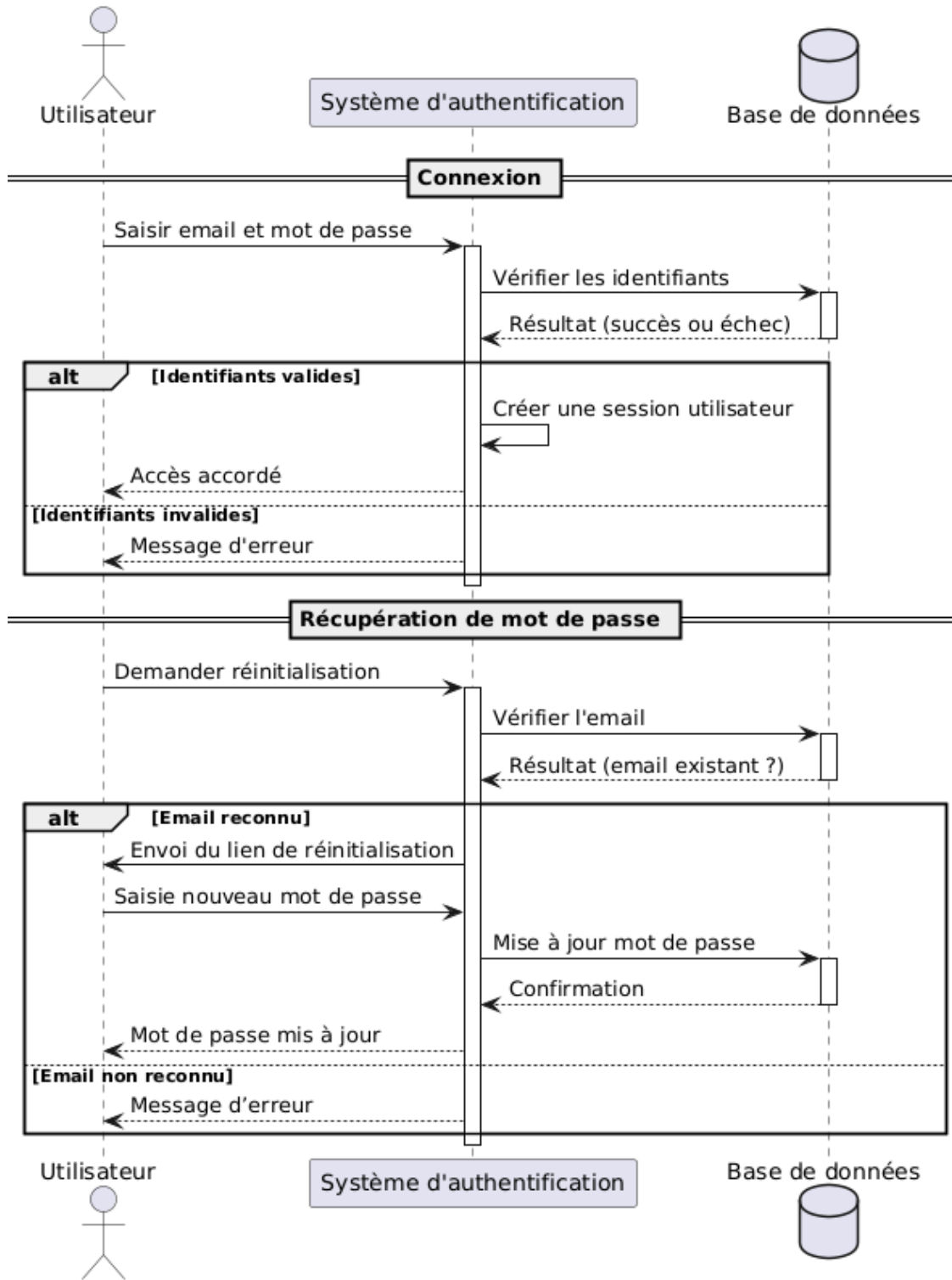


Figure 2-0-5 : Diagramme de séquence d'authentification

2.6.2. Diagnostic préliminaire automatisé

Le diagramme de séquences pour le diagnostic préliminaire automatisé présente le processus par lequel un patient saisit ses symptômes dans un chatbot intelligent. Dès que le patient saisit ses symptômes, le chatbot transmet ces informations à un système d'intelligence artificielle spécialisé dans l'analyse médicale. Ce système analyse les symptômes et renvoie un diagnostic préliminaire au chatbot. Le chatbot affiche alors le diagnostic au patient et lui propose une sélection de médecins spécialisés afin qu'il choisisse un professionnel, permettant ainsi l'envoi automatique de son diagnostic préliminaire à un médecin pour validation, suivi ou prise de rendez-vous.

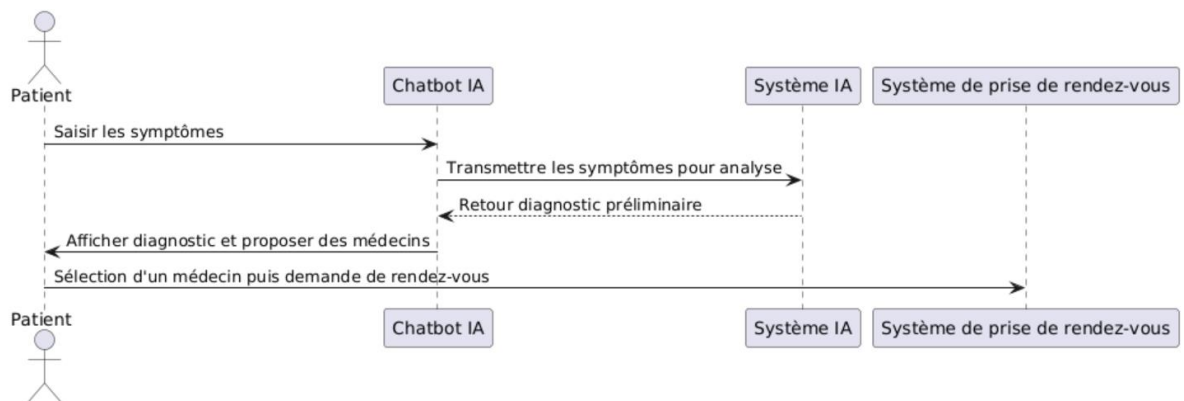


Figure 2-0-6 : Diagramme de séquence de diagnostic préliminaire

2.6.3. Mise à jour du diagnostic par le médecin

Le diagramme concernant la mise à jour du diagnostic par le médecin illustre une séquence où, après qu'un patient ait saisi ses symptômes et obtenu un diagnostic préliminaire via le chatbot, le médecin intervient pour affiner ce diagnostic. Dans un premier temps, le patient saisit ses symptômes et reçoit un diagnostic préliminaire affiché par le chatbot. Le médecin consulte ensuite ce diagnostic via le système et procède à une mise à jour après une analyse plus approfondie. Le système enregistre cette mise à jour dans la base de données et envoie ensuite une notification au patient pour l'informer du diagnostic révisé.

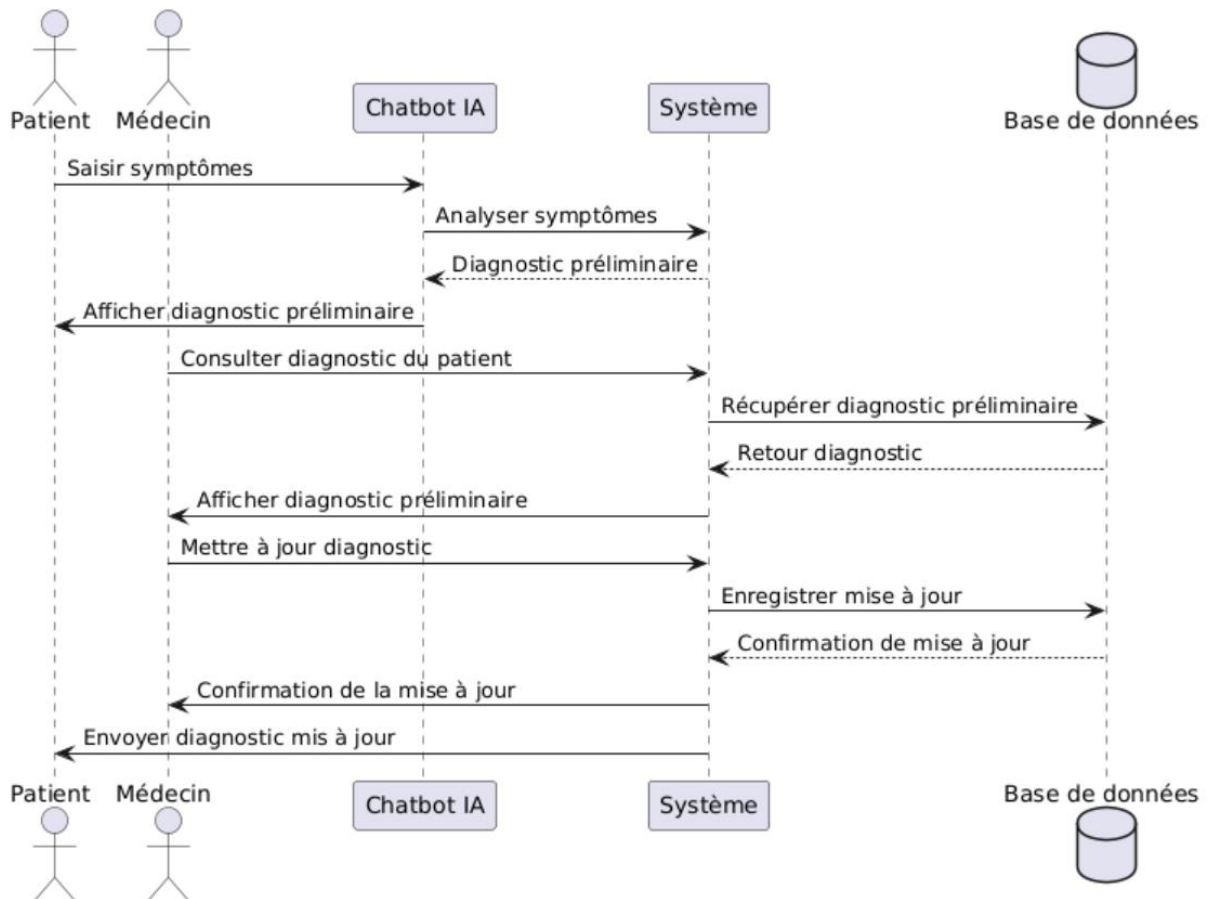


Figure 2-0-7 : Diagramme de séquence de mise à jour de diagnostic

2.6.4. Ordonnance électronique

Le diagramme d'ordonnance électronique décrit la procédure par laquelle le médecin rédige une ordonnance électronique et la transmet au patient. Dans cette séquence, le médecin commence par créer l'ordonnance via le système, qui enregistre cette information dans la base de données. Après validation, le système envoie l'ordonnance au patient pour consultation. Par ailleurs, l'ordonnance est archivée afin de pouvoir être consultée ultérieurement en cas de besoin.

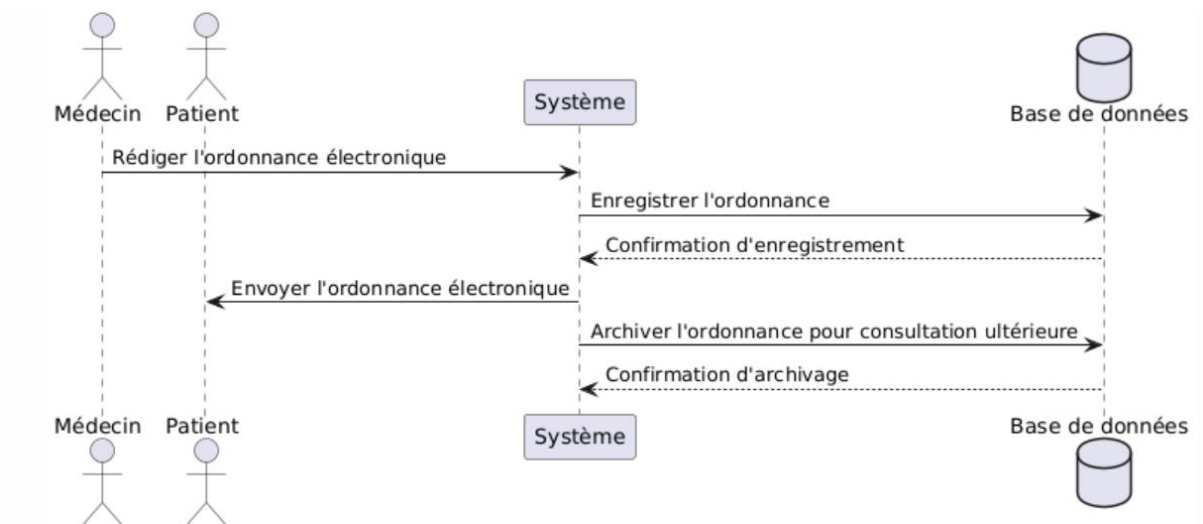


Figure 2-0-8 : Diagramme de séquence d'ordonnance électronique

2.6.5. Supervision et génération de rapports

Le diagramme de supervision et rapports montre comment l'administrateur peut consulter les activités de la plateforme et générer des rapports. Le système interroge la base de données pour récupérer les informations sur les diagnostics et rendez-vous, puis affiche ces données. Lors de la demande de rapport, le système extrait les informations nécessaires, génère le rapport au format PDF ou Excel .

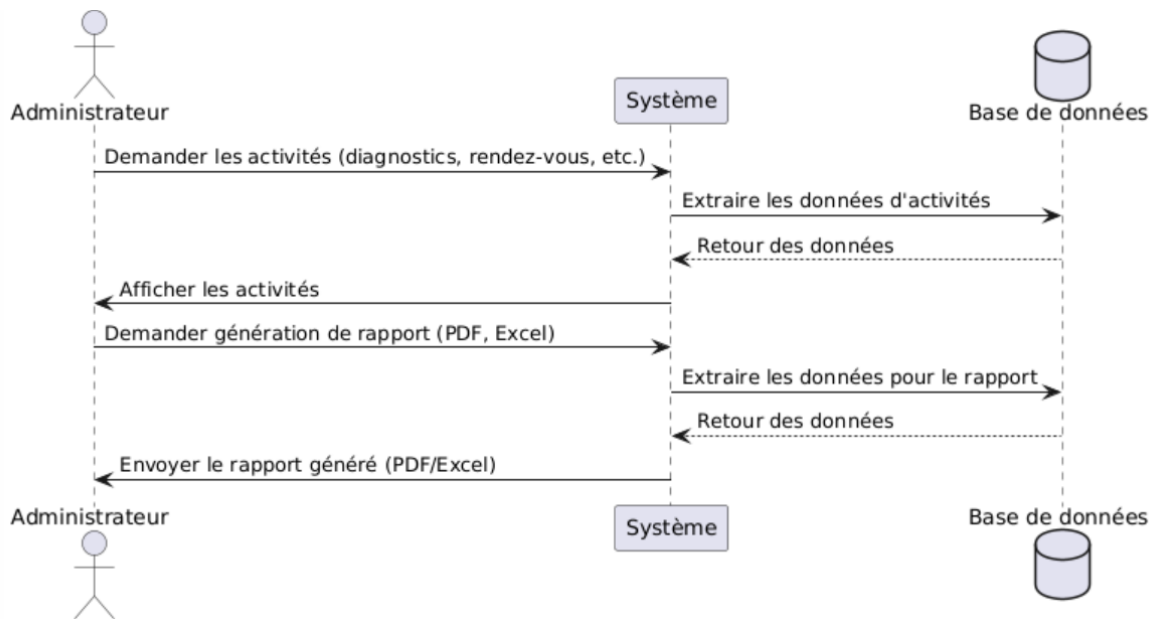


Figure 2-0-9 : Diagramme de séquence de supervision et rapport

2.7. Conclusion Partielle

Dans ce chapitre, nous avons présenté les éléments fondamentaux liés à la conception du système, notamment les diagrammes UML illustrant les cas d'utilisation et la modélisation des classes. L'analyse du diagramme de classes a permis d'identifier les principales entités du système ainsi que leurs relations, mettant en évidence les interactions entre les patients, les médecins, les administrateurs et le chatbot.

L'association entre le patient et le chatbot a également été clarifiée, soulignant le rôle de ce dernier dans l'analyse préliminaire des symptômes et la recommandation des médecins. Ces éléments de conception sont essentiels pour assurer une structuration efficace du système et garantir son bon fonctionnement.

La prochaine étape consistera à approfondir la mise en œuvre technique en s'appuyant sur ces modèles afin d'assurer une transition fluide entre la conception et le développement.

Chapitre 3 : Réalisation de l'Application Web

3.1. Introduction

Ce chapitre présente la phase de concrétisation technique du système de consultation médicale en ligne, articulée autour des spécifications établies au Chapitre 2. L'objectif est de décrire méthodiquement les choix d'implémentation, l'architecture logicielle, et les résultats obtenus, tout en évaluant les défis rencontrés. La démarche adoptée suit un cycle de développement itératif, intégrant les bonnes pratiques des méthodologies en Cascade pour répondre aux exigences fonctionnelles et non fonctionnelles identifiées.

3.2. Technologies utilisées

Le développement de la plateforme de consultation médicale en ligne s'est appuyé sur un ensemble cohérent de technologies modernes, sélectionnées pour leur compatibilité et leur efficacité. Ces technologies ont été organisées selon les différentes couches du système, chacune jouant un rôle spécifique dans le bon fonctionnement de l'application.

➤ Frontend

La couche d'interface utilisateur a été développée avec React.js, une bibliothèque JavaScript performante, permettant la création de composants interactifs et dynamiques. Pour la conception visuelle, Tailwind CSS a été utilisé afin de garantir une interface à la fois esthétique, responsive et facilement personnalisable. Le choix de **TypeScript** pour le développement frontend a permis d'introduire un typage statique, réduisant les erreurs et améliorant la maintenabilité du code. Pour assurer la communication entre le frontend et les différentes API, la bibliothèque Axios a été intégrée. Elle facilite les appels HTTP, notamment pour envoyer les symptômes du patient au modèle d'IA, ou pour interagir avec le backend (gestion des rendez-vous, authentification, récupération de l'historique).

➤ Backend

La logique métier ainsi que la gestion des routes ont été centralisées au sein du backend, développé avec Node.js, un environnement d'exécution JavaScript non bloquant, reconnu pour sa performance dans les applications web modernes. Le framework Express.js a été utilisé pour structurer les routes, définir les *middlewares*, gérer les sessions et créer les différents *endpoints* nécessaires au bon fonctionnement de la plateforme.

Pour le stockage des données critiques telles que les utilisateurs, les diagnostics, les historiques médicaux ou encore les rendez-vous, le système repose principalement sur Firebase Firestore, une base de données NoSQL dans le cloud, offrant une synchronisation en temps réel et une grande scalabilité. L'authentification des utilisateurs est assurée via Firebase Authentication, garantissant une gestion sécurisée et fluide des connexions selon les différents rôles (patient, médecin, administrateur).

En parallèle, pour le développement et les tests en local, JSON Server a été utilisé comme solution légère de simulation d'API REST. Cet outil a permis de structurer une base de données fictive au format JSON, facilitant ainsi le prototypage rapide des fonctionnalités du backend. Cette approche a permis une meilleure séparation des couches de développement et une validation préalable des flux d'information.

➤ **Intelligence Artificielle (IA)**

Le cœur du système de diagnostic repose sur une intelligence artificielle intégrée grâce au framework LangChain. Celui-ci permet de structurer les appels au modèle de langage et de formater les entrées textuelles issues des utilisateurs.

Les prompts ainsi formatés sont transmis à l'API Fireworks, qui donne accès au modèle LLaMA. Ce dernier analyse les symptômes fournis par le patient et génère un diagnostic préliminaire accompagné d'une recommandation de spécialité médicale.

Cette interaction constitue la base de la prise en charge automatisée initiale, avant intervention du médecin pour validation ou ajustement du diagnostic.

3.3. Architecture du système

L'architecture choisie repose sur une structure à trois serveurs indépendants mais communicants :

1. **Serveur Frontend (React + Tailwind)** : Héberge l'interface utilisateur accessible via un navigateur. Il interagit avec la base de données et le serveur IA via des appels API.
2. **Serveur Backend / Base de données (Firebase + Node.js + JSON server)** : Gère l'authentification, les utilisateurs, les historiques médicaux, les rendez-vous et les ordonnances. Dans la phase de prototypage et de test local, un serveur JSON a été utilisé comme outil de simulation de base de données. Cette solution permet de générer rapidement une API REST locale à partir d'un simple fichier JSON, facilitant ainsi le développement initial des fonctionnalités avant le déploiement sur l'environnement cloud.
3. **Serveur IA (Langchain + Fireworks + modèle LLaMA)** : Dédié à la génération du diagnostic préliminaire. Le frontend lui transmet les symptômes via API, et il renvoie une réponse contextualisée.

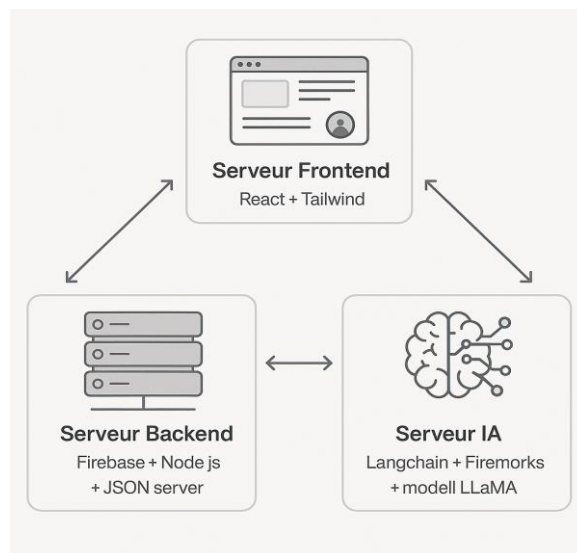


Figure 3-1: Schéma bloc présentant l'architecture global du système

3.4. Méthodologie de réalisation

La démarche adoptée est inspirée du modèle en cascade, mais intégrée dans un cycle itératif. Chaque étape a été validée avant de passer à la suivante :

1. **Analyse des besoins** : Cette première étape a consisté à recueillir, identifier et formuler les besoins des différents utilisateurs du système, à savoir les patients, les médecins et l'administrateur. Elle s'est appuyée sur l'analyse du contexte local (ville de Goma) et sur les enjeux d'accessibilité aux soins spécialisés. Les besoins ont été classés en fonctionnels (prise de rendez-vous, diagnostic, ordonnance) et non fonctionnels (sécurité, accessibilité, performance).
2. **Conception des interfaces et des cas d'utilisation** : Une fois les besoins définis, nous avons conçu l'architecture logique de l'application à travers des diagrammes de cas d'utilisation, des diagrammes de classe. Cette phase a permis de visualiser le parcours utilisateur, les rôles des acteurs, et de planifier les interactions entre les composants du système.
3. **Développement du backend** : La phase de développement s'est appuyée sur Node.js avec TypeScript pour le backend. Les opérations liées à l'authentification, la gestion des utilisateurs, la base de données, et les traitements métiers ont été développées en interaction avec Firebase. Cette couche assure la logique applicative, la sécurité et la gestion centralisée des données.
4. **Intégration de l'IA avec LangChain** : L'intégration de l'intelligence artificielle s'est faite à l'aide du framework LangChain, qui permet de connecter facilement notre application à un modèle de langage tel que LLaMA, via Fireworks API. Cette couche permet au chatbot de générer des diagnostics préliminaires en analysant les symptômes des patients.
5. **Tests fonctionnels** : Des tests manuels ont été réalisés à chaque étape du développement pour vérifier le bon fonctionnement des fonctionnalités : authentification, enregistrement, chatbot, création de rendez-vous, ordonnance. Nous avons également vérifié la cohérence

des rôles (médecins, patients, admins), la validité des diagnostics générés, et la sécurité des données utilisateurs.

3.5. Évaluation des performances

Afin d'évaluer les performances globales du système de consultation médicale en ligne, l'étude ne s'est pas limitée à l'analyse du chatbot médical, mais a porté sur l'ensemble des fonctionnalités clés intégrées dans la plateforme.

L'objectif de cette démarche est de vérifier à la fois la qualité technique, la fiabilité fonctionnelle, et l'expérience utilisateur offerte par le système dans un contexte d'usage réel.

L'évaluation a ainsi pris en compte les aspects suivants :

- La qualité de l'analyse des symptômes par le système, à travers la compréhension des expressions naturelles des patients ;
- La pertinence des diagnostics préliminaires générés par le chatbot, comparée à des diagnostics médicaux de référence ;
- La cohérence entre les diagnostics proposés et les recommandations de spécialistes associés ;
- La fluidité de la navigation dans l'interface, ainsi que son intuitivité pour des utilisateurs non techniciens ;
- La gestion des rendez-vous en ligne, en évaluant les délais de traitement, la fiabilité des notifications et la clarté des créneaux disponibles ;
- La consultation de l'historique médical ainsi que la génération et l'accès aux ordonnances électroniques ;
- Le bon fonctionnement du tableau de bord administratif, incluant la gestion des utilisateurs, la génération de rapports.

Une série de **tests pratiques** a été menée auprès de **18 utilisateurs** (Assistants médicaux et développeurs), qui ont chacun simulé des cas cliniques à l'aide de la plateforme. Ces tests ont permis d'observer le comportement du système dans des situations réalistes, tout en récoltant les impressions des utilisateurs

Nous avons mesuré le taux de correspondance entre les diagnostics générés par le système et ceux issus de sources médicales fiables, telles que des avis d'experts ou des documents de référence. Ce taux reflète le nombre de cas où le diagnostic était jugé conforme. Les recommandations de spécialistes ont également été prises en compte. Les évaluations ont été réalisées à l'aveugle pour éviter tout biais.

Les symptômes testés couvraient un éventail de situations fréquentes telles que : douleurs abdominales, diarrhée, fièvre, maux de tête, vision floue, fatigue, Maux de tête chroniques, etc. Il est également à noter que le système ne se limite pas à l'enregistrement de symptômes : il engage un dialogue interactif avec l'utilisateur, en posant des questions complémentaires sur les antécédents médicaux. Cela permet de reconstituer un mini-historique de la maladie, renforçant la précision du diagnostic automatisé. Un aperçu complet des questions [ici](#)

Tableau 3-1: Le tableau ci-dessous illustre un exemple de cas analysé : « Évaluation de performance »

Symptômes saisis	Diagnostic du système	Médecins recommandés	Taux (%) / Utilisateurs
Douleurs abdominales, diarrhée	Intolérance alimentaire,	Médecin généraliste ou un gastro-entérologue	88,8%
Fièvre, frissons, transpiration, maux de tête	Infection virale(grippe), Infection bactérienne (pneumonie, sinusite)	Pneumologie, Infectiologue	77,7%
Toux grasse, douleurs thoraciques, fièvre	Bronchite aiguë, Pneumonie, Infection respiratoire virale	Pneumologue, Médecin généraliste	88,8%
Maux de tête chroniques, vision floue	Migraine, Hypertension	Médecin généraliste ou un neurologue	94,4%

Soif excessive, urines fréquentes, fatigue	Diabète insipide, diabète sucré, Déshydratation	Endocrinologue ou médecin généraliste	94,4%
Fièvre élevée, Douleurs abdominales, maux de tête intenses, Fatigue et faiblesse extrême, perte d'appétit et perte de poids	Typhoïde, Infection virale ou bactérienne	Médecin généraliste, Infectiologue, gastro-entérologue	72,2%
Maux de tête, vertiges, fatigue, coups de nez, palpitations	Hypertension artérielle, Anémie	Médecin généraliste ou un cardiologue	83,3%

Remarque : Le taux utilisateur (%) représente la proportion de participants pour lesquels le diagnostic proposé correspondait aux cas de référence, établis à partir d'avis d'experts ou de documents médicaux existants.

L'analyse des résultats montre que le système de diagnostic médical présente une performance globalement satisfaisante. Les taux de correspondance entre les diagnostics proposés par le système et ceux issus de la référence médicale indiquent une bonne capacité d'analyse des symptômes saisis. Les pourcentages relativement élevés démontrent que, dans la majorité des cas, le système parvient à identifier des diagnostics proches ou identiques à ceux posés par des professionnels de santé.

En dehors des performances du module de diagnostic, l'évaluation a également porté sur les autres fonctionnalités clés du système. Les utilisateurs ont jugé l'interface intuitive et fluide, la gestion des rendez-vous simple et efficace, ainsi que l'accès aux données médicales bien structuré. Le temps de réponse du système, évalué à une moyenne de 10 secondes, a été jugé satisfaisant, contribuant à une expérience utilisateur rapide et agréable. Le tableau de bord administratif permet de gérer les comptes utilisateurs et de générer des rapports d'activité basés sur les consultations réalisées. Globalement, ces éléments renforcent la qualité d'expérience et démontrent que le système est non seulement fonctionnel, mais aussi adapté à une utilisation réelle dans un contexte de télémédecine.

3.6. Présentation du fonctionnement de l'application

Pour illustrer concrètement le fonctionnement de l'application de consultation médicale en ligne, nous présentons ci-dessous une simulation d'utilisation typique par un patient. Cette démarche a pour objectif de montrer, étape par étape, comment l'utilisateur interagit avec le système, et comment les différents modules (chatbot, diagnostic, recommandation, rendez-vous, génération d'une ordonnance électronique, etc) interviennent pour répondre à son besoin.

3.6.1. Interface Patient

1. Page d'accueil de la plateforme (landing page)

Cette interface représente la page d'accueil de la plateforme de consultation médicale en ligne. Elle présente une vue d'ensemble des services, permet d'accéder à la connexion ou à l'inscription, et offre aux visiteurs un premier aperçu des fonctionnalités proposées.



Figure 3-2: Section Héros de la page d'accueil (Landing Page)

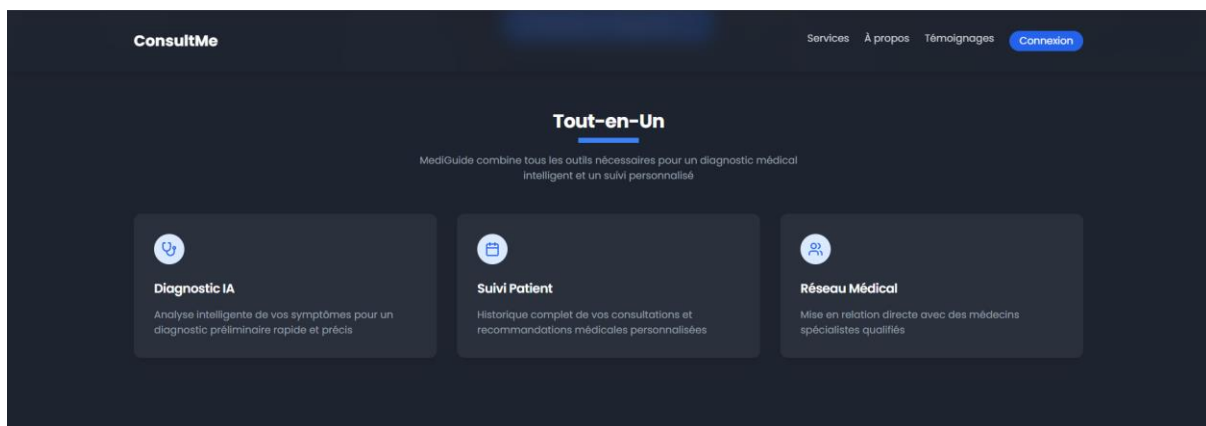


Figure 3-3: Section des services proposés sur la page d'accueil



Figure 3-4: Section “À propos” de la page d’accueil

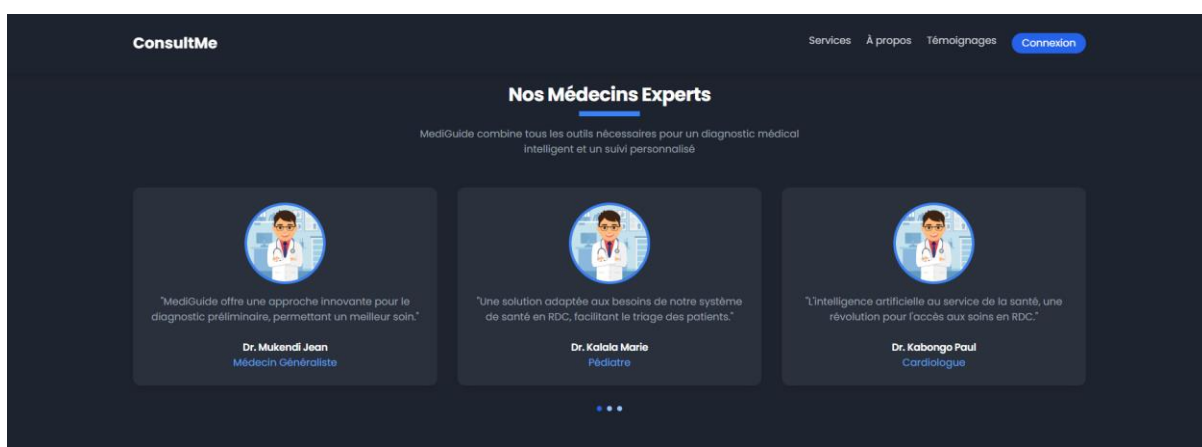


Figure 3-5 Présentation des médecins sur la page d’accueil

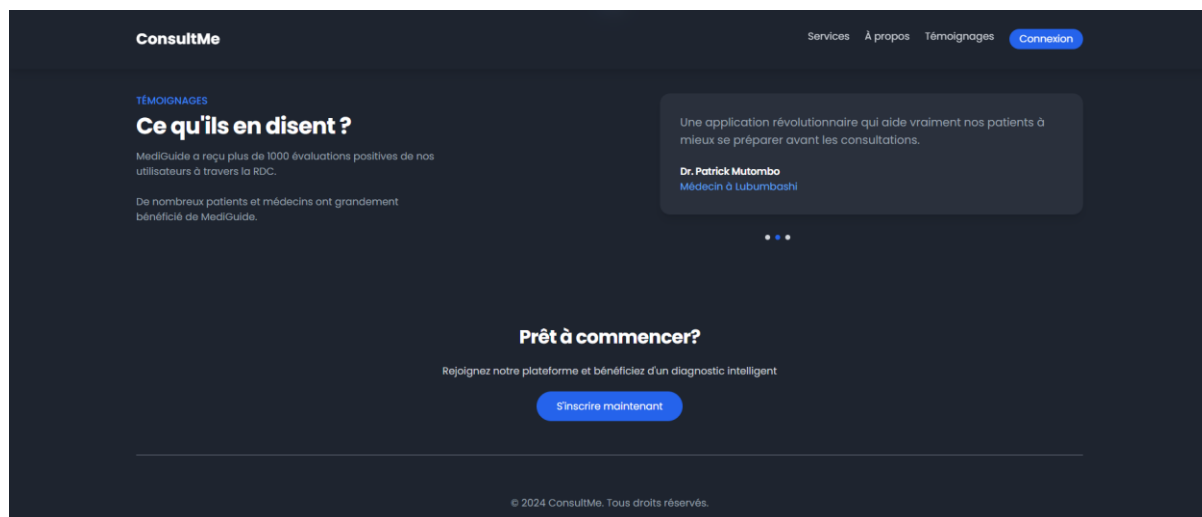


Figure 3-6: Section des témoignages utilisateurs

2. Interface de connexion à la plateforme

Cette page permet aux utilisateurs enregistrés (patients, médecins ou administrateurs) de s'authentifier de manière sécurisée à l'aide de leur identifiant et mot de passe. Elle constitue la porte d'entrée vers l'ensemble des services personnalisés de la plateforme.

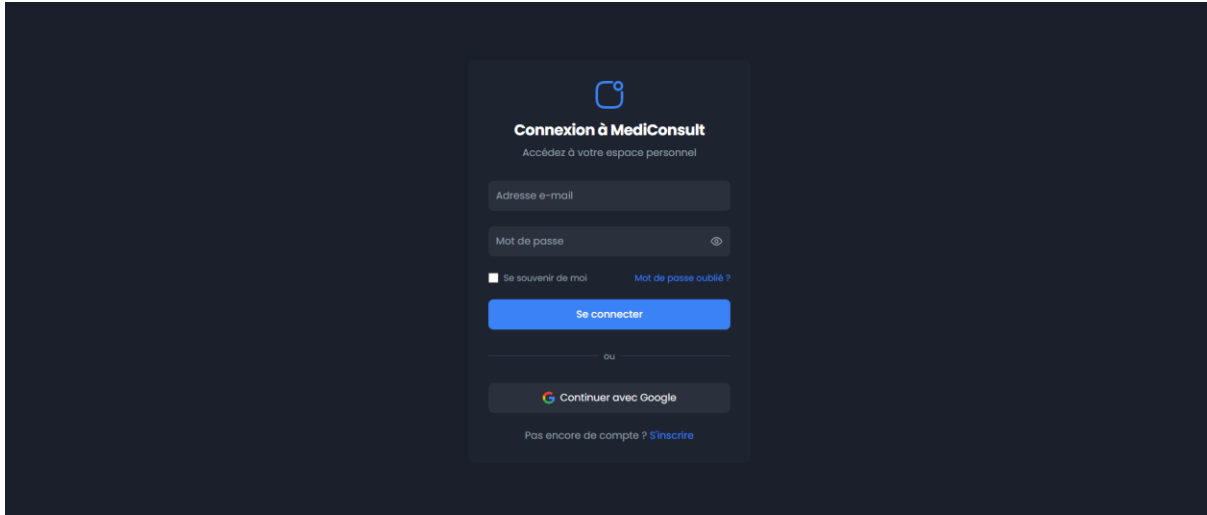


Figure 3-7: Interface de connexion des utilisateurs

3. Formulaire de création de compte pour patient

Cette interface permet à un nouveau patient de s'inscrire sur la plateforme en remplissant un formulaire de création de compte. Les informations saisies sont ensuite validées et enregistrées dans la base de données afin d'activer l'accès personnalisé.

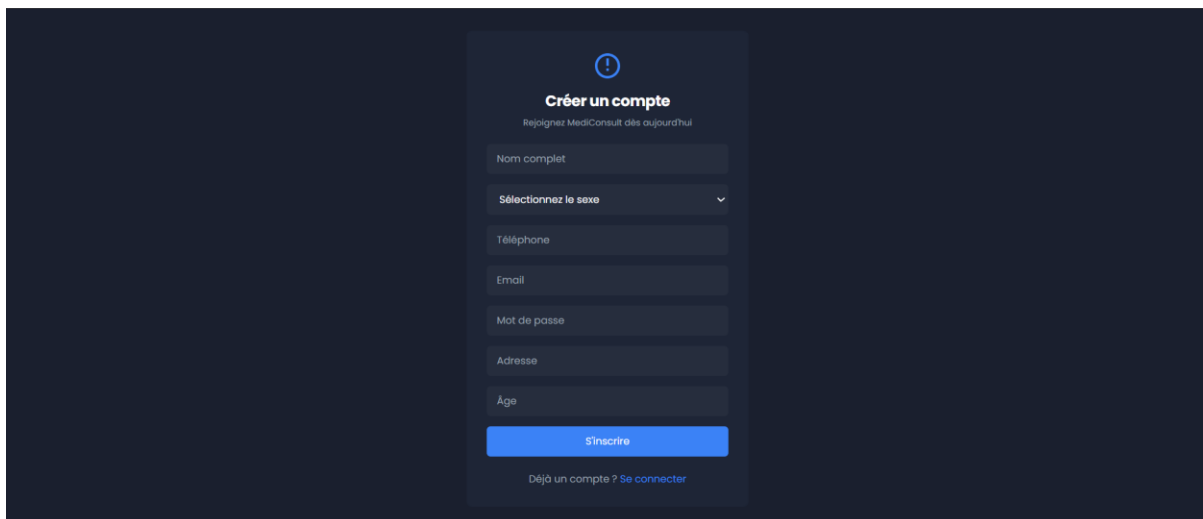


Figure 3-8 : Interface d'inscription des utilisateurs

4. Interface d'interaction entre le patient et le chatbot médical

Cette interface illustre l'échange interactif entre le patient et le chatbot basé sur le modèle LLaMA. Le chatbot recueille les symptômes décrits par le patient afin de proposer un diagnostic préliminaire et orienter l'utilisateur vers un médecin spécialiste qui se trouve dans notre système.

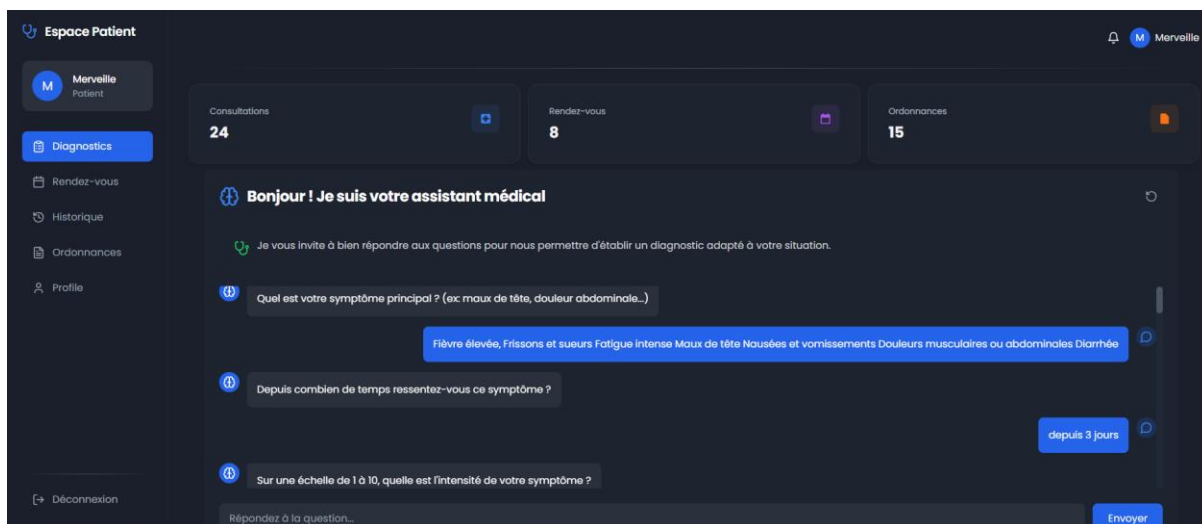


Figure 3-9 : Interface de discussion entre le patient et le chatbot médical

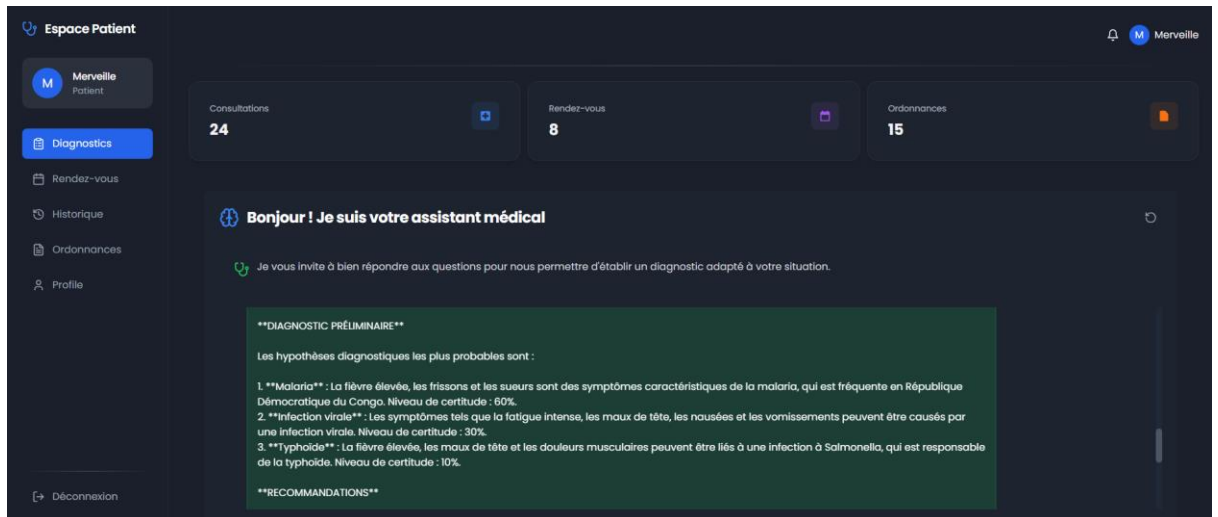


Figure 3-10: Affichage des résultats du diagnostic préliminaire

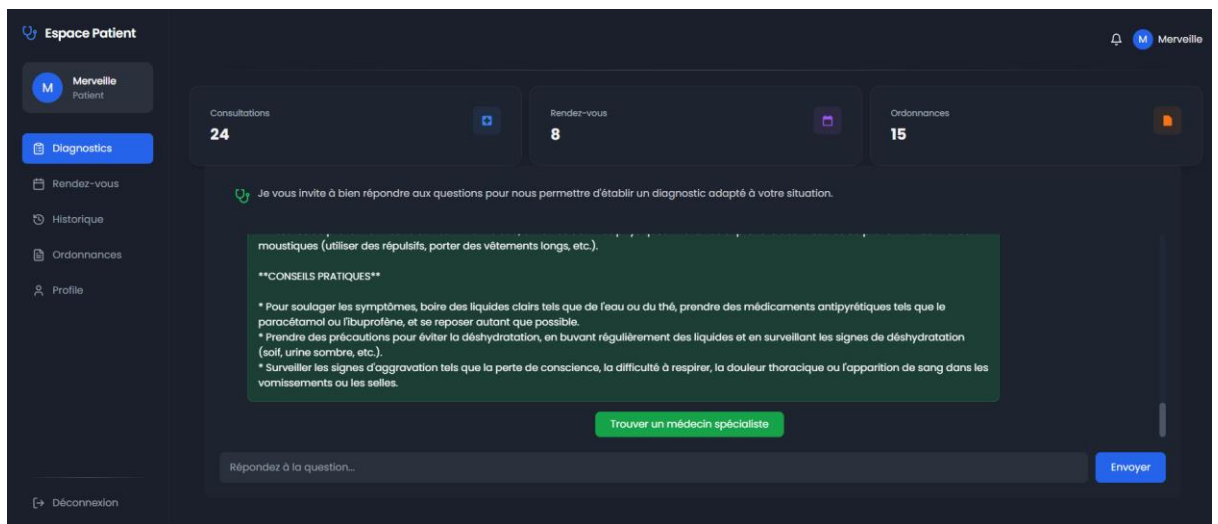


Figure 3-11: Bouton de recherche de médecin disponible

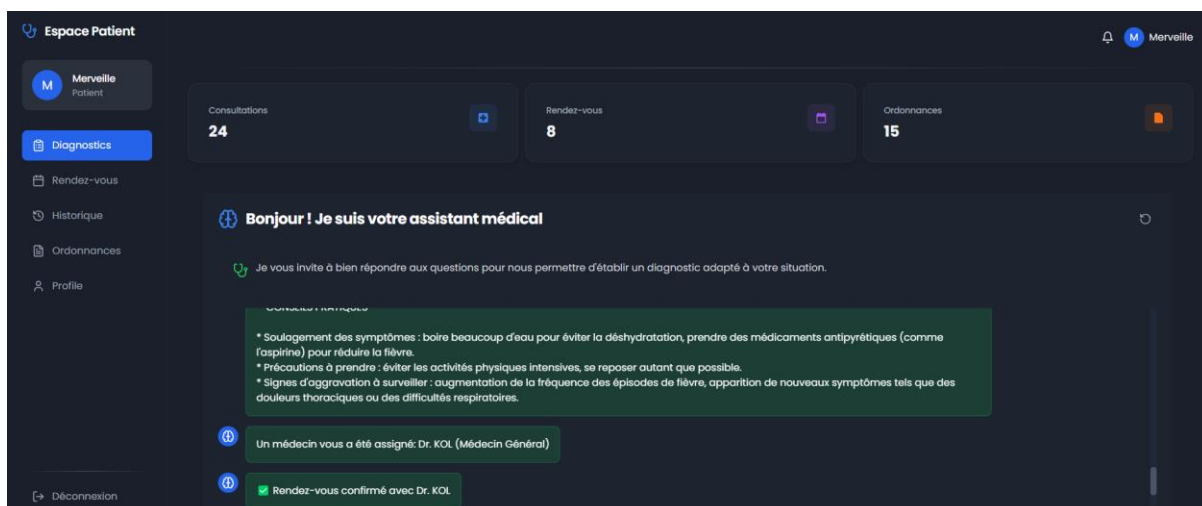


Figure 3-12: Confirmation de l'attribution d'un médecin spécialiste

5. Espace de gestion des rendez-vous et de l'historique médical du patient

Cette section permet au patient de consulter son historique médical (diagnostics, ordonnances, consultations passées et rendez-vous). Elle centralise les informations médicales personnelles pour un meilleur suivi.

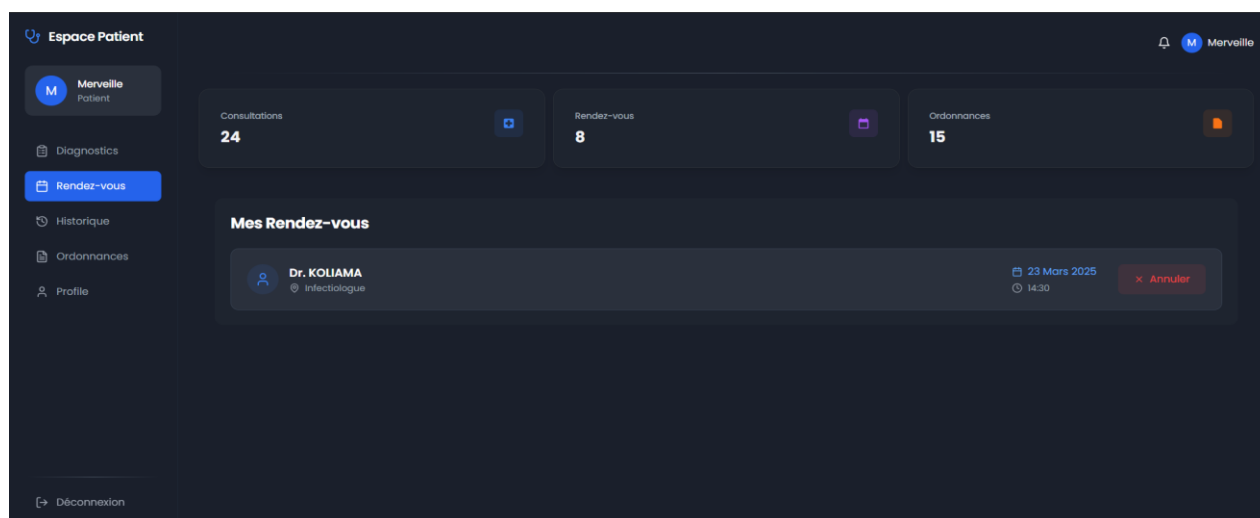


Figure 3-13: Liste des rendez-vous du patient

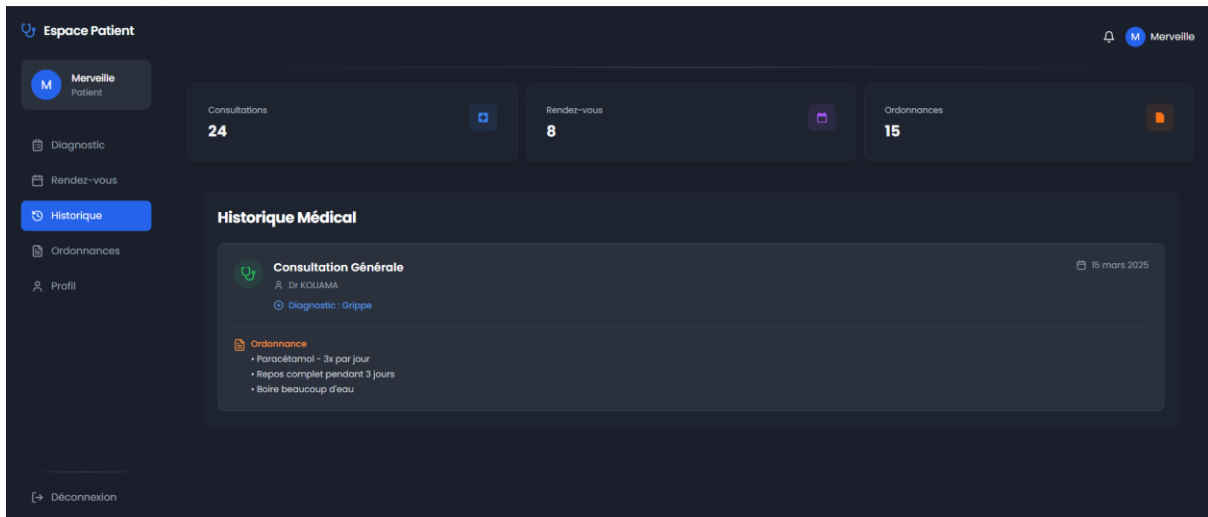


Figure 3-14: Interface de consultation de l'historique médical

3.6.2. Espace Médecin

1. Liste des rendez-vous et diagnostics des patients

Cet espace dédié au médecin lui donne un accès organisé aux rendez-vous planifiés avec ses patients ainsi qu'aux diagnostics en attente de validation ou de mise à jour. Il constitue l'outil principal de gestion des consultations pour les professionnels de santé.

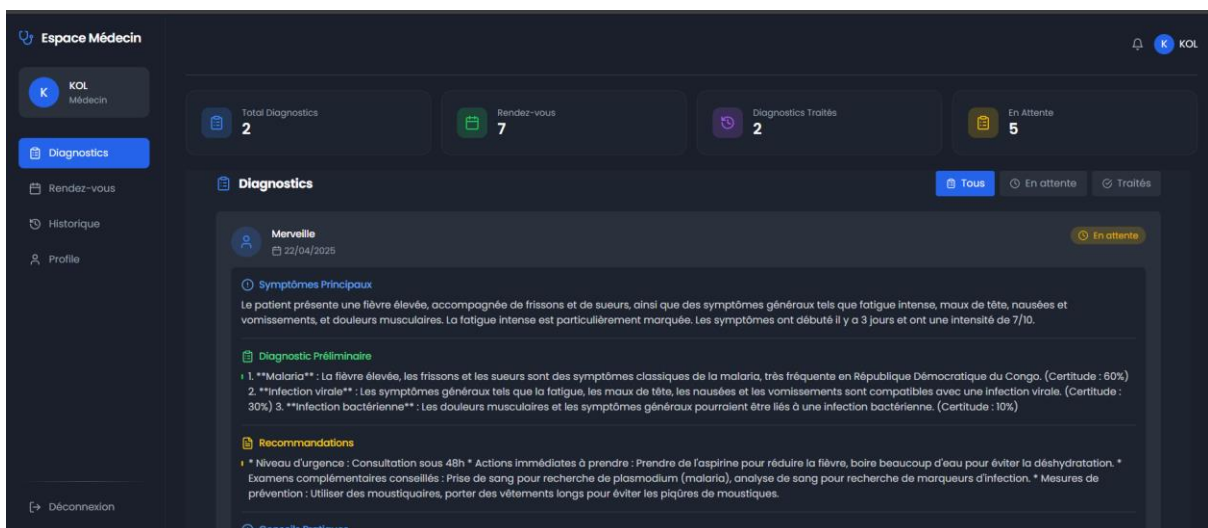


Figure 3-15: Liste des diagnostics en attente de validation

2. Ordonnance électronique

Cette interface permet au médecin de rédiger, valider et envoyer une ordonnance électronique au patient après avoir confirmé le diagnostic. L'ordonnance est ensuite automatiquement archivée dans l'historique médical du patient.

The screenshot shows the 'Espace Médecin' interface with a sidebar on the left containing 'Diagnostics', 'Rendez-vous', 'Historique', and 'Profil'. The main content area is titled 'Nouvelle Ordonnance' and contains a form for patient information. At the top, there are four summary cards: 'Total Diagnostics' (2), 'Rendez-vous' (7), 'Diagnostics Traités' (2), and 'En Attente' (5). The patient information form includes fields for 'Nom du patient' (Merveille), 'Téléphone' (0975408321), 'Email' (mer@gmail.com), and 'Nom du docteur' (KOL). A 'Symptômes' section contains a text area with the following text: '**SYNTHÈSE DES SYMPTÔMES**
Le patient présente une fièvre élevée, accompagnée de frissons et de sueurs, ainsi que des symptômes généraux tels que fatigue intense, maux de tête, nausées et vomissements, et douleurs musculaires. La fatigue intense est particulièrement marquée. Les symptômes ont débuté il y a 3 jours et ont une intensité de 7/10.'

Figure 3-16: Information du patient avant l'envoi d'ordonnance

The screenshot shows the 'Espace Médecin' interface with the sidebar on the left. The main content area is titled 'Prescription' and contains a form for medication details. At the top, there are four summary cards: 'Total Diagnostics' (2), 'Rendez-vous' (7), 'Diagnostics Traités' (2), and 'En Attente' (5). The prescription form includes fields for 'Nom du médicament' (Nom du médicament), 'Forme pharmaceutique' (Sélectionnez une forme), 'Dosage' (Ex: 500mg), 'Durée du traitement' (Ex: 7 jours), and 'Posologie' (Ex: 1 comprimé 3 fois par jour). A large blue button at the bottom is labeled 'Envoyer l'ordonnance'.

Figure 3-17: Formulaire de prescription d'une ordonnance électronique

3. Interface de gestion du profil utilisateur (Patient ou Médecin)

L'utilisateur (patient ou médecin) peut consulter et modifier ses informations personnelles via cette interface. Elle permet notamment la mise à jour du mot de passe, de l'adresse, ou encore des informations professionnelles pour les médecins.

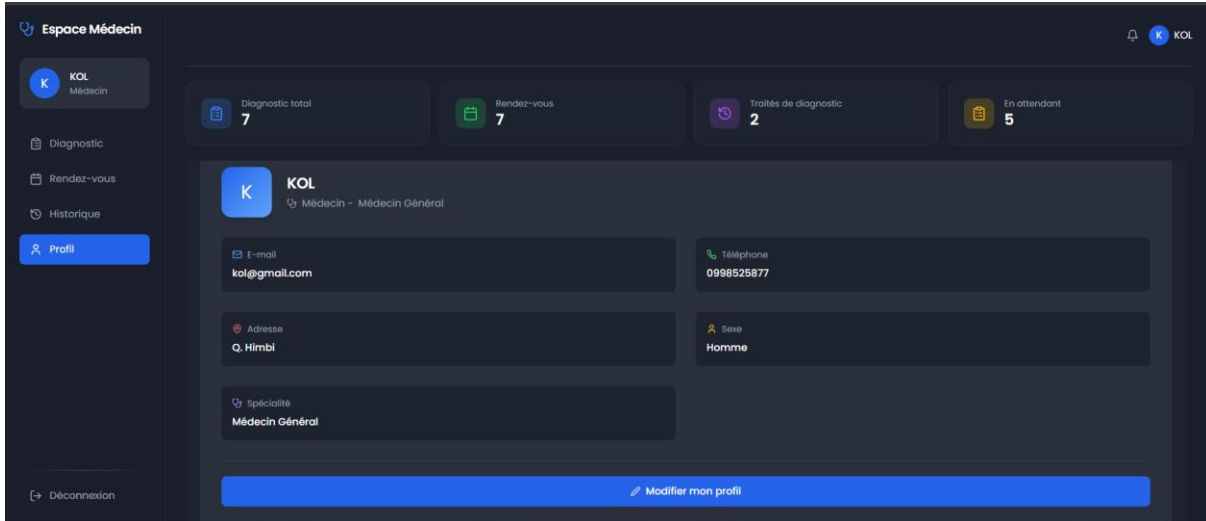


Figure 3-18 Interface du profil du médecin

3.6.3. Espace admin

1. Interface d'administration – Gestion des utilisateurs

Cette interface est réservée à l'administrateur de la plateforme. Elle permet d'ajouter (médecin), modifier ou supprimer les comptes utilisateurs (patients ou médecins), assurant ainsi un contrôle centralisé et sécurisé du système.

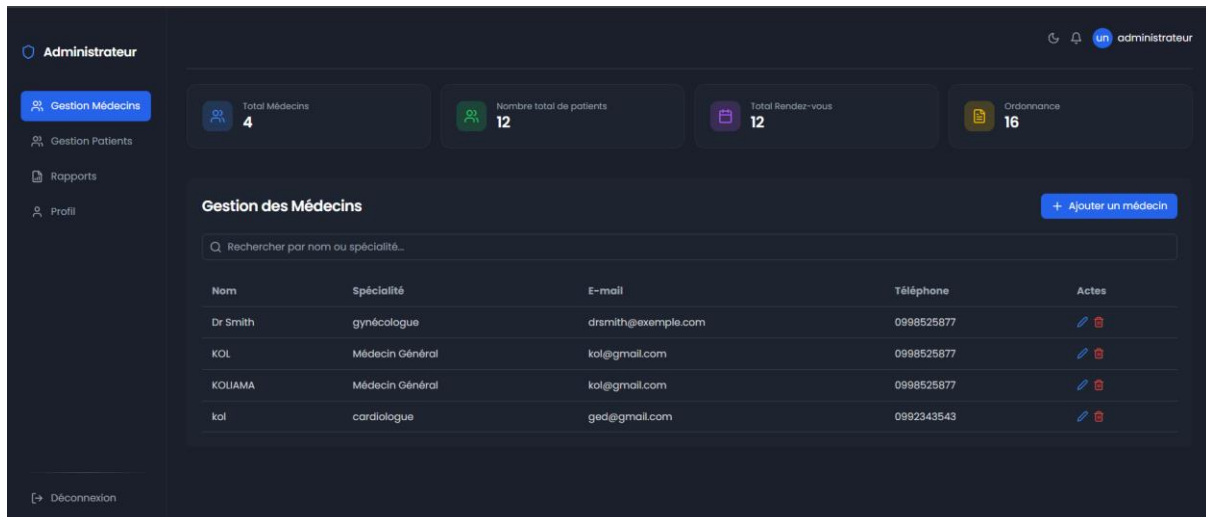


Figure 3-19: Interface de gestion des utilisateurs

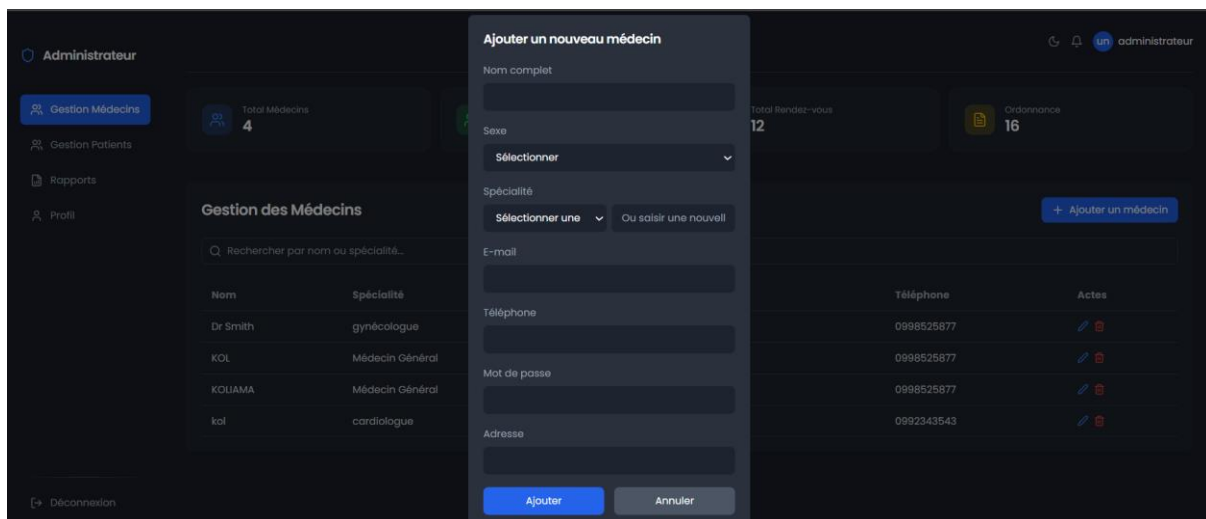


Figure 3-20: Formulaire de création d'un compte médecin

2. Interface de génération de rapports statistiques pour l'administrateur

L'administrateur peut consulter les statistiques d'utilisation de la plateforme (nombre de diagnostics, rendez-vous, ordonnances, utilisateurs actifs) et générer automatiquement des rapports en formats PDF à des fins de suivi.

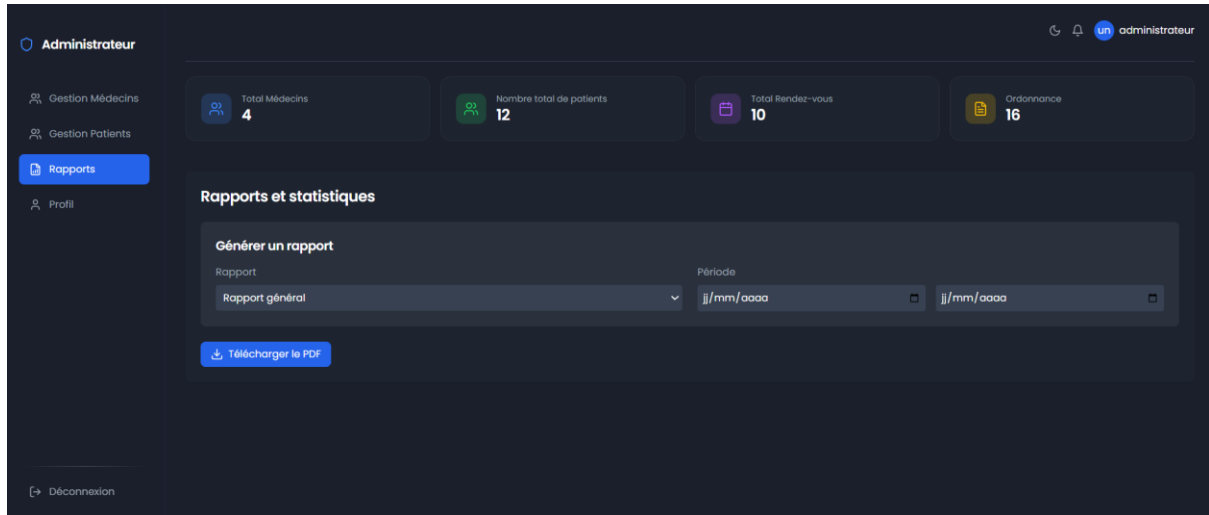


Figure 3-21: Interface de génération automatique des rapports

Ces captures démontrent la concrétisation des spécifications fonctionnelles et offrent une vue d'ensemble de l'interface utilisateur finale du système

3.7 Conclusion partielle

La réalisation de la plateforme de consultation médicale en ligne concrétise les choix technologiques et les spécifications définis lors des phases précédentes. En combinant React, Node.js, Firebase, LangChain et le modèle LLaMA, le système parvient à offrir une expérience fluide, interactive et intelligente, tout en assurant la sécurité et la traçabilité des données médicales. Le système repose sur une architecture modulaire à trois composantes, à savoir une interface utilisateur (frontend), un serveur d'application (backend) et un module d'intelligence artificielle (serveur IA). Cette séparation logique permet de mieux répartir les responsabilités, de faciliter la maintenance et d'assurer une bonne évolutivité du système. La démarche de développement suivie, itérative et structurée, a permis de valider chaque étape du processus. Les tests initiaux ont montré une cohérence encourageante entre les symptômes saisis et les diagnostics générés, posant ainsi les bases pour les futures améliorations du système.

Conclusion générale

Ce travail est un rapport accompagnant le système que nous avons conçu pour la mise en place d'une plateforme web de consultation médicale en ligne adaptée à la ville de Goma. Cette plateforme vise à pallier les difficultés d'accès aux soins spécialisés.

Le système développé intègre un chatbot médical intelligent, basé sur le modèle de langage LLaMA, permettant de proposer un diagnostic préliminaire à partir des symptômes renseignés par l'utilisateur. Ce diagnostic est soumis à validation par un médecin.

L'architecture du système repose sur une séparation claire entre frontend, backend et module d'intelligence artificielle. Nous avons utilisé React.js, Node.js, Firebase et LangChain pour garantir modularité, sécurité et évolutivité de la plateforme.

La phase de tests a permis de vérifier la pertinence des fonctionnalités principales, telles que la fiabilité du diagnostic préliminaire, la gestion des rendez-vous, la génération d'ordonnances électroniques et la supervision des activités administratives. Malgré quelques limitations, notamment l'accès restreint aux modèles d'IA open source et la complexité de gestion des profils utilisateurs multiples, le système reste stable et fonctionnel.

L'analyse finale permet de confirmer les hypothèses formulées au début de ce travail. Nous avons validé que l'intégration d'un chatbot médical peut effectivement faciliter l'orientation des patients vers des soins spécialisés, améliorer l'efficacité des consultations et contribuer à pallier la pénurie de médecins spécialistes dans les zones sous-desservies. Par ailleurs, l'hypothèse selon laquelle une plateforme numérique adaptée aux réalités locales pourrait accroître l'accessibilité et la qualité des soins a également été confirmée.

L'ensemble des objectifs fixés au début de ce travail a été atteint. Le système permet une meilleure orientation des patients et améliore l'accès aux soins spécialisés dans les zones sous-desservies. La plateforme est sécurisée, accessible et adaptée aux réalités locales.

En guise d'ouverture, plusieurs pistes d'amélioration sont envisageables :

- ✓ Intégrer une messagerie instantanée entre médecins et patients ;
- ✓ Développer une application mobile multiplateforme pour renforcer l'accessibilité ;
- ✓ Enrichir progressivement le chatbot avec des cas cliniques réels ;
- ✓ Intégrer un moteur de recommandation personnalisé basé sur l'historique médical du patient.

Ainsi, ce travail pose les bases d'un système de santé numérique, intelligent et inclusif, capable d'évoluer pour mieux répondre aux besoins locaux.

Bibliographie

- [1] W. H. Organization, «World Health Statistics 2022: Monitoring Health for the SDGs,» 2022.
- [2] chu-rouen, «Télémédecine,» chu-rouen, [En ligne]. Available: <https://www.chu-rouen.fr/telemedecine/>. [Accès le 03 Novembre 2024].
- [3] R. FRANÇAISE, «Télésanté : Télémédecine (téléconsultation, télésurveillance...),» RÉPUBLIQUE FRANÇAISE , [En ligne]. Available: <https://www.service-public.fr/particuliers/vosdroits/F34696>. [Accès le 04 11 2024].
- [4] arxiv, «Biais d'automatisation dans la prise de décision médicale assistée par l'IA sous pression temporelle en pathologie computationnelle,» [En ligne]. Available: <https://arxiv.org/html/2411.00998v1>. [Accès le 01 Décembre 2024].
- [5] M. B. H. D. E. D. C. Omar Bentahar, «Les barrières et leviers à l'adoption d'une innovation médicale : le cas de la cabine de télémédecine,» *Gestion 2000, 2021, 38 (3)*, n° %1<https://hal.science/hal-03361254/>, pp. .119-141..
- [6] IBM, «Qu'est-ce que l'intelligence artificielle (IA) ?,» IBM. [En ligne]. [Accès le 05 Novembre 2024].
- [7] AWS, «What Is Machine Learning?,» Amazone, [En ligne]. Available: https://aws.amazon.com/fr/what-is/machine-learning/?nc1=h_ls. [Accès le 14 Novembre 2024].
- [8] IBM, «Qu'est-ce qu'un chatbot ?,» IBM, [En ligne]. Available: <https://www.ibm.com/fr-fr/topics/chatbots#:~:text=IBM,humaine%20avec%20un%20utilisateur%20fina>. [Accès le 05 Novembre 2024].
- [9] Datascientest, «Natural Language Processing (NLP) : Définition et principes,» datascientest.com, [En ligne]. Available: <https://datascientest.com/introduction-au-nlp-natural-language-processing>. [Accès le 05 Novembre 2024].

- [10] Cloudflare, «Qu'est-ce qu'un grand modèle linguistique (LLM) ?», Cloudflare.com, [En ligne]. Available: <https://www.cloudflare.com/fr-fr/learning/ai/what-is-large-language-model/>. [Accès le 05 Novembre 2024].
- [11] <https://aws.amazon.com/>, «autoregressive-models», [En ligne]. Available: <https://aws.amazon.com/fr/what-is/autoregressive-models/>. [Accès le 02 Décembre 2024].
- [12] K. Raphael, «Les Autoencoders – modèles d'apprentissage non supervisé», datascientest.com, [En ligne]. Available: <https://datascientest.com/les-autoencoders-modeles-dapprentissage-non-supervise>. [Accès le 02 Décembre 2024].
- [13] I. Talbi, «L'architecture Seq2Seq en deep learning : fonctionnement et limites», pp. <https://larevueia.fr/larchitecture-seq2seq-en-deep-learning-fonctionnement-et-limites/>, 03 Décembre 2024.
- [14] datascientest, «Multimodal Learning : Qu'est ce que c'est ? A quoi ça sert ?», <https://datascientest.com/>, [En ligne]. Available: <https://datascientest.com/multimodal-learning-tout-savoir>. [Accès le 03 Décembre 2024].
- [15] U. M. William L. Hamilton, Graph Representation Learning, Montréal: Morgan & Claypool publishers., 2020.
- [16] AWS, «Qu'est-ce que le RLHF ?», aws.amazon.com, [En ligne]. Available: <https://aws.amazon.com/fr/what-is/reinforcement-learning-from-human-feedback/>. [Accès le 2024 Décembre 2024].
- [17] .intelligence-artificielle-school., «Qu'est-ce que LLaMA, le modèle linguistique développé par Meta ?», www.intelligence-artificielle-school.com, [En ligne]. Available: <https://www.intelligence-artificielle-school.com/ecole/technologies/large-language-model-meta-ai/>. [Accès le 04 Décembre 2024].
- [18] Meta, «Building Generative AI Responsibly», Septembre 2023. [En ligne]. Available: ai.meta.com/static-resource/building-generative-ai-responsibly. [Accès le 04 Décembre 2024].

- [19] wikipedia, «Diagramme de cas d'utilisation,» /fr.wikipedia.org, [En ligne]. Available: https://fr.wikipedia.org/wiki/Diagramme_de_cas_d%27utilisation. [Accès le 07 Janvier 2025].
- [20] wikipedia, «Cas d'utilisation,» <https://fr.wikipedia.org/>, [En ligne]. Available: https://fr.wikipedia.org/wiki/Cas_d%27utilisation#cite_note-:2-4. [Accès le 07 Janvier 2025].
- [21] [En ligne].
- [22] C. Fourrier, «Introduction à l'apprentissage automatique des graphes,» huggingface.co, 03 Janvier 2023. [En ligne]. Available: <https://huggingface.co/blog/intro-graphml>. [Accès le 04 Décembre 2024].

ANNEXES

1) Code du modèle

```
1. import { ChatFireworks } from "@langchain/community/chat_models/fireworks";
2. import { HumanMessage, SystemMessage } from "@langchain/core/messages";
3. import dotenv from 'dotenv';
4.
5. dotenv.config();
6.
7. interface SymptomData {
8.     principal: string;
9.     duree: string;
10.    intensite: string;
11.    symptomesAssocies: string[];
12.    antecedents: string;
13.    medicaments: string;
14.    allergies: string;
15.}
16.
17. export class MediGuideModel {
18.     private static instance: MediGuideModel;
19.     private model: ChatFireworks;
20.     private systemPrompt: SystemMessage;
21.     private currentStep: number = 0;
22.     private symptomData: Partial<SymptomData> = {};
23.     private readonly questions = [
24.         "Quel est votre symptôme principal ? (ex: maux de tête, douleur abdominale...)",
25.         "Depuis combien de temps ressentez-vous ce symptôme ?",
26.         "Sur une échelle de 1 à 10, quelle est l'intensité de votre symptôme ?",
27.         "Quels autres symptômes associez-vous à ce problème ? (liste séparée par des virgules)",
28.         "Avez-vous des antécédents médicaux pertinents ?",
29.         "Prenez-vous actuellement des médicaments ? Lesquels ?",
30.         "Avez-vous des allergies connues ?"
31.     ];
32.
33.     private constructor() {
34.         const apiKey = process.env.FIREWORKS_API_KEY;
35.         if (!apiKey) {
36.             throw new Error("FIREWORKS_API_KEY is not set in environment variables");
37.         }
38.     }
39. }
```

```

38.
39.     this.model = new ChatFireworks({
40.         apiKey,
41.         model: "accounts/fireworks/models/llama-v3-70b-instruct",
42.         temperature: 0.7,
43.         maxTokens: 2000,
44.     });
45.
46.     this.systemPrompt = new SystemMessage(`Tu es un assistant médical
virtuel professionnel spécialisé dans le diagnostic préliminaire.
47.
48.     Contexte : Tu opères en République Démocratique du Congo (RDC) et
dois adapter tes réponses au contexte médical local.
49.
50.     Après avoir analysé les 7 informations clés du patient, tu dois
fournir une réponse structurée comme suit :
51.
52.     1. SYNTHÈSE DES SYMPTÔMES
53.     - Résumer clairement les informations principales
54.     - Mettre en évidence les points d'attention
55.
56.     2. DIAGNOSTIC PRÉLIMINAIRE
57.     - Proposer 2-3 hypothèses diagnostiques probables
58.     - Expliquer brièvement chaque hypothèse
59.     - Indiquer le niveau de certitude
60.
61.     3. RECOMMANDATIONS
62.     - Niveau d'urgence : immédiat / 48h / surveillance
63.     - Actions immédiates à prendre
64.     - Examens complémentaires conseillés
65.     - Mesures de prévention
66.
67.     4. ORIENTATION MÉDICALE
68.     - Type de spécialiste recommandé
69.     - Justification du choix du spécialiste
70.     - Conseils pour la consultation
71.
72.     5. CONSEILS PRATIQUES
73.     - Soulagement des symptômes
74.     - Précautions à prendre
75.     - Signes d'aggravation à surveiller
76.
77.     Style de communication :

```

```

78.     - Utiliser un ton professionnel et empathique
79.     - Expliquer les termes médicaux en langage simple
80.     - Adapter les recommandations au contexte local
81.     - Être précis et factuel dans les explications`);
82. }
83.
84. public static async getInstance(): Promise<MediGuideModel> {
85.     if (!MediGuideModel.instance) {
86.         MediGuideModel.instance = new MediGuideModel();
87.     }
88.     return MediGuideModel.instance;
89. }
90.
91. public getCurrentQuestion(): string {
92.     return this.currentStep < this.questions.length ?
93.     this.questions[this.currentStep] : "";
94. }
95. public async processAnswer(answer: string): Promise<{ question: string
96. | null; diagnostic?: string }> {
97.     if (!answer.trim()) {
98.         throw new Error("Answer cannot be empty");
99.     }
100.     if (this.currentStep >= this.questions.length) {
101.         return { question: null };
102.     }
103.
104.     this.saveAnswer(answer);
105.     this.currentStep++;
106.
107.     if (this.currentStep < this.questions.length) {
108.         return { question: this.getCurrentQuestion() };
109.     } else {
110.         const diagnostic = await this.generateDiagnostic();
111.         return { question: null, diagnostic };
112.     }
113. }
114.
115. private saveAnswer(answer: string): void {
116.     switch (this.currentStep) {
117.         case 0: this.symptomData.principal = answer; break;
118.         case 1: this.symptomData.duree = answer; break;

```

```

119.         case 2: this.symptomData.intensite = answer; break;
120.         case 3: this.symptomData.symptomesAssocies =
    answer.split(',').map(s => s.trim()); break;
121.         case 4: this.symptomData.antecedents = answer; break;
122.         case 5: this.symptomData.medicaments = answer; break;
123.         case 6: this.symptomData.allergies = answer; break;
124.     }
125. }
126.
127.     private async generateDiagnostic(): Promise<string> {
128.         const prompt = `Sur la base des informations suivantes:
129.         - Symptôme principal: ${this.symptomData.principal}
130.         - Durée: ${this.symptomData.duree}
131.         - Intensité: ${this.symptomData.intensite}/10
132.         - Symptômes associés:
    ${this.symptomData.symptomesAssocies?.join(', ')}
133.         - Antécédents: ${this.symptomData.antecedents}
134.         - Médicaments: ${this.symptomData.medicaments}
135.         - Allergies: ${this.symptomData.allergies}
136.
137.         Fournis:
138.         1. Un diagnostic différentiel en 3 points maximum
139.         2. Des recommandations adaptées
140.         3. Le degré d'urgence (consultation immédiate/sous
    48h/surveillance)
141.         4. Des conseils pour le soulagement des symptômes`;
142.
143.         try {
144.             const response = await
    this.model.invoke([this.systemPrompt, new HumanMessage(prompt)]);
145.             return response.content as string;
146.         } catch (error) {
147.             console.error("Error generating diagnostic:", error);
148.             throw new Error("Failed to generate diagnostic");
149.         }
150.     }
151.
152.     public resetQuestionnaire(): void {
153.         this.currentStep = 0;
154.         this.symptomData = {};
155.     }
156. }

```

2) Code de création d'API

```
3. import express from 'express';
4. import cors from 'cors';
5. import dotenv from 'dotenv';
6. import mediGuideRouter from './api/mediGuideApi';
7.
8. dotenv.config();
9.
10. const app = express();
11. const PORT = process.env.PORT || 5000;
12.
13. app.use(cors({
14.   origin: ['http://localhost:5173', 'http://localhost:3000'],
15.   credentials: true
16. }));
17.
18. app.use(express.json());
19. app.use('/api/mediguide', mediGuideRouter);
20.
21. app.listen(PORT, () => {
22.   console.log(`Server running at http://localhost:${PORT}`);
23. });
```

3) Code pour envoyer le diagnostic au médecin

```
4) import { User } from 'lucide-react';
5) import axios from 'axios';
6) import { Doctor, PatientDiagnostic } from "../../Types/Doctor";
7) import Patient from "../../Types/Patient";
8)
9) interface DiagnosticRequest {
10)   symptoms: string;
11)   specialiste: string;
12) }
13)
14) export const findSpecialist = async (diagnostic: DiagnosticRequest):
    Promise<Doctor | null> => {
15)   try {
16)     // Get patient info from LocalStorage
17)     const patient = JSON.parse(localStorage.getItem('user') || '{}');
18)
```

```

19)     const response = await axios.get('http://localhost:3000/doctors');
20)     const doctors = response.data;
21)
22)     // Extract all possible specialties from the diagnostic symptoms
23)     const possibleSpecialties = diagnostic.symptoms
24)         .toLowerCase()
25)         .split(/[,.]/)
26)         .map(s => s.trim())
27)         .filter(s => s.length > 0);
28)
29)     console.log('Possible specialties from diagnostic:',
possibleSpecialties);
30)
31)     // Find doctors matching any of the possible specialties
32)     const specialists = doctors.filter((doctor: Doctor) => {
33)         const doctorSpeciality =
doctor.specialite.toLowerCase().trim();
34)
35)         // Check if doctor's specialty matches any of the possible
specialties
36)         return possibleSpecialties.some(specialty =>
37)             doctorSpeciality.includes(specialty) ||
38)             specialty.includes(doctorSpeciality) ||
39)             doctorSpeciality === "generaliste"
40)     });
41) });
42)
43) console.log('Found matching specialists:', specialists);
44)
45) if (specialists.length === 0) {
46)     // Fallback to general practitioners if no specialist found
47)     const generalistes = doctors.filter(
48)         (doctor: Doctor) =>
doctor.specialite.toLowerCase().includes("general")
49)     );
50)
51)     if (generalistes.length > 0) {
52)         const selectedDoctor =
generalistes[Math.floor(Math.random() * generalistes.length)];
53)         await saveDiagnostic(selectedDoctor, diagnostic, patient);
54)         return selectedDoctor;
55)     }
56)     return null;

```

```

57)     }
58)
59)     // Select random specialist from matches
60)     const selectedDoctor = specialists[Math.floor(Math.random() *
specialists.length)];
61)     await saveDiagnostic(selectedDoctor, diagnostic, patient);
62)     return selectedDoctor;
63)
64)   } catch (error) {
65)     console.error('Erreur lors de la recherche du spécialiste:',
error);
66)     return null;
67)   }
68) });
69)
70) // Helper function to save diagnostic information
71) async function saveDiagnostic(doctor: Doctor, diagnostic:
DiagnosticRequest, patient: any) {
72)   // Create diagnostic object with connected patient info
73)   const newDiagnostic: PatientDiagnostic = {
74)     patientId: patient.id,
75)     patientName: patient.noms,
76)     date: new Date().toLocaleDateString(),
77)     symptoms: diagnostic.symptoms,
78)     status: "en_attente"
79)   };
80)
81)   // Add diagnostic to doctor's list
82)   doctor.diagnosticPatient.push(newDiagnostic);
83)
84)   // Update doctor in database
85)   await axios.patch(`http://localhost:3000/doctors/${doctor.id}`,
doctor);
86)
87)   // Add diagnostic to diagnostics table
88)   await axios.post('http://localhost:3000/diagnostics', {
89)     id: Date.now().toString(),
90)     patientId: patient.id,
91)     patientName: patient.noms,
92)     date: new Date().toLocaleDateString(),
93)     symptoms: diagnostic.symptoms,
94)     status: "en_attente"
95)   });

```

```
96)}  
97)  
98)export default findSpecialist;
```

4) Code pour envoyer l'ordonnance à un patient

```
import axios from 'axios';
import Ordonnance from "../../Types/Ordonnance";

export const sendOrdonnance = async (ordonnance: Ordonnance): Promise<Ordonnance>
=> {
  try {
    // Send the ordonnance to the database using axios
    const response = await axios.post('http://localhost:3000/ordonnances',
ordonnance);
    const savedOrdonnance = response.data;

    // Log de La nouvelle ordonnance
    console.log('Nouvelle ordonnance ajoutée:', {
      patient: savedOrdonnance.PatientName,
      docteur: savedOrdonnance.DoctorName,
      medicament: savedOrdonnance.nomMedicament,
      date: savedOrdonnance.date
    });

    // Log de L'ordonnance sauvegardée
    console.log('Ordonnance sauvegardée:', savedOrdonnance);

    return savedOrdonnance;
  } catch (error) {
    console.error('Erreur lors de l\'ajout de l\'ordonnance:', error);
    throw error;
  }
};

export default sendOrdonnance;
```

5) Interface Mot de passe oublié

Mot de passe oublié

Entrez votre email pour Réinitialiser votre mot de passe

Adresse e-mail

Envoyer les instructions

[Retour à la connexion](#)

Figure 21 : Gestion de mot de passer

6) Message lorsqu'aucun médecin n'est trouvé

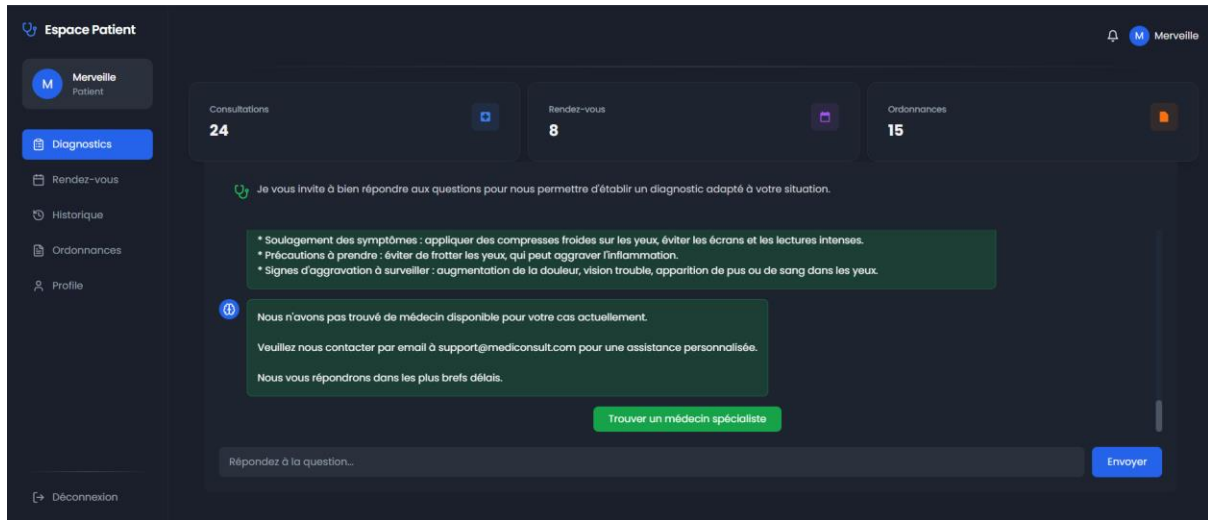


Figure 22 : Aucun médecin trouvé

7) Fiche D'ordonnance

ConsultMe

Centre Médical Spécialisé
123 Avenue de la Santé, Goma, RDC
Tél: +243 995 456 789

Dr. KOLIAMA
Médecin Généraliste
N° Ordre: MED/2023/001

Date: 29/03/2025

ORDONNANCE

Patient:
Nom: Merveille

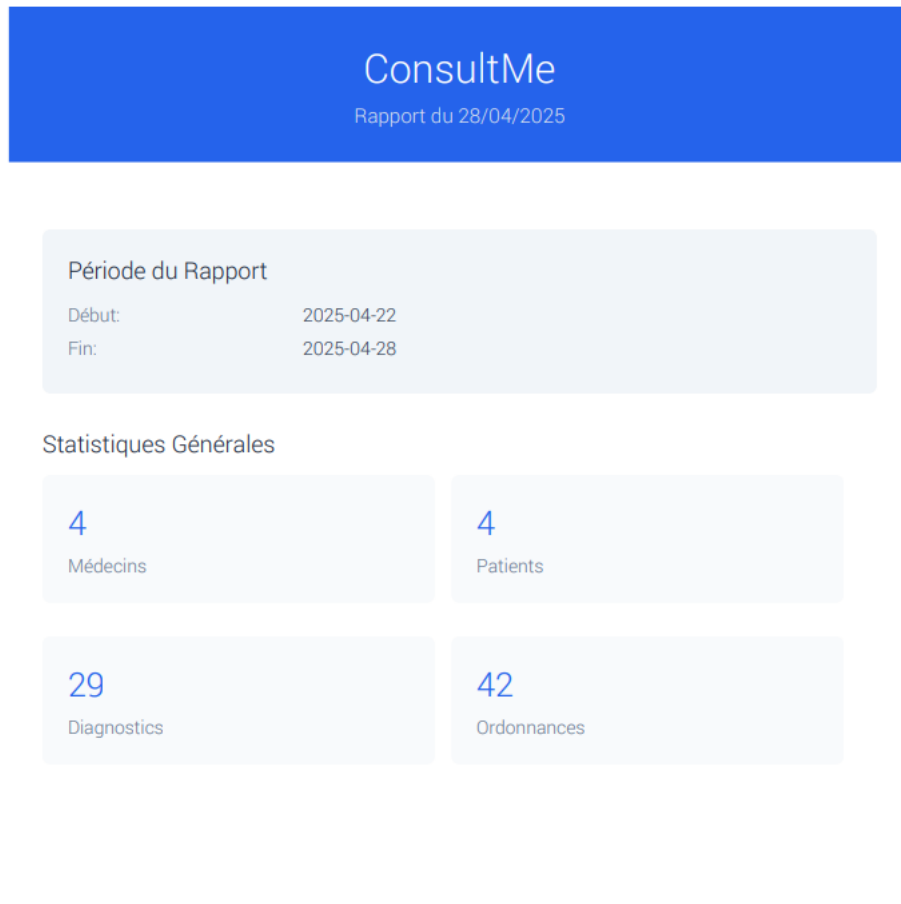
! **Docteur Cold - comprimé**
Dosage: 500mg
Posologie: 3 comprimé par jour
Durée: 3 jours

Symptômes traités:
Grippe

Ce document est authentifié électroniquement par ConsultMe et ne nécessite pas de signature physique

Figure 23 : Fiche d'ordonnance

8) Fiche de rapport



Document généré automatiquement par ConsultMe • Confidentiel

Figure 24 : Fiche de rapport