

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaïa



Faculté des Sciences Exactes
Département Informatique
Mémoire de fin d'étude

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

Option : Génie Logiciel

Thème

Conception et réalisation d'une plateforme de prise de rendez-vous médical

Réaliser par :

M^r BENNAI Brahim

M^{elle} OTMANE Ilhem

Devant le jury composé de :

Président : *M^{elle} OUYAHIA Samira* M.C.B, Université de Béjaïa.

Examinatrice : *M^{elle} ZIDANI Ferroudja* M.C.B, Université de Béjaïa.

Rapporteur : *M^{elle} REBOUH Nadjette* M.A.A, Université de Béjaïa

Année universitaire 2020/2021

Remerciements

En premier, nous aimerions remercier le bon Dieu le tout puissant de nous avoir donné le courage et la volonté de réaliser ce projet.

Nous désirons remercier nos chers parents qui nous ont soutenus et encouragé durant toute notre vie et pendant notre cursus d'étude.

Nos remerciements les plus chaleureux vont à Mme N. REBOUH pour sa disponibilité et ces précieux conseils ainsi que ces remarques qui nous ont permis d'améliorer la qualité de ce travail.

Nous tenons à exprimer toute notre grande gratitude aux membres de jury d'avoir accepté de juger ce travail.

Nos vifs remerciements s'adressent également à tous nos enseignants de la faculté sciences exactes de l'université A/MIRA Bejaia pour la formation qu'ils ont eu le soin de nous apporter le long de notre cursus universitaire.

Dédicaces

A mes adorables parents que j'aime énormément, Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi.

Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de vos prières quotidiennes.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

A mes très chers sœurs Narimane et Sylia et mes très chers frères Younes et Mehani Vous occupez une place particulière dans mon cœur. Je vous dédie ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès.

A mon cher ami M. BENMEHIDI Tu as toujours été pour moi un bon exemple Grâce à toi j'ai appris le sens du travail et de la responsabilité. Je voudrais te remercier pour ton amour, ta générosité, ta compréhension... Ton soutien fut une lumière dans tout mon parcours.

Aucune dédicace ne saurait exprimer l'amour l'estime et le respect que j'ai toujours pour toi. Pour l'aide que tu m'as apporté. Tu m'as soutenu, réconforté et encouragé. Puissent nos liens fraternels se consolider et se pérenniser encore plus.

A mes grands-parents Baya et Meriem, et Hamid que dieu les garde.

A la mémoire de mon grand-père Nacer, que dieu l'accueille dans son vaste paradis.

A tous mes cousins et cousines.

A tous mes camarades pour leurs disponibilités et leurs conseils durant la réalisation de ce travail.

A toute la promotion d'informatique 2021 spécifiquement la section Génie logiciel et tous ceux qui me connaissent.

OTMANE ILHEM

Dédicaces

À mes très chers parents ainsi qu'à mes amis...

BENNAI Brahim

Liste des abréviations

API	:Application Programming Interface
APK	:Android Package
AVD	:Android Virtual Device
DOS	:Disk Operating System
HTML	:Hyper Text Markup Language
IDE	:Integrated Development Environment
LAMP	:Linux, Apache,MySQL,PHP
MVC	:Model-View-Controller
MySQL	:My structured Query language
OS	:Operating System
OMG	:Object Management Group
OMT	:Object Modeling Technique
OOSE	:Object Oriented Software Engineering
PHP	:Hypertexte Preprocessor
QR	:Quick Response
RIM	:Research In Motion
SDK	:Software Development Kit
SPA	:Single Page Application
USB	:Universal Serial Bus
UML	:Unified Modeling Language
UP	:processus unifié

Table des matières

Liste des abréviations	8
Liste des tableaux	8
Liste des figures	10
Introduction Générale	10
1 Description textuel du projet	2
1.1 Introduction	2
1.2 Le réseau Internet	2
1.3 Les technologies Web	3
1.3.1 Site Web	3
1.3.2 Une application Web	3
1.4 L'architecture client/serveur	3
1.4.1 Les types d'architecture client/serveur	4
1.5 Le fonctionnement d'une application web	5
1.6 L'hébergement d'une application web	6
1.7 Les technologies mobiles	6
1.7.1 Définition d'une application mobile	6
1.7.2 Stratégies de développement des applications mobile	7
1.8 Les Systèmes d'exploitation	7
1.9 Motifs de choix du sujet	8
1.10 L'informatique dans le domaine médical	9
1.11 Le médecin et le patient	9
1.11.1 Définition du patient	9
1.11.2 Définition du médecin	9
1.12 Problématique	10
1.13 La gestion des rendez-vous	11

1.13.1	Les caractéristiques de la plateforme de prise de rendez-vous en ligne	11
1.13.2	Les avantages de la plateforme de prise de rendez-vous en ligne . . .	11
1.13.3	Les éléments clés de notre plateforme de prise de rendez-vous en ligne	12
1.14	Etude de l'existant	13
1.14.1	Application mobile Touri	13
1.14.2	Application mobile Tabibe	14
1.14.3	Evaluation des applications déjà présentées	15
1.15	Objectif de notre plateforme HealthClic	16
1.16	Analyse des besoins	17
1.16.1	Besoins fonctionnels	17
1.16.2	Besoins non fonctionnels	18
1.17	Conclusion	18
2	Analyse et conception	19
2.1	Introduction	19
2.2	Analyse conceptuelle	19
2.2.1	Identification des acteurs	19
2.2.2	Identification des cas d'utilisations	21
2.3	Cas d'utilisation détaillés	22
2.3.1	Diagramme de cas d'utilisation	22
2.3.2	Description des cas d'utilisations	26
2.4	Modélisation des diagrammes de séquence	38
2.4.1	Diagramme de séquence « s'authentifier »	39
2.4.2	Diagramme de séquence « Chercher un médecin »	40
2.4.3	Diagramme de séquence « Créer un compte patient »	41
2.4.4	Diagramme de séquence « Prendre un rendez-vous »	42
2.5	Modélisation du diagramme des classes	43
2.5.1	Modèle relationnel	44
2.6	Conclusion	48
3	Réalisation et fonctionnement	49
3.1	Introduction	49
3.2	Architecture du système	49
3.2.1	Le model MVC (Model-View-Controller)	50
3.3	L'environnement de développement de HealthClic	50
3.3.1	L'environnement matériel	50
3.3.2	L'environnement logiciel	50

3.3.3	Langages de programmation utilisés	51
3.3.4	Les FramesWorks utilisés	53
3.4	La charte graphique de HealthClic	54
3.5	Présentation des différentes interfaces de notre plateforme	55
3.5.1	L'organigramme des vues de notre application :	55
3.6	Présentation des interfaces de l'application web	56
3.7	Présentation des interfaces de l'application mobile	65
3.8	Conclusion	74
	Conclusion générale et perspectives	76
	Bibliographie	76

Liste des tableaux

1.1	Avantages et inconvénients des applications Touri et Tabibe	15
1.2	Besoins et Fonctionnalités de la plateforme	17
2.1	Les rôles des acteurs de la plateforme	20
2.2	Identification des cas d'utilisations	21
2.3	Description textuelle de cas d'utilisation « s'authentifier »	27
2.4	Description textuelle de cas d'utilisation « Créer un compte patient »	28
2.5	Description textuelle de cas d'utilisation « Chercher un médecin »	29
2.6	Description textuelle de cas d'utilisation « Prendre rendez-vous »	30
2.7	Description textuelle de cas d'utilisation « Définir la disponibilité du cabinet »	31
2.8	Description textuelle de cas d'utilisation « Modifier la disponibilité »	31
2.9	Description textuelle de cas d'utilisation « Modifier le statut des rendez-vous »	32
2.10	Description textuelle de cas d'utilisation « Ajouter note et ordonnance »	33
2.11	Description textuelle de cas d'utilisation « Créer un membre »	33
2.12	Description textuelle de cas d'utilisation « Modifier les informations d'un membre »	34
2.13	Description textuelle de cas d'utilisation « Supprimer un membre »	34
2.14	Description textuelle de cas d'utilisation « Créer une spécialité »	35
2.15	Description textuelle de cas d'utilisation « Modifier une spécialité »	35
2.16	Description textuelle de cas d'utilisation « Supprimer une spécialité »	36
2.17	Description textuelle de cas d'utilisation « Annuler un rendez-vous »	37
2.18	Description textuelle de cas d'utilisation « déconnecter »	38

Table des figures

1.1	Présentation du fonctionnement de l'application web	5
1.2	Les différents équipements informatiques et leurs système d'exploitation.	8
1.3	Page d'accueil de "Touri"	13
1.4	Page d'accueil "Tabibe"	14
2.1	Diagramme de cas d'utilisation globale	23
2.2	Diagramme de cas d'utilisation de l'acteur administrateur	24
2.3	Diagramme de cas d'utilisation de l'acteur médecin	25
2.4	Diagramme de cas d'utilisation de l'acteur assistant	25
2.5	Diagramme de cas d'utilisation de l'acteur patient	26
2.6	Diagramme de cas d'utilisation « s'authentifier »	27
2.7	Diagramme de séquence « s'authentifier »	39
2.8	Diagramme de séquence « Chercher un médecin »	40
2.9	Diagramme de séquence « Créer un compte patient »	41
2.10	Diagramme de séquence « Prendre un rendez-vous »	42
2.11	Diagramme de classe	43
2.12	Règle de transformation des classes.	45
2.13	Règle de transformation de l'héritage (push-down).	46
2.14	Diagramme de classe après application de la règle de transformation de l'héritage.	47
3.1	le code des couleurs utiliser dans le design de la plateforme HealthClic	54
3.2	Présentation du logo de la plateforme HealthClic	54
3.3	L'organigramme des vues de notre application	55
3.4	Présentation de l'interface d'accueil de l'application web	56
3.5	Présentation du tableau de bord de l'application web	57
3.6	Présentation de l'interface « Définir ses disponibilités »	58
3.7	Présentation de l'interface « Modifier ses disponibilités »	58
3.8	Présentation de l'interface « La liste des patients »	59
3.9	Présentation de l'interface « Ajout d'un utilisateur »	60

3.10	Présentation de l'interface « La liste des membres »	60
3.11	Présentation de l'interface « Profil d'un médecin »	61
3.12	Présentation de l'interface « Modifier un utilisateur »	61
3.13	Présentation de l'interface « Supprimer un utilisateur »	62
3.14	Présentation de l'interface « Créer une nouvelle spécialité »	62
3.15	Présentation de l'interface « Spécialités »	63
3.16	Présentation de l'interface « Démarrage »	65
3.17	Présentation des interfaces de « L'ambording »	66
3.18	Présentation de l'interface « Inscription »	67
3.19	Présentation de l'interface « Authentification »	68
3.20	Présentation de l'interface « Accueil »	69
3.21	Présentation de l'interface « Paramètres »	70
3.22	Présentation de l'interface « Chercher médecin »	71
3.23	Présentation de l'interface « Profil médecin »	72
3.24	Présentation des interfaces « Réservation de rendez-vous »	73

Introduction générale

L'informatique est un domaine d'activité scientifique, technique et industriel, grâce à elle le monde connaît une avance technologique considérable, dans tous les secteurs de l'entreprise et d'autres établissements.

De plus, l'existence d'Internet a rajouter à ces technologies, un atout par lequel le monde est connecté à travers ces outils informatiques. Surtout, les ordinateurs et les Smartphones (téléphones intelligents) qui sont incontournables, ils sont de plus en plus dotés d'une puissance avec des fonctionnalités assez évoluées. L'informatique a impacté considérablement plusieurs métiers comme celle de la médecine.

Les hôpitaux et les cliniques se digitalisent de plus en plus, confirmant ainsi la véritable révolution culturelle et technologique que vit depuis quelques années. Malheureusement, en Algérie, cette avancée notable introduite dans la médecine par la technologie peine à se faire sentir, d'où jusqu'à ce jour, la manière de gérer manuellement est encore dominante. Pour cela, nous proposons, à travers notre projet de fin de cycle, présenter dans ce mémoire ,qui consiste en la conception et réalisation d'une plateforme médicale de prise de rendez-vous en ligne.

Cette plateforme, vise essentiellement à diminuer la charge quotidienne des patients en diminuant le temps d'attente lors de la prise d'un rendez-vous. Afin d'y parvenir, nous avons organisé ce mémoire de la manière suivante :

- Le premier chapitre présente une étude préliminaire sur le domaine à étudier. Dans nous aborderons la problématique, l'objectif et l'étude des solutions existantes.
- Le deuxième chapitre sera consacré à la conception de notre plateforme utilisant le langage UML et le processus UP.
- Le troisième chapitre quant à lui porte sur l'implémentation et la réalisation de notre plateforme suivant la conception faite et aussi sur sa présentation.
- Au final, nous terminerons avec une conclusion générale qui résume notre travail et les connaissances acquises durant la réalisation de notre plateforme ainsi que les perspectives pour cette dernière.

Chapitre 1

Description textuel du projet

1.1 Introduction

Dans ce chapitre , nous allons présenter en premier lieu quelques généralités sur les technologies web et mobiles, et les principaux systèmes d'exploitation sur lesquels repose leur fonctionnement.

Nous allons, par la suite, décrire les deux acteurs principaux sur lesquels s'appuie le domaine médical et présenter les objectifs de notre application. Ce qui nous amène à identifier les capacités du système et les besoins des utilisateurs.

1.2 Le réseau Internet

Internet est connu comme étant un système mondial d'interconnexion de réseau informatique. Il utilise un ensemble standardisé de protocoles de transfert de données. Il est composé de millions de réseaux aussi bien publics, privés, universitaires, que commerciaux et gouvernementaux. Il transporte une large gamme d'information rapide et permet l'élaboration d'applications web et mobile de services variés tel que le courrier électronique et la messagerie instantanée[1].

Cette nouvelle technologie ayant été popularisée par l'apparition du World Wide Web, les deux sont parfois confondus par le public non averti. Le World Wide Web n'est pourtant que l'une des applications d'Internet.

1.3 Les technologies Web

Le développement des applications web présente certaines particularité , au niveau technique et ergonomique.

Cette spécificité nous oblige, au moment de la conception, à préconiser des méthodes de conception et des méthodes de travail dédiées à ce genre d'applications.

Dans ce qui suit, nous définirons les technologies web telles un site web, une application web, et nous présenterons d'une manière brève la différence entre les deux.

1.3.1 Site Web

Un site web est composé d'un ensemble de documents structurés, nommés pages web, stockés (hébergés) sur un ordinateur (serveur) connecté au réseau mondial[3].

Il existe deux types de site web :

1. Site statique

Il est constitué de pages web avec du contenu fixe. Chaque page est codée en HTML et affiche les mêmes informations à chaque visiteur. La modification de ces pages se fait par une intervention de l'administrateur du site web[4].

2. Site dynamique

Les pages du site, qui le composent, peuvent être modifiables par les visiteurs. De plus, entre deux visites sur un même site, le contenu de la page peut être différent sans action de l'administrateur du site Internet. Le serveur qui fait fonctionner le site utilise une technologie de Scripting (comme PHP)ainsi qu'une base de données (comme MySQL).

1.3.2 Une application Web

Une application web est un ensemble de pages, qui interagissent avec les utilisateurs, les unes avec les autres. Une application web sert à manipuler l'information et elle est soumise à des contrôles d'accès et des rôles et un site web sert à exposer l'information[5].

1.4 L'architecture client/serveur

L'architecture client/serveur désigne un mode de communications entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs postes clients du serveur ; chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique[6].

- Caractéristiques d'un serveur :
 - Il est passif (maître).
 - Il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients.
 - Dès qu'une requête lui parvient, il la traite et envoie une réponse.
- Caractéristiques d'un client :
 - Il est actif (ou esclave);
 - Il envoie des requêtes au serveur;
 - Il attend et reçoit les réponses du serveur.

Remarque : Le client et le serveur doivent bien sûr utiliser le même protocole de communication. Un serveur est généralement capable de servir plusieurs clients simultanément[7].

1.4.1 Les types d'architecture client/serveur

1. L'architecture 1-tiers

En règle générale, une application 1-tiers est découpée en trois niveaux (couches) d'abstraction :

- **La couche présentation :** Elle gère l'affichage des données et les interactions de l'application avec l'utilisateur, cette couche gère les saisies au clavier, à la souris et la présentation des informations à l'écran. Dans une application un tiers, les trois couches applicatives sont intimement liées et s'exécutent sur le même ordinateur.
- **La couche métier :** Elle correspond à la partie fonctionnelle de l'application; celle qui implémente la logique et qui décrit les opérations que l'application opère sur les données, en fonction des requêtes d'un utilisateur effectuées au travers de la couche présentation.
- **La couche accès aux données :** Elle regroupe le stockage et les mécanismes d'accès des données à ce qu'elles soient utilisables par l'application au niveau traitement.

2. L'architecture 2-tiers

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers signifiant rangée en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.

3. L'architecture 3-tiers

L'architecture a 3 niveaux (appelée architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a également une architecture partagée entre :

- Un client :c'est-à-dire que l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation.
- serveur d'application : (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur
- Le serveur de données : fournissant au serveur d'application les données dont il a besoin

4. L'architecture n-tiers

On parle d'architecture 3 tiers mais aussi d'architecture n tiers. En effet dans la plupart des applications le niveau intermédiaire est une collection de composants qui sont utilisés dans de nombreux traitements transactionnels. Ces composants peuvent être situés sur un ou plusieurs serveurs physiques[8]

1.5 Le fonctionnement d'une application web

Lorsqu'un serveur web reçoit une requête de page web statique, il transmet simplement cette page au navigateur requérant. En revanche, lorsque le serveur web reçoit une requête de page dynamique, il transmet cette page à une extension logicielle spéciale chargée d'achever la page[9]. Ce logiciel spécial est appelé serveur d'application. Ce dernier lit le code de la page, termine cette page en fonction des instructions figurantes dans le code, puis en retire le code. Il en résulte une page statique que le serveur d'application renvoie au serveur web, qui transmet alors cette page au navigateur requérant. Ce dernier reçoit uniquement du code HTML pur lorsque la page lui est transmise(voir la figure 1.1).

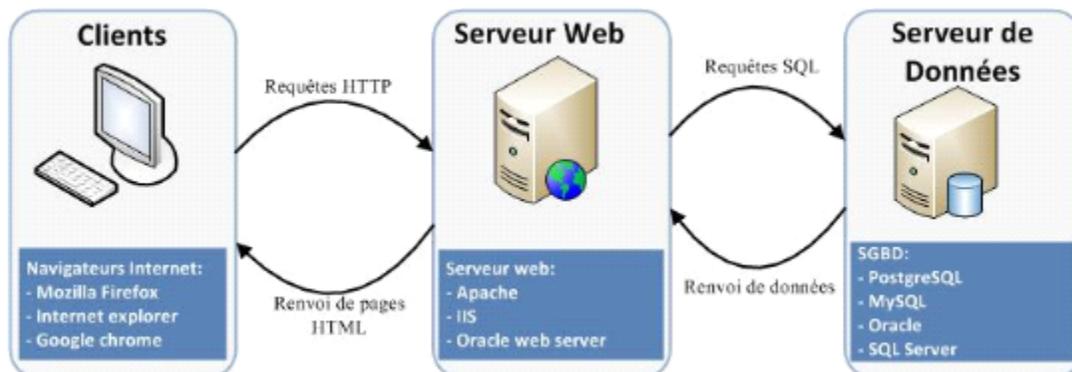


FIGURE 1.1 – Présentation du fonctionnement de l'application web

1.6 L'hébergement d'une application web

L'hébergement est un service visant à rendre un site ou une application web accessible sur Internet. Afin que les pages soient visibles par tout le monde, il faut qu'elles soient stockées sur un ordinateur connecté en permanence à l'internet (serveur).

Un hébergeur est comme une société spécialisée dans l'hébergement de sites web sur des serveurs. Sa principale activité sur internet consiste à installer des serveurs, à les sécuriser, à les tenir à jour en installant les mises à jour de sécurité pour éviter les attaques malveillantes et à les réparer en cas de panne. Il existe plusieurs types d'hébergements dont :

1. **Hébergement gratuit** : Ce type d'hébergement offre un service basique avec un espace de stockage assez limité et une fiabilité qui n'est pas toujours optimale.
2. **Hébergement payant** : Ce type d'hébergement propose un service de qualité, avec de bonnes capacités de stockage ainsi qu'une grande fiabilité. Le support sera également plus disponible et plus réactif en cas de problème et aidera avec des réponses personnalisées[10].

1.7 Les technologies mobiles

Avec le développement de la technologie mobile et l'utilisation largement répandue des téléphones portables les applications mobiles prennent une place importante dans notre quotidien, elles offrent tant de fonctionnalités qui nous facilitent grandement la vie[11].

Dans cette section, nous aborderons les notions d'applications mobiles et des systèmes d'exploitation sur lesquels repose leur fonctionnement.

1.7.1 Définition d'une application mobile

C'est un logiciel applicatif développé pour être installé sur un appareil mobile[12], généralement un téléphone mobile, un téléphone intelligent ou une tablette numérique. Les applications mobiles sont des programmes relativement légers, autonomes, utilisés pour des services de l'information, des médias sociaux, des jeux , etc.

Avec les possibilités matérielles incorporées aux terminaux mobiles (caméra, GPS, gyroscope, etc.), les applications installées sur ces derniers peuvent intégrer des fonctionnalités spécifiques et dédiées pour les utilisateurs, permettant ainsi d'enrichir leur spectre fonctionnel et imaginer des usages non couverts jusqu'à présent par les systèmes d'information tels que la géolocalisation, le scan de QR codes, etc.

1.7.2 Stratégies de développement des applications mobile

La conception d'applications mobiles peut se faire suivant trois stratégies de développement distinctes. Dans ce qui suit, nous donnons un bref aperçu de chaque stratégie[13].

1. **Application native**

Une application native est une application mobile spécifiquement développée pour un système d'exploitation mobile. Elle est conçue avec le langage et les outils associés à son système d'exploitation, et installée directement sur le mobile. Cette installation se fait, soit à travers un téléchargement via Internet, soit par un déploiement depuis un ordinateur connecté au mobile. Les tests, pour vérifier le comportement de ces applications, nécessitent des compétences techniques spécifiques et des appareils très coûteux[14]

2. **Application web**

Une application mobile web est une application développée en HTML, accessible et exécutable par le biais d'un navigateur internet pour téléphone mobile. Elle utilise le navigateur du Smartphone et ne nécessite pas forcément de télécharger l'application[15].

3. **Application hybride**

L'application hybride est une application pour mobiles qui combine des éléments HTML5 sous forme d'application mobile web et des éléments d'une application native permettant l'utilisation des fonctionnalités natives des Smartphones et d'être distribuée en tant qu'application sur les stores des systèmes mobiles (AppStore, PlayStore, etc.) .

1.8 Les Systèmes d'exploitation

En informatique, un système d'exploitation ou operating system (OS) est un groupe de programmes qui facilitent l'utilisation de l'appareil[16]. Il s'agit d'un logiciel qui reçoit des sollicitations pour employer les ressources de la machine comme le disque dur pour stocker de la mémoire, ou des périphériques pour établir une communication visuelle ou auditive[17] . Les ordinateurs portables et de bureau utilisent le système Windows, Linux ou Mac OS, tandis que, les tablettes et les Smartphone utilisent Android ou IOS, (voire la figure 1.2)[18].

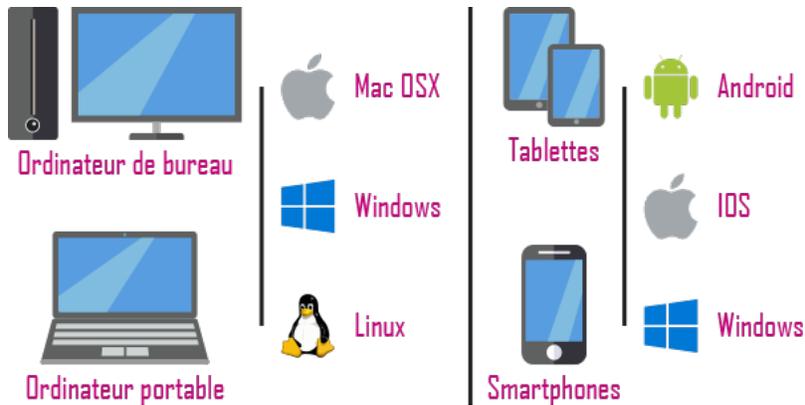


FIGURE 1.2 – Les différents équipements informatiques et leurs système d’exploitation.

1.9 Motifs de choix du sujet

La santé revêt une importance vitale pour tous les êtres humains dans le monde. Quelles que soient nos différences, la santé est notre bien le plus précieux. Une personne en mauvaise santé ne pourra pas profiter pleinement de sa vie. Telle que définit par l’Organisation Mondiale de la Santé (OMS) "la santé est un état de bien-être complet physique, mental et social, et ne consiste pas seulement en une absence de maladie ou d’infirmité". Donc la santé n’est pas seulement le fait d’être malade ou d’être atteint d’une infirmité. C’est une notion relative, ressentie par chaque individu. Aucune mesure réelle ne peut mesurer la santé, puisque la santé est le fait de satisfaire tous les besoins affectifs, nutritionnels, relationnels et sanitaires de l’individu et/ou de la population toute entière.

Il est nécessaire de préciser les motifs de choix de notre sujet qui sont les suivants :

- Notre thème est un sujet très riche en informations.
- C’est un thème d’actualité notamment dans le contexte actuel de l’Algérie.
- Ce sujet permet d’enrichir nos connaissances dans ce domaine.
- C’est un thème qui suit l’évolution de la technologie de l’informatisation des différents systèmes tel que le domaine médical.
- Pouvoir améliorer l’exercice de la médecine par les professionnels de santé, en leur apportant des outils utiles tel que la prise de rendez-vous en ligne.
- Forte volonté de donner un plus à la population algérienne ainsi qu’au marché algérien qui y’est quasiment vierge surtout dans le domaine médical.

1.10 L'informatique dans le domaine médical

L'informatique médicale est devenue au fil des années une vraie discipline scientifique dont les bases et applications sont enseignées non seulement dans tous les domaines de santé (médecine, odontologie, pharmacie, maïeutique, sciences sanitaires et sociales, école de soins infirmiers et écoles de santé publique, etc), mais également dans de nombreux autres cursus (Sciences de la vie, Écoles d'ingénieur et d'économie, etc)

Par l'informatique, les connaissances les plus ésotériques deviennent accessibles à tous. Leur transcription numérique obligée les rattache toutes à un langage commun. Par cette voie, elles deviennent familières à tous. Chacun peut ainsi tirer parti de l'immense capacité de mémorisation de l'ordinateur[19].

L'informatique offre, de plus, la possibilité d'explorer des procédures logiques innombrables dans un temps presque virtuel[20].

L'extension de la capacité de mémorisation, l'accélération des procédures logiques font de l'informatique l'instrument d'une mutation profonde de la démarche médicale et avec elle de l'ensemble du système de santé.

1.11 Le médecin et le patient

1.11.1 Définition du patient

Le mot patient signifie « celui qui endure » ou « celui qui souffre ». C'est une personne qui bénéficie d'examens médicaux, de traitements prodigués par un médecin ou un professionnel de la santé pour faire face à une maladie ou à des blessures. Il peut également bénéficier d'actes de prévention[21].

1.11.2 Définition du médecin

Un médecin est un professionnel de la santé titulaire d'un diplôme de docteur en médecine. Il accueille des patients de tout âge, il soigne selon sa spécialisation des maladies, pathologies et blessures. Cela lui demande donc des connaissances médicales très étendues. Il travaille, généralement, au sein d'une équipe de professionnels de la santé comme le psychologue, le diététicien, le pharmacien, le physicien médical, l'infirmier ou le chirurgien-dentiste[22].

Le médecin généraliste possède une vision complète de son patient. Il les connaît, connaît leurs habitudes, leur hygiène de vie, leurs antécédents familiaux, etc.

1.12 Problématique

Malheureusement, dans le secteur médical algérien à l'heure actuelle, il n'existe pas de système fonctionnel de gestion des rendez-vous. Ce qui conduit à un non-respect et à un réel suivi des horaires prises par chaque patient qui passe des heures devant le cabinet pour pouvoir le prendre, et d'autre passe beaucoup de temps au téléphone en attendant que leurs noms s'ajoutent à la liste des consultations, par contre d'autre lâche l'affaire après plusieurs tentatives parce que le téléphone est toujours occupé.

L'impact de cette mauvaise gestion se traduit par le fait que le patient n'a aucune visibilité sur son heure de consultation, et ainsi ne peut organiser efficacement les autres engagements de sa journée. En effet, dans les établissements de santé, il est possible de prendre un rendez-vous pour une consultation aussi bien avec un médecin généraliste, qu'avec un spécialiste. La prise de rendez-vous se fait actuellement soit en appelant par téléphone soit en se déplaçant vers l'établissement. Les appels téléphoniques ne sont pas toujours concluants parce qu'il n'est pas aisé d'avoir le contact d'un centre de santé ou d'un médecin et les appels ne sont reçus qu'aux horaires d'ouverture du secrétariat du cabinet médical.

De plus, il n'est pas rare de constater que le patient ou le médecin ne soit pas en mesure d'honorer un rendez-vous préalablement validé. Dans ces cas de figure, la seconde partie n'est, généralement, pas tenue informée à temps de l'annulation du rendez-vous.

Il arrive aussi que le patient oublie sa date de consultation, lorsque ce n'est pas sa première consultation dans cet établissement on lui demande la date de sa dernière consultation ; il ne s'en rappelle pas toujours.

Ceci implique l'impossibilité d'un suivi efficace du patient sur une longue durée. En effet, ce siècle marqué par la technologie où les pays recherchent une numérisation totale des activités dans tous les secteurs, d'où une informatisation du système de prise de rendez-vous aux algériens est toujours nécessaire.

Nous proposons, pour cela, une plate-forme "HealthClic", qui sera disponible sous forme d'une application web interactive, fiable, conviviale et facile à utiliser qui sera destinée aux professionnels de santé pour leurs permettre de gérer les rendez-vous dans leurs cabinets médicaux et une application mobile pour les patients afin de pouvoir réserver un rendez-vous en ligne pour une consultation et cela à tout moment et de n'importe quel emplacement.

1.13 La gestion des rendez-vous

Ce mémoire a pour objectif de nous guider dans la réalisation d'une solution de prise de rendez-vous en ligne qui correspond le mieux aux besoins des médecins et des patients. Une plateforme qui permet aux patients de prendre des rendez-vous facilement au moment opportun. Ils peuvent choisir la date et l'heure qui leur conviennent, recevoir une notification de confirmation et sans devoir interagir avec le personnel.

L'adoption de la prise de rendez-vous en ligne est un sujet d'actualité et prend de plus en plus d'ampleur. L'utilisation d'un système de prise de rendez-vous en ligne peut remédier aux problèmes suivants :

- Les patients qui se plaignent de l'attente au téléphone pour prendre un rendez-vous.
- Le secrétariat est, constamment, sollicité par la clientèle.
- Il y a parfois des erreurs dans les plages des horaires.
- La difficulté à atteindre ou maintenir un taux d'assiduité de 80 % et plus.

1.13.1 Les caractéristiques de la plateforme de prise de rendez-vous en ligne

- La prise de rendez-vous en tout temps.
- Un accès sécurisé.
- Des rappels automatisés du rendez-vous.
- Des identifiants individuels pour le personnel.
- Des règles permettant de personnaliser la prise de rendez-vous.
- Des avis d'annulation et des options de modification automatisés.
- La création d'un historique des rendez-vous pour chaque patient.

1.13.2 Les avantages de la plateforme de prise de rendez-vous en ligne

1. Réduction du temps consacré à la prise de rendez-vous

- Moins d'appels entrants, de confirmations et de rappels téléphoniques à gérer contribuent à une gestion plus efficace du temps pour le personnel.
- Les notifications et les rappels automatisés de la prise de rendez-vous en ligne ont grandement contribué à réduire le nombre de rendez-vous manqués et les annulations de la dernière minute augmentent le taux d'assiduité des patients.

2. **Augmentation de la satisfaction du personnel**

Une solution en ligne élimine les tâches complexes et les délais liés à la priorisation des patients au moment de la prise de rendez-vous. Ce qui offre une amélioration au niveau de la satisfaction de leur personnel.

3. **Augmentation de la satisfaction des patients**

La plupart des patients préfèrent utiliser la prise de rendez-vous en ligne plutôt que d'appeler leur clinique et risquer de patienter en ligne.

1.13.3 **Les éléments clés de notre plateforme de prise de rendez-vous en ligne**

1. **Automatisation des confirmations et rappels de rendez-vous**

La solution de prise de rendez-vous en ligne pourrait se configurer de façon à envoyer automatiquement une confirmation du rendez-vous au patient. De plus, le système devrait être en mesure d'envoyer au moins un rappel dans le jour précédant le rendez-vous. Cette fonctionnalité contribue à réduire le taux de rendez-vous manqués[23].

2. **Synchronisation des horaires des patients et des médecins**

Le système doit synchroniser tous les horaires du cabinet médical en un seul endroit. Certes, cette solution offre un portail de rendez-vous pour les patients, mais lorsque les membres du personnel reçoivent une demande de rendez-vous par téléphone, ils doivent l'inscrire manuellement dans le calendrier du médecin. Ce qui fait que la solution est incomplète et répond autant aux besoins du personnel[24].

Il y a souvent plusieurs motifs de consultation qui sont offerts aux patients. Le système devrait permettre le choix parmi plusieurs types de motifs de rendez-vous, d'une durée prédéterminée, avec différents professionnels.

3. **Règles de planification personnalisables**

La solution est personnalisable selon les besoins en termes de création et de gestion des horaires. Les médecins produisent souvent leurs horaires plusieurs mois à l'avance, mais il est possible d'offrir des plages horaires seulement quelques semaines à l'avance aux patients.

4. **Considérations pour l'adoption de la plateforme**

Le système de rendez-vous en ligne s'adapte pour répondre aux besoins des différents types de pratique. Il existe de nombreuses options technologiques disponibles et celles-ci sont toutes accessibles sur le web et ne nécessitent aucune installation de la part du médecin.

1.14 Etude de l'existant

Pour mettre sur pied une plateforme efficace, il est nécessaire de réaliser une étude concernant les méthodes et les stratégies utilisées par nos concurrents pour pouvoir les devancer, et atteindre les besoins des utilisateurs. De ce fait, nous avons effectué des recherches sur le marché pour avoir une idée à propos de notre secteur en ligne. Notre recherche nous a mené à quelques solutions, pratiquement deux présentées par la suite, pour la gestion des prises de rendez-vous mais qui sont peu fonctionnelles. Il existe bien d'autres applications mobiles dans le domaine médical en Algérie qui ne seront pas mentionnées dans ce travail parce qu'elles n'offrent pas le service de prise de rendez-vous médical. Par contre aucune application web de gestion des rendez-vous n'a été conçue pour permettre aux médecins en Algérie de bien gérer leurs rendez-vous en cabinets.

1.14.1 Application mobile Touri

Touri Comme le montre la figure (1.3), est une application de prise de rendez-vous développée à la wilaya d'Adrar en Décembre 2019. Elle offre deux types de réservations de rendez-vous en ligne :

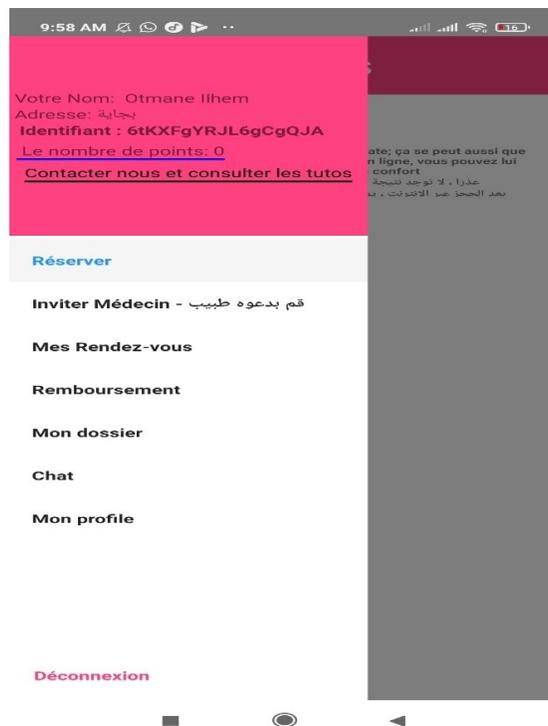


FIGURE 1.3 – Page d'accueil de "Touri"

1. **Réservation sans avancement :**

La prise des rendez-vous en ligne est gratuite mais fonctionnelle seulement la nuit, à partir de 22h.

2. **Réservation avec avancement :**

Une réservation fixée avec un tarif de 500 dinars algériens. Ce qui offre les privilèges d'une réservation différée (à n'importe quelle heure) et la possibilité d'être alerté et informé de l'approche de son tour à distance via son téléphone intelligent, sans être dans l'obligation de surveiller son tour. Des privilèges qui ne sont pas disponibles dans la réservation sans avancement.

1.14.2 Application mobile Tabibe

Tabibe est une solution de mise en relation entre les patients et les professionnels de santé en Algérie. Elle offre l'opportunité de prendre des rendez-vous chez des médecins spécialistes ou des laboratoires d'analyses médicales(voir figure 1.4)[25].

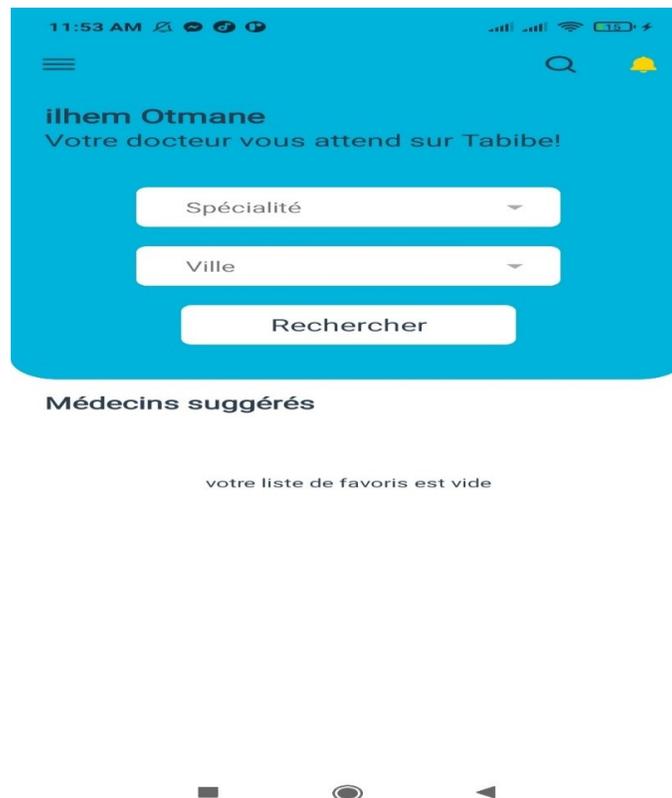


FIGURE 1.4 – Page d'accueil "Tabibe"

1.14.3 Evaluation des applications déjà présentées

Nom de l'application	Avantage	Inconvénient
Touri	<ul style="list-style-type: none"> -Interface intuitive,ergonomique -Une bonne initiative qui donne un plus à la population algérienne ainsi qu'au marché vierge surtout dans le domaine médical 	<ul style="list-style-type: none"> -Des bugs lors de l'inscription sur l'application qui renvoie aux utilisateurs, à chaque fois, l'erreur du mot de passe. -Manque d'affichage des créneaux disponibles pour chaque médecin pour pouvoir réserver. -Impossibilité de réserver un rendez-vous à n'importe quel moment -Manque des informations sur la disponibilité des médecins (les durées de leur congés)
Tabibe	<ul style="list-style-type: none"> -Une application ergonomique facile à utiliser. -Elle affiche des informations détaillées sur les médecins surtout le paiement et le prix de chaque consultation 	<ul style="list-style-type: none"> -mise à jour qui engendrent beaucoup de bugs. -Une application qui ne recouvre pas le territoire algérien en sa totalité -Un temps de chargement marquant chaque fonctionnalité -Manque de confirmation des rendez-vous par e-mail

TABLE 1.1 – Avantages et inconvénients des applications Touri et Tabibe

1.15 Objectif de notre plateforme HealthClic

Suite à l'étude et l'évaluation faites sur les solutions déjà proposées dans le domaine médical de prise de rendez-vous en ligne, nous proposons dans ce travail de concevoir et implémenter une nouvelle plateforme nommée "**HealthClic**" dont l'objectif est de satisfaire les besoins des clients et du personnel médical à propos des prises de rendez-vous.

Cette plateforme vise, principalement, à offrir une meilleure expérience utilisateurs aux praticiens ainsi qu'aux patients et cela en réalisant les objectifs suivants :

- Donner plus d'importance à l'interface homme-machine et simplifier au maximum la gestion des rendez-vous de l'application par la conception d'une interface ergonomique, intuitive et très facile à utiliser qui sera à la portée du large public.
- Améliorer la qualité de service rendu en matière de temps de réponse.
- Faciliter la recherche des praticiens en ajoutant une fonctionnalité qui permet de trier la recherche par le nom du médecin ou par le nom de sa spécialité ou le lieu du travail.
- Maintenir des rappels à travers un Email envoyé la veille d'un rendez-vous.
- Consolider un historique des rendez-vous pour faciliter la gestion des patients dans le cabinet médical.
- Couvrir le maximum du territoire Algérien pour faire profiter le maximum de gens et de cabinets médicaux de cette opportunité.
- Renforcer les informations de chaque praticien en apportant le maximum de détails qui seront consulter par les utilisateurs. Cependant, ils pourront réserver des rendez-vous en toute sécurité.
- Offrir une bonne gestion des profils utilisateurs.
- Pouvoir localiser l'adresse des différents cabinets médicaux sur la carte géographique.
- Bénéficier les praticiens d'un agenda médical qui assure une bonne gestion de rendez-vous.

1.16 Analyse des besoins

1.16.1 Besoins fonctionnels

Besoins	Fonctionnalités
Authentification	Créer une interface avec les champs nécessaires
Créer un compte	Créer un formulaire d'inscription
Gérer un compte	Créer un espace d'inscription
Modifier le profil	Donner à l'admin la possibilité de changer les informations des médecins et des assistants
Consulter un rendez-vous	Une option qui sera disponible dans l'agenda
Chercher un médecin	Avoir une rebrique "rechercher" qui permet au patients d'effectuer une recherche
Ajouter des notes et des ordonnances	Ajouter une option aux médecins qui permet d'ajouter des observations et des prescription
Recevoir des rappelles	Générer automatiquement des rappels et des notifications
Consulter des interface de disponibilité	Avoir un espace qui permet aux assistants et aux médecins de gérer leur rendez-vous du cabinet
Définir les horaires de travail	Avoir un espace ou on sélectionne les horaires ou le médecin travaille
Envoyer des emails de confirmation lors de la prise de rendez-vous	Générer automatiquement par le système
Gérer nos membres	Créer un espace accessible par l'administrateur ou il pourra modifier ou supprimer les comptes des médecins
Consulter la localisation d'un cabinet médical	Intégrer la carte géographique et la position de chaque cabinet sur Google maps
Gérer les spécialités	Permet à l'administrateur d'ajouter, modifier ou supprimer une spécialité
Déconnexion	Ajouter une option qui permet à l'utilisateur de se déconnecter

TABLE 1.2: Besoins et Fonctionnalités de la plateforme

1.16.2 Besoins non fonctionnels

Ces besoins sont soit des besoins optionnels, soit des besoins/contraintes liés à l'implémentation. Ainsi, la plateforme HealthClic doit être :

- Sécurisée
- Dotée d'une bonne expérience utilisateur.
- Facile à utiliser
- Performante

Quant à l'application web, elle devra être :

- **Sécurisée** : L'utilisation de Bootstrap est recommandée pour l'adaptation à tout type de support (mobile, tablette et ordinateur).
- **Responsive** : Facile à utiliser
- **Disponibilité** : Disponible (24h/24h) et (7/7) et depuis n'importe où dans le monde (sans VPN).
- **Bonne ergonomie** : Il faut se fixer sur l'aspect esthétique tout en respectant les contraintes de la charte graphique.

1.17 Conclusion

Dans ce chapitre, nous avons présenté en premier lieu quelques définitions sur les technologies web et mobiles, et les principaux systèmes d'exploitation sur lesquels repose leur fonctionnement. Ensuite, nous avons décrit les acteurs principaux sur lesquels s'appuie le domaine médical et présenter les objectifs de notre application. Enfin nous avons déterminé les besoins fonctionnels et non fonctionnels de notre plateforme.

Dans le prochain chapitre, nous allons décrire de façon détaillée les cas d'utilisation. Pour cela, nous allons définir de manière textuelle toutes les interactions entre les acteurs et le système, concevoir le diagramme de cas d'utilisation et les diagrammes de séquence et enfin le diagramme de classe et le modèle relationnel.

Chapitre 2

Analyse et conception

2.1 Introduction

Nous allons présenter dans ce chapitre les différents acteurs qui interagissent avec le système suivis de la présentation des cas d'utilisation, les diagrammes de séquence, le diagramme de classe. Nous concluons par l'élaboration du modèle logique qui décrit l'implémentation de la base de données.

2.2 Analyse conceptuelle

2.2.1 Identification des acteurs

En UML, un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système. Il existe deux types d'acteurs, à savoir les acteurs principaux, qui utilisent directement les fonctionnalités du système et les acteurs secondaires qui contribuent à la réalisation d'un cas d'utilisation. Les principaux profils qui vont utiliser le système sont les suivants :

Numéro	Acteur	Rôle
	Médecin	Le « médecin », via l'application, pourra bénéficier d'un compte propre à lui qui sera créé par l'administrateur de l'application ou il pourra ajouter ou modifier ses heures de travail le médecin pourra écrire des notes, et même une prescription pour un rendez-vous réservé par un patient
		Pour pouvoir accéder aux différentes fonctionnalités de l'application, le patient doit se connecter s'il possède déjà un

2	Patient	<p>compte, sinon il doit en créer un.</p> <p>Un patient peut chercher un médecin directement à partir de la carte de localisation (Google maps) où chaque médecin enregistré dans le système a un marqueur, de plus il peut chercher un médecin par son nom ou par sa spécialité.</p> <p>Une fois le médecin a été sélectionné, le patient peut prendre un rendez-vous selon la disponibilité du médecin.</p> <p>Après avoir pris un rendez-vous, le patient peut consulter la liste de ses rendez-vous. Tant que le rendez-vous ne dépasse pas un délai bien précis, le patient a la possibilité de l'annuler.</p> <p>Le patient peut aussi changer les informations de son compte.</p> <p>Si le patient rate la consultation plusieurs fois, son compte sera supprimé.</p>
3	Assistant	<p>La présence de l'acteur « assistant » dans notre système a pour but d'alléger les tâches du médecin ,tout en veillant à respecter le secret professionnel , qui est un pilier fondamentale de la relation médecin/patient</p> <p>Pour cela l'assistant aura le rôle d'ajouter, modifier ou supprimer des créneaux selon la disponibilité du médecin.</p> <p>Aussi, c'est à l'assistant de mettre le statut d'un rendez-vous sur visité/non-visité</p>
4	Administrateur	<p>L'administrateur sera l'acteur qui aura le plus de privilèges et de tâches sur notre application.</p> <p>L'administrateur aura la responsabilité d'ajouter des comptes médecins et assistants sur notre application , et de saisir les informations qui apparaîtront dans leur profil ainsi qu'un mot de passe</p> <p>L'administrateur aura aussi la tâche de gérer les spécialités (Ajout, modification ou suppression)</p>

TABLE 2.1: Les rôles des acteurs de la plateforme

2.2.2 Identification des cas d'utilisations

Nous avons déduit les cas d'utilisations qui sont résumés dans le tableau ci-dessous :

N°	Cas d'utilisation		Acteurs
01	S'authentifier		Médecin, Administrateur , Patient, Assistant
02	Déconnecter		
03	Définir la disponibilité du cabinet	Ajouter des créneaux pour une date donnée	Assistant, Médecin
04	Consulter la liste des disponibilités définis	Modifier une disponibilité	Assistant, Médecin
05	Consulter la liste des rendez-vous	Modifier le statut des rendez-vous réservés	Assistant
06	Créer un membre		Admin
07	Consulter la liste membres	Consulter le profil d'un membres	
		Modifier les informations d'un membre	
		Supprimer un membre	
08	Créer une spécialité		
09	Consulter la liste des spécialités	Modifier une spécialité	
		Supprimer une spécialité	
10	Consulter la liste des rdv réservés	Ajouter notes et ordonnance	Médecin
11	Consulter la liste des notes et ordonnances	Consulter les notes et ordonnances d'un patient	
12	Créer un compte patient		Patient
13	Chercher un médecin	Chercher par nom	
		Chercher par spécialité	
14	Prendre un rendez-vous	Ajouter un motif de consultation	
15	Choisir un horaire	Choisir une date	
		Choisir une heure	
16	Consulter les profils des médecins	Voir la position du profil sur la carte de localisation	
17	Modifier les informations du compte		
18	Consulter les rendez-vous	Voir l'historique des rendez-vous	
		Annulez un rendez-vous	

TABLE 2.2: Identification des cas d'utilisations

2.3 Cas d'utilisation détaillés

2.3.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un graphe d'acteurs, qui englobés par la limite du système, des relations de communication entre les acteurs et les cas d'utilisation, et des généralisations de ces cas d'utilisation.

-DIAGRAMME DE CAS D'UTILISATION « GLOBAL »

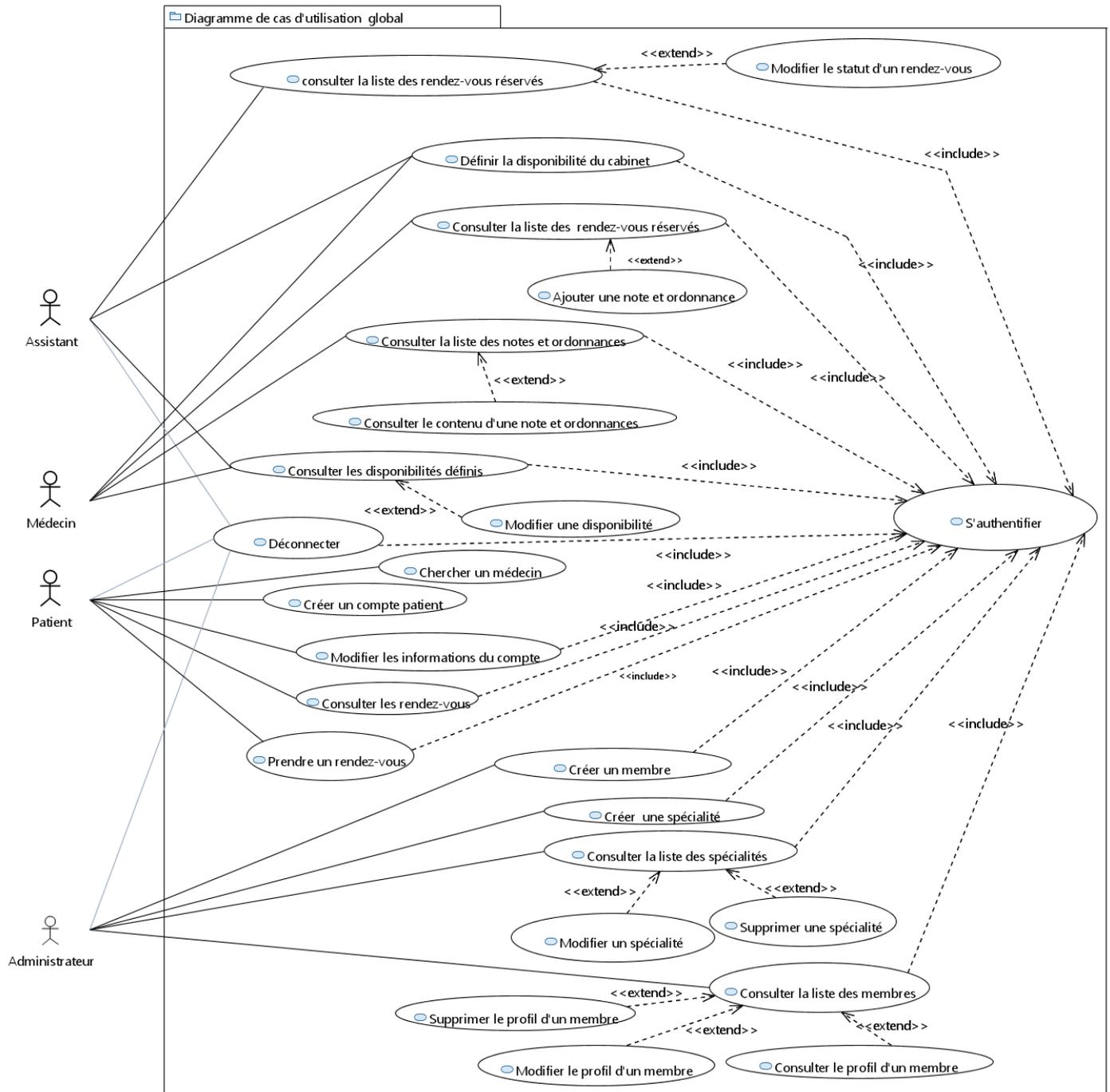


FIGURE 2.1 – Diagramme de cas d'utilisation globale

-DIAGRAMME DE CAS D'UTILISATION DE L'ACTEUR « ADMINISTRATEUR »

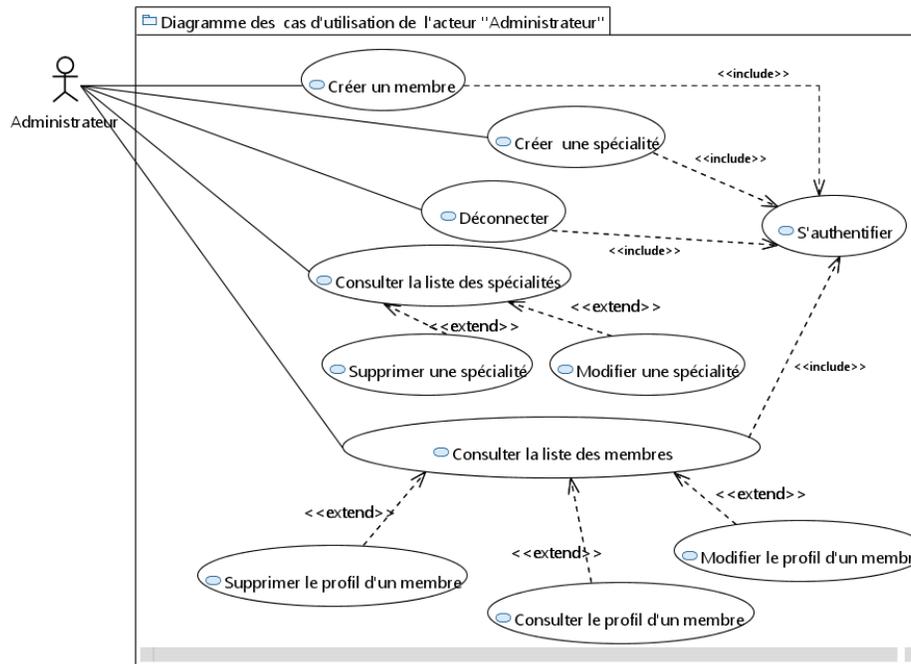


FIGURE 2.2 – Diagramme de cas d'utilisation de l'acteur administrateur

-DIAGRAMME DE CAS D'UTILISATION DE L'ACTEUR « MEDECIN »

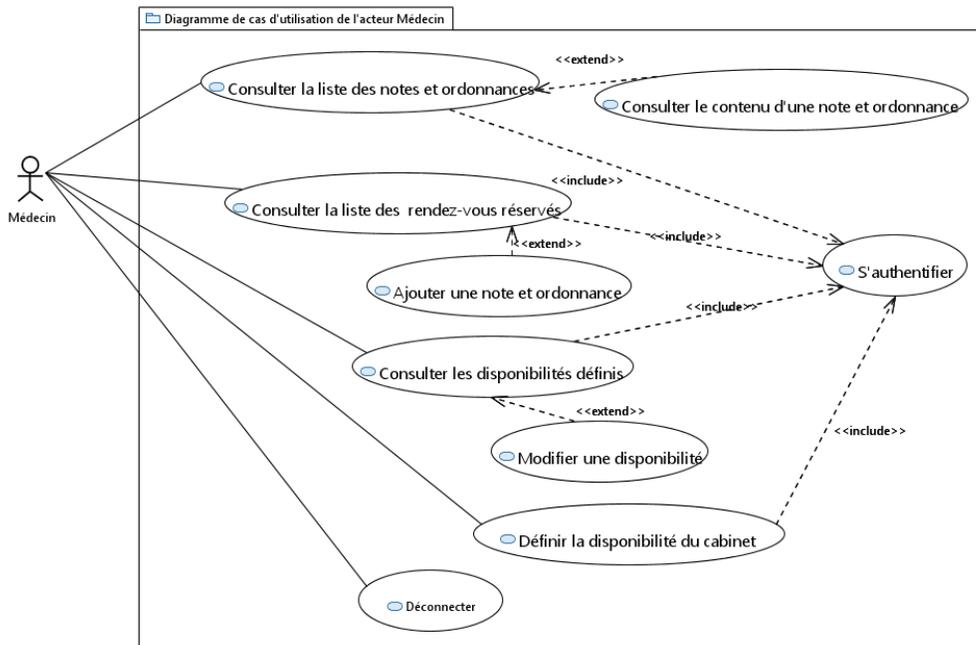


FIGURE 2.3 – Diagramme de cas d'utilisation de l'acteur médecin

-DIAGRAMME DE CAS D'UTILISATION DE L'ACTEUR « ASSISTANT »

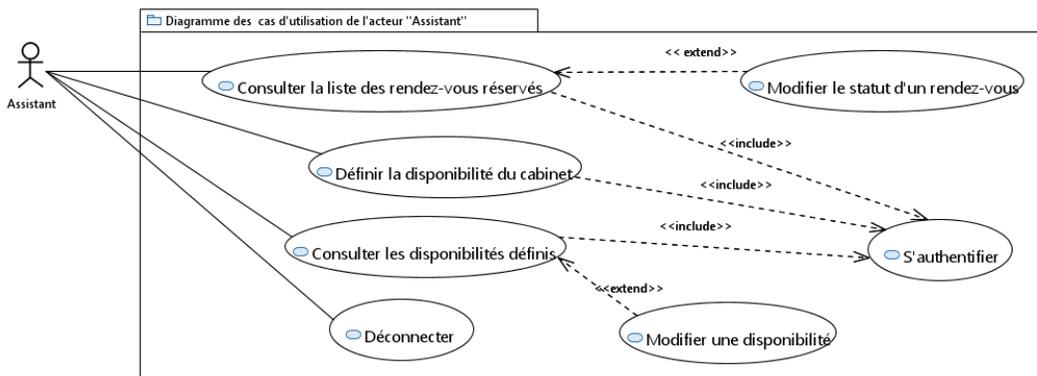


FIGURE 2.4 – Diagramme de cas d'utilisation de l'acteur assistant

-DIAGRAMME DE CAS D'UTILISATION DE L'ACTEUR « PATIENT »

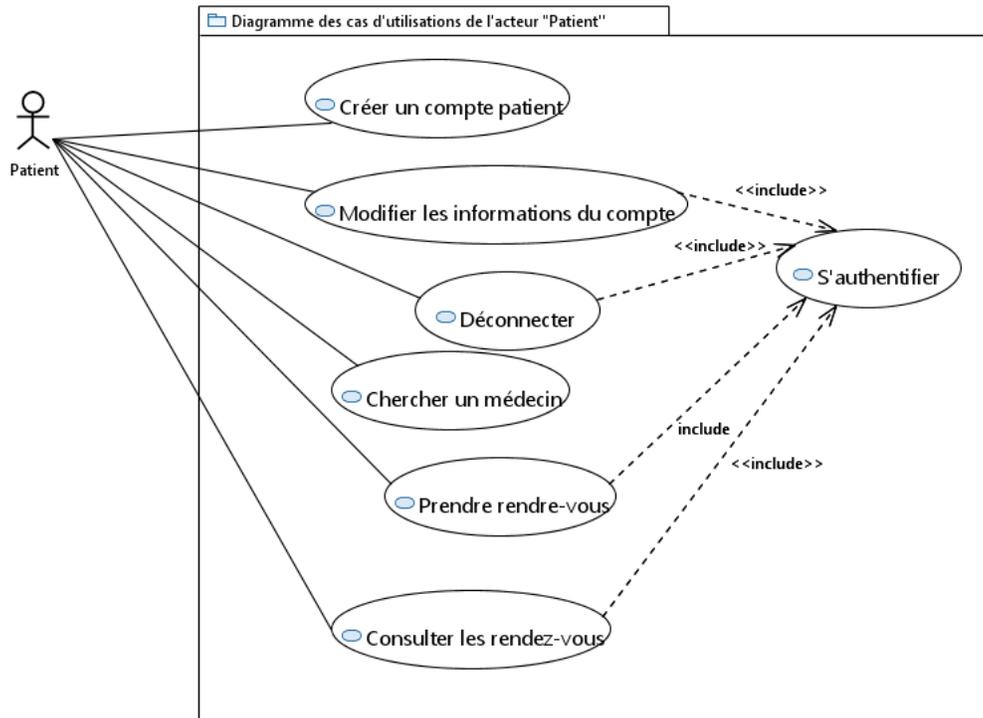


FIGURE 2.5 – Diagramme de cas d'utilisation de l'acteur patient

2.3.2 Description des cas d'utilisations

-Description textuelle de cas d'utilisation « S'authentifier »

C'est un processus par lequel le système s'assure de l'identité de l'utilisateur, il doit s'authentifier par un login et un mot de passe. Après avoir visité notre site, le patient doit remplir un formulaire. Ce formulaire nous servira à l'enregistrer sur notre base de données et de le contacter en cas de besoin (modification, annulation, ets). L'authentification se fait qu'une seule fois. Le patient recevra un identifiant et un mot de passe. La prochaine connexion sur le site sera prise en compte avec son identifiant.

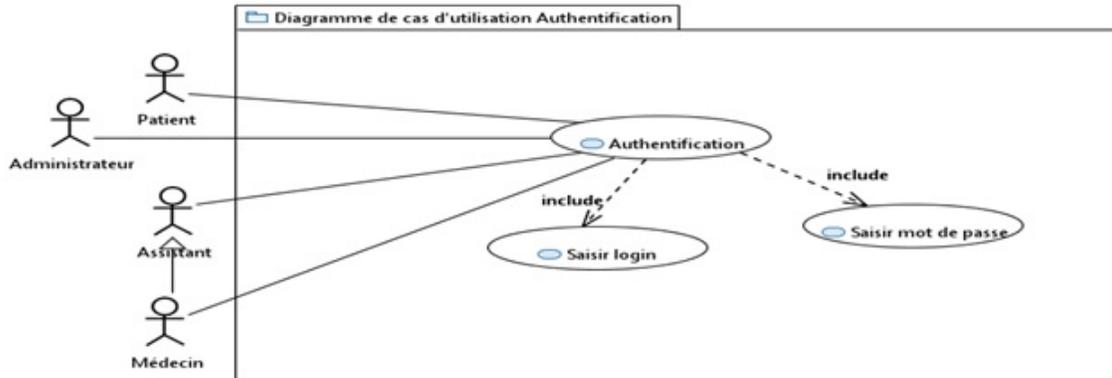


FIGURE 2.6 – Diagramme de cas d'utilisation « s'authentifier »

Cas d'utilisation	S'authentifier
Acteur	Administrateur, Médecin Patient, Assistant
Objectif	Permet à l'acteur d'accéder à son propre espace.
Précondition	* Être connecté à internet. * L'acteur doit avoir un compte
Scénario nominal	1. Le système invite l'utilisateur à entrer son login et son mot de passe 2. L'acteur introduit son identifiant et son mot de passe. 3. Le système vérifie l'existence du compte. 4. Le système donne l'accès à l'interface correspondante.
Scénario alternatif	Si l'email ou le mot de passe sont invalides, le système affiche un message d'erreur. « Votre identifiant ou votre mot de passe est incorrect ».

TABLE 2.3 – Description textuelle de cas d'utilisation « s'authentifier »

-Description textuelle de cas d'utilisation « Créer un compte patient »

Cas d'utilisation	Créer un compte patient (application android)
Acteur	Patient
Objectif	Permet au patient de créer un compte sur l'application mobile. * Être connecté à internet.
Précondition	* Télécharger et installer l'application
Scénario nominal	<ol style="list-style-type: none"> 1. Le patient choisit de créer un compte. 2. Le système affiche le formulaire correspondant. 3. Le patient remplit le formulaire. 4. Le système vérifie les informations saisies. 5. Si les données sont valides, le système enregistre les informations dans la base de données. 6. Le patient sera redirigé à l'interface d'accueil.
Scénario alternatif	<p>Si un champ lui manque la saisie ou le patient saisit des données erronées, le système affiche un message d'erreur au patient.</p> <p>Le patient sera redirigé à l'interface de création du compte.</p>

TABLE 2.4 – Description textuelle de cas d'utilisation « Créer un compte patient »

-Description textuelle de cas d'utilisation « Chercher un médecin »

Le patient peut chercher un médecin par son nom, sa spécialité ou par sa localisation sur la carte. La localisation des médecins sur la carte est disponible en deux modes : terrain et satellite. Le patient peut choisir celui qui lui convient le mieux, c'est une fonctionnalité gérée par l'API de Google adaptée pour un usage facile et rapide. Le patient peut choisir un médecin directement en cliquant sur son marqueur sur la carte. Pour chercher un médecin par spécialité, il suffit de choisir une des spécialités existantes dans la base de données et le système affiche la liste des médecins de cette spécialité. En plus, le patient a la possibilité de chercher un médecin par son nom. Pour cela, il suffit de saisir un nom et le système affiche la liste des médecins portant ce nom.

Cas d'utilisation	Chercher un médecin	
Acteur	Patient	
Description	Permet au patient d'effectuer une recherche des médecins	
Précondition	Être connecté à internet	
Scénario nominal	1. Le patient lance l'application.	
	Possibilité 1 : Le patient choisit un médecin sur la carte	1. Le patient choisit un médecin sur la carte 2. Le système affiche le profil du médecin
	Possibilité 2 : Le patient connaît son médecin.	1. Le patient arrive sur l'interface principale et saisit le nom de son médecin dans la barre de recherche. 2. Le système lui affiche une liste de tous les spécialistes de santé qui porte le nom saisi dans la barre de recherche.
	Possibilité 3 : le patient ne connaît pas de médecin en particulier et cherche ainsi un médecin par spécialité.	1. Le patient arrive sur l'interface principale et saisi le nom d'une spécialité de la médecine dans la barre de recherche. 2. Le système lui affiche une liste des noms des médecins pour la spécialité concernée.
	3. Le patient sélectionne clique sur 4. Le système renvoie d'authentification.	son médecin et consulte son profil et « prendre rendez-vous » l'utilisateur sur l'interface
Scénario alternatif	Authentification échouée	

TABLE 2.5 – Description textuelle de cas d'utilisation « Chercher un médecin »

-Description textuelle de cas d'utilisation « Prendre un rendez-vous »

Cas d'utilisation	Prendre rendez-vous
Acteur	Patient
Description	Permet au patient de prendre un rendez-vous avec un médecin pour une consultation.
Précondition	Être connecté à internet.
Scénario nominal	<ol style="list-style-type: none"> 1. Le patient lance l'application 2. Le patient consulte le profil du médecin et clique sur « prendre un rendez-vous » 3. Le système renvoie l'utilisateur sur l'interface d'authentification. 4. Le patient s'authentifie. 5. Le système affiche le profil du médecin avec les dates et les heures des rendez-vous disponible pour la consultation. 6. Le patient choisit un motif de consultation ainsi que la date et l'heure. 7. Le système affiche un message de confirmation de rendez-vous. 8. Le patient valide par un « OUI » 9. Le système lui envoie sur une interface récapitulant les informations relatives aux rendez-vous. 11. Le patient appuie sur le bouton Terminer pour revenir à l'accueil.
Scénario alternatif	Authentification échouée

TABLE 2.6 – Description textuelle de cas d'utilisation « Prendre rendez-vous »

-Description textuelle de cas d'utilisation « Définir la disponibilité du cabinet »

Cas d'utilisation	Définir la disponibilité du cabinet
Acteur	Médecin, Assistant
Objectif	Permet aux médecins et aux assistants de définir les créneaux ou ils seront disponible pour une date spécifique
Précondition	Authentification.
Scénario nominal	<ol style="list-style-type: none"> 1. Accéder à la page « Définir Mes disponibilités » 2. L'interface se charge sur le navigateur 3. L'acteur choisi la date, puis coche les créneaux où il sera disponible 4. Le système sauvegarde les créneaux sur la base de donné
Scénario alternatif	La date est déjà prise

TABLE 2.7 – Description textuelle de cas d'utilisation « Définir la disponibilité du cabinet »

-Description textuelle de cas d'utilisation « Modifier la disponibilité »

Cas d'utilisation	Modifier la disponibilité
Acteur	Médecin, Assistant
Objectif	Permet d'ajouter ou annuler un créneau pour une date donné.
Précondition	<ol style="list-style-type: none"> 1. S'authentifier 2. Accéder à « Mes disponibilité »
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur choisi la date et Clique sur « Modifier » 2. Le système affiche la page avec les créneaux déjà enregistrés. 3. L'acteur ajoute un ou plusieurs créneaux en le(s) cochant les . créneaux, ou le(s) annule en décochant les créneaux, puis il clique sur confirmer 4. Le système vérifie les champs et effectue une mise à jour et enregistre les données.

TABLE 2.8 – Description textuelle de cas d'utilisation « Modifier la disponibilité »

-Description textuelle de cas d'utilisation « Modifier le statut des rendez-vous »

Cas d'utilisation	Modifier le statut des rendez-vous
Acteur	Assistant
Objectif	Permet de noter la présence ou l'absence des patients ayant réserver un rendez-vous , avec possibilité de filtrer par date
Précondition	S'authentifier Accéder à Lister tous les rendez-vous réservés
Scénario nominal	<p>Possibilité 1 :</p> <ol style="list-style-type: none"> 1- L'assistant clique sur le bouton "rdv visité" 2. Le système change le statut du rendez de visité à non-visité dans la base de donnée 3. Le système mis à jour l'attribut « status » du rendez-vous 4. Le système actualise la page en changeant le statut du rendez-vous. <p>Possibilité 2 :</p> <ol style="list-style-type: none"> 1. L'assistant clique sur le bouton " non-visité" 2. Le système change le statut du rendez de non-visité à visité dans la base de donnée 3. Le système mis à jour l'attribut statut du rendez-vous 14. Le système actualise la page en changeant le statut du rendez-vous

TABLE 2.9 – Description textuelle de cas d'utilisation « Modifier le statut des rendez-vous»

-Description textuelle de cas d'utilisation « Ajouter Note et ordonnance »

Cas d'utilisation	Ajouter Note et ordonnance
Acteur	Médecin
Objectif	Permet au médecin d'écrire des notes relatives au patient et à lui préinscrire une ordonnance, afin de garder une "historique" sur ses patients et d'être mieux organiser
Précondition	Authentification Accéder à « patients d'aujourd'hui »
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur Clique sur le bouton « notes et ordonnance » 2. Le système affiche un formulaire. 3. Le médecin remplit le champ du formulaire. 4. Le médecin valide en cliquant sur enregistrer

TABLE 2.10 – Description textuelle de cas d'utilisation « Ajouter note et ordonnance »

-Description textuelle de cas d'utilisation « Créer un membre»

Cas d'utilisation	Créer un membre
Acteur	Admin
Objectif	Permet à l'administrateur d'ajouter des médecins ou des assistants a la communauté HealthClic
Précondition	<ol style="list-style-type: none"> 1. S'authentifier avec un compte administrateur 2. Cliquer sur "créer un nouveau membre"
Scénario nominal	<ol style="list-style-type: none"> 1. Le système charge la page qui contient le formulaire d'inscription 2. L'administrateur remplit les champs du formulaire, puis . le confirme. 3. Le système vérifie que tous les champs ont été saisi et d'une manière correcte. 4. Le système sauvegarde les données dans la base de donnée

TABLE 2.11 – Description textuelle de cas d'utilisation « Créer un membre »

-Description textuelle de cas d'utilisation « Modifier les informations d'un membre »

Cas d'utilisation	Modifier les informations d'un membre
Acteur	Administrateur
Objectif	Modifier le profil d'un médecin, administrateur ou un assistant
Précondition	Authentification Cliquer sur "Lister nos membre " Cliquer sur l'icône adéquate
Scénario nominal	1. Le système récupère les informations déjà enregistrées dans la base de données 2. Le système les affiche dans un formulaire 3. L'administrateur modifie les champs erronés, puis confirme la modification 4. le système mis à jour la base de données

TABLE 2.12 – Description textuelle de cas d'utilisation « Modifier les informations d'un membre »

-Description textuelle de cas d'utilisation « Supprimer un membre »

Cas d'utilisation	Supprimer un membre
Acteur	Administrateur
Objectif	supprimer un médecin, administrateur ou un assistant
Précondition	Authentification Cliquer sur "Lister nos membre " Cliquer sur l'icône adéquate
Scénario nominal	1. Le système renvoie une page de confirmation de la suppression 2. L'administrateur confirme la suppression 3. Le système mis a jour la base de données en supprimant l'utilisateur

TABLE 2.13 – Description textuelle de cas d'utilisation « Supprimer un membre »

-Description textuelle de cas d'utilisation « Créer une spécialité »

Cas d'utilisation	Créer une spécialité
Acteur	Administrateur
Objectif	Permet d'ajouter une spécialité
Précondition	Authentification
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur clique sur "créer une spécialité " 2. Le système lui renvoi la page contenant un formulaire d'ajout 3. L'administrateur introduit le nom de la spécialité, puis confirme 4. Le système enregistre la nouvelle spécialité dans la base de données
Scénario alternatif	Authentification échouée

TABLE 2.14 – Description textuelle de cas d'utilisation « Créer une spécialité »

-Description textuelle de cas d'utilisation « Modifier une spécialité »

Cas d'utilisation	Modifier une spécialité
Acteur	Administrateur
Objectif	Permet de modifier une spécialité déjà enregistrée
Précondition	<p>Authentification</p> <p>Accéder à Mes spécialités</p>
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur clique l'icône de modification 2. Le système lui renvoi la page contenant un formulaire avec la valeur déjà enregistrée dans la base de données 3. L'acteur modifie le nom de la spécialité, puis il confirme 4. Le système mis à jour la base de données en modifiant l'intitulé de la spécialité
Scénario alternatif	Authentification échouée

TABLE 2.15 – Description textuelle de cas d'utilisation « Modifier une spécialité »

-Description textuelle de cas d'utilisation « Supprimer une spécialité »

Cas d'utilisation	Supprimer une spécialité
Acteur	Administrateur
Objectif	Permet de Supprimer une spécialité déjà enregistrée
Précondition	Authentification Accéder à Mes spécialités
Scénario nominal	1. L'administrateur clique l'icône de suppression de la spécialité base de données 2. Le système mis à jour la base de données en supprimant la spécialité
Scénario alternatif	Authentification échouée

TABLE 2.16 – Description textuelle de cas d'utilisation « Supprimer une spécialité »

-Description textuelle de cas d'utilisation « Annuler un rendez-vous » :

Cas d'utilisation	Annuler un rendez-vous n
Acteur	Patient
Objectif	Permet au patient d'annuler un rendez-vous.
Précondition	1. Authentification 2. Consulter l'historique d'un rendez-vous.
1 Scénario nomina	1. Le Patient sélectionne un rendez-vous et clique sur le bouton Annuler 2. Le système vérifie la date de la prise de rendez-vous et s'assure qu'il ne dépasse pas d'un délai de 24h pour pouvoir l'annuler. 3. Le système lui affiche un message de confirmation. 4. Le patient valide l'annulation de rendez-vous. 5. Le système libère le rendez-vous annulé et effectue une mise à jour de l'agenda.
Scénario alternatif	Si la date de prise de rendez-vous dépasse les 24 heures, le système affiche un message d'erreur indiquant que le rendez-vous ne peut pas être annuler.

TABLE 2.17 – Description textuelle de cas d'utilisation « Annuler un rendez-vous »

-Description textuelle de cas d'utilisation « Déconnexion »

Cas d'utilisation	Déconnecter
Acteur	Administrateur, Patient, Médecin, Assistant
Objectif	Quitter l'application
Précondition	Authentification
Scénario nominal	1. L'acteur se déconnecte de l'application. 2. Le système lui affiche un message de confirmation. 3. L'acteur confirme la déconnexion.
Scénario alternatif	Si l'acteur annule la déconnexion le système réaffiche l'interface d'accueil.

TABLE 2.18 – Description textuelle de cas d'utilisation « déconnecter »

2.4 Modélisation des diagrammes de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Ces communications entre les classes sont reconnues comme des messages. Le diagramme des séquences, énumère des objets horizontalement, et le temps verticalement. Il modélise l'exécution des différents messages en fonction du temps.[32] Pour réaliser les diagrammes des séquences nous avons utilisé des opérateurs d'interactions. Un opérateur d'interaction définit le type d'un fragment composé. Les opérateurs d'interaction que nous avons utilisés dans les diagrammes de séquences sont :

1. **Référence (Ref)** : Cet opérateur désigne que le fragment fait référence à un cas vu précédemment.
2. **Alternative(Alt)** : Cet opérateur désigne que le fragment composé représente un choix de comportement. Une opérande d'interaction au maximum sera choisie. L'opérande choisie doit avoir une expression de garde implicite ou explicite qui a la valeur "true" à ce point de l'interaction.
3. **Loop** : Cet opérateur désigne que le fragment composé représente une boucle.

2.4.1 Diagramme de séquence « s'authentifier »

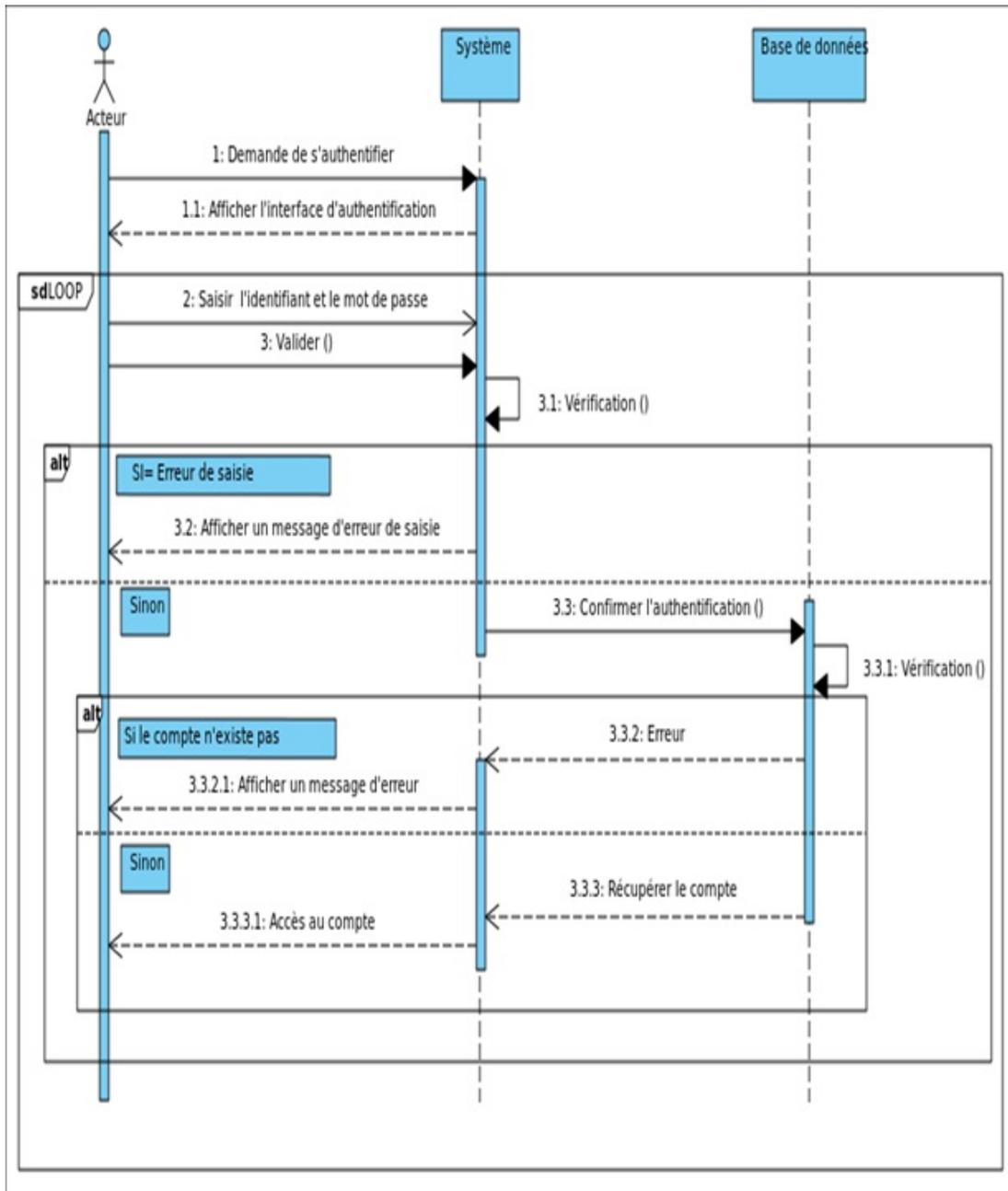


FIGURE 2.7 – Diagramme de séquence « s'authentifier »

2.4.2 Diagramme de séquence « Chercher un médecin »

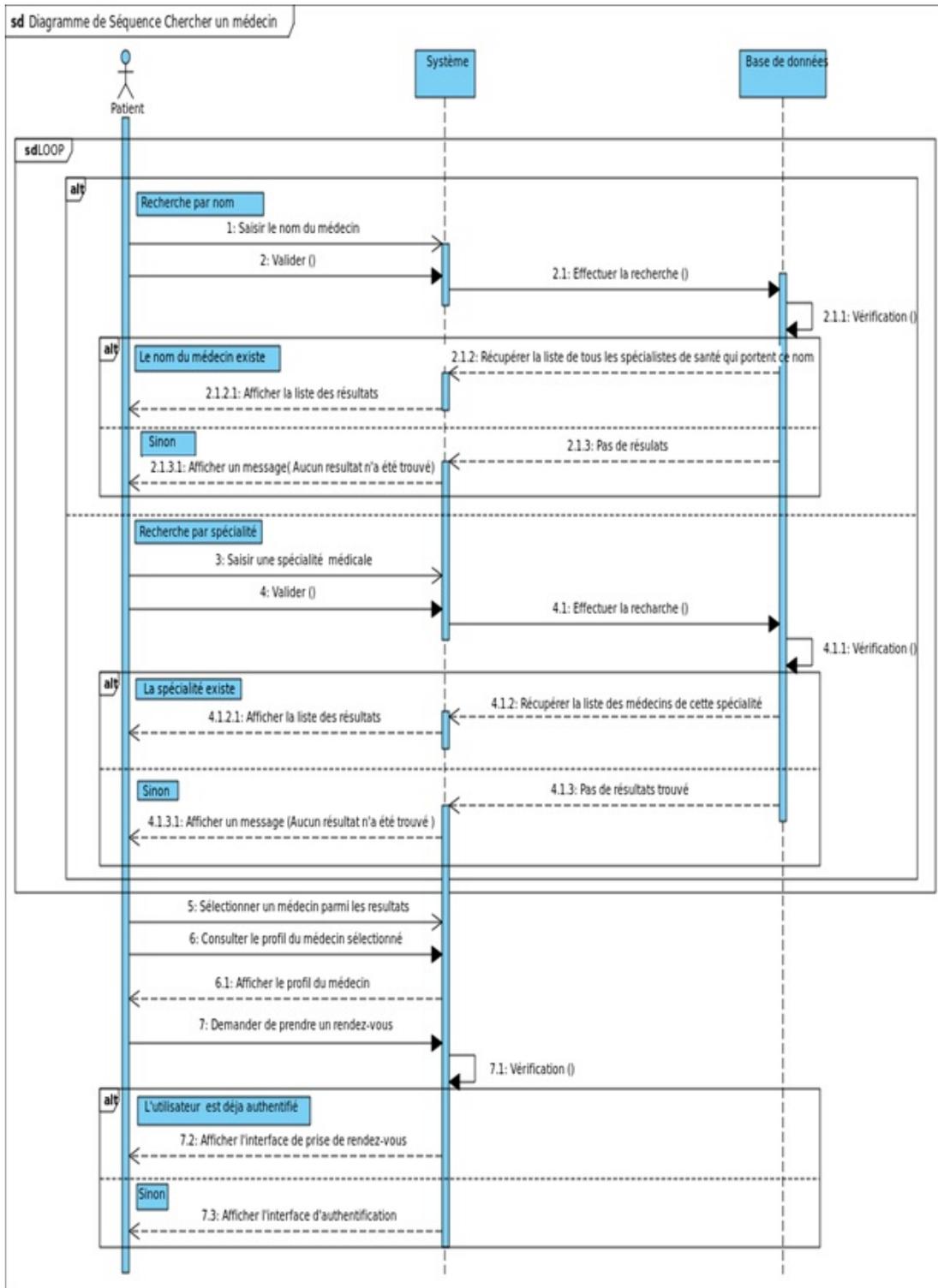


FIGURE 2.8 – Diagramme de séquence « Chercher un médecin »

2.4.3 Diagramme de séquence « Créer un compte patient »

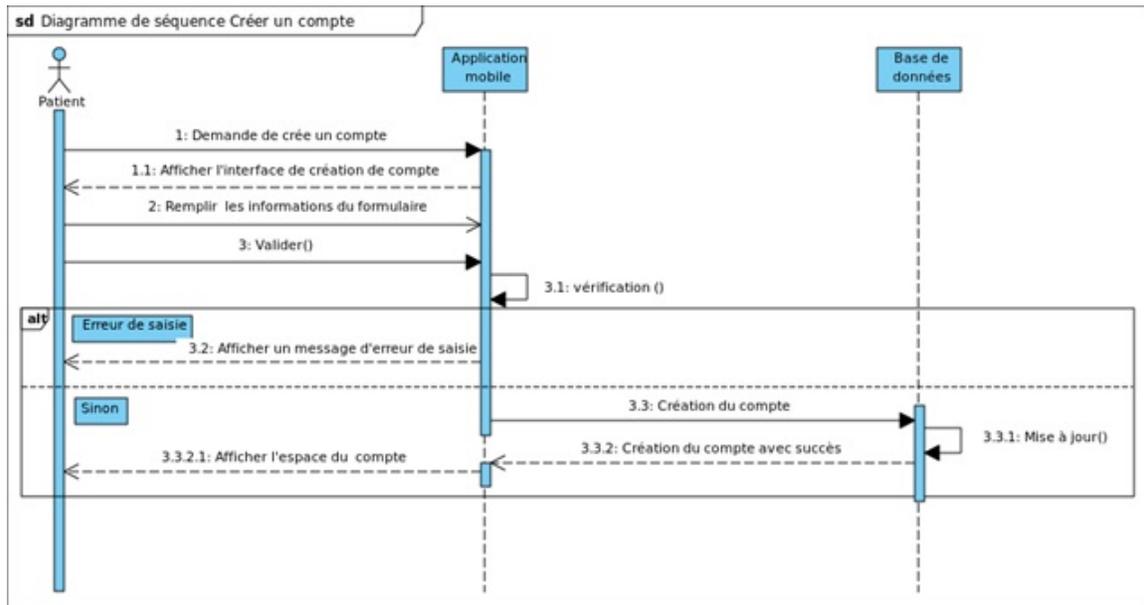


FIGURE 2.9 – Diagramme de séquence « Créer un compte patient »

2.4.4 Diagramme de séquence « Prendre un rendez-vous »

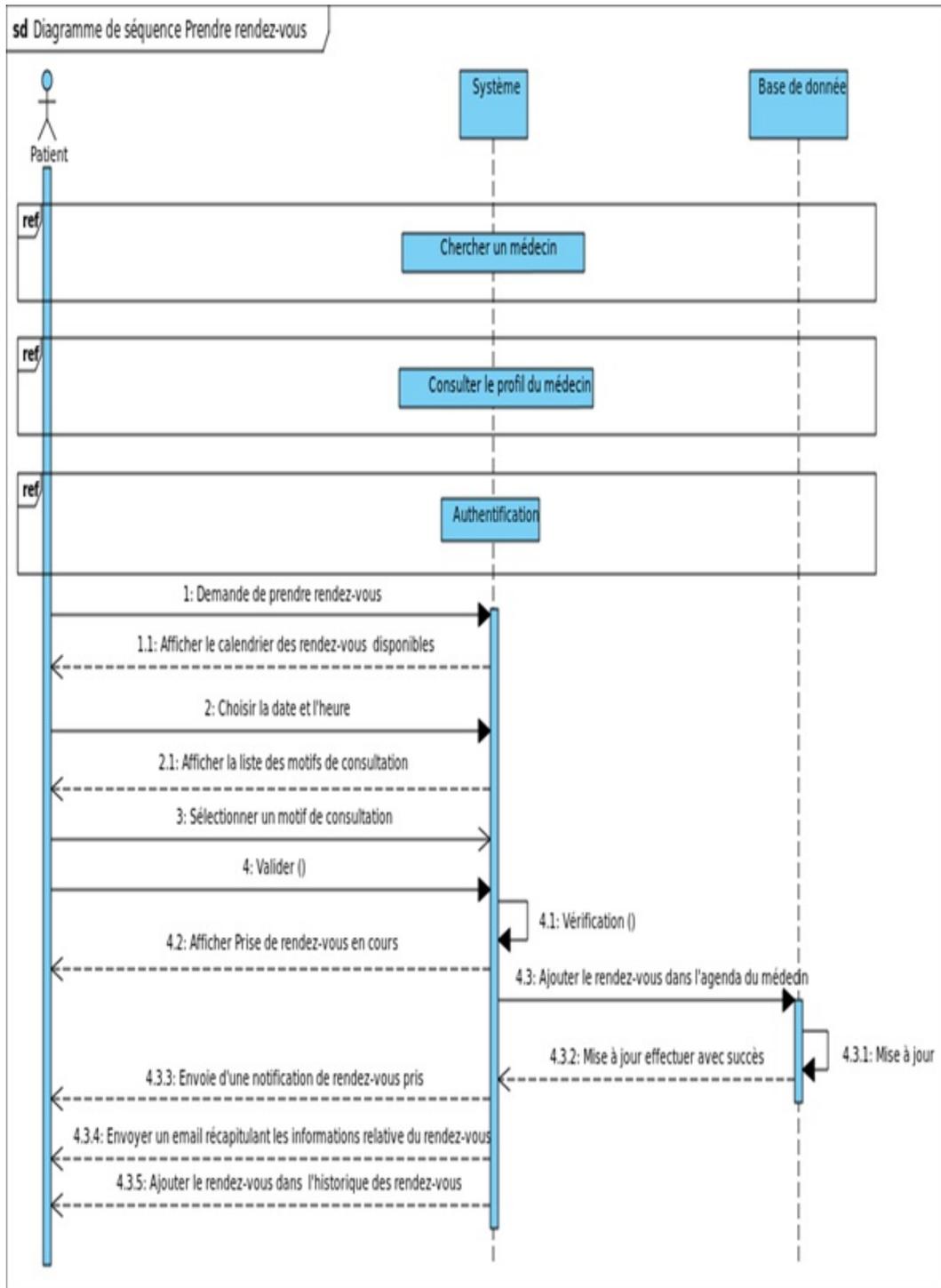


FIGURE 2.10 – Diagramme de séquence « Prendre un rendez-vous »

2.5 Modélisation du diagramme des classes

Le diagramme de classes montre la structure interne d'un système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

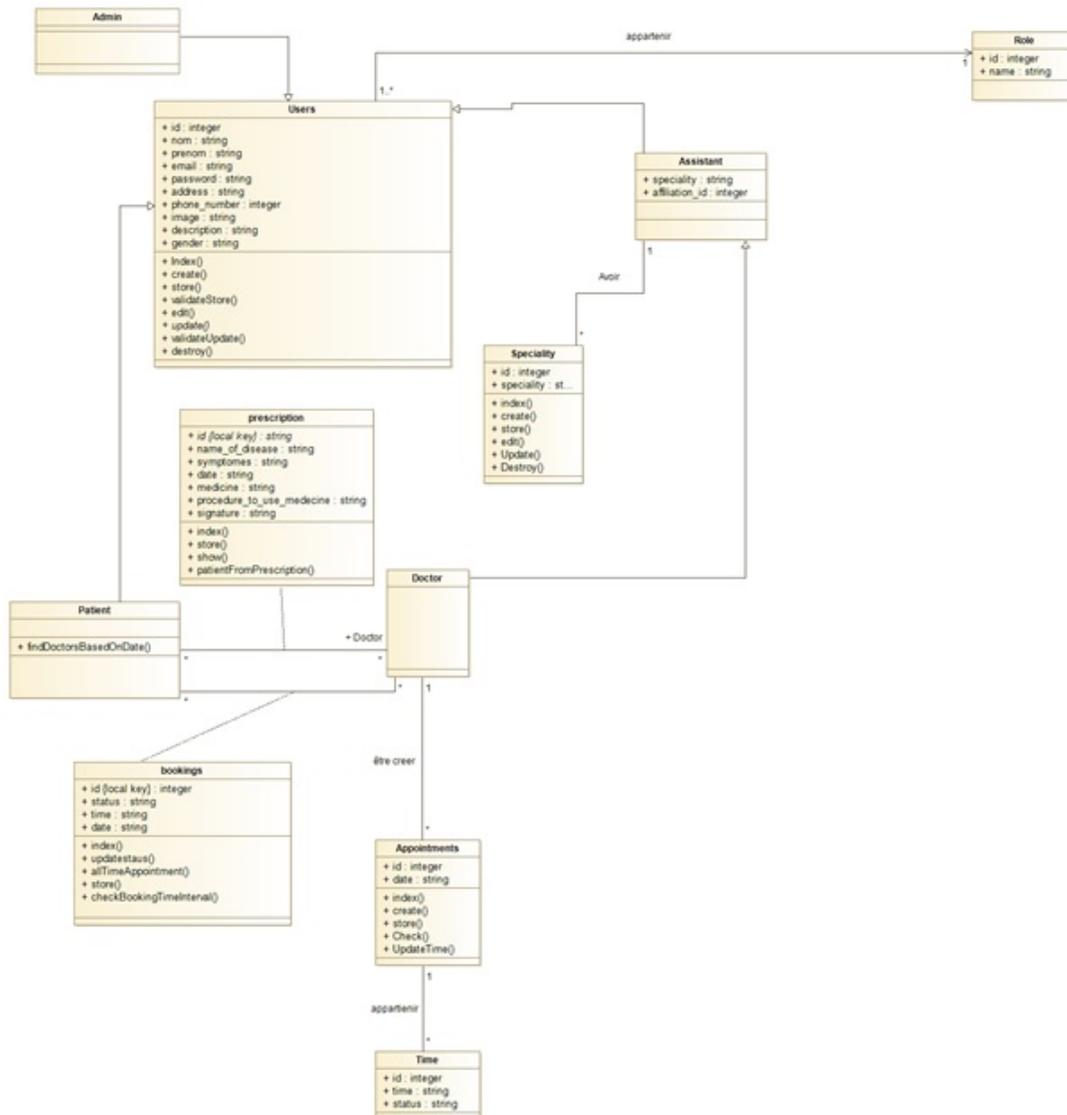


FIGURE 2.11 – Diagramme de classe

2.5.1 Modèle relationnel

-Définition

Dans ce modèle, les données sont stockées dans des tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Un ensemble de données sera donc modélisé par un ensemble de tables. Le succès du modèle relationnel auprès des chercheurs, concepteurs et utilisateurs est dû à la puissance et à la simplicité de ses concepts.

-Terminologie de l'approche relationnelle

Dans ce qui suit, nous décrirons les différentes terminologies de l'approche relationnelle.

Domaine : Il s'agit de l'ensemble des valeurs admises pour un attribut. Il établit les valeurs acceptables dans une colonne. Une relation peut faire appel au même domaine pour plusieurs de ses attributs. La notion de domaine est fondamentale en matière de gestion de données. Elle permet d'exprimer des contraintes d'intégrité sémantique très fortes sur les données d'une base de données.

Classe : Une classe représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques.

Attribut : Un attribut est un identifiant (un nom) décrivant une information stockée dans une base de données.

Clé primaire : Une clé primaire est un ensemble minimal de colonnes permettant d'identifier de manière unique chaque tuple dans une table.

Clé étrangère : Il s'agit d'une ou plusieurs colonnes dans une table qui a pour but d'assurer une liaison entre deux tables. On y arrive en dupliquant la clé primaire de la deuxième table dans la première. On l'appelle aussi clé externe.

Tuple : Un tuple représente une ligne dans une table.

-Les règles de passage au model relationnel

Dans cette section, nous procédons à la construction pas à pas du modèle relationnel implémentant la base de données de notre application à partir du modèle de classes obtenu. Les règles utilisées pour le passage sont décrites ci-après :

Règle1 : Transformation des classes : chaque classe du diagramme UML devient une relation.

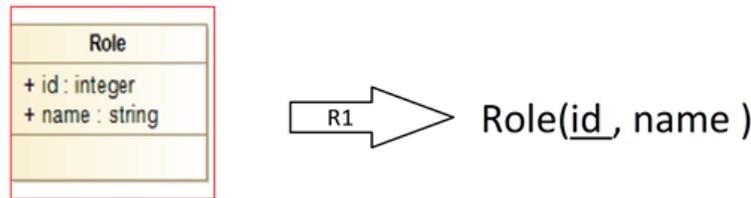


FIGURE 2.12 – Règle de transformation des classes.

En appliquant cette règle à notre modèle, nous obtenons les relations suivantes :

- **Users** (id, nom , prenom , email , password , address , phone_number , speciality , image , description , gender , affiliation_id)
- **Doctor** (id, nom , prenom , email , password , address , phone_number , speciality , image , description , gender)
- **Admin**(id, nom , prenom , email , password , address , phone_number , speciality , image , description , gender)
- **Patient**(id, nom , prenom , email , password , address , phone_number , image , description , gender)
- **Role**(id , name)
- **Bookings**(id , time , date , status)
- **Appointments**(id, date)
- **Times**(id, time , status)
- **Prescription**(id , name_of_disease , symptomes , date , medicine , procedure_to_use_medicine , signature)
- **Speciality**(id_,speciality).

Trois décompositions sont possibles pour traduire la relation d'héritage en fonction des contraintes existantes :

- **Décomposition par distinction** : Il faut transformer chaque sous-classe en une relation. La clé primaire de la sur-classe, migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.
- **Décomposition descendante (push-down)** :S'il existe une contrainte de totalité ou de partition sur l'association d'héritage, il est possible de ne pas traduire la relation

issue de la sur-classe. Il faut alors faire migrer tous ses attributs dans la (les) relation(s) issue(s) de la (des) sous-classe(s).

- **Décomposition ascendante (push-up)** : Il faut supprimer la (les) relation(s) issue(s) de la (des) sous-classe(s) et faire migrer les attributs dans la relation issue de là sur-classe.

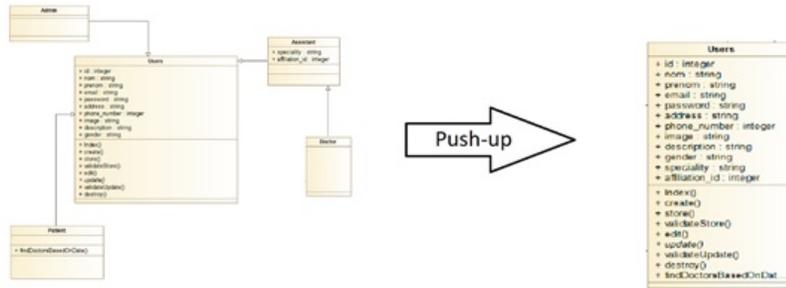


FIGURE 2.13 – Règle de transformation de l’héritage (push-down).

Application de la décomposition descendante (push-up) : En appliquant la règle de transformation de l’héritage nous obtiendrons le diagramme de classe suivant :

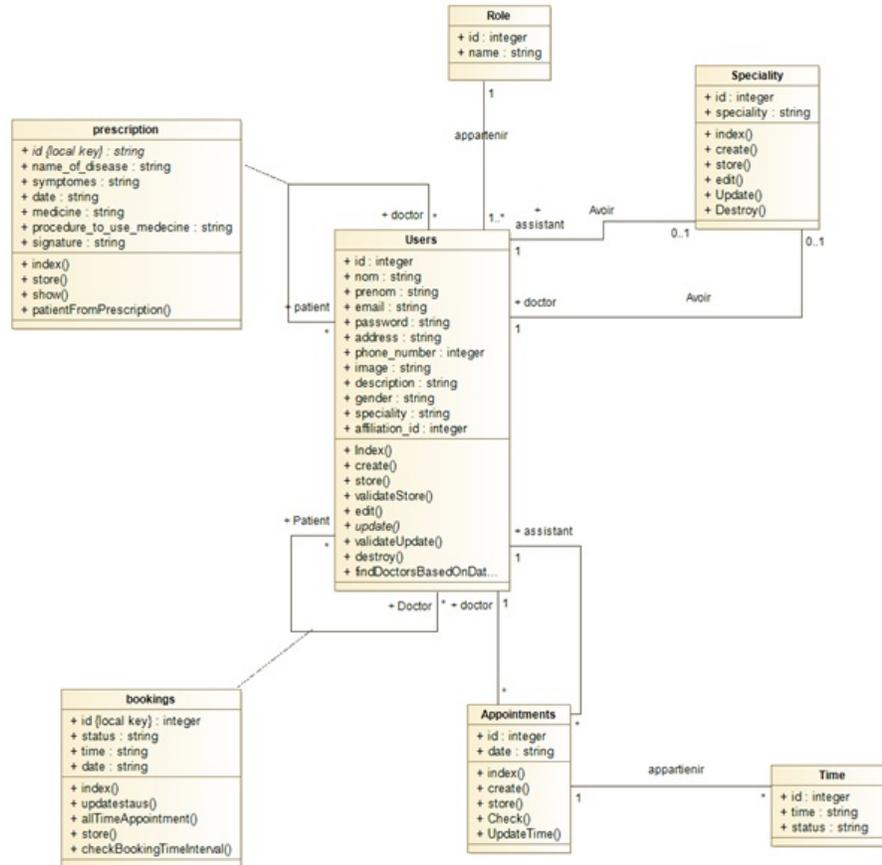


FIGURE 2.14 – Diagramme de classe après application de la règle de transformation de l’héritage.

Règle3 : Transformation des associations : Nous distinguons trois familles d’associations :

- **Association 1...*** : Il faut ajouter un attribut de type clé étrangère dans la relation fils de l’association. L’attribut porte le nom de la clé primaire de la relation père de l’association.
- **Association *... *** : Et n-aire et classe-association : La classe-association devient une relation, et la clé primaire de cette relation est la concaténation des identifiants des classes connectées à l’association.
- **Association 1...1** : Il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L’attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l’association.

En appliquant cette règle à notre modèle, nous obtenons les relations suivantes :

- **Users** (id, nom , prenom , email , password , address , phone_number , speciality , image , description , gender , affiliation_id, #role_id)
- **Role**(id , name).

- **Bookings**(id , time , date , status , #doctor_id , #user_id)
- **Appointments**(id, date , #user_id).
- (id, time , status , #appointment_id).
- **Prescription**(id , name_of_disease , symptomes , date , medicine , procedure_to_use_medicine , signature , #Doctor_id , #user_id).
- **Speciality**(id_ ,speciality).

2.6 Conclusion

Dans ce chapitre, nous avons décrit de façon détaillée les cas d'utilisation en recensant de manière textuelle toutes les interactions entre les acteurs et le système. Nous avons complété cette description textuelle par une représentation graphique UML : le diagramme de séquence. Par la suite, en définissant les relations entre les entités, nous sommes parvenus à concevoir le dictionnaire de données, le diagramme de classes et enfin le modèle relationnel. Donnant ainsi une vue plus structurée des éléments qui formeront la base de données liée à notre application. Enfin, ce chapitre nous a permis de préparer la phase de réalisation qui concrétisera tout ce qui a été présenté jusque-là.

Chapitre 3

Réalisation et fonctionnement

3.1 Introduction

Ce chapitre est consacré à la partie réalisation de notre plate-forme. Il présente une description des outils de développement utilisés ainsi qu'une présentation de notre application mobile et web.

Effectivement, en implémentant une solution qui se base sur des technologies gratuites et open source, nous avons plus de chance d'éviter toutes sortes de problèmes liés aux licences, contrats, etc., réduisant ainsi les coûts.

3.2 Architecture du système

Le projet comporte deux applications à réaliser qui partagent un ensemble d'informations mais sont destinées à différents publics. Pour cela il faudrait que ces applications partagent une base de données commune[33].

Une architecture client-serveur n - tiers se décerne donc comme solution architecturale à notre cas. Au cœur de cette architecture un composant API devra fournir toutes les fonctionnalités métiers citées dans le projet.

Les deux applications web et mobile utilisent une architectures MVC qui permet de bien organiser le code source. Cette architecture permet de séparer la logique du code en trois parties dans des fichiers distincts. Cela nous a aider à savoir quels fichiers créer et surtout définir leur rôle[34]

3.2.1 Le model MVC (Model-View-Controller)

Est un motif d'architecture logicielle destiné aux interfaces graphiques. Il met l'accent sur la séparation entre la logique métier et l'affichage des interfaces. Cette dernière permet une meilleure répartition du travail et une maintenance améliorée.

Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

- **Model (modèle)** : Gère les données et la logique métier.
- **View (vue)** : Gère la disposition et l'affichage.
- **Controller (contrôleur)** : Achemine les commandes des parties "model" et "view".

3.3 L'environnement de développement de HealthClic

3.3.1 L'environnement matériel

Pour le développement de notre application nous avons utilisé un PC portable « HP Probook » dont la configuration est la suivante :

- Processeur Intel ®Core™ i7-6600CPU avec fréquence 2.81 GHz
- Quantité de mémoire vive 8 Go

De plus, pour tester notre application sur les plateformes mobiles, nous avons utilisé un éditeur virtuel

- Name : Pixel 3a API 29
- CPU : Google Play Intel Atom (x86)
- Resolution :1080 x 2220 :440dpi
- API : level 29
- Size on Disk : 13 GB
- Target 10.0(Google Play)
- SD Card : 512M

3.3.2 L'environnement logiciel

Dans cette section nous présenterons l'ensemble des logiciels utilisés. Nous définirons l'environnement de développement Android Studio et les différents outils qui ont servis au développement de notre plateforme.

1. **Android**

Android est un système d'exploitation mobile fondé sur le noyau Linux et développé par Google, lancé en juin 2007[35].

2. **Android Studio**

Est un environnement de développement (IDE) pour développer des applications sous la plateforme Android. Il est basé sur IntelliJ IDEA de JetBrains et permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android [36].

L'environnement devient alors conseillé par Google et l'environnement Eclipse se voit délaissé.

3. **Visual Studio** Visual Studio Code est un éditeur de code source,léger, mais puissant qui s'exécute sur son bureau,et disponible pour Windows,Mac Os et linux. Il est livré avec une prise en charge intégrée de JavaScript,TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages.

1. **MySQL**

Le terme MySQL, pour My Structured Query Language, désigne un serveur de base de données distribuées sous licence libre GNU (General Public License). Il est, la plupart du temps, intégré dans la suite de logiciels LAMP qui comprend un système d'exploitation (Linux), un serveur web (Apache) et un langage de script (PHP).

Créé en 1995, le serveur MySQL peut être utilisé sur de nombreux systèmes d'exploitation (Windows, Mac OS, etc.). Il supporte les langages informatiques SQL et SQL/PSM.

3.3.3 Langages de programmation utilisés

Un langage de programmation est un langage qui peut être utilisé pour exprimer de l'information ou de la connaissance, ou des systèmes dans une structure qui est définie par un ensemble cohérent de règles. Toute application Web est construite à l'aide de plusieurs technologies. La combinaison de ces technologies est appelée une « pile », popularisée par la pile LAMP, qui est un acronyme pour Linux, Apache, MySQL, PHP, qui sont tous des logiciels libres[37].

A mesure que le développement Web mûrissait et que leur interactivité prenait de l'importance, Les applications (Single Page Application) sont devenues plus populaires.

Un SPA est un paradigme d'application Web qui évite d'aller chercher le contenu d'une page Web entière du serveur pour afficher de nouveaux contenus. Il utilise à la place des appels

légers pour le serveur pour obtenir des données ou des extraits et modifie la page Web. Le résultat est plutôt chouette en comparaison à l'ancienne façon de recharger entièrement la page. Cela a entraîné une augmentation des frameworks front-end, car beaucoup de travail a été fait sur le front-end[38].

1. Le langage PHP

Le PHP, pour Hypertext Preprocessor, désigne un langage informatique, autrement dit un langage de script, créé au début des années 1990 par le Canadien et Groenlandais Rasmus Lerdorf. Il est utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage sous une licence libre qui peut donc être utilisé gratuitement.

le langage PHP est souvent associé au serveur de base de données MySQL et au serveur Apache.

2. Le langage SQL

Le SQL (Structured Query Language) est l'un des plus anciens langages de programmation informatiques pour bases de données relationnelles. Il s'agit aussi du plus populaire[39]. Grâce à ce langage, il est possible d'interroger les données par le biais de requêtes sur une base de données. Les informations peuvent aussi être mises à jour et organisées. Des données peuvent être ajoutées ou supprimées. En outre, SQL permet de créer ou de modifier la structure d'un système de base de données, de l'optimiser et d'en contrôler l'accès.

3. Le langage de programmation HTML5

HTML5 pour HyperText Markup Language 5, est une version du célèbre format HTML utilisé pour concevoir les sites Internet. Celui-ci se résume à un langage de balisage qui sert à l'écriture de l'hypertexte indispensable à la mise en forme d'une page Web.

Lancée en octobre 2014, cette version HTML5 apporte de nouveaux éléments et de nouveaux attributs par rapport à la version précédente. Elle offre par exemple la possibilité de définir le contenu principal d'une page Web, d'ajouter une introduction, d'insérer un sous-titre à un contenu multimédia de type vidéo, etc.

4. Le langage de programmation CSS

Le CSS pour Cascading Style Sheets, est un langage informatique utilisé sur Internet, pour la mise en forme de fichiers et de pages HTML. On le traduit en français par feuilles de style en cascade[40].

Apparu dans les années 1990, il se présente comme une alternative à la mise en forme via des balises, notamment HTML. Un peu plus complexe à maîtriser, il permet un

gain de temps considérable dans la mise en forme d'une page web par rapport à ces balises. Grâce à lui, on peut en effet appliquer des règles de mise en forme (titrage, alignement, polices, couleurs, bordures, etc.) à plusieurs documents simultanément.

5. Le langage de programmation JavaScript

JavaScript est un langage de programmation de scripts orienté objet principalement employé dans les pages web interactives. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web[41]. Créé en 1995 par Brendan Eich, en même temps que la technologie Java, le langage JavaScript se distingue des langages serveurs par le fait que l'exécution des tâches est opérée par le navigateur lui-même, sur l'ordinateur de l'utilisateur, et non sur le serveur web. Il s'active donc généralement sur le poste client plutôt que côté serveur[42].

3.3.4 Les Frameworks utilisés

Un Framework est, comme son nom l'indique en anglais, « un cadre de travail ». Son objectif est généralement de simplifier le travail des développeurs informatiques, en leur offrant une architecture "prête à l'emploi" et qui leur permette de ne pas repartir de zéro à chaque nouveau projet[43].

Les principaux avantages des Frameworks sont donc :

- La réutilisation des codes.
- La standardisation de la programmation.
- La formalisation d'une architecture adaptée aux besoins de chaque entreprise.

1. Bootstrap

C'est un Framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript, et fournit aux développeurs des outils pour créer un site facilement[44].

Ce Framework est pensé pour développer des sites avec un design responsif, qui s'adapte à tout type d'écran, et en priorité pour les smartphones[45].

2. Laravel

C'est un Framework PHP open source qui suit la logique du design pattern MVC, et qui propose des outils pour construire une application web. Le créateur de Laravel, Taylor Otwell, a regroupé les bibliothèques les plus performantes pour effectuer, chaque fonctionnalité nécessaire à la création d'un site web[46].

3. React Native

Est un framework d'applications mobiles open source créé par Facebook. Il est utilisé pour développer des applications pour Android, iOS en permettant aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes[47].

3.4 La charte graphique de HealthClic

La charte graphique, dont la dénomination correcte est cahier des normes graphiques, est un document de travail qui contient l'ensemble des règles fondamentales d'utilisation des signes graphiques qui constituent l'identité graphique d'une organisation, d'un projet, d'une entreprise.

Dans ce qui suit, nous présenterons la charte graphique de notre plate-forme HealthClic :

1. Le choix de couleurs

Etant donné que HealthClic sera utilisée dans le domaine médical, ce qui fait qu'on a opté pour une couleur qui est adéquate au thème.

Un vert qui invite au calme et au repos. Il est symbole de croissance, de santé, de fraîcheur et de nature.



FIGURE 3.1 – le code des couleurs utiliser dans le design de la platforme HealthClic

2. Le Logo de l'application

Le logo de notre application, a été choisi d'une manière à ce qu'il soit soft, qu'il mette en avant le nom attribué à nos applications (web et mobile) ainsi que sa plus grande utilité qu'est la prise de rendez-vous en un clic.



FIGURE 3.2 – Présentation du logo de la platforme HealthClic

3.5 Présentation des différentes interfaces de notre plateforme

Dans cette section, nous allons présenter le rendu final de notre travail qui sera illustré par les interfaces d'utilisation des applications web et mobile développées par nos soins.

3.5.1 L'organigramme des vues de notre application :

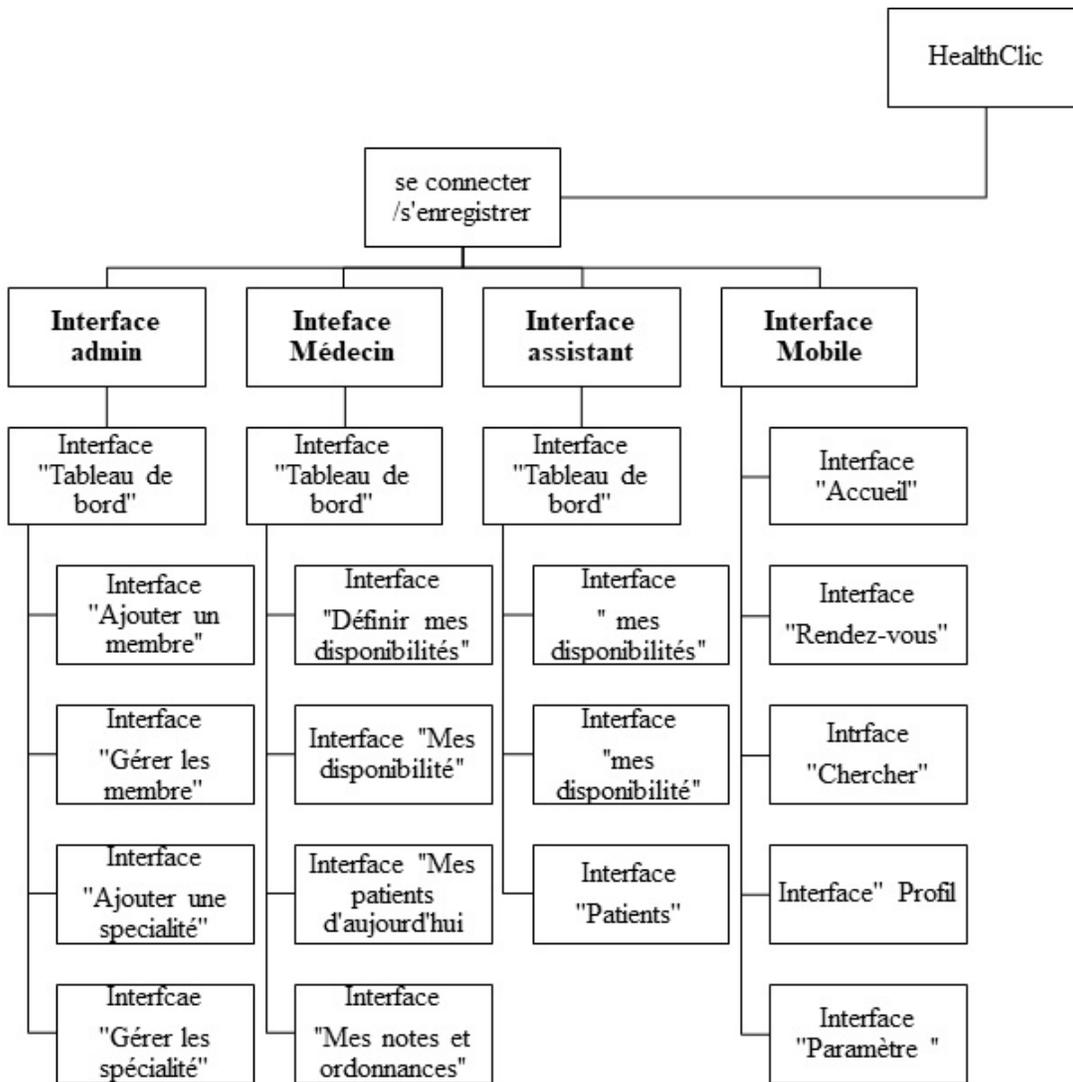


FIGURE 3.3 – L'organigramme des vues de notre application

3.6 Présentation des interfaces de l'application web

Dans cette section, nous présentons les différentes interfaces de notre application. Chaque utilisateur a ses propres interfaces qui lui servent pour exécuter certaines fonctionnalités, ces interfaces sont les suivantes :

— Accueil

En suivant le lien de notre site, l'interface ci-dessous s'affiche. On y trouve une brève présentation du site, et les boutons d'authentification pour les médecins, assistants et administrateurs, et dans le cas où le visiteur est un patient, elle lui permet de télécharger l'application qui lui permettra de s'inscrire ou de se connecter à son compte, si ce dernier est déjà créé (Voir la figure 3.4)

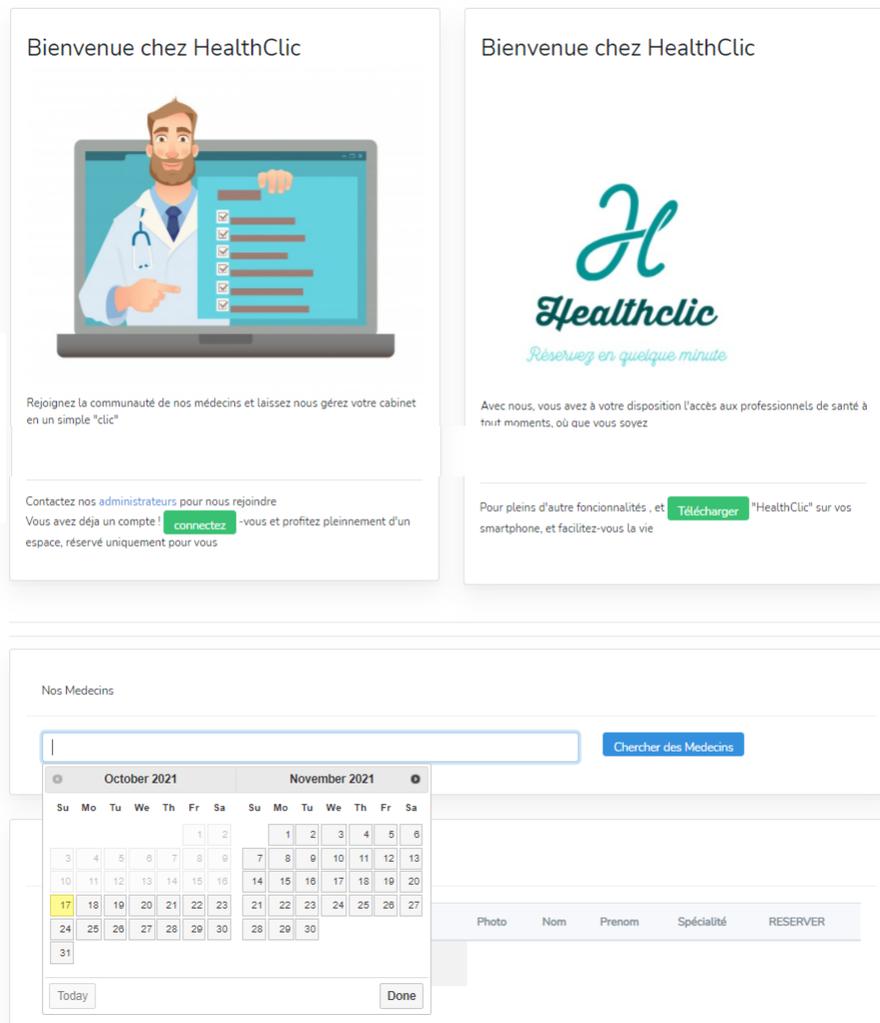


FIGURE 3.4 – Présentation de l'interface d'accueil de l'application web

— Les interfaces « médecins »

Une fois le médecin connecté, il aura le tableau de bord suivant qui s'affichera (voir figure 3.5)

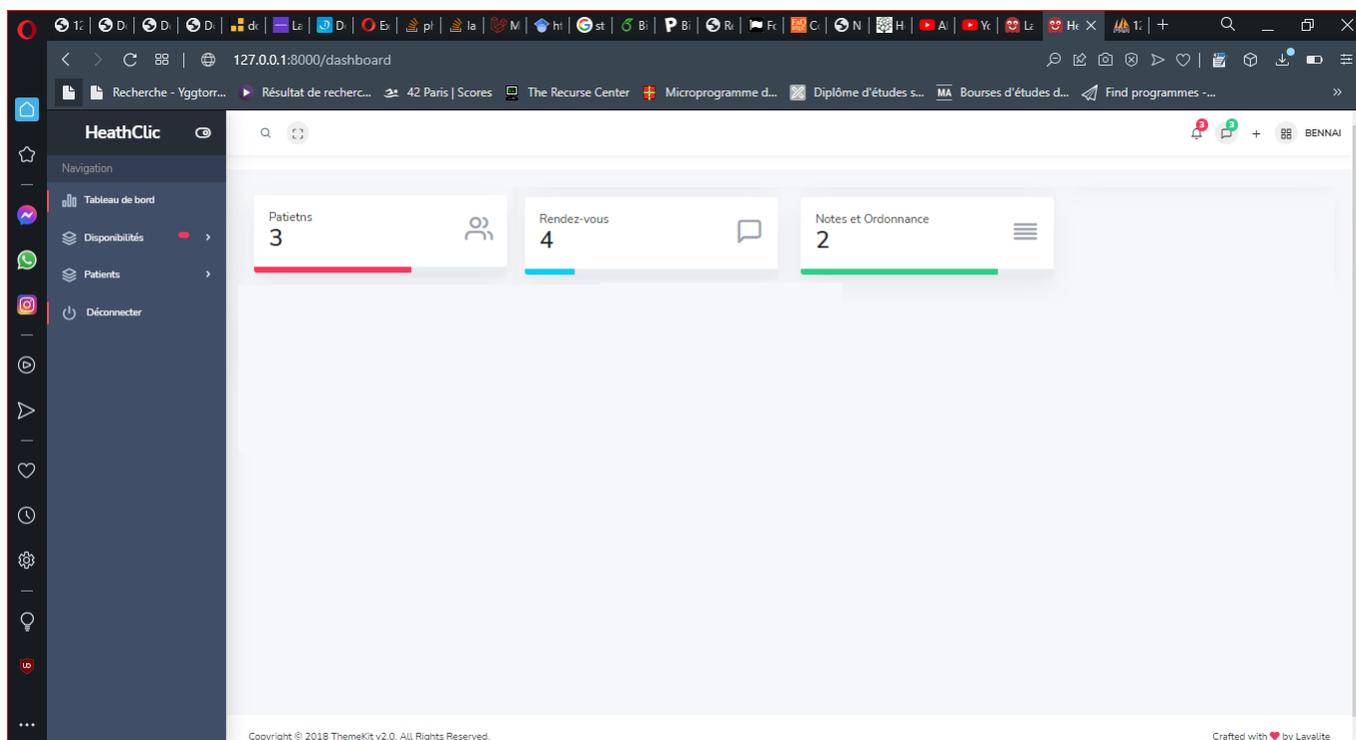


FIGURE 3.5 – Présentation du tableau de bord de l'application web

Le menu sur la gauche de la fenêtre permet au médecin de :

— Définir ses disponibilités (figure 3.6) :

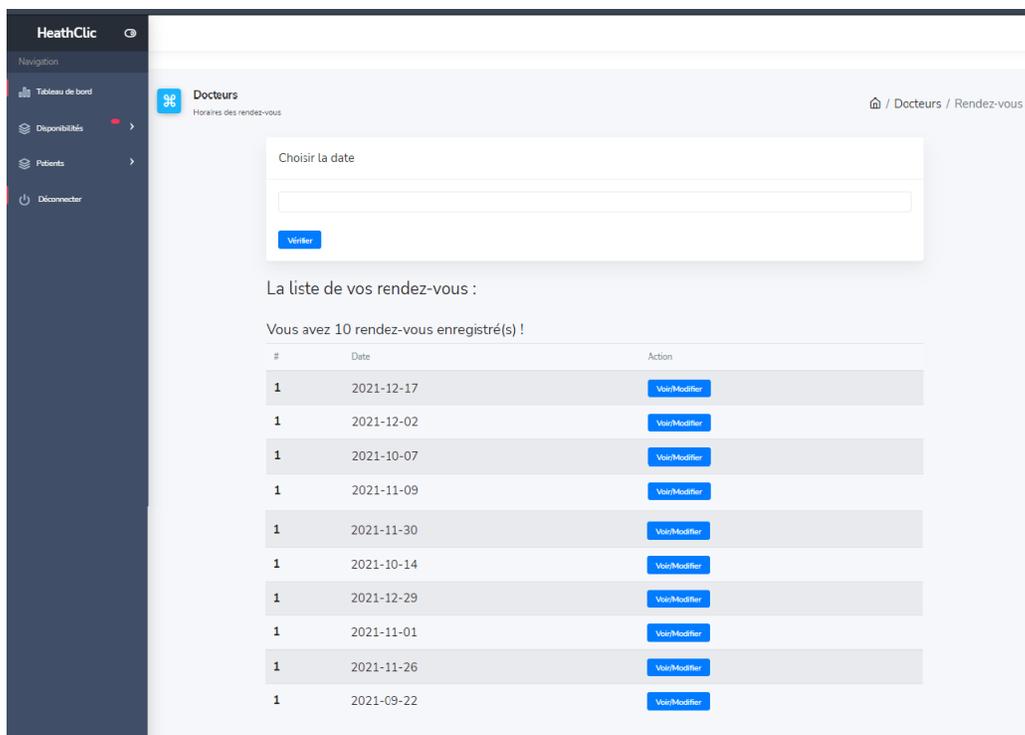


FIGURE 3.6 – Présentation de l’interface « Définir ses disponibilités »

— Voir ou modifier ses disponibilités(figure 3.7)

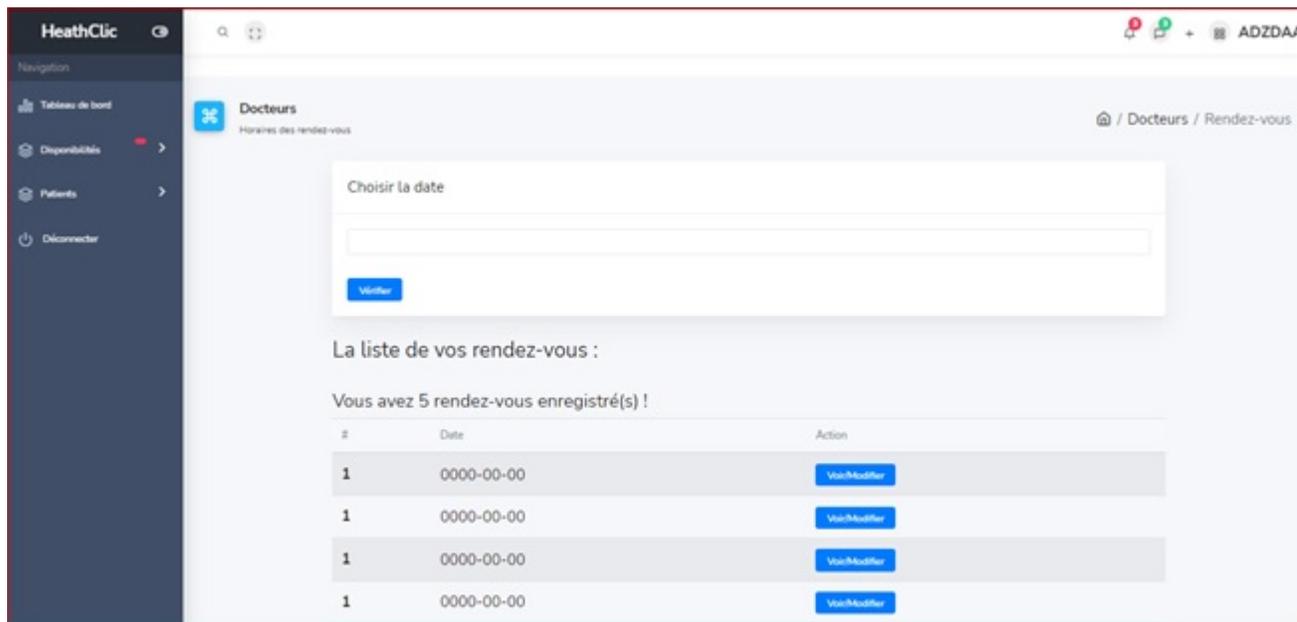


FIGURE 3.7 – Présentation de l’interface « Modifier ses disponibilités »

- Avoir un aperçu sur les patients déjà reçus (figure 3.8)

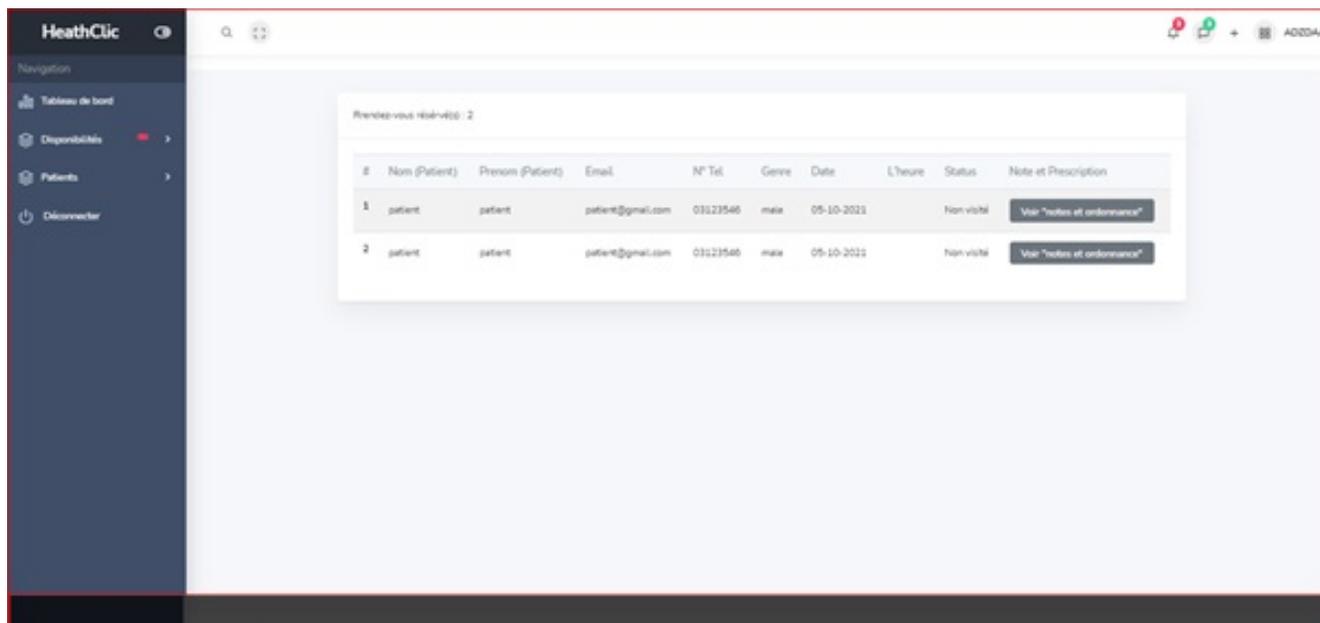


FIGURE 3.8 – Présentation de l'interface « La liste des patients »

- **les interfaces « Administrateur »**

après l'authentification, l'administrateur aura accès au tableau de bord.

En cliquant sur « nos membres », sur la sidebar, vous pouvez :

- Ajouter un nouveau membre : En renseignant un formulaire résumant les informations personnelles, les coordonnées, et le rôle du membre sur la plateforme. Il est à préciser que les membres peuvent être soit, médecins, assistants ou administrateurs (figure 3.9).

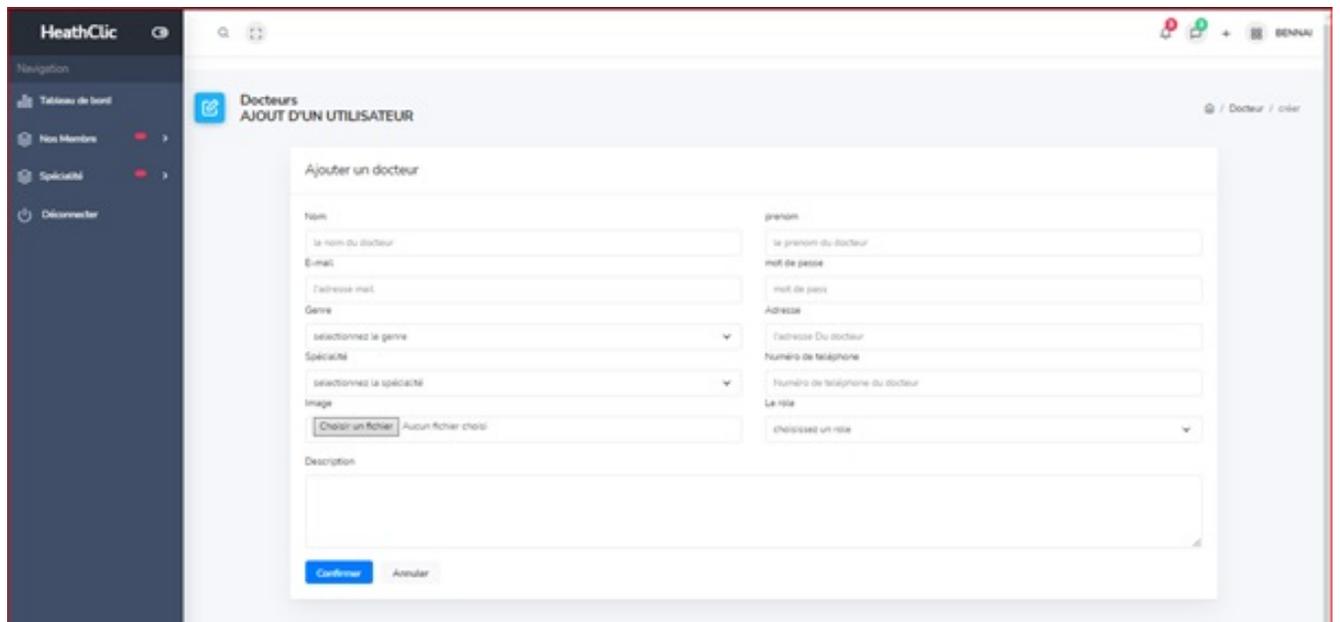


FIGURE 3.9 – Présentation de l'interface « Ajout d'un utilisateur »

— Avoir la liste des membres, tout rôle confondus (figure 3.10)

Nom	Prénom	Avatar	Email	Adresse	Numéro de téléphone	Spécialité
Bennai	brahim		bennai@gmail.com	Sidi ahmed	24	neurologue
Bennai	meriem		meriem@gmail.com	ssss	555	neurologue
BENNAI	Brahim		doctor2@gmail.com		66	cardiologue
qsd	qsd		assisstant@gmail.com	qsd	024	psychologie 1.2
qsd	qsd		aa@gmail.com	qsd	024	psychologie 1.2
BENNAI	Yasmina		yasmina@gmail.com	SA	01	Cardiologie
qsd	d		bennai@gmail.coms	zs	544	Rhumatologie
lkajf	fsldkjgkls		qlskfkljksfkfj@oksfjfs	sqdpsdq	44	Cardiologie
test az	Brahim		bennai@gmail.comssssssss	SA	544	Rhumatologie
test	sqdfq		bennai@gmail.comdfsfstf	sqdpsdq	5555555555555555	Cardiologie
qs.fdf.t:	fsqqs_fq,f		my_email@qmslkfmsl.sdq	sqdpsdq	444	Cardiologie

FIGURE 3.10 – Présentation de l'interface « La liste des membres »

Tout en ayant la possibilité de :

— visualiser un profil voir la (figure 3.11)

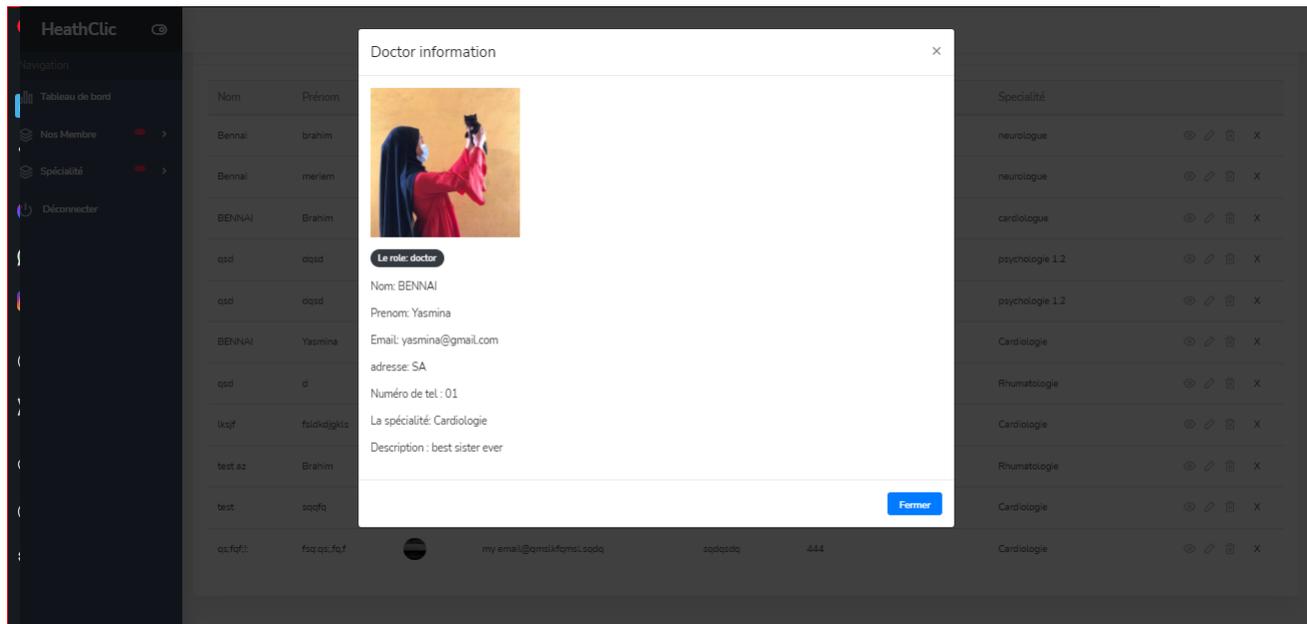


FIGURE 3.11 – Présentation de l'interface « Profil d'un médecin »

— Modifier un utilisateur (figure 3.12)

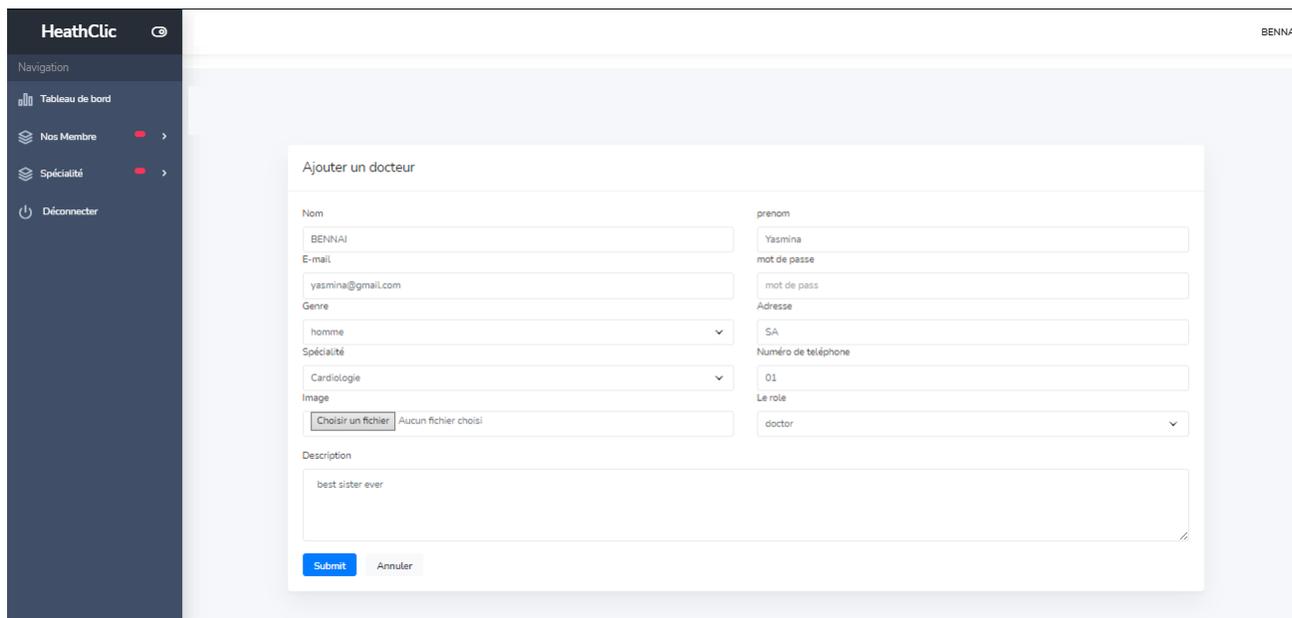


FIGURE 3.12 – Présentation de l'interface « Modifier un utilisateur »

— Supprimer un utilisateur(figure 3.13)

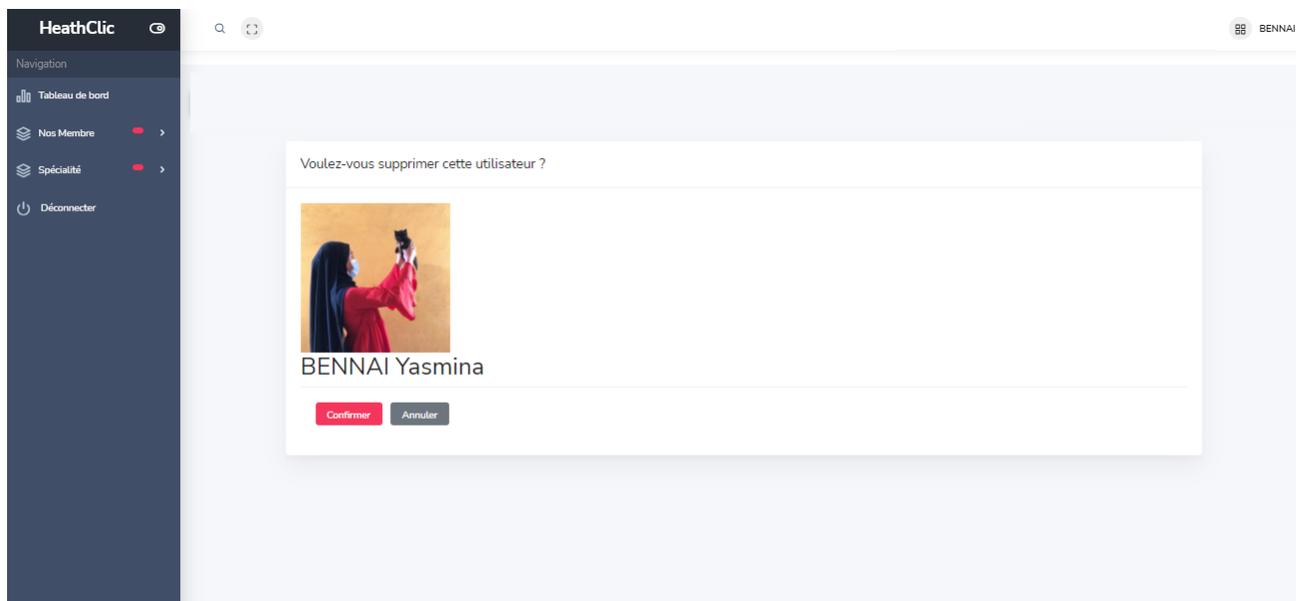


FIGURE 3.13 – Présentation de l’interface « Supprimer un utilisateur »

Toujours sur le menu de gauche, en cliquant sur « Spécialité », nous pouvons :

- Créer une nouvelle spécialité :

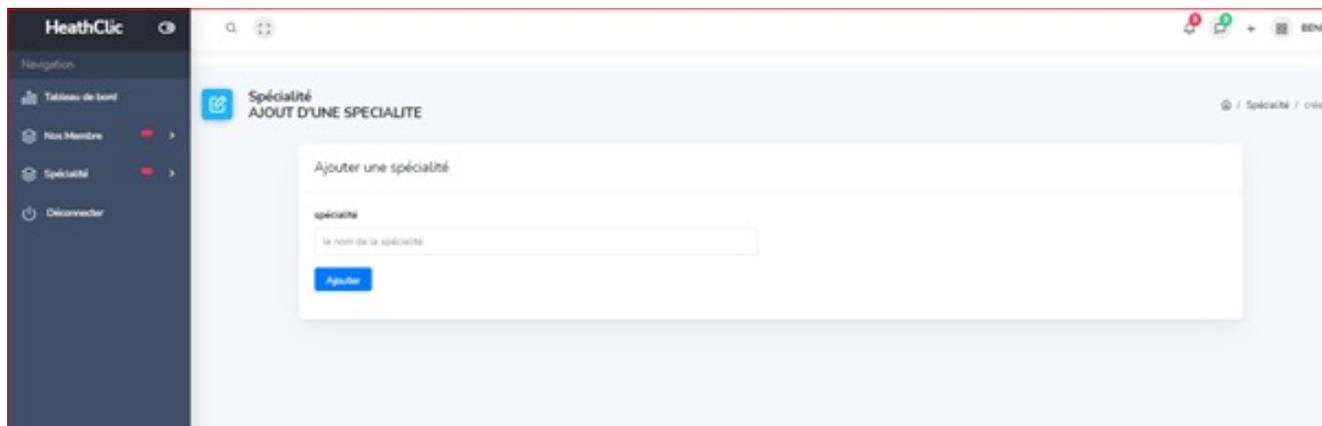


FIGURE 3.14 – Présentation de l’interface « Créer une nouvelle spécialité »

- Voir la liste des spécialités (figure 3.14) :

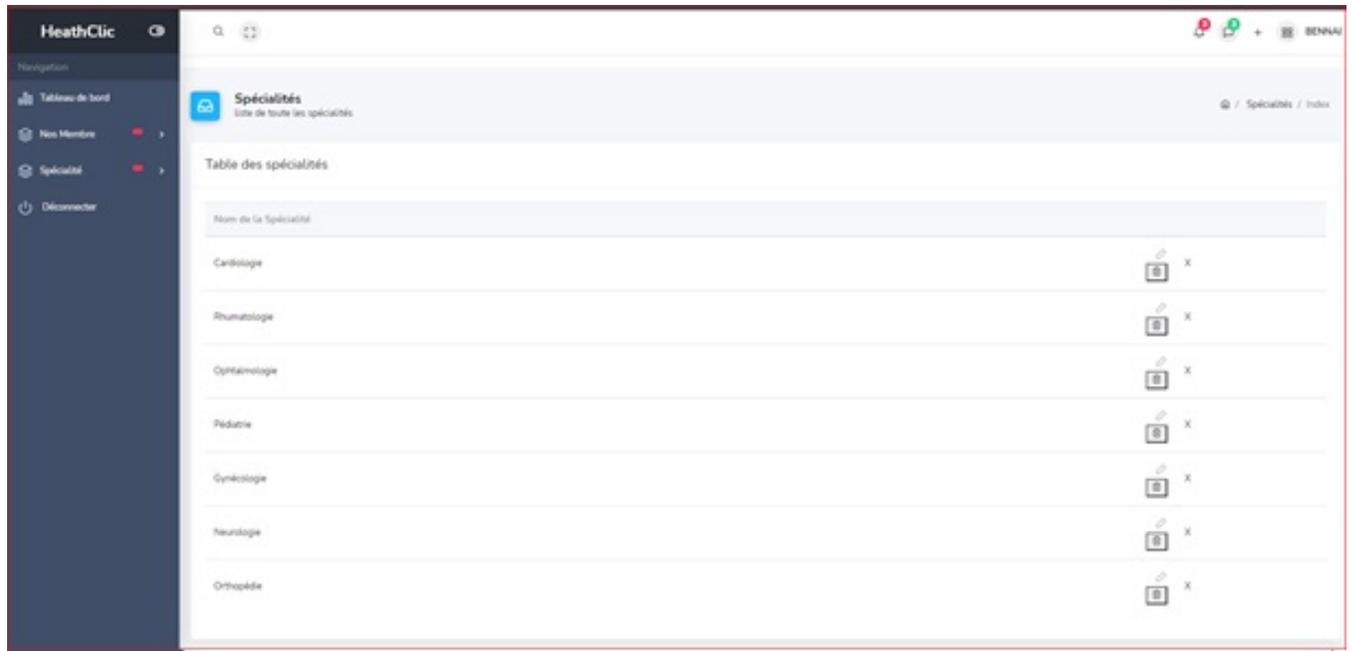


FIGURE 3.15 – Présentation de l'interface « Spécialités »

Sur cette interface, vous pouvez modifier ou supprimer une spécialité déjà existante.

— **Les interfaces de l' Assistant :**

Après l'authentification, l'assistant aura à peu près les mêmes fonctionnalités que son médecin.

- Il aura à définir les disponibilités de son médecin.
 - Voir ou modifier les disponibilités de son médecin.
 - Consultez les rendez-vous d'une date spécifique.
- Sur l'espace assistant, l'acteur doit valider le passage du patient chez le médecin sur une interface regroupant tous les rendez-vous, ou en filtrant par date.

3.7 Présentation des interfaces de l'application mobile

Notre application mobile est responsive qui s'exécute sur les smartphones même sur les tablettes

1. L'interface d'écran de démarrage de l'application mobile

C'est la première interface qui s'affiche au patient après que ce dernier a décidé de télécharger et d'installer l'application dans son smartphone (figure 3.16)



FIGURE 3.16 – Présentation de l'interface « Démarrage »

2. Interface de l'ambording

Une section d'interface (figure 3.17) qui résume les services de base de notre plateforme ce qui fait qu'elle donne une vision générale sur les fonctionnalités basiques de cette dernière.

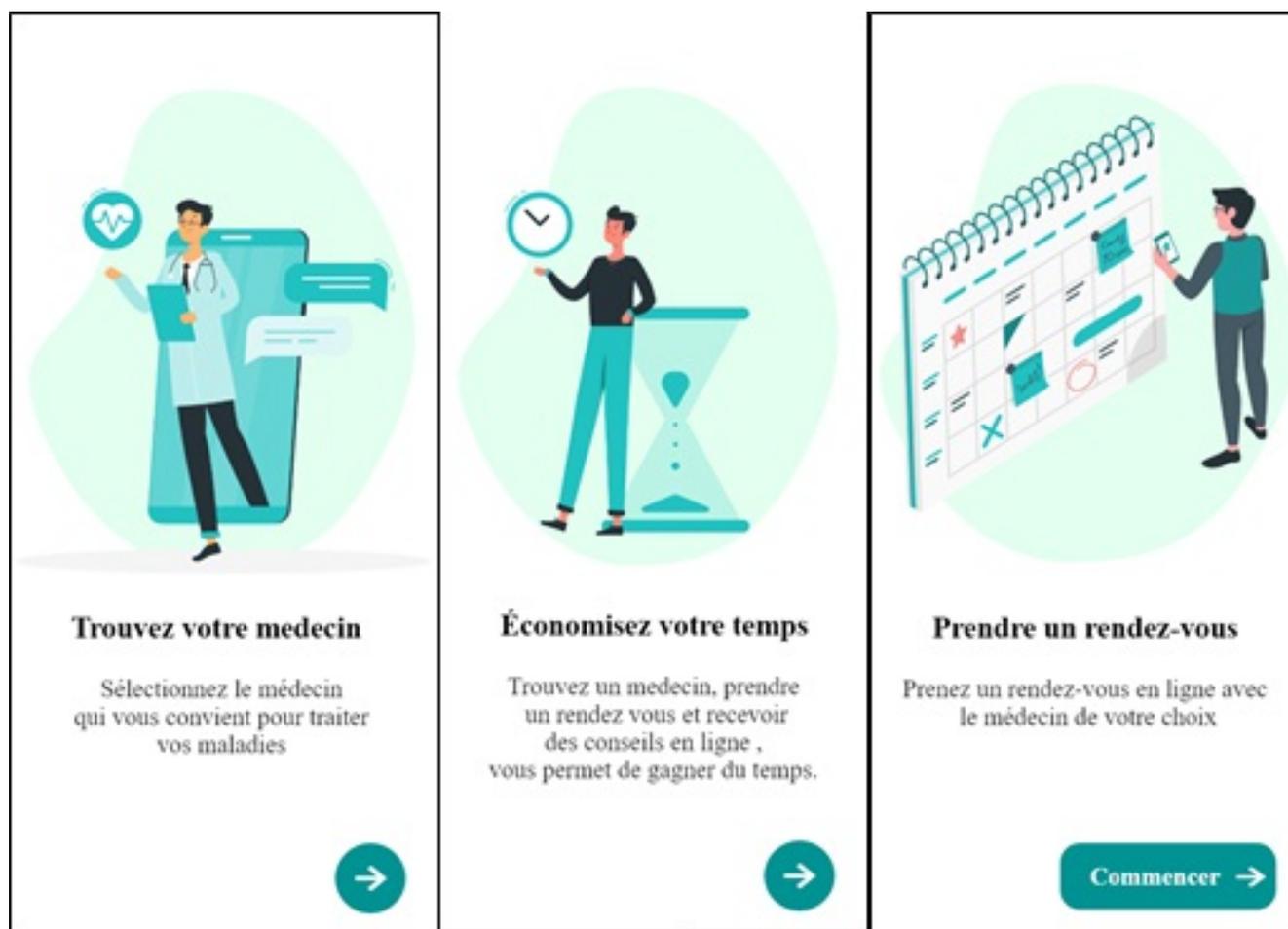


FIGURE 3.17 – Présentation des interfaces de « L'ambording »

3. Interface d'inscription

Cette interface (figure 3.18) contient une suite des champs de saisie que le patient devrait remplir avec ses informations personnelles pour effectuer son inscription.

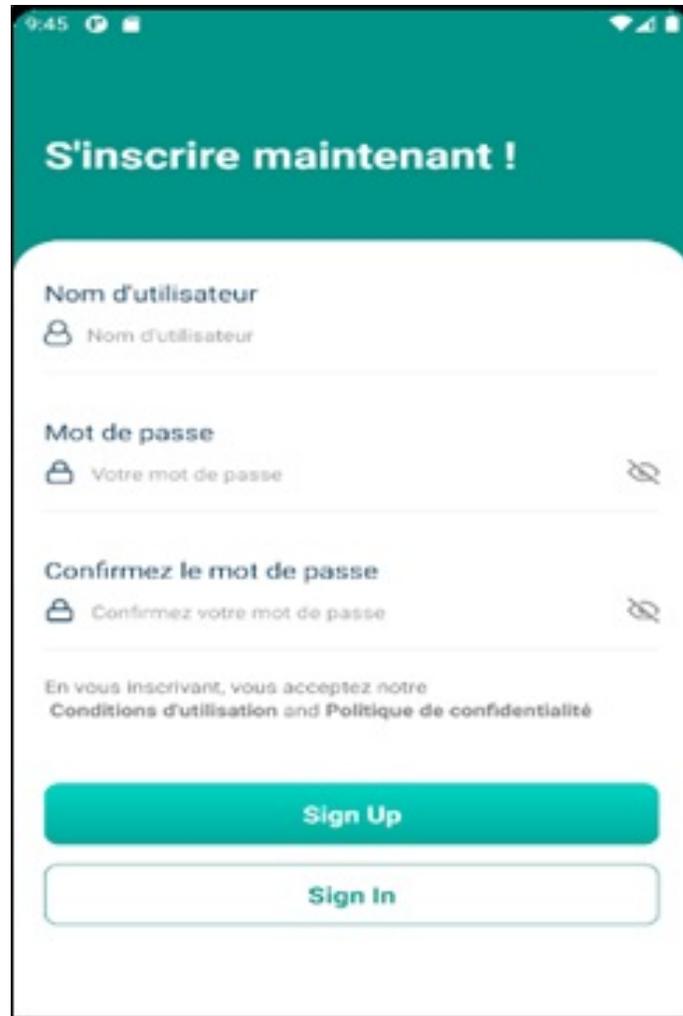


FIGURE 3.18 – Présentation de l'interface « Inscription »

4. Interface d'authentification

Une interface (figure 3.19) qui permet au patient de se connecter à son compte. Pour cela, ce dernier doit saisir son email et son mot de passe pour pouvoir y accéder à l'interface d'accueil.

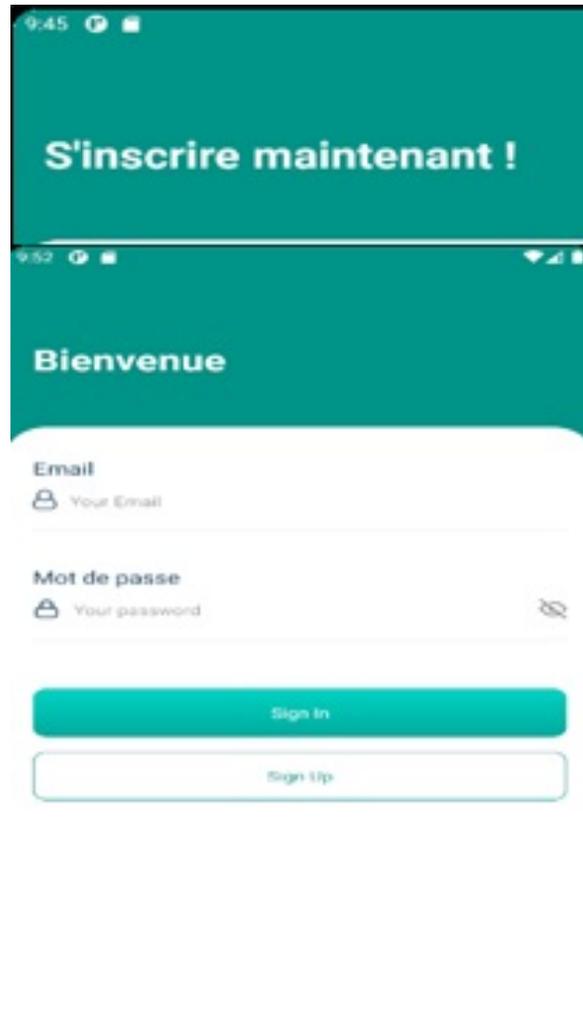


FIGURE 3.19 – Présentation de l'interface « Authentification »

5. Interface d'accueil

L'interface d'accueil (figure 3.20) joue un rôle d'une porte qui mène vers les autres interfaces, à l'aide des deux barres de menu vertical et horizontal qui se situe à gauche et en bas de l'écran.

Le patient navigue facilement vers les autres interfaces de plus, il peut chercher un médecin dans la barre de recherche selon la position de ce dernier.

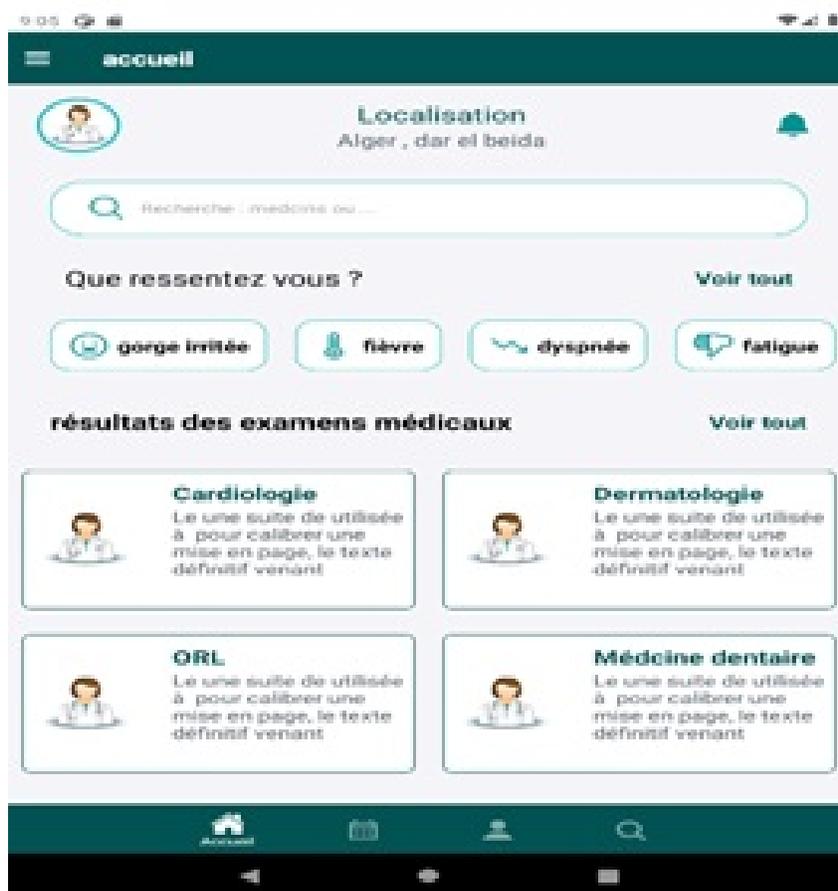


FIGURE 3.20 – Présentation de l'interface « Accueil »

6. Interface Paramètre

Elle permet au patient de modifier les informations personnelles de son compte sur l'application et les valider (figure 3.21).

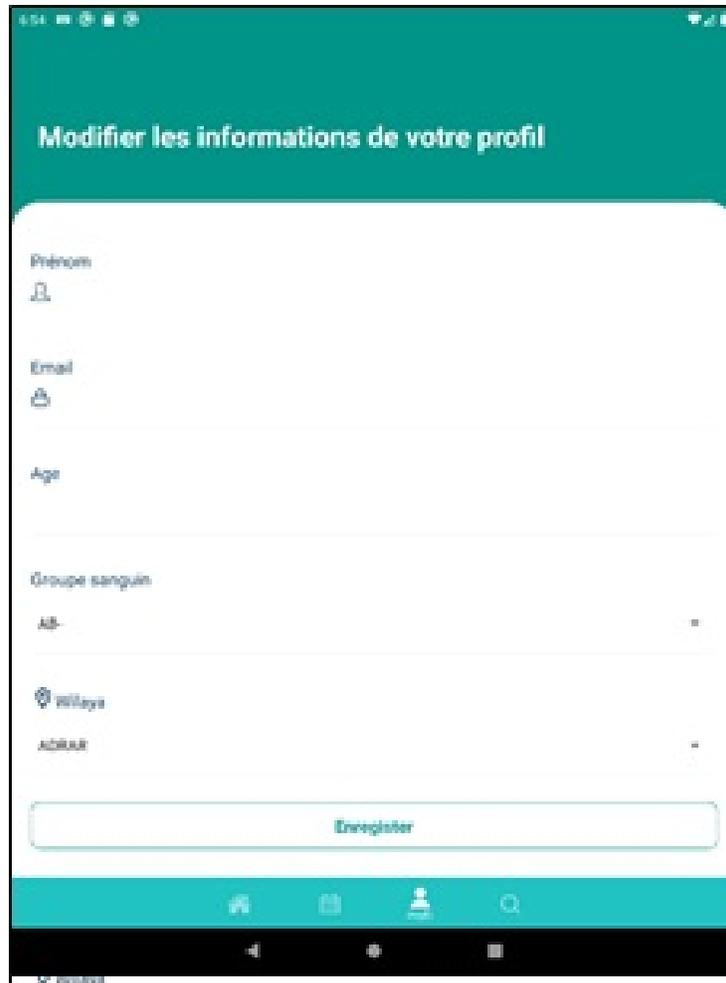


FIGURE 3.21 – Présentation de l'interface « Paramètres »

7. Interface Chercher Médecin

Cette interface (figure 3.22) permet au patient de chercher un médecin. Ce dernier saisie un nom ou une spécialité médicale dans la barre de recherche. Ensuite le système lui affiche une liste des médecins d'où le patient peut sélectionner un seul et consulte son profil.

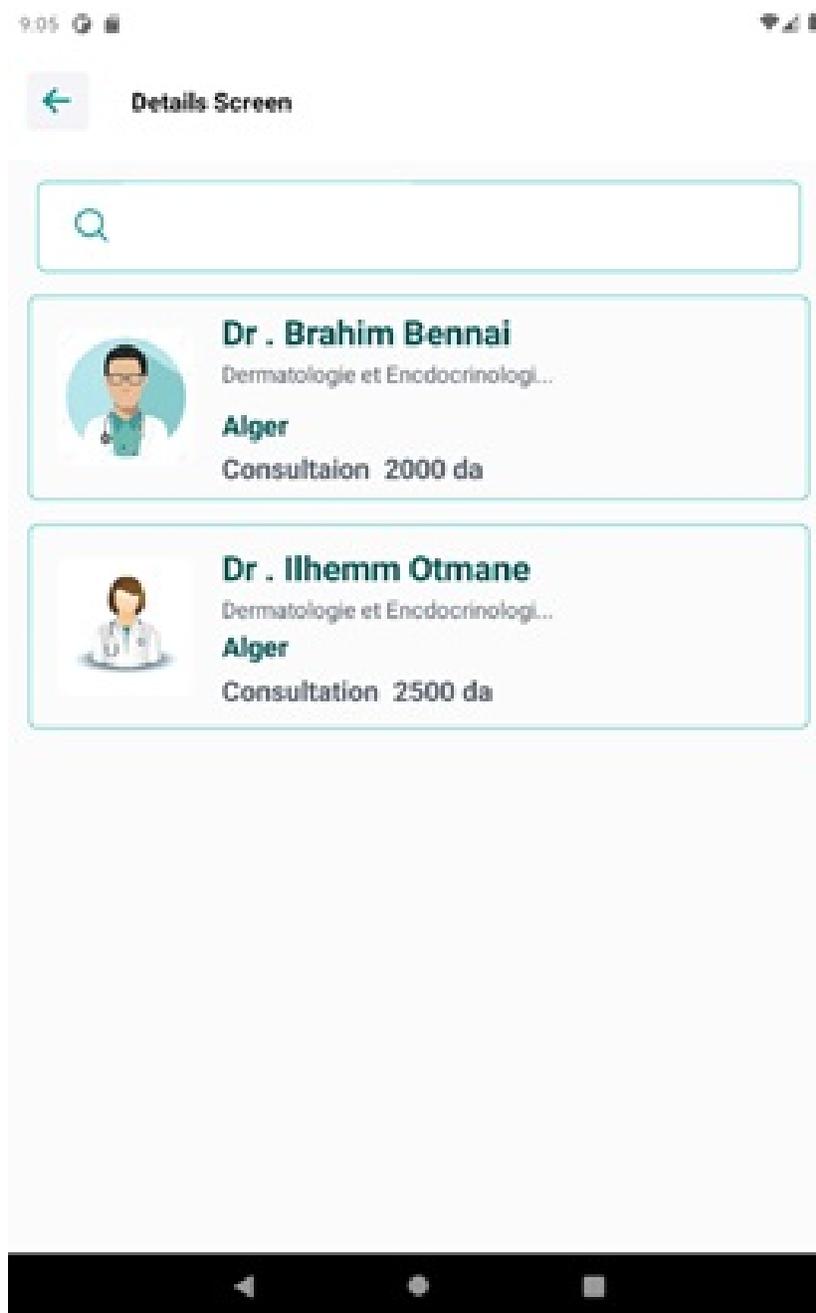


FIGURE 3.22 – Présentation de l'interface « Chercher médecin »

8. Interface profil du médecin

Une interface (figure 3.23) qui affiche les informations détaillées du médecin sélectionné et ensuite cliquer sur prendre un rendez-vous.

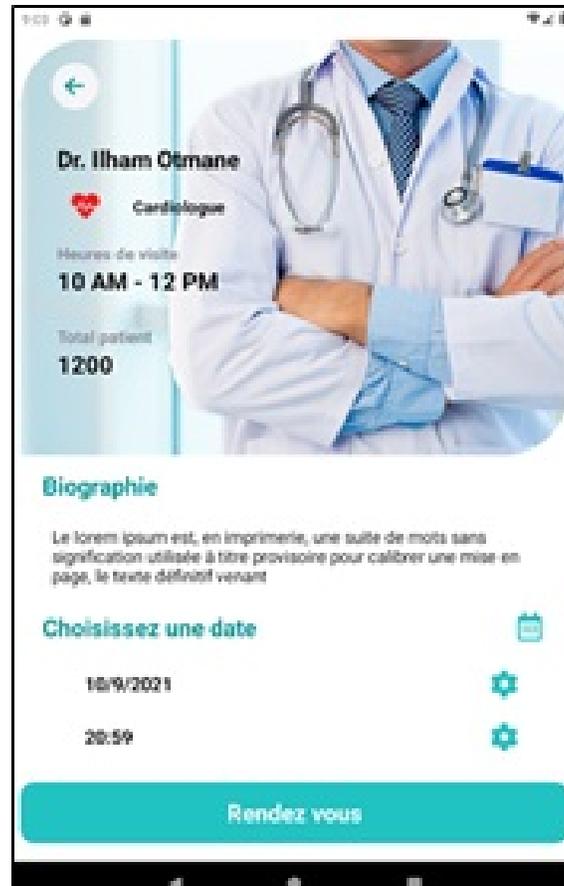


FIGURE 3.23 – Présentation de l'interface « Profil médecin »

9. Interface réservation de rendez-vous

Une interface(figure 3.24) qui donne la possibilité au patient, de choisir un créneau pour la consultation selon parmi les créneaux disponibles du médecin déjà sélectionné et le valider.

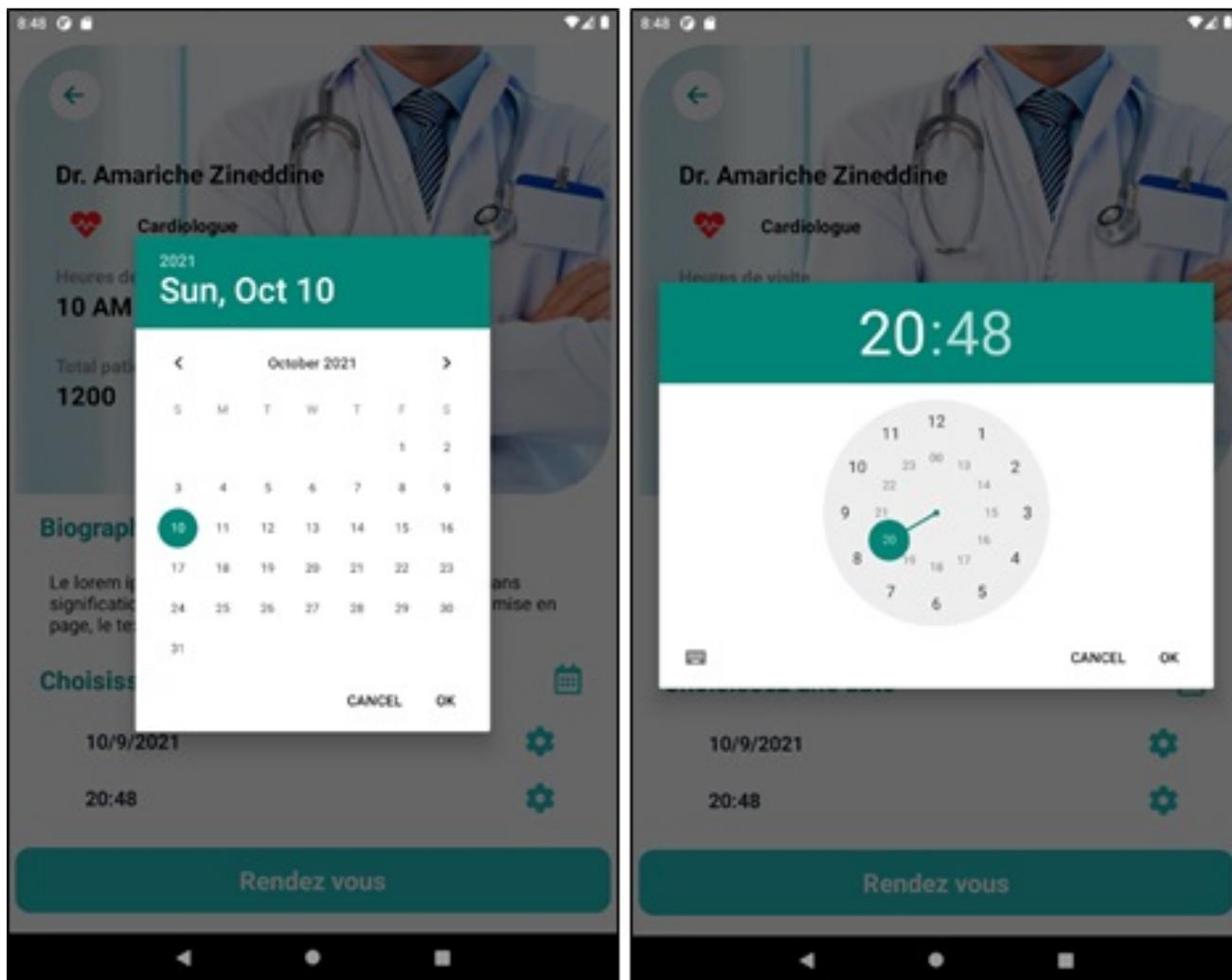


FIGURE 3.24 – Présentation des interfaces « Réservation de rendez-vous »

3.8 Conclusion

Dans ce dernier chapitre, nous avons présenté l'environnement de développement, les langages ainsi que les principaux outils utilisés qui nous ont permis de réaliser nos applications web et mobile.

Nous avons également présenté nos applications à travers une arborescence des vues de ces dernières ainsi qu'à travers les interfaces conçues pour leur utilisation.

Conclusion générale et perspectives

Notre projet de fin d'étude avait, pour but de mettre en place une plateforme de prise de rendez-vous en ligne ; une application mobile de prise de rendez-vous médicale et une application web pour la gestion de ces derniers dans les cabinets médicaux.

Ce document a débuté par une description générale des applications mobiles et web. Nous avons présenté en premier lieu les stratégies préconisées pour le développement de ces applications. Nous avons ensuite décrit les avantages qu'ont apportés ces applications au domaine médical notamment dans la gestion des rendez-vous.

Dans la seconde partie de ce mémoire, nous avons recensé les principales fonctionnalités de la plateforme à réaliser. Ensuite, nous avons présenté les différentes étapes de la conception à commencer par l'analyse des besoins et la spécification des exigences, puis nous avons modélisé les fonctionnalités identifiées à l'issue de cette spécification en utilisant le formalisme UML afin de satisfaire les besoins des utilisateurs finaux de notre plateforme.

De plus, nous avons procédé au choix justifié des technologies avant de passer à l'implémentation, qui nous a permis de développer ces applications en tenant compte de l'architecture matérielle, et de l'environnement logiciel.

Enfin, nous sommes passés à la partie réalisation de notre projet en développant nos deux applications mobiles et web sous la plateforme *Android* avec *react native* et en mettant en œuvre notre base de données relationnelle avec le langage de requête structurée SQL.

Ce projet a fait l'objet d'une expérience à la fois intéressante et enrichissante, qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine du développement et de la conception de systèmes complexes. En outre remédier aux problèmes des rendez-vous nous a honorés en implémentant un service de paiement fonctionnel pour confirmer le rendez-vous.

Des perspectives d'amélioration de notre application restent toutefois indispensables. Nous envisageons ainsi d'ajouter de nouvelles fonctionnalités telles que la téléconsultation, ainsi que l'informatisation des dossiers médicaux qui facilitera le suivi des patients par leurs médecins et qui offre une meilleure gestion des cabinets médicaux.

Résumé

Le mémoire que nous avons présenté, en vue de l'obtention d'un diplôme de master, porte sur la conception et la réalisation d'une plateforme, pour la réservation des rendez-vous médical. Pour implémenter les besoins de notre plateforme, nous nous sommes orientés vers la modélisation qui s'avère être une étape clé de tout travail réussi. Nous avons utilisé le formalisme graphique proposé par UML, qui est un langage basé sur l'orienté objet. Guidés par le processus UP. Nous avons réussi à faire une conception qui nous a facilité la réalisation de notre application. Pour ce qui concerne la réalisation, nous avons pu aboutir à deux applications web et mobile en utilisant des langages de programmation tels que PHP, Laravel, React Native, Javascript, et le serveur local Wampserver. Ces outils dont les performances sont indiscutables nous ont permis d'implémenter la plateforme HealthClic qui répond aux exigences de chaque patient et médecin.

Abstract

The document that we presented in order to obtain a master's degree concerns the design and the implementation of a platform, for booking medical appointments. To implement the requirements specified previously we have turned to modeling which happens to be a key step in any successful work. We used the graphic formalism proposed by UML, the latter is an object-oriented language. Guided by the UP process, we managed to make a good design which made it easier for us to realize our application, in regards to the accomplishment of the platform, we were able to achieve two applications web and mobile using programming languages such as PHP, Laravel, React Native, Javascript, and the local Wampserver server. These tools, whose performance is indisputable, allowed us to implement the platform HealthClic who answers to all the according requirements of each patient and doctor.

Bibliographie

- [1] <https://www.frenchtech-rennes.fr/quels-sont-les-principaux-avantages-de-linternet>. consulté le 25/03/2021.
- [2] Bilal MANSOURI. *Un système de prise de rendez-vous en ligne pour une clinique*. PhD thesis, 2017.
- [3] <http://www.aidice-web.com/accueil/definition-site-web.php>. consulté le 26/03/2021.
- [4] http://fr.wikipedia.org/wiki/Site_Web#.C3.89tapes_de_conception_de_site_web. consulté le 19/04/2021.
- [5] <https://www.ideematic.com/dictionnaire-digital/application-web>. consulté le 27/04/2021.
- [6] <http://www.techno-science.net/?onglet=glossaire&definition=3743>. consulté le 22/04/2021.
- [7] https://fr.wikipedia.org/wiki/Architecture_trois_tiers. consulté le 26/04/2021.
- [8] Rémi Leblond. Vers une architecture n-tiers. *Oral probatoire*, 1999.
- [9] <http://www.commentcamarche.net/contents/221-reseaux-architecture-client-serveur-a-3-> consulté le 26/04/2021.
- [10] <http://www-igm.univ-mlv.fr/~dr/XPOSE2001/perrot/Intro-Comparatif.htm>. consulté le 26/04/2021.
- [11] <https://fr.wikipedia.org/wiki/>. consulté le 30/04/2021.
- [12] François Poirier, Mireille Lehoux, and Louise Briand. *Étude sur les besoins de compétences dans le développement d'applications mobiles*. TechnoCompétences, comité sectoriel de main-d'oeuvre en technologies de l . . . , 2013.
- [13] <http://olivierguillet.com/2012/02/les-dierents-types-dapplications-mobiles-natives-w> Consulté le 30/04/2021.

- [14] <https://fr.wikipedia.org/wiki/application-mobile>. Consulté le 30/04/ 2021.
- [15] <http://www.zdnet.fr/actualites/chires-cles-les-os-pour-smartphones-39790245.htm>. Consulté le 04/05/2021.
- [16] <https://www.electroguide.com/>. consulté le 30/04/2021.
- [17] <https://www.linux.org/>. consulté le 30/04/2021.
- [18] [https://fr.wikipedia.org/wiki/IOS_\(Apple\)](https://fr.wikipedia.org/wiki/IOS_(Apple)). Consulté le 30/04/ 2021.
- [19] <http://www.zdnet.fr/actualites/chires-cles-les-os-pour-smartphones-39790245.htm>. Consulté le 04/05/2021.
- [20] https://www.researchgate.net/publication/321616331_Informatique_medicale_e-Sante_Fondements_et_applications. consulté le 10/05/2021.
- [21] <https://www.hon.ch/Global/pdf/JFIM2011/UtilisationInternetHON.pdf>. consulté le 11/05/2021.
- [22] <https://www.cairn.info/revue-systemes-d-information-et-management-2017-1-page-3.htm>. consulté le 11/05/2021.
- [23] Anis Barka, Abdelkamel Tari, et al. *Conception et réalisation d'une application mobile pour le suivi d'un cabinet médical*. PhD thesis, Universite de bejaia, 2016.
- [24] Arlet Philippe, Nicodème Robert, Delpla Pierre-André, Florent Trapé, Hélène Hanair-Broutin, and Sylvie Cadroy. La relation médecin/malade.
- [25] Bilal MANSOURI. *Un système de prise de rendez-vous en ligne pour une clinique*. PhD thesis, 2017.
- [26] <https://play.google.com/store/apps/details?id=com.boualba.touri&hl=fr&gl=US>. consulter le 16/05/2021.
- [27] <https://www.tabibe.fr/>. consulté le 17/05/2021.
- [28] [<https://developer.mozilla.org/fr/docs/Glossary/MVC>]. consulté le 30/05/2021.
- [29] [<https://www.imagine-developpement.net/developpement-logiciel/application-native>]. consulté le 02/06/2021.
- [30] [https://fr.wikipedia.org/wiki/Application_web]. consulté le 02/06/2021.
- [31] [<https://fr.wikipedia.org/wiki/Applicationhybride>]. consulté le 02/06/2021.
- [32] [<https://fr.wikipedia.org/wiki/Android>]. consulté le 02/06/2021.
- [33] [<https://developer.android.com/about/versions/marshmallow/android-6.0-changes>]. consulté le 03/06/2021..
- [34] [<https://developer.android.com/studio>]. consulté le 03/06/2021.

- [35] [<https://www.jetbrains.com/fr-fr/idea>]. consulté le 03/06/2021.
- [36] [<https://gradle.org>]. consulté le 03/06/2021.
- [37] [<https://code.visualstudio.com>]. consulté le 03/06/2021.
- [38] [<https://getcomposer.org>]. consulté le 03/06/2021.
- [39] [https://www.lebigdata.fr/sql-tout-savoir-guide#Quest-ce_que_SQL]. consulté le 03/06/2021.
- [40] [<https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway>].
- [41] [<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203257-html5-hypertext-markup-langage5-definition-traduction>]. consulté le 04/06/2021.
- [42] [<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203277-css-cascading-style-sheets-definition-traduction>]. consulté le 04/06/2021.
- [43] [<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript>]. consulté le 03/06/2021.
- [44] [<https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/>]. consulté le 04/06/2021.
- [45] [<https://vuejs.org/v2/guide>]. consulté le 04/06/2021.
- [46] [laravel : <https://www.pappleweb.com/index/definition-de-laravel>]. consulté le 04/06/2021.
- [47] [<https://www.web-wave.fr/react-native>]. consulté le 04/06/2021.
- [48] [<https://fr.wikipedia.org/wiki/node.js>]. consulté le 04/06/2021.