

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 d'Informatique



Mémoire de fin d'études

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

Option :
Génie Logiciel

Thème

Conception et réalisation d'une application web de gestion
des rendez-vous d'un cabinet médical

Auteurs :

ALIA Thumar
BOUAOUNI Chifa-Racha

Les jurées :

Dr. BACHIRI Lina Epse KHOUFACHE
Dr. EL BOUHISSI Epse BRAHAMI
Dr. SOUIDAH Lynda

Année universitaire 2021–2022

Remerciements

Tout d'abord, je remercie Allah Tout-Puissant pour ma santé et pour m'avoir donné la volonté et le courage de faire ce travail.

*De plus, je tiens à remercier mon promoteur, **Dr BACHIRI Lina**, pour tous ses précieux conseils, ainsi que les membres du jury qui ont pris le temps de lire notre mémoire et à qui j'ai eu le privilège de le présenter.*

*Je remercie tous les professeurs qui m'ont accompagné durant ma formation. Un grand merci à **mes parents** qui ont joué un rôle énorme afin que je devienne ce que je suis, qui m'ont soutenu tout au long de mes études, que Dieu les bénisse avec une bonne santé afin que je puisse les récompenser.*

*Mes sincères remerciement a mes chères deux frères jumeaux **Younes** et **Lounes**, **mon fiancé** Nasreddine qui m'a encouragé ces dernières années et m'a supporté lors de mes crises d'angoisse et tous mes amis **Walid**, **Fateh**, **Ahmed** et **Boutheyna** qui ont été a mes cotés et qui m'ont soutenu dans les épreuves que j'ai traversées, je leurs souhaite tous le bonheur du monde.*

Et enfin je remercie mon binôme pour sa patience et sa compréhension tout au long de ce travail et a son cousin qui nous a accompagné durant ce projet.

THUMAR

Remerciements

*Je remercie tout d'abord **Allah**, le Très-Haut, qui m'a donné la puissance pour que je puisse terminer ce travail.*

*Je voudrais dans un premier temps remercier mes chers **parents** pour leurs affections inépuisables et leurs précieux conseils. Ils n'ont cessé de prier pour moi durant mon cursus universitaire et m'ont encouragé régulièrement. Je leur dédie ce travail.*

*Je remercie également mon adorable soeur **LAMIS** et mon cher frère **LOTFI** pour leur amour et leur soutien constant.*

*Je tiens remercier, mon promoteur de mémoire **Mme. Lina BACHIRI**, docteur à l'université de Béjaïa pour sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.*

*Je tiens à témoigner toute ma reconnaissance à **M. Abbes BENAYACHE** qui m'a beaucoup appris sur les défis à relever dans le monde de développement web. Il a partagé ses connaissances et expériences dans ce milieu, tout en m'accordant sa confiance et une large indépendance dans le développement des fonctionnalités de l'application.*

Mes remerciements s'étendent à tous mes enseignants du département Informatique de l'Université Abderrahmane Mira de Béjaïa.

Et enfin je remercie mon binôme pour ses efforts, sa patience et sa compréhension tout au long de ce travail.

CHIFA-RACHA

TABLE DES MATIÈRES

Table des matière	
Liste des figures	iii
Liste des tableaux	v
Liste des abréviations	vi
Liste des abréviations	1
Introduction	2
1 Présentation et définitions	3
Introduction	3
1.1 Technologie web	3
1.2 Application web	3
1.3 WEB	4
1.3.1 Définition	4
1.4 Les concepts du web	4
1.4.1 Classification des sites web	5
1.4.2 Les sites web statiques	5
1.4.3 Les site web dynamiques	6
1.5 Architecture client -serveur	6
1.5.1 Les avantages de l'architecture client-serveur sont	7
1.5.2 Les Inconvénients	7
1.5.3 Les caractéristiques d'un serveur	7
1.5.4 Caractéristiques d'un client	7
1.6 Les concepts du web	8
1.7 Définition de cabinet médical	8
1.8 Application web similaire	9
1.8.1 Etude de l'existant au niveau internationale	9

1.8.2	Etude de l'existant au niveau nationale	10
1.9	Choix du processus de développement	10
1.10	Scrum	11
1.10.1	Définition	11
1.10.2	Les rôles dans Scrum	11
1.10.3	Le Backlog produit	12
1.10.4	Le sprint	13
1.11	Méthodologie UML	13
1.11.1	Définitions de base	13
1.11.2	Pourquoi utiliser UML ?	13
1.11.3	Avantage De l'UML :	13
1.11.4	Démarche de modélisation	13
1.11.5	Les diagrammes principaux diagrammes UML	14
	Conclusion	16
2	Analyse et conception	17
	Introduction	17
2.1	Spécification des besoins	17
2.2	Représentation des cas d'utilisation	17
2.2.1	Identification des acteurs	17
2.2.2	Identification des cas d'utilisation (Tâches) de l'application	18
2.2.3	Spécification des tâches	18
2.2.4	Spécification des scénarios	19
2.3	Diagrammes de séquence	25
2.3.1	Définition	25
2.3.2	Diagramme de séquence « s'authentifier »	26
2.4	Diagramme de classe	32
2.4.1	Définition :	32
2.4.2	Le concept de la classe :	32
2.4.3	Le modèle relationnel	33
2.4.4	Les règles de passage :	33
	Conclusion	34
3	Réalisation	35
	Introduction	35
3.1	Le modèle MVC	35
3.1.1	Le Modèle	36
3.1.2	La vue	36
3.1.3	Le Contrôleur	36
3.2	L'environnement de développement	36
3.2.1	Visual Studio Code	36

3.2.2	GitHub	37
3.2.3	Postman	37
3.2.4	Overleaf	37
3.2.5	Visual Paradigm	38
3.3	Développement FullStack	38
3.3.1	BackEnd	38
3.3.2	Frontend	40
3.4	Présentation de quelques interfaces de l'application	42
3.4.1	Interface d'accueil	42
3.4.2	Interface d'inscription	42
3.4.3	Interface d'identification	43
3.4.4	Interface création d'un profil médecin	45
3.4.5	Interface d'affichage du profil d'un médecin	45
3.4.6	Interface Rendez-vous	46
3.4.7	Interface liste des medecins	47
3.4.8	Structure du projet	47
3.5	Conclusion	48
	Conclusion	49
	Références	50

TABLE DES FIGURES

1.1	Fonctionnement d'un site web statique	5
1.2	Fonctionnement d'un site web dynamique	6
1.3	schéma architecture client - serveur	6
1.4	Fonctionnement d'une application web [11]	8
1.5	la méthode scrum.[16]	11
1.6	Les trois composants du langage UML	14
1.7	la liste des neuf principaux diagrammes UML	15
2.1	identificateur des acteurs	18
2.2	Diagramme général des cas d'utilisation du système.	24
2.3	Diagramme de cas d'utilisation d'un patient	25
2.4	Diagramme de cas d'utilisation d'un médecin	25
2.5	Diagramme de séquence Authentification.	26
2.6	Diagramme de séquence	27
2.7	Diagramme de séquence «Consulter planning des RDV».	28
2.8	Diagramme de séquence « Gestion de la liste des patients ».	29
2.9	Diagramme séquence de cas utilisation «Consulter mon dossier médical».	30
2.10	Diagramme de séquence de cas d'utilisation «Consulter planning RDV ».	31
2.11	Figure – Diagramme de classe globale de notre application	32
3.1	Le modèle MVC [5]	35
3.2	Visual Studio Code	36
3.3	GitHub	37
3.4	Postman	37
3.5	Overleaf	37
3.6	Visual Paradigm	38
3.7	Architecture de l' application	38
3.8	logo nodejs	39
3.9	logo express	39

3.10	logo mongoose	39
3.11	logo MongoDB	40
3.12	logo JavaScript	40
3.13	Logo ReactJs	41
3.14	Logo bootstrap	41
3.15	Interface d'accueil de l'application	42
3.16	Interface d'inscription pour un patient.	43
3.17	Interface d'inscription pour un médecin.	43
3.18	Interface d'identification d'un patient	44
3.19	Interface d'identification d'un médecin	44
3.20	Interface de création d'un profil médecin	45
3.21	Interface d'affichage d'un profil médecin	46
3.22	Interface liste des rendez-vous (patient)	46
3.23	Interface liste des rendez-vous (médecin)	47
3.24	Interface liste des profils médecin inscrits sur l'application	47
3.25	Structure du projet	48

LISTE DES TABLEAUX

2.1	Spécification des tâches pour chaque acteur	19
2.2	Spécification des scénarios pour le médecin	21
2.3	Spécification des scénarios pour la secrétaire	22
2.4	Spécification des scénarios pour le patient	23

LISTE DES ABRÉVIATIONS

ALT	Alternative
API	Interfaces de Programmation d'Applications.
DSDM	Dynamic System Development Method.
FDD	Feature Driven Development.
FTP	File Transfert Protocol.
HTML	HyperText Markup Language.
HTTP	HyperText Transfer Protocol.
JSON	JavaScript Object Notation.
MVC	Model-View-Controller.
NPM	Node.js Package Manager.
OPT	optional.
RDV	Rendez-Vous .
REST	REpresentational State Transfer.
REF	Reference.
RUP	Rational Unified Process .
SMS	Short Message Service.
UML	Unified Modeling Language .
UI	User Interface .
URL	Uniform Resource Locator).
UX	User Experience.
WEB	World Wide Web .
XP	Pair Programming .

Aujourd'hui en Algérie, la prise de rendez-vous médical est souvent plus pénible que la maladie du patient elle-même. un patient doit se lever avant l'aube, afin de pouvoir s'inscrire sur une liste d'attente collée généralement sur les portes des cabinets médicaux. a ce jour, le patient est cerné entre les géras de la mauvaise gestion des rendez-vous et l'incapacité des cabinets médicaux à la prise en charge des besoins des patients.

L'intégration des TIC dans la gestion des cabinets médicaux réduira sans doute les ennuis d'attente des patients et ceux des médecins à se préoccuper de la gestion des listes des inscrits, la gestion de la file d'attente, dans certain cas, assurée par un agent qui sera une charge financière de plus pour le médecin. Ce travail qu'on est en train de réaliser, est justement l'une des solutions qui inclut des réponses aux questions posées précédemment en optant à l'usage et l'intégration des TIC dans la gestion des cabinets médicaux.

En un espace d'une vingtaine d'année, l' internet nous a fait changer notre comportement. dans une réunion, pendant le sommeil, à l'école important ou pas, nous devons être joignables. Le titre de ce projet nous reflète deux sous-titres : conception et réalisation, qui seront deux grands chapitres a détaillé.

-Le chapitre conception va nous éclaircir le plan architectural de l'application ainsi que la méthodologie et l'approche suivi pour concevoir cette architecture.

-Le chapitre réalisation va nous montrer comment et avec quels moyens et outils avons-nous développé cette application. mais avant de commencer a détaillé ces deux grands chapitres, nous devons débiter ce projet par quelques généralités sur les applications web, cela sera utile pour pouvoir se situer dans les chapitres qui vont suivre .

1-Introduction

Dans ce chapitre nous allons voir ce qui concerne les bases du web et son fonctionnement et par la suite nous allons définir le cabinet médical, voir la méthode choisie et comment marche-elle et au final la méthodologie l'UML.

1.1 Technologie web

Nous ne pouvons pas faire des affaires dans le monde numérique sans comprendre le minimum des technologies web modernes, car l'idée ne vaut que la réalisation, et la réalisation implique la technologie. Simplement.

1.2 Application web

Un logiciel d'application hébergé sur un serveur et accessible via un navigateur web. au contraire à un logiciel traditionnel, l'utilisation d'une application web ne nécessite pas une installation. il suffit juste d'accéder à l'application à partir d'un navigateur. La tendance actuelle est de proposer une expérience utilisateur et des fonctionnalités équivalentes à un logiciel installé directement sur un ordinateur. les techniques utilisées pour développer des applications web sont les mêmes que celles utilisées pour créer des sites web [1].

1.3 WEB

1.3.1 Définition

Le web est le nom commun pour le (World Wide Web), un sous-ensemble de l'Internet composé de pages accessibles au navigateur web. De nombreuses personnes supposent que le web est identique à Internet et utilisent ces termes de manière interchangeable. Cependant, le terme Internet fait en fait référence au réseau mondial de serveurs qui rend possible le partage d'informations sur le web. Ainsi, bien que le web constitue une grande partie d'Internet, ils ne sont pas identiques. [17] .

1.4 Les concepts du web

- **Page web** : est une ressource du World Wide Web destinée à être visualisée par les visiteurs à l'aide d'un navigateur web. Techniquement, une page web se compose généralement de documents HTML et d'images.

- **Navigateur web** : C'est un logiciel client qui permet à l'utilisateur de chercher et de consulter des documents sur internet, et d'exploiter les liens hypertextuels qu'ils comportent.

- **HTTP (Hyper Text Transfer Protocol)** : Le plus utilisé des protocoles de communication sur le worldwide web. Il a demandé à un client web d'indiquer quelle page il veut obtenir, et au serveur web de lui répondre en lui donnant cette page.

:

- **URL (Uniform Resource Locator)** : Est une chaîne de caractères respectant un schéma de construction bien précis. Il permet d'atteindre une ressource donnée, en passant par internet. Dans le web cette ressource est un document ou un fragment.

:

- **FTP (File Transfert Protocol)** : Est un Protocole de transfert de fichiers, qui est utilisé pour le transfert de fichiers sur internet.

- **Moteur de recherche** : Un moteur de recherche est un serveur qui indexe plusieurs sites web et vous permettent généralement de rechercher les documents qui vous intéressent à l'aide d'un mot clé [14] .

- **Hyper lien** : Est une référence dans le système hypertexte qui permet le transfert automatique du document consulté au document lié. En particulier, les hyperliens sont utilisés dans le World Wide Web pour permettre un simple clic d'une page Web à une autre. Un lien hypertexte a une source (point de départ) et un point d'arrivée (cible).

- **Hyper Text** : L'hypertexte fait référence à un mot, une phrase ou un morceau de texte qui peut être lié à un autre document ou texte. Les hyperliens textuels et les hyperliens graphiques ont été couverts par

L'hypertexte [13].

- **Site web** : Un site web est un ensemble de pages web interconnectées et accessibles au public qui partagent un nom de domaine. des particuliers, des groupes, des entreprises ou des organisations peuvent créer et maintenir des sites web à diverses fins [15].

1.4.1 Classification des sites web

Nous pouvons classer les sites web selon le type des pages qu'ils offrent comme suite :

1.4.2 Les sites web statiques

Un site web statique est la création d'un site web composé de pages Web statiques. Lorsque le contenu d'une page est fixe, on dit qu'il est statique, il ne change pas quel que soit l'utilisateur qui y accède, contrairement aux pages d'un site Web dynamique qui affichent du contenu en fonction des demandes des utilisateurs.

Un site web statique est moins coûteux qu'un site web dynamique, à moins que le site Web ne soit particulièrement volumineux ou qu'un design soigné ne soit une priorité. Toutes les pages sont personnalisées, ce qui permet d'obtenir de bonnes références. L'historique du site est également conservé intact en ligne, ce qui facilite le travail de référence dans le temps.

Le coût de développement d'un site statique est moins élevé que celui d'un site dynamique. Cependant, les mises à jour requièrent certaines compétences informatiques (langage de programmation et notions de webdesign). Si l'entreprise ne dispose pas de ces compétences, elle devra faire appel à un prestataire externe. des mises à jour fréquentes et un grand nombre de sites web rendront cette surveillance plus difficile à gérer.

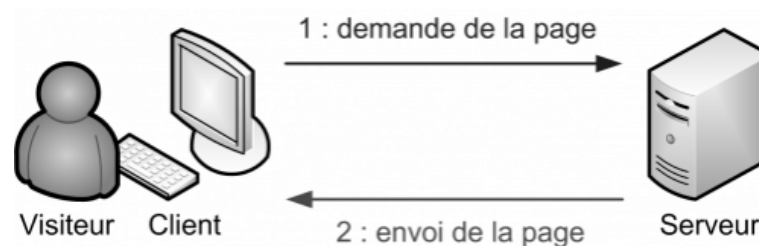


FIGURE 1.1 – Fonctionnement d'un site web statique

1.4.3 Les site web dynamiques

Un site web dynamique est un site Web dont le contenu peut être généré dynamiquement, ce qui signifie que le contenu sera affiché en fonction de l'utilisateur ou d'autres paramètres qui le visitent. Les sites web dynamiques sont créés dans des langages plus complexes que les sites web statiques (ou sites de présentation), y compris l'utilisation de bases de données, qui offrent plus de possibilités de développement que les sites web statiques.

Les avantages d'un site web dynamique

Les mises à jour ne nécessitent pas de connaissances informatiques spécifiques et peuvent être réalisées en interne : pas de risque de dépendance vis-à-vis d'un webmaster.

Les sites web dynamiques sont évolutifs : on peut leur greffer une multitude de modules. ils permettent des fonctionnalités quasi illimitées en termes d'interactivité : newsletters, agendas, paiements en ligne, forums, ... le coût de développement d'un site dynamique est plus élevé que celui d'un site statique. Cependant, les mises à jour étant réalisées au sein de l'entreprise, les économies prennent rapidement le pas sur l'investissement de base.

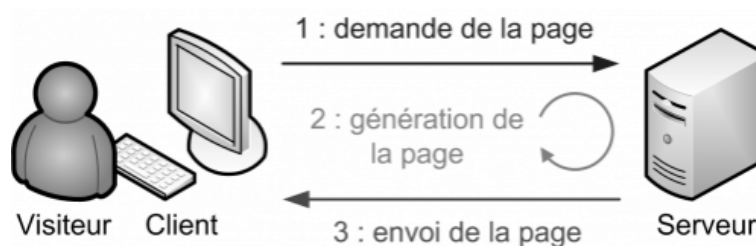


FIGURE 1.2 – Fonctionnement d'un site web dynamique

1.5 Architecture client -serveur

Une architecture client/serveur spécifie un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs postes clients d'un serveur : chaque logiciel client peut envoyer des requêtes au serveur. Un serveur peut être spécialisé en serveur d'application, de fichier, de terminal ou de messagerie électronique.

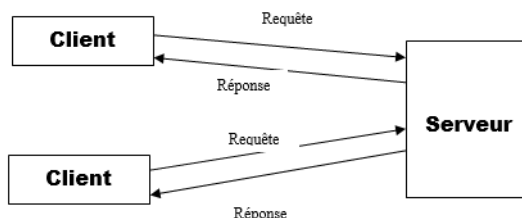


FIGURE 1.3 – schéma architecture client - serveur

1.5.1 Les avantages de l'architecture client-serveur sont

Réduire l'espace mémoire (tous clients vont chercher l'information selon leurs besoins) assurer la cohérence (à chaque instant, le serveur constitue un référentiel unique, similaire pour tous les clients, alors qu'une duplication sur chaque poste client entraîne généralement une divergence entre les différentes versions, et une perte du référentiel commun). Cette garantie de cohérence est considérablement désirable tandis l'information subit des mises à jour. Maîtrise du service (le gestionnaire du serveur gère ce que les programmes clients auront droit à faire ou ne pas faire sur le serveur).

1.5.2 Les Inconvénients

- Le coût d'exploitation élevé (bande passante, câbles, ordinateurs surpuissants) - Lorsqu'un serveur s'interrompt, plus aucun client ne peut accéder aux informations - Le serveur peut supporter une charge limitée. [34] .

caractéristiques du client

1.5.3 Les caractéristiques d'un serveur

- Attend initialement passivement les demandes.
- Écouter et être prêt à répondre aux demandes des clients.
- Lorsqu'une requête lui parvient, il la traite et envoie une réponse.

1.5.4 Caractéristiques d'un client

- Actif en premier.
- Envoie des requêtes au serveur.
- Attendez et recevez une réponse du serveur.
- Le client et le serveur doivent utiliser le même protocole.
- Un serveur peut répondre à plusieurs clients en même temps.

1.6 Les concepts du web

L'utilisateur surf dans l'application Web à partir d'un navigateur Web (client). Mais avant d'accéder à l'interface de l'application, souvent il passe par une page publique. Les demandes de connexion sont ensuite envoyées au serveur Web, qui les transmet au serveur de base de données, puis il vérifie les informations de connexion et enfin renvoie une réponse au serveur web, qui s'affiche sur le logiciel client. C'est le même scénario qui se produit pour toutes les actions effectuées dans l'application Web :

Actions de l'utilisateur au niveau du logiciel client, envoyer une requête au serveur web, interpréter la requête et l'envoyer au serveur de base de données, La réponse est envoyée par le serveur de base de données, l'interprétation de la réponse par le serveur web, Notons enfin l'affichage de la réponse dans le logiciel client.

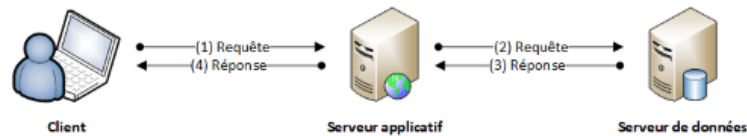


FIGURE 1.4 – Fonctionnement d'une application web [11].

1.7 Définition de cabinet médical

Un cabinet médical est un type d'entreprise où un ou plusieurs médecins diagnostiquent des patients et traitent des maladies, des blessures ou d'autres conditions médicales nécessitant des soins. Dans un cabinet médical, un médecin peut être en mesure d'examiner des patients, d'effectuer des tests, de poser des diagnostics, de proposer des options de traitement, de prescrire des médicaments, etc. [5] .

1.8 Application web similaire

1.8.1 Etude de l'existant au niveau internationale

Doctolib

Doctolib est aujourd'hui le premier site contenant des informations en France sur les professionnels de santé. Une solution en France qui permet au patient la prise de rendez-vous en ligne chez un médecin ou un dentiste à tout moment avec un professionnel de santé et de consulter l'historique de ses consultations depuis un compte personnel. Le patient a la possibilité de gérer son compte, d'annuler ou de modifier son rendez-vous. Au praticien, il permet de gagner du temps, de réduire le taux de rendez-vous non honorés, d'apporter un service nouveau à leur patient. Grâce au chiffrement, Doctolib garantit aux patients, praticiens et secrétaires la propriété, et donc la confidentialité, des informations recueillies sur la plateforme. Ceci a d'autant plus d'importance dans un secteur où la sécurité des données liées au secret médical reste un point d'attention.[7] .

Maiia

Une solution qui permet au patient la prises de rendez-vous en ligne avec un spécialiste de santé, toute les infos utiles pour organiser les rendez-vous : tarifs spécialités, adresse et sms de confirmation et de rappel. Cette prise de rendez-vous en ligne permet un soulagement aux public par une solution de simplification et de rapidité. Maiia offre à tous les patients un profil personnel qui est entièrement sécurisé qui contient l'ensemble des rendez-vous effectués et à venir. Une annulation des rendez-vous plus rapide. La réception des SMS ou email des confirmations et rappels des rendez-vous avec à chaque fois les informations pratiques. Les rendez-vous peuvent être annulés par un SMS. [9] .

Keldoc

Une solution qui permet aux professionnels du médical d'avoir une gestion de leur agenda plus adaptée et efficiente. les rendez-vous se prennent en ligne. un accès direct aux créneaux. Pour empêcher les erreurs de compréhension lors de la prise de rendez-vous. l'implication des SMS de rappel pour éviter les oublis. Ainsi, avec Keldoc les rendez-vous non honorés sont largement diminués. la possibilité de prendre rendez-vous même lorsque le secrétariat est fermé car L'agenda est géré par un télésecrétariat médical professionnel qui appartient aux partenaires de Keldoc.

c'est un outil adéquat à tous les modes d'organisation. un partage des dossiers des patients plus facilement. Il permet aux professionnels de personnaliser leur fiche profil sur Keldoc mais aussi leur sms de rappel. Ce qui leur permet de donner toutes les informations nécessaires pour se rendre au cabinet..[8] .

alldocteur

Allodocteur est une solution en ligne entièrement gratuite qui permet à ses utilisateurs de trouver des praticiens près de chez eux. Du dentiste en passant par le généraliste ou encore le dermatologue, il est très intuitif de prendre un rendez-vous directement sur la plateforme en ligne en seulement quelques clics. Cette solution arrive par un besoin impératif de répondre aux problèmes complexes de prise de rendez-vous pour les patients. Elle permet au patient de gérer ses rendez- Vous (annulation et modification). Tous

les praticiens renseignent directement les informations nécessaires. Cela permet aux utilisateurs de mieux connaître les praticiens et de pouvoir choisir la solution la plus efficace pour répondre à leur problème. [6].

1.8.2 Etude de l'existant au niveau nationale

eSiha

eSiha est une plateforme web et mobile où l'on peut chercher des médecins de toute spécialité désirée, elle est caractérisée par la recherche et la géolocalisation, téléconsultation, la prise de rendez-vous en ligne et un suivi du dossier médical.

- Un seul compte pour toute la famille
- UI/UX adapté pour tous les utilisateurs.[10].

1.9 Choix du processus de développement

Les méthodes Agiles les plus populaires en usage aujourd'hui sont :

- L'extrême Programming (XP),
- Scrum,
- Feature Driven Development (FDD),
- Lean Software Development,
- Agile Unified Process (Agile UP ou AUP)
- Crystal,
- Et Dynamic Systems Development Method (DSDM)

Afin de mettre en place notre projet nous avons choisi la (méthode) SCRUM comme processus de développement, car c'est l'approche la plus efficace pour créer et déployer notre application en équipe afin de gagner en agilité et en flexibilité et accomplir nos buts plus efficacement.[2], [3].

1.10 Scrum

1.10.1 Définition

"Scrum est une méthode agile spécifiée à la gestion de projets. L'objectif est l'amélioration de la productivité des équipes auparavant ralenties par des méthodologies plus lourdes." [16].

Le principe de base est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe d'une à quatre semaines, appelées "Sprints". Chaque sprint possède un but à atteindre, défini par le directeur de produit (appelé aussi le Productowner), à partir duquel sont choisies les fonctionnalités à implémenter dans ce sprint. Un sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel. Pendant ce temps, le Scrum Master a la charge de réduire au maximum les perturbations extérieures et de résoudre les problèmes non techniques de l'équipe.

La figure ci-dessous nous donne une vue globale du processus de développement avec la méthode agile Scrum. Les différents points à prendre en compte sont :

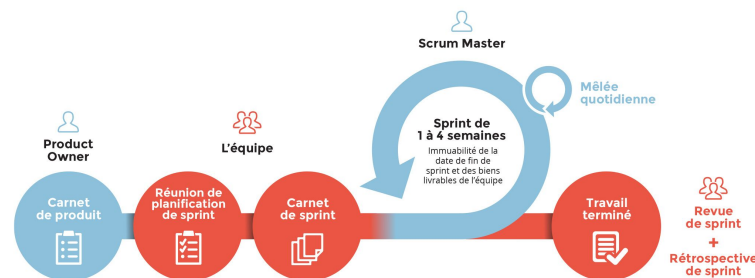


FIGURE 1.5 – la méthode scrum.[16].

1.10.2 Les rôles dans Scrum

En Scrum, comme représenté dans la figure 5, on distingue plusieurs rôles et pour chacun, des responsabilités et des tâches à accomplir :

Le Product Owner

c'est la personne représentant le client Le Product Owner :

- Représente toutes les personnes qui ont un intérêt dans le logiciel qui sera produit.
- Financer le projet.
- Est responsable de l'établissement des besoins initiaux du projet qui sont listés dans Lebacklog produit.
- A des objectifs de retour sur investissement.
- Décide des releases, qui veut dire des versions qui seront mises à disposition des Utilisateurs.
- Responsable de la m à j du backlog produit. Il doit classer régulièrement les besoins exprimés dans le backlog produit pour s'assurer que les fonctionnalités qui généreront le plus de valeur pour le logiciel seront implémentées en priorité. Il doit également mettre à jour les exigences en fonction des incréments de logiciel qu'il teste à chaque fin de sprint.

Scrum Master

Ce rôle apparaît simple mais il est en réalité délicat à appréhender car il diffère énormément de la culture projet traditionnelle. un scrum master n'est pas un chef de projet. Il n'a pas d'autorité hiérarchique sur l'équipe. Il représente un rôle de coach, et de facilitateur. Un catalyseur de projet en quelque sorte. Son rôle est de :

- L'isolation de l'équipe des perturbations extérieures en cours de sprint.
- Apprendre la méthodologie Scrum à l'équipe.
- l'assurance que l'équipe respecte parfaitement la méthode : que chacun fait bien du Scrum et ne diverge pas vers d'vieilles méthodes connues et maîtrisées.
- Répondre aux exigences de l'équipe.
- Faire du scrum dans la culture d'entreprise.
- Travailler avec le Productowner pour sélectionner avec lui les besoins apportant le plus de valeur ajoutée au logiciel. la réussite du projet est liée a lui, au même titre que l'équipe.

L'équipe

Les responsabilités et le pouvoir de décision sont transférés du chef de projet (dans les Organisations traditionnelles) à l'équipe. Ainsi l'équipe :

- Développe les fonctionnalités : elle transforme les besoins en incréments fonctionnels de logiciel.
- Est autogérée et auto organisée : elle décide comment aboutir au mieux à produire le logiciel à partir des exigences précisées par le Productowner.
- Elle est multidisciplinaire. Si les spécialités persistent, elles sont toutes présentes au sein de l'équipe. Les méthodes d'estimation supposent que les personnes ne sont pas spécialisées et que chacun est capable de remplir toutes les tâches du backlog de sprint.
- Responsable collectivement de la réussite de chaque itération. Toutes les cartes sont entre ses mains pour battre le record.

1.10.3 Le Backlog produit

Le backlog produit est un des éléments nécessaire de Scrum. Il s'agit d'un catalogue qui recense toutes les fonctionnalités désirées sur le projet. Il est créé par le Productowner. Les entrées sont représenté dans les termes du client, qui veut dire en termes métier. Le Productowner priorise le backlog en fonction des éléments qui représentent le plus de valeur pour lui en assignant un chiffre qui représente l'importance de chaque élément. Plus le chiffre augmente, plus le besoin est élevée à ses yeux et doit être développé en avantagé par l'équipe.

1.10.4 Le sprint

Un sprint est une période fixe durant lequel le système est développé par l'équipe. Scrum ne recommande pas une durée précise pour le sprint . La durée est de 2 à 4 semaines. C'est à l'équipe de choisir une durée qui lui convient. Une fois qu'elle est choisie, il est primordial que cette durée devienne un délai standard de tous les sprints. le sprint couvre plusieurs phases :

- La réunion de planification du sprint se fait au départ de chaque sprint, pour fixer et organiser les objectifs de ce sprint.
- Se poursuit par le sprint en soi : Pendant lequel se font les développements et les tests avec une réunion quotidienne qui ne dépasse pas les 15 minutes. Pendant ce temps, Le Productowner peut faire évoluer son backlog de produit.
- En fin de sprint ont lieu la revue de sprint pendant laquelle l'équipe présente le système produit et enfin la rétrospective du sprint pour essayer d'enrichir le processus de développement.

1.11 Méthodologie UML

UML (Unified Modeling Language), que l'on peut traduire par « langage de modélisation unifié » est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, et est devenu désormais la référence en termes de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

Le méta-modèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relie. Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues qui sont constituées d'un ou plusieurs diagrammes.

1.11.1 Définitions de base

Acteur : C'est un rôle que peut jouer une personne ou une chose dans un système. L'acteur peut interagir avec un cas d'utilisation, et participer à son scénario. Il est donc dit un objet actif.

Besoins : les besoins recouvrent l'ensemble de tout ce qui apparaît « être nécessaire » aux acteurs du système.

L'Ultra Narrow Band : qui consiste à transmettre les signaux dans une gamme de fréquence la plus étroite possible.

L'étalement de spectre : qui revient à étaler le spectre sur une large bande de fréquences avec une puissance d'émission très faible.

1.11.2 Pourquoi utiliser UML ?

Aujourd'hui, UML représente un standard de modélisation, il permet grâce à sa représentation graphique, d'exprimer visuellement une solution objet, de faciliter la comparaison et l'évolution des solutions, son caractère polyvalent et sa souplesse en font un langage universel. [35] .

1.11.3 Avantage De l'UML :

- Descriptions graphiques. - Vues différentes à des étapes différentes. - Adaptation facile aux méthodes.
- Projet : un bon outil de démarrage du projet.

1.11.4 Démarche de modélisation

Le langage UML propose 9 diagrammes répartis sur les trois axes du niveau conceptuel : fonctionnel, structurel et temporel. La figure 6 représente les trois axes de modélisation du langage UML.

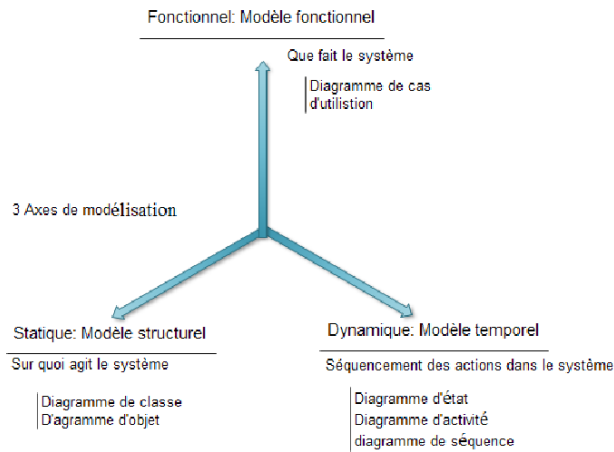


FIGURE 1.6 – Les trois composants du langage UML

Toutes les vues proposées par UML sont complémentaires, elles permettent de mettre en évidence les différents aspects du logiciel à réaliser. On peut organiser une présentation d’UML autour d’un découpage en vues, ou bien différents diagrammes, selon qu’on sépare plutôt les aspects fonctionnels des aspects architecturaux, ou les aspects statiques des aspects dynamiques.[18].

La vue fonctionnelle

Elle est représentée à l’aide de diagramme de cas d’utilisation et de séquences. Elle cherche à appréhender les interactions entre les différents acteurs/utilisateurs et le système, sous forme d’objectifs à atteindre d’un côté et sous forme de séquence d’interactions de l’autre.[18].

La vue structurelle, ou statique

Il réunit les diagrammes de classes et les diagrammes de packages. Les premiers favorisent la structuration des données et tentent d’identifier les objets composants constituant le programme, les attributs, opérations et méthodes, ainsi que les liens et les associations qui les unissent. Le second s’attache à regrouper les classes fortement liées entre elles en des Composants les plus autonomes possibles. A l’intérieur de chaque package, on trouve un diagramme de classe. [18].

La vue dynamique

Qui est exprimée par le diagramme d’État. Cette vue est plus algorithmique et orientée « traitements », elle vise à décrire l’évolution (la dynamique) des objets complexes du programme tout au long de leur cycle de vie. De leur naissance à leur mort, les objets voient leur changement d’état guidé par les interactions avec les autres objets. Dans notre conception, on a retenu le diagramme des cas d’utilisation pour présenter l’aspect fonctionnel de notre système, le diagramme de séquence pour illustrer son aspect dynamique et le diagramme de classe pour présenter la structure de l’application.[18].

1.11.5 Les diagrammes principaux diagrammes UML

La figure suivante présente la liste des neuf principaux diagrammes UML. [33].

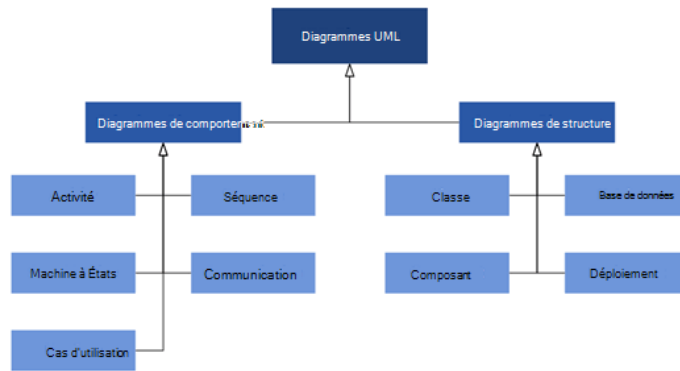


FIGURE 1.7 – la liste des neuf principaux diagrammes UML

Ci-dessous la présentation du contenu de chaque type de diagramme :

Diagrammes de classes

Ce diagramme représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système.

Diagramme d'objets

Le diagramme d'objets permet la représentation d'instances des classes et des liens entre Instances.

Diagramme de cas d'utilisation

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système.

Diagramme de séquence

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges.

Diagramme de communication

Le diagramme de communication constitue une autre représentation des interactions que celle du diagramme de séquence. En effet, le diagramme de communication met plus l'accent sur l'aspect spatial des échanges que l'aspect temporel.

Diagramme d'états-transitions

Les diagrammes d'états-transitions décrivent le comportement interne d'un objet à l'aide d'un automate à états finis.

Diagramme d'activités

Les diagrammes d'activités décrivent le comportement d'une méthode, le déroulement d'un cas d'utilisation et/ou les enchaînements d'activités. Une activité désigne une suite d'actions. Le passage d'une action vers une autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une action et provoquent le début immédiat d'une autre action (elles sont automatiques). Chaque action est représentée par un rectangle dont les coins sont très arrondis. Chaque action est libellée pour décrire ce qui est fait.

Diagramme de collaboration UML

Un diagramme de collaboration est utilisé pour décrire une collection d'objets qui interagissent pour mettre en œuvre un certain comportement dans un contexte. Il est utilisé pour modéliser la fonctionnalité du système, c'est-à-dire, plus précisément, visualiser la relation entre les objets qui collaborent pour effectuer une tâche particulière et modéliser la logique de la mise en œuvre pour une opération complexe.

Diagramme de déploiement UML

Un diagramme de déploiement est un type de diagramme utilisé en UML pour décrire les composants matériels utilisés dans les implémentations de systèmes ainsi que les environnements d'exécution et les artefacts déployés sur le matériel. Il permet de visualiser le système de topologie du matériel, de modéliser les éléments matériels physiques et la relation de communication entre eux, et de planifier l'architecture du système.

Conclusion

Ce chapitre a été consacré à la présentation générale du web, l'architecture client/serveur et son fonctionnement, nous avons vu qu'est-ce que c'est un cabinet médical et quelques applications web semblables à notre futur système. D'autre part, on définit SCRUM et son cycle de vie après avoir cité les raisons de la choisir comme méthode et enfin on a parlé du langage UML et de ses diagrammes. Dans le chapitre suivant nous allons parler de la capture des besoins en études quelques scénarios (diagrammes, etc...).

Introduction

La conception de toute solution logicielle doit être traitée avec précision et détail, précédée d'une analyse bien approfondie et réfléchie, dans ce chapitre nous allons présenter les exigences fonctionnelles et non fonctionnelles, identifier les acteurs interagissant avec notre système ainsi que des diagrammes de cas d'utilisation et de séquence et on finira avec le diagramme de classe et le modèle relationnel.

2.1 Spécification des besoins

Cette étape permet de définir les besoins et les objectifs du système à concevoir ainsi que les différents cas d'utilisation existants. Le modèle résultant de cette étape nous donne le diagramme de cas d'utilisation et les séquences de chacun de ces derniers.

2.2 Représentation des cas d'utilisation

2.2.1 Identification des acteurs

Un acteur correspond au rôle d'une entité externe (utilisateur humain, dispositif matériel, ou autre système) qui interagit directement avec le système étudié, par l'échange de l'information (en entrée et en sortie), on trouve les acteurs en observant les utilisateurs directs du système, les agents de maintenance et les autres systèmes qui interagissent avec lui.





FIGURE 2.1 – identificateur des acteurs

2.2.2 Identification des cas d'utilisation (Tâches) de l'application

Un cas d'utilisation est une unité cohérente d'une fonctionnalité visible de l'extérieur l'initie. Un cas d'utilisation modélise ainsi un service fourni par le système, sans imposer le mode d'exécution de ce service. Chaque acteur que nous avons défini accomplit plusieurs tâches.

2.2.3 Spécification des tâches

Pour chaque acteur identifié précédemment, il convient de rechercher les différentes intentions selon lesquelles il utilise le système.

Acteur	Cas d'utilisation (Tâches)
Médecin	T1 : Authentification. T2 : Gérer la liste des patients T3 : Gestion dossier médical. T4 : Gestion des comptes des personnels médicaux
Secrétaire	T1 : Authentification. T2 : Gérer la liste des patients. T3 : Gestion des rendez-vous
Patient	T1 : Authentification. T2- Prendre des rendez-vous T3- Consulter son historique T4- consulter son carnet de santé

TABLE 2.1 – Spécification des tâches pour chaque acteur

2.2.4 Spécification des scénarios

Afin d'effectuer une tâche associée à un acteur, celui-ci doit effectuer un certain nombre d'actions. On appelle cet enchaînement un scénario.. Le tableau suivant rassemble tous les scénarios pouvant être déclenchés par les différents acteurs.

Pour le médecin

Tâches	Scénarios
T1 : Authentification.	S1 : saisir le login et le mot de passe et valider.
T2 : Gérer la liste des patients	<p>S2 :</p> <ol style="list-style-type: none"> 1 - L'utilisateur clique sur le bouton "Gérer le patient". 2- Le système affiche l'interface correspondante (liste des patients).. 3- L'utilisateur recherche les patients par son nom. [si le patient n'existe pas] <p>S3 : Ajouter un patient :</p> <ol style="list-style-type: none"> 1- L'utilisateur demande d'ajouter un patient. 2- Le système affiche le formulaire à ajouter. 3- L'utilisateur remplit le formulaire d'ajout et valide. 4- Le système vérifie les données saisies et affiche un message d'erreur. Sinon, le message « Ajouté avec succès » s'affichera. <p>[Si la personne existe] :</p> <ul style="list-style-type: none"> - L'utilisateur sélectionne le patient. <p>S4 : Modifier le patient :</p> <ol style="list-style-type: none"> 1- L'utilisateur demande une modification. 2- Le système affiche les informations individuelles du patient. 3- L'utilisateur modifie ses données personnelles et les valide. 4- Le système affiche un message « Modification avec succès » <p>S5 : Afficher les informations sur le patient. :</p> <ol style="list-style-type: none"> 1- L'administrateur demande de consulter les informations patient 2- Le système affiche les données correspondantes 3- L'utilisateur valide la suppression. 4- Le système affiche un message « supprimer avec succès» <p>S6 : Supprimer patient :</p> <ol style="list-style-type: none"> 1- L'utilisateur demande la suppression. 2- Le système demande de confirmer la suppression. 3- L'utilisateur valide la suppression. 4- Le système affiche un message « supprimer avec succès»
T3 : Gestion dossier médical.	<p>S7 :</p> <ol style="list-style-type: none"> 1- Le médecin demande la gestion des dossiers médicaux. 2- Le système affiche liste des patients. 3- Le médecin recherche patient par son nom. <p>[s'il n'existe aucun patient ;]</p> <ul style="list-style-type: none"> - Un message d'erreur est envoyé au système. <p>[Si existe]</p> <ul style="list-style-type: none"> - Le médecin sélectionne le patient. <p>Remplir fiche médicale :</p> <ol style="list-style-type: none"> 1- Le Médecin demande l'ajout de fiche médicale. 2- Le système affiche le formulaire fiche médicale. 3- Le médecin remplit le formulaire puis le valide. 4- Le système affiche un message «fiche ajoutée ». <p>S8 : Ajouter ordonnance :</p> <ol style="list-style-type: none"> 1- Le Médecin demande l'ajout « Ajouter ordonnance ». 2- Le système affiche le formulaire ordonnance. 3- Le médecin remplit le formulaire et le valide. 4- Le system affiche message ordonnance ajoutée. <p>S9 : Consulter / Modifier fiche médicale</p> <ol style="list-style-type: none"> 1- Le Médecin demande de Consulter / Modifier la fiche médicale. 2- Le système affiche le formulaire Consulter / Modifier fiche médicale. 3- Le médecin effectue les modifications de fiche médicale et valide. 4- Le système affiche message fiche médicale modifié.

	<p>S10 : Consulter / Modifier ordonnance</p> <ol style="list-style-type: none"> 1- Le Médecin demande de Consulter / Modifier ordonnance. 2- Le système affiche le formulaire Consulter / Modifier ordonnance. 3- Le médecin effectue les modifications de résultats analyse et valide. 4- Le système affiche message ordonnance modifié Le médecin accède au dossier du patient souhaité puis introduit des modifications avec succès. <p>Le médecin et l'infirmier ajoutent des notifications avec succès.</p>
<p>T4 : Gestion des comptes des personnels médicaux.</p>	<p>S11 :</p> <ol style="list-style-type: none"> 1- Le médecin clique sur le bouton Administration (Patient/Secrétaire). 2- Le système affiche l'interface correspondante (liste des personnes). 3- Le médecin recherche les patients par son nom. <p>[s'il n'existe aucun patient ;]</p> <p>S12 : Ajouter des personnes :</p> <ol style="list-style-type: none"> 1- Le médecin demande du personnel supplémentaire. 2- Le système affiche le formulaire Ajouter. 3- Le médecin remplit un formulaire supplémentaire et vérifie. 4- Le système valide les données saisies et affiche une erreur en cas d'erreur, sinon le système affiche le message "Ajouté avec succès". <p>[Si la personne existe] : - Le médecin choisit le personnel.</p> <p>S13 : Modifier personnel :</p> <ol style="list-style-type: none"> 1- Le médecin demande une modification. 2- Le système affiche des informations personnelles. 3- Le médecin modifie les données du personnel et les valide. 4- Le système affiche un message de modification réussie. <p>S14 : Demander des informations personnelles</p> <ol style="list-style-type: none"> 1- Le médecin demande l'accès aux renseignements personnels. 2- Le système affiche les données correspondantes. <p>S15 : Supprimer des personnes :</p> <ol style="list-style-type: none"> 1- Le médecin demande la suppression d'un personnel. 2- Le système demande de la confirmation de suppression. 3- Le médecin valide la suppression. 4- Le système affiche le message de suppression avec succès.

TABLE 2.2 – Spécification des scénarios pour le médecin

Pour la secrétaire

Tâches	Scénarios
T1 : Authentification.	S1 : saisir le login et le mot de passe et valider.
T2 : Gérer la liste des patients.	<p>S2 :</p> <ol style="list-style-type: none"> 1- L'utilisateur clique sur le bouton «Gérer patient». 2- Le système affiche l'interface correspondante (liste patient). 3- L'utilisateur cherche le nom du patient. [s'il n'existe aucun patient ;] <p>S3 : Ajouter personnel :</p> <ol style="list-style-type: none"> 1- L'utilisateur souhaite ajouter un patient. 2-Le système utilise le formulaire d'ajout. 3- L'utilisateur remplit le formulaire « Ajout et validation ». 4- Le système vérifie les données introduites, et affiche l'erreur en cas d'erreur, sinon il affiche un message « ajouter avec succès ». <p>[Si personnel existe] :</p> <ul style="list-style-type: none"> - L'utilisateur sélectionne le patient. <p>S4 : Modifier patient :</p> <ol style="list-style-type: none"> 1- L'utilisateur demande la modification. 2- Le système affiche des information personnelles 3- L'utilisateur modifie les données de personnel et valide. 4- Le système affiche un message «modifier avec succès». <p>s5 :Consulter les informations patient :</p> <ol style="list-style-type: none"> 1- L'administrateur demande «Consulter les informations patient». 2- Le système affiche les données correspondantes. <p>S6 : Supprimer patient :</p> <ol style="list-style-type: none"> 1- L'utilisateur demande la suppression. 2- Le système demande de confirmer la suppression. 3- L'utilisateur valide la suppression. 4- Le système affiche un message « supprimer avec succès ».
T3 : Gestion des rendez-vous.	<p>S7 :</p> <ol style="list-style-type: none"> 1- Le secrétaire médical demande la gestion de planning RDV. 2- Le système affiche la liste des RDV. <p>S7 : Ajouter RDV :</p> <ol style="list-style-type: none"> 1- Le secrétaire médical demande l'ajout des RDV. 2- Le système affiche liste des patients. 3- Le secrétaire sélectionne le patient et demande d'ajouter son RDV. 4- Le système affiche le formulaire Ajouter. 5- Le secrétaire remplit le formulaire et le valide. 6- Le système vérifie les données introduit, si un champ manquant alors le système affiche un message d'erreur, et la secrétaire réintroduit les données. 7- Le système effectue la vérification de la disponibilité de l'heure RDV. -Si l'heure du RDV disponible est disponible alors le système affiche message RDV et l'ajoute sur la liste RDV.. <p>S8 : Supprimer RDV :</p> <ol style="list-style-type: none"> 1- Le secrétaire médical sélectionne le RDV, clique sur supprimer RDV. 2- Le système demande que la suppression soit confirmée. 3- Le secrétaire médical confirme la suppression. 4- Le système affiche un message de suppression avec succès.

TABLE 2.3 – Spécification des scénarios pour la secrétaire

Tâches	Scénarios
T1 : Authentification.	S1 : -Saisir son login et son mot de passe .
T2-Prendre rendez-vous	S2 : -Choisir la date, l'heure du rdv et valider
T3-Consulter son historique	S3 : -Consulter sa liste des rendez-vous déjà pris
T4-consulter son carnet de santé	S4 : -Visualiser les prestations déjà faites et les ordonnances

TABLE 2.4 – Spécification des scénarios pour le patient

2.2.2.4 Diagramme des cas d'utilisation

Comme son nom l'indique, une use case représente une fonctionnalité du système. Ce diagramme nous permet de cerner toutes les possibilités d'interaction entre le système et les acteurs, ce qui signifie toutes les fonctionnalités que le système doit fournir. Nous rassemblons tous les cas d'utilisation précédemment identifiés, dans un diagramme général comme sur la Figure

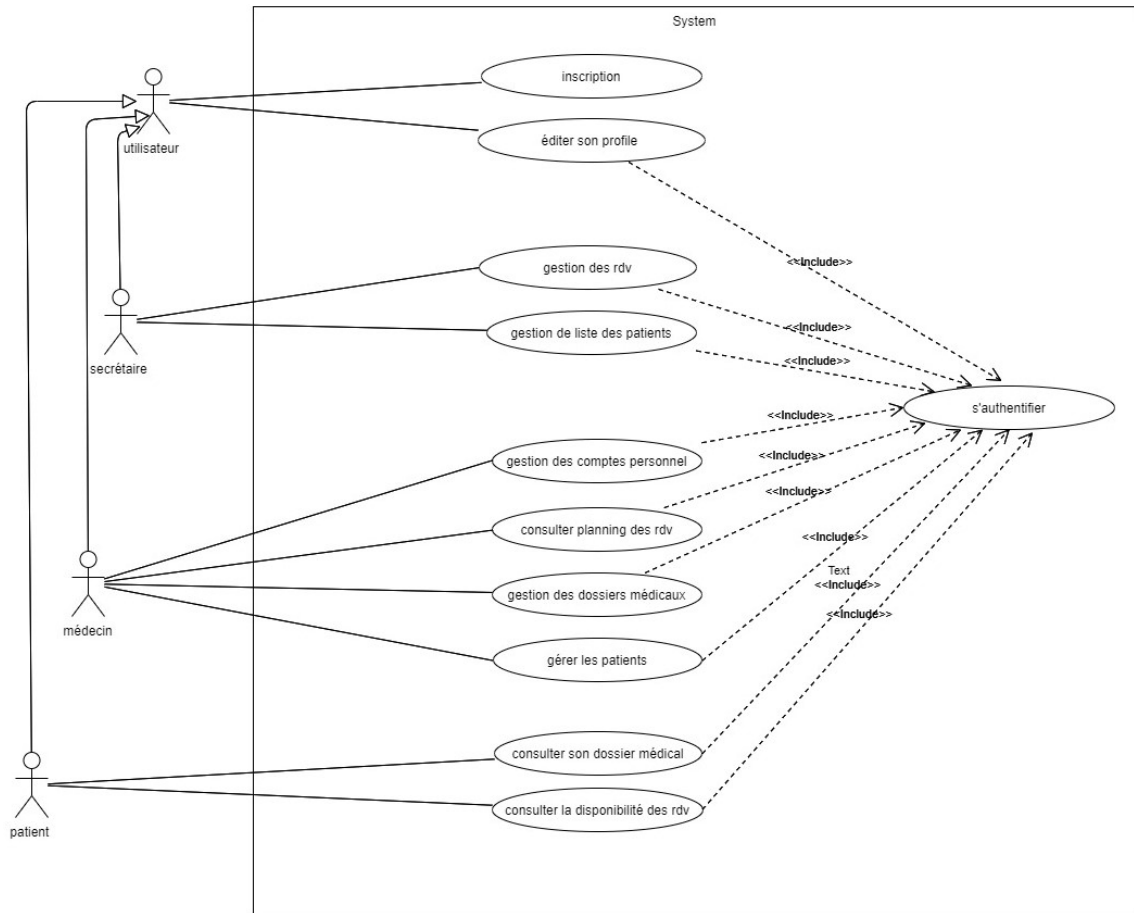


FIGURE 2.2 – Diagramme général des cas d'utilisation du système.

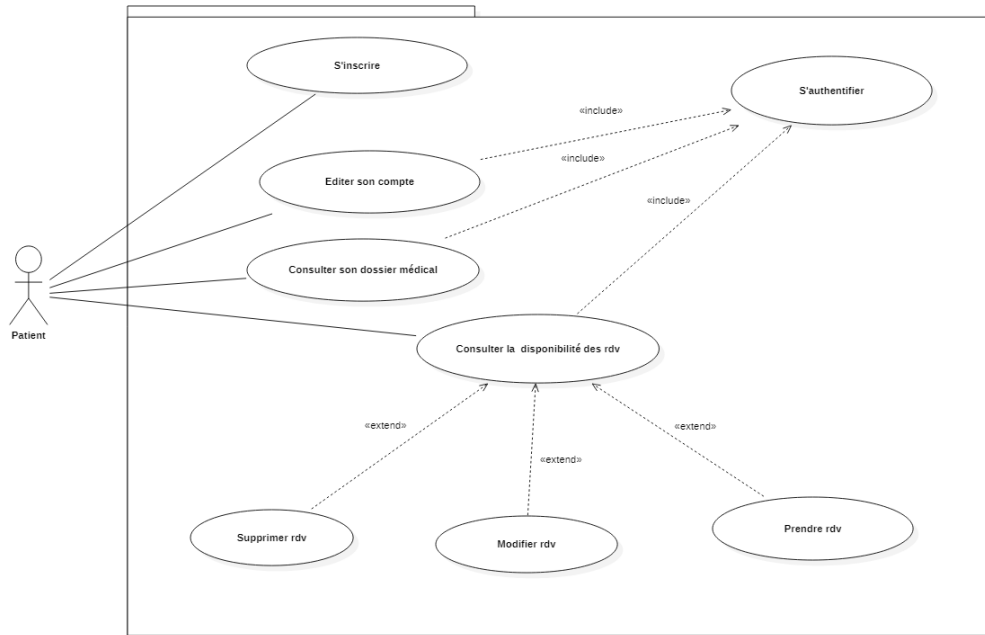


FIGURE 2.3 – Diagramme de cas d'utilisation d'un patient

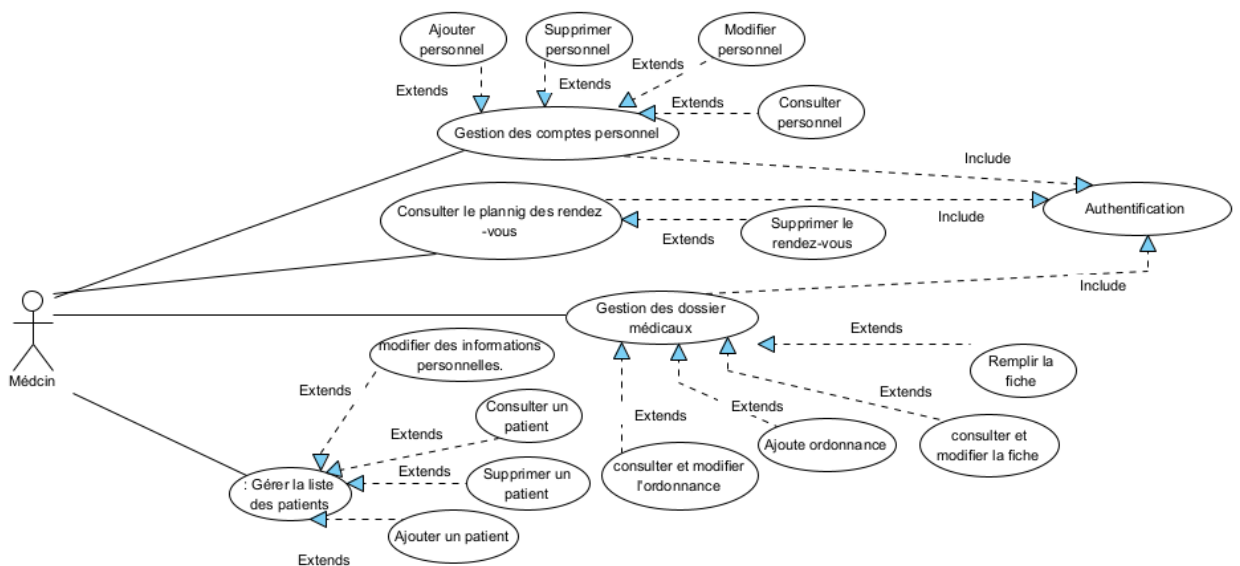


FIGURE 2.4 – Diagramme de cas d'utilisation d'un médecin

2.3 Diagrammes de séquence

2.3.1 Définition

Les diagrammes séquentiels servent à représenter les interactions entre les objets d'un point de vue temporel. La mise au point porte sur la chronologie des messages envoyés.

Scénario : Liste d'actions décrivant l'interaction d'un acteur avec le système.

Interaction : comportement qui inclut un ensemble de messages échangés par un ensemble d'objets dans un contexte donné afin d'accomplir une tâche donnée.

Message : Un message est une transmission d'information unidirectionnelle entre deux objets, l'objet transmetteur et l'objet récepteur.

Et voilà si dessous mon diagramme d'authentification d'où j'ai utilisé trois acteurs : médecin et Secrétaire et patient.

Les opérateurs d'interaction que nous avons utilisés dans les diagrammes de séquences sont :

1. **Référence (Ref)** : Cet opérateur précise que le fragment fait référence à un cas vu précédemment.
2. **Alternative (Alt)** : Cet opérateur indique que le fragment composé est le comportement de choix. On choisira un opérande avec la plus grande interaction. L'opérande choisi doit avoir une expression de garde implicite ou explicite qui prend la valeur true à ce stade de l'interaction.
3. **Loop** : Cet opérateur indique que le segment combiné représente une boucle.
4. **OPT** : exécute une sous-séquence facultative si la condition de garde est vraie [4]

2.3.2 Diagramme de séquence « s'authentifier »

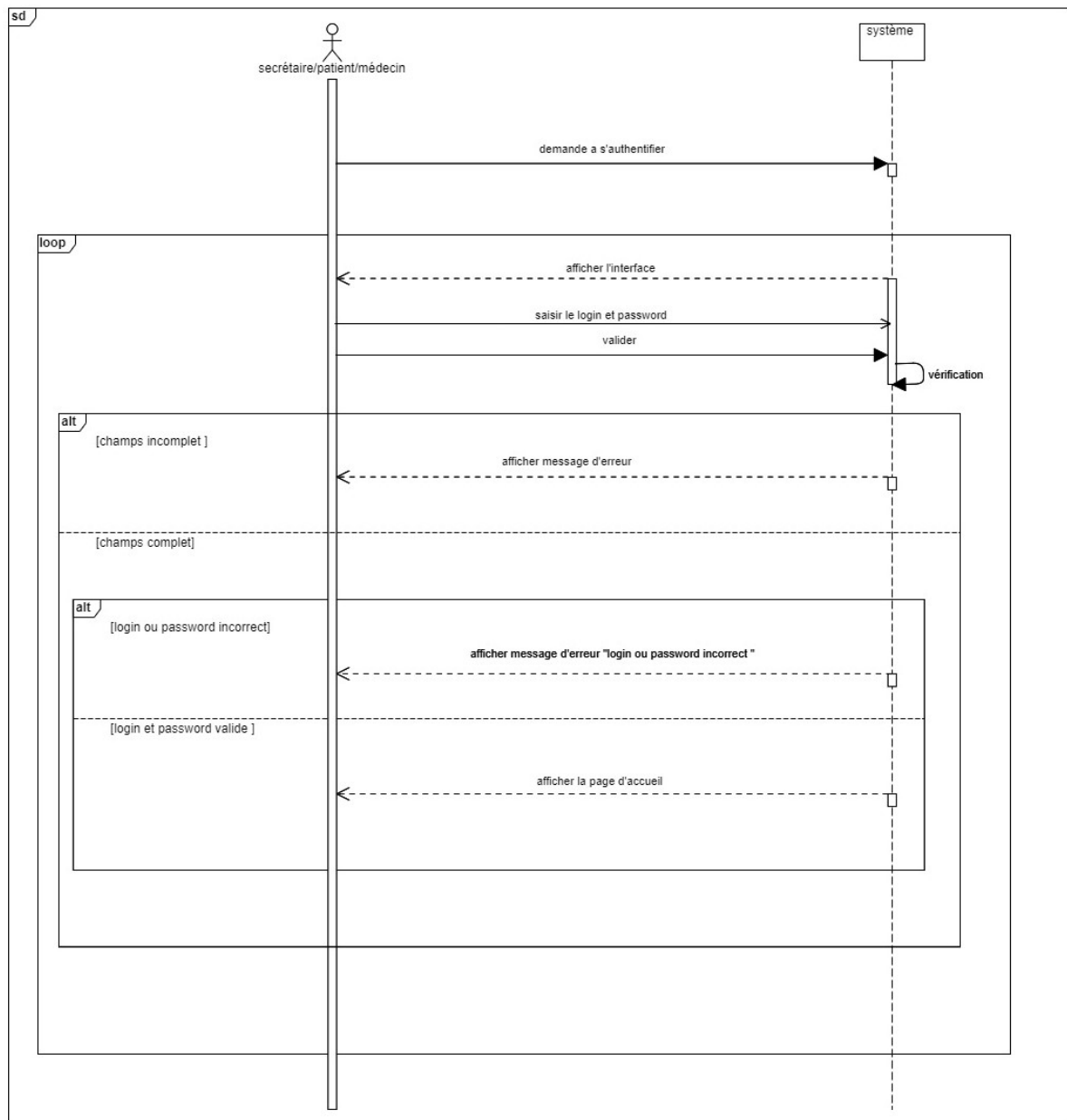


FIGURE 2.5 – Diagramme de séquence Authentification.

La figure montre le diagramme de séquence du cas utilisation « Gestion des comptes personnels médicaux »

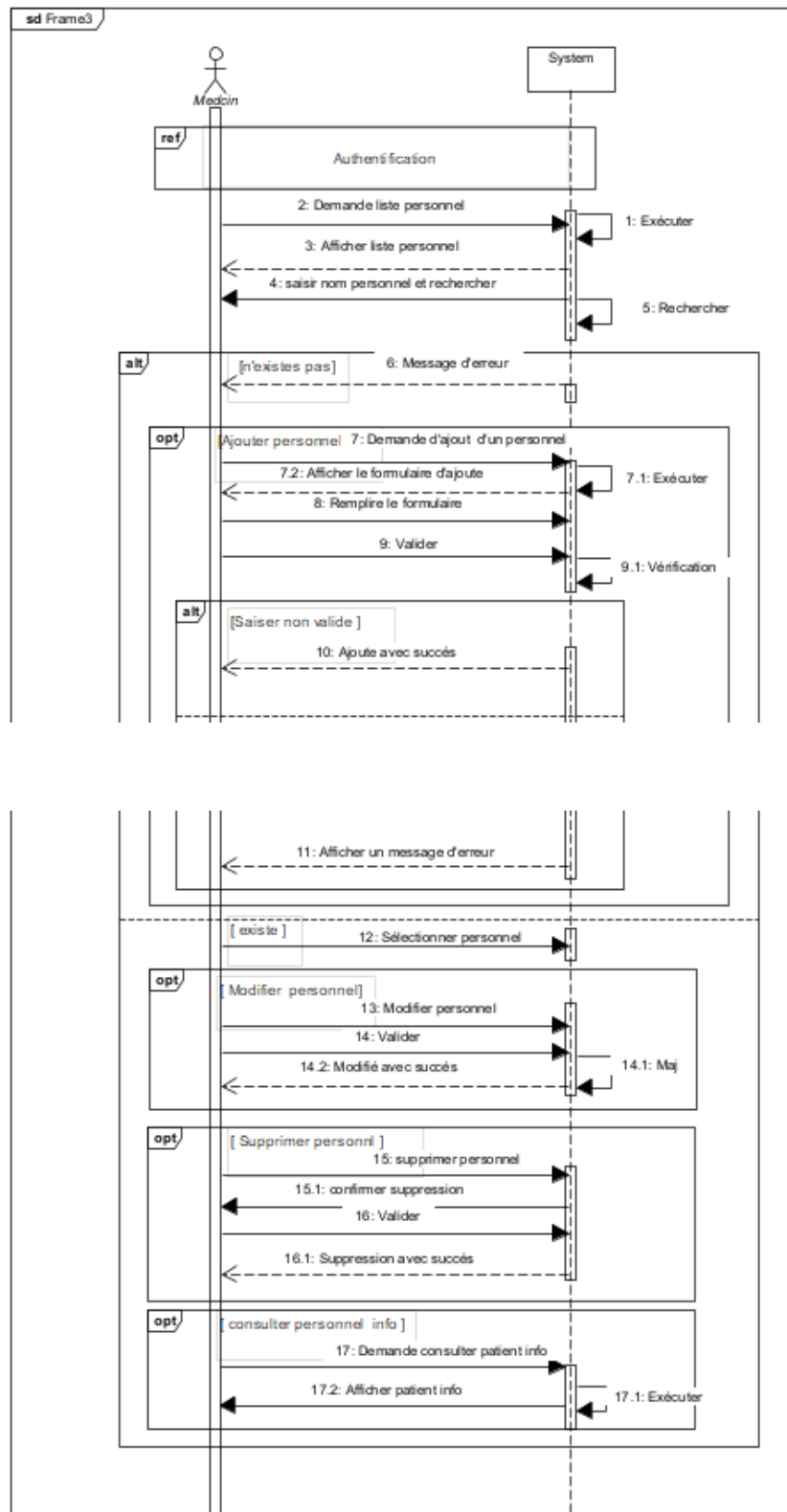


FIGURE 2.6 – Diagramme de séquence

La figure montre le diagramme de séquence « Consulter planning des RDV »

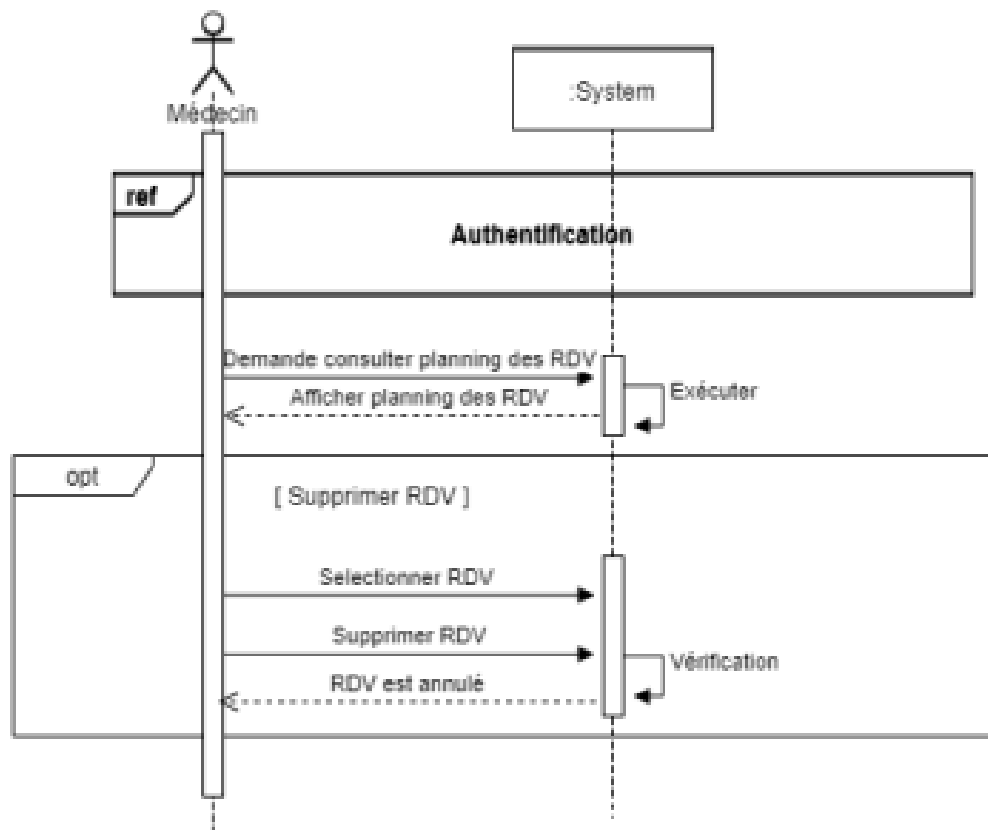


FIGURE 2.7 – Diagramme de séquence «Consulter planning des RDV».

La figure suivante décrit le diagramme de séquence correspondant au cas d'utilisation «Gestion de la liste des patients»

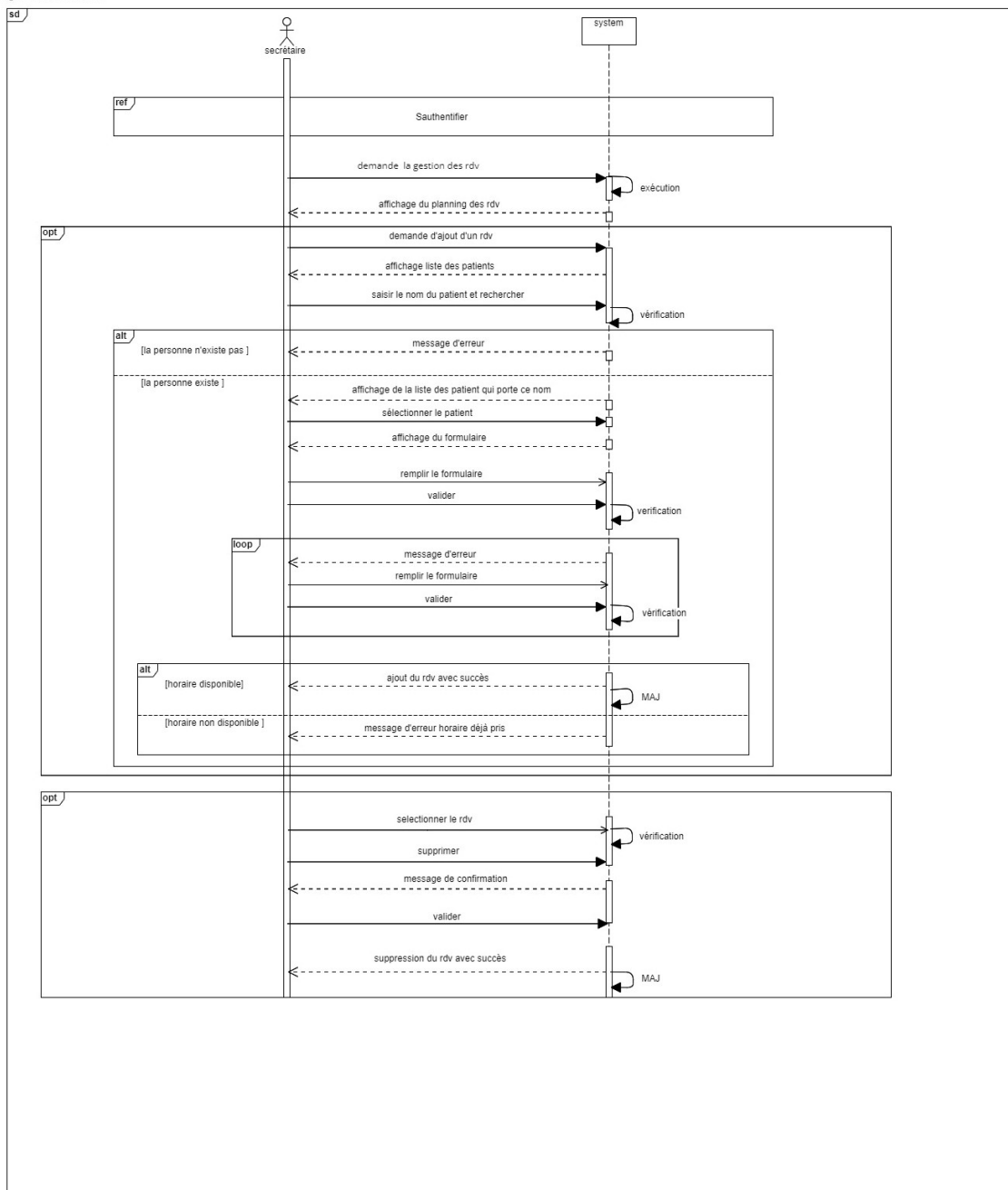


FIGURE 2.8 – Diagramme de séquence « Gestion de la liste des patients ».

La figure suivante montre le diagramme de cas d'utilisation « Consulter mon dossier médical »

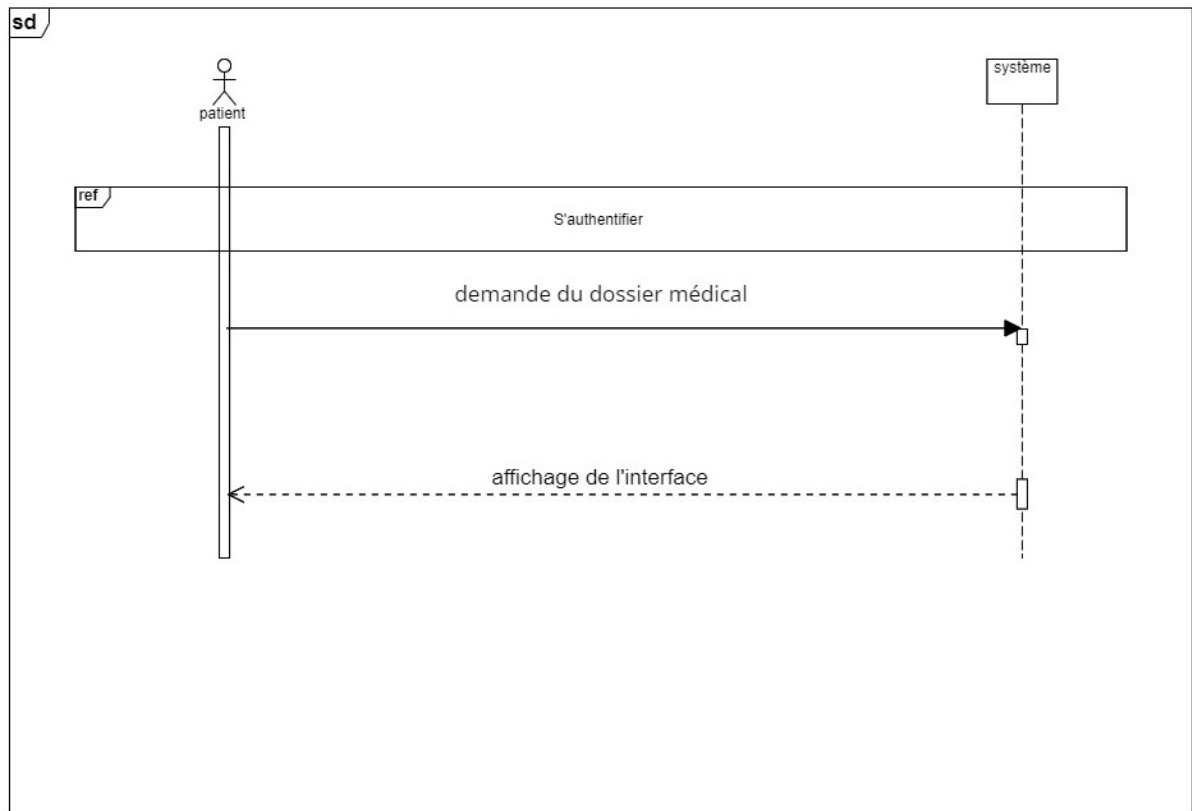


FIGURE 2.9 – Diagramme séquence de cas utilisation «Consulter mon dossier médical».

la figure suivante décrit le diagramme de séquence de cas d'utilisation «Consulter planning RDV

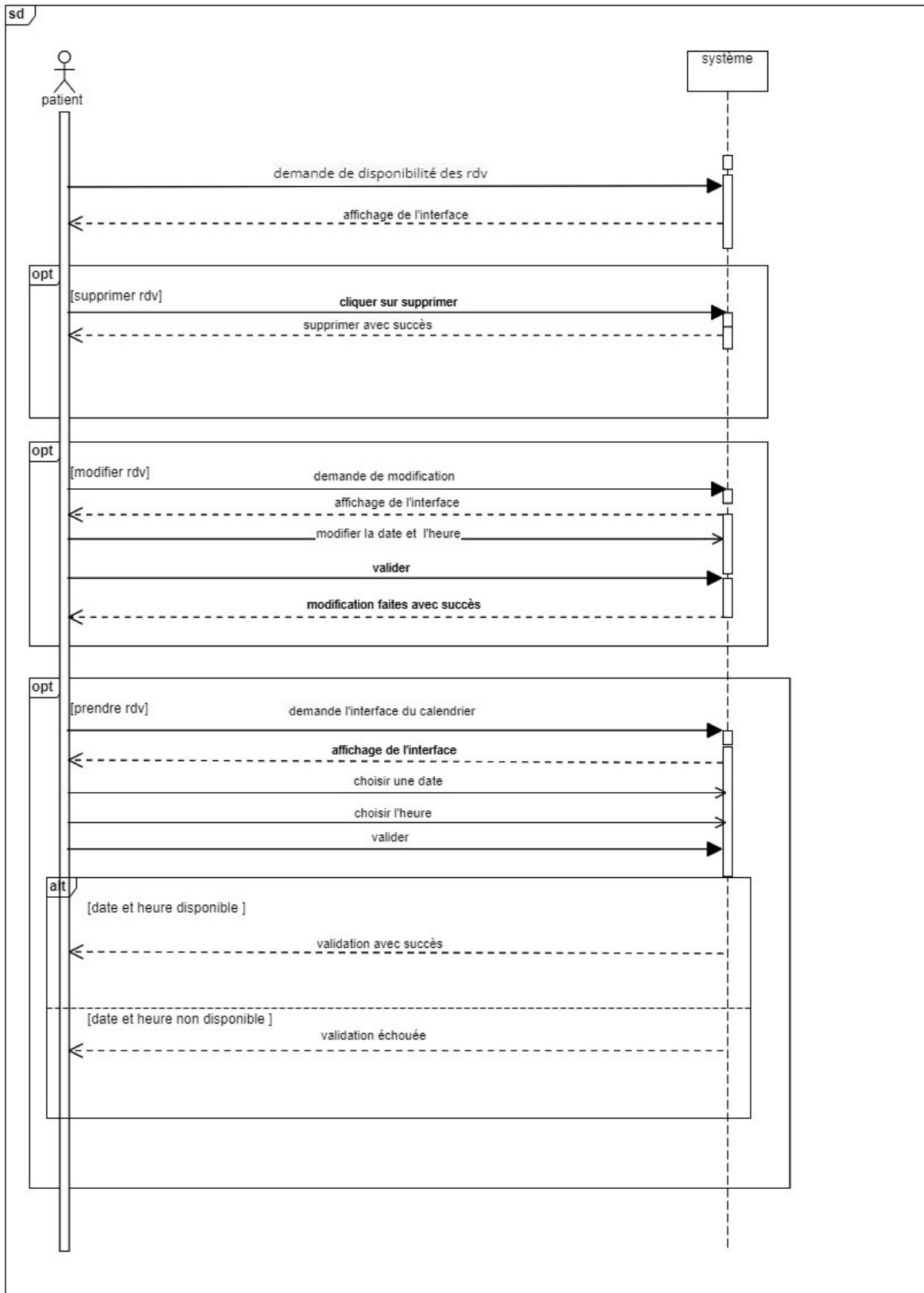


FIGURE 2.10 – Diagramme de séquence de cas d'utilisation «Consulter planning RDV ».

la figure suivante décrit le diagramme de séquence de cas d'utilisation Consulter disponibilité des RDV

2.4 Diagramme de classe

2.4.1 Définition :

Les diagrammes de classes sont probablement les diagrammes les plus couramment utilisés en UML ; ils décrivent les types d'objets qui composent un système et les différents types de relations statiques qui existent entre eux.

2.4.2 Le concept de la classe :

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations) et de relations avec d'autres objets (les associations et les agrégations).[12].La classe est définie par son nom, ses attributs et ses opérations.

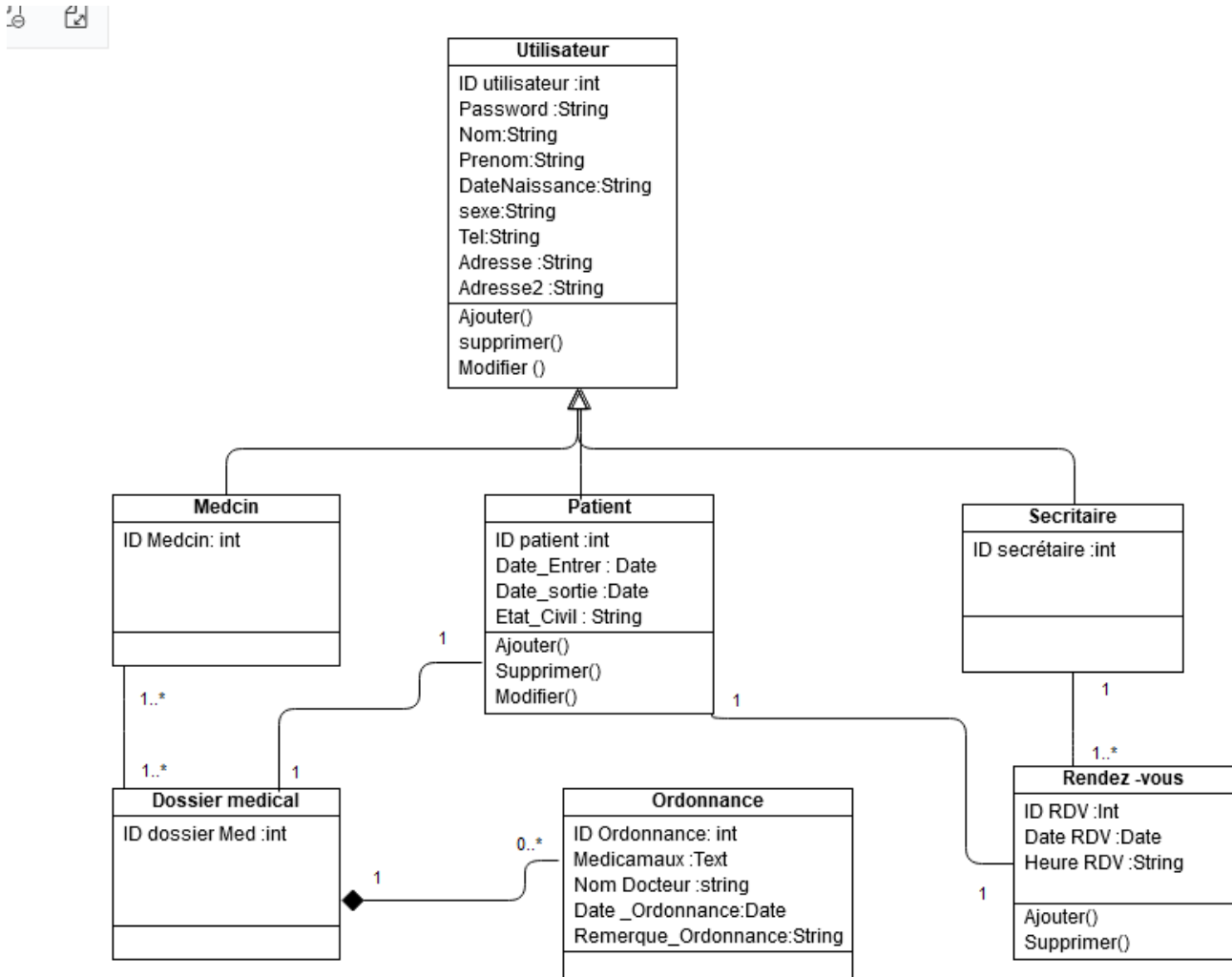


FIGURE 2.11 – Figure – Diagramme de classe globale de notre application

2.4.3 Le modèle relationnel

Modèles conceptuels ou relationnels :

- Sur la base de la description du concept que j'ai faite, nous pouvons faire le modèle relation ; étant donné que les systèmes d'information ne peuvent pas traiter directement celle-ci ; et dans Utilisons les règles de transformation d'UML en relationnel. [31].

- Quelques notions clés :

1. **Relation** : Ceci est une partie du produit cartésien d'une liste de domaines. Il s'agit en fait d'un tableau dont les colonnes correspondent à des Domaines et dont les lignes contiennent des tuples. Un nom est associé avec chacune des colonnes.
2. **Entité** : toutes les entités sont converties en tables, les attributs d'entité deviennent des attributs de la table et l'identifiant de l'entité devient la clé primaire de la table.
3. **Domaine** : Il s'agit d'une collection de valeurs pour un attribut.
4. **Attribut** : C'est une colonne de la relation, caractérisée par le nom.
5. **Association** : Une relation entre deux ou plusieurs entités.
6. **Cardinalité** : utilisée pour définir les conditions dans lesquelles une entité participe à une relation. Cependant, une entité peut participer à plusieurs relations.
7. **L'abrité** : est le nombre d'attributs de la relation.
8. **Clé étrangère** : une propriété qui est la clé primaire d'une autre entité.

2.4.4 Les règles de passage :

Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé. Transformation des associations : il existe trois types d'associations :

1. « **Association 1..*** » : crée des clés étrangères au sein de la relation. Correspondants pour les entités cotées [19]. Cette clé étrangère est la clé primaire de la relation correspondant aux autres entités (n).
2. « **Association *.**** » : crée une relation avec une clé primaire qui comprend des clés étrangères., de références correspondant à association [19].
3. « **Association 1..1** » : On ajoute à la relation un attribut de type clé étrangère de multiplicité minimale égale à 1, qui porte le nom de la clé primaire de l'autre relation à laquelle l'association est associée. Si les deux multiplicités minimales sont égales à un, il est préférable de fusionner les deux relations en une seule.

Utilisateur (Id_utilisateur, password, nom, prenom, datenais, sexe, tel, adresse, adresse2).

Médecin (Id_med).

Patient (Id_pat, date_entr, date_sort, etude_civil).

Secrétaire (Id_secret, #Idrdv).

Dossier médicale (Id_dossmed, #Id_ord).

Consulter (#Id_med, #Id_dossmed).

Ordonnance (Id_ord, medic,nom_doct, dat_ord, remq).

Rendez-vous (Idrdv, dat_rdv, heur_rdv).

Conclusion

Cette partie nous aide à déterminer la fonctionnalité du système, ce qui nous amène à concevoir et concevoir une application qui correspond aux besoins attendus en se basant sur les diagrammes du langage UML.

Introduction

Après une analyse approfondie nous allons mettre en valeur la partie implémentation. Nous discuterons du modèle MVC, des outils (logiciels, frameworks, etc.) qui nous aident à implémenter notre application, et enfin nous présenterons quelques interfaces du système.

3.1 Le modèle MVC

est un modèle architectural qui sépare une application en trois composants logiques principaux : le modèle, la vue et le contrôleur. Chacun de ces composants est conçu pour gérer des aspects de développement spécifiques d'une application. MVC est l'un des frameworks de développement Web standard les plus fréquemment utilisés pour créer des projets évolutifs et extensibles.

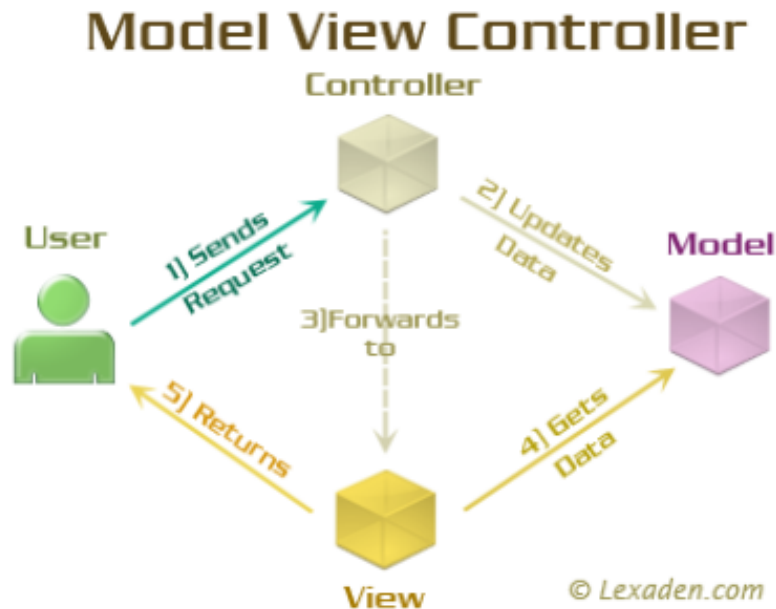


FIGURE 3.1 – Le modèle MVC [5].

3.1.1 Le Modèle

- Il représente les données et les règles qui gouvernent l'accès à et la mise-à-jour de ces données.
- Un modèle représente souvent une approximation logicielle d'un entité/processus réel
- Magnétomètre 3-Axis Digital Accelerometer(1.5g).

3.1.2 La vue

- Elle affiche le contenu d'un modèle.
- Elle spécifie exactement comment les données du modèle doivent être présentées.
- Si le modèle change, la vue doit mettre à jour sa présentation pour les refléter.
- Push Model : la vue s'abonne au modèle pour obtenir des notifications de changement.
- Un modèle représente souvent une approximation logicielle d'un entité/processus réel.
- Pull Model : la vue est responsable d'appeler le modèle quand elle a besoin de retrouver les données les plus actualisées.

3.1.3 Le Contrôleur

- Traduit les interactions de l'utilisateur avec la vue en actions que le modèle exécutera.
- Dans un client GUI, les interactions de l'utilisateur peuvent être un click sur un bouton ou une sélection d'un menu.
- Dans une application web, elles apparaissent comme des requêtes HTTP de type GET et POST.
- En fonction du contexte, un contrôleur peut aussi sélectionner une nouvelle vue à présenter à l'utilisateur, après que les actions soient effectuées par le modèle Les frameworks.
- Un ensemble de bibliothèques, contenant des packages, des classes et un modèle de programmation (orienté objet, MVC, orienté agent, orienté acteur, orienté aspect, etc...)
- Le développeur doit utiliser les packages et classes du framework.
- Ce qui garantit en contre-partie que sa programmation suivra le modèle du framework ; gage de qualité.

3.2 L'environnement de développement

3.2.1 Visual Studio Code

Visual Studio Visual Studio Code est un éditeur de code source,léger, mais performant qui fonctionne sur son bureau,et disponible pour Windows,Mac Os et linux. Il est doté d'un support JavaScript, TypeScript et Node.js intégré et dispose d'un. un écosystème riche en extensions pour d'autres l'angages. [32]



FIGURE 3.2 – Visual Studio Code.

3.2.2 GitHub

GitHub est un service destiné à héberger et développer des logiciels. Il est gratuit pour les projets en open source, et les paquets sont également disponibles pour les projets privés. À l'heure actuelle, GitHub est le service d'hébergement de code le plus populaire parmi les développeurs et les programmeurs libres. Il se sert du système de contrôle dans Git, d'où le nom [20].



FIGURE 3.3 – GitHub .

3.2.3 Postman

Postman est un service destiné à héberger et développer des logiciels. Il est gratuit pour les projets en open source, et les paquets sont également disponibles pour les projets privés. À l'heure actuelle, Postman est le service d'hébergement de code le plus populaire parmi les développeurs et les programmeurs libres. Il se sert du système de contrôle dans Git, d'où le nom [21].



FIGURE 3.4 – Postman

3.2.4 Overleaf

Overleaf est un éditeur LaTeX collaboratif basé sur le cloud utilisé pour écrire, éditer et publier des documents scientifiques. Il s'associe à un large éventail d'éditeurs scientifiques pour fournir des modèles LaTeX de revues officielles et des liens de soumission directs [22].



FIGURE 3.5 – Overleaf .

3.2.5 Visual Paradigm

Est un outil de conception et de gestion puissant, multiplateforme et pourtant facile à utiliser pour les systèmes informatiques. Visual Paradigm fournit aux développeurs de logiciels la plate-forme de développement de pointe pour créer des applications de qualité plus rapidement .[23] .



FIGURE 3.6 – Visual Paradigm .

3.3 Développement FullStack

Une application Web est un logiciel d'application installé sur un serveur et accessible via un navigateur Web. contrairement aux logiciels traditionnels, il n'est pas nécessaire que l'utilisateur d'une application Web l'installe sur son ordinateur, tout ce qu'il faut faire, c'est se connecter à l'application en utilisant son navigateur préféré.

Nous avons découpé notre application en trois parties principales; la partie frontend qui contient le code de l'interface avec le framework React, cette partie communique avec le backend qui est constitué d'une application server développée avec node.js et le framework Mongoose, cette partie interagit à son tour avec la base de données MongoDB qui stocke les données de l'application.

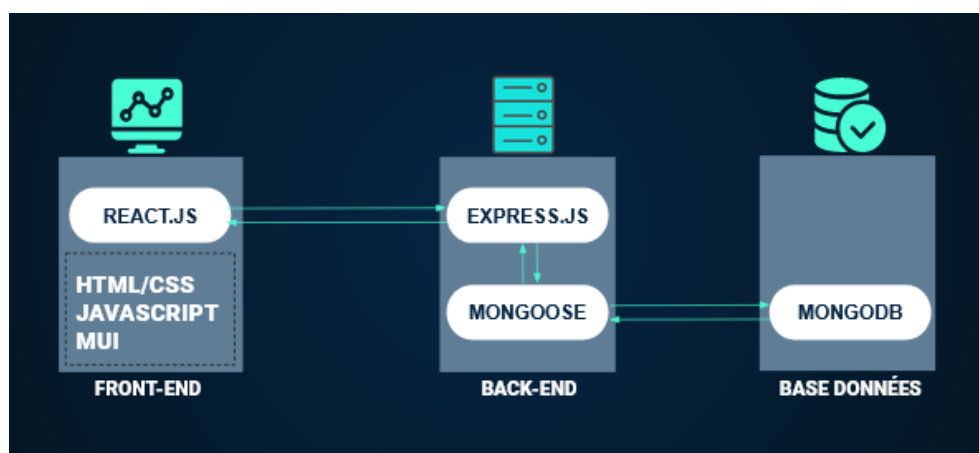


FIGURE 3.7 – Architecture de l' application .

3.3.1 BackEnd

• Langages utilisées

JavaScript : Spécifie un langage de développement informatique, plus précisément un langage de script orienté objet. il est retrouvé notamment sur les pages Internet. Il permet entre autres l'introduction de petites animations ou d'effets sur des pages web ou des pages HTML. Créé par Brendan Eich en 1995, concurrent de la technologie Java, le langage JavaScript diffère des langages serveur en ce que l'exécution des tâches est effectuée par le navigateur lui-même sur l'ordinateur de l'utilisateur, et non sur le serveur Web. Par conséquent, il est généralement activé sur le poste de travail client plutôt que sur le côté serveur.

- **Framework utilisées**

Nodejs : Node JS est un environnement d'exécution Javascript open source sur plusieurs plateformes. Il est conçu pour faire fonctionner JavaScript hors du navigateur, côté serveur. Node JS accède au gestionnaire de noeuds empaquetés (NPM), qui héberge un grand nombre de paquets privés et publics, et sert également à publier des projets Node JS. Node js est très souvent utilisé pour l'écriture de services côté serveur nommés API.[24] .



FIGURE 3.8 – logo nodejs .

Express : Express est un framework d'applications Web node js qui fournit de nombreuses fonctionnalités pour la création d'applications Web et mobiles. Il est utilisé pour créer une application Web monopage, multipage et hybride. C'est une couche construite au-dessus du Node js qui aide à gérer les serveurs et les routes. Express a été créé pour créer facilement des API et des applications Web. cela permet d'économiser beaucoup de temps de codage presque de moitié et rend toujours le Web est efficace. Une autre raison d'utiliser express est qu'il est écrit en Javascript car Javascript est un langage facile même si vous n'avez pas de précédent.[25] .



FIGURE 3.9 – logo express .

Mongoose : Est une bibliothèque ODM (Object Data Modeling) pour MongoDB et Node.js. Cela signifie que Mongoose vous permet de définir des objets à l39;aide d39;un schéma fortement typé qui correspond aux documents MongoDB. Il gère les relations entre les données, fournit une validation de schéma et est utilisé pour convertir entre les objets dans le code et les représentations de ces objets dans MongoDB. Mongoose est un framework JavaScript couramment utilisé dans les applications Nodejs avec des data bases MongoDB. il fournit de nombreuses fonctionnalités pour créer et utiliser des schémas et contient actuellement 8 SchemaTypes, et lorsqu39;il est conservé dans MongoDB, une propriété est enregistrée sous ces SchemaTypes. [26] .



FIGURE 3.10 – logo mongoose .

- **Base de données utilisées**

MongoDB : Est une base de données NoSQL multiplate forme orientée document qui offre des performances élevées, mongoDB est un logiciel libre.et Base de données orientée documentée.Une intégration plus rapide des données dans les applications. Données sous format JSON binaire.Stocker des grandes quantités considérables de données.[27] .



FIGURE 3.11 – logo MongoDB .

3.3.2 Frontend

- **Langages utilisées**

JavaScript : JavaScript spécifie un langage de développement informatique, plus précisément un langage de script orienté objet. il est retrouve notamment sur les pages Internet. Il permet entre autres l39;introduction de petites animations ou d39;effets sur des pages web ou des pages HTML. Créé par Brendan Eich en 1995, concurrent de la technologie Java, le langage JavaScript diffère des langages serveur en ce que l39;exécution des tâches est effectuée par le navigateur lui-même sur l39;ordinateur de l39;utilisateur, et non sur le serveur Web. Par conséquent, il est généralement activé sur le poste de travail client plutôt que sur le côté serveur. [28] .



FIGURE 3.12 – logo JavaScript .

- **Framework utilisées**

Reactjs : Est une librairie JavaScript libre utilisée pour créer des interfaces utilisateur, habituellement pour des applications d'une seule page.

React JS permet aux développeurs d'éditer/modifier et de rafraîchir la page pour afficher les changements sans devoir redémarrer ou recharger la page.

React JS permet la réutilisation de code sur de multiples plateformes.[29] .

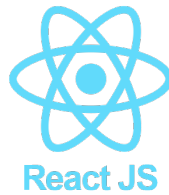


FIGURE 3.13 – Logo ReactJs .

Bootstrap : est un framework développé par l'équipe de réseaux sociaux de Twitter. Disponible en open source (sous licence MIT), il fournit aux développeurs des outils pour créer facilement des sites en utilisant les langages HTML, CSS et JavaScript. Il est conçu pour développer des sites Web avec un design réactif qui fonctionne sur tout type d'écran, principalement pour les smartphones. Il fournit des outils pour les styles préexistants pour la typographie, les boutons, la navigation, etc. Ce type de framework est appelé "framework frontal".[30]



FIGURE 3.14 – Logo bootstrap .

3.4 Présentation de quelques interfaces de l'application

Dans la présente partie, nous décrivons les diverses interfaces de notre application, ces interfaces sont les suivantes :

3.4.1 Interface d'accueil

À la suite du lien de notre application Web, l'interface ci-dessous s'affiche en premier. Il y a une courte vue d'ensemble du site, et les boutons d'authentification pour les médecins et les patients, les utilisateurs (médecins/patients) peuvent s'inscrire et créer un compte ou s'authentifier s'ils ont déjà inscrits. Le bouton Médecins affiche la liste des médecins disponibles sur l'application qui sont prêts à prendre des rendez-vous.

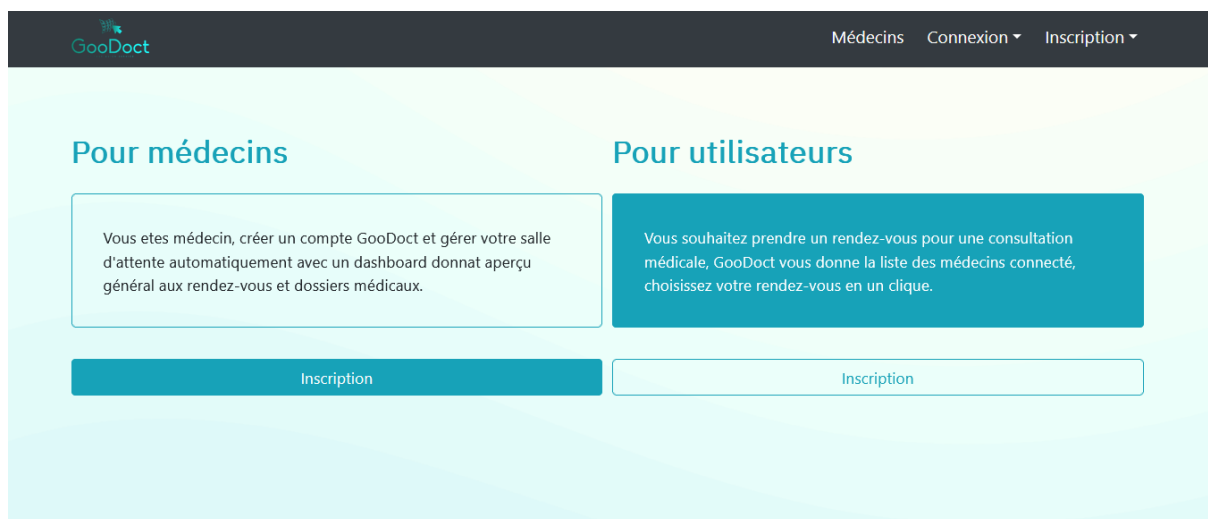


FIGURE 3.15 – Interface d'accueil de l'application .

3.4.2 Interface d'inscription

Pour patients : En cliquant sur le bouton inscription pour utilisateurs (figure 3.15), l'interface d'inscription d'un **patient** s'affiche (figure 3.16), elle permet à un **patient** de créer son compte, il s'agit d'un formulaire que ce dernier doit remplir avec ses renseignements personnels (adresse e-mail, nom, prénom et mot de passe).

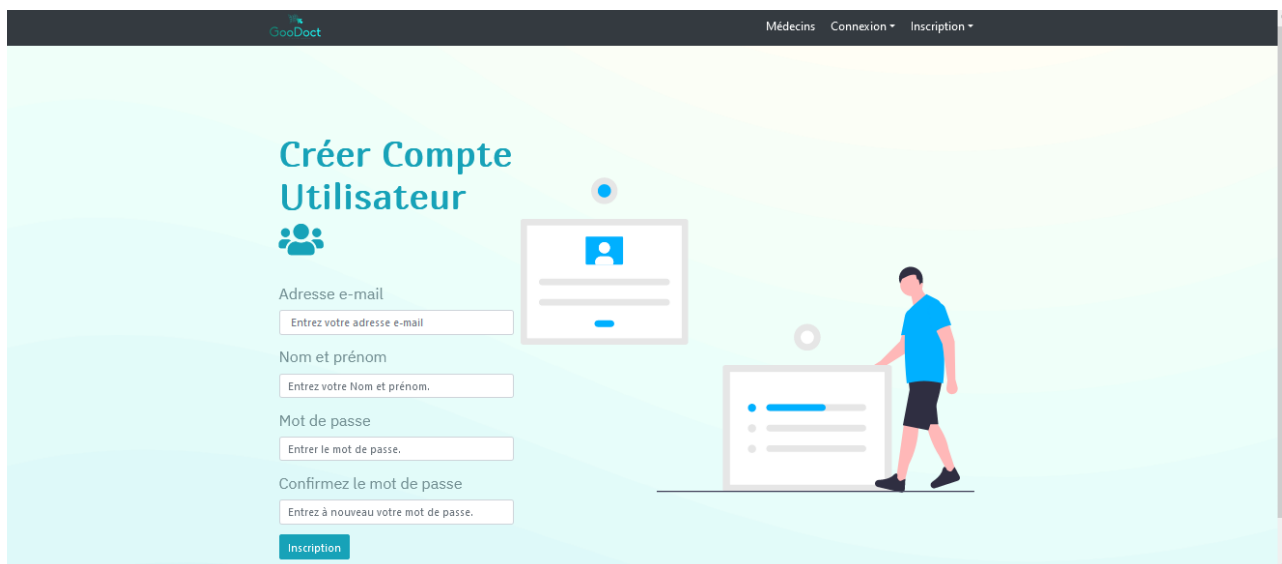


FIGURE 3.16 – Interface d’inscription pour un patient.

Pour médecins : En cliquant sur le bouton inscription pour médecins (figure 3.15), l’interface d’inscription d’un **médecin** s’affiche (figure 3.17), elle contient un ensemble de zones d’entrée que le médecin doit remplir avec ses renseignements personnels, pour effectuer son inscription.

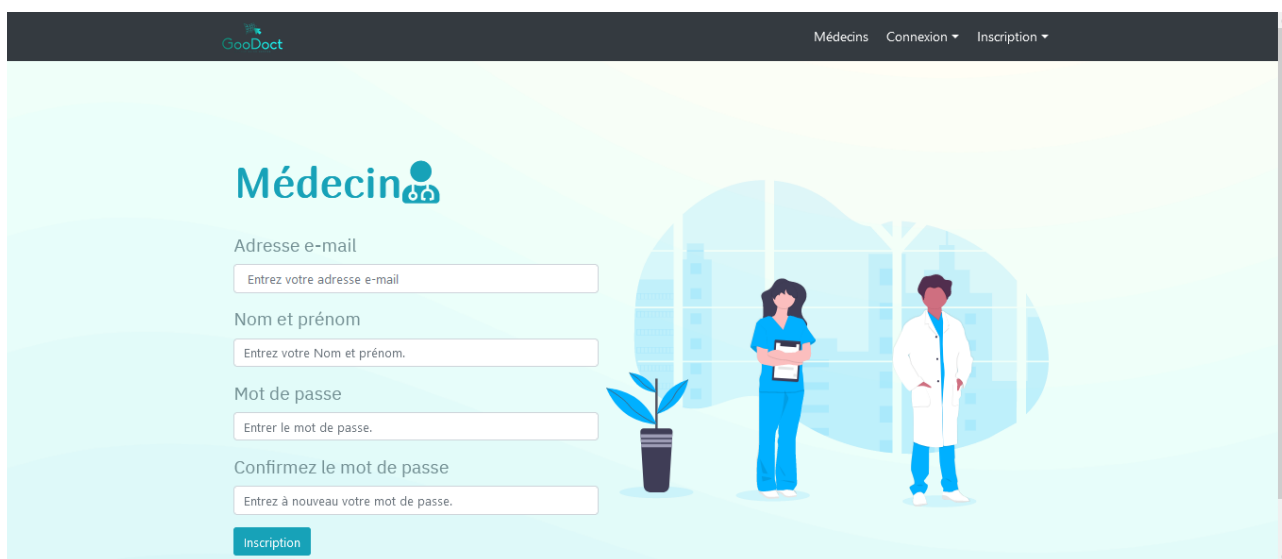


FIGURE 3.17 – Interface d’inscription pour un médecin.

3.4.3 Interface d’identification

Pour patients : Une fois le patient a créé son compte utilisateur avec succès, l’interface (Figure 3.18) lui permet d’accéder à son compte ; pour cela, l’utilisateur (patient) doit saisir son adresse e-mail et son mot de passe. Dans le cas où le patient n’as pas encore effectué son inscription, l’interface lui permet d’accéder à la page d’inscription en cliquant sur le lien "Créer un nouveau compte", il sera redirigé vers la (Figure 3.16).



FIGURE 3.18 – Interface d'identification d'un patient

Pour médecins : Une fois le médecin a créé son compte avec succès, l'interface (Figure 3.19) lui permet d'accéder à son compte ; pour cela, le médecin doit saisir son adresse e-mail et son mot de passe. Dans le cas où le patient n'as pas encore effectuer son inscription, l'interface lui permet d'accéder à la page d'inscription en cliquant sur le lien "Créer un nouveau compte", il sera redériger vers la (Figure 3.19).



FIGURE 3.19 – Interface d'identification d'un médecin

3.4.4 Interface création d'un profil médecin

L'interface profil est réservée uniquement pour les médecins, elle est affichée à la suite de la création d'un compte médecin pour lui permettre de saisir les informations de son cabinet et ses expériences professionnels telque ; sa spécialité, les horaires de travail, le tarif des consultations, etc. (Figure 3.20)

Dashboard Médecin Déconnexion

Créer un profil

Obtenez des informations pour que votre profil se démarque

* = champs requis

* Entrez votre status. ex. Professeur, Spécialiste etc.

Donnez-nous une idée de l'endroit où vous en êtes dans votre carrière

* Clinique

Pourrait être votre propre clinique ou celle que vous travaillez

Spécialité

Donnez-nous une idée de votre spécialité.

* Horraire de travail

À quel moment vous êtes disponible pour les patients, mentionner le jour avec le temps.

Pourrait être votre propre clinique ou celle que vous travaillez

Spécialité

Donnez-nous une idée de votre spécialité.

* Horraire de travail

À quel moment vous êtes disponible pour les patients, mentionner le jour avec le temps.

* Frais de consultation (DA)

Frais de consultation (DA).

* Localisation.

Pourrait être votre propre adresse clinique ou celle que vous travaillez

* Une brief description de vous

Parle-nous un peu de toi

Ajouter des liens de réseau social Optionnelle

Envoyer Go Back

FIGURE 3.20 – Interface de création d'un profil médecin

3.4.5 Interface d'affichage du profil d'un médecin

L'interface profil d'un médecin affiche les détails et les informations du médecin, elle est accessible pour tout les utilisateur afin de voir les détails du médecin (sa spécialité, sont domaine, ses tarifs et les horraire de travail ... etc.) Figure (3.21).

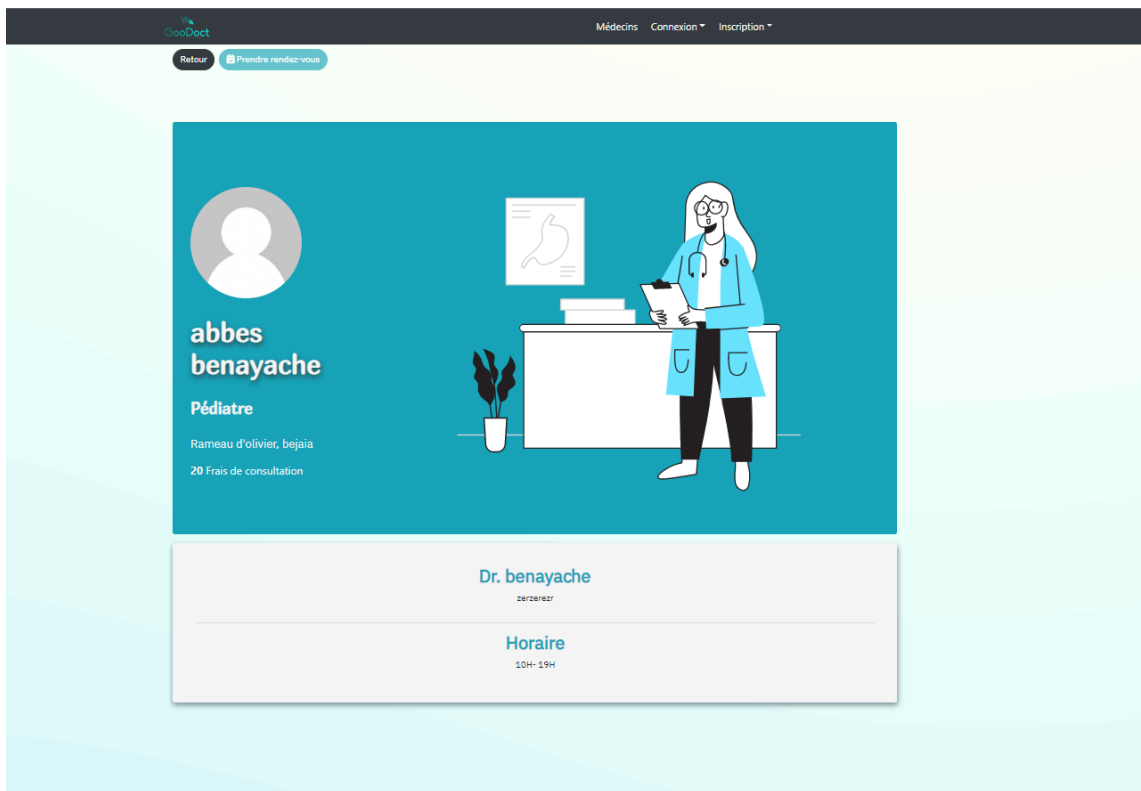


FIGURE 3.21 – Interface d’affichage d’un profil médecin

3.4.6 Interface Rendez-vous

L’interface rendez-vous s’affiche pour le patient et le médecin.

Pour un patient, l’interface (Figure 3.21) lui permet de visualiser sa liste des rendez-vous qu’il a réservée avec les différents médecins (un utilisateur peut prendre plusieurs rendez-vous sur différents cabinets).

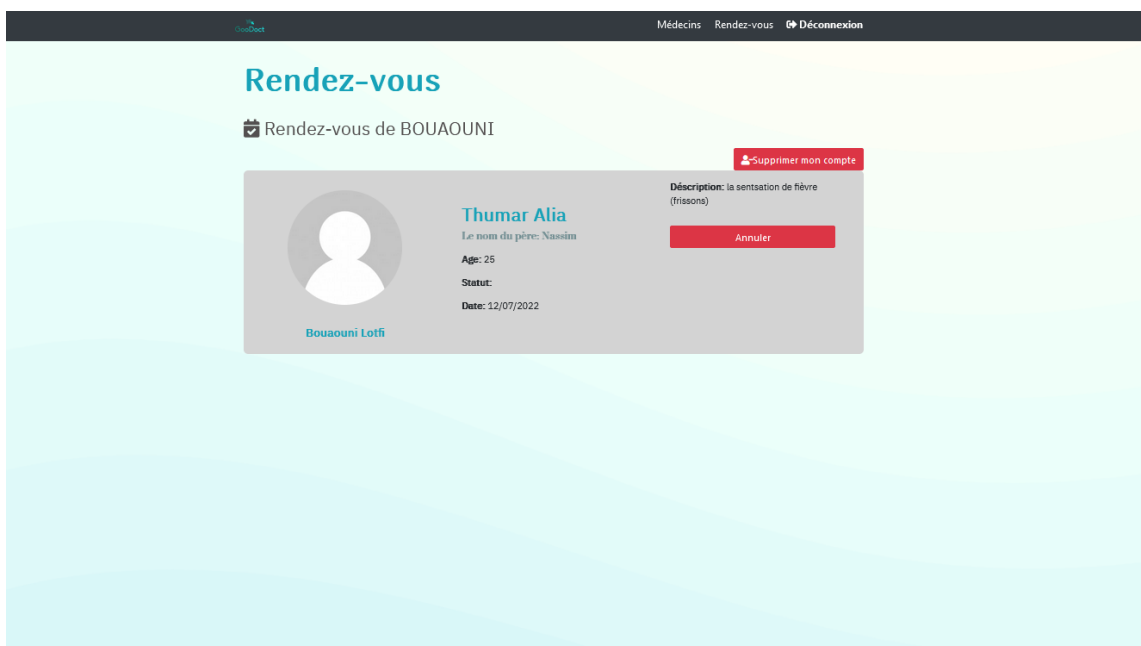


FIGURE 3.22 – Interface liste des rendez-vous (patient)

Pour le médecin l’interface des rendez-vous affiche la liste des rendez-vous que les patients ont pris

après de son cabinet (Figure 3.22). Le médecin peut voir les détails du rendez-vous demandé en cliquant sur le bouton "voir", il peut ne pas accepter le rendez-vous en cliquant sur le bouton "refuser".

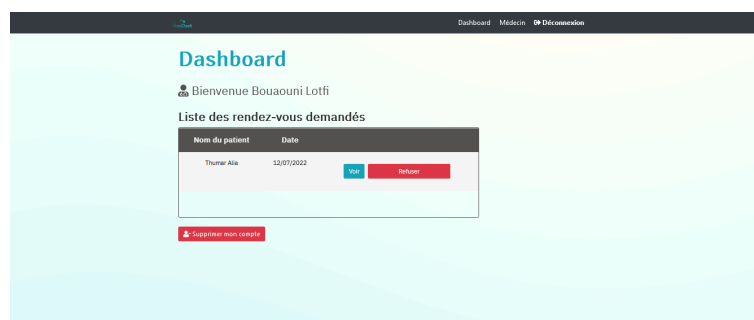


FIGURE 3.23 – Interface liste des rendez-vous (médecin)

3.4.7 Interface liste des medecins

Tout utilisateurs peut visualiser la liste des médecins inscrit sur l'application, afin de prendre un rendez et/ou visualiser le profil du médecin (Figure 3.23).



FIGURE 3.24 – Interface liste des profils médecin inscrits sur l'application

3.4.8 Structure du projet

Voici comment nous avons structurer notre projet React/Node.js afin de respecter les bonnes pratiques et surtout que le projet soit scalable dans le temps. L'idée d'une bonne structure est d'avoir une architecture de dossiers/fichiers lisible par tous et de se rendre directement au bon endroit sans avoir besoin de chercher trop longtemps pour intervenir sur le projet (Figure 3.25).

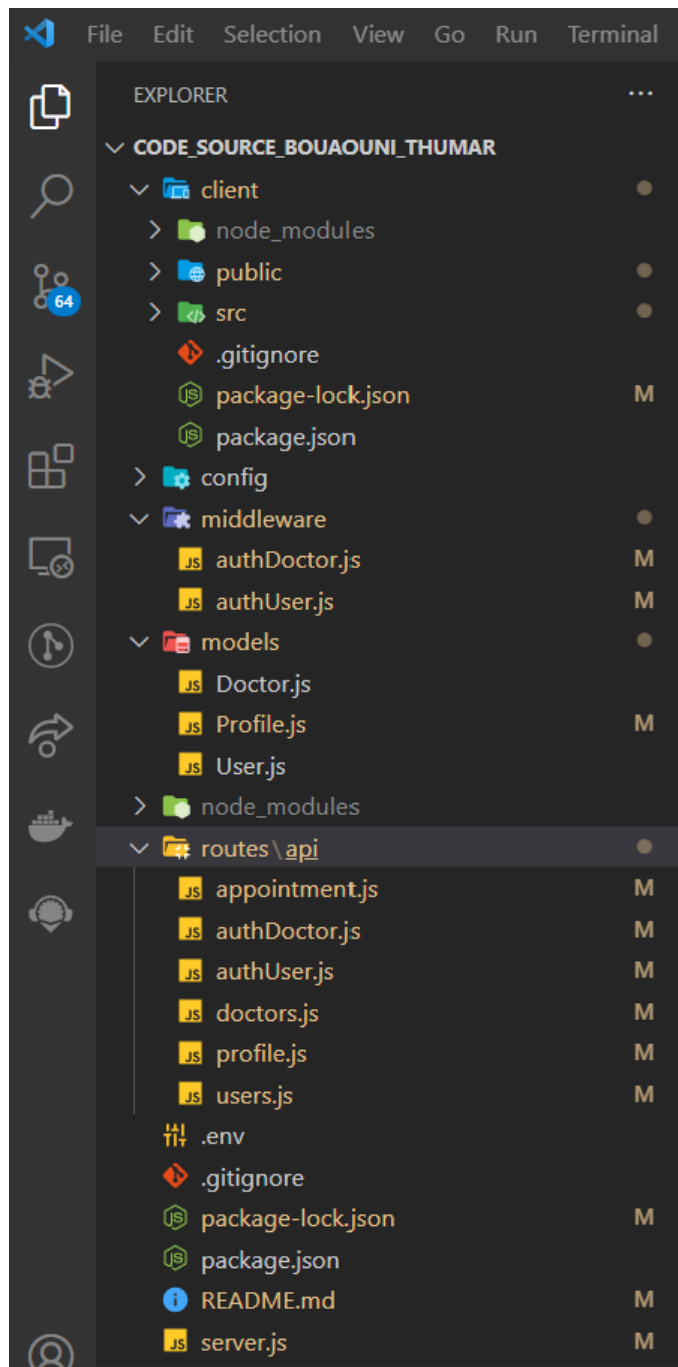


FIGURE 3.25 – Structure du projet

3.5 Conclusion

Ce chapitre est le bilan de notre projet, nous avons décrit la partie implémentation, le modèle MVC, nous avons constaté les outils lesquels sont utilisés pour développer notre système et nous avons montré quelques interfaces pour finir cette section.

CONCLUSION GÉNÉRALE

Dans le cadre de ce projet de fin d'étude en génie logiciel, nous avons conçu avec succès une application web de gestion des rendez-vous d'un cabinet médical.

Notre objectif principal est de créer une application web interactive qui faciliterait la prise de rendez-vous des patients de ce cabinet et permettrait au médecin de bien gérer et d'organiser les dossiers de ses patients.

La conception d'un projet pareil nous a permis d'implémenter notre bagage obtenu durant notre cursus universitaire et d'appliquer de nouvelles technologies, ce qui nous a permis de maîtriser plusieurs langages et outils de développements.

Bien évidemment que nous avons traversé des échecs et des complications durant la modélisation ainsi que durant l'implémentation.

Ce travail nous a donné l'occasion de nous approfondir dans le développement et la conception en bénéficiant de la compréhension du monde professionnel et la gestion du travail en équipe.

Il est à noter que notre application va avoir des améliorations dans le futur :

- Le paiement en ligne.
- Relier notre application web à une application mobile.
- Améliorer les performances de notre application.

RÉFÉRENCES

- [1] Application web, définition de l'application web. <https://www.ideematic.com/>. (Consulté le 26/6/2022).
- [2] Choix du processus de développemen,scrum. <https://asana.com/>.(Consulté le 26/6/2022).
- [3] Choix du processus de développemen,scrum. <https://www.qrpinternational.ch/>.(Consulté le 26/6/2022).
- [4] Diagrammes de séquence,définition deu diagramme de sequence. <http://www-inf.it-sudparis.eu>.
- [5] Définition de cabinet médical. <https://spiegato.com/fr/>. (Consulté le 26/6/2022).
- [6] Etude de l'existant au niveau internationale,allodocteur. <https://allodocteur.fr/>.(Consulté le 26/6/2022).
- [7] Etude de l'existant au niveau internationale,doctolib. <https://www.doctolib.fr/>.(Consulté le 26/6/2022).
- [8] Etude de l'existant au niveau internationale,keldoc. <https://www.keldoc.com/>.(Consulté le 26/6/2022).
- [9] Etude de l'existant au niveau internationale,maiaa. <https://www.maiaa.com/>.(Consulté le 26/6/2022).
- [10] Etude de l'existant au niveau nationale,esiha. <https://www.esiha.net/>.(Consulté le 26/6/2022).
- [11] Fonctionnement d'une application web. www.google.com. (Consulté le 26/6/2022).
- [12] Le concept de la classe. www.uml-sysml.org. (Consulté le 26/6/2022).
- [13] Les concepts du web, hyper text. <https://www.techopedia.com/definition/5411/hypertext>.
- [14] Les concepts du web, moteur de recherche. <https://www.developpez.com/>. (Consulté le 26/6/2022).
- [15] Les concepts du web, site web. <https://www.techopedia.com/definition/5411/website>.(Consulté le 26/6/2022).
- [16] Scrum,définition du scrum. <https://www.scrum.org>. (Consulté le 26/6/2022).
- [17] Web, définition de web. <https://www.techopedia.com>. (Consulté le 26/6/2022).
- [18] Mémoire « développement de solutions numériques innovantes dans le domaine du transport et plus spécifiquement sur l'axe environnemental ». Master's thesis, Diplôme : Master 2 Ingénierie et Innovation en Images et réseaux, 2020.
- [19] R.M.D,Scala,Lesbasesdel'informatiqueetdelaprogrammation,2005.
- [20] <https://github.com/>,. (Consulté le 26/6/2022).
- [21] <https://www.postman.com/>, (Consulté le 26/6/2022).
- [22] <https://overleaf.com/>, (Consulté le 23/5/2022).
- [23] [://www.visual-paradigm.com/](http://www.visual-paradigm.com/). (Consulté le 26/6/2022).
- [24] <https://nodejs.org/en/>, (Consulté le 19/5/2022),.
- [25] <https://expressjs.com/fr/>, (Consulté le 23/5/2022).
- [26] <https://mongoosejs.com/>, (Consulté le 23/5/2022).
- [27] <https://www.mongodb.com/>, (Consulté le 23/5/2022).
- [28] https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript, (Consulté le 17/6/2022).
- [29] <https://fr.reactjs.org/>, (Consulté le 3/6/2022).
- [30] <https://getbootstrap.com/>, (Consulté le 22/6/2022).
- [31] H. ETIEVANT. *Webzine de vulgarisation des sciences et techniques*. « UML2 par la pratique étude cas et exercice corrigés », 2009.
- [32] Josko Marsic Gordan Struklee. Implementating schc in smart meters. <https://code.visualstudio.com/>, (Consulté le 2/6/2022),.

- [33] Dr HAMZA-Lamia. Génie logiciel, les diagrammes principaux diagrammes uml. https://elearning.univ-bejaia.dz/pluginfile.php/519027/mod_resource/content/0/Cours_HAMZA%20Lamia_G%C3%A9nie%20Logiciel.pdf, (Consulté le 22/5/2022).
- [34] Le Cocq Michel. *Applications Client/serveur et Web*. 2017.
- [35] Pascal Roques. *Méthodologie UML, Pourquoi utiliser UML ?* « UML2 par la pratique étude cas et exercice corrigés », 5eme edition eyrolles edition, 2017.

RESUME

Avec ce projet, nous avons créé un site Web dynamique et facile à intégrer.

L'environnement de travail de la clinique. L'application implémente un système qui facilite les rendez-vous en ligne. Notre système est essentiellement conçu pour permettre au client de prendre rendez-vous à un moment qui lui convient.

La modélisation du système s'est faite par UML. Et créé par MERN Stack, le logiciel est développé à l'aide de Visual Studio Code.

ABSTRACT

With this project, we have created a The working environment of the clinic.

The application resulted in a system that facilitates online appointments.

The client is free to make an appointment at a time that suits him. The system consists of UML. And built by MERN Stack using Visual Studio Code.