



République Tunisienne  
\*\*\*\*\*  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
\*\*\*\*\*  
Université de Monastir  
\*\*\*\*\*  
Institut Supérieur d'Informatique et de Mathématiques de Monastir  
\*\*\*\*\*  
Département de Technologie



N° d'ordre :

# Projet de Fin d'Etudes

*Présenté en vue de l'obtention du*

**Diplôme National de Licence Fondamentale en  
Sciences Informatiques**  
**Spécialité :**  
**Sciences Informatiques**  
*par*

*Hmida Mohamed Ali*

*Abdallah Imen*

**Conception et développement d'une plateforme de  
téléconsultation**

*Soutenu le 23/06/2021 devant la commission d'examen composée de :*

**Mr. Mahmoudi Ramzi**  
**Mme. Kotel Sonia**  
**Mr. Ben Salem Malek**  
**Mr. Habbechi Mahmoud**

**Président**  
**Membre**  
**Encadrant Académique**  
**Encadrant Professionnel**

*Année Universitaire : 2020 / 2021*



# Résumé

---

**Aucune entrée de table des matières n'a été trouvée.**

Le présent travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention du diplôme de licence fondamentale en sciences informatiques de l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIM de Monastir).

C'est ainsi que nous avons eu l'occasion de préparer notre projet de fin d'études qui consiste à développer une plateforme de téléconsultation médicale en ligne que nous avons nommée « DOCTENA TN ».

Pour réaliser ce projet, nous avons utilisé différentes technologies telles que le Framework Spring Boot, la bibliothèque React JS, la base de données NoSQL ainsi que le langage unifié de modélisation (UML).

**Mots clés :** *UML, téléconsultation, Spring Boot, React JS, NoSQL*

---

# Abstract

---

This work is part of the end of study project for the obtaining of the basic license diploma in computer science from the Higher Institute of Computer Science and Mathematics of Monastir (ISIM of Monastir).

The goal of this tutored project is to create and develop an online medical teleconsultation platform that we nominate "DOCTENA TN".

To carry out the project we used different technologies such as the Framework Spring Boot, React JS library, NoSQL database as well as the Unified Modeling Language (UML).

**Keywords:** *UML, teleconsultation, Spring Boot, React JS, NoSQL*

---

# Dédicaces

*A ma mère,*

*Qui m'a soutenu toutes les fois quand je n'étais pas alaise. Elle a une importance primordiale car son amour est inconditionnel indéfectible et sans attentes.*

*Quoique je puisse dire et écrire, je ne pourrais jamais exprimer ma grande affection et ma profonde reconnaissance. J'espère ne jamais te décevoir ni trahir ta confiance et tes sacrifices.*

*A mon père,*

*De tous les pères. Tu es le meilleur tu as été et tu seras toujours un exemple pour moi par tes qualités humaines, ta persévérance et perfectionnisme.*

*En témoignage d'années de sacrifices, de sollicitudes, d'encouragement et de prière.*

*Pourriez-vous trouver dans ce travail, le fruit de toutes tes peines et tous tes efforts.*

*En ce jour, j'espère réaliser l'un de tes rêves. La relation amicale entre nous n'a pas de prix*

*Votre écoute avec esprit ouvert pendant mon parcours m'a permis d'arriver à ce niveau.*

*A Mehdi mon très cher frère,*

*Tu as été à mes côtés pendant toutes les étapes de ce travail je t'en suis très reconnaissant. Je te dédie ce travail en témoignage de ma profonde affection et en souvenirs de notre indéfectible et innée union qui s'est renforcée au fil du temps.*

*Au final aucune dédicace ne saurait exprimer mes respects, ma reconnaissance et tout mon amour. Puisse dieu vous préserver et vous procurer santé et bonheur.*

*Mohamed Ali*

*Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers,*

**À MON PÈRE : *ABDALLAH Ramadhan***

*Autant de phrases et d'expressions aussi éloquentes soit-elles ne sauraient exprimer ma gratitude et ma reconnaissance. Tu as su m'inculquer le sens de la responsabilité, de l'optimisme et de la confiance en soi face aux difficultés de la vie. Tes conseils ont toujours guidé mes pas vers la réussite. Ta patience sans fin, ta compréhension et ton encouragement sont pour moi le soutien indispensable que tu as toujours su m'apporter. Je te dois ce que je suis aujourd'hui et ce que je serai demain et je ferai toujours de mon mieux pour rester ta fierté et ne jamais te décevoir. Que Dieu le tout puissant te préserve, t'accorde santé, bonheur, quiétude de l'esprit et te protège de tout mal.*

**À MA MÈRE : *MOULAZEM Ajmia***

*Autant de phrases aussi expressives soient-elles ne sauraient montrer le degré d'amour et d'affection que j'éprouve pour toi. Tu m'as comblé avec ta tendresse et affection tout au long de mon parcours. Tu n'as cessé de me soutenir et de m'encourager durant toutes les années de mes études, tu as toujours été présente à mes cotés pour me consoler quand il fallait. En ce jour mémorable, pour moi ainsi que pour toi, reçoit ce travail en signe de ma vive reconnaissance et ma profonde estime. Puisse le tout puissant te donner santé, bonheur et longue vie afin que je puisse te combler à mon tour.*

**À MES SŒURS**

*Vous étiez toujours présents pour m'aider et m'encourager. Je vous souhaite une vie pleine de bonheur et de succès et que Dieu vous protège et vous garde*

**À MES AMIS ET MES COLLÈGUES**

*Puisse Dieu vous donner santé, bonheur, courage et surtout réussite*

*Imene*

# *Remerciements*

*Dieu Merci !*

*C'est avec plaisir que nous réservons ces quelques lignes en signe de gratitude et de profonde reconnaissance à l'égard de tous ceux qui nous ont aidés dans la réalisation de notre projet tutoré de fin d'étude.*

*Nous tenons particulièrement à exprimer en premier lieu nos profondes reconnaissances, nos respects et nos sincères remerciements à notre encadrant Monsieur Ben Salem Malek pour son assistance permanente, son soutien constant dans le suivi des différentes étapes de la réalisation du projet, ses conseils et critiques fructueux qui nous ont aimablement guidés dans nos efforts et aussi pour sa compréhension, sa disponibilité et sa bienveillance durant toute cette période de travail.*

*Nos respects les plus distingués aux membres du jury qui nous ont honorés d'avoir accepté d'examiner notre travail, nous les remercions pour leur attention et leur écoute.*

*Nous remercions aussi tout le corps enseignant et administratif qui a contribué à notre formation universitaire.*

# SOMMAIRE

<b>INTRODUCTION GÉNÉRALE .....</b>	<b>15</b>
<b>CHAPITRE 1 : CADRE GÉNÉRAL DU PROJET .....</b>	<b>16</b>
<b>1 Présentation du projet .....</b>	<b>16</b>
1.1 Présentation de la société d'accueil .....	16
1.2 Contexte du projet .....	17
<b>2 Étude préalable du projet .....</b>	<b>18</b>
2.1 Analyse de l'existant .....	18
2.1.1 Solutions en Tunisie .....	18
a. <i>La solution « Tobba.tn » .....</i>	18
b. <i>La solution « HelloConsult.tn » .....</i>	19
c. <i>La solution « med.tn » .....</i>	20
2.1.2 Solutions en France .....	21
a. <i>La solution « Leah » .....</i>	21
b. <i>La solution « Doctolib » .....</i>	22
c. <i>La solution « Medaviz » .....</i>	23
2.1.3 Solutions aux États-Unis .....	24
a. <i>La solution « Teledactyl » .....</i>	24
b. <i>La solution « Doctor on demand » .....</i>	25
2.1.4 Solutions en Grande-Bretagne « Babylon Health » .....	27
2.2 Critique de l'existant .....	27
2.3 Solution proposée .....	29
<b>3 Méthodologie de gestion du projet de développement .....</b>	<b>29</b>
3.1 Méthodologies existantes .....	29
3.1.1 Méthode Agile « Scrum » .....	29
3.1.2 Méthode Agile « eXtrême Programming » .....	30
3.1.3 Méthode de processus unifié « RUP » .....	31
3.1.4 Méthode de processus unifié « 2TUP » .....	32
3.2 Etude comparative entre les différentes méthodologies .....	33
3.3 Le choix de la méthodologie à suivre dans ce projet .....	33
<b>4 Mise en pratique du processus 2TUP .....</b>	<b>34</b>



<b>CHAPITRE 2 : ANALYSE ET SPÉCIFICATION DES BESOINS .....</b>	<b>38</b>
1 Identification des acteurs .....	38
2 Les besoins fonctionnels .....	38
2.1 Diagramme de cas d'utilisation général .....	39
2.2 Diagramme de cas d'utilisation détaillé .....	40
2.2.1 Analyse de cas d'utilisation « Gérer son profil » .....	40
2.2.2 Analyse du cas d'utilisation « S'authentifier » .....	41
2.2.3 Analyse du cas d'utilisation « Gérer dossier médical » .....	42
2.2.4 Analyse du cas d'utilisation « Gérer rendez-vous ».....	42
2.2.5 Analyse du cas d'utilisation « Gérer les commandes des patients » .....	43
3 Les besoins non fonctionnels .....	44
4 Diagramme de classe d'analyse .....	44
5 Diagramme De séquence d'analyse .....	46
5.1 Diagramme de séquence d'analyse « prendre rendez-vous » .....	46
5.2 Diagramme de séquence d'analyse « payer Consultation » .....	46
<b>CHAPITRE 3 : CONCEPTION DES BESOINS .....</b>	<b>49</b>
1 Conception architecturale .....	49
1.1 Patron de conception « Model-View-Controller » .....	49
1.2 Déploiement physique de l'architecture logicielle .....	50
2 Conception détaillée .....	50
2.1 Présentation du langage de modélisation UML .....	50
2.2 Le diagramme de classe de conception .....	53
2.3 Diagramme d'activité .....	56
2.3.1 Diagramme d'activité « s'authentifier » .....	56
2.3.2 Diagramme d'activité « envoyer résultat de l'analyse médicale » .....	57
2.4 Diagramme états-transitions .....	57
2.4.1 Diagramme d'état-transition « Consultation » .....	58
2.4.2 Diagramme d'état-transition « Rendez-vous » .....	58
2.5 Diagramme de séquence objet .....	58
2.5.1 Diagramme de séquence de cas d'utilisation « S'authentifier » .....	59
2.5.2 Diagramme de séquence de cas d'utilisation « Prendre rendez-vous » .....	60
<b>CHAPITRE 4 : RÉALISATION .....</b>	<b>65</b>
1 Environnement de développement .....	65

1.1	Architecture physique .....	65
1.2	Environnement matériel .....	65
1.3	Environnement logiciel .....	66
1.3.1	Outils logiciels .....	66
1.4	Technologies de développement .....	67
2	<b>Les composants applicatifs réalisés .....</b>	<b>67</b>
2.1	Interface Accueil .....	67
2.2	Interface Inscription .....	68
2.3	Interface Authentification .....	71
2.4	Interface de recherche d'un médecin .....	71
2.5	Interface Calendrier des rendez-vous du médecin .....	72
2.6	Interface Profil médecin .....	73
2.7	Notification d'un rendez-vous .....	74
2.8	Interface Calendrier du médecin .....	74
2.9	Interface Messagerie .....	75
2.10	Interface Discussion vidéo en ligne .....	76
	<b>CONCLUSION GÉNÉRALE .....</b>	<b>77</b>
	<b>Références bibliographiques .....</b>	<b>78</b>

# LISTE DES FIGURES

Figure 1: Le logo de la société d'accueil [1] .....	16
Figure 2: Page de recherche d'un dentiste sur « tobba.tn » [2] .....	19
Figure 3: Page d'accueil sur « HelloConsult.com » [3] .....	20
Figure 4: Page de recherche d'un dentiste sur « Med.tn » [4] .....	21
Figure 5: Page d'accueil « Leah » [5] .....	22
Figure 6: Page d'accueil « Doctolib » [6] .....	23
Figure 7: Page d'accueil « Medaviz » [7] .....	24
Figure 8: Page d'accueil « TELEDACTYL » [8] .....	25
Figure 9: Page d'accueil « Doctor on demand » [9] .....	26
Figure 10: Page d'accueil « Babylon » [10] .....	27
Figure 11: Logo de la plateforme à développer .....	29
Figure 12: Itération selon la méthode « AGILE SCRUM » [12] .....	30
Figure 13: Itération selon la méthode « Agile eXtreme Programming » [13] .....	31
Figure 14: Itération selon la méthode de processus unifié « RUP » [16] .....	32
Figure 15: Cycle de vie « 2TUP » [18] .....	35
Figure 16: Diagramme du cas d'utilisation général .....	40
Figure 17: Diagramme du cas d'utilisation « Gérer son profil » .....	41
Figure 18: Diagramme du cas d'utilisation « S'authentifier » .....	41
Figure 19: Diagramme du cas d'utilisation « Gérer dossier médical » .....	42
Figure 20: Diagramme du cas d'utilisation « Consulter rendez-vous » .....	43
Figure 21: Diagramme du cas d'utilisation « Gérer les commandes des patients » .....	43
Figure 22: Diagramme de classe d'analyse .....	45
Figure 23: Diagramme de séquence d'analyse « Prendre rendez-vous » .....	47
Figure 24: Diagramme de séquence d'analyse « Payer consultation » .....	48
Figure 25: Le modèle MVC [23] .....	49
Figure 26: Déploiement physique de l'architecture logicielle .....	50
Figure 27: Unified Modeling Language UML [26] .....	51
Figure 28: Diagramme de Kruchten [29] .....	53

Figure 29: Diagramme de classe MVC .....	55
Figure 30: Diagramme d'activité « S'authentifier » .....	56
Figure 31: Diagramme d'activité « envoyer résultat de l'analyse médicale » .....	57
Figure 32: Diagramme états-transitions « Consultation ».....	58
Figure 33: Diagramme états-transitions « Rendez-vous » .....	58
Figure 34: Diagramme de séquence du cas d'utilisation « S'authentifier ».....	60
Figure 35: Diagramme de séquence « Prendre rendez-vous » .....	62
Figure 36: Diagramme de séquence « Proposer rendez-vous en présentiel ».....	64
Figure 37: Interface Accueil.....	68
Figure 38: Interface première étape de création d'un compte.....	68
Figure 39: Interface deuxième étape de création d'un compte .....	69
Figure 40: Interface troisième étape de création d'un compte médecin .....	69
Figure 41: Interface quatrième étape de création d'un compte médecin .....	70
Figure 42: E-mail de vérification .....	70
Figure 43: Interface Authentification .....	71
Figure 44: Page de recherche sur un médecin .....	72
Figure 45: Interface Calendrier des rendez-vous du médecin .....	72
Figure 46: Interface Profil médecin .....	73
Figure 47: Suite interface Profil médecin .....	73
Figure 48: Notification d'un rendez-vous .....	74
Figure 49: Interface Calendrier médecin.....	74
Figure 50: Suite interface calendrier médecin.....	75
Figure 51: Interface Messagerie .....	75
Figure 52: Interface Discussion vidéo.....	76

# **LISTE DES TABLEAUX**

Tableau 1: Informations sur la société d'accueil « eNano-tech » .....	17
Tableau 2: Informations de base sur « tobba.tn » .....	18
Tableau 3: Informations de base sur « HelloConsult.tn » .....	19
Tableau 4: Informations de base sur « med.tn » .....	20
Tableau 5: Informations de base sur « Leah » .....	21
Tableau 6: Informations de base sur « Doctolib » .....	22
Tableau 7: Informations de base sur « Medaviz » .....	23
Tableau 8: Informations de base sur « teledactyl.com » .....	24
Tableau 9: Informations de base sur « Doctor On Demand » .....	26
Tableau 10: Informations de base sur « babylonhealth.com » .....	27
Tableau 11: Comparaison des solutions existantes .....	28
Tableau 12: Comparaison entre des méthodes de gestion des projets. ....	34
Tableau 13: Description du cas d'utilisation « S'authentifier ».....	59
Tableau 14: Suite description du cas d'utilisation « S'authentifier ».....	60
Tableau 15: Description du cas d'utilisation «Prendre rendez-vous».....	61
Tableau 16: Suite tableau 15, description du cas d'utilisation «Prendre rendez-vous»....	62
Tableau 17: Description du cas d'utilisation « Proposer un rendez-vous en présentiel ». 63	
Tableau 18: différents outils logiciels utilisés dans ce projet.....	66
Tableau 19: Différentes technologies utilisées dans notre projet.....	67

# LISTE DES ABRÉVIATIONS

- **ISIMM** : Institut Supérieur d’Informatique et de Mathématiques de Monastir,
- **MVC** : Model-View-Controller,
- **CMS** : Content Management System,
- **DMP** : Dossier Médical Partagé,
- **RDV** : Rendez-vous,
- **API** : Application Programming Interface,
- **XP** : Extreme Programming,
- **PU** : Processus Unifié,
- **RUP** : Rational Unified Process,
- **2TUP** : 2 Track Unified Process,
- **IDE** : Integrated Development Environment,
- **MVC** : Model View Controlling,
- **UML** : Unified Modeling Language,
- **TC** : TéléConsultation.



## INTRODUCTION GÉNÉRALE

Les idées naissent de leur temps, par conséquent, le monde est témoin récemment dans le domaine de la technologie médicale des nombreuses solutions locales et mondiales, pour résoudre les problèmes liés aux rendez-vous et surtout aux consultations médicales. D'autant plus que le monde est récemment témoin d'une épidémie mondiale qui a affecté négativement tous les secteurs, y compris le secteur public et privé du domaine de santé, et donc l'impact sur les pays et leur économie. Pour cette raison, les développeurs web et mobile cherchent à mettre à jour des applications médicales existantes pour répondre aux besoins des utilisateurs, et à développer des nouvelles applications pour confronter la situation imposée par la situation épidémique que le monde connaît récemment. C'est dans ce contexte que s'intègre notre projet de fin d'études qui consiste à réaliser une plateforme de téléconsultation médicale que nous avons nommée « doctena.tn ».

Le présent rapport présente les différentes étapes de la réalisation de notre projet qui s'étalera sur quatre chapitres : Le premier chapitre, intitulé « CADRE GENERAL DU PROJET », présente le cadre général de notre projet ainsi que la description de la solution proposée. La méthodologie de travail est également discutée dans ce chapitre. Le deuxième chapitre, intitulé « ANALYSE ET SPÉCIFICATION DES BESOINS », est consacré à la spécification des besoins fonctionnels, des besoins non fonctionnels et des besoins techniques. Le troisième chapitre, intitulé « CONCEPTION DES BESOINS », présente une conception de l'architecture logicielle ainsi que les différents diagrammes UML utilisés dans la phase de la modélisation statique et dynamique des besoins relatifs à la solution proposée. Le quatrième chapitre, intitulé « RÉALISATION », dédié à présenter les détails de l'implémentation de la solution proposée. Le rapport est clôturé par une conclusion générale présentant la synthèse de notre projet.

# CHAPITRE 1 : CADRE GÉNÉRAL DU PROJET

## Introduction

Le présent chapitre est une introduction à notre projet de fin d'études. En première partie, nous présentons le cadre de travail, puis la société d'accueil. Ensuite, nous décrivons la présentation du projet. Enfin, nous terminons ce chapitre par une analyse des différentes méthodologies de développement et nous faisons une comparaison pour choisir la méthode que nous allons adopter pour la réalisation de notre plateforme web.

## 1 Présentation du projet

Le présent travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention du diplôme de Licence Fondamentale en Sciences Informatiques de l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM). Ce stage a été effectué au sein de la société « eNano-tech », pendant une durée de trois mois.

### 1.1 Présentation de la société d'accueil



*Figure 1: Le logo de la société d'accueil [1]*

« eNano-Tech » est une agence spécialisée dans les solutions informatiques et de communications novatrices et à forte valeur ajoutée. L'expérience acquise au service de ses clients, lui a permis de se positionner rapidement sur le marché des nouvelles technologies de l'information et de la communication. Elle propose à ses clients une relation d'accompagnement et de conseil pour une réalisation réussie de leurs projets, à travers une large panoplie de services couvrant l'ensemble du cycle de vie d'un projet informatique, allant des études des besoins, à la spécification fonctionnelle et technique, au développement et intégration des solutions choisies, et au transfert du savoir-faire en passant par l'assistance à la mise en œuvre des systèmes fournis [1].

## CHAPITRE 1 : CADRE GÉNÉRAL DU PROJET

Elle a des équipes dédiées pour :

- Développement Web,
- Intégration e-Commerce,
- Développement .NET,
- Développement mobile.

Le Tableau 1 présente des informations générales sur cette entreprise.

*Tableau 1: Informations sur la société d'accueil « eNano-tech »*

Raison Sociale	eNano-Tech
Directeur technique	Habbachi Mahmoud
Adresse	Rue Sidi Saraj, Cité El Omrane Monastir, 5000
Téléphone	+216 70 035 774
Email	<a href="mailto:contact@enano-tech.com">contact@enano-tech.com</a>
Site web	<a href="http://www.enano-tech.com">http://www.enano-tech.com</a>

### 1.2 Contexte du projet

Nous avons pris en charge un sujet très important qui consiste à développer une plateforme de téléconsultation médicale en ligne. Ce travail a pour objectif de réaliser une plateforme web qui ne permettrait pas seulement au patient de téléconsulter un médecin en ligne à travers la prise d'un rendez-vous pour une consultation médicale en ligne, mais aussi la possibilité de la prise de rendez-vous pour une consultation médicale en présentiel, ou pour une consultation à domicile selon le choix du patient, avec une majoration au niveau du tarif de consultation.

Chaque projet informatique doit passer obligatoirement par trois phases :

- Etude préalable du projet consistant à jauger sa rentabilité,
- Spécification des besoins et conception générale et détaillée,
- Implémentation des besoins.

Notre projet consiste à étudier le marché des plateformes existantes en terme de consultation en ligne, afin de préparer un cahier de charges incluant les besoins fonctionnels permettant de

proposer une solution adéquate qui simplifierait et faciliterait des procédures archaïques pour jouir du droit universel à la santé.

## **2 Étude préalable du projet**

Nous ne pourrions réaliser ce travail sans avoir une vue et une idée claire sur l'existant dans le marché. Dans cette section, nous allons déterminer les points faibles et les points forts des solutions existantes pour pouvoir dégager les contraintes à respecter pendant la conception et la réalisation du ce projet.

### **2.1 Analyse de l'existant**

Les développeurs ne cessent de satisfaire les besoins de la communauté en offrant des solutions aux divers problèmes quotidiens, dont notamment la téléconsultation médicale. Dans ce que suit, nous présenterons quelques solutions disponibles en Tunisie et à l'étranger dans le but de dégager les principales fonctionnalités afin de concevoir notre plateforme.

#### **2.1.1 Solutions en Tunisie**

##### *a. La solution « Tobba.tn »*

*Tableau 2: Informations de base sur « tobba.tn »*

Logo	Description
	<b>Site web :</b> <a href="http://www.tobba.tn">http://www.tobba.tn</a> <b>Taille de l'entreprise :</b> 11-50 employés <b>Fondée en :</b> 2018 <b>Siège social :</b> Tunis, Tunisie

Le tableau 2 présente des informations de base de la solution « **Tobba.tn** », qui est la première plateforme de téléconsultation en ligne en Tunisie. Elle met en relation les patients avec des médecins pour mener une consultation vidéo et une analyse médicale à distance. La figure 2 présente une page de recherche d'un dentiste.

#### **✚ Le point fort**

- « **tobba.tn** » dispose d'un très haut niveau de sécurité pour la protection des données de santé.

#### **✚ Les points faibles**

Suite à une série de tests effectués sur cette application, nous avons remarqué :

- Plusieurs zones à l'échelle nationale ne sont pas prises en charge par l'application.

## CHAPITRE 1 : CADRE GÉNÉRAL DU PROJET

- Cette solution est limitée pour une application web ; aucune application mobile (Android ou IOS) n'est existante.

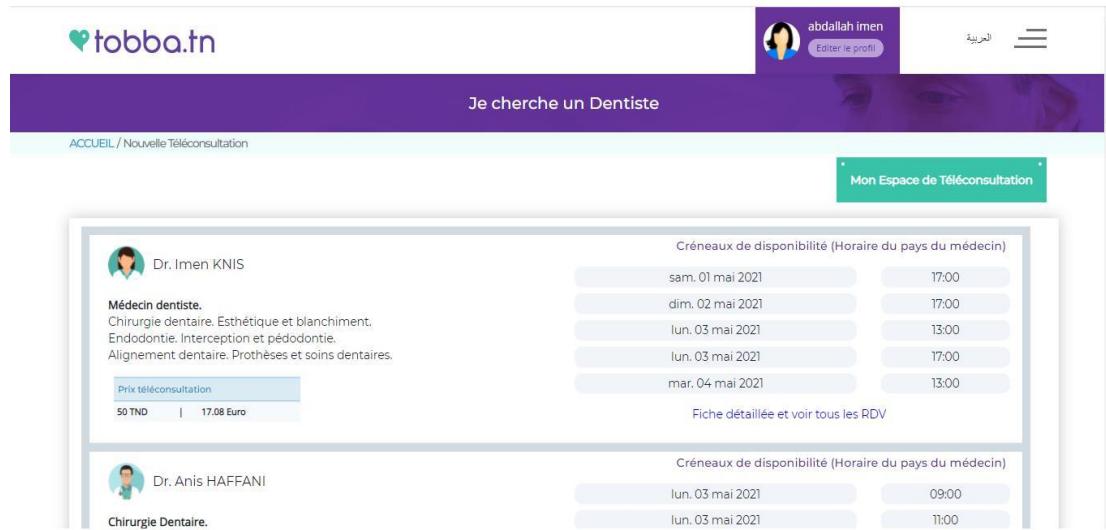


Figure 2: Page de recherche d'un dentiste sur « tobba.tn » [2]

### b. La solution « HelloConsult.tn »

Tableau 3: Informations de base sur « HelloConsult.tn »

Logo	Description
	Site web : <a href="https://www.helloconsult.tn/">https://www.helloconsult.tn/</a> Fondée en : 2015 Siège social : Tunis, Tunisie

Les informations de base de la solution « **HelloConsult.tn** » sont présentées dans le tableau 3. Cette plateforme est une solution réalisée par l'entreprise internationale « Help Group ». Elle vise les personnes qui ne veulent pas solliciter les urgences pour des maux du quotidien, mais qui veulent pouvoir consulter rapidement en toute sérénité avec une flexibilité maximale. La figure 3 présente la page d'accueil de cette solution.

### Les points forts

Plusieurs fonctionnalités sont présentées dans cette solution :

- Echange vidéo sécurisé et en haute définition,
- Suivi du patient par messagerie.

## Le point faible

- Cette solution est limitée pour une application web : aucune application mobile (Android ou IOS) n'est existante.



Figure 3: Page d'accueil sur « HelloConsult.com » [3]

### c. **La solution « med.tn »**

Tableau 4: Informations de base sur « med.tn »

Logo	Description
	<b>Site web:</b> <a href="https://www.med.tn/">https://www.med.tn/</a> <b>Taille de l'entreprise :</b> 11-50 employés <b>Fondée en :</b> 2016 <b>Siège social :</b> Tunis, Tunisie <b>Email:</b> <a href="mailto:contact@med.tn">contact@med.tn</a>

Le tableau 4 présente les informations de base de la solution « **med.tn** ». Cette solution est une plateforme innovante qui permet à un patient de trouver rapidement un médecin proche de lui et de prendre rendez-vous en ligne gratuitement et en un simple clic. Son objectif est d'améliorer l'accès aux soins en Tunisie en mettant à disposition des outils simples pour gérer la santé d'un patient et celle de ses proches. La figure 4 présente une interface de recherche sur un médecin.

## Les points forts

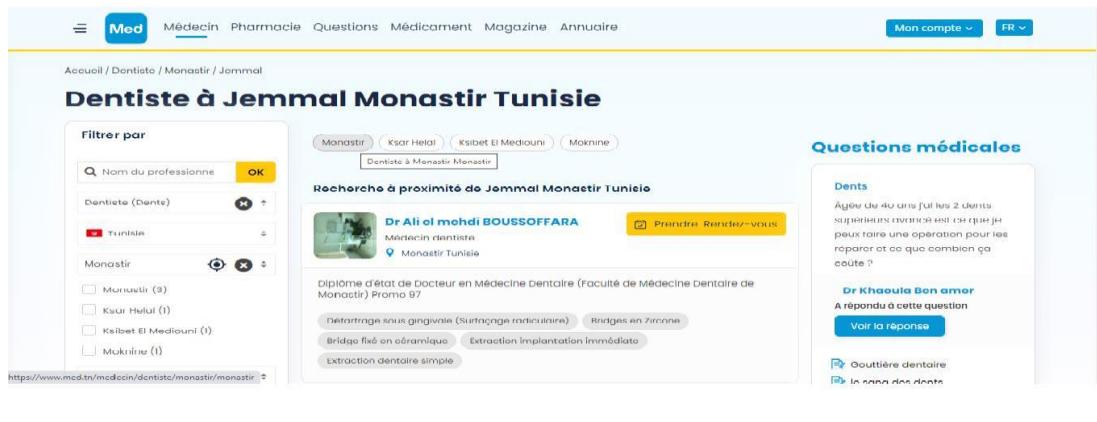
Plusieurs fonctionnalités sont présentées dans cette solution parmi lesquelles on cite :

## CHAPITRE 1 : CADRE GÉNÉRAL DU PROJET

- La recherche des médicaments,
- Les fonctionnalités du système sont facilement accessibles,
- L'interface est ergonomique et claire.

### Le point faible

- La plateforme ne permet pas au patient de faire une consultation en ligne avec un médecin.



The screenshot shows a search interface for dentists in Jemmal, Monastir, Tunisia. On the left, there's a filter panel with dropdowns for 'Nom du professionnel' (Dr Ali el mehdhi BOUSSOFFARA), 'Dentiste (Dents)' (Tunisie), and 'Monastir'. Below these are checkboxes for 'Monsatir (3)', 'Ksar Helal (1)', 'Ksibet El Mediouni (1)', and 'Moknina (1)'. In the center, a search result for 'Dr Ali el mehdhi BOUSSOFFARA' is displayed, including his profile picture, name, title ('Médecin dentiste'), location ('Monastir Tunisie'), and a 'Prendre Rendez-vous' button. To the right, a sidebar titled 'Questions médicales' shows a question about dental implants and a response from 'Dr Khaoula Ben amor'.

Figure 4: Page de recherche d'un dentiste sur « Med.tn » [4]

### 2.1.2 Solutions en France

#### a. La solution « Leah »

Tableau 5: Informations de base sur « Leah »

Logo	Description
	<p><b>Site web :</b> <a href="https://www.leah.care/">https://www.leah.care/</a></p> <p><b>Taille de l'entreprise :</b> 11-50 employés</p> <p><b>Fondée en :</b> 2016</p> <p><b>Siège social :</b> Levallois-Perret, Ile-de-France</p> <p><b>Email :</b> hello@leah.care</p>

Les informations de base de la solution « Leah » sont présentées dans le tableau 5. « Leah » est une solution de travail innovante qui permet au médecin de gérer ses rendez-vous et téléconsulter ses patients en ligne quand ils le souhaitent, depuis son cabinet ou même de chez lui par visioconférence. L'interface d'accueil de cette solution est présentée dans la figure 5.

## Les points forts

Plusieurs fonctions sont fournies dans cette solution :

- Cette solution possède une application Android et IOS,
- Payement en ligne de la téléconsultation disponible,
- Échange des documents avec médecin en toute sécurité : utilisation d'un protocole de sécurité.

## Le point faible

- Pas d'équipement, dispositifs médicaux connectés.



Figure 5: Page d'accueil « Leah » [5]

### b. La solution « Doctolib »

Tableau 6: Informations de base sur « Doctolib »

Logo	Description
	<b>Site web :</b> <a href="https://www.doctolib.fr">https://www.doctolib.fr</a> <b>Taille de l'entreprise :</b> 1 001-5 000 employés <b>Siège social :</b> Levallois-Perret, Île-de-France <b>Fondée en :</b> 2013 <b>Email :</b> <a href="mailto:julia@doctolib.fr">julia@doctolib.fr</a>

«**Doctoli**» qui est une plateforme qui propose un service de gestion en ligne des consultations pour les professionnels de santé et un service de prise de rendez-vous en ligne pour les patients. Les informations de base de cette solution sont présentées dans le tableau 6, et l'interface d'accueil de cette solution est présentée dans la figure 6.

## Les points forts

Plusieurs fonctionnalités sont présentées dans cette solution, parmi lesquelles on cite :

- Active sur les réseaux sociaux,
- Partage des documents sécurisés, comme l'envoi des ordonnances médicales : utilisation d'un protocole de sécurité,
- Messagerie sécurisée : utilisation d'un protocole de sécurité.



Figure 6: Page d'accueil « Doctolib » [6]

### c. La solution « Medaviz »

Tableau 7: Informations de base sur « Medaviz »

Logo	Description
	<b>Site web :</b> <a href="https://www.medaviz.com">https://www.medaviz.com</a> <b>Taille de l'entreprise :</b> 11-50 employés <b>Fondée en :</b> 2014 <b>Siège social :</b> Vannes, France <b>Email :</b> <a href="mailto:contact@medaviz.com">contact@medaviz.com</a>

Le tableau 7 présente les informations de base da la solution « **Medaviz** ». Cette solution accompagne les médecins dans l'organisation de leur temps au quotidien et les aide à améliorer l'offre et l'accès aux soins sur le territoire. La page d'accueil de cette solution est présentée dans la figure 7.

## Les points forts

Plusieurs fonctionnalités sont présentées dans cette solution :

- La solution intègre la récupération des données via des dispositifs médicaux connectés,
- Echange des documents sécurisé,
- Possibilité de paiement de la téléconsultation en ligne.

## Le point faible

- Pas d'agenda en ligne.



Figure 7: Page d'accueil « Medaviz » [7]

### 2.1.3 Solutions aux États-Unis

#### a. *La solution « Teledactyl »*

Tableau 8: Informations de base sur « teledactyl.com »

Logo	Description
	<b>Site web :</b> <a href="http://www.teledactyl.com">http://www.teledactyl.com</a> <b>Taille de l'entreprise :</b> 2-10 employés <b>Siège social :</b> Wellington, état de Floride <b>Fondée en :</b> 2017 <b>Email :</b> <a href="mailto:contact@teledactyl.com">contact@teledactyl.com</a>

## CHAPITRE 1 : CADRE GÉNÉRAL DU PROJET

Le tableau 8 présente les informations de base de la solution « **Teledactyl** », qui est un système de santé virtuel qui réduit les coûts en utilisant la technologie pour orienter les patients vers les options les plus rentables de leur réseau. La figure 8 présente la page d'accueil de cette solution.

### **Les points forts**

Plusieurs fonctionnalités sont présentées dans cette solution parmi lesquelles on évoque:

- Les Rendez-vous sont synchronisés avec Google,
- Possibilité d'inviter un membre de la famille ou un ami lors d'une téléconsultation,
- L'importation et le téléchargement des fichiers sont sécurisés,
- Enregistrement de sessions vidéo nécessaire de la part du médecin pour le dossier médical électronique.

### **Les points faibles**

- Le temps de réponse est un peu lent

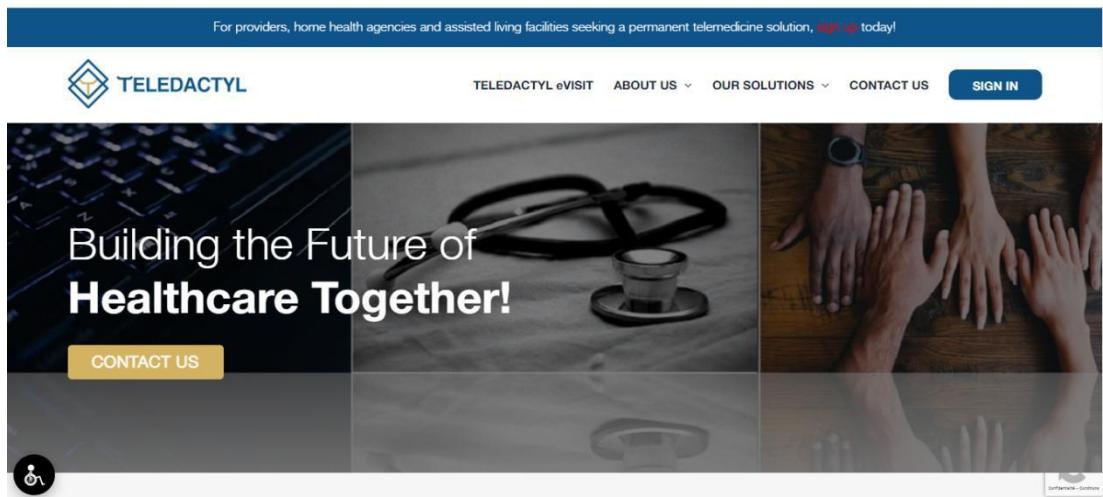


Figure 8: Page d'accueil « **TELEDACTYL** » [8]

#### **b. La solution « Doctor on demand »**

« **Doctor On Demand** », le principal fournisseur de soins virtuels des Etats Unis (tableau 9), réinvente à quoi ressemblent les soins de santé dans le monde d'aujourd'hui. La plate-forme nationale de soins de santé de « Doctor On Demand » place le patient au premier plan en donnant accès aux médecins, aux psychiatres, aux thérapeutes et à une équipe de soins de prestataires cliniques et de coordonnateurs via des visites vidéo, la voix et la messagerie.

## CHAPITRE 1 : CADRE GÉNÉRAL DU PROJET

Grâce à son application mobile et à son site Web classés 5 étoiles, les patients peuvent accéder à des soins coordonnés et de haute qualité dans les 50 États, 24 heures sur 24, 7 jours sur 7. La figure 9 présente la page d'accueil de cette solution.

*Tableau 9: Informations de base sur « Doctor On Demand »*

Logo	Description
	<b>Site web :</b> <a href="http://www.doctorondemand.com">http://www.doctorondemand.com</a> <b>Taille de l'entreprise :</b> 201-500 employés <b>Siège social :</b> San Francisco, CA <b>Fondée en :</b> 2013 <b>Email :</b> <a href="mailto:support@doctorondemand.com">support@doctorondemand.com</a>

### Les points forts

Plusieurs fonctionnalités sont présentées dans cette solution parmi lesquelles on mentionne :

- La possibilité d'une consultation IMMEDIATE dans certains cas d'urgence.
- Lors de test de l'application nous avons trouvé que :
  - Les fonctionnalités du système sont facilement accessibles.
  - L'interface est ergonomique et claire.

### Le point faible

- Le système prend beaucoup de temps pour la recherche des médecins : temps de réponse relativement lent.



*Figure 9: Page d'accueil « Doctor on demand » [9]*

### **2.1.4 Solutions en Grande-Bretagne « *Babylon Health* »**

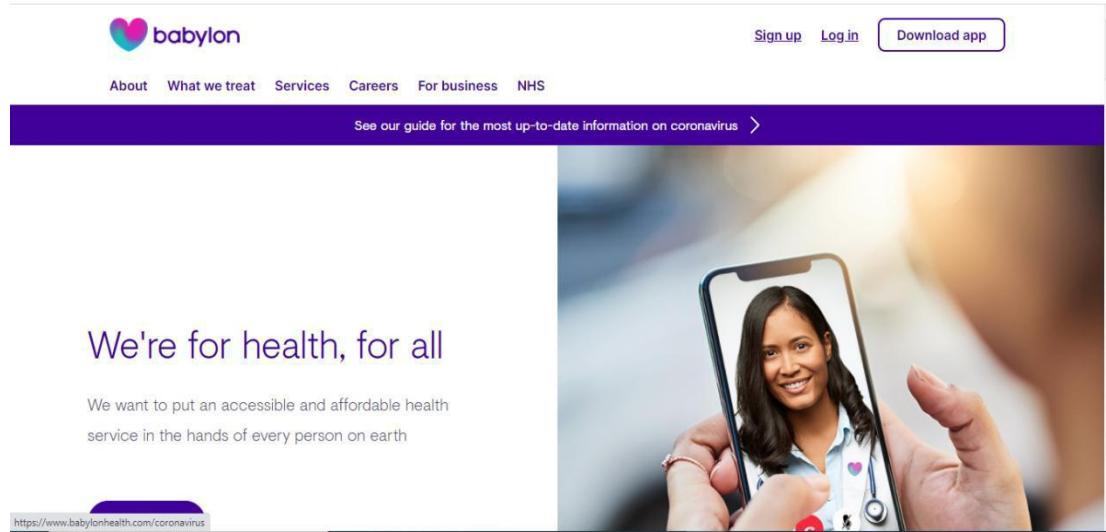
*Tableau 10: Informations de base sur « babylonhealth.com »*

Logo	Description
	<b>Site web :</b> <a href="http://www.babylonhealth.com/">http://www.babylonhealth.com/</a> <b>Taille de l'entreprise :</b> 1 001-5 000 employés <b>Fondée en :</b> 2013 <b>Siège social :</b> Londres, Royaume-Uni <b>Email :</b> <a href="mailto:john.smith@babylonhealth.com">john.smith@babylonhealth.com</a>

Selon l'Organisation mondiale de la santé, la moitié du monde n'a pas accès aux services de santé essentiels. C'est pourquoi chez « **Babylon** », la combinaison de la technologie et l'expertise médicale a pour but de rapprocher les médecins et les gens. Le tableau 10 présente les informations de base de cette solution. Cette plateforme donne aux gens un accès 24 heures sur 24 à des services de santé holistiques abordables - souvent gratuits. La figure 10 présente la page d'accueil de la solution étudiée.

#### **✚ Le point fort**

- Cette solution possède une application Android et IOS.



*Figure 10: Page d'accueil « Babylon » [10]*

### **2.2 Critique de l'existant**

Cette section nous mène à une étude basée sur les observations des différentes plateformes web spécialisées dans le domaine de téléconsultation médicale en ligne. Cette étude nous permet de dégager les points forts et les points faibles afin de déterminer les besoins. Dans le

## CHAPITRE 1 : CADRE GÉNÉRAL DU PROJET

Tableau 11, nous faisons une petite comparaison entre les solutions existantes en se basant sur un ensemble de critères comme celui de la sécurité, et sur un ensemble de fonctionnalités parmi lesquelles on cite : la gestion du cabinet, la gestion de la disponibilité etc. Pour cela nous avons utilisé des symboles différents pour marquer chaque solution, telle que la coche en rouge pour les solutions tunisiennes, la coche en bleu pour les solutions françaises, la coche verte pour les États-Unis et la noir pour Royaume-Uni.

*Tableau 11: Comparaison des solutions existantes*

Solution Critère	Tobba.tn	HelloConsult. tn	Med.tn	Leah	Doctolib	Medavis	Teledactyl	Doctor On Demand	Babylon Health
Fonctionnalité 1 : Gestion du cabinet	✓				✓				
Fonctionnalité 2 : Espace de messagerie		✓		✓			✓	✓	
Fonctionnalité 3 : Téléexpertise						✓			✓
Fonctionnalité 4 : Gestion de la disponibilité	✓	✓	✓		✓		✓	✓	
Fonctionnalité 5 : Recherche d'une pharmacie			✓						
Fonctionnalité 6 : recherche d'un laboratoire	✓		✓						
Fonctionnalité 7: Consultation d'urgence		✓						✓	✓
Fonctionnalité 8: Prise de rendez-vous en ligne	✓	✓	✓		✓		✓	✓	✓
Fonctionnalité 9: Consultation en ligne	✓	✓		✓				✓	✓
Chatbot									
Critère de Sécurité				✓					
Equipement IoT						✓			

## 2.3 Solution proposée

Suite à une étude comparative des solutions nationales et internationales existantes, on a remarqué que chaque solution dégage des forces et des faiblesses, ce qui représente une pression et une difficulté pour l'utilisateur dans son choix de la solution pour laquelle il opterait. Et c'est justement dans cette optique que nous proposons notre nouvelle solution qui permettrait de regrouper et de satisfaire tous les besoins cités au niveau du cahier des charges. On a profité des fonctionnalités des solutions existantes en essayant de combler leurs insuffisances. C'est pour cela nous avons fait notre comparaison à l'échelle internationale pour s'inspirer plus des fonctionnalités inexistantes dans les solutions nationales. Ainsi, nous estimons avoir construit une solution qui pourrait concurrencer les solutions existantes en Tunisie.



*Figure 11: Logo de la plateforme à développer*

## 3 Méthodologie de gestion du projet de développement

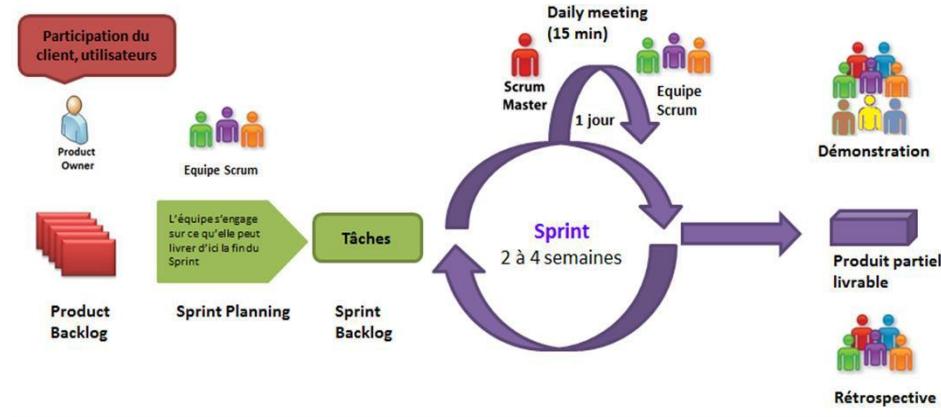
La méthodologie est un système de pratique, de techniques, de procédures et de règles utilisées pour aider les gestionnaires à planifier, initier et mettre en œuvre des projets. Mais différentes méthodes profiteront à différents projets, et tous les modèles de gestion de projet ne sont pas efficaces pour différentes tâches. Afin de savoir quelle méthode nous conviendrait le mieux, nous nous devions de nous familiariser avec ces méthodes et leurs différences.

### 3.1 Méthodologies existantes

#### 3.1.1 Méthode Agile « Scrum »

La méthode Scrum est la plus utilisée parmi les méthodes Agiles existantes, qui tire son nom du terme anglais « Mêlée », en référence à la mêlée du rugby ; elle met l'accent sur l'esprit d'équipe et sur le fait que tous les acteurs doivent avancer dans la même direction pour atteindre un même objectif. Scrum repose sur une intense collaboration de l'équipe qui se

focalise sur une partie limitée et maîtrisable des fonctionnalités à réaliser [11]. La figure 12 présente une itération selon la méthode Scrum.



*Figure 12: Itération selon la méthode « AGILE SCRUM » [12]*

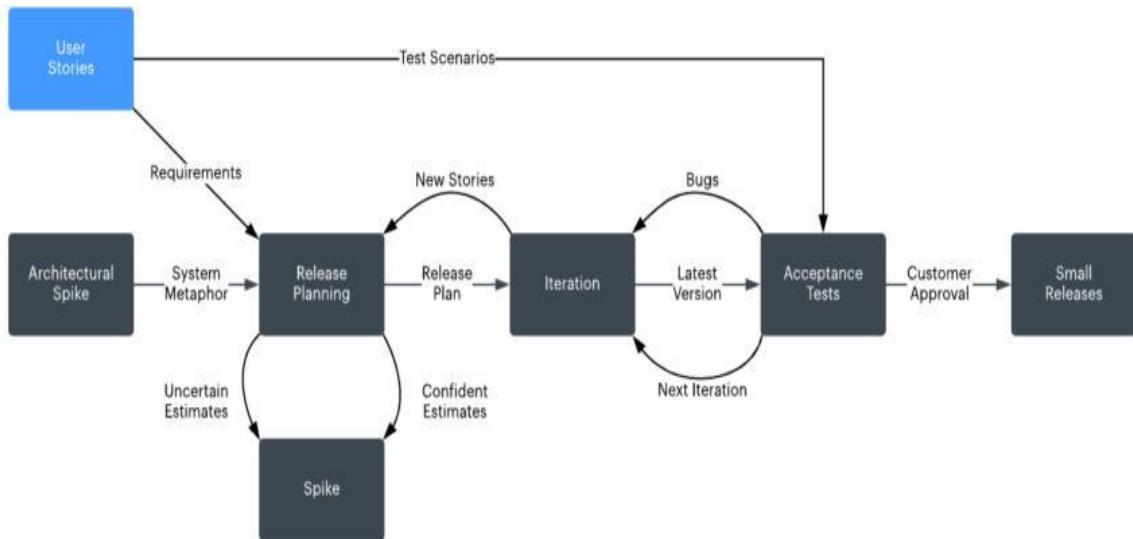
Un projet utilisant la méthode Scrum se base sur des itérations appelées Sprints que se succèdent. Un sprint dure de 2 à 4 semaines. Chaque Sprint commence par une réunion de planification appelée « Sprint planning ». C'est à l'issue de cette réunion, que l'équipe découvre l'objectif du Sprint et chaque besoin qui est décomposé en plusieurs tâches. Durant un Sprint, des réunions quotidiennes appelés « Daily meeting » de moins de 15 minutes permettent à chaque membre de prendre la parole et de faire le point sur ce qu'il a fait, ce qu'il compte faire et les différentes difficultés rencontrées. Pendant ce Sprint, l'équipe développe l'ensemble des besoins embarqués dans ce sprint en analysant, concevant, développant, testant et intégrant les fonctionnalités. Chaque Sprint se termine par une revue du Sprint appelée « Démonstration » durant laquelle les besoins réalisés sont évalués par le Product Owner en présence du client qui valide ou pas le produit partiel et modifie par la suite le backlog du produit. Une réunion « Rétrospective » se fait généralement après la démonstration. Elle a pour but d'examiner ce qui a bien fonctionné, ce qui ne l'a pas été et ce qui peut être amélioré [12].

### **3.1.2 Méthode Agile « eXtreme Programming »**

La méthodologie « eXtreme Programming » (ou XP) est une méthode de gestion de projet qui applique à l'extrême les principes du développement agile, c'est-à-dire se concentrer sur les besoins des clients, mettre en place un développement itératif et l'intégration continue [12].

La figure 13 présente une itération selon la méthode « eXtreme Programming ».

## Extreme Programming (XP) Methodology



*Figure 13: Itération selon la méthode «Agile eXtreme Programming» [13]*

Le but principal de XP est de minimiser les coûts lors des changements nécessaires dans une étape ultérieure du projet, une situation bien concrète dans de nombreux projets. Les principales pratiques à cette fin sont : développement focalisé sur le testage, programmation en pair, intégration continue, avancement par des petits incrémentés, design simple, copropriété du code, des standards de codage et une cadence de travail soutenable [14].

### 3.1.3 Méthode de processus unifié « RUP »

La méthode RUP (Rational Unified Process), est une des émanations de la méthode PU, qui s'attache à donner un cadre précis au développement du logiciel.

C'est une méthode générique, itérative et incrémentale assez lourde mais qui s'adapte très facilement aux processus et aux besoins du développement [15]. La figure 14 présente une itération selon la méthode RUP.



*Figure 14: Itération selon la méthode de processus unifié « RUP » [16]*

Le RUP divise le processus de développement en quatre phases, et chaque phase en itérations.

- **Initialisation :** définition de l'objectif du projet et préparation d'un contrat de réalisation.
- **Élaboration :** planification du projet, spécification de la solution proposée, proposition d'une architecture de base.
- **Construction :** développement du produit.
- **Transition :** transition du produit chez les utilisateurs.

Ces 4 phases sont elles-mêmes divisées en N itération, à la fin desquelles il y a une version du produit livrable [17].

### **3.1.4 Méthode de processus unifié « 2TUP »**

2TUP (2 Tracks Unified Process) est un processus de développement logiciel qui implémente le processus unifié (UP). Chacune des étapes du cycle découle des précédentes. Il préconise un cycle de vie en Y. A partir de la capture des besoins fonctionnels, on définit plusieurs cas d'utilisation représentant chacun un incrément du cycle de développement. Cette méthode favorise des formes de recherche de qualité et de performance intéressantes, telle que le service réutilisation. Le 2TUP présente plusieurs caractéristiques :

- **Un processus itératif et incrémental :** Une itération est une séquence distincte d'activités qui produit des améliorations ou des évolutions du système. Un incrément est la différence entre 2 itérations successives. De plus, le suivi des incréments constitue un excellent contrôle des couts et délais.

- **Un processus piloté par les exigences des utilisateurs :** Mettre l'accent sur l'exigence des utilisateurs.
- **Un processus piloté par les risques :** Incapacité à intégrer les technologies...
- **Un processus de modélisation avec UML :** Un ensemble de diagrammes de UML est utilisé dans des étapes de processus en Y.
- **Un processus centré sur l'architecture :** Le système est décomposé en modules pour des besoins de maintenabilité et d'évolutivité.
- **Un processus orienté vers les composants :** Les regroupements de concepts définissant des packages et des composants dans le modèle. Leur réutilisation peut se situer à tous les niveaux [18].

### **3.2 Etude comparative entre les différentes méthodologies**

A travers cette section, nous mettons l'accent sur quatre différentes méthodes qui sont :

- Scrum,
- eXtreme Programming (XP),
- Rational Unified Process (RUP),
- Two Tracks Unified Process (2TUP).

Ainsi, nous déterminons leurs principaux points forts et points faibles afin de déterminer la meilleure méthode qui sera utilisée dans notre travail. Le tableau 12 présente une comparaison entre ces méthodes de gestion de projet de développement.

### **3.3 Le choix de la méthodologie à suivre dans ce projet**

Dans le cadre de ce projet nous ne partageons pas une tâche ou une sous-tâche d'un grand projet avec une équipe de développement, mais plutôt nous travaillons sur toutes les parties et sur toutes les tâches qui composent notre projet. En effet, notre solution est basée sur un processus de développement bien défini qui commence par la détermination des besoins fonctionnels attendus du système, en passant par la conception, et en finit par le codage. C'est pour cela qu'on a besoin d'un cycle de développement qui dissocie les aspects techniques et les aspects fonctionnels tout en commençant par une étude préliminaire.

Notre choix s'est alors porté vers la méthode 2TUP. Elle est caractérisée par sa nouveauté et elle respecte le cadre de notre projet.

*Tableau 12: Comparaison entre des méthodes de gestion des projets.*

Méthode	Points forts	Points faibles
<b>Agile Scrum</b>	<ul style="list-style-type: none"> <li>✓ Itératif</li> <li>✓ Mise à jour des priorités</li> <li>✓ Résultat conforme aux attentes</li> </ul>	<ul style="list-style-type: none"> <li>✗ Taille de l'équipe</li> <li>✗ Manque de pratique pour obtenir le ‘DONE’ dans le contexte logiciel</li> </ul>
<b>eXtreme Programming(XP)</b>	<ul style="list-style-type: none"> <li>✓ Itératif et planification souple</li> <li>✓ Logiciel stable grâce à des tests continus</li> </ul>	<ul style="list-style-type: none"> <li>✗ Possible uniquement avec une gestion de la version.</li> <li>✗ Nécessite des experts.</li> </ul>
<b>Rational Unified Process (RUP)</b>	<ul style="list-style-type: none"> <li>✓ Gestion des risques dans les projets.</li> <li>✓ Cadre propice à la réutilisation</li> </ul>	<ul style="list-style-type: none"> <li>✗ Coût de personnalisation souvent élevé</li> <li>✗ Vision non évidente ni immédiate</li> </ul>
<b>Two Unified Track Process (2TUP)</b>	<ul style="list-style-type: none"> <li>✓ prise en compte de la gestion et du technologique</li> <li>✓ Itérative et incrémentale</li> <li>✓ indépendante de la taille du projet</li> <li>✓ pilotée par le risque</li> </ul>	<ul style="list-style-type: none"> <li>✗ Ne propose pas de documents types</li> <li>✗ Superficiel sur les phases situées en amont et en aval du développement</li> </ul>

## 4 Mise en pratique du processus 2TUP

Dans notre projet, nous avons utilisé la démarche 2TUP (Two Tracks Unified Process). Le processus 2TUP est un processus de développement logiciel qui implémente le processus uniifié, et qui a pour but d'apporter une réponse aux contraintes des changements fonctionnels et techniques qui s'imposent aux systèmes d'information. Chacune des étapes du cycle découle des précédentes. La figure 15 présente une itération selon la méthode 2TUP.

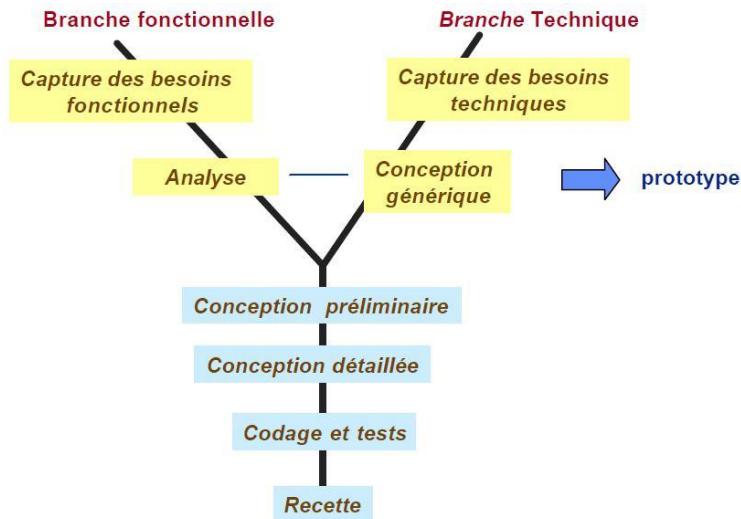


Figure 15: Cycle de vie « 2TUP » [18]

2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels. Il part du constat que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement, suivant un axe fonctionnel et un axe technique. Il distingue ainsi deux branches : une branche fonctionnelle et une branche technique, dont les résultats sont fusionnés pour réaliser le système. On obtient un processus de développement en Y comme l'illustre la figure 15. Il faut noter que le processus 2TUP commence d'abord par une étude préliminaire. Dans cette étude, il s'agit d'identifier les acteurs qui vont interagir avec le système, les messages qu'échangent les acteurs et le système, puis à produire un cahier de charges et enfin à modéliser le contexte.

### 

Elle vise la capture des besoins fonctionnels et l'analyse des spécifications fonctionnelles de manière à déterminer ce que va réaliser le système en terme de métier. C'est ici qu'on identifie et dégage toutes les fonctionnalités du système à réaliser.

- **Capture des besoins fonctionnels**

La spécification des besoins va nous permettre d'avoir une meilleure approche des utilisateurs, des fonctionnalités et de la relation entre les deux. Elle sera sous forme de cas d'utilisation. Pour cela nous allons procéder ainsi :

- Identification des acteurs du nouveau système,
- Identification des cas d'utilisation,
- Description des cas d'utilisation,

- Regroupement des cas d'utilisation en paquetages.

- **Analyse**

Vise à définir et à justifier la solution optimale qui répondra aux exigences du projet et qui tiendra compte de ses contraintes, de même qu'à établir le budget, le calendrier, les contrôles et les critères d'évaluation qui seront appliqués pour l'approbation préliminaire du projet.

 **Branche technique ou « droite »**

Elle permet la capture des besoins non fonctionnels. Il s'agit essentiellement des contraintes que l'application doit prendre en compte comme par exemple les contraintes d'intégration, les contraintes de développement et les contraintes de performances.

- **Capture des besoins techniques**

La capture des besoins techniques recense toutes les contraintes et les choix dimensionnant la conception du système.

La capture des besoins techniques se présente comme suit :

- Capture des spécifications logicielles.
- Capture des spécifications liées à la configuration matérielle.

- **Conception générique**

Il s'agit du découpage en composants nécessaires à la construction de l'architecture technique. Il est généralement conseillé de réaliser un prototype pour assurer la validité de l'architecture. Cette étape permet de minimiser l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles.

 **Phase conception – réalisation**

Caractérisée par deux étapes principales :

- **L'étape conception préliminaire**

Cette étape permet de produire le modèle de conception système. Ce dernier organise le système en composants délivrant les services techniques et fonctionnels, ce qui induit le regroupement des informations des branches techniques et fonctionnelles.

- **L'étape conception détaillée**

Elle permet d'étudier comment réaliser chaque composant. Le résultat fournit l'image prête [19].

## **Conclusion**

Dans ce chapitre, nous avons présenté la société qui nous a accueillis durant notre projet. Puis, nous nous sommes concentrés sur l'exposition du cadre général du projet ainsi que l'étude de l'existant et la solution proposée pour enfin recenser une étude comparative entre les méthodologies de développement existantes.

Dans le chapitre suivant, nous allons mettre le point sur la phase de l'analyse et la spécification des besoins.

# CHAPITRE 2 : ANALYSE ET SPÉCIFICATION DES BESOINS

## Introduction

Après avoir présenté le cadre général du projet, nous allons dans ce qui va suivre entrer dans le vif du projet. Ce chapitre va nous permettre d'identifier tous les acteurs intervenants dans la solution proposée, toutes les fonctionnalités pour tous les acteurs et à minima déterminer le contexte et les attentes du projet à travers le dénombrement des besoins fonctionnels et l'invocation des exigences traduites par les besoins non fonctionnels.

### 1 Identification des acteurs

Les acteurs du système sont des entités qui définissent les rôles joués par les utilisateurs ou les systèmes interagissant avec le système de modélisation. La solution proposée représente quatre acteurs :

-  **Médecin,**
-  **Patient,**
-  **Laboratoire médical,**
-  **Pharmacie.**

### 2 Les besoins fonctionnels

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Elle formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire. Elle est complétée au niveau de la branche droite du Y par la capture des besoins techniques, et elle prépare l'étape suivante de la branche gauche : l'analyse [20].

Définissons à présent les fonctions du point de vue de l'application.

 **Médecin :** Il doit être capable d'effectuer un ensemble de tâches particulières :

- Consulter profil,
- Consulter un patient en ligne,
- Communiquer avec patient,
- Proposer un rendez-vous en présentiel,
- Valider ordonnances patients,

- Gérer rendez-vous,
- Gérer dossier médical.

 **Patient:** Il doit être capable d'effectuer un ensemble de tâches:

- Consulter profil,
- Consulter un médecin en ligne,
- Commander et suivre l'achat des médicaments,
- Trouver une pharmacie à proximité,
- Trouver un laboratoire à proximité (analyse, radiologie...),
- Procéder au paiement d'une consultation médicale,
- Communiquer avec médecin,
- Demander un rendez-vous d'une consultation en ligne,
- Demander un rendez-vous d'une consultation en présentiel,
- Demander un rendez-vous d'une consultation à domicile.

 **Laboratoire :** qui est responsable de ces tâches :

- Consulter profil,
- Consulter les examens des patients (analyse, radiologie...).

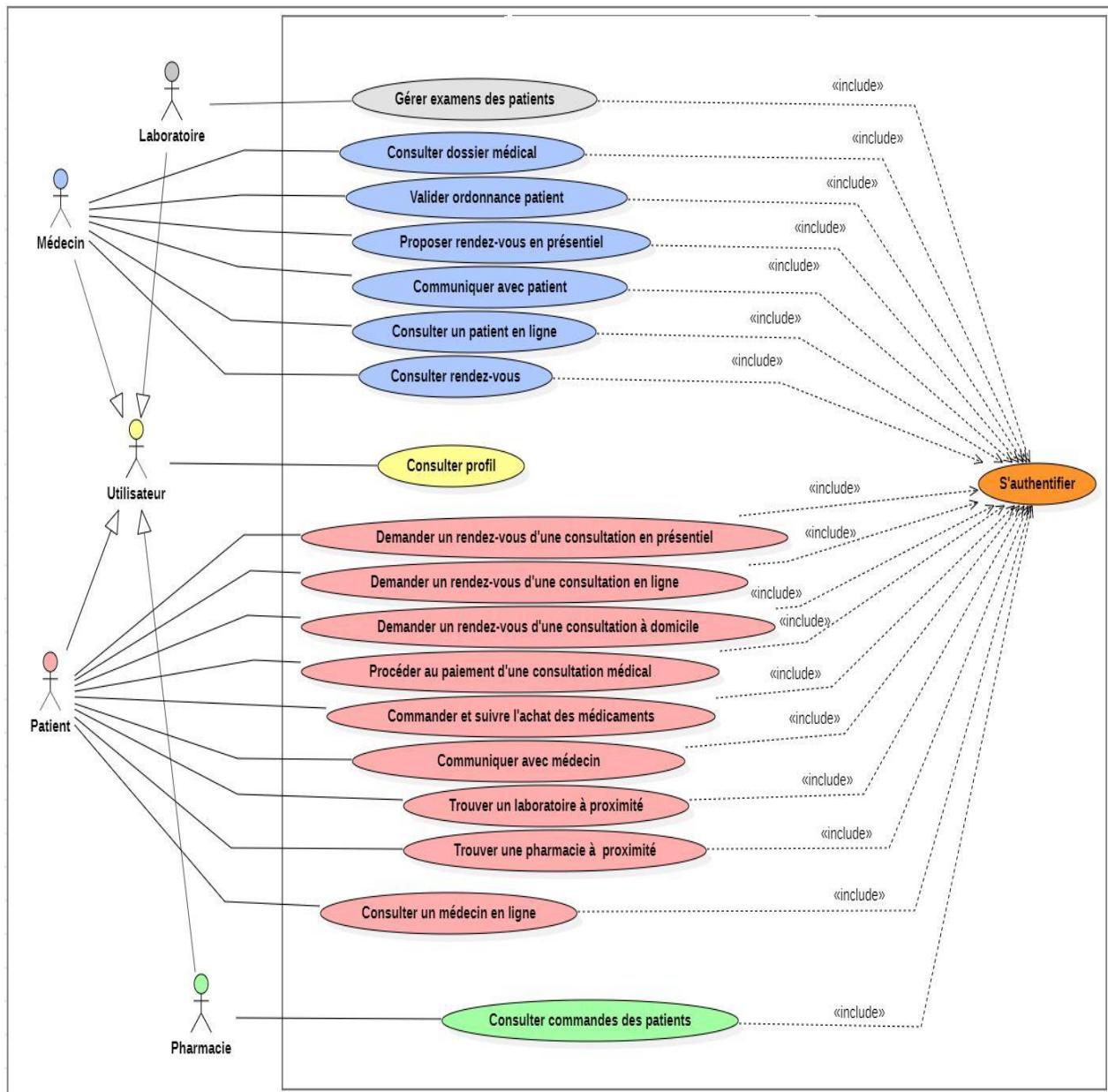
 **Pharmacie :** qui doit être capable de réaliser ces tâches :

- Consulter son profil,
- Gérer les commandes des patients (Ordonnances médicales).

## 2.1 Diagramme de cas d'utilisation général

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisations permettent d'exprimer les besoins des utilisateurs d'un système, ils ont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique [21].

Pour développer des cas d'utilisation, nous diviserons ces derniers en parties : nous présenterons le diagramme de cas d'utilisation général, donc des cas d'utilisation basés sur la distribution effectuée au niveau fonctionnel de l'application. La figure 16 présente le diagramme de cas d'utilisation général relatif à notre solution.



*Figure 16: Diagramme du cas d'utilisation général*

## 2.2 Diagramme de cas d'utilisation détaillé

### 2.2.1 Analyse de cas d'utilisation « Gérer son profil »

Le cas d'utilisation « Gérer son profil » est commun pour tous les acteurs de la plateforme :

- Médecin,
- Patient,
- Laboratoire,
- Pharmacie.

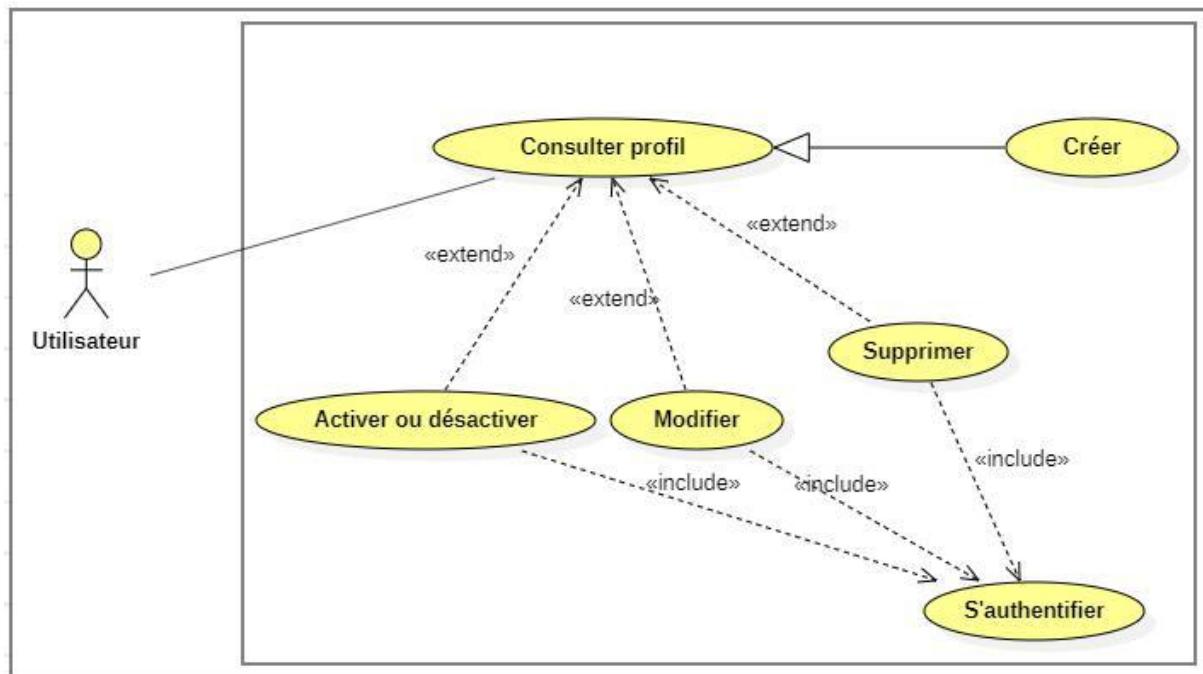


Figure 17: Diagramme du cas d'utilisation « Gérer son profil »

## 2.2.2 Analyse du cas d'utilisation « S'authentifier »

Le cas d'utilisation « S'authentifier » est commun pour tous les acteurs de la plateforme :

- Médecin,
- Patient,
- Laboratoire,
- Pharmacie.

En effet, chaque utilisateur a besoin de s'authentifier, de saisir son login et son mot de passe, pour accéder à son profil et pour bénéficier des fonctionnalités de la plateforme. La figure 18 présente le diagramme de cas d'utilisation « Sauthentifier ».

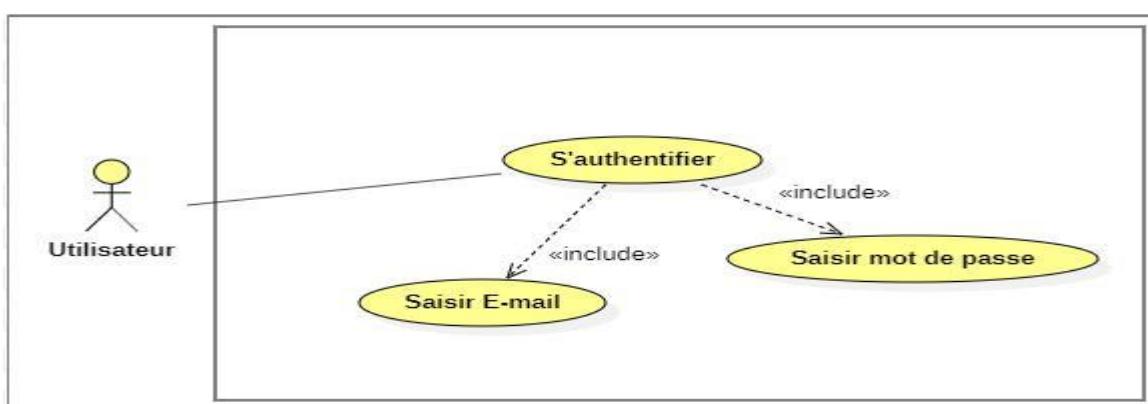
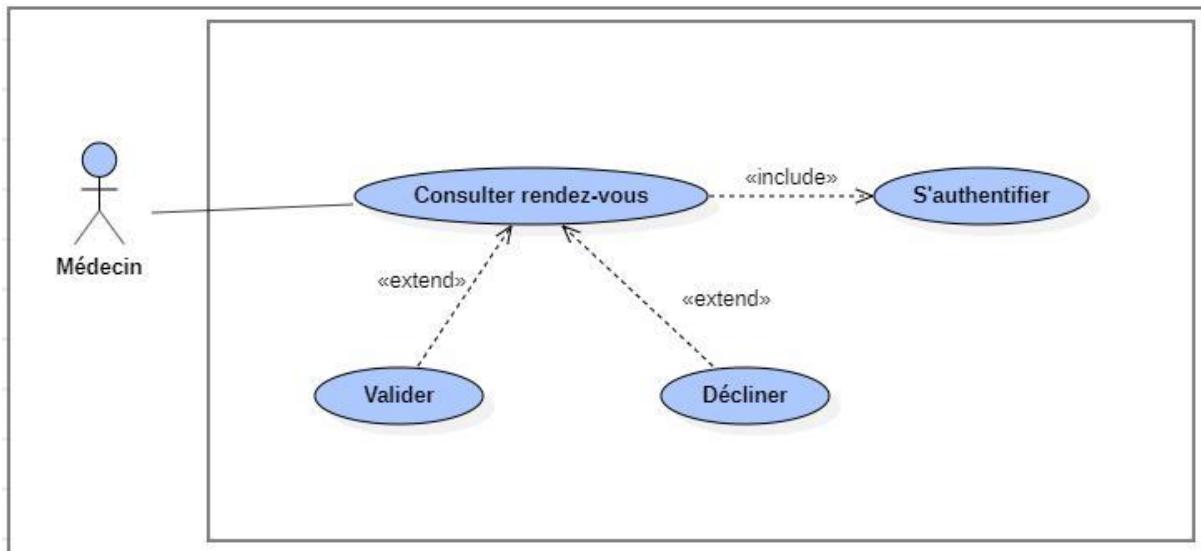


Figure 18: Diagramme du cas d'utilisation « S'authentifier »

### 2.2.3 Analyse du cas d'utilisation « Gérer dossier médical »

Dans le cadre de la gestion d'un dossier médical d'un patient, l'acteur « médecin » (figure 19), peut soit créer un dossier médical pour la première fois, ou bien le modifier suite à sa création ; Et avant d'effectuer ces cas de besoin, il doit s'authentifier en fournissant les informations d'authentification (Login et mot de passe).



*Figure 19: Diagramme du cas d'utilisation « Gérer dossier médical »*

### 2.2.4 Analyse du cas d'utilisation « Gérer rendez-vous »

Dans le cadre de la gestion des rendez-vous, l'acteur « médecin » (figure 20), peut soit valider le rendez-vous planifié par le patient, sinon le décliner selon sa disponibilité ; avant d'effectuer ces cas de besoin, il doit s'authentifier en fournissant les informations d'authentification (Login et mot de passe).

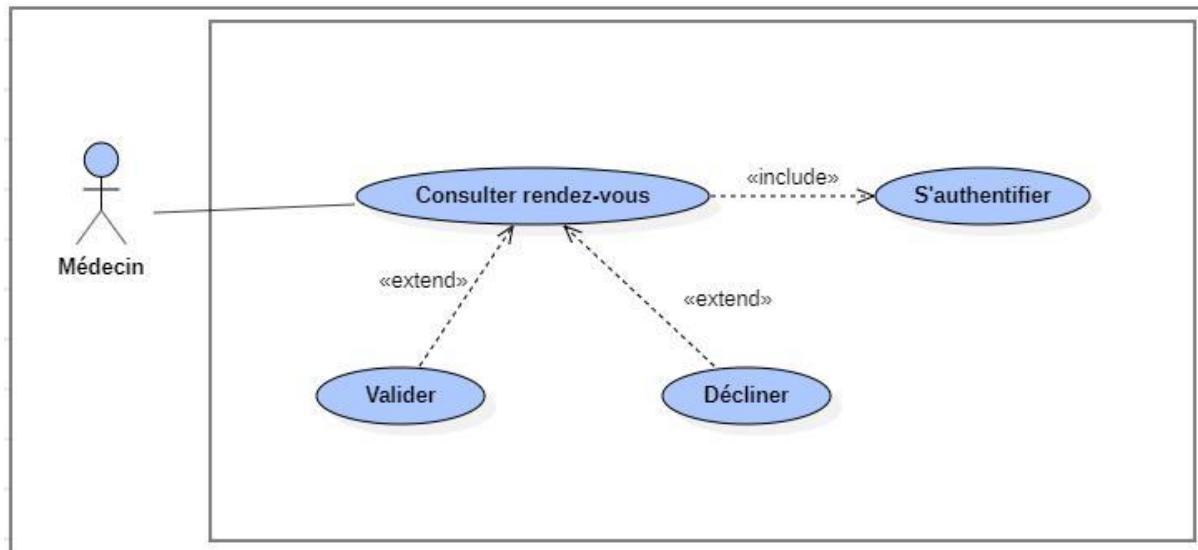


Figure 20: Diagramme du cas d'utilisation « Consulter rendez-vous »

### 2.2.5 Analyse du cas d'utilisation « Gérer les commandes des patients »

Dans le cadre de la gestion des commandes des patients, l'acteur « pharmacie » (figure 21), peut soit valider les commandes des médicaments passées par les patients et les acheminer vers le livreur, soit garder ces commandes en attente ; il doit au préalable s'authentifier en insérant les informations requises (Login et mot de passe).

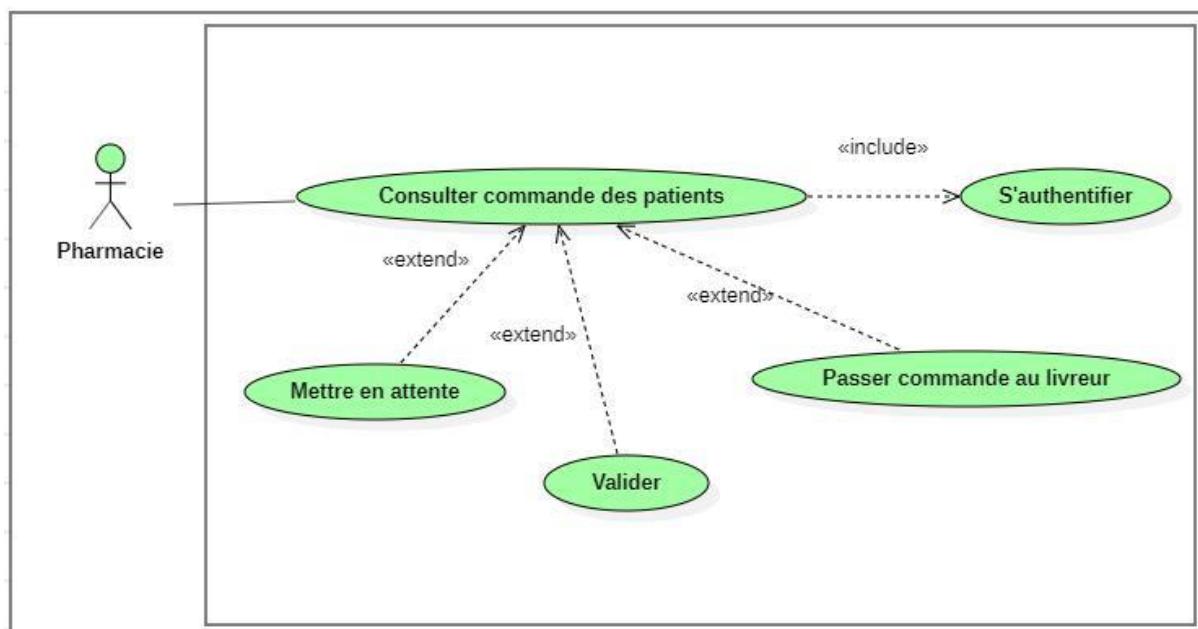


Figure 21: Diagramme du cas d'utilisation « Gérer les commandes des patients »

### 3 Les besoins non fonctionnels

Cette partie décrit les exigences que notre plateforme doit satisfaire de manière informelle. Ce sont les besoins qui permettent d'améliorer la qualité des services de l'application comme la convivialité et l'ergonomie des interfaces, l'amélioration de temps de réponse, la pertinence de résultat. Notre solution doit répondre aux critères suivants :

- **Besoin de performance** : Le système doit répondre à toutes les exigences des usagers d'une manière optimale.
- **Besoin de sécurité** : Le système doit être sécurisé et garantir la protection et la confidentialité des données.
- **La convivialité** : La plateforme doit fournir une interface conviviale, simple et ergonomique. Elle doit présenter un flux logique entre les pages.
- **Besoin de rapidité** : Le système doit avoir une durée d'exécution des traitements le plus proche possible du temps réel.
- **La disponibilité** : C'est l'aptitude d'un dispositif à être en état de fonctionnement et de disponibilité dans des conditions données.

### 4 Diagramme de classe d'analyse

La modélisation conceptuelle des données permet de dégager l'ensemble des données manipulées en vue d'élaborer le diagramme de classes. En effet, ce dernier donne une vue statique du système. Il décrit les types et les objets du système. Il s'agit donc d'une représentation des données du champ de l'étude ainsi que le lien sémantique entre ces données, facilement compréhensible, permettant de décrire le système d'information à l'aide des concepts proposés par le modèle UML [22]. La figure 22 représente le diagramme de classe d'analyse qui contient toutes les informations telles que les classes, les méthodes, les attributs et les associations.

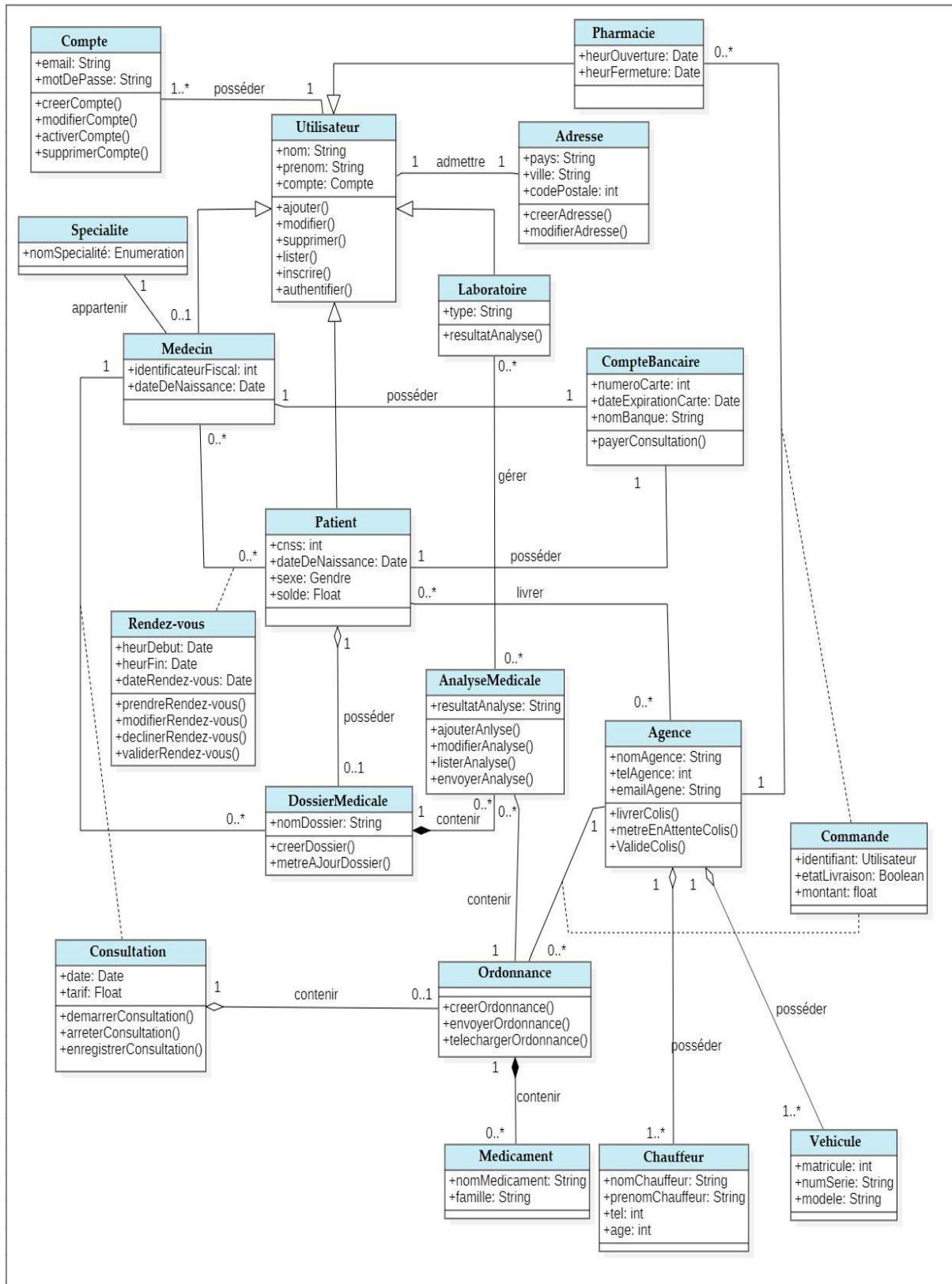


Figure 22: Diagramme de classe d'analyse

## 5 Diagramme De séquence d'analyse

L'objectif du diagramme de séquence système est de montrer les interactions des acteurs externes avec le système. Cette interaction ne détaille pas les objets qui participent aux processus ; en effet le système est présenté comme une boîte noire. Elle présente une vision globale de tout ce que le système peut faire.

Les concepteurs utilisent les diagrammes de séquence d'analyse pour définir les interactions entre les différents composants logiciels (Serveur, base des données, services web, etc..), et pour définir les bases du dialogue Homme/Machine.

### 5.1 Diagramme de séquence d'analyse « prendre rendez-vous »

Pour prendre un rendez-vous, le patient doit accéder au profil du médecin puis sélectionner la date et l'heure du rendez-vous souhaité, et l'enregistrer. Une fois cette étape passée, un formulaire de paiement est affiché pour continuer l'opération de la prise de rendez-vous. A la fin la demande de rendez-vous est enregistrée dans la base de données. La figure 23 présente le diagramme de séquence d'analyse pour le cas d'utilisation « Prendre rendez-vous ».

### 5.2 Diagramme de séquence d'analyse « payer Consultation »

Après la sélection du patient d'un rendez-vous pour une consultation médicale en ligne, le patient est invité pour payer sa consultation médicale afin de rester parmi les bénéficiaires des services fournis par la plateforme. La figure 24 présente le diagramme de séquence d'analyse du cas d'utilisation « Payer consultation ».

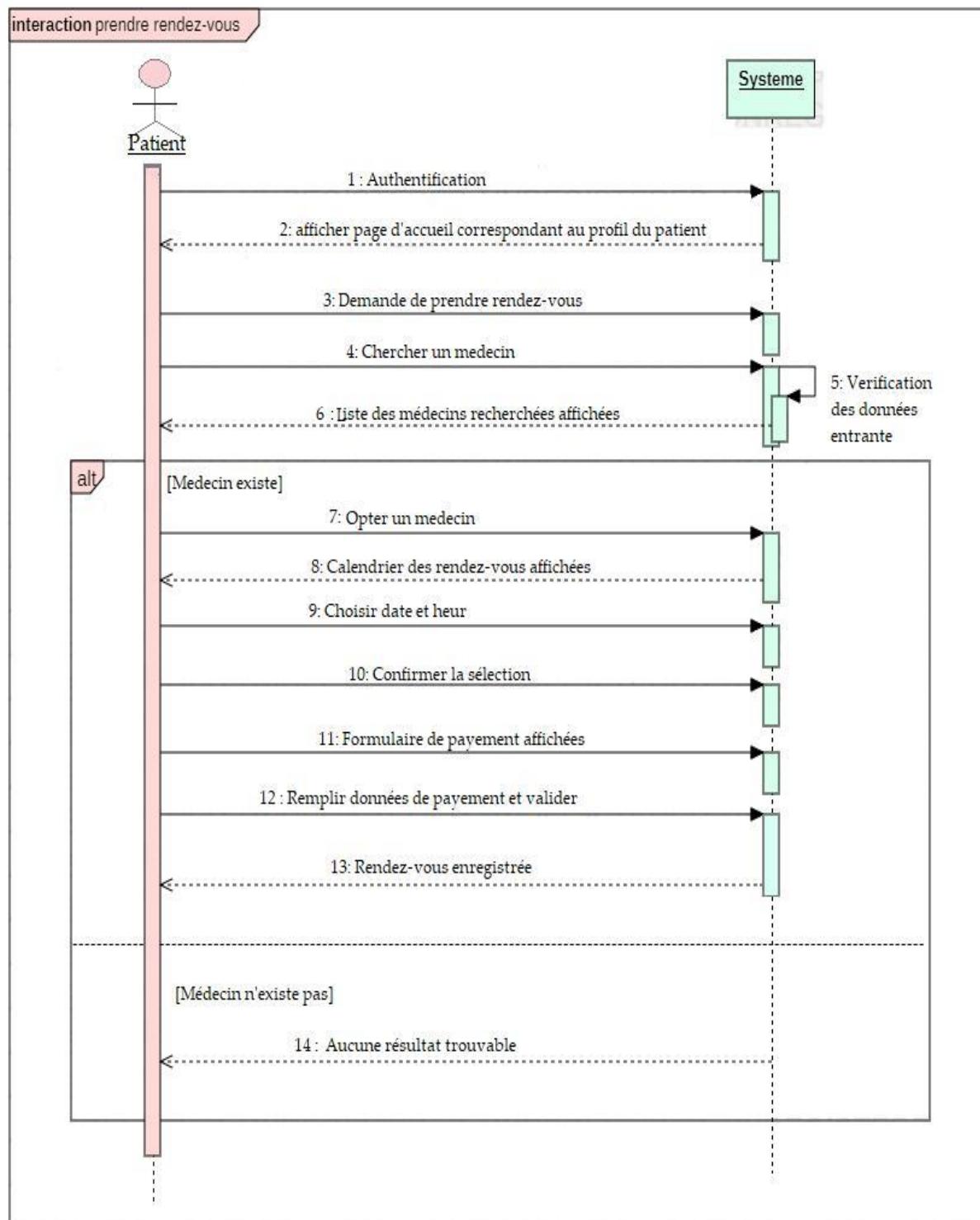


Figure 23: Diagramme de séquence d'analyse « Prendre rendez-vous »

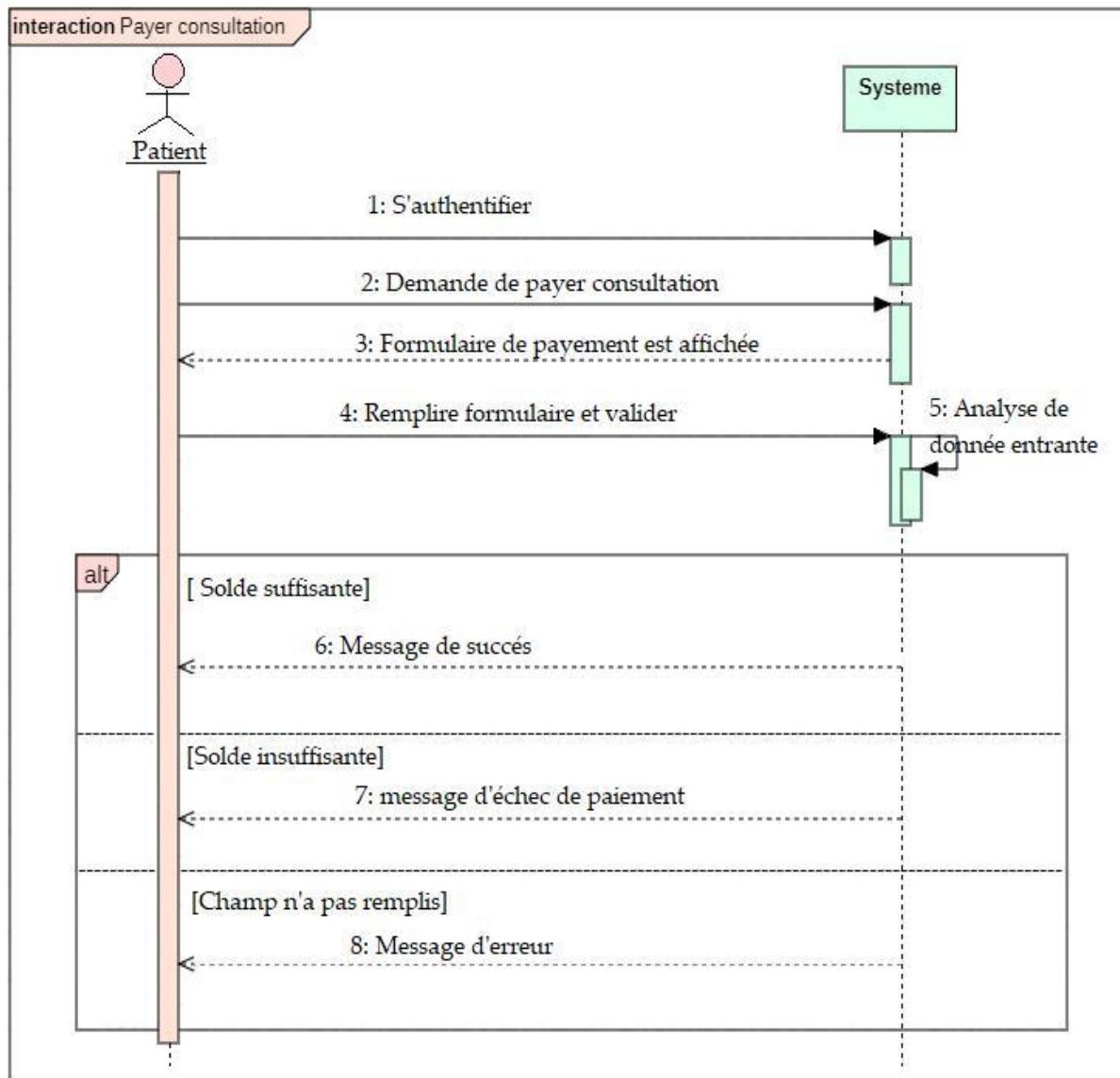


Figure 24: Diagramme de séquence d'analyse « Payer consultation »

## Conclusion

Ce chapitre nous a permis de faire un découpage fonctionnel de notre système par le biais du diagramme de cas d'utilisation, et d'anticiper sur les interfaces qui seront développées ultérieurement.

L'activité d'analyse a permis de fournir une spécification complète des besoins issus du diagramme de cas d'utilisation (s'authentifier, gérer utilisateur, consulter les produits, s'authentifier, s'enregistrer, suivre des commandes...) et de les structurer afin de faciliter leur compréhension.

# CHAPITRE 3 : CONCEPTION DES BESOINS

## Introduction

Dans le cadre de ce chapitre, nous nous intéresserons à la conception architecturale et à la conception de quelques cas d'utilisations via le diagramme de classe système et le diagramme de classe MVC, ensuite, nous présenterons quelques diagrammes d'activité, d'état transition et finirons par les diagrammes de séquences.

## 1 Conception architecturale

Les options d'application architecturales sont décisives dès qu'ils participent à l'efficacité et au temps de développement.

### 1.1 Patron de conception « Model-View-Controller »

L'architecture MVC est un patron utilisé pour le développement des applications web. L'objectif du ce patron est de séparer les données, le traitement et la présentation des données. La règle est « *diviser pour mieux régner* ». Chaque couche est spécialisée et dédiée à des tâches bien déterminées. La figure 25 montre les trois composants principaux et leurs relations entre eux.

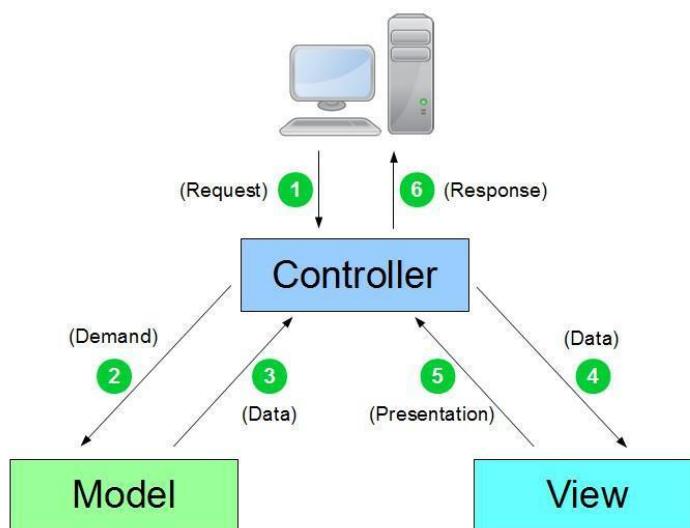


Figure 25: Le modèle MVC [23]

Ces 3 couches travaillent ensemble suivant un scénario précis.

- ➡ L'utilisateur fait une demande au contrôleur, donc il envoie une requête HTTP au contrôleur, et il reçoit comme réponse une View.

- Le contrôleur gère les interactions. D'une manière générale, il va utiliser les données du modèle, les traiter en fonction de l'action de l'utilisateur, et les envoyer à la vue (View) afin de les afficher.
- Le modèle contient toute la logique de l'application : la récupération des données, la modification des données, le traitement entre les données, les calculs, etc...
- La vue (View) a pour seul but d'afficher des informations de manière formatée et présentable pour l'utilisateur. Il n'y a aucune logique dans cette partie.

## 1.2 Déploiement physique de l'architecture logicielle

Dans le contexte du langage de modélisation unifié (UML), un diagramme de déploiement fait partie de la catégorie des diagrammes structurels, car il décrit un aspect du système même. Dans le cas présent, le diagramme de déploiement décrit le déploiement physique des informations générées par le logiciel sur des composants matériels. On appelle artefact l'information qui est générée par le logiciel [24]. La figure 26 représente le diagramme de déploiement relatif au déploiement du diagramme de composants de la solution à implémenter.

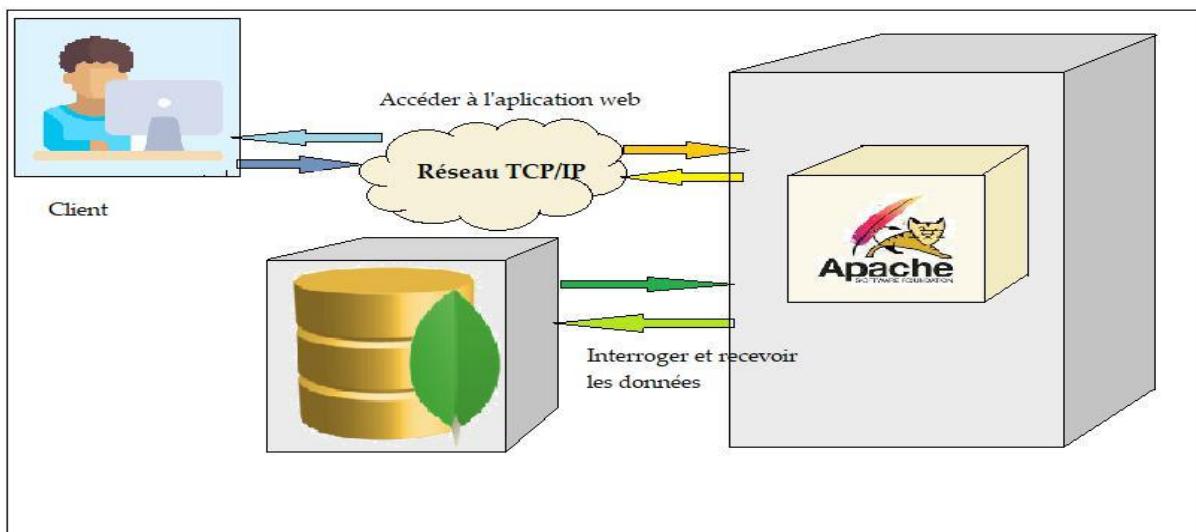


Figure 26: Déploiement physique de l'architecture logicielle

## 2 Conception détaillée

### 2.1 Présentation du langage de modélisation UML

Le langage de modélisation unifié (Unified Modeling Language ou UML) est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets

logiciels, il peut être appliqué à toutes sortes de systèmes, ne se limitant pas au domaine informatique. UML est une référence de modèles permettant de représenter un système informatique et son utilisation prévue dans l'entreprise. L'usage de ces modèles par les informaticiens vise à améliorer la qualité des applications informatiques. L'utilisation des modèles UML sert à :

- La compréhension globale de système d'informations, c'est-à-dire parvenir à une meilleure maîtrise des coûts et des délais dans la conduite des projets d'administration.
- La différenciation entre les diagrammes d'UML.
- L'adaptation à différents systèmes informatiques.

Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions. L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions. Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel [25].



*Figure 27: Unified Modeling Language UML [26]*

UML s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes [27].

- **Les diagrammes structurels:**

- **Le diagramme de classes :** montrant les briques de base statiques : les classes, les associations, les interfaces, les attributs, les opérations et les généralisations...
- **Le diagramme d'objets :** montrant les instances des éléments structurels et leurs liens à l'exécution.
- **Le diagramme de packages :** montrant l'organisation logique du modèle et les relations entre packages.
- **Le diagramme de structure composite :** montrant l'organisation interne d'un élément statique complexe.

- **Le diagramme de composants** : montrant des structures complexes, avec leurs interfaces fournies et requises.

- **Le diagramme de déploiement** : Il montre le déploiement physique sur les ressources matérielles.

#### Les diagrammes comportementaux :

- **Le diagramme de cas d'utilisation** : montrant les interactions fonctionnelles entre les acteurs et le système à l'étude.
- **Le diagramme de vue d'ensemble des interactions** : fusionnant les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.
- **Le diagramme de séquence** : montrant la séquence verticale des messages passés entre objets au sein d'une interaction.
- **Le diagramme de communication** : montrant la communication entre objets dans le plan au sein d'une interaction.
- **Le diagramme de temps** : fusionnant les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.
- **Le diagramme d'activité** : montrant l'enchaînement des actions et décisions au sein d'une activité.
- **Le diagramme d'états** : montrant les différents états et transitions possibles des objets d'une classe.

Nous choisirons les diagrammes les plus adaptés à notre projet sur la base du modèle de « Kruchten ».

Le modèle (4+1) vues, dit de « Kruchten », d'un système informatique permet d'organiser la description du système en plusieurs vues complémentaires, chacune présentant le système selon un point de vue différent. L'utilisation de vues permet de traiter séparément les intérêts des divers groupes d'intervenants (architectes, utilisateurs, développeurs, chefs de projets, etc.) et ainsi de mieux séparer les préoccupations fonctionnelles (le domaine d'application ou métier ciblé par le système, par exemple la banque ou le contrôle aérien) et les préoccupations extra fonctionnelles (les propriétés techniques telles que la sûreté de fonctionnement).

La figure schématisse le modèle (4+1) vues. Ce modèle est composé de cinq vues :

- La vue « **logique** » décrit les aspects statiques et dynamiques d'un système en termes de classes, d'objets, de connexions et de communications. Elle se concentre sur l'abstraction et l'encapsulation.

- La vue « **processus** » capte les aspects de concurrence et de synchronisation, et les décompose en flux d'exécution (processus, fil d'exécution, etc.). Elle se rapporte aux objets actifs et aux interactions.
- La vue « **développement** » représente l'organisation statique des modules (exécutable, codes source, paquetages, etc.) dans l'environnement de développement.
- La vue « **physique** » décrit les différentes ressources matérielles et l'implantation logicielle tenant compte de ces ressources. Donc, elle se rapporte aux nœuds physiques d'exécution et au placement des objets sur les nœuds.
- La dernière vue appelée « **cas d'utilisation** », se concentre sur la cohérence en présentant des scénarios d'utilisation qui mettent en œuvre les éléments des quatre premières vues. Les scénarios sont une abstraction des exigences fonctionnelles de l'application. Cette dernière vue valide en quelque sorte les autres vues et assure la cohérence globale. C'est aussi cette dernière vue qui est construite en premier, juste après l'établissement du cahier des charges, pour fixer les contours du système à réaliser avec ses fonctionnalités appelées, dans la terminologie UML, des cas d'utilisation [28].

### Les 4+1 Vues

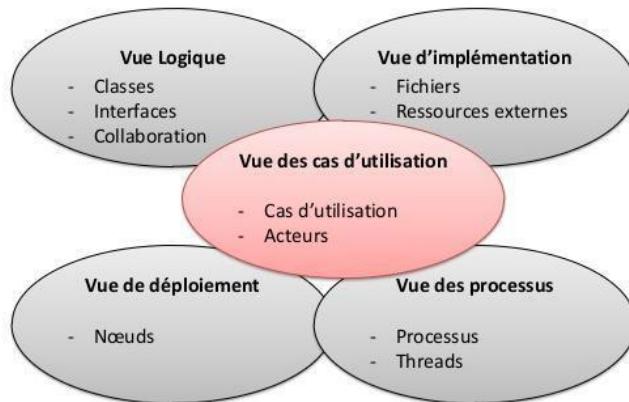


Figure 28: Diagramme de Kruchten [29]

## 2.2 Le diagramme de classe de conception

Le diagramme de classe de conception représente l'architecture conceptuelle du système: décrit les classes utilisées par le système, ainsi que ses connexions qui représentent une maison conceptuelle (héritage) ou une relation organique (agrégation). Bien que le diagramme opérationnel indique un système du point de vue des acteurs, l'image de classe affiche la

structure interne. Il offre une présentation abstraite d'objets pour le système qui interagissent ensemble pour effectuer des cas d'utilisation.

Dans notre cas, pour le diagramme de classe de conception on a choisi le patron de conception MVC pour le développement de notre application web. On a opté pour le framework Spring boot afin d'implémenter les microservices de notre solution proposée, or que la couche controller du patron MVC a besoin de la couche service destinée à l'implémentation des traitements métiers spécifiques à l'application. Cette même couche utilise la couche repository qui est destinée à l'interaction avec les sources des données externes. La figure 29 représente le diagramme de classe MVC contenant les classes de chaque couche du motif de conception : Model, View, Controller. (Les 2 couches Service et Repository ne sont pas illustrées sur notre diagramme).

## CHAPITRE 3 : CONCEPTION DES BESOINS

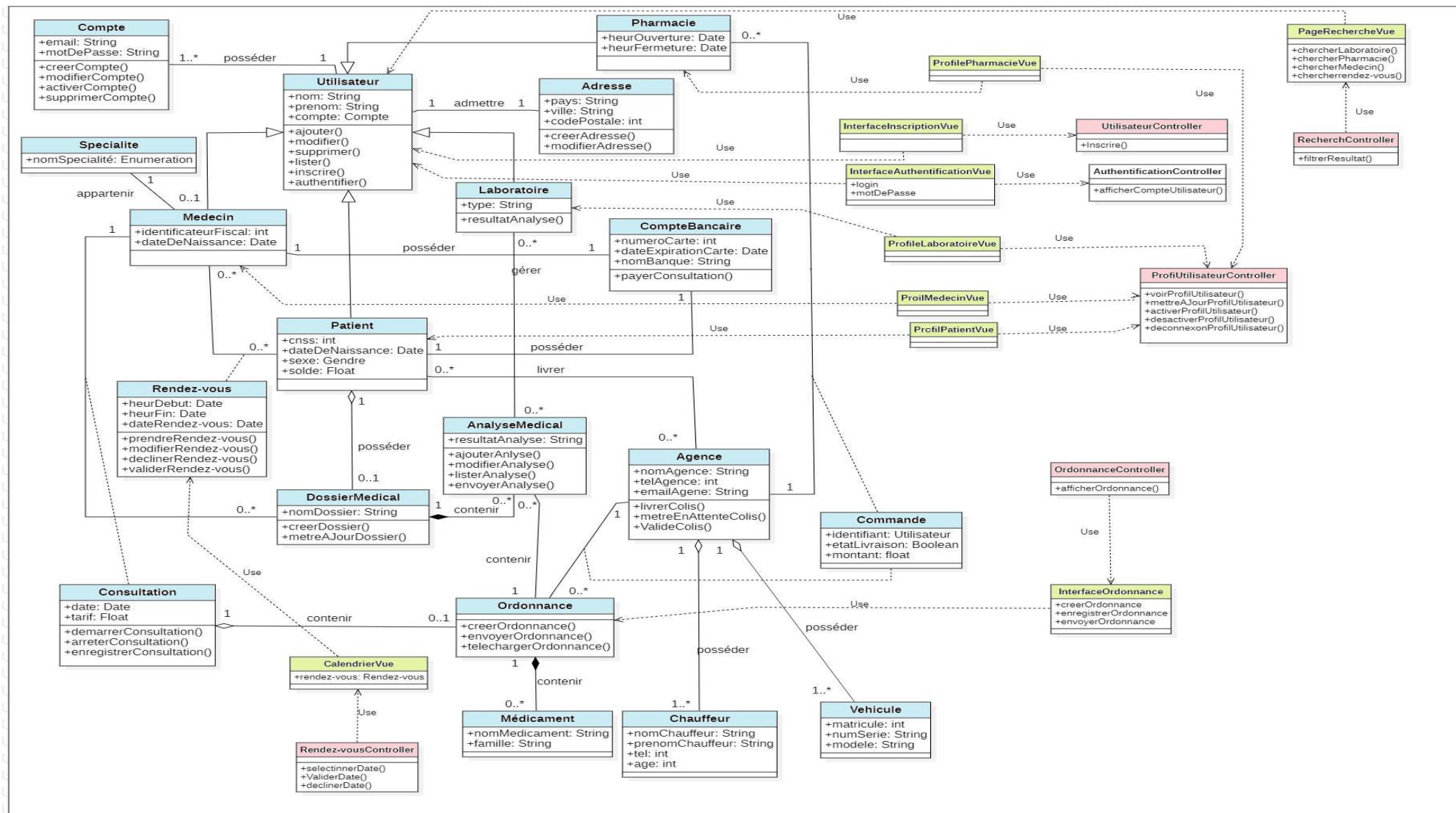


Figure 29: Diagramme de classe MVC

## 2.3 Diagramme d'activité

Dans UML, un diagramme d'activité est utilisé pour afficher la séquence des activités. Les diagrammes d'activité représentent le flux de travail à partir d'un point de départ au point d'arrivée, détaillant les nombreux sentiers de décision, qui existent dans la progression des événements contenus dans l'activité. Ils peuvent être utilisés à des situations de détail, où le traitement parallèle peut survenir dans l'exécution de certaines activités. Les diagrammes d'activités sont utiles pour la modélisation d'entreprise où ils sont utilisés pour détailler les processus impliqués dans des activités commerciales [30].

### 2.3.1 Diagramme d'activité « s'authentifier »

La figure 30 présente le diagramme d'activité pour modéliser le processus d'authentification. Le processus concerne les acteurs suivants :

- ➊ **L'utilisateur** : qui saisit son login et son mot de passe, ou de créer un compte s'il n'est pas déjà titulaire d'un compte.
- ➋ **Le système** : Vérifie les coordonnées entrées par l'utilisateur et affiche la page adéquate selon le type de l'utilisateur.

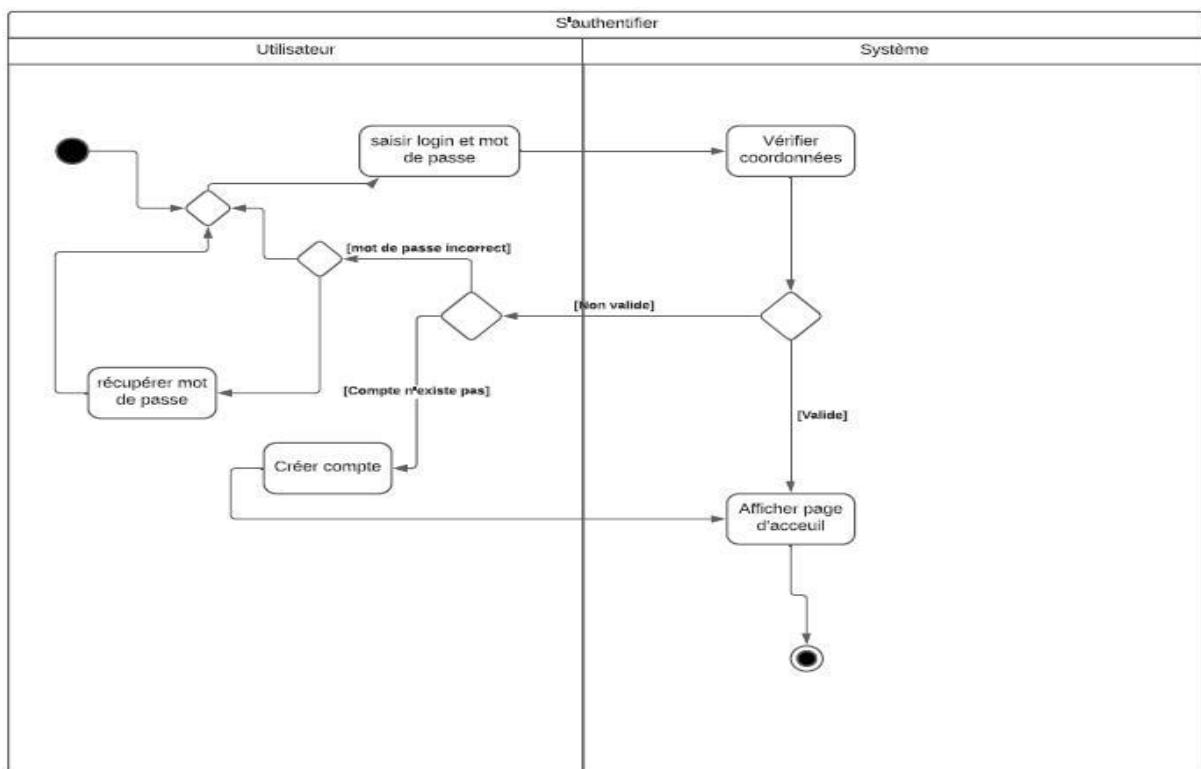


Figure 30: Diagramme d'activité « S'authentifier »

### 2.3.2 Diagramme d'activité « envoyer résultat de l'analyse médicale »

La figure 31 présente le diagramme d'activité pour modéliser le processus « envoyer le résultat de l'analyse médicale ». Le processus concerne les acteurs suivants :

- **Agent de laboratoire** : qui saisit les coordonnées du médecin
- **Système** : cherche le médecin, envoie le résultat de l'analyse et notifie l'agent de la réussite de l'envoi
- **Médecin** : notifié de la réussite de l'envoi

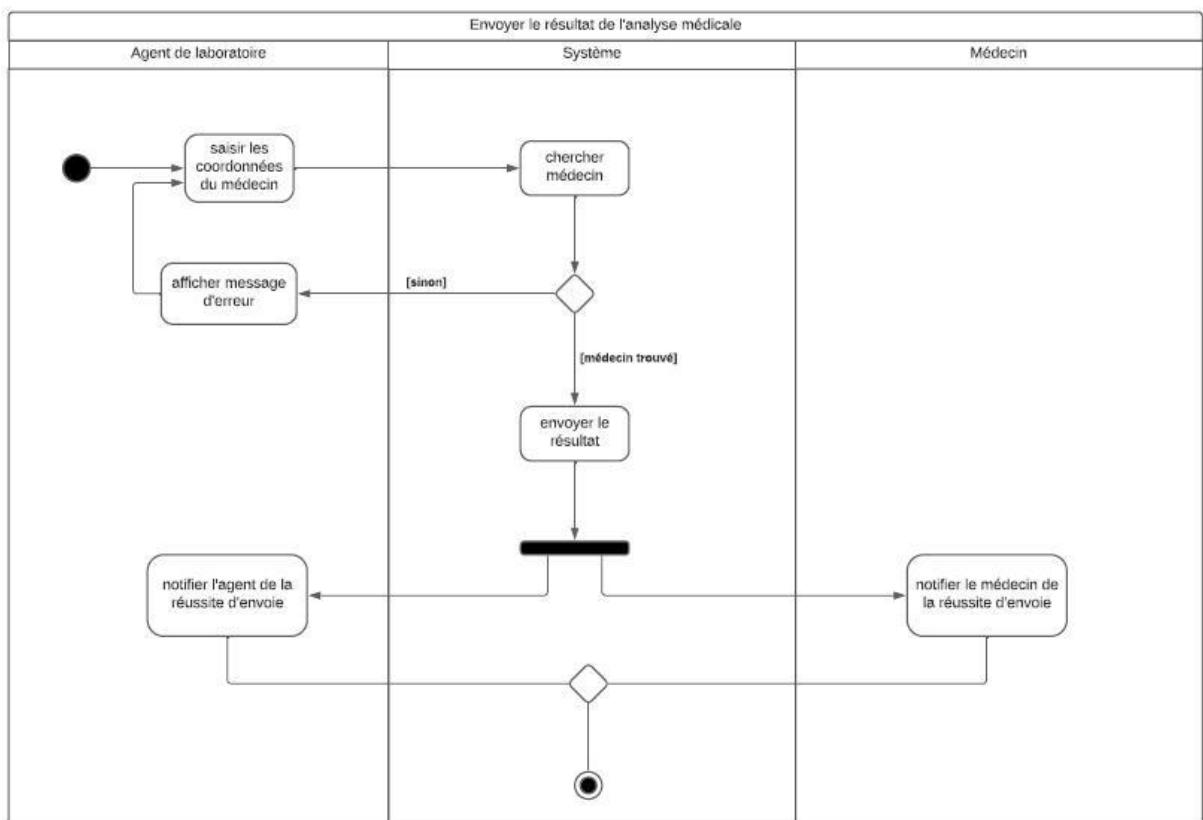


Figure 31: Diagramme d'activité « envoyer résultat de l'analyse médicale »

## 2.4 Diagramme états-transitions

Les diagrammes états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets (de type signaux, invocations de méthode).

Ils spécifient habituellement le comportement d'une instance de classeur (classe ou composant), mais parfois aussi le comportement interne d'autres éléments tels que les cas d'utilisation, les sous-systèmes, les méthodes [31].

#### 2.4.1 Diagramme d'état-transition « Consultation »

La figure 32 présente le diagramme d'états-transition de l'objet consultation.

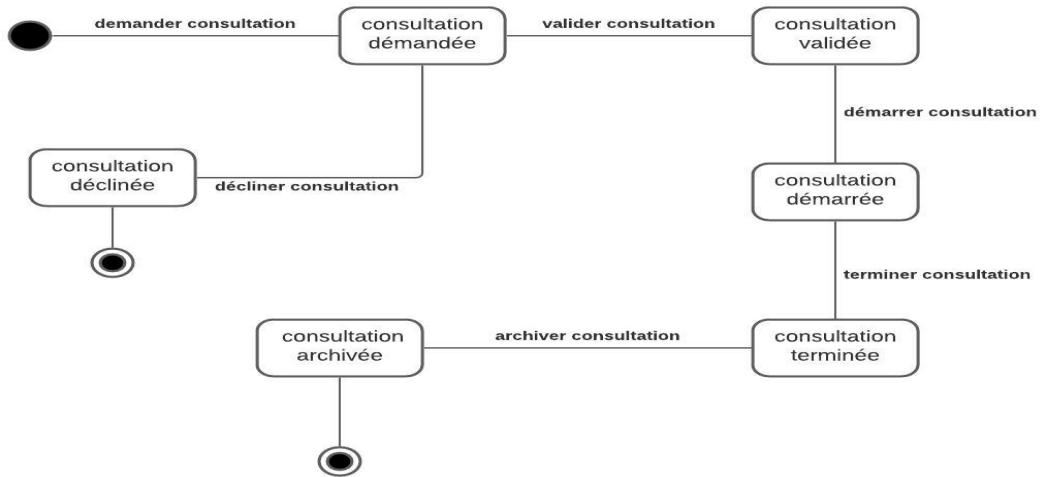


Figure 32: Diagramme états-transitions « Consultation »

#### 2.4.2 Diagramme d'état-transition « Rendez-vous »

La figure 33 présente le diagramme d'état transition de l'objet rendez-vous.

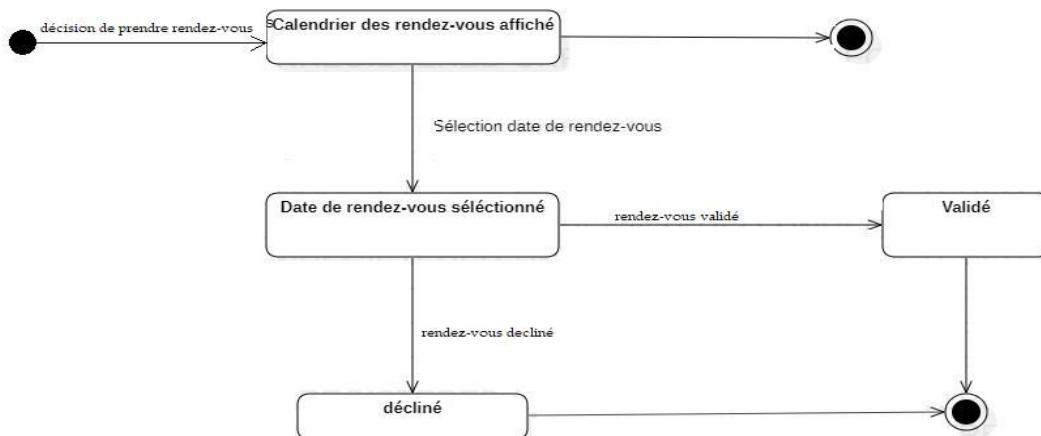


Figure 33: Diagramme états-transitions « Rendez-vous »

#### 2.5 Diagramme de séquence objet

Un diagramme de séquence objet est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquence sont organisés en fonction du temps. Le temps s'écoule au fur et à

mesure que vous parcouriez la page. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence des messages [32].

### 2.5.1 Diagramme de séquence de cas d'utilisation « S'authentifier »

#### a. Description textuelle du cas d'utilisation « S'authentifier »

Les tableaux 13 et 14 présentent une description textuelle du cas d'utilisation S'authentifier.

*Tableau 13: Description du cas d'utilisation « S'authentifier ».*

<b>Titre :</b>	<b>S'authentifier</b>			
<b>But :</b>	Permet à l'utilisateur de s'authentifier pour accéder à son compte.			
<b>Résumé :</b>	L'utilisateur saisit son login et mot de passe dans un formulaire (ces identifiants doivent être compatibles avec les données existantes dans la base de données). Si le compte existe, l'utilisateur accède à son profil. Sinon, un message d'erreur est affiché.			
<b>Acteurs :</b>	Tous les acteurs de la plateforme :  Médecin,  Patient,  Laboratoire,  Pharmacie.			
<b>Description des enchainements :</b>				
<b>Pré condition :</b>	<b>Post condition :</b>			
L'utilisateur possède un compte	L'utilisateur accède à son espace privé.			
<b>Scénario nominal :</b>				
1. L'utilisateur demande l'accès au système. 2. Le système affiche la page d'authentification. 3. L'utilisateur saisit son login et mot de passe. 4. Le système vérifie les informations saisies par l'utilisateur. 5. Le système vérifie l'existence de l'utilisateur. 6. Si l'utilisateur est identifié, le système lui permet d'accéder à son espace privé : (Médecin, Patient, Pharmacie, Laboratoire).				
<b>Scénario d'exception :</b>				

Tableau 14: Suite description du cas d'utilisation « S'authentifier ».

<b>E1.</b> L'utilisateur n'a pas entré le bon identifiant.
<b>1.1</b> Le système affiche un message d'erreur.
<b>1.2</b> Reprise de l'étape 2 du scénario nominal.
<b>E2.</b> L'utilisateur n'existe pas dans la base de données.
<b>2.1</b> Le système lui affiche un message d'erreur et lui indique qu'il n'a pas du compte donc il faut inscrire.

#### a. Diagramme de séquence du cas d'utilisation « S'authentifier »

La figure 34 présente le diagramme de séquence du cas d'utilisation S'authentifier.

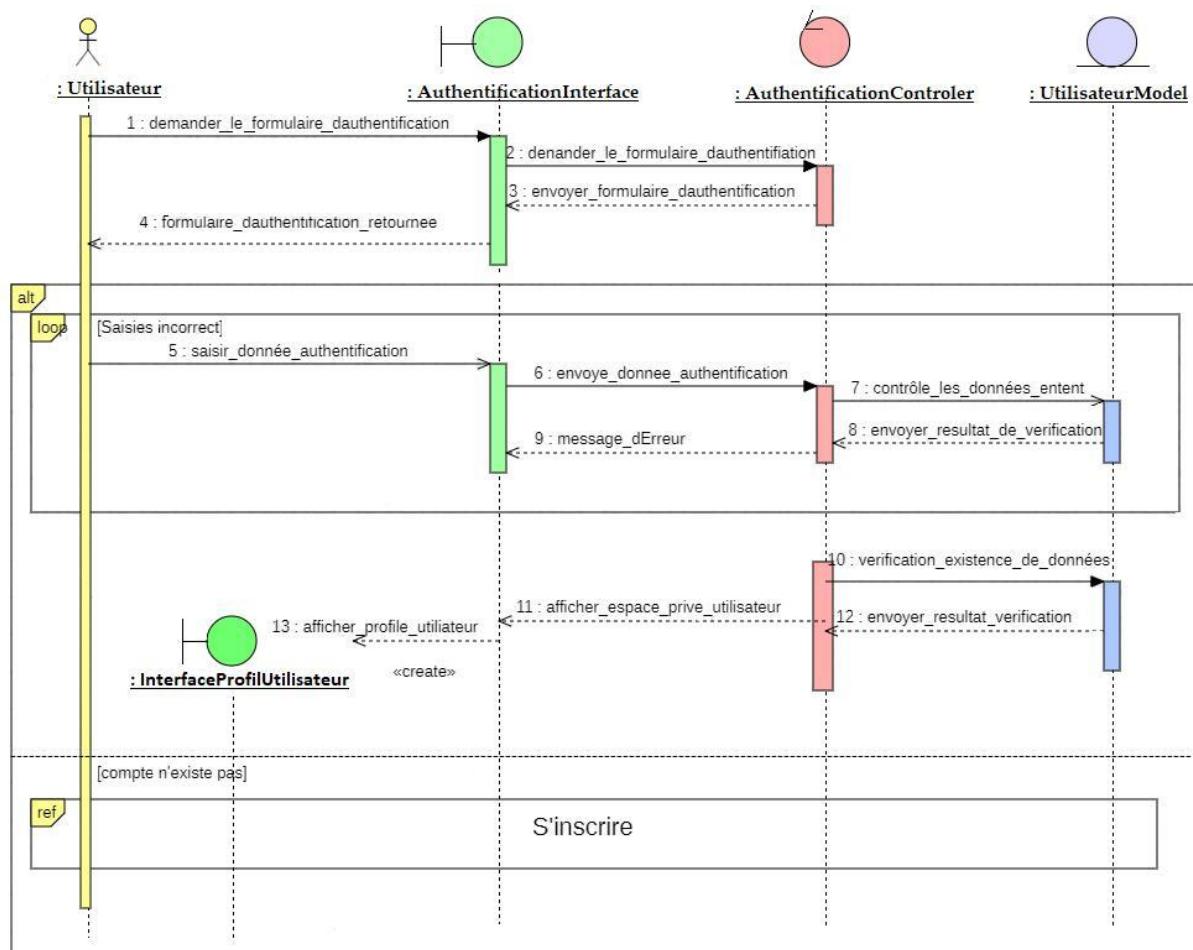


Figure 34: Diagramme de séquence du cas d'utilisation « S'authentifier »

#### 2.5.2 Diagramme de séquence de cas d'utilisation « Prendre rendez-vous »

##### a. Diagramme de séquence « Prendre rendez vous »

*a.1 Description textuelle « prendre rendez-vous »*

Les deux tableaux 15 et 16 présentent une description textuelle du cas d'utilisation prendre rendez-vous.

*Tableau 15: Description du cas d'utilisation «Prendre rendez-vous».*

<b>Titre :</b>	<b>Prendre rendez-vous</b>
<b>But :</b>	Permet au patient de prendre un rendez-vous pour consulter un médecin en présentiel, en ligne, ou à domicile.
<b>Résumé :</b>	Pour prendre rendez-vous, le patient doit consulter le profil du médecin, puis sélectionner la date et l'heure du rendez-vous et l'enregistrer. Après avoir terminé cette étape, le formulaire de paiement s'affichera pour poursuivre l'opération de rendez-vous. Enfin, la demande de réservation est enregistrée dans la base de données.
<b>Acteurs :</b>	Seulement pour le : Patient
<b>Description des enchainements :</b>	
<b>Pré condition :</b>	<b>Post condition :</b>
Le patient est authentifié.	Le patient prend une date de rendez-vous pour une consultation médicale en ligne.
<b>Scénario nominal :</b>	
<ol style="list-style-type: none"> <li>1. Le patient sélectionne dans le menu le lien : « Prendre rendez-vous ».</li> <li>2. Le système renvoie une page de recherche pour aider le patient de trouver son médecin.</li> <li>3. Le patient saisit le nom du médecin recherché, sa spécialité et le type de rendez-vous (en cabinet, en ligne ou à domicile), si le nom du médecin est inconnu pour le patient, la saisie de la spécialité du médecin est suffisante pour effectuer la recherche.</li> <li>4. Le système affiche une page contient le résultat de la recherche.</li> <li>5. Le patient sélectionne le médecin recherché.</li> <li>6. Le système affiche le calendrier des rendez-vous du médecin.</li> <li>7. Le patient sélectionne une date et heure pour sa consultation médicale et valide.</li> <li>8. Le système affiche un formulaire de paiement pour continuer l'opération de la prise de rendez-vous.</li> <li>9. Le patient remplir le formulaire de paiement et enregistre.</li> <li>10. Le système enregistre les informations entrant par le patient dans la base de données.</li> </ol>	

Tableau 16: Suite tableau 15, description du cas d'utilisation «Prendre rendez-vous».

<b>Scénario d'exception :</b> <b>E1.</b> Le nom du médecin est inconnue, n'existe pas dans la base de données. <b>1.1</b> Aucun résultat trouvable. <b>1.2</b> Reprise de l'étape 3.
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### a.2 Diagramme de séquence « Prendre rendez-vous »

La figure 35 présente le diagramme de séquence du cas d'utilisation prendre rendez-vous pour le patient.

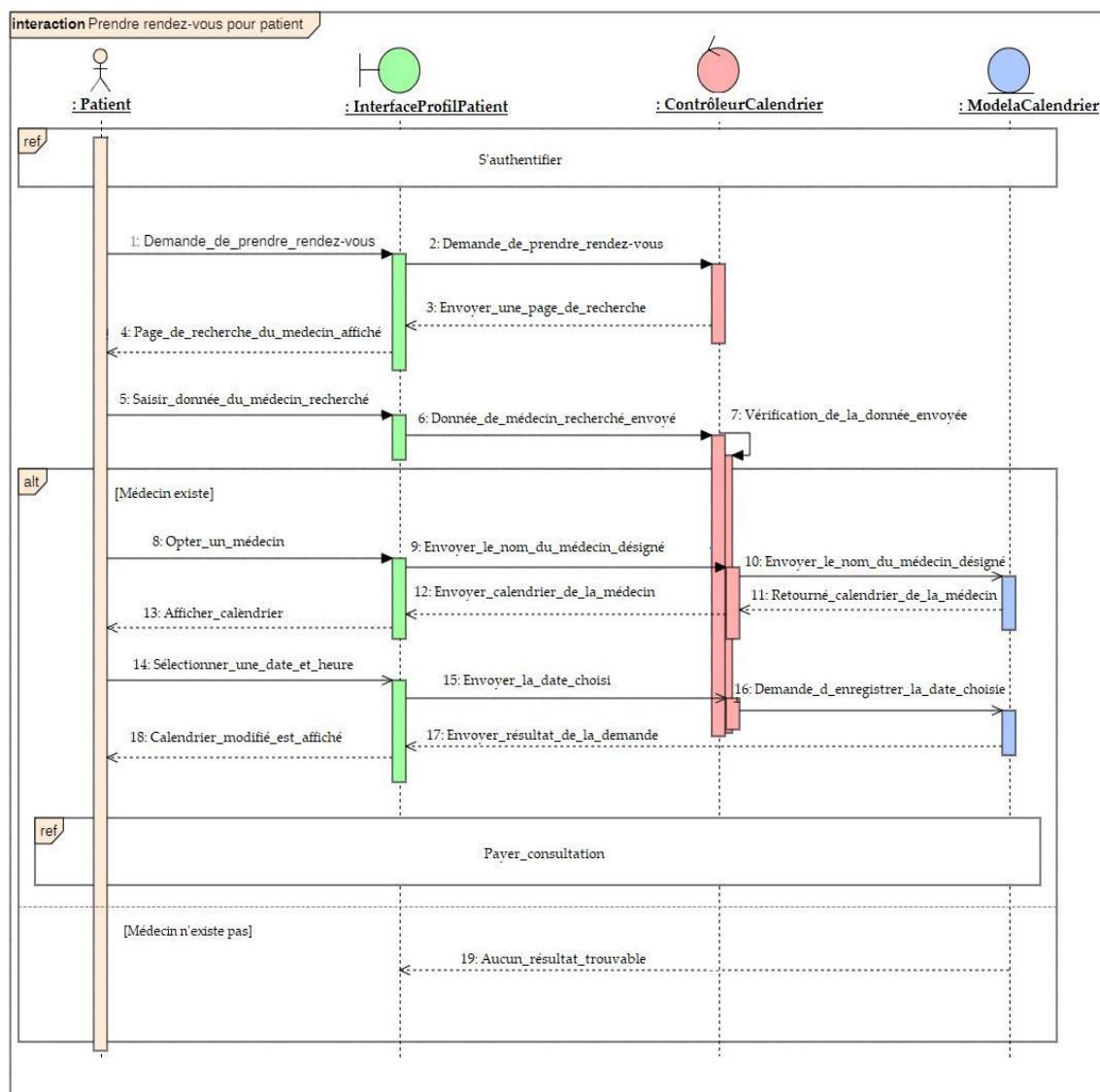


Figure 35: Diagramme de séquence « Prendre rendez-vous »



**b. Diagramme de séquence « Proposer un rendez-vous en présentiel »**

**b.1 Description textuelle du cas d'utilisation « Proposer un rendez-vous en présentiel »**

Le tableau 17 présente une description textuelle du cas d'utilisation proposer un rendez-vous en présentiel.

*Tableau 17: Description du cas d'utilisation « Proposer un rendez-vous en présentiel ».*

<b>Titre :</b>	
<b>But :</b>	
<b>Résumé :</b>	
<b>Acteurs :</b>	
<b>Description des</b>	<b>enchainements :</b>
<b>Pré condition :</b>	<b>Post condition :</b>
Le médecin est authentifié.	Un rendez-vous en présentiel est enregistré dans la base de données.
Une consultation est en cours.	
<b>Scénario nominal :</b>	
1. Le médecin sélectionne dans le menu le lien: «Mes rendez-vous».	
2. Le système affiche la page du calendrier.	
3. Le médecin sélectionne une date et heure pour un rendez-vous en présentiel et enregistre.	
4. Le système enregistre l'ajout du rendez-vous en présentiel dans la base de données.	
<b>Scénario d'exception :</b>	
Pas d'exception.	

### b.2 Diagramme de séquence « Proposer rendez-vous en présentiel »

La figure 36 présente le diagramme de séquence du cas d'utilisation proposer rendez-vous en présentiel.

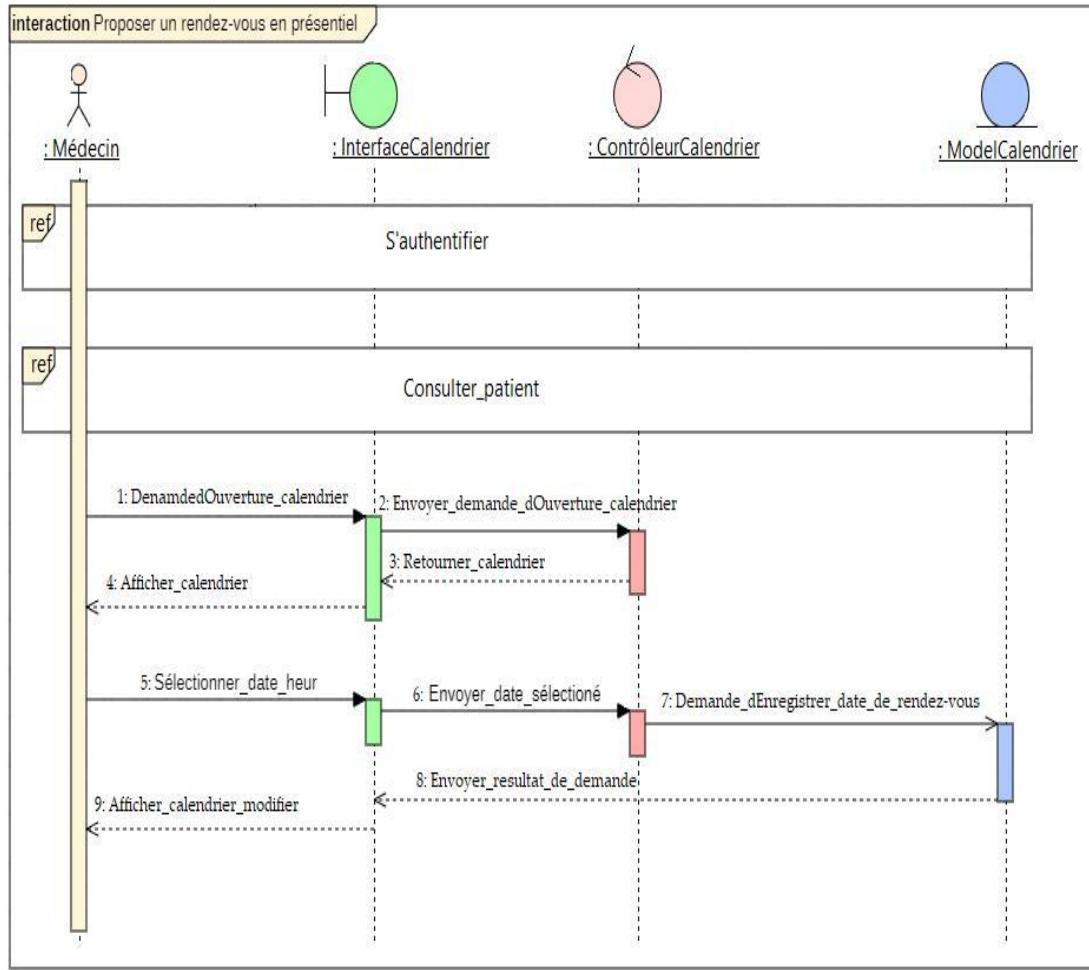


Figure 36: Diagramme de séquence « Proposer rendez-vous en présentiel »

## Conclusion

Comme nous l'avons vu, les activités de conception facilitent la compréhension de notre système à travers les diagrammes d'activités, les diagrammes de séquences, le diagramme de classe de conception, les diagrammes d'activités et diagrammes états-transitions. Le chapitre suivant consiste en une phase d'implémentation de la solution.

# CHAPITRE 4 : RÉALISATION

## Introduction

Après avoir effectué l'étude et la conception de notre plateforme web, nous passons à la phase d'implémentation. Ce chapitre présente le résultat du travail effectué durant ce projet de fin d'études. Nous allons présenter, aussi, l'environnement matériel et les outils de développement utilisés. Nous clôturons ce chapitre par quelques captures d'écrans démontrant les fonctionnalités de notre plateforme web.

## 1 Environnement de développement

### 1.1 Architecture physique

Notre plateforme Web nécessite un serveur Web et une base de données ; une architecture à 3 niveaux est donc requise. L'architecture 3-tier (de l'anglais tier signifiant étage ou niveau) est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et présenter cette application comme un empilement de trois couches, étages, niveaux dont le rôle est clairement défini :

- **La présentation des données** : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur.
- **Le traitement métier des données** : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.
- **Et enfin l'accès aux données persistantes (persistancy en anglais)** : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

### 1.2 Environnement matériel

Pour la réalisation du projet, nous avons utilisé deux pc portables pour le développement avec les caractéristiques :

#### ■ ASUS i3 5ème génération

- **Processeur** : Intel Core i3-5005U (2.0GHz, 3 Mo de mémoire cache, Dual Coeurs),
- **Disque Dur** : 500 Go HDD,
- **Carte graphique**: Intel HD Graphics 4400.

 **LENOVO E15**

- **Processeur :** Intel® Core™ i5-10210U CPU @ 1.60Hz 2.10Ghz,
- **Disque Dur :** 1 To,
- **Carte graphique :** AMD Radeon TM RX 640.

## 1.3 Environnement logiciel

### 1.3.1 Outils logiciels

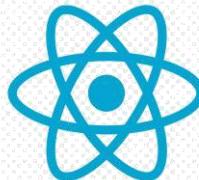
*Tableau 18: différents outils logiciels utilisés dans ce projet*

	<p>IntelliJ IDEA est un environnement de développement intégré de technologie Java destiné au développement de logiciels informatiques. On l'utilise pour le développement du back-end.</p>
	<p>Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la complétion intelligente du code, la refactorisation du code et Git intégré... Supporte les langages : C, C++, Go, CSS, JavaScript, TypeScript, C# et PHP. On l'utilise pour le développement du front-end.</p>
	<p>Postman permet de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en Collections.</p>
	<p>StarUML est un logiciel de modélisation open source. Ce logiciel constitue une excellente option pour une familiarisation à la modélisation.</p>
	<p>Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur ; il est principalement utilisé pour le traitement des photographies numériques. On a utilisé Photoshop pour créer le logo de notre plateforme.</p>

## 1.4 Technologies de développement

Pour le développement de notre plateforme, on a utilisé le framework React.JS pour le développement du front-end, le framework Spring Boot pour le développement du back-end et un système de gestion de base de données mongoDB.

*Tableau 19: Différentes technologies utilisées dans notre projet*

	En informatique, un Spring se définit comme étant un framework libre très riche, parmi les plus réputés au monde. Il permet de construire l'infrastructure d'une application Java et d'en faciliter le développement. Il faut savoir qu'à ses débuts, l'application Java était un langage de programmation, qui s'est peu à peu étendu à une variété de plateformes, allant de l'ordinateur au smartphone. Spring est considéré comme un projet OpenSource, créé et développé par Rod Johnson en juin 2003, qui est par ailleurs le cofondateur de SpringSource.
	React est une bibliothèque JavaScript pour la construction d'interfaces utilisateur (UI).
	MongoDB est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++. Il fait partie de la mouvance NoSQL. Ce système qu'on a choisi, utilise des documents de type JSON pour stocker les données.

## 2 Les composants applicatifs réalisés

### 2.1 Interface Accueil

La page d'accueil est la page la plus importante d'une application web, site internet... Elle présente l'affichage de notre site Web. C'est généralement l'une des pages les plus visitées, et les visiteurs sont généralement autorisés à s'y rediriger. La figure 37 présente l'interface d'accueil de notre plateforme web « doctena.tn ».

## CHAPITRE 4 : RÉALISATION

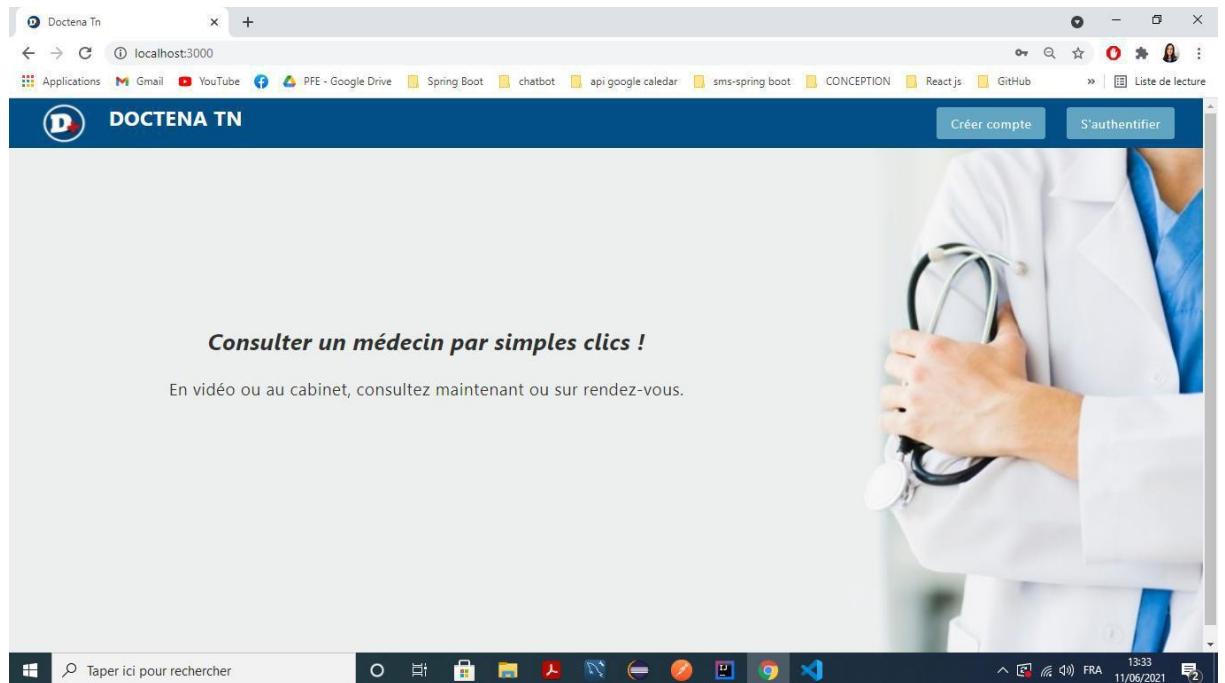


Figure 37: Interface Accueil

## 2.2 Interface Inscription

L'interface de l'inscription est le point d'entrée de chaque application web. Chaque utilisateur doit créer un compte pour pouvoir profiter des services de l'application. Les figures 38, 39, 40 et 41 présentent les étapes de création d'un compte pour un utilisateur de type médecin.

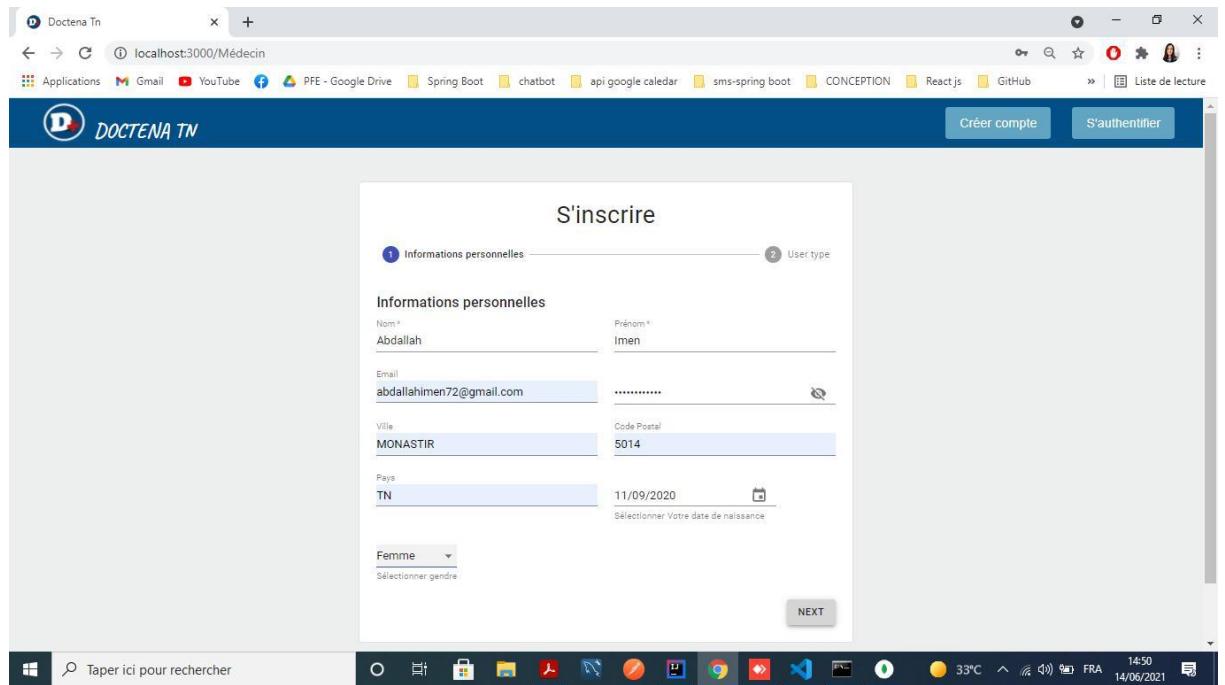


Figure 38: Interface première étape de création d'un compte

## CHAPITRE 4 : RÉALISATION

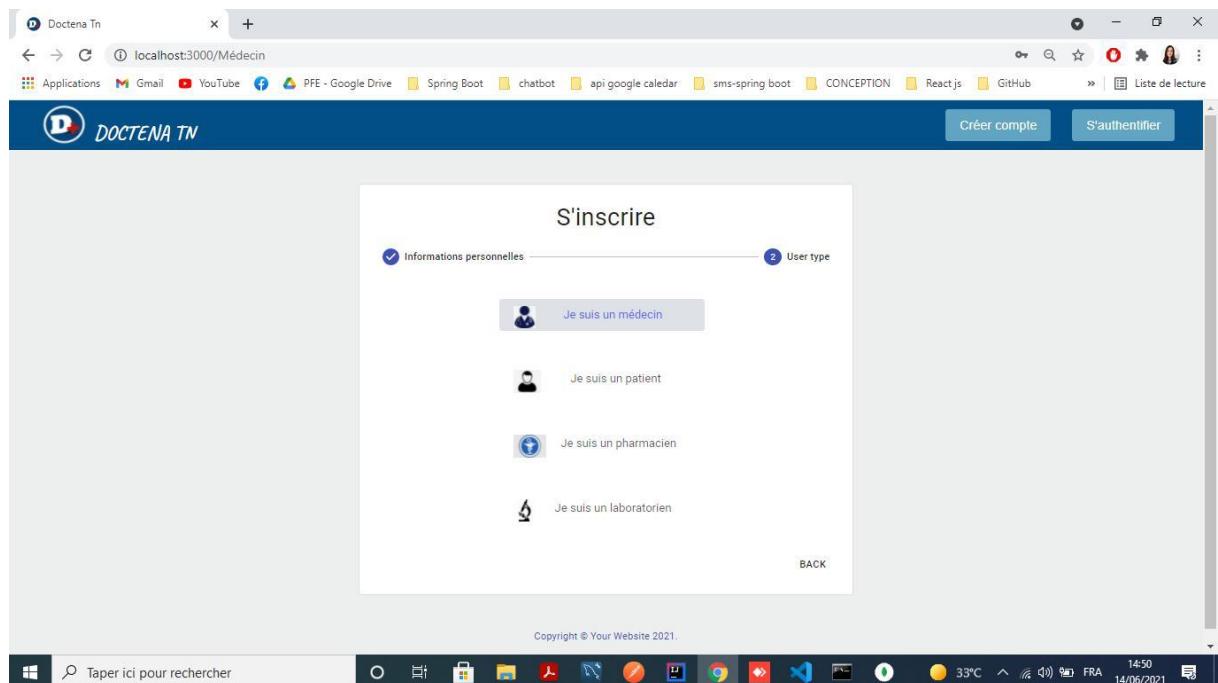


Figure 39: Interface deuxième étape de création d'un compte

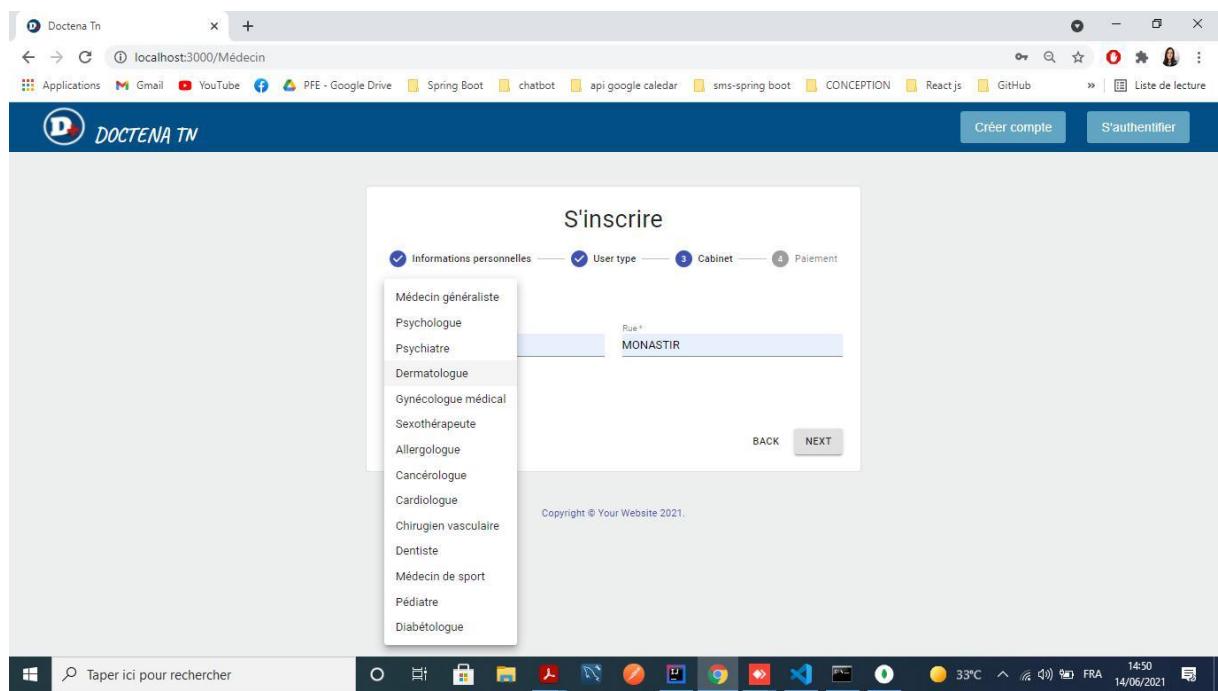


Figure 40: Interface troisième étape de création d'un compte médecin

## CHAPITRE 4 : RÉALISATION

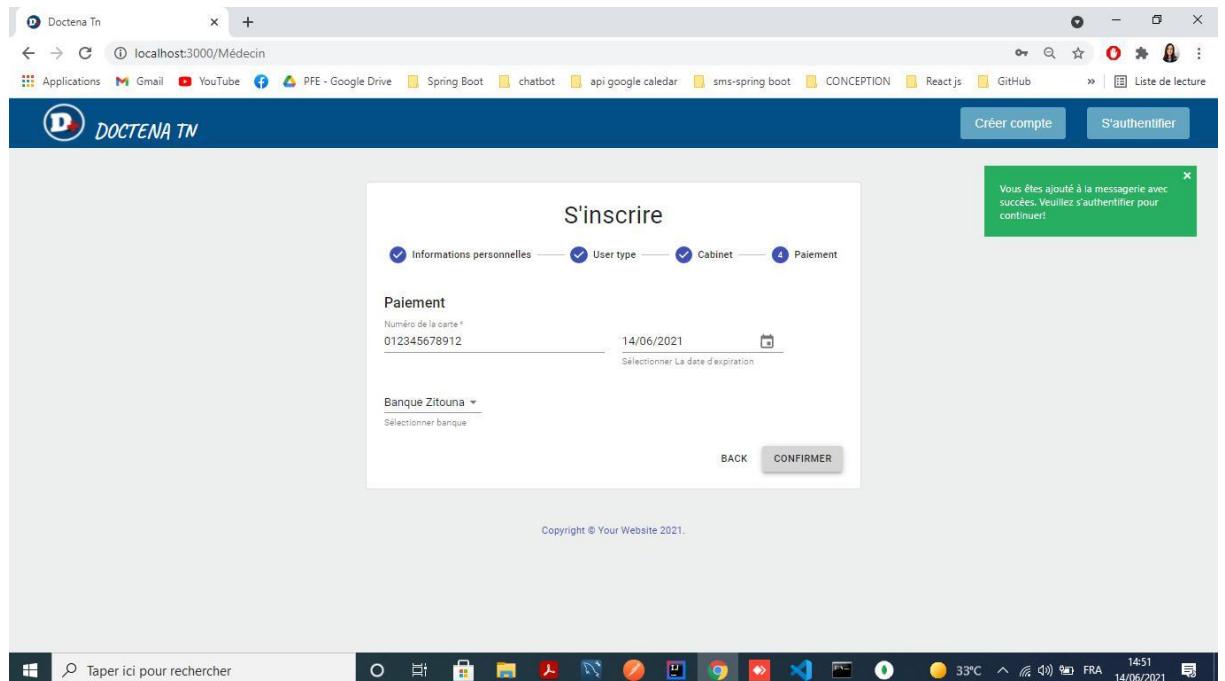


Figure 41: Interface quatrième étape de création d'un compte médecin

Lorsque le médecin termine les étapes de création de son compte, un e-mail de vérification est envoyé à son adresse mail. Et pour terminer la création de son compte, il est invité à cliquer sur le lien en bleu « Activer maintenant », une notification de confirmation est affichée. La figure 42 présente le format du mail envoyé.

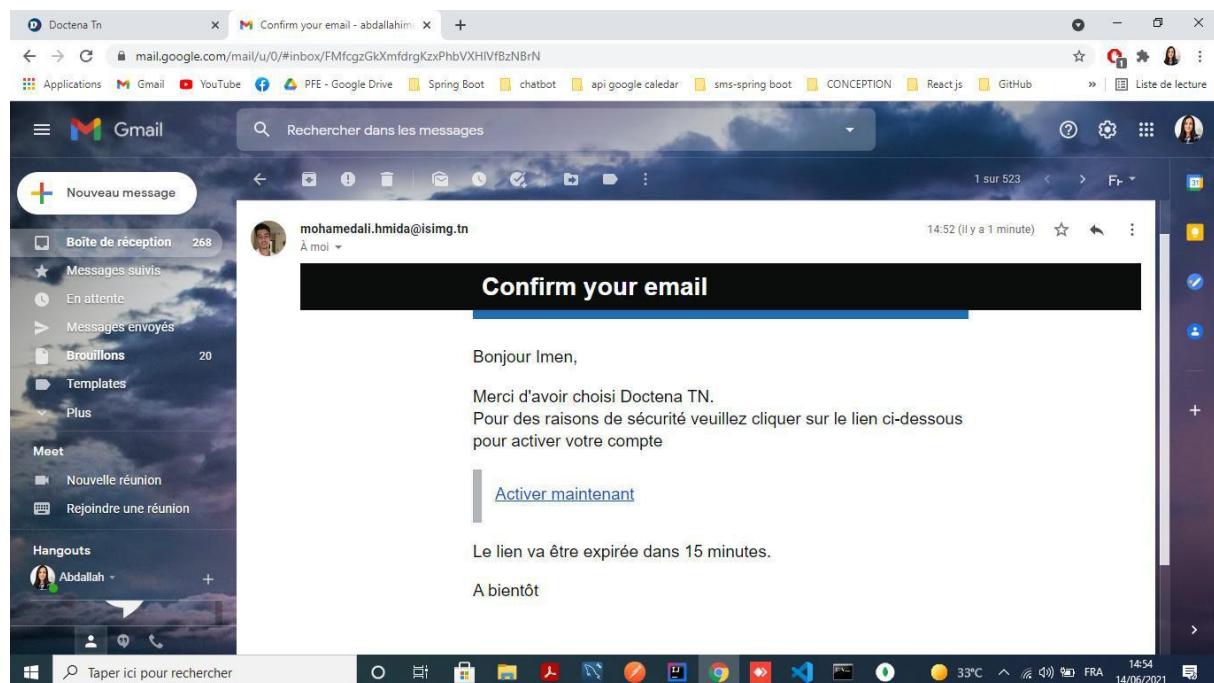


Figure 42: E-mail de vérification

## 2.3 Interface Authentification

L'authentification est un processus de vérification de l'identité d'une personne. L'objectif de l'authentification est de vérifier que la personne n'est pas un imposteur. Pour s'authentifier, l'utilisateur est invité à introduire son email et son mot de passe. La figure 43 présente l'interface de l'authentification de notre plateforme.

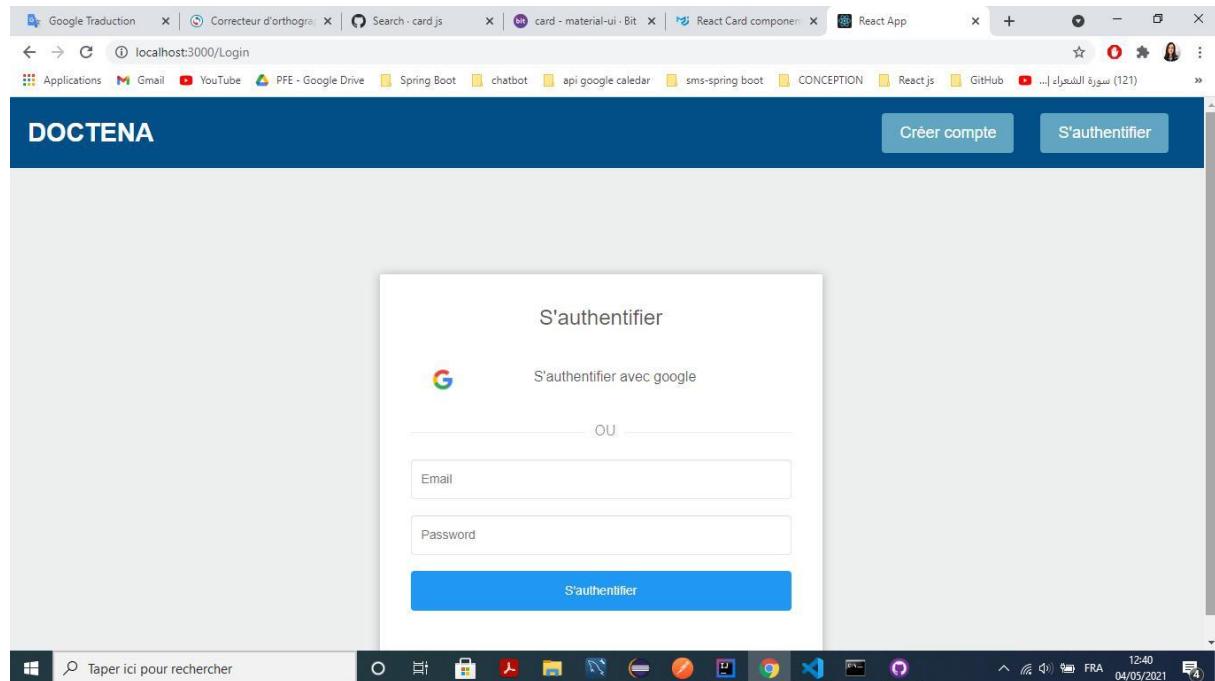


Figure 43: Interface Authentification

## 2.4 Interface de recherche d'un médecin

Après l'authentification, le patient accède à la page de recherche d'un médecin ; il introduit le nom et le prénom d'un médecin recherché, sa spécialité puis il clique sur le bouton de recherche. La figure 44 présente l'interface de recherche sur un médecin et le résultat de recherche.

## CHAPITRE 4 : RÉALISATION

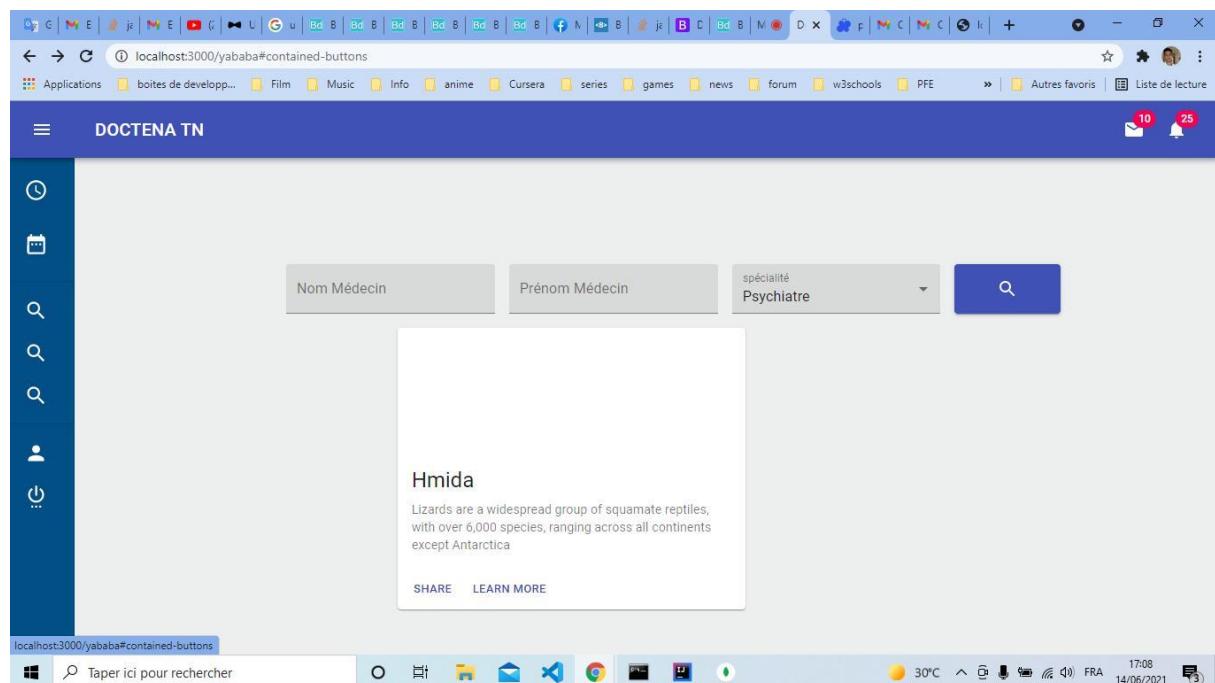


Figure 44: Page de recherche sur un médecin

### 2.5 Interface Calendrier des rendez-vous du médecin

Le patient accède au calendrier du médecin recherché, puis il sélectionne une date de rendez-vous, et valide. La figure 45 présente l'interface de calendrier des rendez-vous.

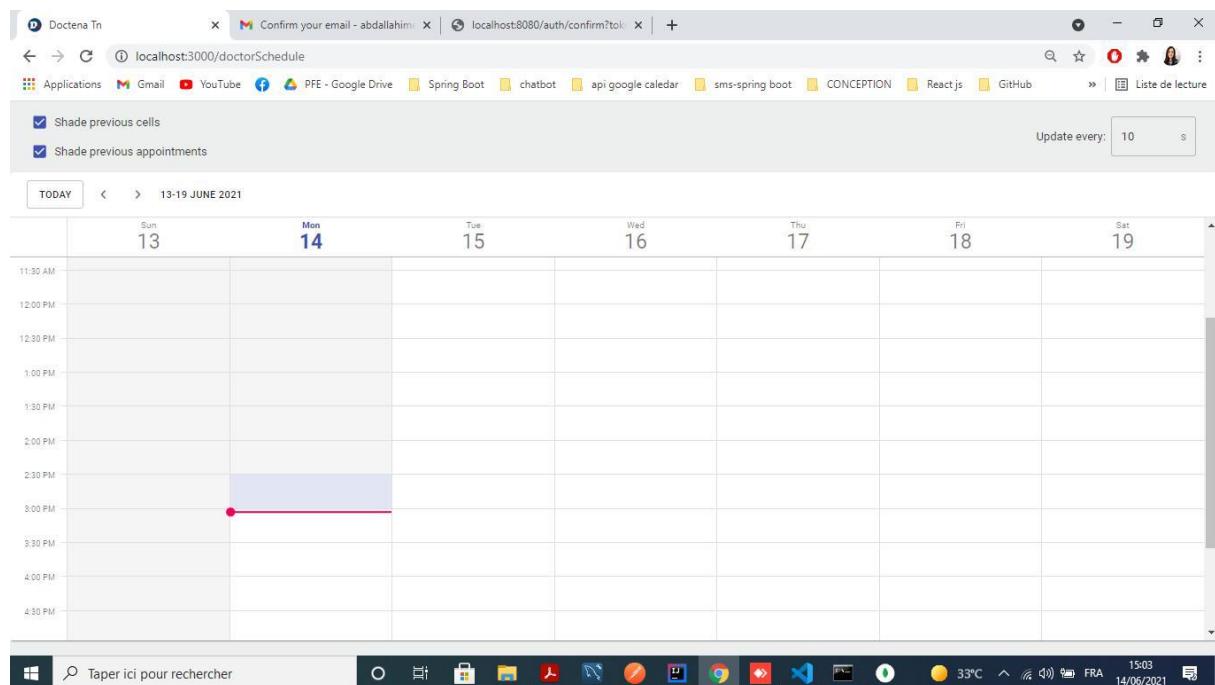


Figure 45: Interface Calendrier des rendez-vous du médecin

## 2.6 Interface Profil médecin

Le médecin accède à son profil afin de bénéficier des fonctionnalités offertes par la plateforme. Les figures 46 et 47 présentent l'interface de profile médecin.

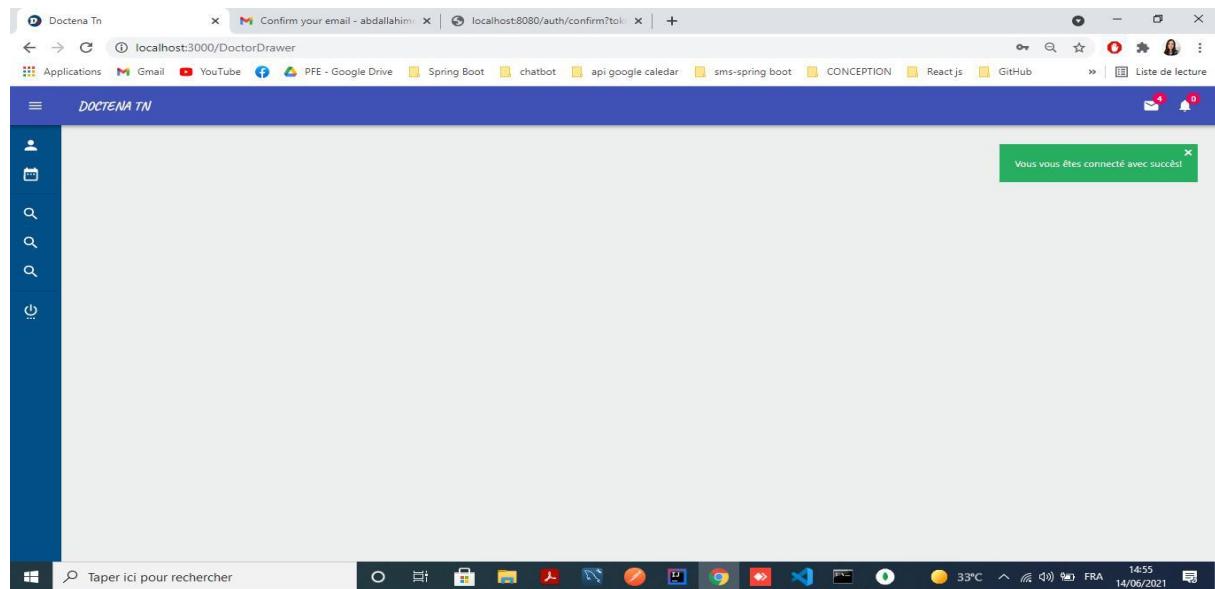


Figure 46: Interface Profil médecin

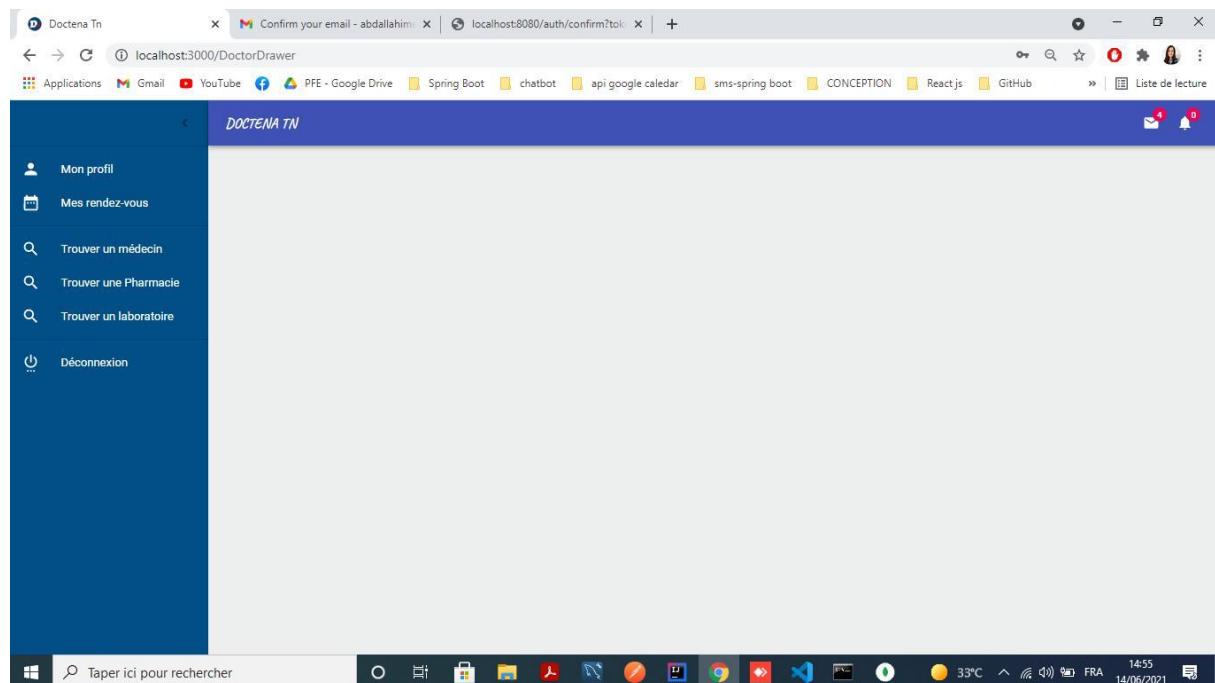


Figure 47: Suite interface Profil médecin

## 2.7 Notification d'un rendez-vous

Suite à la prise d'un rendez-vous par le patient, une notification est envoyée au médecin pour l'informer qu'un rendez-vous est ajouté à son calendrier (figure 48).

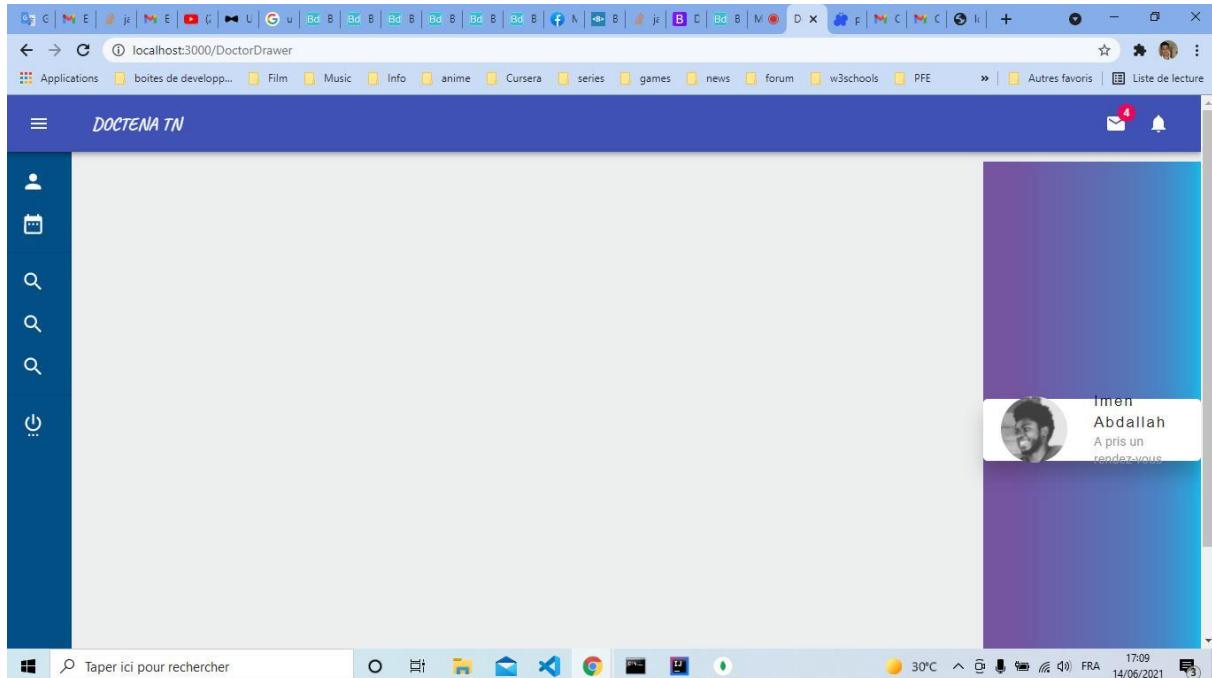


Figure 48: Notification d'un rendez-vous

## 2.8 Interface Calendrier du médecin

La figure 49 présente l'interface du calendrier d'un médecin qui contient toutes les dates des rendez-vous réservés par les patients.

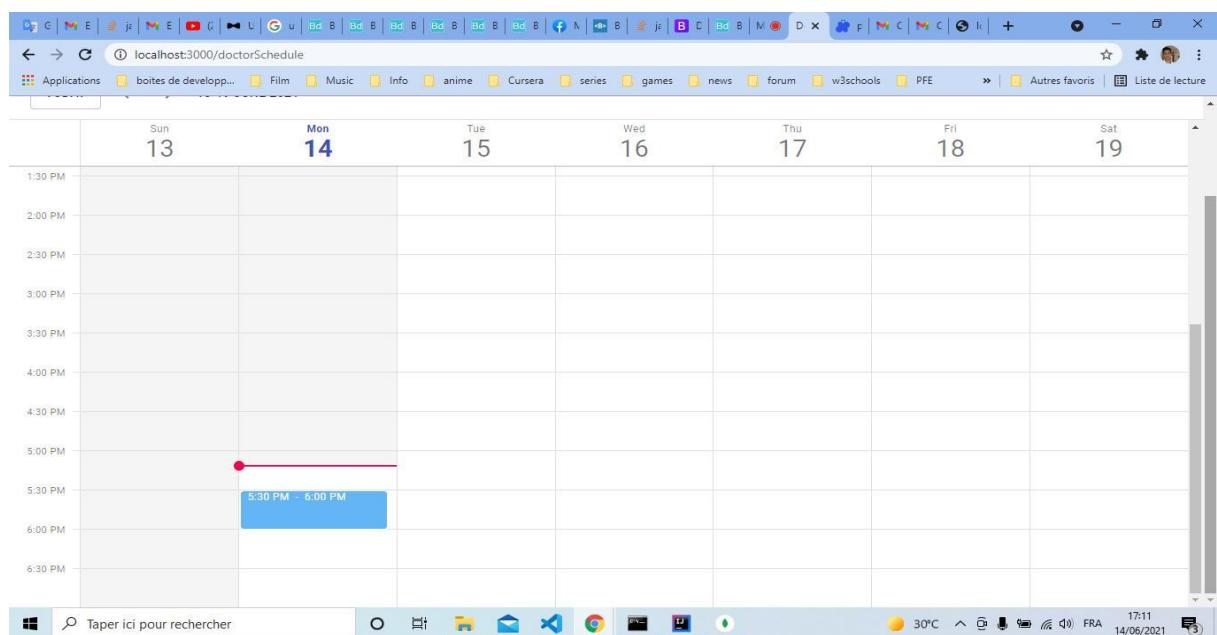


Figure 49: Interface Calendrier médecin

## CHAPITRE 4 : RÉALISATION

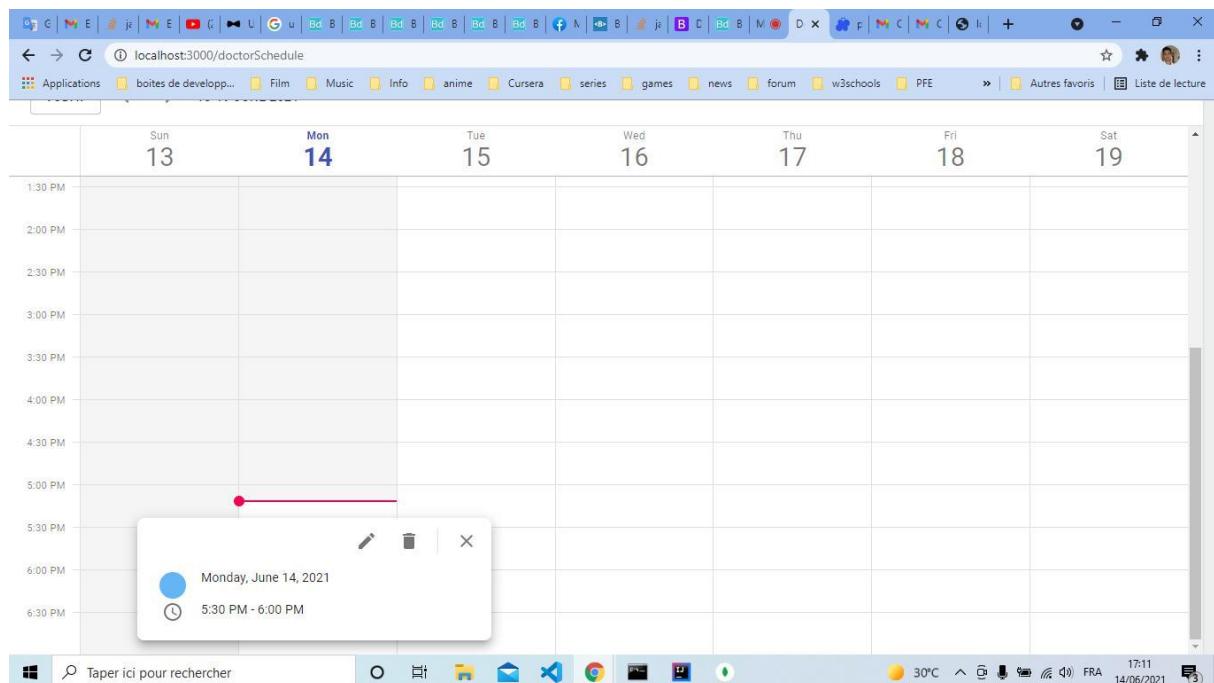


Figure 50: Suite interface calendrier médecin

## 2.9 Interface Messagerie

La figure 51 présente une simple discussion par messages entre un médecin et un patient.

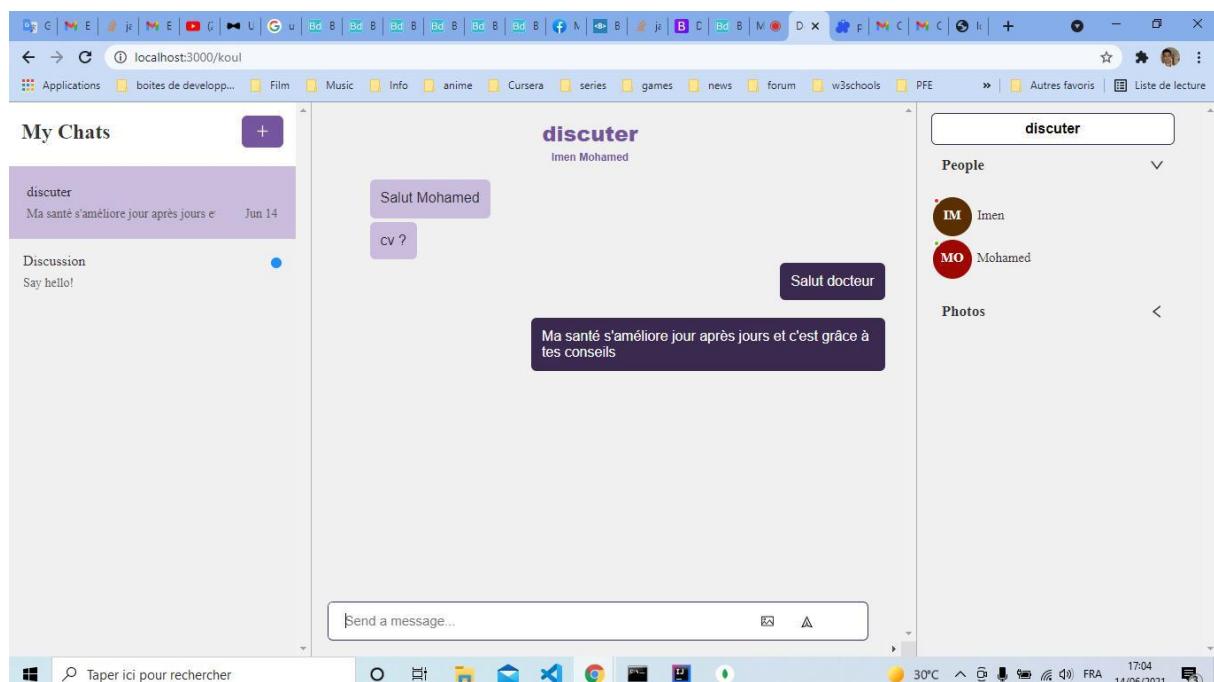


Figure 51: Interface Messagerie

## 2.10 Interface Discussion vidéo en ligne

Une téléconsultation se déroule comme une consultation en présentiel, en partant d'une demande de rendez-vous ou sur proposition du médecin traitant. En amont de la consultation, le médecin envoie un lien Internet à son patient, l'invitant à se connecter à l'heure prévue du rendez-vous à une application sécurisée, depuis un ordinateur équipé d'une webcam et relié à Internet. La figure 52 présente un exemple réel d'une discussion vidéo.

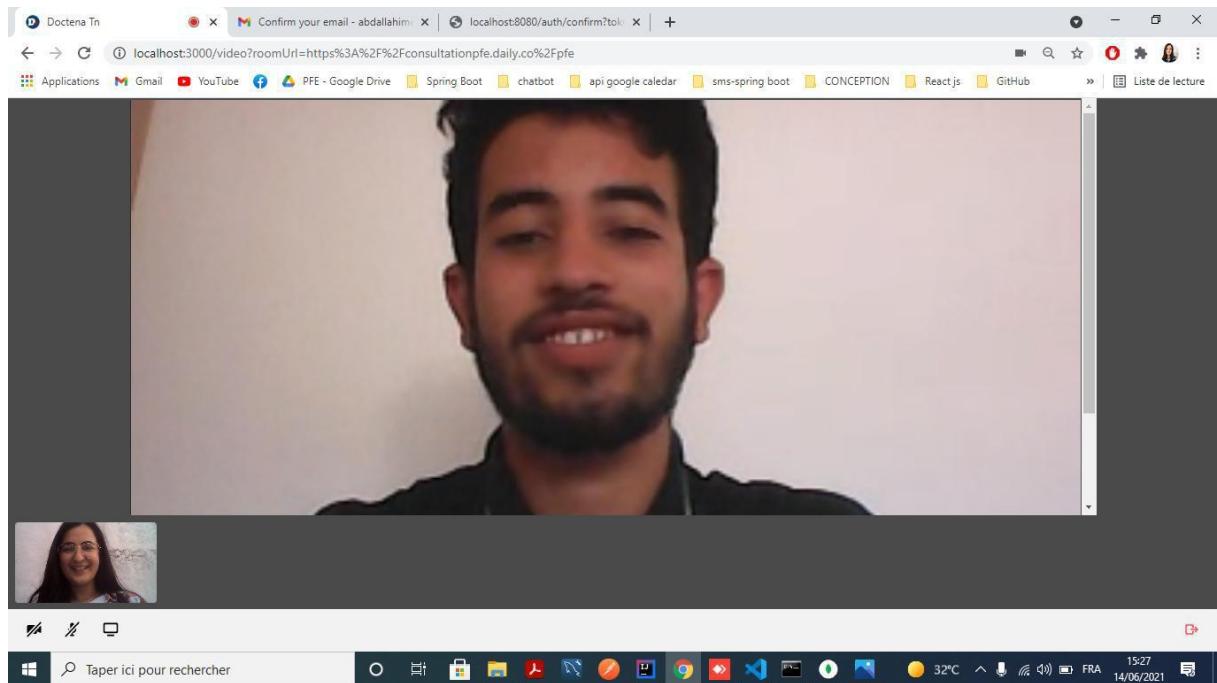


Figure 52: Interface Discussion vidéo

## Conclusion

La phase de réalisation est la phase la plus difficile et la plus délicate. Elle intègre en même temps les informations et les fonctionnalités de tous les acteurs de la plateforme : le patient, le médecin, la pharmacie et le laboratoire. Il est primordial de surveiller le fonctionnement de la plateforme web une fois intégrée, d'une part, pour pouvoir dégager les éventuelles anomalies, d'autre part, pour pouvoir intégrer les améliorations nécessaires.

## CONCLUSION GÉNÉRALE

Le présent rapport est réalisé dans le cadre de notre projet de fin d'études en vue de l'obtention du diplôme de Licence Fondamentale en Sciences Informatiques.

Ce travail a pour objectif de concevoir et de développer une plateforme de téléconsultation médicale en ligne appelée « doctena.tn ». De ce fait, notre système englobe plusieurs médecins de différentes spécialités exerçant en Tunisie. Nous avons ainsi une riche base de données de médecins qui sera disponible aux patients.

Pour pouvoir compléter notre mission, nous avons détaillé les différentes étapes d'analyse, de conception et de réalisation de la solution proposée qui consiste en une plateforme web de téléconsultation médicale en ligne.

Dans le premier chapitre, nous avons commencé par présenter le cadre général du projet, l'étude du projet, l'analyse et la critique de l'existant. Puis, nous avons présenté une étude comparative des différentes méthodes de développement de projets. Ensuite, nous avons décrit la méthode adoptée pour ce projet. Le deuxième chapitre consiste à identifier les acteurs et à collecter les besoins fonctionnels et les besoins non fonctionnels. Une fois nos objectifs fixés, nous avons enchaîné dans le troisième chapitre une conception afin de mener à bien notre projet. Nous avons utilisé UML comme langage de modélisation, 2TUP comme processus de développement et MVC comme patron de conception. Dans le dernier chapitre, nous avons décrit l'environnement du travail. Puis, nous avons enchaîné avec les différentes interfaces de l'application.

Néanmoins, notre travail ne s'arrête pas à ce niveau ; en fait notre application peut être améliorée et enrichie par de nouvelles fonctionnalités :

- Développement d'une application mobile,
- Intégration d'équipements, dispositifs médicaux connectés (IOT),
- Le chatbot (IA) pour rendre les recherches plus simples, plus rapides et plus sécurisées pour libérer du temps de consultation, au profit de l'écoute du patient.

## Références bibliographiques

- [1] <https://enano-tech.com> [Consulté le 20-03-2021]
- [2] <https://www.tobba.tn> [Consulté le 10-04-2021]
- [3] <https://helloconsult.com> [Consulté le 10-04-2021]
- [4] <https://www.med.tn> [Consulté le 10-04-2021]
- [5] <https://www.leah.care> [Consulté le 10-04-2021]
- [6] <https://www.doctolib.fr> [Consulté le 13-04-2021]
- [7] <https://www.medaviz.com> [Consulté le 13-04-2021]
- [8] <https://teledactyl.com> [Consulté le 15-04-2021]
- [9] <http://doctoondemand.com> [Consulté le 15-04-2021]
- [10] <https://www.babylonhealth.com> [Consulté le 15-04-2021]
- [11] <https://blog.dcube.fr/index.php/2014/04/28/scrum-vs-cycle-en-v-2-2> [Consulté le 02-06-2021]
- [12] <https://www.planzone.fr/blog/quest-ce-que-la-methodologie-extreme-programming>  
[Consulté le 02-06-2021]
- [13] <https://www.stillgeek.com/lextreme-programming-xp-un-framework-agile>  
[Consulté le 02-06-2021]
- [14] <https://www.nutcache.com/fr/blog/methode-rup> [Consulté le 05-06-2021]
- [15] <https://www.nutcache.com/fr/blog/methode-> [Consulté le 05-06-2021]
- [16] [http://www.solidbyte.com/Services/Meilleures\\_Pratiques](http://www.solidbyte.com/Services/Meilleures_Pratiques) [Consulté le 05-06-2021]
- [17] [https://baripedia.org/wiki/Mod%C3%A8les\\_de\\_processus\\_de\\_d%C3%A9veloppement\\_des\\_SI](https://baripedia.org/wiki/Mod%C3%A8les_de_processus_de_d%C3%A9veloppement_des_SI)  
[Consulté le 05-06-2021]

- [18] [https://www.memoireonline.com/10/18/10437/m\\_Developpement-d-une-application-d-apprentissage-de-langue3.html](https://www.memoireonline.com/10/18/10437/m_Developpement-d-une-application-d-apprentissage-de-langue3.html) [Consulté le 07-06-2021]
- [19] [https://www.memoireonline.com/05/13/7195/m\\_Mise-en-place-d'une-application-webmapping-de-geolocalisation-des-points-d'intert-de-la-ville6.html#:~:text=2TUP%20est%20un%20processus%20unifi%C3%A9](https://www.memoireonline.com/05/13/7195/m_Mise-en-place-d'une-application-webmapping-de-geolocalisation-des-points-d'intert-de-la-ville6.html#:~:text=2TUP%20est%20un%20processus%20unifi%C3%A9)  
[Consulté le 07-06-2021]
- [20] F. V. Pascal Roques, *UML2 en action De l'analyse des besoins à la conception.*  
[Consulté le 10-06-2021]
- [21] F. Abir, *Mémoire de fin d'étude*, 19/12/2020. [Consulté le 10-05-2021]