

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE

École supérieure Privée  
Technologie et Ingénierie



Rapport de Stage d'été

---

Conception et développement d'une plateforme médicale

---



Réalisé par:  
Bani Salmen

Encadré par:  
Bani Sadok

Année universitaire: 2023/2024

# Table des matières

<b>Introduction Générale</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Organisme d'accueil : . . . . .	3
1.2.1 Présentation : . . . . .	3
1.2.2 Services de l'entreprise : . . . . .	4
1.3 Contexte du stage . . . . .	6
1.4 Conclusion : . . . . .	7
<b>2 État de l'art</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Étude du projet : . . . . .	8
2.2.1 Étude de l'existant : . . . . .	8
2.2.2 Critique de l'existant : . . . . .	9
2.2.3 Solution proposée : . . . . .	9
2.3 Conclusion : . . . . .	10
<b>3 Méthodologies et environnement de travail</b>	<b>11</b>
3.1 Introduction : . . . . .	11
3.2 Méthodologies de conception : . . . . .	11
3.2.1 Méthodes agiles : . . . . .	11
3.2.2 Méthode agile adoptée : Scrum . . . . .	13
3.2.3 Les principales caractéristiques de Scrum sont : . . . . .	13
3.2.4 L'équipe de SCRUM : . . . . .	13
3.2.5 Pourquoi Scrum ? . . . . .	14
3.3 Architecture adoptée . . . . .	14
3.3.1 Architecture MVC . . . . .	14

3.3.2	Architecture MVVM . . . . .	15
3.4	Environnement de travail . . . . .	16
3.4.1	Environnement matériel : . . . . .	16
3.4.2	Environnement logiciel : . . . . .	17
3.5	Conclusion . . . . .	18
<b>4</b>	<b>Spécification des besoins</b>	<b>19</b>
4.1	Introduction . . . . .	19
4.2	Analyse des besoins . . . . .	19
4.2.1	Description de contexte de projet : . . . . .	19
4.2.2	Identification des acteurs : . . . . .	20
4.2.3	Identification des besoins : . . . . .	20
4.2.4	Pilotage du projet avec Scrum : . . . . .	22
4.3	Conclusion . . . . .	27
<b>5</b>	<b>Release 1</b>	<b>28</b>
5.1	Introduction . . . . .	28
5.2	Organisation des sprints . . . . .	28
5.3	Sprint 1 . . . . .	29
5.3.1	Sprint Backlog produit . . . . .	29
5.3.2	Descriptions textuelles des cas d'utilisation . . . . .	29
5.3.3	Diagramme de classe Sprint 1 . . . . .	32
5.3.4	Réalisation et Test . . . . .	33
5.4	Sprint 2 . . . . .	36
5.4.1	Sprint Backlog produit . . . . .	36
5.4.2	Descriptions textuelles des cas d'utilisation . . . . .	36
5.4.3	Diagramme de classe Sprint 2 . . . . .	37
5.4.4	Réalisation et Test . . . . .	38
5.5	Conclusion . . . . .	39
<b>6</b>	<b>Release 2</b>	<b>40</b>
6.1	Introduction . . . . .	40
6.2	Organisation des sprints . . . . .	40
6.3	Sprint 3 . . . . .	40
6.3.1	Sprint Backlog produit . . . . .	40
6.3.2	Descriptions textuelles des cas d'utilisations . . . . .	41
6.3.3	Diagramme de classe Sprint 3 . . . . .	45
6.3.4	Réalisation et Test . . . . .	45
6.4	Sprint 4 . . . . .	47
6.4.1	Sprint Backlog produit . . . . .	47
6.4.2	Descriptions textuelles des cas d'utilisations . . . . .	48

6.4.3	Diagramme de classe Sprint 4 . . . . .	49
6.4.4	Réalisation et Test . . . . .	50
6.5	Conclusion . . . . .	52
<b>Conclusion Générale</b>		<b>53</b>

# Table des figures

1.1	Siège social de Talan Tunisie . . . . .	4
1.2	Secteurs d'activité de Talan Tunisie . . . . .	5
3.1	Méthodologie agile vs. Classique . . . . .	12
3.2	Fonctionnement général de la méthode Scrum . . . . .	13
3.3	L'architecture MVC . . . . .	15
3.4	L'architecture MVVM . . . . .	16
4.1	Diagramme de cas d'utilisation global . . . . .	26
4.2	Diagramme de classe global . . . . .	27
5.1	Diagramme de classe <u>Sprint 1</u> . . . . .	33
5.2	Interface <u>Inscription Patient</u> . . . . .	34
5.3	Interface <u>Inscription Médecin</u> . . . . .	34
5.4	Interface <u>Authentification</u> . . . . .	35
5.5	Interface <u>Gestion Spécialité</u> . . . . .	36
5.6	Diagramme de classe <u>Sprint 2</u> . . . . .	38
5.7	Interface <u>Gestion Notifications</u> . . . . .	39
6.1	Diagramme de classe <u>Sprint 3</u> . . . . .	45
6.2	Interface <u>Analyse</u> . . . . .	46
6.3	Interface <u>Maladie</u> . . . . .	46
6.4	Interface <u>Mesure</u> . . . . .	47
6.5	Diagramme de classe <u>Sprint 4</u> . . . . .	50
6.6	Interface <u>Ajout rendez-vous</u> . . . . .	51
6.7	Interface <u>Gestion rendez-vous</u> . . . . .	52

# Liste des tableaux

2.1	Tableau comparatif . . . . .	9
3.1	Environnement matériel . . . . .	17
4.1	Identification des acteurs . . . . .	20
4.2	Tableau du Backlog produit . . . . .	23
4.3	Planification des Sprints . . . . .	24
4.4	Planification des Releases . . . . .	25
5.1	Tableau du Backlog produit de Sprint 1 . . . . .	29
5.2	Description textuelle du cas d'utilisation Inscription . . . . .	30
5.3	Description textuelle du cas d'utilisation S'authentifier . . . . .	31
5.4	Description textuelle du cas d'utilisation Supprimer spécialité . . . . .	31
5.5	Description textuelle du cas d'utilisation Ajouter spécialité . . . . .	32
5.6	Description textuelle du cas d'utilisation Modifier spécialité . . . . .	32
5.7	Description textuelle de cas d'utilisation <u>Valider Médecin</u> . . . . .	37
5.8	Description textuelle de cas d'utilisation <u>Supprimer Médecin</u> . . . . .	37
6.1	Tableau du Backlog produit de Sprint 3 . . . . .	41
6.2	Description textuelle du cas d'utilisation <u>Ajouter analyse</u> . . . . .	42
6.3	Description textuelle du cas d'utilisation <u>Supprimer analyse</u> . . . . .	42
6.4	Description textuelle du cas d'utilisation <u>Ajouter maladie</u> . . . . .	43
6.5	Description textuelle du cas d'utilisation <u>Supprimer maladie</u> . . . . .	43
6.6	Description textuelle du cas d'utilisation <u>Ajouter mesure</u> . . . . .	44
6.7	Description textuelle du cas d'utilisation <u>Supprimer mesure</u> . . . . .	44
6.8	Tableau du Backlog produit de Sprint 4 . . . . .	47
6.9	Description textuelle du cas d'utilisation <u>Ajouter rendez-vous</u> . . . . .	48
6.10	Description textuelle du cas d'utilisation <u>valider rendez-vous</u> . . . . .	49
6.11	Description textuelle du cas d'utilisation <u>Supprimer rendez-vous</u> . . . . .	49

# Introduction Générale

Au cours de ces dernières années, les ordinateurs et les smartphones (téléphones intelligents) ont connu un véritable essor et sont de plus en plus dotés d'une puissance avec des fonctionnalités assez évoluées. Celles-ci favorisent la création et la distribution en toute simplicité des applications mobiles et web innovantes destinées à satisfaire les besoins des utilisateurs. L'informatique médicale est en plein essor et présente un avantage particulier pour l'innovation.

Les médecins les laboratoires et les cliniques se digitalisent de plus en plus afin d'améliorer leur performance globale de façon qu'ils essayent de profiter au maximum possible pour augmenter leur productivité de travail et pour que la gestion du travail devienne plus agile, confirmant ainsi la véritable révolution culturelle et technologique que vit depuis quelques années, la médecine.

Soigner plus : plus de patients, plus vite, plus efficacement, mais avec moins : moins de ressources, de personnel, de budget : voilà le challenge auquel se trouve confronté aujourd'hui le monde de la santé en général, celui d'une polyclinique en particulier !

C'est dans ce cadre que s'inscrit notre projet intitulé « **Gestion d'une plateforme médicale** » au sein de la société Talan Tunisie.

Ce projet a pour but de concevoir et de développer une application web qui gère les rendez-vous et qui va recouvrir tout le processus médicale.

Dans ce rapport, nous avons choisi de détailler le déroulement du projet en cinq chapitres articulés comme suit :

- Dans un premier chapitre « **Introduction** » Nous présentons l'entreprise dans laquelle vous avez effectué mon stage d'été.
- Le second chapitre « **état de l'art** » , nous mettons notre sujet dans son

---

contexte général, nous présentons notre établissement et nous présentons notre application avec ses différentes problématiques, notre solution proposée .

- Le troisième chapitre « **Méthodologie et environnement de travail** » nous présentons la méthodologie de réalisation suivie.
- Le quatrième chapitre « **Spécification des besoins** » s'intéresse à l'analyse des besoins fonctionnels et non fonctionnels de notre application, ensuite nous faisons une étude détaillée des différents cas d'utilisations possibles avec les différents scénarios possibles afin de concevoir une application réalisable, fiable et efficace.
- Le cinquième chapitre «**Release 1** » et le sixième chapitre «**Release 2** » sont consacrés aux différents sprints qui comporte chacun trois sections : Analyse, Conception, Réalisation et Test.



# Chapitre 1

## Introduction

### 1.1 Introduction

Pendant mon stage chez Talan Tunisie, j'ai eu l'opportunité de plonger dans le monde dynamique et innovant des technologies de l'information. Cette expérience m'a permis de mettre en pratique mes compétences et de contribuer à des projets informatiques passionnants. Dans ce rapport, je vais partager quelques-unes des expériences les plus marquantes de mon stage chez Talan Tunisie et décrire les compétences que j'ai développées au cours de cette période enrichissante.

### 1.2 Organisme d'accueil :

Dans cette section, nous aurons un aperçu général de Talan Tunisie Consulting et de ses principaux services.

#### 1.2.1 Présentation :

Le projet a été réalisé à Talan Tunisie, une société de conseil spécialisée dans l'intégration des Technologies de l'Information et de la Communication (TIC) dans le domaine de la relation client. Talan Tunisie se concentre sur divers secteurs tels que les opérateurs de services, la finance et l'assurance, les télécoms et les médias, l'énergie et les services publics, ainsi que le transport et la logistique. Talan a été fondée en 2002 par Mehdi Houas, Eric Benamou et Philippe Cassoulat. Elle offre son expertise métier, fonctionnel et technologique à l'échelle internationale, avec plus de 3000 collaborateurs et un chiffre d'affaires de près de 310 millions d'euros. La société est présente en France, à New York, à Hong Kong, au Royaume-Uni et à Montréal.



FIGURE 1.1 : Siège social de Talan Tunisie

Talan Tunisie a obtenu des certifications telles que l'ISO 9001 :2015, l'ISO 27001 :2013, le modèle CMMI, et l'ITIL, attestant de son engagement envers la qualité, la sécurité de l'information, la maturité des processus et la gestion des services informatiques.

### 1.2.2 Services de l'entreprise :

Talan a ouvert en 2008 un centre de développement nearshore en Tunisie appelé "Talan Tunisie Consulting" dans le but d'accélérer son développement. Ce centre emploie actuellement environ 180 ingénieurs spécialisés dans les nouvelles technologies et collabore avec d'importants clients européens.

Les services proposés par Talan incluent le conseil et l'assistance en gestion de projet, le redesign et l'optimisation des processus métier, le support pour les projets de transformation majeurs, ainsi que l'alignement des systèmes d'information sur les changements organisationnels et leur support. La société Talan opère dans les secteurs suivants, comme illustré dans la figure 1.2 :

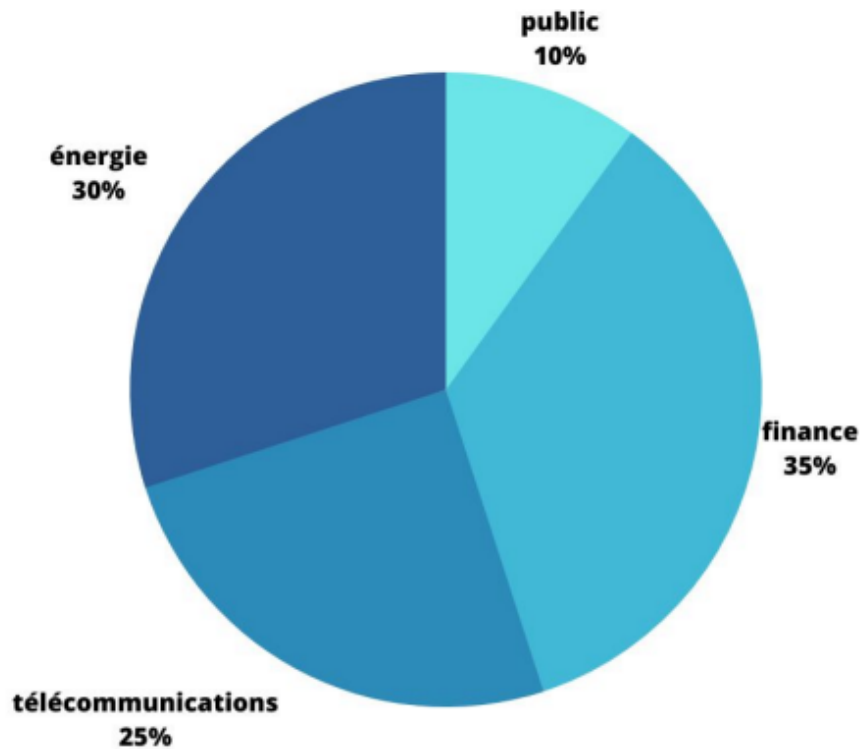


FIGURE 1.2 : Secteurs d'activité de Talan Tunisie

- Le secteur financier : Talan intervient dans plusieurs domaines d'amélioration et d'optimisation dans les banques d'investissement, le secteur de la distribution et l'assurance.
- Le secteur des télécommunications : à travers des projets destinés aux opérateurs de télécommunications et à un ensemble de fournisseurs de services Internet.
- Le secteur public : à travers l'assistance en gestion de projet et l'intégration de nouvelles technologies dans des projets tels que l'administration, la réduction des coûts, la cohérence et l'ouverture des systèmes d'information.
- Le secteur de l'énergie : à travers la ligne de service "Énergie et Services Publics" de Talan, qui regroupe plus de 80 consultants travaillant sur des projets majeurs d'AMOA (Assistance à la Maîtrise d'Ouvrage).

---

## 1.3 Contexte du stage

Le stage d'été se déroule dans le cadre du deuxième cycle d'ingénierie en génie logiciel et se déroule au sein de Talan Tunisie, une entreprise de premier plan spécialisée dans le développement de solutions informatiques. L'objectif principal de ce stage est de me permettre de mettre en pratique mes compétences en génie logiciel en participant au développement d'un site web Médicale. Le projet se base sur l'utilisation de technologies modernes, notamment Spring Boot et Angular.

Voici les détails du contexte du stage :

- Nature du Projet : Le stage est axé sur le développement d'un site web Médicale . Ce site web gère les rendez-vous des patients et va recouvrir tout le processus médicale.
- Technologies Utilisées : e serai amené à travailler avec des technologies de pointe, notamment Spring Boot pour la partie back-end et Angular pour le développement du front-end. Ces technologies sont largement utilisées dans le secteur pour leur robustesse et leur efficacité.
- Développement Front-End et Back-End : Je serai impliqué dans le développement à la fois du front-end et du back-end du site web. Cela comprendra la création d'une interface utilisateur conviviale, la gestion des bases de données, l'intégration de fonctionnalités de rendez-vous, et la sécurisation des transactions en ligne.
- Collaboration en Équipe Agile : Je travaillerai au sein d'une équipe de développement agile composée de développeurs, de chefs de projet, et d'experts en qualité logicielle. Cette collaboration me permettra d'apprendre les meilleures pratiques de développement et de gestion de projet.
- Formation et Encadrement : Talan Tunisie s'engage à fournir une formation approfondie et un encadrement continu pour m'aider à atteindre mes objectifs d'apprentissage et de développement professionnel.
- Objectifs d'Apprentissage : Ce stage d'été offre l'opportunité de mettre en pratique mes compétences, d'acquérir une expérience concrète en développement web, de perfectionner mes compétences techniques, et de contribuer de manière significative à un projet d'envergure.

En résumé, ce stage d'été chez Talan Tunisie, axé sur le développement d'un site web Médicale avec Spring Boot et Angular, offre une expérience d'apprentissage

---

exceptionnelle, me permettant de me familiariser avec des technologies de pointe et de participer activement à un projet d'entreprise innovant.

## **1.4 Conclusion :**

L'objectif principal de mon rapport de stage est de fournir une présentation complète et détaillée de mon expérience au sein de l'entreprise Talan Tunisie. Ce rapport a pour but de mettre en lumière les missions, les projets et les responsabilités qui m'ont été confiés pendant mon stage d'été en génie logiciel, en mettant particulièrement l'accent sur les compétences que j'ai pu développer et les réalisations que j'ai accomplies. En plus de documenter mon expérience, ce rapport vise également à réfléchir sur les apprentissages que j'ai tirés de cette expérience professionnelle, à évaluer ma contribution à l'entreprise, et à formuler des recommandations pertinentes. L'objectif global est de partager un récit authentique de mon stage tout en fournissant des informations utiles à ceux qui s'intéressent à mon parcours académique et professionnel.

# Chapitre 2

## État de l'art

### 2.1 Introduction

Dans ce chapitre, nous fournirons une vue d'ensemble du projet, en décrivant le problème en détail ainsi que la solution proposée pour y remédier.

### 2.2 Étude du projet :

Dans cette section, nous fournirons une vue d'ensemble du projet, en décrivant le problème en détail ainsi que la solution proposée pour y remédier.

#### 2.2.1 Étude de l'existant :

L'étude de l'existant permet de déterminer les points faibles et les points forts des applications actuelles pour pouvoir déterminer les nouveaux besoins, en vue d'en prendre en considération lors de la conception et la réalisation de l'application.

Dans cette section nous présentons deux applications existant déjà la première est sous le nom **Tobba.tn** et la deuxième est sous le nom **Tabibi.tn**.

- **Tobba.tn** : est une plateforme tunisienne de santé numérique qui permet d'offrir des consultations médicales par Internet.
- **Tabibi.tn** est une application web qui permet aux utilisateurs de trouver un médecin sur Internet.

---

### 2.2.2 Critique de l'existant :

La critique de l'existant est une phase primordiale qui se fait après l'étude de l'existant. Cette étape a pour objectif la découverte et la précision des erreurs de l'application actuelle. En fait, les deux applications Tobba.tn et Tabibi.tn souffrent de plusieurs anomalies et les insuffisances.

Voici un tableau comparatif où nous citons quelques points faible de chaque application :

<b>Tobba.tn :</b>	<b>Tabibi.tn :</b>
L'inscription d'un praticien peut prendre un temps.	Les fonctionnalités sont limitées aux médecins seulement, pas d'autres praticiens de santé.
L'interface homme machine est complexe et non compréhensible.	Pas de changement des informations entre les édecins et les patients
Pas de changement des informations entre les praticiens et les patients.	Des problèmes techniques (le service est lent , l'inscription est inopérante).

TABLE 2.1 : Tableau comparatif

### 2.2.3 Solution proposée :

Après avoir examiné le fonctionnement actuel des deux applications, nous allons essayer d'améliorer la qualité de système de gestion médicale. Donc, il est nécessaire de mettre en place des solutions informatiques qui permettront aux utilisateurs de gagner en termes de temps et d'efficacité :

Notre idée est de développer une plate-forme web innovante. Elle a le rôle de faire :

- **Répondre à tous les besoins** des utilisateurs.
- **Gain de temps et simplicité** : proposer des solutions pour que l'utilisation et la manipulation de l'application soient faciles.
- **L'envoi des captures des fichiers** pour faciliter la communication et l'échange de données entre les praticiens et les patients.

---

## 2.3 Conclusion :

Dans ce chapitre, nous avons introduit de l'étude d'existant, mettre l'accent sur la solution actuelle et la solution proposée. Dans le chapitre suivant, nous allons analyser les besoins fonctionnels et non fonctionnels.



# Chapitre 3

## Méthodologies et environnement de travail

### 3.1 Introduction :

Ce chapitre se concentre sur les méthodologies et les outils utilisés pour créer un environnement de travail efficace. Nous explorons les différentes approches, techniques et technologies qui ont été sélectionnées pour mener à bien notre recherche. En mettant l'accent sur la collecte de données, l'analyse, la modélisation et l'utilisation d'outils spécifiques, nous visons à optimiser notre productivité et à garantir la qualité de notre travail.

### 3.2 Méthodologies de conception :

Pour pouvoir choisir la bonne méthodologie, il faut avoir une idée sur les avantages et les inconvénients de quelques-unes et savoir ainsi laquelle répond aux contraintes et aux exigences de notre projet. Dans ce qui suit nous présentons un aperçu sur les différentes méthodologies.

#### 3.2.1 Méthodes agiles :

La méthodologie est une démarche organisée rationnellement pour aboutir à un résultat. Parmi les différentes méthodologies existantes, nous pouvons citer les méthodes classiques tel que le modèle **en cascade** utilisée souvent dans les simples projets dont les besoins sont clairs et bien définis dès le début, le modèle **en Y** utilisé pour le développement des applications mobiles, ainsi que le processus unifié et les méthodologies agiles tels que **Scrum**, **Programming (XP)** et autres caractérisées par leurs souplesses et utilisées dans des projets complexes.

---

Brièvement, les méthodologies agiles sont basées sur 4 valeurs clé qui les différencient des approches classiques :

- L’interaction avec les personnes plutôt que les processus et les outils
- Un produit opérationnel plutôt qu’une documentation pléthorique
- La collaboration avec le client plutôt que la négociation de contrat
- La réactivité face au changement plutôt que le suivi d’un plan

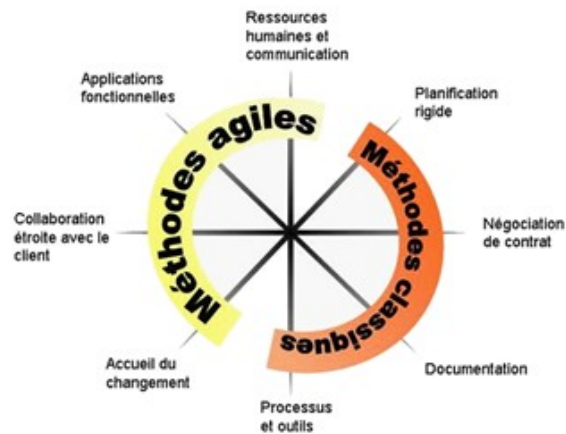


FIGURE 3.1 : Méthodologie agile vs. Classique

---

### 3.2.2 Méthode agile adoptée : Scrum

Pour mener à bien notre projet et de garantir le bon déroulement des différentes phases, nous avons opté pour Scrum comme méthodologie de gestion de projet.

« SCRUM » tient son origine du terme sportif de rugby signifiant : mêlée. Tout comme cet aspect technique de la partie du jeu, la méthodologie demande à ses acteurs d'être soudés dans l'accomplissement d'un projet, dans l'atteinte d'un but. Elle s'appuie sur le découpage des projets en itérations encore nommées « sprints ». Un sprint peut avoir une durée qui varie généralement entre deux semaines et un mois

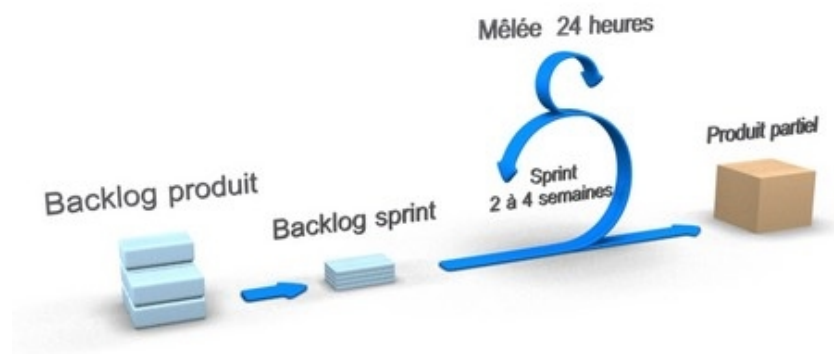


FIGURE 3.2 : Fonctionnement général de la méthode Scrum

### 3.2.3 Les principales caractéristiques de Scrum sont :

- Itératif, lié à des processus incrémentaux.
- Fait pour développer des produits/applications nécessitant une grande adaptabilité.
- Augmenter la communication et maximiser la coopération.
- Un moyen d'augmenter la productivité.

### 3.2.4 L'équipe de SCRUM :

- **Le Product Owner (le propriétaire du produit) :** c'est une personne qui porte la vision du produit à réaliser, généralement c'est un expert dans le domaine.
- **Le Scrum Master (le directeur de produit) :** c'est la personne qui doit assurer le bon déroulement des différents sprints de la release, et qui doit impérativement maîtriser Scrum.

- 
- **Le Scrum Team (l'équipe de Scrum)** : constitué des personnes qui seront chargées d'implémenter les différents besoins du client. Bien évidemment, cette équipe sera constituée des développeurs, des testeurs, etc.

Dans le contexte de notre projet, Bani Sadok est le Product Owner Bani Sadok est Le Scrum Master et nous sommes tous Bani Salmen et Khemir Malek l'équipe de développement.

### 3.2.5 Pourquoi Scrum ?

Nous avons choisi Scrum parmi d'autres méthodes pour sa réponse et correspondance aux besoins de notre projet, puisque Scrum a :

- Une grande capacité d'adaptation au changement grâce à des itérations courtes.
- Une bonne organisation du travail.
- Plus de souplesse et de réactivité.

## 3.3 Architecture adoptée

Pour créer une application flexible, nous avons choisi l'architecture MVC pour la partie Backend et l'architecture MVVM pour le Frontend.

### 3.3.1 Architecture MVC

Suite à l'utilisation de modèle MVC, l'architecture correspond au modèle MVC est l'architecture 3-tiers . C'est une architecture simple à trois niveaux dans laquelle les demandes sont envoyées du navigateur vers la base de données, via un serveur Web.

- Le client : L'ordinateur demandeur de ressources, équipé d'une interface utilisateur généralement un navigateur Web chargé par la présentation.
- Le serveur de l'application web : Ce serveur fournit la ressource en faisant appel à un autre serveur. Pour notre cas, c'est la partie traitement qui utilise java comme langage applicatif.
- Le serveur de base de données : Ce serveur permet de stocker et restituer les données. Il fournit au serveur d'application les données dont il a besoin.

Le modèle MVC décrit une manière d'architecturer une application informatique en la décomposant en 3 sous-parties :

- 
1. Un modèle (Model) : Le modèle responsable de la gestion des données de l'application .Il reçoit une entrée utilisateur de contrôleur.
  2. Une vue (View) : La vue signifie la présentation de modèle dans un format particulier.
  3. Un contrôleur (Controller) : Le contrôleur répond à l'entrée de l'utilisateur et effectue des interactions sur les objets de modèle .Le contrôleur reçoit l'entrée, la valide puis la transmet au modèle.

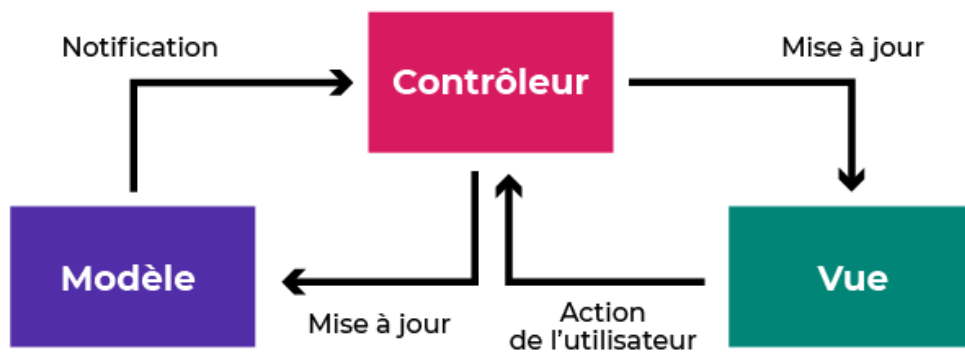


FIGURE 3.3 : L'architecture MVC

### 3.3.2 Architecture MVVM

Le Modèle –Vue-Vue Modèle (en abrégé MVVM, de l'anglais (Model View View-Model)) est une architecture et une méthode de conception utilisée dans le génie logiciel. MVVM est originaire de Microsoft et adapte pour le développement des applications basés sur les technologies Windows Présentation Fondation et Silverlight via l'outil MVVM Light par exemple. Cette méthode permet, tel le modèle MVC (Modèle-vue-Contrôleur), de séparer la vue de la logique et d'accès aux données en accentuant les principes de binding et d'évènement.

- Modèle (Model) : est écrit en HTML et CSS et contient des éléments angular spécifiques (JavaScript objects). Ces modèles sont connectés au ViewModel via une liaison de données.

- 
- Vue (View) : couche visuel de l'application y compris tous les éléments visuels affichés sur l'écran de l'utilisateur et définit dans un modèle (View-Model) : Contient toute la logique de la page. En mode angular, Le VM est représenté la partie Type Script d'un composant (alias la logique de présentation). Nous avons donc découvert qu'un composant angulaire est composé de deux parties : Modèle (View) et une classe (ViewModel).

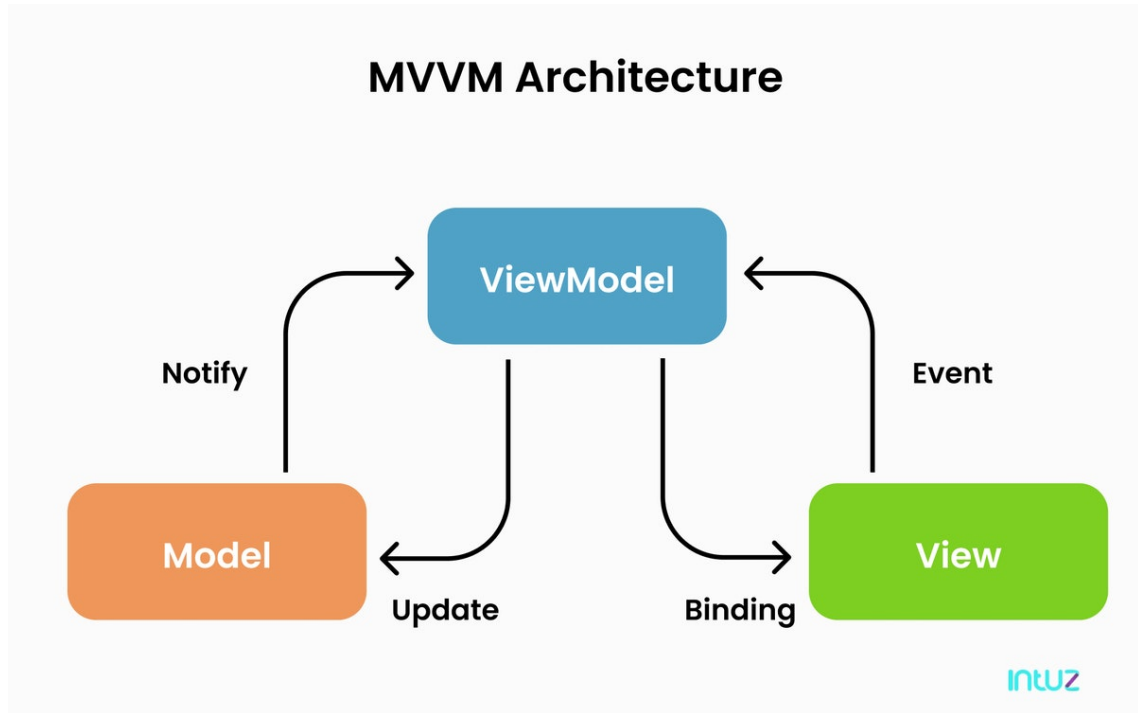


FIGURE 3.4 : L'architecture MVVM

## 3.4 Environnement de travail

Notre travail est réalisé en utilisant un système d'exploitation de type Windows Version Windows 10.

### 3.4.1 Environnement matériel :

Durant la réalisation de notre projet, nous disposons de trois ordinateurs portables qui possèdent les caractéristiques suivantes :

---

<b>Marque :</b>	DELL	HP	HP
<b>Processeur :</b>	Intel i5	Intel i5	Intel i5
<b>Mémoire vive :</b>	4GB	4GB	20GB

TABLE 3.1 : Environnement matériel

### 3.4.2 Environnement logiciel :

Nous avons utilisé des outils logiciels pour le développement de notre application web dont nous citons :

#### 1. Logiciel :

- **Spring Tool Suite** : est une application basée sur Eclipse facilitant la création de projet Spring.
- **Visual Studio Code** : est un éditeur de code source qui peut être utilisé avec une variété de langages de programmation, notamment Java, JavaScript, Go, Node.js et C++.
- **Node JS** : est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau évènementielles hautement concurrentes qui doivent pouvoir monter en charge.
- **Star UML** : est un logiciel de modélisation UML, qui a été cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL.
- **Xampp** : est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.
- **Nutcache** : est un logiciel complet de gestion de projet collaborative. Il permet de suivre les heures travaillées sur l'ensemble des projets et tâches et de mesurer la productivité d'une équipe en temps réel.

---

## 2. Framework :

- **Bootstrap** : est un Framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce Framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce Framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran.
- **Spring Boot** : est un framework de développement applicatif Java open source. Il est particulièrement recommandé pour le développement d'API.
- **Angular** : est un Framework open source écrit en JavaScript qui permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.

## 3. Langage :

- **Java** : est un langage de programmation orienté objet permet de développer des applications client-serveur.
- **SQL** : est le langage de gestion de base de données le plus populaire. Il permet de définir, manipuler et protéger des données de manière simple et schématique.
- **Java Script** : est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet de créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe.

## 3.5 Conclusion

Dans ce chapitre, nous avons présenté les différentes technologies utilisées dans l'implémentation de notre projet.



# Chapitre 4

## Spécification des besoins

### 4.1 Introduction

Dans ce chapitre, nous allons commencer à préparer la première partie de la méthodologie SCRUM appelée notamment « planification du projet » ou bien « sprint 0 ».

C'est une étape primordiale dans le cycle de développement d'un projet. Elle permet de mieux comprendre le travail demandé en dégagant les besoins des différents utilisateurs. Au cours de ce chapitre, nous allons déterminer les besoins et les fonctionnalités du projet, identifier les acteurs actifs, produire le Backlog du produit et enfin, nous allons planifier la répartition initiale des sprints.

### 4.2 Analyse des besoins

Tout au long de cette partie, nous allons identifier et préciser les acteurs, les besoins fonctionnels ainsi que les besoins non fonctionnels pour satisfaire les besoins de notre application.

#### 4.2.1 Description de contexte de projet :

Notre but est la mise en place d'une plate-forme médicale qui regroupe plusieurs fonctionnalités afin de faciliter le travail des praticiens ainsi que la prise des rendez-vous pour les patients . Ce système doit offrir aux utilisateurs, une interface conviviale et simple, leur permettant d'accéder et de bien manipuler cette plate-forme.

---

### 4.2.2 Identification des acteurs :

Un acteur est une entité, physique (personne) ou morale (organisation), extérieure au système étudié, et qui interagit directement avec le système. Les acteurs qui interagissent avec notre application sont :

Acteurs :	Rôle :
Administrateur :	C'est la personne qui a un accès illimité à l'application. Cet acteur va utiliser toutes les fonctionnalités de l'application et peut gérer toute la base de données.
Utilisateur :	On a trois types d'utilisateurs : <ul style="list-style-type: none"><li>• médecin</li><li>• Patient</li><li>• Visiteur</li></ul> Cet acteur va utiliser certaines fonctionnalités de l'application.

TABLE 4.1 : Identification des acteurs

### 4.2.3 Identification des besoins :

Au cours de cette partie, nous allons présenter les fonctionnalités demandées de notre projet tout en spécifiant les besoins fonctionnels et non fonctionnels.

#### 4.2.3.1 Besoins Fonctionnels :

Les besoins fonctionnels (ou un use cas en terme UML) expriment une action que doit effectuer le système en réponse à une demande (sorties qui sont produites pour un ensemble donné d'entrées). Les fonctionnalités de notre système sont regroupées en quatre catégories :

- **Vis à vis l'administrateur :**
  - **Authentication :** Pour accéder à l'application l'administrateur doit saisir son login et son mot de passe.

- 
- **Gestion Médecins** : L'administrateur peut valider ou supprimer la demande d'inscription d'un médecin.
  - **Gestion des spécialités** : L'administrateur peut ajouter, modifier, supprimer une spécialité ou consulter la liste des spécialités.
  - **Vis à vis le médecin** :
    - **Authentification** : Pour accéder à l'application le médecin doit saisir son login et son mot de passe.
    - **Inscription** : Le médecin peut faire remplir le formulaire de l'inscription.
    - **Gestion des rendez-vous** : Le médecin peut valider, supprimer un rendez-vous ou consulter la liste des rendez-vous.
    - **Consulter dossier médical** : Le médecin peut consulter le dossier médical de son patient.
  - **Vis à vis le patient** :
    - **Authentification** : Pour accéder à l'application le patient doit saisir son login et son mot de passe.
    - **Inscription** : Le patient peut faire remplir le formulaire de l'inscription.
    - **Prendre un rendez-vous** : Le patient peut Prendre un rendez-vous chez un médecin.
    - **Gestion dossier médical** : Le patient peut gérer son dossier médical en ajoutant des analyses, des mesures ou des maladies.
    - **Consulter liste des médecines** : Le patient consulter les listes des médecines.
  - **Vis à vis le visiteur** :
    - **Inscription** : Le visiteur peut faire remplir le formulaire de l'inscription.
    - **Consulter liste des médecines** : Le visiteur consulter les listes des médecines.

#### 4.2.3.2 Besoins Non Fonctionnels :

Pour garantir une bonne utilisation de l'application, il existe des principes à respecter au cours de développement :

- **L'ergonomie** : Le système devra offrir aux utilisateurs une interface organisée, compréhensible et riche (message d'erreurs...).

- 
- **La confidentialité** : Assurer la protection de l'information et des données afin qu'elles ne puissent pas être lues ou modifiées par des personnes ou des systèmes non autorisés.
  - **Extensibilité** : l'application doit permettre à tout moment l'ajout d'autres modules.
  - **La performance** : Le logiciel à développer doit être performant à travers ses fonctionnalités et répond à toutes les exigences des usagers d'une manière optimale.
  - **La maintenance** : Le code doit être compréhensible et modulaire pour des raisons de réutilisation et de modification.

#### 4.2.4 Pilotage du projet avec Scrum :

Dans la méthode agile de gestion de projet, Scrum utilise des sprints comme intervalles de temps pendant lesquels l'équipe va compléter un certain nombre de tâches du backlog. L'intervalle de temps dépend des besoins de l'équipe, mais deux semaines est une durée communément utilisée.

Dans cette partie, nous allons présenter au premier lieu le Backlog du produit en deuxième lieu le diagramme de cas d'utilisation global et nous allons finir par le diagramme de classe global.

##### 4.2.4.1 Backlog du produit :

Le Backlog produit est l'artefact le plus important de Scrum, en effet il liste toutes les fonctionnalités, besoins, améliorations et correctifs attendus de produit. Dans cette partie, nous présentons le tableau du Backlog Produit qui contient toutes les fonctionnalités de notre application :

---

<b>Fonctionnalités :</b>	<b>User Stories :</b>	<b>Sprint :</b>
Authentification	En tant que utilisateur ou administrateur je me connecte grâce à un login et un mot de passe.	1
Inscription	En tant que utilisateur je peut inscrire en remplissant le formulaire d'inscription.	1
Gestion des spécialités	En tant que administrateur je peux gérer les spécialités (Ajouter, Supprimer ou Modifier une spécialité).	1
Gestion des médecines	En tant que administrateur je peux gérer les médecines (Valider ou Supprimer la demande d'inscription d'un médecin).	2
Gestion de dossier médicale	En tant que utilisateur (patient) je peux gérer mon dossier médical en ajoutant ou supprimant des analyses , des mesures ou des maladies.	3
Gestion des analyses	En tant que utilisateur (patient) je peux gérer les analyses (Ajouter ou Supprimer une analyse).	
Gestion des maladies	En tant que utilisateur (patient) je peux gérer les maladies (Ajouter ou Supprimer une maladie).	3
Gestion des mesures	En tant que utilisateur (patient) je peux gérer les mesures (Ajouter ou Supprimer une mesure).	3
Gestion des rendez-vous	En tant que utilisateur (patient) je peux ajouter un rendez-vous et en tant que utilisateur (médecin) je peux valider ou supprimer un rendez-vous.	4

TABLE 4.2 : Tableau du Backlog produit

---

#### 4.2.4.2 Planification des Sprints :

La planification du sprint dans le processus “Scrum”, est l’une des étapes les plus importantes servant à planifier le travail à réaliser au cours du sprint. Elle a pour objectif de définir ce qui peut être livré dans le sprint. La planification du sprint est effectuée en collaboration avec toute l’équipe Scrum. Nous avons réparti notre projet en quatre sprints.

Le tableau ci-dessous représente la liste de sprints ainsi que les fonctionnalités réalisées dans chaque sprint :

Liste des Sprints :	Fonctionnalités :
Sprint1 :	<ul style="list-style-type: none"><li>• Authentification</li><li>• Inscription</li></ul>
Sprint2 :	<ul style="list-style-type: none"><li>• Gestion Médecins</li><li>• Gestion Spécialité</li></ul>
Sprint3 :	<ul style="list-style-type: none"><li>• Gestion dossier médical</li><li>• Gestion Analyse</li><li>• Gestion Maladie</li><li>• Gestion Mesure</li></ul>
Sprint4 :	<ul style="list-style-type: none"><li>• Gestion Rendez-vous</li><li>• Envoie des emails</li></ul>

TABLE 4.3 : Planification des Sprints

---

#### 4.2.4.3 Planification des Releases :

Un plan de release est une séquence de sprints à venir, avec une vision du contenu prévu (les éléments de backlog de produit) de ces sprints. Présenté sous forme de tableau, un plan de release est facile à comprendre pour les clients et utilisateurs.

<b>Releases :</b>	<b>Sprints :</b>
Release1 :	Réalisation de Sprint1 et Sprint2
Release2 :	Réalisation de Sprint3 et Sprint4

TABLE 4.4 : Planification des Releases

#### 4.2.4.4 Diagramme de cas d'utilisation global :

Nous débutons par fournir une vue d'ensemble des différentes fonctionnalités offertes par notre application, à l'aide d'un diagramme de cas d'utilisation global qui est un diagramme UML qui permet de recueillir, d'analyser ,d'organiser les besoins et de recenser les grandes fonctionnalités du système.

Cette figure présente le diagramme de cas d'utilisation de notre application :

Diagramme de cas d'utilisation global

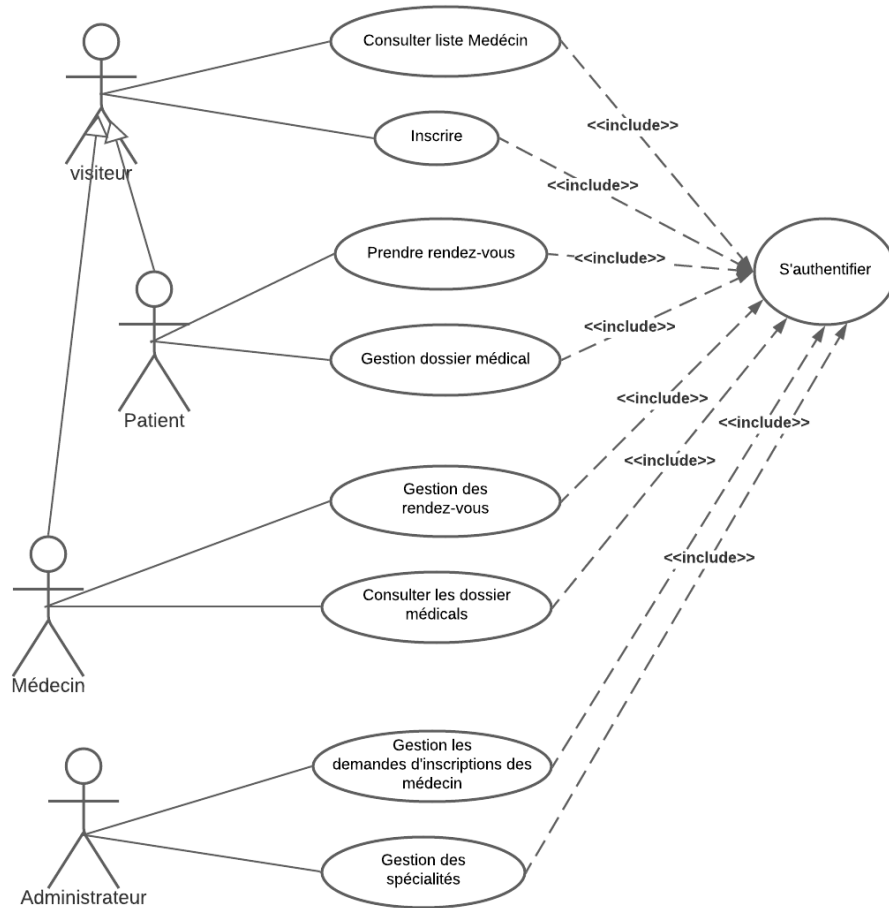


FIGURE 4.1 : Diagramme de cas d'utilisation global

#### 4.2.4.5 Diagramme de classe global :

Le diagramme de classe représente les classes intervenant dans le système. Il est une représentation statique des éléments qui composent un système et de leurs relations. La classe est un concept abstrait qui permet de représenter toutes les entités d'un système. Cette figure illustre le diagramme de classe de notre projet :



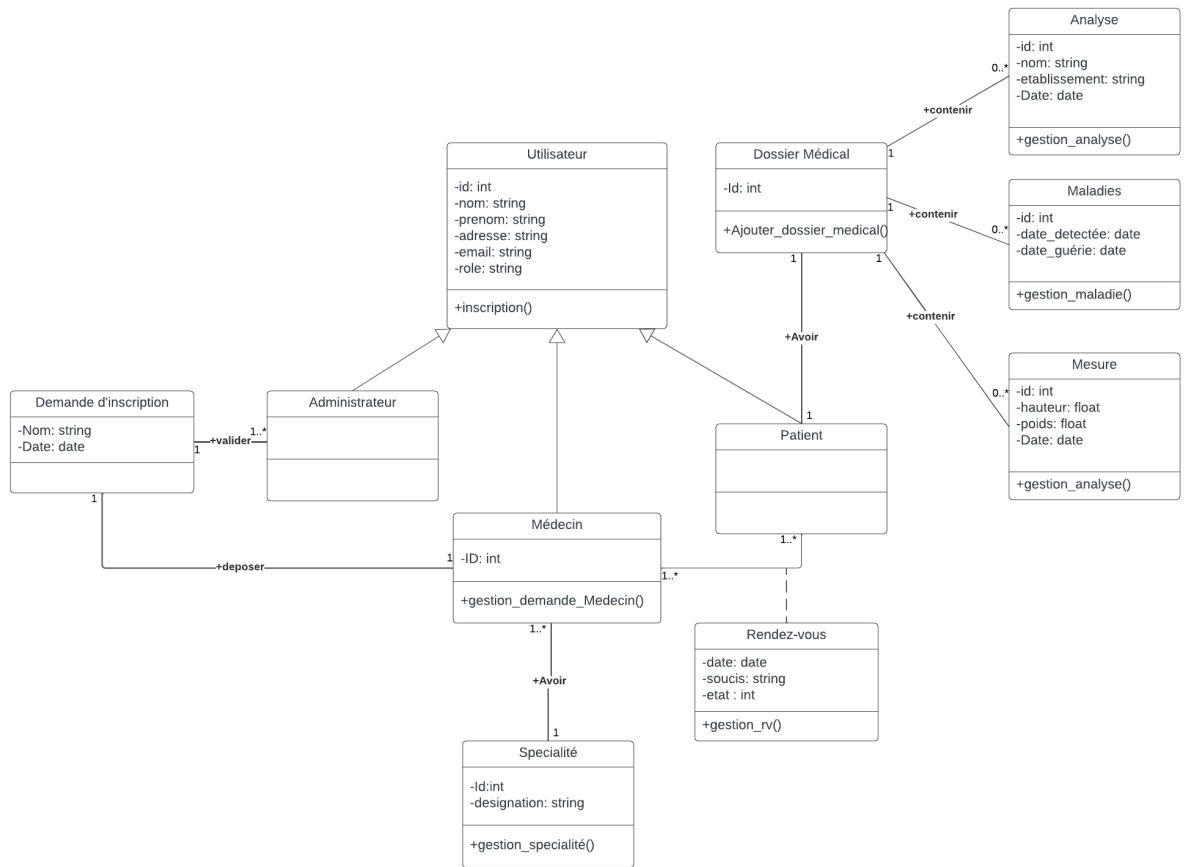


FIGURE 4.2 : Diagramme de classe global

## 4.3 Conclusion

Dans ce chapitre nous avons détaillé les besoins fonctionnels et non fonctionnels de notre projet. Nous avons aussi précisé les acteurs de notre système ainsi que le backlog du produit suivi de diagramme de cas d'utilisation général et de diagramme de classe général. Ainsi que nous avons planifié la réalisation des différents sprints. Le chapitre suivant sera consacré pour la réalisation du sprint 1.

## Release 1

### 5.1 Introduction

Le terme **release** correspond à la livraison d'une version de produit. Par habitude, on parle de release pour considérer la période de temps qui va du début du travail sur cette version jusqu'à sa livraison et qui passe par une série de sprints successifs. Par contre, le terme **sprint** représente le bloc de temps aboutissant à créer un incrément du produit potentiellement livrable. C'est le terme utilisé dans Scrum pour chaque itération. Nous allons, dans ce chapitre, décrire et analyser avec détails les différents sprints du release 1.

### 5.2 Organisation des sprints

Le release 1 est composé de deux sprints :

- **Sprint 1** : Ce sprint permet d'effectuer l'authentification et l'inscription, gérer les spécialités.
- **Sprint 2** : Ce sprint permet d'effectuer la gestion des demandes d'inscription des médecins.

Dans le but de mieux détailler l'aspect fonctionnel de notre projet pour chaque sprint, nous allons respectivement détailler son Backlog, sa vue dynamique : son diagramme de cas d'utilisation ainsi que les diagrammes de séquences associés à chaque cas d'utilisation.

---

## 5.3 Sprint 1

### 5.3.1 Sprint Backlog produit

Dans cette partie, on montre les différentes tâches effectuées pendant ce sprint qui est dédié à la gestion des spécialités et l'inscription, authentification :

ID :	Fonctionnalités :	User Story :
1	Inscription	En tant que utilisateur je veux remplir le formulaire d'inscription.
2	Authentification	En tant que utilisateur je dois m'authentifier afin d'accéder à mon espace.
3	Gestion des spécialités	En tant que administrateur je veux ajouter/-supprimer /modifier des spécialités.

TABLE 5.1 : Tableau du Backlog produit de Sprint 1

### 5.3.2 Descriptions textuelles des cas d'utilisation

Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Il est recommandé de rédiger une description textuelle, car c'est une forme souple qui convient dans bien des situations.

#### 5.3.2.1 Description textuelle de cas d'utilisation « Inscription » :

Le tableau ci-dessous illustre la description textuelle de cas d'utilisation Inscription.

---

<b>Cas d'utilisation</b>	S'inscrire
<b>Acteurs</b>	utilisateur
<b>Pré Condition</b>	Chaque utilisateur doit avoir un formulaire d'inscription
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. Utilisateur demande de s'inscrire.</li> <li>2. Le système affiche l'interface d'inscrire.</li> <li>3. Utilisateur saisie les champs de formulaire puis il valide la connexion.</li> <li>4. Le système vérifie les champs corrects.</li> <li>5. Le système affiche l'interface appropriée.</li> </ol>
<b>Exception</b>	Utilisateur saisie des champs incorrects.Le système affiche un message erroné.

TABLE 5.2 : Description textuelle du cas d'utilisation Inscription

#### 5.3.2.2 Description textuelle de cas d'utilisation « Authentification » :

Le tableau ci-dessous illustre la description textuelle de cas d'utilisation Authentification.

<b>Cas d'utilisation</b>	S'authentifier
<b>Acteurs</b>	utilisateur
<b>Pré Condition</b>	Chaque utilisateur doit avoir son login et son mot de passe.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. Utilisateur demande de s'authentifier.</li> <li>2. Le système affiche l'interface d'authentification.</li> <li>3. Utilisateur saisie son login et mot de passe puis il valide la connexion.</li> <li>4. Le système vérifie le login et le mot de passe correct.</li> <li>5. Le système affiche l'interface appropriée.</li> </ol>
<b>Exception</b>	L'administrateur saisit des champs incorrects. Le système affiche un message.

TABLE 5.3 : Description textuelle du cas d'utilisation S'authentifier

### 5.3.2.3 Description textuelle de cas d'utilisation « Gestion Spécialité » :

Les tableaux ci-dessous illustrent les descriptions textuelles des cas d'utilisation Supprimer, Ajouter et Modifier une spécialité.

<b>Cas d'utilisation</b>	Supprimer spécialité
<b>Acteurs</b>	administrateur
<b>Pré Condition</b>	L'administrateur doit supprimer des spécialité.
<b>Scénario Nominal</b>	L'administrateur demande supprimer un spécialité.
<b>Exception</b>	L'administrateur supprime les champs incorrects. Le système affiche un message erroné.

TABLE 5.4 : Description textuelle du cas d'utilisation Supprimer spécialité

---

<b>Cas d'utilisation</b>	Ajouter spécialité
<b>Acteurs</b>	administrateur
<b>Pré Condition</b>	L'administrateur doit ajouter des spécialités.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur demande d'ajouter une spécialité.</li> <li>2. Le système affiche l'interface d'ajouter spécialité.</li> <li>3. L'administrateur remplit les champs de formulaire puis il valide.</li> <li>4. Le système vérifie les champs corrects.</li> <li>5. Le système enregistre le nouveau spécialité.</li> </ol>
<b>Exception</b>	Utilisateur saisie login et mot de passe incorrect. Le système affiche un message erroné.

TABLE 5.5 : Description textuelle du cas d'utilisation Ajouter spécialité

<b>Cas d'utilisation</b>	Modifier spécialité
<b>Acteurs</b>	administrateur
<b>Pré Condition</b>	L'administrateur doit modifier des spécialités.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur demande de modifier son spécialité.</li> <li>2. Le système affiche l'interface de modifier spécialité.</li> <li>3. L'administrateur remplit les champs de formulaire puis il valide.</li> <li>4. Le système vérifie les champs corrects.</li> <li>5. Le système enregistre les modifications de spécialité.</li> </ol>
<b>Exception</b>	L'administrateur modifier les champs incorrects. Le système affiche un message erroné.

TABLE 5.6 : Description textuelle du cas d'utilisation Modifier spécialité

### 5.3.3 Diagramme de classe Sprint 1

La figure ci-dessous illustre le diagramme de classe Sprint 1 :

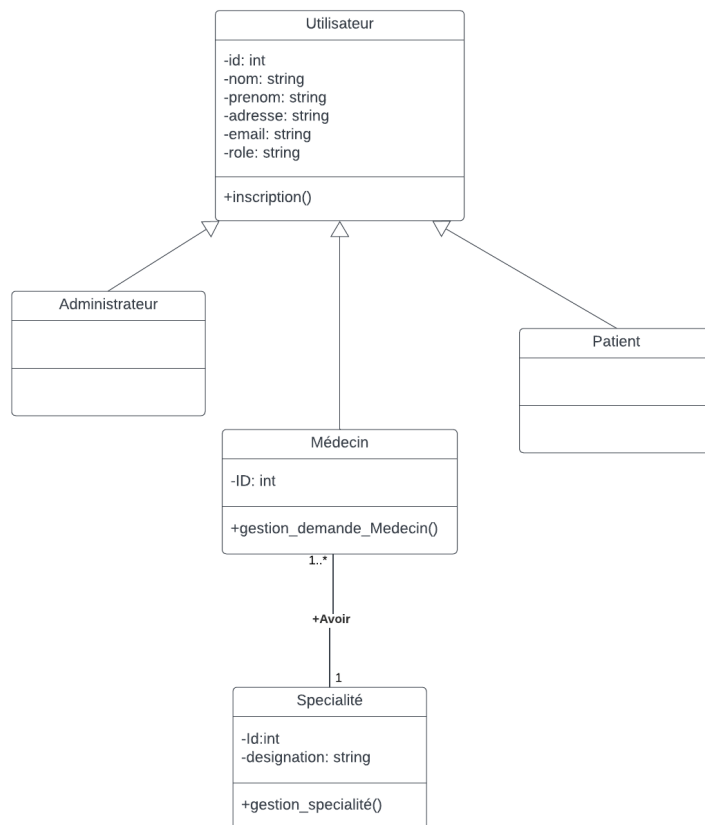


FIGURE 5.1 : Diagramme de classe Sprint 1

### 5.3.4 Réalisation et Test

Cette partie de notre rapport sera essentiellement dédiée à la présentation des interfaces relatives au sprint 1. Nous allons décrire ces interfaces ainsi que leur fonctionnement.

#### 5.3.4.1 Interface « Inscription »

Cette figure présente l'interface d'inscription de patient :

Le patient accède à la page d'inscription dédiée aux patients pour remplir le formulaire d'inscription. Il va saisir les informations nécessaires. Dans le cas d'un champ vide ou les paramètres fournies sont erronés, un message d'erreur sera affiché.

Cette figure présente l'interface d'inscription de Médecin :

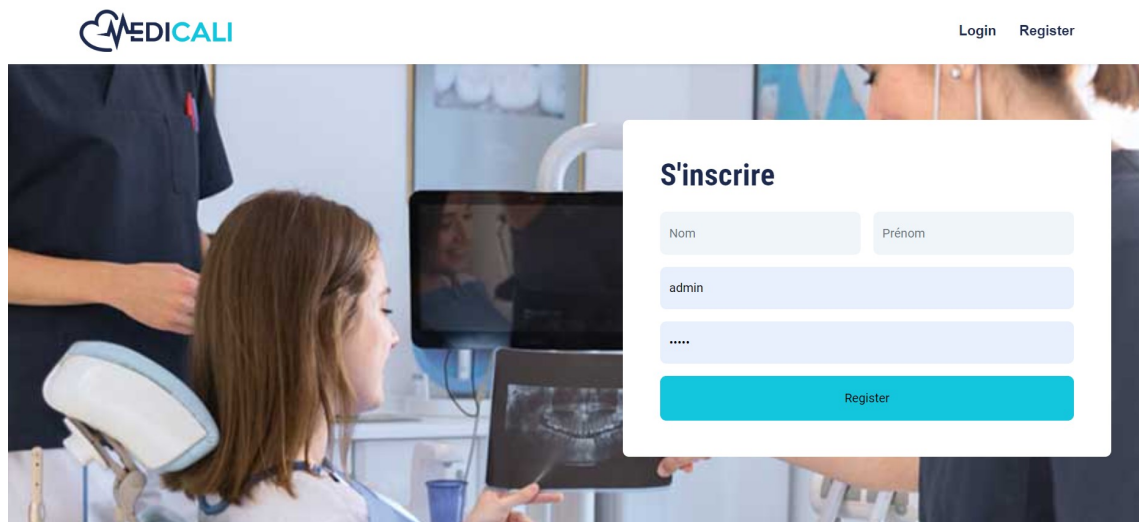


FIGURE 5.2 : Interface Inscription Patient



FIGURE 5.3 : Interface Inscription Médecin

Le médecin accède à la page d'inscription dédiée aux médecins pour remplir le formulaire d'inscription .Il va saisir les informations nécessaires . Dans le cas d'un champ vide ou les paramètres fournies sont erronés, un message d'erreur sera affiché.



---

#### 5.3.4.2 Interface « Authentification »

Cette figure présente l'interface d'authentification :



FIGURE 5.4 : Interface Authentification

L'utilisateur ou l'administrateur accède à la page d'authentification pour se connecter. Il va s'authentifier par la saisie de son login et mot de passe. Dans le cas d'un champ vide ou les paramètres fournies sont erronés, un message d'erreur sera affiché.

#### 5.3.4.3 Interface « Gestion Spécialité »

Cette figure présente l'interface de gestion spécialité :

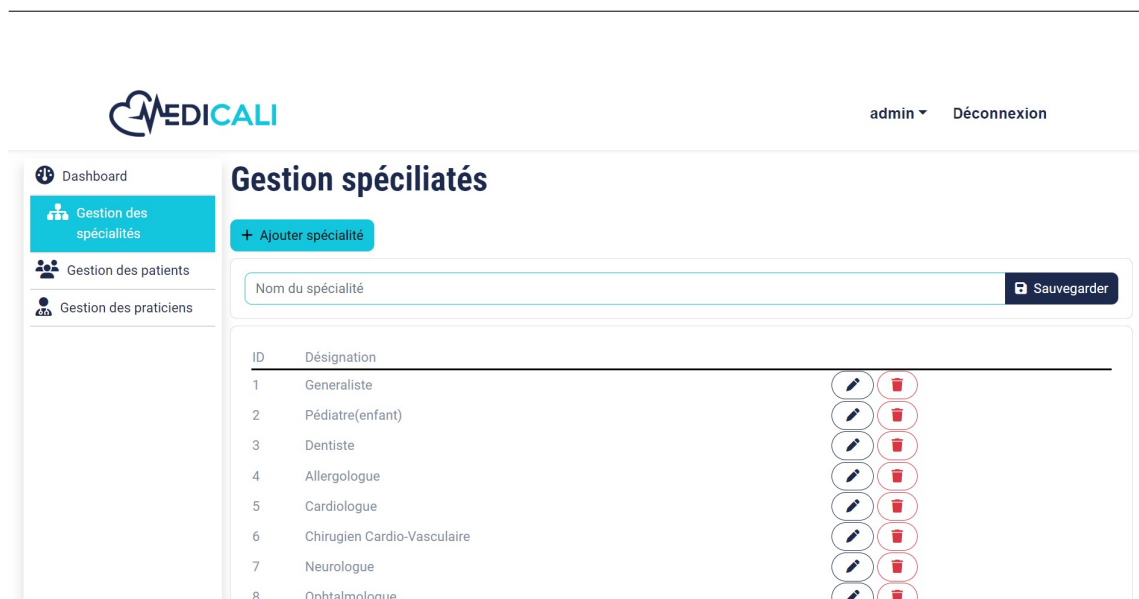


FIGURE 5.5 : Interface Gestion Spécialité

L'interface Spécialité permet à l'administrateur d'ajouter une spécialité ou la supprimer.

## 5.4 Sprint 2

### 5.4.1 Sprint Backlog produit

Dans cette partie, on montre les différentes tâches effectuées pendant ce sprint qui est dédié à la gestion des demandes d'inscription des médecins :

ID :	Fonctionnalités :	User Story :
1	Gestion des demandes d'inscriptions Médecins	En tant que administrateur je peux valider la demande d'inscription ou supprimer la demande d'inscription si le code RPPS de médecin n'est pas valide.

### 5.4.2 Descriptions textuelles des cas d'utilisation

Au cours de cette partie, nous présentons les tableaux des descriptions textuelles des cas d'utilisation de Sprint 2.

---

#### 5.4.2.1 Description textuelle de cas d'utilisation « Gestion des demandes d'inscriptions Médecins » :

Les tableaux ci-dessous illustrent les descriptions textuelles de cas d'utilisation Gestion des demandes d'inscriptions Médecins :

<b>Cas d'utilisation</b>	valider l'inscription de Médecins
<b>Acteurs</b>	administrateur
<b>Pré Condition</b>	L'administrateur doit valider l'inscription de Médecins.
<b>Scénario Nominal</b>	L'administrateur valider un Médecin. Le système enregistre le nouveau Médecin.
<b>Exception</b>	Le système affiche un message erroné.

TABLE 5.7 : Description textuelle de cas d'utilisation Valider Médecin

<b>Cas d'utilisation</b>	Supprimer Médecin
<b>Acteurs</b>	administrateur
<b>Pré Condition</b>	L'administrateur doit supprimer un médecin.
<b>Scénario Nominal</b>	L'administrateur demande supprimer un médecin.
<b>Exception</b>	Le système affiche un message erroné.

TABLE 5.8 : Description textuelle de cas d'utilisation Supprimer Médecin

#### 5.4.3 Diagramme de classe Sprint 2

La figure ci-dessous illustre le diagramme de classe Sprint 2 :

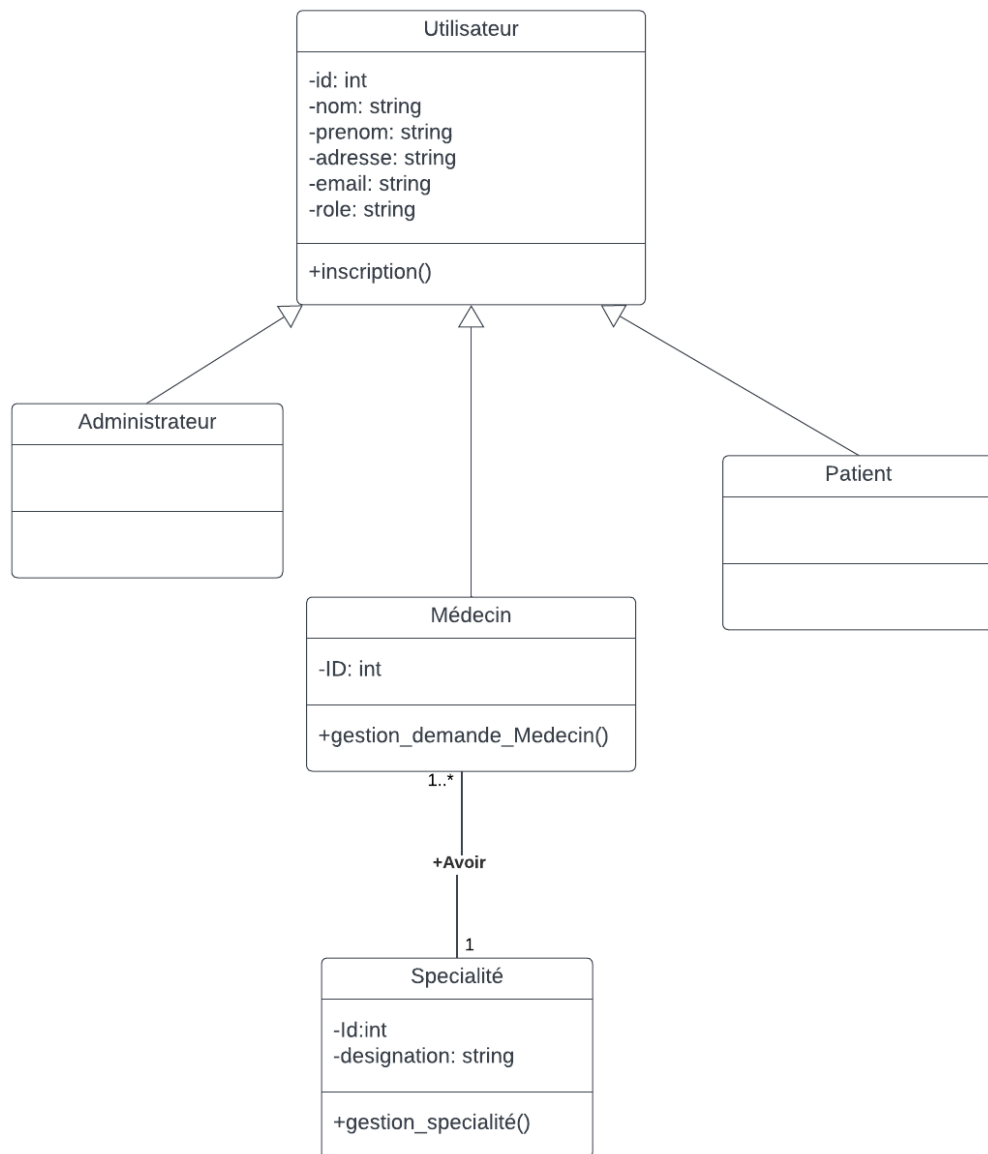


FIGURE 5.6 : Diagramme de classe Sprint 2

#### 5.4.4 Réalisation et Test

Cette partie de notre rapport sera essentiellement dédiée à la présentation des interfaces relatives au sprint 2. Nous allons décrire ces interfaces ainsi que leur fonctionnement.

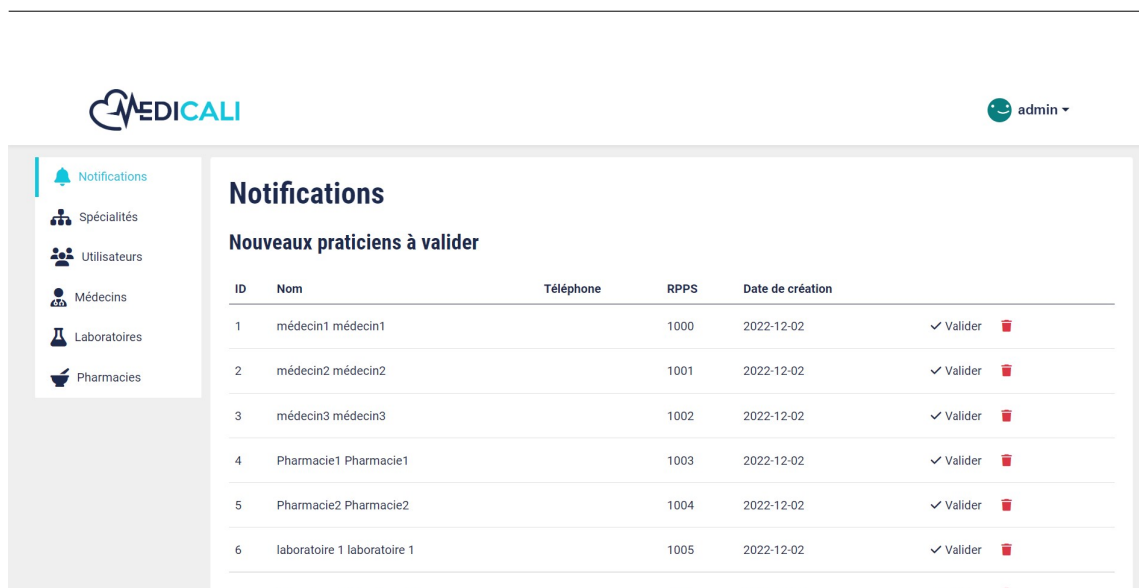


FIGURE 5.7 : Interface Gestion Notifications

L'interface Notifications permet à l'administrateur de valider ou de supprimer les demandes d'inscriptions des différents médecins.

## 5.5 Conclusion

A la fin de ce release nous avons pu réaliser les fonctionnalités relatives à l'inscription des patients et des médecins, l'authentification et la gestion des spécialités et des demandes d'inscriptions des médecins.

# Chapitre 6

## Release 2

### 6.1 Introduction

Pour mieux cerner la réalisation du release 1, nous l'avons réparti sur 2 sprints en fonction du contenu du Backlog qui représente l'ensemble des tâches à accomplir dans un release. Nous allons procéder de la même manière pour ce release.

### 6.2 Organisation des sprints

Le release 2 est composé de deux sprints :

- **Sprint 3** : Ce sprint permet d'effectuer la gestion de dossier médical, gestion des analyses maladies et mesures.
- **Sprint 4** : Ce sprint permet d'effectuer la gestion des rendez-vous

Pareil pour cet release, nous allons détailler pour chaque sprint son Backlog, sa vue dynamique : son diagramme de cas d'utilisation ainsi que les diagrammes de séquences associés à chaque cas d'utilisation.

### 6.3 Sprint 3

#### 6.3.1 Sprint Backlog produit

Dans cette partie, on montre les différentes tâches effectuées pendant ce sprint qui est dédié à la gestion de dossier médical, gestion des analyses maladies et mesures.

---

ID :	Fonctionnalités :	User Story :
1	Gestion Analyse Médical	En tant que patient je veux ajouter un dossier médical en ajoutant des analyses, des maladies et des mesures.
2	Gestion Analyses	En tant que patient je veux ajouter et supprimer des analyses.
3	Gestion Maladies	En tant que patient je veux ajouter et supprimer des Maladies.
4	Gestion Mesures	En tant que patient je veux ajouter et supprimer des Mesures.

TABLE 6.1 : Tableau du Backlog produit de Sprint 3

### 6.3.2 Descriptions textuelles des cas d'utilisations

Au cours de cette partie, nous présentons les tableaux des descriptions textuelles des cas d'utilisation de Sprint 3.

#### 6.3.2.1 Description textuelle de cas d'utilisation « Gestion Analyses » :

Les tableaux ci-dessous illustrent les descriptions textuelles de cas d'utilisation Gestion Analyses :

<b>Cas d'utilisation</b>	Ajouter analyse
<b>Acteurs</b>	Patient
<b>Pré Condition</b>	Un patient doit ajouter des analyses.
<b>Scénario Nominal</b>	1.Un patient demande d'ajouter une analyse a son dossier médicale. 2.Le système affiche l'interface d'ajouter analyse. 3.Un patient remplit les champs de formulaire puis il valide. 4.Le système vérifie les champs corrects. 5.Le système enregistre le nouveau analyse.
<b>Exception</b>	Patient saisie les informations incorrect. Le système affiche un message erroné.

TABLE 6.2 : Description textuelle du cas d'utilisation Ajouter analyse

<b>Cas d'utilisation</b>	Supprimer analyse
<b>Acteurs</b>	Patient
<b>Pré Condition</b>	Un patient doit supprimer des analyses.
<b>Scénario Nominal</b>	Patient demande supprimer un analyse.
<b>Exception</b>	Patient supprime les champs incorrects. Le système affiche un message erroné.

TABLE 6.3 : Description textuelle du cas d'utilisation Supprimer analyse



---

### 6.3.2.2 Description textuelle de cas d'utilisation « Gestion Maladies » :

Les tableaux ci-dessous illustrent les descriptions textuelles de cas d'utilisation Gestion Maladies :

<b>Cas d'utilisation</b>	Ajouter maladie
<b>Acteurs</b>	Patient
<b>Pré Condition</b>	Un patient doit ajouter des maladies.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"><li>1. Un patient demande d'ajouter une maladie a son dossier médicale.</li><li>2. Le système affiche l'interface d'ajouter maladie.</li><li>3. Un patient remplit les champs de formulaire puis il valide.</li><li>4. Le système vérifie les champs corrects.</li><li>5. Le système enregistre le nouveau maladie.</li></ol>
<b>Exception</b>	Patient saisie les informations incorrect. Le système affiche un message erroné.

TABLE 6.4 : Description textuelle du cas d'utilisation Ajouter maladie

<b>Cas d'utilisation</b>	Supprimer maladie
<b>Acteurs</b>	Patient
<b>Pré Condition</b>	Un patient doit supprimer des maladies.
<b>Scénario Nominal</b>	Un patient demande supprimer un maladie.
<b>Exception</b>	Patient supprime les champs incorrects. Le système affiche un message erroné.

TABLE 6.5 : Description textuelle du cas d'utilisation Supprimer maladie

---

### 6.3.2.3 Description textuelle de cas d'utilisation « Gestion Mesures » :

Les tableaux ci-dessous illustrent les descriptions textuelles de cas d'utilisation Gestion Mesures :

<b>Cas d'utilisation</b>	Ajouter mesure
<b>Acteurs</b>	Patient
<b>Pré Condition</b>	Un patient doit ajouter des mesures.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"><li>1. Un patient demande d'ajouter une mesure a son dossier médicale.</li><li>2. Le système affiche l'interface d'ajouter une mesure.</li><li>3. Un patient remplit les champs de formulaire puis il valide.</li><li>4. Le système vérifie les champs corrects.</li><li>5. Le système enregistre le nouveau mesure.</li></ol>
<b>Exception</b>	Patient saisie les informations incorrect. Le système affiche un message erroné.

TABLE 6.6 : Description textuelle du cas d'utilisation Ajouter mesure

<b>Cas d'utilisation</b>	Supprimer mesure
<b>Acteurs</b>	Patient
<b>Pré Condition</b>	Un patient doit supprimer des mesures.
<b>Scénario Nominal</b>	Un patient demande supprimer une mesure.
<b>Exception</b>	Patient supprime les champs incorrects. Le système affiche un message erroné.

TABLE 6.7 : Description textuelle du cas d'utilisation Supprimer mesure

### 6.3.3 Diagramme de classe Sprint 3

La figure ci-dessous illustre le diagramme de classe Sprint 3 :

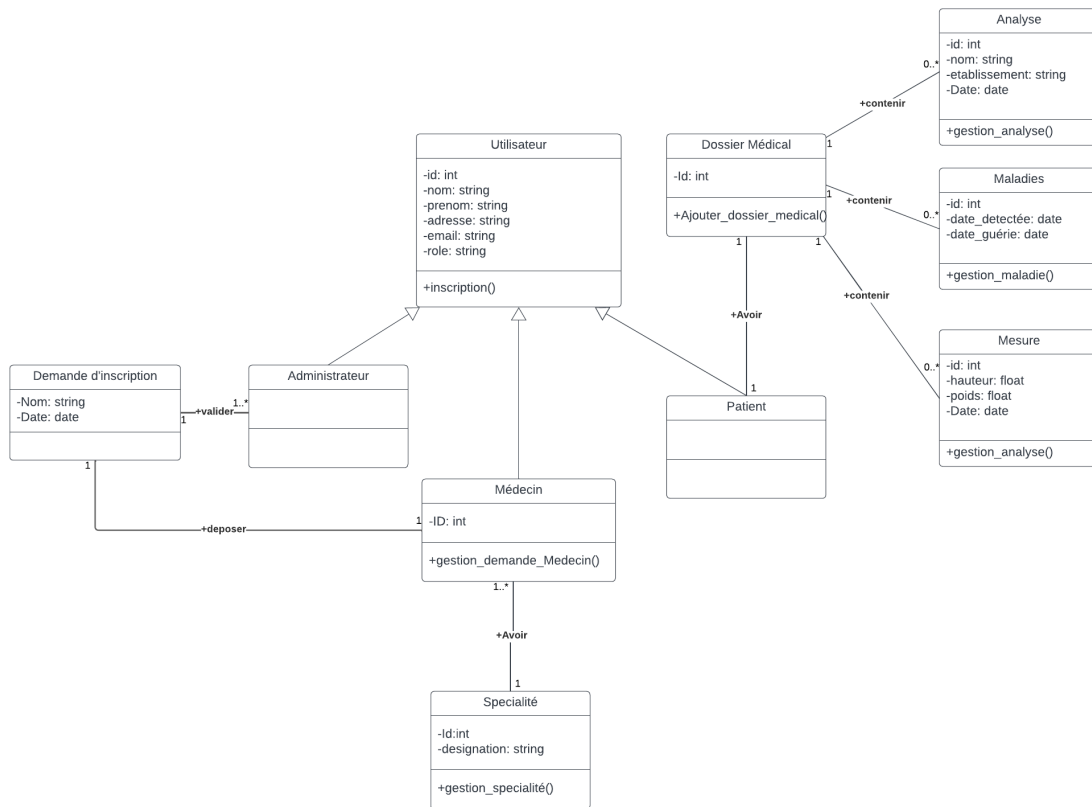


FIGURE 6.1 : Diagramme de classe Sprint 3

### 6.3.4 Réalisation et Test

Cette partie de notre rapport sera essentiellement dédiée à la présentation des interfaces relatives au sprint 3. Nous allons décrire ces interfaces ainsi que leur fonctionnement.

#### 6.3.4.1 Interface « Analyse »

Cette figure présente l'interface Ajout une analyse :

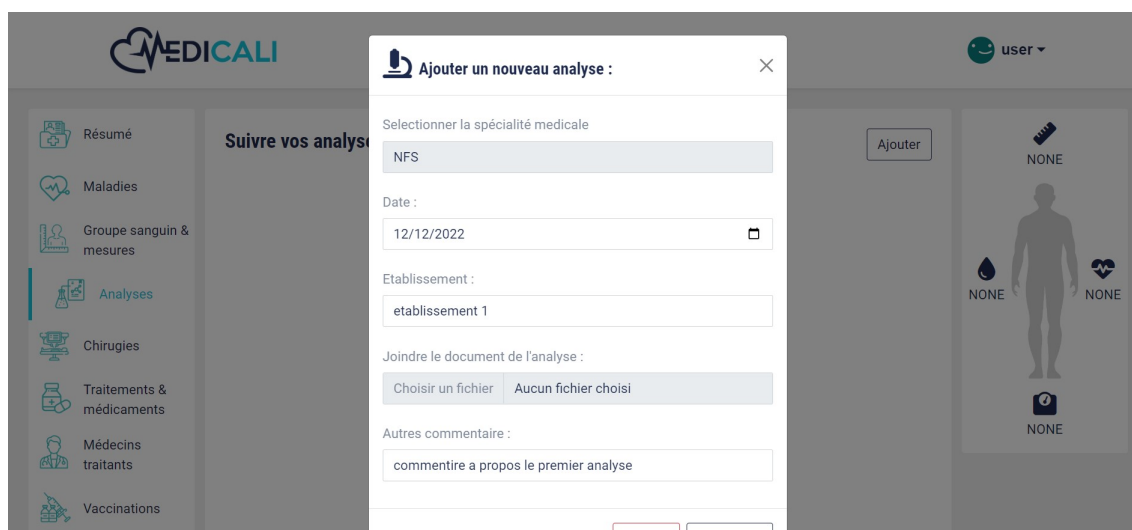


FIGURE 6.2 : Interface Analyse

#### 6.3.4.2 Interface « Maladie »

Cette figure présente l'interface Ajout une maladie

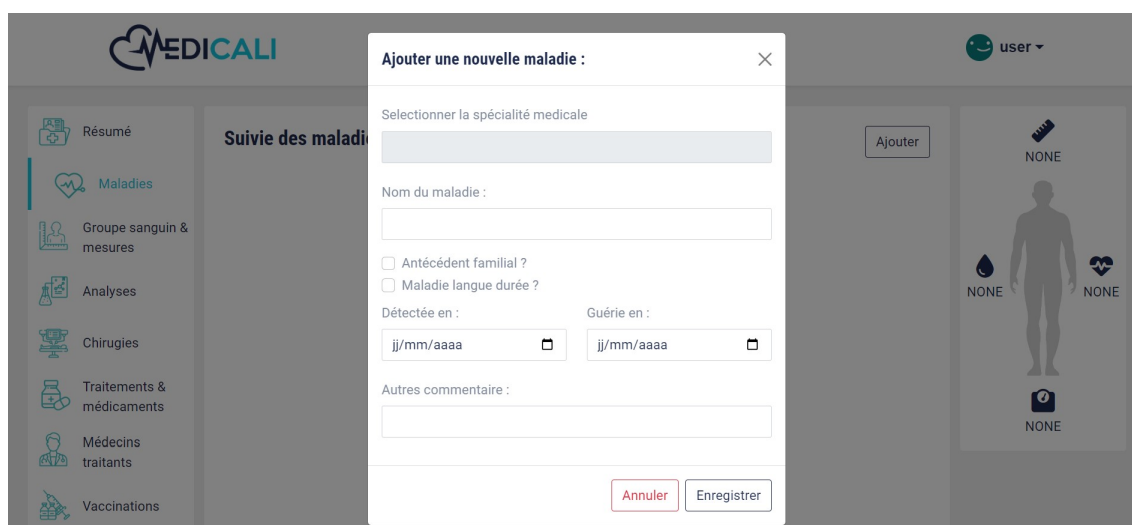


FIGURE 6.3 : Interface Maladie

### 6.3.4.3 Interface « Mesure »

Cette figure présente l'interface Ajout les mesures.

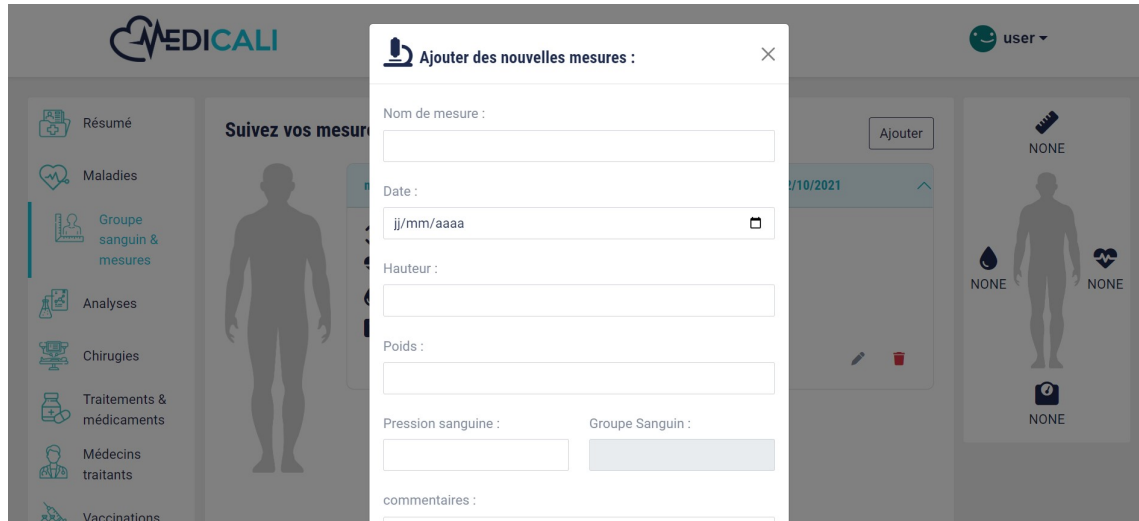


FIGURE 6.4 : Interface Mesure

Le patient accède à la page dossier médical il peut remplir les formulaires d'ajout analyse mesure et maladie. Il va saisir les informations nécessaires. Dans le cas d'un champ vide ou les paramètres fournies sont erronés, un message d'erreur sera affiché.

## 6.4 Sprint 4

### 6.4.1 Sprint Backlog produit

Dans cette partie, on montre les différentes tâches effectuées pendant ce sprint qui est dédié à la gestion des rendez-vous.

ID :	Fonctionnalités :	User Story :
1	Prendre rendez-vous	En tant que patient je veux ajouter un rendez-vous.
2	Gestion rendez-vous	En tant que médecin je veux valider ou supprimer un rendez-vous.

TABLE 6.8 : Tableau du Backlog produit de Sprint 4

---

## 6.4.2 Descriptions textuelles des cas d'utilisations

Au cours de cette partie, nous présentons les tableaux des descriptions textuelles des cas d'utilisation de Sprint 4.

### 6.4.2.1 Description textuelle de cas d'utilisation « Prendre Rendez-vous » :

Le tableaux ci-dessous illustrent les descriptions textuelles de cas d'utilisation Prendre Rendez-vous :

<b>Cas d'utilisation</b>	Ajouter rendez-vous
<b>Acteurs</b>	Patient
<b>Pré Condition</b>	Un patient doit ajouter des rendez-vous.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"><li>1. Un patient demande de prendre d'ajouter un rendez-vous.</li><li>2. Le système affiche l'interface d'ajouter un rendez-vous.</li><li>3. Un patient remplit les champs de formulaire puis il valide.</li><li>4. Le système vérifie les champs corrects.</li><li>5. Le système enregistre le nouveau rendez-vous.</li></ol>
<b>Exception</b>	Patient saisie les informations incorrect. Le système affiche un message erroné.

TABLE 6.9 : Description textuelle du cas d'utilisation Ajouter rendez-vous

### 6.4.2.2 Description textuelle de cas d'utilisation « Gestion Rendez-vous » :

Les tableaux ci-dessous illustrent les descriptions textuelles de cas d'utilisation Gestion Rendez-vous :

---

<b>Cas d'utilisation</b>	valider un rendez-vous
<b>Acteurs</b>	médecin
<b>Pré Condition</b>	un médecin doit valider un rendez-vous.
<b>Scénario Nominal</b>	Un médecin valider un rendez-vous. Le système enregistre le nouveau rendez-vous.
<b>Exception</b>	Le système affiche un message erroné.

TABLE 6.10 : Description textuelle du cas d'utilisation valider rendez-vous

<b>Cas d'utilisation</b>	Supprimer rendez-vous
<b>Acteurs</b>	médecin
<b>Pré Condition</b>	un médecin doit supprimer un rendez-vous.
<b>Scénario Nominal</b>	un médecin demande supprimer un rendez-vous.
<b>Exception</b>	Le système affiche un message erroné.

TABLE 6.11 : Description textuelle du cas d'utilisation Supprimer rendez-vous

### 6.4.3 Diagramme de classe Sprint 4

La figure ci-dessous illustre le diagramme de classe Sprint 4 :

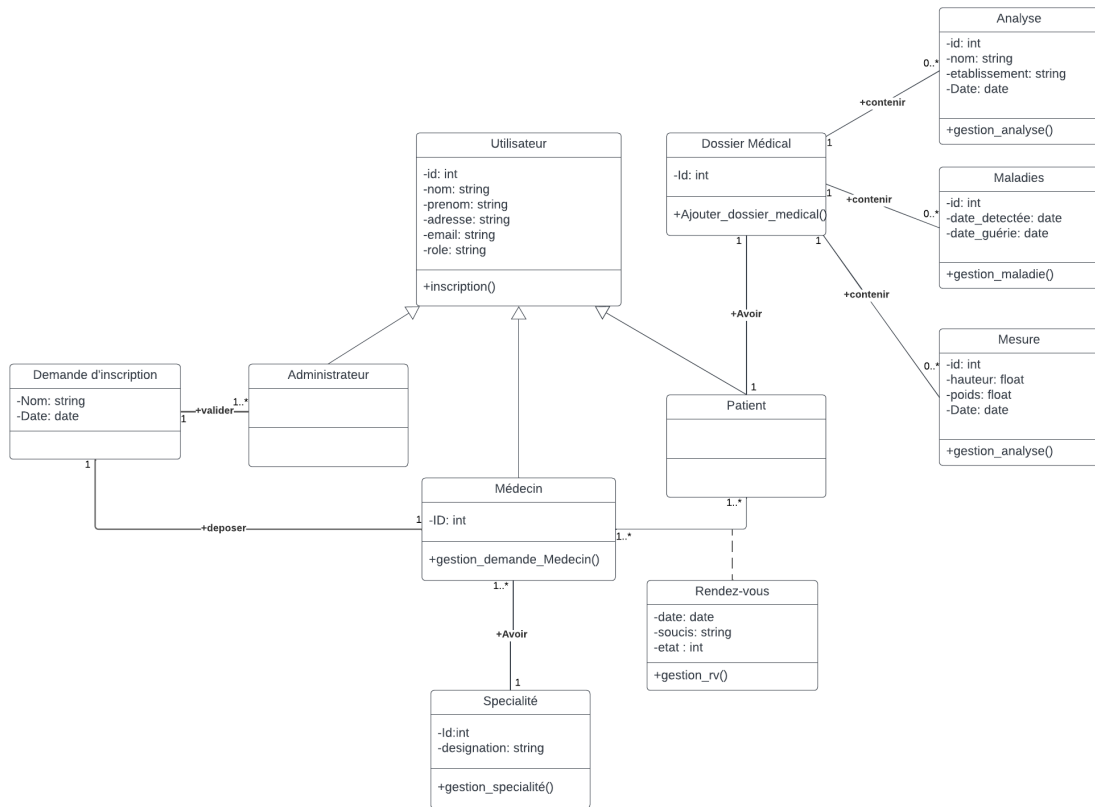


FIGURE 6.5 : Diagramme de classe Sprint 4

## 6.4.4 Réalisation et Test

Cette partie de notre rapport sera essentiellement dédiée à la présentation des interfaces relatives au sprint 4. Nous allons décrire ces interfaces ainsi que leur fonctionnement.

### 6.4.4.1 Interface « Prendre rendez-vous »

Cette figure présente l'interface Ajout rendez-vous.





FIGURE 6.6 : Interface Ajout rendez-vous

Le patient accède à la page de médecin en question et clique sur prendre rendez-vous il peut remplir les formulaires d'ajout. Il va saisir les informations nécessaires. Dans le cas d'un champ vide ou les paramètres fournies sont erronés, un message d'erreur sera affiché.

#### 6.4.4.2 Interface « Gestion rendez-vous »

Cette figure présente l'interface gestion rendez-vous.

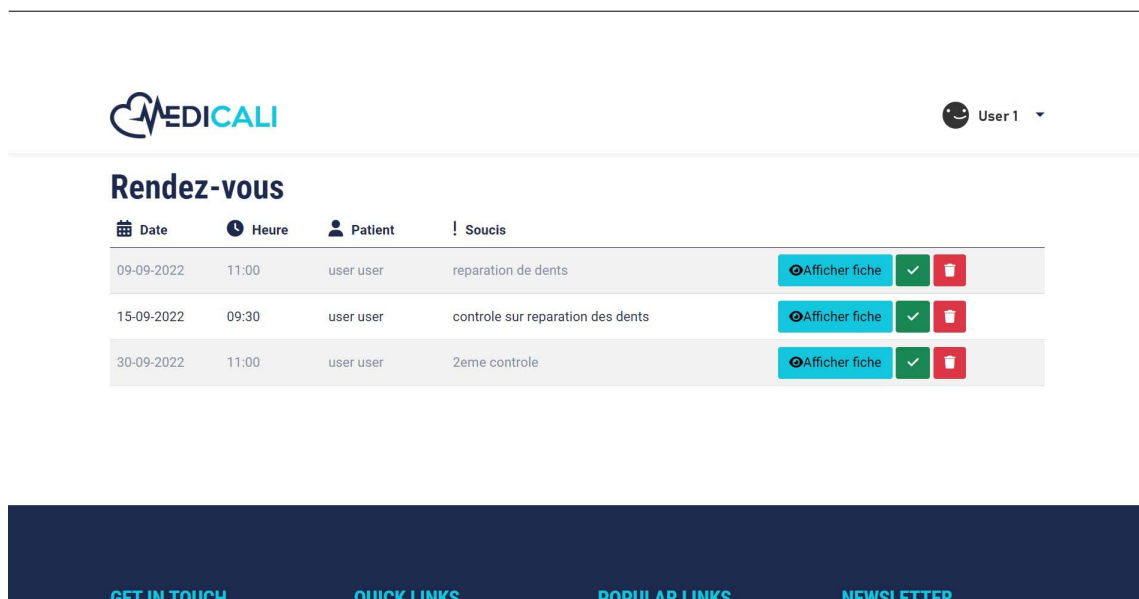


FIGURE 6.7 : Interface Gestion rendez-vous

Le médecin accède à la liste des rendez-vous et clique sur valider ou supprimer le rendez-vous.

## 6.5 Conclusion

A la fin de ce release nous avons pu réaliser les fonctionnalités relatives à la gestion de dossier médicale et gestion des rendez-vous.

# Conclusion Générale

En conclusion, ce projet au sein de la société **Talan Tunisie** était une excellente opportunité pour la mise en œuvre des acquis théoriques que nous avons appris tout au long de notre cursus universitaire, de les enrichir et d'approfondir nos connaissances en informatique.

Cet opportunité nous a été une expérience enrichissante qui nous'a permis de comprendre les enjeux d'un projet de recherche et développement, d'avoir la chance d'être responsable de travail.

Tout au long de ce rapport nous avons détaillé les différentes étapes que nous avons adoptées pour construire une plateforme médicale en suivant la méthodologie Scrum.

Nous avons commencé notre rapport par le cadre général de projet. Ensuite, nous avons effectué une planification du travail en identifiant les différents modules de notre projet. Après, nous avons déterminé les études spécifiques et conceptuelles de chaque sprint et nous avons terminé notre rapport par la description de l'environnement de travail matériel et logiciel, les outils utiliser, les technologies et l'architecture de notre travail.

Dans le future proche, nous visons améliorer notre application en ajoutons un espace de chat entre les différents acteurs et les membres de l'application ainsi d'intégrer un BI (Business Intelligence) à notre application.