

# Table des matières

<b>Dédicace</b>	<b>5</b>
<b>Remerciements</b>	<b>7</b>
<b>Introduction</b>	<b>9</b>
<b>1 Présentation générale</b>	<b>11</b>
Introduction . . . . .	12
1.1 Présentation de l'organisme d'accueil . . . . .	12
1.2 Les solutions de PlanetConectUS . . . . .	13
1.3 Étude de l'existant . . . . .	13
1.4 Critiques de l'existant . . . . .	15
1.5 Solution proposée . . . . .	15
1.6 Méthodologie de conception . . . . .	15
Conclusion . . . . .	16
<b>2 Analyse des besoins et Conception</b>	<b>17</b>
Introduction . . . . .	18
2.1 Identification des besoins . . . . .	18
2.1.1 Les besoins fonctionnels . . . . .	18
2.1.2 Les besoins non fonctionnels . . . . .	18
2.2 Identification des acteurs . . . . .	19
2.3 Diagramme de cas d'utilisation . . . . .	19
2.3.1 Diagramme du cas d'utilisation général . . . . .	20
2.4 Diagramme de classes . . . . .	22
2.5 Planning du traitement des cas d'utilisation . . . . .	24
2.6 Priorité . . . . .	24
2.7 Le backlog du produit . . . . .	24
Conclusion . . . . .	24
<b>3 Sprint 0 : Environnement de travail</b>	<b>25</b>
Introduction . . . . .	26
3.1 Environnement de développement . . . . .	26
3.1.1 Environnement matériel . . . . .	26
3.1.2 Environnement logiciel . . . . .	26
3.2 Architecture . . . . .	30
3.2.1 Architecture logique . . . . .	30
3.2.2 Architecture physique . . . . .	31
Conclusion . . . . .	31

<b>4 Sprint 1 : Chercher un médecin</b>	<b>32</b>
Introduction . . . . .	33
4.1 Backlog du sprint 1 . . . . .	33
4.2 Raffinement du Cas d'utilisation « Chercher un médecin » . . . . .	33
4.3 Description textuelle du cas d'utilisation chercher un médecin par son nom , prénom ou par sa spécialité . . . . .	34
4.4 Diagrammes de séquence . . . . .	34
4.5 Diagrammes de séquence détaillés . . . . .	35
4.6 Réalisation . . . . .	36
Conclusion . . . . .	41
<b>5 Sprint 2 : Prise de rendez vous</b>	<b>42</b>
Introduction . . . . .	43
5.1 Backlog du sprint 2 . . . . .	43
5.2 Raffinement du Cas d'utilisation « Prise de rendez-vous » . . . . .	43
5.3 Description textuelle du cas d'utilisation Prise de rendez-vous . . . . .	44
5.4 Diagrammes de séquence détaillés . . . . .	44
5.5 Réalisation . . . . .	46
Conclusion . . . . .	48
<b>6 Sprint 3 : Le modèle machine Learning</b>	<b>49</b>
Introduction . . . . .	50
6.1 Technologie IA . . . . .	50
6.2 Acquisition des données . . . . .	51
6.3 Prétraitement des données . . . . .	51
6.4 Description des données . . . . .	51
6.5 Représentation des données . . . . .	52
6.6 Modélisation . . . . .	52
6.7 Construction des modèles . . . . .	53
6.8 Evaluation . . . . .	53
6.9 Techniques d'évaluation . . . . .	53
6.10 Evaluation des modèles . . . . .	54
Conclusion . . . . .	55

# Table des figures

1.1	Logo de la société PlanetConectUs . . . . .	12
1.2	Imprime Ecran de l'interface d'accueil de Doctena . . . . .	13
1.3	Imprime Ecran de l'interface d'accueil de HealthPlix SPOT . . . . .	14
1.4	Imprime Ecran de l'interface d'accueil de Mfine . . . . .	14
1.5	La methodologies agile . . . . .	16
2.1	Diagramme de cas d'utilisation globale . . . . .	21
2.2	Diagramme de classes . . . . .	23
3.1	Logo Visual Studio Code . . . . .	26
3.2	Logo MongoDB . . . . .	27
3.3	Logo Colab . . . . .	27
3.4	Logo Postman . . . . .	27
3.5	Logo GitHub . . . . .	27
3.6	Logo Flutter . . . . .	28
3.7	Logo Python . . . . .	28
3.8	Logo Node js . . . . .	28
3.9	Logo Flask . . . . .	29
3.10	Logo Scrapy . . . . .	29
3.11	Logo Socket.IO . . . . .	29
3.12	Logo Visual paradigm online . . . . .	29
3.13	Logo Overleaf . . . . .	30
3.14	Architecture MVVM . . . . .	31
3.15	Architecture physique . . . . .	31
4.1	Diagramme de cas d'utilisation "Chercher un médecin" . . . . .	33
4.2	Diagrammes de séquence chercher un médecin . . . . .	35
4.3	Recherche par nom . . . . .	36
4.4	Recherche par spécialité "Psychiatry" . . . . .	36
4.5	Recherche par spécialité "Therapist" . . . . .	37
4.6	Recherche par spécialité "Rheumatologist" . . . . .	37
4.7	Recherche par AI . . . . .	38
4.8	Recherche par AI "La description du patient" . . . . .	38
4.9	Recherche par AI résultat "Psychiatry" . . . . .	39
4.10	Recherche par AI "La description du patient" . . . . .	39
4.11	Recherche par AI résultat "Rheumatologist" . . . . .	40
4.12	Recherche par AI "La description du patient" . . . . .	40
4.13	Recherche par AI résultat "Gynecologist" . . . . .	41

5.1	Diagramme de cas d'utilisation "Prise de rendez-vous"	43
5.2	Diagrammes de séquence Prise de rendez-vous	45
5.3	Liste des médecins	46
5.4	Profil de médecin sélectionné par le patient	46
5.5	le calendrier du médecin	47
5.6	Le patient choisit une date	47
5.7	Prise de rendez-vous	48
6.1	Organisation de l'intelligence artificielle	50
6.2	Information générale des données selon la spécialité	52
6.3	Information générale des données	52
6.4	Structure de la matrice de confusion	54
6.5	Matrice de confusion du modèle régression logistique	55

# Dédicace

À vec tout respect et amour je dédie ce modeste travail :

À Dieu le tout Miséricordieux, ton Amour, ta Miséricorde et tes Grâces à mon endroit m'ont fortifiée dans la persévérance et l'ardeur au travail.

À ma mère, la sources de tendresse et d'amours pour son soutien tout au long de ma vie scolaire.

À ma soeur,  
ma famille,  
mes chers amis,  
mes enseignants,

et à tous ceux qui ont collaboré de près ou de loin à l'élaboration de ce travail.

Que dieu leur accorde santé et prospérité.

HASSEN

# Remerciements

C'est avec un grand plaisir que nous réservons cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui nous ont aidés à la réalisation de ce travail.

Je tiens à exprimer mes vifs remerciements à monsieur **Mohamed Ali CHaabani**, mon encadrant de l'entreprise pour l'aide et le soutien qu'il m'a accordé tout au long de ce stage, à toute l'équipe de département informatique de m'avoir accueilli au sein de leur équipe et de m'avoir encadré et orienté durant mon PFE.

Je remercie également Mme.**Mariem BOUZOUITA**, professeur à ESPRIT, pour son encadrement, sa disponibilité et ses conseils fructueux qu'elle m'a prodigué le long de notre projet.

Mes remerciements s'adressent aussi à ABIR KACHOURI, pour l'aide et le soutien qu'elle m'a accordé tout au long de ce stage.

Enfin, j'exprime mes remerciements les plus dévoués aux membres de jury qui ont eu l'amabilité de lire et d'évaluer ce modeste travail.

# Introduction

L'informatique est devenue un outil performant dans presque tous les domaines d'activités humaines et tout le monde affirme que de nombreux champs d'application vont encore s'ouvrir. De plus, cet outil va évoluer considérablement au niveau des performances et services.

La technologie de l'information et de la communication a été intégrée dans le système éducatif tunisien pour deux raisons, d'une part, pour diffuser la culture numérique auprès des jeunes de la société et, d'autre part, de faciliter et d'améliorer la gestion d'un parc informatique. Cela a pour but d'aider à mémoriser les données concernant les logiciels et les matériels et de rendre la complexité d'administration simple et efficace.

C'est dans ce contexte que notre projet se présente. La société «PlanetConnectus» nous a proposée de concevoir et de développer une application mobile qui permet de faciliter et de gérer la classification des maladies des patients selon le texte de description des symptômes.

Notre but essentiel étant, d'une part de remédier aux problèmes de complexités et de perte du temps dans la gestion du matériel et d'améliorer la qualité de service, et d'autre part d'expérimenter certains outils techniques pour le développement de cette application.

Pour la modélisation de notre projet, nous adoptons la Méthodologie Agile comme méthode de conception qui se veut plus pragmatique que les méthodes traditionnelles.

Ce mémoire s'articule autour de six chapitres :

- Dans le premier chapitre intitulé « Présentation générale », nous présenterons le cadre du travail, une critique de l'existant, le sujet qui nous a été proposé et l'identification des besoins
- Dans le deuxième chapitre intitulé « Analyse des besoins et Conception », deuxième chapitre de notre travail, nous détaillerons les principaux cas d'utilisations de notre système et modélisation de l'application au moyen des différents diagrammes offerts par UML en présentant un raffinement des besoins déjà spécifiés.

Le reste de ce chapitre sera consacrés à la priorité du backlog produit.

Les quatre chapitres restants seront consacrés à la réalisation de nos sprints,

- Sprint 0 seront consacrés à la préparation de l'environnement de travail, à la compréhension du sujet,
- Le Sprint 1 sera dédié au développement du volet : chercher un médecin,
- Le Sprint 2 sera dédié au développement du volet : Prise de rendez-vous,
- Le Sprint 3 sera dédié au modèle machine Learning capable de détecter la maladie à partir des symptômes décrits par les patients.

Nous conclurons notre rapport par une conclusion dans laquelle nous résumerons nos travaux et discuterons de certaines orientations futures.

# Chapitre 1

## Présentation générale

## Introduction

Dans ce premier chapitre, nous allons définir le cadre général du projet. Tout d'abord nous allons présenter l'organisme d'accueil, nous dériverons une idée sur ses missions ainsi que les services qu'il offre.

Nous fournirons un organigramme qui nous aidera à schématiser les différents constituants de la compagnie avec ses différents départements et services et on va s'intéresser à la présentation du département informatique. Ensuite nous allons définir le système existant, dégager ses problèmes et essayer de proposer une solution.

### 1.1 Présentation de l'organisme d'accueil

PlanetConectUs est la société qui nous a accueilli pour effectuer notre stage PFE.

C'est l'un des principaux fournisseurs de solutions IOT (Internet Of Things).

Le développement d'applications IOT permet aux entreprises de combler le fossé entre le monde réel et le numérique grâce à une toute nouvelle expérience.

Les services de l'Internet des objets permettent aux entreprises de réussir en transformant leurs processus métier, leur infrastructure réseau et leurs modèles opérationnels.

Les équipes d'experts de PlanetConectUs ont travaillé sur de nombreux projets liés à l'IOT, en créant des solutions Internet des objets de bout en bout.

L'ensemble des services proposés par PlanetConectUs comprend la consultation, le développement et l'intégration de services IOT.

Elle a aussi ses propres solutions qui sont citées ci-dessous(1.2 Les solutions de PlanetConectUS).



FIGURE 1.1 – Logo de la société PlanetConectUs

## 1.2 Les solutions de PlanetConectUS

- Indoor Positioning system : La manière la plus simple de décrire les systèmes de positionnement en intérieur (IPS) est de les considérer comme un GPS pour les environnements intérieurs. IPS peut être utilisé pour localiser des personnes ou des objets à l'intérieur des bâtiments, généralement via un appareil mobile tel qu'un téléphone intelligent ou une tablette. Bien que la technologie soit plus récente que le GPS, les services qui tirent parti de l'IPS gagnent rapidement du terrain dans des endroits tels que les centres commerciaux, les hôpitaux, les aéroports et autres lieux intérieurs où la navigation et d'autres services de localisation (LBS) peuvent s'avérer indispensables.
- PayGate : C'est une application mobile qui transforme votre téléphone portable en clé pour ouvrir tous les portails compatibles. Il suffit de télécharger l'application et d'activer le Bluetooth et lorsque le client passe à une vitesse qui peut aller jusqu'à 30 km / h, le système détecte automatiquement le mobile qui peut être n'importe où (dans le véhicule, dans la poche du conducteur, etc) puis ouvre directement le portail en moins d'une seconde et l'argent est déduit du compte de la personne.
- Pikit : Cette solution a été développée pour Carrefour. Elle permet de scanner les produits et les ajouter au panier. Switch : est une application mobile et web, elle a été développée pour Switch albabtain.
- Dukkan el hai : Est une application mobile qui met en relation les utilisateurs qui ont besoin de consommables en épices avec l'épicerie la plus proche du quartier. Elle a été développée pour shomoos.
- Sanaadi : Est une plateforme qui regroupe des milliers de spécialistes dans différents domaines pour fournir des services sociaux, éducatifs et de divertissement. Il cible presque tous les groupes de la communauté, comme les personnes âgées, les personnes ayant des besoins spéciaux, les jeunes et les enfants. Elle a été développée pour shomoos.

## 1.3 Étude de l'existant

Cette partie présente une étude sur le système existant à partir des enquêtes menées avec les responsables des différentes directions qui interagissent avec ce système.

Il existe plusieurs solutions pour la gestion des prises de rendez-vous dans les établissements de santé. Il s'agit notamment de :

- Doctena qui est une plateforme dédiée à aider les patients à trouver des médecins et à planifier leur prochaine visite médicale. Elle utilise le mieux les dernières techniques en ligne et mobiles afin de réduire le temps d'attente des patients pour consulter un médecin.

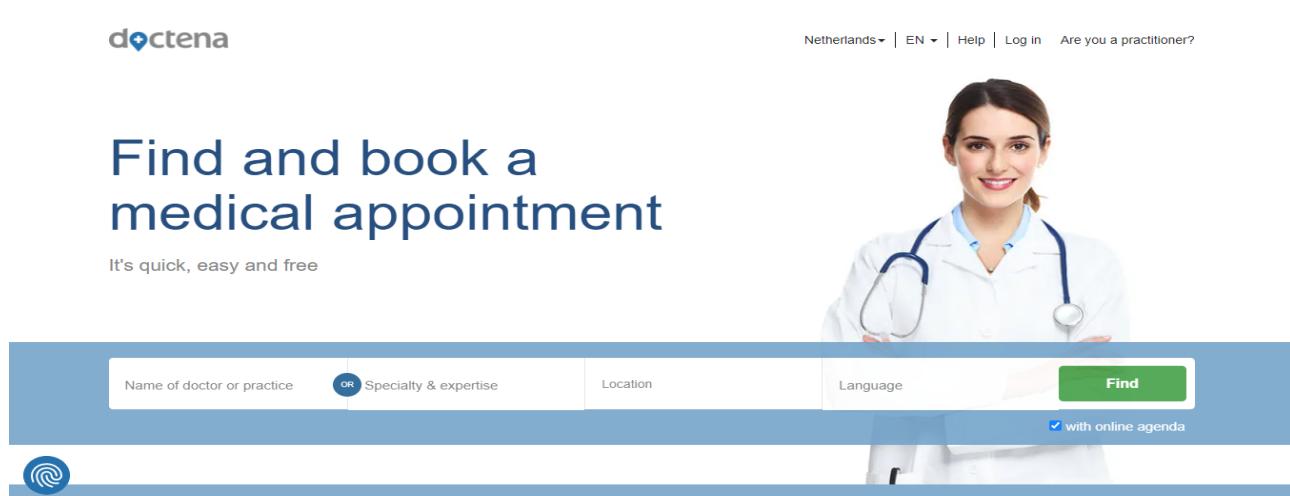


FIGURE 1.2 – Imprime Ecran de l'interface d'accueil de Doctena

- HealthPlix SPOT a été le pionnier de la numérisation des soins de santé grâce à sa plate-forme de DME alimentée par l'IA et prête pour l'avenir pour les médecins en Inde. La start-up permet actuellement à plus de 10 000 médecins d'obtenir de meilleurs résultats de santé pour leurs patients en fournissant une aide à la décision clinique au point de service.

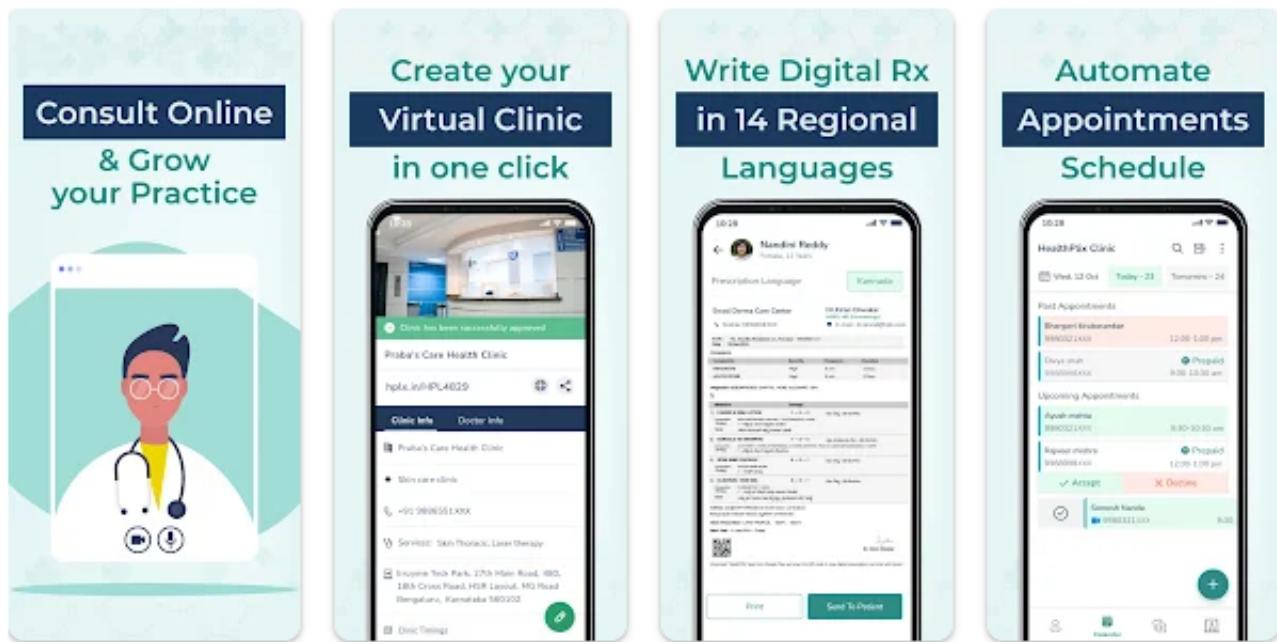


FIGURE 1.3 – Imprime Ecran de l'interface d'accueil de HealthPlix SPOT

- Mfine est une plateforme numérique de soins de santé primaires à la demande qui propose des diagnostics professionnels et des services de bilan de santé pouvant être utilisés dans le confort de la maison, du bureau ou même d'un spa. Depuis sa création, Mfine a redéfini le secteur des soins de santé primaires. Animée par la passion de prodiguer des soins, animée d'un souci constant de la qualité et guidée par une intelligence artificielle révolutionnaire, l'effort de Mfine a toujours été de rendre la bonne santé plus accessible, fiable et sans tracas pour tous. Avec l'aide de Mfine, les utilisateurs peuvent instantanément consulter des médecins en ligne et obtenir des bilans de santé à domicile.

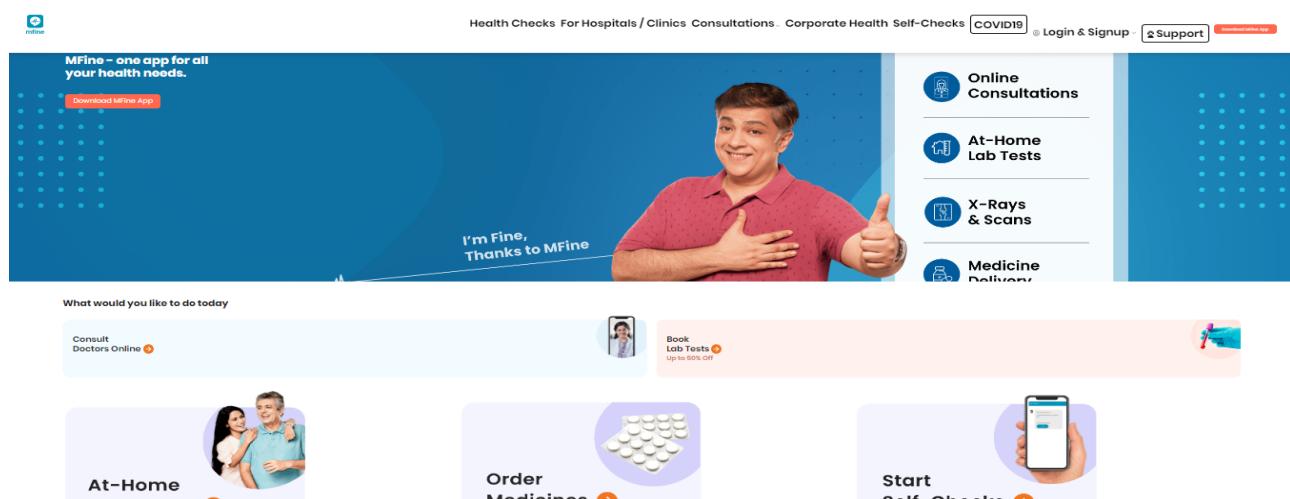


FIGURE 1.4 – Imprime Ecran de l'interface d'accueil de Mfine

## 1.4 Critiques de l'existant

La plupart des patients sont capables de décrire et détailler les symptômes ou/et les douleurs qu'ils ressentent sans pouvoir spécifier la maladie.

Le patient peut alors dans ce cas se diriger vers un mauvais spécialiste dû au manque de connaissances dans le domaine de la santé et perdre du temps pour être pris en charge.

Le problème qui se pose alors est comment faire pour construire un système capable de détecter automatiquement la maladie de manière fiable et précise en se basant sur la description du patient ?

## 1.5 Solution proposée

Le projet qui nous a été confié consiste à la réalisation d'une application mobile, assurant plusieurs fonctionnalités qui consistent à :

Gérer des acteurs et comptes que se soit pour les patients ou pour les médecins et à gérer aussi des réservations des rendez-vous.

Elle permet également à filtrer des recherches multiples, c'est à dire recherche selon différents critères.

Cette application assure une communication fluide et ce par la messagerie en temps réel ou par appel vocal.

Elle peut aussi faciliter la localisation des emplacements des médecins et le tracking et gérer la location de la maison et de son emplacement d'une façon commode.

Notre solution consiste à créer un modèle de Machine Learning d'apprentissage automatique supervisé qui doit être capable de détecter automatiquement les maladies depuis les symptômes saisis par les patients.

## 1.6 Méthodologie de conception

Il existe une diversité de méthodologies de gestion de projet disponibles dont l'objectif est d'améliorer la planification du projet, de résoudre des problèmes et de gérer les ressources du projet. Les méthodologies agiles se sont avérées plus efficaces que les méthodes traditionnelles , c'est pour cette raison la décision a été de les adopter. Méthodologies Agiles : L'objectif principal de la pratique agile est de découvrir les exigences et de développer des solutions grâce à la coopération entre des équipes auto-organisées et interconnectées et leurs clients finaux . Il existe plusieurs méthodologies agiles disponibles, notamment FDD, SCRUM, AUP et XP. Pour mener à bien notre projet, nous aurons besoin d'une méthodologie dynamique et collaborative basée sur le dialogue et la communication. De ce fait, la méthodologie la plus appropriée dans notre cas est agile SCRUM. C'est en fait la méthodologie que nous allons adopter. Agile Scrum : SCRUM est un cadre méthodologique de gestion de projet. C'est aujourd'hui la méthodologie Agile la plus utilisée car elle est simple, pragmatique, transparente et empirique. En conséquence, il est beaucoup plus facile pour les membres de l'équipe de travailler ensemble. Scrum team : Il n'y a que trois rôles dans SCRUM :

- Product Owner : Cette personne est à la charge de réaliser la vision du produit et d'interagir avec l'équipe de développement. Il s'agit généralement d'un expert en la matière dans le domaine d'activité du projet.
- Development team : Ce groupe est chargé de traduire les besoins exprimés par le Product Owner en fonctionnalités utilisables par la suite.
- Scrum Master : C'est la personne qui connaît Scrum et qui s'assure qu'il est utilisé correctement.
- Scrum Events : Les événements SCRUM sont des concepts fondamentaux du développement agile.

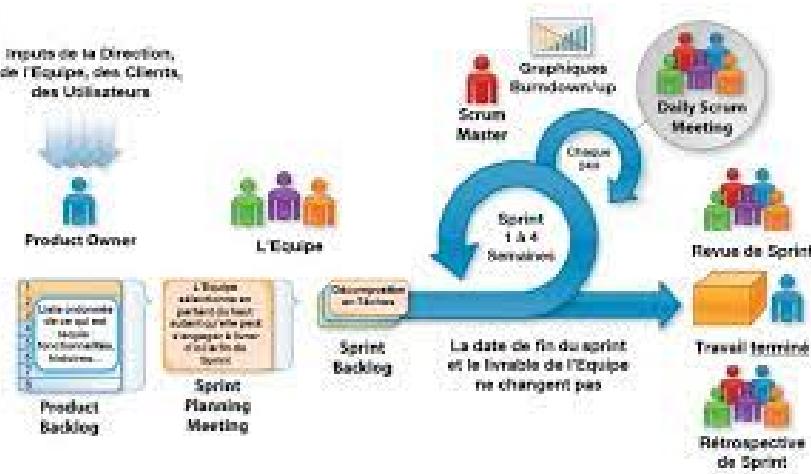


FIGURE 1.5 – La méthodologie agile

Voici les événements qui se sont produits :

- Sprint Planning : Le Sprint Planning s'organise au début de chaque sprint. Il s'agit d'une réunion entre le propriétaire du produit et l'équipe de développement pour examiner le backlog du produit.
- Le Daily Scrum Meeting : Le Daily Scrum Meeting est un rassemblement généraliste qui doit avoir lieu tous les matins pour définir les objectifs de la journée et identifier les éventuels blocages.
- La Sprint Review : Il s'agit d'une réunion où les œuvres créées seront présentées.
- Rétrospective de sprint : Il s'agit d'une réunion tenue à la fin d'un sprint pour discuter du processus, des problèmes qui se sont posés et de la façon dont les choses se sont déroulées afin de concevoir un plan d'action solide pour le prochain sprint.
- Artefacts Scrum : SCRUM possède également trois artefacts. Voici les artefacts :
- Le Sprint Backlog : n'est accessible qu'à l'équipe de développement. En fait, c'est l'ensemble des composants qui seront développés et testés par l'équipe.
- Product Backlog : Rassemble toutes les fonctionnalités et les exigences du produit pour le développement.
- Le Product Incrément : Créé par l'équipe de développement lors de chaque sprint, il regroupe les tâches ou user stories qui font partie de ce sprint. Il n'est disponible que pour l'équipe de développement. En fait, c'est l'ensemble des composants qui seront développés et testés par l'équipe. Par la suite, nous créerons un langage de modélisation que nous utiliserons afin de mieux visualiser la conception de notre système.

## Conclusion

Dans ce chapitre, nous avons introduit la société PlanetConectus à travers une description générale de ses domaines d'activités. Nous avons fait une étude de l'existant et nous avons aussi déterminé les insuffisances relevées.

Dans le chapitre suivant, nous allons présenter les besoins des différents utilisateurs de l'application.

## Chapitre 2

# Analyse des besoins et Conception

## Introduction

L'étape de la spécification constitue la base d'un bon départ de notre travail. En effet, l'adéquation de toute application à réaliser aux besoins de ses utilisateurs garantira sa réussite. Pour cela, nous étudions.

Dans ce chapitre, les besoins fonctionnels et non fonctionnels de notre client.

Nous présentons une spécification de ces besoins par les diagrammes de cas d'utilisation et diagramme de classe suivant la modélisation UML.

### 2.1 Identification des besoins

A ce niveau d'abstraction, nous devons capturer les besoins principaux des utilisateurs.

Il ne faut pas chercher la complexité, mais clarifier, filtrer et organiser les besoins.

Le but de cette phase est de définir le contour du système à modéliser, et de capturer les fonctionnalités principales du système.

#### 2.1.1 Les besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins fonctionnels qui expriment une action qui doit être menée sur l'infrastructure à définir en réponse à une demande, c'est le besoin exprimés par les utilisateurs. Cette partie structure les exigences fonctionnelles ébauchées.

En effet le système doit permettre de :

- Authentifier les patients ou médecins,
- Gérer des réservations des rendez-vous,
- Recherche selon différents critères,
- Messagerie en temps réel, call vocal,
- Localiser les emplacements des médecins et le tracking,
- Gestion de location de la maison et sa localisation,
- Classification des maladies des patients selon le texte de description des symptômes(AI),
- Les interfaces doivent être conviviales et faciles à utiliser,

#### 2.1.2 Les besoins non fonctionnels

Les besoins non fonctionnels spécifient les propriétés du système, notamment les contraintes d'environnement et d'implémentation , les performances, les dépendances de plate-forme, la capacité de maintenance, l'extensibilité et la fiabilité.

Ce sont les caractéristiques techniques imposées par l'application. Elles sont exprimées selon trois volets principaux à savoir :

- **Les exigences de qualité :** Il faut penser à concevoir des interfaces faciles à comprendre, simples à utiliser et contenant toutes les données nécessaires pour aider l'utilisateur à réaliser son travail sans difficulté.
- **Les exigences de performance :** Elles s'imposent aux besoins fonctionnels à certaines conditions telles que la vitesse, le débit, le temps de réponse et l'utilisation de la mémoire.  
Donc la performance du système est fondamentale pour améliorer son utilisation.
- **Les exigences de sécurité :** Pour le développement de l'application, des mesures de sécurité doivent être prises en compte, à savoir :
  - L'authentification des utilisateurs du système

- La gestion des droits d'accès au système
- Besoin d'établissement de la connexion au niveau d'accès (Ajouter, modifier, lire, écrire...)
- Politique de réutilisation
- Déconnexion

## 2.2 Identification des acteurs

L'analyse débute par la recherche des acteurs du système, nous rappelons qu'« un acteur est une entité (humaine ou machine) située hors du système permettant de déterminer les limites en jouant un rôle important et en déclenchant un stimulus initial entraînant une réaction» (Pierre-Alain et Nathalie, 2004).

L'application à réaliser nécessite les acteurs suivants :

TABLE 2.1 – Liste d'identification des acteurs

Nom d'acteur	Types	Fonctionnalités
L'administrateur	Humain	<p>C'est le gérant de l'application, il a une visibilité totale sur la base de donnée. Il a pour tâches de gérer tout le système.</p> <ul style="list-style-type: none"> <li>— La gestion des maisons à louer</li> <li>— la localisation des maison à louer</li> </ul>
Médecin	Humain	<p>Après l'authentification, il a le droit de réaliser les taches spécifiques suivantes.</p> <ul style="list-style-type: none"> <li>— Gérer les rendez-vous</li> <li>— Consulter l'historique des rendez-vous</li> <li>— Consulter l'état des RDV</li> <li>— Lancer la conversation chat en temps réel et communication/appel vocal avec le patient</li> <li>— Gérer son profil</li> </ul>
Patient	Humain	<p>Cette personne après l'authentification, il a le droit de réaliser les taches Spécifiques suivantes.</p> <ul style="list-style-type: none"> <li>— Saisir la description des symptômes(AI)</li> <li>— Trouver un médecin</li> <li>— Gérer les rendez-vous avec les médecin</li> <li>— Gérer la réservation des maisons à louer</li> <li>— Consulter l'historique des rendez-vous</li> <li>— Consulter l'état des RDV</li> <li>— Consulter le map pour visualiser l'emplacement de médecin</li> <li>— Utiliser le chat en temps réel et call vocal avec le médecin</li> <li>— Consulter le map pour visualiser l'emplacement des maisons à louer</li> <li>— Consulter la liste des maisons à louer</li> <li>— Gérer son profil</li> </ul>

## 2.3 Diagramme de cas d'utilisation

Les cas d'utilisation décrivent le comportement du système du point de vue utilisateur sous forme d'actions et de réactions. Un cas d'utilisation indique une fonctionnalité du système déclenchée par un acteur externe au système. Ce genre de diagramme permet de mettre en place et de comprendre les besoins du client. Dans le diagramme, interviennent trois éléments : les acteurs, le système et les cas d'utilisation. L'acteur représente un rôle joué par une personne ou un autre système qui interagit avec le système en cours de modélisation. Un cas d'utilisation regroupe plusieurs scénarios du système.

### 2.3.1 Diagramme du cas d'utilisation général

Le modèle de cas d'utilisations englobe l'ensemble des acteurs et des cas d'utilisations. Les cas d'utilisations favorisent la mise en œuvre d'un développement itératif. Chaque itération est pilotée par les cas d'utilisations.

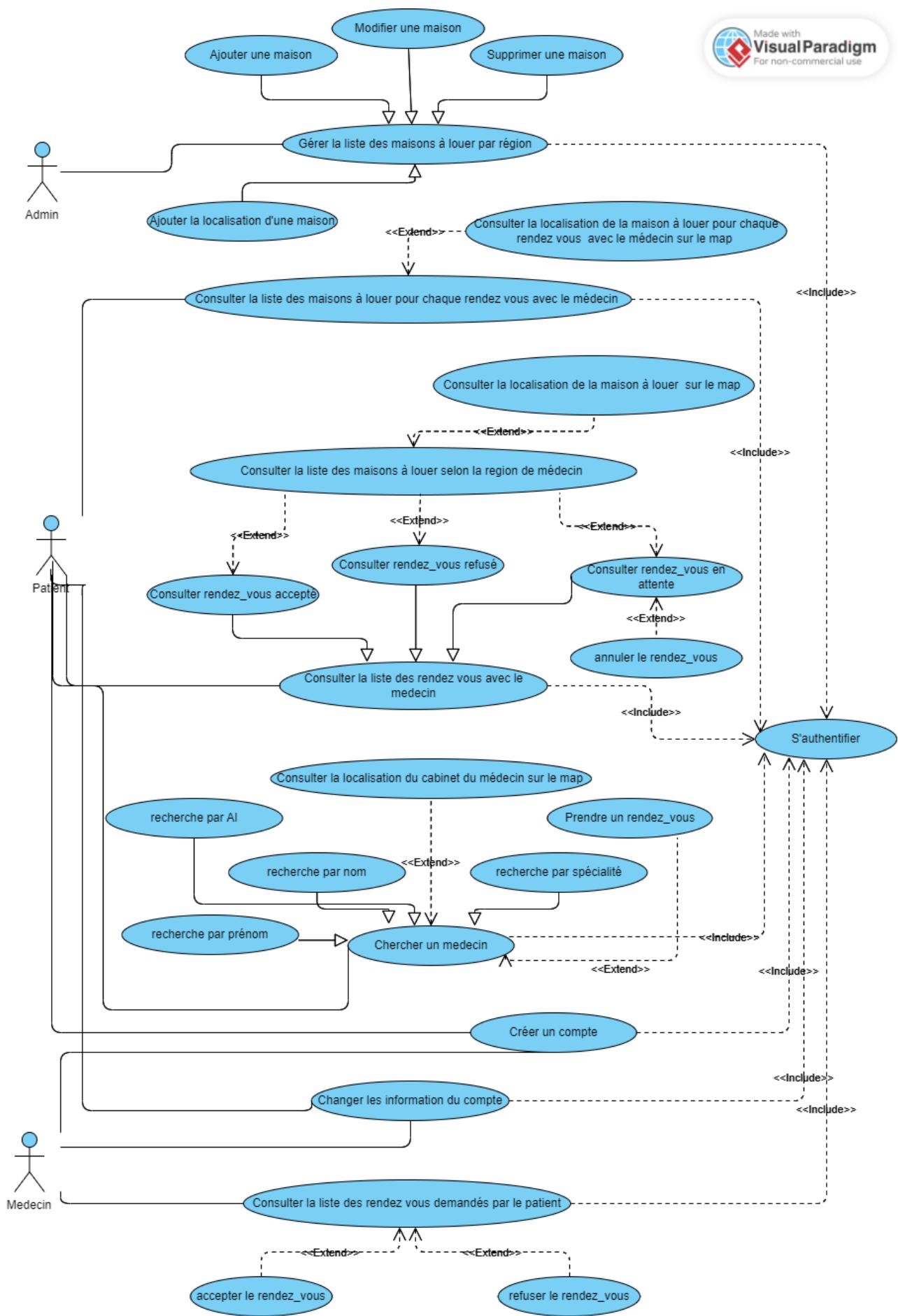


FIGURE 2.1 – Diagramme de cas d'utilisation globale

- A chaque opération effectuée, tous les acteurs du système doivent s'authentifier moyennant un nom d'utilisateur et un mot de passe.
- Nous avons choisi la présence de l'agent sur le diagramme de séquence et le diagramme de cas d'utilisation à la suite de l'accès qui lui a été attribué par l'administrateur.

## 2.4 Diagramme de classes

Nous allons nous intéresser à expliciter les classes élaborées durant le cycle de développement de l'application. Le diagramme de classes constitue un élément très important de la modélisation : il permet de définir quelles seront les composantes du système final : il ne permet en revanche pas de définir le nombre et l'état des instances individuelles. Néanmoins, on constate souvent qu'un diagramme de classes proprement réalisé permet de structurer le travail de développement de manière très efficace ; il permet aussi, dans le cas de travaux réalisés en groupe (ce qui est pratiquement toujours le cas dans les milieux industriels), de séparer les composantes de manière à pouvoir répartir le travail de développement entre les membres du groupe.

Une classe est composée :

- Des attributs : il s'agit des données caractérisant l'objet.
- Des méthodes : les méthodes d'un objet caractérisent son comportement, c'est-à-dire l'ensemble des actions que l'objet est à même de réaliser.
- Identité : L'objet possède une identité, qui permet de le distinguer des autres objets, indépendant de son état.

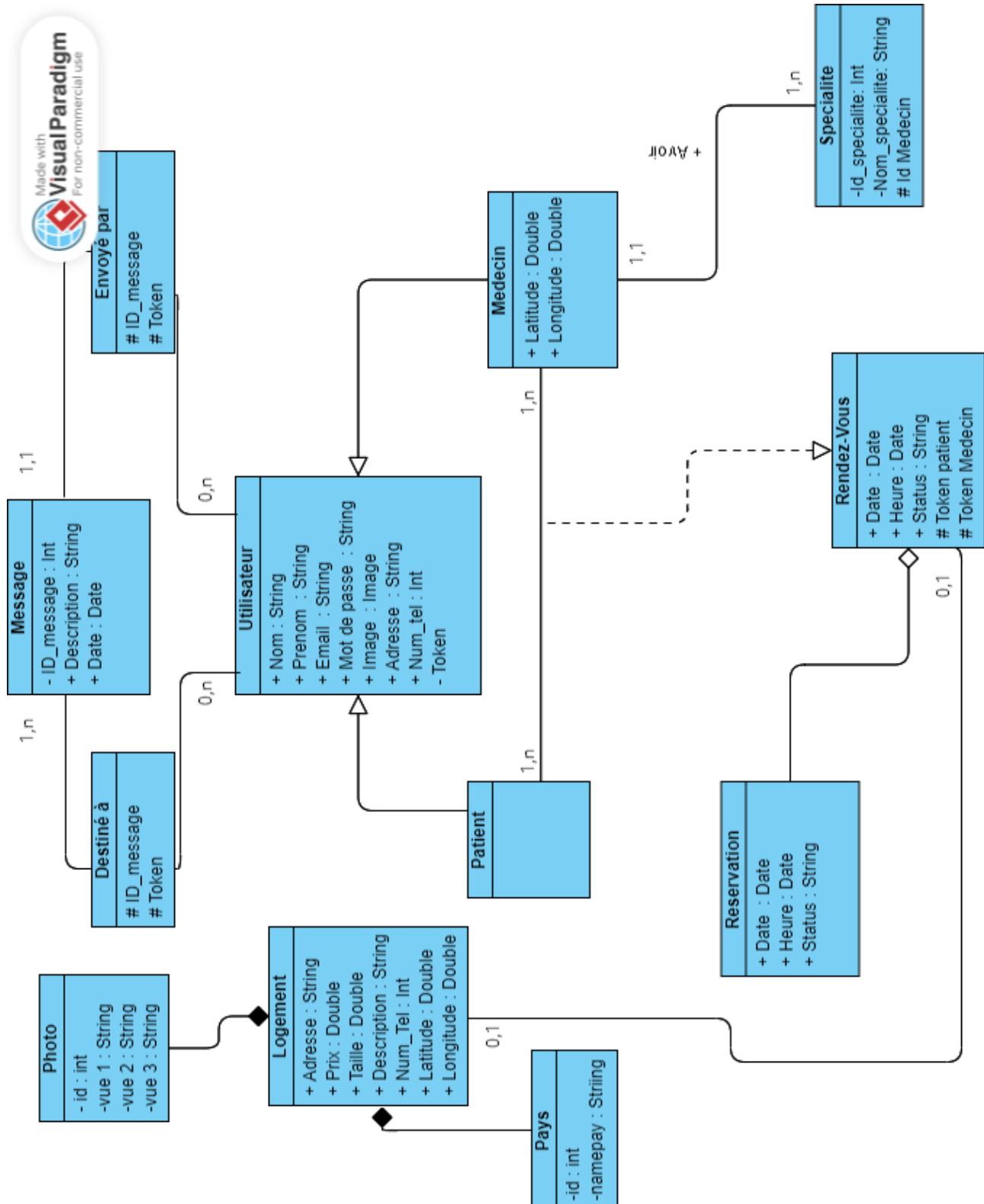


FIGURE 2.2 – Diagramme de classes

## 2.5 Planning du traitement des cas d'utilisation

Après l'identification des cas d'utilisation, nous devons maintenant les classifier. La classification des cas d'utilisation doit tenir compte d'un facteur principal qui est la priorité. Cette technique est généralement utilisée lors de la conception des applications se basant sur le processus unifié, mais elle reste valable et intéressante pour notre cas.

## 2.6 Priorité

Généralement, on dit qu'un cas d'utilisation A est plus prioritaire que B, si sa réalisation accélère la stabilisation du système. Le choix des priorités dans cette section s'est basé sur la dépendance entre les fonctionnalités de l'application. Nous pouvons dégager trois niveaux de priorité qui sont : priorité haute, moyenne et faible

## 2.7 Le backlog du produit

Le backlog du produit est l'artefact le plus important de Scrum, c'est l'ensemble des caractéristiques fonctionnelles ou techniques qui constituent le produit souhaité. Les caractéristiques fonctionnelles sont appelées des « histoires utilisateur » (user story) et les caractéristiques techniques sont appelées des « histoires techniques » (technical story). Le tableau ci-dessous résume le backlog produit de notre application. Il est à noter que nous n'avons pas cité les histoires techniques comme la préparation de la maquette graphique, les travaux de conception et les jeux de tests, etc.

## Conclusion

Dans ce chapitre, nous sommes concentré dans primordialement sur la Détermination des besoins fonctionnels et non fonctionnels que l'application aura à satisfaire.

Seuls l'architecture et l'environnement de travail sont à préciser, ce que nous verrons au chapitre suivant.

## Chapitre 3

Sprint 0 : Environnement de travail

## Introduction

Ce chapitre traite les aspects techniques liés à l'implémentation et la mise en œuvre de notre application. Nous y présenterons d'abord nos choix des techniques et outils adoptés pour le développement du système. Après cela, nous passerons en revue les langages de programmation et les architectures à suivre.

### 3.1 Environnement de développement

Durant la phase de développement de notre application, nous avons utilisé un ensemble d'outils et de moyens techniques qui constituent l'environnement général de travail. Ce dernier peut être réparti en deux types : un environnement matériel et un environnement logiciel. En ce qui suit, nous allons décrire et détailler chacun d'entre eux.

#### 3.1.1 Environnement matériel

Dans le cadre de ce projet, nous avons utilisé un ordinateur portable dont voici ses configurations :

Marque	Processeur	RAM	Système d'exploitation
Hp	Intel(R) Core (TM) i3 CPU @ 2.13 GHZ	16 Go	Windows 10 64-bits

#### 3.1.2 Environnement logiciel

##### Choix d'outils de développement :

Nous avons utilisé les outils de travail Mangodb, Flutter, python3 et Nodjs suite à la recommandation de l'entreprise qui utilise déjà ces langages de programmation dans leurs projets.

##### Visual Studio Code

Visual Studio Code est une version réduite de l'environnement officiel de développement Microsoft qui se concentre exclusivement sur l'éditeur de code. Il est multiplateforme et supporte la syntaxe pour un large éventail de langages.



FIGURE 3.1 – Logo Visual Studio Code

##### MongoDb

Est un système de gestion de base de données orienté document, qui peut être distribué sur un nombre quelconque d'ordinateurs et ne nécessite pas un diagramme prédéfini des données. Il est écrit en C++.



FIGURE 3.2 – Logo MongoDB

## Colab

Colab (ou "Colaboratory") vous permet d'écrire et d'exécuter le code Python dans votre navigateur avec

- Aucune configuration requise
- Accès gratuit aux GPU
- Partage facile



FIGURE 3.3 – Logo Colab

## Postman

Postman est un outil de développement d'API qui comprend une variété de fonctionnalités. Il nous permet de gérer et de tester notre API REST.



FIGURE 3.4 – Logo Postman

## GitHub

GitHub offre un service d'hébergement et de gestion du développement de logiciels basé sur le Web. En effet, il contribue à la maintenance des dépôts git afin d'assurer la collaboration et l'échange du code avec tous les membres de l'équipe.



FIGURE 3.5 – Logo GitHub

## Flutter

Flutter est un framework Google open source pour créer de belles applications multi-plateforme nativement compilées à partir d'une seule base de code.



FIGURE 3.6 – Logo Flutter

## Python

C'est un langage de programmation de haut niveau. Il supporte la programmation impérative structurée, fonctionnelle et orientée de l'objet. Plusieurs bibliothèques sont prévues pour faciliter les développements dans les domaines de l'apprentissage automatique, du Big Data et de la science des données.

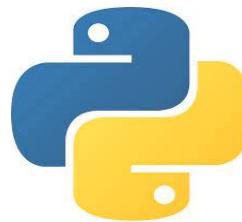


FIGURE 3.7 – Logo Python

## Node js

NodeJS est un environnement d'exécution pour l'utilisation de la partie JavaScript serveur. Grâce à son fonctionnement non bloquant, il permet de concevoir des applications en réseau performantes, telles qu'un serveur web, une API ou un job CRON.

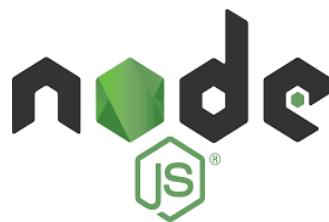


FIGURE 3.8 – Logo Node js

## Flask

Il s'agit d'un microframework libre pour le développement web en Python. Il est basé sur deux modules dont le but principal est de préserver la flexibilité de la programmation Python en intégrant un système de modèles.

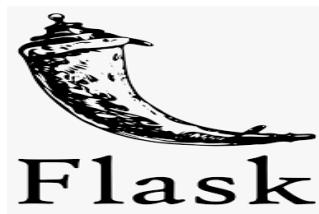


FIGURE 3.9 – Logo Flask

### Scrapy

Scrapy est un framework Python qui sert à faire du web scraping. Scrapy est adapté aux grands projets de web scraping. En effet, les projets Scrapy possèdent une structure assez claire qui facilite l'entretien et l'évolutivité. En outre, le framework offre une certaine vitesse à cause de l'asynchronisme des requêtes (Scrapy utilise Twisted).



FIGURE 3.10 – Logo Scrapy

### Socket.IO

Socket.IO est une bibliothèque qui permet une communication à faible latence, bidirectionnelle et basée sur les événements entre un client et un serveur. Il repose sur le protocole WebSocket et offre des garanties supplémentaires telles qu'un mode dégradé en HTTP long-polling ou la reconnexion automatique.



FIGURE 3.11 – Logo Socket.IO

### Outils de conception : Visual paradigm online

Visual Paradigm Online (VP Online) Free Edition est un logiciel de dessin en ligne GRATUIT développé pour la communauté. Il permet de dessiner des diagrammes gratuits avec un outil de dessin en ligne facile à utiliser sans limitations telles que le nombre de diagrammes, la taille du fichier de projet,etc.



FIGURE 3.12 – Logo Visual paradigm online

## Réalisation de mise en oeuvre de rapport : Overleaf

Overleaf est une plateforme en ligne gratuite permettant d'éditer du texte en LATEX sans aucun téléchargement d'application. En outre, elle offre la possibilité de rédiger des documents de manière collaborative, de proposer ses documents directement à différents éditeurs (IEEE Journal, Springer, etc.) ou plateformes d'archives ouvertes (arXiv, engrxiv, etc.) pour une éventuelle publication.



FIGURE 3.13 – Logo Overleaf

## 3.2 Architecture

### 3.2.1 Architecture logique

Nous Mentionnons les trois méthodes d'architectures logiques existantes : MVC ,MVP et MVVM.  
Nous avons utilisé la méthode d'architecte (MVVM) tout au long du projet.

Model-View-ViewModel (MVVM) est un modèle d'architecture logicielle qui prend en charge la séparation de l'interface utilisateur (qui est la vue) du développement de la logique métier ou de la logique principale (modèle). Le modèle de vue à l'intérieur de MVVM est le pont responsable de la conversion des données d'une manière qui se comporte conformément aux changements qui se produisent sur l'interface utilisateur.

De plus, pour connaître les responsabilités des trois composants, il est également important de comprendre comment les composants interagissent les uns avec les autres. Au niveau le plus élevé, la vue "connaît" le modèle de vue et le modèle de vue "connaît" le modèle, mais le modèle ne connaît pas le modèle de vue et le modèle de vue ne connaît pas la vue.

Le modèle (ou Model) :

Les modèles contiennent des informations mais ne gèrent généralement pas le comportement. Ils ne formatent pas les informations et n'influencent pas l'apparence des données. Le modèle dans le modèle de conception MVVM représente les données réelles qui seront utilisées dans le développement d'applications.

La vue (ou View) :

Il s'agit essentiellement de la seule partie de l'application avec laquelle les utilisateurs interagissent réellement. Par exemple, l'utilisateur appuie sur le bouton, fait défiler la liste, modifie les paramètres, etc. Ces événements sont ensuite transmis au modèle de vue, qui effectue le traitement et renvoie la réponse attendue de l'utilisateur (qui est une forme d'interface utilisateur). Il est important de se rappeler que la vue n'est pas responsable ici de la gestion de l'état.

Le ViewModel :

Le ViewModel agit comme un intermédiaire entre la vue et le modèle, de manière à fournir des données à l'interface utilisateur. Le ViewModel peut également exposer des méthodes pour aider à maintenir l'état de la

vue, mettre à jour le modèle en fonction des actions sur une vue et déclencher des événements sur la vue. Pour Flutter, nous avons un écouteur appelé ChangeNotifier qui permet au ViewModel d'informer ou de mettre à jour la vue chaque fois que les données sont mises à jour.

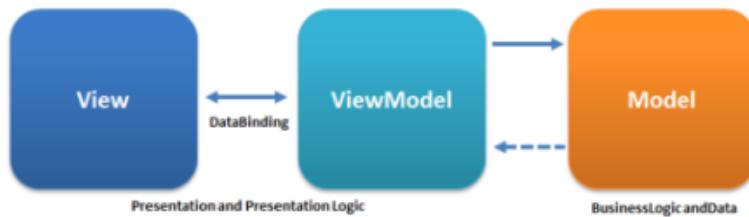


FIGURE 3.14 – Architecture MVVM

### 3.2.2 Architecture physique

Pour répondre aux besoins de nos clients et assurer la fiabilité de nos services, nous avons choisi une approche architecturale à trois niveaux. Cette conception permet de modéliser l'application sous la forme d'une hiérarchie à trois niveaux, avec les rôles suivants définis :

- Présentation tierce : qui comprend tout ce qui concerne le client.
- Tiers Web : Nos ViewModel communiquent avec le Front-End (vues) et la base de données.
- Couche de persistance : c'est tout ce qui concerne la base de données.

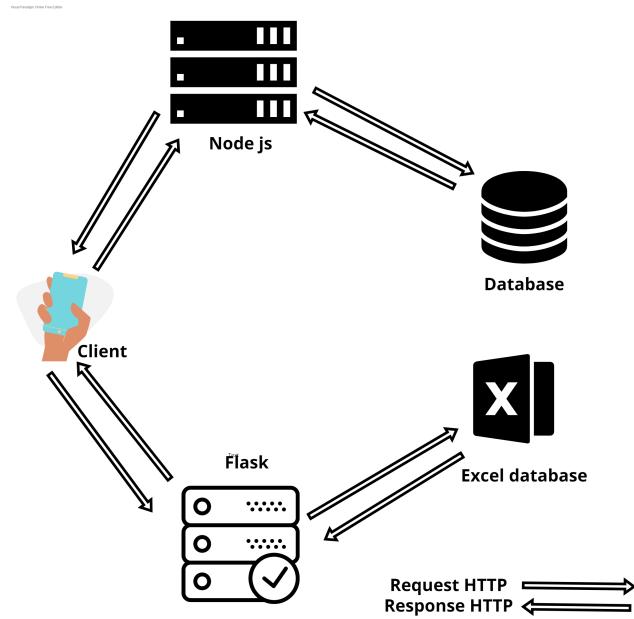


FIGURE 3.15 – Architecture physique

## Conclusion

Au niveau de ce chapitre, nous avons présenté l'environnement de développement de notre travail, les outils utilisés ainsi que les architectures (logique et physique). Il est maintenant temps de commencer l'analyse et le développement du sprint 1.

## Chapitre 4

Sprint 1 : Chercher un médecin

## Introduction

Dans ce chapitre nous avons créé un système de recherche pour un médecin selon le besoin de patient, on va détailler ce système premièrement par le backlog, puis par le diagramme de cas d'utilisation, par la suite la description textuelle du cas d'utilisation chercher un médecin par son nom, prénom ou par sa spécialité et par IA, détaillait le diagramme de séquence et la partie implémentation.

### 4.1 Backlog du sprint 1

Une fois nous avons défini le but de notre sprint, il est temps de décider quelles sont les fonctionnalités et leurs degrés d'importance inclus dans ce dernier.

TABLE 4.1 – Backlog du sprint 1

ID	User Story	Condition de satisfaction	Priorité
1	En tant qu'utilisateur, je peux chercher un médecin	<ul style="list-style-type: none"> <li>— Le patient peut chercher un médecin avec l'intelligence artificiel,</li> <li>— Le patient peut chercher un médecin par son nom ou prénom,</li> <li>— Le patient peut chercher un médecin par sa spécialité</li> </ul>	1

### 4.2 Raffinement du Cas d'utilisation « Chercher un médecin »

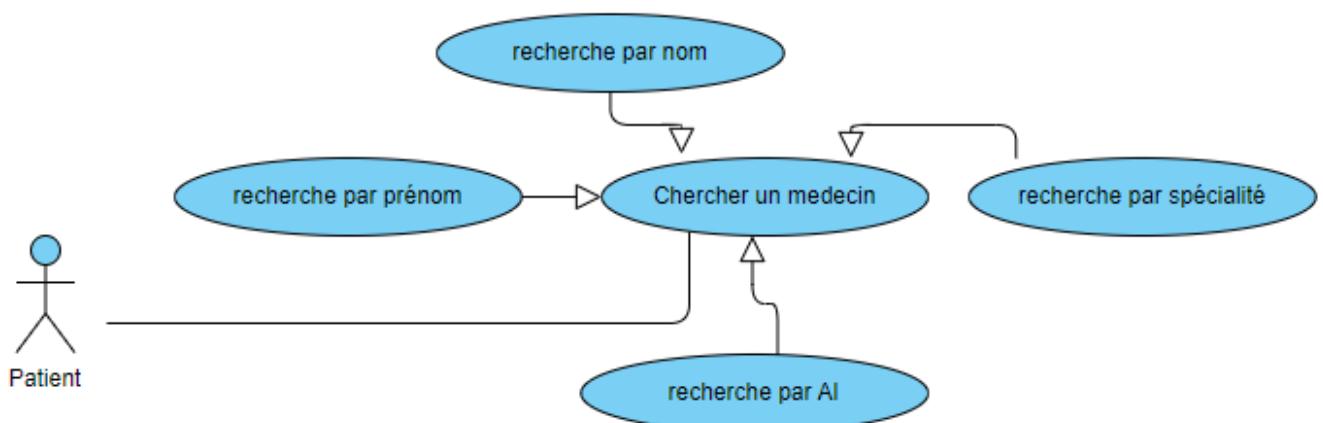


FIGURE 4.1 – Diagramme de cas d'utilisation "Chercher un médecin"

### 4.3 Description textuelle du cas d'utilisation chercher un médecin par son nom , prénom ou par sa spécialité

TABLE 4.2 – Description textuelle du cas d'utilisation chercher un médecin par son nom , prénom ou par sa spécialité

Cas d'utilisation	Chercher un médecin
Acteur	Patient
Précondition	<ul style="list-style-type: none"> <li>— Une connexion au serveur doit être établie.</li> <li>— Acteur authentifié.</li> <li>— Liste des docteurs consultée selon le texte où il va saisir.</li> </ul>
Postcondition	L'interface s'affiche
Scénario nominal	<ul style="list-style-type: none"> <li>— 1. Le patient clique sur "Search doctor"</li> <li>— 2. Le patient clique sur la zone de texte chercher un médecin par son nom , prénom ou par sa spécialité</li> <li>— 3. Le patient clique sur le boutons "Search"</li> <li>— 4. le système affiche le module et ses informations</li> </ul>
Scénario alternatif	[Aucun médecin trouvé] : Affiche une liste vide

TABLE 4.3 – Description textuelle du cas d'utilisation chercher un médecin par la technologie IA

Cas d'utilisation	Chercher un médecin par la technologie IA
Acteur	Patient
Précondition	<ul style="list-style-type: none"> <li>— Une connexion au serveur doit être établie.</li> <li>— Acteur authentifié.</li> <li>— Liste des docteurs consultée selon le texte où il va saisir.</li> </ul>
Postcondition	L'interface affiche la liste des médecins
Scénario nominal	<ul style="list-style-type: none"> <li>— 1. Le patient clique sur "Home"</li> <li>— 2. Le patient clique sur la zone de texte chercher un médecin par la technologie IA Le patient doit détailler leur maladie dans description</li> <li>— 3. Le patient clique sur le bouton "Search"</li> <li>— 4. le système affiche le module et ses informations</li> </ul>
Scénario alternatif	[Aucun médecin trouvé] : Affiche une liste vide

### 4.4 Diagrammes de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Cela indique les objets que l'acteur manipulera et les opérations qui se produiront. Il contient plusieurs objets et les messages transmis entre ceux-ci dans le cadre d'un framework utilisateur.

Les types de messages

UML fournit un certain nombre de stéréotypes graphiques qui décrivent la nature du message. •Message simple : Message qui ne spécifie aucune caractéristique d'envoi et de réception particulière.

•Message minuté (Timeout) : Bloque l'expéditeur pendant un temps donné (qui peut être spécifié dans une contrainte), en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié.

•Message synchrone : Bloque l'expéditeur jusqu'à la prise en compte du message par le destinataire.

Le flux de contrôle passe de l'émetteur au récepteur (l'émetteur devient passif et le récepteur actif) à la prise

en compte du message.

- Message asynchrone : N'interrompe pas l'exécution de l'expéditeur.

Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité).

- Message dérobant : N'interrompe pas l'exécution de l'expéditeur et ne déclenche une opération chez le récepteur que s'il s'est préalablement mis en attente de ce message.

## 4.5 Diagrammes de séquence détaillés

Nous vous présenterons les différents diagrammes de séquence liés aux cas d'utilisation "Chercher un médecin"

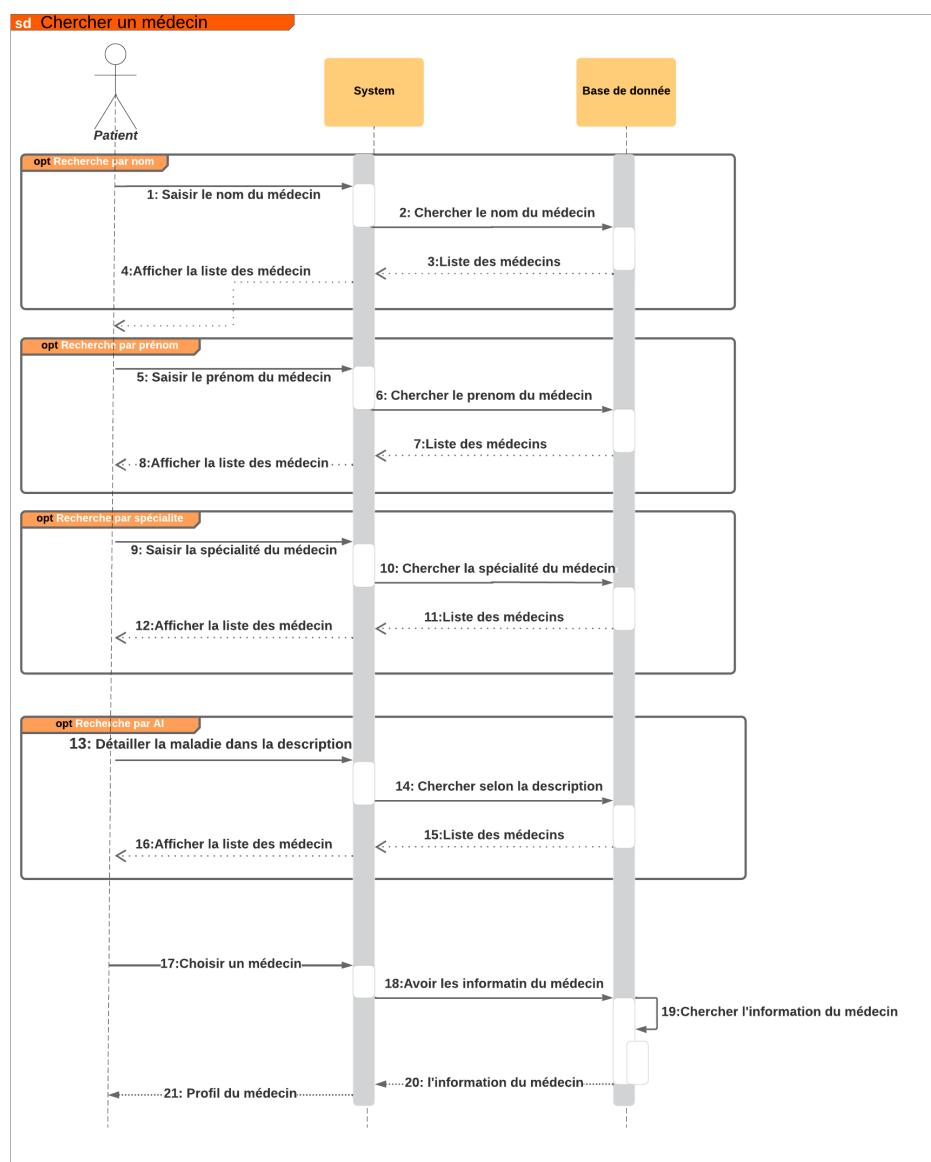


FIGURE 4.2 – Diagrammes de séquence chercher un médecin

## 4.6 Réalisation

Vous trouverez ci-dessous quelques captures d'écran de l'application.

Cette figure montre comment l'utilisateur/patient peut faire une recherche par le prénom du médecin.

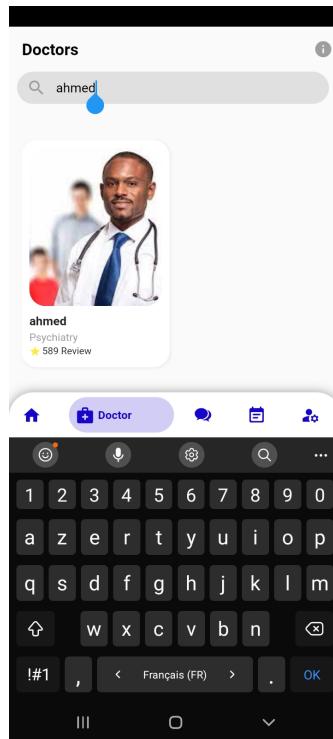


FIGURE 4.3 – Recherche par nom

Cette figure montre comment l'utilisateur/patient peut faire une recherche par spécialité exemple "Psychiatry".

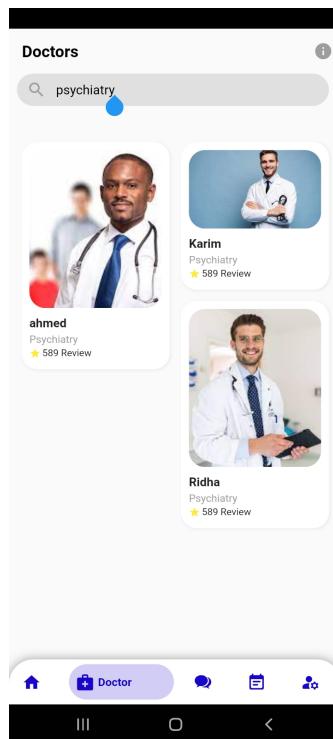


FIGURE 4.4 – Recherche par spécialité "Psychiatry"

Cette figure montre comment l'utilisateur/patient peut faire une recherche par spécialité exemple "Therapist".

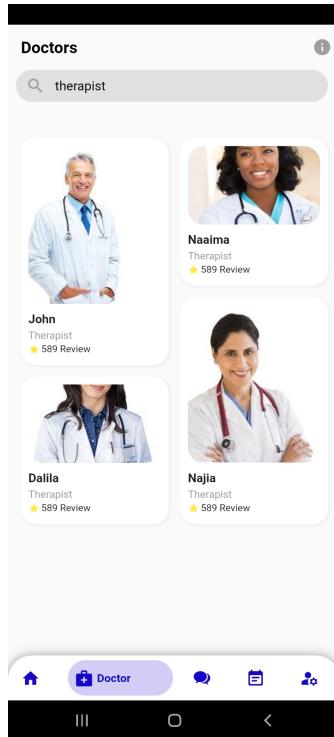


FIGURE 4.5 – Recherche par spécialité "Therapist"

Cette figure montre comment l'utilisateur/patient peut faire une recherche par spécialité exemple "Rheumatologist".

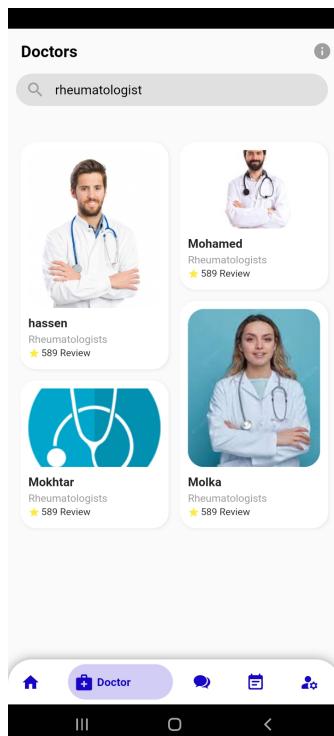


FIGURE 4.6 – Recherche par spécialité "Rheumatologist"

Cette figure montre l'interface que le patient utilise pour décrire ses symptômes et par le biais de l'intelligence artificielle une liste des médecins en concordance avec le texte inséré figurera.

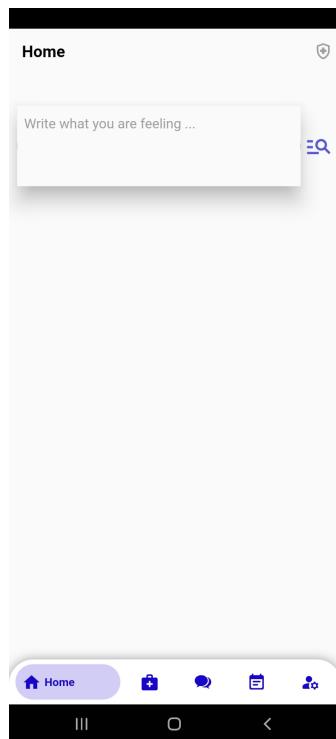


FIGURE 4.7 – Recherche par AI

Cette figure est un exemple de texte que l'utilisateur peut insérer pour décrire ses symptômes

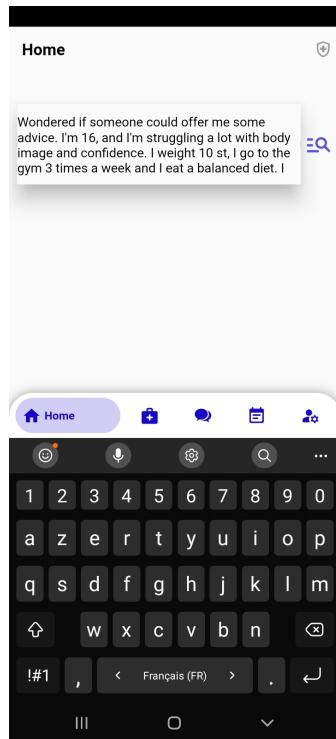


FIGURE 4.8 – Recherche par AI "La description du patient"

Cette figure montre la liste des médecins adéquats aux symptômes à travers l'intelligence artificielle

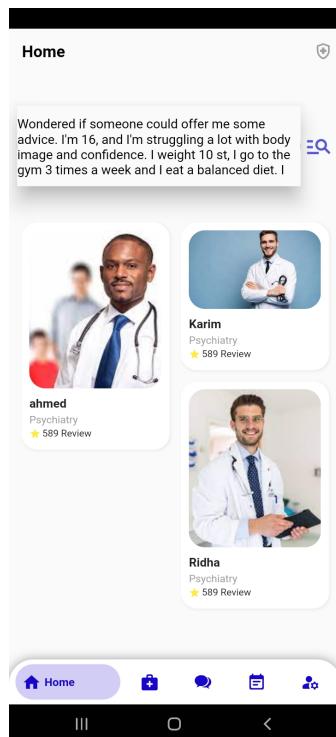


FIGURE 4.9 – Recherche par AI résultatat "Psychiatry"

Voici la même démarche ci-dessous avec d'autres exemples.

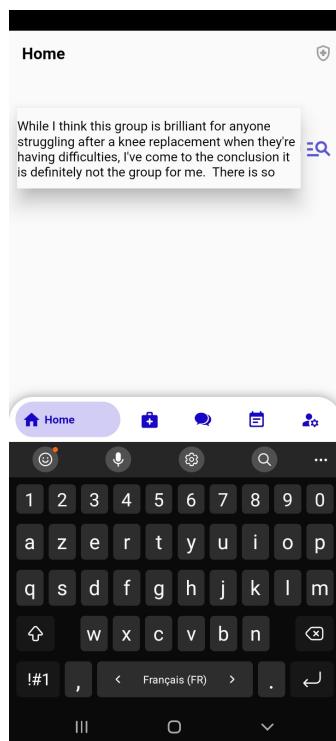


FIGURE 4.10 – Recherche par AI "La description du patient"

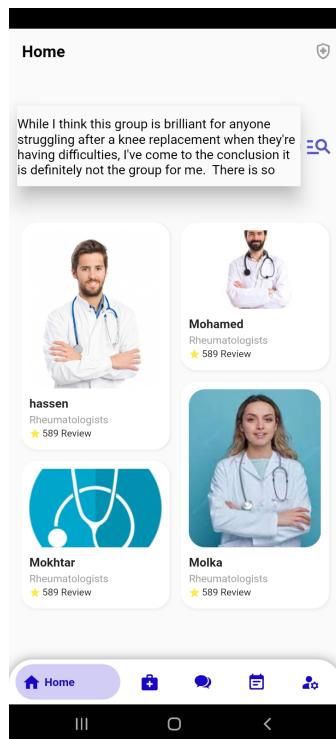


FIGURE 4.11 – Recherche par AI résultatat "Rheumatologist"

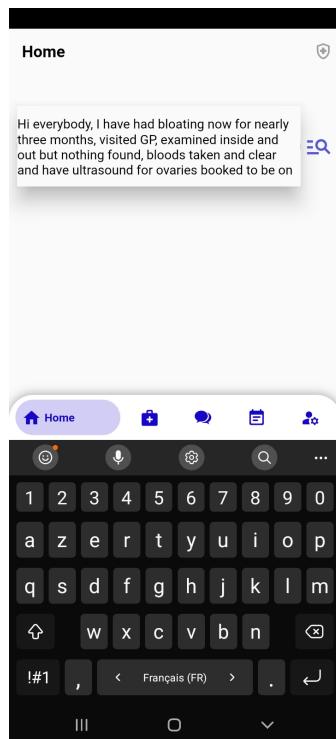


FIGURE 4.12 – Recherche par AI "La description du patient"

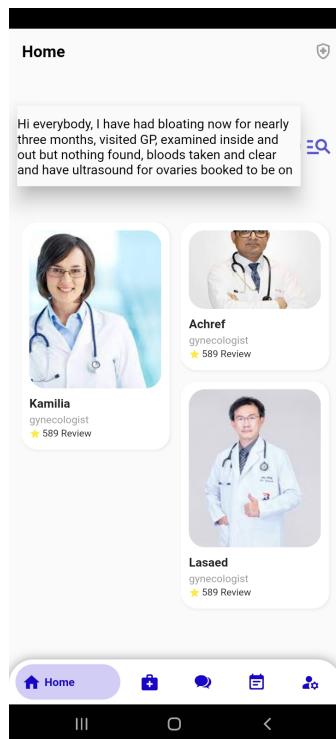


FIGURE 4.13 – Recherche par AI résultat "Gynecologist"

## Conclusion

Au niveau de ce chapitre, nous avons implémenté un système entièrement adaptable aux besoins de l'application. dans le sprint suivant, nous développerons le module prise rendez-vous.

# Chapitre 5

Sprint 2 : Prise de rendez vous

## Introduction

Après avoir présenté notre premier sprint, nous allons exposer dans ce chapitre le deuxième sprint selon la même démarche que le premier sprint qui contient donc le sprint backlog, la partie analyse, la partie conception et la partie implémentation.

### 5.1 Backlog du sprint 2

Une fois nous avons défini le but de notre sprint, il est temps de décider quelles sont les fonctionnalités et leurs degrés d'importance inclus dans ce dernier.

TABLE 5.1 – Backlog du sprint 2

ID	User Story	Condition de satisfaction	Priorité
1	En tant qu'utilisateur, je peux prendre un rendez-vous	<ul style="list-style-type: none"> <li>— Le patient peut choisir une date de rendez-vous</li> <li>— Le patient peut choisir l'heure de rendez-vous</li> <li>— Ce cas permet au patient de prendre un rendez-vous avec un médecin</li> </ul>	1

### 5.2 Raffinement du Cas d'utilisation « Prise de rendez-vous »

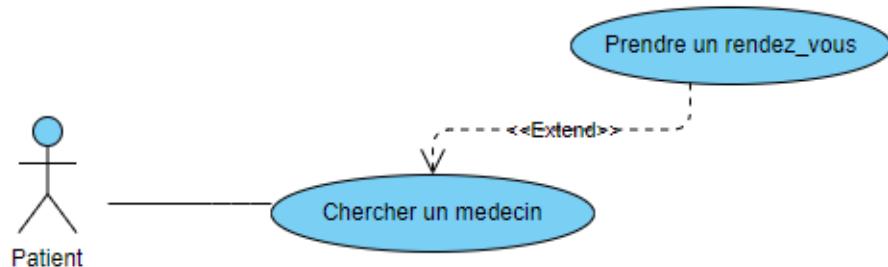


FIGURE 5.1 – Diagramme de cas d'utilisation "Prise de rendez-vous"

### 5.3 Description textuelle du cas d'utilisation Prise de rendez-vous

TABLE 5.2 – Description textuelle du cas d'utilisation Prise de rendez-vous

Cas d'utilisation	Prise de rendez-vous
Acteur	Patient
Précondition	<ul style="list-style-type: none"> <li>— Une connexion au serveur doit être établie.</li> <li>— Acteur authentifié.</li> <li>— Choisir un médecin.</li> <li>— La date ne correspond pas à la disponibilité du médecin.</li> </ul>
Postcondition	Le rendez-vous enregistré
Scénario nominal	<ul style="list-style-type: none"> <li>— 1. Le patient sélectionne la date du rendez-vous</li> <li>— 2. Le patient sélectionne l'heure du rendez-vous.</li> <li>— 3. Le système vérifie la disponibilité du médecin.</li> <li>— 4. Le rendez-vous est enregistré dans la base de données</li> </ul>
Scénario alternatif	[Aucune date sélectionné] : Affichage d'un message d'erreur "La date est introuvable"

### 5.4 Diagrammes de séquence détaillés

Nous vous présenterons les différents diagrammes de séquence liés aux cas d'utilisation "Prise de rendez-vous"

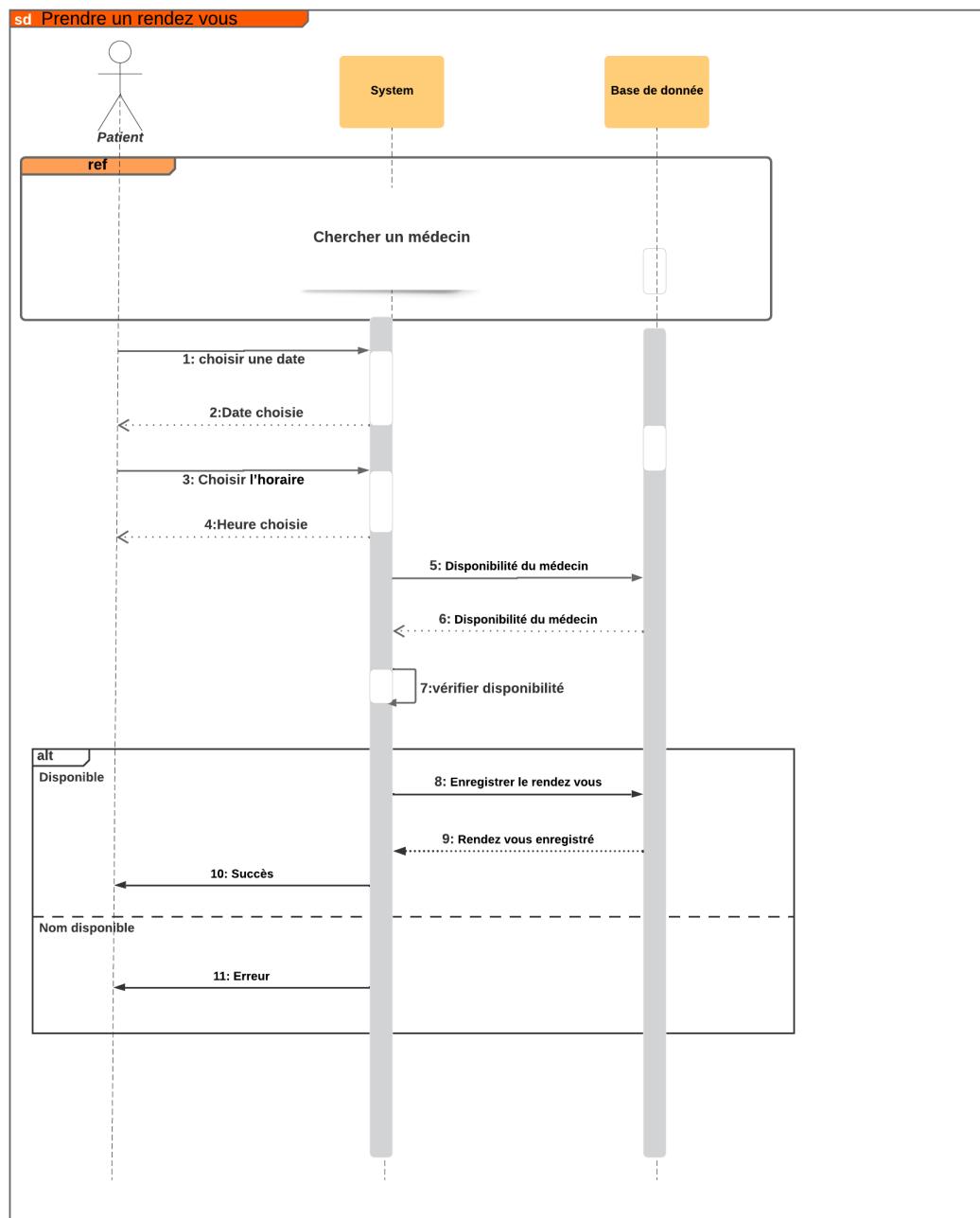


FIGURE 5.2 – Diagrammes de séquence Prise de rendez-vous

## 5.5 Réalisation

Vous trouverez ci-dessous quelques captures d'écran de l'application.  
Cette figure montre la liste des médecins.

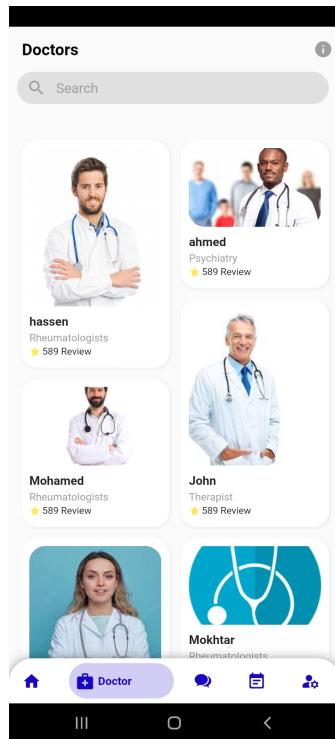


FIGURE 5.3 – Liste des médecins

Cette figure montre la sélection du médecin adéquat par l'utilisateur/patient.

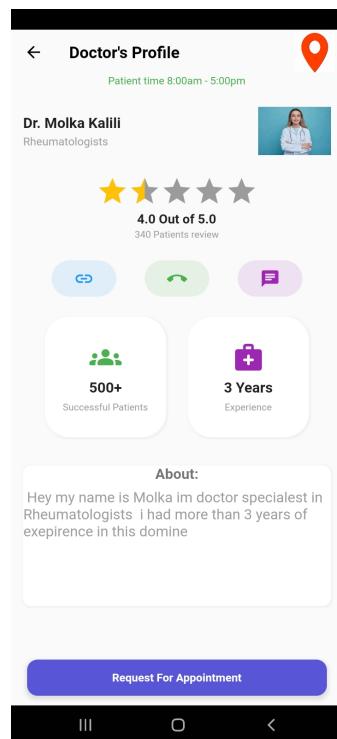


FIGURE 5.4 – Profil de médecin sélectionné par le patient

Cette figure montre comment la prise de RDV peut être effectuée par l'utilisateur/patient

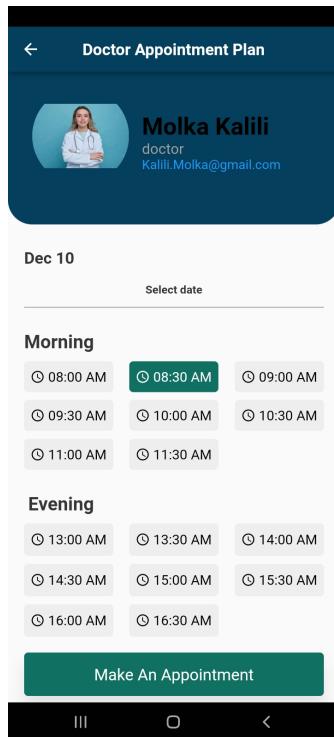


FIGURE 5.5 – le calendrier du médecin

Cette figure montre comment l'utilisateur/patient peut sélectionner la date du RDV

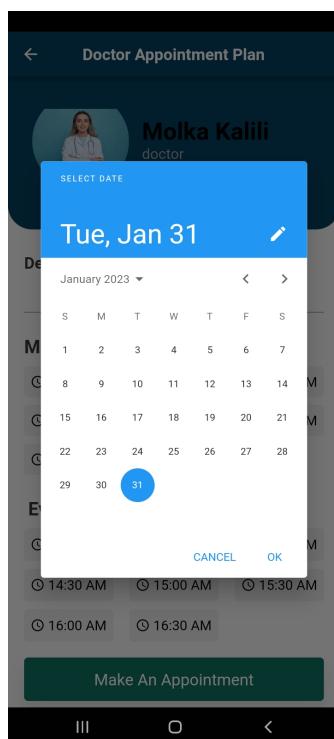


FIGURE 5.6 – Le patient choisit une date

Cette figure montre comment l'utilisateur/patient peut sélectionner l'horaire du RDV

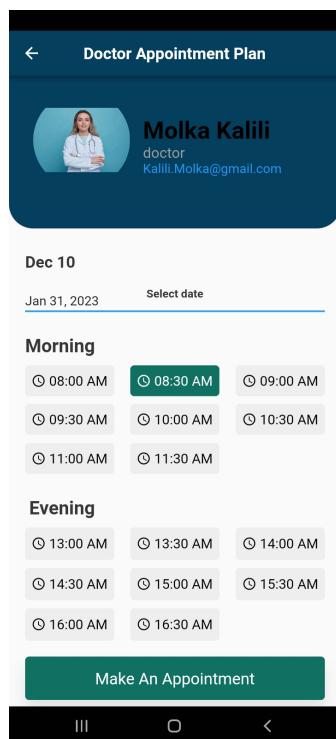


FIGURE 5.7 – Prise de rendez-vous

## Conclusion

Tout au long de ce chapitre nous avons exposé de façon précise tout le travail du premier sprint tout en détaillant toutes les étapes des méthodes scrum. Le chapitre suivant sera consacré à la partie de l'intelligence artificielle.

## Chapitre 6

Sprint 3 : Le modèle machine Learning

## Introduction

Dans ce chapitre nous commençons par définir la technologie IA. Ensuite, nous aborderons la technique IA différence entre Machine Learning et Deep Learning ,prétraitement des données,construction des modèles et finalement evaluation des modèles.

### 6.1 Technologie IA

La Machine Learning et le Deep Learning sont de l'Intelligence Artificielle. L'Intelligence Artificielle (IA) telle que nous la connaissons est une Intelligence Artificielle faible, par opposition à l'IA forte, qui n'existe pas encore. Aujourd'hui, les machines sont capables de reproduire un comportement humain, mais sans conscience. Plus tard, leurs capacités pourraient croître au point de se transformer en machines dotées de conscience, de sensibilité et d'esprit.

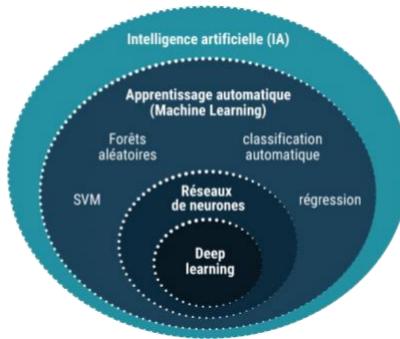


FIGURE 6.1 – Organisation de l'intelligence artificielle

- Machine Learning :

La Machine Learning ou apprentissage automatique est un domaine scientifique, et plus particulièrement une sous-catégorie de l'intelligence artificielle. Elle consiste à laisser des algorithmes découvrir des 'patterns', à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques.

- Deep Learning :

Le Deep Learning ou apprentissage profond est l'une des technologies principales de la Machine Learning. Avec le Deep Learning, nous parlons d'algorithmes capables de mimer les actions du cerveau humain grâce à des réseaux de neurones artificielles. Les réseaux sont composés de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente.

- Différence entre Machine Learning et Deep Learning :

Machine Learning et Deep Learning sont devenus des termes extrêmement utilisés dans le cadre de nos activités, avec des applications toujours plus nombreuses. Lorsque l'on parle de Deep Learning, nous parlons d'algorithmes capables de mimer les actions du cerveau humain grâce à des réseaux de neurones d'où le terme d'Intelligence Artificielle. Il est souvent expliqué que la différence entre Machine Learning et Deep Learning réside dans le fait que les algorithmes de Machine Learning vont traiter des données quantitatives et structurées (des valeurs numériques), contrairement à celles de Deep Learning qui traiteront des données non-structurées, comme le son, le texte, et l'image.

## 6.2 Acquisition des données

Notre projet est comme tout projet de data science basée sur les données. Comme mentionné, nos principales sources sont patient.info, med.tn,etc... Cela nous donne toutes sortes de données qui peuvent être utilisées dans différents classificateurs. Dans une première étape, nous définissons les techniques et bibliothèques utilisées pour la collecte de données, qui seront décrites dans la deuxième étape

- Collecte des données :

La collecte des données est le cœur d'un projet data science. Les sites web et principalement patient.info, med.tn constituent la meilleure source de collecte de données puisqu'il y a un nombre important des publications en dialecte Anglais. Nous avons constaté qu'il y a une méthode pour extraire les données depuis les sites web qui est le web Scraping. Dans ce qui suit, nous allons la détailler.

- Web Scraping :

Le web Scraping est une méthode qui permet d'extraire des informations à partir d'un site ou d'une page web. Python est parmi les langages qui offrent des bibliothèques pour faciliter l'utilisation de cette méthode nous pouvons citer les bibliothèques : request, beautifulsoup, tweepy, sélénium, scrapy. Nous avons choisi les deux dernières bibliothèques pour extraire les données à partir de patient.info et med.tn.

## 6.3 Prétraitement des données

La préparation des données est l'une des étapes les plus importantes et les plus chronophages. En fait, cela prend 50 à 70 pour cent de notre temps et de nos efforts sur un projet basé sur des estimations. Il existe plusieurs techniques de traitement automatique du langage. Nous décrivons les techniques les plus importantes.

- Suppression des HTML : Étant donné que nous utilisons le web scraping pour collecter des données à partir de sites Web, il est courant de trouver des balises HTML dans nos données, nous avons créé une fonction pour les supprimer.

- Suppression des emojis : Les emojis jouent un rôle très important de nos jours car les utilisateurs peuvent s'exprimer à l'aide d'emojis, nous avons donc implémenté la possibilité de supprimer les emojis des données sources.

- Suppression des mots vides : La suppression des mots vides vous permet de vous concentrer sur les mots qui comptent vraiment. Ils sont des termes qui apparaissent fréquemment dans un langage naturel. Leur présence ne fait absolument aucune différence sémantiquement et lexicalement, et leur élimination réduit la taille du document en supprimant les mots insignifiants. Par conséquent, le temps de traitement et le temps d'apprentissage seront considérablement réduits.

- Suppression des nombres : Nous avons mis au point une fonction remove number pour supprimer les chiffres des données recueillies.

- Suppression des espaces blancs (whiteSpaces) : Nous avons constaté qu'il n'était pas possible d'obtenir des informations à partir d'espaces, de retours à la ligne et de tabulations, c'est-à-dire que leur suppression n'affectait pas les performances de notre modèle.

- Suppression des hyperliens ou URL : Lors de la phase de « collecte de données », nous pouvons trouver des URL dans les publications ou les commentaires, nous avons donc développé une fonctionnalité qui permet de supprimer les liens hypertexte des données sources.

## 6.4 Description des données

Les données médicales sont extrêmement difficiles à trouver en raison des réglementations de confidentialité (Health Insurance Portability and Accountability Act(HIPAA)). Nous avons trouvé 2 sites qui ont des données

qu'on peut le scrapper sont :

- Patient.info
- Med.Tn

Pour explorer nos données, nous avons utilisé la méthode info () de Pandas en python

```
polymyalgiarheumaticaandgca      5001
depression                          4010
hipreplacement                      3000
irritablebowelsyndrome              3000
kneeproblems                        3000
menopause                           3000
mitrazapine                         2811
Name: Speciality, dtype: int64
```

FIGURE 6.2 – Information générale des données selon la spécialité

A partir de cette figure, on remarque que le fichier Excel contenant les données contient initialement 23822 lignes et 2 colonnes de type d'objet.

	Description	Speciality
0	below are links to various resources for thr p...	hipreplacement
1	hi im ryan year old male and in the next coup...	hipreplacement
2	i really dont want sedation unless i absolutel...	hipreplacement
3	im originally from england living in the nethe...	hipreplacement
4	hi i am new here and hoping to get some advice...	hipreplacement
...	...	...
23817	i had a discussion with him about us not being...	depression
23818	so to sum this entire situation up i have one ...	depression
23819	so i recently started taking of celexa in the...	depression
23820	i feel so alone even in a room full of family ...	depression
23821	just wondered if anyone had experienced delaye...	depression

23822 rows × 2 columns

FIGURE 6.3 – Information générale des données

## 6.5 Représentation des données

La représentation des données est l'une des étapes de la phase de compréhension des données qui est utilisée pour cerner les variables prédictives et cibles. Cette étape est très importante afin de choisir la méthode de travail à suivre et le modèle approprié.

- Variables cibles :

Les variables cibles correspondent aux variables que nous voulons prédire à partir des variables prédictives. Dans notre cas c'est la colonne de specialist qu'on veut prédire.

- Variables prédictives :

La science des données considère les variables prédictives ou indépendantes comme des variables caractéristiques. Dans notre projet c'est la variable description.

## 6.6 Modélisation

Après avoir effectué un prétraitement de données en les nettoyant et éliminant les bruits, nous allons développer un ensemble des modèles de classification textuelle et nous allons procéder à leurs évaluations. Avant de poursuivre leur construction, nous devons choisir la technique utilisée. La prochaine partie sera consacrée au détail de la procédure adoptée pendant la sélection. En effet, suite à nos recherches les techniques les plus utilisées dans l'apprentissage automatique (Machine Learning) sont : Naïve Bayes, Logistic Regression, Support Vector Machines(SVM) et Arbre de Décision. Nous avons donc adopté les quatre techniques sur nos données

d'entrée afin de construire chaque fois un modèle qui sera ensuite évalué.

## 6.7 Construction des modèles

Au cours de cette étape, nous définissons et expliquons les différents algorithmes de classification supervisés que nous utilisons pour le problème de classification des maladies.

- Naïve Bayes :

Naive Bayes est un algorithme populaire en Machine Learning. C'est un algorithme du Supervised Learning utilisé pour la classification. Il est particulièrement utile pour les problématiques de classification de texte. Un exemple d'utilisation du Naive Bayes est celui du filtre anti-spam.

- Logistic Regression :

La régression logistique est une technique de classification empruntée par l'apprentissage automatique au domaine des statistiques. La régression logistique est une méthode statistique d'analyse d'un ensemble de données dans lequel il existe une ou plusieurs variables indépendantes qui déterminent un résultat. L'intention derrière l'utilisation de la régression logistique est de trouver le modèle le mieux adapté pour décrire la relation entre la variable dépendante et la variable indépendante.

- Support Vector Machines(SVM) :

L'objectif de l'algorithme de la machine à vecteurs de support est de trouver un hyperplan dans un espace à N dimensions (N - le nombre de caractéristiques) qui classe distinctement les points de données. Pour séparer les deux classes de points de données, il existe de nombreux hyperplans possibles qui pourraient être choisis. Notre objectif est de trouver un plan qui a la marge maximale, c'est-à-dire la distance maximale entre les points de données des deux classes. La maximisation de la distance de marge fournit un certain renforcement afin que les futurs points de données puissent être classés avec plus de confiance.

- Arbre de Décision :

L'arbre de décision est l'un des premiers algorithmes de la Machine Learning que les Data Scientists apprennent au cours de leur formation. Il est utilisé pour représenter visuellement et explicitement les décisions et la prise de décision pour des problèmes de classification ainsi que pour des problèmes de régression. Il représente aussi l'élément de base de plusieurs modèles comme le Random Forest ou XGBoost.

## 6.8 Evaluation

Après avoir présenté les modèles utilisés nous procédons à évaluer chaque modèle construit en se basant sur les techniques d'évaluation suivantes.

## 6.9 Techniques d'évaluation

La matrice de confusion :

La matrice de confusion est utilisée pour illustrer les performances du modèle en se basant sur les métriques suivantes :

- Vrai Positif « TP » : présente les cas positifs correctement prédits
- Vrai Négatif « TN » : présente les cas négatifs correctement prédits
- Faux Positif « PF » : présente les cas négatifs mal classés
- Faux Négatif « FN » : présente les cas positifs mal classés

Confusion matrix		Reality	
		Negative : 0	Positive : 1
Prediction	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

FIGURE 6.4 – Structure de la matrice de confusion

- Le rapport de classification :

Nous pouvons dégager le rapport de classification contenant les métriques nécessaires pour évaluer la performance du modèle à partir de la matrice de confusion. Dans ce qui suit nous avons défini ces différentes métriques :

- Accuracy : est le pourcentage de données correctement prédits sur tous les points de données. Plus formellement, il est déterminé par :  $\text{Accuracy} = (\text{TP}+\text{TN})/(\text{TP}+\text{TN}+\text{FP}+\text{FN})$
- Précision : est la mesure du nombre de prédictions positives faites qui sont correctes (VP), la formule pour cela est :  $\text{Precision} = \text{TP}/(\text{TP}+\text{FP})$
- Rappel ou Recall : est la mesure du nombre des cas classés comme positifs, parmi le nombre total d'exemples positifs, déterminé par :  $\text{Recall} = \text{TP}/(\text{TP}+\text{FN})$
- F1-score : c'est la combinaison entre les deux métriques Précision et Rappel, il est généré par la formule suivante :  $\text{Recall} = \text{TP}/(\text{TP}+\text{FN})$

## 6.10 Evaluation des modèles

Pour évaluer nos modèles, nous choisissons donc d'utiliser la matrice de confusion présentée ci-dessus. Toutes les performances des modèles obtenus sont présentées dans le tableau suivant.

TABLE 6.1 – Les performances des modèles

Modèles	Precision	Recall	F1-Score	Accuracy
Naïve Bayes	0.87	0.87	0.87	0.86
Logistic Regression	0.88	0.87	0.88	0.87
SVM	0.86	0.86	0.86	0.86
Arbre de décision	0.79	0.79	0.79	0.78

Après l'analyse du tableau , nous constatons que les résultats sont presque proches. De plus, nous avons remarqué que le modèle de régression logistique est le meilleur résultat avec une valeur d'exactitude respectivement de 0.88.

Nous avons donc décidé d'utiliser ce dernier comme modèle de classification des documents.

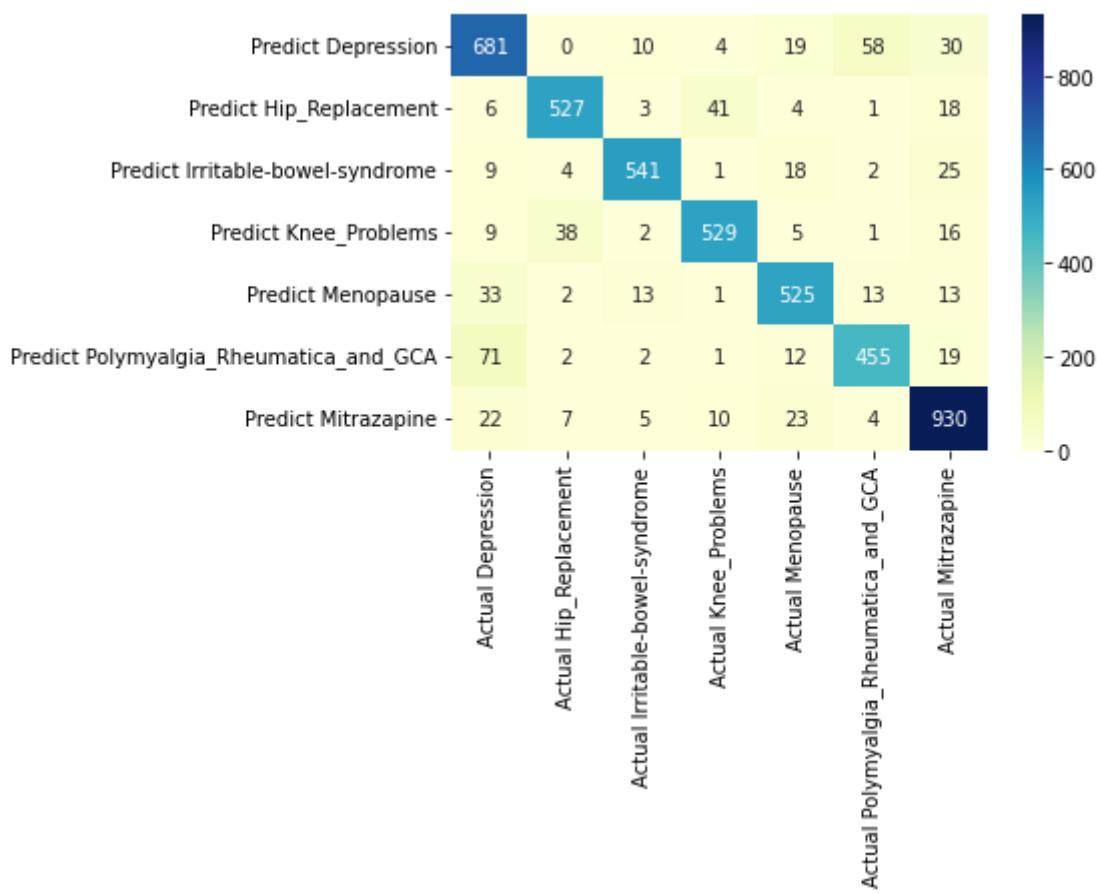


FIGURE 6.5 – Matrice de confusion du modèle régression logistique

## Conclusion

Dans ce chapitre, nous avons conçu nos modèles pour évaluer et sélectionner le meilleur de ces modèles pour utilisation dans le monde réel. À ce stade, nous avons terminé avec succès de validé notre application afin de livrer un produit conforme aux exigences. Ce chapitre est terminé, nous allons maintenant passer à la conclusion générale.

# Conclusion générale

**A** la fin de ce stage, nous souhaitons avoir réalisé un travail sérieux et convaincant sans la prétention qu'il soit parfait.

L'objectif de ce projet réside, essentiellement, dans l'élaboration d'une application mobile « Tabibi » au sein de la société "planetconnectus" permettant d'aider les patients à chercher des médecins qui peuvent traiter leurs maladies.

C'est une application intelligente. Notre projet de fin d'études, intitulé « Classification des textes de description des patients selon les maladies » qui consiste à trouver des médecins en fonction des maladies des futurs patients. En fait, c'est une expérience très prépondérante par le biais de laquelle nous avons pu passer par toutes les étapes de réalisation d'une application informatique. Autrement dit de la compréhension du métier au développement en passant par la phase d'analyse et de conception.

De plus, ce projet nous a permis non seulement de mettre en œuvre les connaissances acquises durant nos trois années d'études à L'École supérieure privée d'ingénierie et de technologie(ESPRIT) mais aussi de maîtriser le développement Mobile, de découvrir et de nous former sur les nouvelles technologies respectives.

Ajoutons également, que ce stage nous a offert l'opportunité de nous intégrer dans l'environnement de l'entreprise et d'améliorer nos capacités à la vie professionnelle. Il nous a aussi permis de rehausser nos connaissances et notre formation théorique et pratique acquise tout au long de nos trois années de formation.

# Webographies

1. <https://flutter.dev/>.
2. <https://code.visualstudio.com/>.
3. <https://www.postman.com/>.
4. <https://www.mongodb.com/>.
5. <https://nodejs.org/en/>.
6. <https://flask.palletsprojects.com/en/2.2.x/>.
7. <https://www.overleaf.com/>.
8. <https://www.appventurez.com/blog/introduction-mvvm-architecture-flutter>
9. <https://mrmint.fr/naive-bayes-classifier>
10. <https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592>
11. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
12. <https://blent.ai/arbres-de-decision-en-machine-learning/>