



## Filière : Sciences Mathématiques et Informatiques

### Projet de fin étude Semestre S6

### Mémoire

**Intitulé : Application mobile de prise des rendez-vous pour les consultations et le suivi médical à distance des patients**

Présenté par

**BEN KHALI Doha**

**AZAMI Youssef**

Encadrant : Pr. Khalid ALAOUI ZIDANI



Soutenu le 28/05/2022

Jury :

Examineur 1 : Pr. Nouredine EN NAHNAHI

Examineur 2 : Pr. Khalid EL FAHSSI

Encadrant : Pr. Khalid ALAOUI ZIDANI

*Année Universitaire : 2021/2022*

**Filière : Sciences Mathématiques et Informatiques**

**Projet de fin étude  
Semestre S6**

**Mémoire**

**Intitulé : Application mobile de prise des rendez-vous pour les consultations et le suivi médical à distance des patients**

Présenté par

**BEN KHALI Doha**

**AZAMI Youssef**

Encadrant : Pr. Khalid ALAOUI ZIDANI

Soutenu le 28/05/2022

Jury :

Examineur 1 : Pr. Nouredine EN NAHNAHI

Examineur 2 : Pr. Khalid EL FAHSSI

Encadrant : Pr. Khalid ALAOUI ZIDANI

*Année Universitaire : 2021/2022*

---

## Remerciement

Nous tenons à exprimer nos sincères remerciements à tous les Professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

En seconde lieu, nous tenons à remercier notre encadrant monsieur Khalid ALAOUI ZIDANI, et monsieur Nouredine EN NAHNAHI pour l'orientation, les conseils et les remarques pertinentes, et tous ceux qui ont contribué à réussir notre mémoire.

Merci à l'équipe pédagogique de notre faculté des mathématiques et d'informatique pour avoir répondu à nos questions tout au long de ces trois années.

Nos vifs remerciements vont également aux Membres du Jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Ainsi, nous nous adressons nos remerciements les plus chaleureux à toutes les personnes qui nous ont aidés de près ou de loin par le fruit de leur connaissance pendant toute la durée de notre parcours éducatif.

---

## Résumé

La crise sanitaire qu'on a vécu ces deux dernières années à cause de la pandémie COVID-19, nous a obligé de penser qu'il faut profiter de toutes les technologies et outils afin d'assurer des services à distance, surtout lorsque le problème est lié à la santé des individus.

Le suivi des patients à distance est devenu de plus en plus un défi pour protéger la vie des gens et de faciliter que possible la période du traitement d'un patient, justement notre travail s'articule dans le cadre d'un projet de fin d'études visant à développer une application mobile sous Android et iOS, et qui a pour rôle la bonne gestion de santé et un bon suivi médical.

Cette application permet aux professionnels de santé (médecins) d'introduire les informations médicales de leurs patients, pour faciliter la tâche et la sécuriser, et aux patients de consulter et de vérifier leurs dossiers médicaux à tout moment en tous lieux avec facilité et confiance dans le but d'assurer le plus haut degré de suivi médical de l'état de santé de ces derniers, et aussi de garantir la bonne communication et interaction entre le médecin et leurs patients.

---

# Table des matières

<b>Introduction</b>	<b>11</b>
<b>1 Chapitre 1. Analyse et étude de cahier de charges</b>	<b>12</b>
1.1 Introduction . . . . .	13
1.2 Étude et Analyse de cahier de charges . . . . .	13
1.2.1 Problématique . . . . .	13
1.2.2 Objectifs . . . . .	13
1.2.3 Etude de faisabilité . . . . .	14
1.2.4 La solution proposée . . . . .	14
1.2.5 Avantages de notre solution . . . . .	14
1.2.6 Spécification des besoins . . . . .	15
1.2.7 Analyse de l'existant . . . . .	16
1.2.8 Qualité du projet . . . . .	17
1.2.9 Planification . . . . .	18
1.3 Conclusion . . . . .	19
<b>2 Chapitre 2. Conception et modélisation</b>	<b>20</b>
2.1 Introduction . . . . .	21

2.2	UML . . . . .	21
2.2.1	Intérêt de la modélisation . . . . .	21
2.2.2	Présentation du langage UML . . . . .	21
2.2.3	Outil de modélisation . . . . .	23
2.3	Les acteurs . . . . .	24
2.4	Diagramme de cas d'utilisation . . . . .	24
2.5	Diagramme de séquence . . . . .	40
2.6	Diagramme de Classe . . . . .	53
2.7	Conclusion . . . . .	54
<b>3</b>	<b>CHAPITRE 3 : Les technologies utilisées et réalisation</b>	<b>55</b>
3.1	Introduction . . . . .	56
3.2	Environnement de travail . . . . .	56
3.2.1	Environnement matériel . . . . .	56
3.2.2	Environnement logiciel . . . . .	56
3.2.3	La Base De Données . . . . .	58
3.2.4	Choix des Technique . . . . .	58
3.3	Les interfaces graphiques . . . . .	67
3.3.1	Application « Ma santé » en action . . . . .	68
3.4	Conclusion . . . . .	73
	<b>Conclusion Générale</b>	<b>74</b>
	<b>Annexe</b>	<b>75</b>

**Webographie**

**76**

---

## Liste des tableaux

2.1	« création de compte patient/médecin»	28
2.2	« authentification ».	29
2.3	« optimiser les informations ».	31
2.4	« prise des rendez-vous ».	32
2.5	« choix de médecin ».	33
2.6	« communication avec médecin ».	34
2.7	« Evaluation».	35
2.8	« contacter mes patients »	36
2.9	« gestion des rendez-vous ».	37
2.10	« optimiser le temps de disponibilité ».	38
2.11	« gestion des comptes utilisateurs ».	40
3.1	Environnement matériel	56



---

## Table des figures

1.1	Cycle de vie en V. . . . .	18
1.2	Diagramme de Gantt. . . . .	19
2.1	logo UML. . . . .	21
2.2	les différents diagrammes d'UML. . . . .	22
2.3	Enterprise Architect. . . . .	23
2.4	Interface d'Enterprise Architect. . . . .	24
2.5	Diagramme de cas d'utilisation. . . . .	26
2.6	cas d'utilisation « création de compte patient ». . . . .	27
2.7	cas d'utilisation « création de compte médecin ». . . . .	27
2.8	cas d'utilisation « Authentification ». . . . .	28
2.9	cas d'utilisation « optimiser les informations du médecin ». . . . .	30
2.10	cas d'utilisation « optimiser les informations du patient ». . . . .	30
2.11	cas d'utilisation « prise des rendez-vous ». . . . .	31
2.12	cas d'utilisation « choix de médecin ». . . . .	32
2.13	cas d'utilisation « communication avec un médecin ». . . . .	33
2.14	cas d'utilisation « évaluation». . . . .	34

2.15 cas d'utilisation « contacter mes patients ».	35
2.16 cas d'utilisation « gestion des rendez-vous ».	36
2.17 cas d'utilisation « optimiser le temps de disponibilité ».	37
2.18 cas d'utilisation « gestion des comptes utilisateurs ».	39
2.19 séquence de la « création de compte médecin ».	41
2.20 séquence de la « création du compte patient ».	42
2.21 séquence de « L'authentification ».	43
2.22 séquence de « choix de médecin ».	44
2.23 séquence de la « prise d'un rendez-vous ».	45
2.24 séquence de la « communication avec médecin ».	46
2.25 séquence de la « Evaluation ».	47
2.26 séquence de la « contacter mes patients ».	48
2.27 séquence de la « gestion des rendez-vous ».	49
2.28 séquence de la « optimiser les informations ».	50
2.29 séquence de la « optimiser le temps de disponibilité ».	51
2.30 séquence de la « gestion des comptes utilisateurs ».	52
2.31 diagramme de classe.	53
3.1 « la base de données ».	66
3.2 Interfaces de réception.	68
3.3 Interfaces de la connexion et la création des comptes utilisateurs.	69
3.4 Interfaces des fonctionnalités de l'espace patient.	70
3.5 Interface de la prise d'un rendez-vous.	71
3.6 Interfaces des fonctionnalités de l'espace médecin.	72

3.7 Interfaces de la mise à jour de temps de disponibilité. . . . .	73
---------------------------------------------------------------------	----

---

# INTRODUCTION GÉNÉRALE

Actuellement, le monde connaît un avancé technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information, elle joue un rôle important dans le quotidien de l'être humain.

Les smartphones sont devenus de véritables objets du quotidien notamment pour les patients (malades) et les professionnels de santé, gratuites ou payantes, les applications mobiles touchent aujourd'hui tous les domaines de la vie courante : l'actualité, l'éducation, les finances, la santé ou encore les loisirs et les divertissements. C'est pourquoi on trouve de plus en plus d'applications mobiles variées et utiles destinées aux patients et professionnels de santé qui répondent à de nombreux besoins.

Dans ce cadre méthodologique, nous sommes tenus à réaliser le projet de fin d'études qui a pour but de développer une application mobile qui contribue à un rapprochement entre les professionnels et les patients avec l'objectif d'impliquer ces derniers d'une manière plus directe dans tout ce qui concerne leur santé. En utilisant le Framework Flutter.

Par le présent rapport, nous allons essayer de décrire les différentes étapes suivies pour réaliser notre projet ;

On présente dans le premier chapitre le contexte du notre travail dans lequel nous présentons, la problématique de notre sujet, ensuite nous déterminons les différents objectifs et l'étude de faisabilité de l'application, puis nous proposons une solution de ce problème. Finalement, nous traitons la spécification des besoins et les avantages de notre solution. Le deuxième chapitre est dédié à l'analyse et la conception de notre projet suivant le processus UP d'UML (Unified Modeling Language).

Enfin, le troisième chapitre qui décrit le projet avec la présentation des choix technique de réalisation. Celui-ci est consacré à la présentation de l'environnement matériel et logiciel utilisé.

# Chapitre 1. Analyse et étude de cahier de charges

## Sommaire

<b>1.1</b>	<b>Introduction</b>	<b>13</b>
<b>1.2</b>	<b>Étude et Analyse de cahier de charges</b>	<b>13</b>
1.2.1	Problématique	13
1.2.2	Objectifs	13
1.2.3	Etude de faisabilité	14
1.2.4	La solution proposée	14
1.2.5	Avantages de notre solution	14
1.2.6	Spécification des besoins	15
1.2.7	Analyse de l'existant	16
1.2.8	Qualité du projet	17
1.2.9	Planification	18
<b>1.3</b>	<b>Conclusion</b>	<b>19</b>

## **1.1 Introduction**

Le cahier de charges constitue le cœur de la phase d'analyse d'un projet et de la détermination des besoins. Cette étape est primordiale pour la mise en route de tout projet informatique ou autre, et qui permet de définir le contexte de fonctionnement, d'assurer la communication entre les acteurs du projet. Il sert à présenter le projet, la problématique, la planification, et définir les objectifs que doit atteindre la solution.

## **1.2 Étude et Analyse de cahier de charges**

### **1.2.1 Problématique**

Les patients souffrent souvent de manque de communication avec les professionnels de santé. La prise des rendez-vous, la recherche et le choix des médecins, et les demandes d'informations sont parmi les soucis des citoyens à l'égard des services de santé. Le développement d'une application de communication entre le patient et le prestataire de soins présente une solution bénéfique aussi bien pour le patient que le professionnel de santé afin de se communiquer et échanger les informations.

### **1.2.2 Objectifs**

Ma Santé est une application pour la Gestion des communications entre patient, et tout fonctionnaire de santé en mettant en disposition un espace de communication libre qui permet d'effectuer à tout moment une démarche médicale à distance.

L'application permet :

☐ Pour les professionnels de santé :

- \* Suivi médical des patients.
- \* Consultation des patients à distance.
- \* Optimisation de Créneau de disponibilité en toute simplicité et rapidité.

☐ Pour les patients :

- \* Garantir une meilleur gestion de santé.
- \* Offrir un bon suivi médical de l'état des patients.
- \* Assurer une bonne communication et interaction avec les médecins.

### 1.2.3 Etude de faisabilité

D'après l'analyse du cahier de charges et les différentes réunions avec le maître d'ouvrage MOA, nous sortons par deux besoins afin de réaliser ce projet :

- ☐ **Coté matériel** : Notre application a besoin d'un serveur dans lequel se trouve une base de données contenant toutes les informations et un ordinateur.
- ☐ **Coté développement** : On a besoin des logiciels pour développer l'application mobile, et les scripts de communications avec le serveur.

Donc l'application demandée est faisable, mais avec des petites modifications et un grand travail dans le coté conceptuel.

Afin de réaliser ce projet, la période de 2 mois apparait raisonnable et suffisante pour le terminer.

Après avoir bien préciser une cible pour notre projet, une réalisation de ce dernier peut aboutir à un grand succès dans le domaine médical comme il peut aboutir à une grande perte des ressources et de temps. Ainsi notre objectif est de bien trouver un environnement facilitant toute interaction entre médecin et ces patients ce qui est assuré par notre application.

### 1.2.4 La solution proposée

Notre solution consiste à réaliser une application mobile Android et iOS multi -espaces pour les patients et les médecins.

- ☐ **Espace patient** : un espace permettant au patient de prendre des rendez- vous en ligne, effectuer des consultations et obtenir des avis médicaux d'une manière rapide et confidentielle.
- ☐ **Espace médecin** : un espace dédié aux médecins multi spécialistes et qui permet une gestion d'horaire organisationnelle complète des rendez-vous, et d'effectuer des consultations à distance.

### 1.2.5 Avantages de notre solution

La solution proposée (application Ma Santé) répond à maints problèmes affrontés par les cadres médicaux et leurs patients, comme elle assure toutes les fonctionnalités indiquées dans notre cahier de charge. Parmi ces avantages :

- \* Gestion des rendez- vous prise par le patient à distance.
- \* Elargissement du principe de traitement des patients en ligne.
- \* Garantir un espace facile et rapide en accès pour tout patient en cas d'urgence.
- \* Interaction directe avec les autres professionnels de santé.
- \* Réponse aux problèmes de mobilité des citoyens en recherche de l'hospitalisation.

### 1.2.6 Spécification des besoins

La spécification des besoins constitue la phase de départ de toute application à développer, dans laquelle nous allons identifier les besoins de notre application.

Nous distinguons deux types de besoins.

#### **Besoins fonctionnels :**

Il s'agit des fonctionnalités du système, ce sont les besoins spécifiant un comportement d'entrée sortie du système.

##### ☐ Espace client :

- \* L'authentification du patient.
- \* Création d'un compte.
- \* Choix de médecin.
- \* Prise de rendez-vous.
- \* Communication avec médecin.
- \* Noter un médecin.

##### ☐ Espace médecin :

- \* L'authentification des médecins.
- \* Création d'un compte.
- \* Mis à jour de temp de disponibilité.
- \* Optimiser les informations.
- \* Contacter les patients.

#### **Besoins non fonctionnels :**



Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. En ce qui concerne nos applications, nous avons dégagé les besoins suivants :

- \* La disponibilité : l'application doit être disponible pour être utilisée par n'importe quel utilisateur.
- \* La sécurité : l'application devra être hautement sécurisée, les informations de tout patient doivent être confidentielles et uniquement accessibles par le client et son fonctionnaire médical.
- \* La fiabilité : les données fournies par l'application doivent être fiables.
- \* Facilité d'utilisation : l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur.
- \* Extensibilité : l'application peut être améliorée par l'ajout d'autres fonctionnalités pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.
- \* La performance : l'application devra être performante c'est-à-dire que le système doit réagir dans un délai précis quelque soit l'action de l'utilisateur.

## **1.2.7 Analyse de l'existant**

### **1.2.7.1 Définition**

Afin d'approfondir notre compréhension du sujet et avoir une idée plus claire sur notre projet et ses fonctions attendues, nous avons mené une étude sur deux applications qui s'inscrivent dans le même cadre que notre travail.

Nous analysons dans un premier temps l'application de « Voiladoc patient » ; et ensuite le site « Dabadoc».

### **1.2.7.2 Analyse de l'application « Voiladoc patient »**

Voiladoc est une plateforme de téléconsultation et de rendez-vous médicaux en ligne pour téléphones mobiles qui connecte les patients à des professionnels de santé.

Cette application offre plusieurs services comme :  
La prise des rendez-vous avec un médecin généraliste ou spécialiste pour une téléconsultation

immédiate ou différée, pour une visite au cabinet médical / hôpital / clinique, ou pour une visite à domicile d'un professionnel de santé (médecin, infirmier(ère), physiothérapeute, sage-femme) [1].

### 1.2.7.3 Analyse de site « Dabadoc »

DabaDoc est une plateforme de gestion de rendez-vous médicaux en ligne . Cette plateforme garantit un changement quotidien des médecins et celui de leurs patients. Il optimise la gestion des cabinets médicaux grâce à ses multiples fonctionnalités intégrées et permet aux patients de prendre rendez-vous en quelques clics 24h/7j gratuitement. Fondé en 2014, DabaDoc a pour mission de simplifier le processus de recherche de médecins, garantir aux patients une meilleure expérience de prise de rendez-vous et permettre aux médecins une meilleure gestion de leurs rendez-vous.

Ce site web offre plusieurs services comme :  
permettre aux utilisateurs de trouver le médecin ou le dentiste qui les convient, et de prendre rendez-vous directement en ligne. ainsi qu'il gère la disponibilité des médecins et organise les différentes tâches de ces derniers dans leurs cabinets [2].

### 1.2.8 Qualité du projet

Choix de cycle de vie :

Le cycle en V est un modèle de gestion de projet qui implique toutes les étapes du cycle de vie d'un projet : conception, réalisation, validation.

On a choisi ce cycle pour ces nombreux avantages :

Le principal avantage du cycle en V est qu'il évite de revenir en arrière incessamment pour redéfinir les spécifications initiales, comme un cliquet. Chaque phase de conception demande la rédaction d'une documentation précise et exhaustive, où chaque point doit être validé par le produit final. Dès lors qu'une étape est validée, on ne revient pas en arrière et on passe à l'étape suivante sur une base solide ; c'est la principale force du cycle en V [3, 4].

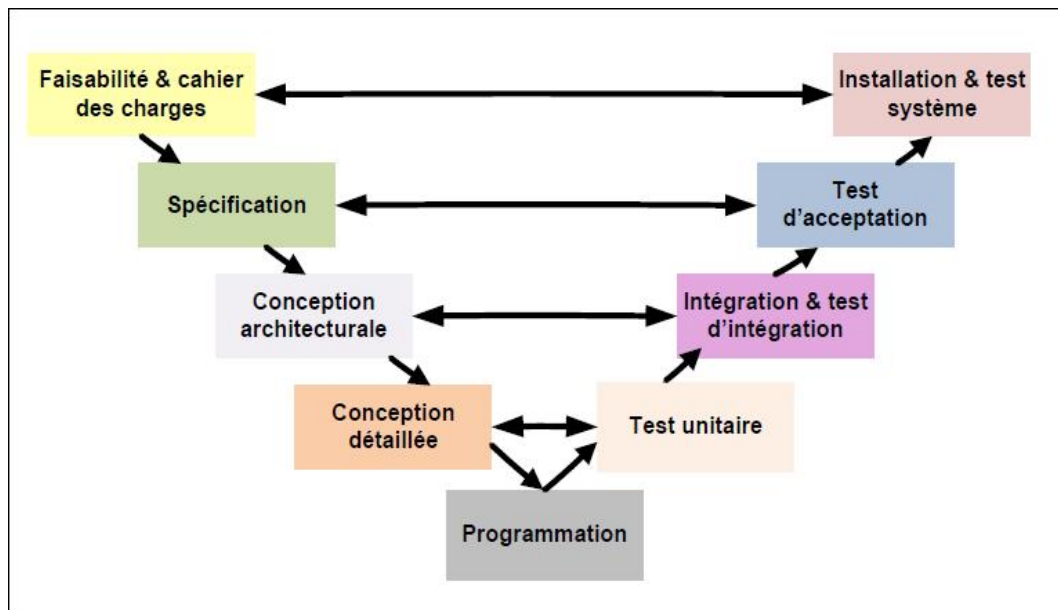


FIGURE 1.1 – Cycle de vie en V.

### 1.2.9 Planification

Pour mieux planifier, analyser et contrôler le bon déroulement de la réalisation du projet, ainsi qu'assurer une bonne qualité du produit dans des délais fixés et une conformité entre ce qui est défini et ce qui est obtenu, nous avons utilisé les notions de la gestion des projets informatiques et le respect des normes qualités. Dans ce cadre nous allons présenter le diagramme de GANTT.

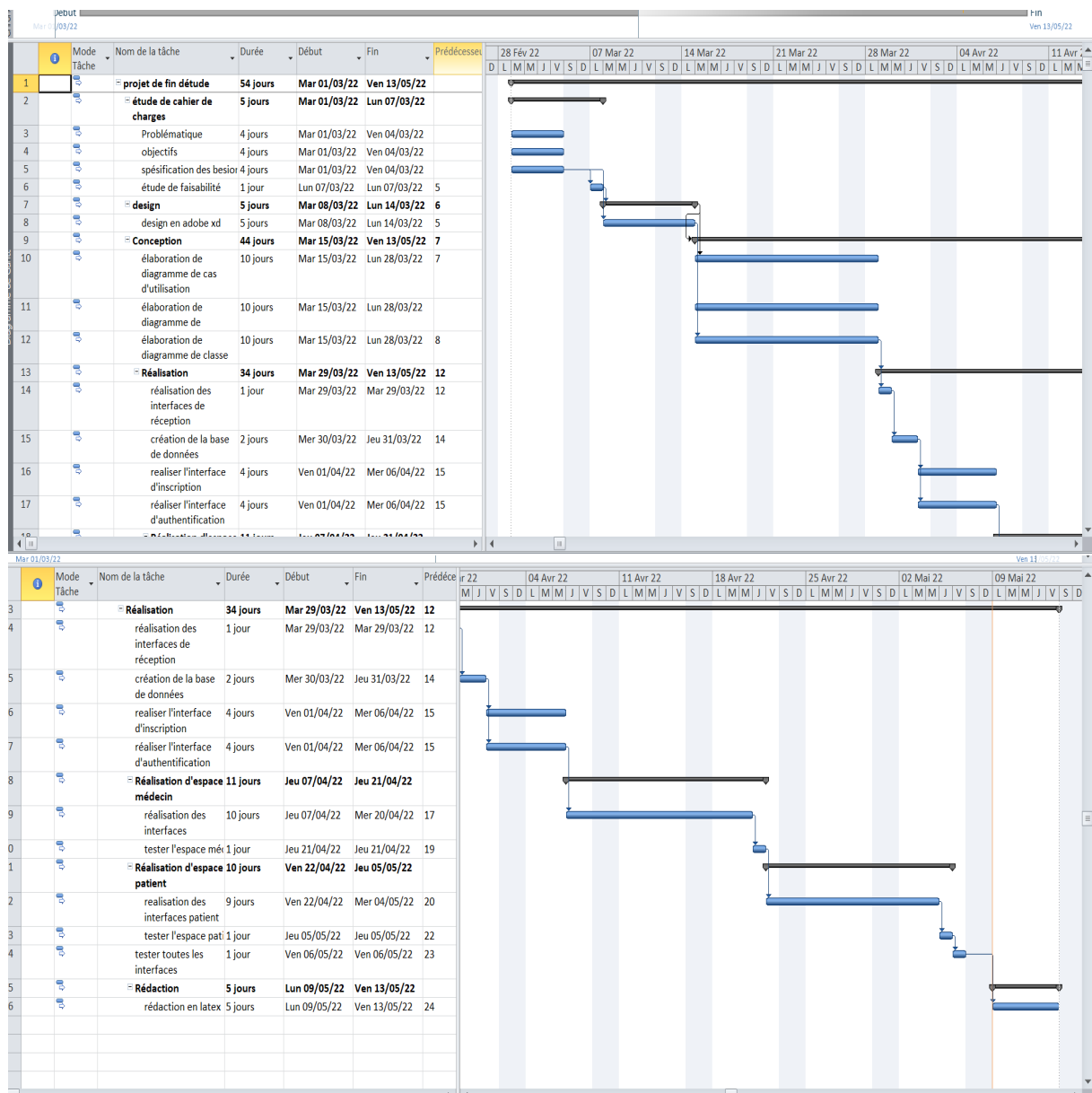


FIGURE 1.2 – Diagramme de Gantt.

## 1.3 Conclusion

Au cours de ce chapitre, nous avons présenté l'étude préliminaire du projet en présentant la problématique, les objectifs, l'étude de faisabilité, la solution proposée et ses avantages, la spécification des besoins, la qualité de projet et la planification. Le chapitre suivant sera dédié pour la phase d'analyse et de conception.

## Chapitre 2. Conception et modélisation

### Sommaire

<b>2.1</b>	<b>Introduction</b>	<b>21</b>
<b>2.2</b>	<b>UML</b>	<b>21</b>
2.2.1	Intérêt de la modélisation	21
2.2.2	Présentation du langage UML	21
2.2.3	Outil de modélisation	23
<b>2.3</b>	<b>Les acteurs</b>	<b>24</b>
<b>2.4</b>	<b>Diagramme de cas d'utilisation</b>	<b>24</b>
<b>2.5</b>	<b>Diagramme de séquence</b>	<b>40</b>
<b>2.6</b>	<b>Diagramme de Classe</b>	<b>53</b>
<b>2.7</b>	<b>Conclusion</b>	<b>54</b>

## 2.1 Introduction

Comme n'importe quel type de projet, un projet informatique nécessite une phase d'analyse, suivie d'une étape de conception.

Afin d'aboutir à une meilleure organisation et une bonne maîtrise du travail et donc arriver à déployer de meilleures applications, il est nécessaire de suivre une démarche méthodologique rigoureuse. Pour cela le choix d'une méthode d'analyse et de conception est d'une très grande importance.

## 2.2 UML

### 2.2.1 Intérêt de la modélisation

Modéliser, c'est décrire de manière visuelle et graphique les besoins et les solutions fonctionnelles et techniques de notre projet logiciel. Un logiciel qui a été réalisé sans analyse et sans conception (étapes où l'on modélise le futur logiciel) risque lui aussi de ne pas répondre aux besoins, de comporter des anomalies et d'être très difficile à maintenir [5].

### 2.2.2 Présentation du langage UML



FIGURE 2.1 – logo UML.

**UML ( Unified Modeling Langage )** : est un langage de modélisation graphique à base de pictogrammes, conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

UML est destiné à faciliter la conception des documents nécessaires au développement d'un logiciel orienté objet, comme standard de modélisation de l'architecture logicielle [6, 7].

L'UML est constitué de diagrammes qui servent à visualiser et décrire la structure et le comportement des objets qui se trouvent dans un système. Il permet de présenter des systèmes logiciels complexes de manière plus simple et compréhensible qu'avec du code informatique.

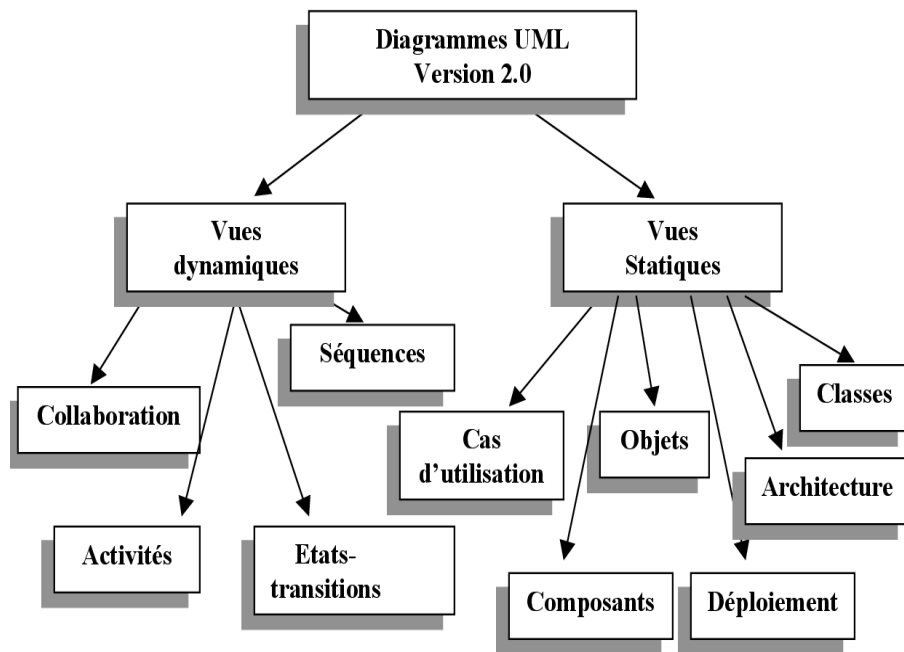


FIGURE 2.2 – les différents diagrammes d'UML.

### 2.2.3 Outil de modélisation

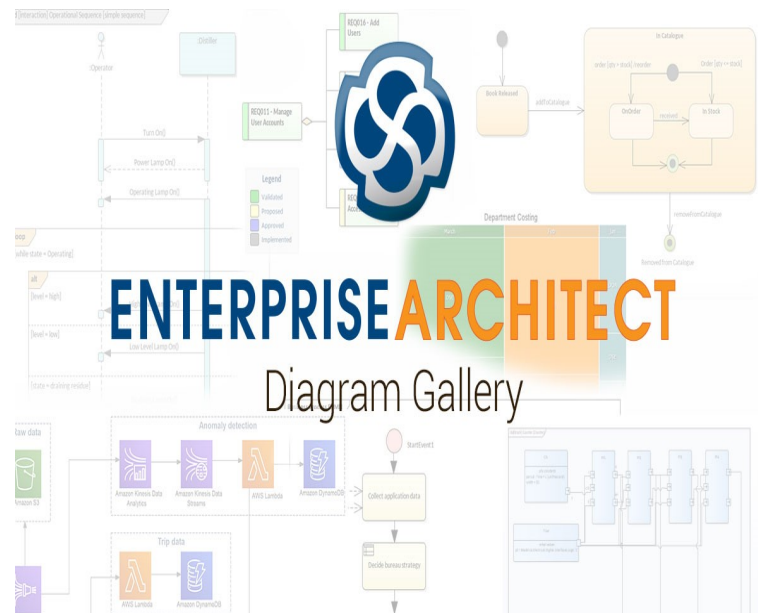


FIGURE 2.3 – Enterprise Architect.

Enterprise Architect est un outil multiposte, graphique conçu pour aider à construire des systèmes robustes et maintenables. Et en utilisant de haute qualité, les rapports incorporés et la documentation, vous pouvez offrir une vision vraiment partagée facilement et précisément [8].

#### Description d'interface

L'interface est simple et facile d'utilisation, les objets sont classés à gauche de l'écran par thèmes, un moteur de recherche vous permet d'interroger la base de données des Dessins ce qui vous permet une simple manipulation des différents objets utilisés par différents diagrammes de l'UML.



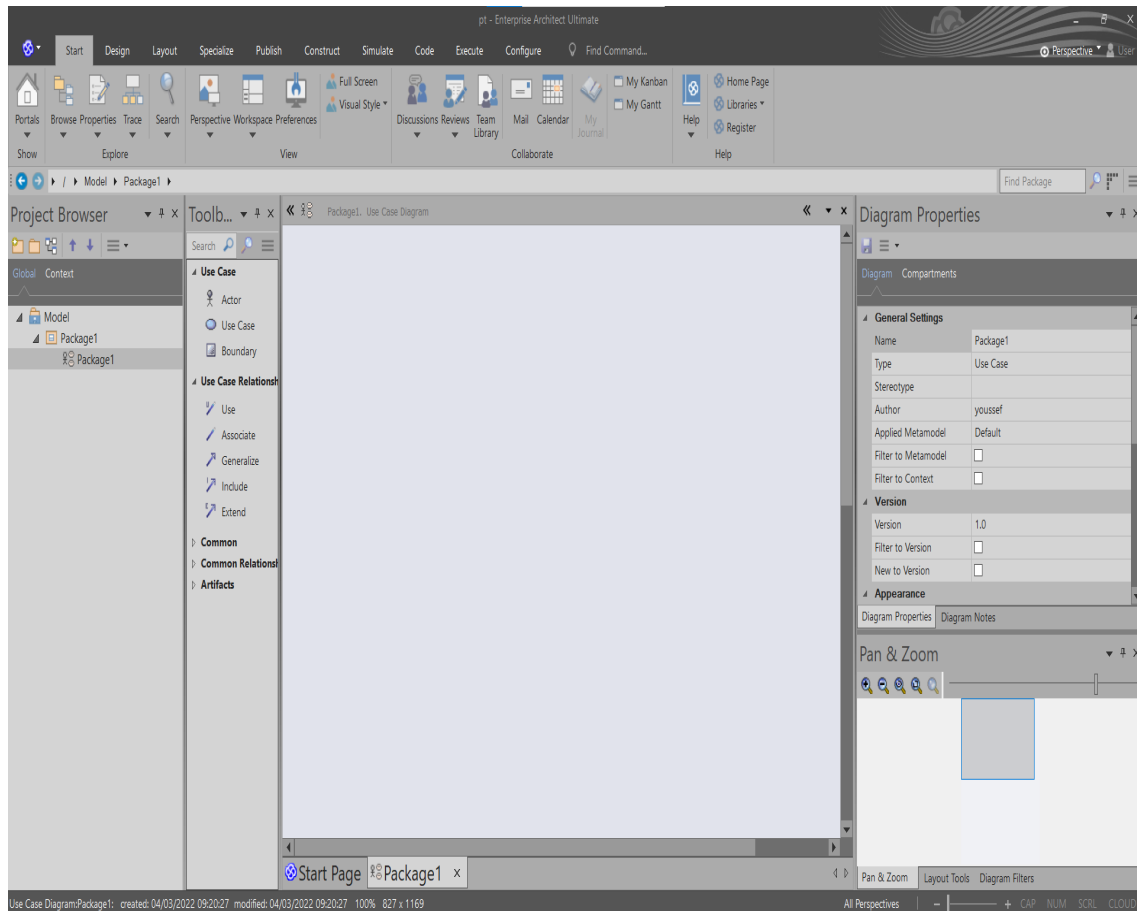


FIGURE 2.4 – Interface d'Enterprise Architect.

## 2.3 Les acteurs

- **Médecin** : visionne les rendez-vous, gère les consultations des patients.
- **Patient** : Personne qui subit une thérapie diagnostic, son rôle consiste à demander des rendez-vous afin de faire des consultations médicales à distance
- **Administrateur** : faire la gestion des comptes utilisateurs, possède la possibilité de supprimer ou désactiver un compte.

## 2.4 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de structurer les besoins des utilisateurs et les objectifs correspondants d'un système, il donne une vision globale du comportement fonctionnel d'un

système logiciel.

Dans les figures qui suivent, nous présenterons les cas d'utilisations qui mettent en évidence les principales fonctionnalités de chaque acteur dans le système [9].

# Diagramme de cas d'utilisation générale

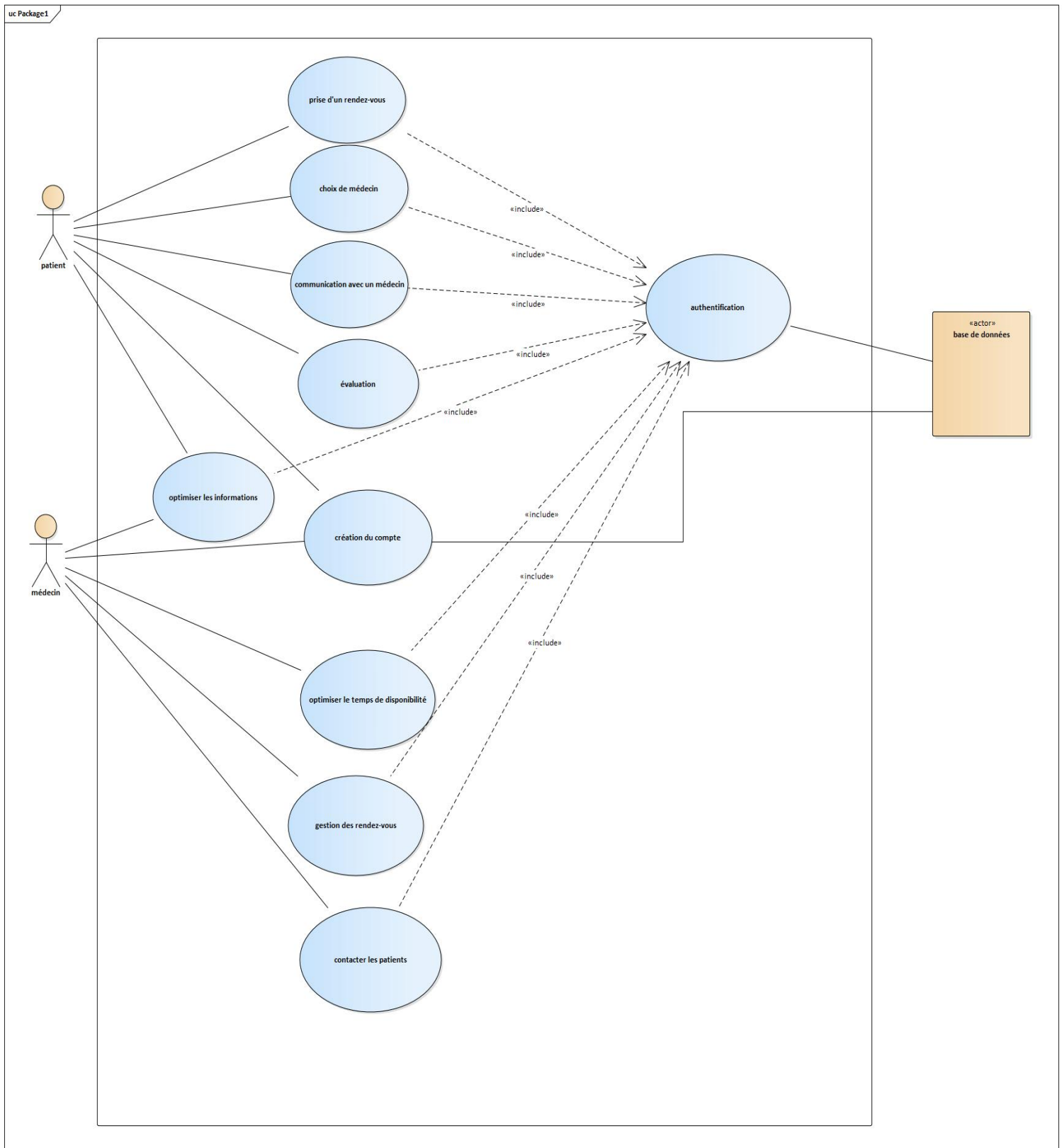


FIGURE 2.5 – Diagramme de cas d'utilisation.

## Cas d'utilisation « création de compte Patient »

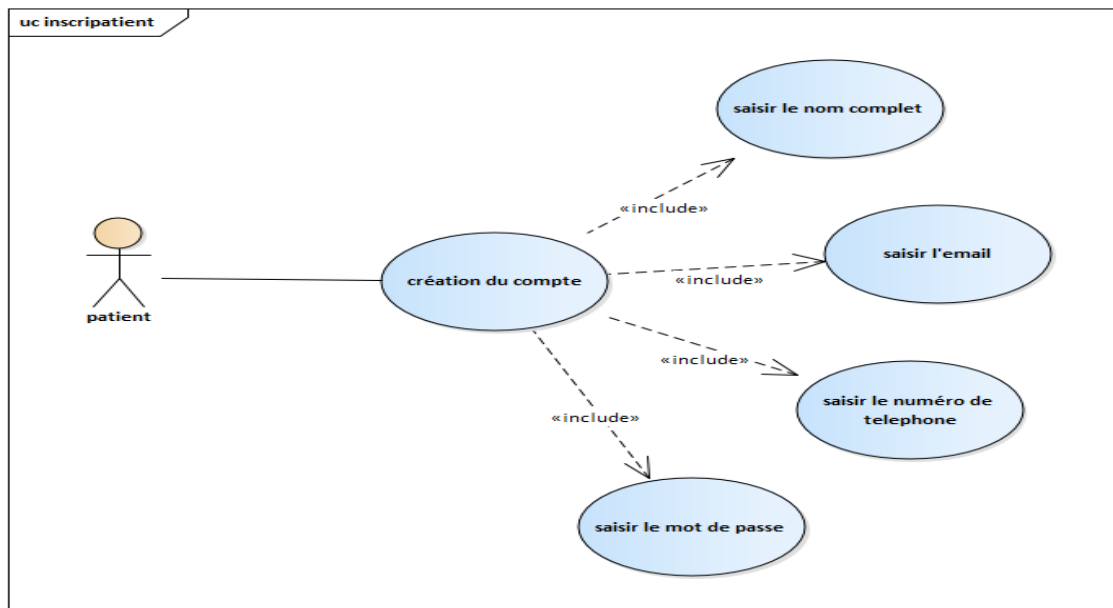


FIGURE 2.6 – cas d'utilisation « création de compte patient ».

## Cas d'utilisation « création de compte Médecin »

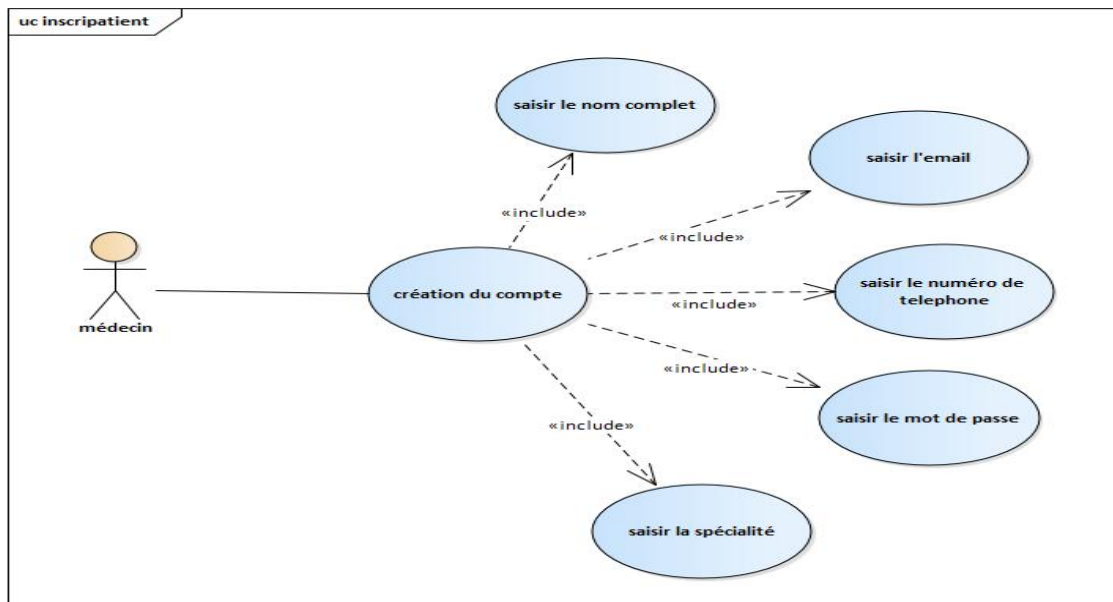


FIGURE 2.7 – cas d'utilisation « création de compte médecin ».

## Description du cas « création du compte patient/médecin »

SOMMAIRE D'IDENTIFICATION	
Titre	Création du compte.
But	La création d'un nouveau compte pour l'utilisateur.
Résumé	L'utilisateur choisit s'il est médecin ou patient puis créer un nouveau compte.
Acteur	Patient, médecin.
Description d enchainement.	
Post condition	La création d'un nouveau compte utilisateur et l'ajout de ce dernier à la base de données.
Scenario nominale	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur « vous êtes médecin » pour utilisateur médecin et sur « consulter en ligne » pour un patient.</li> <li>2. L'utilisateur remplit tous les champs qui sont affichés par le système.</li> <li>3. L'utilisateur clique sur le bouton « s'inscrire » pour l'envoi des informations.</li> <li>4. Le système ajoute les informations à la base de données.</li> <li>5. L'utilisateur va connecter à son espace.</li> </ol>
Scenario alternatif	<ol style="list-style-type: none"> <li>1. L'utilisateur n'a pas rempli un champ ou les données sont incorrectes ou invalides.</li> <li>2. Le système affiche un message d'erreur.</li> </ol>

TABLE 2.1 – « création de compte patient/médecin »

## Cas d'utilisation « authentification »

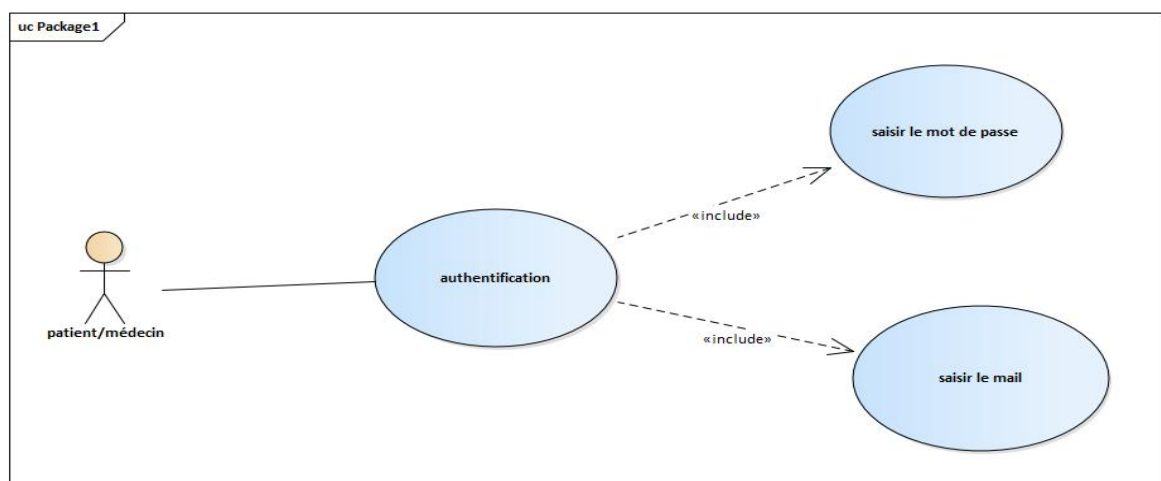


FIGURE 2.8 – cas d'utilisation « Authentification ».

## Description du cas d'utilisation « authentification »

SOMMAIRE D'IDENTIFICATION	
Titre	Authentification.
But	Avoir un accès à l'application.
Résumé	L'utilisateur clique sur «se connecter» pour se connecter à son espace personnel soit un patient soit un médecin.
Acteur	Le patient, le médecin.
Description d'enchaînement.	
Précondition	L'utilisateur avoir déjà un compte.
Scénario nominale	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur « vous êtes médecin » pour utilisateur médecin et sur « consulter en ligne » pour un patient.</li> <li>2. L'utilisateur clique sur « se connecter ».</li> <li>3. Le système affiche les champs à remplir pour la connexion.</li> <li>4. L'utilisateur remplit les champs qui sont affichés.</li> <li>5. L'utilisateur clique sur le bouton « confirmer »</li> <li>6. Le système vérifie si le compte existe.</li> <li>7. Le système affiche l'espace de cet utilisateur s'il existe.</li> </ol>
Scénario alternatif	<ol style="list-style-type: none"> <li>1. L'utilisateur n'a pas rempli un champ ou les données incorrectes.</li> <li>2. Le système affiche un message d'erreur.</li> <li>3. Retour à l'étape 1 du scénario nominal pour lancer à nouveau la connexion.</li> </ol>

TABLE 2.2 – « authentification ».

## Cas d'utilisation « optimiser les informations pour le médecin »

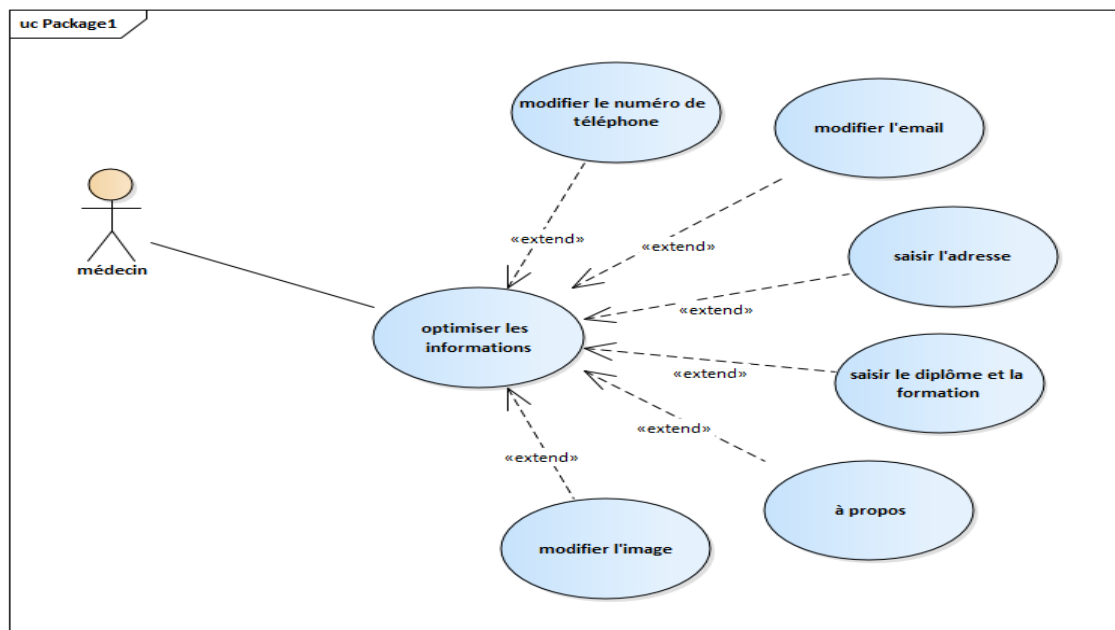


FIGURE 2.9 – cas d'utilisation « optimiser les informations du médecin ».

## Cas d'utilisation « optimiser les informations pour le patient »

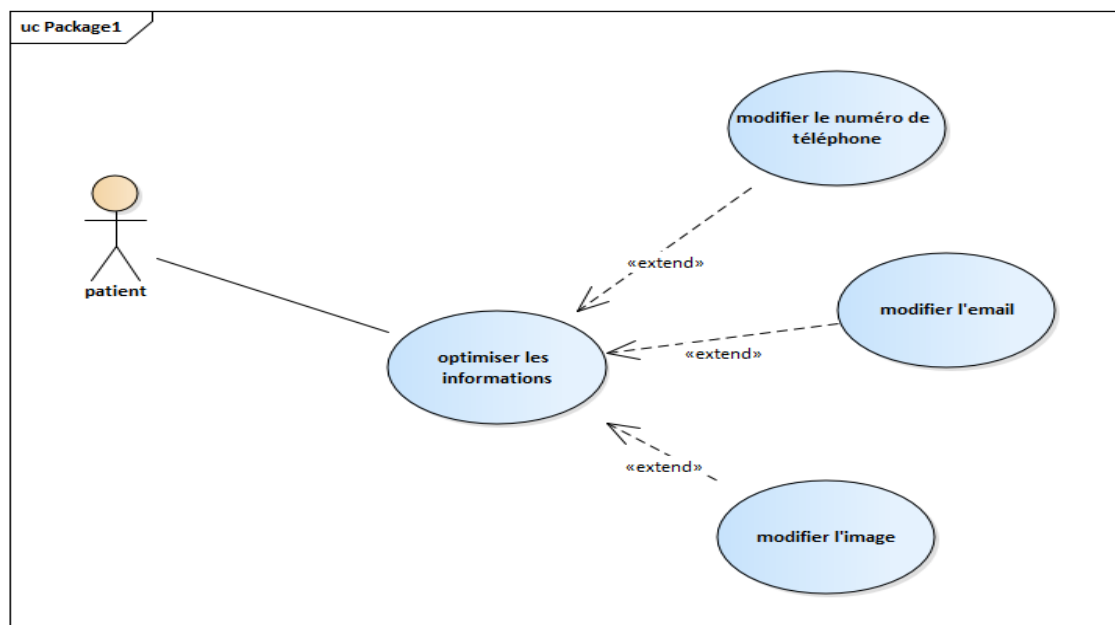


FIGURE 2.10 – cas d'utilisation « optimiser les informations du patient ».

Description du cas d'utilisation « optimiser les informations pour le médecin/le patient »

SOMMAIRE D'IDENTIFICATION	
Titre	Optimiser les informations.
But	Ajouter des informations au profil.
Résumé	L'utilisateur accède à son profil et remplit les champs vides.
Acteur	Le patient, le médecin.
Description d'enchaînement.	
Précondition	L'utilisateur a déjà un compte.
Scenario nominale	<ol style="list-style-type: none"> <li>1. Authentification.</li> <li>2. L'utilisateur se redirige vers son profil.</li> <li>3. Le système affiche les champs vides à remplir.</li> <li>4. L'utilisateur remplit les champs.</li> <li>5. Le système enregistre les informations.</li> </ol>
Scenario alternatif	

TABLE 2.3 – « optimiser les informations ».

Cas d'utilisation « prise des rendez-vous »

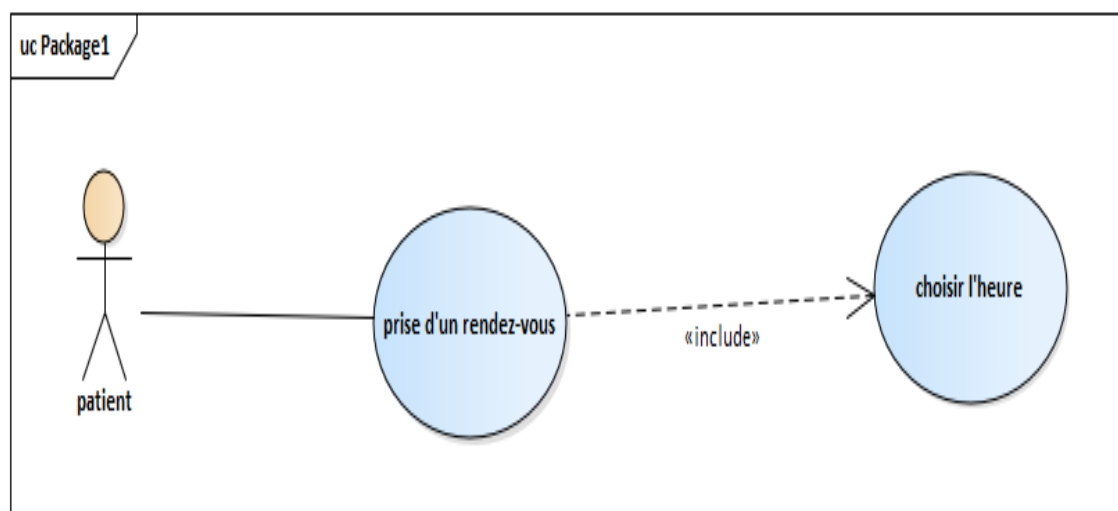


FIGURE 2.11 – cas d'utilisation « prise des rendez-vous ».



## Description du cas d'utilisation « prise des rendez-vous »

SOMMAIRE D'IDENTIFICATION	
Titre	Prise de rendez-vous.
But	Prendre un rendez-vous avec le médecin pour une consultation.
Résumé	Le patient clique sur le bouton « prendre un rendez-vous » pour choisir un temps pour la consultation.
Acteur	Le patient.
Description d'enchaînement	
Précondition	L'utilisateur a déjà un compte.
Scenario nominale	<ol style="list-style-type: none"> <li>1. L'utilisateur doit connecter à son espace « Authentification ».</li> <li>2. Le patient choisit un médecin.</li> <li>3. Le système affiche le profil du médecin.</li> <li>4. Le patient clique sur le bouton « prendre un rendez-vous ».</li> <li>5. Le système affiche la page qui contient les horaires disponibles.</li> <li>6. L'utilisateur choisit l'heure.</li> <li>7. Le système affiche une demande de confirmation.</li> <li>8. L'utilisateur clique sur le bouton « confirmer ».</li> <li>9. Le système envoie un message « rendez-vous bien enregistré ».</li> </ol>
Scenario alternatif	

TABLE 2.4 – « prise des rendez-vous ».

## Cas d'utilisation « choix de médecin »

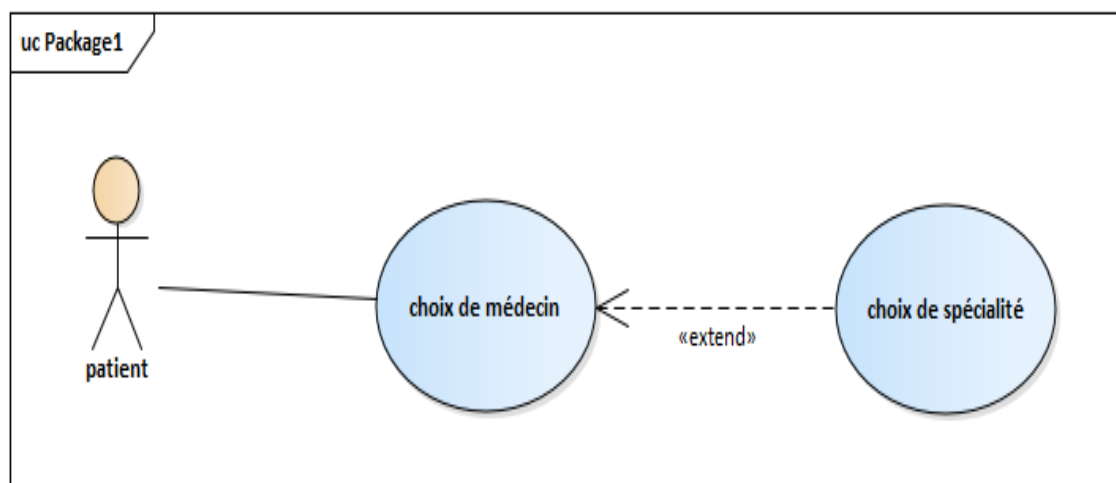


FIGURE 2.12 – cas d'utilisation « choix de médecin ».

## Description du cas d'utilisation « choix de médecin »

SOMMAIRE D'IDENTIFICATION	
Titre	Choix de médecin.
But	Choisir la spécialité et le médecin selon le désir de l'utilisateur.
Résumé	Le patient clique sur « recherche » afin de choisir la spécialité qui lui intéresse, après choisir un médecin.
Acteur	Le patient.
Description d'enchaînement	
Précondition	L'utilisateur doit avoir déjà un compte et il est positionné dans son espace.
Scenario nominale	<ol style="list-style-type: none"> <li>1. Authentification.</li> <li>2. L'utilisateur clique sur la barre de recherche et saisit une spécialité.</li> <li>3. Le système affiche la liste des médecins ayant cette spécialité.</li> <li>4. L'utilisateur choisit un médecin.</li> </ol>
Scenario alternatif	<ol style="list-style-type: none"> <li>1. La spécialité recherchée est introuvable.</li> <li>2. Le système affiche un message d'erreur.</li> </ol>

TABLE 2.5 – « choix de médecin ».

## Cas d'utilisation « communication avec un médecin »

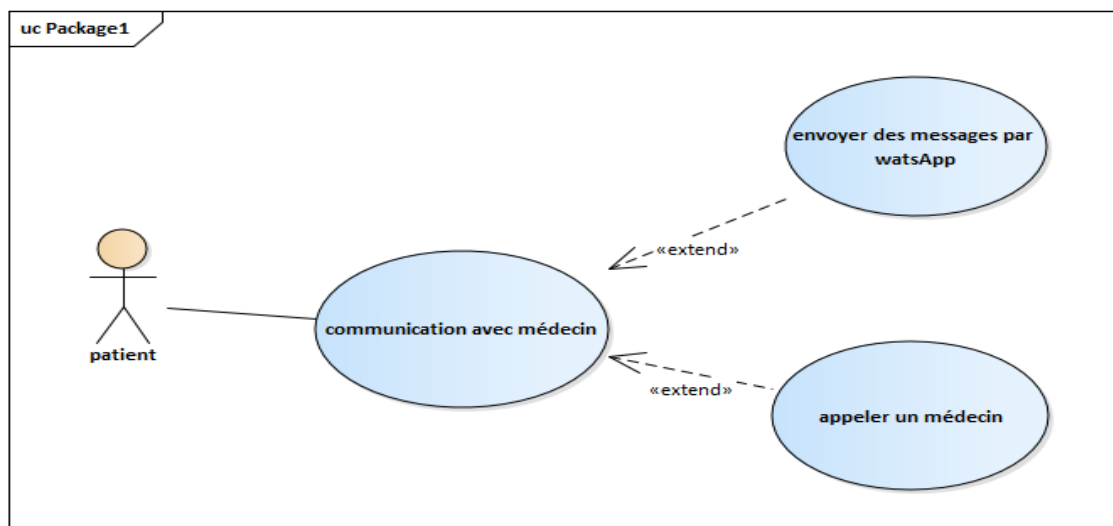


FIGURE 2.13 – cas d'utilisation « communication avec un médecin ».

## Description du cas d'utilisation « communication avec un médecin »

SOMMAIRE D'IDENTIFICATION	
Titre	Communication avec médecin.
But	Assurer Un espace d'échange des messages entre un patient et son médecin (documents médicaux, images, ordonnance...).
Résumé	Le patient clique sur le bouton « Contactes » dont il trouve une liste des médecins qu'il peut les appeler, leur envoyer des messages via WhatsApp.
Acteur	Le patient.
Description d'enchaînement	
Précondition	L'utilisateur doit avoir déjà un compte.
Scenario nominale	<ol style="list-style-type: none"> <li>1. L'utilisateur se connecte à son espace.</li> <li>2. L'utilisateur clique sur le bouton « Contactes ».</li> <li>3. Le système affiche une liste des médecins dont il peut appeler, envoyer un mail ou envoyer des messages via WhatsApp.</li> <li>4. L'utilisateur choisit le type de communication désirable.</li> <li>5. Le système ouvre l'espace de communication convenable.</li> </ol>
Scenario alternatif	

TABLE 2.6 – « communication avec médecin ».

## Cas d'utilisation « évaluation »

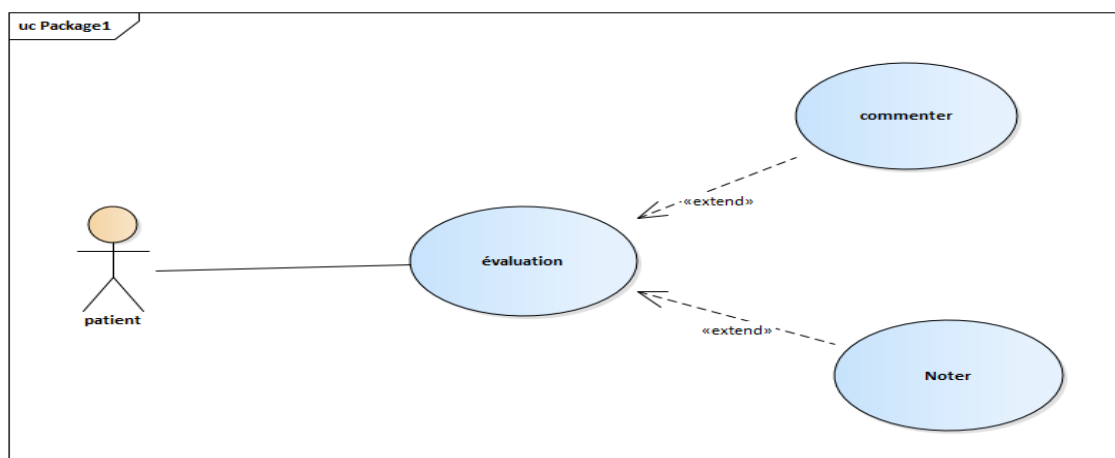


FIGURE 2.14 – cas d'utilisation « évaluation ».

## Description du cas d'utilisation « évaluation »

SOMMAIRE D'IDENTIFICATION	
Titre	Evaluation.
But	Donner un commentaire ou noter un médecin.
Résumé	Le patient clique sur l'icône de « évaluation » dont il peut donner son avis sur le médecin et son service.
Acteur	Le patient.
Description d'enchaînement	
Précondition	L'utilisateur doit avoir déjà un compte.
Scénario nominale	<ol style="list-style-type: none"> <li>1. L'utilisateur se connecte à son espace,</li> <li>2. L'utilisateur accède à l'espace « contacts »,</li> <li>3. Choisir un contact,</li> <li>4. Cliquer sur l'icône de « évaluation »,</li> <li>5. Ecrire un commentaire ou noter le médecin.</li> <li>6. Le système enregistre l'évaluation.</li> </ol>
Scénario alternatif	

TABLE 2.7 – « Evaluation ».

## Cas d'utilisation « contacter mes patients »

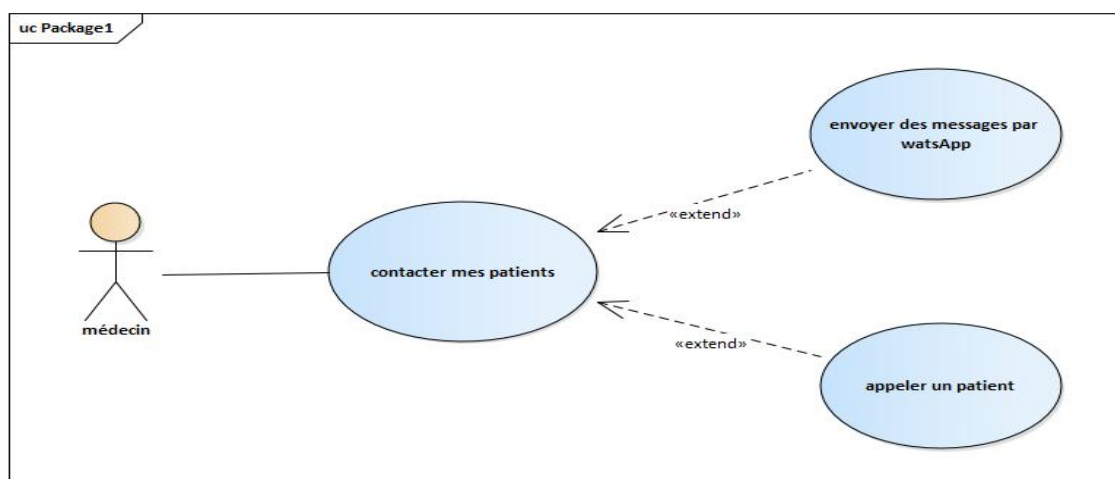


FIGURE 2.15 – cas d'utilisation « contacter mes patients ».

## Description du cas d'utilisation « contacter mes patients »

SOMMAIRE D'IDENTIFICATION	
Titre	Contacter mes patients.
But	Un espace d'échange des messages entre le médecin et son patient (documents médicaux, images, facture, ordonnance ...).
Résumé	Le médecin clique sur l'icône de « contacts » dont il trouve Une liste des patients qui ont envoyé des rendez-vous.
Acteur	Médecin.
Description d'enchaînement	
Précondition	Le médecin doit avoir déjà un compte et doit être connecté à son espace.
Scenario nominale	<ol style="list-style-type: none"> <li>1. Le médecin se connecte à son espace.</li> <li>2. Le médecin clique sur l'icône de « contacts ».</li> <li>3. Le système affiche une liste des patients dont il peut contacter.</li> <li>4. Le médecin choisit un patient.</li> <li>5. Le médecin choisit un type de communication.</li> <li>6. Le système ouvre l'espace de communication convenable.</li> </ol>
Scenario alternatif	

TABLE 2.8 – « contacter mes patients »

## Cas d'utilisation « gestion des rendez-vous »

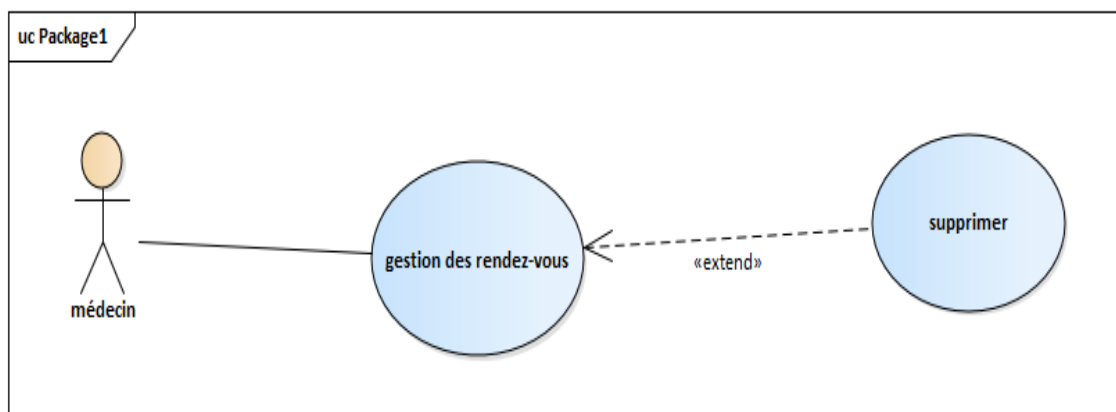


FIGURE 2.16 – cas d'utilisation « gestion des rendez-vous ».

## Description du cas d'utilisation « gestion des rendez-vous »

SOMMAIRE D'IDENTIFICATION	
Titre	Gestion des rendez-vous.
But	Permettre aux médecins de gérer la liste des rendez-vous envoyés Par les patients.
Résumé	Le médecin clique sur l'icône de « rendez-vous » dont il trouve tous les rendez- vous prévus avec la possibilité de supprimer l'un de ces derniers.
Acteur	Médecin.
Description d'enchaînement	
Précondition	Le médecin doit avoir déjà un compte et doit être connecté à son espace.
Scénario nominale	<ol style="list-style-type: none"> <li>1. Le médecin se connecte à son espace.</li> <li>2. Le médecin clique sur l'icône « rendez-vous ».</li> <li>3. Le système le dirige vers une page contenant Les nouvelles demandes des rendez-vous qu'il a la possibilité de les supprimer.</li> </ol>
Scénario alternatif	

TABLE 2.9 – « gestion des rendez-vous ».

## Cas d'utilisation « optimiser le temps de disponibilité »

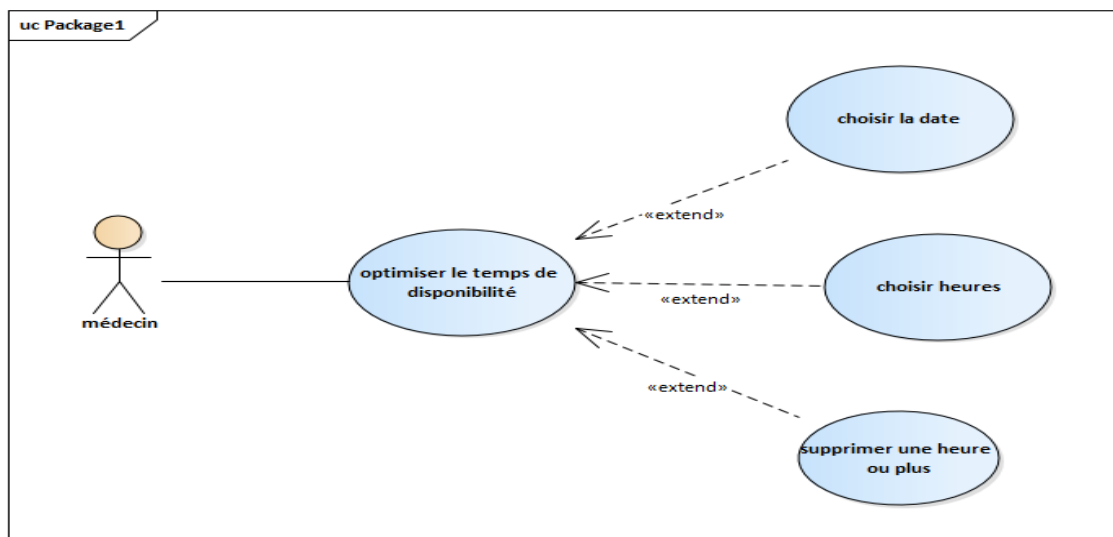


FIGURE 2.17 – cas d'utilisation « optimiser le temps de disponibilité ».

## Description du cas d'utilisation « optimiser le temps de disponibilité »

SOMMAIRE D'IDENTIFICATION	
Titre	Optimiser le temps de disponibilité.
But	Permettre aux médecins de faire des mises à jour de la date et les heures qui y sont disponibles.
Résumé	Le médecin clique sur l'icône de « optimiser » afin de trouver un espace de mise à jour des dates et des d'heures.
Acteur	Médecin.
Description d'enchaînement	
Précondition	Le médecin doit avoir déjà un compte et doit être connecté à son espace.
Scénario nominale	<ol style="list-style-type: none"> <li>1. Le médecin se connecte à son espace,</li> <li>2. Le médecin clique sur l'icône de « optimiser ».</li> <li>3. Le système le dirige vers une page de mise à jour.</li> <li>4. Le médecin choisit une date et les heures de disponibilités.</li> <li>5. Le système affiche les dernières mises à jour.</li> </ol>
Scénario alternatif	

TABLE 2.10 – « optimiser le temps de disponibilité ».

**Administrateur :**

Cas d'utilisation « gestion des comptes utilisateurs »

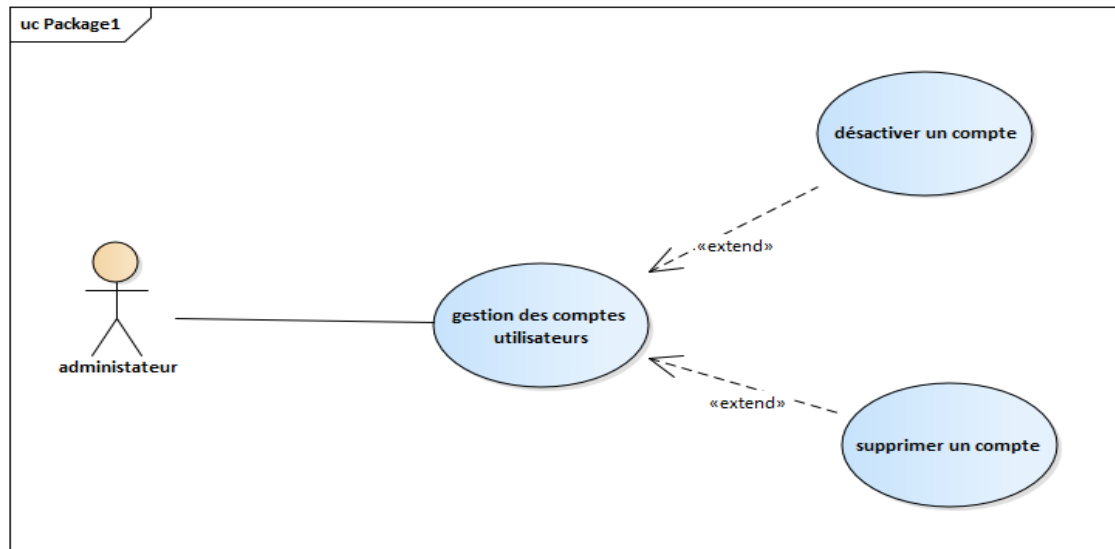


FIGURE 2.18 – cas d'utilisation « gestion des comptes utilisateurs ».



Description du cas d'utilisation « gestion des comptes utilisateurs »

SOMMAIRE D'IDENTIFICATION	
Titre	Gestion des comptes utilisateurs.
But	Permettre à l'administrateur d'application de gérer les comptes des utilisateurs soit patient ou médecin.
Résumé	L'administrateur se connecte au Firebase où il trouve une liste des comptes utilisateurs qu'il peut supprimer ou désactiver.
Acteur	Administrateur.
Description d'enchaînement	
Précondition	L'administrateur doit être connecté à son espace.
Scénario nominale	<ol style="list-style-type: none"> <li>1. L'administrateur se connecte au Firebase.</li> <li>2. Accéder à l'espace d'authentification.</li> <li>3. Cliquer sur Users.</li> <li>4. Le système affiche la liste des comptes.</li> <li>5. Sélectionner un compte et cliquer sur le menu.</li> <li>6. Le système affiche les options disponibles.</li> <li>7. Supprimer/désactiver un compte.</li> <li>8. Demande de confirmation.</li> <li>9. L'administrateur confirmer l'action.</li> <li>10. Le système exécute la demande et affiche un message d'exécution.</li> </ol>
Scénario alternatif	

TABLE 2.11 – « gestion des comptes utilisateurs ».

## 2.5 Diagramme de séquence

Le diagramme de séquence permet de représenter les interactions entre différents objets, selon un point de vue temporel en se basant sur la chronologie des envois de messages. Le temps est représenté comme s'écoulant du haut vers le bas le long des « lignes de vie » [10].

## Diagramme de séquence de la « création du compte médecin »

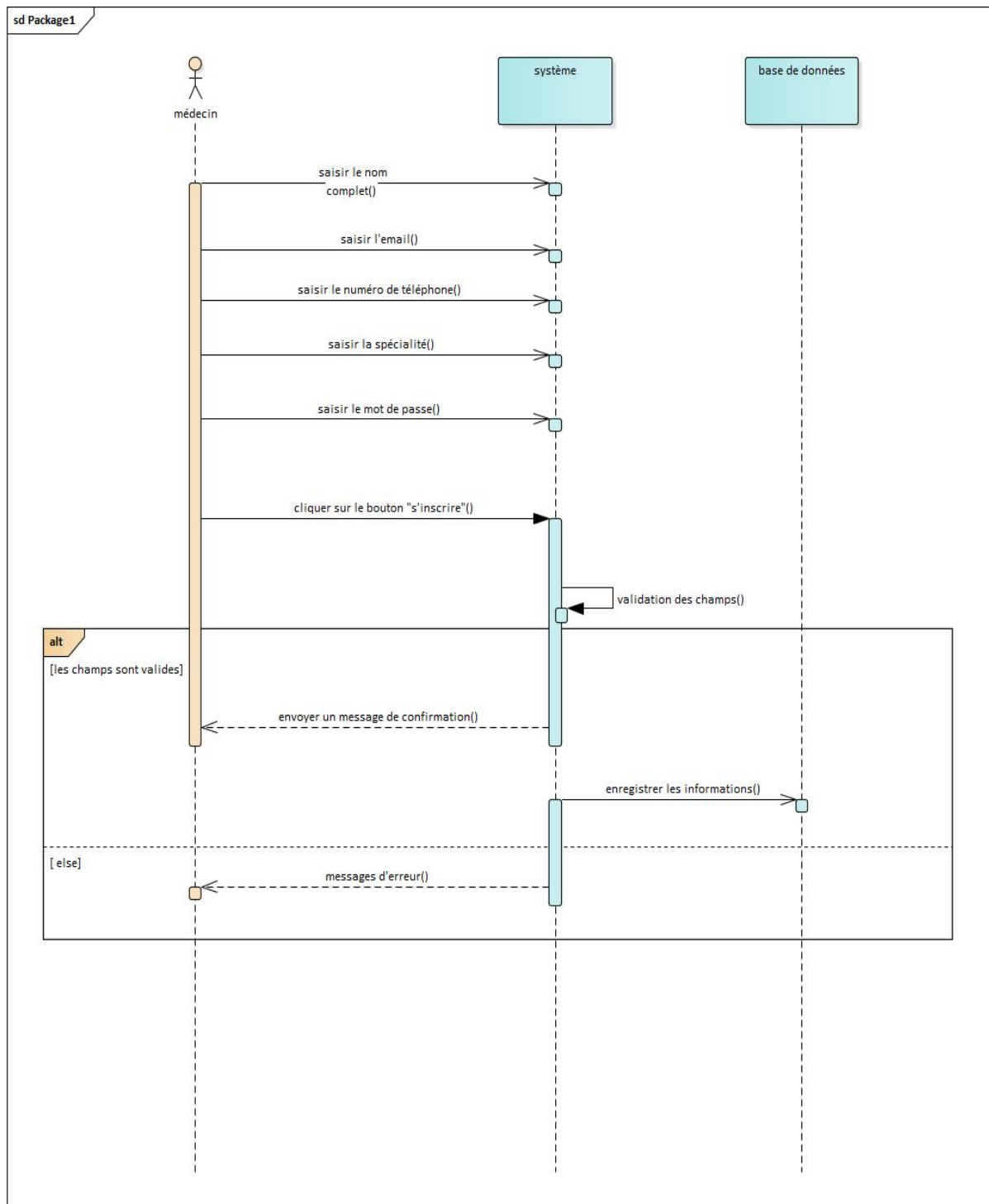


FIGURE 2.19 – séquence de la « création de compte médecin ».

## Diagramme de séquence de la « création du compte patient »

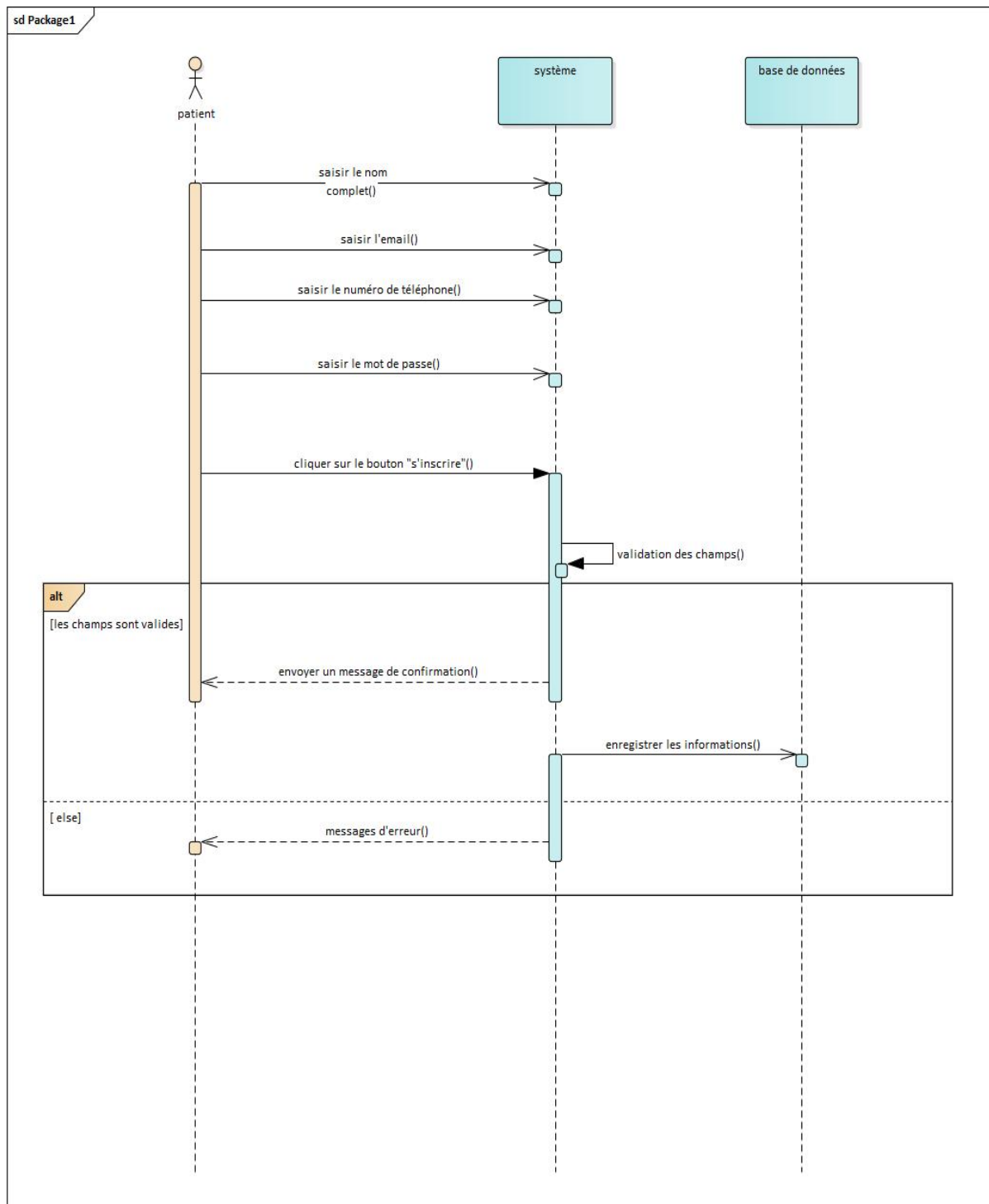


FIGURE 2.20 – séquence de la « création du compte patient ».

## Diagramme de séquence de la « connexion ou authentification »

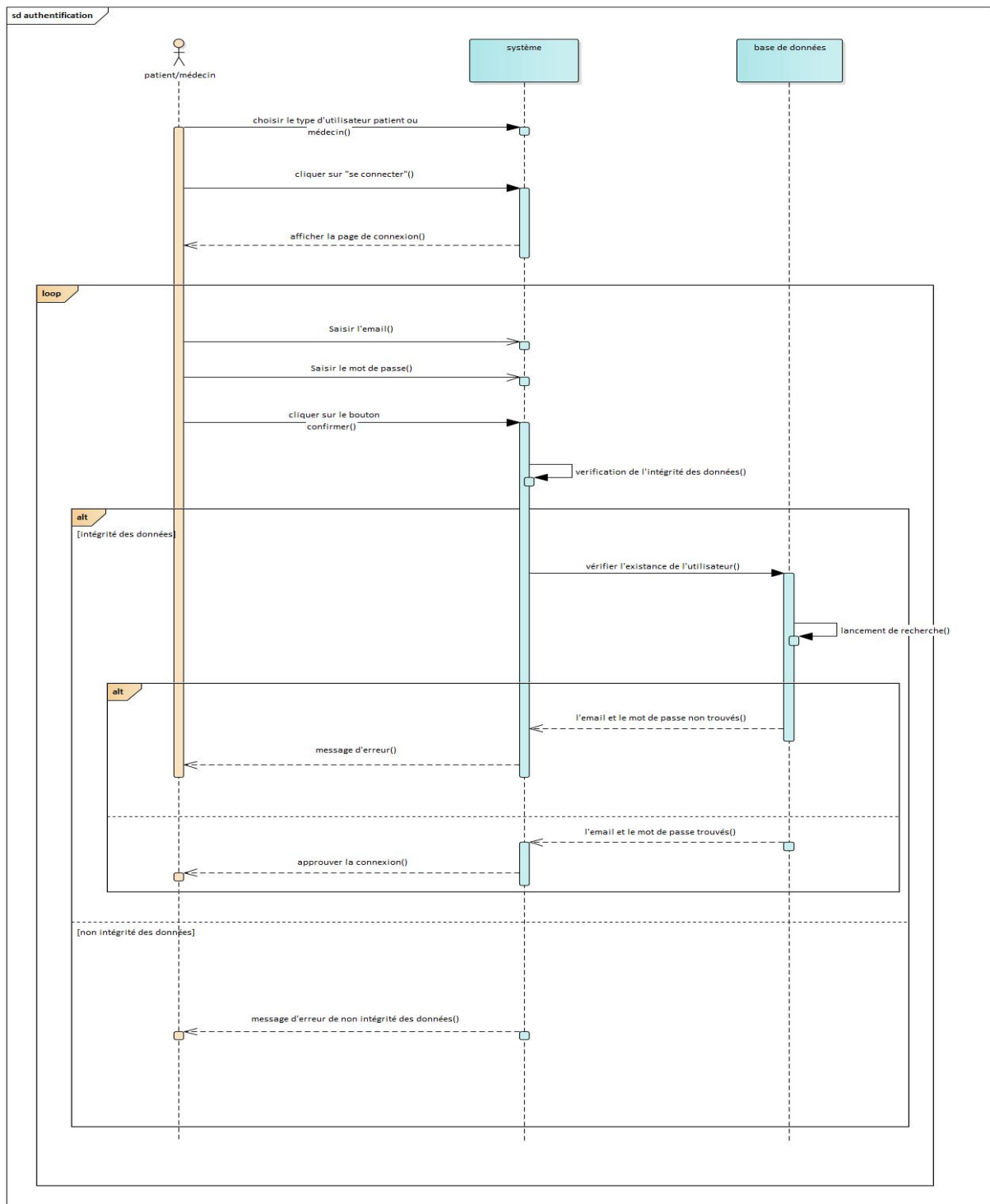


FIGURE 2.21 – séquence de « L'authentification ».

## Diagramme de séquence de « choix du médecin »

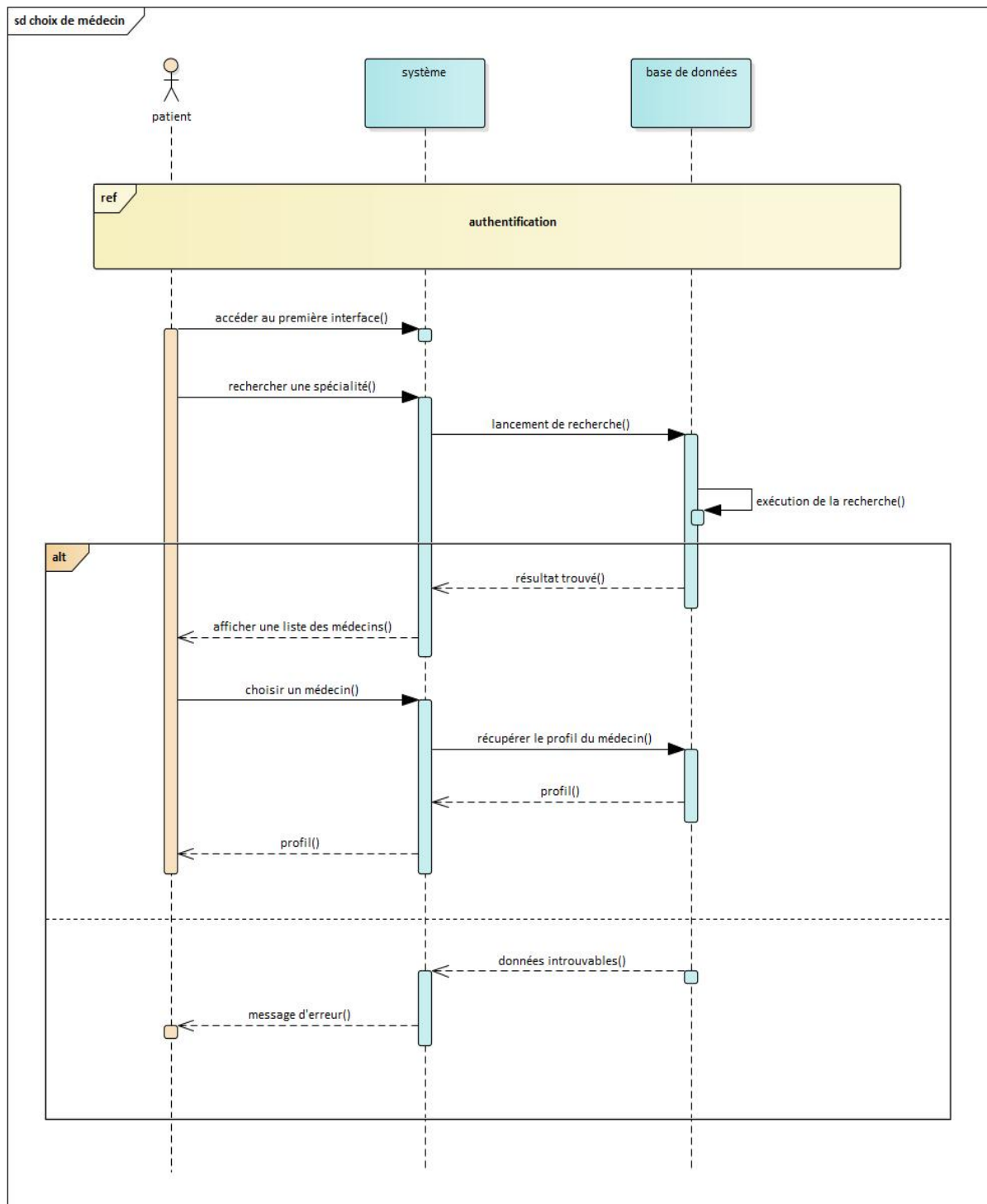


FIGURE 2.22 – séquence de « choix de médecin ».

## Diagramme de séquence de la « prise d'un rendez-vous »

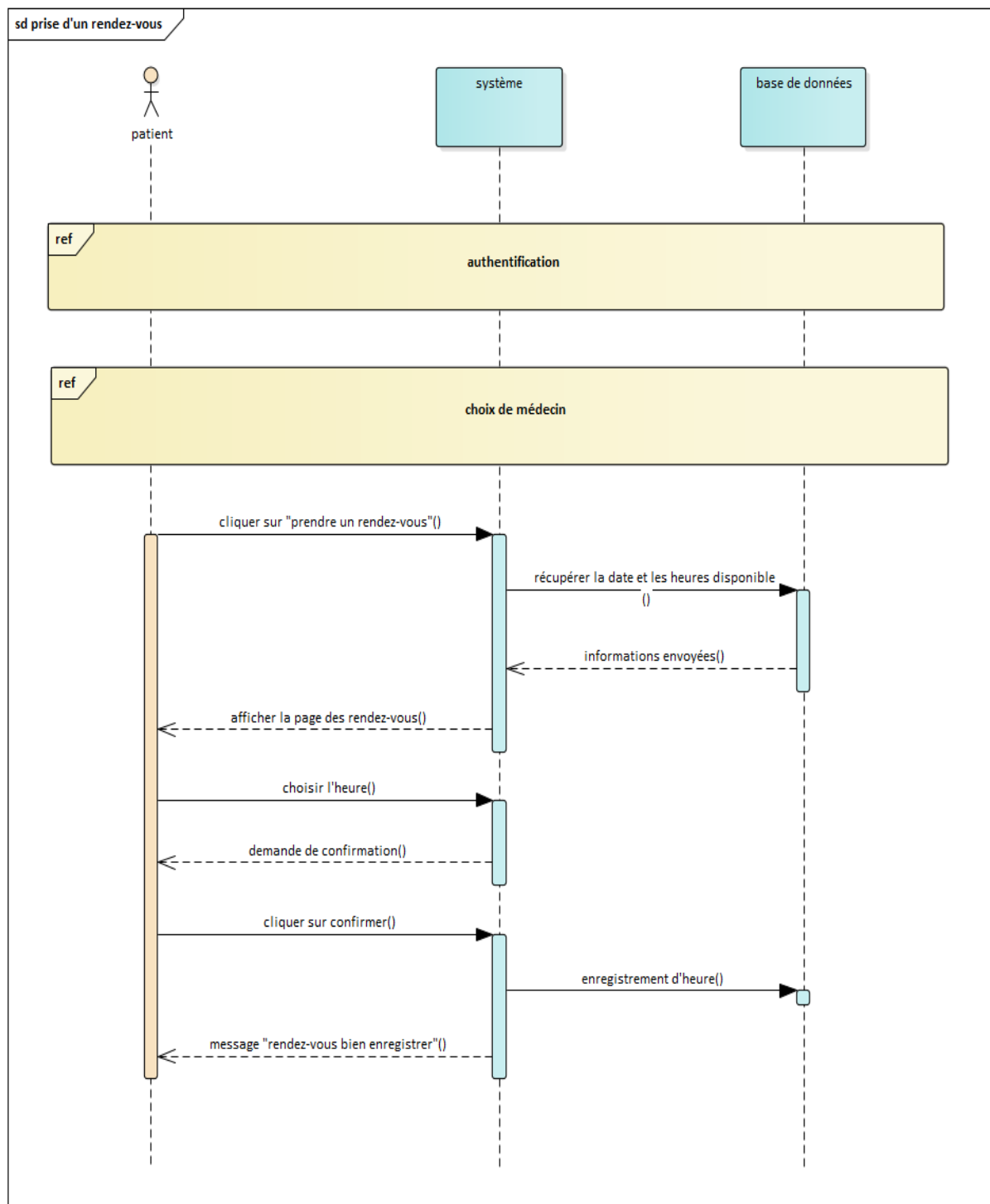


FIGURE 2.23 – séquence de la « prise d'un rendez-vous ».

## Diagramme de séquence de « communication avec médecin »

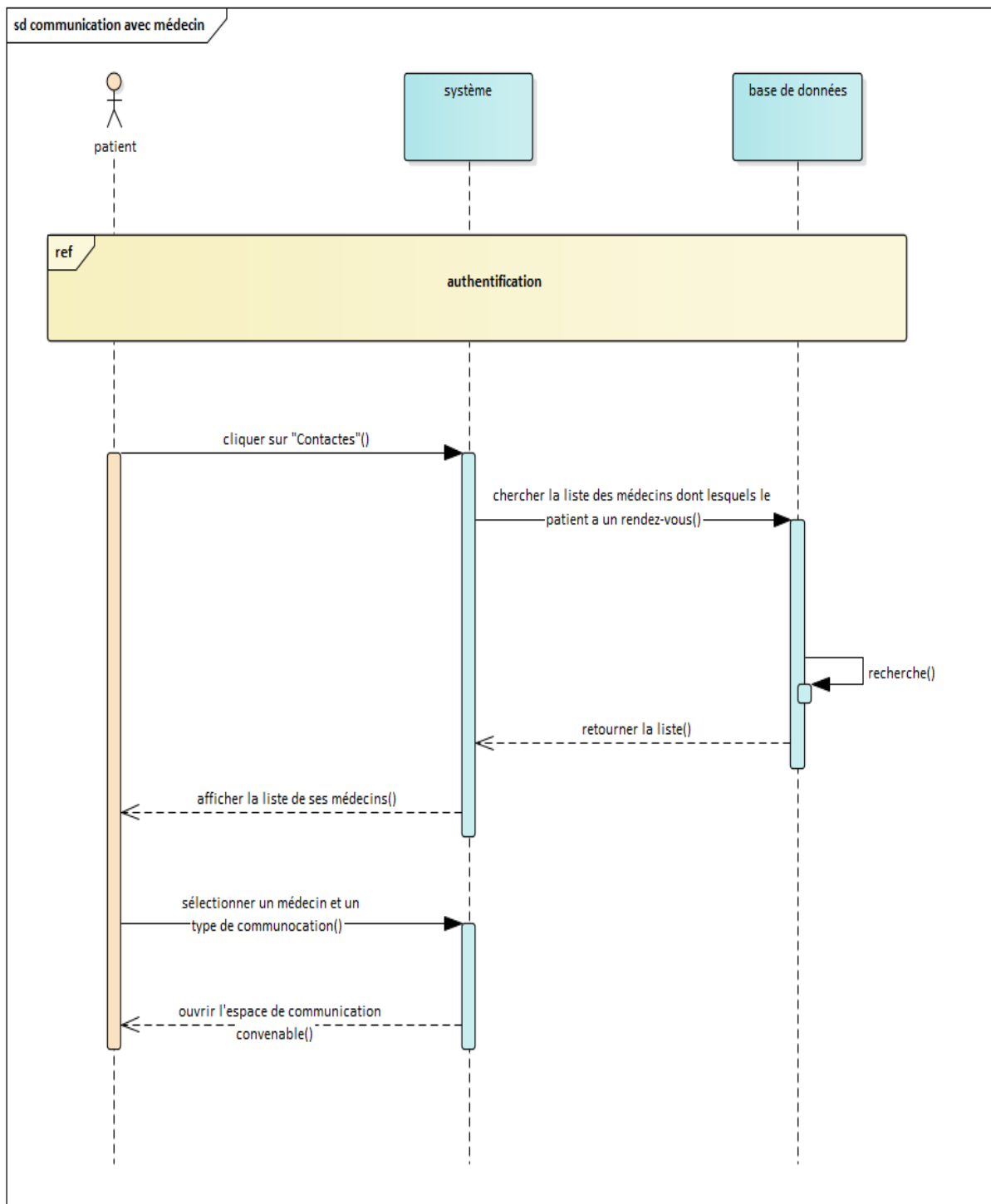


FIGURE 2.24 – séquence de la « communication avec médecin ».

## Diagramme de séquence de « évaluation »

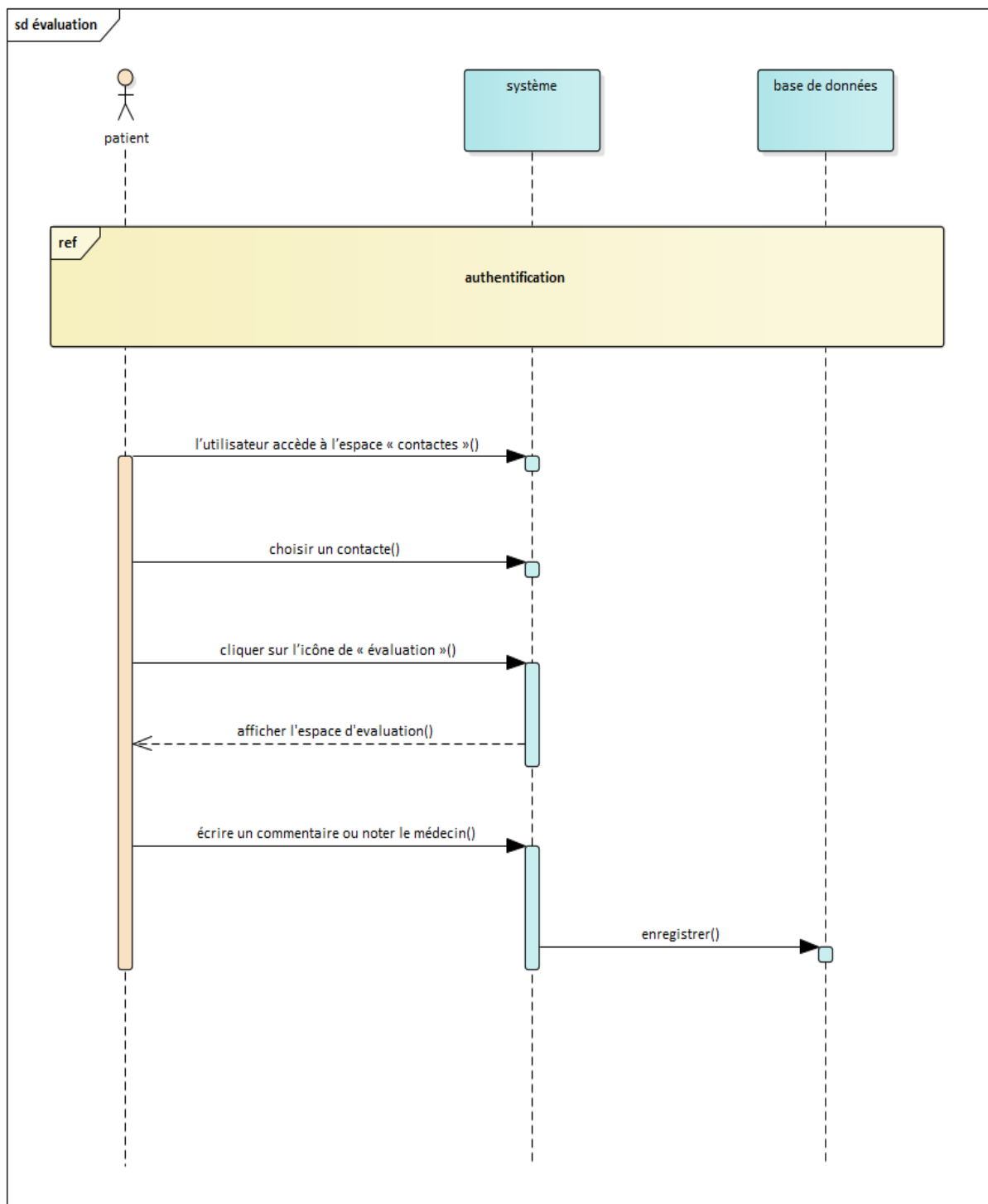


FIGURE 2.25 – séquence de la « Evaluation ».



## Diagramme de séquence de « contacter mes patients »

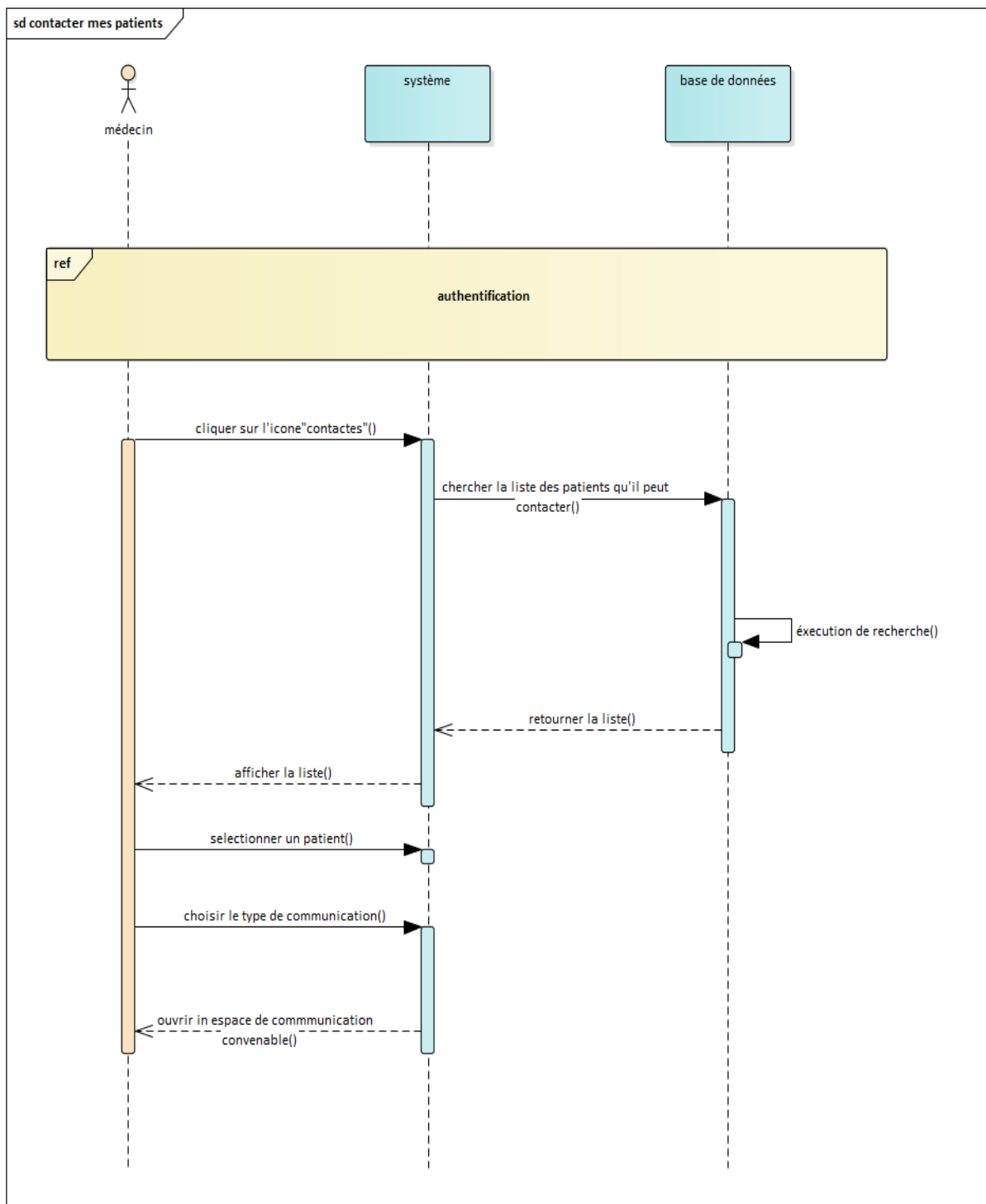


FIGURE 2.26 – séquence de la « contacter mes patients ».

## Diagramme de séquence de « gestion des rendez-vous »

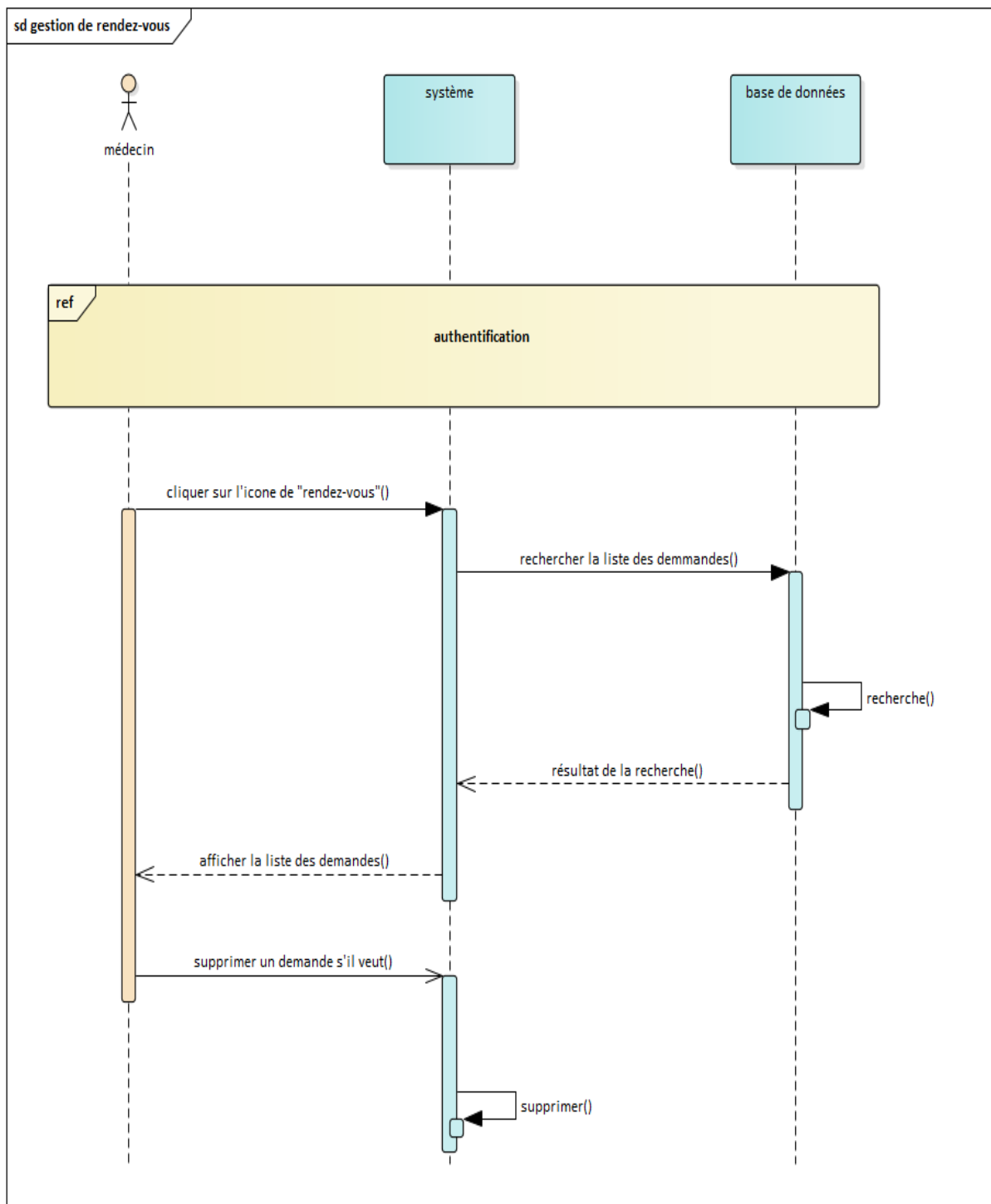


FIGURE 2.27 – séquence de la « gestion des rendez-vous ».

## Diagramme de séquence de « optimiser les informations »

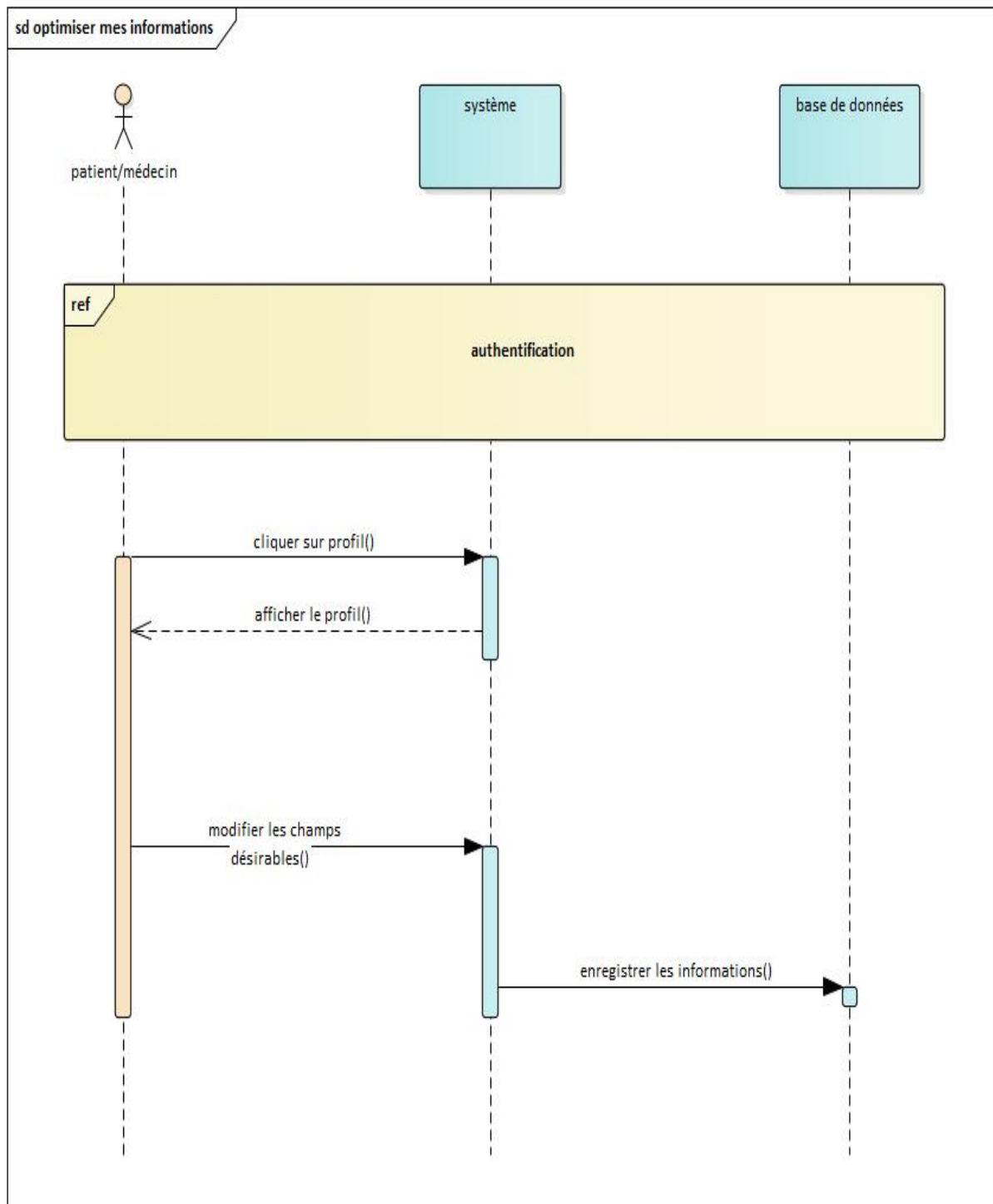


FIGURE 2.28 – séquence de la « optimiser les informations ».

## Diagramme de séquence de « optimiser le temps de disponibilité »

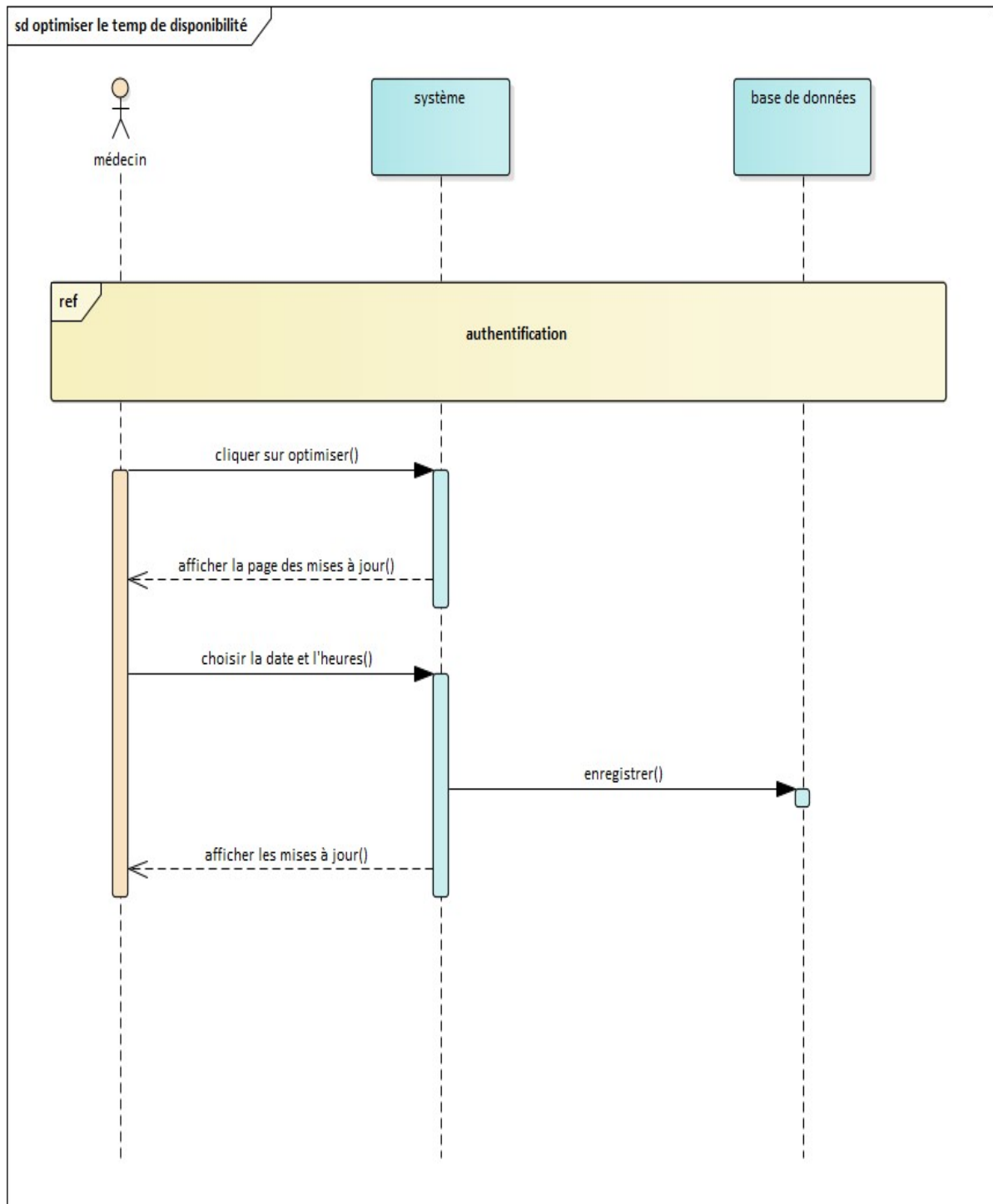


FIGURE 2.29 – séquence de la « optimiser le temps de disponibilité ».

### Diagramme de séquence de « gestion des comptes utilisateurs »

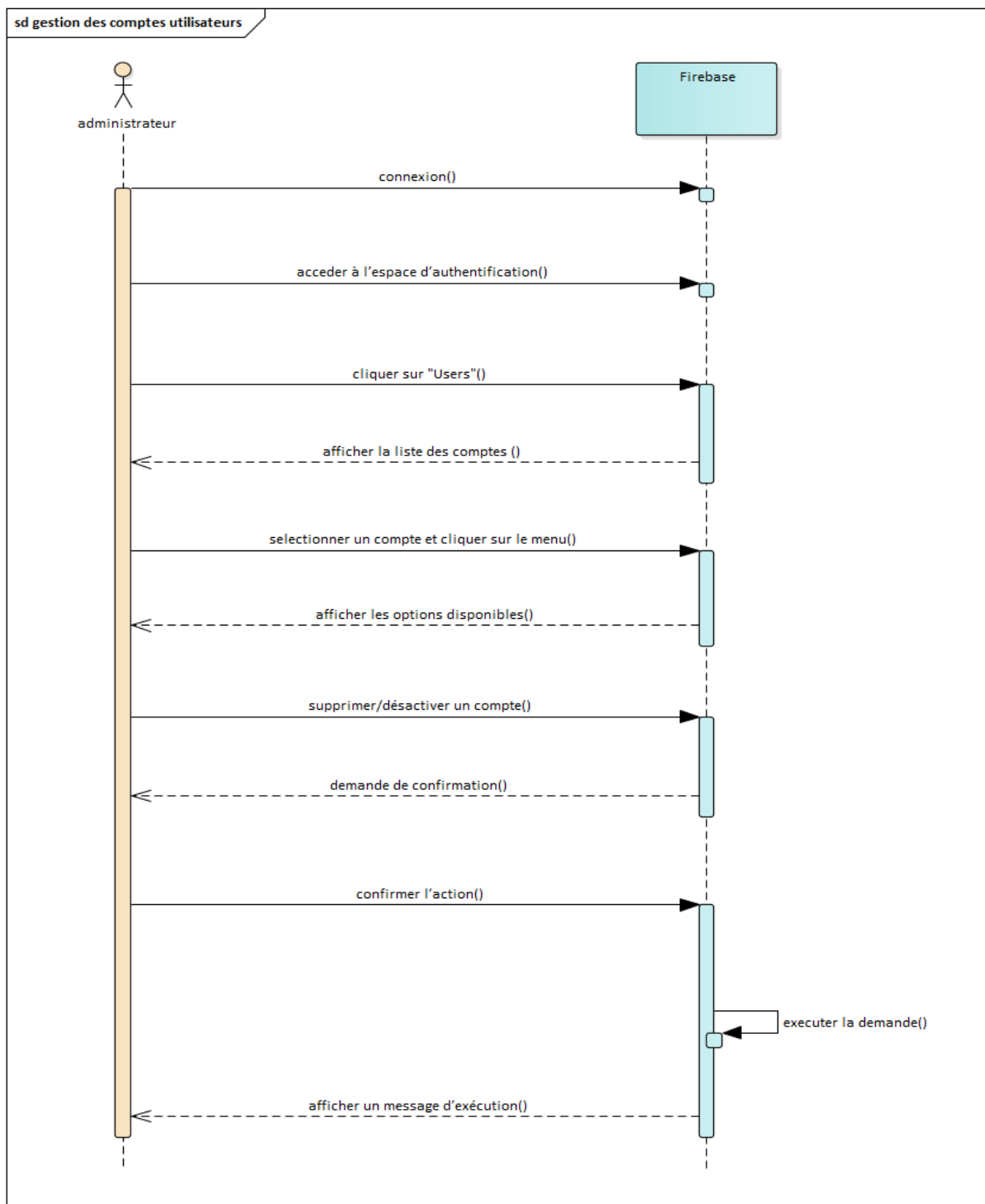


FIGURE 2.30 – séquence de la « gestion des comptes utilisateurs ».

## 2.6 Diagramme de Classe

Le diagramme de classe constitue un élément très important dans la modélisation d'un système.

Ce diagramme représente la structure statique d'un système et il contient les différentes entités composant le système ainsi que les dépendances qui les relient. La figure suivante présente le diagramme de classe global du système comportant les classes manipulées, des attributs et quelques méthodes [11].

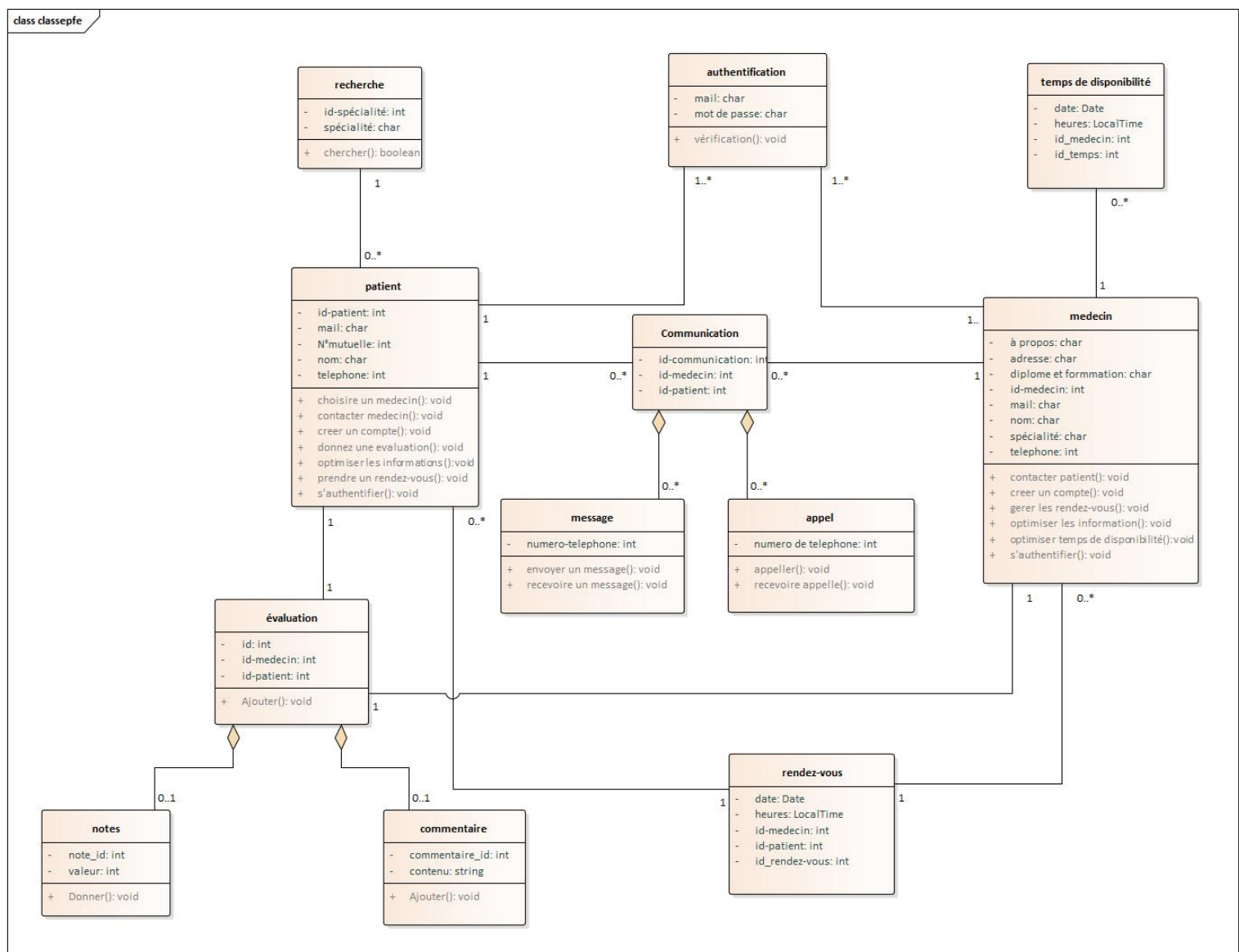


FIGURE 2.31 – diagramme de classe.

## **2.7 Conclusion**

Dans ce chapitre, nous avons proposé une démarche d'analyse et conception pour développer notre application qui est faite grâce à des diagrammes UML. Nous avons d'abord commencé par la spécification des cas d'utilisations dans un premier temps suivi des diagrammes de séquence et le diagramme de classe.

Dans le chapitre suivant nous allons étudier en détails les outils et les langages utilisés durant la phase de constructif.

## CHAPITRE 3 : Les technologies utilisées et réalisation

### Sommaire

<b>3.1</b>	<b>Introduction</b>	<b>56</b>
<b>3.2</b>	<b>Environnement de travail</b>	<b>56</b>
3.2.1	Environnement matériel	56
3.2.2	Environnement logiciel	56
3.2.3	La Base De Données	58
3.2.4	Choix des Techniques	58
<b>3.3</b>	<b>Les interfaces graphiques</b>	<b>67</b>
3.3.1	Application « Ma santé » en action	68
<b>3.4</b>	<b>Conclusion</b>	<b>73</b>



## 3.1 Introduction

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir complété le contenu du chapitre précédant ‘Conception et modélisation’, nous aborderons la partie implémentation dans ce qui suit. Dans ce chapitre nous présenterons la description des environnements matériels et logiciels qui nous ont permis de réaliser notre projet, Les technologies et les langages de programmation que nous avons utilisés. Ensuite nous expliquerons le fonctionnement de notre application mobile ‘Ma Santé’ en présentant ses différentes interfaces qui permettent l’interaction entre l’utilisateur et le système

## 3.2 Environnement de travail

### 3.2.1 Environnement matériel

Marque	Dell	Huawei
Processeur	Intel Core i7-8200U	Intel Core i3-10110U
RAM	8 Go	8 Go
Disque dur	256 Go ssd	256 Go ssd
Système d’exploitation	Windows 10	Windows 10

TABLE 3.1 – Environnement matériel

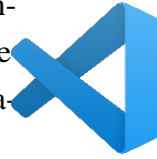
### 3.2.2 Environnement logiciel

- Environnement de développement :

**Android Studio** : Android Studio est un environnement de développement des applications Android, est un des plus puissants éditeurs de code, il offre des solutions plus rapides pour développer des applications performantes et de qualité destinée aux appareils Android. De même il inclut tout ce dont vous avez besoin pour développer une application : éditeur de code et débogueur intelligents, outils d’analyse des performances, émulateurs, et bien plus [12, 13, 14].



**Visual Studio Code** : Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git [15].



- Logiciel de Design :

**Adobe Xd** : C'est un outil de conception d'expérience utilisateur vectorielle pour applications Web et applications mobiles, aider à pré visualiser le résultat du travail directement sur les appareils mobiles. Adobe XD prend en charge le Wireframing du site Web et la création de prototypes clic-through [16].



- Logiciel de modélisation :

**Enterprise Architect** : est un Logiciel de modélisation et un outil complet d'analyse et de conception d'UML, édité par la société australienne Sparx Systèmes. Il comporte le rendement de documentation de haute qualité de flexibilité, il est l'un des logiciels de conception et de modélisation les plus reconnus [17].



- Logiciel de planification :

**Microsoft Office Project** : également appelé Microsoft Project, est un logiciel de planification, de gestion de projets édité par Microsoft. Il permet la planification de projet, gérer les ressources et le budget, ainsi que d'analyser et communiquer les données des projets, et de même c'est le logiciel de gestion de projet le plus utilisé au monde. Les principaux modules de Microsoft Project comprennent le travail de projet et ses équipes, les calendriers et les finances.



Il est conçu pour aider les utilisateurs à définir des objectifs réalistes pour les équipes de projet et les clients en créant des plannings, en répartissant les ressources et en gérant les budgets [18, 19, 20].

### 3.2.3 La Base De Données



FireBase est le nom d'une plateforme mobile de Google qui facilite la création de backend à la fois évolutif et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications pour le mobile et pour le web.

L'objectif de la création de `Firebase.google.com` en 2011 par « James Tamplin » et « Andrew Lee » est d'éviter aux professionnels et aux particuliers de s'engager dans un processus complexe de création et de maintenance d'une architecture serveur.

De plus, la plateforme peut être exploitée par plusieurs utilisateurs en même temps sans connaître un quelconque bug. La praticité est également au rendez-vous grâce à ses fonctionnalités intuitives. Depuis le rachat de la plateforme par Google en 2014, FireBase SDKs a connu de nombreuses améliorations et n'a de cesse de satisfaire ses utilisateurs [21].

### 3.2.4 Choix des Techniques

Avant de choisir la technologie utilisée pour développer notre application, nous allons discuter un petit peu les différentes technologies populaires dans le monde de développement des applications mobiles, en spécifiant les avantages et les inconvénients de chacun.

#### 3.2.4.1 JAVA



Java est un langage de programmation inspiré du langage C++, avec un modèle de programmation orienté objet. Il permet de créer des applications natives, interactives avec l'utilisateur, une fois que vous avez écrit votre page elle ne bouge plus. On ne peut pas gérer les clics ni rien, donc c'est ce que java va nous permettre : créer l'interaction avec l'utilisateur [22, 23].

#### □ **Avantages :**

##### \* Multithreading

Java est un langage multithread. Le multithreading permet de lancer ou d'exécuter plusieurs tâches simultanément dans les applications. Ces tâches sont appelées "threads". Elles ne sont rien d'autre que les plus petites unités de traitement [24].

Le multithreading présente des avantages tels qu'une utilisation efficace des ressources matérielles, un temps de réponse rapide, de meilleures performances et des coûts de maintenances réduites dans les applications Java.

##### \* Gestion de la mémoire

En général, la gestion de la mémoire est une tâche difficile pour les développeurs. Elle est tout à fait essentielle pour le bon fonctionnement et l'efficacité des applications. Mais, lorsqu'il s'agit du développement d'application Java, la gestion de la mémoire est assez facile car Java effectue automatiquement la gestion de la mémoire et la collecte des déchets. Les objets Java sont stockés dans un "tas" [24].

Au démarrage de l'application web, un tas est créé et sa taille peut augmenter ou diminuer pendant l'exécution de l'application. Au fur et à mesure que le tas grandit, les déchets sont collectés et éliminés automatiquement. Ainsi, il y a de l'espace pour que de nouveaux objets puissent y entrer. Cela rend les applications optimisées et très performantes. La menace de plantage des applications diminue également et on peut facilement trouver la fuite de mémoire en cas de mauvaise situation.

##### \* Évolutivité

Le niveau d'extensibilité que fournit Java est une raison majeure de sa popularité dans le monde du développement de logiciels. Les composants de Java sont largement disponibles. Si les développeurs veulent faire évoluer une application verticalement ou horizontalement, c'est plus facile en Java. Ainsi, si l'on doit améliorer les performances et le temps de réponse de l'application Java, il faudra moins de temps et moins d'efforts de codage [24].

### \* Sécurité

Java a été conçu pour être exploité dans des environnements serveur et distribués. Dans ce but, la sécurité n'a pas été négligée. Java permet la construction de systèmes inaltérables et sans virus. Inconvénient [24].

#### □ Inconvénients :

Java est un langage Android verbeux. Coder en Java requiert une certaine expérience et une grande concentration. Il utilise une grande partie de la mémoire virtuelle de la machine. Celle-ci a donc besoin de plusieurs espaces disponibles. Elle doit traiter les valeurs statiques, les instances de classes, équilibrer l'occupation mémoire.

Il ne s'exécute pas directement dans le système d'exploitation du smartphone. La machine virtuelle Java est une couche d'abstraction. Elle se situe entre le hardware du téléphone et les réquisitions de Java. Pour qu'il soit interprété par le device. Java doit d'abord être compilé. Il est moins rapide qu'un langage compilé en natif comme C++. Il peut connaître des difficultés d'accès aux ressources matérielles [25].

### 3.2.4.2 Swift



Swift a été créé par Apple spécialement pour développer des applications pour iOS, mais aussi pour Mac. C'est un langage open source de plus en plus utilisé, en passe de dépasser les langages traditionnels pour le développement d'applications pour téléphones. Swift se base sur le langage Objective-C. Les éléments Swift peuvent s'y intégrer directement [26].

#### □ Avantages :

L'avantage de Swift est qu'il existe une documentation de plus en plus dense, nourrie par les développeurs qui s'y intéressent. Apple met également un grand nombre de ressources à disposition des professionnels et des étudiants [26].

□ **Inconvénients :**

Pour l'instant, l'inconvénient de Swift peut être le fait qu'il se limite à Apple. Cela ne devrait pas durer longtemps, puisqu'il est prévu que le langage s'adapte également à Linux. Du côté des performances, il semble que Swift ait encore une marge de progression [26].

### 3.2.4.3 React Native



React Native est un Framework JavaScript basé sur la bibliothèque React, tous deux créés par Facebook, et il permet de développer des applications iOS et Android avec une seule base de code [27].

□ **Avantages [28] :**

- Rechargement à chaud

React Native dispose de la fonction de rechargement à chaud qui permet à toutes les images d'être reflétées sur l'application presque en même temps, ce qui la fait ressembler à un aperçu en direct. Les changements de code peuvent être vus tout de suite. Les développeurs l'adorent car ils peuvent voir les résultats de leur travail sans attendre, et les entreprises l'adorent car il n'y a pas de perte de temps.

- Super performance, sans aucun compromis

Avec ses contrôles et ses modules natifs, React Native présente un excellent niveau de performances par rapport aux autres frameworks. React Native interagit avec les composants Android et iOS natifs (ciblés) et restitue directement les API natives. Les performances s'améliorent car il utilise un thread séparé de l'interface utilisateur. Ce plus grand avantage se produit lorsque React utilise des API natives.

- Option de rafraîchissement rapide

Avec l'option d'actualisation rapide, les développeurs peuvent exécuter l'application même lors de la mise à jour de l'application et modifier l'interface utilisateur. Comme les changements sont visibles immédiatement, le développeur peut gagner beaucoup de temps et améliorer sa productivité.

□ **Inconvénients** [28] :

- Beaucoup de modules abandonnés :

React Native possède un nombre énorme de bibliothèques. Malheureusement, un trop grand nombre d'entre elles sont de mauvaise qualité ou ont été complètement abandonnées.

- Des applications plus lourdes que des applications natives :

Les applications écrites en React Native doivent être capables d'exécuter du code JavaScript (JavaScript Virtual Machine). Android ne dispose pas de cette fonctionnalité par défaut, ce qui signifie que les applications doivent inclure une bibliothèque qui supporte le code JavaScript, ce qui donne des applications plus volumineuses que leurs homologues natives d'Android.

- Faible sécurité

les applications développées à l'aide de React Native sont très peu sécurisées et peuvent être facilement pénétrées par un autre développeur.

#### 3.2.4.4 Dart



Dart est un langage de programmation orienté objet optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, serveur et web. Il permet de créer des applications à la fois pour Android et iOS [29].

□ **Avantages** [30] :

- Langue facile à apprendre
- Livré avec une bonne documentation :

Les développeurs trouvent que Dart est un bon premier langage de programmation à apprendre car il a une excellente introduction et une très bonne documentation.

- Facteur de performance élevé

Les applications exécutées dans Dart s'exécutent plus rapidement que dans d'autres langages de programmation.

- Excellent support d'outillage

Le langage de programmation dispose d'outils incroyables pour prendre en charge le développement d'applications.

- Peut compiler en instantanés autonomes

Cette fonctionnalité est possible avec d'autres langues, mais elle est simple et rapide avec Dart. Les scripts Dart peuvent compiler eux-mêmes en instantanés autonomes, c'est-à-dire sans nécessité d'autres programmes ou bibliothèques.

### 3.2.4.5 Flutter



Flutter est un framework de développement d'applications mobiles open source de Google met à disposition une grande variété de bibliothèques d'éléments d'UI standard pour Android et iOS [31].



La principale raison de sa popularité est qu'il prend en charge la création d'applications multiplateformes. Flutter est également utilisé pour créer des apps interactives qui s'exécutent sur des pages web ou sur le bureau [32].

Le Flutter SDK se base sur le langage de programmation Dart également développé par Google.

□ **Avantages** [33, 34] :

- Les performances

En tant que framework multiplateforme, Flutter offre des performances inégalées par rapport à ses concurrents (React Native, NativeScript, Ionic). Flutter compile les conceptions vers le code natif. Contrairement à React Native, Flutter rend les widgets directement à partir de la bibliothèque native plutôt que de télécharger les bibliothèques et les composants sur l'appareil avant de les rendre.

- La Sécurité

Google a conçu le framework Flutter en tenant compte de tous les problèmes de sécurité des applications modernes. On peut trouver des plugins fiables et bien testés dans Flutter pour atténuer les risques de sécurité tels que les failles d'authentification des utilisateurs, les injections de code malveillant et les fuites de données.

- Gamme de widgets

Flutter dispose d'une large gamme de widgets. Ces widgets ont des capacités étendues qui permettent aux développeurs de construire facilement des interfaces complexes.

- Très rentable

Comparé aux autres frameworks de développement d'applications multiplateformes, Flutter est plus rentable. Il répond à toutes les exigences de chaque entreprise, quelque soit son modèle et sa taille.

- Productivité améliorée

Flutter dispose d'une fonction de rechargement à chaud qui permet aux développeurs et aux concepteurs de se coordonner efficacement et de vérifier les changements immédiatement sans modifier le code.

Pendant le processus de développement de l'application, toutes les modifications de l'interface utilisateur sont visibles pour les développeurs et les concepteurs. Cela leur permet d'économiser du temps et des efforts.

- Un développement plus rapide

Offrir les résultats les plus rapides. Les développeurs peuvent déboguer et tester les codes rapidement.

- Une maintenance accélérée et optimisée

Les corrections de bugs sont rapides et régulières.

- La conception design considérablement simplifiée

Grâce à Flutter il est beaucoup plus facile d'intégrer des animations dans les applications mobiles.

#### **3.2.4.6 Notre choix**

Pour la mise en œuvre de notre application multiplateforme, nous avons choisi le langage de programmation Dart et le Framework Flutter grâce à ses meilleurs avantages afin de la développer d'une manière rapide et simplifiée et qui fonctionnera à la fois sur les plateformes iOS et Android.

Et pour la base de Données (back-end) on a opté de travailler avec « FireBase » en tant que Plate-forme de développement d'application de bout en bout.

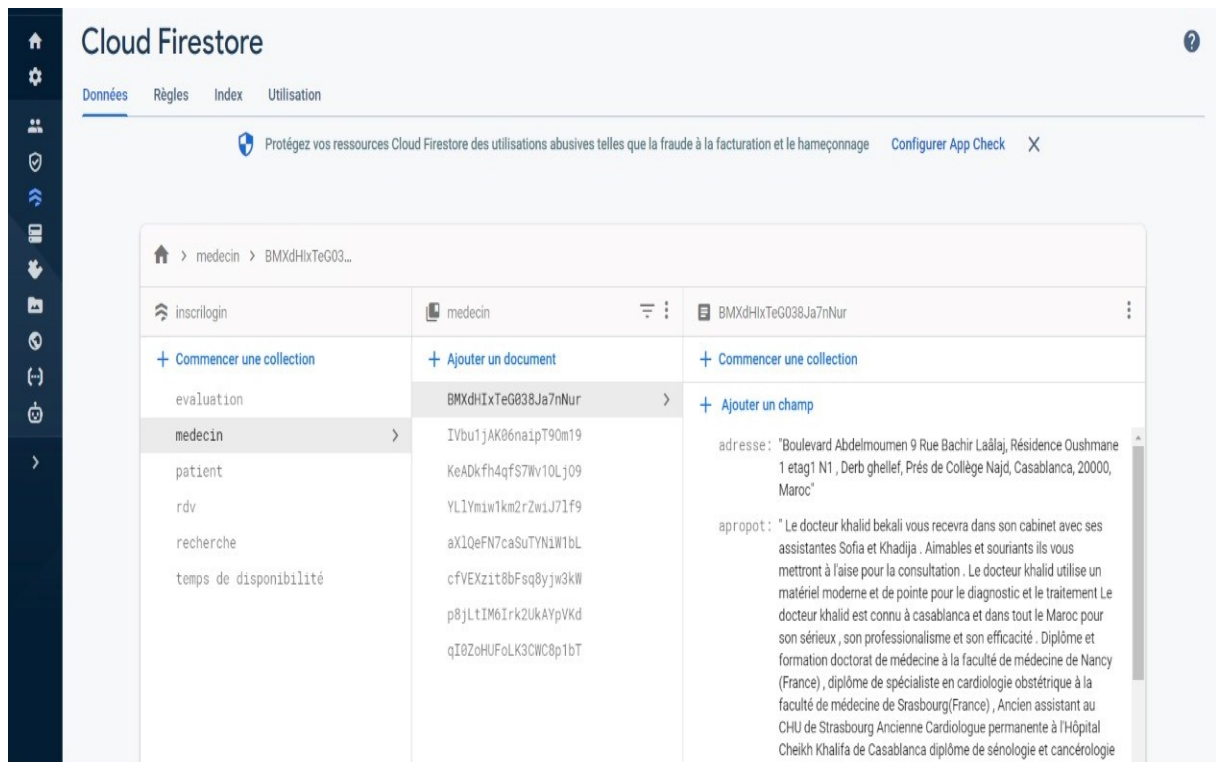


FIGURE 3.1 – « la base de données ».

Car elle est caractérisée par [35] :

1. Démarrage gratuit
2. Vitesse de développement
3. Plate-forme de développement d'applications de bout en bout
4. Proposé par Google
5. Les développeurs peuvent se concentrer sur le développement frontend
6. Il n'y a pas de serveur
7. Il offre des capacités d'apprentissage automatique
8. Génère du trafic vers vos applications
9. Contrôle des erreurs
10. Sécurité

## 3.3 Les interfaces graphiques

Les interfaces graphiques de l'application sont très importantes, car elles permettent de faciliter le dialogue entre l'homme et la machine ainsi que d'améliorer les performances de l'application. Dans cette partie nous présentons les principales fonctionnalités de notre application par la description des interfaces.

### 3.3.1 Application « Ma santé » en action

#### 3.3.1.1 Les interfaces de réception

Ce sont les premières interfaces trouvées par l'utilisateur dans sa première visite.

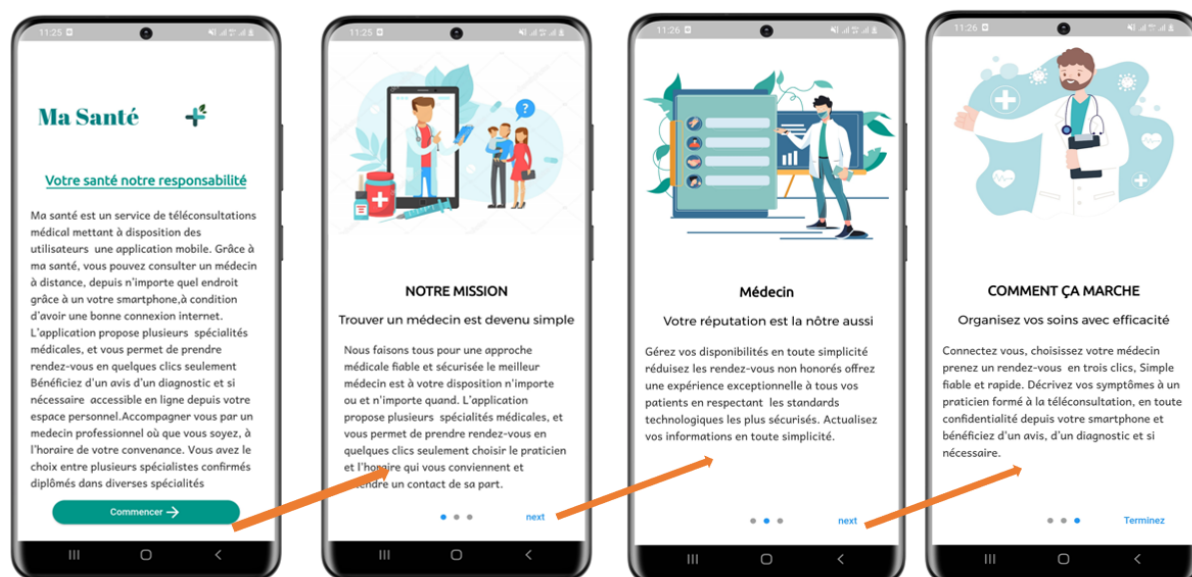


FIGURE 3.2 – Interfaces de réception.

### 3.3.1.2 Les interfaces de la connexion et la création des comptes utilisateurs

En ce qui concerne la première utilisation de l'application, l'utilisateur doit choisir s'il est un patient (consulter en ligne) ou un médecin (vous êtes un médecin), puis il doit créer un compte s'il n'en a pas ou faire une connexion.



FIGURE 3.3 – Interfaces de la connexion et la création des comptes utilisateurs.

### 3.3.1.3 Les interfaces des fonctionnalités de l'espace patient

Une fois le patient est authentifié, il accède à l'interface d'accueil, par laquelle il peut accéder aux différentes fonctionnalités de l'application.

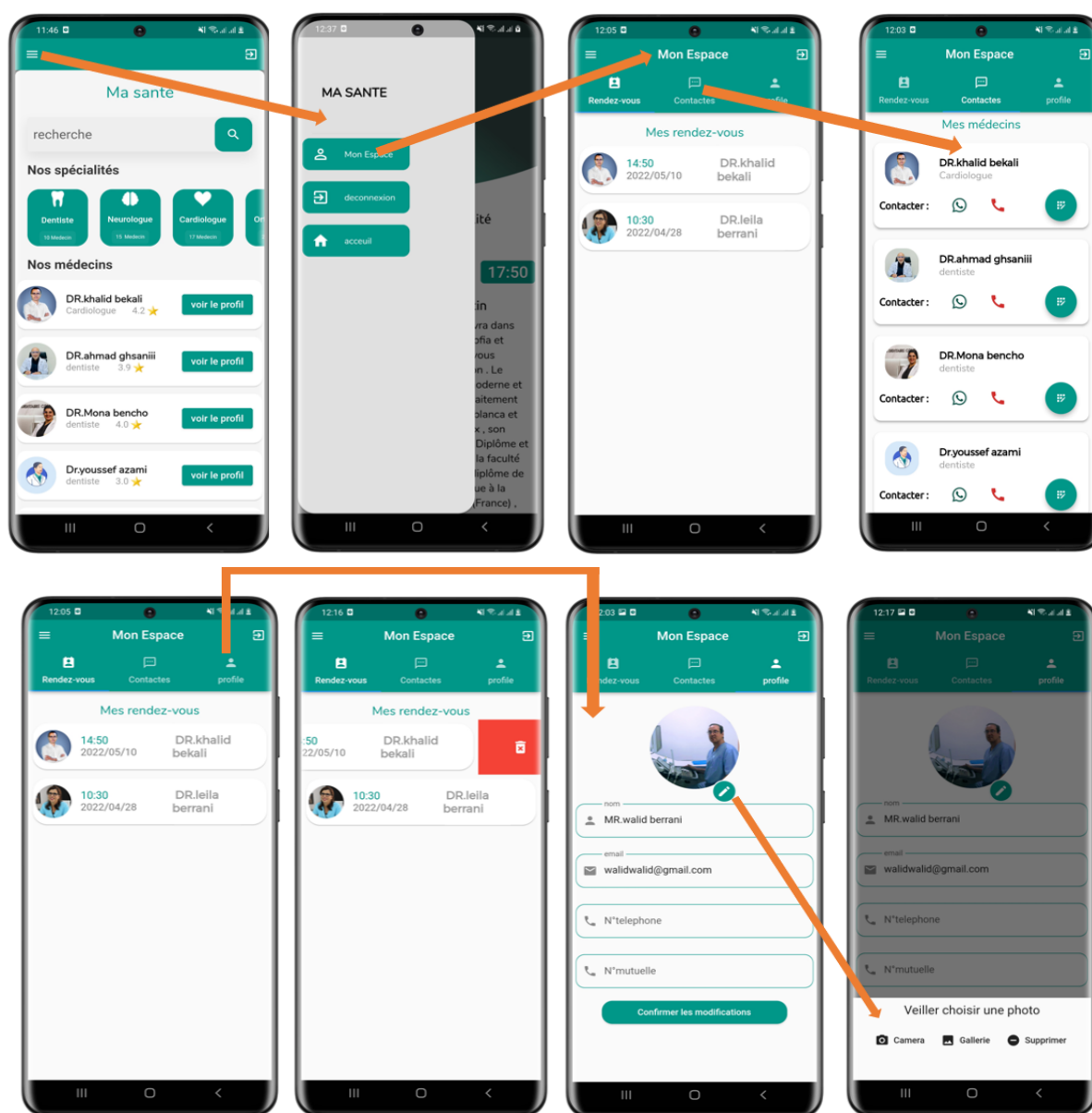


FIGURE 3.4 – Interfaces des fonctionnalités de l'espace patient.

### 3.3.1.4 Les interface de la prise d'un rendez-vous

Pour que le patient prenne un rendez-vous, il accède au profil du médecin, vérifie ses horaires, et effectue ensuite sa prise de rendez-vous.

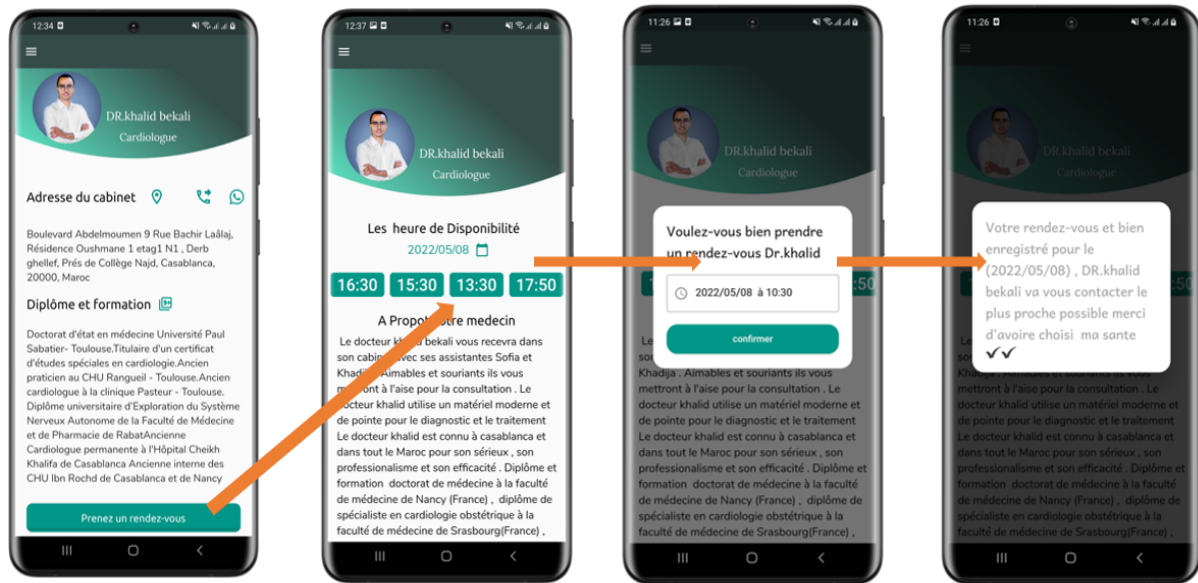


FIGURE 3.5 – Interface de la prise d'un rendez-vous.



### 3.3.1.5 les interfaces des fonctionnalités de l'espace médecin

Ces interfaces permettent au médecin d'accéder aux différents services de l'application.

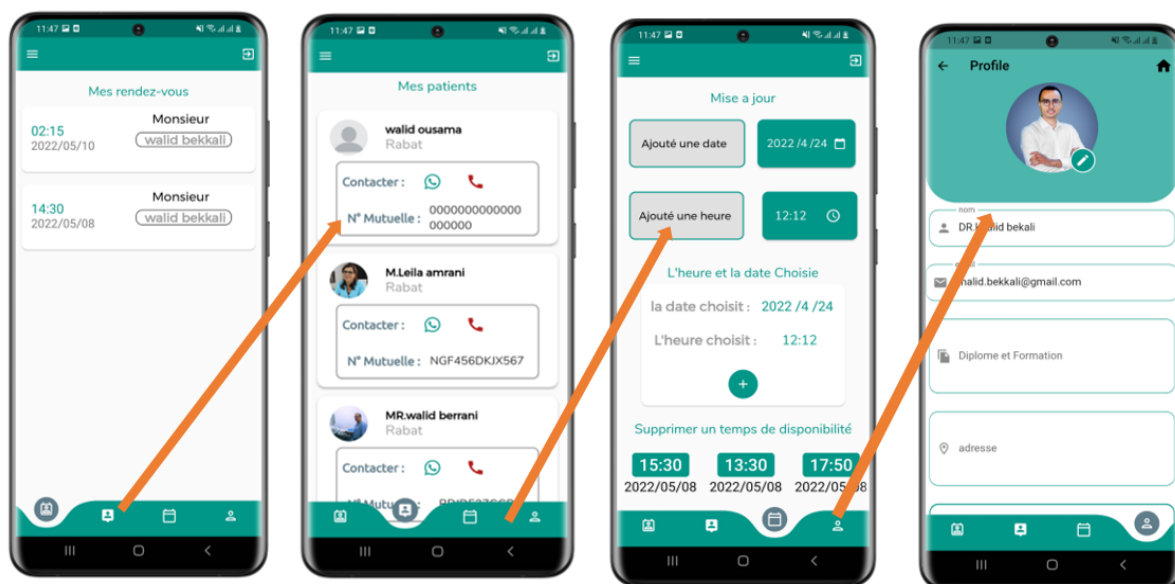


FIGURE 3.6 – Interfaces des fonctionnalités de l'espace médecin.

### 3.3.1.6 les interfaces de la mise à jour de temps de disponibilité

Ces interfaces permettent au médecin de mettre à jour son temps de disponibilité en choisissant la date et l'heure.

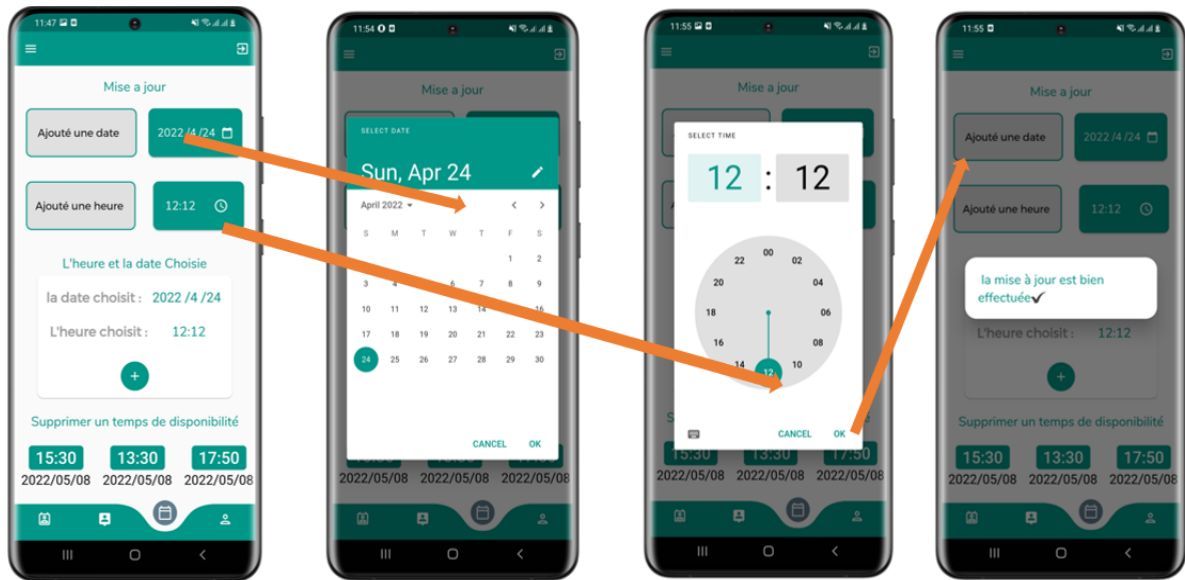


FIGURE 3.7 – Interfaces de la mise à jour de temps de disponibilité.

## 3.4 Conclusion

La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application. Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement, les outils et les langages de développement associés à notre système. En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.

---

## Conclusion Générale

De nos jours personne, ne peut nier que la téléphonie mobile est le secteur le plus dynamique, le plus rentable et le plus innovant de toute l'industrie des Télécommunications. Notamment le suivi de l'état de santé des patients, le suivi entre patients et professionnels de santé.

Notre projet s'agit de développer une application mobile sous Android et iOS «Ma Santé», pour patients et médecins qui vise à une meilleure gestion de santé, un bon démarche médical à tout moment en tous lieux d'une manière facile.

Ce projet nous a été très bénéfique, car nous avons enrichi nos connaissances sur les deux plans : théorique et pratique. Il nous a aussi permis de découvrir et d'acquérir de nouvelles connaissances en matière de développement mobile.

Finalement, notre travail ne s'arrête pas à ce niveau, en effet plusieurs fonctionnalités peuvent être ajoutées à notre application.

---

## Annexe

### Documentation des outils :

- \* documentation officiel de flutter <https://flutter.dev/docs> Consulté le 8 mai 2022
- \* les bibliothèques utiles <https://pub.dev/> Consulté le 15 mars 2022
- \* la communauté de flutter sur GitHub <https://github.com/flutter> Consulté le 8 mai 2022
- \* FireBase <https://firebase.google.com/> Consulté le 10 mai 2022
- \* cours de flutter <https://www.javatpoint.com/flutter> Consulté le 8 mai 2022
- \* Documentation pour les développeurs d'application <https://developer.android.com/docs>.  
Consulté le 13 mars

### Liens de téléchargement :

- \* Android Studio <https://developer.android.com/studio/> Consulté le 14 mars 2022
- \* éditeur de texte en ligne de langage Dart <https://dartpad.dev/>. Consulté le 3 mars 2022
- \* Enterprise Architect <https://sparxsystems.com/products/ea/15.1/>. Consulté le 2 octobre 2021
- \* Adobe Xd <https://www.adobe.com/products/xd.html> Consulté le 7 mars 2022
- \* Ms Project [tps://www.commentcamarche.net/download/telecharger-34056851-microsoftoffice-project-standard](https://www.commentcamarche.net/download/telecharger-34056851-microsoftoffice-project-standard) Consulté le 30 avril 2022
- \* Visual studio code <https://code.visualstudio.com/> Consulté le 16 mars 2022

---

## Webographie

- [1] Voiladoc, <https://voiladoc.ma/a-propos-de-voiladoc/> Consulté le 1 mai 2022
- [2] Dabadoc, <https://www.dabadoc.com/a-propos>. Consulté le 1 mai 2022
- [3] Cycle de vie, <https://www.manager-go.com/gestion-de-projet/cycle-en-v.htm> Consulté le 7 mars 2022
- [4] Cycle de vie, <https://www.appvizer.fr/magazine/operations/gestion-de-projet/cycle-v> le 7 mars 2022
- [5] Intérêt de la modélisation, <https://www.bfpro.fr/linteret-de-la-modelisation-3d/> Consulté le 15 mars 2022
- [6] UML, [https://fr.wikipedia.org/wiki/UML\(informatique\)](https://fr.wikipedia.org/wiki/UML(informatique)) Consulté le 15 mars 2022
- [7] UML, <https://www.futura-sciences.com/tech/definitions/informatique-uml-3979/> le 15 mars 2022
- [8] Entreprise Architect, <https://sparxsystems.fr/products/ea/?ref=eaRes> Consulté le 15 mars 2022
- [9] Diagramme de Cas d'utilisation, <https://laurent-audibert.developpez.com/Cours/UML/?page=diagramme-cas-utilisation><sup>s</sup> Consulté le 16 mars 2022
- [10] Diagramme de Séquence, <https://www.ibm.com/docs/fr/rsm/7.5.0?topic=uml-sequence-diagrams> Consulté le 21 mars 2022
- [11] Diagramme de Classe, <https://www.ibm.com/docs/fr/rsar/9.5?topic=diagrams-class> le 26 mars 2022
- [12] Android Studio, <https://www.encyclopedie.fr/definition/AndroidStudio> Consulté le 1 avril 2022
- [13] Android Studio, <https://fr.theastrologypage.com/android-studio> Consulté le 1 avril 2022
- [14] Android Studio, <https://fr.wikipedia.org/wiki/AndroidStudio> Consulté le 1 avril 2022
- [15] Visual Studio Code, <https://framalibre.org/content/visual-studio> Consulté le 1 avril 2022
- [16] Adobe Xd, <https://www.blogdumoderateur.com/tools/adobe-xd/> Consulté le 1 avril 2022
- [17] Entreprise Architect, <https://www.encyclopedie.fr/definition/EnterpriseArchitect> le 2 avril 2022
- [18] Microsoft Office Project, <https://www.techno-science.net/definition/10899.html> le 2 avril 2022
- [19] Microsoft Office Project, <https://management.savoir.fr/ms-project/> Consulté le 2 avril 2022
- [20] Microsoft Office Project, <https://www.cours-gratuit.com/cours-project/cours-d-initiation-a-ms-project-pour-debutant> Consulté le 2 avril 2022

- 
- [21] [Firebase](https://junto.fr/blog/firebase/), [https ://junto.fr/blog/firebase/](https://junto.fr/blog/firebase/) Consulté le 3 avril 2022
- [22] [Java](https://fr.goodbarber.com/creer-app/7-creer-une-application-mobile-avec-java/), [https ://fr.goodbarber.com/creer-app/7-creer-une-application-mobile-avec-java/](https://fr.goodbarber.com/creer-app/7-creer-une-application-mobile-avec-java/) le 3 avril 2022
- [23] [Java](https://www.lemagit.fr/definition/Java), [https ://www.lemagit.fr/definition/Java](https://www.lemagit.fr/definition/Java) Consulté le 3 avril 2022
- [24] [Avantages de Java](https://ibracilinks.com/blog/quand-et-pourquoi-java-est-utilise-pour-le-developpement-dapplications), [https ://ibracilinks.com/blog/quand-et-pourquoi-java-est-utilise-pour-le-developpement- dapplications](https://ibracilinks.com/blog/quand-et-pourquoi-java-est-utilise-pour-le-developpement-dapplications) Consulté le 3 avril 2022
- [25] [Inconvénients de Java](https://fr.acervolima.com/avantages-et-inconvenients-de-java/), [https ://fr.acervolima.com/avantages-et-inconvenients-de-java/](https://fr.acervolima.com/avantages-et-inconvenients-de-java/) le 3 avril 2022
- [26] [Avantages et Inconvénients de swift](https://blog.hubspot.fr/website/langage-de-programmation), [https ://blog.hubspot.fr/website/langage-de-programmation](https://blog.hubspot.fr/website/langage-de-programmation) 4 avril
- [27] [React Native](https://mobiskill.fr/blog/conseils-emploi-tech/developpement-mobile-2021-react-native-vs-flutter-2/), [https ://mobiskill.fr/blog/conseils-emploi-tech/developpement-mobile-2021-react-native-vs-flutter-2/](https://mobiskill.fr/blog/conseils-emploi-tech/developpement-mobile-2021-react-native-vs-flutter-2/) Consulté le 4 avril 2022
- [28] [Avantages et Inconvénients de React Native](https://www.software-developer-india.com/fr/avantages-et-inconvenients-de-react-native/), [https ://www.software-developer-india.com/fr/avantages-et-inconvenients-de-react-native/](https://www.software-developer-india.com/fr/avantages-et-inconvenients-de-react-native/) Consulté le 4 avril 2022
- [29] [Dart](https://fr.wikipedia.org/wiki/Dart(langage)), [https ://fr.wikipedia.org/wiki/Dart\(langage\)](https://fr.wikipedia.org/wiki/Dart(langage)) Consulté le 4 avril 2022
- [30] [Avantages de Dart](https://www.software-developer-india.com/fr/langage-de-programmation-dart-quels-sont-les-avantages/), [https ://www.software-developer-india.com/fr/langage-de-programmation-dart-quels-sont-les-avantages/](https://www.software-developer-india.com/fr/langage-de-programmation-dart-quels-sont-les-avantages/) Consulté le 4 avril 2022
- [31] [Flutter](https://www.ionos.fr/digitalguide/sites-internet/developpement-web/flutter-cest-quoi/), [https ://www.ionos.fr/digitalguide/sites-internet/developpement-web/flutter-cest-quoi/](https://www.ionos.fr/digitalguide/sites-internet/developpement-web/flutter-cest-quoi/) le 5 avril
- [32] [Flutter](https://mobiskill.fr/blog/conseils-emploi-tech/pourquoi-utiliser-flutter-en-2022/), [https ://mobiskill.fr/blog/conseils-emploi-tech/pourquoi-utiliser-flutter-en-2022/](https://mobiskill.fr/blog/conseils-emploi-tech/pourquoi-utiliser-flutter-en-2022/) le 5 avril 2022
- [33] [Avantages de flutter](https://mobiskill.fr/blog/conseils-emploi-tech/pourquoi-utiliser-flutter-en-2022/), [https ://mobiskill.fr/blog/conseils-emploi-tech/pourquoi-utiliser-flutter-en-2022/](https://mobiskill.fr/blog/conseils-emploi-tech/pourquoi-utiliser-flutter-en-2022/) Consulté le 5 avril 2022
- [34] [Avantages de flutter](https://www.hello-pomelo.com/articles/flutter-quel-avenir), [https ://www.hello-pomelo.com/articles/flutter-quel-avenir](https://www.hello-pomelo.com/articles/flutter-quel-avenir) Consulté le 5 avril 2022
- [35] [caractéristiques de Firebase](https://blog.back4app.com/fr/quest-ce-que-firebase/AvantagesdeFirebase), [https ://blog.back4app.com/fr/quest-ce-que-firebase/AvantagesdeFirebase](https://blog.back4app.com/fr/quest-ce-que-firebase/AvantagesdeFirebase) Consulté le 6 avril 2022

---