

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université A. Mira de Béjaïa  
Faculté des Sciences Exactes  
Département d'informatique



## Mémoire de fin d'étude

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

Option : Administration et sécurité des réseaux (ASR)

## *Thème*

---

*Conception et réalisation d'une  
application de gestion des patients d'un  
cabinet médical.*

---

**Encadrés par :**

Mme. EL BOUHISSI Houda.

**Réalisé par :**

M<sup>r</sup>. AROUR Laid

**Devant le jury composé de :**

**Présidente :** Mme. TAHAKOURT Zineb.

**Examineur :** Mr. MEHAOUED Kamal.

# Remerciements

*Avant tout commencement je remercie dieu le tout puissant de nous avoir donné la force, la patience et le courage durant toutes ses années d'études.*

*Je tiens à exprimer mes remerciements avec un grand plaisir et un grand respect à mon encadrante Dr EL BOUHISSI Houda, pour ses conseils, sa disponibilité et ses encouragements qui m'ont permis de réaliser ce travail dans les meilleures conditions.*

*Je remercie aussi tous les enseignants qui m'ont enseigné et encadré durant toutes ces années.*

*Je remercie aussi tous mes chers et mes proches qui ont été à mes côtés tout ce temps.*

# Dédicaces

*Je dédie ce travail à :*

*À la mémoire de tous mes chers*

*À mes très chers parents*

*À mes chères sœurs*

*À toute ma famille*

*À tous mes chers amis*

# **Table de matières**

## TABLES DES MATIERES

|                             |      |
|-----------------------------|------|
| Table des matières.....     | I    |
| Liste des figures.....      | V    |
| Liste des tableaux .....    | VII  |
| Liste des abréviations..... | VIII |
| Introduction général .....  | 1    |

## CHAPITRE 1 :ÉTUDE PRÉLIMINAIRE ET CAPTURE DES BESOINS

|   |   |
|---|---|
| 1.1 Introduction .....                                      | 3 |
| 1.2 Présentation de projet.....                             | 3 |
| 1.2.1 Gestion et suivi du dossier médical des patients..... | 3 |
| 1.2.2 Gestion des rendez-vous .....                         | 4 |
| 1.2.3 Gestion des Patients .....                            | 4 |
| 1.2.4 Gestion des ordonnances et médicaments.....           | 5 |
| 1.3 Problématique .....                                     | 5 |
| 1.4 Les besoins fonctionnelle .....                         | 6 |
| 1.5 Les besoins non fonctionnelle .....                     | 7 |
| 1.6 Proposition d'une Solution .....                        | 7 |
| 1.7 Méthodologie de développement .....                     | 7 |
| 1.7.1 Définition du Processus Unifié .....                  | 7 |
| 1.7.2 Phases du Processus Unifié .....                      | 7 |
| 1.7.3 Activités du processus unifié .....                   | 8 |

|  |    |
|--|----|
| 1.8 Langage de modélisation .....        | 8  |
| 1.8.1 UnifiedModelingLanguage (UML)..... | 8  |
| 1.8.2 Bases d'UML.....                   | 9  |
| 1.8.3 Diagrammes UML utilisés .....      | 9  |
| 1.9 Conclusion.....                      | 10 |

## CHAPITRE 2 : CONCEPTION

|  |    |
|--|----|
| 2.1 Introduction .....   | 11 |
| 2.2. Les acteurs du système .....                                | 11 |
| 2.2.1 Définition .....   | 11 |
| 2.2.2 Identification des acteurs.....                            | 11 |
| 2.2.3 Identification des cas d'utilisation .....                 | 12 |
| 2.3 Cas d'utilisation .....                                      | 13 |
| 2.4 Diagramme de cas d'utilisations .....                        | 13 |
| 2.4.1 Diagramme de cas d'utilisation de médecin.....             | 13 |
| 2.4.2 Diagramme de cas d'utilisation de secrétaire médicale..... | 14 |
| 2.4.3 Diagramme de cas d'utilisation de l'administrateur.....    | 15 |
| 2.4.4 Diagramme de cas d'utilisation global .....                | 15 |
| 2.5 Description textuelle des cas d'utilisations .....           | 17 |
| 2.6 Réalisation des diagrammes de séquence .....                 | 21 |
| 2.7 Réalisation du diagramme de classe.....                      | 30 |
| 2.7.1 Dictionnaire de données .....                              | 30 |
| 2.7.2 Diagramme de classes .....                                 | 31 |

|  |    |
|--|----|
| 2.7.3 Le modèle relationnel .....                    | 32 |
| 2.7.3.1 Règles de passage au modèle relationnel..... | 32 |
| 2.7.3.2 Modèle relationnel.....                      | 32 |
| 2.8 Conclusion.....                                  | 33 |

## CHAPITRE 3 : RÉALISATION

|   |    |
|---|----|
| 3.1 Introduction .....                        | 34 |
| 3.2 Application desktop .....                 | 34 |
| 3.3 Outils de développement .....             | 34 |
| 3.3.1 Développement des diagrammes .....      | 34 |
| 3.3.2 NetBeans .....                          | 35 |
| 3.3.3 PhpMyAdmin .....                        | 35 |
| 3.3.4 MYSQL .....                             | 36 |
| 3.4 Langage de programmation .....            | 36 |
| 3.4.1 JAVA.....                               | 36 |
| 3.5 Architecture global de l'application..... | 36 |
| 3.6 Les interface de l'application .....      | 37 |
| 3.6.1 Interface authentification .....        | 37 |
| 3.6.2 Interface gestion des patients .....    | 38 |
| 3.6.3 Interface consultation .....            | 39 |
| 3.6.4 Interface rend-vous .....               | 39 |
| 3.6.5 Interface médicament .....              | 40 |
| 3.6.6 interface ordonnance .....              | 40 |

|  |    |
|--|----|
| 3.6.7 Interface certificat .....                                 | 41 |
| 3.6.8 Interface gestion d'utilisateur .....                      | 42 |
| 3.6.9 interface gérer les informations d'un cabinet médical..... | 43 |
| 3.7 Conclusion .....   | 43 |
| Conclusion général .....   | 44 |
| Bibliographie.....   | 45 |



## LISTE DES FIGURES

|  |    |
|--|----|
| <b>1.1</b> : Historique d'UML .....  | 9  |
| <b>2.1</b> : diagramme de cas d'utilisation de l'acteur médecin.....                   | 14 |
| <b>2.2</b> : diagramme de cas d'utilisation de l'acteur secrétaire.....                | 14 |
| <b>2.3</b> : diagramme de cas d'utilisation d'administrateur .....                     | 15 |
| <b>2.4</b> : diagramme de cas d'utilisation globale.....                               | 16 |
| <b>2.5</b> : Diagramme de séquence du cas d'utilisation (Authentification).....        | 22 |
| <b>2.6</b> : Diagramme de séquence du cas d'utilisation (Gérer les consultations)..... | 23 |
| <b>2.7</b> : Diagramme de séquence du cas d'utilisation (Gérer les médicaments).....   | 24 |
| <b>2.8</b> : Diagramme de séquence du cas d'utilisation (Gérer les ordonnance).....    | 25 |
| <b>2.9</b> : Diagramme de séquence du cas d'utilisation (Établir Certificat).....      | 26 |
| <b>2.10</b> : Diagramme de séquence du cas d'utilisation (Gérer les patients).....     | 27 |
| <b>2.11</b> : Diagramme de séquence du cas d'utilisation (Gérer les rendez-vous).....  | 28 |
| <b>2.12</b> : Diagramme de séquence du cas d'utilisation (Gérer les utilisateurs)..... | 29 |
| <b>2.13</b> : Diagramme de classes.....  | 31 |
| <b>3.1</b> : Représentation de l'architecture globale de l'application.....            | 37 |
| <b>3.2</b> : Interface d'authentification.....   | 38 |
| <b>3.3</b> : Interface Gestion de patient.....   | 38 |
| <b>3.4</b> : Interface Gestion de la consultation.....                                 | 39 |
| <b>3.5</b> : Interface Gestion de rend-vous.....                                       | 39 |
| <b>3.6</b> : Interface Médicament.....   | 40 |
| <b>3.7</b> : Interface Gestion d'ordonnance.....                                       | 40 |
| <b>3.8</b> : Interface ordonnance.....   | 41 |

|   |    |
|---|----|
| <b>3.9</b> : Interface Gestion de certificat.....             | 41 |
| <b>3.10</b> : Interface certificat médicale.....              | 42 |
| <b>3.11</b> : Interface Gestion des utilisateurs.....         | 42 |
| <b>3.12</b> : Interface information d'un cabinet médical..... | 43 |

|  |    |
|--|----|
| <b>1.1</b> Descriptif des diagrammes UML utilisée.....                             | 9  |
| <b>2.1</b> Identification des cas d'utilisation.....                               | 11 |
| <b>2.2</b> Description textuelle du cas d'utilisation authentification.....        | 16 |
| <b>2.3</b> Description textuelle du cas d'utilisation gérer les consultations..... | 17 |
| <b>2.4</b> Description textuelle du cas d'utilisation gérer les ordonnances.....   | 17 |
| <b>2.5</b> Description textuelle du cas d'utilisation gérer les patients.....      | 18 |
| <b>2.6</b> Description textuelle du cas d'utilisation gérer les Rendez-vous.....   | 19 |
| <b>2.7</b> Description textuelle du cas d'utilisation gérer les médicaments.....   | 19 |
| <b>2.8</b> Description textuelle du cas d'utilisation certificat.....              | 19 |
| <b>2.9</b> Description textuelle du cas d'utilisation gérer utilisateurs.....      | 20 |
| <b>2.10</b> Représentatif du dictionnaire de données.....                          | 29 |

## LISTE DES ABRÉVIATIONS

- **BDD** Base De Données.
- **IDE** Integrated Development Environment..
- **MySql** My Structured Query Language.
- **UML** Unified Modeling language.
- **UP** Unified Process.
- **POO** Programmation orient objet.
- **SGBD** System de Gestion de Base de Données.
- **SQL** Structured Query Language.

# **Introduction générale**

## Introduction générale

Actuellement, le monde connaît une avance 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 développement du travail.

Avant l'invention de l'ordinateur, on enregistrait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tels que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières...etc.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de données des entreprises, ce qui est la partie essentielle dans leur développement aujourd'hui.

Les cabines médicales font partie intégrante des établissements que l'informatique pourra beaucoup aider. Jusqu'à ce jour, la manière de gérer manuellement est encore dominante, d'où la nécessité d'introduire l'informatique dans les gestions et les organisations.

Le but de notre mémoire de fin d'étude est de concevoir et réaliser une application pour la gestion d'un cabinet médical.

Ainsi, notre mémoire est organisé en trois principaux chapitres :

- Le premier chapitre « Étude préliminaire et capture des besoins » est consacré à la présentation de notre projet. La spécification des besoins nous permettra de délimiter notre problématique et les objectifs visés, tout en introduisant la méthodologie adoptée et les étapes à suivre pour concevoir une application de gestion des patients.
- Le deuxième chapitre porte sur la conception, il regroupe toutes les étapes de notre processus de développement en utilisant le langage de modélisation UML et le processus unifié UP.

- Le troisième et dernier chapitre est consacré à la réalisation où nous allons définir tous les outils qui nous ont permis de réaliser notre application et quelques interfaces y seront présentées.

Enfin, notre mémoire se termine par une conclusion générale.

# **1. Étude préliminaire et capture des besoins**



## **1.1 Introduction :**

Dans ce chapitre, nous allons présenter le contexte de notre projet de fin d'étude notamment les problèmes relatifs à la gestion des patients au sein d'un cabinet médical. Aussi, nous allons décrire la solution proposée et la méthodologie de travail adoptée pour développer et mettre en œuvre notre solution.

## **1.2 Présentation de projet:**

Notre mission dans le cadre de ce projet est de créer une application (Desktop) permettant de gérer les patients d'un cabinet médical. Il s'agit de définir les responsabilités de la gestion, mettre à jour les données, organiser des données collectées afin de concevoir des fichiers de bases pour le médecin, de renforcer le contrôle et la confrontation, assurer une meilleure gestion médicale des patients et une cohérence de l'information et enfin faciliter le travail des responsables.

Notre application aura comme principale fonctionnalités :

- Gestion et suivi du dossier médical des patients.
- Gestion des Rendez-vous des patients.
- Gestion des Patients.
- Gestion des ordonnances et médicaments.

### **1.2.1 Gestion et suivi du dossier médical des patients :**

La consultation est l'activité principale du cabinet médical, le patient arrive à un cabinet médical pour la première fois, pour se faire consulter par le médecin, ce dernier lui amène la fiche médicale descriptive ainsi que son dossier médical.

L'écoute attentive et patiente des propos du patient est un moment privilégié de la consultation.

L'entretien doit se dérouler dans la stricte intimité et confidentialité pour permettre au patient de s'exprimer clairement et sincèrement sur ses préoccupations.

Ensuite, le médecin l'examine à l'aide de ses outils (Stéthoscope, Tensiomètre, Thermomètre, etc.), il rédige l'ordonnance qui contient les noms des médicaments, la quantité et la consommation.

Parfois, le médecin peut demander au patient de faire des examens complémentaires (Bilan biologique ou Bilan radiologique), il fixe un autre rendez-vous au patient pour attribuer le diagnostic adéquat en fonction des résultats.

A chaque consultation, et selon l'état de santé du patient, le médecin peut délivrer un certificat médical justifiant un arrêt d'activité.

En cas de visite de contrôle ou visite périodique d'un patient en maladie de longue durée, celle-ci sera programmée à un moment précis de la journée.

A chaque consultation, un résumé de la nouvelle consultation et du traitement donné seront porté sur le dossier médical.

### **1.2.2 Gestion des rendez-vous :**

Il peut être nécessaire d'organiser sa consultation sur rendez-vous si le besoin s'en fait sentir et le médecin se doit de les respecter scrupuleusement, le cas échéant, ceci doit être signalé aux patients. Cependant, il faut tenir compte des urgences qui ne peuvent souffrir aucune attente et admettre également la souplesse et la disponibilité requises.

La prise d'un rendez-vous s'effectue directement ou par une communication téléphonique en donnant le nom, le prénom, la date et l'heure souhaitée. La secrétaire est chargée de remplir les renseignements sur la fiche d'un patient (Nom, Prénom ...).

### **1.2.3 Gestion des Patient:**

S'il s'agit d'un ancien patient, la secrétaire demande le nom et prénom pour effectuer la recherche de sa fiche parmi les fiches médicales.

La fiche du patient doit contenir les informations suivantes : matricule, nom, prénom, date de naissance, adresse, sexe, téléphone.

L'observation médicale rédigée par le médecin doit comprendre les antécédents du patient qui sont les données de son terrain (chirurgicaux, familiales, allergie, maladie chronique...).

Ces données sont capitales pour les consultations ultérieures et toutes thérapeutiques.

### **1.2.4 Gestion des ordonnances et médicaments :**

Après l'examen médical, le médecin prescrit un traitement au malade, sur une ordonnance. L'ordonnance comporte les différents médicaments avec leurs formes, consommation, quantité.

### 1.3 Problématique :

En général, les cabines médicales ont des difficultés pour gérer et organiser les patients.

Parmi ces difficultés, nous citons :

- Absence d'un moyen de recherche rapide : pour chercher une fiche, le médecin doit faire une recherche manuelle, en parcourant son archive selon le nom du patient, ce qui engendre une perte de temps. Ainsi, cette recherche est parfois en face du risque que les fiches peuvent se mélanger et surtout leurs contenus. Aussi, la perte de temps qui est remarquable en cas d'augmentation du nombre des patients pour la consultation.
- La gestion des rendez-vous, se fait d'une manière manuelle ce qui engendre un risque d'oubli ou chevauchement des rendez-vous.
- Absence de la notion de confidentialité à cause de la non séparation entre fiche médicale et dossier médical : la secrétaire peut accéder aux informations confidentielles du patient, or le respect du secret médical impose que seul le médecin peut consulter ce dossier.

L'informatique est presque devenu indispensable au niveau de tous les domaines dans notre vie courante, et en considérant les problèmes cités ci-dessus, nous proposons la conception et la réalisation d'une application pour répondre aux besoins d'un cabinet médical.

### 1.4 Les besoins fonctionnels :

Les besoins fonctionnels se rapportent aux fonctionnalités que l'application doit offrir pour satisfaire les utilisateurs.

Les fonctionnalités que doit intégrer l'application à développer sont :

**Gestion des consultations :** Cette fonctionnalité consiste à suivre l'état des patients à savoir :

- La fiche de consultation (diagnostique).
- D'ajouter, modifier et supprimer une consultation.

**Gestion des ordonnances et médicaments :** Cette fonctionnalité permet au médecin lui donne un traitement (le médicament) après le diagnostic (une ordonnance d'aller acheter) ou un certificat descriptif de l'état de santé de patient.

Le système imprime une ordonnance ou certificat au patient.

**Gestion des patients :** cette fonctionnalité permet à la secrétaire médicale :

- D'ajouter, modifier et supprimer des patients.

**Gestion des rendez-vous :** cette fonctionnalité permet à la secrétaire médicale :

- D'ajouter, modifier et supprimer des rend-vous.

**Gestion des utilisateurs :** cette fonctionnalité permet d'ajouter, modifier, supprimer un utilisateur.

## 1.5 Les besoins non fonctionnels :

Les besoins non fonctionnels sont indispensables et permettent l'amélioration de la qualité de notre application. Ils agissent comme des contraintes sur les solutions, mais leur prise en considération fait éviter plusieurs incohérences dans le système. Ce dernier doit répondre aux exigences suivantes :

- **Authentification :** le système doit permettre à l'utilisateur de saisir son login et son mot de passe pour accéder aux différentes fonctionnalités. Cette fonctionnalité assure la sécurité du système.
- **Ergonomie :** le système devra offrir aux utilisateurs une interface qui soit la plus conviviale possible afin de limiter le nombre d'écrans. Par ailleurs, l'interactivité devra être adaptée (usage du clavier, menu, etc.).
- **Fiabilité :** le système doit être fiable (l'utilisateur doit avoir confiance en la qualité de son produit, pour mieux s'occuper du malade tant le domaine est sensible).

## 1.6 Proposition d'une Solution :

En réponse aux anomalies citées précédemment, nous proposons une solution qui consiste à concevoir et à développer une application qui facilitera les insuffisances et les défaillances. Notre application permettra l'organisation et l'automatisation des tâches.

L'informatisation de la gestion des patients d'un cabinet a des avantages certains :

- La mise en place d'une application permet d'optimiser le temps d'accès aux différentes données, éviter les tâches pénibles et ennuyeuses. Elle permet entre autre de définir une bonne organisation des données collectées pour faciliter la recherche des documents, aider le médecin pour la prise de décision avec des supports informatisés à l'appui.
- Le système à réaliser permet la gestion des médicaments de façon rapide pour avoir des informations en temps réel.
- Gérer les droits d'accès afin de permettre un accès sélectif aux différents menus et attribuer des responsabilités à chaque utilisateur : on doit assurer la séparation entre fiche médicale et dossier médical seul le médecin peut consulter le dossier médical.

## **1.7 Méthodologie de développement :**

Selon la nature de notre projet, nous avons jugés que PU (processus unifié) serait le plus adapté à la mise en œuvre de notre application.

### **1.7.1 Définition du Processus Unifié :**

PU ou UP (Unified Process), est une méthode orientée objets pour le développement de logiciel, c'est une méthode générique itérative et incrémental, pilotée par les cas d'utilisation et centrée sur l'architecture [2].

Le processus unifié fournit un cadre au développement logiciel pour la construction de systèmes orientés objet.

### **1.7.2 Phases du Processus Unifié :**

La méthode PU se base sur quatre phases :

- **Analyse des besoins** : Établir une vision globale du projet où on spécifie les besoins et on étudie la faisabilité du projet.
- **Élaboration** : On reprend les éléments de l'analyse des besoins et on développe une architecture de référence, les risques et la plupart des besoins sont identifiés.
- **Construction** : Finaliser l'analyse, la conception, l'implémentation et les tests puis transformer l'architecture de référence en produit exécutable tout en veillant à respecter son intégrité.

- **Transition** : Livraison du produit au client afin d'effectuer des essais pour détecter d'éventuelles anomalies.

### 1.7.3 Activités du processus unifié :

Chaque phase est constituée d'une succession d'activités. Les activités de la méthode UP sont [2] :

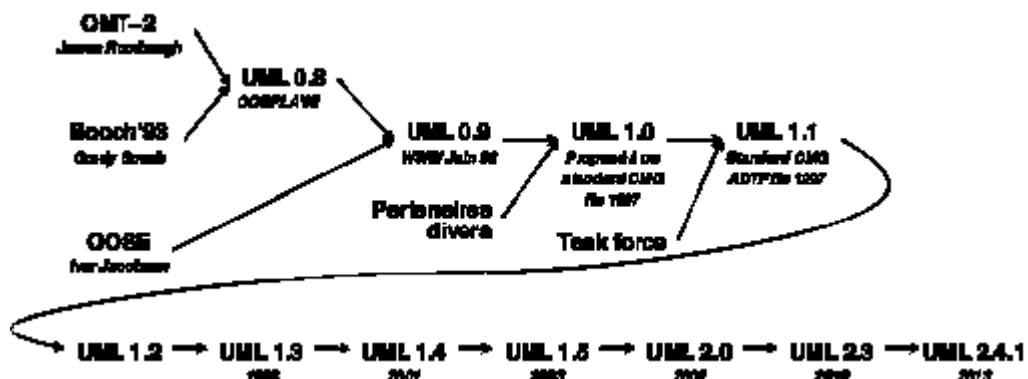
- **Expression des besoins** : Compréhension et expression des besoins et des exigences du client qu'elle soit fonctionnelle ou non fonctionnelle.
- **Analyse et Conception** : Permet d'acquérir une compréhension approfondie des contraintes liées aux outils de réalisation en prenant en compte le choix d'architecture technique retenu pour le développement et l'exploitation système.
- **Implémentation** : On implémente le système sous forme de composants, bibliothèques et de fichiers. Elle a pour objectif de planifier l'intégration.
- **Tests** : Permettent de vérifier les résultats de l'implémentation de toutes les exigences et de s'assurer de la bonne intégration de tous les composants dans le logiciel.
- **Déploiement** : Livraison et exploitation du produit.

## 1.8 Langage de modélisation :

Sachant que le processus unifié exige l'utilisation d'UML, notre modélisation de la solution se fera en utilisant des diagrammes UML.

### 1.8.1 Unified Modeling Language (UML):

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.[3]



**Figure 1.1:** Historique d'UML.**1.8.2 Bases d'UML**

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

Ce langage est certes issu du développement logiciel mais pourrait être appliqué à toute science fondée sur la description d'un système. Dans l'immédiat, UML intéresse fortement les spécialistes de l'ingénierie système. [4]

UML unifie également les notations nécessaires aux différents activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la spécification jusqu'au codage.

Dans ce cadre, un concept appartenant aux besoins des utilisateurs projette sa réalité dans le modèle de conception et dans le codage. [4]

**1.8.3 Diagrammes UML utilisés :**

Pour modéliser notre système, nous avons utilisé les trois types de diagrammes suivants:

| Diagramme            | Objectifs  | Type     |
|----------------------|--|----------|
| Diagramme de classes | 1. Point central de la modélisation du système pour décrire ce que le système doit faire (analyse) et avec quoi il va le faire (conception).<br>2. Représentation de la structure statique | Statique |

|                                |  |               |
|--------------------------------|--|---------------|
|                                | du système d'information.<br>3. Modélisation des classes et de leurs relations.  |               |
| Diagramme de cas d'utilisation | 1. Décrire la manière dont une organisation ou un système externe doivent interagir avec le système.<br>2. décrire ce que doit faire le système.<br>3. Mettre en évidence les services rendus par le système           | fonctionnelle |
| Diagramme de séquence          | 1. validation des cas d'utilisation pour comprendre la logique de l'application.<br>2. Complète le diagramme des cas d'utilisation en mettant en évidence les objets et leurs interactions d'un point de vue temporel. | dynamique     |

**Tableau 1.1** Descriptif des diagrammes UML utilisée.

## 1.9 Conclusion :

Dans ce premier chapitre, nous avons présenté et identifié notre projet. Ensuite, nous avons déterminé les différents problèmes rencontrés, et les besoins fonctionnels et non fonctionnels, puis nous avons opté une méthodologie pour notre application.

Dans le chapitre suivant, nous servira de base pour la conception de notre application qui fera l'objet du prochain chapitre.



## **2. Conception**

## 2.1 Introduction :

Dans le cycle de vie d'un projet, la conception représente une phase primordiale et déterminante pour produire une application de qualité. Dans ce chapitre dédié à la conception, nous allons définir le rôle de chaque acteur qui interagit avec le système. Nous allons aussi modéliser leurs rôles sous forme de diagrammes de cas d'utilisation, puis nous définirons les cas d'utilisation, de séquence et de classe pour passer enfin à l'élaboration de la base de données.

## 2.2. Les acteurs du système :

### 2.2.1 Définition :

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages éventuellement porteurs de données [5].

### 2.2.2 Identification des acteurs:

- **Médecin :**

Cet acteur représente la personne qui diagnostique et traite les maladies. Un médecin est un professionnel de la santé titulaire d'un diplôme de docteur en médecine, son rôle au sein de notre application se résume comme suit :

- Gestion des consultations.
- Gestion des ordonnances et médicaments.
- Ajouter un certificat.

- **Secrétaire (Assistante médicale) :**

Cet acteur est le pivot de la relation entre le médecin, le patient et l'institution. La secrétaire est la collaboratrice directe du médecin et le lien privilégié du patient ; elle occupe un poste clé dans le fonctionnement de toute organisation médicale et assume plusieurs tâches qui peuvent être différentes d'une structure à une autre. Les tâches d'une secrétaire se résument comme suit :

- Gestion des patients.
- Gestion des rendez-vous.

### L'administrateur :

Le seul acteur ayant le droit d'accès, la mise à jour de l'application et à sa maintenance en cas de panne, gérer les secrétaires et les médecins.

### 2.2.3 Identification des cas d'utilisation :

Dans l'application à développer, nous avons identifié les cas d'utilisation suivant (Tableau 2.1) :

| N° | Acteurs                | Cas d'utilisation   |
|----|------------------------|---|
| 1  | Secrétaire/médecin     | <ul style="list-style-type: none"> <li>▪ Authentification (saisie du nom, du mot de passe et type d'utilisateur).</li> </ul>  |
| 2  | Médecin/administrateur | <ul style="list-style-type: none"> <li>▪ Gestion des consultations (ajouter, modifier, supprimer, voir l'historique des consultations)</li> <li>▪ Gestion des ordonnances (sélectionner le patient, ajouter un médicament, imprimer, supprimer)</li> <li>▪ Ajouter un certificat (sélectionner le patient, état de santé, imprimer la certificat).</li> <li>▪ Gestion des médicaments (ajouter, modifier, supprimer).</li> <li>▪ Gestion des utilisateurs.</li> </ul> |
| 3  | Secrétaire             | <ul style="list-style-type: none"> <li>▪ Gestion des patients (ajouter, modifier, supprimer, annuler, liste des patients).</li> <li>▪ Gestion des rend-vous (ajouter, modifier, supprimer, annuler, liste des rend-vous).</li> </ul>  |

**Tableau 2.1** : Identification des cas d'utilisation.

### 2.3 Cas d'utilisation :

- ❖ **Authentification** : Cette fonctionnalité permet à un utilisateur de s'authentifier avant d'accéder à l'application, elle permet aussi de s'assurer de l'identité de l'utilisateur.
- ❖ **Gestion des consultations** : offre la possibilité au médecin d'enregistrer, modifier et supprimer une consultation.
- ❖ **Gestion des ordonnances et médicaments** : permet au médecin d'ajouter des médicaments aux ordonnances, la quantité, la consommation et d'ajouter un certificat.
- ❖ **Gestion des patients** : permet au secrétaire médicale d'ajouter, annuler, rechercher, modifier, ou supprimer une fiche du patient.
- ❖ **Gestion des rendez-vous** : permet au secrétaire médicale d'ajouter ou annuler un rendez-vous et afficher la date et l'heure de rendez-vous.
- ❖ **Gestion d'utilisateur** : cette fonctionnalité permet de gérer les utilisateurs à savoir, ajouter, modifier ou supprimer un utilisateur.

### 2.4 Diagramme des cas d'utilisation :

#### Définition :

Les diagrammes des cas d'utilisation décrivent les utilisations requises d'un système, ou ce qu'un système est supposé faire. Les principaux concepts de ces diagrammes sont les acteurs, cas d'utilisation et sujets. Un sujet représente un système avec lequel les acteurs et autres sujets interagissent. Le comportement requis du sujet est décrit par les cas d'utilisation [6].

#### 2.4.1 Diagramme de cas d'utilisation de médecin:

Le diagramme ci-dessous représente les cas d'utilisations identifiés pour le médecin.

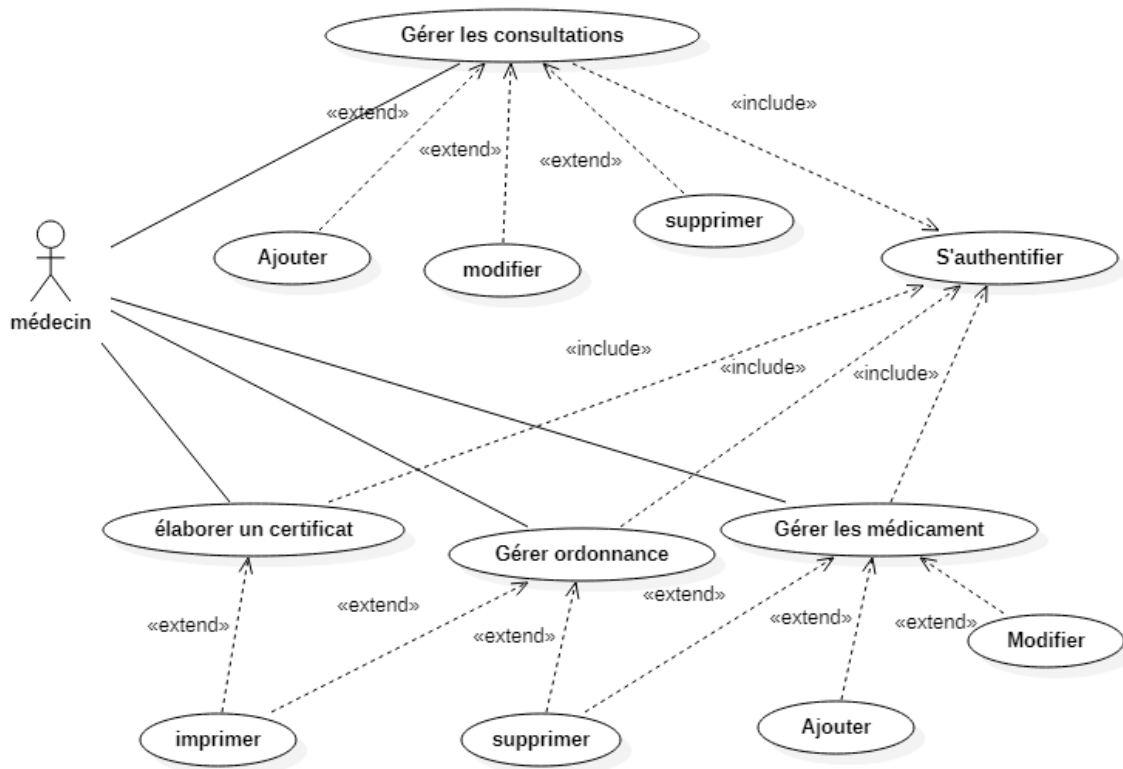


Figure2.1 : diagramme de cas d'utilisation de l'acteur médecin.

**2.4.2 Diagramme de cas d'utilisation de la secrétaire médicale :**

Le diagramme ci-dessous représente les cas d'utilisations identifiés pour la secrétaire médicale.

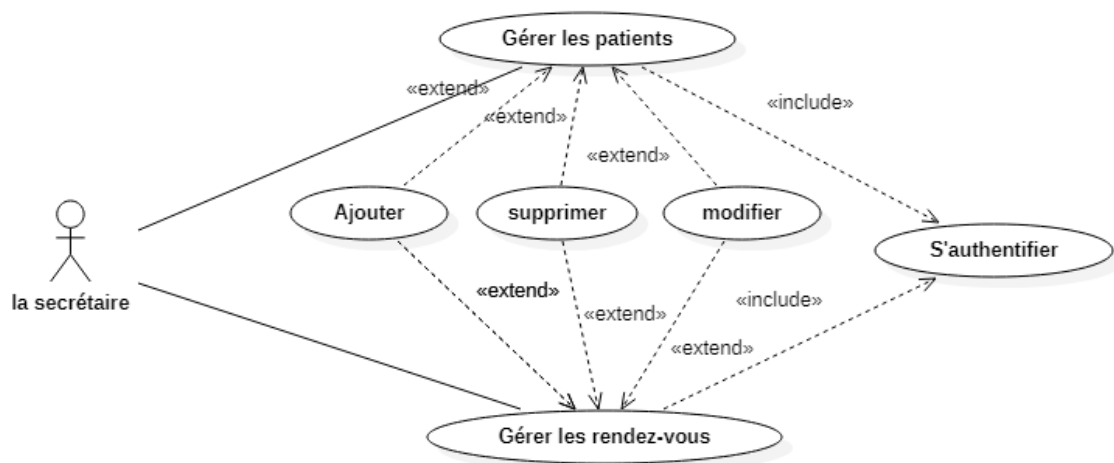


Figure 2.2 : diagramme de cas d'utilisation de l'acteur secrétaire.

### 2.4.3 Diagramme de cas d'utilisation de l'administrateur:

Le diagramme ci-dessous représente les cas d'utilisations identifiés pour l'administrateur.

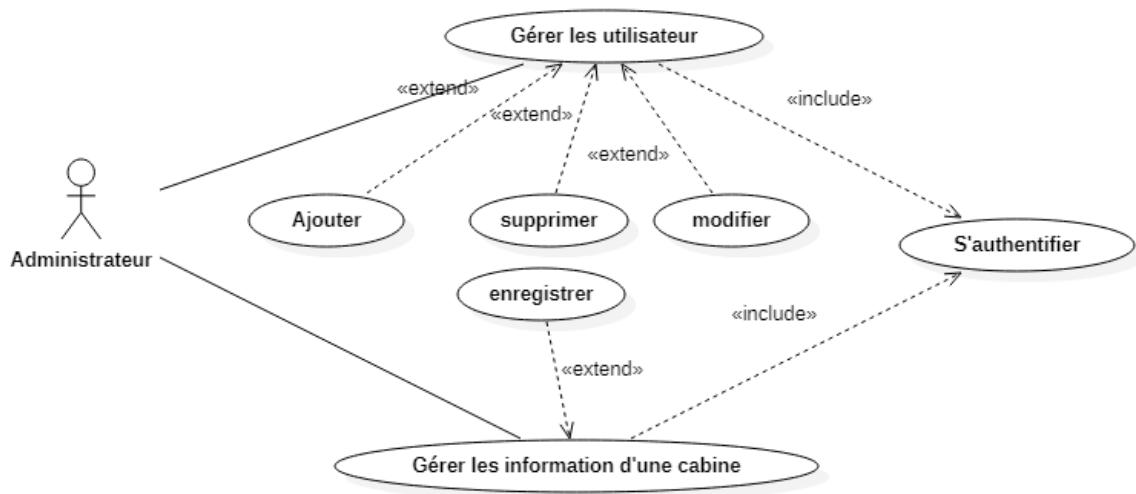


Figure 2.3 : diagramme de cas d'utilisation de l'administrateur.

**2.4.4) Diagramme de cas d'utilisation global :** Le diagramme de cas d'utilisation global représente les différentes fonctions de notre application.

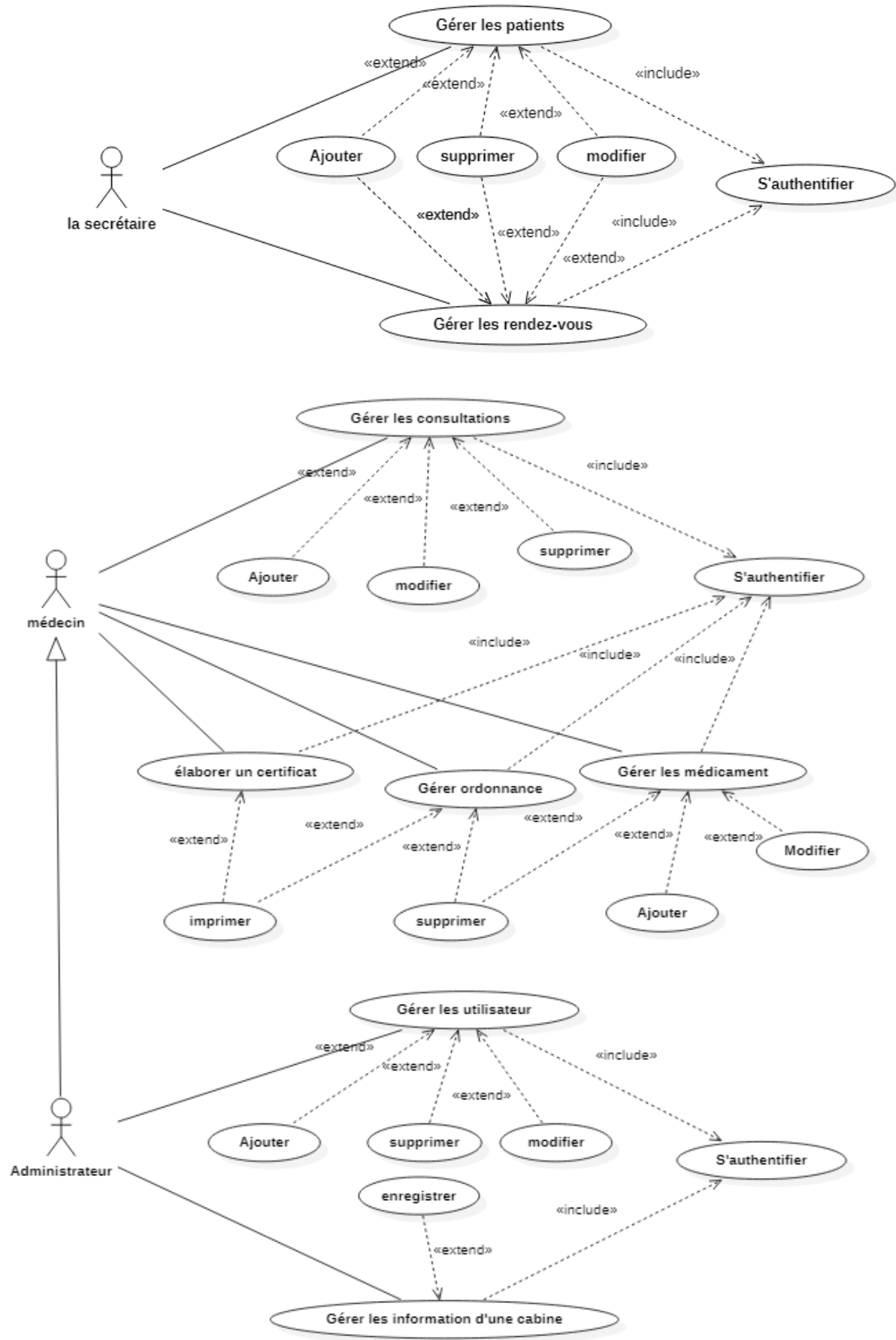


Figure 2.4 : diagramme de cas d'utilisation globale.

## 2.5 Description textuelle des cas d'utilisations :

### Cas d'utilisation N°1 : S'authentifier

|                   |  |
|-------------------|--|
| Acteur principale | Médecin/secrétaire médicale  |
| Objectif          | S'authentifier avant d'accéder à l'application.  |
| Scénario          | <ol style="list-style-type: none"> <li>1. L'utilisateur lance l'application ;</li> <li>2. Le système demande à l'utilisateur de s'authentifier.</li> <li>3. L'utilisateur saisit son nom, son mot de passe, et le type d'utilisateur.</li> <li>4. Le système vérifie la conformité des informations saisies en envoyant une requête à la base des données.</li> <li>5. La requête est vérifiée par le système et une réponse favorable est envoyée.</li> <li>6. L'utilisateur accède au menu principal.</li> </ol> |
| Alternative       | En cas de réponse défavorable, le système affiche un message d'erreur en cas d'erreur de saisie ou bien d'un champ incomplet (retour à 3).   |

**Tableau 2.2** : Description textuelle du cas d'utilisation Authentification.

### Cas d'utilisation N°2 : Gérer les consultations

|                   |   |
|-------------------|---|
| Acteur principale | Médecin   |
| Objectif          | Consulter le patient  |
| Précondition      | Authentification  |
| Scénario          | <ol style="list-style-type: none"> <li>1. Le médecin accède à l'interface gérer la consultation</li> <li>2. Le médecin : <ul style="list-style-type: none"> <li>• Saisie toute les informations de consultation, puis clique sur ajouter.</li> <li>• Sélectionne une consultation puis clique sur supprimer.</li> <li>• Sélectionne un patient et modifie les informations puis clique sur modifier.</li> </ul> </li> <li>3. Le système envoie la requête à la Base de données.</li> <li>4. Le système affiche le message de succès.</li> </ol> |



|             |  |
|-------------|--|
|             |  |
| Alternative | Le système affiche un message en cas d'une erreur de saisie ou bien d'un champ incomplet (retour a -2-). |

**Tableau 2.3** : Description textuelle du cas d'utilisation Gérer les consultations.

**Cas d'utilisation N°3 : Gérer les ordonnances :**

|                   |  |
|-------------------|--|
| Acteur principale | Médecin  |
| Objectif          | Établir une ordonnance aux patients.   |
| Précondition      | Authentification   |
| Scénario          | <ol style="list-style-type: none"> <li>1. Le médecin accède à l'interface gérer les ordonnances.</li> <li>2. Le médecin : <ul style="list-style-type: none"> <li>• Sélectionne le nom des médicaments et toute information utile (nombre de fois, quantité, ...).</li> <li>• Sélectionne une ordonnance puis clique sur afficher ou imprimer.</li> </ul> </li> <li>3. Le système envoie la requête à la base des données.</li> <li>4. Le système affiche le message de succès.</li> <li>5. imprimer l'ordonnance.</li> </ol> |
| Alternative       | Le système affiche un message d'erreur en cas d'une erreur de saisie ou bien d'un champ incomplet (retour a -2-).  |

**Tableau 2.4** : Description textuelle du cas d'utilisation Gérer les ordonnances

**Cas d'utilisation N°4 : Gérer les patients**

|                   |  |
|-------------------|--|
| Acteur principale | La secrétaire médicale.  |
| Objectif          | Enregistrer la fiche patients.   |
| Précondition      | Authentification   |
| Scénario          | <ol style="list-style-type: none"> <li>1. La secrétaire accède à l'interface gérer les patients.</li> <li>2. La secrétaire : <ul style="list-style-type: none"> <li>• Saisie le nom du patient et toute l'affiliation (prénom, date de naissance, adresse ...), puis clique sur ajouter.</li> <li>• Sélectionne un patient puis clique sur supprimer.</li> <li>• Sélectionne un patient et modifie les champs puis clique sur modifier.</li> </ul> </li> <li>3. Le système envoie la requête à la base de données.</li> <li>5. Le système affiche le message de succès.</li> </ol> |
| Alternative       | Le système affiche un message en cas d'une erreur de saisie ou bien d'un champ incomplet (retour a -2-).   |

**Tableau 2.5** : Description textuelle du cas d'utilisation Gérer les patients**Cas d'utilisation N°5 : Gestion des Rendez-vous.**

|                   |  |
|-------------------|--|
| Acteur principale | La secrétaire médicale   |
| Objectif          | Le patient prend un rendez-vous pour une consultation.   |
| Précondition      | Authentification   |
| Scénario          | <ol style="list-style-type: none"> <li>1. La secrétaire accède à l'interface gérer les rendez-vous.</li> <li>2. La secrétaire : <ul style="list-style-type: none"> <li>• Sélectionne le patient puis ajoute la date et l'heure du rendez-vous</li> <li>• Sélectionne le patient puis clique sur supprimer.</li> <li>• Sélectionne un patient et modifie les champs puis clique sur modifier.</li> </ul> </li> <li>3. Le système envoie la requête à la base de données.</li> <li>4. Après la vérification dans la base de données.</li> <li>5. Le système affiche le message de succès.</li> </ol> |

|             |  |
|-------------|--|
|             |  |
| Alternative | Le système affiche un message en cas d'une erreur de saisie ou bien d'un champ incomplet (retour à -2-). |

**Tableau 2.6** : Description textuelle du cas d'utilisation Gérer les Rendez-vous.

### Cas d'utilisation N°6: Gestion des médicaments.

|                   |  |
|-------------------|--|
| Acteur principale | Médecin  |
| Objectif          | Gérer les médicaments pour ajouter à l'ordonnance.   |
| Précondition      | Authentification   |
| Scénario          | <ol style="list-style-type: none"> <li>1. Le médecin accède à l'interface médicament.</li> <li>2. Le médecin : <ul style="list-style-type: none"> <li>• Saisie le nom du médicament puis clique sur ajouter.</li> <li>• Sélectionne un médicament puis clique sur supprimer</li> </ul> </li> <li>3. Le système envoie la requête à la base de données.</li> <li>4. Le système affiche le message de succès.</li> </ol> |
| Alternative       | Le système affiche un message en cas d'une erreur de saisie ou bien d'un champ incomplet (retour à -2-).   |

**Tableau 2.7** : Description textuelle du cas d'utilisation Gérer les médicaments.

### Cas d'utilisation N°7: Établir un certificat.

|                   |  |
|-------------------|--|
| Acteur principale | Médecin  |
| Objectif          | Imprimer un certificat médical au patient.   |
| Précondition      | Authentification   |
| Scénario          | <ol style="list-style-type: none"> <li>1. Le médecin accède à l'interface médicament.</li> <li>2. Le médecin saisie l'état de santé de patient puis clique sur imprimer.</li> <li>3. Le système envoie la requête à la base de données.</li> <li>4. Le système affiche le certificat.</li> </ol> |
| Alternative       | Le système affiche un message en cas d'une erreur de saisie ou bien d'un champ incomplet (retour à -2-).   |

**Tableau 2.8** : Description textuelle du cas d'utilisation certificat.

**Cas d'utilisation N°8: Gérer les utilisateurs.**

|                   |   |
|-------------------|---|
| Acteur principale | Médecin   |
| Objectif          | Gérer les utilisateurs du système.  |
| Précondition      | Authentification  |
| Scénario          | <ol style="list-style-type: none"> <li>1. Le médecin (administrateur) accède à l'interface paramètre.</li> <li>2. Le médecin : <ul style="list-style-type: none"> <li>• Saisie le nom et mot de passe et leur rôle puis cliquer sur ajouter.</li> <li>• Sélectionne un utilisateur puis clique sur supprimer.</li> <li>• Sélectionne un utilisateur et modifie les champs puis clique sur modifier.</li> </ul> </li> <li>3. Le système envoie la requête à la base de données.</li> <li>4. Le système affiche le certificat.</li> </ol> |
| Alternative       | Le système affiche un message en cas d'une erreur de saisie ou bien d'un champ incomplet (retour à -2-).  |

**Tableau 2.9** : Description textuelle du cas d'utilisation Gérer les utilisateurs.**2.6 Réalisation des diagrammes de séquence :**

- **Diagramme de séquence N°1 : cas d'utilisation Authentification :**

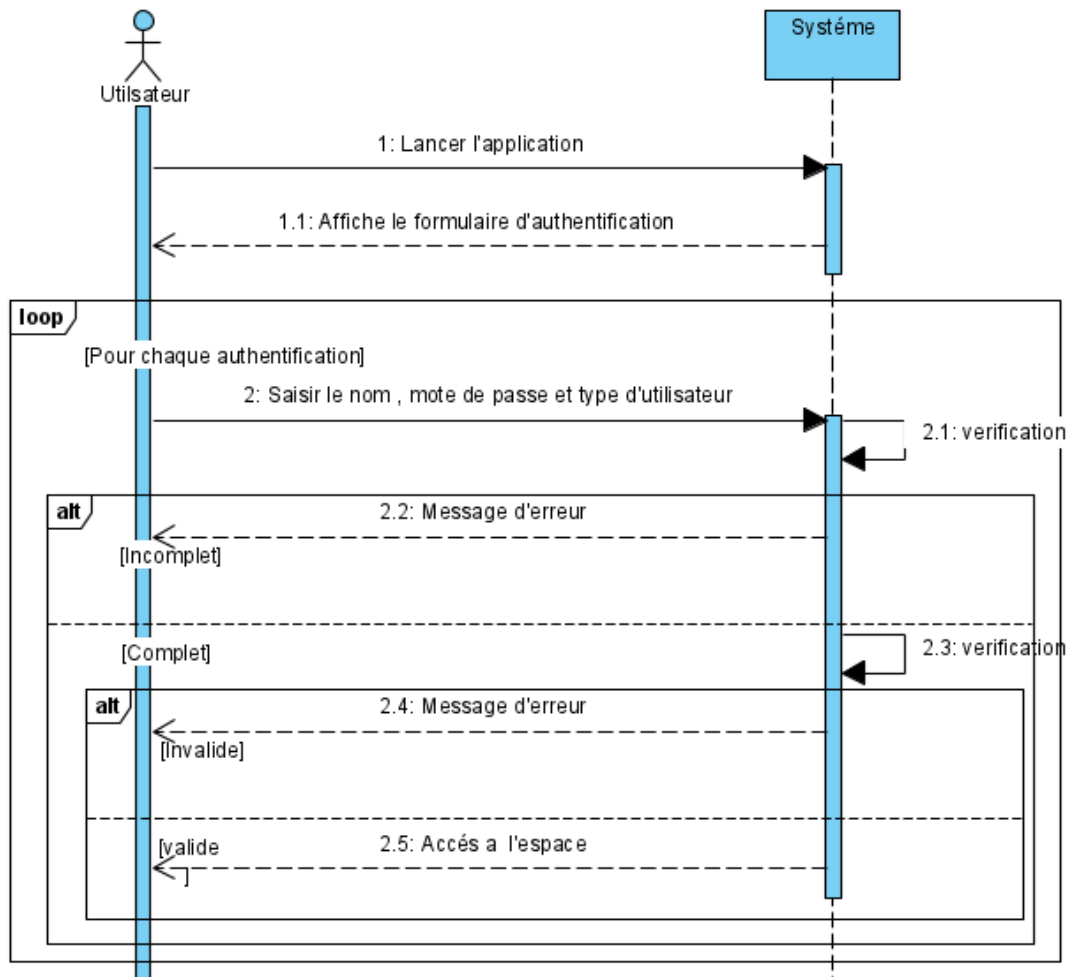


Figure 2.5 : Diagramme de séquence du cas d'utilisation (Authentification).

- Diagramme de séquence N°2 : cas d'utilisation Gérer les consultations :

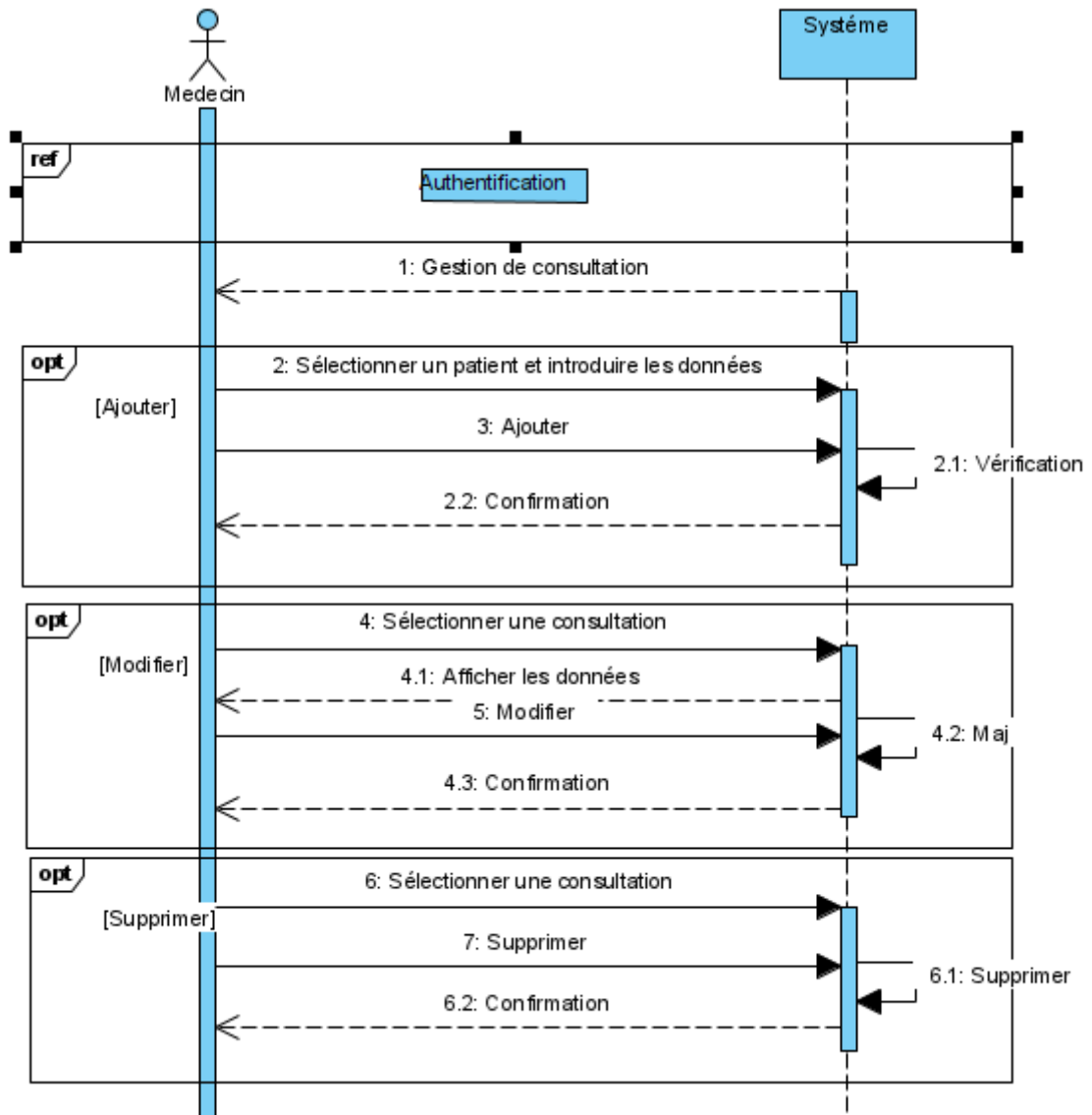


Figure 2.6 : Diagramme de séquence du cas d'utilisation (Gérer les consultations).

- Diagramme de séquence N°3 : cas d'utilisation gérer les médicaments :

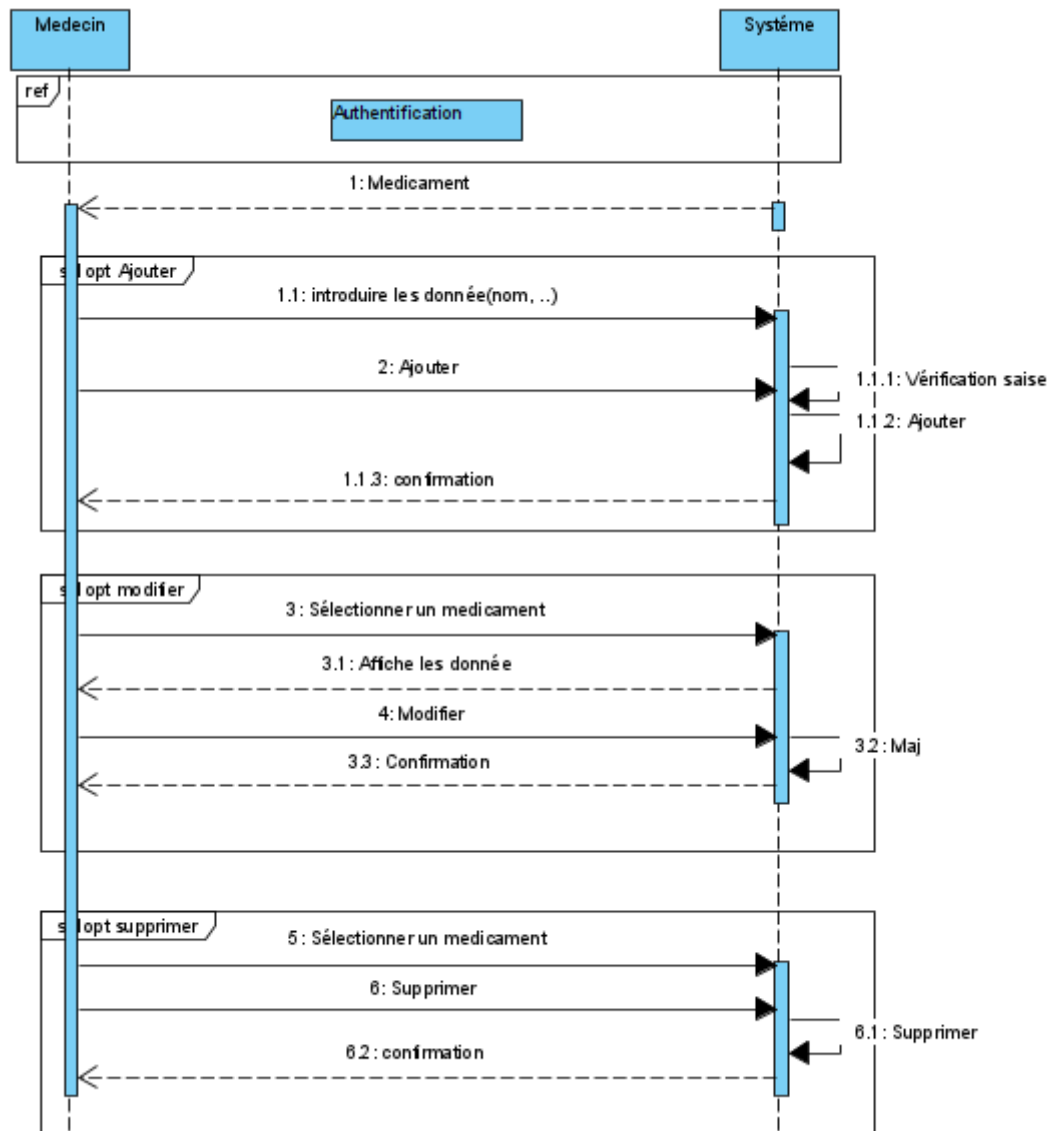


Figure 2.7 : Diagramme de séquence du cas d'utilisation (Gérer les médicaments).

Diagramme de séquence N°4 : cas d'utilisation gérer les ordonnances :

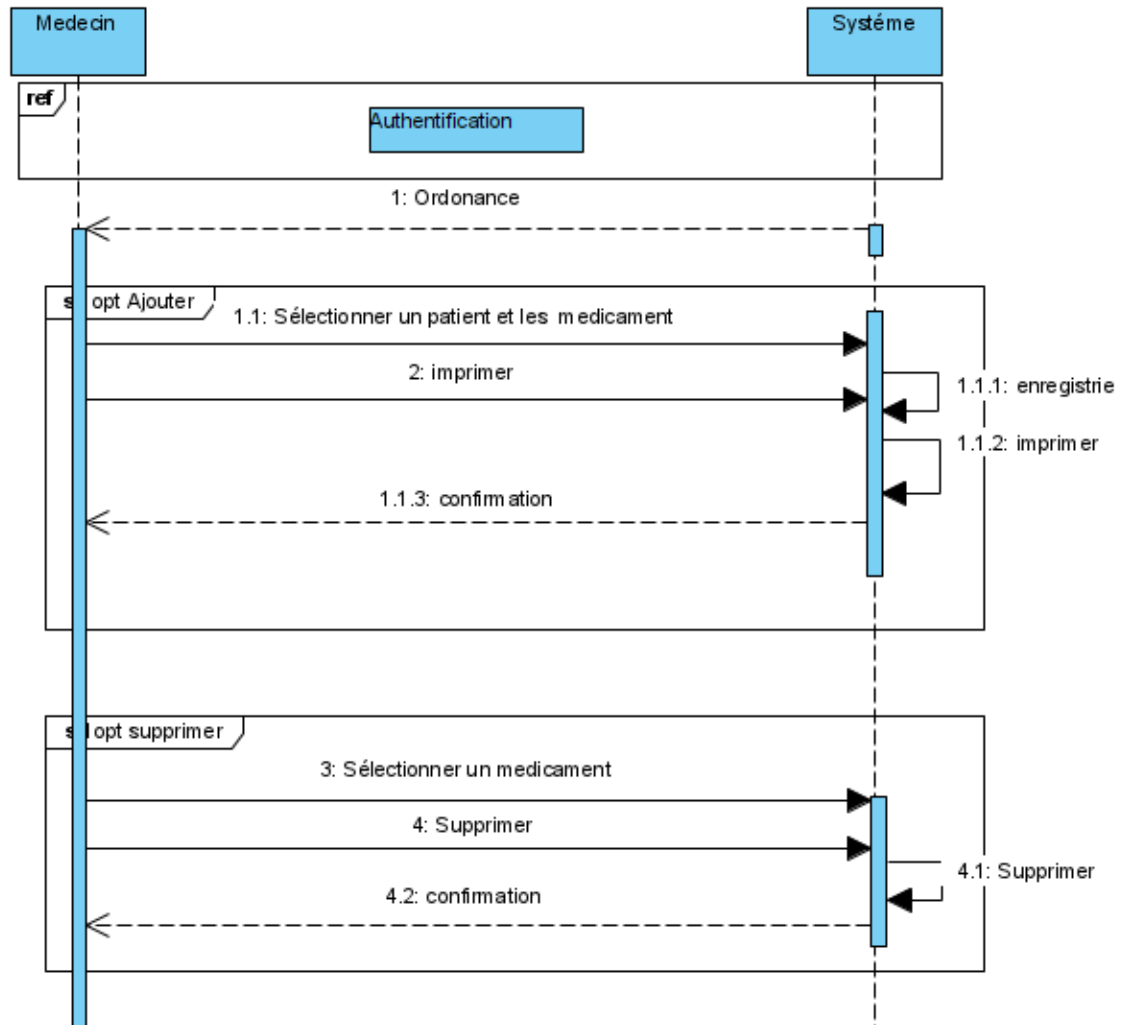
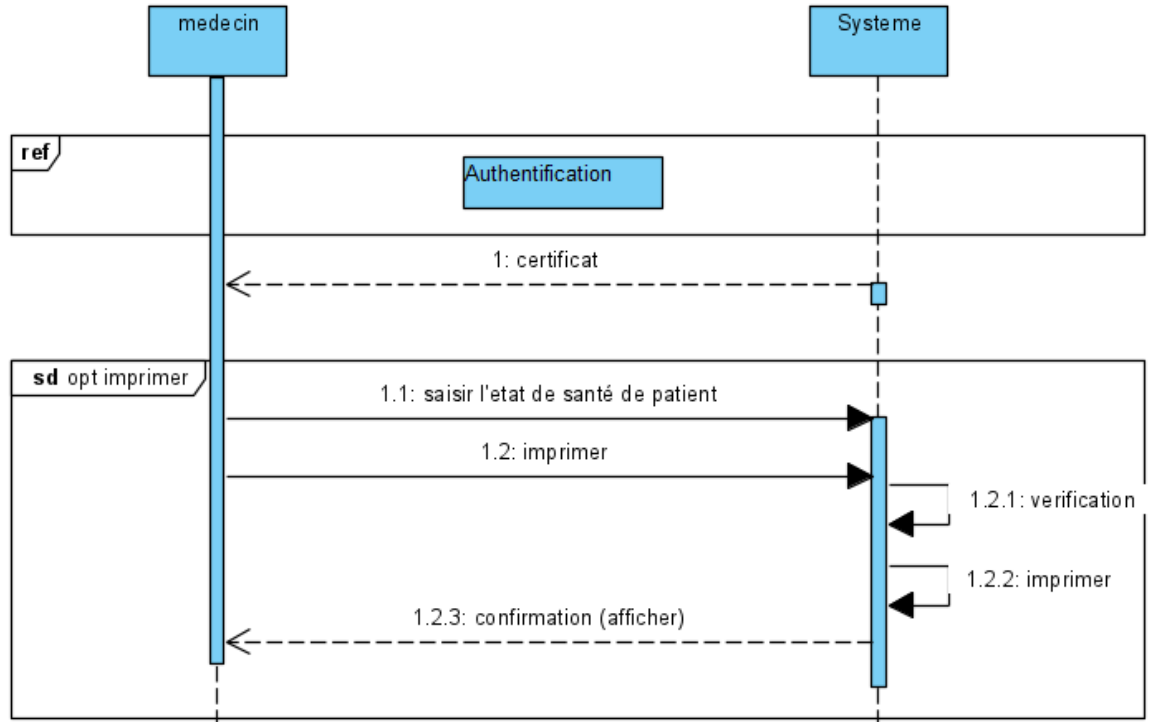


Figure 2.8 : Diagramme de séquence du cas d'utilisation (Gérer les Ordonnances).



**Diagramme de séquence N°5 : cas d'utilisation établir Certificat :**



**Figure 2.9 :** Diagramme de séquence du cas d'utilisation (Établir Certificat).

- Diagramme de séquence N°6 : cas d'utilisation Gérer les patients :

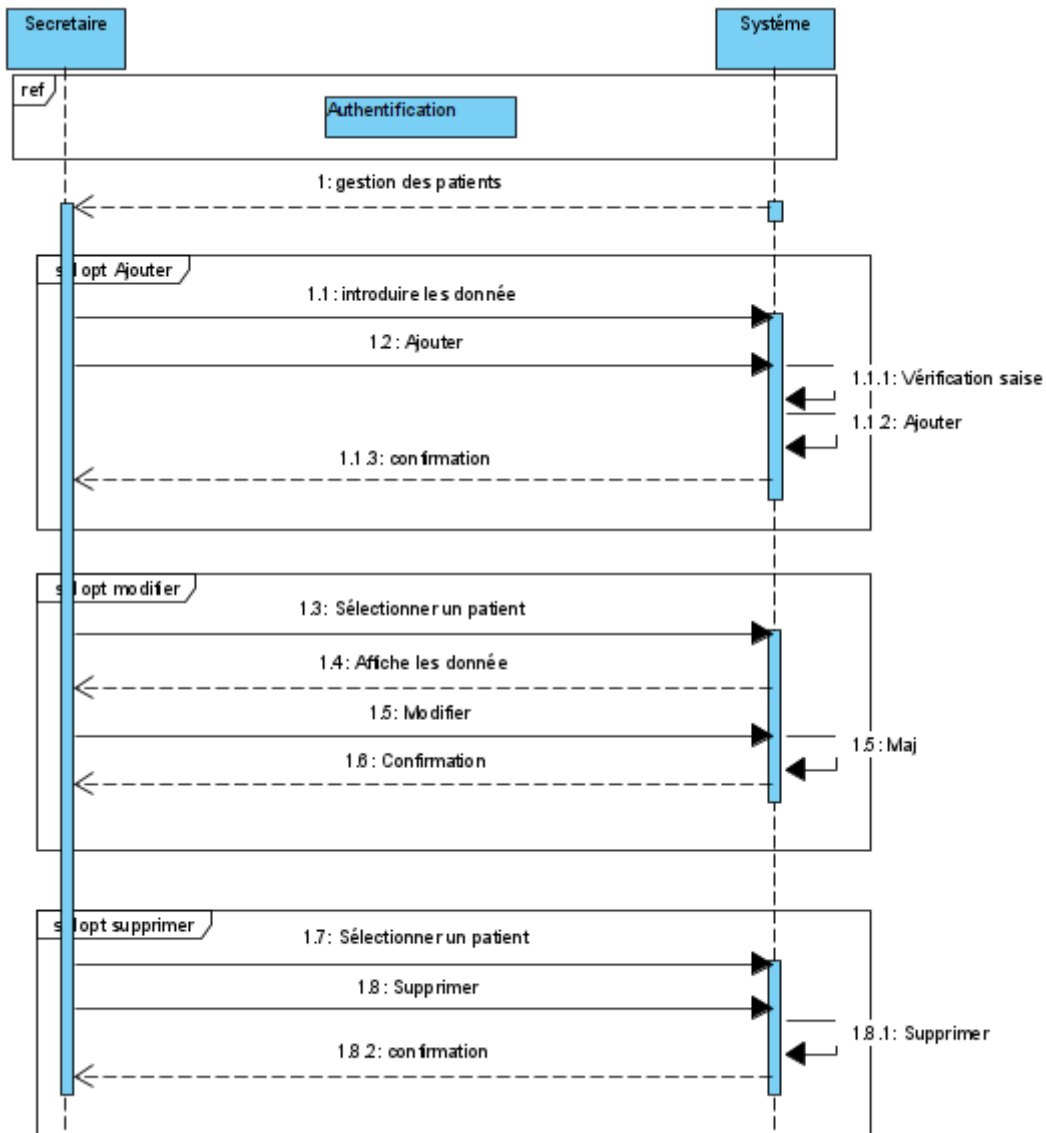


Figure 2.10 : Diagramme de séquence du cas d'utilisation (Gérer les patients).

- Diagramme de séquence N°7 : cas d'utilisation Gérer les rendez-vous :

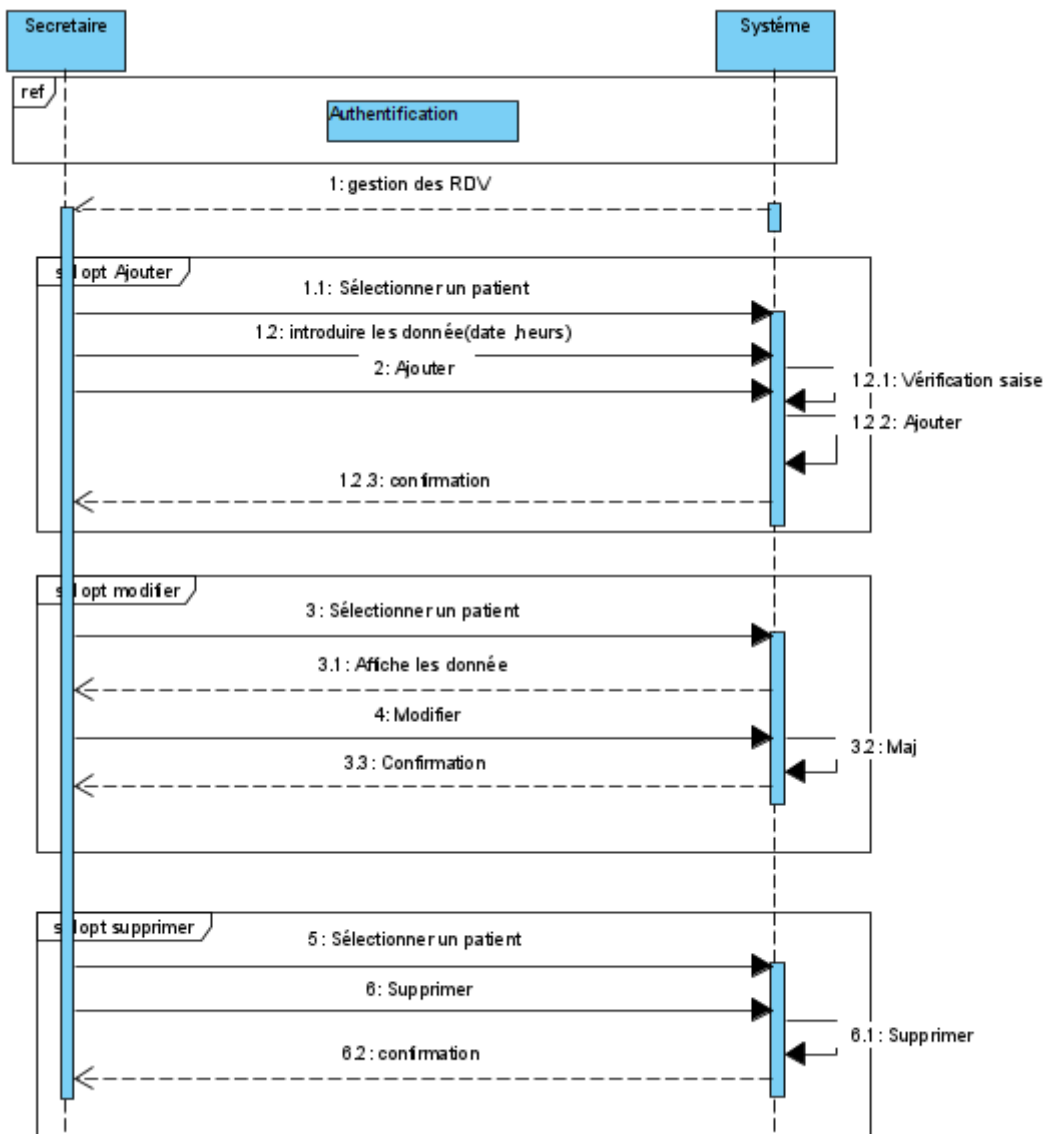


Figure 2.11 : Diagramme de séquence du cas d'utilisation (Gérer les rendez-vous).

- Diagramme de séquence N°8: cas d'utilisation Gérer les utilisateurs :

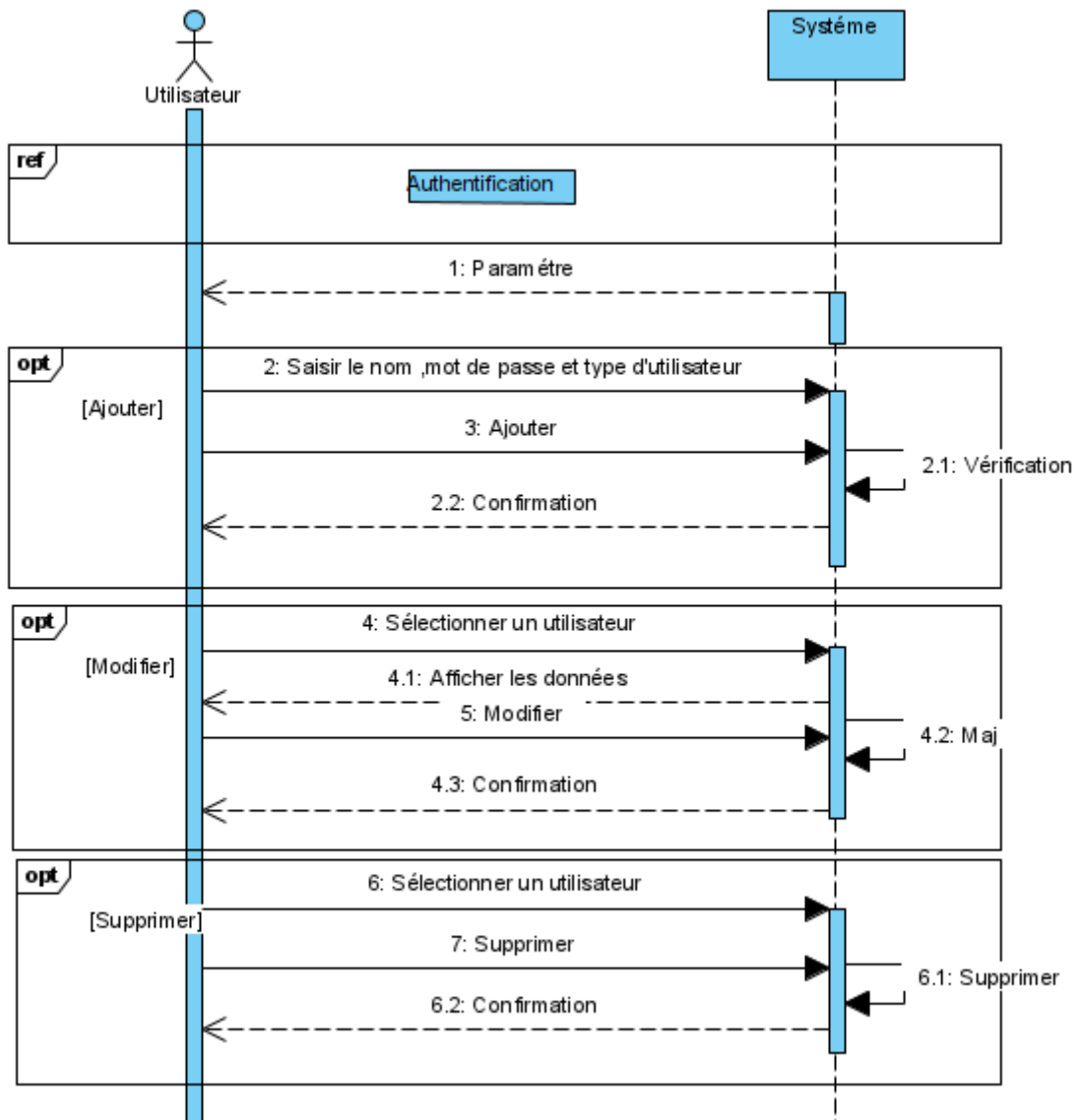


Figure 2.12 : Diagramme de séquence du cas d'utilisation (Gérer les utilisateurs).

## 2.7 Réalisation du diagramme de classe

### 2.7.1 Dictionnaire de données :

| <i>Classe</i> | <i>Attributs</i> | <i>Signification</i>           | <i>Type</i>    | <i>taille</i> |
|---------------|------------------|--------------------------------|----------------|---------------|
| Patient       | id_pat           | Identificateur du patient.     | Entier         |               |
|               | Nom_pat          | Nom du patient.                | Alphanumérique |               |
|               | Prenom_pat       | Prénom du patient.             | Alphanumérique |               |
|               | Date_nai_pat     | Date de naissance du patient.  | Date           |               |
|               | Adresse_pat      | Adresse du patient             | Alphanumérique |               |
|               | Num_tel_pat      | Numéro de téléphone du patient | Alphanumérique |               |
|               | Sexe_pat         | Sexe du patient                | Alphanumérique |               |
| Consultation  | id_cons          | Identificateur de consultation | Entier         |               |
|               | Date_cons        | Date de consultation           | Date           |               |
|               | Diag_cons        | Diagnostique de consultation   | Alphanumérique |               |
| Rendez-vous   | id_rdv           | Identificateur du rendez-vous  | Entier         |               |
|               | Date_rdv         |                                | Date           |               |
|               | Heure_rdv        | Date et heure du rendez-vous   | Date           |               |
| Ordonnance    | Num_ord          | Identificateur d'ordonnance.   | Entier         |               |
|               | Date_ord         | Date d'ordonnance              | Date           |               |
| Médicament    | id_med           | Identificateur de médicament   | Entier         |               |
|               | Nom_med          | Nom de médicament              | Alphanumérique |               |
| Certificat    | Num_cer          | Identité de certificat         | Entier         |               |
|               | Date_cer         | Date de certificat             | Date           |               |
|               | Etat_de_sant     | État de santé                  | Alphanumérique |               |

**Tableau 2.10** : Tableau représentant le dictionnaire de données.

### 2.7.2 Diagramme de classe

Le diagramme de classes est sans doute le diagramme le plus important à représenter pour les méthodes d'analyse orientées objet. En effet, il permet de spécifier qui intervient à l'intérieur du système.

Un diagramme de classes fait abstraction des aspects dynamiques et temporels du système, il permet de représenter une vue statique du système d'information.

Il s'agit plutôt des relations entre les classes, des services rendus et utilisés par chacune d'elles et de l'articulation de l'ensemble [7].

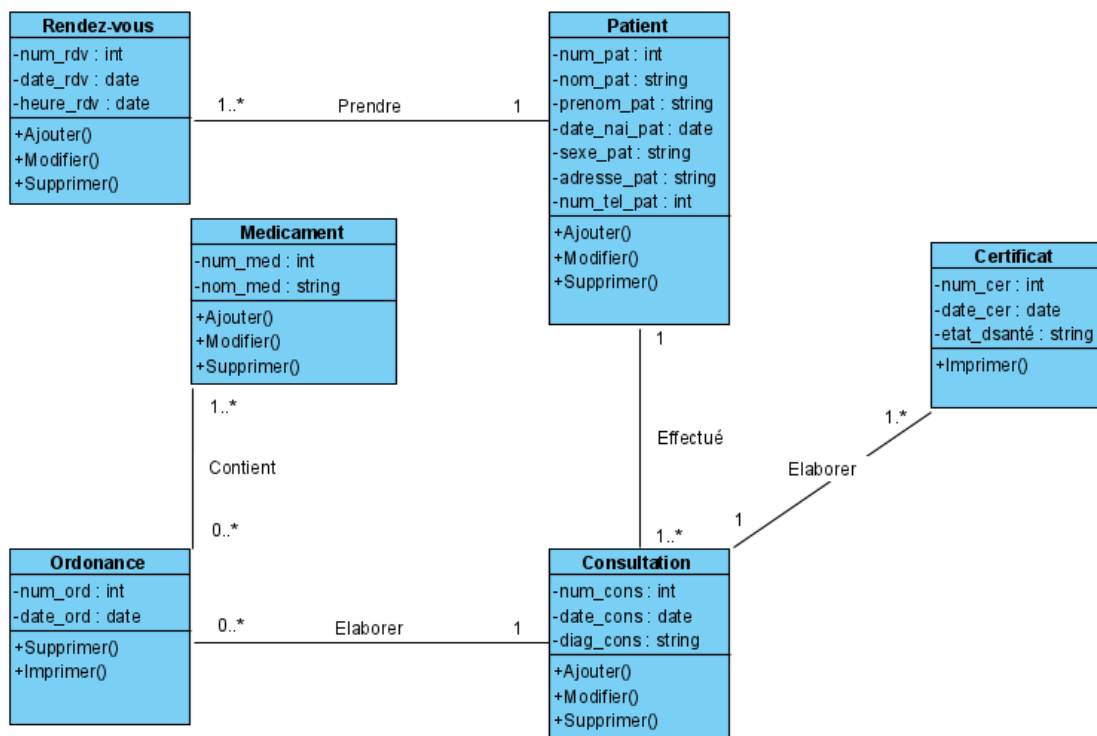


Figure 2.13: Diagramme de classes.

### 2.7.3 Le modèle relationnel :

Le modèle relationnel représente une manière d'organiser mes données de la manière suivante :

- Sous forme de tables à deux dimensions, encore appelées relations, dont les lignes sont appelées n-uplet ou *tuple* en anglais.
- Les données sont manipulées par des opérateurs de l'algèbre relationnelle.
- L'état cohérent de la base est défini par un ensemble de contraintes d'intégrité.

#### 2.7.3.1 Règles de passage au modèle relationnel:

Les règles utilisées pour le passage d'un diagramme de classes à un modèle relationnel sont les suivantes [8] :

- Toute entité devient une relation ayant pour clé primaire son identifiant.
- Chaque propriété se transforme en attribut.
- Toute association non hiérarchique (de type [n, n] ou de dimension > 2) devient une relation. La clé primaire est formée par la concaténation (juxtaposition) de l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.

#### 2.7.3.2 Modèle relationnel:

Après avoir appliqué les règles de passage cités précédemment, nous avons établi le schéma relationnel suivant :

- *Patients* ( id\_pat, Nom\_pat, Prenom\_pat, Date\_nai\_pat, Adresse\_pat, Num\_tel\_pat, Sexe\_pat, #id\_rdv),
- *Consultation* ( id\_cons, Date\_cons, Diag\_cons, #id\_pat),
- *Rendez\_vous* ( id\_rdv, Date\_rdv, Heure\_rdv, #id\_pat),
- *Ordonnance* ( id\_ord, Date\_ord, #id\_cons),
- *Certificate*( id\_cer, Date\_cer, etatde\_sante, #id\_cons),
- *Medicament* ( id\_med, Nom\_med.),

- *Continent (#id\_ord,# id med)*

### **2.8 Conclusion :**

Dans ce chapitre, nous avons présenté les différents diagrammes de modélisation à savoir, les cas d'utilisation, les digrammes de séquence ainsi que le diagramme de classe de notre future application.

La conception de l'application est la partie la plus importante pour bien mettre en œuvre l'application à réaliser.

Le chapitre suivant sera consacré à la réalisation de notre application qui englobe les outils utilisés ainsi que des interfaces.



## **3. Réalisation**

### 3.1. Introduction :

Dans le chapitre précédent, nous avons abordé les différentes étapes permettant la conception d'un système informatique adapté aux exigences et leurs besoins.

Ce dernier chapitre est consacré à la présentation des langages et des outils de développement utilisés pour réaliser notre application. Notre étude est accompagnée de quelques interfaces commentées de l'application implémentée. Ainsi, chaque interface de l'application permet de définir les opérations pouvant être effectuées.

Rappelons que projet consiste à concevoir et implémenter une application desktop dédiée à la gestion d'un cabinet médical.

### 3.2. Application desktop :

Une « application de bureau » (**Desktop application** en anglais) est un logiciel applicatif qui affiche son interface graphique dans un environnement de bureau, il est hébergé et exécuté par l'ordinateur de l'utilisateur.

Une application desktop possède quelques avantages à savoir :

- Les applications de bureau sont confinées à un emplacement physique et ont donc des contraintes d'utilisabilité
- Les applications de bureau doivent être installées séparément sur chaque ordinateur. La mise à jour des applications est également fastidieuse avec les applications de bureau, car elle doit être effectuée sur chaque ordinateur
- Les applications de bureau sont autonomes par nature et ne rencontrent donc aucun obstacle résultant de la connectivité Internet [9].

### 3.3. Outils de développement :

#### 3.3.1. Développement des diagrammes :

Pour réaliser les diagrammes UML ayant servis à modéliser notre application, nous avons utilisé un logiciel de développement de diagramme Visual Paradigm.

**Visual Paradigm** est un logiciel de conception de diagrammes dans le cadre d'une programmation. Tout en un, il possède plusieurs options permettant une large possibilité de modélisation en ULM.

- Modélisation : le logiciel Visual Paradigm offre de nombreux outils pour créer différents types de schémas comme les diagrammes d'exigences et de cas d'utilisation. Il possède bon nombre de navigateurs permettant de personnaliser chaque élément.
- Analyse et manipulation de codes sources : Visual Paradigm permet de générer des codes sources en divers langages comme le Java ou C++ à partir du modèle créé. Inversement, il permet de produire un modèle à partir de codes sources.
- Capture avec la souris : Visual Paradigm permet d'utiliser la souris pour manipuler facilement le diagramme à créer. Tous les types d'actions sont pris en charge dont le traçage et la commande ainsi que la connexion des éléments [10].

### 3.3.2. NetBeans :

NetBeans est un environnement de développement intégré (EDI), placé en Open Source par Sun en juin 2000. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C+, JavaScript, PHP, HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris, Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java) [11].

### 3.3.3. PhpMyAdmin :

PhpMyAdmin est une interface web en PHP pour administrer à distance les SGBD MySQL et MariaDB.

Il permet d'administrer les éléments suivants :

- Les bases de données
- Les tables et leurs champs (ajout, suppression, définition du type)
- Les index, les clés primaires et étrangères.
- Les utilisateurs de la base et leurs permissions [12].

Les requêtes SQL peuvent être effectuées via l'interface utilisateur, tandis que vous avez toujours la possibilité d'exécuter directement toute instruction SQL.

#### **3.3.4. MYSQL :**

MYSQL est un système de gestion de base de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

Le principe d'une base de données relationnelle est d'enregistrer les informations dans des tables qui représentent des regroupements de données par sujets (table des produits, table d'utilisateur par exemple). Les tables sont reliées entre elles par des relations [13].

### **3.4. Langage de programmation :**

#### **3.4.1. JAVA:**

Java est un langage de programmation orienté objet et un environnement d'exécution, développé par Sun Microsystems. Il fut présenté officiellement en 1995. Java était à la base un langage pour Internet, pour pouvoir rendre plus dynamiques les pages (tout comme le JavaScript aujourd'hui). Mais Java a beaucoup évolué et est devenu un langage de programmation très puissant. Java est aujourd'hui officiellement supportée par Sun, mais certaines entreprises comme IBM font beaucoup pour Java.

### **3.5. Architecture global de l'application :**

L'application est accessible via une interface conviviale, l'utilisateur doit d'abord introduire son pseudo et son mot de passe. Après, il pourra accéder à toutes les fonctionnalités directement comme la gestion des patients, la gestion des rendez-vous, ...etc.

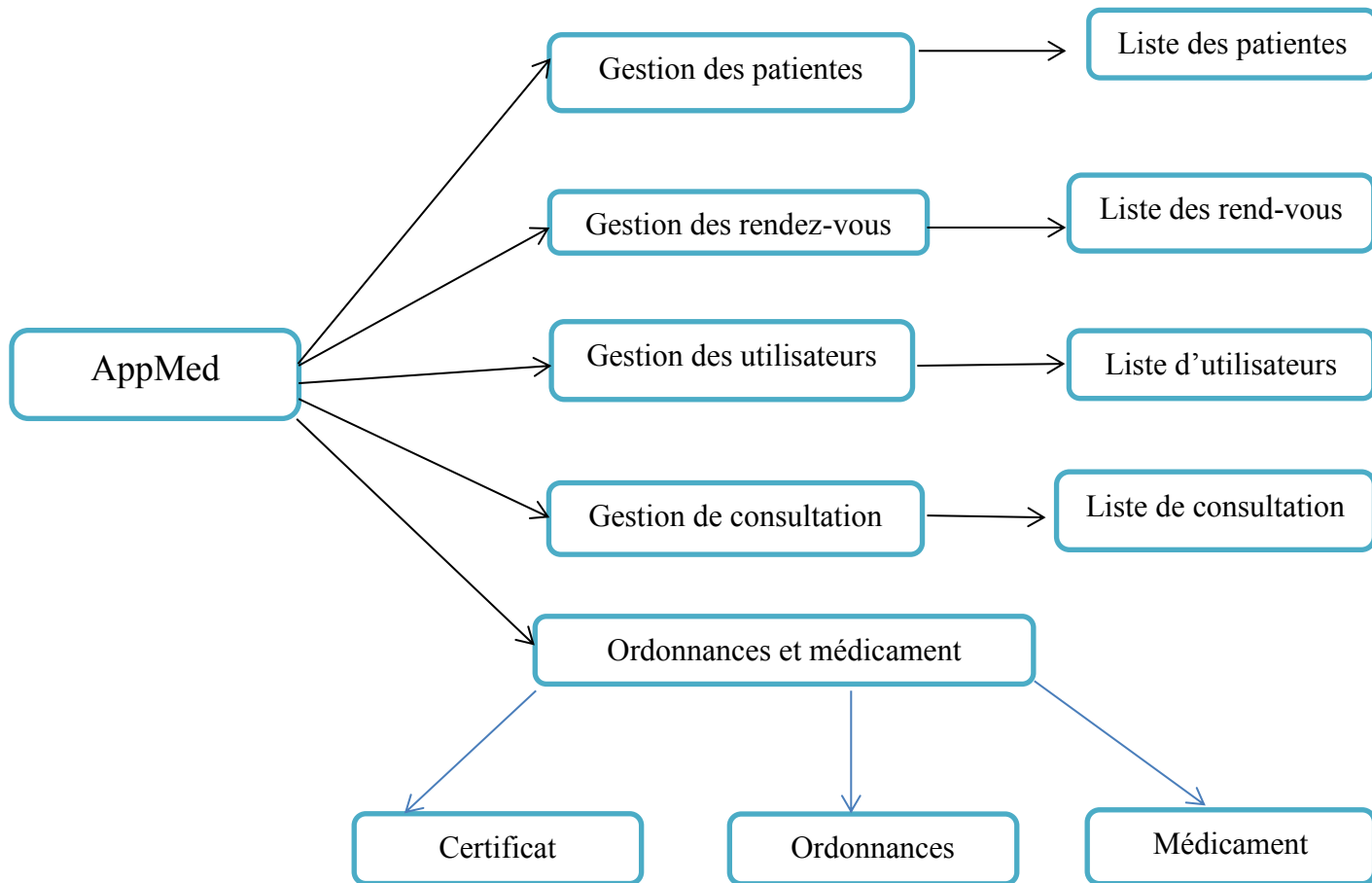


Figure 3.1: Représentation de l'architecture globale de l'application.

### 3.6. Interfaces de l'application :

#### 3.6.1 Interface authentification :

Les utilisateurs de l'application doivent impérativement saisir le nom, le mot de passe et le type d'utilisateur pour accéder.

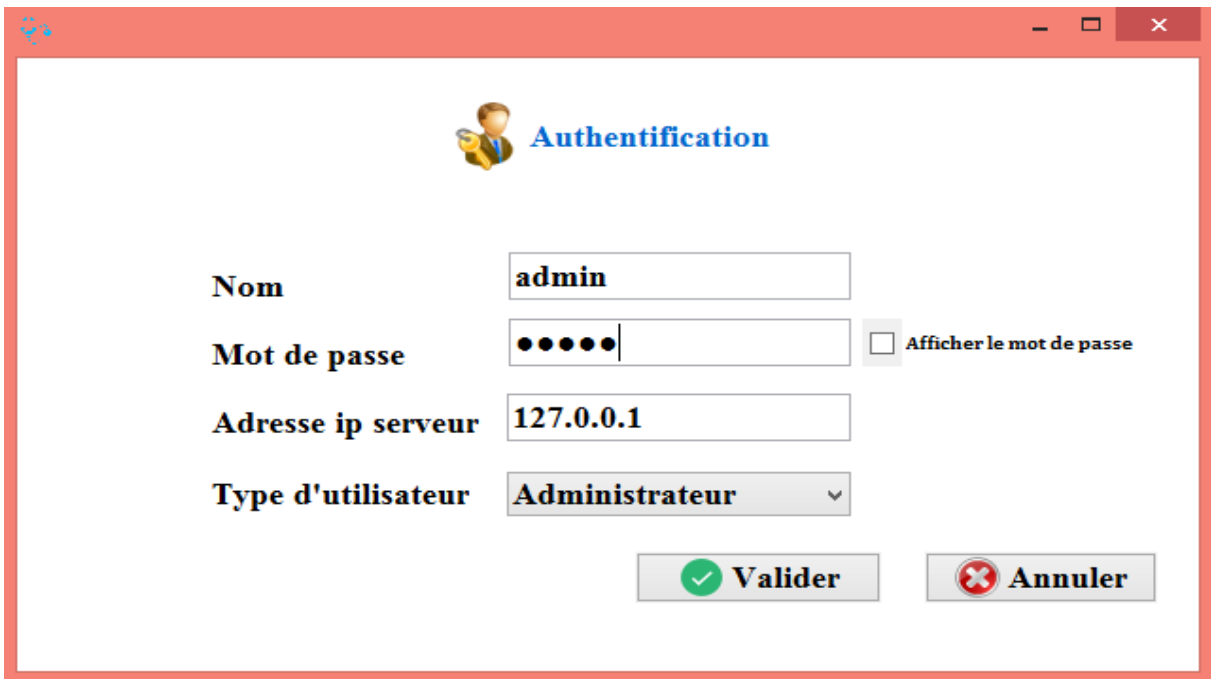


Figure 3.2 : Interface d'authentification.

### 3.6.2 Interface gestion des patients :

Cette figure représente l'interface relative au patient avec les possibilités d'inscription d'un nouveau patient, de modification, et de suppression.

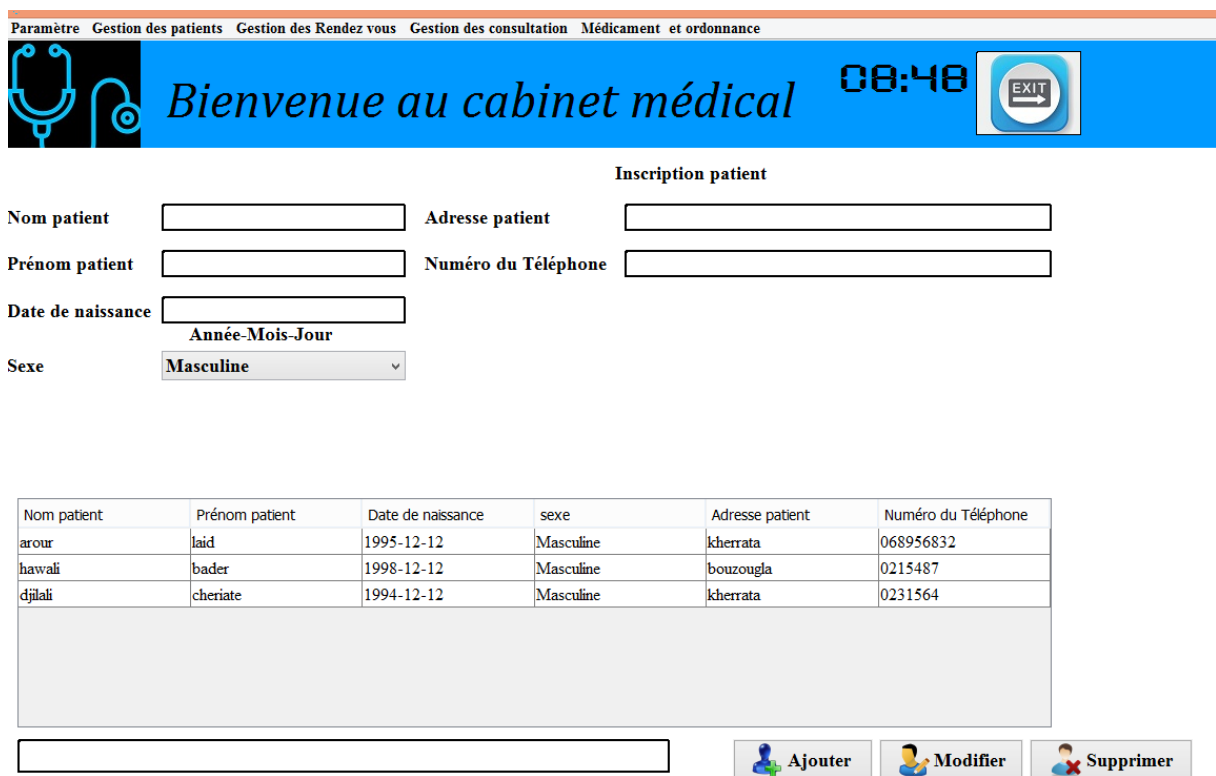


Figure 3.3 : Interface Gestion des patients.

### 3.6.3 Interface consultation :

Cette figure représente l'interface de la consultation des patients par le médecin.

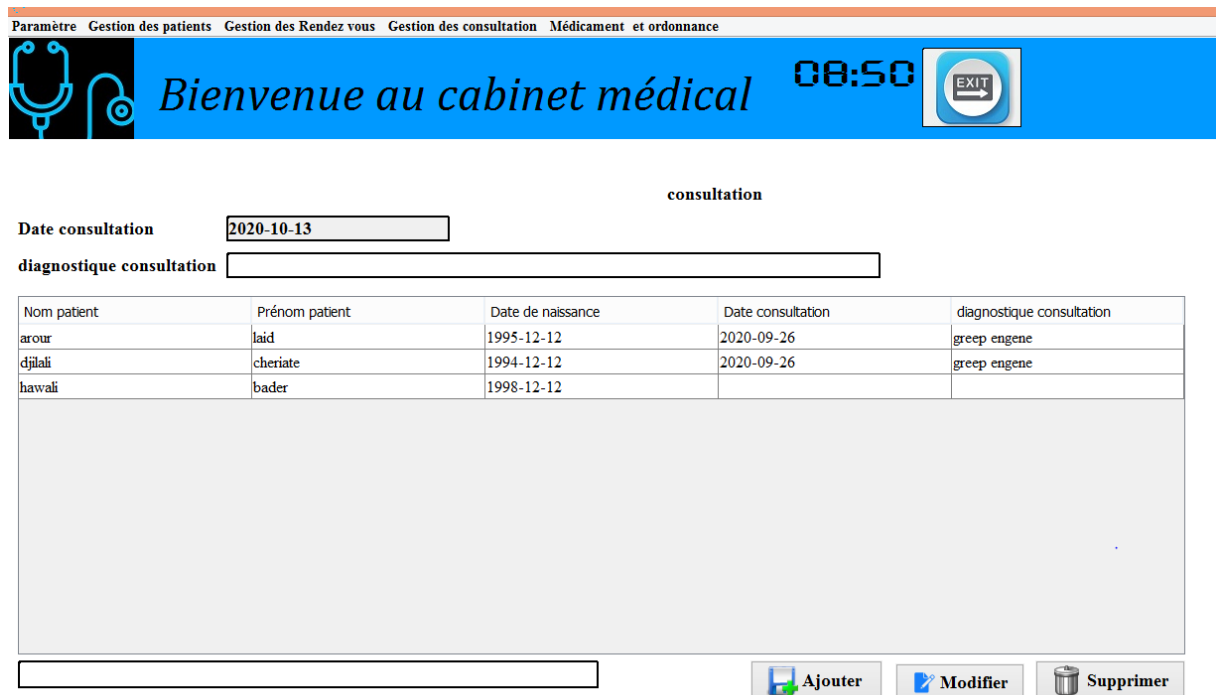


Figure 3.4 : Interface Gestion consultation.

### 3.6.4 Interface rendez-vous :

Cette capture d'écran représente un scénario de prise de rendez-vous.

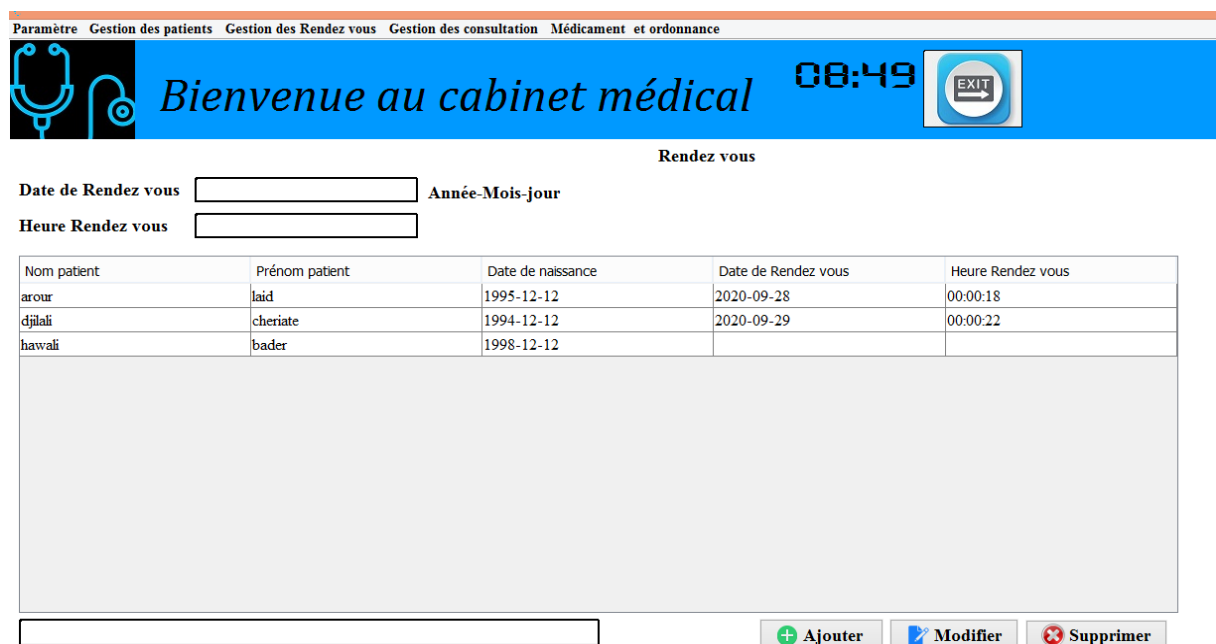


Figure 3.5 : Interface Gestion de rendez-vous.

### 3.6.5 Interface médicament :

Cette interface présente la procédure de gestion des médicaments.

Paramètre Gestion des patients Gestion des Rendez vous Gestion des consultation Médicament et ordonnance

Bienvenue au cabinet médical 08:51 EXIT

Médicament

Nom médicament

| ID médicament | Nom médicament |
|---------------|----------------|
| 5             | dolpran        |
| 6             | espigec        |
| 7             | romafed        |

Figure 3.6 : Interface Médicament.

### 3.6.6 Interface ordonnance :

Cette interface affiche les modalités d'établissement d'une ordonnance au patient.

Paramètre Gestion des patients Gestion des Rendez vous Gestion des consultation Médicament et ordonnance

Bienvenue au cabinet médical 08:53 EXIT

Ordonnance

Médicament  Patient

| Nom Médicament | Quantité | Consommation |
|----------------|----------|--------------|
| dolpran        |          |              |
| espigec        |          |              |
| romafed        |          |              |

Figure 3.7 : Interface Gestion d'ordonnance.





la santé bejaia cartie sgheer bejaia

Nom patient: arour laid date naissance: 1993-06-02

### Ordonnoce

| Nom Médicament | Quantité | Consommation        |
|----------------|----------|---------------------|
| doléprene      | 1        | Deux fois par jour  |
| romafed        | 2        | Trois fois par jour |
| espigic        | 2        | Trois fois par jour |

### Le médecin

date: 2020-09-09

Figure 3.8 : Interface ordonnance.

### 3.6.7 Interface certificat :

Cette interface présente comment établir un certificat médical à un patient.

Paramètre Gestion des patients Gestion des Rendez vous Gestion des consultation Médicament et ordonnance

Bienvenue au cabinet médical 08:55 EXIT

Patient arour-laid Etat de santé

Imprimer

Figure 3.9 : Interface Gestion de certificat.



la santé bejaia quartie sgheer bejaia

**Certificat médical**

**Je soussigner DR**  
**Certifier Avoir examiné arour laid**  
**Né(e) le 1993-12-12**  
**Suit a cet examen sent état de santé**  
**est une tres bon santé apte de faire le sport**

**Le médecin**

**Fait le: 2020-09-21**

Figure 3.10 : Interface certificat médicale.

**3.6.8 Interface gestion d'utilisateur :** La figure suivante présente l'interface relative à la gestion des utilisateurs.

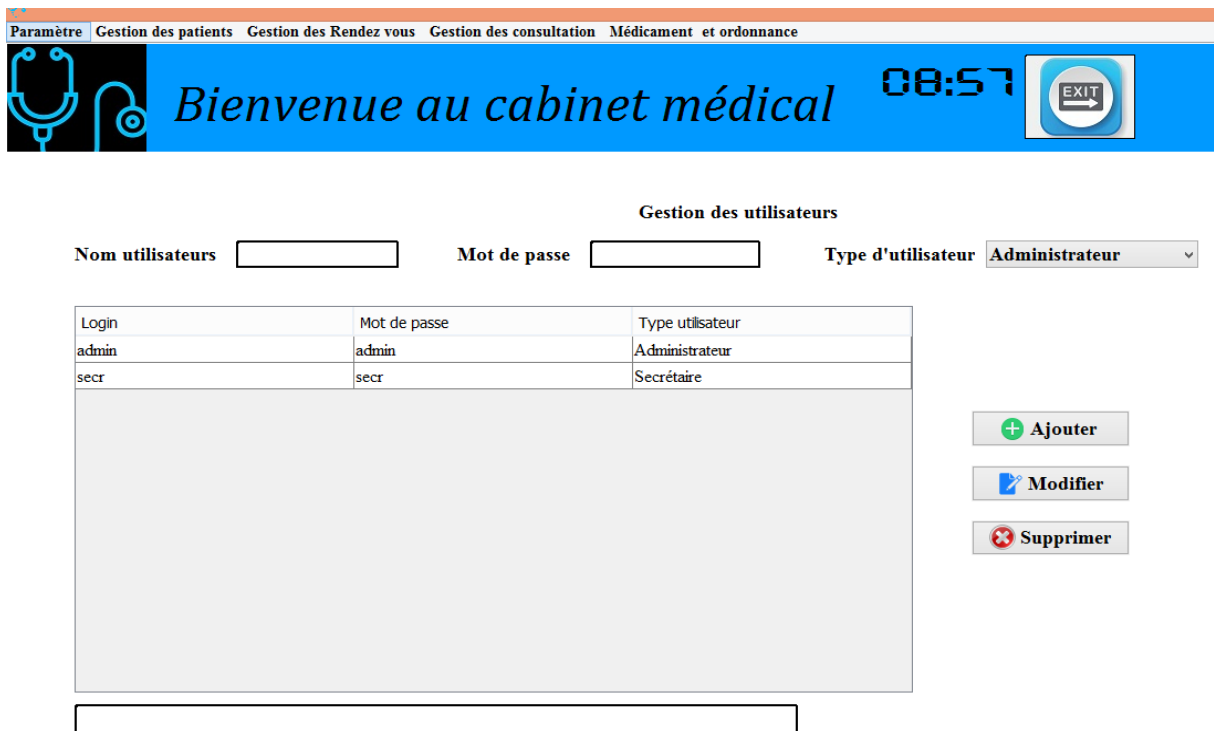
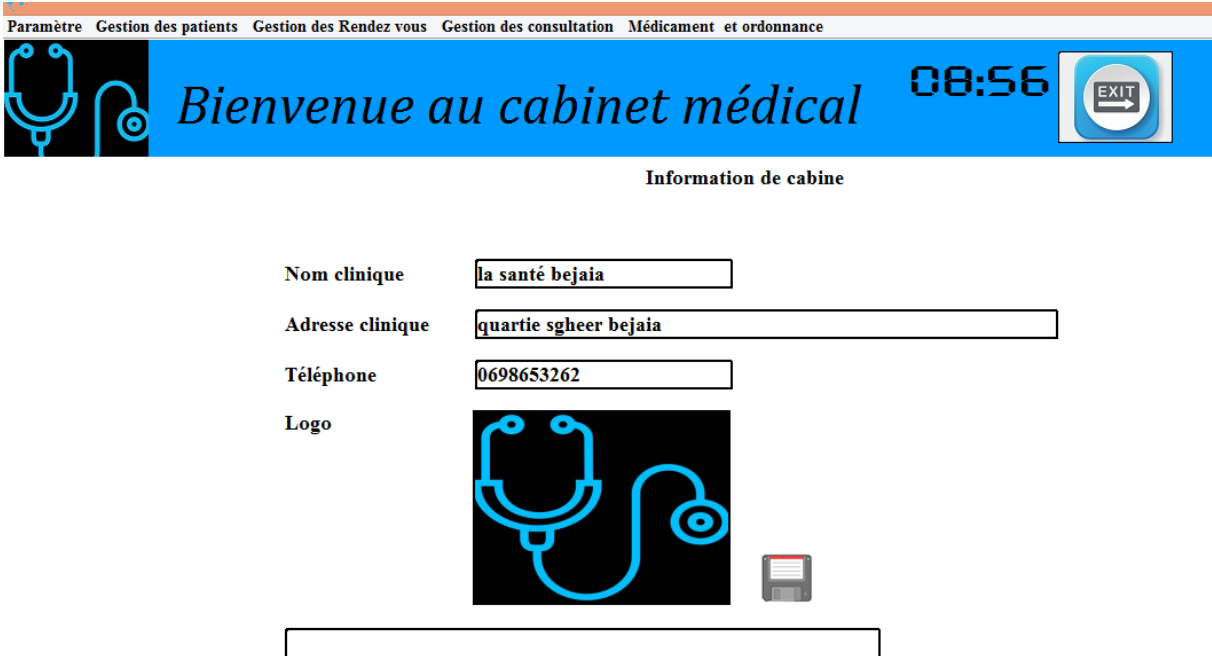


Figure 3.11 : Interface Gestion des utilisateurs.

### 3.6.9 Interface gérer les informations d'un cabinet médical :

Cette interface présente les informations relatives à un cabinet médical.



The screenshot shows a web application interface for managing medical cabinet information. At the top, there is a navigation menu with the following items: Paramètre, Gestion des patients, Gestion des Rendez vous, Gestion des consultation, and Médicament et ordonnance. Below the menu is a blue header bar with a stethoscope icon on the left, the text "Bienvenue au cabinet médical" in the center, the time "08:56" on the right, and an "EXIT" button. Below the header bar, the title "Information de cabine" is centered. The main content area contains a form with the following fields:

|                  |  |
|------------------|--|
| Nom clinique     | <input type="text" value="la santé bejaia"/>   |
| Adresse clinique | <input type="text" value="quartie sgheer bejaia"/>                                     |
| Téléphone        | <input type="text" value="0698653262"/>  |
| Logo             | <input type="text" value="Stethoscope icon"/> <input type="button" value="Save icon"/> |

Below the logo field, there is an empty text input field.

**Figure 3.12 :** Interface information d'un cabinet médical.

### 3.7. Conclusion :

Dans ce chapitre nous avons présenté les aspects pratiques liés à la réalisation de notre application, à savoir les outils de développement nécessaires pour le fonctionnement de notre application, en passant par la représentation graphique de l'architecture globale de l'application, et quelques interfaces illustrer le travail qui a été fait.

# **Conclusion générale**

## Conclusion générale

Au cours de ce travail, nous avons présenté les différentes étapes ayant conduit à la mise en œuvre d'une application de bureau dédiée à la gestion d'un cabinet médical qui renferment la gestion des patients, des rendez-vous, des consultations, des ordonnances ainsi que les médicaments.

Nous avons commencé par recenser les difficultés qui rencontrent un cabinet médical afin d'apporter une solution adéquate.

Le langage de modélisation UML et le processus unifié UP ont constitué le support de l'analyse des besoins et la conception de notre application via les différents diagrammes UML couvrant les aspects fonctionnels, dynamiques et statiques de tout le développement.

Pour réaliser notre application, nous avons utilisé le 'PhpMyAdmin MYSQL' pour implémenter une base de données et le langage java qui est basé sur la programmation orientée objets.

Ce projet a fait l'objet d'une expérience intéressante, très bénéfique pour nous. En effet, il nous a permis d'enrichir nos connaissances théoriques et compétences dans le domaine de la conception et de la programmation. Ajoutant à ceci, la mise en application des connaissances acquises tout au long de nos études.

En plus, c'était une bonne occasion pour réaliser un travail concret avec des objectifs clairs et bien définis.

Cependant des perspectives restent envisageables, telles que l'enrichissement de l'application par la fonctionnalité permettant d'établir des statistiques ainsi que l'amélioration de la qualité des renseignements via une application Android et iOS qui permettra de compléter notre application de bureau.

# **Bibliographie**

# Bibliographie

- [1] <http://www.memoireonline.com/02/13/7039/Application-Gestion-de-cabinet-medicale-en-Java.html> (14 septembre 2020).
- [2] G.PICARD SMA/G2/ENS Mines, définition d'UP Saint-Etienne, 2009.
- [3] A. V. ANDRE PASCAL, Développement de logiciels avec UML 2 et OCL. ellipses, 2013.
- [4] R. P. FRANCK VALLEE, UML 2 en action, de l'analyse des besoins à la conception. 4<sup>ème</sup> édition, EYROLLES, 2007.
- [5] P. ROQUES, UML2 par la pratique, Eyrolles, 2006.
- [6] Référence RAD Studio UML 2.0 2015
- [7] SYLVAIN CHERRIER Diagramme de classes UML 2015.
- [8] <http://stephanie.laporte.pagesperso-orange.fr/Pdf/passageMCDMR.pdf> Consulter (26 aout2020).
- [9] <https://www.avestagroup.net/DetailsEN.aspx.consulter> (28 aout 2020).
- [10] <https://air.imag.fr/index.php/StarUML>. consulter (29 aout 2020).
- [11] <https://www.techno-science.net/definition/5346.html.consulter> (29 aout 2020).
- [12] <https://doc.ubuntu-fr.org/phpmyadmin>. (06 septembre 2020).
- [13] <http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimedaiinformatique/m-glossaire-informatique-et-multimedia/448-mysql-definition.html>. (05 septembre2020).

## **Résumé**

*Le but de notre travail est la conception et réalisation d'une application de bureau dédiée à la gestion d'un cabinet médical. Cette application permet une gestion efficace des patients, des rendez-vous, des consultations et des ordonnances.*

*Ce travail a été réalisé en utilisant le processus unifié et le langage de modélisation UML. Nous avons implémenté notre application avec le langage java et le MySQL PHPMyAdmin comme Système de Gestion des bases de données.*

**Mots clés :** *Application de bureau, JAVA, MYSQL PHPMyAdmin, SGBD, UP, UML.*

## **Abstract**

*The goal of our work is the design and implementation of an office application dedicated to the management of a medical office. This app enables effective management of patients, appointments, consultations and prescriptions.*

*This work was done using the unified process and the UML modeling language*

*We implemented our application with java language and MySQL PHPMyAdmin as Database Management System.*

**Keywords:** *Desktop application, JAVA, MYSQL PHPMyAdmin, DBMS, UP, UML*