



# Rapport de projet

4eme année  
Ingénierie Informatique et Réseaux

---

Étude et développement d'un système de gestion  
de clinique

---

## Réaliser par :

AIT EL QADI Anas  
IZMAR Mohamed Taha

## Encadré par :

MR.Nabil AQILI  
MME.Kawtar IMGHARENE



# Dédicace

À tous ceux qui ont contribué à la concrétisation de ce projet, nous avons ensemble traversé des épreuves et trouvé des inspirations, partagé des rires et des moments de réflexion.

Ce projet est bien plus qu'une simple réalisation ; il est le reflet de notre dévouement, de notre créativité et de notre esprit de collaboration. Chaque membre de notre groupe a apporté une touche unique, illuminant ce projet de manière spéciale et précieuse.

Que cette dédicace témoigne de notre travail acharné, de notre solidarité et de notre passion commune. Puisseons-nous nous souvenir de ce projet comme d'une preuve indélébile de ce que nous sommes capables d'accomplir ensemble.

# Remerciements

Nous souhaitons exprimer notre profonde gratitude à tous ceux qui ont contribué à la réalisation de ce projet de fin d'études. Tout d'abord, nous remercions sincèrement Monsieur le Directeur de l'EMSI pour la formation de qualité qu'il nous a offerte. Grâce à son leadership et à sa vision, nous avons bénéficié d'un environnement académique exceptionnel, propice à notre épanouissement et à notre réussite. Nos sincères remerciements vont également à nos professeurs, dont l'expertise, la patience et l'engagement ont été essentiels à notre apprentissage. Leurs enseignements et leurs conseils nous ont guidés tout au long de notre parcours académique, nous permettant de développer les compétences nécessaires pour relever les défis de demain. À nos amis, merci pour votre soutien constant, vos encouragements et pour les moments de camaraderie qui ont rendu cette expérience encore plus mémorable. Votre présence à nos côtés a été une source de motivation et de réconfort. Enfin, nous souhaitons exprimer notre infinie reconnaissance à nos parents. Votre amour, votre soutien inconditionnel et vos sacrifices ont été le pilier sur lequel nous avons construit nos rêves. Sans vous, rien de tout cela n'aurait été possible. Ce projet est le fruit de la collaboration, du travail acharné et du soutien de chacun d'entre vous. Merci de nous avoir permis de réaliser cet accomplissement.

# Résumé

Ce rapport discute du processus de développement ainsi que des principales fonctionnalités associées à une application de gestion de clinique développée en Java Swing. L'objectif fondamental de l'application est de fournir une interface graphique permettant de gérer efficacement les aspects administratifs et médicaux d'une clinique, notamment la gestion des patients et des rendez-vous.

L'implémentation de cette application s'est déroulée en plusieurs étapes : la planification, la construction et le test .

Conception d'une interface utilisateur intuitive pour guider les interactions avec l'application. L'interface comprend des modules pour l'ajout et la gestion des patients, la prise de rendez-vous, ainsi que la consultation et mise à jour des dossiers médicaux. Pour réaliser l'application, Java Swing a été utilisé pour construire l'interface graphique, permettant une expérience utilisateur fluide et réactive. Des fonctionnalités comme la gestion des fenêtres, les boutons, les champs de texte et les tableaux ont été intégrées pour offrir une interaction simple et directe. Des tests ont été effectués à chaque étape du cycle de vie du développement pour assurer la qualité et la fiabilité de l'application. Ces tests incluent des tests unitaires pour vérifier le bon fonctionnement des différentes fonctions, ainsi que des tests d'intégration pour valider la communication entre les différentes parties du système.

En fin de compte, cette application de gestion de clinique propose une solution complète, facile à utiliser, et adaptée aux besoins des professionnels de santé pour une gestion optimale de leurs activités cliniques.

# Abstract

This report discusses the development process as well as key features associated with a clinic management application developed in Java Swing. The basic objective of the application is to provide a graphical interface to efficiently manage the administrative and medical aspects of a clinic, including patient and appointment management.

The implementation of this application took place in several stages : planning, building and testing.

Designed an intuitive user interface to guide interactions with the application. The interface includes modules for adding and managing patients, making appointments, as well as consulting and updating medical records. To create the application, Java Swing was used to build the GUI, allowing for a smooth and responsive user experience. Features like window management, buttons, text fields and tables have been integrated to provide simple and straightforward interaction. Testing was carried out at each stage of the development lifecycle to ensure the quality and reliability of the application. These tests include unit tests to verify the correct functioning of the different functions, as well as integration tests to validate the communication between the different parts of the system.

Ultimately, this clinic management application offers a complete, easy-to-use solution adapted to the needs of healthcare professionals for optimal management of their clinical activities.

# Table des matières

<b>Dédicace</b>	<b>2</b>
<b>Remerciements</b>	<b>3</b>
<b>Résumé</b>	<b>4</b>
<b>Abstract</b>	<b>5</b>
<b>Liste des sigles et acronymes</b>	<b>10</b>
<b>Introduction générale</b>	<b>11</b>
<b>1 Présentation et analyse du projet</b>	<b>12</b>
1.1 Introduction . . . . .	13
1.2 Périmètre du projet . . . . .	13
1.2.1 But . . . . .	13
1.2.2 Missions . . . . .	13
1.2.3 Contraintes . . . . .	13
1.2.4 Les Sprints . . . . .	14
1.3 Étude de l'existant . . . . .	14
1.3.1 Concept métier . . . . .	14
1.4 Étude fonctionnelle . . . . .	15
1.4.1 Objectifs fonctionnels . . . . .	15
1.5 Besoins fonctionnels . . . . .	16
1.5.1 Bloc fonctionnel : réceptionniste . . . . .	16
1.5.2 Bloc fonctionnel : administrateur . . . . .	16
1.5.3 Bloc fonctionnel : infirmière/infirmier . . . . .	17
1.5.4 Bloc fonctionnel : médecin . . . . .	19
1.6 Conclusion . . . . .	19
<b>2 Conception et étude pratique</b>	<b>21</b>
2.1 Introduction . . . . .	22
2.2 Analyse des besoins . . . . .	22
2.2.1 Les besoins fonctionnels . . . . .	22
2.2.2 Les besoins non fonctionnels . . . . .	22
2.3 Langage de conception et conception général . . . . .	23
2.3.1 Langage UML . . . . .	23
2.3.2 Les différents types de diagrammes UML . . . . .	24
2.4 Conception général . . . . .	24

2.4.1	Diagrammes de cas d'utilisation . . . . .	24
2.4.2	Diagrammes de classe . . . . .	26
2.4.3	Diagramme de séquence cas d'ajout d'un patient . . . . .	27
2.4.4	Diagramme de séquence cas de modification des données d'un patient . . . . .	28
2.5	Description Textuelle . . . . .	29
2.5.1	cas : traiter les patient . . . . .	29
2.5.2	cas : consultation de la liste des patient et des rendez-vous . . . . .	30
2.6	Conclusion . . . . .	31
<b>3</b>	<b>Mise en place l'application</b>	<b>32</b>
3.1	Introduction . . . . .	33
3.2	Outils . . . . .	33
3.2.1	JAVA SWING . . . . .	33
3.2.2	composants de Java Swing . . . . .	33
3.2.3	NetBeans IDE . . . . .	35
3.2.4	MAMP . . . . .	35
3.2.5	PHPmyadmin . . . . .	36
3.2.6	Enterprise Architec . . . . .	36
3.3	Premier sprint . . . . .	38
3.3.1	Page de connexion : . . . . .	38
3.3.2	Gestion des réceptionnistes : . . . . .	39
3.3.3	Gestion des administrateurs : . . . . .	41
3.4	Deuxième sprint . . . . .	42
3.4.1	Gestion des infirmières/infirmiers : . . . . .	42
3.4.2	Gestion des médecins : . . . . .	45
3.5	Conclusion . . . . .	46
	<b>Conclusion générale</b>	<b>47</b>
	<b>Perspectives</b>	<b>48</b>
	<b>Webographie</b>	<b>49</b>



# Table des figures

1.1	les sprints . . . . .	14
1.2	Règles de gestion du réceptionniste . . . . .	16
1.3	Règles de gestion de l'administrateur . . . . .	17
1.4	Règles de gestion de infirmière/infirmier . . . . .	18
1.5	Règles de gestion du médecin . . . . .	19
2.1	diagramme de cas d'utilisation globale . . . . .	25
2.2	diagramme de classe . . . . .	26
2.3	diagramme de séquence cas d'ajout d'un patient . . . . .	27
2.4	diagramme de séquence cas de modification d'un patient 1 . . . . .	28
2.5	diagramme de séquence cas de modification d'un patient 2 . . . . .	29
3.1	JAVA SWING logo . . . . .	33
3.2	composants de Java Swing . . . . .	34
3.3	netbeans logo . . . . .	35
3.4	js logo . . . . .	35
3.5	phpmyadmin . . . . .	36
3.6	entreprise architect logo . . . . .	37
3.7	page de connexion . . . . .	38
3.8	Diagramme de cas d'utilisation des réceptionnistes . . . . .	39
3.9	page du réceptionniste . . . . .	40
3.10	Diagramme de cas d'utilisation d'administrateurs . . . . .	41
3.11	page d'administrateur . . . . .	42
3.12	Diagramme de cas d'utilisation des infirmières/infirmiers . . . . .	43
3.13	page des infirmières/infirmiers . . . . .	44
3.14	Diagramme de cas d'utilisation des médecins . . . . .	45
3.15	page des médecins . . . . .	46

# Liste des tableaux

1	Tableau des sigles et acronymes . . . . .	10
---	---	----

# Liste des sigles et acronymes

Sigle	Description
JAVA S	JAVA SWING
IDE	integrated development environment
UML	Unified Modeling Language

TABLE 1 – Tableau des sigles et acronymes

# Introduction générale

Avec l'évolution rapide des besoins du secteur de la santé et l'importance croissante de l'efficacité dans la gestion des établissements médicaux, la gestion des informations cliniques est devenue un enjeu crucial. La capacité à organiser et à suivre les rendez-vous, les dossiers médicaux et les informations des patients est essentielle pour garantir une prise en charge de qualité et optimiser le fonctionnement de la clinique.

Notre projet, réalisé lors de notre stage en tant qu'équipe, se concentre sur le développement d'une application de gestion de clinique. Cette solution vise à fournir aux professionnels de santé un outil efficace et convivial pour gérer les différents aspects administratifs et médicaux d'une clinique, allant de la gestion des patients à la planification des rendez-vous et au suivi des dossiers médicaux.

L'objectif de ce rapport est d'offrir une analyse approfondie du cycle de développement de cette application, en mettant en évidence les choix technologiques, les processus de conception et les fonctionnalités clés qui ont contribué à créer une solution pratique et performante. Nous avons pris soin d'assurer une gestion optimisée des données des patients tout en garantissant la sécurité et la confidentialité de l'information.

Au cours de ce projet, nous avons mis en pratique nos compétences en développement d'applications de bureau, en conception d'interfaces utilisateur et en gestion de projet pour surmonter les défis liés à la création d'une solution complète et fiable. Ce rapport fournit un aperçu détaillé de notre approche de conception, de l'analyse des besoins initiaux à la mise en œuvre des fonctionnalités principales, tout en soulignant les enseignements tirés et les réussites obtenues tout au long du développement.

# Chapitre 1

## Présentation et analyse du projet

## 1.1 Introduction

La gestion efficace d'une clinique repose sur des processus bien structurés permettant de coordonner les différentes interactions entre les patients, les médecins et le personnel administratif. Dans ce contexte, les outils numériques jouent un rôle crucial pour optimiser la gestion des rendez-vous, des dossiers médicaux et des ressources internes.

Ce projet s'inscrit dans cette démarche d'optimisation en proposant le développement d'une application de gestion de clinique, conçue pour répondre aux besoins spécifiques des établissements de santé. À travers une interface intuitive et des fonctionnalités adaptées, l'application vise à simplifier les tâches administratives, améliorer la prise en charge des patients et garantir un suivi rigoureux des informations médicales.

Dans ce chapitre, nous présenterons les grandes lignes du projet, en décrivant le contexte qui a motivé sa réalisation, les objectifs visés, ainsi que les fonctionnalités principales de l'application. Cette présentation permettra de mieux comprendre les choix techniques et fonctionnels adoptés tout au long du développement de la solution.

## 1.2 Périmètre du projet

### 1.2.1 But

Étude, la conception et la réalisation d'un système web de gestion de clinique, couvrant les modules suivants :

- Gestion des chambres
- Gestion des patients
- Gestion des consultations
- Gestion financière
- Gestion des personelles

### 1.2.2 Missions

- Étude technique
- Étude fonctionnelle
- Conception
- Réalisation
- Déploiement
- Documentation

### 1.2.3 Contraintes

- Délais (Fin Janvier 2025)
- Budget
- Technologie
- Lois

### 1.2.4 Les Sprints

<b>Sprint</b>	<b>Phase</b>	<b>Fonctionnalité</b>	<b>Date de livraison</b>	<b>Responsables</b>
0	initiale	-création de cahier des charges	09/12/2024	Ingénieures IT
1	conception réalisation Test	-conception du projet -Login/logout -Gestion des admins -Gestion réceptionniste	23/12/2024	Ingénieures dev
2	conception réalisation Test	-Gestion des médecins -Gestion des infirmiers -amélioration -Test final	30/12/2025	Ingénieures dev

FIGURE 1.1 – les sprints

## 1.3 Étude de l'existant

### 1.3.1 Concept métier

Le concept métier de l'application repose sur l'amélioration de la gestion administrative, médicale et logistique des cliniques. Elle vise à centraliser et automatiser les tâches critiques tout en offrant une interface intuitive aux différents utilisateurs (administrateurs, médecins, infirmiers, et patients).

**Les principaux besoins métier sont :**

Gestion des dossiers médicaux : Centraliser les dossiers des patients (historique médical, prescriptions, rapports, etc.) pour un accès rapide et sécurisé.

Planification et gestion des rendez-vous : Simplifier la prise de rendez-vous pour les patients et la gestion des emplois du temps des médecins.

Facturation et paiements : Générer des factures, gérer les paiements et suivre les transactions financières.

Gestion des ressources humaines : Assurer la gestion des employés, leurs plannings, et leurs fiches de performance.

Suivi des stocks et approvisionnement : Gérer efficacement les stocks (médicaments, équipements médicaux) et alerter en cas de niveau critique.

Rapports et statistiques : Générer des tableaux de bord et des statistiques pour une meilleure prise de décision par la direction.

**Ce concept vise à répondre aux problématiques actuelles telles que :**

La dispersion des informations médicales.

Les erreurs dans les plannings et la facturation.

Le manque de suivi en temps réel des performances cliniques.

## 1.4 Étude fonctionnelle

### 1.4.1 Objectifs fonctionnels

Les objectifs fonctionnels décrivent les grandes lignes des fonctionnalités à intégrer dans l'application afin de répondre aux problématiques spécifiques de gestion des cliniques :

**Simplification des processus administratifs :**

Centraliser la gestion des dossiers médicaux, rendez-vous, facturations, et ressources humaines.

**Automatisation des tâches courantes :**

Réduire les erreurs et améliorer l'efficacité grâce à des rappels automatiques, une gestion optimisée des plannings et des alertes pour les stocks.

**Amélioration de l'expérience utilisateur :**

Offrir une interface intuitive et ergonomique permettant à chaque utilisateur (patients, médecins, administrateurs) d'accéder rapidement aux services nécessaires.

**Accès sécurisé et confidentialité :**

Garantir la sécurité des données médicales grâce à une gestion stricte des accès et au respect des normes comme le RGPD.

**Génération de statistiques et rapports :**

Fournir des indicateurs et des rapports en temps réel pour la prise de décisions stratégiques par la direction.



## 1.5 Besoins fonctionnels

L'application Web de gestion des cliniques est conçue pour s'adapter aux besoins spécifiques de différents types d'utilisateurs. Chaque catégorie d'utilisateur a des fonctionnalités dédiées, adaptées à son rôle au sein de la clinique. Voici les détails :

### 1.5.1 Bloc fonctionnel : réceptionniste

#### 1. User stories :

Ce bloc permet le traitement et la manipulation des commandes de la réception , il couvre les fonctionnalités suivantes :

En tant que réceptionniste , Le système permet de :

Ajouter/modifier/supprimer un patient  
Ajouter/modifier/supprimer un rendez-vous  
Consulter la liste des patients et des rendez-vous

#### 2. Règles de gestion :

Référence	Description
<b>RG 01</b>	<p><b>Un réceptionniste est caractériser par :</b></p> <ul style="list-style-type: none"> <li>- Un nom</li> <li>- Un prénom</li> <li>- Date de naissance</li> <li>- Matricule</li> <li>- Date de début de service</li> <li>- Poste</li> </ul>

FIGURE 1.2 – Règles de gestion du réceptionniste

### 1.5.2 Bloc fonctionnel : administrateur

#### 1. User stories :

Ce bloc permet le traitement et la manipulation des personelles , il couvre les

En tant qu'administrateur, Le système permet de :

Ajouter/modifier/supprimer un personnel  
Consulter la liste des personnes

## 2. Règles de gestion :

Référence	Description
<b>RG 01</b>	<b>Un administrateur est caractériser par :</b> <ul style="list-style-type: none"><li>- Un nom</li><li>- Un prénom</li><li>- Date de naissance</li><li>- Matricule</li><li>- Date de début de service</li><li>- Poste</li></ul>

FIGURE 1.3 – Règles de gestion de l'administrateur

### 1.5.3 Bloc fonctionnel : infirmière/infirmier

#### 1. User stories :

Ce bloc permet le traitement et la manipulation des commandes d'une infirmière ou un infirmier,

En tant que infirmière/infirmier, Le système permet de :

Suivre et de modifier l'état des patients

Consulter la liste des réclamations

## 2. Règles de gestion :

Référence	Description
RG 01	Une infirmière ou un infirmier est caractériser par : <ul style="list-style-type: none"><li>- Un nom</li><li>- Un prénom</li><li>- Date de naissance</li><li>- Matricule</li><li>- Date de début de service</li><li>- Poste</li><li>- Spécialité</li></ul>
RG 02	En cas de manipulation de l'état par l'infirmier , en doit confirmer les modification

FIGURE 1.4 – Règles de gestion de infirmière/infirmier

### 1.5.4 Bloc fonctionnel : médecin

#### 1. User stories :

Ce bloc permet le traitement et la manipulation des commandes du médecin ,

En tant que infirmière/infirmier, Le système permet de :

Consulter la liste des patients et des rendez-vous

Suivez et de modifier l'état des patients

Réclamer en cas d'un problème (un problème sur les patients)

Consulter la liste des réclamations

#### 2. Règles de gestion :

Référence	Description
RG 01	<p>Un médecin est caractériser par :</p> <ul style="list-style-type: none"> <li>- Un nom</li> <li>- Un prénom</li> <li>- Date de naissance</li> <li>- Matricule</li> <li>- Date de début de service</li> <li>- Poste</li> <li>- Spécialité</li> </ul>
RG 02	En cas d'un problème , la déclaration de la réclamation est obligatoire

FIGURE 1.5 – Règles de gestion du médecin

## 1.6 Conclusion

La mise en place d'une application de gestion de clinique constitue une solution innovante et indispensable pour répondre aux exigences croissantes des établissements de santé en matière de digitalisation et d'optimisation des processus. À travers une interface conviviale et des fonctionnalités adaptées, ce projet vise à simplifier la gestion administrative, à assurer une meilleure prise en charge des patients et à renforcer la coordination entre les différents acteurs de la clinique.

Cette présentation du projet a permis de mettre en lumière les principaux enjeux auxquels répond cette application, tout en introduisant les objectifs visés. Les prochaines

sections détailleront le processus de développement, les choix techniques effectués ainsi que les résultats obtenus, afin d'évaluer la pertinence de cette solution dans un environnement médical en constante évolution.

## Chapitre 2

### Conception et étude pratique

## 2.1 Introduction

La conception joue un rôle essentiel dans la réalisation de solutions innovantes. Que ce soit dans le domaine de l'ingénierie, de l'architecture, du design ou d'autres domaines créatifs, la conception est le processus clé qui permet de transformer une idée abstraite en une réalité tangible. Ce chapitre se concentre sur l'importance de la conception et explore les différentes étapes et approches nécessaires pour concevoir des solutions innovantes. En résumé, ce chapitre de conception nous permettra de plonger dans l'univers de l'application, commençant par l'élaboration du diagramme de cas d'utilisation et terminer par le diagramme de séquence .

## 2.2 Analyse des besoins

La mise en place d'une application de gestion de clinique repose sur une analyse approfondie des besoins fonctionnels et non fonctionnels afin de garantir une solution efficace, intuitive et conforme aux exigences du secteur médical.

### 2.2.1 Les besoins fonctionnels

Les besoins fonctionnels spécifient les fonctionnalités principales que l'application doit offrir pour répondre aux attentes des utilisateurs. L'application de gestion de clinique doit permettre les opérations suivantes :

**Gestion des patients :** L'application doit permettre d'ajouter, de modifier et de supprimer les informations des patients, notamment leur identité, leur historique médical et leurs rendez-vous. Un formulaire dédié doit être disponible pour saisir ces informations avec une validation des champs obligatoires.

**Gestion des rendez-vous :** L'application doit offrir une fonctionnalité de gestion des rendez-vous entre les patients et les médecins. Les administrateurs doivent pouvoir planifier, reprogrammer ou annuler des rendez-vous tout en assurant un suivi des disponibilités des médecins.

**Interface utilisateur :** L'application doit offrir une interface conviviale et intuitive permettant aux administrateurs d'accéder facilement aux différentes fonctionnalités. Un tableau de bord centralisé doit permettre de consulter rapidement les informations essentielles, telles que les rendez-vous du jour et les patients en attente.

### 2.2.2 Les besoins non fonctionnels

En complément des besoins fonctionnels, il est essentiel de prendre en compte les contraintes non fonctionnelles qui garantissent la qualité, la sécurité et la performance de l'application.

**Sécurité :** L'application doit assurer la protection des données sensibles des patients et du personnel médical. La gestion des accès doit être mise en place pour garantir que seules les personnes autorisées peuvent accéder aux informations confidentielles.

**Performance :** L'application doit être capable de gérer un grand volume de données, notamment les informations des patients et les rendez-vous, sans altérer les performances globales.

**Fiabilité et disponibilité :** L'application doit être disponible en continu pour permettre un accès permanent aux informations critiques, telles que les dossiers médicaux et les rendez-vous.

**Compatibilité :** L'application doit être compatible avec différents systèmes d'exploitation et terminaux, y compris les ordinateurs de bureau, les tablettes et les smartphones, pour offrir une flexibilité maximale.

**Maintenance :** L'application doit être facile à maintenir. Les mises à jour doivent pouvoir être déployées rapidement pour corriger les éventuelles anomalies ou améliorer les fonctionnalités existantes. Un suivi des logs doit également être mis en place pour faciliter la gestion des incidents.

**Conformité :** L'application doit respecter les normes de sécurité et de confidentialité des données médicales, conformément aux réglementations en vigueur dans le secteur de la santé.

**Facilité d'utilisation :** L'interface utilisateur doit être simple et intuitive afin de faciliter la prise en main de l'application par les administrateurs, les médecins et le personnel médical. Chaque fonctionnalité doit être facilement accessible, avec une navigation claire et fluide.

## 2.3 Langage de conception et conception général

### 2.3.1 Langage UML

L'UML (Unified Modeling Language ou Langage de modélisation unifiée en français) est un langage graphique de modélisation informatique. Ce langage est désormais la référence en modélisation objet, ou programmation orientée objet. Cette dernière consiste à modéliser des éléments du monde réel (immeuble, ingrédients, personne, logos, organes du corps...) ou virtuel (temps, prix, compétence...) en un ensemble d'entités informatiques appelées « objet ». 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. L'UML a des applications dans le développement logiciel, mais aussi dans l'industrie (pour modéliser les flux de processus par exemple), dans l'ingénierie ou le marketing.

### 2.3.2 Les différents types de diagrammes UML

L'UML définit 14 types de diagrammes divisés en deux catégories.

1. Les diagrammes de structure représentent les éléments du système, leurs propriétés et leurs relations entre eux :

- Diagramme de classes
- Diagramme d'objets
- Diagramme de composants
- Diagramme de structure composite
- Diagramme d'ensemble
- Diagramme de déploiement
- Diagramme de profil

2. Les diagrammes de comportement représentent les processus et les interactions entre les objets

- Diagramme de cas d'utilisation
- Diagramme d'activité
- Diagramme d'état-transition
- Diagramme de séquence
- Diagramme de communication
- Diagramme de temps
- Diagramme d'aperçu d'interaction

## 2.4 Conception général

### 2.4.1 Diagrammes de cas d'utilisation

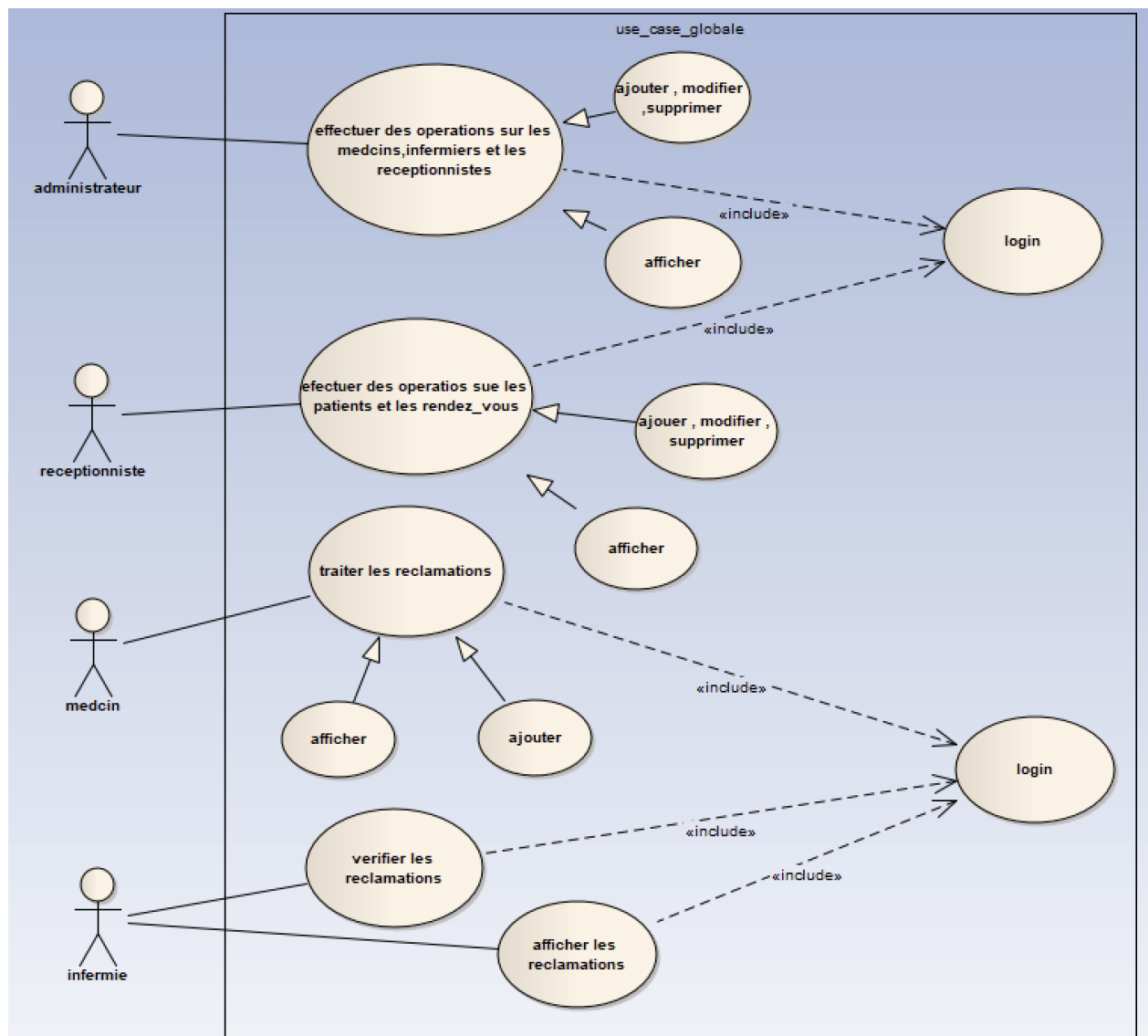


FIGURE 2.1 – diagramme de cas d'utilisation globale

## 2.4.2 Diagrammes de classe

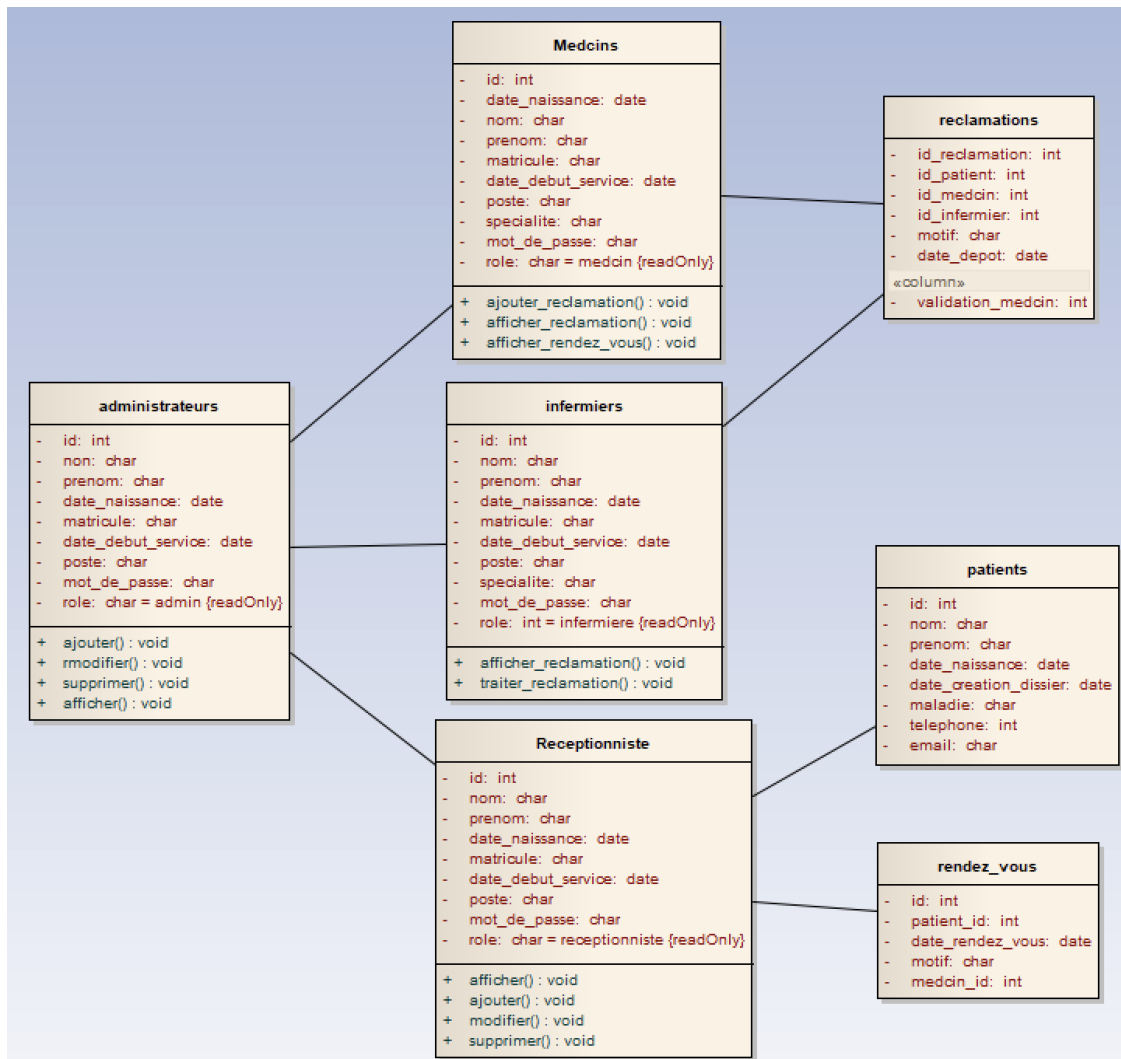


FIGURE 2.2 – diagramme de classe

### 2.4.3 Diagramme de séquence cas d'ajout d'un patient

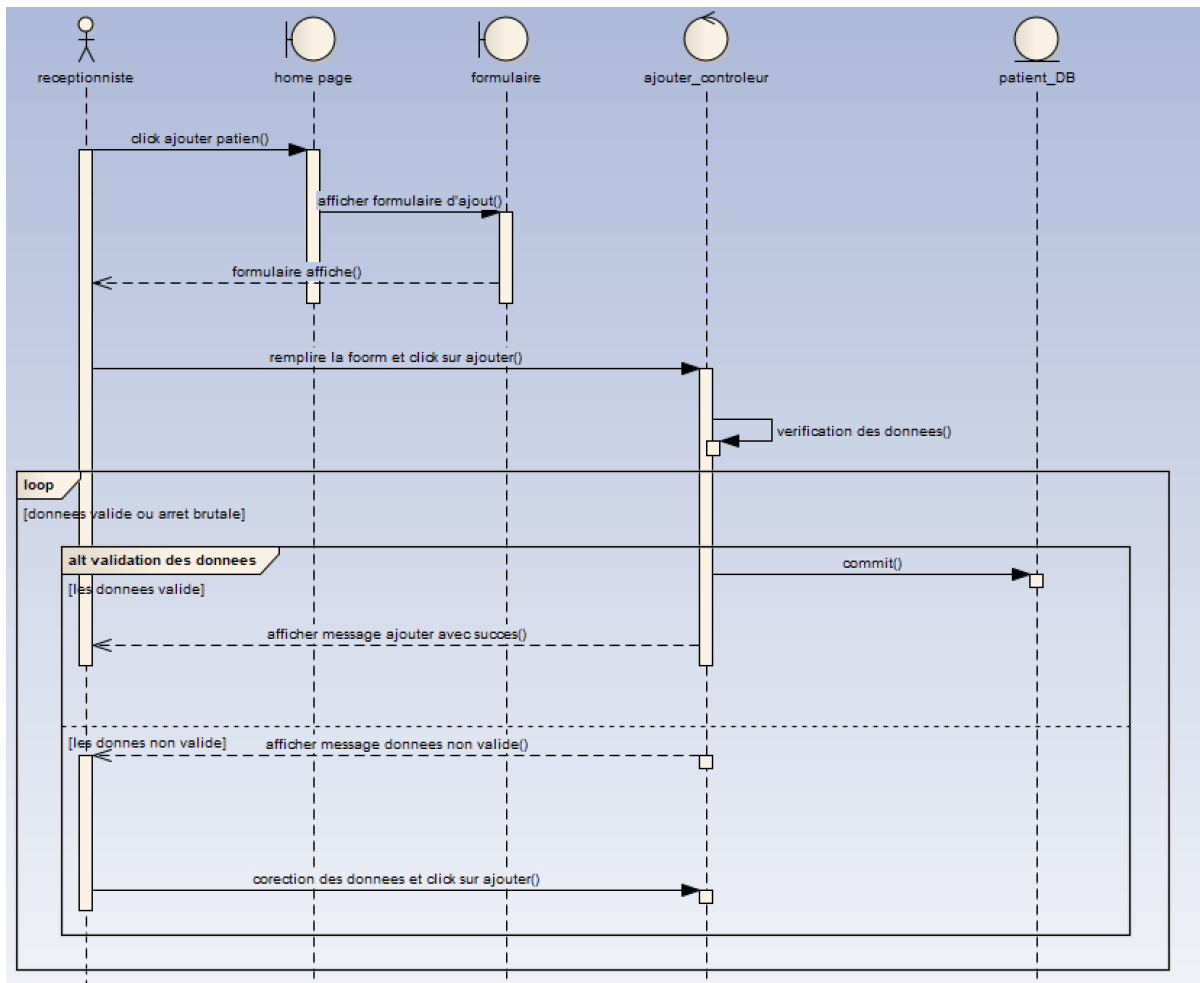


FIGURE 2.3 – diagramme de séquence cas d'ajout d'un patient

### 2.4.4 Diagramme de séquence cas de modification des données d'un patient

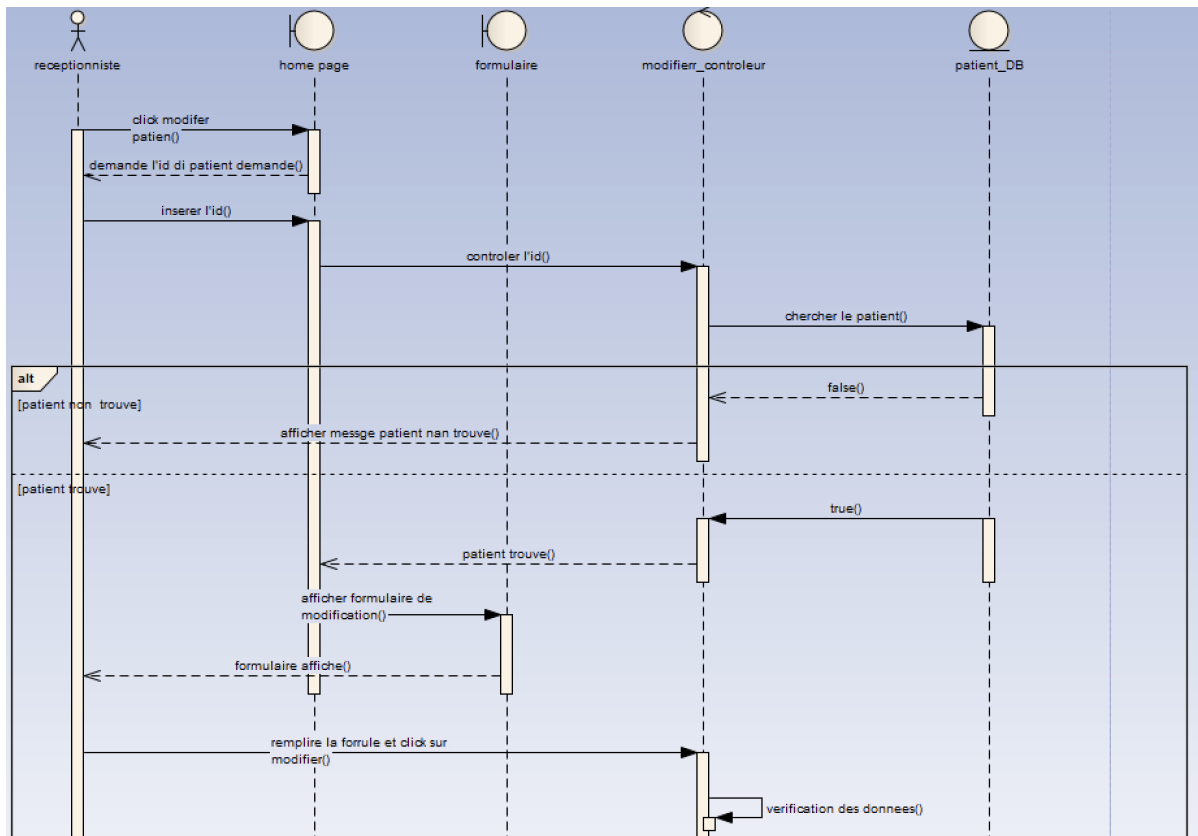


FIGURE 2.4 – diagramme de séquence cas de modification d'un patient 1

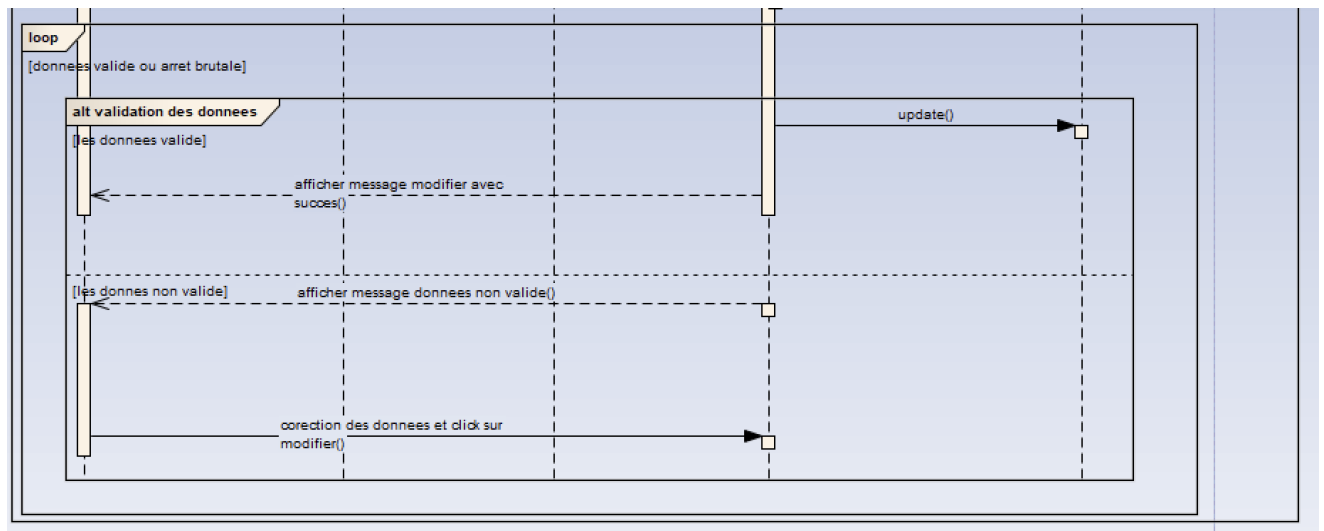


FIGURE 2.5 – diagramme de séquence cas de modification d'un patient 2

## 2.5 Description Textuelle

### 2.5.1 cas : traiter les patient

#### Partie 1 : Identification

##### Titre :

Traiter les patients.

##### Résumé :

Ce cas d'utilisation permet de gérer les informations des patients dans le système. Les utilisateurs autorisés peuvent ajouter, modifier ou supprimer des informations relatives aux patients.

##### Acteurs :

Acteurs principaux : Réceptionniste. Acteurs secondaires : Base de données patients.

##### Date de création :

23/12/2025

##### Dernière mise à jour :

02/01/2025

##### Responsable :

Responsable des systèmes d'information.

#### Partie 2 : Description des Scénarios

##### Préconditions :

L'utilisateur doit être authentifié et disposer des droits d'accès appropriés. Le système doit être connecté à la base de données et opérationnel.

##### Scénario nominal :

L'utilisateur accède à l'interface de gestion des patients. L'utilisateur sélectionne une action (ajouter, modifier ou supprimer)

Ajouter : L'utilisateur remplit le formulaire avec les informations du patient. Le système valide les données et les enregistre dans la base de données.

**Modifier :**

L'utilisateur recherche le patient à modifier. L'utilisateur met à jour les informations nécessaires. Le système enregistre les modifications.

**Supprimer :**

L'utilisateur sélectionne un patient à supprimer. Le système demande confirmation. Le système supprime les informations du patient de la base de données. Le système affiche un message de confirmation et met à jour la liste des patients.

**Scénarios alternatifs :**

SA1 : Données invalides lors de l'ajout ou la modification. Le système informe l'utilisateur des erreurs et lui demande de corriger les données.

SA2 : Le patient à supprimer est lié à des données critiques (par exemple, un dossier médical ouvert). Le système bloque la suppression et affiche un message expliquant la raison.

**Scénarios d'exception :**

SE1 : La base de données est inaccessible. Le système affiche un message d'erreur et enregistre l'opération comme échouée.

SE2 : L'utilisateur n'a pas les droits nécessaires pour effectuer l'action. Le système refuse l'accès et enregistre l'événement pour audit.

**Postconditions :**

Les informations du patient sont ajoutées, modifiées ou supprimées correctement dans la base de données.

## 2.5.2 cas : consultation de la liste des patient et des rendez-vous

### Partie 1 : Identification

**Titre :**

Consulter la liste des patients et des rendez-vous.

**Résumé :**

Ce cas d'utilisation permet aux utilisateurs autorisés de visualiser la liste complète des patients et des rendez-vous enregistrés dans le système.

**Acteurs :**

Acteurs principaux : Médecin, Secrétaire médicale, infirmier. Acteurs secondaires : Système de gestion des rendez-vous et des patients.

**Date de création :**

23/12/2025

**Dernière mise à jour :**

02/01/2025

**Responsable :**

Responsable des opérations.

### Partie 2 : Description des Scénarios

**Préconditions :**

L'utilisateur doit être authentifié avec des droits d'accès pour consulter ces informations. La base de données des patients et des rendez-vous doit être opérationnelle.

**Scénario nominal :**

L'utilisateur accède à l'interface de consultation. L'utilisateur sélectionne la liste qu'il

souhaite consulter (patients ou rendez-vous).

Pour les patients :

Le système affiche la liste des patients triée par date d'ajou.

Pour les rendez-vous :

Le système affiche la liste des rendez-vous triée par date d'ajout .

Le système affiche les informations demandées de manière claire et organisée.

**Scénarios alternatifs :**

SA1 : La liste contient un grand nombre d'éléments. Le système pagine les résultats pour éviter des temps de chargement excessifs.

**Scénarios d'exception :**

SE1 : La base de données est inaccessible. Le système affiche un message d'erreur et indique que l'opération ne peut pas être effectuée pour le moment.

SE2 : L'utilisateur n'a pas les droits pour consulter une liste spécifique. Le système refuse l'accès et affiche un message expliquant la restriction.

**Postconditions :**

L'utilisateur a accès à la liste des patients ou des rendez-vous demandés.

Les informations affichées sont exactes et à jour.

## 2.6 Conclusion

Nous avons structuré notre application autour de compétences architecturales avec une distinction nette, et l'aide fut fournie par des diagrammes de conception pour donner le développement clair et consistant. A travers plusieurs schémas de cahier de charges, séquences de classe et autres diagrammes, nous avons pu tracer des fondations solides pour démarrer et réussir notre projet.



## Chapitre 3

### Mise en place l'application

## 3.1 Introduction

Pour faciliter la gestion de l'information des cliniques et améliorer l'efficacité des processus administratifs et médicaux, le développement d'une application du bureau dédiée à la gestion des patients, des rendez-vous et des dossiers médicaux est considérée comme une solution optimale. Cette application, conçue avec Java Swing, permet de centraliser et de gérer efficacement les données essentielles au sein d'une interface utilisateur conviviale et réactive.

## 3.2 Outils

### 3.2.1 JAVA SWING



FIGURE 3.1 – JAVA SWING logo

Java Swing est une bibliothèque graphique incluse dans le JDK (Java Development Kit) qui permet de créer des interfaces utilisateurs pour les applications de bureau. Elle offre une large gamme de composants visuels tels que des fenêtres, des boutons, des menus et des champs de texte, facilitant ainsi le développement d'interfaces intuitives et réactives pour des applications Java.

### 3.2.2 composants de Java Swing

Cette figure illustre la hiérarchie des principaux composants de Java Swing, utilisés pour créer des interfaces graphiques dans les applications de bureau. Au sommet de la hiérarchie se trouve le bloc principal intitulé Swing Components, qui regroupe l'ensemble des composants graphiques de l'interface utilisateur.

Les composants sont ensuite classés en différentes catégories :

ImageIcon : Ce composant permet d'afficher des images dans une interface graphique.

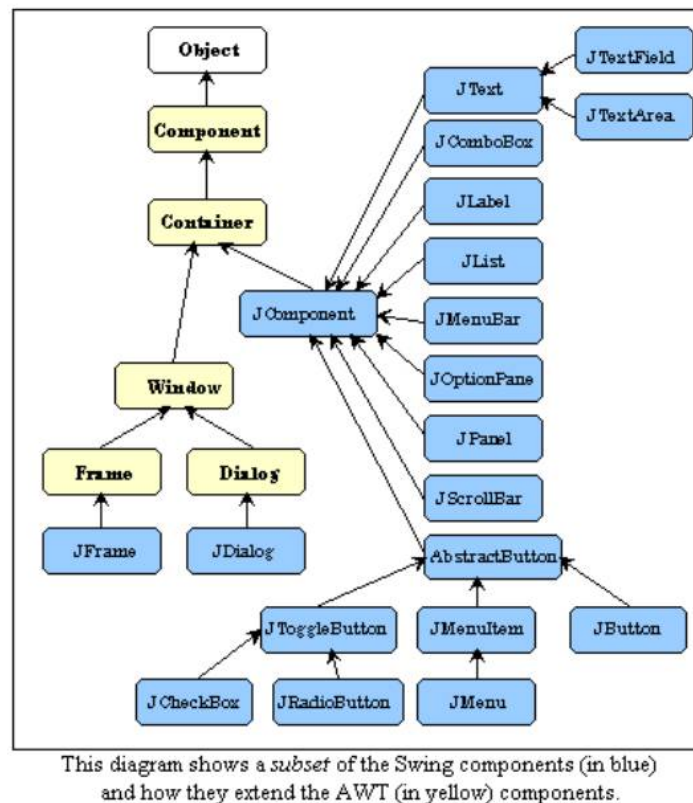


FIGURE 3.2 – composants de Java Swing

JCheckBox : Une case à cocher permettant de sélectionner ou désélectionner une option.

**JButton** : Un bouton cliquable qui déclenche une action spécifique.

**JLabel** : Un composant permettant d'afficher du texte statique ou une image.

**JPasswordField** : Un champ de saisie conçu pour la saisie de mots de passe, masquant le texte entré.

JText (regroupant plusieurs champs de texte) :

UITextField : Un champ de saisie permettant d'entrer une seule ligne de texte.

JTextArea : Un champ de saisie permettant d'entrer plusieurs lignes de texte.

JList : Un composant permettant d'afficher une liste de plusieurs éléments.

JComboBox : Une boîte déroulante permettant de sélectionner une option parmi plusieurs.

JFileChooser : Un composant utilisé pour ouvrir une boîte de dialogue de sélection de fichiers.

Cette figure montre la variété des composants fournis par Java Swing pour concevoir des interfaces interactives et dynamiques, allant des boutons aux champs de texte, en passant par les listes déroulantes et les sélecteurs de fichiers. Ces composants sont essentiels pour la création d'applications de bureau complètes et ergonomiques.

### 3.2.3 NetBeans IDE



FIGURE 3.3 – netbeans logo

NetBeans est un environnement de développement intégré (IDE) open source qui supporte plusieurs langages de programmation, dont Java, PHP, et HTML. Il fournit des outils pour l'écriture de code, le débogage, la gestion de projets et la construction d'applications, tout en offrant un support pour des frameworks populaires, ce qui en fait un choix populaire pour les développeurs Java.

### 3.2.4 MAMP



FIGURE 3.4 – js logo

MAMP est une plateforme qui permet de configurer facilement un environnement de développement local pour les applications web. Elle inclut Apache (serveur web), MySQL (système de gestion de base de données) et PHP, ce qui permet de tester et de développer des sites web dynamiques sur un serveur local avant de les déployer en production.

### 3.2.5 PHPmyadmin



FIGURE 3.5 – phpmyadmin

phpMyAdmin est une application web populaire qui permet de gérer les bases de données MySQL via une interface graphique. Elle permet de créer, modifier, interroger et administrer les bases de données sans avoir besoin de connaissances avancées en ligne de commande, rendant la gestion des bases de données plus accessible aux développeurs.

### 3.2.6 Enterprise Architect

Enterprise Architect est un logiciel de modélisation UML (Unified Modeling Language) utilisé pour concevoir des architectures de systèmes d'information complexes. Il permet de créer des diagrammes pour modéliser les processus, les systèmes, et les structures d'entreprise, ce qui facilite la compréhension et la gestion des projets informatiques à grande échelle.

**Enterprise Architect offre des fonctions de modélisation orientée objet basée, avec un support complet d'UML, conforme aux normes UML 2.x :**

Diagrammes d'activité

Diagrammes de séquence

Diagrammes de communication

Diagrammes chronologiques



FIGURE 3.6 – enterprise architect logo

Diagrammes de transitions d'états

Diagrammes de flux d'informations

Diagrammes de vue d'ensemble d'interactions

Diagrammes de profils

Diagrammes de classes

Diagrammes de structure composite

Diagrammes de composants

Diagrammes d'objets

Diagrammes de paquetages

Diagrammes de cas d'utilisation

**Enterprise Architect dispose également de capacités de modélisation en SysML :**

Diagrammes de besoins

Diagrammes de blocs

Diagrammes internes de blocs

Enterprise Architec permet également de modéliser avec des représentations graphiques plus anciennes telles que :

Organigrammes

Diagrammes Entités-Associations (ERD)

Diagrammes de flux de données (DFD)

## 3.3 Premier sprint

### 3.3.1 Page de connexion :

#### 3. Réalisation :

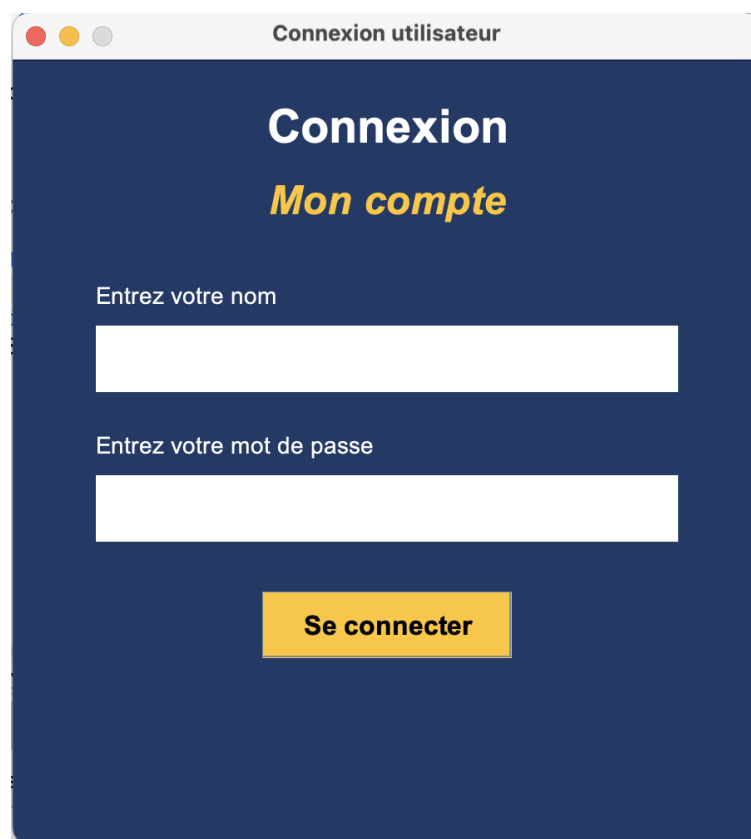


FIGURE 3.7 – page de connexion

La page de connexion de l'application offre une interface simple et sécurisée, permettant aux utilisateurs de se connecter à l'application en utilisant uniquement leur nom d'utilisateur et mot de passe. Dès la connexion, l'interface détecte automatiquement le rôle de l'utilisateur, que ce soit réceptionniste, médecin, infirmier ou administrateur, grâce à des mécanismes d'authentification intégrés. Cela garantit que chaque utilisateur accède à des

fonctionnalités et informations adaptées à son profil, assurant ainsi une gestion fluide et personnalisée des tâches au sein de la clinique.

### 3.3.2 Gestion des réceptionnistes :

#### 1. User stories :

Ce bloc permet le traitement et la manipulation des commandes de la réception , il couvre les fonctionnalités suivantes :

En tant que réceptionniste , Le système permet de :

Ajouter/modifier/supprimer un patient  
Ajouter/modifier/supprimer un rendez-vous  
Consulter la liste des patients et des rendez-vous

#### 2. diagramme de cas d'utilisation :

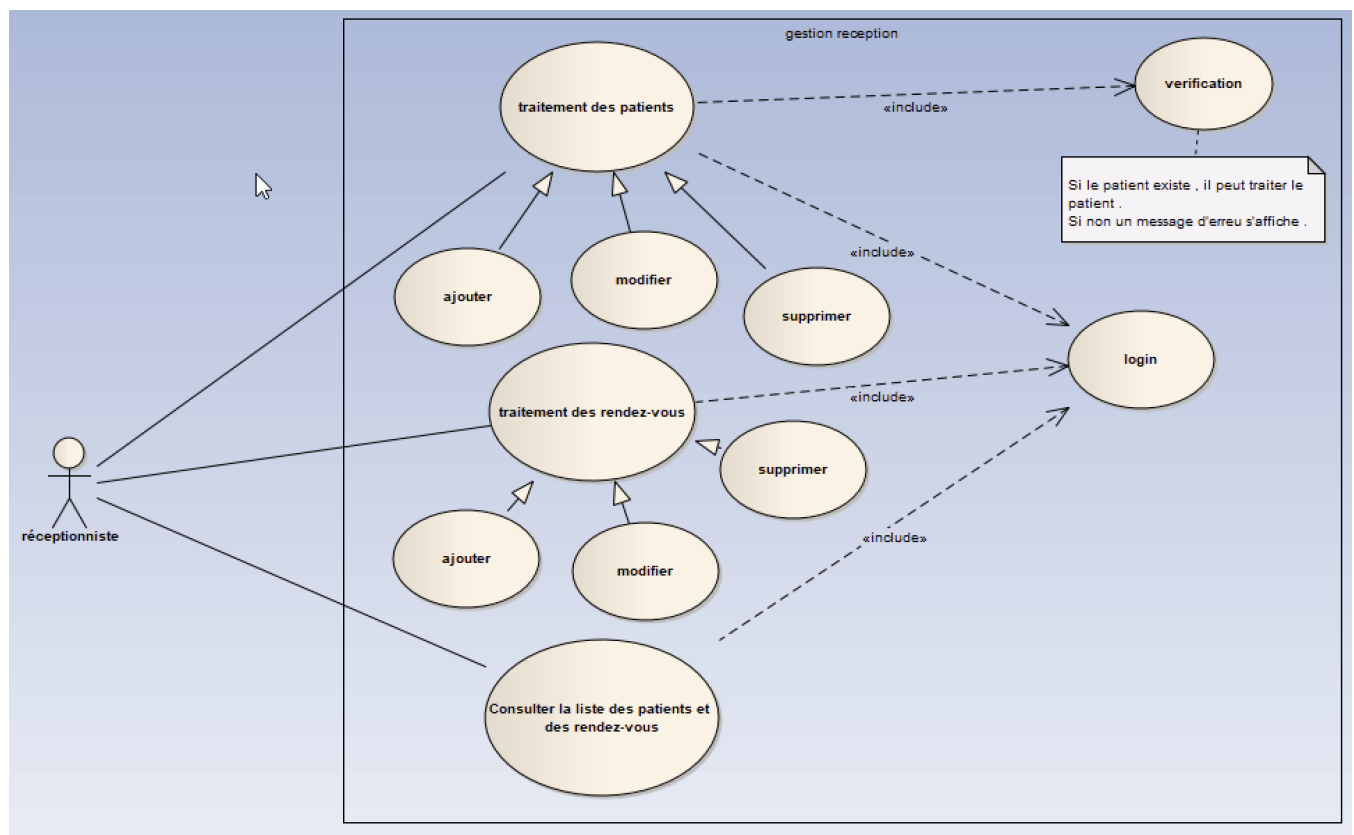


FIGURE 3.8 – Diagramme de cas d'utilisation des réceptionnistes



### 3. Réalisation :



FIGURE 3.9 – page du réceptionniste

La page du réceptionniste de l'application offre des fonctionnalités essentielles pour assurer une gestion efficace des patients et des rendez-vous. Le réceptionniste peut afficher, ajouter, modifier et supprimer les informations des patients, tout en ayant la possibilité de gérer leurs rendez-vous en fonction des disponibilités des médecins. Cette interface permet également de suivre l'historique des rendez-vous et d'effectuer des mises à jour ou des annulations selon les besoins. Enfin, pour garantir la sécurité des données, une option de déconnexion est disponible, permettant au réceptionniste de quitter la session de manière sécurisée après chaque utilisation.

### 3.3.3 Gestion des administrateurs :

#### 1. User stories :

Ce bloc permet le traitement et la manipulation des personelles , il couvre les

En tant qu'administrateur, Le système permet de :

Ajouter/modifier/supprimer un personnel

Consulter la liste des personelles

#### 2. Riagramme de cas d'utilisation :

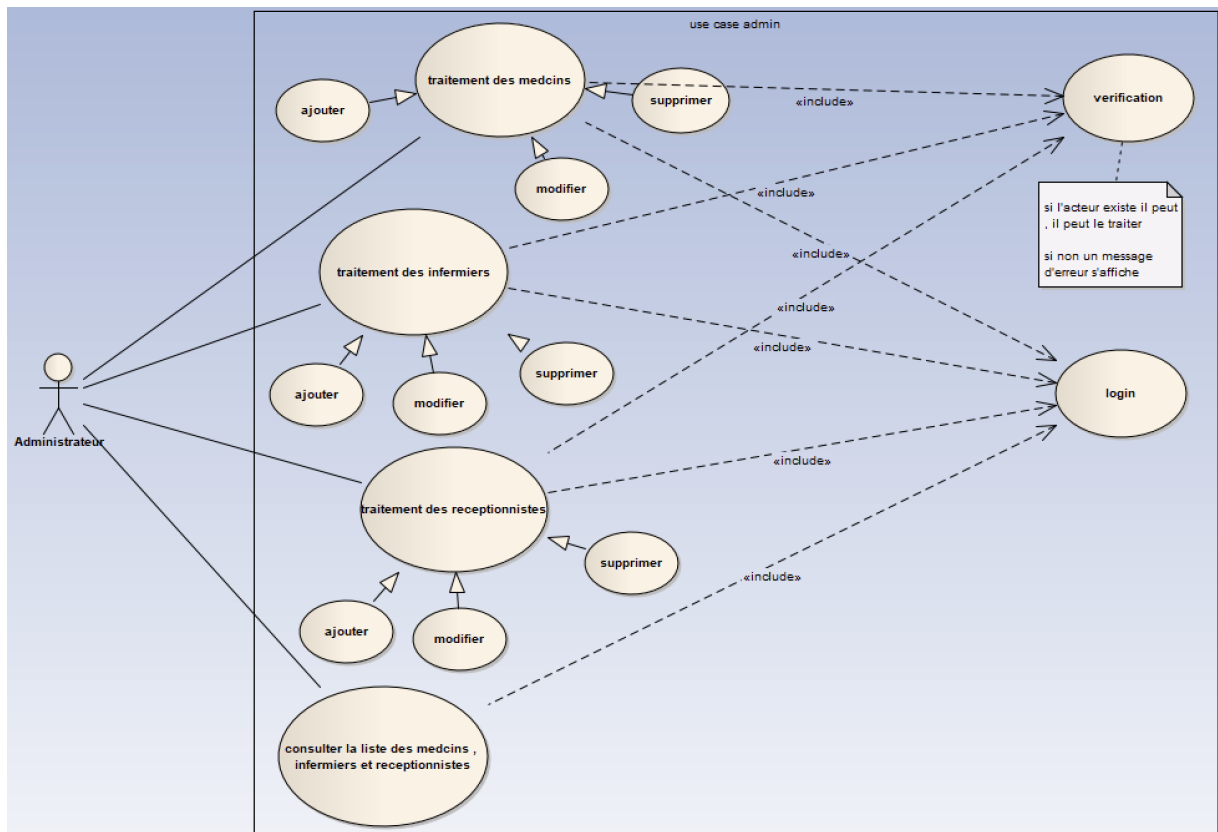


FIGURE 3.10 – Diagramme de cas d'utilisation d'administrateurs

#### 3. Réalisation :

La page de l'administrateur de l'application offre un ensemble complet de fonctionnalités pour gérer les différents utilisateurs de la clinique. L'administrateur peut afficher, modifier et ajouter des informations relatives aux médecins, infirmiers et réceptionnistes, permettant ainsi une gestion fluide et à jour du personnel médical et administratif. Cette interface intuitive permet de mettre à jour les détails des employés, de les ajouter au système, ou de les supprimer si nécessaire. De plus, l'administrateur dispose d'une option de

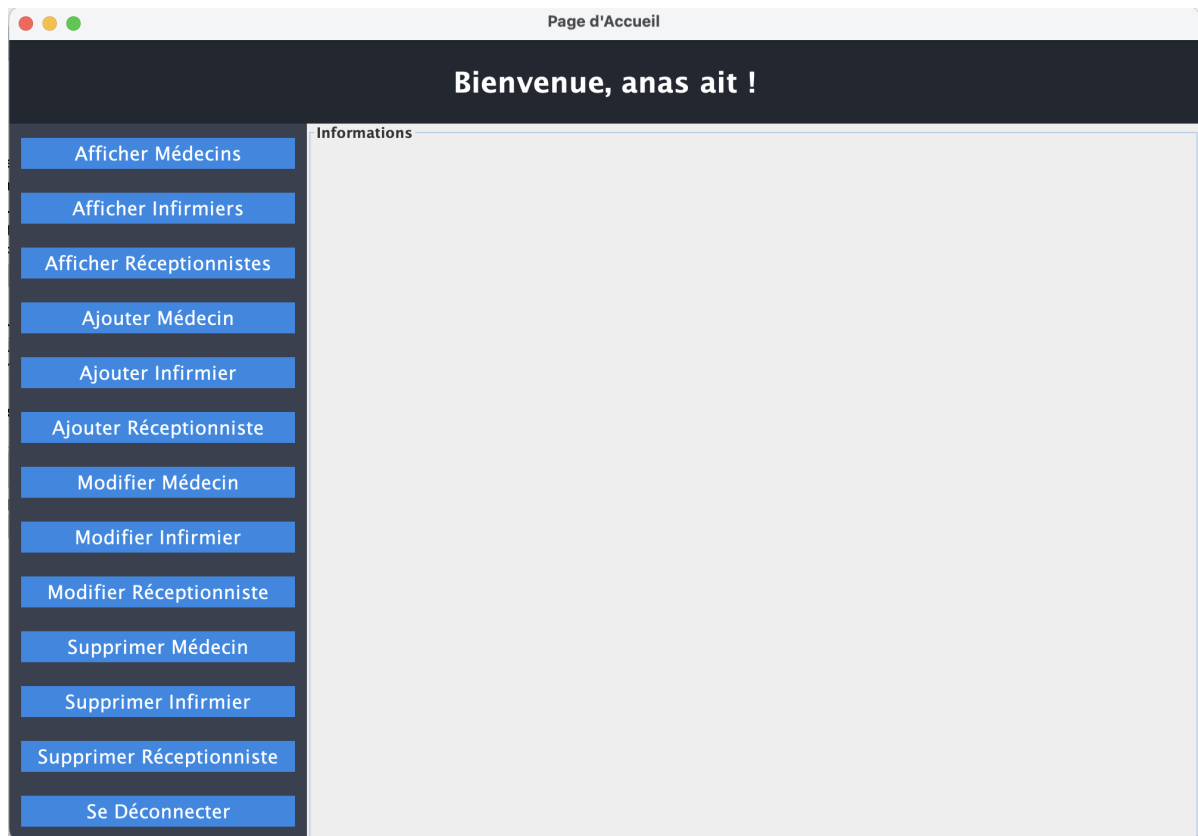


FIGURE 3.11 – page d'administrateur

déconnexion sécurisée pour garantir la confidentialité des informations à la fin de chaque session.

## 3.4 Deuxième sprint

### 3.4.1 Gestion des infirmières/infirmiers :

#### 1. User stories :

Ce bloc permet le traitement et la manipulation des commandes d'une infirmière ou un infirmier,

En tant que infirmière/infirmier, Le système permet de :

Suivre et de modifier l'état des patients

Consulter la liste des réclamations

#### 2. diagramme de cas d'utilisation :

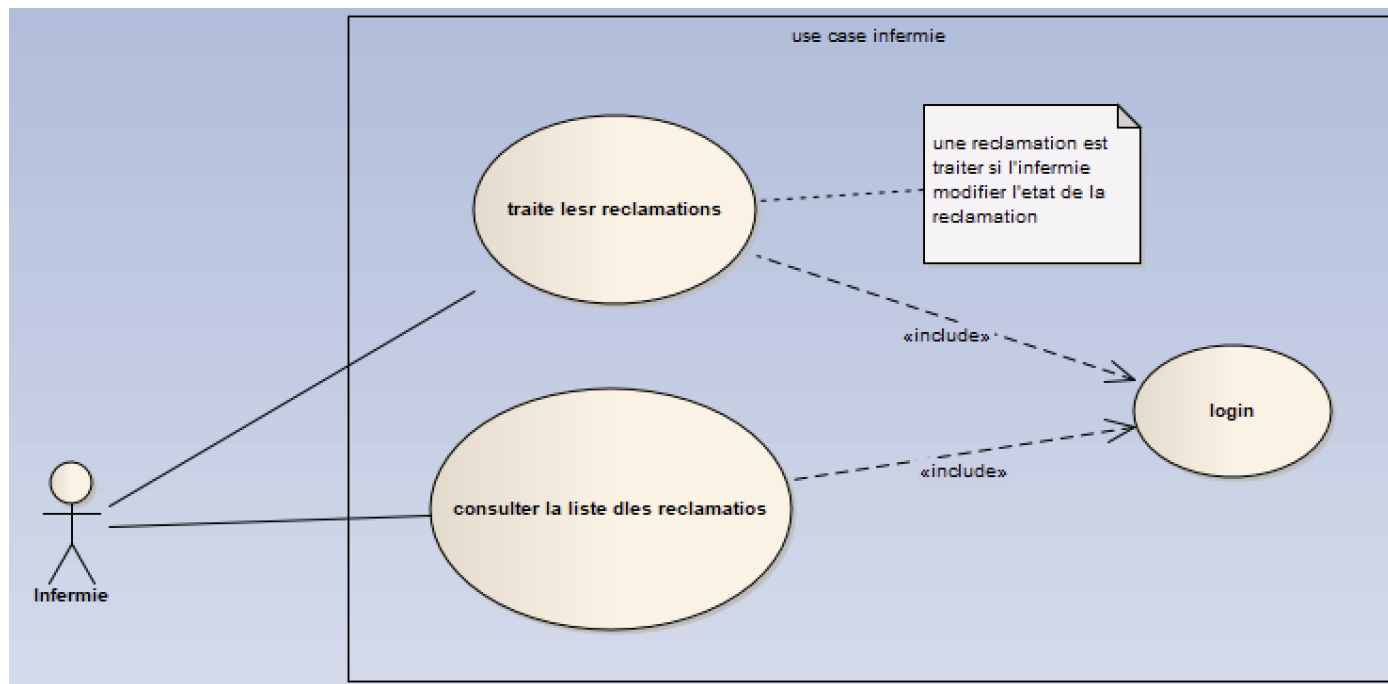


FIGURE 3.12 – Diagramme de cas d'utilisation des infirmières/infirmiers

### 3. Réalisation :

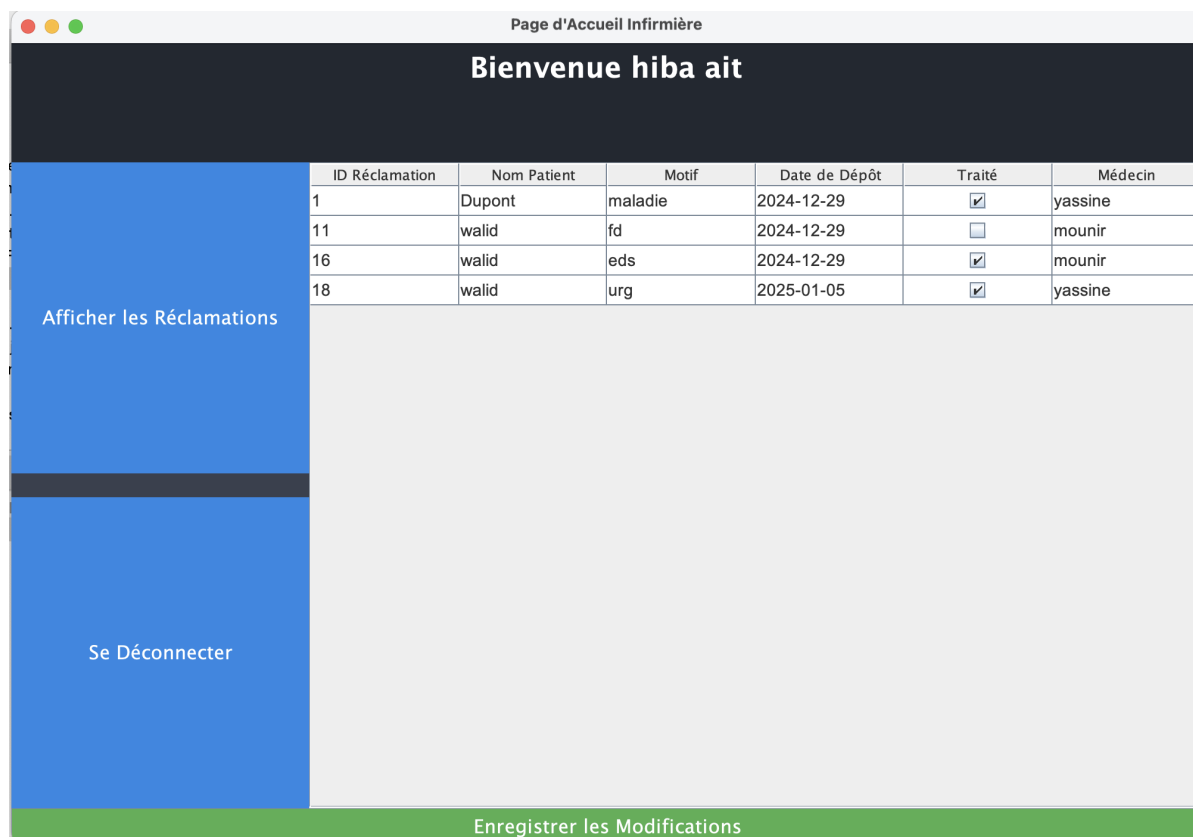


FIGURE 3.13 – page des infirmières/infirmiers

La page des infirmières et infirmiers de l'application permet de gérer efficacement les réclamations des médecins. L'infirmier(ère) peut consulter les réclamations formulées par les médecins concernant les soins ou les traitements des patients. Une fois qu'une réclamation a été prise en charge, l'infirmier(ère) peut mettre à jour son statut pour indiquer qu'elle est désormais traitée, assurant ainsi un suivi clair et précis des actions entreprises. L'interface est conçue pour faciliter la gestion des réclamations tout en garantissant une bonne communication entre les différents membres du personnel médical. De plus, une option de déconnexion est disponible pour permettre à l'infirmier(ère) de quitter la session de manière sécurisée à la fin de son travail.

### 3.4.2 Gestion des médecins :

#### 1. User stories :

Ce bloc permet le traitement et la manipulation des commandes du médecin ,

En tant que infirmière/infirmier, Le système permet de :

Consulter la liste des patients et des rendez-vous

Suivez et de modifier l'état des patients

Réclamer en cas d'un problème (un problème sur les patients)

Consulter la liste des réclamations

#### 2. diagramme de cas d'utilisation :

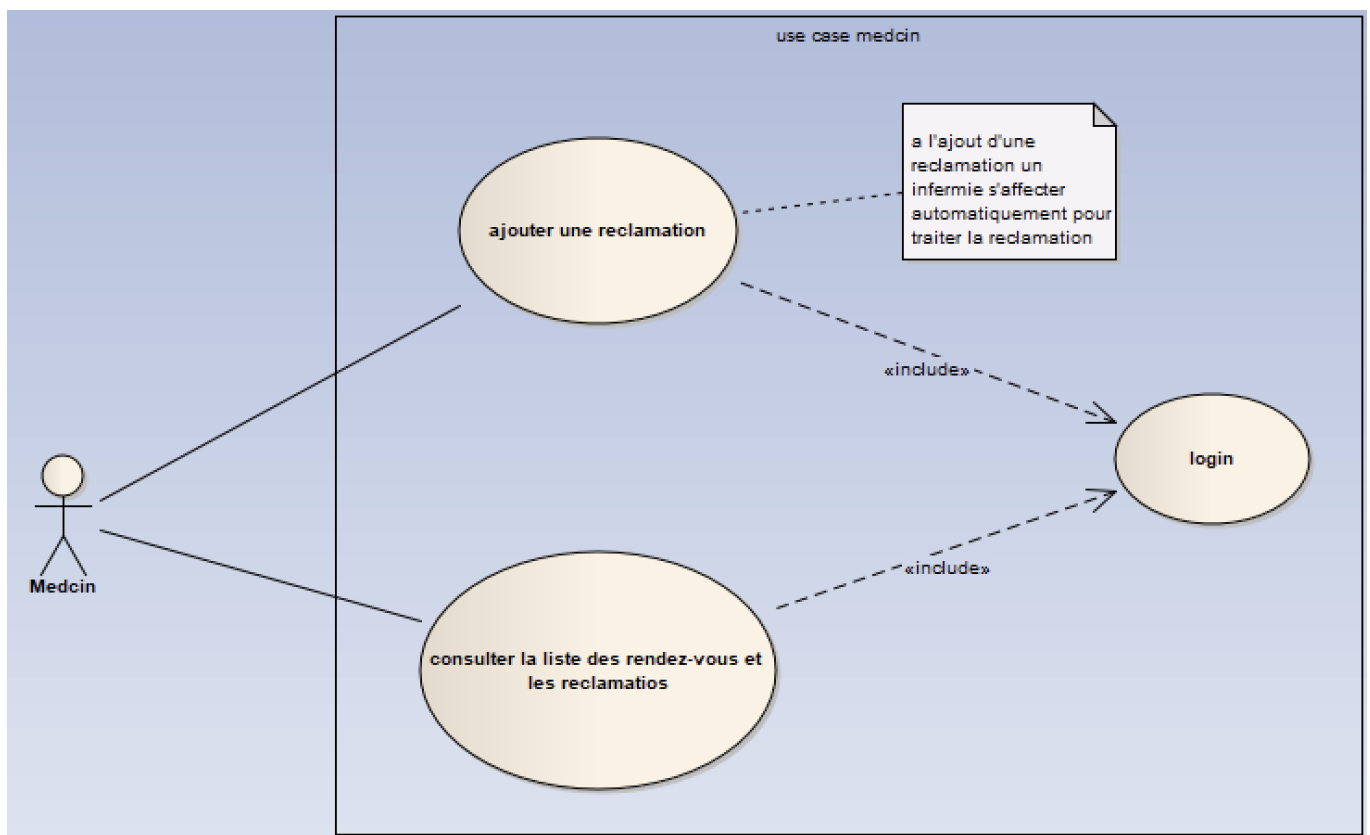


FIGURE 3.14 – Diagramme de cas d'utilisation des médecins

### 3. Réalisation :



FIGURE 3.15 – page des médecins

La page des médecins de l'application offre un ensemble de fonctionnalités essentielles pour une gestion efficace des soins. Le médecin peut consulter les réclamations existantes et en ajouter de nouvelles si nécessaire, permettant ainsi une communication fluide avec les autres membres du personnel. Il a également la possibilité d'afficher tous les rendez-vous qui lui sont attribués, facilitant ainsi la planification de sa journée et la gestion de son emploi du temps. Cette interface intuitive permet au médecin de suivre de manière optimale ses interactions avec les patients et les autres professionnels de la clinique. Enfin, une option de déconnexion est disponible, garantissant la sécurité et la confidentialité des informations une fois la session terminée.

## 3.5 Conclusion

Pour conclure cette section consacrée à l'étude pratique de notre projet, nous avons minutieusement analysé les outils de développement utilisés et la séquence des interfaces. Nos éléments sont essentiels afin de garantir la solidité technique et l'ergonomie de notre application. Le choix méticuleux des outils et la planification minutieuse des interfaces représentent deux étapes cruciales pour réaliser notre vision.

# Conclusion générale

En conclusion, notre application de gestion de clinique, développée avec Java Swing, constitue une avancée significative dans la simplification des processus administratifs et médicaux au sein des établissements de santé. Nous avons conçu une plateforme de bureau intuitive et sécurisée permettant aux professionnels de santé de gérer efficacement les dossiers des patients, les rendez-vous, les traitements et les suivis médicaux.

Grâce à l'utilisation de technologies robustes telles que Java Swing et à une approche de conception orientée utilisateur, nous avons intégré des fonctionnalités clés, notamment la gestion des dossiers médicaux, la planification des rendez-vous, la génération de rapports et la gestion des alertes pour les médecins et les patients. Ces outils permettent aux cliniques d'améliorer leur efficacité opérationnelle, de réduire les erreurs administratives et d'assurer un meilleur suivi des patients.

Le développement de cette application nous a permis de relever plusieurs défis techniques, notamment la création d'une interface graphique fluide et réactive, la gestion des événements utilisateurs et l'optimisation des interactions entre les différents modules de l'application. Ces obstacles nous ont permis de renforcer nos compétences en développement Java, en conception d'interfaces graphiques et en gestion de projet.

Nous sommes convaincus que notre application apportera une valeur ajoutée considérable aux cliniques et cabinets médicaux en leur offrant un outil complet, fiable et facile à utiliser. Les leçons apprises et les succès obtenus au cours de ce projet constituent une base solide pour de futures améliorations. Ces évolutions pourraient inclure l'ajout de fonctionnalités telles que l'intégration de bases de données centralisées, la gestion des paiements et le suivi des prescriptions, assurant ainsi que notre solution répondra aux besoins croissants du secteur médical.



# Perspectives

Le développement de l'application de gestion de clinique ouvre la voie à plusieurs améliorations et évolutions futures qui permettraient d'enrichir ses fonctionnalités et d'élargir son champ d'application. Voici quelques perspectives prometteuses pour rendre l'application plus complète, performante et adaptée aux besoins des professionnels de la santé :

**Intégration des méthodes de paiement automatisées** L'ajout d'une fonctionnalité permettant de gérer les paiements directement au sein de l'application représenterait une avancée significative. Il serait possible d'intégrer des modules de paiement électronique pour permettre aux patients de régler leurs consultations via des moyens sécurisés, tels que les cartes bancaires, les portefeuilles électroniques ou les virements bancaires. Cette automatisation réduirait les tâches administratives liées à la gestion des paiements et garantirait une traçabilité financière plus rigoureuse.

**Création d'interfaces dédiées aux patients** L'ajout d'interfaces spécifiques pour les patients permettrait à ces derniers d'accéder à leur dossier médical en ligne, de consulter leurs rendez-vous, de recevoir des notifications sur les traitements à suivre ou encore de prendre des rendez-vous directement via l'application. Cette fonctionnalité renforcerait la relation patient-clinique en favorisant une communication plus fluide et un suivi médical personnalisé.

**Gestion des ressources médicales et des stocks** Une autre perspective importante consisterait à intégrer un module de gestion des ressources médicales et des stocks. Cette fonctionnalité permettrait de suivre en temps réel les équipements, les fournitures médicales et les médicaments disponibles dans la clinique. Elle offrirait des alertes automatiques en cas de faible stock, assurant ainsi une meilleure gestion logistique et évitant les ruptures d'approvisionnement.

**Ajout de fonctionnalités de reporting avancé** L'application pourrait bénéficier de fonctionnalités de reporting avancé permettant aux responsables de la clinique de suivre les performances de leur établissement. Il serait possible de générer des rapports statistiques sur le nombre de patients, les consultations, les recettes financières, ou encore les performances des différents services médicaux.

**Amélioration de la sécurité des données** Enfin, la sécurité des données patient étant une priorité dans le secteur médical, il serait essentiel de renforcer la protection des informations sensibles par des méthodes d'authentification avancées (comme la double authentification) et l'implémentation de protocoles de chiffrement pour les données échangées.

# Webographie

1. Pour phpmyadmin : <https://www.phpmyadmin.net/>
2. Pour mamp : <https://www.mamp.info/>
3. Pour entreprise architecte : <https://sparxsystems.fr/>
4. Pour l'éditeur netbeans : <https://netbeans.apache.org/front/main/index.html>
5. Pour java swing (guide) : <https://www.javatpoint.com/java-swing>