



## Rapport de Mini Projet en Java



# Conception et Réalisation d'une Application de Gestion d'un établissement Hospitalier.

Par

*Aya Zouay*

Deuxième année licence fondamentale en sciences de  
l'informatique

Professeur : Monsieur Walid Karamti et Madame Rahma Dimassi

Année Universitaire 2017 - 2018

# Sommaire

<b>SOMMAIRE .....</b>	<b>2</b>
<b>INTRODUCTION GENERALE .....</b>	<b>3</b>
<b>Présentation générale du projet .....</b>	<b>4</b>
1.1 Introduction .....	4
1.2 Problématique et Objectif .....	5
1.2.1 Problématiques du système: .....	5
1.2.1 Objectif de cette application: .....	5
<b>Analyse et Conception de Système .....</b>	<b>6</b>
2.1 Introduction.....	6
2.2 Diagramme de cas d'utilisation :.....	6
Tableau 2.2: Description textuelle du cas d'utilisation « Ajouter patient ».....	8
Tableau 2.2: Description textuelle du cas d'utilisation « S'authentifier ».....	9
Tableau 2.3: Description textuelle du cas d'utilisation « Modifier patient ».....	10
2.3 Diagramme d'activité.....	10
Diagramme d'activité de recherche.....	13
2.4 Diagramme de classes .....	14
Diagramme de classes .....	14
Conclusion: .....	15
<b>Réalisation de Système .....</b>	<b>16</b>
3.1 Introduction :.....	16
3.2 Environnements de travail .....	16
3.3 Réalisation : .....	19
Conclusion :.....	30
<b>CONCLUSION GENERALE .....</b>	<b>31</b>

# Introduction générale

L'informatique ne cesse d'envahir les différents domaines des activités humaines. Cela s'explique par son apport incontestable pour ceux qui l'utilisent. En effet, cet outil permet entre autre l'automatisation des traitements, la conservation des données, l'exécution rapide des tâches, etc.

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

Les hôpitaux font partie intégrante des établissements que l'informatique pourra beaucoup aider. D'où la nécessité d'introduire l'informatique dans les administrations hospitalières.

## Présentation générale du projet

---

### 1.1 Introduction

Cette application est développée pour la gestion d'un établissement hospitalier (hôpital, clinique..) et sera utilisé par les personnels de l'établissement.

L'administrateur a le droit d'utiliser les principales fonctionnalités de l'application. Il peut gérer les patients, les médecins, les réceptionnistes et les rendez-vous. Il peut ajouter, modifier supprimer et chercher les différents informations. Tout médecin peut également consulter ses rendez-vous et voir son profil.

Le réceptionniste utilise l'application pour gérer les patients et les rendez-vous et il peut consulter son profil.

Chaque personne est identifié par son identifiant et caractérisé par son nom, prénom, Age, adresse, groupe sanguin, Numéro téléphone.. Le médecin est caractérisé par son spécialité et le réceptionniste par son qualification.

L'application doit permettre

- La mise à jour la base des données lorsque cela est demandé par l'utilisateur.

- L'ajout des informations.
- L'affichage des informations pour l'administrateur.
- La recherche des informations.

## 1.2 Problématique et Objectif

### 1.2.1 Problématiques du système:

Pour un hôpital qui dispose d'un système de fonctionnement manuel :

- Manque de sécurité (information, authentification)
- Manque de fiabilité.
- Perte et erreurs de données et de documents à cause du volume important des informations traités.
- Recherche difficile sur les registres qui engendre une perte de temps.
- Nombre important des archives accumulés qui engendre une difficulté de stockage
- Dégradation et régression des archives à force de leur utilisation trop fréquente.

### 1.2.1 Objectif de cette application:

- Faciliter le processus de gestion du patient.
- Faciliter le processus de gestion du personnel.
- la disponibilité permanente des informations.
- Gain de temps.
- Faciliter le processus de réservation le rendez-vous.

## Analyse et Conception de Système

---

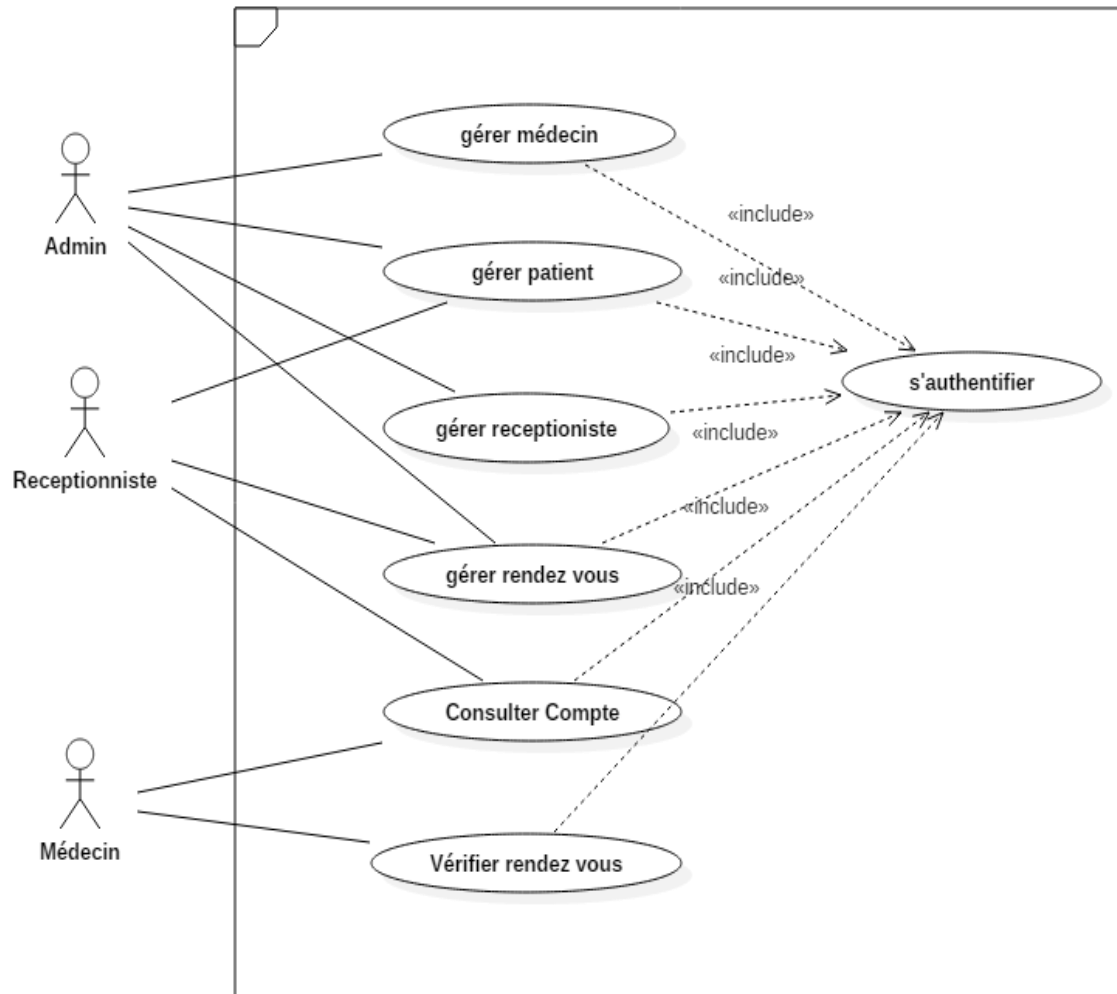
### 2.1 Introduction

Ce deuxième chapitre traite les étapes fondamentales pour le déroulement et le développement de notre système de gestion d'un établissement hospitalier.

Pour la conception et la réalisation de l'application, on a adopté de modéliser graphiquement et de construire un système fiable et stable

avec le formalisme UML (Unified Modeling Language), qui s'impose aujourd'hui comme le langage de modélisation objet standardisé pour la conception des logiciels.

### 2.2 Diagramme de cas d'utilisation :



Diagrammes des cas d'utilisation global

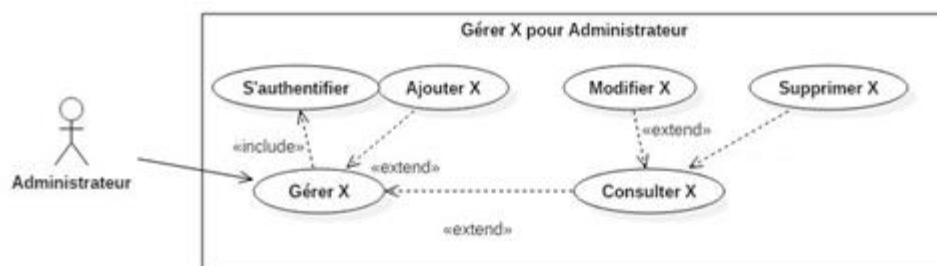


Diagramme de cas d'utilisation de gestion X pour l'administrateur

**Tableau 2.2: Description textuelle du cas d'utilisation « Ajouter patient »**

Cas d'utilisation	Ajouter un patient
Acteurs	Administrateur/Réceptionniste
Objectif	Il permet à l'administrateur ou le réceptionniste d'ajouter un patient
Précondition	Authentification. Succès d'authentification.
Postcondition	patient ajouté.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur/ le réceptionniste choisit l'ajout d'un nouvel patient,</li> <li>2. Le système affiche le formulaire à remplir,</li> <li>3. L'administrateur/réceptionniste saisi les coordonnées du nouvel utilisateur,</li> <li>4. Le système vérifie les données,</li> <li>5. Le système enregistre le patient dans la base des données.</li> </ol>
Scénario alternatif	<p>A1 : Champs obligatoires non valides ou vides.</p> <p>Cet enchaînement démarre au point 4.</p> <p>5. Le système affiche un message d'erreur.</p> <p>Le scénario reprend au point 2.</p>



Le tableau suivant permet de décrire le cas d'utilisation « S'authentifier » de point de vue acteurs, objectif, préconditions, post condition et scénarios :

**Tableau 2.2: Description textuelle du cas d'utilisation « S'authentifier »**

Cas d'utilisation	S'authentifier
Acteur	Administrateur et autres utilisateurs
Objectif	Il permet à l'acteur de l'identifier en saisissant son login et son mot de passe.
Précondition	L'acteur doit être présent dans la base de données.
Postcondition	Acteur Authentifié. La page d'accueil s'affiche
Scénario nominal	1. L'acteur ouvre l'application, 2. Le système affiche la page d'authentification, 3. L'acteur saisit le login et le mot de passe, 4. Le système vérifie l'existence des données, 5. Le système affiche la page d'accueil.
Scénario alternatif	A1 : Erreur d'authentification : Login ou mot de passe non valide Cet enchaînement démarre au point 4 5. Le système affiche un message d'erreur. Le scénario comprend au point 2. A2 : Champs obligatoires vides. Cet enchaînement démarre au point 4. 5. Le système affiche un message d'erreur. Le scénario reprend au point 2

**Tableau 2.3: Description textuelle du cas d'utilisation « Modifier patient »**

Cas d'utilisation	Modifier un patient
Acteurs	Administrateur/réceptionniste
Objectif	Il permet à l'administrateur/ réceptionniste de modifier un patient.
Précondition	Authentification. Succès de consultation de la liste des utilisateurs..
Postcondition	Utilisateur modifié.

## 2.3 Diagramme d'activité

Il donne une vision des enchaînements des activités propre à une opération ou à un cas d'utilisation.

Le diagramme d'activité est attaché à une catégorie de classes et décrit le déroulement des activités de cette catégorie. Le déroulement s'appelle "flot de contrôle". Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les conditions de séquence

Les figures suivantes présentent les diagrammes d'activités du cas d'utilisation.

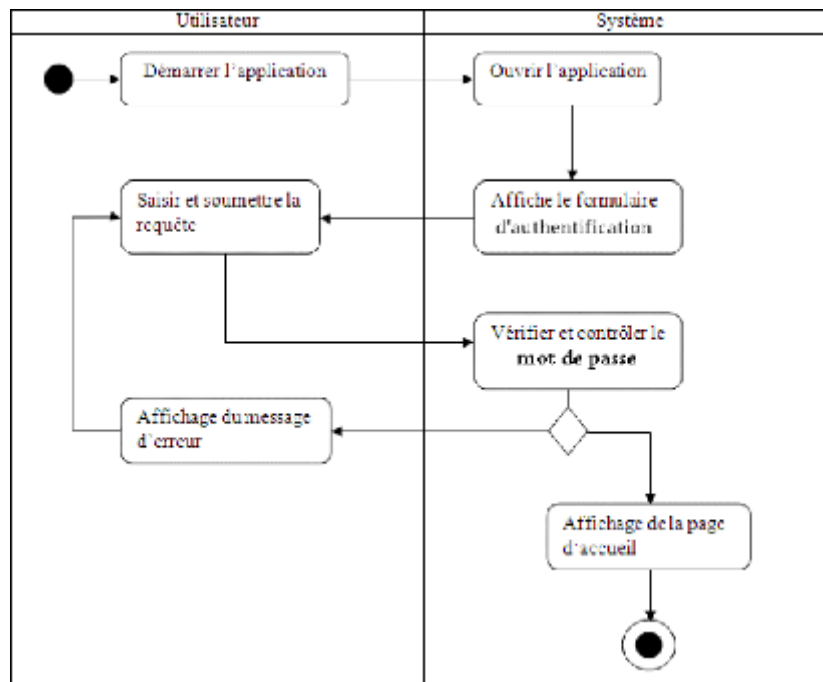


Diagramme d'activité de l'authentification

### Diagramme d'activité d'ajout

Après une demande d'ajout d'une donnée par l'utilisateur (administrateur, réceptionniste), le système lui affiche le formulaire d'ajout pour qu'il puisse saisir ces données et confirmer leur enregistrement au niveau de la base de données.

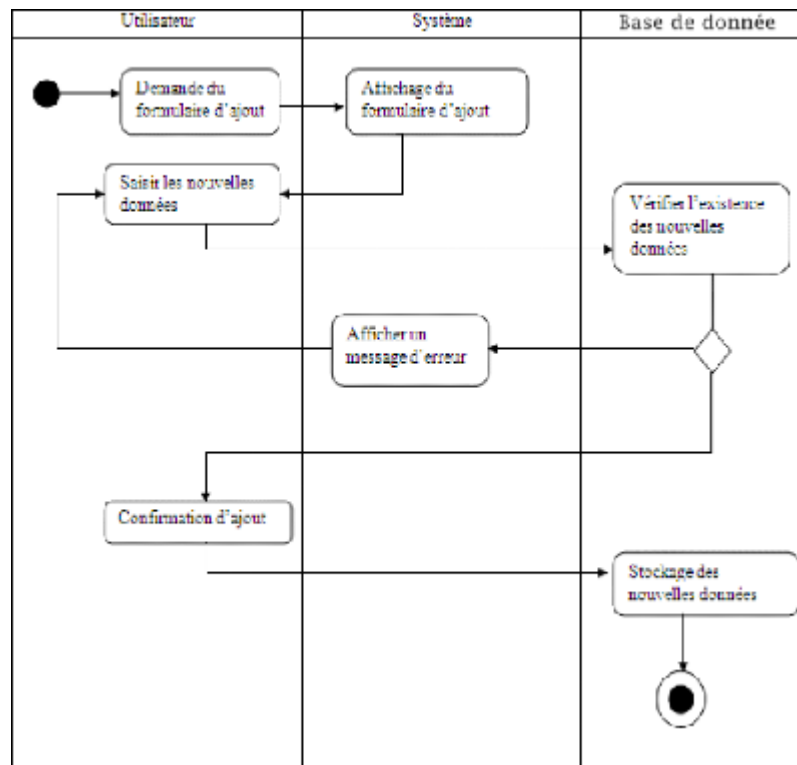
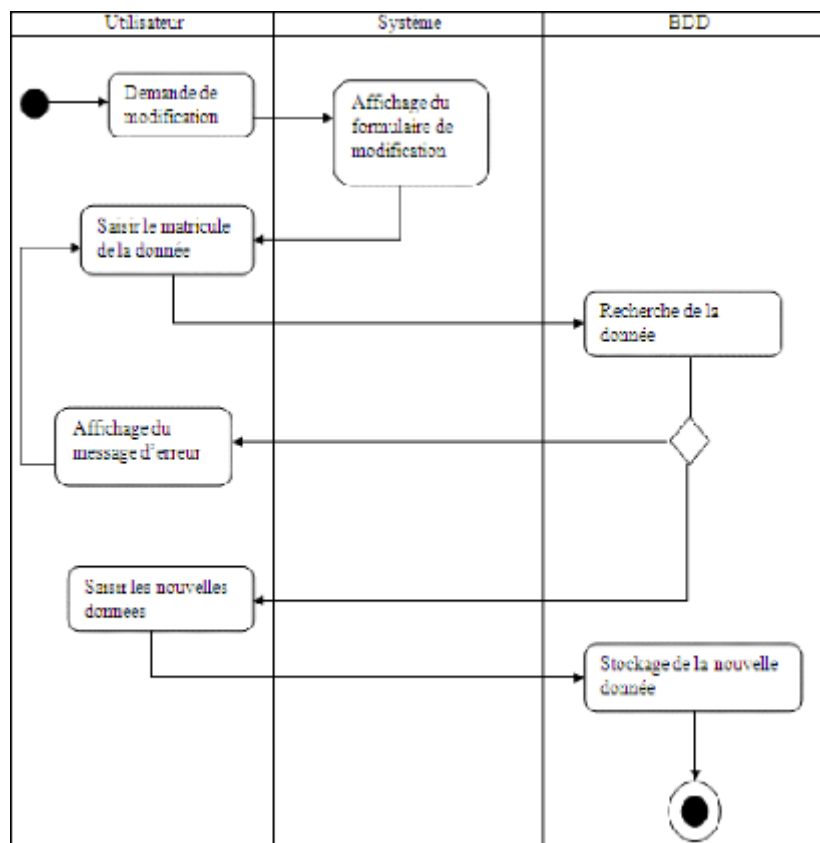
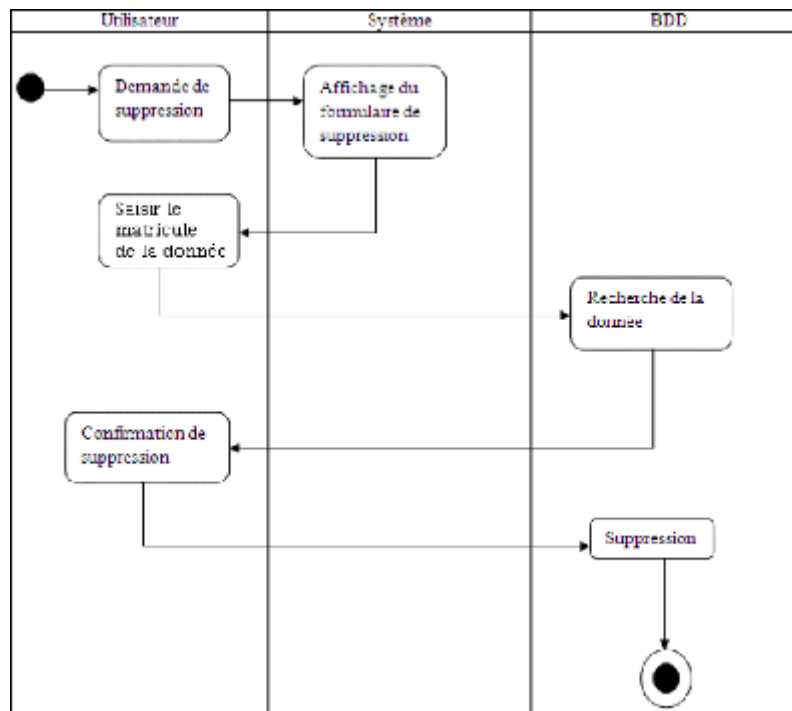


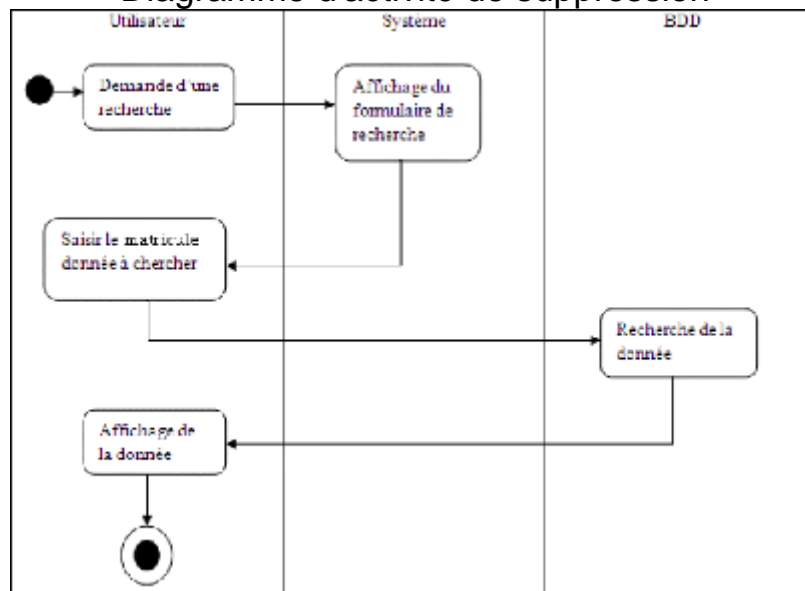
Diagramme d'activité d'ajout



## Diagramme d'activité de modification



## Diagramme d'activité de suppression



## Diagramme d'activité de recherche

## 2.4 Diagramme de classes

Notre diagramme de classes sert à identifier la structure des diverses classes comme les propriétés par exemple les attributs, les méthodes relatives à celles-ci ainsi que les relations existantes entre-elles comme l'héritage et les différentes associations au sein du système courant.

La figure suivante présente de diagramme de classes pour cette application.

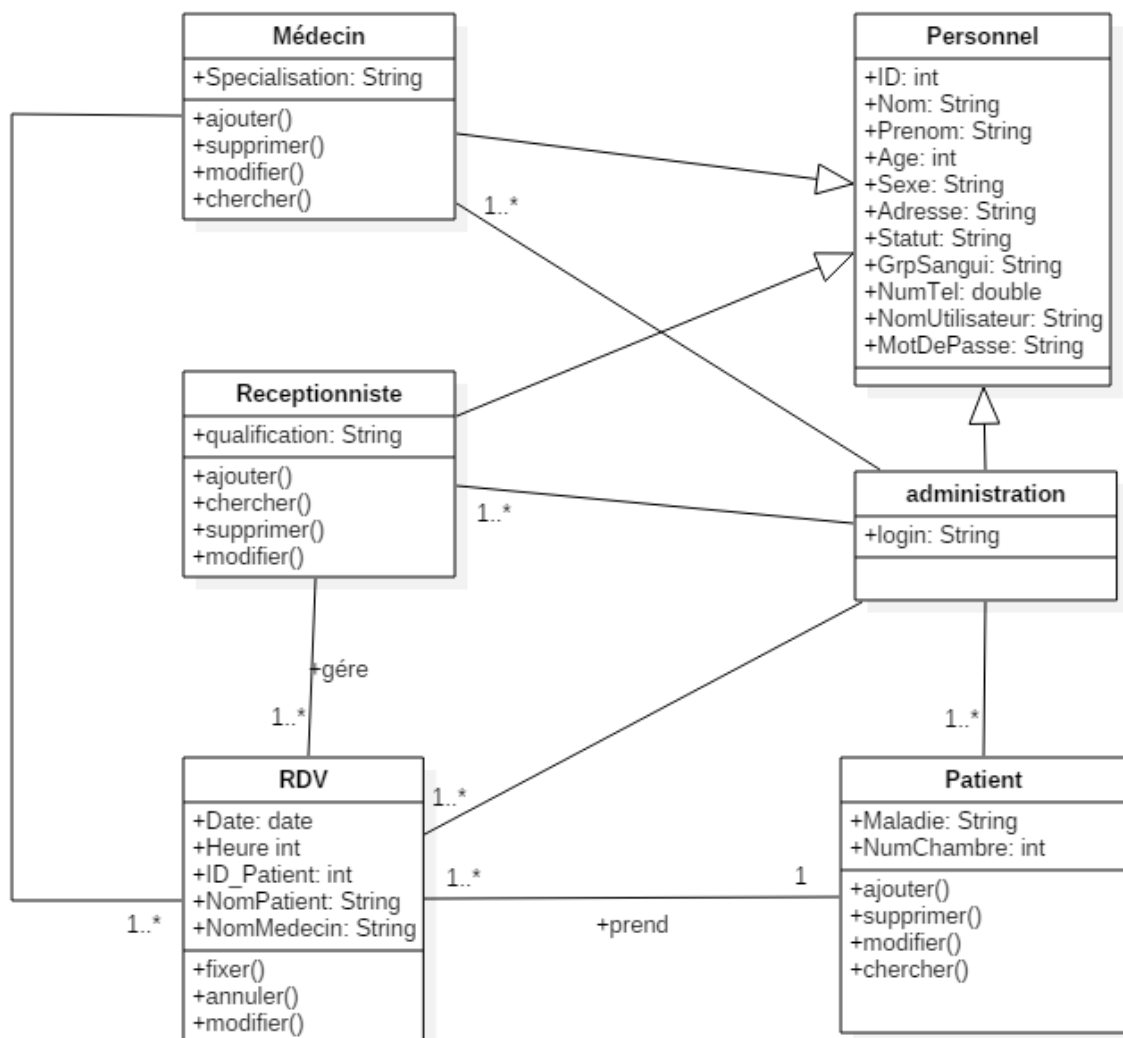


Diagramme de classes

## **Conclusion:**

Ce chapitre a été consacré pour la spécification des besoins fonctionnels du système résultant, ce qui correspondait aux différentes activités de la première phase du cycle de développement du notre système.

## Réalisation de Système

---

### 3.1 Introduction :

on va entamer dans ce chapitre la partie réalisation et implémentation dans laquelle on s'assure que le système est prêt pour être exploité par les utilisateurs finaux. A la fin de ce chapitre, les objectifs doivent avoir été atteints et le projet doit être clos.

### 3.2 Environnements de travail

Dans cette partie, on présente l'environnement technique relatif à la réalisation de l'application.

#### 3.2.1 Environnement logiciel

Les différents outils que j'ai utilisés pour développer l'application sont les suivants :

**Langage :** Java

Pour la construction des interfaces graphiques on a utilisé java GUI frameworks (Swing/AWT).

**SGBD :**

La principale mission est bien entendu d'effectuer la liaison entre MySQL et mon projet Java. Ce qu'on souhaite, c'est interagir avec la base depuis le code de l'application.



Pour faire la connexion entre une application Java et MySQL, on a usité un remède standard s'intitule JDBC qui est une API

La figure ci-dessous montre les liens entre Java et MySQL à travers les API et Les JAR.

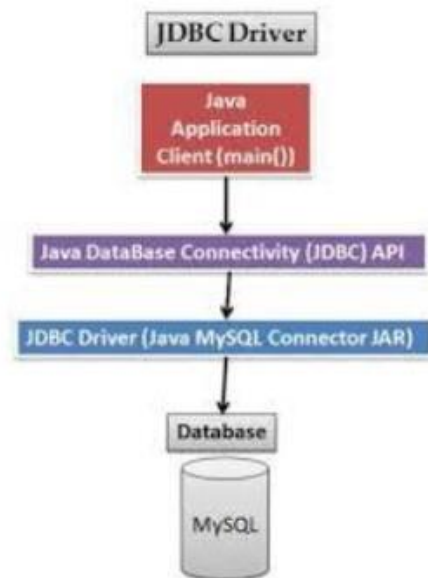


Figure: Liaison entre application Java et MySQL

### Logiciel :

L'environnement qu'on a adopté pour améliorer notre application est Netbeans IDE 8.2

Photoshop : pour le design des interfaces de l'application.

#### • WampServer :

Pour gérer les données, on a pris le choix d'utiliser WampServer qui est une plateforme web de type WAMP.

WampServer est un environnement qui contient deux serveurs qui sont Apache et MySQL, c'est un interpréteur de script PHP ainsi que phpMyAdmin pour l'administration web des bases MySQL.

La figure ci-dessous montre un imprime écran du menu de WampServer.



Figure :Menu de WampServer

Dans l'environnement du WampServer, on va utiliser que la partie phpMyAdmin pour administrer les bases des données MySQL.

Il existe plusieurs façons d'accéder à sa base de données et d'y faire des modifications. On peut utiliser une ligne de commande, exécuter des requêtes SQL ou faire appel à un programme qui permet d'avoir une vue ensemble. Ici on va découvrir phpMyAdmin, un des outils les plus connus permettant de manipuler une base des données MySQL.

PhpMyAdmin est l'une de plus célèbre interfaces pour gérer une base des données MySQL. Cette interface permet d'exécuter des connaissances dans le domaine des bases des données, des nombreuses requêtes comme les créations de table des données, les insertions, les mises à jour, les suppressions les modifications de structure de la base des données.

Ce système est très pratique pour sauvegarder une base des données sous forme de fichier .sql et ainsi transférer facilement ses données.

De plus celui-ci accepte la formulation de requête SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

## **Justification du choix d'utilisation de la langue anglaise au niveau des interfaces :**

Maitriser la langue anglaise devient indispensable pour être un développeur :

- on trouve énormément de livres et de ressources informatiques en anglais, qui n'ont pas leurs équivalents en français
- on trouve plus rapidement les solutions aux problèmes et en plus elles seront souvent de bien meilleure qualité.
- Faciliter aux utilisateurs étrangers l'utilisation de l'application puisque l'anglais est la langue commune à l'international.

### **3.3 Réalisation :**

La conception des interfaces de l'application est une étape très importante puisque toutes les interactions avec le cœur de l'application passent à travers ces interfaces, on doit alors guider l'utilisateur avec les messages d'erreurs et de notification si besoin, ainsi présenter un système complet. Dans cette partie, nous allons présenter quelques interfaces de l'application, répondant aux recommandations ergonomiques de compatibilité, de guidage, de clarté, d'homogénéité et de souplesse.

(je vais exposer quelques scénarios d'exécution à travers des captures d'écran.)

La figure suivante représente la page d'accueil de notre application :

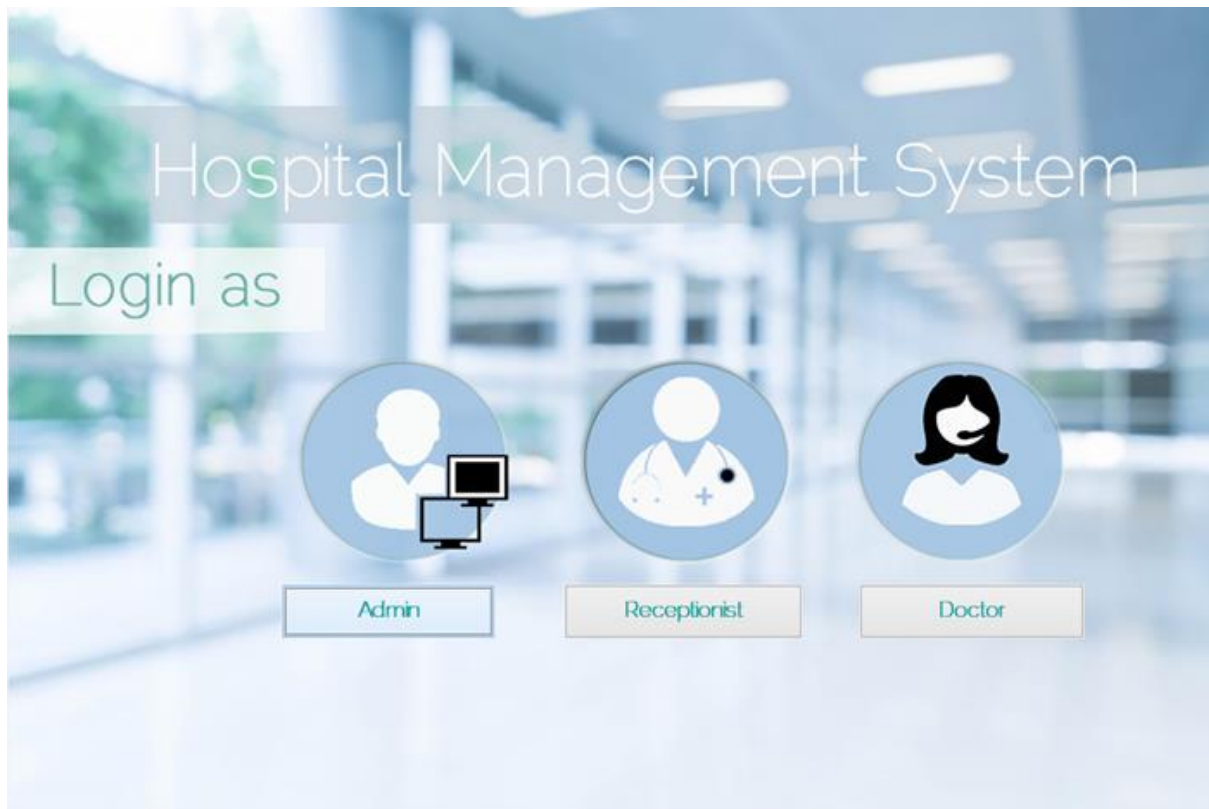


Figure 3.1: Interface "Page d'accueil"

À travers cette interface, l'administrateur, le réceptionniste et le médecin peuvent accéder à leurs interfaces d'authentification afin de manipuler les fonctionnalités de l'application.

La figure suivante représente l'interface d'authentification pour l'utilisateur

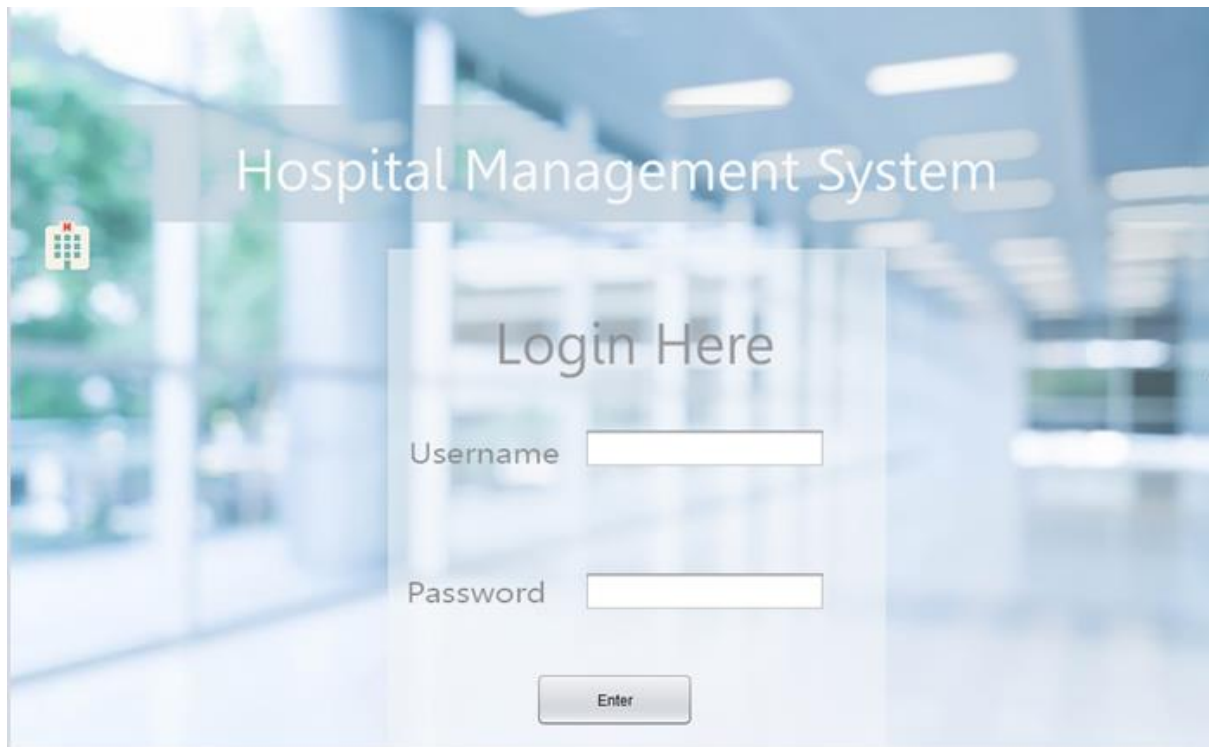
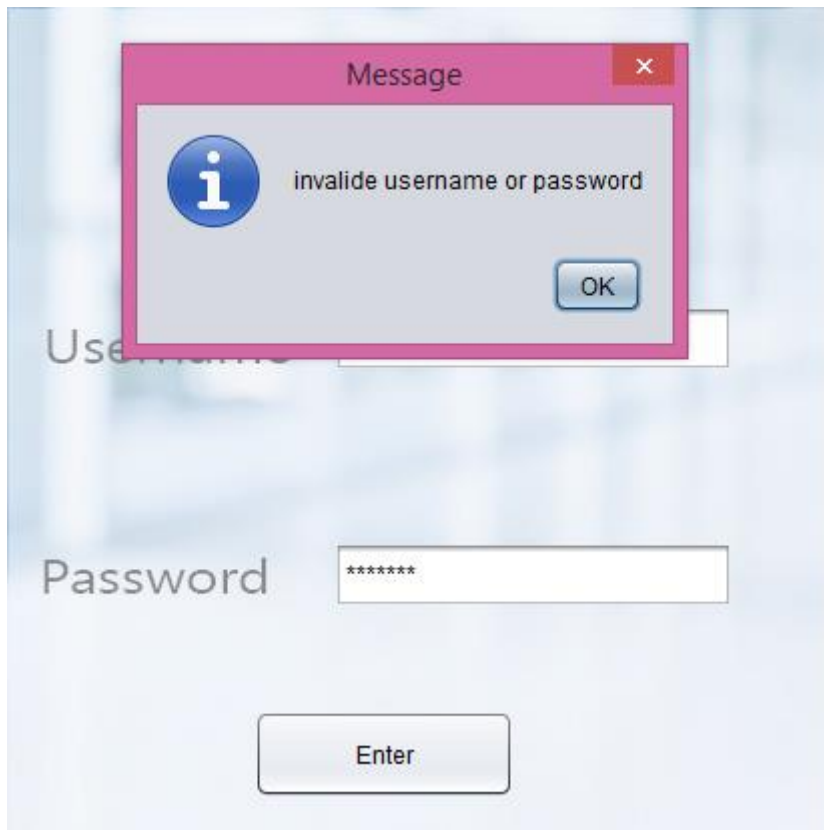


Figure 3.2 : Interface "Authentification"



Cette icône permet le retour à la page d'accueil



En cas d'un nom d'utilisateur ou d'un mot de passe invalide, un message d'erreur apparaît.

Nous avons choisi l'administrateur comme utilisateur vu qu'il présente à travers ces interactions la majeure partie des principales fonctionnalités de l'application

Lors de l'authentification, il passe à une interface de portail d'administrateur :

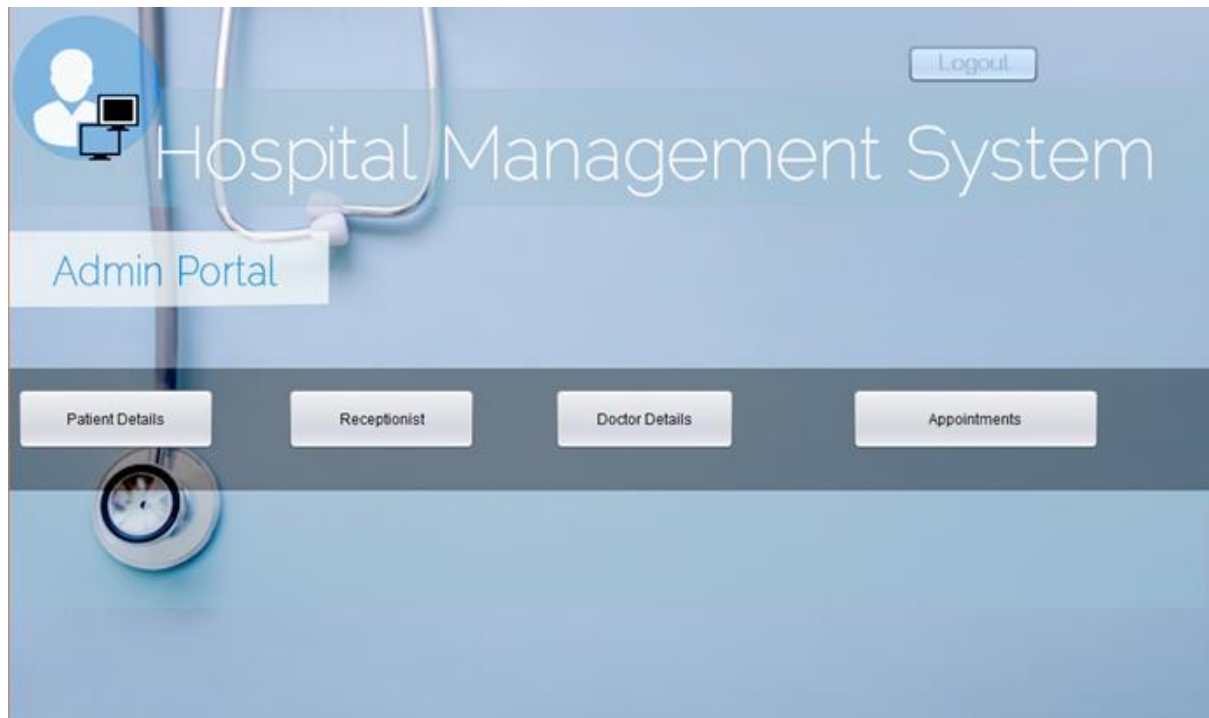


Figure 3.3 : Interface "Admin Portal"

Comme on a dit AU NIVEAU DE chapitre 2 , l'administrateur peut gérer les patients , les réceptionnistes , les médecins et les rendez-vous

Dans ce qui suit, nous allons détailler les scénarios de la gestion des patients sous forme des captures d'écran.

L'administrateur peut ajouter, modifier supprimer, chercher un patient, de plus il peut voir la liste des patients ajoutés.

The screenshot displays a web application interface for managing patients. At the top, there is a navigation bar with buttons: "Add Patient", "Delete Patient", "Update Patient", "Search Patient", and "View Patient". Below this, the main form area is divided into two columns. The left column contains fields for "Patient ID :", "Name :", "Age :", "Gender :" (with a dropdown menu showing "Male"), "Adresse :", and "Status". The right column contains fields for "Second Na...", "Disease", "Blood Group :" (with a dropdown menu showing "A+"), and "Room Numb...". There are two buttons in the top right: "Add Patient!" and "Logout". On the right side of the form, there is a cartoon illustration of a person with dark hair, a fever (indicated by a red thermometer in their mouth and sweat drops on their forehead), and a blue shirt.

Figure 3.4 : Interface "gérer patient"

Dans cette interface on a utilisé une « JTappedPane » pour faciliter l'utilisation des opérations et pour améliorer la visibilité.

This is a close-up view of the navigation bar from the previous screenshot. It shows a horizontal row of five buttons: "Add Patient", "Delete Patient", "Update Patient", "Search Patient", and "View Patient". The buttons are light blue with dark text and are separated by small gaps.

L'administrateur a le droit d'ajouter un nouveau patient. Il doit remplir le formulaire et valide sa saisie.



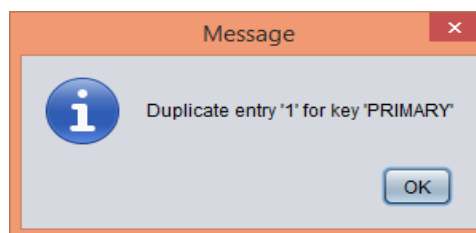
The screenshot shows a web application window titled "Ajouter un patient". At the top, there are five tabs: "Add Patient", "Delete Patient", "Update Patient", "Search Patient", and "View Patient". The "Add Patient" tab is active. The form contains the following fields and values:

- Patient ID : 1
- Name : Soumaya
- Age : 29
- Gender : Female (dropdown menu)
- Adresse : Bizerte
- Status : Single
- Second Na... : Atia
- Disease : Tuberculosis
- Blood Group : B+ (dropdown menu)
- Room Numb... : 102

Buttons include "Add Patient!", "Logout", and "OK" in the message dialog. A cartoon character of a person with a fever is on the right side of the form.

Figure 3.5 : Interface "Ajouter un patient"

En cas erreur de saisie (duplication de clé primaire par exemple) ou d'absence d'information , des messages d'erreurs apparaissent pour alerter l'administrateur.



Pour voir les patients ajoutées :

Add Patient Delete Patient Update Patient Search Patient View Patient									
ID	Name	Second Name	Age	Desiese	Gender	Blood Group	Adresse	Status	Room Number
01	Soumaya	Atia	29	Tuberculosis	Female	B+	Bizerte	Single	102
02	Samer	Aziz	33	CVA	Male	O-	Mahdia	Divorced	110
03	Aya	Zouay	20	Flu	Female	A+	Monastir	Single	309
04	James	King	24	Chlorea	Male	A+	heart	Single	555
05	Sarra	Mouradi	15	Influenza	Male	AB+	Beja	Single	341

Add Patient
Delete Patient
Update Patient
Search Patient
View Patient

Patient ID :

Name : 
Second Na... :

Age : 
Dicese :

Gender :

Adresse :

Status : 
Room Numb... :

update
are you sure you want to update patient record?

Figure 3.6 : Interface « Modifier patient »

À travers la figure Précédente, l'administrateur peut modifier un patient.

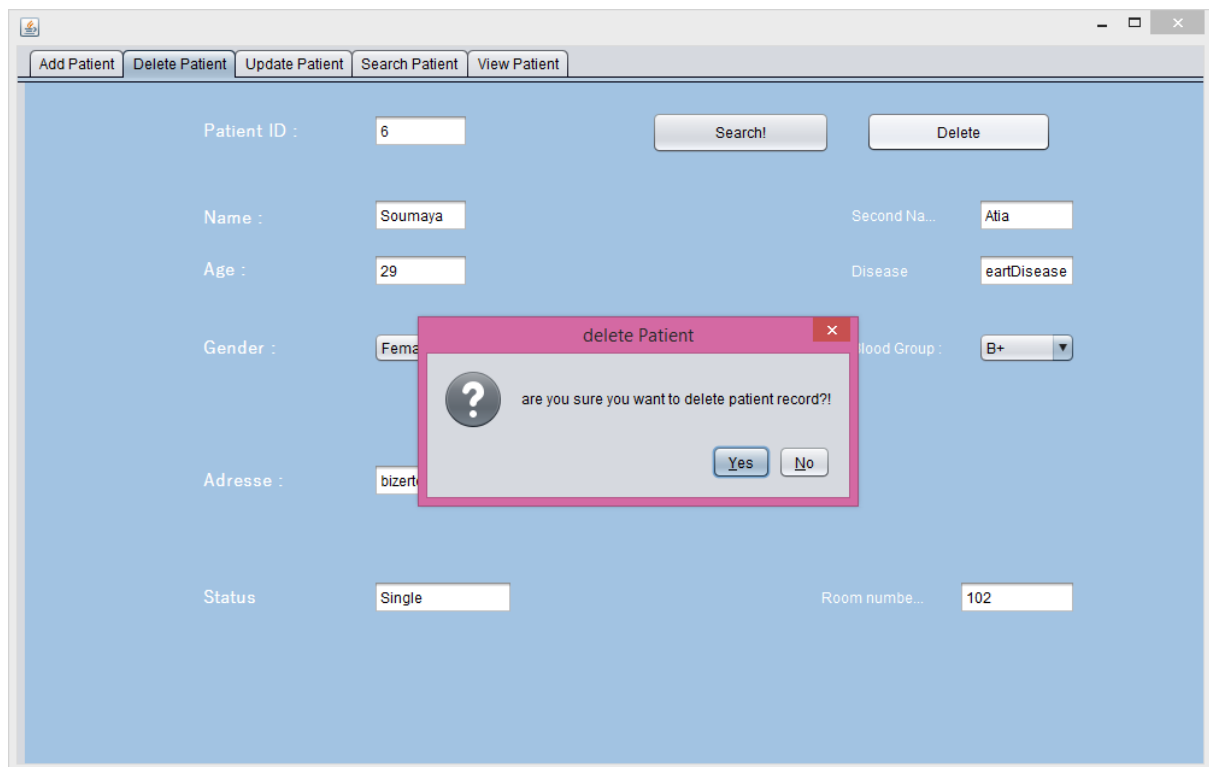


Figure 3.7 : Interface « Supprimer patient »

Afin de modifier ou supprimer un patient l'administrateur doit entrer l'ID de ce dernier. En plus l'identifiant doit être déjà existant, sinon un message d'erreur apparait.

Comme les patients, l'administrateur peut également gérer les médecins, les réceptionnistes et les rendez-vous :

Add Doctor
Delete Doctor
Update Doctor
Search Doctor
View Doctor

Doctor ID :

Name : 
Second Name :

Age : 
Specialization :

Gender : 
Blood Group :

Status 
Phone Number :

Adresse :

username

password




Figure 3.8 : Interface « Gérer Médecin »

Add Receptionist
Delete Receptionist
Update Receptionist
Search Receptionist
View Receptionist

Receptionist ID :

Name : 
Second Name :

Age : 
Qualification :

Gender : 
Blood Group :

Adresse :

Status 
Phone Num...

username

password

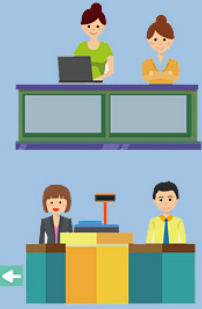
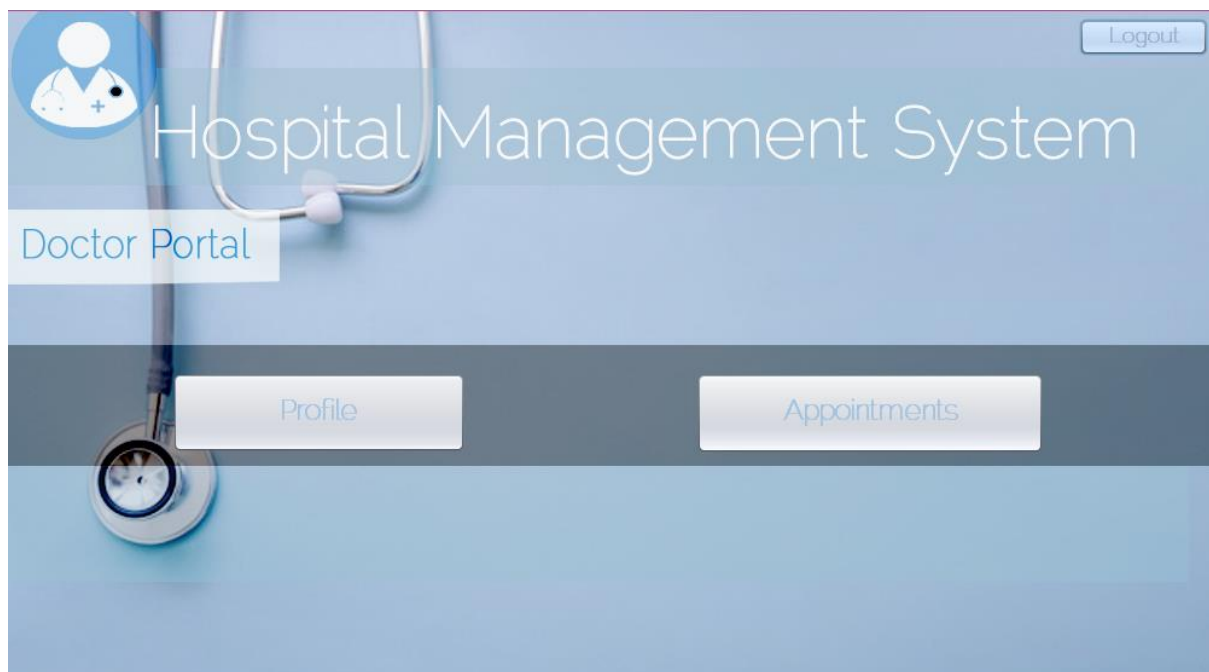


Figure 3.9 : Interface « Gérer Réceptionniste »

The interface for managing appointments includes a top navigation bar with three tabs: "Add Appointment", "Edit Appointment", and "View Appointments". Below the tabs, there are five input fields arranged in two rows: "Patient ID", "Patient name", "Date", "Doctor name", and "Time". An "Add Appointment!" button is positioned to the right of the "Patient ID" field. On the right side of the interface, there is an illustration of a hand holding a blue pen, marking a date on a calendar with a red circle.

**Figure 3.10 :** Interface « Gérer RDV »

Si un médecin de l'hôpital veut utiliser l'application , il doit s'authentifier en cliquant sur « Doctor » et entrer son username et son password. La portail de medecin est la suivante :



**Figure 3.11:** Interface « Doctor Portal »

Le médecin peut voir son profil en entrant son identifiant et il peut éditer les rendez-vous.

De même pour le réceptionniste qui peut voir son profil, gérer les patients et gérer les rendez-vous.

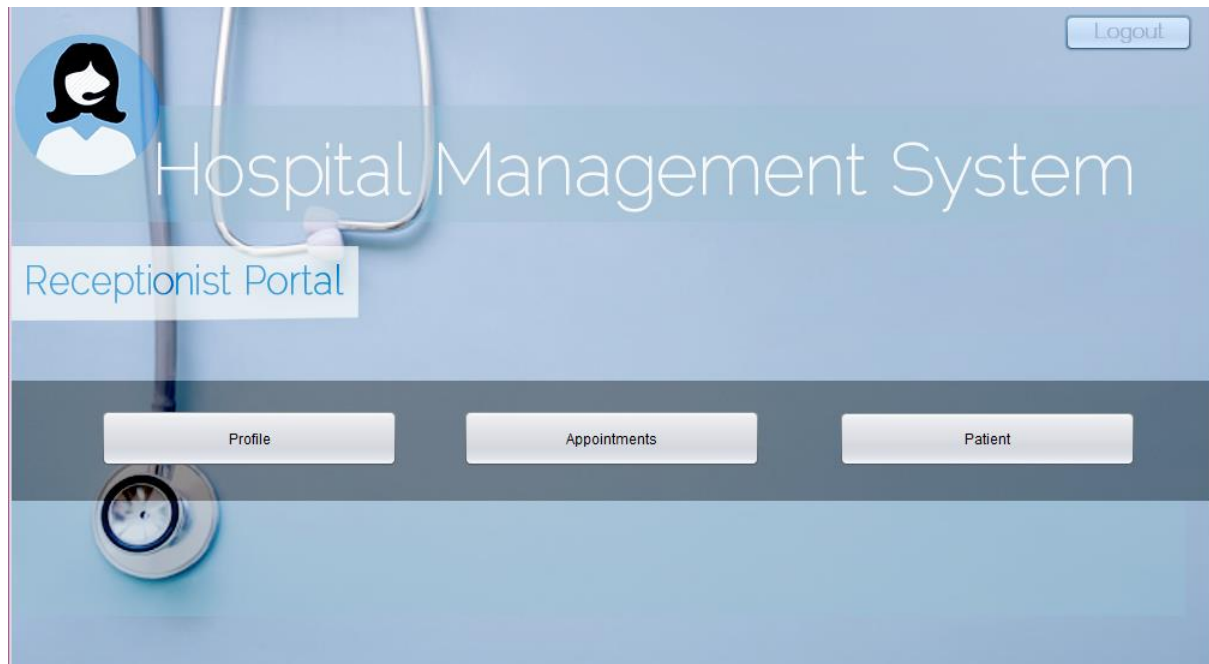


Figure 3.1é: Interface « Receptionist Portal»

## Conclusion :

A travers ce chapitre, on a présenté la réalisation de l'application, en représentant quelques interfaces graphiques qu'on a jugé les plus importantes et en décrivant brièvement comment on a planifié notre projet.

## Conclusion générale

L'objectif de mon projet était de concevoir et implémenter un système de gestion d'un hôpital. Le point de départ de la réalisation de ce projet était une récolte des informations nécessaires pour connaître les besoins d'un système d'établissement hospitalier et les problèmes des hôpitaux qui disposent d'un système de fonctionnement manuel. Par la suite, j'ai m'intéressé à l'analyse et la spécification des besoins qui m'a permis de distinguer les différents acteurs interagissant avec l'application visée. L'objectif de la partie suivante était la conception détaillée, dans laquelle j'ai fixé la structure globale de l'application. Le dernier volet de mon projet était la partie réalisation qui a été consacré à la présentation des interfaces les plus significatives de cette application.