



---

**Développement d'application de Gestion de Cabinet Médical GCM**

---



**Réalisé par :**

**Ayadi Khouloud**

**Salhi Roufaida**

**Ferchichi Afef**

**Zammel Manel**

**1DNI3**

## Table de matière

<b>INTRODUCTION .....</b>	<b>2</b>
<b>Chapitre 1: Contexte de travail.....</b>	<b>3</b>
<b>I. Introduction .....</b>	<b>4</b>
<b>II. L'idée émergente du projet : .....</b>	<b>4</b>
<b>III. Problématique : .....</b>	<b>4</b>
<b>IV. Objectifs et travail demandé : .....</b>	<b>4</b>
1. Travail demandé: .....	4
2. Les objectifs : .....	5
<b>V. Conclusion :.....</b>	<b>6</b>
<b>Chapitre 2 : Etat de l'art .....</b>	<b>7</b>
<b>I. Introduction :.....</b>	<b>8</b>
<b>II. Etude de l'existant :.....</b>	<b>8</b>
<b>III. Critique de l'existant : .....</b>	<b>8</b>
<b>IV. Solution proposée : .....</b>	<b>9</b>
<b>V. Conclusion :.....</b>	<b>9</b>
<b>Chapitre 3 : Etude conceptuelle .....</b>	<b>10</b>
<b>I. Introduction :.....</b>	<b>11</b>
<b>II. Choix du langage de conception :.....</b>	<b>11</b>
1. Le Choix d'UML .....	11
2. Les points forts d'UML : .....	11
3. Les points faibles d'UML : .....	12
<b>III. Partie Conception :.....</b>	<b>12</b>
1. Diagramme de cas d'utilisation : .....	12
2. Diagramme de cas d'utilisation Global :.....	13
3. Description détaillée des principaux cas d'utilisations :.....	14
4. Diagramme de classe : .....	18
5. Schéma Relationnel de la base des données : .....	19
<b>Chapitre 4 : Partie Réalisation.....</b>	<b>20</b>
<b>I. Introduction :.....</b>	<b>21</b>
<b>II. Environnement de développement : .....</b>	<b>21</b>
<b>III. Les interfaces graphiques de l'application GCM :.....</b>	<b>22</b>
<b>IV. Conclusion.....</b>	<b>27</b>
<b>Conclusion et perspectives .....</b>	<b>28</b>

# INTRODUCTION

De nos jours, la technologie informatique est appliquée dans plusieurs domaines. L'un est la santé. En effet, une médecine moderne est de qualité nécessite un traitement rationnel et sûr de l'information médicale. En effet, la complexité croissante de la médecine actuelle pousse de manière naturelle à la mise en place des applications bureautiques robustes étant capables d'aider le médecin dans ses tâches quotidiennes.

Grâce au développement des logiciels et des réseaux, les applications bureautiques seront capables de satisfaire un large éventail de besoins notamment dans le domaine de la santé.

Dans ce cadre, on a choisi le cabinet médical comme établissement traité dans notre sujet, dont l'informatique joue un rôle important. En effet, la croissance de la population hospitalière nécessite la mise en place d'une gestion rationnelle prise et rapide, or et jusqu'à ce jour, la manière de gérer manuellement est encore dominante d'où la nécessité d'introduire l'informatique dans les administrations hospitalières.

C'est dans cette optique se situe notre projet qui consiste à faire la conception et la réalisation d'une application bureautique de gestion de cabinet médical. Cette application représente une solution aux différents problèmes rencontrés par les médecins.

Notre rapport de projet de gestion de cabinet médical est scindé en quatre chapitres :

Au niveau du premier chapitre, nous introduisons le contexte de travail de L'application. Le deuxième chapitre présente un état de l'art. Le troisième chapitre, montre les différents diagrammes UML utilisés pour l'analyse et la conception de notre application. Dans le quatrième chapitre nous décrivons l'environnement de développement matériel et logiciel et nous détaillons la méthodologie de travail et les différentes étapes de la mise en place de ce projet.

Finalement, nous clôturons ce rapport par une conclusion générale du travail accompli suivi par des perspectives de notre projet.

# Chapitre 1: Contexte de travail



## **I. Introduction**

Ce chapitre présente une description détaillée de notre projet. En fait on a mis en évidence les différentes fonctionnalités de l'application ainsi que ses objectifs.

## **II. L'idée émergente du projet :**

L'idée principale de notre projet est de créer une application bureautique de gestion de cabinet médical GCM qui offre aux médecins plusieurs services tels que la gestion des patients, la gestion des consultations, la gestion des RDV, la gestion des factures, afin de faciliter les tâches qui se traitent manuellement.

## **III. Problématique :**

Les établissements médicaux sont des organismes un peu complexes associent plusieurs types d'unités et gèrent plusieurs services et des activités hospitalières ce qui provoque plusieurs difficultés et des problèmes pour les médecins (la perte de temps, difficulté de stockage, risque de perte d'archives). Donc nous nous proposons de faciliter les tâches grâce à notre application qui peut fournir aux médecins la gestion de ses patients en fait il a la possibilité de lister et chercher à un patient quelconque. De plus il peut gérer ses RDV, les consultations (les observations et les prescriptions médicales) et les factures.

## **IV. Objectifs et travail demandé :**

### **1. Travail demandé:**

Le projet consiste à développer une application bureautique destinée pour les médecins qui ont des problèmes de gestion de travail au sein d'un cabinet médical. Notre application "gestion de cabinet médical (GCM)" permet de résoudre ces problèmes dans un temps réduit et avec une manière très facile, on vise les tâches suivantes à atteindre à travers la réalisation de notre application :

- **S'authentifier :** Le médecin doit s'authentifier afin d'assurer la sécurité par le saisi de son nom utilisateur (login) et son mot de passe (password) avant l'accès à l'application.
- **La gestion des patients :** Dans ce cas le médecin peut ajouter, supprimer un patient bien déterminé et rechercher ou modifier ses données.
- **La gestion des consultations :** Dans cette tâche le médecin peut ajouter des nouvelles consultations dans lesquelles il peut noter ses observations et son prescription médicale pour un patient bien déterminé, aussi il peut supprimer ou modifier les données d'une consultation.
- **La gestion des RDV :** Dans ce cas le médecin peut modifier, supprimer ou bien ajouter des nouveaux RDV demandé par un patient
- **La gestion des factures :** notre application permet aux médecins d'effectuer cette tâche où ils peuvent créer une nouvelle facture, supprimer ou bien modifier une facture existante.

## 2. Les objectifs :

Notre projet a pour objectif d'améliorer le fonctionnement et faciliter l'admission au sein d'un cabinet médical afin d'y remédier à tous ces problèmes on peut assigner à notre étude les objectifs suivants :

- **Gain de temps :** Notre application permet aux médecins d'effectuer les tâches quotidiennes pénibles qui se traitent manuellement et qui peuvent prendre beaucoup du temps.
- **Rapidité dans l'établissement de travail :** Notre projet offre aux médecins un espace de sauvegarde de leurs travaux et leurs notes sans la nécessité d'utiliser les ressources manuelles.
- **Sécurité des données :** Notre travail propose une bonne sécurité des données personnelles car l'utilisateur doit s'authentifier avant l'accès à l'application.
- **Automatiser les taches qui se traitent manuellement :** Notre projet permet l'utilisateur de résoudre les problèmes de traitement de façon manuelle comme les

problèmes d'archivage car cette technique entraîne parfois la perte des données importantes ou bien la duplication des dossiers médicaux.

- **Proposer une bonne sauvegarde :** Grâce à notre application les nouvelles et les anciennes données restent toujours sauvegardées qu'on peut les utiliser en cas de besoin.

## **V. Conclusion :**

Nous avons détaillé dans ce chapitre les fonctionnalités et les objectifs de notre application. Dans le prochain chapitre intitulé « Etude préalable » nous allons faire une étude de l'existant

## **Chapitre 2 : Etat de l'art**





## **I. Introduction :**

Ce chapitre présente un état de l'art : il s'agit d'une étude de l'existant suivie de critique permettant au projet de présenter une amélioration résumant l'ensemble des solutions retenues.

## **II. Etude de l'existant :**

Dans cette section, on mène une étude basée sur les observations de différentes applications similaires à l'application qu'on est demandé de réaliser. Dans le but de résoudre les problèmes cités précédemment, plusieurs outils ont été créés à base des nouvelles technologies. Parmi lesquels nous pouvons citer : Tabib qui présente une application web permet à l'utilisateur de prendre des RDV et la gestion des patients ; GRATIN-GCM se présente aussi une application web permettant de gérer les consultations et les ordonnances et les certificats des patients. Ce qui concerne les applications mobiles on trouve ImediaSante qui a comme fonctionnalités la prise des rendez-vous, l'appel des urgences en cas de secours ; Diseases Dictionary Medical qui représente un manuel hors ligne qui comporte presque tous les troubles médicaux et les maladies par définitions, symptômes, origines et qui donne des informations sur les traitements associés.

## **III. Critique de l'existant :**

Cette partie a pour but de dégager les insuffisances et les défaillances du système actuel relatif à la gestion d'un cabinet médical. En effet, certaines applications ont des fonctionnalités limitées par exemple ImediaSante ne contient pas la fonctionnalité d'annulation et de changement des RDV.

### **Pour La méthode manuelle :**

- Les travaux manuels sont élevés, lourds et pénibles qui se présente d'une façon répétitive à savoir l'archivage et la consultation des fiches des patients.
- Absence d'un moyen de recherche rapide. En effet la recherche des fiches des patients par exemple se fait d'une manière manuelle fiche par fiche ce qui entraîne beaucoup de problèmes tel que la perte de temps et la perte des fiches.
- La gestion des RDV, se fait d'une manière manuelle ce qui engendre un risque d'oubli ou chevauchement des RDV.

- La perte de temps qui est remarquable en cas d'augmentation du nombre des patients pour la consultation.
- La gestion des documents administratives tout à la longueur de la journée qui sont : les prescriptions médicales.

#### **IV. Solution proposée :**

Après avoir déterminé les insuffisances et les inconvénients du système actuel, on propose une approche de solution qui consiste à concevoir et à développer une application qui évitera les fautes les défaillances énumérés précédemment. On propose alors de programmer une nouvelle application bureautique permettant l'organisation et l'automatisation des tâches.

#### **V. Conclusion :**

Ce chapitre a été consacré pour l'étude de l'existant. Dans le chapitre suivant, nous étudierons la phase de conception.

## **Chapitre 3 : Etude conceptuelle**



# I. Introduction :

Ce chapitre s'intéresse à la partie conceptuelle qui constitue une phase primordiale dans le processus de développement vu l'importance des décisions prises et leurs impacts sur la qualité et la performance de la solution produite. Nous allons commencer par une description de l'outil de modélisation utilisé. Ensuite, nous allons présenter le digramme de cas d'utilisation et les digrammes de classes de notre système. Enfin, nous allons présenter l'aspect dynamique de notre système par des diagrammes de séquences.

## II. Choix du langage de conception :

### 1. Le Choix d'UML

Par rapport à toutes les méthodes orientées objets qui sont en utilisation, seule UML a la capacité de satisfaire tous les besoins de conception requises par les entreprises et les sociétés informatiques. En effet, il unifie les notations nécessaires aux différentes activités d'un processus de développement et offre en plus les moyens d'établir le suivi des décisions prises depuis les spécifications jusqu'au codage.

#### ➤ Notion d'UML :

UML (Unified Modeling Language : « langage de modélisation unifié ») est un langage



graphique de modélisation orienté objet des données et des traitements développés, en réponse à l'appel de proposition lancée par OMG (Object Management Group) dont le but de définir une méthode standard pour la modélisation des applications construites à l'aide d'objet.

### 2. Les points forts d'UML :

✚ UML est un langage formel et normalisé :

- ✓ Il permet le gain de précision,
- ✓ Encourage l'utilisation d'outils
- ✓ Gage de stabilité.

🚦 UML est un support de communication performant :

- ✓ Il cadre l'analyse
- ✓ Facilite la compréhension de représentations abstraites complexes.
- ✓ Son caractère polyvalent et sa souplesse en font un langage universel.

### **3. Les points faibles d'UML :**

- ✓ La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- ✓ UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.
- ✓ Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.

## **III. Partie Conception :**

### **1. Diagramme de cas d'utilisation :**

#### **i. Définition :**

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet. Ils sont plus appropriés pour le développement de cas d'utilisation. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur et un système.

#### **ii. Description des acteurs :**

Au niveau de cette section, nous présentons l'acteur susceptible d'interagir avec le système, mais avant cela, nous donnons une définition du concept "acteur".

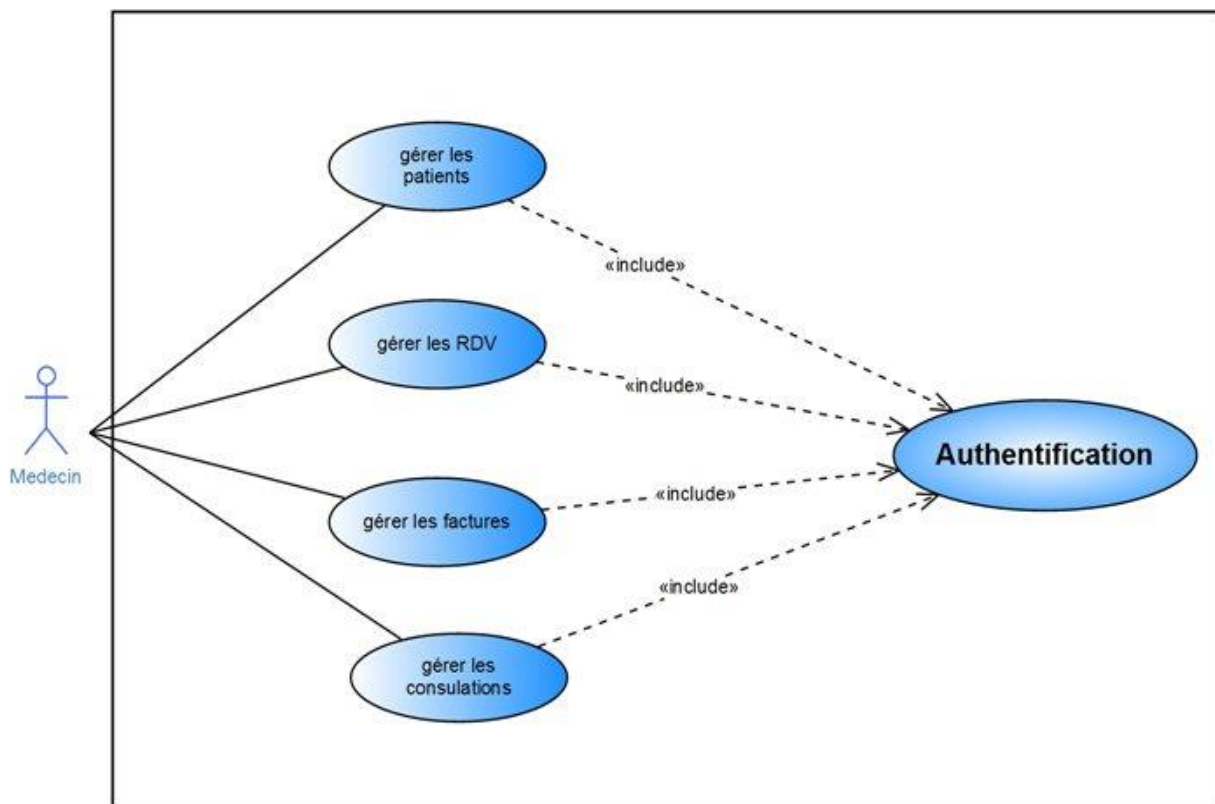
Un acteur : représente l'abstraction d'un rôle joué par des entités externes qui interagissent directement avec le système étudié.

Dans cette partie nous allons identifier les différents acteurs qui contribuent au déroulement de notre application.

Acteur	Description
<b>Médecin</b>	<ul style="list-style-type: none"> <li>▪ Gérer les consultations</li> <li>▪ Gérer les RDV</li> <li>▪ Gérer les factures</li> <li>▪ Gérer les patients</li> </ul>

Tableau: Description des acteurs

## 2. Diagramme de cas d'utilisation Global :



➡ Cette figure représente le diagramme du cas d'utilisation général. D'ailleurs, nous avons présenté le principal acteur et leurs rôles, de plus nous avons remarqué également les relations qui relient les différents cas d'utilisation qui indiquent que le cas d'utilisation devrait être accompli en amont.

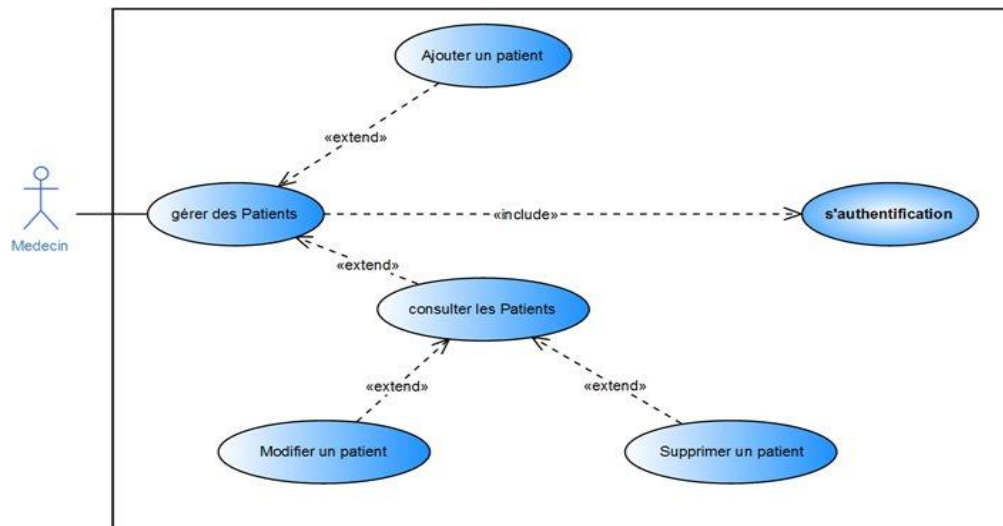
### 3. Description détaillée des principaux cas d'utilisations :

Un cas d'utilisation est un moyen de collecter et décrire les besoins des acteurs du système.

C'est une représentation de fonctionnalité du système suivant les points de vue des utilisateurs.

Pour le cas de notre application, ce diagramme contient :

#### i. Cas d'utilisation de gestion des patients :



➔ Description textuelle :

✚ **Nom de cas :** gérer les patients

✚ **But :** Le médecin peut gérer les patients en effectuant soit l'ajout, la suppression, la modification, la recherche d'un patient.

✚ **Acteur :** Le médecin.

✚ **Séquencement :** ce cas commence lorsque le médecin clique sur «Patients »

✚ **Précondition :** l'administrateur s'est authentifié sur le système

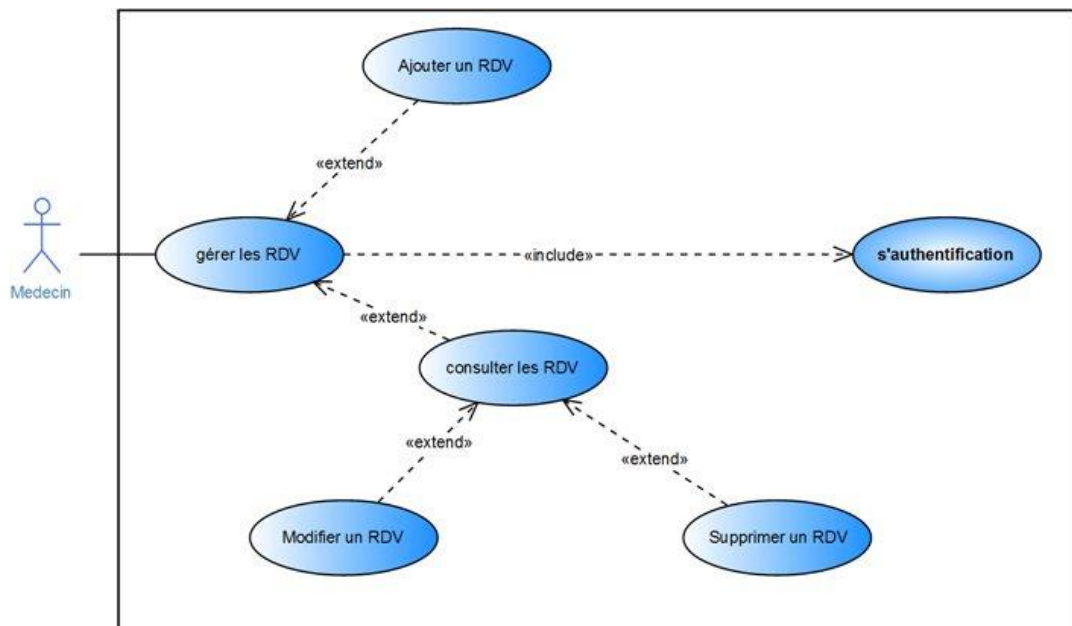
✚ **Scénario nominal :**

1. Le médecin clique sur le bouton «Patients».
2. Le système affiche l'interface de la liste des patients.
3. Le médecin peut choisir parmi l'action « ajouter », « modifier », « supprimer », « supprimer » qui se trouvent dans menu bar
4. Si le médecin choisit « Ajouter » donc il saisit les informations relatives au nouvel patient; s'il choisit « modifier » il doit rechercher au patient lequel veut le modifier puis il change ses données de même pour supprimer patient il cherche puis il supprime.

✚ **Scénario alternatif :** L'application envoie un message d'erreur lors de l'authentification.

✚ **Postcondition :** l'application affiche un message de confirmation

ii. Cas d'utilisation de gestion des RDV :



➔ Description textuelle :

✚ **Nom de cas :** gérer les RDV

✚ **But :** Le médecin peut gérer les RDV en effectuant soit l'ajout, la suppression, la modification d'un RDV.

✚ **Acteur :** Le médecin.

✚ **Séquencement :** ce cas commence lorsque le médecin clique sur «RDV »

✚ **Précondition :** l'administrateur s'est authentifié sur le système

✚ **Scénario nominal :**

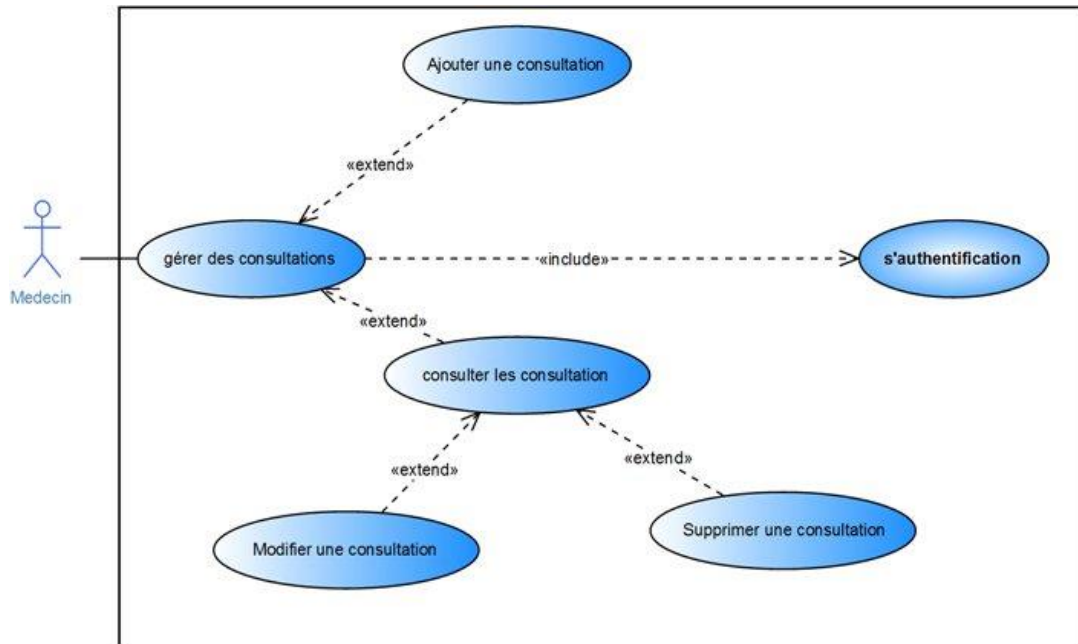
1. Le médecin clique sur le bouton «RDV».
2. Le système affiche l'interface de la liste des RDV.
3. Le médecin peut choisir parmi l'action « ajouter », « modifier », « supprimer » qui se trouvent dans menu bar
4. Si le médecin choisit « Ajouter » donc il saisit les informations relatives au nouvel RDV; s'il choisit « modifier » il doit rechercher au RDV lequel veut le modifier puis il change ses données de même pour supprimer RDV il cherche puis il supprime.

✚ **Scénario alternatif :** L'application envoie un message d'erreur lors de l'authentification.

✚ **Postcondition :** l'application affiche un message de confirmation



iii. Cas d'utilisation de gestion des consultations :



➔ Description textuelle :

✚ **Nom de cas :** gérer les consultations

✚ **But :** Le médecin peut gérer les consultations en effectuant soit l'ajout, la suppression, la modification d'une consultation.

✚ **Acteur :** Le médecin.

✚ **Séquencement :** ce cas commence lorsque le médecin clique sur «Consultations »

✚ **Précondition :** l'administrateur s'est authentifié sur le système

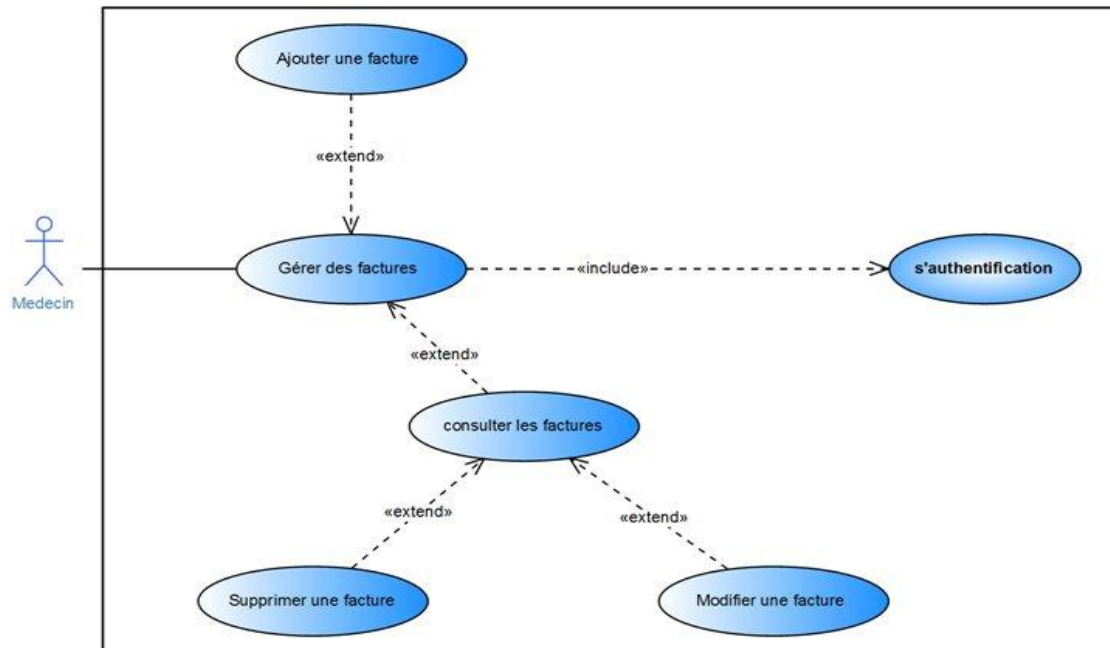
✚ **Scénario nominal :**

1. Le médecin clique sur le bouton «Consultations».
2. Le système affiche l'interface de la liste des consultations.
3. Le médecin peut choisir parmi l'action « ajouter », « modifier », « supprimer » qui se trouvent dans menu bar
4. Si le médecin choisit « Ajouter » donc il saisit les informations relatives à la nouvelle consultation; s'il choisit « modifier » il doit rechercher à la consultation laquelle veut la modifier puis il change ses données de même pour supprimer consultation il cherche puis il supprime.

✚ **Scénario alternatif :** L'application envoie un message d'erreur lors de l'authentification.

✚ **Postcondition :** l'application affiche un message de confirmation

iv. Cas d'utilisation de gestion des factures :



➔ Description textuelle :

✚ **Nom de cas :** gérer les factures

✚ **But :** Le médecin peut gérer les factures en effectuant soit l'ajout, la suppression, la modification d'une facture.

✚ **Acteur :** Le médecin.

✚ **Séquencement :** ce cas commence lorsque le médecin clique sur «Facture »

✚ **Précondition :** l'administrateur s'est authentifié sur le système

✚ **Scénario nominal :**

1. Le médecin clique sur le bouton «Facture».
2. Le système affiche l'interface de la liste des factures.
3. Le médecin peut choisir parmi l'action « ajouter », « modifier », « supprimer » qui se trouvent dans menu bar
4. Si le médecin choisit « Ajouter » donc il saisit les informations relatives à la nouvelle facture; s'il choisit « modifier » il doit rechercher à la facture laquelle veut la modifier puis il change ses données de même pour supprimer consultation il cherche puis il supprime.

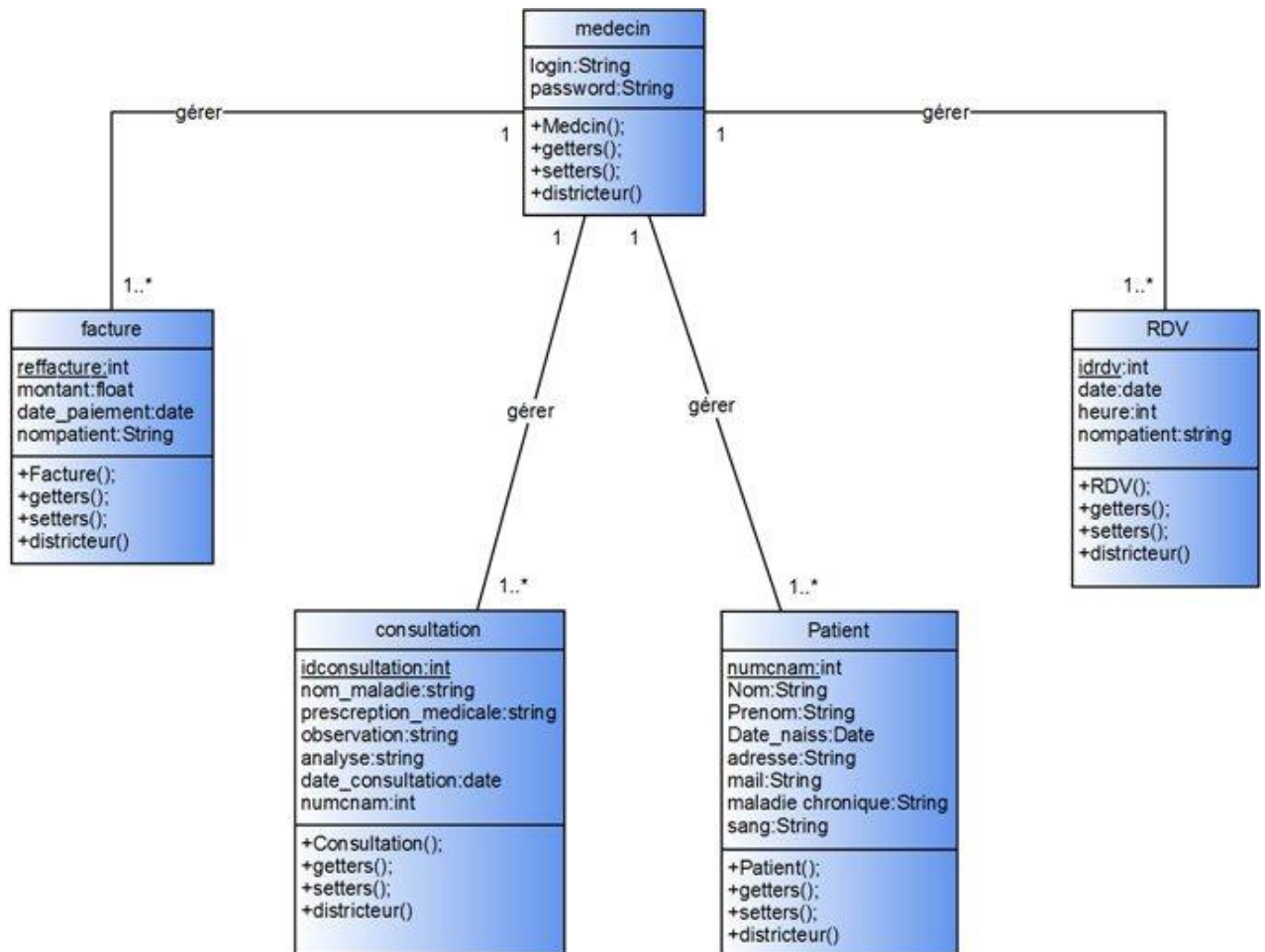
✚ **Scénario alternatif :** L'application envoie un message d'erreur lors de l'authentification.

✚ **Postcondition :** l'application affiche un message de confirmation

#### 4. Diagramme de classe :

##### i. Définition :

Le diagramme de classe représente un élément très important de la modélisation. Il représente la structure statique du système en termes de classes et de relations entre classe. Le but de ce diagramme est de modéliser les entités du système. Ainsi, il permet de définir les composantes du système final.



## 5. Schéma Relationnel de la base des données :

### i. Définition

Il est possible de traduire un diagramme de classe en modèle relationnel. Aujourd'hui, lors de la conception de base de données, il devient de plus en plus courant d'utiliser la modélisation UML plutôt que le traditionnel modèle entités-associations.

### ii. Modèle relationnel :

Le modèle relationnel représente la structure de la base de données, à savoir

- La structure des tables (nom, nom des champs)
- Les liens entre elles.
- La clé primaire, obligatoire pour chaque table, permet d'identifier de façon unique chaque enregistrement de la table.
- La clé étrangère matérialise

**Médecin** (login, Password)

**Patient** (numcnam, Nom, Prenom, Date\_naiss, adresse, mail, maladie chronique, sang)

**RDV** (idrdv, heure, date)

**Consultation** (idconsultation, nom\_maladie, prescription-médicale, observation, analyse, #numcnam)

**Facture** (reffacture, date paiement, montant, nompatient)

## **Chapitre 4 : Réalisation**



## I. Introduction :

Dans ce chapitre on va illustrer brièvement les environnements logiciels de développement de l'application puis on va détailler les outils utilisés pour la réalisation. De plus on va mettre en évidence les langages utilisés pour développer notre application.

## II. Environnement de développement :



**WampServer**

Est une plateforme de développement WEB sous Windows pour des applications Web dynamique à l'aide du serveur Apache, du langage de scriptes PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données.



**Photoshop**

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par **Adobe**. Il est principalement utilisé pour le traitement de photographies numériques, mais sert également à la création d'images.



**NetBeans**

NetBeans est un environnement de développement intégré. En plus de Java, NetBeans permet la prise en charge native de divers langages tels que le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby).

### III. Les interfaces graphiques de l'application GCM :

#### 🚦 Interface authentification :

L'authentification de l'utilisateur est obligatoire pour chaque accès à l'application. C'est la première tâche que l'utilisateur doit effectuer pour accéder au menu principal.



Figure1: Interface Authentification de GCM

- Après le saisi du nom d'utilisateur et du mot de passe l'utilisateur doit appuyer sur le bouton « connecter » pour ouvrir le menu principal. Si un champ obligatoire est vide un message d'erreur s'affiche, comme illustre la figure ci-dessous.

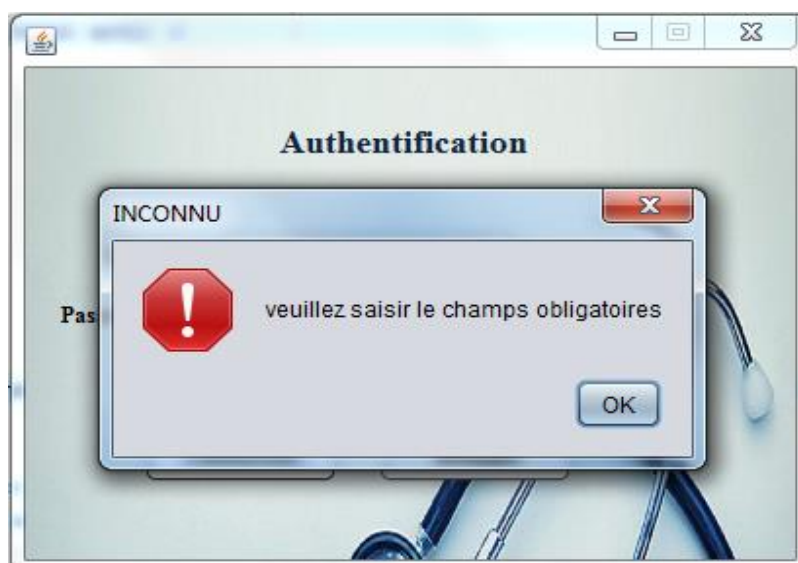


Figure 2: interface de message d'erreur des champs vide

- Si les informations sont incorrectes un message d'erreur s'affiche :



Figure 3: interface de message d'erreur des champs incorrects

#### **Interface d'accueil :**

Après avoir accédé au menu principal le médecin peut choisir patients, consultation, facture ou bien RDV. La figure suivante définit ces différentes tâches :

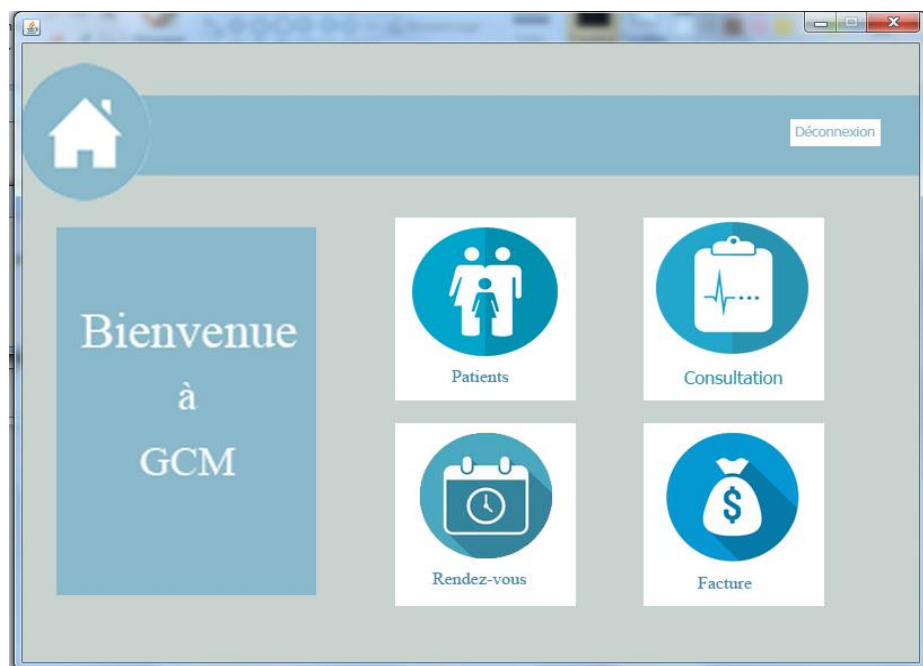


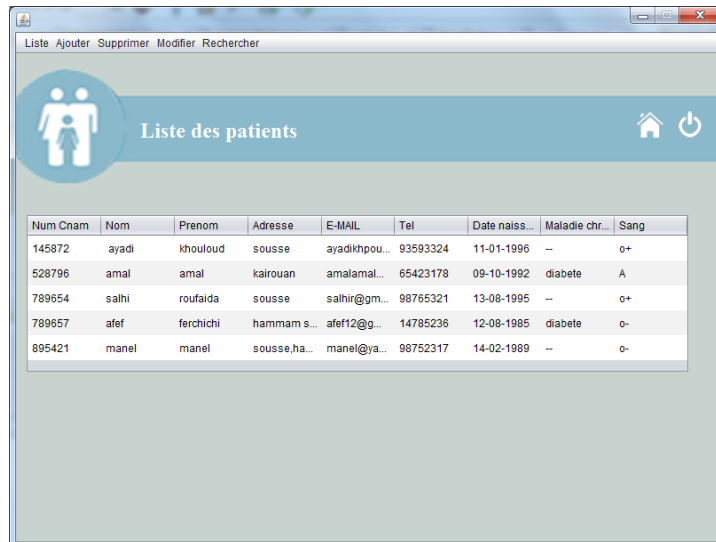
Figure 4: interface d'accueil

#### **Interface Liste patients :**

Lorsque l'utilisateur choisit patients, la liste des patients s'affiche puis il peut ajouter/modifier/supprimer ou bien rechercher à un patient ou bien il peut se déconnecter ou revenir à l'accueil

Ce qui illustre la figure suivante :





**Figure 5: interface de Liste des patients**

### Interface ajouter un patient :

Pour ajouter un patient l'utilisateur doit choisir « ajouter » dans le menubar et donc l'interface d'ajout s'affiche comme illustre la figure suivante.

Nom:  Prénom:

Date Naissance:  Adresse:

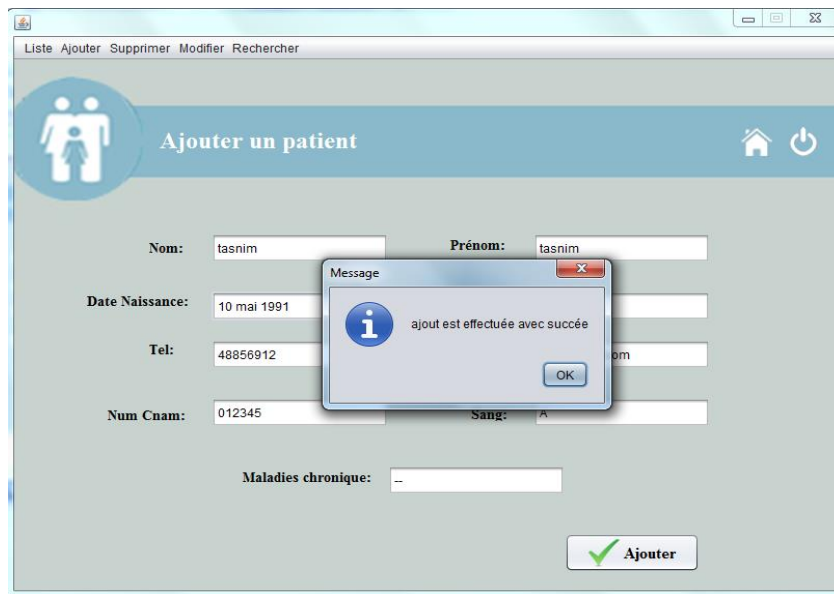
Tel:  E-mail:

Num Cnam:  Sang:

Maladies chronique:

**Figure 6: Interface d'ajout d'un patient**

- Après la saisie des champs un message de confirmation d'ajout sera affiché.

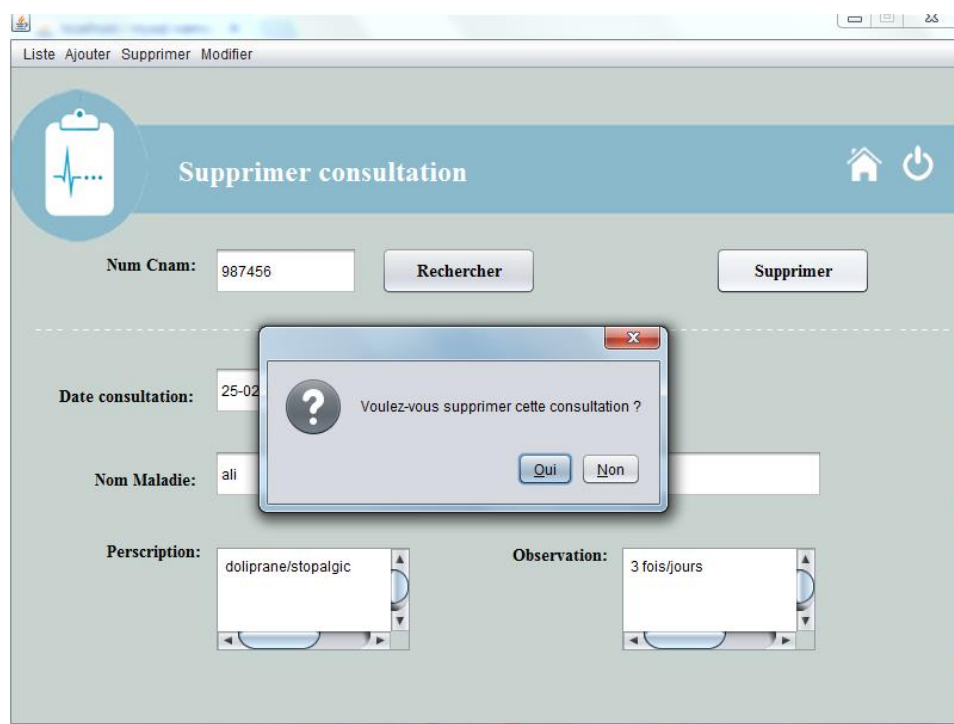


**Figure 7: Interface de confirmation d'ajout**

**Remarque :** c'est exactement la même chose pour les RDV, les consultations et les factures.

#### **Interface de suppression d'une consultation :**

Pour supprimer une consultation l'utilisateur choisit »supprimer » dans menubar il cherche à la consultation laquelle veut la supprimer et après l'application affiche un message de confirmation de suppression définitive et après le clique sur ok un message s'affiche pour indiquer que le message a été supprimé.



### 🚦 Interface Recherche d'un patient :

L'utilisateur peut chercher à un patient quelconque. De même il peut se déconnecter ou bien revenir au page d'accueil.

The screenshot shows a web application window titled 'Rechercher un patient'. At the top, there is a navigation bar with links: 'Liste', 'Ajouter', 'Supprimer', 'Modifier', and 'Rechercher'. Below the navigation bar, there is a header section with a family icon and the title 'Rechercher un patient'. On the right side of the header, there are icons for home and power. The main content area contains a search form. It starts with a label 'Num Cnam:' followed by a text input field containing '147258' and a 'Rechercher' button. Below this, there is a dashed line. Underneath the dashed line, there are several input fields for patient details: 'Nom:' with 'ayadi', 'Prénom:' with 'khouloud', 'Date Naissance:' with '11-01-1996', 'Adresse:' with 'sousse', 'Tel:' with '93593324', 'E-mail:' with 'ayadikhouloud@gmail.com', 'Maladies chronique:' with '--', and 'Sang:' with 'o+'.

### 🚦 Interface de modification RDV :

L'utilisateur peut modifier les données d'un tel RDV et après la modification d'un message de confirmation s'affiche et après le clique sur ok un autre message s'affiche pour indiquer que la modification est terminée comme illustre la figure suivante :

The screenshot shows a web application window titled 'Modifier des RDV'. At the top, there is a navigation bar with links: 'Liste', 'Ajouter', 'Supprimer', and 'Modifier'. Below the navigation bar, there is a header section with a calendar icon and the title 'Modifier des RDV'. On the right side of the header, there are icons for home and power. The main content area contains a form for modifying an appointment. It starts with a label 'Nom du Patient:' followed by a text input field containing 'samar sam'. To the right of this field is a 'Modifier' button. Below this, there is a dashed line. Underneath the dashed line, there are input fields for 'Date:' with '01-02-2018' and 'Heure:' with '09:25'. A confirmation dialog box is overlaid on the form. The dialog box has a question mark icon and the text 'voulez vous modifier ce rdv ?'. It has two buttons: 'Oui' and 'Non'.

#### **IV. Conclusion**

A travers ce chapitre, nous avons présenté la partie réalisation de l'application représentant quelques interfaces graphiques que nous avons jugées les plus importantes.

## Conclusion

Vue l'importance considérable de la santé dans la vie quotidienne, il faut fournir des services de qualité permettant la bonne organisation de ce secteur, d'où la nécessité d'offrir une application qui peut répondre à ces besoins. C'est dans ce cadre que s'est situé notre projet de développement. Il fut question d'une implémentation d'une application bureautique capable de gérer un cabinet médical.

Au terme de la réalisation de ce rapport de ce projet « Développement d'une application bureautique de gestion de cabinet médical ». Nous avons situé le projet dans son cadre général tout en mettant l'accent sur son originalité. Nous avons ensuite étudié les solutions possibles pour atteindre l'objectif par rapport à la situation existante. En effet, notre application a réussi à automatiser les processus des gestions dans un cabinet médical et de réduire sa durée réelle

Comme perspectives nous proposons d'ajouter d'autres tâches tel que donner la possibilité à un patient de consulter ses consultations en plus nous proposons d'intégrer une base de données contenant les pharmacies et les laboratoires existants pour faciliter les tâches pour un patient.

