

# **Rapport de projet fin d'année**

## **3<sup>ème</sup>**

**Ingénierie Informatique et Réseaux**

**Sous le thème**

---

# **LA GESTION D'UN CABINET MEDICAL**

---

**Réalisé par :**

TITI SAID

EL IBRAHIMI JAWAD

**Encadré par :**

Tuteur de l'école : Pr. EL BOUJAMAI

ANNEE UNIVERSITAIRE: 2023-2024

## **Remerciement :**

Nous tenons à exprimer notre sincère gratitude envers notre professeur, Mr EL BOUJAMAI, pour son dévouement et son encadrement tout au long du projet. Nous souhaitons également remercier l'ensemble du corps pédagogique de l'EMSI et tous les enseignants qui ont enrichi notre parcours scolaire. Leur soutien et leur expertise ont été d'une valeur inestimable pour notre développement académique et professionnel.

Nous dédions ce travail à

Nos très chers parents, Rien au monde ne pourrait compenser les sacrifices que vous avez consentis pour notre éducation et pour notre bien-être afin que nous puissions nous consacrer pleinement à nos études. Puisse Dieu, le Tout-Puissant, vous accorder santé, prospérité et longévité.

À toute notre famille, Merci pour votre soutien inconditionnel.

À tous nos chers amis, Merci pour votre soutien moral et votre aide précieuse.

À tous nos professeurs, Pour leurs efforts inestimables afin de nous assurer une formation d'excellence.

## **Resume :**

Notre projet de fin d'année se concentre sur le développement d'une application web innovante dédiée à l'optimisation de la gestion des cabinets médicaux. Les applications web constituent une composante essentielle de notre quotidien numérique moderne, conçues pour fonctionner à travers un navigateur web et offrant une variété de fonctionnalités et de services accessibles en ligne. Depuis leur essor dans les années 1990, les applications web ont évolué de simples sites statiques à des plateformes interactives et dynamiques, répondant à une gamme étendue de besoins utilisateurs.

À l'époque moderne, la gestion des cabinets médicaux joue un rôle crucial dans l'amélioration des soins de santé en mettant à disposition des professionnels de santé et des patients des outils essentiels pour la planification des rendez-vous, la gestion des dossiers médicaux et la communication. Cette avancée rapide dans le domaine de la santé a engendré une forte demande pour des solutions de gestion médicale sécurisées, fiables et conviviales. Ainsi, notre projet se focalise sur le développement d'une application web innovante dédiée à l'optimisation de la gestion d'un cabinet médical.

Ce rapport présente une analyse détaillée du processus de développement de notre application, mettant en lumière les technologies, les outils et les méthodologies utilisés pour créer une plateforme de gestion médicale robuste et performante. Nous avons axé notre approche sur la sécurisation des données médicales, l'optimisation de l'expérience utilisateur et la mise en œuvre de fonctionnalités avancées pour répondre aux attentes des utilisateurs modernes, tant les praticiens que les patients.

Au cours de ce projet, nous avons eu l'occasion d'appliquer nos compétences en programmation, en conception d'interface utilisateur et en gestion de projet, tout en relevant les défis inhérents au développement d'une solution de gestion médicale complète. Ce rapport offre un aperçu complet de notre démarche méthodologique, de l'analyse des besoins à la mise en œuvre des fonctionnalités clés, et met en lumière les enseignements tirés et les succès obtenus au cours de cette année enrichissante.

Pour la réalisation de ce projet, nous avons utilisé divers outils et technologies. Django, un framework web open-source en Python, a été choisi pour favoriser le développement rapide et une conception propre et pragmatique. HTML et CSS ont été utilisés pour structurer et styliser les pages web, tandis que Bootstrap a permis de créer des interfaces web et des applications mobiles responsives de manière efficace. XAMPP a servi d'environnement de développement web pour la gestion des serveurs locaux, et VS Code, un éditeur de code léger et performant, a été employé pour le développement.

En développant cette application, nous visons à transformer la gestion des cabinets médicaux en simplifiant les processus administratifs, en améliorant la communication et en sécurisant les données médicales. Cette avancée technologique pourrait servir de modèle pour d'autres projets similaires dans le futur, contribuant à l'amélioration continue des services de santé numériques.

## **Abstract :**

This project focuses on the development of an innovative web application dedicated to optimizing the management of medical practices. Web applications are essential components of our modern digital lives, designed to operate through web browsers and offering a variety of accessible functionalities and services. Since their rise in the 1990s, web applications have evolved from simple static sites to interactive and dynamic platforms, meeting a wide range of user needs.

In the contemporary medical field, the management of medical practices plays a crucial role in improving healthcare by providing essential tools for scheduling appointments, managing medical records, and facilitating communication. This rapid advancement in the healthcare domain has generated a high demand for secure, reliable, and user-friendly medical management solutions. Therefore, our project focuses on developing an innovative web application dedicated to optimizing medical practice management.

The report provides a detailed analysis of the development process of our application, highlighting the technologies, tools, and methodologies used to create a robust and high-performance medical management platform. Our approach focuses on securing medical data, optimizing user experience, and implementing advanced functionalities to meet the expectations of modern users, including both practitioners and patients.

Throughout this project, we have applied our skills in programming, user interface design, and project management, overcoming the inherent challenges of developing a comprehensive medical management solution. The report offers a complete overview of our methodological approach, from needs analysis to the implementation of key functionalities, and highlights the lessons learned and successes achieved during this enriching year.

For the execution of this project, various tools and technologies were utilized. Django, an open-source web framework in Python, was chosen for its rapid development capabilities and clean, pragmatic design. HTML and CSS were used for structuring and styling web pages, while Bootstrap enabled the efficient creation of responsive web interfaces and mobile applications. XAMPP served as a web development environment for managing local servers, and VS Code, a lightweight and versatile code editor, was employed for development.

By developing this application, we aim to transform the management of medical practices by simplifying administrative processes, improving communication, and securing medical data. This technological advancement could serve as a model for similar future projects, contributing to the continuous improvement of digital health services.

## Table des matières

<b>Introduction.....</b>	<b>5</b>
<b>Chapitre 1 : Contexte générale du projet.....</b>	<b>5</b>
1.    Cadre du Projet.....	5
2.    Objectifs du projet .....	6
<b>Chapitre 2 : Analyse et Conception.....</b>	<b>7</b>
1-    Introduction.....	7
2-    Diagramme de cas d'utilisation ( USE CASE ) .....	7
2.1 Patient.....	8
2.2 Docteur.....	9
2.3 Administrateur.....	10
3-    Diagramme de classes.....	11
4-    Diagramme de sequence .....	12
4.1 La connexion pour l'utilisateur.....	12
4.2 L'ajout pour le Docteur.....	14
4.3 L'ajout pour le Patient.....	15
4.4 L'ajout du Rendez-vous.....	17
<b>Chapitre 3 : Réalisation.....</b>	<b>18</b>
1-    Les outils de développement.....	18
2.    Enchainement des interfaces .....	20
<b>Conclusion .....</b>	<b>26</b>

## **1- Introduction :**

Les applications web constituent une composante essentielle de notre quotidien numérique moderne. Elles sont conçues pour fonctionner à travers un navigateur web et offrent une variété de fonctionnalités et de services accessibles en ligne. Depuis leur essor dans les années 1990, les applications web ont évolué de simples sites statiques à des plateformes interactives et dynamiques, répondant à une gamme étendue de besoins utilisateurs.

À l'époque moderne, la gestion des cabinets médicaux joue un rôle crucial dans l'amélioration des soins de santé en mettant à disposition des professionnels de santé et des patients des outils essentiels pour la planification des rendez-vous, la gestion des dossiers médicaux et la communication. Cette avancée rapide dans le domaine de la santé a engendré une forte demande pour des solutions de gestion médicale sécurisées, fiables et conviviales. Ainsi, notre projet de cette année se focalise sur le développement d'une application web innovante dédiée à l'optimisation de la gestion d'un cabinet médical.

Ce rapport présente une analyse détaillée du processus de développement de notre application, mettant en lumière les technologies, les outils et les méthodologies utilisés pour créer une plateforme de gestion médicale robuste et performante. Nous avons axé notre approche sur la sécurisation des données médicales, l'optimisation de l'expérience utilisateur et la mise en œuvre de fonctionnalités avancées pour répondre aux attentes des utilisateurs modernes, tant les praticiens que les patients.

Au cours de ce projet, nous avons eu l'occasion d'appliquer nos compétences en programmation, en conception d'interface utilisateur et en gestion de projet, tout en relevant les défis inhérents au développement d'une solution de gestion médicale complète. Ce rapport offre un aperçu complet de notre démarche méthodologique, de l'analyse des besoins à la mise en œuvre des fonctionnalités clés, et met en lumière les enseignements tirés et les succès obtenus au cours de cette année enrichissante.

En conclusion, la gestion des cabinets médicaux à l'ère numérique nécessite une intégration harmonieuse de technologies avancées et une compréhension approfondie des besoins des utilisateurs. Notre projet de fin d'année, en développant une application web innovante, s'efforce de répondre à ces besoins de manière efficace et sécurisée, tout en offrant une expérience utilisateur exceptionnelle. Nous espérons que notre plateforme apportera une réelle valeur ajoutée aux praticiens et aux patients, en rendant la gestion des soins de santé plus efficace, plus sûre et plus accessible.

# **Chapitre 1 : Contexte générale du projet**

## **1- Cadre du Projet**

Dans le contexte actuel de transformation numérique, la gestion des cabinets médicaux a évolué pour intégrer des technologies avancées qui optimisent les processus administratifs et médicaux. Cette modernisation vise à améliorer l'efficacité des soins, la gestion des données patients et la communication entre les différentes parties prenantes. Le projet de développement d'une application web pour la gestion d'un cabinet médical s'inscrit dans cette dynamique, cherchant à répondre aux besoins croissants de sécurité, de fiabilité et de convivialité des plateformes de gestion médicale.

## **2- Objectifs du projet**

L'objectif principal de ce projet est de créer une solution innovante et complète qui permet aux professionnels de santé de gérer efficacement leurs activités quotidiennes, tout en assurant une expérience utilisateur optimale pour les patients. Pour ce faire, nous avons choisi des technologies modernes telles que Django pour le backend et HTML, CSS pour le frontend. Ces choix technologiques sont motivés par la robustesse, la sécurité et la scalabilité qu'ils offrent, essentiels pour une application de gestion médicale.

Ce rapport présente une exploration approfondie du processus de développement de notre application web pour la gestion des cabinets médicaux. Les objectifs principaux du projet sont les suivants :

### **a- Automatisation des Rendez-vous :**

Mettre en place un système en ligne permettant aux patients de prendre, modifier et annuler des rendez-vous avec leurs médecins de manière autonome, réduisant ainsi la charge administrative.

### **b- Gestion des Dossiers Médicaux :**

Développer une base de données sécurisée pour stocker et gérer les dossiers médicaux des patients, garantissant un accès rapide et confidentiel aux informations médicales par les professionnels de santé autorisés.

### **c- Suivi des Interactions Médecin-Patient :**

Créer un module pour enregistrer et suivre toutes les interactions entre les médecins et les patients, facilitant ainsi la continuité des soins et une prise en charge personnalisée.

### **d- Optimisation de la Charge de Travail des Médecins :**

Automatiser les tâches administratives répétitives pour permettre aux médecins de se concentrer davantage sur les soins aux patients, améliorant ainsi l'efficacité et la qualité des services médicaux.

**e- Facilité d'Utilisation:**

Concevoir une interface utilisateur intuitive et conviviale pour garantir une utilisation facile et pratique pour tous les utilisateurs.

En atteignant ces objectifs, notre projet vise à transformer la gestion des cabinets médicaux en offrant aux professionnels de santé et aux patients une solution intégrée, sécurisée et efficace pour améliorer la qualité des soins et la gestion administrative.

## **Chapitre 2 : Analyse et Conception**

### **1- Introduction :**

La méthode que nous avons adopté pour réaliser l'analyse et la conception de notre système d'information est la méthode **UML** : Elle permet la séparation entre les données et les traitements effectués en plusieurs modèles conceptuels qui sont répartis sur 3 diagrammes : Le diagramme de cas d'utilisation, le diagramme de classe, diagramme de séquence. Dans cette partie, nous allons présenter quelques-unes de ces méthodes.

Le langage de modélisation unifié (UML, pour Unified Modeling Language) est un langage standardisé utilisé en génie logiciel pour visualiser, spécifier, construire et documenter les artefacts d'un système logiciel. Une vue est composée d'un ou plusieurs diagrammes, et on distingue deux types de vues :

**Vues statiques** : Elles représentent le système de manière physique et comprennent :

- Les diagrammes de classes
- Les diagrammes de cas d'utilisation

**Vues dynamiques** : Elles montrent le fonctionnement du système et comprennent :

- Les diagrammes de séquence

UML est utile à toutes les étapes de la conception du projet. Parmi ses avantages, on peut citer :

- Des descriptions graphiques
- Des vues différentes à chaque étape
- Un recoupement des descriptions

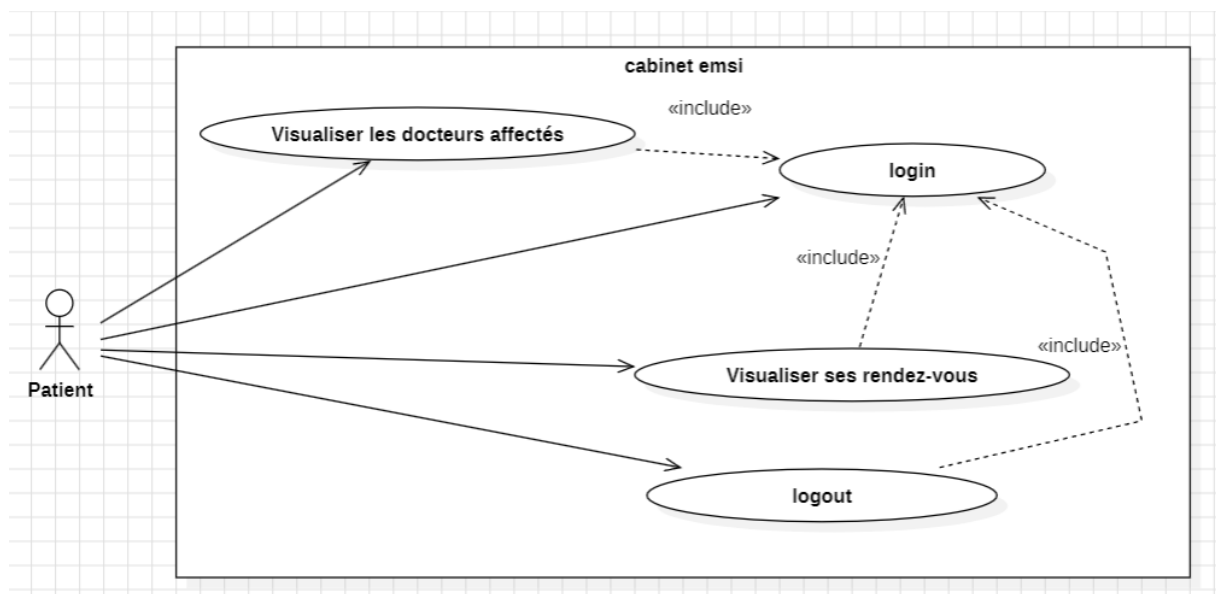
Les incohérences peuvent être mises en évidence par des incomplétudes, et UML s'adapte facilement aux différentes méthodes. C'est également un excellent outil pour démarrer un projet.



## 2- Diagramme de Cas d'utilisation ( USE CASE)

Acteurs	Fonctions
Administrateur	Login  CRUD (Create, Read, Update, Delete) Docteurs  CRUD (Create, Read, Update, Delete) Patients  CRUD (Create, Read, Update, Delete) Appointment  Logout
Docteur	Login  Read Patient , Discharge Patient  Read Appointment, Delete Appointment  Logout
Patient	Login  Read Docteurs  Read Appointment  Logout

### 2.1 Patient :



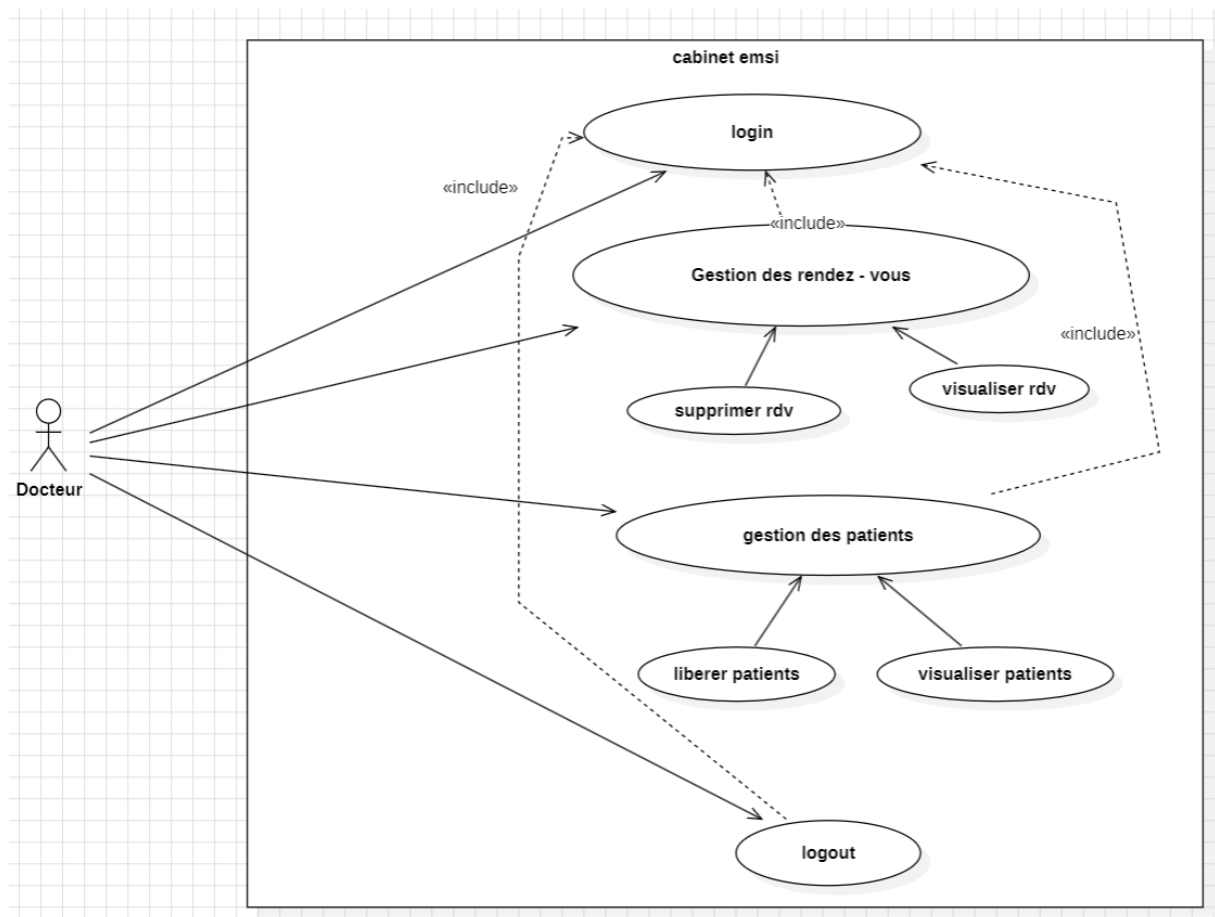
Ce diagramme représente les interactions entre un patient et le système de gestion d'un cabinet médical (cabinet emsi).

L'acteur principal du diagramme est le patient. Le patient peut effectuer plusieurs actions dans le système :

1. **Visualiser les docteurs affectés** : Cette fonctionnalité permet au patient de voir les docteurs qui lui sont assignés. Avant d'accéder à cette information, le patient doit être connecté au système.
2. **Se connecter (login)** : Pour accéder à la plupart des fonctionnalités du système, le patient doit d'abord se connecter. Cette étape est essentielle pour sécuriser l'accès aux données personnelles et aux services du patient.
3. **Visualiser ses rendez-vous** : Une fois connecté, le patient peut consulter ses rendez-vous. Cette action est incluse dans le processus de connexion, car elle nécessite l'authentification du patient.
4. **Se déconnecter (logout)** : Après avoir terminé ses actions, le patient peut se déconnecter du système pour assurer la sécurité de ses informations.

Les relations entre ces actions sont indiquées par des lignes. Par exemple, pour visualiser les docteurs affectés ou ses rendez-vous, le patient doit d'abord se connecter, comme indiqué par les lignes d'inclusion ("include") reliant ces actions au processus de connexion.

## 2.2 Docteur :

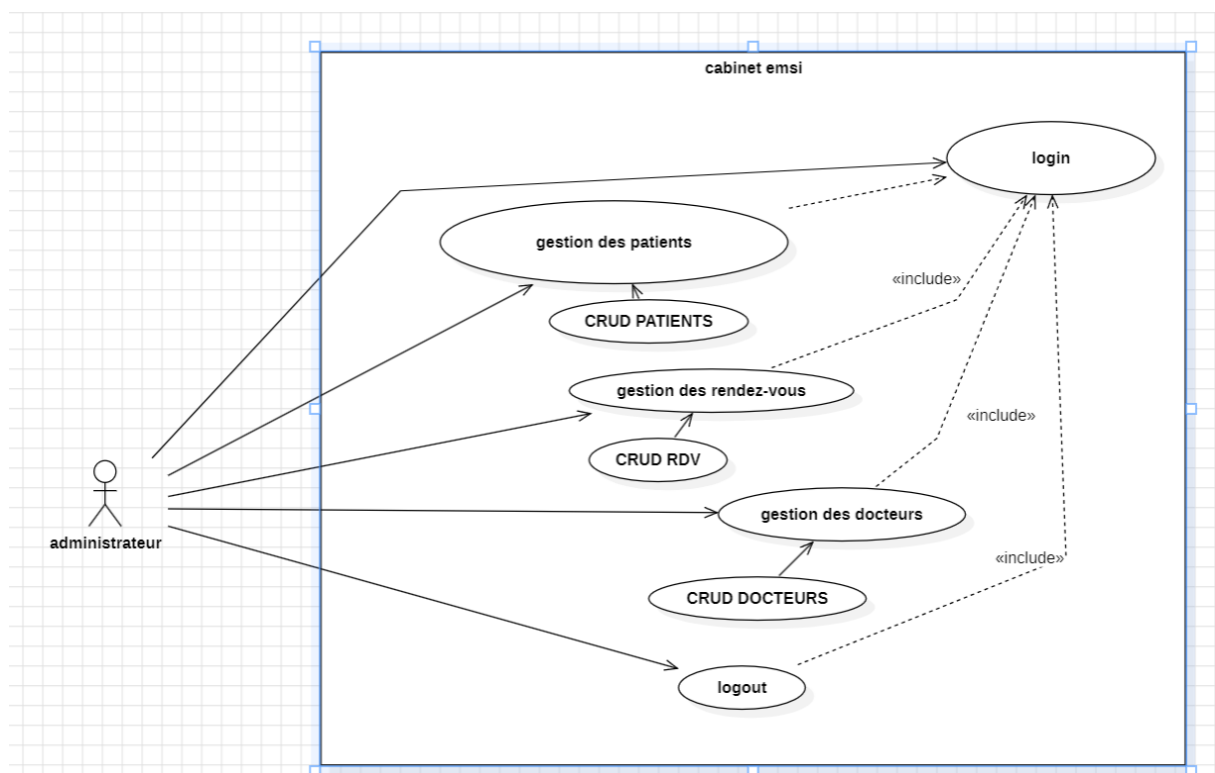


L'acteur principal du diagramme est le docteur. Le docteur peut effectuer plusieurs actions dans le système :

1. **Se connecter (login)** : Avant de pouvoir accéder à d'autres fonctionnalités, le docteur doit se connecter au système. Cette étape est essentielle pour sécuriser l'accès aux données et aux services.
2. **Gestion des rendez-vous** : Une fois connecté, le docteur peut gérer les rendez-vous. Cette gestion inclut plusieurs sous-actions :
  - **Supprimer rendez-vous (supprimer rdv)** : Le docteur peut annuler ou supprimer des rendez-vous.
  - **Visualiser rendez-vous (visualiser rdv)** : Le docteur peut consulter la liste des rendez-vous planifiés.
3. **Gestion des patients** : Le docteur peut aussi gérer les informations des patients, ce qui inclut deux sous-actions :
  - **Libérer patients (liberer patients)** : Le docteur peut libérer des patients, probablement après un traitement ou une consultation.
  - **Visualiser patients** : Le docteur peut consulter les informations des patients.
4. **Se déconnecter (logout)** : Après avoir terminé ses tâches, le docteur peut se déconnecter du système pour assurer la sécurité des informations.

Les relations entre ces actions sont indiquées par des lignes. Par exemple, la gestion des rendez-vous et des patients nécessite d'abord que le docteur soit connecté, comme le montrent les lignes d'inclusion ("include") reliant ces actions au processus de connexion.

### 2.3 Administrateur :

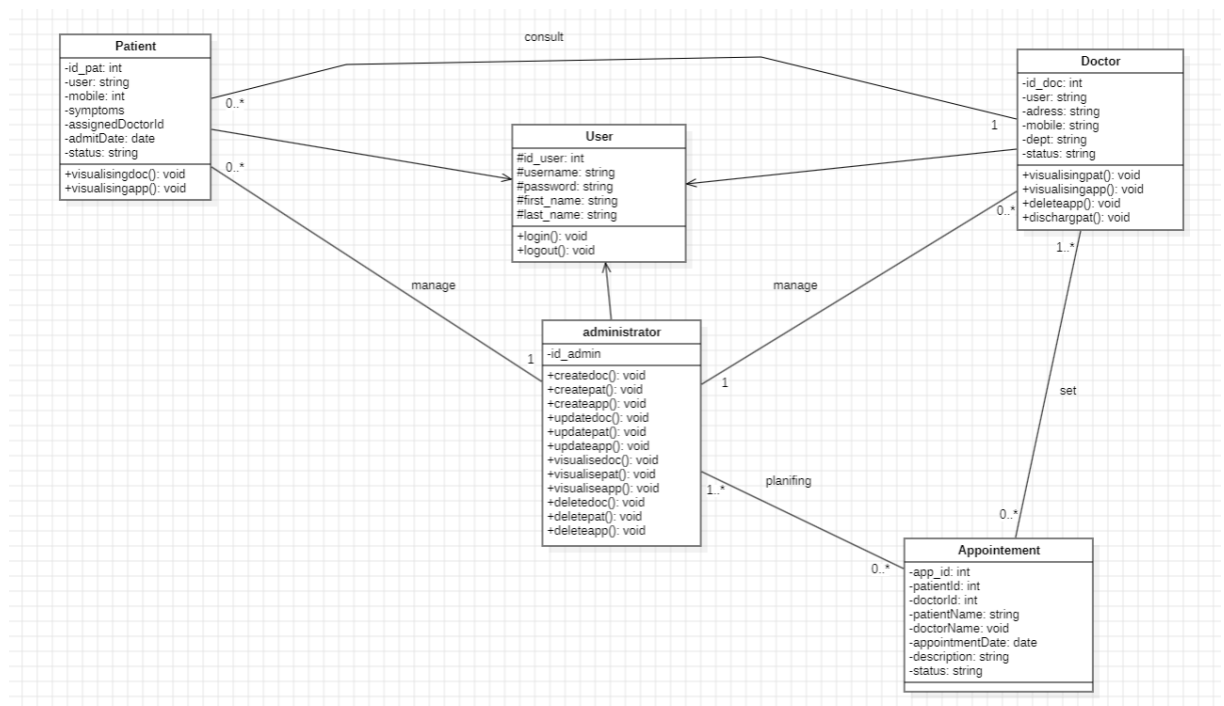


L'acteur principal du diagramme est l'administrateur. L'administrateur peut effectuer plusieurs actions dans le système :

1. **Se connecter (login)** : Avant de pouvoir accéder à d'autres fonctionnalités, l'administrateur doit se connecter au système. Cette étape est essentielle pour sécuriser l'accès aux données et aux services.
2. **Gestion des patients** : Une fois connecté, l'administrateur peut gérer les patients. Cette gestion inclut plusieurs sous-actions :
  - **CRUD Patients** : L'administrateur peut créer, lire, mettre à jour et supprimer les informations des patients.
3. **Gestion des rendez-vous** : L'administrateur peut également gérer les rendez-vous, ce qui inclut :
  - **CRUD RDV** : L'administrateur peut créer, lire, mettre à jour et supprimer les rendez-vous.
4. **Gestion des docteurs** : L'administrateur peut gérer les informations des docteurs, ce qui inclut :
  - **CRUD Docteurs** : L'administrateur peut créer, lire, mettre à jour et supprimer les informations des docteurs.
5. **Se déconnecter (logout)** : Après avoir terminé ses tâches, l'administrateur peut se déconnecter du système pour assurer la sécurité des informations.

Les relations entre ces actions sont indiquées par des lignes. Par exemple, la gestion des patients, des rendez-vous et des docteurs nécessite d'abord que l'administrateur soit connecté, comme le montrent les lignes d'inclusion (« include ») reliant ces actions au processus de connexion.

### 3- Diagramme de classes :



La classe **Patient** comprend des attributs tels que l'identifiant du patient, le nom d'utilisateur, le numéro de mobile, les symptômes, l'identifiant du médecin assigné, la date d'admission, et le statut du patient. Elle inclut également des méthodes pour visualiser les informations du médecin assigné et les rendez-vous. La classe **User** inclut des attributs comme l'identifiant de l'utilisateur, le nom d'utilisateur, le mot de passe, le prénom et le nom de famille, ainsi que des méthodes pour se connecter et se déconnecter.

La classe **Doctor** comprend des attributs tels que l'identifiant du médecin, le nom d'utilisateur, l'adresse, le numéro de mobile, le département, et le statut. Elle possède des méthodes pour visualiser les informations des patients et des rendez-vous, supprimer des rendez-vous, et décharger des patients.

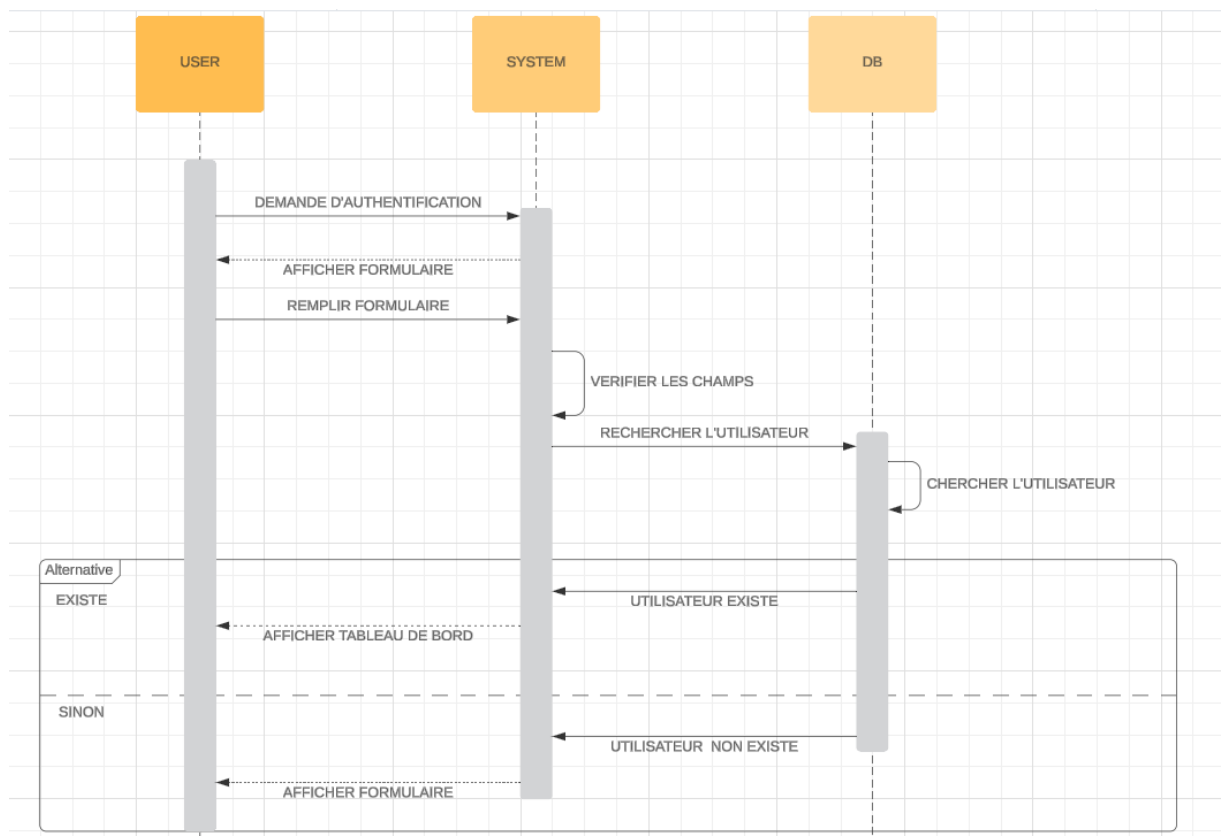
La classe **Administrator** a pour attribut l'identifiant de l'administrateur et des méthodes pour créer, mettre à jour, visualiser et supprimer des médecins, des patients, et des rendez-vous.

Enfin, la classe **Appointment** comprend des attributs comme l'identifiant du rendez-vous, les identifiants du patient et du médecin, le nom du patient, le nom du médecin, la date du rendez-vous, la description, et le statut.

Ces classes interagissent pour gérer efficacement les informations sur les patients, les médecins, les rendez-vous et les utilisateurs dans le cadre d'un système de gestion hospitalière.

#### 4- Diagramme de sequence :

##### 4.1 La connexion pour l'utilisateur :



Ce diagramme de séquence UML illustre le processus d'authentification d'un utilisateur dans un système. Il décrit l'interaction entre trois acteurs principaux : l'utilisateur, le système et la base de données (DB).

### Description du Processus

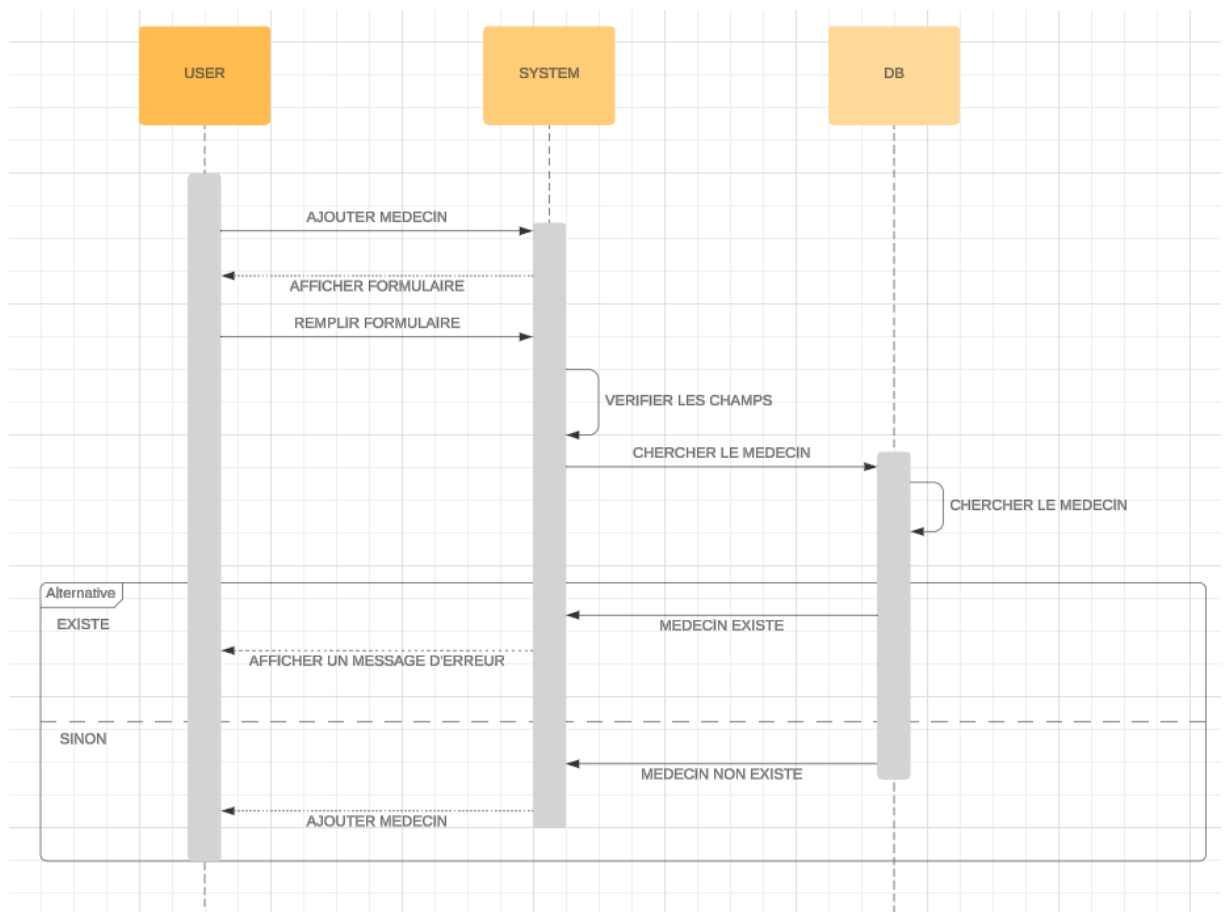
1. **Demande d'authentification :**
  - L'utilisateur initie le processus en envoyant une demande d'authentification au système.
2. **Affichage du formulaire :**
  - Le système répond à la demande en affichant un formulaire d'authentification pour que l'utilisateur puisse entrer ses informations.
3. **Remplissage du formulaire :**
  - L'utilisateur remplit le formulaire avec ses informations de connexion (nom d'utilisateur et mot de passe) et soumet le formulaire.
4. **Vérification des champs :**
  - Le système reçoit les informations soumises par l'utilisateur et vérifie les champs pour s'assurer qu'ils sont correctement remplis.
5. **Recherche de l'utilisateur :**
  - Une fois les champs vérifiés, le système recherche l'utilisateur dans la base de données.
  - Le système envoie une requête à la base de données pour chercher l'utilisateur avec les informations fournies.
6. **Recherche dans la base de données :**
  - La base de données reçoit la requête et cherche les informations de l'utilisateur.
  - La base de données renvoie les résultats de la recherche au système.
7. **Vérification de l'existence de l'utilisateur :**
  - Le système vérifie si l'utilisateur existe dans la base de données :
    - **Si l'utilisateur existe :**
      - Le système affiche le tableau de bord de l'utilisateur, indiquant une connexion réussie.
    - **Si l'utilisateur n'existe pas :**
      - Le système informe l'utilisateur que les informations fournies sont incorrectes ou que l'utilisateur n'existe pas, puis affiche à nouveau le formulaire d'authentification pour permettre à l'utilisateur de réessayer.

### Scénarios Alternatifs

- **Utilisateur existe :**
  - Le système affiche le tableau de bord de l'utilisateur après une connexion réussie.
- **Utilisateur n'existe pas :**
  - Le système affiche un message d'erreur et renvoie l'utilisateur au formulaire d'authentification.

Ce diagramme de séquence montre de manière claire et organisée les étapes et interactions nécessaires pour authentifier un utilisateur, en passant par la vérification des informations et la gestion des différents scénarios en fonction de l'existence ou non de l'utilisateur dans la base de données.

## 4.2 L'ajout pour le docteur :



Ce diagramme de séquence UML illustre le processus d'ajout d'un médecin dans un système. Il décrit l'interaction entre trois entités principales : l'utilisateur, le système, et la base de données (DB).

### Description du Processus

- 1. Initiation de l'ajout du médecin :**
  - L'utilisateur commence par envoyer une demande d'ajout de médecin au système.
- 2. Affichage du formulaire :**
  - Le système affiche un formulaire pour l'ajout du médecin, permettant à l'utilisateur de saisir les informations nécessaires.
- 3. Soumission du formulaire :**
  - L'utilisateur remplit le formulaire avec les informations requises pour le médecin et le soumet au système.
- 4. Vérification des informations :**
  - Le système vérifie que tous les champs du formulaire sont correctement remplis.
- 5. Recherche dans la base de données :**

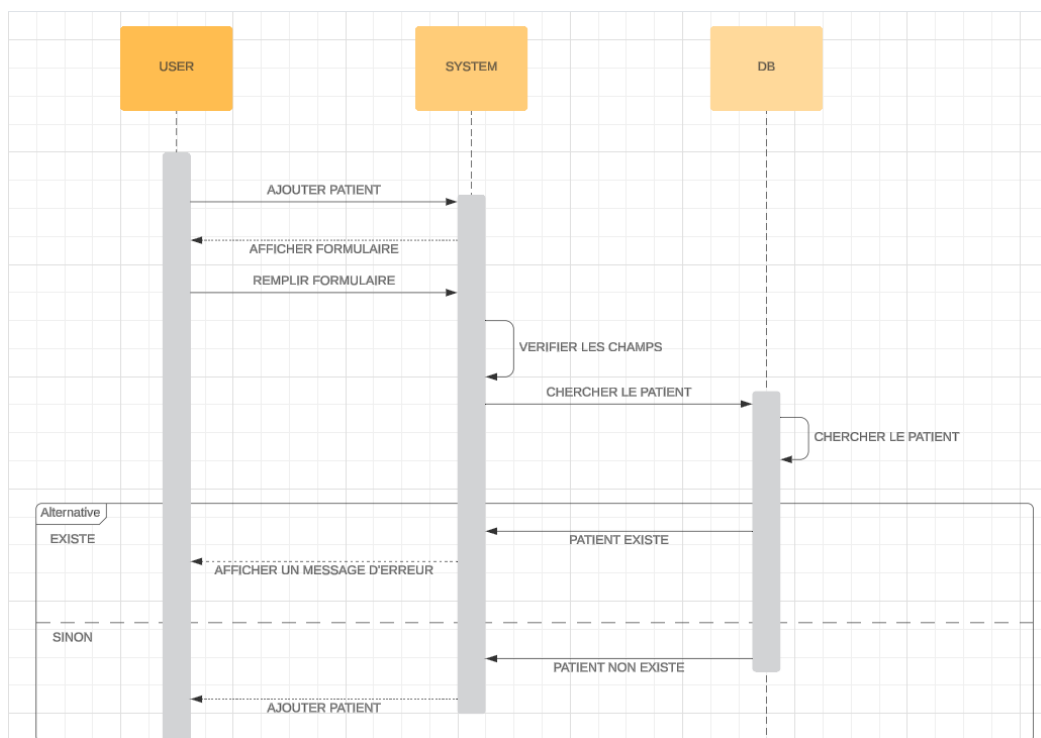
- Le système envoie une requête à la base de données pour vérifier si le médecin existe déjà.
6. **Traitement de la requête par la base de données :**
- La base de données recherche les informations du médecin et renvoie les résultats au système.
7. **Vérification de l'existence du médecin :**
- Le système analyse les résultats de la base de données :
    - **Si le médecin existe :**
      - Le système affiche un message d'erreur à l'utilisateur, indiquant que le médecin est déjà enregistré.
    - **Si le médecin n'existe pas :**
      - Le système procède à l'ajout du médecin dans la base de données et confirme l'ajout réussi.

#### Scénarios Alternatifs

- **Médecin déjà existant :**
  - Lorsque le médecin existe déjà dans la base de données, le système informe l'utilisateur de l'erreur et lui propose de réessayer avec des informations différentes.
- **Médecin inexistant :**
  - Si le médecin n'existe pas, le système ajoute les nouvelles informations du médecin à la base de données et affiche un message de confirmation de succès à l'utilisateur.

Ce diagramme de séquence présente de manière détaillée les étapes et interactions nécessaires pour ajouter un médecin dans le système, y compris les vérifications et les réponses appropriées basées sur l'existence ou non du médecin dans la base de données.

### 4.3 L'ajout pour le patient :





## Description du Processus

L'utilisateur initie le processus en envoyant une demande d'ajout de patient au système. En réponse, le système affiche un formulaire permettant à l'utilisateur de saisir les informations nécessaires pour le patient. L'utilisateur remplit ce formulaire avec les informations pertinentes, telles que le nom et le contact du patient, puis le soumet au système.

Une fois le formulaire soumis, le système vérifie que tous les champs sont correctement remplis. Ensuite, le système envoie une requête à la base de données pour vérifier si le patient existe déjà. La base de données traite cette requête, cherche les informations du patient et renvoie les résultats au système.

Le système analyse alors les résultats renvoyés par la base de données. Si le patient existe déjà, le système affiche un message d'erreur à l'utilisateur, indiquant que le patient est déjà enregistré. Si le patient n'existe pas, le système procède à l'ajout du patient dans la base de données et confirme l'ajout réussi.

## Scénarios Alternatifs

Deux scénarios alternatifs sont possibles :

### 1. Patient déjà existant :

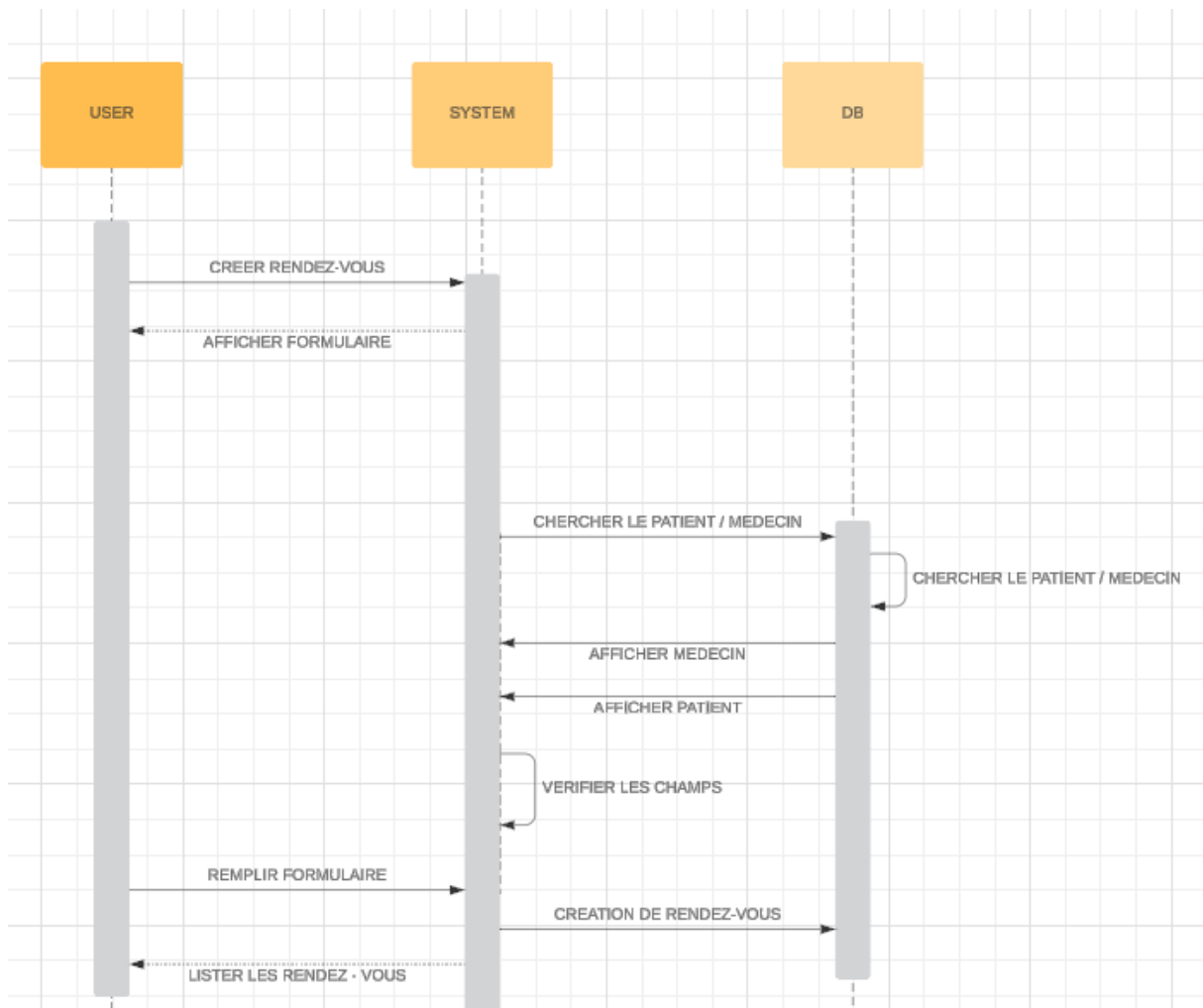
- Si le patient existe déjà dans la base de données, le système informe l'utilisateur de l'erreur et lui propose de réessayer avec des informations différentes.

### 2. Patient inexistant :

- Si le patient n'existe pas, le système ajoute les nouvelles informations du patient à la base de données et affiche un message de confirmation de succès à l'utilisateur.

Ce diagramme de séquence montre de manière claire et organisée les étapes et interactions nécessaires pour ajouter un patient dans le système, y compris les vérifications des informations soumises et la gestion des différents scénarios basés sur l'existence ou non du patient dans la base de données.

## 4.4 L'ajout du rendez-vous :



## Description du Processus

L'utilisateur initie le processus en envoyant une demande de création de rendez-vous au système. En réponse, le système affiche un formulaire permettant à l'utilisateur de saisir les informations nécessaires pour le rendez-vous, telles que la date, l'heure, le patient et le médecin.

L'utilisateur remplit ce formulaire avec les informations du rendez-vous et le soumet au système. Le système vérifie alors que tous les champs du formulaire sont correctement remplis.

Ensuite, le système envoie une requête à la base de données pour vérifier l'existence du patient et du médecin. La base de données traite cette requête, cherche les informations du patient et du médecin, puis renvoie les résultats au système.

Le système affiche les informations du médecin et du patient trouvés dans la base de données. Si les informations du patient et du médecin sont correctes, le système procède à la création du rendez-vous dans la base de données. Le système envoie alors une requête à la base de données pour enregistrer le rendez-vous.

Une fois le rendez-vous créé, le système confirme la création à l'utilisateur et affiche la liste des rendez-vous pour permettre à l'utilisateur de vérifier le nouvel ajout.

### Scénarios Alternatifs

- **Patient ou médecin non existant :**
  - Si le patient ou le médecin n'existe pas dans la base de données, le système affiche un message d'erreur et demande à l'utilisateur de vérifier les informations saisies.

Ce diagramme de séquence présente de manière claire et organisée les étapes et interactions nécessaires pour créer un rendez-vous dans le système, en incluant les vérifications nécessaires et la gestion des différents scénarios en fonction de l'existence ou non des informations du patient et du médecin dans la base de données.

## Chapitre 3 : Réalisation :

### 1- Les outils de développement :

**Django** : est un framework web open-source de haut niveau écrit en Python, conçu pour favoriser le développement rapide et une conception propre et pragmatique. Créé en 2005 par Adrian Holovaty et Simon Willison, il est devenu l'un des frameworks les plus populaires pour le développement d'applications web.



**HTML** : est le langage de balisage standard utilisé pour créer des pages web. Il structure le contenu et définit les éléments de base d'une page, tels que les titres, les paragraphes, les liens, les images, et autres composants multimédias. HTML utilise des balises pour entourer et structurer différents éléments du contenu, permettant ainsi aux navigateurs web de rendre correctement les pages.



**Css** : est un langage de feuille de style utilisé pour décrire la présentation des documents HTML ou XML. CSS contrôle l'apparence visuelle des pages web, en séparant le contenu

de la présentation, ce qui permet de maintenir et de modifier plus facilement le style des sites web.



**MySql** : est un système de gestion de base de données relationnelle (SGBDR) open-source, développé initialement par MySQL AB et maintenant maintenu par Oracle Corporation. Il utilise le langage SQL (Structured Query Language) pour gérer et manipuler les données, et est largement utilisé pour les applications web et les systèmes d'information.



**Bootstrap** : est un framework front-end open-source développé par Twitter. Il est utilisé pour créer des interfaces web et des applications mobiles responsives rapidement et efficacement. Bootstrap utilise HTML, CSS et JavaScript pour simplifier le développement en fournissant des composants pré-stylisés et des outils de mise en page.



**Xampp** : est un environnement de développement web open-source, multiplateforme et facile à installer, développé par Apache Friends. Il est conçu pour simplifier le processus de configuration et de gestion d'un serveur web local sur votre ordinateur, permettant aux développeurs de créer et de tester des applications web localement avant de les déployer sur un serveur en production.



**Vscode** : souvent abrégé en VS Code, est un éditeur de code source léger, gratuit et multiplateforme développé par Microsoft. Il est particulièrement apprécié par les développeurs pour sa polyvalence, ses performances élevées et sa richesse en fonctionnalités, en faisant l'un des éditeurs de choix pour de nombreux environnements de développement.



**StarUml** : est un outil de modélisation et de diagrammation de logiciels qui permet aux développeurs de créer des diagrammes UML (Unified Modeling Language) et d'autres types de diagrammes liés à la conception de logiciels. Il est utilisé pour visualiser, concevoir et documenter des systèmes logiciels, en utilisant une interface graphique intuitive qui facilite la création de diagrammes de classes, de diagrammes de séquence, de diagrammes d'activités, et d'autres encore.



## 2- Enchainement des interfaces :

Après avoir lancé notre projet via le terminal de commande, nous utilisons la commande suivante : **py manage.py runserver** pour démarrer notre serveur. Cela affiche l'interface d'accueil où nous avons le choix entre administrateur, docteur ou patient pour nous connecter.



Si l'utilisateur est un administrateur, l'interface affiche une navigation contenant des boutons pour se connecter ou s'inscrire, comme illustré ci-dessous :

The screenshot shows the 'Page de Connexion Administrateur' (Administrator Login Page) of the 'CABINET EMSI' application. The header bar includes the application name and navigation links for 'Admin', 'Médecin', and 'Patient'. The main content area features a title 'Page de Connexion Administrateur', two input fields for 'Nom utilisateur' and 'Mot de passe', and a 'Se Connecter' button. A link for users without an account is provided: 'Vous n avez pas de compte ? [Inscrivez-vous ici](#)'. The footer contains the copyright notice: 'Copyright © 2024 Développé par Jawad El Ibrahim et Said Titi'.

Pour créer un administrateur, vous pouvez cliquer sur "inscrivez-vous ici". Vous serez alors invité à fournir les informations nécessaires pour le créer.

The screenshot shows the 'Ajouter un Nouvel Administrateur au Cabinet' (Add New Administrator to the Cabinet) page. The header is identical to the previous page. The main content area has a title 'Ajouter un Nouvel Administrateur au Cabinet' and four input fields: 'Prénom', 'Nom de famille', 'Nom utilisateur', and 'Mot de passe'. A 'Soumettre' button is located below the first two fields. A link for existing users is provided: 'Vous avez déjà un compte ? [Connectez-vous ici](#)'. The footer contains the same copyright notice: 'Copyright © 2024 Développé par Jawad El Ibrahim et Said Titi'.

Après avoir créé un administrateur, vous pouvez ensuite cliquer sur "Se connecter" pour vous authentifier sur notre application web.

CABINET EMSI

Admin

Médecin

Patient

Page de Connexion Administrateur

Nom utilisateur

Mot de passe

Se Connecter

Vous n avez pas de compte ?[Inscrivez-vous ici](#)

Copyright © 2024 Développé par Jawad El Ibrahimy et Said Titi

L'image décrit une interface spécialisée pour l'administrateur, qui permet d'ajouter des docteurs, des patients et des rendez-vous. Chaque colonne contient des informations détaillées sur les docteurs, les patients et les rendez-vous, montrant les relations entre eux et permettant à l'administrateur de gérer efficacement ces données.

CABINET EMSI

Tableau de Bord

Médecin

Patient

Rendez-vous

Déconnexion

5

Médecins

4

Patients

4

Rendez-vous

Médecins Récents

Nom	Département	Contact	Statut
anas tayibi	médecine nucléaire	+212635399 034	En Attente
mohssin tazi	médecine palliative	+212698320 895	En Attente
yassin lel alaoui	chirurgie esthétique	+634561265 5	Permanent
hakima tanji	gastro-entérologie	+672397123 4	Permanent
saad boras	médecine palliative	+212726496 245	Permanent
khalid noury	médecine générale	+212765759 432	Permanent
hiba nsiri	médecine nucléaire	+212626924 879	Permanent

Patients Récents

Nom	Symptômes	Contact	Adresse	Statut
jawad el ibrahimi	Phlébite	+2126252 73993	el jouhary	Admis
yousra essamadi	Hépatite A	+2127623 59293	el gharabli	Admis
anour tahiri	Appendicite	+2126383 92012	agdal 2	Admis
samira el ibrahimi	Glaucome	+2126213 57919	el hussima	Admis

Pour ajouter un docteur en tant qu'administrateur, vous cliquez d'abord sur "Médecin" en haut à gauche, puis sur "Enregistrer Médecin". Ensuite, une interface s'affiche où vous pouvez saisir les informations sur le docteur.

CABINET EMSI

Tableau de Bord Médecin Patient Rendez-vous Déconnexion

**Ajouter un Nouveau**

Prénom

Nom de famille

Nom utilisateur

Mot de passe

médecine générale

Contact

Adresse

Choose File No file chosen

Inscrire

Dossier Médecin

Enregistrer Médecin

Approuver Médecin

Spécialisation Médecin

Pour ajouter un patient, vous commencez par cliquer sur "Patient" en haut à gauche, puis sur "Admettre Patient". Ensuite, une interface s'affiche où vous saisissez les informations requises pour le patient dans le menu. Pour vérifier que le patient a été créé avec succès, vous cliquez sur "Dossier Patient".

CABINET EMSI

Tableau de Bord Médecin Patient Rendez-vous Déconnexion

**Admettre un Patient au Cabinet**

Prénom

Nom de Famille

Nom utilisateur

Mot de Passe

Adresse

Contact

Symptômes

Name and Department

Choose File No file chosen

Admettre

Dossier Patient

Admettre Patient

Libérer Patient

Pour ajouter un rendez-vous, vous cliquez d'abord sur "Rendez-vous" en haut à droite, puis sur "Ajouter rendez-vous". Ensuite, une interface s'affiche où vous pouvez saisir les détails du rendez-vous à ajouter dans le Menu.



CABINET EMSI

Tableau de BordMédecinPatientRendez-vousDéconnexion

Détails de la réservation de rendez-vous

Description

Doctor Name and DepartmenPatient Name and Symptoms

Réserver

Si vous êtes un docteur, cette interface affiche la navigation pour voir vos patients ainsi que leurs rendez-vous.

CABINET EMSI

Tableau de BordPatientRendez-vousDéconnexion

2

Rendez-vous



1

Patients Sous  
Votre Charge

0

Patients Sortis

Rendez-vous Récents Pour Vous

Nom du Patient	Photo	Description	Contact	Adresse	Date
samira		20-08-2023 12:00	+212638392012	agdal 2	July 1, 2024
anour		15-07-2023 09:00	+212621357919	el hussima	July 1, 2024

© 2024 Développé par Jawad El Ibrahimy et Said Titi

Si vous êtes un patient, après vous être connecté avec les informations fournies par l'administrateur, l'interface affichera une navigation qui comprendra votre docteur assigné ainsi que vos rendez-vous avec les dates correspondantes.

## Informations sur Médecins

Champ	Données
Nom du Médecin	saad boras
Symptômes	Phlébite
Contact du Médecin	+212726496245
Département du Médecin	médecine palliative
Adresse du Médecin	Yacoub El Mansour Rabat
Date d'Admission	July 1, 2024

## Vos Rendez-vous

Nom du Médecin	Description	Date	Status
----------------	-------------	------	--------

## **Conclusion :**

La gestion des cabinets médicaux à l'ère numérique nécessite une intégration harmonieuse de technologies avancées et une compréhension approfondie des besoins des utilisateurs. Notre projet de fin d'année, en développant une application web innovante, s'efforce de répondre à ces besoins de manière efficace et sécurisée, tout en offrant une expérience utilisateur exceptionnelle.

Nous espérons que notre plateforme apportera une réelle valeur ajoutée aux praticiens et aux patients, en rendant la gestion des soins de santé plus efficace, plus sûre et plus accessible. Ce projet met en lumière l'importance de l'optimisation et de l'innovation technologique dans le domaine de la santé, soulignant les bénéfices potentiels pour la société dans son ensemble.

En conclusion, notre application web vise à transformer la gestion des cabinets médicaux en simplifiant les processus administratifs, en améliorant la communication et en sécurisant les données médicales. Cette avancée technologique pourrait servir de modèle pour d'autres projets similaires dans le futur, contribuant à l'amélioration continue des services de santé numériques.