

Rapport

Mise En Place D'une Application de gestion des patients :

Encadre par : Mohamed
Youssfi

Réalisé par : BOUCHAIB
NAIMY



SOMMAIRE

1. Technologies Serveur / BackEnd.....	3
2. Technologies Serveur /FrontEnd	3
3. La création du projet.....	4
3.1. Structure de projet.....	
3.2. Configuration de sécurité.....	
3.2. Authentification.....	
4. Les interfaces réalisées.....	
4.1. L'authentification.....	
4.2. Page principale (cote admin)	
4.3. Recherche d'un Patient	
4.4. Suppression d'un patient.....	
4.5. Pagination.....	
4.6. L'ajout d'un Patient.....	
4.7. Modification d'un Patient.....	
4.8. Les Contraintes des champs à remplir.....	
4.9. Les Contraintes des champs à remplir.....	
4.10. Interface principale (cote technicien) :.....	
4.11. Déconnexion (Log Out).....	

Résumé :

L'objectif de mon projet , présenté dans ce rapport, est la conception et la réalisation d'une application conceptuelle simple et améliorer de gestion de cabinet médical. Le processus de développement a était réaliser durant trois phases : j'ai récupéré les informations de la cabinet. La deuxième phase c'est la conception du système avec le langage de modélisation UML. La troisième phase c'est l'implémentation de mon système, ou j'ai utilisé SPRING BOOT, et l'architecture jee MYSQL , HTML, CSS, Bootstrap , THYMLEAF , LAMBOK.

Remerciement :

Je tiens à exprimer mes remerciements avec un grand plaisir et un grand respect à DIEU tout puissant et miséricordieux, notre professeur MOHAMED YOUSSEF, qui n'a cessé de me faire profiter de ses précieux conseils et remarques. Sa disponibilité et ses encouragements m'ont permis de réaliser ce travail dans les meilleures conditions.

➤ INTRODUCTION

Dans le cadre de la formation du module JEE et afin d'appliquer les méthodologies et les notions enseignées tout au long du semestre, je devais réaliser une application de gestion des patients sous le cadre d'un control.

La mission qui m'a été confiée est tout d'abord, la réalisation d'une application web dynamique de gestion des patients avec le Framework Spring Boot, en travaillant sur toutes les fonctionnalités fournies par celui-ci

- ✓ **Une gestion des dépendances Spring simplifiée**
- ✓ **Un déploiement facilité**
- ✓ **Intégrez directement Tomcat, Jetty ou Undertow (inutile de déployer des fichiers WAR)**
- ✓ **La configuration automatique de bibliothèques Spring et autres**
- ✓ **La configuration des propriétés externes plus lisible**
- ✓ **Facilités pour créer des repositories**
- ✓ **Des possibilités de déclarer des sorties JSON multiples**
- ✓ **L'exposition des ressources par REST juste avec une annotation**
- ✓ **Aucune génération de code et aucune exigence pour la configuration XML**

Spring Boot est un Framework qui permet la mise en place d'application Spring rapidement et facilement. Il se base sur le Framework Spring et permet de s'affranchir de la plupart des configurations de celui-ci à mettre en place pour créer une application. Les principales fonctions :

Java : un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très Facilement portables sur plusieurs systèmes d'exploitation

➤ Technologies Serveur / BackEnd :

➤ JPA

JPA (Java Persistence API) est un standard de la plateforme Java EE gérant la persistance des données. Il permet la gestion de correspondance entre des objets Java (entity) et une base de données.

➤ Spring Boot

Spring Boot est un framework qui permet la mise en place d'application Spring rapidement et facilement. Il se base sur le Framework Spring et permet de s'affranchir de la plupart des configurations de celui-ci à mettre en place pour créer une application.

➤ Java 8

Java 8 est la dernière version de Java et offre notamment la possibilité d'utiliser des lambda expressions, une nouvelle API de gestion de dates et heures ainsi que l'utilisation des "Optionals" qui permettent une meilleure gestion des références null.

➤ H2

H2 est un système de gestion de base de données relationnelle écrit en Java. H2 supporte un sous-ensemble du standard SQL et propose des interfaces de programmation SQL et JDBC.

➤ Structure de la couche service Rest

L'accès aux données de chaque module coté back-end se fait à travers une classe RestService. Cette couche joue le rôle d'un pont entre la «couche métier» qui va fournir les données nécessaires au client et la «couche présentation» qui va les afficher. Le format des données circulantes entre ces deux couches n'est d'autre que le format Json qui fournit un support pour une écriture simple et légère au format texte.

➤ Spring MVC

Spring MVC permet de construire des applications Web en Java. Comme son nom le suggère, il utilise le principe du Modèle/Vue/Contrôleur (MVC) en association avec le modèle IoC (Inversion ofControl) du Spring Framework. Spring MVC permet de bâtir des applications Web en se basant sur des technologies Java déjà existantes (comme les JSP pour la création de vues).

➤ La DispatcherServlet

Au cœur de Spring MVC, on trouve la DispatcherServlet. Avec Spring MVC, il n'est pas nécessaire de créer des servlets puisque le Framework la fournit déjà. Tout ce que nous avons à faire est de déclarer cette servlet dans notre fichier web.xml.

➤ Spring Security

Spring Security, est un contrôleur d'authentification flexible et puissant pour assurer une application Web Java basé sur Spring.

➤ Hibernate Framework

Est un Framework open source gérant la persistance des objets en base de données relationnelle. Hibernate est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement Java EE complet : WebSphere, JBoss Application Server et Oracle WebLogic Server

➤ Maven

Maven est un outil permettant d'automatiser la gestion de projets Java. Il offre entre autres les fonctionnalités suivantes :

- **Compilation et déploiement des applications Java (JAR, WAR)**
- **Gestion des librairies requises par l'application**
- **Exécution des tests unitaires**
- **Génération des documentations du projet (site web, pdf, Latex)**
- **Intégration dans différents IDE (Eclipse, JBuilder)**

➤ PHP

est un langage informatique utilisé sur l'internet. Le terme PHP est un acronyme récursif de "PHP: Hypertext Preprocessor". Ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que MySQL. Langage permettant de "rendre" dynamique un site web avec des échanges client/serveur

➤ SQL

Est un langage informatique normalisé servant à exploiter des bases de données relationnelles. Langage permettant d'exploiter des bases de données relationnelles. Permet de rechercher, d'ajouter, modifier ou encore supprimer des données et l'organisation des données

➤ Technologies Serveur / FrontEnd

➤ BOOTSTRAP

Un Framework développé par l'équipe du réseau social Twitter, utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Il sert à développer des sites avec un design responsive, qui s'adapte à tout type d'écran. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore.

Chapitre 1

Conception du système

2.1 Introduction :

Dans ce chapitre vas représenter la conception de mon système je vais faire un appel à la boîte à outils UML tout en respectant le processus d'unification UP à savoir : le diagramme des cas d'utilisation, le diagramme de séquences et le diagramme de classes.

2.2 Analyse de besoin

2.2.1 Définition d'UML :

UML (Unified Modeling Language) est un langage ou formalisme de modélisation orienté objet qui représente un moyen de spécifier et respecter les composants d'un système d'information. UML est un standard car à partir de 1997 il est devenu une norme de l'Object Management Group (OMG) [3].

Parmi les diagrammes d'UML :

- **Les diagrammes de structure ou statique**
- **Les diagrammes de comportements**

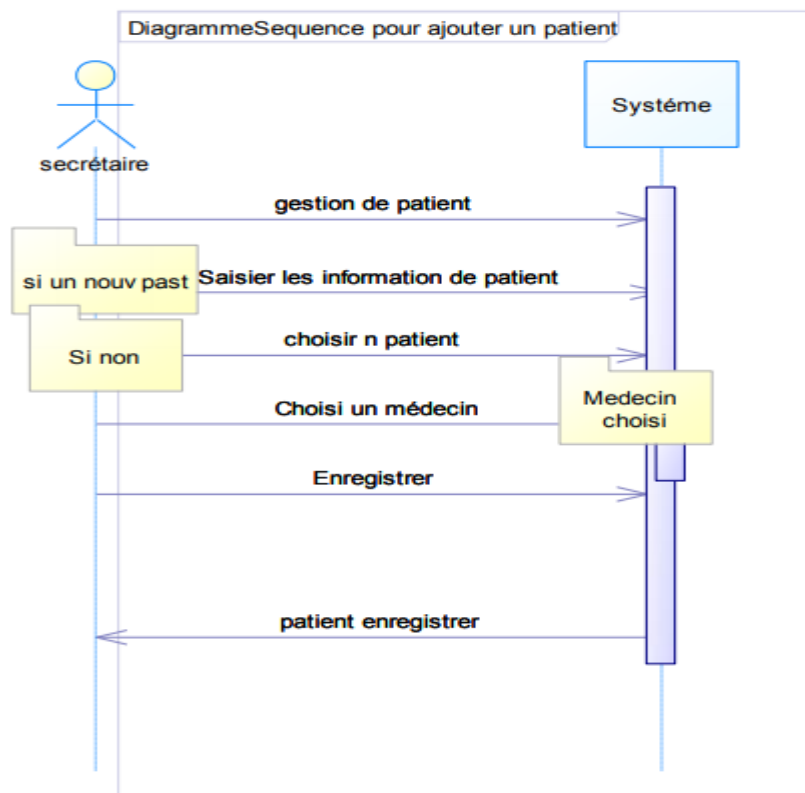
Les Diagrammes de structure ou statique :

- **les diagrammes des classes ou les diagrammes d'objet**
- **les diagrammes de composants**
- **les diagrammes de déploiements**
- **les diagrammes de structure composite**
- **les diagrammes de package**

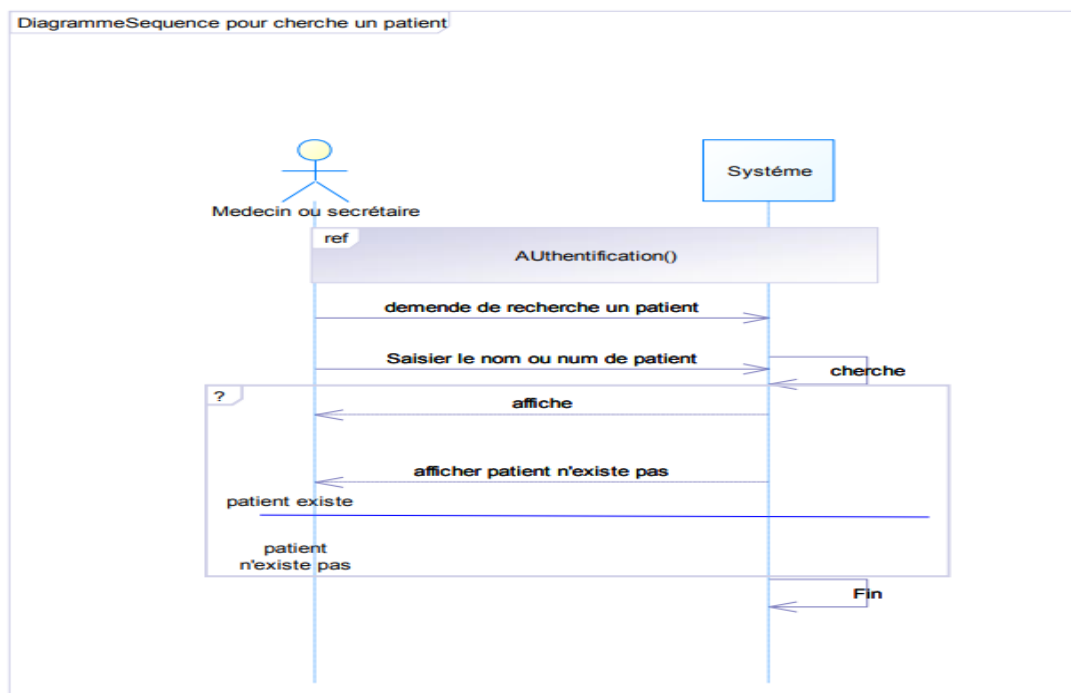
les diagrammes de comportement :

- **les diagrammes, activité ou les diagrammes de cas d'utilisation**
- **les diagrammes de machine à états**
- **les diagrammes de séquence**
- **les diagrammes de communication**
- **les diagrammes de vue d'ensemble des interactions ou les diagrammes de timing. [2]**

➤ Diagramme d'ajout d'un patient :



➤ Diagramme de recherche d'un patient :

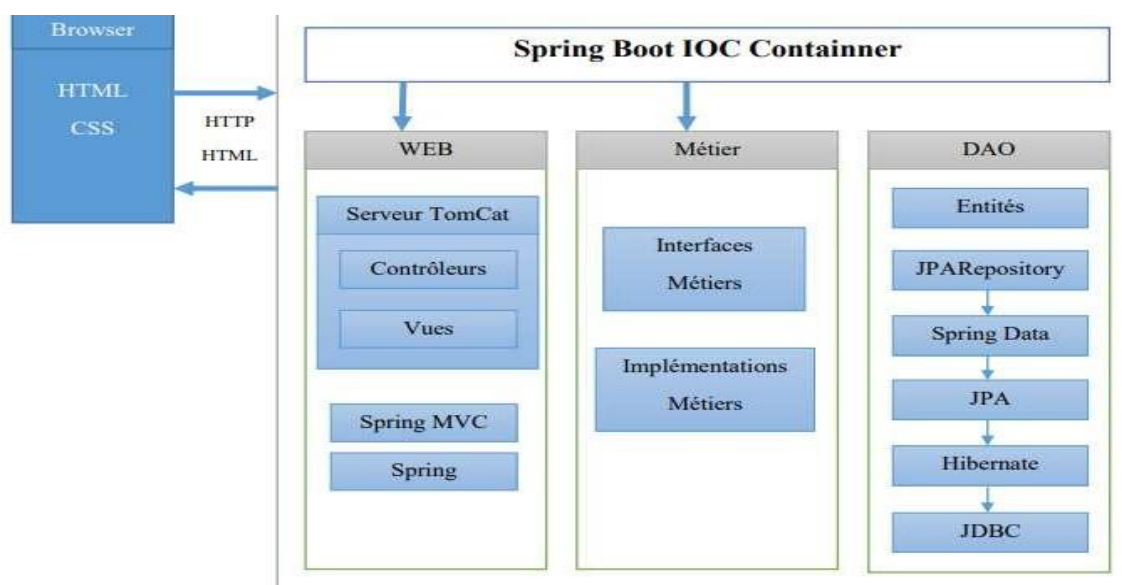


➤ La création du projet

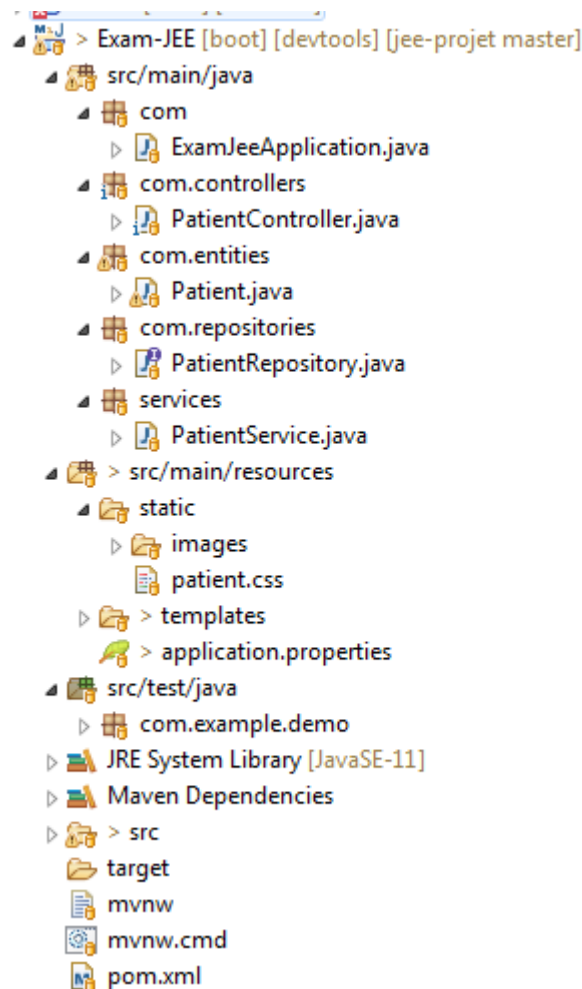
➤ Structure de projet

On peut symboliser 1ns concepts de cohésion et, de couplage dans le domaine des classes de la façon suivante : une classe a une forte cohésion si tous ses attributs sont, employés par ses méthodes et deux classes sont couplées si un changement dans l'une exige des changements dans l'autre. La cohésion forte indique que les méthodes de la classe collaborent pour accomplir une fonctionnalité, tandis que, le couplage faible indique que la classe est plutôt indépendante. Ainsi, il serait nécessaire de créer les applications en développant des composants avec forte cohésion et, faible couplage. En garantissant une forte cohésion et un faible couplage dans le développement d'une application, on gagne en qualité tout en diminuant la complexité. Bien que ce principe soit, simple à énoncer et à imaginer, il est difficile de le mettre en pratique.

La création de projet Spring Boot nécessite la mise en place d'une organisation de sa structure suivant les couches et les modules que Spring fournit. L'illustration ci-dessous montre la structure de Spring Boot :



- ✓ **IoC Container** : C'est la base la plus importante et également la base, le fondement du Spring. C'est le rôle de la configuration et de la gestion de dépendances et la gestion du cycle de vie des objets Java
- ✓ Il fait La configuration automatique pour lier les implémentations à leurs interfaces (injection de dépendances)
- ✓ **WEB** : il contient un serveur web Tomcat et l'ensemble des contrôleurs qui permet de traiter les requêtes. Ainsi, l'ensemble des vues HTML.
- ✓ **Métier** : il contient les interfaces des métiers et les implémentations.
- ✓ **DAO** : Contient toutes les classes Repository permettant l'accès à la base de données



➤ Configuration de sécurité

Spring Security est un module incontournable d'une application développée en Spring. Il apporte tout le nécessaire pour sécuriser une application et il a l'avantage d'être vraiment personnalisable. Les deux notions essentielles sont : l'authentification (savoir qui je suis) et les autorisations (savoir ce que j'ai le droit de faire). Spring Security essaye d'apporter une solution à ces deux problématiques. Pour configurer cette application j'utilise la Java Config. Cette dernière est plus qu'encouragée avec les dernières versions de Spring. La classe permettant de le créer devra être annotée par `@EnableWebSecurity` et hérité de la classe abstraite `WebSecurityConfigurerAdapter` qui fournit trois méthodes dont chacune a un rôle

➤ Authentification

La méthode configure pour gérer l'authentification : il prend comme paramètre un objet de type `AuthenticationManagerBuilder` qui permet de définir quel sera le mode d'authentification des utilisateurs et de vérifier la cohérence entre les informations saisies et celles qui se trouvent dans la base de données

API

Which of the following is NOT a JSON value?

- ☐ 'undefined'
- ☐ null
- ☐ false
- ☒ nan

✓ Correct

➤ Les interfaces réalisées

➤ **L'authentification :**

API

Lequel des éléments suivants n'est PAS un objet JSON?


- ☐ {"Name": "John", "age": {"year": "1983", "month": 10}}
- ☐ {"Name": "John", "age": "10"}
- ☐ {"Name": "John", "age": {"1983", "10"}}
- ☒ {"Name": "John", "age": 10}

✗ Incorrect

5 s

La première page qui s'affiche est la page d'authentification qui compose d'un formulaire dans lequel un utilisateur doit saisir son identification, ici c'est le nom d'utilisateur et le mot de passe. Cette page est appelée automatiquement quand un utilisateur fait appel à une requête qui nécessite une authentification. C'est le module Spring-Security qui occupe de gérer ces requêtes en utilisant la classe WebSecurityConfigurerAdapter fourni. Une fois les informations sont correctes, l'application sera redirigée vers la page d'accueil qui contient le tableau de bord. Dans le cas contraire l'application reste dans la page d'authentification en notifiant l'utilisateur par un message que les informations saisies sont incorrect.

➤ Page principale (cote admin) :

 Patients Medecins Patients ▾

Patients


ID	NOM	DATE NAISSANCE	MALADE	SCORE		
25	Naimy	2021-05-16 03:57:01.0	false	600		
26	hasan	2021-05-16 03:57:01.0	false	900		
27	moad	2021-05-16 03:57:01.0	false	100		

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 Next

L'interface principale présente la liste des patients avec ses caractéristiques qui sont le nom la date de naissance, est ce qui sont malades ou pas ainsi que leurs scores

Cette elle se compose d'un nav barre, un champ de recherche et deux boutons une pour supprimer et l'autre pour modifier un patient

➤ Recherche d'un Patient :

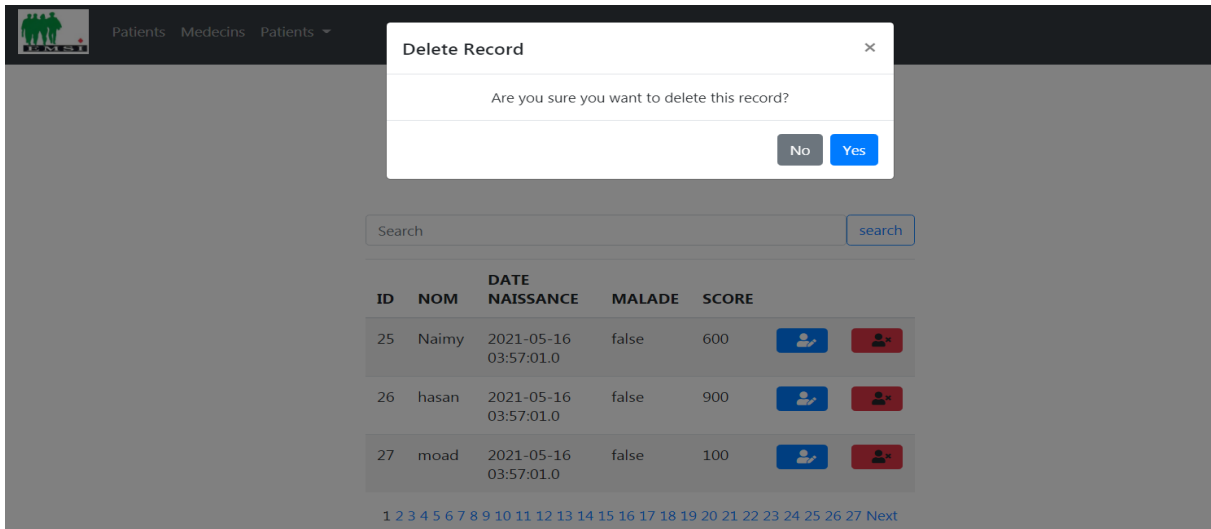
 Patients Medecins Patients ▾

Patients

ID	NOM	DATE NAISSANCE	MALADE	SCORE		
25	Naimy	2021-05-16 03:57:01.0	false	600		
26	hasan	2021-05-16 03:57:01.0	false	900		
27	moad	2021-05-16 03:57:01.0	false	100		

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 Next

➤ Suppression d'un patient :

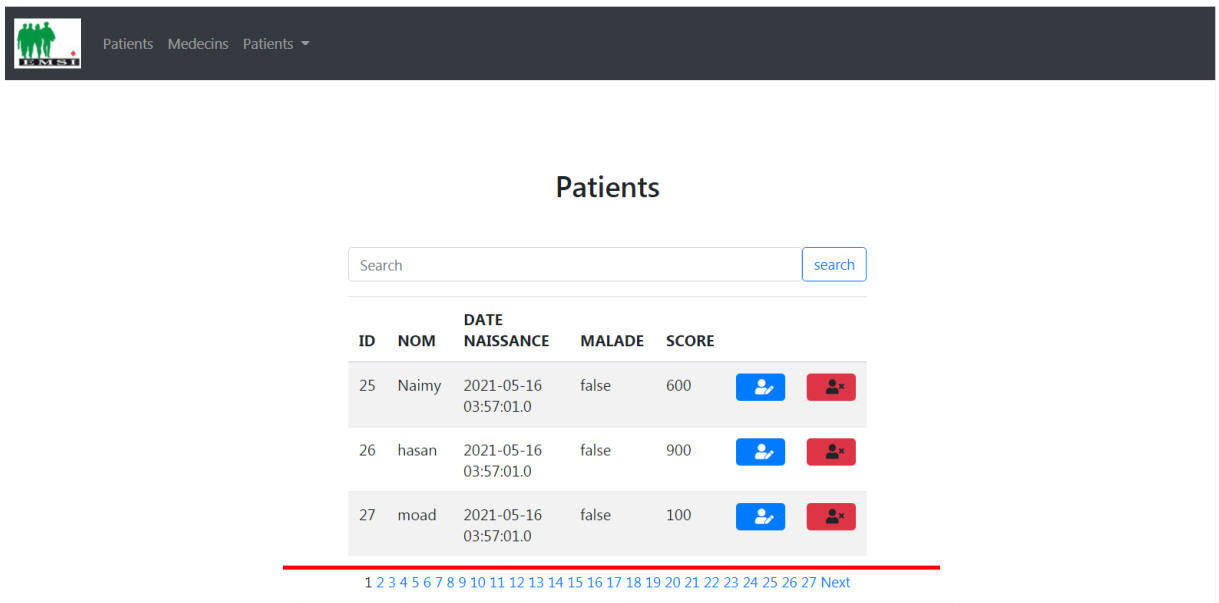


The screenshot shows a web application interface with a dark header containing a logo and navigation links: "Patients", "Medecins", and "Patients" with a dropdown arrow. A modal dialog titled "Delete Record" is open, asking "Are you sure you want to delete this record?" with "No" and "Yes" buttons. Below the modal, there is a search bar and a table of patients.

ID	NOM	DATE NAISSANCE	MALADE	SCORE		
25	Naimy	2021-05-16 03:57:01.0	false	600		
26	hasan	2021-05-16 03:57:01.0	false	900		
27	moad	2021-05-16 03:57:01.0	false	100		

Below the table is a pagination bar with numbers 1 through 27 and a "Next" link.

➤ pagination



The screenshot shows the "Patients" page of the web application. The header is dark with the logo and navigation links. The page title "Patients" is centered. Below it is a search bar and a table of patients.

ID	NOM	DATE NAISSANCE	MALADE	SCORE		
25	Naimy	2021-05-16 03:57:01.0	false	600		
26	hasan	2021-05-16 03:57:01.0	false	900		
27	moad	2021-05-16 03:57:01.0	false	100		

Below the table is a pagination bar with numbers 1 through 27 and a "Next" link, highlighted with a red border.

➤ L'ajout d'un Patient /EDITE



Ajouter Patient

Nom

Date Naissance



score

Malade ?

☐

Malade

☐

Non Malade

Submit

➤ **Les Contraintes des champs à remplir :**

➤ **Deconnexion (Log Out) :**