

DEPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

Compte rendu de l'activité pratique 7

**Technologie Web JEE : http, Servlet, JSP, MVC Spring MVC,
Thymeleaf
Parti II**

**Filière :
« Génie du Logiciel et des Systèmes Informatiques Distribués »
GLSID2**

Module : Architecture Distribuée et Middlewares

Élaboré par :

ELMAJNI Khaoula

Encadré par :

M. YOUSSEFI Mohammed

Année Universitaire : 2021-2022

Introduction

Dans ce travail on va continuer au développement web avec le framework Spring et notamment son outil Spring Boot, et on va bien illustrer l'Architecture Web et comment l'échange des données se fait, et bien précisément l'architecture Web JEE avec le design pattern Spring MVC, et on va expérimenter cette architecture avec un rendu coté serveur avec le générateur de template Thymeleaf, aussi on va initier la sécurité avec Spring Security en faisant un système d'authentification.

Description

Notre application une application qui permet de gérer des patients.

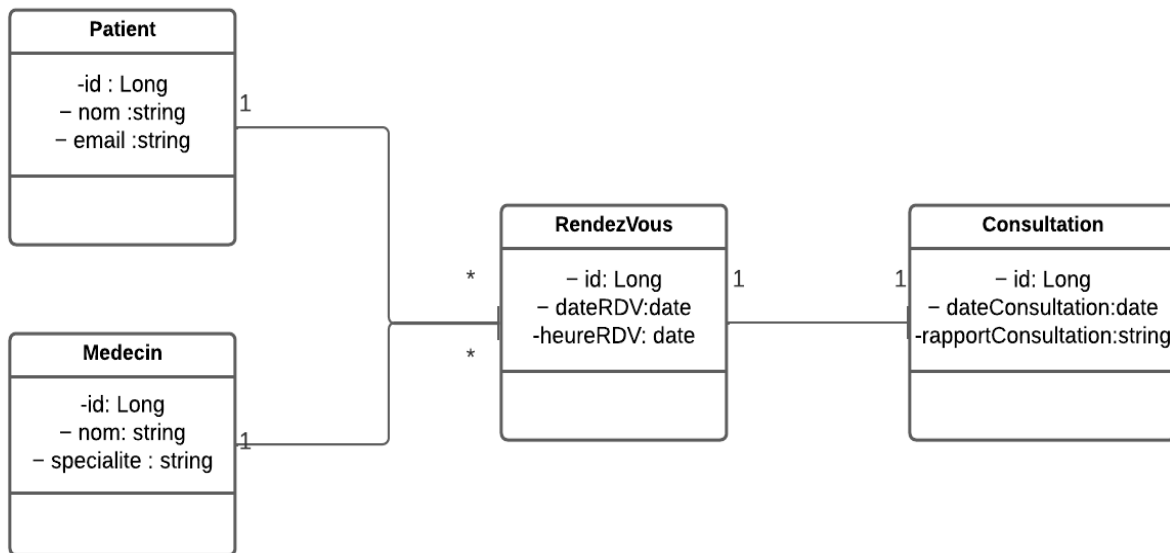
Chaque patient est défini par :

- Son ID de type Long
- Son Nom de type String
- Sa Date de naissance
- Un attribut qui indique si le patient est malade ou non
- Un score de type int

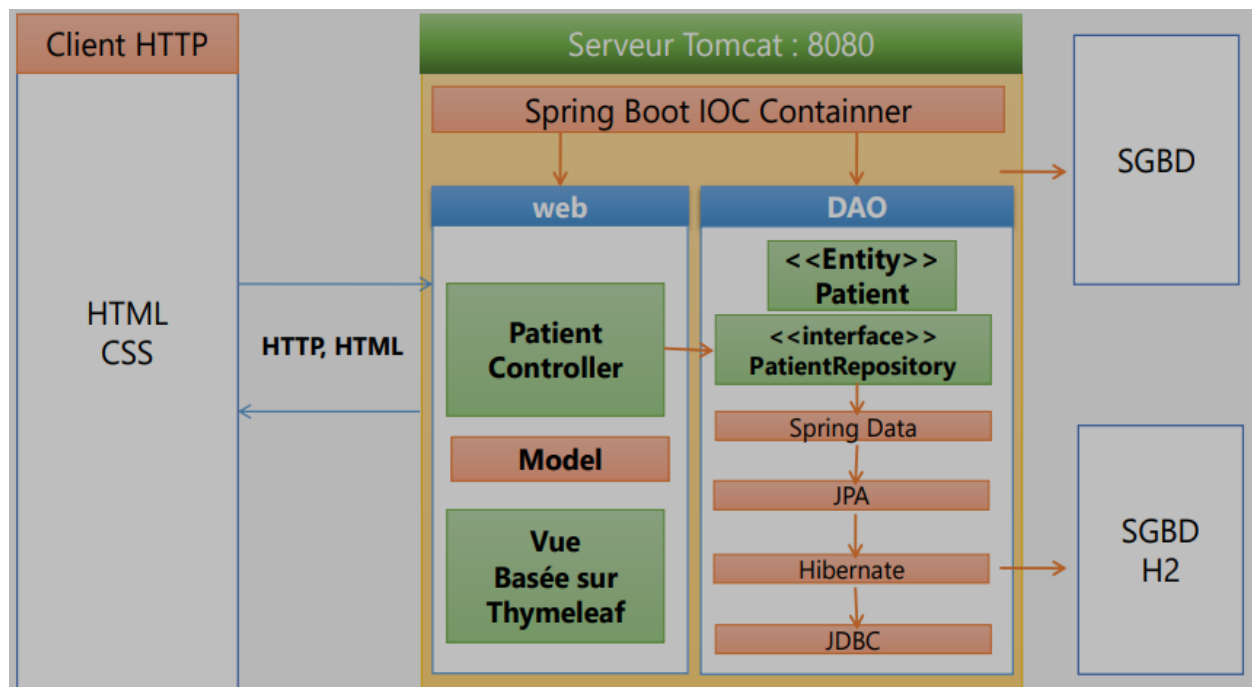
Les données sont stockées dans une base de données MySQL La couche web respecte MVC coté serveur.

```
@Entity
@Data @NoArgsConstructor
@AllArgsConstructor
public class Patient {
    @Id @GeneratedValue(strategy =
GenerationType.IDENTITY)
    private Long id;
    @NotEmpty
    @Size(min = 3, max = 50)
    private String nom;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date dateNaissance;
    private boolean malade;
    @DecimalMin("0")
    private int score;
}
```

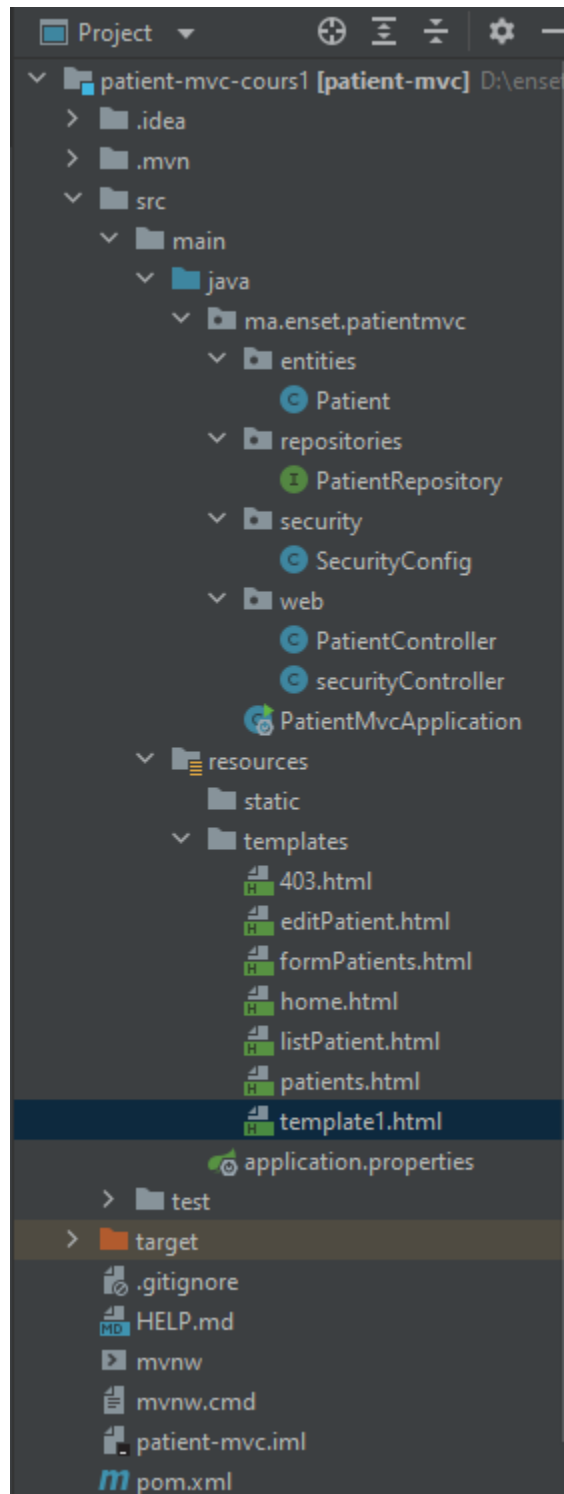
Le modèle conceptuel de données de l'application



Architecture du projet



Structure du projet



Exigences fonctionnelles

Dans cette application, on peut ;

- **Gérer des les patients :**
 1. Ajouter un patient
 2. Consulter tous les patients
 3. Consulter un patient
 4. Modifier les données d'un patient
 5. Supprimer un patient
 6. chercher les patients dont le nom contient un mot clé.
- **Gérer les médecins :**
- **Gérer les rendez-vous :**
- **Gérer les consultations :**

Template

Généralement toutes les page d'une application web partagent le même contenu html (Header, footer, menus, etc..) et pour éviter des faire des copies coller dans toutes les pages, il est important de définit une page Template qui définit

- Toutes les parties fixes de toutes les pages (header, footer, menus, etc...)
- Déclarer les sections qui changeront de contenue en fonction de chaque page.

Alors : Thymeleaf Layout Dialect Maven Dependency Pour Spring Boot

```
<dependency>  
  <groupId>nz.net.ultraq.thymeleaf</groupId>  
  <artifactId>thymeleaf-layout-dialect</artifactId>  
</dependency>
```

Créer une page html 'template':

En ajoutant le dialect thymeleaf :

```
xmlns:th="http://www.thymeleaf.org"  
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"  
ajouter la section dans la page template:
```

```
<section layout:fragment="content1">  
</section>
```

Puis on va utiliser cette template dans nos pages, en ajoutant aussi le dialect de thymeleaf:

```
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"  
layout:decorate="template1"
```

cette page je vais la décorer avec cette template.

Mettre le contenu des pages à l'intérieur d'une section appelée 'content1', comme j'indique au thymeleaf va chercher une partie qui s'appelle 'content1' puis placer ce code dedant.

```
<div layout:fragment="content1">  
</div>
```

La validation

Pour faire la validation 4 choses à faire:

1- Ajouter la dependance

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-  
validation</artifactId>  
</dependency>
```

2- Ajouter les annotations de validation

```
@NotEmpty  
@Size(min = 3, max = 50)  
private String nom;  
@DecimalMin("0")  
private int score;
```

3- Au niveau de controleur, ajouter l'annotation de @Valid et l'objet

BindingResult pour stocker les erreurs

4- Au niveau de html on utilise 'th:errors'

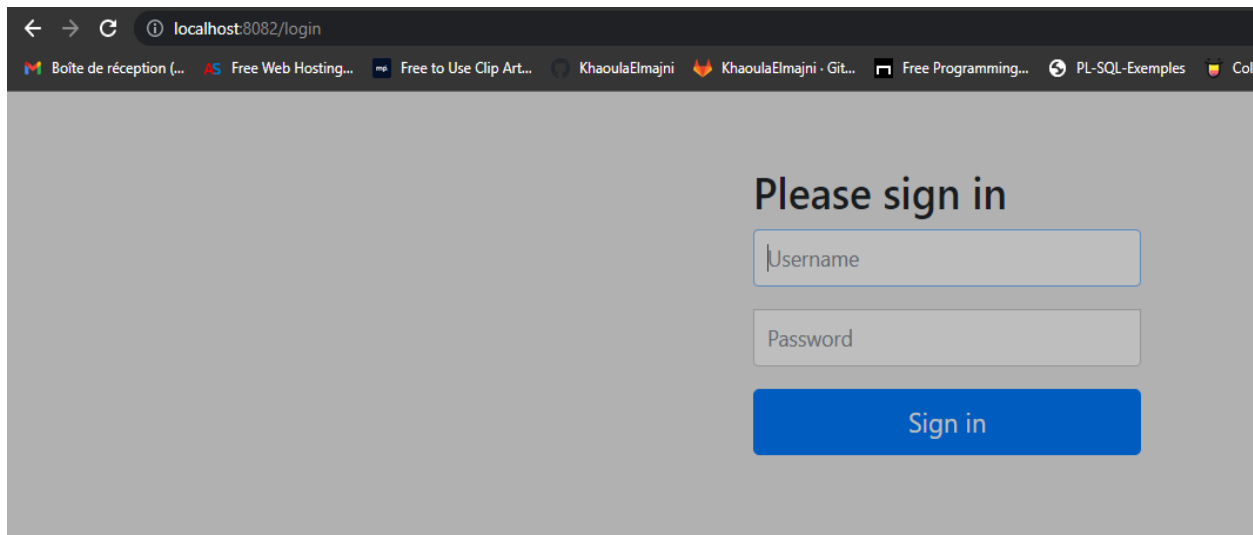
```
<span th:errors="${patient.nom}" class="text-  
danger"></span>
```

La partie de la sécurité avec Spring Security:

Ajouter un système d'authentification, alors d'abord il faut ajouter la dépendance:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Après le 1er redémarrage de l'application, Spring va exécuter la configuration par défaut:



Ici les requêtes sont reçues tout d'abord par Spring Security Filter, ensuite vers le servlet.

On peut changer la configuration par défaut, et utiliser une configuration personnalisée adaptée avec nos besoins.

On va créer une classe de configuration personnalisée, qui hérite de 'WebSecurityConfigurerAdapter'

```
@Configuration
@EnableWebSecurity
public class SecurityConfig extends
WebSecurityConfigurerAdapter {
    @Override
    protected void configure(AuthenticationManagerBuilder
auth) throws Exception {
        super.configure(auth);
    }
    @Override
```

```
protected void configure(HttpSecurity http) throws  
Exception {  
    super.configure(http);  
}  
}
```

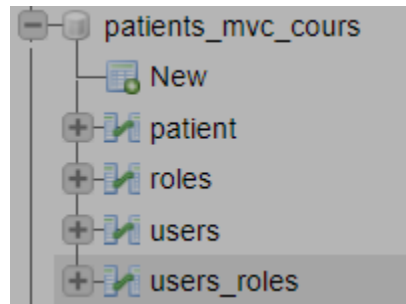
la gestion contextuelle:

la gestion des roles et ses fonctionnalités:

```
<dependency>  
    <groupId>org.thymeleaf.extras</groupId>  
    <artifactId>thymeleaf-extras-  
springsecurity5</artifactId>  
</dependency>
```













Authentification avec des données de la base de données:

On crée les tables suivantes qui seront utilisés dans l'authentification par la suite:



	username	password	active
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	admin	\$2a\$10\$0QdaeCVa4mCTg.wi/hQ2oeZZ6hczsJUGYBGIlulyK9bc...	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	user1	\$2a\$10\$0QdaeCVa4mCTg.wi/hQ2oeZZ6hczsJUGYBGIlulyK9bc...	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	user2	\$2a\$10\$0QdaeCVa4mCTg.wi/hQ2oeZZ6hczsJUGYBGIlulyK9bc...	1

	role
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	ADMIN
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	USER

					username	role
<input type="checkbox"/>	 Edit	 Copy	 Delete	admin	ADMIN	
<input type="checkbox"/>	 Edit	 Copy	 Delete	admin	USER	
<input type="checkbox"/>	 Edit	 Copy	 Delete	user1	USER	
<input type="checkbox"/>	 Edit	 Copy	 Delete	user2	USER	

Les vues/interfaces de notre application

localhost:8082/user/index

Boîte de réception (L... Free Web Hosting... Free to Use Clip Art... KhaoulaElmajni KhaoulaElmajni - Git... Free Programming... PL-SQL-Exemples Color Palettes for D... Learn PL/SQL Tutori... Gratuit PowerPoint...

Home Médecins Rendez-vous Consultations Patients

admin

liste des patients

Chercher un patient : Nom du patient

ID	Nom	Date de Naissance	Malade	Score	Action
3	Ali	2022-03-28	false	9	
4	Zakaria	2022-03-28	true	56	
5	khaoula	2022-03-28	false	12	
6	Ahmed	2022-03-28	true	102	
7	Alialia	2022-03-28	false	9	

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

f t G @ in

© 2020 Copyright: Khaoula

localhost:8082/admin/formPatients

Boîte de réception (L... Free Web Hosting... Free to Use Clip Art... KhaoulaElmajni KhaoulaElmajni - Git... Free Programming... PL-SQL-Exemples Color Palettes for D... Learn PL/SQL Tutori... Gratuit PowerPoint...

Home Médecins Rendez-vous Consultations Patients

user1

Ajouter un nouveau patient

Nom

Date de Naissance

mm/dd/yyyy

Malade ☐

Score

0

f t G @ in

© 2020 Copyright: Khaoula

localhost/admin/editPatient?id=6&keyword=&page=0

Boîte de réception Free Web Hosting... Free to Use Clip Art... KhaoulaElmajni KhaoulaElmajni - Git... Free Programming... PL-SQL-Exemples Color Palettes for D... Learn PL/SQL Tutori... Gratuit PowerPoint...

Home Médecins Rendez-vous Consultations Patients user1

Modifier patient Ahmed

ID 6

Nom
Ahmed

Date de Naissance
03/28/2022

Malade ☒

Score
102

+ -

f t G i in

© 2020 Copyright: Khaoula

localhost/admin/editPatient?id=98&keyword=&page=18

Boîte de réception Free Web Hosting... Free to Use Clip Art... KhaoulaElmajni KhaoulaElmajni - Git... Free Programming... PL-SQL-Exemples Color Palettes for D... Learn PL/SQL Tutori... Gratuit PowerPoint...

Home Médecins Rendez-vous Consultations Patients user1

Modifier patient jawad

ID 98

Nom
jawad

Date de Naissance
10/08/2021

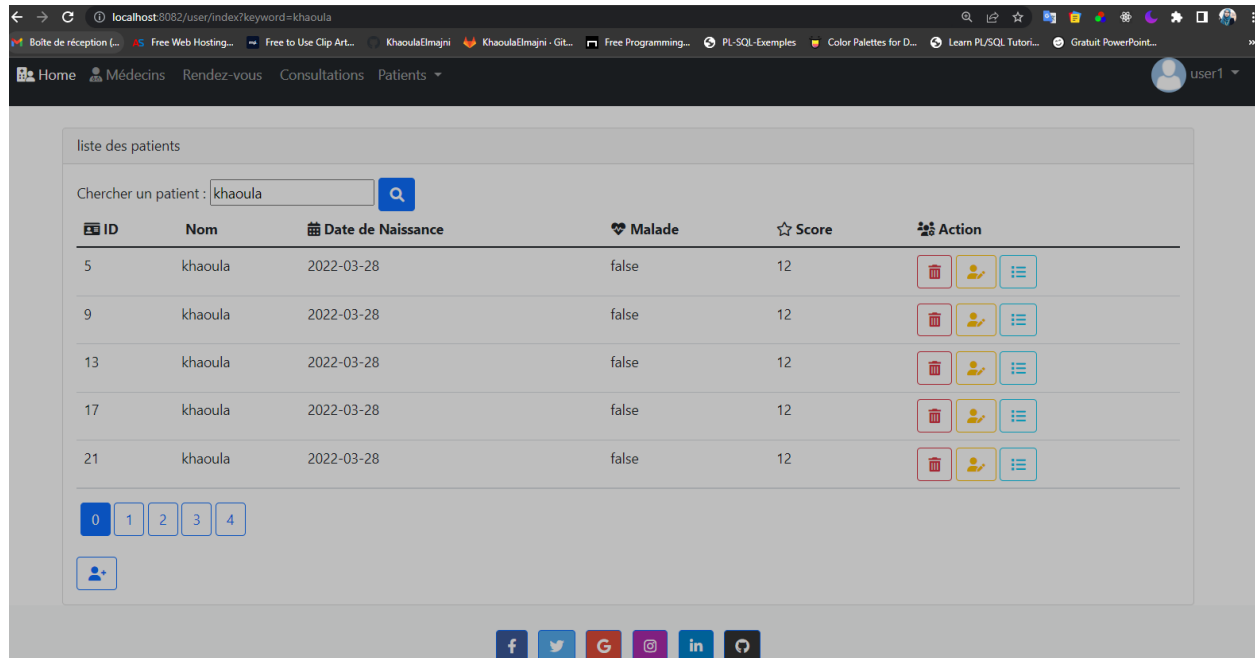
Malade ☒

Score
58

+ -

f t G i in

© 2020 Copyright: Khaoula



Conclusion

Ce travail nous a permis de bien connaître l'architecture web, et on a initié avec l'architecture web JEE et le design pattern Spring MVC, et on a travaillé avec le rendu coté serveur avec le moteur de template thymeleaf, aussi l'authentification avec Spring Security.