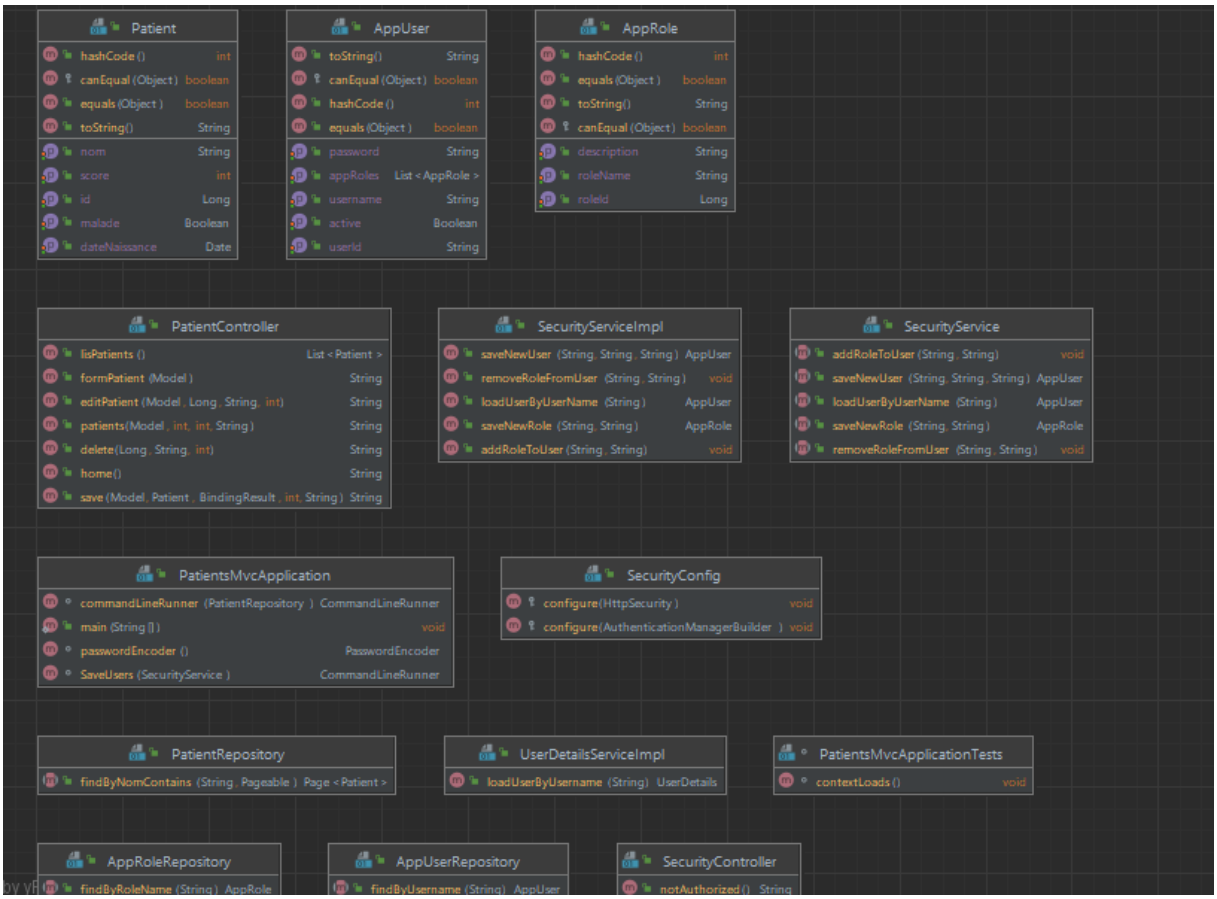
Activité pratique 4

Partie Spring Security

Objectif :

Sécuriser l'accès à l'application qui permet de gérer les patients en utilisant spring security . l'accès nécessite un username et un mot de passe.

Architecture globale



1- Ajouter la dépendance Maven de spring Security

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

2- In memory authentification :

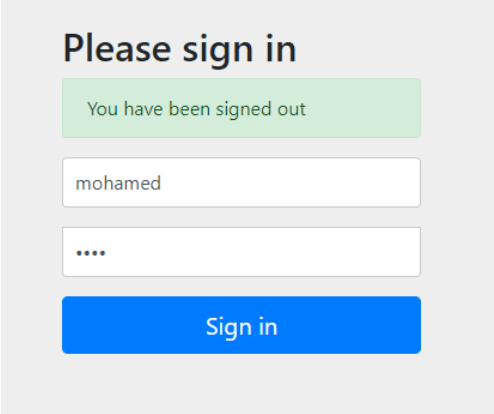
Rôle user et rôle admin

```
String encodedPWD=passwordEncoder.encode( rawPassword: "1234");
System.out.println(encodedPWD);
auth.inMemoryAuthentication().withUser( username: "user1").password(encodedPWD).roles("USER");
auth.inMemoryAuthentication().withUser( username: "user2").password(passwordEncoder.encode( rawPassword: "1111")).roles("USER");
auth.inMemoryAuthentication().withUser( username: "admin").password(passwordEncoder.encode( rawPassword: "2345")).roles("USER","ADMIN");
```

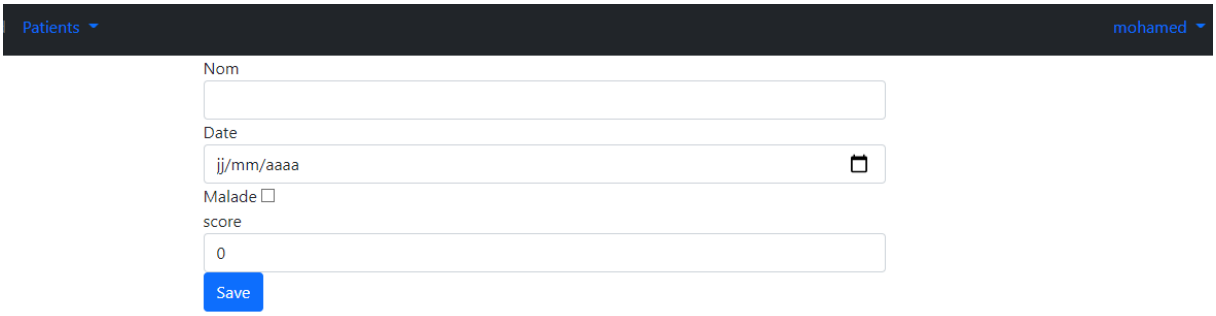
3- JDBC authentification :

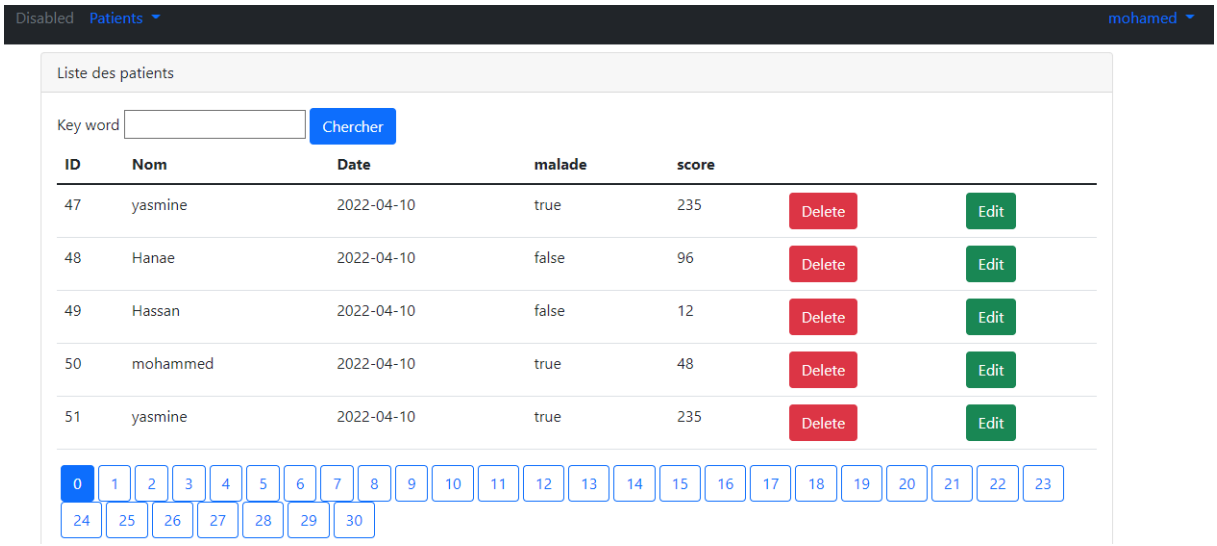
```
auth.jdbcAuthentication()
        .dataSource(dataSource)
        .usersByUsernameQuery("select username as principal, password as credentials, active from users where username=?")
        .authoritiesByUsernameQuery("select username as principal, role as role from users_roles where username=?")
        .rolePrefix("ROLE_")
        .passwordEncoder(passwordEncoder);
```

4- UserDetailsService authentification :



Modification des informations





Suppression d'un patient