

Atelier 3 :Spring mvc , thymeleaf , Spring Data

1. Introduction :

L'objectif est de créer une application web JEE basée sur spring mvc , spring data jpa , et thymeleaf , qui permet de gérer les patients.

2. Enoncé :

Partie 1 :

- Afficher patients
- Faire la pagination
- Chercher les patients
- Supprimer un patient

Partie 2 :

- Créer une page template basée sur thymeleaf layout
- Saisir et Ajouter Patients
- Faire la validation du formulaire
- Editer et mettre à jour un patient

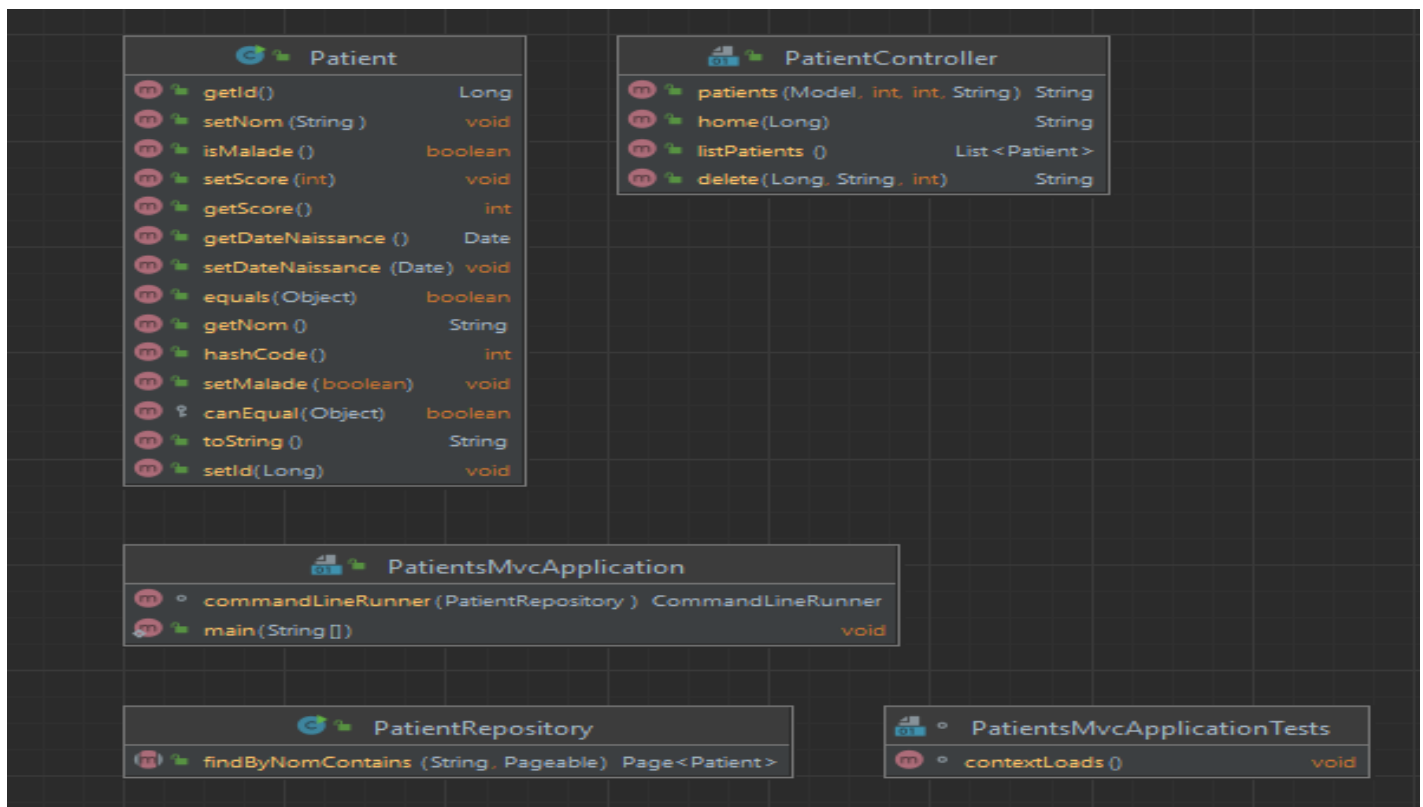
Partie 3 & 4 :

- Ajouter la dépendance Maven de Spring Security
- Personnaliser la configuration de Spring Security pour ajouter les restrictions suivantes avec la stratégie InMemoryAuthentication :
 - Authentification avec le rôle USER pour pouvoir consulter les Patients. - Authentification avec le rôle ADMIN pour pouvoir Ajouter, Editer, mettre à jour et supprimer des Patients.

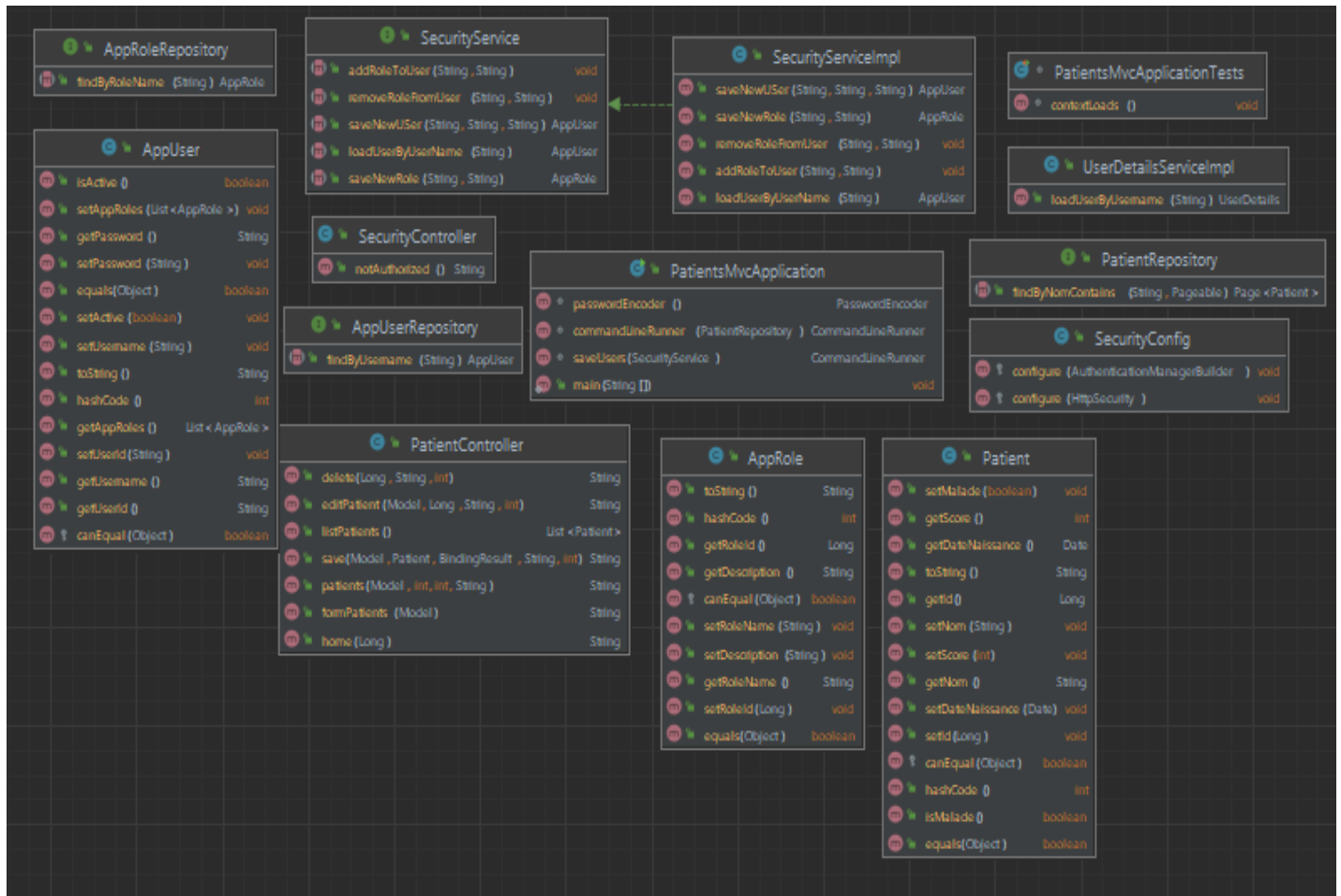
- Permettre l'accès aux ressources statiques (/webjars/**) sans aucune authentification.
- Basculer de la stratégie InMemoryAuthentication vers JDBCAuthentication
- Basculer vers la stratégie UserDetailsService

3. Conception et architecture :

- Partie 1 & 2 :



• Partie 3 & 4 :



4. Code Source :

- Github Repository :

<https://github.com/felgarti/atelier3-SpringMVC-Thymeleaf-SpringData.git>

5. Captures d'écran :

- Partie 1 :

○ afficher :

Liste des patients					
Keyword		<input type="text"/>			
		<input type="button" value="Chercher"/>			
ID	Nom	Date de naissance	Malade	Score	
2	Hamza	2022-04-10 21:37:33.0	true	18	<input type="button" value="Delete"/>
3	Manal	2022-04-10 21:37:33.0	false	17	<input type="button" value="Delete"/>
4	Yousra	2022-04-10 21:37:33.0	true	16	<input type="button" value="Delete"/>
6	Hamza	2022-04-10 21:38:03.0	true	18	<input type="button" value="Delete"/>
7	Manal	2022-04-10 21:38:04.0	false	17	<input type="button" value="Delete"/>
<div><input type="button" value="0"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> <input type="button" value="6"/> <input type="button" value="7"/> <input type="button" value="8"/> <input type="button" value="9"/> <input type="button" value="10"/> <input type="button" value="11"/> <input type="button" value="12"/> <input type="button" value="13"/> <input type="button" value="14"/> <input type="button" value="15"/> <input type="button" value="16"/> <input type="button" value="17"/> <input type="button" value="18"/> <input type="button" value="19"/> <input type="button" value="20"/></div>					

○ Pagination :

Liste des patients					
Keyword		<input type="text"/>			
		<input type="button" value="Chercher"/>			
ID	Nom	Date de naissance	Malade	Score	
19	Manal	2022-04-10 21:46:28.0	false	17	<input type="button" value="Delete"/>
20	Yousra	2022-04-10 21:46:28.0	true	16	<input type="button" value="Delete"/>
21	Fatima	2022-04-10 21:48:48.0	false	15	<input type="button" value="Delete"/>
22	Hamza	2022-04-10 21:48:48.0	true	18	<input type="button" value="Delete"/>
23	Manal	2022-04-10 21:48:48.0	false	17	<input type="button" value="Delete"/>
<div><input type="button" value="0"/> <input type="button" value="1"/> <input type="button" value="2"/> <input checked="" type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> <input type="button" value="6"/> <input type="button" value="7"/> <input type="button" value="8"/> <input type="button" value="9"/> <input type="button" value="10"/> <input type="button" value="11"/> <input type="button" value="12"/> <input type="button" value="13"/> <input type="button" value="14"/> <input type="button" value="15"/> <input type="button" value="16"/> <input type="button" value="17"/> <input type="button" value="18"/> <input type="button" value="19"/> <input type="button" value="20"/></div>					

○ chercher :

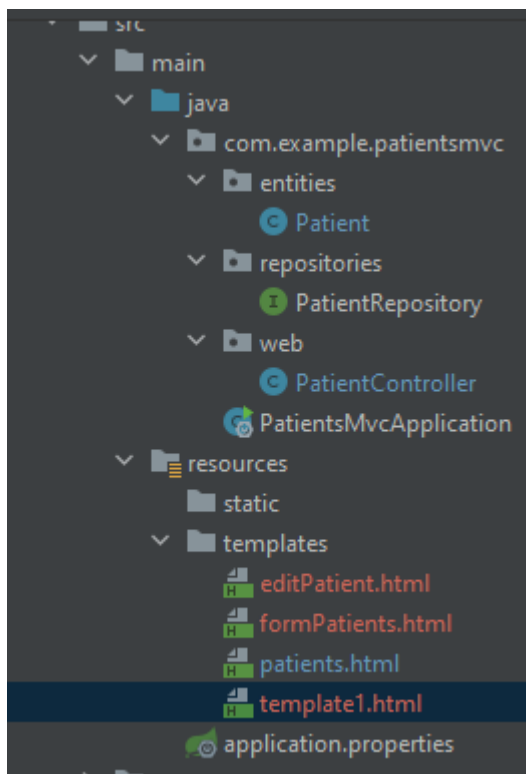
Keyword		<input type="text" value="fatima"/>			
		<input type="button" value="Chercher"/>			
ID	Nom	Date de naissance	Malade	Score	
13	Fatima	2022-04-10 21:40:25.0	false	15	<input type="button" value="Delete"/>
17	Fatima	2022-04-10 21:46:28.0	false	15	<input type="button" value="Delete"/>
21	Fatima	2022-04-10 21:48:48.0	false	15	<input type="button" value="Delete"/>
25	Fatima	2022-04-10 21:50:45.0	false	15	<input type="button" value="Delete"/>
29	Fatima	2022-04-10 21:53:56.0	false	15	<input type="button" value="Delete"/>
<div><input type="button" value="0"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/></div>					

- **supprimer (exp: ID= 2) :**

Keyword	<input type="text"/>	<input type="button" value="Chercher"/>			
ID	Nom	Date de naissance	Malade	Score	
3	Manal	2022-04-10 21:37:33.0	false	17	<input type="button" value="Delete"/>
4	Yousra	2022-04-10 21:37:33.0	true	16	<input type="button" value="Delete"/>
6	Hamza	2022-04-10 21:38:03.0	true	18	<input type="button" value="Delete"/>
7	Manal	2022-04-10 21:38:04.0	false	17	<input type="button" value="Delete"/>
8	Yousra	2022-04-10 21:38:04.0	true	16	<input type="button" value="Delete"/>

- **Partie 2 :**

- **Folder structure :**



○ Ajouter :

Nom

patientajouté2

Date de naissance

jj/mm/aaaa



Malade ☒

Score

18

Enregistrer

Liste des patients

Keyword

Chercher

ID	Nom	Date de naissance	Malade	Score	
145	patientajouté	2022-04-05	false	12	Delete

01234567891011121314151617181920212223

2425262728

Nom

ne doit pas être vide

la taille doit être comprise entre 4 et 40

Date de naissance

jj/mm/aaaa



Malade ☐

Score

0

Enregistrer

○ Edit :

ID

13

Nom

Fatima modifiée

Date de naissance

10/04/2022



Malade ☐

Score

15

Enregistrer

Liste des patients

Keyword

fatima

Chercher

ID	Nom	Date de naissance	Malade	Score		
13	Fatima modifiée	2022-04-10	false	15	Delete	Edit
17	Fatima	2022-04-10	false	15	Delete	Edit
21	Fatima	2022-04-10	false	15	Delete	Edit
25	Fatima	2022-04-10	false	15	Delete	Edit
29	Fatima	2022-04-10	false	15	Delete	Edit

0 1 2 3 4 5 6 7 8 9 10

• Navbar template 1 :

Home Link Patients ▾

Nouveau

Chercher

- **Partie 3 & 4 : Spring security**
 - **Entities : App Role et appUser :**

```

13 usages  felgarti
@Entity
@NoArgsConstructor
@Data
public class AppRole {
    @Id@GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long roleId ;
    @Column(unique = true)
    private String roleName
    ;
    private String description ;
}

```

```

@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class AppUser {
    @Id
    private String userId ;
    @Column(unique = true)
    private String username ;
    private String password ;
    private boolean active ;
    @ManyToMany(fetch = FetchType.EAGER)
    private List<AppRole> appRoles= new ArrayList<>() ;
}

```

- **Repository : AppRoleRepository**

```

2 usages  felgarti*
public interface AppRoleRepository extends JpaRepository<AppRole, Long> {
    3 usages  felgarti
    AppRole findByRoleName(String roleName ) ;
}

```

- **Repository : UserRoleRepository**

```

public interface AppUserRepository extends JpaRepository<AppUser, String> {
    3 usages  felgarti
    AppUser findByUsername(String username ) ;
}

```


- Service : SecurityService

```
1 implementation  felgati
public interface SecurityService {
    1 implementation  felgati
    AppUser saveNewUser(String username , String password , String verifyPassword );
    1 implementation  felgati
    AppRole saveNewRole(String rolename , String description ) ;
    //27
    1 implementation  felgati
    void addRoleToUser(String username , String rolename ) ;
    1 implementation  felgati
    void removeRoleFromUser(String username , String rolename ) ;
    1 usage 1 implementation  felgati
    AppUser loadUserByUserName(String username) ;
}
```

- Service : SecurityServiceImpl

```
@Service
@Slf4j
@AllArgsConstructor
@Transactional
public class SecurityServiceImpl implements SecurityService {
    5 usages
    private AppUserRepository appUserRepository ;
    4 usages
    private AppRoleRepository appRoleRepository ;
    1 usage
    private PasswordEncoder passwordEncoder ;
    felgati
    @Override
    public AppUser saveNewUser(String username, String password, String verifyPassword) {
        if(!password.equals(verifyPassword)) throw new RuntimeException(" passwords not matching ") ;
        String hashedPWD= passwordEncoder.encode(password) ;
        AppUser appUser = new AppUser() ;
        appUser.setUserId(UUID.randomUUID().toString());
        appUser.setUsername(username);
        appUser.setPassword(hashedPWD);
        appUser.setActive(true);
        AppUser saved =appUserRepository.save(appUser) ;

        return saved;
    }
}
```

```

@Override
public AppRole saveNewRole(String rolename, String description) {
    AppRole appRole = appRoleRepository.findByRoleName(rolename) ;
    if(appRole!=null) {
        new RuntimeException("Role "+rolename + " already exists .") ;
    }
    appRole= new AppRole() ;
    appRole.setRoleName(rolename);
    appRole.setDescription(description);
    appRoleRepository.save(appRole) ;
    return appRole;
}

```

```

@Override
public void addRoleToUser(String username, String rolename) {
    AppUser appUser = appUserRepository.findByUsername(username) ;
    if(appUser!=null) {
        new RuntimeException("User "+username + " not found .") ;
    }
    AppRole appRole= appRoleRepository.findByRoleName(rolename) ;
    if(appRole!=null) {
        new RuntimeException("Role "+rolename + " not found .") ;
    }
    appUser.getAppRoles().add(appRole) ;
    appUserRepository.save(appUser) ;
}

```

```

@Override
public void removeRoleFromUser(String username, String rolename) {
    AppUser appUser = appUserRepository.findByUsername(username);
    if(appUser!=null) {
        new RuntimeException("User "+username + " not found .") ;
    }
    AppRole appRole= appRoleRepository.findByRoleName(rolename) ;
    if(appRole!=null) {
        new RuntimeException("Role "+rolename + " not found .") ;}
    appUser.getAppRoles().remove(appRole) ;
}

1 usage  felgarti *
@Override
public AppUser loadUserByUsername(String username) {
    return appUserRepository.findByUsername(username) ;}

```

- Service : UserDetailsServiceImpl

```

@Service
public class UserDetailsServiceImpl implements UserDetailsService {

    2 usages
    private SecurityService securityService ;

    felgarti
    public UserDetailsServiceImpl(SecurityService securityService) {
        this.securityService = securityService;
    }

    felgarti *
    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        AppUser appUser= securityService.loadUserByUsername(username) ;
        Collection<GrantedAuthority> authorities1=appUser.getAppRoles().stream().map(role->
            new SimpleGrantedAuthority(role.getRoleName())).collect(Collectors.toList());
        User user = new User(appUser.getUsername() , appUser.getPassword(), authorities1) ;
        return user;
    }
}

```

- SecurityConfig :

```

@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    @Autowired
    private DataSource dataSource;

    @Autowired
    private UserDetailsServiceImpl userDetailsService;

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.userDetailsService(userDetailsService);
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.formLogin();
        http.authorizeRequests().antMatchers("/admin/**").hasAuthority("ADMIN");
        http.authorizeRequests().antMatchers("/user/**").hasAuthority("USER");
        http.authorizeRequests().antMatchers("/user/**").hasAuthority("ADMIN");
        http.authorizeRequests().antMatchers("/").permitAll();
        http.authorizeRequests().antMatchers("/webjars/**").permitAll();
        http.authorizeRequests().anyRequest().authenticated();
        http.exceptionHandling().accessDeniedPage(accessDeniedUrl: "/403");
    }
}

```

- **MVC APP : les users créés**

```

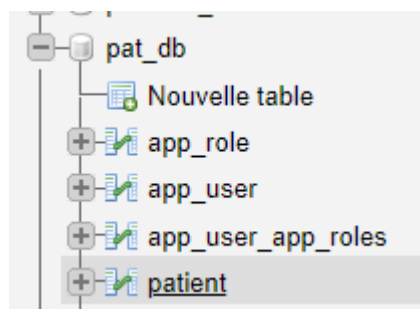
@Bean CommandLineRunner saveUsers(SecurityService securityService){
    return args -> {
        securityService.saveNewUser("fatima" , "1234","1234") ;
        securityService.saveNewUser("yousra" , "1234","1234") ;
        securityService.saveNewRole("USER" , " une desc ") ;
        securityService.saveNewRole("ADMIN" , "" );
        securityService.addRoleToUser("fatima" , "ADMIN");
        securityService.addRoleToUser("yousra" , "USER");

    } ;
}

@Bean
PasswordEncoder passwordEncoder()
{
    return new BCryptPasswordEncoder();
}

```

- **DataBase :**



- **Login :**

- **ADMIN : exp : fatima**

The login form has a title 'Please sign in'. It contains two input fields: the first is for the username, containing the text 'fatima'; the second is for the password, masked with four dots '....'. Below the fields is a blue button labeled 'Sign in'.

- USER : exp : yousra

6. Bonus :

- recherche selon 2 critères : nom et score

Keyword

Chercher

ID	Nom	Date de naissance	Malade	Score		
1	Fatima	2022-04-26	false	15	Delete	Edit
5	Fatima	2022-04-26	false	15	Delete	Edit
9	Fatima	2022-04-26	false	15	Delete	Edit
13	Fatima	2022-04-26	false	15	Delete	Edit
17	Fatima	2022-04-26	false	15	Delete	Edit

0

1

2

3

4

5

6

7

Keyword Chercher

ID	Nom	Date de naissance	Malade	Score		
144	Hamza	2022-04-27	true	18	Delete	Edit
148	Hamza	2022-04-27	true	18	Delete	Edit
151	Hassan	2022-04-27	false	18	Delete	Edit
152	Hamza	2022-04-27	true	18	Delete	Edit

0

1

2

3

4

5

6

7

- Entité medecin :

Liste des Medecins

Keyword

ID	Nom	Email	Specialite		
1	medecin1	mail@email.com	cardio	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
2	medecin2	bro@email.com	neuro	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
3	medecin3	test@email.com	famille	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
4	medecin4	ik@email.com	traumato	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
5	medecin1	mail@email.com	cardio	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>

0	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	28	29	30	31																

- Ajout de medecin :

Nom

la taille doit être comprise entre 4 et 40

Email

Specialité

158	medchir	med@email.com	chirurgien	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
-----	---------	---------------	------------	---------------------------------------	-------------------------------------

- recherche selon 2 critères : nom et spécialité

Keyword [Chercher](#)

ID	Nom	Email	Specialite
1	medecin1	mail@email.com	cardio
5	medecin1	mail@email.com	cardio
9	medecin1	mail@email.com	cardio
13	medecin1	mail@email.com	cardio
17	medecin1	mail@email.com	cardio

Keyword [Chercher](#)

ID	Nom	Email	Specialite		
161	medecin5	test@email.com	neuro	Delete	Edit
162	medecin8	ik@email.com	neuro	Delete	Edit

- **Inscription : signup (en tant que USER)**

[Login](#) [Signup](#)

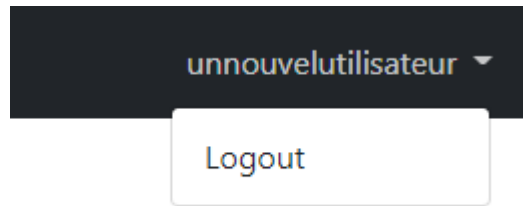
[localhost:8082/signup](#)

Username

password

Verify your password

[Enregistrer](#)



- Dans la base de données :

<input type="checkbox"/>	Éditer	Copier	Supprimer	99027b01-7a95-429f-b3cd-60dc6fc11b99	1	\$2a\$10\$Zh6eyOL88PcUFX5yas79ON5cvxSZuHB8/ggELsAy3n...	unnouvelutilisateur
99027b01-7a95-429f-b3cd-60dc6fc11b99					1		

****Role avec id=1 est le role USER**

7. Conclusion :

Cet atelier a englobé toutes les fonctionnalités basiques de manipulation de données : ajouter, supprimer, modifier ,afficher et rechercher , en utilisant thymeleaf layout et th pour créer les templates, et spring security pour la gestion des différents types d'utilisateurs .