



Rapport réalisé par
Messary Mehdi

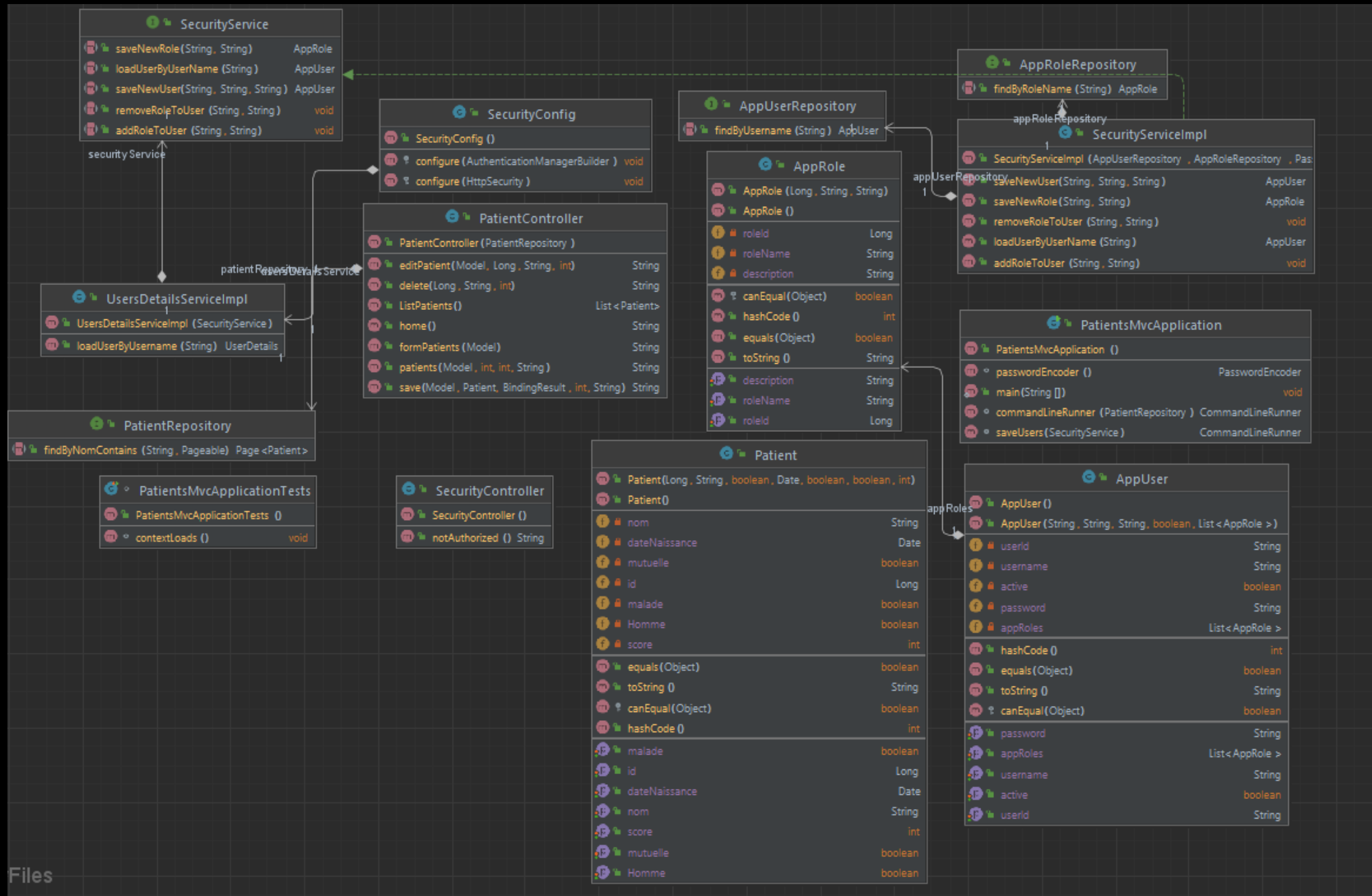
Spring MVC, Thymeleaf, Spring Data et
Spring Security & User Details Service

Enoncé:

- Partie 1 :
 - ● Affichage des patients
 - ● La pagination
 - ● Recherche des patients
 - ● Suppression d'un patient
- Partie 2 :
 - ● Créer une page template basée sur thymeleaf layout
 - ● Saisie et Ajout de patients
 - ● Formulaire
 - ● Editer les attributs d'un patient

-
- Partie 3 & 4 :
 - Ajout de la dépendance Maven de Spring Security
 - Personnaliser Spring Security pour ajouter les restriction avec la stratégie InMemoryAuthentication :
 - Authentification avec le rôle USER pour
 - pouvoir consulter les Patients.
 - Authentification avec le rôle ADMIN
 - pour pouvoir ajouter, editer, supprimer des patients.

Conception et Architecture



Code Source

<https://github.com/MessaryMehdi/Atelier3-4.git>

Page Sign In

Please sign in

You have been signed out

Sign in

Affichage & Pagination

Liste des patients

Key Word [Chercher](#)

ID	Nom	Homme	Date	Malade	Mutuelle	Score
17	Hassan	false	2022-04-25	false	false	12
21	Hassan	false	2022-04-25	false	false	12
22	Mehdi	false	2022-04-25	true	false	33
25	Hassan	false	2022-04-25	false	false	12
26	Mehdi	false	2022-04-25	true	false	33

[0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#)
[22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#) [31](#) [32](#) [33](#) [34](#)

Liste des patients

Key Word [Chercher](#)

ID	Nom	Homme	Date	Malade	Mutuelle	Score
319	Ali	true	2022-04-26	false	false	160
320	Hassan	true	2022-04-26	false	true	122
321	Mehdi	true	2022-04-26	true	false	333
322	Fatima	false	2022-04-26	true	true	130
323	Ali	true	2022-04-26	false	false	160

[0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#)
[22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#) [31](#) [32](#) [33](#) [34](#)

Recherche par Nom

Liste des patients

Key Word

Chercher

ID	Nom	Homme	Date	Malade	Mutuelle	Score
313	Mehdi	true	2022-04-26	true	false	333
317	Mehdi	true	2022-04-26	true	false	333
321	Mehdi	true	2022-04-26	true	false	333
325	Mehdi	true	2022-04-26	true	false	333
329	Mehdi	true	2022-04-26	true	false	333

0

1

2

3

4

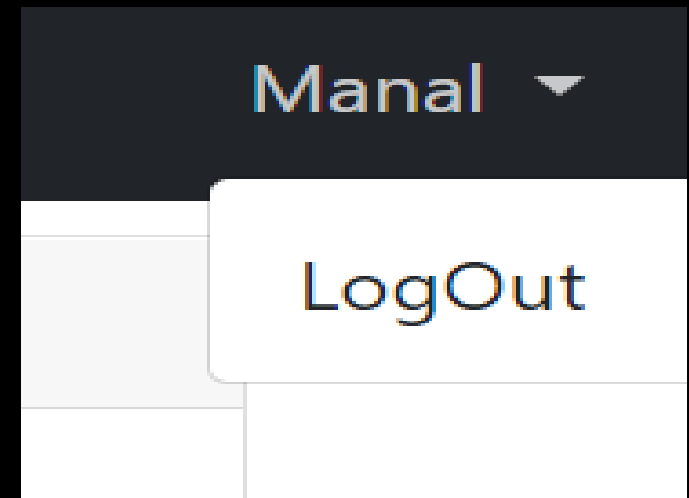
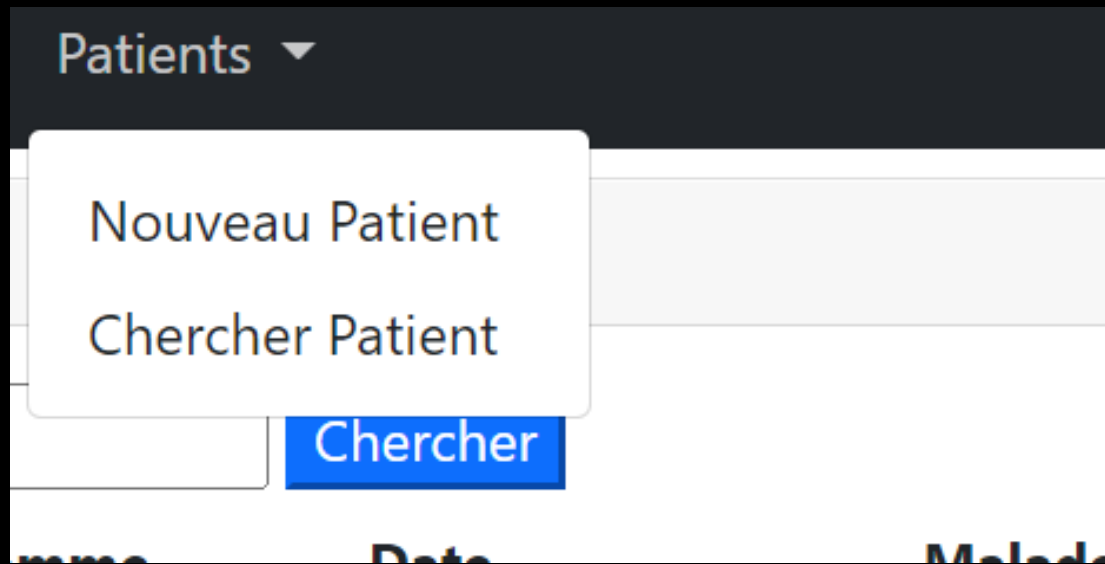
5

6

7

8

Nav Bar & Log Out



Message de confirmation
avant suppression d'un
patient (Hassan, ID 17)

Médecin Hopital Disabled Patients ▼

localhost:8082 indique
etes-vous sûr?

OK

Annuler

Liste des patients

Key Word

Chercher

ID	Nom	Homme	Date	Malade	Mutuelle	Score	
17	Hassan	false	2022-04-25	false	false	12	Delete
21	Hassan	false	2022-04-25	false	false	12	Delete
22	Mehdi	false	2022-04-25	true	false	33	Delete
25	Hassan	false	2022-04-25	false	false	12	Delete
26	Mehdi	false	2022-04-25	true	false	33	Delete

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

22

23

24

25

26

27

28

29

30

31

32

33

34

Hassan, ID=17 a
bien été supprimé

Liste des patients

Key Word

Chercher

ID	Nom	Homme	Date	Malade	Mutuelle	Score	
21	Hassan	false	2022-04-25	false	false	12	Delete
22	Mehdi	false	2022-04-25	true	false	33	Delete
25	Hassan	false	2022-04-25	false	false	12	Delete
26	Mehdi	false	2022-04-25	true	false	33	Delete
29	Hassan	false	2022-04-25	false	false	12	Delete

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

22

23

24

25

26

27

28

29

30

31

32

33


34

Ajout d'un patient

Nom

Homme ☐

Date de naissance

jj/mm/aaaa 

Malade ☐

Mutuelle ☐

Score

0


Save

Nom

NouveauPatient

Homme ☐

Date de naissance

10/07/2000 

Malade ☒

Mutuelle ☒

Score

128

Save

Le nouveau patient
a bien été ajouté

Liste des patients

Key Word

Chercher

ID

Nom

Homme

Date

Malade

Mutuelle

Score

344

NouveauPatient

false

2000-07-10

true

true

128

Delete

0

Editor (Mehdi, ID 22)

Liste des patients

Key Word

Chercher

ID	Nom	Homme	Date	Malade	Mutuelle	Score		
21	Hassan	false	2022-04-25	false	false	12	<button>Delete</button>	<button>Edit</button>
22	Mehdi	false	2022-04-25	true	false	33	<button>Delete</button>	<button>Edit</button>
25	Hassan	false	2022-04-25	false	false	12	<button>Delete</button>	<button>Edit</button>
26	Mehdi	false	2022-04-25	true	false	33	<button>Delete</button>	<button>Edit</button>
29	Hassan	false	2022-04-25	false	false	12	<button>Delete</button>	<button>Edit</button>

0

1

2

3

4

5

6

7

8

9

1

[illegible]

2

13

1

1

11

5

17

18

1

11

1

22

2

11

5

26

2

1

1

9

30

3.

1

[illegible]

1

Mehdi a bien été modifié

ID : 345

Homme

false

Nom

MehdiModified

Date de naissance

jj/mm/aaaa

Malade ☐

Mutuelle ☐

Score

330

Save

Liste des patients

Key Word

Chercher

ID	Nom	Homme	Date	Malade	Mutuelle	Score		
345	MehdiModified	false		false	false	330	Delete	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

32

33

34

35

ENTITIES

App User & App Role

13 usages

```
@Entity
@Data @NoArgsConstructor @AllArgsConstructor
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<>();
}
```

REPOSITORIES

3 usages

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

3 usages

```
public interface AppRoleRepository extends JpaRepository<AppRole,String> {
    AppRole findByRoleName(String roleName);
}
```


SERVICES

Security Service

5 usages 1 implementation

```
public interface SecurityService {
```

3 usages 1 implementation

```
AppUser saveNewUser(String username, String password, String rePassword);
```

2 usages 1 implementation

```
AppRole saveNewRole(String roleName, String description);
```

4 usages 1 implementation

```
void addRoleToUser(String username, String roleName);
```

1 implementation

```
void removeRoleToUser(String username, String roleName);
```

1 usage 1 implementation

```
AppUser loadUserByUserName(String username);
```

```
}
```

Security Service

Impl

```
@Service
@Slf4j
@AllArgsConstructor
@Transactional
public class SecurityServiceImpl implements SecurityService {

    4 usages
    private AppUserRepository appUserRepository;

    4 usages
    private AppRoleRepository appRoleRepository;

    1 usage
    private PasswordEncoder passwordEncoder;

    3 usages
    @Override
    public AppUser saveNewUser(String username, String password, String rePassword) {
        if (!password.equals(rePassword)) throw new RuntimeException("Password does not match");
        String hashedPWD=passwordEncoder.encode(password);
        AppUser appUser=new AppUser();
        appUser.setUserId(UUID.randomUUID().toString());
        appUser.setUsername(username);
        appUser.setPassword(hashedPWD);
        appUser.setActive(true);
        AppUser savedAppUser=appUserRepository.save(appUser);
        return appUser;
    }
}
```

```
@Override
public AppRole saveNewRole(String roleName, String description) {
    AppRole appRole =appRoleRepository.findByRoleName(roleName);
    if (appRole!=null) throw new RuntimeException("Role" +roleName+ "Already exist");
    appRole=new AppRole();
    appRole.setRoleName(roleName);
    appRole.setDescription(description);
    AppRole savedAppRole=appRoleRepository.save(appRole);
    return savedAppRole;
}
```

```
4 usages
@Override
public void addRoleToUser(String username, String roleName) {
    AppUser appUser=appUserRepository.findByUsername(username);
    if (appUser==null) throw new RuntimeException("User not found");
    AppRole appRole=appRoleRepository.findByRoleName(roleName);
    if (appRole==null) throw new RuntimeException("Role not found");
    appUser.getAppRoles().add(appRole);
}
```

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

1 usage

```
@Override
public AppUser loadUserByUserName(String username) {

    return appUserRepository.findByUsername(username);
}
```

```
}
```

User Details Service

```
@Service
public class UsersDetailsServiceImpl implements UserDetailsService {
    2 usages
    private SecurityService securityService;

    public UsersDetailsServiceImpl(SecurityService securityService) { this.securityService = securityService; }

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        AppUser appUser=securityService.loadUserByUserName(username);

        /* Collection<GrantedAuthority>authorities= new ArrayList();
        appUser.getAppRoles().forEach(role -> {
            SimpleGrantedAuthority authority=new SimpleGrantedAuthority(role.getRoleName());
            authorities.add(authority);
        });*/

        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;
    }
}
```

Security Config

```
@Configuration
@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

    @Autowired
    private DataSource dataSource;

    1 usage

    @Autowired
    private UsersDetailsServiceImpl usersDetailsService;

    @Autowired

    private PasswordEncoder passwordEncoder;

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
```

```
        auth.userDetailsService(usersDetailsService);
```

```
    }
```

```
    @Override
```

```
    protected void configure(HttpSecurity http) throws Exception {
```

```
        http.formLogin();
```

```
        http.authorizeRequests().antMatchers(...antPatterns: "/").permitAll();
```

```
        http.authorizeRequests().antMatchers(...antPatterns: "/admin/**").hasAuthority("ADMIN");
```

```
        http.authorizeRequests().antMatchers(...antPatterns: "/user/**").hasAuthority("USER");
```

```
        http.authorizeRequests().antMatchers(...antPatterns: "/webjars/**").permitAll();
```

```
        http.authorizeRequests().anyRequest().authenticated();
```

```
        http.exceptionHandling().accessDeniedPage( accessDeniedUrl: "/403");
```

```
    }
```

MVC New Users

```
CommandLineRunner saveUsers(SecurityService securityService){  
    return args -> {  
        securityService.saveNewUser( username: "Farouk", password: "1234", rePassword: "1234");  
        securityService.saveNewUser( username: "Abdou", password: "1234", rePassword: "1234");  
        securityService.saveNewUser( username: "Manal", password: "1234", rePassword: "1234");  
  
        securityService.saveNewRole( roleName: "USER", description: "");  
        securityService.saveNewRole( roleName: "ADMIN", description: "");  
  
        securityService.addRoleToUser( username: "Farouk", roleName: "USER");  
        securityService.addRoleToUser( username: "Abdou", roleName: "USER");  
        securityService.addRoleToUser( username: "Manal", roleName: "ADMIN");  
        securityService.addRoleToUser( username: "Manal", roleName: "USER");  
    };  
}
```