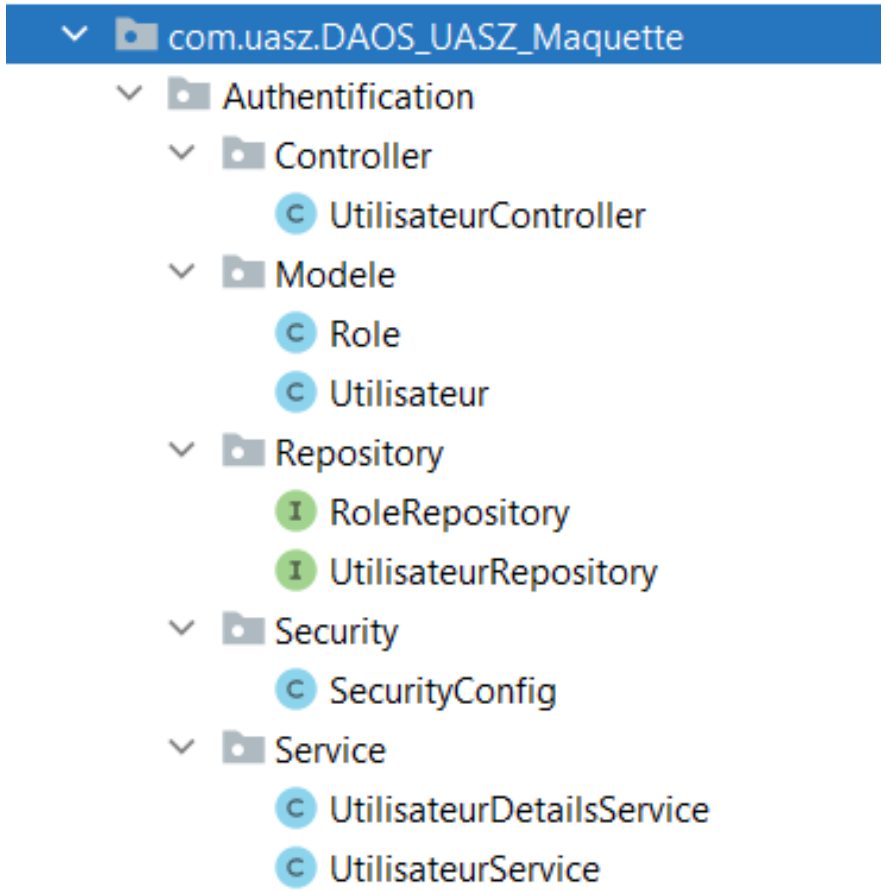
DAOS

Package Authentification



Pom.xml

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

Modèle : Utilisateur et Role

```java
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Utilisateur {
    @Id
    @GeneratedValue(strategy= GenerationType.IDENTITY)
    private Long id;
    @Column(unique = true)
    private String username;
    @NotNull
    private String password;
    private String nom;
    private String prenom;
    @Temporal(TemporalType.TIMESTAMP)
    private Date dateCreation;
    private boolean active;
    @ManyToMany(fetch = FetchType.EAGER)
    private List<Role> roles;
}
```

```java
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Role {
    @Id
    private String role;
}
```

Repository : Utilisateur et role

```java
@Repository
public interface UtilisateurRepository extends JpaRepository<Utilisateur, Long> {
    3 usages
    @Query("SELECT u FROM Utilisateur u WHERE u.username = :username")
    Utilisateur findUtilisateurByUsername(@Param("username") String email);
}

@Repository
public interface RoleRepository extends JpaRepository<Role, String> {
    1 usage
    @Query("SELECT r FROM Role r WHERE r.role = :role")
    Role findRoleByRole(@Param("role") String role);
}
```

Service : Utilisateur

```java
@Service
@Transactional
public class UtilisateurService {

    3 usages
    @Autowired
    private UtilisateurRepository utilisateurRepository;

    2 usages
    @Autowired
    private RoleRepository roleRepository;

    4 usages
    public Utilisateur ajouter_Utilisateur(Utilisateur utilisateur){
        utilisateurRepository.save(utilisateur);
        return utilisateur;
    }

    public Role ajouter_Role(Role role){
        roleRepository.save(role);
        return role;
    }

    4 usages
    public void ajouter_UtilisateurRoles(Utilisateur utilisateur, Role role){
        Utilisateur user = utilisateurRepository.findUtilisateurByUsername(utilisateur.getUsername());
        Role profil = roleRepository.findRoleByRole(role.getRole());
        user.getRoles().add(profil);
    }

    2 usages
    public Utilisateur rechercher_Utilisateur(String username){
        Utilisateur utilisateur = utilisateurRepository.findUtilisateurByUsername(username);
        return utilisateur;
    }

}
```

Service : UtilisateurDetails

```java
@Service
public class UtilisateurDetailsService implements UserDetailsService {
    2 usages
    private UtilisateurRepository utilisateurRepository;
    public UtilisateurDetailsService(UtilisateurRepository utilisateurRepository) {
        this.utilisateurRepository = utilisateurRepository;
    }
    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        Utilisateur utilisateur = utilisateurRepository.findUtilisateurByUsername(username);
        String[] roles = utilisateur.getRoles().stream().map(u->u.getRole()).toArray(String[]::new);
        UserDetails userDetails =
                org.springframework.security.core.userdetails.User.builder()
                        .username(utilisateur.getUsername())
                        .password(utilisateur.getPassword())
                        .roles(roles)
                        .build();

        return userDetails;
    }
}
```

Security : SecurityConfig

```java
@Configuration
@EnableWebSecurity
public class SecurityConfig {

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

    private static final String[] FOR_MAQUETTE = {"/maquette/**"};

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
                .authorizeHttpRequests()
                .requestMatchers( ...patterns: "/login**","/logout**").permitAll()
                .requestMatchers(FOR_MAQUETTE).hasAnyRole( ...roles: "ADMIN","RESP_PEDAGOGIE")
                .anyRequest().authenticated();
        http
                .formLogin()
                .loginPage("/login")
                .defaultSuccessUrl("/")
                .successForwardUrl("/")
                .permitAll();

        return http.build();
    }

}
```

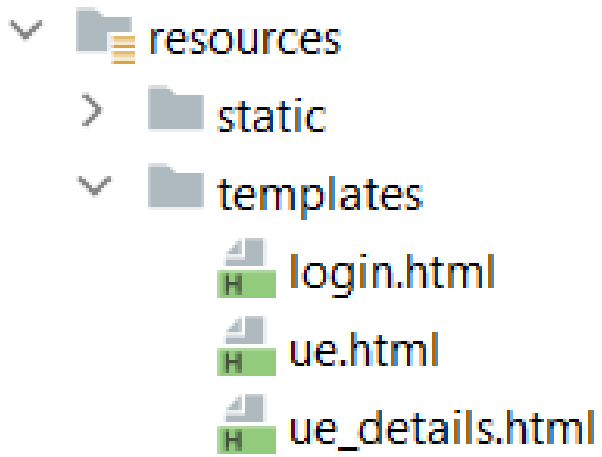
Controller : Utilisateur

```java
@Controller
public class UtilisateurController {
    1 usage
    @Autowired
    private UtilisateurService utilisateurService;

    @RequestMapping(value = "/login")
    public String index(){
        return "login";
    }

    @RequestMapping("/")
    public String login(Principal principal) {
        String url="login";
        Utilisateur utilisateur=utilisateurService.rechercher_Utilisateur(principal.getName());
        System.out.println(utilisateur.getRoles().get(0).getRole());
        if(utilisateur.getRoles().get(0).getRole().equals("ADMIN")){
            url="redirect:/maquette/ue";
        }
        else if (utilisateur.getRoles().get(0).getRole().equals("RESP_PEDAGOGIE")){
            url="redirect:/maquette/ue";
        }
        return url;
    }
    @RequestMapping(value = "/logout")
    public String logOutAndRedirectToLoginPage(Authentication authentication,
                                               HttpServletRequest request,
                                               HttpServletResponse response) {
        if (authentication != null) {
            new SecurityContextLogoutHandler().logout(request, response, authentication);
        }
        return "redirect:/login?logout=true";
    }
}
```

Run

```
@Autowired
private UtilisateurService utilisateurService;
private PasswordEncoder passwordEncoder;
1 usage
public PasswordEncoder passwordEncoder() { return new BCryptPasswordEncoder(); }

String password = passwordEncoder().encode( rawPassword: "Passer123");

Utilisateur user_1=utilisateurService.ajouter_Utilisateur(new Utilisateur( id: null, username: "admin@uasz.sn",password, nor
utilisateurService.ajouter_UtilisateurRoles(user_1,admin);

Utilisateur user_2=utilisateurService.ajouter_Utilisateur(new Utilisateur( id: null, username: "dieng@uasz.sn",password, nor
utilisateurService.ajouter_UtilisateurRoles(user_2,responsable);

Utilisateur user_3=utilisateurService.ajouter_Utilisateur(new Utilisateur( id: null, username: "diallo@uasz.sn",password, n
utilisateurService.ajouter_UtilisateurRoles(user_3,chef);

Utilisateur user_4=utilisateurService.ajouter_Utilisateur(new Utilisateur( id: null, username: "diop@uasz.sn",password, nom
utilisateurService.ajouter_UtilisateurRoles(user_4,enseignant);
```

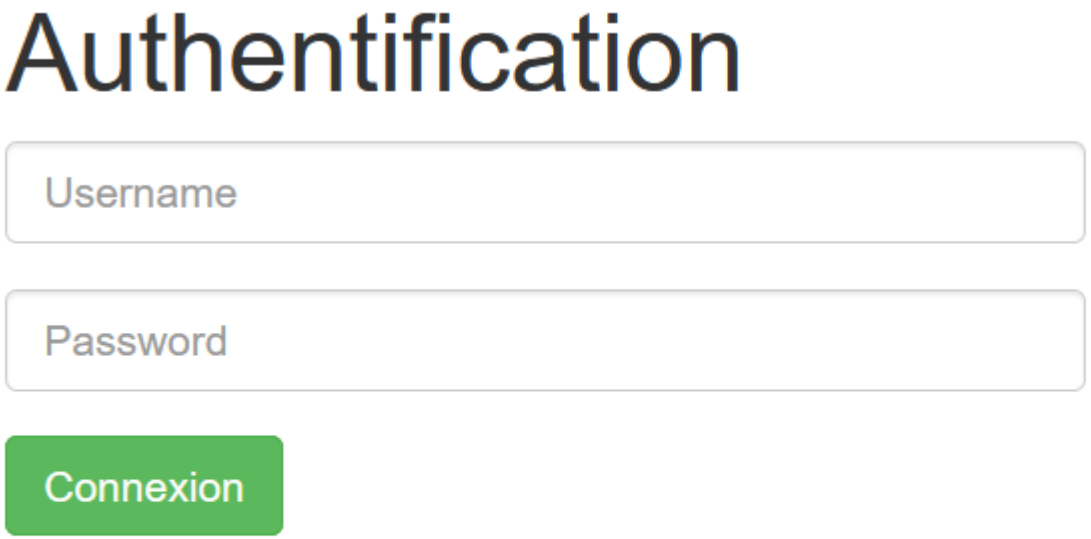
Vue


resources
static
templates
login.html
ue.html
ue_details.html

Vue : login.html

```
<div class="row">
    <div class="col-xs-12 col-sm-6 col-sm-offset-1">
        <h1>Authentification</h1>
        <form th:action="@{/login}" name="login" role="form" class="form-horizontal" method="post" accept-charset="utf-8">
            <div class="form-group">
                <div class="col-md-8">
                    <input name="username" th:placeholder="Username" class="form-control" type="text"/>
                </div>
            </div>
            <div class="form-group">
                <div class="col-md-8">
                    <input name="password" th:placeholder="Password" class="form-control" type="password"/>
                </div>
            </div>
            <div class="form-group">
                <div class="col-md-offset-0 col-md-8">
                    <input  class="btn btn-success btn btn-success" type="submit" th:value="Connexion"/>
                </div>
            </div>
        </form>
    </div>
</div>
```

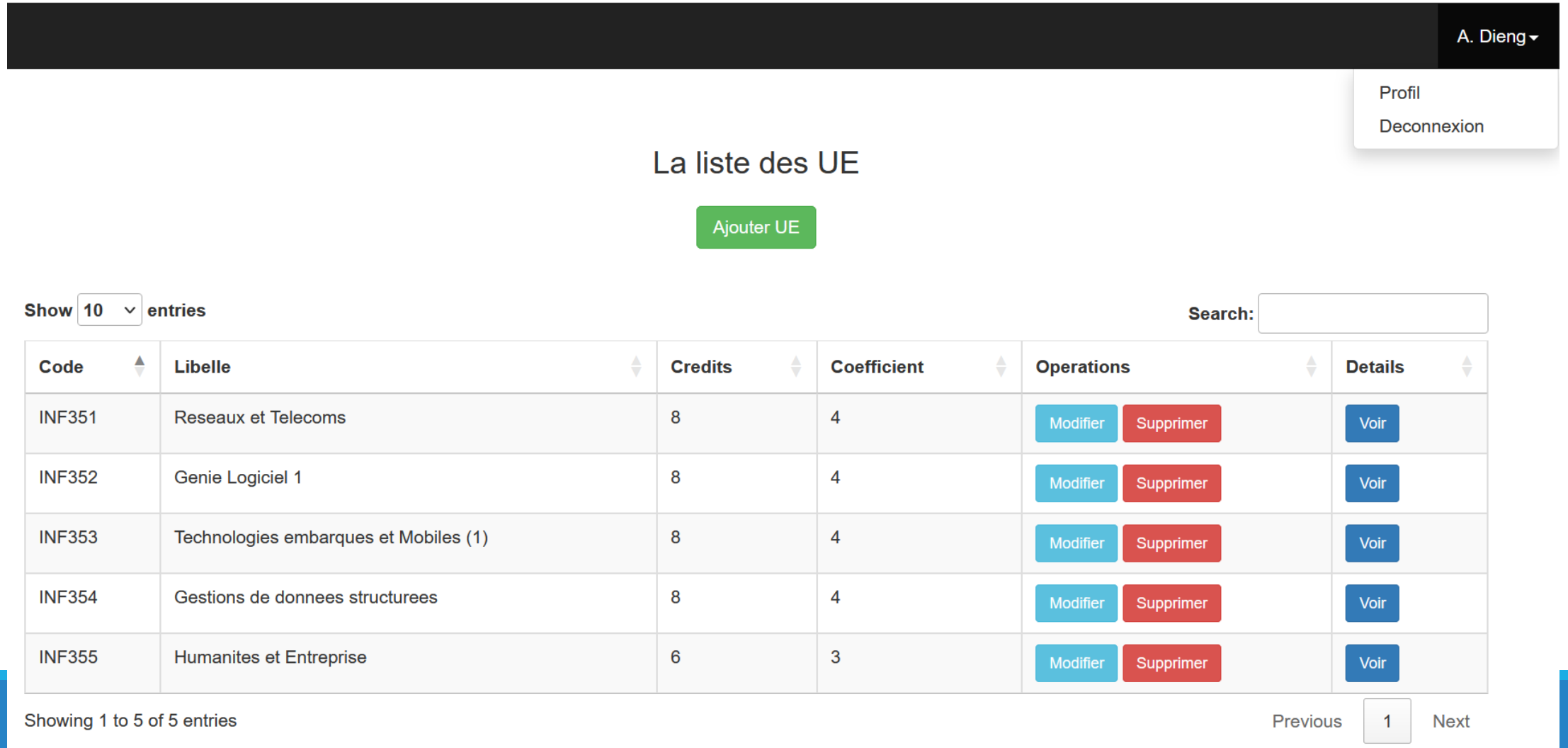
Vue : login.html



Vue : ue.html

```
<header>
    <nav class="navbar navbar-inverse">
        <div class="container-fluid">
            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                <ul class="nav navbar-nav navbar-right">
                    <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                            <span th:text="${prenom+'. '+nom}"></span><span class="caret"></span>
                        </a>
                        <ul class="dropdown-menu">
                            <li>
                                <a><span>Profil</span></a>
                            </li>
                            <li><a th:href="@{/logout}">
                                <span>Deconnexion</span></a>
                            </li>
                        </ul>
                    </li>
                </ul>
            </div><!-- /.navbar-collapse -->
        </div><!-- /.container-fluid -->
    </nav>
</header>
```

Vue : ue.html


A. Dieng
Profil
Deconnexion
La liste des UE
Ajouter UE
Show 10 entries
Search:
Code
Libelle
Credits
Coefficient
Operations
Details
INF351 Reseaux et Telecoms 8 4 Modifier Supprimer Voir
INF352 Genie Logiciel 1 8 4 Modifier Supprimer Voir
INF353 Technologies embarques et Mobiles (1) 8 4 Modifier Supprimer Voir
INF354 Gestions de donnees structurees 8 4 Modifier Supprimer Voir
INF355 Humanites et Entreprise 6 3 Modifier Supprimer Voir
Showing 1 to 5 of 5 entries
Previous
1
Next