

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

▶ LI

Nom d'usage

▶ Entrez votre nom d'usage ici.

Prénom

▶ David

Adresse

▶ 25 rue Pierre et Marie Curie, 94200

Titre professionnel visé

Concepteur développeur d'application

MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Intitulé de l'activité-type n° 1 : Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

p.

5

- ▶ Intitulé de l'exemple n° 1 p.
- ▶ Intitulé de l'exemple n° 2 p.
- ▶ Intitulé de l'exemple n° 3 p.

Intitulé de l'activité-type n° 2 : Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

p.

- ▶ Intitulé de l'exemple n° 1 p.
- ▶ Intitulé de l'exemple n° 2 p.
- ▶ Intitulé de l'exemple n° 3 p.

Intitulé de l'activité-type n° 3 : Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

p.

- ▶ Intitulé de l'exemple n° 1 p.
- ▶ Intitulé de l'exemple n° 2 p.
- ▶ Intitulé de l'exemple n° 3 p.

Intitulé de l'activité-type n° 4

p.

- ▶ Intitulé de l'exemple n° 1 p.
- ▶ Intitulé de l'exemple n° 2 p.
- ▶ Intitulé de l'exemple n° 3 p.

Titres, diplômes, CQP, attestations de formation *(facultatif)*

p.

Déclaration sur l'honneur

p.

Documents illustrant la pratique professionnelle *(facultatif)*

p.

Annexes *(Si le RC le prévoit)*

p.

DOSSIER PROFESSIONNEL (DP)

EXEMPLES DE PRATIQUE PROFESSIONNELLE

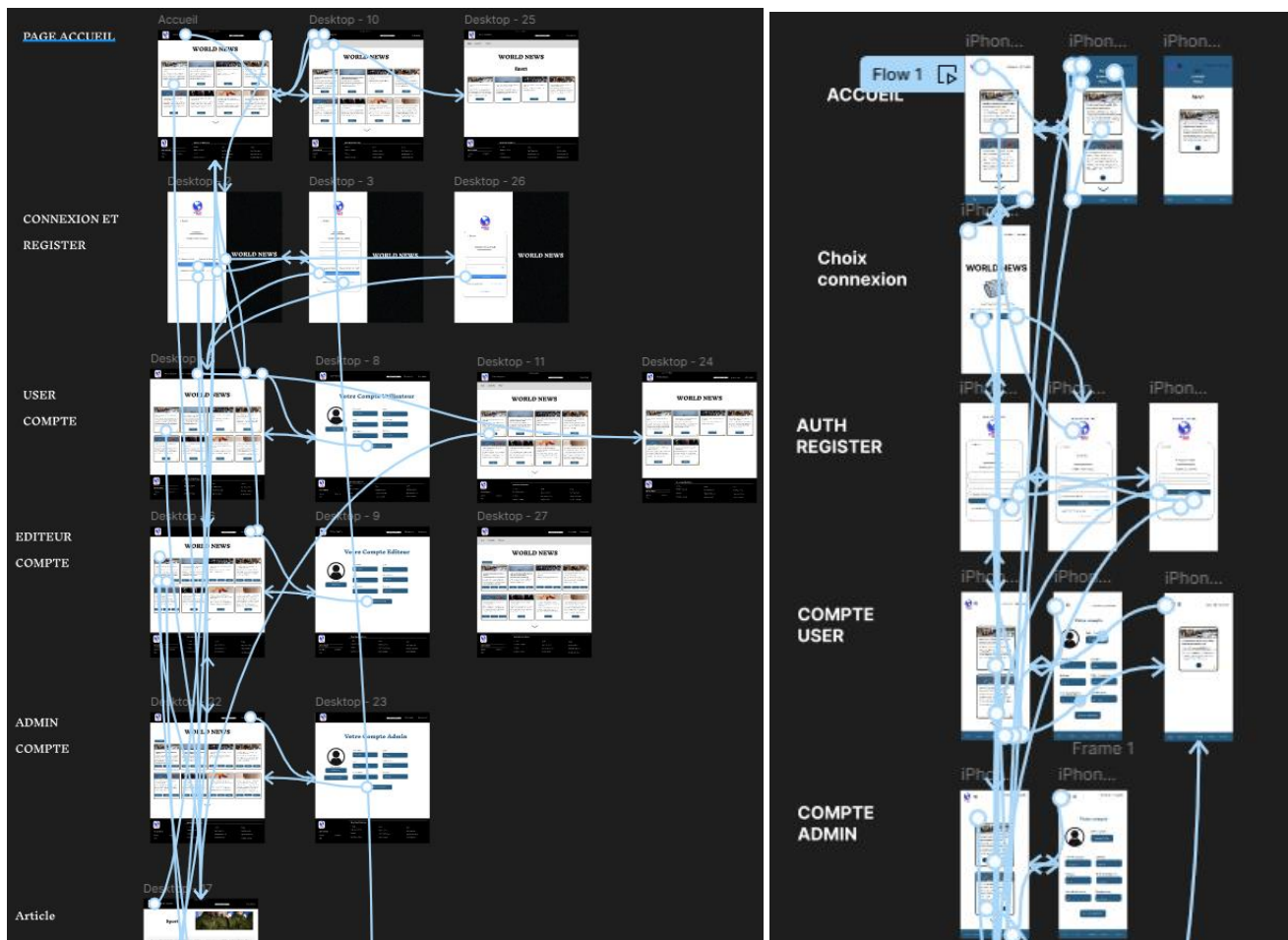
Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 ► Maquetter une application

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans ce premier exemple, j'ai conçu une maquette en cours de formation qui m'a permis de savoir comment concevoir parfaitement une maquette avec Figma. Il a pour thème l'affichage des articles. Dans cette maquette, on a 2 types, la version desktop et mobile pour visualiser ce que cela pourrait donner dans le projet. J'ai ajouté des interactions et des animations pour améliorer l'expérience utilisateur. Mais avant cela, j'ai dû regarder les besoins des utilisateurs, les user stories, ainsi que le cahier des charges pour mieux perfectionner ma maquette.



2. Précisez les moyens utilisés :

- Utilisation du logiciel Figma pour les maquettes

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul

4. Contexte

Nom de l'entreprise, organisme ou association ► *En formation*

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : 15/04/2024 au : 17/04/2024

5. Informations complémentaires (facultatif)

Activité-type 2

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

**Exemple n° 1 ► Développer des apis qui se connecte avec la base de données
En intégrant les recommandations de sécurité**

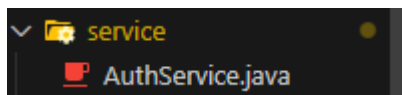
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Ce projet que j'ai fait a pour but d'utiliser uniquement des API sans utiliser d'interface utilisateur. En effet, cela me permet de gagner en compétences sur la conception des API et ainsi du langage Java en utilisant Spring Boot. J'ai d'abord utilisé le cahier des charges du projet, puis, en suite de cela, j'ai développé un diagramme de séquence, puis pour finir, j'ai utilisé la méthodologie Merise pour ne pas me tromper dans la conception de ma base de données, et en suite de cela, j'ai pu commencer le lancement de mon projet en ajoutant les dépendances, comme Spring Security, pour gérer la sécurité de l'application.

Voici l'importation de la dépendance de Spring Security :

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

J'ai créé un service pour l'authentification et l'inscription des utilisateurs :



```
public AuthService(
    UserRepository userRepository,
    AuthenticationManager authenticationManager,
    PasswordEncoder passwordEncoder
) {
    this.authenticationManager = authenticationManager;
    this.userRepository = userRepository;
    this.passwordEncoder = passwordEncoder;
}

public User signup(RegisterDto input) {
    User user = new User();
    user.setFullName(input.getFullName());
    user.setEmail(input.getEmail());
}
```



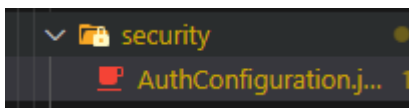
```
user.setRole(Role.USER);
user.setPassword(passwordEncoder.encode(input.getPassword()));

return userRepository.save(user);
}

public User authenticate(LoginDto input) {
    authenticationManager.authenticate(
        new UsernamePasswordAuthenticationToken(
            input.getEmail(),
            input.getPassword()
        )
    );

    return userRepository.findByEmail(input.getEmail())
        .orElseThrow();
}
```

Et j'ai créé une configuration qui permet l'accès aux ressources :



```
@Configuration
@EnableWebSecurity
public class AuthConfiguration {
    private final AuthenticationProvider authenticationProvider;
    private final JwtAuthenticationFilter jwtAuthenticationFilter;

    public AuthConfiguration(
        JwtAuthenticationFilter jwtAuthenticationFilter,
        AuthenticationProvider authenticationProvider) {
        this.authenticationProvider = authenticationProvider;
        this.jwtAuthenticationFilter = jwtAuthenticationFilter;
    }

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception
    {
        http
            .csrf(AbstractHttpConfigurer::disable)
            .authorizeHttpRequests(requests -> requests
                .requestMatchers("/auth/signup", "/auth/login").permitAll()
                .requestMatchers("/evenement/**", "/epreuve/**",
                    "/billet/**").authenticated()
            );
    }
}
```

```
        )
        .sessionManagement(session ->
session.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
        .authenticationProvider(authenticationProvider)
        .addFilterBefore(jwtAuthenticationFilter,
UsernamePasswordAuthenticationFilter.class);

        return http.build();
    }

    @Bean
    CorsConfigurationSource corsConfigurationSource() {
        CorsConfiguration configuration = new CorsConfiguration();

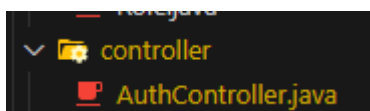
        configuration.setAllowedOrigins(List.of("http://localhost:9090"));
        configuration.setAllowedMethods(List.of("GET", "POST"));
        configuration.setAllowedHeaders(List.of("Authorization", "Content-Type"));

        UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();

        source.registerCorsConfiguration("/**", configuration);

        return source;
    }
}
```

J'ai créé un contrôleur pour l'authentification :



Dans ce contrôleur, on peut aller

```
@RequestMapping("/auth")
@RestController
public class AuthController {
    private final JwtService jwtService;

    private final AuthService authenticationService;

    public AuthController(JwtService jwtService, AuthService authenticationService) {
        this.jwtService = jwtService;
        this.authenticationService = authenticationService;
    }
}
```

```
@PostMapping("/signup")
public ResponseEntity<User> register(@RequestBody RegisterDto registerUserDto) {
    User registeredUser = authenticationService.signup(registerUserDto);
    return new ResponseEntity<>(registeredUser, HttpStatus.CREATED);
}

@PostMapping("/login")
public ResponseEntity<LoginResponse> authenticate(
    @RequestBody LoginDto loginUserDto) {
    User authenticatedUser = authenticationService.authenticate(loginUserDto);

    String jwtToken = jwtService.generateToken(authenticatedUser);

    LoginResponse loginResponse = new LoginResponse();
    loginResponse.setToken(jwtToken);
    loginResponse.setExpiresIn(jwtService.getExpirationTime());

    return new ResponseEntity<>(loginResponse, HttpStatus.OK);
}
```

Dans ce contrôleur, il permet d'accéder au route /signup et /login. Il utilise les DTO pour la validation des données et en sécurisant l'accès aux ressources avec des jetons JWT.

```
@RequestMapping("/auth")
@RestController
public class AuthController {
    private final JwtService jwtService;

    private final AuthService authenticationService;

    public AuthController(JwtService jwtService, AuthService authenticationService) {
        this.jwtService = jwtService;
        this.authenticationService = authenticationService;
    }

    @PostMapping("/signup")
    public ResponseEntity<User> register(@RequestBody RegisterDto registerUserDto) {
        User registeredUser = authenticationService.signup(registerUserDto);
        return new ResponseEntity<>(registeredUser, HttpStatus.CREATED);
    }

    @PostMapping("/login")
    public ResponseEntity<LoginResponse> authenticate(
        @RequestBody LoginDto loginUserDto) {
        User authenticatedUser = authenticationService.authenticate(loginUserDto);
    }
}
```

```
String jwtToken = jwtService.generateToken(authenticatedUser);

LoginResponse loginResponse = new LoginResponse();
loginResponse.setToken(jwtToken);
loginResponse.setExpiresIn(jwtService.getExpirationTime());

return new ResponseEntity<>(loginResponse, HttpStatus.OK);
}
```

Et pour finir, voici mon entité User avec ses différents champs :

```
@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class User implements UserDetails {
    @Id
    @GeneratedValue(strategy = GenerationType.UUID)
    @Column(nullable = false)
    private String id;

    @Column(nullable = false)
    private String fullName;

    @Column(unique = true, length = 100, nullable = false)
    private String email;

    @Column(nullable = false)
    private String password;

    @CreationTimestamp
    @Column(updatable = false, name = "created_at")
    private Date createdAt;

    @UpdateTimestamp
    @Column(name = "updated_at")
    private Date updatedAt;

    @Enumerated(EnumType.STRING)
    private Role role;

    @OneToMany(mappedBy = "user")
    private Set<Billet> billets = new HashSet<>();
}
```

DOSSIER PROFESSIONNEL (DP)

```
@Override
public Collection<? extends GrantedAuthority> getAuthorities() {
    Set<Role> roles = Set.of(role);
    List<SimpleGrantedAuthority> authorities = new ArrayList<>();

    for (Role role : roles) {
        authorities.add(new SimpleGrantedAuthority(role.name()));
    }
    return authorities;
}
```

2. Précisez les moyens utilisés :

La majorité du temps, j'utilise le document officiel de Spring pour me documenter sur le projet et Stack Overflow pour déboguer mon projet.
Sinon les ressources : VsCode / MySQL / PhpMyAdmin(Wampserver) / Spring boot / Java

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul

4. Contexte

Nom de l'entreprise, organisme ou association ► *En formation*

Chantier, atelier, service ► *Cliquez ici pour taper du texte.*

Période d'exercice ► Du : *01/05/2024* au : *31/05/2024*

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 3

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 1 ► Développer une application mobile

DOSSIER PROFESSIONNEL (DP)

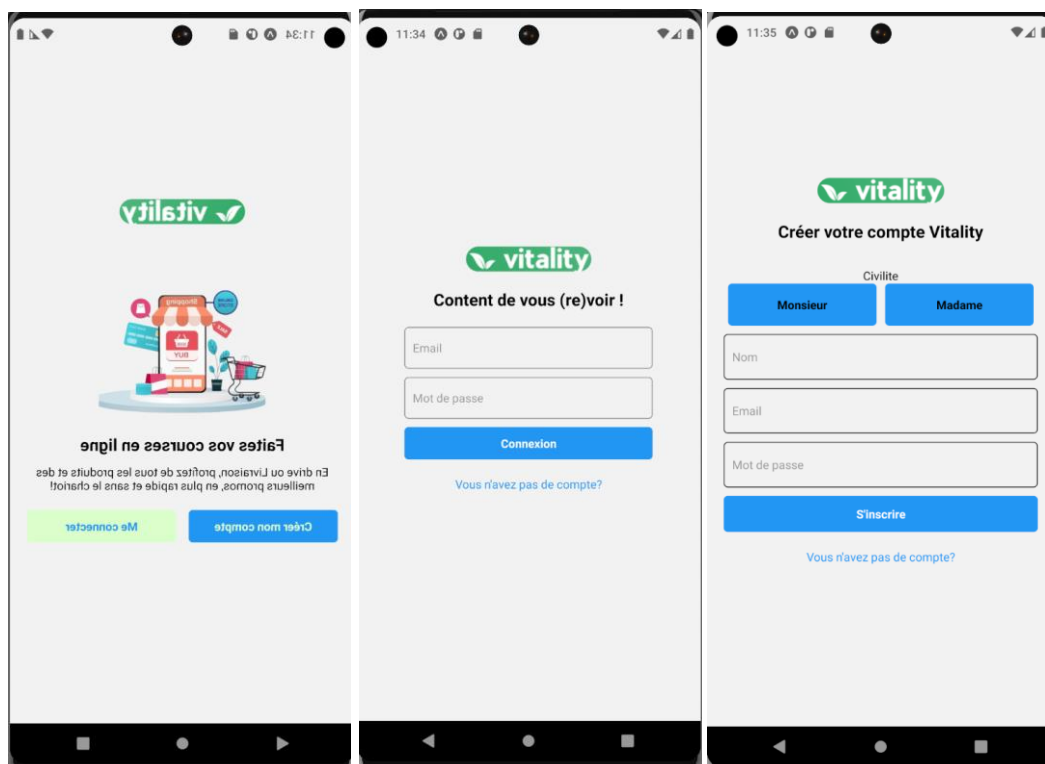
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Ce projet, est un projet personnel que j'ai fait cette année sur le même thème que le projet que je vous ai présenté, c'est-à-dire "vitalitymark". La seule différence, c'est que le projet est sur React Native, il est donc adapté sur mobile. Il utilise le serveur d'API Platform de Symfony pour récupérer les données de la base de données. Pour mettre cela en place, il faut préconfigurer notre package "nelmio_cors.yaml" pour qu'il gère les accès aux requêtes HTTP de mon projet mobile.

Voici mon package :

```
nelmio_cors:
  defaults:
    allow_credentials: false
    allow_origin: ["*"]
    allow_headers: ["*"]
    allow_methods: ["GET", "OPTIONS", "POST", "PUT", "PATCH", "DELETE"]
    expose_headers: []
    max_age: 3600
    hosts: []
    origin_regex: false
    forced_allow_origin_value: ~
  paths:
    "^/": ~
```

Dans mon package, j'ai fait en sorte que toutes les origines des requêtes soient acceptées, mais en réalité c'est mauvais. Mais dans la branche main, je restreindrais les origines des requêtes.



Dessus j'ai donc trois écrans, le premier est l'affichage de l'accueil, le deuxième, est la partie connexion et pour finir l'inscription.

Ce code, il s'agit de ma logique pour me connecter. Malheureusement, je n'ai rien mis en place en termes de sécurité dans l'application mobile. J'ai juste fait une vérification pour savoir si l'utilisateur existe ou non. Le saltRounds devrait être une chaîne de caractères aléatoires pour que le mot de passe soit différent même si c'est le même, ou encore l'ajout de tokens JWT ou bien des sessions.

```
try {
  const saltRounds = 10;
  const hashedPassword = bcrypt.hashSync(password, saltRounds);
  const response = await axios.get("http://192.168.1.47:8000/api/users", {
    params: {
      email: email,
      password: hashedPassword,
    },
  });

  if (response.data["hydra:totalItems"] === 1) {
    const user = response.data["hydra:member"][0];
    const isPasswordCorrect = bcrypt.compareSync(password, user.password);

    if (user.email === email && isPasswordCorrect === true) {
      await AsyncStorage.setItem("userId", user.id.toString());
      if (user.roles && user.roles.includes("ROLE_ADMIN")) {
        console.log("Redirecting to AccueilAdmin");
        navigation.navigate("AccueilAdmin");
      } else if (user.roles && user.roles.includes("ROLE_USER")) {
        console.log("Redirecting to AccueilUser");
        navigation.navigate("AccueilUser");
      } else {
        Alert.alert(
          "Erreur",
          "Vous n'avez pas les autorisations nécessaires."
        );
      }
    }
  }
}
```

2. Précisez les moyens utilisés :

J'ai utilisé le document officiel de React Native ainsi de Stack Overflow, et j'ai demandé de l'aide dans des serveurs communautaires.

Sinon les ressources : VsCode / MySQL / PhpMyAdmin(Wampserver) / React Native / Javascript

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul

4. Contexte

Nom de l'entreprise, organisme ou association ► Cliquez ici pour taper du texte.

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : 08/03/2024 au : 13/05/2024

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

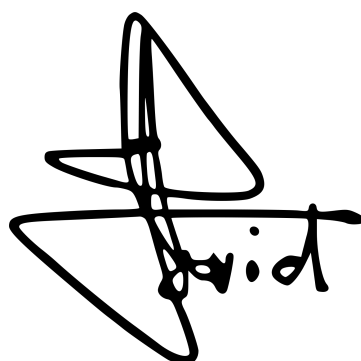
Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] David LI ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à Paris le 13/06/2024

pour faire valoir ce que de droit.

Signature :

A handwritten signature in black ink, appearing to read 'David LI', with a large, stylized loop above the name.

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)