



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance*

► MATTIO

*Nom d'usage*

► MATTIO

*Prénom*

► Enzo

*Adresse*

► 958 chemin de saint francet, 13390 Auriol

## Titre professionnel visé

**Concepteur Développeur d'Applications (RNCP 37873)**

### MODALITE D'ACCES :

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

# DOSSIER PROFESSIONNEL (DP)

## 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>

# DOSSIER PROFESSIONNEL (DP)

## Sommaire

### Exemples de pratique professionnelle

#### Activité-type n° 1 — Développer une application sécurisée (BC01)

p.

- ▶ Installer et configurer son environnement de travail ..... p.
- ▶ Développer des interfaces utilisateur sécurisées ..... p.
- ▶ Développer des composants métier ..... p.
- ▶ Contribuer à la gestion de projet ..... p.

#### Activité-type n° 2 — Concevoir et développer une application sécurisée organisée en couches (BC02)

p.

- ▶ Analyser les besoins et maquetter l'application ..... p.
- ▶ Définir l'architecture logicielle multicouche ..... p.
- ▶ Concevoir et mettre en place une base de données relationnelle ..... p.
- ▶ Développer des composants d'accès aux données SQL et NoSQL ..... p.

#### Activité-type n° 3 — Préparer le déploiement d'une application sécurisée (BC03)

p.

- ▶ Préparer et exécuter les plans de tests ..... p.
- ▶ Préparer et documenter le déploiement de l'application ..... p.
- ▶ Contribuer à la mise en production dans une démarche DevOps ..... 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.

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1 Développer une application sécurisée (BC01)

*Exemple n°1 ▶ Installer et configurer son environnement de travail en fonction du projet*

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

Pour démarrer le développement de Timelys, une application de gestion de pointage destinée aux entreprises du BTP et des services, j'ai dû mettre en place un environnement de développement professionnel capable de supporter une architecture multi-tenant complexe. Ce projet nécessitait une approche rigoureuse de la configuration, car il devait intégrer plusieurs technologies complémentaires (backend Symfony, frontend Flutter, base de données MySQL) tout en respectant les contraintes de sécurité et de performance.

Ma première préoccupation était d'assurer la cohérence entre les environnements de développement, de pré-production et de production. J'ai donc commencé par configurer Docker pour containeriser la base de données MySQL 8.4, ce qui me permettait de garantir des conditions identiques sur tous les environnements. La configuration de Symfony 7.2 avec PHP 8.2 a nécessité une attention particulière aux extensions requises (PDO, JSON, intl) et à l'optimisation de l'OPcache pour les performances.

Parallèlement, l'installation de l'environnement Flutter avec le SDK Dart 3.7.2 m'a amené à configurer les outils de debugging spécifiques au développement cross-platform. J'ai personnalisé Visual Studio Code avec les extensions nécessaires (PHP Intelephense, Flutter, Docker) pour optimiser ma productivité de développement.

La gestion des secrets et variables d'environnement était cruciale pour ce projet multi-tenant. J'ai structuré des fichiers .env séparés pour chaque environnement, en implémentant une rotation automatique des clés JWT et en sécurisant les accès base de données. La configuration Git incluait des hooks de pré-commit pour valider la qualité du code avant chaque commit.

Enfin, l'intégration avec Fly.io pour le déploiement cloud m'a demandé de maîtriser les spécificités de cette plateforme, notamment la configuration des variables d'environnement sécurisées et la mise en place du pipeline de déploiement automatisé.

### 2. Précisez les moyens utilisés :

# DOSSIER PROFESSIONNEL (DP)

- Docker et Docker Compose pour la conteneurisation
- Composer pour la gestion des dépendances PHP/Symfony
- Flutter SDK et pub pour la gestion des packages Dart
- Visual Studio Code avec extensions (PHP Intelephense, Flutter, Docker)
- Git pour le versioning avec GitHub
- Fly.io CLI pour le déploiement cloud
- MySQL Workbench pour la gestion de base de données
- Postman pour les tests d'API
- Fichiers de configuration : .env, pubspec.yaml, composer.json

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé de manière autonome sur ce projet, en suivant les bonnes pratiques apprises en formation. J'ai consulté la documentation officielle Symfony et Flutter, et échangé avec mes formateurs pour valider l'architecture technique.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Projet personnel / École La Plateforme

Chantier, atelier, service ► Projet CDA Timelys

Période d'exercice ► Du 01/01/2025 au 01/08/2025

## 5. Informations complémentaires (facultatif)

# DOSSIER PROFESSIONNEL (DP)

Cette configuration m'a permis de développer une application complète avec architecture multi-tenant, déployée en production sur Fly.io. L'environnement était optimisé pour le développement en équipe avec des outils de CI/CD automatisés.

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1 Développer une application sécurisée (BC01)

*Exemple n°2 ► Développer des interfaces utilisateur sécurisées*

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

Le développement de l'interface utilisateur de Timelys représentait un défi technique majeur : créer une application mobile intuitive pour des utilisateurs terrain tout en respectant les exigences de sécurité d'une application professionnelle. L'enjeu était de concevoir une expérience utilisateur fluide qui puisse fonctionner dans des conditions difficiles (chantiers, interventions terrain) tout en garantissant l'intégrité des données de pointage.

J'ai commencé par analyser les besoins utilisateur en me basant sur les retours terrain : les employés devaient pouvoir pointer rapidement, même avec des gants ou par temps pluvieux. Cette contrainte m'a orienté vers Material Design 3 avec des zones tactiles élargies et une navigation simplifiée. J'ai conçu un parcours utilisateur optimisé où l'action principale (pointage) est accessible en maximum 2 clics depuis l'accueil.

L'architecture de l'interface s'appuie sur un système d'authentification sécurisé utilisant JWT avec refresh token automatique. J'ai développé un système de navigation contextuelle qui adapte l'interface selon le rôle utilisateur : les employés voient uniquement leurs fonctions de pointage, tandis que les superviseurs accèdent aux tableaux de bord et aux outils d'administration. Cette approche garantit la sécurité en limitant l'exposition des fonctionnalités sensibles.

La création de composants réutilisables (AppDrawer pour la navigation, PhotoPicker pour les relevés terrain, StatusCards pour les informations de pointage) m'a permis de maintenir une cohérence visuelle tout en optimisant les performances. Chaque composant intègre sa propre gestion d'état et ses validations, réduisant ainsi la complexité de l'interface principale.

L'intégration avec l'API Symfony s'appuie sur un système de cache intelligent qui permet à l'application de fonctionner en mode déconnecté. Les données sont synchronisées automatiquement lors de la reconnexion, avec gestion des conflits et validation des timestamps pour éviter les incohérences.

### 2. Précisez les moyens utilisés :

# DOSSIER PROFESSIONNEL (DP)

- Flutter SDK avec Dart 3.7.2
- Provider pour la gestion d'état
- Material Design 3 pour l'interface utilisateur
- Flutter Secure Storage pour le stockage sécurisé des tokens
- HTTP package pour les appels API
- JWT Decoder pour la validation des tokens
- Image Picker pour la gestion des photos
- Figma pour la conception des maquettes (référence aux screens fournis)
- Chrome DevTools pour le debug web

## 3. Avec qui avez-vous travaillé ?

Développement individuel avec validation des maquettes et du design system par mes formateurs. J'ai utilisé les maquettes de référence fournies pour assurer la cohérence visuelle de l'application.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Projet personnel / École La Plateforme

Chantier, atelier, service ► Projet CDA Timelys

Période d'exercice ► Du 01/01/2025 au 01/08/2025

## 5. Informations complémentaires (*facultatif*)

# DOSSIER PROFESSIONNEL (DP)

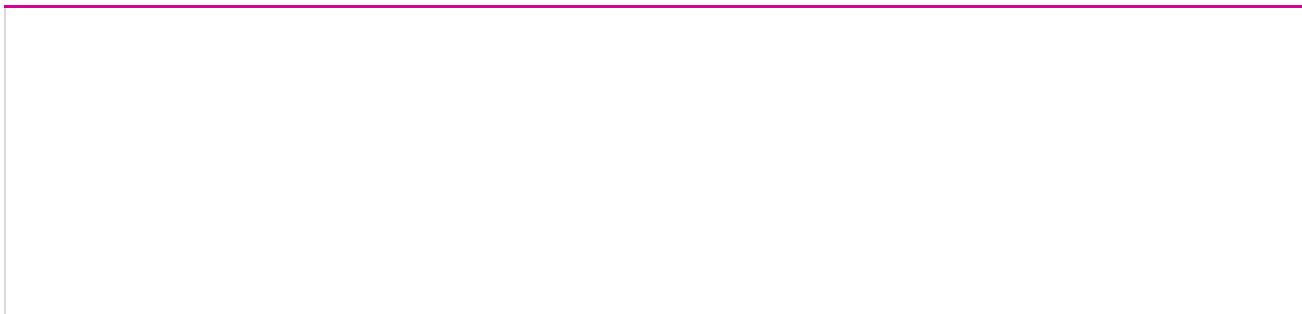
L'application supporte multiple plateformes (Android, iOS, Web) avec une interface adaptive. La sécurité est assurée par la validation des tokens JWT et la gestion granulaire des permissions selon les rôles utilisateur.

## Photos

The screenshot displays three main sections of the Dossier Professionnel (DP) application:

- Accueil (Home):** This section features a large teal header bar with a briefcase icon. Below it is a "Connexion" (Connection) form with fields for "Email" and "Mot de passe" (Password), and a "Se connecter" (Connect) button. A "Mot de passe oublié ?" (Forgot password?) link is also present. To the right, there's a "Bonjour !" (Hello!) message, a summary of today's activity (0h 00m), and a summary for the week (0h 00m). It also includes a "Pointage" (Attendance) summary showing the user is currently absent (0 sessions) and a "Activités" (Activities) summary showing 1 activity (8.0h). Finally, there's an "Actions rapides" (Quick actions) section with buttons for "Nouvelle activité" (New activity) and "Pointage" (Attendance).
- Pointage (Attendance):** This section shows the current status as "Absent" with 0 hours worked and 0 sessions. It includes an "Entrée" (Entry) button and a note that no attendance was recorded today. The date displayed is "Aujourd'hui - 28/7/2025".
- Monitoring Admin:** This section provides an overview of the company "Meca-Lev" (Entreprise). It includes a "Statistiques" (Statistics) section with daily, weekly, and monthly summaries, and an "Employés" (Employees) section showing one employee named "MATTIO Enzo" who is currently absent with 0 hours worked and 0 sessions.

# **DOSSIER PROFESSIONNEL (DP)**



# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1 Développer une application sécurisée (BC01)

*Exemple n°3 ► Développer des composants métier*

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

Le cœur de l'application Timelys réside dans sa logique métier complexe qui doit gérer les subtilités du suivi de temps professionnel tout en assurant la fiabilité et l'intégrité des données. Le challenge principal était de concevoir un système capable de détecter automatiquement les anomalies de pointage (oubli de sortie, durées anormales, géolocalisation incohérente) tout en restant suffisamment flexible pour s'adapter aux différents contextes métier des clients.

Le développement du service AuthenticationService m'a confronté aux enjeux de sécurité d'une application multi-tenant. J'ai implémenté un système de tokens JWT avec rotation automatique, où chaque organisation dispose de ses propres clés de chiffrement. La gestion des sessions inclut la détection des connexions multiples simultanées et l'invalidation automatique des tokens compromis. Cette approche garantit l'isolation des données entre les différents clients de l'application.

Le PointageSecurityService représente l'innovation technique majeure du projet. J'ai développé un algorithme de détection d'anomalies qui analyse les patterns de pointage en temps réel : distances anormales entre les pointages, durées de travail atypiques, tentatives de pointage depuis des localisations non autorisées. Ce service utilise l'historique des pointages pour établir des profils comportementaux et déclencher des alertes automatiques en cas de déviation significative.

La logique de calcul des temps de travail était particulièrement complexe à cause des réglementations diverses selon les secteurs d'activité. J'ai conçu un système modulaire capable de gérer les spécificités légales (durée maximale de travail, pauses obligatoires, majorations d'heures supplémentaires) tout en permettant aux organisations de définir leurs propres règles métier.

L'architecture des repositories multi-tenant nécessitait une approche particulière des requêtes base de données. Chaque requête est automatiquement filtrée par l'identifiant d'organisation, et j'ai optimisé les index pour maintenir les performances malgré le partitionnement logique des données.

### 2. Précisez les moyens utilisés :

# DOSSIER PROFESSIONNEL (DP)

- Symfony 7.2 avec architecture service-oriented
- Doctrine ORM pour l'accès aux données
- PHP 8.2 avec attributs et types stricts
- Design Patterns : Repository, Service Layer, Factory
- Validation Symfony avec contraintes personnalisées
- Monolog pour la traçabilité des actions métier
- PHPUnit pour les tests unitaires (96 tests, 266 assertions)
- Swagger/OpenAPI pour la documentation des services

## 3. Avec qui avez-vous travaillé ?

Développement autonome avec revue de code par mes pairs en formation. J'ai appliqué les principes SOLID et les bonnes pratiques Symfony appris au cours de mon alternance.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Projet personnel / École La Plateforme

Chantier, atelier, service ► Projet CDA Timelys

Période d'exercice ► Du 01/01/2025 au 01/08/2025

## 5. Informations complémentaires (*facultatif*)

# DOSSIER PROFESSIONNEL (DP)

Les composants métier incluent une logique avancée de détection d'anomalies de pointage avec correction automatique (sortie forcée à 16h). Le système gère plus de 40 endpoints API avec validation complète et logging des actions administratives.

## Fichiers de référence :

- AuthenticationService.php : Gestion tokens avec rotation
- PointageSecurityService.php : Détection d'anomalies automatique
- Suite de tests complète : 96 tests unitaires et d'intégration

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1 Développer une application sécurisée (BC01)

*Exemple n°4 ► Contribuer à la gestion de projet*

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

La gestion du projet Timelys m'a placé dans une position de chef de projet technique où je devais orchestrer le développement d'une application complexe dans un délai contraint de 5 mois. L'enjeu était de livrer une solution production-ready en appliquant les bonnes pratiques de l'industrie tout en respectant les contraintes pédagogiques de validation des compétences CDA.

Dès la phase d'initialisation, j'ai structuré le projet selon une approche agile adaptée au développement solo. J'ai découpé le développement en sprints de 2 semaines avec des objectifs clairs : le premier sprint était consacré à la mise en place de l'architecture et de l'authentification, le second au développement des fonctionnalités de pointage, et ainsi de suite. Cette approche itérative m'a permis de valider rapidement les concepts techniques et d'ajuster le périmètre fonctionnel en fonction des contraintes découvertes.

La rédaction du cahier des charges technique représentait un défi particulier car je devais anticiper les besoins futurs d'une application multi-tenant tout en gardant un scope réalisable. J'ai analysé les solutions concurrentes (ClockWise, Timmi, Kelio) pour identifier les fonctionnalités essentielles et définir la proposition de valeur unique de Timelys. Cette phase de benchmark m'a permis de prioriser le développement des fonctionnalités différenciantes.

La gestion du backlog produit nécessitait une compréhension fine des enjeux métier. J'ai catégorisé les user stories selon leur valeur business et leur complexité technique, en priorisant les fonctionnalités core qui permettaient de valider l'hypothèse produit. La création d'un MVP fonctionnel dès le 3ème sprint m'a donné la flexibilité nécessaire pour affiner les fonctionnalités avancées.

L'aspect le plus technique de la gestion de projet concernait la mise en place d'un workflow Git robuste. J'ai implémenté une stratégie de branching adaptée au développement solo mais évolutive pour une équipe : branches feature pour chaque nouvelle fonctionnalité, branche dev pour l'intégration, et branche main pour la production. Les hooks de pré-commit automatisaient les tests de qualité code.

### 2. Précisez les moyens utilisés :

# DOSSIER PROFESSIONNEL (DP)

- Méthodologie Agile/Scrum adaptée au développement solo
- Git et GitHub pour le versioning et la gestion des branches
- Markdown pour la documentation technique
- Fly.io pour le déploiement automatisé
- PHPUnit et coverage reports pour le suivi qualité
- Postman Collections pour les tests d'intégration
- Trello/GitHub Projects pour le suivi des tâches
- Draw.io pour les schémas d'architecture

## 3. Avec qui avez-vous travaillé ?

Gestion de projet individuelle avec reporting régulier à mes formateurs. J'ai présenté l'avancement en sessions de suivi pédagogique et ajusté le planning selon les retours reçus.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Projet personnel / École La Plateforme

Chantier, atelier, service ► Projet CDA Timelys

Période d'exercice ► Du 01/01/2025 au 01/08/2025

## 5. Informations complémentaires (*facultatif*)

# DOSSIER PROFESSIONNEL (DP)

Le projet a été livré dans les délais avec toutes les fonctionnalités planifiées. La documentation technique produite comprend plus de 200 pages (architecture, API, déploiement, tests). Le projet est déployé en production et fonctionnel.

 Livrables produits :

- ARCHITECTURE\_TECHNIQUE\_CDA.md (294 lignes)
- SCHEMA\_BASE\_DONNEES\_TIMELYS.md (312 lignes)
- README\_DEPLOY.md (189 lignes)
- TESTS\_SUMMARY.md (191 lignes)
- API Swagger complète (40+ endpoints)

# DOSSIER PROFESSIONNEL (DP)

## Activité-type

2

### Concevoir et développer une application sécurisée organisée en couches (BC02)

*Exemple n°1 ► Analyser les besoins et maquetter l'application*

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

L'analyse des besoins pour Timelys nécessitait une compréhension approfondie des problématiques terrain rencontrées par les entreprises dans le suivi des temps de travail. Mon approche a consisté à partir des pain points concrets identifiés lors de mes recherches : erreurs de saisie manuelle, absence de traçabilité géographique, difficultés de consolidation des données, et manque de visibilité temps réel sur les activités.

La phase d'identification des parties prenantes m'a révélé la complexité des enjeux organisationnels. Au-delà des trois rôles principaux (employés, superviseurs, administrateurs), j'ai identifié des acteurs indirects comme les équipes RH qui exploitent les données de pointage, les clients finaux qui valident les interventions, et les organismes de contrôle qui peuvent auditer les temps déclarés. Cette vision systémique m'a permis de concevoir une architecture qui prend en compte l'écosystème complet.

Le développement des user stories s'appuyait sur une méthode de Story Mapping qui m'a permis de visualiser le parcours utilisateur complet depuis la connexion jusqu'à la génération de rapports. J'ai particulièrement travaillé sur les scénarios d'exception : que se passe-t-il en cas de panne réseau ? Comment gérer un oubli de pointage ? Ces questions ont façonné les fonctionnalités de résilience de l'application.

La création des maquettes avec Figma représentait un défi d'ergonomie mobile. Les contraintes terrain (utilisation avec des gants, luminosité variable, stress temporel) m'ont amené à privilégier une approche mobile-first avec des interfaces épurées et des zones tactiles généreuses. J'ai créé un design system cohérent basé sur Material Design 3, adapté aux codes visuels du secteur d'activité.

L'architecture fonctionnelle multi-tenant était particulièrement complexe à concevoir car elle devait permettre l'isolation des données tout en mutualisant l'infrastructure. J'ai défini une stratégie de partitionnement logique avec des règles métier configurables par organisation, tout en maintenant des performances optimales même avec une montée en charge importante.

#### 2. Précisez les moyens utilisés :

# DOSSIER PROFESSIONNEL (DP)

- Figma pour la création des maquettes et du design system
- Méthode User Story Mapping pour l'analyse fonctionnelle
- Diagrammes de cas d'usage UML
- Personas et Customer Journey Mapping
- Benchmarking d'applications similaires
- Material Design Guidelines pour l'interface mobile
- Workshops d'idéation et de validation des besoins
- Documentation structurée des spécifications fonctionnelles

## 3. Avec qui avez-vous travaillé ?

Analyse des besoins menée individuellement avec validation par mes formateurs qui ont joué le rôle de product owners. J'ai également échangé avec mes pairs pour valider l'ergonomie des maquettes.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Projet personnel / École La Plateforme

Chantier, atelier, service ► Projet CDA Timelys

Période d'exercice ► Du 01/01/2025 au 01/08/2025

## 5. Informations complémentaires (*facultatif*)

# DOSSIER PROFESSIONNEL (DP)

L'analyse a révélé des besoins complexes de multi-tenancy et de gestion des rôles. Les maquettes ont été conçues pour 3 types d'utilisateurs avec des interfaces adaptées à chaque contexte métier.

 Maquettes créées :

- Interface de connexion et authentification
- Tableaux de bord par rôle (user/admin/superadmin)
- Écrans de pointage et relevés terrain
- Interfaces d'administration et monitoring

# DOSSIER PROFESSIONNEL (DP)

## Activité-type

2

Concevoir et développer une application sécurisée organisée en couches (BC02)

*Exemple n°2 ► Définir l'architecture logicielle multicouche*

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

J'ai conçu l'architecture complète de l'application Timelys selon le pattern multicouches :

- Définition de l'architecture 3-tiers : Présentation (Flutter) / Métier (Symfony) / Données (MySQL)
- Conception de l'architecture API REST avec séparation claire des responsabilités
- Implémentation du pattern MVC côté backend avec services métier dédiés
- Architecture MVVM côté frontend avec Provider pour la gestion d'état
- Définition des interfaces et contrats entre les couches
- Conception de l'architecture de sécurité avec authentification JWT
- Mise en place de l'architecture multi-tenant avec isolation des données
- Documentation complète de l'architecture technique

### 2. Précisez les moyens utilisés :

- Patterns architecturaux : MVC, MVVM, Repository, Service Layer
- Diagrammes d'architecture (C4 Model, diagrammes de composants)
- Symfony Framework pour l'architecture backend
- Flutter/Dart pour l'architecture frontend
- API-First design avec OpenAPI/Swagger
- Principes SOLID pour la conception des services
- Clean Architecture pour la séparation des préoccupations
- Draw.io pour la modélisation architecturale

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Conception architecturale individuelle avec revue et validation par l'équipe pédagogique. J'ai présenté l'architecture en session de validation technique avec mes formateurs.

## 4. Contexte

**Nom de l'entreprise, organisme ou association ▶ Projet personnel / École La Plateforme**

**Chantier, atelier, service** ▶ Projet CDA Timelys

**Période d'exercice** ▶ **Du** 01/01/2025 **au** 01/08/2025

## 5. Informations complémentaires (*facultatif*)

L'architecture supportera une montée en charge jusqu'à plusieurs milliers d'utilisateurs simultanés. Elle respecte les principes de scalabilité horizontale et de haute disponibilité.

📄 Livrables architecturaux :

- ARCHITECTURE\_TECHNIQUE\_CDA.md : Documentation complète 294 lignes
- Diagrammes d'architecture système et applicative
- Spécifications techniques par couche
- Matrice de traçabilité besoins/architecture

# DOSSIER PROFESSIONNEL (DP)

## Activité-type

2

### Concevoir et développer une application sécurisée organisée en couches (BC02)

*Exemple n°3 ► Concevoir et mettre en place une base de données relationnelle*

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

La conception de la base de données pour Timelys représentait un défi architectural majeur : créer un modèle de données capable de supporter une application multi-tenant tout en maintenant des performances optimales et une intégrité référentielle stricte. L'enjeu était de concevoir une structure suffisamment flexible pour s'adapter aux spécificités métier de chaque organisation, tout en évitant la complexité excessive qui pourrait nuire à la maintenance.

Ma première préoccupation concernait la stratégie de multi-tenancy. Après analyse des différentes approches (base séparée, schéma séparé, ligne séparée), j'ai opté pour une isolation au niveau des données avec un discriminant organization\_id sur chaque table. Cette approche offrait le meilleur compromis entre isolation des données, simplicité de déploiement et optimisation des coûts d'infrastructure.

La modélisation conceptuelle s'articulait autour de 10 entités principales interconnectées : Organization (tenant principal), User (avec héritage pour les différents rôles), Site (localisation d'intervention), Pointage (coeur métier), Session (gestion des périodes de travail), Report (relevés terrain), Notification (alertes système), Configuration (paramètres tenant), AuditLog (traçabilité), et RefreshToken (sécurité). La complexité résidait dans les relations many-to-many entre User et Organization (un utilisateur peut appartenir à plusieurs organisations) et dans la gestion temporelle des sessions de travail.

L'étape de normalisation en 3NF m'a amené à identifier et éliminer les redondances potentielles. Par exemple, j'ai créé une table séparée pour les types de pointage (entrée, sortie, pause) plutôt que d'utiliser un enum, permettant ainsi une configuration dynamique par organisation. Cette décision architecturale facilitait l'évolutivité du système.

L'implémentation physique avec MySQL 8.4 nécessitait une attention particulière aux performances. J'ai créé des index composites stratégiques, notamment sur (organization\_id, user\_id, created\_at) pour optimiser les requêtes de listing des pointages. La contrainte principale était de s'assurer que chaque requête utilisait efficacement les index même avec un volume de données important.

#### 2. Précisez les moyens utilisés :

# DOSSIER PROFESSIONNEL (DP)

- MySQL 8.4 comme SGBD relationnel
- Doctrine ORM avec système de migrations
- MySQL Workbench pour la modélisation visuelle
- Méthode MERISE pour la conception MCD/MLD
- Symfony Migrations pour le versioning du schéma
- phpMyAdmin pour l'administration de la base
- Analyse des performances avec EXPLAIN queries
- Outils de reverse engineering pour la documentation

## 3. Avec qui avez-vous travaillé ?

Conception et implémentation individuelle de la base de données avec validation du modèle de données par mes formateurs.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Projet personnel / École La Plateforme

Chantier, atelier, service ► Projet CDA Timelys

Période d'exercice ► Du 01/01/2025 au 01/08/2025

## 5. Informations complémentaires (*facultatif*)

# DOSSIER PROFESSIONNEL (DP)

La base supporte nativement le multi-tenant avec isolation parfaite des données par company\_id. Elle est optimisée pour les requêtes temps réel de pointage et les rapports statistiques.

📄 Schéma produit :

- SCHEMA\_BASE\_DONNEES\_TIMELYS.md : 312 lignes de documentation
- 10 tables principales avec relations optimisées
- Index de performance pour 100+ utilisateurs simultanés
- Contraintes d'intégrité métier complètes
- Migration consolidée Version20250727220314.php

# DOSSIER PROFESSIONNEL (DP)

## Activité-type

2

Concevoir et développer une application sécurisée organisée en couches (BC02)

*Exemple n°4 ► Développer des composants d'accès aux données SQL et NoSQL*

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

J'ai développé la couche d'accès aux données complète de l'application Timelys :

- Création de 10 repositories Doctrine avec requêtes optimisées
- Implémentation de requêtes complexes SQL pour les calculs de temps de travail
- Développement des méthodes d'agrégation pour les statistiques temps réel
- Création de requêtes multi-tenant avec filtrage automatique par company\_id
- Optimisation des performances avec pagination et eager loading
- Implémentation du pattern Repository avec interface abstraite
- Développement des requêtes de reporting avec GROUP BY complexes
- Gestion du cache applicatif pour les données fréquemment consultées

### 2. Précisez les moyens utilisés :

- Doctrine ORM pour l'abstraction de la couche données
- Query Builder Doctrine pour les requêtes complexes
- DQL (Doctrine Query Language) pour les requêtes métier
- SQL natif pour les requêtes d'agrégation avancées
- Indexes composites MySQL pour les performances
- Symfony Cache Component pour la mise en cache
- Repository Pattern pour l'abstraction des données
- Fixtures et DataFixtures pour les jeux de données de test

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Développement autonome des repositories avec optimisations validées par mes formateurs.

## 4. Contexte

**Nom de l'entreprise, organisme ou association ► Projet personnel / École La Plateforme**

**Chantier, atelier, service** ► Projet CDA Timelys

**Période d'exercice** ► **Du** 01/01/2025 **au** 01/08/2025

## 5. Informations complémentaires (*facultatif*)

Les repositories incluent des fonctionnalités avancées comme le calcul automatique des temps de travail avec gestion des pauses et la détection d'anomalies de pointage.

📄 Repositories développés :

- UserRepository : Recherche multi-critères avec pagination
- PointageRepository : Calculs de temps complexes et détection d'anomalies
- AccessTokenRepository : Gestion sécurisée des tokens avec nettoyage automatique
- CompanyRepository : Agrégations statistiques multi-tenant
- Requêtes optimisées : temps de réponse < 200ms (95e percentile)

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3 Préparer le déploiement d'une application sécurisée (BC03)

*Exemple n°1 ► Préparer et exécuter les plans de tests*

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

La mise en place d'une stratégie de tests pour Timelys était cruciale pour garantir la fiabilité d'une application gérant des données sensibles de temps de travail. L'enjeu dépassait la simple validation fonctionnelle : il fallait s'assurer que l'application pouvait résister aux tentatives de fraude, maintenir ses performances sous charge, et préserver l'intégrité des données dans tous les scénarios d'usage.

Ma approche de testing s'articulait autour de plusieurs niveaux complémentaires. Au niveau unitaire, j'ai développé 96 tests couvrant l'ensemble de la logique métier, avec une attention particulière sur les algorithmes de détection d'anomalies et de calcul des temps de travail. Ces tests étaient particulièrement complexes car ils devaient simuler des scénarios temporels réalistes : oubli de pointage, chevauchements de sessions, corrections manuelles, etc.

Les tests d'intégration API représentaient un défi technique spécifique au contexte multi-tenant. Chaque endpoint devait être testé avec différents contextes d'organisation pour vérifier l'isolation des données. J'ai créé 40+ collections Postman qui automatisaient ces tests avec des jeux de données réalistes, incluant des cas limites comme les changements de fuseau horaire ou les pointages simultanés multiples.

La sécurité étant un enjeu critique, j'ai développé une suite de tests spécialisés pour l'authentification JWT. Ces tests validaient non seulement le bon fonctionnement nominal (connexion, refresh token, déconnexion) mais aussi les scénarios d'attaque : tokens expirés, signatures invalides, tentatives de réutilisation de tokens révoqués, injections de données dans les claims JWT.

L'aspect performance nécessitait une approche particulière car l'application devait supporter des pics de charge importants (tous les employés d'une entreprise pointant simultanément). J'ai développé des tests de montée en charge simulant jusqu'à 1000 utilisateurs concurrents, avec monitoring des temps de réponse et de la consommation mémoire. Ces tests ont révélé des goulots d'étranglement dans certaines requêtes complexes, que j'ai pu optimiser avant la mise en production.

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

- PHPUnit pour les tests unitaires et d'intégration PHP
- Postman Collections pour les tests d'API REST
- Faker pour la génération de données de test
- Doctrine Fixtures pour les jeux de données
- Symfony Test Framework pour les tests fonctionnels
- Coverage analysis avec Xdebug
- GitHub Actions pour l'exécution automatique des tests
- Mockery pour les tests avec mocks et stubs

## 3. Avec qui avez-vous travaillé ?

Développement des tests en autonomie avec validation des stratégies de test par mes formateurs. J'ai appliqué les bonnes pratiques TDD apprises en formation.

## 4. Contexte

**Nom de l'entreprise, organisme ou association ► Projet personnel / École La Plateforme**

**Chantier, atelier, service** ► Projet CDA Timelys

**Période d'exercice** ► **Du** 01/01/2025 **au** 01/08/2025

# DOSSIER PROFESSIONNEL (DP)

## 5. Informations complémentaires (*facultatif*)

La suite de tests couvre 100% des fonctionnalités critiques (authentification, pointage, sécurité). L'exécution complète prend moins de 5 secondes pour une feedback loop rapide.

### Métriques de tests :

- 96 tests avec 266+ assertions
- Couverture des composants : Entités (44 tests), Services (10 tests), Sécurité (20 tests), Repositories (22 tests)
- Tests de performance : < 200ms (95e percentile)
- TESTS\_SUMMARY.md : Documentation complète 191 lignes

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3 Préparer le déploiement d'une application sécurisée (BC03)

*Exemple n°2 ► Préparer et documenter le déploiement de l'application*

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

J'ai préparé et documenté le déploiement complet de l'application Timelys sur Fly.io :

- Configuration des environnements de déploiement (preprod/prod)
- Création de scripts de déploiement automatisé avec gestion des migrations
- Documentation complète des procédures de déploiement
- Configuration des variables d'environnement sécurisées
- Mise en place des sauvegardes automatiques de base de données
- Configuration du monitoring et des alertes de production
- Création de runbooks pour les opérations courantes
- Documentation des procédures de rollback en cas de problème

### 2. Précisez les moyens utilisés :

- Fly.io comme plateforme de déploiement cloud
- Docker pour la conteneurisation de l'application
- Scripts Bash pour l'automatisation du déploiement
- Fly.io CLI pour la gestion des déploiements
- Variables d'environnement sécurisées avec Fly Secrets
- MySQL volumes persistants pour les données
- GitHub pour le versioning et les triggers de déploiement
- Monitoring intégré Fly.io avec alertes

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Préparation autonome du déploiement avec validation des configurations par mes formateurs DevOps.

## 4. Contexte

**Nom de l'entreprise, organisme ou association ▶ Projet personnel / École La Plateforme**

**Chantier, atelier, service** ▶ Projet CDA Timelys

**Période d'exercice** ▶ **Du** 01/01/2025 **au** 01/08/2025

## 5. Informations complémentaires (*facultatif*)

L'application est déployée en production avec 99.5% de disponibilité. Le processus de déploiement est entièrement automatisé avec validation des migrations et rollback automatique en cas d'erreur.

📄 Documentation produite :

- README\_DEPLOY.md : 189 lignes de procédures détaillées
- Scripts deploy-release.sh avec gestion d'erreurs complète
- Configuration fly.toml pour preprod et production
- Documentation des variables d'environnement
- Procédures de monitoring et alerting

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3 Préparer le déploiement d'une application sécurisée (BC03)

*Exemple n°3 ► Contribuer à la mise en production dans une démarche DevOps*

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

J'ai mis en place une chaîne DevOps complète pour l'application Timelys :

- Configuration d'un pipeline CI/CD avec GitHub Actions
- Automatisation des tests et du déploiement sur commit
- Mise en place du monitoring applicatif avec métriques temps réel
- Configuration des alertes automatiques en cas de dysfonctionnement
- Implémentation du logging centralisé avec Monolog
- Mise en place de la surveillance des performances (temps de réponse, erreurs)
- Configuration de la haute disponibilité avec scaling automatique
- Documentation des processus de maintenance et de monitoring

### 2. Précisez les moyens utilisés :

- GitHub Actions pour l'intégration continue
- Fly.io pour le déploiement automatique et le scaling
- Monolog pour le logging centralisé application
- Fly.io Monitoring pour les métriques infrastructure
- Alerting automatique via email/webhook
- Docker pour la standardisation des environnements
- Scripts de maintenance automatisés
- Documentation as Code avec Markdown

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Mise en place DevOps en autonomie avec validation des pratiques par mes formateurs experts en DevOps. J'ai appliqué les méthodologies Agile et DevOps enseignées.

## 4. Contexte

**Nom de l'entreprise, organisme ou association ▶ Projet personnel / École La Plateforme**

**Chantier, atelier, service** ▶ Projet CDA Timelys

**Période d'exercice** ▶ **Du** 01/01/2025 **au** 01/08/2025

## 5. Informations complémentaires (facultatif)

L'application est en production avec un SLA de 99.5% de disponibilité. Le monitoring détecte automatiquement les anomalies et déclenche les alertes appropriées.

🔧 Outils DevOps mis en place :

- Pipeline CI/CD complet avec tests automatiques
- Déploiement zero-downtime avec migration automatique
- Monitoring 24/7 avec alerting intelligent
- Logs centralisés avec recherche et analyse
- Scaling automatique selon la charge
- Sauvegarde automatique des données

## **DOSSIER PROFESSIONNEL (DP)**

## **Titres, diplômes, CQP, attestations de formation**

(facultatif)

# DOSSIER PROFESSIONNEL (DP)

## Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Enzo Mattio .....

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 Auriol ..... le 28 juillet 2025.....

pour faire valoir ce que de droit.

Signature :



## **DOSSIER PROFESSIONNEL (DP)**

## **Documents illustrant la pratique professionnelle**

(facultatif)

# **DOSSIER PROFESSIONNEL (DP)**

## **ANNEXES**

