



# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance* ▶ Verdier  
*Nom d'usage* ▶ Verdier  
*Prénom* ▶ Lucas  
*Adresse* ▶ 109, avenue Clot Bey, 13008 Marseille.

## Titre professionnel visé

*Concepteur Développeur d'Applications*

### MODALITÉ D'ACCÈS :

- ☒ 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.*



## Sommaire

### Exemples de pratique professionnelle

<b>Intitulé de l'activité-type n° 1</b>	<b>p.</b>	<b>5</b>
- Intitulé de l'exemple n° 1	p. p.	
- Intitulé de l'exemple n° 2	p. p.	
- Intitulé de l'exemple n° 3	p p.	
<b>Intitulé de l'activité-type n° 2</b>	<b>p.</b>	
- Intitulé de l'exemple n° 1	p. p.	
- Intitulé de l'exemple n° 2	p. p.	
- Intitulé de l'exemple n° 3	p p.	
<b>Intitulé de l'activité-type n° 3</b>	<b>p.</b>	
- Intitulé de l'exemple n° 1	p. p.	
- Intitulé de l'exemple n° 2	p. p.	
- Intitulé de l'exemple n° 3	p p.	
<b>Titres, diplômes, CQP, attestations de formation</b> <i>(facultatif)</i>	<b>p.</b>	
<b>Déclaration sur l'honneur</b>	<b>p.</b>	
<b>Documents illustrant la pratique professionnelle</b> <i>(facultatif)</i>	<b>p.</b>	
<b>Annexes</b> <i>(Si le RC le prévoit)</i>	<b>p.</b>	

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## Activité-type 1

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

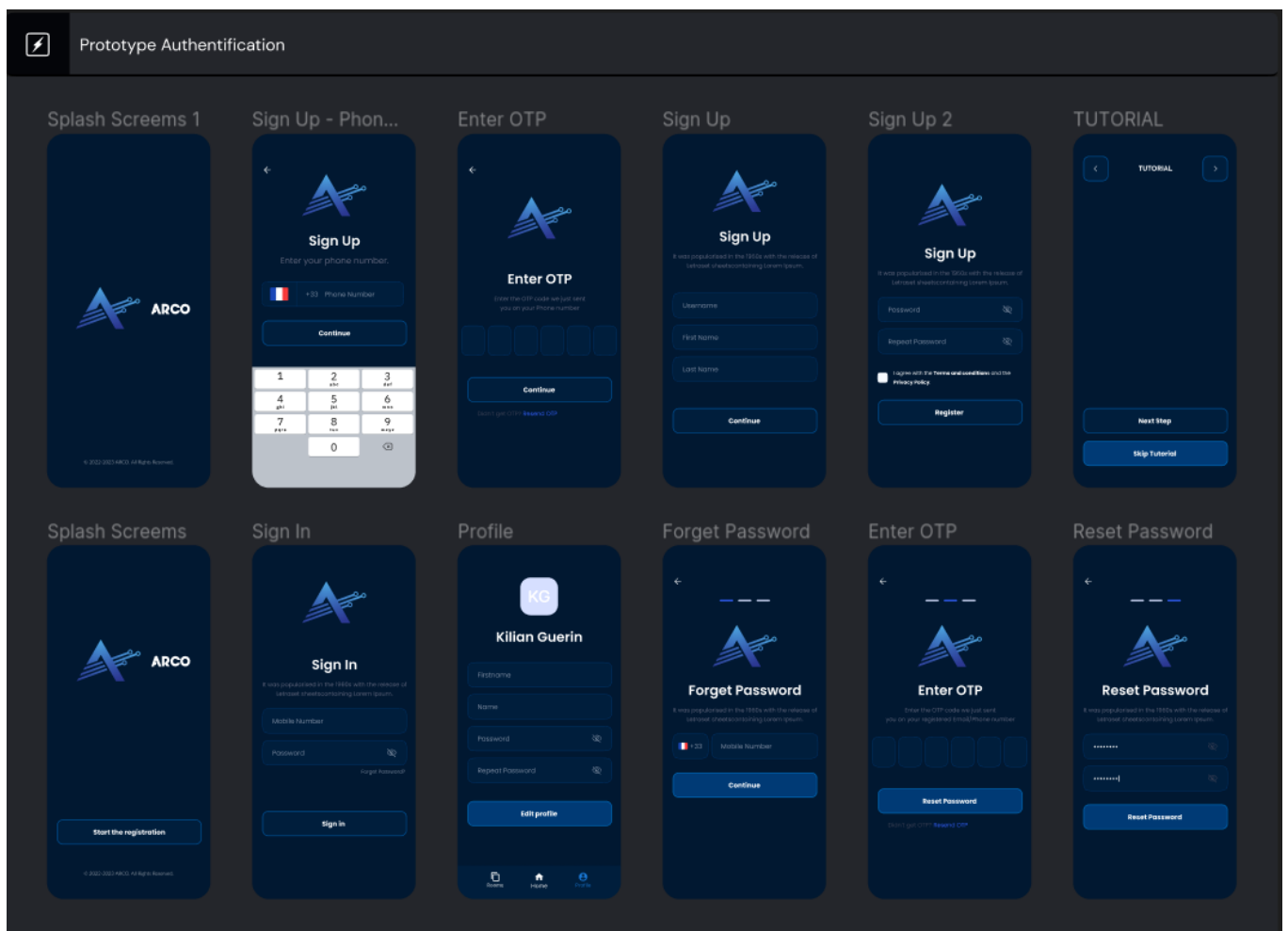
Exemple n°1 - Projet application mobile "Arco"

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

Dans le cadre du projet de la réalisation d'une application mobile à La Plateforme, j'ai eu l'occasion de **maquetter mon application** avant de développer les écrans.

Avant de nous lancer dans la réalisation des maquettes nous avons tout d'abord décidé d'un code couleurs : #FFFFFF, #001E3C et #003A75.

L'outil **Figma** m'a permis de travailler de manière collaborative avec mes coéquipiers, ce qui a facilité l'échange de nos idées.



Compétences REAC couvertes par ce projet :

- Maquetter une application

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement
- Concevoir une application

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- OS : Fedora
- Figma : maquettes
- Trello : gestion de projet

## 3. Avec qui avez-vous travaillé ?

Equipe projet "ARCO" :

- Kilian Guerin
- Bassem Sataf
- Badr Sebaa

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Equipe projet ARCO

Chantier, atelier, service ▶ ARCO

Période d'exercice ▶ Du : 02/01/23 au : 27/01/23

## 5. Informations complémentaires (facultatif)

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

## Projet desktop Python "Sokoban"

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

Dans le cadre scolaire nous avons dû développer une application desktop en Python. C'est un jeu Sokoban qui se lance en mode fenêtre, où l'utilisateur peut interagir avec l'interface via des événements claviers.

Pour pouvoir gagner une partie, l'utilisateur dispose d'un personnage, qu'il peut déplacer grâce aux touches directionnelles du clavier, et doit déplacer toutes les caisses sur les cibles.

### Fichier de configuration de l'application (settings.py):

Pour réaliser cette application et notamment gérer les collisions entre le personnage et les caisses j'ai utilisé la librairie PyGame qui vient avec beaucoup de fonctions utiles à la réalisation d'un jeu.

J'ai développé cette application en Programmation Orientée Objet (POO). La POO permet par la création de classe pour chaque composants du jeu (Personnage, Mur, Cibles, Caisse) une implémentation simple et facilement modulable grâce aux attributs et méthodes de classe, ainsi qu'à l'héritage qui permet de faire hériter des propriétés depuis une classe mère vers une classe fille.

Dans notre exemple j'ai une classe Item qui a pour attribut une superficie, une position (x,y), une couleur etc... Et ma classe Character hérite de Item, de ce fait je peux utiliser les mêmes attributs que sa classe mère (exemple en 'informations complémentaires').

Exemple de code qui gère les collision personnage - caisses :

```
def update(self, event):
    self.player.move_player(event)
    pygame.draw.rect(self.surface, (255, 255, 255), (self.player.last_move_x,
self.player.last_move_y, CELL_WIDTH, CELL_HEIGHT))
    for wall in self.wall_list:
        if pygame.Rect.collidect(self.player.rect, wall.rect):
            self.player.previous_pos()

    for box in self.box_list:
        for hole in self.hole_list:
            if pygame.Rect.collidect(box.rect, hole.rect):
                self.hole_list.remove(hole)
            if pygame.Rect.collidect(self.player.rect, box.rect):
                box.move(event)
                box.draw_box()

    self.player.draw_player()

    if len(self.hole_list) <= 0:
        print('YOU WIN!!')
```

Compétences REAC validées par ce projet :

- Développer une interface utilisateur de type desktop

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- OS: Fedora
- IDE: PyCharm
- PyGame

## 3. Avec qui avez-vous travaillé ?

Seul.



# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Projet scolaire solo.*

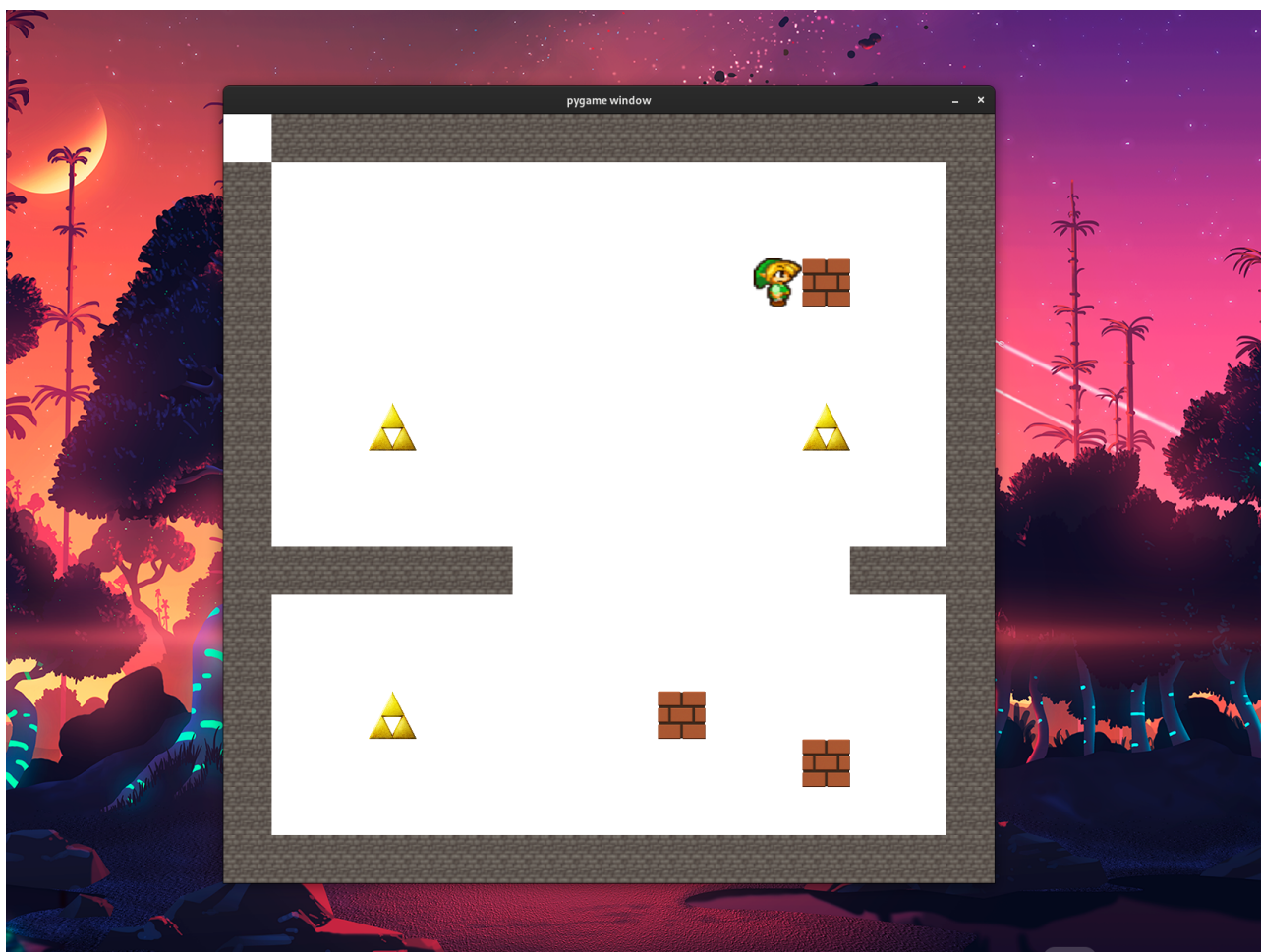
Chantier, atelier, service ▶ *Réaliser un jeu 'SOKOBAN' en python*

Période d'exercice ▶ Du : 17/04/23 au : 22/04/23

## 5. Informations complémentaires (facultatif)

```
class Character(Item):
    last_move_x = 0
    last_move_y = 0

    def __init__(self, surface, pos_x, pos_y, width, height, color):
        super().__init__(surface, pos_x, pos_y, width, height, color)
        self.rect = pygame.draw.rect(self.surface, self.color, (self.pos_x, self.pos_y, self.width,
self.height))
        self.image = pygame.transform.scale(pygame.image.load('textures/link_f0.png'),
                                            (self.width, self.height))
```



## Activité-type 1 Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n° 3 - Projet "ViaxeoFiles"

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

**ViaxeoFiles** est le projet fil rouge qui m'a été confié par l'entreprise où j'effectue mon alternance : **Viaxoft**.

C'est une PME qui développe des applications web destinées à **accompagner les entreprises du tourisme dans leur quotidien** (plateforme de réservation de prestations aériennes, hôtelière et terrestre, facturation, gestion des dossiers clients, business decision...).

ViaxeoFiles est un micro-service faisant partie d'un projet d'envergure de **migration de nos outils et écrans actuels vers une nouvelle plateforme basée sur des technologies plus récentes**. L'existant est une application avec un back JAVA et un front GWT.

La nouvelle plateforme est **développée en JAVA Springboot (Spring WebFlux) et un front en REACT JS**. J'ai donc la tâche d'en développer une partie avec ViaxeoFiles. L'application est destinée à nos clients mais aussi à nos collaborateurs en interne. C'est un **système performant d'import de fichiers CSV, XML et JSON**.

**La première étape consiste à importer le fichier** (au clic ou drag'n'drop). L'accent est mis sur la performance car nos clients ont des fichiers très volumineux.

Lors de la **seconde étape**, appelée "étape de mapping de données", le client **relie ses données** (ses headers) avec **nos headers stockés en base de données** au moyen d'un schéma d'import (voir photo). Une fois fait, un **lecture totale du fichier est faite côté back**, afin de relever toutes les erreurs possible dans les champs :

- erreur de typage (exemple: un int est renseigné alors qu'une string est attendue, pour un prénom par exemple).
- erreur de length (une string de minimum 2 caractères est attendue, un seul est fourni).
- une valeur est vide alors que cette dernière est annotée "required" par les schéma d'import.

Pour l'instant si le fichier présente des erreurs, l'upload final est bloqué, et un **rapport d'erreur** est rédigé puis affiché au client avec la valeur en erreur, le header correspondant, le type d'erreur et la ligne où l'erreur est présente.

S'il n'y a pas d'erreur, le fichier est **importé en base de données, avec les headers modifiés** correspondant à ceux présents en bdd et sélectionnés par l'utilisateur.

Au **second import d'un fichier déjà uploadé en base**, un **remplissage automatique des champs est effectué** afin de faciliter les imports futur d'un même fichier. Il est cependant toujours possible de modifier le mapping, mapping qui sera ensuite mis à jour en base afin de le proposer automatiquement

au prochain import.

Tout au long de l'import une **gestion de l'encodage est effectuée**. A savoir qu'à la première étape l'encodage du fichier en input est **détecté**, et à l'upload final, le fichier est **converti dans l'encodage "cible"** définie dans le schéma d'import.

Sur ce projet j'ai effectué les tâches suivantes :

## 1. Front

- Réaliser tous les **écrans du processus complet d'import (création de composants réutilisables)**.
- **Appels API** afin de réceptionner les réponses du back et les afficher à l'utilisateur. Mais aussi, **recueillir, trier et formater les données générées par les actions de l'utilisateur** et les **envoyer au bon format au back moyennant des requêtes API**.
- Gestion et affichage des erreurs.

## 2. Back

- **Création de routes API permettant d'interagir avec le côté client**. L'API interagit avec une **base de données PostgreSQL**.
- Des **méthodes services** permettant de **créer le fichier initial en base**, cette dernière **détecte l'encodage du fichier** et si les headers du fichier en input sont présents dans notre table file-mapping. Une fois ce traitement effectué, une **réponse est envoyée au front** pour les étapes utilisateurs suivantes et le fichier est **stocké dans un bucket S3**.
- Des **méthodes qui gèrent l'import final et réécrivent le fichier client**. Ces dernières réécrivent le fichier avec les headers mappés par le front, **lisent l'intégralité du fichier** afin de vérifier si des erreurs sont présentes et ainsi **lever des exceptions**. Si il n'y a pas d'erreurs, le fichier est **uploadé sur le bucket S3**, si c'est un **premier import**, un **nouvelle ligne est créée dans la table file-mapping** qui stocke le mapping, si c'est un **second** (ou plus) **import** la ligne associée dans la table file-mapping et **mise à jour avec le nouveau mapping** (si des modifications ont été effectuées par l'utilisateur).
- **Gestion de l'encodage, détection et réécriture selon l'encodage cible**.
- **Génération d'un rapport d'erreur** renvoyé via une Exception côté client.
- Création et modifications de **tests unitaires** JAVA (JUnit/Mockito) afin de tester les nouvelles classes et méthodes implémentées.

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- Environnement : Ubuntu, IntelliJ.
- Technologies : JAVA Spring WebFlux, ReactJS.
- Base de données : PostgreSQL.
- FastCSV (librairie JAVA qui gère la lecture et réécriture de fichier csv).
- Minio (bucket S3 où sont stockés les fichiers).
- Material UI (libraires de composant JS pour le front).

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

- Junit et Mockito (permet de créer des mock) pour les tests unitaires.

### 3. Avec qui avez-vous travaillé ?

#### En équipe avec :

- Mr. Julien Torrielli (développeur et également maître de stage).
- Mr. Jean-Michel François (product owner).

### 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Viaxoft

Chantier, atelier, service ▶ ViaxeoFiles

Période d'exercice ▶ Année scolaire 2022 - 2023.

### 5. Informations complémentaires *(facultatif)*

## Activité-type 2 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 1 - Projet application mobile "ARCO"

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

Dans le cadre du développement de notre application mobile, nous avons développé une API en NodeJs pour interagir avec notre base de données (MongoDB) et envoyer/recevoir ces données côté client (front React Native).

Pour faciliter le travail en équipe dès l'API terminée, je l'ai déployé sur Plesk.

L'API comprend :

- des routes permettant l'accès aux données (GET, POST, PUT, DELETE).
- un middleware permettant de gérer l'authentification grâce à un Token d'identification utilisateur et un refresh Token qui s'update tous les 'N temps' afin de re-vérifier l'intégrité de l'utilisateur.
- API déployé sur un hébergeur (Plesk). Exemple d'URL :  
"<https://lucas-verdier.students-laplateforme.io/api/message/room/:id>" (cette route nécessite le token d'identification + le refresh token).

Webistes & Domains >

### Node.js on [lucas-verdier.students-laplateforme.io](https://lucas-verdier.students-laplateforme.io)

Restart App NPM install Run script Disable Node.js

Node.js Version	<a href="#">16.20.0</a>
Package Manager	<a href="#">npm</a> ⓘ This is what we detected, you can change it
Document Root	<a href="#">/httpdocs/app-mobile-chat</a>
Application Mode	<a href="#">development</a>
Application URL	<a href="http://lucas-verdier.students-laplateforme.io">http://lucas-verdier.students-laplateforme.io</a>
Application Root	<a href="#">/httpdocs/app-mobile-chat</a> [open]
Application Startup File	<a href="#">server.js</a> [edit]
Custom environment variables	<ul style="list-style-type: none"><li>- URL: mongodb+srv://Admin: [redacted]@app-mobile-chat.nkx8vhk.mongodb.net/ARCO?retryWrites=true&amp;w=majority</li><li>- ACCESS_TOKEN_SECRET: [redacted]</li><li>- REFRESH_TOKEN_SECRET: [redacted]</li><li>- ALGO_MSG: [redacted]</li><li>- KEY_MSG: [redacted]</li></ul> <a href="#">[specify]</a>

Variables d'environnement

Exemples de routes :

```
router.post('/send', auth, sendMessage);
router.delete('/delete', auth, deleteMessage);
router.get('/all', auth, getAllRoomMessages);
router.get('/room/:id', auth, getMessagesByRoomId);
router.get('/decrypt/:id', auth, decryptMessage)
```

Fonctionnement de l'API :

- L'API fait le lien de manière sécurisée entre la base de données et le côté client.

MongoDB <—> API NodeJS <—> Front React Native

Compétences REAC validées par ce projet :

- Construire une application organisée en couches.
- Préparer et exécuter les déploiement d'une application.
- Développer des composants d'accès aux données.
- Développer des composants dans le langage d'une base de données.

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- OS : Fedora
- IDE: IntelliJ
- NodeJS
- Plesk
- MongoDB

## 3. Avec qui avez-vous travaillé ?

Equipe projet "ARCO" :

- Kilian Guerin

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

- Bassem Sataf
- Badr Sebaa

### 4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Equipe projet ARCO*

Chantier, atelier, service ▶ *ARCO*

Période d'exercice ▶ Du : *02/01/23* au : *27/01/23*

### 5. Informations complémentaires *(facultatif)*



## Activité-type 3

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

Exemple n° 1 - Projet application mobile "ARCO"

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

Dans le cadre de notre projet de réalisation d'une application mobile, nous avons dû concevoir une base de données afin d'envoyer et recevoir des données vers/depuis l'API.

Ayant à développer une application mobile qui a pour fonction métier un chat général simple en temps réel nous avons opté pour une base de données NoSQL (Not Only Structured Query Language), en l'occurrence MongoDB.

Avantages d'une base de données NoSQL pour une application mobile de chat:

- Modèle de données flexibles: les données peuvent être facilement ajoutées et/ou supprimées sans impacter la structure de la base de données.
- Facilement scalable: permet de gérer un très grand flux de données.
- Disponibilité de la données. Le NoSQL permet d'avoir de très bonnes performances sur la vitesse d'accès aux données.
- Il est aussi possible de travailler avec des données structurées (Not ONLY sql).

Cependant si l'appli avait été plus complexe et nécessitait des requêtes plus complexes, avec beaucoup de jointures par exemple, nous aurions opté pour une base de données SQL.

Exemples de requêtes vers la base de données :

```
const createChatRoom = async (req, res) => {
  const { name, usersInRoom } = req.body;
  try {
    if (checkRole(req.auth.role)) {
      const chatRoom = new ChatRoom({ name, usersInRoom });
      await chatRoom.save();
      res.status(201).json({ chatRoom });
    }
  } catch (err) {
    res.status(401).json({ err });
  }
};
```

```
const getMessagesByRoomId = async (req, res) => {
  try {
    if (!req.params.hasOwnProperty('id')) {
      return res.status(400).json({ message: 'Missing room ID' });
    }

    const messages = await Message.find({chatroom_id: req.params.id});
    let decryptedObj = [];
    for (let i = 0; i < messages.length; i++) {
      const decryptedText = await decryptMessage(messages[i]._id);
      decryptedObj.push([
        decryptedText,
        messages[i].user_id,
        messages[i].createdAt,
      ]);
    }
    res
      .status(200)
      .json({
        status: "ok",
        message: `Messages de la room ${req.params.id}`,
        data: decryptedObj,
      });
  } catch (error) {
    res.status(500).json({status: "Erreur.", message: error.message });
  }
}
```

Compétences REAC validées par le projet :

- Concevoir une base de données.
- Mettre en place une base de données.
- Développer des composants dans le langage d'une base de données

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- OS : Fedora
- IDE: IntelliJ
- NodeJS

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

- Plesk
- MongoDB

## 3. Avec qui avez-vous travaillé ?

Equipe projet "ARCO" :

- Kilian Guerin
- Bassem Sataf
- Badr Sebaa

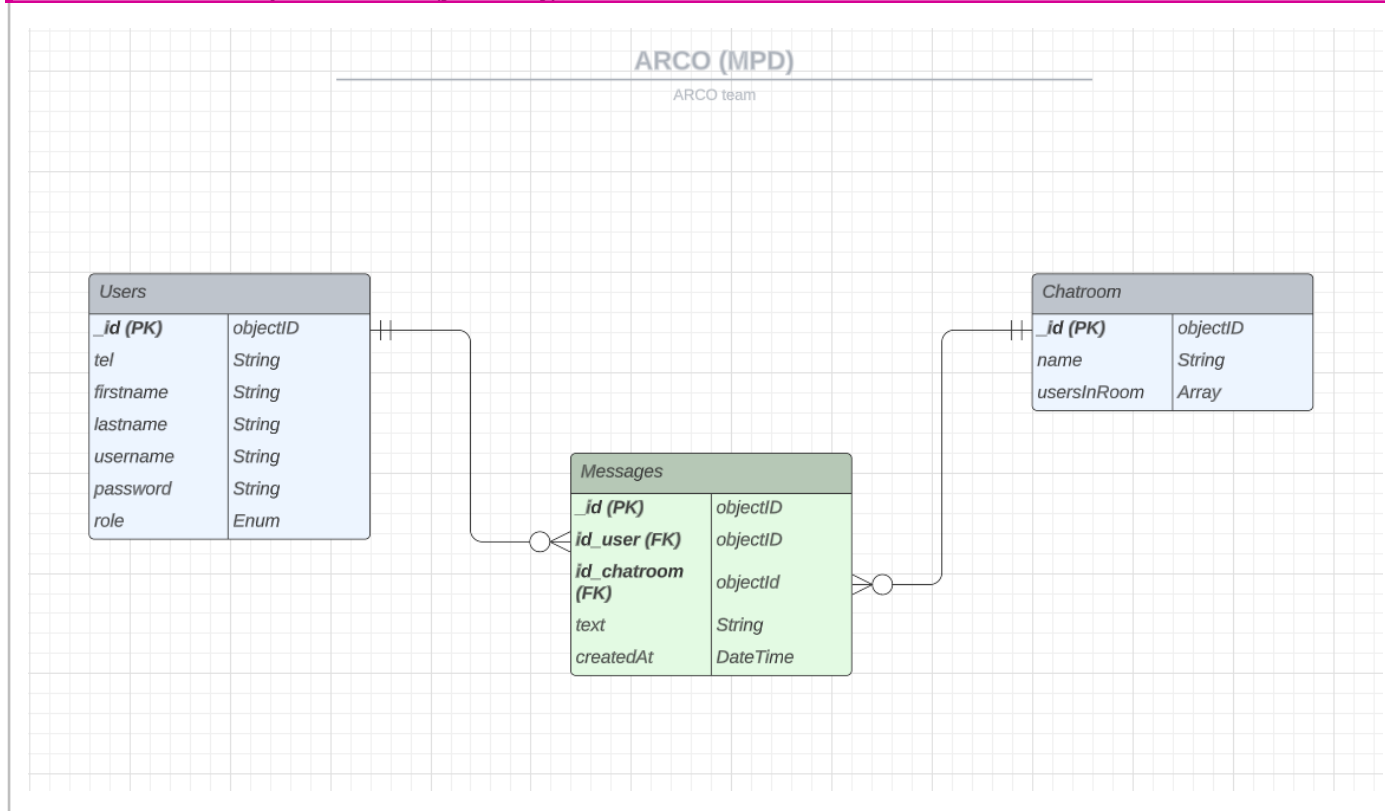
## 4. Contexte

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 : Cliquez ici au : Cliquez ici

## 5. Informations complémentaires (facultatif)



## Activité-type 3

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

Exemple n° 1 - Projet desktop Python "Sokoban"

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

#### 2. Précisez les moyens utilisés :

#### 3. Avec qui avez-vous travaillé ?

Seul.

#### 4. Contexte

Nom de l'entreprise, organisme ou association ▶

Chantier, atelier, service ▶

Période d'exercice ▶ Du : 17/04/23      au : 22/04/23

#### 5. Informations complémentaires (facultatif)

## Activité-type 3

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

Exemple n° 1 ▶ Projet application mobile "ARCO"

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

Dans le cadre de notre projet de réalisation d'une application mobile, nous avons dû concevoir une base de données

#### 2. Précisez les moyens utilisés :

#### 3. Avec qui avez-vous travaillé ?

#### 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 : Cliquez ici au : Cliquez ici

#### 5. Informations complémentaires (facultatif)

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

### 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] [Cliquez ici pour taper du texte.](#) ,  
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 [Cliquez ici pour taper du texte.](#) le [Cliquez ici pour choisir une date](#)  
pour faire valoir ce que de droit.

Signature :

### Documents illustrant la pratique professionnelle

*(facultatif)*

Intitulé
Cliquez ici pour taper du texte.



---

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

---

### ANNEXES

*(Si le RC le prévoit)*