



DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ MONDERER.
Nom d'usage ▶ MONDERER.
Prénom ▶ NAOMI
Adresse ▶ 2 rue Saint Dominique, 13001, Marseille

Titre professionnel visé

Concepteur développeur d'application web et mobile

MODALITÉ D'ACCÈS :

- 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.

DOSSIER PROFESSIONNEL (DP)

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	p.	5
▸ Intitulé de l'exemple n° 1	p.	p.
▸ Intitulé de l'exemple n° 2	p.	p.
▸ Intitulé de l'exemple n° 3	p	p.
Intitulé de l'activité-type n° 2	p.	
▸ Intitulé de l'exemple n° 1	p.	p.
▸ Intitulé de l'exemple n° 2	p.	p.
▸ Intitulé de l'exemple n° 3	p	p.
Intitulé de l'activité-type n° 3	p.	
▸ Intitulé de l'exemple n° 1	p.	p.
▸ Intitulé de l'exemple n° 2	p.	p.
▸ Intitulé de l'exemple n° 3	p	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

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

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

Exemple n°1 - AL-BEYT - maquetter une application

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

Cet exemple me permet de valider la compétence :

- Maquetter une application

Dans le cadre d'un projet en collaboration avec l'association al-beyt , nous avons créé une application web.

Nous avons réalisé dans un premier temps des maquettes basse fidélité puis haute fidélité sur **Figma**. Nous avons également utilisé des outils de prototypage qui ont permis à l'association de se projeter plus facilement en termes de futures fonctionnalités.

wireframe



DOSSIER PROFESSIONNEL (DP)

Haute fidélité

The screenshots illustrate the following pages:

- accueil.php**: Home page featuring a large banner with Arabic text and a portrait, followed by a section for the next event.
- eventsPass.php**: Page listing events, including "CHICHAS DE LA PENSEE" and "AL-BEYT X CONSULTANT".
- actus.php**: News page showing several news items with small thumbnail images.
- artistes.php**: Page displaying a grid of artist profiles, each with a thumbnail and name.
- event.php**: Event detail page for "al-beyt" with a large image, title, and description.
- actu.php**: Article detail page with a large image, title, subtitle, and detailed text content.
- artiste.php**: Artist detail page for "Korn Artiste" with a large image and multiple banner slots for "CHICHAS DE LA PENSEE".
- presentation.php**: General presentation page with sections for "Qu'est ce que le collectif de-dyst?", "A propos de notre démarcation", and "Qui sommes-nous?".
- accueil.php**: Another view of the home page.

2. Précisez les moyens utilisés :

Pour réaliser ces maquettes nous avons utilisé Figma,

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai collaboré avec Daouda Sarr durant ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **collectif al-beyt**

Chantier, atelier, service ▶ application web

Période d'exercice ▶ Du : **01/05/2022** au : **20/07/2022**

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

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

Exemple n° 2 - Le jeu du Snake - Développer une interface utilisateur de type desktop

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

Cet Exemple couvre le domaine de compétence suivant :

- Développer une interface utilisateur de type desktop

Dans le cadre de ma formation, j'ai développé le jeu du snake en Python.

Pour cela, j'ai utilisé la librairie Pygames, qui me permet de créer des éléments graphiques ainsi que d'écouter sur des événements bien précis comme sur celui des touches du clavier.

Ce projet a été codé en Orienté Objet pour pouvoir au mieux faire interagir mes éléments et une plus grande lisibilité du code.

J'ai commencé par initier le jeu avec la fonction init de Pygame et définir la surface sur laquelle on va jouer. J'ai rajouté une fonction draw_grid pour dessiner le fond de mon interface afin d'être sur de la position de mon snake et de sa nourriture.

```
● ● ●  
pygame.init()  
  
nbr_col = 10  
nbr_row = 15  
cell_size = 50  
  
screen = pygame.display.set_mode(size=(nbr_col * cell_size, nbr_row * cell_size))  
  
# Fonction pour tracer la grille de mon interface  
def draw_grid():  
    for i in range(0,nbr_col):  
        for j in range(0, nbr_row):  
            rect = pygame.Rect(i * cell_size, j * cell_size, cell_size, cell_size)  
            pygame.draw.rect(screen, pygame.Color('black'), rect, 1)
```

Cette dernière fonction est ensuite appelée dans ma game loop:

DOSSIER PROFESSIONNEL (DP)

```
● ● ●
```

```
while game_on:
    # Event pour quitter le jeu en cliquant sur la croix
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_on = False
            sys.exit()
    # Mettre à jour la position du snake et de la nourriture
    if event.type == screen_update:
        game.update()

    # Event pour les flèches du clavier
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_LEFT:
            if game.snake.direction != 'RIGHT':
                game.snake.direction = 'LEFT'
        if event.key == pygame.K_RIGHT:
            if game.snake.direction != 'LEFT':
                game.snake.direction = 'RIGHT'
        if event.key == pygame.K_UP:
            if game.snake.direction != 'DOWN':
                game.snake.direction = 'TOP'
        if event.key == pygame.K_DOWN:
            if game.snake.direction != 'TOP':
                game.snake.direction = 'DOWN'

    # Dessiner la grille
    draw_grid()
```

J'ai également créé une classe pour retourner la position de mon bloc et qui possède comme attributs des coordonnées x et y

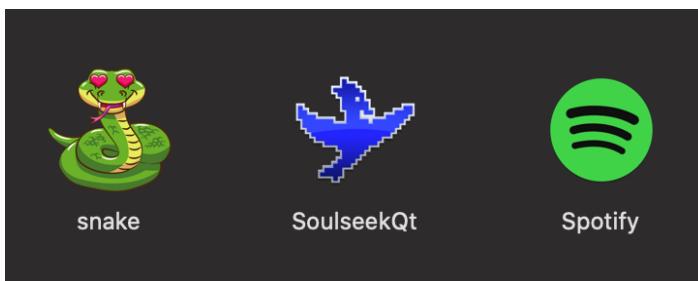
```
class Block:
    def __init__(self, x_pos, y_pos):
        self.x = x_pos
        self.y = y_pos
```

Je construis également le bloc de nourriture dans une **classe**. Dans le **constructeur**, j'applique aléatoirement une position en utilisant la fonction randint de la librairie **random**. Puis je stocke la position en utilisant une instance de la classe Block. Je crée également une fonction qui me permettra de dessiner la nourriture à la position précédente, retourner et appliquer à l'instance self.block.

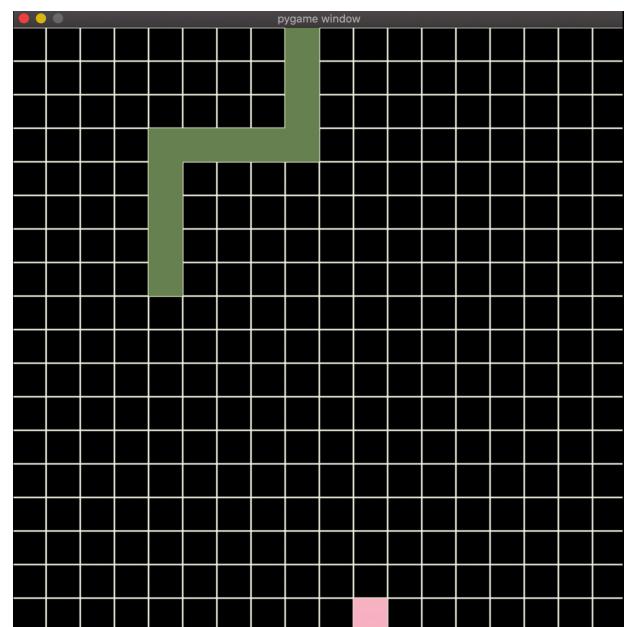
DOSSIER PROFESSIONNEL (DP)

```
class Food:  
    def __init__(self):  
        x = random.randint(0, nbr_col - 1)  
        y = random.randint(0, nbr_row - 1)  
        self.block = Block(x, y)  
  
    def draw_food(self):  
        rect = pygame.Rect(self.block.x * cell_size, self.block.y * cell_size, cell_size, cell_size)  
        pygame.draw.rect(screen, pygame.Color('green'), rect)
```

Pour finir, j'ai créé un exécutable en un seul fichier et ne plus passer par la ligne de commande pour lancer mon script, j'ai utilisé la librairie **PyInstaller** et utilisé la commande (le dernier paramètre me sert à lui assigner une icon) : **pyinstaller snake.py --onefile**



--windowed --icon snake-icon.icns



2. Précisez les moyens utilisés :

J'ai utilisé :

- VScode
- Python comme langage
- Pip comme gestionnaire de package
- les librairies : pygame , random sys, pyinstaller

3. Avec qui avez-vous travaillé ?

J'ai fait ce projet seule

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ➤ **laPlateforme**

Chantier, atelier, service ➤ **snake**

Période d'exercice ➤ Du : **10/03/2023** au : **14/03/2023**

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

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

Exemple n° 3 - Chuu chat

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

La compétence validée par cette activité est la suivante:

- Développer des composants d'accès aux données.

Nous avons installé le module via la commande “npm install mysql2”, puis nous avons initialisé le module dans le point d’entrée de notre api via la méthode require(). Pour se connecter à la base de données nous avons utilisé la méthode createConnection() permettant de passer les paramètres host, password, database et port afin de rentrer les valeurs correspondantes et d’établir la connexion à la base de données.

```
const mysql = require('mysql');
var connection
connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'root',
  database: 'chat',
  port: '3306',
});
connection.connect(function(err) {
  if (err) {
    console.error('Error connecting to database:', err);
    return;
  }
  console.log(`Connected to database on port ${connection.config.port}`);
});
module.exports = connection
```

capture d'écran du fichier back/database.js

Nous avons ensuite récupéré sur le fichier adminController.js du dossier Controller notre fichier database.js et nous l'avons stocké dans la constante “db” qui sera l'objet grâce auquel nous pourrons exécuter toutes nos queries via la fonction db.query() et de ce fait interagir avec la base de données.

DOSSIER PROFESSIONNEL (DP)

Dans un premier temps nous passons en paramètre de notre fonction fléchée les paramètres attendus par le framework Express (req, res), afin de récupérer des éléments de la requête http venant du front et la réponse qui lui sera renvoyée depuis cette fonction.

```
// permet à l'admin de modifier le rôle et le login d'un user.
const adminUpdateRole = (req, res) => {
    const id_role = parseInt(req.body.id_role)
    const userId = parseInt(req.params.id)
    const sql = 'UPDATE users SET id_role = ? WHERE id = ?'
    db.query(sql, [id_role, userId] , function(error, data){
        if(error) throw error;
        else {
            if (id_role === 0) {
                const sql = `DELETE FROM participants WHERE id_user = ${userId}`
                db.query(sql, function(error){
                    if(error) throw error;
                    else res.send('Deleted from chat-rooms.').status(204);
                })
            }
            else res.send(data).status(204);
        }
    })
}
```

capture d'écran du fichier back/Controller/adminController.js

On voit dans l'extrait de code ci-dessus que l'id_role et l'id des utilisateurs sont envoyés en paramètre de la fonction query() et que la réponse renvoie via la fonction send() un message et la fonction status() le statut code http de cette dernière.

2. Précisez les moyens utilisés :

- ✓ VScode
- ✓ npm comme gestionnaires de package
- ✓ ExpressJs comme framework
- ✓ NodeJs comme environnement
- ✓ SQL comme langage pour interagir avec la base de données

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Laura SAVICKAITE Laura Scognamiglio Dorian Palace

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 (DP)

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

Exemple n° 4 - e-boutique

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

Ce projet valide les compétences :

- **Développer la partie back-end d'une interface utilisateur web**
- **Développer la partie front-end d'une interface utilisateur web**

Il a été réalisé avec les langages suivants : HTML / CSS / PHP/ SQL. Ce projet se rapproche d'une architecture MVC (model, view, controller) bien que nous ayons pris certaines libertés dans l'implémentation de celle-ci. Le code se découpe donc en 3 dossiers qui sont eux-même composés par des classes. Chaque classe du dossier Model représente une entité de la base de données et contient les requêtes associées. Nous avons respecté les conventions de nommage des classes.

Je me suis occupée de l'interface d'administration (back-office) par le développement d'un CRUD (create, read, update, delete) pour chacune des entités devant permettre à l'administrateur la gestion de son site:

-Gestion des produits

-Gestion des catégories et des sous-catégories

-Gestion des auteurs

-Gestion des utilisateurs

J'ai également implémenté l'interface de l'inscription et de connexion des utilisateurs. La connexion à la base de données se fait à l'aide de PDO via l'utilisation de requêtes préparées. Nous avons aussi utilisé certaines fonctions natives à PHP (telles que htmlspecialchars ou trim). Ces outils nous ont permis d'assurer la sécurité du site contre certaines failles (XSS ou les injections SQL).

Pour ce projet, je devais aussi concevoir la partie front-end du back-Office afin de pouvoir en faire une bonne gestion depuis l'interface. J'ai donc conçu des maquettes et séparé les pages de gestion en fonction des actions possibles.

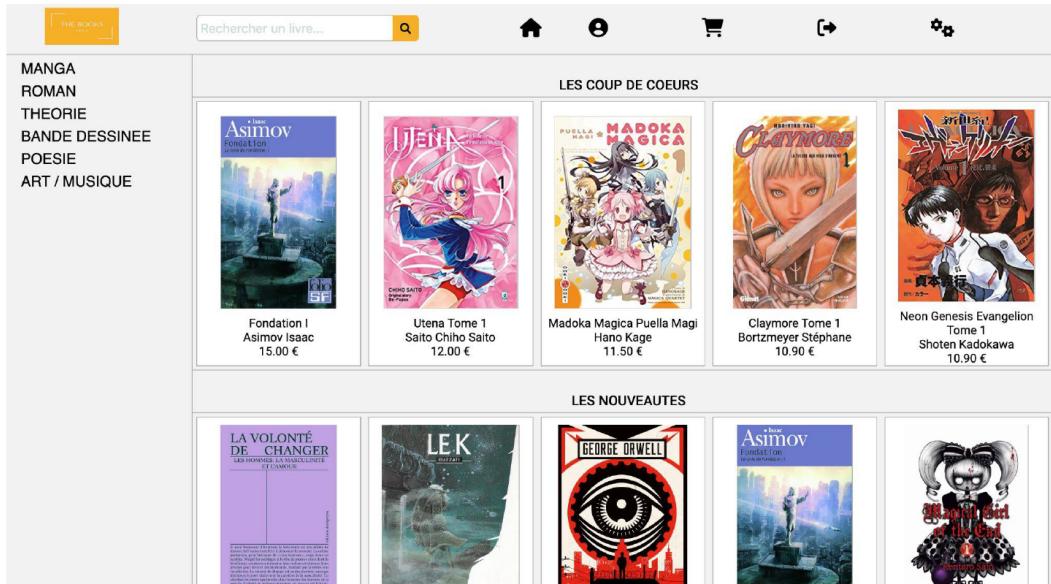
5 pages permettent ces actions:

- GererUtilisateur.php
- AjouterLivre.php
- GererLivres.php

DOSSIER PROFESSIONNEL (DP)

- AjouterCategories.php
- GererCategories.php

l'interface se compose de deux espaces : d'un côté le menu de gestion et de l'autre l'affichage du contenu à gérer ou des formulaires à remplir en fonction de la query à exécuter.



Capture d'écran de la page d'accueil côté utilisateur

The back office interface has a yellow sidebar on the left with links: BACK OFFICE, Gérer les utilisateurs, Ajouter un livre, Gérer les livres, Ajouter des Options, and Gérer les options. The main area shows two forms. The first form, 'NOUVEAU GENRE', has a 'Genre:' input field and a 'AJOUTER UN GENRE' button. The second form, 'NOUVEAU SOUS-GENRE', has a 'Genre:' dropdown set to 'MANGA', a 'Sous-Genre:' input field, and an 'AJOUTER UN SOUS-GENRE' button.

Capture d'écran de la page de AjouterCategorie.php

DOSSIER PROFESSIONNEL (DP)

The screenshot shows a web application interface for adding a new book. The left sidebar is titled 'BACK OFFICE' and contains links for managing users, adding books, and other options. The main area has a title 'ENREGISTRER UN NOUVEL OUVRAGE'. The form fields include:

- Nom de l'ouvrage: CyberStructure
- Auteur.ice: Bortzmeyer Stéphane
- Description: A large text area containing a paragraph about digital life.
- Mettre en avant cet article: non
- Editeur: Cyber
- Catégorie: THEORIE
- Sous-Catégorie: Technologie
- Nombre d'articles à ajouter au stock: 90
- Prix: 17.00
- Choisir une image: cyberstructure-l900.jpg
- AJOUTER UN LIVRE button

Capture d'écran de la page de AjouterLivre.php

The screenshot shows a 'Gestion des Categories' section with a table:

ID	CATÉGORIES	MODIFIER	SUPPRIMER
3	MANGA	X	X
4	ROMAN	X	X
5	THEORIE	X	X
6	BANDE DESSINEE	X	X
8	POESIE	X	X
9	ART / MUSIQUE	X	X

The screenshot also shows a 'Gestion des Sous-Catégories' section with a table:

ID	CATÉGORIES	SOUS-CATÉGORIES	MODIFIER	SUPPRIMER
3	MANGA	Mecha	X	X
3	MANGA	Dark Fantasy	X	X
3	MANGA	Magical Girl	X	X

Capture d'écran de la page de gestionCategories.php

2. Précisez les moyens utilisés :

- ✓ Visual Studio Code pour la création et l'édition de code.
- ✓ GitHub pour le travail en équipe et le versionning.
- ✓ SQL comme langage pour communiquer avec la base de données.
- ✓ Mysql comme gestionnaire de base de données.

DOSSIER PROFESSIONNEL (DP)

- ✓ HTML / CSS
- ✓ JAVASCRIPT pour la validation des formulaires
- ✓ PHP pour le traitement des données des formulaires
- ✓ Fontawesome pour l'importation d'icônes gratuites
- ✓ Google font pour le choix des polices

3. Avec qui avez-vous travaillé ?

Jerome Lacroix et Sébastien Bargier

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 (DP)

Activité-type 2 Concevoir et développer la persistance des données en intégrant les recommandations de sécurité.

Exemple n° 1 - Chu chat

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

Pour ce projet, nous avons créé une base de données pour répondre à la demande du client.

Ce projet valide les compétences :

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

Pour ce projet nous avons développé une API, qui nous permet d'établir une couche contenant la logique métier du projet et de faire la jointure entre l'accès aux données afin de fournir une interface entre l'application et la source de données.

Sur cette base de données, on va pouvoir manipuler les différentes informations liées :

- à l'utilisateur
- aux chatroom
- aux messages

On a utilisé node Js ainsi que la librairie MySQL2. En effet nous avons utilisé SQL comme langage pour interagir avec notre base de données et MySQL comme gestionnaire de base de données relationnelles.

Nous nous sommes appuyés sur la méthode “**Merise**” pour concevoir notre base de données SQL. On a commencé par réaliser sur l'outil **figjam.io**, le **MCD** (modèle conceptuel de données) et enfin le **MLD** (modèle logique de données) en annexe 1 et 2.

Nous avons mis en place notre base de données via le software phpMyAdmin en créant la base de données et les tables relationnelles, puis la table de liaison depuis l'interface du software. Nous avons imposé des contraintes à certains champs de nos tables en créant des contraintes grâce à des requêtes de delete en cascade.

Nous avons installé le module via la commande “`npm install mysql2`”, puis nous avons initialisé le module dans le point d'entrée de notre api via la méthode `require()`. Pour se connecter à la base de

DOSSIER PROFESSIONNEL (DP)

données nous avons utilisé la méthode `createConnection()` permettant de passer les paramètres `host`, `password`, `database` et `port` afin de rentrer les valeurs correspondantes et d'établir la connexion à la base de données.

```
const mysql = require('mysql');
var connection
connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'root',
  database: 'chat',
  port: '3306',
});
connection.connect(function(err) {
  if (err) {
    console.error('Error connecting to database:', err);
    return;
  }
  console.log(`Connected to database on port ${connection.config.port}!`);
});
module.exports = connection
```

capture d'écran du fichier back/database.js

Nous avons ensuite récupéré sur le fichier `adminController.js` du dossier `Controller` notre fichier `database.js` et nous l'avons stocké dans la constante “`db`” qui sera l'objet grâce auquel nous pourrons exécuter toutes nos queries via la fonction `db.query()` et de ce fait interagir avec la base de données.

Dans un premier temps nous passons en paramètre de notre fonction fléchée les paramètres attendus par le framework Express (`req, res`), afin de récupérer des éléments de la requête http venant du front et la réponse qui lui sera renvoyée depuis cette fonction.

DOSSIER PROFESSIONNEL (DP)

```
// permet à l'admin de modifier le rôle et le login d'un user.
const adminUpdateRole = (req, res) => {
    const id_role = parseInt(req.body.id_role)
    const userId = parseInt(req.params.id)
    const sql = 'UPDATE users SET id_role = ? WHERE id = ?'
    db.query(sql, [id_role, userId] , function(error, data){
        if(error) throw error;
        else {
            if (id_role === 0) {
                const sql = `DELETE FROM participants WHERE id_user = ${userId}`
                db.query(sql, function(error){
                    if(error) throw error;
                    else res.send('Deleted from chat-rooms.').status(204);
                })
            }
            else res.send(data).status(204);
        }
    })
}
```

capture d'écran du fichier back/Controller/adminController.js

On voit dans l'extrait de code ci-dessus que l'id_role et l'id des utilisateurs sont envoyés en paramètre de la fonction query() et que la réponse renvoie via la fonction send() un message et la fonction status() le statut code http de cette dernière.

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 (DP)

Activité-type **2** **Concevoir et développer la persistance des données en intégrant les recommandations de sécurité.**

Exemple n° 2 - al-beyt

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 ?

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 (DP)

Activité-type

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

Exemple n° 1 - Chuu Chat

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

Les compétences validées par cette activité sont les suivantes :

- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement: trello méthodologie branch git technologie agile
- Concevoir une application. : maquettes, méthode merise,
- Développer des composants métier. : api les routes et leurs fonctionnalités
- Construire une application organisée en couches.: architecture multi-couche (schéma) découpé en 3 - couche interface utilisateur - couche de logique métier - couche persistance des données.
- Développer une application mobile. (navigation, composant front,
- Préparer et exécuter les plans de tests d'une application.
- Préparer et exécuter le déploiement d'une application.

Dans le cadre du projet professionnel de deuxième année, nous avons développé une application de chat (sur le thème de la K-Pop) composée de 3 couches :

- une application mobile native (destinée aux utilisateurs du chat)
- une application web (destinée aux administrateurs du chat)
- une API REST qui fait le lien entre la base de donnée du chat et les application web et mobiles

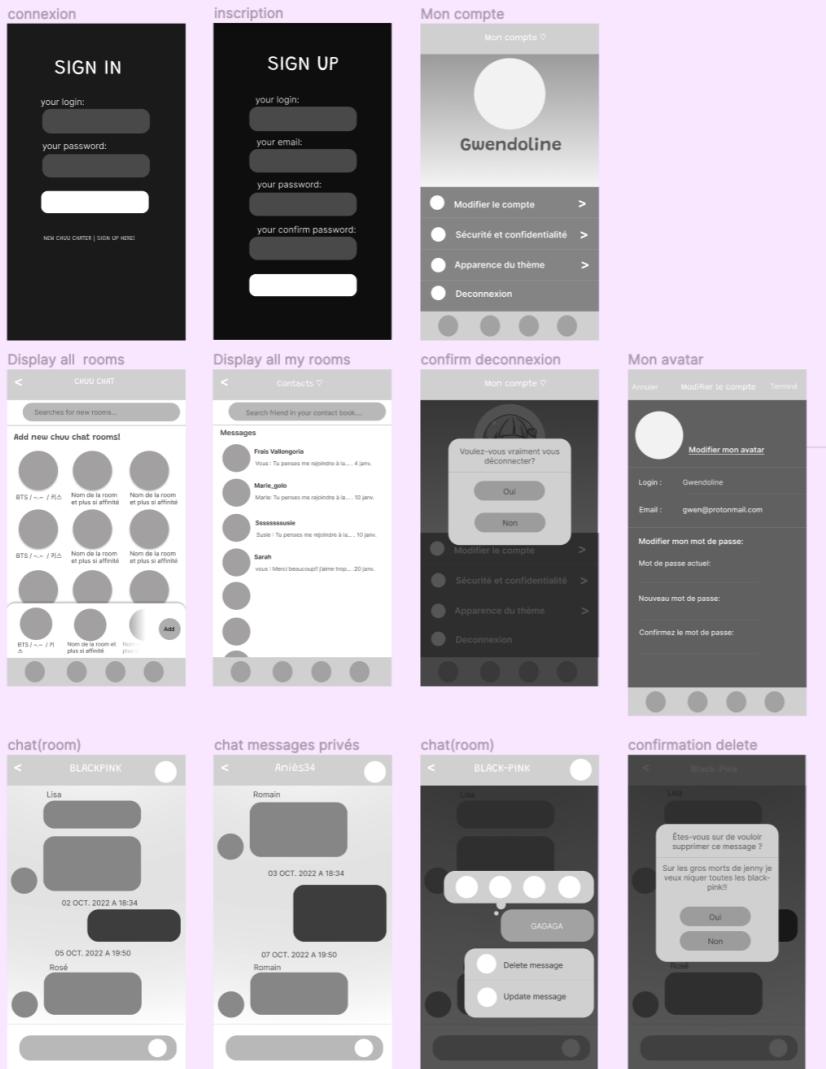
Pour ce faire, il a d'abord fallu définir le cadre du projet et de sa gestion, ainsi que l'environnement de développement commun à l'équipe de développement. Nous avons donc choisi d'appliquer une des méthodes agiles (**SCRUM**) pour ce projet (planification de sprints, code review, daily standup...) en utilisant l'outil **Trello**.

Après avoir défini la charte graphique, nous avons réalisé des maquettes sur **Figma** pour concevoir l'interface graphique des applications web et mobile. voir les maquettes ci-dessous.

DOSSIER PROFESSIONNEL (DP)

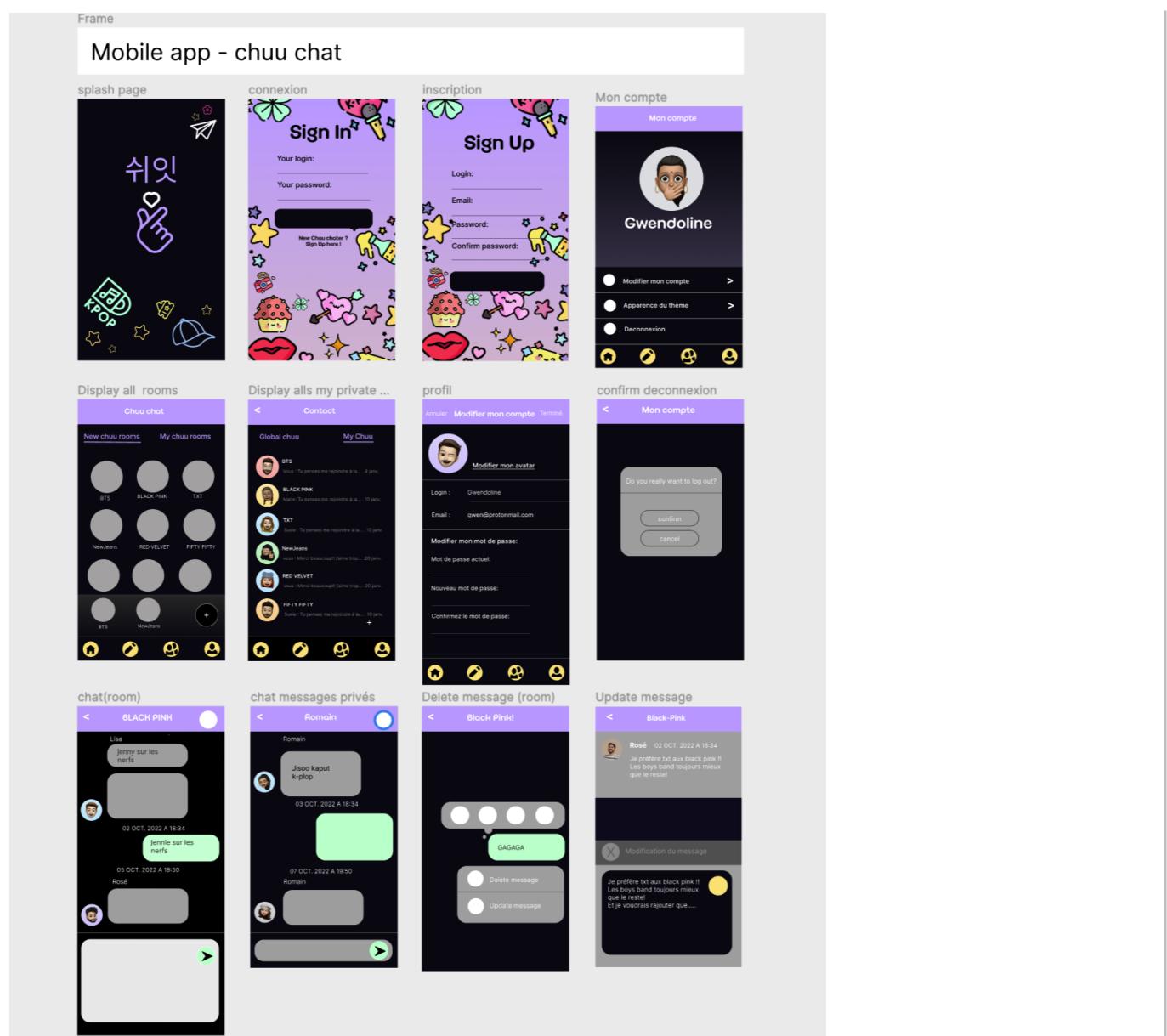
TITRE

Mobile app - Chuu chat



Maquette app mobile Basse fidélité

DOSSIER PROFESSIONNEL (DP)



Maquette app mobile Haute fidélité

Après cela, il a également fallu concevoir le schéma de données de la BDD en utilisant la méthode **MERISE** en réalisant les plans de conception qui sont présentés en annexe 1 et 2 .

En parallèle nous avons choisi les technologies pour développer ce projet : **React Native** et **Expo** pour l'application mobile, **Javascript** pour l'application web, ainsi que **Node.js** et **Express.js** pour l'API. Avant de commencer le développement, nous avons créé plusieurs repositories **GitHub** en mettant en place des règles afin de pouvoir travailler collaborativement en suivant la méthode SCRUM. (scope et nommage des branches).

Les composants métiers de l'application sont développés dans l'API. Chaque route correspond à une fonctionnalité de l'utilisateur ou de l'administrateur.

DOSSIER PROFESSIONNEL (DP)

Les routes sont donc regroupées via le routeur d'Express (utilisateurs, salons, messages, participations, administration). Ces dernières appellent chacune une fonction qui est gérée par le controller correspondant. Cette fonction comprend la logique métier, les queries SQL à exécuter et le response de l'api vers le client.

L'application est organisée en multi-couche et fonctionne grâce à l'architecture client-serveur.

Les 3 couches applicatives sont:

- **La couche client** matérialisée par l'application mobile et web
- **La couche applicative** (couche de logique métier) matérialisée par l'API
- **La couche de persistance des données** matérialisée par la base de données.

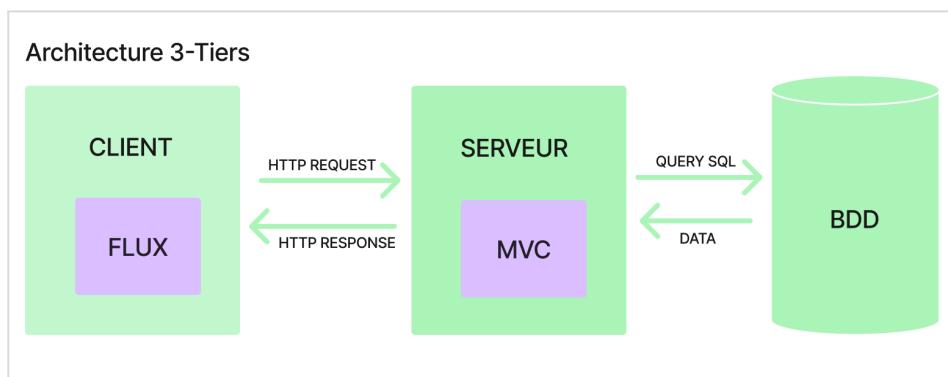


Schéma des architectures du projet

Notre application mobile est un chat en ligne qui a pour but de rassembler les fan de musique k-pop en une grande communauté. Comme précisé plus haut nous avons utilisé **react Native** pour la partie mobile et les fonctionnalités de l'api ainsi que la **librairie d'expo** qui nous a permis de simuler notre **environnement de développement** en temps réel. Pour se faire nous avons organisé notre projet front en dossiers.

Voici une énumération des dossiers les plus importants du front:

- assets: on y trouve toutes les images au format png
- composant: on y trouve les sous composants qui seront appelés dans le dossier screen afin d'y être emboîté ensemble et de former la totalité de ce qui sera rendu via l'interface utilisateur.
- navigation : on y trouve les fichiers principaux responsables de la navigation sur le front. Nous avons choisi d'utiliser react navigation et avons mis en place un système de navigation imbriquée entre la tab bar et les différentes vues de l'application.
- screen : on y trouve les composants qui sont rassemblés pour former les vues.

Cette application regroupe différentes fonctionnalités :

DOSSIER PROFESSIONNEL (DP)

- Inscription /connexion
- S'ajouter à une room
- Poster un message dans une room
- Modifier son profil

Nous avons mis en place la génération d'un **token unique** lors de l'authentification des utilisateurs sur le portail de connexion via la librairie logicielle du **json web token** qui nous permet de sécuriser les requêtes depuis le côté client vers le serveur. Ce token sera vérifié à chaque action demandée par les **requêtes http**. Il est stocké dans le **keyStore** ou **keyChain** de l'appareil utilisé grâce au service de **secureStore**. Il s'agit d'un service de l'api d'expo qui permet de **récupérer et stocker le token** de manière sécurisé via un algorithme de **chiffrement/déchiffrement** du token en fonction des besoins du projet.

Sécurité du JWT:

- 1.**Le token permet de valider des requêtes sans utiliser de mot de passe pour vérifier l'utilisateur** en faisant circuler celui-ci de nombreuses fois entre le client et le serveur, ce qui exposerait l'application et la base de données à des failles.
- 2.Dans le cas d'un vol du jeton, **la date d'expiration du token** permet d'éviter un vol de ce jeton sur le long terme
3. **La signature du JWT** garantit l'intégrité des données. Si le contenu du JWT est modifié après avoir été émis, la signature devient invalide et le token est considéré comme non fiable.
4. **Authentification de l'utilisateur grâce au payload:** ces informations peuvent être vérifiées par le serveur à chaque demande pour s'assurer que l'utilisateur est authentifié et a les autorisations nécessaires pour accéder à certaines ressources.

Pour faire fonctionner le chat nous avons eu recours à la technologie des web socket en utilisant la librairie socket.io côté serveur et socket.io-client. Nous avons donc implémenter les sockets côté serveur :

DOSSIER PROFESSIONNEL (DP)

```
io.on('connection', (socket) => {
    console.log('a user connected');
    socket.on('disconnect', () => {
        console.log('user disconnected');
    });
    socket.on('joinIn', (id_room) => {
        socket.join(id_room);
    })
});

server.listen(port, () => {
    console.log(`Socket.IO server running on port :${port}`);
});
```

Pour que le serveur connaisse les id des rooms, on appose à socket un “événement” nommé “joinIn” - dans l’exemple au-dessus, il est écouté via le “.on”.

Cet événement est émis du côté client dans notre composant ChatRoom qui détient l’ensemble des id des rooms. C’est également dans ce composant qu’on appelle celui des Messages à qui on fait passer la connexion socket en props.

Puis l’implémentation des socket côté client:

```
const [socket, setSocket] = useState(io("http://localhost:3000"));

useEffect(() => {
    socket.emit('joinIn', route.params.id_room)
```

On instancie l’objet “io” et on lui passe par défaut l’adresse url du serveur. On stocke cet objet dans la constante “socket”. On accroche l’événement .emit à ce nouvel objet qui permet de joindre l’id de la room dans laquelle l’utilisateur se trouve au serveur afin d’insérer le message en base de données dans la bonne room.

2. Précisez les moyens utilisés :

- **Figma** pour la réalisation des maquettes
- **Trello** pour la gestion de projet
- **GitHub** pour le versionning collaboratif
- **React Native** pour l’application native
- **Expo** pour l’environnement de développement et le déploiement d’application native
- **Node.js et Express.js** pour le développement de l’API
- **Javascript** pour les calls API de l’application web
- **PHP** pour les pages de l’application web

DOSSIER PROFESSIONNEL (DP)

!

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*)

Les compétences validées par cette activité sont les suivantes :

- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement.
- Concevoir une application.
- Développer des composants métier.
- Construire une application organisée en couches.
- Développer une application mobile.
- Préparer et exécuter les plans de tests d'une application.
- Préparer et exécuter le déploiement d'une application.

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.

DOSSIER PROFESSIONNEL (DP)

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 :

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)

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

- OK_(1-al-beyt) Maquetter une application.
- OK_(2-snake) Développer une interface utilisateur de type desktop
- OK(3-chuu-chat) Développer des composants d'accès aux données.
- OK (boutique en ligne) Développer la partie front-end d'une interface utilisateur web.x boutique
- OK (boutique en ligne) Développer la partie back-end d'une interface utilisateur web. x boutique

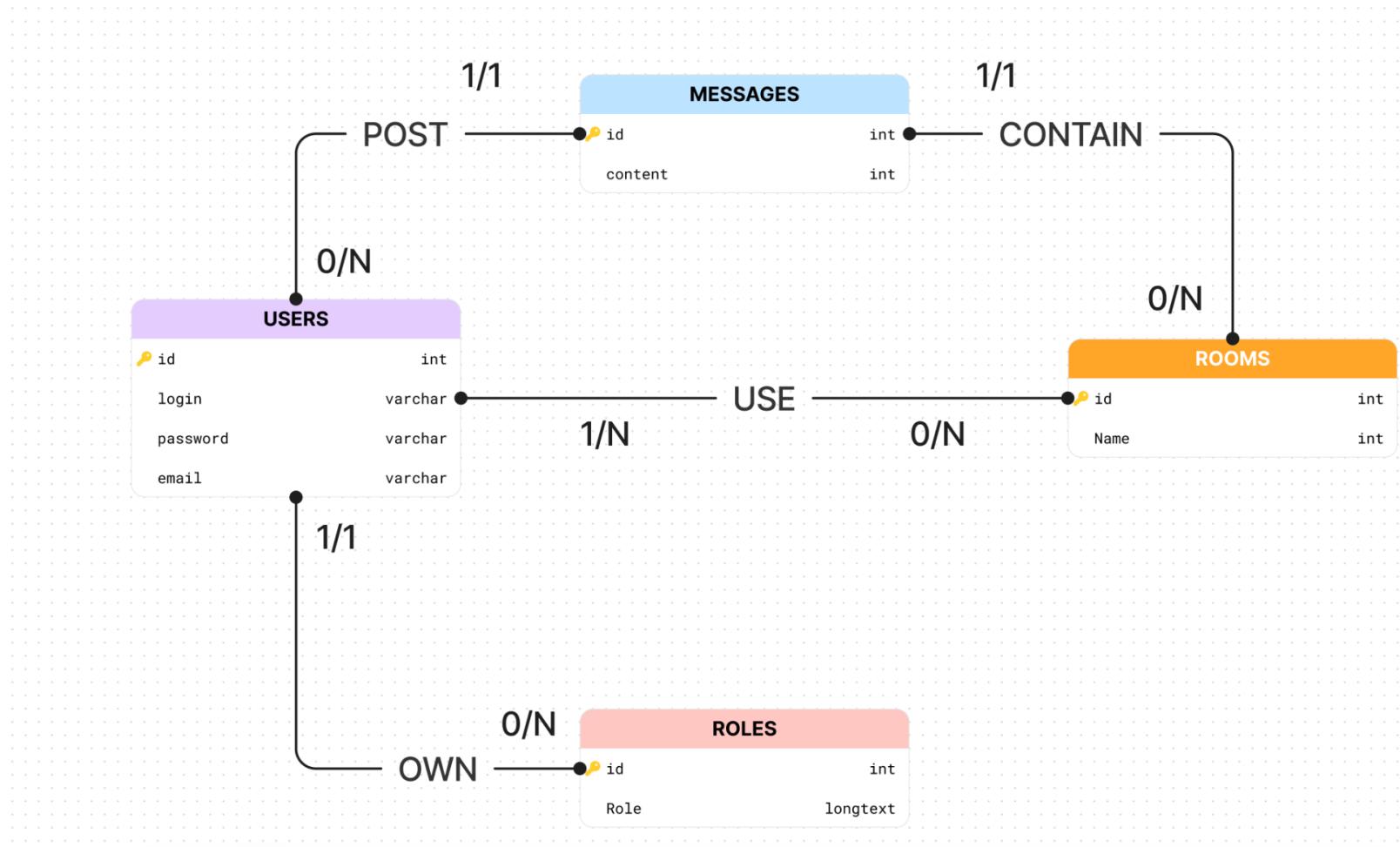
2-Concevoir et développer la persistance des données en intégrant les recommandations de sécurité.

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

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

- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement.
- Concevoir une application.
- Développer des composants métier.
- Construire une application organisée en couches.
- Développer une application mobile.
- Préparer et exécuter les plans de tests d'une application.
- Préparer et exécuter le déploiement d'une application.

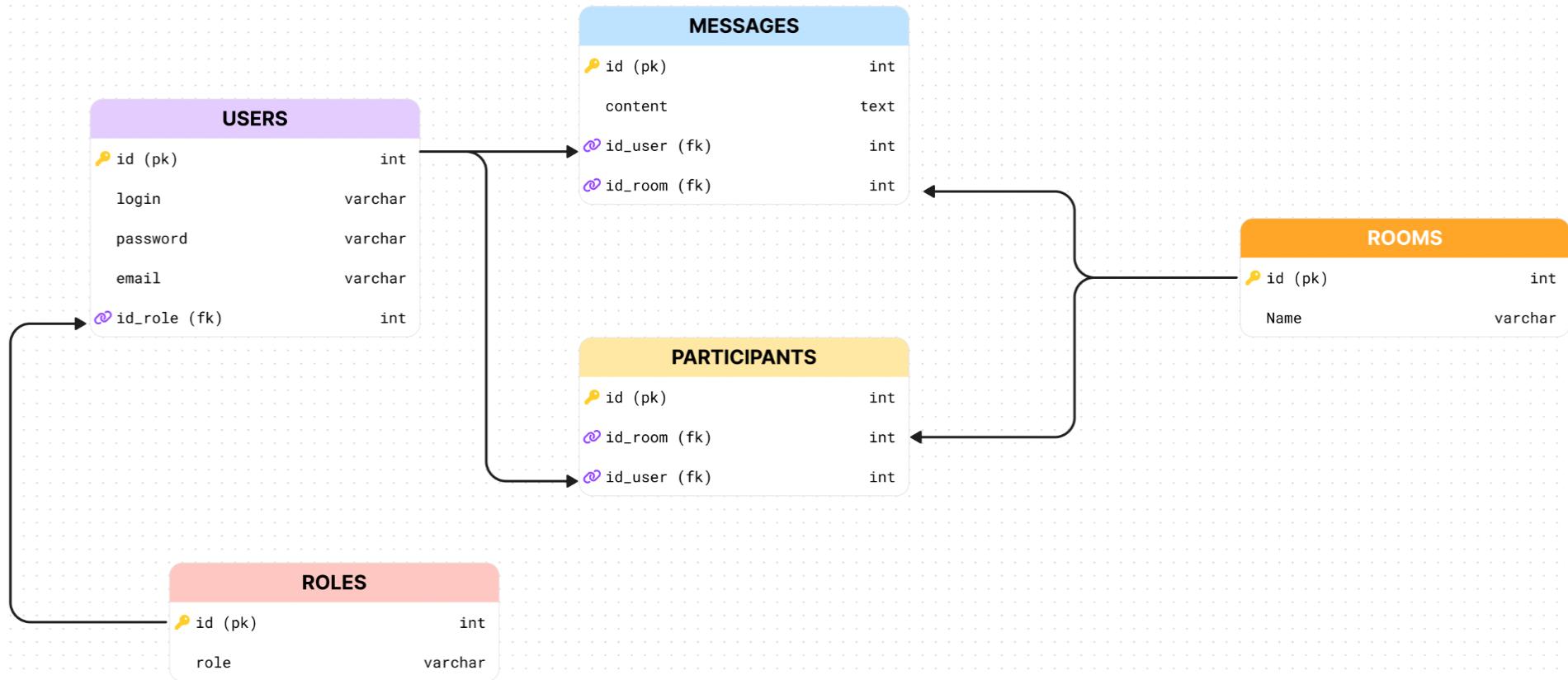
DOSSIER PROFESSIONNEL (DP)



Plan du modèle conceptuel de données (MCD)

DOSSIER PROFESSIONNEL-Version du 11/09/2017

DOSSIER PROFESSIONNEL (DP)



Plan du modèle logique de données (MLD)

DOSSIER PROFESSIONNEL ^(DP)
