



DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ MELONO

Prénom ▶ BORIS

Adresse ▶ 136, BOULEVARD NATIONAL
13003 MARSEILLE

Titre professionnel visé

Concepteur développeur d'applications

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 :

DOSSIER PROFESSIONNEL ^(DP)

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

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



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

Sommaire

Exemples de pratique professionnelle

Développer des composants d'interface	p.	5
- Développement d'une application todo app web en symfony	p.	5
▶ Développement d'un jeu SOKOBAN en python	p.	8
Développer la persistance des données	p.	11
- Développement d'une API (API PLATFORM)	p.	11
Développer une application multicouche	p.	15
- Développement d'une application todolist	p.	15

DOSSIER PROFESSIONNEL ^(DP)

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

p. 19

Déclaration sur l'honneur

p. 20

Documents illustrant la pratique professionnelle *(facultatif)*

p. 21

Annexes *(Si le RC le prévoit)*

p. 22

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1 Développer des composants d'interface

Exemple n°1 - Développement d'une application todo app en symfony

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

Lors de ma formation concepteur développeur web/mobile à La plateforme qui est une école du web sur Marseille, j'ai eu l'occasion de développer à ma propre initiative une todo App en Symfony.

Cette application a pour but d'ajouter des tâches à faire, d'en créer, ainsi que d'en supprimer en décrivant chaque activité à faire en pouvant ajouter ou supprimer chaque donnée entrée.

J'ai défini les différents besoins pour que mon application fonctionne notamment sur chaque ajout, suppression ou création de choses à faire, mon application j'ai choisi de la faire en anglais car ça me permet de travailler et d'améliorer mon anglais donc c'est un choix, au niveau du fonctionnement de l'application nous avons :

Sur la page d'accueil : avec le nom de chaque activité à faire ainsi que la description avec la possibilité de pouvoir supprimer ou ajouter un élément.

L'utilisateur peut ajouter ou supprimer une tâche.

Dans chaque rubrique activité, description de l'activité, supprimer une activité, avec toute ces rubriques on a accès à une todoapp simple et compréhensible.

Ainsi mon choix s'est porté sur Symfony, qui est un Framework PHP, vu lors de ma

Formation, Symfony est un puissant Framework PHP utilisé pour développer des applications web ou des sites web complexes, voire haut de gamme. Relativement simple d'accès, Symfony est un ensemble de composants (ou « bibliothèques ») qui facilitent le développement web en réduisant de façon

DOSSIER PROFESSIONNEL ^(DP)

considérable le temps et l'effort requis pour créer des composants génériques.

Je n'ai pas utilisé de styles particuliers pour élaborer mon application, ce qui est du logo mon choix s'est porté sur une todoapp simple d'utilisation donc mon choix de ne pas mettre de logo est intentionnel, pour ce qui est de la conception et le maquetage j'ai conçu un wireframe qui montre une idée du fonctionnement global de l'application que je vous joins ci-dessous :



Voici le rendu de mon application :

Vue de la dashboard :

Todo App with Symfony

#	Activity	Description	Delete
Add a todo			

DOSSIER PROFESSIONNEL ^(DP)

Vue de la création de tâche :

Add Todo

Title

Content

Add todo

Return to menu

Vue de la dashboard avec les différentes options :

Todo App with Symfony

#	Activity	Description	Delete
10	Programming	Create a CRUD on Symfony	Delete
11	Run	Run for a while	Delete
12	Study	Study Math	Delete

Add a todo

DOSSIER PROFESSIONNEL ^(DP)

2. Précisez les moyens utilisés :

Les moyens utilisés pour ce projet :

Editeur de texte :

J'ai utilisé Visual studio code, qui est un environnement de développement intégré (IDE). Il permet la prise en de divers langages d'où Symfony qui est un Framework PHP.

Création des maquettes :

Pour réaliser les maquettes, j'ai utilisé un outil de type Paint, qui est un outil disponible sur tout ordinateur, gratuit avec la possibilité de concevoir comme on le souhaite.

Documentations :

J'ai utilisé comme documentations le site : <https://symfony.com/doc/current/setup.html>

qui s'adresse aux développeurs, chef de

Projets, directeurs techniques ou, plus largement. Mets à disposition l'apprentissage

Du Framework (Routage et contrôleur, les Templates avec Twig, les bases de données).

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet mais en amont on avait déjà eu le même type de projet mais pour un

DOSSIER PROFESSIONNEL ^(DP)

blog avec l'aide de notre formateur scolaire Remy Viscaino .

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *La plateforme*

Chantier, atelier, service ▶ Dans le cadre de la formation : Développement d'une application web

Période d'exercice ▶ Du : 20/09/2021 au : 25/09/2021

5. Informations complémentaires (facultatif)

Activité-type 1 Développer des composants d'interface

Exemple n° 1 - Développement d'un jeu SOKOBAN en python

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

Tout d'abord, Le sokoban est un petit jeu vidéo de puzzle inventé au Japon au début des années 1980. Il met en scène un gardien d'entrepôt (traduction littérale du nom du jeu) - le joueur - qui se déplace dans un entrepôt, représenté par une grille de cases, et dont le but est de ranger des caisses sur des cases cibles.

Le but de ce projet était de réaliser un jeu en python avec plusieurs niveaux de jeux, ainsi que la possibilité de sauvegarde des parties.

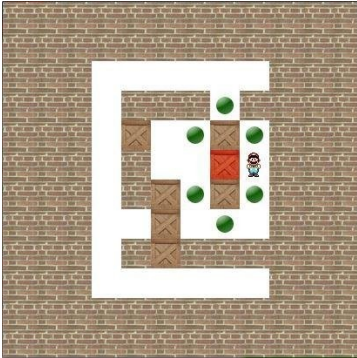
Voici une maquette liée au jeu Sokoban en python :

L'utilisateur choisit un niveau pour pouvoir jouer sur son interface de jeu.



Le joueur est redirigé, par la suite sur son niveau de jeu avec le but de pousser des caisses et de

bouger le personnage Mario, voici cela démontré ci-dessous :



Réalisation

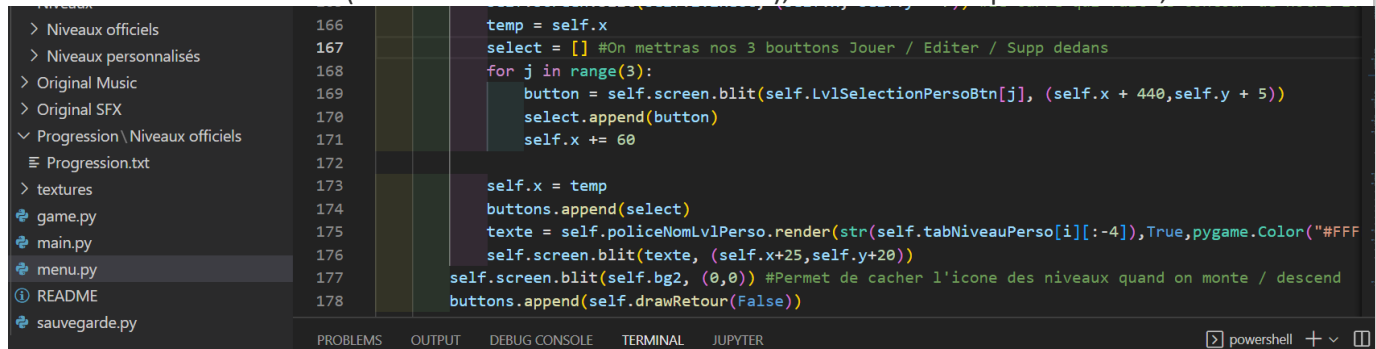
Pour l'étape de réalisation, j'ai codé avec le langage de programmation Python qui est orienté objet c'est-à-dire qu'elle permet de créer des entités (objets) que l'on peut manipuler. La programmation orientée objet impose des structures solides et claires. Les objets peuvent interagir entre eux, cela facilite grandement la compréhension du code et sa maintenance.

Je vais vous expliquer et montrer comment je récupère les données de chaque joueur par-rapport à son niveau qui lui permet lorsqu'il réussit le niveau d'avancer ou de reprendre là où le joueur s'est arrêté, ces données sont sauvegardées ce qui donne l'opportunité d'une nouvelle partie avec possibilité d'un autre niveau.

Partons d'une situation initiale, dans le cas de mon jeu permet à tous les utilisateurs de jouer, chaque utilisateur aura le droit de créer ou de choisir un niveau personnalisé ou pas et sa partie sera sauvegardée et le joueur continuera là où il s'est arrêté et de progresser, ou pourra supprimer un niveau sauvegarder dans « EDITER LEVEL »

Mes composants d'interface :

Boutons « Jouer (Levels choix de niveaux), Niveaux personnalisés, éditer »



```
166 temp = self.x
167 select = [] #On mettras nos 3 boutons Jouer / Editer / Supp dedans
168 for j in range(3):
169     button = self.screen.blit(self.LvlSelectionPersoBtn[j], (self.x + 440, self.y + 5))
170     select.append(button)
171     self.x += 60
172
173 self.x = temp
174 buttons.append(select)
175 texte = self.policeNomLvlPerso.render(str(self.tabNiveauPerso[i][-4]), True, pygame.Color("#FFF
176 self.screen.blit(texte, (self.x+25, self.y+20))
177 self.screen.blit(self.bg2, (0,0)) #Permet de cacher l'icone des niveaux quand on monte / descend
178 buttons.append(self.drawRetour(False))
```

DOSSIER PROFESSIONNEL ^(DP)

```
import pygame
2
3
4 def sauvegarde(niveau,nom): #Fonction pour sauvegarder un niveau (pour l'éditeur de ni
5     fileName = "Niveaux/"+nom+".txt"
6     i=0
7     tmp=nom
8     while os.path.isfile(fileName)==True: #Si un fichier du même nom existe déjà on change le nom po
9         i+=1
10        tmp = nom+" (" +str(i)+")"
11        fileName = "Niveaux/"+tmp+".txt"
12    if i!=0:
13        nom = nom+" (" +str(i)+")"
14    fichier = open("Niveaux/"+nom+".txt", "w") #Ensuite on crée le fichier
15    fichier.write(str(niveau)) #Et on écrit le niveau dedans
16    fichier.close()
17
18 def read_niveauperso(nom): #Fonction pour lire un niveau personnalisé
19     fileName = "Niveaux/Niveaux personnalisés/"+nom
20     if os.path.isfile(fileName)==False: #Cas où le fichier n'existe pas
21         return None
22     else:
23         f = open(fileName,'r') #Sinon on lit le contenu du fichier et on le converti en tableau (eval() co
24         message = f.read()
25         f.close()
26         return eval(message)
27
28 def read_niveauofficiel(nom): #Fonction pour lire un niveau officiel
29     fileName = "Niveaux/Niveaux officiels/"+nom+".txt"
30     if os.path.isfile(fileName)==False: #Cas où le fichier n'existe pas
31         return None
```

2. Précisez les moyens utilisés :

Les moyens utilisés pour ce projet

Editeur de texte :

J'ai utilisé Visual studio code, qui est un environnement de développement intégré (IDE). Il permet la prise en charge native de divers langages d'où Python qui fut mon choix pour faire ce jeu, ce langage permet de faire des jeux ainsi que des applications.

Tests Unitaires :

Pour vérifier mes tests unitaires, j'ai utilisé Python Unittest qui est un outil qui permet de tester notre Code.

Documentations :

J'ai utilisé comme documentations le site :

DOSSIER PROFESSIONNEL ^(DP)

<https://www.pygame.org/news>

Ce site s'adresse aux personnes aimant coder les jeux, développeurs, chef de Projets, directeurs techniques ou, plus largement. Nous donne toutes les clefs nécessaires au développement d'un jeu en python.

5 les

3. Avec qui avez-vous travaillé ?

J'ai travaillé en tout autonomie sur ce projet avec l'accompagnement de Remy Viscaino

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *La plateforme*

Chantier, atelier, service ▶ Dans le cadre de la formation : Développement d'un jeu en Python

Période d'exercice ▶ Du : 20/04/2022 au : 23/04/2022

5. Informations complémentaires (facultatif)

Activité-type 2 Développer la persistance des données

Exemple n° 1 - Développement d'une API TODOLIST (API PLATFORM)

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

Une API, ou interface de programmation d'application, est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications.

Pour réaliser cette activité type, j'ai divisé l'activité en plusieurs phases qui sont les suivantes :

1. Phase d'analyse

Après le recensement de mes besoins, tout cela m'a emmené à établir des règles de gestion suivantes :

Un utilisateur ou plusieurs peuvent créer une ou plusieurs tâches ;

Une tâche peut être attribué à un ou plusieurs utilisateurs

Une tâche peut avoir un ou plusieurs tags

Un tag ou plusieurs tags ne peuvent posséder qu'une tâche

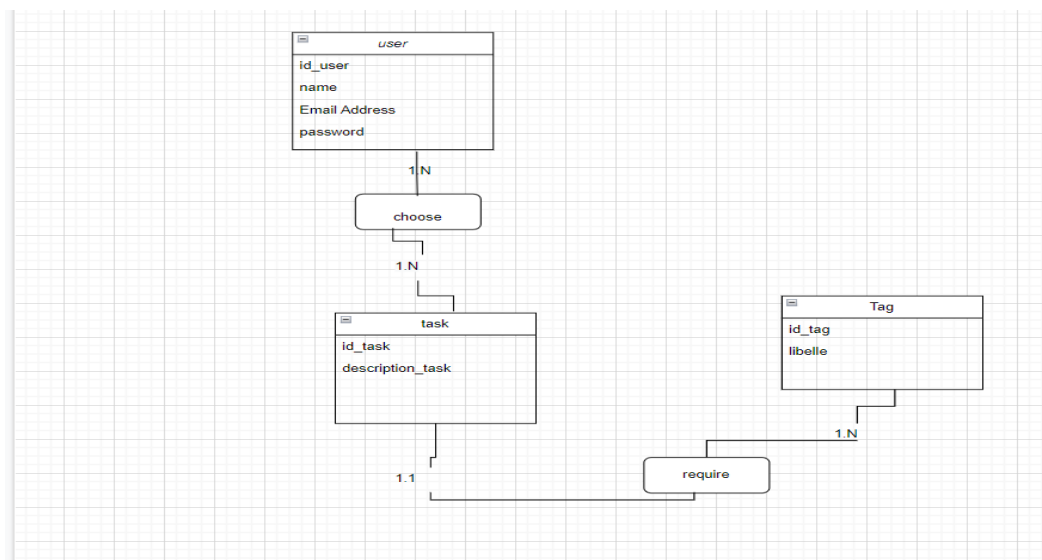
2. Phase de conception technique

Durant cette phase, J'ai mis en place les différents modèles

DOSSIER PROFESSIONNEL ^(DP)

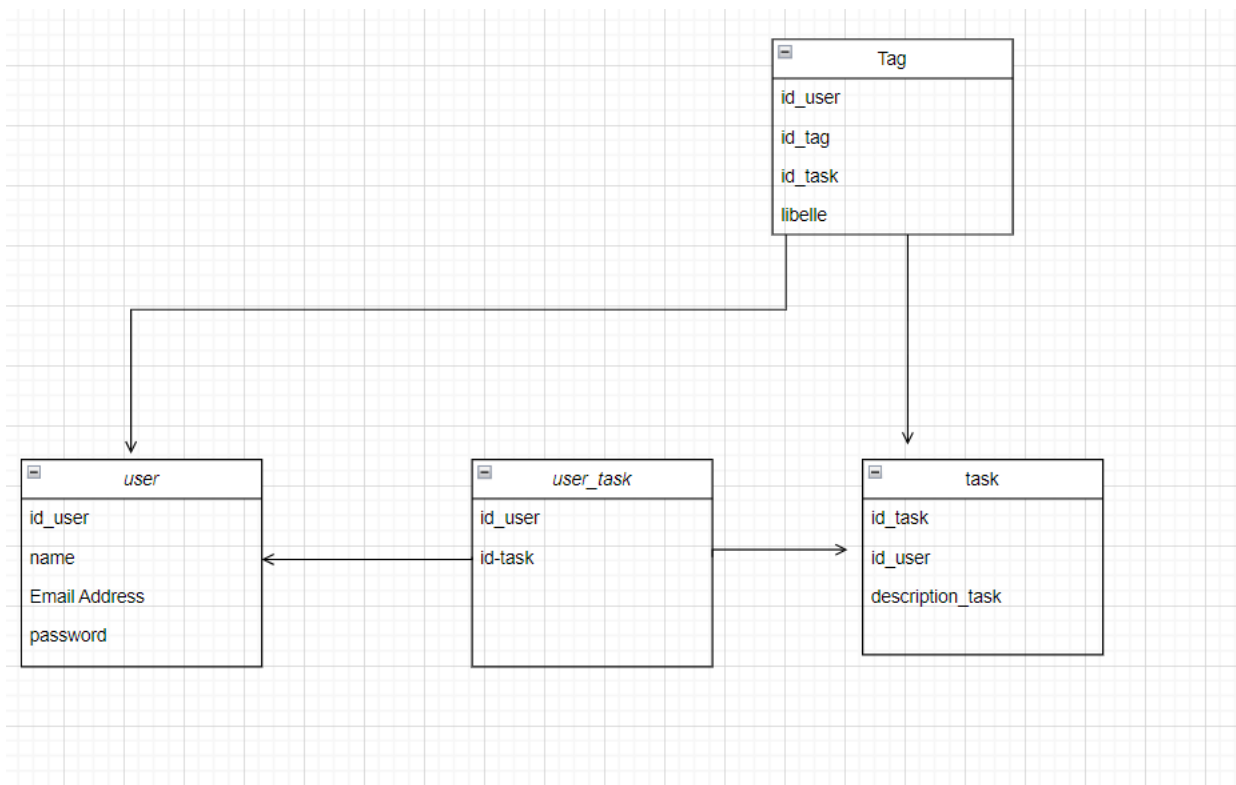
Voici les différents modèles que j'ai réalisés dans l'ordre suivant :

I. Le modèle conceptuel de données du type modèle entité-association. Qui permet de décrire le système d'information à l'aide d'entité.



II. Modèle logique de données, qui consiste à décrire la structure selon laquelle les données seront stockées dans la base de données. Voici le modèle logique de données

DOSSIER PROFESSIONNEL ^(DP)

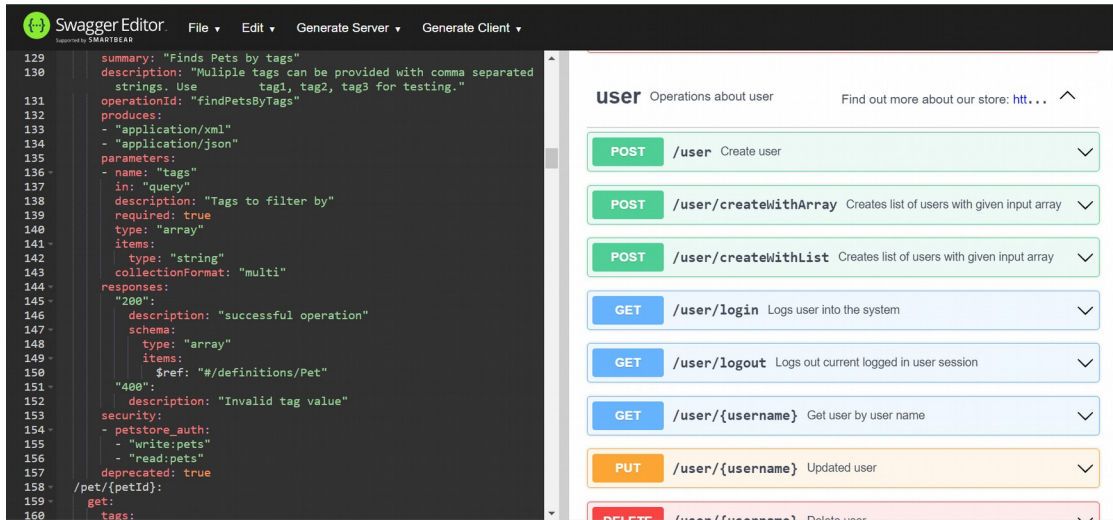


Après tout ceci, j'ai listé les différentes routes dont

API aura besoin. Cette liste représentera la documentation de

Mon API. Pour se faire, un outil tel que Swagger a été utilisé comme exemple ci-dessous :

DOSSIER PROFESSIONNEL ^(DP)



3. Phase de réalisation

Pour cette phase, nous avons tout réaliser en ligne de commande notamment la création de nos entités, qui par la suite nous a produit cette classe de type '@Entity' et généré automatiquement ma base de données ainsi que mes tables par le biais de Doctrine.

Ce script permet de créer une entité Task qui sera par la suite exposé sur notre API

```
<?php

namespace App\Entity;

use ApiPlatform\Core\Annotation\ApiResource;
use App\Repository\TaskRepository;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity(repositoryClass: TaskRepository::class)]
#[ApiResource]
class Task
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column(type: 'integer')]
    private $id;

    #[ORM\Column(type: 'string', length: 255)]
    private $description_task;

    #[ORM\ManyToOne(targetEntity: User::class, inversedBy: 'relation')]
    private $user_task;

    #[ORM\OneToMany(mappedBy: 'task', targetEntity: User::class)]
    private $relation;

    public function __construct()
    {
    }
```

Par la suite la création d'une base de données faire la migration est primordial :

```
$ php bin/console doctrine:database:create
```

```
$ php bin/console make:migration
```

Après cette commande notre table Task est créé et notre SQL est généré ce qui nous permet d'envoyer nos données

2. Précisez les moyens utilisés :

Les moyens utilisés pour ce projet :

Φ Outil de modélisation

J'ai utilisé Draw.io pour ce qui est de mon MCD et MLD, diagrams.net est un logiciel de dessin graphique multiplateforme gratuit et open source développée en HTML5 et JavaScript. Son interface peut être utilisée pour créer des diagrammes tels que des organigrammes, des structures filaires, des diagrammes UML, des organigrammes et des diagrammes de réseau.

SGBDR

J'ai utilisé comme système de gestion de base de données relationnelles PhpMyAdmin qui incorpore le langage SQL.

Outil de gestion de projet

Pour gérer ce projet, j'ai utilisé Github. Qui propose permet de stocker et gérer du code pour des projets de développement, un système de versionning, et surtout un tableau de bord qui permet de planifier chacune des fonctionnalités de l'application sous forme d'étiquettes.

Documentations

Je me suis documenté sur Symfony, voici deux sources qui me paraissent pertinentes :

https://symfony.com/doc/current/page_creation.html#creating-a-page-route-and-controller

<https://symfony.com/doc/current/doctrine.html#migrations-creating-the-database-tables-schema> :

ce site met à disposition une documentation complète sur les bonnes pratiques et l'utilisation de

DOSSIER PROFESSIONNEL ^(DP)

Symfony, Doctrine et la création des classes ainsi que la création d'une base de donnée et sa migration pour pouvoir créer nos différentes tables .

3. Avec qui avez-vous travaillé ?

J'ai travaillé en tout autonomie sur ce projet avec l'accompagnement de mon formateur Remy Viscaino

4. Contexte

Nom de l'entreprise, organisme ou association ► *La plateforme*

Chantier, atelier, service ► Dans le cadre de la formation : Développement d'une API

Période d'exercice ► Du : *29/11/2021* au : *09/12/21*

5. Informations complémentaires *(facultatif)*

Activité-

type 3 Développer une application

Exemple n° 1 ► Développement d'une application type TODOLIST

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

L'application Todolist permet l'ajout, la suppression de tâche, devant se connecter et inscription pour l'accès à chaque utilisateur pour notre application.

Scénario :

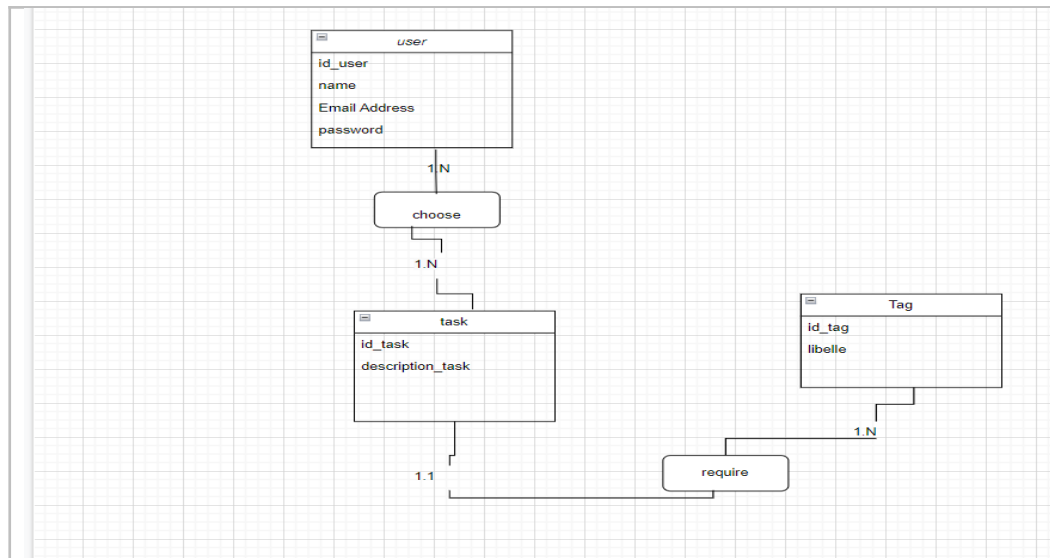
L'utilisateur se connecte, si l'utilisateur n'a pas de compte, il peut s'inscrire. Lorsque l'utilisateur s'est

enfin connecté, il accède à un espace où il pourra ajouter , modifier des tâches .

Conception technique :

Pour mieux illustrer et avoir une vue détaillée de notre application, j'ai réalisé un MCD pour mieux illustrer les parties importantes de mon application qui montrent les données qui seront utilisées par le système d'information ci-dessous :

DOSSIER PROFESSIONNEL ^(DP)



Réalisation

Composants métier :

J'ai codé ce composant de l'application ci-dessous :

Ce composant me permet de compléter, supprimer, mettre à jour les taches

DOSSIER PROFESSIONNEL ^(DP)

```
const Todo = ({ todos, completeTodo, removeTodo, updateTodo }) => {
  const [edit, setEdit] = useState({
    id: null,
    value: ''
  });

  const submitUpdate = value => {
    updateTodo(edit.id, value);
    setEdit({
      id: null,
      value: ''
    });
  };

  if (edit.id) {
```

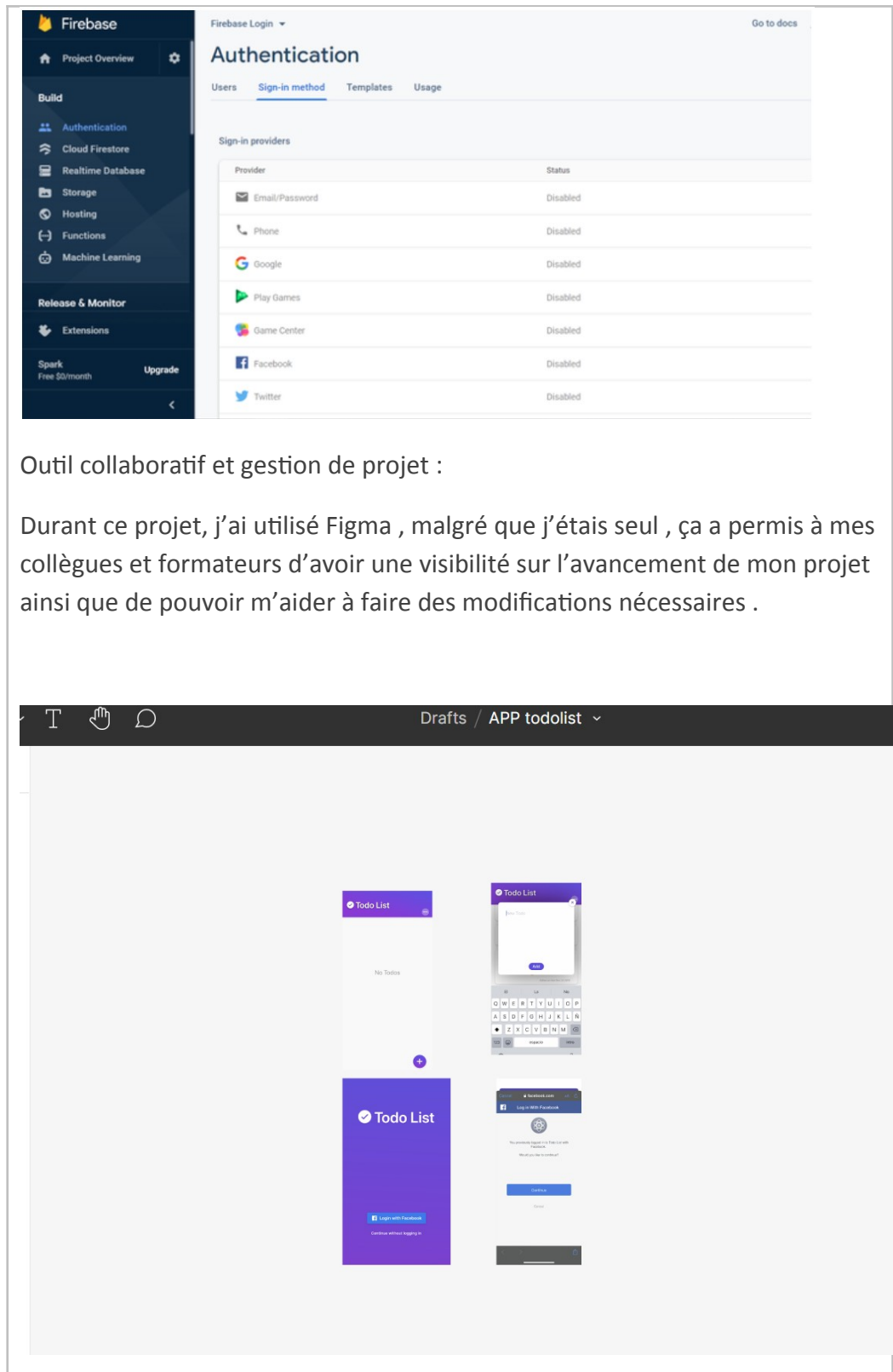
Système d'authentification

Pour mettre en place mon système d'authentification, j'ai installé dans mon projet avec NPM firebase:

```
npm install -g firebase
```

Cette application utilise l'authentification par facebook proposée par Firebase afin de sécuriser la base de données.

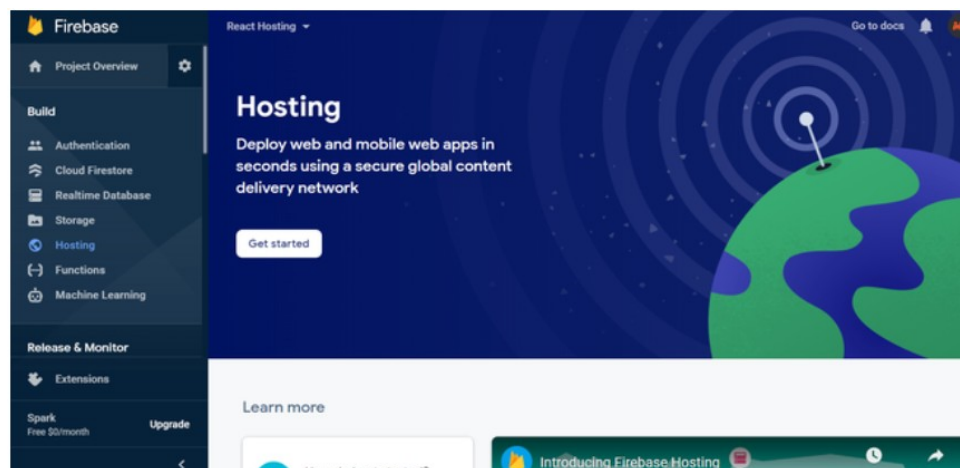
DOSSIER PROFESSIONNEL ^(DP)



Déploiement

Configuration de l'application

Pour configurer l'application, j'ai choisi firebase hosting qui permet de déployer une application que ce soit web ou mobile.



DOSSIER PROFESSIONNEL ^(DP)

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL ^(DP)

Les moyens utilisés pour ce projet :

Editeur de texte

J'ai utilisé Visual Studio code qui est un environnement de développement pour développer des

Applications mobiles. Il intègre le langage React, langage que nous avons décidé d'utiliser

pour ce projet.

Base de données

J'ai utilisé comme système de gestion de base de données firebase.

Outils collaboratifs et de gestion

Ce projet a été réalisé seul car du au COVID, j'ai eu des complications médicales, mais avec un accompagnement des formateurs.

J'ai utilisé :

Github : qui a permis d'avoir une base de projet identique. Avec le système de partage, mes formateurs ont pu accéder à mon projet pour un suivi.

Figma : qui est un outil collaboratif et de conception

Documentations

React: <https://fr.reactjs.org/tutorial/tutorial.html> & <https://reactnative.dev/>

DOSSIER PROFESSIONNEL ^(DP)

Firestore: <https://console.firebase.google.com/>

Expo: <https://expo.dev/>

3. Avec qui avez-vous travaillé ?

J'ai travaillé en toute autonomie sur ce projet avec l'accompagnement de mon formateur Remy Viscaïno

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *La plateforme*

Chantier, atelier, service ▶ Dans le cadre de la formation : développement d'une application Mobile

Période d'exercice ▶ Du : 20/06/2022 au : 05/07/22

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL ^(DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

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

DOSSIER PROFESSIONNEL ^(DP)

Déclaration sur l'honneur

Je soussigné(e) Boris Melono ,

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 Marseille

le 25/06/2022

pour faire valoir ce que de droit.

Signature :



DOSSIER PROFESSIONNEL ^(DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL ^(DP)

ANNEXES

(Si le RC le prévoit)

DOSSIER PROFESSIONNEL ^(DP)
