

# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance*      ➤ HADJADJI  
*Nom d'usage*        ➤ HADJADJI  
*Prénom*                ➤ MOHAMED  
*Adresse*                ➤ 44 TRAVERSE ADOUL BTC 13015 MARSEILLE

## Titre professionnel visé

*RNCP - Concepteur Développeur d'Application*

### 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.*

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

## Sommaire

<b>Intitulé de l'activité-type n° 1: Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>6</b>
- Maquetter l'application (notes de frais)	p.	6
- Développer une interface utilisateur de type desktop (Sokoban)	p.	11
- Développer des composants d'accès aux données (Notes de frais)	p.	14
- Développer la partie front-end d'une interface utilisateur web (Thalassanté)	p.	16
- Développer la partie back-end d'une interface utilisateur web (API notes)	p.	19
<b>Intitulé de l'activité-type n° 2: Concevoir et développer la persistance des données en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>23</b>
- Concevoir une base de données (Port de Marseille)	p.	23
- Mettre en place une base de données (Notes de frais)	p.	27
- Développer des composants dans le langage d'une base de données	p.	29
<b>Intitulé de l'activité-type n° 3: Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>32</b>
- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement	p.	32
- Concevoir une application	p.	35
- Développer des composants métier	p.	39
- Construire une application organisée en couches	p.	41
- Développer une application mobile	p.	44
- Préparer et exécuter les plans de tests d'une application	p.	50
- Préparer et exécuter le déploiement d'une application	p.	
<b>Titres, diplômes, CQP, attestations de formation (facultatif)</b>	<b>p.</b>	
<b>Déclaration sur l'honneur</b>	<b>p.</b>	

---

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

---

**Documents illustrant la pratique professionnelle** *(facultatif)*

p.

**Annexes** *(Si le RC le prévoit)*

p.

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

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

*Exemple n°1 ► Maquetter l'application notes de frais*

---

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

---

Une note de frais est une dépense relative au travail du salarié qui peut être remboursée par l'employeur. Exemple : restaurant, ticket de parking, billet de train...

Les sociétés souhaitent réaliser un logiciel pour simplifier le traitement de leurs notes de frais. Actuellement, les salariés envoient leurs notes de frais par email, ce qui génère beaucoup de travail pour le service RH. Nous souhaiterions réaliser un outil pour simplifier ce processus.

Via une application web, le salarié peut saisir une demande de remboursement pour une note de frais. Il devra uploader un justificatif (par exemple, une photo d'une facture d'un restaurant, ou un fichier PDF) et saisir le montant de la note. Il lui est possible d'ajouter un commentaire. Il doit également écrire son nom.

Dans une autre partie de l'application réservée au service RH, il est ensuite possible de valider ou de refuser les notes de frais.

Avant de commencer la maquette de l'application, j'ai fait une étude des applications concurrentes sur *Play Store* afin de me diriger vers un design à la fois esthétique et intuitif à utiliser, et aussi pour observer plus de détails afin de réaliser une application la plus performante et professionnelle possible.

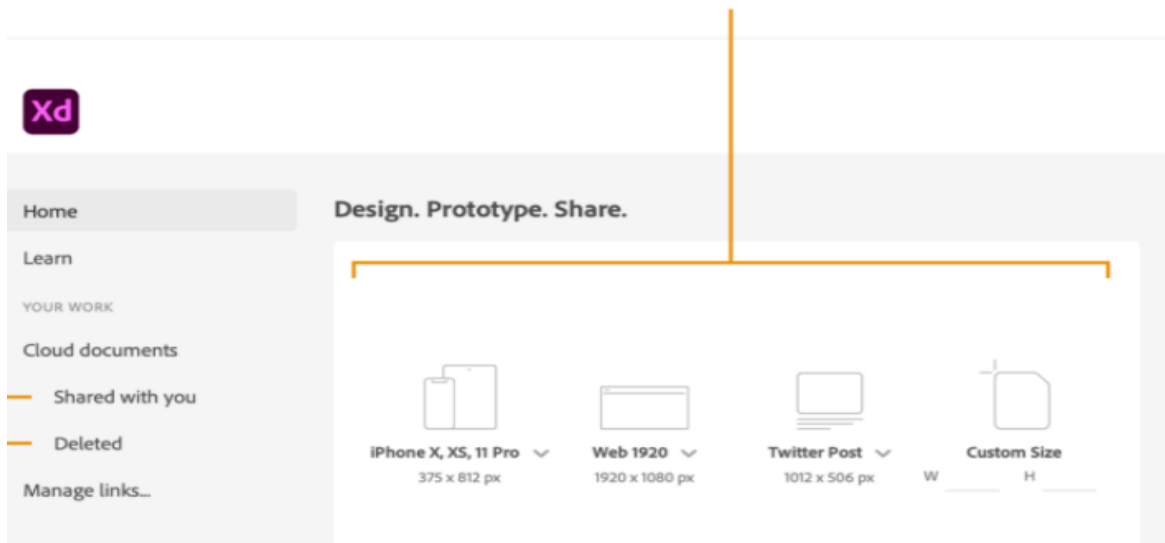
Ensuite j'ai créé la maquette qui illustre toutes les pages avec *Adobe XD*, grâce à sa bibliothèque d'éléments très fournie qui m'a permis de répertorier les zones du site pour une structuration optimale de l'information, et d'avoir des détails des zonings avec l'intégration du contenu dans les blocs. C'est un moyen de partager ma vision du projet avec les membres d'équipe et d'examiner les différentes possibilités.



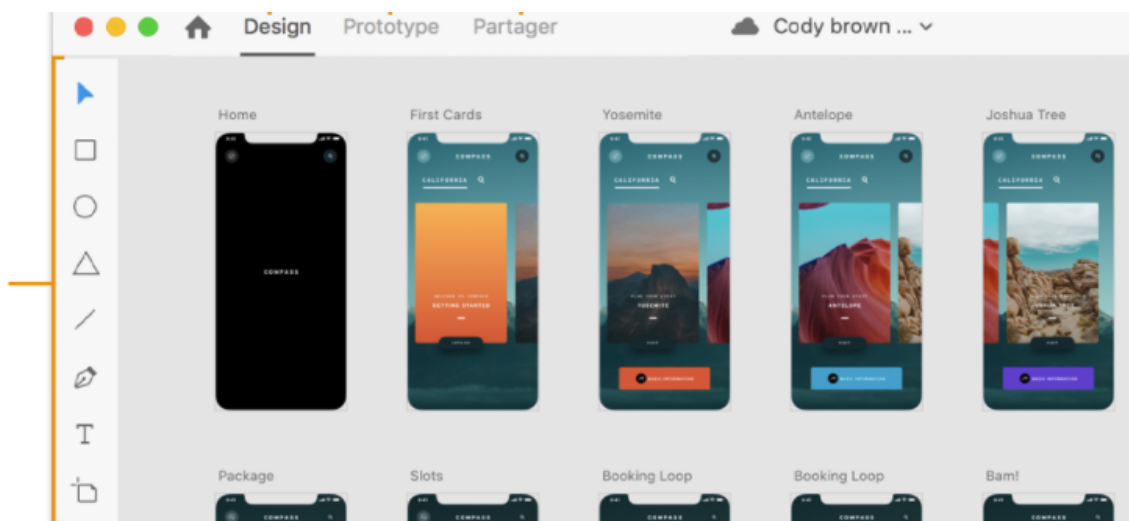
Dans Adobe XD, nous pouvons créer et manipuler nos designs et prototypes à l'aide de divers éléments, tels que les barres d'outils, les panneaux et l'inspecteur Propriétés. Ces éléments constituent l'espace de travail d'Adobe XD.

L'écran d'accueil vous permet d'accéder rapidement aux paramètres prédéfinis de plans de travail et aux fichiers récents.

# DOSSIER PROFESSIONNEL (DP)



J'ai utilisé L'espace de travail de **macOS**, puis le mode **Design**. Dans **la barre d'outils**, j'ai sélectionné **Rectangle**, pour dessiner les différents blocs des pages de l'affichage mobile, puis j'ai fait glisser des captures d'images dans ces cadres pour donner l'aspect d'écran mobile.

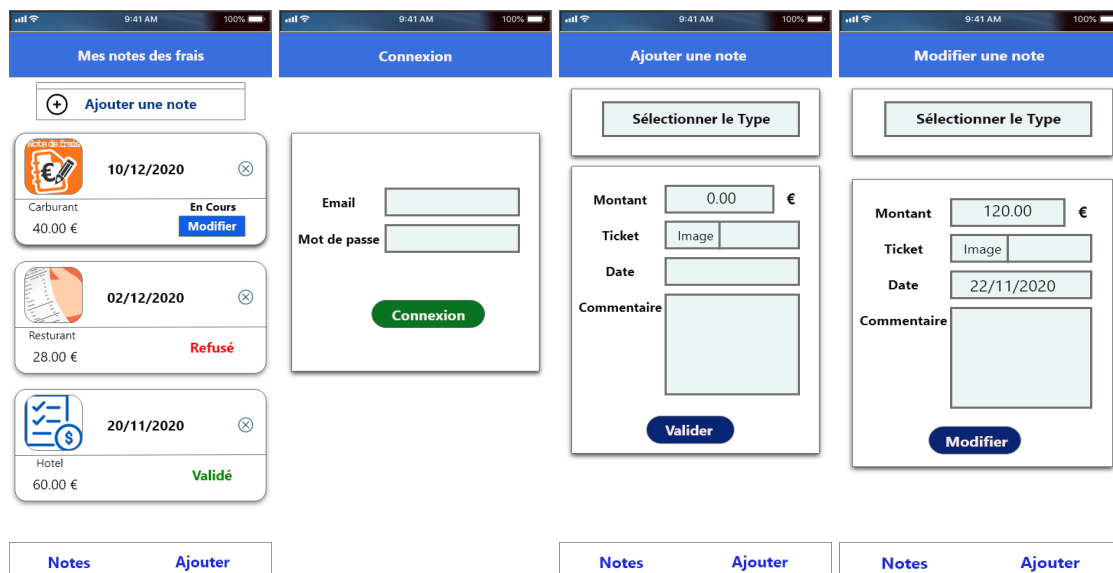


Pour les titres, les textes et labels j'ai utilisé l'outil **Texte**, j'ai modifier les polices, les couleurs et les tailles dans **l'Inspecteur Propriétés**.

Pour les boutons, j'ai utilisé encore des rectangles, avec des textes à l'intérieur, puis j'ai changé les couleurs de fond dans l'inspecteur propriétés.



# DOSSIER PROFESSIONNEL (DP)



## 2. Précisez les moyens utilisés :

J'ai utilisé *Adobe XD*, c'est un logiciel gratuit qui nous permet de designer des interfaces, sites internet et applications en toute simplicité. Ce logiciel met l'accent sur le prototypage.



## 3. Avec qui avez-vous travaillé ?

J'ai réalisé cette maquette avec mes camarades de projet, Grégory Fauvel et Olivier Crozet.

---

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

---

## 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *10/01/2021* au : *12/01/2021*

## 5. Informations complémentaires *(facultatif)*

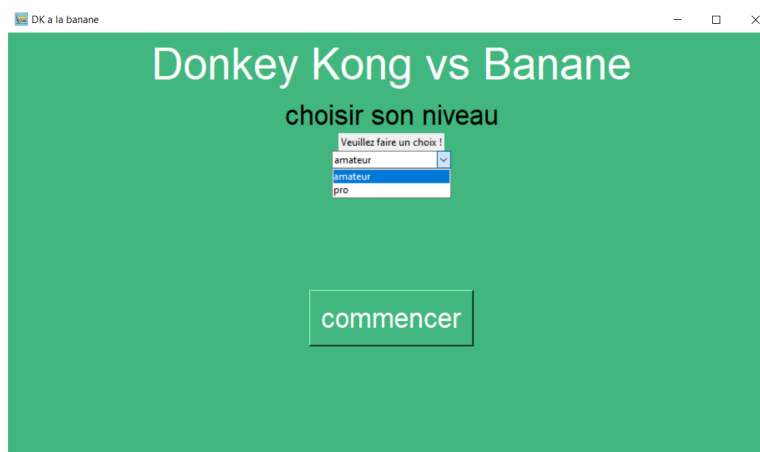
## Activité-type 1 Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

*Exemple n° 2 - Développer une interface utilisateur de type desktop (Sokoban)*

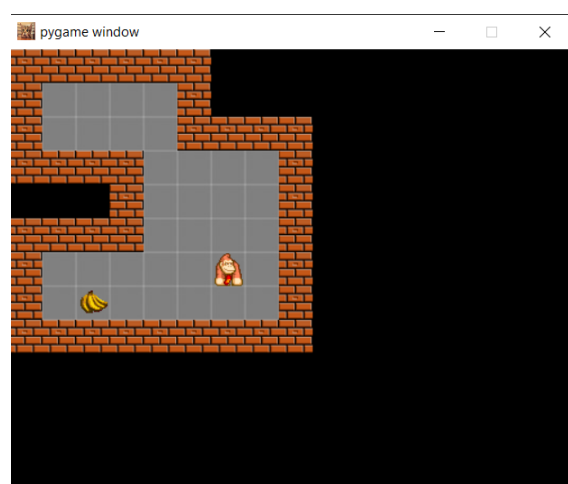
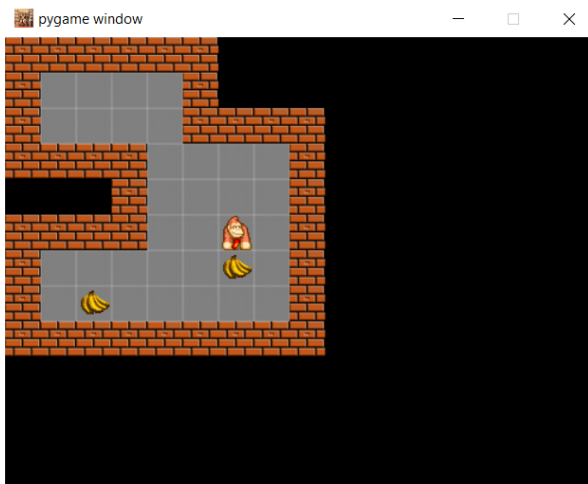
### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Nous avons développé un jeu Sokoban en **Python**. Le jeu utilise le personnage Donkey kong qui mange des bananes en suivant un chemin précis.

Au début du jeu nous arrivons sur une interface qui nous permet de choisir le niveau, ce qui changera la map.



En cliquant sur commencer, le jeu se lance, Le joueur peut alors se déplacer avec les touches directionnels de son clavier et manger le maximum de bananes.



La fonction qui interprète la carte est définie comme suit:

- Nous avons défini le 0 comme étant l'image des murs d'une taille de 30px.
- Le 1 comme étant le sol avec une taille de 30 px.
- Le 88 pour notre amis donkey kong.

```
#carte 1
PREFAB = [
[0,0,0,0,0,0,3,3,3],
[0,1,1,1,1,0,3,3,3],
[0,1,1,1,88,0,0,0,0],
[0,0,0,0,1,1,1,1,0],
[3,3,3,0,1,1,1,1,0],
[0,0,0,0,1,1,1,1,0],
[0,1,1,1,1,1,88,1,0],
[0,1,88,1,1,1,1,1,0],
[0,0,0,0,0,0,0,0,0]
]
```

```
def mappy():
    x = 0
    y = 0
    for y in range (0,9):
        for x in range (0,9):
            if PREFAB[y][x] == 0:
                screen.blit(murImage,(30 * x,30 * y))
            elif PREFAB[y][x] == 1:
                screen.blit(solImage,(30 * x,30 * y))
            elif PREFAB[y][x] == 88:
                screen.blit(solImage,(30 * x,30 * y))
                screen.blit(banane,(30 * x,30 * y))
            x +=1
        x =0
        y +=1
```

Pour faire fonctionner tout cela, nous avons créer la fenêtre qui affichera ce jeu paramétré de la façon suivante:

```
# creer une premiere fenetre
window = Tk()
# personnaliser la fenetre
window.title("DK a la banane")
# dimension fenetre
window.geometry("920x520")
window.minsize(480, 360)
# changer le logo
window.iconbitmap("logo/logo-la-plateforme.ico")
#couleur background
window.config(background='#41B77F')
```

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

### 2. Précisez les moyens utilisés :

Pour développer ce jeu, nous avons utilisé **Python**, c'est le langage de programmation open source, le plus utilisé dans le domaine du Machine Learning, du Big Data et de la Data Science.



### 3. Avec qui avez-vous travaillé ?

J'ai développé cette interface utilisateur avec mes camarades de projet, Grégory Fauvel et Olivier Crozet.

### 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : au :

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

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

Exemple n° 3 - Développer des composants d'accès aux données notes de frais

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

Notre application *notes de frais*, nécessite l'utilisation des données des utilisateurs et des notes pour son fonctionnement, nous avons donc créé une base des données composée de deux tables (**Users & Expense Reports**).

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> expense_reports	★ Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB	latin1_swedish_ci	16,0 kio	-
<input type="checkbox"/> users	★ Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB	latin1_swedish_ci	16,0 kio	-
2 tables	Somme	14	InnoDB	latin1_swedish_ci	32,0 kio	0 o

Pour développer les composants d'accès à la base de données, nous avons créé une **API REST** qui utilise la méthode **CRUD**. Le terme CRUD est étroitement lié avec la gestion des données numériques. Plus précisément, CRUD est un acronyme des noms des quatre opérations de base de la gestion de la persistance des données et applications :

- Create (créer)
- Read ou Retrieve (lire)
- Update (mettre à jour)
- Delete ou Destroy (supprimer)

Nos utilisateurs doivent pouvoir supprimer, ajouter, mettre à jour des données sur l'application. Ces quelques pages de code peuvent donc servir de base à un début de back office.

CRUD-Opération	SQL	RESTful HTTP	Description
Create	INSERT	POST	- Création des utilisateurs. - Création des notes.
Read	SELECT	GET	- Affichage des utilisateurs. - Affichage des notes.
Update	UPDATE	PUT	- Modification des utilisateurs. - Modification des notes.
Delete	DELETE	DELETE	- Supprimer des utilisateurs.

## DOSSIER PROFESSIONNEL (DP)

			- Supprimer des notes.
--	--	--	------------------------

### 2. Précisez les moyens utilisés :

Pour la création de la base des données, nous avons utilisé **phpMyAdmin**, un outil logiciel gratuit écrit en PHP destiné à gérer l'administration de MySQL sur le Web.



Pour la création de l'API REST, nous avons utilisé **CodeIgniter 3**, un *framework d'application Web* pour les développeurs qui préfèrent la vitesse à un grand choix de fonctions. Selon la page officielle de CodeIgniter, l'objectif principal du framework PHP open-source est d'offrir un maximum de performance et de flexibilité dans une structure de programmation la plus simple possible.



Pour la partie code php, nous avons utilisé **Visual Studio Code** comme IDE.



### 3. Avec qui avez-vous travaillé ?

J'ai développé cette API avec mes camarades de projet, Grégory Fauvel et Olivier Crozet.

### 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *22/12/2020* au :

## Activité-type 1

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

**Exemple n° 4** - Développer la partie front-end d'une interface utilisateur web (Thalassanté)

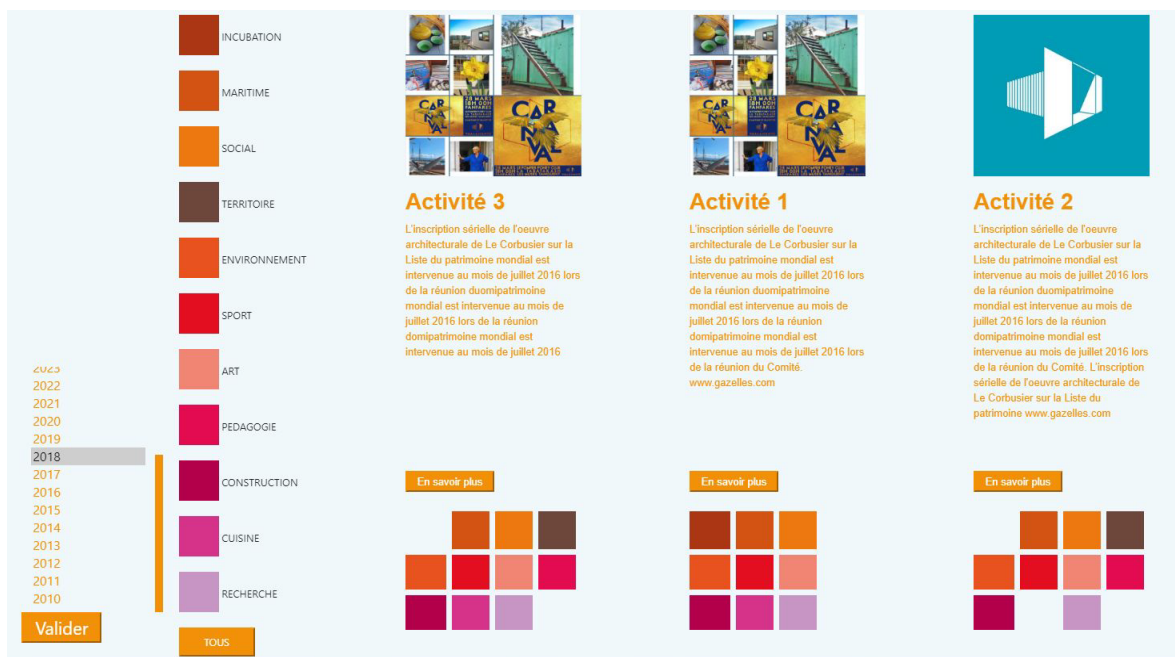
#### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Thalassanté est une association et un lieu de vie composé de conteneurs maritimes aménagés implanté en bord de mer, à l'Estaque, depuis 20 ans.

L'objectif de ce projet est de créer le site de l'association dépourvue d'un site web. C'est une seule page, qui contient plusieurs parties ( Présentation - Images - Lieux - Activités - Agenda - Évènements - Contact).

Pour réaliser l'interface statique et adaptable de les parties Activités et événements, j'ai utilisé HTML, CSS, JAVASCRIPT.

J'ai commencé par la structure et la mise en place des balises html avec les images, textes et boutons, tout en respectant l'ordre des éléments par rapport au cahier des charges. Ensuite avec le CSS en utilisant de "flex" et "media queries", j'ai respecté l'emplacement des éléments ainsi que la taille qui s'adapte avec l'affichage de l'écran (laptop, tablette, smartphone).







J'ai utilisé Javascript pour cacher la suite de texte de présentation d'activités, et en cliquant sur la flèche, la suite s'affiche. Et ça grâce aux événements jQuery **“.click”** et **“.toggle”**.

```
/////////fait apparaitre le texte
$('#mh-js-img-icone').click(function(){
    $('#oc-para-act-cacher').toggle(250); //affiche et desaffiche
})
```

J'ai ajouté aussi des popup pour les affiches activités lorsque l'utilisateur clic sur “En savoir plus”, en utilisant l'événement **“.dialog”**.

```
$(function() {
    var fiche = "#fic"+ act[i]['id'];
    $('`${fiche}`').dialog({
        autoOpen: false,
        width: 1030
    });
    var bofic = "#affic"+ act[i]['id'];
    $(document).on("click", `${bofic}`, function() {
        $('`${fiche}`').dialog('open');
    });
});
```

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

## 2. Précisez les moyens utilisés :

Pour insérer les codes HTML et CSS et JS j'ai utilisé "Sublim text".

## 3. Avec qui avez-vous travaillé ?

J'ai fait ce projet avec mon camarade Olivier.

## 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *10/06/2020* au : *02/07/2020*

## 5. Informations complémentaires (facultatif)

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

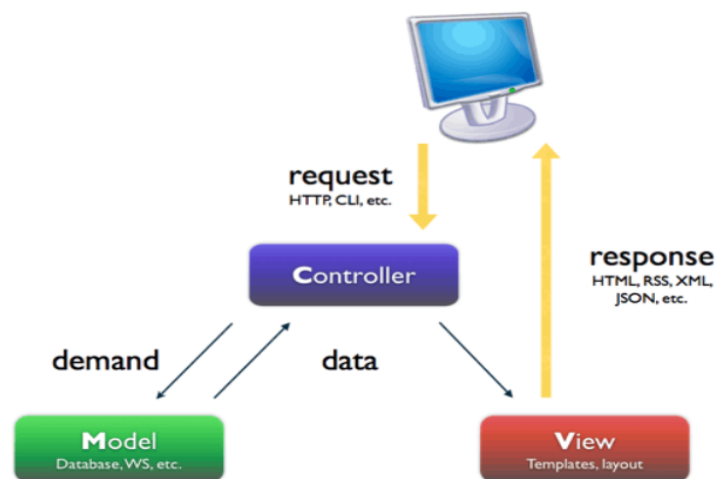
**Exemple n° 5** - Développer la partie back-end d'une interface utilisateur web (API notes)

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

L'API notes représente la partie back-end de projet, Pour la développer nous avons utilisé CodeIgniter3. Nous commençons donc par l'installer en téléchargeant le fichier codeigniter sous format zip via le lien:

<https://codeigniter.com/userguide3/installation/downloads.html>

En décompressant le fichier zip dans notre serveur *plesk*, ça nous a permis d'avoir une structure de fichier MVC (**Model - View - Controller**).



- **Model:** Le modèle se compose des structures de données. D'une certaine manière, ceux-ci contiennent généralement les ensembles de fonctions qui aident à récupérer, insérer et mettre à jour les informations dans la base de données.
- **View:** La vue est l'information qui est présentée à l'utilisateur. Une vue dans le cas normal est une page Web, mais dans le cas de Codeigniter, il peut s'agir d'un fragment de page comme un en-tête ou un pied de page, ou il peut s'agir de tout autre type de page comme RSS ainsi que pour l'exécution de l'interface utilisateur.
- **Controller:** Le contrôleur implique l'implémentation logique pour l'écriture du programme et la création de la page Web. Le contrôleur sert d'intermédiaire entre le modèle et la vue et toutes les autres ressources nécessaires pour traiter la requête HTTP et générer une page Web.

Dans le fichier *application/config/config.php*, nous avons défini l'URL de notre base de données ('<http://mohamed-hadjadji.students-laplateforme.io/apinote/>'). Puis nous avons inséré les

coordonnées de la base des données dans le fichier *application/config/database.php*.

```
'hostname' => 'localhost',  
'username' => 'root',  
'password' => 'root',  
'database' => 'mohamed-hadjadji_ems',  
'dbdriver' => 'mysqli',
```

Dans le *Model*, nous avons créé deux fichiers models (**User\_model & Note\_model.php**):

- **User\_model:** qui contient les fonctions suivantes:
  - ❖ *Create()*: Pour insérer des utilisateurs.
  - ❖ *Connect()*: Pour la connexion des utilisateurs.
  - ❖ *Logout()*: Pour la déconnexion.
  - ❖ *All()*: Pour afficher la liste des utilisateurs.
  - ❖ *GetUser(\$userId)*: Pour sélectionner l'identifiant de l'utilisateur.
  - ❖ *UpdateUser(\$userId)*: Pour la modification de l'utilisateur, sélectionné par son id.
  - ❖ *DeleteUser(\$userId)*: Pour supprimer l'utilisateur, sélectionné par son id.
  - ❖ *Note\_user(\$userId)*: Pour afficher les notes de frais de l'utilisateur, sélectionné par son id.
- **Note\_model:** qui contient les fonctions suivantes:
  - ❖ *Create\_note()*: Pour insérer des notes de frais.
  - ❖ *All\_note (\$userId)*: Pour afficher la liste des notes de l'utilisateur connecté.
  - ❖ *GetNote (\$noteId)*: Pour sélectionner l'identifiant de la note.
  - ❖ *UpdateNote (\$noteId)*: Pour la modification de la note, sélectionné par son id.
  - ❖ *DeleteNote (\$noteId)*: Pour supprimer la note, sélectionné par son id.
  - ❖ *UpdatestatutNote (\$noteId)*: Pour modifier le statut de la note, sélectionné par son id.

Dans le Controller, nous avons créé deux fichiers controllers (**User.php & Note.php & jwtToken.php**):

- **User:** qui contient les fonctions suivantes:
  - ❖ *Index ()*: Qui appelle la fonction *All()* de *User\_model*, pour la récupération des données utilisateurs en format *JSON*.
  - ❖ *GetId (\$userId)*: Qui appelle la fonction *GetUser (\$userId)* de *User\_model*, et qui sélectionne l'id utilisateur.
  - ❖ *Create()*: Qui appelle la fonction *Create()* de *User\_model*, pour l'insertion des données utilisateurs.
  - ❖ *Edit (\$userId)*: Qui appelle la fonction *UpdateUser (\$userId)* de *User\_model*, pour la modification de l'utilisateur sélectionné par son id.
  - ❖ *Connect ()*: Qui appelle la fonction *Connect()* de *User\_model*, pour la génération de **token**, et la connexion des utilisateurs
  - ❖ *Disconnect()*: Qui appelle la fonction *Logout ()* de *User\_model*, pour la déconnexion.

- ❖ *Delete (\$userId)*: Qui appelle la fonction *UpdateUser(\$userId)* de *User\_model*, pour supprimer l'utilisateur, sélectionné par son id.
- ❖ *NoteUser (\$userId)*: Qui appelle la fonction *Note\_user(\$userId)* de *User\_model*, pour sélectionner les notes de l'utilisateur par son id en format *JSON*.
- **Note:** qui contient les fonctions suivantes:
  - ❖ *Index\_note (\$userId)*: Qui appelle la fonction *All\_note(userId)* de *Note\_model*, pour la récupération des données des notes de l'utilisateur connecté, en format *JSON*.
  - ❖ *GetIdNote (\$noteld)*: Qui appelle la fonction *GetNote(\$noteld)* de *Note\_model*, et qui sélectionne l'id note.
  - ❖ *Create\_note ()*: Qui appelle la fonction *Create\_note()* de *Note\_model*, pour l'insertion des données notes de frais.
  - ❖ *Edit\_note (\$noteld)*: Qui appelle la fonction *UpdateNote (\$noteld)* de *Note\_model*, pour la modification de la note sélectionnée par son id.
  - ❖ *Delete\_note (\$noteld)*: Qui appelle la fonction *DeleteNote (\$noteld)* de *Note\_model*, Pour supprimer la note, sélectionné par son id.
  - ❖ *EditRH\_note (\$noteld)*: Qui appelle la fonction *UpdatestatutNote (\$noteld)* de *Note\_model*, pour modifier le statut de la note, sélectionné par son id.
- **jwt Token:** qui contient la fonction *LoginToken()*, qui génère le *Token* une fois que l'utilisateur est connecté.

Nous n'utilisons pas de View, car l'affichage se fait dans la partie front de l'application, qui utilise l'API note avec ses routes, appelant les différentes fonctions du controller, pour toutes les tâches réalisés dans l'application.

## 2. Précisez les moyens utilisés :

Pour la création de l'API REST, nous avons utilisé **CodeIgniter 3**, un *framework d'application Web* pour les développeurs qui préfèrent la vitesse à un grand choix de fonctions. Selon la page officielle de CodeIgniter, l'objectif principal du framework PHP open-source est d'offrir un maximum de performance et de flexibilité dans une structure de programmation la plus simple possible.



Pour la partie code php, nous avons utilisé **Visual Studio Code** comme IDE.

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

Pour tester l'ensemble des routes de l'API, nous avons utilisé **Postman**.



POSTMAN

### 3. Avec qui avez-vous travaillé ?

J'ai développé cette API avec mes camarades de projet, Grégory Fauvel et Olivier Crozet.

### 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *22/12/2020* au :

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

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

*Exemple n° 1 - Concevoir une base de données du projet Port de Marseille*

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

Il s'agit de créer une application, qui sera capitale pour le bon fonctionnement du port de Marseille. C'est par elle que se fera l'organisation des navires entrant et sortant, que seront gérées les ressources du port tels que les quais, ainsi que de tout le personnel employé par le port. Il est donc indispensable que l'application soit complète, infaillible d'un point de vue sécurité, et facile d'utilisation.

Le port de Marseille est une infrastructure construite par l'homme, située sur le littoral maritime Méditerranéen. Il est aménagé avec des quais qui sont reliés à d'autres moyens de transport (routier, ferroviaire, etc.). Il emploie des dockers et des marins qui effectuent différentes actions afin de s'assurer du déroulement de l'activité. Il existe différents types de quais qui permettent d'accueillir un ou plusieurs types de bateau. Il existe quatre types de bateau :

- Les bateaux de marchandise
- Les bateaux de croisière
- Les bateaux militaires.
- Les bateaux de plaisance.

Les bateaux appartiennent à une compagnie (hors bateaux de plaisance) qui est reliée à un pays. Ces compagnies ont par défaut une activité (marchande, militaire, tourisme...). Une compagnie peut cependant avoir plusieurs types d'activité.

Chacun des bateaux appartient à une catégorie:

- Lourd
- Moyen
- Léger

Chacun des bateaux a un type de chargement particulier. Il en existe trois types en fonction de ce qu'ils transportent :

- Urgent
- Dangereux
- Normal

Chacun des bateaux doit avoir une localisation :

- En mer
- A quai

Lorsqu'un bateau est à quai, il doit être possible de savoir quel quai. Un historique d'entrée et de sortie doit permettre de savoir quel bateau était à quel endroit à une date donnée.

Il existe plusieurs types de missions auxquelles les marins ou les dockers peuvent être assignés:

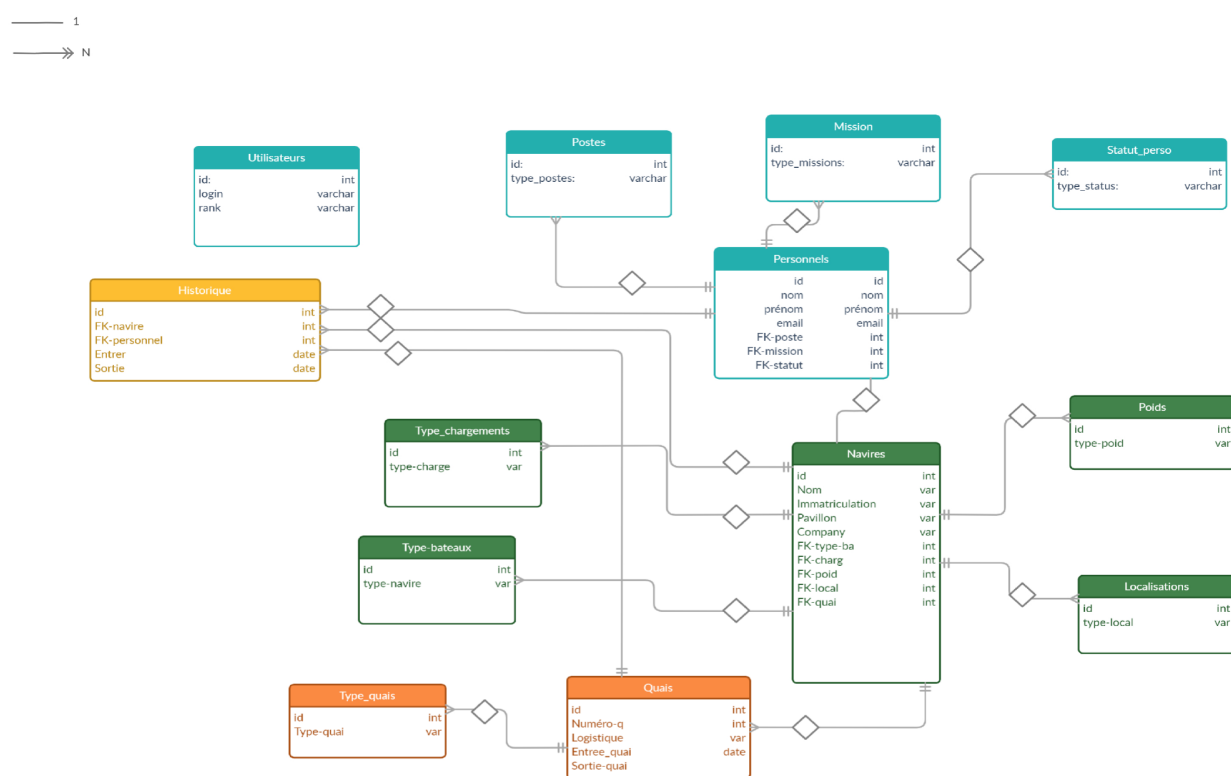
- Déchargement
- En mer

# DOSSIER PROFESSIONNEL (DP)

- Repos
- En attente de job
- Maintenance

Quand un bateau quitte le port, il faut y assigner des marins. Le port doit garder un historique des chantiers : quels dockers et marins travaillaient sur quels chantiers et quelles étaient leurs missions. L'historique doit également comporter le bateau et le quai concerné, les dates de début et de fin de l'évènement.

Tout d'abord, il faut mettre en place une base de données complète, exposer sous forme de **MPD**. Nous avons utilisé



L'application contient deux parties :

## - Une partie administrateur ou gestion :

C'est un tableau de bord qui contient 4 commandes et qu'ils sont :

- **Insertion Personnels** : qui ouvre une page qui contient la liste de tout le personnel de port avec toutes les informations (nom, prénom, email,...), et un bouton supprimer pour chaque ouvrier, et aussi un bouton en haut de la liste qui affiche le formulaire d'inscription personnel pour insérer toutes les informations des ouvriers. En cliquant sur une personne s'affiche un formulaire pour modifier ses informations. Enfin une barre de recherche pour le personnel.

Cette commande utilise 4 tables de la base de données :

**(Personnels - Postes - Missions - Statut\_perso).**



- **Insertion Navires:** qui ouvre une page qui contient la liste de tous les navires de port avec toutes les informations (nom, immatriculation, company, pavillon,... ), un bouton supprimer pour chaque bateau, et aussi un bouton en haut de la liste qui affiche le formulaire d'insertion navire pour insérer toutes les informations des bateaux. Enfin une barre de recherche pour les navires.

Cette commande utilise 5 tables de la base de données:

**(Navires - Type\_chargements - Type\_bateaux - Poids - Localisations).**

- **Inscription utilisateurs:** qui ouvre une page qui contient le formulaire Inscription utilisateur, avec Login, Password et rank, et qui permet d'ajouter des utilisateurs pour l'application et définit le statut (administrateur, utilisateur).

Cette commande utilise la table **Utilisateurs**.

## - Une partie Responsable personnels:

C'est un tableau de bord qui contient 4 commandes et qu'ils sont:

- **Personnels:** qui ouvre une page qui contient la liste de tout le personnel de port avec toutes les informations (nom, prénom, email,...), et aussi une barre de recherche pour le personnel.

Cette commande utilise 4 tables de la base de données:

**(Personnels - Postes - Missions - Statut\_perso).**

- **Navire:** qui ouvre une page qui contient la liste de tous les navires de port avec toutes les informations (nom, immatriculation, company, pavillon,... ), et aussi une barre de recherche pour les navires.

Cette commande utilise 5 tables de la base de données:

**(Navires - Type\_chargements - Type\_bateaux - Poids - Localisations).**

- **Quais:** qui ouvre une page qui contient la liste de tous les quais de port avec toutes les informations (numéro, type, entrée, sortie... ).

Cette commande utilise 2 tables de la base de données:

**(Quais - Type\_quais).**

## - Une partie commune:

- **Connexion:** une page avec un formulaire de connexion pour les utilisateurs de l'application.

Cette commande utilise la table **Utilisateurs**.

- **Historique:** qui ouvre une page avec des listes des personnels et navires ainsi que les quais. et qui affiche historique des chantiers:

quels dockers et marins travaillaient sur quels chantiers et quelles étaient leurs missions.

Cette commande utilise la table **Historiques**.

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

## 2. Précisez les moyens utilisés :

Pour réaliser le MPD de ce projet, nous avons utilisé le site.

<https://app.creately.com/>



## 3. Avec qui avez-vous travaillé ?

J'ai réalisé le MPD de la base de données avec mes camarades de projet, Grégory Fauvel et Olivier Crozet.

## 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *22/11/2020* au : *22/11/2020*

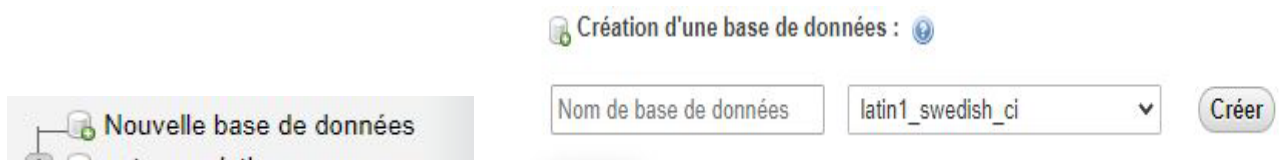
## 5. Informations complémentaires *(facultatif)*

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

*Exemple n° 2 - Mettre en place une base de données (notes de frais)*

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

Nous avons créé cette base de données avec “phpMyAdmin”, sur la table des commandes, puis sur “Nouvelle base de données” nous avons inséré le nom, puis valider.



En suite nous avons créé 2 tables qui sont les suivantes :

1. Une table **Users** pour stocker toutes les informations d'utilisateur insérées pendant l'inscription. Elle est composée de:

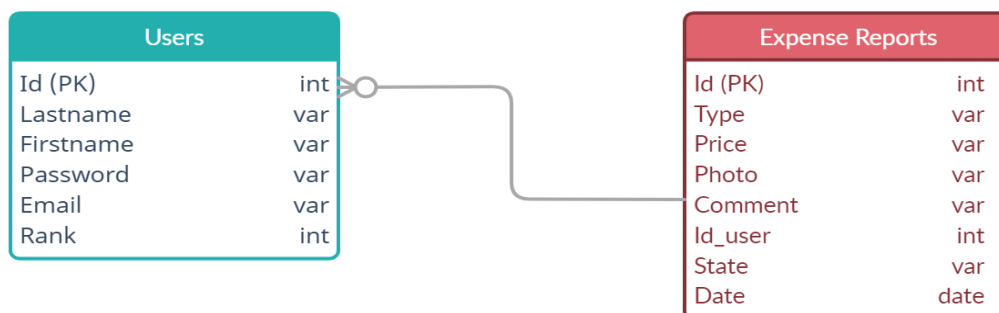
- Id utilisateur en int auto\_increment pour les numéros d'identification.
- Lastname en varchar avec une taille de 255.
- Firstname en varchar avec une taille de 255.
- Password en varchar avec une taille de 255.
- Email en varchar avec une taille de 255.
- Rank en int.

2. Une table **Expense Reports** pour stocker les informations de chaque note de frais inséré par l'utilisateur

pendant la création des articles dans l'admin panel. Elle est composée de:

- Id note en int auto\_increment.
- Type en varchar taille 255.
- Prix en varchar taille 65.
- Photo en varchar taille 255.
- Comment en varchar taille 255.
- Id\_user en int.
- State en varchar 255.
- Date en date.

## DOSSIER PROFESSIONNEL (DP)



### 2. Précisez les moyens utilisés :

Pour la création de la base de données, nous avons utilisé “phpMyAdmin”, un outil logiciel gratuit écrit en PHP destiné à gérer l'administration de MySQL sur le Web.

Pour réaliser le MPD de ce projet, nous avons utilisé le site.

<https://app.creately.com/>



### 3. Avec qui avez-vous travaillé ?

J’ai réalisé le MPD et la base de données avec mes camarades de projet, Grégory Fauvel et Olivier Crozet.

### 4. Contexte

Nom de l’entreprise, organisme ou association ▶ *Centre de formation La plateforme.*

Chantier, atelier, service ▶ *Conception développement d’application*

Période d’exercice ▶ Du : 27/11/2020 au : 27/11/2020

### 5. Informations complémentaires (facultatif)

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

*Exemple n° 3 - Développer des composants dans le langage d'une base de données*

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

#### Table USERS Class Model\_Users du Contrôleur User

- **Create (create) Création d'un Employés (RH)**  
function create(\$formArray)  
{  
    \$this->db->insert('users',\$formArray)}  
}
- **Read ou Retrieve (Userlist) Liste des Utilisateurs (RH)**  
function read ()  
{  
    return \$users = \$this->db->get('users')->result\_array();//select tous les Users}  
}
- **Update (updateUser) Modification compte Employés (RH)**  
function updateUser(\$userId,\$formArray) {  
    \$this->db->where('id',\$userId);  
    \$this->db->update('users',\$formArray);//update Users where id =  
}  
}
- **Delete or Destroy (delete) Suppression compte Employés (RH)**  
function deleteUser(\$userId) {  
    \$this->db->where('id',\$userId);  
    \$this->db->delete('users');//delete Users where id =  
}  
}

#### Table Notes Classe Model\_Note du Contrôleur Note

- **Create (create) Création d'une note(Employé)**

```
function create_note($formArray)
{$this->db->insert('expense_reports', $formArray);}
```

- **Read ou Retrieve (Notelist)** Liste des notes de l'employé connecté.(Employé)

```
function getNote($noteld)
{$this->db->where('id',$noteld);
return $note =$this->db->get('expense_reports')}
```

- **Update (edit)** Modification de la note (Employé->statut)

Employé->   function updateNote(\$noteld,\$formArray)

```
          {$this->db->where('id',$noteld);
          $this->db->update('expense_reports',$formArray);}
```

RH->       function updatestatutNote(\$noteld,\$formArray)

```
          {$this->db->where('id',$noteld);
          $this->db->update('expense_reports',$formArray);}
```

- **Delete or Destroy (delete)** Suppression de la note (Employé)

```
function deleteNote($noteld)
{$this->db->where('id',$noteld);
$this->db->delete('expense_reports'); //DELETE FROM expense_reports where id = ?
}
```

## 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.*

---

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

---

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 une application multicouche répartie en intégrant les recommandations de sécurité

**Exemple n° 1** - Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement (Note de frais)

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

Pour l'organisation de notre projet, nous avons utilisé des outils, qui fournissent un environnement professionnel adapté :

- **Trello** : c'est l'outil de gestion visuelle du travail qui permet aux équipes de proposer des idées, de planifier, de gérer et de célébrer leur travail ensemble de manière collaborative, productive et organisée.

Que mon équipe et moi-même commençons quelque chose de nouveau ou que nous essayions d'organiser notre travail existant, Trello s'adapte à n'importe quel projet. Il nous aide à simplifier et standardiser le processus de travail de notre équipe de manière intuitive. Trello est convivial, mais permet de gérer les projets les plus complexes de notre équipe.

- **Diagramme de Gantt** : couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois etc.). Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin. Ce diagramme permet donc de visualiser d'un seul coup d'œil :

- Les différentes tâches à envisager
- La date de début et la date de fin de chaque tâche
- La durée escomptée de chaque tâche
- Le chevauchement éventuel des tâches, et la durée de ce chevauchement
- La date de début et la date de fin du projet dans son ensemble

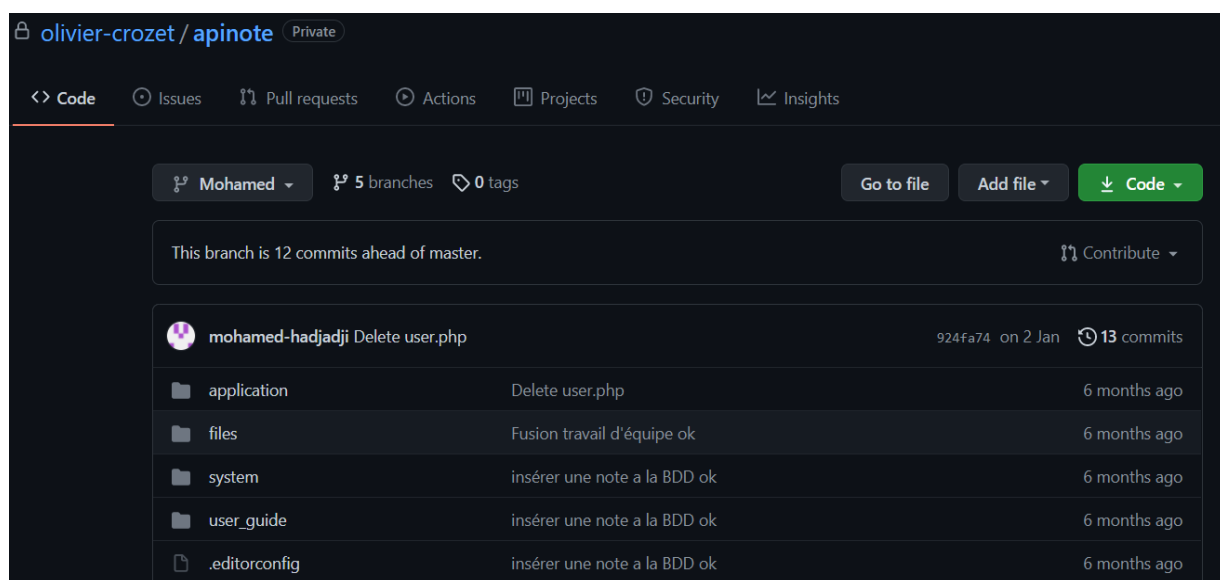
En résumé, un diagramme de Gantt répertorie toutes les tâches à accomplir pour mener le projet à bien, et indique la date à laquelle ces tâches doivent être effectuées (le planning).



# DOSSIER PROFESSIONNEL (DP)

NUMÉRO	TITRE DE LA TÂCHE	PROPRIÉTAIRE DE LA TÂCHE	DATE DE DÉBUT	DATE LIMITE	DURÉE	TÂCHE TERMINÉE (EN %)	SEMAINE 1					SEMAINE 2					SEMAINE 3					SEMAINE 4				
							L	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L	M	M	J	V
<b>1</b>	<b>Conception et mise en route du projet</b>																									
1.1	Diagramme bdd	Gregory	26/11/20	27/11/20	1	100 %																				
1.1.1	Creation bdd	Gregory	27/11/20	27/11/20	0	100 %																				
1.2	Documentation api	Gregory	15/01/21	21/01/21	6	90 %																				
1.3	Maquette	Mohamed	10/01/21	12/01/21	2	90 %																				
<b>2</b>	<b>Développement api</b>																									
2.1	Insertion d'une personne	Gregory	22/12/20	26/12/20	4	100 %																				
2.2	Récupération de la liste des personnes existantes	Gregory	27/12/20	30/12/20	3	100 %																				
2.3	Récupérer les informations détaillées d'une personne	Gregory	03/01/21	05/01/21	2	100 %																				
2.5	Supprimer une personne	Gregory	06/01/21	08/01/21	2	100 %																				
2.6	Créer une note de frais	Mohamed	22/12/20	26/12/20	4	100 %																				
2.7	Récupérer la liste des notes de frais	Mohamed	22/12/20	26/12/20	4	100 %																				
2.8	Récupérer les informations relatives à une note de frais	Mohamed	27/12/20	29/12/20	2	100 %																				
2.9	Mettre à jour les informations relatives à une note de frais	Olivier	27/12/20	29/12/20	2	100 %																				
2.10	Supprimer une note de frais	Mohamed	03/01/21	06/01/21	3	100 %																				

- Github:** les développeurs informatiques du monde entier utilisent ce service, devenu en une décennie seulement l'un des sites les plus populaires.  
 Il permet aux développeurs de stocker et de partager, publiquement ou non, le code qu'ils créent. La plate-forme accueille ainsi, dit-elle, plus de 80 millions de projets, qu'il s'agisse de logiciels, de sites Web, d'applications pour mobile ou tous autres types de programme informatique, et ce quel que soit le langage de programmation utilisé.  
 Le site est aussi un espace collaboratif. Chaque utilisateur peut contribuer aux projets mis en ligne publiquement sur GitHub, en proposant des modifications.



## 2. Précisez les moyens utilisés :

Pour la gestion du projet, nous avons utilisé **Trello**



Pour la mise en place d'un **Diagramme de gantt**, nous avons utilisé **Google Sheets**



Pour héberger et partager les fichiers avec les membres d'équipes:



### 3. Avec qui avez-vous travaillé ?

J'ai géré et organisé ce projet avec mon équipe (Gregory et Olivier)

### 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *25/11/2020* au : *26/11/2020*

### 5. Informations complémentaires (facultatif)

## Activité-type 3 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 2 - Concevoir une application (Notes des frais)

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

#### 1.1 Explorer et examiner

Avant de se lancer dans le développement de l'application, il faut savoir précisément vers quel résultat on veut se diriger, mais également vers celui qu'on veut absolument éviter. Pour cela, nous avons téléchargé et examiné un panel varié d'applications déjà existantes. Visiter un maximum d'applications permet de chercher de l'inspiration, mais aussi de développer un regard critique vis-à-vis de certaines fonctionnalités.

#### 1.2 Cahier des charges et besoins

##### 1.2.1 Une interface utilisateur pour le RH:

Les ressources humaines de l'entreprise ont besoin de voir la liste des salariés avec leur informations personnelles, et ils ont le droit de créer, modifier ou supprimer des comptes utilisateurs. Et ils ont aussi l'accès aux notes de frais ajoutées par les salariés, avec la possibilité de valider ou refuser ces notes de frais en cours.

Après la connexion l'utilisateur avec le statut *RH*, a une barre de navigation qui contient deux pages:

- **Page Liste d'utilisateurs:**

C'est la page d'atterrissage, avec toutes les informations des utilisateurs (noms, prénoms, emails, statuts), et aussi trois boutons qui sont:

- **Modifier:** Pour afficher la page *Modifier l'utilisateur*, elle contient un formulaire avec les informations de l'utilisateur, et la possibilité de les modifier.
- **Effacer:** Pour la suppression d'un utilisateur, en cliquant sur le bouton une fenêtre d'alerte s'affiche, pour confirmer ou annuler.
- **Notes:** Pour afficher la page *Notes d'utilisateur*, elle contient la liste de notes de frais de la personne concerné, plus un bouton pour modifier le statut (Validé, Refusé).

- **Page Ajouter un utilisateur:**

C'est la page de création des comptes utilisateurs à l'aide d'un formulaire de création qui contient (nom, prénom, mot de passe, email, statut). Et un bouton pour la validation.

## 1.2.2 Une interface utilisateur pour les salariés

Le salarié à l'accès pour sa liste des notes de frais, avec leur informations. Il a la possibilité de créer et supprimer ses propres notes, ou les modifier si elles sont en mode en cours.

Après la connexion de salarié avec son email et le mot de passe, il a une barre de navigation qui contient deux pages:

- **Page Liste des notes:**

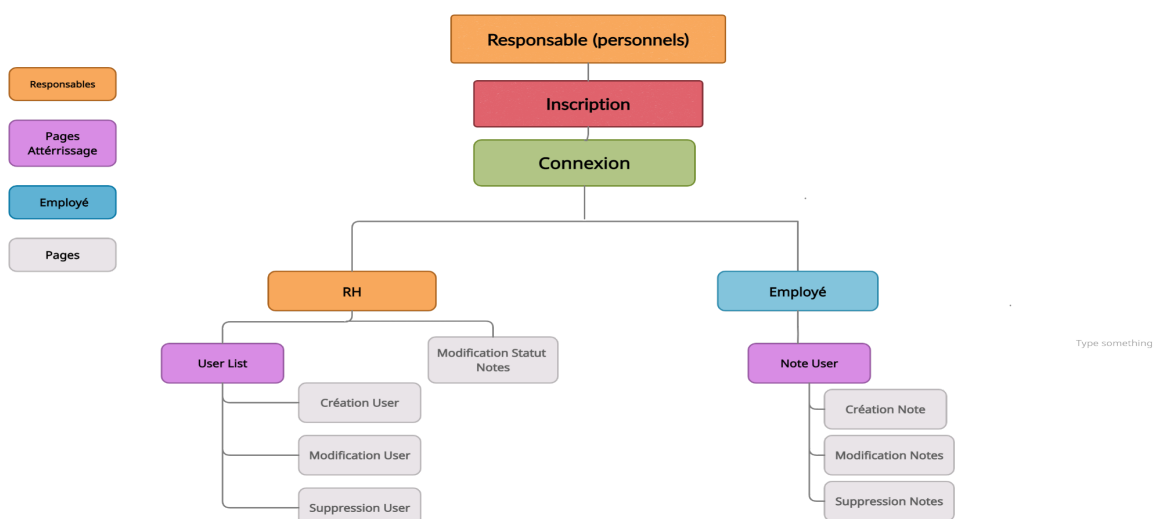
C'est la page d'atterrissage, avec toutes les informations des notes (types, montants, date, statut), et aussi le justificatif de la facture en photo ou fichier PDF. ainsi que deux boutons pour:

- **Modifier la note:** seulement si la note est en mode *En Cours*. On cliquant sur modifier, la page *modifier une note* s'affiche, elle contient un formulaire avec les informations de la note, il est donc possible de les modifier si besoin.
- **Supprimer:** pour effacer une note de sa propre liste, en cliquant sur le bouton une fenêtre d'alerte s'affiche, pour confirmer ou annuler.

- **Page Ajouter une note:**

Le salarié peut aussi accéder à cette page en cliquant sur le bouton plus, situé au début de la liste des notes. Cette page contient le formulaire de création des notes frais avec ( Type, Montant, Ticket, Date, Commentaire). Et aussi la possibilité d'uploader une photo ou un fichier pdf de la facture justificative.

Après analyse des besoins dans le cahier des charges, nous avons maqueté l'arborescence des pages à afficher pour l'application, nous avons donc défini deux parties, qui s'affichent selon le statut de l'utilisateur:



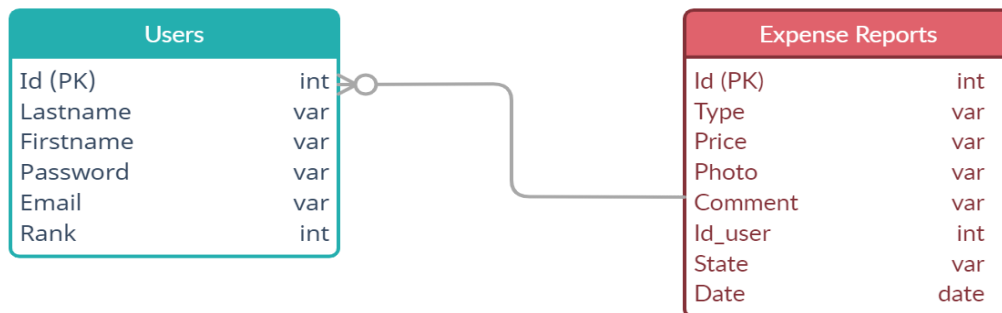
# DOSSIER PROFESSIONNEL (DP)

## 1.3 Maquetter l'application

Nous avons maqueté l'application en utilisant Adobe XD, pour plus de détails, vous pouvez consulter *l'exemple 01 de l'activité type 01*.

## 1.4 Conception de la base de données

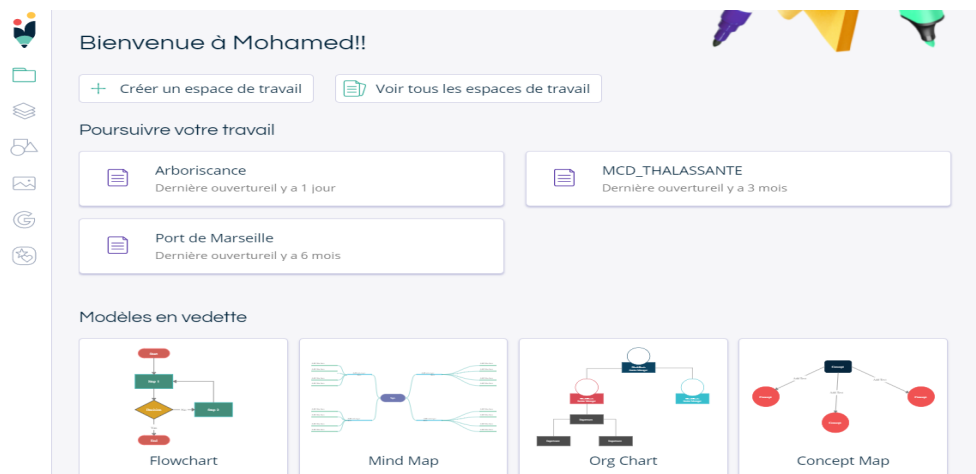
Nous avons créé le MCD de la base de données en utilisant le site *Creately.com*



## 2. Précisez les moyens utilisés :

Pour réaliser la maquette de l'arborescence de l'application, nous avons utilisé le site.

<https://app.creately.com/>



Nous avons utilisé *Adobe XD*, c'est un logiciel gratuit qui nous permet de designer des interfaces, sites internet et applications en toute simplicité. Ce logiciel met l'accent sur le prototypage.

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

### 3. Avec qui avez-vous travaillé ?

J'ai conçu ce projet avec mes camarades Gregory et Olivier.

### 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *10/01/2021* au : *15/01/2021*

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

## Activité-type 3 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 3 - Développer des composants métier (Notes de frais)

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

Les composants métiers de notre application sont classés en deux parties:

#### 1.1 Composants métiers pour le RH:

- Se connecter en statut RH
- Afficher les salariés.
- Ajouter des salariés.
- Modifier les salariés.
- Supprimer des salariés.
- Valider ou refuser les notes de frais des salariés.

#### 1.2 Composants métiers pour les salariés:

- Se connecter en statut salarié.
- Afficher les notes de frais.
- Ajouter des notes de frais.
- Modifier les notes de frais en mode "En-Cours".
- Supprimer des notes de frais.

### 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.

---

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

---

Période d'exercice ▶ Du : Cliquez ici au : Cliquez ici

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

---



## Activité-type 3 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

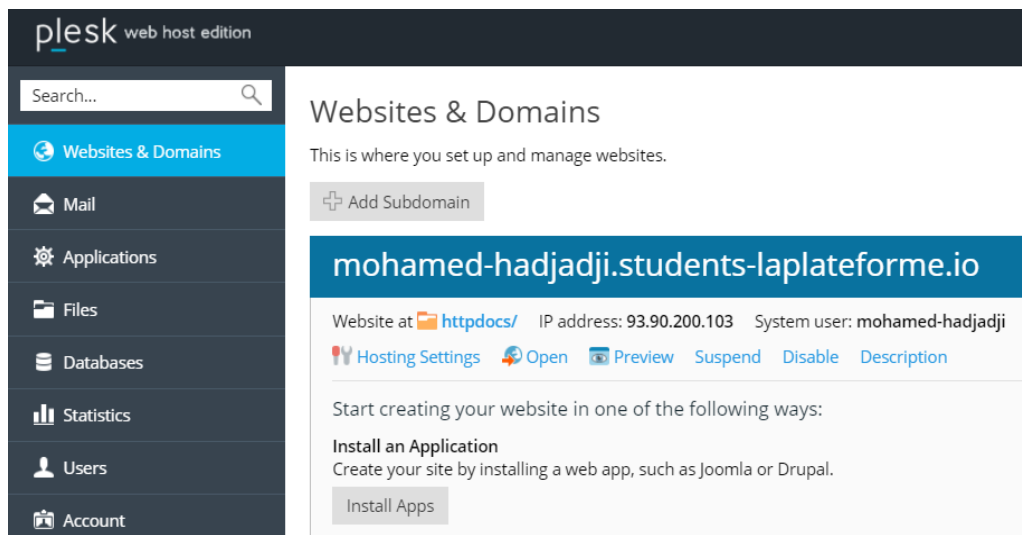
Exemple n° 4 - Construire une application organisée en couches (Notes de frais)

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

Notre architecture de l'application contient trois couches, réparties en deux parties:

#### 1.1 Partie Serveur

Notre serveur est un hébergeur qu'on peut gérer avec une interface de gestion **Plesk**.



Cette partie contient deux couches:

##### 1.1.1 La Base de données:

Nous avons créé la bdd de l'application avec la commande "Database" puis "Add database". Cette option utilise *phpMyAdmin*.

Plesk utilise Le mode de sécurité renforcé protège les données confidentielles stockées dans la base de données, et cela permet de:

- Tous les mots de passe stockés dans la base de données Plesk sont chiffrés à l'aide de la clé secrète de Plesk. Ainsi, même si un tiers obtient un dump de la base de données Plesk, vos clients ne sont pas compromis.
- Les données confidentielles (par. : les mots de passe utilisateur) peuvent être récupérées à l'aide de l'API de Plesk.
- Les mails de récupération de mots de passe ne contiennent plus le mot de passe en texte brut. À

la place, il contient un lien vers un formulaire de réinitialisation du mot de passe.

### 1.1.2 API note:

Dans la commande "Files", nous avons téléchargé tous les dossiers et les fichiers de code qui sont réparties en *MVC*, que nous avons créés avec **CodeIgniter 3**.

L'API récupère les informations envoyées par les utilisateurs (RH - salariés), et les insère dans la bdd, et elle envoie les données à l'application sous forme **JSON**.

### 1.2 Partie Application

L'application notes de frais que nous avons développé avec **React Native**, représente la couche de l'utilisateur, c'est l'interface de communication entre les utilisateurs et la bdd. Pour cela, nous avons utilisé **Axios**, c'est une bibliothèque JavaScript fonctionnant comme un client HTTP. Elle permet de communiquer avec des API en utilisant des requêtes. Comme avec les autres clients HTTP, il est possible de créer des requêtes avec les méthodes (POST - GET - PUT - Delete).

Schéma Architecture en couches:



## 2. Précisez les moyens utilisés :

Pour héberger l'API et la bdd nous avons utilisé Plesk:

**plesk**

Pour la création de l'API REST, nous avons utilisé **CodeIgniter 3**



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

Pour développer l'application nous avons utilisé **React Native**



Pour la partie code php, nous avons utilisé **Visual Studio Code** comme IDE.



### 3. Avec qui avez-vous travaillé ?

J'ai conçu ce projet avec mes camarades Gregory et Olivier.

### 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *15/01/2021* au : *25/02/2021*

### 5. Informations complémentaires (facultatif)

## Activité-type 3 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 5 - Développer une application mobile (Note de frais)

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

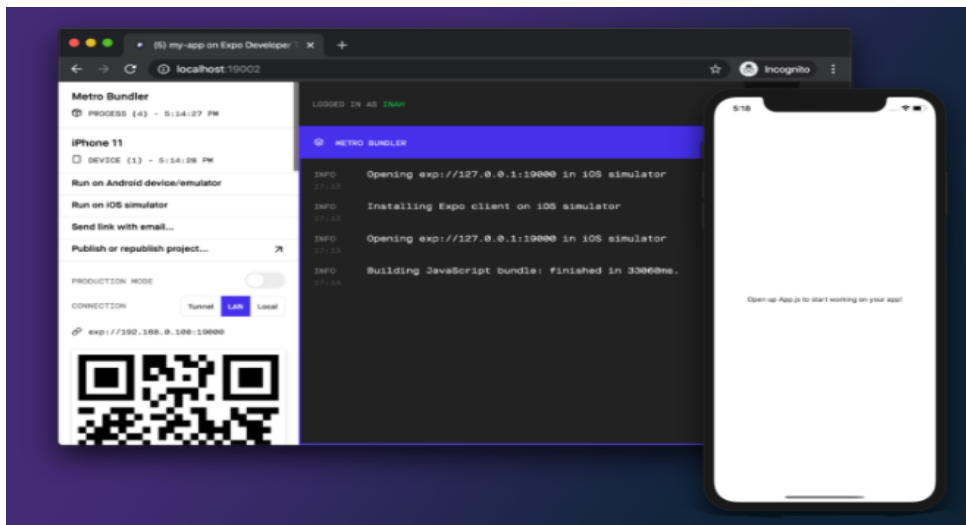
Nous avons développé l'application note de frais avec **React Native**, c'est un framework qui utilise le langage de programmation JavaScript et les bibliothèques React développées par Facebook pour développer des interfaces webs. Sauf qu'il s'agit désormais de mobile.

Le moyen le plus simple de commencer est d'utiliser **Expo CLI**. Expo est un ensemble d'outils construits autour de React Native et, bien qu'il possède de nombreuses fonctionnalités, la fonctionnalité la plus pertinente pour nous en ce moment est qu'il peut nous permettre d'écrire une application React Native en quelques minutes.

Nous avons donc utilisé **npm** pour installer Expo avec la commande `npm install -g expo-cli`, dans le dossier "appapinote", que nous avons créé pour l'application, et qui va contenir tous les sous-dossiers et les composantes. Puis nous avons installé **Expo Go** dans le téléphone via *Play Store*. Qui va devenir notre simulateur pour tester notre application en temps réel.



Pour lancer Expo, nous avons utilisé la commande `npm start` ou `expo start`, il ouvrira **Expo Developer Tools**, une interface utilisateur graphique basée sur un navigateur permettant de configurer rapidement les appareils de test et émulateurs pour le développement d'Expo, l'appareil de diffusion en continu et les journaux de regroupement et le déploiement de mises à jour.



Après l'exécution de l'application, nous avons commencé par la modifier, en ouvrant le fichier **App.js**, c'est le point de départ de notre application qui contient un composante standard prédéfinis du React native (**App**).

1. Pour La navigation entre les pages, nous avons ajouté **Navi.js**, puis on a installer et importer les bibliothèques suivantes:

```
import { NavigationContainer } from '@react-navigation/native';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
import { createStackNavigator } from '@react-navigation/stack';
```

Après nous avons ajouté la fonction **NavHome()**, qui utilise un système de navigation **Stack.Navigator**

```
export function NavHome() {
  return (
    <NavigationContainer independent={true}>
      <Stack.Navigator headerMode={false}>
        <Stack.Screen name="HomePage" component={HomePage} />
        <Stack.Screen name="MainPage" component={MainPage} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

Cette fonction est appelée par *App.js* pour afficher en premier la page de *Connexion des Utilisateurs* "**HomePage**", Puis une redirection vers la page "**MainPage**", qui fait la navigation entre les deux parties de l'application (RH - Salarié), en récupérant le **Rank** de l'utilisateur après connexion, via le **Token** envoyer par l'**API**, puis stocker dans le **Local Storage**.

## HomePage

```
axios

.post("https://mohamed-hadjadji.students-laplateforme.io/apinote/index.php/User/
connect",data)

    .then((response) => {
        axios.defaults.headers.common['Token'] =
`Bearer${response.data}`; // une fois connecté, on configure axios pour qu'il
utilise le token à chaque requête
        this.save('token', `${response.data.substr(1)}`);
        // faire une redirection vers MainPage.js
        // console.log('hello',this.props.navigation);
        this.props.navigation.navigate('MainPage');
    })
```

## MainPage

```
import jwt_decode from "jwt-decode";

const MainPage = (props) => {

    const [doRedirect, setDoRedirect] = useState(0);

    useEffect(() => {
        const fetchData = async() => {
            if (doRedirect === 0) {
                await SecureStore.getItemAsync('token').then((val) => {
                    var decoded = jwt_decode(val);
                    if (decoded.rank == "RH") {
                        setDoRedirect(1)
                        console.log("test");
                    } else {
                        setDoRedirect(2)
                    }
                })
            }
        }
        if (doRedirect === 0) return <></>;
        if (doRedirect === 1) return <AppNav />;
        if (doRedirect === 2) return <AppBout />;
    })
```

Nous avons utilisé **jwt\_decode** pour décoder le Token.

2. Puis nous avons ajouté les fonctions **Navi()** et **Navi\_Salarie()** dans Navi.js, pour naviguer entre les

pages de chaque partie de l'application (RH - Salariés). En utilisant le système de navigation `Tab.Navigator`.

```
const Tab = createBottomTabNavigator();
function Navi() {
  return (
    <Tab.Navigator>
      <Tab.Screen name="Users" component={UserList} />
      <Tab.Screen name="Ajouter User" component={Inputs} />
    </Tab.Navigator>
  );
}
```

```
const TabSalarie = createBottomTabNavigator();

export function Navi_salarie() {

  return (

    <TabSalarie.Navigator>
      <TabSalarie.Screen name="Notes" component={Notes}/>
      <TabSalarie.Screen name="InsertionNote" component={InsertionNote}/>
    </TabSalarie.Navigator>

  );
}
```

3. Pour faire la connexion entre l'application et la base de données, en utilisant l'API note, nous avons fait appel à la bibliothèque **Axios**. En installant avec la commande `npm install axios`.

Une fois installé, nous avons communiqué avec l'API en utilisant des requêtes avec les méthodes (POST - GET - PUT - Delete). Par exemple:

### Récupérer la liste des notes de l'utilisateur

```
async componentDidMount() {
  await SecureStore.getItemAsync('token').then((val) => {
    var decoded = jwt_decode(val);
    var iduser = decoded.id;
    console.log(decoded);
  });
}
```

```
axios.get('https://mohamed-hadjadji.students-laplateforme.io/apinote/index.php/Note/index_note/'+idUser)
    .then(response => {

        this.setState(prevState => ({
            NoteList: prevState.NoteList = response.data

        }));
    });
```

### Supprimer une note

```
Delete = (id ) => {
    axios

.delete("https://mohamed-hadjadji.students-laplateforme.io/apinote/index.php/Note/delete_note/"+id )
    .then((response) => {
        this.props.navigation.navigate('Notes');
    })
    .catch((err) => {
        throw err;
    });
};
```

## 2. Précisez les moyens utilisés :

Pour développer l'application nous avons utilisé **React Native**, et **Expo**



Pour la partie code php, nous avons utilisé **Visual Studio Code** comme IDE.





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

## 3. Avec qui avez-vous travaillé ?

J'ai conçu ce projet avec mes camarades Gregory et Olivier.

## 4. Contexte

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : *15/01/2021* au :

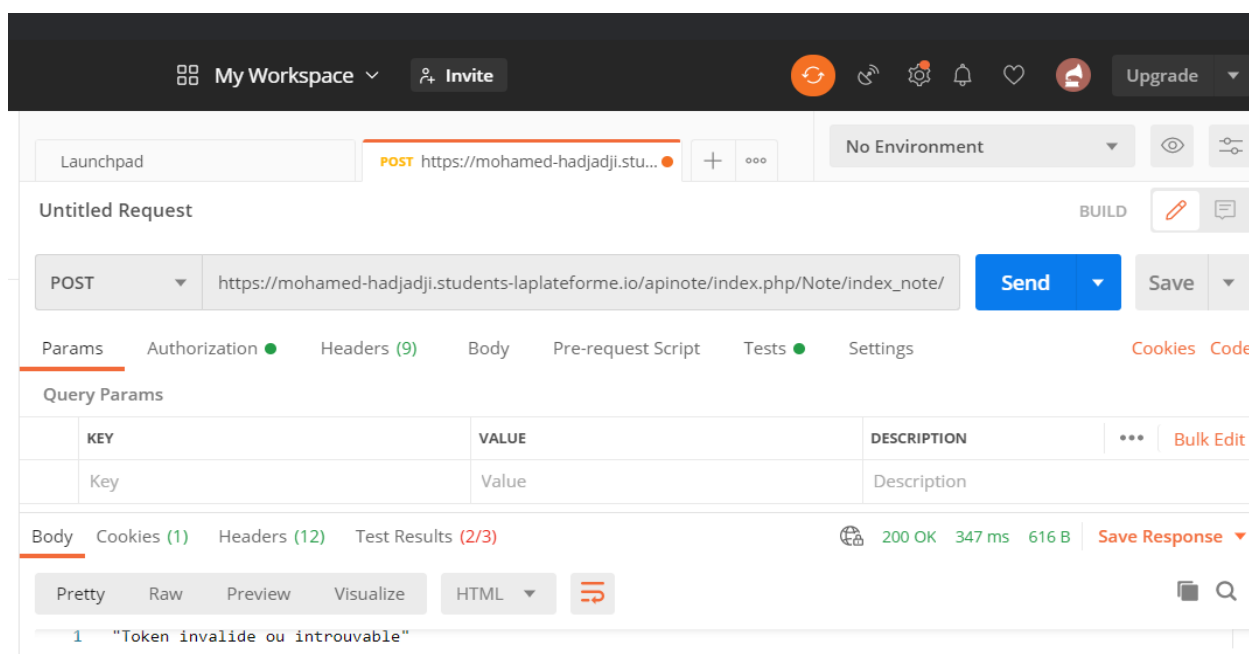
## 5. Informations complémentaires *(facultatif)*

## Activité-type 3 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 6 - Préparer et exécuter les plans de tests d'une application (Notes de frais)

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

Pour effectuer des tests sur l'API, nous avons utilisé **Postman** qui nous permet de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en Collections.



Pour les tests de Token générer par notre fonction, nous avons utilisé **JWT.io**

# DOSSIER PROFESSIONNEL (DP)

[Débogueur](#)[Bibliothèques](#)[introduction](#)[Interroger](#)[Obtenez un tee-shirt !](#)

## Encodé COLLER UN JETON ICI

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG91IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQsw5c
```

## Décodé MODIFIER LA CHARGE UTILE ET LE SECRET

ENTÊTE: ALGORITHME ET TYPE DE JETON

```
{  "alg": "HS256",  "typ": "JWT"}
```

CHARGE UTILE: LES DONNÉES

```
{  "sous": "1234567890",  "nom": "John Doe",  "iat": 1516239022}
```

Pour tester l’affichage de l’application, nous avons utilisé **Expo**.

## 2. Précisez les moyens utilisés :

Postman, JWT, Expo



POSTMAN



Expo



## 3. Avec qui avez-vous travaillé ?

J’ai conçu ce projet avec mes camarades Gregory et Olivier.

## 4. Contexte

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

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

Chantier, atelier, service ▶ *Conception développement d'application*

Période d'exercice ▶ Du : au :

## 5. Informations complémentaires *(facultatif)*

## Activité-type 3 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

**Exemple n° 6** ▶ Préparer et exécuter le déploiement d'une application

### 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)

### 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)*