



Travail Pratique Individuel (TPI)

Journal de bord

Centre de formation: CFPT Ecole d'informatique

cf. <https://edu.ge.ch/site/cfpt-informatique/>

Orientation du CFC : Développement d'applications

Volée : 2019/2020

Apprenti : Adrien Witzig

Maître d'apprentissage : Sandrine Weber

Experts : Pascal Court et Borys Folomietow

Consignes :

Généralités :

Le journal de travail sert à documenter les activités quotidiennes, les problèmes rencontrés ainsi que les éventuels travaux imprévus, aides et heures supplémentaires. Il est clairement structuré et se réfère au plan du projet.

Utilisation du journal de bord :

- Créer chaque jour une nouvelle section indiquant explicitement la date du jour.
- Commencer par énumérer les objectifs du jour.
- Se servir ensuite du carnet comme support de brouillon, s'en servir pour faire des schémas permettant de structurer sa pensée.

Si certains schémas ne sont pas aisément insérables directement dans ce document, le nom explicite du fichier peut être indiqué et le document correspondant mis en pièce jointe (dans l'espace partagé).

- Chaque commentaire doit comporter l'heure exacte à laquelle il a été renseigné (heures et minutes).
- Noter les bugs rencontrés, même s'ils sont traités immédiatement.
- En fin de journée, faire un petit bilan. Celui-ci peut être fait sur les objectifs énumérés en début de journée. Ne pas oublier de citer ce qui a été fait sans que cela n'ait été prévu comme objectif du matin.
- Le candidat est le responsable de la bonne tenue de ce document.

Date: lundi 25 mai 2020

8h : J'ai reçu l'énoncé par les experts. Je fixe mes objectifs d'aujourd'hui :

- Réaliser la planification prévisionnelle
- Crée la connexion
- Crée l'ajout de projets par un enseignant
- Crée un modèle (style) du site
-

Je réalise la maquette des pages :

- index.php → Contrôleur
- /views → Vues
- /model → Modèle

Je choisis directement des couleurs qui me plaisent sur <https://paletton.com/> pour la barre de navigation ainsi que la couleur de fond du site. Je crée une liste avec des éléments pour les liens de ma barre de navigation. Pour rendre le lien cliquable j'ai rajouté des balises <a>. A l'aide du CSS, j'ai pu aligner mes éléments avec la propriété inline. J'ai également changé la police du nom du site : Je suis passé par une variante de police → font-variant : small-caps. J'ai utilisé la fonctionnalité CSS :hover pour que la couleur d'un élément change si l'utilisateur passe la souris dessus.

9h : Entretien avec Mme Weber, Les identifiants de connexions seront gérés par l'administrateur, il n'y aura donc pas de connexion via EEL. Les jours de la semaine sont supprimés car les projets ne fonctionnent plus comme ça. Elle m'a aussi averti qu'il fallait faire un manuel utilisateur. De plus, elle m'a envoyé le modèle du journal de bord ainsi que de la planification. Nous avons fixés les rendez-vous tous les deux jours à 9h.

9h30 : Je choisis GitHub pour sauvegarder mon projet et être sûr de ne rien perdre. Je passe donc par le logiciel GitHub Desktop et comme mon éditeur de code est Atom, il y a la possibilité d'avoir un menu raccourci directement dans celui-ci.

J'ajoute la librairie Bootstrap pour faciliter le style de mon site. Cependant, j'ai fait la barre de navigation en CSS pur. J'utilise, à travers Bootstrap, le système de grille qui permet de positionner les éléments. J'ajoute également un container (classe Bootstrap) pour centrer les éléments.

Architecture du système de grille

```
<div class="container">
  <div class="row">
    <div class="col"></div>
    <div class="col"></div>
  </div>
</div>
```

J'ai ajouté dans la racine un dossier `db/` qui contient le script databaseConnection.php

qui s'occupe de la connexion pour la base de données.

J'ai créé le script [./model/logout.php](#) qui est appelé via le lien « Déconnexion » présent sur la barre de navigation. Dans le script, je détruis la session avec la fonction [session_destroy\(\)](#) de PHP, puis je redirige l'utilisateur sur la page [login.php](#).

10h : Je crée la page [projetGestion.php](#) qui sera la page de gestion pour les enseignants et la page [projetEleve.php](#) qui sera la page dédiée aux élèves. Pour afficher la bonne page par rapport au statut de l'utilisateur j'insère, dans le contrôleur, ce morceau de code :

```
if (isset($_SESSION["statut"])) {  
    switch ($_SESSION["statut"]) {  
        case 2:  
            include 'views/projetsGestion.php';  
            break;  
  
        case 3:  
            include 'views/projetsEleve.php';  
            break;  
    }  
}
```

1) Je vérifie le statut de l'utilisateur grâce à la variable de session.

2) J'affiche la page en fonction du statut (2 → Enseignant / 3 → Elève).

13h : J'ai créé le formulaire d'ajout de projets qui contient :

- Titre du projet
- Description du projet
- Nom du client
- Durée du projet (en semestre)
- Date de début du projet

Je réalise également les requêtes pour l'ajout d'un projet.

15h : J'ai créé la documentation et je l'ai complétée pendant 1h.

16h : J'ai créé le planning prévisionnel et il est difficile de savoir combien de temps chaque fonctionnalité met à être construite et dans quel ordre. J'y suis donc allé étape par étape et avec cette méthode j'ai beaucoup mieux réussi.

Date: mardi 26 mai 2020

8h : Avancement sur la documentation. Ensuite j'ai créé les fonctions pour récupérer les projets et les afficher dans la page de gestion de projets des enseignants. Pour l'instant, je les affiche dans un élément `.card` de `Bootstrap`. Cependant, je vais regarder pour les remplacer par un `.list-group` car je trouve cela plus pratique. J'ai modifié le style externe (`style.css`) et je garde donc les éléments `.card` de `Bootstrap` pour afficher les miniatures des projets, car j'ai trouvé le moyen d'afficher correctement les lignes (3 projets par ligne) des projets. Je n'affiche que les projets créés par l'enseignant.

10h :

J'avance sur la documentation : sur le rappel du cahier des charges et la bibliographie des sites déjà utilisés jusqu'à là.

11h : Je voudrais trouver un moyen d'afficher les projets détaillés en passant l'id du projet en POST au lieu de GET. Je change un peu le style du site, surtout les couleurs, que je choisis de <http://paletton.com/>.

13h : Je crée donc un formulaire qui contient un bouton de type `submit` et un champ `input` caché par le paramètre `hidden` qui contient l'id du projet. Lorsque l'on appuie sur le bouton « Aller à » sur la miniature d'un projet, cela envoie une requête qui récupère les informations du projet et les affiche dans la page de détail d'un projet. L'avantage de passer cette information en POST et non en GET est que l'utilisateur ne pourra pas modifier la variable via l'URL. Je crée donc la page de détail ou le détail du projet s'affiche et les élèves associés.

15h30 : Je rajoute un message d'erreur sur le login lorsque la connexion échoue. En effet, j'ajoute une variable de session lors de l'échec de connexion. Sur la page de connexion, je n'affiche le message que si la variable en question existe. Directement après l'avoir affiché, je supprime la variable avec la fonction `unset()` de PHP.

16h30 : Je prépare des questions pour Mme Weber :

- 1) Je voulais savoir si les critères devaient être réutilisables entre projets.
- 2) Est-ce que je dois réaliser le stockage par année des projets ? (Je ne me souviens plus de ce que vous m'aviez dit à ce sujet)
- 3) La présence hebdomadaire étant supprimée je voulais savoir si le champ Année de la table élève l'était aussi

Réponse de Mme Weber qui m'affirme que les critères doivent être réutilisables, que le stockage n'est pas nécessaire mais que l'année ne doit pas être supprimée.

Date: mercredi 27 mai 2020

8h : Je prépare en avance, la requête pour compter le nombre d'élèves par projet afin de ne pas dépasser 4.

```
SELECT COUNT(*) FROM utilisateurs LEFT JOIN travaille_pour
ON idUtilisateur = idEleve WHERE idProjet = 2
```

9h : Suite à des questions posées à Mme Weber, elle m'a confirmé que :

- Il pouvait y avoir plusieurs projets qui utilisent les mêmes critères

- Le stockage par année n'était pas à faire
- L'année dans la table élève ne devait être supprimée

Je continue la conception de la base de données et dessine un schéma de base de données.

10h : Point bloquant : Je n'arrive pas à comprendre où dois-je afficher les critères et comment sont-ils liés aux évaluations.

10h30 : Si l'enseignant essaye d'ajouter un élève alors qu'il y a déjà 4 élèves dans le groupe, j'affiche un message d'erreur. Je réalise ceci avec la requête ci-dessus. Je dois donc vérifier au moment d'ajouter un élève si le projet contient déjà le nombre maximal de 4.

11h30 : Résolution du point bloquant de 10h : Je vais créer une table criteres_projets qui indiquera quels critères sont pris en compte en fonctions des projets, c'est-à-dire, lorsque l'enseignant crée un projet il devra ajouter ou choisir des critères existant et il devra les attribuer à son projet .

13h : Je commence l'ajout d'un utilisateur à un projet. Premièrement, Je dois afficher la liste des élèves importée par l'administrateur en en omettant les élèves déjà ajouté au projet. Pour ceci j'écris une requête SQL :

Point bloquant : je n'arrive pas à écrire la requête car je dois choisir seulement les élèves qui ne sont pas encore dans le projet. J'ai regardé sur le site [w3school](http://w3school.com) différentes alternatives pour écrire cette requête. Tout d'abord, J'ai étudié RIGHT JOIN Mais celle-ci ne convient pas dans cette situation. Ensuite j'ai essayer avec UNION seulement le résultat était faux. Même chose pour EXCEPT / MINUS qui n'était pas adapté. Après 2 heures de recherches et tentatives J'ai réussi avec les sous-requêtes comme ceci :

```
SELECT * FROM utilisateurs WHERE Statut = 3 AND idUtilisateur NOT IN
(SELECT idUtilisateur FROM travaille_pour LEFT JOIN utilisateurs ON idUtilisateur = idEleve
WHERE idProjet = ?)
```

15h30 : Je continue donc avec l'affichage de des utilisateurs dans le détail de la page. Afin d'afficher également une croix rouge permettant de supprimer un élève du projet.

Date: jeudi 27 mai 2020

8h : Je commence la fonction de suppression d'un utilisateur au sein d'un projet. Je vais devoir à chaque utilisateur affiché crée un formulaire dynamique qui enverra en POST l'information de l'utilisateur à supprimer. Je crée donc le script delete.

10h30 : Point bloquant : J'ai supprimé un élève et j'en ai rajouté un autre pour vérifier ma requête et l'élève ne s'affiche plus donc je vais résoudre le problème.

11h : J'ai trouvé le problème, c'était que j'avais supprimé seulement l'utilisateur mais pas son id dans la table travaille_pour.

Maintenant je vais faire en sorte qu'on ne puisse ajouter que 4 élèves maximum à un projet.

12h : J'ai fini la vérification j'affiche un message d'erreur qui indique que le nombre

d'élèves maximum est atteint pour ce projet.

13h : Je rajoute sur l'affichage des élèves, un label qui indique si l'élève a été évalué ou non, pour ceci j'ai d'abord étudié les Badges de Bootstrap.

14h : Je fini de traduire le MCD en MLD et ayant un doute j'envoie un mail à Mme Weber en proposant 3 possibilités pour être sûr d'être sur la bonne voie :

1)

ENSEIGNANTS(idEnseignant, Nom, Prenom, Email, MotDePasse, Statut)
ELEVES(idEleve, Nom, Prenom, Email, MotDePasse, Statut, Annee)

2)

UTILISATEURS(idUtilisateur, Nom, Prenom, Email, MotDePasse, Statut)
ELEVES(idEleve, Annee, #idUtilisateur)

3)

UTILISATEURS(idUtilisateur, Nom, Prenom, Email, MotDePasse, Statut, Annee)
Et laisser vide le champ année pour les enseignants.

15h15 : En attendant je complète la documentation en copiant le MCD dans la partie correspondante je complète aussi le MLD. Je complète également la partie description détaillée du projet dans la documentation.

16h : Rendez-vous avec les experts : Tout c'est bien passé, ils m'ont conseillé de se concentrer sur la documentation car je n'en n'avais pas encore fait assez à ce niveau-là.

Date: vendredi 28 mai 2020

8h : Je fais en sorte qu'un projet puisse être modifié ou supprimer par l'enseignant. Je note aussi un entretien à 13h30 par la demande de Mme Weber pour clarifier le MCD.

Lorsqu'un projet est supprimé, nous devons supprimer toutes les entrées possédant idProjet dans les tables suivantes :

- TRAVAILLE_POUR
- CRITERES_PROJETS
- EVALUATIONS

Et ensuite supprimer dans PROJETS.

Pour ce faire J'utilise une boucle qui exécute la même requête sur les 4 tables qui contiennent idProjet.

```
$tablesIdProjet = array("travaille_pour", "criteres_projets", "evaluations",  
"projets");  
foreach ($tablesIdProjet as $table) {  
    $query = $db->prepare("DELETE FROM $table WHERE `idProjet` = ? ");  
    $query->bindParam(1, $dataProjet["idProjet"]);  
    $query->execute();  
}
```

9h : Je commence la page « gestion de critères » ou l'enseignant pourra ajouter des

critères mais aussi consulté les critères existants.

10h : Je réalise la modification d'un projet par un enseignant. Lorsque l'enseignant appuie sur « Modifier », cela ouvre un modal (Bootstrap) qui affiche tous les champs avec leur valeur à ce moment-là. Lorsque l'utilisateur aura fini les changements, cela mettra à jour les champs de la base de données.

13h : Je crée et j'insère dans la documentation, 3 maquettes de page.

13h30 : Entretien vidéo avec Mme Weber où j'ai pu lui partager mon écran.

14h : Je transforme la base de données selon les directives de Mme Weber.

15 h : je crée la section critères ou je dois récupérer les critères depuis la table `criteres_projets`. Je les récupère et affiche par catégories de projet, ceci, étant géré par l'administrateur.

16 h : Lors de l'ajout d'un critère existant, je récupère, par catégorie, les critères disponible seulement.

Date: lundi 1er juin 2020: férié.

Date: mardi 2 juin 2020

8h : je réalise la page d'évaluation d'un élève dans un projet, elle s'appellera projetEvaluation.php et ne sera accessible qu'au enseignants.

Cette page inclut les critères et pour chaque critère il faut remplir le nombre de points obtenus et un commentaire/observation du critère. Ensuite l'enseignant valide l'évaluation et la note apparaît dans la gestion de projets.

10h : Je commence la partie Administrateur, je rappelle que la page est affichée selon le statut donc je n'ai rien à faire de ce côté. Je crée des tableaux où l'administrateur pourra Voir/modifier/supprimer des projets, utilisateurs, catégories de critères et des domaines de projets

13h : La fonction getNote() retourne la note avec ce calcul :

- $\text{points obtenus} / \text{points total} \times 4.5 + 1.5$

```
'SELECT SUM(pointsObtenus) as currentPt, SUM(criteres.pointsMax) as totalPt FROM evaluations INNER JOIN criteres ON evaluations.idCritere = criteres.idCritere WHERE idEleve = ? AND idProjet = ?
```

14h : Le bouton « Voir évaluation » redirige vers la page voirEvaluation.php qui récupère les évaluations par critères et affiche la note finale.

Date: mercredi 3 juin 2020

8h : Je règle la durée d'un projet sur soit ½ année ou 1 année, comme Mme Weber m'a conseillé.

8h 30 : Je réalise la création du rapport d'évaluation qui se fera en fichier .pdf

10h : J'utilise la librairie FPDF, qui permet avec ses fonctions Cell() et MultiCell() de créer un tableau qui me permet d'afficher correctement les différents critères dans le rapport d'évaluation.

Voilà une partie du code qui crée le document PDF :

```
if (isset($_POST["download"])) {
    $pdf = new FPDF();
    $pdf->SetAutoPageBreak( false);
    $pdf->AddPage();
    $pointsObtenus += $evaluation["pointsObtenus"];
    $pointsTotal += $evaluation["pointsMax"];

    foreach (getCategoriesCriteres() as $categorie) {
        if (!empty(getCriteresByCategories($categorie["idCategorie"], $idProjet))) {
            $pdf->SetFont('Arial','B',18);
            $pdf->Cell(180, 10, utf8_decode($categorie["categorie"]), '', 1, '');
            $pdf->Cell(180, 4, "", '', 1);
            foreach (getEvaluations($idEleve, $idProjet, $categorie["idCategorie"]) as $evaluation) {
                $pdf->SetFont('Arial','B',16);
                $pdf->Cell(180, 10, utf8_decode($evaluation["critere"]), 'LTRB', 1, 'C');
                $pdf->SetFont('Arial','',12);
                if ($pdf->GetStringWidth(utf8_decode($evaluation["definition"])) >= 180) {
                    $pdf->MultiCell(180, 10, utf8_decode($evaluation["definition"]), 'LR', 1);
                }
                else {
                    $pdf->Cell(180, 10, utf8_decode($evaluation["definition"]), 'LR', 1);
                }
                if ($pdf->GetStringWidth(utf8_decode($evaluation["observation"])) >= 180) {
                    $pdf->MultiCell(180, 10, "Commentaire: ".utf8_decode($evaluation["observation"]), 'LTR', 1);
                }
                else {
                    $pdf->Cell(180, 10, "Commentaire: ".utf8_decode($evaluation["observation"]), 'LTR', 1);
                }
                $pdf->SetFont('Arial','B',16);
                $pointsObtenus += $evaluation["pointsObtenus"];
                $pointsTotal += $evaluation["pointsMax"];
                $pdf->Cell(180, 13, "Points: ".$evaluation["pointsObtenus"]." / ".$evaluation["pointsMax"], 'LTBR', 1, 'R');
                $pdf->Cell(180, 10, "", '', 1);
            }
        }
    }
}
```

La fonction utf8_decode() permet d'afficher les accents.

13h : Je modifie la création de PDF en rajoutant le nom du projet, le nom de l'élève et la date. Je rajoute également un if() qui vérifie si le bas de la page a été atteint , si le bas de la page a été atteint, je rajoute une nouvelle page.

14h30 : Point bloquant, mes fenêtre modale ne s'affichent plus je vais donc vérifier les erreurs.

15h : Je crée le style de la gestion administrateur (catégories, domaines, utilisateurs et projets), avec les tableaux Bootstrap. Je poursuis la partie back-end¹

Date: jeudi 4 juin 2020

8h : J'ai résolu le point bloquant, il y avait 2 id portant le même nom.

9h : Je fais la « modification de projet » par un enseignant. J'affiche un modal où les

¹ Partie non visible de la programmation

informations du projet sont dans un <input> et on peut modifier les champs et enregistrer. En plus des requêtes UPDATE.

10h30 : Je résous des bugs d'affichage comme la modal qui ne s'ouvre plus ou la classe d'un utilisateur qui ne se modifie pas.

Date: vendredi 5 juin 2020

8h : Je continue la documentation technique car j'ai un peu de retard sur celle-ci.

13h : Je crée une grande partie des Tests, et les effectuent directement.

16h : je crée le manuel utilisateur et le remplis partiellement.

Date: lundi 8 juin 2020

8h : Je corrige certains bugs de messages d'erreur et je complète l'ajout de la classe et l'année dans l'ajout un utilisateur (administrateur).

9h : Je fini le Manuel utilisateur.

13 h : Après un entretien avec Mme Weber à 12h30, elle m'a conseillé de me dépêcher pour le résumé du TPI. Je le commence donc maintenant.

15h : J'ai fini le résumé, je vais donc imprimer le code source en PDF pour ne pas être encombré de ça demain.

Date: mardi 9 juin 2020

8h : Je termine la documentation, vérifie le nom de fichiers leur contenus. Comme Le résumé du TPI : Vérifier les fautes d'orthographe.

11 h : Je vérifie tous les documents une dernière fois et je vérifie surtout la documentation.