

Planning web



Sacha Grenier – Cin4b
ETML - Lausanne
Durée : 105 heures
Chef de projet : Gruaz Gilbert
Experts : Mottier André et Borboen Nicolas

Table des matières

1 SPÉCIFICATIONS.....	3
1.1 TITRE.....	3
1.2 DESCRIPTION	3
1.3 MATÉRIEL ET LOGICIELS À DISPOSITION.....	3
1.4 PRÉREQUIS.....	3
1.5 CAHIER DES CHARGES	3
2 PLANIFICATION INITIALE	4
3 ANALYSE	5
3.1 OPPORTUNITÉS	5
3.2 DOCUMENT D'ANALYSE ET CONCEPTION	5
3.2.1 <i>Page principale</i>	6
3.2.2 <i>La page Planning des utilisateurs</i>	8
3.2.3 <i>Page travailleurs</i>	8
3.2.4 <i>Base de données</i>	9
3.3 CONCEPTION DES TESTS	13
3.4 PLANIFICATION DÉTAILLÉE.....	13
4 DOSSIER DE RÉALISATION	15
4.1.1 <i>Mise en place de l'environnement</i>	15
4.1.2 <i>Fichier de migration</i>	17
4.1.3 <i>Travailleur</i>	18
4.1.4 <i>Menu de gestion de niveaux</i>	20
4.1.5 <i>Planning général</i>	23
4.1.6 <i>Planning des travailleurs</i>	28
4.2 MODIFICATIONS.....	29
4.2.1 <i>Modifications à faire</i>	29
4.2.2 <i>Modifications effectuées</i>	30
5 TESTS.....	31
5.1 DOSSIER DES TESTS	31
6 CONCLUSION.....	33
6.1 BILAN DES FONCTIONNALITÉS DEMANDÉES	33
6.2 BILAN DE LA PLANIFICATION	33
6.3 BILAN PERSONNEL.....	33
7 DIVERS.....	34
7.1 JOURNAL DE TRAVAIL.....	34
7.2 WEBOGRAPHIE	34
8 ANNEXES	35
8.1 CAHIER DE CHARGES.....	35
8.2 SCRIPT SQL OBTENU AVEC L'APPLICATION JMERISE.....	38
8.3 JOURNAL DE TRAVAIL.....	41

1 SPÉCIFICATIONS

1.1 Titre

Plateforme de planification hebdomadaire d'ateliers et de travailleurs.

1.2 Description

Inspiré d'une situation réelle vécue au GRAAP-F (fondation qui s'occupe de travailleurs AI), l'idée de ce projet est de réaliser une plateforme Web qui permet, pour un chef d'ateliers, de faire une planification hebdomadaire de plusieurs ateliers dans lesquels travaillent des employés au bénéfice de l'AI.

Une réalisation générique de ce projet permettra d'utiliser la plateforme pour planifier des classes avec des cours modulaires, ou des élèves inscrits dans des activités de camp de sport, etc.

1.3 Matériel et logiciels à disposition

1 ordinateur standard ETML, avec la structure habituelle.

A mettre en œuvre par le candidat :

- Une architecture WAMP (Xampp),
- Un framework PHP Laravel
- Un dépôt GIT

1.4 Prérequis

Avoir suivi les modules ICH à l'ETML, les projets et effectué des stages.

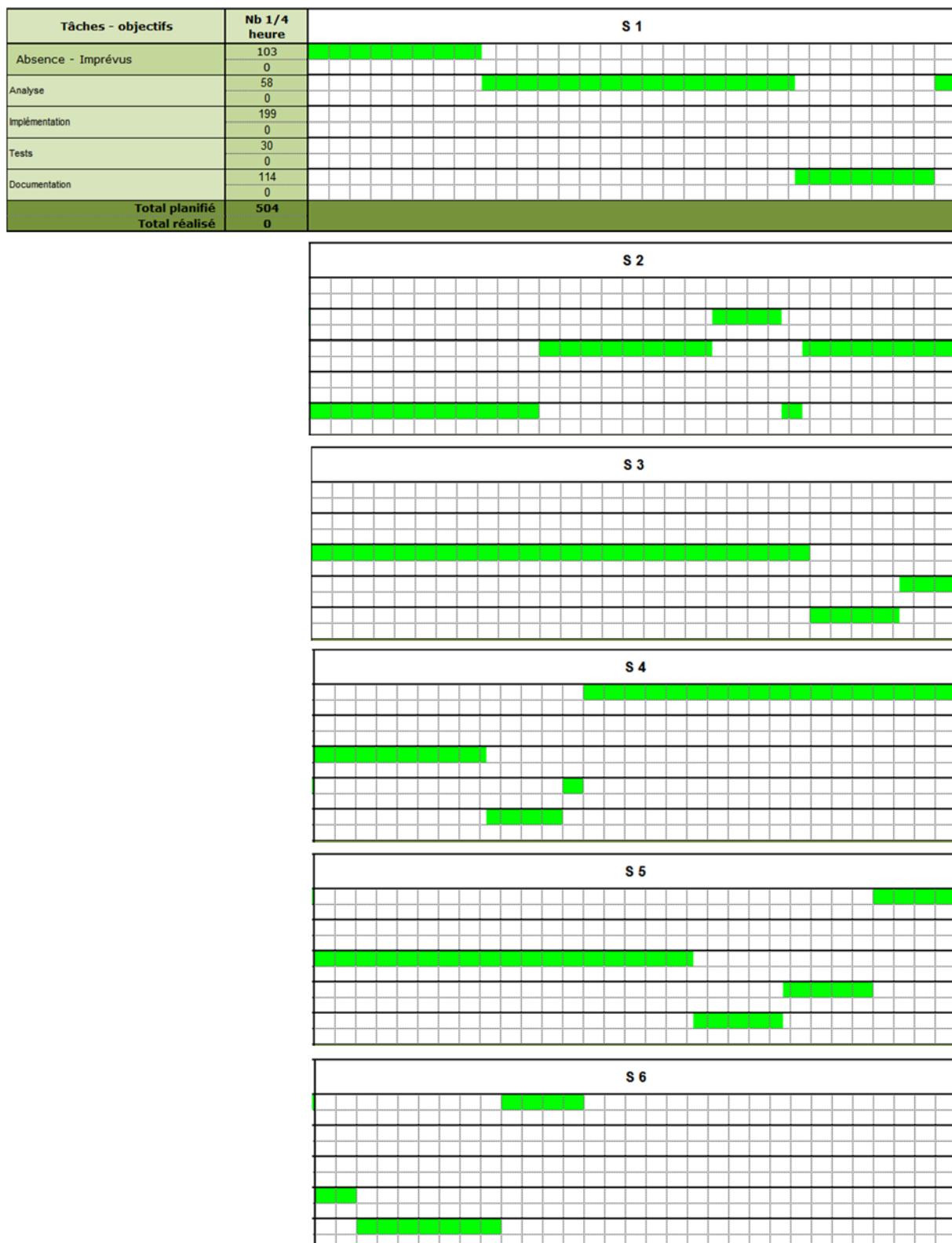
1.5 Cahier des charges

https://github.com/SachaGrenier/TPI/blob/master/TPI_files/H-TPI-greniersa-web-Cahier_de_charges.docx

Accessible au [chapitre 8.1](#)

2 PLANIFICATION INITIALE

Voici le Gantt que j'ai obtenu en faisant ma planification initiale :



3 ANALYSE

3.1 Opportunités

La création de ce site web va me permettre de :

- M'améliorer avec le framework Laravel
- M'améliorer en Javascript, jQuery et AJAX
- Apprendre comment créer un planning dynamique

Les difficultés seront :

- Le temps à disposition
- Créer la grille sans utiliser de librairie
- Site responsive

3.2 Document d'analyse et conception

Le but du site est de créer un planning générique qui pourra être utilisé dans plusieurs domaines. Pour me mettre en situation, j'ai choisi le domaine donné dans le cahier de charges qui est le GRAAP-F. L'objectif est donc d'attribuer des tâches à des utilisateurs et de les implémenter dans un planning sous forme de grille hebdomadaire. Il doit être possible d'attribuer des tâches à des utilisateurs lors de la semaine courante, mais aussi lors des semaines suivantes. Les semaines précédentes seront affichées mais ne pourront pas être modifiées, elles feront office « d'historique ».

En ce qui concerne les pages du site, j'ai créé des maquettes correspondant à chaque page du site.

3.2.1 Page principale

Cette page affichera le planning général, avec diverses possibilités d'ajout, suppression et modification de données.

The screenshot shows a weekly planning grid for workers across four levels (Niveau 1 to Niveau 4) over five days (Lundi to Vendredi). The grid is divided into morning (Matin) and afternoon (Après-midi) sessions. A red box highlights the top navigation bar with tabs: Planning général, Planning travailleurs, and Travailleurs. Numbered callouts point to specific features:

- 1**: Top right corner of the header.
- 2**: Navigation arrows for moving between weeks.
- 3**: Action buttons (+ and X) for managing rows.
- 4**: Column header for the Tuesday afternoon session.
- 5**: Example of a cell containing a dropdown menu for selecting users.
- 6**: Example of a cell containing a user name (Pierre_Dupont).
- 7**: Bottom right corner of the page.
- 8**: Bottom right corner of the header.

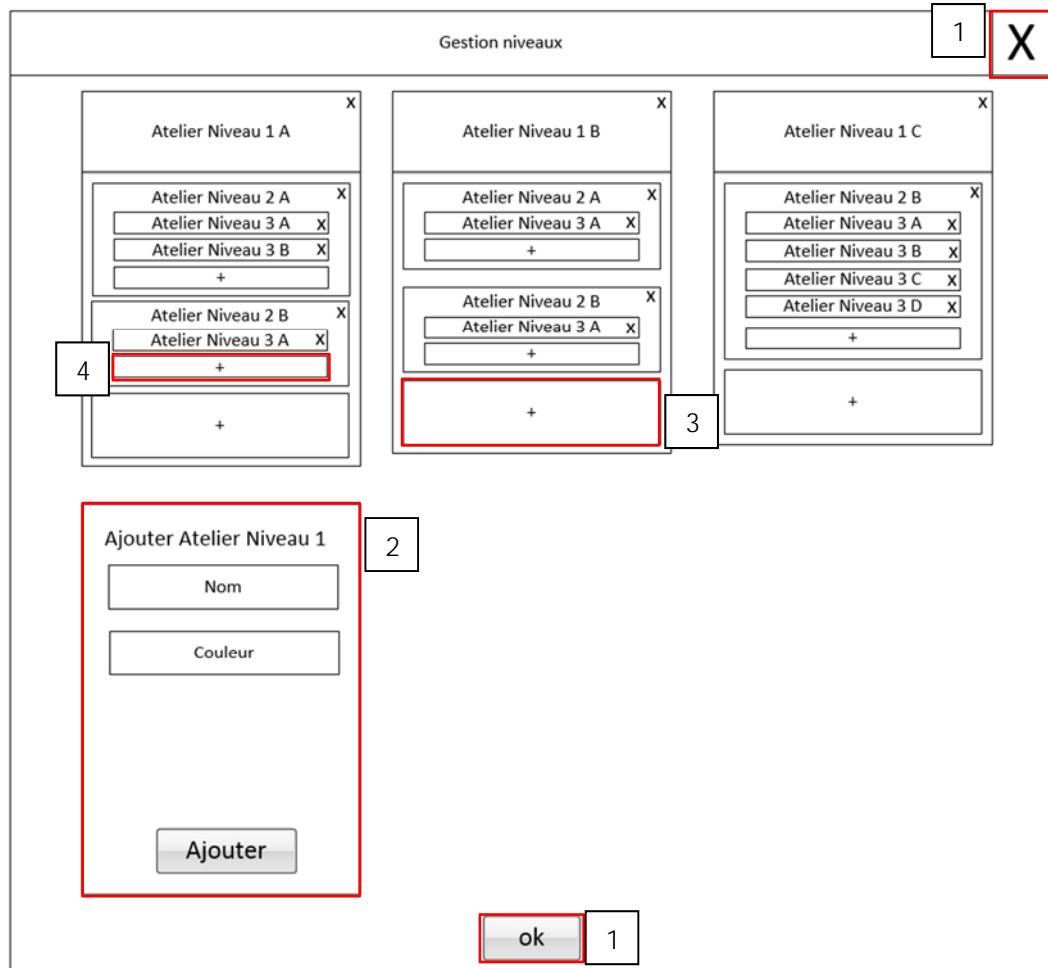
Description :

- 1) Menu d'entête, comportant les liens vers les 3 pages du site.
- 2) Barre de navigation indiquant la semaine actuellement affichée dans la grille, avec deux boutons en forme de triangle pour se déplacer entre les semaines.
- 3) Boutons + (ajouter) et x (supprimer) qui affecteront les lignes. Normalement ces boutons seront cachés et visibles uniquement lorsque l'utilisateur passe sa souris sur la ligne souhaitée. Le bouton + ajoutera une ligne en dessous et prendra exactement la même catégorie et la même tâche. Le bouton X supprimera la ligne du tableau, avec confirmation de l'utilisateur.
- 4) Colonne correspondant au Mardi, qui se sépare en Matin et Après-midi.
- 5) Les catégories et tâches qui en héritent seront affichées ici. Au moins une ligne par tâche sera toujours affichée.
- 6) Exemple de cellule, correspondant au mardi après-midi. Ici, l'utilisateur pourra cliquer pour afficher champ de recherche, qui une fois complété de quelques lettres proposera les différents utilisateurs correspondants (Autocomplete).

7) Bouton d'impression pour imprimer (ou enregistrer sous format PDF) la semaine affichée.

8) Bouton qui ouvrira un menu sous forme de pop-up, et qui servira à modifier les catégories et les tâches.

Menu :



Description :

- 1) Boutons pour fermer la pop-up.
- 2) Formulaire pour ajouter un atelier de niveau 1.
- 3) Bouton + pour ajouter un atelier de niveau 2. Une fois le bouton cliqué, un champ texte apparaîtra et appuyer sur la touche entrée ajoutera l'atelier.
- 4) Idem que pour le point 3) mais pour ajouter un atelier de niveau 3.

Peu importe les niveaux, tous les ateliers seront supprimables à l'aide de la croix qui se situe à droite de chaque atelier. Bien évidemment, si un atelier de niveau 2 est supprimé, tous les ateliers de niveau 3 associés le seront aussi.

3.2.2 La page Planning des utilisateurs

Cette page sera uniquement en lecture et rien de sera modifiable. Avec les boutons en forme de triangle, il sera possible de se déplacer à travers les semaines et d'y afficher le planning. Elle représente la même grille que pour la page principale mais sous une autre forme. Voici le modèle :

Description :

- 1) Barre de navigation indiquant la semaine actuellement affichée dans la grille, avec deux boutons en forme de triangle pour se déplacer entre les semaines.
 - 2)Colonnes indiquant le nom et le prénom des utilisateurs.
 - 3)Colonne MSP, indiquant le Maitre socioprofessionnel.
 - 4)Colonne %, indiquant le pourcentage effectif/prévu.
 - 5) Bouton d'impression pour imprimer (ou enregistrer sous format PDF) la semaine affichée

3.2.3 Page travailleurs

Cette page affichera la liste des travailleurs sous forme de tableau. Pour ce faire, je vais utiliser la librairie DataTables (<https://datatables.net/>) qui propose un tableau dynamique où il est possible de trier les données et qui utilise une pagination.

Planning général		Planning travailleurs	Travailleurs																																																																																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="8" style="text-align: right; padding-bottom: 5px;">Recherche</th></tr> <tr> <th>ID</th><th>Prénom</th><th>Nom</th><th>Nom d'utilisateur</th><th>MSP</th><th>%</th><th>Supprimer</th><th></th></tr> </thead> <tbody> <tr> <td>1</td><td>Pierre</td><td>Dupont</td><td>Pierre_Dupont</td><td>SP</td><td>90</td><td>X</td><td></td></tr> <tr> <td>2</td><td>Jean</td><td>Bolomey</td><td>Jean_Bolomey</td><td>SP</td><td>50</td><td>X</td><td></td></tr> <tr> <td></td><td></td><td></td><td>Prénom_Nom</td><td>Sélectionner ▾</td><td></td><td>Ajouter</td><td></td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>								Recherche								ID	Prénom	Nom	Nom d'utilisateur	MSP	%	Supprimer		1	Pierre	Dupont	Pierre_Dupont	SP	90	X		2	Jean	Bolomey	Jean_Bolomey	SP	50	X					Prénom_Nom	Sélectionner ▾		Ajouter																																					
Recherche																																																																																			
ID	Prénom	Nom	Nom d'utilisateur	MSP	%	Supprimer																																																																													
1	Pierre	Dupont	Pierre_Dupont	SP	90	X																																																																													
2	Jean	Bolomey	Jean_Bolomey	SP	50	X																																																																													
			Prénom_Nom	Sélectionner ▾		Ajouter																																																																													
1	Stéphane	Pottier	SP	X	X	X	X																																																																												
<input type="button" value="Prénom"/> <input type="button" value="Nom"/> <input type="button" value="Initiales"/> <input type="button" value="Ajouter"/>																																																																																			

Description :

- 1)Tableau dynamique comportant la liste des travailleurs générée par DataTables.
La dernière ligne est sous forme de formulaire pour ajouter un travailleur dans la liste.
- 2)Liste des Maitres socioprofessionnel, sous forme de tableau simple, sans utiliser DataTables.
- 3)Formulaire pour ajouter un MSP à la liste.

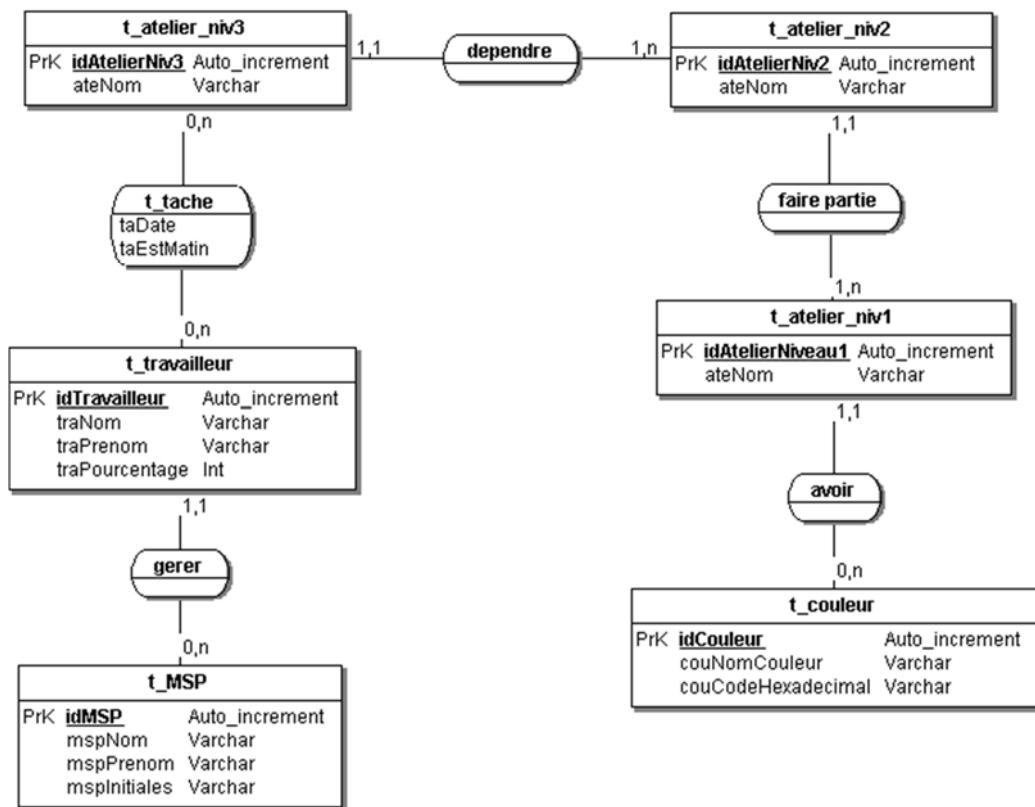
3.2.4 Base de données

Pour la base de données, je vais utiliser MySQL. Laravel fournit quelques outils pour aider à la construire, notamment un système de migrations que je vais utiliser pour créer mes tables. Le MCD, MLD et le script SQL suivants ont été réalisés à l'aide de l'application jMerise, que vous pouvez consulter ici :

<http://www.jfreesoft.com/JMerise/>. Cet outil permet de créer un MCD qui se transforme ensuite en MLD et en un MPD, sous forme de script SQL. J'ai donc dû nommer mes tables spécifiquement pour que le MLD construit par jMerise corresponde aux normes FTMI.

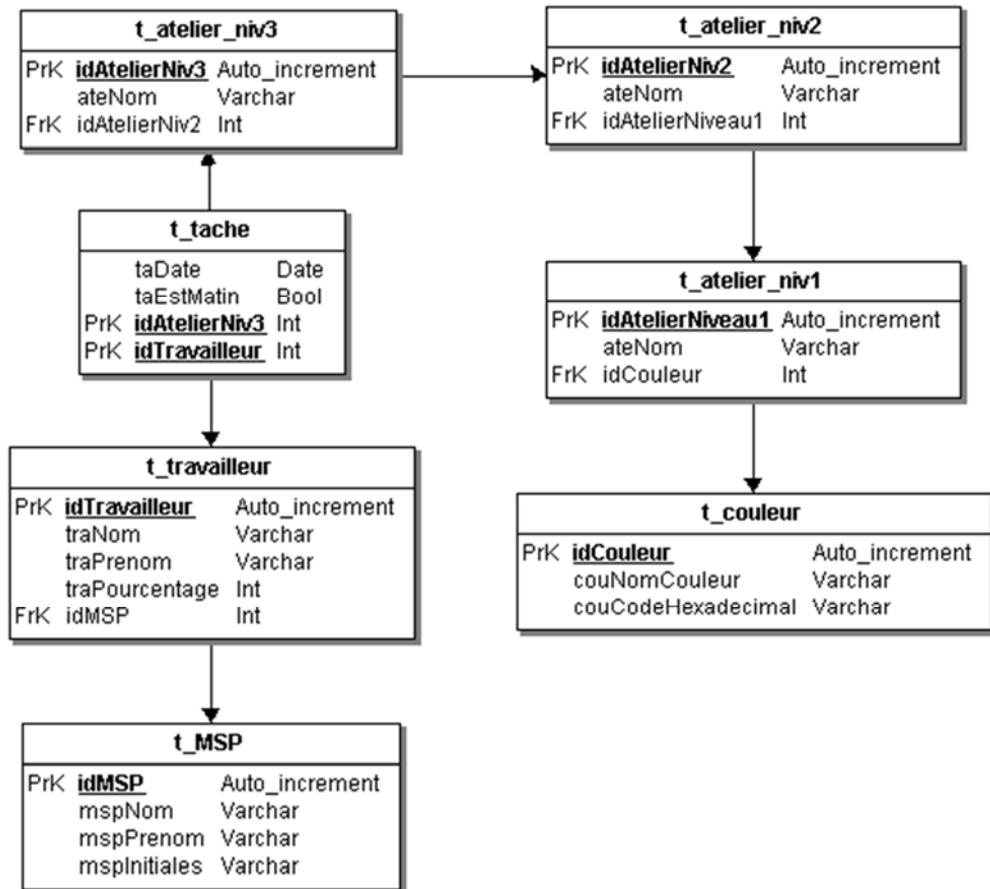
On y retrouve donc 5 tables basiques et une table intermédiaire.

Voici le MCD :



Etant donné que les cardinalités entre la table « t_atelier_niv3 » et « t_travailleur » sont de 0,N, j'ai donc prévu la table intermédiaire « t_tache ». Elle correspond à un atelier X assigné à un travailleur Y.

S'en suit le MLD :



Suite à ceci l'application jMerise m'a proposé un script SQL pour créer la base de données. Vous le trouverez dans les annexes au [chapitre 8.2](#), ou en cliquant ici : [Script SQL](#)

Et enfin le MPD, dans lequel j'ai modifié le nom des champs pour respecter les conventions du framework Laravel (snake case) et que vous pouvez retrouver ici : <https://laravel.com/docs/5.4/eloquent#eloquent-model-conventions>

MDP :

Table color					
Attribut	Type + Taille	Valeur par défaut	Contrainte	Index	NULL interdit
id	INT(10) (UNSIGNED)	Auto_increment	clé primaire	PRIMARY	Oui
name	VARCHAR (40)	-	-	-	Oui
hex	VARCHAR (40)	-	-	-	Oui
Table workshop_level_1					
Attribut	Type + Taille	Valeur par défaut	Contrainte	Index	NULL interdit
id	INT(10) (UNSIGNED)	Auto_increment	clé primaire	PRIMARY	Oui
name	VARCHAR (40)	-	-	-	Oui
color_id	int(10) (UNSIGNED)	1 (blanc)	clé étrangère	INDEX	Oui
Table workshop_level_2					
Attribut	Type + Taille	Valeur par défaut	Contrainte	Index	NULL interdit
id	INT(10) (UNSIGNED)	Auto_increment	clé primaire	PRIMARY	Oui
name	VARCHAR (40)	-	-	-	Oui
workshop_level_1_id	int(10) (UNSIGNED)	-	clé étrangère	INDEX	Oui
Table workshop_level_3					
Attribut	Type + Taille	Valeur par défaut	Contrainte	Index	NULL interdit
id	INT(10) (UNSIGNED)	Auto_increment	clé primaire	PRIMARY	Oui
lastname	VARCHAR (40)	-	-	-	Oui
firstname	VARCHAR (40)	-	-	-	Oui
username	VARCHAR (40)	-	-	-	Oui
percentage	INT(3)				Oui
MSP_id	int(10) (UNSIGNED)		clé étrangère	INDEX	Oui
created_at	DATETIME	date création	-	-	Oui
updated_at	DATETIME	date création	-	-	Oui
Table worker					
Attribut	Type + Taille	Valeur par défaut	Contrainte	Index	NULL interdit
id	INT(10) (UNSIGNED)	Auto_increment	clé primaire	PRIMARY	Oui
lastname	VARCHAR (40)	-	-	-	Oui
firstname	VARCHAR (40)	-	-	-	Oui
username	VARCHAR (40)	-	-	-	Oui
percentage	INT(3)				Oui
MSP_id	int(10) (UNSIGNED)		clé étrangère	INDEX	Oui
created_at	DATETIME	date création	-	-	Oui
updated_at	DATETIME	date création	-	-	Oui
Table task					
Attribut	Type + Taille	Valeur par défaut	Contrainte	Index	NULL interdit
id	INT(10) (UNSIGNED)	Auto_increment	clé primaire	PRIMARY	Oui
workshop_level_3_id	INT(10) (UNSIGNED)	-	clé étrangère	INDEX	Oui
worker_id	INT(10) (UNSIGNED)	-	clé étrangère	INDEX	Oui
isMorning	BOOLEAN	-	-	-	Oui
date	DATE	-	-	-	Oui
Table MSP					
Attribut	Type + Taille	Valeur par défaut	Contrainte	Index	NULL interdit
id	INT(10) (UNSIGNED)	Auto_increment	clé primaire	PRIMARY	Oui
firstname	VARCHAR (40)	-	-	-	Oui
lastname	VARCHAR (40)	-	-	-	Oui
initials	VARCHAR (4)	-	-	-	Oui

3.3 Conception des tests

Mes tests vont être effectués suivant une grille que je vais créer et qui aura les colonnes suivantes :

Fonctionnalité à tester	Résultat attendu	Date	Résultat obtenu	Solutions	Etat du bug	Temps passé à résoudre
-------------------------	------------------	------	-----------------	-----------	-------------	------------------------

Au fur et à mesure du projet, j'ajouterais des fonctionnalités à tester dans la grille.

3.4 Planification détaillée

Semaine		1	Explications: qu'est-ce qui se fait et comment ?
Tâche	Durée [1/4 h.]		
Absence - Imprévu	25		Rendez-vous chez le dentiste pour opération
Création de la planification initiale	7		
Analyse	25		Analyse du cahier de charges, recherches sur internet et analyse de la structure de la base de données (MCD MLD MPD)
Analyse	3		Recherches sur internet pour savoir quel librairie utiliser pour les tableaux ainsi que celle pour générer des PDF
Journal de travail	2		
Prise de connaissance du projet	3		Scéance avec l'expert
Création MCD, MLD et MPD	9		Création du MCD, MLD et MPD suite à ce qui a été analysé
Création des maquettes	9		Création de la page principale sous visio
Documentation	10		Implémentation de la partie analyse (maquettes, MCD MLD et MPD) dans le rapport de projet
Total semaine	93	Max. 93	
Semaine		2	Explications: qu'est-ce qui se fait et comment ?
Tâche	Durée [1/4 h.]		
Documentation	19		Implémentation de la partie analyse (maquettes, MCD MLD et MPD) dans le rapport de projet
Mise en place du dépôt GitHub	8		Création de la structure de dossier du projet et assignation au dépôt GitHub
Création des maquettes	12		Création des pages restantes sous visio
Création de la planification détaillée	9		
Création de migrations	4		Création du script pour insérer toutes les tables dans la base de données
Journal de travail	3		
Création du layout	16		Création du header
Développement du tableau de travailleurs	22		Développement des fonctions pour ajouter et supprimer des travailleurs dans la base de données. Mise en place du tableau DataTables et des formulaires.
Total semaine	93	Max. 93	

Semaine		3	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	
Documentation	4	Report du développement du tableau de travailleurs dans la documentation	
Création de la grille de test	3	Création de la grille de tests et ajout de tests sur les fonctions créées pour le tableau de travailleurs	
Développement du tableau de MSP	16	Développement des fonctions pour ajouter et supprimer des MSP dans la base de données. Mise en place des formulaires.	
Documentation	4	Report du développement du tableau de MSP dans la documentation	
Création de la grille de test	2	Ajout de tests sur les fonctions créées pour le tableau de MSP	
Développement du planning sur la page planning général	14	Mise en place du tableau sous forme HTML	
Développement du menu de gestion des niveaux	33	Ajout de menu sous forme de pop-up, développement des fonctions pour ajouter ou supprimer des ateliers et gestion des hiérarchies dans les niveaux	
Documentation	4	Report du développement du menu dans la documentation	
Création de la grille de test	2	Ajout de tests sur les fonctions créées pour le menu	
Journal de travail	3		
Tests	8	Tests des fonctionnalités en suivant la grille, corrections des erreurs	
Total semaine	93	Max. 93	
Semaine		4	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	
Développement du planning sur la page planning général	38	Développement de toutes les fonctionnalités nécessaires : autocomplete et possibilité de suppression sur chaque cellule, ajout et suppression de lignes, génération des semaines suivantes et affichage des semaines précédentes	
Journal de travail	1		
Absence - Imprévus	54	Pont de l'ascension	
Total semaine	93	Max. 93	
Semaine		5	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	
Développement du planning sur la page planning général	18	Développement de toutes les fonctionnalités nécessaires : autocomplete et possibilité de suppression sur chaque cellule, ajout et suppression de lignes, génération des semaines suivantes et affichage des semaines précédentes	
Documentation	20	Report du développement du planning dans la documentation	
Création de la grille de test	3	Ajout de tests sur les fonctions créées pour le planning	
Tests	7	Tests des fonctionnalités en suivant la grille, corrections des erreurs	
Journal de travail	3		
Absence - Imprévus	12	Examen d'ECG - Mercredi après-midi	
Développement du panning sur la page planning travailleurs	20	Affichage du planning sous une autre forme et gestion des semaines précédentes et suivantes	
Création de la grille de test	3	Ajout de tests sur les fonctions créées pour les plannings	
Tests	7	Tests des fonctionnalités en suivant la grille, corrections des erreurs	
Total semaine	93	Max. 93	
Semaine		6	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	
Documentation	20	Ecriture de l'abstract	
Journal de travail	2		
Absence - Imprévus	12	Lundi de pentecôte	
Documentation	5	Peaufinage de la documentation	
Total semaine	39	Max. 93	

4 DOSSIER DE RÉALISATION

4.1.1 Mise en place de l'environnement

Premièrement, j'ai installé une machine virtuelle avec VMware Workstation Player 12. J'y ai installé Windows 10, mais peu importe la version. Une fois que l'OS a été mis en place et que quelques outils y ont été installés (Notepad++, Sublime Text, Git Extensions), j'ai installé l'outil Composer qui permet notamment d'installer Laravel avec la commande suivante : « composer global require "laravel/installer" ».

Laravel à présent installé, j'ai exécuté la commande « laravel new tpi ». Cette commande crée un répertoire dans le dossier courant de la console. Pour mon cas, c'était dans « C:\xampp\htdocs\ ». Ce répertoire fait office de structure de base pour mon projet. Il a installé tout ce dont j'avais besoin pour commencer mon projet. Il faut juste modifier le fichier « .env » pour lui permettre d'accueillir la base de données.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_planning
DB_USERNAME=root
DB_PASSWORD=
```

En ce qui concerne le serveur apache et le serveur MySQL, un serveur virtuel XAMPP a fait l'affaire. Voici les différentes versions de mes outils :

Serveur web

- Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.6.28
- Version du client de base de données : libmysql - mysqld 5.0.11-dev - 20120503 - \$Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a \$
- Extension PHP : mysqli ⓘ
- Version de PHP : 5.6.28

J'ai modifié le fichier vHosts pour qu'il aille directement pointer sur mon projet.

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/tpi/public"
    ServerName localhost
</VirtualHost>
```

Lors de mon projet, j'ai remarqué que j'allais avoir besoin d'utiliser des formulaires. Laravel propose une librairie pour gérer les formulaires qui se trouve être très pratique. Pour l'installer, c'est très simple. Il faut se rendre à la racine du projet avec la console, puis executer la commande suivante :

composer require "laravelcollective/html":'^5.2.0"

Une fois le processus d'installation effectué par la commande, il faut modifier le fichier de configuration du projet pour y accueillir la nouvelle librairie.

Le fichier est le suivant :

<https://github.com/SachaGrenier/TPI/blob/master/config/app.php>

Ce qu'il faut ajouter dans le tableau « providers » :

Collective\Html\HtmlServiceProvider::class,

Ce qu'il faut ajouter dans le tableau « aliases » :

'Form' => Collective\Html\FormFacade::class,

'Html' => Collective\Html\HtmlFacade::class,

J'ai aussi du ajouter la librairie utilisée pour la génération des fichiers PDF. Au départ, j'ai installé wkhtmltopdf (<https://github.com/barryvdh/laravel-snappy>).

Mais suite à des tests, je me suis rendu compte que ce n'était pas la librairie qui convenait aux besoins du projet. Je l'ai donc désinstallé, puis ai trouvé une librairie plus adéquate. Il s'agit de domPDF. La marche à suivre pour l'installer est la même que pour laravelcollective html : <https://github.com/barryvdh/laravel-dompdf>

Documentation de la librairie HTML : <https://laravelcollective.com/docs/5.2/html>

Documentation officielle pour Installer Laravel : <https://laravel.com/docs/5.4>

Pour simplement importer mon projet :

Premièrement, il faut avoir XAMPP et se rendre dans le dossier htdocs pour cloner mon projet avec la commande suivante :

« git clone <https://github.com/SachaGrenier/TPI/> ». Ensuite, il faut modifier le fichier vHost (comme ci-dessus) pour pouvoir y accéder depuis le lien « <http://localhost> ». Enfin, il faut créer une base de données avec le nom : db_planning puis exécuter le script suivant :

https://github.com/SachaGrenier/TPI/blob/master/TPI_files/db_planning.sql

(peut être fait directement dans PHPmyadmin depuis l'onglet « importer »).

4.1.2 Fichier de migration

Ce fichier représente les fichiers MCD MLD et MPD que j'ai fournis précédemment. La base de données de Laravel fonctionne avec des migrations. C'est beaucoup plus pratique de modifier un élément de la base de données sans devoir tout supprimer. Pour créer un fichier de migration, il faut utiliser la console et inscrire la commande suivante :

```
C:\xampp\htdocs\tpi>php artisan make:migration create_all_tables
Created Migration: 2017_05_11_062800_create_all_tables
```

Dans ce cas-là, je donne déjà le nom de ma migration qui est « create _all_tables ». Une fois le fichier créé, il suffit de le remplir avec la tables et les champs souhaités. Ci-dessous un exemple de table dans une migration :

```
//color
Schema::create('color', function (Blueprint $table) {
    $table->increments('id');
    $table->string('name');
    $table->string('hex');
});
```

Lien vers le fichier de migration :

https://github.com/SachaGrenier/TPI/blob/master/database/migrations/2017_05_11_062800_create_all_tables.php

J'ai choisi de ne pas vous fournir le fichier de migration en annexe car le script SQL qui est déjà présent en soit fait l'affaire. Il est juste sous une autre forme, vous pouvez toujours y accéder avec le lien ci-dessus.

4.1.3 Travailleurs

Travailleurs							
Afficher 10 résultats <input type="button" value="Rechercher"/> <input type="button" value="Ajouter"/>							
ID	Prénom	Nom	Nom d'utilisateur	%	MSP	Crée le	Supprimer
2	Pierre	Bolomey	Pierre_Bolomey	90	Stéphane Pottier (SP)	29 mai 2017	<input type="button" value="X"/>
12	Jean	Pierre	Jean_Pierre	50	Joseph Dupont (JD)	11 mai 2017	<input type="button" value="X"/>
13	Albert	Dumoulin	Albert_Dumoulin	60	Joseph Dupont (JD)	12 mai 2017	<input type="button" value="X"/>
14	Jacques	Dupond	Jacques_Dupond	50	Stéphane Pottier (SP)	15 mai 2017	<input type="button" value="X"/>

5 résultats (Affiche de 1 à 5)

Cette page est séparée en deux tableaux : Le tableau des travailleurs et le tableau des Maîtres socioprofessionnels.

Pour ce qui est du tableau des travailleurs, j'ai utilisé la librairie JavaScript DataTables (<https://datatables.net/>).

Les données des travailleurs sont récupérées en AJAX à l'aide de la fonction « getWorkersArray ». Cette fonction procède ainsi :

- 1) Récupération de tous les travailleurs inscrits dans la base de données.
- 2) Création d'un tableau vide.
- 3) Utilisation d'une boucle qui parcourt chaque travailleur et qui :
 - a. Modifie le format de la date
 - b. Modifie la manière dont est exprimé le MSP
 - c. Propose un lien sous forme de bouton pour supprimer le travailleur

Une fois les objets modifiés, ils sont au fur et à mesure implémentés dans le tableau précédemment vide.

```

/**
 * Creates an array with the Workers Data and adds a form at the last line of the array
 */
public function getWorkersArray()
{
    $workers = Worker::all();

    setLocale(LC_TIME, config('app.locale'));

    $form = new \stdClass();

    $form->id = "";
    $form->firstname = "<input class='form-control' type='text' placeholder='Prénom' id='firstname'>";
    $form->lastname = "<input class='form-control' type='text' placeholder='Nom' id='lastname'>";
    $form->username = "<input class='form-control' type='text' id='username' placeholder='Nom d'utilisateur'";
    $form->percentage = "<input class='form-control percentage' type='text' placeholder='%' id='percentage'>";
    $form->MSP_initials = $this->getMSPSelection();
    $form->created_at = Carbon::now()->formatLocalized('%d %B %Y');
    $form->delete_link = '<p id="_token" style="display:none">' . csrf_token() . '</p><button class="btn btn-secondary middle-button" id="confirm">Ajouter</button>';

    $array[] = $form;

    foreach ($workers as $key => $value)
    {
        $worker = new \stdClass();
        $worker->id = $value->id;
        $worker->firstname = $value->firstname;
        $worker->lastname = $value->lastname;
        $worker->username = $value->username;
        $worker->percentage = $value->percentage;
        $worker->MSP_initials = ".$value->msp['firstname']."' . ".$value->msp['lastname'] . ".$value->msp['initials'] . ")";
        $worker->created_at = $value->created_at->formatLocalized('%d %B %Y');
        $worker->delete_link = "<button onclick='deleteRow(" . $value->id . ",this)' value=''. csrf_token() .'" class='btn btn-danger middle-button'>X</button>";

        $array[] = $worker;
    }
}

return $array;
}

```

Activate Window
Go to Settings to ac

J'ai ajouté une ligne à la fin du tableau qui fait office de formulaire pour ajouter un travailleur.

A la fin, le tableau est retourné et j'ai mappé DataTables pour qu'il puisse facilement mettre chaque case dans la bonne colonne. La valeur de « aTargets » définit à quelle colonne appartient la valeur de « mData ».

```

"aoColumnDefs": [
    {
        "aTargets": [0],
        "mData": "id",
    },
    {
        "aTargets": [1],
        "mData": "firstname"
    },
    {
        "aTargets": [2],
        "mData": "lastname"
    },
    {
        "aTargets": [3],
        "mData": "username"
    },
    {
        "aTargets": [4],
        "mData": "percentage"
    },
    {
        "aTargets": [5],
        "mData": "MSP_initials"
    },
    {
        "aTargets": [6],
        "mData": "created_at"
    },
    {
        "aTargets": [7],
        "mData": "delete_link"
    }
]);

```

La deuxième partie de la page est utilisée pour afficher le tableau des Maîtres socioprofessionnels.

Maîtres socioprofessionnels

Prénom Nom Initials					Créer
ID	Prénom	Nom	Initials	Supprimer	
1	Stéphane	Pottier	SP		
2	Joseph	Dupont	JD		
25	Jean	Robert	JR		
26	Bernard	Joubert	BJ		

Etant donné que ce tableau ne va pas beaucoup être utilisé, j'ai préféré utiliser un tableau simple, rempli en PHP et où les formulaires ne sont pas dynamiques.

Voici comment le tableau des Maîtres socioprofessionnels est affiché :

```
@foreach ($MSPs as $MSP)
    <tr>
        <td>{{ $MSP->id }}</td>
        <td>{{ $MSP->firstname }}</td>
        <td>{{ $MSP->lastname }}</td>
        <td>{{ $MSP->initials }}</td>
        <td>
            {{ Form::open(array('url' => 'deletemsp','method'=>'POST','class' => 'form-inline', 'onsubmit' => 'deleteMsp(this)')) }}
            {{ Form::hidden('msp_id',$MSP->id) }}
            {{ Form::submit('X',[ 'class' => 'btn btn-danger middle-button']) }}
            {{ Form::close() }}
        </td>
    </tr>
@endforeach
```

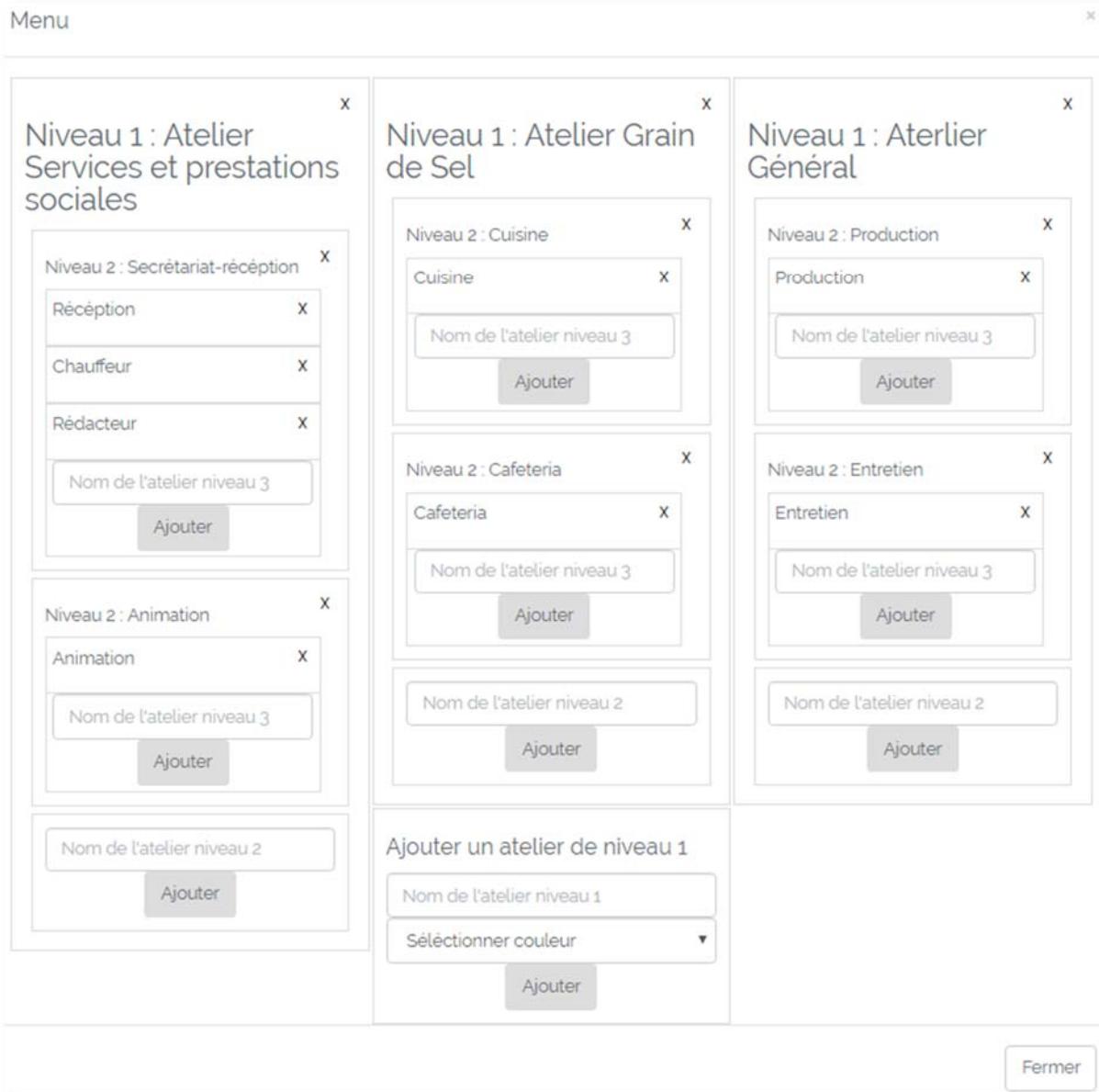
4.1.4 Menu de gestion de niveaux

Ce menu est disponible sous forme de pop-up suite au clic sur le bouton « Gestion niveaux » présent sur la page de planning. Il représente tous les niveaux de la base de données, présentés sous forme hiérarchique.

Le tableau est généré en PHP et HTML mais est récupéré à l'aide d'AJAX. Etant donné que les formulaires sont aussi envoyés en AJAX, cela permet à l'utilisateur d'effectuer plusieurs actions sur son menu sans devoir rafraîchir la page à chaque fois et relancer le menu en appuyant sur le bouton qui permet son affichage.

Chaque atelier, peu importe son niveau, peut être supprimé à l'aide du bouton sous forme de croix. Si l'utilisateur souhaite supprimer un atelier de niveau 2 alors que plusieurs ateliers de niveau 3 y sont attachés, une erreur intervient indiquant

que l'opération est impossible. L'utilisateur doit alors d'abord supprimer les ateliers de niveau 3 pour réaliser son action.



Au premier abord ce menu semble un peu confus. Voici comment il faut le comprendre :

Niveau 1 : Ce sont les casiers principaux. Le titre est en grand et à l'intérieur se situent les ateliers concernant les niveaux plus élevés. Le dernier casier est constitué d'un formulaire pour ajouter un atelier de niveau 1. Dans le formulaire, on peut définir un nom d'atelier ainsi qu'une couleur.

Note : La couleur sélectionnée pour l'atelier de niveau 1 sera affichée par la suite

Niveau 2 : Ces casiers se situent directement dans un casier de niveau 1. Dans chaque casier de niveau 1, un formulaire simple avec juste un champ texte est à disposition pour ajouter un atelier de niveau 2.

Niveau 3 : Les casiers de niveau 3 représentent directement les ateliers. Ils sont à l'intérieur des casiers de niveau 2. Comme pour le niveau 2, un formulaire simple est à disposition dans chaque casier de niveau 2 pour ajouter un atelier.

4.1.5 Planning général

Voici le planning général en l'état actuel. Les logins des travailleurs sont inscrits dans les cellules pour représenter leur présence à un atelier.

< Semaine précédente			Semaine du 29 au 02 juin 2017										Semaine suivante >			Gestion niveaux
Niv 1	Niv 2	Niv 3	Lundi 29		Mardi 30		Mercredi 31		Jeudi 01		Vendredi 02		Action			
			Matin	Après-midi	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi				
Atelier Services et prestations sociales	Secrétariat-réception	Récéption	Pierre_Bolomey	Jean_Pierre	Albert_Dumoulin	Jean_Pierre	Pierre_Bolomey	Jean_Pierre			Albert_Dumoulin		Pierre_Bolomey	+ X		
		Chauffeur	Jean_Pierre	Pierre_Bolomey										+ X		
		Rédacteur	Albert_Dumoulin	Albert_Dumoulin		Jacques_Dupond								+ X		
		Animation	Animation	Jacques_Dupond	Jacques_Dupond									+ X		
Atelier Grain de Sel	Cuisine	Cuisine			Jacques_Dupond		Jean_Pierre							+ X		
	Cafeteria	Cafeteria			Jean_Pierre	Pierre_Bolomey								+ X		
Atelier Général	Production	Production			Pierre_Bolomey								Albert_Dumoulin	+ X		
	Entretien	Entretien							Pierre_Bolomey		Jacques_Dupond			+ X		

Le tableau est généré en html et PHP, mais une grande partie des fonctions sont en JavaScript (jQuery) et Ajax. Il m'a paru important d'avoir un outil dynamique mais cela a rendu la tâche plus compliquée.

La base du tableau est générée en PHP et HTML. Pour ce faire, j'ai combiné 3 foreach (un par niveau). Dans chaque boucle, je récupère la liste des niveaux et avec ceci je crée ma ligne. Au final, le tableau apparaît. La difficulté de cette partie du code est que j'ai dû prendre en compte les rowspan. Ceux-ci sont utilisés pour faire en sorte qu'une cellule prenne plus de place sur la hauteur, ce qui fait que la cellule du dessous ne doit pas exister. J'ai utilisé les rowspan pour le niveau 1 et 2.

Les cellules du tableau sont toutes générées de la même manière, c'est-à-dire avec des métadonnées mais toutes vides:

```
$date = Carbon::parse($week["days"]["tuesday"])->formatLocalized('%Y-%m-%d');
echo '<td ' . $color . ' ondblclick="showForm(this)" data-ismorning="1" data-date="' . $date . '
    " data-workshop_id="'.$level_3->id.'" class="cell"></td>';
```

\$date -> récupère la date du tableau \$week, généré lors de la création de la vue et la formate.

\$color -> couleur définie au début de la boucle foreach, utilisée dans toutes les cellules de la même ligne.

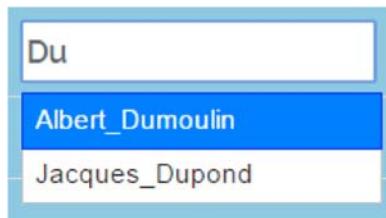
Ondblclick -> Evènement javascript qui s'active lorsque la case est double cliquée par l'utilisateur. La fonction showForm() permet d'y afficher le champ pour ajouter un travailleur à la tâche.

Les métadonnées sont : ismorning,date et workshop_id.

Pour les remplir avec les noms d'utilisateurs des travailleurs, j'ai créé une fonction qui récupère les travailleurs assignés à des tâches et qui les inscrit dans les cases, grâce aux métadonnées :

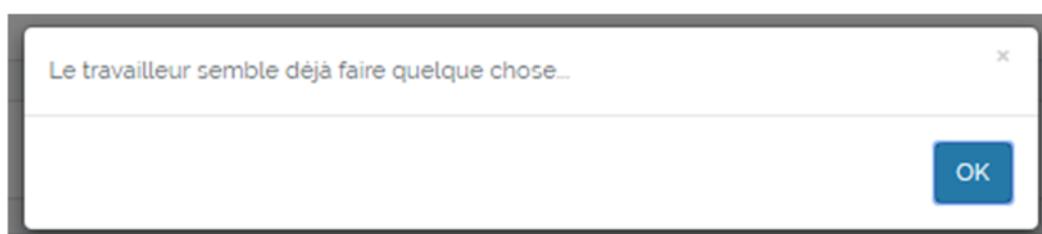
```
function initializeTable()
{
    $.ajax({
        url: '/getplanningcells',
        type: 'GET',
        success: function (data)
        {
            data = JSON.parse(data);
            for (var i = data.length - 1; i >= 0; i--)
            {
                $('[data-date="'+ data[i].date +'"][data-ismorning="'+ data[i].isMorning +'"]')[data-workshop_id="'+ data[i].workshop_level_3 +'"]').text(data[i].text);
            }
        }
    });
}
```

Pour assigner un travailleur à une tâche, un double clic sur une case fera apparaître un champ texte qui est en fait un « autocomplete ». Cela signifie qu'à partir du moment où certaines lettres que l'utilisateur a rentrées correspondent au prénom ou au nom d'un travailleur, il est proposé dans une liste :



Pour valider la sélection, un simple clic sur le choix du travailleur fait office d'envoi de requête. A partir de ce moment, la fonction qui enregistre le travailleur dans l'atelier vérifie d'abord que le travailleur ne soit pas occupé à une autre tâche à ce moment là.

Dans ce cas là, un message d'erreur apparaît pour indiquer que l'utilisateur fait fausse route :



Si la requête est acceptée, le tableau se met à jour avec les nouvelles valeurs et le travailleur apparaît alors dans la table :

Matin
Albert_Dumoulin

Il est aussi possible de supprimer un travailleur assigné à un atelier. Pour ce faire, lorsque l'utilisateur passe sa souris sur les cellules, une croix apparaît à côté des logins lors que ceux-ci sont inscrits :

Après-midi	Matin	Après-midi
Jean_Pierre	Jean_Pierre X	Jean_Pierre
Jean_Pierre	Jean_Pierre	

Comme dit précédemment, les interactions avec la base de données sont effectuées à l'aide de AJAX. Voici le modèle de base que j'utilise pour effectuer mes requêtes :

```
// Variable to hold request
var request;

// Disable the button while the ajax operation
$(button).prop("disabled", true);

// Fire off the request to /link
request = $.ajax({
    url: '/link',
    type: 'POST', //could be "GET"
    data: {label1: data1, label2: data2}
});

// Callback handler that will be called on success
request.done(function () {
    console.log("everything went good");
});

// Callback handler that will be called on failure
request.fail(function (jqXHR, textStatus, errorThrown){
    // Log the error to the console
    console.error(
        "The following error occurred: "+
        textStatus, errorThrown
    );
});

// Callback handler that will be called regardless
// if the request failed or succeeded
request.always(function () {
    // Reenable the inputs
    $(button).prop("disabled", false);
});
```

Voici pour exemple de fonction qui traite une requête AJAX la fonction que j'ai utilisée pour supprimer une tâche de la base de données :

```
/**
 * Removes worker in workshop from ajax array
 */
@return response code (with message if the response is negative)
*/
public function RemoveWorkerAtWorkshop()
{
    if(Request::ajax())
    {
        // Getting all post data
        $data = Input::all();
        try
        {
            $worker = Worker::where('username',$data["worker_username"])->get();

            $worker[0]->workshop_level_3()->wherePivot('date', '=', $data["date"])->wherePivot('isMorning', '=', $data["ismorning"])
                ]->detach($data["workshop_level_3"]);
            return response(200);
        }
        catch(\Exception $er)
        {
            return response(" Une erreur est intervenue" ,400);
        }
    }
    else
        return response($default_general_error_message,500);
}
```

Ici, la fonction vérifie d'abord que des données envoyées ont bien été récupérées.

Ensuite, les données sont enregistrées dans le tableau « \$data », où les valeurs sont accessibles à l'aide de leurs clés. Toutes les interactions sont dans un « try catch », pour pouvoir fournir une erreur à l'utilisateur si les données qui ont été envoyées étaient incorrectes.

Dans tous les cas, une réponse http est envoyée à l'utilisateur suite à la requête. A savoir que si la réponse est positive (code 200), il n'aura pas de notification contrairement à une réponse négative (code 400 ou 500).

Le bouton tout en bas à gauche du planning permet de transformer le planning actuellement affiché sous forme de PDF. Lors de clic, le fichier PDF est généré et directement affiché dans le navigateur. Evidemment, si ce dernier ne peut pas lire les fichiers PDF, le fichier est téléchargé.

Bugs connus :

Lors du développement de mon projet j'ai remarqué quelques bugs. Voici une liste qui normalement est exhaustive :

- Responsivité ; il semble que le planning sous sa forme actuelle ne soit pas responsive. Il reste utilisable avec un petit format. Je n'ai pas pu tester s'il fonctionnait depuis un smartphone.

J'estime 3-4 jours supplémentaires nécessaires pour que le planning soit parfaitement responsive et utilisable sur smartphone (si ce n'est pas déjà le cas).

- Ajout-suppression de lignes ; En soit, l'ajout et la suppression des lignes ajoutées fonctionnent. Le problème est que à partir du moment où l'utilisateur souhaite utiliser la ligne ajoutée en y insérant un travailleur, la ligne du dessus se fait overwrite.

J'estime une semaine pour :

- o Faire en sorte que le travailleur inséré n'overwrite pas.
 - o Le tableau puisse être généré avec les lignes qui ont été ajoutées.
 - o Les lignes supprimables qui se font supprimées suppriment les tâches de la base de données.
- Historique modifiable ; C'était une fonction que je souhaitais implémenter. Les semaines précédentes de la semaine actuelles n'étaient pas sensées pouvoir être modifiées. Actuellement, le planning est modifiable peu importe la semaine.

J'estime 2 jours pour que le planning s'affiche en mode historique.

4.1.6 Planning des travailleurs

Le planning des travailleurs se présente sous cette forme. Il est possible de parcourir les semaines pour y afficher le planning. Rien n'est modifiable de ce côté.

					Lundi 29		Mardi 30		Mercredi 31		Jeudi 01		Vendredi 02	
ID	Nom	Prénom	MSP	%	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi
2	Bolomey	Pierre	SP	90	Récéption	Chauffeur	Production	Cafeteria	Récéption		Entretien			Récéption
12	Pierre	Jean	JD	50	Récéption	Chauffeur	Récéption	Cafeteria	Récéption	Cuisine				Récéption
13	Dumoulin	Albert	JD	60	Rédacteur	Rédacteur	Récéption					Récéption		Production
14	Dupond	Jacques	SP	50	Animation	Animation	Rédacteur	Cuisine					Entretien	

[PDF](#)

La fonction ci-dessous est utilisée pour récupérer le tableau de cellules des ateliers par travailleur, avec la couleur de l'atelier en fond.

Le paramètre \$worker_id sert à récupérer les ateliers effectués par le travailleur en utilisant son ID et \$dates à fournir la plage de dates du planning. Le foreach parcourt les dates et insère à chaque fois deux cellules, pleines ou vides dans le tableau. La difficulté ici a été de gérer si la tâche était durant le matin ou l'après-midi et d'y insérer une case vide au bon endroit (switch).

```
static public function getWorkerWorkshops($worker_id,$dates)
{
    $worker = Worker::find($worker_id);

    $array = array();

    $style = 'style="background-color: ';
    $style_end = "'";

    foreach ($dates as $date)
    {
        $tasks = $worker->workshop_level_3()->wherePivot('date', '=', $date)->get();

        switch (count($tasks))
        {
            case 0:
                array_push($array, "<td></td>");
                array_push($array, "<td></td>");
                continue;

            case 2:
                $color0 = $style.$tasks[0]->workshop_level_2->workshop_level_1->color->hex.$style_end;
                $color1 = $style.$tasks[1]->workshop_level_2->workshop_level_1->color->hex.$style_end;
                $array[] = '<td '.$color0.'>'.$tasks[0]->name.'</td>';
                $array[] = '<td '.$color1.'>'.$tasks[1]->name.'</td>';
                continue;

            case 1:
                if ($tasks[0]->pivot->isMorning)
                {
                    $color0 = $style.$tasks[0]->workshop_level_2->workshop_level_1->color->hex.$style_end;
                    $array[] = '<td '.$color0.'>'.$tasks[0]->name.'</td>';
                    array_push($array, "<td></td>");
                }
                else
                {
                    $color0 = $style.$tasks[0]->workshop_level_2->workshop_level_1->color->hex.$style_end;
                    array_push($array, "<td></td>");
                    $array[] = '<td '.$color0.'>'.$tasks[0]->name.'</td>';
                }
                break;
        }
    }
    return $array;
}
```

4.2 Modifications

4.2.1 Modifications à faire

- Afficher la couleur de l'atelier de niveau 1 dans le menu de niveaux (15 min).
- Dans le planning, mettre en place un autofocus sur le champs texte qui apparaît lors du double clic sur une cellule (15 min).

- Dans le planning, faire en sorte que la croix qui apparaît quand l'utilisateur passe sa souris sur une cellule ne décale pas tout le planning (2h).
- Lors de la génération de planning sous format PDF, masquer la colonne action (15 min).
- Modifier le style général des fichiers PDF générés (1h).
- Calculer le pourcentage effectif sur le planning des travailleurs (1h).

4.2.2 Modifications effectuées

- Dans le tableau des travailleurs, j'ai créé une fonction qui vérifie si le nom d'utilisateur du travailleur est libre avant de l'insérer dans la base de données (5 min)
- Dans le formulaire pour ajouter un travailleur, j'ai limité entre 1 et 100 % le champ de pourcentage. Une fois la requête envoyée, si la valeur donnée ne correspond pas (ex 400) renvoie une erreur "Veuillez vérifier le pourcentage" (5 min).

5 TESTS

5.1 Dossier des tests

Les tests ont été effectués en suivant la grille suivante :

Fonctionnalité à tester	Résultat attendu	date	Résultat obtenu	Solutions	Etat	Temps Passé
Page travailleurs, tableau des travailleurs						
Insérer un travailleur dans la base de données à l'aide du formulaire	Le travailleur est ajouté dans la base de données, le tableau s'est bien mis à jour avec le nouveau travailleur	18.mai	Correspond			
Insérer un travailleur avec un nom d'utilisateur identiques à celles d'un autre travailleur	Le page se rafraîchit et une erreur intervient indiquant que les initiales choisies sont déjà utilisées	18.mai	BUG : La requête est tout de même acceptée	Vérifier si le nom d'utilisateur est libre avant de l'insérer dans la base de données	Réparé	5 min
Insérer un travailleur sans indiquer de Maître Sociaux Professionnel	Une erreur intervient indiquant que le formulaire n'est pas complet	18.mai	Correspond, avec l'erreur suivante: "Veuillez vérifier que tous les champs aient bien été remplis correctement"			
Insérer un travailleur et laisser les champs vide	Une erreur intervient indiquant que le formulaire n'est pas complet	18.mai	Correspond, avec l'erreur suivante: "Veuillez vérifier que tous les champs aient bien été remplis correctement"			
Insérer un travailleur en indiquant le pourcentage sans utiliser de chiffres valables (exemple : vingt, 200)	Une erreur intervient indiquant que le formulaire n'est pas valable	18.mai	Correspond, avec l'erreur suivante: "Veuillez vérifier que tous les champs aient bien été remplis correctement" BUG : les chiffres trop grands ou trop petits sont acceptés	Limiter entre 1 et 100 %, et si la valeur donnée ne correspond pas(ex 400) renvoie une erreur "Veuillez vérifier le pourcentage"	Réparé	5 min
Supprimer un travailleur	Le travailleur est supprimé de la base de données, le tableau s'est bien mis à jour	18.mai	Correspond			
Trier les travailleur par tous les tri possibles (id, prénom, nom, nom d'utilisateur, %,MSP,date de création)	Les travailleurs sont tous triés correctement dans la table	18.mai	Correspond			
Changer de page pour afficher d'autres travailleurs	La pagination fonctionne et ce qui est affiché dans le tableau correspond	18.mai	Correspond			
Page travailleurs, tableau des Maîtres socioprofessionnels						
Insérer un Maître socioprofessionnels dans la base de données à l'aide du formulaire	Le page se rafraîchit et le MSP apparaît dans le tableau	18.mai	Correspond			
Insérer un MSP sans indiquer d'initiales	Le page se rafraîchit et une erreur intervient indiquant que le formulaire n'est pas complet	18.mai	Correspond			
Insérer un MSP avec des initiales identiques à celles d'un autre MSP	Le page se rafraîchit et une erreur intervient indiquant que les initiales choisies sont déjà utilisées	18.mai	BUG : La requête est tout de même acceptée	Vérifier si les initiales sont libre avant de l'insérer dans la base de	Réparé	5 min
Insérer un MSP et laisser les champs vide	Le page se rafraîchit et une erreur intervient indiquant que le formulaire n'est pas complet	18.mai	Correspond			
Supprimer un MSP	Le page se rafraîchit et le MSP n'apparaît plus dans le tableau	18.mai	Correspond			

Page planning, menu de gestion de niveaux d'atelier				
La pop-up pour gérer les niveaux s'affiche lorsque le bouton gestion niveaux est cliqué	Le menu s'affiche correctement sous forme de pop-up	18.mai	Correspond	Amélioration possible: afficher les couleurs des niveaux 1
Insérer un niveau 1 à l'aide du formulaire	Le niveau 1 est inséré et le menu se met à jour	18.mai	Correspond	
Insérer un niveau 1 mais ne pas indiquer de couleur	La requête est refusée avec un erreur indiquant que le formulaire n'est pas complet	18.mai	Correspond	
Insérer un niveau 2 à l'aide du formulaire	Le niveau 2 est inséré dans le niveau 1 correspondant et le menu se met à jour	18.mai	Correspond	
Insérer un niveau 3 à l'aide du formulaire	Le niveau 3 est inséré dans le niveau 2 correspondant et le menu se met à jour	18.mai	Correspond	
Supprimer un niveau 1 à l'aide du bouton	Le niveau 1 est supprimé et le menu se met à jour	18.mai	Correspond	
Supprimer un niveau 2 à l'aide du bouton	Le niveau 2 est supprimé du niveau 1 correspondant et le menu se met à jour	18.mai	Correspond	
Supprimer un niveau 3 à l'aide du bouton	Le niveau 3 est supprimé du niveau 2 correspondant et le menu se met à jour	18.mai	Correspond	
Insérer un niveau 2 ou 3 sans indiquer de nom d'atelier	La requête est refusée avec un erreur indiquant que le formulaire n'est pas complet	18.mai	Correspond, mais avec un BUG : Le fait de superposer 2 popup fige le menu, qui en soit n'est pas si problématique.	

Page planning, planning				
Affichage du planning	Le planning apparaît à la semaine en cours et affiche correctement la semaine en cours. Les couleurs correspondent aux couleurs des ateliers définies	02.juin	Correspond, mais les semaines précédentes sont tout de même modifiables	
Click sur le bouton "Semaine précédente" ou "Semaine suivante"	Déplace le planning à la semaine souhaitée (précédente ou suivante) et met à jour la semaine affichée	02.juin	Correspond	
Double cliquer 2 fois sur une case	Fait apparaître le formulaire pour ajouter un travailleur à la tâche	02.juin	Correspond	Amélioration possible: ajout d'autofocus sur le champs texte
Ecrire une à plusieurs lettres dans le champs	Une liste de travailleurs dont le nom d'utilisateur qui correspond aux lettres inscrites apparaît	02.juin	Correspond	
Clic sur le nom d'utilisateur proposé dans la liste	Ajouter le travailleur à la case et rafraîchit le tableau. Si le travailleur est redondant dans la colonne (2 activités en même temps), indique une erreur	02.juin	Correspond, avec un message d'erreur "Le travailleur semble déjà faire quelque chose.."	
Passer sa souris sur une cellule qui contient un nom d'utilisateur	Fait apparaître une croix rouge à droite du nom d'utilisateur	02.juin	Correspond	Amélioration possible : La croix apparaît sans faire bouger toutes les autres cases (du à la longueur de la case)
Clic sur la croix apparue	Supprimer l'entrée de la base de données et rafraîchit le tableau	02.juin	Correspond	
Clic sur le bouton "+" dans la colonne action	Duplicile la ligne, sans les noms d'utilisateur	02.juin	Correspond	
Double clic sur une cellule de la nouvelle ligne	Fonctionne comme les autres	02.juin	BUG : overwrite du nom d'utilisateur en dessus et la colonne disparaît après le rafraîchissement du tableau	Mise en place de tout un système pour accepter plusieurs lignes par atelier, 5 jours de travail nécessaires BUG présent
Clic sur la croix dans la colonne action	Supprime les lignes dupliquées uniquement	02.juin	Correspond	
Clic sur le bouton PDF	Un document PDF s'ouvre avec le planning (et peut donc être imprimé)	02.juin	Correspond	Amélioration possible : masquer les boutons de la colonne "action" et modifier les styles

Page planning travailleurs, planning				
Affichage du planning	Le planning s'affiche correctement, les ateliers correspondent aux utilisateurs et les couleurs des ateliers sont respectées	02.juin	Correspond	
Click sur le bouton "Semaine précédente" ou "Semaine suivante"	Déplace le planning à la semaine souhaitée (précédente ou suivante) et met à jour la semaine affichée	02.juin	Correspond	
Clic sur le bouton PDF	Un document PDF s'ouvre avec le planning (et peut donc être imprimé)	02.juin	Correspond	Amélioration possible : Modifier le style du tableau

6 CONCLUSION

6.1 Bilan des fonctionnalités demandées

Dans l'ensemble le projet est tout de même bien avancé. Il est vrai qu'une fonctionnalité majeure est manquante et que des améliorations sont possibles. Le problème principal pour moi a été le temps. J'ai perdu pas mal de temps avec les fonctions qui affichent le planning, car c'était d'une grande difficulté. J'ai aussi perdu un peu de temps avec la première librairie que je voulais utiliser pour générer les PDF, mais j'ai pu trouver une alternative et faire fonctionner le tout, avec difficulté.

J'ai réussi à rendre mon application générique, mais il est nécessaire d'y avoir 3 niveaux.

Rendre le site responsive était une partie facile grâce à Bootstrap. Mais pour le planning, c'était une autre histoire. Etant donné qu'il a été fait main, même avec les styles Bootstrap je n'ai pas réussi à le rendre responsive à cause du temps.

J'ai aussi trouvé que l'utilisation de GIT était très utile. J'ai pu constater mon avancement dans le projet grâce aux commits et pu vérifier mon code ajouté et supprimé avant les push.

Pour terminer mon projet j'estimerais une durée supplémentaire nécessaire de deux petites semaines.

6.2 Bilan de la planification

Je pense avoir suivi correctement le planning. Je remarque que j'ai notamment perdu du temps sur l'analyse et sur le développement du planning.

6.3 Bilan personnel

Si le projet était à refaire, j'aurais peut-être essayé d'utiliser une librairie pour le planning. Au moins une librairie qui m'aurait permis de gérer les rowspan plus facilement. Je dois avouer que les tableaux html sont très casse-cou.

Aussi, j'ai voulu rendre mon application le plus dynamique possible. Pour ce faire, j'ai utilisé pas mal de JavaScript. Ce projet m'a beaucoup apporté au niveau de jQuery & AJAX. Je connaissais un peu comment cela fonctionnait avant de comment mon projet mais j'ai pu en apprendre beaucoup. C'est en tous cas un bon point pour moi.

Pour moi, le projet est suffisamment bien commenté pour être repris et terminé par quelqu'un d'autre (ou même moi-même).

7 DIVERS

7.1 Journal de travail

[Voir chapitre 8.3](#)

7.2 Webographie

- Dépôt du projet : <https://github.com/SachaGrenier/TPI>
- Laravel : <https://laravel.com/docs/5.4>
- Wkhtmltopdf : <https://github.com/barryvdh/laravel-snappy>
- Dompdf : <https://github.com/barryvdh/laravel-dompdf>
- Datatables : <https://datatables.net>
- jMerise : <http://www.jfreesoft.com/JMerise/>
- Librairie laravel HTML : <https://laravelcollective.com/docs/5.2/html>

8 ANNEXES

8.1 Cahier de charges

1 INFORMATIONS GENERALES

Candidat :	Nom : GRENIER mailto:maillot.grenierso@etml.educanet2.ch	Prénom : SACHA :
Lieu de travail :	ETML, Sébeillon 12 1004 Lausanne	
Chef de projet :	Nom : Gruaz mailto:maillot.alain.gruaz@vd.ceducanet2.ch	Prénom : Gilbert : 079 338 7808
Expert 1 :	Nom : Mottier mailto:andre.mottier@cpnv.ch	Prénom : André :
Expert 2 :	Nom : Borboen mailto:nicolas.borboen@epfl.ch	Prénom : Nicolas :
Dates de réalisation :	Du lundi 1 mai au mercredi 7 juin à 16h35 (au lieu de 11h25) Remarque : le jeudi après-midi 4 mai, le candidat sera absent pour raison de santé. D'où le report d'une journée pour le délai de fin de réalisation.	
Horaire de travail : (Basé sur l'horaire officiel)	Lundi 08h00-11h25 Penteôte, 5 juin 2017 Mardi Mercredi 08h00-12H15/13h10-16h35 Exa CG 31 mai 2017 après-midi Jeudi 08h00-11H25/12h20-16h35 Pont de l'Aso, 25 mai 2017 Vendredi 08h00-12H15/13h10-16h35 Pont de l'Aso, 26 mai 2017	
Présentation :	Entre le mercredi 14 et jeudi 15 juin 2017	
Nombre d'heures :	Environ 105 heures	
Planning (en H ou %)	Analyse : 10%, Implémentation 60%, Tests 10%, Doc. 20%	

2 PROCÉDURE

Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1er jour.

Le cahier des charges est approuvé par la i-CQ VD. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.

Le candidat a connaissance de la feuille d'appréciation avant de débuter le travail.

Le candidat est entièrement responsable de la sécurité de ses données.

En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son chef de projet.

Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier de projet.

A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires. Cette dernière doit être en tout point identique à la version électronique.

3 TITRE

Plateforme de planification hebdomadaire d'ateliers et de travailleurs

4 SUJET

Inspiré d'une situation réelle vécue au GRAAP-F (Fondation qui s'occupe de travailleurs AI), l'idée de ce projet est de réaliser une plate-forme Web qui permet, pour un chef d'ateliers, de faire une planification hebdomadaire, de plusieurs ateliers, dans lesquels travaillent des employés au bénéfice de l'AI.

Une réalisation **générique** de ce projet pourra permettre d'utiliser la plateforme pour planifier des classes avec des cours modulaires, ou des élèves inscrits dans des activités de camp de sport, etc ...

5 MATÉRIEL ET LOGICIEL À DISPOSITION

1 ordinateur standard ETML, avec la structure habituelle

A mettre en œuvre par le candidat :

- Une architecture WAMP (Easyphp, Xampp ou Wamp Server),
- Un framework PHP Laravel
- Un dépôt GIT

6 PRÉREQUIS

Avoir suivi les modules ICH à l'ETML, les projets et effectué des stages

7 DESCRIPTIF DU PROJET

Soit un l'exemple fictif suivant :

Planning hebdomadaire des ateliers

Catégories	Ateliers	jundi 09 janv	mardi 10 janv	mercredi 11 janv	jeudi 12 janv	vendredi 13 janv	
		Matin	Après-midi	Matin	Après-midi	Matin	Après-midi
Ateliers d'accueil	Accès	Edu_Personne	Alors_Demande	Edu_Personne	Edu_Personne	Edu_Personne	Fran_Schedule
	Accès			Edu_Personne	Edu_Personne	Edu_Personne	
	Accès			Edu_Personne	Edu_Personne	Edu_Personne	
	Téléphoniste	Mario_Etats_Mari	Eric_Personne	Yannick_Mariage	Yannick_Personne	Mario_Personne	
Ateliers restauration	Cantine	Cantine_Random	Cantine_Random	Cantine_Random	Cantine_Random	Cantine_Random	Cantine_Random
	Cantine	Arnaud_Bureau	Arnaud_Bureau	Arnaud_Bureau	Arnaud_Bureau	Arnaud_Bureau	Gratelle_Marche
	Cantine	Gratelle_Marche	Gratelle_Marche	Gratelle_Marche	Gratelle_Marche	Gratelle_Marche	Gratelle_Marche
	Cafeteria						
Ateliers généraux	Production	Mélanie_Boutiques	Mélanie_Boutiques	Olivier_Couleur	Olivier_Couleur	Olivier_Couleur	Mélanie_Boutiques
	Production	Carole_Du_Site_Forette	Manuel_Lapres_Et_Cou	Carole_Du_Site_Forette	Manuel_Lapres	Dominique_Habite	Carole_Du_Site_Forette
	Production	Maud_Dispar	Arnaud_Méhanismes	Dominique_Habite	Dominique_Habite	Gwendoline_Habite	Jean-Baptiste_Solal
	Entretien	Jonathan_Solal					
Total		9	8	8	9	8	7
Effectif total		77					4

Il s'agit, dans cet exemple, de permettre à l'utilisateur d'ajouter, modifier et supprimer les noms des travailleurs qui travaillent dans un atelier à un moment donné.

Les fonctionnalités suivantes doivent être disponibles (ce qui n'apparaît pas dans l'exemple) :

- Détection automatique de redondances. Une même personne ne peut pas apparaître dans 2 ateliers durant la même période (matin ou après-midi)
- Il faut pouvoir facilement ajouter ou supprimer des lignes ateliers, s'il y a plus de monde ou moins, en imaginant qu'un atelier peut accueillir au moins 20 travailleurs.
- Des masques permettront de saisir les noms des travailleurs, les noms des ateliers et les noms des catégories
- Les noms d'ateliers doivent toujours correspondre à la catégorie associée. Dans cette exemple, les 3 couleurs correspondent aux 3 catégories d'atelier. Cet aspect devrait aussi avoir une certaine flexibilité, par exemple jusqu'à 15 catégories.
- Il doit être possible d'imprimer le planning

8 POINTS ÉVALUÉS DURANT LE PROJET

- Le comportement et l'engagement durant toute la période pour le travail
- La bonne tenue à jour, systématique, des documents (JNLTRAV, rapport, etc.)
- Les interactions avec le chef de projet

9 LIVRABLES

Le candidat est responsable de mettre à disposition, pour son chef de projet et les deux experts, un dépôt GIT, sur un cloud, avec :

- La planification initiale
- Le rapport de projet
- Le journal de travail
- Le code qu'il produit

10 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les trois points spécifiques suivants :

1. *Le bon fonctionnement de ce qui sera livré, avec des données réalistes en nombres suffisants pour donner une idée du fonctionnement de l'application.*
2. *La flexibilité de la solution proposée, en vue d'être adaptée et utilisée dans d'autres situations analogues (horaires d'écoles, organisations de camps, etc.)*
3. *La portabilité du dispositif dans un nouvel environnement*

11 VALIDATION

	Lu et approuvé le :	Signature :
Candidat :		
Expert n°1 :		
Expert n° 2 :		
Chef de projet :		

8.2 Script SQL obtenu avec l'application jMerise

```
#-----
#      Script MySQL.
#-----
```

```
#-----
# Table: t_atelier_niv3
#-----
```

```
CREATE TABLE t_atelier_niv3(
    idAtelierNiv3 int (11) Auto_increment NOT NULL ,
    ateNom      Varchar (25) NOT NULL ,
    idAtelierNiv2 Int NOT NULL ,
    PRIMARY KEY (idAtelierNiv3 )
)ENGINE=InnoDB;
```

```
#-----
# Table: t_atelier_niv2
#-----
```

```
CREATE TABLE t_atelier_niv2(
    idAtelierNiv2 int (11) Auto_increment NOT NULL ,
    ateNom      Varchar (25) NOT NULL ,
    idAtelierNiveau1 Int NOT NULL ,
    PRIMARY KEY (idAtelierNiv2 )
)ENGINE=InnoDB;
```

```
#-----
# Table: t_travailleur
#-----
```

```
CREATE TABLE t_travailleur(
    idTravailleur int (11) Auto_increment NOT NULL ,
    traNom      Varchar (25) NOT NULL ,
    traPrenom   Varchar (25) NOT NULL ,
    traPourcentage Int NOT NULL ,
    idMSP       Int NOT NULL ,
    PRIMARY KEY (idTravailleur )
)ENGINE=InnoDB;
```

```
#-----
# Table: t_atelier_niv1
#-----
```

```
CREATE TABLE t_atelier_niv1(
    idAtelierNiveau1 int (11) Auto_increment NOT NULL ,
    ateNom      Varchar (25) NOT NULL ,
    idCouleur   Int NOT NULL ,
    PRIMARY KEY (idAtelierNiveau1 )
)ENGINE=InnoDB;
```

```
#-----
# Table: t_couleur
#-----
```

```
CREATE TABLE t_couleur(
    idCouleur     int (11) Auto_increment NOT NULL ,
    couNomCouleur Varchar (25) NOT NULL ,
    couCodeHexadecimal Varchar (40) NOT NULL ,
    PRIMARY KEY (idCouleur )
)ENGINE=InnoDB;
```

```
#-----
# Table: t_MSP
#-----
```

```
CREATE TABLE t_MSP(
    idMSP      int (11) Auto_increment NOT NULL ,
    mspNom     Varchar (25) NOT NULL ,
    mspPrenom  Varchar (25) NOT NULL ,
    msplInitials Varchar (4) NOT NULL ,
```

```
    PRIMARY KEY (idMSP )
)ENGINE=InnoDB;
```

```
#-----
# Table: t_tache
#-----
```

```
CREATE TABLE t_tache(
    taDate      Date ,
    taEstMatin  Bool ,
    idAtelierNiv3 Int NOT NULL ,
    idTravailleur Int NOT NULL ,
    PRIMARY KEY (idAtelierNiv3 ,idTravailleur )
)ENGINE=InnoDB;
```

```
ALTER TABLE t_atelier_niv3 ADD CONSTRAINT FK_t_atelier_niv3_idAtelierNiv2 FOREIGN KEY
(idAtelierNiv2) REFERENCES t_atelier_niv2(idAtelierNiv2);
ALTER TABLE t_atelier_niv2 ADD CONSTRAINT FK_t_atelier_niv2_idAtelierNiveau1 FOREIGN KEY
(idAtelierNiveau1) REFERENCES t_atelier_niv1(idAtelierNiveau1);
ALTER TABLE t_travailleur ADD CONSTRAINT FK_t_travailleur_idMSP FOREIGN KEY (idMSP)
REFERENCES t_MSP(idMSP);
ALTER TABLE t_atelier_niv1 ADD CONSTRAINT FK_t_atelier_niv1_idCouleur FOREIGN KEY
(idCouleur) REFERENCES t_couleur(idCouleur);
ALTER TABLE t_tache ADD CONSTRAINT FK_t_tache_idAtelierNiv3 FOREIGN KEY (idAtelierNiv3)
REFERENCES t_atelier_niv3(idAtelierNiv3);
ALTER TABLE t_tache ADD CONSTRAINT FK_t_tache_idTravailleur FOREIGN KEY (idTravailleur)
REFERENCES t_travailleur(idTravailleur);
```

8.3 Journal de travail

Semaine 1		Date: lundi 1 mai 2017	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	
Lundi			
Analyse	1	Analyse du cahier de charges, recherches sur internet et analyse de la structure de la base de données (basique)	
Création de la planification initiale	7		
Prise de connaissance du projet	3	Scéance avec l'expert	
Journal de travail	1	Journal de travail	
Mercredi			
Analyse	12	Analyse du cahier de charges : Qu'est-ce que je vais devoir faire, comment, combien de pages, etc.	
Analyse	3	Recherches sur internet (libraires pdf, tableaux, etc)	https://datatables.net/ https://wkhtmltopdf.org/
Analyse	2	Analyse de la structure de la base de données (basique). Pour ce faire, j'ai imprimé l'exemple donné par Mr.Gruaz et analysé les tables dont j'allais avoir besoin.	MCD crée à partir du jeudi, https://github.com/SachaGrenier/TPI/blob/master/TPI_files/DB/MCD.PNG
	9	Suite à l'analyse de l'exemple donné par Mr.Gruaz, j'ai voulu créer la maquette de la page principale du site	https://github.com/SachaGrenier/TPI/blob/master/TPI_files/Maquette/MM-TPI-greniersa-web-Maquette.vsdx
Jeudi			
Analyse	3	Suite de la maquette, où j'ai attaqué la deuxième page du site	https://github.com/SachaGrenier/TPI/blob/master/TPI_files/Maquette/MM-TPI-greniersa-web-Maquette.vsdx
Absence - Imprévus	18	Rendez-vous chez le dentiste pour opération	
Documentation	1	Journal de travail (mercredi et jeudi)	
		Création MCD MLD sous visio	Documents finalement réalisés à l'aide de l'outil JMerise, voir résultat ici : https://github.com/SachaGrenier/TPI/tree/master/TPI_files/DB (voir fichiers MCD.png et MLD.png)
Vendredi			
Documentation	19	Implémentation de la partie analyse dans la documentation, notamment les maquettes avec une description détaillée.	
Absence - Imprévus	2	Retard de 30 min à cause du rendez-vous chez l'orthodontiste (prévu)	
Analyse	3	Création MPD sous excel	https://github.com/SachaGrenier/TPI/tree/master/TPI_files/DB
Documentation	1	Journal de travail	
Analyse	3	Création de la maquette pour la page "Travailleurs" sous visio	https://github.com/SachaGrenier/TPI/blob/master/TPI_files/Maquette/MM-TPI-greniersa-web-Maquette.vsdx
Total semaine	93	Max. 93	

Semaine	2	Date: lundi 8 mai 2017	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Analyse	5	> Convenu avec GGZ d'utiliser l'outil JMerise pour MCD-MLD-MPD. > Clarification du cahier de charges : Correction des maquettes faites : système de niveaux d'ateliers plutôt que de catégories => cohérence avec le classeur d'exemple excel > Correction de la base de données pour accueillir les chefs d'ateliers ainsi que le pourcentage de travail pour les utilisateurs	http://www.jfreesoft.com/JMerise/
Création MCD, MLD et MPD	7	Modification du MCD, MLD et MPD	https://github.com/SachaGrenier/TPI/tree/master/TPI_files/DB
Mercredi			
Création des maquettes	4	Améliorations sur les maquettes : Ajout des systèmes de niveaux, fin de la page "travailleurs"	
Documentation	2	Insertion des nouveaux MCD, MLD et MPD dans la documentation et ajout de commentaires	
Documentation	8	Ajout des maquettes dans la documentation avec une description	
Mise en place du dépôt GitHub	2	Création du dépôt GitHub	https://github.com/SachaGrenier/TPI/
Création de la planification détaillée	9		https://github.com/SachaGrenier/TPI/blob/master/TPI_files/T-TPI-greniersa-web-Planification_detailee.xslm
Journal de travail	2		
Jeudi			
Journal de travail	2	Mise à jour du des tâches journal de travail en utilisant celles de la planification détaillée	
Création de migrations	6	Création du fichier de migration de base pour générer la base de données	https://github.com/SachaGrenier/TPI/blob/master/database/migrations/2017_05_11_062800_create_all_tables.php
Création du layout	6	Création d'une base de design et implémentation des fichiers CSS et JS (exemple : jQuery, DataTables). Le fichier sera modifié au fur et à mesure pour accueillir différentes librairies	https://github.com/SachaGrenier/TPI/blob/master/resources/views/layouts/app.blade.php
Développement du tableau de travailleurs	13	>Implémentation de DataTables sur la page >Création de la fonction "getWorkersArray" qui récupère les travailleurs depuis la DB et le formate (ajax) >Création de la fonction "deleteWorker" pour supprimer un travailleur de la DB (ajax)	
Vendredi			
Développement du tableau de travailleurs	16	>Création de la fonction "addWorker" qui ajoute un travailleur dans la DB à l'aide d'un formulaire (ajax)	http://tutsnare.com/post-data-using-ajax-in-laravel-5/
Développement du tableau de MSP	9	>Implémentation du tableau sur la page >Création de la fonction getAllMSPs pour récupérer tous les Maître Sociaux Professionnels de la DB >Création de la fonction "addMSP" qui ajoute un MSP dans la DB à l'aide d'un formulaire >Création de la fonction deleteMSP pour supprimer un travailleur de la DB >Ajout d'une librairie laravel pour utiliser un système de formulaire optimal	
Journal de travail	1		
Documentation	1	Ajout d'une première partie de la description du tableau de travailleurs	
Total semaine	93	Max. 93	
Semaine	3	Date: lundi 15 mai 2017	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Création de la grille de test	1	Création d'un fichier de base, à remplir au fur et à mesure	
Analyse	1	Discussion avec GGZ : Eventuelles corrections sur le MCD à faire	
Journal de travail	1		
Documentation	4	Description de l'installation de l'infrastructure, compléction de la description de la page travailleurs et début explications de la partie MSP	
Développement du planning sur la page planning général	5	Début de la création du planning sous format HTML	
Mercredi			
Développement du menu de gestion des niveaux	20	>Création de la fonction "getMenu" pour afficher le menu en javascript >Développement des fonctions pour ajouter et supprimer des menus de différents niveaux de la base de données >Mise en place des fonctions ajax pour envoyer les données aux fonctions ci-dessus >Travail sur le style du menu, pour qu'il soit plus simple et intuitif	
Documentation	4	Description du menu de niveaux	
Journal de travail	1		
Création de la grille de test	2	Ajout des tests à effectuer jeudi matin	https://github.com/SachaGrenier/TPI/blob/master/TPI_files/X-TPI-greniersa-web-Grille_de_tests.xlsx
Jeudi			
Tests	3	Completion des tests en suivant la grille, note des améliorations à effectuer : >Gestion des pourcentages (entre 1 et 100%) >Modification de la table des travailleurs pour enlever la possibilité de trier la colonne supprimer >Modification des messages d'erreurs, pour qu'ils soient plus performants	https://github.com/SachaGrenier/TPI/blob/master/TPI_files/X-TPI-greniersa-web-Grille_de_tests.xlsx
Développement du planning sur la page planning général	23	>Mise en place des niveaux sur le planning, pour l'instant directement en PHP sur la page. Sera peut-être modifié pour accueil de l'ajax dans le futur >Début d'implémentation du formulaire pour ajouter un travailleur à une tâche (autocomplete) >Affichage des dates en fonction de la semaine en cours	https://jqueryui.com/autocomplete/ http://carbon.nesbot.com/docs/ -> utiliser pour la gestion des dates
Journal de travail	1		
Vendredi			
Tests	3	Réalisation des tests suivant la grille : quelques fonctions sont à retoucher	https://github.com/SachaGrenier/TPI/blob/master/TPI_files/X-TPI-greniersa-web-Grille_de_tests.xlsx
Journal de travail	1		
Développement du planning sur la page planning général	23	>Création de la structure pour afficher les travailleurs dans les cases >Fin d'implémentation de l'autocomplete pour ajouter des travailleurs à des tâches avec vérification de redondance >Création de la fonction qui génère la liste des tâches et création de la fonction qui les assigne dans le tableau à l'aide de jQuery et AJAX	
Total semaine	93	Max. 93	

Semaine	4	Date: lundi 22 mai 2017	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Développement du planning sur la page planning général	6	Mise en place des fonctions pour déplacer le planning à travers les semaines	
Analyse	2	Visite de Mr.Borboen	
Journal de travail	1		
Mercredi			
Analyse	1	Discussion avec le chef de projet : > taille des collones sur le tableau >intuitivité	
Journal de travail	1		
Développement du planning sur la page planning général	25	>Développement de la fonction pour ajouter des lignes au tableau >Suppression d'un travailleur à une tâche	
Documentation	3	Documentation de la page planning	
Absence - Imprévus	54		
Total semaine	93	Max. 93	
Semaine	5	Date: lundi 29 mai 2017	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Documentation	6	Mise à jour de la documentation	
Développement du planning sur la page planning général	5	Développement de la fonction pour ajouter des lignes au tableau	
Journal de travail	1		
Mercredi			
Développement du planning sur la page planning général	6	Développement de la fonction pour ajouter des lignes au tableau enfin terminée	
Développement du planning sur la page planning général	6	Développement de la fonction pour supprimer des lignes au tableau	
Développement du planning sur la page planning général	3	Installation de wkhtmltopdf pour créer les PDF	https://github.com/barryvdh/laravel-snappy
Absence - Imprévus	12	Examen culture générale	
Jeudi			
Absence - Imprévus	1	Scénario de classe	
Développement du planning sur la page planning général	11	Mise en place de wkhtmltopdf, jusqu'à que je me rende compte que la librairie n'était pas adaptée à mes besoins	
Développement du planning sur la page planning général	12	Mise en place de domPDF et création de la fonction pour générer le fichier pdf du planning	
Documentation	3	Ajout de documentation par rapport au planning	https://github.com/barryvdh/laravel-dompdf
Vendredi			
Journal de travail	1		
Tests	6	Création des tests pour la page de planning et tests	
Documentation	6	Ajout de documentation par rapport au planning	
Développement du planning sur la page planning travailleurs	8	Développement du planning sous forme readonly	
Documentation	4	Ajout de documentation par rapport à la page planning travailleurs	
Tests	2	Création des tests pour la page de planning travailleurs et tests	
Total semaine	93	Max. 93	
Semaine	6	Date: lundi 5 juin 2017	
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Lundi			
Absence - Imprévus	12	Lundi de pentecôte	
Mercredi			
Documentation	23	Fin et peaufinage de la documentation Impression et envoi des documents	
Absence - Imprévus	4	Rendez-vous chez orthodontiste	
Total semaine	39	Max. 93	