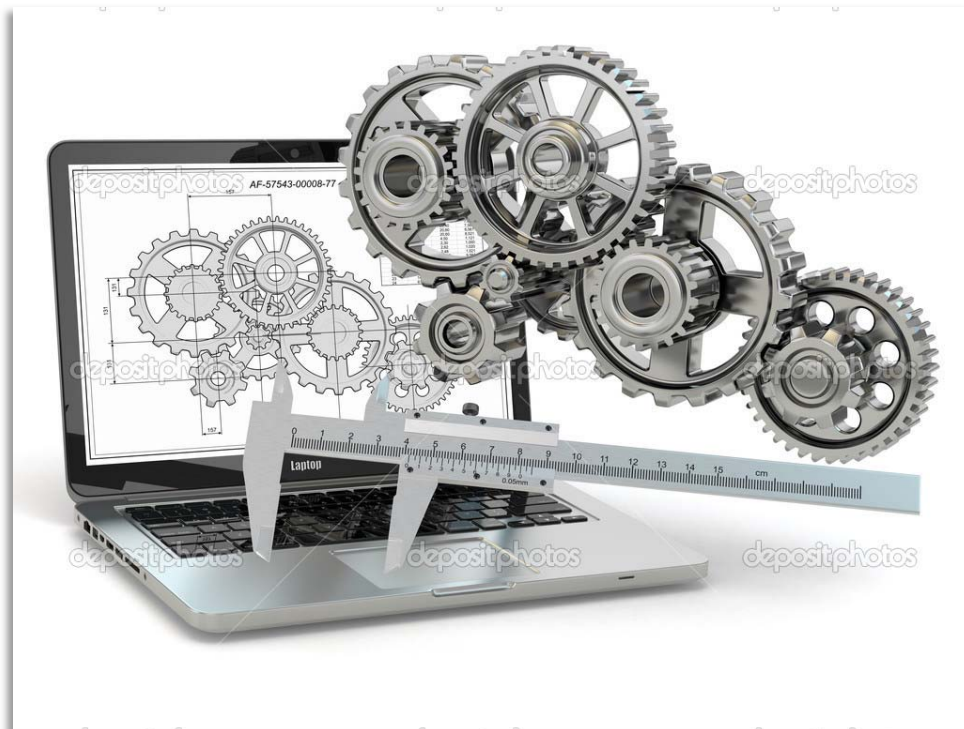


Planning web



(Une image originale représentant le projet)

Sacha Grenier – Cin4b

ETML - Lausanne

Durée : 105 heures

Chef de projet : Gruaz Gilbert

Experts : Mottier André, 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 utilisateurs</i>	8
3.3	CONCEPTION DES TESTS	13
3.4	PLANIFICATION DÉTAILLÉE	13
4	RÉALISATION.....	ERREUR ! SIGNET NON DEFINI.
4.1	DOSSIER DE RÉALISATION	15
4.1.1	<i>Mise en place de l'environnement</i>	15
4.2	MODIFICATIONS.....	23
5	TESTS.....	24
5.1	DOSSIER DES TESTS	24
6	CONCLUSION.....	24
6.1	BILAN DES FONCTIONNALITÉS DEMANDÉES	26
6.2	BILAN DE LA PLANIFICATION	26
6.3	BILAN PERSONNEL.....	26
7	DIVERS.....	26
7.1	JOURNAL DE TRAVAIL.....	26
7.2	BIBLIOGRAPHIE.....	26
7.3	WEBOGRAPHIE	26
8	ANNEXES	26
8.1	SCRIPT SQL OBTENU AVEC L'APPLICATION JMERISE.....	27

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

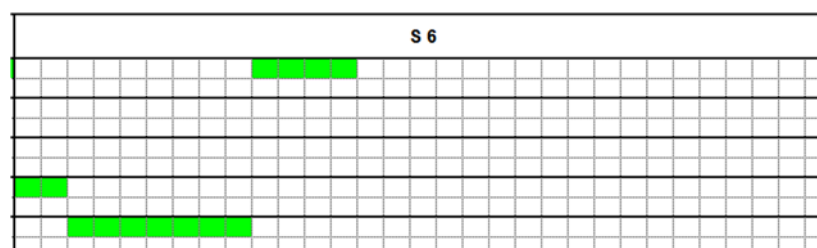
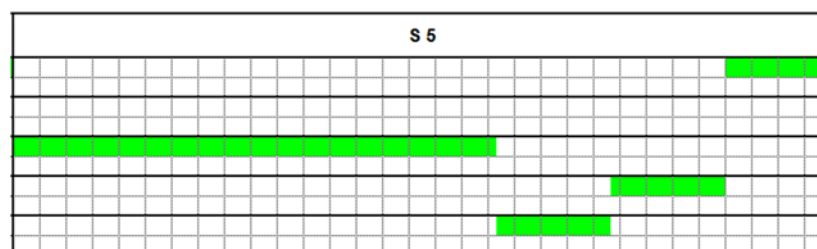
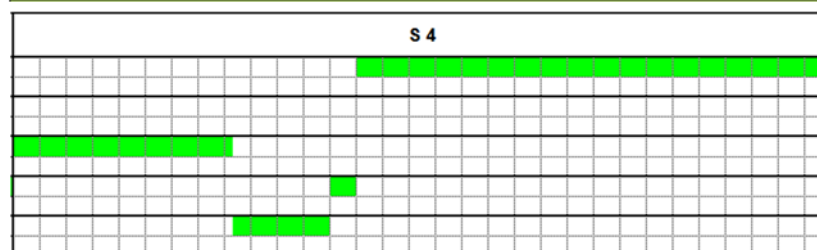
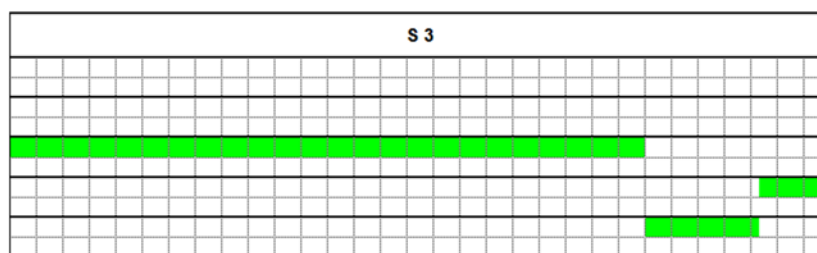
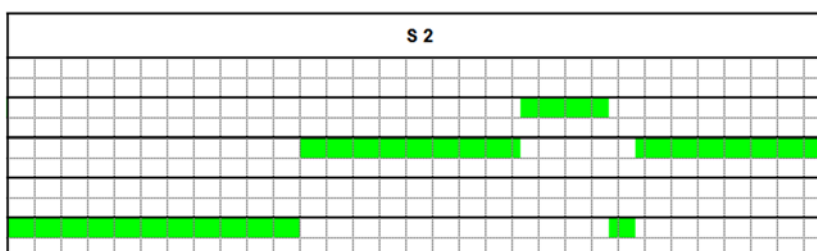
1.5 Cahier des charges

Le document fourni par le chef de projet fait foi. Il doit être mis en annexe ➔ [Lien sur CDC](#)

2 PLANIFICATION INITIALE

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

Tâches - objectifs	Nb 1/4 heure	S 1
Absence - Imprévus	103 0	
Analyse	58 0	
Implémentation	199 0	
Tests	30 0	
Documentation	114 0	
Total planifié	504	
Total réalisé	0	



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
- Créer une grille de planning et la rendre dynamique

Les difficultés seront :

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

Ainsi que les solutions possibles

- Si les spécifications de départ ne laissent pas de doutes sur la manière de réaliser un projet, ce chapitre ne fera que renvoyer le lecteur aux spécifications.

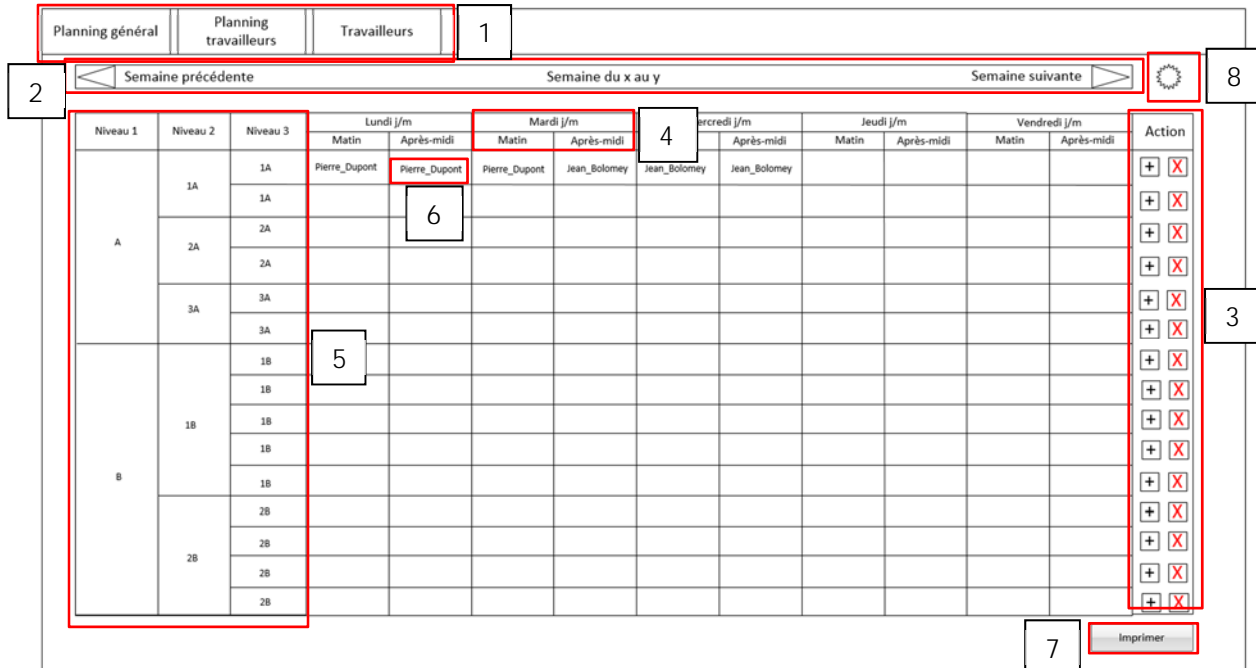
3.2 Document d'analyse et conception

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 concernant 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 web interface for a general planning tool. At the top, there is a navigation bar with three tabs: 'Planning général' (highlighted with callout 1), 'Planning travailleurs', and 'Travailleurs'. Below this is a header section with callout 2 containing navigation buttons for 'Semaine précédente', 'Semaine du x au y', and 'Semaine suivante', along with a settings gear icon (callout 8). The main area is a grid with columns for days of the week (Lundi to Vendredi) and sub-columns for 'Matin' and 'Après-midi'. Callout 4 points to the 'Mardi' column headers. The grid rows are categorized by 'Niveau 1' and 'Niveau 2'. Under 'Niveau 1', there is a section 'A' with rows 1A, 2A, and 3A. Under 'Niveau 2', there is a section 'B' with rows 1B, 2B, and 3B. Callout 5 points to the 'Niveau 2' section. Callout 6 points to a specific cell in the 'Après-midi' column for 'Mardi' under 'Niveau 2'. To the right of the grid is an 'Action' column (callout 3) with '+' and 'X' buttons for each row. At the bottom right, there is an 'Imprimer' button (callout 7).

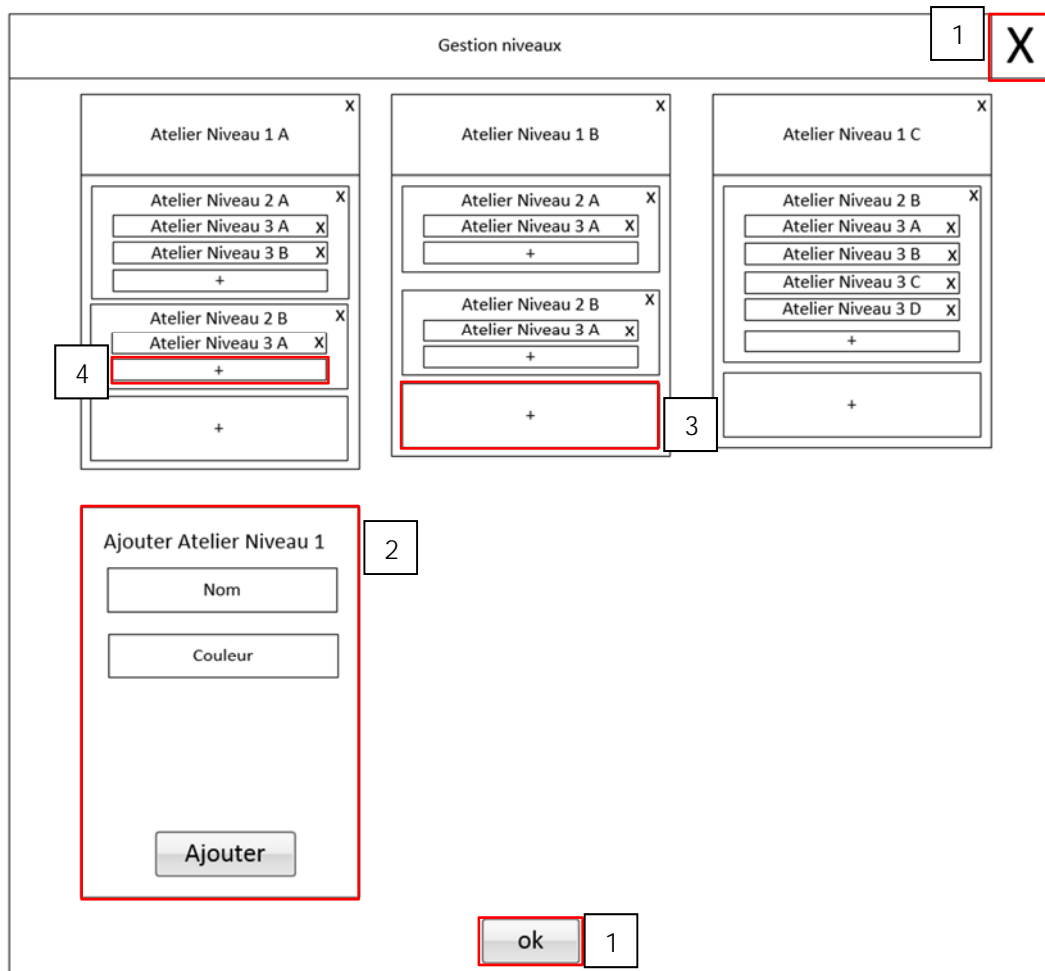
Description :

- 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 (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 :



The screenshot shows a web application titled "Gestion niveaux". It features three main columns representing different levels: "Atelier Niveau 1 A", "Atelier Niveau 1 B", and "Atelier Niveau 1 C". Each column contains a list of sub-levels (e.g., "Atelier Niveau 2 A", "Atelier Niveau 3 A", "Atelier Niveau 3 B") with a small 'X' button next to each item. A red box highlights a '+' button in the "Atelier Niveau 3 A" section of the first column, with a callout '4'. Another red box highlights a '+' button in the "Atelier Niveau 2 B" section of the second column, with a callout '3'. A third red box highlights a '+' button in the "Atelier Niveau 3 A" section of the third column, with a callout '2'. A fourth red box highlights a '+' button in the "Atelier Niveau 2 B" section of the third column, with a callout '1'. At the bottom left, there is a form titled "Ajouter Atelier Niveau 1" with fields for "Nom" and "Couleur", and an "Ajouter" button. At the bottom right, there is an "ok" button with a callout '1'. A red 'X' button is located in the top right corner of the interface.

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 :

Planning général

Planning travailleurs

Travailleurs

1

◀

Semaine précédente

Semaine du x au y

Semaine suivante

▶

Nom	Prénom	MSP	%	Lundi j/m		Mardi j/m		Mercredi j/m		Jeudi j/m		Vendredi j/m	
				Matin	Après-midi	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi
Dupont	Pierre	SP	30/90	1A	1A	1A							
Bolomey	Jean	SP	30/50				1A	1A	1A				

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 Maître socioprofessionnel
- 4) Colonne %, indiquant le pourcentage effectif/prévu
- 5) Bouton d'impression pour imprimer (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																																																	
<div style="text-align: right;">Recherche</div> <table border="1"> <thead> <tr> <th>ID</th> <th>Prénom</th> <th>Nom</th> <th>Nom d'utilisateur</th> <th>MSP</th> <th>%</th> <th>Supprimer</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> </tr> <tr> <td>2</td> <td>Jean</td> <td>Bolomey</td> <td>Jean_Bolomey</td> <td>SP</td> <td>50</td> <td>X</td> </tr> <tr> <td></td> <td><input type="text"/></td> <td><input type="text"/></td> <td>Prénom_Nom</td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="button" value="Ajouter"/></td> </tr> <tr><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></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>			ID	Prénom	Nom	Nom d'utilisateur	MSP	%	Supprimer	1	Pierre	Dupont	Pierre_Dupont	SP	90	X	2	Jean	Bolomey	Jean_Bolomey	SP	50	X		<input type="text"/>	<input type="text"/>	Prénom_Nom	<input type="text"/>	<input type="text"/>	<input type="button" value="Ajouter"/>																					
ID	Prénom	Nom	Nom d'utilisateur	MSP	%	Supprimer																																													
1	Pierre	Dupont	Pierre_Dupont	SP	90	X																																													
2	Jean	Bolomey	Jean_Bolomey	SP	50	X																																													
	<input type="text"/>	<input type="text"/>	Prénom_Nom	<input type="text"/>	<input type="text"/>	<input type="button" value="Ajouter"/>																																													
<table border="1"> <thead> <tr> <th>ID</th> <th>Prénom</th> <th>Nom</th> <th>Initiales</th> <th>Suppr</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Stéphane</td> <td>Pottier</td> <td>SP</td> <td>X</td> </tr> <tr><td> </td><td> </td><td> </td><td> </td><td>X</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td>X</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td>X</td></tr> </tbody> </table>			ID	Prénom	Nom	Initiales	Suppr	1	Stéphane	Pottier	SP	X					X					X					X																								
ID	Prénom	Nom	Initiales	Suppr																																															
1	Stéphane	Pottier	SP	X																																															
				X																																															
				X																																															
				X																																															
<div> <input type="text" value="Prénom"/> <input type="text" value="Nom"/> <input type="text" value="Initiales"/> <input type="button" value="Ajouter"/> </div>																																																			

Description :

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

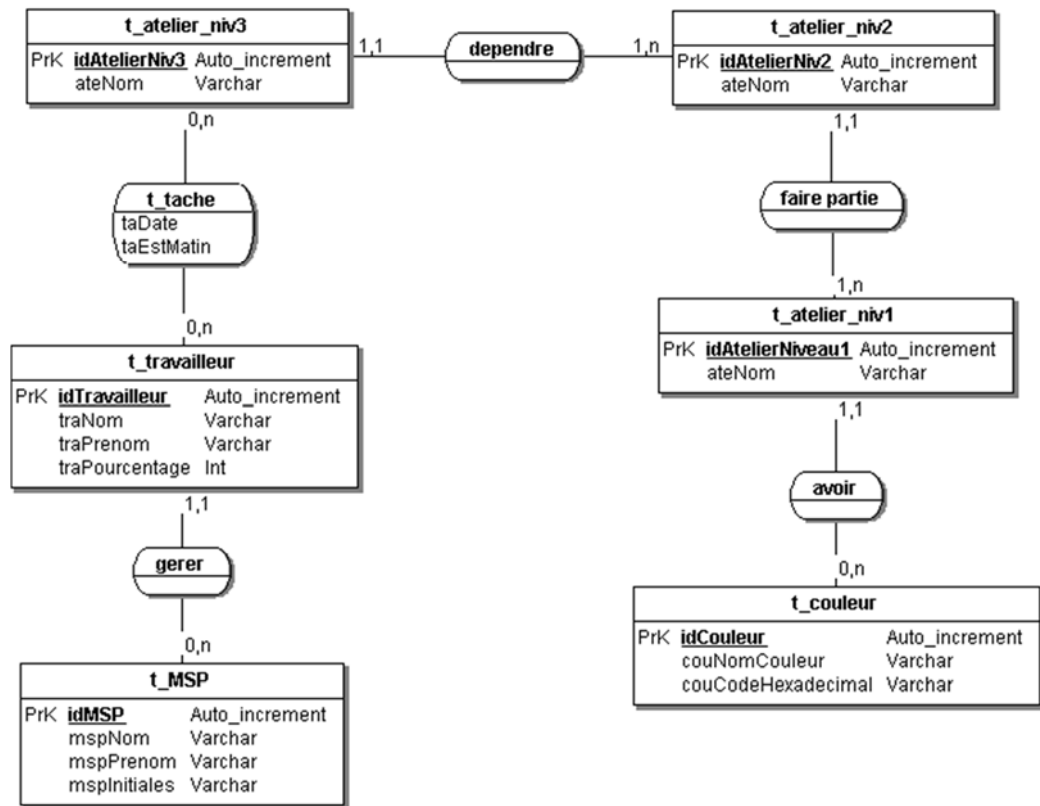
3.2.4 Base de données

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 ETML.

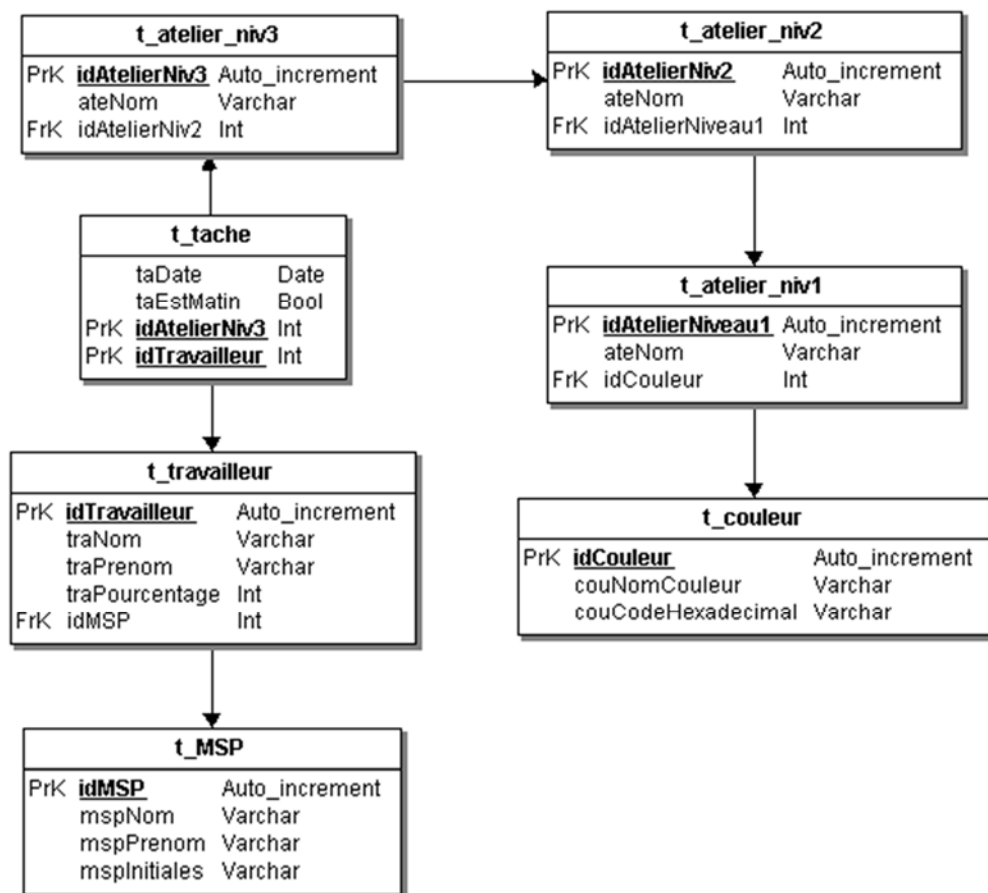
On y retrouve donc 5 tables basiques et une table intermédiaire.

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, 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>

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
name	VARCHAR (40)	-	-	-	Oui
workshop_level_2_id	int(10) (UNSIGNED)	-	clé étrangère	INDEX	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
-------------------------	------------------	------	-----------------	-----------

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

3.4 Planification détaillée

Semaine 1		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
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		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
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 planning 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

3.5 Dossier de Réalisation

3.5.1 Mise en place de l'environnement

Premièrement, j'ai installé une machine virtuelle avec VMare 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.

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 :



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 exécuter 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,

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

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

3.5.2 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

3.5.3 Page « Travailleurs »

Travailleurs

Afficher 10 résultats

ID	Prénom	Nom	Nom d'utilisateur	%	MSP	Crée le	Supprimer
	<input type="text" value="Prénom"/>	<input type="text" value="Nom"/>	<input type="text" value="Nom d'utilisateur"/>	<input type="text" value="%"/>	<input type="text" value="Sélectionner MSP"/>	29 mai 2017	<input type="button" value="Ajouter"/>
2	Pierre	Bolomey	Pierre_Bolomey	90	Stéphane Pottier (SP)	11 mai 2017	<input type="button" value="X"/>
12	Jean	Pierre	Jean_Pierre	50	Joseph Dupont (JD)	12 mai 2017	<input type="button" value="X"/>
13	Albert	Dumoulin	Albert_Dumoulin	60	Joseph Dupont (JD)	15 mai 2017	<input type="button" value="X"/>
14	Jacques	Dupond	Jacques_Dupond	50	Stéphane Pottier (SP)	17 mai 2017	<input type="button" value="X"/>

5 résultats (Affiche de 1 à 5) 1

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

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.

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

    $array = [];

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

    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;
    }
    $form = new stdClass();

    $form->id = "";
    $form->firstname = "<input class='table-input' type='text' placeholder='Prénom' id='firstname'>";
    $form->lastname = "<input class='table-input' type='text' placeholder='Nom' id='lastname'>";
    $form->username = "<input class='table-input' type='text' id='username' placeholder='Nom d'utilisateur'>";
    $form->percentage = "<input class='table-input 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;

    return $array;
}
```

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

Maîtres socioprofessionnel

ID	Prénom	Nom	Initiales	Supprimer
1	Stéphane	Pottier	SP	<input type="button" value="X"/>
2	Joseph	Dupont	JD	<input type="button" value="X"/>
25	Jean	Robert	JR	<input type="button" value="X"/>
26	Bernard	Joubert	BJ	<input type="button" value="X"/>

```
@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
```

3.5.4 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 représente 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.

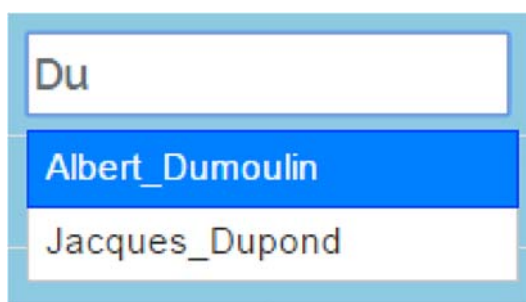
3.5.5 Planning général

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 du 15 au 19 mai 2017									
Semaine précédente			Semaine suivante			Gestion niveaux			
	Niv 1	Niv 2	Niv 3	Lundi 15		Mardi 16		Mercredi 17	
				Matin	Après-midi	Matin	Après-midi	Matin	Après-midi
	Atelier Services et prestations sociales	Secrétariat-réception	Réception		Jacques_Dupond				
			Chauffeur	Albert_Dumoulin					
			Redacteur	Jean_Pierre					
		Animation	Animation	Jean_Pierre					
	Atelier Grain de Sel	Cuisine	Cuisine	Jacques_Dupond					
		Cafeteria	Cafeteria	Jean_Pierre					
	Atelier Général	Production	Production						
		Entretien	Entretien				Albert_Dumoulin		
Action									

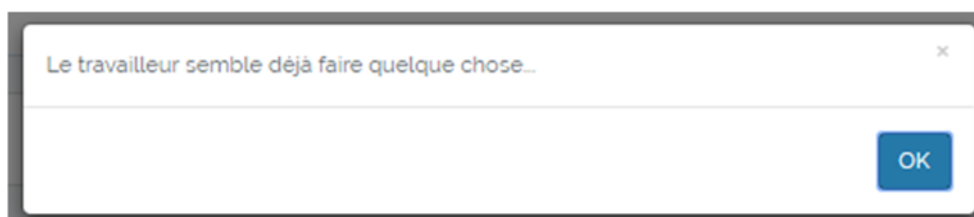
La table est générée en html et PHP, mais une grande partie des fonctions sont en JavaScript (jQuery) et Ajax. Il m'a paru important d'avoir un outil dynamique mais cela a rendu la tâche plus compliquée.

Pour assigner un utilisateur 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 de l'utilisateur 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 la.

Dans ce cas la, 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 :



Il est aussi possible de supprimer un utilisateur à une tâche. Pour ce faire, lorsque l'utilisateur passe sa souris sur les cellules, une croix apparaît à côté des logins lorsque 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);
});
```

3.6 Modifications

- Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.
- Date, raison, description, etc.

4 TESTS

4.1 Dossier des tests

Les tests ont été effectués en suivant la grille suivante :

Functionalité à tester	Résultat attendu	date	Résultat obtenu	Solutions
Page travailleurs, tableau de 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 rafraichit et une erreur intervient indiquant que les initiales choisies sont déjà utilisées	18.mai	La requête est tout de même acceptée	Vérifier si le nom d'utilisateur est libre avant de l'insérer dans la
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"	
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"	
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"	
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 travailleurs par tous les tris 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 de Maître socioprofessionnels				
Insérer un Maître socioprofessionnels dans la base de données à l'aide du formulaire	Le page se rafraichit et le MSP apparaît dans le tableau	18.mai	Correspond	
Insérer un MSP sans indiquer d'initiales	Le page se rafraichit 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 rafraichit et une erreur intervient indiquant que les initiales choisies sont déjà utilisées	18.mai	La requête est tout de même acceptée	Vérifier si les initiales sont libre avant de l'insérer dans la base de
Insérer un MSP et laisser les champs vide	Le page se rafraichit et une erreur intervient indiquant que le formulaire n'est pas complet	18.mai	Correspond	
Supprimer un MSP	Le page se rafraichit 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	
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	

5 CONCLUSION

5.1 Bilan des fonctionnalités demandées

- Il s'agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.
- Si ce n'est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu'il reste à accomplir pour terminer le tout.

5.2 Bilan de la planification

- Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différences entre les planifications initiales et détaillées avec le journal de travail.

5.3 Bilan personnel

- Si c'était à refaire:
 - Qu'est-ce qu'il faudrait garder ? Les plus et les moins ?
 - Qu'est-ce qu'il faudrait gérer, réaliser ou traiter différemment ?
- Qu'est que ce projet m'a appris ?
- Suite à donner, améliorations souhaitables, ...
- Remerciements, signature, etc.

6 DIVERS

6.1 Journal de travail

- Date, activité (description qui permet de reproduire le cheminement du projet), durée, liens et références sur des documents externes. Lorsqu'une activité de recherches a été entreprise, il convient d'énumérer ce qui a été trouvé, avec les références.

6.2 Bibliographie

- Références des livres, revues et publications utilisés durant le projet.

6.3 Webographie

- Références des sites Internet consultés durant le projet.

7 ANNEXES

- **Cahier des charges**
- Listing du code source (partiel ou, plus rarement complet)
- Guide(s) d'utilisation et/ou guide de l'administrateur
- Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).
- Extraits de catalogue, documentation de fabricant, etc.

7.1 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 ,  
    mspInitiales 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);
```