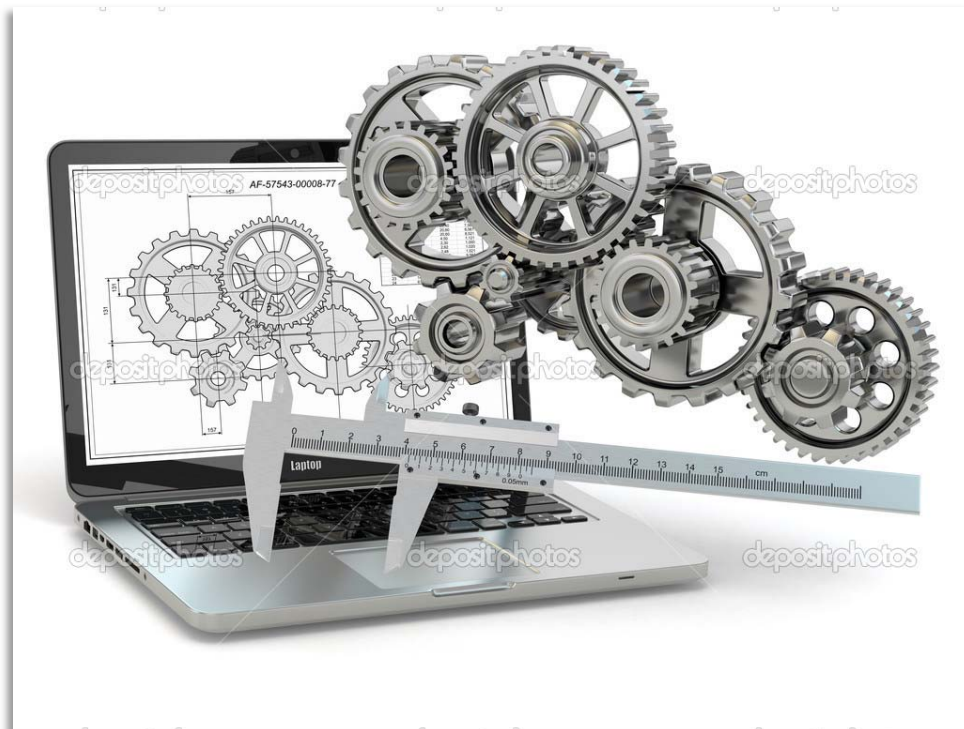


Planning web



(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.....	13
4.1	DOSSIER DE RÉALISATION	13
4.1.1	<i>Mise en place de l'environnement</i>	13
4.2	MODIFICATIONS.....	14
5	TESTS.....	14
5.1	DOSSIER DES TESTS	14
6	CONCLUSION.....	14
6.1	BILAN DES FONCTIONNALITÉS DEMANDÉES	14
6.2	BILAN DE LA PLANIFICATION	14
6.3	BILAN PERSONNEL.....	14
7	DIVERS.....	15
7.1	JOURNAL DE TRAVAIL.....	15
7.2	BIBLIOGRAPHIE.....	15
7.3	WEBOGRAPHIE	15
8	ANNEXES	15
8.1	SCRIPT SQL OBTENU AVEC L'APPLICATION JMERISE.....	15

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

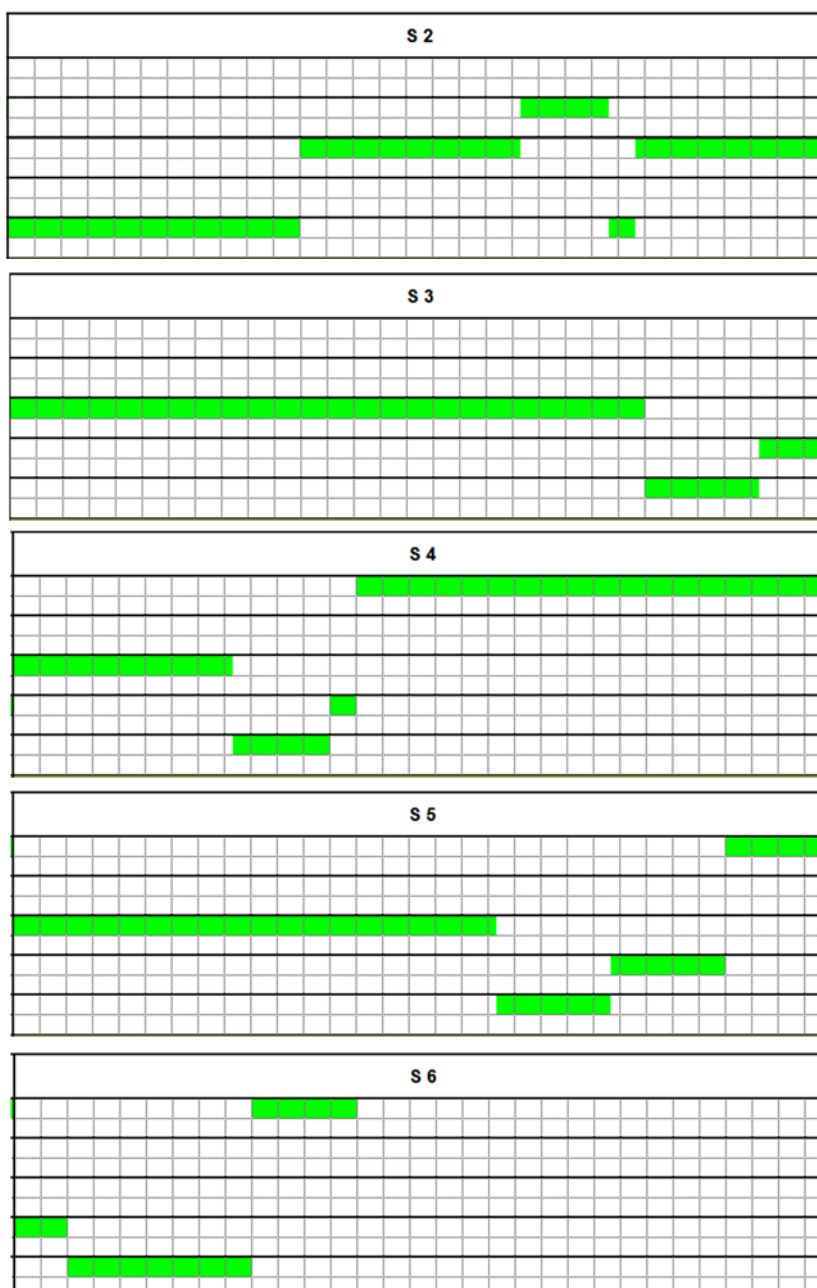
Avoir suivi les modules ICH à l'ETML, les projets et effectué des stages

1.5 Cahier des charges

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

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.

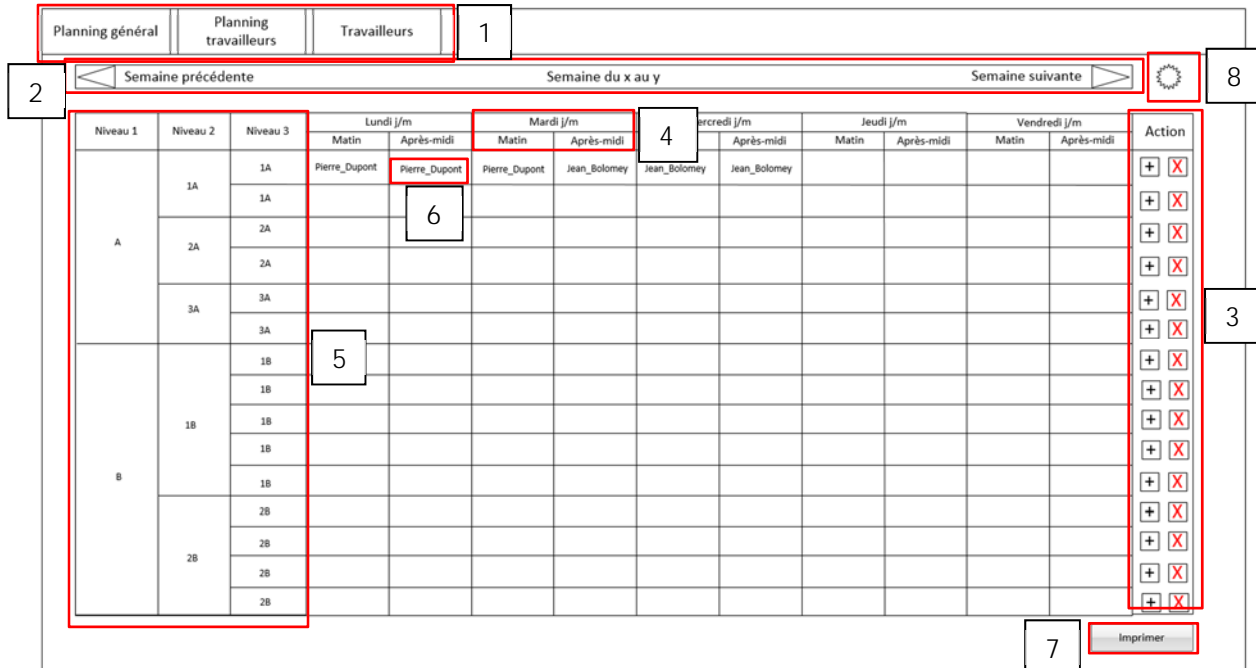
3.2 Document d'analyse et conception

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. It features a header with three tabs: 'Planning général' (highlighted with callout 1), 'Planning travailleurs', and 'Travailleurs'. Below the tabs is a navigation bar (callout 2) with 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 (callout 3) with columns for days of the week (Lundi to Vendredi) and rows for levels (Niveau 1, 2, 3) and tasks (1A, 2A, 3A, 1B, 2B). The grid cells contain names like 'Pierre_Dupont' and 'Jean_Bolomey'. A specific cell for 'Mardi' (callout 4) is highlighted. A search field (callout 6) is visible within a cell. An 'Action' column on the right (callout 3) contains '+' and 'X' buttons for each row. A footer bar (callout 7) includes an 'Imprimer' button.

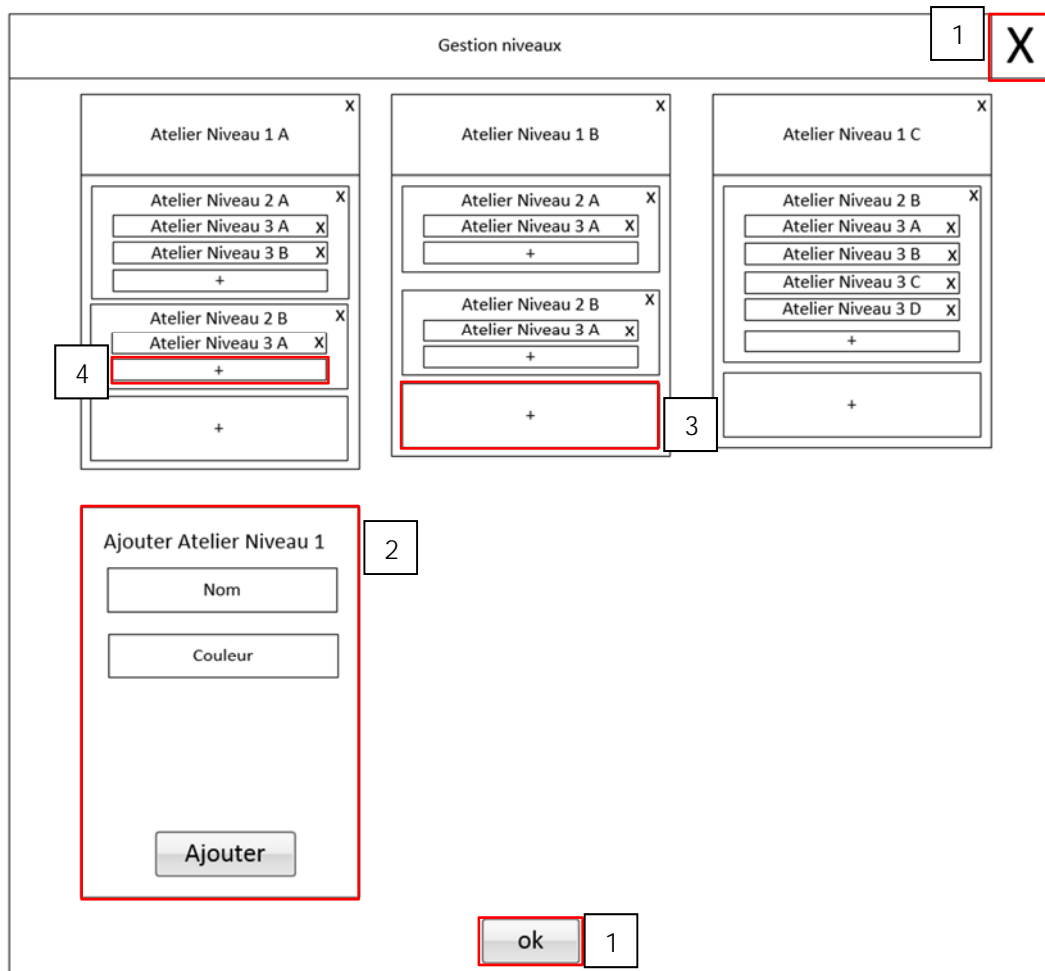
Description :

- 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 a grid of workshop management panels. Each panel has a title (e.g., "Atelier Niveau 1 A"), a close button (X), and a list of sub-workshops (e.g., "Atelier Niveau 2 A", "Atelier Niveau 3 A", "Atelier Niveau 3 B"). Below the list is a "+" button. A red box highlights the "+" button in the "Atelier Niveau 2 A" panel, with a callout "4". Another red box highlights the "+" button in the "Atelier Niveau 1 B" panel, with a callout "3". A third red box highlights the "+" button in the "Atelier Niveau 2 B" panel, with a callout "4". A fourth red box highlights the "+" button in the "Atelier Niveau 1 A" panel, with a callout "4". At the bottom left, there is a form titled "Ajouter Atelier Niveau 1" with fields for "Nom" and "Couleur", and an "Ajouter" button. A red box highlights this form, with a callout "2". At the bottom right, there is an "ok" button, with a callout "1". At the top right, there is a close button (X) for the main window, with a callout "1".

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 Sociaux Professionnel
- 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 sociaux professionnels, sous forme de tableau simple, sans utiliser DataTables.
- 3) Formulaire pour ajouter un MSP à la liste

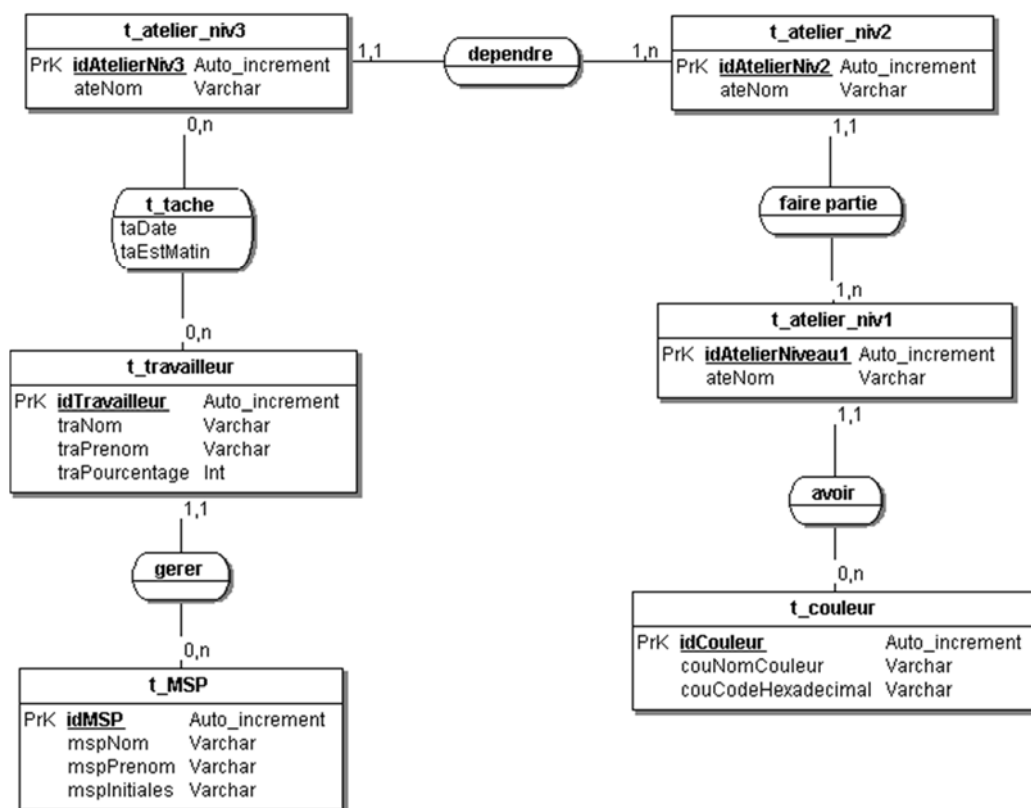
3.2.4 Base de données

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.

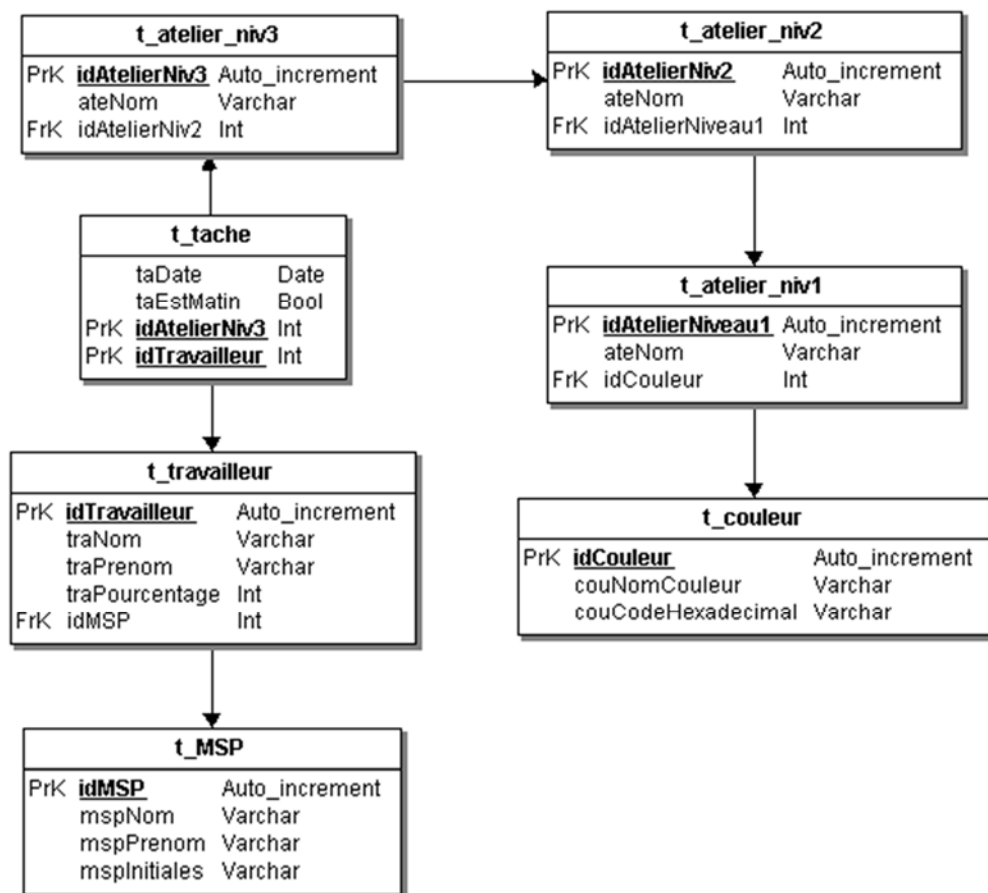
On y retrouve donc 5 tables basiques et une table intermédiaire.

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

- Ce paragraphe permet de spécifier la stratégie de test qui sera menée au point 5.1
- Qui, quand, avec quelles données, dans quel ordre, etc.
- Avec quels matériels, quels services, etc.

3.4 Planification détaillée

- A ce stade, après l'analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, ...) peut être finalisé.
- Le planning détaillé doit s'inscrire dans le planning initial. Il faut que l'on puisse situer cette planification détaillée par rapport à la planification initiale.

4 RÉALISATION

4.1 Dossier de Réalisation

4.1.1 Mise en place de l'environnement

4.1.2 Création de la page « Travailleurs »

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

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 pour supprimer le travailleur

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

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.

- Cette partie permet de reproduire ou reprendre le projet par un tiers.
- Pour chaque étape, il faut décrire sa mise en œuvre. Typiquement :
 - Versions des outils logiciels utilisés (OS, applications, pilotes, librairies, etc.)
 - Configurations spéciales des outils (Equipements, PC, machines, outillage, etc.)
 - Code source commenté des éléments logiciels développés.
 - Modèle physique d'une base de données.
 - Arbrescences des documents produits.
 - Schémas, plans d'adressages, plan de nommage, etc.
- Il faut décrire le parcours de réalisation et justifier les choix.

4.2 Modifications

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

5 TESTS

5.1 Dossier des tests

- On dresse le bilan des tests effectués (qui, quand, avec quelles données...) sous forme de procédure. Lorsque cela est possible, fournir un tableau des tests effectués avec les résultats attendus et obtenus, ainsi que les actions à entreprendre en conséquence (et une estimation de leur durée).
- Si des tests prévus dans la stratégie n'ont pas pu être effectués :
 - raison, décisions, etc.
- Liste des bugs répertoriés avec la date de découverte et leur état :
 - Corrigé, date de correction, corrigé par, etc.

6 CONCLUSION

6.1 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.

6.2 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.

6.3 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.

7 DIVERS

7.1 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.

7.2 Bibliographie

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

7.3 Webographie

- Références des sites Internet consultés durant le projet.

8 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.

8.1 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);
```