

Gestion de projet

Genoud Malorie, Lacombe Mickaël, Sejmenovic Aida, CPNV

Contents

1	Résumé	1
2	Introduction	1
3	Corps du rapport	2
3.1	Cahier des charges	2
3.2	Mise en place du projet	3
3.3	Théorie	3
3.4	Liste des tests à effectuer	8
3.5	Résultats des tests	10
3.6	Erreurs restantes	12
3.7	Fonctionnalités accomplies	13
4	Pré-requis	14
5	Conclusion	15
6	Références	15
7	Annexes	15

1 Résumé

Ce rapport décrit la réalisation technique d'une plateforme web, permettant de créer divers projets, d'assigner des utilisateurs à des projets et à des tâches ainsi qu'une gestion des tâches.

2 Introduction

Dans le cadre du CPNV dans le module MAW2 (Module d'Application Web 2), nous devons concevoir une plateforme Web de gestion de projet à l'aide d'un Framework. Elle contiendra

le planning initial et final d'un projet ainsi qu'une partie gestion des tâches permettant de construire le journal de bord.

Le site permettra de mieux gérer toute la partie planning et gestion des tâches dans un quelconque projet, peu importe le cadre dans lequel il est fait.

Ce site sera une partie intégrante de l'Intranet du CPNV afin d'offrir cette gestion à tous les utilisateurs, qu'ils soient "élève" ou "professeur".

Il y a divers buts que nous devons atteindre. Il devra permettre:

- De créer un nouveau projet
- D'inviter des membres pour ce projet
- D'ajouter des nouvelles tâches qu'elles soient "parente", "enfant" ou simplement "racine"
- De démarrer et stopper une tâche
- De supprimer ou éditer une tâche
- D'ajouter un ou plusieurs utilisateurs à une tâche
- D'afficher le planning en fonction des tâches.

3 Corps du rapport

3.1 Cahier des charges

3.1.1 Fonctionnalités principales

- Les tâches auront une durée en heure dans le planning.
- Attribution des tâches sur plusieurs personnes (les personnes du groupe peuvent s'ajouter à une tâche en cours par une autre personne du groupe).
- L'élève pourra voir toutes ses tâches sous forme de liste.
- Dans le journal de travail on récoltera le nom de la tâche, la durée par le planning et sa véritable durée pour permettre de comparer.
- Dans les tâches, l'élève pourra « commencer », « arrêter » et « reprendre » une tâche (l'application devra traquer quand il a commencé et arrêté). La tâche aura donc plusieurs « début » et « fin ».
- L'élève peut avoir un historique de ses tâches pour voir par exemple les dernières tâches effectuées, il ne peut pas les modifier, s'il veut les modifier il doit demander au professeur.
- Chaque tâche peut avoir des commentaires par la personne qui a la tâche ou par le professeur.
- Mettre un champ de recherche pour les commentaires et tâches.

3.1.2 Fonctionnalités secondaires

- Les tâches sont modifiables/supprimables que par les élèves et les profs même en cours du projet (planification).
- Les tâches et le projet ne seront pas visibles par les autres groupes de personnes.
- Il existera donc 2 types d'utilisateurs :
 - Élèves
 - Professeurs

- Le professeur devra pouvoir voir en temps réel comment les élèves activent et désactivent leur tâches comme un historique. On retrouvera donc le nom de l'élève et sa tâche avec la durée.

3.1.3 Fonctionnalités supplémentaire

Ces fonctionnalités ont été discutées avec M. Carrel Xavier. Cependant, nous n'allons pas implémenter toutes les fonctionnalités que M. Carrel nous a demandées. Après discussion, nous avons décidé d'implémenter:

- Une liste d'objectifs généraux. Chaque objectif général est constitué d'un énoncé, accompagné d'une checklist de critères qui permettront de dire si l'objectif est atteint ou non
- Un journal de bord, qui contient les événements marquants du projet. Il est éditable (ajout uniquement) par n'importe quel membre du projet.

3.2 Mise en place du projet

3.2.1 Outils

Pour la réalisation de notre projet, nous avons utilisé les outils suivants:

- Microsoft Viso 2010
- MySQL Workbench 6.3
- JetBrains PhpStorm 2016.1
- Laravel 5.2
- Trello
- Github => https://github.com/MalorieGenoud/Gestion_Projet_MAW2.git
- Xampp v3.2.2
- Wamp v3.0.0

3.3 Théorie

Cette partie contient l'explication des **Route**, des **Controller**, des **Middleware** et des **Models** que nous avons créés tout le long de notre projet.

3.3.1 Route

Nous avons structuré nos différentes **Route** en deux blocs. Le premier bloc gère tout ce qui concerne l'authentification d'un utilisateur:

```
Route::get('login', 'SessionController@create');
```

La première **Route** permet de renvoyer à l'utilisateur sur la vue de **Login** en appelant le **Controller SessionController@create** si celui-ci n'est pas connecté, d'où l'utilisation de la méthode **GET** car dans ce cas-ci on prend des données via l'url.

```
Route::post('login', 'SessionController@store');
```

La deuxième **Route** permet de contrôler la validité de l'e-mail et le mot de passe par rapport à la base de données en appelant le **Controller SessionController@store**, d'où l'utilisation de la méthode **POST** car dans ce cas-ci en renvoie des données.

Le deuxième bloc regroupe toutes les **Route** nécessaire à toute la conception de la gestion de projet:

```
Route::group(['middleware' => 'auth'], function(){
    ...
})
```

Cette **Route** regroupe toutes les autres (d'où l'utilisation de la méthode **GROUP** et permet à un utilisateur connecté d'accéder au reste du site.

3.3.1.1 Route Task La **Route Task** contient les méthodes suivantes:

- **TaskController@show** => D'afficher les détails de la tâche sélectionnée
- **TaskController@create** => De retourner la vue de création des tâches simple ou parentes
- **TaskController@store** => De créer des tâches simple ou parentes, de les mettre à jour
- **TaskController@createChildren** => De retourner la vue de création des tâches enfants
- **TaskController@storeChildren** => De créer des tâches enfants
- **TaskController@destroy** => De supprimer des tâches parents et/ou enfants
- **TaskController@edit** => D'éditer des tâches
- **TaskController@play** => De commencer une tâche
- **TaskController@stop** => De stopper une tâche
- **TaskController@users** => D'afficher les utilisateurs ayant une ou des tâches en communs
- **TaskController@storeUsers** => D'ajouter un ou plusieurs utilisateurs à une tâche
- **TaskController@userTaskDelete** => De supprimer un utilisateur à une tâche

3.3.1.2 Route Project La **Route Project** contient les méthodes suivantes:

- **ProjectController@index** => De retourner la vue des projets
- **ProjectController@show** => D'afficher les projets ainsi que les utilisateurs du projet
- **ProjectController@edit** => De retourner la vue d'édition des projets
- **ProjectController@task** => De retourner la vue des tâches du projet sélectionné
- **ProjectController@create** => De retourner la vue de création des projets
- **ProjectController@store** => De créer un ou plusieurs projets
- **ProjectController@createTask** => De retourner la vue de création d'une tâche
- **ProjectController@storeTask** => De créer une ou plusieurs tâches
- **ProjectController@destroyUser** => De supprimer un ou plusieurs utilisateurs d'un projet
- **ProjectController@storeTarget** => De créer un objectif
- **ProjectController@validateTarget** => De valider un objectif
- **ProjectController@getTarget** => De retourner la vue des objectifs

3.3.1.3 Route App La **Route App** contient la méthode suivante:

- **SessionController@destroy** => De détruire la session actuelle quand l'utilisateur se déconnecte

3.3.1.4 Route Invitation Projects La **Route Invitation Projects** contient les méthodes suivantes:

- **InvitationController@show** => De retourner la vue des invitations
- **InvitationController@store** => De créer une nouvelle invitation
- **InvitationController@wait** => De retourner la vue des invitations en attentes
- **InvitationController@edit** => De retourner la vue d'édition des invitations
- **InvitationController@** => De mettre à jour les invitations en *Accept* et d'ajouter l'utilisateur au(x) projet(s)
- **InvitationController@refuse** => De mettre à jours les invitations en *Refuse*

3.3.1.5 Route Event La **Route Event** contient les méthodes suivantes:

- **EventController@show** => De retourner la vue des événements
- **EventController@store** => D'ajouter un événement
- **EventController@formEvent** => Retourne la vue du formulaire d'ajout d'un événement

3.3.1.6 Route Comment La **Route Comment** contient les méthodes suivantes:

- **CommentController@show** => De retourner la vue des commentaires
- **CommentController@store** => D'ajouter un commentaire sur une tâche

3.3.1.7 Route User La **Route User** contient les méthodes suivantes:

- **UserController@show** => De retourner la vue des informations de l'utilisateur
- **UserController@storeAvatar** => D'ajouter un avatar à l'utilisateur

3.3.1.8 Route File La **Route File** contient la méthode suivante:

- **FileController@show** => De retourner la vue des fichiers
- **FileController@store** => D'ajouter des fichiers

3.3.2 Controller

Nous avons créé différents **Controller** pour gérer les différentes fonctionnalités de notre site:

- **TaskController**
- **ProjectController**
- **InvitationController**

- **UserController**
- **PlanningController**
- **CommentController**
- **EventController**
- **FileController**

Nous avons modifié un **Controller** déjà existant en rajoutant une méthode:

- **SessionController** => Ajout de la méthode **store**

Les différentes méthodes des **Controller** ont été expliquées dans la partie **Route**.

3.3.3 Middleware

3.3.3.1 ProjectControl Ce **Middleware** permet que lorsqu'un utilisateur s'authentifie sur le site, celui-ci va vérifier le rôle de l'utilisateur. Si son rôle est **Eleve**, il peut accéder aux fonctionnalités du site. Néanmoins si ce n'est pas le cas, il ne peut pas y accéder.

3.3.4 Models

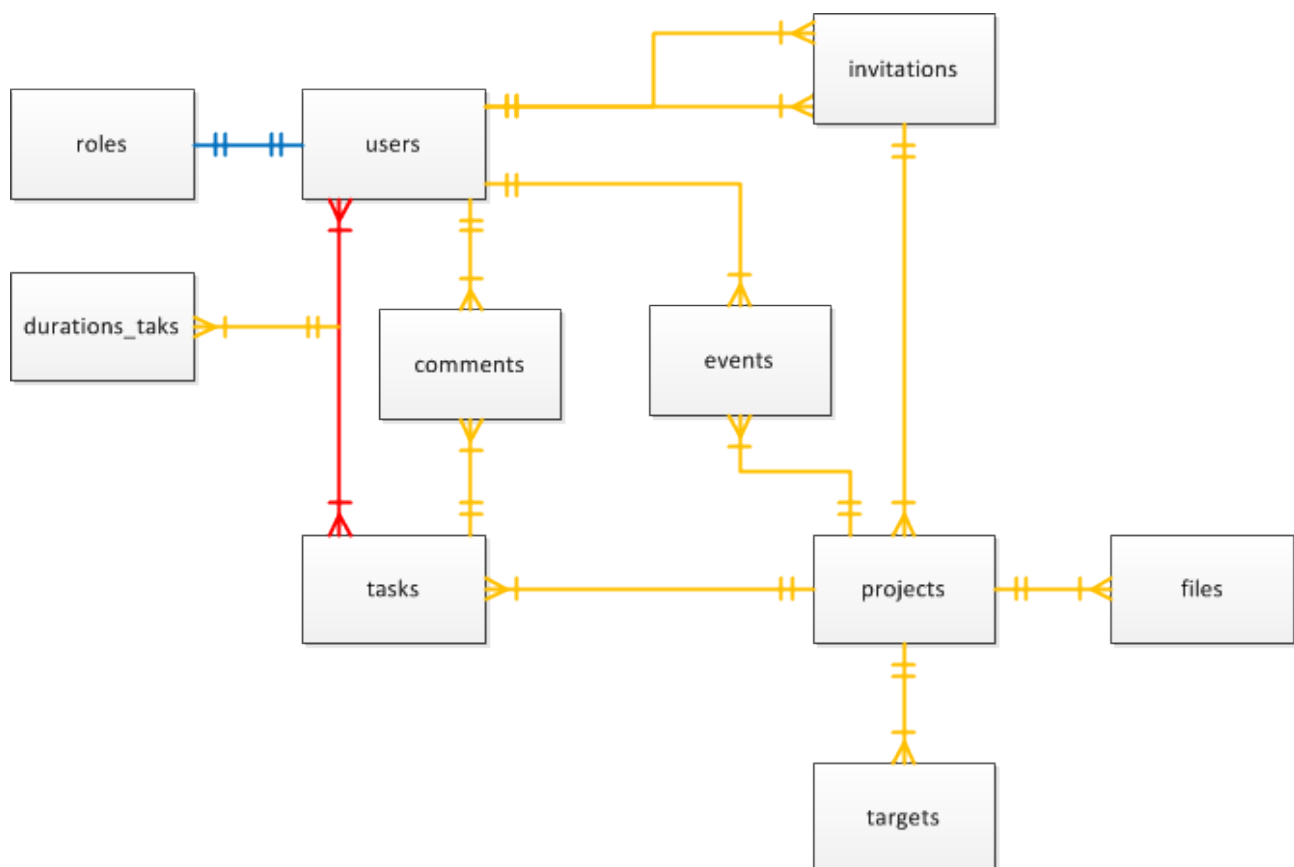


Figure 1: MCD

3.3.4.1 MCD

3.3.4.2 MLD Voir la partie **Annexe**

3.3.4.3 Migrations Nous avons créé pour chaque table un fichier de **Migration** afin de nous faciliter la mise à jour de la base de données étant donné que nous avons dû ajouter ou supprimer plusieurs champs dans les tables

3.3.5 Views

Nous avons différentes vues:

- La vue d'authentification d'un utilisateur
- La vue qui affiche tous les projets ainsi que les utilisateurs associés
- La vue de création d'un projet
- La vue des invitations en attente de l'utilisateur connecté avec la possibilité d'accepter ou de refuser une invitation
- La vue des informations propre à l'utilisateur avec l'ajout d'un avatar.
- La vue d'un des projet de l'utilisateur connecté comportant:
 - Le planning
 - Les tâches associées à l'utilisateur avec la possibilité de terminer, de commencer ou de stopper une tâche
 - La liste des tâches du projet contenant:
 - La création d'une tâche racine
 - L'ajout d'une tâche enfant à une tâche parente
 - L'édition d'une tâche
 - La suppression d'une tâche
 - L'ajout d'un ou plusieurs utilisateurs à une tâche
 - Les détails d'une tâche contenant:
 - La durée initiale
 - La date du jalon
 - Les différents rush avec les informations suivantes:
 - * L'affichage de la date de création du rush
 - * L'affichage de la date de fin du rush
 - * Le nom de l'utilisateur qui a effectué le rush
 - * La durée du rush
 - La liste des commentaires avec les informations suivantes:
 - * Le commentaire
 - * La date de création du commentaire
 - * Le nom de l'utilisateur qui a écrit le commentaire
 - L'ajout d'un commentaire
 - Les informations du projet contenant:
 - Le nom du projet
 - La date de début
 - Une description
 - Les membres du projet avec:
 - * Les informations des membres comme le nom et prénom ainsi que l'e-mail
 - * L'ajout d'un membre
 - * La suppression d'un membre
 - * Voir les invitations en attente, acceptées ou refusées

- La liste des événements majeurs ainsi que l’ajout d’un événement manuellement contenant:
- Le nom de celui qui a créé l’événement
- Le nom de l’événement
- La date de création
- La liste des fichiers ajoutés ainsi que la possibilité d’ajouter des fichiers

3.3.6 Plugin planning

Pour l’affichage du planning, nous avons utilisé un plugin de Google Charts (voir **Références**).

3.4 Liste des tests à effectuer

- L’authentification avec vérification de l’e-mail et du mot de passe
- La déconnexion
- Dans la vue global des projets:
 - L’affichage de la liste des projets avec:
 - La vue des projets propre à un utilisateur ayant pour rôle “Eleve”
 - La vue de tous les projets propre à un utilisateur ayant pour rôle “Prof”
 - L’affichage des utilisateurs associés au(x) projet(s)
 - L’accès à la vue détaillée du projet lorsqu’on clique sur le nom du projet
 - La création d’un projet
 - L’affichage d’un pop-up des invitations en attente de l’utilisateur connecté, avec la possibilité d’acceptée ou de refusée
 - La possibilité d’accepter ou de refuser une invitation
- Dans la vue détaillée du projet de utilisateur l’utilisateur connecté:
 - L’affichage correcte du planning
 - Dans la partie **Vos tâches**
 - L’affichage des tâches propres à l’utilisateur connecté
 - L’affichage de toutes les tâches racines et/ou parentes
 - L’affichage de l’avancement en pourcent de la tâche racine/parente/enfant (pour les tâches parentes, le pourcentage se fait en fonction des tâches enfants)
 - La possibilité de valider la fin d’une tâche
 - La possibilité de démarrer/stopper une tâche
 - L’affichage des tâches enfants d’une tâche parente
 - L’affichage du temps des tâches (pour les tâches parents, leur temps est calcul en fonction du temps de leurs enfants en plus de leur temps de base)
 - Dans la partie **Les tâches du projet**
 - L’affichage de toutes les tâches racines et/ou parents
 - L’affichage de l’avancement en pourcent de la tâche racine/parente/enfant (pour les tâches parentes, le pourcentage se fait en fonction des tâches enfants)
 - La possibilité de créer une tâche racine
 - La création d’une tâche enfant d’une tâche existante
 - L’édition d’une tâche racine/parente/enfant
 - La suppression d’une tâche

- L'ajout d'un ou plusieurs utilisateurs à une tâche sauf s'il y a déjà le ou les utilisateurs associés à la tâche
- L'affichage des tâches enfants d'une tâches
- L'affichage du temps des tâches (pour les tâches parents, leur temps est calcul en fonction du temps de leurs enfants en plus de leur temps de base)
- Dans la partie **Détail de la tâche**
- L'affichage du nom de la tâche avec:
 - * L'affichage de la durée initiale de la tâche
 - * L'affichage de la date du jalon (si y a une date de jalon)
- L'affichage des différents rush effectués sur la tâche avec:
 - * L'affichage de la date de création du rush
 - * L'affichage de la date de fin du rush
 - * Le nom de l'utilisateur qui a effectué le rush
 - * La durée du rush
- L'affichage des commentaires avec:
 - * Le commentaire
 - * La date de création du commentaire
 - * Le nom de l'utilisateur qui a écrit le commentaire
- L'ajout de commentaires
- Dans la partie **Informations du projet**
- L'affichage du nom du projet
- L'affichage du début du projet
- L'affichage de la description du projet
- L'affichage des différents membres du projet
- La suppression d'un membre
- L'ajout d'un membre non présent dans le projet mais avec une invitation en attente ou qui n'a pas encore été invité ou qui aurait pu refusé involontairement l'invitation
- L'affichage des invitations du projet ayant pour statut *en attente*, *refusée* ou *acceptée*
- Dans la partie **Événements majeurs**
- L'affichage des événements avec:
 - * La liste des événements
 - * La description de l'évènement
 - * La date de création
- L'ajout d'un événement manuellement
- Le champ de recherche d'un commentaire ou d'une tâche
- Dans la partie **Fichiers**
- La possibilité d'entrer un nom pour le fichier
- La possibilité de choisir tout type de fichiers
- La possibilité d'envoyer le fichier
- La possibilité de voir le fichier envoyé avec:
 - * Le nom originel du fichier
 - * Le nouveau nom du fichier
 - * La taille du fichier
- Le champ de recherche d'un commentaire ou d'une tâche
- Dans la partie **Informations sur l'utilisateur**
 - L'affichage du nom de l'utilisateur

- L’affichage correcte de son avatar (s’il l’a déjà ajouté)
- La recherche d’un avatar depuis le bouton **Parcourir...**
- L’ajout d’un avatar depuis le bouton **Envoyer**
- L’affichage de l’e-mail de l’utilisateur
- L’affichage du rôle de l’utilisateur (**Élève** ou **Prof**)

3.5 Résultats des tests

- L’authentification avec vérification de l’e-mail et du mot de passe => **FONCTIONNELLE À 100%**
- La déconnexion => **FONCTIONNELLE À 100%**
- Dans la vue global des projets:
 - L’affichage de la liste des projets avec:
 - La vue des projets propre à un utilisateur ayant pour rôle “Eleve” => **FONCTIONNELLE À 100%**
 - La vue de tous les projets propre à un utilisateur ayant pour rôle “Prof” => **FONCTIONNELLE À 100%**
 - L’affichage des utilisateurs associés au(x) projet(s) => **FONCTIONNELLE À 100%**
 - L’accès à la vue détaillée du projet lorsqu’on clique sur le nom du projet => **FONCTIONNELLE À 100%**
 - La création d’un projet => **FONCTIONNELLE À 100%**
 - L’affichage d’un pop-up des invitations en attente de l’utilisateur connecté, avec la possibilité d’acceptée ou de refusée => **FONCTIONNELLE À 100%**
 - La possibilité d’accepter ou de refuser une invitation => **FONCTIONNELLE À 100%**
- Dans la vue détaillée du projet de utilisateur l’utilisateur connecté:
 - L’affichage correcte du planning => **CHANGEZ DE PLUGIN**
 - Dans la partie **Vos tâches**
 - L’affichage des tâches propres à l’utilisateur connecté => **FONCTIONNELLE À 100%**
 - L’affichage de toutes les tâches racines et/ou parentes => **FONCTIONNELLE À 100%**
 - L’affichage de l’avancement en pourcent de la tâche racine/parente/enfant (pour les tâches parentes, le pourcentage se fait en fonction des tâches enfants) => **FONCTIONNELLE À 100%**
 - La possibilité de valider la fin d’une tâche => **FONCTIONNELLE À 50%**
 - La possibilité de démarrer/stopper une tâche => **FONCTIONNELLE À 100%**
 - L’affichage des tâches enfants d’une tâche parente => **FONCTIONNELLE À 100%**
 - L’affichage du temps des tâches (pour les tâches parents, leur temps est calcul en fonction du temps de leurs enfants en plus de leur temps de base) => **FONCTIONNELLE À 100%**
 - Dans la partie **Les tâches du projet**
 - L’affichage de toutes les tâches racines et/ou parents => **FONCTIONNELLE À 100%**

- L’affichage de l’avancement en pourcent de la tâche racine/parente/enfant (pour les tâches parentes, le pourcentage se fait en fonction des tâches enfants) => **FONCTIONNELLE À 100%**
- La possibilité de créer une tâche racine => **FONCTIONNELLE À 100%**
- La création d’une tâche enfant d’une tâche existante => **FONCTIONNELLE À 100%**
- L’édition d’une tâche racine/parente/enfant => **FONCTIONNELLE À 100%**
- La suppression d’une tâche => **FONCTIONNELLE À 100%**
- L’ajout d’un ou plusieurs utilisateurs à une tâche sauf s’il y a déjà le ou les utilisateurs associés à la tâche => **FONCTIONNELLE À 100%**
- L’affichage des tâches enfants d’une tâches => **FONCTIONNELLE À 100%**
- L’affichage du temps des tâches (pour les tâches parents, leur temps est calcul en fonction du temps de leurs enfants en plus de leur temps de base) => **FONCTIONNELLE À 100%**
- Dans la partie **Détail de la tâche**
- L’affichage du nom de la tâche avec:
 - * L’affichage de la durée initiale de la tâche => **FONCTIONNELLE À 100%**
 - * L’affichage de la date du jalon (si y a une date de jalon) => **FONCTIONNELLE À 100%**
- L’affichage des différents rush effectués sur la tâche avec:
 - * L’affichage de la date de création du rush => **FONCTIONNELLE À 100%**
 - * L’affichage de la date de fin du rush => **FONCTIONNELLE À 100%**
 - * Le nom de l’utilisateur qui a effectué le rush => **FONCTIONNELLE À 100%**
 - * La durée du rush => **FONCTIONNELLE À 100%**
- L’affichage des commentaires avec:
 - * Le commentaire => **FONCTIONNELLE À 100%**
 - * La date de création du commentaire => **FONCTIONNELLE À 100%**
 - * Le nom de l’utilisateur qui a écrit le commentaire => **FONCTIONNELLE À 100%**
- L’ajout de commentaires => **FONCTIONNELLE À 100%**
- Dans la partie **Informations du projet**
- L’affichage du nom du projet => **FONCTIONNELLE À 100%**
- L’affichage du début du projet => **FONCTIONNELLE À 100%**
- L’affichage de la description du projet => **FONCTIONNELLE À 100%**
- L’affichage des différents membres du projet => **FONCTIONNELLE À 100%**
- L’ajout d’un membre non présent dans le projet mais avec une invitation en attente ou qui n’a pas encore été invité ou qui aurait pu refusé involontairement l’invitation => **FONCTIONNELLE À 100%**
- L’ajout d’un membre => **FONCTIONNELLE À 100%**
- L’affichage des invitations du projet ayant pour statut *en attente, refusée* ou *acceptée* => **FONCTIONNELLE À 100%**
- Dans la partie **Événements majeurs**
- L’affichage des événements avec:
 - * La liste des événements => **FONCTIONNELLE À 100%**
 - * La description de l’évènement => **FONCTIONNELLE À 100%**
 - * La date de création => **FONCTIONNELLE À 100%**

- L'ajout d'un événement manuellement => **FONCTIONNELLE À 50%**
- Le champ de recherche d'un commentaire ou d'une tâche => **FONCTIONNELLE À 100%**
- Dans la partie **Fichiers**
 - La possibilité d'entrer un nom pour le fichier => **FONCTIONNELLE À 100%**
 - La possibilité de choisir tout type de fichiers => **FONCTIONNELLE À 100%**
 - La possibilité d'envoyer le fichier => **FONCTIONNELLE À 100%**
 - La possibilité de voir le fichier envoyé avec:
 - * Le nom originel du fichier => **FONCTIONNELLE À 100%**
 - * Le nouveau nom du fichier => **FONCTIONNELLE À 100%**
 - * La taille du fichier => **FONCTIONNELLE À 100%**
- Le champ de recherche d'un commentaire ou d'une tâche => **FONCTIONNELLE À 100%**
- Dans la partie **Informations sur l'utilisateur**
 - L'affichage du nom de l'utilisateur => **FONCTIONNELLE à 100%**
 - L'affichage correcte de son avatar (s'il l'a déjà ajouté) => **FONCTIONNELLE À 100%**
 - La recherche d'un avatar depuis le bouton **Parcourir...** => **FONCTIONNELLE à 100%**
 - L'ajout d'un avatar depuis le bouton **Envoyer** => **FONCTIONNELLE À 100%**
 - L'affichage de l'e-mail de l'utilisateur => **FONCTIONNELLE à 100%**
 - L'affichage du rôle de l'utilisateur (**Élève** ou **Prof**) => **FONCTIONNELLE À 100%**

3.6 Erreurs restantes

Voici la liste des erreurs restantes:

- L'affichage correcte du planning
 - Concernant le plugin pour le planning, nous avons au début utilisé le plugin de Google Charts (voir dans **Références**), cependant, celui-ci n'intégrait pas la gestion du temps des tâches. Nous avons donc décidé de laisser ce plugin de côté et de corriger le reste des bugs restants et de terminer les fonctionnalités restantes.
- La possibilité de valider la fin d'une tâche
 - Pour la possibilité de valider une tâche, nous n'avons fait qu'un message pop-up s'affiche pour savoir si l'on peut valider ou non la tâche, notamment dans le cas où si les tâches enfants d'une tâche parente ne sont pas finis, l'utilisateur ne peut pas valider la tâche parente.
- L'ajout d'un événement manuellement
 - Pour l'ajout d'un événement manuellement, nous nous sommes rendu compte que nous avons besoin d'un **Observer** afin de pouvoir gérer le fait que non-seulement des événements s'ajoutent, par exemple lorsque l'on crée une tâche, mais aussi de pouvoir les ajouter manuellement. Nous avons donc décidé de laissez seulement l'ajout d'événement lorsque l'on crée une tâche car nous nous sommes rendus compte trop tard de ce que nous avons besoin et nous avons préféré corriger les quelques bugs que nous avons

3.7 Fonctionnalités accomplies

3.7.1 Fonctionnalités principales

- Les tâches auront une durée en heure dans le planning. => **PAS ACCOMPLIE**
- Attribution des tâches sur plusieurs personnes (les personnes du groupe peuvent s'ajouter à une tâche en cours par une autre personne du groupe). => **ACCOMPLIE À 100%**
- L'élève pourra voir toutes ses tâches sous forme de liste. => **ACCOMPLIE À 100%**
- Dans le journal de travail on récoltera le nom de la tâche, la durée par le planning et sa véritable durée pour permettre de comparer. => **PAS ACCOMPLIE**
- Dans les tâches, l'élève pourra « commencer », « arrêter » et « reprendre » une tâche (l'application devra traquer quand il a commencé et arrêté). La tâche aura donc plusieurs « début » et « fin ». => **ACCOMPLIE À 100%**
- L'élève peut avoir un historique de ses tâches pour voir par exemple les dernières tâches effectuées, il ne peut pas les modifier, s'il veut les modifier il doit demander au professeur. => **ACCOMPLIE À 80%**
- Chaque tâche peut avoir des commentaires par la personne qui a la tâche ou par le professeur. => **ACCOMPLIE À 100%**
- Mettre un champ de recherche pour les commentaires et tâches. **ACCOMPLIE À 100%**

3.7.2 Fonctionnalités secondaires

- Les tâches sont modifiables/supprimables que par les élèves et les profs même en cours du projet (planification). => **ACCOMPLIES À 100%**
- Les tâches et le projet ne seront pas visibles par les autres groupes de personnes => **ACCOMPLIE À 100%**
- Il existera donc 2 types d'utilisateurs : => **ACCOMPLIE À 100%**
 - Élèves
 - Professeurs
- Le professeur devra pouvoir voir en temps réel comment les élèves activent et désactivent leur tâches comme un historique. On retrouvera donc le nom de l'élève et sa tâche avec la durée. => **ACCOMPLIES À 50%**

3.7.3 Fonctionnalités supplémentaires

- Une liste d'objectifs généraux. Chaque objectif général est constitué d'un énoncé, accompagné d'une checklist de critères qui permettront de dire si l'objectif est atteint ou non. => **ACCOMPLIE À 100%**
- Un journal de bord, qui contient les événements marquants du projet. Il est éditable (ajout uniquement) par n'importe quel membre du projet. => **ACCOMPLIE À 50%**

3.7.4 Résumé des fonctionnalités pas ou quasi accomplies

3.7.4.1 Fonctionnalités principales

- Dans le journal de travail on récoltera le nom de la tâche, la durée par le planning et sa véritable durée pour permettre de comparer.

- Nous n'avons pas fait cette fonctionnalités car nous avons préféré terminer le reste des fonctionnalités que nous avons jugé plus importante que celle-ci.
- L'élève peut avoir un historique de ses tâches pour voir par exemple les dernières tâches effectuées, il ne peut pas les modifier, s'il veut les modifier il doit demander au professeur.
 - Nous nous sommes occupés seulement de la partie affichage des différents rush sur une tâche, car nous avons décidé de mettre de côté la gestion des droits et de les gérer seulement dans le cas où le reste des fonctionnalités seraient finies.

Fonctionnalités secondaires

- Le professeur devra pouvoir voir en temps réel comment les élèves activent et désactivent leur tâches comme un historique. On retrouvera donc le nom de l'élève et sa tâche avec la durée.
 - Il est possible de voir les différents rush des élèves, cependant on ne peut pas les voir en temps réel car pour les voir il faut cliquer sur chaque tâche pour voir les rush.

Fonctionnalités supplémentaires

- Un journal de bord, qui contient les événements marquants du projet. Il est éditable (ajout uniquement) par n'importe quel membre du projet.
 - Comme dit dans la partie **Erreurs restantes**, nous avons besoin d'un **Observer** afin de pouvoir gérer la partie de l'ajout manuellement. Seule la partie des ajouts automatiques des événements fonctionne.

4 Pré-requis

Voici la marche à suivre pour l'installation du projet:

1. Vérifiez dans un premier temps que vous disposez des logiciels suivants:
 - Wamp ou Xampp pour l'hébergement du site ainsi que de la base de données
 - Github pour faire un clone du repository
2. Allez Dans le dossier **www** ou **htdocs**, faites un clique droit -> Git Bash here -> `git clone https://github.com/MalorieGenoud/Gestion_Projet_MAW2.git`
3. Faites un `php artisan migrate --seed` afin de faire l'installation de la base de données. Vous pouvez aussi utiliser le fichier **bones.sql** et l'importer dans votre serveur.
4. Faites ensuite `git checkout developement` afin d'accéder au site. Faites un `composer update` afin d'être sûr que tout soit à jour par rapport au Framework Laravel.
5. Vous pouvez maintenant utiliser le site.
 1. Login et mot de passe des utilisateurs avec pour rôle **Eleve**:
 - Mickaël Lacombe : **lacombe@cpnv.cv** ; **lacombe123**
 - Aida Sejmenovic : **sejmenovic@cpnv.ch** ; **sejmenovic123**
 - Malorie Genoud : **genoud@cpnv.ch** ; **genoud123**
 2. Login et mot de passe des utilisateurs avec pour rôle **Prof**:

- Pascal Hurni : hurni@cpnv.ch ; **hurni123**

Toutefois, si vous constatez des erreurs liés à la base de données, supprimez-la, faites un `php artisan migrate` sans la méthode `--seed` via GitHub là où se situe votre projet (exemple `C:\xampp\htdocs\Gestion_Projet_MAW2`), puis importez le fichier `insertion.sql` afin de récupérer les données de base.

Si vous ne pouvez pas ajouter de fichiers, allez dans le fichier `php.ini` de Wamp ou Xampp, décommentez la ligne `extension=php_fileinfo.dll` (pour Windows) ou `php_value extension fileinfo.so` (pour Mac ou Linux). Pensez à redémarrer votre serveur pour prendre en compte les changements.

Si d'autres soucis se présentent, vous pouvez prendre contact avec:

- Mickaël Lacombe : mickaël.lacombe@cpnv.ch
- Aida Sejmenovic : aida.sejmenovic@cpnv.ch
- Malorie Genoud : malorie.genoud@cpnv.ch

5 Conclusion

Nous avons accompli en grande partie la quasi totalité des fonctionnalités comme la création d'un projet ainsi que de tâches, l'ajout de plusieurs utilisateurs pour une tâche, ainsi de suite.

Concernant les objectifs que nous nous étions fixés, il y a seulement un seul que nous n'avons pas pu atteindre. C'est celui concernant l'affichage du planning. Comme dit plus haut, ceci est dû au fait que nous avons utilisé un plugin pas adapté à nos besoins et que nous avons préféré mettre ceci de côté et de se concentrer sur le reste.

Concernant le projet de manière général, nous avons eu quelques difficultés au départ à savoir comment nous allions organiser tout notre affichage afin que tout soit un minimum ergonomique étant donné que nous n'avons qu'un simple design de base.

6 Références

[Champ de recherche]<https://metrogeek.fr/ajouter-recherche-application-laravel/>

[Plugin Google Charts]<https://developers.google.com/chart/interactive/docs/gallery/ganttchart#events>

[Google charts possiblement utilisable] <https://developers.google.com/chart/interactive/docs/gallery/timeline>

[Documentation Laravel]<https://laravel.com/>

[Autres liens]<https://scotch.io/tutorials/debugging-queries-in-laravel>

7 Annexes

Pour voir le MLD, allez dans le dossier `MCD_MLD` -> `MLD.png`