

Mariama Diallo

Maxime Leroux

Hossam Ouldsouilem

Licence 3 Informatique Fondements et Applications

2020/2021



Rapport de stage

projet Agora

Encadrant : Jean-Philippe Babau, Professeur

à l'université de Bretagne occidentale.

Sommaire

Remerciements

- I. Introduction
- II. Projet global
 - 1) Objectifs
 - 2) utilisateurs
 - 3) fonctionnalités
- III. Environnement technique
 - 1) wamp
 - 2) Codeigniter
 - 3) Composer
 - 4) PHPdoc
 - 5) PHPUnit
- IV. Pmwiki
 - 1) Présentation
 - 2) Téléchargement et installation
 - 3) Initialisation
 - 4) établissement de la version française
 - 5) utilisation
 - a) Édition de base
 - b) Recette Cookbook
 - c) Liens et redirections
 - d) intégration application web
- V. Propositions
 - 1) Fonctionnalités
 - a) Ajout d'une proposition
 - b) Édition d'une proposition
 - c) Suppression d'une proposition
 - 2) Architecture logicielle
 - 3) Modèle de base de données
- VI. Résultats
 - 1) Département
 - 2) Groupe
 - 3) Proposition
- VII. Conclusion

Remerciements

**Nous tenons à remercier M. Jean-Philippe Babau, enseignant à l'UBO et encadrant de notre
Projet, de nous avoir offert la possibilité de faire ce projet et de nous avoir accompagné au
cour de celui-ci.**

I) Introduction

Avec la présence de la covid19 les offres de stages n'ont pas permis à tout le monde de trouver une entreprise pour effectuer un stage dans le but de valider la licence 3 informatique. C'est dans ce contexte que des projets ont été proposés par les professeurs pour remplacer le stage. Nos choix portant sur le même projet, on devait travailler ensemble du 12 mai au 04 juin 2021 dans le but développer une application web (type wiki) pour faciliter l'élaboration collective de propositions au sein d'une association, communauté... .

À travers ce rapport nous allons voir comment cette application a été développée en expliquant le projet global, l'environnement technique, pmwiki, les propositions et les résultats.

II) Le projet Agora

1) objectif

Le projet Agora a pour objectif de mettre en place un outil web qui permet de faciliter la mise au point de propositions dans un mode collaboratif pour des communautés, des associations et des organisations.

2) Les utilisateurs

Pour le projet Agora on a un seul type d'utilisateurs pour gérer les propositions.

3) Les fonctionnalités à développer

Pour les fonctionnalités à développer on a la mise au point de propositions qui sont composées de :

- **d'une description:** texte court
- **d'une résolution:** texte modifiable
- **d'un avis général:** texte
- **d'une liste de corrections** demandée qui possède un numéro, un descriptif(texte) et un état qui est «à faire » ou « fait »
- **d'une réponse:** texte.

Ces fonctionnalités s'accompagnent d'un certain nombre d'exigences. Ces dernières sont :

en créant une proposition la description est dans l'état débloqué à la version 1 ;

à part la description qui est un champ à remplir lors de la création tous les champs doivent être vides.

De plus une description est limitée à 80 caractères.

Pour développer ces fonctionnalités deux étapes qui ont durées chacune deux semaines ont dû être respectées.

La première étape consistait à la prise en main des outils de développements et le choix de la librairie parmi pmwiki, mediawiki et phpwiki. Ainsi on devait:

Mettre en place de l'environnement de développement(visual studio, création d'un compte sur framagit pour héberger le projet, PHP objet, MySQL, Composer, codeigniter, PHPDoc, PHPUnit)

Comprendre et Mettre en place l'architecture MVC (models, views, controller),

créer le site « Hello World » qui permet de faire des opérations et effectuer des tests sur les outils de développement.

Installer et tester des librairies PHP pour créer des espaces de rédaction partagés.

fonctionnalités	dokuwiki	pmwiki	mediawiki	phpwiki
Ctrl Y Z	doc - Testé	doc - Testé	doc - Testé	erreur d'installation
Ctrl X C V	doc - Testé	doc - Testé	doc - Testé	
police de caractères		doc - Testé	doc - Testé	
taille		doc - Testé	F21 Testé	
gras, italique, souligné	doc - Testé	doc - Testé	doc-Téste	
insertion tableau	doc - Testé	doc - Testé	doc-Téste	
insertion image	doc - Testé	doc - Testé	doc-Téste	
couleur textes		doc - Testé	doc-Téste	
suivi de version		doc - Testé	doc-Téste	
gestion des commentaires		doc - Testé		
gestion utilisateurs "proposer des modifications"		doc - Testé	doc	
validation/refus des modifications				

vérification orthographe		doc - Testé		
intégration dans une page web				
controle acces ecriture			testé	
documentation installation		doc - Testé	doc-Téste	
documentation utilisation		doc - Testé	doc-Téste	
			F21 9 ou plus doc-Téste	
version PHP	7.2.34	PHP 4.3 or Later	PHP 7.4.0 à 7.4.2	
BD	système de fichiers	système de fichiers	My SQL 5.5.8, Maria DB, PostGreSQL 9.2	

Résultats des testes réalisés pour le choix du wiki

La deuxième et dernière étape est la mise en place du wiki.

Sur cette étape on doit pouvoir gérer la liste des propositions qui est affichée sur la première page. Ainsi pour le faire on a un icône d'ajout d'une proposition et un icône de suppression et d'édition pour chaque proposition.

Le clique sur l'icône d'édition devrait permettre d'obtenir une page dédiée à la proposition avec ces divers champs et le dernier numéro de version de la proposition.

La zone résolution est éditée en mode wiki :

La librairie choisie doit pouvoir permettre de :

- modifier le texte par une mise en forme
- insérer une image ou un tableau
- ajouter ou de supprimer des commentaires.

La librairie pmwiki répondant à toutes ces critères est celle qui a donc été choisie pour l'application.

De plus on doit pouvoir modifier une description, un avis et la réponse à une correction. Enfin, l'utilisateur peut :

- bloquer ou débloquer l'édition du champ résolution
- peut revenir via une page dédiée à une version antérieure de la résolution
- ajouter, supprimer ou éditer une correction : descriptif et fait/ à fait
- dérouler une correction pour afficher la réponse .

III) Environnement technique

1) WAMPserver

Wampserver est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache², du langage de script PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

Il existe des versions pour d'autres systèmes d'exploitation.


- LAMP (pour Linux)
- MAMP (pour mac Os)

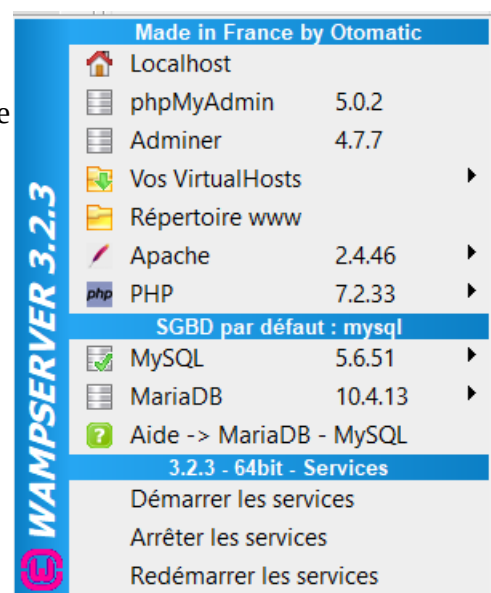
2) Téléchargement et installation

Pour ce projet, nous utilisons Wampserver.

Lien pour le téléchargement de WAMPserver : <https://www.wampserver.com/>

3) Présentation

Une fois Wampserver installé cette icône  apparaîtra dans la liste des icônes en bas à droite de votre écran. Cette icône peut être de 3 couleurs différentes, vert quand tout fonctionne, orange quand seulement certains services fonctionnent et rouge quand rien ne fonctionne.



1. En cliquant dessus vous verrez apparaître cette fenêtre.
2. Qui vous donnera les différentes versions pour les logiciels auxquels on a accès.
- 3.
4. Si les versions installées ne sont pas celles qui nous intéressent, il est possible d'en installer d'autres.
5. Les versions que nous voulions étaient :
6. PHP 7.2
7. MySQL 5.6

Lien pour télécharger d'autres versions : <https://wampserver.aviatechno.net/>

Addons Wampserver 3

Applications Toutes les versions disponibles

PhpMyAdmin

Phpmyadmin 4.9.7 MD5

PhpMyAdmin 4.9.7 - Dernière version supportée par PHP 5.5, 5.6 et 7.0. Cette version peut être ajoutée à une version existante, vous aurez le choix lors de l'installation.

Phpmyadmin 5.1.1 MD5

PhpMyAdmin 5.1.1 ne supporte pas PHP 5.5, 5.6 et 7.0. Supporté par PHP 7.1, 7.2, 7.3, 7.4 et 8. Cette version peut être ajoutée à une version existante, vous aurez le choix lors de l'installation.

Adminer

Gestion de base de données en un seul fichier PHP. Adminer (anciennement phpMinAdmin) est un outil léger de gestion complète de base de données écrit en PHP. Adminer fonctionne parfaitement avec PHP 7 & 8 et MySQL 5.7 & 8

Adminer 4.8.1 MD5

PhpSysInfo

Phpsysinfo 3.3.4 MD5

Apache Toutes les versions disponibles

Apache 32 bit x86 28 versions

Apache 2.4.48 32 bit x86 MD5

Apache 64 bit x64 26 versions

Apache 2.4.48 64 bit x64 MD5

Important: Avant d'installer Apache 2.4.48, la mise à jour Wampserver 3.2.5 ou plus est nécessaire. Il est également indispensable d'avoir préalablement installé les Visual C++ Redistributable Packages VC 2015-2019 (VS16) 14.29.30037 ou plus.

PHP Toutes les versions disponibles

PHP 32 bit x86 198 versions

PHP 5.6.40 32 bit x86 MD5

PHP 7.0.33 32 bit x86 MD5

PHP 7.1.33 32 bit x86 MD5

PHP 7.2.34 32 bit x86 MD5

PHP 7.3.28 32 bit x86 MD5

PHP 7.4.20 32 bit x86 MD5

PHP 8.0.7 32 bit x86 MD5

MySQL Toutes les versions disponibles

MySQL 32 bit x86 7 versions

MySQL 5.1.72 32 bit x86 MD5

MySQL 5.5.62 32 bit x86 MD5

MySQL 5.6.51 32 bit x86 MD5

MySQL 5.7.34 32 bit x86 MD5

MySQL 64 bit x64t 11 versions

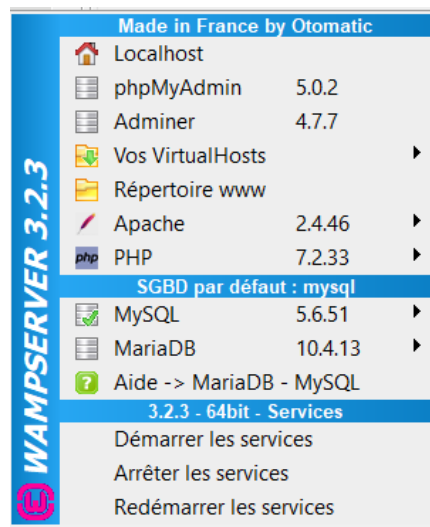
MySQL 5.5.62 64 bit x64 MD5

MySQL 5.6.51 64 bit x64 MD5

Sur ce site on pourra récupérer les addons pour WAMPserveur3 une fois les fichiers téléchargés il faudra les déposer à l'emplacement « .\wamp64\bin » dans le fichier correspondant à ce qui a été téléchargé.

apache	10/05/2021 16:19	Dossier de fichiers
mariadb	10/05/2021 16:19	Dossier de fichiers
mysql	10/05/2021 16:22	Dossier de fichiers
php	10/05/2021 16:20	Dossier de fichiers

Ensuite il faudra relancer Wampserver en faisant « redémarrer les services ».



VirtualHost

Le virtualHost servira à installer le site web en créant un nom de domaine pour la mise en place de l'application sur notre machine.

Nous pouvons aussi créer des virtualhost en allant sur vos virtualHost puis Gestion VirtualHost.

Nom du Virtual Host Pas d'espace - Pas de tiret bas (_) Requis

Chemin complet absolu du dossier VirtualHost - Exemples : C:/wamp/www/projet/ ou E:/www/site1/ Requis

Pour le nom du virtualHost il est possible de mettre ce que l'on veut et pour le chemin, il faudra mettre celui de l'application que vous créez.

2)codeigniter

Codeigniter est un framework Web écrit en PHP, qui vante une conception logicielle compacte rendant le développement d'applications Web plus rapide et plus efficace.

Nous utiliserons CodeIgniter 3 pour notre application.

Lien de téléchargement : <https://codeigniter.com/download>

Une fois codeigniter téléchargé mettez le fichier décompressé dans le répertoire .\wamp64\www\

cache	11/05/2021 09:12	Dossier de fichiers
config	11/05/2021 09:12	Dossier de fichiers
controllers	28/05/2021 14:32	Dossier de fichiers
core	11/05/2021 09:12	Dossier de fichiers
docs	13/05/2021 09:05	Dossier de fichiers
helpers	11/05/2021 09:12	Dossier de fichiers
hooks	11/05/2021 09:12	Dossier de fichiers
language	11/05/2021 09:12	Dossier de fichiers
libraries	11/05/2021 09:12	Dossier de fichiers
logs	04/06/2021 10:56	Dossier de fichiers
models	28/05/2021 14:42	Dossier de fichiers
tests	14/05/2021 08:36	Dossier de fichiers
third_party	11/05/2021 09:12	Dossier de fichiers
views	28/05/2021 14:47	Dossier de fichiers

En allant dans le dossier application, vous arriverez sur l'image ci-dessus les fichiers qui nous intéressent sont :

- config.php

Dans le dossier config il y aura le fichier config.php à compléter en mettant votre base URL.

```
$config['base_url'] =
```

- **database.php**

```
$db['default'] = array(
    'dsn' => '',
    'hostname' => 'agora',
    'username' => 'root',
    'password' => '',
    'database' => 'mydb',
);
```

hostname :mettre le nom du virtual host
 username :utilisateur phpMyAdmin
 password :mdp phpMyAdmin
 dataBase:nom de la base de donnée

3)Composer

A quoi ça sert ?

Composer est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin.

Installation

Pendant l'installation, lorsque l'on vous demandera de choisir un fichier PHP, il faudra aller dans le répertoire `.\wamp64\bin\php` et choisir la version 7.2

Ensuite dans l'invite de commande, il faudra tapé « composer » pour vérifier qu'il a bien été installé.

Site pour télécharger Composer :

(<https://getcomposer.org/download/>)

La configuration est à faire lorsque l'on crée un projet

Site pour la configuration du projet :

(<https://getcomposer.org/doc/01-basic-usage.md#composer-json-project-setup>)

4) phpdoc

A quoi ça sert ?

Phpdoc permettra de créer une documentation pour les projets PHP

Installation

Pour l'installer, il faut modifier le fichier composer.json et ajouter la ligne suivante

```
"require-dev": {  
    |  
    "phpdocumentor/phpdocumentor": "2.*"  
},
```

Dans l'invite de commande à l'emplacement de l'application « php composer.phar update »

```
/**  
 * permet de faire la multiplication entre deux nombres  
 * @param int $val1  
 * valeur choisie par l'utilisateur sur une liste deroulante  
 * @param int $val2  
 * valeur saisie par l'utilisateur  
 * @return int  
 */  
  
public function mult($val1,$val2)  
{  
    $result=$val1 * $val2;  
    return $result;  
}
```

Pour la phpdoc s'écrit au-dessus de la fonction entre « /** » et « */ ».

À la première « * » on fait une petite explication de la fonction

Ensuite, on met les « @param » le type en premier et le nom de la variable ensuite

Et à la fin, on fait un @return et on met le type de la valeur qui est retourné par la fonction

Une fois la documentation écrite, dans l'invite de commande dans le répertoire de l'application
entré la commande « >php phpDocumentor.phar -d controllers -t docs/api »

-d nom du dossier ou vous avez écrit la phpdoc

-t l'endroit ou l'on va créer la phpdoc

5) phpunit

phpunit c est quoi ?

PHPUnit est un framework open source de test unitaire dédié au langage de programmation PHP.

Qu'est ce qu'un test unitaire

Le principe d'un test unitaire est très simple : il s'agit d'exécuter du code provenant de l'application à tester et de vérifier que tout s'est bien déroulé comme prévu.

Cela signifie en général que vous allez vérifier que la valeur retournée est bien la valeur attendue et/ou que certaines méthodes ont bien été appelées.

Installation

Lien pour télécharger phpunit8: <https://phpunit.de/getting-started/phpunit-8.html>

Pensez a prendre la version de PHPUnit compatible avec votre version de PHP, dans notre cas ce sera PHPUnit8.

Dans le fichier composer.json situé à l'emplacement

```
> wamp64 > www > CodeIgniter
```

rajoutez la ligne ci-dessous

```
"require-dev": {  
    "phpunit/phpunit": "^8",
```

Dans l'invite de commande, tapez la ligne suivante à l'emplacement de l'application « php composer.phar update ».

Faire un test avec PHPUnit

```
<?php
use PHPUnit\Framework\TestCase;

class ComputationTest extends TestCase
{
    public function testadd()
    {
        $cpt = new Computation();
        $test = 3;
        $this->assertSame($test, $cpt->add(1 , 2));
    }
}
```

La ligne : « use PHPUnit\Framework\TestCase; » est à mettre au début de chaque fichier test pour utiliser la librairie de test puis le extends TestCase pour accéder a la librairie

Une règle à respecter : le nom de toutes vos méthodes de test doit être préfixé par « test ».

D’abord on appelle la méthode, puis ensuite on créer un exemple appelé test ici ensuite, on utilise le assertSame pour vérifier si la valeur teste est bien égale a la valeur renvoyée par la méthode

Dans le teste très simple ci-dessus, on vérifie que le résultat renvoyé par la fonction « add » est bien le résultat de l’addition.

IV) pmwiki

1) Présentation

PmWiki est un système de type [wiki](#) pour la création et l'entretien collectif de sites Internet. Les pages PmWiki ont le même aspect et fonctionnent comme des pages Internet ordinaires, sauf qu'elles possèdent un lien pour "[Éditer](#)" ce qui permet de modifier ou d'ajouter facilement (pas toujours) des pages à un site, en utilisant les règles d'[édition de base](#). Vous n'avez pas besoin de connaître le langage HTML ou les CSS. L'édition des pages peut être laissée ouverte à tout public ou restreinte à un petit groupe d'auteurs.

2) Téléchargement et Installation

- La version de PM wiki utilisé est 2.2.139 est téléchargeable via le lien : pmwiki-latest.zip.
- Décompresser l'archive et le placer dans le dossier Codeigniter













Nom	Modifié le	Type	Taille
 cookbook	22/05/2021 20:35	Dossier de fichiers	
 docs	05/05/2021 07:17	Dossier de fichiers	
 local	17/05/2021 17:08	Dossier de fichiers	
 pub	19/05/2021 15:41	Dossier de fichiers	
 scripts	19/05/2021 14:44	Dossier de fichiers	
 wiki.d	03/06/2021 23:13	Dossier de fichiers	
 wikilib.d	22/05/2021 20:51	Dossier de fichiers	
 xeslib.d	22/05/2021 20:51	Dossier de fichiers	
 images	21/05/2021 13:59	Fichier JPG	6 Ko
 pmwiki	09/04/2021 06:31	PHP Script File	99 Ko
 README	02/06/2006 15:13	Document texte	2 Ko
 test	19/05/2021 14:37	Document texte	1 Ko

Figure 1 Contenu de l'archive PM Wiki

README.txt	Un court fichier d'introduction
local/	Répertoire des fichiers de configuration locale
pmwiki.php	Code principal de PM Wiki
pub/	Répertoire des fichiers à accès public
css/	Répertoire des fichiers .css génériques
skins/	Répertoire des Thèmes
guiedit/	
sample-config.php	Un exemple de fichier de configuration
scripts/	Scripts étendant les fonctionnalités de PmWiki
wikilib.d/	Répertoire des pages par défaut de PmWiki
cookbook/	Répertoire des compléments de configuration

3) Initialisation

Lorsque vous venez d'installer un PmWiki, le fichier *local/config.php* n'existe pas, et vous devez en créer un si vous désirez effectuer une personnalisation.

Voici un fichier config.php simple :

```
1 <?php
2 $WikiTitle = "MyNewWiki";
3 $PageLogoUrl = "http://www.example.com/mylogo.gif";
4 ?>
```

Figure 2 Initialisation du fichier conf

4. Établissement de la version Française

Si vous souhaitez utiliser une version de PM Wiki dans une autre langue que l'anglais, téléchargez le module de langue archivé au format zip ([i18n/i18n-fr.zip](http://www.pmwiki.org/pub/pmwiki/i18n/i18n-fr.zip)) depuis <http://www.pmwiki.org/pub/pmwiki/>. Ensuite copiez les fichiers de votre version nationale dans le répertoire *wikilib.d/* comme décrit plus haut.

Il y a deux répertoires comprimés dans les archives 'i18n', *scripts* et *wikilib.d*. Copiez les fichiers dans leurs répertoires respectifs de votre installation. Par exemple, pour une localisation en Français, PmWikiFr.* and PmWiki.* doivent être contenus dans le **même** répertoire.

5) Utilisation

a) Édition de base

PmWiki permet aux utilisateurs d'éditer une page à l'aide de quelques outils d'Édition Comme des boutons GUI qui facilite l'édition de texte ou l'ajout de liens, ces derniers sont activés en ajoutant dans le fichier local/config.php la ligne

```
34 $EnableGUIButtons = 1;
```



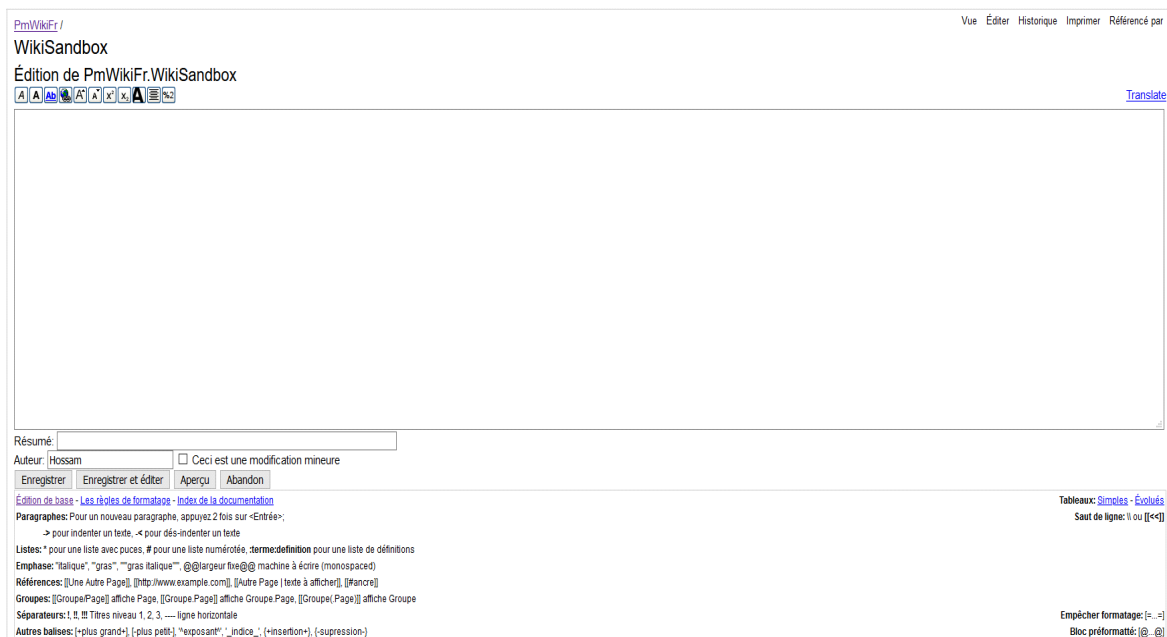


Figure 3 : Interface d'édition d'une page pm wiki

b) Recettes Cookbook

PMwiki propose plusieurs add-ons scripts pour personnaliser vos pages de différentes manières, ces derniers sont faciles à installer et à configurer.

• Exemple1 : ExpandingMenu

Un menu latéral qui prend en charge les listes imbriquées où les éléments imbriqués sont affichés uniquement si l'utilisateur fait partie de ce groupe.

Ajoutez [emenu.php](#) dans le répertoire cookbook.

Ajoutez cette ligne dans le fichier local/config.php

`include_once("cookbook/emenu.php");`

<pre>(:emenu:) * [[Main/Home Page]] * [[Main/wikiSandbox]] * [[PmWiki/Documentation Index]] ** [[PmWiki/Installation]] ** [[PmWiki/Release Notes]] ** [[PmWiki/ChangeLog]] * [[Cookbook/Cookbook]] ** [[Cookbook/Administration]] ** [[Cookbook/Uploads]] ** [[Cookbook/Blog]] (:emenuend:)</pre>	<ul style="list-style-type: none"> • Home Page • WikiSandbox • Documentation Index • Cookbook <ul style="list-style-type: none"> ◦ Administration ◦ Uploads ◦ Blog
---	--

Figure 4 : Exécution du add-on ExpandingMenu

- **Exemple2: Toggle**

Ajoute des liens ou des boutons pour basculer (masquer/afficher) les sections et objets div

Téléchargez [toggle.php](#) et copiez-le dans le dossier du livre de recettes. Installez en ajoutant à config.php :

```
include_once("$FarmD/cookbook/toggle.php") ;
```

Short syntax:

```
(:toggle hide box1 button=1:)
>>id=box1 border='1px solid #999' padding=5px bgcolor=#edf<<
The text in this section can be hidden/shown
>><<
```

Show

Short syntax:

```
(:toggle hide box1 button=1:)
>>id=box1 border='1px solid #999' padding=5px bgcolor=#edf<<
The text in this section can be hidden/shown
>><<
```

Hide

The text in this section can be hidden/shown

- **Pour l'accès au livre des recettes visiter le lien :**

<https://www.pmwiki.org/wiki/Cookbook/Cookbook>

3) Liens et redirections :

PmWiki crée un lien en utilisant le texte entre les doubles crochets.

Pour ce faire, il supprime les espaces entre les mots et met automatiquement en majuscule la première lettre de chaque mot après les espaces ou toute autre ponctuation (comme ~). Ainsi, [[Wiki Sandbox]], [[wiki sandbox]] et [[WikiSandbox]] s'affichent tous différemment mais créent le même lien vers la page intitulée WikiSandbox. Ou en d'autres termes, PmWiki créera automatiquement le "nom du chemin du lien" en utilisant le nom de la page dans CamelCase, mais le "texte du lien" s'affichera dans le format que vous l'avez entré.

Il existe trois façons d'obtenir un texte de lien différent :

- **Masquer le texte du lien.**

Le texte du lien entre (parenthèses) ne sera pas affiché, de sorte que [[(wiki) sandbox]] est lié à WikiSandbox mais s'affiche en tant que sandbox. Pour les adresses contenant réellement des parenthèses, utilisez %28 et %29 <http://www.example.com/linkwith%28parenthese%29>.

- **Modifier le texte du lien.**

Vous pouvez spécifier un autre texte de lien après une accolade verticale, comme dans [[WikiSandbox | une aire de jeu]], ou vous pouvez utiliser une flèche (->) pour inverser l'ordre du texte du lien et de la cible, comme dans [[une aire de jeu -> WikiSandbox]]. Les deux liens s'affichent comme une aire de jeu.

What to type	What it looks like
* [[(wiki) sandbox]]	• sandbox
* [[(wiki) sandbox]]es	• sandboxes
* [[WikiSandbox wiki sandbox]],	• wiki sandbox ,
* [[WikiSandbox wiki sandbox]]es	• wiki sandboxes
* [[BasicEditing +]]	• Basic PmWiki editing rules

Figure 5: Liens et redirections

d) Intégration application web

Pour pouvoir utiliser PM wiki en local sur une application web il faut placer le répertoire pm wiki dans le répertoire de votre projet sous wamp64\www\nomDuRepertoireRacineDeVotreProjet.

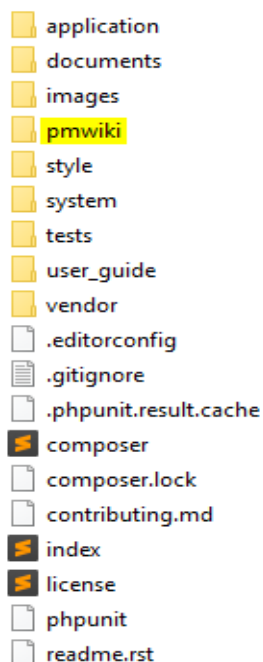



Figure : Emplacement pm wiki dans application web

V) Propositions

1) fonctionnalités

a) ajout d'une proposition

- Cas d'utilisation de PMWIKI au sein du projet Agora

Lors de l'ajout d'une proposition à l'aide de l'icône  dans la base de données un lien de page pm wiki relative à cette proposition est généré et stocké dans la base de données.

+ Options							
←T→	▼	pro_id	pro_nom	pro_fic	pro_avis	pro_pmwikilink	grou_id
□	✎ Éditer	✂ Copier	✖ Supprimer	1	Exemple Proposition	prop1.php ok	1
						http://agora/pmwiki/pmwiki.php?n=Main.prop1	

Ceci génère un fichier Main.prop1 appartenant au groupe Main et le place dans le répertoire wiki.d

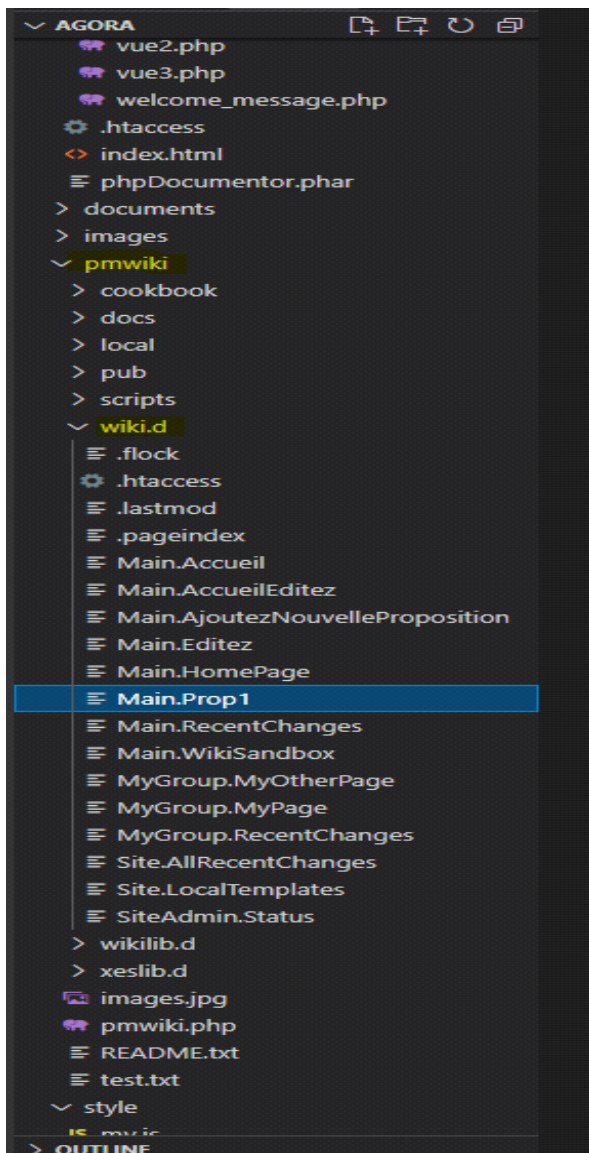

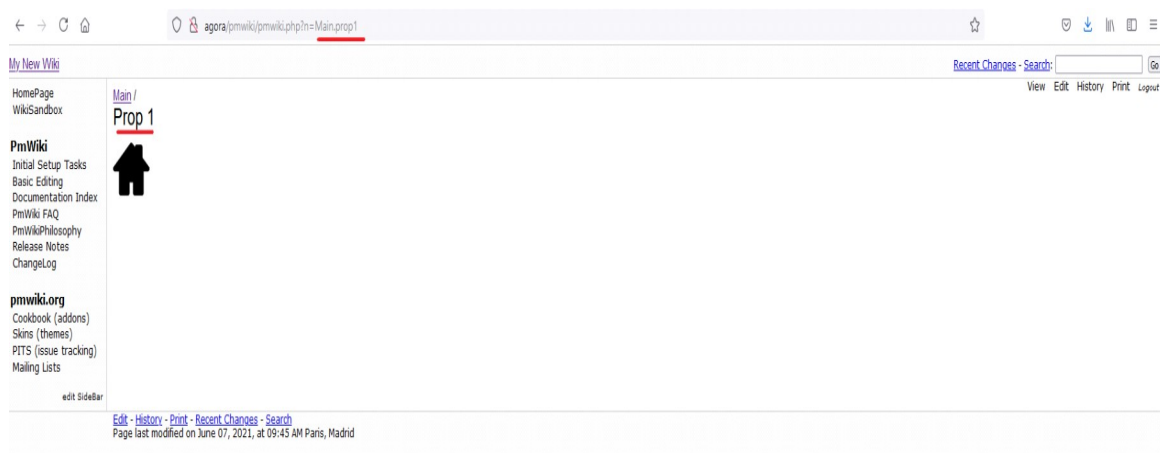


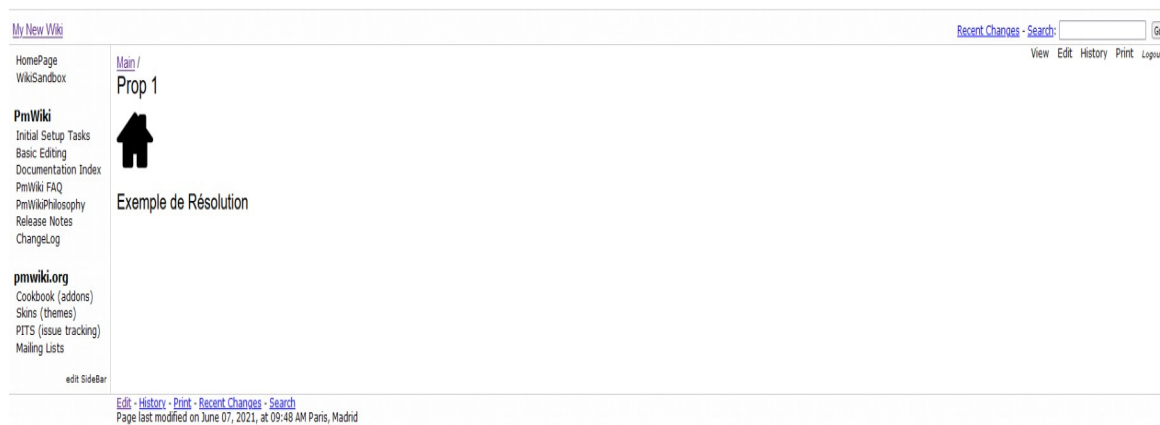
Figure : Emplacement du fichier de proposition

On peut accéder à cette page pm wiki en appuyant sur l'icône Wiki .

Ceci nous mène vers une page PMWIKI portant le numéro d'identification de la proposition ajoutée



Après cela on peut éditer notre page pm wiki en appuyant sur le bouton [Edit](#).



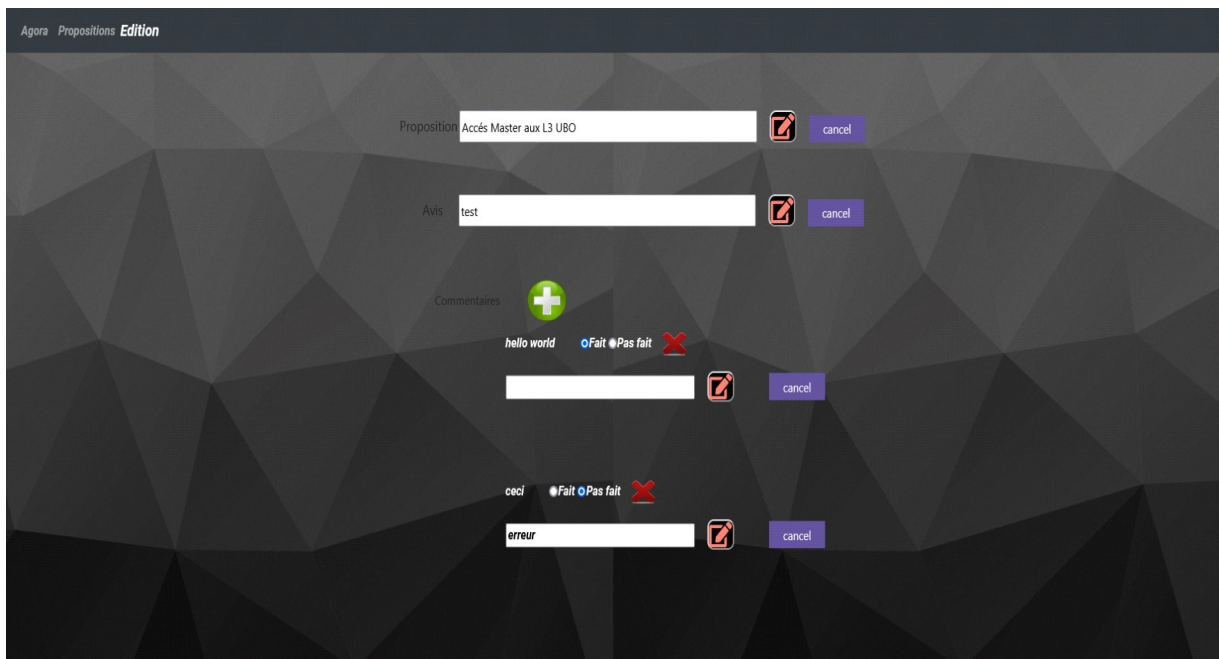
b) Édition d'une proposition :



La page d'édition est accessible partir de l'icône .


Un formulaire est présenté ou le nom de modification de la proposition a lieu.

On peut aussi ajouter des commentaires relatifs à cette proposition ainsi que des réponses à chaque commentaire.



c) Suppression d'une proposition :



Le bouton  sert à supprimer une proposition dans la base de données ainsi que les commentaires associés et aussi la page PM wiki associée.

2) Architecture logicielle :

L'architecture logicielle utilisée dans notre application est basée sur de la programmation objet orienté (POO), et respecte le motif MVC (**Model, View, Controller**) qui se compose de trois modules ou objets principaux à utiliser dans le développement de logiciels :

- Un *modèle* représentant la structure logique sous-jacente des données dans une application logicielle, ainsi que la classe supérieure qui y est associée. Ce modèle d'objet ne contient aucune information sur l'interface utilisateur.
- Une *vue*, autrement dit un ensemble de classes représentant les éléments de l'interface utilisateur (tous ceux que l'utilisateur voit à l'écran et avec lesquels il peut interagir : boutons, boîtes de dialogue, etc.).
- Un *contrôleur* représentant les classes qui se connectent au modèle et à la vue, et servant à la communication entre les classes dans le modèle et la vue.

Trois Classes sont utilisées

Classe Controller : Permet de récupérer et envoyer des données depuis les vues vers des méthodes de la classe api et ensuite de charger des vues avec ces informations.

Classe Api : Permet de transmettre les méthodes en faisant appel à d'autres méthodes de la classe Manage_db.

Classe Manage_Db : Cette classe sert à envoyer des requêtes vers la base de données et d'effectuer différents traitements et manipulations CRUD sur des données.

Pour que ces classes puissent communiquer entre elles on utilise la ligne ci-dessous pour pouvoir faire appel à une classe précise.

```
require(FCPATH."application\models\Manage_DB.php");
```

Les variables de type classe sont déclarées comme ceci

```
/**
 * Manage_DB
 *
 * @var Manage_DB
 */
public $mdb;
```

Cette architecture permet aussi de déclarer des constructeurs pour les classes. Les classes qui possèdent une méthode constructrice appellent cette méthode à chaque création d'une nouvelle instance de l'objet, ce qui est intéressant pour toutes les initialisations dont l'objet a besoin avant d'être utilisé.

```
class Controller extends CI_Controller {

/**
 * Api
 *
 * @var Api
 */
public $api;
public $data;

public function __construct()
```

```

{
    parent::__construct();
    $this->load->helper('url');
    $this->api=new Api();
    $this->data['configname']= $this->api->get_config_name();
    $this->data['grpname']= $this->api->get_grp_name();
}

```

3) Modèle de base de données

Après la mise en place de l'environnement de développement et la compréhension du cahier des charges on a la modélisation de la base de données qui est un élément indispensable pour le développement de l'application. Cette modélisation comprend trois parties :

- **le diagramme de classe uml** : le diagramme permet de montrer les différentes classes , leurs attributs et la relation qui existe entre ces attributs. Ces relations sont illustrées par des chiffres et des lettres et sont appelés des cardinalités.

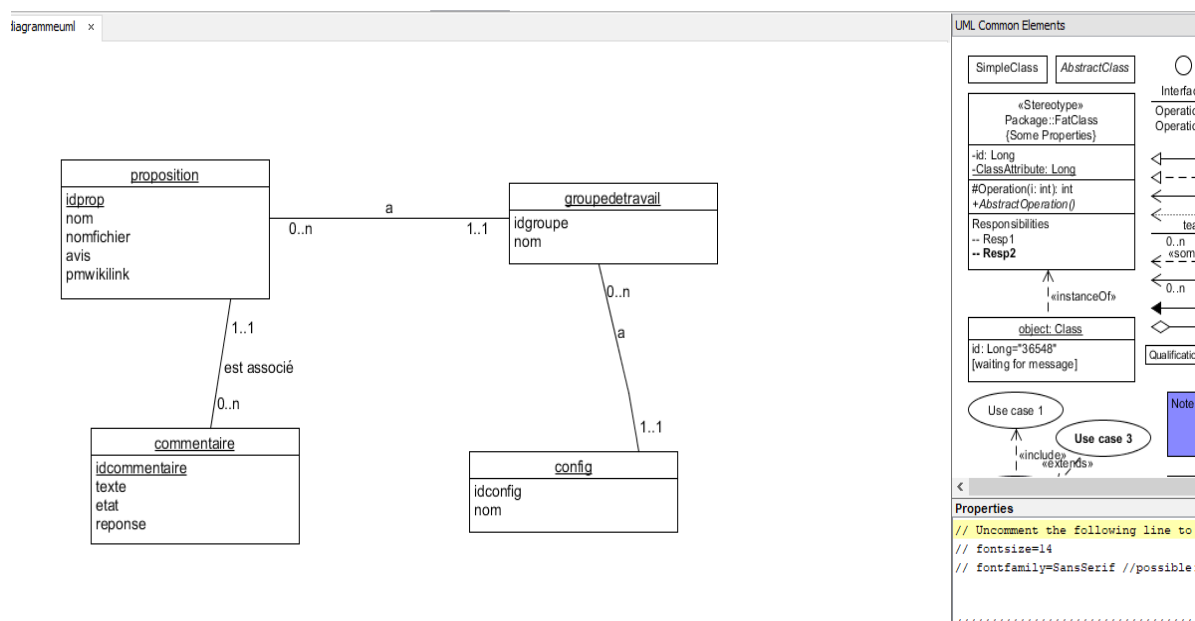


Diagramme de classes uml réalisé sur umlet

Sur cet diagramme on a quatre classes qui sont config, département, proposition et commentaire et on peut connaître les liens entre ces différentes classes.

Exemple entre proposition et groupe de travail :

Sur l'image ci-dessus on peut lire qu'une proposition a un et un seul groupe de travail et qu'un groupe de travail a 0 ou plusieurs propositions.

- **Modèle de la base de données relationnelle :**

c'est un modèle qui est créé grâce au diagramme de classes uml dont les différences sont la migration des clés primaires entre les différentes tables (qui deviennent des clés étrangères) et la disparition des cardinalités. La représentation de cet modèle sur mysqlworkbench permet de générer le code de création de la base de données sous format SQL.

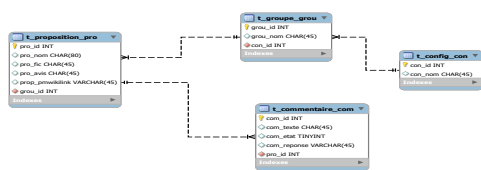


Schéma de la base de données relationnelle réalisé sur mysqlworkbench

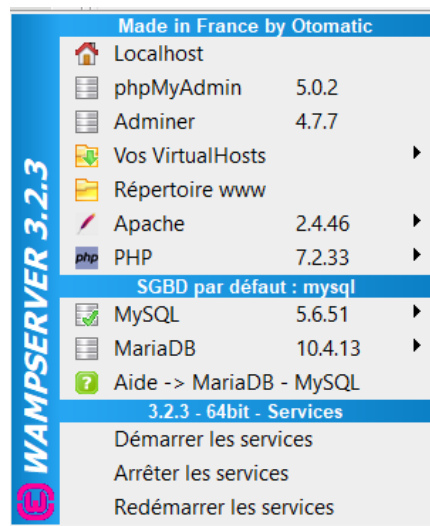
- **Création de la base de données et l'insertion du jeu de données sur phpMyAdmin**

Le code de la base de données généré sous format SQL sur mysqlworkbench sans le mot « VISIBLE » permet de créer la base de données sur phpMyAdmin en faisant une importation. Ainsi on peut insérer le jeu de données et commencer le développement de l'application sur codeigniter.

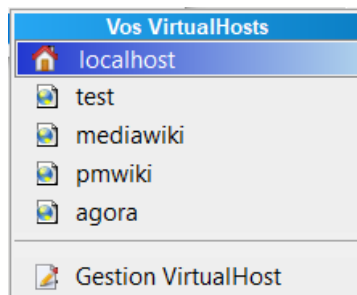
VI) Résultats

Lancer le projet

- Pour lancer l'application appuyé sur wamp



Sur cette fenêtre aller sur « Vos virtualHost



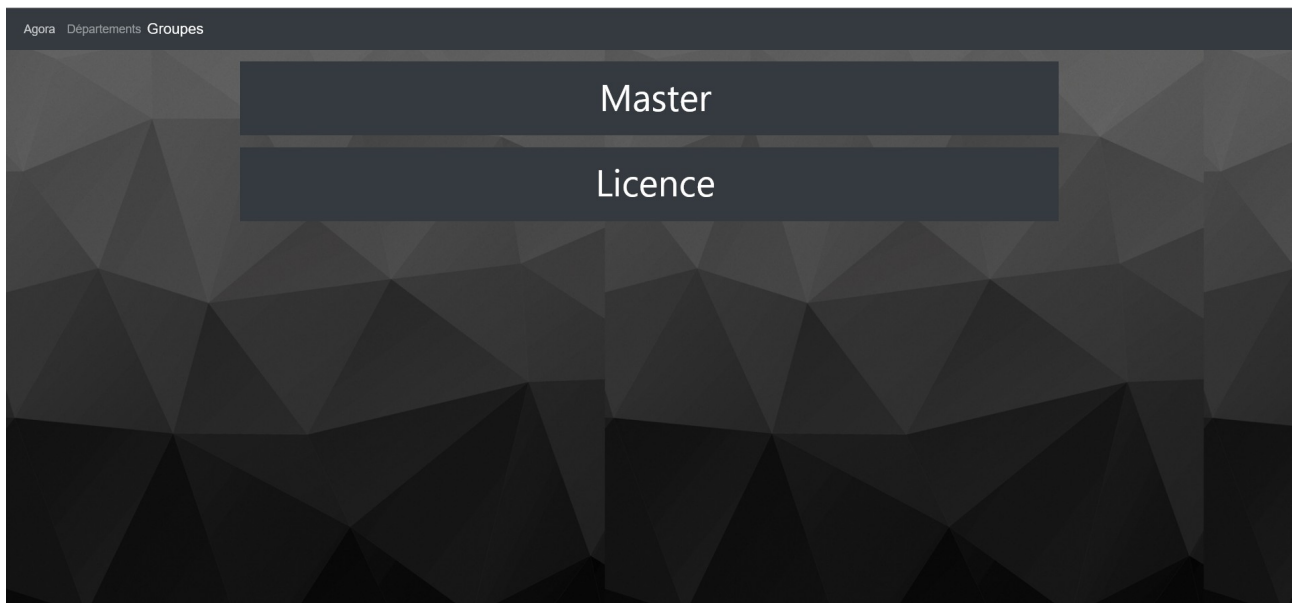
Ensuite lancer le virtualHost que vous

avez crée

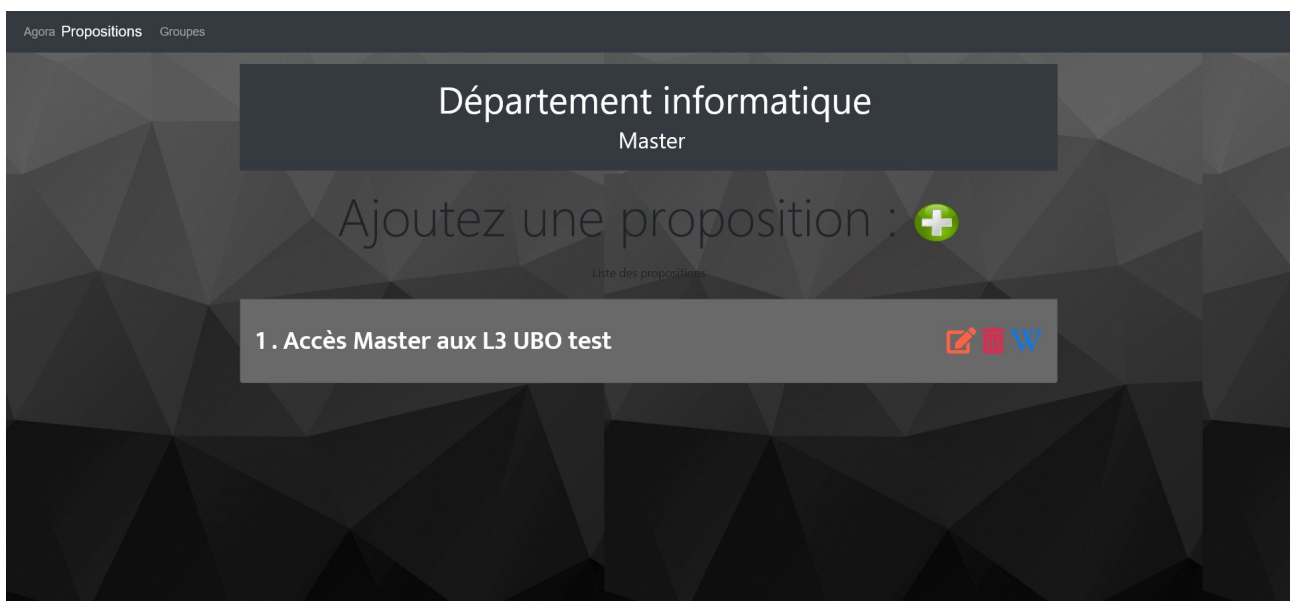
Lorsque l'on lance le site, on arrive sur la page ci-dessus ou nous pouvons choisir le département sur lequel on veut accéder.



Une fois, le département choisit on doit choisir le groupe auquel on veut participer



Une fois arrivé sur cette page, on pourra voir les différentes propositions, il y aura plusieurs actions possible.



Ce bouton permettra de créer de nouvelles propositions.

On arrivera sur la page ci-dessous, on pourra écrire la proposition en appuyant sur submit on enregistrera la proposition et on retournera la page d'avant.

En utilisant cancel, on retournera à la page d'avant sans créer de nouvelles propositions.

2

Proposition :

Accès Master aux L3 UBO test | x

submit

cancel




Ce bouton permettra d'accéder à la page d'édition d'une proposition.

Agora Propositions **Edition**


1 Proposition Accès Master aux L3 UBO test [edit icon] [cancel]

2 Avis [edit icon] [cancel]

3 Commentaire [edit icon] [cancel] [plus icon]

Dans le champ numéros 1, on retrouvera le nom de la proposition que l'on pourra modifier puis enregistrer avec le bouton .

Dans le champ numéros 2, on pourra mettre un avis.



Dans le champ numéros 3, on pourra écrire un commentaire puis appuyer  pour créer le commentaire et faire apparaître les champs suivants :

1. info ● Fait ○ Pas fait 

[input field]





On pourra choisir entre cocher entre fais et pas fais si on a vu le commentaire en mettant du texte pour expliquer ce que l'on a fait ou non puis on appuie sûr  pour valider, la  permettra de supprimer le commentaire.

Ce bouton permettra de supprimer la proposition.



Ce bouton permettra d'accéder à la page wiki

Après avoir appuyé sur le bouton wiki, on arrivera sur une page de connexion.

Prop 1

Password required

Name:

Password:

Pour le tests nous avons mis les identifiants administrateur pré programmé dans le pmwiki

On arrive sur la page suivante

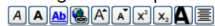


En appuyant sur ce bouton, on retournera à la page des propositions.

Lorsque l'on arrive sur la page, on ne pourra pas écrire, il faudra cliquer sur le bouton « Edit » pour entrer en mode édition.

Prop 1

Editing Main.Prop1



%width=60%[[http://agora/index.php/controller/props/1|https://cdn.discordapp.com/attachments/841672965967052830/846732906881220608/telechargement.png]]

Summary:

Author: ☐ This is a minor edit

On entrera en mode édition, on pourra écrire du texte et appliquer une mise en forme.

VII) Conclusion

Faire ce projet nous a donné l'opportunité de découvrir les qualités qu'un développeur web doit avoir, la découverte de nouveaux outils de développement, l'organisation que peut avoir une entreprise et surtout de connaître certains de nos défauts dans le domaine de l'informatique. Même si le développement web n'est pas le métier que certains d'entre nous veux exercer à la fin de nos études on a pu se faire une idée de ce qui nous attend dans le monde du travail ce qui va nous permettre de se préparer à l'avance pour bien réussir nos objectifs. De plus cette expérience nous a donné une autre vision de la programmation avec la méthode agile et on pense que cette méthode devrait être introduite dans le programme des études en informatique. Même si toutes les fonctionnalités demandées dans le cahier des charges ont été faites certaines pourraient être améliorées en gérant la session de connexions, l'envoi par mail des informations de la proposition et l'ajout de règles d'édition.