

Ninon Morère



Technologies de l'Information  
et de la Formation

Mémoire de stage

## REMERCIEMENTS

Je remercie chaleureusement M. ESPLAN Nicolas ainsi que M. LESAULT Alexandre pour m'avoir permis d'effectuer mon stage à Andil. Également, je tiens à remercier M. HEGUY Olivier et M. TROUVÉ Arnaud pour l'accompagnement et le suivi de mes missions, ainsi que toute l'équipe.

Enfin, un grand merci à Florence CALMETTES pour son aide et tous ses conseils, tout au long de ma formation.

# SOMMAIRE

Présentation personnelle	4
Abstract	5

## I. Cahier des charges

1. Présentation de la société	6
2. Présentation des missions	6
3. Planning des missions	7
4. Exigences techniques	8
5. Exigences fonctionnelles	8

## II. Spécifications fonctionnelles

1. Un point sur l'authentification dans Moodle	10
2. Ma réalisation	10
a. Use case	10
b. Diagramme de séquence	11
c. Maquettes	12
d. Maquettes (réfection graphique)	13 – 15

## III. Conception

1. Modèle conceptuel de données	16
2. Modèle logique de données	17

## IV. Spécifications techniques

### *Plugin d'authentification CNCGP*

1. Outils et langages	18
2. La tâche	19
3. Le code	
a. Introduction	20
b. Généralités	20
c. Installation de la table cncgpmembers	20
d. Les fonctions	21
e. Le formulaire d'authentification	22
f. Communiquer avec la base de données	23
g. Le style	24
4. Le rendu final	24

### *Réfection graphique d'un plug-in de statistiques*

1. Les templates 'mustache'	25
2. Mise en place	26
3. Version finale	29
4. Améliorations	31

## V. Fonctionnalités

1. Les restrictions	32
2. Le bug	32
a. Les variables d'origines	33
b. La stack trace	33
3. Ma solution	33

VI. Conclusion	35
----------------	----

# LE COMMENCEMENT

Je m'appelle Ninon Morère, j'ai 27 ans et j'ai le souhait de devenir développeuse web.

J'ai commencé un parcours d'infographiste à l'âge de 17 ans au lycée de Cahors puis au lycée de Montpellier. Je suis enfin revenue sur Toulouse pour faire un BTS appliqué en graphisme et print. Après plusieurs années dans l'administratif, j'ai décidé de renouer avec ma branche initiale et de me mettre au goût du jour en exportant mon savoir et mes capacités sur la toile. Ainsi j'ai effectué cette formation à l'Adrar dans le but de me familiariser et, pourquoi pas, me spécialiser dans le développement.

Une amie à moi travaillant actuellement à Juriscampus, a croisé ma demande de stage et m'a proposé très gentiment de la faire suivre à la direction d'Andil. Ayant été élève du lycée Saliège en formation Webmaster, j'ai eu la chance de connaître messieurs HÉGUY, LESAULT et ESPLAN en cours, respectivement, de HTML/CSS, de droit et de stratégie commerciale. Ainsi, j'ai pu me retrouver dans un environnement familier où mettre en application mes compétences. Arnaud Trouvé, mon tuteur de stage, m'a énormément guidé sur le langage et la logique MOODLE, me permettant ainsi d'observer la vie en entreprise, concernant ce cursus.

## ABSTRACT

My project consists in developing many 'plug-ins' for e-learning courses. Andil is a company which develops e-learning websites based on LMS (learning management systems) called MOODLE for many firms specializing in property management. The team offers many plug-ins, modules, or ways to fit to the client's projects. The clients' requirements can vary from a single registration button to a lesson layout including media for example. Andil manages the support by hotline to assist and guide the MOODLE users to resolve issues (log in errors, password forgotten, sending certificates and so on).

To I used a language specific to MOODLE and based on PHP. My two teammates were dedicated to MOODLE support and plug-in creation. One on them is an MOODLE expert and regularly helps the MOODLE core to be improved in term of security or code optimization. He is a real mastermind and I benefited greatly from his expertise. The client was very demanding and the platform always had to be upgraded.

I managed to finish two and a half plug-ins out of three due to a lack of time. I found it difficult at first to get to know this new world but I the end I am very satisfied with what I've learnt from this experience.

# CAHIER DES CHARGES

## 1. Présentation de la société

**ANDIL** est une SARL (société à responsabilité limitée) uni-personnelle. La société est en activité depuis 16 ans et est localisée à LABEGE (31670). Elle est spécialisée dans le secteur d'activité du conseil en systèmes et logiciels informatiques. Andil est une société relativement polyvalente puisqu'elle intervient notamment dans le e-learning (formation à distance), le développement web, l'édition de logiciels et le conseil.

Andil travaille avec la société **JurisCampus**, toutes deux sous la responsabilité des mêmes dirigeants. JurisCampus est un institut de formation professionnelle juridique spécialisée en Droit et en Gestion du Patrimoine. Elle assure donc la production de contenus pédagogiques à destination de ses clients.

La société Andil a également un institut de formation : **AndilCampus**. Cet institut est spécialisé dans les métiers du numérique et de l'innovation digitale. L'expertise pédagogique d'AndilCampus s'appuie sur son savoir-faire en matière de déploiements d'offres de formation en E-learning. AndilCampus est également à l'initiative de la formation d'expert en stratégie digitale et webmaster online.

J'ai effectué mon stage du 07/01/2019 au 15/03/2019 au sein d'Andil. Il a été décidé de me confier comme mission durant ces dix semaines de stage, la conception ainsi que le développement de plugins et de session de débogage pour les plateformes Moodle d'Andil.

**Ces plugins** ont pour but d'améliorer l'expérience utilisateur lorsque celui-ci se rend sur son compte via les différentes plateformes. De facto, **les sessions de débogage** permettent une utilisation optimale de toutes les composantes des plateformes Moodle, mises à disposition par Andil. Il est important de faire valoir un rendu qualitatif et professionnel, dans un souci de crédibilité, quant aux cours dispensés et aux certifications finales délivrées.

## 2. Présentation des missions

### Plugin d'authentification CNCGP

Les plateformes e-learning dont la maintenance est gérée par Andil, proposent des formations payantes pour professionnels et particuliers. L'une d'elle, appartenant à la CNCGP, souhaiterait pouvoir intégrer une liste de membres dont le paiement a été effectué en amont par leur structure. Les nouveaux arrivants auraient donc la possibilité de récupérer leurs informations personnelles par le biais d'un code membre que la CNCGP aurait fourni à leur société. Pour cela il faudrait donc deux méthodes d'authentification : une par code membre pour les employés dont leur société ont déjà payé, et l'autre sans code membre, plus classique, pour les particuliers ou personnes sans structure.

## Débug d'un plug-in d'importation de template de certificats

Très souvent il y a des cas d'erreurs que nous ne pouvons prévoir et qui finissent par arriver au cours des utilisations de la plateforme. Le cas ici concerne le module de cours et notamment, de certification, de MOODLE.

Une migration a été faite en début d'année, dans le but d'avoir une plateforme propre contrairement à l'ancienne, qui accusait le poids des bugs et des tests peu concluants. Il était nécessaire d'y importer tous les cours. Cependant, les chapitres avec une certification à la clé présentaient tous le même bug : le modèle PDF de base n'était plus disponible car relié à un autre MOODLE.

## Réfection graphique d'un plug-in de statistiques

Comme expliqué dans l'introduction, Andil possède sa propre plateforme e-learning : andilcampus. Des cours y sont dispensés par des formateurs et professeurs agréés. Un élève, lorsqu'il se connecte à son compte, voit sur son tableau de bord ses statistiques : le nombre de pages visitées, ses notes, les conversations suivies dans les différents forums, etc. Il est fonctionnel et utilisé.

Ce plugin de statistiques a été codé par l'équipe dans laquelle je fais mon stage, qui a mis de côté l'aspect design de la chose. Ainsi, les informations sont mal disposées, peu agréables à visualiser et il faut à chaque fois cliquer sur des liens pour déplier l'information nécessaire. C'est laborieux et non ergonomique.

Je suis amenée à proposer aujourd'hui une touche plus design et esthétique afin de parfaire la présentation des outils, à l'aide de bootstrap 4. Je propose donc une vision plus artistique et élégante de l'ensemble du plug-in.

### 3. Planning des missions

Janvier	
du 07/01 au 31/01	Plugin d'authentification pour la CNCGP
Février	
01/02 au 08/02	Débug d'un import de template de certificat
Tout le mois	Réfection graphique d'un plugin de statistique



## 4. Exigences techniques

### Plugin d'authentification CNCGP

- Utilisation de MOODLE obligatoire ;
- Interface d'authentification ;
- Les interfaces doivent être graphiquement et esthétiquement simples. (Pas d'encombrement avec des détails techniques et une abondance de champs texte) ;
- On doit donc trouver d'une part le code unique pour s'inscrire, d'autre part une interface d'authentification complète (avec les informations du type nom, prénom, âge...).

### Réfection graphique d'un plug-in de statistiques

- Utilisation de bootstrap obligatoire ;
- Respecter le code du plugin existant ;
- Informations visuelles et claires, sans surcharge ;
- Adapté au thème déjà en place.

## 5. Exigences fonctionnelles

### Plugin d'authentification CNCGP

Mon projet a été de réaliser un plugin, avec les outils déjà en place, qui permet aux utilisateurs de s'authentifier grâce à un identifiant unique appelé « code membre ». Une fois celui-ci entré et validé, un mail lui est automatiquement envoyé, avec un mot de passe temporaire à changer dès sa première connexion. Ses informations personnelles ont déjà été enregistrées par un des organismes partenaires de *JurisCampus*, dans une table spéciale de la base de données, de manière à ce que l'utilisateur ait un profil déjà prérempli. En parallèle, ceux n'ayant pas de compte peuvent en créer un.

A l'installation, le plugin crée une table nommée 'cncgpmembers'. Dans l'interface d'authentification, il présentera deux liens : un pour les codes membre, un pour ceux sans code membre.

*Sans code membre* : renvoi sur l'authentification classique proposée par MOODLE.

*Avec code membre* : renvoi sur un formulaire où on pourra voir le champ « code membre », le champ « email » et un bouton d'envoi.

L'utilisateur, s'il existe dans la table 'cncgpmembers', sera alors enregistré avec toutes ses informations, dans la table 'user' de MOODLE. En parallèle, un mail lui sera envoyé avec un mot de passe aléatoire, haché en base de données, qui lui permettra de se connecter quand il le souhaite.

Lors d'un import d'utilisateurs dans la table 'cncgpmembers' depuis un fichier de type .csv (updatemembers.csv), le plugin vérifiera si les codes membre des utilisateurs ont changé. Si oui, il leur enverra un email avec leur nouveau code membre.

## Réfection graphique d'un plug-in de statistiques

Les balises html que nous connaissons n'ont pas été écrites en dur. Il a fallu passer par des templates en .mustache et donc, par leur syntaxe, pour pouvoir afficher les informations utiles. Dans le cas présent, je n'avais rien à enlever, simplement à réorganiser.

En parallèle, la feuille de style du thème se trouvait dans l'administration système de la plateforme moodle.

## II. SPECIFICATIONS FONCTIONNELLES

### Plugin d'authentification CNCGP

#### 1. Un point sur l'authentification dans MOODLE

Sur MOODLE, l'authentification est le système d'inscription en tant que nouveau membre. Il y a une différenciation de terme entre « l'inscription » et « l'authentification ». En effet, MOODLE interprète le terme « inscription » comme un membre qui s'inscrit à un cours ou à des leçons. L'authentification est l'enregistrement de nouveaux membres dans la base de données. L'accès à son profil grâce à ses identifiants sera nommé « connexion ».

#### 2. Ma réalisation

##### a. Use case

Le Use Case est valable pour l'entièreté du site.  
On imagine alors facilement deux cas de figure : Ulysse et Achille.

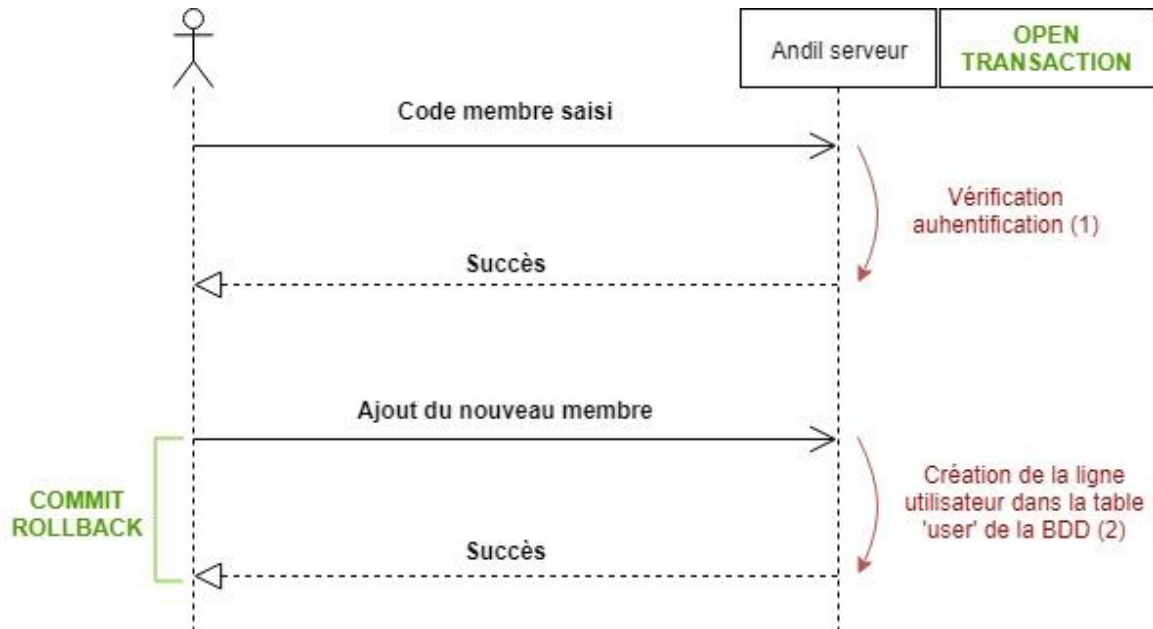
Ulysse est un cadre dynamique de trente-deux ans, habitant à Lyon, exerçant une profession de commercial dans un cabinet de conseils en gestion de patrimoine. Bien qu'il ne soit pas lui-même conseiller, il démarché les particuliers pour proposer les prestations de son agence. Il a alors besoin d'une certaine connaissance dans le domaine, attestée dans un premier temps par ses études et, au fil des ans, par certaines certifications. Son employeur le positionnera alors, aux frais de l'entreprise, sur des cours et leçons spécifiques qui lui permettront de mettre ses connaissances à jour, tout en continuant de travailler à côté. Il aura donc un lot d'heures de cours, auprès d'une des institutions de juriscampus, plus mesuré par rapport à ses collègues conseillers. Ces derniers, pour une mise à niveau, auraient des cours totalement différents et beaucoup plus techniques. Ulysse se connectera avec un code membre que lui aura fourni son employeur et donc, par extension, l'institution de juriscampus à laquelle il aura adhéré.

Achille avoisine les quarante ans, vit à Rennes et a enchaîné quelques petits boulots qui lui ont toujours permis de payer ses factures sans pour autant déclencher chez lui une passion quelconque. Jusqu'à ce qu'il tombe sur une offre d'emploi lui révélant, après quelques recherches, un cursus qui le passionne : la finance. Et, notamment, l'investissement financier. Le secteur de la gestion de patrimoine est tout trouvé et il décide, en tant que particulier, de se lancer dans une formation complète évoquant les points essentiels de cette branche. Il cherche et trouve une des institutions gérées par juriscampus. Il téléphone, commande une brochure et décide, au bout de quelques jours, de souscrire. La formation est alors à ses frais, mais la réorientation professionnelle est prise en compte dans les cursus, de manière à effectuer les mises à niveau nécessaires. Achille pourra donc créer son compte classiquement avec une confirmation par mail, vu qu'il n'appartient à aucune structure et n'a aucun code membre.

## b. Diagramme de séquence

Voici les diagrammes de séquence des deux utilisations possibles du plugin : avec code membre et sans code membre.

### → Code membre



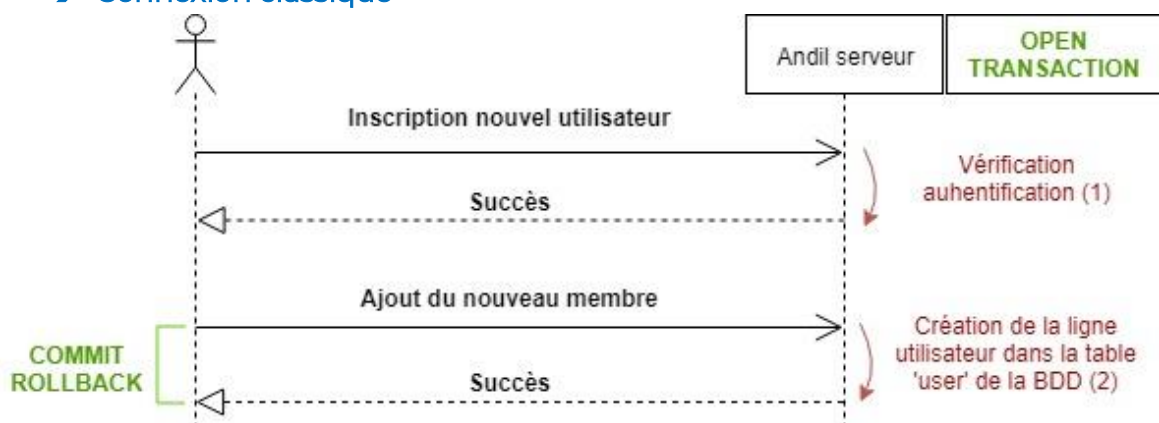
#### 1. Authentification

- 1.1 Code membre inexistant
- 1.2 Echech d'authentification

#### 2. Mise à jour BDD

- 2.1 Si utilisateur déjà existant alors afficher un message d'erreur
- 2.2 Fin diagramme de séquence

### → Connexion classique



#### 1. Authentification

- 1.1 Email et/ou pseudo existe déjà
- 1.2 Echech d'authentification

#### 2. Mise à jour BDD

- 2.1 Tous les champs ne sont pas correctement renseignés
- 2.2 Fin diagramme de séquence

### c. Maquettes

Comme expliqué ci-dessus, il est primordial pour l'utilisateur d'accéder à un écran lui indiquant deux choix, en fonction de son profil : est-il préinscrit ou non ?

Maquette d'un écran web pour la formation de la Chambre Nationale des Conseils en Gestion de Patrimoine. L'interface comprend une barre de menu en haut à gauche et deux boutons de fermeture en haut à droite. Le titre principal est "Formations Chambre Nationale des Conseils en Gestion de Patrimoine". En dessous, il y a deux boutons : "J'ai un code membre" et "Je n'ai pas de code membre". Un message d'information indique que les heures de formation ne seront comptabilisées que si l'utilisateur est membre et indique son code MB.

Menu

Formations Chambre Nationale des Conseils en Gestion de Patrimoine

J'ai un code membre Je n'ai pas de code membre

Si vous êtes adhérent de la CNCGP, vous devez accéder à votre espace de formation en cliquant sur « Je suis membre » et en indiquant votre code MB. A défaut, vos heures ne pourront pas être comptabilisées par la CNCGP ; elles ne figureront pas dans votre carnet de formation de la CNCGP.

Fig. I, E. Maquette

Une fois 'j'ai un code membre' cliqué, il doit entrer son code membre et son email pour effectuer la création de compte dans la table user et l'envoi de mot de passe automatique.

Maquette d'un écran web pour la création de compte. L'interface est similaire à la précédente, mais avec un message d'information sur la recherche du code MB. En dessous, il y a des champs de saisie pour le "Code membre" et l'"Email", suivis d'un bouton "Envoyer".

Menu

Formations Chambre Nationale des Conseils en Gestion de Patrimoine

Vous pouvez retrouver votre code MB sur le site internet de la Chambre, sur votre espace adhérent, le code MB s'affiche en haut à droite de l'écran. En cas de difficulté, contactez la CNCGP au 01.42.56.76.50

Code membre \* Email \*

Envoyer

Fig. II, E. Maquette

## Réfection graphique d'un plug-in de statistiques

### d. Maquettes

Design originel :

Rapport hebdomadaire


Rapport hebdomadaire du 11.02.19

Choisissez une date

11

février

2019



Enregistrer

[Graphique d'activité](#)

[Activité sur le site](#)

Nombre de connexions cette semaine: 2

Nombre de pages vues cette semaine: 7

[Rapport des notes](#)

Aucun note rapporté cette semaine

[Rapport des forums](#)

Aucun message lu cette semaine

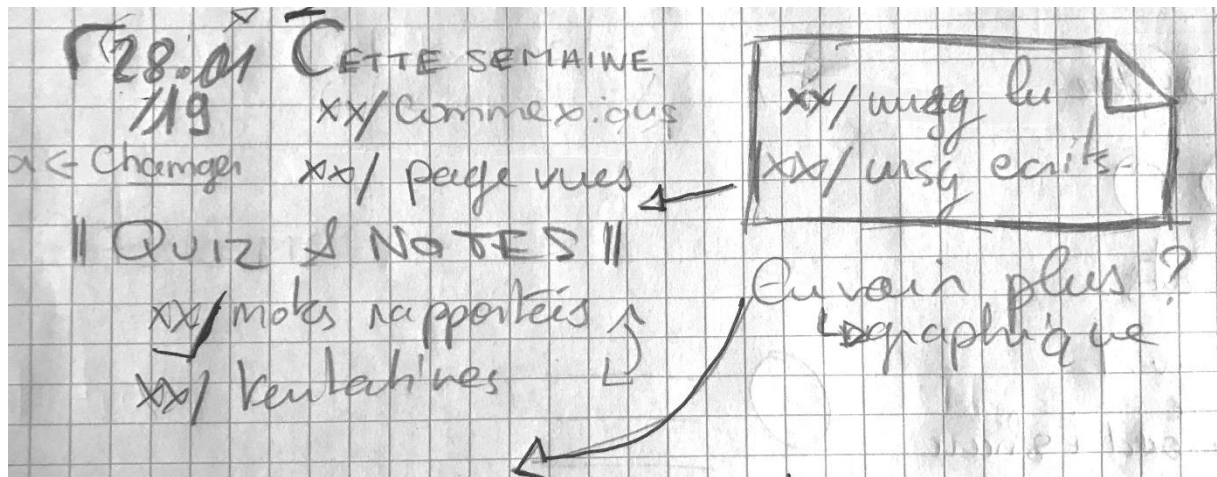
Aucun message écrit cette semaine

[Rapport quiz](#)

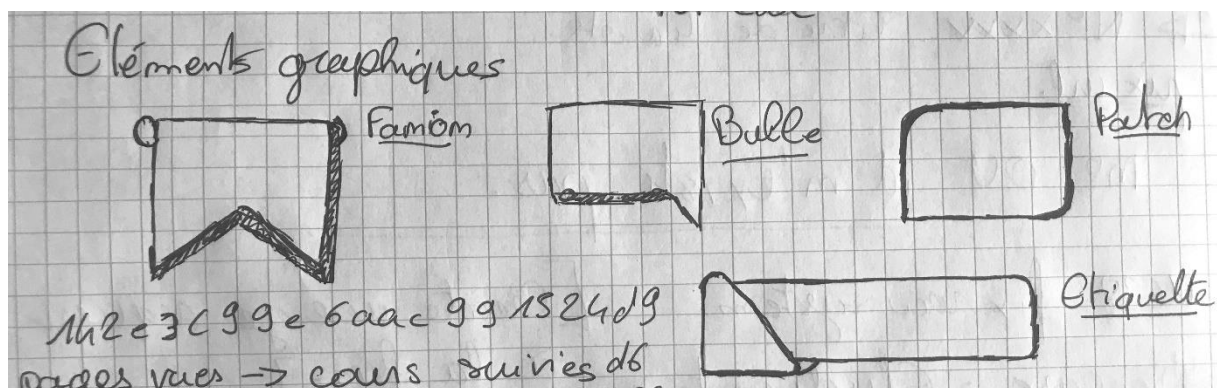
Aucun tentative cette semaine

Chaque titre en bleu est un lien sur lequel il faut cliquer pour dérouler le texte juste en dessous. Que ce soit sur web ou mobile, aucune praticité dans cette façon de faire. On perdait également beaucoup d'espace également et personne n'avait envie de lire ces informations. Il fallait le rendre plus pratique, plus attractif et surtout, plus pertinent. Qu'en un coup d'œil toutes les informations soient visibles. De plus, comme le bloc du résumé des leçons en cours doit venir juste en dessous, il ne fallait pas qu'il soit totalement englouti sous la ligne d'horizon à cause du plugin de statistiques. Il y avait donc, clairement, une hiérarchie à revoir et à refaire.

Avant de toucher au design, j'ai décidé de faire quelques maquettes pour visualiser la manière dont j'allais organiser cela. Ce sont des maquettes manuelles plus pour comprendre dans quelles directions je pouvais partir et jusqu'où j'étais prête à aller. Dans ce genre de situation, une multitude de possibilités créatives s'offrent à moi et il me faut les dessiner pour canaliser cette énergie créative. Ce sont donc des griffonnages personnels et peu qualitatifs mais ils m'ont permis de comprendre un peu mieux le besoin réel.

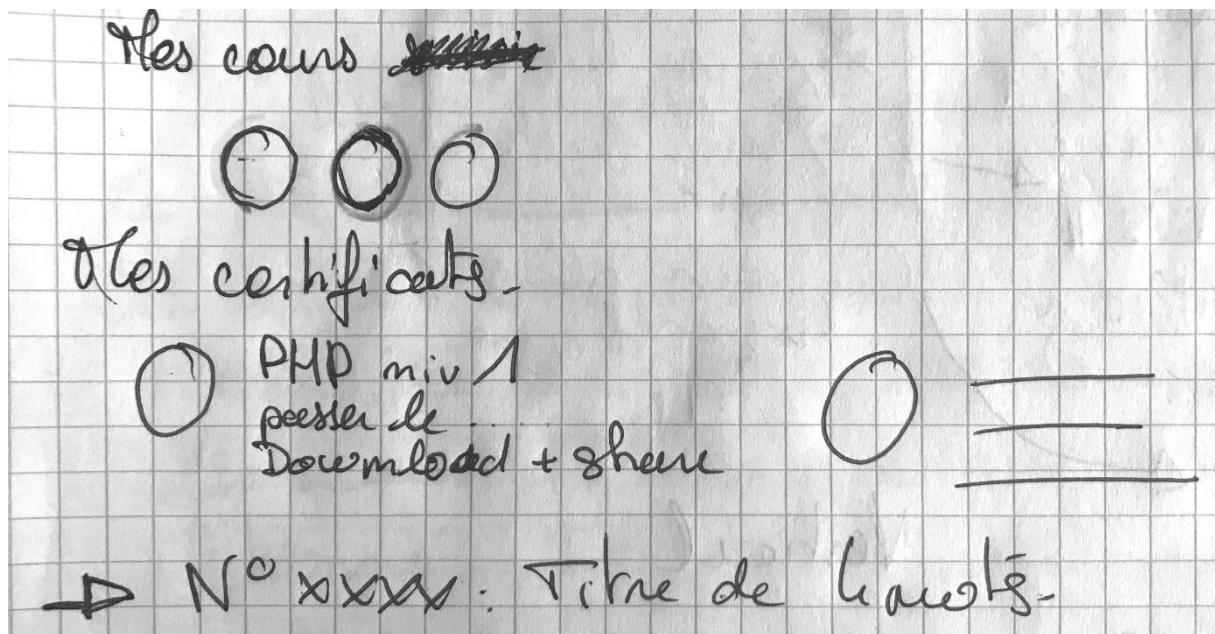


1. Une maquette globale avec une compression des éléments et une feuille de style très stylisée. Point négatif : beaucoup trop 'chargé' tout et ne respecte pas le style graphique du site d'andilcampus.

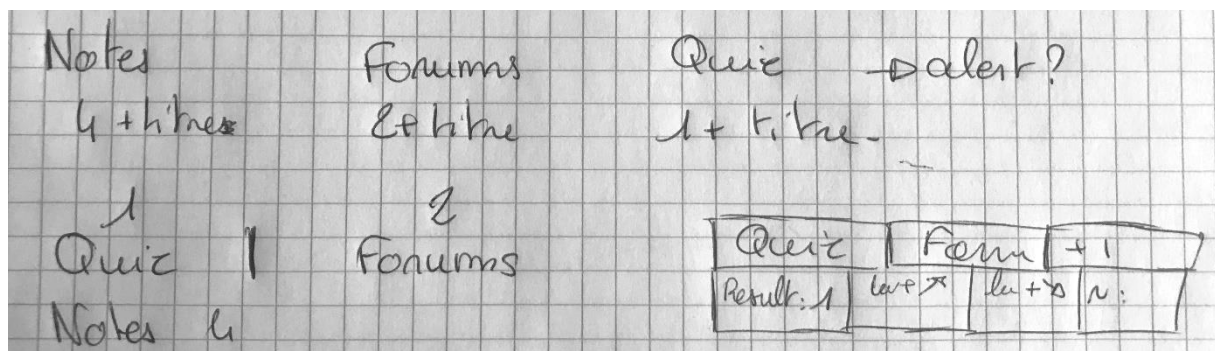


2. Des éléments graphiques assez parlants pour entourer les chiffres ou encore mettre en avant des éléments importants. Quelque chose sera fait mais de beaucoup plus neutre car pour le besoin présent, c'était trop typé.





3. Idées de petites icônes et design intéressants, donnant du dynamisme au contenu.



4. J'ai finalement opté pour un tableau ou, tout du moins, une forme de tableau. Linéaire, restreinte et bien sûr, responsive au possible.



# III. CONCEPTION

## 1. Modèle conceptuel de données (MCD)

La communication principale se fait exclusivement entre trois tables.

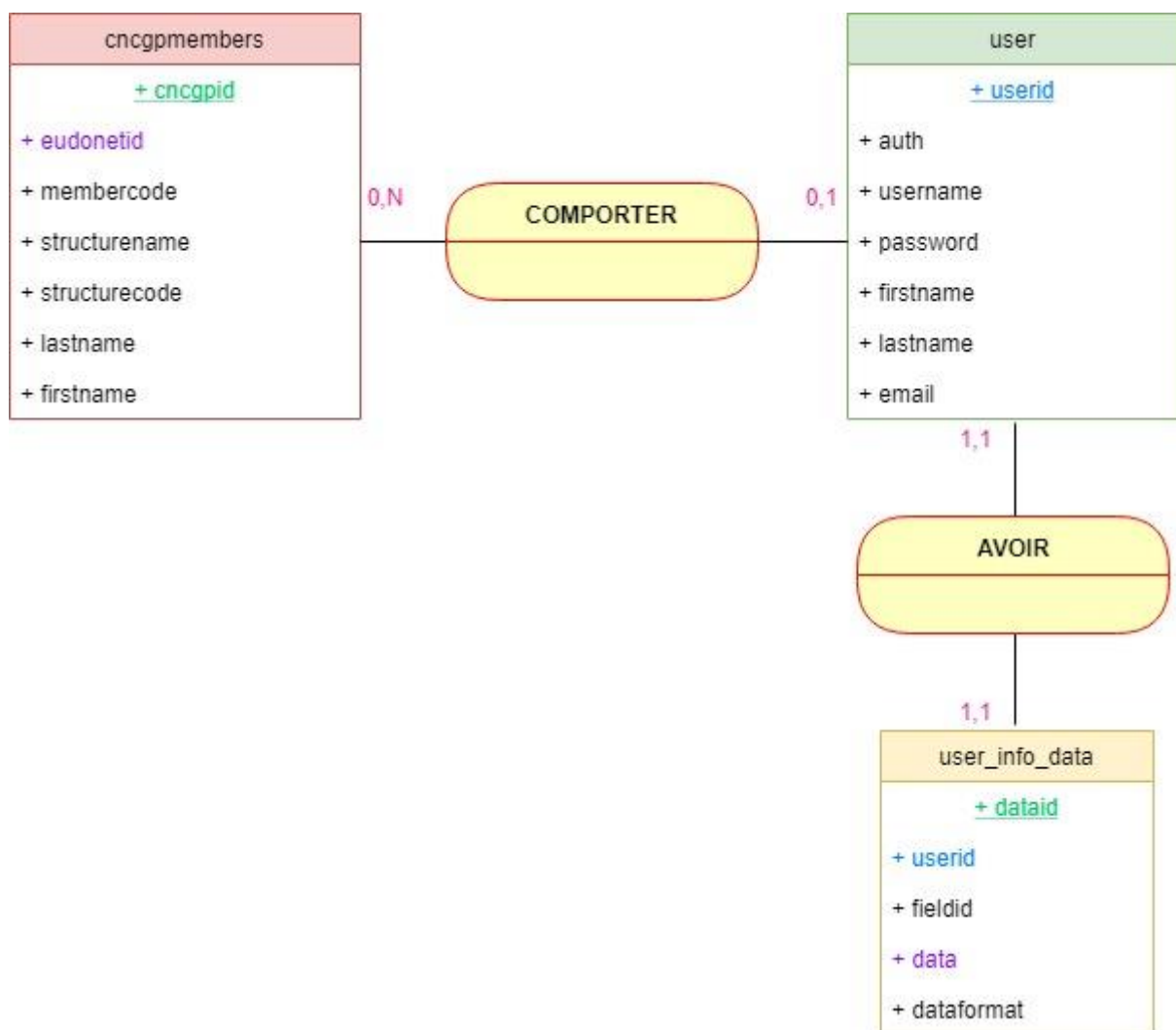
Cncgpmembers : l'import du .csv se fait dans cette table, où une partie des champs de profil sont renseignés.

User : gère les utilisateurs qui viennent de s'authentifier et recense les comptes suspendus

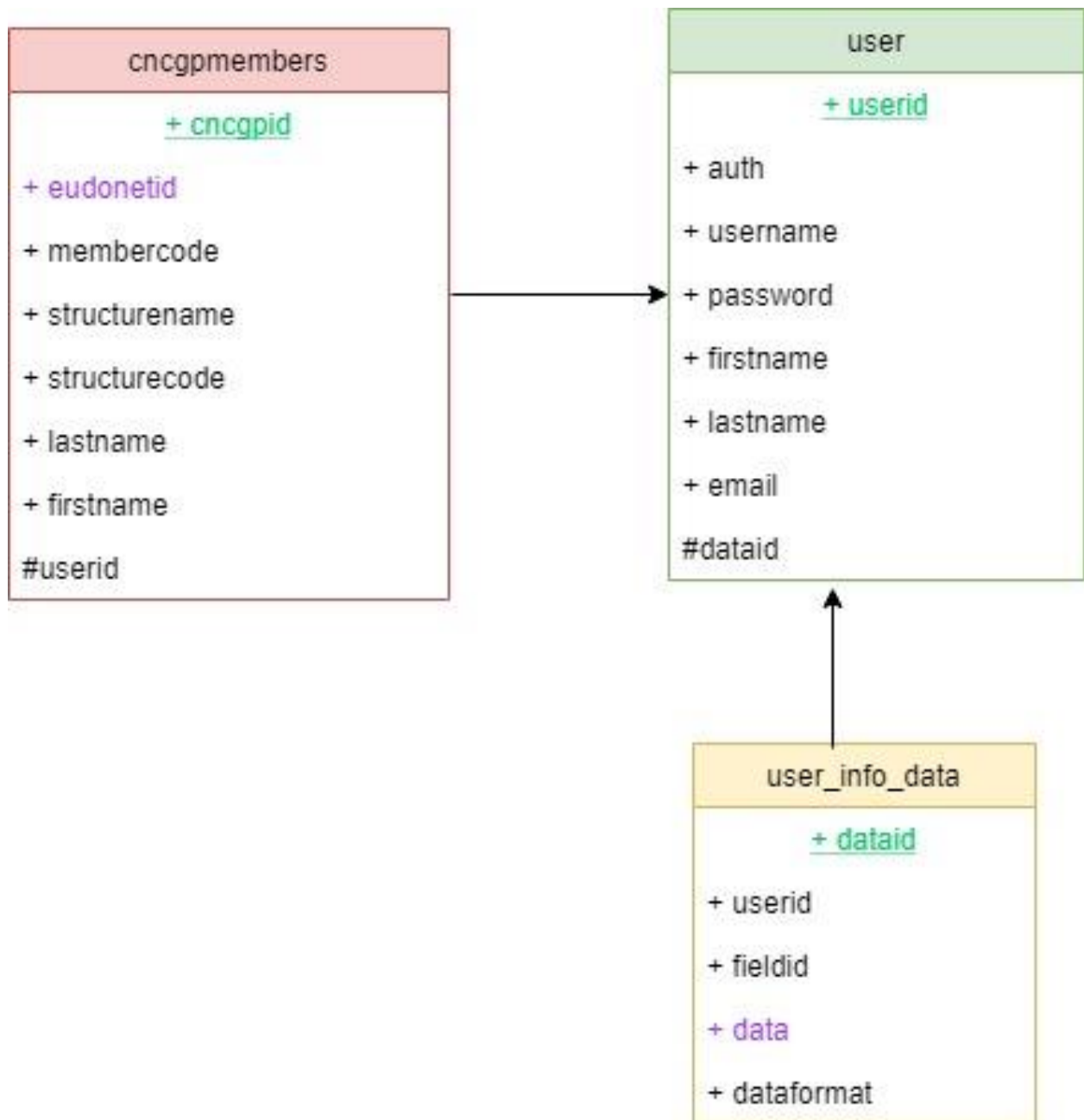
User\_info\_data : table où est stocké le numéro unique et immuable de chaque utilisateur qui est en structure ('eudonetid' référencé dans le champ 'data')

Le processus de membre ajouté : Si suite à l'import du .csv dans cncgpmembers un membre est créé, il faudra que celui-ci fasse une inscription sur le site, à l'aide de son code membre, pour que ses informations basculent et dans la table user et dans la table user\_info\_data.

Le processus de membre supprimé : Si un membre est supprimé de CNCGPmembers il est suspendu (mais toujours existant) dans la table user.



## 2. Modèle logique de données (MLD)



## IV. SPECIFICATIONS TECHNIQUES

### Plugin d'authentification CNCGP

#### 1. Outils et langages

Andil, étant spécialisé dans la mise à disposition de plateformes e-learning, utilise comme **LMS : MOODLE**. Un LMS ne se différencie pas beaucoup d'un CMS. En effet, plutôt que de la gestion de contenu, on parlera de gestion de cours avec la possibilité de mettre en place des quizz et des notes. Les élèves ont un ratio de réussite et peuvent prétendre à une certification une fois un certain score atteint.

Le logiciel de codage utilisé est l'**IDE PhpStorm** créé par JetBrains, sous licence Windows. Celui installé sur mon poste de travail a été configuré par les développeurs de la société pour convenir à la nomenclature, notamment concernant les erreurs et bonnes pratiques, propre à MOODLE. Je m'étais bien habituée à SublimText dans un cadre personnel et Visual Studio Code dans un cadre professionnel. Mais ça ne m'a aucunement dérangée de découvrir un IDE un peu plus poussé.

**Git** et **Wamp** étaient également indispensables. L'un pour pouvoir communiquer avec le repository de la société et l'autre pour me permettre de travailler en local, sur ma propre machine, en simulant un serveur web. Dans le même cadre, **PHPMyAdmin** était ma base de données locale, ne pouvant, pour des raisons évidentes, accéder à la base de données effective en production.

De manière plus générale j'ai été intégré à un planning d'attribution de tâches. **EasyRedMine** permet de calculer son temps de travail, ses horaires journaliers et de pouvoir communiquer avec les différentes équipes sur les projets qui nous sont assignés.

#### Nécessaire pour le mémoire

Pour les maquettes j'ai utilisé un site gratuit et très simple du nom **de Wireframe.cc**.

Concernant la réalisation du MCD et du MLD, **draw.io** était parfait. Je m'en étais déjà servis plusieurs fois pendant la formation, plus particulièrement en JAVA.

## 2. La tâche

EasyRedMine est un gestionnaire de tâches interne à l'entreprise, permettant à chacun de savoir qui est sur quel projet, et combien de temps prennent ces derniers à être réalisés. De même, une priorité d'exécution de la tâche peut être accordée selon l'urgence et la tenue des délais. Il y a toujours une description du projet et une personne assignée à sa réalisation. Sans s'attarder outre-mesure sur le fond, la forme est de cette manière :

### JurisCampus CNCGP - Evolution

#### #11227 - auth\_cncgp : développement initial

2019-01-09 16:55 - Arnaud Trouvé

Statut:	En cours	Date de début:	2019-01-07
Priorité:	Haute	Date d'échéance:	
Assignée:	David Aragon	% réalisé:	100%
Jalon:		Temps estimé:	0 heures
		Temps passé:	9 heures 30 minutes

#### Description

---

Authentification par code membre

##### Descriptif

Plugin d'authentification uniquement. Actif sur les sites où il sera possible d'acquérir un compte avec et sans code membre. Après avoir cliqué sur Connexion > Créer un compte, deux possibilités : **inscription par code membre** et **inscription sans code membre**.

Il y a un historique précis des actions effectuées sur la tâche. On peut également laisser des commentaires et des annotations, pour les collègues qui travailleraient avec nous sur la même tâche.

#### Historique

---

#22832 - 2019-01-09 16:56 - Ninon Morère

- Description mis à jour

#22833 - 2019-01-09 17:18 - Ninon Morère

- Description mis à jour

#22834 - 2019-01-09 17:20 - Ninon Morère

- Description mis à jour

#22887 - 2019-01-10 14:56 - Ninon Morère

- Description mis à jour

### 3. Le code

#### Plugin d'authentification CNCGP

##### a. Introduction

Malgré une utilisation « clé en main » tout comme un CMS pourrait l'être, les fonctionnalités qu'il y a en back-office dépendent d'un développement pointu. MOODLE utilise un langage basé sur le PHP, avec des subtilités propres à lui. En effet, il fonctionne, entre autres, avec des variables globales déjà définies dans ses librairies natives qu'il est possible d'utiliser pour plus de facilités.

##### b. Généralités

Dans un premier temps il faut comprendre qu'un plugin s'installe, se configure (si besoin est) et se désinstalle via l'interface d'administration MOODLE. Une fois l'installation faite, dans l'onglet 'authentification', on vient choisir dans une liste déroulante le plugin d'authentification que l'on souhaite utiliser, de manière à l'activer.

Ceci réglé, nous pouvons nous pencher sur le codage et les customisations nécessaires. Il est de notoriété publique, dans la communauté MOODLE, de ne jamais créer un plugin du début à la fin. MOODLE ayant fini de naître en 2001, il existe déjà une multitude de plugins plus ou moins compatibles avec les nouvelles versions, qui nous permettent de toujours nous inspirer, reprendre, limitant ainsi le temps dépensé et l'énergie à fournir. Bien évidemment, il n'est pas question pour autant de bâcler le travail.

Ici, notre inspiration vient de l'authentification par confirmation email, native à MOODLE. Le procédé d'origine est simple et quasiment similaire à notre projet : le membre remplit un formulaire pour s'enregistrer sur le site, il le valide, un mail lui est envoyé avec un lien à usage unique sur lequel il doit cliquer pour valider son compte. Une fois fait, il peut s'identifier à sa guise.

Notre but est de légèrement en changer la donne pour permettre à l'utilisateur d'arriver sur un choix (avec deux boutons) lorsqu'il cliquera sur 'créer son compte' : *Je possède un code membre* et *Je ne possède pas de code membre*. On créera alors réellement le codage nécessaire pour la partie *Je possède un code membre*, contrairement à la seconde partie, qui sera donc l'authentification originale par confirmation email abordée précédemment.

##### c. Installation de la table cncgpmembers

Dès que le plugin est installé, la table cncgpmembers est créée avec tous les champs associés. On passe par un fichier de type XML qui fait partie des premiers fichiers lus par MOODLE lors de l'installation des plugins.

Une fois fait, on peut aller vérifier dans sa base de données que la table a bien été créée.

#### d. Les fonctions

Les plugins d'authentification ont un fichier principal et appelé en premier lieu, et qui se nomme « auth.php » (un peu à l'égal de « index.php » sur un site web classique). C'est dans celui-ci que l'on code les fonctions principales au plugin et qui font appel à d'autres fichiers.

Lorsque l'utilisateur clique sur 'créer un compte' il se trouve devant un formulaire avec deux choix (fig. I, E. Maquette). C'est donc la fonction `signup_form()`, dans `auth.php` qui régit cela.

Variables globales →  
Formulaires utilisés →  
Appelle le formulaire →  
En cas d'erreur de chargement, retourne la fonction initiale →

```
/** Custom sign up form */  
  
public function signup_form() {  
  
    global $CFG, $type;  
  
    require_once($CFG->dirroot . '/auth/cnccgp/classes/is_member_form.php');  
    require_once($CFG->dirroot . '/auth/cnccgp/classes/member_signup_form.php');  
  
    $type = optional_param('typeaccount', 'selection', PARAM_TEXT);  
  
    if ($type === 'selection') {  
        return new is_member_form(null, null, 'post', "", array('autocomplete' => 'on'));  
    } else if ($type === 'membercode') {  
        return new member_signup_form(null, null, 'post', "", array('autocomplete' => 'on'));  
    } else if ($type === 'notmember') {  
        return new email_override_signup_form();  
    } else {  
        return parent::signup_form();  
    }  
}
```

`$CFG` = configuration files, variable faisant référence à la configuration initiale de MOODLE.

`$type` = permet de donner un paramètre optionnel aux deux liens, de manière à pouvoir créer une condition. Sans cela, il serait impossible de renvoyer sur les différents formulaires. Je l'ai déclaré en 'global' car je la récupère de la classe 'is\_member\_form.php'.

> Et n'oublions pas la syntaxe d'appel de formulaire :

```
function moodleform($action=null, $customdata=null, $method='post', $target='',  
$attributes=null, $editable=true)
```



**\$action** : l'url vers lequel le form renvoie au moment d'être submit.

**\$customdata** : ici, fait référence à un élément natif de MOODLE tel un quizz, un cours, une leçon, un post de forum...

**\$method** : la méthode avec laquelle le formulaire est soumis. MOODLE ne traite que les méthodes 'post'

**\$target** : cela faisait référence au cadre dans lequel le form était submit. Déprécié avec les nouvelles versions de MOODLE, on considère qu'il est aujourd'hui inutile de le remplir. Toutefois, il existe toujours par soucis de compatibilité avec des versions antérieures non mises à jour et l'utilisant encore.

**\$attributes** : ici nous avons soumis le formulaire à l'auto-complétion. Mais il existe divers paramètres pouvant être sollicités ici.

**\$editable** : définie sur TRUE automatiquement. Permet de ne pas verrouiller la complétion du formulaire par l'utilisateur.

### e. Le formulaire d'authentification

L'utilisateur clique sur *J'ai un code membre* et accède à un formulaire de données (fig. II, E. Maquette). Celui-ci ne contient que deux champs et un bouton : Code membre + adresse email.

La classe appelée est `member_signup_form` **extends** `moodleform`. Moodleform est une classe native à moodle qui a pour but de générer, protéger et soumettre les formulaires ainsi que les informations entrées dans chaque champ. A l'intérieur de cette classe nous avons créé la fonction `definition()`. Elle est très majoritairement suivie d'une fonction `validation()`. Cette fonction est appelée quand le formulaire est soumis. Nativement, elle 'écoute' `definition()` et attend une méthode en POST.

Voyons de plus près le contenu de `definition()` :

	<pre>\$mform = \$this-&gt;_form;</pre>
titre image	<pre>\$mform-&gt;addElement('html', get_string('auth_desc_membercode', 'auth_cncgp')); \$mform-&gt;addElement('html', html_writer::img(...));</pre>
champ caché	<pre>\$mform-&gt;addElement('hidden', 'typeaccount', 'membercode'); \$mform-&gt;setType('typeaccount', PARAM_TEXT);</pre>
input code	<pre>\$mform-&gt;addElement('text', 'membercode', get_string('membercode', 'auth_cncgp')); \$mform-&gt;setType('membercode', PARAM_TEXT); \$mform-&gt;addRule('membercode', get_string('missingusername'), 'required', null, 'client');</pre>
input email	<pre>\$mform-&gt;addElement('text', 'memberemail', get_string('email', 'auth_cncgp')); \$mform-&gt;setType('memberemail', core_user::get_property_type('email')); \$mform-&gt;addRule('memberemail', get_string('missingemail'), 'required', null, 'client');</pre>
bouton	<pre>\$this-&gt;add_action_buttons(false, get_string('btn_send', 'auth_cncgp'));</pre>

Il est nécessaire de déclarer la construction de formulaire dans la variable `$mform`. Cette nomenclature est à respecter, au regard des bonnes pratiques de MOODLE. Le `$this` cible l'objet courant qui est, ici, la classe `member_signup_form` et construit donc un formulaire en objet.

`addElement` : c'est l'élément html que l'on souhaite mettre dans notre formulaire (texte, label, input...)  
`setType` : définit en paramètre le type de l'élément (text, raw, clean, notags, int, action)  
`addRule` : définit une contrainte vis-à-vis de l'élément  
`add_action_buttons` : le bouton

En premier lieu a été créé un titre avec juste en dessous une image (en base64, dont j'ai masqué les lignes de code interminables).

Ensuite il fallait un champ caché qui nous permettait de savoir quel paramètre contenait la page actuelle. Ici, nous indiquons que le 'typeaccount' est 'membercode'.

Pourquoi un champ caché ? Ce dernier était nécessaire pour emmener sur le bon formulaire, depuis la page d'accueil de création de compte (schéma page 18).

Les trois prochaines lignes indiquent l'input où il faut renseigner le code membre, son type (du texte) et sa contrainte (information requise). Idem pour le bloc d'en dessous mais cette fois-ci, on précise dans le type qu'il s'agit d'une adresse email.

#### f. Communiquer avec la base de données

MOODLE exécute la requête de connexion en base de données via une variable globale qui se nomme `$DB`. Cette variable instancie la classe native `moodle_database.php` qui permet d'établir toutes les requêtes SQL (sécurisées) avec toutes sortes de BDD tel que PHPMyAdmin, MongoDB, PostgreSQL, etc. Il faut donc en respecter la syntaxe au risque de créer des erreurs et, plus généralement, des failles de sécurité.

```
public function validation($data, $files) {  
    // Override parent data validation.  
    global $DB;
```

Les requêtes SQL changent donc légèrement, pour pouvoir s'adapter à la composition de MOODLE, un peu comme si on utilisait un framework.

L'exemple d'un « SELECT » :

```
// We take user's informations in 'cncgpmembers' table.  
$results = $DB->get_records('cncgpmembers', ['membercode' => $mc]);
```



Nous avons l'habitude d'être confronté à cette méthode :

```
SELECT 'membercode' FROM 'cncgpmembers' WHERE $mc = ?
```

Avec MOODLE cette méthode s'est vue revisitée pour être fonctionnelle. Il a été instauré une syntaxe, compréhensible et commune à tous les langages SQL, permettant ainsi un large choix d'associations d'outils de travail.

Comme détaillé plus haut, on entend donc `$DB` la variable globale qui effectue toute la connexion à la base de données, paramétrée pendant la configuration MOODLE.

`Get_records` est l'attribut qui permet d'aller chercher les informations en base, la traduction littérale de notre SELECT.

La table *cncgpmembers* est citée et suivie d'un tableau (que nous pouvons aussi écrire `array('membercode' => $mc)`) qui contient l'information à chercher ainsi que la condition de résultat.

Ici nous cherchons donc à récupérer la liste des membres, dont le code membre équivaut à celui entré dans le formulaire précédant (\$mc est la valeur récupérée de l'input quand le formulaire est soumis).

**En savoir plus :** [La documentation](#) MOODLE vis-à-vis des formulaires propose, dans ses exemples, des solutions très techniques pour permettre d'aller chercher l'information pertinente, sans risquer de se perdre dans des conversions pour chaque base de données.

Ici il était question d'un SELECT mais on peut faire un delete, un update, etc. La valeur 'get\_record' changera alors pour 'delete\_records' ou encore 'update-records'.

## g. Le style

Le design du site étant géré par un template prédéfini appelé 'thème', aucune modification ou création n'a été nécessaire. Il est possible de surcharger un thème MOODLE mais, dans ce cas-ci, c'était inutile.

## 4. Le rendu final

A voir sur <https://cncgp.juriscampus.fr>

Connexion > créer un compte

## Réfection graphique d'un plug-in de statistiques

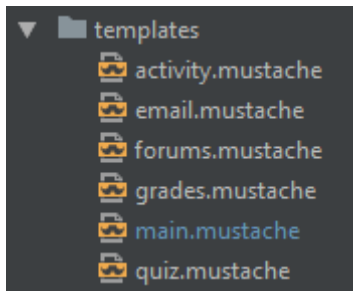
### 1. Les templates 'moustache'

Un nom amusant pour un élément qui définit la manière dont le contenu des pages sera organisé et embelli. En effet, ceci vient du fait que le contenu entre balises est traduit en variables, et ces variables sont entourées d'accolades, gentiment appelées des 'moustaches'.

```
<h5> {{# str }} activity_chart, block_weekly_report {{/ str }}</h5>
```

 Un titre en H5.

La vue générale dans l'explorateur PHPSTORM.



La syntaxe reste cependant assez simple. Comme montré ci-dessus, on garde les balises classiques et on rajoute les éléments qui seront, en définitif, notre contenu. En fonction de la donnée que l'on veut afficher, on demandera un type défini (ex : *str* qui est pour string donc tout ce qui est texte), ou un type que nous auront créé nous même (on pense à un calendrier, des posts de forum ou un graphique, notamment).

Fichier : forums.mustache

```
{{^posts_read}}
  {{# str }} forums_no_posts_read, block_weekly_report {{/ str }}
{{/posts_read}}
```

Donne :

Aucun message lu

Le type d'élément concerne les sujets lus sur les forums des cours où l'utilisateur est inscrit (*posts\_read*). L'accent circonflexe définit la condition « Si aucun post lu » et la 'balise' `{{# str}}` définit la suite : « alors afficher la chaîne de caractère suivante ».

## 2. Mise en place

La documentation de Bootstrap étant bien faite, je me suis servie de ce qu'elle pouvait proposer pour mettre en forme le tout.

Le code initial fonctionnant à la perfection je n'ai touché qu'aux templates mustache, le cœur du design. J'ai tout d'abord réorganisé les éléments en mettant la *semaine en cours* tout en haut, puis *l'activité globale*, ensuite *les notes*, *les forums*, *les quiz* et enfin *le graphique d'activité*.

Il y a eu une première version qui ressemblait à cela :

RAPPORT HEBDOMADAIRE

Semaine du **11.02.19** (Changer la date)  
Votre activité : 1 connexion(s) et 47 page(s) vue(s)

Notes

1 note(s) cette semaine

Note la plus élevée: 0

Note la plus basse: 0

Moyenne des notes: 0

[Graphique d'activité \(clic\)](#)

Forums

Aucun message lu  
1 message(s) écrit(s) dans 1 discussion(s)

Quiz

Aucune tentative

Le code correspondant ressemble à cela pour la partie générale (celle qui gère les blocs notes, forums, etc...)

### main.mustache

```
25 <h1>{{# str }} header, block_weekly_report {{/ str }} <span class="text-warning">{{date}}</span></h1>
26
27 <a class="m-y-1 btn btn-primary" data-toggle="collapse" href="#calendar-{{uniqid}}">
28   {{# str }} calendar_form, block_weekly_report {{/ str }}
29 </a> <a class="toggle-activity btn btn-secondary" aria-expanded="false" data-toggle="collapse" href="#blocks">
30   {{# str }} btncollapse, block_weekly_report {{/ str }}
31 </a>
32 <br>
33 <div class="calendar {{{collapse_var}}}" id="calendar-{{uniqid}}">
34   {{{calendar_form}}}
35 </div>
36
37 {{< block_weekly_report/activity>}}
38 {{/ block_weekly_report/activity}}
39
40 <div class="row main" id="blocks">
41   {{< block_weekly_report/grades>}}
42   {{/ block_weekly_report/grades}}
43
44   {{< block_weekly_report/forums>}}
45   {{/ block_weekly_report/forums}}
46
47   {{< block_weekly_report/quiz>}}
48   {{/ block_weekly_report/quiz}}
49 </div>
50
51 <h5> {{# str }} activity_chart, block_weekly_report {{/ str }}</h5>
52
53 <a data-toggle="collapse" href="#activity-chart-{{uniqid}}"> {{# str }} activity_chart_click, block_weekly_report {{/ str }}</a>
54 <div id="activity-chart-{{uniqid}}" class="activity {{{collapse_var}}}">
55   {{{activity_chart}}}
56 </div>
```

Très rapidement on distingue plusieurs parties :

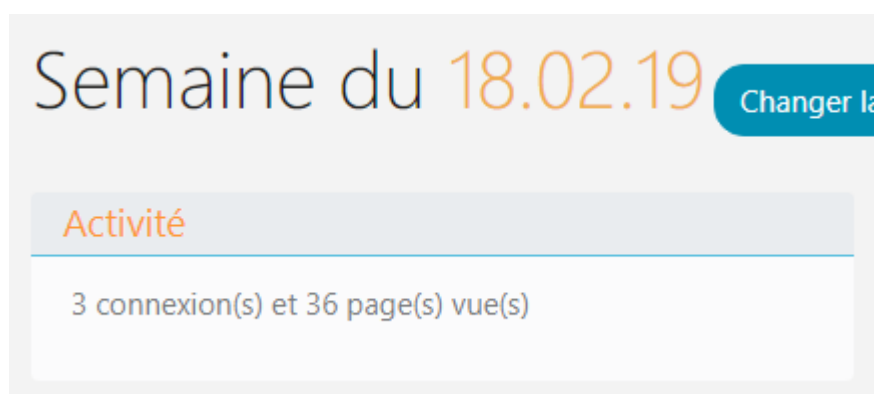
- Le titre en H1
- Les deux liens <a> du calendrier
- La partie du calendrier
- Tous les blocs (block\_)
- Le graphique d'activité

« header, block\_weekly\_report » se rapporte au fichier langue du plugin. Il se réfère à l'ID se nommant « header » pour récupérer la chaîne de caractères et l'afficher. La syntaxe est la même pour tout MOODLE comme nous avons pu le voir quelques fois, dans mes explications précédentes.

**Note :** Les variables orange entre accolades se rapportent à des parties de code auxquelles je n'ai pas eu accès pour ce plug-in ci. Comme je n'étais qu'en charge du design, l'accès m'était restreint.

Mon travail a consisté à rajouter des balises HTML connues ainsi que les class (en vert). Je n'ai pas touché les id ni même les informations entre accolades. Les « data-toggle » sont les informations cachées qui ne sont révélées que lors du clic sur un bouton.

Par ailleurs, je n'ai utilisé que la nomenclature de Bootstrap. Ainsi je suis passée de quelques onglets qui s'ouvraient et se fermaient en un tableau un peu plus coloré. Cependant, ce design ne faisait pas l'unanimité. On distinguait peu l'activité, il y avait une information inutile dans « Notes » et en général, il était beaucoup trop imposant. Ayant pour vocation d'être en haut du tableau de bord, il devait prendre le moins de place possible, pour que les élèves puissent accéder aux cours rapidement. Les marges, la grosseur de la typographie, ainsi que tous les éléments environnants devaient être revus pour quelque chose de plus compact. Le bloc « activité » a été déplacé dans une petite carte, comme le reste et le H1 a été modifié pour être rétréci.



### 3. Version finale

On m'avait demandé de faire quelques essais d'une version très épurée et très légère mais elle n'a pas été retenue. A côté j'ai peaufiné celle plaisant à tous, et je l'ai rendu responsive. Lorsqu'on passe à une taille inférieure à 1000px, un bouton « afficher/masquer votre activité » apparaît pour permettre de gagner en espace, ergonomie et fluidité sur tablettes et mobiles.

RAPPORT HEBDOMADAIRE

Semaine du 11.02.19

Changer la date

Activité

7 connexion(s) et 120 page(s) vue(s)

Graphique d'activité (clic)

Notes

1 note(s) cette semaine

Note la plus élevée: 0

Note la plus basse: 0

Moyenne des notes: 0

Forums

Aucun message lu

1 message(s) écrit(s) dans discussion(s)

Quiz

Aucune tentative

andilcampus

Aucune activité

RAPPORT HEBDOMADAIRE

Semaine du 11.02.19

Changer la date

Montrer/Cacher votre activité

Activité

7 connexion(s) et 120 page(s) vue(s)

Notes

Aucune note rapportée cette semaine

Forums

Aucun message lu

1 message(s) écrit(s) dans discussion(s)

Quiz

Aucune tentative

Graphique d'activité (clic)

## 4. Améliorations

Une dizaine de jours plus tard, lorsque je réalisais un plug-in qui devait gérer un envoi de mail automatique, nous avons parlé à nouveau du design de ce bloc. Le concepteur développeur web et logiciels qui s'occupe de valider les projets, a évoqué des améliorations à apporter, beaucoup plus techniques. Il y avait, entre autres détails, donner la possibilité aux utilisateurs de laisser des blocs masqués ou affichés en permanence en fonction du besoin et du souhait. Autre fonctionnalité, pouvoir personnaliser la position des cartes, et pourquoi pas en enlever ou en rajouter en fonction de ce qui nous intéresse. Malheureusement, mon stage étant presque fini et ces améliorations n'étant pas dans les priorités les plus urgentes de mes collègues, il ne m'a pas été possible de travailler dessus. Cependant, j'ai pu constater que même les projets en production sont toujours améliorés, et améliorables. La réflexion sur le perfectionnement des outils est constante.



## V. FONCTIONNALITES

### Débug d'un plug-in d'importation de template de certificats

#### 1. Un point sur les restrictions et les certifications

Pour valider une certification et avoir un diplôme il est possible de mettre en place des restrictions. Ce sont des contraintes que l'étudiant doit respecter et remplir, pour pouvoir accéder à certains cours, des tests ou encore des documents officiels.

---

#### Questionnaire de révision

**Accès restreint** Non disponible à moins que : L'activité **La réglementation Abus de marché au sein de l'UE** soit marquée comme achevée

---

Une fois toutes ces contraintes passées, lorsque l'utilisateur cliquera sur le lien, une certification se générera à partir d'un modèle PDF (que les administrateurs auront déposé sur la plateforme) et où les informations liées à l'utilisateur sont reportées automatiquement.

#### 2. Le bug

Et dès qu'un administrateur souhaitait modifier les paramètres de certification pour enlever ou rajouter un template PDF, cela levait une exception que voici. Les erreurs MOODLE sont découpées en trois parties :

Exception : Call to undefined method HTML\_QuickForm\_Error::setSelected()

[Plus d'informations sur cette erreur](#)

Exception : l'erreur intelligible

**Debug info:**

Error code: generalexceptionmessage

Le Débug : le type d'erreur

**Stack trace:**

- line 161 of \mod\templatecertificate\mod\_form.php: Error thrown
- line 204 of \lib\formslib.php: call to mod\_templatecertificate\_mod\_form->definition()
- line 99 of \course\moodleform\_mod.php: call to moodleform->\_\_construct()
- line 141 of \course\modedit.php: call to moodleform\_mod->\_\_construct()

Stack trace : le suivie d'erreur pour pointer là où ça a bloqué. MOODLE pointe ses propres librairies (natives) mais le problème part avant tout du plugin.

---

Classe initiale : **mod\_form** ► C'est la classe qui va être appelée dès qu'on va vouloir gérer les paramètres 'template' des certificats.

#### a. Les variables d'origine

```
public function definition() {  
    global $CFG, $USER, $DB, $PAGE;  
    $certificate = null;  
    if ($PAGE->cm != null) {  
        $certificate = $DB->get_record('templatecertificate', array('id' => $PAGE->cm->instance), '', MUST_EXIST);  
    }  
}
```

definition() est le corps de la classe mod\_form

On définit une variable `$certificate` auquel on assigne une valeur nulle. Ensuite on enclenche une condition qui va vérifier que :

Si la valeur de `cm` (= course\_module, propriété native de MOODLE) qui a dans les informations de la page en cours (`$PAGE`) est différent de la valeur 'null'

Alors on exécute la requête de base de données qui est de sélectionner l'enregistrement où l'`id` correspondrait au numéro `instance`, contenu dans le `cm`, contenu dans les informations de la page actuellement utilisée. Et de stocker cette requête dans la variable `$certificate`.

#### b. La stack trace

```
if ($certificate) {  
    $mform->setDefault('usetemplate', 'checked');  
    $mform->getElement('selectedtemplate')->setSelected($certificate->templateid);  
}
```

La ligne 161 qui a levé l'exception. On voit ici que la condition n'a pas de SINON. On vérifie juste si le template existe. Si oui, alors on simule l'élément HTML 'select' (liste déroulante) avec tous les modèles template déjà mis sur la plateforme. Dans le cas contraire, ça lève donc une erreur.

### 3. Ma solution

```
if ($certificate) {  
    if (count($selecttemplates) == 0 || $DB->record_exists('templatecertificate_pdf', array($certificate->templateid))) {  
        echo get_string('error_template_message', 'templatecertificate');  
    } else if (count($selecttemplates) > 0) {  
        $mform->setDefault('usetemplate', 'checked');  
        $mform->getElement('selectedtemplate')->setSelected($certificate->templateid);  
    } else {  
        return false;  
    }  
}
```

`$selecttemplate` est un tableau qui a été créé pour mettre tous les enregistrements de template existants (et non seulement sur la page au moment de l'ouverture des paramètres de gestion des templates).

J'ai donc créé une condition qui permet de dire que :

Si le tableau n'a aucun enregistrement OU que l'id du template de la table templatecertificate\_pdf est à zéro, alors on marque un message d'erreur. Ce qui permet à l'administrateur d'accéder à la page des templates du cours, même si le PDF n'est plus disponible ou est corrompu et ce, sans lever d'erreur.

>\_ J'ai quand même tenu à préciser la condition initiale dans un SINON SI car, très souvent, tout mettre dans le ELSE peut être trompeur, le ELSE étant une partie de condition très vague.

## VI. CONCLUSION

J'ai choisi d'expliquer en détails les missions qui me paraissaient pertinentes pendant le stage, tout en faisant le tour du travail que l'on m'a donné à effectuer. Cela m'a permis de voir, en conditions réelles, comment pouvait naître un projet et comment il évoluait. Je me suis également rendue compte que tout n'était pas parfait d'office. Il y avait de nombreuses retouches à faire avant que ce ne soit validé et fonctionnel.

J'ai eu la chance de travailler en binôme et quelques fois en trinôme. Les personnalités de chacun nous enrichissent, par le partage de nos idées, et par la découverte des visions propre à chacun. Il est très intéressant de discuter, et même parfois de confronter nos opinions afin de faire mûrir les réflexions.

J'ai pu confirmer le plaisir et la passion que m'inspirait ce métier, qui vient compléter et développer mon cursus initial. J'aimerais me diriger dans le front-end ou plus encore, me spécialiser dans l'application mobile. Ce sont là des champs d'application du métier qui me passionnent tout particulièrement et dans lequel je me vois bien évoluer.

Pour finir, je retiens une conclusion positive de cette expérience enrichissante, et souhaite vraiment approfondir ma passion et mes connaissances. Je remercie encore une fois, en ce sens, l'Adrar et Andil pour m'avoir donné la chance de pouvoir me lier à nouveau avec le monde du numérique, que je pensais définitivement fermé pour moi.