

2014

Rapport d'activités



Rémi PERAL
Lycée Ozenne
8/25/2014

Sommaire

1) Analyse du besoin

1.a) L'ENFA

1.b) Le projet

2) Développement de l'application

2.a) Introduction au mvc

2.b) Le modèle

2.c) Le contrôleur

2.d) La vue

3) Bilan

3.a) Projet fini

3.b) Bilan personnel

1) Analyse du besoin

1.a) L'ENFA

L'Ecole Nationale de Formation Agronomique (ENFA) de Toulouse-Auzeville est un des établissements publics d'enseignement supérieur et de recherche du Ministère de l'Agriculture, de l'Agro-Alimentaire et de la Forêt.

La plupart des machines, dont celles qui m'ont permises de réaliser ce stage étaient équipées du système d'exploitation Windows 7, la suite bureautique Microsoft Office ainsi que l'utilitaire de développement Notepad++.

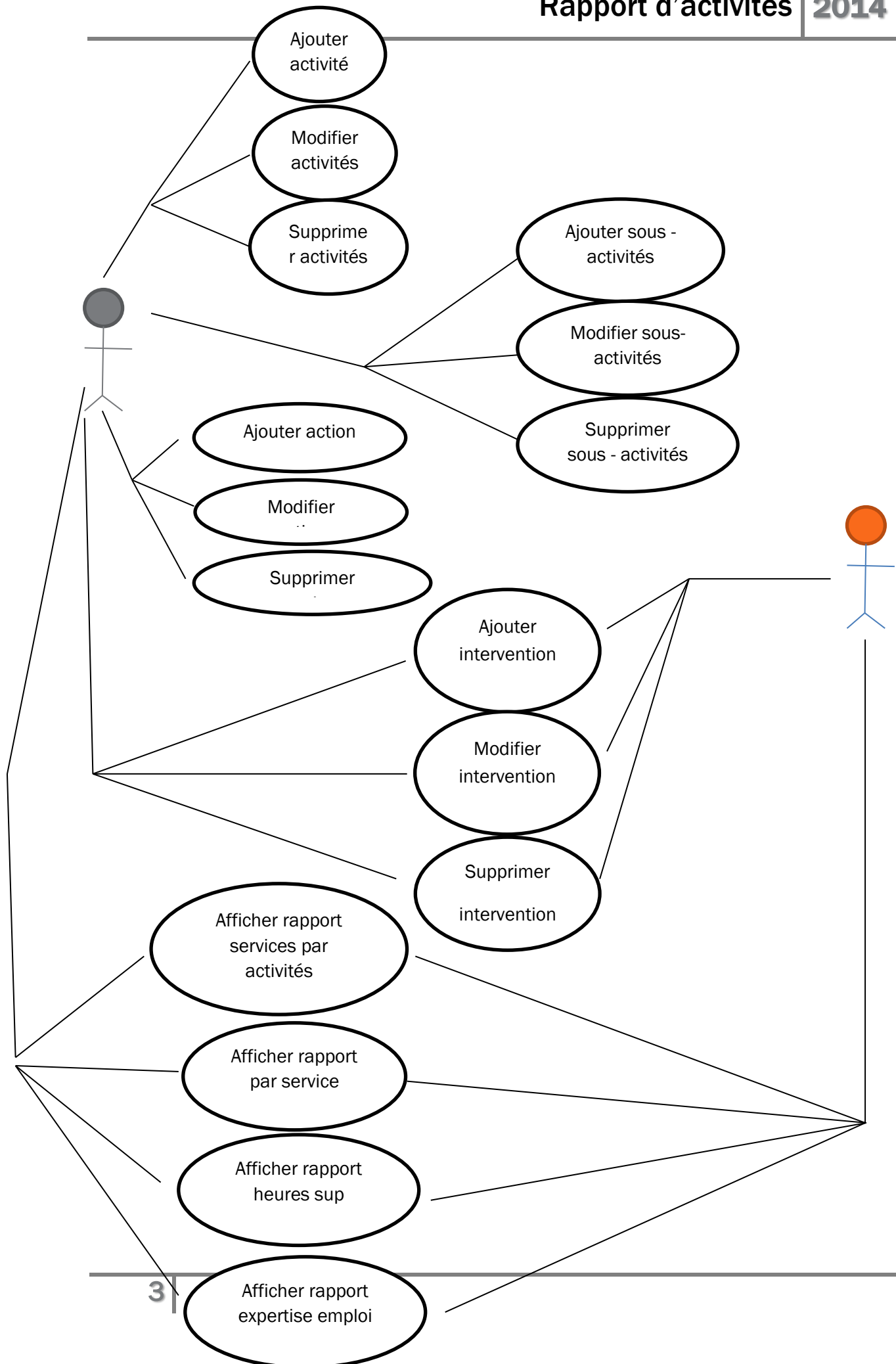
A ma disposition, une session administrateur afin de pouvoir télécharger les éventuelles technologies manquantes et pouvoir effectuer les tests nécessaires à la réalisation du projet.

1.b) Le projet

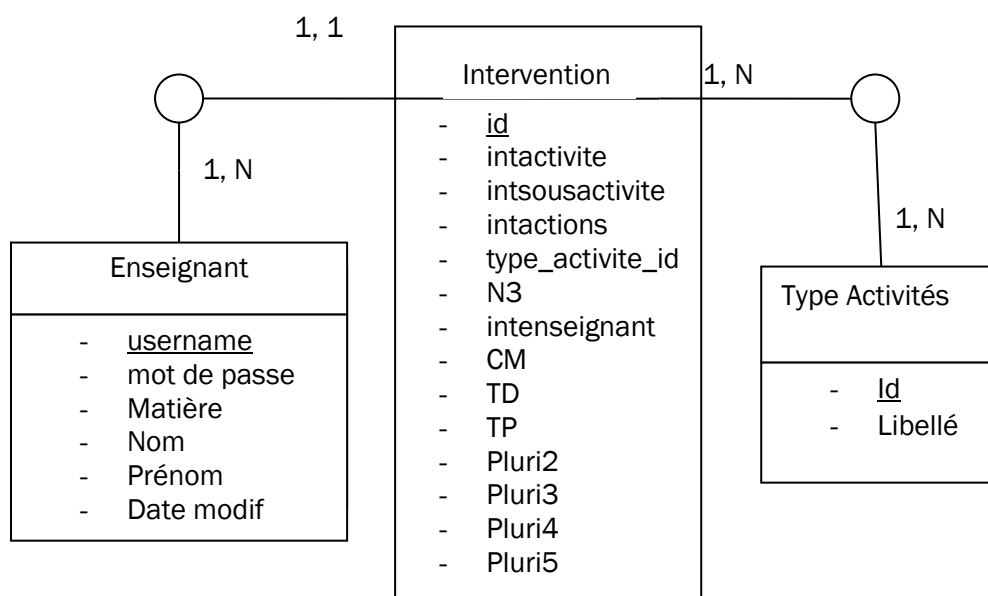
Le but de ce stage était de développer une application permettant aux enseignants de l'école de pouvoir consulter leur total d'heures effectuées, le type de ces heures ainsi que leurs conditions. L'application devait afficher ainsi la différence entre les heures à réaliser et les heures réalisées afin de pouvoir par la suite calculer le montant des heures supplémentaires. Il devait être possible pour les enseignants de commenter chaque ligne de la fiche afin que le gestionnaire de l'application puisse y apporter des modifications en cas d'erreurs. Le gestionnaire de l'application, habitué au format Excel, devait pouvoir alimenter la base via un formulaire sous ce format. Il devait aussi pouvoir consulter les éventuels commentaires laissés par les enseignants afin de remédier au litige engagé. Afin de vérifier, si éventuelle modification eut été faite par l'administrateur, l'enseignant devait en être averti durant sa connexion sur l'application.

Le formulaire devait être divisé en « interventions », qui sont en fait le nom attribué au type d'enseignement réalisé. Ces interventions comportent des activités, des sous-activités ainsi que des actions, qui permettent de spécifier et préciser au mieux le type d'intervention pratiqué. Chaque intervention est donc divisé, en plus de ces trois parties, par un troisième niveau, par un type d'activité, ainsi que par le nom de la condition dans lequel s'est déroulé le cours ; si c'est un TP, un TD, un contrôle, si c'est un cours donnée à plusieurs... Chaque type de condition donnant droit à un pourcentage de paiement différent. L'application finale devait donc afficher le nombre d'heures réalisées pondérées par type d'activités, par types d'interventions, par type de cours ainsi que le total final de toutes les heures.

Il a ensuite été décidé de créer une fiche des cas d'utilisations pour aider au développement et à l'analyse de la situation.



Afin de pouvoir réaliser ce projet, il a été décidé de développer une application web, accessible par le corps enseignant et l'administrateur grâce à un système de connexion. L'application serait hébergée sur le serveur et isolé sur la partie intranet de l'école. Une base de données, créée sur ce même serveur, alimenterait l'application, et devait être gérable par l'administrateur via un formulaire Excel.



Il a été décidé que la base de données serait développée sous MySQL, format déjà utilisé par l'école. L'application web quant à elle, serait développée en Php grâce à un framework, Cake Php, afin d'accélérer le développement de l'application, le framework comprenant déjà une structure et une mise en page ergonomique. L'application de conversion formulaire Excel / BDD MySQL serait développée en VBA, cette fois par un autre stagiaire.

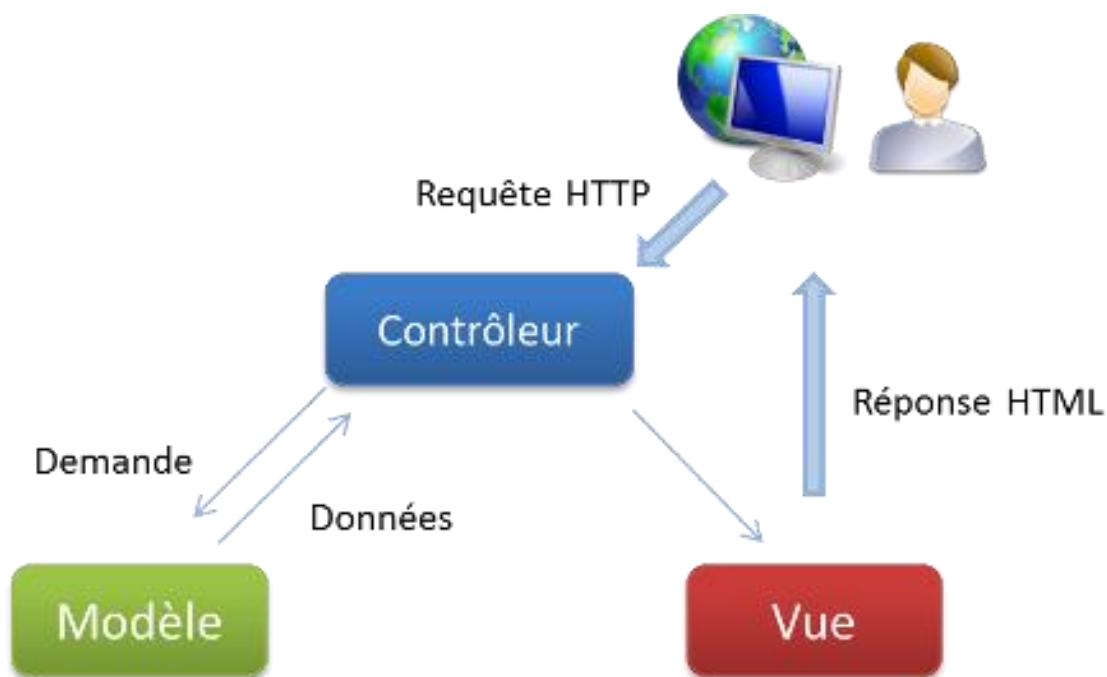
2) Développement de l'application

2.a) Introduction au modèle mvc

Pour développer cette application, nous avons utilisé un framework du nom de CakePhp, le rapport temps d'apprentissage/efficacité étant celui jugé meilleur sur le marché, suivant un motif de conception Modèle-Vue-Contrôleur (MVC). Ce mode de programmation permet d'en séparer les différents composants.

Ce modèle regroupe donc les fonctions nécessaires en trois catégories :

- 1) un modèle (modèle de données),
- 2) une vue (présentation, interface utilisateur)
- 3) un contrôleur (logique de contrôle, gestion des événements, synchronisation)

**Le modèle :**

Il représente le cœur de l'application. Son rôle est de s'occuper du traitement des données et des interactions avec la base de données. Il regroupe la gestion de ces données et est responsable de leur intégrité. Il comporte également des méthodes standards pour mettre à jour ces données, ainsi que des méthodes pour les récupérer. Les résultats qu'il renvoie ne s'occupent pas de la présentation, le modèle n'ayant aucun lien direct vers le contrôleur, tout ceci étant traité par le patron Observateur.

La vue :

C'est la fonction qui permet à l'utilisateur d'interagir avec l'application. Il est chargé de présenter les résultats renvoyés par le modèle, et de recevoir les actions qu'effectue l'utilisateur. Ces éléments sont ensuite envoyés au contrôleur.

Le contrôleur :

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jours.

2.b) Partie modèle

Le modèle est le reflet d'une table contenue dans une base de données. Grâce au patron Observateur, Cakephp sait, en respectant les conventions de nommage, la table qui correspond au modèle. Il faut cependant indiquer dans le code du modèle les liens et les jointures entre les tables concernées dans le code.

```

1 <?php
2 class Intervention extends AppModel {
3     var $belongsTo= "TypesActivite";
4 }?>

```

Le procédé pour la table qui permet l'authentification et l'enregistrement d'un utilisateur est quelque peu différent. Il faut inclure dans le modèle les conditions d'enregistrement, les particularités qu'il compose. Ces conditions sont enregistrées dans des tableaux. Le programme sait ainsi comment réagir quand elles ne sont pas respectées.

```

1 <?php // app/Model/User.php
2 App::uses('AppModel', 'Model');
3 App::uses('SimplePasswordHasher', 'Controller/Component/Auth');
4 class User extends AppModel {
5     public $name = 'User';
6     public $validate = array(
7         'username' => array(
8             'required' => array(
9                 'rule' => array('notEmpty'),
10                'message' => 'Un nom d\'utilisateur est requis'
11            )
12        ),
13        'password' => array(
14            'required' => array(
15                'rule' => array('notEmpty'),
16                'message' => 'Un mot de passe est requis'
17            )
18        ),
19    );

```

Les données traitées étant sensible, il a été décidé d'inclure un programme de cryptage de mot de passe au moment de l'enregistrement de l'utilisateur. Ainsi, les mots de passes visibles dans la base de données associé au nom des utilisateurs, n'est pas celui entré par l'enseignant.

```

    public function beforeSave($options = array()) {
        if (isset($this->data[$this->alias]['password'])) {
            $passwordHasher = new SimplePasswordHasher();
            $this->data[$this->alias]['password'] = $passwordHasher->hash(
                $this->data[$this->alias]['password']
            );
        }
        return true;
    }
}

```

2.c) Partie contrôleur

C'est dans le contrôleur que les requêtes permettant de récupérer les données sont exécutées. Il est possible, lorsqu'elles sont complexes, de récupérer directement les données dans la base, mais la manière la plus simple est de passer par le modèle. Chaque requête est stockée dans une fonction, fonction que l'on peut appeler dans la vue quand on le désire.

```

1 <?php
2 class FichesController extends ApplicationController {
3
4     function index($nom)
5     {
6         $nom;
7         $this->set('noms', $nom);
8     }
9

```

Fonction simple qui prends en entrée un nom et qui permet de renvoyer toutes les données correspondantes à ce nom du modèle Intervention

Fonction complexe qui permet de renvoyer des données particulières dont le nom (intenseignant dans la table) est le même de la table interventions.

```

10 function interventions($nom)
11 {
12
13     return $groupesActiv = ClassRegistry::init('Intervention')->find('list', array( 'fields' =>
14     array('Intervention.intsousactivite', 'Intervention.intenseignant', 'Intervention.intactivite' ), 'conditions' =>
15     array("Intervention.intenseignant = $nom")));
16

```

Chaque contrôleur associé à une vue particulière est basée ainsi sur le même principe, seules les données extraites et les requêtes varient d'un contrôleur à un autre, en fonction de la partie à traiter.

2.d) Partie vue

Chaque vue, chaque écran est relié à un contrôleur. Un contrôleur peut avoir plusieurs vues. Les vues, en fonction de leur utilité, sont regroupées dans les dossiers qui correspondent à un même contrôleur.

administrations	10/06/2014 15:30	Dossier de fichiers	
Elements	10/06/2014 09:20	Dossier de fichiers	
Emails	10/06/2014 09:20	Dossier de fichiers	
Errors	10/06/2014 09:20	Dossier de fichiers	
fiches	10/06/2014 15:30	Dossier de fichiers	
Helper	10/06/2014 09:20	Dossier de fichiers	
interventions	10/06/2014 15:30	Dossier de fichiers	
Layouts	10/06/2014 09:20	Dossier de fichiers	
Pages	10/06/2014 09:20	Dossier de fichiers	
Scaffolds	10/06/2014 09:20	Dossier de fichiers	
users	17/06/2014 16:14	Dossier de fichiers	

add.ctp	18/06/2014 10:55	Fichier CTP
login.ctp	18/06/2014 11:02	Fichier CTP

C'est dans la vue que l'on traite les données récupérées dans le contrôleur. On peut ainsi stocker les résultats dans une variable.

```

37 $date = $this->requestAction('/interventions/date');
38 $calculs = $this->requestAction('/interventions/calculs');
39 $somme = $this->requestAction('/interventions/somme');
40 $action = $this->requestAction('/interventions/actions');
41 $compte = $this->requestAction('/interventions/compte');
42 $nbCompte = count($compte);

```

Les données stockées variables étant des tableaux, il est obligatoire, pour les exploiter, de créer des boucles qui permettent de les afficher.

```

55
56 ?></td><td style = "width:320px;"><?php
57 for($j = 0; $j < $acti; $j++)
58 {
59     ?><?php
60     echo $sousact[$j];
61
62     for($f = 0; $f < $nbCompte; $f++)
63     {
64         if($compte[$f]['interventions']['intsousactivite'] == $sousact[$j])
65         {
66             echo "</br></br>";
67         }
68     }
69

```

Il est également possible de saisir des informations dans la vue, qui seront ensuite envoyées au contrôleur, qui se chargera d'exécuter la requête permettant de mettre à jour la base de données.

```

if($calcul['Intervention']['intactions'] == $posAction[$h])
{
    echo $this->Form->create(array('url' => array('controller' => 'interventions', 'action' => 'edit', 'div' => false, 'type' => "hidden")));

    echo $this->Form->input('id', array('type' => 'hidden', 'div' => false, 'value' => $calcul['Intervention']['id']));
    echo $this->Form->input('commentaires', array('label' => false, 'div' => false));
    echo" ";

    echo $this->Form->end('Valider', array('div' => false));
}

```

Il suffit d'indiquer au programme où la mise à jour doit être faite, dans quelles conditions etc...

3) Bilan

3.a) Projet fini

Page d'authentification

Vous ne pouvez pas accéder à cette page

Veuillez entrer votre identifiant ainsi que votre mot de passe

Username*

Password*

Connexion

Fiche des heures

N0 Activités	N1 Sous Activité	N2 Actions	N3	Types	CM	TD	Plu2	Plu3	Plu4	Plu5	Total Heures	Pond	Commentaires	
FI enseignant	Externes, IAE	Ext PV, IAE PV, VV	Sem 38	Ens	1,5	0	0	0	0	0	2,2		<div>Valider</div>	
		Réservés	Rés MSP	Sem 49	Ens	0	0	2	1,5	0	0	2,2		<div>Valider</div>
	Externes	Rés Transversal	Sem 38	Ens	0	2	0	0	0	0	2,0		<div>Valider</div>	
				Sem 49	Ens	0	4	0	0	0	0	4,0		<div>Valider</div>
		Rés EPS	Sem 02	Ens	0	3,5	0	0	0	0	3,5		<div>Valider</div>	
		Ext HG	Sem 51	Ens	2,5	0	0	0	0	0	3,8		<div>Valider</div>	
		Ext transversal	Sem 50	Ens	0	2	0	0	0	0	2,0		<div>Valider</div>	
		Réservés / Internes	Rés/Int Transversal	Sem 40	Ens	0	2	0	0	0	0	2,0		<div>Valider</div>
		Rés/Int Lettres Histoire	Sem 06	Ens	0	0	4	0	0	0	3,0		<div>Valider</div>	
	Réservés / Internes / Liste d'	Rés/Int, LA Transversal	Sem 42	Ens	0	2	0	0	0	0	2,0		<div>Valider</div>	
Réservés / Liste Aptitude	Rés/LA Tranversal	Sem 47	Ens	0	2	0	0	0	0	2,0		<div>Valider</div>		
FI enseignant total					4	17,5	6	0	0	0	27,5			
F diplômantes	Master EnFA	M1ENFA UE2.1	Encadrement	Ens	0	10,51	0	0	0	0	10,5		<div>Valider</div>	
				Rest	0	3,5	0	0	0	0	3,5		<div>Valider</div>	
				Recherche	0	192	0	0	0	0	192,0		<div>Valider</div>	
			Recherche	0	46,5	0	0	0	0	46,5		<div>Valider</div>		
			Encadrement	Rest	0	3,5	0	0	0	0	3,5		<div>Valider</div>	
		M1ENFA UE4 M2ENFA UE7EA_S9 M2ENFA UE7IF_S9	Resp UE/ssU	Rest	0	3,75	0	0	0	0	3,8		<div>Valider</div>	
			Ens	0	25	0	0	0	0	25,0		<div>Valider</div>		
			Ens	10	22	0	0	0	0	37,0		<div>Valider</div>		
			Ens	0	7	0	0	0	0	7,0		<div>Valider</div>		
F diplômantes total					10	313,76	0	0	0	0	323,76			
Recherche	Tps Recherche Modulation recherche		Encadrement	Rest	0	3,5	0	0	0	0	3,5		<div>Valider</div>	
				Recherche	0	192	0	0	0	0	192,0		<div>Valider</div>	
				Recherche	0	46,5	0	0	0	0	46,5		<div>Valider</div>	
			Encadrement	Rest	0	3,5	0	0	0	0	3,5		<div>Valider</div>	
			Resp UE/ssU	Rest	0	3,75	0	0	0	0	3,8		<div>Valider</div>	

L'administrateur à accès à la même fiche, mais il ne peut pas modifier les commentaires, seulement les consulter. Une liste affiche uniquement les lignes qui font l'objet d'un litige, ainsi que le nom de l'enseignant concerné.

Liste des litiges

Enseignant	N0 Activités	N1 Sous Activité	N2 Actions	Type activités	CM	TD	Pluri2	Pluri3	Pluri4	Pluri5	Commentaires
CARRE	F Continue	PRF	Accompagnement personnalisé/In	Ens	0	10.5	0	0	0	0	nhgc
CARRE	F Continue	PRF	Livret de compétences et publi	Ens	0	7	0	0	0	0	;kjhkjhg
CARRE	F Continue	PRF	histoire des arts au DNB	Ens	0	8	0	0	0	0	qsdgf
CARRE	FI enseignant	Externes	Ext Lettres	Ens	7	3.5	6.5	0	0	0	kuyg
CARRE	FI enseignant	Externes	Rés/int Lettres	Ens	0	24	0	0	0	0	dfqsqfd

[Déconnexion](#)

3.b) Bilan personnel

Ce stage a été très bénéfique pour moi, puisqu'il s'agissait de ma première vraie expérience professionnelle. J'ai ainsi pu, à travers ce stage, découvrir le monde du travail, appliquer mes connaissances aux services des autres ainsi que les responsabilités qui en découlent.

Le simple fait d'utiliser une méthode de programmation qui m'était alors inconnue, m'a permis de compléter mes connaissances et d'avoir eu l'expérience d'une réelle mise en situation. J'ai donc dû apprendre à me servir d'éléments dont je ne m'étais jamais servi, tout en me basant sur ce que je savais déjà et pouvoir ainsi proposer une application fonctionnelle et viable.