



Rapport de TER

Master 1 Informatique Pour les Sciences
"IPS"

« Amélioration de la plateforme des stages informatiques »

Mai 2017

Encadré par :
Monsieur Eric BOURREAU

Soutenu par :

ALIOUACHE	Mohammed
KERMOUCHE	Yacine
SAKHI	Idris
BENSADOUNE	Mohand Saïd

Université de Montpellier 2016/2017

Table des matières

1.	Introduction-----	3
2.	Contexte d'étude-----	4
3.	Déploiement de la plateforme sur le localhost de nos machines -----	5
4.	Présentation générale de la PGS -----	5
5.	Utilisation de Bugzilla-----	5
6.	Résolution des bugs de la plateforme -----	6
7.1.	Liste de bugs traités -----	6
7.1.1.	Généralisation du switch des rôles -----	7
7.1.2.	Amélioration de l'affichage des détails de l'entreprise-----	7
7.1.3.	Export des stages depuis la plateforme (format csv)-----	8
7.1.4.	Champs du formulaire de création de stage -----	8
7.1.5.	Amélioration du pop-up du calendrier des date-----	9
7.1.6.	Option de multi-sélections sur les stages -----	9
7.1.7.	Ajout de colonnes étudiant et tuteur au listing de stages -----	10
7.1.8.	Ajout d'un bouton de modification pour le display-stage-----	10
7.1.9.	Sauvegarde des motifs de modification des soutenances-----	10
7.1.10.	Limite temporelle d'affichage des soutenances de l'année n-1 -----	12
7.1.11.	Bonne mise à jour dans la bdd quand un tuteur se propose dans le planning des soutenances -----	12
7.1.12.	Adaptation du site aux étudiants M1 et L3 -----	13
7.1.13.	Affichage sélectif des stages (selon le parcours de l'étudiant authentifier) -----	14
7.1.14.	Recherche avec mots clés -----	15
7.1.15.	Initiation d'une nouvelle section « Embauche »-----	15
7.1.16.	Sauvegarde des pièces jointes -----	19
7.1.17.	Dématérialisation de la fiche de suivi étudiant-----	22
7.	Conclusion-----	25
8.	Référence -----	26

1. Introduction

La programmation orientée objet POO : Cette notion a vu le jour dans les années 1960, elle a été créée par des norvégiens, ensuite reprise par d'autres (Brad et al. 1989).

Elle consiste à définir des briques logicielles qui seront des objets. Ces objets font référence à des concepts, idées ou entités qu'on voit de tous les jours (https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet). Une fois les briques définies, les interactions peuvent s'ajouter.

POO et PHP : comme tout autre langage de POO, PHP nous permet de faire idem en plus de sa qualité coté WEB. Voici à quoi ressemblerai la structure d'une classe dans PHP

```
<?php
class nomClasse
{
    var $attribut1;
    var $attribut2;
    var $attribut3;
    var $attribut4;

    function nomFonction1() {

    }

    function nomFonction2() {

    }

    function nomFonction3() {

    }
}
?>
```

(<https://openclassrooms.com/courses/programmez-en-orientee-objet-en-php/introduction-a-la-poo>)

Maintenant le fait d'utiliser de la programmation orientée objets en PHP, nous donne un grand avantage de simplification de l'interaction avec la base de données.

Si on dit que chaque classe représentera une table dans la base de données, les attributs de la classe vont donc représenter les colonnes des tables. Donc les fonctions nous permettront de mettre à jour la base de données avec toutes les modifications possibles, (INSERT INTO, SELECT ... FROM, UPDATE ... SET, DELETE FROM). Aussi les fonctions qui sont définies une fois pourront être appelées autant de fois désirées.

2. Contexte d'étude

La plateforme des stages de la faculté des sciences, étant conçue par des étudiants en informatique, continue encore cette année de s'améliorer. Vue l'ajout de nouvelles fonctionnalités chaque année, de nouveaux bugs dans le site font leurs apparitions. La gestion électronique des documents devient de plus en plus inévitable avec l'augmentation du volume des documents à stocker. La GED permet principalement de gérer les documents chargeable et téléchargeable dans le site, ici les meilleurs avantages d'une mise en place d'une GED : Organisation, efficacité, sécurité, facilité d'accès... L'accès à tous les documents est assuré par le propriétaire du document, par défaut, aucun autre utilisateur ne peut y accéder à un document d'un autre sauf si celui-ci l'autorise. Les opérations possibles sur le document sont celles spécifiées par le propriétaire ([M Hajmiragha 2001](#)). On voit bien qu'on retrouve ce que proposent les box de stockage sur internet. Et depuis, la gestion des documents a pris une grande ampleur et n'a cessé de s'améliorer.

Certains proposent un accès à un stockage distant par un hyperviseur via des connections TCP/IP ([Patten et al. 2015](#)). D'autres se sont penchés plus sur l'apport d'autres notions à la GED tel que le stream ([Motwani et al. 2005](#)). Dans le cas de la plateforme des stages, le souci et le fait d'avoir beaucoup de documents à stocker côté serveur, et la capacité de stockage dédiée sera bientôt remplie. Le cas de sécurité des fichiers n'a pas vraiment d'importance vu que les documents sont stockés au serveur et que celui-ci est déjà bien sécurisé. L'une des possibilités est d'utiliser un box déjà existant et de générer un lien pour chaque fichier vers le box afin de le télécharger, ces liens seront stockés dans la table des pièces jointes.

Aussi, la dématérialisation des fiches qui sont en pièces jointes contribue à améliorer la gestion des données, ainsi quelques fichiers pourront être sauvegardés sous forme de formulaires dans la base de données. Et à partir de là qu'il a fallu investiguer sur la possibilité de génération des documents à partir des formulaires pour reconstituer les fichiers dématérialisés à tous moments. Le plus important dans le site est d'assurer le CRUD.

La librairie FPDF est une classe PHP qui permet la génération des fichiers PDF. La classe est gratuite d'utilisation et on peut la personnaliser comme on le souhaite, d'ailleurs, le premier F veut dire Free. La classe dispose de plusieurs options plutôt avancées et suffisantes pour tirer profit dans le cas de génération des fiches facilement imprimables. Dans les différentes options, on peut trouver la possibilité de mettre des images JPEG et PNG, mise en page de base, Couleurs... (www.fpdf.org). Il faudra mettre en place les améliorations et les solutions de bugs en respectant les différents rôles des utilisateurs identifiés.

La plateforme des stages est programmée avec du **Php** et **Mysql**, sachant qu'en combinant ces deux-là, un site web peut être bien entretenu. Du Php orienté objet est utilisé pour faciliter le passage à Mysql et la sauvegarde des données. ([L.Welling & L.Thomson 2009](#)).

3. Déploiement de la plateforme sur le localhost de nos machines

La plateforme a été déployée avec succès sur les localhosts des étudiants, enfin presque. Le seul indésirable est le fait que les scripts, dans la partie consultation de la base de donnée ne marche pas à merveille. Les fonctions prédéfinies de MySQL ne marchent pas directement et nous impose d'utiliser par exemple : `mysqli_connect()` à la place de `mysql_connect()` (3 machines sur 4 ont le même souci).

Aussi, on a rencontré le problème du fait d'utiliser un `SELECT DISTINCT` avec `ORDER BY` (voir <https://github.com/doctrine/doctrine2/issues/5622> comme exemple). Néanmoins, il existe des configurations mysql qui évite ce genre de problèmes. On peut mentionner aussi l'erreur qui survient quand un `INSERT INTO` avec des valeurs par default des attributs.

4. Présentation générale de la PGS

Le site est constitué de pages php situées directement dans le dossier mère, pour naviguer dans les différentes pages web de la plateforme dont on trouvera le fichier `index.php`, qui est la page d'accueil. Il a aussi des scripts contrôleurs des pages et des formulaires remplis par les utilisateurs. Ces contrôleurs se trouvent dans le sous dossier `./controleurs/`. Le sous dossier `./includes/` contient des scripts de configuration et les scripts de classes qui représentent la partie orientée objets du Php.

Ces classes avec les méthodes définies à l'intérieur permettent le contact avec la base de donnée, et assure la lecture et écriture tout en assurant la sécurité. La base de données est constituée de plusieurs tables nécessaires pour le site. On n'oublie pas aussi les scripts JavaScript et les feuilles de style associés.

Très important aussi, le fichier `conf.php`, destinée aux nouvelles tables créées dans la base de données sont déclarer dans ce fichier afin que toutes les requêtes anciennes ou nouvelles aient la même forme.

5. Utilisation de Bugzilla

Suivant le site de Bugzilla, voici en gros les différents points à quoi Bugzilla peut servir :

- suivre les bogues et modifications du code source d'un logiciel ;
- communiquer avec les membres de l'équipe ;
- soumettre et valider des correctifs ;
- gérer l'assurance qualité.

Bugzilla permet d'organiser vos processus de développement de logiciels. Les projets réussis sont souvent le résultat d'une organisation et d'une communication efficaces. Bugzilla est un outil puissant qui aidera votre équipe à s'organiser et à communiquer efficacement.

Le package de Bugzilla est constitué essentiellement des fichiers cgi et des fichiers pm, les fichiers pm sont des programmes perl. Les soucis avec ce package, est le fait que son installation est vraiment pénible. En plus les erreurs rencontrées diffèrent d'un ordi à un autre.

Jusqu'à présent, on a pu mettre en place sa base de données, la configuration du ScriptAlias de son dossier d'exécution dans le webserver. Mais actuellement au lancement de `l'index.cgi` , il y a plein d'erreurs internes.

Ticket d'installation de Bugzilla sur les machines de l'université :

Concernant le ticket ouvert par Monsieur Bourreau pour que la DSI puisse installer Bugzilla sur nos machine à l'université, nous avons essayé de le faire et ainsi eux la confirmation que nous, seuls étions pas capable d'exécuter cette taches.

Nous ne sommes simplement pas sudoers sur ses comptes de l'université, ainsi à l'étape du manuel d'installation de Bugzilla **2.2.4.1.1.2. Apache httpd avec mod_cgi**, en finissons cette étape montrer ci bas, il est simplement impossible de sauvegarder la modification sur ce fichier.

2.2.4.1.1. Apache httpd with mod_cgi

To configure your Apache web server to work with Bugzilla while using mod_cgi, do the following:

1. Load `httpd.conf` in your editor. In Fedora and Red Hat Linux, this file is found in `/etc/httpd/conf`.
2. Apache uses `<Directory>` directives to permit fine-grained permission setting. Add the following lines to a directive that applies to the location of your Bugzilla installation. (If such a section does not exist, you'll want to add one.) In this example, Bugzilla has been installed at `/var/www/html/bugzilla`.

```
<Directory /var/www/html/bugzilla>
AddHandler cgi-script .cgi
Options +Indexes +ExecCGI
DirectoryIndex index.cgi
AllowOverride Limit
</Directory>
```

These instructions: allow apache to run .cgi files found within the bugzilla directory; instructs the server to look for a file called `index.cgi` if someone only types the directory name into the browser; and allows Bugzilla's `.htaccess` files to override global permissions.

6. Résolution des bugs de la plateforme

Suite au partage de rôles ainsi que des différents bugs à essayer de résoudre à l'aide notamment de Bugzilla, pour une meilleure organisation et suivi du travail, voici les différents bugs résolus et améliorations apportées au gestionnaire des stages.

7.1. Liste de bugs traités

- Généralisation du switch des rôles.
- Amélioration de l'affichage des détails de l'entreprise
- Export des stages depuis la plateforme (format csv)
- Champs du formulaire de création de stage
- Amélioration du pop-up du calendrier de dates
- Option de multi-sélections sur les stages
- Ajout de colonnes étudiant et tuteur au listing de stages
- Ajout d'un bouton de modifications pour le display-stage
- Sauvegarde des motifs de modification des soutenances
- Limite temporelle d'affichage des soutenances de l'année n-1
- Adaptation du site aux étudiants M1 et L3
- Affichage sélectif des stages
- Recherche avec mots clés

- Initiation d'une nouvelle section « Embauche »
- Sauvegarde de pièces jointes
- Bonnes mises à jour dans la base de données lors de modifications dans le planning des soutenances

7.1.1. Généralisation du switch des rôles

Au départ, seulement les administrateurs pouvaient faire ça mais maintenant, il sera généralisé pour que les responsables et le secrétariat puissent le faire aussi. Ainsi ils pourront mieux faire face aux difficultés rencontrées par les étudiants. Il existait déjà un formulaire html qui permet le changement de rôles, donc ce qui restait à faire est de changer les conditions d'accès à ce formulaire.

Voici le changement apporté sur le fichier **/includes/inc.includes.php** :

```

61      if (($currentUser->role == SECURITY_ADM || $currentUser->role == SECURITY_SECR) || $currentUser->role ==
62          SECURITY_RESP){
63          if (isset($_SESSION['overrideRole'])){
64              $currentUser->OverrideRole($_SESSION['overrideRole']);
65          }

```

Il a fallu aussi changer le fichier **/includes/class.template.php** pour bien assurer le fonctionnement de cette option :

```

221      if ($currentUser->isOverrided() || ($currentUser->role == SECURITY_ADM || $currentUser->role ==
222          SECURITY_SECR) || $currentUser->role == SECURITY_RESP){
223          $col = 3;
224          $identif.="<b>&nbsp;(".$currentUser->role."</b>&nbsp;</td><td align=\"left\"><form method=\"POST
225              \" action=\"override-role.php\" style=\"float:right;margin-right:5px;\"><select name=\"
226              newRole\">";
227          if ($currentUser->isOverrided()){
228              $identif.='<option selected="selected" value="STOP">Retour</option>';
229          }
230          $identif.='<option value="ETUD">ETUD</option><option value="RESP">RESP</option>
231              <option value="ENT">ENT</option><option value="SECR">SECR</option><option value="TUT">TUT</
232              option><option value="INV">INV</option></select>&nbsp;<input type="submit" value="Go"
233              name="send" /></form>';
234      }

```

7.1.2. Amélioration de l'affichage des détails de l'entreprise

Donc au lieu d'afficher les valeurs par default enregistrées dans la base de données, ce sera mieux de ne rien afficher pour ne pas fausser les informations. Par exemple, si l'information du local de l'entreprise n'est pas saisie par l'utilisateur, il faudrait ne pas afficher le non qui est la valeur par default de la table. Deux fichiers ont été modifiés.

Voici la modification apportée sur le fichier **display-entreprise.php** :

```

59      if ($entreprise->nbSalaries==0)
60          $content.="<tr><td><b>Nombre de salari&eacute;s : </b></td><td> --- </a></td></tr>\n";
61      else
62          $content.="<tr><td><b>Nombre de salari&eacute;s : </b></td><td> ".$entreprise->nbSalaries."</a></td></tr>\n";
63      if($entreprise->local!="") {
64          $content.="<tr><td><b>Possède un local?</b></td><td>";
65          if ($entreprise->local == 1) {
66              $content .=" " Oui</td></tr>\n";
67          }
68          else if ($entreprise->local == 0) {
69              $content .=" " --- </td></tr>\n";
70          }
71      }

```


La même modification a été apportée sur le fichier **display-stage.php** dans la section de l’affichage des informations concernant l’entreprise.

7.1.3. Export des stages depuis la plateforme (format csv)

Un fichier a déjà été codé par les étudiants de l’année passée mais, la requête SQL consultant la base de données a dû être changée pour son bon fonctionnement. Dans la requête, il a fallu faire plein de jointure pour récupérer différentes informations dans différentes tables.

Voici le changement apporté sur le fichier **exp-stg.php** :

```

16 if ($_POST['annee']!='toutes'){
17     $annees=explode("/",$_POST['annee']);
18     $req=$PDO->prepare("SELECT DISTINCT c.NOM,c.PRENOM,d.numeroEtudiant,b.VALEURNOTE,e.niveau,e.acronyme as
    FORMATION,a.TITRE FROM pgs_stage a,pgs_stage_note b,pgs_utilisateur c,pgs_etudiant d,pgs_parcours e WHERE
    a.IDETUDIANT=c.IDUSER AND a.IDSTAGE=b.IDSTAGE AND e.idParcours=d.idParcours AND c.IDUSER=d.idEtudiant
    AND a.DATEDEBUT<'".$annees[1]."-06-30" AND a.DATEDEBUT>'".$annees[0]."-09-01" UNION SELECT DISTINCT
    c.NOM,c.PRENOM,d.numeroEtudiant,Null as VALEURNOTE,e.niveau,e.acronyme as FORMATION,a.TITRE FROM
    pgs_stage a,pgs_utilisateur c,pgs_etudiant d,pgs_parcours e WHERE a.IDETUDIANT=c.IDUSER AND
    e.idParcours=d.idParcours AND c.IDUSER=d.idEtudiant AND a.DATEDEBUT<'".$annees[1]."-06-30" AND
    a.DATEDEBUT>'".$annees[0]."-09-01" ORDER BY FORMATION ");
19 }
20 else{
21     $req=$PDO->prepare("SELECT DISTINCT c.NOM,c.PRENOM,d.numeroEtudiant,b.VALEURNOTE,e.niveau,e.acronyme as
    FORMATION,a.TITRE FROM pgs_stage a,pgs_stage_note b,pgs_utilisateur c,pgs_etudiant d,pgs_parcours e WHERE
    a.IDETUDIANT=c.IDUSER AND a.IDSTAGE=b.IDSTAGE AND e.idParcours=d.idParcours AND c.IDUSER=d.idEtudiant
    UNION SELECT DISTINCT c.NOM,c.PRENOM,d.numeroEtudiant,Null as VALEURNOTE,e.niveau,e.acronyme as
    FORMATION,a.TITRE FROM pgs_stage a,pgs_utilisateur c,pgs_etudiant d,pgs_parcours e WHERE a.IDETUDIANT=c.
    IDUSER AND e.idParcours=d.idParcours AND c.IDUSER=d.idEtudiant ORDER BY FORMATION");
22 }
23
24 $req->execute();
25 $data=$req->fetchAll();
26
27 CSV::export($data,'export stage');
```

7.1.4. Champs du formulaire de création de stage

Certains champs, après analyse des besoins et des prérequis pour une saisie optimale d’un nouveau stage, ont été mis en obligatoire pour éviter d’avoir plein de vides dans la base de données. Par exemple, le titre de stage, le niveau requis...

Donc le fichier **create-stage.php** a été un peu modifié :

```

325 <tr><td>Titre : </td><td>'.$HTML->getInputTextRequired("titreStage", "", "inputtext", 65).'</td></tr>
326 <tr><td>Niveau requis (M1, M2...): </td><td>'.$HTML->getInputTextRequired("niveauRequis", "", "
    inputtext", 65).'</td></tr>
```

Mais le fait de faire ça a causé quelques soucis avec le bouton annuler la saisie d’un nouveau stage qui est lui aussi un bouton de type submit, donc pour annuler, il faut d’abord remplir et ça ce n’est pas pratique. Le bouton Annuler n’est plus un submit grâce à une nouvelle fonction dans la classe HTML qui est dans le fichier **/includes/class.HTML.php** :

```

82 function getInputSubmitCancel($name, $value, $enabled=true, $cssClass="", $autre=""){
83     //retourne un input de type submit
84     $result = '<input type="button" onclick="window.location.href=\'cancelStageCreation.php\';" id="'.$name.'"
    name="'.$name.'" class="'.$cssClass.'" value="'.$value.'" enabled="'.$enabled.'"';
85     $result.= '</>';
86
87     return $result;
88 }
```

Et cette fonction est appelée depuis le fichier **create-stage.php** :


```

338      <tr><td width="90%">'. $HTML->getInputSubmit("send", "    Terminer    ").'</td><td>'. $HTML->
      getInputSubmitCancel("reset", "Annuler la creation").'</td></tr>

```

Puis la fonction fait appel à un nouveau fichier **CancelStageCreation.php** qui vide les données enregistrées dans la variable \$_SESSION pendant la création s'il y'en a et il y aura redirection de la page vers l'accueil :

```

1  <?php
2  include_once("includes/inc.includes.php");
3
4  if (isset($_SESSION['etape_courante']))
5      unset($_SESSION['etape_courante']);
6  if (isset($_SESSION['position']))
7      unset($_SESSION['position']);
8  if (isset($_SESSION['createStage']))
9      unset($_SESSION['createStage']);
10 $_SESSION['msg'] = "Création du stage annulée";
11 redirect("list-stages.php");
12 ?>

```

7.1.5. Amélioration du pop-up du calendrier des date

Lors de l'apparition de ce pop-up la page a tendance à remonter vers le haut et le calendrier s'affiche vers la partie haute et gauche de l'écran. D'abord pour parvenir au dérangement de la page qui remonte vers le haut, il faut seulement mettre dans le href du lien du calendrier "javascript: void(0); " :

```

330      <tr><td>Date de début : </td><td>'. $HTML->getInputTextRequired("dateDebutStage", "----/--/--",
      "inputtextdisabled", 15, false, true).&nbsp; format (aaaa-mm-jj) <a href="javascript:void(0)"
      onClick="'. $scal->get_strPopup("calendrier/pop.calendrier.php", 350, 170).'"></a></td></tr>

```

Ensuite, pour placer le pop-up au bon endroit, l'idéal est de récupérer les coordonnées de la souris et de le placer juste à côté. Dans le fichier **/calendrier/class.calendrier.php**, on apporte le changement suivant :

```

356  function get_strPopup($page, $l, $h){
357      $stab = explode('?', $page);
358      if (count($stab) > 1){
359          $cc = '&';
360      }else{
361          $cc = '?';
362      }
363      $page .= $cc.'frm='.$this->form.'&ch='.$this->champ;
364      $str = "var x=event.clientX-175;var y=event.clientY+105;window.open('". $page."
      ', 'calendrier', 'top='+y+', left='+x+', width='".$l."', height='".$h."', scrollbars=0').focus();"
365      return $str;
366  }

```

7.1.6. Option de multi-sélections sur les stages

Dans la page qui affiche la liste des stages, cette option sera accessible que par les responsables, les administrateurs et le secrétariat. Donc on ajoutera des checkbox devant tous les stages, une barre d'outils qui pourra appliquer des options communes comme supprimer et valider.

Voici les modifications apportées sur le code. Sur le fichier **list-stage.php** :

Pour ajouter les check box dans un formulaire,

```

368      if ($role == SECURITY_ADM or $role == SECURITY_RESP or $role == SECURITY_SECR){
369          $content.="<td>'. $HTML->getInputCheckBoxJS('multichoiceselect'. $i, $strval($stage->id), "show_toolbar();")."
          </td>";
370      }

```

Et pour ajouter la barre d'outils,

```

214     if ($role == SECURITY_ADM or $role == SECURITY_RESP or $role == SECURITY_SECR){
215         $content.='<form methd="POST" action="multi_select_options.php">';
216         $content .= '&nbsp;<center><div id="toolbar_multiselection" style="display:none"><input type="submit"
                        id="multiple_edit" name="validate" title="Valider" value=" " style="background-image:url(\'\'.'$DirImages.'
                        check.png\'); background-size: 20px 20px;background-repeat: no-repeat;">';
217         $content .= '&nbsp;<input type="submit" id="multiple_erase" name="erase" title="Supprimer" value=" "
                        style="background-image:url(\'\'.'$DirImages.'eraser.png\');background-size: 20px 20px;background-repeat:
                        no-repeat;">';
218         $content .= '&nbsp;<input type="submit" id="multiple_recyclage" name="recyclage" title="" value=" "
                        style="background-image:url(\'\'.'$DirImages.'recyclage.png\');background-size: 20px 20px;background-repeat:
                        no-repeat;"></div></center></br>';
219     }

```

Cette barre d'outil est en « **display:none** » en démarrage, pour l'afficher, il faut sélectionner au moins un stage, l'affichage est traité par une fonction JavaScript :

```

25     function show_toolbar(){
26         toolbar_multiselection.style.display="block";
27     }

```

Une fois les stages sélectionnés et l'option à appliquer est appelé, un nouveau fichier est créé pour le traitement (Consulter le fichier : **multi_select_options.php**)

7.1.7. Ajout de colonnes étudiant et tuteur au listing de stages

Dans le cas où l'utilisateur connecté est un tuteur. Voici les ajouts de codes qui permettent de faire ça : les ajouts sont dans le fichier **list-stage.php**.

```

304     if (($role == SECURITY_TUT)){
305         $content .= "<th width=\"70px\">$enteteNomEtudiant</td>
306         <th width=\"70px\">$enteteNomTuteur</td>
307         ";
308     }

```

```

392     if (($role == SECURITY_TUT)){
393         $content .= "<td><a href=\"display-user.php?iduser=" . $etudiant->id . "\">
394         $etudiant->prenom."&nbsp;". $etudiant->nom. "</a></td>";
395         $content .= "<td><a href=\"display-user.php?iduser=" . $tuteur->id . "\">
396         $tuteur->prenom."&nbsp;". $tuteur->nom. "</a></td>";
397     }

```

7.1.8. Ajout d'un bouton de modification pour le display-stage

Un bouton Modifier a été ajouté dans la page **display-stage.php** pour passer directement à la page **edit-stage.php** sans devoir revenir en arrière et cliquer sur le crayon de modification. Bien sûr, ce bouton est accessibles seulement à ceux qui ont le crayon de modification (ADM, RESP, SECR).

```

110     if (($role == SECURITY_RESP) or ($role == SECURITY_ADM) or ($role == SECURITY_SECR)){
111         $content.='<td><input type="button" value="Modifier" onClick="javascript:window.location=\'
                        edit-stage.php?act=edit&idstage='.$stage->id.'\'></td>';
112     }

```

7.1.9. Sauvegarde des motifs de modification des soutenances

Pour ça, une class **includes/class.soutenance.modif.php** a été ajouté et une table pgs_soutenance_modif a été créée dans la base de données. Un champ textArea a été ajouté dans les formulaires de modification pour récupérer le motif. Voici par exemple dans le fichier **edit-soutenance.php** il a été ajouté aussi dans le fichier **calendrier.php** et **calendrierHorizontal.php**.

```

189 <tr>
190 <td>Motif de modification</td>
191 <td>'<!--getInputTextAreaRequired("motifModification").'</td>
192 </tr>

```

Grace à la nouvelle classe `SoutenanceModification`, on pourra sauvegarder les motifs dans la table.

```

47 function save() {
48     global $BD, $bdSoutenanceModification;
49     if (!$this->isLoaded()) {
50         // on fait un Insert
51         $sql = "INSERT INTO $bdSoutenanceModification
52             (idSoutenance,motif,idUser)
53             VALUES
54             (" . $BD->escapeString($this -> idSoutenance) . ",
55             " . $BD->escapeString($this -> motif) . ",
56             " . $BD->escapeString($this -> idUser) . ")";
57         $BD -> query($sql);
58     }
59     return false;
60 }

```

La méthode `save()` est appelée juste après avoir réussi à modifier une soutenance :

```

54 $checkmodif=$soutenance->save();
55 if ($checkmodif){
56     include_once 'includes/class.soutenance.modif.php';
57     $soutenanceModification =new SoutenanceModification();
58     $motifModification=$ POST['motifModification'];
59     $soutenanceModification->init($soutenance->idSoutenance,$motifModification,$currentUser->id);
60     $soutenanceModification->save();
61 }
62 $_SESSION['msg']="Modification du cr&eacute;neau enregistr&eacute;e";

```

Ainsi un historique de modification peut être affiché dans le fichier **edit-soutenance.php**. Par défaut seulement les trois derniers changements seront affichés, mais un lien permettra d'afficher plus.

Pour récupérer les motifs de modification de la base de données, la nouvelle classe suivante et sa méthode sont utilisées :

```

63 class ListeModifSoutenances{
64     var $list;
65     function __construct(){
66         $this->liste = array();
67     }
68     function getModifications($idSoutenance){
69         global $BD, $bdSoutenanceModification;
70         $sql = "SELECT *
71             FROM $bdSoutenanceModification
72             WHERE idSoutenance = ".$idSoutenance." ORDER BY idModification DESC";
73         $res = $BD->query($sql);
74         if ( $BD->numRows == 0 )
75             $this->liste=array();
76         else
77         { //il y a des resultats
78             while ( $BD->next($res) )
79             {
80                 $row = $BD->result;
81                 $soutenanceModification= new SoutenanceModification($row->idModification);
82                 array_push($this->liste, $soutenanceModification);
83             }
84         }
85     }
86 }

```

La méthode `getModifications(id)` est appelée depuis le fichier **edit-soutenance.php** ligne:96, puis affichée dans un `<fieldset>`, et en utilisant une fonction JavaScript qui permet d'afficher plus que trois modifications(**Plus_Modif(a)**).

Puis dans le calendrier de soutenance un formulaire de confirmation et de saisie de motif de modification est aussi mis en place, dans le fichier **calendrier.php**, ligne:224++, afficher avec une fonction JavaScript aussi : **Modify_Soutenance(e)**.

7.1.10. Limite temporelle d’affichage des soutenances de l’année n-1

Nous ne voulons pas afficher les soutenances de l’année passée pendant l’année courante, mais aussi les modifications seront fermées une semaine avant le début des soutenances pour les tuteurs et les étudiants. Donc la date de connexion est récupérée et testée par rapport aux dates des soutenances saisies dans la configuration.

Voici les modifications apportées sur le fichier **calendrier.php**. Notez bien que les mêmes modifications sont aussi appliquées dans le calendrier version horizontale.

```
55 $content2='';
56 $dateAct=date('d-m-Y');
57 $dateAct=new DateTime($dateAct);
58 $strdat1='30-09-' . ($sdd->format('Y'));
59 $strdat2='01-04-' . ($sdd->format('Y'));
60 $datecomp1=new DateTime($strdat1);
61 $datecomp2=new DateTime($strdat2);
62 $dateFermeturemodifSoutenance=$sdd->format('d-m-Y');
63 $dateFermeturemodifSoutenance=strtotime($dateFermeturemodifSoutenance);
64 $dateFermeturemodifSoutenance=$dateFermeturemodifSoutenance-7*3600*24;
65 $dateFermeturemodifSoutenance=date('d-m-Y', $dateFermeturemodifSoutenance);
66 $dateFermeturemodifSoutenance=new DateTime($dateFermeturemodifSoutenance);
```

Pour la condition que les soutenances sont bien de l’année courante :

```
68 if ($dateAct<$datecomp1 && $dateAct>$datecomp2){
```

Sinon le message suivant sera affiché :

```
346 else{
347     $content.='<center><h3>Les soutenances ne sont pas encore disponible pour l\'ann&eacute;e en cours</h3></
348     center>';
}
```

Aussi la fermeture des modifications de soutenances est effectuée une semaine avant le début :

```
72 if ($dateFermeturemodifSoutenance>$dateAct){
73     $content.='<p style="color:green ">La modification sera ferm&eacute;e le:'. $
        dateFermeturemodifSoutenance->format('d/m/Y'). '</p>';
74 }
75 else{
76     $content.='<p style="color:red ">La modification est ferm&eacute;e le:'. $dateFermeturemodifSoutenance
        ->format('d/m/Y'). '</p>';
77 }
```

La modification reste toujours valable pour les responsables, les administrateurs et le secrétariat :

```
223 if ($dateAct<$dateFermeturemodifSoutenance || $role==SECURITY_ADM||$role==SECURITY_SECR||$role==
    SECURITY_RESP){
```

7.1.11. Bonne mise à jour dans la bdd quand un tuteur se propose dans le planning des soutenances

Au début, la plateforme ne fait pas une bonne liaison entre les modifications apportées sur le planning des soutenances et la partie d’édition des stages. Ceci est dû au fait que le code traitant ces modifications ne fait pas les bonnes mises à jour dans la base de données. Et donc quand un responsable part dans l’édition d’un stage afin de confirmer l’ajout du rapporteur, il ne retrouve même pas ce dernier qui s’est proposé et le responsable devra le remettre encore lui-même.

Le code traitant cette partie a été analysée et il a été trouvé que la table des stages n’est pas mise à jour lors du traitement donc quelques lignes de code ont été ajoutées pour parvenir à l’améliorer :

Dans le fichier **AffectationSoutenance.php**, il a fallu traiter le cas où un rapporteur s’engage et bien sur le cas contraire aussi:

Ce bout de code monte l'état du stage de VALIDETUTEUR à CHOISIRAP, et si un responsable le confirme ça passera à VALIDERAPPORTEUR

```
124 $stage=new Stage($getSoutenance->idStage);
125 $stage->etat="CHOISIRAP";
126 $stage->idRapporteur=$currentUser->id;
127 $stage->save();
```

Et dans le code suivant, le cas contraire est assuré :

```
105 $stage=new Stage($getSoutenance->idStage);
106 $stage->etat="VALIDETUTEUR";
107 $stage->idRapporteur=-1;
108 $stage->save();
```

7.1.12. Adaptation du site aux étudiants M1 et L3

Une fonction de vérification du fait que l'étudiant soit un étudiant M1(ou L3), ou pas est mise en place dans la classe de l'utilisateur. La fonction renvoie vrai si l'étudiant est un M1 et faux dans le cas contraire. Voici le code de la fonction :

```
function estM1(){
    global $BD, $bdInfo;
    $sql = "SELECT idParcours FROM " . $bdInfo . " WHERE idEtudiant = " . $this->id;
    $res = $BD->query($sql);
    $donne = mysql_fetch_assoc($res);
    $sql2 = "SELECT niveau FROM pgs_parcours WHERE idParcours = " . $donne["idParcours"];
    $res2 = $BD->query($sql2);
    $donne2 = mysql_fetch_assoc($res2);
    if ($donne2["niveau"] == "M1" || $donne2["niveau"] == "L3") {
        return true;
    }
    else {
        return false;
    }
}
```

Ensuite il faudra utiliser cette fonction comme test pour faire en sorte de choisir ce qu'il sera affiché pour un étudiant M1, L3 ou pas.

Une remarque très importante : vu que les étudiants M1 sont enregistrés dans la base de données en tant qu'invités, il faudra aussi s'assurer qu'il y'aura pas les étudiants M1 partout dans la plateforme comme le cas des M2.

Voici quelques exemples d'utilisation de la fonction : un exemple concret est de cacher la section soutenance vu qu'ils n'auront pas de soutenances pour leurs stages.

Pour faire ce travail il faut qu'on test au moment de la création de la page si l'étudiant est un M1 ou L3 la partie soutenance ne sera pas incluse dans le DOM. En fait, le Template de la page sera carrément modifier.

```
////////////////////////////////////
// menu soutenance 2017
if (isset($this->menuElements["soutenance"]) && count($this->menuElements["soutenance"]) > 0 )
{
    if ($currentUser->estEtudiant() && $currentUser->estM1()){
    }else{
        $menuHTML .= '<div class="menu_gauche_groupe" style="display: show" ><div
            class="menu_gauche_titre" >Soutenance</div>'. "\n";
        foreach($this->menuElements["soutenance"] as $menuItem)
        {
            $menuHTML .= '<a class="lien" href="'. $menuItem['Url'] . ">' . $menuItem['Item'] . '</a>'. "\n";
        }
        $menuHTML .= "</div><br/>". "\n";
    }
}
}
```

La Figure suivante montre le code qui fait ce travail et celle qui vient après montre la différence quand un étudiant M1 ou M2 sont connecté.



7.1.13. Affichage sélectif des stages (selon le parcours de l'étudiant authentifié)

Seuls les stages autorisés au parcours de l'étudiant authentifié seront affichés par default dans le listing des stages. Par contre, voir les autres stages est toujours une possibilité. Pour choisir les bons stages il a fallu modifier la requête Mysql qui récupère les stages.

Dans la figure suivante vous trouvez le script qui initialise le parcours avant d'envoyer la requête SQL, il s'agit du fichier **controleur-list-stages.php**.

```
$aucunparcours=false;
$GETparcours="";
if ( isset ( $_GET['parcours'] ) )
{
    $parcours= new Parcours($_GET['parcours']);
    if ( $parcours->isLoading() )
        $GETparcours=$parcours->id;
    if ( $_GET['parcours'] == "aucun" )
    {
        $GETparcours="aucun";
        $aucunparcours=true;
    }
}
elseif ($currentUser->estEtudiant() && !isset ( $_GET['parcours'] ) ){
    $parcours= new Parcours($currentUser->getParcours2());
    //$parcours = "\" . $currentUser->getParcours2() . "\"";
}else{
    //$parcours = "aucun";
    $parcours = new parcours();
    // $aucunparcours=true;
}
}
```

Donc, si l'utilisateur connecté est un étudiant, qui est vérifié avec la fonction **estEtudiant()**, qui renvoie un booléen, la variable **parcours** sera initialisée avec l'id de ce dernier.

La fonction **getParcours2()** renvoie un string, qui correspond à la discipline de l'étudiant connecté. Cette fonction se trouve dans la classe étudiant.

Dans cette amélioration il faut bien faire attention que ça ne se chevauche pas avec les autres conditions parce que dans l’affichage des stages il y a beaucoup de conditions qui changent d’un utilisateur à un autre.

7.1.14. Recherche avec mots clés

Il existe un sérieux bug dans la recherche des stages. Premièrement, la recherche n’est pas vraiment efficace car pour trouver les stages en relation avec un mot clé, il faudra écrire le mot clé complètement sinon il trouvera rien, donc il y’a une amélioration sur ce point en soignant plus la requête de recherche. Ensuite, quand des résultats sont renvoyés, on ne peut pas suivre le lien du stage à cause d’une balise <a> manquante qui va renvoyer au bon stage. Cette partie est aussi résolue.

La figure suivante concerne la modification de la requête de la recherche, il suffit qu’une partie d’un mot qui se trouve dans le titre ou bien dans la description du stage matche les mots clé entrés, le stage concerné sera récupéré.

```
// condition sur les mots clés
if ( is_array ($motsCles) )
{
    for ($i=0; $i < count($motsCles); $i++)
    {
        $word = $motsCles[$i];
        $word = $BD->escapeString($word);
        $conditions[]="(s.TITRE LIKE '%" . $word . "%'
                        OR s.DESCRPTION (LIKE '%" . $word . "%')";
    }
}
```

Et la figure en dessous montre les nouvelles balises qui seront imprimé dans les deux cas.

```
// on surligne les mots cles trouves
if ( is_array($motsCles) )
{
    $titrestage=$stage->titre;
    for ($j=0; $j < count($motsCles); $j++)
    {
        $word=$motsCles[$j];
        $titrestage = "<a href='display-stage.php?idstage=" . $stage->id . "'> " . @
        preg_replace("/(\b".$word."(s|e|x|es)?\b)/i","<span class=\"trouve\">$1</span>",$
        titrestage). "</a>";
    }
}
else{
    $titrestage= "<a href='display-stage.php?idstage=" . $stage->id . "'> " . $stage->titre . "</a>";
}
```

7.1.15. Initiation d’une nouvelle section « Embauche »

Elle nous permettra de suivre les étudiants après leurs formations (suivre leurs embauches et leurs futurs). D’abord avec cette nouvelle section, l’université pourra garder contact avec les étudiants pour des futures collaborations. Ensuite, les étudiants pourront avoir une idée sur le taux de recrutement dans leurs parcours et aussi les boites qui recrutent le plus afin de bien choisir leurs stages et de bien se lancer dans leurs vies professionnelles.

Dans cette partie, il a fallu créer de nouvelles classes et de nouvelles tables qui nous permettent de sauvegarder les informations nécessaires. Les embauches sont saisissables soit par le tuteur universitaire de l’étudiant soit par l’étudiant lui-même. Tous les autres étudiants pourront voir les statistiques issues de ce sondage mais ils ne pourront pas voir tous les détails.

- Structure générale de la table :

Nous étions obligés de créer une nouvelle table qui contiendra les informations nécessaire d'une embauche. Ces informations sont montrées dans les figures suivantes. On pourra accéder à toutes les autres informations à partir de ce tableau, à l'aide de la classe embauche et les requêtes qu'elle contient.

IDSTAGE	IDetudiant ID de l'étudiant	IDENTREPRISE	COMMENTAIRE	conratType	DATEMBAUCHE
2308	2318	76	fgqddfh	CDD	2017-04-18
2303	2473	767	je suiiiiis embauché	CDI	0000-00-00
2291	2371	637	tres bien	CDD	2017-04-20
2222	2097	76	xvxbx	CDI	2017-04-22

Chaque embauche est associé à un stage qui est définie avec le **IDSTAGE**, un étudiant **IDETUDIANT**, une entreprise d'embauche **IDENTREPRISE**, un commentaire mis par l'étudiant ou par son tuteur, ce commentaire concerne l'embauche, un type de contrat pour faire les statistiques et la date d'embauche.

- Le HTML de la partie embauche et les contrôleurs :

Il peut y avoir plusieurs scenarios pour cette partie, le premier est la saisie d'une embauche. Pour accéder à cette fonctionnalité il faut que l'utilisateur connecté soit l'étudiant embauché ou son tuteur. Quand cet utilisateur rentre dans les détails de stage, une partie embauche est affiché pour saisir les informations nécessaires.

Comme vous pouvez voir sur le script suivant, avant de mettre le formulaire à remplir en teste si rien n'était déjà saisi, si oui on affiche les informations mise dans la base de données. Sinon, on affiche le formulaire à remplir (celui dans la page précédente).

```

496
497 if($etatStage=="VALIDETUTEUR") {
498     if (((($idUser == $idUser) || ($idUser == $stage->idEtudiant)) || ($role == SECURITY_ADM) || ($role ==
499         SECURITY_SECR || $role == SECURITY_SECR)) {
500         $embauche .= "<br><br><legend class='legendh2'>Embauche </legend><br><br>";
501         $content .= str_to_noaccent($embauche);
502         include_once("includes/inc.includes.php");
503         // $stage = new stage();
504         $valeurEmbauche = $stage->getEmbauche();
505         // echo $valeurEmbauche->idStage;
506         if ($valeurEmbauche->idStage != -1) { // l'étudiant a été embauché
507             $class = ($valeurEmbauche->idStage == 1) ? "class='fond2'" : "class='fond1'"; // couleur du fond
508             if ($idUser == $stage->idEtudiant) {
509                 $content .= "<br> des information concernant votre embauche étaient saisie <br><br>";
510             } else {
511                 $content .= "<br> l'étudiant a été embauché dans la boîte ". $valeurEmbauche->nom. " le ". frenchDate($valeurEmbauche->
512                     dateEmbauche). " en contrat ". $valeurEmbauche->contratType. "<br><br>";
513             }
514         } else {
515             $content .= "<br>Aucune information concernant l'embauche n'était saisie pour ce stage<br><br>";
516         }
517         include("calendrier/class.calendrier.php");
518         if (((($idUser == $idUser) || ($idUser == $stage->idEtudiant)) || ($role == SECURITY_ADM) || ($role ==
519             SECURITY_SECR)) && $valeurEmbauche->idStage != -1) {
520             $content .= "<br><h5>saisissez la date d'embauche si l'étudiant est pris après le stage </h5>";
521             $calDDS = new Calendrier("embauche", "dateEmbauche");
522             $ajouteEmbauche = "";
523             $ajouteEmbauche .= "<br><form name='embauche' method='POST' action='add-job.php?idstage=' . $stage->id . '>";
524             $ajouteEmbauche .= "<td width='300px'>Date d'embauche </td>";
525             $ajouteEmbauche .= "<td>.<input type='text' value='YYYY-MM-DD' />";
526             $ajouteEmbauche .= "<br>";
527             $ajouteEmbauche .= "<img src='calendrier/calendrier_icon.jpg' /></td>";
528             $ajouteEmbauche .= "<td colspan='2'><input type='text' value='Commentaire sur l'embauche' /></td>";
529             $ajouteEmbauche .= "<br>";
530             $ajouteEmbauche .= "<h5>Veuillez choisir le type de contrat </h5><br>";
531             $ajouteEmbauche .= "<input type='radio' name='contratType' value='CDI' /> CDI <input type='radio' name='contratType' value='CDD' /> CDD <input type='radio' name='contratType' value='CDD' /> CDD";
532             $ajouteEmbauche .= "<input type='submit' value='Valider' /></form>";
533             $content .= $ajouteEmbauche;
534         }
535     }
536     $content .= "</fieldset>";
537 }

```

Une fois tout est rempli et vérifié, le contrôleur qui se nomme **add-job.php** envoie les éléments du formulaire à la classe embauche pour que ça soit enregistré dans la table.

Pour cela on passe par la fonction **ajouterEmbauche(\$commentaire, \$contratType, \$dateEmbauche)**, cette fonction fait partie de la classe stage. Une fois la fonction en question est appelée, elle appelle une autre fonction **save()** de la classe embauche.

```

326     function ajouterEmbauche ($commentaire, $contratType, $dateEmbauche){
327         $embauche = new embauche();
328         $embauche->init( $this, $commentaire, $contratType, $dateEmbauche);
329         return $embauche->save();
330     }
331

```

- Partie statistique d'embauche :

Pour permettre aux étudiants et aux personnels de l'université de suivre les étudiants embauchés et le taux d'embauche des entreprises, nous avons créé un endroit où ces informations seront rassemblées et affichées.

Deux parties existent : la première est le suivi d'embauche, où les stages qui ont abouti à une embauche sont affichés, la deuxième c'est le classement des entreprises.

L'affichage des informations se diffère selon l'utilisateur connecté.

- Personnels de l'université :

Pour les administrateurs et les tuteurs, ils peuvent voir toutes les informations concernant l'étudiant, l'entreprise et le stage. Ils peuvent aussi contacter l'étudiant pour lui demander le déroulement de l'embauche ou d'autres informations...

Listing d'embauches

Voici la liste des étudiants embauché apres le stages.

Rechercher des informations sur l'embauche par critères :

Année : 2016/2017 ▼

Type de contrat : CDI ▼

Rechercher

Nom	Prénom	Stage	Entreprise	Date d'embauche	Contrat
Cham	Rémi	Développement d'un jeu vidéo sur PC	Studio M	2017-05-24	CDD
Idir	Fahem	Tierce maintenance applicative d'un outil de reporting BNP Paribas IP	IBM	2017-04-22	CDI
Yan	Shi	Mise en place d'un éditeur de profils de métadonnées pour une plate-forme de traitement d'information géographique	Geomatys	2017-04-20	CDD
El Ouasti	Redoine	Ingénieur Informatique – Langage Naturel – Cognitive Systems « Déploiement un prototype d'exécution de tâches à partir d'un agent conversationnel »	IBM	2017-04-18	CDD

- Affichage pour les étudiants :

Les étudiants ne peuvent pas voir qui a été embauché, par contre ils peuvent voir le stage pour avoir une idée sur les thèmes des stages qui ont abouti à une embauche.

Listing d'embauches

Voici la liste des étudiants embauché apres le stages.

Rechercher des informations sur l'embauche par critères :

Année : 2016/2017 ▼

Type de contrat : CDI ▼

Rechercher

Stage	Entreprise	Date d'embauche	Contrat
Développement d'un jeu vidéo sur PC	Studio M	2017-05-24	CDD
Tierce maintenance applicative d'un outil de reporting BNP Paribas IP	IBM	2017-04-22	CDI
Mise en place d'un éditeur de profils de métadonnées pour une plate-forme de traitement d'information géographique	Geomatys	2017-04-20	CDD
Ingénieur Informatique – Langage Naturel – Cognitive Systems « Déploiement un prototype d'exécution de tâches à partir d'un agent conversationnel »	IBM	2017-04-18	CDD
Stagiaire Ingénieur (H/F) – Développement Java d'interfaces homme-machine pour des logiciels de modélisation multi-physiques en environnement spatia	ARTENUM	0000-00-00	CDI

- Une partie en commun :

Il s'agit du classement des entreprises, peut import le rôle de la personne connectés elle aura le même affichage.

Cela est montre dans la figure ci-dessous.

Listing des entreprises qui embauche le plus	
Voici la liste des entreprises qui embauche le plus.	
Rechercher des informations sur l'embauche par critères :	
Année : 2016/2017 ▼	
Type de contrat : CDI ▼	
Rechercher	
Entreprise	nombre D'embauche
IBM	2
Geomatys	1
Studio M	1
ARTENUM	1

Il existe des modes de recherche et de classement, par exemple on peut chercher l'embauche de l'année précédente ou par type de contrat (CDI ou CDD).

Les deux types d'affichages sont gérés par deux scripts : **list-embauche.php** et **list-embauche-entreprise.php**, qui font appel à leur tour aux classes **listeEmbauche** et **listeEmbaucheEntreprise**.

7.1.16. Sauvegarde des pièces jointes

Elles seront sauvegardées directement dans la base de données sous forme de **blobs*** en binaire au lieu de le faire dans un dossier avec le code source de la plateforme, cela a été fait par étapes :

- Première modification: le script pour le téléchargement (Upload):

Pour éviter la surcharge sur le serveur, et vu que la base de données est intégrée dans un autre serveur on a opté pour la solution d'enregistrer les pièces jointes sur la base de données directement.

Un blob est un grand objet binaire qui peut contenir différents types de données. Il existe quatre types de blob : TINYBLOB, BLOB, MEDIUMBLOB et LONGBLOB. La seule différence entre les quatre types est la quantité de données acceptées. Dans notre cas nous avons choisi le type LONGBLOB. Les valeurs d'un BLOB sont traitées comme étant un texte binaire.

- Création de la table pgs_pieces_jointes :

Nous allons créer la table `pgs_pieces_jointes`, où seront insérés les fichiers mis par les utilisateurs. On pourra créer la table directement depuis PhpMyAdmin avec les lignes de requête MySQL, ou les inclure directement dans un fichier php. Voir la table en figure ci-dessus.

* : La création d'un tableau était impérative avec le type blob pour les fichiers à enregistrer.

```
96 CREATE TABLE `pgs_pieces_jointes` (  
97   `IDPIECEJOINTE` int(11) NOT NULL,  
98   `IDSTAGE` int(11) NOT NULL,  
99   `NOM` varchar(50) DEFAULT NULL,  
100  `TYPE` varchar(30) DEFAULT NULL,  
101  `IDUSER` int(11) NOT NULL,  
102  `ANNEE` int(11) NOT NULL,  
103  `EXTENSION` varchar(20) DEFAULT NULL,  
104  `SIZE` int(11) NOT NULL,  
105  `CONTENT` longblob NOT NULL  
106 )  
107
```

Il ne faut pas que les fichiers mis par les utilisateurs dépassent une certaine limite dans ce cas : **LONGBLOB < 4294967296 Octets, c.à.d. < 4096 Mo = 4 Go.**

- Stockage de fichier dans les tables :

Tout d'abord il faut que le champ du formulaire qui va comprendre la pièce jointe soit encrypté

```
<legend class="legendh2">Nouvelle pièce jointe .doc/.pdf/.odt/.docx/.txt</legend>  
<form enctype="multipart/form-data" method="POST" name="formdisplaystage" action="display-stage.php"="">
```

pour les fichiers binaires, ce qu'on peut voir dans la figure suivante,

Après avoir signalé qu'on attend un fichier dans le formulaire avec la méthode POST, on va traiter le fichier dans un autre PHP. Avant d'enregistrer le fichier nous allons vérifier si ce n'est pas un faux, sa taille et son extension.

```
$piecejointe = new PieceJointe();
$fp = fopen($tmpName, 'r');
$content = fread($fp, filesize($tmpName));
// $content = addslashes($content);
$piecejointe->nom = escapeBalises($nomFichier);
$piecejointe->type = $_POST['typePj'];
$piecejointe->idStage = $modifStage->id;
// $piecejointe->idUser = $modifStage->idUser;
$piecejointe->annee = date("Y");
$piecejointe->extension = $type;
$piecejointe->size = $taille;
$piecejointe->content = $content;
$piecejointe->save();
// check if file exist
if($piecejointe->fileExist() > 1){
    $_SESSION['msg'] = "Erreur fichier existe deja ". $piecejointe->fileExist();
    $piecejointe->supprPieceJointe();
    redirect("display-stage.php?idstage=".$modifStage->id);
}
else{
    if(fopen($tmpName, 'r')){
        fclose($fp);
        $messAjoutPj = "<p class='messageOK'>Document correctement ajout&eacute;. ". $PATH.$
            DirUploads.$piecejointe->idPieceJointe.$piecejointe->type."stage".$modifStage->id.".".
            $type."</p>";
    }
    // d?placement de la pi?ce jointe
    else{
        $_SESSION['msg'] = "D?placement du fichier impossible : ". $PATH.$DirUploads.$piecejointe->
            idPieceJointe.$piecejointe->type."stage".$modifStage->id." l'erreure est : ".$_FILES['
            fichier']['error'];
        $piecejointe->supprPieceJointe();
        redirect("display-stage.php?idstage=".$modifStage->id);
    }
}
```

Les fichiers acceptés sont seulement les fichiers .docx, .pdf, .odt, .doc et .txt . Les plus communément utilisés.

Dans cette partie aucune modification n'est faite par rapport aux années précédentes, sauf pour : la classe la classe **PiecesJointes** qui a été modifier vu qu'on a un nouveau attribut dans notre table qui est « CONTENT », alors qu'avant on enregistre que le chemin où le fichier est transféré. La particularité dans ce cas est le fait qu'on est obligé de lire le fichier avant de l'enregistrer, puis on vérifie si pour ce stage il n'existe pas dans la table déjà une pièce jointes avec le même nom, cela est le travail de la fonction **fileExist()** qui renvoi un chiffre supérieur a >1 si cette dernière condition est vérifier, donc on supprime si le fichier est ajouté vu qu'on fait un save juste avant. A la fin on est redirigé vers la page du stage courant.

La fonction **save()** fait un ajout dans la tables pgs_pieces_jointes. Tous les éléments sont récupérer dans le script ci-dessus qui sont : le nom, extension, type, idStage ... etc

- Téléchargement d'un fichier enregistré :

Affichage des pièces jointes :

Une fois sur la liste d'un stage l'id de stage est récupérable avec un simple **\$_GET['idstage']**, une fois l'id est récupéré, on appelle la fonction **getPiecesJointesByStage(\$stage)**. La variable \$stage est une variable de type Stage donc il faut la créer avec le id récupérer récemment.

La fonction mentionnée précédemment renvoie une liste de pièces jointes et le contrôleur « **contrôleur-display-stage.php** » s'occupe de l'affichage. Dans la figure suivante on peut voir le script qui renvoie la liste des pièces jointes par stage.

```
function getPiecesJointesByStage($stage)
{ // retourne les pièces jointes sur le stage $stage

    global $BD, $bdPiecesJointes;
    $requete = "SELECT IDPIECEJOINTE FROM " . $bdPiecesJointes . " WHERE IDSTAGE = '" . $stage->id."'";
    //print($requete)."<br>";
    $res=$BD->query($requete);
    if ( $BD->numRows == 0 )
    {
        $this->liste=array();
        return null;
    }
    while ( $BD->next($res) ){
        $row = $BD->result;
        $piecejointe = new Piecejointe($row->IDPIECEJOINTE);
        array_push($this->liste,$piecejointe);
    }
}
```

Téléchargement :

Au moment d'affichage des pièces jointes pour chaque stage on ajoute aux balises des **<href>**, le nom de la pièce jointe, id du stage et le type et une redirection vers un script qui fait le travail d'envoi au client, de la pièce jointe depuis le serveur. Il s'agit de **get-file.php** ou la fonction **download()** qui se trouve dans la classe des pièces jointe, est appelé .

```
if(isset($_GET['idStage'])){

    $piecejointe = new PieceJointe();
    $piecejointe->nom = $_GET['Pjnom'];
    $piecejointe->idStage = $_GET['idStage'];

    list($name, $type, $size, $content)=$piecejointe->download();
    header("Cache-Control: no-cache private");
    header("Content-Description: File Transfer");
    header("Content-disposition: attachment; filename=".$name);
    header("Content-Type: application/octet-stream");
    header("Content-Transfer-Encoding: binary");
    header('Accept-Ranges: bytes');
    header('Content-Length: ' . strlen($content));
    header("Pragma: no-cache");
    header("Expires: 0");
    ob_clean();
    flush();
    echo $content;
    exit;
}
```

Fonctionnement:

On doit créer une instance de la classe **PieceJointe()**, on lui attribue après le nom et le idStage qui étaient inclus dans l'adresse de redirection et récupérer avec les **\$_GET**, comme montré sur la figure ci-dessus.

Nous devons créer après une liste qui prenne les éléments renvoyé par la fonction **download()** (nom de la pièce jointe, le type, la taille et le contenu binaire). Ensuite les entête: « **header** » sert à préparer le fichier au téléchargement, parmi ces entêtes, il en existe un indispensable pour le bon fonctionnement, il s'agit de l'entête « **header("Content-Transfer-Encoding: binary");** » pour signaler qu'on va passer du binaire au format original.

Les deux ligne qui précède le **echo \$content (ob_clean() et flush())** servent à nettoyer le tampon de sortie donc s'il existait des **echo** avant ces deux ligne il seront vider sinon on risque d'avoir des messages dans le fichier de sortie.

Finalement, **echo \$content** imprime le contenu du fichier dans le fichier a téléchargé.

Remarque : le téléchargement a été généralisé partout pas seulement pour les stages, donc on trouve le même principe dans le profil des étudiants pour que les entreprises puissent les voir. La même table est utilisée pour enregistrer les pièces jointes.

7.1.17. Dématérialisation de la fiche de suivi étudiant

Le but étant de ne plus avoir de fiche à télécharger, imprimer et envoyer pour remplir par l'intéressé, puis scanner et enrichir la base de données des différentes fiches, mais aussi l'archiver au secrétariat. À l'heure du numérique, toutes ses actions pouvant se faire sans avoir à utiliser ni papier ni stylo, cela restera une option seulement.

Nous avons décidé de prioriser la fiche de suivi du stage à remplir par l'étudiant, à fin de voir son constat à propos du stage et du tuteur qui le suit en stage, notamment les contacts entre eux et les visites. Puis éventuellement étendre cette action à toutes les fiches de la plateforme.

Dans un premier temps, l'étudiant authentifie sur la plateforme, étant associé un stage, voit sa fiche pré-remplie lorsqu'il s'apprête à la saisir, comme le montre la figure suivante qui affiche le contenu du fichier **create-fiche-suivi-etude.php**:

(Les différents noms affichés sont à titre indicatif)

The screenshot displays a web application interface for a student's stage follow-up form. The top navigation bar indicates the user is logged in as 'mailetud@gmail.com' and provides links for 'Modifier son mot de passe' and 'Deconnexion'. The left sidebar contains three main sections: 'Général' (with links for 'Accueil' and 'Profil'), 'Stages' (with links for 'Mon Stage', 'Listing des stages', 'Télécharger la fiche', and 'Proposer un stage'), and 'Soutenance' (with links for 'Planning de soutenance', 'Statistique enseignants', 'Télécharger la fiche stage entreprise', and 'Saisir la fiche de suivi étudiant'). The main content area is titled 'Fiche de suivi étudiant' and contains a form with the following pre-filled data: Nom: BENSADOUNE, Prénom: Mohand, Parcours: IPS, Niveau: M2, Enseignant(e) référent(e): Aloulou Hamdi, and Tuteur en entreprise: GUERFI Larbi. Below this, there is a section for 'Suivi' with instructions to indicate contacts with the referring teacher. It includes a date field (aaaa-mm-jj), radio buttons for 'Type de contact' (Telephone, Mail, Visite), and a text area for 'Commentaires'. The form is designed to be user-friendly and efficient for data entry.


```

21 $ficheSuivi=new FicheSuiviEtud();
22 $ficheSuivi=new FicheSuiviEtud($ficheSuivi->getIdFicheByIdStage($_GET['idStage']));
23 $commentaireBilan=$ficheSuivi->commentaireBilan;
24 $embaucheOuiNon=$ficheSuivi->embauche;
25 $pourquoi=$ficheSuivi->commentaireEmbauche;
26
27 //Write the page
28 $pdf = new FPDF("P","mm","A4");
29 ob_start();
30 $pdf->AddPage();
31 $pdf->SetMargins(20,20);
32 $pdf->Image('images/logo_UM.png',15,15,22,22);
33 $pdf->Image('images/logo_FDS.jpeg',37,15,18,22);
34 $pdf->Image('images/logo_INFORMATIQUE.jpeg',165,15,32,22);
35 $pdf->SetFont('Arial','B',18);
36 $pdf->Cell(0,70,'Fiche de suivi de stage en entreprise',0,0,'C');
37 $pdf->Ln(5);
38 $pdf->SetFont('Arial','',8);
39 $pdf->Cell(0,70,utf8_decode('à compléter par l\'étudiant'),0,0,'C');
40 $pdf->Rect(15,55,180,30);
41
42 $pdf->SetFont('Arial','B',15);
43
44
45 $pdf->Ln(10);
46 $pdf->SetFont('Arial','',12);
47 $pdf->Cell(0,70,utf8_decode("Nom de l'étudiant : ".$etudiant->nom." ".$etudiant->prenom),0,0,'L');
48 $pdf->Ln(10);
49 $pdf->Cell(0,70,utf8_decode("Nom du tuteur universitaire : ".$tuteuruniv->nom." ".$tuteuruniv->
    prenom),0,0,'L');
50 $pdf->Ln(10);
51 $pdf->Cell(0,70,utf8_decode("Nom du tuteur entreprise : ".$tuteurent->nom." ".$tuteurent->prenom),0,
    0,'L');
52 $pdf->Ln(10);
53 $pdf->SetFont('Arial','',8);
54 $pdf->Cell(0,70,utf8_decode("Suivi : indiquer les différents contacts que vous avez pu avoir avec
    votre enseignant référent."),0,0,'J');

```

Voici à quoi ressemble une partie d'un fichier PDF généré ainsi...

Si l'étudiant a déjà rempli sa fiche de suivi, un fichier comme l'image à droite sera générée, sinon c celle à gauche qui est vide.

Fiche de suivi de stage en entreprise																	
à compléter par l'étudiant																	
<p>Nom de l'étudiant :</p> <p>Nom du tuteur universitaire :</p> <p>Nom du tuteur entreprise :</p>																	
<p>Suivi : indiquer les différents contacts que vous avez pu avoir avec votre enseignant référent.</p> <table border="1"> <thead> <tr> <th>Date</th> <th>Type de contact</th> <th>Commentaires</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </tbody> </table>			Date	Type de contact	Commentaires												
Date	Type de contact	Commentaires															
<p>Date de début du stage:</p> <p>Date de fin du stage:</p> <p>Bilan du stage pratique (sur l'entreprise, votre encadrement, l'adéquation de votre formation, autre...):</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>L'entreprise est-elle prête à embaucher le stagiaire ? Oui Non</p> <p>Pourquoi ?</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>Fait le, Signature</p>																	
<p>Remarque : cette fiche nous sert à faire des statistiques et ne contribue en rien dans la note finale de stage.</p>																	

Fiche de suivi de stage en entreprise																	
à compléter par l'étudiant																	
<p>Nom de l'étudiant : Etudiant etu</p> <p>Nom du tuteur universitaire : DONY christophe</p> <p>Nom du tuteur entreprise : CHALAL ZAAK</p>																	
<p>Suivi : indiquer les différents contacts que vous avez pu avoir avec votre enseignant référent.</p> <table border="1"> <thead> <tr> <th>Date</th> <th>Type de contact</th> <th>Commentaires</th> </tr> </thead> <tbody> <tr> <td>2017-09-17</td> <td>Telephone</td> <td>encore appel</td> </tr> <tr> <td>2017-09-18</td> <td>Visite</td> <td>il m'a rendu visite pour profiter du resto de ma compagnie</td> </tr> <tr> <td>2017-09-20</td> <td>Mail</td> <td>quelques mails pour savoir ou en est mon travail et me donner des conseils</td> </tr> <tr> <td>2017-09-19</td> <td>Telephone</td> <td>il m'a appelé pour prendre de mes nouvelles</td> </tr> </tbody> </table>			Date	Type de contact	Commentaires	2017-09-17	Telephone	encore appel	2017-09-18	Visite	il m'a rendu visite pour profiter du resto de ma compagnie	2017-09-20	Mail	quelques mails pour savoir ou en est mon travail et me donner des conseils	2017-09-19	Telephone	il m'a appelé pour prendre de mes nouvelles
Date	Type de contact	Commentaires															
2017-09-17	Telephone	encore appel															
2017-09-18	Visite	il m'a rendu visite pour profiter du resto de ma compagnie															
2017-09-20	Mail	quelques mails pour savoir ou en est mon travail et me donner des conseils															
2017-09-19	Telephone	il m'a appelé pour prendre de mes nouvelles															
<p>Date de début du stage: 2017-03-08</p> <p>Date de fin du stage: 2017-10-26</p> <p>Bilan du stage pratique (sur l'entreprise, votre encadrement, l'adéquation de votre formation, autre...):</p> <p>le stage a été une superbe experience pour moi, j'ai vu un peu en quoi ressemble le monde professionnel, j'ai pris pas mal de cafés par contre</p> <p>L'entreprise est-elle prête à embaucher le stagiaire ? Non</p> <p>Pourquoi ?</p> <p>malheureusement il y a pas suite a mon stage</p> <p>Fait le, Signature</p>																	
<p>Remarque : cette fiche nous sert à faire des statistiques et ne contribue en rien dans la note finale de stage.</p>																	

Cela dit, quant aux différents rôles, leurs options de création (C), vue ou lecture (R), modification (U) et suppression (D), nous avons résumé cela dans le tableau suivant :

Fiche de suivi de stage étudiant	Etudiant	Tuteur	Rapporteur	Administrateur	Secrèterait	Entreprise
Droits :	CRUD	R	R	CRUD	R	R

7. Conclusion

Ce TER nous a mené, maint fois, à exprimer non seulement nos compétences et nos acquis pendant cette année, mais à améliorer notre homogénéité à travailler en groupe et à partager et organiser les différentes problématiques rencontrée au fil du travail.

Principalement, le travail consistait en quelque sorte à faire du « Refactoring », c.à.d. au réusinage de code déjà prêt, mais aussi, nous avons eu à améliorer et penser de nouvelles solutions. L’aspect inattendu d’un PHP orienté objet reste indiscutablement ‘‘Le’’ gain le plus important à nos yeux pendant ce TER(du très petit sommet de notre expérience), connu comme un langage assez brut, nous avons pu redécouvrir un PHP très intéressant, bien carré.

8. Référence

https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet

<https://openclassrooms.com/courses/programmez-en-orientee-objet-en-php/introduction-a-la-poo>

<http://bugzilla.readthedocs.io/en/latest/installing>

<http://www.fpdf.org>

Brad et al. 1989 Object-Oriented Programming: An Evolutionary Approach

M. Hajmiragha - Document management system 2001

L. Welling & L. Thomson - 2009 PHP & MySQL

M. Motwani et al. 2005

Patten et al. 2015