

## Introduction :

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 une box déjà existante et de générer un lien pour chaque fichier vers la 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](http://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](#)).

## Lors du 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 données ne marche pas un 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()`. 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.

### 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 ecriture 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.

### Résolution des bugs de la plateforme :

Voici les différents bugs résolus et améliorations apportées au gestionnaire des stages :

1. Généraliser le fait de switcher les 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          }  
66      }
```

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              $identif.='<option value=\"ETUD\">ETUD</option><option value=\"RESP\">RESP</option>  
230              <option value=\"ENT\">ENT</option><option value=\"SECR\">SECR</option><option value=\"TUT\">TUT</  
231                  option><option value=\"INV\">INV</option></select>&nbsp;<input type=\"submit\" value=\"Go  
232                  name=\"send\" /></form>';  
233          }  
234      }
```

2.Améliorer 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     {
61         $content.="<tr><td><b>Nombre de salari  s : </b></td><td> --- </a></td></tr>\n";
62     }
63     else
64     {
65         $content.="<tr><td><b>Nombre de salari  s : </b></td><td> ". $entreprise->nbSalaries."</a></td></tr>\n";
66     }
67     if($entreprise->local!="") {
68         $content.="<tr><td><b>Poss  de un local?</b></td><td>";
69         if ($entreprise->local == 1) {
70             $content .=" Oui</td></tr>\n";
71         }
72         else if ($entreprise->local == 0) {
73             $content .=" --- </td></tr>\n";
74         }
75     }
76 }

```

La m  me modification a   t   apport  e sur le fichier **display-stage.php** dans la section de l'affichage des informations concernant l'entreprise.

3.L'export des stages sous format csv depuis la plateforme. Un fichier a d  j     t   cod   par les   tudiants de l'ann  e pass  e mais, la requ  te SQL consultant la base de donn  es a du   tre changer pour son bon fonctionnement. Dans la requ  te, il a fallu faire plein de jointure pour r  cup  rer diff  rentes informations dans diff  rentes tables.

Voici le changement apport   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
19             FORMATION,a.TITRE FROM pgs_stage a,pgs_stage_note b,pgs_utilisateur c,pgs_etudiant d,pgs_parcours e WHERE
20             a.IDETUDIANT=c.IDUSER AND a.IDSTAGE=b.IDSTAGE AND e.idParcours=d.idParcours AND c.IDUSER=d.idEtudiant
21             AND a.DATEDEBUT<'". $annees[1]."-06-30' AND a.DATEDEBUT>'". $annees[0]."-09-01' UNION SELECT DISTINCT
22             c.NOM,c.PRENOM,d.numeroEtudiant,Null as VALEURNOTE,e.niveau,e.acronyme as FORMATION,a.TITRE FROM
23             pgs_stage a,pgs_utilisateur c,pgs_etudiant d,pgs_parcours e WHERE a.IDETUDIANT=c.IDUSER AND
24             e.idParcours=d.idParcours AND c.IDUSER=d.idEtudiant AND a.DATEDEBUT<'". $annees[1]."-06-30' AND
25             a.DATEDEBUT>'". $annees[0]."-09-01' ORDER BY FORMATION ");
26     }
27     else{
28         $req=$PDO->prepare("SELECT DISTINCT c.NOM,c.PRENOM,d.numeroEtudiant,b.VALEURNOTE,e.niveau,e.acronyme as
29             FORMATION,a.TITRE FROM pgs_stage a,pgs_stage_note b,pgs_utilisateur c,pgs_etudiant d,pgs_parcours e WHERE
30             a.IDETUDIANT=c.IDUSER AND a.IDSTAGE=b.IDSTAGE AND e.idParcours=d.idParcours AND c.IDUSER=d.idEtudiant
31             UNION SELECT DISTINCT c.NOM,c.PRENOM,d.numeroEtudiant,Null as VALEURNOTE,e.niveau,e.acronyme as
32             FORMATION,a.TITRE FROM pgs_stage a,pgs_utilisateur c,pgs_etudiant d,pgs_parcours e WHERE a.IDETUDIANT=c.
33             IDUSER AND e.idParcours=d.idParcours AND c.IDUSER=d.idEtudiant ORDER BY FORMATION");
34     }
35     $req->execute();
36     $data=$req->fetchAll();
37     CSV::export($data,'export stage');
38 }

```

4.Quelques champs dans le formulaire de cr  ation de stage ont   t   mis en obligatoire pour   viter d'avoir plein de vides dans la base de donn  es. Par exemple, le titre de stage, le niveau requis...

Donc le fichier **create-stage.php** a   t   un peu modifi   :

```

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", "", "
327         inputtext", 65).</td></tr>

```

Mais le fait de faire   a a caus   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   a 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>
```

Et 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 ?>
```

5. Amélioration du pop-up du calendrier pour choisir les dates. 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="'. $cal->get_strPopup("calendrier/pop.calendrier.php", 350, 170). "'></a></td></tr>
```

En suite 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 **/includes/class.calendrier.php**, on apporte le changement suivant :

>>>>image à venir

6. Faire une option de multi sélection 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 check box 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 checks box dans un formulaire,

```

368         if ($role == SECURITY_ADM or $role == SECURITY_RESP or $role == SECURITY_SECR){
369             $content.="<td>".$HTML->getInputCheckBoxJS('multichoiceselect'.$i, $stage->id), "show_toolbar();")."
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, **multi\_select\_options.php** :

```

1  <?php
2  include_once("includes/inc.includes.php");
3  $user=new Utilisateur($currentUser->id);
4  $security = new Security($currentUser);
5  $security->autorise(SECURITY_RESP);
6  $security->autorise(SECURITY_ADM);
7  $security->autorise(SECURITY_SECR);
8
9  if (!$security->canDisplayPage()){
10     $_SESSION['msg'] = "Vous n'avez pas les autorisations requises pour faire des modifications";
11     redirect("list-stages.php");
12 }
13
14 if ( $user->role == SECURITY_RESP or $user->role == SECURITY_ADM or $user->role == SECURITY_SECR){
15
16     if ($_GET["validate"]){
17         foreach ($GET as $key => $value) {
18             if ($key!='validate'){
19                 //traitement
20                 $stage=new Stage($value);
21                 if ($stage->etat=="PROPOSEE"){
22                     $stage->modifierEtatStage("VALIDE");
23                     $_SESSION['msg'] = "Stages valid&eacute;s";
24                 }
25                 echo $stage->etat;
26             }
27         }
28     }
29     if ($_GET["erase"]){
30         foreach ($GET as $key => $value) {
31             if ($key!='erase'){
32                 //traitement
33                 $stage=new Stage($value);
34                 $stage->effacerStage();
35                 $_SESSION['msg'] = "Stages supprim&eacute;s";
36             }
37         }
38     }
39 }
40
41 $_SESSION['msg'] .= "/Modifications enregistr&eacute;es";
42 redirect("list-stages.php");
43 }
44 ?>

```

7. Ajouter les colonnes étudiant et tuteur dans le listing des 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         $etudiant->prenom.&nbsp;&nbsp;&\">$etudiant->nom. "</a></td>";
395         $content .= "<td><a href=\"display-user.php?iduser=" . $tuteur->id . "\">
396         $tuteur->prenom.&nbsp;&nbsp;&\">$tuteur->nom. "</a></td>";
397     }

```

8. 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 accessible 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         $content.="<td><input type=\"button\" value=\"Modifier\" onClick=\"javascript:window.location=\"'
112         edit-stage.php?act=edit&idstage='.$stage->id.'\"></td>";

```

9. Enregistrer les motifs de modification des soutenances par les utilisateurs. Pour ça, une class **includes/class.soutenance.modif.php** a été ajoutée 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>'.<input type="text" value="">'.</td>
192     </tr>

```

Grâce 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 ça méthode sont utilisées :

```

63 class ListeModifSoutenances{
64     var $liste;
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** :

```

92 $lastModif='<table width="100%"><thead>
93     <tr width="100%"><td width="200px"><b>Modificateur</b></td><td><b>Motif de modification</b></td></tr></thead><tbody>';
94 include_once("includes/class.soutenance.modif.php");
95 $ListeModifs=new ListeModifSoutenances();
96 $ListeModifs->getModifications($GETsoutenance->idSoutenance);
97 if (count($ListeModifs->liste)==0){
98     $lastModif.="<tr><td><center>Aucune ancienne modification</center><td></td></tr>";
99 }
100 elseif ((count($ListeModifs->liste)>0) && (count($ListeModifs->liste)<4)){
101     for ($i=0; $i < count($ListeModifs->liste); $i++){
102         $modif=$ListeModifs->liste[$i];
103         $userModif=new Utilisateur($modif->idUser);
104         $lastModif.="<tr><td>".$userModif->nom.' '.$userModif->prenom.' ('.$userModif->role.')</td><td>'.$modif->motif.'</td></tr>';
105     }
106     $lastModif.='</tbody></table>';
107
108
109 }
110 else{
111     for ($i=0; $i < 3; $i++){
112         $modif=$ListeModifs->liste[$i];
113         $userModif=new Utilisateur($modif->idUser);
114         $lastModif.="<tr><td>".$userModif->nom.' '.$userModif->prenom.' ('.$userModif->role.')</td><td>'.$modif->motif.'</td></tr>';
115     }
116     $lastModif.='</tbody></table>';
117     $lastModif.='<table id="plusmodifsout" style="display:none" width="100%"><thead width="100%"></thead><tbody width="100%">';
118     for ($i=3; $i < count($ListeModifs->liste); $i++){
119         $modif=$ListeModifs->liste[$i];
120         $userModif=new Utilisateur($modif->idUser);
121         $lastModif.="<tr width="100%"><td width="200px">".$userModif->nom.' '.$userModif->prenom.' ('.$userModif->role.')</td><td>'.$modif->motif.'</td></tr>';
122     }
123     $lastModif.='</tbody></table>';
124
125 }
126
127
128 if (count($ListeModifs->liste)>5){
129     $lastModif.='<center><a id="afficherplusmodifsoutenances" href="javascript:void(0)"
130         onClick="javascript:Plus_Modif(this)">Plus</a></center>';
131 }

```

Et affichée dans un fieldset :

```
132 $content.='<div class="divLastModificationSoutenance">
133 <fieldset>
134 <legend>Dernières Modifications</legend>
135 <br />
136 '.$lastModif.'
137 </fieldset>
138 </div>
139 <br/>';
```

Et voici la fonction JavaScript qui permet d'afficher plus que trois modifications :

```
29 function Plus_Modif(a){
30     if (a.innerHTML=="Moins"){
31         plusmodifsout.style.display='none';
32         a.innerHTML="Plus";
33     }
34     else{
35         plusmodifsout.style.display='block';
36         a.innerHTML="Moins";
37     }
38 }
```

Et 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**:

```
224 if ($role==SECURITY_ETUD || $role==SECURITY_TUT){
225     $content.=' onclick="Javascript:Modify_Soutenance(\'Popup_Confirmation_Modification_Soutenance\'.'.$creneau->idCreneau.'\');" ' ;
226     $content.=' class="creneauCalendrier cliquable" ' ;
227     $content.=' class="creneauCalendrier" >';
228
229     $content2.='
230     <div id="Popup_Confirmation_Modification_Soutenance\'.'.$creneau->idCreneau.'"
231     style="display:none">
232     <form method="POST" action="affectationSoutenance.php?idCreneau='.$creneau->idCreneau.'">
233     if ($role==SECURITY_ETUD){
234         $content2.='<b><p>Votre soutenance sera dans le nouveau cr&eacuteneau suivant: '.$creneau->dateDeb.' &agrave; '.$creneau->heureDeb.'</p></b>';
235     }
236     if ($role==SECURITY_TUT){
237         $content2.='<b><p>Vous serez comme deuxi&egraveme tuteur &agrave; la soutenance
238         organis&eacute;e le: '.$creneau->dateDeb.' &agrave; '.$creneau->heureDeb.'</p></b>';
239     }
240     $content2.='<table width="100%">
241     <tr>
242     <td>Motif de modification :</td>
243     <td>'.$HTML->getInputTextAreaRequired("motifModification").'</td>
244     </tr>
245     <tr>
246     <td><input type="button" value="Retourner vers le planning"
247     onclick="Javascript:Modify_Soutenance(\'Popup_Confirmation_Modification_Soutenance\'.'.$creneau->idCreneau.'\');" /></td>
248     <td>'.$HTML->getInputSubmit("send", "Valider les changements").'</td>
249     </tr>
250     </table>
251     </form>
252     </div>';
253 }
254 else{
255     $content.=' onclick="window.location=\'affectationSoutenance.php?idCreneau='.$creneau->idCreneau.'\' " ' ;
256     $content.=' class="creneauCalendrier cliquable" ' ;
257     $content.=' class="creneauCalendrier" >';
258 }
```

Et pour afficher ce formulaire de confirmation, on fait appel à la fonction JavaScript suivante :



```

13 function Modify_Soutenance(e){
14     var idCreneau=document.getElementById(e);
15     if (idCreneau.style.display=='block') {
16         idCreneau.style.display='none';
17         idCalendrier.style.display='block';
18     }
19     else{
20         idCreneau.style.display='block';
21         idCalendrier.style.display='none';
22     }
23 }

```

10. Ne pas afficher les soutenances de l'année passée pendant l'année courante et 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 $strdat1='30-09-' . ($dd->format('Y'));
59 $strdat2='01-04-' . ($dd->format('Y'));
60 $datecomp1=new DateTime($strdat1);
61 $datecomp2=new DateTime($strdat2);
62 $dateFermeturemodifSoutenance=$dd->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 le fermeture des modifications de soutenances est effectuée une semaine avant le début :

```

72 if ($dateFermeturemodifSoutenance>$dateAct){
73     $content.='<p style="color:green ">La modification sera ferm&eacute;e le:'.$
74     dateFermeturemodifSoutenance->format('d/m/Y'). '</p>';
75 }
76 else{
77     $content.='<p style="color:red ">La modification est ferm&eacute;e le:'.$dateFermeturemodifSoutenance
78     ->format('d/m/Y'). '</p>';
79 }

```

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){

```

**References :**

M Hajmiragha Document management system 2001

Patten et al. 2015

Motwani et al. 2005

[www.fpdf.org](http://www.fpdf.org)

L.Welling & L.Thomson 2009 PHP & Mysql