DokuMission

Contexte

Développement

Il s'agit d'adapter pour l'entreprise de créer un gestionnaire de documentation, qui permettra aux salariés de gérer/créer puis consulter/rechercher dans une documentation partagée.

Outils de développement

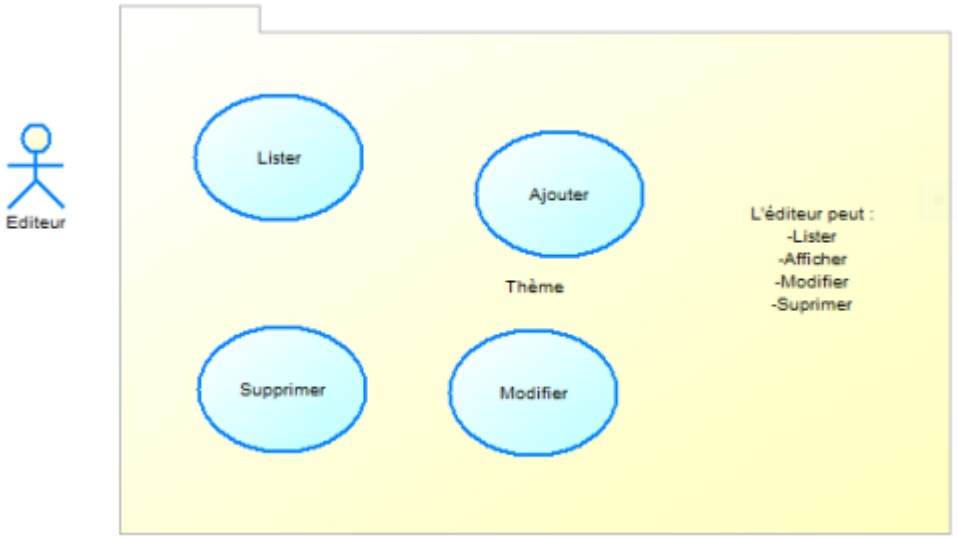
- PHP/Mysql
- Doctrine pour le mappage relationnel/Objet
- CodeIgniter pour la mise en place MVC

Travail fourni (Thème)

Document

Libelle	Document
Scénarii	desciptif_textuel.pdf
Base de donnée	docu_4_.sql
Application	dokumission.zip

Diagramme de cas d'utilisation des thèmes



Modéle MVC

Modéle

Pour pouvoir travailler sur les thèmes, il a fallu agir sur 4 classes métier:

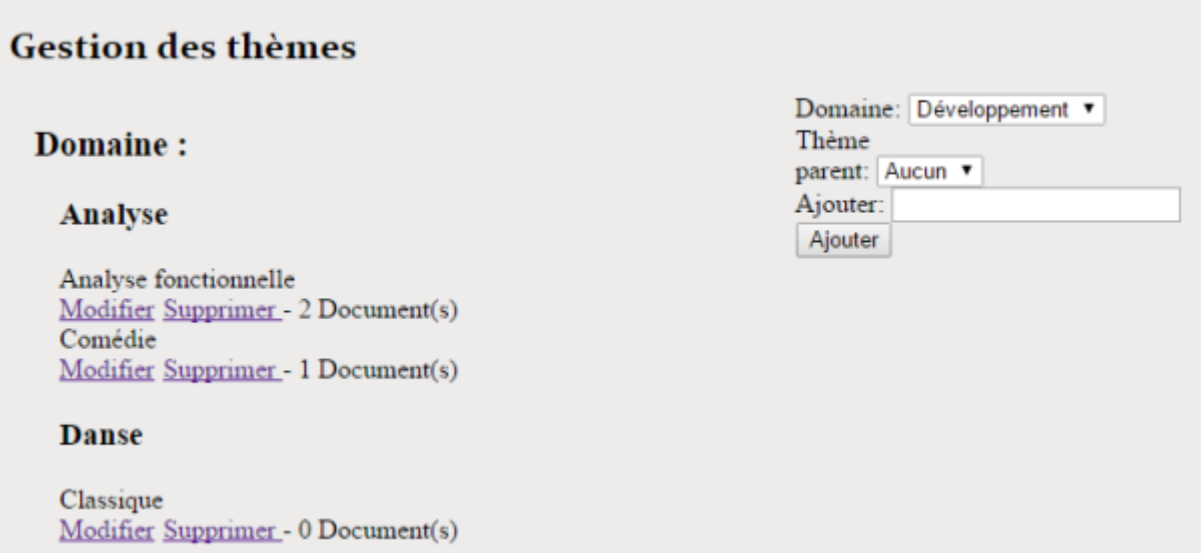
1. Domaine
2. Thème
3. Document
4. Utilisateur

- Un utilisateur est associé à des documents s'il en possède.
- Un utilisateur peut travailler dans un ou plusieurs domaines.
- Les documents sont identifiés dans des thèmes.
- Un thème appartient à un domaine.
- Un thème peut posséder un thème parent.
- Un thème appartient à un utilisateur.

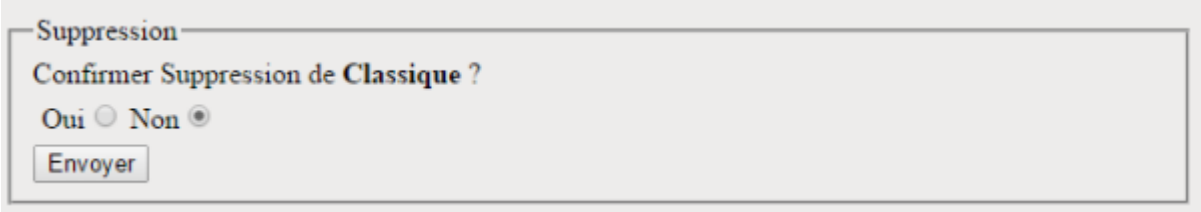
Vue

Pour la gestion des thèmes, 3 vues sont utilisées:

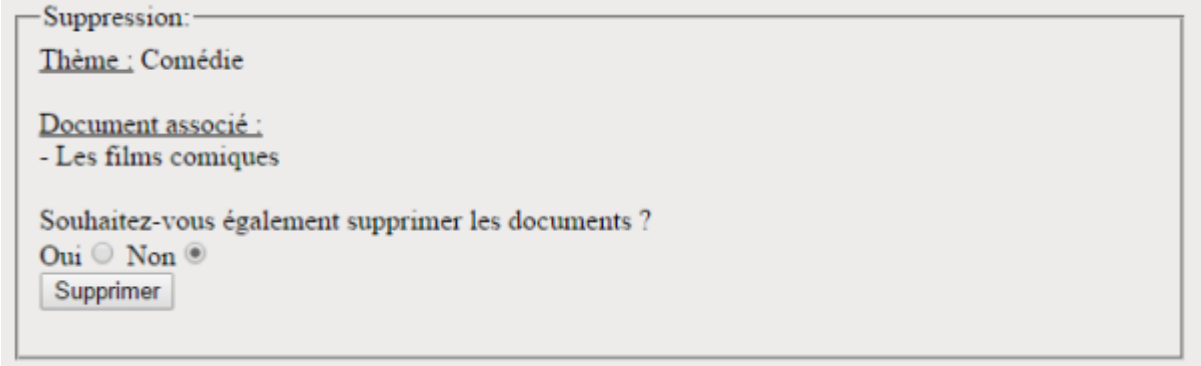
- v_theme (Vue générale, liste les thèmes triés + formulaire d'ajout et de modification)



- v_confirmdeletetheme (La vue s'affiche si aucun document n'est affecté)



- v_formdeletetheme (Avant suppression, on demande si l'on souhaite garder les documents)



Contrôleur

La gestion des thèmes est effectuée dans Gtheme. Le contrôleur possède plusieurs méthode pour optimiser la mise en place des fonctionnalités.

Méthodes	**Description**
index	Méthode appelée par défaut (appelle les vue supplémentaire tel que le menu, footer...)
refresh	Appelée par l'index, initialise les outils nécessaires au premier affichage de la page (Paramètre de la BDD/fonction JS)
addParentThem/updateParentThem	Ces méthodes ajoutent/modifient l'ensemble des thèmes qui existent pour le domaine choisi pour les ajouter dans les listes déroulante
add	Ajoute un nouveau thème
theme_modif	Est appelée quand on veut modifier un thème, celle-ci pré-remplit le domaine/thème parent
update	Met à jour les informations modifiées
deleteForm	Cette méthode vérifie si le thème possède des documents, suivant le résultat la vue affichée ne sera pas la même
checkConfirmDelete	Appelée par **deleteForm** s'il n'y a pas de document
delete	Supprime le thème
saveDocs	Vérifie si l'utilisateur a décidé de garder ou non les documents
updateDocs	Dans la table document, met l'identifiant de thème à nul
refresh	Supprime le document

Librairie

Appelée dans le constructeur : *$this→load→library('Modelutils');* Utilisation de la librairie **ModelUtils**, dans cette classe 2 modèles ont étés créés.

- cleanPost (nettoie les variables)
- ifempty (Reçoit un tableau pour vérifier chaque contenu, si vide, retourne false)

```
<?php
/**
     * Récupére un array puis vérifie si une variable est vide
     * @param $params
     * @return boolean
     */
    public function ifempty($params=array()){
        $checked=true;
```

```
        foreach ($params as $param){
            if($checked==true){
                if(empty($param)){
                    $checked=false;
                }
            }
        }
        return $checked;
    }
```

Requêtes doctrine utilisées

Les méthodes doctrine sont appelées de la manière suivante : *$this→doctrine→em→*

Pour un *Select multiple*:

```
<?php
      //Création de la requête paramêtrée
      $queryDomaine = $this->doctrine->em->createQuery("SELECT d FROM Domaine d
JOIN d.monde m WHERE m.id=:utilisateur");
      //Paramêtre
      $queryDomaine->setParameter('utilisateur',$user);
      //Récupére tous les résultats
      $domaine = $queryDomaine->getResult();
```

Pour un *Select* (une seul ligne) :

```
<?php
     $queryThemeid=$this->doctrine->em->createQuery("SELECT t FROM theme t WHERE
t.id=:theme");
     $queryThemeid->setParameter('theme',$themeid);
     //Récupére un seul résultat
     $themeid=$queryThemeid->getSingleResult();
```

Pour un *Update et Insert* :

```
<?php
    //Envoi d'une instance
    $this->doctrine->em->persist($theme);
    //Execution de la requête
    $this->doctrine->em->flush();
```

Jsutils

Liste des méthodes utilisées :

1. *postFormAndBindTo* Poste le formulaire
2. *getAndBindTo* Effectue une action suite à un événement (les paramètres de l'id Html sont envoyés automatiquement)
3. *click* Lors d'un clique sur un élément on appelle une autre méthode
4. *show* Affiche l'élément mis en paramètre
5. *hide* Cache l'élément mis en paramètre
6. *get* Effectue une redirection
7. *doSomethingOn* Peut servir à ajouter ou supprimer des éléments dans un contenu existant
8. *compile* Ajoute la portion de code écrite vers la vue

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/etudiants/2014/beaugrand/dokumission?rev=1418891903

Last update: **2019/08/31 14:31**