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

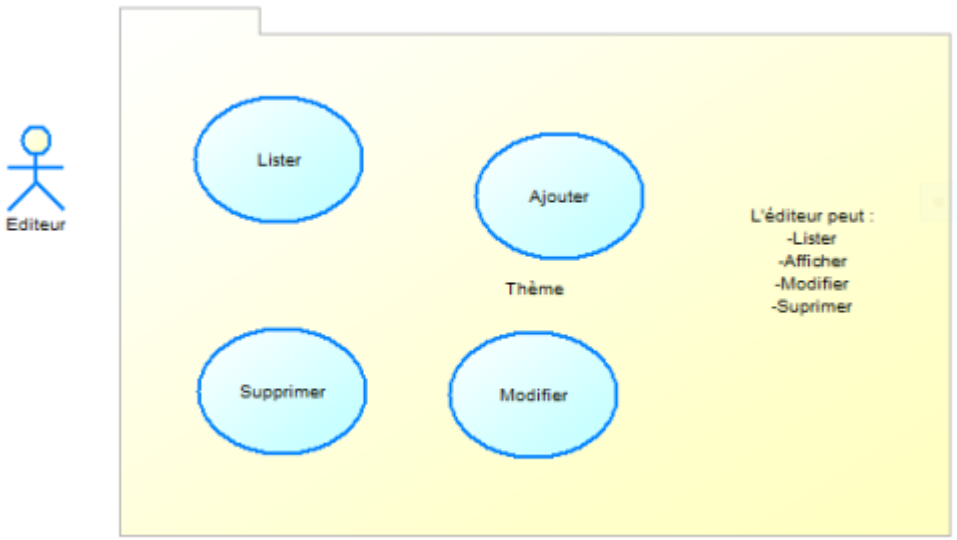
- PHP/Mysql
- Doctrine pour le mappage relationnel/Objet
- CodeIgniter pour la mise en place MVC

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 possèdent.
- Un utilisateur peut travailler dans un ou plusieurs domaines.
- Les documents sont identifier dans des thèmes.
- Un thème appartient à un domaine.
- Un thème peut posséder un thème parent.
- Un thème appartient à un utilisateur.

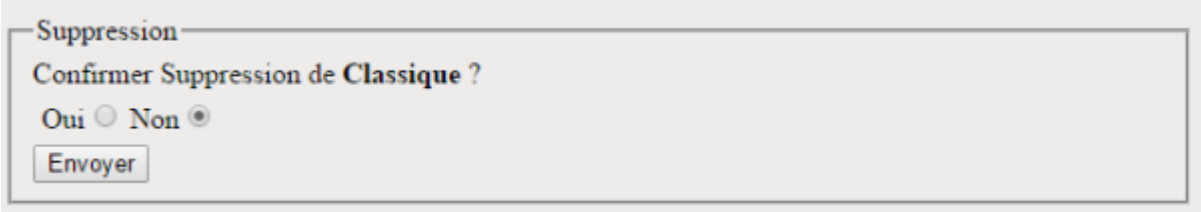
Vue

Pour la gestion des thèmes 3 vues sont utilisées:

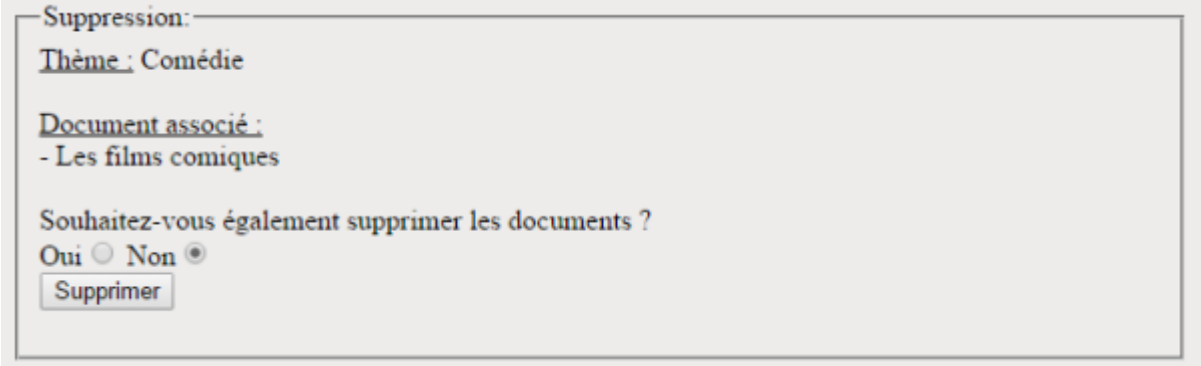
- v_theme (Vue général, liste les thèmes triés + formulaire d'ajout et de modification)



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



- v_formdeletetheme (Avant suppression, on demande s'il on souhaite garder les documents)



Controlleur

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

- index (Méthode appelée par défault “appel les vue supplémentaire tel que le menu, footer...”)
- refresh (Appelée par l'index “initialise les outils nécessaire au premier affichage de la page (Paramêtre de la BDD/fonction JS)”)
- addParentThem/updateParentThem (Ces méthode ajoute/modifie l'ensemble des thèmes qui existe pour le domaine choisi pour les ajouter dans les listes déroulante)
- add (Ajoute un nouveau thème)
- theme_modif (Est appelé quand on veut modifier un thème, celle-ci pré-remplit le domaine, thème parent)
- update (met à jour les informations modifiées)
- deleteForm (Cette méthode verifie 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 theme)
- saveDocs (Verifi si l'utilisateur a décidé de garder ou non les documents)
- updateDocs (Dans la table document, met l'identifiant de thème à nul)
- deleteDocs (Supprime le document)

Librairie

Appelé dans le constructeur : *$this→load→library('Modelutils');* Utilisation de la librairie **ModelUtils**, dans cette classe 2 modéles ont étaient crées.

- cleanPost (nettoie les variables)
- ifempty (Recoit 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;
    }
```

```
/**
 * 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;
}
```

Requête doctrine utilisées

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

Pour un **Select** :

1. Appel de *createQuery* (création de la requête)
2. *getResult* (Récuération multiple)
3. *getSingleResult* (Récupérer une seul valeur)

Pour un **Update**:

1. Appel de *createQuery*
2. *execute* (Execute la requête de mise à jour)

Pour une **insertion**:

1. *Persist* (Prépare l'instance à être insérée)
2. *flush* (Ajoute l'instance dans la base)

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 envoyé automatiquement)
3. *click* Lors d'un clicque sur un élément on appel une autre méthode
4. *show* Affiche l'élément mis en paramêtre
5. *hide* Cache l'élément mis en paramêtre
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=1418736979

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