Docu-Pro

Contexte

Etablissement

Une grande surface spécialisée dans différents secteurs (notion de Monde).

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

- Développement sous Eclipse.
- Utilisation de doctrine + DAO
- Utilisation du framework CodeIgniter.

Langages

- PHP
- MySQL

MVC

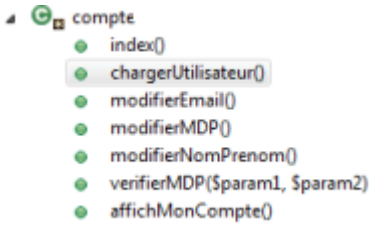
Modèles

Le seul modèle qui est utilisé est « utilisateur » car la gestion de son compte est indépendant a l'utilisateur.



Contrôleurs

Il existe également qu'un seul contrôleur « compte » qui permet de gérer son compte.



Pour le chargement de l'utilisateur je n'ai pas la fonction de connexion pour la tester. Donc j'ai mis par défaut id=3 en attendant.

Nom	Description
Index	Appel fonction affichMonCompte()
_chargerUtilisateur()	Chargement de l'utilisateur
_modifierEmail()	Modification de l'adresse Mail
_modifierMDP()	Modification du mot de passe
_modifierNomPrenom()	Modification du nom ou (et) prénom
verifierMDP()	Vérification du mot de passe
_affichMonCompte()	Affichage des informations de l'utilisateur

Vue

Nom	Description
V_header	Affichage de l'en tête de la page
v_gererMonCompte	Affiche les informations de l'utilisateur + gestion
v_footer	pied-de-page

Base de donnée

aucun changement à était fait mais quelques insertions pour les tests.

Lien
docu.sql

Exemple de code

Utilisation de Jsutils

```
/**
 * @briefs Affichage des informations de l'utilisateur
 * @details Modification des informations personalles de l'utilisateur
 */
public function _affichMonCompte(){
    $this->jsutils->click("#BtnMDP", $this->jsutils->show('#password'));
```

```
        $this->jsutils->click("#LienRetourMDP", $this->jsutils->hide('#password'));
        $this->jsutils->click("#BtnInfo", $this->jsutils->show('#info'));
        $this->jsutils->click("#LienRetourInfo", $this->jsutils->hide('#info'));
        $this->jsutils->click("#BtnEmail", $this->jsutils->show('#divEmail'));
        $this->jsutils->click("#LienRetourEmail",
$this->jsutils->hide('#divEmail'));
        $this->jsutils->postFormAndBindTo("#BtnValider1", "click",
"/wikiPro/compte/_modifierMDP","password","#MsgPasse");
        $this->jsutils->postFormAndBindTo("#BtnValider2", "click",
"/wikiPro/compte/_modifierNomPrenom","info","#MsgInfo");
        $this->jsutils->postFormAndBindTo("#BtnValider3", "click",
"/wikiPro/compte/_modifierEmail","divEmail","#MsgEmail");
        $this->jsutils->compile();
        $utilisateur = $this->_chargerUtilisateur();
        $this->load->view("v_gererMonCompte",array("user"=>$utilisateur));
    }
```

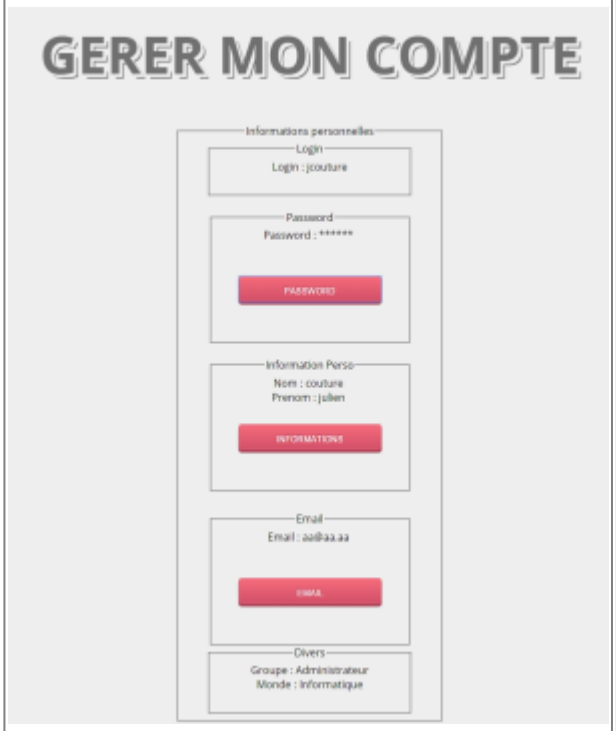
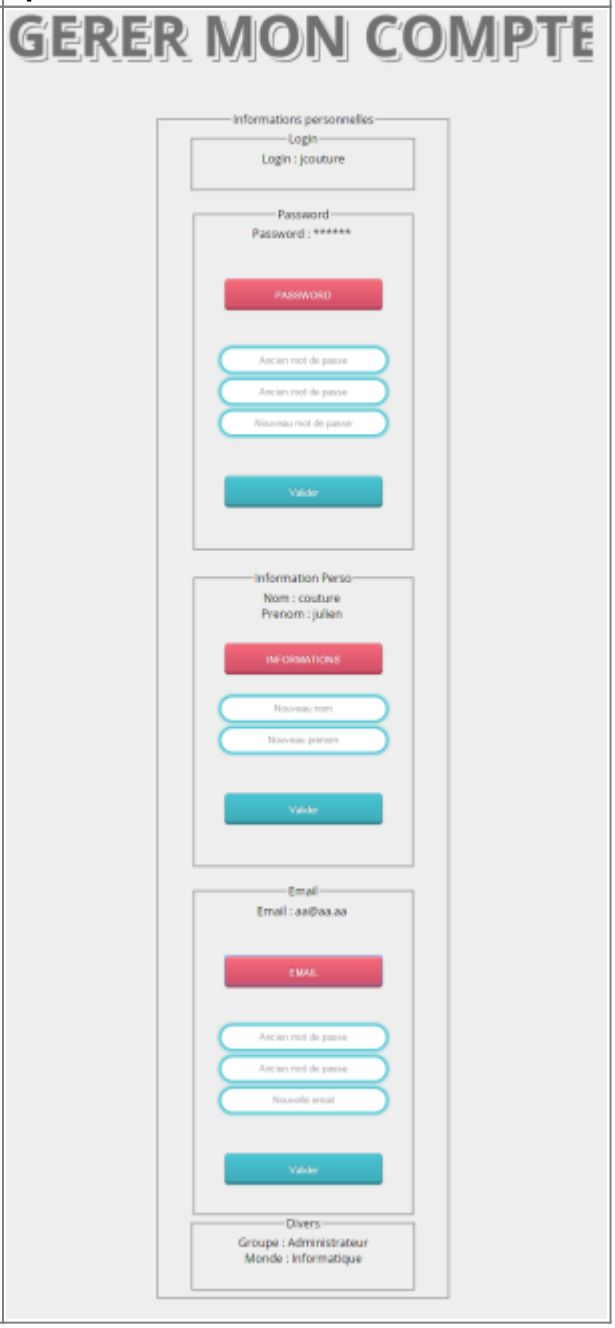
Utilisation de DAO

```
    /**
     * @brief Modification du mot de passe
     * @details    saisi de l'ancien mot de passe + nouveau mot de passe
     * @details Vérifie si l'un des champs n'est pas vide
     */
    public function _modifierMDP(){
        $Passe1 = $_POST['AncienPassword'];
        $passe2 = $_POST['AncienPassword2'];
        $newPasse = $_POST['NewPassword'];
        $user = $this->_chargerUtilisateur();
        if($Passe1 != null || $passe2 != null || $newPasse != null){
            if($this->_verifierMDP($Passe1,$passe2)){
                $user->setPassword($newPasse);
                DAO\update($user);
                echo "Modification de votre mot de passe réussi";
            }
        }
        else{
            echo "veuillez saisir les champs mots de passe";
        }
    }
```

Exemple DQL (doctrine)

```
<?php
//Création de la requête d'update
$query = $this->doctrine->em->createQuery("UPDATE utilisateur u SET u.nom ='%$nom%'
WHERE u.id=1");
//Exécution de la requête.
$search = $contenu->execute();
```

Design

Avant	Aprés
	

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

Permanent link:
http://slamwiki2.kobject.net/etudiants/2014/julien.couture/docupro?rev=1419776961

Last update: **2019/08/31 14:32**