

Le cas d'utilisation présente plusieurs actions bien distinctes dans ses parties normale ou d'extension :

1. **L'administrateur demande à se connecter**
2. Le système demande le nom de user et le mot de passe.
3. **L'internaute saisit le nom de user et le mot de passe**
4. Le système retourne la liste des catégories
5. **L'administrateur sélectionne une catégorie**
6. Le système retourne la liste des articles de la catégorie (id et description) avec pour chaque article la possibilité de modifier ou supprimer. Une option d'ajout de produit pour cette caté
7. **L'administrateur sélectionne modifier**
8. Le système retourne les infos de l'article, description et prix.
9. **L'administrateur modifie les infos et envoie le formulaire**
10. le système enregistre la modification

Extensions

- 11 **L'administrateur sélectionne supprimer**
- 12 Le système demande si l'article doit bien être supprimé
- 13 **L'administrateur confirme ou pas**
- 14 Le système supprime ou non (selon la réponse en 7.3)
- 15 **L'administrateur demande à créer un nouveau produit**
- 16 Le système retourne un formulaire de saisies
- 17 **L'administrateur remplit le formulaire et envoie le formulaire**
- 18 Le système enregistre les informations
- 19 L'administrateur demande (à tout moment) de se déconnecter
- 19.1 Le système déconnecte l'administrateur

Ainsi le switch du contrôleur contiendra 9 actions (au moins) ; les scénarios particuliers seront traités comme des cas particuliers éventuels à l'intérieur de chaque case.

1) Points 1 à 4 du cas d'utilisation

Nous allons aussi traiter le cas particulier associé noté x.1 et x.2.

Le contrôleur présente les deux actions associées :

```

1  <?php
2  if(isset($_REQUEST['action']))
3      $action = $_REQUEST['action'];
4  else
5      $action = 'connexion';
6
7  switch($action)
8  {
9      case 'connexion':
10     {
11         if(!testConnecte())
12         {
13             include('vues/v_connexion.php');
14         }
15         else
16         {
17             $lesCategories = $pdo->getLesCategories();
18             include('vues/v_deconnexion.php');
19             include('vues/v_categoriesAdmin.php');
20         }
21     }
22     break;
23     case 'validConnexion':
24     {
25         $user = $_REQUEST['user'];
26         $mdp = $_REQUEST['mdp'];
27         $n = $pdo->validerAdmin($user,$mdp);
28         if($n == 1)
29         {
30             enregAdmin();
31             $lesCategories = $pdo->getLesCategories();
32             include('vues/v_deconnexion.php');
33             include('vues/v_categoriesAdmin.php');
34         }
35         else
36         {
37             $message = "Erreur d'identifiant ou mot de passe !";
38             include('vues/v_message.php');
39             include('vues/v_connexion.php');
40         }
41     }
42     break;

```

Commentaire :

- Lignes 1 à 5

Ce test est effectué sur la page afin de "contrer" des intrusions par l'url

Le code est explicite, la fonction validerAdmin va tester en base :

```

public function validerAdmin($user,$mdp)
{
    $req="select count(*) as nb from administrateur where nom = '$user' and mdp='$mdp'";
    $res = PdoLafleur::$monPdo->query($req);
    $laLigne = $res->fetch();
    return $laLigne['nb'];
}

```

les deux fonctions enregAdmin() et testConnecte() gèrent la session de l'administrateur :

```

/**
 * Teste si un administrateur est connecté
 * crée une nouvelle entrée dans le tableau de variables sessions
 * @return vrai ou faux
 */
function enregAdmin()
{
    $_SESSION['admin'] = 1;
}
/**
 * Teste si un administrateur est connecté
 * @return vrai ou faux
 */
function estConnecte()
{
    return isset($_SESSION['admin']);
}

```

La vue `v_connexion` est un simple formulaire :

Administration

User*	<input type="text" value="titi"/>
mdp*	<input type="password" value="...."/>
<input type="button" value="Valider"/> <input type="button" value="Annuler"/>	

Le bouton `valider` doit bien sûr pointer sur l'action `validConnexion` :

```
<form method='POST' action='index.php?uc=administrer&action=validConnexion'>
```

On peut remarquer que la vue des catégories existe déjà pour un cas d'utilisation du Front Office, néanmoins la vue des catégories pour l'administrateur s'en distingue au niveau de l'UR catégorie. Il pouvait être envisagé de reprendre la vue existante et de faire un test sur l'UC. Cela n'aurait pas été une bonne solution car on maintiendrait dans cette hypothèse une adhé et le contrôleur, ce qui n'est pas l'objectif de cette architecture. Il est ainsi préférable de créer une nouvelle vue propre à cet UC :

```

<ul id="categories">
<?php
foreach( $lesCategories as $uneCategorie)
{
    $idCategorie = $uneCategorie['id'];
    $libCategorie = $uneCategorie['libelle'];
    ?>
    <li>
    <a href="index.php?uc=administrer&categorie=<?php echo $idCategorie?>&action=voirProduits"><?php echo $libCategorie ?
    </li>
<?php
}
?>
</ul>

```

1) Points 5 à 10 du cas d'utilisation

Les points 5 et 6 ne posent pas de problèmes :

```









43     case 'voirProduits':
44     {
45         if(estConnecte())
46         {
47             include('vues/v_deconnexion.php');
48             $lesCategories = $pdo->getLesCategories();
49             include("vues/v_categoriesAdmin.php");
50             $categorie = $_REQUEST['categorie'];
51             $lesProduits = $pdo->getLesProduitsDeCategorie($categorie);
52             include("vues/v_produitsAdmin.php");
53         }
54         else
55         {
56             include('vues/v_connexion.php');
57         }
58     }
59     break;
60 }

```

Le test -ligne 45- est néanmoins fait au préalable afin de sécuriser l'application. Par ailleurs la ligne 47 ajoute la vue de déconnexion, cf le point 19 du cas d'utilisation.

La vue des produits est propre à l'administrateur, il faut ajouter un lien pour créer un nouveau produit (cf cas d'utilisation) :

- [Déconnexion](#)

Composition	
Fleurs	
Plantes	
c01 Panier de fleurs variées : 52.00 Euros	 
c02 Coup de charme jaune : 38.00 Euros	 
c03 Bel arrangement de fleurs de saison : 68.00 Euros	 
c04 Coup de charme vert : 41.00 Euros	 
c05 Très beau panier de fleurs précieuses : 98.00 Euros	 
c06 Bel assemblage de fleurs précieuses : 68.00 Euros	 
c07 Présentation prestigieuse : 128.00 Euros	 
ajouter un article de cette catégorie	

Le code associé est :

```
<div id="produits">
  <?php
    foreach( $lesProduits as $unProduit)
    {
      $id = $unProduit['id'];
      $description = $unProduit['description'];
      $prix = $unProduit['prix'];
    }
  <?>
  <ul>
    <li><?php echo $id ?></li>
    <li><?php echo $description ?></li>
    <li><?php echo " : ".$prix." Euros" ?></li>
    <li><a href="index.php?uc=administrer&idProduit=<?php echo $id ?>&action=modifier">
      </li></a></li>
    <li><a href="index.php?uc=administrer&idProduit=<?php echo $id ?>&action=supprimer">
      </li></a></li>
  </ul>
  <?php
  }
  <?>
  <ul>
    <li><a href="index.php?uc=administrer&action=ajouter&categorie=<?php echo $categorie ?>">ajouter un article de cette cat
  </li>
  </ul>
</div>
```

La gestion de la modification se fait en 2 temps (cf cas d'utilisation et scénario particulier)

```

    case 'modifier' :
    {
        if(estConnecte())
        {
            $idProduit = $_REQUEST['idProduit'];
            $infosProduit = $pdo->getProduit($idProduit) ;
            $description = $infosProduit['description'];
            $prix = $infosProduit['prix'];
            include('vues/v_deconnexion.php');
            include("vues/v_modifProduit.php");
        }
        else
        {
            include('vues/v_connexion.php');
        }
        break;
    }
    case 'confirmerModifier':
    {
        if(estConnecte())
        {
            $idProduit = $_REQUEST['idProduit'];
            $description = $_REQUEST['description'];
            $prix = $_REQUEST['prix'];
            $pdo->modifierProduit($idProduit,$description,$prix);
            $message = "Produit modifié !";
            include('vues/v_message.php');
            $lesCategories = $pdo->getLesCategories();
            include('vues/v_deconnexion.php');
            include('vues/v_categoriesAdmin.php');
        }
        else
        {
            include('vues/v_connexion.php');
        }
        break;
    }
}

```

La première vue externe (action *modifierProduit*) prend la forme suivante :

Produit

description*

Plante fleurie

prix*

43.00

Valider

Annuler

Concernant l'action *confirmerModifProduit* la gestion d'erreur porte sur l'unique champ prix. La fonction modifier produit génère un ordre SQL update.

Les points 11 à 14 concernent la suppression, la demande de validation de la suppression -point 12- est gérée par du JavaScript dans la vue des produits pour l'administrateur.

```

<li><a href="index.php?uc=administrer&idProduit=<?php echo $id ?>&action=supprimer"
onclick="return confirm('Voulez-vous vraiment supprimer ce produit?');">
</a></li>

```

Un ordre delete from dans le modèle supprime, si confirmation, ce produit

Les derniers points - 15 à 18 - concernent l'ajout d'un article :

L'ajout commence par l'appel du formulaire d'ajout de produit :

```

case 'ajouter':
{
    if(estConnecte())
    {
        $categorie = $_REQUEST['categorie'];
        include('vues/v_creer_produit.php');
    }
    else
    {
        include('vues/v_connexion.php');
    }
    break;
}

```

Le formulaire *v_creer_produit* fait passer la catégorie dans un champ caché :

```

<div id="creationCommande">
<form method = "POST" action = "index.php?uc=administrer&action=confirmerAjout">
  <fieldset>
    <legend>Création d'un produit</legend>
    <p>
      <label for= "description">Description*</label>
      <input id = "description" type = "text" name="description" value = "" size="30" maxlength="50">
    </p>
    <p>
      <label for = "prix">Prix*</label>
      <input id = "prix" type = "text" name="prix" value="" size="5" maxlength="5">
    </p>
    <p>
      <label for = "image">image* </label>
      <input id = "image" type= "text" name="image" value="" size="20" maxlength="20">
    </p>
    <input type = "hidden" name = "categorie" value = "<?php echo $categorie ?>">
    <p>
      <input type= "submit" value = "Valider" name = "valider">
      <input type= "reset" value = "Annuler" name = "annuler">
    </p>
  </form>
</div>

```

La soumission de ce formulaire va appeler la validation de l'insertion :

```

case 'confirmerAjout':
{
  if(estConnecte())
  {
    $description = $_REQUEST['description'];
    $prix = $_REQUEST['prix'];
    $image = $_REQUEST['image'];
    $categorie = $_REQUEST['categorie'];
    $pdo->nouveauProduit($description , $prix , $image , $categorie);
  }
  else
  {
    include('vues/v_connexion.php');
  }
  break;
}

```

La méthode *nouveauProduit* doit ajouter le produit en créant un nouvel id :

```

157 public function nouveauProduit($description , $prix , $image , $categorie)
158 {
159   $req = "select max(id) as maxi from produit where produit.idcategorie = '$categorie'";
160   $res = PdoLafleur::$monPdo->query($req);
161   $laLigne = $res->fetch();
162   $maxi = $laLigne['maxi'];
163   $longueur = strlen($maxi);
164   $numMaxProduitActuel = substr($maxi,1,$longueur - 1);
165   $numProduitNouveau = $numMaxProduitActuel + 1;
166   if($numProduitNouveau < 10)
167   {
168     $numProduitNouveau = "0".$numProduitNouveau;
169   }
170   $idProduit = substr($maxi,0,1).$numProduitNouveau;
171   $req = "insert into produit(id, description, image,prix, idCategorie)
172   values ('$idProduit', '$description', '$image', '$prix', '$categorie')";
173   $res = PdoLafleur::$monPdo->exec($req);
174

```

Commentaires :

- La ligne 162 récupère l'actuel identifiant *maximum* pour la catégorie
- la ligne 164 extrait la valeur numérique de l'identifiant
- La ligne 165 incrémente ce maximum
- La ligne 170 "reconstruit" l'identifiant en concaténant l'initiale de la catégorie et la nouvelle valeur numérique maximum
- La ligne 168 teste et ajoute éventuellement un "0" si la valeur numérique est sur un chiffre.