

Date de réalisation : 30-04-2020

Date de modification : 08-10-2020



DOSSIER FONCTIONNEL

Gestion des actions utilisateur : AU03 Ajout d'un j'aime

Espace administrateur : EA03 Ajout d'une catégorie

Sommaire

Historique des modifications	3
Ensemble des fonctionnalités	4
Gestion des actions utilisateur	5
Introduction	5
Description du cas d'utilisation.....	5
Objectifs.....	6
Portée et limitations	6
Solution	7
Processus	7
Interface utilisateur – Page flux des questions.....	10
Espace administrateur	11
Introduction	11
Description du cas d'utilisation.....	11
Objectifs.....	12
Portée et limitations	12
Solution	13
Processus	13
Interface utilisateur – Page administration/ gestions des catégories	14
Définition des termes	16

Historiques des modifications

Date	Nom	Description de la version
30-04-2020	Lucas Boganin / laura Gonçalves	Création du document
08-10-2020	Lucas Boganin	Ajout de contenus

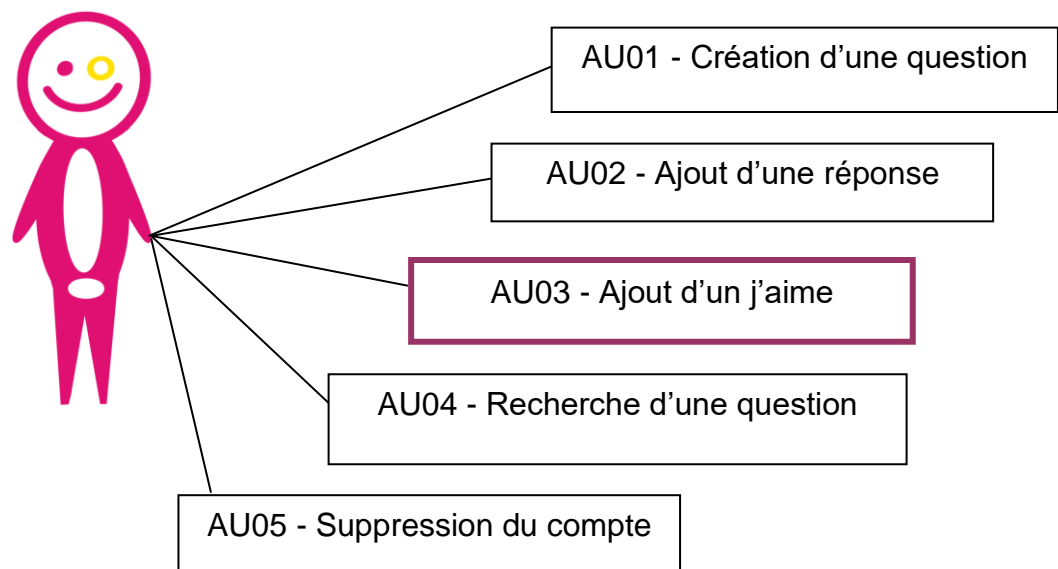
Ensemble des fonctionnalités

Fonctionnalité	Descriptif fonctionnalité	Ressource
AS - Gestion des accès du site	Inscription : l'utilisateur va pouvoir créer un compte	Laura
	Authentification : l'utilisateur va pouvoir s'authentifier et se connecter au site s'il possède un compte	
	Déconnexion : L'utilisateur se déconnecte du site	
AU - Gestion des actions utilisateur	Création d'une question : l'utilisateur peut créer une question publique ou privée	Lucas
	Ajout d'une réponse : l'utilisateur peut répondre à une question	
	Ajout d'un j'aime	
	Recherche d'une question	
	Suppression du compte	Laura
EA - Espace administrateur	Suppression d'une question	Lilian
	Suppression d'une catégorie	Lucas
	Ajout d'une catégorie	
EPU - Espace profil de l'utilisateur	Modification du profil	Lilian
	Ajout d'un avatar	
A - Gestion des amis	Envoyer une demande	Lucas
	Supprimer une demande	
	Accepter une demande	
	Suppression d'un ami	

Gestion des actions utilisateur

Introduction

Description du cas d'utilisation



L'ajout d'un j'aime un droit accessible à l'ensemble des utilisateurs du site.
L'ajout d'un j'aime fait partie de la fonctionnalité Gestion des actions utilisateur (AU).

Objectifs

Permettre à l'utilisateur de pouvoir aimé ou non la question d'un autre utilisateur.

Performance : le traitement pour effectuer le j'aime doit être transparent pour l'utilisateur, le changement d'état doit se faire instantanément.

Simplicité : un bouton simple au design commun

Portée et limitations

La solution décrite dans le présent dossier doit répondre aux objectifs et besoins mentionnés ci-dessus. Pour des raisons de délai de réalisation, il a été exclu pour le moment de permettre la possibilité de dislike une question par l'utilisateur.

Solution

Processus

Affichage d'un bouton j'aime d'une question

```
// fonction like en liens
$leVote = 0;
$couleurOn = " ";
$nombreVote = calculeVoteQuestion($laQuestion['Id_question']);
$leVote = recupVoteQuestion($laQuestion['Id_question'],$SESSION['utilisateur']['id']);
if(isset($leVote) && !empty($leVote)){
    $leVote = $leVote[0]['Action_vote'];
    $couleurOn = "fas fa-heart like-on";
    $leVote = $leVote + 1;
}else{
    $leVote = 1;
}
```

Fonction qui récupère le nombre de vote d'une question dans la BDD.

Fonction qui récupère le vote de l'utilisateur sur une question dans la BDD.

Si l'on récupère des données sur le vote de l'utilisateur sur une question via la fonction précédente, alors on assignera des valeurs pour permettre l'affichage actif du bouton j'aime.

Sinon on assignera des valeurs pour permettre l'affichage non-actif du bouton j'aime.

```
<button type="button" class="btn-like" href = "./traitement/like-fonction.php?vote=<?php echo $leVote?>
&amp;id_question=<?php echo $question['Id_question']?>&amp;nbe=<?php echo $nombreVote[0]['COUNT(*)']?>">
  <i class="<?php
    if($leVote === 1){
      $couleurOn = "far fa-heart";
    }
    echo $couleurOn;
  ?>"></i>
</button>
<span><?php echo $nombreVote[0]['COUNT(*)']?></span>
```

Le bouton j'aime lors du chargement de la page va charger les données issues des fonctions ci-dessus (capture précédente) et afficher l'état du bouton et le nombre de j'aime de la question.

Script JavaScript « requêtes ajax »

1) Lorsque qu'un bouton j'aime est actionné, le script exécute une fonction javascript qui va effectuer une requête ajax.

```
function($ :jQuery | HTMLElement ){  
    //fonction bouton like/unlike(requete ajax)  
    $('.affichage-question').on('click', '.btn-like', function(e){  
        e.preventDefault()  
        var link = this  
        var parent = $(link).parent(".footer-question")  
        var httpRequest = new XMLHttpRequest()  
        httpRequest.onreadystatechange = function(){  
            if(httpRequest.readyState === 4){  
                if(httpRequest.status === 200){  
                    $(parent).replaceWith(httpRequest.responseText)  
                }else{  
                    alert('impossible de contacter le serveur')  
                }  
            }  
        }  
        httpRequest.open( method: 'GET', this.getAttribute( qualifiedName: 'href'), async: true)  
        httpRequest.send()  
    });  
};
```

3) Récupère le résultat du script PHP fonction-like et remplace l'actuel bouton j'aime par le résultat du script PHP.

2) Récupération du lien du bouton et envoi de la donnée "vote" et "nbe" via la méthode « GET » au script PHP fonction-like (voir page suivante).

nbe : donnée qui indique le nombre de vote de la question

vote : donnée qui donne l'état du bouton j'aime de la question sur l'interface de l'utilisateur (actionner/non-actionner).

Vote = un j'aime

Script PHP fonction-like

Récupération de la donnée "vote" et "nbe" via la méthode « GET ».

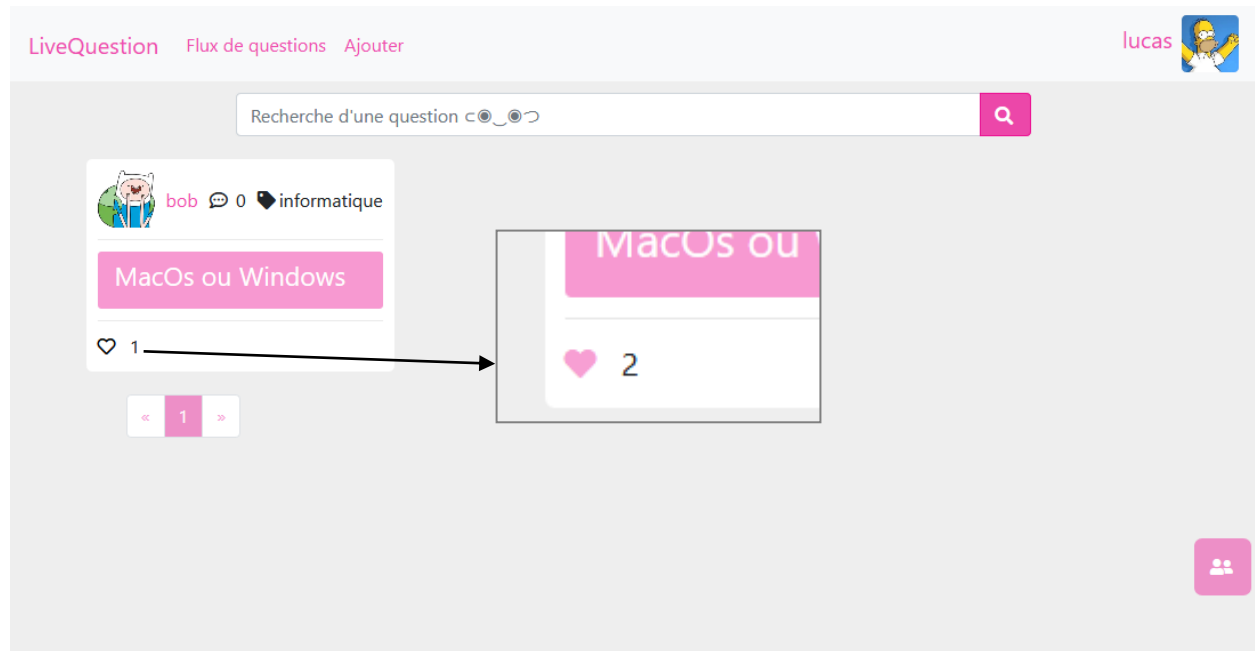
En fonction de l'état du vote le script va afficher soit la partie en bleu, soit la partie en rouge.

```
// traitement des likes et des unlikes
if(!empty($_GET['vote'])){
    //données pour le vote
    $id = $_SESSION['utilisateur']['id'];
    $vote = $_GET['vote'];
    $idQuestion = $_GET['id_question'];
    $donneeVote = [
        'id_profil' => $id,
        'vote' => $vote,
        'id_question' => $idQuestion,
    ];
    $nombreVote = $_GET['nbe'];
    switch($vote){
        case 1:
            ajoutVote($donneeVote);
            ?>
            <div class="footer-question">
                <button type="button" class="btn-like like-on" href = "../traitement/like-fonction.php?vote=<?php echo $vote + 1?>
                &amp;id_question=<?php echo $idQuestion?>&amp;nbe=<?php echo $nombreVote + 1?>">
                    <i class="fas fa-heart"></i>
                </button>
                <span><?php echo $nombreVote + 1?></span>
            </div>
            <?php
            break;
        case 2:
            suppVote($donneeVote);
            ?>
            <div class="footer-question">
                <button type="button" class="btn-like" href = "../traitement/like-fonction.php?vote=<?php echo $vote - 1?>
                &amp;id_question=<?php echo $idQuestion?>&amp;nbe=<?php echo $nombreVote - 1?>">
                    <i class="far fa-heart"></i>
                </button>
                <span><?php echo $nombreVote - 1?></span>
            </div>
            <?php
            break;
    }
}
```

Si L'utilisateur à actionner le bouton j'aime, le script va afficher ce résultat et exécuter une fonction qui va ajouter le vote de l'utilisateur sur la question dans la BDD.

Si L'utilisateur à dé actionner le bouton j'aime, le script va afficher ce résultat et exécuter une fonction qui va supprimer le vote de l'utilisateur sur la question dans la BDD.

Interface utilisateur



Événement	Action
Chargement de la page	Au chargement de la page, affiche de la page flux de questions, qui affiche l'ensemble des questions publique et privée.
Lorsque l'utilisateur clique sur le bouton j'aime (On)	Le logo du bouton passe au rose, et la valeur s'incrémente.
Lorsque l'utilisateur clique sur le bouton j'aime (Off)	Le logo du bouton passe à la couleur d'origine, et la valeur décrémente.

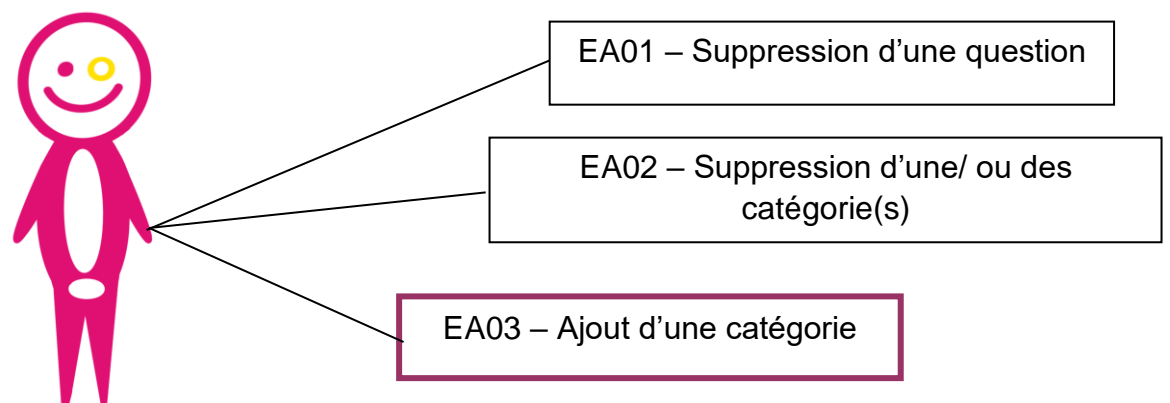
On : l'utilisateur n'avait pas encore utiliser le bouton j'aime donc il l'actionne.

Off : l'utilisateur avait déjà cliquer sur le bouton j'aime donc il le dé actionne.

Espace administrateur

Introduction

Description du cas d'utilisation



L’ajout d’une catégorie est un droit donné uniquement aux administrateurs du site. L’ajout d’une catégorie fait partie de la fonctionnalité Espace administrateur (EA).

Objectifs

Permettre à l'administrateur d'ajouter une catégorie pour augmenter l'éventail de catégories du site pour l'ensemble des utilisateurs.

Simplicité : interface simple qui permet à l'administrateur d'ajouter une catégorie et afficher un message d'erreur quand celui-ci fait une erreur.

Portée et limitations

La solution décrite dans le présent dossier doit répondre aux objectifs et besoins mentionnés ci-dessus. Pour des raisons de délai de réalisation, il a été exclu pour le moment de permettre la possibilité d'ajouter plusieurs catégories par traitement.

Solution

Processus

Affichage du formulaire d'ajout d'une catégorie

Le formulaire utilisera la méthode « POST » pour le traitement des données

Vérification de la validité du champ et si la catégorie saisie existe déjà, si la catégorie est existante un message d'erreur s'affichera : « catégorie déjà existante »

```
<div class="form-group col-md-4">
  <label for="une_categorie">Insérer une nouvelle catégorie:</label>
  <input type="text" class="form-control" id="une_categorie" name="une_categorie">
  <span class="erreur">
    <?php
      if(!empty($_POST['une_categorie']) && rechercheCateg($_POST['une_categorie']) === True){
        echo 'catégorie déjà existante';
      }
    ?>
  </span>
</div>
```


Fonction qui va établir une recherche de la catégorie saisie, et retourner vrai s'il trouve notre catégorie, ou soit faux s'il ne la trouve pas

```
if(!empty($_POST['une_categorie']) && rechercheCateg($_POST['une_categorie']) !== True){
  ajoutCateg($_POST['une_categorie']);
}
```

Fonction qui va ajouter la catégorie saisie dans la BDD

Vérification, si la catégorie saisie n'existe pas dans la BDD, et que le champ est valide, alors la catégorie sera ajoutée à la BDD

Interface utilisateur

LiveQuestion Flux de questions Ajouter lucas 

Gestion des catégories

Insérer une nouvelle catégorie:

Supprimer des catégories: aucune options selectionné

aucune options selectionné

informatique

cinema

jeux video

star wars

Gestion des catégories

Insérer une nouvelle catégorie:

Supprimer des catégories: aucune options selectionné

Test avec une catégorie déjà existante dans la BDD

Gestion des catégories

Insérer une nouvelle catégorie:

Supprimer des catégories: aucune options selectionné

catégorie déjà existante

Résultat après avoir soumis la saisie, affichage d'un message d'erreur

Gestion des catégories

Insérer une nouvelle catégorie:

Supprimer des catégories:

aucune options selectionné ▼ valider

Test avec une catégorie qui n'est pas existante dans la BDD

Gestion des catégories

Insérer une nouvelle catégorie:

Supprimer des catégories:

aucune options selectionné ▼ valider

- informatique
- cinema
- jeux video
- star wars
- ordinateur**

Résultat après avoir soumis la saisie, aucun message d'erreur et ajout de la catégorie saisie.

Evénement	Action
Chargement de la page	Au chargement de la page, affichage du formulaire d'ajout et de suppression de catégorie
Saisie d'une catégorie par un administrateur	Les vérifications suivantes sont effectuées : <ol style="list-style-type: none"> 1) Validité du champ. 2) Existence de la "catégorie" saisie, si oui affiche un message d'erreur : « catégorie déjà existante ». 3) Non-existence de la "catégorie" saisie, si oui ajoute la catégorie dans la BDD.

Définition des termes

Terme	Définition
Utilisateur	Membre ou administrateur du site.
Membre	Utilisateur qui possède un accès au site mais n'ayant aucuns droits d'administration.
Administrateur	Utilisateur qui possède un accès au site mais possédant des droits sur l'administration du site (accès espace administrateur).
BDD	Base de données.
Méthode «GET»	Passage d'une ou/plusieurs donnée(s) via l'URL.
Méthode «POST »	Les données du formulaire n'apparaissent pas dans l'URL.