

Date de réalisation : 30-04-2020

Date de modification : 06-12-2020



DOSSIER FONCTIONNELLE ET TECHNIQUE

Historiques des modifications

Date	Nom	Description de la version
30-04-2020	Lucas Boganin / Laura Gonçalves	Création du document
08-10-2020	Lucas Boganin / Laura Gonçalves	Ajout de contenus
06-12-2020	Lucas Boganin / Laura Gonçalves	Ajout et modification de contenus

Contexte

Développement d'un réseau social : Live Question.

Ce site doit permettre l'inscription ou la connexion.

Il doit aussi permettre à l'utilisateur de poser des questions et de répondre à celle des autres utilisateurs, mais aussi de se désinscrire gérer son profil ou encore d'aimer les questions ou réponses des autres utilisateurs.

Enfin le site se doit de contenir un espace administrateur qui peut supprimer une question ou une réponse.

Sommaire

Spécification fonctionnelle	4
Les données	4
Dictionnaire de données	4
Model entités relation	6
Schéma relationnelle	7
Script SQL	9
Les fonctionnalités	12
Backlog	12
L'arborescence	13
Les cas d'utilisation	14
Les maquettes	15
Spécification technique	20
Gestion des actions utilisateur - AU03 Ajout d'un j'aime	20
Description du cas d'utilisation	20
Objectifs	21
Portée et limitations	21
Solution	22
Processus	22
Interface utilisateur – Page flux des questions	25
Espace administrateur - EA03 Ajout d'une catégorie	26
Description du cas d'utilisation	26
Objectifs	27
Portée et limitations	27
Solution	28
Processus	28
Interface utilisateur – Page administration/ gestions des catégories	29
Définition des termes	31

Autoformation.....	32
<i>Résumé</i>	32
<i>Usage</i>	33
Conduite de projet	34
<i>Répartition des tâches</i>	34
<i>Compte rendu de réunion</i>	35

Spécification fonctionnelle

Les données

Dictionnaire de données

NOM	CODE	TYPE	COMMENTAIRE
id_profil	Identifiant du profil	Int	Auto-incrémente
Pseudo_profil	Pseudonyme du profil	Varchar	
Mail_profil	Adresse mail du profil	Varchar	
motDePasse_profil	Mot de passe du profil	Varchar	
genre_profil	Le sexe du profil	Varchar	
avatar	Icône du profil	Varchar	
id_role	Identifiant du rôle	Int	Auto-incrémente
libelle_role	Représente le rôle : administrateur ou utilisateur	Varchar	
Id_categorie	Identifiant de la catégorie	Int	Auto-incrémente
libelle_categorie	Représente le nom de la catégorie	Varchar	
Id_question	Identifiant de la question	int	Auto-incrémente
titre_question	Titre de la question	Varchar	
Date_creation	Date création de la question	datetime	
Visible_question	Visibilité de la question : public ou privée	Varchar	
Id_reponse	Identifiant de la réponse	Int	Auto-incrémente
Contenu_reponse	Contenu de la réponse	Varchar	
Date_reponse	Date création de la réponse	datetime	

is_pending	Etat de la demande d'amis : 0 en cours et 1 accepter	bool	
Id_vote	Identifiant du vote	Int	Auto-incrémente
Action_vote	Valeur du vote	Int	

Model entité relation

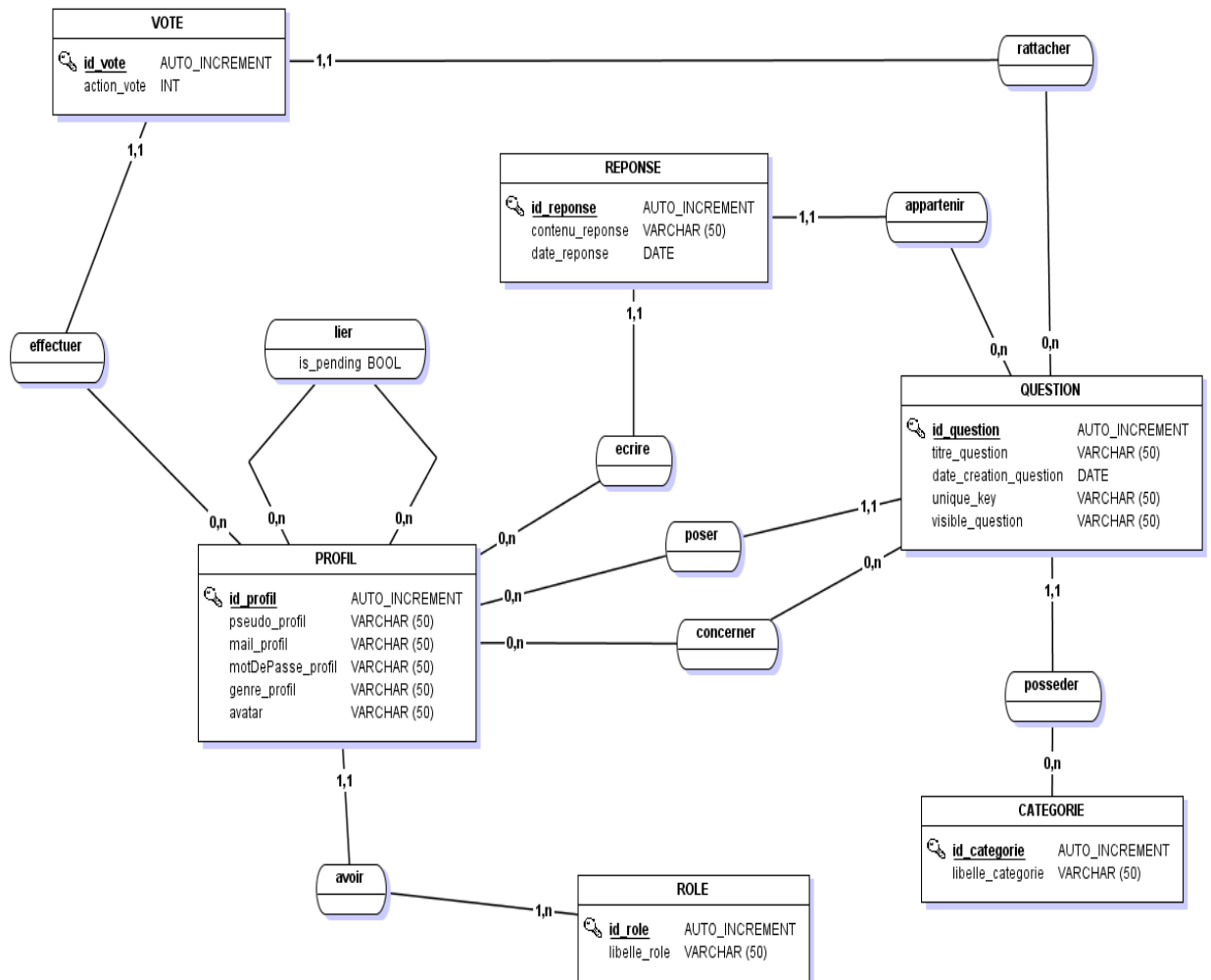


Schéma relationnel

1. Table lier

AMIS (id_profil, id_profil_liier, is_pending)

Clé primaire : id_profil, id_profil_liier

Clé étrangère : id_profil en référence à la table profil

2. Table catégorie

CATEGORIE (id_categorie, libelle_categorie)

Clé primaire : id_categorie

3. Table profil

PROFIL(id_profil, pseudo_profil, mail_profil, motDePasse_Profil, genre_profil, avatar, id_role)

Clé primaire : id_profil

Clé étrangère : id_role en référence à la table rôle

4. Table rôle

ROLE(id_role, libelle_role)

Clé primaire: id_role

5. Table question

QUESTION(id_question, titre_question, date_creation_question, visible_key, id_categorie, id_profil)

Clé primaire : id_question

Clé étrangère : id_categorie en référence à la table catégorie
id_profil en référence à la table profil

6. Table réponse

REPONSE(id_reponse, contenu_reponse, date_reponse, id_question, id_profil)

Clé primaire : id_reponse

Clé étrangère : id_question en référence à la table question
id_profil en référence à la table profil

7. Table vote

VOTE(id_vote, action_vote, id_question, id_profil)

Clé primaire: id_vote

Clé étrangère : id_question en référence à la table question
id_profil en référence à la table profil

8. Table concern

CONCERNER(id_profl,id_question)

Clé primaire: id_profl, id_question

Clé étrangère : id_profil en référence à la table profil

Id_question en référence à la table question

Script SQL

```
CREATE TABLE CATEGORIE(  
    id_categorie    Int Auto_increment NOT NULL ,  
    libelle_categorie Varchar (50) NOT NULL  
    ,CONSTRAINT CATEGORIE_PK PRIMARY KEY (id_categorie)  
)ENGINE=InnoDB;  
  
CREATE TABLE ROLE(  
    id_role    Int Auto_increment NOT NULL ,  
    libelle_role Varchar (50) NOT NULL  
    ,CONSTRAINT ROLE_PK PRIMARY KEY (id_role)  
)ENGINE=InnoDB;  
  
CREATE TABLE PROFIL(  
    id_profil    Int Auto_increment NOT NULL ,  
    pseudo_profil  Varchar (50) NOT NULL ,  
    mail_profil    Varchar (50) NOT NULL ,  
    motDePasse_profil Varchar (50) NOT NULL ,  
    genre_profil    Varchar (50) NOT NULL ,  
    avatar          Varchar (50) NOT NULL ,  
    id_role          Int NOT NULL  
    ,CONSTRAINT PROFIL_PK PRIMARY KEY (id_profil)  
    ,CONSTRAINT PROFIL_ROLE_FK FOREIGN KEY (id_role) REFERENCES ROLE(id_role)  
)ENGINE=InnoDB;
```

```
CREATE TABLE QUESTION(  
    id_question      Int Auto_increment NOT NULL ,  
    titre_question   Varchar (50) NOT NULL ,  
    date_creation_question Date NOT NULL ,  
    unique_key       Varchar (50) NOT NULL ,  
    visible_question Varchar (50) NOT NULL ,  
    id_categorie     Int NOT NULL ,  
    id_profil        Int NOT NULL  
        ,CONSTRAINT QUESTION_PK PRIMARY KEY (id_question)  
        ,CONSTRAINT QUESTION_CATEGORIE_FK FOREIGN KEY (id_categorie) REFERENCES  
CATEGORIE(id_categorie)  
        ,CONSTRAINT QUESTION_PROFILO_FK FOREIGN KEY (id_profil) REFERENCES PROFIL(id_profil)  
)ENGINE=InnoDB;
```

```
CREATE TABLE VOTE(  
    id_vote  Int Auto_increment NOT NULL ,  
    action_vote Int NOT NULL ,  
    id_question Int NOT NULL ,  
    id_profil  Int NOT NULL  
        ,CONSTRAINT VOTE_PK PRIMARY KEY (id_vote)  
        ,CONSTRAINT VOTE_QUESTION_FK FOREIGN KEY (id_question) REFERENCES QUESTION(id_question)  
        ,CONSTRAINT VOTE_PROFILO_FK FOREIGN KEY (id_profil) REFERENCES PROFIL(id_profil)  
)ENGINE=InnoDB;
```

```
CREATE TABLE REPONSE(  
    id_reponse  Int Auto_increment NOT NULL ,  
    contenu_reponse Varchar (50) NOT NULL ,  
    date_reponse Date NOT NULL ,  
    id_profil   Int NOT NULL ,  
    id_question Int NOT NULL  
        ,CONSTRAINT REPONSE_PK PRIMARY KEY (id_reponse)  
        ,CONSTRAINT REPONSE_PROFIL_FK FOREIGN KEY (id_profil) REFERENCES PROFIL(id_profil)  
        ,CONSTRAINT REPONSE_QUESTION0_FK FOREIGN KEY (id_question) REFERENCES  
QUESTION(id_question)  
)ENGINE=InnoDB;
```

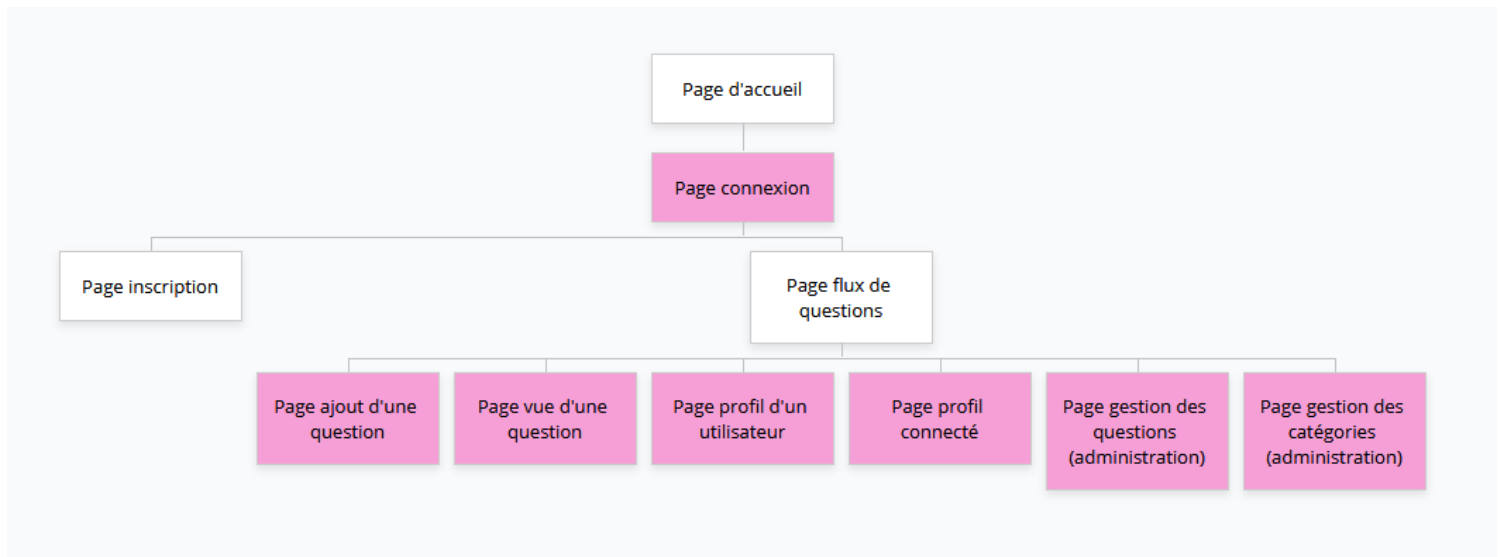
```
CREATE TABLE concernner(  
    id_profil Int NOT NULL ,  
    id_question Int NOT NULL  
    ,CONSTRAINT concernner_PK PRIMARY KEY (id_profil,id_question)  
    ,CONSTRAINT concernner_PROFIL_FK FOREIGN KEY (id_profil) REFERENCES PROFIL(id_profil)  
    ,CONSTRAINT concernner_QUESTION0_FK FOREIGN KEY (id_question) REFERENCES  
QUESTION(id_question)  
)ENGINE=InnoDB;
```

```
CREATE TABLE lier(  
    id_profil Int NOT NULL ,  
    id_profil_lier Int NOT NULL ,  
    is_pending Bool NOT NULL  
    ,CONSTRAINT lier_PK PRIMARY KEY (id_profil,id_profil_lier)  
    ,CONSTRAINT lier_PROFIL_FK FOREIGN KEY (id_profil) REFERENCES PROFIL(id_profil)  
    ,CONSTRAINT lier_PROFILO_FK FOREIGN KEY (id_profil_lier) REFERENCES PROFIL(id_profil)  
)ENGINE=InnoDB;
```

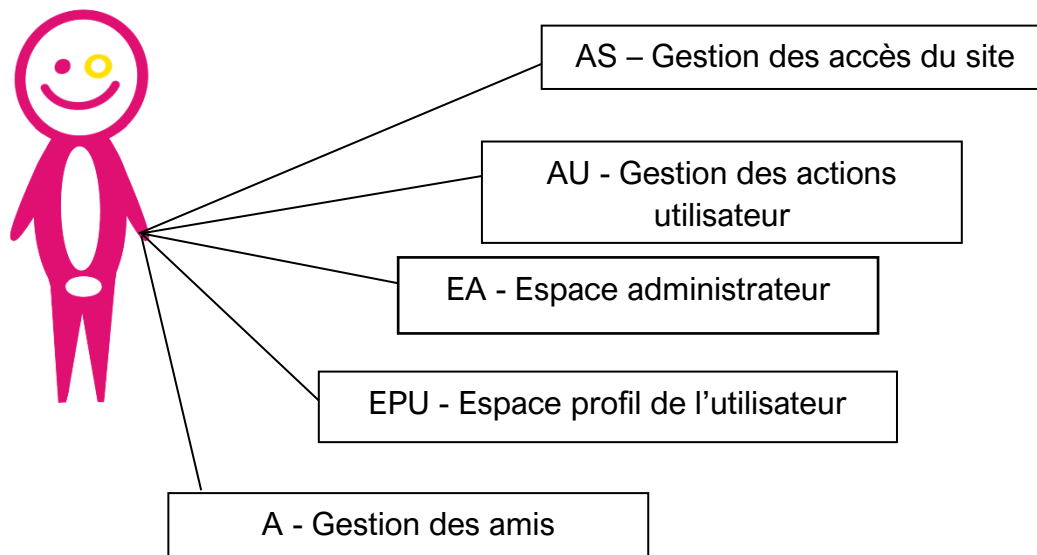
Les fonctionnalités

Backlog

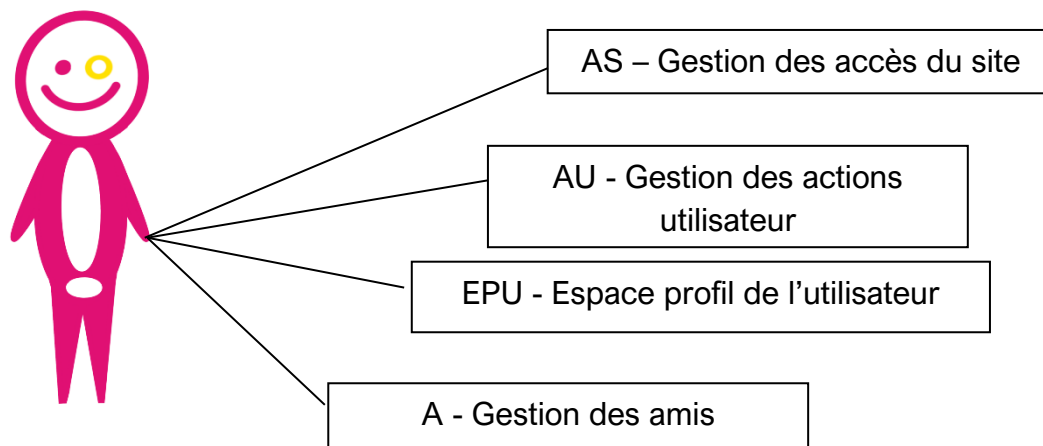
Fonctionnalité	Descriptif fonctionnalité
AS - Gestion des accès du site	Inscription : l'utilisateur va pouvoir créer un compte
	Authentification : l'utilisateur va pouvoir s'authentifier et se connecter au site s'il possède un compte
	Déconnexion : L'utilisateur pourra se déconnecter du site
AU - Gestion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



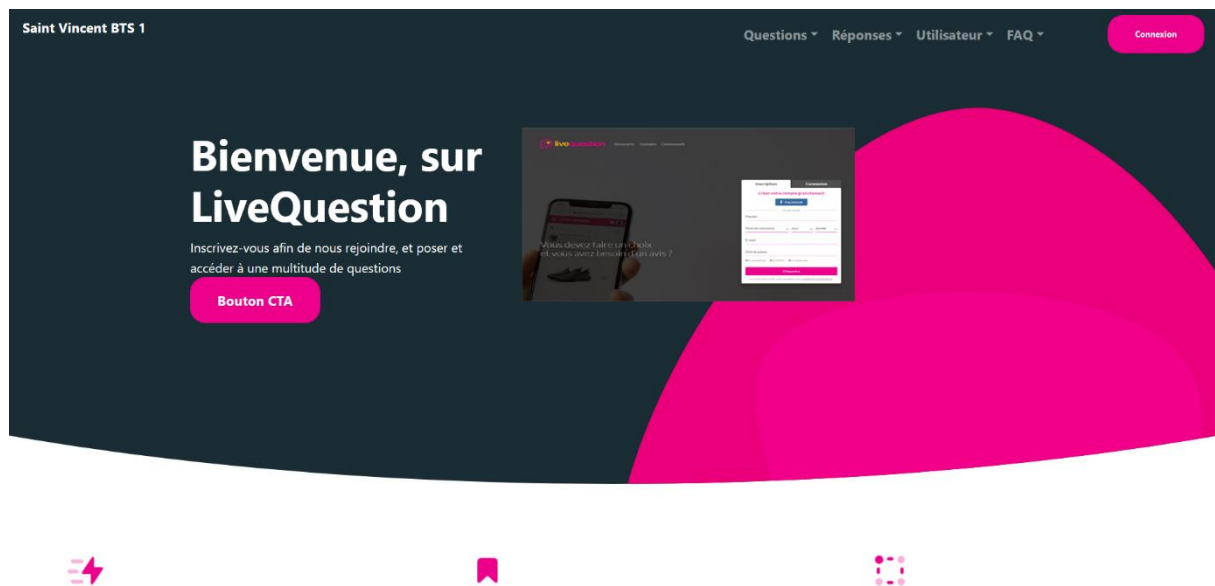
Administrateur



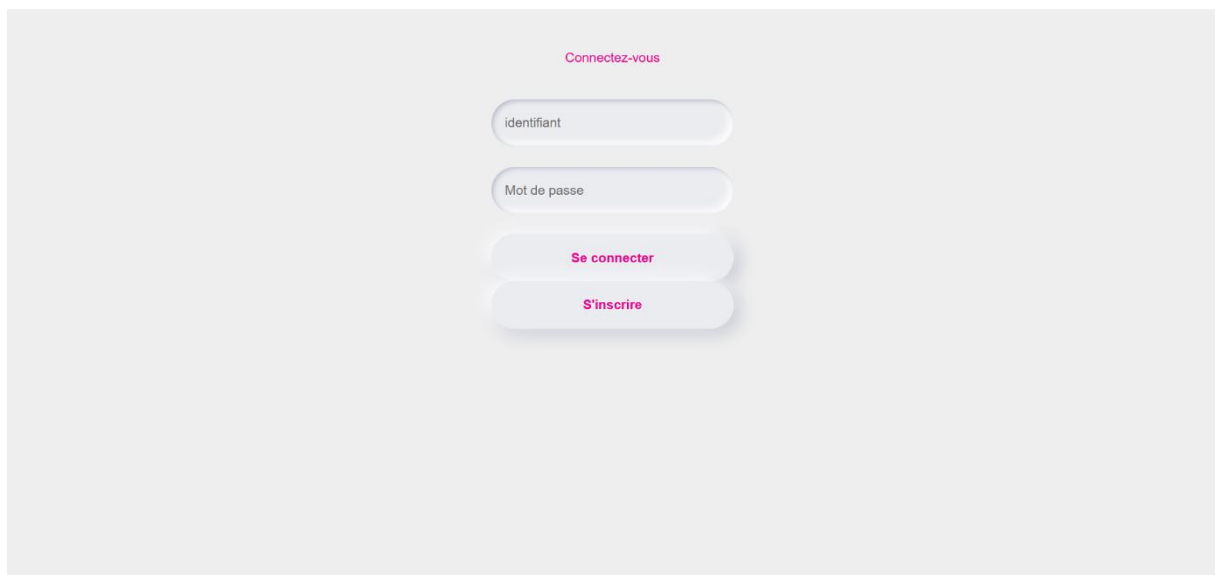
Utilisateur



Page d'accueil



Page de connexion



Page d'inscription

Inscrivez-vous

Identifiant

Adresse Mail

Sexe Femme


Mot de passe


Vérification du Mot de passe



S'inscrire

Page flux de questions

LiveQuestion Flux de questions Ajouter

lucas 


Recherche d'une question 

 bob 0  informatique

MacOs ou Windows


♥ 1

« 1 »



Page ajout d'une question

[LiveQuestion](#) [Flux de questions](#) [Ajouter](#)

lucas 

Posez une question

Quelle est votre question ?

catégorie

cinema


Qui pourra voir votre question ?


aucune options selectionné

valider

Page vue d'une question

[LiveQuestion](#) [Flux de questions](#) [Ajouter](#)

lucas 

 bob 2020-10-09 14:29:50 0

MacOs ou Windows


2


Ajouter une réponse



valider

Page d'un profil utilisateur


[LiveQuestion](#) [Flux de questions](#) [Ajouter](#)

lucas 

 bob, Homme


bob  0  informatique

MacOs ou Windows

 2

Page gestion des catégories


[LiveQuestion](#) [Flux de questions](#) [Ajouter](#)

lucas 

Gestion des catégories


Insérer une nouvelle catégorie:

Supprimer des catégories:


aucune options selectionné 


valider

Page gestion des questions

LiveQuestion Flux de questions Ajouter			lucas 
Question	Date de publication	Auteur	
MacOs ou Windows	2020-10-09 14:29:50	bob	X

Page du profil connecté

LiveQuestion Flux de questions Ajouter	lucas 
--	---



Votre avatar

Aucun fichier sélectionné.

Votre identifiant

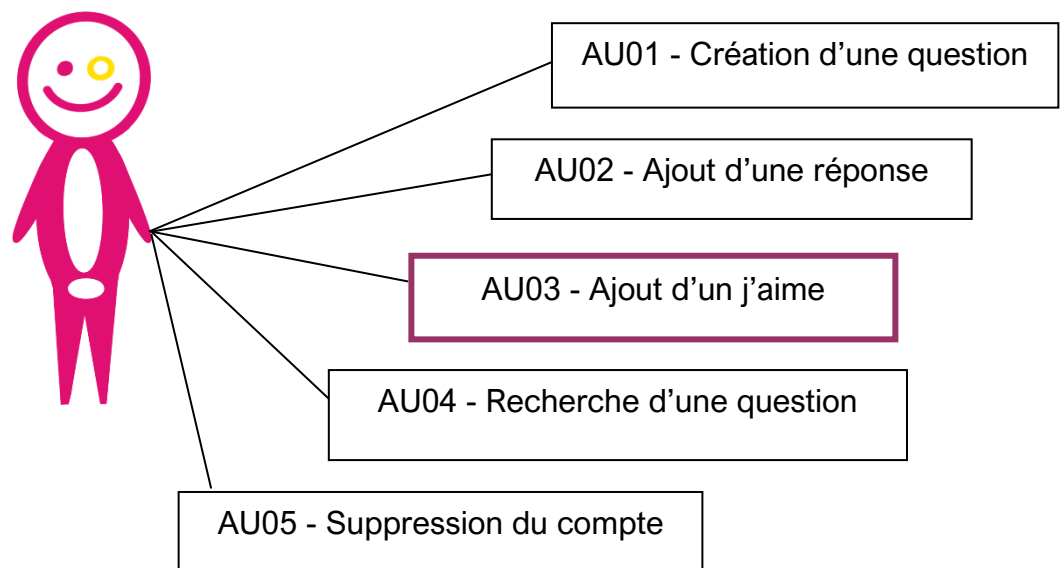
Votre adresse mail

Votre mot de passe

Spécification technique

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

Description du cas d'utilisation



L'ajout d'un j'aime est 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 aimer une 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 mettre un non-j'aime sur une question par l'utilisateur.

Solution

Processus

Affichage du 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 de la question dans la BDD.

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

Si l'on récupère des données sur le vote de l'utilisateur sur la 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 = "fas 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 « requête ajax »

1) Lorsque qu'un bouton j'aime est activé, 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 j'aime et envoie des données "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 fournit l'état du bouton j'aime sur la question (actionné/non-actionné).

vote = un j'aime

Script PHP fonction-like

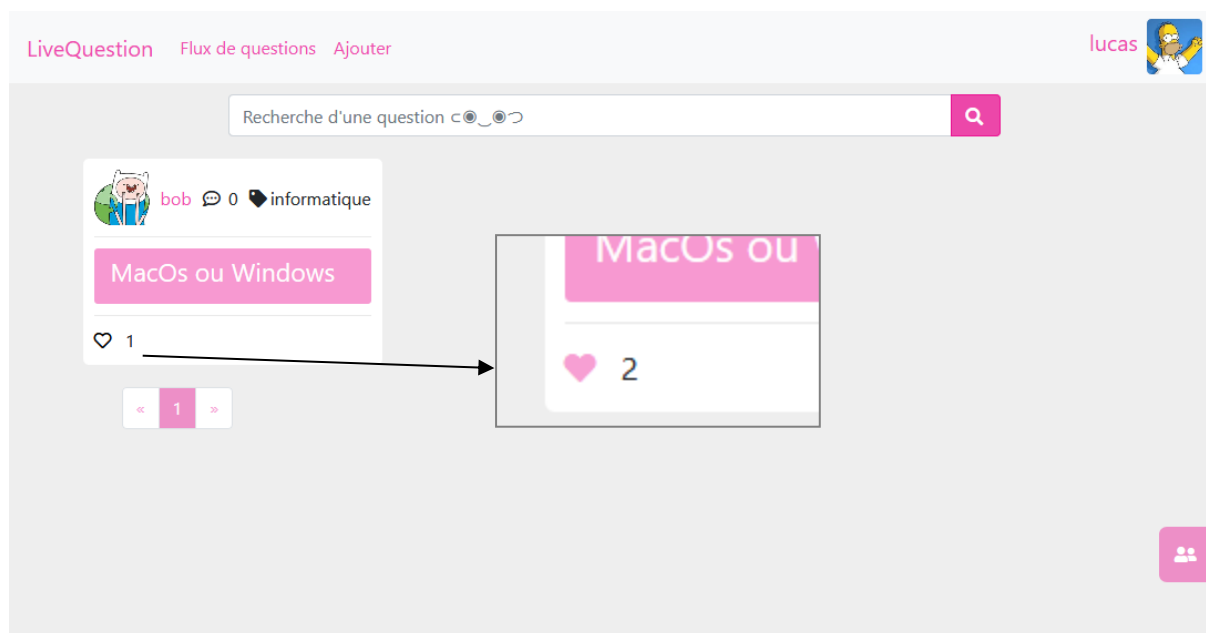
Récupération des données "vote" et "nbe" via la méthode « GET ».

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 unlike
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 a activé le bouton j'aime, le script va alors afficher ce résultat, et exécuter une fonction qui va ajouter le vote de l'utilisateur en lien avec la question dans la BDD.

Si l'utilisateur a désactivé le bouton j'aime, le script va afficher ce résultat et exécuter une fonction qui va supprimer le vote de l'utilisateur en lien avec la question dans la BDD.



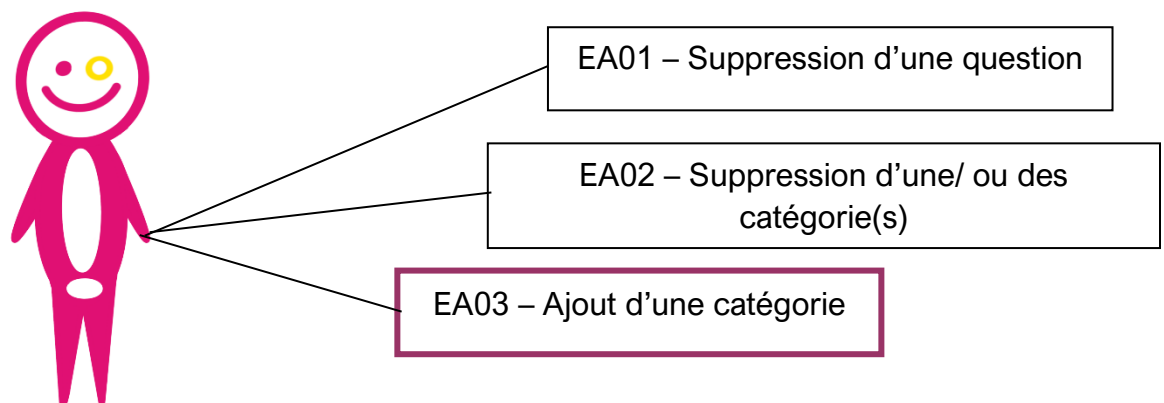
Événement	Action
Chargement de la page	Au chargement de la page flux de questions : affichage de l'ensemble des questions publiques et privées.
Lorsque l'utilisateur clique sur le bouton j'aime (On)	Le logo du bouton j'aime passe au rose, et la valeur s'incrémente.
Lorsque l'utilisateur clique sur le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

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

Off : l'utilisateur avait déjà utilisé le bouton j'aime donc il le désactive.

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 le nombre 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 aux 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 vérification de l'existence de la catégorie, si la catégorie est déjà 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, elle retournera vrai si elle trouve notre catégorie, ou retournera faux dans le cas contraire.

```
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.

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	<p>Les vérifications suivantes sont effectuées :</p> <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, ajout de 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 pas d'accès à l'espace administration
Administrateur	Utilisateur qui possède un accès au site et un accès l'espace l'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.

Résumé

AJAX (Asynchronous JavaScript and XML)

AJAX n'est pas une technologie en soi, mais un terme désignant une « nouvelle » approche utilisant un ensemble de technologies existantes, dont : **HTML** ou **XHTML**, les **feuilles de styles CSS**, **JavaScript**, le modèle objet de document (**DOM**), **XML**, **XSLT**, et l'**objet XMLHttpRequest**.

Lorsque ces technologies sont combinées dans le modèle AJAX, les applications Web sont capables de réaliser des mises à jour rapides de l'interface utilisateur sans devoir recharger la page entière du navigateur.

Les applications fonctionnent plus rapidement et sont plus réactives aux actions de l'utilisateur.

Usage

Nous avons eu recours à la méthode AJAX lors de la réalisation des fonctionnalités suivantes :

- **AU03 Ajout d'un j'aime**
- **AU04 - Recherche d'une question**

Pourquoi avoir eu recours à la méthode AJAX ?

Tout simplement car nous voulions rendre plus dynamique l'utilisation de ces fonctionnalités et permettre une transparence au niveau des traitements pour l'utilisateur, d'où le choix de la méthode **AJAX** qui utilise une technologie **asynchrone** (le programme continue à s'exécuter en attendant la réponse de la **méthode asynchrone**)

Spécification fonctionnelle

Répartition des tâches

Fonctionnalité	Ressource
AS - Gestion des accès du site 15/03/2020	Laura
AU - Gestion des actions utilisateur 30/03/2020	Lucas
	Laura
EA - Espace administrateur 15/04/2020	Lilian
	Lucas
EPU - Espace profil de l'utilisateur 30/03/2020	Lilian
A - Gestion des amis 01/05/2020	Lucas

Compte rendu de réunion

COMPTE RENDU DE REUNION DU LUNDI 30 MARS 2020
--

Étaient présents :

Madame Laura Gonçalves

Monsieur Lucas Boganin

Monsieur Lilian Jollivet

ORDRE DU JOUR

POINT 1 : MISE AU POINT SUR LA SEMAINE PASSER

1.1. Fonctionnalités finies

1.2. Problèmes rencontrés

POINT 2 : MISE AU POINT DE LA SEMAINE A VENIR

1.1 Fonctionnalités à commencer

1.1 Fonctionnalités à finir

POINT 1 : MISE AU POINT SUR LA SEMAINE PASSER

1.1. Fonctionnalités finies

Page flux de question est une tâche finie pour débiter la semaine du 30 mars. Elle s'ajoute aux tâches Page de connexion et Page d'inscription qui ont été finies pour entamer la semaine du 23 mars.

1.2. Problèmes rencontrés

Problème lié au front de la home page : le problème est lié au javascript, lorsque la navbar scroll la couleur du background et de la font devrait changer mais aucun changement ne se produit.

POINT 1 : MISE AU POINT DE LA SEMAINE A VENIR

1.1. Fonctionnalités à commencer

Page ajout d'une question et vu d'une question sont les fonctionnalités prévues pour la semaine du 30 mars 2020 avec la Home page.

1.2. Fonctionnalités à finir

Navbar de la home page doit être fonctionnelle pour la prochaine semaine