

---

# Projet web - Rapport

---

9 mai 2017

HOU Qinghua, LUO Chunmei  
WU Mengsi et ZHAO Xinru

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Vue générale du projet</b>	<b>1</b>
2.1	Concept . . . . .	1
2.2	Fonctionnalités proposées . . . . .	1
2.3	Particularités de notre site . . . . .	1
<b>3</b>	<b>Organisation du projet</b>	<b>1</b>
3.1	Répartition des tâches . . . . .	1
3.2	Organisation . . . . .	2
<b>4</b>	<b>Développement</b>	<b>2</b>
4.1	La page d'accueil . . . . .	2
4.2	Base de données . . . . .	3
4.3	Système d'authentification . . . . .	5
4.3.1	Inscription . . . . .	5
4.3.2	Chiffrement des mots de passe . . . . .	5
4.3.3	Connexion . . . . .	5
4.3.4	Injection SQL . . . . .	5
4.4	Compte administrateur . . . . .	6
4.5	Système de <i>commentaires</i> . . . . .	6
4.6	Système de <i>likes</i> . . . . .	6
4.7	Hébergement des images . . . . .	7
<b>5</b>	<b>Conclusion</b>	<b>8</b>

# 1 Introduction

## 2 Vue générale du projet

### 2.1 Concept

Tout d'abord, nous souhaitons créer un site web dédié aux jeunes femmes et leur permettant de partager leurs passions ou leurs expériences de vie, principalement en passant par des photos et une petite description.

### 2.2 Fonctionnalités proposées

Ainsi, nous créons un site où les gens, après s'être inscrit, peuvent poster des photos, y ajouter du texte de description, mettent un titre et un *tag*. D'autres utilisateurs peuvent ensuite consulter ces posts, ajouter des *likes* et poster des commentaires. De plus, ces articles sont classés par *tags*, qui sont "Animal", "Fashion", "Food" et "Tour".

Chaque semaine, les articles les plus populaires sont sélectionnés et apparaîtront sur la page d'accueil dans le slideshow.

Aussi, les utilisateurs ont chacun un profil éditable et peuvent consulter le profil des autres utilisateurs.

### 2.3 Particularités de notre site

Notre site se distingue des réseaux sociaux classiques comme Facebook ou Twitter. Nous nous centrons sur les articles postés par les utilisateurs et non sur les utilisateurs eux-mêmes car les gens qui le consultent ont pour but premier de trouver le produit adapté et non pour se faire des amis.

## 3 Organisation du projet

### 3.1 Répartition des tâches

On a divisé le projet en plusieurs fonctionnalités et chacune de nous s'occupe de quelques de ces fonctionnalités. D'abord, Xinru se charge de construire la base de données qui doit contenir 7 tables dont on a discuté en avance, du profil utilisateur et

du système de *likes*. Et puis, Mengsi s'occupe du système d'authentification (login, logout et register) et le système de commentaires. Qinghua s'occupe de l'ajout d'un nouvel article et l'hébergement des photos dans notre base de données. Et Chunmei fera tout ce qui concerne l'apparence du site pour obtenir un style homogène pour toutes les pages.

## 3.2 Organisation

En premier, nous avons choisi de nous occuper des fonctionnalités les plus importantes, c'est-à-dire de la base de données, du système d'authentification, du profil utilisateur et une page d'accueil ayant un minimum de boutons pour pouvoir accéder à la page d'inscription, de connexion et de profil utilisateur.

Puis, dans un deuxième temps, nous nous occupons de l'hébergement des photos car c'est sur cela que repose notre site. Ensuite, du système de *likes* et de *comments* et aussi du design de la page d'accueil et du site en général.

# 4 Développement

## 4.1 La page d'accueil

Notre page d'accueil se divise en 5 parties.

Tout en haut se trouve notre logo : un lapin tenant une caméra, qui représente notre site web *The Rabbit Book*. Puis, il y a une barre de menu qui contient les boutons *Home*, *Sign up* et *Log in* et lorsqu'on est connecté, ces boutons deviennent : *Home*, *Create article*, *My Profil* et *Log out*.

Comme tous nos articles sont classés par tags, on peut accéder à tous les articles correspondant à un tag via les boutons en dessous du menu.

Ensuite vient le slideshow, qui passe en diaporama les articles les plus populaires de la semaine pour chaque catégorie.

Enfin, tout en bas se trouve les 16 articles les plus récents postés par nos utilisateurs. On peut trouver y les informations comme la photo de couverture de l'article, son titre, sa description, les nombre de likes et le nombre de commentaires que possèdent cet article.

## 4.2 Base de données

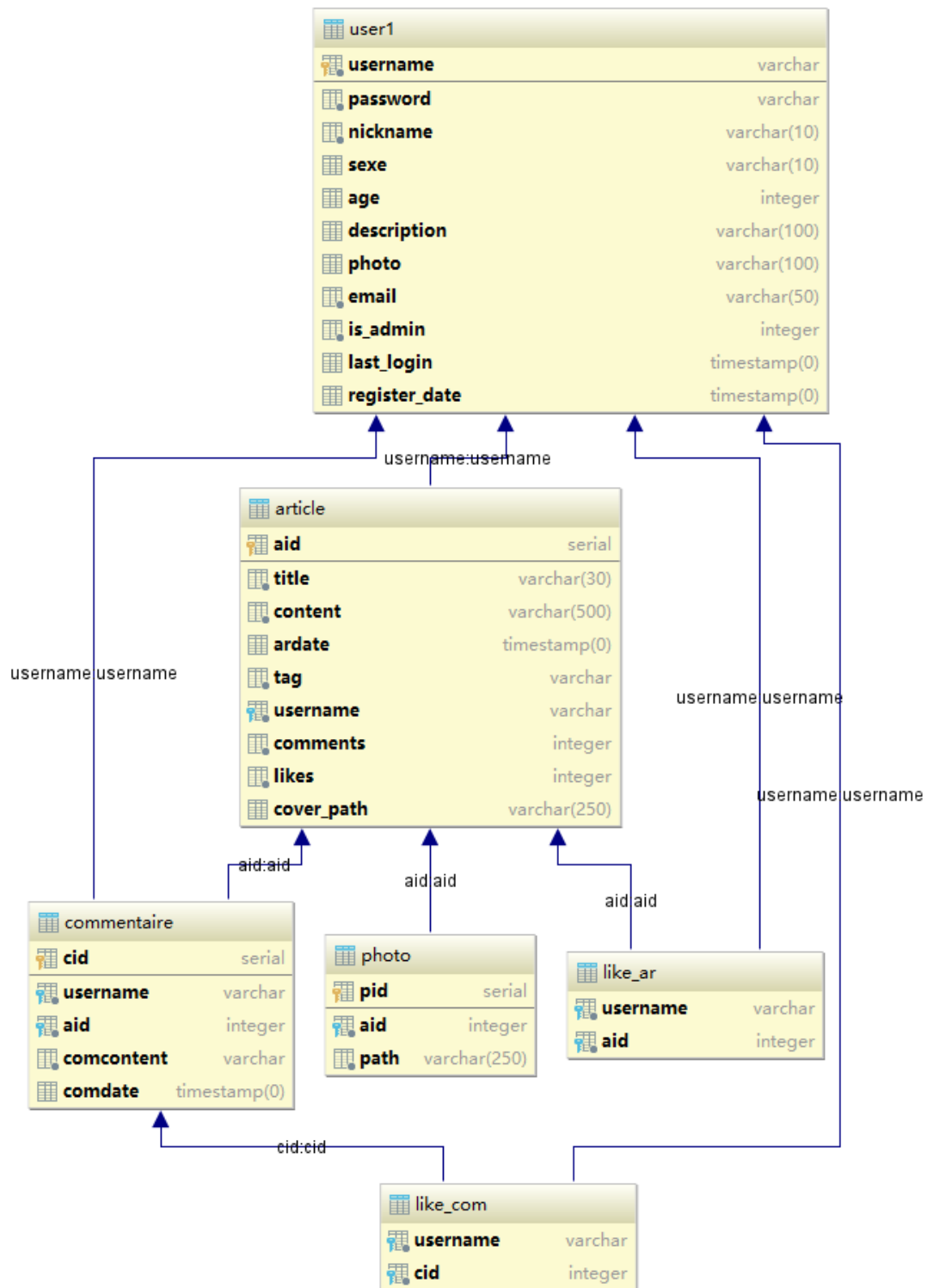
En milieu du projet, on a amélioré la table `article` et ajouté trois nouveaux attributs :

1. `likes` et `comments` pour le nombre de "likes" et "commentaires".
2. `cover_path` correspondant au chemin menant à la couverture d'un article.

Nous avons réalisé ces choix car nous pensons qu'il serait plus facile d'accéder à des informations comme le nombre de *like* que possède un article en passant par des requêtes SQL simples. Ensuite à chaque fois que l'on ajoute un nouveau *like* ou un nouveau commentaire dans la base de données, nous mettons à jour ces attributs.

La table `like_com`, correspondant à un *like* pour les commentaires n'est pas utilisée par manque de temps. Mais elle est similaire à la table `like_ar` pour les articles.

Voir le schéma ci-dessous :



Powered by yFiles

FIGURE 1: Schéma UML de la base de données

## 4.3 Système d'authentification

### 4.3.1 Inscription

L'inscription comprend le remplissage du formulaire d'inscription, la vérification de ces données en arrière-plan et la création d'un nouvel utilisateur dans la base de données.

L'utilisateur remplit d'abord le formulaire à la page `regis.html`, qui contient 4 champs : *username*, *password*, *confirm password* et *email*. Ces champs sont tous obligatoires. Pour empêcher les spams, on dispose d'un système de vérification *I'm not a robot*. Ces données sont ensuite envoyées à `addUser.php` qui va vérifier si le nom d'utilisateur et le mot de passe contiennent uniquement les caractères autorisés, c'est-à-dire majuscules, minuscules, chiffres, '-' et '\_' et si l'adresse mail donné est correct.

Ensuite ces données sont ajoutées à la base de données via la fonction `add_user($username, $password, $email)`.

### 4.3.2 Chiffrement des mots de passe

Avant d'ajouter le nouvel utilisateur dans la base de données, on va chiffrer le mot de passe. On utilise l'algorithme CRYPT\_BLOWFISH à l'aide d'une fonction de hachage en *php* `password_hash`.

### 4.3.3 Connexion

Pour se connecter à notre site, l'utilisateur doit entrer son identifiant *username* et son mot de passe *password*. Puis on compare le mot de passe entré avec celui enregistré dans la base de données avec la fonction *php* `password_verify` (puisque le mot de passe est chiffré).

### 4.3.4 Injection SQL

Pour éviter les injections SQL, c'est-à-dire l'injection de code malicieux dans les requêtes SQL via l'entrée de page web pour pirater le mot de passe des utilisateurs, on procède à une vérification des données, les caractères autres que les majuscules, minuscules, chiffres, '-' et '\_' sont interdits.

## 4.4 Compte administrateur

Nous avons créé un compte administrateur `admin` directement dans la base de données ayant un certain nombre de droits. Par exemple, lorsqu'il est connecté, un bouton *Click Me for delete* apparaît dans la page `openArticle.php` lorsqu'on ouvre un article. L'administrateur peut ensuite supprimer.

La fonctionnalité permettant de bannir un utilisateur n'a pas été réalisée par manque de temps.

## 4.5 Système de *commentaires*

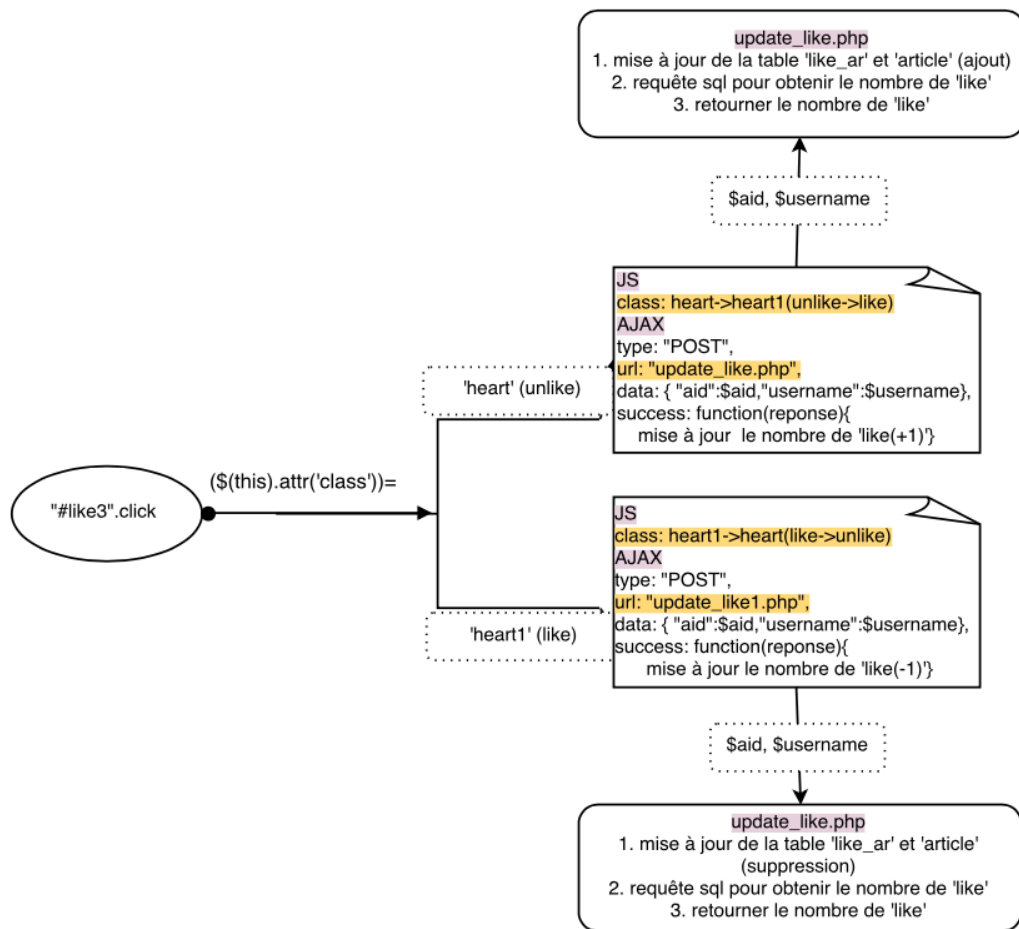
Les utilisateurs peuvent poster des commentaires pour tous les articles. Dans chaque page d'article, il y a une zone de texte dans lequel on peut entrer des commentaires. Les commentaires associés à l'article se trouvent en dessous de cette zone de texte et sont classés par ordre décroissante en fonction de la date de publication. Une fois qu'un nouveau commentaire est enregistré dans la base de données, la page de l'article est rafraîchie et le nouveau commentaire apparaîtra au dessus des anciens commentaires.

## 4.6 Système de *likes*

On utilise jQuery pour réaliser la fonction *like*.

Le schéma ci-dessous explique comment fonctionne le système de *likes* :



FIGURE 2: Schéma explicatif du système de *likes*

## 4.7 Hébergement des images

Pour sauvegarder les images postées par les utilisateurs, nous avons choisi de les enregistrer dans un dossier nommé *upload*. Ensuite dans notre base de données, nous allons sauvegarder le chemin menant à ces images.

Les images doivent vérifier une certaine condition : les types de fichiers autorisés sont `jpg`, `jpeg` et `gif` ; la taille du fichier doit être inférieure à 2000 ko et les différentes photos ne doivent pas avoir le même nom.

Ainsi, on peut utiliser le chemin de l'image pour identifier un article. C'est ce que l'on a choisi de faire pour accéder à un article de la page d'accueil.

## 5 Conclusion

Pour conclure, il y a 3 aspects que l'on peut améliorer dans notre projet. Le premier est le fait que chaque utilisateur doit pouvoir envoyer plusieurs photos pour chaque article, nous n'avons choisi pour simplifier de ne leur laisser choisir qu'une seule photo par article.

Le second est que l'utilisateur doit pouvoir rechercher les articles par mot-clé. Ça veut dire que si un utilisateur tape un mot-clé, il pourra trouver tous les articles dont le titre contient ce mot-clé.

Enfin, si un utilisateur oublie son mot de passe, il pourra le retrouver avec son adresse mail.

Ce projet nous a permis de comprendre l'importance de la collaboration dans l'équipe. On a pu partager les tâches et résoudre les problèmes ensemble. On a appris beaucoup de techniques pour le développement web et on a pu développer notre capacité de travailler indépendamment en autonomie.