Rapport de projet WEB : TrocIIE

Matthieu Pouliquen - Fatiha Ougali - Meryem Harakat - Louise Dessert

12 mai 2020

Contents

1 Introduction

Nous allons vous présenter notre projet : TrocIIE ! TrocIIE est une plateforme de troc en ligne gratuite qui permet des prêts d'objets entre les différents utilisateurs.
Ceci permettra une bonne avancé pour l'écologie ! En effet il n'est pas rare que certains objets ne nous soient utiles que occasionnellement ou bien que nous n'ayons pas besoin de les utiliser pendant une certaine période et dans ce cas il serait bien dommage de ne pas en faire profiter les autres ! Surtout dans un monde où l'on pointe souvent du doigt la surconsommation des biens et le gaspillage.
De plus certaines personnes n'ont pas les moyens d'investir dans l'achat d'objet car leur pouvoir d'achat est très réduit. Grâce à TrocIIE leur quotidien pourrait être facilité ! De l'emprunt d'outil de jardinage au livre en passant par les ustensiles de cuisines ce système d'emprunt pourrait être bénéfique pour tous !

Nous observerons en premier lieu l'architecture de notre projet
Ensuite nous allons nous penchez sur la gestions des utilisateurs, c'est à dire les membres du sites.
Puis nous nous interresserons à la gestions de objets mis en circulation pour le troc.
Enfin nous etudierons la gestion des messages

2 Répartition des rôles

Pour gagner en temps et en efficacité, la répartition des taches c'est effectuées selon des structures indépendantes qu'il a fallut lier par la suite.

La gestion des membre (User.php, UserRepository.php, connexion.php, deconnect.php, deleteUser.php, menuConnect.php, OtherUsrA.php, OtherUsrM.php, personalmodif.php, signalUser.php, statusUser.php, warnUser.php, usercnt.php, inscription.php) et la gestion des messages (Messages.php, MessageRepository.php, msgUser.php, sendMsg.php) a été attribuée à Matthieu Pouliquen.

La gestion des objet (addObjet.php, RechercheObj.php, voirObj.php, ajoutFavoris.php) ainsi que la mise en forme du site (footer.php, header.php, style.css) et l'initialisation de la table (init.sql) a été attribuée à Fatiha Ougali.

La gestion des emprunts (demandeEmprunt.php) a été attribuée à Meryem Harakat

Louise Dessert ayant eu des problèmes informatiques, elle ne pouvait pas acceder au site et tester ses fichiers. Elle c'est concentrée sur la rédaction du

rapport de la soutenance et la gestion du serveur discord et du répertoire git.

3 Présentation de l'architecture

Afin de mettre en place notre site, nous avons eu recourt à l'utilisation de différentes bases de données et établit les relations que ces dernières devaient avoir entre elles grâce à au digramme UML ci dessous
Cela nous a permit de manipuler correctement nos tables et de ne pas nous perdre en répetant des tâches déjà effectuée.

init.sql crée ces tables
User.php, Messages.php définissent des classes users et messages.
UserRepository.php, MessageRepository.php possède l'ensemble des fonctions permettant de manipuler les tables notament :
traduire
- les table en éléments de classe User/Messages,
- récupérer un Utilisateur à partir de son id,pseudo,
- récupérer un Message à partir de l'id de son recepteur/emetteur,
- supprimer une ligne d'une table,
- insérer une ligne dans une table,
- modifier une ligne d'une table selon l'id,

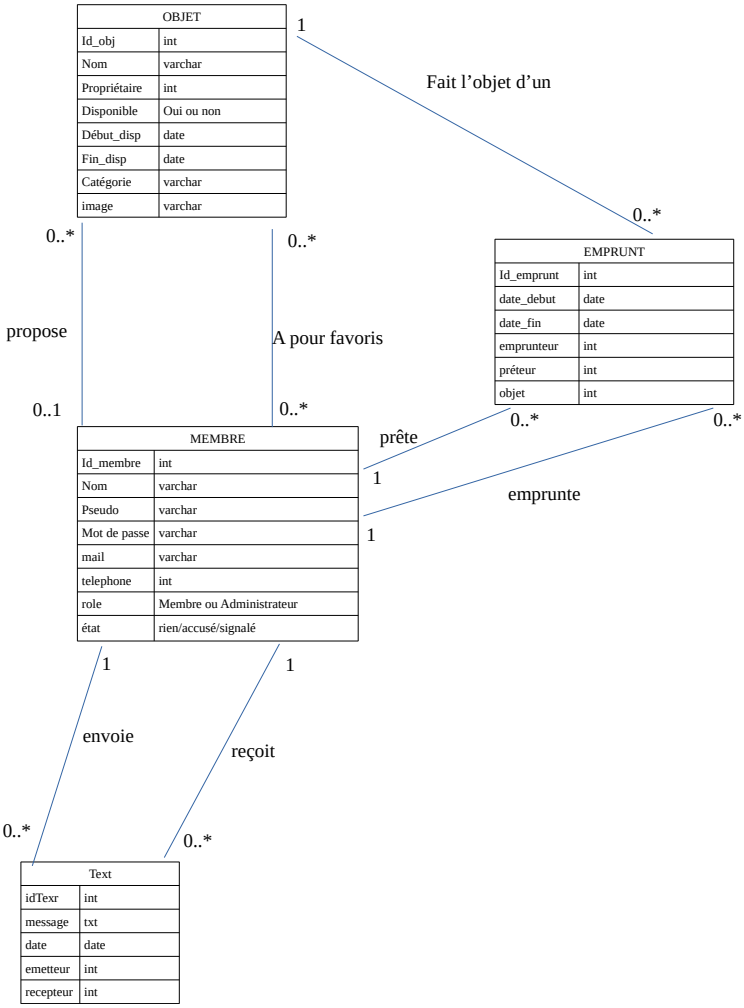

Diagramme UML

4 La gestions des différents membres

4.1 Les types de membres

Il y a plusieurs types d'utilisateur sur notre site :

- Les personnes non connectées qui sont alors considérées comme des visiteurs et dont les droit sont limités par souci de protections des données des utilisateurs.

- Les membres de TrocIIE qui sont des utilisateurs simple connectés actuellement sur le site et qui accèdent aux services proposé par le site

- Les administrateurs qui sont des utilisateurs connectés execerçant le role de modérateur, ils ont accès en plus des services normaux du site à des fonctionnalités propres à ce statut.

4.2 Les membres simples

Pour gerer les utilisateurs simples nous avons procédé de la façon suivante :

Inscription (inscription.php) : ils remplissent un formulaire avec un champ pseudo et password obligatoires, puis on parcourt la table Membre pour vérifier si pseudo n'en fait pas déjà parti. Si il n'y est pas, on ajoute dans la table membre ce pseudonyme et on redirige vers usercnt.php, on fait prendre à SESSION['id'] la valeur de l'id de la ligne ajoutée dans la base de donnée. Sinon on affiche Pseudo déjà pris .

Connexion (connexion.php) : on remplit un formulaire avec les champs pseudo et password obligatoire, puis on parcourt la table Membre pour vérifier si le pseudo et le password correspondant s'y trouvent et sont bien associés. Si il y correspondance on redirige l'utilisateur vers usercnt.php on fait prendre à SESSION['id'] la valeur de l'id de la ligne de la base de donnée correspondante. Sinon on affiche Mot de passe ou pseudo incorrect .

Modification des données personnelles : Dans usercnt.php,on peut avoir un affichage des données de l'utilisateur. Il y a un bouton modification lié au fichier personalmodif.php qui permet de modifier dans la table la ligne concernant l'utilisateur connecté.
Les modifications utilisateurs ont été séparées en 3 fonction (modif, modif-role et modif-etat), car ce sont des fonctionnalités distinctes (modifier ses informations personnelles ne devrait pas modifier son rôle, modifier son rôle ne devrait pas modifier son état,...). Ici l'utilisateur n'utilise que la première fonction.

Informations sur les autres utilisateurs : Dans OtherUsrM.php, on a un affichage de la liste des membres avec des boutons lié à warnUser.php qui modifient l'état d'un utilisateur en utilisant la fonction modif-etat de modification de la classe membre. Il est impossible pour l'utilisateur de modifier son propre état.

Suppression de compte : Il y a un bouton lié à deleteUser.php, pour supprimer son compte, cela supprime le membre.

Le menu pour les utilisateurs connectés qui permet d'utiliser ces fonctionnalités est un formulaire avec différents boutons redirigeant vers usercnt.php, msgUser.php OtherUsrA.php ou OtherUsrM.php. Il est inclut dans chacun de ses fichiers et vérifie le rôle de l'utilisateur pour le rediriger vers OtherUsrA.php ou OtherUsrM.php.

4.3 Les administrateurs

Les administrateurs profitent des mêmes fonctionalités que les utilisateurs membre, et en possède plus concernant les autres utilisateurs.

Informations sur les autres utilisateurs : Dans OtherUsrA.php , la liste des membres est affichée avec des boutons lié à signalUser.php, statusUser.php qui modifient le rôle ou l'état d'un utilisateur en utilisant modif-etat et modif-role. Il est impossible pour l'utilisateur de modifier son propre role ou son propre état.

5 La gestion des objets

Le principe fondamental de notre site est le troc d'objets. Pour cela nous avons du mettre en place différentes fonctionnalités afin de rendre système efficient. Nous avons procédé de la façon suivante :

Ajout d'un objet : L'utilisateur doit entrer dans un formulaire les informations nom de l'objet et catégorie de l'objet qui sont obligatoire . On verifie que l'utilisateur est bien connecté sinon on renvoire un message pour lui signaler qu'il doit se connecter pour effectuer cette opération. Le membre peut également ajouter une photo pour illustrer son post, et on verifiera alors que le fichier proposé est bien du format (JPG, JPEG, PNG). L'objet sera ensuite associé à l'identifiant de l'utilisateur ayant proposé ce dernier.

Rechercher un objet : Lutilisateur rempli un formulaire pour rechercher l'objet. Il peut proceder par nom, catégorie, ses dates de disponibilité. On recupère

les données du formulaire puis on recherche dans notre table objet les entrées correspondant à la recherche. S'il y en a on affiche les différents résultat. Sinon on affiche qu'aucun objet n'a été trouvé.

Voir un objet : On verifie si l'objet existe dans la table des objets et l'ont recupère les informations qui lui sont liées et également si l'objet fait partit des favoris de l'utilisateur ou s'il est emprunté. On a accès aux boutons ajouter aux favoris, faire une demande d'emprunt si l'objet n'est pas celui de l'utilisateur actuellement connecté.Si l'objet n'existe pas on le signale à l'utilisateur.

Faire une demande d'emprunt : L'utilisateur rempli un formulaire pour les dates durant lequelles il souhaite emprunter l'objet. Nous verifions que celles ci sont disponibles. Si c'est le cas on ajoute l'objet à la table emprunt avec les dates de l'emprunt, l'identifiant de l'objet et la personne demandant l'emprunt et une demande d'emprunt est envoyé au propriétaire de l'objet. Sinon , si l'utilisateur n'a pas rentré le bon format pour une date on l'en informe. Si l'objet n'est pas disponible sur les dates choisie on informe l'utilisateur.

Ajouter aux favoris : Lorsque le membre ajoute un objet à ces favoris. On verifie qu'il n'est pas propriétaire de l'objet et on ajoute l'objet à la table des favoris, en l'associant à l'utilisateur.

6 La gestions des messages

La récupération des messages ce fait par deux requête à la table Text, l'une pour récupérer les messages normaux, l'autre pour récupérer ceux envoyés à tous.

En effet, plutot que d'enregistrer plusieurs fois le même message envoyé à tous (pour 100 utilisateur, on enregistre un même message de l'administrateur vers l'utilisateur 1, de l'administrateur vers l'utilisateur 2,...) nous avons décidé d'enregistrer ce message à tous comme un unique message qui dans la table est de la forme :
id text=n, message='message' ,date msg=date, emetteur=id emetteur recepteur=id emetteur.

Ainsi pour récupérer les messages à tous il suffit de récupérer les lignes de la table où emetteur= recepteur. Par ailleurs, pour éviter qu'un membre puisse envoyer un message à tous en se l'envoyant à soi-même, un test à été ajouté dans le fichier sndMsg.php.
Il aurait également été possible de "réserver" un indice de la table membre (et empêcher toute création de membre avec cet indice) et faire la requete id text=n, message='message' ,date msg=date, emetteur=id emetteur recepteur=N avec N l'indice reservé.

Pour joindre les tables Membre et Text afin récupérer les pseudonymes plu-

tot que l'indice de l'emetteur/recepteur, nous avons effectué une jointure sur l'indice du membre entre Text et deux tables membre renomées.

Dans msgUser.php on a accès à un affichage des messages reçus et envoyés. il y a deux formulaires d'envoi de message (pour membre ou administrateur), un programme javascript affiche l'un ou l'autre selon le rôle de l'utilisateur. Les formulaires sont reliés à sendMsg.php qui ajoute les messages dans les tables s'ils sont corrects.

7 Conclusion

Nous espérons sincèrement que notre site pourra être utile et accessible à tous afin de favoriser au maximum la réutilisation des objets, le partage et même la rencontre entre les individus. Nous pensons qu'un tel système pourra vraiment être sur le long terme bénéfique à tous.

Nous avons cependant quelque pistes d'améliorations pour notre site :

Nous n'avons pas reussi à mettre en place certaines des fonctionnalités que nous souhaitions fournir au depart telle que :

- La possibilité de voir les objets sans être connecté
- Un test d'email lors de l'inscription pour vérifier que le compte n'est pas "frauduleux"
- Un système de mot de passe oublié
- Supprimer le compte d'un utilisateur signalé
- Connaitre les motifs d'un signalement et rendre public un signalement pour un administrateur
- Enlever l'accès au site aux utilisateurs lors des maintenances

L'interface ne possède pas un design très travaillé et peut donc rebuter certain utilisateurs. Ainssi nous pourrions retravailler l'aspect visuel de notre site pour le rendre plus attrayant.