

# Rapport de projet WEB : TrocIIE

Matthieu Pouliquen - Fatiha Ougali - Meryem Harakat - Louise Dessert

12 mai 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Répartition des rôles</b>	<b>2</b>
<b>3</b>	<b>Présentation de l'architecture</b>	<b>3</b>
<b>4</b>	<b>La gestion des différents membres</b>	<b>5</b>
4.1	Les types de membres . . . . .	5
4.2	Les membres simples . . . . .	5
4.3	Les administrateurs . . . . .	6
<b>5</b>	<b>La gestion des objets</b>	<b>6</b>
<b>6</b>	<b>La gestions des messages</b>	<b>8</b>
<b>7</b>	<b>Conclusion</b>	<b>9</b>

# 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 avancée 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 cuisine ce système d'emprunt pourrait être bénéfique pour tous !

Nous observerons en premier lieu l'architecture de notre projet

Ensuite nous allons nous pencher sur la gestion des utilisateurs, c'est à dire les membres du site.

Puis nous nous intéresserons à la gestion des objets mis en circulation pour le troc.

Enfin nous étudierons la gestion des messages

# 2 Répartition des rôles

Pour gagner en temps et en efficacité, la répartition des tâches s'est effectuée selon des structures indépendantes qu'il a fallut lier par la suite.

La gestion des membres (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 objets (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 accéder au site et tester ses fichiers. Elle s'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 recours à l'utilisation de différentes bases de données et établit les relations que ces dernières devaient avoir entre elles grâce au digramme UML ci dessous

Cela nous a permis de manipuler correctement nos tables et de ne pas nous perdre en répétant des tâches déjà effectuée.

init.sql crée ces tables

User.php, Messages.php définissent des classes users et messages.

UserRepository.php, MessageRepository.php possèdent l'ensemble des fonctions permettant de manipuler les tables notamment :

- traduire les tables 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 récepteur/émetteur,
- supprimer une ligne d'une table,
- insérer une ligne dans une table,
- modifier une ligne d'une table selon l'id,

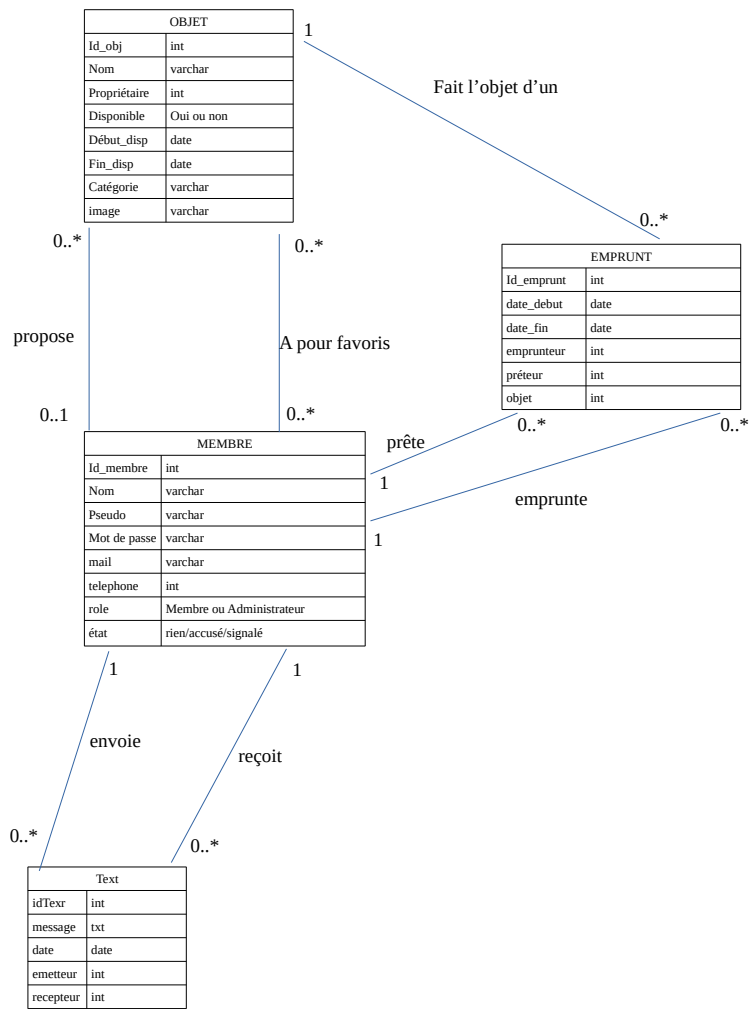


Diagramme UML

## 4 La gestion des différents membres

### 4.1 Les types de membres

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

- Les personnes non connectées qui sont alors considérées comme des visiteurs et dont les droits sont limités par souci de protection des données des utilisateurs.
- Les membres de TrocIIE qui sont des utilisateurs simples connectés actuellement sur le site et qui accèdent aux services proposés par le site
- Les administrateurs qui sont des utilisateurs connectés exerçant le rôle de modérateurs. Ils ont accès en plus des services normaux du site à des fonctionnalités propres à ce statut.

### 4.2 Les membres simples

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

Inscription (inscription.php) : ils remplissent un formulaire avec un champs pseudo et password obligatoires, puis on parcourt la table Membre pour vérifier si le 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 utilisateur ont été séparées en 3 fonctions (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.

The screenshot shows a web interface for 'TROCIE.com'. At the top, there's a navigation bar with links: 'Inscription', 'Connexion', 'Rechercher un objet', and 'Ajouter un objet'. Below this, a central box says 'Connecté en tant que chauve.' To the left, a 'Menu' section contains links for 'Informations personnelles', 'Autres utilisateurs', and 'Messages'. The main content area is titled 'Mon compte' and displays a table of user information. The table has columns: nom, Prenom, Pseudo, email, password, adresse, telephone, Role, etat, and Action. The first row shows data for 'jeremy morvan' with a role of 'Membre rien' and a 'Delete' button. Below the table are buttons for 'Modifications' and 'Deconnexion'. A copyright notice 'copyright © Ougali - 2020 All Right reserved' is at the bottom right.

nom	Prenom	Pseudo	email	password	adresse	telephone	Role	etat	Action
jeremy	morvan	chauve	pin@80 pelicano		4 rue lare	0240	Membre rien		Delete

Compte d'un utilisateur

### 4.3 Les administrateurs

Les administrateurs profitent des mêmes fonctionnalités que les utilisateurs membre, et en possèdent 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 rôle 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 obligatoires . On vérifie que l'utilisateur est bien connecté, sinon on renvoie 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 vérifiera 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 : L'utilisateur remplit un formulaire pour rechercher l'objet. Il peut procéder par nom, catégorie, ou dates de disponibilité. On récupère 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 résultats avec leurs informations. Sinon on affiche qu'aucun objet n'a été trouvé.

Voir un objet : On vérifie si l'objet existe dans la table des objets et l'on récupère les informations qui lui sont liées et également si l'objet fait partie des favoris de l'utilisateur ou s'il est emprunté. On a accès aux boutons ajouter aux favoris et 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 remplit un formulaire pour les dates durant lesquelles il souhaite emprunter l'objet. Nous vérifions 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 envoyée 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 choisies on informe l'utilisateur.

Ajouter aux favoris : Lorsque le membre ajoute un objet à ces favoris. On vérifie 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. On informe l'utilisateur que l'action a bien été effectuée.

Connect   en tant que ch  uve.

**pomme**  
 Cat  gorie : fruit  
**Disponibilit  s**  
 Pas d'emprunts pr  vus, l'objet est disponible.  
**Demande d'emprunt**  
 Date d'emprunt :   
 Date de retour :   
 Effectuer la demande

**Mes emprunts**  
**Emprunts pr  vus**  
 Pas d'emprunts pr  vus.  
**Historique des emprunts**  
 Vous n'avez jamais emprunt   cet objet.

Decconnexion

Emprunt d'un objet

## 6 La gestions des messages

La r  cup  ration des messages se fait par deux requ  tes    la table Text, l'une pour r  cup  rer les messages normaux, l'autre pour r  cup  rer ceux envoy  s    tous.

En effet, plut  t que d'enregistrer plusieurs fois le m  me message envoy      tous (pour 100 utilisateur, on enregistre un m  me message de l'administrateur vers l'utilisateur 1, de l'administrateur vers l'utilisateur 2,...) nous avons d  cid   d'enregistrer ce message    tous comme un unique message qui dans la table est de la forme :

id text=n, message='message' ,date msg=date, emetteur=id emetteur recep-  
 teur=id emetteur.

Ainsi pour r  cup  rer les messages    tous il suffit de r  cup  rer les lignes de la table o   emetteur= recep- teur. Par ailleurs, pour   viter qu'un membre puisse envoyer un message    tous en se l'envoyant    soi-m  me, un test      t   ajout   dans le fichier sndMsg.php.

Il aurait   galement   t   possible de "r  server" un indice de la table membre (et emp  cher toute cr  ation de membre avec cet indice) et faire la requete id text=n, message='message' ,date msg=date, emetteur=id emetteur recep- teur=N avec N l'indice reserv  .

Pour joindre les tables Membre et Text afin r  cup  rer les pseudonymes plu  t que l'indice de l'emetteur/recep- teur, nous avons effectu   une jointure sur l'indice du membre entre Text et deux tables membre renomm  es.

Dans msgUser.php on a acc  s    un affichage des messages re  us et envoy  s. il y a deux formulaires d'envoi de message (pour membre ou administrateur), un programme javascript affiche l'un ou l'autre selon le r  le de l'utilisateur. Les formulaires sont reli  s    sndMsg.php qui ajoute les messages dans les tables s'ils sont corrects.





Messagerie d'un utilisateur

## 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 réussi à mettre en place certaines des fonctionnalités que nous souhaitions fournir au départ 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
- Connaître 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 certains utilisateurs. Ainsi nous pourrions retravailler l'aspect visuel de notre site pour le rendre plus attrayant.