Documentation technique

AnnoLigne - TPI 2015

Antonio PISANELLO

Introduction 4

Etude d’opportunité 5

Anibis 5

Olx 6

Annoligne 7

Analyse fonctionnelle 7

Généralité 7

Description des pages 8

Différentes interfaces (maquettes) 10

Accueil 10

Connexion 12

Inscription 12

Recherche 13

Afficher annonce 14

Menu annonce 15

Modifier annonce 15

Gestion compte 16

Gestion utilisateur 17

Gestion annonces 17

Gestion catégories 18

Analyse organique 19

Description de la base de donnée 19

Description des tables 19

Table Utilisateur : 19

Table Annonces: 20

Table Catégorie: 20

Model PLD 21

Description des fonctions 21

Généralité 21

Fonction bdd 22

Fonctions site 26

Fonctions de connexion 28

Fonctions de recuperation de donnée 29

Fonction affichages 32

Fonction les plus importantes / complexe / utilisée 36

Stockage des photos 41

Est-ce que tout fonctionne ? 42

Plan de test 42

Rapports de tests 42

Planning 43

Planning prévisionnel 43

Planning réel 43

Conclusion 44

Quelles sont les différences entre ce que je voulais faire et ce que j’ai fait ? 44

Quelles ont été les difficultés rencontrées ? 44

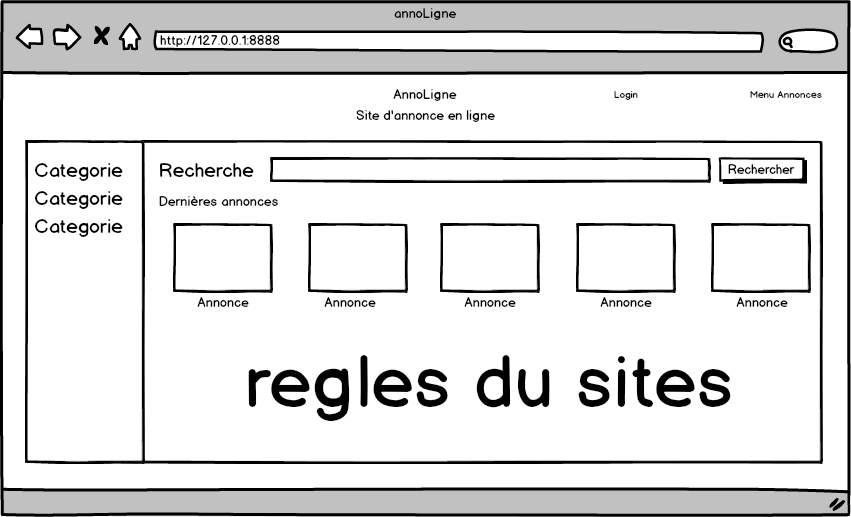
Quelles sont les futures améliorations possibles ? 44

Que m’a apporté ce projet ? 45

# Introduction

Dans le cadre de ma formation d’informaticien généraliste, je dois créer un travail pratique individuel (TPI). Mon projet sera un site web développé à l’aide d’HTML5, CSS3, PHP, SQL et JavaScript.

Mon site sera un site d’annonces en ligne comme par exemple anibis.ch ou bien olx.ch. Il comportera une page d’accueil, la possibilité de voir une annonce, poster une annonce, éditer / supprimer son annonce, ainsi que l’implémentation d’une recherche, connexion, inscription et la gestion du site, à savoir gestion des utilisateurs et des annonces ainsi que des catégories.



Ces fonctionnalités ne seront pas actives pour chaque personne par exemple un utilisateur pas encore connecté est un utilisateur anonyme et il ne peut pas poster une annonce. Les fonctionnalités par apport au statut sont définies dans le tableau ci-dessous.

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Utilisateur Anonyme | Utilisateur authentifié | Administrateur |
| S'inscrire | X |  |  |
| Se connecter | X |  |  |
| Se déconnecter |  | X | X |
| Voir une annonce | X | X | X |
| Poster une annonce |  | X | X |
| Editer une annonce |  | X | X |
| Supprimer une annonce |  | X | X |
| Rechercher une annonce | X | X | X |
| Rechercher par catégorie | X | X | X |
| Gérer les Utilisateurs |  |  | X |
| Gérer les annonces |  |  | X |
| Gérer les catégorie |  |  | X |

# Etude d’opportunité

Les sites d’annonces en ligne existent dans tous les pays et sont très nombreux sur le net. Certains critères les différencient, certains sont payants, et d’autres sont spécialisés uniquement dans un domaine (autoscout, motoscout, immoscout).

Mon site étant un site d’annonce gratuit je vais m’intéresser a deux grand de la toile Suisse a savoir olx.ch et anibis.ch.

## Anibis

Macintosh HD:Users:antoniopisanello:Documents:Developpement:Developpement Internet:TPI2015:Documentation:logo:banniere_anibis.png

A vu le jour en mars 2001, il est très simple d’utilisation et tout aussi complet. Sur anibis.ch on peut trouver de tout (pour autant que ça ait été validé). Sa recherche fonctionne très bien il y a aussi une recherche avancée, par exemple pour un véhicule on peut spécifier le nombre de kilomètres que l’on veut au maximum. Il a aussi une fonctionnalité de tri des annonces recherchée, par exemple par prix ou date d’ajout. Quand on est sur une annonce on peut facilement avoir des infos sur le vendeur, son adresse, mail, numéro de téléphone (s’il souhaite l’afficher) et on peut accéder à toutes ses annonces. De plus directement depuis la page web on peut envoyer un mail sans forcément avoir à retenir l’adresse et devoir la rentrée plus tard dans notre service mail. Anibis.ch est gratuit, mais en payant (au maximum on reste pour moins de 6chf / semaine). Les offres payantes sont plus visibles, à savoir elles seront équipées de texte en gras ainsi que des couleurs fluo pour attirer les utilisateurs et aussi elles seront affichées sur la page d’accueil du site.

Ce que j’aime chez Anibis, c’est qu’étant donné que c’est un site Suisse, et que la Suisse se découpe en trois régions linguistiques, on peut trouver des annonces en français, italien et allemand. Quand on fait une recherche il nous sort d’abord les annonces dans notre langue ensuite les annonces dans les langues différentes. C’est vraiment un très bon point qui donne du plaisir a naviguer et utiliser ce site.

Une annonce sur Anibis contient un titre, une description de l’objet un type de prix, prix fixe, à discuter, etc. Le propriétaire peut s’il le veut mettre des photos. Elle appartient a un type de catégorie, la catégorie est utilisée pour la recherche mais aussi pour des produits plus spécialiser, comme cité plus haut les véhicule ont des champs en plus.

Pour moi le point faible d’anibis.ch est qu’il est gratuit en partit, lorsqu’on fait une recherche sur le site par exemple pour un « MacBook Air » en ayant envie de trouver un ordinateur, on va devoir passer énormément de pages, car les premières pages sont très souvent des pièces pour les MacBook Air et ce sont des annonces qui ne nous intéressent pas. Il est aussi bien chargé en niveau pub ce qui peut perturber la navigation.

## Olx



A vu le jour en 2006, ce site est présent partout sur le web et dans presque tous les pays. Avant de voir le nom olx.ch le site s’appelait ricadolino.ch et était un site de la société Ricardo. Il a été vendu a olx. Le slogan d’olx est « Publier gratuitement, vendre simplement ».

Il permet comme Anibis de poster des annonces gratuitement, les rechercher il dispose aussi d’une recherche avancée. Quand on regarde une annonce on a aussi des infos sur le vendeur, on peut le contacter aussi. Contrairement à anibis.ch, olx.ch nous montre des articles similaires et il nous montre directement les autres du vendeur.

Ce que j’aime particulièrement chez olx.ch c’est qu’il n’y a pas de pub, la navigation est alors pas fatigante. De plus sur la page d’accueil il y a une liste des marchés aux puces dans notre région et on peut y ajouter son marcher au puce.

Une annonce sur Olx contient un titre une description, il peut y avoir des photos. Le prix aussi peut différer selon certains critères. L’annonce fait aussi partit d’une catégorie, ce qui rend sa recherche simplifiée. Il n’y a pas comme sur Anibis des champs supplémentaires pour certains types de produit. Il n’y a d’ailleurs aucun champ de base.

Le point faible que je reprocherais a olx.ch, serait le fait que les annonces concernant un produit soient toutes mélangée, je m’explique si on recherche une  « Ducati 848 » on trouvera plein d’annonces concernant cette motos, mais aussi on peut avoir une annonce en allemand suivis d’une annonce en français. C’est moins organisé qu’anibis.ch

## Annoligne

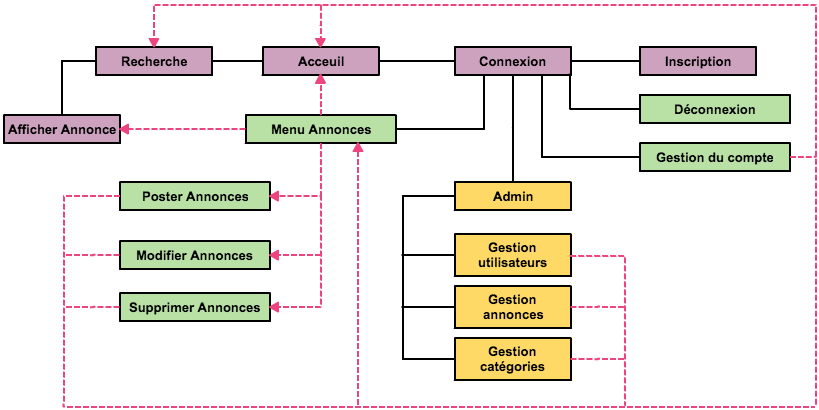
Comparons maintenant ces deus géants du web à mon projet de TPI. Tout d’abord on peut voir que les deux sites sont sur le web depuis un bon moment. Comme l’a dit Laurent Boatto (fondateur du site anibis.ch) lors d’une interview sur internet au site internet-marketer, « La première version du site est sortie en mars 2001. De nombreuses fonctionnalités ont été rajoutées par la suite, notamment les galeries photos, les cartes géographiques, les flux RSS, etc.»[[1]](#footnote-1). Cette citation montre qu’au début anibis.ch n’était pas du tout le site que l’on connaît maintenant. Il a fallut beaucoup de temps investi pour arriver à un site très complet. Contrairement a mon TPI qui dure 80heures. De plus mon site est dans une vision plus proche de ses utilisateurs, a tout moment un utilisateur qui se sent l’envie et qui a le temps peut envoyer un mail (en privé) à un administrateur pour le faire rejoindre l’équipe des admins. Sinon les principales fonctionnalités d’un site d’annonce en ligne, qui se retrouve chez Anibis et chez Olx seront présentes dans mon site.

# Analyse fonctionnelle

## Généralité

Voici le mindMap de navigation de mon site, découpé en trois couleurs :

* Violet : Utilisateur anonyme
* Vert   : Utilisateur Authentifié
* Jaune  : Admin (accès aussi aux fonctions d’utilisateur authentifié)



Mon site, comme dit plus haut, se découpe en 3 partie pour 3 types d’utilisateurs. Les pages Accueil, connexion, menu annonce, gestion du compte ainsi que les 3 pages admins (qui se regroupe dans un menu déroulant) sont des lien accessible par toutes les pages du site (des contraintes de connexion peuvent empêcher certains liens).

## Description des pages

**Accueil :** La page d’accueil affiche les 4 dernières annonces ajoutées. Elle affiche aussi les principales règles afin que tous les utilisateurs qui auront lus ces règles pourront utiliser le site correctement et elle permet aussi de lancer une recherche.

**Connexion  :** Cette page permet à un utilisateur anonyme de se connecter ou bien de s’inscrire. Si un utilisateur se connecte ce lien se transforme en déconnexion. Quand l’utilisateur entre son pseudo et son mot de passe, la page connexion fait appel a la page login pour se connecter ainsi le code est sur une autre page. Cette redirection n’est pas visible pour l’utilisateur. La page connexion envoie a la page login le pseudo/mail et le mot de passe en paramètre « POST » pour que la page login puisse les récupérer.

**Inscription :** Une fois le formulaire correctement remplit l’utilisateur est inscrit. Il sera connecter directement, pas besoin de retourner sur la page connexion. Les fonction d’inscription se font sur même page, par contre comme pour le login une fois que la fonction d’inscription a été exécutée, elle redirige sur le login afin que l’utilisateur puisse se connecter.

**Recherche :** Pour accéder a cette page il y a deux façon soit on a lancer une rechercher depuis la page accueil et la le PHP va récupérer le texte entré et va aller chercher dans la base de donnée les annonces qui pourraient nous intéresser. Ou alors en cliquant sur une catégorie, a ce moment la le PHP vas aller dans la base et récupérer toutes les annonces de cette catégorie.

**Afficher annonce  :** Comme le dit son nom elle permet d’afficher une annonce, et pour afficher une annonce il suffit simplement de cliquer sur une annonce (sur l’index, la recherche, la gestion d’annonce ou bien le menu annonce). Cette page affichera le titre, la catégorie, la description, le prix, les éventuelles photos et des infos supplémentaire (la date de validité ainsi que le pseudo du vendeur). Il sera aussi possible d’envoyer un mail au vendeur (depuis une application externe). Pour afficher une annonce j’aurais besoin de l’ID de cette dernière, et je le passerais en parametre dans l’url. Il y aura aussi un autre parametre, mais pas toujours présent, ce parametre sera le parametre droit qui permet de voir une annonce meme si elle n’est plus active ou bien pas validée. L’utilisation de se parametre est utile par exemple pour une annonce qui n’as pas encore ete validée par un administrateur mais comme c’est notre annonce on doit pouvoir la voir.

**Menu annonce :** Menu annonce est une page web qui permet à un utilisateur authentifié d’insérer une annonce et de pouvoir voir ses annonces déjà publiées. Depuis cette page il peut (s’il a déjà publié des annonces), les supprimer ainsi que les modifier. Modifier l’annonce se fait sur une page distincte alors que supprimer est juste une commande exécutée sur menu annonce. Pour modifier l’annonce comme pour l’affichage, je passerais l’id en parametre. Supprimer une annonce renvoie sur une page de confirmation de suppression, on passe aussi l’ID en parametre et sur la page de confirmation on recupere l’annonce et on la supprime.

**Modifier annonce** **:** Comme son nom l’indique encore une fois, elle permet de modifier une annonce. On y accède en cliquant sur modifier dans le menu annonce. Dans cette page il est possible de modifier le titre, la catégorie, le prix, les photos, la description et le jour où l’on veut qu’elle entre en compte. Un point fort de cette page est qu’elle nous donne un aperçut de l’annonce tout en la modifiant. On peut aussi supprimer l’annonce depuis cette page. Comme avant et pour presque tout le site je passe l’ID en paramètre lors de la suppression, une confirmation est demandée.

**Gestion compte :** Gestion compte permet à tout utilisateur connecté de changer certaines infos sur lui, à savoir : son image d’avatar, son Email, et son mot de passe. Pour savoir si un utilisateur a un avatar ou pas, on récupère la valeur booléenne « Avatar » dans la base de donnée dans la table Utilisateur à l’aide de l’ID. l’id est passé en paramètre.

**Gestion utilisateurs :** Cette page permet à un administrateur de rendre un utilisateur Admin ou bien de le supprimer. Elle affiche aussi le nombre d’annonces postées par ce dernier. Pour rendre un utilisateur admin il y a un bouton qui renvoi sur cette page et qui change la valeur « Statut » dans la table Utilisateurs. Pour savoir de quel utilisateur on parle on passe en paramètre dans l’url id de l’utilisateur. Si l’administrateur decide de supprimer l’utilisateur une confirmation de suppression lui ait demandé.

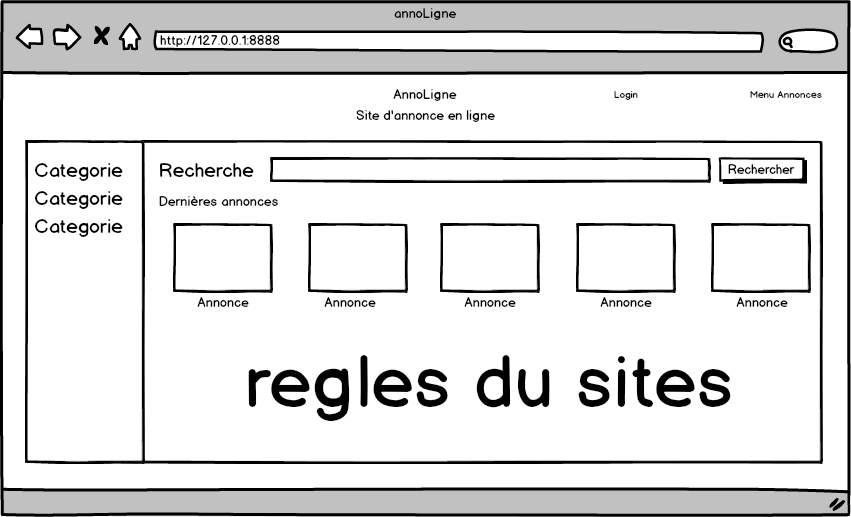
**Gestion annonces :** Cette page regroupe toutes les annonces n’ayant pas encore été validée. Une annonce ajoutée par un utilisateur est ajoutée dans la base mais n’est pas valide, seul un administrateur a le pouvoir de la rendre valide. S’il la trouve inapproprié il peut la supprimer. Si l’admin decide de supprimer l’annonce il devra confirmer qu’il veut bien la suppriemer. De son coter l’utilisateur dans la page menu annonce peut lui activer ou pas son annonce. L’administrateur recevra seulement les annonces actives et qui n’ont pas encore été validée. Comme pour toutes les nouvelles annonces, a chaque fois qu’on modifie l’annonce elle repasse a inactive.

**Gestion catégories :** Un administrateur utilisera cette page pour voir les catégories du site. Si une catégorie est inappropriée il peut la supprimer ou la modifier, toutes les annonces de cette catégorie, seront transférée dans la catégorie privée « INCONNUE » et son propriétaire devra changer la catégorie dans la modification d’annonce afin de la rendre de nouveau visible.

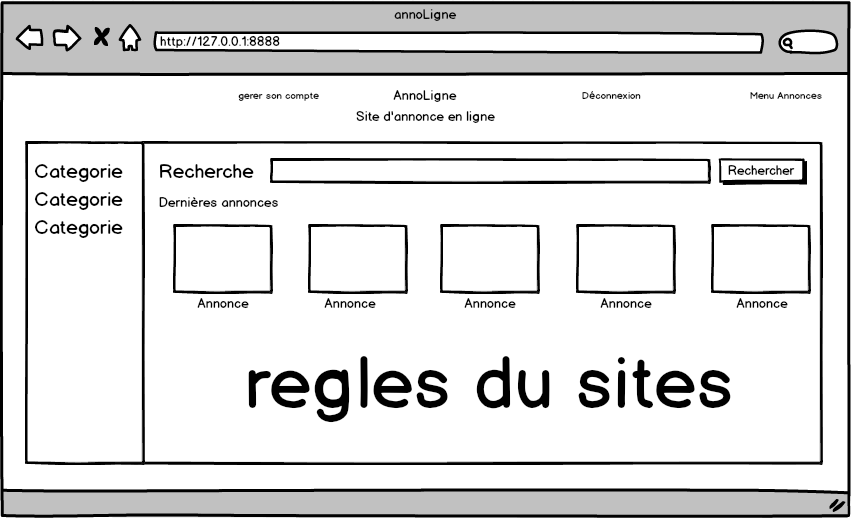
## Différentes interfaces (maquettes)

### Accueil

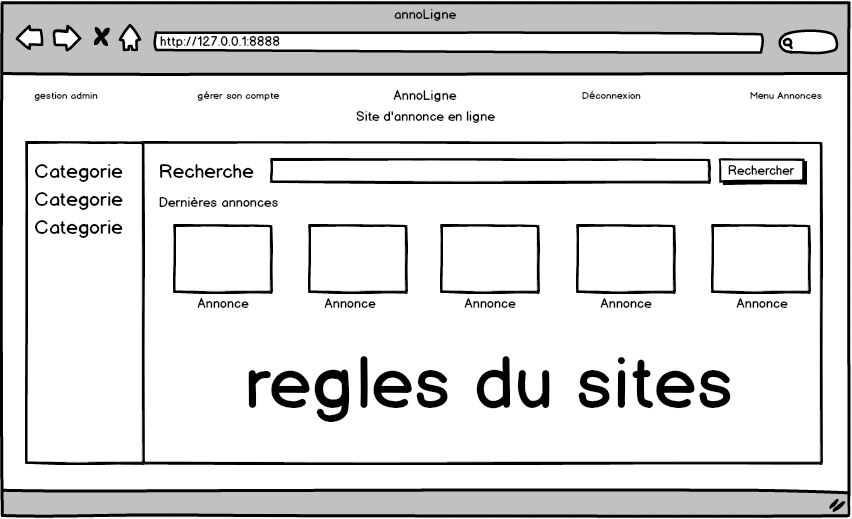
#### Utilisateur anonyme



#### Utilisateur authentifié

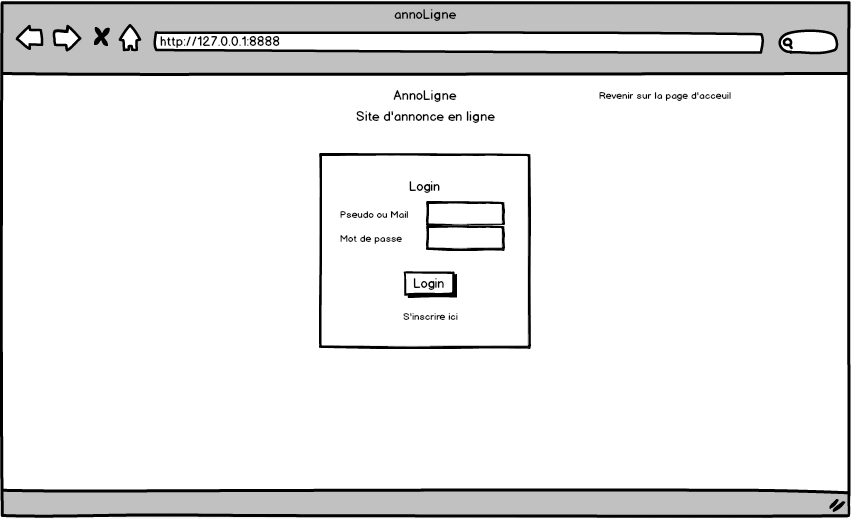


#### Admin



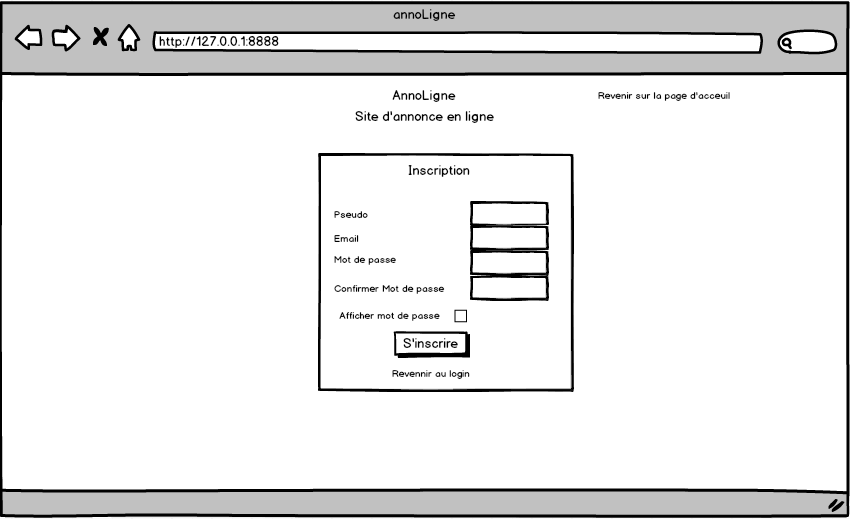
La différence entre les trois interfaces est que suivant qui est connecté des liens apparaissent dans le menu en haut.

### Connexion



On peut se connecter de deux façons différentes soit en utilisant l’adresse e-mail ou bien en entrant le pseudo. Le bouton login permet de lancer la requête de login et le lien « s’inscrire ici » nous envoie sur la page inscription. Il y a aussi un lien en haut a droite qui nous ramène à la page d’accueille.

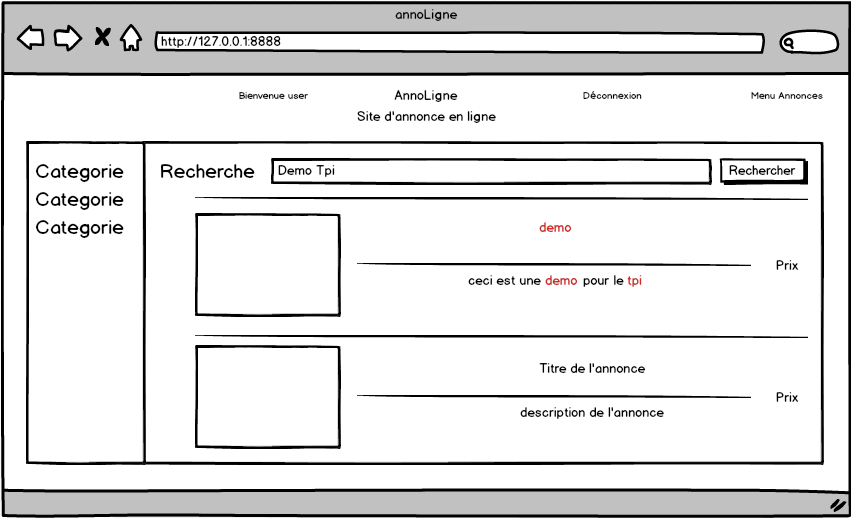
### Inscription



En complétant comme il faut le formulaire et en appuyant sur le bouton inscription on s’inscrit au site et si tout à bien été remplit et comme il le faut on est connecter directement et redirigé à l’accueil. Pour s’inscrire il faut rentrer un Pseudo, un Email, un MDP et le confirmer. Le checkBox en bas du formulaire permet de rendre les deux champs mot de passe visible. Le lien « Revenir au login » permet de revenir en arrière et de se retrouver sur la page de connexion. Comme pour la page Connexion le lien en haut a droite permet de retourner sur la page d’accueil.

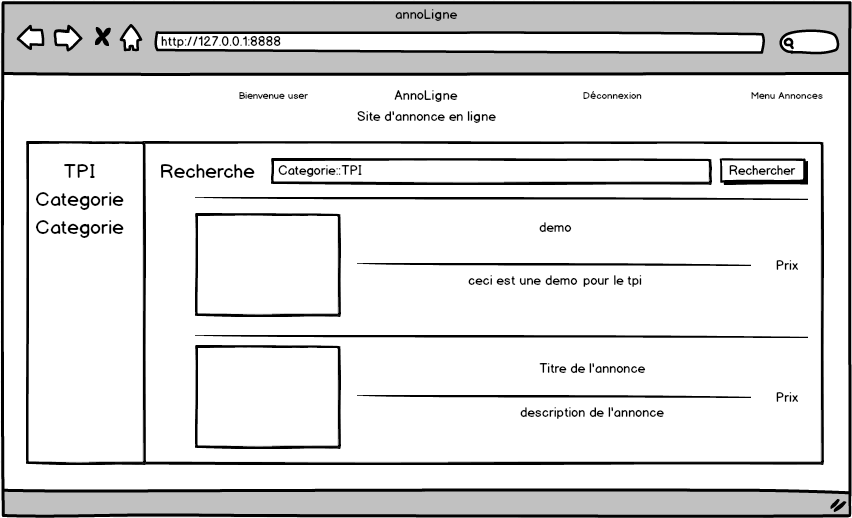
### Recherche

#### TextBox recherche



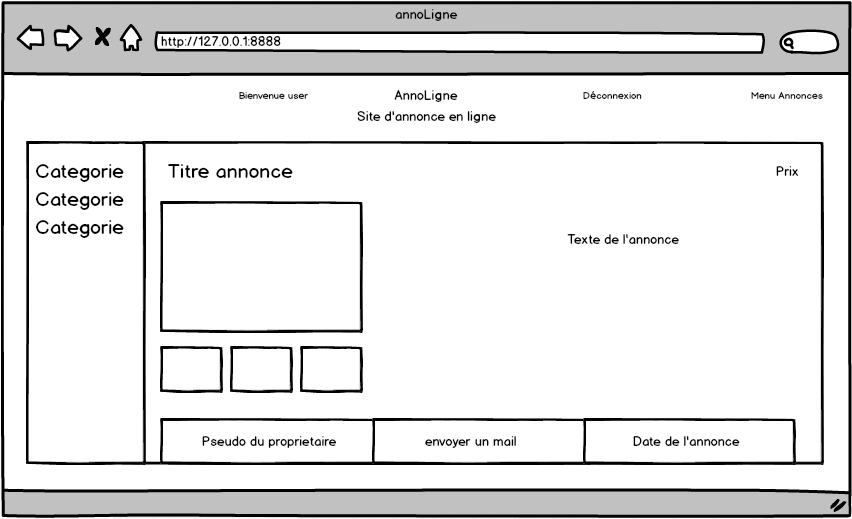
La recherche après avoir utiliser le textBox de recherche va sélectionner tous les mots recherchés qui se retrouvent dans le titre ou la description et quand elle trouve une concordance elle réécrit le texte en rouge pour montrer ce qu’elle a trouver.

#### Catégorie



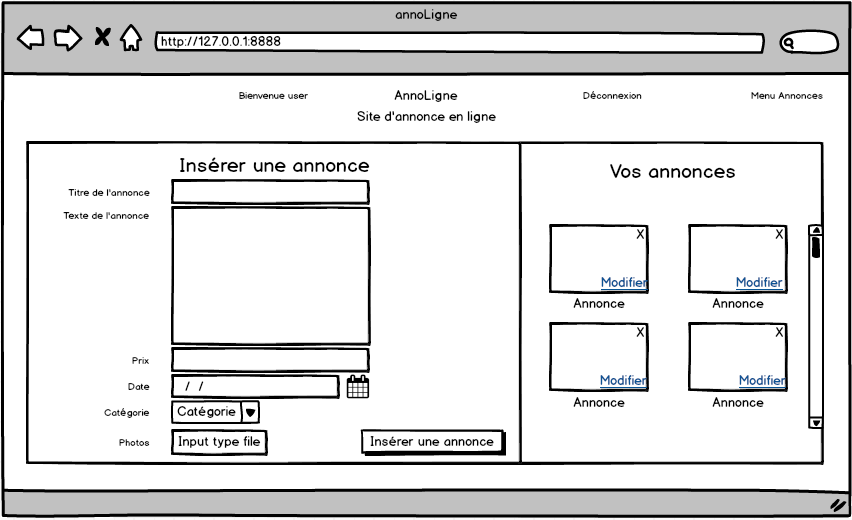
La recherche par catégorie affiche toutes les annonces d’une catégorie.

### Afficher annonce



En cliquant sur les photo miniatures, ca les change avec la photo principale et donc permet de voir les miniatures en plus grand.

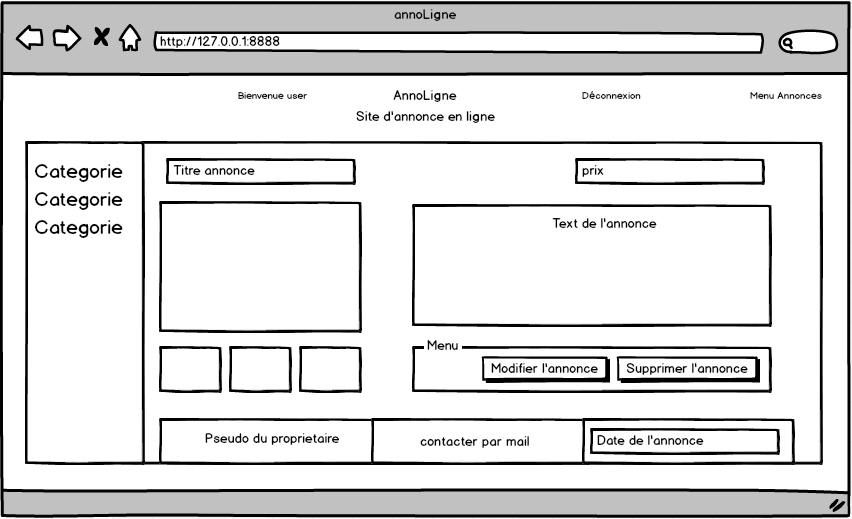
### Menu annonce



Le champ titre de l’annonce est un simple input text, le texte de l’annonce un textArea, le prix est un input text, la date un champ date, catégorie un select, et les photos un input type file multiSelect. Une fois tous les champs remplis et le bouton « Insérer une annonce » l’annonce est ajoutée dans la base mais un admin doit l’approuvée pour qu’elle soit visible.

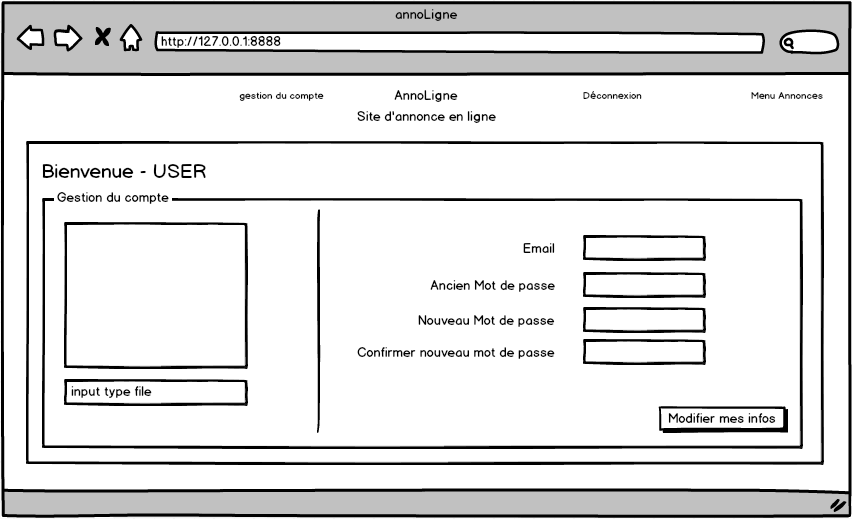
Dans la section « Vos Annonces » deux liens sont présents sur chaque photo des annonces. Le premier lien une croix, permettra de supprimer l’annonce. Si ce lien est cliqué une confirmation de suppression sera demandée. Le second lien modifier, permettra de se rendre sur la page de modification des annonces afin de modifier cette annonce.

### Modifier annonce



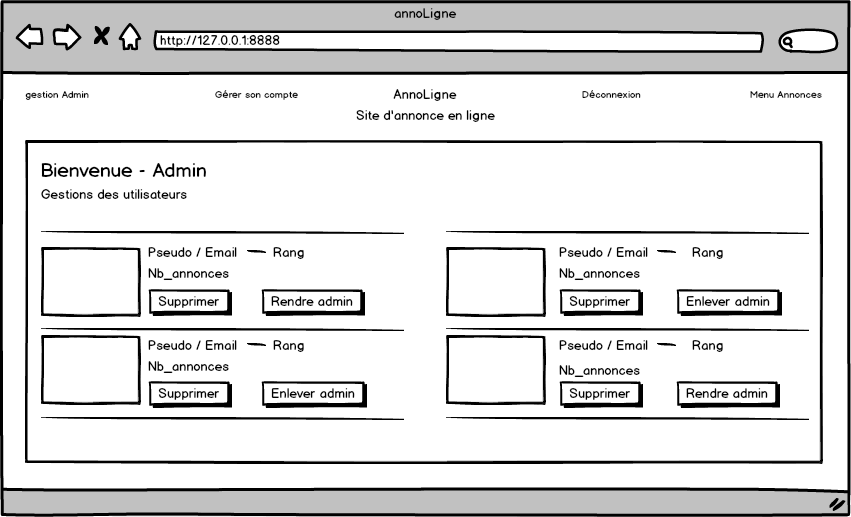
Pour modifier l’annonce, il faut que tous les champs soient remplis comme il le faut et il ne faut pas oublier de valider le formulaire avec le bouton modifier l’annonce sinon les changements ne seront pas pris en compte. En appuyant sur « supprimer l’annonce » une confirmation est demandée afin de pas commettre une erreure.

### Gestion compte



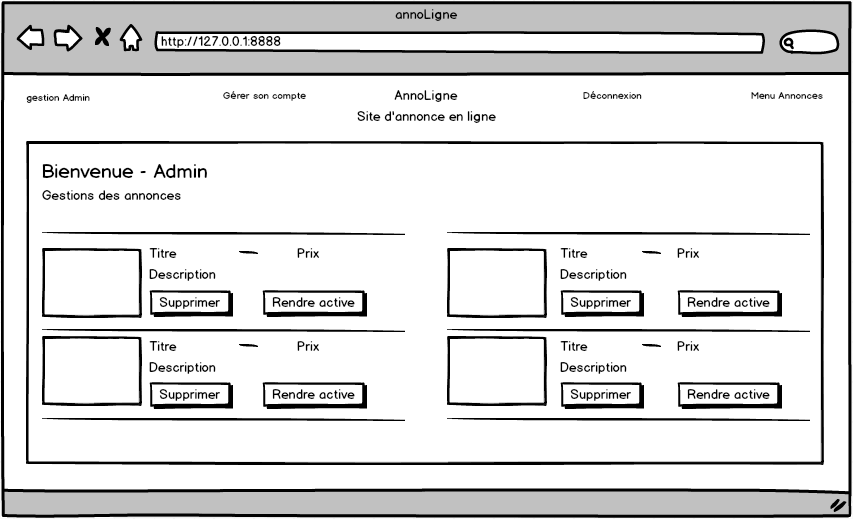
En cliquant sur le bouton modifier, tous les champs remplis seront changer si le champs est vide il ne sera pas modifier.

### Gestion utilisateur



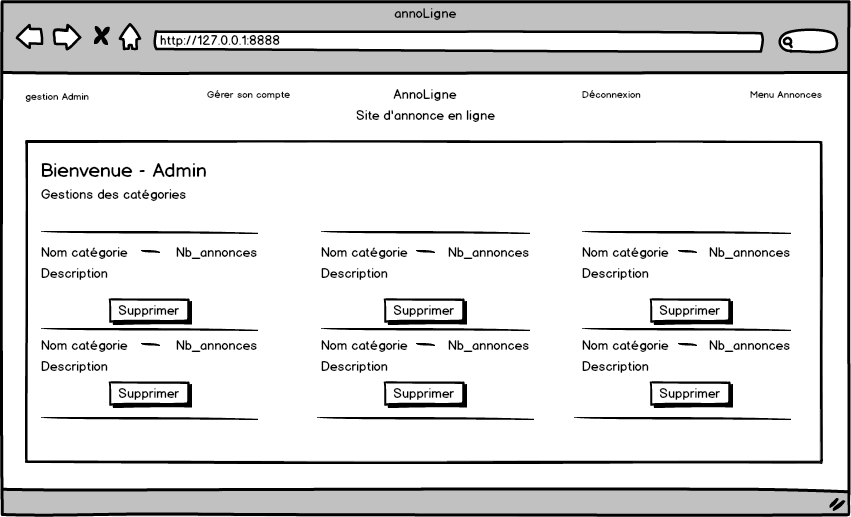
En cliquant sur supprimer une confirmation de suppression sera demandée à l’admin.

### Gestion annonces



Seuls les annonces non actives sont visible, une annonce qui a été validée ne peut plus être modifiée ou supprimée par un administrateur. Seul un vendeur peut supprimer son annonce validée. Si un administrateur veut supprimer une annonce et qu’il appuie sur le bouton « Supprimer » une confirmation sera demandée.

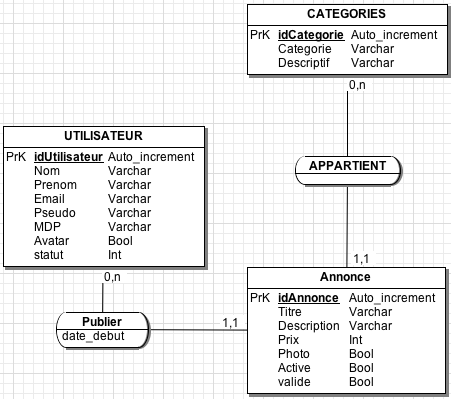
### Gestion catégories



Si un administrateur veut supprimer une catégorie une fois de plus une confirmation de suppression lui sera demandée.

# Analyse organique

## Description de la base de donnée



Ma base de donnée comporte 3 tables, La table utilisateur, la table annonce et la table catégories.

Un utilisateur peut poster plusieurs annonces mais une annonce est postée par un seul et unique utilisateur. De la même façon une catégorie peut appartenir a plusieurs annonce mais une annonce contient qu’une seule catégorie.

### Description des tables

### Table Utilisateur :

|  |  |  |
| --- | --- | --- |
| Nom | Type | Description |
| IdUtilisateur | Int  autoIncrement  clé primaire | Id des utilisateur afin d’avoir une valeure unique |
| Nom | Varchar | Nom de l’utilisateur |
| Prénom | Varchar | Prénom de l’utilisateur |
| Email | Varchar  unique | Email de l’utilisateur |
| Pseudo | Varchar  unique | Pseudo de l’utilisateur |
| MDP | Varchar | Mot de passe de l’utilisateur |
| Avatar | Booléen | Valeure booléenne qui indique si l’utilisateur a un avatar ou pas |
| Statut | Int | Statut de l’utilisateur  0 utilisateur  1 admin |

### Table Annonces:

|  |  |  |
| --- | --- | --- |
| Nom | Type | Description |
| idAnnonce | Int  Autoincrément  Clé primaire | Id unique de l’annonce |
| Titre | Varchar | Titre de l’annonce |
| Description | Varchar | Description de l’annonce |
| Prix | Varchar | Prix de l’annonce |
| Photo | Booléen | Valeure booléenne qui indique si l’annonce contient des photos ou pas |
| Active | Booléen | Valeure booléenne qui indique si l’utilisateur veut afficher son annonce ou pas |
| Valide | Booléen | Valeure booléenne qui indique si un admin a validé l’annonce |

### Table Catégorie:

|  |  |  |
| --- | --- | --- |
| Nom | Type | Description |
| idCategorie | Int  Autoincrément  Clé primaire | Id unique pour chaque catégorie |
| Categorie | Varchar | Nom de la catégorie |
| Descriptif | Varchar | Description de la catégorie |

### Model PLD

Utilisateurs

(IdUtilisateur, nom, prenom, email, pseudo, mdp, avatar, statut)

Annonces

(IdAnnonce, titre, description, photo, prix, active, valide, #idUtilisateur, #idCategorie)

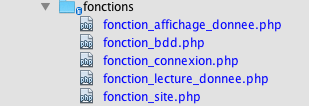
Categorie

(IdCategorie, categorie, descriptif)

## Description des fonctions

### Généralité

J’utilise plusieurs fonctions dans mon site. J’ai des fonctions qui traite des données sans avoir besoin de connexion a la base de donnée, j’ai des fonction qui récupèrent des infos de la base de donnée, j’ai des fonction qui affiche ce que je récupère de la base de donnée, d’autre qui font des actions sur la base et des fonctions pour tout ce qui est connexion, inscription. Pour des raisons de lisibilités, j’ai toutes ces fonctions réparties dans plusieurs feuilles php différentient car aller chercher une fonction en particulier alors qu’elles sont toutes dans la même feuille c’est compliqué.



Ces pages sont indépendante les unes des autres, mis a part les pages, fonction\_affichage\_donnee.php qui regroupe les fonctions pour afficher ce que l’on récupère de la base qui se trouvent dans fonction\_lecture\_donnee.php. De règle générale si j’ajoute une nouvelle fonction dans l’une je dois en ajouter une autre en rapport dans l’autre.

L’école ne nous a par appris a faire du MVC (Model, vue, contrôleur), et je n’ai malheureusement pas eu le temps de l’apprendre assez avant le TPI pour pouvoir me permettre de l’utiliser lors de ce projet. Néanmoins je sais que si possible il faut séparer un maximum les données de l’affichage du traitement. C’est pourquoi tout mon site fonctionne ainsi, d’abord je récupère mes données, et je les afficher. Ce qui fait que chaque fonction d’affichage prend en paramètre un tableau, et ce tableau est une autre fonction.

Macintosh HD:Users:antoniopisanello:Desktop:Capture d’écran 2015-05-13 à 00.12.07.png

### Fonction bdd

Les fonctions BDD sont toutes les fonction qui ont un rapport avec la base de donnée masi qui ne recuperent aucune donnée, par exemple toutes les fonction d’update et d’insert seront la dedans.

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la fonction | Description | Paramètres entrée | Valeur retournée |
| Connexion | Permet de se connecter a une base de donnée et de generer un objet PDO | $db\_name : le nom de la base de donnée a se connecter  $host : le serveur ou est stoquée la base de donnée  $user : l'utilisateur pour se connecter a la base  $pwd : son mot de passe | retourne un objet PDO |
| Récupère\_user\_by\_id | Va chercher des infos dans base de donnée pour un utilisateur précis, à l’aide de son ID  \* c’est une fonction de recuperation de donnée qui normalement devrait se trouver dans la page lecture donnée mais étant donné que c’est pas déstiné a un affichage j’ai préféré la mettre dasn la page fonction\_bdd.php | $id : l’id de la personne que l’on veut rechercher  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Retourne un tableau  (Prenom, Nom, Statut) |
| Modification\_avatar | Change la valeure booléenne « Avatar » dans la base de donnée | $id : l’id de la personne pour  laquelle on veut change sa valeure  $avatar : la valeure a mettre  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Ajout\_annonce | Ajoute une annonce dans base de donnée | $titre : titre de l’annonce a ajouter  $description : description de l’annonce  $date : date de début de l’annonce  $prix : prix de l’annonce  $idUtilisateur : id de l’utilisateur qui a ajouté l’annonce  $idCategorie : id de la categorie dans laquelle se trouve l’annonce  $photo : valeure booléenne qui précise s’il y a des photos ou pas  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php |  |
| Maj\_annonce | Met a jour une annonce en changeant ses données | $id : id de l’annonce a modifier  $titre : titre de l’annonce a modifier  $description : description de l’annonce  $date : date de début de l’annonce  $prix : prix de l’annonce  $Categorie : id de la categorie dans laquelle se trouve l’annonce  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Supprimer\_annonce | Supprime un annonce de la base | $id : id de l’annonce a supprimer  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Maj\_photo | Change la valeure booléenne dans la table annonce afin de savoir si une annonce a une photo ou pas | $id : id de l’annonce a modifier  $valeure\_photo : valeure booléenne qui précise s’il y a des photos ou pas  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Maj\_active | Change la valeure booléenne dans la table annonce afin de savoir si une annonce est active ou pas | $id : id de l’annonce a modifier  $valeure\_active : valeure booléenne qui précise si l’annonce est active ou pas  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Maj\_valide | Change la valeure booléenne dans la table annonce afin de savoir si une annonce est valide ou pas | $id : id de l’annonce a modifier  $valeure\_valide : valeure booléenne qui précise si l’annonce est valide ou pas  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Supprimer\_categorie | Supprime une catégorie | $id : id de la catégorie a supprimer  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Maj\_categorie | Met a jour une catégorie | $id : id de la catégorie a modifier  $nom\_categorie : nom e la catégorie a mettre dans la base  $descriptif : descriptif de la catégorie  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Ajouter\_categorie | Insère une nouvelle catégorie | $id : id de la catégorie a ajouter  $nom\_categorie : nom e la catégorie a mettre dans la base  $descriptif : descriptif de la catégorie  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Supprimer\_utilisateur | Supprime un utilisateur | $id : id de l’utilisateur a supprimer  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Maj\_admin | Change un utilisateur | $id : id de l’utilisateur a modifier  $valeure : valeure booléenne qui précise si l’utilisateur devient admin ou pas  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | RIEN |
| Search | Fonction de recherche d’annonce, elle va chercher dans la base de donnée les annonces qui correspondent avec des termes rechrchés | $tableau : tableau PHP de string ce seront les termes a rechercher  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | $tableau\_text\_gras : tableau a deux dimension qui montres les valeures trouvées et elles sont en gras |

### Fonctions site

Fonctions site regroupe toutes les fonctions, qui n’ont rien a voir avec la base de donnée. Ce sont des fonction plus basé systeme et calcul afin de faire des test.

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la fonction | Description | Paramètres entrée | Valeure retournée |
| Var\_dump\_pre | Permet de fair un var\_dump preformater en une ligne | $var : variable a tester | Var\_dump préformatter |
| Contient\_chiffre | test une chaine afin de savoir s'il y a des chiffre dedans | $chaine : chaine a tester | retourne oui ou non en fonction de ce qu'il trouve |
| couper\_avec\_separateur | permet de decouper une chaine a partir d'un separateur | $chaine : chaine à tester  $separateur : separeteur pour separer les éléments d'une chaine | tableau avec chaques element dans une case |
| couper\_espaces | Coupe une chaine a chaques fois qu'il y a un espace | $chaine : chaine à tester | tableau avec chaques element dans une case |
| date\_ajourdhui | retourne la date d'ajourd'hui au format YYYY-mm-dd |  | Objet date avec la date d’aujourd’hui |
| savoir\_les\_jours\_restants | calcule le nombre de jours entre aujourd'hui et une date donnée  retourne le nombre de jours ains que s'il a reussi a calculer ou pas | $date\_user : date a verifier | tableau : [0] => le nombre de jours | [1] => s'il a reussi ou pas |
| changer\_formats | Change le formate entre un $\_File et les formats connus pour les images | $type : valeur du champs format du $\_File | string : le format normal |
| mettre\_fichier\_dossier\_dans\_tableau | Lis les fichier d'un dossier et le retorune sous forme de tableau PHP | $dossier\_parametre : dossier a lire | tableau : avec les fichiers du dossier |
| dossier\_existe | Verifie si un dossier existe | $dossier : dossier a verifier | boolean : retourne true si le dossier existe false si non |
| supprimer\_photo | Supprimer les photos pour une annonce | $nom\_photo : nom de la photo a supprimer  $id\_annonce : l’id est utilisé comme nom de dossier | RIEN |
| Supprimer\_photo\_avatar | Supprimer la photo d’un avatar | $nom\_photo : nom de la photo a supprimer | RIEN |
| Efface\_dossier | Supprime un dossier | $dossier : le chemin du dossier a supprimer | RIEN |
| copie\_donnee\_unique | Lis un tableau et regarde s'il y des doublons si oui il les supprime afin d'avoir uniquement des données unique | $tableau : tableau a verifier | tableau : tableau unique |
| savoir\_nombre\_photo\_max | Lis un dossier et retourne le nombres d'élements | $id\_annonce : l'id est utilisé comme nom de dossier | max\_num : int avec le nombre de fichiers dans le dossier |
| afficher\_erreur | En fonction d'un chiffre la fonction retourne une érreure  \*pas toujours une eurreure, possibilité de retourner une remarque ou une confirmation | $erreur : chiffre de l'eurreure | retourne un alerte javascript avec l'erreur |
| creer\_menu\_admin | Crée les liens necessaires pour accéder au menu admin | $chemin : chemin pour se rendre a la racine du site afin de pouvoir l’utiliser sur plusieurs pages | Retourne un String avec les liens crées |
| convertir\_statut | convertit un int en string afin d’ecrire admin et non 1 | $statut : valeure int du statut de l’utilisateur | Retourne un string |
| mettre\_text\_en\_gras | Met du texte en rouge lorsqu’il le trouve | $mot\_rechercher : mot a chercher dans la plage  $plage : suite de mot dans laquelle se trouve peut etre le mot rechercher | Un string la plage avec les modification possiblement apportée |

### Fonctions de connexion

Les fonctions de connexion sont utilisées pour se connecter et s’inscrire au site.

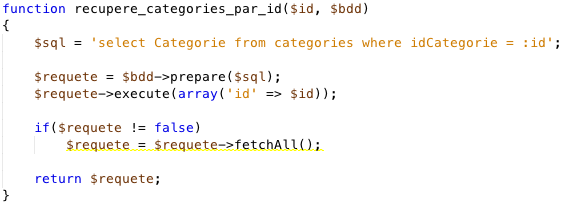
|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la fonction | Description | Paramètres entrée | Valeur retournée |
| Login | Fonction qui permet de connecter un utilisateur | $user : le pseudo ou le mail pour se connecter  $mdp : le mot de passe pour se connecter  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Retourne un tableau avec l’id de l’utilisateur ou alors le tableau est vide |
| Verifie\_formulaire | Fonction qui verifie le formulaire d’inscription afin de pas entrer n’importe quoi | $tableau : tableau avec les valeures du formulaire  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Retourne rien mais si tous les champs sont correctement entrées il inscrit l’utilisateur et redirige la page web |
| Inscription | Inscrit un nouvel utilisateur a la base de donnée | $tableau : tableau contenant les infos nécessaires pour l’inscription  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Retourne un int avec l’id de la personne entrée |

### Fonctions de recuperation de donnée

Les fonctions de récupération de données fonctionnent toutes de la même logique. A savoir une requête SQL avec de possible paramètre ou pas. S’il y a des paramètres je les « bind » (les sécurise afin d’éviter les injection SQL). Une fois que j’ai récupéré ce que j’ai besoin j’utilise une fonction d’affichage pour afficher les données que j’ai récupéré.

Se référrer a la figure 4 qui montre bien ce fonctionnement.

Voici un exemple de fonction de lecture de donnée, on peut voir le « bindage » des parametres et la structure de mes fonction de lecture de données.



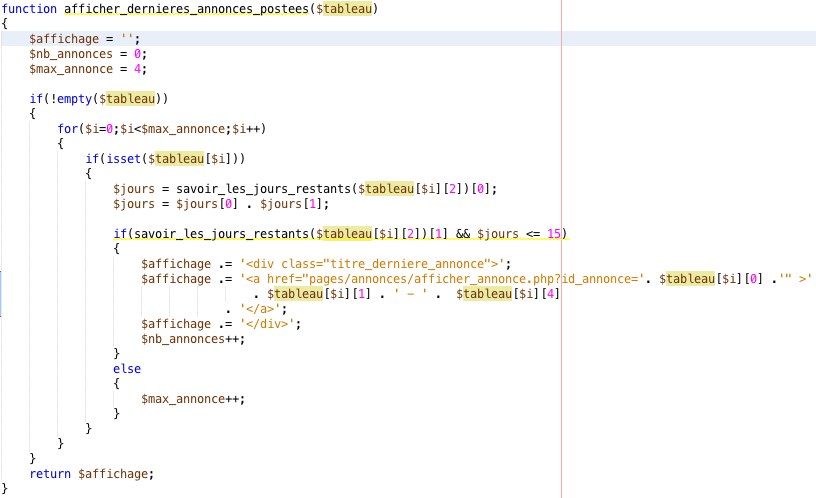
|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la fonction | Description | Paramètres entrée | Valeur retournée |
| Recupere\_categorie | Recupere les categories de la base et les regroupe par nom | $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant les categories |
| Recupere\_dernieres\_annonces\_postee | Recupere toutes les annonces et les trie par date decroissante | $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant les annonces |
| Recupere\_utilisateur\_par\_id | Recupere les infos de l’utilisateur par apport a l’id  \*Elle differe de la fonction citée precédement, car celle ci est utilisé pour les avatar et selectionne plus de choses | $id : id de l’utilisateur a récuperer  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant les infos de l’utilisateur |
| Recupere\_annonces\_utilisateur | Recupere les annonces d’un utilisateur, pour antant qu’elle soient validée et activent | $id : id de l’utilisateur  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant les annonces |
| Recupere\_annonce\_par \_id | Recupere une annonce en utilisant son id | $id : id de l’annonce  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant l’annonce |
| Recupere\_annonce\_par \_categorie | Recupere les annonces en donnant une categorie | $nom\_categorie : le nom de la categorie pour recuperer les annonces  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant les annonces |
| Recupere\_categorie\_par\_id | Recupere une catégorie d’après son id | $id : id de la catégorie  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant la catégorie |
| Recupere\_annonces\_non\_valide | Recupere les annonces qui sont en attente de validation | $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Un tableau contenant les annonces |
| Recupere\_categories\_gestion\_categorie | Recupere toutes les catégories  \*Aucun tri | $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant toutes les catégories |
| Recupere\_nb\_annonce\_par\_categorie | Recupere le nombre d’annonces active et valide par annonces | $id : id de la catégorie  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant le nombre d’annonces |
| Recupere\_categories\_par\_id\_gestion | Récupère une catégorie pour l’afficher dans la page modifier catégorie | $id : id de la catégorie  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant la catégorie |
| Recupere\_utilisateur\_gestion | Recupere tous les utilisateur pour les affichés dans la page gestion avec le nombre d’annonces qu’ils on publiés | $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Tableau : contenant tous les utilisateurs |
| Recupere\_annonces\_ajax | Recupere les annonces et met en forme une du texte afin de le récupérer en AJAX | $requete : la requete de selection des annonces  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php | Du texte sur une page php |

### Fonction affichages

Les fonctions d’affichages fonctionnent toutes de la même façon, c’est à dire un tableau est passé en paramètre. Ce tableau est une autre fonction qui s’occupe de récupérer les données. Ensuite grace a ce tableau et aussi du fait que je connaisse sa structure je peux créer l’affichage. L’affichage est généré automatiquement, normalement seul la structure de la page web basic (banniere, contenant et pied de page) sont en HTML le reste est générer en php.

Se référrer a la figure 4 qui montre bien ce fonctionnement.

Voici un exemple de fonction d’affichage, on vient qu’elle prend un tableau en paramètre et que la fonction l’utilise pour afficher des infos.

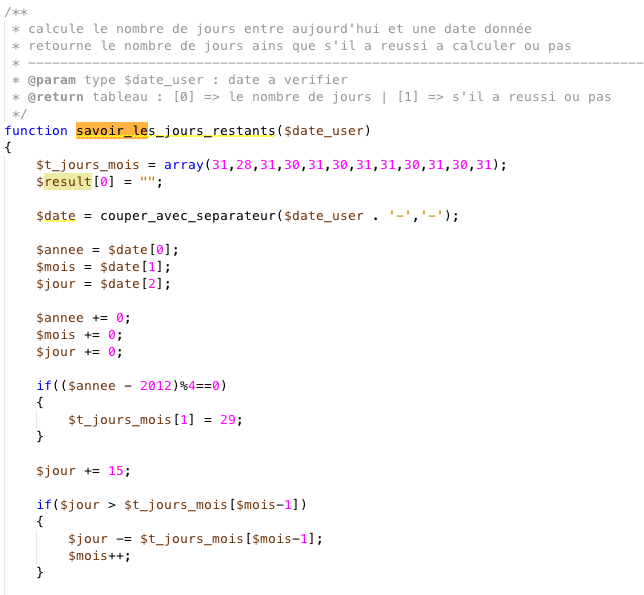


|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la fonction | Description | Paramètres entrée | Valeur retournée |
| Afficher\_categories | Affiche les categories sur le coté avec le nombre d’annonce dans la catégorie | $tableau : les catégories récupérées de la base  $mode : mode d’affichage en fonction de la page (surtout pour les liens)  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php (c’est une des seules fonctions qui en a besoin dans l’affichage, car elle doit récupérer le nombre d’annonce par catégorie) | $affichage : string contenant l’html généré par le PHP |
| Afficher\_photo\_dernieres\_annonces | Affiche les photos sur la page index | $tableau : tableau des annonces triées par date décroissante | $affichage : string contenant l’html généré par le PHP |
| Afficher\_dernieres\_annonces\_postees | Affiche le titre de l’annonce ainsi que le prix sur la page index | $tableau : tableau des annonces triées par date décroissante | $affichage : string contenant l’html généré par le PHP |
| Affiche\_combobox\_categories | Affiche le combobox pour selectionner la catégorie dans la page menu annonce et modifier annonce | $tableau : tableau des catégories récupérées de la base | $affichage : string contenant l’html généré par le PHP |
| Afficher\_annonce\_utilisateur | Affiche les annonces dans la rubrique « Vos annonces » de la page menu annonce | $tableau : tableau des annonces pour un utilisateur | $affichage : string contenant l’html généré par le PHP |
| Afficher\_annonce | Affiche une annonce dans la page afficher\_annonce | $tableau : tableau contenant l’annonce  $id : id de l’annonce  $droit : pour afficher l’annonce meme si elle st non valide et non active  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php (c’est une des seules fonctions qui en a besoin dans l’affichage, car elle doit récupérer la catégorie) | $affichage : string contenant l’html généré par le PHP |
| Afficher\_photo\_annonce | Affiche la photo principale de l’annonce | $id\_annonce : l’id est utilisé comme nom de dossier et on en a besoin dans le chemin | $photo : balise img contenant la photo et en meme temps c’est aussi un tableau qui contient les autres photos et qui sera disponible pour une autre fonction |
| Affiche\_photo\_miniature | Affihe les petites photos en dessous de la photo principale dans la page affiche\_annonce | $tableau : le tableau photo cité ci-dessus | $affichage : string contenant l’html généré par le PHP |
| Afficher\_annonces\_gestion\_annonce | Affiche les annonces qui sont active mais pas valides pour les affichés dans la page gestion annonce | $tableau : tableau des annonces recupérés de la base | $affichage : string contenant l’html généré par le PHP |
| Afficher\_categorie\_gestion\_categorie | Affiche les catégories dans la page gestion categories | $tableau : tableau des catégories récupéré de la base  $bdd : l’instance PDO qui permet de faire la liaison entre la base et le php (c’est une des seules fonctions qui en a besoin dans l’affichage, car elle doit récupérer le nombre d’annonce par catégorie) | $affichage : string contenant l’html généré par le PHP |
| Afficher\_utilisateur\_gestion | Affiche l’utilisateur dans la page gestion utilisateurs | $tableau : tableau des utilisateur précedment récupérés de la base | $affichage : string contenant l’html généré par le PHP |
| Afficher\_annonce\_recherchee | Afficher les annonces trouvées dans la base sur la page recherche | $tableau : tableau contenant les annonces trouvées sur la base | $affichage : string contenant l’html généré par le PHP |

## Fonction les plus importantes / complexe / utilisée

J’ai beaucoup de fonction mais la plus part sont des fonction de lecture de donnée et d’affichage. Ces fonction sont presque identiques les unes des autres et ne sont pas très complexe. Elles ont leur importance mais d’autres fonctions sont beaucoup plus complexe et j’ai du beaucoup plus réfléchir dessus. Elles méritent donc que je les explique un peu plus que de simples commentaires et descriptions.

Ma première fonction sera « savoir\_les\_jours\_restants », elle n’est pas forcément compliquée, mais je l’utilise beaucoup et elle est importante car elle test si une annonce peut être affichée ou pas. Elle prend la date de l’annonce, elle prend la date d’aujourd’hui et elle fait la différence entre les deux afin de retourner le nombre de jours restants.



En tout premier j’initialise un tableau avec les jours par mois. Je sépare la date de l’annonce afin d’avoir dans trois variables le jour, le mois et l’année. Je vérifie que l’année n’est pas une année bissextile si oui je rajoute un jour au mois de février.

Ensuite j’additionne de 15 ma variable jour. Si ma variable jour dépasse le nombre de jours total du mois en cour, j’additionne le mois et je soustrais les jours.

Par exemple :

23 janvier + 15 = 38 janvier (date impossible)

Alors je soustrait le nombre de jour de janvier (31) et j’additionne le mois.

38 janvier – 31 = 7 et j’additionne les mois ce qui devient février.

La date butoir de l’annonce est alors le 7 février.

La première partie de la fonction est faite, on sait jusqu'à quand l’annonce est valide. Maintenant il faut savoir combien de jours cela représente. Pour se faire il faut regarder le code qui suit.



Je crée une nouvelle date avec les nouvelles infos que j’ai. Ensuite je récupère la date d’aujourd’hui. L’objet date mets a disposition une fonction qui permet de soustraire deux date. Cette fonction s’appelle « diff ». J’utilise donc cette fonction pour soustraire la date butoir avec la date d’aujourd’hui. Ensuite je récupère cette différence dans une variable toujours de type date, et j’utilise une autre fonction qui permet de retourner différentes choses. Moi je l’utilise pour qu’il me retourne le nombre de jours entre les deux dates. Tout d’abord je test si les jours son supérieur ou égaux a 0 si non l’annonce est dépassée et donc expirée. Si au contraire le résultat est positif, je reteste le nombre de jours afin d’afficher différent messages par apports au nombre de jours restants.

A la fin je retourne un tableau à deux cases qui en [0] retourne le message par apport aux jours et en [1] je retourne true ou false. True si l’annonce est valide False si l’annonce est expirée.

L’autre fonction sur laquelle je suis resté pas mal de temps et que j’ai en de la peine a faire, c’est la fonction qui permet de mettre le texte en que l’on recherche en rouge. Il existe plusieurs fonctions sur internet qui font ça, mais j’ai préféré faire ma fonction moi même.

Cette fonction prend deux paramètres, $mot\_rechercher et $plage. $mot\_rechercher comme son nom l’indique, stockera le mot a rechercher et $plage est le texte dans laquelle il faut rechercher.

Je vais utilisé quelques variables, $array\_tmp sera un tableau temporaire qui stockera l’évolution de la plage. J’expliquerais son fonctionnement un peu plus tard. $index servira d’index a tableau\_tmp, $count\_char est une variable de type int qui va être incrémentée, elle servira pour tester si on a trouver le mot ou pas, $b\_fin est une variable booléenne qui servira pour sortir de la boucle while.



Tout d’abord j’initialise mes variables, et je met dans mon $array\_tmp en indice 0 la plage de recherche. En tout premier je test si le mot que je recherche contient contient uniquement des chiffres, si non on continue si oui le programme va planter alors on quitte. Ensuite on cherche a savoir si le mot contient plus d’un caractère, comme pour avant si oui on continue si non on quitte car le programme ne fonctionnera pas. En suite on boucle, j’instancie une boucle while avec une variable booléenne. Tant que la variable vaut true, la boucle continue. Dans cette boucle on instancie une autre boucle mais une boucle for cette fois. La boucle for va lire chaque caractère de la plage de recherche. Et on test, si le caractère lu ($array\_tmp[$index][$y]) est égale au premier caractère du mot rechercher.

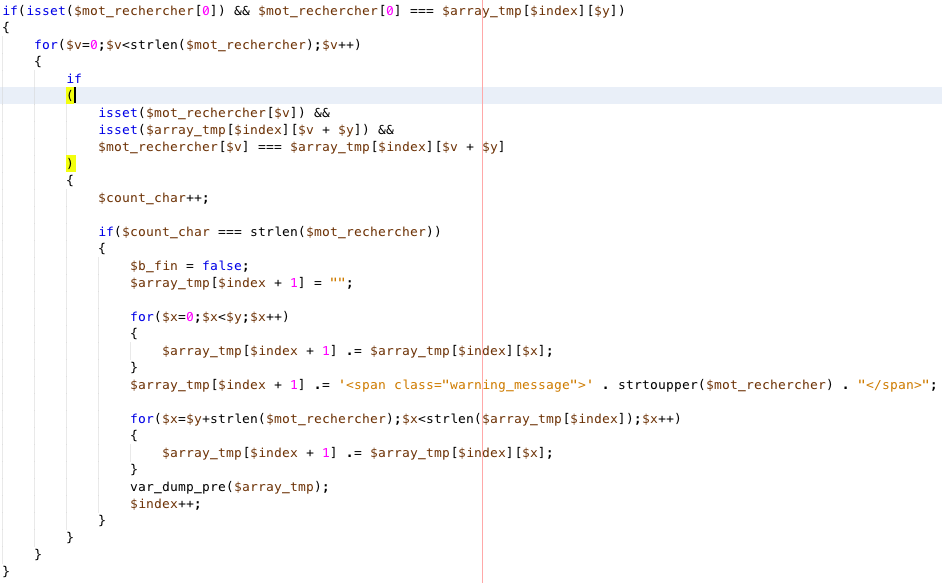
Par exemple je recherche le mot « vendre » dans la plage MT-07 a vendre.

La boucle for va lire chaque caractère de la plage

(‘M’, ‘T’,’-‘,’0’,’7’,’ ‘,’a’,’ ’,’v’,’e‘,’n’,’d’,’r’,’e’) des que le caractère lu par la boucle est égale au premier caractère rechercher

[v]endre == MT-07 a [v]endre

il va entrer dans le if.



Une fois dans le if une autre boucle for va regarder si les caractères qui suivent le « V » sont identiques au mot rechercher. Si oui on incrémente $count\_char si non on passe sans rien faire. On répète cette étape pour chaque caractère. Si count char vaut la même valeur que la longueur du $mot\_rechercher cela veut dire qu’on a trouver le mot.

Si le mot a été trouvé on arrête la boucle while, on prend notre $array\_tmp et on lui ajoute une case, dans cette case on recopie le texte qu’il y a avant la première lettre a chercher.

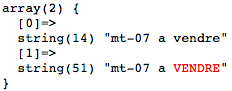
« MT-07 a  »

Ensuite on insère une balise span qui permettra de mettre le texte en rouge. Et on recopie le texte a rechercher

Vendre

On referme la balise span et pour finir on ajoute le reste du texte.

La variable $array\_tmp fonctionne ainsi, dans la première case il y a la plage de recherche et a chaque fois que l’on trouve le mot que l’ont recherche on ajoute une case au tableau. Dans cette case on recopie la partie de la plage qui nous intéresse pas, et on formate le texte rechercher et on finit la phrase.



On voit bien que la première fois on a un texte normal a qui est écrit. Et que la fois d’après le texte est formaté. Cette fonction retourne un string. Le string retourner est la dernière case du tableau $array\_tmp.

## Stockage des photos

Pour savoir si une annonce dispose de photo ou pas il y a une valeur booléenne dans la table. Mais rien n’indique comment sont stockée les photos.

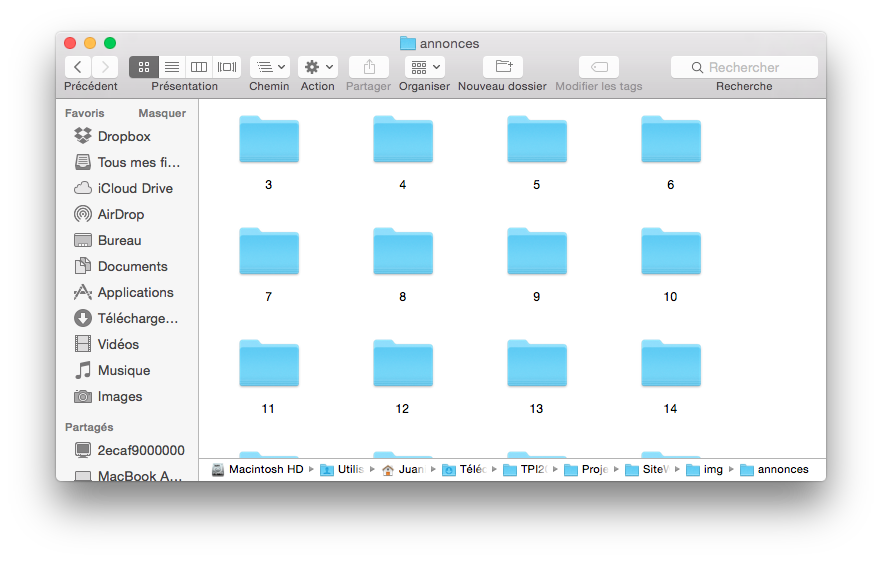
#### Pour les annonces

Les photos stockées pour les annonces sont disponible dans un dossier sur le serveur. C’est a dire que s’il faut changer de serveur, il ne faut pas oublier de copier les photos sur le nouveau serveur.

Le chemin du dossier est

/img/annonces/[idAnnonce]/0.ext

On peut remarquer plusieurs choses dans cette ligne, déjà j’utilise l’id de l’annonce comme dossier



Cette façon me permet de récupérer les images facilement de plus je suis sur que deux dossier n’auront pas le même nom.

La deuxième chose que l’ont peut remarquer dans la ligne juste au dessus, c’est que le nom de la photo commence par 0. La raison est que j’utilise une boucle for pour afficher les photos en miniature. Et donc le nom de la photo se doit d’être unique pour afficher la bonne photo et je me sert de l’indice de boucle comme nom de photo. Une autre méthode possible aurait été de récupérer tous les nom du dossier dans un tableau et d’utiliser leur nom. Pour terminer j’utilise la fonction changer\_format pour mettre la bonne extension de photo. Cette fonction récupère ce que la variable globale $\_FILE renvois comme type de fichier et le traite afin d’en faire une extension.

Si l’annonce ne possède aucune image, uneimage par default sera séléctionnée. Cette image se trouve sur le serveur a l’adresse suivante

/img/image\_site/ No\_Image\_Available.png

#### Pour les avatars

Pour les avatars c’est un peu la même chose. J’utilise aussi l’id de l’utilisateur comme nom de dossier. Le changement comparé a avant est que le nom de la photo est avatar.ext (remplacer ext par la bonne extension).

J’utilise la même fonction qu’avant pour savoir l’extension.

Ses images se trouvent dans le dossier

/img/avatar/[idUtilisateur]/avatar.ext

si la valeur booléenne dans la table, qui indique si l’utilisateur utilise un avatar vaut false, une valeur par default sera séléctionnée. La valeur par default est une image montrant une silhoute. Cette image est disponible apres le chemin suivant.

/img/image\_site/avatar.jpeg

# Est-ce que tout fonctionne ?

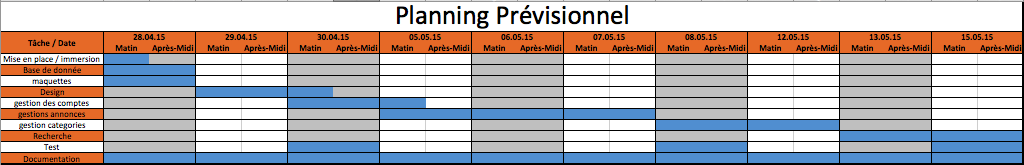
## Plan de test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Plan de test | | | | |
| N° Test | **Auteur** | **Scénario** | **Résultat attendu** | **Ok / KO** |
| Toutes les pages | | | |  |
| T001 | Antonio Pisanello | si on est connécter les liens s'affichent sur toutes les tables | les liens s'affichent sur toutes les tables |  |
| T002 | Antonio Pisanello | si on est admin lien menu admin | menu déroulant admin |  |
| T003 | Antonio Pisanello | si aucune categorie existe | affichage d'un msg |  |
| T004 | Antonio Pisanello | si une ou plusieurs categorie existe | affichage des catégorie |  |
| T005 | Antonio Pisanello | si des annonces existent | affichage du nombre d'annonce a coté de la catégorie |  |
| T006 | Antonio Pisanello | pas le droit d'aller sur des pages auxquels on a pas l'autorisation | redirection et msg d'erreur |  |
| Page Index | | | |  |
| T007 | Antonio Pisanello | si aucune annonce existe | un message doit apparaître |  |
| T008 | Antonio Pisanello | si une ou plusieurs annonces existe | Affichage des 4 dèrnieresannonces |  |
| T009 | Antonio Pisanello | login correctement | redirection page index |  |
| Page connexion | | | |  |
| T010 | Antonio Pisanello | pseudo ou mdp faux | message d'erreur et mémoire du pseudo |  |
| T011 | Antonio Pisanello | pseudo ou mdp non entré | message d'erreur |  |
| Page inscription | | | |  |
| T012 | Antonio Pisanello | si un champs du formulaire n'est pas remplis | message d'erreur |  |
| T013 | Antonio Pisanello | si l'adresse email n'est pas valide | message d'erreur |  |
| T014 | Antonio Pisanello | si le pseudo est deja utilisé | message d'erreur |  |
| T015 | Antonio Pisanello | si l'email est deja utilise | message d'erreur |  |
| T016 | Antonio Pisanello | inscription correcte | redirection a la page index et connexion directe |  |
| T017 | Antonio Pisanello | si les mots de passe ne concordent pas | message d'erreur |  |
| page Menu annonce | | | |  |
| T018 | Antonio Pisanello | si on a pas d'annonce | affichage d'un msg |  |
| T019 | Antonio Pisanello | si aucune photo est selectionner | photo par default |  |
| T020 | Antonio Pisanello | si on supprime une annonce | confirmation de suppression |  |
| T021 | Antonio Pisanello | si on ajoute une annonce | active |  |
| Page afficher annonce | | | |  |
| T022 | Antonio Pisanello | si l'annonce eest expirée | affichage d'un msg |  |
| T023 | Antonio Pisanello | si l'annonce n'existe pas | affichage d'un msg |  |
| T024 | Antonio Pisanello | si l'annonce est expirée mais qu'on y accede depuis gestion annnonce ou menu annonce | Affichage de l'annonce |  |
| T025 | Antonio Pisanello | s'il n'y a pas de photo | affichage d'une photo par default |  |
| T026 | Antonio Pisanello | si des photos existent | affichage des photos |  |
| Page modifier annonce | | | |  |
| T026 | Antonio Pisanello | si l'annonce n'existe pas | affichage d'un msg |  |
| T027 | Antonio Pisanello | si on ajouter es photos | les photos s'ajoutent |  |
| T028 | Antonio Pisanello | si on supprime des photos | les photos se suppriment |  |
| Page gestion compte | | | |  |
| T029 | Antonio Pisanello | si on a un avatar | affichage de son avatar |  |
| T030 | Antonio Pisanello | si on a pas d'avatar | affichage d'une photo par default |  |
| T031 | Antonio Pisanello | recuperation du pseudo et maisl dans la base | affichage en tant que placeholder |  |
| T032 | Antonio Pisanello | si on supprime l'avatar | affichage d'une photo par default |  |
| T033 | Antonio Pisanello | si on change l'avatar | modification de l'avatar |  |
| Page gestion utilisateur | | | |  |
| T034 | Antonio Pisanello | si l'utilisateur a un avatar | affichage de son avatar |  |
| T035 | Antonio Pisanello | si l'utilisateur n'a pas d'avatar | affichge d'une photo par default |  |
| T036 | Antonio Pisanello | si l'utilisateur a poster des annonces | affichages du nombre d'annonces qu'il a poster |  |
| T037 | Antonio Pisanello | suppression d'un utilisateur | confirmation de suppression |  |
| T038 | Antonio Pisanello | si l'utilisateur n'a pas poster d'annonce | affichage du nombre d'annonce postée (0) |  |
| Page gestion annonces | | | |  |
| T039 | Antonio Pisanello | si une annonce est active mais pas valide | affichage des annonces |  |
| T040 | Antonio Pisanello | si l'annonce contient une photo | affichage de la photo |  |
| T041 | Antonio Pisanello | si l'annonce contient pas de photos | affichage d'une photo par default |  |
| T042 | Antonio Pisanello | suppression de l'annonce | confirmation de suppression |  |
| T043 | Antonio Pisanello | quand on clique dessus | Affichage de l'annonce |  |
| Page gestion catégories | | | |  |
| T044 | Antonio Pisanello | affichage de toutes les catégories | affichage de toutes les catégories |  |
| T045 | Antonio Pisanello | suppression de la catégorie | confirmation de suppression |  |
| T046 | Antonio Pisanello | si la catégorie contient un descriptif | affichage du descriptif |  |
| Page recherche | | | |  |
| T047 | Antonio Pisanello | si on recherche des chiffres | pas pris en compte |  |
| T048 | Antonio Pisanello | quand on ecrit on affiche deja quelques annonces en ajax | affichage des annonces en ajax |  |

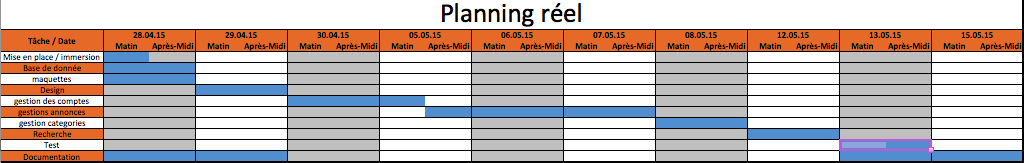
## Rapports de tests

# Planning

## Planning prévisionnel



## Planning réel



# Conclusion

## Quelles sont les différences entre ce que je voulais faire et ce que j’ai fait ?

Une chose qui ne figurait pas sur le cahier des charges mais que j’ai rajouté, est la recherche ajax. Quand on commence a écrire dans le champs de recherche d’une annonce, une lise d’annonces correspondantes a ce que l’on vient d’écrire apparaît. Cette fonctionnalité a été rajoutée après coup, car j’ai eu un peu d’avance. J’avais appris l’ajax en cour M306 afin de le mettre dans mon TPI si j’e avais le temps. J’ai eu le temps car j’ai fait plus vite que prévu la recherche et tout ce qui était en rapport avec les annonces. J’ai donc pu prendre ce temps pour créer ma fonction ajax et l’intégrer au site.

## Quelles ont été les difficultés rencontrées ?

J’ai rencontré des difficultées lors de la création de la fonction de mise en gras du texte. Comme je l’ai dit plus c’est la fonction la plus complexe de mon application. J’ai pris beaucoup de temps a réfléchir comment j’allais la faire pour qu’elle fonctionne et surtout qu’elle fonctionne rapidement. J’aurais très bien pu aller chercher un code tout fait sur internet qui attendait juste d’étre copié collé, mais je trouvais important de réussir cette fonction par moi même. De plus je suis sur ainsi de bien comprendre le code. Mis a part les soucis de concéption de cette fonction, j’ai eu d’autres soucis, toujours avec cette fonctoin, mais cette fois en rapport avec les mots que je recherchaient. Si je rentrais que des chiffres la fonction plantait, car pour elle quand elle trouvait un chiffre, elle le prennait pas comme string mais comme int et elle n’arrivait pas a mettre une égalité. Alors j’a’i essayer de mettre 3= a la suite dans le if, ce qui force a tester l’égalité et le fais que les variables doivent être du même type. Mais malgré cette opération mon programme ne voulait pas fonctionner. Donc la derniere solution qui me restait pour traiter ce bug, a été de tester si ce que j’entre comme mot a rechercher contient uniqement des chiffre ou pas. S’il contient uniquement des chiffres je ne rentre pas dasn la fonction s’il en contient pas je peu entrer sans soucis dans la fonciton.

Sinon mis a part ça je n’ai pas eu de soucis lors du dérroulement du TPI, d’ailleur j’avais même de l’avance sur le code source par apport a mon planning.

## Quelles sont les futures améliorations possibles ?

Pour la suite j’imaginais l’intégration de favoris afin de pouvoir garder en mémoire une annonce. Ce serait pas trop difficile a faire, il faudrait ajouter une table favoris avec deux champs, idAnnonce et idUtilisateur.

Une autre amélioration possible serait que quand on est dans gestion utilisateur et que l’on clique sur utilisateur, on devrait pouvoir voir les annonces qu’il a posté afin de voir si l’utilisateur poste des bêtises ou des choses hors sujet, par exemple si l’utilisateur vend un service et non un objet.

Intégrer l’envoie de mail depuis le serveur, pour que se soit une confirmation d’inscription, notification pour un suppression d’annonce par l’admin ou le fait d’avoir rendu l’annonce active. La dernière possibilité serait dans afficher annonce d’envoyer un mail directement sans devoir utiliser une adresse mail externe.

Une autre chose serait comme sur anibis.ch, de pouvoir avoir des sous catégories et des infos plus précises par catégories. Je pense que le fait d’avoir plus d’infos sur le vendeur adresse, numéros de téléphone serait une amélioration conséquente.

Une autre chose qui pourrait rendre ce site plus lucratif serait de changer le temps et de vendre des offres plus longues.

Je pense que de revoir l’ergonomie serait pas du luxe, je dois avouer que j’ai un peu laisser l’ergonomie de coté. Le fait d’aller dans le menu annonce pour ajouter une annonce ou voir les annonces que l’on a déjà ajouté n’est pas top. Je verrais bien un lien menu annonce et quand on va dessus un menu déroulant avec deux liens s’afficheraient, insérer annonce et voir vos annonces. Ainsi chaque fonctionnalité serait sur une page différente.

## Que m’a apporté ce projet ?

Ce projet m’a apporté plus de confiance en moi. J’ai été capable de réalisé plusieurs choses que je me jugeais incapable de faire telles que la réalisation d’un site complet en 10 jours seulement. De plus j’ai été impressionné par ma propre implication dans ce travail, en effet, je ne m’étais jamais autant investi dans un travail comme je l’ai fait durant ce TPI.

Maintenant je réalise aussi que parfois l’essentiel n’est pas de faire les choses le mieux possible mais de faire ce qui nous est demandé en priorité avant de commencer à perdre du temps sur des détails qui peuvent paraître parfois futiles.

La réalisation de ce projet n’a fait que m’ouvrir les yeux et réalisé que durant ces quatre années j’ai non seulement appris à programmer mais j’ai aussi acquis des façons de penser et de travailler différentes qui me permettent aujourd’hui de travailler de manière beaucoup plus efficace. Même si toutefois, je sais qu’il me reste encore beaucoup à apprendre et beaucoup d’expérience à acquérir avant de devenir un programmeur émérite.

1. Source : http://www.internet-marketer.ch/blog/articles/36-anibis-petites-annonces-interview [↑](#footnote-ref-1)