Documentation Technique

Direct Prod

01 mai 2019

CFPT Informatique – Travail Pratique Individuelle

Romain Peretti – IDA-P4A

Maître d’apprentissage : Jossi Sébastien

Experts : Fontaine Philippe & Aegerter Fredy

Table des matières

[1. Introduction 4](#_Toc9453885)

[1.1 Généralités 4](#_Toc9453886)

[1.2 Pourquoi ce projet 4](#_Toc9453887)

[2. Cahier des charges 4](#_Toc9453888)

[2.1 Objectif 4](#_Toc9453889)

[2.2 Tâches & Fonctionnalités 4](#_Toc9453890)

[2.3 Matériels & logiciels 4](#_Toc9453891)

[3. Analyse fonctionnelle 5](#_Toc9453892)

[3.1 Barre de navigation 5](#_Toc9453893)

[3.1.1 Barre de navigation non connecté 5](#_Toc9453894)

[3.1.2 Barre de navigation connecté 5](#_Toc9453895)

[3.1.3 Barre de navigation de l’administrateur 5](#_Toc9453896)

[3.2 Créer un compte 6](#_Toc9453897)

[3.3 Connexion 7](#_Toc9453898)

[3.4 L’accueil 7](#_Toc9453899)

[3.4.1 L’accueil non connecté 7](#_Toc9453900)

[3.4.2 L’accueil connecté 8](#_Toc9453901)

[3.5 Détails d’une annonce 9](#_Toc9453902)

[3.5.1 Détails d’une annonce non connecté 9](#_Toc9453903)

[3.5.2 Détails d’une annonce connecté 10](#_Toc9453904)

[3.6 Création d’une annonce 11](#_Toc9453905)

[3.7 Modification d’une annonce 12](#_Toc9453906)

[3.8 Suppression d’une annonce 13](#_Toc9453907)

[3.9 Détails de notre profil 14](#_Toc9453908)

[3.10 Modification de notre profil 15](#_Toc9453909)

[3.11 Suppression de notre profil 17](#_Toc9453910)

[3.12 Admin 17](#_Toc9453911)

[4. Analyse organique 18](#_Toc9453912)

[4.1 Base de données 18](#_Toc9453913)

[4.1.1 Modèle conceptuel proposé 18](#_Toc9453914)

[4.1.2 Modèle conceptuel modifié 18](#_Toc9453915)

[4.1.3 Modèle physique 19](#_Toc9453916)

[4.1.4 Détails des tables 20](#_Toc9453917)

[4.1.4.1 Table « users » 20](#_Toc9453918)

[4.1.4.2 Table « advertisements » 20](#_Toc9453919)

[4.1.4.2 Table « rates » 20](#_Toc9453920)

[4.1.4.2 Table « pictures » 20](#_Toc9453921)

[4.1.4.2 Table « roles » 20](#_Toc9453922)

[4.2 Architecture du code 21](#_Toc9453923)

[4.2.1 Arborescence de fichier 21](#_Toc9453924)

[4.2.2 Les classes 21](#_Toc9453925)

[4.2.2.1 Classe « User » 21](#_Toc9453926)

[4.2.2.2 Classe « Advertisement » 22](#_Toc9453927)

[4.2.2.3 Classe « Rating » 22](#_Toc9453928)

[4.2.2.4 Classe « Picture » 22](#_Toc9453929)

[4.2.2.5 Classe « Database » 22](#_Toc9453930)

[4.2.3 Les manageurs 23](#_Toc9453931)

[4.2.3.1 Classe « UserManager » 23](#_Toc9453932)

[4.2.3.2 Classe « AdvertisementManager » 24](#_Toc9453933)

[4.2.3.3 Classe « RatingManager » 25](#_Toc9453934)

[4.2.3.4 Classe « PictureManager » 25](#_Toc9453935)

[5. Outils externes 26](#_Toc9453936)

[5.1 Bootstrap 26](#_Toc9453937)

[5.2 GitHub 26](#_Toc9453938)

[5.3 Font Awesome 26](#_Toc9453939)

[5.4 SweetAlert 27](#_Toc9453940)

[6. Installation 27](#_Toc9453941)

[7. Tests 27](#_Toc9453942)

[7.1 Page d’inscription 27](#_Toc9453943)

[7.2 Page de connexion 27](#_Toc9453944)

[7.3 Page d’accueil 28](#_Toc9453945)

[7.4 Page de création d’une annonce 29](#_Toc9453946)

[7.5 Page de modification d’une annonce 29](#_Toc9453947)

[7.6 Page de suppression d’une annonce 30](#_Toc9453948)

[7.7 Page de profil 30](#_Toc9453949)

[7.8 Page de modification de son profil 31](#_Toc9453950)

[7.9 Page de suppression de son profil 31](#_Toc9453951)

[7.10 Page d’administration 31](#_Toc9453952)

[8. Améliorations possibles 32](#_Toc9453953)

[9. Conclusion 33](#_Toc9453954)

[9.1 Difficultés rencontrées 33](#_Toc9453955)

[9.2 Bilan personnel 33](#_Toc9453956)

[9.3 Ce que ce projet m’a apporté 33](#_Toc9453957)

[10. Bibliographie 34](#_Toc9453958)

[10.1 Codes repris 34](#_Toc9453959)

[10.2 Sites utilisés 34](#_Toc9453960)

[11. Planning 0](#_Toc9453961)

[11.1 Planning prévisionnel 0](#_Toc9453962)

[11.2 Planning effectif 1](#_Toc9453963)

[11.3 Différence entre les plannings 0](#_Toc9453964)

[12. Table des illustrations 0](#_Toc9453965)

[12.1 Figures 0](#_Toc9453966)

[12.2 Tableaux 0](#_Toc9453967)

[13. Annexes 1](#_Toc9453968)

# 1. Introduction

## 1.1 Généralités

Arrivant à la fin de ma formation d’informaticien, je dois fournir un Travail Pratique Individuelle (TPI) pour pouvoir obtenir mon certificat d’informaticien. J’ai choisi de réaliser un projet web et l’on m’a proposé Direct Prod.

## 1.2 Pourquoi ce projet

Début 2019 on m’a demandé quel type de projet je voulais réaliser pour mon TPI, j’ai décidé de faire un site web car durant ma troisième j’ai principalement fait du web, laissant le C# de côté. Par la suite, n’ayant pas d’idée de projet précis à faire j’ai préféré laisser mon maître d’apprentissage choisir le site, sachant que ce serai sans doute un site de vente.

# 2. Cahier des charges

## 2.1 Objectif

L’objectif de ce projet web est de réaliser une plateforme d’annonces pour des produits de consommation, où les utilisateurs pourront voir, évaluer et créer des annonces.

## 2.2 Tâches & Fonctionnalités

Pendant ce projet, j’ai plusieurs tâches et fonctionnalités à développer. Pour ce qui est des tâches les voici :

* Réalisation des maquettes du site
* Création de la base de données
* Réalisation des pages du site
* Élaboration d’un rapport de test
* Réalisation d’une documentation technique
* Réalisation d’un manuel utilisateur
* Conception d’un journal de bord
* Réalisation d’un résumé du travail

Ensuite, les fonctionnalités :

* Inscription, connexion et déconnexion d’un utilisateur
* Création d’annonce
* Modification d’une annonce
* Suppression d’une annonce
* Modification du profil
* Suppression du profil
* Évaluation d’une annonce
* Validation d’une annonce
* Recherche des annonces

## 2.3 Matériels & logiciels

Pour la réalisation de ce projet j’ai eu à ma disposition ce matériels :

* 1 ordinateur de l’école
* 2 écrans
* 1 journal de bord vierge

Puis, comme logiciel j’ai utilisé :

* Apache (v. 2.4.28)
* Php (v. 7.0)
* MySQL Workbench (v. 8.0)
* Visual Studio Code
* Pencil
* Bootstrap (v. 4.2.1)
* Microsoft Office 2016

# 3. Analyse fonctionnelle

## 3.1 Barre de navigation

Dans mon site il y aura plusieurs barres de navigation suivant le rôle de l’utilisateur.

### 3.1.1 Barre de navigation non connecté

C

C

A

B



Figure : Barre de navigation d’un utilisateur non connecté

A : Ce lien redirige l’utilisateur vers la page d’accueil (voir Fig. 6).

B : Ce lien redirige l’utilisateur vers la page de connexion (voir Fig. 5).

C : Ce lien redirige l’utilisateur vers la page d’inscription (voir Fig. 4).

### 3.1.2 Barre de navigation connecté

C

C

B

A



Figure : Barre de navigation d’un utilisateur connecté

A : Ce lien redirige l’utilisateur vers la page d’accueil (voir Fig. 7).

B : Ce lien redirige l’utilisateur vers la page de son profil (voir Fig. 13).

C : Ce lien permet la déconnexion de l’utilisateur.

### 3.1.3 Barre de navigation de l’administrateur

D

C

C

C

B

A



Figure : Barre de navigation de l’administrateur

A : Ce lien redirige l’administrateur vers la page d’accueil (voir Fig. 7).

B : Ce lien redirige l’administrateur vers la page de son profil (voir Fig. 13).

C : Ce lien redirige l’administrateur vers la page d’administration (voir Fig. 16).

D : Ce lien permet la déconnexion de l’administrateur.

## 3.2 Créer un compte

Sur cette page l’utilisateur peut s’inscrire au site. Tous les champs sont obligatoires. Les seules règles à respecter pour pouvoir se créer un compte sont :

* L’email ne doit pas déjà être utilisé
* Les deux mots de passe doivent être égaux

I

H

G

F

E

D

C

B

A

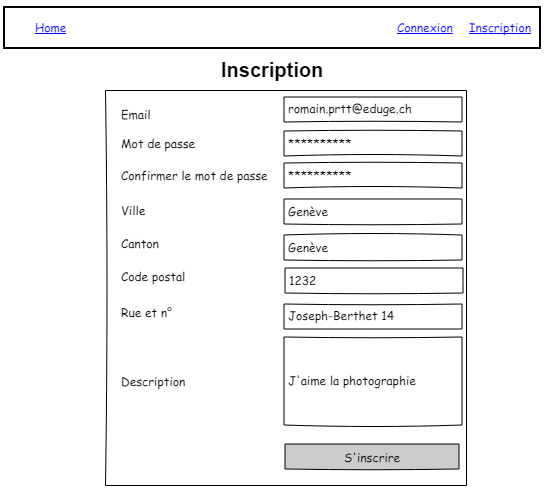


Figure : Créer un compte

A : Email de l’utilisateur

B : Mot de passe de l’utilisateur

C : Confirmation du mot de passe

D : Ville de résidence de l’utilisateur

E : Canton de résidence de l’utilisateur

F : Code postal de l’utilisateur

G : Rue et numéro de résidence de l’utilisateur

H : Description de l’utilisateur

I : Bouton validant l’inscription de l’utilisateur.

## 3.3 Connexion

Cette page permet à un utilisateur de se connecter au site, pour cela il doit spécifier son email et son mot de passe.

B

A

C

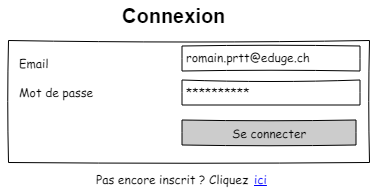


Figure : Connexion

A : Email de l’utilisateur

B : Mot de passe de l’utilisateur

C : Bouton de connexion

D : Lien redirigeant sur la page de création d’un compte (voir Fig. 4).

## 3.4 L’accueil

### 3.4.1 L’accueil non connecté

L’accueil est la page principale de mon site, les annonces affichées sont uniquement celles validées par l’administrateur. On peut la consulter sans être connecté mais la fonctionnalité de création d’une annonce est désactivée.

B

A

C

C1



Figure : Page d’accueil d’un utilisateur non connecté

A : Boite de saisie pour faire une recherche dans les annonces.

B : Lance la recherche.

C : Affichage d’une annonce avec ses détails importants.

C1 : Bouton redirigeant vers la page des détails de l’annonce (voir Fig. 8)

### 3.4.2 L’accueil connecté

Lorsqu’on est connecté le bouton pour créer une annonce apparait (Point A).

De plus, si une annonce nous appartient (Point B) deux boutons supplémentaires apparaissent sur l’annonce (Point B2, B3).

BV

A

C

D3

D2

D1

D

E



Figure : Page d’accueil d’un utilisateur connecté

A : Boite de saisie pour faire une recherche dans les annonces.

B : Lance la recherche.

C : Bouton permettant de créer une annonce (voir Fig. 10).

D : Affichage d’une annonce m’appartenant, c’est pourquoi deux nouveaux boutons apparaissent.

D1 : Bouton permettant d’être redirigé vers la page des détails de l’annonce (voir Fig. 9)

D2 : Redirection vers la page de modification de l’annonce (voir Fig. 11).

D3 : Redirection vers la page de suppression (voir Fig. 12).

E : Affichage d’une annonce avec ses détails importants.

## 3.5 Détails d’une annonce

Cette page affiche toutes les informations ainsi que les images liées à l’annonce.

### 3.5.1 Détails d’une annonce non connecté

B

A1

A

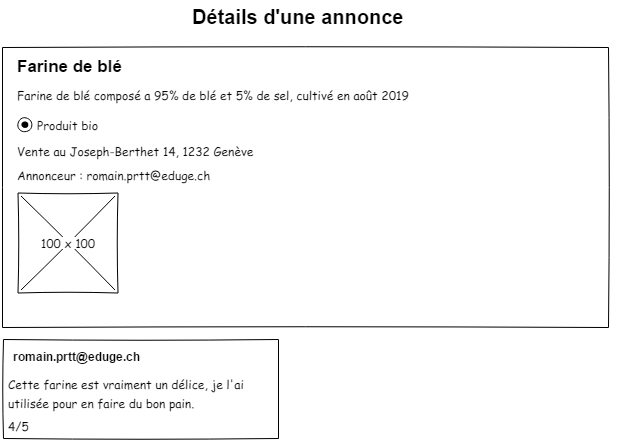


Figure : Détails d’une annonce avec un utilisateur non connecté

A : Description de l’annonce avec affichages des images liées.

A1 : Affichage des images liés en miniature, si on clique sur l’image elle s’ouvre en grand.

B : Commentaire et note laisser par un utilisateur.

### 3.5.2 Détails d’une annonce connecté

A

B

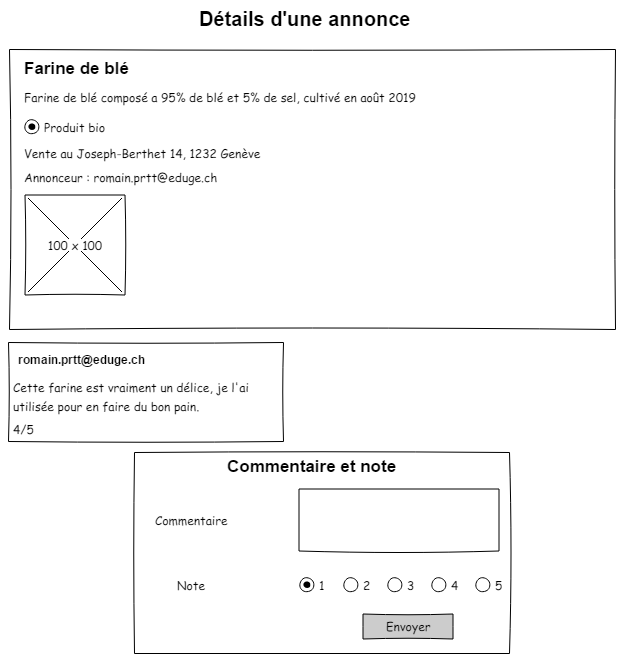


Figure : Détails d’une annonce avec un utilisateur connecté

A : Formulaire permettant de laisser un commentaire et une note à l’annonce.

B : Confirme l’envoie du commentaire et de la note dans la base de données.

## 3.6 Création d’une annonce

Lors de l’appui sur le bouton de création d’une annonce, l’utilisateur est redirigé vers un formulaire. Les champs « Titre », « Description » et « Bio » sont obligatoires, mais il n’est pas nécessaire de mettre une image pour pouvoir créer une annonce.

D

C

B

A

E

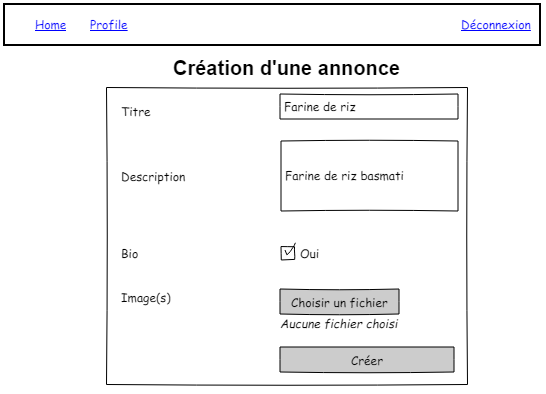


Figure : Formulaire de création d’une annonce

A : Le titre de l’annonce

B : La description du produit

C : Case à cocher si le produit est bio

D : L’utilisateur peut choisir une ou plusieurs images pour son annonce

E : Bouton Confirmant la création de l’annonce

## 3.7 Modification d’une annonce

Lors de la modification d’une annonce tous les champs du formulaire sont remplis avec les informations de la base de données.

F

E

DA

C

B

A

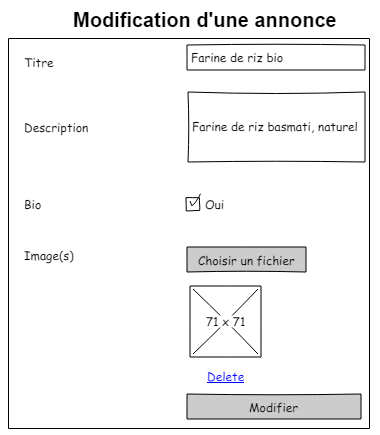


Figure : Modification d’une annonce

A : Le titre de l’annonce

B : La description du produit

C : Case à cocher si le produit est bio

D : L’utilisateur peut choisir une ou plusieurs images pour son annonce

E : Les images anciennement choisies sont affichées avec un lien supprimant l’image.

F : Confirme la modification des informations dans la base de données.

## 3.8 Suppression d’une annonce

La page de suppression d’une annonce est surtout une confirmation de la part de l’utilisateur qu’il veut bel et bien supprimer cette annonce.

B

A

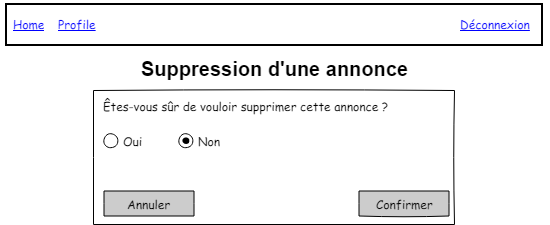


Figure : Confirmation de la suppression d’une annonce

A : Redirige vers la page d’accueil.

B : Supprime votre compte de la base de données et redirige vers la page d’accueil.

## 3.9 Détails de notre profil

Cette page montre les détails de notre profil ainsi que toutes les annonces créées.

B3

B2

B1

A2

A1

B

A

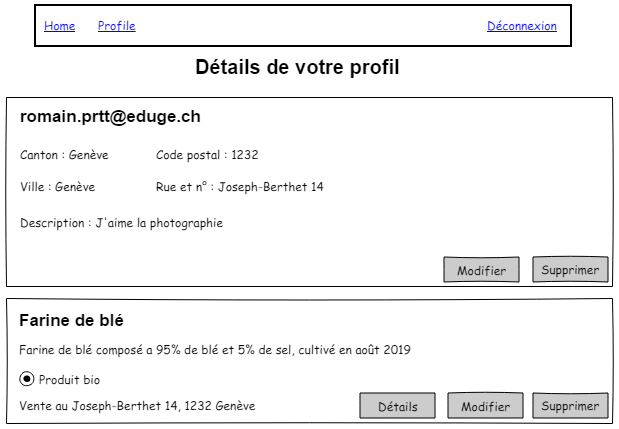


Figure : Détails de notre profil

A : Description de l’utilisateur.

A1 : Redirige vers la page de modification de son profil (voir Fig. 14).

A2 : Redirige vers la page de suppression de son compte (voir Fig. 15).

B : Affichage des annonces nous appartenant.

B1 : Bouton permettant d’être redirigé vers la page des détails de l’annonce (voir Fig. 9)

B2 : Redirection vers la page de modification de l’annonce (voir Fig. 11).

B3 : Redirection vers la page de suppression (voir Fig. 12).

## 3.10 Modification de notre profil

Lors de la modification de notre profil tous les champs du formulaire sont remplis avec les informations de la base de données sauf pour les mots de passe.

L’email n’est pas modifiable car il sert d’identifiant pour l’utilisateur. Les champs « Nouveau mot de passe » et « Confirmer le nouveau mot de passe » ne sont pas obligatoire, si l’on ne veut pas changer de mot de passe. Pour confirmer le changement des informations l’utilisateur doit taper son mot de passe actuel dans le champ à cet effet et cliquer sur le bouton « Modifier ».

I

H

GA

E

F

D

C

B

A

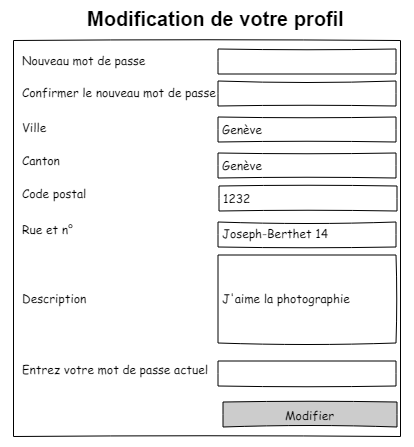


Figure : Modification de notre profil

A : Le nouveau mot de passe de l’utilisateur

B : La confirmation du nouveau mot de passe de l’utilisateur

C : Ville de résidence de l’utilisateur

D : Canton de résidence de l’utilisateur

E : Code postal de l’utilisateur

F : Rue et numéro de résidence de l’utilisateur

G : Description de l’utilisateur

H : Le mot de passe de l’utilisateur

I : Bouton validant la modification des informations de l’utilisateur.

## 3.11 Suppression de notre profil

Pour supprimer son profil, on est redirigé vers une page de confirmation de suppression. Pour confirmer la suppression de son profil il faut cocher le radio bouton « Oui » et appuyer sur le bouton.

B

A

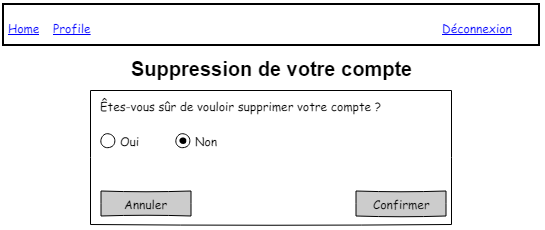


Figure : Confirmation de la suppression de votre compte

A : Redirige vers la page de votre compte.

B : Supprime votre compte de la base de données et redirige vers la page d’accueil.

## 3.12 Admin

Cette page est réservée à l’admin personne d’autre n’y a accès. Elle permet à l’admin de valider ou de refuser les nouvelles annonces que les utilisateurs on créer.

D

C

B

A

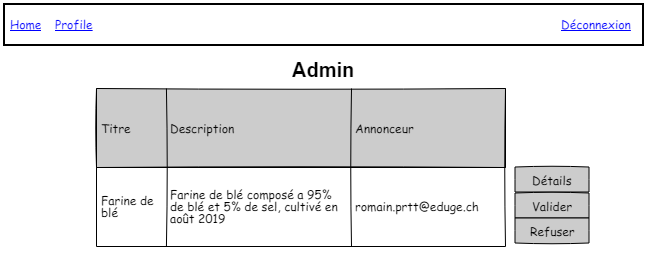


Figure : Administration des nouvelles annonces

A : Informations importantes de l’annonce

B : Redirige vers la page des détails (voir Fig. 9)

C : Valide la nouvelle annonce.

D : Supprime l’annonce de la base de données.

# 4. Analyse organique

## 4.1 Base de données

### 4.1.1 Modèle conceptuel proposé

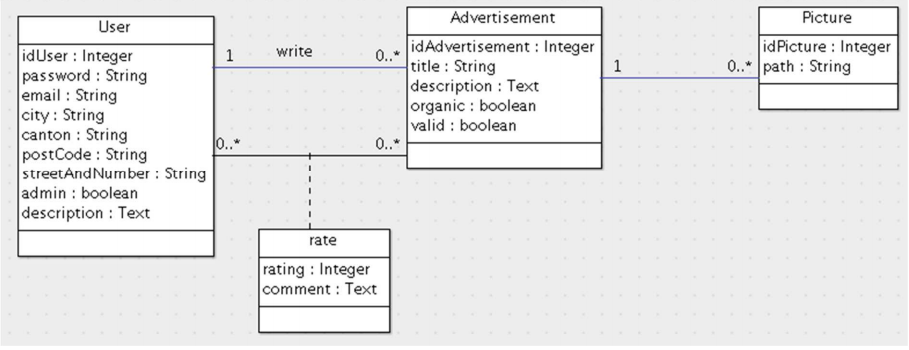


Figure : Modèle conceptuel proposé

### 4.1.2 Modèle conceptuel modifié

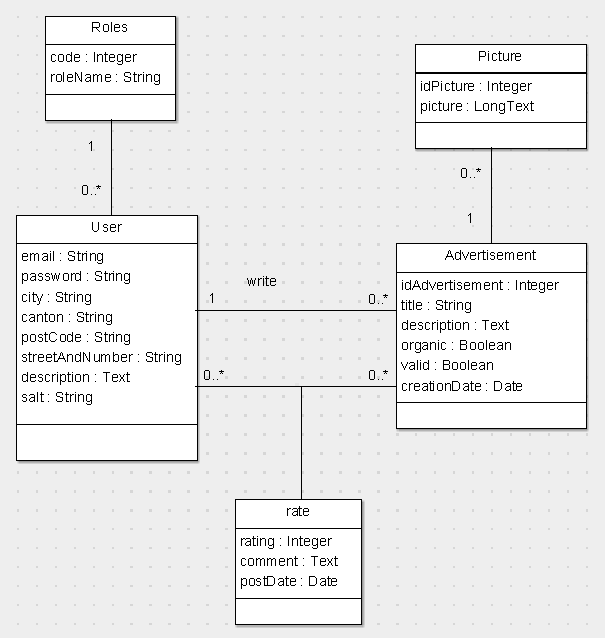


Figure : Modèle conceptuel modifié

### 4.1.3 Modèle physique

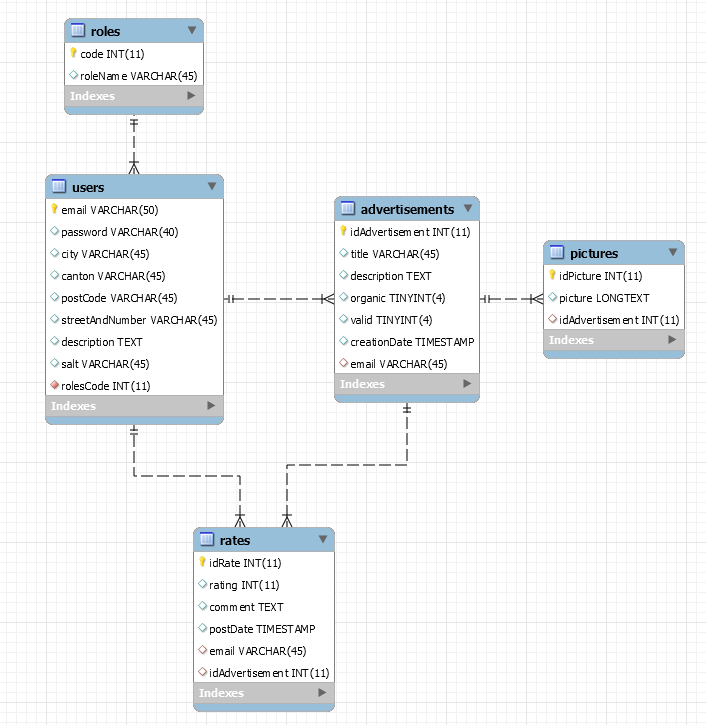


Figure : Modèle physique

### 4.1.4 Détails des tables

|  |  |
| --- | --- |
| Abréviation | Signification |
| PK | Clé primaire |
| NN | Non nul (Empêche la valeur NULL) |
| AI | Auto incrément |
| CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP | Met à jour la date lors de la création et de la modification |

Tableau : Légende des propriétés des attributs

#### 4.1.4.1 Table « users »

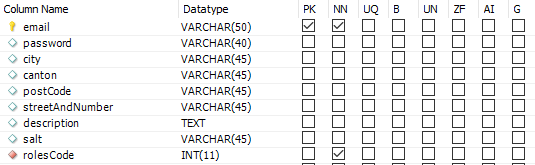


Figure : Table « users »

#### 4.1.4.2 Table « advertisements »

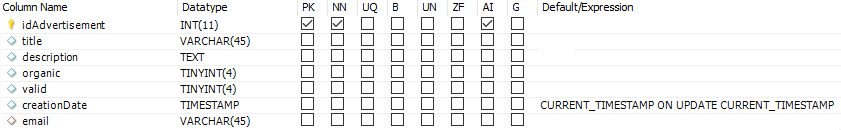


Figure : Table « advertisements »

#### 4.1.4.2 Table « rates »

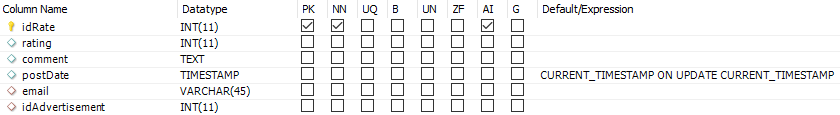


Figure : Table « rates »

#### 4.1.4.2 Table « pictures »

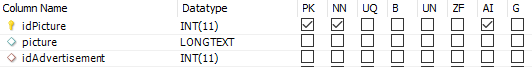


Figure : Table « pictures »

#### 4.1.4.2 Table « roles »



Figure : Table « roles »

## 4.2 Architecture du code

### 4.2.1 Arborescence de fichier

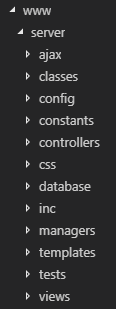


Figure : Arborescence de fichier

* « ajax » contient les fichiers
* « classes » contient les différentes classes et constructeur pour les tables de la base de données.
* « config » contient un fichier avec les variables de connexion à la base de données.
* « constants » contient un fichier avec toutes les variables constantes utilisées sur le site.
* « controllers » contient les fichiers faisant le lien entre les vues et les managers.
* « css » contient le dossier de bootstrap ainsi qu’un CSS personnel.
* « database » contient le fichier de connexion à la base de données.
* « inc » contient les fichiers que j’inclut dans mes pages, comme la barre de navigation ou le pied de page.
* « managers » contient les classes contenant les fonctions avec les requêtes SQL.
* « templates » contient la maquette du site en HTML.
* « tests » contient les fichiers de tests pour chacun des manageurs.
* « views » contient toutes les vues de mon application web.

### 4.2.2 Les classes

#### 4.2.2.1 Classe « User »

Cette classe contient les informations sur un utilisateur

|  |  |  |
| --- | --- | --- |
| Nom de la variable | Type | Description |
| **email** | STRING | L’email de l’utilisateur |
| **password** | STRING | Le mot de passe de l'utilisateur |
| **city** | STRING | La ville de l'utilisateur |
| **canton** | STRING | Le canton de l'utilisateur |
| **postCode** | INT | Le code postal de l'utilisateur |
| **streetAndNumber** | STRING | Le numéro et le nom de la rue de l'utilisateur |
| **description** | STRING | La description de l'utilisateur |
| **salt** | STRING | Le salt du mot de passe de l'utilisateur |
| **roles\_code** | INT | Le code du rôle de l'utilisateur |

Tableau : Variables de la classe « User »

#### 4.2.2.2 Classe « Advertisement »

Cette classe contient les informations sur une annonce

|  |  |  |
| --- | --- | --- |
| Nom de la variable | Type | Description |
| **idAdvertisement** | INT | L'id de l'annonce |
| **title** | STRING | Le titre de l’annonce |
| **description** | STRING | La description de l’annonce |
| **organic** | BOOLEAN | Un bool définissant si l'annonce est organique ou non |
| **valid** | BOOLEAN | Un bool définissant si l'annonce est validé par l'admin |
| **creationDate** | STRING | La date de création de l'annonce |
| **userEmail** | STRING | L'email de l'utilisateur auquel l'annonce est assigné |

Tableau : Variables de la classe « Advertisement »

#### 4.2.2.3 Classe « Rating »

|  |  |  |
| --- | --- | --- |
| Nom de la variable | Type | Description |
| **idRate** | INT | L'id de l'évaluation |
| **rating** | INT | La note donné par l'utilisateur |
| **comment** | STRING | Le commentaire |
| **postDate** | STRING | La date de création de l'évaluation |
| **userEmail** | STRING | L'email de l'utilisateur ayant mis un commentaire |
| **idAdvertisement** | INT | L'id de l'annonce auquel l’évaluation est assignée |

Tableau : Variables de la classe « Rating »

#### 4.2.2.4 Classe « Picture »

|  |  |  |
| --- | --- | --- |
| Nom de la variable | Type | Description |
| **idPicture** | INT | L’id de l’image |
| **Picture** | STRING | L’image en base 64 |
| **idAdvertisement** | INT | L’id de l’annonce auquel l’image est assignée |

Tableau : Variables de la classe « Picture »

#### 4.2.2.5 Classe « Database »

Pour ce travail j’ai utilisé une classe permettant l’accès à la base de données développée par M. Aigroz et déjà utilisé lors de ma troisième année. Cette classe retourne une instance PDO et utilise le Design Pattern Singleton.

|  |  |
| --- | --- |
| Nom de la fonction | Description |
| private **\_\_construct()** | Constructeur privé pour que personne ne puisse créer une nouvelle instance via  ' = new Database();' |
| private **\_\_clone()** | Clone privé pour que personne ne puisse cloner l'instance |
| public static **getInstance()** | Retourne l'instance de la Database ou créer une connexion initiale |
| Final public static **\_\_callStatic($chrMethod,$arrArguments)** | Est lancée lorsque l'on invoque des méthodes inaccessibles dans un contexte statique.  L'argument **$chrMethod** est le nom de la méthode appelée. L'argument **$arrArguments** est un tableau contenant les paramètres passés à la méthode **$chrMethod**. |

Tableau  : Fonctions de la classe « Database »

### 4.2.3 Les manageurs

#### 4.2.3.1 Classe « UserManager »

Les actions que peuvent exécuter les utilisateurs sont gérés par les requêtes SQL ci-dessous.

|  |  |  |
| --- | --- | --- |
| Nom de la fonction | Paramètre(s) | Description |
| **GetUserByEmail** | $email | @brief Retourne un utilisateur en fonction de son email  @param string email L'email de l'utilisateur qu'on recherche.  @return   * [User] L'utilisateur de type User. * False Une erreur est survenue. * Null L'utilisateur n'est pas trouvé. |
| **CreateUser** | $u | @brief Ajoute un utilisateur  @param [User] L'objet User que l'on veut insérer.  @return boolean   * True Ok. * False Une erreur est survenue. * Null L'utilisateur existe déjà. |
| **UpdateUser** | $u | @brief Modifie un utilisateur  @param [User] L'objet User que l'on veut modifier.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **DeleteUser** | $email | @brief Supprime un utilisateur en fonction de son email  @param string email L'email de l'utilisateur qu'on veut supprimer.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **UserExist** | $email | @brief Vérifie si un utilisateur existe déjà avec l'email spécifié  @param string email L'email de l'utilisateur qu'on recherche.  @return boolean   * True L'utilisateur existe. * False Une erreur est survenue. * Null L'utilisateur n'existe pas. |
| **Login** | $email, $password | @brief Essaie de connecter l'utilisateur  @param string email L'email de l'utilisateur.  @param string password Le mot de passe de l'utilisateur.  @return boolean   * True La connexion a réussie. * False Une erreur est survenue. * Null L'utilisateur n'existe pas |

Tableau : Fonctions de la classe « UserManager »

#### 4.2.3.2 Classe « AdvertisementManager »

|  |  |  |
| --- | --- | --- |
| Nom de la fonction | Paramètre(s) | Description |
| **GetValidatedAds** | / | @brief Retourne toutes les annonces validées par l'administrateur  @return array   * Un tableau contenant toutes les annonces de type "Advertisement". * False Une erreur est survenue. |
| **GetInvalidAds** | / | @brief Retourne toutes les annonces non validées par l'administrateur  @return array   * Un tableau contenant toutes les annonces de type "Advertisement". * False Une erreur est survenue. |
| **GetAdById** | $idAd | @brief Retourne l'annonce que l'on recherche par son identifiant  @param int idAd L'id de l'annonce que l'on recherche.  @return [Advertisement]   * L'annonce de type "Advertisement". * False Une erreur est survenue. * Null L'annonce n'est pas trouvée. |
| **GetAdsByUserEmail** | $email | @brief Retourne l'annonce que l'on recherche par l'identifiant de son annonceur  @param string email L'email de l'annonceur.  @return array   * Un tableau contenant toutes les annonces de type "Advertisement". * False Une erreur est survenue. |
| **CreateAd** | $a | @brief Ajout d'une annonce  @param [Advertisement] L'objet "Advertisement" qu'on veut insérer dans la base.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **UpdateAd** | $a | @brief Modification d'une annonce  @param [Advertisement] L'objet "Advertisement" qu'on veut modifier dans la base.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **UpdateAdToValid** | $idAd | @brief Valide une annonce  @param int idAd L'id de l'annonce que l'on veut validée.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **DeleteAd** | $idAd | @brief Suppression d'une annonce  @param int idAd L'id de l'annonce qu'on veut supprimer.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **DeleteAdsOfUser** | $email | @brief Suppression de toutes les annonces d'un utilisateur  @param string email L'email de l'utilisateur.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **Research** | $searchContent, $organic | @brief Retourne les annonces en rapport avec la recherche  @param string searchContent La recherche de l'utilisateur.  @param boolean organic Si l'utilisateur recherche que des annonces bio.  @return array   * Un tableau contenant toutes les annonces de type "Advertisement". * False Une erreur est survenue. |

Tableau : Fonctions de la classe « AdvertisementManager »

#### 4.2.3.3 Classe « RatingManager »

|  |  |  |
| --- | --- | --- |
| Nom de la fonction | Paramètre(s) | Description |
| **GetRatingsOfAnAd** | $idAd | @brief Retourne toutes les commentaires et notes en rapport avec l'annonce spécifié  @param int idAd L'id de l'annonce qu'on recherche  @return array   * Un tableau contenant toutes les commentaires et notes de type "Rating". * False Une erreur est survenue. |
| **GetScoreOfAnAd** | $idAd | @brief Retourne le score donné par tous les commentaires des utilisateurs  @param int idAd L'id de l'annonce qu'on recherche  @return float   * Un nombre avec un chiffre derrière la virgule étant la moyenne de tous les commentaires. * False Une erreur est survenue. |
| **CreateRating** | $r | @brief Ajout d'un commentaire et d'une note  @param [Rating] Le commentaire et la note de type "Rating"  @return boolean   * True Ok. * False Une erreur est survenue. |
| **DeleteRatingsOfAnAd** | $idAd | @brief Suppression de toutes les évaluations d'une annonce  @param int idAd L'id de l'annonce.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **DeleteRatingsOfUser** | $email | @brief Suppression de toutes les évaluations d'un utilisateur  @param string email L'email de l'utilisateur.  @return boolean   * True Ok. * False Une erreur est survenue. |

Tableau : Fonctions de la classe « RatingManager »

#### 4.2.3.4 Classe « PictureManager »

|  |  |  |
| --- | --- | --- |
| Nom de la fonction | Paramètre(s) | Description |
| **GetPictureById** | $idPicture | @brief Retourne l'annonce que l'on recherche par l'identifiant de son annonceur  @param int idPicture L'id de l'image qu'on recherche.  @return [Picture]   * L'image de type "Picture". * False Une erreur est survenue. * Null Aucune image. |
| **GetPicturesByAdId** | $idAd | @brief Retourne les images en rapport avec l'annonce spécifié  @param int idPicture L'id de l'annonce qu'on recherche.  @return array   * Un tableau contenant toutes les images. * False Une erreur est survenue. |
| **CreatePicture** | $p | @brief Ajout d'une image  @param int idAd L'id de l'annonce auquel est lié l'image.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **DeletePicture** | $idPicture | @brief Suppression d'une image  @param int idPicture L'id de l'image à supprimer.  @return boolean   * True Ok. * False Une erreur est survenue. |
| **DeletePicturesOfAnAd** | $idAd | @brief Suppression des images lié à une annonce  @param int idAd L'id de l'annonce.  @return boolean   * True Ok. * False Une erreur est survenue. |

Tableau : Fonctions de la classe « PictureManager »

# 5. Outils externes

## 5.1 Bootstrap

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.[[1]](#footnote-1)

J’ai utilisé bootstrap sur mon site afin d’avoir une interface responsive et de ne pas perdre trop de temps sur le design. J’ai préféré télécharger la version de bootstrap et de la mettre sur mon site.

## 5.2 GitHub

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.[[2]](#footnote-2)

J’ai utilisé ce service pour pouvoir sauvegarder chaque jour mon projet.

## 5.3 Font Awesome

Font Awesome est une police d'écriture et un outils d'icônes qui se base sur CSS, LESS et SASS.[[3]](#footnote-3)

Je l’ai utilisé pour pouvoir afficher des icônes à différents endroits de mon site. Pour l’utiliser il suffit d’utiliser le CDN donner sur leur site.

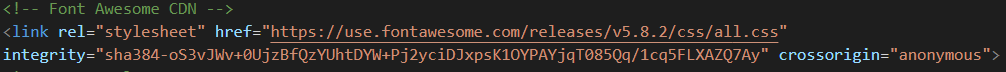


Figure : Lien CDN de Font Awesome

## 5.4 SweetAlert

SweetAlert est un remplacement des alertes de JavaScript, il a été conçu pour Bootstrap.[[4]](#footnote-4)

Je l’ai utilisé pour afficher un message d’erreur lors de la connexion en ajax. Pour l’utiliser il suffit d’utiliser le CDN donner sur leur site.



Figure : Lien CDN de SweetAlert

# 6. Installation

# 7. Tests

## 7.1 Page d’inscription

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 1 | Vérifier que la confirmation du mot de passe fonctionne | Dans le formulaire mettre un mot de passe différent dans la confirmation | Message d’erreur | OK | ∅ | ∅ |
| 2 | Inscription au site | Remplir le formulaire | Message de confirmation | OK | ∅ | ∅ |
| 3 | Vérifier que l’email soit unique lors de l’inscription | Remplir le champs email avec un email déjà pris | Message d’erreur : email déjà pris | OK | ∅ | ∅ |

Tableau : Tests de la page d’inscription

## 7.2 Page de connexion

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 4 | La connexion au site | Remplir le formulaire de connexion | L’utilisateur est redirigé vers la page d’accueil | OK | ∅ | ∅ |
| 5 | La connexion au site avec un mauvais mot de passe | Remplir le formulaire avec un mauvais mot de passe | Message d’erreur | OK | ∅ | ∅ |
| 6 | La déconnexion | Se déconnecter en cliquant sur le bouton  C:\Users\Admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\btnLogout.png | L’utilisateur est déconnecté | OK | ∅ | ∅ |

Tableau : Tests de la page de connexion

## 7.3 Page d’accueil

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 7 | Recherche des annonces de Lausanne | Sélectionner « Ville » dans la liste déroulante et rechercher « lausanne » puis cliquer sur le bouton | Affichage de 2 annonces « Pain à l’ancienne » et « Tomate et jus de tomate » appartenant à un utilisateur habitant bien à Lausanne | Affichage d’annonces non validées par l’administrateur | Oui | Changement de la requête SQL |
| 8 | Recherche des annonces bio et de Genève | Sélectionner « Canton » dans la liste déroulante et recherche « geneve » puis cliquer sur le bouton btnSearch | Affichage de 3 annonces avec le logo bio | OK | ∅ | ∅ |
| 9 | Recherche des annonces d’artichauts | Sélectionner « Titre de l’annonce » dans la liste déroulante et recherche « artichaut » puis cliquer sur le bouton btnSearch | Affichage de 2 annonces avec dans leur titre le nom « artichaut » | OK | ∅ | ∅ |
| 10 | Recherche des annonces avec une évaluation de 2 étoiles | Sélectionner « Evaluation de l’annonce » dans la liste déroulante et recherche « 2 » puis cliquer sur le bouton btnSearch | Affichage d’une annonce avec 2 étoiles d’évaluation | OK | ∅ | ∅ |
| 11 | Recherche avec un titre inconnue | Sélectionner « Titre de l’annonce » dans la liste déroulante et recherche « Essence » puis cliquer sur le bouton btnSearch | Affichage d’un message annonçant qu’il n’y a aucunes annonces pour cette recherche | OK | ∅ | ∅ |
| 12 | Le bouton  disparait quand on n’est pas connecté | Se déconnecter en cliquant sur le bouton | Le bouton  disparait | OK | ∅ | ∅ |

Tableau : Tests de la page d’accueil

## 7.4 Page de création d’une annonce

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 13 | Création d’une annonce sans image | Remplir le formulaire sans sélectionner d’image | L’annonce se créé avec un message de confirmation | OK | ∅ | ∅ |
| 14 | Création d’une annonce avec de grandes images | Remplir le formulaire avec des images de haut résolution | L’annonce se crée avec uniquement les images valides, s’il n’y en a pas l’annonce n’est pas créée | OK | ∅ | ∅ |
| 15 | Création d’une annonce avec une image valide et une autre invalide | Remplir le formulaire avec une image de moins de 3 Mo et une de plus de 3 Mo | L’annonce se crée avec uniquement les images valides, et un message affiche une erreur sur le fichier trop volumineux | OK | ∅ | ∅ |
| 16 | Création d’une annonce avec une mauvaise extension de fichier | Ajouter une image | Affichage d’une erreur sur le type de fichier | OK | ∅ | ∅ |
| 17 | Création d’une annonce avec des images | Remplir le formulaire avec des images acceptables | L’annonce se créé avec un message de confirmation | OK | ∅ | ∅ |

Tableau : Tests de la page de création d’une annonce

## 7.5 Page de modification d’une annonce

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 18 | Modifier une annonce en supprimant toutes les images | Cliquer sur le bouton en dessous de chaque image | Toutes les images ont été supprimées | OK | ∅ | ∅ |
| 19 | Modifier une annonce en ajoutant des images | Sélectionner des images et cliquer sur modifier | Les images sélectionner ont bien été ajoutées | OK | ∅ | ∅ |
| 20 | Modifier une annonce en ajoutant un fichier avec une mauvaise extension | Sélectionner un fichier .docx | Un message d’erreur spécifique s’affiche | OK | ∅ | ∅ |
| 21 | Modifier une annonce en ajoutant un fichier trop volumineux | Sélectionner un fichier de plus de 3 Mo | Un message d’erreur spécifique s’affiche | OK | ∅ | ∅ |

Tableau : Tests de la page de modification d’une annonce

## 7.6 Page de suppression d’une annonce

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 22 | Aller sur la page de suppression d’une annonce ne nous appartenant pas | Taper dans l’URL l’id d’une annonce ne nous appartenant pas | Redirection vers la page d’accueil | OK | ∅ | ∅ |
| 23 | Supprimer une annonce | Aller sur la page de suppression et cliquer sur le bouton | Affichage d’un message de confirmation et redirection vers la page d’accueil | OK | ∅ | ∅ |

Tableau : Tests de la page de suppression d’une annonce

## 7.7 Page de profil

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 24 | Je suis déconnecté et j’essaie d’aller sur la page de profil de quelqu’un d’autre | Taper dans l’URL l’email d’un utilisateur | Redirection vers la page d’accueil | OK | ∅ | ∅ |

Tableau : Tests de la page de profil

## 7.8 Page de modification de son profil

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 25 | Vérifier que la confirmation du mot de passe marche | Mettre un mauvais mot de passe dans le champ de confirmation | Message d’erreur spécifique | OK | ∅ | ∅ |
| 26 | Vérifier que l’application des modifications marche que si le mot de passe est correct | Mettre un mauvais mot de passe dans le champ du mot de passe actuel | Message d’erreur spécifique | OK | ∅ | ∅ |
| 27 | Vérifier que les modifications s’appliquent | Modifier des champs et rentrer le bon mot de passe | Message de confirmation et redirection vers la page du profil | La modification s’applique mais le mot de passe devient un string vide ("") | Oui | Utilisez un opérateur comparaison (!=) plutôt que isset() |

Tableau : Tests de la page de modification de son profil

## 7.9 Page de suppression de son profil

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 28 | Aller sur la page de suppression d’un autre utilisateur | Taper dans l’URL l’email de l’utilisateur | Redirection vers la page d’accueil | OK | ∅ | ∅ |
| 29 | Supprimer son profil | Aller sur la page de suppression et cliquer sur le bouton | Affichage d’un message de confirmation et redirection vers la page d’accueil | OK | ∅ | ∅ |

Tableau : Tests de la page de suppression de son profil

## 7.10 Page d’administration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **A tester** | **Comment ?** | **Résultat attendu ?** | **Résultat ?** | **Bug ?** | **Solution si problème ?** |
| 30 | La validation d’une annonce marche | On clique sur le bouton | Message de confirmation et elle disparait de la liste | OK | ∅ | ∅ |
| 31 | La suppression d’une annonce marche | On clique sur le bouton | Message de confirmation et elle disparait de la liste | OK | ∅ | ∅ |
| 32 | Je ne suis pas admin et j’essaie d’atteindre cette page | Taper l’URL de la page | Redirection vers la page d’accueil | OK | ∅ | ∅ |
| 33 | La redirection vers la page des détails | On clique sur le bouton | Redirection vers la page des détails | OK | ∅ | ∅ |
| 34 | La modification d’un utilisateur en administrateur | On clique sur le bouton  d’un utilisateur standard | Message de confirmation et changement du bouton | OK | ∅ | ∅ |
| 35 | La modification d’un administrateur en utilisateur | On clique sur le bouton    d’un administrateur | Message de confirmation et changement du bouton | OK | ∅ | ∅ |
| 36 | Modification de l’administrateur actuellement entrain de faire les changements | On clique sur le bouton  de son compte | Message de confirmation et redirection vers la page d’accueil | La modification s’applique mais l’utilisateur n’est pas redirigé vers la page d’accueil | Oui | Modifier la variable « role » dans la super global $\_SESSION |

Tableau : Tests de la page de suppression

# 8. Améliorations possibles

Pouvoir modifier / supprimer les commentaires

Ne pas pouvoir évaluer sa propre annonce mais pouvoir la commenter

Pouvoir voire le profil des autres utilisateurs et voir leurs autres annonces

Un email est envoyé lors de la validation / suppression d’une nouvelle annonce par l’administrateur

Faire un système de pagination sur la page d’administration car s’il y a beaucoup d’utilisateur ou d’annonce la page serra immense.

Utilisez le langage JavaScript

# 9. Conclusion

## 9.1 Difficultés rencontrées

La requête SQL de la recherche

Modification d’une annonce -> suppression d’une image

## 9.2 Bilan personnel

Ce projet met en place toutes les notions acquises durant ces 4 ans et j’en suis fier. Pouvoir réaliser un travail complet avec une documentation et une gestion du temps a été une expérience gratifiant, j’ai pu remarquer que prévoir le temps des tâches n’est pas une chose facile. Ce qui m’a amené à prendre de l’avance au début du projet et commencé plus tôt les tâches d’ampleurs pour pouvoir tenir mon planning. Je suis content d’avoir pu démontrer mes compétences.

## 9.3 Ce que ce projet m’a apporté

Ce projet m’a permis de me rendre compte qu’une bonne analyse au départ du projet, évite de faire pleins de petits changements tout au long du projet qui au bout du compte prennent beaucoup de temps.

# 10. Bibliographie

## 10.1 Codes repris

* Classe « Database », pris des projets de l’École Entreprise de la 3ème année et réalisé par M. Aigroz.

## 10.2 Sites utilisés

* Documentation PHP : <https://php.net/>
* Documentation SQL : <https://sql.sh/>
* Site Stack Overflow : <https://stackoverflow.com/>
* Site W3Schools : <https://www.w3schools.com/>
* Site Bootstrap : <https://getbootstrap.com/>

# 11. Planning

## 11.1 Planning prévisionnel

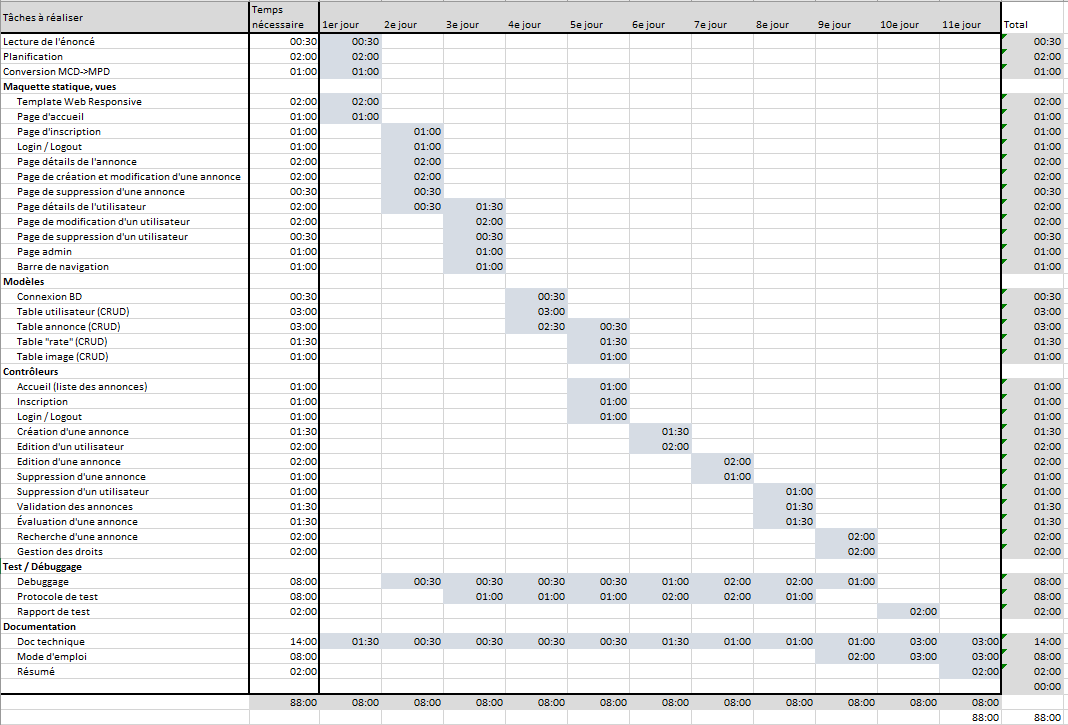


Figure : Planning prévisionnel

## 11.2 Planning effectif

## 11.3 Différence entre les plannings

# 12. Table des illustrations

## 12.1 Figures

[Figure 1 : Barre de navigation d’un utilisateur non connecté 5](#_Toc9426861)

[Figure 2 : Barre de navigation d’un utilisateur connecté 5](#_Toc9426862)

[Figure 3 : Barre de navigation de l’administrateur 5](#_Toc9426863)

[Figure 4 : Créer un compte 6](#_Toc9426864)

[Figure 5 : Connexion 7](#_Toc9426865)

[Figure 6: Page d’accueil d’un utilisateur non connecté 7](#_Toc9426866)

[Figure 7: Page d’accueil d’un utilisateur connecté 8](#_Toc9426867)

[Figure 8 : Détails d’une annonce avec un utilisateur non connecté 9](#_Toc9426868)

[Figure 9 : Détails d’une annonce avec un utilisateur connecté 10](#_Toc9426869)

[Figure 10: Formulaire de création d’une annonce 11](#_Toc9426870)

[Figure 11 : Modification d’une annonce 12](#_Toc9426871)

[Figure 12 : Confirmation de la suppression d’une annonce 13](#_Toc9426872)

[Figure 13 : Détails de notre profil 14](#_Toc9426873)

[Figure 14 : Modification de notre profil 15](#_Toc9426874)

[Figure 15 : Confirmation de la suppression de votre compte 17](#_Toc9426875)

[Figure 16 : Administration des nouvelles annonces 17](#_Toc9426876)

[Figure 17 : Modèle conceptuel proposé 18](#_Toc9426877)

[Figure 18 : Modèle conceptuel modifié 18](#_Toc9426878)

[Figure 19 : Modèle physique 19](#_Toc9426879)

[Figure 20 : Table « users » 20](#_Toc9426880)

[Figure 21 : Table « advertisements » 20](#_Toc9426881)

[Figure 22 : Table « rates » 20](#_Toc9426882)

[Figure 23 : Table « pictures » 20](#_Toc9426883)

[Figure 24 : Table « roles » 20](#_Toc9426884)

[Figure 25 : Arborescence de fichier 21](#_Toc9426885)

[Figure 26 : Lien CDN de Font Awesome 27](#_Toc9426886)

[Figure 27 : Lien CDN de SweetAlert 27](#_Toc9426887)

[Figure 28 : Planning prévisionnel 0](#_Toc9426888)

## 12.2 Tableaux

[Tableau 1 : Légende des propriétés des attributs 20](#_Toc9426841)

[Tableau 2 : Variables de la classe « User » 22](#_Toc9426842)

[Tableau 3 : Variables de la classe « Advertisement » 22](#_Toc9426843)

[Tableau 4 : Variables de la classe « Rating » 22](#_Toc9426844)

[Tableau 5 : Variables de la classe « Picture » 22](#_Toc9426845)

[Tableau 6 : Fonctions de la classe « Database » 23](#_Toc9426846)

[Tableau 7 : Fonctions de la classe « UserManager » 24](#_Toc9426847)

[Tableau 8 : Fonctions de la classe « AdvertisementManager » 25](#_Toc9426848)

[Tableau 9 : Fonctions de la classe « RatingManager » 25](#_Toc9426849)

[Tableau 10 : Fonctions de la classe « PictureManager » 26](#_Toc9426850)

[Tableau 11 : Tests de la page d’inscription 27](#_Toc9426851)

[Tableau 12 : Tests de la page de connexion 28](#_Toc9426852)

[Tableau 13 : Tests de la page d’accueil 28](#_Toc9426853)

[Tableau 14 : Tests de la page de création d’une annonce 29](#_Toc9426854)

[Tableau 15 : Tests de la page de modification d’une annonce 29](#_Toc9426855)

[Tableau 16 : Tests de la page de suppression d’une annonce 29](#_Toc9426856)

[Tableau 17 : Tests de la page de profil 30](#_Toc9426857)

[Tableau 18 : Tests de la page de modification de son profil 30](#_Toc9426858)

[Tableau 19 : Tests de la page de suppression de son profil 30](#_Toc9426859)

[Tableau 20 : Tests de la page de suppression 31](#_Toc9426860)

# 13. Annexes

* Code source
* Manuel utilisateur
* Journal de bord manuscrit

1. Source Wikipédia : <https://fr.wikipedia.org/wiki/Bootstrap_%28framework%29> [↑](#footnote-ref-1)
2. Source Wikipédia : <https://fr.wikipedia.org/wiki/GitHub> [↑](#footnote-ref-2)
3. Source Wikipédia : <https://fr.wikipedia.org/wiki/Font_Awesome> [↑](#footnote-ref-3)
4. Source GitHub : <https://github.com/lipis/bootstrap-sweetalert#sweetalert-for-bootstrap> [↑](#footnote-ref-4)