Documentation technique

Ackermann Gawen

06.05.2019 v1.0

[1. Introduction 3](#_Toc8646492)

[2. Contenu 3](#_Toc8646493)

[2.1Sujet 3](#_Toc8646494)

[2.2 But 3](#_Toc8646495)

[2.3 Spécifications 3](#_Toc8646496)

[2.4 Environnement de développement 3](#_Toc8646497)

[2.5 Livrable 3](#_Toc8646498)

[3. Analyse fonctionnelle 4](#_Toc8646499)

[3.0 Les différents profils utilisateur 4](#_Toc8646500)

[3.1 Navbar 4](#_Toc8646501)

[3.1.1 Visiteur 4](#_Toc8646502)

[3.1.2 Utilisateur authentifié 4](#_Toc8646503)

[3.1.3 Administrateur 4](#_Toc8646504)

[3.2 Accueil 5](#_Toc8646505)

[3.2.1 Système de filtrage 5](#_Toc8646506)

[3.2.2 Description d’un « bloc » d’annonce 6](#_Toc8646507)

[3.3 Détails d’une annonce 7](#_Toc8646508)

[3.3.1 Acheter un article (Non authentifié) 8](#_Toc8646509)

[3.4 Connexion 9](#_Toc8646510)

[3.5 Inscription 10](#_Toc8646511)

[3.6 Création d’une annonce 11](#_Toc8646512)

[3.7 Mes annonces 12](#_Toc8646513)

[3.8 Modification d’une annonce 13](#_Toc8646514)

[3.9 Gestion des annonces 14](#_Toc8646515)

[3.9.1 Changement d’état 14](#_Toc8646516)

[3.10 Gestion des utilisateurs 15](#_Toc8646517)

[4. Analyse organique 16](#_Toc8646518)

[4.1 Modèle Physique (Modèle Relationnel) 16](#_Toc8646519)

[4.2 Classes utilisées 17](#_Toc8646520)

[4.2.1 Class Ad 17](#_Toc8646521)

[4.2.2 Class Brand 17](#_Toc8646522)

[4.2.3 Class Country 17](#_Toc8646523)

[4.2.4 Class Gender 17](#_Toc8646524)

[4.2.5 Class Model 17](#_Toc8646525)

[4.2.6 Class Picture 18](#_Toc8646526)

[4.2.7 Class Role 18](#_Toc8646527)

[4.2.8 Class Size 18](#_Toc8646528)

[4.2.9 Class State 18](#_Toc8646529)

[4.2.10 Class User 19](#_Toc8646530)

[4.3 Manager utilisés 19](#_Toc8646531)

[4.3.1 AdManager 19](#_Toc8646532)

[4.3.1.1 getAds 19](#_Toc8646533)

[4.3.1.2 getAdById 19](#_Toc8646534)

[4.3.1.3 createAd 19](#_Toc8646535)

[4.3.2 BrandManager 20](#_Toc8646536)

[4.3.2.1 getBrandsName 20](#_Toc8646537)

[4.3.2.2 getAllBrands 20](#_Toc8646538)

[4.3.3 CountryManager 20](#_Toc8646539)

[4.3.3.1 getCountrysName 20](#_Toc8646540)

[4.3.3.2 getAllCountries 20](#_Toc8646541)

[4.3.4 GenderManager 20](#_Toc8646542)

[4.3.4.1 getGendersName 20](#_Toc8646543)

[4.3.4.2 getAllGenders 20](#_Toc8646544)

[4.3.5 ModelManager 21](#_Toc8646545)

[4.3.5.1 getModelsName 21](#_Toc8646546)

[4.3.5.2 getAllModels 21](#_Toc8646547)

[4.3.6 PictureManager 21](#_Toc8646548)

[4.3.6.1 insertPicturesForAd 21](#_Toc8646549)

[4.3.6.2 getPicturesForAnAd 21](#_Toc8646550)

[4.3.7 SizeManager 22](#_Toc8646551)

[4.3.7.1 getSizesName 22](#_Toc8646552)

[4.3.7.2 getAllSizes 22](#_Toc8646553)

[4.3.8 StateManager 22](#_Toc8646554)

[4.3.8.1 getStatesName 22](#_Toc8646555)

[4.3.8.2 getAllStates 22](#_Toc8646556)

[4.3.10 UserManager 23](#_Toc8646557)

[4.3.10.1 getUsers 23](#_Toc8646558)

[4.3.10.2 getUserByNickname 23](#_Toc8646559)

[4.3.10.3 createUser 23](#_Toc8646560)

[4.3.10.4 UserExist 23](#_Toc8646561)

[4.3.10.5 Connection 23](#_Toc8646562)

[5. Tests 24](#_Toc8646563)

[5.1 Scénarios de test 24](#_Toc8646564)

[5.1.1 Test No 1 24](#_Toc8646565)

[6. Planning prévisionnel 24](#_Toc8646566)

[7. Planning réel 24](#_Toc8646567)

[8. Conclusion 24](#_Toc8646568)

# 1. Introduction

Pour le Travail Pratique Individuel (TPI), nous avons connu quelques changements quant aux années précédentes.   
Cette année, nous n’avions pas accès à notre énoncé jusqu’au jour précédent le début du TPI. Mon chef de projet pour le TPI est M. Aigroz, c’est lui qui m’a fourni le travail à réaliser.

# 2**. Contenu**

## 2.1Sujet

Création d’un site web de vente d’habits prénommé « Seconde Main ».

## 2.2 But

Réaliser une application web permettant à des visiteurs de parcourir les annonces postées sur le site et de faire des achats. Les utilisateurs peuvent acheter des articles aussi mais peuvent en créer / modifier / supprimer. L’administrateur ne peut pas faire ce dont le visiteur ainsi que l’utilisateur connecté, en revanche il peut gérer les statuts des utilisateurs et ceux des annonces aussi.

## 2.3 Spécifications

Un visiteur doit inscrire son email et/ou un mot de passe lors de l’achat d’un article.  
Un utilisateur connecté n’as pas besoin de réaliser cette étape.  
La taille maximum d’un habit est XXL.

## 2.4 Environnement de développement

* Un PC standard école, 2 écrans
* Windows 10
* Apache / Module PHP / MySQL Server
* MySQL Workbench
* Visual Studio Code
* Suite Office

## 2.5 Livrable

* Planning
* Documentation technique
* Le résumé du TPI
* Manuel utilisateur séparé ou intégré à l’application
* Journal de travail

# 3. Analyse fonctionnelle

## 3.0 Les différents profils utilisateur

L’application gère trois types de profils : le visiteur, l’utilisateur authentifié et l’administrateur.  
Chacun de ces profils donne accès à des fonctionnalités.  
Ces fonctionnalités sont décrites ci-après.

## 3.1 Navbar

En fonction du profil, la barre de navigation intègre des fonctions liées à son rôle.

### 3.1.1 Visiteur

**A**

**B**

**1**



1. Bouton pour se diriger vers l’accueil du site. (Voir fig. 4)

### 3.1.2 Utilisateur authentifié

**D**

**C**

**B**

**A**

**2**



1. Bouton redirigeant vers l’accueil du site. (Voir fig. 4)
2. Bouton redirigeant sur la page permettant de créer une annonce
3. Bouton redirigeant sur la page répertoriant toutes les annonces postées par l’utilisateur connecté
4. Bouton permettant la déconnexion de l’utilisateur

### 3.1.3 Administrateur

**A**

**D**

**C**

**B**

**3**



1. Bouton redirigeant vers l’accueil du site. (Voir fig. 4)
2. Bouton redirigeant sur la page permettant de gérer les utilisateurs
3. Bouton redirigeant sur la page de gestions des annonces répertoriant toutes les annonces postées par les utilisateurs connectés
4. Bouton permettant la déconnexion de l’utilisateur

## 3.2 Accueil

La page principale, accessible aux visiteurs, répertorie les dernières annonces publiées par les utilisateurs. L’utilisateur a la possibilité de filtrer les annonces. Il accède à une annonce en cliquant sur l’image ou le texte joint.

**4**



/!\ ATTENTION : changer la barre de filtre

**F**

**D**

**B**

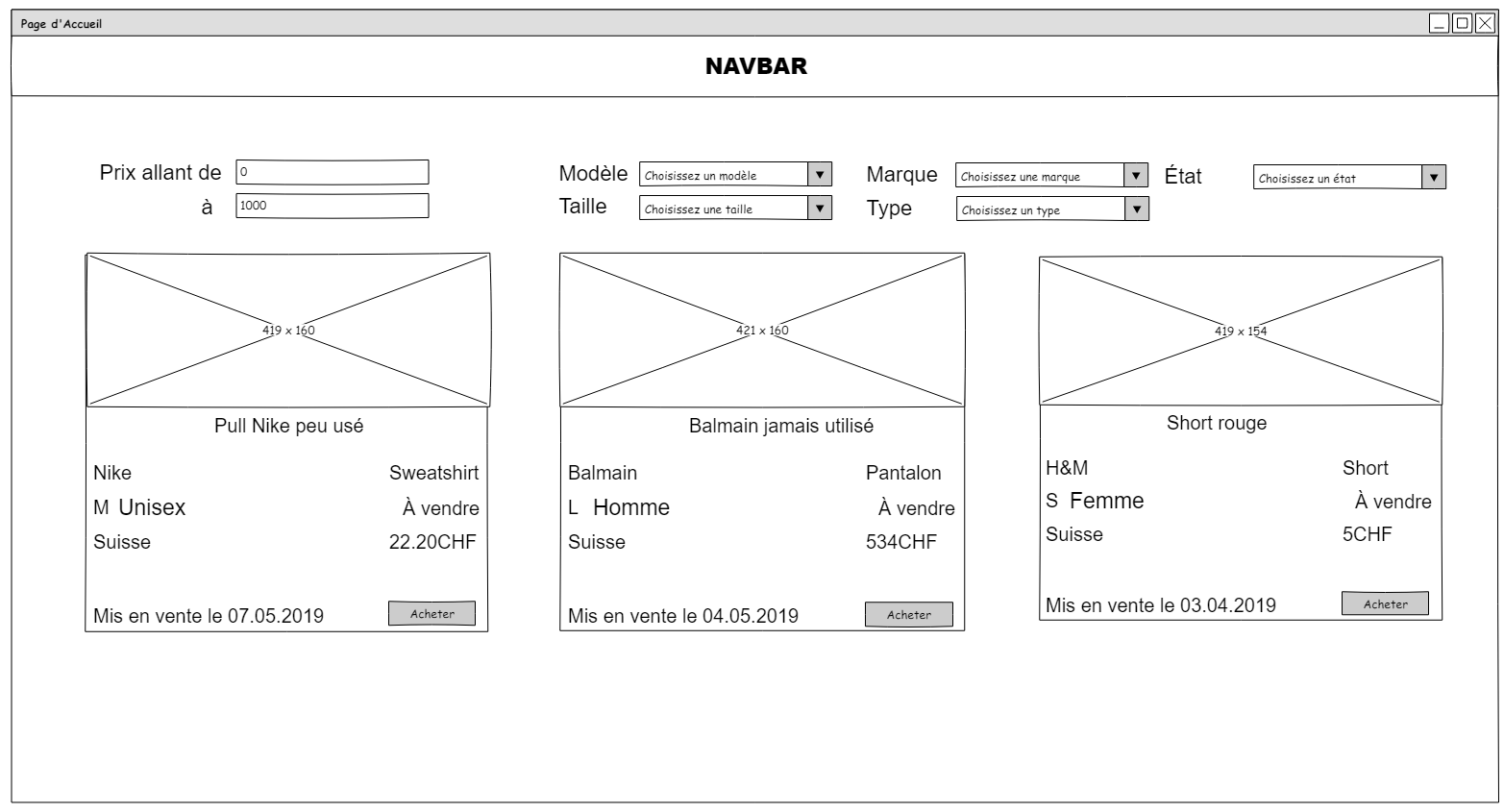
### 3.2.1 Système de filtrage

**A**

**E**

**C**

**5**



1. Plage de prix
2. Le modèle (Exemple : T-Shirt, Pantalon, Short, Pull-Over, etc…)
3. La taille de l’habit sous la forme de lettre tels que « L », « M », …, « XXL »
4. La marque du produit, par exemple « Nike », « Puma », « H&M », etc…
5. Détermine si le produit est pour femme, homme ou unisexe
6. L’état de l’annonce (Il existe « à vendre », « vendu », « en cours de vente », « suspendre », « supprimer »)

### 3.2.2 Description d’un « bloc » d’annonce

Voici ce à quoi ressemble un « bloc » d’annonce. Lors du clic sur l’image de l’annonce, elle est redirigée sur la page de description d’un article. (Voir fig. 7)

**6**

**D**

**C**

**B**

**A**



1. L’image représentant le produit
2. Le titre du produit
3. La marque du produit
4. Le prix du produit

## 3.3 Détails d’une annonce

**K**

**J**

**I**

**H**

**G**

**F**

**E**

**D**

**C**

**B**

**A**

**7**



1. L’image représentant le produit
2. Le titre du produit
3. La description du produit
4. La marque du produit
5. La taille du produit (Les tailles vont de XS à XXL)
6. Le modèle du produit (Il existe toute sortes de modèle tels que : « T-Shirt », « Sweatshirt », « Chemise », etc…)
7. Le type de produit (Il existe : Femme, Homme et Unisexe)
8. Le pays du vendeur
9. Bouton d’achat du produit (S’il s’agit d’un visiteur, il sera redirigé sur une page de confirmation avec son email, voir fig. 8)
10. Le prix du produit
11. Date de mise en vente

### 3.3.1 Acheter un article (Non authentifié)

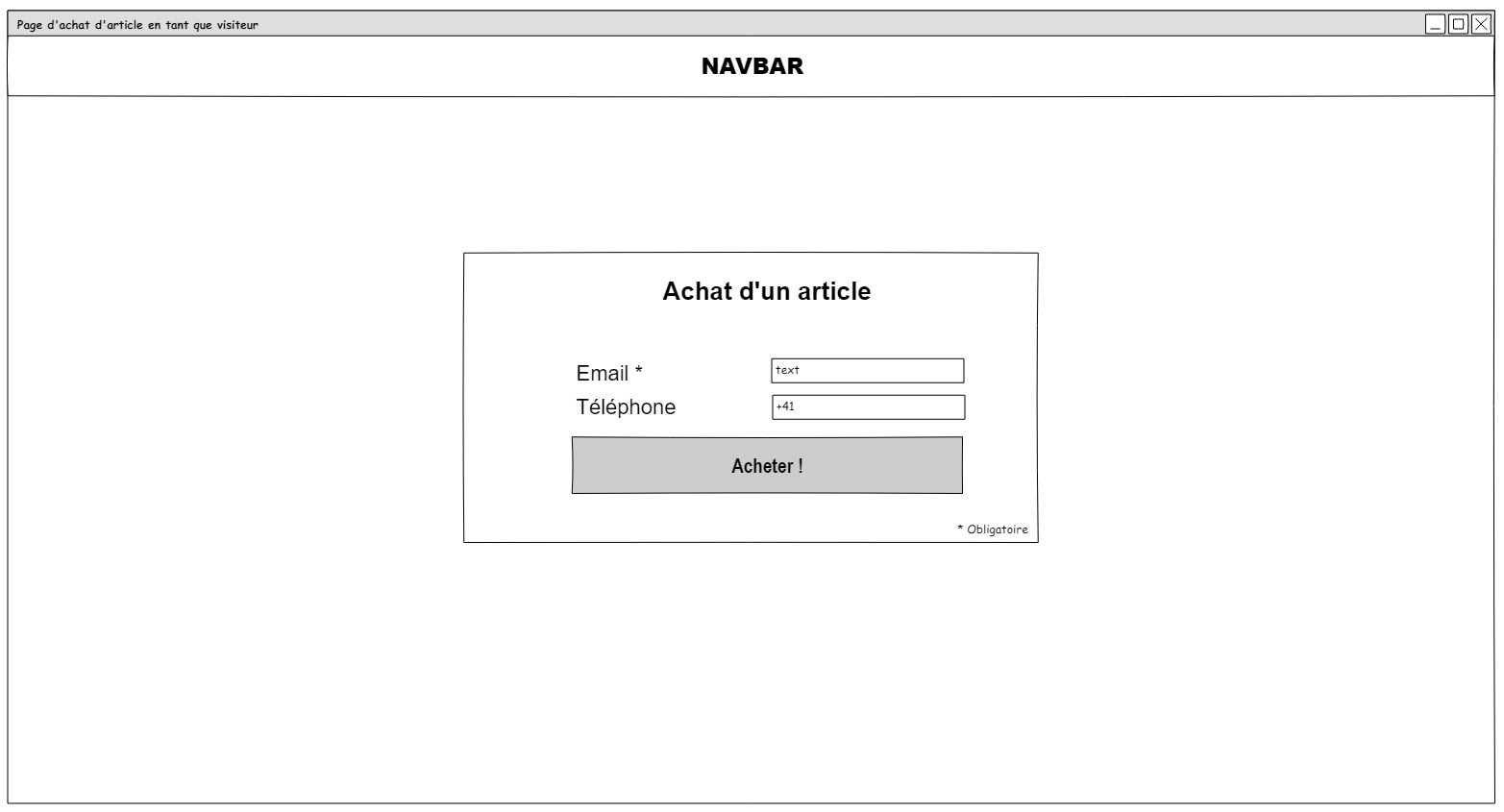
Page confirmant l’achat d’un article pour un visiteur.

**C**

**B**

**A**

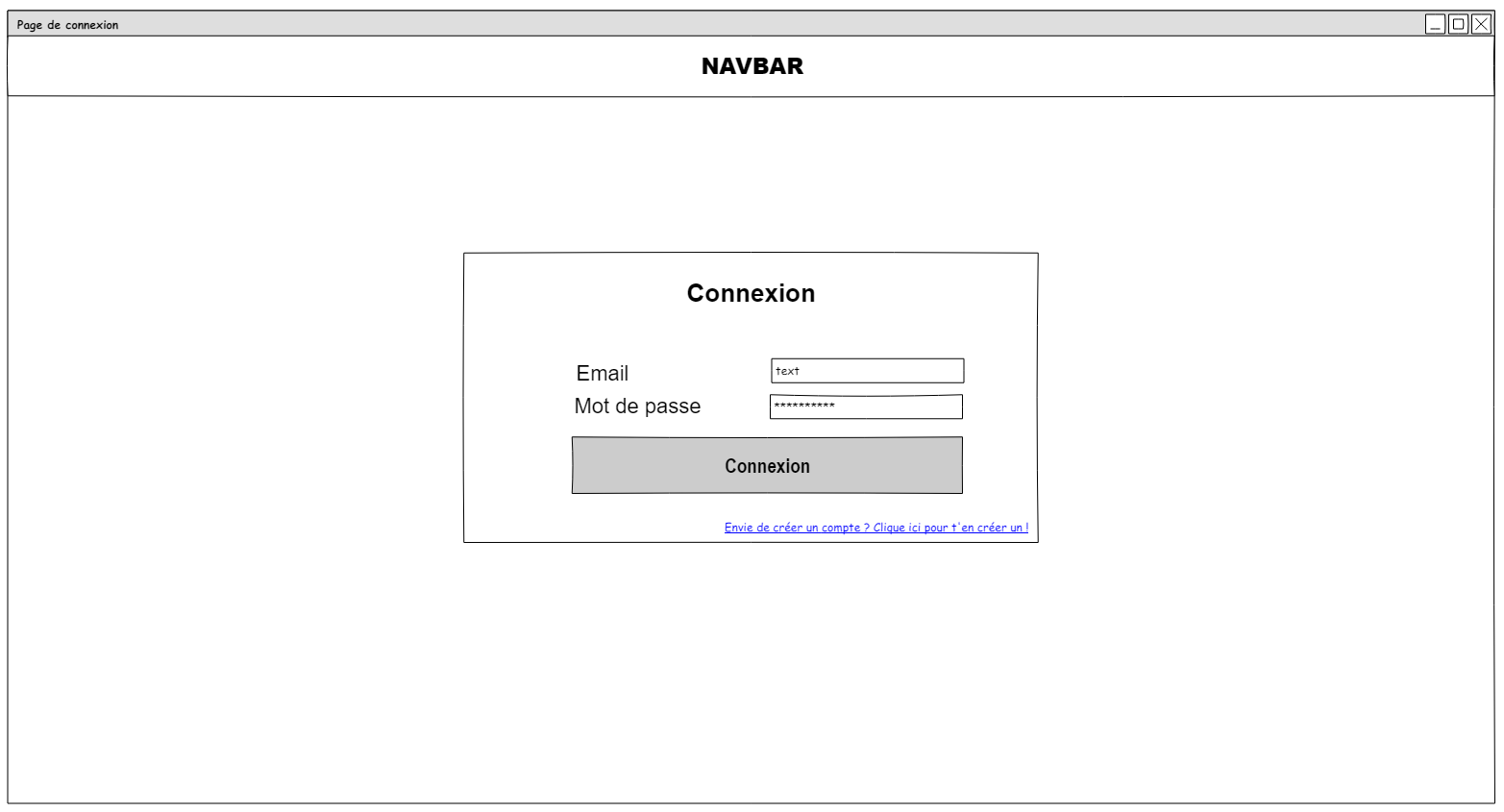
**8**



1. Email à fournir pour acheter un article (Obligatoire)
2. N° de téléphone
3. Bouton de validation d’achat

## 3.4 Connexion

Page permettant à l’utilisateur de se connecter. L’utilisateur doit spécifier son adresse email et son mot de passe.



**D**

**8**

**C**

**A**

**B**

1. Email de l’utilisateur
2. Mot de passe de l’utilisateur
3. Bouton de connexion
4. Lien redirigeant sur la page de création de compte

## 3.5 Inscription

Page d’inscription pour les utilisateurs.

**H**

**G**

**F**

**E**

**A**

**B**

**C**

**D**

**9**



1. Pseudonyme de l’utilisateur
2. Nom de famille de l’utilisateur
3. Prénom de l’utilisateur
4. Pays de résidence de l’utilisateur
5. N° de téléphone de l’utilisateur
6. Email de l’utilisateur
7. Mot de passe de l’utilisateur
8. Bouton de validation d’inscription

## 3.6 Création d’une annonce

Page permettant à l’utilisateur connecté de créer une annonce.

**10**

**I**

**B**

**F**

**G**

**E**

**A**

**C**

**H**

**D**



1. Titre de l’annonce
2. Marque du produit
3. Modèle du produit
4. Taille du produit
5. Type du produit
6. État de l’annonce
7. Prix de vente
8. Bouton permettant de charger une ou plusieurs images
9. Bouton de création de l’annonce

## 3.7 Mes annonces

Page affichant à l’utilisateur toutes les annonces qu’il a posté. Comme pour les autres pages affichant des annonces, le système de tri est présent (Pour plus d’informations, voir Fig. 5).  
 Un bouton de modification est disponible, lors du clic sur ce dernier, l’utilisateur sera redirigé sur la page de modification de l’annonce. (Voir fig. 12)

**A**

**11**



1. Bouton de modification de l’annonce

## 3.8 Modification d’une annonce

Page de modification des informations pour l’annonce désirée.

**A**

**B**

**12**



(Pour avoir des informations complémentaires pour chaque champ, voir Fig. 10)

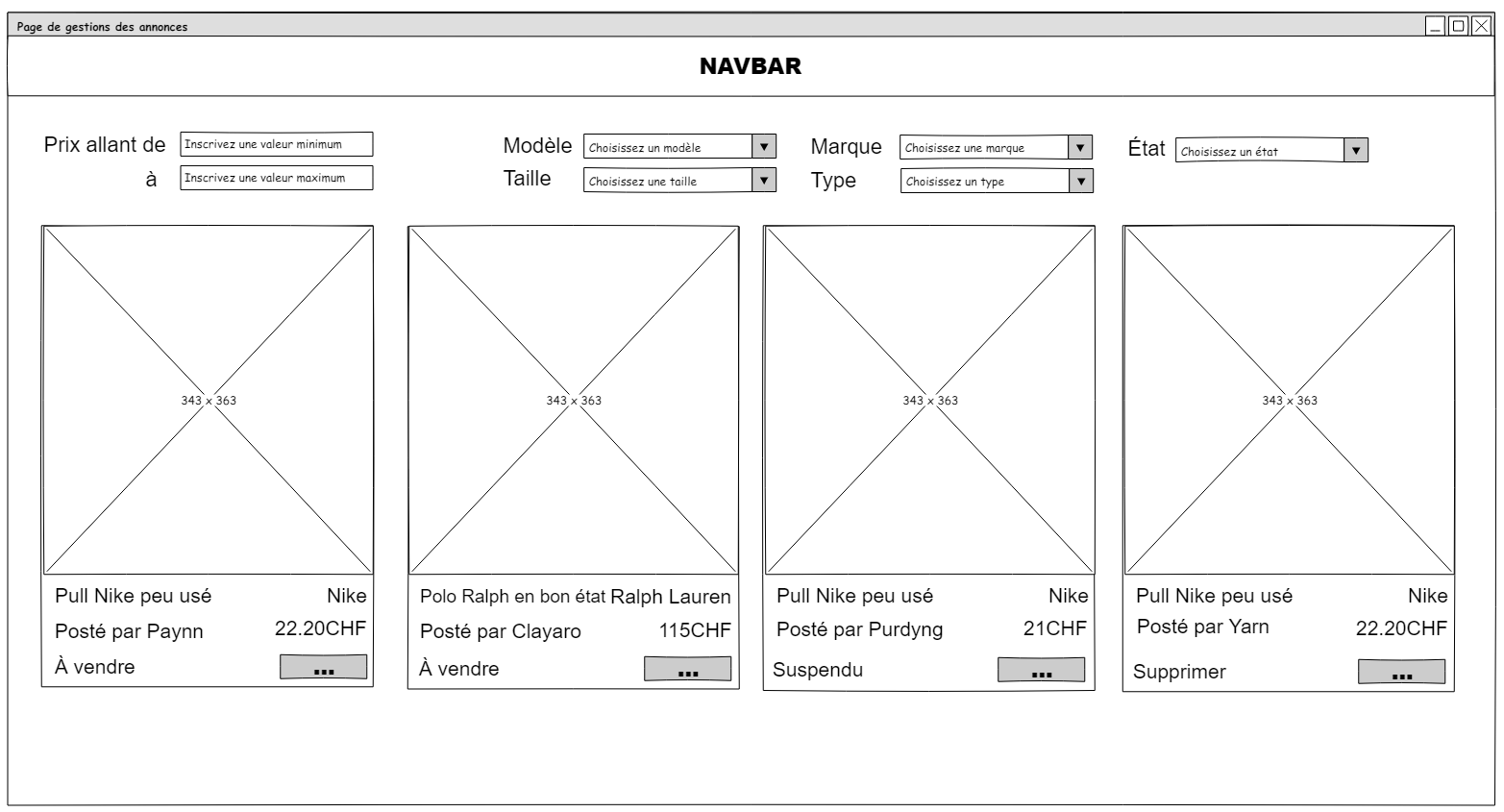
1. Toutes les images liées à l’annonce
2. Bouton de suppression d’une image

## 3.9 Gestion des annonces

Page d’administration permettant à l’administrateur de changer l’état des annonces publiées.

**A**

**13**



1. Bouton de changement d’état d’une annonce (voir fig. 14)

### 3.9.1 Changement d’état

Lors du clic sur le bouton « … », un menu déroulant apparaît laissant le choix à l’administrateur de l’état de l’annonce.

**A**

**14**



1. Menu déroulant affichant les états, lors du clic sur l’un des états, ce dernier deviendra l’état actuel de l’annonce.

## 3.10 Gestion des utilisateurs

Page d’administration permettant à l’administrateur de changer le statut d’un utilisateur inscrit. Il peut changer leur statut en choisissant celui voulu dans la liste déroulante.

**15**

**B**

**A**

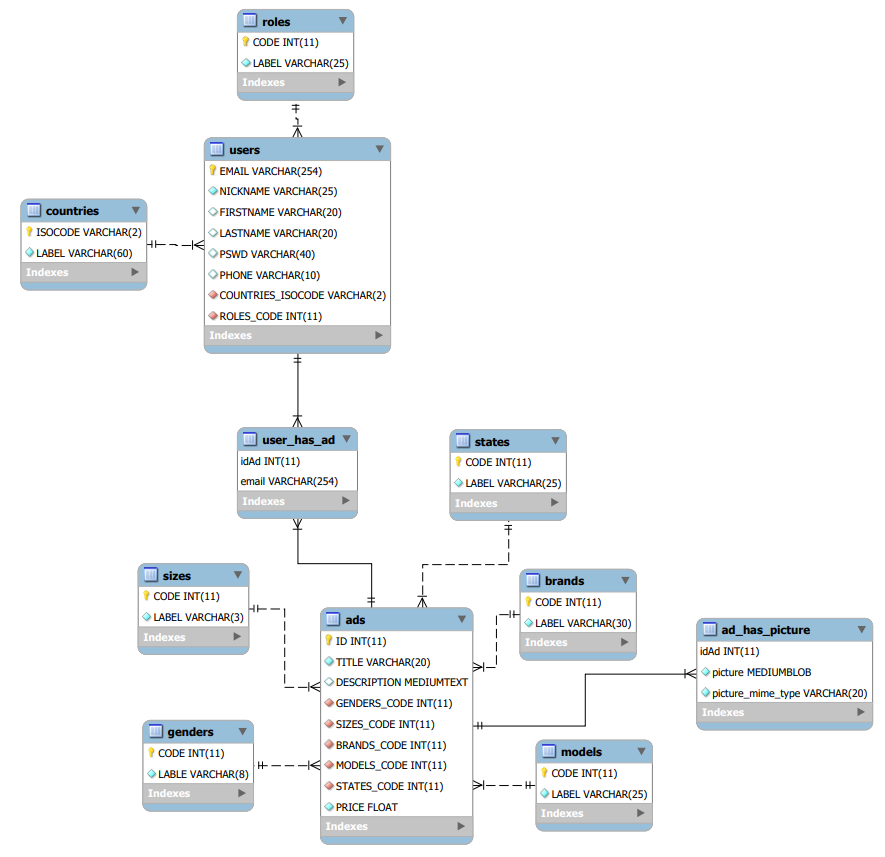


1. Pseudonyme de l’utilisateur
2. Statut de l’utilisateur

# 4. Analyse organique

## 4.1 Modèle Physique (Modèle Relationnel)

**16**



À MODIFIER QUAND LE SITE EST TERMINÉ POUR ÉVITER DE CHANGER À CHAQUE AJOUT, EN CAS D’OUBLIS

## 4.2 Classes utilisées

### 4.2.1 Class Ad

Cette classe contient les informations sur une Annonce

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| id | L'id de l'annonce | INT |
| nickname | Le pseudonynme de la personne détenant l'annonce | STRING |
| title | Le titre de l'annonce | STRING |
| description | La description du produit | STRING |
| gender | Le code du type du produit | INT |
| size | Le code de la taille du produit | INT |
| brand | Le code de la marque du produit | INT |
| model | Le code du modèle du produit | INT |
| state | Le code de l'état du produit | INT |
| price | Le prix du produit | FLOAT |
| postingDate | La date de mise en ligne de l'annonce | DATETIME |

### 4.2.2 Class Brand

Cette classe contient les informations des différentes marques.

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| code | Le code de la marque | INT |
| label | Le libellé de la marque | STRING |

### 4.2.3 Class Country

Cette classe contient les informations des différents pays.

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| isocode | Le code ISO du pays | INT |
| label | Le libellé du pays | STRING |

### 4.2.4 Class Gender

Cette classe contient les informations des différents types.

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| code | Le code du type | INT |
| label | Le libellé du type | STRING |

### 4.2.5 Class Model

Cette classe contient les informations des différents modèles d’habits.

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| code | Le code du modèle | INT |
| label | Le libellé du modèle | STRING |

### 4.2.6 Class Picture

Cette classe contient les informations sur une Image.

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| idAd | L'id de l'annonce à laquelle l'image est reliée | STRING |
| index | L'ordre d'ajout de l'image | INT |
| img | L'image encodée en base64 | STRING |

### 4.2.7 Class Role

Cette classe contient les informations des différents rôles.

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| code | Le code du rôle | INT |
| label | Le libellé du rôle | STRING |

### 4.2.8 Class Size

Cette classe contient les informations des différentes tailles d'habits.

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| code | Le code de la taille | INT |
| label | Le libellé de la taille | STRING |

### 4.2.9 Class State

Cette classe contient les informations des différents états.

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| code | Le code de l'état | INT |
| label | Le libellé de l'état | STRING |

### 4.2.10 Class User

Cette classe contient les informations sur une personne (utilisateur) inscrite sur Seconde Main

|  |  |  |
| --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE |
| email | L'email de la personne | STRING |
| nickname | Le pseudo de la personne | STRING |
| firstname | Le prénom de la personne | STRING |
| lastname | Le nom de famille de la personne | STRING |
| phone | Le N° de téléphone de la personne | STRING |
| country | Le code du pays de résidence de la personne | INT |
| role | Le code du rôle de la personne | INT |
| pswd | Le mot de passe de la personne | STRING |
| status | Le status de l’utilisateur | INT |

## 4.3 Manager utilisés

### 4.3.1 AdManager

Cette classe contient les fonctions traitant ou récupérant des données concernant une annonce

#### 4.3.1.1 getAds

Fonction récupérant les informations de toutes les annonces

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| aucun |  |  |
|  | Tableau d’Ad | Toutes les annonces |

#### 4.3.1.2 getAdById

Fonction récupérant les informations d'une annonce

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| id | INT |  |
|  | Object Ad | Les informations de l'annonce |

#### 4.3.1.3 createAd

Fonction créant une annonce

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| Ad | Object Ad |  |
| pictures | STRING[] |  |
|  | Boolean | True si OK, False si problème d'insertion |

#### 4.3.1.4 getAdsFromUser

Fonction récupérant les annonces postées par l'utilisateur connecté

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| nickname | STRING |  |
|  | Tableau d’Ad | Les informations de toutes les annonces de l'utilisateur |

#### 4.3.1.5 getAdsUsersNickname

Fonction récupérant le pseudonyme du propriétaire de l'annonce

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| idAd | INT |  |
|  | STRING | Le pseudonyme du propriétaire de l'annonce |

#### 4.3.1.6 modifyAd

Fonction modifiant une annonce

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| Ad | Object Ad |  |
|  | Boolean | True si OK, False si problème de modification |

#### 4.3.1.7 modifyAdsState

Fonction modifiant l’état d’une annonce

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| Ad | Object Ad |  |
|  | Boolean | True si OK, False si problème de modification |

#### 4.3.1.8 getAdsWithFilter

Fonction récupérant les annonces selon le filtre appliqué

|  |  |  |  |
| --- | --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE | VALEUR RETOURNÉE |
| minPrice | Le prix minimum du produit | INT |  |
| maxPrice | Le prix maximum du produit | INT |  |
| brand | La marque du produit | INT |  |
| model | Le modèle du produit | INT |  |
| size | La taille du produit | INT |  |
| type | Le type du produit | INT |  |
| state | L'état de l'annonce | INT |  |
|  |  | Ad[] | Toutes les annonces existantes |

#### 4.3.1.9 getAdsFromUserWithFilter

Fonction récupérant les annonces selon le filtre appliqué du propriétaire de l'annonce

|  |  |  |  |
| --- | --- | --- | --- |
| VARIABLES | DESCRIPTION | TYPE | VALEUR RETOURNÉE |
| minPrice | Le prix minimum du produit | INT |  |
| maxPrice | Le prix maximum du produit | INT |  |
| brand | La marque du produit | INT |  |
| model | Le modèle du produit | INT |  |
| size | La taille du produit | INT |  |
| type | Le type du produit | INT |  |
| state | L'état de l'annonce | INT |  |
| nickname | Le pseudo du détenteur de l'annonce | STRING |  |
|  |  | Ad[] | Toutes les annonces existantes |

### 4.3.2 BrandManager

Cette classe contient les fonctions récupérant les données concernant une marque

#### 4.3.2.1 getBrandsName

Fonction récupérant le nom de la marque

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| id | INT |  |
|  | STRING | Le libellé de la marque |

#### 4.3.2.2 getAllBrands

Fonction récupérant toutes les informations concernant les marques

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| aucun |  |  |
|  | Tableau de Brand | les informations concernant les marques |

### 4.3.3 CountryManager

Cette classe contient les fonctions récupérant les données concernant les pays

#### 4.3.3.1 getCountrysName

Fonction récupérant le label du pays en fonction de son Code ISO

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| isocode | STRING |  |
|  | STRING | Le libellé du pays |

#### 4.3.3.2 getAllCountries

Fonction récupérant toutes les informations concernant les pays

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| aucun |  |  |
|  | Tableau de Country | les informations concernant les pays |

### 4.3.4 GenderManager

Cette classe contient les fonctions récupérant les données concernant le type d'un habit

#### 4.3.4.1 getGendersName

Fonction récupérant le label du type en fonction de son ID

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| id | INT |  |
|  | STRING | Le libellé du type |

#### 4.3.4.2 getAllGenders

Fonction récupérant toutes les informations concernant les types

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| aucun |  |  |
|  | Tableau de Gender | les informations concernant les types |

### 4.3.5 ModelManager

Cette classe contient les fonctions récupérant les données concernant les différents modèles d'habits

#### 4.3.5.1 getModelsName

Fonction récupérant le label du modèle en fonction de son ID

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| id | INT |  |
|  | STRING | Le libellé du modèle |

#### 4.3.5.2 getAllModels

Fonction récupérant toutes les informations concernant les modèles

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| aucun |  |  |
|  | Tableau de Model | les informations concernant les différents modèles |

### 4.3.6 PictureManager

Cette classe contient les fonctions récupérant / Ajoutant les données concernant les images d'une annonce

#### 4.3.6.1 insertPicturesForAd

Fonction insérant l(es) image(s) associé(es)

|  |  |  |  |
| --- | --- | --- | --- |
| paramètre(s) | Description | TYPE | valeur retournée |
| id | L'ID de l'annonce | INT |  |
| Pictures | Tableau d’images encodées en base 64 | STRING[] |  |
|  |  | Boolean | True si OK, False si problème d'insertion |

#### 4.3.6.2 getPicturesForAnAd

Fonction récupérant les images affiliées à une annonce

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| id | L'ID de l'annonce | INT |
|  | Tableau de Picture | L(es) image(s) de l'annonce |

### 4.3.7 SizeManager

Cette classe contient les fonctions récupérant concernant les tailles d'habits

#### 4.3.7.1 getSizesName

Fonction récupérant le label de la taille en fonction de son ID

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| id | INT |  |
|  | STRING | Le libellé de la taille |

#### 4.3.7.2 getAllSizes

Fonction récupérant toutes les informations concernant les tailles d’habits

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| aucun |  |  |
|  | Tableau de Size | les informations concernant les différents modèles |

### 4.3.8 StateManager

Cette classe contient les fonctions récupérant concernant les différents états possibles pour une annonce

#### 4.3.8.1 getStatesName

Fonction récupérant le label de l'état en fonction de son ID

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| id | INT |  |
|  | STRING | Le libellé de l'état |

#### 4.3.8.2 getAllStates

Fonction récupérant toutes les informations concernant les tailles d’habits

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| aucun |  |  |
|  | Tableau de State | les informations concernant les différents états |

### 4.3.9 StatusManager

Cette classe contient les fonctions récupérant concernant les différents statuts possibles pour un utilisateur

#### 4.3.9.1 getStatusName

Fonction récupérant le label du statut en fonction de son ID

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| id | INT |  |
|  | STRING | Le libellé de l'état |

#### 4.3.9.2 getAllStatus

Fonction récupérant toutes les informations concernant les statuts

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| aucun |  |  |
|  | Tableau de State | les informations concernant les différents états |

### 4.3.10 UserManager

Cette classe contient les fonctions traitant ou récupérant des données concernant un utilisateur

#### 4.3.10.1 getUsers

Fonction récupérant les informations de tous les utilisateurs

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| AUCUN |  |  |
|  | Tableau d’User | Les informations des utilisateurs |

#### 4.3.10.2 getUserByNickname

Fonction récupérant les informations d'un utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| nickname | STRING |  |
|  | Objet User | Les informations de l’utilisateur |

#### 4.3.10.2 getUserByEmail

Fonction récupérant les informations d'un utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| email | STRING |  |
|  | Objet User | Les informations de l’utilisateur |

#### 4.3.10.3 createUser

Fonction créant un utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| User | Objet User |  |
|  | Boolean | True si OK, False si problème d'insertion |

#### 4.3.10.4 UserExist

Fonction vérifiant l'existence d'un utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| email | STRING |  |
|  | Boolean | True si OK, False si inexistant |

#### 4.3.10.5 Connection

Fonction connectant un utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| email | STRING |  |
| password | STRING |  |
|  | Boolean | True si OK, False si un champ est incorrect |

#### 4.3.10.6 modifyUsersStatus

Fonction modifiant le statut d'un utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| nickname | STRING |  |
| idStatus | INT |  |
|  | Boolean | True si OK, False si problème de modification |

#### 4.3.10.7 createToken

Fonction créant un token de validation en comptant le nombre de lettre du pseudonyme de l'utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| nickname | STRING |  |
|  | Boolean | True si OK, False si problème de modification |

### 4.3.11 SessionManager

Classe récupérant et assignant à la variable $\_SESSION les différentes informations à remplir.

#### 4.3.11.1 SetUser

Fonction assignant les informations de l’utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| user | Objet User |  |
|  |  | AUCUNE |

#### 4.3.11.2 GetUser

Fonction récupérant les informations de l’utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| AUCUN | Objet User |  |
|  | Objet User | Les informations de l’utilisateur |

#### 4.3.11.3 setNickname

Fonction assignant le pseudonyme de l’utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| nickname | STRING |  |
|  |  | AUCUNE |

#### 4.3.11.3 GetNickname

Fonction récupérant le pseudonyme de l’utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| AUCUN |  |  |
|  | STRING | Le pseudonyme de l’utilisateur |

#### 4.3.11.3 GetRole

Fonction récupérant le rôle de l’utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| AUCUN |  |  |
|  | STRING | Le rôle de l’utilisateur |

### 4.3.12 MailManager

Classe envoyant les emails

#### 4.3.12.1 initMailer

Fonction créant une instance SMTP

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| AUCUN |  |  |
|  | Instance Swift\_SmtpTransport | Le SMTP |

#### 4.3.12.2 sendMailForTokenValidation

Fonction envoyant un mail lors de la création de compte

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| token | STRING |  |
| emailUser | STRING |  |
|  | boolean | True si ok, False si problème d'envois |

#### 4.3.12.3 sendMailForAPurchase

Fonction envoyant un mail lors d'une demande d'achat

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| userWhoPurchase | STRING |  |
| userPhone | STRING |  |
| userWhoSell | STRING |  |
| adTitle | STRING |  |
|  | boolean | True si ok, False si problème d'envois |

#### 4.3.12.4 sendMailWhenUsersStatusChange

Fonction envoyant un mail lors du changement de statut d'un utilisateur

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| emailUsersStatusModified | STRING |  |
| state | INT |  |
| adTitle | STRING |  |
|  | boolean | True si ok, False si problème d'envois |

#### 4.3.12.4 sendMailWhenUsersAdsStateChange

Fonction envoyant un mail lors du changement de l'état d'une annonce

|  |  |  |
| --- | --- | --- |
| paramètre(s) | TYPE | valeur retournée |
| emailOwnersAd | STRING |  |
| status | INT |  |
|  | boolean | True si ok, False si problème d'envois |

# 5. Tests

## 5.1 Scénarios de test

### 5.1.1 Test No 1

# 6. Planning prévisionnel

# 7. Planning réel

# 8. Conclusion

Bilan du projet