

# **Documentation Technique : Search & Find Job**

**par: Rodrigues dos Santos F.**

**Suivi par : Garchery S.**

**Classe IFA-P3a**

---

**Search & Find Job**

# 1. Table des matières

---

## 1. Table des matières

## 2. Introduction

## 3. Étude d'opportunité

- 3.1 Pourquoi ce sujet ?
- 3.2 Ce que mon projet apporte en plus
- 3.3 État actuel

## 4. Méthodologie

- Récolte d'informations
- Documentation
- Décisions
- Réalisation
- Contrôles
- Évaluation

## 5. Rappel du cahier des charges

- 5.1 Objectif
- 5.2 Matériel à disposition
- 5.3 Description de l'application
  - Types d'utilisateurs
  - Fonctionnalités
  - Une annonce
  - Création de comptes
  - Vues pour les annonceurs
  - Vues pour les chercheurs
  - Vues pour les administrateurs
  - Remarques générales
  - Modèle de données
  - Manuel utilisateur
- 5.4 Livrables

## 6. Planification

- 6.1 Planning Prévisionnel
- 6.2 Planning Réel
- 6.3 Maquettes
  - Accueil
  - Login
  - Inscription
  - Annonces
  - Mes annonces
  - Modifier Annonce
  - Wishlist
  - Administration : Gestion des Mots-Clés
  - Administration : Gestion des Utilisateurs

## 7. Analyse Fonctionnelle

- 7.1 Fonctionnalités
  - 7.1.1 Description Globale
  - 7.1.2 Description des Fonctionnalités
    - Connexion d'un utilisateur
    - Inscription d'un nouvel utilisateur
    - Recherche d'annonces par titre et contenu
    - Recherche d'annonces par mots-clés
    - Affichage d'annonces
    - Créer une Annonce
    - Ajout d'un média à une annonce
    - Retrait de média d'une annonce
    - Mise à Jour du média d'une annonce

- Modification d'annonces
- Suppression d'annonces
- Affichage d'informations d'annonce
- Ajouter à la wishlist utilisateur
- Afficher la wishlist d'un utilisateur
- Afficher les utilisateurs ayant ajouté une annonce à leur wishlist
- CRUD Mots-Clés
- Gestion des utilisateurs

## 7.2 Interfaces

### 7.2.1 Index

- Index
- Index Chercheur
- Index Annonceur
- Index Administrateur

### 7.2.2 Barre de Navigation

- Barre de Navigation
- Barre de Navigation du Chercheur
- Barre de Navigation de l'Annonceur
- Barre de Navigation de l'Administrateur

### 7.2.3 Annonces

- Annonces
- Mes annonces

### 7.2.4 Annonce

- Annonce (Chercheur)
- Annonce (Annonceur)

### 7.2.5 Créer Annonce

### 7.2.6 Modifier Annonce

### 7.2.7 Supprimer Annonce

### 7.2.8 Login

### 7.2.9 Inscription

### 7.2.10 Wishlist

### 7.2.11 Gestion Utilisateurs

### 7.2.12 Gestion Mots-Clés

## 8. Analyse Organique

### 8.1 Diagramme Général

### 8.2 Environnement

### 8.3 Base de donnée

#### 8.3.1 MLD de la Base de donnée

#### 8.3.2 Description des tables

- Annonces
- Annonces\_has\_keywords
- Keywords
- Utilisateurs
- Wishlist

### 8.4 Description des fonctions

#### 8.4.1 Requêtes SQL

- Modèle de base
- Modèle transaction

db()

#### 8.4.2 Fonctions d'authentification

- ChangeLoginState()
- VerifyIfMailExists()
- ConnectUser()
- RegisterUser()

#### 8.4.3 Fonctions d'alertes

- SetAlert()
- GetAlert()
- ShowAlert()

#### 8.4.4 Fonctions d'annonces

CheckMedia()

CreerAnnonce()

GetAnnoncesFromSearchAnnonceur()

GetAnnoncesFromSearchChercheur()

UploadMedia()

#### 8.4.5 Fonctions de Wishlist

## 2. Introduction

---

Cette documentation a pour but de présenter le projet web "Search & Find Job" réalisé dans le cadre de mon Travail Pratique Individuel afin de pouvoir valider mon CFC d'informaticien.

L'objectif de ce TPI est la réalisation d'un site web en PHP, HTML, CSS, Bootstrap et SQL qui permet aux utilisateurs s'y rendant de s'inscrire en tant que "Annonceur" ou "Chercheur". L'annonceur pouvant créer des annonces qu'ensuite le chercheur pour ajouter à sa liste de souhaits (Wishlist) étant l'ensemble des annonces que ce dernier trouve pertinentes. Pour finir, il existe le type d'utilisateur "Administrateur" qui peut gérer les mots-clés utilisés par les annonceurs lors de la création d'annonces et gérer les comptes utilisateurs en modifiant leur type ou mot de passes.

## 3. Étude d'opportunité

---

### 3.1 Pourquoi ce sujet ?

---

Je me suis lancé sur ce projet car aujourd'hui la demande d'emploi est toujours de plus en plus grandissante et ce n'est pas forcément très simple surtout pour les jeunes de trouver ces dits emplois. C'est pourquoi m'est venu l'idée de créer ce site afin de mettre en relation plus aisément les employeurs et les potentiel intéressés.

### 3.2 Ce que mon projet apporte en plus

---

Contrairement aux sites comme Jobscout24.ch ou jobup.ch, j'ai décidé de miser avant-tout sur l'aspect simple d'utilisation et allant droit au but. Je trouve ces sites peu simple d'utilisation et possédant un affichage assez encombré et peu visuellement attrayant ce qui peut repousser des potentiels chercheurs d'emplois.

C'est pourquoi j'ai mis en place un système très simpliste semblable à un site de vente d'objets divers sauf que dans ce cas c'est pour un emploi, d'où l'aisance d'utilisation étant le but principal visé. Le système de recherche de mot clés et de recherche dans le titre et le contenu permettant de trouver directement ce que l'on cherche accélère la recherche et réduit le temps perdu à chercher les informations qui nous intéressent sur une interface trop chargée ou peu intuitive.

Tout le site a été conçu de sorte qu'une nouvelle fonctionnalité quelconque puisse être implémentée avec aisance dû à l'utilisation d'un système "Modèle-Vue". Ce qui ne restreint en aucun cas quiconque souhaitant d'avantage agrandir la quantité d'outils fournis à l'utilisateur ou pour de futures mises à jour du site sans empêcher le fonctionnement général pour autant.

### 3.3 État actuel

---

À l'heure actuelle le site permet aux utilisateurs d'utiliser une bonne quantité de fonctionnalités diverses comme :

Pour un utilisateur non connecté:

- Se connecter à un compte existant
- S'inscrire

Pour un utilisateur de type "Chercheur":

- Peut parcourir l'ensemble des annonces disponible.

- Effectuer des recherches ciblées à l'aide de mots-clés et de recherche par titre et contenu d'annonce.
- Ajouter les annonces intéressantes à une "Wishlist".

Pour un utilisateur de type "Annonceur":

- Peut parcourir l'ensemble des annonces créées par ce dernier.
- Effectuer des recherches ciblées comme pour le chercheur mais sur ses propres annonces.
- Modification/Suppression de n'importe laquelle de ses annonces.
- Liste des personnes ayant ajouté une de ses annonces à leur Wishlist ainsi que leur email.

Pour un utilisateur de type "Administrateur":

- Peut gérer l'ensemble des utilisateurs en ayant la possibilité de modifier le type et leur mot de passe dans le besoin.
- Peut gérer l'ensemble des mots-clés pouvant être utilisés par la suite lors de la création d'annonces et recherche de ces dernières.

## 4. Méthodologie

---

Afin d'effectuer ce travail j'ai choisi comme méthodologie de travail la méthode "Waterfall". Elle consiste à répartir le travail à effectuer en phases chacune composées de tâches et est construit de sorte que la tâche suivante ne peut être effectuée que si la précédente a été accomplie avec succès. Pour mon projet les 6 phases sont les suivantes :

1. Récolte d'informations
2. Documentation
3. Décisions
4. Réalisation
5. Contrôles
6. Évaluation

### Récolte d'informations

---

Dans cette phase, je lis l'énoncé, je m'assure d'avoir bien compris ce qui m'est demandé et ensuite je cherche des exemples existant semblables à ce qui m'est demandé.

### Documentation

---

Dans cette phase, je définis les tâches à effectuer, combien de temps elles me prendront, je crée le GitHub, je cherche des templates à utiliser, j'établis les maquettes du site et je rédige la documentation (Ce dernier point est une tâche répétée jusqu'à la fin du projet).

### Décisions

---

Dans cette phase, je réfléchis aux technologies dont je vais avoir besoin pour réaliser le projet.

### Réalisation

---

Dans cette phase, je vais implémenter et programmer les différentes fonctionnalités et vues nécessaires au fonctionnement du projet.

### Contrôles

---

Dans cette phase, je vais rédiger et effectuer une batterie de tests sur mon projet et m'assurer que tout fonctionne comme prévu.

# Évaluation

---

Dans cette dernière phase, je rédige le bilan du projet, la conclusion, le rapport et je maintiens le journal de bord de manière continue durant tout le projet jusqu'au dernier jour.

## 5. Rappel du cahier des charges

---

### 5.1 Objectif

---

L'objectif de cette application est de fournir un outil de mise en relation annonceurs / chercheurs pour des annonces de travail.

### 5.2 Matériel à disposition

---

- Un PC standard école, 2 écrans
- Windows 10
- WAMP ou WSL
- IDE à Choix
- Suite Office (rédaction rapport)

### 5.3 Description de l'application

---

L'objectif de cette application est de fournir un outil de mise en relation annonceurs / chercheurs pour des annonces de travail.

#### Types d'utilisateurs

L'application dispose de 3 types d'utilisateurs :

- Les annonceurs : peuvent déposer des annonces
- Les chercheurs : peuvent parcourir, rechercher et sauvegarder les annonces disponibles sur le site
- Les administrateurs pourront gérer l'ensemble des données et des ressources disponibles

#### Fonctionnalités

L'application permet de déposer des annonces, avec des dates de publication, de les agrémenter de média et de mot clef pour faciliter la recherche. Les chercheurs peuvent sauvegarder les annonces qui les intéressent.

L'annonceur peut voir combien de fois son annonce est « wishlistée ».

#### Une annonce

Une annonce de job est composée des informations suivantes :

- Des dates de publication : début et fin L'annonce est visible par les chercheurs uniquement durant la période de visibilité
- Un titre (255 caractères maxi)
- Un descriptif : texte libre au format HTML
- Un média (optionnel) de type pdf ou image
- Un ou des mots clefs parmi ceux fournis par l'application

## Création de comptes

Tout utilisateur peut créer un compte soit de type annonceur, soit de type chercheur. Les fonctionnalités étant différentes. Le processus de création se base sur un login / password simple. Dans le cadre du TPI il ne sera pas implémenté la restauration de mot de passe.

## Vues pour les annonceurs

Les annonceurs disposent des vues / fonctions suivantes :

- Vue de liste de mes annonces avec : la plage de publication, le titre, les mots clefs et le nombre de «wishlist»
- Ajouter une annonce
- Vue de détail pour une annonce avec le login et la date pour chaque follower (ceux qui ont wishlisté l'annonce)
- Modifier une annonce
- Supprimer une annonce

## Vues pour les chercheurs

Les chercheurs disposent des vues / fonctions suivantes :

- La liste des annonces triées par date de publication.
- Une fonction de recherche portant sur : le titre, le contenu, les mots clefs et le résultat de la recherche se fera sous forme d'une liste d'annonces
- Les vues de détails pour les annonces avec un lien sur le média si disponible.
- Ajouter une annonce à leur wishlist (depuis la vue de détail)
- Gérer leur wishlist

## Vues pour les administrateurs

Les administrateurs disposent des vues / fonctions suivantes :

- Gérer les utilisateurs. (type et mot de passe)
- Gérer (CRUD) la liste des mots clefs

## Remarques générales

- Le programme doit respecter les règles de codage, MV au minimum
- Utilisation de bootstrap pour la réalisation d'un template responsive (le design n'est pas important mais l'implémentation doit respecter la structure de bootstrap)

## Modèle de données

Le candidat peut être amené à apporter des modifications sur le modèle après discussion avec son formateur.

## Manuel utilisateur

La documentation utilisateur sera intégrée dans l'application. Elle sera présentée comme aide directement dans les différents formulaires. Pour les autres informations, le candidat mettra en place une page FAQ statique.

## 5.4 Livrables

---

Le candidat livrera par email, au format PDF, les documents suivants aux experts lors de la fin de son TPI



- Plannings (prévisionnel et réel)
- Résumé A4
- Rapport de projet
- Source code (pdf)
- Export pdf de la FAQ et des captures d'écran présentant l'aide intégrée aux formulaires

Le candidat livrera également à son maître d'apprentissage :

- Idem que pour les experts
- Journal de travail
- Source code et scripts SQL + procédure de mise en production

## 6. Planification

### 6.1 Planning Prévisionnel

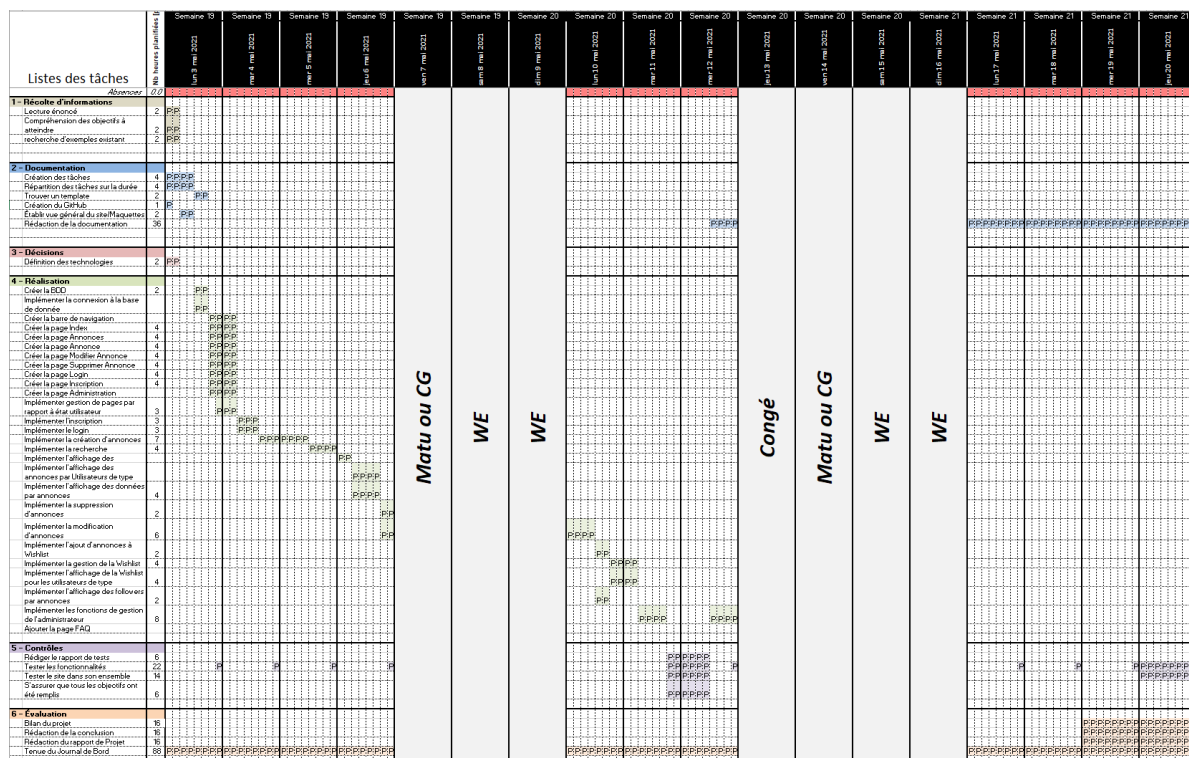


Fig 2. Planning prévisionnel

### 6.2 Planning Réel

### 6.3 Maquettes

#### Accueil

### Accueil

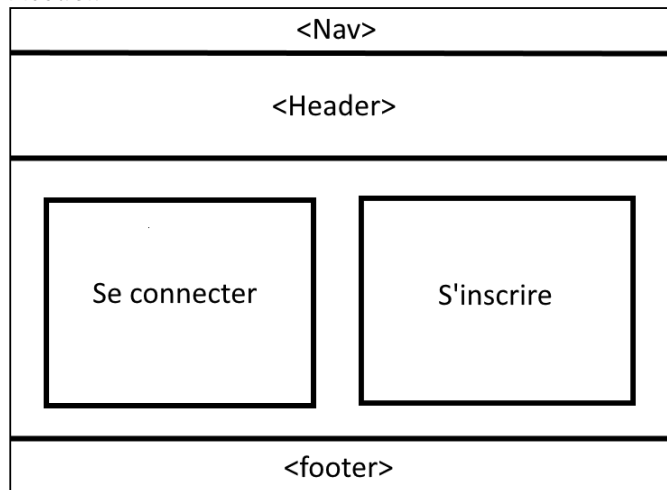


Fig 3. Maquette de l'accueil (Non connecté)

### Accueil (Chercheur)

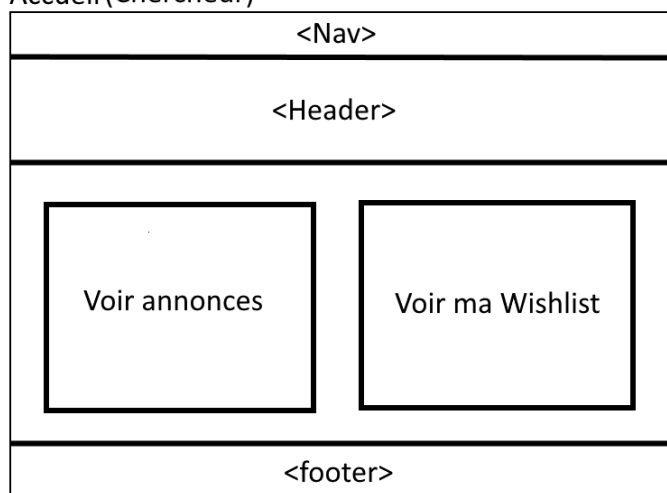


Fig 4. Maquette de l'accueil pour le Chercheur

### Accueil (Annonceur)

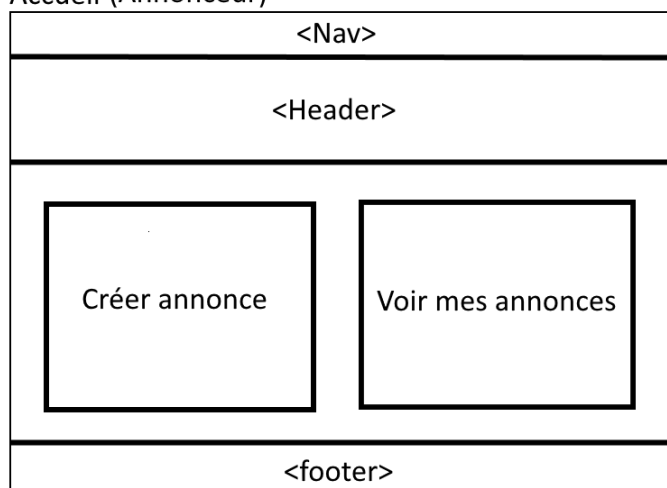


Fig 5. Maquette de l'accueil pour l'annonceur

## Accueil (Administrateur)

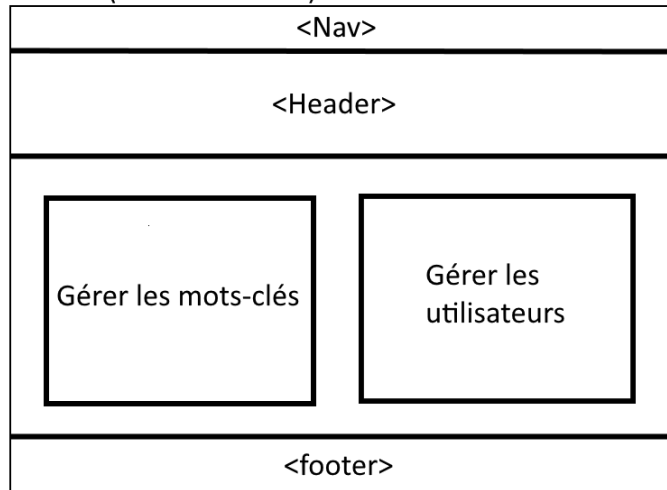


Fig 6. Maquette de l'accueil pour l'administrateur

## Login

### Login

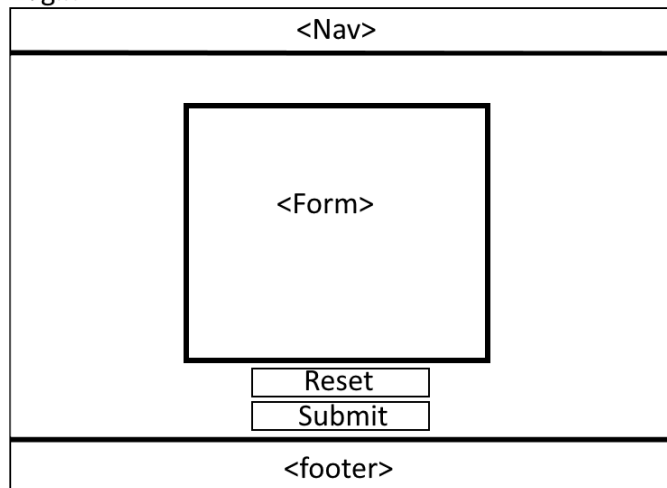


Fig 7. Maquette de la page login

## Inscription

### Inscription

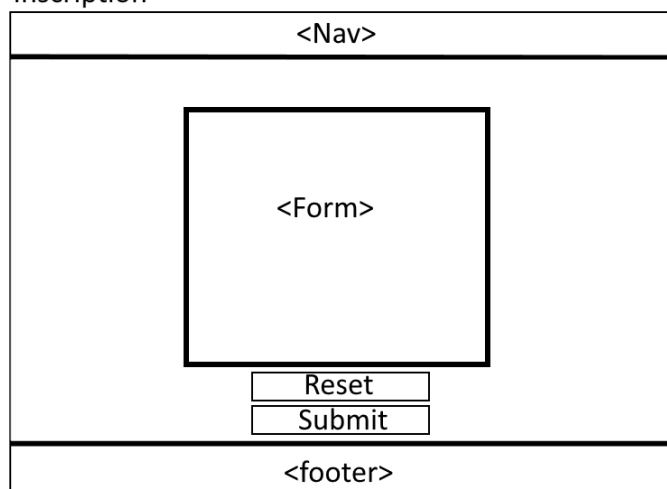


Fig 7. Maquette de la page inscription

## Annonces

Annonces

<Nav>
<Header>
<div>&lt;input&gt; &lt;input&gt; &lt;input&gt; Submit</div> <div>Annonce</div> <div>Annonce</div> <div>Plus d'annonces</div>
<footer>

Fig 8. Maquette de la page annonces pour le chercheur

## Mes annonces

Mes annonces

<Nav>
<Header>
<div>&lt;input&gt; &lt;input&gt; &lt;input&gt; Submit</div> <div>Annonce <a href="#">Modifier</a> <a href="#">Supprimer</a></div> <div>Annonce <a href="#">Modifier</a> <a href="#">Supprimer</a></div> <div>Plus d'annonces</div>
<footer>

Fig 9. Maquette de la page Mes Annonces pour l'annonceur

## Modifier Annonce

Modifier annonce

<Nav>
<div>&lt;Form&gt;</div> <div>Reset</div> <div>Submit</div>
<footer>

Fig 10. Maquette de la page Modifier annonce

## Wishlist

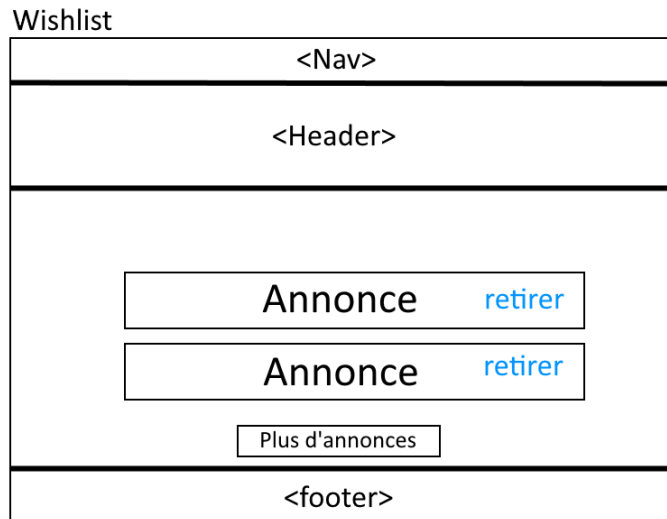


Fig 11. Maquette de la page Wishlist

## Administration : Gestion des Mots-Clés

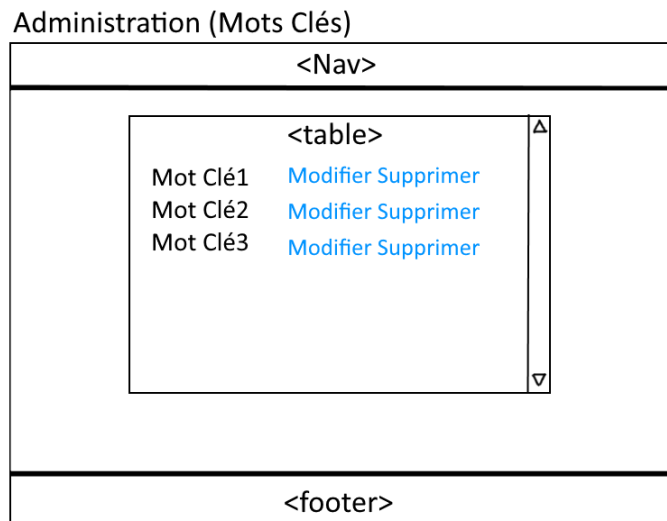


Fig 12. Maquette de la page Administration pour la gestion des mots-clés

## Administration : Gestion des Utilisateurs

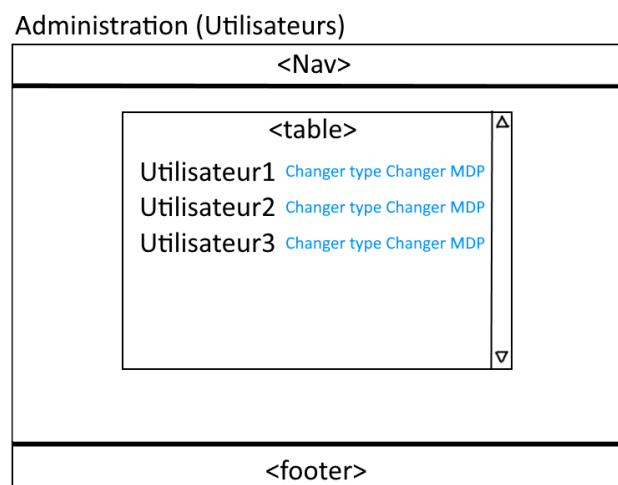


Fig 13. Maquette de la page Administration pour la gestion des Utilisateurs

# 7. Analyse Fonctionnelle

---

## 7.1 Fonctionnalités

---

### 7.1.1 Description Globale

L'ensemble des fonctionnalités présente sur le site actuellement :

- Connexion d'un utilisateur
- Inscription d'un nouvel utilisateur
- Recherche d'annonces par titre et contenu
- Recherche d'annonces par mots-clés
- Affichage d'annonces
- Créer une annonce
- Ajout de média à une annonce
- Retrait de média d'une annonce
- Mise à jour du média d'une annonce
- Modification d'annonces
- Suppression d'annonces
- Affichage d'informations d'annonce
- Ajouter à la wishlist utilisateur
- Afficher la wishlist d'un utilisateur
- Afficher les utilisateurs ayant ajouté une annonce à leur wishlist
- CRUD Mots-Clés
- Gestion des Utilisateurs

### 7.1.2 Description des Fonctionnalités

#### Connexion d'un utilisateur

Afin de pouvoir se connecter l'utilisateur est obligé de rentrer son email et son mot de passe. Ces données sont ensuite envoyées au serveur qui va les traiter. Dans le cas où :

- Le mot de passe est erroné
- L'email fourni n'existe pas dans la base de donnée

Une erreur est affichée. Sinon l'utilisateur est connecté et renvoyé à la page d'accueil.

#### Inscription d'un nouvel utilisateur

Afin de pouvoir s'inscrire, l'utilisateur est obligé de fournir

- Un email (L'email ne doit pas être déjà utilisé par un autre compte)
- Un mot de passe
- Une confirmation du mot de passe
- Choisir entre Annonceur et Chercheur

Une fois cela fait, les données sont envoyées au serveur qui va les traiter. Dans le cas où :

- L'email existe déjà dans la base de donnée
- Les mots de passes ne correspondent pas
- L'email ne correspond pas aux standards de mails
- Le type fourni n'existe pas

Une erreur est affichée. Sinon l'utilisateur est inscrit dans la base de donnée et ensuite est connecté puis renvoyé à la page d'accueil.

## **Recherche d'annonces par titre et contenu**

Lorsque l'utilisateur se trouve sur la page Annonces ou Mes Annonces, il a l'option d'effectuer une recherche sur le titre et la description des annonces existantes. Si l'utilisateur entre un à plusieurs caractères ou mots dans le champ en question et qu'il presse le bouton Chercher, les données du champ sont envoyées au serveur qui va ensuite parcourir toutes les annonces à la recherche d'annonces comportant ce qui a été fourni dans le champ par l'utilisateur et lui renvoyer les annonces correspondantes.

## **Recherche d'annonces par mots-clés**

Lorsque l'utilisateur se trouve sur la page Annonces ou Mes Annonces, il a l'option d'effectuer une recherche sur la liste de mots-clés des annonces existantes. Si l'utilisateur choisit un à plusieurs mots-clés et qu'il presse le bouton Chercher, les données du select multiple contenant les mots-clés sont envoyées au serveur qui va ensuite parcourir toutes les annonces à la recherche d'annonces comportant un à plusieurs des mots-clés fournis par l'utilisateur et lui renvoyer les annonces correspondantes.

## **Affichage d'annonces**

Lorsque l'utilisateur se trouve sur la page Annonces ou Mes Annonces et que ce dernier a effectué une recherche ou non, les résultats de cette recherche vont renvoyer une liste d'annonces à afficher et une fois traitée sera affichée à l'utilisateur. Même si l'utilisateur n'effectue pas de recherche particulière, les annonces les plus récentes seront affichées.

Si l'utilisateur presse le bouton Plus d'Annonces, d'avantage d'annonces seront affichées à l'écran par une certaine quantité définie sur le serveur.

Si aucune annonce n'est trouvée, un simple message est affiché expliquant qu'aucune annonce ne correspond à la recherche.

## **Créer une Annonce**

Un annonceur souhaitant créer une nouvelle annonce se rends sur la page créer annonce et doit ensuite fournir une certaine quantité d'informations avant de pouvoir créer l'annonce :

- Un titre d'annonce
- Une description d'annonce
- Un à plusieurs mots-clés
- (Optionnel) Un média de type PDF ou une image (PNG, JPEG/JPG ou BMP)

Une fois que toutes ces informations sont fournies et que l'utilisateur presse le bouton créer annonce, le tout est envoyé au serveur. Dans le cas où :

- Un des champs obligatoires n'est pas rempli
- un ou plusieurs mots-clés fournis n'existent pas dans la base de donnée
- Le média fourni ne correspond à aucun des types valides

Une erreur est affichée. Sinon, l'annonce est créée et l'utilisateur est redirigé vers la page Mes Annonces.

## Ajout d'un média à une annonce

Lors de la création d'une annonce ou la modification d'une annonce, l'annonceur peut fournir un média de type PDF ou une Image (JPEG/JPG, PNG ou BMP) et ce média sera envoyé au serveur afin d'être traité. Dans le cas où :

- Le média est trop lourd
- Le type de média ne correspond pas

Une erreur est affichée. Sinon, le média est enregistré sur le serveur et dans la base de donnée dans l'annonce en question.

## Retrait de média d'une annonce

Lors de la modification d'une annonce par un annonceur, ce dernier peut décider de retirer le média actuel de l'annonce. Une fois la case cochée et le bouton de confirmation pressé, la requête va être envoyée au serveur qui va supprimer de la base de donnée et du serveur le média et renvoyer l'annonceur vers la page Mes Annonces si la requête a bien réussi ou sinon afficher un message d'erreur.

## Mise à jour du média d'une annonce

Lors de la modification d'une annonce par un annonceur, ce dernier peut décider de mettre à jour le média de son annonce et comme dans le procédé d'ajout de média il peut donc fournir un média du bon type et ensuite ce dernier sera envoyé au serveur qui va procéder à mettre à jour le média de l'annonce tout en supprimant l'ancien et renvoyer l'annonceur vers la page Mes Annonces ou afficher une erreur.

## Modification d'annonces

Un annonceur souhaitant modifier une annonce peut cliquer sur modifier sur l'annonce correspondante en étant sur la page Mes Annonces et va avoir la possibilité de modifier les informations de l'annonce qui seront envoyées au serveur afin d'être traitées. Dans le cas où :

- Un ou plusieurs champs ne sont pas remplis
- Le média fournit n'est pas du bon type
- Un ou plusieurs tags sélectionnés ne sont pas du bon type

Une erreur est affichée, sinon l'utilisateur est redirigé vers la page Mes Annonces.

## Suppression d'annonces

L'annonceur souhaitant supprimer se rends sur la page Mes Annonces où il peut cliquer sur supprimer à coté de l'annonce voulue et une popup va apparaître demandant la confirmation de suppression et si l'utilisateur presse OK, l'annonce est supprimée.

## Affichage d'informations d'annonce

Lorsque l'utilisateur clique sur une annonce, en arrivant sur la page de l'annonce en question les informations transmises sur l'annonce dans l'url sont envoyées au serveur qui va récupérer les informations de l'annonce avant de renvoyer la page à l'utilisateur contenant les informations de l'annonce.



## Ajouter à la wishlist utilisateur

Lorsqu'un chercheur se trouve sur la page d'informations d'une annonce, s'il ne possède pas déjà l'annonce dans sa wishlist un bouton est affiché. S'il le presse, l'information d'ajout de l'annonce à sa wishlist est transmise au serveur qui va effectuer l'ajout et si l'ajout réussit, l'utilisateur reçoit un message de confirmation ou un message d'erreur.

## Afficher la wishlist d'un utilisateur

Un utilisateur souhaitant consulter sa wishlist peut cliquer sur Ma Wishlist afin d'être redirigé vers la page en question qui va récupérer les informations de l'utilisateur et lui renvoyer la page avec les annonces contenues dans sa wishlist affichées.

## Afficher les utilisateurs ayant ajouté une annonce à leur wishlist

Un annonceur souhaitant visionner quels utilisateurs ont ajouté une de ces annonces à leur wishlist peut se rendre sur la page de l'annonce en question et au moment où il se rends sur la page, les données le concernant sont envoyées au serveur qui va lui retourner la page avec les informations concernant les utilisateurs ayant ajouté son annonce à leur wishlist d'affichées en plus de celles de l'annonce.

## CRUD Mots-Clés

Lorsqu'un Administrateur se trouve sur la page Gestion des Mots-Clés, il a plusieurs options mises à disposition dont :

- La création de nouveau Mots-Clés
- La modification de Mots-Clés existant
- La suppression de Mots-Clés existant

L'information en rapport avec l'action entreprise par l'utilisateur est ensuite envoyée au serveur qui va traiter la demande. Dans le cas où :

- Un champ de Mot-Clé existant est laissé vide
- L'id d'un Mot-Clé n'est pas fourni
- Un problème de connexion au serveur survient

Une erreur est affichée, sinon un message de confirmation est affiché à la place.

## Gestion des utilisateurs

Lorsqu'un Administrateur se trouve sur la page Gestion des Utilisateurs, il a quelques options mises à disposition dont :

- La mise à jour du type d'un ou plusieurs utilisateurs
- La mise à jour du mot de passe d'un ou plusieurs utilisateurs

Les informations sont ensuite envoyées au serveur qui va traiter ces dernières. Dans le cas où :

- L'id d'un ou plusieurs utilisateurs n'est pas donné
- L'id d'un ou plusieurs utilisateurs n'existe pas dans la base de donnée
- Un ou plusieurs des types n'existe pas dans la base de donnée

Une erreur est affichée, sinon un message de confirmation est affiché à la place

## 7.2 Interfaces

---

## 7.2.1 Index

La page index a pour principale vocation celle de rediriger l'utilisateur vers les pages auxquelles il a accès et d'une manière plus présentable que la barre de navigation étant la première vue qu'il verra en se connectant.

### Index

Sur cette page, l'utilisateur **Non connecté** peut décider de se rendre sur la page "Se connecter" ou "Créer un compte".



Fig 14. Page Index en étant non connecté

### Index Chercheur

Sur cette page, l'utilisateur de type **Chercheur** peut décider de se rendre sur la page "Annonces" ou "Ma Wishlist".

# Trouvez votre Job rêvé !!!



PARCOURIR LES ANNONCES



GÉRER MA WISHLIST

©Copyright 2021 Search & Find Job | Programmé par Rodrigues F.  
©Copyright 2018 Jober Desk | Template par [ThemezHub](#)

Fig 15. Page Index pour l'utilisateur de type Chercheur

## Index Annonceur

Sur cette page, l'utilisateur de type **Annonceur** peut décider de se rendre sur la page "Créer une annonce" ou "Mes Annonces".

# Trouvez votre Job rêvé !!!



CRÉER UNE ANNONCE



PARCOURIR MES ANNONCES

©Copyright 2021 Search & Find Job | Programmé par Rodrigues F.  
©Copyright 2018 Jober Desk | Template par [ThemezHub](#)

Fig 16. Page Index pour l'utilisateur de type Annonceur

## Index Administrateur

Sur cette page, l'utilisateur de type **Administrateur** peut décider de se rendre sur la page "Gestion d'utilisateurs" ou "Gestion de mots-clés".



Fig 17. Page Index pour l'utilisateur de type Administrateur

### 7.2.2 Barre de Navigation

Elle contient les divers pages accessibles par l'utilisateur. Lorsqu'un utilisateur se rends sur un des liens présent sur la barre de navigation, ce dernier est mis en surbrillance.

#### Barre de Navigation

Cette barre n'est visible uniquement que pour l'utilisateur lorsqu'il n'est pas connecté.



Fig 18. Barre de Navigation lorsque l'utilisateur n'est pas connecté

Dépendant du type d'utilisateur, le contenu de la barre de navigation va changer :

#### Barre de Navigation du Chercheur



Fig 19. Barre de Navigation du Chercheur

#### Barre de Navigation de l'Annonceseur

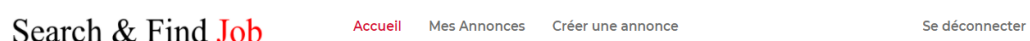


Fig 20. Barre de Navigation de l'Annonceseur

## Barre de Navigation de l'Administrateur

Search & Find **Job**

Accueil

Gérer les utilisateurs

Gérer les mots-clés

Se déconnecter

Fig 21. Barre de Navigation de l'Administrateur

### 7.2.3 Annonces

#### Annonces

- Effectuer une recherche :
  - Parmi les noms d'annonces existantes
  - Parmi le contenu des annonces existantes
  - En sélectionnant un à plusieurs mots-clés
- Accéder à l'annonce voulue en cliquant dessus
- Voir la date à laquelle l'annonce a été créée

Search & Find **Job**

Accueil

**Annonces**

Ma Wishlist

Se déconnecter

Trouvez votre job rêvé ! ! !

#### Cherchez votre annonce

Rechercher par:

Titre ou description

Mots-Clés

Rechercher une annonce

Nothing selected

Rechercher

Annonce1

🕒 2021-05-18 13:26:04

PLUS D'ANNONCES

©Copyright 2021 Search & Find Job | Programmé par Rodrigues F.  
©Copyright 2018 Jober Desk | Template par **ThemezHub**

Fig 22. Page Annonces pour l'utilisateur de type Annonceur uniquement

#### Mes annonces

Cette page n'est disponible uniquement que pour les utilisateurs de type **Annonceur**. Sur cette page l'annonceur peut

- Faire les mêmes actions que le Chercheur dans Annonces

Cependant il peut aussi :

- Modifier ses annonces en appuyant sur modifier sur chaque annonces qu'il souhaite modifier
- Supprimer ses annonces en appuyant sur supprimer sur chaque annonces qu'il souhaite supprimer
- Voir la date de début, de fin et à laquelle l'annonce a été créée

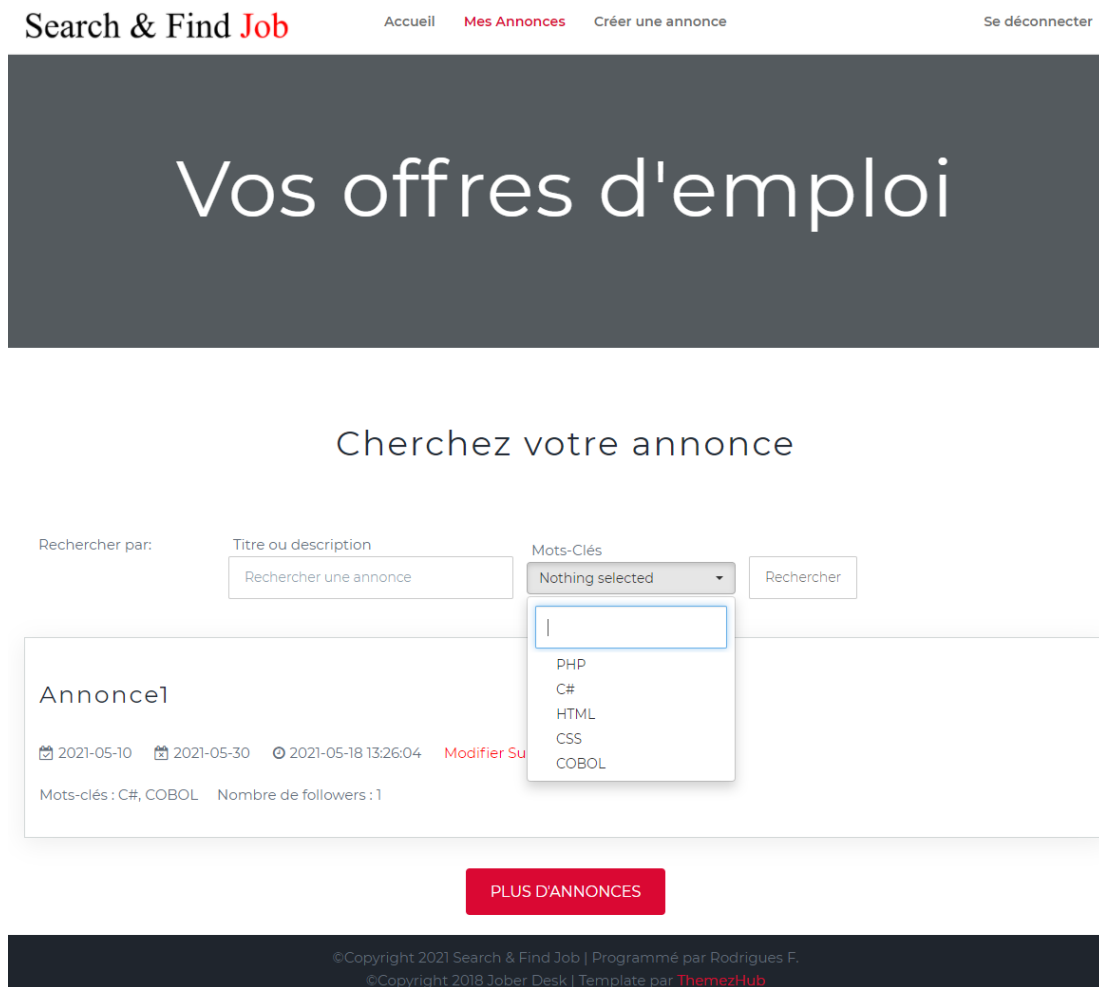


Fig 23. Page Annonces pour l'utilisateur de type Annonceur uniquement

## 7.2.4 Annonce

La page Annonce permet de consulter les informations d'une annonce. Si un média a été inclut avec cette dernière, l'utilisateur consultant l'annonce peut cliquer sur un bouton pour télécharger le média.

### Annonce (Chercheur)

Le chercheur possède un bouton pour ajouter l'annonce à sa wishlist (Si ce n'est pas déjà le cas, sinon le bouton ne s'affiche pas).

## Annonce1

Description

Test

Ajouter annonce à wishlist



Télécharger l'image

Fig 24. Page Annonce pour l'utilisateur de type Chercheur uniquement

### Annonce (Annonceur)

L'annonceur lui peut voir une liste de tous ceux qui ont ajouté son annonce à leur wishlist. Leur email ainsi que la date à laquelle ils l'ont ajouté sont fournis.

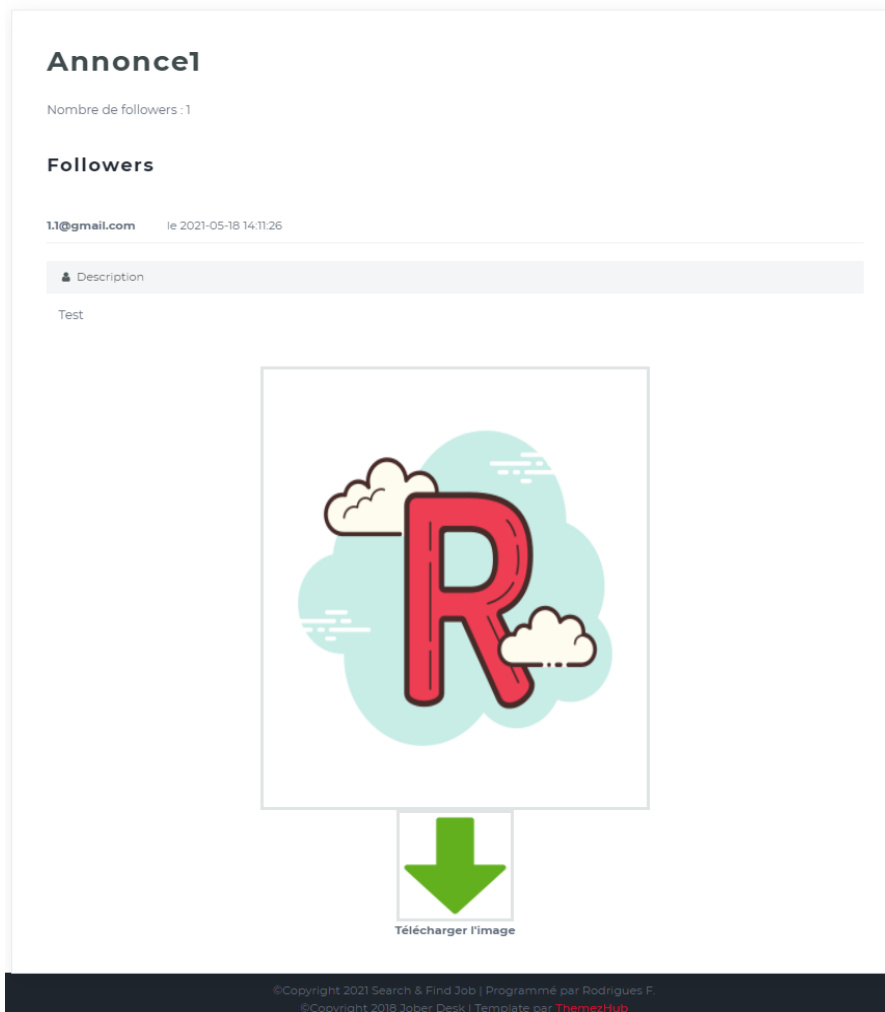


Fig 25. Page Annonce pour l'utilisateur de type Annonceur uniquement

## 7.2.5 Créer Annonce

L'annonceur, afin de créer une annonce, doit fournir les informations suivantes

- Un nom d'annonce
- Une description
- Une Date de Début et de Fin d'annonce
- un à plusieurs Tags
- (Optionnel) Une image ou PDF

Une fois tout cela remplis, l'Annonceur pourra créer une annonce.



## Créez votre annonce

**Nom de l'annonce****Description de votre annonce****Date de début de votre annonce****Date de fin de votre annonce****Les tags de votre annonce (Veuillez en sélectionner 1 à plusieurs)****Média souhaitant être inclu à votre annonce (Image ou un fichier PDF) (Optionnel)**

RESET

CRÉER ANNONCE

©Copyright 2021 Search & Find Job | Programmé par Rodrigues F.  
©Copyright 2018 Jober Desk | Template par [ThemezHub](#)

Fig 26. Page Créer une annonce pour l'utilisateur de type Annonceur uniquement

## 7.2.6 Modifier Annonce

Cette page est uniquement accessible par un utilisateur de type Annonceur et sur une annonce qu'il a lui-même créé.

Une fois sur cette page l'utilisateur a plusieurs options de modification qui se présentent à lui :

- Changer le nom de l'annonce
- La description de l'annonce
- La date de Début et/ou de Fin de l'annonce
- Les tags de l'annonce
- Changer le média de l'annonce ou en ajouter un
- (Seulement si un média est déjà présent) Retirer le média de l'annonce

**Nom de l'annonce**

  
**Description de votre annonce**

**Date de début de votre annonce**

**Date de fin de votre annonce**

**Les tags de votre annonce (Veuillez en sélectionner 1 à plusieurs)**

**Média souhaitant être inclut à votre annonce (Image ou un fichier PDF) (Optionnel)**

 Aucun fichier choisi

**Supprimer le média actuel (À cocher si oui)**

☐

METTRE À JOUR

©Copyright 2021 Search & Find Job | Programmé par Rodrigues F.  
©Copyright 2018 Jobber Desk | Template par [ThemezHub](#)

Fig 27. Page de modification d'annonce pour l'utilisateur de type Annonceur uniquement

## 7.2.7 Supprimer Annonce

Lorsque l'Annonceur presse le bouton "Supprimer" en étant sur "Mes Annonces", une popup apparaît demandant la confirmation de suppression et si l'utilisateur appuie sur OK, l'annonce est supprimée.

localhost indique

Voulez-vous vraiment supprimer cette annonce ?

Fig 28. Popup de suppression d'annonce sur Mes Annonces pour l'utilisateur de type Annonceur uniquement

## 7.2.8 Login

Afin de se connecter, l'utilisateur doit fournir son adresse email ainsi que son mot de passe.

## Search &amp; Find Job

RESET

SE CONNECTER

Vous n'avez pas de compte ? [Créez un compte](#)

©Copyright 2021 Search &amp; Find Job | Programmé par Rodrigues F.

©Copyright 2018 Jober Desk | Template par [ThemezHub](#)

Fig 29. Page de login pour tout utilisateur n'étant pas connecté

## 7.2.9 Inscription

Afin de s'inscrire, l'utilisateur doit :

- Fournir un email non utilisé sur le site
- Un mot de passe ainsi que le rentrer une deuxième fois
- Sélectionner le type de compte qu'il souhaite créer

Une fois tout cela remplis, il pourra s'inscrire.

## Search &amp; Find Job

-- Veuillez sélectionner une option --

-- Veuillez sélectionner une option --

Chercheur

Annonceur

RESET

S'INSCRIRE

Vous possédez déjà un compte ? [Connectez vous](#)

©Copyright 2021 Search &amp; Find Job | Programmé par Rodrigues F.

©Copyright 2018 Jober Desk | Template par [ThemezHub](#)

Fig 30. Page d'inscription pour tout utilisateur n'étant pas connecté

## 7.2.10 Wishlist

La page Ma Wishlist permet au Chercheur de consulter toutes les annonces qu'il a ajouté à sa wishlist ainsi que d'enlever celles qu'il ne désire plus suivre.



Fig 31. Page Ma Wishlist d'un utilisateur de type Chercheur

## 7.2.11 Gestion Utilisateurs

La page Gestion d'Utilisateurs uniquement accessible par un utilisateur de type Administrateur permet à ce dernier de :

- Modifier le type d'un ou plusieurs utilisateurs entre
  - Chercheur
  - Annonceur
  - Administrateur
- Fournir un nouveau mot de passe à 1 ou plusieurs utilisateurs

## Gestion d'Utilisateurs

Gérez les types et mots de passes des utilisateurs

#	Login	Type	Mot de Passe
1	1.1@gmail.com	Chercheur	Nouveau mot de passe
2	2.2@gmail.com	Annonceur	Nouveau mot de passe
3	3.3@gmail.com	Admin	Nouveau mot de passe
4	4.4@gmail.com	Annonceur	Nouveau mot de passe
5	5.5@gmail.com	Chercheur	Nouveau mot de passe

APPLIQUER LES CHANGEMENTS

©Copyright 2021 Search & Find Job | Programmé par Rodrigues F.

©Copyright 2018 Jober Desk | Template par [ThemezHub](#)

Fig 32. Page Gestion d'Utilisateurs pour un utilisateur de type Administrateur

### 7.2.12 Gestion Mots-Clés

La page Gestion de mots-clés uniquement accessible par un utilisateur de type Administrateur permet à ce dernier de :

- Modifier un à plusieurs labels de mots-clés
- Supprimer un à plusieurs Mots-Clés
- Ajouter un à plusieurs nouveaux mots-clés

## Gestion des Mots-Clés

Ajoutez, modifiez ou supprimez des mots clés

#	Label	Supprimer
8	PHP	<input type="checkbox"/>
9	C#	<input type="checkbox"/>
10	HTML	<input type="checkbox"/>
11	CSS	<input type="checkbox"/>
15	COBOL	<input type="checkbox"/>

Ajouter des mots clés + -

Nouveau Mot-Clé N°0

APPLIQUER LES CHANGEMENTS

©Copyright 2021 Search & Find Job | Programmé par Rodrigues F.

©Copyright 2018 Jobber Desk | Template par [ThemezHub](#)

Fig 33. Page Gestion de mots-clés pour un utilisateur de type Administrateur

# 8. Analyse Organique

## 8.1 Diagramme Général

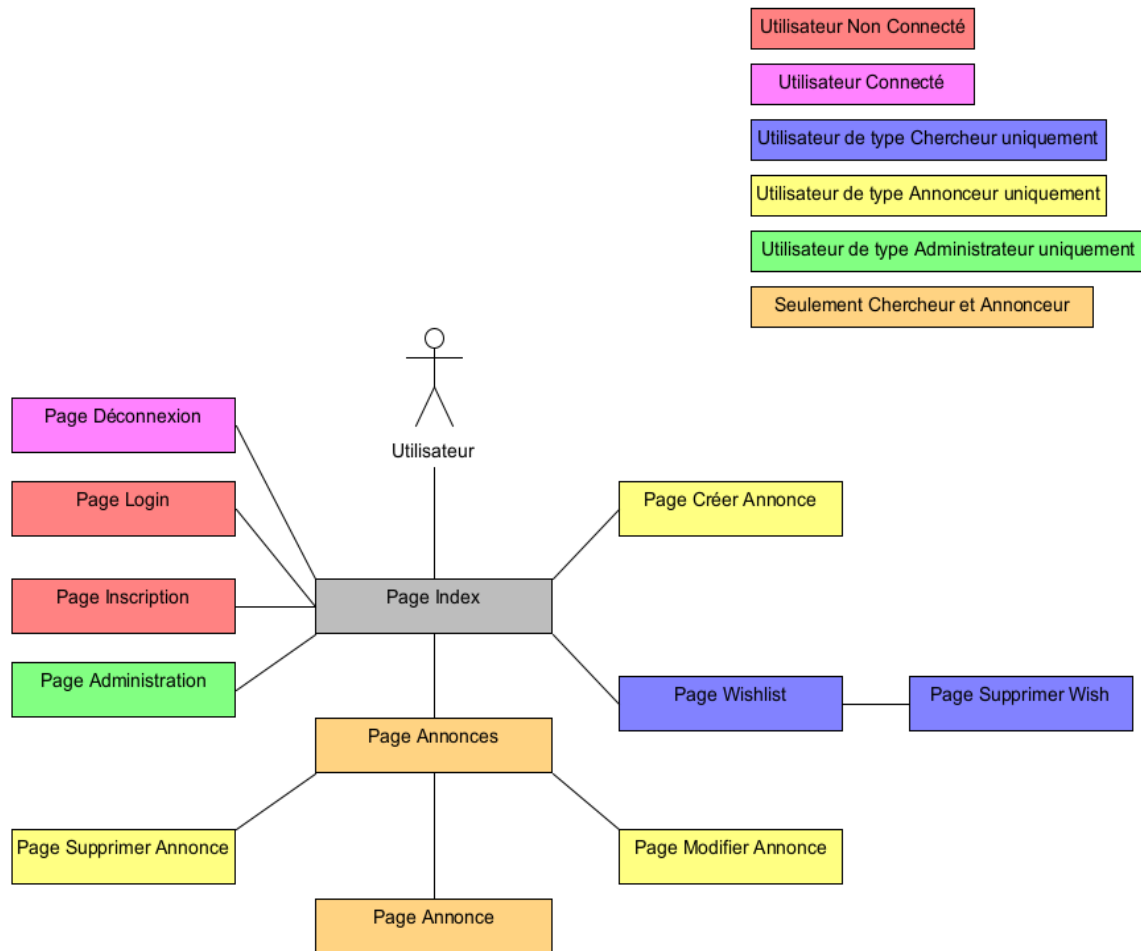


Fig 34. Diagramme général du site

## 8.2 Environnement

Afin de développer ce site Web j'ai fait usage d'un serveur Windows Subsystem Linux (WSL) sur Debian utilisant Apache version 2.4.38 et l'IDE utilisé est Visual Studio Code version 1.55.2.

Pour la création de maquettes je les ai réalisées sur Paint.net. Le serveur utilise PHP version 7.3.27-1. Le serveur de base de données est sous MariaDB et est administré à l'aide de phpMyAdmin version 4.9.7.

Le versionnage de mon projet est géré par Github Desktop qui est l'application permettant de versionner le tout sur le GitHub de mon projet.

La documentation a été rédigée à l'aide de Typora en Markdown et certains Diagrammes à l'aide de UMLet.

## 8.3 Base de donnée

### 8.3.1 MLD de la Base de donnée

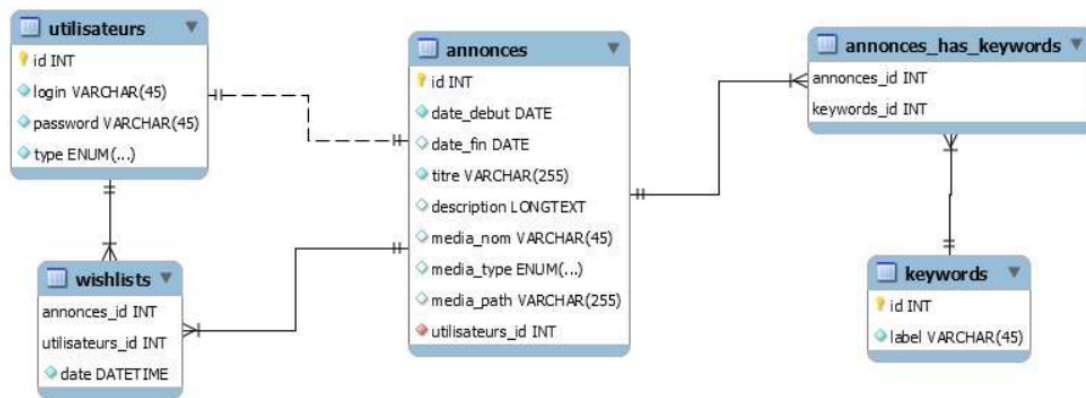


Fig 34. Modèle de la base de donnée utilisée

Ce modèle logique de donnée m'a été fourni dans mon énoncé et je me suis donc basé sur ce dernier pour réaliser la base de donnée.

### 8.3.2 Description des tables

#### Annonces

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	id	int(10)		UNSIGNED	Non	Aucun(e)		AUTO_INCREMENT
2	date_debut	date			Non	Aucun(e)		
3	date_fin	date			Non	Aucun(e)		
4	date_publication	datetime			Non	current_timestamp()		
5	titre	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)		
6	description	longtext	utf8mb4_general_ci		Non			
7	media_path	varchar(255)	utf8mb4_general_ci		Oui	NULL		
8	media_nom	varchar(45)	utf8mb4_general_ci		Oui	NULL		
9	media_type	enum('png', 'pdf', 'jpg', 'jpeg', 'bmp')	utf8mb4_general_ci		Oui	NULL		
10	utilisateurs_id	int(10)		UNSIGNED	Non	Aucun(e)		

Fig 35. Structure de la table Annonces

Cette table sert à stocker les différentes annonces, leurs informations et le créateur de l'annonce.

#### Annonces\_has\_keywords

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	annonces_id	int(10)		UNSIGNED	Non	Aucun(e)		
2	keywords_id	int(10)		UNSIGNED	Non	Aucun(e)		

Fig 36. Structure de la table Annonces\_has\_keywords

Cette table permet de savoir quels mots-clés sont associés à quelles annonces à l'aide de 2 clé étrangères pointant respectivement sur la table Annonces et Keywords.



## Keywords


	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1	id 	int(10)		UNSIGNED	Non	Aucun(e)		AUTO_INCREMENT
<input type="checkbox"/>	2	label	varchar(45)	utf8mb4_general_ci		Non	Aucun(e)		

Fig 37. Structure de la table Keywords

Cette table sert à stocker les différents mots-clés pouvant être utilisé par les utilisateurs.

## Utilisateurs


	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1	id 	int(10)		UNSIGNED	Non	Aucun(e)		AUTO_INCREMENT
<input type="checkbox"/>	2	login	varchar(45)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	3	password	varchar(60)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	4	type	enum("Admin", "Chercheur", "Annonceur")	utf8mb4_general_ci		Non	Aucun(e)		

Fig 38. Structure de la table Utilisateurs

Cette table sert à stocker les utilisateurs ainsi que leurs données.

## Wishlist



	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1	annonces_id 	int(10)		UNSIGNED	Non	Aucun(e)		
<input type="checkbox"/>	2	utilisateurs_id 	int(10)		UNSIGNED	Non	Aucun(e)		
<input type="checkbox"/>	3	date	datetime			Non	current_timestamp()		

Fig 39. Structure de la table Wishlist

Cette table permet de savoir quelles annonces ont été ajoutées à la wishlist de quel utilisateur et à quelle date.

## 8.4 Description des fonctions

### 8.4.1 Requêtes SQL

Afin d'exécuter des requêtes SQL, 2 modèles de fonction ont été implémentés et utilisés selon l'utilisation voulue.

#### Modèle de base

```
function function_name($content1, $content2, $content3, $content4, $content5)
{
    //Déclaration requête
    static $ps = null;
    $sql = "INSERT INTO `table` (`content1`, `content2`, `content3`, `content4`, `content5`) VALUES (:CONTENT1, :CONTENT2, :CONTENT3, :CONTENT4, :CONTENT5)";

    //Déclaration Prepare statement si pas déjà préparé
    if ($ps == null) {
        $ps = db()->prepare($sql);
    }
    $answer = false;
```

```

try {
    //Assignation des paramètres
    $ps->bindParam(':CONTENT1', $content1, PDO::PARAM_STR);
    $ps->bindParam(':CONTENT2', $content2, PDO::PARAM_STR);
    $ps->bindParam(':CONTENT3', $content3, PDO::PARAM_STR);
    $ps->bindParam(':CONTENT4', $content4, PDO::PARAM_STR);
    $ps->bindParam(':CONTENT5', $content5, PDO::PARAM_STR);
    //Exécution, dans ce cas, si l'exécution réussit, la réponse est true
    if($ps->execute())
        $answer = true;
    //Si une erreur survient, l'echo
} catch (PDOException $e) {
    echo $e->getMessage();
}
//Retourne la réponse après l'exécution de la requête
return $answer;
}

```

Ce premier modèle sert à effectuer des méthodes CRUD et consistent tous d'une requête SQL définie dans la variable **\$sql**, cette requête est ensuite préparée si elle n'a pas déjà été préparée lors d'une utilisation ultérieure de la méthode. La préparation servant à récupérer tous les index de la base de donnée une première fois afin de réduire le temps d'exécution pour des requêtes similaires en utilisant une requête identique à chaque exécutions mais du au fait de l'utilisation de paramètres les valeurs fournies peuvent changer entre chaque exécutions.

Les paramètres se voient ensuite assignés leur donnée et pour finir la requête est exécutée et si l'exécution a réussi avec succès, la variable de réponse est définie à **true**, sinon elle reste **false**. Dans le cas où une exception survient, elle est affichée.

## Modèle transaction

```

function function_name($content1, $content2) {
    try {
        //Début de la transaction
        db()->beginTransaction();
        //Déclaration des requêtes
        $sql1 = "INSERT INTO champ1 VALUES(:content1)";
        $sql2 = "INSERT INTO champ2 VALUES(:content2)";
        //Déclaration des prepare statement si ce n'est pas déjà le cas
        static $ps1 = null;
        if($ps1 == null)
            $ps1 = db()->prepare($sql1)

        static $ps2 = null;
        if($ps2 == null)
            $ps2 = db()->prepare($sql2)

        //Assignation de la variable de la première requête
        $ps1->bindParam(':content1', $content1, PDO::PARAM_STR);
        //Exécution de la requête
        $ps1->execute();

        //Assignation de la variable de la première requête
        $ps2->bindParam(':content2', $content2, PDO::PARAM_STR);
        //Exécution de la requête
        $ps2->execute();
    }
}

```

```

        //Fin de la transaction
        getConnexionBDD()->commit();
        //Si aucune erreur n'est survenue, l'exécution de toutes les requêtes a
réussi avec succès et la valeur true est retournée
        return true;
    }
    //Si une erreur survient, toutes les requêtes exécutées sont annulées avec un
rollback et la valeur false est retournée
    catch (Exception $e)
    {
        db()->rollback();
        return false;
    }
}

```

Ce deuxième modèle sert à exécuter de multiples requêtes SQL à la fois et consiste donc de plusieurs requêtes SQL. La fonction commence avec un **beginTransaction()** qui va marquer le début de la transaction. Chacune des requête est définie dans une variable propre à elle et est ensuite préparée si ce n'était pas déjà fait auparavant. Ensuite chaque paramètre de requête se voit assigné ses données et les requêtes sont exécutées une après l'autre. Si aucune erreur ne survient, la transaction est terminée à l'aide d'un **commit()** et la valeur **true** est retournée.

Dans le cas où une erreur serait survenue lors de l'exécution des requêtes, elles sont toutes annulées et la valeur **false** est retournée.

La fonction qui suit se trouve dans db.inc.php :

## db()

Cette fonction permet la connexion à la base de donnée en créant un objet PDO qui possède les informations nécessaires à l'exécution de requêtes sur le serveur. Le code qui suit montre le code utilisé pour créer cet objet.

```

$dbc = new PDO('mysql:host=' . HOST . ';dbname=' . DBNAME, DBUSER, DBPWD,
array(
    PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8",
    PDO::ATTR_PERSISTENT => true

```

Après la création de l'objet, ce dernier est retourné à chaque appels et comme il est stocké en dans une variable static, l'objet n'a plus besoin d'être recréé.

## 8.4.2 Fonctions d'authentification

les fonctions qui suivent sont présentes dans user\_func.inc.php :

### ChangeLoginState()

Cette fonction permet de mettre à jour la variable de session **loggedIn** dépendant de si l'utilisateur est connecté ou non

## VerifyIfMailExists()

Cette fonction permet de chercher dans la base de donnée tous les comptes possédant l'email fournit à l'aide d'une requête SQL et si des résultats sont retournés, l'email existe déjà, sinon l'email n'existe pas.

## ConnectUser()

Cette fonction sert à connecter un utilisateur en testant son email et mot de passe par rapport à ceux dans la base de donnée.

Premièrement, l'email est testé à l'aide de **VerifyIfMailExists()**, étant une requête SQL qui retourne true si des résultats sont retournés ou false si ce n'est pas le cas, pour s'assurer que le mail existe bien dans la base de donnée. Si c'est le cas, la fonction **GetUserInfo** va récupérer les données de l'utilisateur grâce à son email et ensuite la fonction password\_verify() de PHP va tester le mot de passe Hashé récupéré et le mot de passe fournit par l'utilisateur. S'ils correspondent l'utilisateur sera connecté à l'aide de **ChangeLoginState(true)** et la valeur **true** sera renvoyée pour confirmer la réussite de la connexion, sinon la valeur **false** est renvoyée.

## RegisterUser()

Cette fonction sert à enregistrer un nouvel utilisateur dans la base de donnée à l'aide des données fournies par l'utilisateur.

Tout d'abord, à l'aide de **VerifyIfMailExists()** l'email fournit est testé afin de s'assurer qu'il n'existe pas déjà. Ensuite le mot de passe fournit est hashé et la fonction **CreateUser()** est appelée afin de créer le nouvel utilisateur dans la base de donnée. Si la création du compte est réussie, la fonction **ChangeLoginState(true)** est appelée et l'utilisateur est connecté et la valeur true est retournée, sinon la valeur false est retournée.

## 8.4.3 Fonctions d'alertes

les fonctions qui suivent sont présentes dans alert.inc.php :

### SetAlert()

Cette fonction permet de définir le type d'alerte et le numéro correspondant au message voulu en dans des variables en **\$\_POST**.

### GetAlert()

Cette fonction va retourner le type et le numéro de l'alerte actuelle s'il y en a une de définie.

### ShowAlert()

Cette fonction va afficher l'alerte définie en choisissant le bon message par rapport au type d'alerte défini et le numéro.

## 8.4.4 Fonctions d'annonces

les fonctions qui suivent sont présentes dans annonce\_func.inc.php :

## **CheckMedia()**

Cette fonction va traiter le média fournit dans la variable globale **\$\_FILES["media"]** et s'assurer qu'il fait moins de 20mo et qu'il est bien de type PDF, PNG, JPG/JPEG ou BMP. Si le média est traité sans problèmes le chemin, nom et type du média est retourné dans un array, sinon une erreur est définie avec **SetAlert()**. Si aucun média n'est fournit ces 3 valeurs sont définies à null

## **CreerAnnonce()**

Cette fonction est utilisée pour ajouter de nouvelles annonces à la base de donnée grâce aux données fournies par l'annonceur.

## **GetAnnoncesFromSearchAnnonceur()**

Cette fonction va effectuer une recherche dans la base de données et sur les annonces en concordance avec les paramètres de recherche fournit et retourner les annonces trouvées. Cette fonction n'est appelée lors d'une recherche effectuée par un annonceur.

## **GetAnnoncesFromSearchChercheur()**

Cette fonction possède le même procédé que la fonction précédente mais est propre aux recherches des Chercheur

## **UploadMedia()**

Cette fonction lorsqu'appelée va déplacer les fichiers uploadés dans la variable **\$\_FILES["media"]** et les stocker dans un dossier sur le serveur. Si l'opération réussit la valeur true est retournée, sinon false est retournée.

## **8.4.5 Fonctions de Wishlist**

les fonctions qui suivent sont présentes dans wishlist\_func.inc.php :