Contenu

[Contenu 1](#_Toc485296391)

[Introduction 2](#_Toc485296392)

[Préambule 2](#_Toc485296393)

[Description 2](#_Toc485296394)

[Organisation 2](#_Toc485296395)

[Etude d’opportunité 2](#_Toc485296396)

[Pourquoi ce sujet 2](#_Toc485296397)

[Pourquoi le PHP 3](#_Toc485296398)

[Description de l’existant 3](#_Toc485296399)

[Ce que mon projet à de plus 3](#_Toc485296400)

[Analyse fonctionnelle 3](#_Toc485296401)

[Fonctionnalités des utilisateurs 3](#_Toc485296402)

[L’API Google Maps 3](#_Toc485296403)

[Le Géocoding de Google Map 3](#_Toc485296404)

[Interface 4](#_Toc485296405)

[Structure 4](#_Toc485296406)

[Page d’accueil 5](#_Toc485296407)

[Page de connexion 6](#_Toc485296408)

[Page inscription 7](#_Toc485296409)

[Page du profil 8](#_Toc485296410)

[Page ajouter un animal 9](#_Toc485296411)

[Page rechercher 10](#_Toc485296412)

[Page résultats 11](#_Toc485296413)

[Page disponibilités 12](#_Toc485296414)

[La base de données 13](#_Toc485296415)

[Description des tables 14](#_Toc485296416)

[La table adresses 14](#_Toc485296417)

[La table utilisateurs 14](#_Toc485296418)

[La table animaux 14](#_Toc485296419)

[La table especes 15](#_Toc485296420)

[La table peut\_garder 15](#_Toc485296421)

[La table horaires 15](#_Toc485296422)

[La table disponible 15](#_Toc485296423)

[La vue recherche 16](#_Toc485296424)

[Fonctions 16](#_Toc485296425)

[Fonctions de la page phptohtml.php 16](#_Toc485296426)

[La page dao.php 17](#_Toc485296427)

[Le JavaScript 20](#_Toc485296428)

[myMap(lat, lng) 20](#_Toc485296429)

[addMarker(lat, lng, nom) 20](#_Toc485296430)

## Introduction

### Préambule

Le premier animal de compagnie a été le chien issu des loups. Ensuite les animaux de compagnies rares ont été ramenés par des explorateurs ou des colonisateurs augmentant le nombre d’animaux domestiques. Dans les années 1700 les chats et les chiens sont utilisés pour combattre l’invasion des rats. Lorsque cette invasion sera terminée les animaux resteront dans les foyers pour devenir des animaux de compagnies. En 2010 environ 49% des foyers français avait au moins un animal de compagnie pour un total d’environ 59 millions d’animaux de compagnies. En Belgique c’est 50%[[1]](#endnote-1) de la population qui a un animal de compagnies représentant 1,3 millions[[2]](#endnote-2) de mascottes et en suisse c’est 30% pour un peu moins de 2 millions d’animaux de compagnies[[3]](#endnote-3). Cela fait autant d’animaux qui devront potentiellement se faire garder un jour ou plus si leurs propriétaire on des obligations qui leurs empêchent de s’en occuper.

### Description

Le site Buddy-Sitter a pour but d’aider les personnes qui cherchent à faire garder leurs animaux de compagnies. C’est un problème au quelle une majorité des personnes propriétaires d’animaux sont confrontée principalement lors de départ en vacances. Il arrive aussi parfois que des personnes ne puissent pas s’occuper de leurs animaux par exemple juste une journée. Il est difficile de trouver des gardiens dans son entourage et les personnes préfèrent généralement ne pas avoir à payer pour ce genre de service.

### Organisation

Afin d’assurer un bon environnement de travail certaines précaution ont été prises. J’utilise donc GitHub afin de sécuriser le travail effectué, cela permet aussi de voir les versions et les modifications qui ont été faites chaque fois que j’ai sauvegardé les fichiers.

## Etude d’opportunité

### Pourquoi ce sujet

J’ai choisi de faire ce sujet car c’est un site qui est utile et qui facilitera la recherche d’un gardien pour son animal de compagnie. Cette idée m’est venue lorsque ma grand-mère voulait faire garder son beagle. Elle ne trouvait pas de personnes pour garder son chien et je souhaitais l’aider mais je ne savais pas vraiment comment puis j’ai eu cette idée de site web.

### Pourquoi le PHP

J’ai choisi de faire ce projet en PHP car cela me paraissait plus logique que de faire une application qui devrait être téléchargée par tous les utilisateurs. De plus nous n’avons pas vu durant notre formation les fonctionnalités web en C#. La version web était donc la plus logique.

### Description de l’existant

Il existe déjà des sites web permettant de rechercher des gardiens pour ses animaux de compagnie. Ils permettent de s’enregistrer et de faire des recherches pour garder ou pour garder des animaux. Cependant ils ne permettent pas tous d’enregistrer ses animaux.

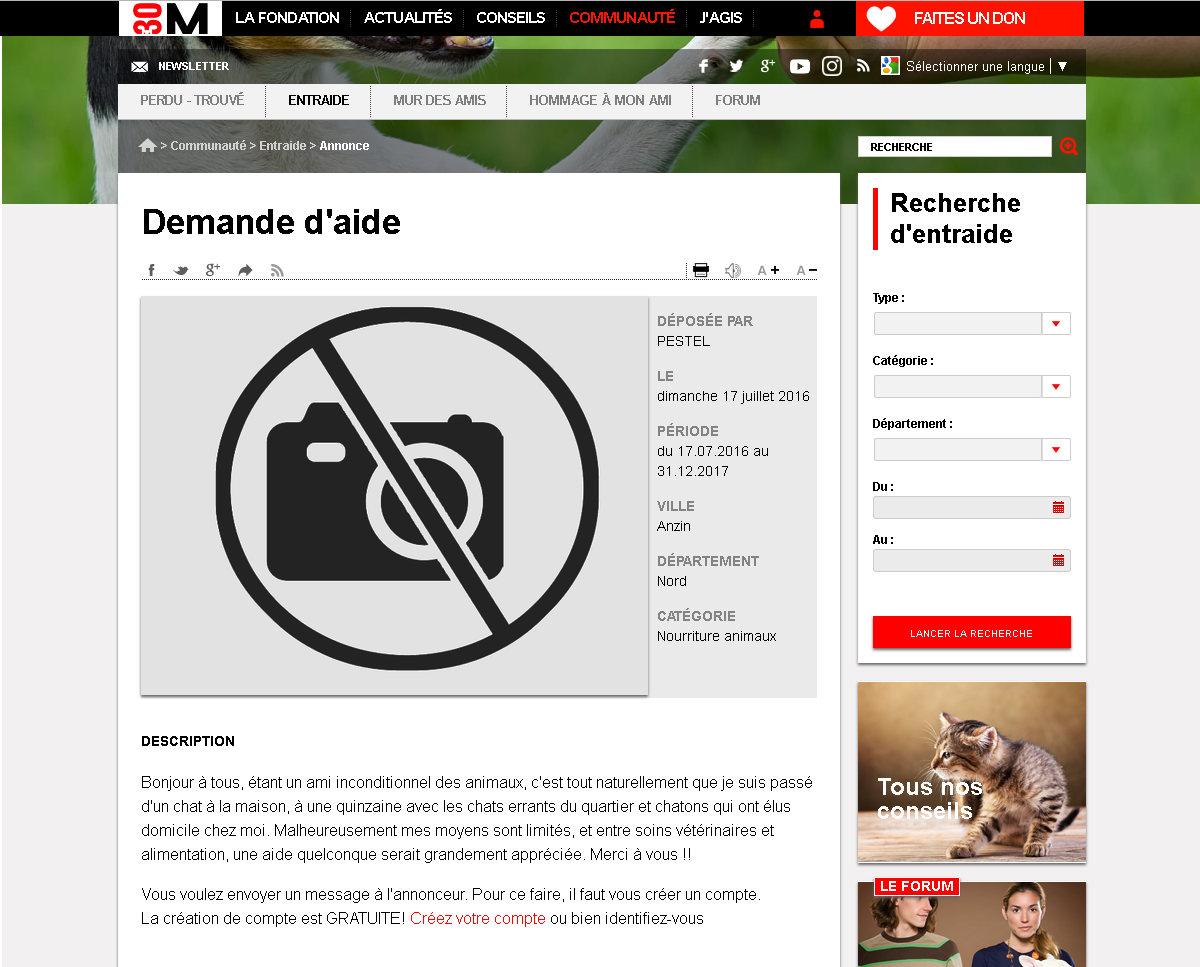
Par exemple le site <http://www.30millionsdamis.fr> propose dans son onglet communauté une section entraide. Cette section propose une carte de la France. En cliquant sur un département nous voyons les demandes d’aide et les propositions d’aide.



Une fois sur la page d’un département il est possible de voir les demandes d’aides et les propositions d’aides, qui peuvent concerner le gardiennage ou d’autres choses

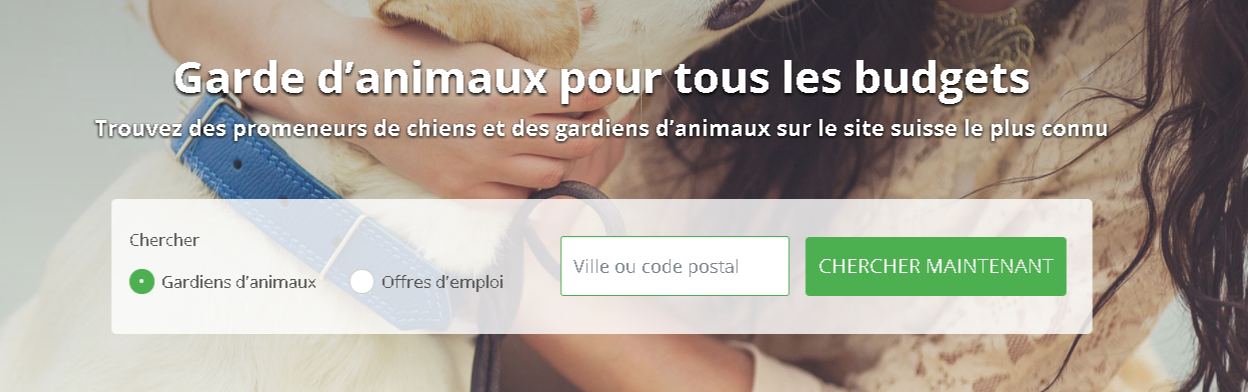


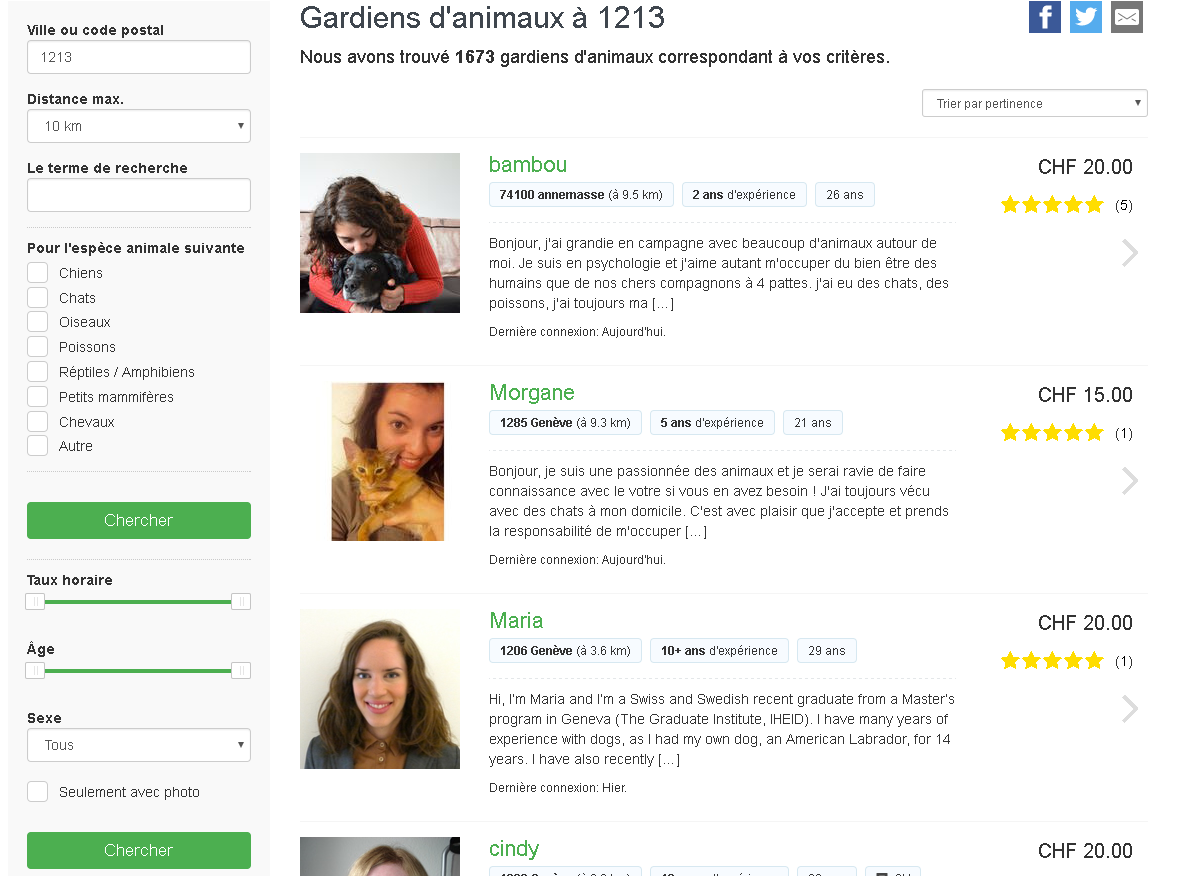
Il est possible de cliquer sur une annonce pour afficher tous les détails.



Sur la page de détail d’une annonce, il est possible de voir une photo si le créateur de l’annonce en a mis une. On peut aussi voir son nom la date à laquelle il à créer l’annonce, la ville ainsi que le département et une catégorie.

Le site <http://petsitting24.ch> permet de rechercher des gardiens ou des personnes qui cherchent à faire garder leurs animaux de compagnie en fonction du code postal principalement.





Sur la partie de droite il est possible d’affiner la recherche. Pour affiner la recherche on peut apporter des précisions générales, concernant l’animal à faire garder ou concernant le gardien.

Pour les précisions générales il est possible de changer le code postal, il est possible de choisir une distance maximum. Le champ « terme de la rechercher » est un champ qui permet de filtrer en général : le mot qui sera inscrit soit dans le nom du gardien ou dans sa description.

Pour les précisions concernant l’animal à faire garder il est possible de choisir l’espèce à laquelle appartient l’animal.

Dans la partie concernant le gardien on peut choisir le salaire par heure qui est demandé, l’age de la personne ainsi que son sexe. Il est aussi possible de cocher une case pour n’afficher que les gardiens qui ont mis une photo.

### Ce que mon projet à de plus

Mon projet ressemble pour beaucoup de choses à certains sites du genre qui existent déjà mais il propose une chose de plus : la géolocalisation. En effet cette dernière est utilisée afin de proposer des gardiens proches du propriétaire de l’animal.

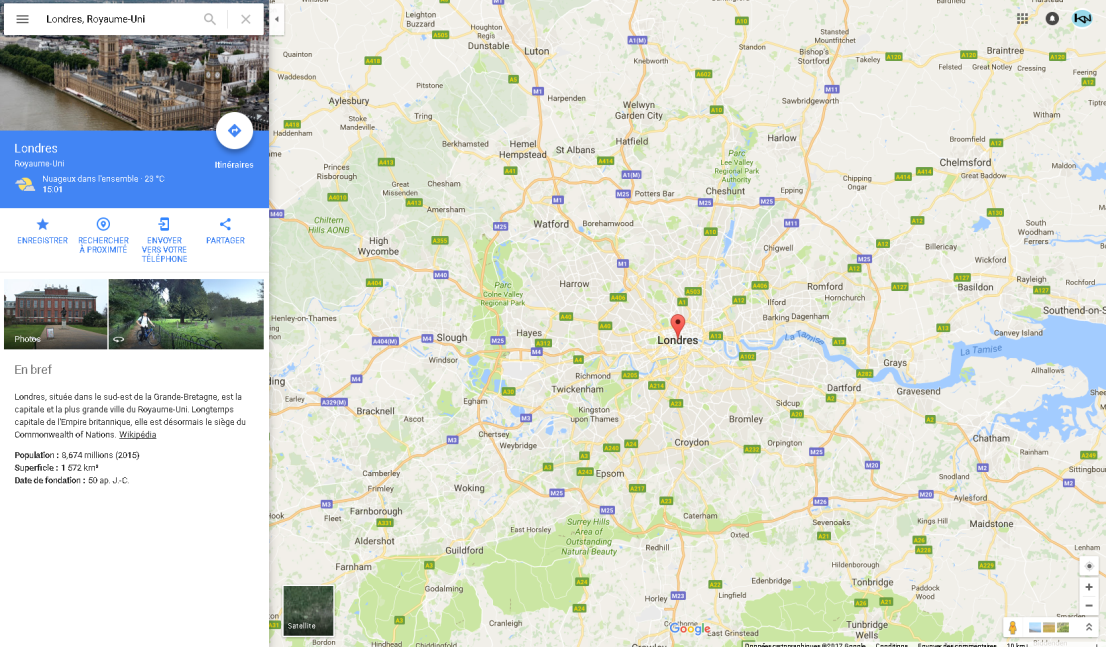
## Analyse fonctionnelle

### Fonctionnalités des utilisateurs

Les utilisateurs ont accès à la page d’accueil et peuvent s’inscrire ou se connecter. Une fois connectés ils peuvent voir leurs profils. Depuis le profil ils peuvent modifier leurs informations ou leurs animaux, ils peuvent aussi ajouter des animaux. Ils peuvent modifier leurs disponibilités depuis leurs profils. S’ils ont des animaux inscrits ils peuvent faire une recherche pour les faire garder.

### L’API Google Maps

Cette API propose plusieurs fonctionnalités. Celles que je vais utiliser sont afficher une carte et placer des marqueurs dessus.



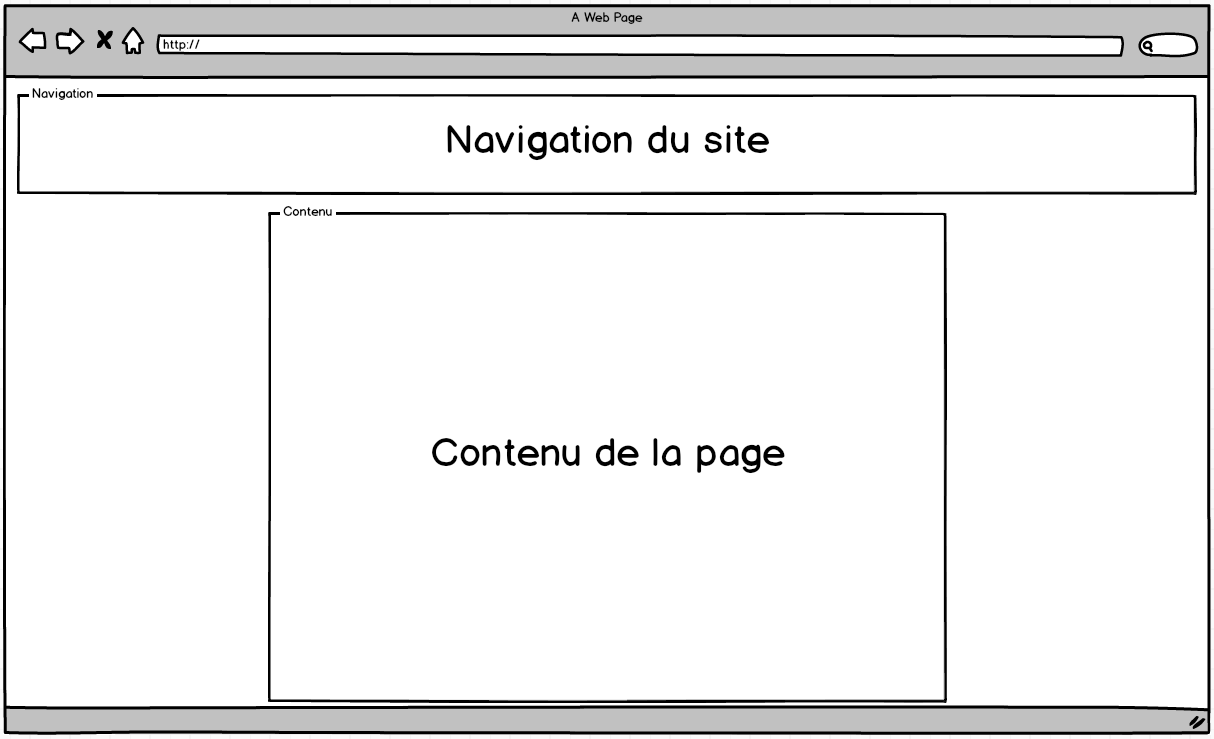
Cette API propose aussi une fonctionnalité de Géocoding que je vais développer dans le point suivant. Cette API est fournie par Google, est gratuite et n’a pas de limite de requêtes par jour. Pour l’utiliser il faut néanmoins l’activer dans le gestionnaire d’API de Google. En l’activant Google fournit une clé d’API qu’il faut ensuite placer dans l’URL de requête de l’API.

### Le Géocoding de Google Map

Cette fonctionnalité de l’API permet de transformer une adresse en coordonnées géographiques qui seront ensuite utilisée pour trouver des gardiens et les afficher sur une carte.

### Interface

#### Structure



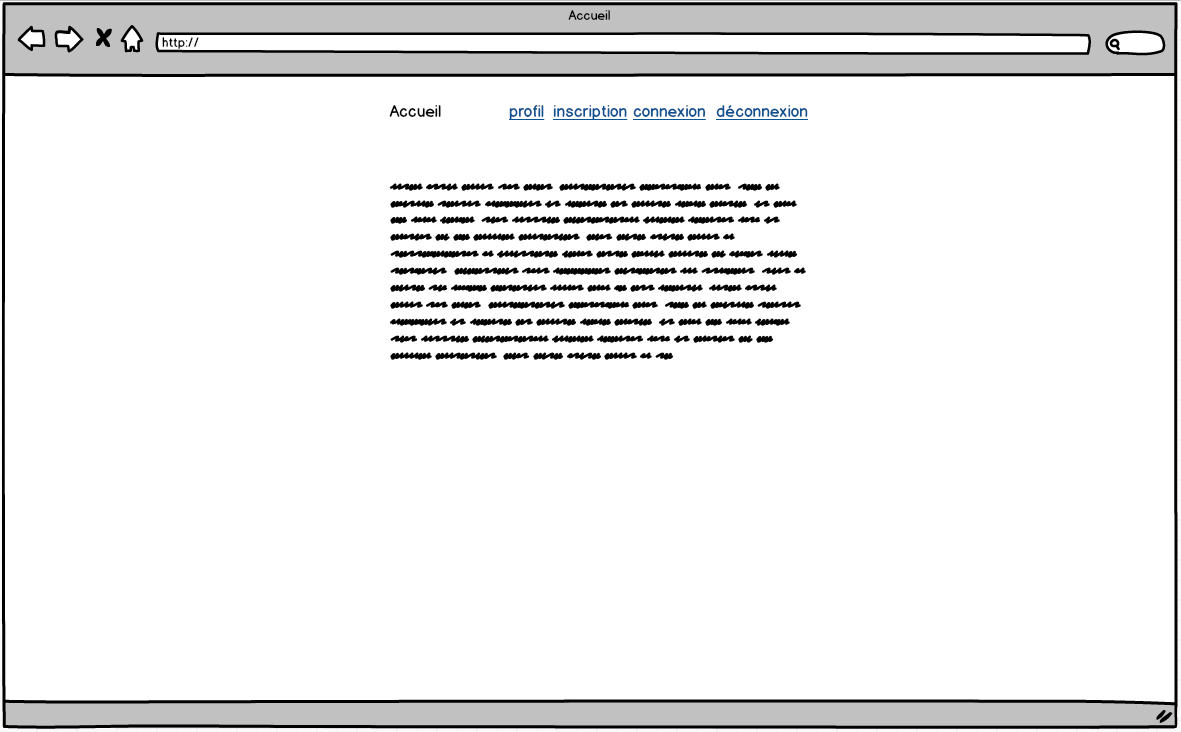
Toutes les pages du site sont divisées en deux parties :

* La barre de navigation
* Le contenu de la page

##### Design du site

J’ai utilisé Bootstrap[[4]](#endnote-4) pour faire le design de mon site. Boostrap permet de faire un site compatible pour tous types de support, de plus les thèmes représentent un gain de temps considérable en proposant des designs terminés qu’il suffit de télécharger. Le thème que j’ai choisi s’appelle Freelancer et je l’ai trouvé sur le site StartBootstrap[[5]](#endnote-5). C’est un thème gratuit et sur la page du thème[[6]](#endnote-6) il est possible de tester une démonstration et de le télécharger.

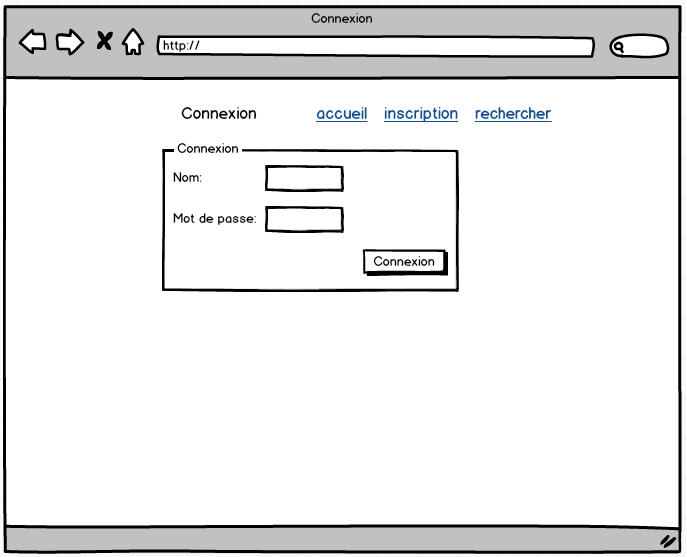
#### Page d’accueil



La page d’accueil est la première page du site sur laquelle l’internaute arrive. Sur cette page il y a un texte de présentations et d’explications du site.

Les liens qui sont affiché par défaut sont les liens pour l’inscription et pour la connexion. Si un utilisateur est déjà connecté les liens sont pour accéder au profil, rechercher un gardien ou pour se déconnecter.

#### Page de connexion

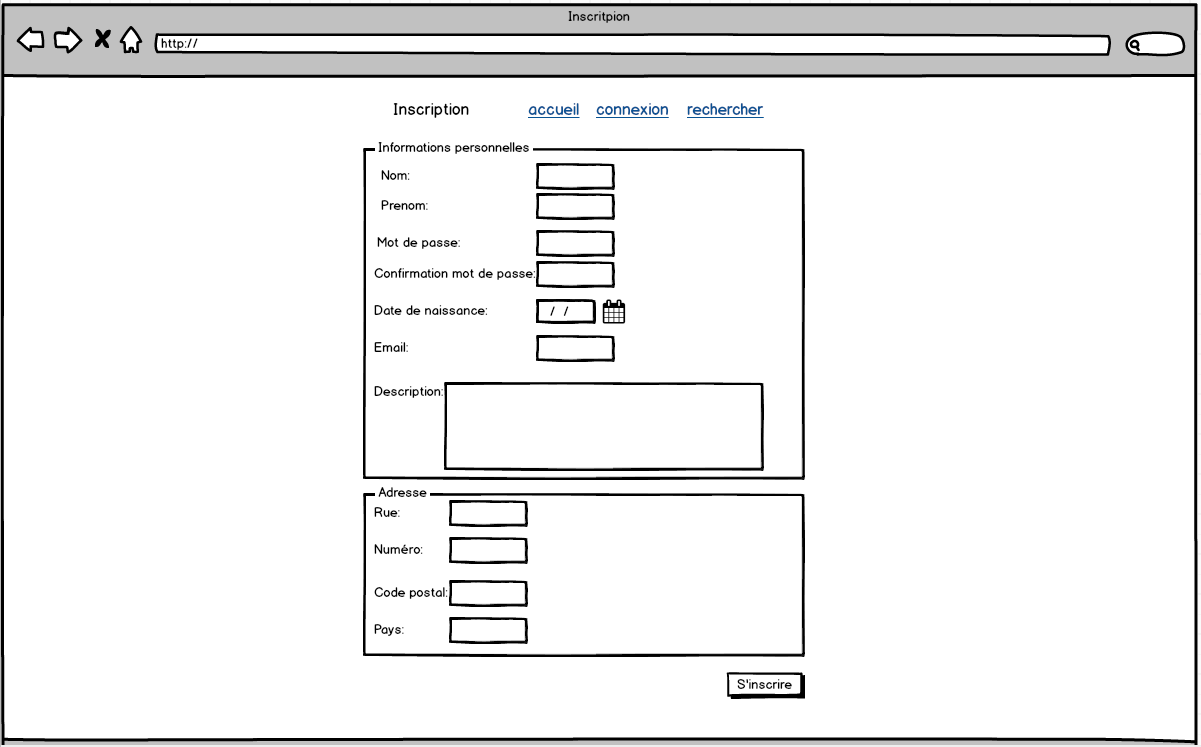


On arrive sur cette page lorsqu’on clique sur le lien de connexion sur la page d’accueil.

Pour se connecter il faut déjà avoir un compte et rentrer son nom et son mot de passe.

Depuis cette page il est possible d’aller sur la page d’inscription si on n’a pas déjà de compte ou de retourner sur la page d’accueil.

#### Page inscription



Cette page est celle qui permet de s’inscrire sur le site. On peut y accéder depuis la page d’accueil ou depuis la page de connexion.

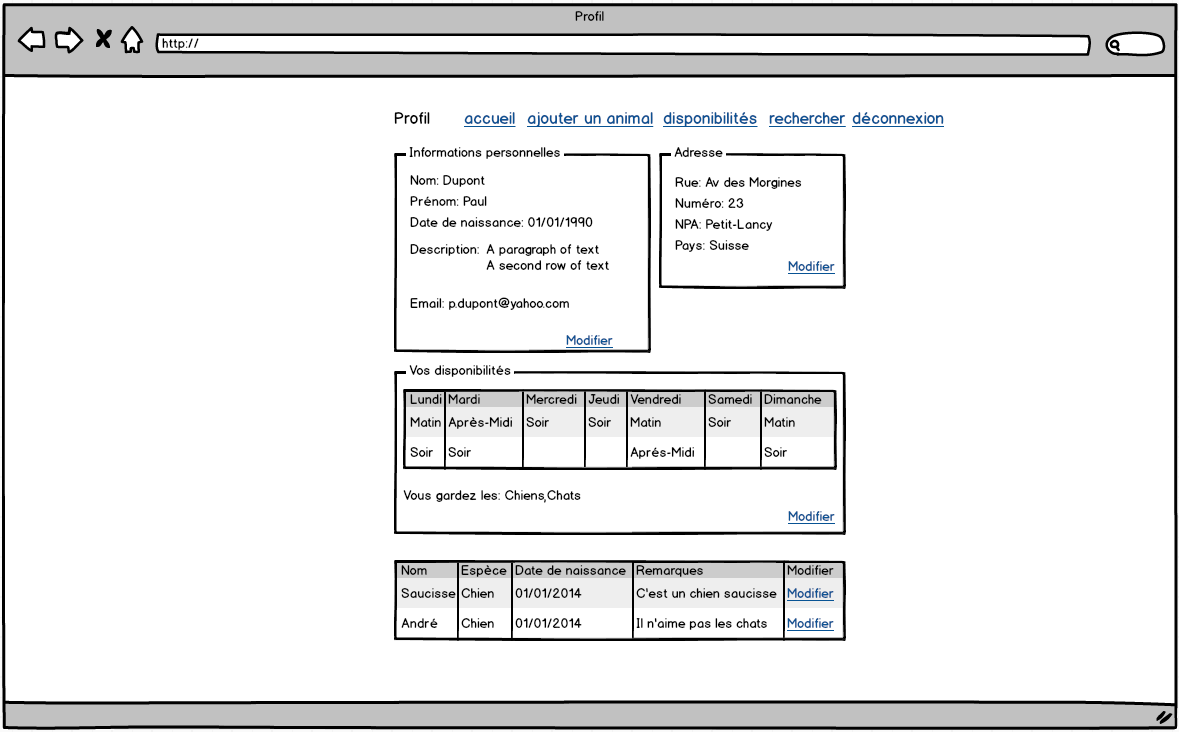
Elle est découpée en deux parties : la première est celle qui regroupe les informations d’identité. Il faut renseigner son nom, son prénom, son adresse email, son mot de passe qu’il faut confirmer, sa date de naissance et une description.

La deuxième est celle qui regroupe les informations de l’adresse. Il faut renseigner le pays, la rue, le numéro de rue et le code postal.

Quand le formulaire est validé les informations sont insérées dans la base de données, puis l’utilisateur est automatiquement connecté au site. Il est ensuite redirigé vers la page d’accueil.

Depuis cette page il est possible d’aller sur la page de connexion ou de retourner à l’accueil.

#### Page du profil



On peut accéder à cette page si on est connecté Cette page permet de voir et de modifier tout ce qui est en rapport avec l’utilisateur.

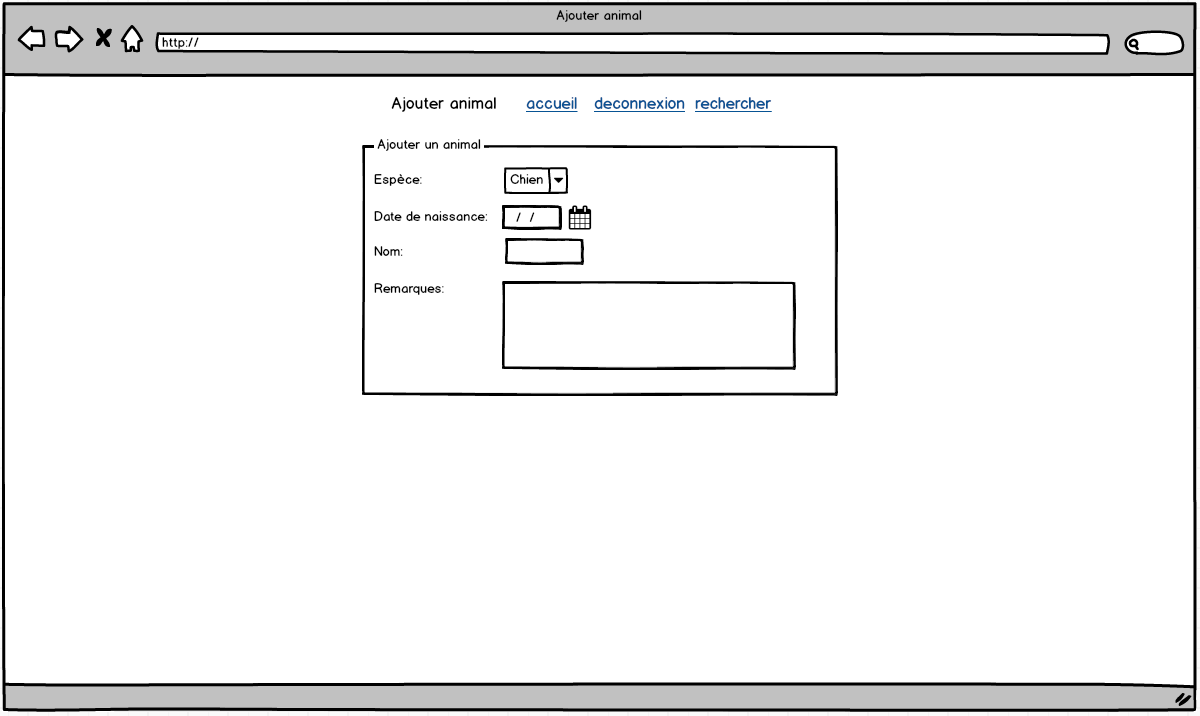
L’utilisateur peut modifier toutes ses informations personnelles en cliquant sur un des liens modifier de la section informations personnelles ou de la section adresse. S’il veut modifier son adresse il est redirigé vers une page qui reprend la deuxième partie de la page inscription. Les champs sont déjà remplis, il peut les modifier et enregistrer les modifications. Il est ensuite redirigé vers la page du profil. Si il veut modifier ses informations personnelles il est redirigé vers une page qui reprend la première partie de la page d’inscription mais les champs sont déjà remplis, il peut les modifier et ensuite enregistrer les modifications. Il est obligé de réinsérer sont mot de passe car il est stocké en sha1 dans la base de données et ne peut donc pas être récupérer.

Il peut modifier les données de ses animaux en cliquant sur le lien modifier dans la dernière colonne du tableau des animaux. Il est redirigé vers une page semblable à celle pour ajouter un animal mais les champs sont déjà remplis et il peut les modifier puis enregistrer les modifications.

Il peut modifier ses disponibilités en cliquant sur le lien modifier. Il est redirigé vers la page disponibilites avec les cases déjà cochées, il peut les modifier et les enregistrer.

Depuis cette page il est possible de retourner à l’accueil, d’ajouter un animal de compagnie, d’accéder à la page de disponibilités, de faire une recherche pour garder un animal ou de se déconnecter.

#### Page ajouter un animal



Cette page est accessible uniquement depuis le profil et donc uniquement lorsqu’on est connecté au site.

Elle permet de rajouter un animal de compagnie, en demandant l’espèce, la date de naissance le nom et d’éventuelles remarques sur l’animal.

Lorsque l’on valide le formulaire l’animal est ajouté dans la base de données puis on est redirigé vers la page du profil et nous pouvons voir tout de suite le nouvel animal sur la page.

Depuis cette page il est possible d’accéder à l’accueil, au profil, aux disponibilités ou de se déconnecter.

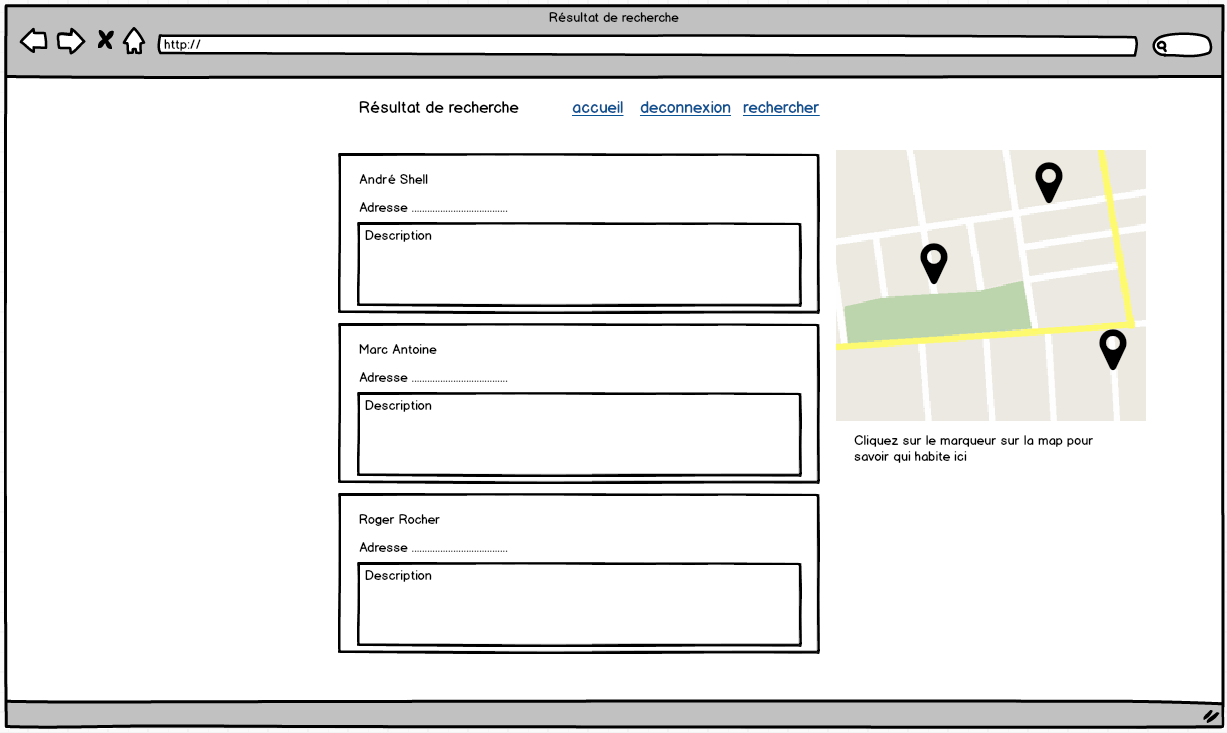
#### Page rechercher



Cette page permet de faire une recherche pour faire garder un animal. Comme l’animal est choisi dans une liste déroulante il faut être connecté et avoir ajouté un animal.

On choisit donc l’animal à faire garder, quand on veut le faire garder et la distance maximal entre notre domicile et celui du gardien. Une fois le formulaire validé on est redirigé vers la page résultats.

#### Page résultats

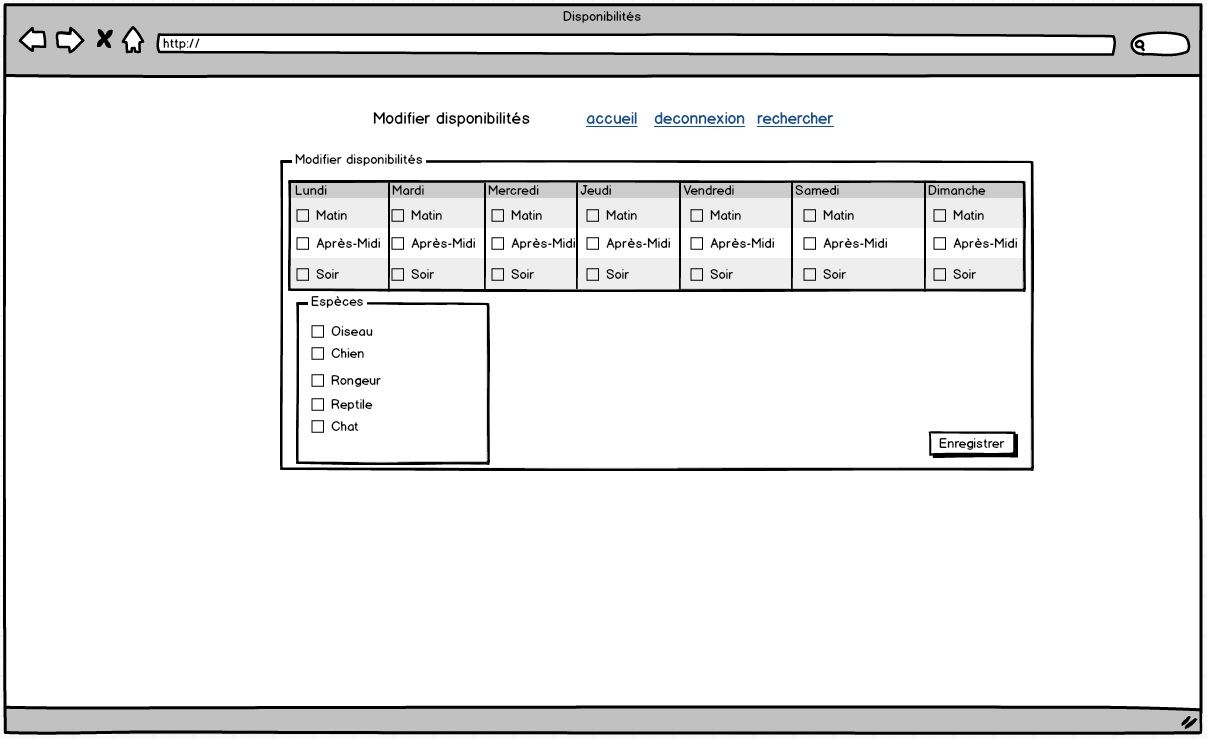


Cette page s’affiche lorsque l’on valide une recherche pour un animal.

Les gardiens sont affiché si il sont d’accord de garder l’espèce d’animal que l’on cherche à faire garder, si il peuvent le garder pour chaque moment de la semaine que l’on souhaite et si ils sont à moins de la distance maximum que l’on a précisé sur la page de recherche.

Elle comporte sur la partie gauche les informations des gardiens disponibles et sur la droite une carte de Google maps qui permet de voir ou habitent les gardiens disponibles. Il est possible de cliquer sur le marqueur sur la carte pour savoir qui habite à cet endroit.

#### Page disponibilités



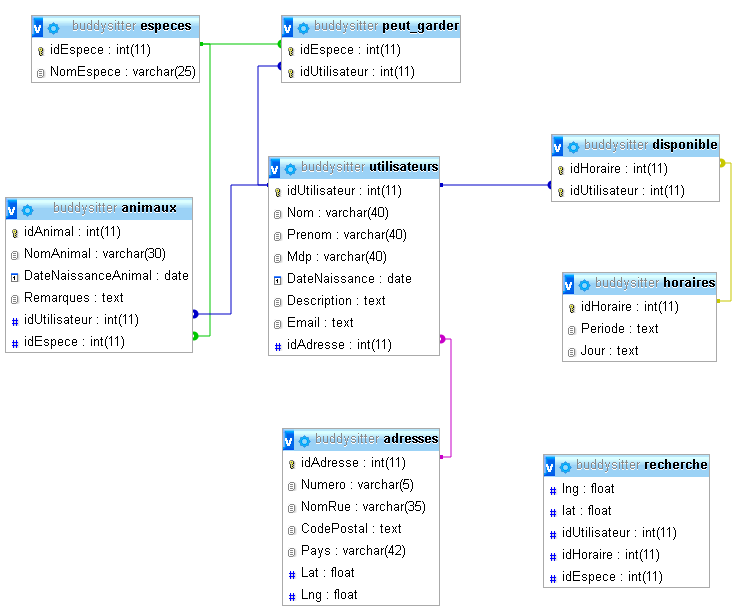
Cette page permet de modifier les disponibilités en tant que gardien. Par défaut lorsqu’un utilisateur s’inscrit il n’est jamais disponible et n’est donc pas gardien mais uniquement propriétaire. Il peut ensuite ajouter des disponibilités et donc devenir gardien via cette page qu’il peut atteindre depuis son profil.

Il doit renseigner les moments de la semaine ou il est disponible et les animaux qu’il est d’accord de garder.

Depuis cette page il est possible de retourner à l’accueil, d’aller sur son profil, d’aller sur la page pour ajouter un animal ou de se déconnecter.

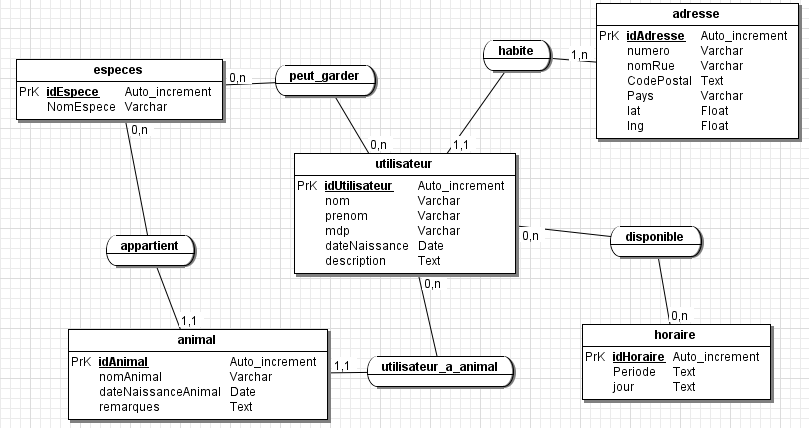
### La base de données

Toutes les données du site sont stockées dans une base de données MySQL dont voici le modèle logique de données.



#### Le modèle conceptuelle de données

Voici le modèle conceptuelle de la base de données.



La table utilisateurs est la table principale car toutes les autres tables y sont reliées.

La relation « habite » permet de relier la table utilisateur à la table adresse. Un utilisateur ne peut habiter qu’a un seul endroit (la cardinalité 1,1) mais une adresse peut être la même pour plusieurs utilisateurs (la cardinalité 1,n) si ils habitent au même endroit.

La relation « disponible » permet de relier la table utilisateur à la table horaire. Un utilisateur peut être disponible pour aucun ou plusieurs horaire (la cardinalité 0,n) et un horaire peut être choisi par personnes ou par plusieurs personnes (la cardinalité 0,n).

La relation « utilisateur\_a\_animal » permet de relier la table animal à la table utilisateur. Un utilisateur peut ne pas avoir d’animaux ou en avoir plusieurs (la cardinalité 0,n) et un animal ne peut appartenir qu’a un seul utilisateur (la cardinalité 1,1).

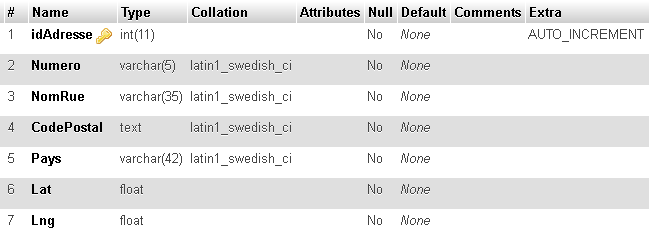
La relation « appartient » permet de relier la table animal à la table especes. Un animal ne peut ne peut appartenir qu’a une seul espèce (la cardinalité 1,1) et une espèce peut être utilisée aucune fois ou plusieurs fois (la cardinalité 0,n).

La relation « peut-garder » permet de relier la table espece à la table utilisateur. Un utilisateur peut garder aucune ou plusieurs espèces (la cardinalité 0,n) et une espèce peut être garder par personne ou par plusieurs utilisateurs (la cardinalité 0,n)

### Description des tables

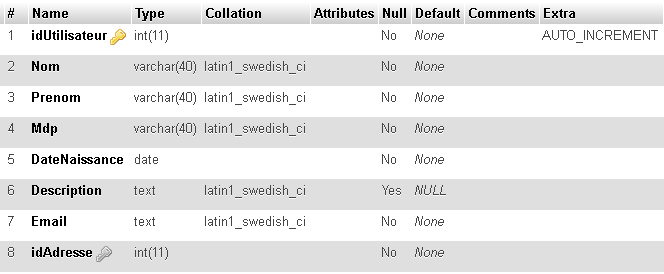
#### La table adresses

La table adresse contient les adresses des utilisateurs. L’adresse est reliée à l’utilisateur par son id. L’id de l’adresse est dans la table utilisateur. L’adresse est composée d’un numéro de rue, d’un nom de rue, du code postal et du pays. La latitude et la longitude sont récupérées grâce à l’api de google maps géocode.



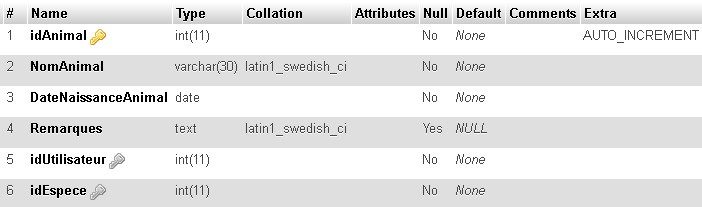
#### La table utilisateurs

La table utilisateurs stocke toutes les informations d’un utilisateur et fait le lien avec son adresse. Un utilisateur a un nom et prénom, une date de naissance, un Email et une description. Les mot de passe des utilisateurs sont stocké encodé en sha1.



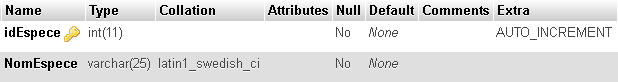
#### La table animaux

La table animaux stocke les informations des animaux. Les animaux sont relié a leur maître par l’idUtilisateur qu’ils contiennent. Leur espèce et déterminée par l’idEspece. Un animal a un nom, une date de naissance et d’éventuelles remarques.



#### La table especes

La table espèces contient les espèces qui peuvent être choisies dans le site. Elle est utilisée par l’utilisateur quand il choisit quelle espèce il peut garder et par l’animal pour déterminer son espèce. La table est composée de l’idEspece et du nom de l’espèce.



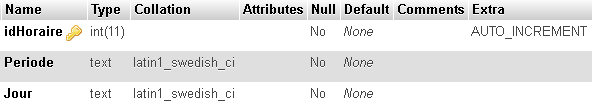
#### La table peut\_garder

La table peut\_garder permet de stocké les espèces que l’utilisateur est d’accord de garder. Elle est composée de l’id de l’espèce et de l’id de l’utilisateur.



#### La table horaires

La table horaires contient les horaires de la semaine, c’est-à-dire par exemple : Lundi Matin. Elle est composée de l’id de l’horaire de la période de la journée et de la journée.



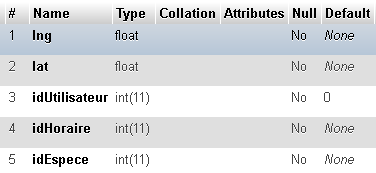
#### La table disponible

La table disponible permet de stocké les horaire ou un utilisateur est disponible. Elle est composée d l’id de l’utilisateur et de l’id de l’horaire.



#### La vue recherche

La vue recherche contient les informations nécessaires pour faire la recherche d’un gardien. Elle contient la latitude et la longitude de l’adresse, l’id de l’utilisateur, l’id de l’horaire pour le quelle un utilisateur est disponible et l’id de l’espèce qu’un gardien est d’accord de garder.



### Arborescence des fichiers du site



### Fonctions

#### Fonctions de la page phptohtml.php

Cette page me permet de mettre en forme des informations que j’ai en php dans des variables ou des tableaux.

##### afficherUtilisateur($infos)

Cette fonction permet d’afficher les informations d’un utilisateur dans des labels. Les informations ne peuvent donc pas être modifiées. La fonction affiche aussi un lien modifier qui redirige vers la page modifierinformations.php qui est la page pour modifier les informations d’un utilisateur. La variable $infos est le tableau qui contient toutes les informations d’un utilisateur récupérées dans la base de données. Elle contient donc son nom, prénom, date de naissance, adresse Email et sa description. Cette fonction est utilisée sur la page du profil.

##### afficherAdresse($infos)

Cette fonction est l’analogue de la fonction décrite ci-dessus à la seule différence qu’elle affiche les informations de l’adresse d’un utilisateur et que le lien modifier redirige vers la page modifieradresse.php. La variable $infos est un tableau associatif qui contient toutes les informations d’une adresse. Elle contient donc le code postal, la latitude, la longitude, le nom de la rue, le numéro ainsi que le pays. La latitude et la longitude ne sont pas affichées et ne peuvent pas être modifiées manuellement. Cette fonction est utilisée sur la page de profil.

##### afficherAnimaux($infos,$especes)

Cette fonction permet d’afficher tous les animaux d’un utilisateur sous forme de tableau. La variable $infos contient les infos des animaux d’un utilisateur et $especes contient les espèces car les animaux contiennent uniquement l’id de l’espèce donc pour avoir un affichage user friendly j’affiche les espèces. Cette fonction et utilisée sur la page de profil.

##### afficherEspeces($infos, $especes)

Cette fonction permet d’afficher des checkbox sur la page de disponibilités afin de pouvoir sélectionner les espèces que l’on est d’accord de garder. La variable $infos contient les disponibilités de l’utilisateur, c’est-à-dire les espèces qu’il est d’accord de garder. La variable $especes contient toutes les espèces et leur nom afin de pouvoir les afficher.

##### afficherAnimauxSelect($infos)

Cette fonction permet d’afficher une liste déroulante avec les animaux d’un utilisateur. Elle est utilisée sur la page de recherche. La variable $infos contient les animaux de l’utilisateur.

##### afficherDisponibilites($disponible, $horaires, $modif)

Cette fonction permet d’afficher un tableau dont chaque colonne est un jour de la semaine. Dans chaque colonne il y a les trois moments de la journée. Cette fonction est utilisée sur la page de profil, sur la page de disponibilités et sur la page de recherche. La variable $disponible est un tableau qui contient les disponibilités d’un utilisateur, on donne un tableau vide si l’on veut que aucune case soit cochée par défaut. La variable $horaires contient toutes les horaires de la semaine de la table horaires. La variable $modif est pour dire si les disponibilités peuvent être modifiée ou pas, par exemple sur la page de profil les disponibilités ne doivent pas pouvoir être modifiées.

##### afficherGardien($infos,$adresse)

Cette fonction permet d’afficher un gardien sur la page de résultat d’une recherche. Elle affiche le nom et prénom en titre puis l’adresse, l’adresse Email et la description du gardien. La variable $infos contient les informations personnelles du gardien. La variable $adresse contient l’adresse du gardien.

#### La page dao.php

Dao signifie Data Access Object, on comprend donc que cette page sert à avoir accès à la base de données. Cette page fait la connexion avec la base de données et regroupe les fonctions pour sélectionner, ajouter, modifier ou supprimer des données.

##### maConnexion()

Cette fonction permet de faire la connexion avec la base de données. Elle est utilisée dans chaque fonction de la page dao.php.

inscriptionUtilisateur($nom, $prenom, $mdp, $naissance, $description, $pays, $rue, $numero, $npa, $lat, $lng, $mail)

Cette fonction permet d’inscrire un utilisateur et son adresse dans la base de données. Cette fonction commence par appeler la fonction insererAdresse que je décrirais juste après, puis insert dans la table utilisateur les valeurs de $nom, $prenom, $mdp, $naissance, $description et de $mail. Cette fonction retourne l’id de l’insertion. Cette fonction est utilisée sur la page d’inscrption.

##### insererAdresse($pays, $rue, $numero, $npa, $lat, $lng)

Cette fonction permet d’insérer une adresse dans la base de données. La requête sql est un insert dans la table adresses. Cette fonction retourne l’id de l’insertion qui pourra être utilisé dans la fonction inscriptionUtilisateur(). Cette fonction est utilisée sur la page d’inscription.

##### connexion($mdp, $nom)

Cette fonction sélectionne l’id de l’utilisateur son nom et son mot de passe si son nom est égal à $nom et si son mot de passe est égale à $mdp et retourne un tableau. Pour savoir si un utilisateur peut se connecter il suffit ensuite de vérifier que le tableau ne soit pas vide. Cette fonction est utilisée sur la page de connexion.

##### recupererEspeces()

Cette fonction permet de récupérer toutes les informations de la table espèces et les retourne sous forme de tableau. Cette fonction est utilisée sur la page des disponibilités, la page de profil et sur la page pour ajouter un animal.

##### inscrptionAnimal($espece, $nom, $naissance, $remarques, $idUtilisateur)

Cette fonction permet d’insérer un animal dans la base de données et retourne l’id de l’insertion. Cette fonction est utilisée sur la page d’inscription d’un animal.

##### recupererUtilisateur($idUtilisateur)

Cette fonction permet de récupérer toutes les informations d’un utilisateur dont l’id est égal à $idUtilisateur et retourne le résultat sous forme d’un tableau. Cette fonction est utilisée sur la page pour modifier ses informations personnelles.

##### recupererAnimauxDepuisUtilisateur($idUtilisateur)

Cette fonction permet de récupérer toutes les informations de tous les animaux qui appartiennent à un utilisateur dont l’id est égal à $idUtilisateur. Les informations des animaux sont retournées dans un tableau. Cette fonciton est utilisée sur la page de profil et sur la page pour rechercher un gardien.

##### recupererAnimal($idAnimal)

Cette fonction permet de récupérer toutes les informations d’un animal dont l’id est égal à $idAnimal. Ces informations sont retournées dans un tableau. Cette fonction est utilisée sur la page résultat et sur la page pour modifier un animal.

##### recupererAdresse($idAdresse)

Cette fonction permet de récupère l’adresse dont l’id est égal à $idAdresse. L’adresse est retournée sous forme de tableau. Cette fonction est utilisée sur la page de profil, sur la pour modifier son adresse et sur la page de résultat d’une recherche.

##### recupererAdresseDeUtilisateur($idUtilisateur)

Cette fonction permet de récupérer l’adresse de l’utilisateur dont l’id est égal à $idUtilisateur. L’adresse est retournée dans un tableau. Cette fonction est utilisée sur la page de résultat d’une recherche.

##### recupererHoraires()

Cette fonction permet de récupérer toutes les informations de la table horaires. Les informations sont retournées sous forme de tableau.

##### insererDisponibilites($idHoraire, $idUtilisateur)

Cette fonction permet d’enregistrer les disponibilités d’un utilisateur dans la base de données. La variable $idHoraire est l’id de l’horaire. Cette fonction retourne l’id de l’insertion.

##### deleteDisponibilites($idUtilisateur)

Cette fonction permet de supprimer toutes les disponibilités d’un utilisateur.

##### insererChoixRace($idEspece, $idUtilisateur)

Cette fonction permet d’insérer les espèces que le gardien est d’accord de garder. Cette fonction retourne l’id de l’insertion.

##### supprimerChoixRaces($idUtilisateur)

Cette fonction permet de supprimer toutes les espèces que le gardien est d’accord de garder.

##### recupererDisponibilités($idUtilisateur)

Cette fonction permet de récupérer les moments de la semaine que le gardien est disponible. Les moments sont retournés dans un tableau.

##### modifierAdresse($numero, $rue, $npa, $pays, $lat, $lng, $idAdresse)

Cette fonction permet de modifier l’adresse dont l’id est égal à $idAdresse. La latitude et la longitude de la nouvelle adresse sont récupérer grâce à l’API de google maps au moment de la validation du formulaire sur la page modifieradresse.

##### modifierUtilisateur($nom, $prenom, $mdp, $naissance, $description, $idUtilisateur, $email)

Cette fonction permet de modifier un utilisateur dont l’id est égal à $idUtilisateur.

##### modifierAnimal($nom, $naissance, $remarque, $espece, $idAnimal)

Cette fonction permet de modifier un animal dont l’id est égal à $idAnimal

##### recupererEspeceGardable($idUtilisateur)

Cette fonction permet de récupérer l’id des espèces que le gardien qui a comme id $idUtilisateur est d’accord de garder. Les espèces sont retournées dans un tableau.

##### rechercherGardien($dispos, $idEspece, $lat, $lng, $distance)

Cette fonction est celle qui permet de rechercher un gardien qui correspond aux critères de recherche. La requête est effectuée sur la vue recherche. La variable $dispos contient les moments de la semaine pour lesquelles on recherche un gardien. La variable $idEspece contient l’id de l’espèce de notre animal que l’on cherche à faire garder. Les variables $lat et $lng contiennent respectivement la latitude et la longitude de l’adresse du propriétaire de l’animal. La variable $distance est la distance maximale entre le domicile du propriétaire et celle du gardien.

Le tableau $dispos est transformé en chaine de caractères grâce à la méthode implode afin que les disponibilités puissent être insérée dans la requête.

Voici la requête :

SELECT \* FROM recherche WHERE idHoraire IN ( $ids ) AND idEspece = $idEspece AND (DEGREES(Acos(sin(radians($lat))\*sin(radians(lat))+cos(radians($lat))\*cos(radians(lat))\*cos(radians($lng-lng)))))\*110.544 < $distance GROUP BY idUtilisateur ;

La première clause du where (idHoraire IN ( $ids )) permet de tester que le gardien est disponible pour chaque moments de la semaine.

La deuxième clause (idEspece = $idEspece) permet de tester si le gardien est d’accord de garder ce type d’animal.

La troisième clause ((DEGREES(Acos(sin(radians($lat))\*sin(radians(lat))+cos(radians($lat))\*cos(radians(lat))\*cos(radians($lng-lng)))))\*110.544 < $distance) permet de tester si la distance entre les deux domicile est plus petite que $distance. Le calcul qui permet de calculer la distance entre deux coordonnées a été récupérée dans le livre « PHP Advanced and Object-Oriented Programming » de l’auteur Larry Ullman dans le chapitre « Advanced Database Cocepts ».

Les gardiens disponibles sont retournés dans un tableau.

### Le JavaScript

Il y a très peu de JavaScript dans le site. Il est utilisé pour afficher la carte ainsi que les marqueurs sur la page de résultats d’une recherche. Il y a deux fonctions sur cette page en JavaScript.

#### myMap(lat, lng)

Cette fonction permet de créer et d’afficher la carte sur la page. On commence par définir les propriétés de la carte : le centre de la carte et le niveau de zoom initial. Les variables lat et lng sont les coordonnées ou se trouvera le centre de la carte, et le zoom est initialisé au niveau 8. Ensuite la carte est créée et placée dans le div qui a comme id « googleMap ».

#### addMarker(lat, lng, nom)

Cette fonction permet de placer un nouveau marqueur sur la carte. On commence par créer une variable de type LatLng qui est une variable qui contient la latitude et la longitude. Elle prend la valeur des variables lat et lng. On crée ensuite un nouveau marqueur qui aura comme position la variable que l’on vient de créer de type LatLng. Le marqueur est ensuite ajouté sur la carte et on ajoute un événement lorsque l’on clique sur le marqueur. Cette événement ce produit lors d’un clique et affiche une fenêtre d’information qui contient le nom du gardien récupéré grâce à la variable nom.

Plan de tests

|  |  |  |  |
| --- | --- | --- | --- |
| N° Test | Auteur | Scénario | Résultat attendu |
| 01 | HN | Aller sur la page index.php | Avoir un affichage décent avec les trois rubriques de présentation |
| 02 | HN | Sans être connecte sur index.php | Avoir les liens de connexion et d’inscription |
| 03 | HN | Sur la page index.php cliquez sur le lien de connexion | Être redirigé vers la page de connexion |
| 04 | HN | Sur la page index.php cliquer sur le lien d’inscription | Être redirigé vers la page d’inscription |
| 05 | HN | Sur la page de connexion insérer un nom et un mot de passe valide | Être connecté et rediriger vers index.php |
| 06 | HN | Sur la page de connexion insérer un mot de passe et un nom invalide | Un message d’erreur s’affiche |
| 07 | HN | Sur la page d’index, être connecté | Avoir les liens de profil, rechercher et déconnexion affichés |
| 08 | HN | Cliquer sur le lien profil depuis index.php | Être redirigé vers la page du profil |
| 09 | HN | Sur la page de profil, connecté | Avoir un affichage décent avec les informations personnelles, les animaux et les disponibilités. Avoir les liens d’accueil, ajouter un animal, disponibilités, rechercher et déconnexion |
| 10 | HN | Sans être connecté accéder à /profil.php | Être redirigé vers la page d’accueil |
| 11 | HN | Sans avoir d’animaux essayer de rechercher un gardien |  |
| 12 | HN |  |  |
| 13 | HN |  |  |
| 14 | HN |  |  |
| 15 | HN |  |  |
| 16 | HN |  |  |
| 17 | HN |  |  |
| 18 | HN |  |  |
| 19 | HN |  |  |
| 20 | HN |  |  |
| 21 | HN |  |  |
| 22 | HN |  |  |
| 23 | HN |  |  |
| 24 | HN |  |  |
| 25 | HN |  |  |
| 26 | HN |  |  |
| 27 | HN |  |  |

1. <https://fr.wikipedia.org/wiki/Animal_de_compagnie> [↑](#endnote-ref-1)
2. <https://www.google.ch/search?safe=strict&q=nombre+d%27animaux+de+compagnie+en+belgique&oq=nombre+d%27animaux+de+compagnie+en+belgique&gs_l=serp.3..0.189520.190820.0.190993.8.8.0.0.0.0.106.476.7j1.8.0....0...1.1.64.serp..0.8.476.bqdPc323dtQ> [↑](#endnote-ref-2)
3. <https://www.bonus.ch/ND/Animaux-a-poils-a-plumes-ou-a-ecailles-en-suisse-ID201107012SEV.aspx> [↑](#endnote-ref-3)
4. <http://getbootstrap.com/> [↑](#endnote-ref-4)
5. <https://startbootstrap.com/> [↑](#endnote-ref-5)
6. <https://startbootstrap.com/template-overviews/freelancer/> [↑](#endnote-ref-6)