|  |
| --- |
| Documentation technique de BuddySitter |
| TPI 2017 |
| Hervé Neuenschwander  I.FA-P3B |
|  |



Contenu

[Contenu 1](#_Toc485570601)

[Introduction 3](#_Toc485570602)

[Préambule 3](#_Toc485570603)

[Description 3](#_Toc485570604)

[Organisation 3](#_Toc485570605)

[Etude d’opportunité 3](#_Toc485570606)

[Pourquoi ce sujet 3](#_Toc485570607)

[Pourquoi le PHP 3](#_Toc485570608)

[Description de l’existant 3](#_Toc485570609)

[Ce que mon projet à de plus 6](#_Toc485570610)

[Analyse fonctionnelle 6](#_Toc485570611)

[Fonctionnalités des utilisateurs 6](#_Toc485570612)

[L’API Google Map 7](#_Toc485570613)

[Le Géocoding de Google Map 7](#_Toc485570614)

[Interface 8](#_Toc485570615)

[Structure 8](#_Toc485570616)

[Page d’accueil 9](#_Toc485570617)

[Page de connexion 10](#_Toc485570618)

[Page inscription 11](#_Toc485570619)

[Page du profil 12](#_Toc485570620)

[Page ajouter un animal 13](#_Toc485570621)

[Page rechercher 14](#_Toc485570622)

[Page résultats 15](#_Toc485570623)

[Page disponibilités 16](#_Toc485570624)

[La base de données 17](#_Toc485570625)

[Le modèle conceptuel de données 17](#_Toc485570626)

[Description des tables 18](#_Toc485570627)

[La table adresses 18](#_Toc485570628)

[La table utilisateurs 19](#_Toc485570629)

[La table animaux 19](#_Toc485570630)

[La table especes 20](#_Toc485570631)

[La table peut\_garder 20](#_Toc485570632)

[La table horaires 20](#_Toc485570633)

[La table disponible 20](#_Toc485570634)

[La vue recherche 20](#_Toc485570635)

[Arborescence des fichiers du site 21](#_Toc485570636)

[Fonctions 21](#_Toc485570637)

[Fonctions de la page phptohtml.php 21](#_Toc485570638)

[La page dao.php 23](#_Toc485570639)

[La page ajoutanimal.php 27](#_Toc485570640)

[La page connexion.php 27](#_Toc485570641)

[La page deconnexion.php 28](#_Toc485570642)

[La page disponibilites.php 28](#_Toc485570643)

[La page index.php 29](#_Toc485570644)

[La page inscription.php 29](#_Toc485570645)

[La page modifieradresse.php 29](#_Toc485570646)

[La page modifieranimal.php 29](#_Toc485570647)

[La page modifierinformations.php 29](#_Toc485570648)

[La page profil.php 29](#_Toc485570649)

[La page rechercher.php 30](#_Toc485570650)

[La page resultats.php 30](#_Toc485570651)

[Plan de tests 31](#_Toc485570652)

[Rapport de test 33](#_Toc485570653)

[Conclusion 34](#_Toc485570654)

[Respect du cahier des charges 34](#_Toc485570655)

[Différence de la base de données 34](#_Toc485570656)

[Difficultés particulières 34](#_Toc485570657)

[Ce que ce projet m’a apporté 34](#_Toc485570658)

[Suites possible du projet 34](#_Toc485570659)

## Introduction

### Préambule

Le premier animal de compagnie a été le chien, issu du loup. Ensuite des 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 ont été utilisés pour combattre l’invasion de rat. Lorsque cette invasion fut terminée les animaux resteront dans les foyers pour devenir des animaux de compagnies. En 2010 environ 49% des foyers français avaient au moins un animal de compagnie pour un total d’environ 59 millions d’animaux de compagnie. En Belgique c’est 50%[[1]](#endnote-1) de la population qui a un animal de compagnie, représentant 1,3 millions[[2]](#endnote-2) de mascottes et en suisse c’est 30% de la population pour un peu moins de 2 millions d’animaux de compagnie[[3]](#endnote-3). Cela fait autant d’animaux qui devront potentiellement se faire garder un ou plusieurs jours si leurs propriétaires ont des obligations qui leur 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 compagnie. C’est un problème auquel une majorité des personnes propriétaires d’animaux sont confrontées 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 précautions 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 site car il est utile et il 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 personne 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 est plus simple pour les utilisateurs d’accéder à un site web plutôt que de télécharger une application. 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 faire garder ou pour garder des animaux. Cependant ils ne permettent pas tous d’enregistrer ses animaux.

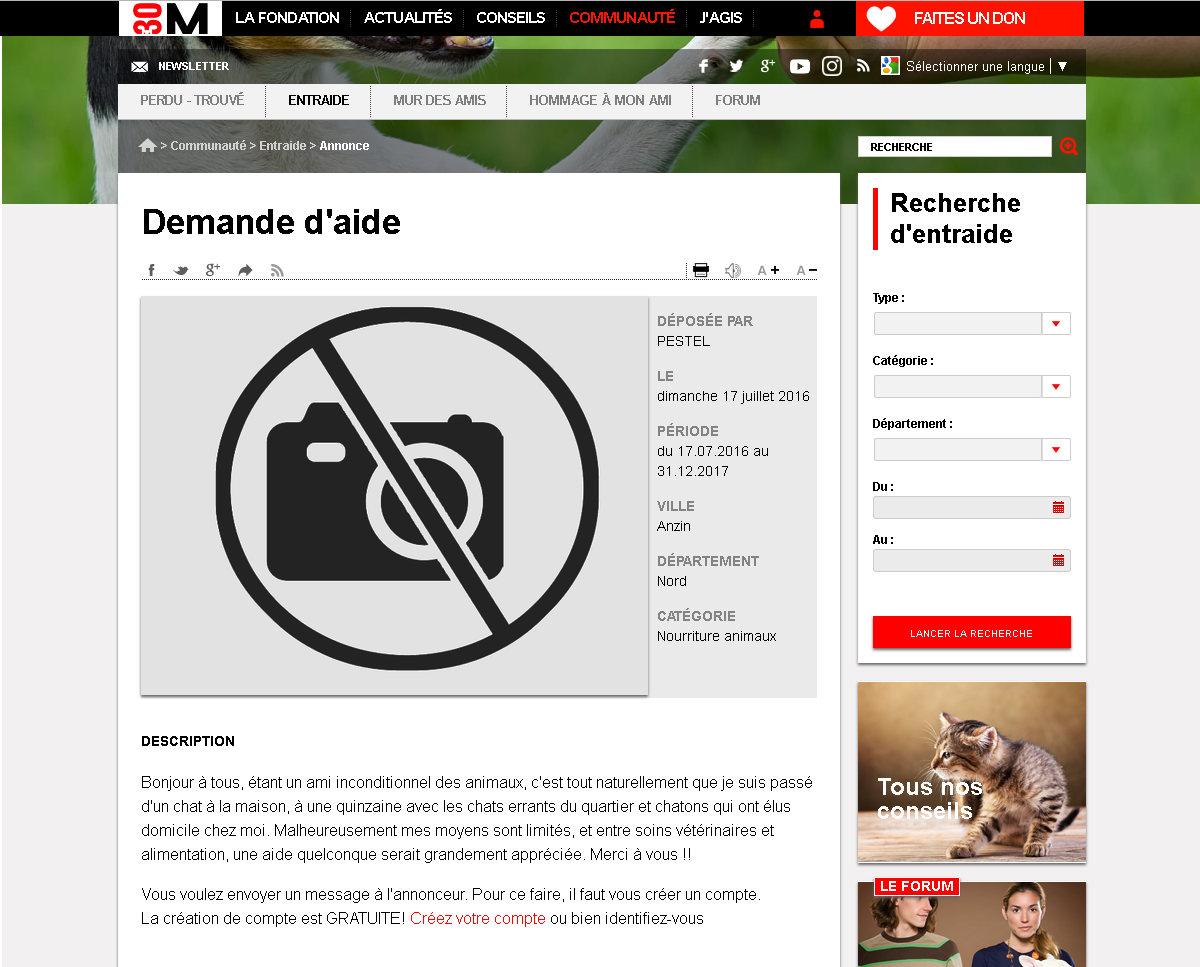
Par exemple le site <http://www.30millionsdamis.fr> propose dans son onglet communauté une section entraide. Cette section propose une carte de la France. En cliquant sur un département nous voyons les demandes d’aide et les propositions d’aide.



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, tel que de la nourriture pour animaux ou du transport d’animaux.

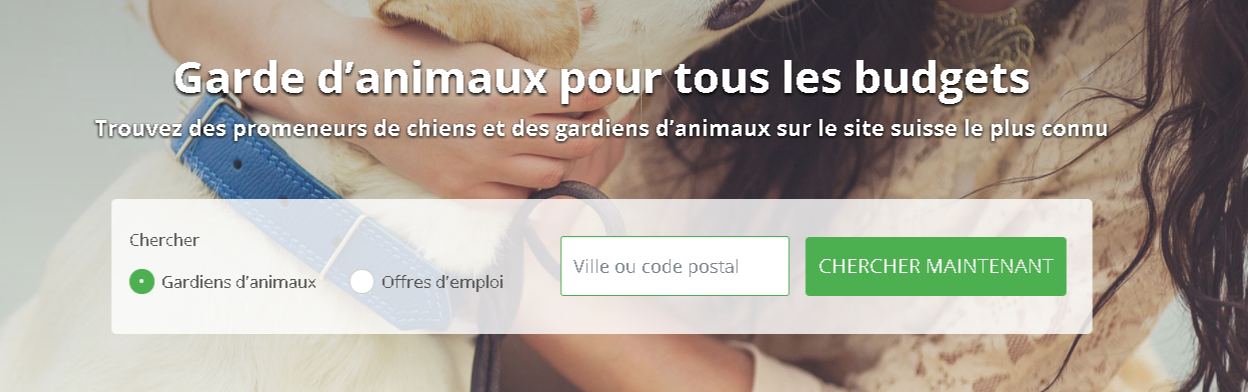


Il est possible de cliquer sur une annonce pour afficher tous les détails.

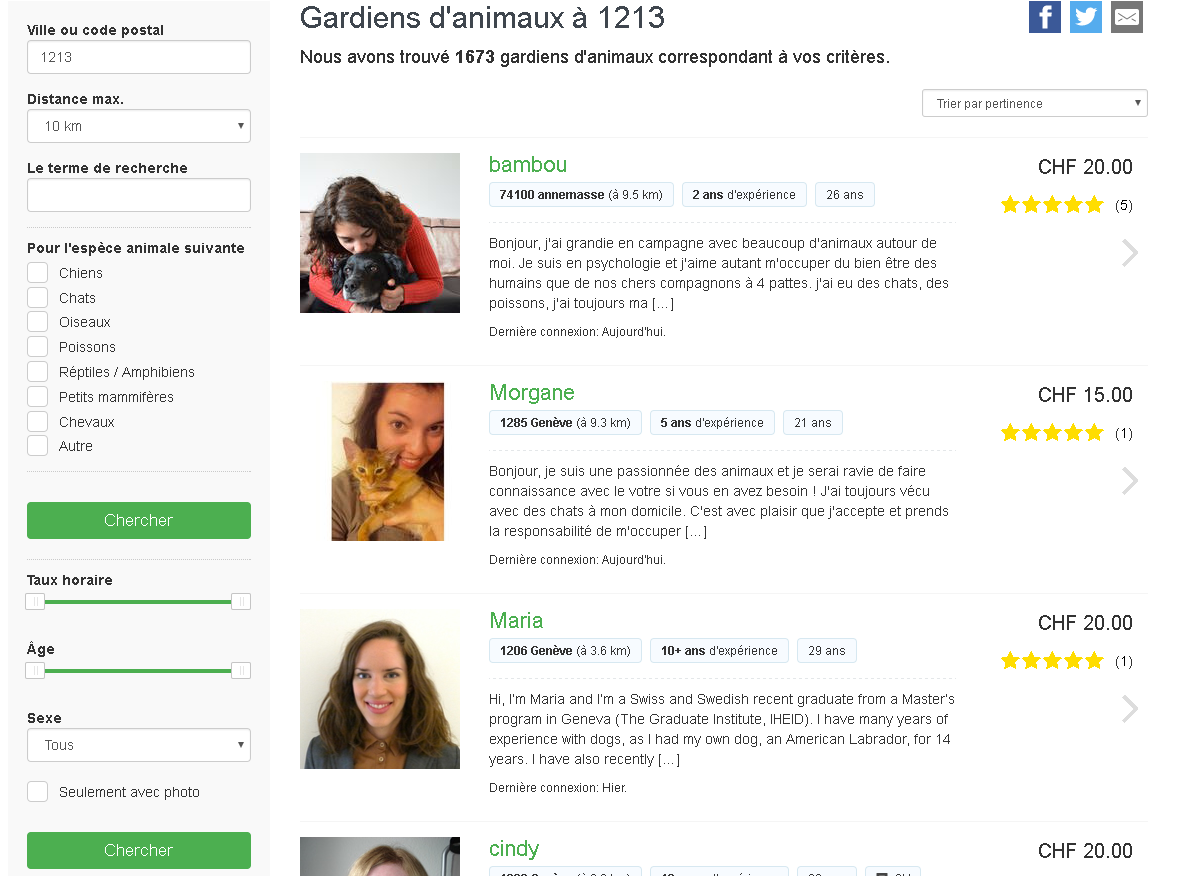


Sur la page de détail d’une annonce, il est possible de voir une photo si le créateur de l’annonce en a mis une. On peut aussi voir son nom, la date à laquelle il a créé l’annonce, la ville ainsi que le département et une catégorie.

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.



Voici la page de résultat de ce site.



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’âge 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 deux choses de plus : la géolocalisation et l’enregistrement des animaux. La géolocalisation permet de proposer des gardiens qui sont proches du propriétaire. L’enregistrement permet une recherche plus facile.

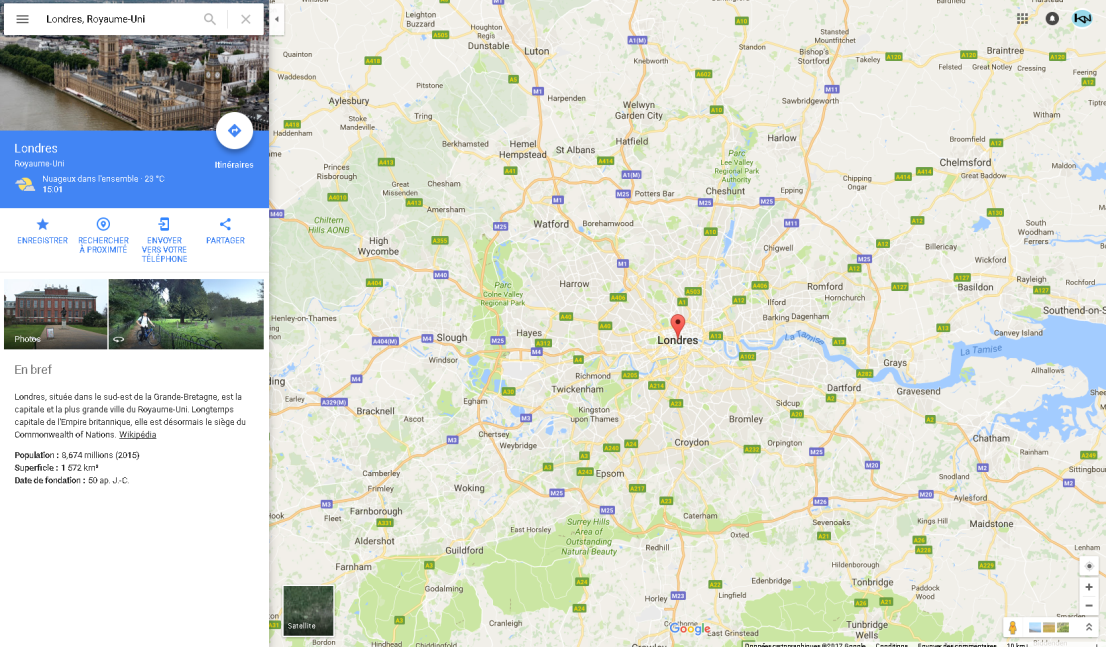
## Analyse fonctionnelle

### 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 leur profil. 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 Map

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



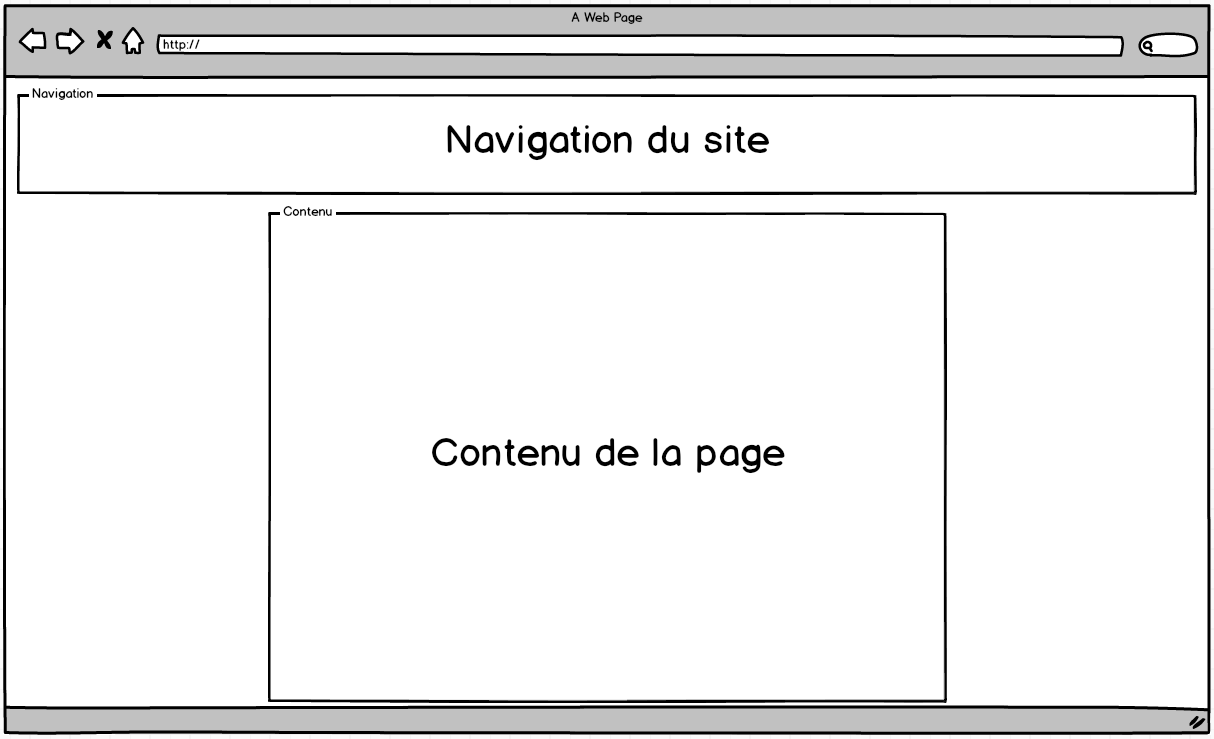
Cette API propose aussi une fonctionnalité de Géocoding que je vais développer dans le point suivant. Cette API est fournie par Google, est gratuite et n’a pas de limite de requêtes par jour. Pour l’utiliser il faut néanmoins l’activer dans le gestionnaire d’API de Google. En l’activant Google fournit 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 utilisées pour trouver des gardiens et les afficher sur une carte.

### Interface

#### Structure



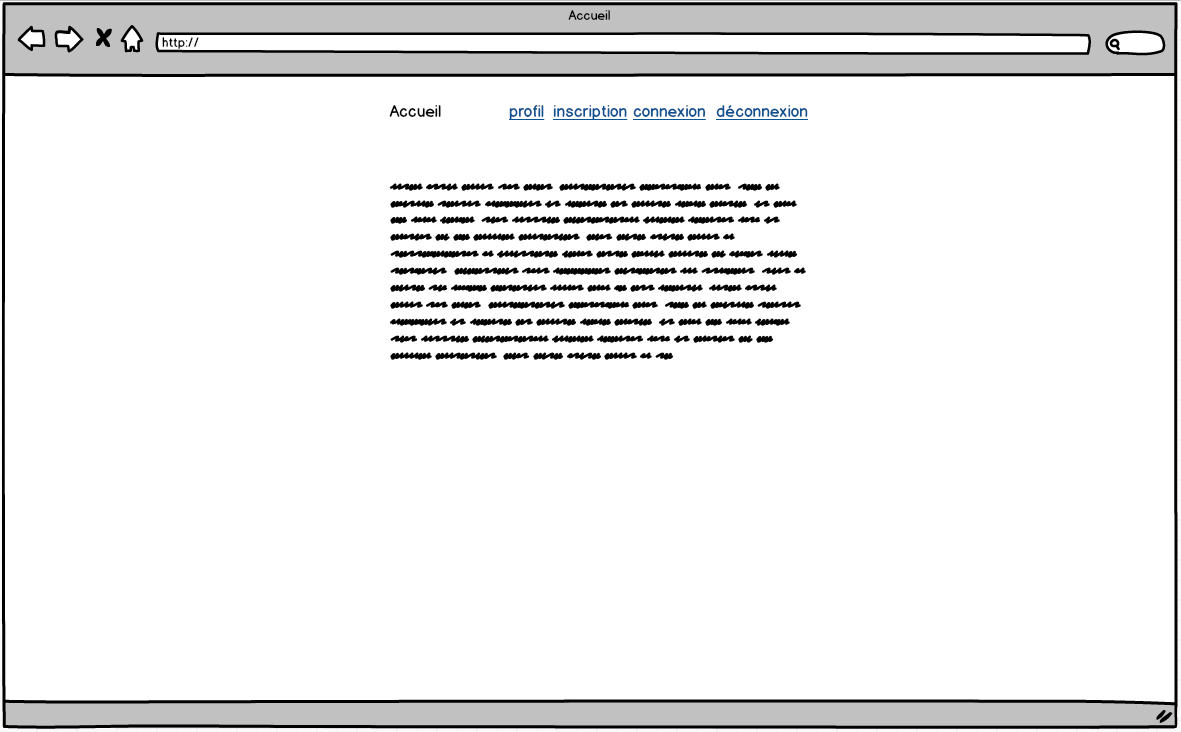
Toutes les pages du site sont divisées en deux parties :

* 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. Bootstrap 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.

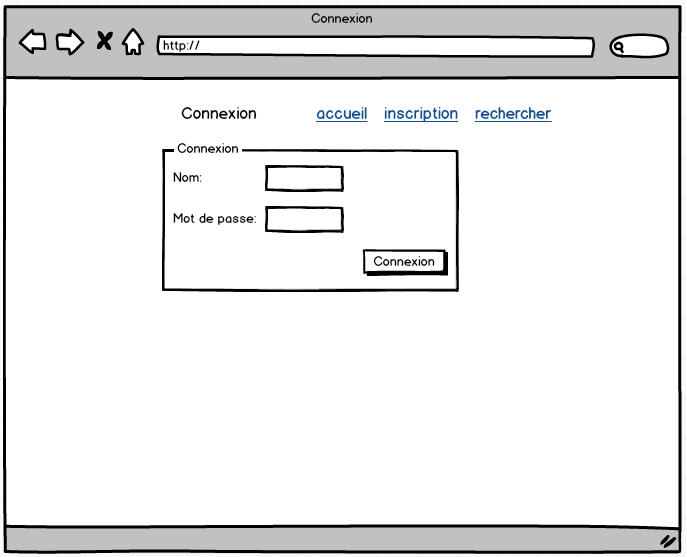
#### Page d’accueil



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

Les liens qui sont affichés 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

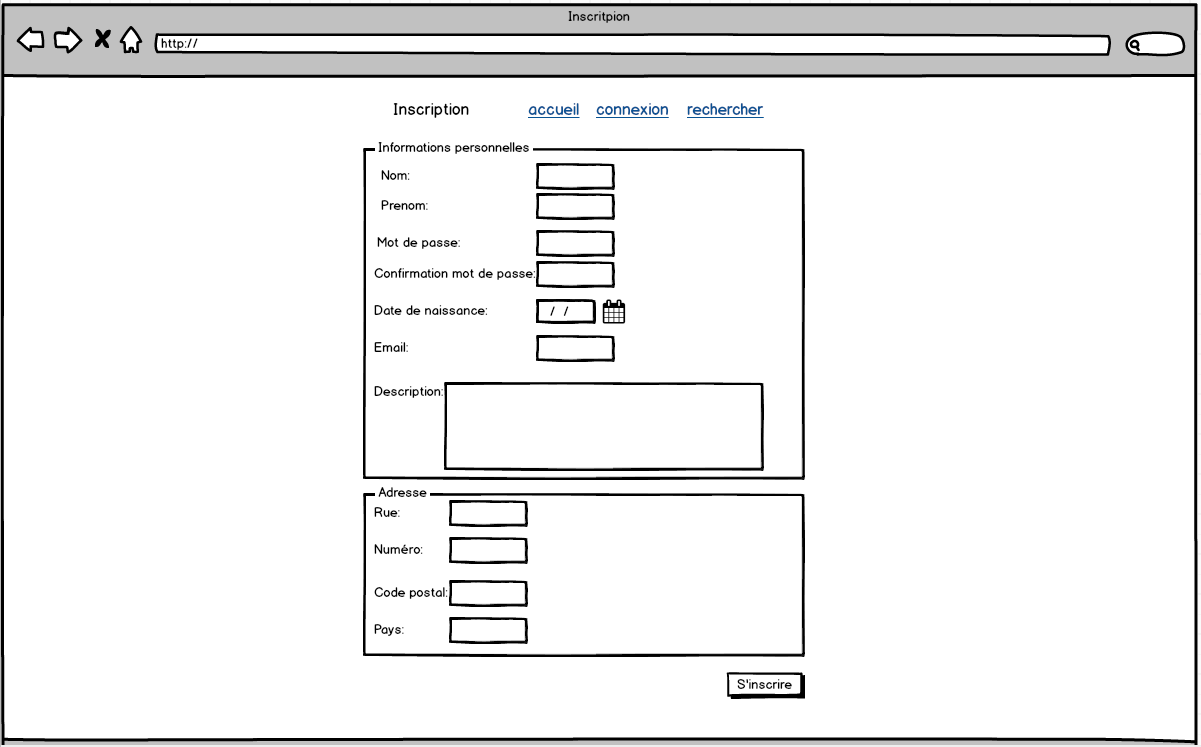


On arrive sur cette page lorsque l’on clique sur le lien de connexion sur la page d’accueil.

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.

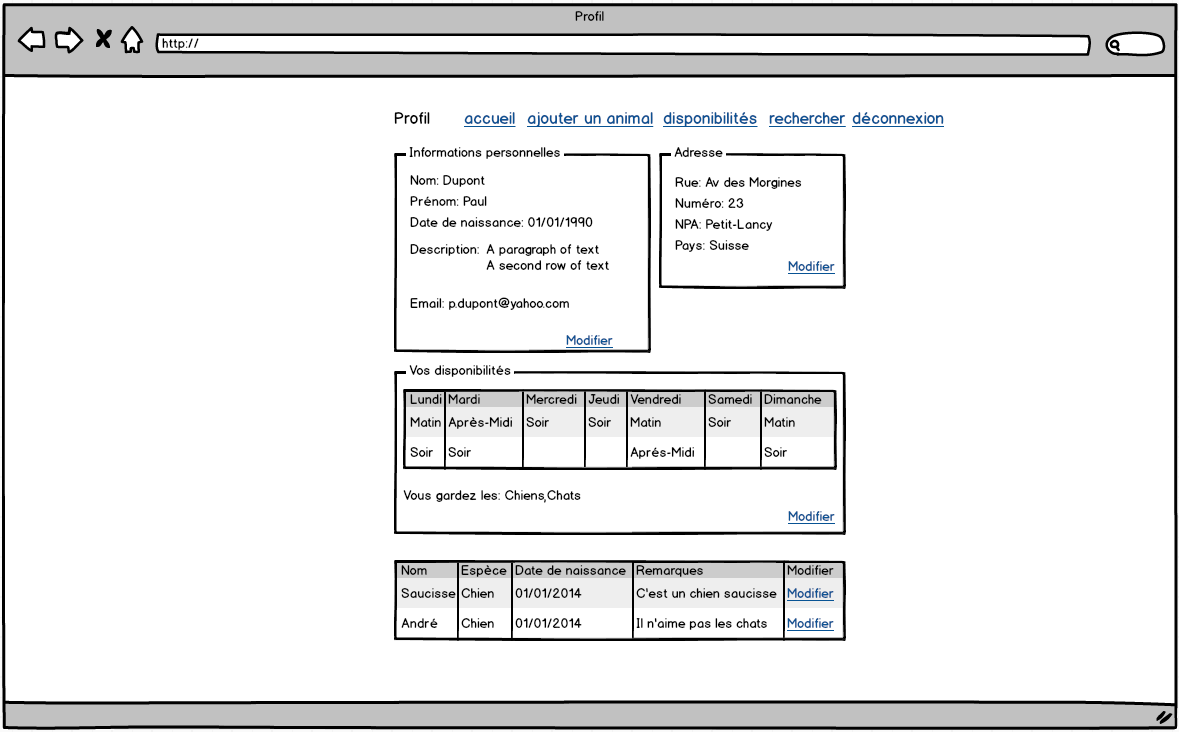
Elle est découpée en deux parties : la première est celle qui regroupe les informations d’identité. Il faut renseigner son nom, son prénom, son adresse email, son mot de passe qu’il faut confirmer, sa date de naissance et une description.

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.

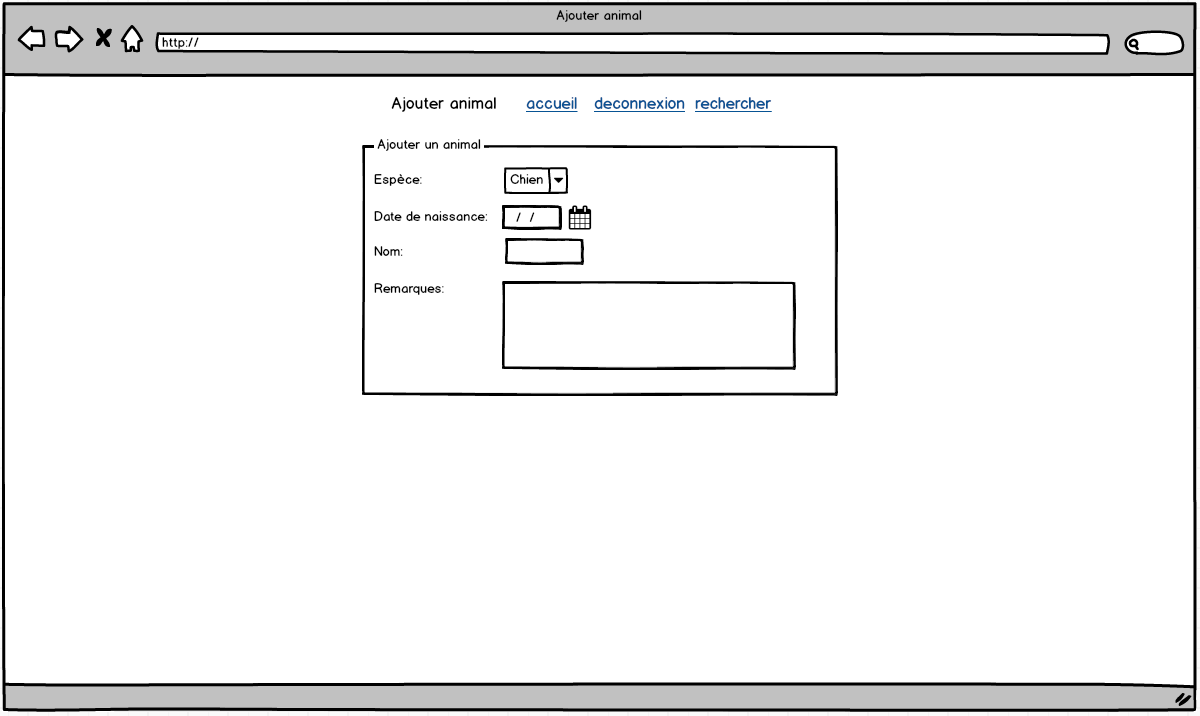
L’utilisateur peut modifier toutes ses informations personnelles en cliquant sur un des liens « modifier » de la section informations personnelles ou de la section adresse. S’il veut modifier son adresse il est redirigé vers une page qui reprend la deuxième partie de la page d’inscription. Les champs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

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 des disponibilités avec les cases déjà cochées s’il a déjà des disponibilités, il peut les modifier ou en ajouter 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 lorsque l’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.

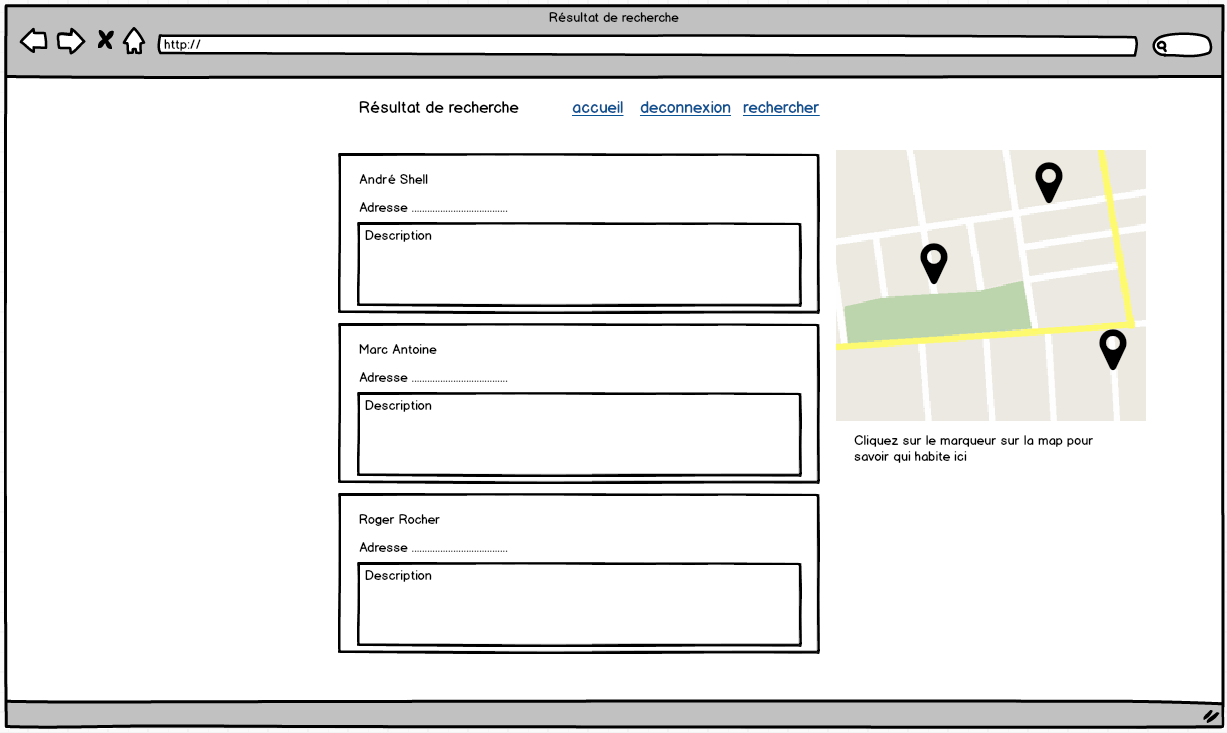
#### Page rechercher



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 maximale entre notre domicile et celui du gardien. Une fois le formulaire validé on est redirigé vers la page des résultats.

#### Page résultats

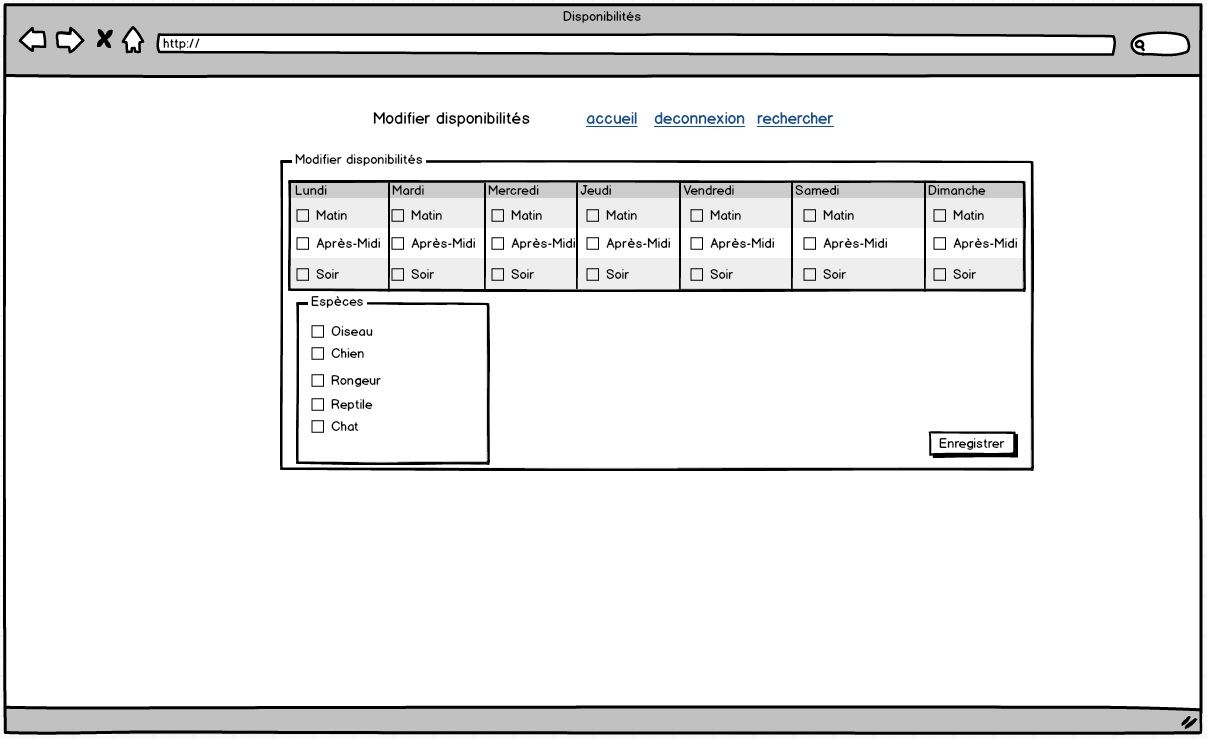


Cette page s’affiche lorsque l’on valide une recherche de gardien pour un animal.

Les gardiens sont affiché si ils sont d’accord de garder l’espèce d’animal que l’on cherche à faire garder, si ils 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 précisée sur la page de recherche.

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

#### Page disponibilités



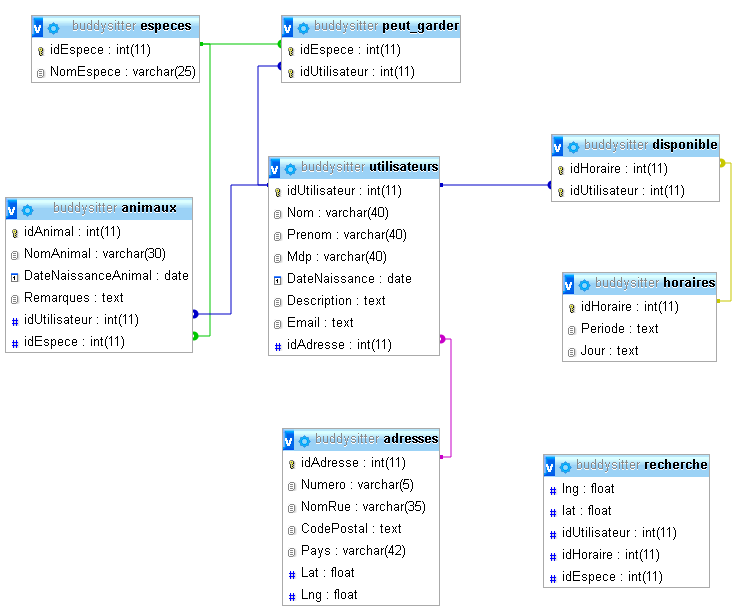
Cette page permet de modifier les disponibilités en tant que gardien. Par défaut lorsqu’un utilisateur s’inscrit il n’est jamais disponible et n’est donc pas gardien mais uniquement propriétaire. Il peut ensuite ajouter des disponibilités et donc devenir gardien via cette page qu’il peut atteindre depuis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

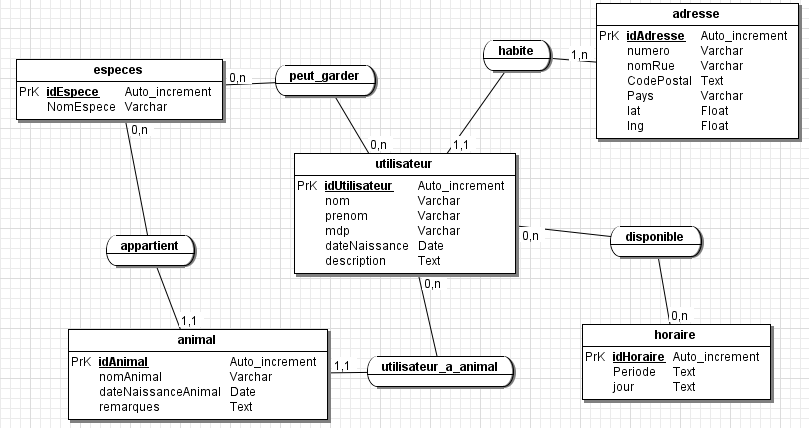
### La base de données

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



#### Le modèle conceptuel de données

Voici le modèle conceptuel 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’à 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 horaires (la cardinalité 0,n) et un horaire peut être choisi par personne 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’à un seul utilisateur (la cardinalité 1,1).

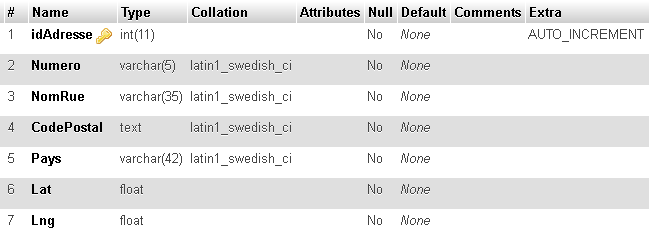
La relation « appartient » permet de relier la table animal à la table especes. Un animal ne peut appartenir qu’à une seule espèce (la cardinalité 1,1) et une espèce peut être utilisée aucune fois ou plusieurs fois (la cardinalité 0,n).

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 gardée par personne ou par plusieurs utilisateurs (la cardinalité 0,n)

### Description des tables

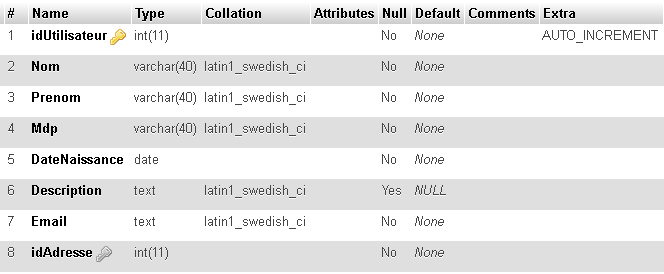
#### La table adresses

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.



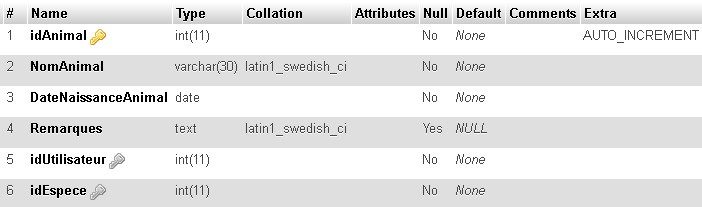
#### La table utilisateurs

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 mots de passe des utilisateurs sont stockés encodés en sha1.



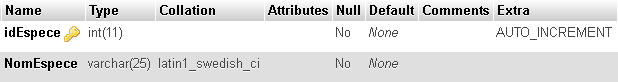
#### La table animaux

La table animaux stocke les informations des animaux. Les animaux sont reliés à leur maître par l’idUtilisateur qu’ils contiennent. Leur espèce est 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.



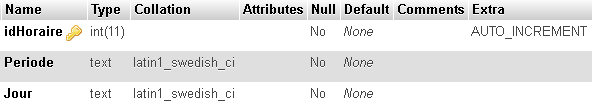
#### La table peut\_garder

La table peut\_garder permet de stocker 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 du jour de la semaine.



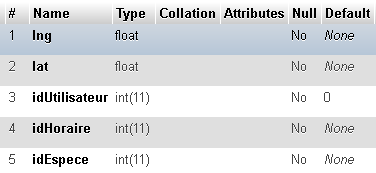
#### La table disponible

La table disponible permet de stocker les horaires ou un utilisateur est disponible. Elle est composée de 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 lequel 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 est 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. Si c’est la première fois qu’il met des disponibilités aucune case n’est cochée. 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 qu’aucune case soit cochée par défaut. La variable $horaires contient tous les horaires de la semaine de la table horaires. La variable $modif est pour dire si les disponibilités peuvent être modifiées 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’inscription.

##### 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 par d’autres fonctions sur la page dao.php.

##### 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 égal 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. Voici le tableau que cette fonction retourne :



##### 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. Voici le tableau qu’elle retourne :



##### 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 fonction est utilisée sur la page de profil et sur la page pour rechercher un gardien. Voici le tableau retourné :



##### 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. Cette fonction retourne le même tableau que la fonction ci-dessus.

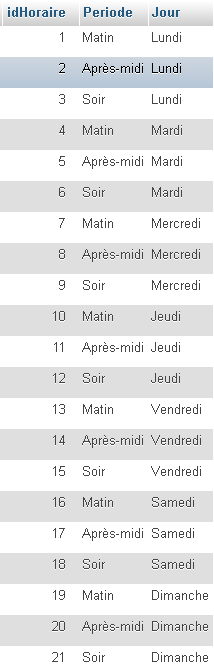
##### recupererAdresse($idAdresse)

Cette fonction permet de récupérer 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 pour modifier son adresse et sur la page de résultat d’une recherche. Voici le tableau qu’elle retourne :

##### 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. Cette fonction retourne le même tableau que la fonction recupererAdresse().

##### recupererHoraires()

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

##### 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 et la variable $idUtilisateur est l’id de l’utilisateur. Cette fonction retourne l’id de l’insertion.

##### deleteDisponibilites($idUtilisateur)

Cette fonction permet de supprimer toutes les disponibilités d’un utilisateur dont l’id est égal à $idUtilisateur. Pour supprimer les disponibilités la fonction fait un delete sur la table disponible quand l’id est le bon.

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

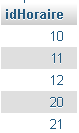
Cette fonction permet d’insérer les espèces que le gardien est d’accord de garder. La variable $idEspece correspond à l’id d’une espèce que le gardien peut garder et la variable $idUtilisateur contient l’id du gardien. Cette fonction retourne l’id de l’insertion.

##### supprimerChoixRaces($idUtilisateur)

Cette fonction permet de supprimer toutes les espèces que le gardien (dont l’id correspond à $idUtilisateur) était d’accord de garder.

##### recupererDisponibilites($idUtilisateur)

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



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

Cette fonction permet de modifier l’adresse dont l’id est égal à $idAdresse. Les nouvelles informations de l’adresse sont stocké dans les variables $numero, $rue et $npa. La latitude et la longitude de la nouvelle adresse sont récupérées grâce à l’API de Google map au moment de la validation du formulaire sur la page modifieradresse.

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

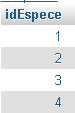
Cette fonction permet de modifier un utilisateur dont l’id est égal à $idUtilisateur. Les nouvelles informations de l’utilisateur sont stockées dans les variables $nom, $prenom, $mdp, $naissance, $description et $email.

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

Cette fonction permet de modifier un animal dont l’id est égal à $idAnimal. Les nouvelles données de l’animal sont stockés dans les variables $nom, $naissance, $remarque et $espece.

##### 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. Voici ce qui est retourné :



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

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. La variable $idUtilisateur contient l’id de l’utilisateur qui fait la recherche.

Le tableau $dispos est transformé en chaine de caractères grâce à la méthode implode afin que les disponibilités puissent être insérées 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 AND idUtilisateur != $idUtilisateur GROUP BY idUtilisateur ;

La première clause du where (idHoraire IN ( $ids )) permet de tester que le gardien est disponible pour chaque moment 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 domiciles 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 ».

La quatrième clause (idUtilisateur != $idUtilisateur) évite que la personne qui fait la recherche se retrouve dans les gardiens disponibles-

Les gardiens disponibles sont retournés dans un tableau. Voici une possibilité de tableau retourné par cette fonction :



#### La page ajoutanimal.php

Cette page permet à un utilisateur authentifié d’ajouter un animal de compagnie.

La première chose sur cette page est la vérification que l’utilisateur soit connecté. Pour cela on vérifie que la variable id dans le tableau de $\_session existe. Si ce n’est pas le cas le site redirige vers l’accueil.

La liste déroulante est créée automatiquement en php en utilisant le tableau qui est retourné par la fonction recupererEspeces().

Une fois que le formulaire est validé on inscrit l’animal en utilisant la fonction inscrptionAnimal.

#### La page connexion.php

Cette page permet à un utilisateur de se connecter au site.

La page connexion est créée entièrement en HTML et utilise le PHP que pour tester la connexion.

Premièrement on regarde si la variable erreur existe dans le tableau $\_GET, et si elle existe on regarde si elle vaut 1 si c’est le cas on stocke un message d’erreur qui sera affiché plus tard dans la page.

Ensuite quand le formulaire est validé on regarde que la fonction connexion() retourne plus au moins une ligne. Si c’est le cas l’utilisateur est connecté au site et on enregistre son id dans $\_session ce qui nous permettra sur toutes les pages de savoir si un utilisateur est connecté ou pas. Si la fonction ne retourne pas au moins une ligne on redirige vers la même page mais en ajoutant la variable erreur qui a comme valeur 1 pour afficher un message expliquant pourquoi il n’est pas connecté.

J’ai décidé de faire ce type d’erreur avec des numéros car on peut rencontrer plusieurs problèmes lors de la connexion et chaque problème aurait son numéro et donc son message d’erreur.

#### La page deconnexion.php

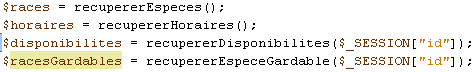
Cette page permet à un utilisateur de se déconnecter du site.

Cette page vide le tableau $\_session puis fait un session\_destroy qui détruit la session active. En vidant le tableau $\_session on supprime la variable id qui est stockée dedans. Puisque les tests pour savoir si un utilisateur est connecté sont faits avec cette variable, l’utilisateur est déconnecté.

#### La page disponibilites.php

Cette page permet à un utilisateur authentifié de changer ses disponibilités en tant que gardien.

Premièrement on récupère plusieurs informations de la base de données qui seront stockées dans des variables grâce à quatre fonctions comme suit :



Cette page est créée en partie par des fonctions php. Le tableau de disponibilités est créée grâce à la fonction afficherDisponibilites() qui prend en paramètres $disponibilités, $horaires et finalement le booléen true car les disponibilités peuvent être modifiées.

Les checkbox des espèces sont créés grâce à la fonction afficherEspeces qui prend comme paramètre $races et $racesgardables.

Ensuite lorsque le formulaire est validé on supprime toutes les disponibilités avec les fonctions deleteDisponibilites() et supprimerChoixRaces() puis si des cases ont été coché pour les races et pour les disponibilités les nouvelles disponibilités sont ajouté dans la base de données grâce aux fonction insererDisponibilites() et insererChoixRace (). On est redirigé vers la page de profil.

Si seuls des moments de la semaine ont été sélectionnés un message d’erreur s’affiche pour avertir l’utilisateur qu’il doit choisir aussi des espèces à garder. Si seules des espèces ont été choisies mais aucun moment dans la semaine un message d’erreur s’affiche pour avertir l’utilisateur qu’il doit aussi choisir un moment de la semaine. Si rien n’a été coché on part du principe que c’est voulu par l’utilisateur et qu’il ne veut plus garder d’animaux et il est redirigé vers la page de profil.

#### La page index.php

La page index est la page d’accueil du site et la première que l’utilisateur voit.

Les liens dans la navigation sont différents suivant si l’utilisateur est connecté ou pas. S’il n’est pas connecté il a les liens de connexion et la page d’inscription. S’il est connecté il voit les liens de profil, de recherche et de déconnexion.

#### La page inscription.php

Cette page permet à un utilisateur de s’inscrire sur le site. Il doit y rentrer son adresse et ses informations personnelles.

Le formulaire est en sticky form ce qui veut dire que s’il y a un problème pendant l’inscription les données rentrées dans le formulaire ne seront pas perdues.

Lorsque le formulaire est validé on remplace les espaces dans le nom de la rue par des « + », car pour faire la recherche de la latitude et longitude il ne faut pas d’espaces. Ensuite on utilise l’API de Google map pour récupérer la latitude et la longitude depuis l’adresse. Si aucune adresse n’a été trouvée par l’API un message d’erreur en avertit l’utilisateur. Sinon on utilise la fonction inscriptionUtilisateur() avec les données du formulaire pour inscrire l’utilisateur. Puis on connecte l’utilisateur grâce à la fonction connexion() et on met son id dans le $\_session. Il est ensuite redirigé vers la page d’accueil.

#### La page modifieradresse.php

Cette page permet à un utilisateur authentifié de modifier son adresse.

Pour afficher les données actuelles dans le formulaire on récupère l’adresse de l’utilisateur grâce à recupererAdresse().

Lorsque l’utilisateur valide le formulaire les espaces du nom de la rue sont remplacés par des « + » pour réutiliser l’API de Google map pour récupérer la longitude et la latitude. Si aucune adresse n’a été trouvée par l’API un message en avertit l’utilisateur. Sinon les nouvelles données sont mises dans la base de données grâce à la fonction modifierAdresse().

#### La page modifieranimal.php

Cette page permet à un utilisateur authentifié de modifier les informations d’un animal.

La liste déroulante est affichée grâce à du PHP. Lorsque le formulaire est validé on modifie les données de l’animal dans la base de données grâce à la fonction modifierAnimal().

#### La page modifierinformations.php

Cette page permet à l’utilisateur de modifier ses données personnelles.

Les données de l’utilisateur sont récupérées grâce à la fonction recupererUtilisateur(). Les données sont ensuite affichées dans le formulaire.

Lorsque le formulaire est validé les données sont modifiées grâce à la fonction recupererUtilisateur().

#### La page profil.php

Cette page permet d’afficher toutes les informations d’un utilisateur. Il peut y voir ses informations personnelles, son adresse ses animaux et ses disponibilités.

Beaucoup de données doivent être récupérées dans la base de données pour que tout s’affiche sur la page. Elles sont récupérées comme suit :



Les informations de l’utilisateur sont affichées grâce à la fonction afficherUtilisateur() et l’adresse avec la fonction afficherAdresse().

Les animaux sont affichés avec la fonction afficherAnimaux() et les disponibilités avec la fonction afficherDisponibilites(). Cette dernière fonction prend comme paramètre $disponibilites, recupererHoraires() et false car les disponibilités ne peuvent pas être modifiées sur cette page.

#### La page rechercher.php

Cette page permet à un utilisateur authentifié de rechercher un gardien.

Cette page peut afficher deux messages d’erreur, ce sont les premiers tests qui le font.

Plusieurs données doivent être récupérées dans la base de données comme suit :



La liste déroulante des animaux est affichée avec la fonction afficherAnimauxSelect().

Le nom du propriétaire est récupéré dans le tableau $utilisateur.

Le tableau des moments à garder est affiché grâce à la fonction afficherDisponibilites(). Le premier paramètre est un tableau vide car aucune case ne doit déjà être cochée, le deuxième paramètre est $horaires et le troisième est true car les cases peuvent être cochées.

Le formulaire de cette page est traité par la page resultats.php.

#### La page resultats.php

Cette page permet d’afficher les gardiens disponibles pour une certaine recherche.

On commence par récupérer les informations de l’animal à faire garder et de l’adresse du propriétaire. Ensuite grâce à la fonction rechercherGardien() on cherche des gardiens. Si aucun gardien n’est trouvé on redirige vers la page de recherche avec le code d’erreur « 2 » qui affichera un message d’erreur pour dire que personne n’est disponible.

Si le formulaire a été envoyé sans aucun moment de la semaine on redirige vers la page de recherche avec le code d’erreur « 1 » qui affichera un message d’erreur demandant de choisir un moment dans la semaine.

La carte de Google map est ensuite créé et deux fonctions sont initialisées.

##### 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 où 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. Cet événement se produit lors d’un clique et affiche une fenêtre d’information qui contient le nom du gardien récupéré grâce à la variable nom.

La carte est ensuite affichée et centrée sur le premier gardien disponible.

Pour chaque gardien on va utiliser la fonction afficherGardien() qui affiche un petit résumé du gardien et placer un marqueur sur la carte avec la fonction addMarker().

### 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 | Sur la page pour rechercher un gardien | Avoir le choix entre les différents animaux |
| 12 | HN | Sans avoir d’animaux essayer de rechercher un gardien | On ne peut pas si l’on ne sélectionne pas d’animal |
| 13 | HN | Rechercher un gardien pour un animal sans sélectionner de moment dans la semaine | Affichage d’un message d’erreur qui demande qu’on sélectionne un moment dans la semaine |
| 14 | HN | Faire une recherche pour un animal en sélectionnant un moment de la semaine | Être redirigé vers la page de résultats et avoir la carte avec au moins un marqueur et un gardien affiché |
| 15 | HN | Sur la page de résultats | Avoir les gardiens qui s’affichent |
| 16 | HN | Sur la page de résultat cliquer sur le marqueur | Avoir le nom du gardien qui s’affiche |
| 17 | HN | Sur la page d’inscription valider le formulaire sans rien rentrer | Un message demande de remplir les données. |
| 18 | HN | Sur la page d’inscription ne pas mettre les deux mêmes mots de passe | Un message demande de remettre les mots de passes sans perdre le reste du formulaire. |
| 19 | HN | Sur la page d’inscription remplir le formulaire correctement | Être inscrit et connecté au site |
| 20 | HN | Sur la page pour modifier ses informations | Les champs sont déjà remplis sauf les mots de passe |
| 21 | HN | Sur la page pour modifier ses informations ne pas mettre les mêmes mots de passe | Un message d’erreur avertit l’utilisateur |
| 22 | HN | Sur la page pour modifier ses informations changer le nom | Sur le profil le nom est changé |
| 23 | HN | Sur la page pour modifier son adresse | Les champs sont déjà remplis |
| 24 | HN | Sur la page pour modifier son adresse modifier une donnée | La donnée est changée sur la page de profil |
| 25 | HN | Sur la page pour modifier un animal changer le nom de l’animal | Le nom de l’animal est modifié sur la page du profil |
| 26 | HN | Sur la page pour modifier ses disponibilités modifier ses disponibilités | Les nouvelles disponibilités s’affichent sur le profil |
| 27 | HN | Ne mettre que des disponibilités pour les espèces mais pas pour la semaine | Avoir un message d’avertissement qui avertit qu’on n’a pas mis de moment dans la semaine |
| 28 | HN | Ne mettre que des disponibilités pour la semaine et pas pour les espèces | Avoir un message d’avertissement |
| 29 | HN | Sur la page pour ajouter un animal ne remplir aucun champs et valider le formulaire | Un message demande de remplir les champs |
| 30 | HN | Sur la page pour ajouter un animal ajouter un animal | Le nouvel animal s’affiche sur la page de profil |

### Rapport de test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N° de test | Testeur | Résultat obtenu | Ok | N° anomalie |
| 1 | HN | Les trois rubriques sont là L’image aussi et les liens connexion et inscription | Oui |  |
| 2 | HN | Les liens connexion et inscription sont affichés | Oui |  |
| 3 | HN | Je suis redirigé vers la page de connexion | Oui |  |
| 4 | HN | Je suis redirigé vers la page d’inscription | Oui |  |
| 5 | HN | Je suis connecté et redirigé vers la page d’accueil | Oui |  |
| 6 | HN | Un message dit « Le mot de passe et le nom d’utilisateur ne correspondent pas » | Oui |  |
| 7 | HN | J’ai les lien de profil rechercher et déconnexion | Oui |  |
| 8 | HN | Je suis redirigé vers ma page de profil | Oui |  |
| 9 | HN | Toutes les informations sont affichées. Les liens disponible sont : accueil, ajouter un animal, disponibilités, rechercher et déconnexion | Oui |  |
| 10 | HN | Je suis redirigé vers la page d’accueil. | Oui |  |
| 11 | HN | Je peux choisir un animal | Oui |  |
| 12 | HN | Un message dit « Personne ne correspond à votre recherche » | Oui |  |
| 13 | HN | Un message dit « Veuillez rentrer un moment de la semaine> | Oui |  |
| 14 | HN | Un gardien est affiché et aussi un marqueur | Oui |  |
| 15 | HN | Un gardien s’affiche | Oui |  |
| 16 | HN | Le nom du gardien s’affiche dans une petite fenêtre blanche | Oui |  |
| 17 | HN | Un message dit veuillez renseigner ce champ | Oui |  |
| 18 | HN | Un message dit « Les deux mots de passe ne correspondent pas » | Oui |  |
| 19 | HN | Je suis inscrit et connecté au site | Oui |  |
| 20 | HN | Les champs sont déjà remplis sauf les mots de passe | Oui |  |
| 21 | HN | Un message dit « Les deux mots de passe ne correspondent pas » | Oui |  |
| 22 | HN | Le nouveau nom s’affiche sur le profil | Oui |  |
| 23 | HN | Les quatre champs sont déjà remplis | Oui |  |
| 24 | HN | La nouvelle donnée s’affiche sur le profil | Oui |  |
| 25 | HN | Le nouveau nom s’affiche sur le profil | Oui |  |
| 26 | HN | Les nouvelles disponibilités s’affichent sur le profil | Oui |  |
| 27 | HN | Aucun message d’avertissement n’apparait et les nouvelles disponibilités s’affichent | NON | Err001 |
| 28 | HN | Aucun message d’avertissement n’apparait et les nouvelles disponibilités s’affichent | NON | Err002 |
| 29 | HN | Un message demande de remplir les champs | Oui |  |
| 30 | HN | Le nouvel animal s’affiche sur le profil | Oui |  |

### Conclusion

#### Respect du cahier des charges

Le travail final respecte le cahier des charges. En effet un utilisateur peut décrire ses animaux de compagnie et ses données personnelles. Un utilisateur peut endosser le rôle de propriétaire ou de gardien. En tant que gardien il peut proposer ses disponibilités à la demi-journée. En tant que propriétaire il peut choisir les moments de gardiennage et il peut rechercher des gardiens dans une région donnée. Le site propose des gardiens pour l’utilisateur dans la région donnée et le propriétaire peut ensuite choisir un gardien. Les utilisateurs sont effectivement géolocalisés afin de proposer des gardiens qui soient dans la même région.

#### Différence de la base de données

Le modèle conceptuel de données prévu et celui utilisé ont des différences. Premièrement car le modèle prévu était une ébauche et ne comprenait pas toutes les colonnes qui seraient utilisées pour le projet. Deuxièmement car plus je réfléchissais au fonctionnement du site plus je rencontrai des problèmes qui me demandaient de modifier le modèle.

#### Différence des maquettes

Il y a plusieurs différences entre les maquettes et le design final du site. Les principales différences sont dans les liens qui sont affichés dans la barre de navigation. En effet sur les maquettes les liens sont différents suivant la page sur laquelle l’utilisateur se trouve. Sur le site les liens sont quasiment partout les mêmes, ce qui permet une navigation plus aisée. L’affichage de la page de résultats est inversé. C’est-à-dire que la carte au lieu d’être à droite se trouve à gauche. Cela n’a pas été un choix de ma part mais une contrainte du code. Sur la page de profil un lien pour ajouter un animal a été ajouté sous le tableau des animaux.

#### Difficultés particulières

La plupart des difficultés que j’ai rencontrées étaient des erreurs de ma part lors de l’écriture du code. Par exemple de simple faute de frappes m’ont parfois pris plusieurs minutes avant de les trouver.

J’ai rencontré quelques problèmes pour l’affichage de la carte et des marqueurs sur la page de résultats. En effet pour avoir un code propre je voulais que la partie en JavaScript ne soit pas au milieu de la page, mais en le plaçant au début ou à la fin cela ne fonctionnait pas et j’ai tenté différentes manières mais rien ne fonctionnait. Ce problème aurait pu être évité si je m’étais résolu plus rapidement à déplacer le code en JavaScript.

#### Ce que ce projet m’a apporté

Ce projet était très intéressant car c’était un des premiers projets que nous devions réaliser entièrement de l’idée jusqu’à la documentation utilisateur.

Avant de commencer ce projet j’ai appris à utiliser l’API de Google map et plus précisément la fonctionnalité de Géocoding.

#### Suites possible du projet

Le projet correspond au cahier des charges mais certaines choses pourraient être améliorées ou ajoutées. Par exemple une des choses qui pourrait être ajoutée serait la fonctionnalité d’ajouter des photos pour son profil ou pour ses animaux. On pourrait ajouter une fonctionnalité pour noter les gardiens qui ont gardé notre animal. Mais pour cela il faudrait savoir si le gardien a bien gardé notre animal pour donner accès au formulaire de note. Cela représente une grosse quantité de travail et l’ajout de tables dans la base de données.

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)