

# **Pet Weight**

Chef de projet : Cindy Hardegger

Expert n°1 : Alain Roy

Expert n°2 : Nicolas Glassey

Table des matières

1	Analyse préliminaire .....	4
1.1	Introduction .....	4
1.2	Objectifs.....	4
1.3	Planification initiale .....	5
2	Analyse / Conception.....	5
2.1	Outils .....	5
2.2	Méthode de travail .....	5
2.3	Dossier de conception .....	5
2.3.1	MCD .....	6
2.3.2	MLD .....	7
2.3.3	Framework.....	8
2.3.4	Maquettes.....	8
2.4	JavaScript / SQL.....	13
2.4.1	Connexion à la base de données. ....	13
2.4.2	Connexion/Déconnexion.....	13
2.4.3	Lister les animaux de compagnie .....	13
2.4.4	Ajouter un nouveau poids .....	13
2.4.5	Ajouter un nouvel animal .....	13
2.4.6	Modifier un animal .....	14
2.4.7	Modification d'un utilisateur .....	14
2.4.8	Graphiques .....	14
2.4.9	Profil utilisateur .....	14
2.5	Stratégie de test.....	15
2.6	Risques techniques .....	17
2.7	Planification .....	17
3	Réalisation.....	17
3.1	Dossier de réalisation .....	17
3.1.1	Répertoire.....	17
3.1.2	Connexion .....	17
3.1.3	Inscription .....	18
3.1.4	Affichage des animaux de compagnie .....	18
3.1.5	Affichage des graphiques .....	19
3.1.6	Ajouter un animal de compagnie .....	20
3.1.7	Ajouter un nouveau poids .....	21
3.1.8	Modifier un utilisateur.....	21
3.1.9	Supprimer un animal.....	22
3.1.10	Modifier un animal de compagnie .....	22
	Utilisateurs de la base de donnée .....	22
3.2	Problèmes rencontrés.....	22
3.3	Description des tests effectués .....	24
3.4	Erreurs restantes .....	30
3.5	Liste des documents fournis .....	31
4	Conclusions .....	31
4.1	Bilan personnel.....	31

---

4.2	Améliorations possibles .....	31
5	Annexes.....	32
5.1	Glossaire .....	32
5.2	Résumé du rapport du TPI / version succincte de la documentation.....	32
5.3	Sources .....	33
5.4	Journal de travail .....	33
5.5	Manuel d'Installation .....	34
5.6	Manuel d'Utilisation.....	34
5.7	Archives du projet.....	34

# **1 Analyse préliminaire**

## **1.1 Introduction**

Ce projet est réalisé dans le cadre de mon TPI de 4<sup>ème</sup> année. Le but étant de confirmer les connaissances acquises au cours de ma formation.

Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1er jour. Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.

Le candidat a connaissance de la feuille d'appréciation avant de débiter le travail. Le candidat est entièrement responsable de la sécurité de ses données.

En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP. Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier.

A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L'un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.

## **1.2 Objectifs**

L'objectif est de fournir une application permettant de gérer le poids de ses animaux de compagnie. L'application doit être visualisable et fonctionnelle sur smartphone.

Il doit être possible d'afficher pour chaque animal une série de données, de rentrer le poids à une date et de visualiser un graphique de l'évolution du poids.

### **Conception de l'application**

- Elaborer une maquette de l'application (1<sup>re</sup> page). Les choix de conception doivent être justifiés dans le rapport.
- Modéliser et mettre en oeuvre une base de données (MCD, MLD et MPD).
- Concrétiser le rendu graphique de votre maquette (Création du html et css avec un Framework CSS – justifier le choix).

### **Caractéristique d'un animal de compagnie**

L'application doit permettre de visualiser, insérer, modifier un animal.

Un animal de compagnie peut être un chien ou un chat (pour une évolution future de l'application, il doit être possible d'ajouter d'autres types d'animaux dans un second temps).

Il est caractérisé par un nom, une date de naissance (attention, parfois, il n'y a que l'année de naissance et pas la date complète), une puce électronique (pas obligatoire), un descriptif (pas obligatoire) et une photo (pas obligatoire – une photo par défaut sera insérée). Par la suite, il est possible que d'autres éléments viennent s'ajouter.

En cas de décès de l'animal, ajouter une date de décès et faire en sorte de voir l'information visuellement si l'option "Voir les animaux décédés" est cochée dans la configuration de l'utilisateur.

### 1.3 Planification initiale

La planification initiale se trouve en annexe : [planif\\_initial.pdf](#)

## 2 Analyse / Conception

### 2.1 Outils

- Ordinateur ETML Standard

Tout le déroulement de ce projet va être exécuté à l'aide d'un ordinateur standard fourni par l'ETML ainsi que les logiciels ci-dessous.

- Serveur SQL

Afin d'exploiter la base de données il faut utiliser un serveur MySQL. J'ai opté pour EasyPHP car il est mis à disposition par l'ETML et intègre phpMyAdmin qui est un outil très utile pour gérer ma base de données facilement.

- PhpMyAdmin

Pour la gestion de la base de données j'ai utilisé l'outil PhpStorm

- PhpStorm

Pour coder la suite j'ai utilisé PhpStorm à la version 2017.1.4

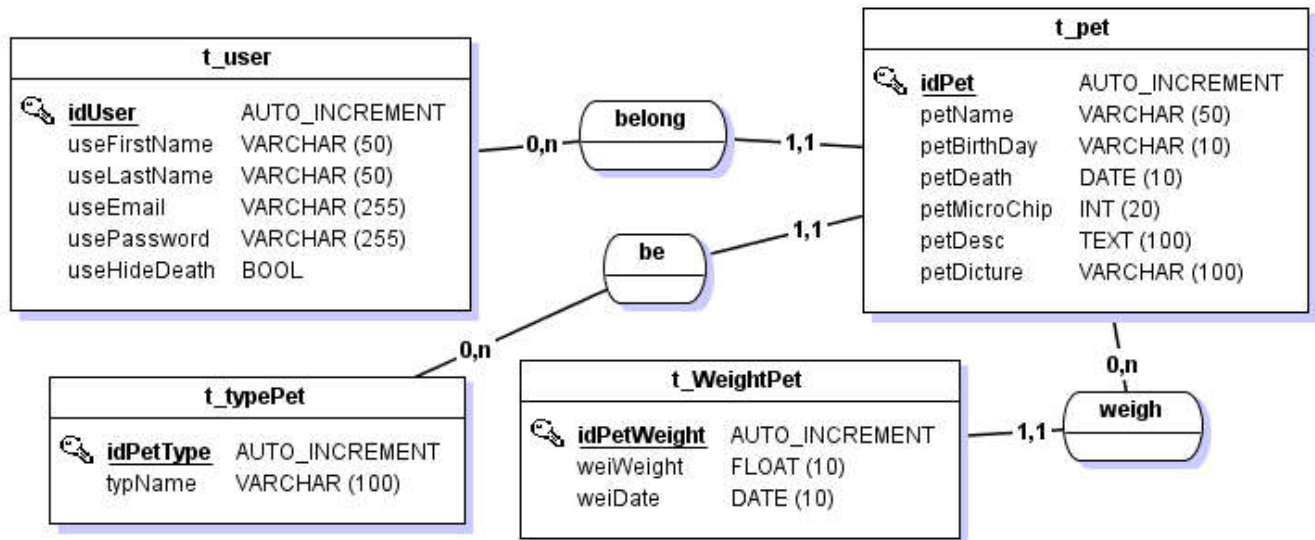
### 2.2 Méthode de travail

La méthode de travail utilisée pour ce projet est la méthode Gantt. Cette méthode est adaptée car je travaille seul. Avec la méthode agile j'aurai divisé mon projet en plusieurs parties. Hors, un projet de 89 heures est suffisamment court pour que le projet soit fait d'une traite.

Chaque étape du développement du site web est documentée au fur et à mesure. Tous les mercredi et vendredi une version du site ainsi qu'une version du rapport et du journal de travail est enregistrée et la date de sauvegarde est indiquée. Ces sauvegardes sont ensuite uploadées sur GitHub et gardé dans mon H: de l'ETML. De cette façon je peux récupérer à tout moment mes documents si mon profil ETML est supprimé.

### 2.3 Dossier de conception

## 2.3.1 MCD



MCD

Dans la mise en place du site web il est nécessaire de mettre en place une base de données. Pour ce faire je commence par créer un MCD (modèle conceptuel de données).

Le MCD est constitué de quatre tables.

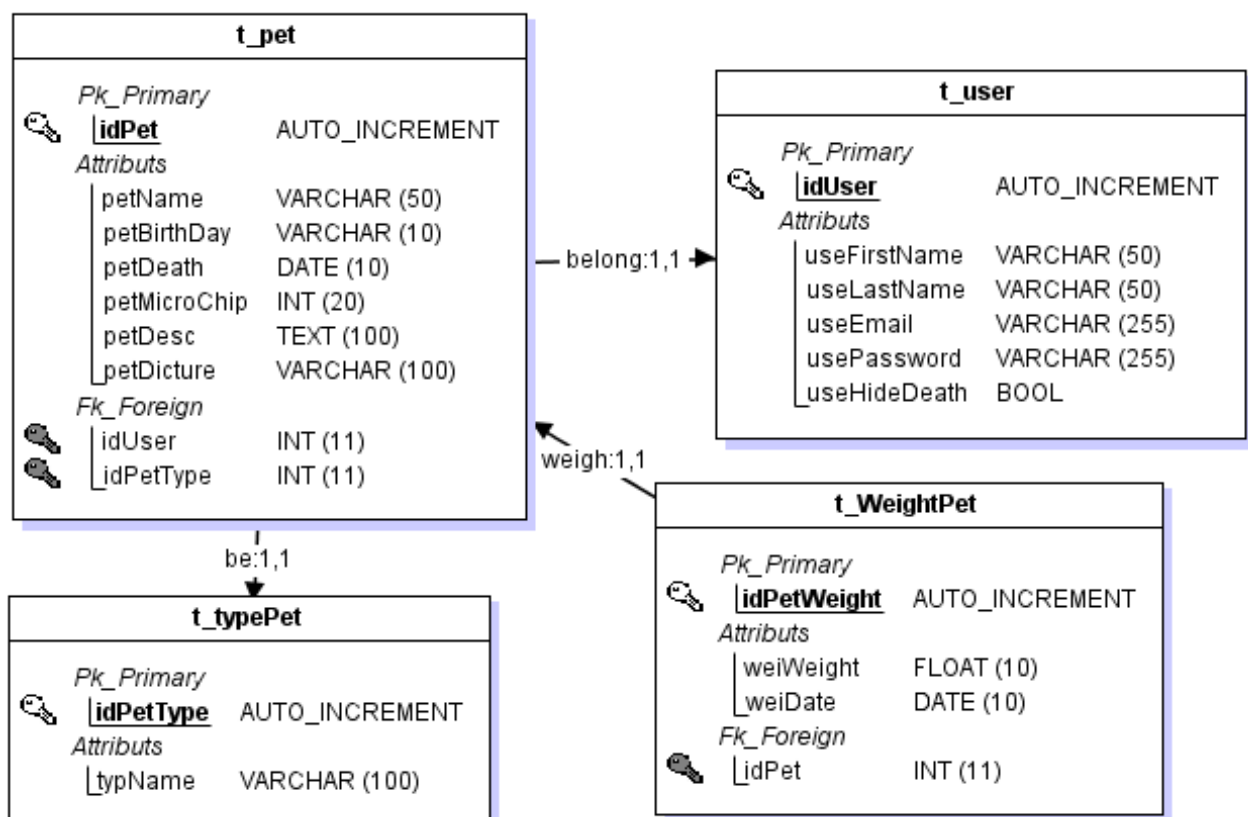
- **t\_user**  
La table **t\_user** est constituée des paramètres utilisateurs. Chaque utilisateur possèdera un prénom et un nom. Il possèdera également un email qui fera office de login. Lorsqu'un utilisateur s'inscrit on s'assure que l'email n'est pas déjà utilisé. De cette façon on s'assure que chaque utilisateur est unique. De plus l'email peut être utile lors de l'oubli du mot de passe. Le champ **useHideDeath** est utilisé afin de définir si l'utilisateur souhaite afficher la date de décès des animaux. Il sera rempli de la valeur "0" ou "1". Si c'est la valeur 0, alors les dates seront visibles.
- **t\_pet**  
La table **t\_pet** est constituée des paramètres des animaux de compagnies. Chaque animal de compagnie possèdera un nom ainsi qu'une date de naissance. Une race lui sera également attribuée obligatoirement grâce au lien de la table **t\_pet** et de la table **t\_typePet**. Chaque animal de compagnie pourra aussi être caractérisé par une Microchip, une description, une image et une date de décès.
- **t\_TypePet**  
La table **t\_typePet** permet de créer des races d'animaux de compagnies. Ce paramètre a été mis dans une table à part car il est spécifié dans le cahier des charges que des catégories pourront être ajoutées plus tard. De cette manière on peut également s'assurer qu'un utilisateur ne donne pas à son animal de compagnie une race inexistante.
- **t\_Weightpet**

La table `t_Weightpet` quant à elle va contenir les différents poids. Chaque poids sera caractérisé par le poids en lui-même ainsi que par une date. Il a été décidé de créer une table à part contenant les différents poids car si le poids était un champ de la table `t_pet` un animal ne pourrait posséder qu'un seul poids et on ne pourrait pas avoir accès à l'historique de tous les poids.

- Liaisons

Lorsqu'un utilisateur ajoute un animal, celui-ci sera obligatoirement doté d'une seule race. De même lorsqu'un poids est ajouté, celui-ci ne peut appartenir qu'à un seul animal. Cependant un animal peut ne posséder aucun poids ou en posséder plusieurs. Un utilisateur peut ne posséder aucun animal ou en posséder plusieurs.

### 2.3.2 MLD



MLD

Dans le MLD (modèle logique des données) les différentes tables sont liées de la manière suivante.

La table `t_pet` possède deux clés étrangères. La première, `idUser` permet de définir à qui appartient l'animal de compagnie. La seconde, `idPetType`, permet d'identifier la race de l'animal. Chaque animal ne possède qu'un seul maître et qu'une seule race. Le champs `petBirthDay` est en format `varchar` car comme discuté avec mon chef de projet un utilisateur pourra entrer une date sans préciser l'année. Et si cela arrive et que dans la base de données ce champ est un champ `date` cela va causer des erreurs.

La table `t_WeightPet` possède une seule clé étrangère. Elle permet d'identifier à quel animal appartient le poids. Un poids ne peut appartenir qu'à un seul animal, ni plus, ni moins.

La table `t_typePet` quant à elle ne possède pas de clé étrangère. Elle est reliée à la table `t_pet` car chaque race peut contenir zéro ou plusieurs animaux. Il en est de même pour la table `t_user`.

Le champ `usePassword` est limité à 255 caractères car les mots de passe seront hashés à l'aide de la méthode `password_hash`. Avec cette méthode le hash du mot de passe ne peut pas dépasser 255 caractères.

### 2.3.3 Framework

Le site web sera développé à l'aide d'un Framework comprenant une librairie graphique. Celui-ci permet de gérer toute la partie graphique du site à l'aide de CSS et de Javascript préconfigurés.

Pour ce projet je vais utiliser le Framework nommé Materialize.

J'ai choisi ce Template car lors de la plupart de mes projets j'ai utilisé celui-ci, c'est donc le Template que je maîtrise le mieux. Il possède à la fois des éléments css, afin de gérer la partie design du site, mais également du javascript, afin de faciliter certaines fonctionnalités telle que le responsive.

Bien que Bootstrap soit supporté par presque tous les navigateur web et qu'il possède également une librairie graphique j'ai favorisé Materialize, car je trouve le design des icônes plus modernes. De plus celui-ci est également supporté par beaucoup de navigateurs.

### 2.3.4 Maquettes

Le site sera constitué de 9 pages.

- Page de connexion

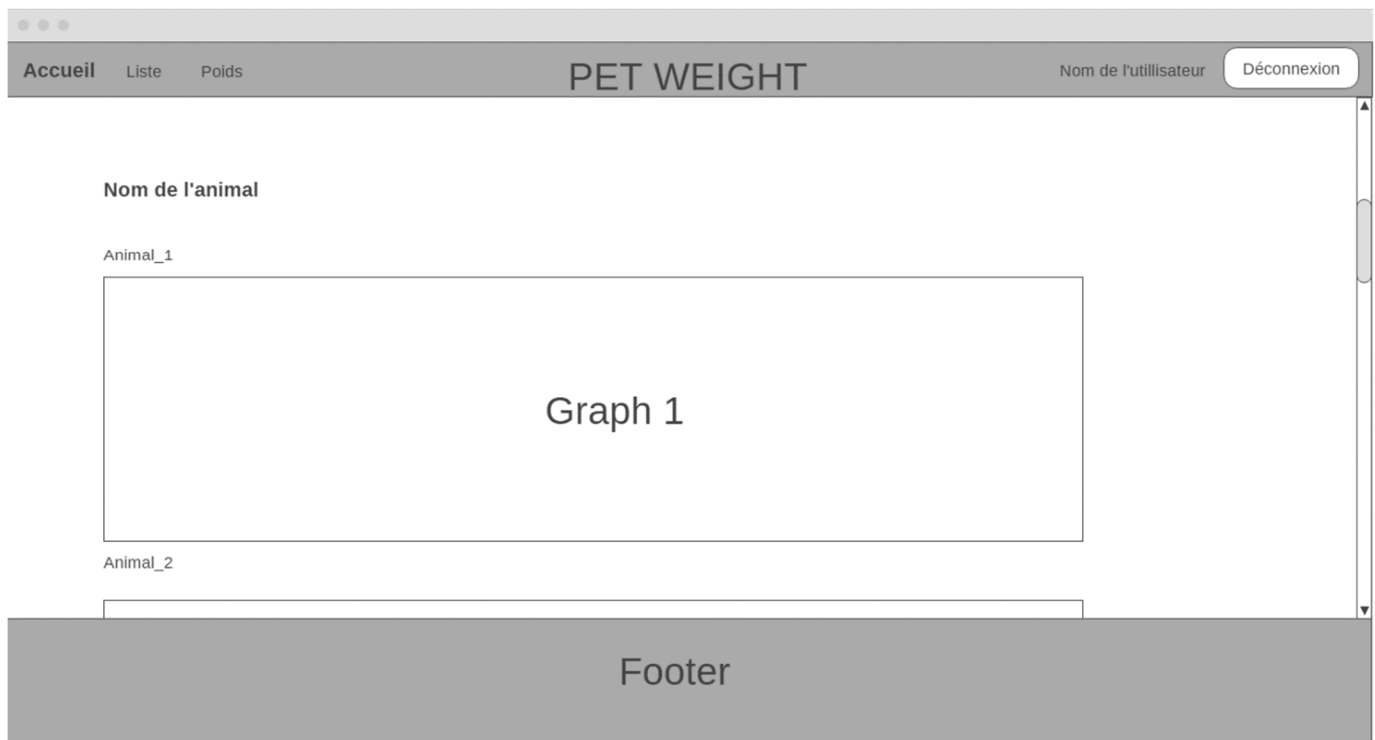




*Page de connexion*

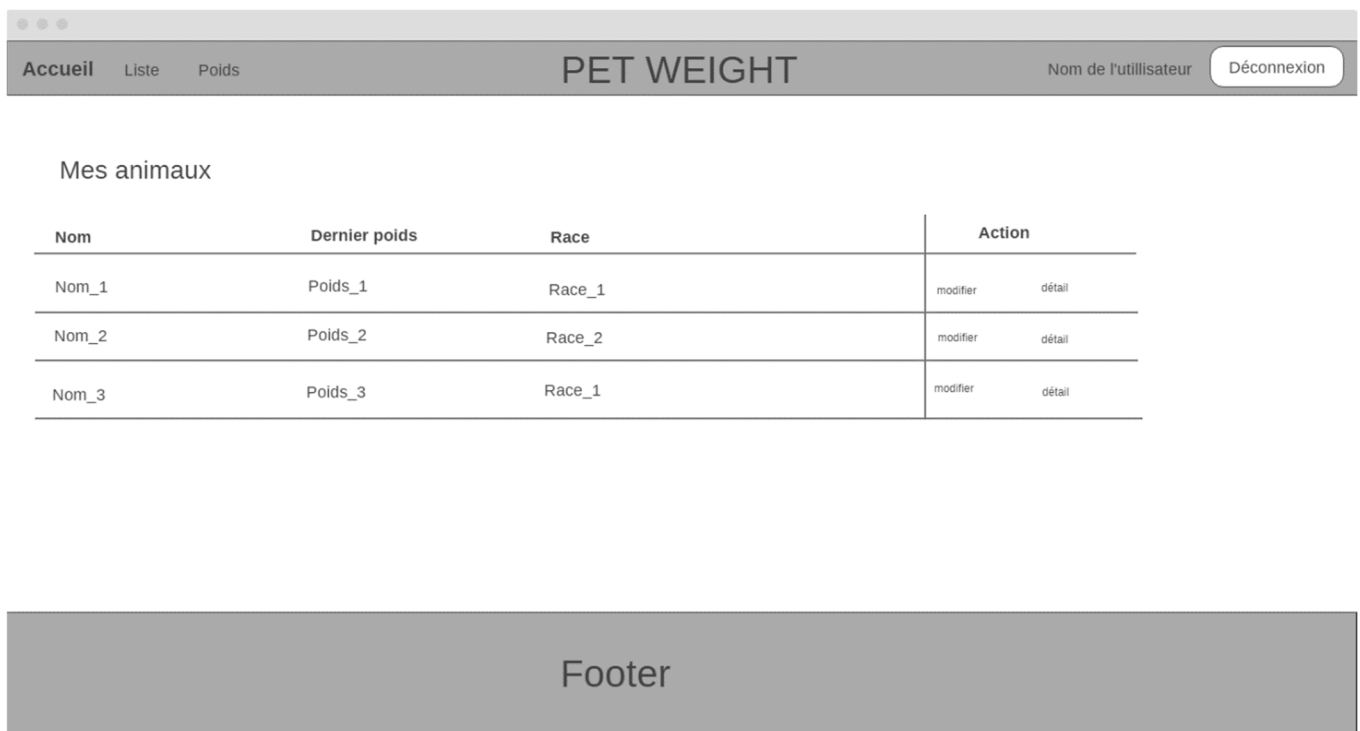
Cette page apparaît lors de l'arrivée sur le site. Il suffit d'entrer son email ainsi que son mot de passe. Si l'utilisateur ne s'est pas encore inscrit une deuxième page s'ouvre afin de lui permettre de s'inscrire. J'ai pris exemple sur la page de connexion de Google, que j'aime beaucoup car elle est épurée.

- Page d'inscription  
La page sera quasiment identique à la page de connexion
- Page des graphiques affichants les poids

*Affichage des graphiques*

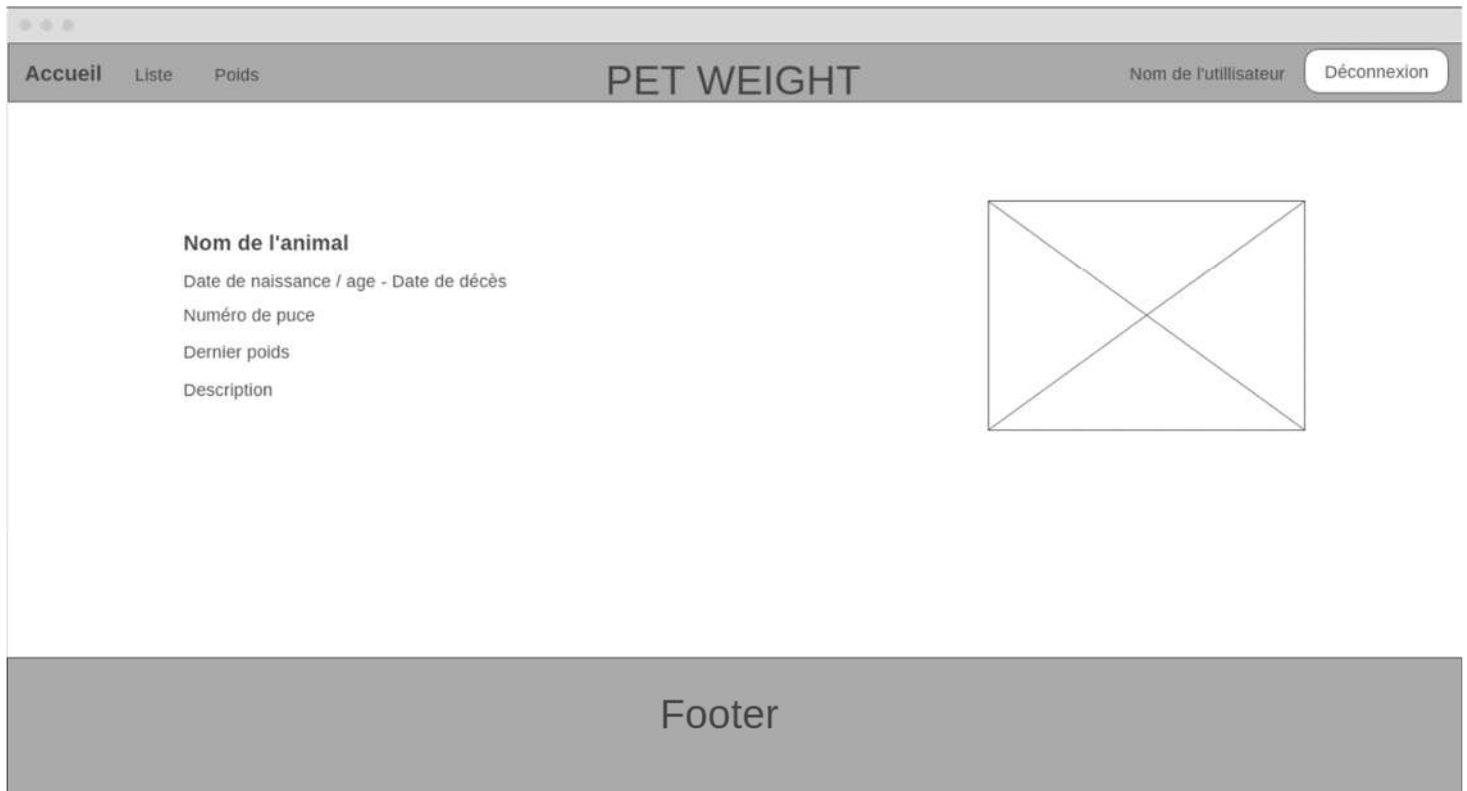
Cette page permettra à l'utilisateur de voir l'évolution du poids de ses animaux de compagnie à l'aide de graphiques.

- Page listant ses animaux de compagnie

*Affichage de animaux*

Cette page permet d'afficher ses animaux de compagnie. Ainsi que d'accéder aux détails de ceux-ci et de les modifier.

- Page affichant les détails d'un animal



*Détail d'un animal*

Dans cette page se trouve tous les détails d'un animal de compagnie. Si l'utilisateur n'a pas coché l'option afin d'afficher les dates de décès elles ne s'afficheront pas. Il est également possible de visualiser une photo de l'animal de compagnie.

- Page d'ajout de poids

The screenshot shows a web browser window with the title 'PET WEIGHT'. The navigation bar includes 'Accueil', 'Liste', and 'Poids'. The main content area is titled 'Mes animaux' and contains a form for adding a new weight. The form has a title 'Entrer un nouveau poids' and three input fields: 'Animal' (a dropdown menu), 'Date' (a text input), and 'Poids' (a text input with a small up/down arrow icon). Below these fields is an 'Ajouter' button. The footer area is labeled 'Footer' and contains the text 'Ajout d'un poids'.

Cette page permet d'ajouter un poids à un animal. Cette nouvelle donnée est alors visible dans le graphique correspondant sur la page des graphiques, mais également dans les détails de l'animal.

- Page affichant le profil d'un utilisateur

The screenshot shows a web browser window with the title 'PET WEIGHT'. The navigation bar includes 'Accueil', 'Liste', and 'Poids'. The main content area is titled 'Nom de l'utilisateur' and contains a form for user profile management. The form has several sections: 'Prénom' (text input), 'Nom de famille' (text input), 'Changer de mot de passe' (a section with three text inputs for 'Mot de passe actuel', 'Nouveau mot de passe', and 'Configurer nouveau mot de passe'), and a radio button labeled 'Voir le décès des animaux'. Below these inputs is a 'Sauvegarder la configuration' button. The footer area is labeled 'Footer' and contains the text 'Page utilisateur'.

Sur cette page on peut visualiser ses informations personnelles telles que son email, son nom et son prénom. Il est également possible changer son mot de passe et de choisir d'afficher les dates de décès.

## 2.4 JavaScript / SQL

### 2.4.1 Connexion à la base de données.

Pour que le site web puisse récupérer les données de façon dynamique il doit être connecté à la base de données. Pour cela j'utilise une fonction.

### 2.4.2 Connexion/Déconnexion

Lors de la connexion l'utilisateur utilise son email. De cette manière plusieurs utilisateurs peuvent avoir le même nom sans que cela ne pose problème. Il permet également d'envoyer un email de réinitialisation au cas où celui-ci ne se souvient pas de son mot passe. Cette fonctionnalité n'est pas prévue mais si plus tard il est décidé de la mettre en place on pourra le faire sans problème.

Le mot de passe sera également hashé à l'aide la fonction "password\_hash". De cette manière si un utilisateur parvient à accéder à la base de données il n'a pas accès aux mots de passe.

Lorsqu'un utilisateur se connecte le mot de passe qu'il rentre est hashé à l'aide de la même méthode. Il ne reste alors plus qu'à comparer les deux hash. Si toutes les conditions sont remplies la session est ouverte et l'utilisateur est alors redirigé vers la liste de ses animaux.

Lorsque l'utilisateur est connecté le bouton Déconnexion se trouve en haut à droite. En cliquant sur celui-ci la session est détruite et l'utilisateur est redirigé à la page de Connexion.

### 2.4.3 Lister les animaux de compagnie

Chaque utilisateur a la possibilité de visualiser tous ses animaux sur une seule page. Cette page sera considérée comme la page d'accueil des utilisateurs. Depuis cette page le détail de chaque animal est accessible.

### 2.4.4 Ajouter un nouveau poids

Comme démontré dans les maquettes, la page permettant d'ajouter le poids d'un animal est constitué d'une liste déroulant. Celle-ci récupère les animaux de l'utilisateur connecté et les affichent. L'utilisateur n'a alors plus qu'à sélectionner la date du poids et le poids en lui-même.

### 2.4.5 Ajouter un nouvel animal

Chaque utilisateur aura la possibilité d'enregistrer de nouveaux animaux. Pour cela il doit se rendre sur le formulaire d'ajout. Il doit alors obligatoirement indiquer le nom de son animal ainsi que sa date de naissance. Il a aussi la possibilité d'indiquer la date de décès, le numéro de la puce, une description et une photo de son animal. Par défaut la race sélectionnée est le chien, cependant l'utilisateur peut la changer grâce à une liste déroulante. Un bouton de réinitialisation est présent si l'utilisateur souhaite vider le formulaire. Une fois que tous les champs obligatoires sont remplis l'utilisateur peut valider.

Le champ utilisé pour le nom de l'animal est sous forme texte. Il en est de même pour le champ utilisé et pour la date de naissance afin d'être cohérent avec la base de données. Le champ de la date de décès quant à lui est sous forme

date. Le champ contenant le numéro des puces sera sous forme numérique. La partie permettant aux utilisateurs de donner une description à leur animal est plus grande afin de permettre aux utilisateurs de rentrer plus d'informations. Quant à la photo elle pourra être importée grâce à un champ prévu à cet effet par le Framework.

#### 2.4.6 Modifier un animal

Chaque utilisateur pourra modifier les caractéristiques de son chat ou de son chien. Pour cela il aura accès au même formulaire que pour l'ajout d'un animal à l'exception que les champs seront déjà remplis. De cette façon l'utilisateur ne modifiera que les champs qu'il estime nécessaire. Les informations seront alors renvoyées au serveur.

#### 2.4.7 Modification d'un utilisateur

Lorsqu'un utilisateur souhaite modifier son mot de passe il le fait depuis son profil utilisateur. L'utilisateur doit entrer son ancien mot de passe afin de s'assurer que la personne se trouvant devant l'écran est bien le propriétaire du compte. Puis il doit entrer le nouveau mot de passe à deux reprises. De cette façon on s'assure qu'il l'a rentré correctement. Lors que l'utilisateur valide une requête "UPDATE" est envoyée. Si un nouveau mot de passe a été inscrit, celui-ci est hashé et le nouveau hash remplace celui présent dans la base de donnée. De plus si l'utilisateur a modifié l'option lui permettant de cacher les dates de décès cette option est également changée. Si l'utilisateur ne remplit aucun champ mais valide tout de même on vérifie que la checkbox n'a pas été modifiée.

#### 2.4.8 Graphiques

Pour afficher l'évolution du poids des animaux de compagnie, des graphiques seront utilisés. Pour cela je vais utiliser un Framework JavaScript.

Après plusieurs tests j'ai opté pour "Chart.js". Il faut récupérer les données de la base de données afin de pouvoir les afficher dans le graphique. Il faut juste les mettre à la forme adéquate. Chart.js est très facile à intégrer à un projet. De plus leur site est bien fait, il est donc facile de trouver de la documentation. Il est également personnalisable facilement.

Chaque animal possédera son propre graphique dans sa page détails. De plus une page à elle seul combinera tous les graphiques afin de ne pas avoir à aller dans toutes les pages détails de chaque animal afin de les visualiser.

#### 2.4.9 Profil utilisateur

L'utilisateur à également la possibilité d'afficher son profil et de changer son mot de passe. Pour cela j'ai créé une page constituée de trois champs à remplir.

Si l'utilisateur souhaite changer son mot de passe il doit inscrire son mot de passe actuel ainsi que deux fois son nouveau mot de passe. Il peut alors "Sauvegarder les modifications". Lors que l'utilisateur appuis sur le bouton les données dans la base de données sont mises à jour. A ne pas oublier que l'utilisateur peut également décider de cacher les dates de décès des animaux de compagnie. Il lui suffit de cocher ou décocher l'option se trouvant au bas de la page.

## 2.5 Stratégie de test

- **Test de l'inscription**

Sur la page de connexion il faut cliquer sur le bouton "Créer un compte" qui redirige sur une page d'inscription. Sur cette page, on y inscrit le prénom et le nom de famille et l'email du nouvel utilisateur. Il faut également définir le mot de passe. On y définit également l'option permettant à l'utilisateur de cacher les dates de décès. Une fois que tous ces champs sont remplis il faut valider grâce au bouton "Valider". L'utilisateur sera alors dirigé vers la page affichant la liste de ses animaux de compagnie.

**Résultat attendu**

Dans la base de données toutes les informations ont été entrées et le mot de passe est inscrit en crypter.

- **Test de connexion**

Pour s'assurer qu'un utilisateur peut se connecter au site il faut mettre en place différents tests.

Le premier consiste à entrer le nom d'utilisateur et le mot de passe d'un utilisateur créé précédemment.

Le mot de passe doit être crypté dans la base de données mais celui inscrit ne doit pas l'être.

**Résultat attendu**

L'utilisateur est redirigé vers la page d'accueil avec le nom de son compte en haut à droite.

- **Test d'ajout de poids**

Sur la page permettant d'ajouter un poids à un animal se trouve une combobox. Il faut s'assurer que dans celle-ci se trouve uniquement le nom des animaux appartenant à l'utilisateur connecté. Une fois la date et le poids sélectionnés on doit Valider.

**Résultat attendu**

Le nouveau poids doit être visible dans le graphique associé à l'animal.

- **Test d'affichage des animaux de l'utilisateur**

Lorsque l'utilisateur se connecte il est redirigé sur la page affichant la liste de ses animaux.

**Résultat attendu**

Pour chaque animal doit être affiché son nom, le dernier poids entré ainsi que sa race. Un bouton permettant à chaque utilisateur de visualiser les détails de l'animal doit être présent.

- **Test d'affichage des graphiques**

Il faut se rendre sur la page d'affichage des graphiques ainsi que sur la page détail d'un animal.

**Résultat attendu**

Sur chaque page détails des animaux, l'évolution de leur poids doit être affiché sous forme de graphique linéaire. Une page doit également contenir tous les graphiques regroupés.

Chaque graphique doit afficher sur axe x la date où le poids a été pris. L'axe Y est le poids du dit animal. Pour s'assurer que tout fonctionne parfaitement il faut se rendre sur les pages susnommés et s'assurer que les graphiques s'affichant dans la page détails soient cohérents avec les données de la base de données. Quant à la page regroupant tous les graphiques, chaque animal doit être identifiable et le graphique doit être lisible.

- **Test d'affichage des détails**

Depuis la page regroupant tous animaux se rendre sur la page des détails grâce au bouton "+".

**Résultat attendu**

S'assurer que les informations affichées soient cohérentes et qu'elles appartiennent bien à l'animal sélectionné. S'assurer également que l'image de l'animal en question s'affiche correctement ainsi que le graphique.

- **Test de la modification du compte utilisateur**

Une fois sur la page des détails de l'utilisateur, il faut inscrire le mot de passe actuel puis le nouveau mot de passe à deux reprises. On coche également l'option "voire le décès des animaux". Une fois la sauvegarde des modifications effectuée se rendre dans les détails d'un animal.

**Résultat attendu**

Il suffit de vérifier que la date de décès de l'animal est cachée ou visible en fonction de ce qui a été décidé précédemment.

- **Test de modification des animaux**

Il faut se rendre sur la page regroupant tous les animaux. Depuis là cliquer sur le bouton pour modifier un animal. Une fois sur la page de modification modifier certaines informations de l'animal puis valider.

**Résultat attendu**

Dans la page détail de l'animal modifié les nouvelles informations doivent être visibles.

- **Test de la fonctionnalité Responsive.**

Cliquer sur F12 puis en sélectionnant l'option d'afficher la visualisation téléphonique. Choisir le format de téléphone iPhone 6/7/8. Il suffit alors de visiter le site d'effectuer les tests précédents.

**Résultat attendu**

Tous les boutons doivent être utilisable et tous toutes les données doivent être lisibles.

- **Test d'ajout d'animal de compagnie**

Se rendre sur le formulaire d'ajout d'un animal. Le remplir avec des informations puis valider.

**Résultat attendu**

Le nouvel animal doit être visible dans la liste de ses animaux, et ses détails doivent être cohérents.



## 2.6 Risques techniques

- Une des risques présents dans ce projet est le fait que je n'ai jamais mis en place de graphiques dans un site web. Je risque donc de perdre du temps face à des problèmes jamais rencontrés.
- La perte de document est également un risque présent. Pour y remédié j'ai mis en place une sauvegarde régulière à la fois en local sur mon ordinateur, sur le H: de l'ETML, mais également en ligne grâce le site web [gitHub](#).

## 2.7 Planification

La planification se trouve en annexe : jdt.xlsm

# 3 Réalisation

## 3.1 Dossier de réalisation

### 3.1.1 Répertoire

Afin de faciliter la gestion des fichiers de mon projet j'ai subdivisé mon projet sous plusieurs dossiers.

Le premier est le dossier materialize. À l'intérieur se trouve tous ce qui concerne l'aspect visuel du site. C'est-à-dire le css est le javascript propre au framework, mais également mes fichiers css.

Le deuxième dossier se nomme Pages. Celui-ci regroupe toutes les pages du site telle que la page d'accueil, l'affichage des animaux de compagnie ou des graphiques.

Le troisième dossier comprend tous les dossier php qui ne sont pas une page d'affichage. Par exemple, les pages gérant la connexion, l'inscription ou alors le hashage des mots de passe. On y trouve également les partis php servants "d'Include" comme les menus ou les footers.

Finalement un dossier image se trouve également dans le répertoire du projet. Celui permet de stocker les différentes images utilisées sur le site. De cette manière la base de données ne se trouve pas trop encombrée car seul le nom des images est stocké à l'intérieur.

Un fichier PHP regroupe toutes les requêtes SQL qui sont utilisées à plusieurs reprises telle que la récupération des données d'un animal de compagnie. Quant aux requêtes que je n'utilise qu'une seule fois, je les exécute directement dans la page PHP nécessaire. De cette manière j'évite de faire appel à la base de données à répétition pour les mêmes informations. Je pourrai y mettre toutes les requêtes cependant si j'opte pour cette option les petites requêtes, ne récupérant qu'une seule donnée, seraient perdues parmi toutes les autres requêtes alors que si je les laisse sur la page concernée mon code est plus lisible.

### 3.1.2 Connexion

Lorsqu'un utilisateur rentre son email et son mot de passe ceux-ci sont envoyés à une page appelée "confirmConexion.php". C'est sur cette page toutes les actions vont être réalisées. En premier lieu on va récupérer les caractéristiques de l'utilisateur en question à l'aide d'une requête SQL.

```
SELECT useFirstName, useEmail, usePassword FROM t_user WHERE useEmail = '$login'
```

Il ne nous reste alors plus qu'à comparer les informations entrées par l'utilisateur avec celles trouvées dans la base de données. Si toutes les informations concordent alors l'utilisateur est connecté. Si l'utilisateur n'est pas trouvé dans la base de données ou que les deux hash ne sont pas équivalents, un message d'erreur apparaît et l'utilisateur est invité à réessayer. Il est important de mettre la ligne de code "session\_start()" au début de chaque page. Car sinon la connexion ne prendra pas effet.

### 3.1.3 Inscription

Lorsqu'un nouvel utilisateur arrive sur le site il peut s'inscrire. Pour cela il doit se rendre sur la page d'inscription accessible depuis la page de connexion. Une fois sur la page d'inscription, l'utilisateur doit alors rentrer son prénom, son nom, son email, l'utilisateur peut choisir de cacher ou d'afficher la date de décès des animaux à l'aide d'une checkbox, ainsi qu'un mot de passe. De la même manière que pour la connexion les données sont envoyées sur la seconde page. Puis on récupère les utilisateurs existant dans la base de données. On s'assure que le mot de passe rentré deux fois a été fait correctement et que l'email de l'utilisateur n'est pas déjà utilisé. Si tous les paramètres sont remplis alors le mot de passe est hashé puis on inscrit l'utilisateur dans la base de données.

```
INSERT INTO t_user (idUser, useFirstName, useLastName, useEmail, usePassword, useHideDeath)  
VALUES (NULL, '$firstName', '$lastName', '$Email', '$pwdHash', '$hideDeath')
```

Il ne reste alors plus qu'à connecter l'utilisateur et à le rediriger vers la page affichants ses animaux de compagnies.

### 3.1.4 Affichage des animaux de compagnie

Chaque utilisateur a la possibilité de visualiser la liste de tous ses animaux de compagnie. Pour cela j'ai utilisé deux requêtes SQL. La première requête récupère la race ainsi que le nom de l'animal sur deux tables différentes. La deuxième va être exécutée dans une boucle parcourant le résultat de la première commande. De cette manière elle ne va récupérer que le dernier poids de chaque animal.

```
foreach ($getdata as $line) {  
    $PetID = $line["idPet"];  
    echo "<tr><td>" . $line["petName"] . "</td>";  
    echo "<td>";  
        $getWeight = $allRequest->getWeightByPet($line["idPet"]);  
        $getWeightData = $exeRequest->executeQuerySelect($getWeight);  
        if($getWeightData[0]["weiWeight"] != "") {  
            echo $getWeightData[0]["weiWeight"];  
        }  
        else{  
            echo "N/A";  
        }  
}
```

Le bout de code ci-dessus permet de lister tous les résultats de la première requête. Puis les lignes 5 et 6 exécutent la seconde requête à chaque itération afin de récupérer tous les poids d'un animal. Puis j'y affiche le dernier poids.

### 3.1.5 Affichage des graphiques

Dans la page détails de chaque animal se trouve un graphique affichant l'évolution du poids de celui-ci.

On exécute une requête récupérant le nom de l'animal concerné ainsi que ses différents poids et la date d'entrée de celui-ci.

```
SELECT weiWeight,weiDate,petName,t_weightpet.idPet  
FROM t_weightpet  
LEFT JOIN t_pet ON t_weightpet.idPet = t_pet.idPet  
WHERE t_pet.idPet = $petId
```

Une fois les données récupérées, il faut les mettre en forme afin que les guillemets soient placés au bon endroit. Pour cela il faut utiliser for qui parcourt chaque donnée et mettre un guillemet après chaque donnée excepté la dernière. Une fois que c'est fait il faut créer le graphique en lui-même à l'aide de la balise "<canvas>". Par la suite dans les balises "<script>" on configure le graphique en lui-même, comme ci-dessous.

```

<canvas style="..." id="myChart"></canvas>
<script>
    var ctx = document.getElementById('myChart');
    var myChart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: [<?php echo $dates; ?>],
            datasets: [{
                //First Data
                label: '<?php echo $getgraficData[0]['petName'];?>',
                data: [<?php echo $allPetWeight; ?>],
                borderColor: 'rgb(0,0,0)',
                pointBackgroundColor: 'rgb(0,0,0)',
                backgroundColor: 'transparent',
            },]
        },
        options: {
            responsive: true,
            scales: {
                yAxes: [{
                    ticks: {
                        min: 0,
                        max: 10,
                        stepSize: 1
                    }
                }]
            }
        }
    });
</script>

```

On y définit la couleur, le type de graphique, dans notre cas de type line, ou encore quelles données afficher.

Cette méthode fonctionne pour la page détails de chaque animal. Pour la page regroupant tous les graphiques je m'y suis pris différemment. J'y ai créé une boucle passant en revue tous les animaux appartenant à l'utilisateur connecté. A chaque passage dans la boucle j'y exécute une requête afin de récupérer uniquement les données du bon animal. Puis j'affiche le graphique avec les bonnes valeurs.

### 3.1.6 Ajouter un animal de compagnie

Comme précisé dans la partie analyse du rapport, l'ajout d'un animal se fait à travers un formulaire. Une fois que l'utilisateur a validé les données sont récupérées grâce à la méthode POST puis récupérée dans une autre page qui quant à elle exécute une requête SQL "INSERT". Lors de la mise en place de cette fonctionnalité j'ai eu quelques soucis. Lors de l'exécution de la requête l'erreur suivante apparaissait.

```
#1292 - Incorrect date value: '' for column 'petDeath' at row 1
```

Après un certain temps de recherche et avec l'aide de mon collègue Jonathan Mayor j'ai identifié le problème. Celui-ci vient du fait que si l'utilisateur n'inscrit aucune donnée dans le champ censé récupérer la date de décès la requête ne sait pas quoi mettre dans la base de données et retourne donc une erreur. Pour contourner ce problème, lors de la récupération des données, il faut vérifier si les champs de la date de décès et du numéro de puce sont vides. Si c'est le cas alors on remplit la valeur avec la valeur "NULL". De cette façon s'il n'y a pas de valeur la base de données est configurée à NULL. Chaque animal doit également avoir la possibilité de contenir une image. Pour cela un champ de type file est présent dans le formulaire. Le nom du fichier est récupéré en POST puis stocké dans une variable. Dans une seconde variable nommée \$dest on y combine le futur emplacement de l'image à son nom. L'image quant à elle, est enregistrée dans un dossier temporaire. Finalement grâce à la méthode : move\_uploaded\_file() on déplace l'image dans le dossier définitif.

### 3.1.7 Ajouter un nouveau poids

Chaque utilisateur a la possibilité d'ajouter un poids à son animal de compagnie. Une page contient un formulaire composé d'une combobox contenant le nom de tous les animaux appartenant à l'utilisateur, d'un champ permettant d'entrer la date et d'un champ afin d'y indiquer le poids de l'animal de compagnie.

Lors de la validation les données sont envoyées sur une page afin de la valider. Une commande "INSERT" est exécutée puis l'utilisateur est redirigé sur la page de l'animal à qui il a ajouté une donnée. De cette façon il peut s'assurer que tout s'est déroulé comme prévu.

### 3.1.8 Modifier un utilisateur

La page du profil utilisateur est composée de toutes les infos de l'utilisateur telles que son Email, son prénom et son nom de famille. Sur la même page se trouvent trois champs qui permettent à l'utilisateur de changer son mot de passe. Lorsque l'utilisateur valide la modification, les informations sont envoyées grâce à la méthode POST à une page annexe.

Sur cette page en tout premier lieu on vérifie que le mot de passe entré par l'utilisateur correspond à celui se trouvant dans la base de données. Pour cela on utilise la même méthode que lors de la connexion. Une fois qu'on s'est assuré que le mot de passe est correct on s'assure que l'utilisateur a rentré le même mot de passe dans les deux champs restants. Si tous les paramètres sont corrects le hash du mot de passe est remplacé dans la base de données grâce à la commande suivante.

```
UPDATE t_user
SET usePassword = '$pwdHash', useHideDeath = '$HideDeath'
WHERE t_user.useFirstName = '$_SESSION[Pseudo]'
```

Si tous les champs sont laissés vides, la même commande est exécutée à l'exception que le mot de passe n'est pas modifié mais uniquement le champ permettant de cacher les morts. Pour cela lors de la récupération des données du formulaire on vérifie

si la checkbox était cochée grâce à un "if(isset)". Si c'est le cas une variable est défini à 0.

### 3.1.9 Supprimer un animal

En accédant aux détails d'un animal, son propriétaire doit être capable de le supprimer. Pour cela tout au bas de la page se trouve un bouton de suppression. Lorsque celui-ci est sélectionné un message de validation apparaît puis l'animal en question est supprimé.

### 3.1.10 Modifier un animal de compagnie

Sur la page regroupant tous les utilisateurs se trouve un bouton permettant de modifier l'animal en question. L'utilisateur est redirigé sur la page permettant d'ajouter un animal. Cependant j'y envoie l'information qu'il s'agit d'une modification grâce à la méthode GET. On y définit l'id de l'animal à modifier ainsi que le fait que l'on souhaite le modifier. Le formulaire d'ajout est alors rempli automatiquement par les informations de l'animal de compagnie. Lors de la validation les informations sont envoyées de la même manière que pour l'ajout. Cependant s'il s'agit d'une modification la requête exécutée n'est pas un input mais bien une update.

Ce projet a été réalisé à l'aide de windows 7. Tous mes tests en été fait grâce au navigateur Google Chrome. J'ai utilisé l'application phpStrome 2017.1.4 pour coder. Ma base de donn   a   t   g  r  e gr  ce    l'outil EasyPhp et PhpMyAdmin 4.7.0

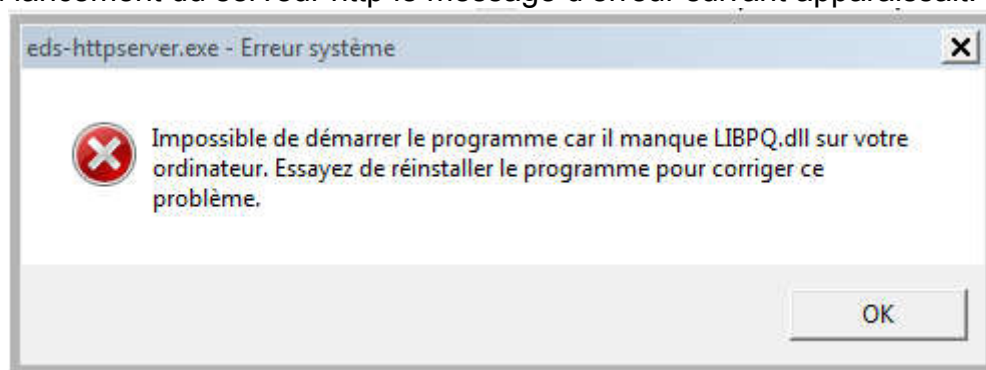
### Utilisateurs de la base de donn  e

Email de connexion	Mot de passe
samuel.dadie@gmail.com	Samuel
premier.testeur@gmail.com	qwertz
cindy.hardegger@vd.educanet2.ch	.Password-
alain.roy@vd.oai.ch	.Password-
nicolas.glassey@cpnv.ch	.Password-

## 3.2 Probl  mes rencontr  s

### Lancement du serveur HTTP

Lors du lancement du serveur http le message d'erreur suivant apparaissait.



*Erreur lors du lancement du serveur*

Cependant tout fonctionne parfaitement. Je n'ai pas remédié à ce problème pour le moment. Apparemment ce problème viendrait du ma version de EasyPhp.

#### Problème d'affichage des poids

Lors de l'affichage de de tous les animaux la race et le nom s'affichent correctement, cependant le poids affiché n'est pas le dernier rentré ou même le dernier au niveau de la date. En tout premier lieu tous les poids étaient affichés ce qui avait comme conséquence qu'un animal apparaissait autant de fois qu'il possédait de poids différents. Pour cela j'ai intégré le paramètre "Group by" dans ma commande SQL. Pour que cette commande fonctionne, toutes les relations "INNER JOIN" doivent être transformées en "LEFT JOIN". Malheureusement bien que chaque animal n'apparaissait qu'une seule fois le poids affiché n'est toujours pas le dernier rentré. Pour résoudre ce problème j'ai ajouté le paramètre Distinct pour qu'il prenne uniquement les éléments différents et j'ai enlevé l'option Group By, sans succès.

```
SELECT weiDate,weiWeight,petName,typName,t_pet.idPet
FROM   t_pet
LEFT JOIN t_weightpet ON t_weightpet.idPet = t_pet.idPet
LEFT JOIN t_typepet ON t_typepet.idPetType = t_pet.idPetType
LEFT JOIN t_user ON t_pet.idUser = t_user.idUser
WHERE  t_user.useFirstName = '$idUser'
GROUP BY(petName)
ORDER By idPetWeight DESC
```

Finalement avec l'aide de Mme Hardegger j'ai suis venu à la conclusion qu'il serait bien plus simple de réaliser cette tâche avec deux requêtes différentes. La première commande sélectionne toutes les informations excepté le poids des animaux.

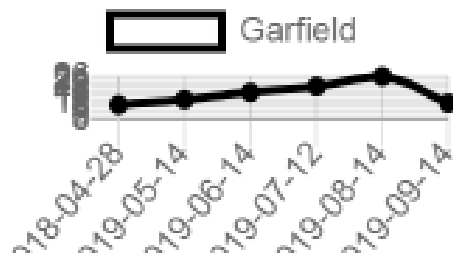
```
SELECT petName, typName, idPet, t_user.idUser
FROM t_pet, t_typepet, t_user
WHERE t_pet.idPetType = t_typepet.idPetType
AND t_user.idUser = t_pet.idUser
AND useFirstName = '$idUser'
```

La deuxième requête quant à elle récupère uniquement le dernier poids de l'animaux demandé. Il ne me reste alors plus qu'à parcourir tous les animaux récupérés et d'y exécuter la deuxième requête afin d'obtenir que le dernier poids. De cette façon j'ai toutes les informations dont j'ai besoin, il ne me reste plus qu'à les afficher.

#### Problème d'affichage de graphiques

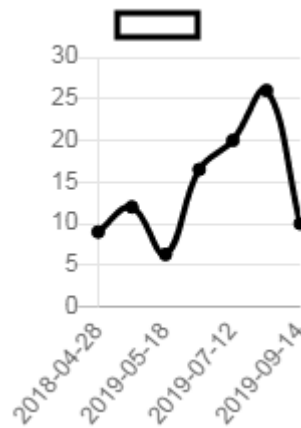
Lors de l'affichage des graphiques tout fonctionnait bien temps que le site était sous forma ordinateur. Mais dès que celui était mis au format téléphone les graphiques devenaient illisibles.





Affichage des graphiques au format téléphone

Pour résoudre ce problème j'ai mis le canevas du script dans une div. Cette div est configurée dans un css qui surclasse le css de Materialize. Il ne restait alors plus qu'à définir la taille des graphiques. Cette solution a été mise en œuvre à la fois sur la page détaillant les animaux mais également sur la page contenant tous les graphiques.



Graphique lisible

Les données du graphique sont donc lisibles. Cependant si le graphique est composé de beaucoup d'éléments il n'apparaîtront pas sur l'axe x et pour les visualiser il faut cliquer sur le graphique en lui-même.

#### Modification d'un animal

Lors de la modification d'un animal, le formulaire est déjà rempli par les données de l'animal de compagnie à modifier. Cependant le nom de l'image n'est pas récupéré. Ce qui fait que dans la base de données le nom de l'image est remplacé par une valeur vide. Pour cela dans la balise "<input>" j'y ai ajouté un nom et un id. Avec cette méthode je peux récupérer le nom du fichier grâce à la méthode POST plutôt que la méthode FILE. Il ne me reste plus qu'à indiquer le nom de l'image dans le formulaire et le tour est joué.

### **3.3 Description des tests effectués**

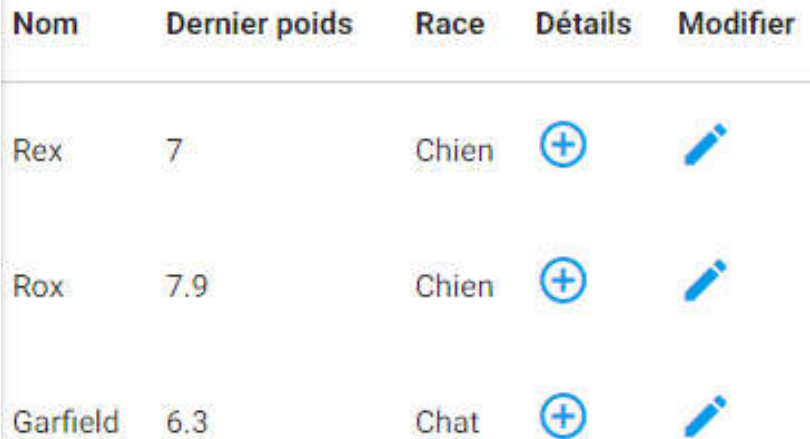
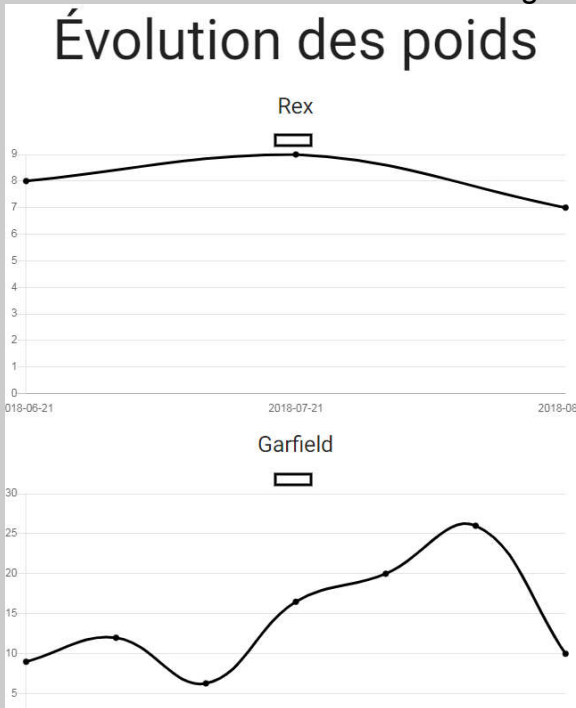
Tous les tests ont été sur le navigateur web Google Chrome en utilisant EasyPhp.

Nom du test	Déroulement du test	Résultat
Test de l'inscription	Sur la page d'inscription le champ prénom, Email ainsi que les mots de passe sont obligatoires. Si on entre un email déjà utilisé un message apparaît pour l'indiquer. Si	



	<p>le mot de passe n'est pas identique dans les deux champs l'utilisateur en est également informé.</p> <div><h1>Pet Weight</h1><h2>Inscription</h2><div><div>Prénom</div><div>Nom de famille</div></div><div><div>*Email</div></div><div><div>*Mot de passe</div><div>*Mot de passe</div></div><div><input type="checkbox"/> Voir le décès des animaux</div><div>VALIDER</div><div>Email déjà existant</div><div>Connexion</div></div>	
<b>Test de connexion</b>	<p>Une fois sur la page de connexion, j'y entre mon email ainsi que mon mot de passe. Si l'un d'entre eux est erroné alors un message l'indiquant apparaît. Si toutes les informations sont correctes l'utilisateur est redirigé vers la page regroupant ses animaux et son nom d'utilisateur apparaît en haut à droite de l'écran.</p>	

	<div><h1>Pet Weight</h1><h2>Pet Weight</h2><div><input type="text" value="samuel.dadie@gmail.com"/></div><div><input type="password" value="....."/></div><div>Email ou mot de passe erroné</div><div>Mot de passe oublié ?</div><div>CONNEXION</div><div>Créer un compte</div></div>
<b>Test d'ajout de poids</b>	<p>Sur la page d'ajout j'y sélectionne l'animal à qui je souhaite ajouter un poids. Dans le deuxième champ j'y inscrit la date où la date a été ajoutée. Dans le troisième champ j'y inscrit un chiffre à virgule. Puis je valide. Je me retrouve alors redirigé vers la page détails de l'animal à qui je viens d'ajouter un poids. Sur le graphique je peux visualiser ce dernier.</p> <div><div>Entrer un nouveau poids</div><div><div>Rex</div><div>jj.mm.aaaa</div><div>Poids</div><div>AJOUTER</div></div></div>
<b>Test d'affichage des animaux de l'utilisateur</b>	<p>La page d'accueil est la page regroupant tous les animaux de l'utilisateur. Je peux y visualiser le nom, le dernier poids, ainsi que sa race. Chaque animal comporte également un bouton détails et un bouton modifier. Seul les animaux de l'utilisateur connecté sont visibles. J'ai également modifié</p>

	<p>l'option pour l'affichage des animaux décédés. Une fois revenu sur la page d'affichage des animaux, les animaux comportant une date de décès ne sont plus visibles.</p> 	
<b>Test d'affichage des graphiques</b>	<p>Sur la page regroupant les graphiques, on y trouve les graphiques de tous les animaux appartenant à l'utilisateur connecté. Ils sont tous alignés verticalement.</p> 	
<b>Test d'affichage des détails</b>	<p>Lorsque l'on clique sur le bouton détails d'un animal on est redirigé vers sa page détails. On y trouve alors tous ses détails (nom, date de naissance et de mort, numéro de puce, dernier poids ajouté et description) ainsi qu'une photo de l'animal et le graphique de l'évolution de son poids.</p>	

Garfield

2018-03-05

Numéro de puce :

Dernier poids : 6.3

Date	Poids
2018-04-28	10
2019-05-14	12
2019-06-14	16
2019-07-12	20
2019-08-14	26
2019-09-14	10
2019-05-18	6

SUPPRIMER L'ANIMAL

## Test de la modification du compte utilisateur

Sur la page utilisateur je rentre le mot de passe actuel ainsi qu'un nouveau mot de passe à deux reprises. Je décoche également l'option "Voir" les animaux décédés" et je valide. Un message apparaît alors afin d'indiquer à l'utilisateur que les modifications ont été effectuées. Si le mot de passe est erroné un message apparaît afin de l'indiquer à l'utilisateur.

	<p>samuel.dadie@gmail.com</p> <p>Prénom : Samuel</p> <p>Nom de famille : Dadié</p> <p><b>Changer de mot de passe</b></p> <p>Mot de passe actuel: _____</p> <p>Nouveau mot de passe: _____</p> <p>Confirmer le mode de passe: _____</p> <p><input checked="" type="checkbox"/> Voir les animaux décédé</p> <p><a href="#">Sauvegarder les modifications</a></p> <p>Mot de passe incorrecte</p>	
<b>Test d'ajout d'un animal de compagnie</b>	J'ai complété le formulaire permettant d'ajouter un animal. Si un champ obligatoire n'est pas rempli un pop-up indique à l'utilisateur que celui-ci doit être rempli. Suite à la validation, le nouvel animal se retrouve dans la liste des animaux.	

	<div> <div>Test</div> <div>* Date de naissance 29.05.2019</div> <div>Date de décès jj.mm.aaaa</div> <div>Numéro de puce</div> <div>Description du test</div> <div>PHOTO grumpy_cat.jpg</div> <div>Chat</div> <div>RÉINITIALISER VALIDER &gt;</div> <div>Test N/A Chat + ✎</div> </div>	
<b>Suppression d'un animal</b>	Dans la page détails de chaque animal se trouve un bouton permettant sa suppression. Lorsque l'on clique dessus l'animal est supprimé et on est redirigé vers la liste de ses animaux.	
<b>Test de modification des animaux</b>	Une fois sur la page de modification on y voit le même formulaire que lors de l'ajout d'un animal. A l'exception prêt que tous les champs sont déjà remplis. J'y ai modifié tous les champs. Je me retrouve alors redirigé sur la liste des animaux. En allant dans les détails je peux constater les changements effectués.	
<b>Test de la déconnexion</b>	En cliquant sur le bouton "Déconnexion" en haut à droite de la page je suis déconnecté et me retrouve sur la page de connexion	
<b>Test du responsive</b>	En se rendant sur le site web sous format téléphone (iPhone6/7/8) toutes les fonctionnalités du site fonctionnent à l'exception du formulaire d'ajout et de modification. Les champs à remplir ne sont pas visible.	

### 3.4 Erreurs restantes

*S'il reste encore des erreurs:*

- Description détaillée
- Conséquences sur l'utilisation du produit

### 3.5 Liste des documents fournis

Rapport de TPI  
Journal de travail comprenant également la planification  
Dossier de projet

## 4 Conclusions

Tâches	Commentaires	Résultats
Pouvoir visualiser le détail de chaque animal	Toutes les informations sont visibles	
Pouvoir ajouter un poids à un animal	L'ajout est fonctionnel et l'utilisateur ne peut rentrer que des chiffres (chiffre à virgule compris)	
Pouvoir visualiser tous les graphiques sur une seule page	Tous les graphiques sont visibles sur une seule page. J'ai eu des difficultés à les rendre visible en format téléphone.	
Pouvoir changer les paramètres utilisateur	Depuis la page utilisateur il est possible de changer son mot de passe mais également de permettre à un utilisateur de cacher les animaux décédés	
Pouvoir ajouter un animal	Un formulaire est disponible pour l'ajout d'un animal. On peut également y ajouter une image.	
Pouvoir modifier un animal	La modification se fait sous le même format que pour l'ajout. Cependant j'ai eu quelque difficultés pour faire apparaître l'image dans le formulaire et pouvoir la modifier.	
Gestion du compte utilisateur	La connexion, l'inscription, et la modification du compte utilisateur est fonctionnel	

### 4.1 Bilan personnel

Dans l'ensemble je suis satisfait du projet. Même si le déroulement du projet n'est pas très proche de ma planification son avancement s'est fait de façon régulière avec quasiment aucun retard. Je pense que la méthode Gantt était la plus appropriée. Je suis un peu déçu de ne pas avoir pu éliminer toutes les exceptions par manque de temps.

### 4.2 Améliorations possibles

Par la suite les points suivants de ce projet pourraient être améliorés.

La fonctionnalité responsive est fonctionnelle, cependant elle pourrait être améliorée. Certaines pages ne sont pas très agréables visuellement.

La fonctionnalité d'oubli de mot de passe pourrait être mis en place. Chaque utilisateur possède un email. Je pourrai donc implémenter la fonction permettant d'envoyer un mail de réinitialisation du mot de passe.

## 5 Annexes

### 5.1 Glossaire

- MCD

Un modèle conceptuel de données est un schéma composé des différentes tables lié entre elles. Elle permet de visualiser la méthode qui est utilisée dans la base de données pour stocker les données

- MLD

Un modèle logique des données est l'évolution du MCD. Cette version est un peu plus technique. Elle comprend les ..... qui permettent de définir la façon dont les éléments seront lié l'un à l'autre.

- MPD

Le Modèle physique des données est la forme finale du MCD. Elle se présente sous forme de base de données généralement en SQL. C'est la version que nous exploitons dans un site web.

- Framework

Ensemble d'éléments permettant de gérer toute une partie d'un logiciel. Dans mon projet cela se présente sous la forme de fichiers Javascript.

- Librairie graphique

Librairie d'éléments servant à gérer la partie graphique d'un site web par exemple.

### 5.2 Résumé du rapport du TPI / version succincte de la documentation

- **Situation de départ**

Le but de ce projet est de mettre en place un site web permettant d'y inscrire ses animaux de compagnies et de l'évolution de leur poids à l'aide d'un graphique. Pour cela lors de l'arrivée sur le site il est automatiquement demandé de se connecter. Dans la base de données les mots de passes sont hashés grâce à la



méthode "password\_hash()". On peut alors y inscrire un animal de compagnie et y entrer la dernière pesée de celui-ci.

Chaque utilisateur a accès à une page utilisateur où il peut modifier ses informations personnelles. Une page de résumé est également présente permettant de visualiser tous les graphiques de ses animaux de compagnies. Il peut visualiser tous les détails de ses animaux et les modifier à volonté. Chaque animal est constitué d'un nom et d'une date de naissance. De façon facultatif il peut également posséder un numéro de puce, une image, des poids et une date de décès.

Pour mettre en place ce projet j'ai utilisé deux framework. Le premier, Materialize met permet de gérer l'aspect visuel ainsi que de faciliter certaines fonctionnalités en javascript.

Le second, Chart.js quant à lui me permet de gérer tous les graphiques du site. Pour le mettre en place j'ai utilisé PhpStorm 2017.1.4 ainsi que PhpMyAdmin. Les tests se sont déroulés sur le navigateur web Google Chrome.

- **Mise en œuvre**

Lors de la mise en place du projet j'ai eu quelques difficultés avec l'affichage des animaux de compagnies. Pour récupérer les données dans la base de données j'ai tenté de le faire à l'aide d'une seule requête. Cependant en le faisant de la sorte je ne pouvais pas récupérer les données dans 2 tables différentes tout en ne récupérant que la dernière entrée d'une autre table. J'ai donc opté pour le faire en deux requêtes.

Pour m'assurer que le site est bien fonctionnel j'ai effectué toute une série de tests. Je me suis inscrit avec un nouvel email. Une fois connecté j'ai ajouté puis supprimer un animal. J'ai également ajouté plusieurs nouveaux poids à un animal. Finalement j'ai visité chaque page du site afin de m'assurer qu'elles étaient toutes accessibles et lisibles.

- **Résultats.**

A la fin de ce projet tout fonctionne. Cependant certaines exceptions sont toujours présentes. Je pourrais continuer ce projet en ajoutant la validation du compte par l'envoi d'un email afin de s'assurer que l'email utilisé existe. Ou la réinitialisation du mot de passe grâce à l'envoi d'un email.

### 5.3 Sources

<https://materializecss.com>

<https://www.chartjs.org>

<https://stackoverflow.com>

Jonathan Mayor

Grégory Poget

Cindy Hardegger

### 5.4 Journal de travail

Date	Durée	Activité	Remarques

## 5.5 Manuel d'Installation

## 5.6 Manuel d'Utilisation

Lors de votre arrivée sur le site vous vous retrouvez devant une page de connexion. Si vous n'avez pas de compte cliquez sur "Créer un compte". Vous vous retrouvez alors sur la page d'inscription. Tous les champs sont obligatoires à l'exception du nom de famille. Une fois inscrit vous êtes connecté et redirigé sur la page regroupant vos animaux. Pour en ajouter il faut cliquer sur le bouton vert. Vous vous retrouvez alors sur un formulaire où vous pouvez y entrer toutes les informations de l'animal. Les champs comprenant une étoile sont obligatoires. Une fois que vos animaux sont ajoutés au site vous pouvez leur ajouter un poids en allant sur la page "Nouveau poids" grâce au menu se trouvant en haut. Une fois le poids ajouté vous êtes redirigé sur la page de l'animal en question. Cette page est également accessible depuis la liste des animaux grâce au logo détails. Le logo "modifier" quant à lui vous permet de modifier un animal. Il est également possible de visualiser l'ensemble des graphiques grâce à la page "Graphique".

## 5.7 Archives du projet

*Media, ... dans une fourre en plastique*