

# **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	Concept .....	5
2.3.1	MCD .....	5
2.3.2	MLD .....	7
2.3.3	MPD.....	7
2.3.4	Framework.....	8
2.3.5	Maquettes.....	8
2.4	JavaScript / SQL.....	11
2.4.1	Connexion à la base de donnée. ....	11
2.4.2	Connexion/Déconnexion.....	12
2.4.3	Lister les animaux de compagnies.....	12
2.4.4	Ajouter un nouveau poids .....	12
2.4.5	Ajouter un nouvel animal .....	12
2.4.6	Modifier un animal .....	12
2.4.7	Modification d'un utilisateur .....	13
2.4.8	Graphiques .....	13
2.4.9	Profil utilisateur .....	13
2.5	Stratégie de test.....	13
2.6	Risques techniques .....	15
2.7	Planification .....	15
2.8	Dossier de conception .....	15
3	Réalisation.....	16
3.1	Dossier de réalisation .....	16
3.1.1	Répertoire.....	16
3.1.2	Connexion .....	16
3.1.3	Inscription .....	17
3.1.4	Affichage des animaux de compagnies .....	17
3.1.5	Affichage des graphiques .....	17
3.1.6	Ajouter un animal de compagnie .....	19
3.1.7	Ajouter un nouveau poids .....	19
3.1.8	Modifier un utilisateur.....	19
3.2	Utilisateurs de la base de donnée .....	20
3.3	Problèmes rencontrés.....	20
3.4	Description des tests effectués.....	22
3.5	Erreurs restantes .....	22
3.6	Liste des documents fournis .....	22
4	Conclusions .....	23

---

5	Annexes.....	23
5.1	Résumé du rapport du TPI / version succincte de la documentation .....	23
5.2	Sources – Bibliographie.....	23
5.3	Journal de travail .....	23
5.4	Manuel d'Installation .....	23
5.5	Manuel d'Utilisation.....	23
5.6	Archives du projet.....	24

**NOTE L'INTENTION DES UTILISATEURS DE CE CANEVAS:**

*Toutes les parties en italiques sont là pour aider à comprendre ce qu'il faut mettre dans cette partie du document. Elles n'ont donc aucune raison d'être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n'aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l'alourdir inutilement.*

## **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 (1page). 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 fournis 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

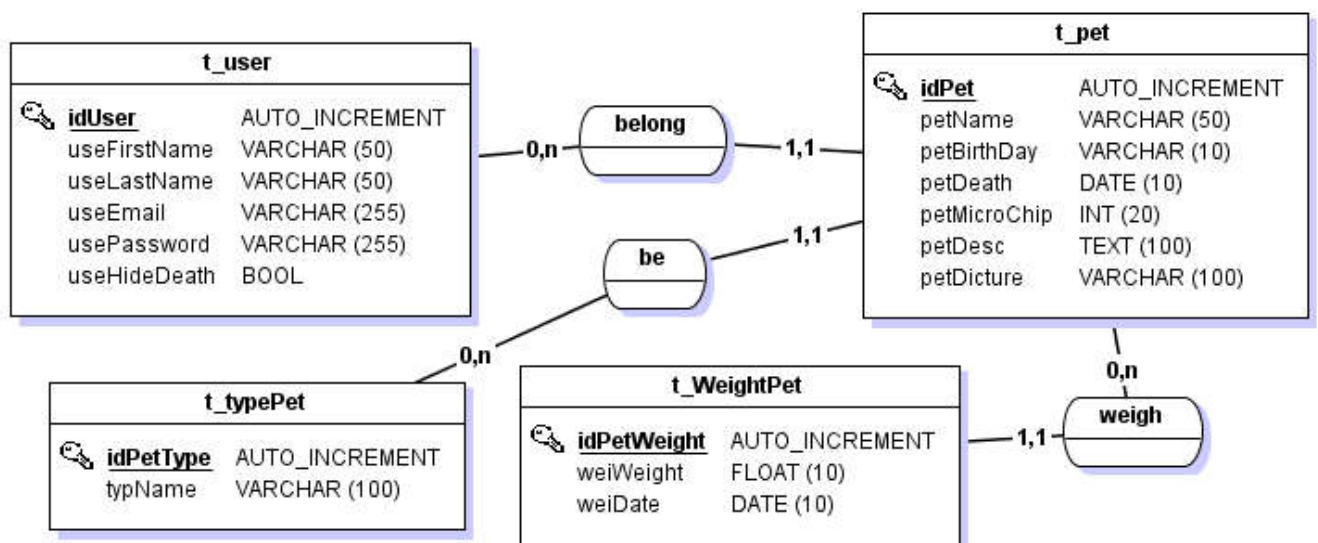
Pour coder la suite j'ai utilisé PhpStorm à la version 2017.1.4

### 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'aurais divisé mais projet en plusieurs parties. Hors, un projet de 89 heures est suffisamment court pour que le projet soit fait d'une traite.

### 2.3 Concept

#### 2.3.1 MCD

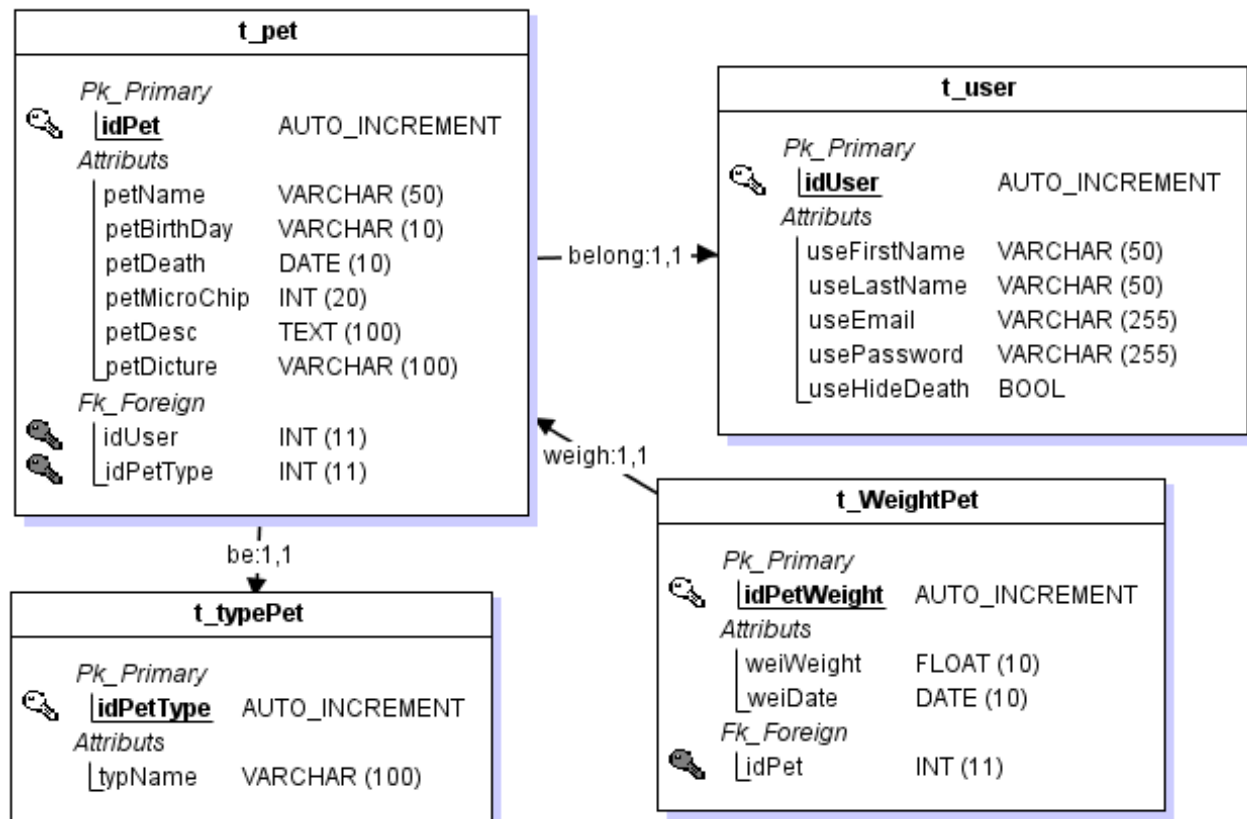


Dans la mise en place du site web il est nécessaire de mettre en place une base de donnée. Pour ce faire je commence par créer un MCD (modèle conceptuel de données).

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 visible.
- **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



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 donnée 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 passes seront hashé 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 MPD

### 2.3.4 Framework

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

Pour ce projet je vais utiliser le Framework nommé Materialize.

J'ai choisis 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 tel que le responsive.

### 2.3.5 Maquettes

Le site sera constitué de ... pages.

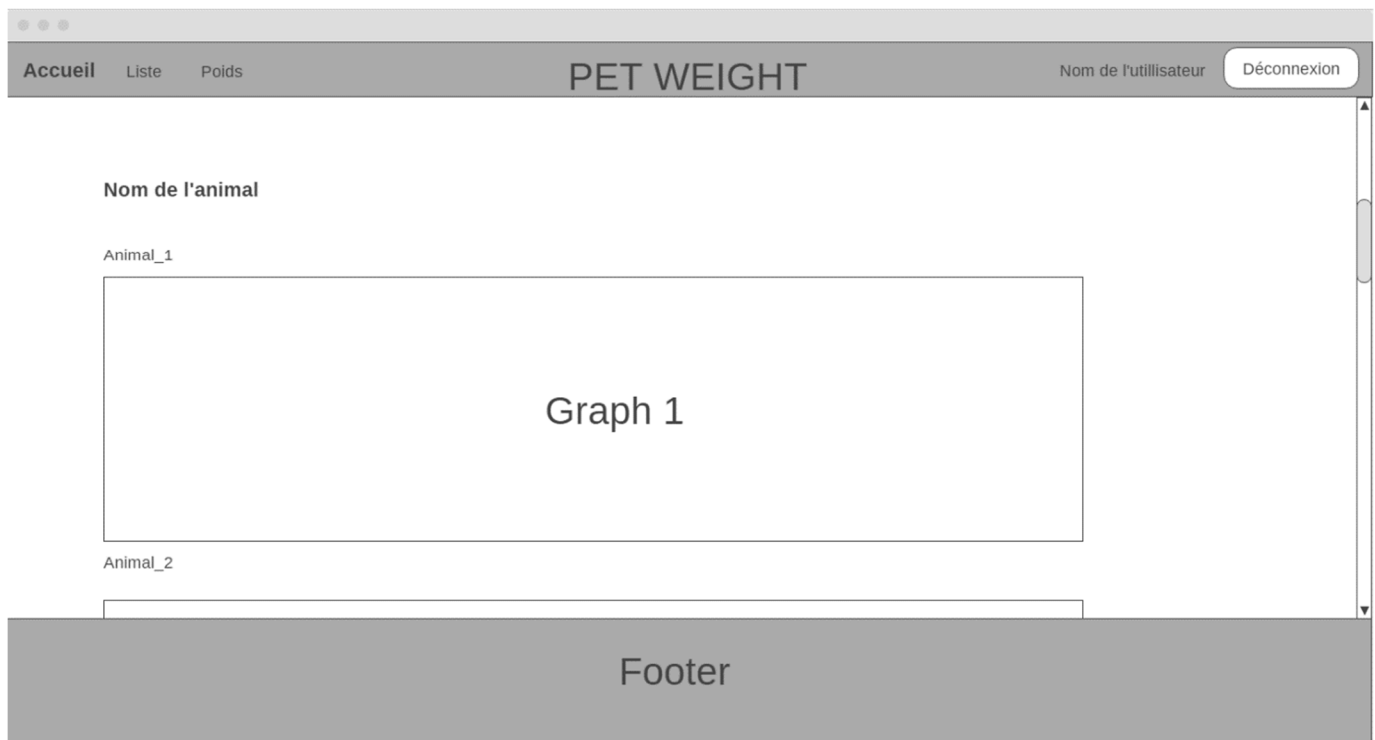
- Page de connexion

La maquette représente une page de connexion centrée. Elle a un titre principal 'PET WEIGHT' en haut, suivi d'un sous-titre 'Connexion'. En dessous, il y a deux champs de saisie : 'Email' et 'Mot de passe'. Une lien 'Mot de passe oublié ?' est placé sous le champ de mot de passe. Au bas de la section, il y a deux boutons : 'Créer un compte' et '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.

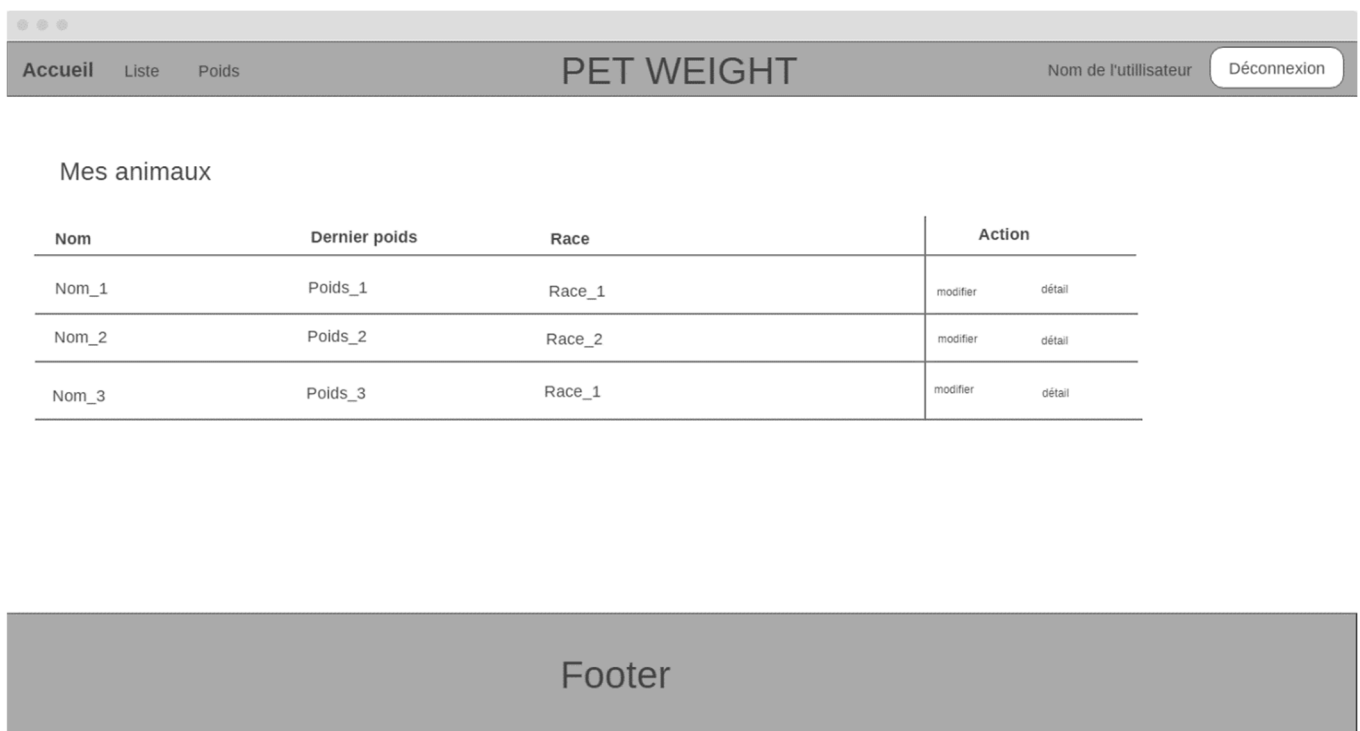
- Page d'inscription
- Page des graphiques affichants les poids





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

- Page listant ses animaux de compagnies



Cette page permet d'afficher ses animaux de compagnies. Ainsi que d'accéder aux détails de ceux-ci.

- Page affichant les détails d'un animal

Accueil Liste Poids

PET WEIGHT

Nom de l'utilisateur Déconnexion

Nom de l'animal

Date de naissance / age - Date de décès

Numéro de puce

Dernier poids

Description

Footer

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

Accueil Liste Poids

PET WEIGHT

Nom de l'utilisateur Déconnexion

Mes animaux

Entrer un nouveau poids

Animal

Date

Poids

Ajouter

Footer

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

Accueil Liste Poids PET WEIGHT Nom de l'utilisateur Déconnexion

Nom de l'utilisateur

Prénom

Nom de famille

Changer de mot de passe

Mot de passe actuel

Nouveau mot de passe

Configurer nouveau mot de passe

☐ Voir le décès des animaux

Sauvegarder la configuration

Footer

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

*Le concept complet avec toutes ses annexes:*

*Par exemple :*

- *Multimédia: carte de site, maquettes papier, story board préliminaire, ...*
- *Bases de données: interfaces graphiques, modèle conceptuel.*
- *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle...*
- ...

## 2.4 JavaScript / SQL

### 2.4.1 Connexion à la base de donnée.

Pour que le site web puisse récupérer les données de façon dynamique il doit être connecté à la base de donnée. 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 donnée il n'a pas accès aux mots de passes.

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.

Lors que 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 compagnies

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 et 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 pour la date de naissance afin d'être cohérent avec la base de donnée. 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 utilisateur de donner une description à leur animal est plus grand afin de permettre aux utilisateur 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. 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 remplis 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 utilisateurs 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 détail. De plus une page à elle seul combinera tous les graphiques afin de ne pas avoir à aller dans toutes les pages détail 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 un page constitué 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 donnée sont mises à jours. A ne pas oublier que l'utilisateur peut également décider de cacher les dates de décès des animaux de compagnies. 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 compagnies.

Il faut s'assurer que dans la base de donnée toutes les informations ont été entrées et que 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 donnée mais celui inscrit ne doit pas l'être.

- **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. Il faut alors aller voir les graphiques et s'assurer que le nouveau poids a été ajouté au bon animal.

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

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

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

Chaque graphique doit afficher sur l'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 susnommées et s'assurer que les graphiques s'affichent dans la page détail 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**

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 "voir le décès des animaux". Une fois la sauvegarde des modifications effectuées se rendre dans les détails d'un animal. 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 modification**

- Test de la fonctionnalité Responsive.
- Test d'ajout

## 2.6 Risques techniques

- *risques techniques (complexité, manque de compétences, ...).*

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, ...).*

## 2.7 Planification

*Révision de la planification initiale du projet :*

- *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
- *partage des tâches en cas de travail à plusieurs.*

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

## 2.8 Dossier de conception

*Fournir tous les documents de conception:*

- *le choix du matériel HW*
- *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
- *le choix des outils logiciels pour la réalisation et l'utilisation*
- *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...*
- *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
- *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

### 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 tel que la page d'accueil l'affichage des animaux de compagnie ou les graphiques. Le troisième dossier, lui comprend toutes les dossier php qui ne sont pas une page qui a pour but de s'afficher. 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 donnée ne se trouve pas trop encombré car seul le nom des images est stocké à l'intérieur.

Un fichier PHP regroupe toutes les requêtes SQL qui sont utilisée à plusieurs reprises tel 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 donnée 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 seul donné serait perdu 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 envoyé à une page appelée "confirmConexion.php". C'est sur cette page toute les actions vont être réalisée. En premier lieux on va récupérer les caractéristiques l'utilisateur en question à l'aide d'une requêtes SQL.

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

Il ne nous reste alors plus qu'à comparer les informations entrées par l'utilisateurs avec celles trouvée dans la base de donnée. Si toutes les informations concordent alors l'utilisateur est connecté. Si l'utilisateur n'est pas trouvé dans la base de donnée ou que les deux hash ne sont pas équivalent un message d'erreur apparaît et l'utilisateur est invité à réessayé. 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 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 envoyés sur second page. Puis on récupère les utilisateurs existant dans la base de donnée. 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 donnée.

```
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 affichant ses animaux de compagnies.

### 3.1.4 Affichage des animaux de compagnies

Chaque utilisateur a la possibilité de visualiser la liste de tous ses animaux de compagnies. 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";
        }
}
```

### 3.1.5 Affichage des graphiques

Dans la page détail 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 maîtriser en formes 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 s'est fait il faut créer le graphique en lui-même à l'aide de la balise "`<canvas>`". Par la suite dans les balise "`<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 $getgraticData[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 fonction pour la page détail 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 apparaît.

```
#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 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 mettre dans la base de donnée 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 vident. 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 donnée est configurée à NULL.

### 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 utilisation est composée de toutes les infos du l'utilisateur telle que son Email, son prénom et son nom de famille. Sur la même page se trouve trois champs qui permettent à l'utilisateur de changer son mot de passe. Lors que 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 donnée. 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 correcte 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 donnée grâce à la la commande suivante.

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

Si tous les champs sont laissés vide la même commande est exécutée à l'exception que le mot de pas passe n'est pas modifié mais uniquement le champ permettant de cacher les morts. Pour cela lors de la récupération des données du formulaire en vérifie si la checkbox était cochée grâce à un "if(isset)". Si c'est le cas une variable est définit à 0.

*Décrire la réalisation "physique" de votre projet*

- la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)
- les versions des systèmes d'exploitation et des outils logiciels
- la description exacte du matériel
- le numéro de version de votre produit !
- programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.

*NOTE : Évitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...*

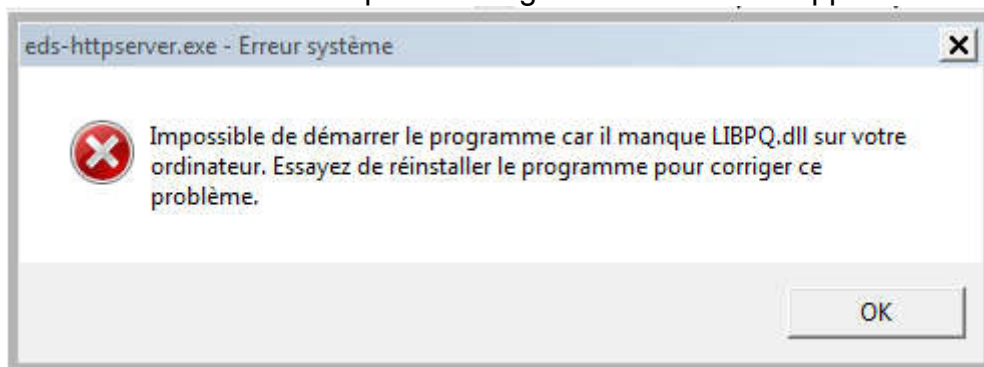
### 3.2 Utilisateurs de la base de donnée

Email de connexion	Mot de passe
samuel.dadie@gmail.com	Samuel
premier.testeur@gmail.com	qwertz

### 3.3 Problèmes rencontrés

#### Lancement du serveur HTTP

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



Cependant tout fonctionne parfaitement. Je n'ai pas remédié à ce problème pour le moment.

#### Problème d'affichage des poids

Lors de l'affichage de tous les animaux la race et le nom s'affiche 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
```

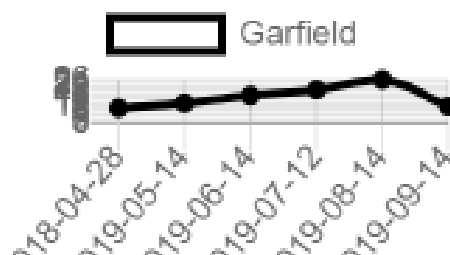
Finalement avec l'aide de Mme Hardegger j'ai suis venu à la conclusion qu'il serait bien plus simple de réaliser cette tâche avec deux requêtes différentes. La première commande sélectionne toutes les informations excepté le poids des animaux.

```
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'animal demandé. Il ne me reste alors plus qu'à parcourir tous les animaux récupérer et d'y exécuter la deuxième requête afin d'obtenir que le dernier poids. De cette façon j'ai toutes les infos 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 forme ordinateur. Mais dès que celui était mis au format téléphone les graphiques devenaient illisibles.



Pour résoudre ce problème j'ai mis le canevas du script dans un div. Cette div est configuré 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.

### 3.4 Description des tests effectués

Nom du test	Déroulement du test	Résultat
Test de l'inscription		
Test de connexion		
Test d'ajout de poids		
Test d'affichage des animaux de l'utilisateur		
Test d'affichage des graphiques		
Test d'affichage des détails		
Test de la modification du compte utilisateur		

- les conditions exactes de chaque test
- les preuves de test (papier ou fichier)
- tests sans preuve: fournir au moins une description

### 3.5 Erreurs restantes

*S'il reste encore des erreurs:*

- Description détaillée
- Conséquences sur l'utilisation du produit
- Actions envisagées ou possibles

### 3.6 Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

- le rapport de projet
- le manuel d'Installation (en annexe)
- le manuel d'Utilisation avec des exemples graphiques (en annexe)
- autres...

## 4 Conclusions

*Développez en tous cas les points suivants:*

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

Tâches	Commentaires	Résultats
Pouvoir visualiser les détails de chaque animal		
Pouvoir ajouter un poids à un animal		
Pouvoir visualiser tous les graphiques sur une seule page		
Pouvoir changer le mot de passe du compte utilisateur		
Pouvoir ajouter un animal		
Pouvoir modifier un animal		

## 5 Annexes

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

### 5.2 Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)*

### 5.3 Journal de travail

Date	Durée	Activité	Remarques

### 5.4 Manuel d'Installation

### 5.5 Manuel d'Utilisation

## 5.6 Archives du projet

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