

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	Concept	5
2.1.1	MCD	5
2.1.2	MLD	6
2.1.3	MPD.....	7
2.1.4	Framework.....	7
2.1.5	Maquettes.....	8
2.2	JavaScript.....	11
2.2.1	Connexion/Déconnexion.....	11
2.2.2	Modification du mot de passe	12
2.2.3	Graphique.....	12
2.2.4	Connexion à la base de donnée.	12
2.2.5	Profil utilisateur	12
2.3	Stratégie de test.....	13
2.4	Risques techniques	14
2.5	Planification	14
2.6	Dossier de conception	14
3	Réalisation.....	15
3.1	Dossier de réalisation	15
3.1.1	Répertoire.....	15
3.1.2	Connexion	15
3.1.3	Inscription	16
3.1.4	Affichage des animaux de compagnies	16
3.1.5	Affichage des graphiques	17
3.1.6	Ajouter un nouveau poids	18
3.2	Problèmes rencontré	19
3.3	Description des tests effectués.....	19
3.4	Erreurs restantes	19
3.5	Liste des documents fournis	20
4	Conclusions	20
5	Annexes.....	20
5.1	Résumé du rapport du TPI / version succincte de la documentation	20
5.2	Sources – Bibliographie.....	20
5.3	Journal de travail	20
5.4	Manuel d'Installation	21
5.5	Manuel d'Utilisation.....	21
5.6	Archives du projet.....	21

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^{ème} année. Le but étant de confirmer les connaissances acquises au cours de ma formation.

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.

1.2 Objectifs

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

Ces éléments peuvent être repris des spécifications de départ.

1.3 Planification initiale

La planification initiale se trouve en annexe : [planif_initial.pdf](#)

2 Analyse / Conception

2.1 Outils

- EasyPhp
- PhpMyAdmin
- PhpStorm
- Suite Office

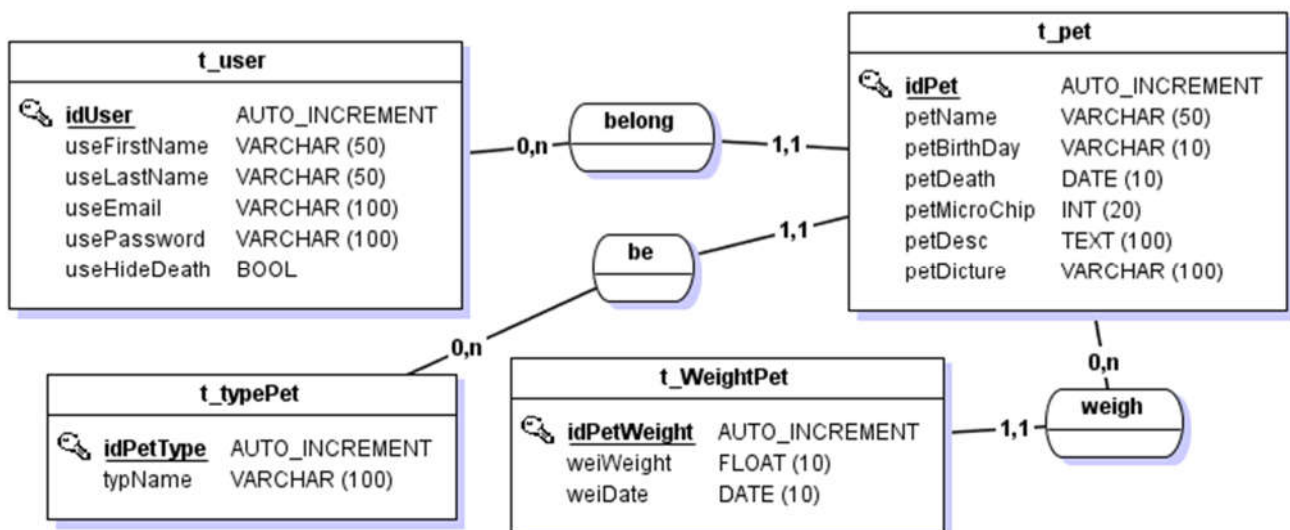
2.2 Méthode de travail

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 partis. 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.

- `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.

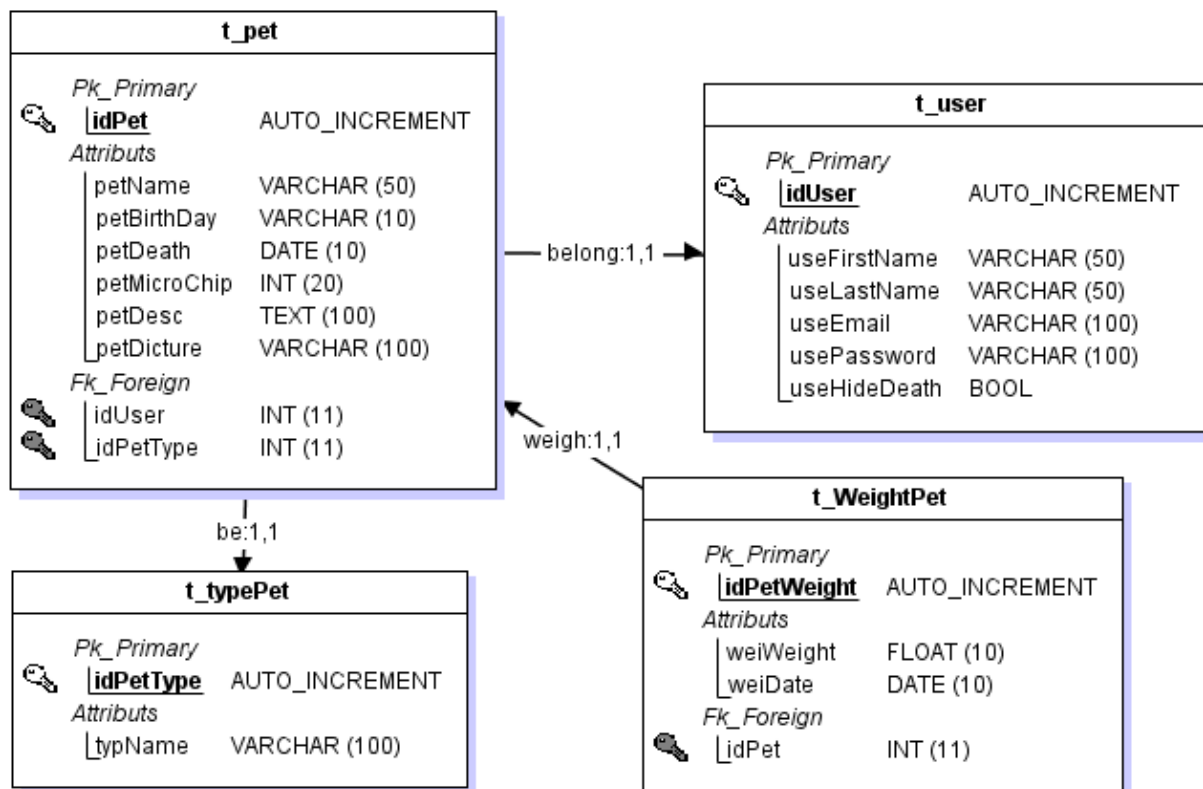
- `t_Weightpet`

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 gère également le responsive.

2.3.5 Maquettes

Le site sera constitué de ... pages.

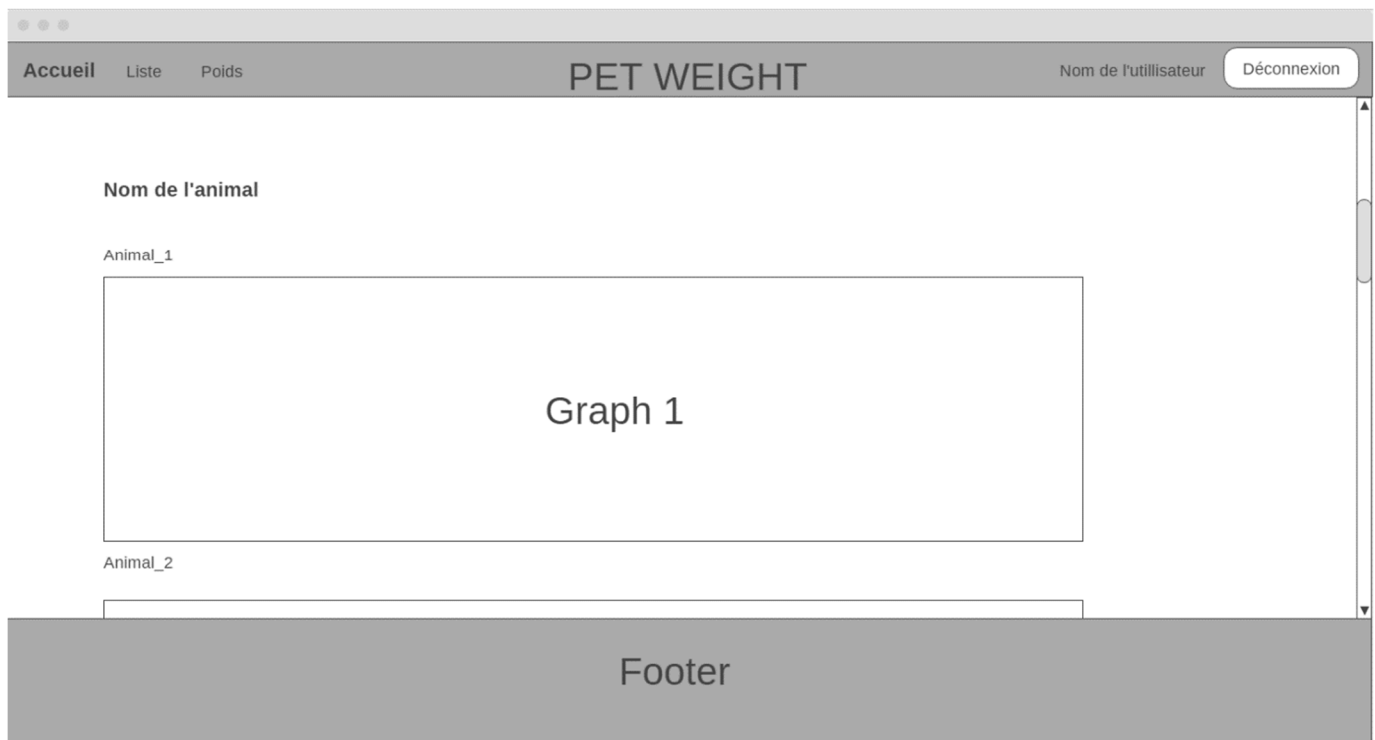
- Page de connexion



La maquette présente une interface de connexion centrée sur l'écran. Le titre 'PET WEIGHT' est en haut. En dessous, le mot 'Connexion' est centré. Il y a deux champs de saisie : 'Email' et 'Mot de passe'. Un lien 'Mot de passe oublié ?' est placé sous le champ de mot de passe. En bas à gauche, il y a un lien 'Créer un compte'. En bas à droite, il y a un bouton 'Connexion'.

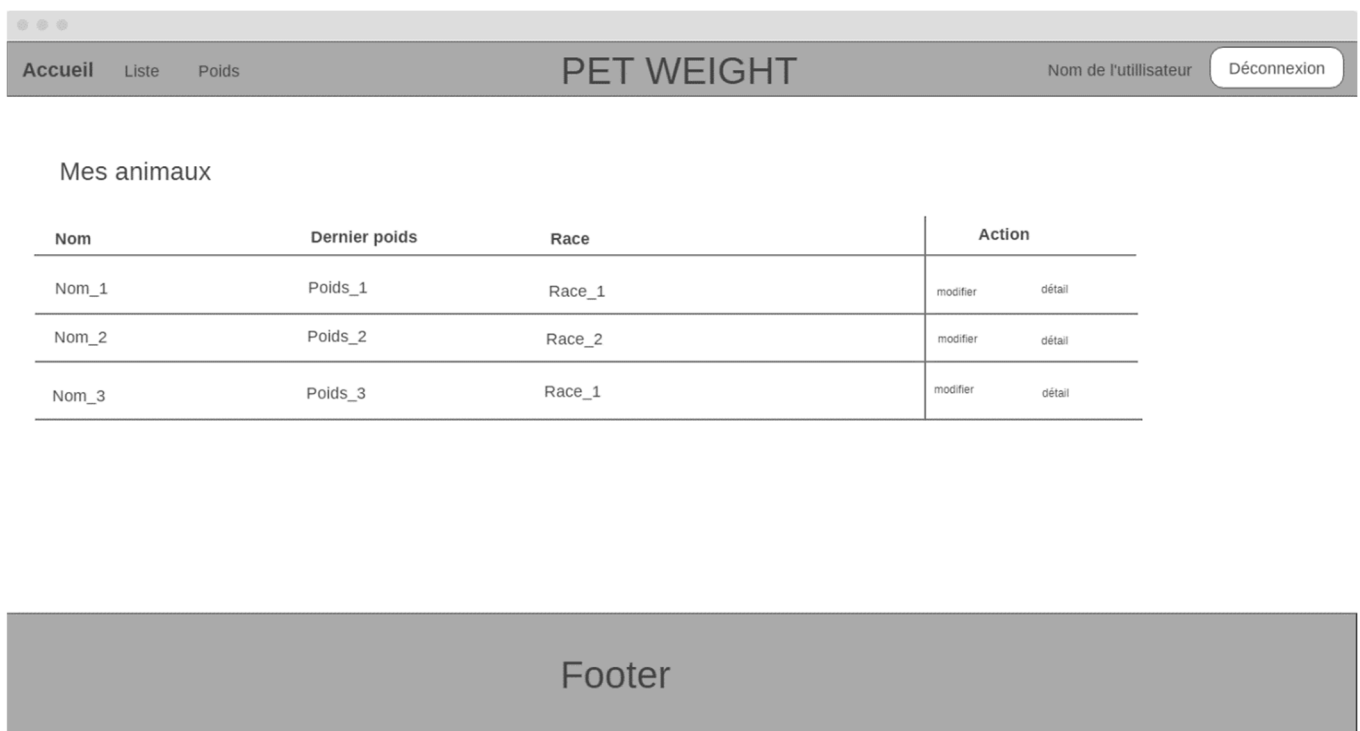
Cette page apparaît lors de l'arrivée sur le site. Il suffit d'entrer son email ainsi que son mot de passe. Si l'utilisateur ne s'est pas encore inscrit une deuxième page s'ouvre afin de lui permettre de s'inscrire.

- 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

The screenshot shows a web application interface for 'PET WEIGHT'. The header bar contains navigation links: 'Accueil', 'Liste', 'Poids', the application name 'PET WEIGHT', the user's name 'Nom de l'utilisateur', and a 'Déconnexion' button. The main content area is titled 'Nom de l'utilisateur' and contains a form for user profile management. The form includes input fields for 'Prénom', 'Nom de famille', 'Mot de passe actuel', 'Nouveau mot de passe', and 'Configurer nouveau mot de passe'. There is also a radio button labeled 'Voir le décès des animaux' and a 'Sauvegarder la configuration' button. The footer is a simple grey bar with the word '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 Serveur SQL

Afin d'exploiter la base de donnée 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 donnée facilement.

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 hasher 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 a rentré est hashé à l'aide de la même méthode. Il ne reste alors plus qu'à comparer les deux hash.

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 Modification du mot de passe

Lorsqu'un utilisateur souhaite modifier son mot de passe il le fait depuis son profil utilisateur. 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 changé.

2.4.4 Graphique

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.

2.4.5 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.

2.4.6 Profil utilisateur

L'utilisateur a 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

Décrire la stratégie globale de test:

- *Types de des tests et ordre dans lequel ils seront effectués.*
- *les moyens à mettre en œuvre.*
- *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
- *données de test à prévoir (données réelles ?).*
- *les testeurs extérieurs éventuels.*
- **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 entrée 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 sélectionné on doit Valider. Il faut alors aller voir les graphiques et s'assurer que le nouveau poids a été ajouter au bon animal.
- **Test d'affichage des animaux de l'utilisateur**

Lors ce que l'utilisateur se connecte il est redirigé sur le 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 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 susnommées et s'assurer que les graphiques s'affichant 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 "voire 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 caché 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.

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 que 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, userEmail, usePassword FROM t_user WHERE userEmail = '$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 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 existants 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, userEmail, 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 mis en utilisation une requête SQL avec plusieurs INNER JOIN. De cette manière il est possible de récupérer le dernier poids de chaque animal ainsi que sa race en plus de son nom, étant donné que toutes ces informations sont enregistrées dans trois tables différentes.


```
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
```

Cette commande ne sélectionne que les animaux appartenant à l'utilisateur connecté. En premier lieu la commande était faite à l'aide d'INNER JOIN. Cependant si avec les "INNER JOIN" la commande ne prend en compte les animaux respectant toutes les paramètres. De ce fait si un animal n'a pas encore de poids il ne sera pas affiché. Alors que en utilisant un "LEFT JOIN" si un champ est vide il affichera les autres champs tout en laissant les autres vides. La ligne "GROUP BY" elle permet de ne récupérer que seul fois un animal même si celui-ci a plusieurs poids. En récupérant les données par ordre décroissant ça me permet de ne récupérer que le dernier poids entré.

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 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 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 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.

Décrire la réalisation "physique" de votre projet

- les répertoires où le logiciel est installé
- 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 : Evitez 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 Problèmes rencontré

Certains animaux manquants

3.3 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

3.4 Erreurs restantes

S'il reste encore des erreurs:

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

3.5 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

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