

Documentation technique

I. Réflexion initiale sur le projet

Dans le cadre de mon Evaluation en Cours de Formation (ECF) pour le projet Arcadia, les besoins étaient multiples. Il fallait en effet proposer un projet complet à la fois en front-end qu'en back-end et c'est donc tout naturellement que je me suis tourné vers le PHP avec PDO pour réaliser ce projet.

PHP permet de créer des pages web dynamiques en affichant les informations mise à jour depuis la base de données sans avoir à changer le code en « brut » en HTML par exemple. Ce côté flexible du langage PHP m'a donc fait me tourner vers ce dernier. Bien qu'il soit le langage principal du projet, j'ai également fait le choix d'utiliser d'autres langages de façon à proposer une application web fonctionnelle :

- HTML 5 : pour sa facilité et son accessibilité dans la création et la réalisation de la structure des pages de mon application web,
- Bootstrap : pour le gain de temps offert lors de la création de mon application web, me permettant de me concentrer sur les points plus techniques de cette dernière, notamment les back-end et les bases de données relationnelles,
- CSS 3 : pour le responsive de mon site et la praticité de ce langage pour la stylisation des pages de mon application,
- Javascript : afin d'apporter un certain dynamisme à ma page web, avec par exemple le carrousel de photos sur la page d'accueil, ou encore des menus déroulant au clic afin d'afficher plus d'informations si le visiteur du site le souhaite,
- SQL : langage privilégié pour la création de base de données, il m'a été utile tout au long de mon projet, que ce soit pour créer ou modifier ma base de données, mais également lors de requêtes plus précises avec par exemple l'insertion de

données directement dans les tables concernées lors de certaines actions des utilisateurs de l'application web Arcadia.

La création de mon application web s'est donc articulée autour de ces 5 principaux langages (PHP, HTML, CSS, JS et SQL) avec l'aide de Bootstrap pour la structure. L'association de ces 5 langages permet un développement efficace aussi bien pour la partie front-end que back-end, de façon à proposer une application web dynamique et qui réponde aux besoins du projet.

II. Configuration de l'environnement de travail

Concernant mon environnement de travail, j'ai choisi d'utiliser un IDE gratuit, Visual Studio Code, qui offre une prise en main facile et dispose de solutions complètes pour l'ensemble des langages utilisés pour la création de mon application web. Les différents add-ons facilitent l'écriture du code mais également la lecture des erreurs. Ayant commencé l'apprentissage du code sur ce dernier, son utilisation me semblait évidente pour ce projet.

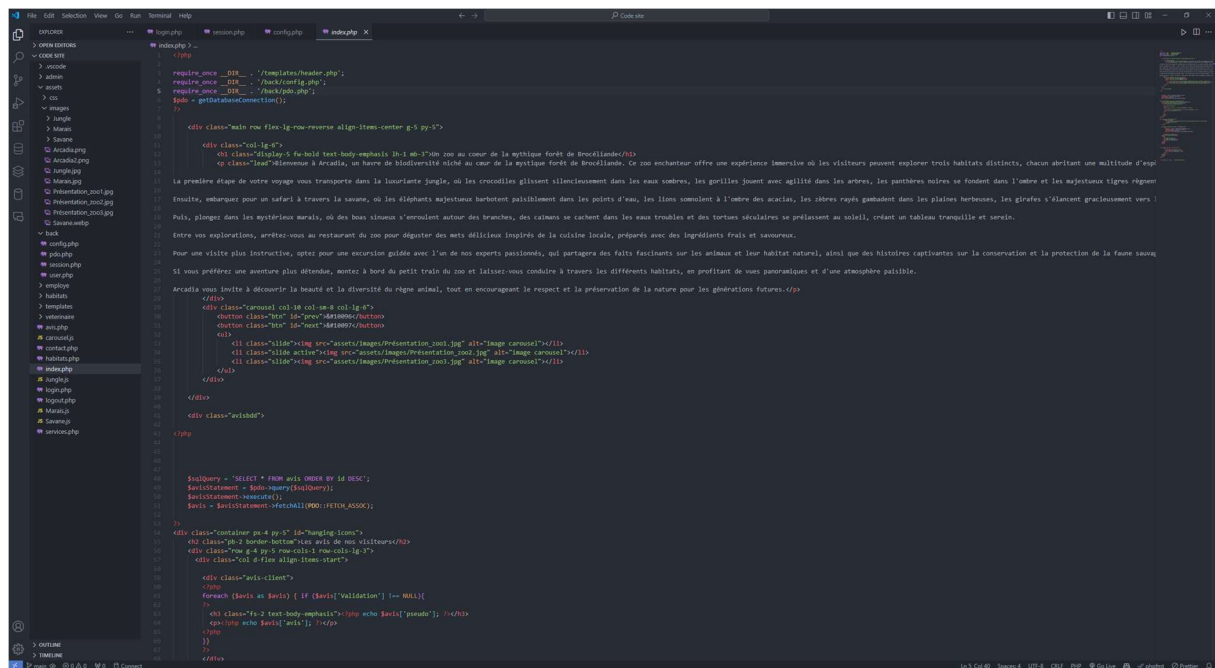


Image 1 : capture d'écran de l'IDE Visual Studio Code (index.php)

Afin de disposer d'un environnement local fonctionnel, j'ai utilisé Xampp, qui propose un serveur web local facile d'utilisation et prêt à l'emploi, couplé à une base de données hébergée sur PhpMyAdmin. Ce dernier permet une introduction ludique au langage SQL de façon à comprendre ce dernier et pouvoir ensuite écrire soi-même ses requêtes.

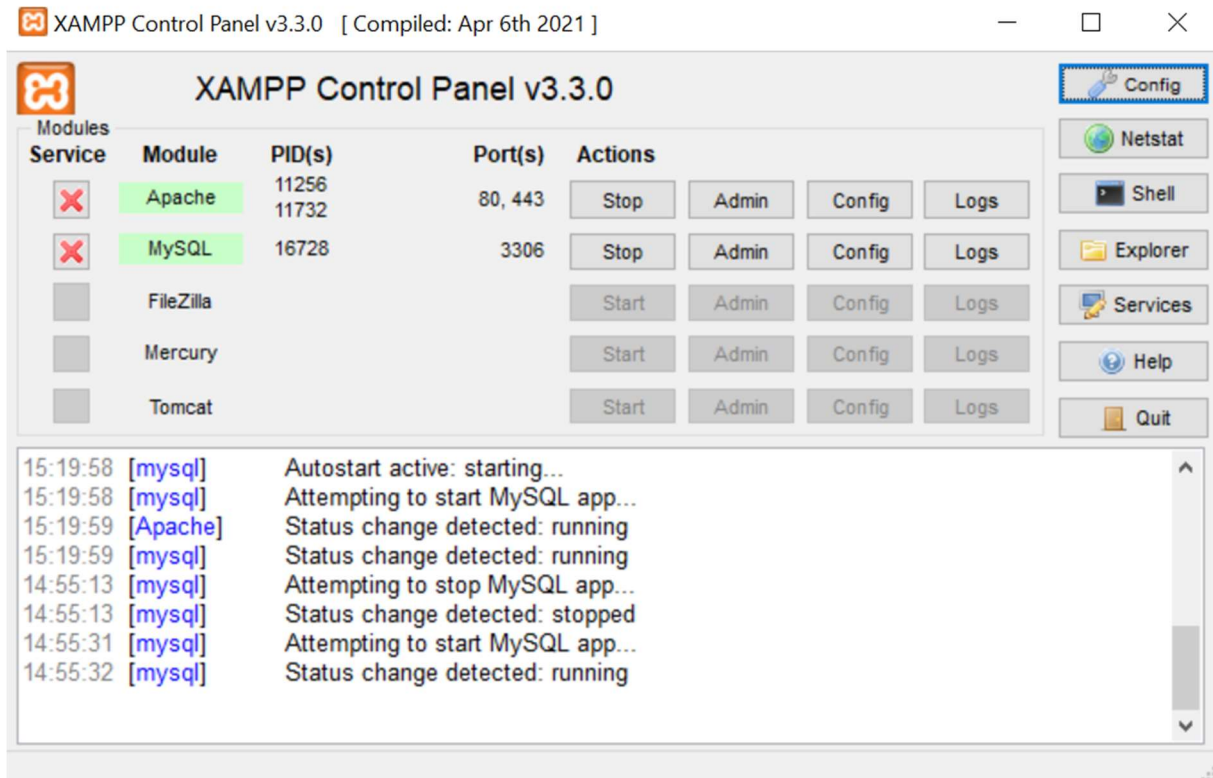


Image 2 : capture d'écran du panneau de contrôle de Xampp

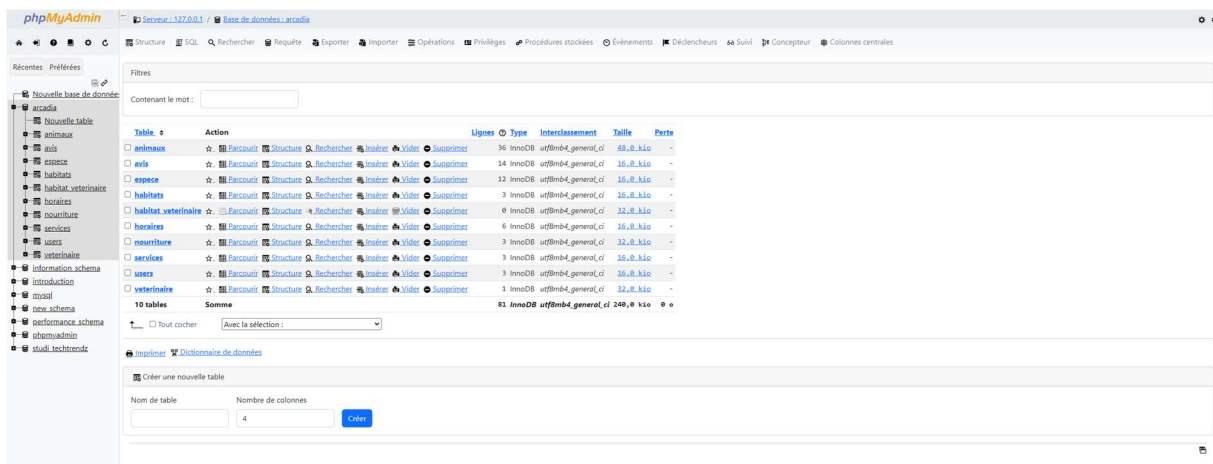


Image 3 : capture d'écran de la base de données du projet – phpmyadmin

III. Déploiement de l'application web

Enfin, j'ai choisi de faire le déploiement de mon application web via Heroku, qui, relié à mon repository GitHub, permet une mise en ligne facile de mon application web. La base de donnée est, elle, hébergée sur OVH.

Le déploiement de mon application web s'est déroulé de la façon suivante :

- Création d'un repository distant sur GitHub via Git GUI
- Clone de mon dossier de code depuis le terminal de Visual Studio Code
- Utilisation des commandes « `git add .` », « `git commit -m` » et « `git push` » de façon à déposer mes modifications sur une branche appelé « `refacto` », que j'ai ensuite Merge avec la branche « `main` », qui elle est reliée à mon compte Heroku

Afin d'effectuer la connexion entre ma base de données OVH et mon code, il a été nécessaire de changer les identifiants de connexion, ainsi, le fichier `pdo.php` a été modifié en conséquence, de façon à faire le lien avec la nouvelle base de données.

IV. Modèle conceptuel de données (MCD)

J'ai fait le choix de créer moi-même ma base de données et de ne pas utiliser celle proposée en annexe du sujet de façon à pouvoir réfléchir sur le projet Arcadia dans sa globalité.

Voici le modèle conceptuel de données du projet :

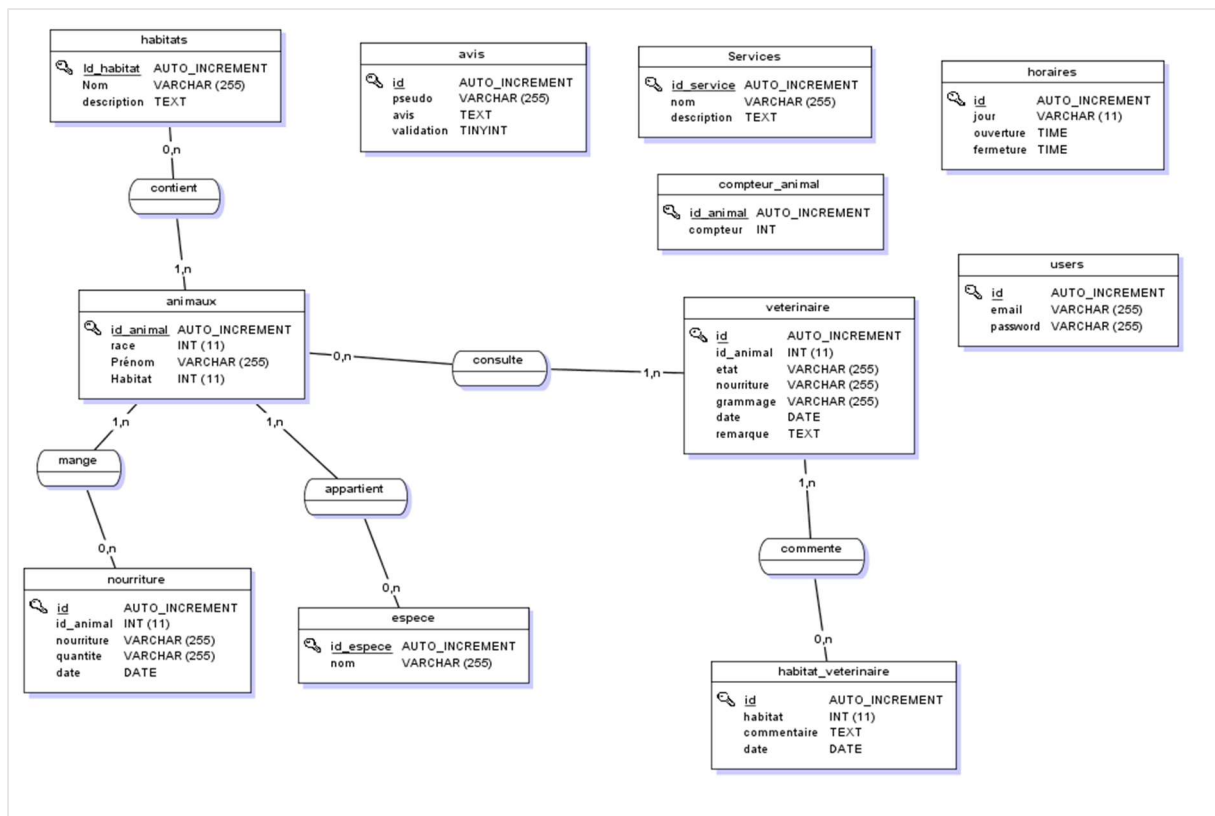


Image 4 : Modèle Conceptuel de Données (MCD)

V. Diagramme d'utilisation et diagramme de séquence

Afin de visualiser au mieux les interactions entre les utilisateurs de mon application web et cette dernière, il convenait de réaliser un diagramme d'utilisation, représentant les utilisations de chacun de l'application web Arcadia.

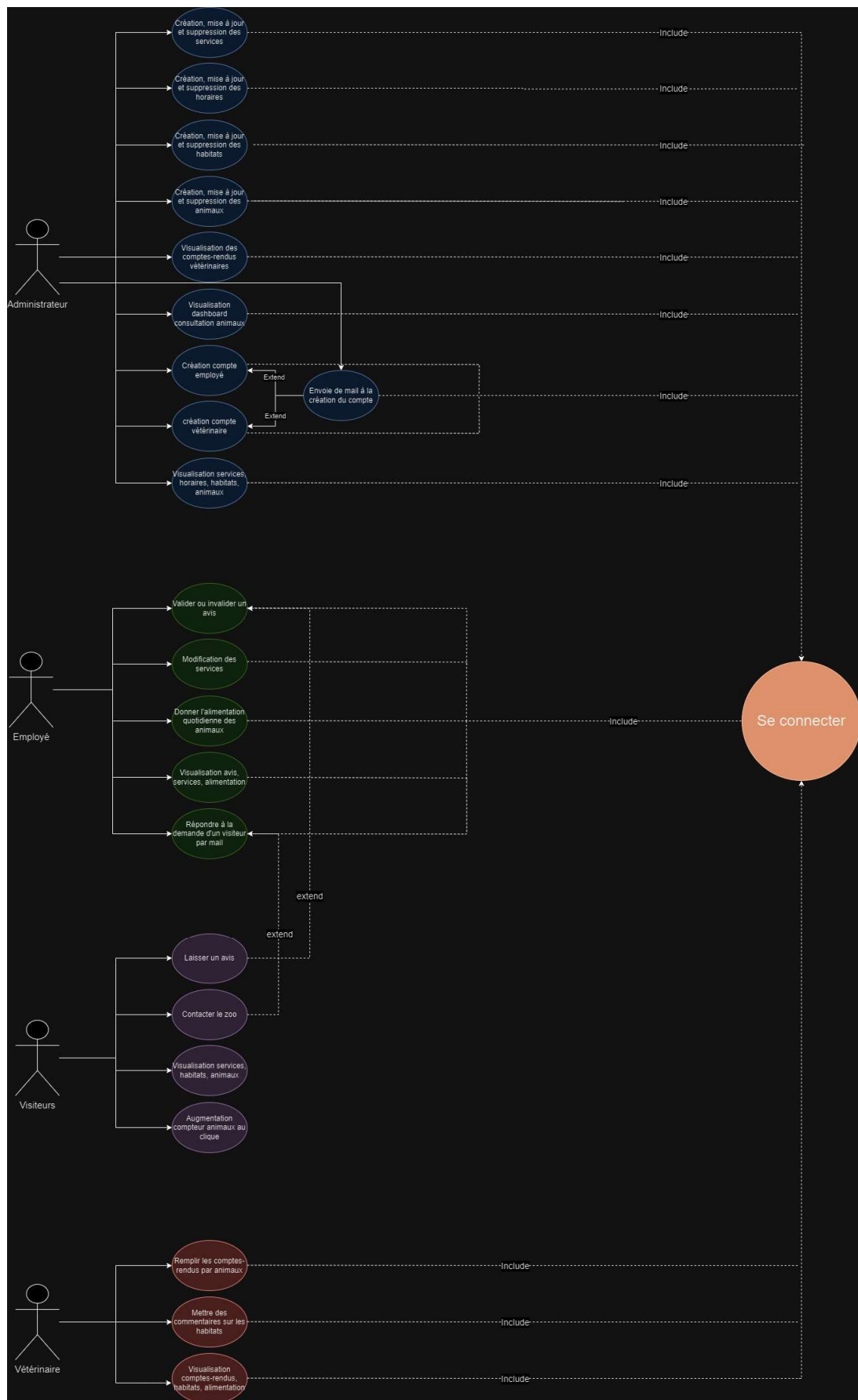


Image 5 : Diagramme de cas d'utilisation

Concernant les différents acteurs du projet (gérant et employés du zoo ainsi que le vétérinaire), il devait leur être proposé de se connecter afin d'accéder à des espaces personnalisés. Voici le diagramme de séquence concernant la connexion de ces acteurs :

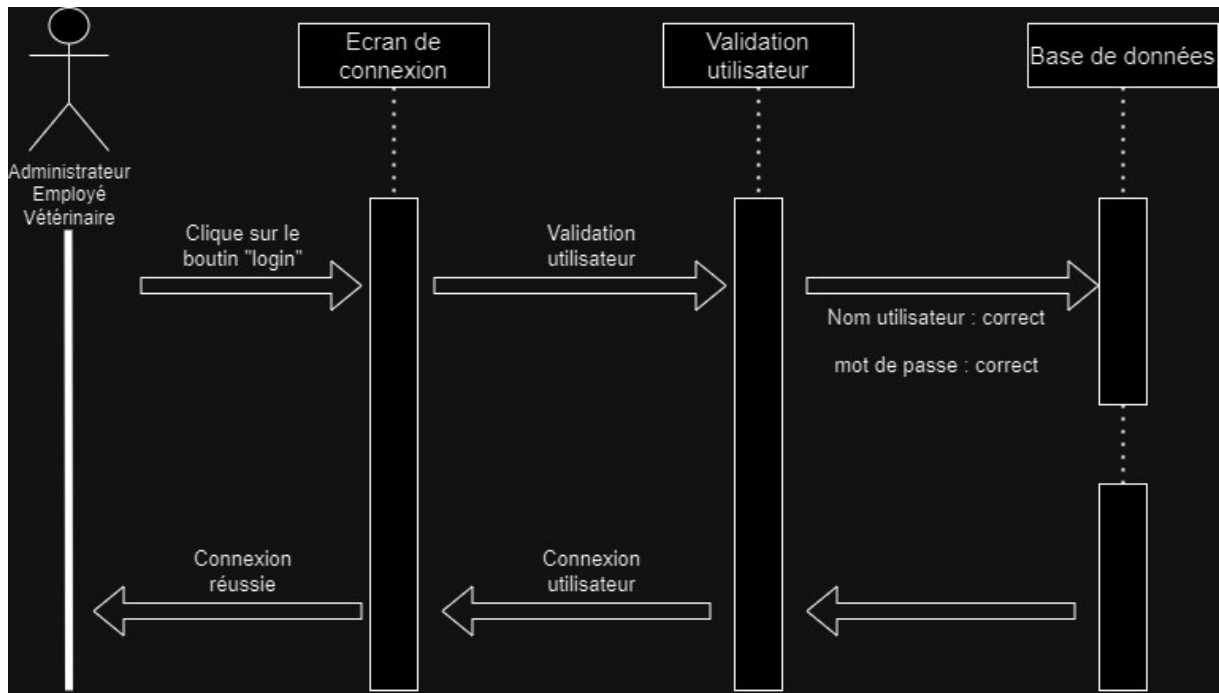


Image 6 : Diagramme de séquence