

Documentation Technique

Réflexions initiale technologique sur le sujet

En analysant les exigences de l'ECF et mon désir de créer un projet sans utiliser de frameworks ou de bibliothèques externes afin d'améliorer mes compétences, j'ai opté pour la solution technique décrite ci-dessous.

Après avoir envisagé initialement d'utiliser Symfony pour ce projet web, j'ai décidé qu'il serait peut-être préférable d'approfondir mes connaissances en PHP et de le développer en utilisant le design pattern MVC, l'un des patterns les plus couramment utilisés sur le web que je souhaite maîtriser au mieux.

J'ai donc choisi d'utiliser PHP avec PDO pour communiquer avec ma base de données relationnelle. Pour le comptage des clics des animaux avec la partie NoSQL, j'ai d'abord envisagé d'utiliser MongoDB, mais après avoir constaté que l'hébergeur sur lequel je comptais déployer mon application prenait également en charge CouchDB, et après avoir examiné sa documentation, j'ai jugé qu'il serait plus facile à mettre en place en utilisant son API avec Curl de PHP, je l'ai donc choisi.

Pour la partie Front-end, j'utiliserai le CSS et le JavaScript. Le JavaScript pour rendre certaines parties du site dynamiques en utilisant fetch pour communiquer avec le back-end, comme par exemple l'affichage des avis d'utilisateurs, la modal de création d'un nouveau avis, l'upload d'image, etc. Ou aussi la création de scripts pour gérer les carrousels, la gestion des menus.

Utilisation des media queries dans mon CSS pour gérer le responsive de l'application, ainsi que les transitions et les animations avec les keyframes pour rendre le site plus dynamique.

Pour la partie des mail, utilisation de la fonction mail de PHP.

Pour le déploiement, j'ai choisi d'utiliser AlwaysData avec son offre gratuite de 100 Mo. Mon projet étant réalisé sans framework ni bibliothèque externe, j'aurai suffisamment d'espace libre pour alimenter le site.

Configuration de votre environnement de travail

Ce projet a été réalisé sous le système d'exploitation Linux, sur l'environnement Ubuntu. J'ai utilisé XAMPP 8.2.4 pour mon serveur local et PHP 8.3.

Les navigateurs Firefox/Chromium afin d'être sûr que mon site est bien le même rendu.

En tant IDE, VScode, ayant l'habitude de l'utiliser.

J'ai utiliser sendMail pour les test de mail en local.

Pour la gestion et le suivi du projet, j'ai utilisé Trello en utilisant la méthode Kanban.

Diagramme de classe

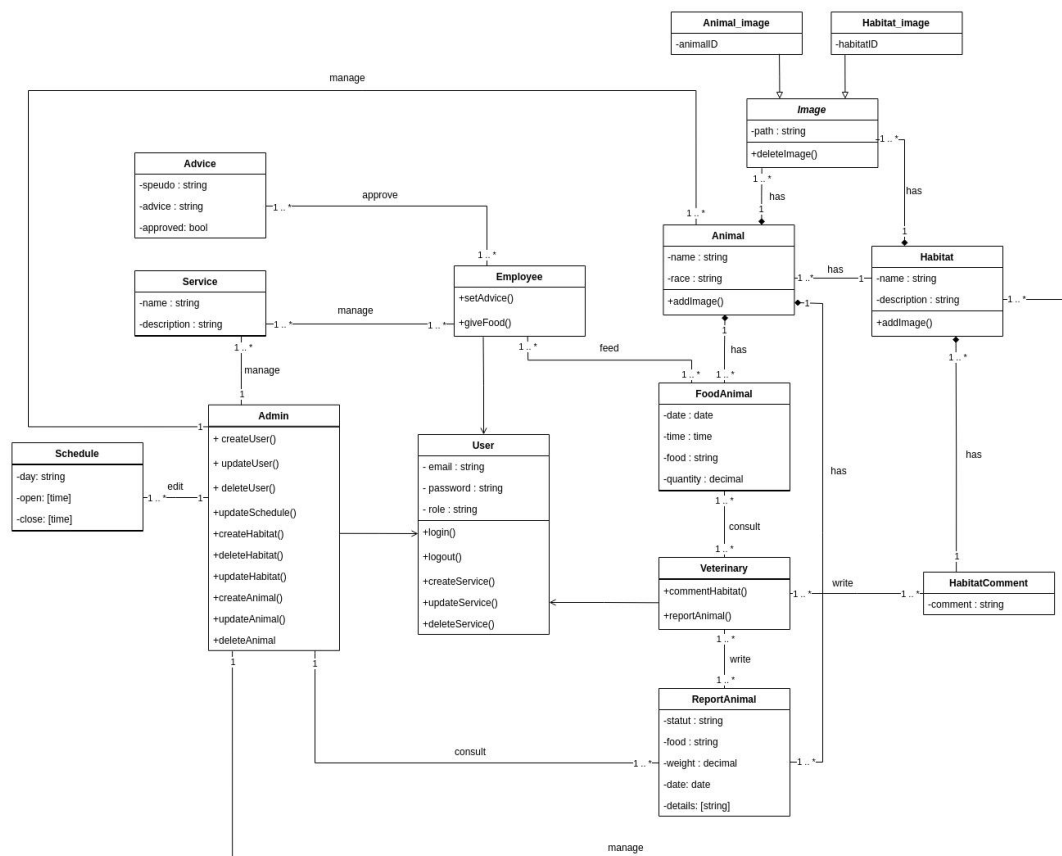


Diagramme de cas d'utilisation

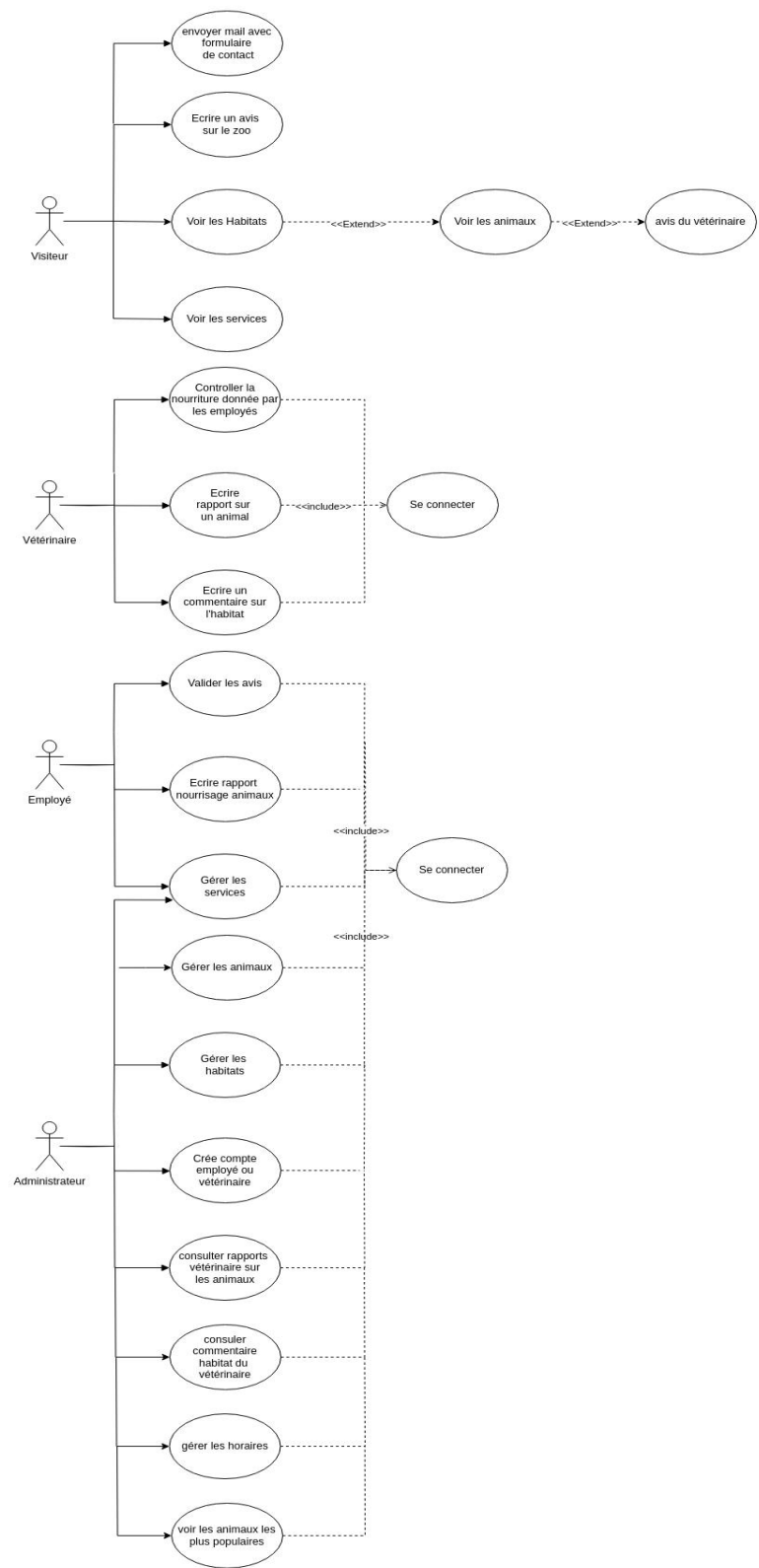
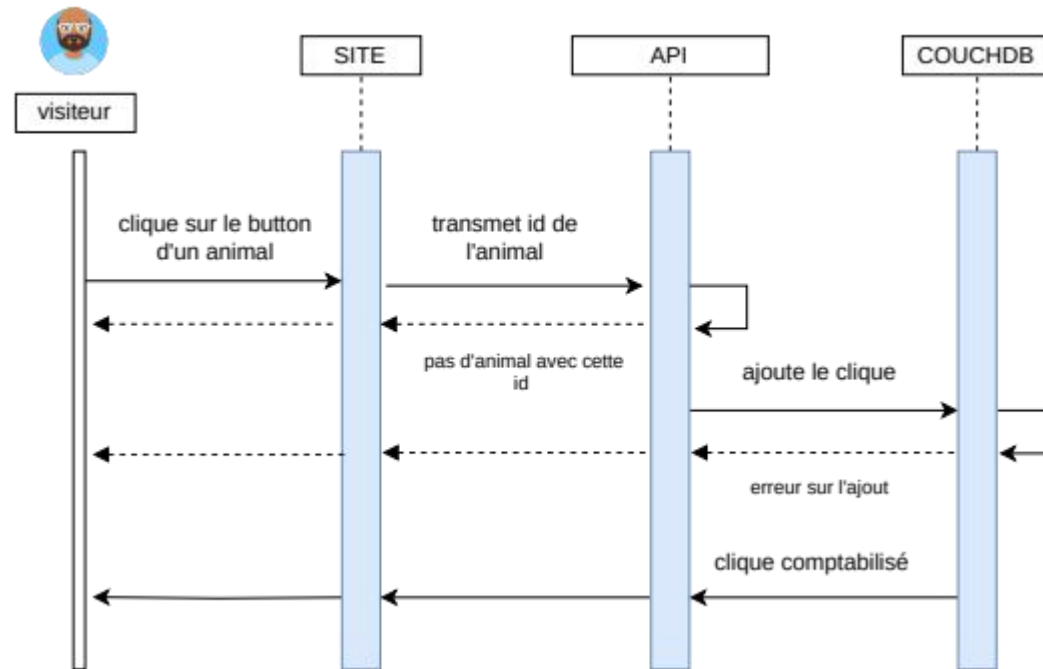


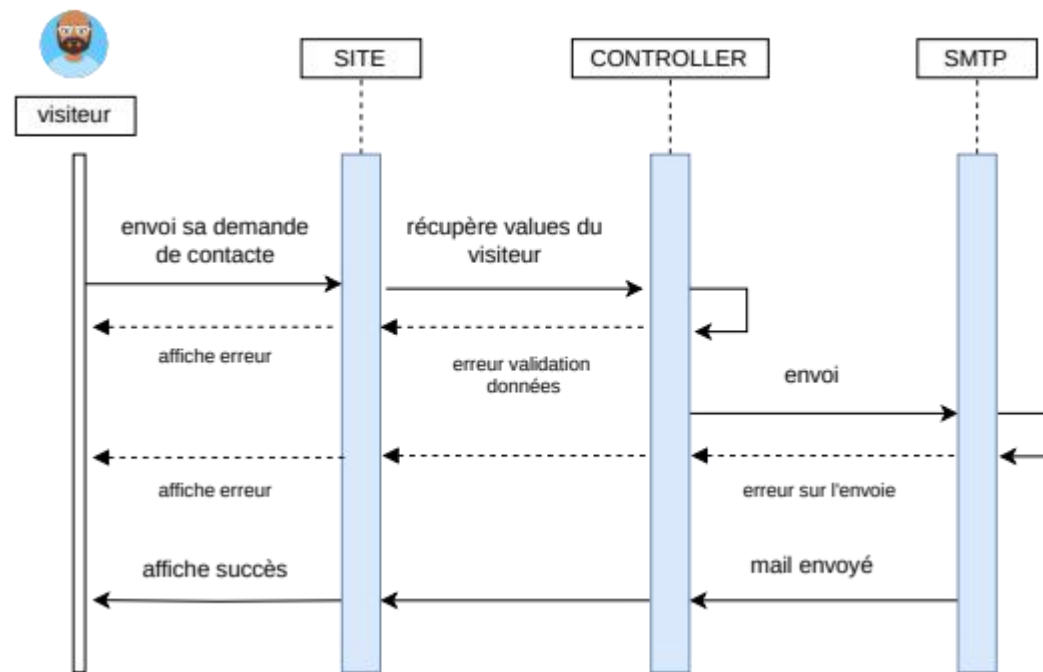
Diagramme de séquence

Ce diagramme de séquence représente l'incrémentation d'un clique quand un visiteur clique sur le bouton d'un animal.

Avec un fetch j'appelle mon api, lui transmet l'id de l'animal, mon api fait la requête à ma base de données noSQL couchDB.



Ce diagramme-ci représente l'envoi d'un mail à l'adresse du Zoo quand un visiteur utilise le formulaire de contact. Au submit du formulaire le contrôleur récupère les informations du visiteur, les vérifie et s'ils sont valides envoient les données à ma class mail qui envoie un mail avec la fonction mail() de PHP.



Déploiement de l'application

Afin de déployer mon site avec l'hébergeur AlwaysData il m'a fallu quelques étapes:

Récupérer l'accès au serveur avec FTP, j'ai utilisé une automatisation avec une action Github, afin d'upload directement ma branche main.

Création et alimentation de ma base de données SQL avec mon fichier arcadia.sql.
Création de ma base de données no SQL avec couchDB et création de l'index afin de permettre la recherche des animaux les plus cliqué.

Configuration de mon config pour y renseigner les informations pour la base de données mySQL et couchDB.

Pour la configuration du SMTP je n'ai rien eu à faire tout été déjà configuré, tout à fonctionner du premier coup.

Une fois tout bien configuré, les corrections de routes que j'avais déclaré en chemin absolu et autres, il me restait simplement à hydrater mon site avec plus de contenu et de vérifier que tout fonctionne bien.