



Documentation technique



par Cédric Phung



Spécifications techniques

Serveur :

Heroku : Plateforme de déploiement

PHP 8.2 : Langage programmation côté serveur

PDO (PHP Data Objects) : Extension PHP pour accéder aux bases de données.

Base de Données :

ClearDB : Service de base de données MySQL hébergé.

MongoDB Atlas : Service cloud pour MongoDB

Front-end :

HTML5 : Langage de balisage pour structurer le contenu des pages web

CSS3 : Langage de feuilles de style

Bootstrap 5 : Framework CSS

JavaScript : Langage de programmation côté client

Back-end :

PHP 8.2 : Langage de programmation côté serveur

Symfony 6.4 : Framework PHP

Doctrine : ORM pour manipuler les bases de données

MySQL : Système de gestion de base de données relationnelle

MongoDB : Base de données NoSQL pour le stockage et la gestion des données non relationnelles.

Cloud :

AWS S3 : Service de Stockage



Choix des technologies

Front-End :

Pour le développement du front-end, j'ai choisi d'utiliser les technologies suivantes :

HTML5 : Pour structurer le contenu des pages web de manière sémantique.

CSS3 : Pour appliquer des styles personnalisés et définir des media queries spécifiques afin de garantir une présentation cohérente sur différents appareils.

Bootstrap 5 : Framework CSS qui offre des composants préconçus qui facilitent la création d'interfaces responsives. Grâce à son système de grilles, Bootstrap permet de gérer facilement et efficacement le positionnement et les breakpoints, ce qui accélère le développement.

JavaScript : Pour ajouter de l'interactivité et permettre des mises à jour dynamiques des données, telles que les filtres et les tris, sans nécessiter un rechargement complet de la page.

Back-End :

Pour le back-end, j'ai choisi d'utiliser les technologies suivantes :

Symfony 6.4 : Framework PHP qui offre une vaste gamme de bundles utiles, tels qu'EasyAdmin, qui facilite la création d'un espace d'administration complet. Symfony fournit également un profiler intégré pour une gestion efficace des erreurs. En termes de sécurité, Symfony protège contre diverses attaques, telles que les attaques CSRF (Cross-Site Request Forgery), grâce à son bundle de sécurité. Il assure également une gestion robuste des sessions et une validation sécurisée des formulaires.

Doctrine : ORM (Object-Relational Mapping) qui permet de prévenir les injections SQL en utilisant des requêtes préparées. Cette méthode sépare les instructions SQL des données fournies par les utilisateurs, ce qui renforce la sécurité contre les attaques par injection SQL.



MySQL : Système de gestion de base de données relationnelle bénéficiant d'une vaste communauté de développeurs. Cette communauté offre un soutien précieux et des solutions rapides aux problèmes rencontrés.

MongoDB : Base de données NoSQL populaire pour sa flexibilité. Son modèle de données basé sur des documents permet une gestion efficace des données non relationnelles.

Serveur :

Heroku : Plateforme de déploiement offrant un excellent service client avec une assistance rapide et efficace. L'intégration directe avec GitHub simplifie le processus de déploiement, et l'utilisation de la CLI facilite la gestion de la mise en production et des configurations. Heroku permet également une connexion simplifiée avec ClearDB, et sa grande communauté en ligne offre de nombreuses ressources et solutions.

Cloud :

AWS : Le service de stockage AWS (Amazon Web Services) me permet de télécharger mes images directement sur le cloud, ce qui évite d'alourdir mon application et améliore ses performances.

Configuration de l'environnement de travail

- Installation Visual Studio Code

Visual Studio Code est un éditeur de code puissant, qui offre des fonctionnalités avancées telles que l'autocomplétion, et l'intégration de nombreux plugins.

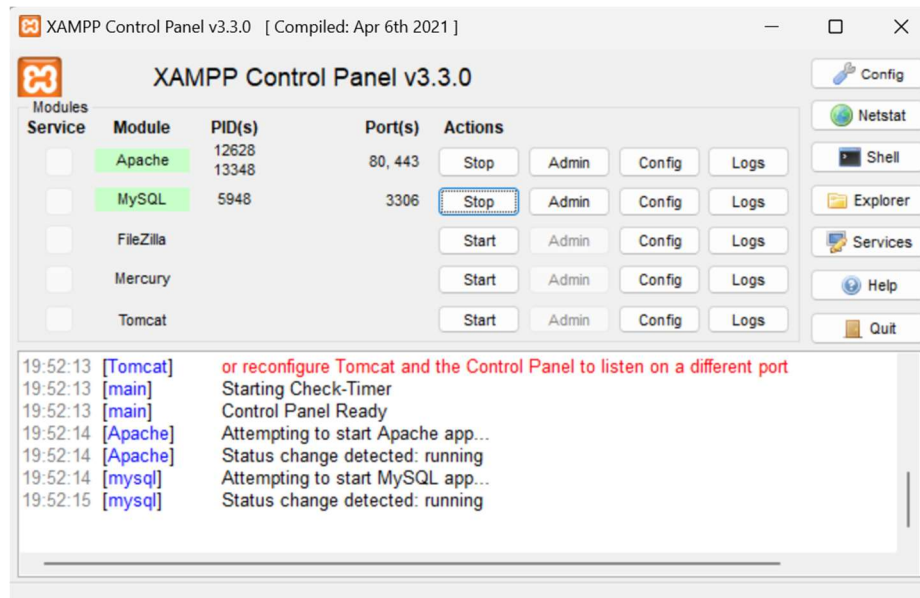
- Installation XAMPP

XAMPP permet de configurer un environnement web complet en local pour tester



l'application en développement. Il inclut Apache (serveur web), MySQL (base de données relationnelle), et PHP.

Lancement de XAMPP :



- Installation MongoDB

MongoDB est une base de données NoSQL permettant de gérer des données en local. Pour intégrer MongoDB avec PHP, il est nécessaire de télécharger et d'installer l'extension PHP PECL pour MongoDB, qui fournit les outils nécessaires pour interagir avec la base de données depuis les scripts PHP et de l'ajouter dans la configuration du fichier php.ini.

Extension PHP dans C:\xampp\php\ext :

-	php_mongodb.dll	08/07/2024 11:05	Extension de l'application
---	-----------------	------------------	----------------------------

- Installation de Symfony : Framework PHP qui facilite le développement d'applications web.

Création d'un nouveau projet :

```
symfony new zoo-broceliande --version=6.4 -webapp
```

Lancement d'un serveur local :



`symfony serve -d`

Création de la base de données :

`symfony console database :create`

Exécuter les migrations de la base de données :

`symfony console doctrine :migrations :migrate`

- **Installation Composer**

Composer est un gestionnaire de dépendances pour PHP. Il permet l'installation, la mise à jour, et la gestion des bibliothèques et des packages PHP.

- **Configuration des fichiers**

Le fichier `.env.local` : contient les paramètres spécifiques à l'environnement local, tels que les informations de connexion aux bases de données SQL (MySQL) et NoSQL (MongoDB).

Configuration `.env.local` :

```
CLEARDB_DATABASE_URL="mysql://root:@localhost:3306/zoo_arcadia"
# CLEARDB_DATABASE_URL="mysql://bba9cc640de7f:5321c453@eu-cluster-west-01.k8s.cleardb.net/
heroku_160217e5999377a?reconnect=true"
# DATABASE_URL="postgresql://app:!ChangeMe!@127.0.0.1:5432/app?serverVersion=16&charset=utf8"

###< doctrine/doctrine-bundle ###
# MONGODB_URL="mongodb+srv://cedric:KHFCuvPbwkyATVd8@zoo-arcadia.sib6qms.mongodb.net/?
retryWrites=true&w=majority&appName=zoo-arcadia"
MONGODB_URL="mongodb://localhost:27017/zoo-arcadia"
MONGODB_DB="zoo_arcadia"
```

- **Installation GitHub** : Plateforme de gestion de versions qui permet de suivre les modifications du code, de collaborer avec d'autres développeurs, et de revenir en arrière en cas d'erreur.

Diagrammes

Diagramme de cas d'utilisation

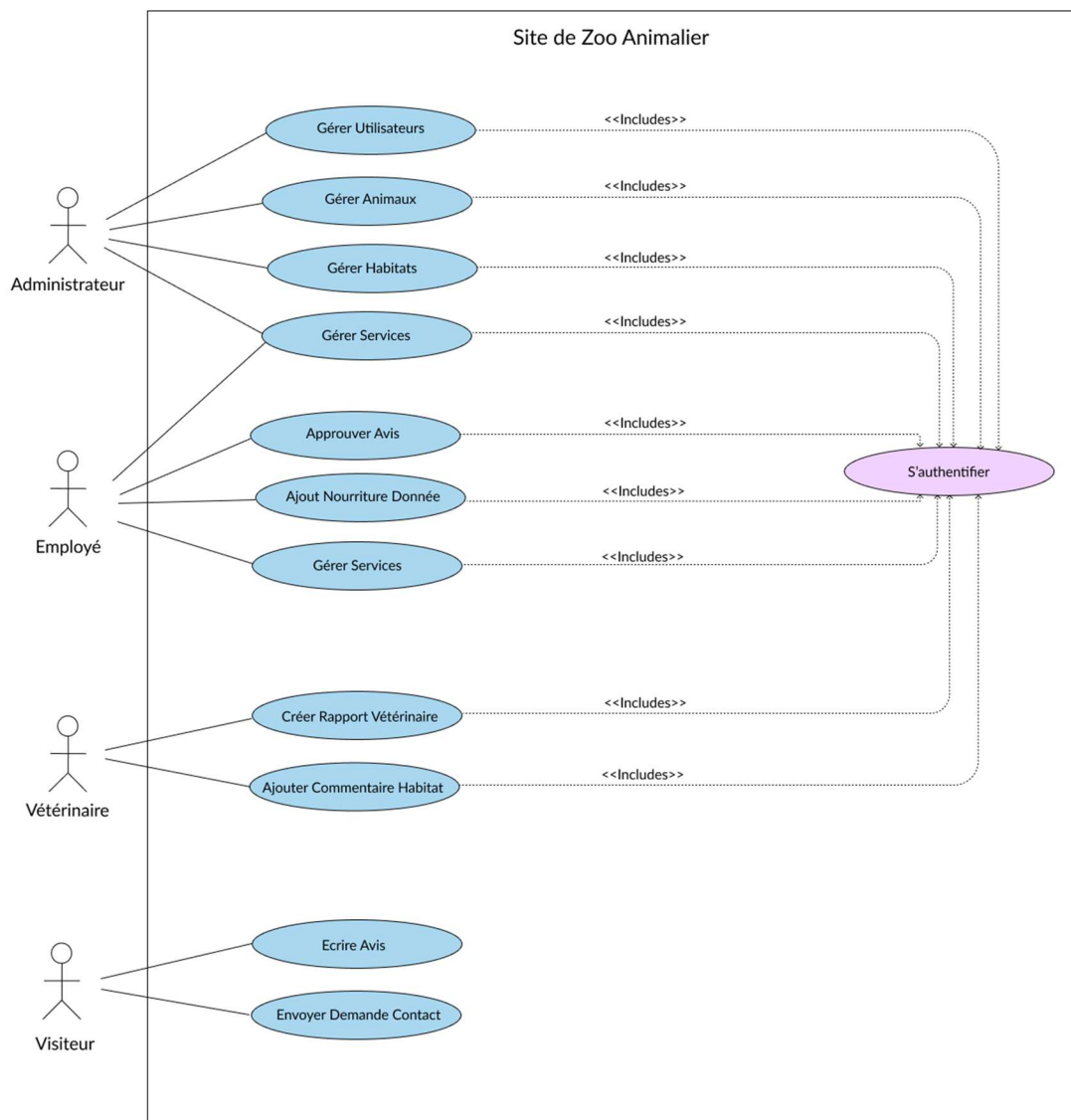


Diagramme de séquence : Création d'un service

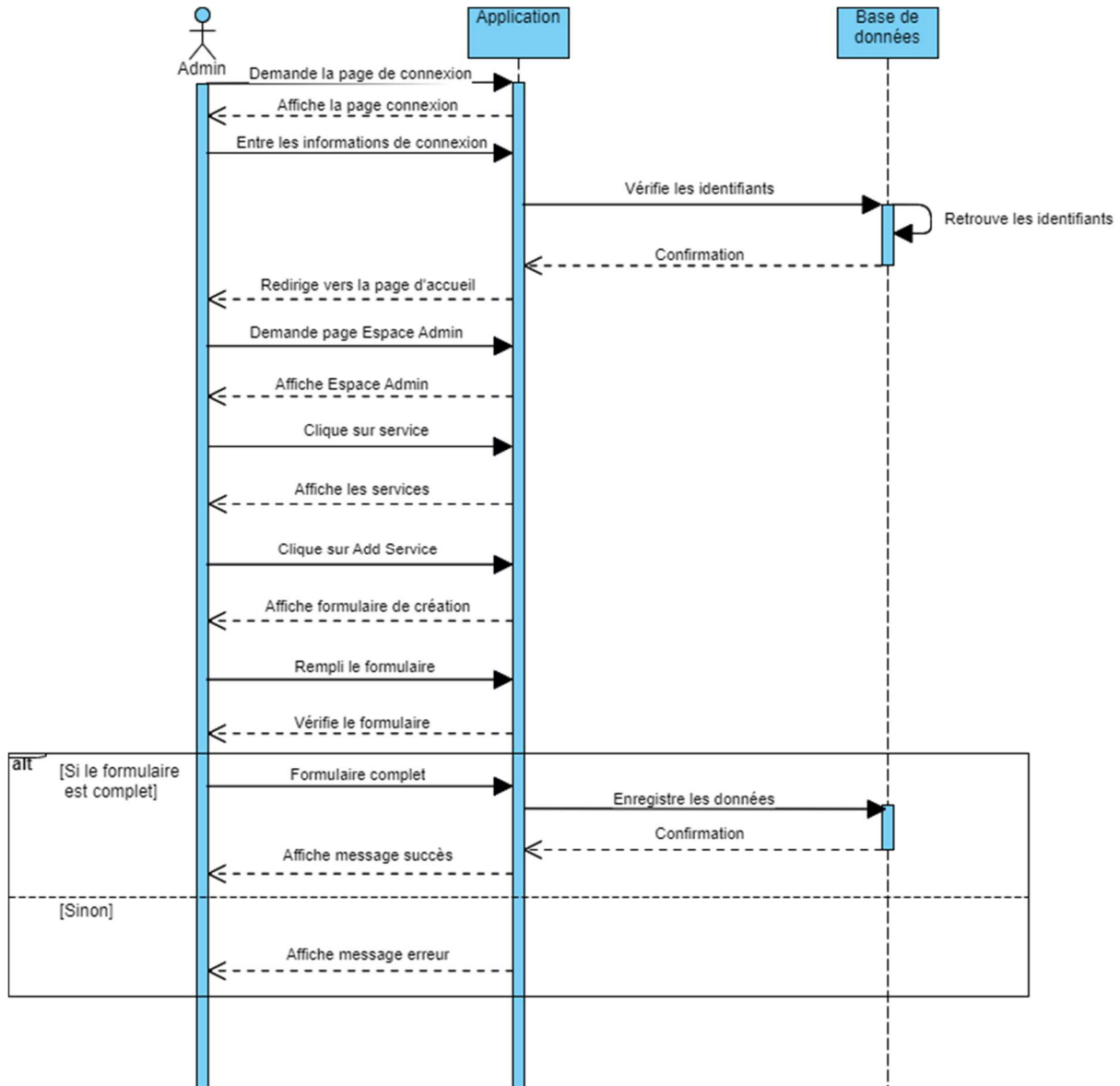
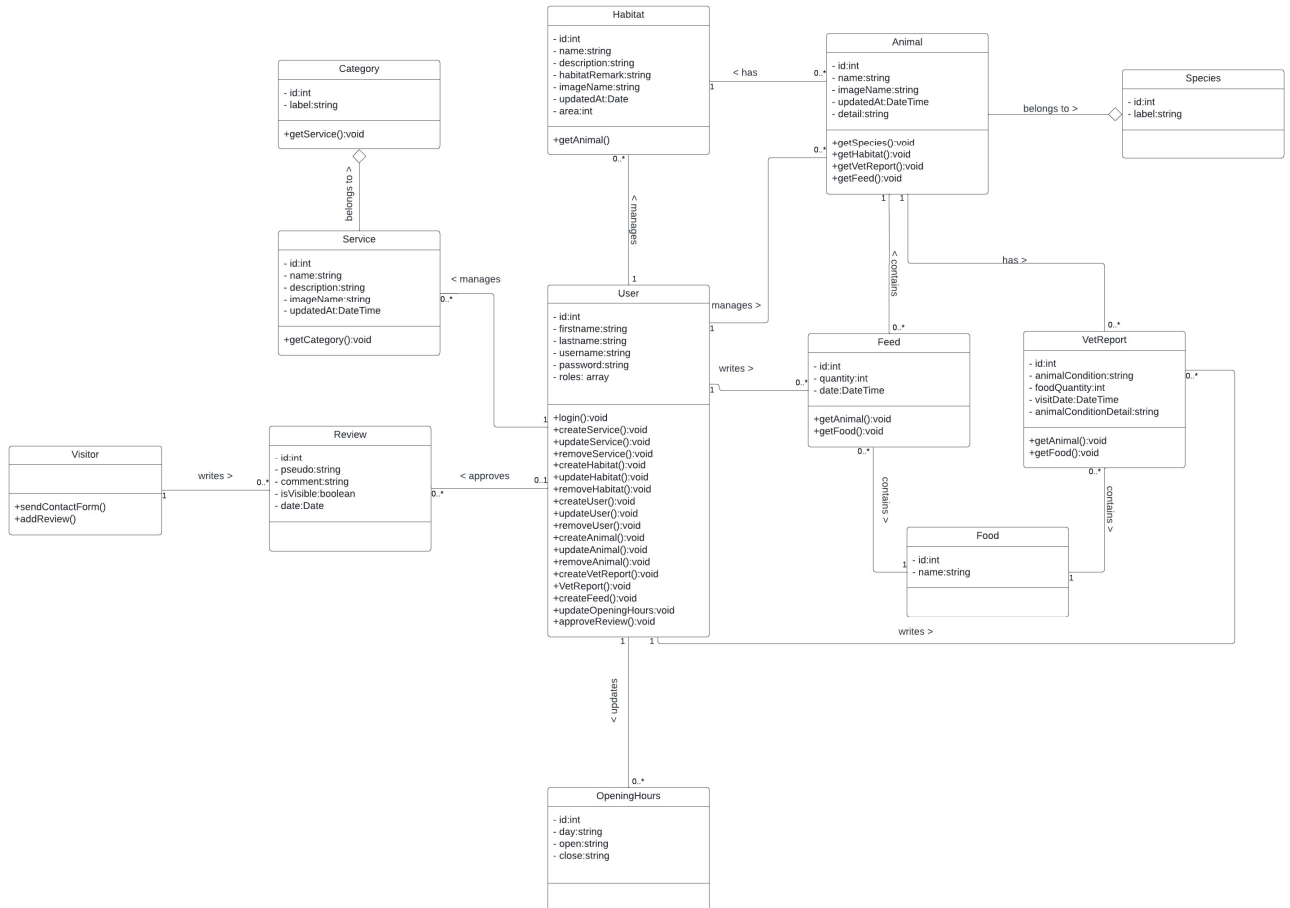


Diagramme de classes



Déploiement de l'application

Sur Heroku :

J'ai créé d'une nouvelle application : *zoo-broceliande-arcadia*

Dans déploiement, j'ai connecté mon compte GitHub et le dépôt de mon application.

J'ai ajouté ClearDB MySQL : dans Ressources puis Add-ons. Les variables d'environnement pour la base de données ClearDB sont automatiquement ajoutées à votre configuration Heroku.



Dans les variables d'environnement, j'ai passé APP_ENV à prod.

Sur MongoDB Atlas : j'ai créé un nouveau projet et j'ai ajouté l'URL et la base de données dans les variables d'environnement d'Heroku.

AWS : J'ai extrait les variables d'environnement depuis AWS et les ai ajoutées à la configuration de mon application sur Heroku.

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some additions come with their own.

KEY	VALUE	
APP_ENV	prod	✎ ✕
APP_SECRET	b1c03f9beea891bd86b1d9a9adec574e	✎ ✕
APP_UPLOAD_SERVICE_URI_PREFIX	https://zoo-arcadia.s3.eu-north-1.amazonaws.com	✎ ✕
AWS_ACCESS_KEY_ID	AKIATCKASDMTSCB3J7W4	✎ ✕
AWS_BUCKET_NAME	zoo-arcadia	✎ ✕
AWS_ENDPOINT	s3.amazonaws.com	✎ ✕
AWS_REGION	eu-north-1	✎ ✕
AWS_SECRET_ACCESS_KEY	1tzBzv5geZ1cketz5grSUT2Q3lmpF12RFBQ7Q66I	✎ ✕
CLEARDB_DATABASE_URL	mysql://bbad9cc640de7f:5321c453@eu-cluste	✎ ✕
MAILER_DSN	smtp://4ecb9a6b8511e8:bd67534083d4e@sand	✎ ✕
MESSENGER_TRANSPORT_DSN	doctrine://default?auto_setup=0	✎ ✕
MONGODB_DB	zoo-arcadia	✎ ✕
MONGODB_URL	mongodb+srv://cedric:KHFCuvPbukyATVd8@zoo	✎ ✕
KEY	VALUE	Add

Variables d'environnement sur Heroku

Sur Symfony :

J'ai utilisé Composer pour mettre à jour les dépendances en excluant les packages de développement avec la commande `composer install --no-dev --optimize-autoloader`. Ensuite, j'ai ajouté le fichier Procfile et le fichier .htaccess.

- Procfile : Ce fichier permet de définir à Heroku les processus à exécuter lors du démarrage de l'application. Par exemple, il peut spécifier la commande pour démarrer le serveur web.



- `.htaccess` : Ce fichier est un fichier de configuration pour le serveur web Apache. Il permet de configurer des fonctionnalités telles que la réécriture des URL, la protection par mot de passe, et d'autres paramètres de configuration du serveur.

Déploiement avec Heroku CLI :

Grâce à Heroku CLI, après m'être connecté à mon projet, je peux utiliser la commande suivante pour déployer les modifications : `git merge heroku main`