

# Le site du zoo ARCADIA

## I) Réflexion sur la gestion de projet :

Après lecture du sujet, j'ai classé les différents points exposés par ordre de réalisation. Les basiques et dispensables en premier car sans eux, on ne peut réaliser des tâches plus complexes.

Les demandes estimées plus complexes telles que les consultations, comptage des animaux, gestion des droits utilisateurs ont été réalisés au fur et à mesure des avancements.

Au final, cette méthode a nécessité de faire des allers- retours entre les différents aspects.

En plus, je souhaitais utiliser des techniques différentes, afin de me familiariser avec chacune d'elles. Par exemple, gestion des polices par soit fichier css, soit en directement dans la fenêtre.

Un point me semblait difficile à mettre en place concerne l'utilisation d'une base non relationnelle comme indiquée dans le sujet pour la comptabilisation de la consultation des animaux. J'ai opté pour la mise en place d'une colonne consultation dans la table animal. Elle s'alimente au fur et à mesure des consultations par les visiteurs.

Le logiciel de gestion de projet est trello.

Il est accessible à l'adresse suivante :

<https://trello.com/b/33XSgGdl/arcadia>

Il permet de décrire les fonctionnalités demandées par le directeur du zoo. Le tableau est prévu sur 5 colonnes : Fonctionnalité, A faire, En cours, Terminée, Livrée). Les fonctionnalités sont reprises par ordre d'importance dans la colonne Fonctionnalité. Une croix sera placée selon l'avancement des fonctionnalités dans les différentes colonnes.

Les cartes basculeront dans l'état d'avancement lorsque la case A faire, En cours, terminée ou livrée seront cochées.

## II) Documentation technique :

Utilisation pour le projet d'un PC portable windows 10 pro.

Afin de mettre en place le projet, utilisation de PHP 8.2 associé au framework Symfony 7.0.9.

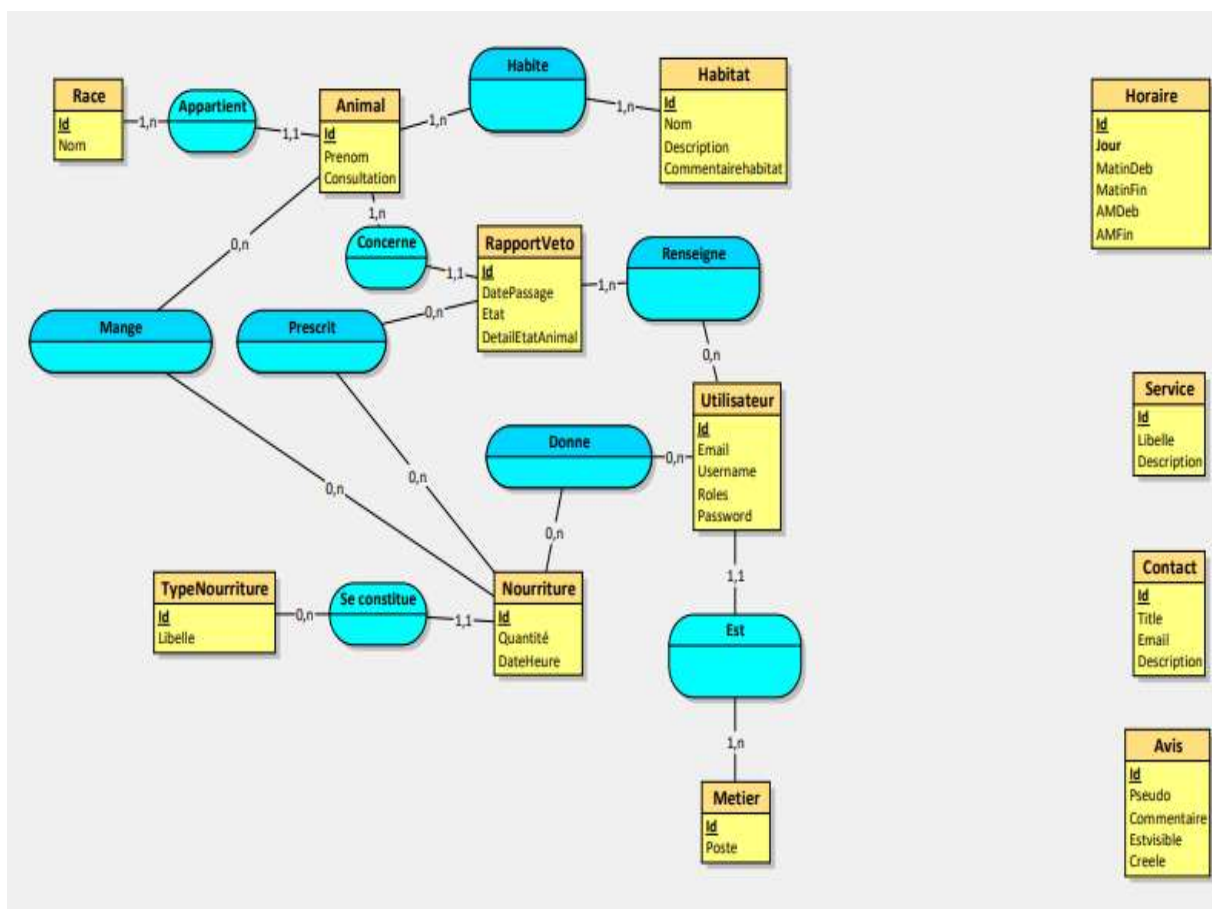
Utilisation également de css pour les feuilles de style. De fichiers twig.

J'ai essayé d'utiliser des techniques et présentations différentes afin de pratiquer au mieux ces différents aspects.

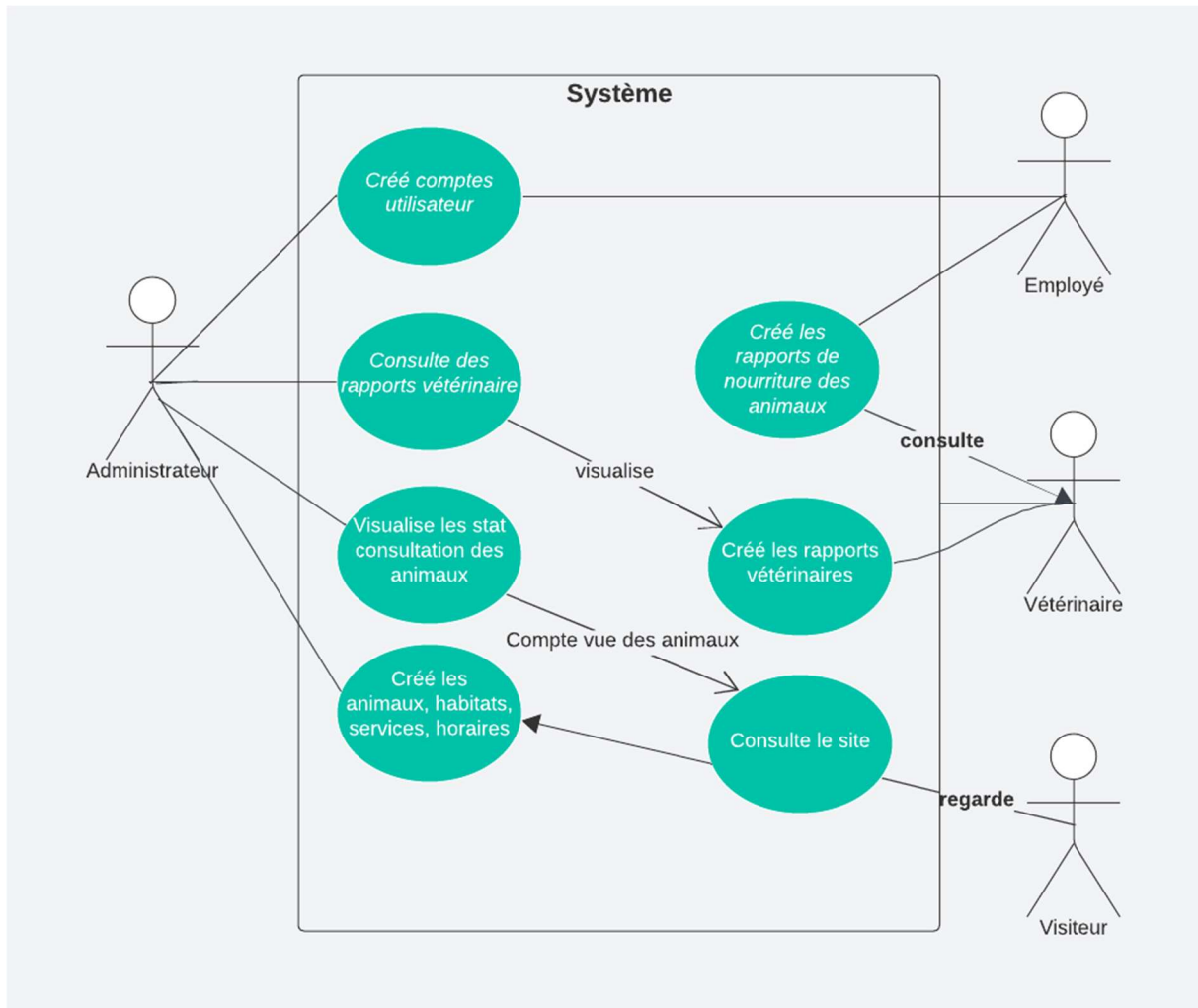
Pour le serveur local, j'utilise XAMPP avec un moteur de base de données MariaDB (base de données relationnelles en open source).

Pour la sécurité des mots de passe même s'il y a seulement l'administrateur qui crée des comptes utilisateurs, j'ai décidé de les sécuriser avec la méthode du hashage proposé par Symfony.

### III) Modèle conceptuel des données :



#### IV) Diagramme d'utilisation



#### Diagramme de séquence

Les participants :

- Utilisateur (Admin, Employé, Vétérinaire, Visiteur)
- Système de Connexion
- Système de Gestion des Comptes
- Service d'Email
- Système de Gestion des Rapports et des Aliments
- Système de Consultation (pour les visiteurs)
- Système de Gestion des Avis
- Système de Statistiques

Les interactions :

Une vision des différentes interactions :

### Connexion

- **Utilisateur -> Système de Connexion** : Demande de connexion
- **Système de Connexion -> Utilisateur** : Demande de autorisations
- **Utilisateur -> Système de Connexion** : Envoie des autorisations
- **Système de Connexion -> Utilisateur** : Confirmation de connexion (ou échec)

### Création de comptes

- **Admin -> Système de Gestion des Comptes** : Créer un compte (Employé ou Vétérinaire)
- **Système de Gestion des Comptes -> Base de Données** : Enregistrer le compte
- **Base de Données -> Système de Gestion des Comptes** : Confirmation d'enregistrement
- **Système de Gestion des Comptes -> Service d'Email** : Envoyer un email de bienvenue
- **Service d'Email -> Employé/Vétérinaire** : Email de création de compte

### Rapports du vétérinaire

- **Vétérinaire -> Système de Gestion des Rapports et des Aliments** : Créer un rapport sur l'état des animaux
- **Système de Gestion des Rapports et des Aliments -> Base de Données** : Enregistrer le rapport
- **Base de Données -> Système de Gestion des Rapports et des Aliments** : Confirmation d'enregistrement

### Saisie de la nourriture

- **Employé -> Système de Gestion des Rapports et des Aliments** : Saisie de la nourriture donnée
- **Système de Gestion des Rapports et des Aliments -> Base de Données** : Enregistrer les informations sur la nourriture
- **Base de Données -> Système de Gestion des Rapports et des Aliments** : Confirmation d'enregistrement

### Consultation par le vétérinaire

- **Vétérinaire -> Système de Gestion des Rapports et des Aliments** : Voir la nourriture donnée
- **Système de Gestion des Rapports et des Aliments -> Base de Données** : Récupérer les informations sur la nourriture
- **Base de Données -> Système de Gestion des Rapports et des Aliments** : Informations sur la nourriture
- **Système de Gestion des Rapports et des Aliments -> Vétérinaire** : Afficher les informations

### Consultation des rapports par l'administrateur

- **Admin -> Système de Gestion des Rapports et des Aliments** : Voir les rapports sur l'état des animaux
- **Système de Gestion des Rapports et des Aliments -> Base de Données** : Récupérer les rapports
- **Base de Données -> Système de Gestion des Rapports et des Aliments** : Rapports
- **Système de Gestion des Rapports et des Aliments -> Admin** : Afficher les rapports

### Consultation par les visiteurs

- **Visiteur -> Système de Consultation** : Voir les habitats, services, animaux, avis
- **Système de Consultation -> Base de Données** : Récupérer les informations
- **Base de Données -> Système de Consultation** : Informations
- **Système de Consultation -> Visiteur** : Afficher les informations

### Validation des avis

- **Visiteur -> Système de Consultation** : Soumettre un avis
- **Système de Consultation -> Base de Données** : Enregistrer l'avis (non validé)
- **Base de Données -> Système de Consultation** : Confirmation d'enregistrement
- **Employé -> Système de Gestion des Avis** : Valider l'avis
- **Système de Gestion des Avis -> Base de Données** : Mettre à jour l'avis (validé)
- **Base de Données -> Système de Gestion des Avis** : Confirmation de mise à jour

### Demandes de contact

- **Visiteur -> Système de Consultation** : Faire une demande de contact
- **Système de Consultation -> Base de Données** : Enregistrer la demande
- **Base de Données -> Système de Consultation** : Confirmation d'enregistrement
- **Employé -> Système de Consultation** : Voir les demandes de contact
- **Système de Consultation -> Base de Données** : Récupérer les demandes
- **Base de Données -> Système de Consultation** : Demandes de contact
- **Système de Consultation -> Employé** : Afficher les demandes
- **Employé -> Visiteur** : Répondre à la demande de contact

### Statistiques de consultation des animaux

- **Visiteur -> Système de Consultation** : Consulter un animal
- **Système de Consultation -> Base de Données** : Enregistrer la consultation
- **Base de Données -> Système de Consultation** : Confirmation d'enregistrement
- **Admin -> Système de Statistiques** : Voir les statistiques des consultations

- **Système de Statistiques -> Base de Données** : Récupérer les données de consultation
  - **Base de Données -> Système de Statistiques** : Données de consultation
  - **Système de Statistiques -> Admin** : Afficher les statistiques
- V) Documentation de déploiement de l'application

Depuis le dépôt public de git dans un projet symfony, vous devez cloner le dépôt dans votre environnement de développement local en procédant de la façon suivante

Ouvrez votre terminal et exécutez la commande `git clone` suivie de l'URL du dépôt. Dans notre cas git clone git clone <https://github.com/catced/ZOO.git>

Accédez au dossier du projet cloné par : `cd zoo`

Vous devez installer les dépendances du projet en utilisant la commande : `Composer install`

Positionnez vous dans le chemin permettant d'accéder aux fichiers. Par exemple : `C:\xampp\www\zoo` pour lancer toutes les commandes. A adapter selon le répertoire d'installation de xampp.

Configurez les variables d'environnement dans le fichier `.env`. Vous devrez peut-être le modifier, notamment l'accès à la base de données.

```
DATABASE_URL="mysql://root:@127.0.0.1:3306/ARCADIA?serverVersion10.11.2-MariaDB&charset=utf8mb4"
```

Selon votre cas, vous devrez adapter le terme `root`, à l'identifiant de votre moteur de base de données si vous l'avez changé.

Indiquer le mot de passe (après les `:`) défini pour accéder au moteur de la base

Changer le port éventuellement. Il doit correspondre à celui paramétré dans le moteur de base de données.

Laissez `ARCADIA` pour le nom de votre base de données.

Créez la base de données à partir de la commande suivante : `php bin/console doctrine:database:create`. Avant de lancer la commande, vérifiez le chemin de travail indiqué dans l'interface.

Démarrez le serveur de développement Symfony en utilisant les commandes : `symfony serve`

Dans le cas où vous n'avez pas installé Symfony CLI, vous pouvez utiliser :

```
php bin/console server:run
```

Lorsque le serveur de développement est prêt, vous devez lancer le moteur de base de données mysql et le serveur Apache présent dans Xampp en cliquant sur les boutons Start.

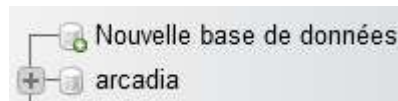
Si les services ont bien démarré, vous devriez avoir le nom des modules avec un fond vert.

| Modules                  |        |                |         |            |
|--------------------------|--------|----------------|---------|------------|
| Service                  | Module | PID(s)         | Port(s) | Actions    |
| <input type="checkbox"/> | Apache | 24040<br>15580 | 80, 443 | Stop Admin |
| <input type="checkbox"/> | MySQL  | 26268          | 3306    | Stop Admin |

Maintenant, nous allons importer le fichier import.sql. Il contient la structure des tables et les données propres à chaque table.

Pour cela, cliquez sur le bouton Admin.

Sélectionner votre base de données.



Cliquez sur le bouton Importer



Dans Fichier à importer, allez chercher le fichier ARCADIA.sql.. Il se trouve à la racine du projet, puis cliquez sur le bouton Importer en bas de la page.

Les tables sont désormais présentes dans la base de données et les données sont également importées.