

Documentation Technique de l'Application

Zoo Arcadia

Réflexions Initiales Technologiques

Objectif : Créer un site vitrine interactif pour le zoo Arcadia, permettant aux clients de visiter virtuellement le zoo, laisser des avis, ou contacter le zoo par courriel. Les administrateurs, employés, et vétérinaires doivent avoir des tableaux de bord adaptés à leurs rôles pour gérer les différents aspects du zoo.

Technologies choisies :

- **Frontend :** Twig (Css/Javascript) pour ça forte compatibilité avec le backend symfony et Bootstrap pour ça facilité d'adaptation.
- **Backend :** Symfony (PHP) pour sa robustesse et facilité de gestion des rôles et permissions.
- **Base de données :** MySQL pour une gestion efficace et structurée des données.
- **Cache et files d'attente :** Redis pour gérer les action no-sql d'incrémentation de vue sur les pages animaux.
- **Authentification :** JWT (JSON Web Token) pour sécuriser les sessions des utilisateurs.
- **Serveur :** Apache ou Nginx pour servir les fichiers de l'application.
- **Outils de gestion de projet :** Trello pour la gestion des sprints et le suivi des tâches.

Configuration de l'Environnement de Travail

Prérequis :

- **IDE** : VS Code.
- **Node.js** : Version 18.x ou supérieure pour gérer les dépendances front-end.
- **Composer** : Pour gérer les dépendances PHP.
- **MySQL** : Version 8.0 ou supérieure.
- **Redis** : Version 6.x ou supérieure.
- **PHP** : Version 8.2 ou supérieure.
- **Apache**: Pour servir l'application.

Étapes de configuration :

1. Initialiser le dépôt :

```
git clone https://github.com/monarche05/zoo-arcadia-v2.git
cd zoo-arcadia-v2
```

2. Installer les dépendances backend :

```
composer update
```

3. Installer les dépendances frontend :

```
npm install
```

4. Configurer le fichier .env :

```
APP_ENV=dev
APP_SECRET= "Votre clé secret"
DATABASE_URL="mysql://"user":"mot de passe"@127.0.0.1:3306/"database"?
serverVersion=10.11.2-MariaDB&charset=utf8mb4"
MESSENGER_TRANSPORT_DSN=doctrine://default?auto_setup=0
REDIS_URL=redis://127.0.0.1:6379
```

Note : Symfony génère automatiquement les clés secrètes à l'installation mais dans notre cas nous devons la générer.

Créez un fichier secretGen.php:

```
<?php  
echo bin2hex(random_bytes(16));
```

Exécutez le fichier :

```
php secretGen.php
```

5. Initialiser la base de données :

```
php bin/console doctrine:database:create  
mkdir -p C:\your-path\zoo-arcadia-v2\migrations  
php bin/console make:migration  
php bin/console doctrine:migrations:migrate
```

6. Installer et démarrer le server Redis :

```
curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o  
/usr/share/keyrings/redis-archive-keyring.gpg  
  
echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg]  
https://packages.redis.io/deb $(lsb_release -cs) main" | sudo tee  
/etc/apt/sources.list.d/redis.list  
  
sudo apt-get update  
sudo apt-get install redis  
  
sudo service redis-server start
```

7. Démarrer le serveur :

```
symfony server:start
```

Diagramme d'Utilisation

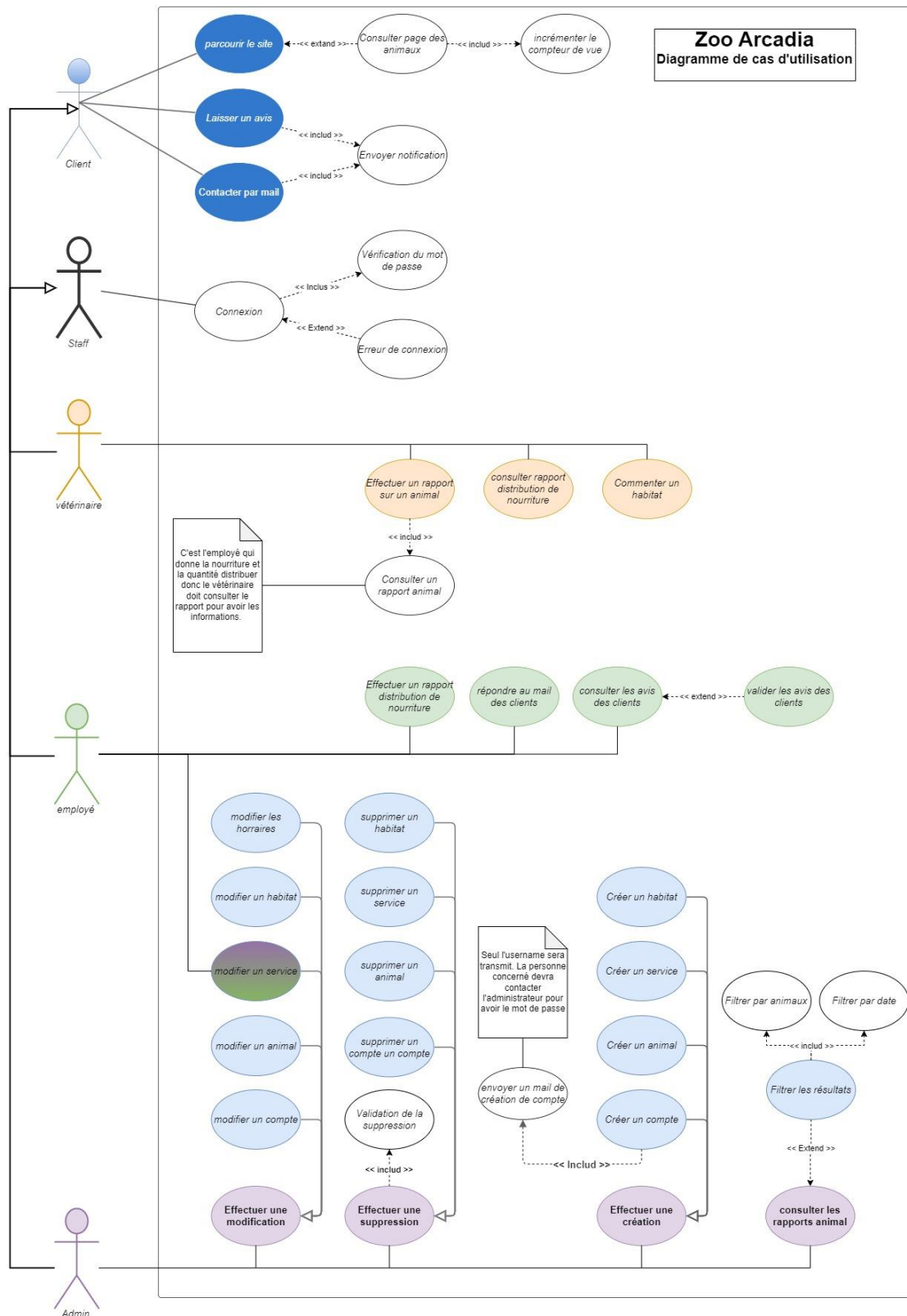
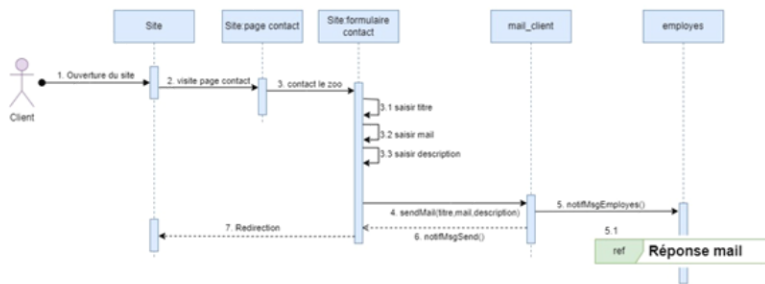


Diagramme de Séquence (Client)

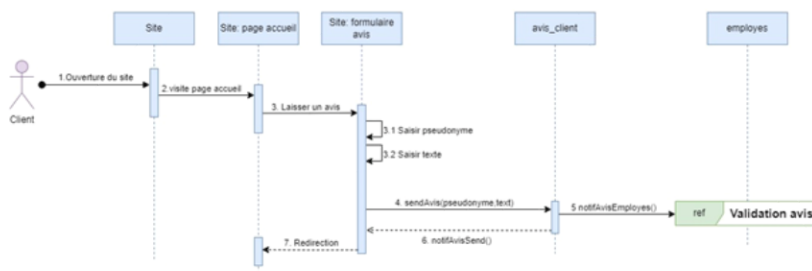
Client:Contacter



Nom: Contacter
Acteur: Client, Employé
Contexte: Le client contact via le formulaire de contact de la page "Contact"
Scénario principal:

1. Le client accède au site avec l'adresse HTTP
2. Le client vas sur la page contact
3. Le client remplit le formulaire de contact devant contenir:
 - 3.1 un titre
 - 3.2 un e-mail
 - 3.3 une description
4. Le client envoie le formulaire via le bouton "Envoyer"
5. Le système envoie une notification à l'acteur "Employé"
- 5.1. L'employé pourras répondre par mail une foi qu'il aura pris connaissance du message
6. Le système envoie un message de confirmation d'envoi au client
7. Le client est redirigé vers la page d'accueil

Client:Laisser un avis



Nom: Laisser un avis
Acteur: Client, Employé
Contexte: Le client laisse un avis via le formulaire de la page "Accueil"
Scénario principal:

1. Le client accède au site avec l'adresse HTTP
2. Le client vas sur la page accueil
3. Le client remplit le formulaire d'avis devant contenir:
 - 3.1 un pseudonyme
 - 3.2 un texte
4. Le client envoie le formulaire via le bouton "Envoyer"
5. Le système envoie une notification à l'acteur "Employé"
6. Le système envoie un message de confirmation d'envoi au client
7. Le client est redirigé vers la page d'accueil

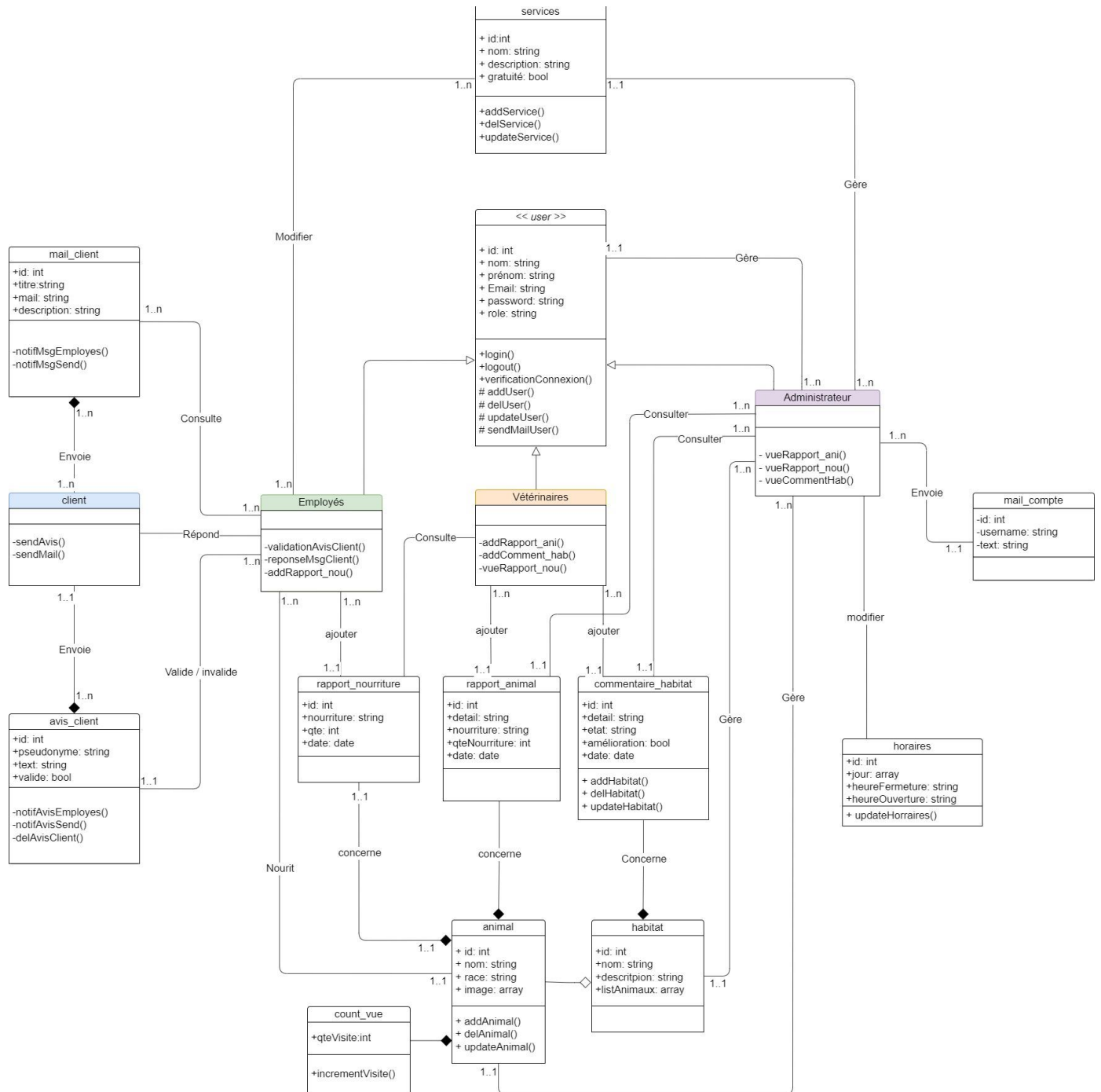
Client:Visite page animal



Nom: Visite page animal
Acteur: Client
Contexte: Le client visite la page d'un animal
Scénario principal:

1. Le client accède au site avec l'adresse HTTP
2. Le client vas sur la page animaux
3. Le client visite la page d'un animal
4. Le système incrémente le conteur de vue

Diagramme de class



Documentation du Déploiement de l'Application

Étapes de déploiement :

1. Préparer l'environnement de production :

- Installer et configurer le serveur web (Apache/Nginx).
- Installer MySQL et Redis.

2. Configurer les variables d'environnement :

- Définir les variables d'environnement nécessaires dans le fichier `.env` de l'application ou dans les paramètres du site proposé par l'hébergeur.

3. Déployer le code :

- Cloner le dépôt sur le serveur de production.
- Installer les dépendances avec Composer et npm.
- Générer les assets frontend.

4. Initialiser la base de données :

- Créer la base de données et exécuter les migrations.

```
php bin/console doctrine:database:create
mkdir -p C:\your-path\zoo-arcadia-v2\migrations
php bin/console make:migration
php bin/console doctrine:migrations:migrate
```

5. Configurer le serveur web :

- Configurer le virtual host pour pointer vers le dossier public de l'application.
- Redémarrer le serveur web.

6. Vérifier le déploiement :

- Accéder à l'application via le navigateur pour s'assurer que tout fonctionne correctement.