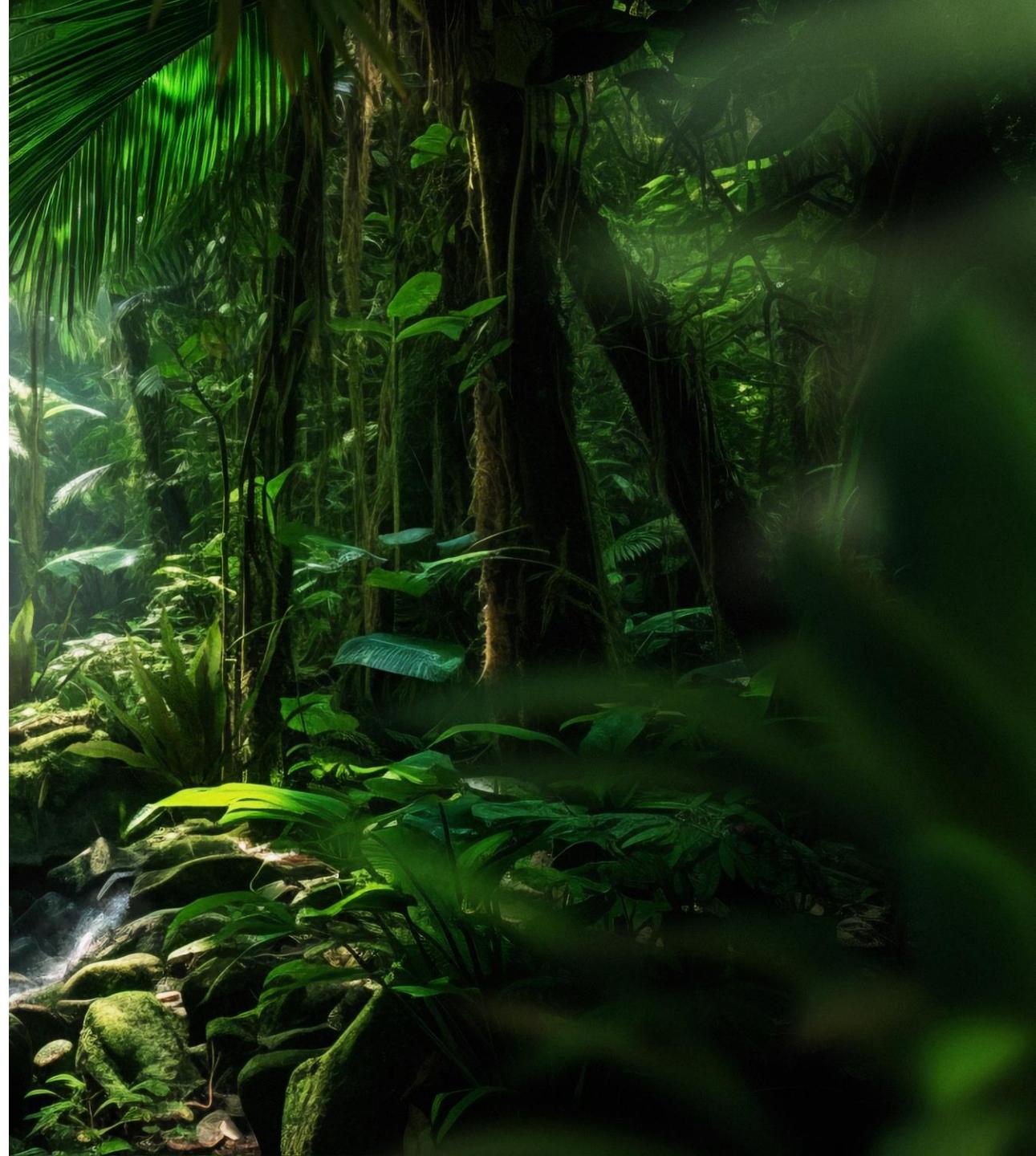


# TP DWWW

Sébastien Borgne

26/09/24



# Déroulé de la présentation

- 1. Présentation personnelle**
- 2. Présentation du projet**
- 3. Cahier des charges**
- 4. Stack technique**
- 5. Gestion de projet**
- 6. GitHub**
- 7. Charte Graphique, wireframes & maquettes**
- 8. Conception de la base de données**
- 9. Création des entités**
- 10. Dynamisation de l'interface**
- 11. Cycle de vie d'une fonctionnalité**
- 12. Mécanismes de sécurité**
- 13. Conteneurisation pour un environnement de développement**
- 14. Protection des Données**
- 15. Accessibilité**

# Présentation personnelle :

2017 : BAC STI2D SIN

2017 - 2019 : Licence Psy + Cuisine weekends

2019 - 2020 : Commis de cuisine

2020 - 2021 : CAP 1An Cuisine

2021 - 2022 : MCCDR

2022 - 2023 : Second de cuisine

2023 - 202? : Reconversion, TP DWWM

2024 : Alternance B3 Rufiji Capital & Sup' de Vinci

# Projet Arcadia

- ▶ Évaluation en Cours de Formation ( ECF )
- ▶ Sujet & Cahier des charges imposés
- ▶ Projet en autonomie



# Zoo Arcadia :



Forêt de Brocéliande, en Bretagne



- José, directeur du zoo
- Employées
- Vétérinaires
- Public



Communication publique, opérations métiers quotidiennes pour salariés & directeur.

# Cahier des charges :

- ▶ Gestion des comptes
- ▶ Gestion accès privilégiés
- ▶ Application métier: Opérations quotidiennes depuis espace réservés aux salariés
- ▶ Avis clients et demandes de contact
- ▶ Affichage services, habitats, avis clients, animaux & leur dernier bilan vétérinaire.

# Stack technique :

Front :



Moteur de templates :



Back :

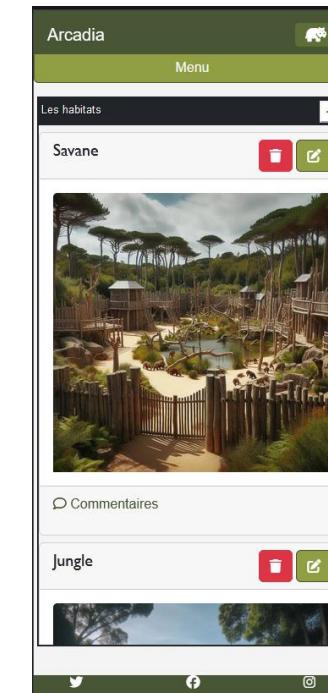
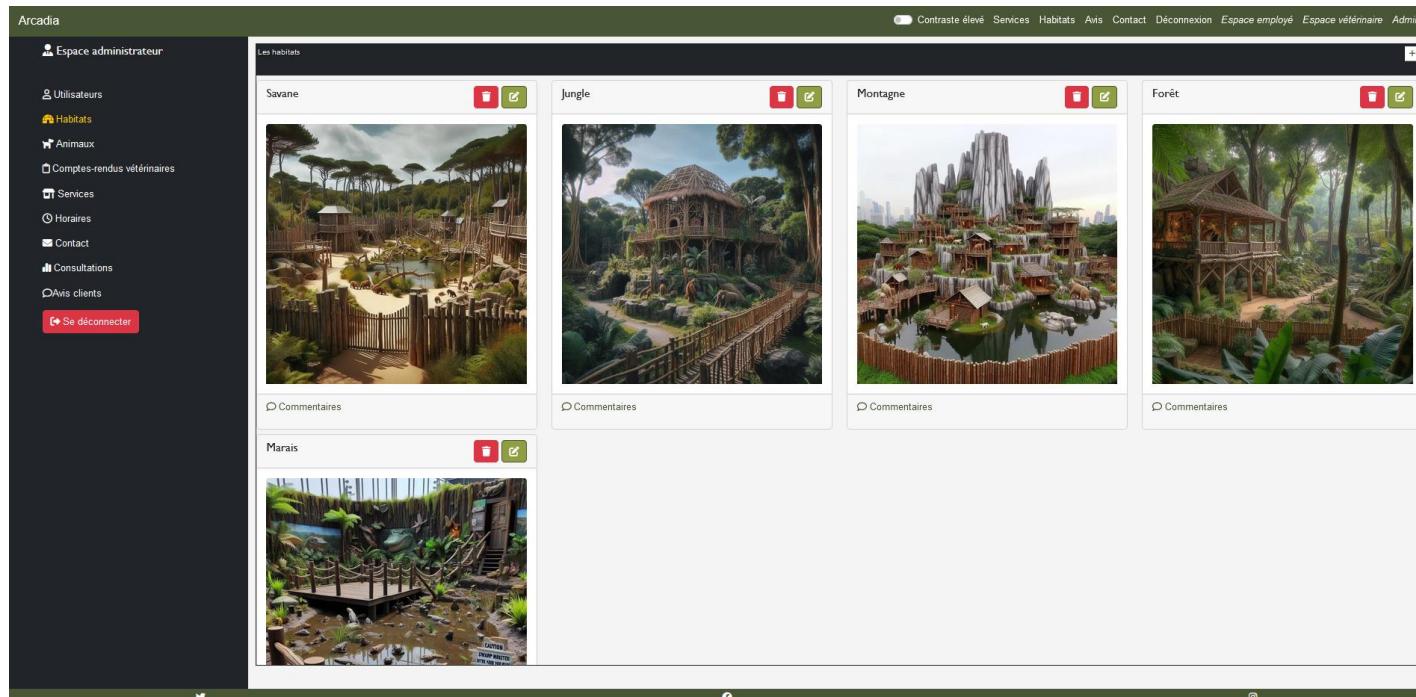


Bases de données :



# Bootstrap : Style, mise en page & responsive

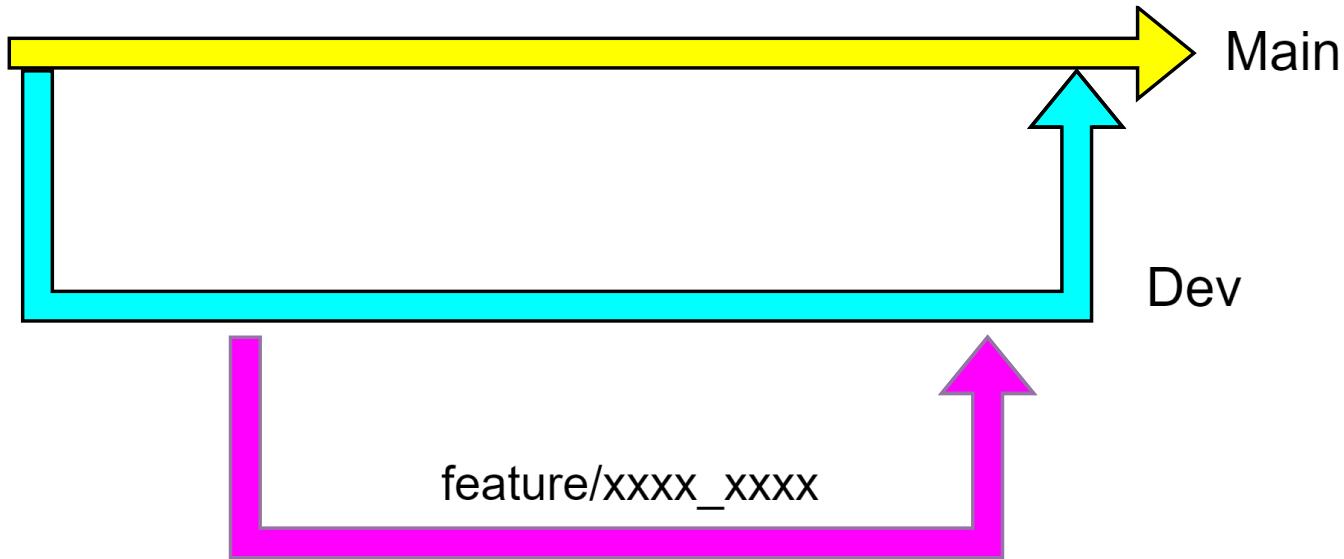
- ▶ Classes CSS préconstruites
- ▶ Mise en page utilisant grille 12 colonnes
- ▶ S'adapte aux différents écrans



# Gestion de projet : Trello

Backlog	Do	Doing	Done	Production
Fonctionnalités à développer	Prochaines fonctionnalités à être développées	Développement de la fonctionnalité en cours	Le développement de la fonctionnalité est terminé et son intégration a été testé.	La nouvelle fonctionnalité est en production

# Git & GitHub :



- Main : Version de l'application en production
- Dev: Branche de développement
- Branche feature : Fonctionnalité en développement

# **Charte graphique, maquettes et wireframes**

*Figma*

# Charte graphique :

Permet de définir les éléments de l'identité visuelle de l'entreprise sur les différents supports de communication & garantit sa cohérence.

## CHARTE GRAPHIQUE PROJET ARCADIA



PRIMARY

RGB(144,159,67)



SECONDARY

RGB(95,11,58)



DARK

RGB(73,86,53)



DARKER

RGB(19,29,0)



WHITE

RGB(255,255,255)



ACTIVE

RGB(205,203,0)



TITRE

Open Sans



LOGO NOIR



LOGO BLANC



BTN-PRIMARY



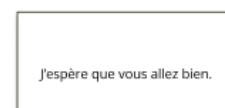
BTN-SECONDARY



BTN-DANGER



BTN-INFO

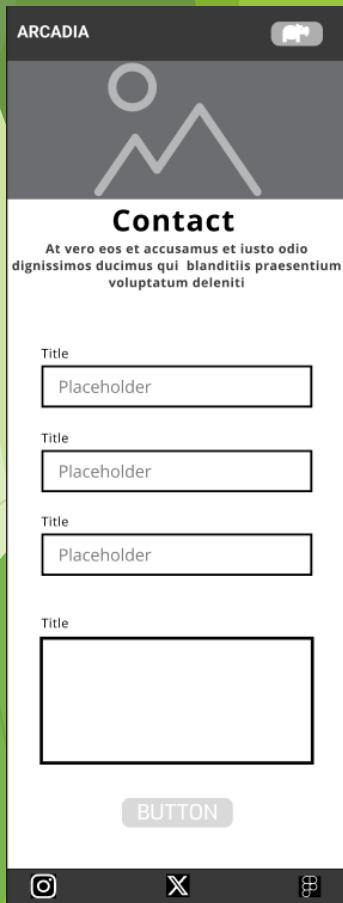
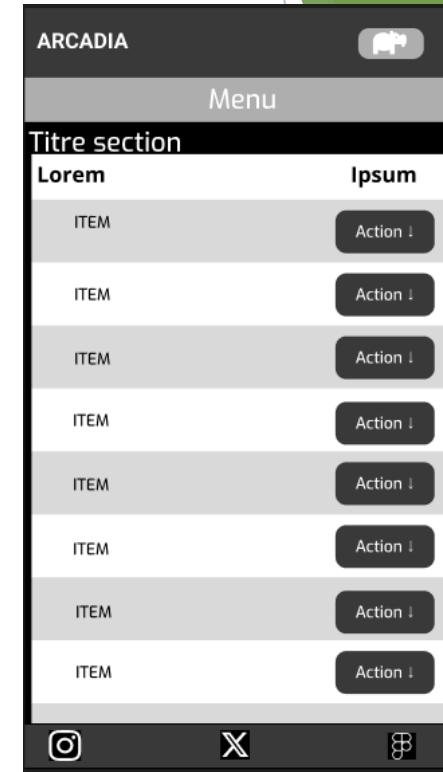
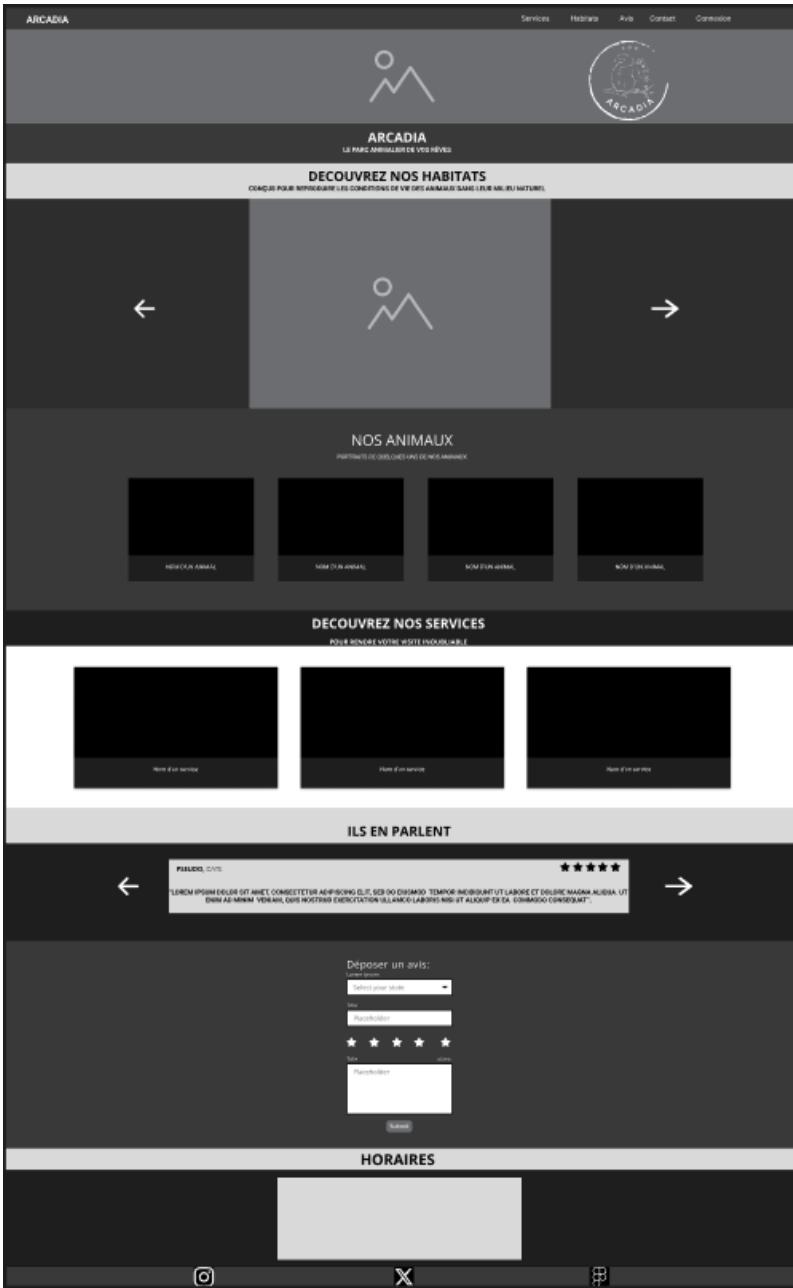


J'espère que vous allez bien.

TEXTE  
Open Sans

# Wireframes :

Recherche d'ergonomie  
Disposition du contenu  
Rôle et position des zones  
Hiérarchisation du contenu



# Maquettes :

Intègre couleurs et identité de marque  
Représentation réaliste de l'interface

The image displays three wireframe prototypes for the Arcadia zoo, showcasing different views of the website and mobile application.

- Website Mockup:** This full-page wireframe shows the homepage of the Arcadia website. It features a header with the "ARCADIA" logo and navigation links for Accès, Habitats, Ans, Garder, and Commandez. The main content includes a banner with two zebras, a stamp-like graphic, and the text "Arcadia Le parc animalier de vos rêves". Below this is a section titled "Découvrez nos habitats" with a sub-section "Savane". A central image shows a savanna landscape with animals. To the left and right are arrows indicating a scrollable area. Further down, there's a "Nos animaux" section with four animal portraits (Lion, Girafe, Tigre, Panthère noire) and their names. The bottom section, "Découvrez nos services", lists "Restaurant", "Balade en petit train", and "Visite guidée des habitats" with corresponding descriptions and icons. A footer at the bottom includes social media icons for Instagram, X (Twitter), and Facebook.
- Mobile App Mockup - Menu Screen:** This vertical wireframe shows the mobile application's menu screen. The top bar has the "ARCADIA" logo and a menu icon. The main content area is titled "Services" and lists "Service" and "Action" columns for "Restaurant", "Balade en petit train", and "Visite guidée des habitats", each with an "Action" button. At the bottom are social media icons for Instagram, X (Twitter), and Facebook.
- Mobile App Mockup - Contact Form Screen:** This vertical wireframe shows a contact form screen. The top bar has the "ARCADIA" logo and a menu icon. The main content asks "Ma demande concerne le zoo :" and provides a dropdown menu with "Arcadia". It includes fields for "Titre" (Placeholder), "Mail" (example@mail.com), and "Message". At the bottom are social media icons for Instagram, X (Twitter), and Facebook.

# **Conception de la base de données**

*Looping*

# Modèle Conceptuel de données:

- ▶ Représentation graphique des entités et associations.
- ▶ Identifier & structurer données
- ▶ Comprendre relations

## Éléments:

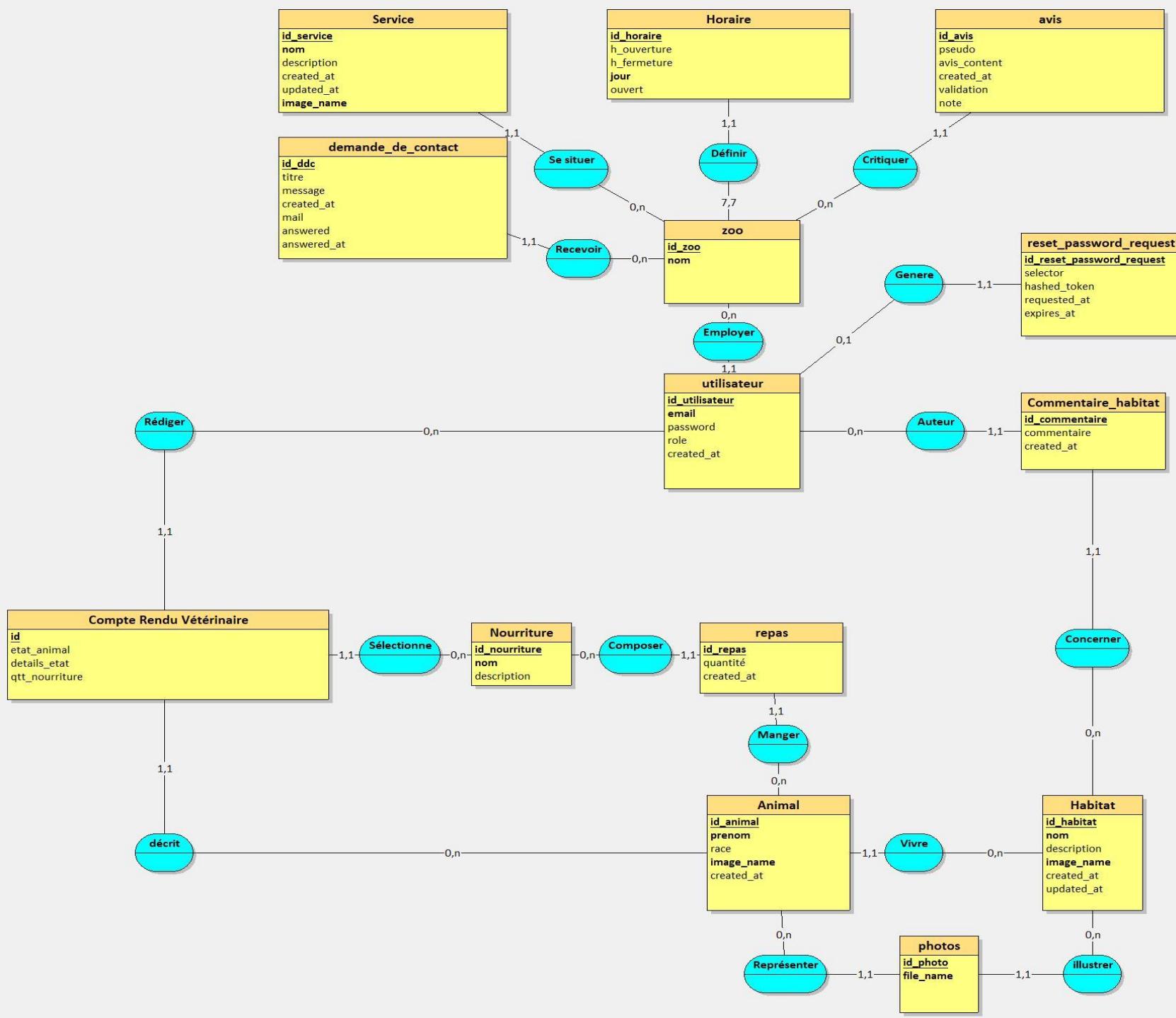
Entités

Attributs

Association: Relations entre les entités

Cardinalités: Min & Max liaisons entre entités

# MCD



# Création de la base de données:

creation\_db.sql :

```
1 CREATE DATABASE IF NOT EXISTS arcadia_db CHARACTER SET utf8 COLLATE utf8_general_ci;
2
3 USE arcadia_db;
4
5 CREATE TABLE Zoo (
6     id INT AUTO_INCREMENT PRIMARY KEY,
7     nom VARCHAR(255)
8 );
9
10 > CREATE TABLE Animal (...);
11
12 > CREATE TABLE Avis (...);
13
14
15 CREATE TABLE commentaire_habitat [
16     id INT AUTO_INCREMENT PRIMARY KEY,
17     commentaire TEXT,
18     habitat_id INT,
19     created_at DATETIME,
20     updated_at DATETIME NULL,
21     auteur_id INT
22 ];
23
24
25 > CREATE TABLE demande_contact (...);
26
27
28 CREATE TABLE Habitat (
29     id INT AUTO_INCREMENT PRIMARY KEY,
30     nom VARCHAR(255),
31     image_name VARCHAR(255) NULL,
32     description VARCHAR(512),
33     updated_at DATETIME NULL
34 );
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142 ALTER TABLE commentaire_habitat ADD CONSTRAINT FK_habitat_id_for_commentaire FOREIGN KEY (habitat_id) REFERENCES Habitat(id);
143 ALTER TABLE commentaire_habitat ADD CONSTRAINT FK_auteur_id_for_commentaire FOREIGN KEY (auteur_id) REFERENCES Users(id) ON DELETE SET NULL;
144
```

- Création base de données
- Création des tables
- Ajout des contraintes

# **Création des entités :**

*Symfony Maker Bundle*

# Création de la classe ‘avis’ :

```
PS C:\Workspace\Arcadia-Copy> symfony console make:entity

Class name of the entity to create or update (e.g. BraveKangaroo):
> review

created: src/Entity/Review.php
created: src/Repository/ReviewRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> []
```

# Clé étrangère ‘id\_zoo’:

- Nom du champ
- Type de donnée
- Classe liée
- Not null.

```
New property name (press <return> to stop adding fields):  
> id_zoo  
  
Field type (enter ? to see all types) [string]:  
> ManyToOne  
  
What class should this entity be related to?:  
> zoo  
  
Is the Review.id_zoo property allowed to be null (nullable)? (yes/no) [yes]:  
> no
```

*Many To One*

- Plusieurs avis peuvent concerner le même zoo

# **Dynamisation de l'interface :**

Dashboard admin

# Dashboard admin :

Arcadia

Espace administrateur

Utilisateurs Habitats Animaux Comptes-rendus vétérinaires Services Horaires Contact Consultations Avis clients

Se déconnecter

Contraste élevé Services Habitats Avis Contact Déconnexion Espace employé Espace vétérinaire Admin

Les animaux

Alex	Marty	Melman	Scrat
Alex	Marty	Melman	Scrat
Billy	Sirius	Moto-Moto	Gloria
Lucie	Gia	René	

Arcadia

Espace administrateur

Utilisateurs Habitats Animaux Comptes-rendus vétérinaires Services Horaires Contact Consultations Avis clients

Se déconnecter

Les animaux

Erreurs lors de la récupération des animaux

- Une page
- Chargement dynamique du contenu
- Requêtes asynchrones

# Récupération des animaux :

Controller PHP:

- Utilisation méthode repository
- Sérialisation JSON avec contexte
- Gestion Erreur
- Accepte GET, Renvoie JSON 200 / 500

```
#[IsGranted('ROLE_USER')]
#[Route('/all', name: 'getAnimals', methods: ['GET'])]
public function getAnimals(AnimalRepository $animalRepository): JsonResponse
{
    try{
        $animals = $animalRepository->findAll();
        $context = ['groups' => 'animal:read'];
        return JsonResponse::fromJsonString($this->serializer->serialize(
            $animals, 'json', $context),
            Response::HTTP_OK);

    } catch (\Exception $e) {
        return new JsonResponse([
            'error' => 'An error occurred',
            'message' => $e->getMessage()
        ], Response::HTTP_INTERNAL_SERVER_ERROR);
    }
}
```

Extrait de  
AnimalController.php

# Récupération des animaux :

Javascript - Requête asynchrone

```
4  async function getAnimals(){
5    let myHeaders = new Headers();
6    myHeaders.append('Content-Type', 'application/json');
7    let requestOptions = {
8      method: 'GET',
9      headers: myHeaders,
10     redirect: 'follow'
11   };
12   try {
13     let response = await fetch('animal/all', requestOptions);
14     if (!response.ok) {
15       throw new Error();
16     }
17     let result = await response.json();
18     let animals = result;
19     let animalsList = document.getElementById('animalsList');
20     while (animalsList.firstChild) {
21       animalsList.removeChild(animalsList.firstChild);
22     }
23
24     // Cr ation carte pour chaque animal
25     animals.forEach(animal => {
26       ...
27     });
28   }
29   catch (error) {
30     let errorMessage = document.createElement('div');
31     errorMessage.classList.add('alert', 'alert-danger');
32     errorMessage.textContent = 'Erreur lors de la r cup ration des animaux';
33     document.getElementById('animalsList').appendChild(errorMessage);
34   }
35 }
```

- Cr ation objet Headers
- D efinition options requ te
- Envoi Fetch asynchrone
- V rification r ponse
- Conversion r ponse json
- Vider l ment HTML  
animalsList

Extraits de allAnimals.js

# Récupération des animaux :

Javascript - Modification du DOM :

```
// Crédation carte pour chaque animal
animals.forEach(animal => {
  // Creation carte
  let card = document.createElement('div');
  card.classList.add('card', 'col-lg-3', 'col-md-12', 'col-sm-12', 'mt-1');

  // Header de la carte
  let cardHeader = document.createElement('div');
  cardHeader.classList.add('card-header');
  let cardTitle = document.createElement('h5');
  cardTitle.classList.add('card-title');
  cardTitle.textContent = animal.prenom;
  cardHeader.appendChild(cardTitle);

  // Body de la carte
  let cardBody = document.createElement('div');
  cardBody.classList.add('card-body');
  let cardImg = document.createElement('img');
  cardImg.src = `/images/animal/${animal.imageName}`;
  cardImg.classList.add('card-img-top', 'animal-img');
  cardImg.alt = `photo de ${animal.prenom}`;
  cardBody.appendChild(cardImg);

  // Footer & Dropdown de la carte
  let cardFooter = document.createElement('div');
  cardFooter.classList.add('card-footer');

  let dropdown = document.createElement('div');
  dropdown.classList.add('dropdown');
```

Extraits de allAnimals.js

animals.forEach :

- ▶ Crédation éléments HTML
- ▶ Ajout classes (Bootstrap)
- ▶ Pas de innerHtml

```
// Append header, body, and footer to card
card.appendChild(cardHeader);
card.appendChild(cardBody);
card.appendChild(cardFooter);

// Append card to animalsList
animalsList.appendChild(card);
```

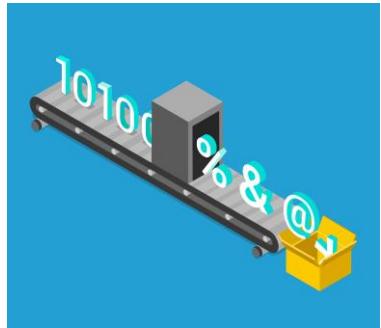
# **Cycle de vie d'une fonctionnalité :**

# La fonctionnalité:

Chiffrement emails lors du dépôt d'une demande de contact



**Qui ?**  
*Les visiteurs du site*



**Comment ?**  
*Création d'un service de chiffrement*



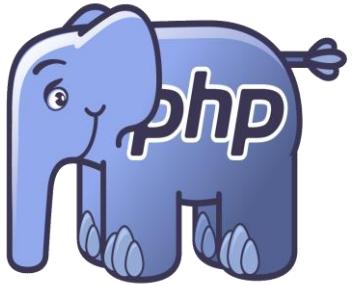
**Pourquoi ?**  
*Renforcer la protection des données  
des utilisateurs de l'application.*

# Etapes :

- ▶ Conception : Choix des technos, recueil informations exigences sécurité.
- ▶ Développement du service
- ▶ Développement des tests
- ▶ Intégration
- ▶ Déploiement

# Spécifications techniques & conception :

Documentation PHP, ANSSI & CNIL



php Downloads Documentation Get Involved Help php 2.3

PHP Manual > Function Reference > Cryptography Extensions > OpenSSL

OpenSSL Functions

Table of Contents

- [openssl\\_cipher\\_iv\\_length](#) — Gets the cipher iv length
- [openssl\\_cipher\\_key\\_length](#) — Gets the cipher key length
- [openssl\\_cms\\_decrypt](#) — Decrypt a CMS message
- [openssl\\_cms\\_encrypt](#) — Encrypt a CMS message
- [openssl\\_cms\\_read](#) — Export the CMS file to an array of PEM certificates
- [openssl\\_cms\\_sign](#) — Sign a file
- [openssl\\_cms\\_verify](#) — Verify a CMS signature
- [openssl\\_csr\\_export](#) — Exports a CSR as a string
- [openssl\\_csr\\_export\\_to\\_file](#) — Exports a CSR to a file
- [openssl\\_csr\\_get\\_public\\_key](#) — Returns the public key of a CSR
- [openssl\\_csr\\_get\\_subject](#) — Returns the subject of a CSR
- [openssl\\_csr\\_new](#) — Generates a CSR
- [openssl\\_csr\\_sign](#) — Sign a CSR with another certificate (or itself) and generate a certificate
- [openssl\\_decrypt](#) — Decrypts data
- [openssl\\_dh\\_compute\\_key](#) — Computes shared secret for public value of remote DH public key and local DH key
- [openssl\\_digest](#) — Computes a digest
- [openssl\\_encrypt](#) — Encrypts data

## Mécanisme conforme

- Le mode de chiffrement CBC utilisant une primitive de chiffrement conforme au référentiel comme l'AES et des valeurs initiales aléatoirement choisies pour chaque message et transmises en clair est un mécanisme de chiffrement symétrique conforme au référentiel.



**CNIL.**

## Utiliser les tailles de clés suffisantes

- pour AES, les clés de 128, 192 ou 256 bits sont considérées comme suffisantes ;
- pour les algorithmes basés sur RSA, il est recommandé d'utiliser des modules et exposants secrets d'au moins 2 048 bits ou 3 072 bits, avec des exposants publics, pour le chiffrement, supérieurs à 65 536 bits

# Développement :

## EncryptionService.php :

```
1  <?php
2  namespace App\Service;
3
4  class EncryptionService {
5
6      private $cipher = 'AES-128-CBC'; // Algorithme de chiffrement
7      private $key; // Clé de chiffrement à partir du .env(.env.local)
8      private $iv ; // 'Vecteur d'initialisation'
9
10
11     // Récupérer la clé à l'instanciation de la classe
12     public function __construct(){
13         $this->key = $_ENV[ 'APP_KEY'];
14     }
15 }
```

# Chiffrer et déchiffrer :

Méthodes de la classe EncryptionService :

```
15     // Chiffrer
16     public function encrypt($data){
17         $iv = random_bytes(openssl_cipher_iv_length($this->cipher));
18         // IV aléatoire taille appropriée pour l'algorithme
19         $encryptedData = openssl_encrypt($data, $this->cipher, $this->key, 0, $iv);
20         // Chiffrer en utilisant $données, algo, clé, options, IV
21         return base64_encode($iv . $encryptedData);
22         // Encodage en base64 de l'IV et des données chiffrées
23     }
24 }
25 }
```

```
// Déchiffrer
public function decrypt($encryptedData) {
    $data = base64_decode($encryptedData);
    // Décoder depuis base64
    $ivLength = openssl_cipher_iv_length($this->cipher);
    // Récupérer la taille de l'IV
    $iv = substr($data, 0, $ivLength);
    // Récupérer l'IV
    $ciphertext = substr($data, $ivLength);
    // Récupérer les données chiffrées
    return openssl_decrypt($ciphertext, $this->cipher, $this->key, 0, $iv);
    // Déchiffrer en utilisant les données chiffrées, l'algo, la clé, les options, l'IV
}
```

# Développement des tests:

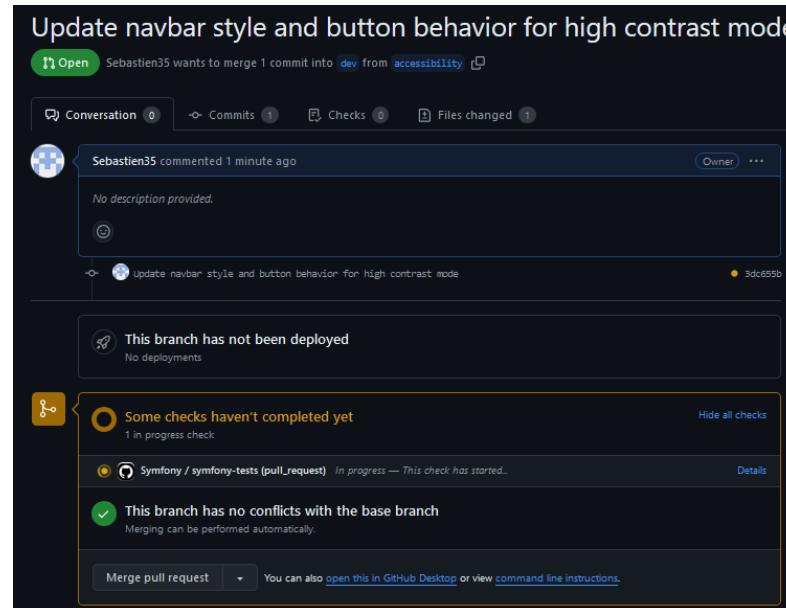
## PHPUnit

```
1 <?php
2 namespace App\Tests;
3
4 use App\Service\EncryptionService;
5 use PHPUnit\Framework\TestCase;
6
7 class EncryptionServiceTest extends TestCase
8 {
9     private $encryptionService;
10
11     protected function setUp(): void
12     {
13         $this->encryptionService = new EncryptionService();
14     }
15
16     public function testEncryptDecrypt()
17     {
18         $data = 'Hello World';
19         $encrypted = $this->encryptionService->encrypt($data);
20         $encrypted2 = $this->encryptionService->encrypt($data);
21
22         $this->assertNotEquals($data, $encrypted);
23         $this->assertNotEquals($encrypted, $encrypted2);
24
25         $decrypted = $this->encryptionService->decrypt($encrypted);
26         $decrypted2 = $this->encryptionService->decrypt($encrypted2);
27
28         $this->assertEquals($data, $decrypted);
29         $this->assertEquals($decrypted, $decrypted2);
30     }
31 }
```

- ▶ Extends TestCase pour utiliser les méthodes de PHPUnit.
- ▶ setUp() : initialise \$encryptionService avec nouvelle instance classe.
- ▶ Assertions vérifient bon fonctionnement chiffrement déchiffrement en validant résultat distincts et intégrité des données

# Intégration

## Exécution automatisée des tests via un workflow



```
- name: Start MongoDB
  uses: supercharge/mongodb-github-action@1.10.0
  with:
    mongodb-version: '4.4'
    mongodb-port: 27017

- name: Create Database
  run:
    - mkdir -p data
    - touch data/database.sqlite

- name: Migrate Database
  env:
    DATABASE_URL: sqlite:///kernel.project_dir%/data/database.sqlite
  run:
    - php bin/console doctrine:database:create --no-interaction
    - php bin/console doctrine:schema:create --no-interaction

- name: Execute tests (Unit and Feature tests) via PHPUnit
  env:
    DATABASE_URL: sqlite:///kernel.project_dir%/data/database.sqlite
  run:
    - php bin/console doctrine:schema:create --env=test --no-interaction
    - vendor/bin/phpunit
```

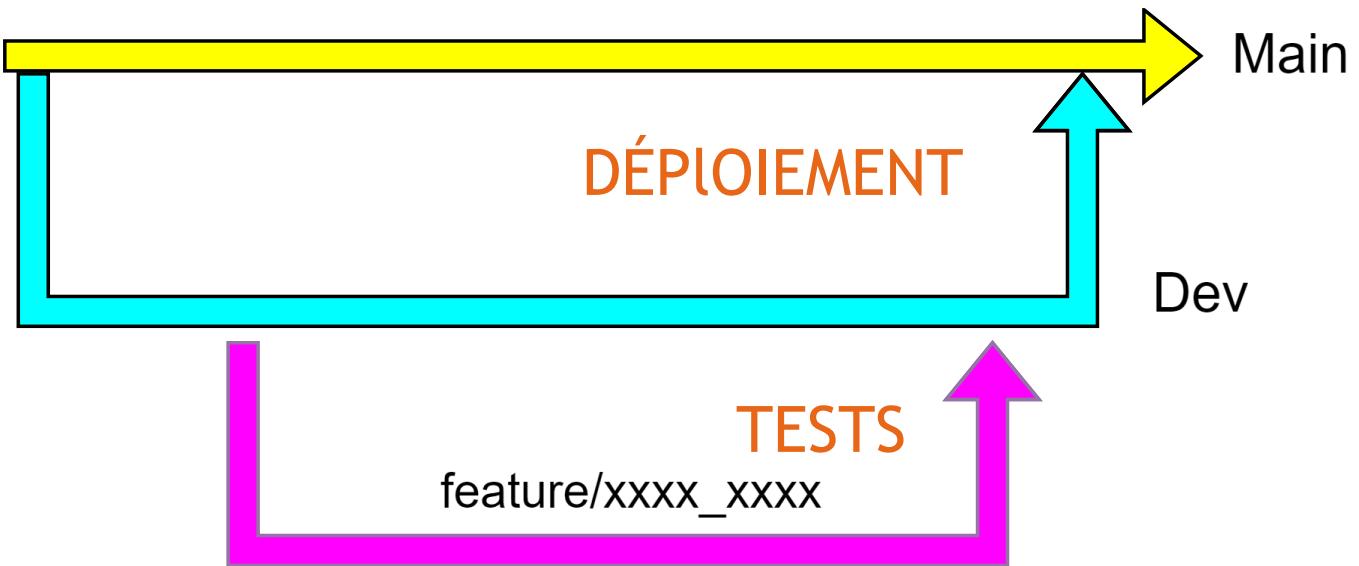
# Déploiement

Déploiement automatisé via un workflow GitHub

```
7 jobs:
8   deploy:
9     runs-on: ubuntu-latest
10
11   env:
12     SSHPASS: ${{ secrets.SSHPASS }}
13
14   steps:
15     - name: Checkout repository
16       uses: actions/checkout@v2
17
18     - name: Install sshpass & update apt
19       run:
20         |
21         sudo apt update
22         sudo apt install sshpass
23
24     - name: Deploy application
25       run:
26         |
27         sshpass -e ssh -o strictHostKeyChecking=no ${{ secrets.SSH_USER }}@${{ secrets.SSH_IP }} "
28           cd htdocs &&
29           cd arcadia-zoo.online/ &&
30           cd Arcadia &&
31           git pull https://github.com/Sebastien35/Arcadia.git &&
32           composer install
33         "
```

- SSH vers répertoire de l'app
- Git pull
- Installation dépendances avec composer

The screenshot shows a GitHub Actions workflow run for a merge pull request. The main header indicates a successful merge of pull request #114 from Sebastien35/dev into #138. Below this, the "Summary" section shows a single job named "deploy" which succeeded. The "Workflow file" tab is selected, showing the YAML code for the workflow. The "Run details" section provides a summary of the job's execution, listing the steps: Set up job, Checkout repository, Install sshpass & update apt, Deploy application, Post Checkout repository, and Complete job. Each step is marked as successful.



# Mécanismes de sécurité

# Sécurité :

## Symfony security:

- ▶ Protection des routes (/veterinaire, /employe, /admin)
- ▶ Nombre de tentatives de connexion (*brute force*)
- ▶ Hash des mots de passe en bdd (Bcrypt) (Salage automatique)
- ▶ Utilisation de mot de passe forts
- ▶ Jetons CSRF sur formulaires Symfony
- ▶ Mime Type

## Anti XSS :

- ▶ Validation & Nettoyage des entrées utilisateurs (purifier)

## Chiffrement des emails :

- ▶ Open ssl, AES-128-CBC

## SSL / TLS :

- ▶ Chiffrement des données client <-> application (**HTTPS** forcé par Nginx)

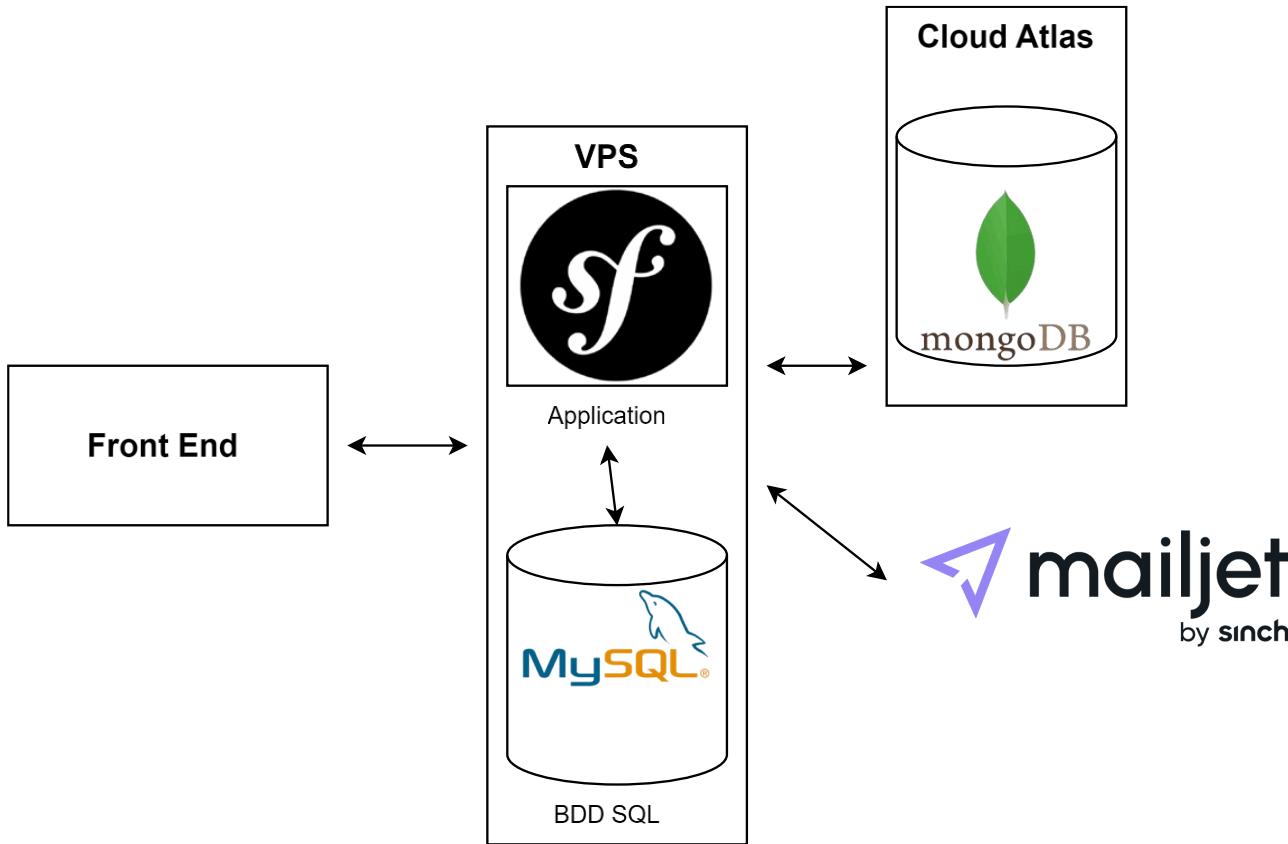
## Injection SQL:

- ▶ Méthodes natives de Doctrine ou méthodes personnalisées sécurisées.

## Sécurisation des configurations:

- ▶ **.env, .env.local & GitHub Secrets.**

# Architecture :



# Conteneurisation pour un environnement de développement :





- ▶ Consistance de l'environnement
- ▶ Facilité de configuration
- ▶ Reproductibilité
- ▶ Facilité de collaboration



- ▶ Image de l'application : Dockerfile
- ▶ Orchestration des conteneurs: Docker-compose

# Conteneurisation pour un environnement de développement:

## Dockerfile

```
1  FROM php:8.2-fpm
2
3  RUN apt update && apt install -y \
4      nano \
5      libpng-dev \
6      libjpeg-dev \
7      libzip-dev \
8      libfreetype6-dev \
9      libonig-dev \
10     libxml2-dev \
11     unzip \
12     build-essential \
13     git \
14     nginx \
15     libssl-dev \
16     default-mysql-client \
17     && docker-php-ext-configure gd --with-freetype --with-jpeg \
18     && docker-php-ext-install -j$(nproc) gd \
19     && docker-php-ext-enable opcache \
20     && docker-php-ext-install xsl \
21     && docker-php-ext-install zip \
22     && docker-php-ext-install intl \
23     && docker-php-ext-install soap \
24     && pecl install mongodb \
25     && docker-php-ext-enable mongodb
26
27 RUN docker-php-ext-install pdo_mysql \
28     && docker-php-ext-enable pdo_mysql
29
30 COPY --from=composer:latest /usr/bin/composer /usr/bin/composer
31 COPY docker/php/conf.d/* /usr/local/etc/php/conf.d/
32
33 COPY ./docker/nginx/default.conf /etc/nginx/conf.d/default.conf
34
35 WORKDIR /var/www
36
37 COPY . .
38
39 ENV COMPOSER_ALLOW_SUPERUSER 1
40
41 RUN composer install
42
43 EXPOSE 9000
44 CMD ["php-fpm"]
```

- Image de base
- Installation dépendances
- Extension PHP
- Installation Composer
- Configuration Nginx
- Copie code application
- Expose 9000
- Lance serveur php-fpm

# Conteneurisation pour un environnement de développement:

## Docker-compose

```
1  version: '3.8'
2
3  services:
4    app:
5      build:
6        context: .
7        dockerfile: Dockerfile
8        container_name: symfony_app
9        environment:
10          - DATABASE_URL=mysql://root:docker1234@mysql:3306/arcadia_db
11          - MONGODB_URL=mongodb://mongo:27017
12    > volumes:...
13    depends_on:
14      - mysql
15      - mongo
16
17    nginx:
18      image: nginx:alpine
19      container_name: nginx
20      restart: unless-stopped
21      volumes:
22        - ./docker/nginx/default.conf:/etc/nginx/conf.d/default.conf
23        - ./var/www/html
24      ports:
25        - "9000:80"
26      depends_on:
27        - app
28
29    > mysql:...
30    > phpmyadmin:...
31    > mongo:...
32
33    volumes:
34      mysql_data:
35      mongo_data:
36
```

### Création différents conteneurs:

- ▶ Application Symfony
- ▶ Base de données MySQL
- ▶ Base de données MongoDB
- ▶ PhpMyAdmin
- ▶ Serveur Nginx

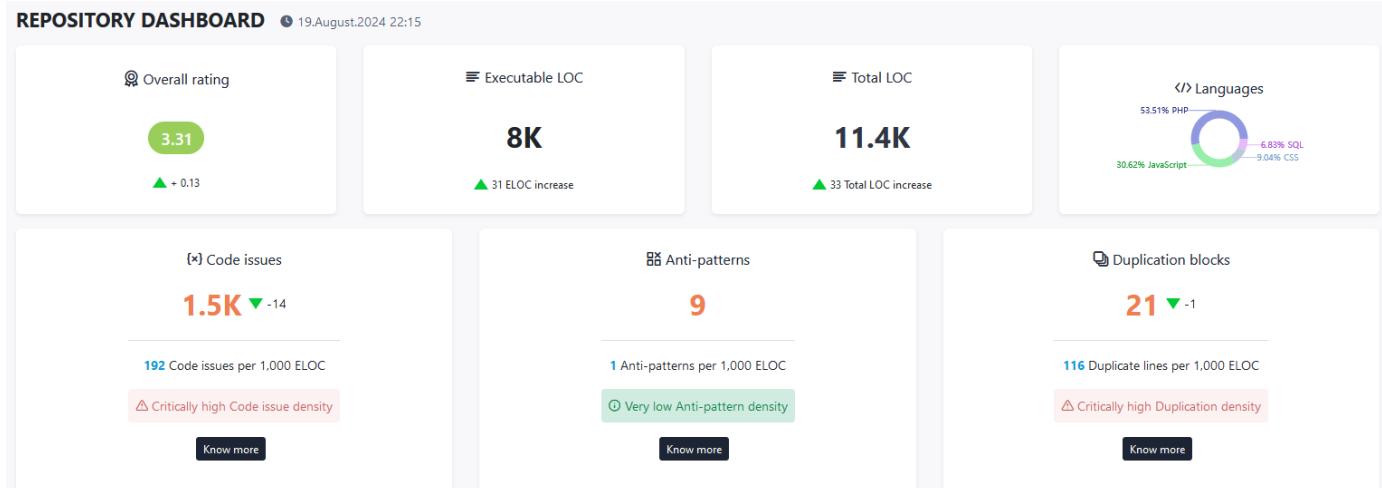
# Règlement Général sur la Protection des Données

- ▶ Les données traitées :
  - ▶ Objectif du traitement
  - ▶ Qui y a accès ?
  - ▶ Durée de conservation
- ▶ Respect du droit des personnes :
  - ▶ Information des personnes
  - ▶ Fondement juridique
  - ▶ Comment exercer droits sur les données (email)
- ▶ Sécurisation des données :
  - ▶ Chiffrement des emails
  - ▶ Utilisation pseudo pour les avis

# Vérification de la qualité du code:

Embold.io par browserstack

- ▶ Interface intuitive
- ▶ Complément aux linters de l'IDE
- ▶ Vue d'ensemble



Exemple d'erreur :  
(code inatteignable)

no-unreachable

**Issue type**: Code issue | **Criticality**: High | **Code checker**: ESLint

**Tags**: No tags | **KPIs**: Efficiency | **Issue key**: ESLINT-JS-158

**File (Line number)**

employee.js (218) [View code](#)

**Your code**

```
1 throw new Error('Erreur');
2 window.location.reload();
3 }
```

**Description**

Unreachable code. (no-unreachable)

Because the return, throw, break, and continue statements unconditionally exit a block of code, any statements after them cannot be executed. Unreachable statements are usually a mistake.

# Veille technologique :

feedly

The screenshot shows the Feedly mobile application interface. At the top, there's a navigation bar with icons for Today, Read Later, Feedly AI, and Integrations & API. A blue "Upgrade" button is located at the top right. The main title "Programming" is centered above a "MOST POPULAR" section. The "FEEDS" sidebar on the left lists categories like All, Programming, and various news sources. The "PROGRAMMING" category is selected. Below the sidebar, there's a "BOARDS" section with a placeholder board for saving articles. The "MOST POPULAR" section displays several articles:

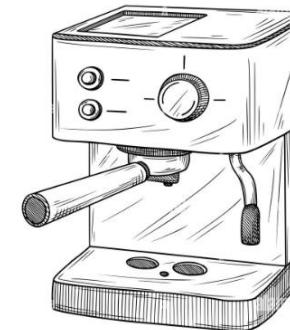
- Gérer l'état avec Pinia** by Grafikart.fr (published 2 hours ago). Description: Aujourd'hui, nous allons explorer ensemble Pinia, un gestionnaire d'État global pour Vue.js. Même si vous pouvez créer des valeurs...
- PHP 8.4.0 Beta 3 now available for testing - PHP: Hypertext Preprocessor** by Planet PHP (published 6 days ago). Description: The PHP team is pleased to announce the release of PHP 8.4.0, Beta 3. This is the first beta release, continuing the PHP 8.4 release cycle, the rough outline of which is specified in the PHP Wiki. For source downloads of PHP...
- Stratégies pour une manipulation efficace de la mémoire du DOM en JavaScript moderne** by Human Coders News . JavaScript (published 16 days ago). Description: Cet article de Frontend Masters explique des méthodes modernes pour manipuler le Document Object Model (DOM) de manière efficace en mémoire en utilisant JavaScript. Il offre des techniques et astuces pour...
- TypeScript avec Vue.js** by Grafikart.fr (published 1 day ago). Description: Dans ce chapitre nous allons voir comment utiliser TypeScript dans le cadre d'un projet Vue.js. La vérification du code et la validation...

On the right side, there's a "YOU MIGHT ALSO LIKE" section featuring other feeds like "Thoughtworks Technology Podcast" and "Thoughtworks (uploads) on YouTube". A "Explore" button is located at the bottom right of the main content area.

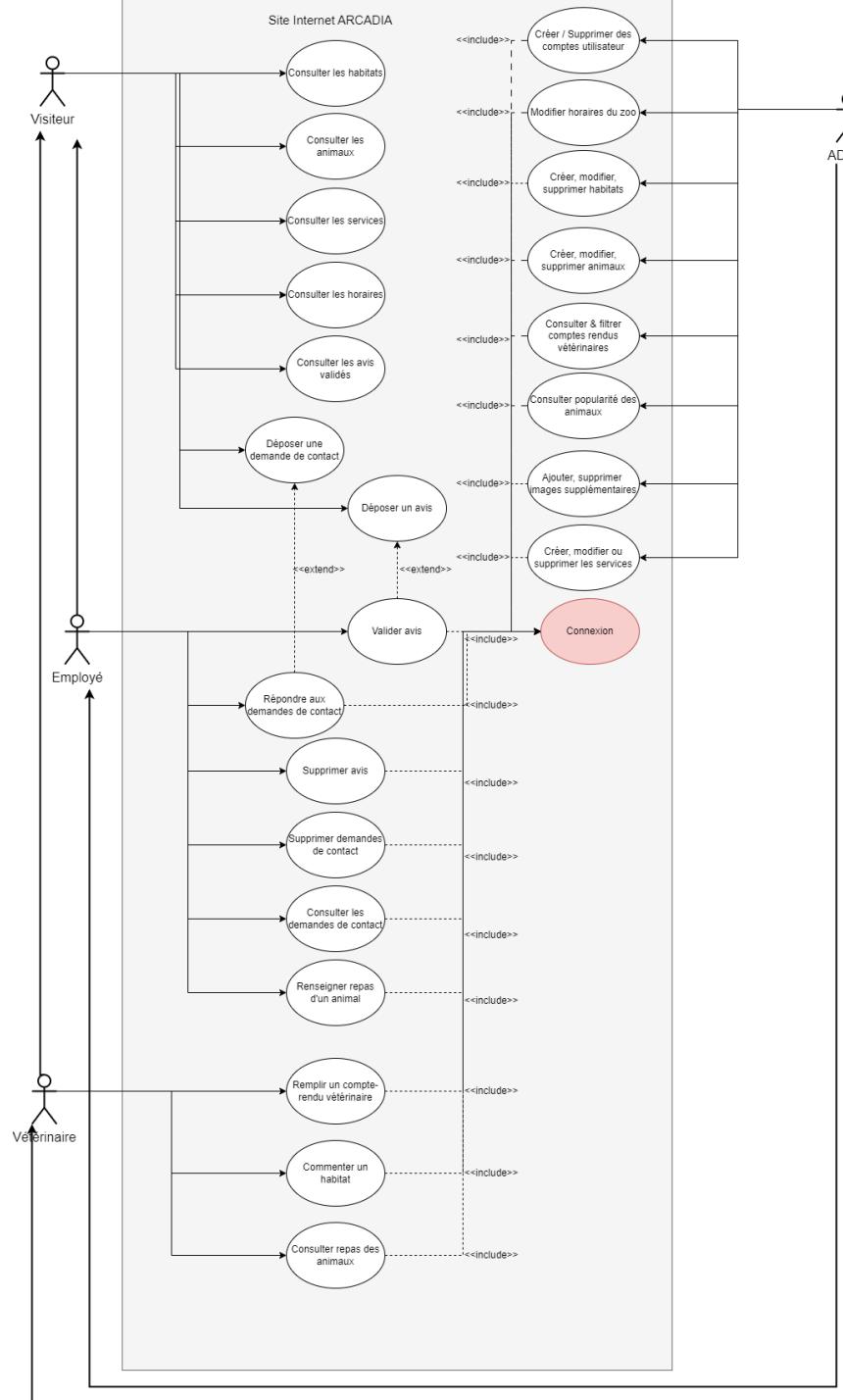
- Abonnement à des canaux
- Permet épuration du contenu

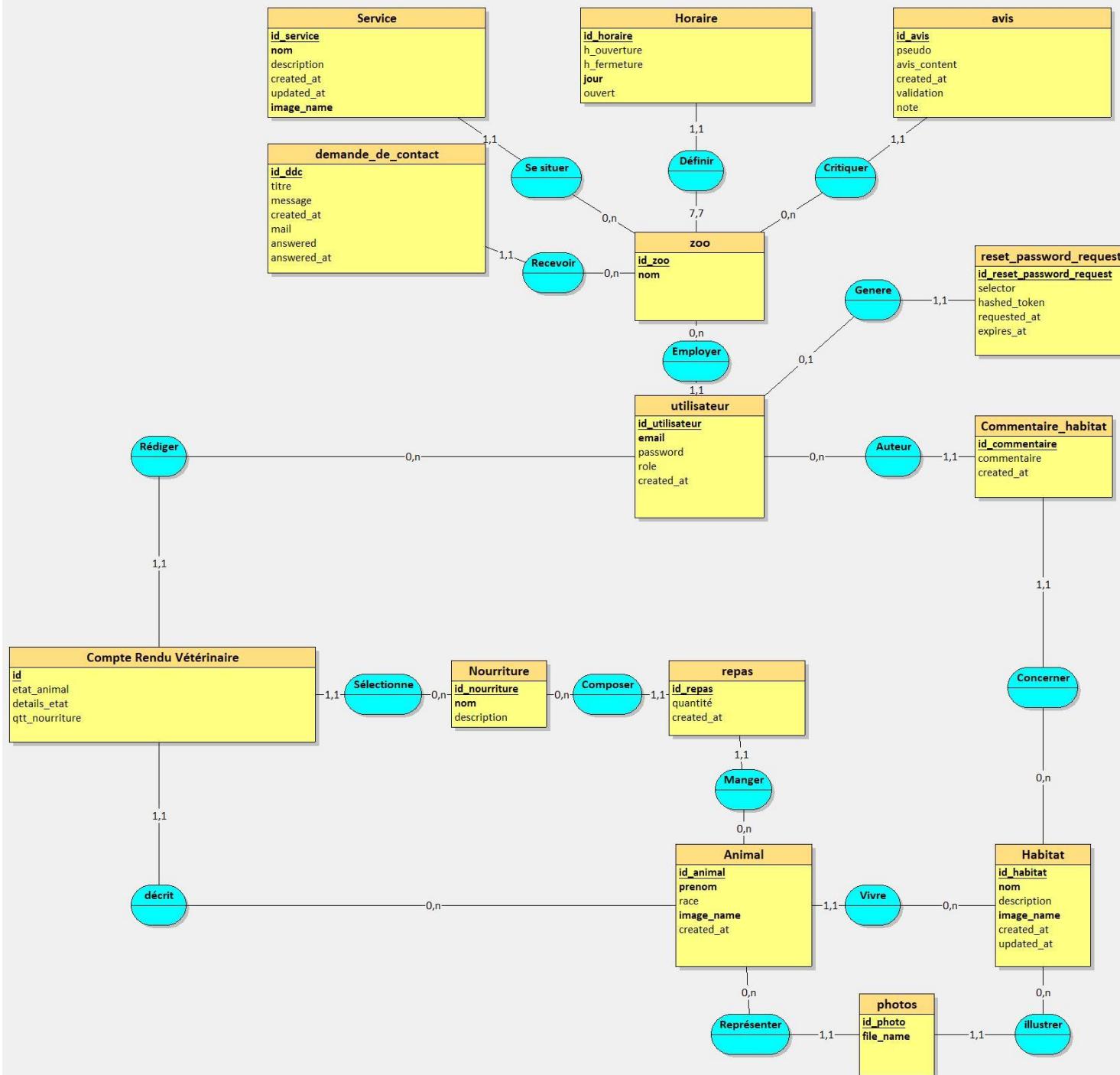
# Conclusion

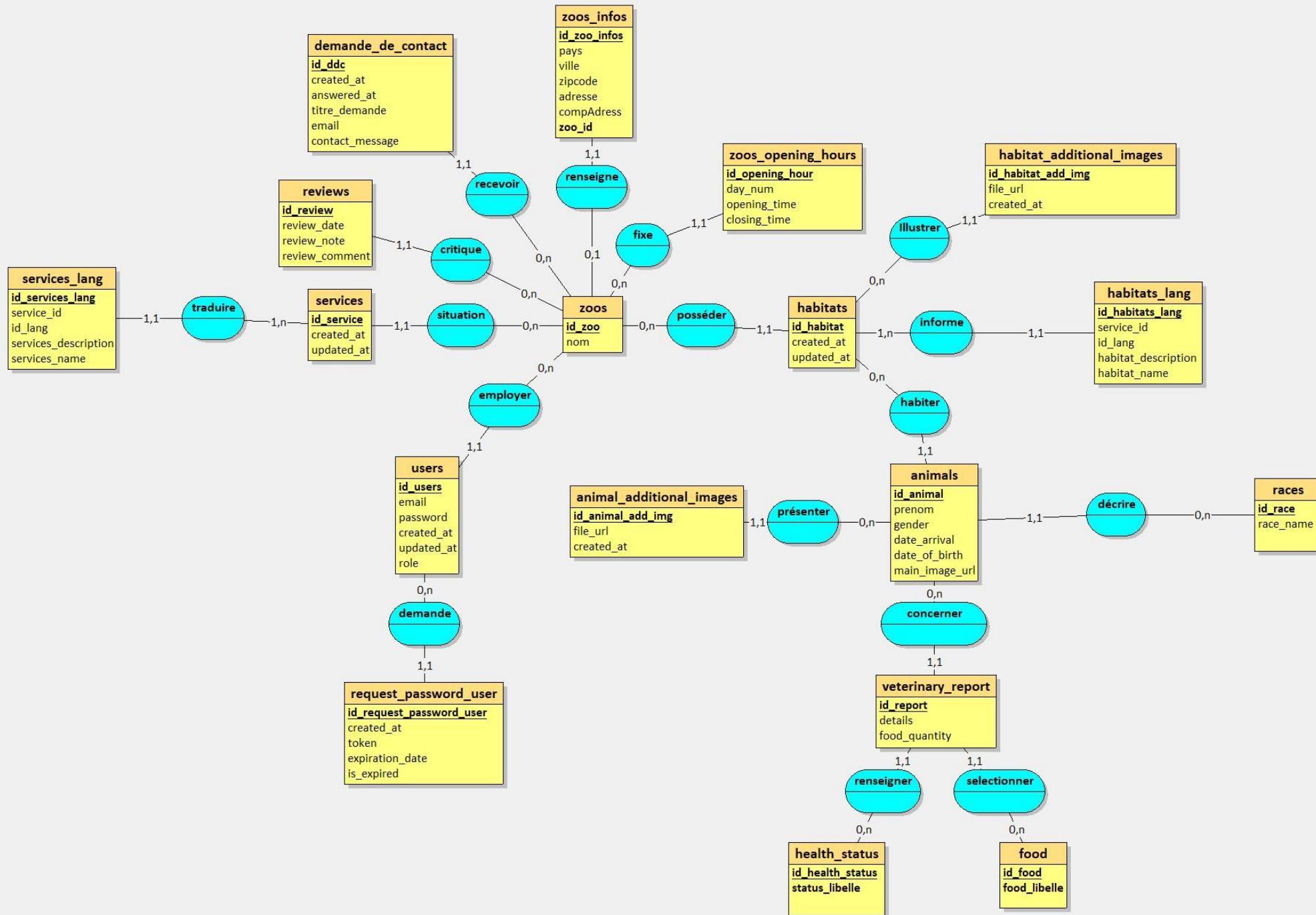
# Remerciements :



# Annexes







```
zoo = (id_zoo SMALLINT, nom VARCHAR(50));
utilisateur = (id_utilisateur INT, email VARCHAR(320), password VARCHAR(255), role VARCHAR(50), created_at DATETIME,
#zoo_id_zoo);
Nourriture = (id_nourriture INT, nom VARCHAR(50), description TEXT);
Habitat = (id_habitat INT, nom VARCHAR(50), description TEXT, image_name TEXT, created_at DATETIME, updated_at
DATETIME);
Commentaire_habitat = (id_commentaire INT, commentaire TEXT, created_at DATETIME, #Habitat_id_habitat,
#utilisateur_id_utilisateur);
demande_de_contact = (id_ddc INT, titre VARCHAR(50), message TEXT, created_at DATETIME, mail VARCHAR(320), answered
BOOLEAN, answered_at DATETIME, #zoo_id_zoo);
Service = (id_service INT, nom VARCHAR(50), description TEXT, created_at DATETIME, updated_at DATETIME, image_name
TEXT, #zoo_id_zoo);
avis = (id_avis INT, pseudo VARCHAR(50), avis_content TEXT, created_at DATETIME, validation BOOLEAN, note TINYINT,
#zoo_id_zoo);
Horaire = (id_horaire INT, h_ouverture TIME, h_fermeture TIME, jour VARCHAR(50), ouvert BOOLEAN, #zoo_id_zoo);
reset_password_request = (id_reset_password_request INT, selector VARCHAR(20), hashed_token VARCHAR(100),
requested_at DATETIME, expires_at DATETIME, #utilisateur_id_utilisateur);
Animal = (id_animal INT, prenom VARCHAR(50), race VARCHAR(50), image_name TEXT, created_at DATETIME,
#Habitat_id_habitat);
photos = (id_photo INT, file_name TEXT, #Habitat_id_habitat, #Animal_id_animal);
Compte_Rendu_Vétérinaire = (id_crv SMALLINT, etat_animal VARCHAR(50), details_etat TEXT, qtt_nourriture VARCHAR(50),
#utilisateur_id_utilisateur, #Animal_id_animal, #Nourriture_id_nourriture);
repas = (id_repas INT, quantité VARCHAR(50), created_at DATETIME, #Nourriture_id_nourriture, #Animal_id_animal);
```

```
1 <?php
2
3 namespace App\Service;
4
5 use Symfony\Component\Mailer\MailerInterface;
6 use Symfony\Component\DependencyInjection\Attribute\Autowire;
7 use Symfony\Component\Mime\Email;
8 use Symfony\Bridge\Twig\Mime\TemplatedEmail;
9
10 class MailerService
11 {
12
13     public function __construct(
14         #[Autowire('%admin_email%')] private string $adminEmail,
15         private readonly MailerInterface $mailer
16     )
17     {
18         $this->adminEmail = $adminEmail;
19     }
20
21     public function sendWelcomeEmail(string $to, array $context): void
22     {
23         $email = (new TemplatedEmail())
24             ->from($this->adminEmail)
25             ->to($to)
26             ->subject('Bienvenue chez Arcadia!')
27             ->htmlTemplate('emails/welcome_email.html.twig')
28             ->context($context);
29         $this->mailer->send($email);
30     }
31
32     public function sendResponseMail(string $to, string $text): void
33     {
34         $email = (new Email())
35             ->from($this->adminEmail)
36             ->to($to)
37             ->subject('Réponse à votre demande')
38             ->text($text);
39
40         $this->mailer->send($email);
41     }
42 }
43
44 }
```

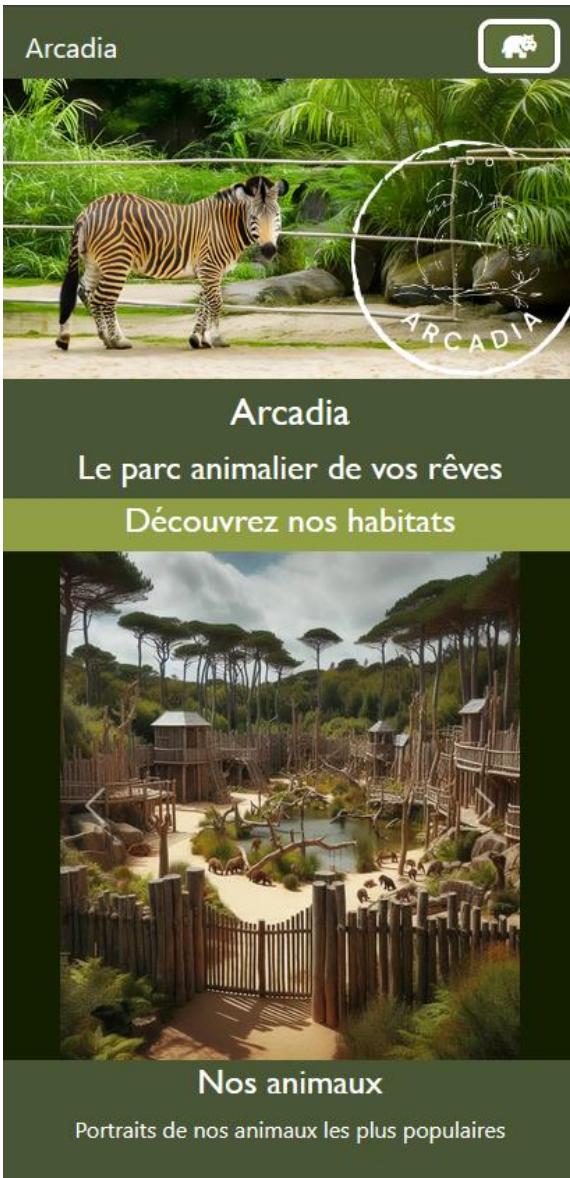
# Email d'un client (fictif) chiffré dans la base de données:

```
mysql> SELECT * FROM demande_contact WHERE id = 19;
+----+-----+-----+-----+-----+-----+
| id | titre | message | mail | created_at | answered_at | answered | zoo_id |
+----+-----+-----+-----+-----+-----+
| 19 | Test chiffrement email | Il s'agit d'un test | tUs0x9bZFERGPLEXKX7jiEhiRXp2ck4rSGN5V1NsYXFyZ3RiV2p5RzlHUiitsdGlsa1FxNXNCY0pkMwC9 | 2024-07-19 13:20:49 | NULL | 0 | 1 |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

# Sauvegardes quotidiennes de la base de données

```
/home/arcadia-zoo/backups/databases/arcadia-db
root@srv503604:/home/arcadia-zoo/backups/databases/arcadia-db# ls
2024-08-22  2024-08-23  2024-08-24  2024-08-25  2024-08-26  2024-08-27  2024-08-28  2024-08-29
root@srv503604:/home/arcadia-zoo/backups/databases/arcadia-db# cd 2024-08-29
root@srv503604:/home/arcadia-zoo/backups/databases/arcadia-db/2024-08-29# ls
arcadia-db_1724901302.sql.gz
```

# Mode de contraste élevé :



# Rapport lighthouse 29/08



## METRICS

First Contentful Paint  
**2.1 s**

Total Blocking Time  
**0 ms**

Speed Index  
**2.1 s**

[Expand view](#)

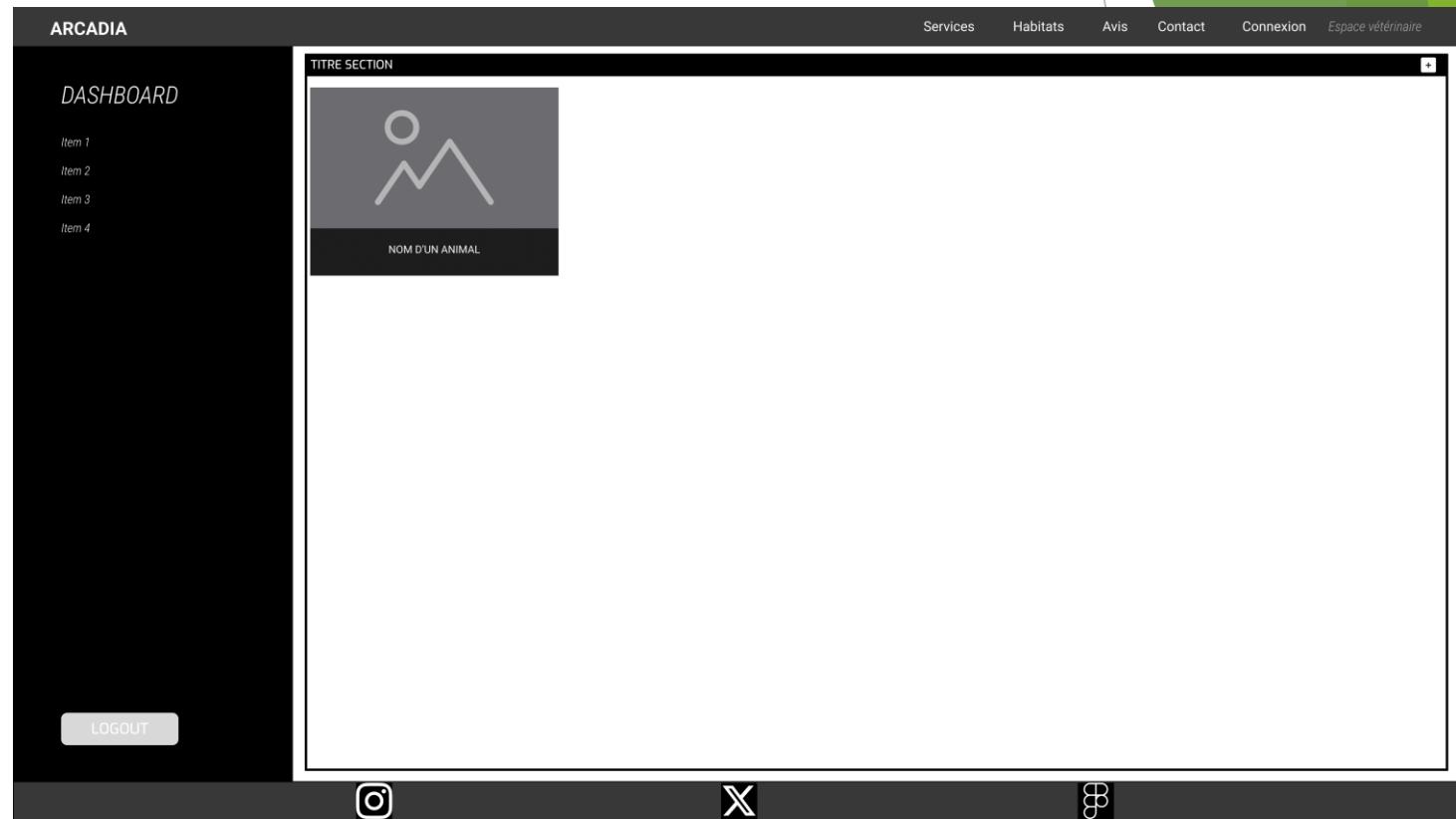
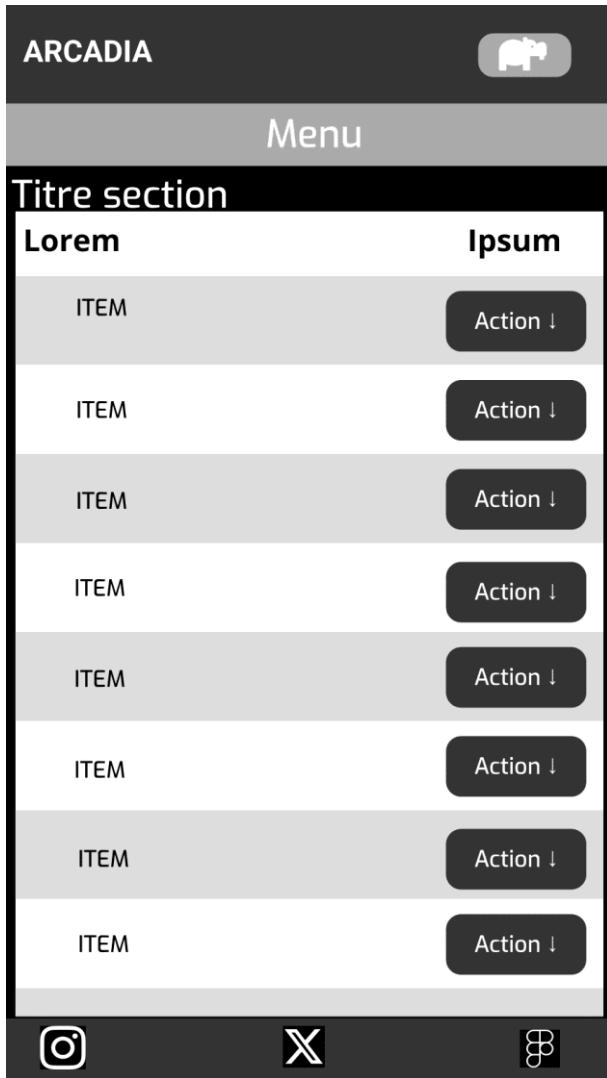
Largest Contentful Paint  
**5.2 s**

Cumulative Layout Shift  
**0**

# Wireframes

The wireframe shows the main landing page of the Arcadia website. At the top, there's a header with the logo (stylized mountains and a circle) and the text "ARCADIA LE PARC ANIMALIER DE VOS RÊVES". Below the header, a banner says "DECOUVREZ NOS HABITATS CONÇUS POUR REPRODUIRE LES CONDITIONS DE VIE DES ANIMAUX DANS LEUR MILIEU NATUREL". The main content area features a large central image placeholder with arrows on either side, labeled "NOS ANIMAUX PORTRAITS DE QUELQUES UNS DE NOS ANIMAUX". Below this, another section is titled "DECOUVREZ NOS SERVICES POUR RENDRE VOTRE VISITE INOUBLIABLE". At the bottom, there's a testimonial section "ILS EN PARLENT" with a review snippet and a "Déposer un avis" form.

This wireframe represents a services page from the Arcadia website. It includes a header with the logo and the text "ARCADIA LE PARC ANIMALIER DE VOS RÊVES". Below the header, a banner says "DECOUVREZ NOS HABITATS". The main content area displays a vertical list of service categories, each with a placeholder image and a title like "Nom d'un animal" or "Services". At the bottom, there's a testimonial section "AVIS" showing a review snippet and a "Déposer un avis" form.





## Contact

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti

Title

Title

Title

Title

BUTTON



ARCADIA

Services Habitats Avis Contact Connexion

## CONTACT

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti

Titre

Placeholder

Email

Placeholder

Message:

Placeholder

BUTTON

Instagram icon

X icon

Facebook icon

# Maquettes

ARCADIA

Services Habitats Avis Contact Connexion

Arcadia  
Le parc animalier de vos rêves

Découvrez nos habitats  
Conçus pour reproduire les conditions de vie dans leur milieu naturel

Savanne

Nos animaux:  
Portraits de quelques uns de nos animaux

LION    GIRAFE    PANTHÈRE DES NEIGES    PANTHÈRE NOIRE

Découvrez nos services  
Pour rendre votre visite inoubliable

Restaurant Profitez d'une pause gourmande près des animaux.

Balade en petit train Une balade autour du zoo dans un petit train

Visite guidée des habitats Visitez les animaux accompagnés d'un guide vétérinaire.

Ils en parlent !

Pseudo visiteur 01/04/2024

Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placet facere possimus, omnis voluptas assumenda est.

Déposer un avis:

Pseudo visiteur Arcadia Entrer un nom:   
Votre commentaire:  0/255  
Envoyer

Horaires

Lundi 08:00 18:00  
Mardi 08:00 18:00  
Mercredi 08:00 18:00  
Jeudi 08:00 18:00  
Vendredi 08:00 18:00  
Samedi 08:00 18:00  
Dimanche 08:00 18:00

[Instagram](#) [Twitter](#) [Facebook](#)

ARCADIA

Le parc animalier de vos rêves

Découvrez nos habitats

Nos animaux

LION

GIRAFE

PANTHÈRE DES NEIGES

PANTHÈRE NOIRE

Découvrez nos services

Restaurant Profitez d'une pause gourmande près des animaux.

Balade en petit train Une balade autour du zoo dans un petit train

Visite guidée des habitats Visitez les animaux accompagnés d'un guide vétérinaire.

Ils en parlent !

Pseudo 08/08/2024

Un arcus ad minima tensio, quis extorunt exercitiorum ultim' corporis: succant bioncam, nisi ut aliquip eu ea commode consequatur?

Déposer un avis:

Pseudo Entrer un nom:

Votre commentaire:  0/255  
Envoyer

Horaires

Lundi 08:00 18:00  
Mardi 08:00 18:00  
Mercredi 08:00 18:00  
Jeudi 08:00 18:00  
Vendredi 08:00 18:00  
Samedi 08:00 18:00  
Dimanche 08:00 18:00

[Instagram](#) [Twitter](#) [Facebook](#)

**ARCADIA**



**Menu**

**Services**

Service	Action
Restaurant	Action ↓
Balade en petit train	Action ↓
Visite guidée des habitats	Action ↓

[!\[\]\(ae4d6aa98879720e7f105369efa33a6b\_img.jpg\)](#) [!\[\]\(120f84555a25420901a4f2b9efa271da\_img.jpg\)](#) [!\[\]\(2a873aecb809e3b306dfc6a35a1d32b6\_img.jpg\)](#)

**ARCADIA**

Services

Espace employé

Avis

**Services**

Repas des animaux

Contact

Service

Restaurant

Balade en petit train

Visite guidée des habitats

Action

Action ↓

Action ↓

Action ↓

[!\[\]\(19a063e07fff63a5f05d54fde1d55c61\_img.jpg\)](#) [!\[\]\(1858c3aa58f4be00e0b1cc6a2f0d8837\_img.jpg\)](#) [!\[\]\(bfb6ea2cc704bc1b25c692f2e0e9ecfe\_img.jpg\)](#)

**ARCADIA**



Ma demande concerne le zoo :

Arcadia

Titre

Placeholder

Mail:

exemple@mail.com

Message

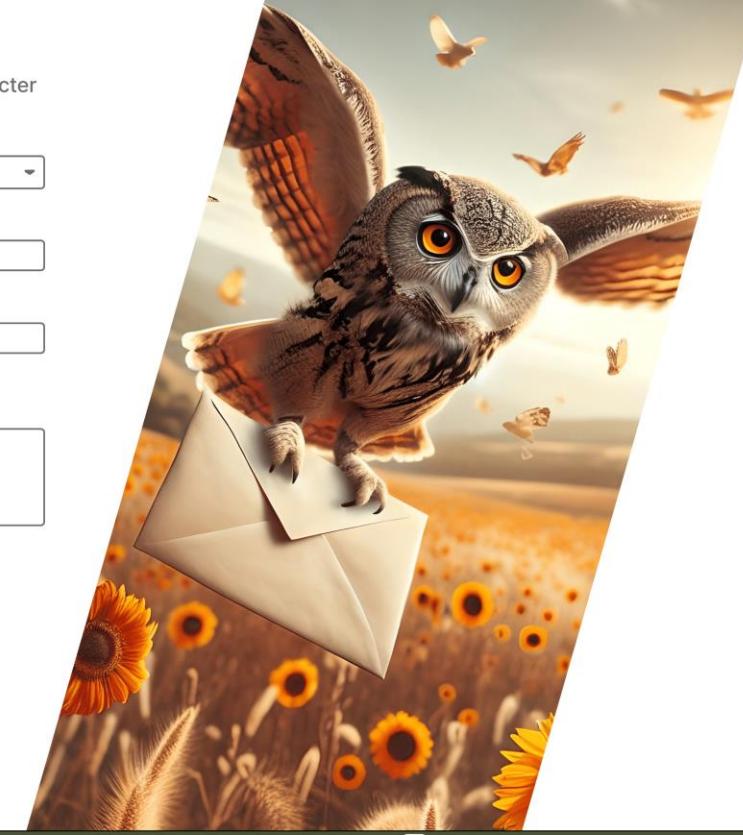
Instagram icon

X icon

Facebook icon

The form is a contact request page for a zoo named Arcadia. It includes fields for the zoo name (Arcadia), title (Placeholder), email (exemple@mail.com), and a message area. Social media icons for Instagram, X (Twitter), and Facebook are at the bottom.

**ARCADIA**



**Demande de contact**

Laissez un message à nos employés, ils se chargeront de vous recontacter

Ma demande concerne le zoo :

Arcadia

Titre

Placeholder

Mail

exemple@mail.com

Message

Envoyer

Instagram icon

X icon

Facebook icon

This is a contact form titled "Demande de contact" for the zoo "Arcadia". It asks users to leave a message for employees who will get back to them. Fields include the zoo name (Arcadia), title (Placeholder), email (exemple@mail.com), and a message area. A large owl image is on the right. Social media icons for Instagram, X (Twitter), and Facebook are at the bottom.