

NOM	BORGNE
Prénom	Sébastien
Date de naissance	29/10/1999

## Document Technique

### Graduate Développeur

*(Android, Angular, Flutter, Front End, Full Stack, IOS, PHP/Symfony)*

#### A. Spécifications techniques

Serveur // Déploiement :

- Serveur privé virtuel Hostinger
- Nginx
- Version PHP 8.2
- Extension PHP : mongoDB
- MariaDB (version 10.6)

Front end

- HTML 5
- CSS 3
- Bootstrap
- Javascript

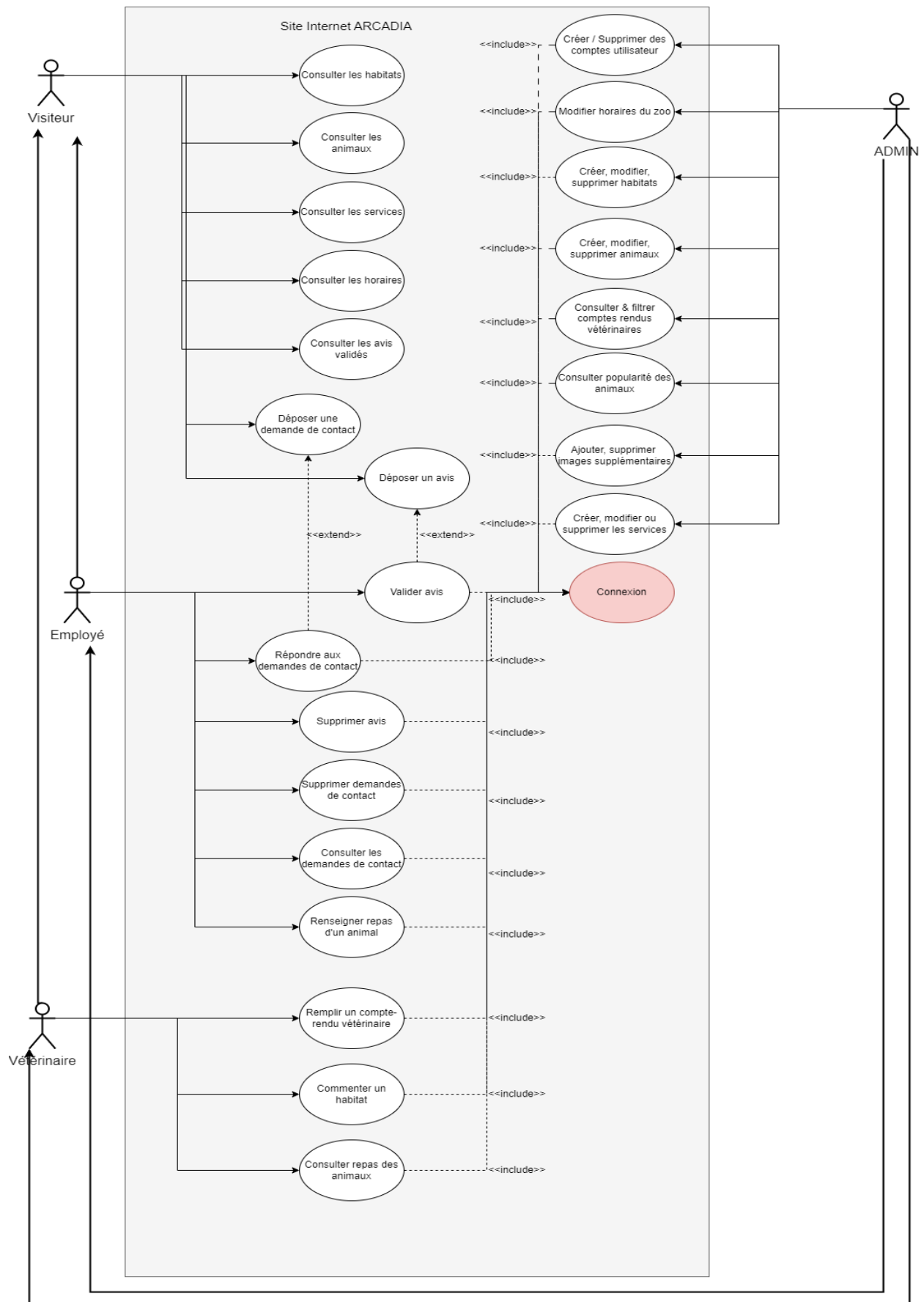
Back end

- PHP 8.2
- Symfony 7
- Doctrine ODM & ORM Bundles

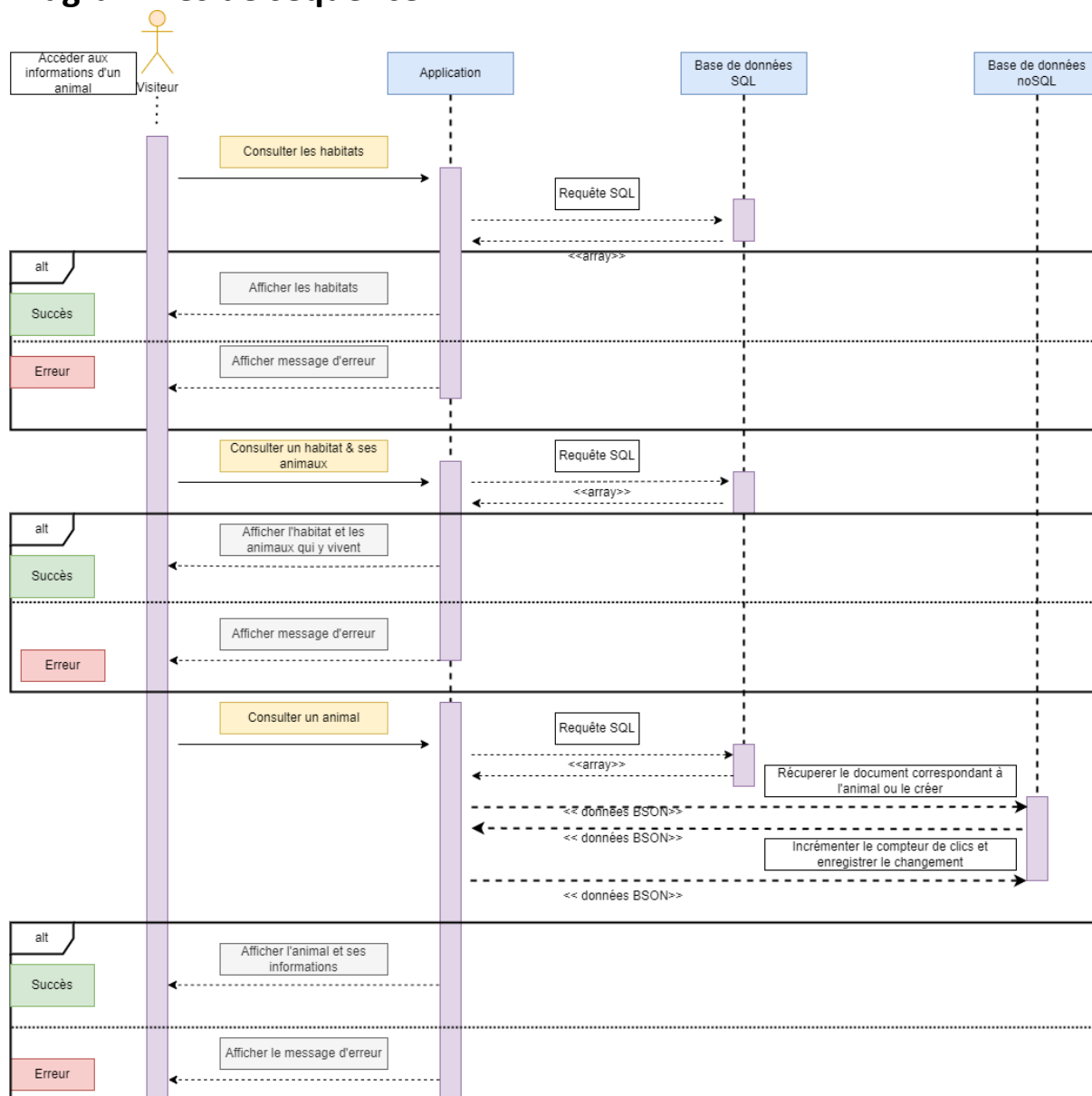
Outils vérification du code :

- ESLint (extension VSCode qui permet de vérifier la qualité du code ES et JS)
- Linter (utilise PHP code sniffer)

## Diagramme de Cas d'utilisation

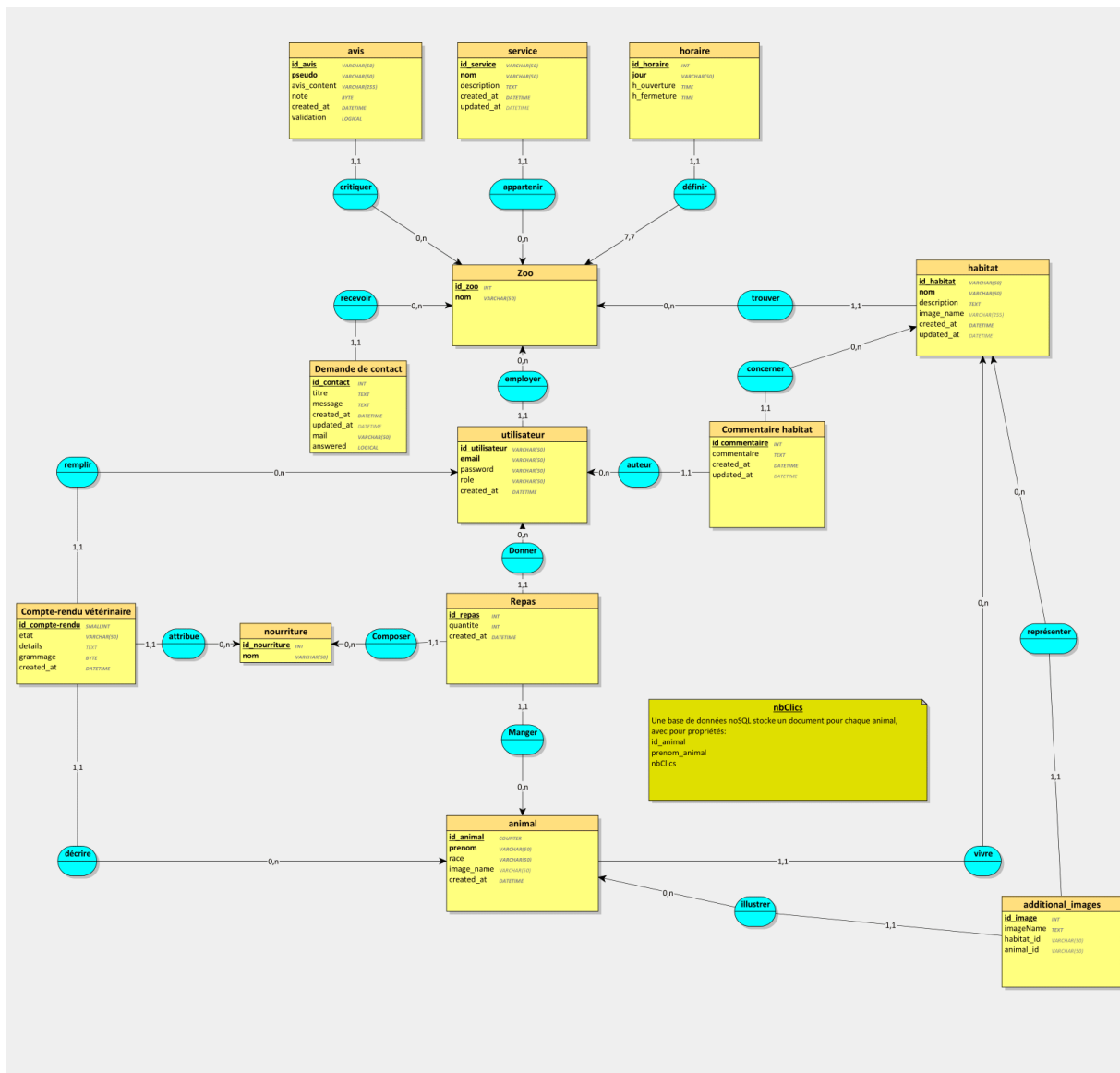


## Diagrammes de séquence

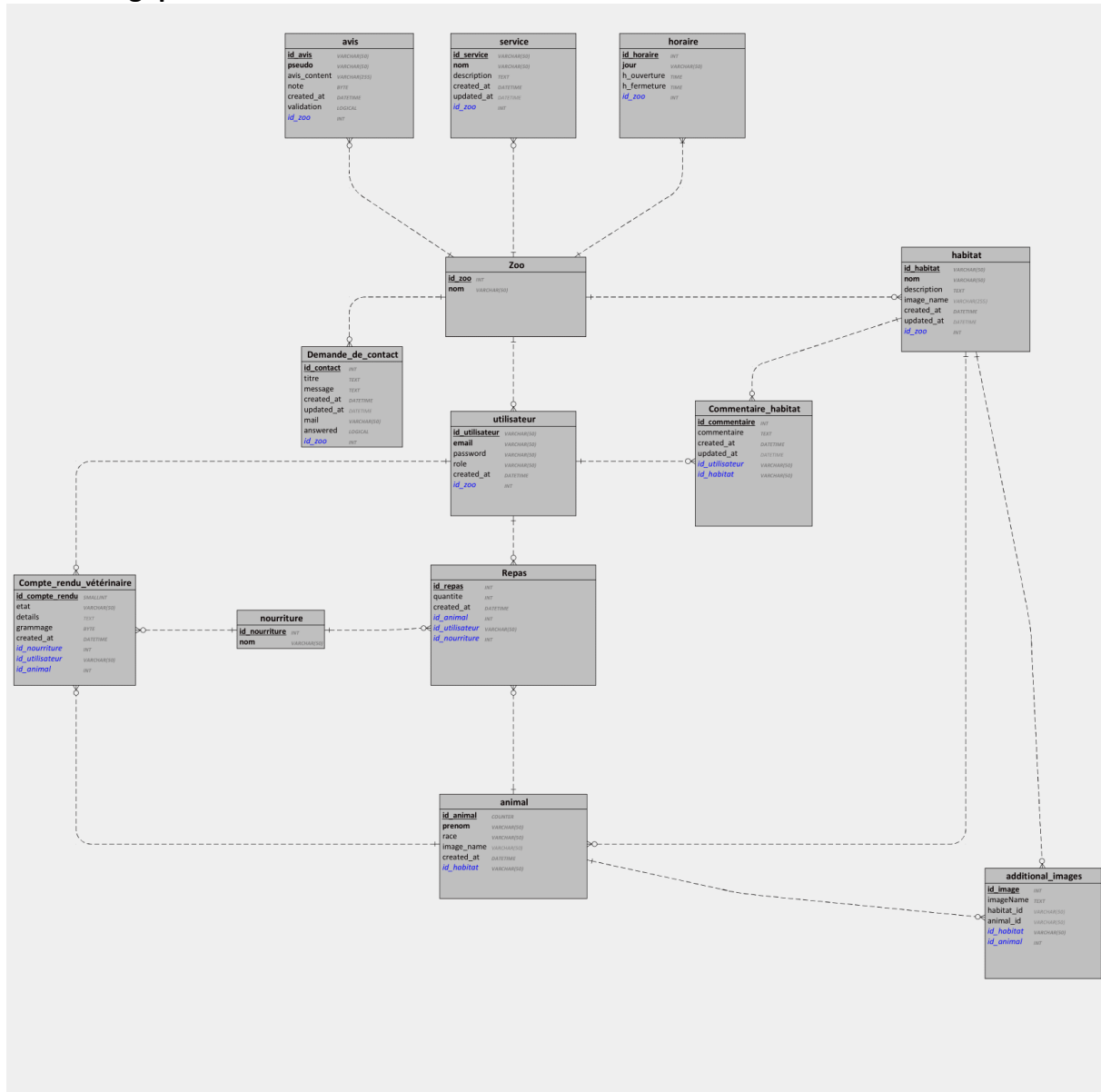


## Diagramme Méthode MERISE

### Modèle conceptuel de données



## Modèle logique de données :



```

nourriture = (id_nourriture INT, nom VARCHAR(50));
Zoo = (id_zoo INT, nom VARCHAR(50));
utilisateur = (id_utilisateur VARCHAR(50), email VARCHAR(50), password VARCHAR(50), role VARCHAR(50),
created_at DATETIME, #id_zoo);
horaire = (id_horaire INT, jour VARCHAR(50), h_ouverture TIME, h_fermeture TIME, #id_zoo);
avis = (id_avis VARCHAR(50), pseudo VARCHAR(50), avis_content VARCHAR(255), note BYTE, created_at
DATETIME, validation LOGICAL, #id_zoo);
habitat = (id_habitat VARCHAR(50), nom VARCHAR(50), description TEXT, image_name VARCHAR(255), created_at
DATETIME, updated_at DATETIME, #id_zoo);
service = (id_service VARCHAR(50), nom VARCHAR(50), description TEXT, created_at DATETIME, updated_at
DATETIME, #id_zoo);
Demande_de_contact = (id_contact INT, titre TEXT, message TEXT, created_at DATETIME, updated_at DATETIME,
mail VARCHAR(50), answered LOGICAL, #id_zoo);
Commentaire_habitat = (id_commentaire INT, commentaire TEXT, created_at DATETIME, updated_at DATETIME,
#id_utilisateur, #id_habitat);
animal = (id_animal COUNTER, prenom VARCHAR(50), race VARCHAR(50), image_name VARCHAR(50), created_at
DATETIME, #id_habitat);
Compte_rendu_veterinaire = (id_compte_rendu SMALLINT, etat VARCHAR(50), details TEXT, grammage BYTE,
created_at DATETIME, #id_nourriture, #id_utilisateur, #id_animal);
Repas = (id_repas INT, quantite INT, created_at DATETIME, #id_animal, #id_utilisateur, #id_nourriture);
additional_images = (id_image INT, imageName TEXT, habitat_id VARCHAR(50), animal_id VARCHAR(50),
#id_habitat, #id_animal);

```

## Charte graphique :

### 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



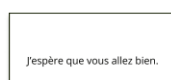
LOGO NOIR



LOGO BLANC

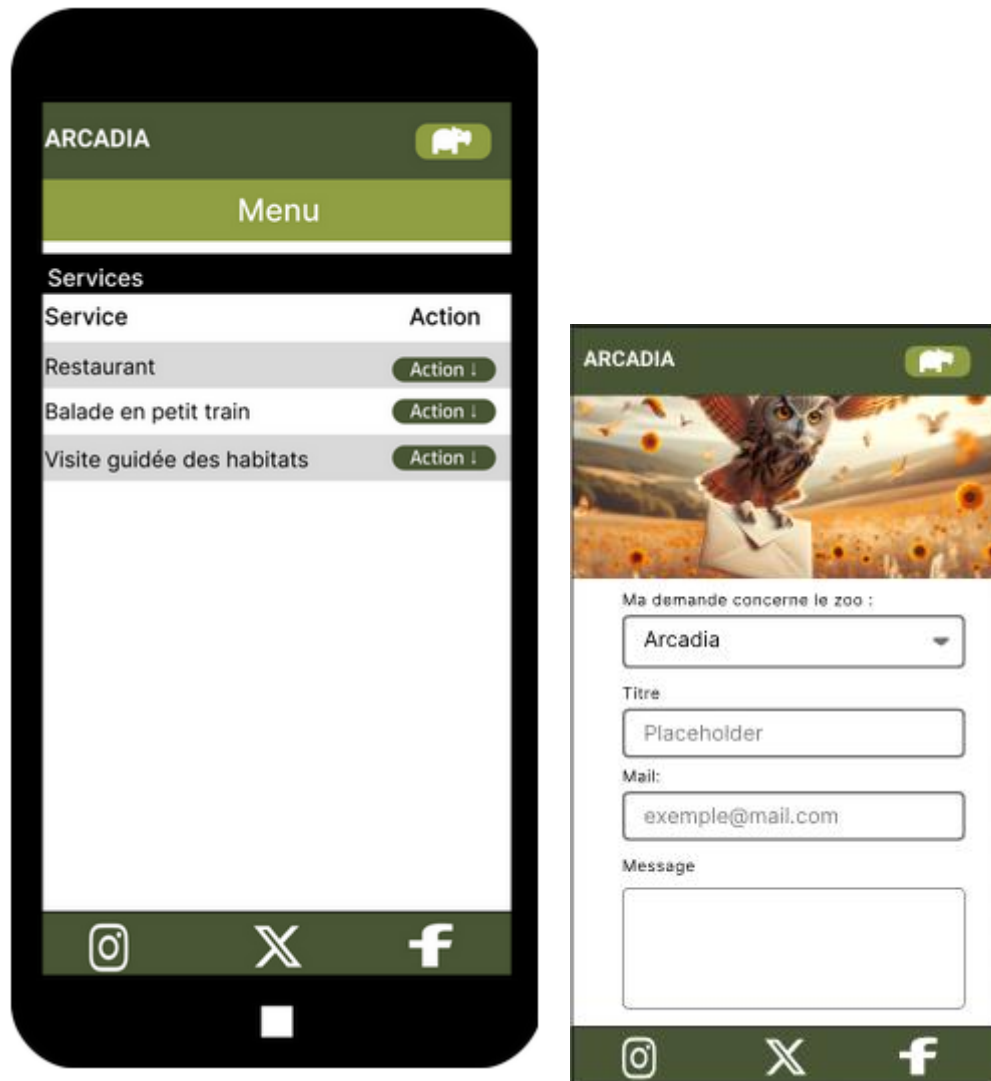


BOUTONS



TEXTE  
Open Sans

## Maquette Mobile :



*Exemple de maquette mobile*

Les maquettes sont disponibles sur disponibles à l'adresse :  
<https://www.figma.com/design/PR4xGpCdkuMdbPZWbBvg79/ECF-ARCADIA?node-id=7-613&t=LJOrvIVNoSh9j2PC-0>

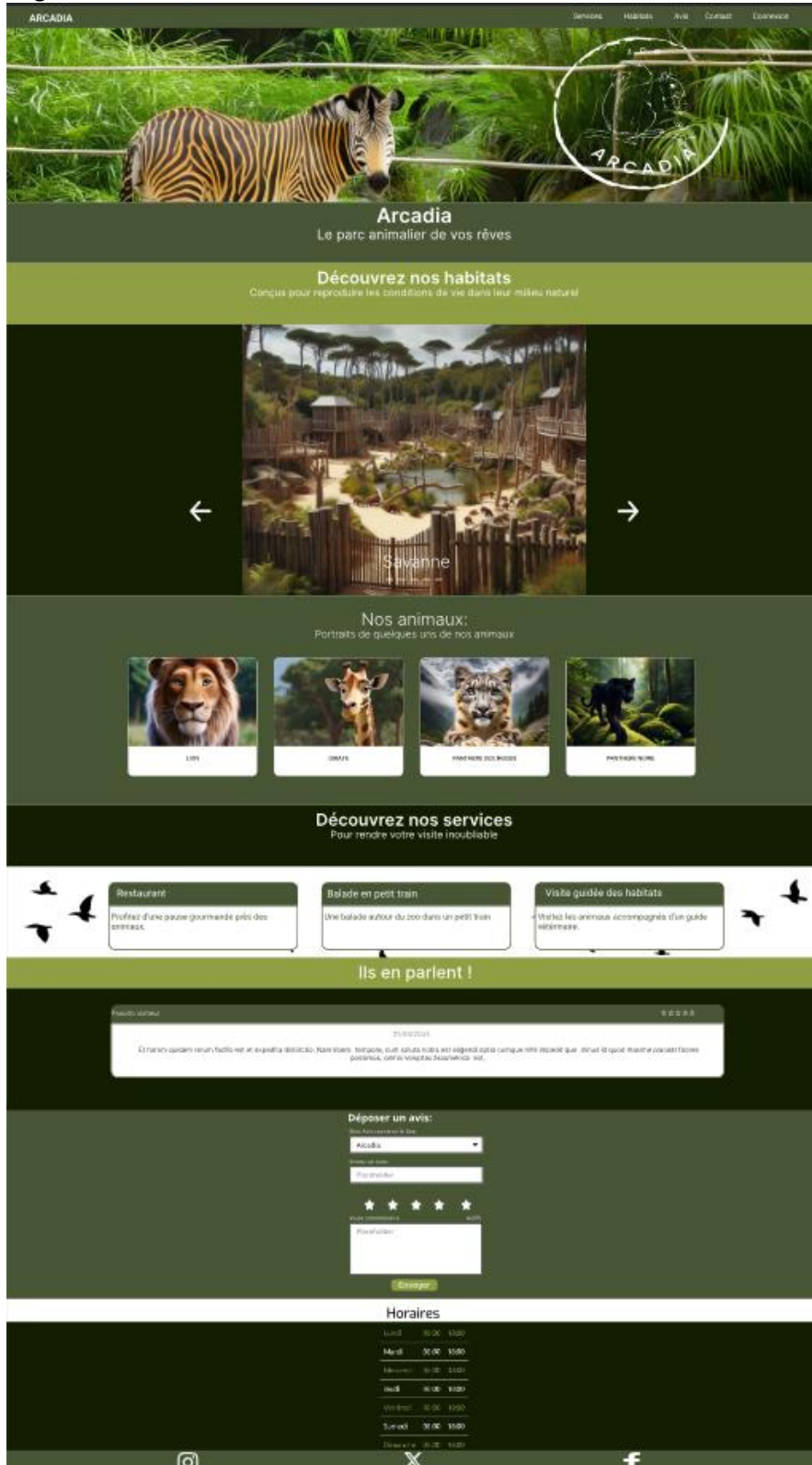


## Maquettes desktop :

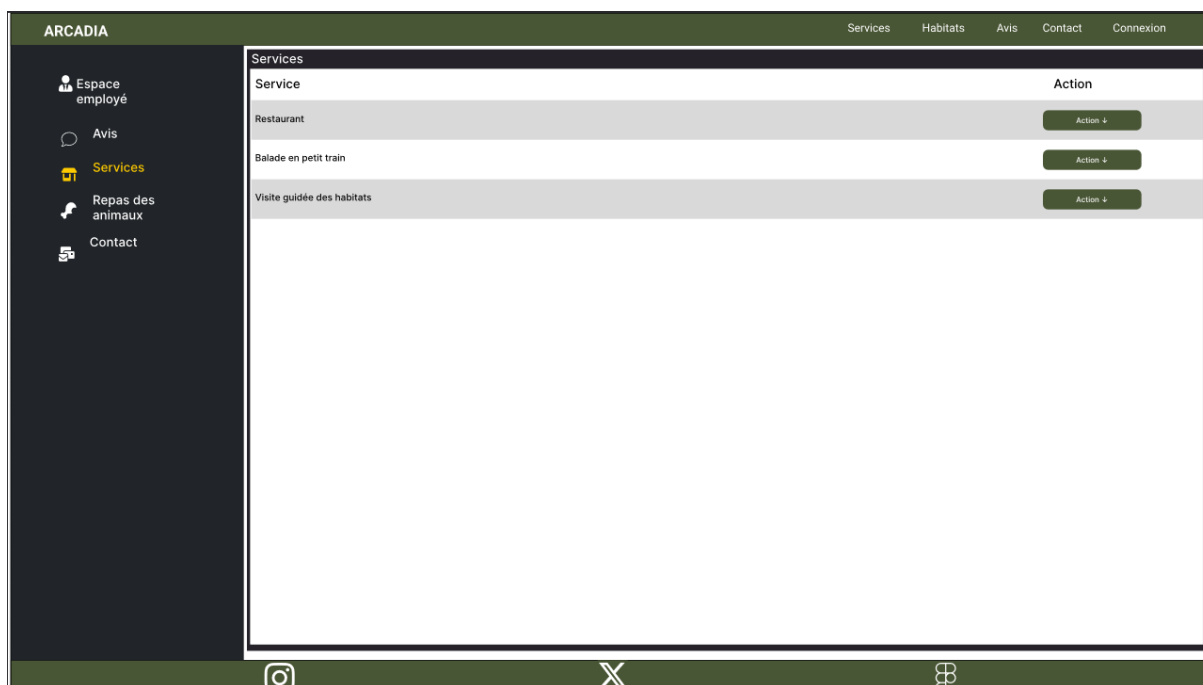
Les maquettes sont disponibles sur disponibles à l'adresse :

<https://www.figma.com/design/PR4xGpCdkuMdbPZWbBvg79/ECF-ARCADIA?node-id=7-613&t=LJOrIVNoSh9j2PC-0>

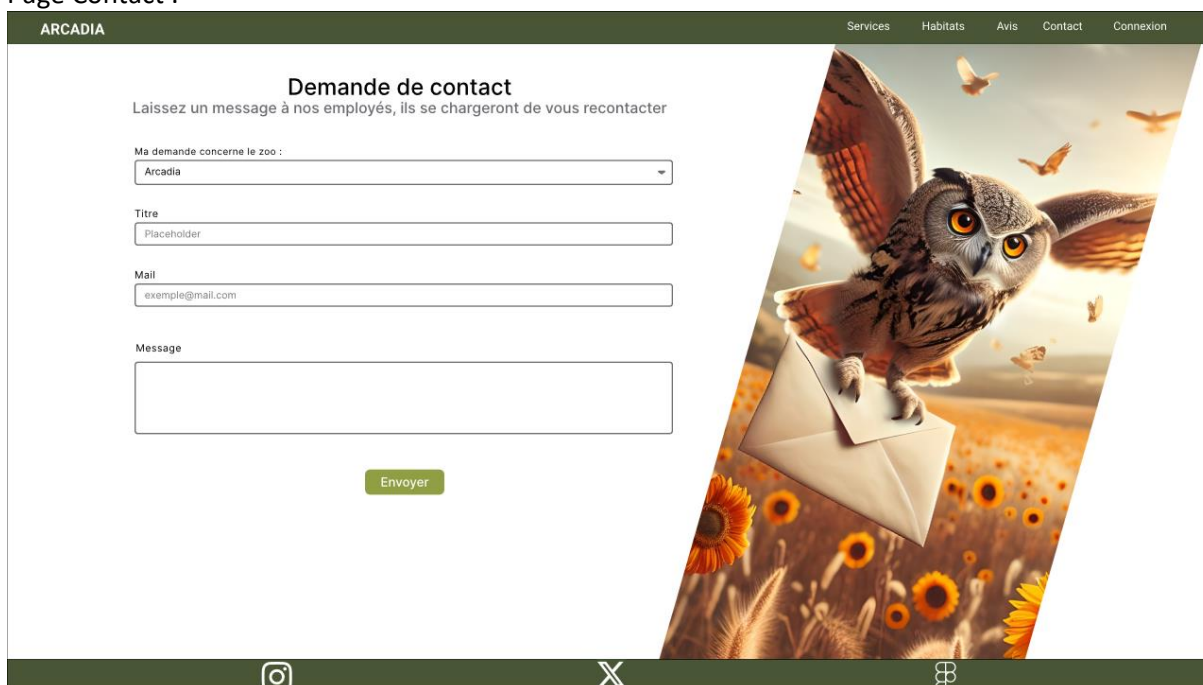
### Page d'accueil :



Dashboard employé :



Page Contact :



## Déploiement de l'application :

Pour déployer l'application, j'ai loué un serveur privé virtuel (VPS) chez Hostinger. Cela me permet de facilement configurer l'environnement de déploiement, les versions de PHP et de ses extensions. J'ai créé un fichier `main.yaml` dans le dossier `.github` qui permet de créer et configurer une action github personnalisée.

Ce fichier indique que quand un push a lieu sur la branche `main` du dépôt git, un script utilisant `SSH_PASS` effectue un `git pull` dans le répertoire dédié au site Arcadia sur mon VPS. Cela permet de déployer automatiquement les dernières mises à jour sur le site. On peut surveiller l'avancement du déploiement avec le panneau de contrôle des actions GitHub.

```
- name: Deploy application
  run: |
    sshpass -e ssh -o StrictHostKeyChecking=no ${ secrets.SSH_USER }}@${ secrets.SSH_IP }} "
    cd htdocs &&
    cd arcadia-zoo.online/ &&
    cd Arcadia &&
    git pull
    composer install
    "
```

Les dépendances du projet étant indiquées dans le fichier `composer`, il faut effectuer un `composer update` et un `composer install` au premier déploiement pour mettre à jour et installer les dépendances, mais aussi si des modifications sont apportées à la liste des dépendances.

La base de données relationnelle est hébergée directement sur le serveur. Quant à la base de données non relationnelle, elle est hébergée sur un cluster MongoDB (AWS Ireland).

J'ai loué le nom de domaine `arcadia-zoo.online` à travers l'aide d'Hostinger toujours, cela pour une durée de un an facilement renouvelable.

Pour travailler de manière optimale, il faut également implémenter une action qui réalise les tests de manière automatisée, pour coller à la démarche DevOps. J'ai par exemple mis en place un test fonctionnel concernant la connexion à la page d'accueil du site. Il me permet de vérifier que la connexion aux bases de données relationnelle et non relationnelle fonctionne correctement, car si la connexion ne se faisait pas, la réponse ne serait pas 'successful' (status 200).

```
class AcceuilControllerTest extends WebTestCase
{
    public function testAcceuil(): void
    {
        $client = static::createClient();
        $crawler = $client->request('GET', '/');

        $this->assertResponseIsSuccessful();
        $this->assertSelectorTextContains('h1', 'Arcadia');
    }
}
```

