Zoo Arcadia - Documentation technique

1. Réflexions initiales technologiques

Symfony, MySQL et MongoDB sont des choix pertinents pour ce projet en raison de leurs caractéristiques et de leurs avantages respectifs.

Symfony est un framework PHP populaire, largement utilisé dans le développement d'applications complexes et évolutives. Il offre une structure robuste pour le développement web, facilitant la gestion des différentes fonctionnalités de l'application. De plus, Symfony est connu pour sa sécurité et sa stabilité, ce qui est crucial pour un projet comme celui-ci.

MySQL est un système de gestion de base de données relationnelle largement utilisé dans le développement web. Il est capable de stocker et de gérer de grandes quantités de données de manière efficace, ce qui est indispensable pour une application qui gère des informations sur les habitats, les animaux, les vétérinaires, les employés et les visiteurs. De plus, MySQL offre des fonctionnalités avancées pour la gestion des données, facilitant ainsi la mise en place de requêtes complexes pour récupérer les informations nécessaires.

MongoDB, d'autre part, est une base de données NoSQL qui est idéale pour stocker des données non structurées ou semi-structurées, ce qui est essentiel pour la gestion des statistiques de consultation des habitats par les visiteurs. MongoDB est capable de gérer de manière efficace des volumes importants de données en temps réel, ce qui est crucial pour le suivi en temps réel des consultations des visiteurs sur l'application.

En combinant Symfony pour le développement de l'application, MySQL pour la gestion des données structurées et MongoDB pour le suivi des statistiques en temps réel, l'application sera en mesure de répondre efficacement à tous les besoins fonctionnels et techniques du projet. Ces technologies offrent une combinaison puissante pour garantir la sécurité, la stabilité et la performance de l'application, tout en assurant une expérience utilisateur optimale pour les visiteurs du zoo Arcadia.

Zoo Arcadia - Documentation technique

Configuration de l'environnement de travail

- Installation de VSCode
- Installation de WAMP
- Installation de composer
- Installation de MongoDb Community Edition pour développer en local
- Création d'un dossier de travail
- Dans ce dossier initiation d'un depot git :

$ cd dossier_de_travail

$ git init

- Installation complète de Symfony

$ composer create-project symfony/skeleton:"7.0."*

$ composer require webapp

- Installation de Bootstrap

$ composer require twbs/bootstrap

- Configuration Apache vhosts

```
<VirtualHost *:80>
   ServerName www.zoo-arcadia.dvlp
   DocumentRoot "I:\formation\Zoo-Arcadia/public"
   <Directory  "I:\formation\Zoo-Arcadia/public/">
      Options +Indexes +Includes +FollowSymLinks +MultiViews
      AllowOverride All
      Require local
   </Directory>
   ErrorLog "${INSTALL_DIR}/logs/mon_projet_web-error.log"
   CustomLog "${INSTALL_DIR}/logs/mon_projet_web-access.log" common
</VirtualHost>
```

- Configuration d'un fichier .env.local à la racine du projet Symfony

```
DATABASE_URL="mysql://username:password@127.0.0.1:3306/sf_zoo_arcadia?serverVersion=8.0.32&charset=utf8mb4"
MAILER_DSN=smtp://username:password@sandbox.smtp.mailtrap.io:2525
MONGODB_URL=mongodb+srv://username:password@localhost:27017
MONGODB_DB=arcadia
```

- Téléchargement du driver mongodb pour php dans le dossier C:\wamp64\bin\php\php8.2.0\ext

Zoo Arcadia - Documentation technique

- Ajout de la ligne *extension=mongodb* dans le fichier php.ini
- Installation de sass-bundle

  $ *composer require symfonycast/sass-bundle*
- Installation de HTML Sanitizer

  $ *composer require symfony/html-sanitizer*
- Installation de DoctrineMongoDBBundle

  $ *composer require doctrine/mongodb-odm-bundle*
- Créer un repository sur Github
- 1er commit git

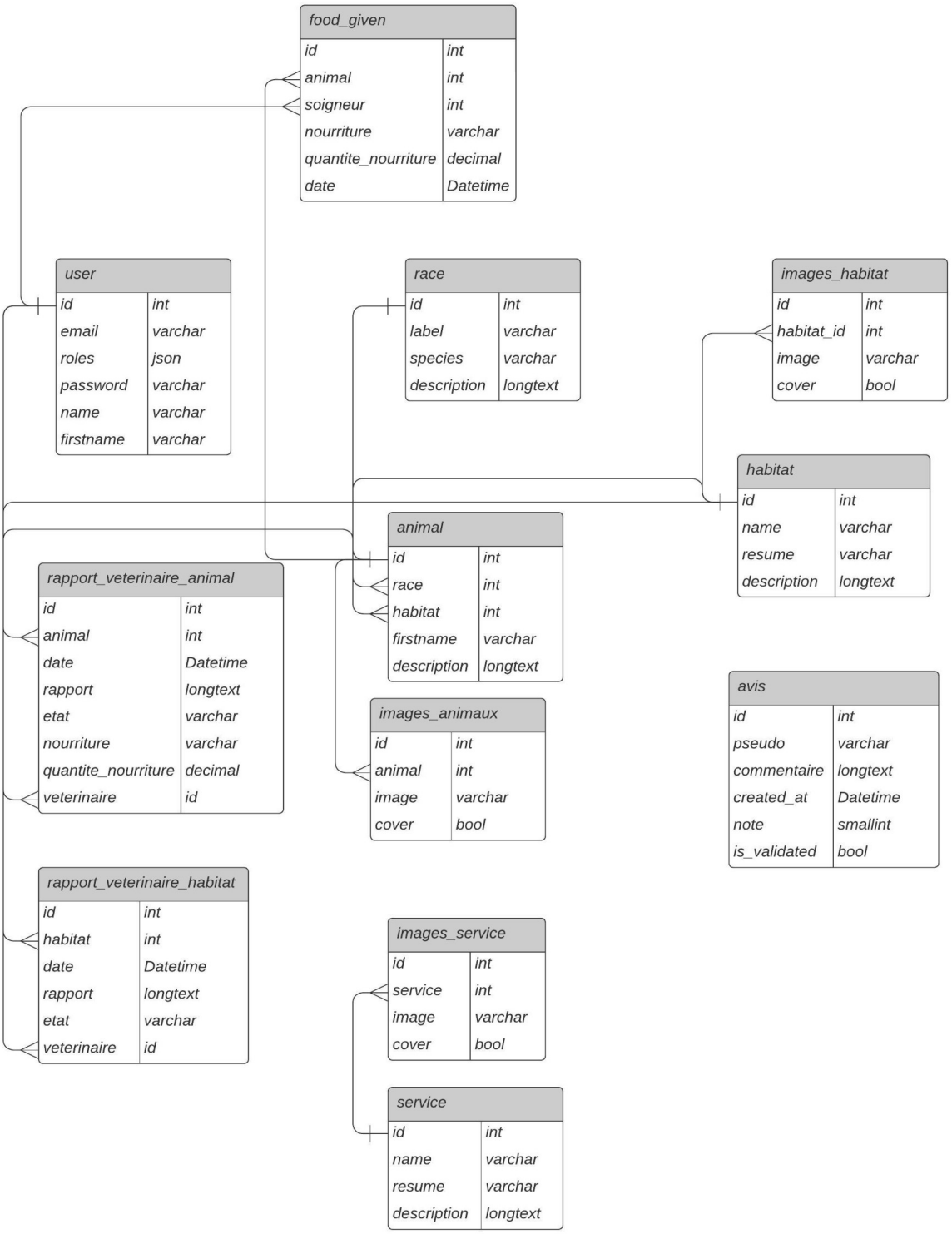
  $ *git add .*

  $ git commit -m ‘’First commit’’
- Ajouter le remote

  $ git remote add origin https://github.com/julobeau/Zoo-Arcadia
- Pousser le 1er commit

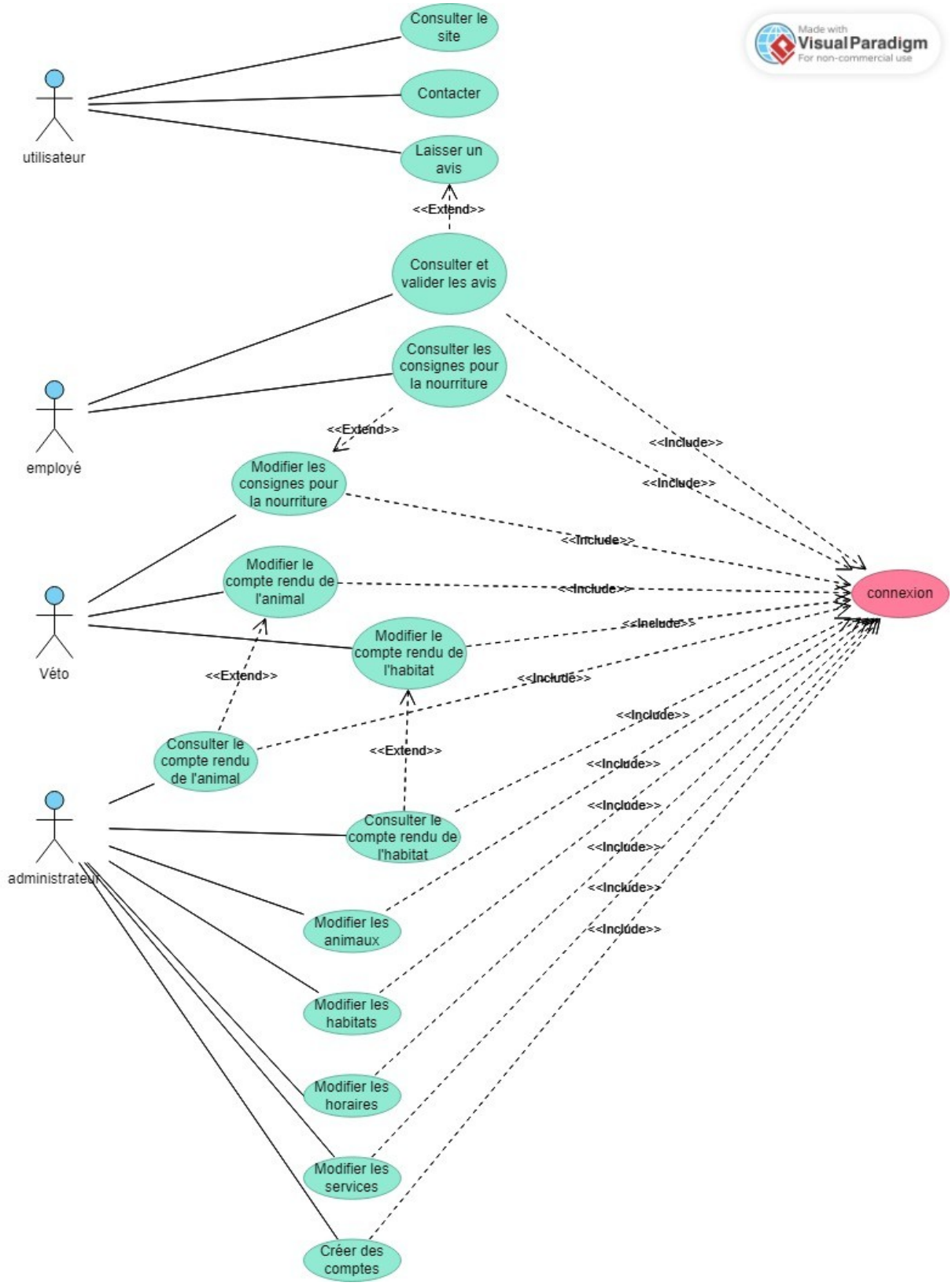
Zoo Arcadia - Documentation technique

2. **Modèle conceptuel de données**



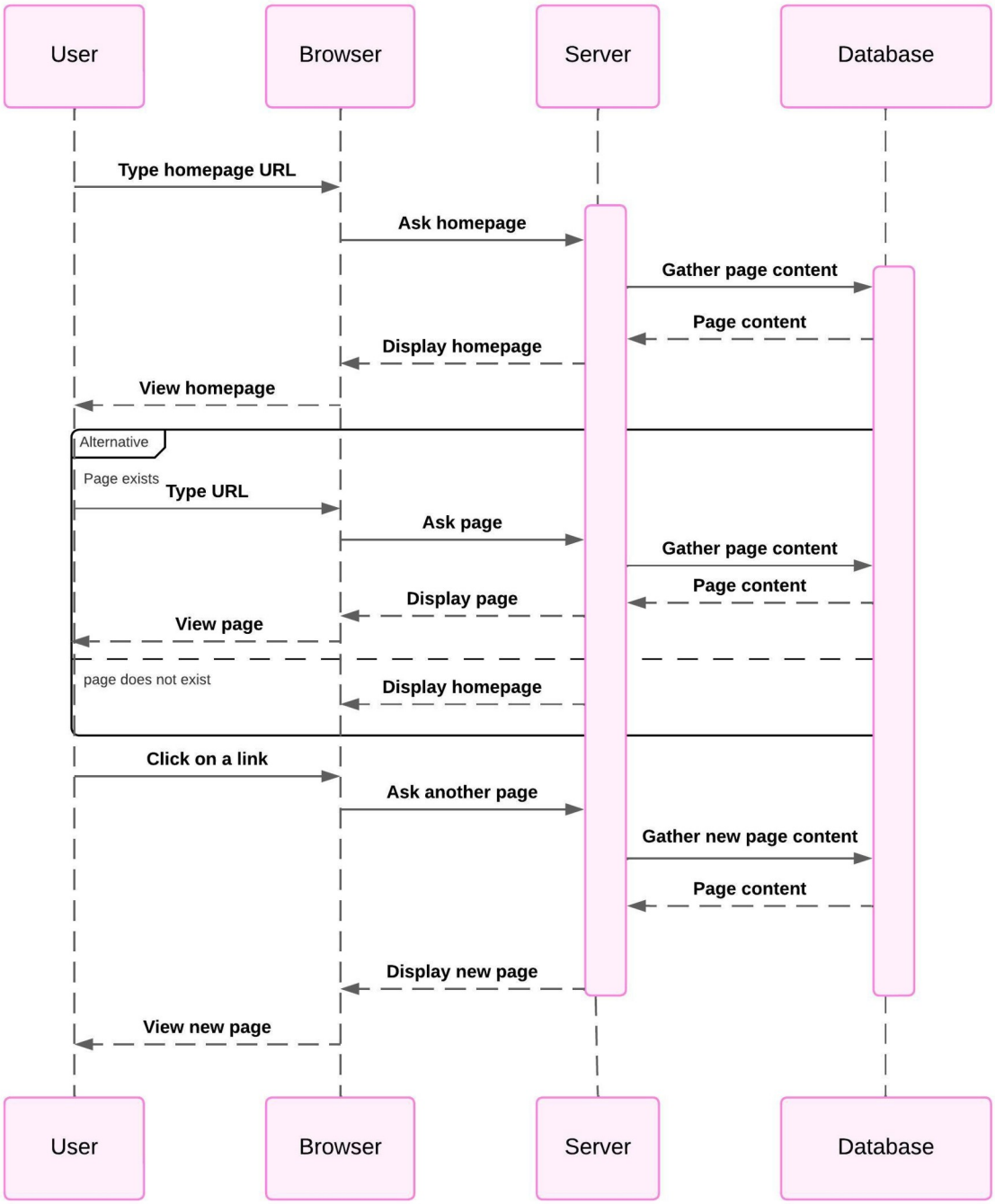
Zoo Arcadia - Documentation technique

3. **Diagramme d'utilisation**

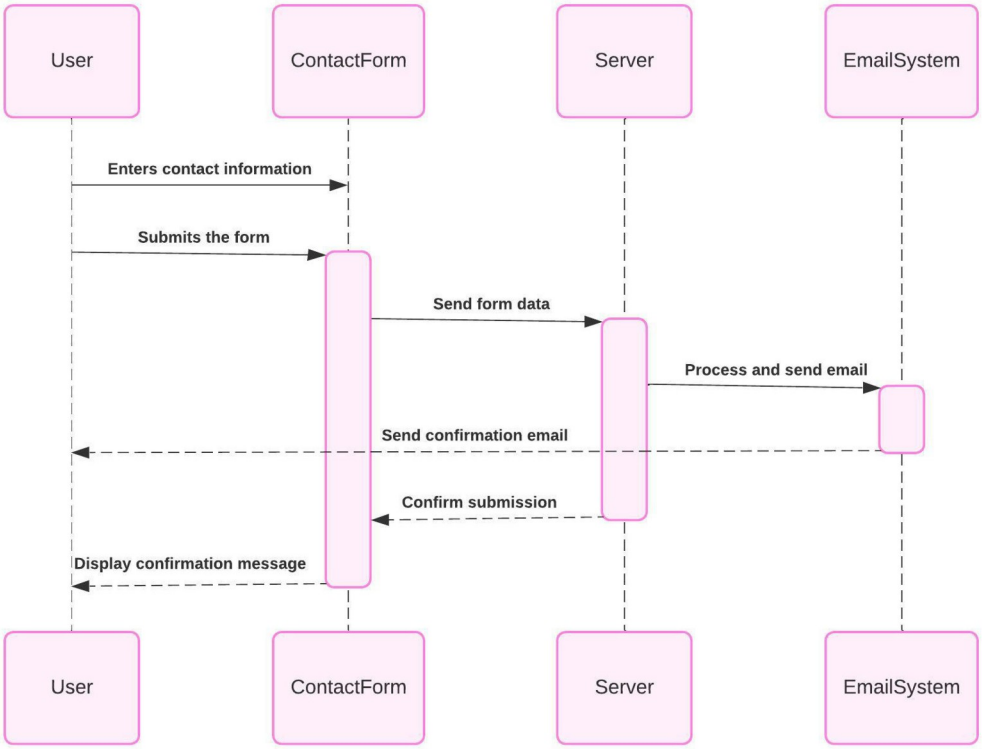


4. Diagramme de séquence

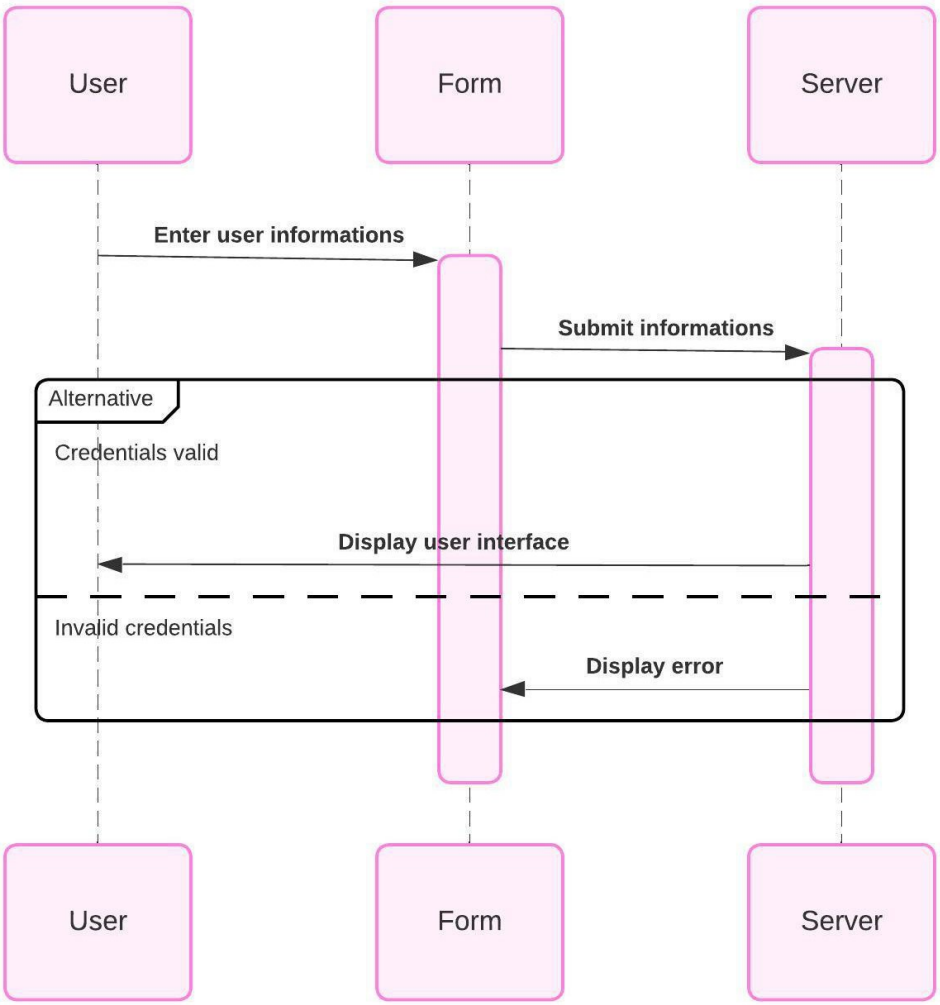
1. *Navigation*



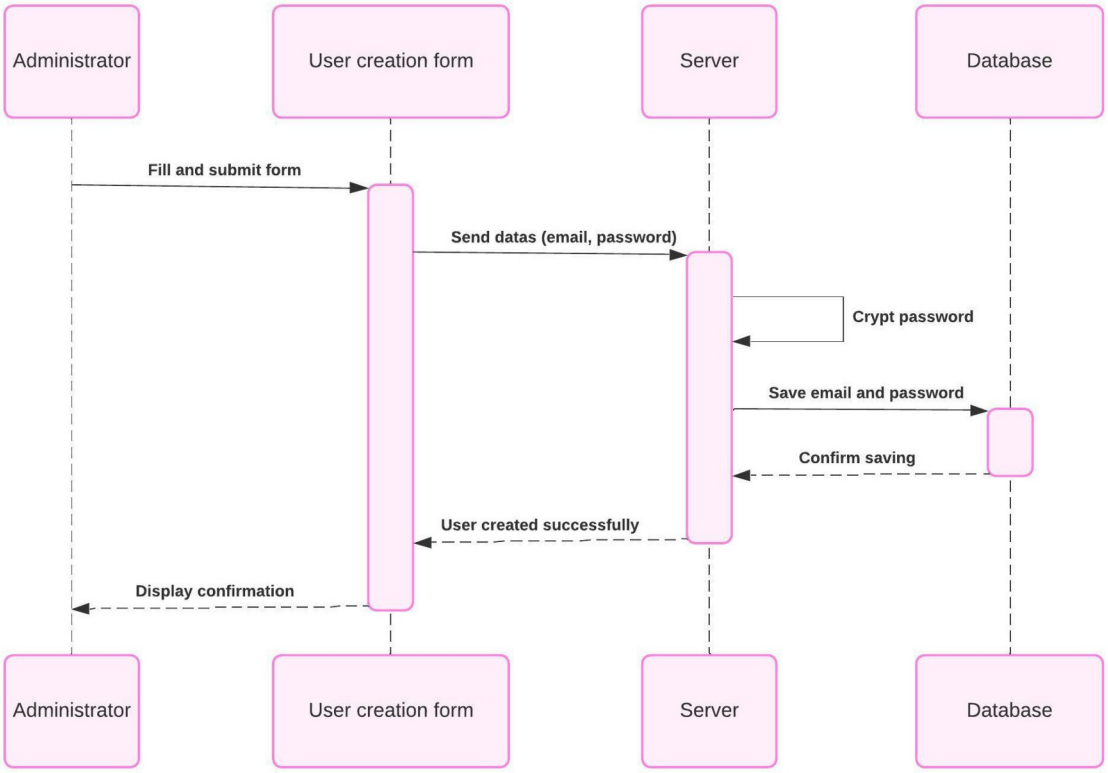
2. Contact


User
ContactForm
Server
EmailSystem
Enters contact information
Submits the form
Send form data
Process and send email
Send confirmation email
Confirm submission
Display confirmation message
User
ContactForm
Server
EmailSystem

3. Connexion


User
Form
Server
Enter user informations
Submit informations
Alternative
Credentials valid
Display user interface
Invalid credentials
Display error
User
Form
Server

4. ***Création d'un nouvel utilisateur***



Zoo Arcadia - Documentation technique

5. **Déploiement de l'application**

L'application a été déployée sur Heroku
https://zoo-arcadia-jb-9f78cb1dd18e.herokuapp.com/
Etapes :
- Création d'un compte Heroku
- Création d'une nouvelle app sur Heroku
- Ajout d'un Buildpack: heroku/php
- Ajouter un service de base de données MySQL (JawsDB)
- Création d'un compte MongoDB Atlas
- Création d'une nouvelle branche Git prod
- Modifier le fichier config/packages/framework.yaml:

```
framework:
  #...
  trusted_proxies: '127.0.0.1,REMOTE_ADDR'
  trusted_headers: ['x-forwarded-for', 'x-forwarded-host', 'x-forwarded-proto', 'x-forwarded-port', 'x-forwarded-prefix']
```

- Connexion de l'application au repository GitHub
- Création d'un fichier Procfile à la racine du projet contenant :

```
web: heroku-php-apache2 public/
```

- Choisir la branche à déployer (prod)
- Deploy Branch
- Configurer les variables d'environement :
  - APP_ENV = prod
  - APP_SECRET = xxxxxxxxxxxxxxxxxxxx
  - DATABASE_URL = mysql://user:mdp@k9xdebw4k3zynl4u.cbetxkdyhwsb.us-east-1.rds.amazonaws.com:3306/p7jgfgdrm7ahah24
  - JAWSDB_URL = mysql://user:mdp@k9xdebw4k3zynl4u.cbetxkdyhwsb.us-east-1.rds.amazonaws.com:3306/p7jgfgdrm7ahah24
  - MONGODB_URI = mongodb+srv://user:mdp@cluster0.cdotgrs.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
  - MAILER_DSN = smtp://user:mdp@sandbox.smtp.mailtrap.io:2525
- Création de la base de données

```
php bin/console doctrine:database:create
```

- Migration des tables

```
php bin/console doctrine:migrations:migrate
```