

Documentation technique

Sommaire

I.	Réflexion initiale technologique.....	2
II.	Configuration de l'environnement de travail.....	2
1.	Front-end.....	2
2.	Back-end.....	2
3.	Déploiement :.....	2
III.	Diagramme de classe.....	2
IV.	Diagrammes d'utilisation et de séquence.....	4
V.	Documentation du déploiement.....	5
1.	Choix de la plateforme	5
2.	Etapas du déploiement	5
a.	Mise en ligne de l'application.....	5
b.	Mise en ligne de la base de données PostGreSQL sur Alwaysdata	6
c.	Modification du fichier de configuration	6
d.	Mise à jour de l'application déployée	6

I. Réflexion initiale technologique

Le client souhaite disposer d'un site internet permettant d'accroître sa visibilité. Ce site sera également un outil de travail puisqu'il permettra aux employés et au vétérinaire de saisir des informations concernant les animaux. Ce site permettra également au gérant de suivre la popularité des animaux.

Afin de réaliser ce projet, plusieurs éléments devront donc être mis en place :

- Un côté front-end permettant l'affichage des données et des éléments souhaités en prenant en compte les souhaits du gérant (notamment que le site internet reflète la sobriété écologique à laquelle adhère le zoo) ;
- Un côté back-end permettant d'échanger avec des bases de données dans lesquelles seront stockées les informations ;
- Afin de faciliter la gestion du projet, celui-ci sera envoyé sur Git ;
- A la fin, le site doit être déployé.

II. Configuration de l'environnement de travail

Les technologies utilisées pour ce projet sont les suivantes :

1. Front-end

- HTML 5,
- CSS et Bootstrap : bootstrap permet une mise en page simple, rapide et responsive.
- Javascript : javascript a été utilisé pour créer des affichages dynamiques, notamment via la librairie Popper pour les carrousels et les affichages des avis. Javascript a également été utilisé pour les redirections automatiques, pour filtrer les résultats de recherche lors de l'affichage des comptes-rendus du vétérinaire par l'administrateur et pour l'affichage du graphique des clics par animaux.

2. Back-end

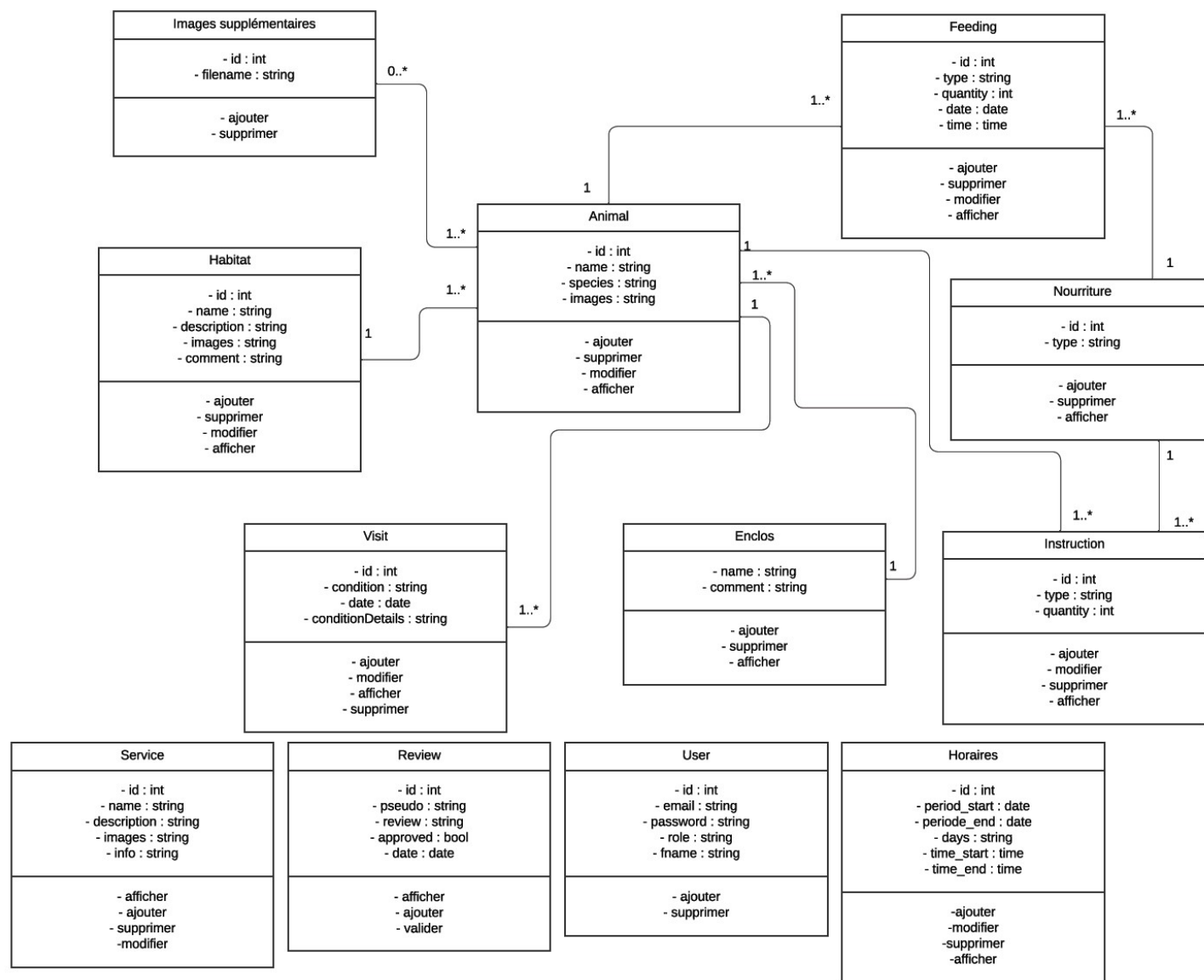
- PHP version 8.2 avec PDO pour la connexion à la base de données.
- Base de données relationnelle : PostgreSQL version 16.0
- Base de données non relationnelle : MongoDB Atlas

3. Déploiement :

- Le site a été déployé via Heroku et la base de données PostgreSQL a été mise en ligne sur AlwaysData.

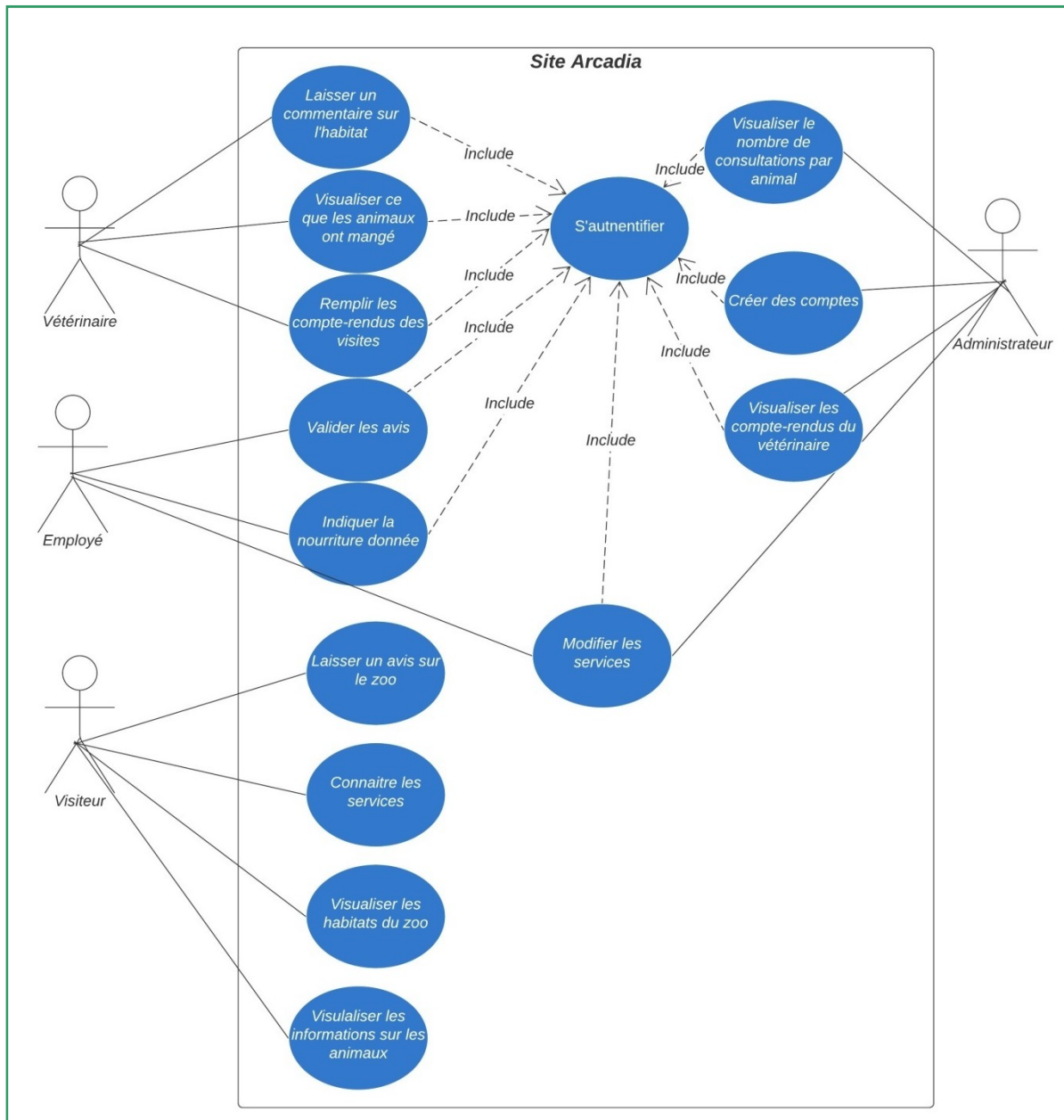
III. Diagramme de classe

Le diagramme de classe est présenté ci-après.

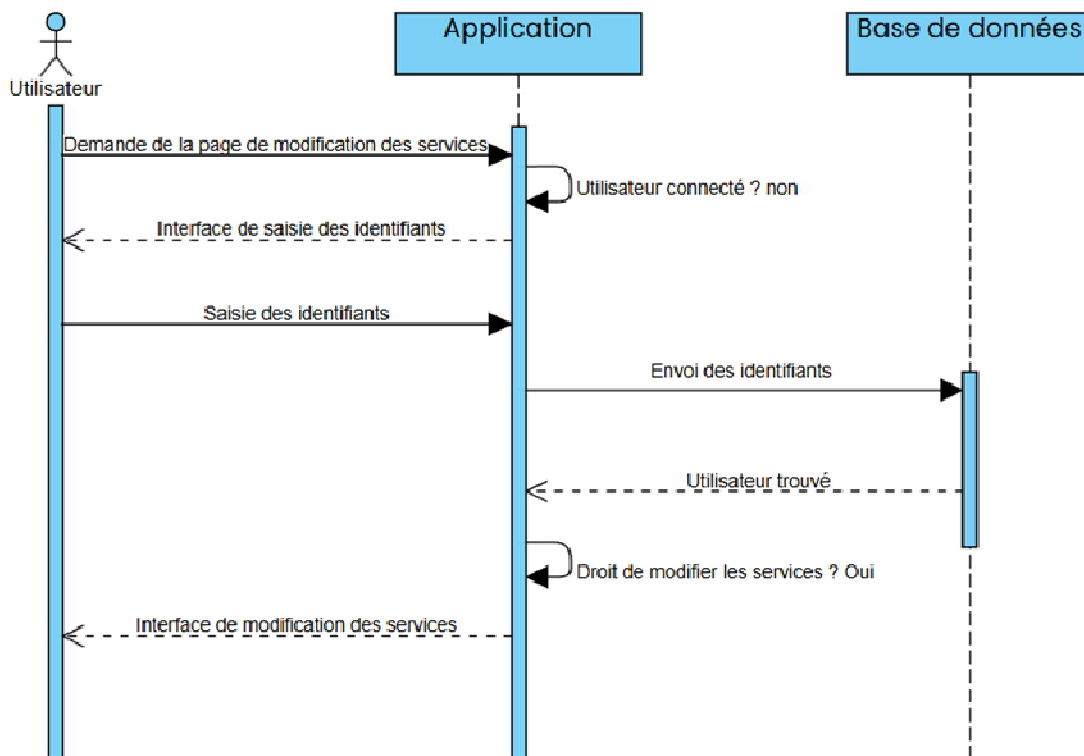


IV. Diagrammes d'utilisation et de séquence

Le diagramme d'utilisation est présenté ci-après.



Le diagramme de séquence est présenté ci-après. Il présente le cas où un employé souhaiterait modifier les données relatives à un service du zoo.



V. Documentation du déploiement

1. Choix de la plateforme

J'ai choisi de déployer le site sur Heroku pour sa simplicité d'utilisation et son coût assez faible dans le cadre d'une utilisation basique. Toujours afin de réduire les coûts, je n'ai pas utilisé d'add-on pour PostGreSql sur Heroku mais j'ai transféré la base de données sur AlwaysData afin de bénéficier d'un hébergement gratuit.

2. Etapes du déploiement

a. Mise en ligne de l'application

Le déploiement a été réalisé après installation du CLI Heroku. Le projet était déjà versionné sur GIT.

Les commandes suivantes ont été entrées dans le terminal :

- Heroku login → connexion à mon compte Heroku
- Heroku create → création du dépôt git Heroku et de la branche main
- Git push heroku main → déploiement de l'application

Pour redéployer l'application après avoir effectué des modifications, il faut réaliser les commandes suivantes :

- git add . → ajout des nouveaux fichiers

- git commit -m « message du commit » → création d'un nouveau commit
- git push heroku main → envoyer les modifications sur la branche heroku (et donc déploiement de la mise à jour)

b. Mise en ligne de la base de données PostgreSQL sur Alwaysdata

Les étapes suivantes ont permis la mise en ligne de la base de données sur alwaysdata.

- Export de la base de données PostgreSQL pour obtenir un fichier .sql
- Création de la base de données sur AlwaysData et import du fichier .sql

c. Modification du fichier de configuration

Une fois la base de données en ligne, le fichier de configuration a été modifié pour que pdo puisse se connecter à cette base de données.

- Modification du fichier lib/config.php pour pouvoir se connecter à la base de données sur AlwaysData avec une condition sur la valeur de getenv() :

```
if (getenv('alwaysdata') !== false) {
    $domain = '';
    $hostname = 'postgresql-fannyelimar.alwaysdata.net';
    $username = 'fannyelimar';
    $password = 'elimarmdp';
    $database = 'fannyelimar_arcadiadb';
} else {
    $domain = 'localhost';
    $hostname = 'localhost';
    $username = 'user';
    $password = 'arcadiamdp';
    $database = 'arcadiadb';
}
define("_DOMAIN_", $domain);
define("_DB_SERVER_", $hostname);
```

- Sur Heroku, modifications des variables d'environnement pour ajouter les données suivantes :
 - o Key : alwaysdata
 - o Value : postgresql-fannyelimar.alwaysdata.net

d. Mise à jour de l'application déployée

Les modifications apportées au fichier config.php ont été envoyées sur le git d'heroku en effectuant les commandes suivantes : git add . / git commit -m « configuration de la base de données alwaysdata » / git push heroku main