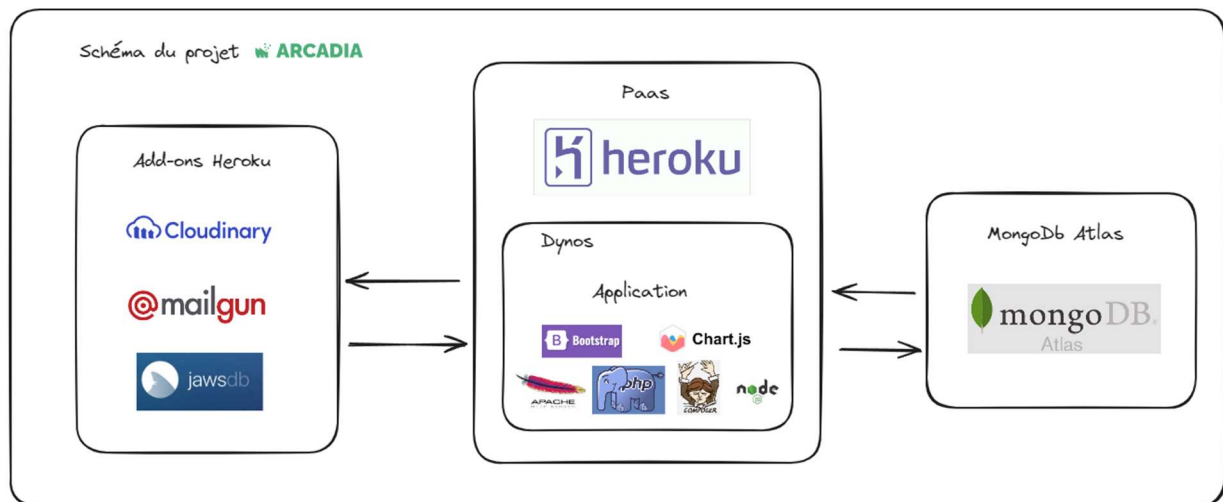


Documentation technique de l'application

1. Réflexions initiales technologiques sur le sujet



Front-end

Pour la partie Front-end, j'ai choisi de la réaliser avec le langage de balisage HTML.

Pour le style vais utiliser le framework Bootstrap dans sa version 5.3.3. qui est un framework reconnu (utilisé par Twitter). Il a pour avantage de gagner du temps au niveau du développement (les boutons sont déjà créés...), il rend le code plus robuste.

De plus avec son système de grille il permet de réaliser des applications web 100% responsive.

Il est également open-source et compatible avec tous les navigateurs modernes.

J'ai également utilisé le préprocesseur Sass pour surcharger le CSS de Bootstrap et éviter de modifier directement les fichiers d'installation, cette pratique est recommandée sur le site de Bootstrap.

Sass permet également d'imbriquer les sélecteurs comme c'est le cas avec HTML. Il rend le code plus agréable à lire.

Exemple :

```

<!-- HTML-->
<div class=" container ">
  <h1 class=" bigTitle ">
  </h1>
</div>

```

```

//AVEC SASS
.container {
  background-color: red;

  .bigTitle {
    color: white
  }
}

//CSS sans SASS
.container {
  background-color: red;
}

.container .bigTitle {
  color: white
}

```

Pour dynamiser mes pages je vais utiliser javascript pour la validation des formulaires ainsi que pour les statistiques du tableau de bord et autres...

Je vais également utiliser la librairie Chart.js pour une représentation graphique sur le tableau de bord administrateur.

Back-end

Pour le back, j'ai choisi PHP qui est un langage sécurisé de script côté serveur. C'est un langage open source disposant d'une grande communauté qui facilite le débogage.

Je vais utiliser l'extension PDO pour interagir avec ma base de données MariaDB.

Je vais également utiliser « Composer » pour installer mes packages. Je vais utiliser :

- le package « mongodb/mongodb » pour ma base de donnée MongoDB,
- le package « vlucas/phpdotenv » pour créer mon fichier « .env » pour sécuriser mes identifiants de bdd et autres,
- le package « cloudinary/cloudinary_php » pour stocker mes images
- le package « mailgun/mailgun-php,

Bases de données

Le client nous impose 2 types de base de données, une relationnelle et une noSQL.

J'ai choisi MariaDB pour la bdd relationnelle. C'est un langage open source gratuit, qui offre des performances plus rapide que MySQL. Wikipedia et Google utilise MAriaDB.

J'ai choisi MongoDB comme bdd noSQL. MongoDB est très flexible. C'est une bdd reconnu et fiable utilisé par IBM, Amazon, Microsoft..

Stockage des images et envoi des mails

Pour le stockage des images je vais utiliser Cloudinary. C'est un solution simple et gratuite pour stocker les images chargés.

Pour l'envoi des mails du site je vais utiliser Mailgun qui a une version gratuite très intéressante .

Le seul petit inconvénient est que nous devons ajouter les destinataires sur le site de Mailgun sauf si on ajoute un nom de domaine.

Mais j'ai recommandé à José (directeur du zoo) de créer un nom de domaine, c'est peu cher et ça rendra son application plus professionnelle et lui apportera aussi une meilleur visibilité.

Déploiement

J'ai choisi de déployer mon application sur Heroku.

Heroku est un Paas, il est spécialisé dans le déploiement d'applications. C'est un fournisseur de services, il construit et maintien l'infrastructure du serveur et des bdd.

Heroku permet un déploiement rapide.

Diagramme conceptuel de données

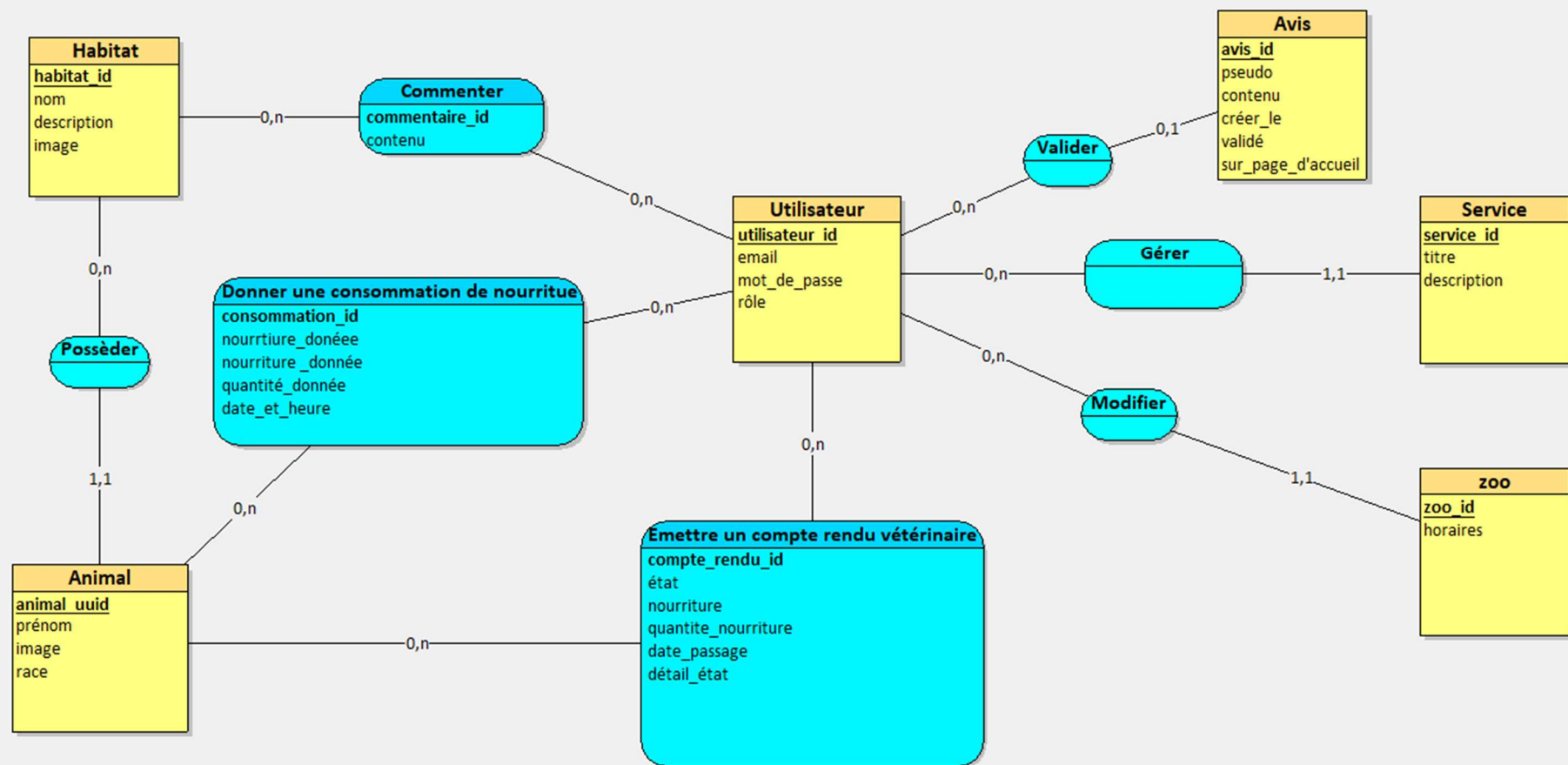


Diagramme de cas d'utilisation

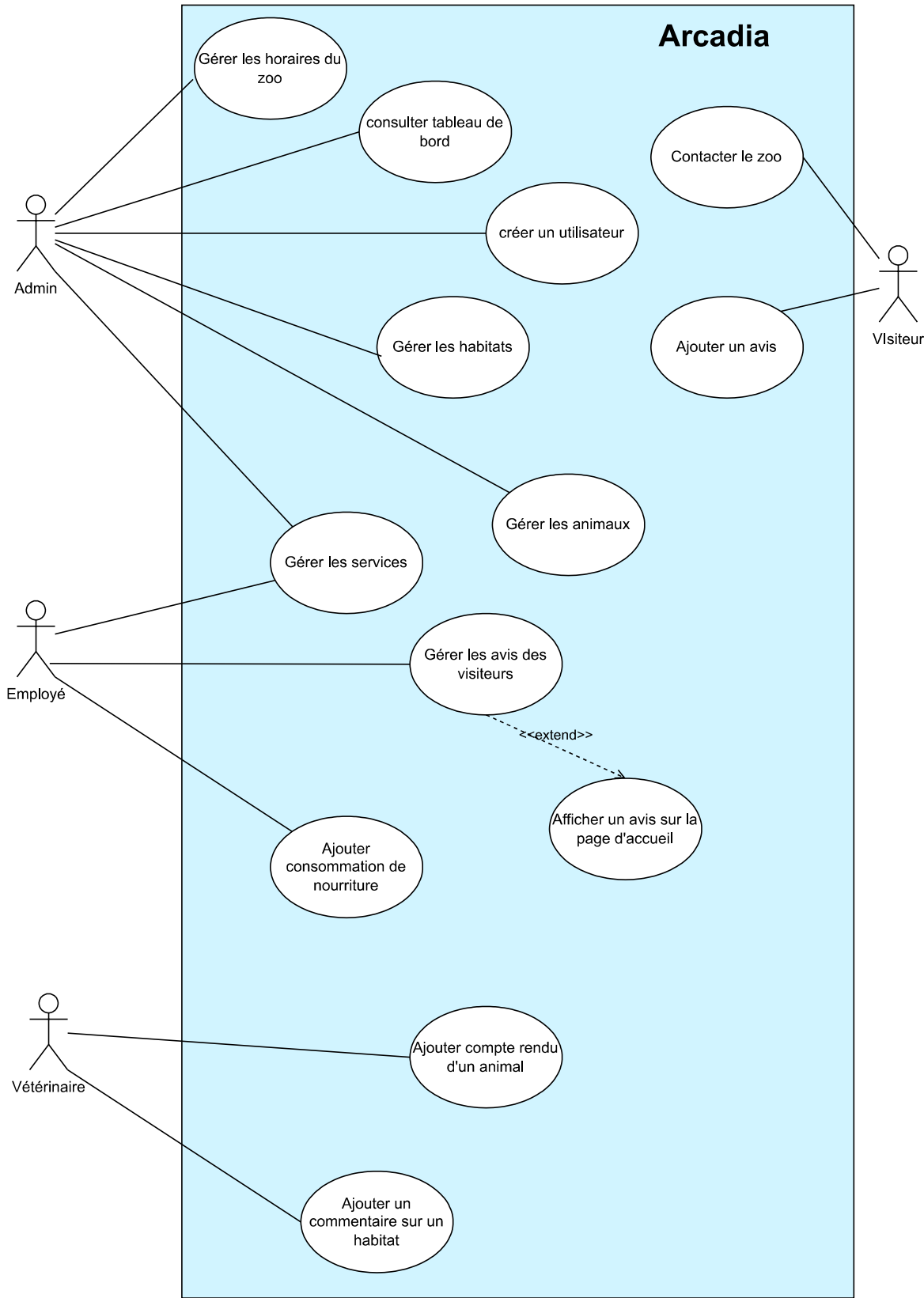


Diagramme de séquence
modifier un service

