

# **documentation technique de l'application**

## **1. Réflexions initiales technologiques**

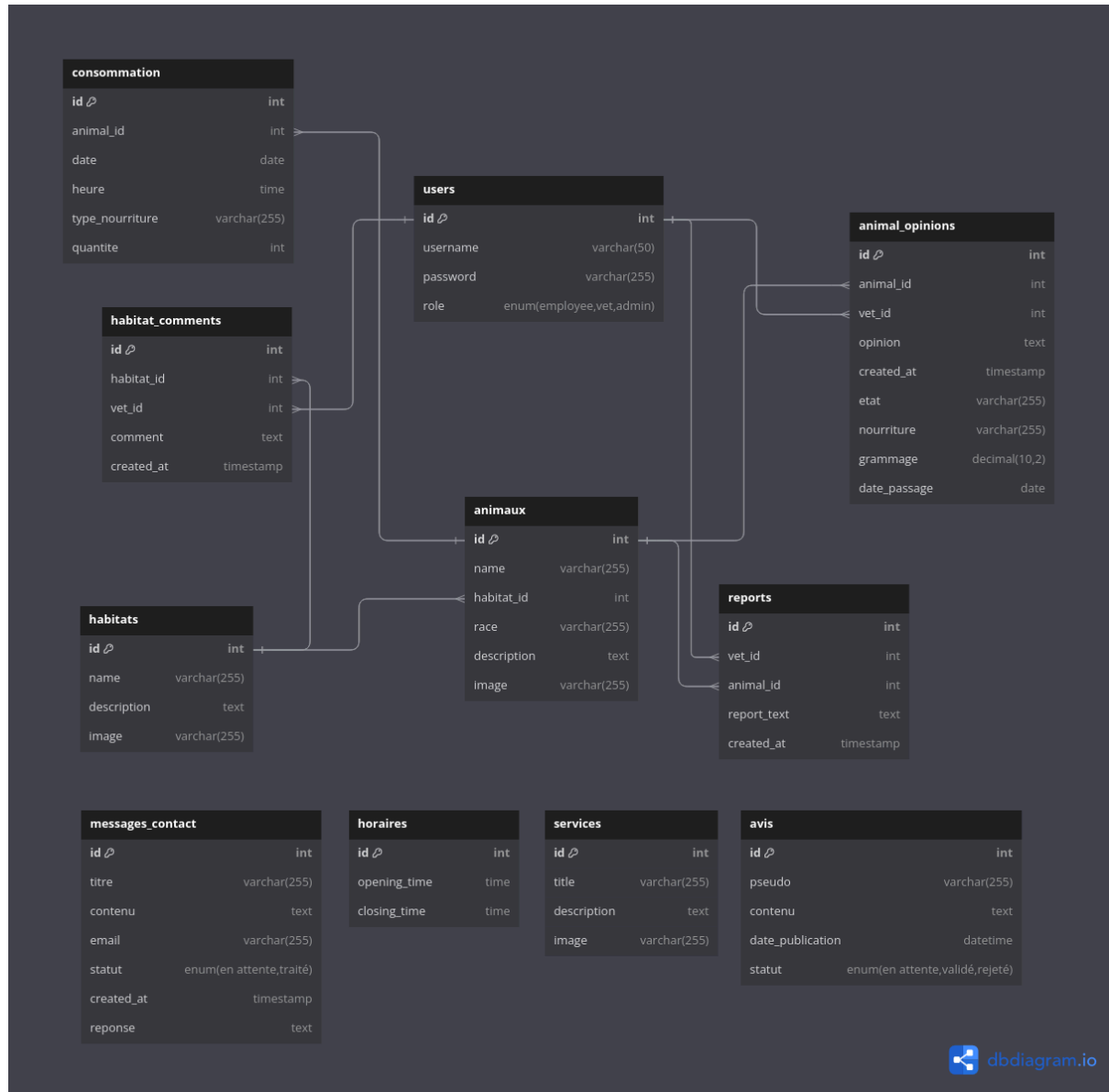
Pour le développement du site, j'ai choisi d'utiliser HTML, CSS et JavaScript pour garantir une interface utilisateur moderne et interactive. PHP et MySQL ont été sélectionnés pour gérer la logique côté serveur et la persistance des données relatives aux animaux, aux réservations, etc. J'ai opté pour Composer comme gestionnaire de dépendances afin d'intégrer facilement des bibliothèques tierces telles que PHPMailer (pour l'envoi d'emails) et MongoDB (pour certaines données non structurées). Le choix de PhpStorm comme IDE nous permet d'optimiser le processus de développement grâce à ses fonctionnalités avancées de débogage et d'édition.

## **2. Configuration de votre environnement de travail**

Pour configurer l'environnement de développement, il est nécessaire d'installer un serveur local tel que XAMPP ou MAMP, qui inclut PHP et MySQL. Une fois installé, clonez le projet depuis le dépôt Git et utilisez Composer pour installer les dépendances nécessaires. Configurez ensuite votre base de données MySQL avec le fichier SQL inclus, puis paramétrez les variables d'environnement pour des services comme PHPMailer et MongoDB installés grâce à Composer.

### 3. Modèle de la base de données (Diagramme ER)

#### Schéma :



#### La structure plus détaillée (format écrit dans dbdiagram.io) :

Table users {id int [pk, increment], username varchar(50), password varchar(255), role enum('employee', 'vet', 'admin')}

Table services {id int [pk, increment], title varchar(255), description text, image varchar(255)}

Table habitats {id int [pk, increment], name varchar(255), description text, image varchar(255)}

Table avis {id int [pk, increment], pseudo varchar(255), contenu text, date\_publication, datetime [null], statut enum('en attente', 'validé', 'rejeté') [null]}

Table horaires {id int [pk, increment], opening\_time time, closing\_time time}

Table messages\_contact {id int [pk, increment], titre varchar(255), contenu text, email varchar(255), statut enum('en attente', 'traité') [null], created\_at timestamp [null], reponse text [null]}

Table animaux {id int [pk, increment], name varchar(255), habitat\_id int [ref: > habitats.id, null], race varchar(255), description text [null], image varchar(255) [null]}

Table animal\_opinions {id int [pk, increment], animal\_id int [ref: > animaux.id, null], vet\_id int [ref: > users.id, null], opinion text, created\_at timestamp [null], etat varchar(255) [null], nourriture varchar(255) [null], grammage decimal(10,2) [null], date\_passage date [null]}

Table consommation {id int [pk, increment], animal\_id int [ref: > animaux.id, null], date date [null], heure time [null], type\_nourriture varchar(255) [null], quantite int [null] }

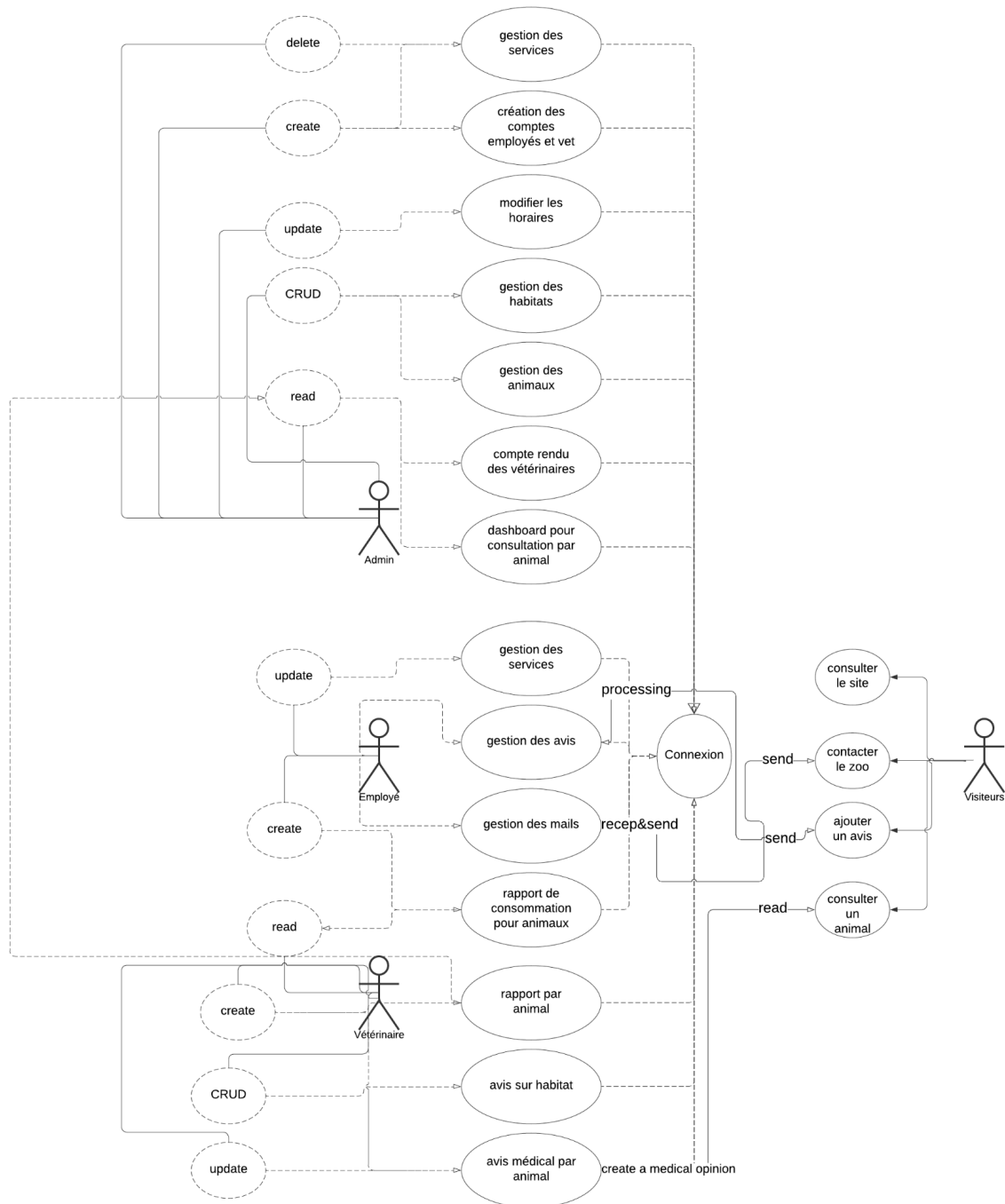
Table habitat\_comments {id int [pk, increment], habitat\_id int [ref: > habitats.id], vet\_id int [ref: > users.id], comment text, created\_at timestamp [null]}

Table reports {id int [pk, increment], vet\_id int [ref: > users.id, null], animal\_id int [ref: > animaux.id, null], report\_text text [null], created\_at timestamp [null]}

Dbdiagram.io m'a permis de schématiser les tables de ma base de donnée ainsi pour bien assimilé la structure lors de sa création.

## 4. Diagramme de cas d'utilisation

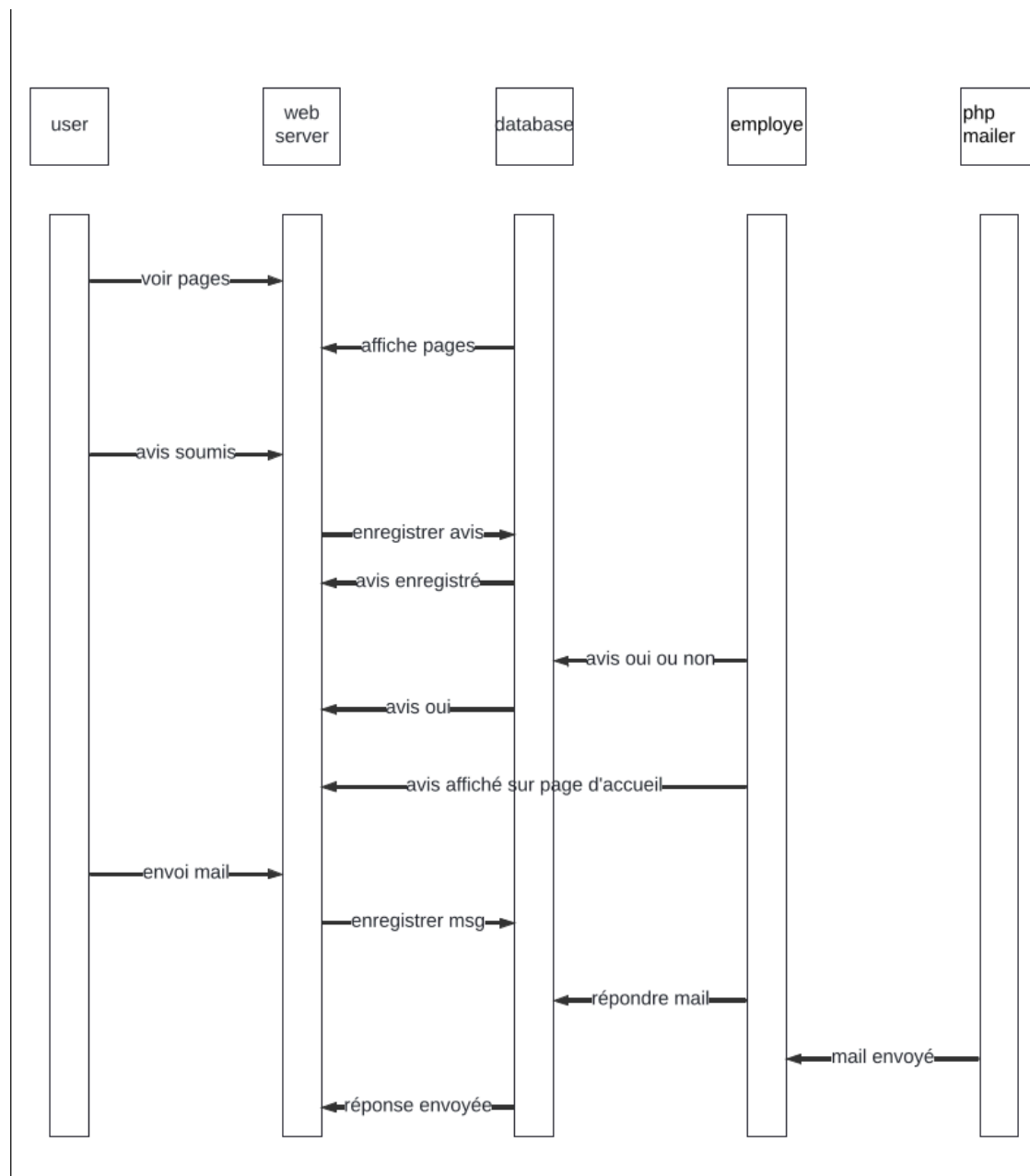
### Schéma :



Le schéma réalisé sur Lucidchart m'a aidé à visualiser chaque fonctionnalité pour chaque acteur en fonction de leurs rôles.

## 5. Diagramme de séquence

schéma :



Le schéma construit sur Lucidchart a pu m'éclairer sur la façon dont la communication se fait entre chaque point, du visiteur (user) au traiteur de mail (phpmailer).

## 6. Documentation du déploiement

Pour déployer l'application sur un serveur de production, commencez par vérifier que toutes les dépendances sont installées à l'aide de `composer install`. Transférez ensuite les fichiers de l'application via FTP ou SSH sur le serveur de production. Configurez les variables d'environnement dans le fichier `.env` pour le serveur de production (`.env` optionnel), notamment pour la base de données et les services comme PHPMailer. Enfin, créez la base de données sur le serveur de production et exécutez le script SQL pour créer les tables nécessaires ou importer les données de la base de donnée.