

ARCADIA ZOO

A photograph of a male lion with a large, full mane, resting in a field of tall, dry, light-colored grass. The lion is positioned in the center-right of the frame, looking slightly towards the camera. The background is a soft-focus landscape of hills under a clear sky.

DOCUMENTATION TECHNIQUE

MANUEL D'UTILISATION

Bienvenue dans l'application web Arcadia Zoo.

Ce manuel d'utilisation vous guidera à travers les fonctionnalités principales de l'application pour différents utilisateurs : Visiteurs, Administrateur, Employés, et Vétérinaires.

Ce manuel est conçu pour assurer une utilisation fluide et efficace de l'application web Arcadia Zoo, en maximisant l'expérience utilisateur pour chaque type d'utilisateur.

1. Accueil et Navigation

- Page d'accueil : Présente les informations générales sur le zoo, les habitats, les services, et les avis des visiteurs.
- Menu de navigation : Accès facile aux différentes sections de l'application : Accueil, Services, Habitats, Connexion, Donate, et Contact.

2. Fonctionnalités pour les Visiteurs

- Consulter les animaux et habitats : Visualisez les différents habitats et les animaux qui y résident. Cliquez sur un habitat pour voir la liste des animaux, puis sur un animal pour voir ses détails et l'état de santé.
- Laisser un avis : Soumettez un avis sur votre visite, qui sera validé par un employé avant d'apparaître sur le site.
- Contacter le zoo : Remplissez un formulaire de contact pour envoyer un message au zoo.
- Donate : Envoyez de l'argent pour soutenir le zoo grâce à la section dédiée aux dons.

3. Espace Administrateur

- Gestion des comptes : Créez et gérez les comptes des employés et des vétérinaires. L'administrateur est le seul à pouvoir effectuer des modifications sur les fonctionnalités et les droits des employés et vétérinaires.
- Modification des services et habitats : Ajoutez, modifiez, ou supprimez des services et des habitats.
- Consultation des comptes rendus vétérinaires : Accédez aux rapports détaillés sur l'état des animaux et filtrez par date ou par animal.
- Dashboard de consultations : Visualisez les statistiques de consultation des animaux pour comprendre les préférences des visiteurs.
- Droits étendus : L'administrateur dispose de droits étendus pour gérer les erreurs des autres comptes et assurer la bonne marche de l'application.

4. Espace Employé

- Validation des avis : Approuvez ou rejetez les avis des visiteurs.
- Gestion des services : Modifiez les services proposés par le zoo.
- Enregistrement de la consommation alimentaire : Ajoutez des entrées sur la nourriture donnée aux animaux, avec date et heure.

5. Espace Vétérinaire

- Enregistrement des comptes rendus : Soumettez des rapports sur l'état de santé des animaux, incluant la nourriture et les détails de chaque passage.
- Commentaire sur les habitats : Ajoutez des commentaires sur l'état des habitats.
- Consultation des historiques alimentaires : Visualisez les entrées alimentaires enregistrées par les employés.

6. Connexion

- Accès restreint : Seuls les administrateurs, employés et vétérinaires peuvent se connecter. Les visiteurs n'ont pas la possibilité de créer un compte.

7. Sécurité et Déploiement

- Sécurité des données : Toutes les informations sont sécurisées grâce aux fonctionnalités de sécurité intégrées de Laravel, telles que la protection CSRF, la validation des entrées, et la gestion des sessions sécurisées.
- Déploiement : Suivez les instructions dans le fichier README.md pour déployer l'application localement.

Les pages externes sont conçues en mobile-first, car plus de 70% des visiteurs utilisent des appareils mobiles. Les dashboards et pages internes sont en version desktop responsive, optimisées pour une utilisation sur ordinateur, mais restent compréhensibles sur des écrans plus petits.

Le site et l'application sont conçus en anglais, avec des plans pour ajouter plusieurs langues à l'avenir.

RÉFLEXIONS TECHNOLOGIQUES

Choix Techniques Front-end

Technologies Utilisées :

- Bootstrap 5.2
- HTML5
- SASS
- JavaScript
- Vite

Raisons et Avantages :

Bootstrap 5.2 :

- Facilité d'utilisation : Bootstrap fournit des composants préconstruits et un système de grille flexible, permettant de développer rapidement des interfaces utilisateurs cohérentes et esthétiques.
- Responsive Design : Avec son système de grille adaptable, Bootstrap garantit que le site s'affiche correctement sur tous les types de périphériques, crucial pour une approche mobile-first.
- Personnalisation : Bien que Bootstrap offre des composants par défaut, il permet également une grande personnalisation, facilitant l'adaptation du design aux besoins spécifiques du projet.

HTML5 :

- Standard Moderne : HTML5 apporte de nouvelles balises et fonctionnalités, améliorant la sémantique et la structure des pages web. Cela permet une meilleure lisibilité du code et une optimisation SEO.
- Compatibilité : Assure une compatibilité optimale avec les navigateurs modernes, garantissant une expérience utilisateur uniforme.

SASS :

- Modularité et Maintenance : SASS permet d'écrire du CSS de manière plus modulaire et maintenable, grâce à des fonctionnalités comme les variables, mixins, et fonctions. Cela simplifie la gestion des styles sur de grands projets.
- Préprocesseur Puissant : La capacité de SASS à être compilé en CSS permet d'utiliser des fonctionnalités avancées tout en garantissant une performance optimale et une compatibilité avec tous les navigateurs.

JavaScript :

- Interactivité : JavaScript est essentiel pour ajouter des fonctionnalités dynamiques et interactives à l'application, enrichissant l'expérience utilisateur.
- Support Étendu : Étant le langage de script le plus largement supporté par les navigateurs, il assure une large accessibilité et compatibilité des fonctionnalités dynamiques.

Vite :

- Développement Rapide : Vite est un outil de build et de développement rapide, optimisé pour les projets modernes. Il offre un temps de démarrage quasi instantané et des rechargements de module ultra-rapides.
- Compilation Efficace : Vite utilise des optimisations avancées pour le bundling et la minification du code, ce qui améliore la performance et la réactivité de l'application.

Choix Techniques Back-end

Technologies Utilisées :

- Laravel 11.9 (Laravel Herd)
- PHP 8.2
- SQLite
- Redis

Raisons et Avantages :

Laravel 11.9 (Laravel Herd) :

- Framework Puissant et Moderne : Laravel est un framework PHP qui offre des outils avancés pour la gestion des routes, des bases de données, des services web et bien plus. Il permet une structure de code propre et maintenable.
- Ecosystème Riche : Laravel bénéficie d'un écosystème riche avec des bibliothèques et des packages qui accélèrent le développement. Il intègre des outils comme Eloquent ORM pour une interaction simplifiée avec la base de données.
- Sécurité : Laravel intègre des fonctionnalités de sécurité robustes comme la protection CSRF, la validation des entrées, et la gestion des sessions sécurisées.

PHP 8.2 :

- Performance et Sécurité : PHP 8.2 apporte des améliorations significatives en termes de performance et de sécurité par rapport aux versions précédentes. Cela permet une exécution plus rapide des scripts et une meilleure gestion des ressources.
- Nouvelles Fonctionnalités : PHP 8.2 introduit de nouvelles fonctionnalités de langage et des améliorations syntaxiques, facilitant l'écriture de code plus clair et plus performant.

SQLite :

- Simplicité et Efficacité : SQLite est une base de données légère et intégrée, parfaite pour les applications de taille moyenne. Elle est facile à configurer et ne nécessite pas de serveur de base de données séparé.
- Compatibilité avec Laravel Herd : Utilisée avec Laravel Herd, SQLite offre une intégration fluide et une performance satisfaisante pour les besoins de l'application.

Redis :

- Base de Données NoSQL : Redis est utilisé pour la gestion des données non relationnelles, offrant une grande flexibilité et des performances élevées pour les opérations de cache et de file d'attente.
- Haute Performance : Redis est réputé pour sa rapidité et son efficacité, ce qui en fait un choix idéal pour améliorer la performance globale de l'application.

Accessibilité et Internationalisation

Langue Utilisée :

- Anglais

Raisons et Avantages :

1. Accessibilité Internationale : L'utilisation de l'anglais, langue internationale, permet de rendre l'application accessible à un public plus large, y compris les touristes et les chercheurs d'informations sur le zoo à l'international.
2. Maîtrise de l'Anglais : Étant donné que l'anglais est largement maîtrisé dans le domaine de la technologie et des affaires, cela facilite la collaboration et l'utilisation par des utilisateurs et des développeurs du monde entier.
3. Plans Futurs : Le site et l'application sont actuellement en anglais, mais des plans sont en place pour ajouter plusieurs langues à l'avenir, ce qui améliorera encore l'accessibilité et l'inclusivité.

Conclusion

L'approche mobile-first adoptée pour les pages externes est basée sur le fait que plus de 70% des visiteurs utilisent des appareils mobiles. Les dashboards et pages internes sont en version desktop responsive, optimisés pour une utilisation sur ordinateur tout en restant accessibles et compréhensibles sur des écrans plus petits.

Les choix technologiques faits pour ce projet assurent une expérience utilisateur optimale, une performance élevée, et une sécurité renforcée grâce aux outils et pratiques modernes intégrés dans le développement.

ENVIRONNEMENT DE DÉVELOPPEMENT

Configuration de mon environnement de travail

Pour développer l'application web Arcadia Zoo, j'ai configuré un environnement de travail optimisé en utilisant Visual Studio Code (VS Code), ainsi que plusieurs outils et extensions pour améliorer la productivité et la qualité du code.

Éditeur de Code : Visual Studio Code (VS Code)

VS Code est un éditeur de code source léger et puissant, largement utilisé dans l'industrie pour ses fonctionnalités robustes et ses capacités d'extension.

Voici une description des extensions et outils utilisés dans mon environnement de travail :

Extensions VS Code

PHP Intelephense :

- Offre des fonctionnalités avancées d'autocomplétion et d'analyse du code PHP, améliorant la productivité et la précision du développement.

SASS :

- Facilite la syntaxe et la compilation SASS, permettant de structurer et gérer les styles de manière modulaire et maintenable.

Laravel Artisan :

- Intègre les commandes Artisan directement dans VS Code, simplifiant la gestion des tâches et des migrations Laravel.

Stylelint :

- Outil de linting pour CSS/SASS, aidant à maintenir des styles cohérents et à repérer les erreurs dans les fichiers de style.

Laravel Extra Intellisense :

- Ajoute des fonctionnalités d'Intellisense supplémentaires pour Laravel, améliorant l'expérience de développement avec des suggestions de code spécifiques à Laravel.

Laravel Go to View :

- Permet de naviguer facilement entre les routes et les vues Laravel, accélérant le développement et la navigation dans le code.

HTMLHint :

- Outil de linting pour HTML, aidant à repérer et corriger les erreurs dans les fichiers HTML.

SCSS IntelliSense :

- Offre des fonctionnalités d'autocomplétion et de validation pour SCSS, améliorant la productivité lors de l'écriture des styles.

Outils Complémentaires

Composer :

- Gestionnaire de dépendances PHP utilisé pour installer et gérer les packages nécessaires au projet Laravel. Il simplifie l'installation et la mise à jour des bibliothèques PHP.

Vite :

- Outil de build moderne pour JavaScript, utilisé pour le développement et la compilation rapide des ressources front-end. Vite permet un rechargement rapide et une optimisation du build, ce qui est essentiel pour une expérience de développement fluide.

GitHub :

- Plateforme de contrôle de version utilisée pour héberger le code source du projet. Elle permet une collaboration facile, le suivi des modifications et la gestion des versions du code.

Laravel Herd :

- Une version de Laravel optimisée pour le développement rapide. Laravel Herd facilite la gestion des environnements de développement, permettant un déploiement et une configuration simplifiés, tout en offrant des outils avancés pour les développeurs Laravel.

Conclusion

En utilisant Visual Studio Code avec les extensions appropriées, Composer pour la gestion des dépendances, Vite pour le build front-end, GitHub pour le contrôle de version, et Laravel Herd pour un développement rapide et efficace, j'ai configuré un environnement de travail robuste et performant.

Cette configuration assure une productivité élevée, une qualité de code maintenable et une expérience de développement fluide pour l'application web Arcadia Zoo.

DÉPLOIEMENT DE L'APPLICATION

Ce document décrit les étapes nécessaires pour déployer l'application web Arcadia Zoo en utilisant Laravel, SQLite, Vite, npm, HTTPS, et un serveur Caddy.

Prérequis

- Serveur web (Caddy)
- PHP 8.2
- SQLite
- Composer
- Node.js et npm
- Git
- Deployer

Étapes de Déploiement

1. Installation de Deployer

- Installer Deployer en tant que dépendance de développement :

composer require --dev deployer/deployer

- Initialiser Deployer :

vendor/bin/dep init

2. Configuration SSH

- Configurer SSH pour une connexion sécurisée au serveur :

• nano -w ~/.ssh/config

• Host * IdentityFile ~/.ssh/id_rsa

- Tester la connexion SSH :

ssh root@innash.net

3. Provisionnement avec Deployer

- Provisionner le serveur avec Deployer :

vendor/bin/dep provision -o remote_user=root

4. Déploiement de l'Application

- Déployer l'application sur le serveur :

```
vendor/bin/dep deploy
```

5. Résolution des Problèmes

- En cas d'erreur 500, vérifier les logs :

```
tail -f storage/logs/laravel.log
```

- Générer une clé d'application Laravel :

```
php artisan key:generate
```

- Régénérer le cache de configuration :

```
php artisan config:cache
```

6. Configuration de la Base de Données SQLite

- Assurer que le fichier de base de données SQLite est créé et accessible :

```
touch /path/to/database.sqlite
```

- Configurer le fichier .env pour utiliser SQLite :

```
DB_CONNECTION=sqlite
```

```
DB_DATABASE=/path/to/database.sqlite
```

7. Configuration de Caddy

- Configurer Caddy pour servir l'application via HTTPS :

```
innash.net { root * /path/to/arcadia-zoo/public encode gzip php_fastcgi unix//run/php/php8.2-fpm.sock file_server }
```

Conclusion

Le déploiement de l'application Arcadia Zoo nécessite plusieurs étapes cruciales incluant la configuration SSH, l'utilisation de Deployer pour le provisionnement et le déploiement, et la configuration du serveur web Caddy. En suivant ces étapes, l'application peut être déployée de manière sécurisée et efficace.

Diagramme de classes

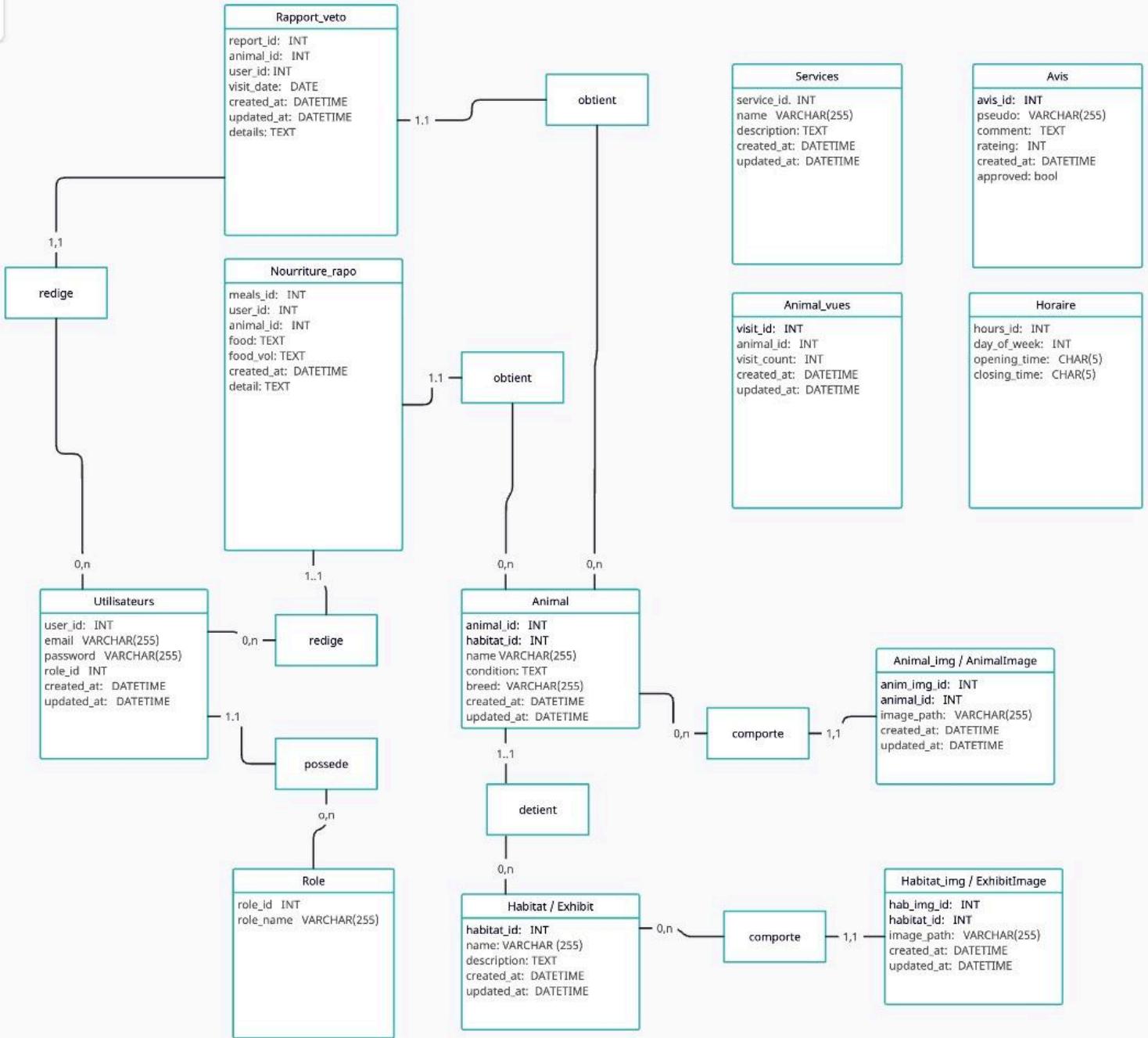


Diagramme de cas d'utilisation

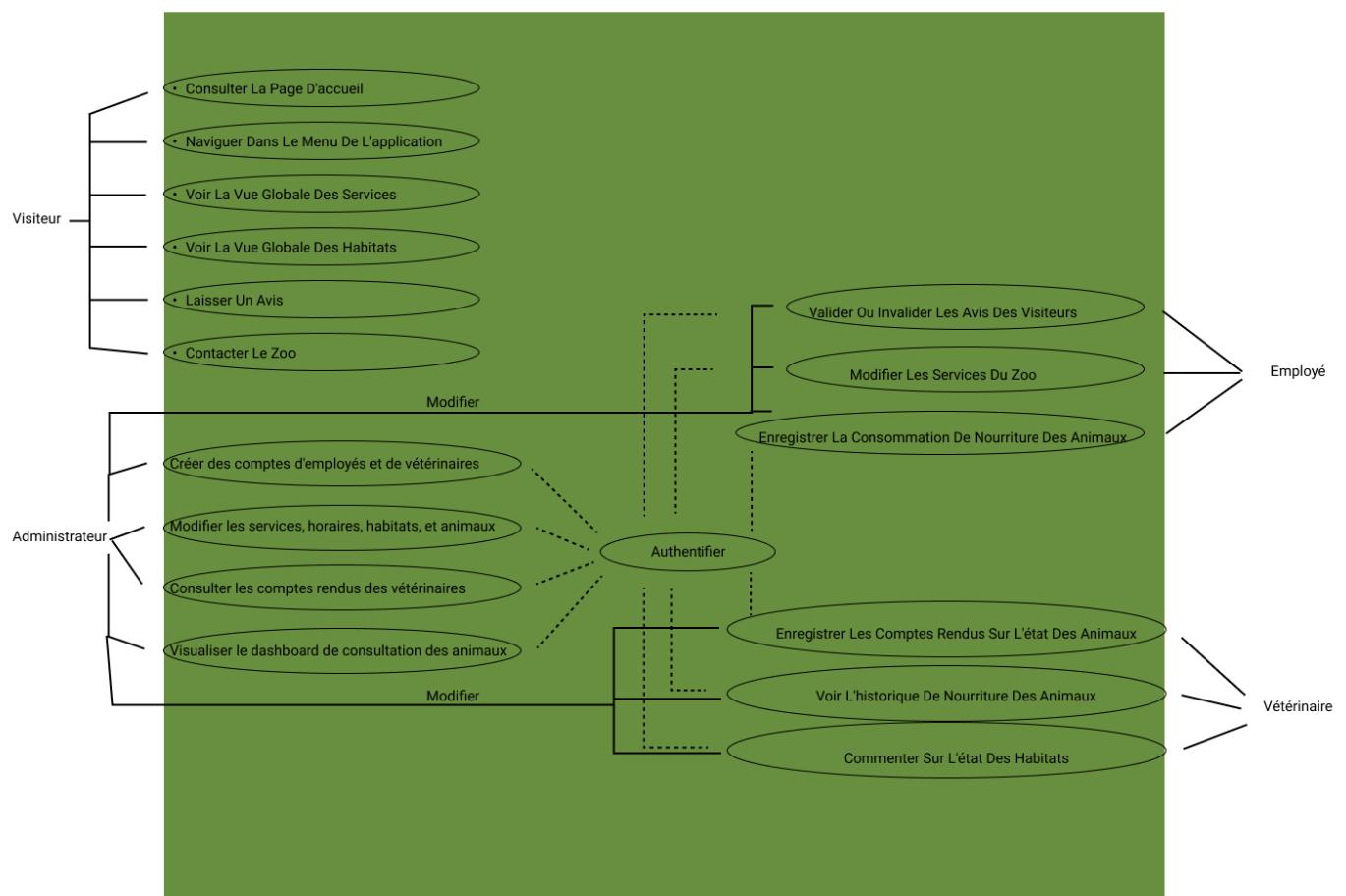


Diagramme de séquences

Consultation des détails d'un animal" par un visiteur.

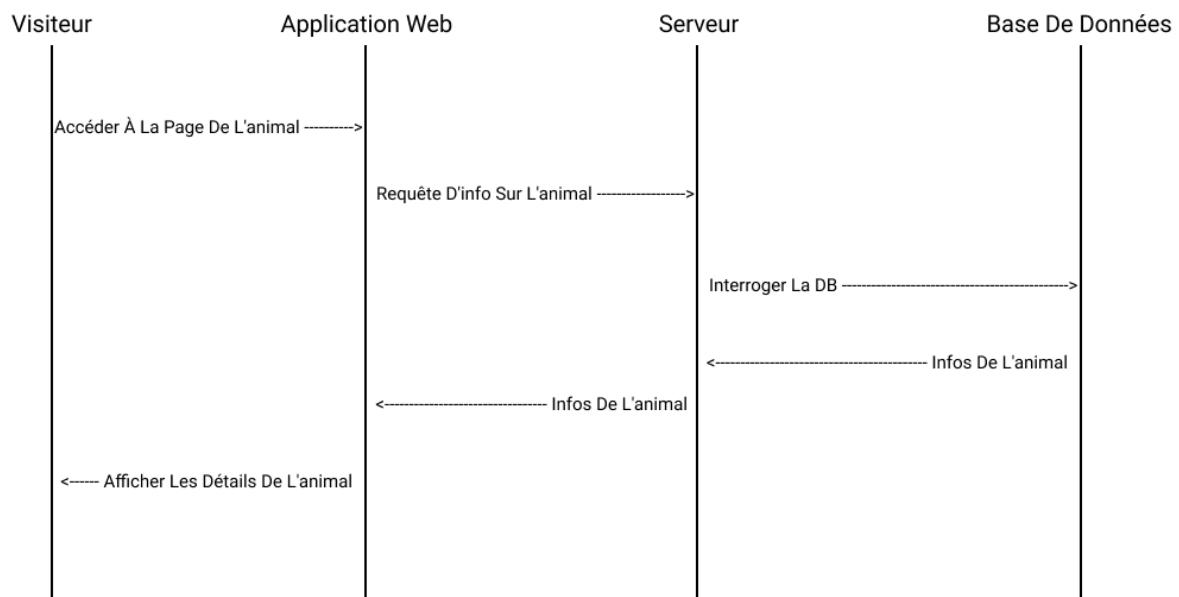
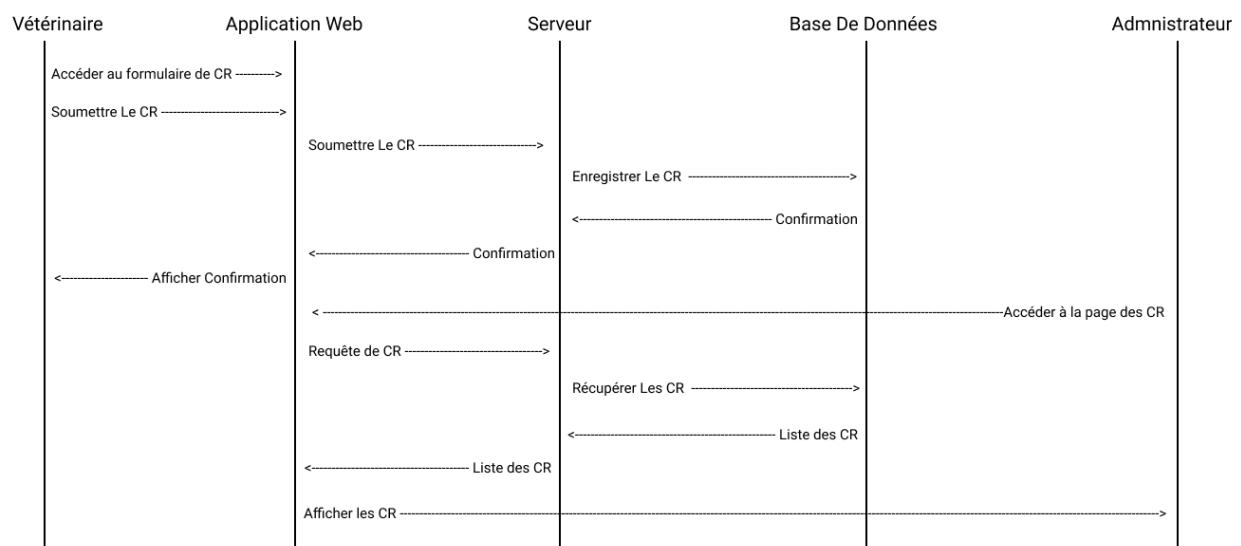


Diagramme de séquences

Enregistrement d'un compte rendu (CR) par le vétérinaire et sa consultation par l'administrateur.



GESTION DU PROJET "ARCADIA ZOO"

Méthode de Gestion : Kanban

Lien : <https://trello.com/b/x0AL6YOB/arcadia>

Pour le développement de l'application web du zoo Arcadia, j'ai utilisé Trello avec la méthode Kanban pour organiser les tâches et suivre l'avancement du projet.

Colonnes du Kanban :

Backlog

- Cette colonne contient toutes les tâches initialement identifiées pour le projet, mais qui ne sont pas encore prêtes à être travaillées. Elle sert de réserve de tâches à prioriser et à planifier.

Design

- Dans cette colonne, les tâches liées à la conception des interfaces utilisateur et à la création des maquettes sont gérées. Cela inclut la création des wireframes, des maquettes visuelles, et des prototypes interactifs.

To Do (À faire)

- Les tâches prêtes à être travaillées sont déplacées ici. Elles sont priorisées et planifiées pour les prochains cycles de travail.

Doing (En cours)

- Les tâches en cours de réalisation par les membres de l'équipe sont placées dans cette colonne. Cela permet de suivre ce qui est actuellement en développement ou en traitement.

Testing (En test)

- Une fois qu'une tâche est complétée, elle passe en phase de test pour validation. Les tests peuvent inclure des tests unitaires, des tests d'intégration, et des tests utilisateurs pour s'assurer de la qualité et de la fonctionnalité.

Done (Terminé)

- Les tâches qui ont passé les tests avec succès et qui sont entièrement terminées sont déplacées dans cette colonne. Cela indique que la fonctionnalité est prête pour le déploiement ou déjà en production.

Avantages de la Méthode Kanban

- Visualisation du flux de travail. La méthode Kanban offre une vue claire et concise de l'état d'avancement du projet, identifiant facilement les goulots d'étranglement et les tâches prioritaires.
- Flexibilité. Cette méthode permet une grande flexibilité et adaptation rapide aux changements, facilitant l'ajustement des priorités en temps réel.
- Transparence et Communication. La gestion transparente des tâches améliore la communication et la collaboration au sein de l'équipe, assurant que tous les membres sont alignés sur les objectifs et les délais du projet.

En utilisant cette structure Kanban, nous pouvons suivre efficacement les progrès, prioriser les tâches et garantir que les livrables sont de haute qualité et livrés dans les délais.