

Présentation de l'entreprise

- **Zoo Arcadia** (depuis 1960, près de la forêt de Brocéliande (Bretagne, France))
- **Panel d'animaux réparti par habitat** (savane, jungle, marais), prennent soin de leur santé
- **Des contrôles quotidiens sont effectués tous les jours par des vétérinaires** sur les animaux avant l'ouverture du zoo (tout va bien, santé ok, nourriture suffisante (calculée selon un grammage précisé par le vétérinaire))
- Très bons résultats financiers, animaux heureux, José (directeur) a de grandes ambitions
- **Application permettant aux visiteurs de visualiser**
 - les animaux, leurs états
 - les services du zoo
 - les horaires du zoo
- Application qui permettra **d'augmenter la notoriété et l'image de marque du zoo**
- **Entreprise DevSoft** (recruté après obtention du diplôme)

Description du projet

Une application web avec **thème et couleurs en lien avec l'écologie**
Zoo indépendant au niveau des énergies
L'utilisateur doit **ressentir les valeurs de l'écologie**

Page d'accueil (Visiteur concerné)

- Présentation du zoo avec images
- Mentionner les différents habitats, services et animaux présents dans le zoo
- Avis du zoo

Menu du site (Visiteur concerné)

- Retour vers page d'accueil
- Accès aux services
- Accès aux habitats
- Connexion (possible uniquement aux vétérinaires, employés et admin)
- Contact

Page Services (Visiteur concerné)

- Vue globale avec interface simple et récapitulative des services proposés par le zoo
- Services configurables depuis l'interface « Admin » :
 - Nom du service
 - Sa description
- Services disponibles :
 - Restauration
 - Visite des habitats avec un guide (gratuit)
 - Visite du zoo en petit train

Page Habitats (Visiteur concerné)

- **Mention de tous les habitats avec les animaux associés**
- **Vue globale des habitats (image et nom)**
- **Au clic de chaque habitat, vue détaillée de l'habitat avec animaux associé et description**
- **Caractéristiques « Habitat » :**
 - **Nom**
 - **Image(s)**
 - **Description**
 - **Liste des animaux associés**
- **Caractéristiques « Animal » :**
 - **Prénom**
 - **Race**
 - **Image(s)**
 - **Habitat affecté**
- **Informations données par le vétérinaire (depuis un espace dédié) sur un animal :**
 - **État de l'animal**
 - **Nourriture proposée et grammage**
 - **Date de passage**
 - **Détail de l'état de l'animal (facultatif)**
- **Au clic sur un animal, le visiteur doit pouvoir visualiser :**
 - **Détail de l'animal (propriétés)**
 - **Avis du vétérinaire**

Page Avis (Visiteur concerné)

- **Un visiteur peut laisser un commentaire**
- **Commentaire contenant pseudo et un champ texte « Avis »**
- **Avis soumis à validation pour un employé (depuis espace employé) qui pourra autoriser ou non la publication de l'avis (sur cette même page)**

Page Administrateur (Admin concerné)

- Création de compte « employé » et « vétérinaire » en fournissant mail (username) et mdp
- L'utilisateur du compte recevra un mail avec son username mais pas le mdp (il devra le demander à l'admin)
- L'admin peut modifier services, horaires, habitats, animaux (création, m.à.j, suppression)
- Emplacement avec tous les comptes-rendus des vétérinaires (filtres à ajouter pour trier en fonction d'un animal ou d'une date)
- Dashboard pour voir le nombre de consultations/animal
- Compte à créer mais impossible de créer un compte Admin depuis l'application

Page Employé (Employé concerné)

Depuis cet espace, un employé peut :

- Valider ou non un avis
- Modifier les services du zoo
- Comme il s'occupe de donner l'alimentation quotidienne aux animaux, il doit pouvoir sélectionner un animal et lui ajouter une conso de nourriture (sélectionner date, heure, nourriture et quantité)

Page Vétérinaire (Vétérinaire concerné)

- Remplir les comptes-rendus par animaux
- Possibilité de mettre un commentaire sur les habitats (donner son avis sur son état et une potentielle amélioration)
- Vue de ce que chaque animal a mangé (grâce à la saisie de l'employé)

Page Connexion (Admin, vétérinaire et employé concernés)

- Seuls l'admin, le vétérinaire et l'employé peuvent se connecter
- Pour se connecter : rentrer username (mail) et mdp
- Page indisponible pour un visiteur (création de compte impossible)

Page Contact (Visiteur concerné)

- **Contacter le zoo depuis cette page**
- **Formulaire avec titre, description (message) et mail (pour recevoir une réponse)**
- **Message envoyé par mail au zoo** (l'employé peut répondre directement par mail)

Page Statistiques sur la consultation des habitats (Visiteur concerné)

- **+1 consultation quand un visiteur cliquera sur un animal**
- **Info stockée dans une BDD non relationnelle**
Par ex, compteur d'un animal passe à 2 quand son profil est visité (s'il était à 1 avant)
- **Données exploitées dans le Dashboard Admin** (voir quels animaux plaisent le plus)

Schéma BDD

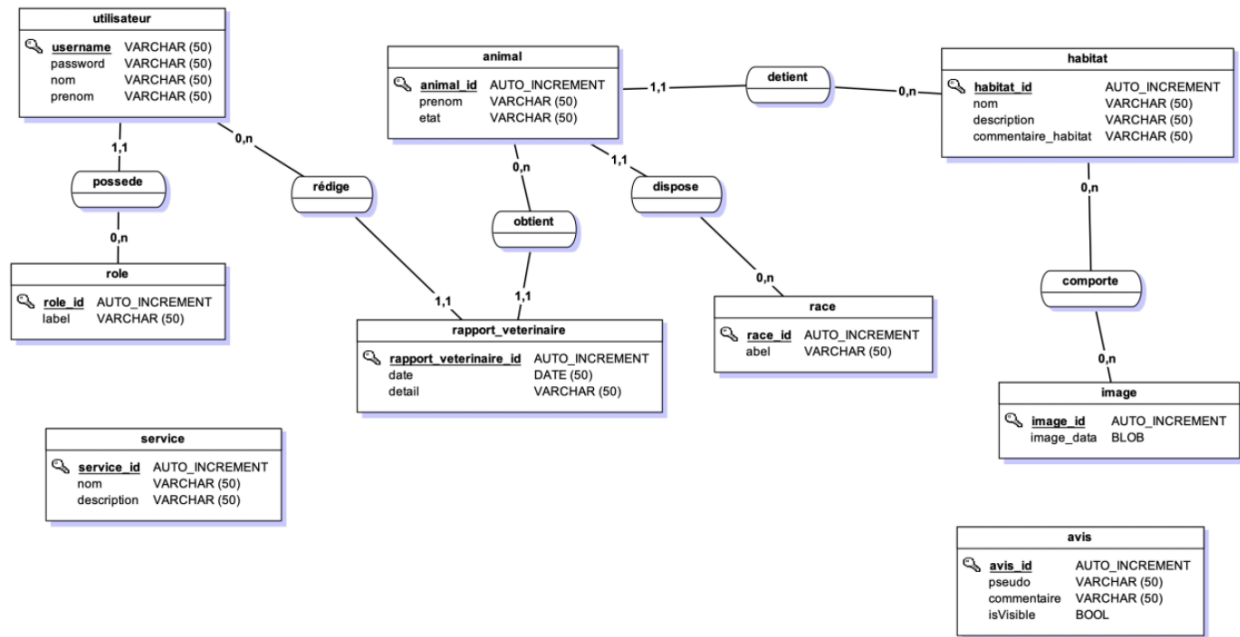


Figure 1 : base de données de l'application

Exemple de Stack technique possible

Front : HTML, CSS (Bootstrap), JS

Back : PHP avec PDO (PHP Data Objects)

BDD relationnelle : MySQL, MariaDB, PostgreSQL

BDD NoSQL : MongoDB

Déploiement : Fly.io, Heroku, Azure, Vercel

Objectifs

- Passer sur chaque étape nécessaire à l'élaboration de la solution :
 - Analyse des besoins
 - Maquettage
 - Intégration
 - Développement des règles de gestion
 - Déploiement
- Dans la copie, détailler :
 - Dispositions prises pour la sécurité
 - Choix techniques (les justifier et exposer point de vue sur différentes techno)
- Validation bloc Front :
 - Pages « Accueil », « Menu », « Services », « Habitats », « Avis » et « Contact »
- Validation bloc Back :
 - Pages « Services », « Habitats », « Admin », « Employé », « Vétérinaire », « Connexion » et « Statistiques »

Livrables

- Lien dépôt(s) GitHub (public) *1
- Lien application déployée
- Lien logiciel gestion de projet (Trello, Notion...) **

* Contenu du Git

- Fichier README.md expliquant la démarche à suivre pour déployer l'application en local
- Bonnes pratiques de Git :
 - Branche principale
 - Branche de développement : chaque fonctionnalité dans une branche issue de la branche de développement puis mergée dans cette dernière après test) + branche développement mergée dans branche principale quand tests ok
- Fichiers de création de BDD et intégration de données (fichier SQL)
- Manuel d'utilisation en PDF :
 - Présenter l'application
 - Donner des id pour tests
- Charte graphique en PDF :
 - Palette de couleurs et polices utilisées
 - Export des maquettes attendues (wireframes et mockup)
(3 maquettes bureautiques et 3 mobiles)
- Documentation sur la gestion de projet (l'expliquer)
- Documentation technique de l'application :
 - Réflexions initiales technologiques
 - Configuration environnement de travail
 - Modèle conceptuel de données (ou diagramme de classe)
 - Diagramme d'utilisation et diagramme de séquence
 - Documentation du déploiement de l'application avec démarches et étapes suivies

** Contenu de la gestion de projet (sous la forme d'un kanban (Trello))

- Colonne des fonctionnalités prévues et ordonnées par priorité
- Colonne des fonctionnalités que je compte faire
- Colonne des fonctionnalités que je fais actuellement
- Colonne des fonctionnalités terminées (branche de développement)
- Colonne des fonctionnalités mergées sur la branche principale