ECF Projet Arcadia

Le zoo Arcadia est un zoo situé au Nord de la france en Bretagne, cotoyé chaque année par plus de 200K visiteurs essentiellement francais, dans le zoo le panel d’animaux est grand, et ceux ci sont repartis par habitats (savane, jungle, marais).

les veterinaires du zoo font extrement attention a la santé des animaux, chaque jour ils viennent faire des controles sur chaque animal, avant l’ouverture du zoo.

Le propriétaire José a envie d’une application web qui permettrai aux visiteurs de visualiser les animaux, leurs états et visualiser les services ainsi que les horaires du zoo.

José souhaite avoir une application web dont les couleurs et le thème font penser à de l’écologie.

Le zoo est construit a base de materiaux naturels, l’energie qui fait fonctionner l’emsemble du zoo est naturelle, et redistribuent l’energie en trop le long des sentiers du zoo.

Conception de l’application

Apres avoir pris connaissances des besoins utilisateurs, j’ai redessiné sur support papier les scenarios de chaque fonctionnalitées de l’application.

Afin d’avoir un idée visuel de l’application j’ai réalisé une framework ainsi qu’une maquette des differentes pages du site qui seront accessibles par les visiteurs.

Intégration des fonctionnalités suivantes : API interne pour le suivi du traffic publique du site,

billeterie en ligne

la page d’accueil (visiteurs)

doit comporter une presentation do zoo et quelques images.

Mention des differents habitats, services, ainsi que les animaux possédé par le zoo.

Les avis du zoo.

Le menu de l’application (visiteurs)

Point de depart de navigation dans l’application contient :

- vue globale des pages du site.

- retour vers la page d’accueil

- acces a tous les services

- acces a tout les habitats

- mini-jeu sur l’ecologie

- connexion (possible uniquement pour les vétérinaires, employé et administrateur)

- contact

La vue globale de tous les services (visiteurs)

proposer une interface simple et recapitulative de tous les services que propose le parc.

Les services doivent être configurable depuis l’espace “Administrateur”.

Un service a un nom, une description.

Listes des services :

- restauration

- visite des habitats avec un guide (gratuit)

- visite du zoo en petit train

Vue globale des habitats (visiteurs)

Mention de tous les habitats que propose le zoo et les animaux associées (affiche que l’image et le nom), puis au clic, l’element affiche le detail de l’habitat ainsi que le detail de chaque animal lorsque la souris les survolent.

Habitat : nom, 1,2 images, description, liste d’animaux.

Animal : prenom, race, 1,2 images, habitat affecté.

Fonctionalité UX (visiteurs)

Lors du passage du veterinaire, il saisie depuis son espace veterinaire, letat de l’animal, la nourriture proposé, le grammage de la nourriture, la derniere de passage, le détail de l’etat de l’animal (facultatif)

Fonctionalité UI (visiteurs)

Ainsi depuis la page global des habitat le visiteur doit pouvoir visualiser les details de l’animal, mais aussi l’avis du veterinaire.

Fonctionnalité UX (visiteurs)

Un visiteur, peut laisser un commentaire depuis la page d’accueil, ce qui affichera sur le site juste un pseudo ainsi qu’un champ “avis” sous forme d’un texte. (Necessite validation de l’employé pour etre publié)

Depuis l’espace employé, le veterinaire pourra autoriser ou non l’avis a apparaitre sur la page d’accueil.

Espace administrateur (Administrateur)

L’admin, peut créer un compte de type “employé” et “vétérinaire”,en fournissant un courriel

( nom d’utilisateur du compte), ainsi qu’un mot de passe.

Réception par mail dans la boite mail de l’employé concerné, précisant son mail, sans le mot de passe, qui lui devra etre demandé en contactant l’administrateur.

L’admin Peut modifier les services, horaires, habitats, et animaux du zoo (creation mise a jour, et suppression, ( CRUD ) )

Dans l’espace admin, emplacement de tous les comptes rendu du veterinaire, avec possibilité d’appliquer filtres de recherches par animal ou par dates.

Dashboard affichant le nombre de consultation par animal par les visiteurs.

Enfin la creation de compte admin ne doit pas etre possible depuis l’application.

Espace employé (employé)

Un employé depuis son espace, peut valider ou invalider un avis, peut aussi modifier les services du zoo.

L’employé s’occupe de l’alimentation quotidienne des animaux, il doit pouvoir selectionner un aniaml et lui ajouter une consommation de nourriture, pour ajouter une consommation, il doit fournir la date, l’heure, puis la nourriture donnée , ainsi que la quantité.

Espace vétérinaire (veterinaire)

Le veterinaire passe quotidiennement dans le zoo, il remplira les compte rendu par animaux.

Peut mettre un commentaire sur les habitats afin qu’il/elle puisse donner son avis son etat, et s’il faut l’ameliorer ou non.

Peut voir depuis sont espace veterinaire et par animal tout ce que l’anial a pu manger vus la saisie de l’employé sur son espace employé.

Connexion (admin, veterinaire, employé)

Seul une personne de type administrateur, vétérinaire ou encore employé peut se connecter.

Un visiteur ne peut pas devenir utilisateur, et donc, ne peux pas se crée de compte.

Se connecter : saisir son nom d’utilisateur (mail) suivi de mot de passe.

Contact (visiteur)

Un visiteur peut contacter le zoo s’il le souhaite.

Acceder a la page contact, puis un formulaire va demander un titre, un description,ainsi qu’un mail afin d’obtenir une réponse.

la demande est envoyée par mail au zoo, et la réponse d’un employé peut se faire par mail aussi.

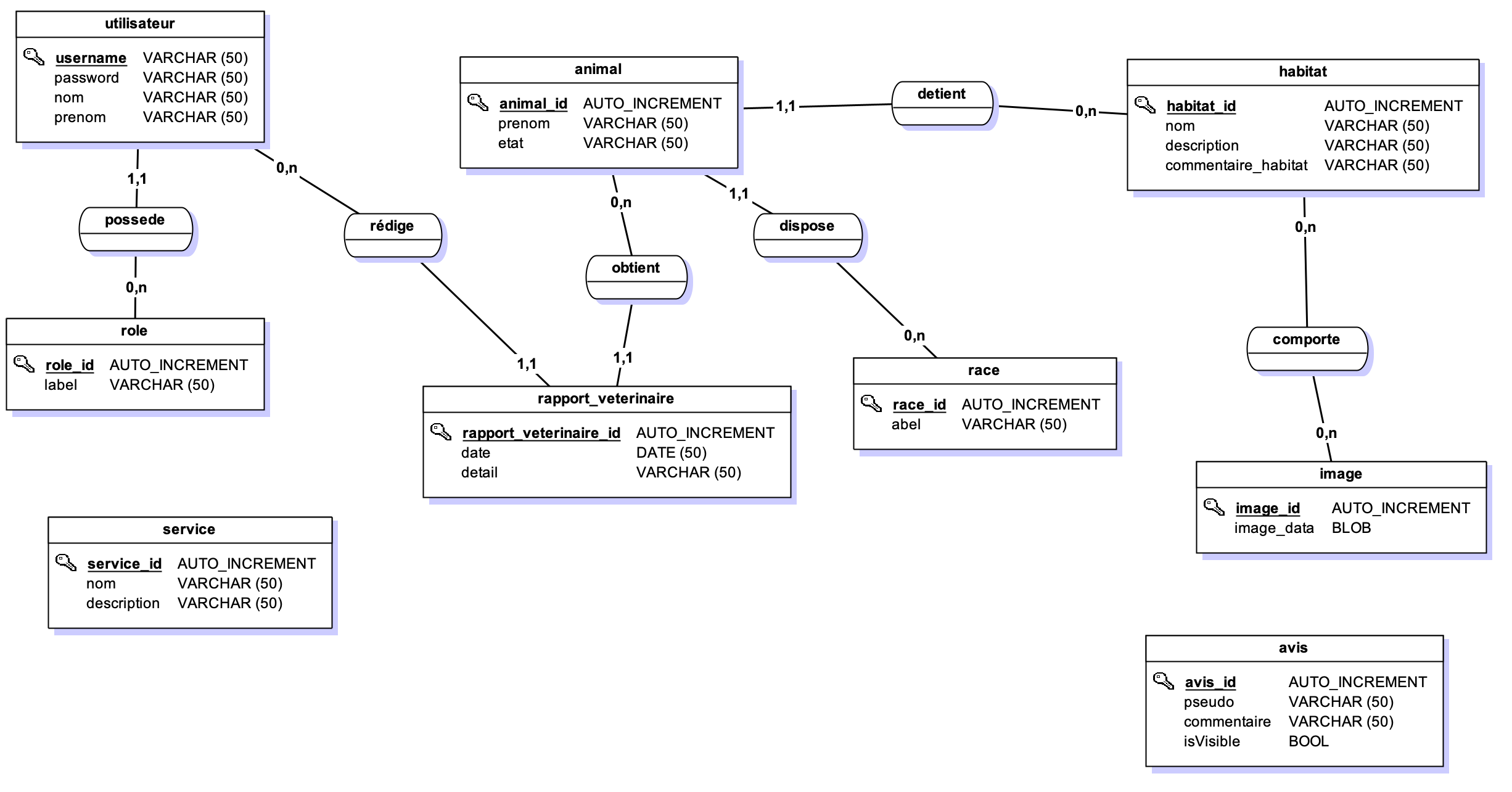
Statistique sur la consultation des habitats (visiteur)

Pour le visiteur, le fait de cliquer sur un animal, fera augmenter la consultation de 1 pour l’animal donné.

Cette information devra être stocké dans une base de données non relationnelle.

Ces données sont exploitées dans le Dashboard administrateur afin que José puisse visualiser

quels animaux plaisent le plus.

Schema de la base de données :

Stack technique :

front : HTML 5, CSS <= 5.3.6, JS

Back-end : PhpMyAdmin utilisation de PDO

BDD : relationnelle : MySQL

BDD noSQL : MongoDB

deploiement