

## TP – Développeur Web et Web Mobile

### Énoncé

**Durée indicative :** 70 h

**Documents autorisés :** annexes

**Matériel autorisé :** calculatrice, tableur, etc.

**Votre examen comporte :**



- ✓ Cet **énoncé** qui vous présente le sujet de l'épreuve
- ✓ Une **copie à rendre** (Excel ou Word) que vous devez télécharger, remplir informatiquement et déposer dans l'espace de dépôt prévu à cet effet.

**Renommer votre copie à rendre Word ou Excel comme suit :**

TP\_DWWM\_Sept/Oct/Nov/Déc24/Janv/Févr/Mars/Avril25\_copiearendre\_**NOM\_Prenom**



**Objectifs de l'évaluation :**

L'évaluation en cours de formation que vous allez réaliser a pour vocation de figurer dans votre **livret d'évaluation**. Il sera donc remis à votre jury le jour des épreuves du titre professionnel accompagné de votre évaluation et du sujet initial.



Nous vous demandons de vous mettre en situation d'examen. Prenez le temps de lire le sujet et de manipuler les annexes afin de répondre en situation professionnelle aux questions et problématiques évoquées dans le sujet

### À vous de jouer !

## Compétences du référentiel concernées

Toutes les compétences du titre professionnel développeur Web et Web Mobile seront analysées grâce à ce projet.

### Activité – Type 1 : Développer la partie front-end d'une application web ou web mobile sécurisée

- ➔ Installer et configurer son environnement de travail en fonction du projet web ou web mobile
- ➔ Maquetter des interfaces utilisateur web ou web mobile
- ➔ Réaliser des interfaces utilisateur statiques web ou web mobile
- ➔ Développer la partie dynamique des interfaces utilisateur web ou web mobile

### Activité – Type 2 : Développer la partie back-end d'une application web ou web mobile sécurisée

- ➔ Mettre en place une base de données relationnelle
- ➔ Développer des composants d'accès aux données SQL et NoSQL
- ➔ Développer des composants métier coté serveur
- ➔ Documenter le déploiement d'une application dynamique web ou web mobile

## Présentation de l'entreprise

Arcadia est un zoo situé en France près de la forêt de Brocéliande, en Bretagne depuis 1960. Ils possèdent tout un panel d'animaux, réparti par habitat (savane, jungle, marais) et font extrêmement attention à leurs santé. Chaque jour, plusieurs vétérinaires viennent afin d'effectuer les contrôles sur chaque animal avant l'ouverture du zoo afin de s'assurer que tout se passe bien, de même, toute la nourriture donnée est calculée afin d'avoir le bon grammage (le bon grammage est précisé dans le rapport du vétérinaire).

Le zoo, se porte très bien financièrement, les animaux sont heureux. Cela fait la fierté de son directeur, José, qui a de grandes ambitions.

A ce jour, l'informatique et lui ça fait deux, mais, il a envie d'une application web qui **permettra aux visiteurs de visualiser les animaux, leurs états et visualiser les services ainsi que les horaires du zoo.** Ne sachant pas comment faire, José a demandé à Josette, son assistante, de trouver une entreprise lui permettant d'obtenir cette application et augmenter ainsi la notoriété et l'image de marque du zoo.

Après l'obtention de votre diplôme, vous avez été embauché au sein de DevSoft, entreprise qui a été choisi par Josette afin de réaliser l'application.

## Description du projet

Après avoir été missionné par DevSoft pour travailler sur ce projet, José se permet de venir vous consulter afin de vous donner une information importante : **"Je souhaite avoir une application web dont les couleurs et le thème font penser à de l'écologie,** le zoo est entièrement indépendant au niveau des énergies et il faut que lorsqu'un utilisateur consulte l'application, il ressente les valeurs de l'écologie dont nous sommes fiers"

Puis, il se met à vous exposer le besoin :

## US 1 : Page d'accueil

*Utilisateur concerné : Visiteur*

La page d'accueil doit comporter :

- Présentation du zoo en y incorporant quelques images
- Mentionnez les différents habitats, services ainsi que les animaux que possède le zoo
- Les avis du Zoo

## US 2 : Menu de l'application

*Utilisateur concerné : Visiteur*

Le menu est essentiel à l'application, il permet de faciliter la navigation et de fluidifier le trafic, ainsi, il doit composer au minimum :

- Retour vers la page d'accueil
- Accès à tous les services
- Accès à tous les habitats
- Connexion (la connexion sera possible uniquement pour les vétérinaires, employé et administrateur)
- Contact

## US 3 : vue globale de tous les services

*Utilisateur concerné : Visiteur*

Une vue globale est nécessaire afin de proposer une interface simple et récapitulative de tous les services que propose le parc.

Les services doivent être configurable depuis l'espace "Administrateur", un service dispose des caractéristiques suivantes :

- Un nom
- Une description

Pour le moment, le parc dispose de plusieurs services : restauration, visite des habitats avec un guide (gratuit), visite du zoo en petit train

## US 4 : vue globale des habitats

*Utilisateur concerné : Visiteur*

Cette page, doit mentionner tous les habitats que propose le zoo et les animaux associés. José vous propose une vue tout d'abord de tous les habitats (en affichant que l'image et le nom), puis, au clic sur celui-ci, on affiche le détail avec les animaux ainsi que la description.

Un habitat est caractérisé par :

- Un nom
- Une ou des images
- Une description de l'habitat
- Une liste d'animaux

Un animal est caractérisé par :

- Un Prénom
- Une race
- Une ou des images
- Un habitat où il est affecté

Le vétérinaire, passe régulièrement et saisie des informations depuis son espace sur un animal donné en mentionnant :

- L'état de l'animal
- La nourriture proposée
- Le grammage de la nourriture
- Date de passage
- Détail de l'état de l'animal (information facultative)

Il est très important pour José de mentionner l'état de l'animal sur cette page.

Enfin, au clic sur un animal de l'habitat, le visiteur doit pouvoir visualiser le détail de l'animal, ce qui inclue ses propriétés mais également l'avis du vétérinaire.

## US 5 : Avis

*Utilisateur concerné : Visiteur*

Un visiteur, peut laisser un commentaire s'il le désire. Ce commentaire, contiendra juste un pseudo ainsi qu'un champ "avis" texte. Cet avis sera ensuite soumis à validation par l'employé. Depuis l'espace employé, il pourra ensuite autoriser ou non l'avis à apparaître sur la page d'accueil. La soumission d'un avis se fait par cette même page.

## US 6 : Espace Administrateur

*Utilisateur concerné : Administrateur*

L'admin, peut créer un compte de type "employé" et "vétérinaire", il doit pour cela, fournir un courriel (qui sera l'username) ainsi qu'un mot de passe.

L'utilisateur en question, va recevoir un mail avec son username, cependant, le mot de passe n'est pas communiqué dans le mail. Il devra se rapprocher de l'administrateur afin de l'obtenir.

Il peut également modifier les services, horaire, habitat et animaux du zoo (création, mise à jour et suppression).

Ensuite, l'espace administrateur doit avoir un emplacement avec tous les comptes rendus du vétérinaire, des filtres sont à positionner afin de pouvoir trier et filtrer les comptes rendus sur un animal ou une date.

Enfin, il doit pouvoir avoir un Dashboard lui montrant le nombre de consultation par animal (ce besoin est décrit en US 11)

José précise que vous devez lui créer ce compte et qu'il ne doit pas être possible de créer un compte Administrateur depuis l'application.

## US 7 : Espace Employé

*Utilisateur concerné : employé*

Un employé depuis son espace, peut valider un avis ou l'invalidier. Il peut également modifier les services du zoo.

L'employé, va donner l'alimentation quotidienne aux animaux, il va ainsi depuis son espace, pouvoir sélectionner un animal et lui ajouter une consommation de nourriture, pour cela il va devoir donner la date ainsi que l'heure, puis, la nourriture donnée ainsi que la quantité.

## US 8 : Espace Vétérinaire

*Utilisateur concerné : vétérinaire*

Un vétérinaire passe quotidiennement dans le zoo, ainsi, depuis son espace, il remplira les comptes rendus par animaux. Le détail est disponible en US 4

Il peut en plus, mettre un commentaire sur les habitats afin qu'il/elle puisse donner son avis son état et s'il faut l'améliorer ou non.

Le vétérinaire voit également sur son espace et par animal, tout ce que l'animal a pu manger via la saisie de l'employé sur son espace.

## US 9 : Connexion

*Utilisateur concerné : Administrateur, vétérinaire, employé*

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

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

Pour se connecter, il devra juste saisir son username (mail) suivi de son mot de passe.

## US 10 : Contact

*Utilisateur concerné : Visiteur*

Un visiteur peut contacter le zoo s'il le souhaite, pour cela, il devra accéder à la page contact depuis le menu applicatif. Pour donner suite à cela, il aura accès à un formulaire qui va lui demander un titre, une description ainsi que son mail afin qu'il puisse obtenir une réponse.

Pour donner suite à cet envoi, la demande est envoyée par mail au zoo. L'employé peut répondre à la demande directement par mail.

## US 11 : Statistique sur la consultation des habitats

*Utilisateur concerné : Visiteur*

Un visiteur, quand il cliquera sur l'animal, va augmenter la consultation de 1 pour l'animal donné.

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

Par exemple, si un visiteur consulte l'animal portant le prénom "Médor", alors le compteur de "Médor" passera à 2 (s'il avait eu 1 avant).

Ces données sont exploitées dans le Dashboard administrateur afin que José puisse visualiser quels animaux plaisent le plus.

José conclue sur le fait que vous êtes dans l'obligation de déployer votre / vos applications.

Un schéma de données possible vous a été communiqué en annexe 1, vous pouvez l'implémenter, ou, faire le vôtre.

## Annexe 1 : schémas de données

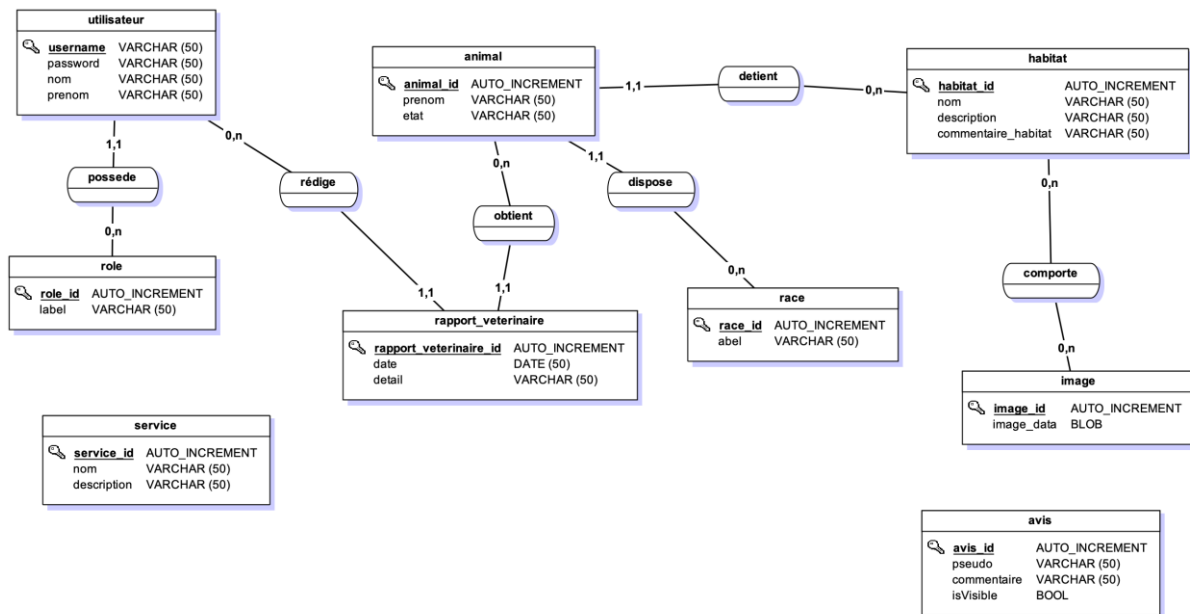


Figure 1 : base de données de l'application

## Stack technique

Aucune technologie n'est obligatoire pour cet ECF, à l'exception de l'utilisation d'une base de données relationnelle et non relationnelle.

Voici un exemple de stack technique possible :

- Front : HTML 5, CSS (Bootstrap), JS
- Back-end : PHP avec utilisation de PDO
- Base de données relationnelle : MySQL, MariaDB ou PostgreSQL
- Base de données NoSQL : MongoDB
- Déploiement : fly.io, Heroku, Azure, vercel



## Objectif & Livrables

L'objectif de cet ECF est que vous passiez sur chaque étape nécessaire à l'élaboration d'une solution de qualité : analyse des besoins, maquettage, intégration, développement des règles de gestion ainsi que le déploiement.

Vous devrez détailler dans votre copie les dispositions que vous avez prises notamment sur la sécurité. De plus, vous devrez justifier tous vos choix techniques : il est très important de pouvoir les justifier et exposer votre point de vue sur différentes technologies.

La validation par bloc étant possible, voici les US que vous devez effectuer pour valider les différents blocs.

- Activité type 1 : US 1,2,3,4,5 et 10
- Activité type 2 : US 3,4,6,7,8,9,11

Les livrables de ce sujet, sont les suivants :

- ➔ Le lien du (ou des) dépôt(s) GitHub **PUBLIC** où sera présent le code de votre application
- ➔ Le lien de votre / vos applications déployés
- ➔ Le lien vers votre logiciel de gestion de projet (Jira, Notion, Trello, etc.)

Votre git, devra comporter :

- Un fichier README.md contenant la démarche à suivre afin de déployer votre application en local
- Les bonnes pratiques de git doivent être appliquées
  - Une branche principale
  - Une branche de développement
    - Chaque fonctionnalité sera une branche issue de la branche développement, après test, le merge sera effectué vers la branche développement
    - Une fois que la branche développement est correctement testée, il faudra effectuer un merge vers la branche principale
- Les fichiers de création de base de données mais également d'intégration de données (cela doit être un fichier SQL). (L'utilisation de fixture et / ou de migration n'implique pas que vous maîtrisez le SQL)
- Manuel d'utilisation en format PDF

- Il doit présenter l'application et donner des identifiants afin de réaliser les différents parcours possibles.
- Une charte graphique en format PDF
  - Palette de couleurs utilisé ainsi que la police
  - L'export des maquettes attendues (wireframes & mockup) (3 maquettes bureautiques et 3 maquettes mobiles)
- Une documentation traitant de votre gestion de projet
  - Explication de votre gestion de projet
- Une documentation technique de votre application
  - Réflexions initiale technologique sur le sujet
  - Configuration de votre environnement de travail
  - Modèle conceptuel de données (ou diagramme de classe)
  - Diagramme d'utilisation ainsi que le diagramme de séquence
  - Documentation du déploiement de votre application expliquant votre démarche ainsi que les différentes étapes.

Votre gestion de projet, devra être sous la forme d'un kanban partagé afin d'avoir une vue globale de votre réalisation. Il devra comprendre :

- Une colonne recensant toutes les fonctionnalités prévues et ordonnée par priorité
- Une colonne recensant les fonctionnalités que vous comptez faire, prévu de développement (ou dans le sprint si vous êtes en agile)
- Une colonne qui recensant les fonctionnalités que vous faites actuellement
- Une colonne qui énumère les fonctionnalités terminées (sur la branche de développement)
- Enfin, vous pouvez effectuer une dernière colonne stipulant les fonctionnalités qui ont été merge dans la branche principale