

NOM	Gil Martinez
Prénom	Juan Esteban
Date de naissance	06/03/2005

Copie à rendre

TP – Développeur Web et Web Mobile

Documents à compléter et à rendre

Lien du git : <https://github.com/juan-gil-05/EcoRide>

Lien de l'outil de gestion de projet :

<https://trello.com/invite/b/673a49b2994c6744b91097a6/ATTI72494569d4fba1e48cf14a237b619ed4364F5111/ecf-ecoride>

Lien du déploiement : <https://ecoride.juangil.fr>

Login et mot de passe administrateur :

Login = admin@gmail.com | Mot de passe = 123456789Admin.

SANS CES ELEMENTS, VOTRE COPIE SERA REJETEE

Partie 1 : Analyse des besoins

1. Effectuez un résumé du projet en français d'une longueur d'environ 20 lignes soit 200 à 250 mots

Le projet se base en la création d'une application web pour la startup « EcoRide », une startup qui propose aux personnes de participer et créer des covoiturages, afin de réduire l'impact environnemental.

Dans un contexte mondial marqué par le réchauffement climatique et la hausse des coûts de transport, EcoRide propose une alternative écologique, économique et collaborative aux trajets individuels en voiture.

C'est pour cela que José, directeur technique de la startup a décidé de concevoir une application intuitive, simple, efficace et accessible à tous les utilisateurs qu'ils soient passagers ou conducteurs.

Les conducteurs peuvent créer des trajets, ajouter différentes voitures, définir leurs préférences, visualiser les avis laissés par les passagers et d'autres fonctionnalités.

Les passagers, de leur côté, peuvent rechercher des trajets, réserver leur place, noter les conducteurs et laisser des commentaires si le covoiturage s'est mal passé. Afin de garantir la fiabilité des services, une équipe d'employées qui est constamment attentive aux suivis des trajets en covoiturages a accès à une interface de modération pour surveiller les avis et signalements.

De la même manière, la plateforme EcoRide intègre une interface administrateur, permettant à José de consulter des graphiques dynamiques avec différentes statistiques de l'application, de modérer les comptes utilisateurs et employées et de suivre la croissance de la plateforme.

2. Exprimez le cahier des charges, l'expression du besoin ou les spécifications fonctionnelles du projet

Le projet EcoRide a pour but de concevoir une application web des covoiturages avec une approche écologique, permettant de mettre en relation des passagers et conducteurs souhaitant partager un trajet en voiture, dans le but de réduire l'empreinte carbone et les coûts de déplacement.

Expression du besoin

La startup souhaite disposer de :

- Une interface dynamique et responsive qui s'adapte à tous les écrans.
- Une présentation de la startup avec des images et les mentions légales du site web.
- Une page qui affiche tous les covoiturages trouvés après la recherche avec certaines informations du trajet et du conducteur.
- Une page de contact, afin de pouvoir envoyer un message à l'administrateur de la startup.
- Un système de création de compte et de connexion, afin d'accéder à son espace personnel.
- Un système pour filtrer la recherche des covoiturages, afin de trouver uniquement les trajets correspondant aux besoins du passager.
- Une page qui présente tous les détails d'un covoiturage précis, pour que le passager puisse les consulter.
- Un système de création, recherche et réservations des trajets en covoiturage.
- Différents rôles pour les comptes utilisateurs tels que passager, chauffeur, passager-chauffeur, employé et admin.
- Une page pour visualiser les covoiturages dans lesquels l'utilisateur participe en tant que passager et en tant que conducteur.
- Un système pour démarrer, annuler ou indiquer comme terminé un covoiturage.
- Un système de notification par email si le conducteur annule un covoiturage et si le conducteur indique que le covoiturage est terminé, c'est-à-dire qu'il est arrivé à la destination finale.
- Un espace employé dédié à la modération des avis et signalements sur les covoiturages.
- Un espace administrateur qui permet de gérer les comptes utilisateurs et suivre les statistiques via des graphiques dynamiques.

Spécifications fonctionnelles principales :

- Gestion des rôles : visiteur, utilisateur (passager / conducteur), employé et administrateur.
- Concernant les covoiturages :
 - Création
 - Recherche
 - Recherche avec des filtres
 - Participer
 - Démarrer
 - Indiquer comme terminé
 - Annuler le trajet
 - Quitter le covoiturage
 - Valider un covoiturage après participation
- Soumettre des avis et notes du conducteur
- Laisser un commentaire si le covoiturage s'est mal passé
- Gestion des voitures par le chauffeur
- Gestion des préférences personnelles par le chauffeur
- Envoi d'e-mails aux participants du covoiturage SI :
 - Le conducteur annule le trajet
 - Le conducteur indique que le covoiturage est terminé
- Gestion des avis et signalements par l'employé
- Gestion des comptes utilisateurs par l'administrateur :
 - Pour créer un nouveau compte d'employé
 - Pour suspendre des comptes aussi bien utilisateur et employé
- Affichage des graphiques dynamiques par l'administrateur :
 - Nombre des covoiturages par jour
 - Crédits gagnés par la plateforme par jour
 - Nombre des crédits totaux gagnés par la plateforme

Voir le tableau Kanban avec la gestion de projet où vous pouvez trouver le Cahier des charges exprimé plus en détail.

Partie 2 : Spécifications technique

1. Spécifiez les technologies que vous avez utilisé en justifiant les conditions d'utilisation et pourquoi le choix de ses éléments

Pour le Frontend j'ai utilisé :

HTML pour générer la structure du site web.

CSS avec le préprocesseur **Sass** pour ajouter du style d'une façon plus organisé, par exemple avec la possibilité d'indenter le code.

Le Framework **Bootstrap** afin d'accélérer par exemple la création des modales, le responsive du site et également pour la personnalisation que j'ai donnée avec Sass, par exemple pour les couleurs du site.

Le gestionnaire des paquets **npm** (Node Package Manager) pour l'installation de Bootstrap et Bootstrap Icons, afin d'avoir les paquets directement sur ma machine et de cette façon pouvoir personnaliser Bootstrap avec Sass et améliorer la performance.

JavaScript pour rendre le site dynamique, par exemple pour cacher des éléments après avoir cliqué un bouton, aussi pour faire des requêtes AJAX avec l'API fetch, afin d'éviter le rechargement de la page lors d'un envoi d'un formulaire par exemple.

La bibliothèque **SweetAlert2** pour gérer l'affichage des alertes aux utilisateurs, je l'ai utilisée car elle a une bonne documentation est c'est très flexible.

La bibliothèque **Chart.js** pour gérer les graphiques dynamiques dans l'espace administrateur, je l'ai utilisée pour sa facilité à implémenter et toutes les fonctionnalités qui propose.

La bibliothèque **DataTables** pour gérer le tableau des utilisateurs dans l'espace admin, je l'ai utilisée car elle propose des fonctionnes avancées comme le tri, recherche, la pagination, entre autres.

Pour le Backend j'ai utilisé :

PHP pure, sans Framework et en version 8.4.5, j'ai utilisé PHP car c'est le langage proposé dans le programme de formation pour développer le backend, en plus de cela c'est un langage conçu spécifiquement pour le web qui s'intègre parfaitement avec html, css et js, également, il a une grande communauté, par conséquent il y beaucoup de documentation dans le web qui facilite l'utilisation du PHP. J'ai décidé de ne pas utiliser un Framework parce que je voulais vraiment comprendre les bases du développement coté serveur, comme par exemple, le modèle MVC ou les

concepts de la POO, ou encore l'utilisation des requêtes SQL sans un ORM, afin d'approfondir mes connaissances.

Le gestionnaire des dépendances **Composer** afin d'installer différentes bibliothèques comme PHPMailer, phpdotenv, entre autres. Je l'ai aussi utilisée pour le chargement automatique qui permet de ne pas avoir à faire des 'require' dans tous les fichiers PHP.

La bibliothèque **PHPMailer** pour l'envoi d'emails, je l'ai utilisé car elle implémente un serveur SMTP qui facilite l'envoi d'emails et aussi parce qu'elle utilise une authentification et chiffrement SSL/TLS qui rendre les emails plus sécurisés et moins vulnérables aux spam.

La bibliothèque **PHPdotenv** avec l'objectif d'utiliser des variables d'environnement pour enregistrer les données sensibles, par exemple les paramètres de connexion aux bases des données afin de minimiser les risques de sécurité.

Pour les bases des données j'ai utilisé :

MySQL avec **MySQL Workbench** pour la base de données relationnelle, dans laquelle j'ai enregistré différentes données qui se relient entre elles, comme des covoiturages qu'ont une relation avec un conducteur et un passager, entre autres relations. J'ai choisi MySQL car c'est l'un des SGBD le plus utilisé dans le monde, et par conséquent il a une très bonne documentation, est supporté par presque tous les hébergeurs web et il a une syntaxe simple et facile à utiliser.

MongoDB avec **MongoDB Compass** pour la base de données NOSQL, avec laquelle j'ai géré la fonctionnalité des avis laissés par les passagers. J'ai choisi MongoDB car il utilise une syntaxe JSON pour l'enregistrement des documents, syntaxe avec laquelle je suis déjà familiarisé, et en plus de cela c'était le SGBD proposé dans la médiathèque Studi. C'est aussi bien utile pour le déploiement car il offre un hébergement cloud officielle, **MongoDB Atlas**, en facilitant l'étape du déploiement de l'application web.

La **bibliothèque MongoDB pour PHP**, afin d'établir la connexion avec la base de données depuis PHP.

L'extension **MongoDB PHP Driver** pour pouvoir se communiquer nativement avec MongoDB.

2. Comment avez-vous mis en place votre environnement de travail ? Justifiez vos choix. (README.md)

Étant donné que ma machine personnelle a le système d'exploitation MacOS, j'ai installé MAMP qu'intègre un serveur web Apache, un système de gestion des bases de données MySQL et le langage PHP, avec cette base je pouvais déjà commencer le développement en local, cependant, il me fallait aussi un IDE (environnement de développement), ce pour cela que j'ai téléchargé Visual Studio Code. Ensuite, j'ai créé un dossier sur ma machine afin de commencer le projet.

Après cela j'ai initialisé mon repository git dans le dossier et sur GitHub, afin d'avoir enregistré toute la trace du développement du projet. Après j'ai installé le Framework Bootstrap via npm, en suit j'ai installé le préprocesseur SASS via npm, pour pouvoir implémenter du code sass sur le projet. J'ai importé tous les font nécessaires pour le projet. Et je démarré le développement Frontend de quelques pages en suivant les maquettes.

Dans VSC j'ai installé les extensions suivantes afin d'améliorer le processus de développement :

- Live Sass Compiler, pour compiler en temps réel le code sass en css.
- Prettier – Code formatter, pour maintenir le code 'propre' d'une façon rapide et automatique.
- Material Icon Theme, pour afficher des icones dans les dossiers du projet, afin de les identifier facilement.
- Error lens, pour afficher les lignes d'erreurs directement dans le code.
- Image preview, pour voir un aperçu des images dans le code.

3. Énumérez les mécanismes de sécurité que vous avez mis en place, aussi bien sur vos formulaires que sur les composants front-end ainsi que back-end.

- Requêtes préparées pour éviter les injection SQL.
- Vérification des champs des tous les formulaires.
- Hashage des mots de passe pour ne pas les enregistrer en clair dans la base de données
- Déconnexion automatique après inactivité prolongée
- Vérification de connexion et rôles dans les routes du site web, par exemple : uniquement un utilisateur avec le rôle d'administrateur peut accéder à la page avec les graphiques dynamiques de la plateforme.
- Vérification des images mise en ligne pour le profil des conducteur, enregistrement en base de données avec des clés uniques, afin d'éviter des erreurs avec des noms des images identiques.
- Implémentation du fichier .env qui permet de protéger les informations sensibles
- Chiffrer les identifiants transmis via l'URL, afin d'éviter qu'ils soient visibles ou compréhensibles par l'utilisateur.

4. Décrivez une veille technologique que vous avez effectuée, sur les vulnérabilités de sécurité.

J'ai effectué une veille technologique sur les vulnérabilités de sécurité web, avec le top 10 de l'OWASP (Open Web Application Security), qui est une organisation internationale qui regroupe les plus critiques failles de sécurité dans les sites web. Je trouve que tous les points sont très intéressants, et importants à mettre en place, cependant, je me suis principalement concentré aux failles suivantes :

Le contrôle d'accès rompu, afin d'éviter que les utilisateurs puissent se connecter au site web avec les privilèges d'un autre rôle à celui qu'ils ont, par exemple, avec le rôle d'administrateur. Pour se protéger contre cela je sais qu'idéalement il faut utiliser des jetons d'authentification, cependant, à cause de ne pas avoir assez de temps, je me suis permis de le faire de cette façon : avant que l'utilisateur accède à une des pages du site web, j'ai fait une vérification côté serveur pour bien valider qu'il est connecté et qu'il a le rôle nécessaire pour y accéder.

Les injections, avec l'objectif de protéger les données qui sont enregistrées, j'ai utilisé des requêtes préparées qui préviennent que l'utilisateur puisse récupérer du code dans les requêtes des formulaires par exemple.

Défauts de cryptage, avec le chiffrement des mots de passes et des paramètres avec d'information sensible passées dans l'url, comme : pour voir le détail d'un covoiturage, j'envoie l'identifiant du covoiturage dans l'url, cependant, il est chiffré et par conséquent il est illisible pour l'utilisateur.

J'ai aussi lu un blog du gouvernement français, concernant la sécurité dans les mots de passe, donc, j'ai implémenté des vérifications au moment de créer un nouveau mot de passe, avec l'utilisation des regex pour demander à l'utilisateur de renseigner un mot de passe sécurisé. Voici le lien (<https://www.francenum.gouv.fr/magazine-du-numerique/combien-de-temps-un-pirate-met-il-pour-trouver-votre-mot-de-passe-comment>)

Pour ma veille, j'ai utilisé :

- Des blogs sur internet.
- Des vidéos sur YouTube qui expliquent les différentes vulnérabilités avec d'exemples.
- Le contenu de la médiathèque Studi.
- Le site web Mozilla Developer Network (MDN)

Je suis conscient que je peux améliorer plusieurs aspects concernant la sécurité dans le projet, tels que l'utilisation des jetons d'authentification pour bien valider que c'est l'utilisateur correct qui a fait la demande au serveur, ou limiter les tentatives de connexion en utilisant des identifiants incorrects, entre autres.

1. Décrivez une situation de travail ayant nécessité une recherche durant le projet à partir de site anglophone. N'oubliez pas de citer la source

La plupart de contenu et des documentations je les consulte en anglais, parce que je trouve qu'il a beaucoup plus d'information et qui est plus claire, également pour les vidéo tutoriels sur les différentes technologies, la grande majorité de contenu est anglophone.

Par exemple pour implémenter la connexion de mon projet a une base de données MongoDB je suivi la documentation officielle de MongoDB en anglais depuis le site : (<https://www.mongodb.com/docs/php-library/current/get-started/>).

Également, pour la création des tables dans l'espace administrateur, j'ai utilisé directement la documentation de la bibliothèque JS DataTable : (<https://datatables.net/manual/installation>).

De la même façon, je me suis renseigné pour l'utilisation de la bibliothèque ChartJS avec cette série des vidéos sur YouTube qui est complètement en anglais : (<https://www.youtube.com/playlist?list=PLc1g3vwxhg1Xr2r2VodhuthKo0GC1c7V8>)

2. Mentionnez l'extrait du site anglophone qui vous a aidé dans la question précédente en effectuant une traduction en français.

L'extrait suivant m'aider à établir la connexion avec la base de données MongoDB :

Anglais

Download and Install

Install dependencies:

Before you begin developing, ensure that you have the following dependencies installed on your local machine:

PHP version 7.4 or later

Composer version 2.0 or later

Install the MongoDB PHP extension

Run the following command to install the mongodb PHP extension:

Sudo pecl install mongodb

Update your PHP configuration file

To enable the mongodb extension in your PHP configuration file, add the following line to the top of your php.ini file:

Extension=mongodb.so

Tip

You can locate your php.ini. file by running the following command in your shell.

Install the MongoDB PHP Library

To install the MongoDB Library, run the following command in your directory:

```
composer require mongodb/mongodb
```

After installing the library, include Composer's autoload.php file by adding the following code to the top of your file:

```
require_once '/vendor/autoload.php';
```

Français

Télécharger et installer

Installation des dépendances

Avant que vous commenciez à développer, assurez-vous d'avoir les dépendances suivantes installées sur votre machine :

PHP version 7.4 ou ultérieur

Composer version 2.0 ou ultérieur

Installation de MongoDB PHP extension

Exécutez la commande suivante pour installer l'extension mongodb PHP

```
sudo pecl install mongodb
```

Modifiez votre fichier de configuration PHP

Pour activer l'extension mongodb dans votre fichier de configuration PHP, ajoutez la ligne suivante au début de votre fichier php.ini :

```
Extension=mongodb.so
```

Tip

Vous pouvez localiser votre fichier php.ini en exécutant la commande suivante

Dans votre terminal :

```
php -ini
```

Installer la bibliothèque MongoDB PHP

Pour installer la MongoDB PHP, exécutez la commande suivante dans votre répertoire (à la racine du projet)

```
Composer require mongodb/mongodb
```

Après avoir installé la bibliothèque, importez le fichier de Composer autoload.php en ajoutant le code suivant au début de votre fichier :

```
require_once '/vendor/autoload.php';
```

Partie 4 : Informations complémentaire

1. Autres ressources

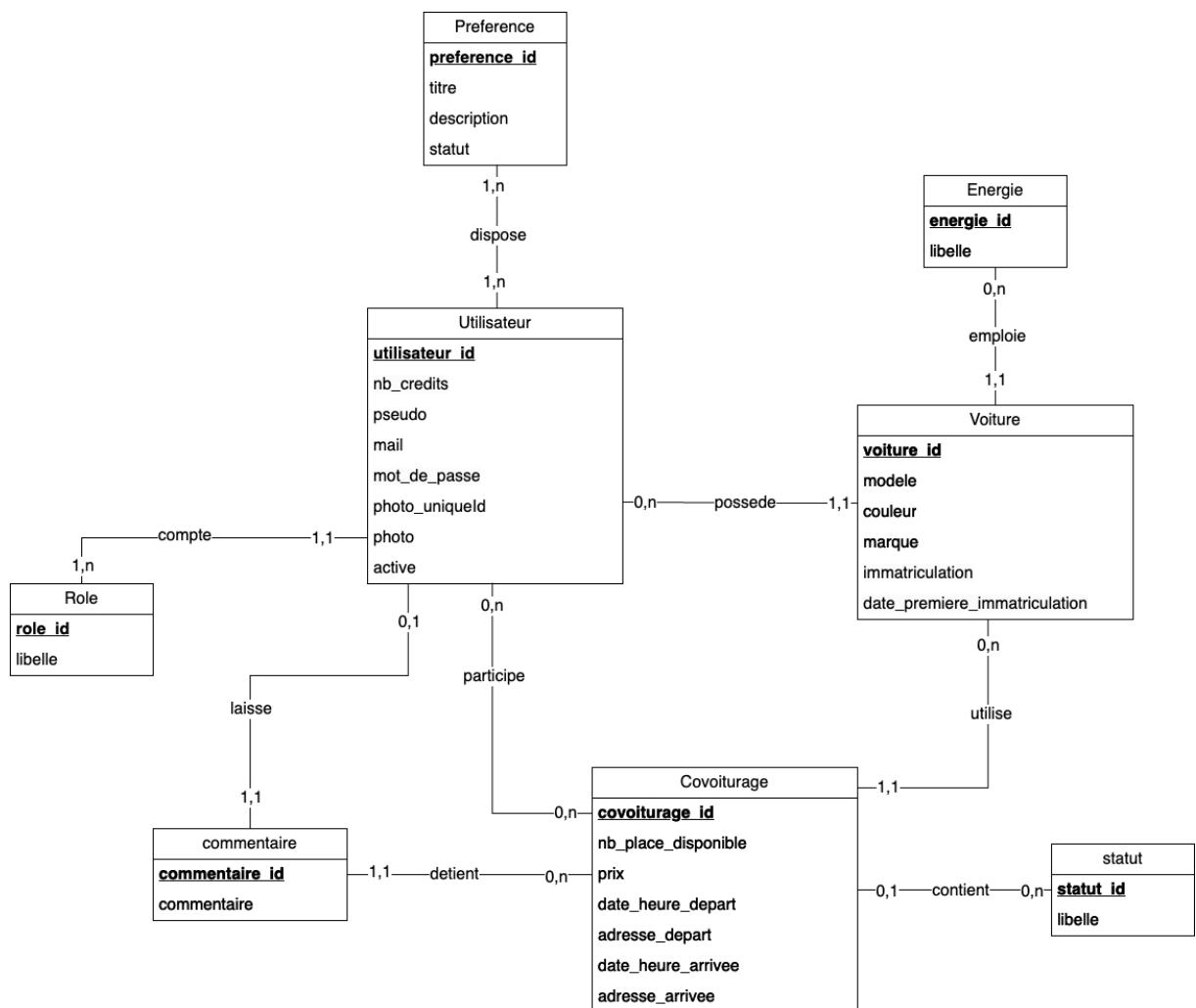
Dans le repository GitHub j'ai mis en pièce jointe dans le dossier : documentation/

- Un fichier .sql avec plusieurs requêtes qui son été utilisées pour la gestion de la base de données MySQL

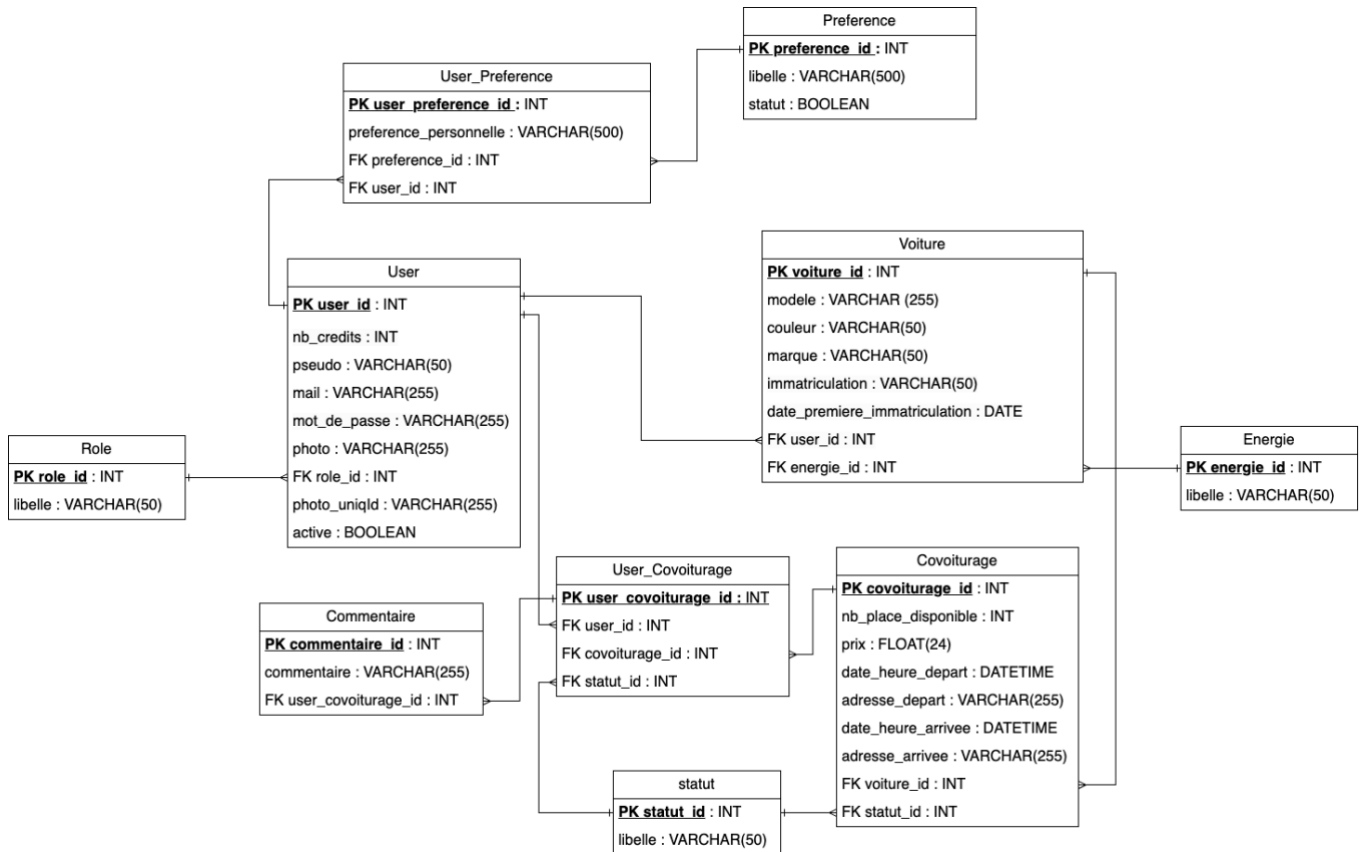
- Le manuel d'utilisation en format PDF avec la présentation de l'application et les identifiants afin de réaliser les différents parcours.
- La charte graphique en format PDF avec la palette de couleurs utilisée ainsi que la police et les maquettes mobiles et desktop.
- Les diagrammes MCD et MLD
- Le diagramme de cas d'utilisation

2. Informations complémentaires

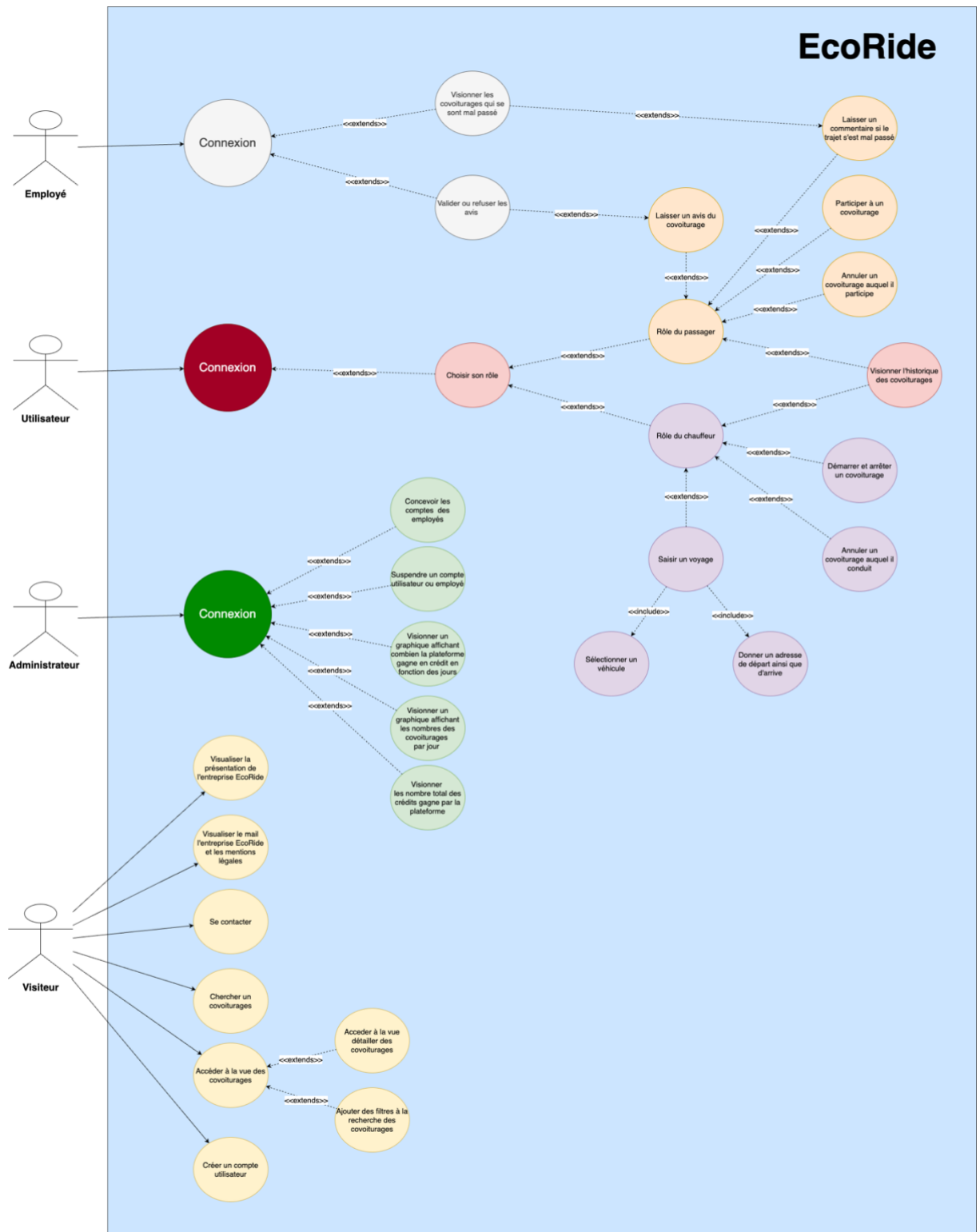
- **Diagramme Conceptuel des données**



- Diagramme Logic des données



- Diagramme de cas d'utilisation



- **Documentation du déploiement de l'application :**

Pour déployer mon application web, j'ai suivi les étapes suivantes :

1. J'ai acheté un **nom de domaine** via **NameCheap**, puis utilisé l'**hébergement web mutualisé** proposé par le même fournisseur. J'ai choisi cet hébergement car il offre :

- un **support natif pour PHP**,
- une **base de données MySQL** intégrée via **phpMyAdmin**,
- un accès FTP via **cPanel**,
- un **certificat SSL gratuit** pour sécuriser les échanges HTTPS.

2. Pour la base MongoDB, j'ai utilisé **MongoDB Atlas**, la plateforme cloud officielle de MongoDB. Elle m'a permis de :

- connecter mon application PHP à MongoDB via l'URL sécurisée fournie,
- bénéficier d'un hébergement cloud gratuit et sécurisé.

3. J'ai utilisé le logiciel **Filezilla** pour transférer l'ensemble de mon projet local vers l'hébergement.

4. J'ai exporté ma base de données MySQL locale au format **.sql**, puis je l'ai importée dans une base MySQL créée depuis **cPanel** en utilisant **phpMyAdmin**.

5. Pour les fonctionnalités d'e-mails, j'ai configuré le **protocole SMTP** en utilisant les identifiants fournis par le service mail de cPanel.