NOM	UNY
Prénom	Marc
Date de naissance	15 février 1994

Copie à rendre

TP – Développeur Web et Web Mobile

Documents à compléter et à rendre

Lien du git : https://github.com/MalinLapin/EcfEcoRide
Lien de l'outil de gestion de projet : https://trello.com/b/BMecWhlk/projet-ecoride-ecf
Lien du déploiement : https://stark-mountain-00422-f4d7d334b310.herokuapp.com/
Lien figma : https://www.figma.com/design/wfOrza0ck2YJ2w1Nncsuuw/ECF---EcoRide?node-id=0-1&t=78nDOPxMEeP3X0ob-1
Login et mot de passe administrateur : jose.ecoride@test.com **Test123456789***
Login et mot de passe employé : marc.uny@test.com **Test123456789***
Login et mot de passe utilisateur : mystere.uny@test.com **Test123456789***

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**

EcoRide est une startup française, ayant pour objectif de réduire l'impact environnemental des déplacements en encourageant le covoiturage. Elle prône une approche écologique et aspire à devenir la principale plateforme de covoiturage pour les voyageurs soucieux de l'environnement et ceux recherchant une solution économique pour leurs déplacements.

Le directeur technique d'EcoRide, souhaite développer une **application web** dont les **couleurs** et le **thème** évoquent l'**écologie**. Cette application doit gérer uniquement les déplacements en **voiture** et offrir une expérience utilisateur intuitive et engageante.

Dès la **page d'accueil**, les **visiteurs** pourront rechercher un itinéraire de covoiturage.

Via le **menu de navigation**, le visiteur pourra accéder :

- Retour vers **la page d'accueil**.
- Un accès à la **page de covoiturage**, cependant un visiteur ne pourra participer à un trajet qu'une fois connecter à son compte.
- Les **employés** et **utilisateurs** pourront se **connecter**, afin d'accéder à toutes les fonctionnalités dues à leur rôle, via une adresse mail et un mot de passe.

L'**administrateur** à accès à l'ensemble des services, il sera, également, en charge de créer les comptes employé.
- **Formulaire de contact** pour contacter un **employé** EcoRide en cas de problème.
- Accès à son **compte personnel**, afin de modifier ses données personnelles. Il pourra aussi avoir accès à son historique de transport. Il aura la possibilité via son compte de lier des véhicules afin de proposer lui-même des trajets.

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

2.1 Expression du besoin

- Créer une page d'accueil qui présente l'entreprise et permet de rechercher des trajets.
- Développer une application web qui gère les déplacements en voitures
- Utiliser une base de données relationnelle et non relationnelle pour stocker les données

2.2 Spécifications fonctionnelles

Page d'accueil

- Présentation de l'entreprise avec description et image
- Barre de recherche permettant de trouver un itinéraire
- Pied de page comprenant email de contact et lien vers les mentions légales

Menu de navigation

- Accès aux principales fonctionnalités : page d'accueil, page de covoiturage, contact et inscription

Une fois le visiteur connecté et de devenue utilisateur de nouvelles possibilités lui seront proposer comme son compte.

Les employés auront accès à leur espace directement, ils ne pourront proposer ou participer à des covoiturages.

Pour finir, seul l'administrateur aura accès à l'ensemble des fonctionnalités, des utilisateurs et employés, en plus de son espace propre de gestion de la plateforme.

Vue des covoiturages

- Un formulaire de recherche de covoiturage
- La liste des covoiturages correspondant à la recherche, cette liste comprendra :
    - Le pseudo, la photo et la note du chauffeur

- Le nombre de place restante
- Le prix
- La date et l'heure du départ ainsi que notre arrivée estimées
- Une mention pour les voyages écologique, c'est-à-dire réaliser en voiture électrique
- Un bouton détail

- Un filtre des covoiturages, afin de :
  - Choisir les trajets écologiques
  - Le prix du voyage maximum
  - La durée du voyage estimé
  - La note du chauffeur.

Vue détaillée d'un covoiturage

- Les avis du conducteur
- Les informations du véhicule
- Visionner les préférences du conducteur

Si le détail convient, il est possible de cliquer sur un bouton « *participer* ». Si nous sommes visiteurs, le clique nous emmènera nous connecter ou nous inscrire. Si nous sommes déjà connectés il y aura vérification de notre solde de crédit auquel cas, soit nous seront dirigé vers une API de paiement, soit vers notre participation qui mettra à jour le nombre de place disponible ainsi que notre solde de crédit.

Création de compte

- Un formulaire d'inscription avec
  - Nom
  - Prénom
  - Email
  - Pseudo
  - Mot de passe sécurisé (longueur, caractères spéciaux, etc..)

Espace Utilisateur / compte

- Ses informations personnelles
- Son historique de trajet avec les trajets inscrit qui ne sont pas encore réalisé

Un utilisateur ou un chauffeur peut annuler une participation, cela se fera via ses trajets enregistrés. En cas d'annulation, un mail sera envoyé à toutes les parties.

Si l'utilisateur souhaite proposer des trajets, il lui sera possible d'ajouter des véhicules via un formulaire comprenant :

- Marque
- Modèles
- Couleur
- Plaque d'immatriculation
- Date de première immatriculation

- Préférences, c'est-à-dire si le véhicule est fumeur ou non-fumeur, si les animaux sont acceptés

Une fois des véhicules enregistrés l'utilisateur pourra proposer des trajets via un nouveau formulaire comprenant

- Ville de départ et d'arriver
- Adresse de départ et d'arriver
- Le prix de la place
- Le nombre de place disponible
- Le véhicule choisi

Le nombre de place disponible se fera par trajet, car une personne peut très bien ne pas être seule au moment de sa proposition de covoiturage.

Lorsque le chauffeur arrivera sur son lieu de rendez-vous, il devra se rendre sur son compte, au niveau des trajets, et le bouton d'annulation aura fait place à un bouton « démarrer le covoiturage » Une fois le trajet lancer ce même bouton permettra de finir le covoiturage.

Une fois le trajet effectué, les passagers recevront un mail afin de noter et de laisser un avis sur le chauffeur. En cas de mauvaise prestation un employé devra contacter le chauffeur avant de permettre la mise à jour des crédits gagné.

Espace employé

- Liste des avis émis par les passagers avant leur rendu publique.
- Les informations relatives au chauffeur et passager concerné par l'avis.

Chaque avis émis par un passage ne sera rendu publique qu'une fois validé par un employé. En cas de mauvaise appréciation, l'employé devra contacter toutes les parties afin de régler le litige, une fois ceci fait l'avis pourra être rendu publique.

Espace Administrateur

- Accès a l'ensemble des services de la plateforme.
- Un espace personnel contenant
    - Graphique affichant le nombre de covoiturage par jour, mois ou année.
    - Graphique affichant les bénéfices en crédit de la plateforme
    - Une fenêtre de création de compte employé
    - Une fenêtre de suspension ou bannissement de compte.
    - Des cards sur le nombre totale de crédit gagné depuis le début de le déploiement de l'application

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

1.1 Front-end : HTML5,CSS3 et Java script

HTML/CSS sont les langages de base du développement web. Ils permettent de créer des pages web structuré et stylisé, ce qui est essentiel pour une application web comme EcoRide. De plus l'utilisation des balises sémantique, proposer par HTML5, permettent un meilleur référencement naturel (SEO) qui est important pour une entreprise qui souhaite s'imposer comme le leader dans un domaine.

JavaScript ajoute du dynamisme et de l'interactivité. Il est essentiel pour une meilleure expérience utilisateur pour le public ciblé.

Ces langages sont supportés par tous les navigateurs modernes, ce qui garantit une compatibilité maximale. De plus, ces technologies sont largement documentées, ce qui facilite l'apprentissage et la résolution de problèmes, au vu de ma formation.

1.2 Back-end : PHP avec PDO

PHP est un langage côté serveur largement utilisé et introduit dans ma formation. Il s'intègre bien avec MySQL, ce qui est idéal pour gérer les données des utilisateurs et des trajets. De plus, la plupart des hébergeurs web supportent PHP, ce qui simplifie le déploiement.

PDO est une extension native de PHP que j'ai sélectionnée pour interagir avec ma base de données. Elle me permet d'utiliser des requêtes préparées, protégeant ainsi mon application contre les injections SQL. De plus, PDO est une interface multi-SGBD, ce qui facilite une migration éventuelle vers un autre système de gestion de bases de données.

1.3 Base de données : MySQL (SQL) et MongoDB (NoSQL).

MySQL est idéal pour stocker des données structurées comme les informations des utilisateurs ou les covoiturages. Cette technologie est fiable et performante, sa facilité de configuration en fait un bon choix pour un débutant en développement web.

En complément, j'ai intégré **MongoDB** pour les données non structurées ou faiblement structurées, comme les avis ou les préférences des chauffeurs, MongoDB en tant

que base NoSQL, permet une grande flexibilité de stockage grâce à son format JSON, ce qui me permet d'ajouter facilement de nouvelles données sans restructurer des tables.

1.4 Gestion des dépendances : Composer

Composer est l'outil que j'utilise pour gérer les dépendances de mon projet. Il me permet d'installer, de mettre à jour et de gérer facilement les bibliothèques tierces comme PHPMailer, Dotenv ou encore Autoload. Composer me garantit une gestion cohérente des versions et des dépendances, ce qui simplifie la collaboration et réduit les conflits.

1.5 Variables d'environnement : Dotenv

J'ai d'abord installé Dotenv via Composer, puis j'ai créé un fichier .env à la racine de mon projet pour y conserver des informations sensibles, comme DB_HOST, DB_USER, DB_PASS et diverses clés API. Dotenv me permet de garder ces informations dans un fichier .env exclu du dépôt de code, renforçant ainsi la sécurité de mon application.

1.6 Envoi de mail : PHPMailer

Pour gérer l'envoi d'emails, j'ai opté pour PHPMailer, qui offre des fonctionnalités avancées comme le support de SMTP et l'envoi d'emails en HTML. Cela améliore la délivrabilité des messages par rapport à la fonction mail native de PHP et limite les risques d'être considéré comme du spam.

Pour l'installation J'ai ouvert le terminal dans le répertoire de mon projet et utilisé Composer pour installer PHPMailer en exécutant la commande :

''composer require phpmailer/phpmailer''

Une fois installé, j'ai intégré PHPMailer dans mon autoloader et j'ai configuré les paramètres SMTP pour personnaliser l'envoi (serveur, email, mot de passe) dans mon .env afin qu'ils soit sécuriser.

J'ai ensuite testé l'envoi d'un email pour vérifier que tout fonctionnait bien, dans la page contact de l'application

En résumé

Ces choix technologiques répondent aux besoins de mon projet en termes de performance, sécurité et évolutivité. Chaque technologie apporte des avantages spécifiques, et leur combinaison m'a permis de construire une application robuste et évolutive, prête pour un usage à long terme.

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

- **GPT Image** : Pour la création du logo de la plateforme
- **GPT-4.1** : Pour la création du slogan, diverse explication et vérification de code.
- **Copilote** : Pour la création de commentaire en PhpDoc de mes différentes méthodes.
- **Figma** : Pour sa simplicité à faire des maquettes
- **Google** : Pour les recherches sur les sites comme MDN ou PHP
- **VS Code** : Comme IDE, car c'est un éditeur très simple qui bénéficie de beaucoup d'extensions afin d'aider à coder plus facilement
- **GitHub** : car il permet un stockage de son projet en ligne
- **Css** : pour styliser, adapter et optimiser mon site, car il permet une flexibilité avec une accessibilité et une facilité de gestion.
- **Java Script** : pour ajouter de l'interactivité, dynamiser les pages web et améliorer l'expérience utilisateur en temps réel.
- **PHP 8.2** : pour générer dynamiquement mes pages web côté serveur, il me permet une gestion efficace des données et des interactions avec la base de données.
- **Html5** : car c'est le langage standard pour structurer et présenté mes contenus, me permet aussi d'avoir des fonctionnalités avancées et une meilleure interactivité et compatibilité avec une large gamme de navigateur
- **Composer :** Pour utiliser l'autoload, les namespaces et PhpUnit.
- **Heroku :** Pour un déploiement simplifié et parce que le student pack nous permet de l'avoir gratuitement.
- **Looping :** Pour le modèle conceptuel de données.
- **Draw.io :** Pour le diagramme de cas d'utilisation.
- **PHPmailer :** Pour l'envoi de mail pour la fin de trajet.
- **Charts.**JS : Bibliothèque pour les graphiques dynamique en Js.
- **MongoCompass :** Permet une visualisation de ma base de donnée noSQL.

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.**

3.1 Validation des entrés utilisateurs pour éviter les injections SQL :

Pour sécuriser mon application contre les injections SQL, j'ai mis en place une validation des entrés utilisateurs via des requêtes préparées. En utilisant des paramètres nommés, chaque valeur fournie par l'utilisateur est passée séparément de la requête SQL, ce qui empêche les données d'altérer la structure de la requête. Grâce à cette méthode, le risque d'exécution de code malveillant dans la base de données est éliminé, garantissant ainsi la sécurité des informations stockées.

3.2 Protection contre les attaques XSS :

Pour empêcher les attaques XSS, j'ai assaini toutes les données reçues en appliquant json_encode aux entrées utilisateurs, ne laissant passer que du texte sans exécution de balise. Cela bloque toute tentative d'injection de code JavaScript malveillant.

3.3 Protection CSRF :

Pour sécuriser mes formulaires avec une requête POST, j'ai utilisé des jetons CSRF. C'est une donnée qui est transmise durant l'affichage du formulaire et qui est envoyée avec celui-ci. Coter serveur nous vérifions d'abord que la donnée reçue est la même que celle qui a été transmise pour s'assurer qu'il n'y a pas eu d'usurpation de session.

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

Dans le cadre d'une veille technologique axée sur les vulnérabilités de sécurité, j'ai exploré les dernières failles critiques recensées et les meilleures pratiques pour renforcer la protection des applications. J'ai consulté des sources comme **cert.ssi.gouv.fr** (**Centre gouvernemental de veille, d'alerte et de réponse aux attaques informatiques**) il fournit des alertes sur les vulnérabilités et les menaces informatiques actuelles, ainsi que des fiches pour améliorer la sécurité numérique

Partie 3 : Recherche

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**

Dans le cadre de mon projet, pour ajouter plus de sécurité, je me suis renseigné sur les jetons CSRF, qui m'a permis de protéger les applications contre les attaques par falsification de requêtes POST.

Source : https://developer.mozilla.org/en-US/docs/Web/Security/Attacks/CSRF

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

CSRF tokens

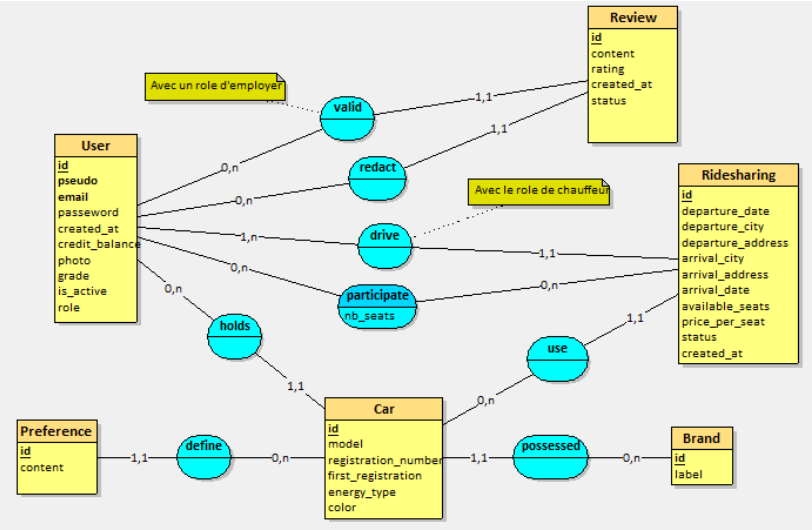
In this defense, when the server serves a page, it embeds an unpredictable value in the page, called the CSRF token. Then when the legitimate page sends the state-changing request to the server, it includes the CSRF token in the HTTP request. The server can then check the token value and carries out the request only if it matches. Because an attacker can't guess the token value, they can't issue a successful forgery. Even if the attacker does discover a token after it has been used, the request can't be replayed if the token changes every time.

Ce que j'ai compris est que, lorsque qu'une page inclue un jeton CSRF, le serveur peut vérifier si la requête est légitime uniquement si le jeton reçu est le même que celui transmis. Il y a aussi le faite que même si une personne malveillante trouve le jeton celui-ci change à chaque utilisation donc le jeton trouver sera de ce fait obsolète.

Partie 4 : Informations complémentaire

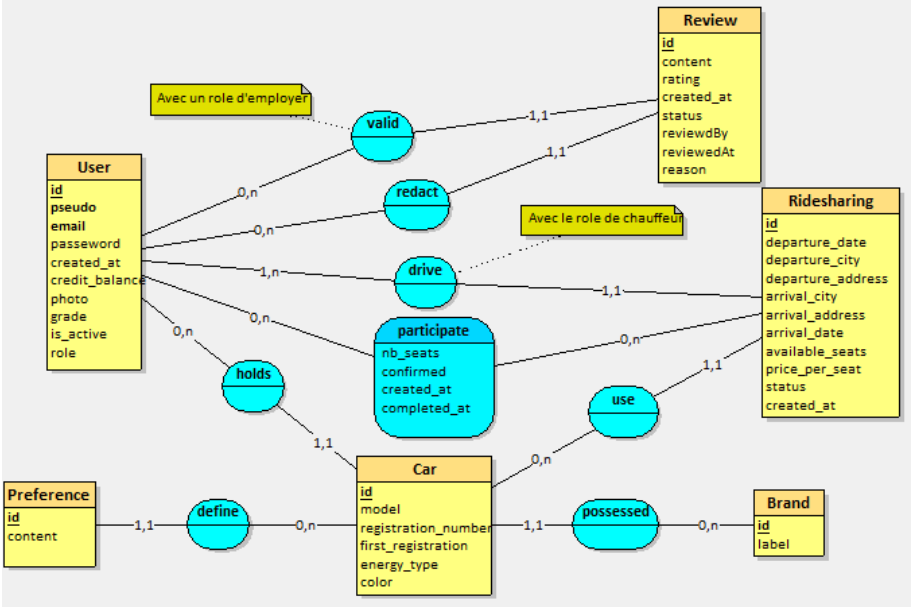
1. Autres ressources

1.1 MCD



Ce diagramme est l'initiale, mais à la suite du développement de l'application, il sera amené à être modifié afin de subvenir à l'ensemble des demandes du client qui n'ont peut-être pas été bien prises en compte à ce stade.

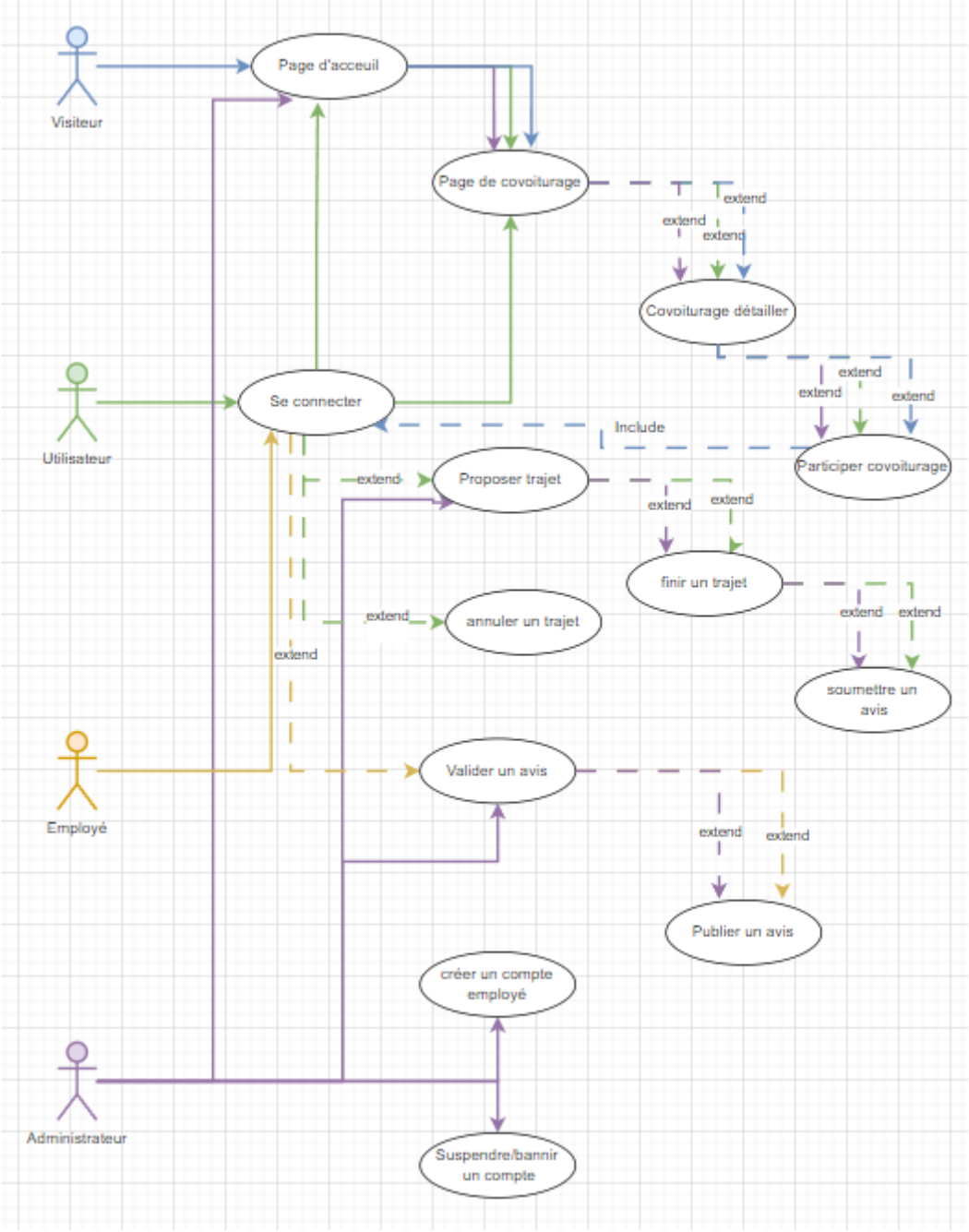
Après réflexion, ce diagramme sera utilisé :



Les modifications apporté sont tout d'abord sur la table d'association, un boolean confirmed pour enregistrer en bdd uniquement si l'utilisateur a fini la double confirmation demander par le client. Il y a aussi le dateTime completed_at qui sera renseigné lorsque la participation sera fini. Ensuite, pour la table Review, j'ai rajouté un int avec reviewedBy afin de savoir quel employé a valider ou refuser un avis. Il y a aussi la date de cette validation/refus et pour finir la raison pour laquelle l'employé a refusé de valider un avis sera enregistrer.

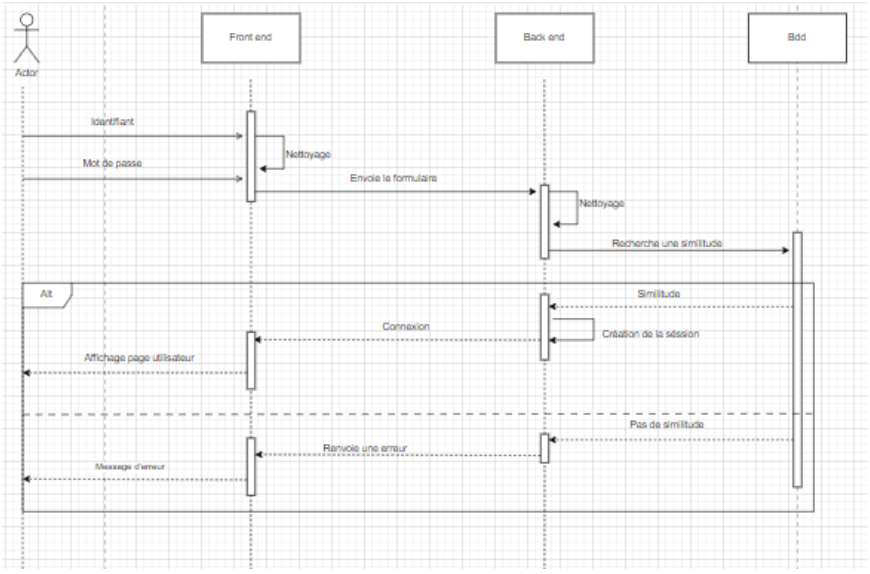
Les tables Review et Preference seront gérer en noSql avec mongoDb. Le schéma documents/collections permet de meilleur performance même avec d'importante masse de données. Un chauffeur pourra soumettre plusieurs préférences pour un trajet ce qui implique une masse de données plusieurs fois plus importante que le nombre de trajet. De même pour les avis, pour un trajet plusieurs passager pour emmètre des avis. D'où mon choix de technologie pour ces tables là.

1.2 Diagramme de cas d'utilisation



1.3 diagramme de séquences.

1.3.1 Login :



1.3.2 Laisser un avis