

Documentation Technique de l'Application de Covoiturage

1. Réflexions initiales technologiques

Le choix technologique s'est orienté vers une architecture moderne et modulaire :

- **Front-end** : Angular, pour sa structure claire et son intégration aisée avec les API REST.
- **Back-end** : ASP.NET Core (C#), assurant la gestion des routes API et la sécurité via JWT.
- **Base de données** : MySQL, un SGBD relationnel performant et compatible avec Entity Framework.
- **Serveur** : IIS ou Apache pour le déploiement, avec hébergement possible sur un VPS ou Azure.

2. Configuration de l'environnement de travail

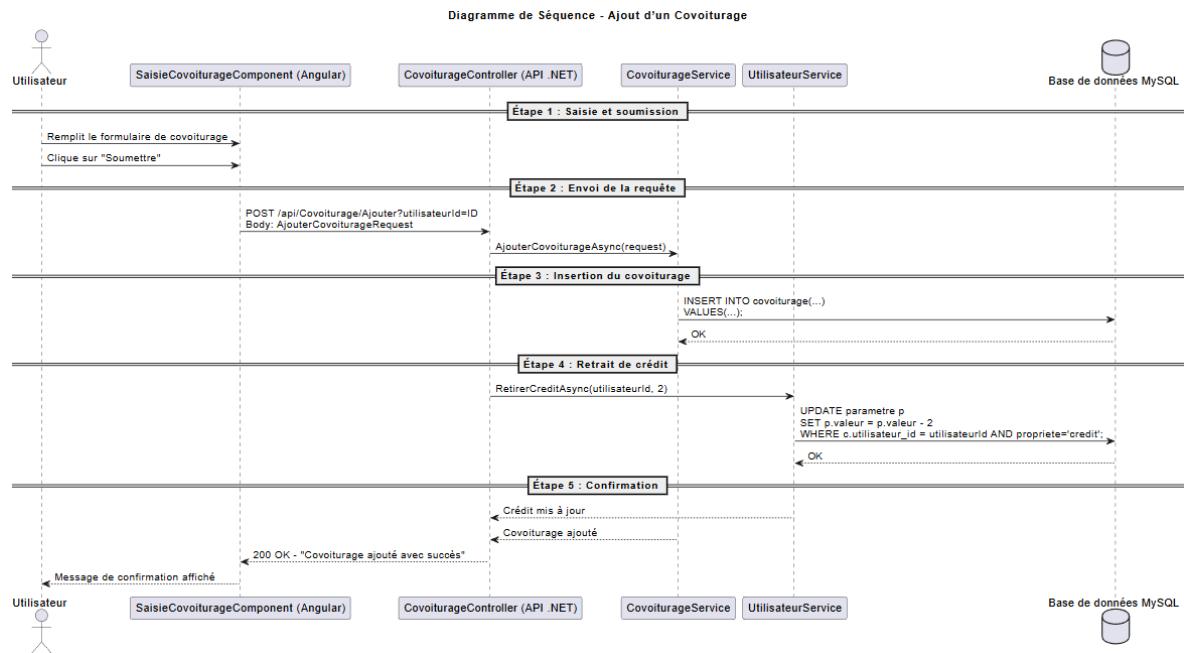
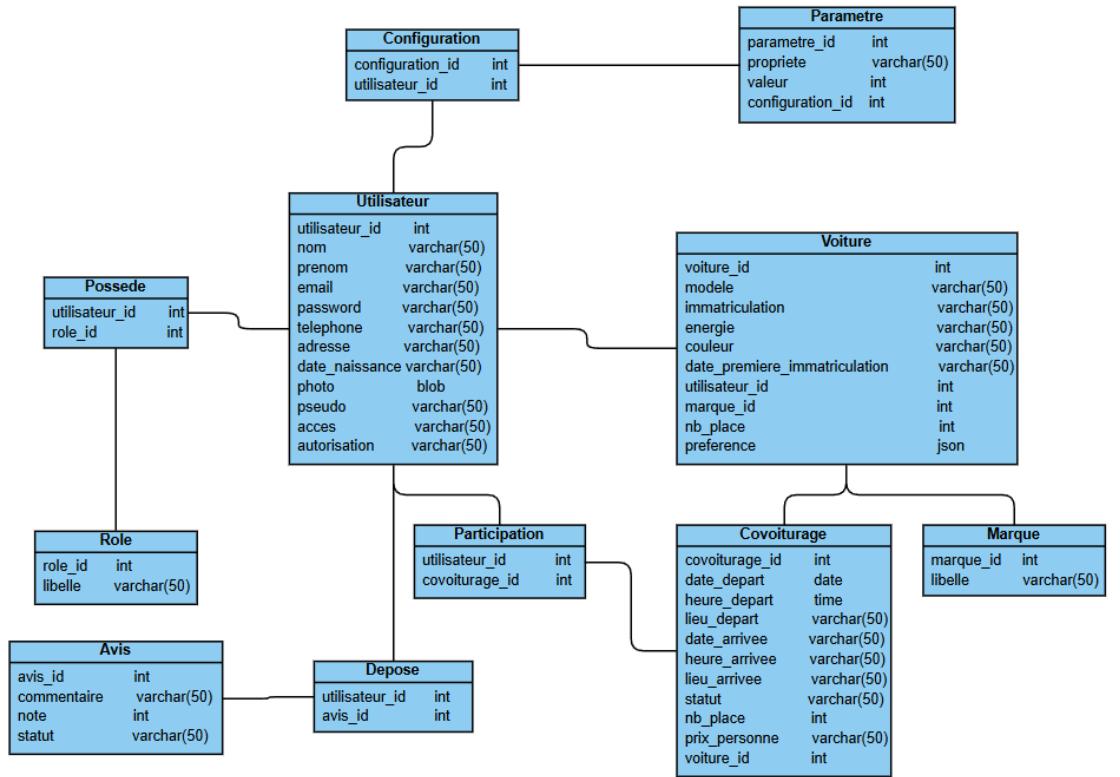
- IDE : Visual Studio 2022 pour le back-end et Visual Studio Code pour Angular.
- Node.js et Angular CLI installés pour le développement du front.
- MySQL Workbench pour la gestion et la visualisation des tables. - GitHub utilisé pour la gestion du versionnement du projet.

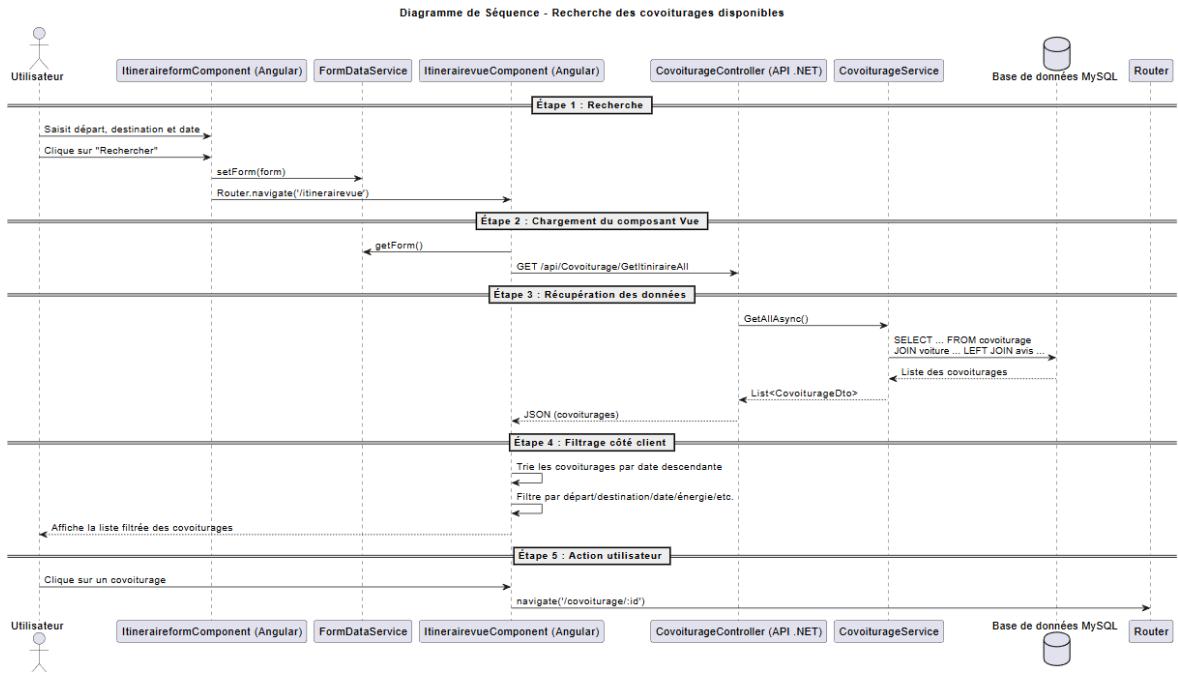
3. Modèle conceptuel de données (MCD)

- **Utilisateur** : Contient les informations personnelles, l'adresse et le droit d'accès de chaque utilisateur.
- **Voiture** : Associée à un utilisateur (chauffeur), stocke les informations du véhicule.
- **Covoiturage** : Relie un chauffeur, une voiture, un lieu de départ et d'arrivée, ainsi qu'un prix.
- **Rôle** : Détermine les positions (chauffeur, passager).

- **Possède** : table d'association entre utilisateur et rôle.
- **Avis** : Contient les évaluations laissées par les utilisateurs sur une expérience de covoiturage. Chaque avis inclut une note, un commentaire, et un statut (par exemple : "validé", "en attente", "refusé").
Un avis est généralement déposé par un utilisateur (souvent un passager) à propos d'un chauffeur ou d'un trajet.
- **Configuration** : Table d'association entre utilisateur et paramètre.
- **Paramètre** : Liste les différentes propriétés de configuration pour un utilisateur.
- **Marque** : Répertorie les marques de véhicules disponibles dans l'application.
Elle est utilisée comme référence dans la table Voiture afin d'assurer la cohérence des données et d'éviter les doublons.
- **Participation** : Représente la participation d'un utilisateur à un covoiturage.
Chaque ligne relie un utilisateur (souvent passager) à un trajet donné.
Cette table permet de gérer les réservations et d'identifier les passagers associés à un covoiturage.
- **Dépose** : Associe un utilisateur à un avis qu'il a déposé.
Cette table permet de savoir quel utilisateur a laissé quel avis, et garantit qu'un même utilisateur ne peut déposer qu'un avis unique pour un trajet ou une situation donnée.
- **Temp_avis** : Stocke temporairement les avis avant leur validation définitive.
Les avis présents ici sont en attente de modération ou d'approbation avant d'être transférés vers la table Avis.
Elle conserve les informations de l'auteur, du covoiturage concerné, et le contenu de l'avis.

4. Diagrammes UML





5. Documentation du déploiement

Le déploiement s'effectue en plusieurs étapes :

1. Compilation du projet Angular (`ng build --prod`).
2. Publication du back-end ASP.NET Core via Visual Studio ou en ligne de commande.
3. Copie des fichiers Angular dans le dossier `wwwroot` du projet .NET.
4. Configuration de la base MySQL sur le serveur distant.
5. Mise à jour de la chaîne de connexion dans `appsettings.json`.
6. Démarrage du service IIS ou configuration du reverse proxy.