

# 1. Réflexions Initiales Technologiques

Avant de démarrer le développement, j'ai pris en compte les technologies que je maîtrisais le mieux tout en m'assurant qu'elles répondent aux besoins du projet.

- **Langage de programmation** : J'ai choisi **PHP** car il est bien adapté aux applications web dynamiques.
- **Technologies front-end** :
  - J'ai choisi **HTML5** pour structurer les pages de l'application grâce à ses balises sémantiques modernes.
  - **CSS3** a été utilisé pour le style et la mise en page.
  - **Bootstrap** a été adopté pour ses composants préconçus et son système de grille responsive.
- **Framework ou CMS** : J'ai décidé de ne pas utiliser de **CMS** comme WordPress afin de garder une flexibilité totale sur le développement. À la place, j'ai codé l'interface manuellement en **HTML, CSS, Bootstrap et PHP**.
- **Environnement de développement** :
  - J'ai utilisé **Visual Studio Code (VS Code)** comme éditeur de code en raison de mon apprentissage sur celui-ci, de sa prise en charge avancée de **PHP, HTML et CSS**, et de ses nombreuses extensions comme **PHP Intelephense et PHP Server**.
- **Base de données** : **MySQL**, géré via **PHPMyAdmin et XAMPP**, a été choisi pour sa facilité d'administration.
- **Outil de maquettage** : J'ai choisi **Figma** car c'est sur ce dernier que nous avons travaillé.
- **Déploiement** : Après avoir envisagé plusieurs solutions (**fly.io, Heroku**), j'ai choisi **Heroku** pour sa simplicité et sa compatibilité avec mon stack technologique.
  - Pour gérer les dépendances et assurer la prise en charge de l'extension GD sur Heroku, j'ai utilisé Composer afin d'installer GD, ce qui a permis de manipuler les images efficacement dans mon application

Ces choix ont été faits en prenant en compte **la facilité de mise en œuvre, la documentation disponible et la rapidité d'exécution**.

## 2. Mon Environnement de Travail

Pour réaliser ce projet, j'ai utilisé plusieurs outils technologiques facilitant la conception, le développement et le déploiement de l'application.

### Maquettisation

J'ai choisi **Figma** pour la conception des maquettes en raison de sa simplicité d'utilisation et de ses fonctionnalités avancées en matière de conception UX/UI. Son système de gestion des flux de navigation (flows) permet une visualisation intuitive des interactions entre les différents écrans de l'application.

[Accessible en cliquant ici](#)

### Gestion de Projet

Pour organiser le travail et assurer un suivi efficace des tâches, j'ai utilisé **Lucidspark**. Cet outil collaboratif se distingue par sa facilité d'utilisation et ses fonctionnalités de brainstorming visuel, ce qui a permis de structurer les idées et d'optimiser la gestion des tâches.

[Accessible en cliquant ici](#)

### Développement

Le développement de l'application a été réalisé sous **Visual Studio Code** (VS Code), agrémenté de plusieurs extensions utiles :

- **PHP Intelephense** : Un puissant moteur d'analyse statique pour PHP permettant une autocomplétion et une vérification avancée du code.
- **PHP Intellisense** : Une extension offrant des suggestions de code intelligentes et une navigation améliorée.
- **PHP Server** : Permet d'exécuter un serveur PHP local rapidement sans configuration complexe.

### Base de Données

Pour la gestion de la base de données, j'ai utilisé **XAMPP**, une solution regroupant Apache, MySQL, et PHPMyAdmin, facilitant ainsi l'hébergement local et la gestion des bases de données via une interface graphique.

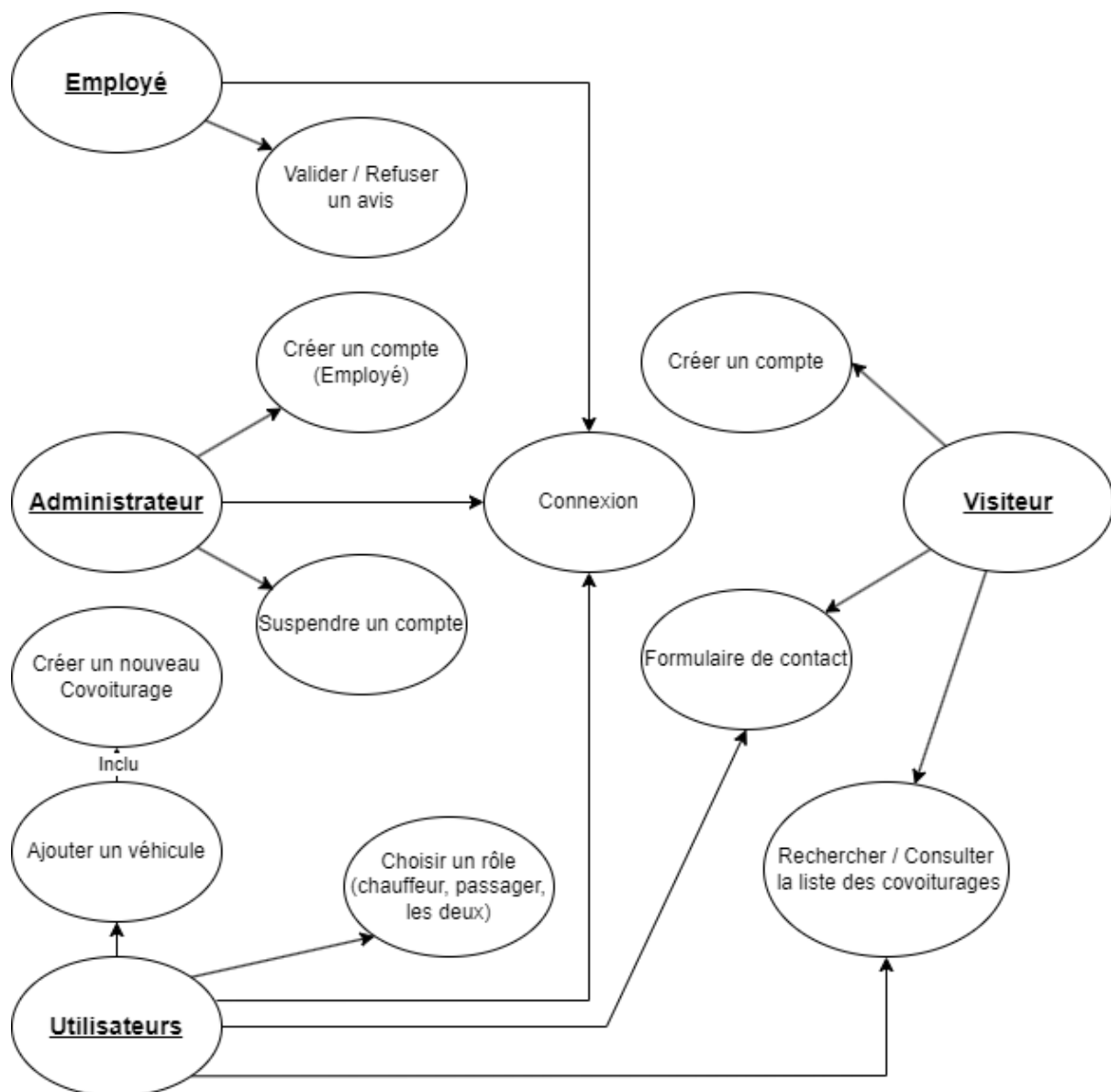
## Déploiement

L'application a été déployée sur **Heroku**, une plateforme cloud permettant l'hébergement d'applications web.

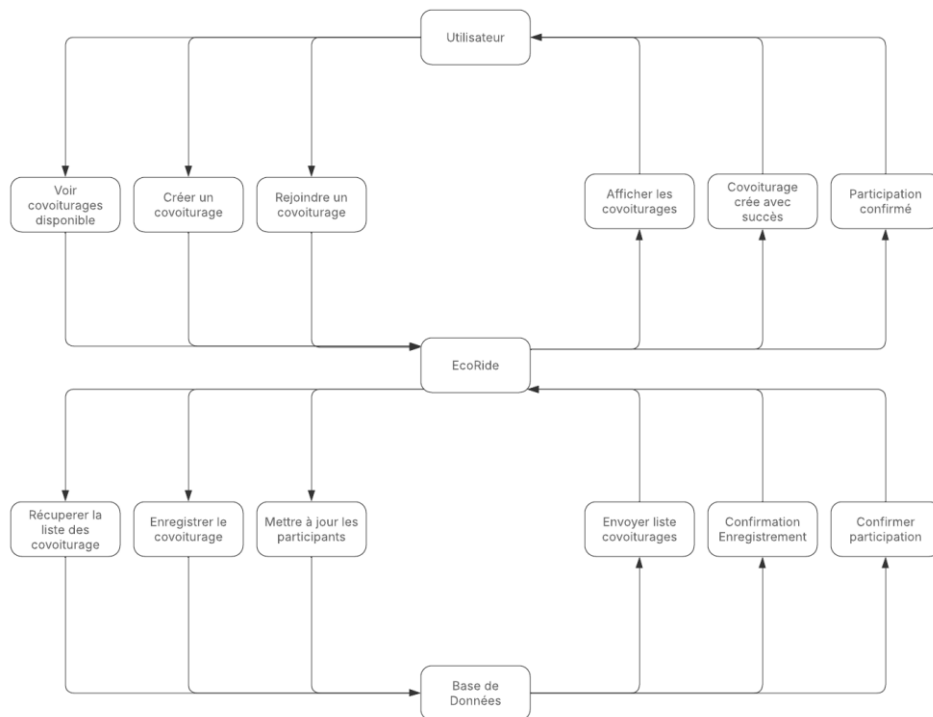
## 3. Diagramme de Classe

Le diagramme de classe utilisé est celui proposé dans le sujet du projet.

## 4. Diagramme d'Utilisation et Diagramme de Séquence



**Diagramme de séquence  
covoiturage**



## 5. Déploiement de l'Application : Démarche et Étapes

Le déploiement de l'application sur Heroku a nécessité plusieurs étapes :

### 1. Installation d'Heroku

- a. J'ai installé Heroku CLI sur mon PC afin de pouvoir interagir avec la plateforme depuis mon terminal.

### 2. Configuration de PowerShell

- a. J'ai configuré **PowerShell** pour qu'il puisse exécuter Heroku directement via Visual Studio Code, facilitant ainsi les commandes de déploiement.

### 3. Utilisation de Git pour le Déploiement

- a. Après avoir préparé mon projet, j'ai utilisé Git pour envoyer mon code sur Heroku.
- b. Depuis la branche principale (**Main**) de Visual Studio Code, j'ai exécuté la commande suivante pour déployer l'application :

```
git push heroku main
```

- c. Cette commande permet d'envoyer le code source directement sur Heroku et de déployer l'application en ligne.

