Documentation technique d’EcoRide

1. Réflexions initiales technologiques

- **Symfony 7.2 (PHP 8.4)**

  Choisi pour sa stabilité, son MVC clair et son écosystème riche (sécurité CSRF, authentification, composants réutilisables). La migration vers la version 7.2.6 a corrigé un bug critique de formulaire en production.

- **Doctrine ORM (MySQL)**

  Permet un mapping objet-relationnel fluide, gestion automatique des migrations et requêtes SQL optimisées.

- **MongoDB**

  Stockage des préférences utilisateur de manière schéma-less, pour plus de flexibilité et de montée en charge.

- **Docker + Apache**

  Garantit une parité totale entre les environnements de développement et de production, facilite l’isolation des services et accélère les déploiements.

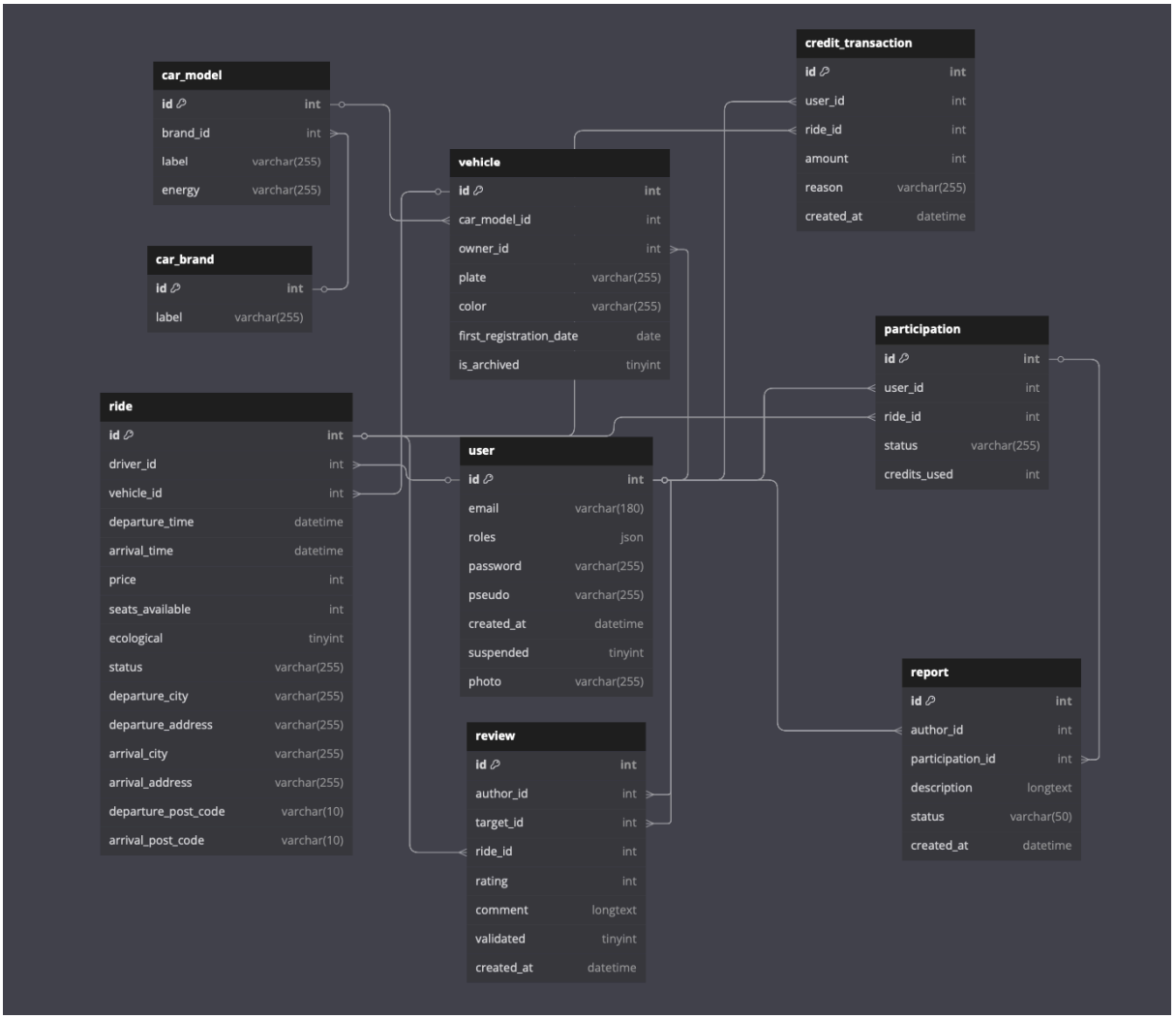
- **GitHub**

  Pour la gestion du code source et du versioning, j’ai choisi GitHub : j’y héberge le dépôt et j’applique un workflow GitFlow structuré (branche dev pour le développement quotidien, branches feature/... pour chaque nouvelle fonctionnalité, main pour les versions stables). Chaque évolution passe par une Pull Request, assortie d’une revue de code et d’une validation automatisée, ce qui garantit une traçabilité fine des modifications et une collaboration fluide.

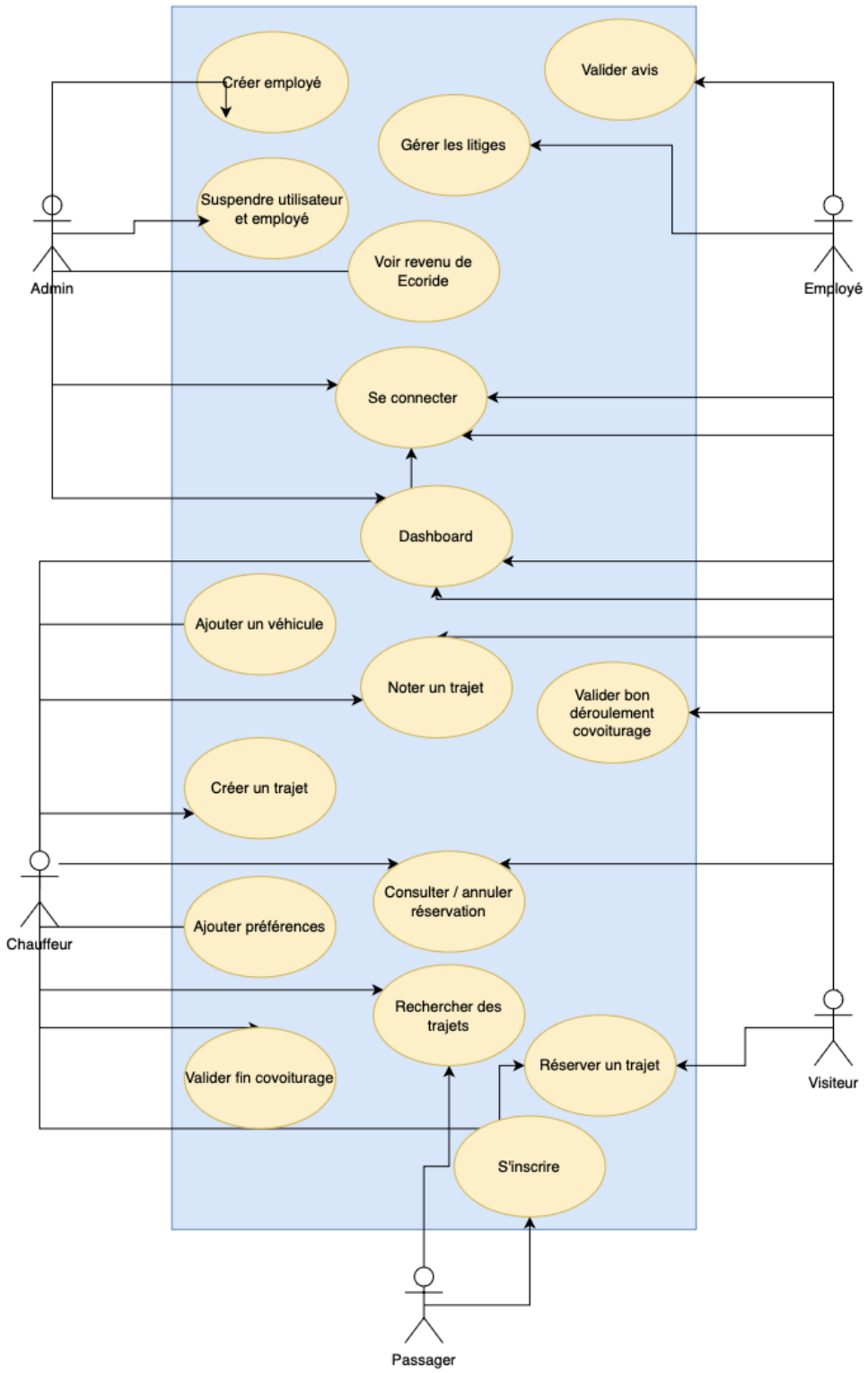
2. Configuration de l’environnement de travail

- **Matériel**
  - MacBook Air M1 (8 Go RAM, SSD 256 Go) sous macOS Sequoia
- **IDE & extensions**
  - VS Code + PHP Intelephense, Symfony Snippets, Docker, GitLens
- **Versionning**
  - Git (GitFlow)
- **Conteneurs Docker**
  - Apache/PHP 8.4, MySQL 8.0, MongoDB, phpMyAdmin
  - docker compose up -d / docker compose exec apache bash
- **Outils CLI**
  - Composer (install, update)
  - Heroku CLI (heroku login, git push heroku main)
- **Tests & debug**
  - Symfony Profiler (toolbar Chrome)
  - Postman pour API

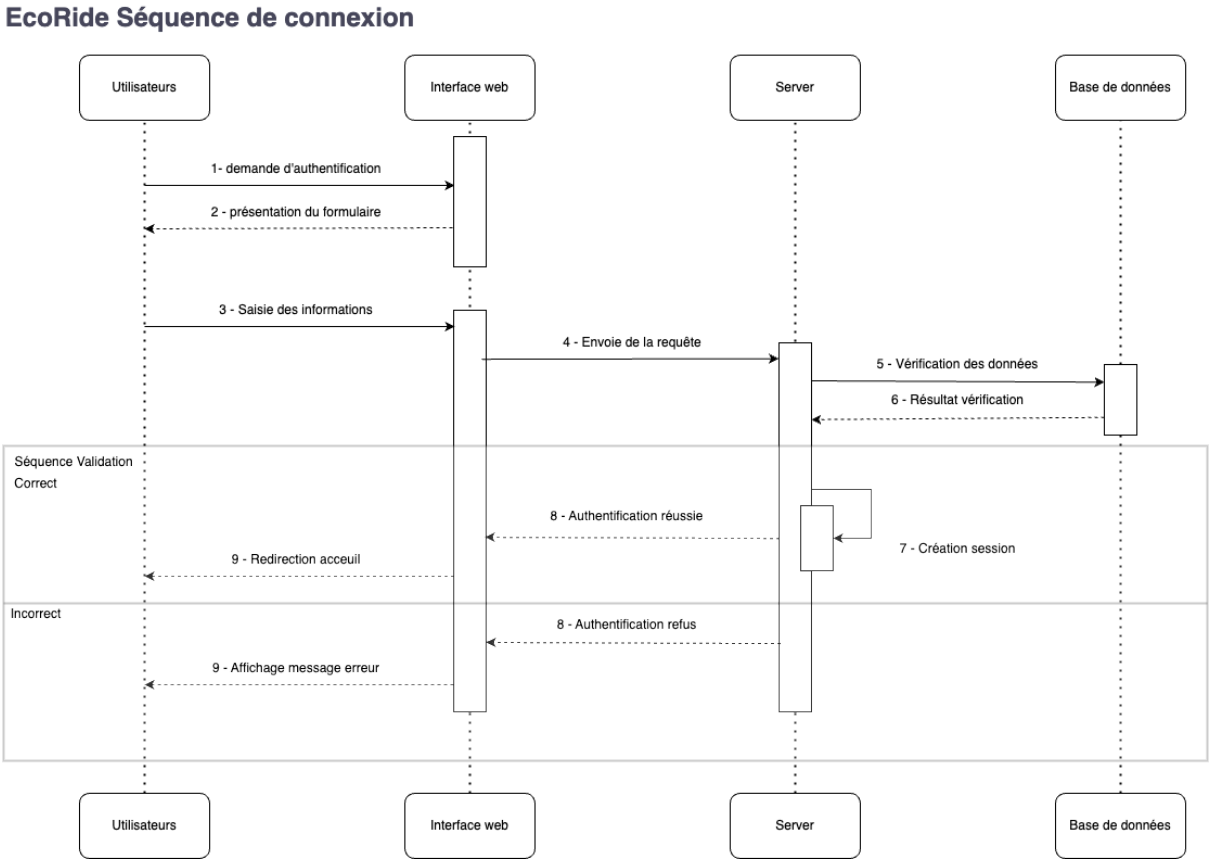
2. Diagramme de classe



3. Diagramme de cas d’utilisation


Créer employé
Valider avis
Gérer les litiges
Suspendre utilisateur et employé
Admin
Voir revenu de Ecoride
Employé
Se connecter
Dashboard
Ajouter un véhicule
Noter un trajet
Valider bon déroulement covoiturage
Créer un trajet
Chauffeur
Consulter / annuler réservation
Ajouter préférences
Rechercher des trajets
Valider fin covoiturage
Réserver un trajet
Visiteur
S'inscrire
Passager

4. Diagramme de classe



5. Documentation du déploiement

Cette section détaille le processus de déploiement de l'application **EcoRide**, depuis la configuration locale jusqu'à la mise en production sur Heroku.

---

1. Préparation de l'environnement

Fichier d'environnement

Créer (ou mettre à jour) .env.local à la racine du projet avec les variables suivantes :

```
APP_ENV=prod
APP_SECRET=<votre_clé_secrète>
```

```
DATABASE_URL=<URL_MYSQL_HEROKU>
MAILER_DSN=<DSN_SMTP>
MONGODB_URL=<URL_MONGODB>
ORS_API_KEY=<votre_clé_ORS>
```

Configuration Heroku

- Provisionner l'add-on MySQL (JawsDB ou ClearDB) :

```
heroku addons:create jawsdb:kitefin --app ecoride-study-project
```

- Définir les variables de configuration :

```
heroku config:set APP_SECRET=... DATABASE_URL=... MAILER_DSN=... MONG
  --app ecoride-study-project
```

- Récupérer et stocker les clés d'envoi d'emails (Mailgun ou SMTP Gmail).

---

2. Construction et tests locaux

Lancement des conteneurs Docker

```
docker compose up --build -d
```

1. **Vérification fonctionnelle**
   - Application : http://localhost:8080
   - Interface MySQL (phpMyAdmin) : http://localhost:8081
2. **Tests automatisés**

```
docker compose exec apache bash
php bin/phpunit
```

Corriger toute erreur avant de poursuivre.

---

3. Déploiement manuel sur Heroku

1. **Connexion et lien Git**

```
heroku login
heroku git:remote -a ecoride-study-project
```

1. **Préparation du code**
    - S'assurer d'être sur la branche main avec tous les commits fusionnés depuis dev.
    - Pousser le sous-répertoire app :

```
git subtree push --prefix app heroku main
```

1. **Installation & migrations**Heroku exécutera automatiquement :

```
composer install --no-dev --optimize-autoloader
```

1. Puis, lancer les migrations :

```
heroku run php bin/console doctrine:migrations:migrate --no-interaction
```

1. **Vérification post-déploiement**
    - Ouvrir l'application :

```
https://ecoride-study-project.herokuapp.com
```

- Consulter les logs en temps réel :

```
heroku logs --tail
```

---

4. Maintenance et suivi

- **Rollback**En cas de régression, restaurer une version précédente depuis le dashboard Heroku.
- **Sauvegardes régulières**Mettre en place des dumps programmés de la base MySQL et de MongoDB pour garantir la résilience des données.