Écoride

Documentation technique

# Sommaire – Documentation Technique

I. Introduction

1.1 Présentation du projet  
1.2 Contexte et objectifs (EcoRide)  
1.3 Périmètre fonctionnel et contraintes

II. Analyse des besoins

2.1 Étude des utilisateurs et rôles (Visiteur, Utilisateur, Employé, Administrateur)  
2.2 Analyse des User Stories (US1 à US13)  
2.3 Spécifications fonctionnelles principales  
2.4 Contraintes techniques et de sécurité

III. Architecture de l’application

3.1 Stack technique choisie (Front-end, Back-end, Base de données relationnelle & NoSQL, Déploiement)  
3.2 Architecture logicielle (MVC, API, etc.)  
3.3 Schéma global de l’application

IV. Conception

4.1 Charte graphique (palette de couleurs, typographies, thèmes)  
4.2 Maquettes (wireframes & mockups – desktop & mobile)  
4.3 Modèle conceptuel de données (MCD)  
4.4 Diagrammes UML

* Diagramme de cas d’utilisation
* Diagramme de séquence
* Diagramme de classes (si applicable)

## V. Développement Front-end

5.1 Installation et configuration de l’environnement  
5.2 Composants de l’interface utilisateur  
5.3 Gestion de la navigation et des vues  
5.4 Dynamisation des interfaces (JS/Framework)

## VI. Développement Back-end

6.1 Mise en place de la base de données relationnelle  
6.2 Gestion des données NoSQL  
6.3 Développement des API et logique métier  
6.4 Sécurité (authentification, autorisations, stockage des mots de passe, etc.)

## VII. Déploiement

7.1 Procédure d’installation en local (README.md)  
7.2 Déploiement sur l’hébergement choisi (fly.io, Heroku, Vercel, etc.)  
7.3 Gestion des environnements (dev, test, prod)

## VIII. Gestion de projet

8.1 Méthodologie (Agile/Kanban)  
8.2 Outils utilisés (Trello, Jira, Notion, etc.)  
8.3 Suivi des versions (Git workflow : branches, merges, tests)

## IX. Manuel d’utilisation

9.1 Présentation de l’application  
9.2 Parcours utilisateur (Visiteur → Utilisateur → Chauffeur/Passager, Employé, Admin)  
9.3 Identifiants de démonstration  
9.4 Fonctionnalités principales (recherche, réservation, gestion des trajets, etc.)

## X. Conclusion

10.1 Récapitulatif des choix techniques  
10.2 Améliorations possibles et évolutions futures

I. Introduction

1.1 Présentation du projet

…  
1.2 Contexte et objectifs (Ecoride)

…  
1.3 Périmètre fonctionnel et contraintes

…

II. Analyse des besoins

2.1 Étude des utilisateurs et rôles (Visiteur, Utilisateur, Employé, Administrateur)

…  
2.2 Analyse des User Stories (US1 à US13)

…  
2.3 Spécifications fonctionnelles principales

Mise en place d’une application de covoit accès sur les énergies vertes

Objectif : Rouler à plusieurs en réduisant l’impact carbonne de chaque individu  
2.4 Contraintes techniques et de sécurité

Contrainte techniques : Mise en place d’un système dynamique et simple d’utilisation, intuitif (UI UX).

Contrainte sécurité : Sécurisation des données User à bien faire ! IMPORTANT ++

III. Architecture de l’application

3.1 Stack technique choisie (Front-end, Back-end, Base de données relationnelle & NoSQL, Déploiement)

HTML CSS JS pour le front, alimenté via twig, le moteur de template de symfony.

Back : php, alimenté par Symfony et toutes ses dépendances  
BDDR MySQL 8.4

BBD NoSQL : MongoDB

Déploiement Héroku  
3.2 Architecture logicielle (MVC, API, etc.)

Utilisateur du framework PHP symfonyn l’architecture logicielle est donc en Model View Controller  
3.3 Schéma global de l’application

…

IV. Conception

4.1 Charte graphique (palette de couleurs, typographies, thèmes)

…  
4.2 Maquettes (wireframes & mockups – desktop & mobile)

..  
4.3 Modèle conceptuel de données (MCD)

…  
4.4 Diagrammes UML

* Diagramme de cas d’utilisation
* Diagramme de séquence
* Diagramme de classes (si applicable)

## V. Développement Front-end

5.1 Installation et configuration de l’environnement

Mise en place de Bootstrap pour le framework CSS (compatibilité forte avec Symfony), utilisateur d’un moteur de template présent par défaut dans symfony : twig  
5.2 Composants de l’interface utilisateur

…  
5.3 Gestion de la navigation et des vues

La gestion des vues et gérer par symfony en faisant appel au controller.  
5.4 Dynamisation des interfaces (JS/Framework)

…

## VI. Développement Back-end

6.1 Mise en place de la base de données relationnelle

Mise en place de la BDD MySQL avec docker,   
docker pull mysql :8.4, lancement d’un container avec les variables d’environnement suivante :

MYSQL\_USER: defaultUser

MYSQL\_PASSWORD: MySq%40root170520

MYSQL\_DATABASE: ecoride\_project

MYSQL\_ROOT\_PASSWORD: RoOt%40root170520

Mise en place des volumes docker pour garder en mémoire les changements dans la BDD

6.2 Gestion des données NoSQL

…  
6.3 Développement des API et logique métier

Mise en place des différentes entités  
6.4 Sécurité (authentification, autorisations, stockage des mots de passe, etc.)

Mise en place via Symfony du système d’authentification par défaut séxurisé (sauf pour le CRSF TOKEN ATTACK), make:User, Make:Login Ajustement des champs User avec le diagramme UML.

Mise en place d’un utilisateur MySQL par défaut et non d’un root pour éviter d’attribuer les pleins pouvoirs à la connexion PDO

## VII. Déploiement

7.1 Procédure d’installation en local (README.md)

Procédure à copier via le README.md  
7.2 Déploiement sur l’hébergement choisi (fly.io, Heroku, Vercel, etc.)

Déploiement Heroku  
7.3 Gestion des environnements (dev, test, prod)

…

## VIII. Gestion de projet

8.1 Méthodologie (Agile/Kanban)

Méthodologie kanban avec mise en place agile.  
8.2 Outils utilisés (Trello, Jira, Notion, etc.)

Outil utilisé : Trello,

lien :   
8.3 Suivi des versions (Git workflow : branches, merges, tests)

Mise en place d’un git

Lien :

Branches : main, develop, home\_feature **…**

## IX. Manuel d’utilisation

9.1 Présentation de l’application

…  
9.2 Parcours utilisateur (Visiteur → Utilisateur → Chauffeur/Passager, Employé, Admin)

…  
9.3 Identifiants de démonstration

…  
9.4 Fonctionnalités principales (recherche, réservation, gestion des trajets, etc.)

…

## X. Conclusion

10.1 Récapitulatif des choix techniques

…  
10.2 Améliorations possibles et évolutions futures

…