

Documentation - Gestion de Projet : EcoRide-2025

1. Méthodologie de travail

Pour mener à bien le projet EcoRide-2025, j'ai adopté une approche Kanban, centrée sur l'organisation visuelle des tâches et l'adaptabilité. Chaque tâche fonctionnelle a été formulée sous forme de user story, ce qui m'a permis de garder une vision utilisateur claire tout au long du projet.

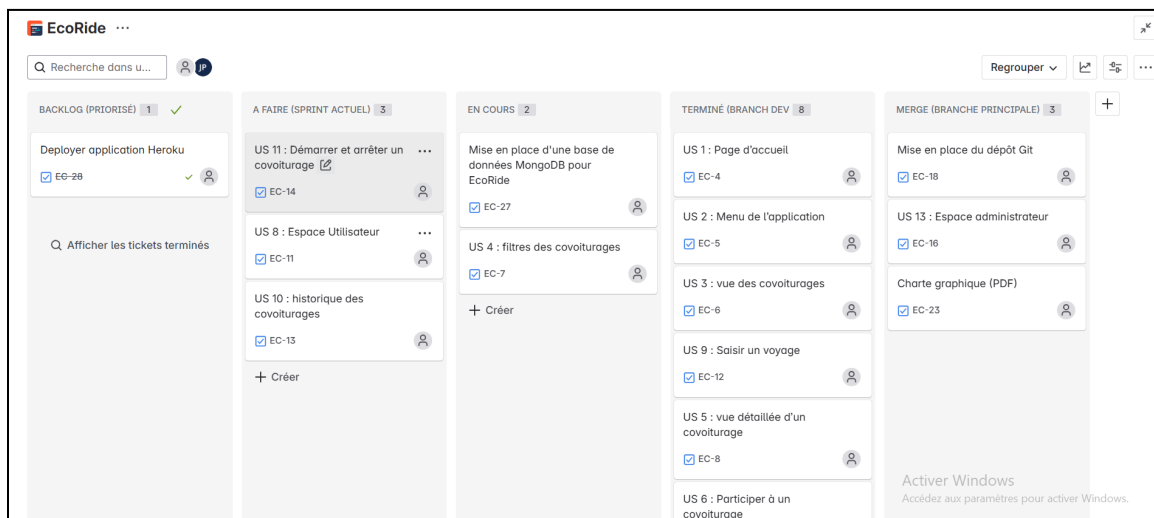
Ces user stories étaient organisées dans un tableau Kanban en cinq colonnes distinctes, représentant les différentes phases du cycle de développement :

1. Fonctionnalités prévues : une colonne recensant toutes les fonctionnalités identifiées dès le départ, ordonnées par priorité, pour garder une vision globale du périmètre du projet.
2. À développer (Backlog actif) : les fonctionnalités planifiées à court terme, celles que je comptais traiter pendant une session de travail ou un sprint.
3. En cours : les tâches actuellement en développement.
4. Terminées : les fonctionnalités codées et testées localement, prêtes à être mergées dans la branche de développement.
5. Mergées sur la branche principale : les fonctionnalités finalisées et considérées comme stables, après validation complète.

Cette méthode m'a permis une grande flexibilité dans les ajustements, une clarté dans les priorités, et une traçabilité précise de l'évolution du projet.

Kanban:

<https://jonathanpina.atlassian.net/jira/software/projects/EC/boards/34?atlOrigin=eyJpIjoiMDBhODM5MWM5NGQ5NGVjMWE3MmFiYmRjMzhiYWYxYmEiLCJwIjoiaU9>



2. Planification

Le travail a été réparti sur plusieurs jours, avec un investissement d'environ 1 à 2 heures par jour. Cette régularité a permis une évolution continue du projet sans surcharge de travail.

3. Évolutions et ajustements

Le développement du projet EcoRide-2025 a suivi une approche progressive et structurée. J'ai commencé par concevoir l'ensemble des maquettes (mockups) et des wireframes sur figma pour toute la partie front-end, afin de définir visuellement les différentes interfaces et leurs comportements. Cela m'a permis d'avoir une vision claire de l'architecture de navigation du site dès le départ.

Une fois cette étape validée, je me suis concentré sur la partie back-end en réalisant les diagrammes de classes et d'utilisation, ce qui a structuré ma réflexion autour de la base de données, des entités et des interactions utilisateurs. Ces schémas ont été fondamentaux pour concevoir une architecture technique cohérente.

J'ai ensuite utilisé Symfony, un framework PHP robuste, pour développer le site en intégrant progressivement les fonctionnalités. Tout au long du développement, j'ai apporté des ajustements au front-end, améliorant certains éléments visuels et ergonomiques à mesure que le back-end se consolidait. Ces évolutions ont été motivées par des idées nouvelles, des tests et des besoins techniques rencontrés en chemin.

3.1. Intégration de MongoDB Atlas (Audit et Logs)

Utilisation de MongoDB Atlas :

J'ai intégré MongoDB comme base NoSQL complémentaire via l'extension PHP MongoDB. Elle est principalement utilisée pour stocker les logs de connexions utilisateurs (audit trail). Cela permet une meilleure séparation des données applicatives et des données de suivi, tout en facilitant l'analyse.

L'objectif de cette intégration était d'assurer un suivi précis des connexions utilisateurs, séparé de la base relationnelle (PostgreSQL), et faciliter une future analyse de sécurité ou d'usage.

4. Gestion de la charge de travail

Travaillant seul sur ce projet, j'ai géré les priorités en m'appuyant sur mon expérience acquise dans des projets précédents. J'ai également consulté régulièrement la documentation officielle des technologies utilisées, ce qui m'a permis de progresser efficacement et de résoudre rapidement les difficultés rencontrées.

5. Bonnes pratiques Git

Pour organiser le code de manière professionnelle et structurée, j'ai appliqué les bonnes pratiques de gestion de version avec Git. Cela m'a permis de tester chaque fonctionnalité indépendamment et d'assurer une intégration progressive et sécurisée.

Organisation des branches :

- Branche principale (main) : contient uniquement du code stabilisé, testé et prêt pour la production.
- Branche de développement (dev) : accueille les fonctionnalités validées, en attente de merge final.
- Branches fonctionnelles : chaque fonctionnalité ou correctif est développée dans une branche dédiée issue de dev (ex : feature/connexion-utilisateur, feature/formulaire-trajet).

Le code a ensuite été déployé sur Heroku, une plateforme cloud facilitant la mise en ligne d'applications Symfony. Ce choix m'a permis d'avoir une version en ligne du site accessible à tout moment, pour les tests ou démonstrations.

6. Test

Pour tester l'application dans les différents rôles utilisateurs, voici les identifiants à utiliser :

L'application EcoRide a été testée selon les différents rôles définis (Administrateur, Employé, Client) pour garantir le bon fonctionnement de toutes les fonctionnalités.

6.1 Rôles disponibles pour test :

Rôle	Email	Mot de passe
Administrateur	admin@ecoride.com	EcoRide_Admin@2025!
Employé	employe@ecoride.com	EcoRide_Employe@2025!
Utilisateur	user@ecoride.com	EcoRide_User@2025!

6.2 Parcours de test recommandés :

Administrateur

- Se connecter via l'espace Admin
- Créer un compte Employé
- Accéder aux statistiques globales
- Suspendre un compte Utilisateur

Employé

- Se connecter via l'espace Employé
- Valider ou refuser les avis déposés par les passagers
- Consulter les trajets signalés ou conflictuels
- Gérer les interventions auprès des conducteurs

Utilisateur (Client)

- Se connecter avec un compte existant
- Rechercher un trajet (par ville et date)
- Participer à un covoiturage si des crédits sont disponibles
- Voir l'historique des trajets
- Noter un chauffeur à la fin d'un trajet
- Modifier ses préférences (fumeur, animaux, musique...)
- Devenir chauffeur

Autres tests fonctionnels

- Vérification du système de crédits (débit et recharge)
- Affichage conditionnel des itinéraires écologiques
- Affichage des filtres de recherche (note, prix, durée, éco)
- Vérification du comportement sur mobile et desktop

. Conclusion

Cette gestion de projet m'a permis de développer une application complète en respectant les bonnes pratiques de développement web. L'organisation en Kanban, l'utilisation de Git structuré, et l'autonomie dans la recherche d'informations ont été les clés pour structurer mon travail et livrer un projet fonctionnel, cohérent et professionnel.