

Gestion de projet Ecoride

Méthode de travail

Le projet a été mené de manière autonome, depuis l'analyse du besoin jusqu'à la réalisation technique. L'objectif était de structurer efficacement le travail tout en conservant de la souplesse pour s'adapter aux imprévus.

Les étapes de travail ont été les suivantes :

- **Lecture du cahier des charges** pour identifier les fonctionnalités essentielles
- **Découpage fonctionnel en cartes Trello**, priorisées selon leur importance
- **Prototypage rapide** lorsque nécessaire (maquettes d'écrans, formulaire, vues mobiles)
- **Développement progressif**, avec tests à chaque étape
- **Itérations** en fonction des nouvelles idées ou contraintes rencontrées

Suivi fonctionnel avec Trello

L'outil **Trello** a été utilisé pour planifier et suivre le développement des fonctionnalités issues du cahier des charges. Le tableau suivait un flux en colonnes représentant les différentes étapes du cycle de vie d'une tâche.

Colonnes utilisées :

- **Backlog** : toutes les fonctionnalités issues du cahier des charges à développer, classées par priorité
- **Conception (+mobile !)** : tâches nécessitant une réflexion ou une maquette préalable
- **À faire** : tâches à développer pendant le sprint
- **En cours** : tâches en cours de réalisation
- **À terminer plus tard** : tâches temporairement mises en attente (ex : dépendance à une donnée non disponible)
- **Terminé 🎉** : tâches finalisées et testées, prêtes à être livrées

Chaque carte Trello représentait une fonctionnalité complète. Elle comprenait :

- une description claire du besoin fonctionnel
- l'utilisateur concerné
- les données attendues
- des règles métier éventuelles
- des commentaires pour suivre les ajustements

Les étiquettes de couleur permettaient d'identifier facilement les vues associées (ex. : Menu application, Page d'accueil, Historique des covoiturages, etc.).

Exemple d'une carte :

Détail d'un covoiturage

dans la liste **TERMINÉ** ▼

Étiquettes

Vue détaillée d'un covoiturage +

Notifications

👁 Suivre

Date de début

23 févr. ▼

Description

Modifier

📄 Liste covoiturages disponibles avec champs **TERMINÉ** depuis le bouton détail

"Utilisateur concerné : Visiteur, Utilisateur"

Afficher tous les champs présent sur la liste +

- Les avis du conducteur
- Le modèle ainsi que la marque de véhicule (également l'énergie utilisé)
- Visionner les préférences du conducteur"

+ **Bouton Participer**

Pièces jointes

Ajouter

Liens

Figma mobile ...

Figma ...

Suivi technique avec GitHub

En parallèle de l'organisation fonctionnelle du projet via Trello, tous les aspects liés à la maintenance technique, aux améliorations et aux changements de code ont été gérés directement dans GitHub, via le système d'issues.

Pourquoi GitHub ?

- Pour distinguer clairement les tâches techniques des tâches fonctionnelles
- Pour suivre l'historique des modifications
- Pour lier directement les correctifs aux branches et commits

Système d'étiquettes

Pour faciliter le tri et la compréhension rapide, les issues ont été classées avec plusieurs labels personnalisés :

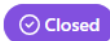
- **Type** : front improvements, back improvements, enhancement
- **Localisation** : page-userSpace, page-login, page-carpoolDetails, etc.


Cette structuration permettait :

- d'identifier rapidement l'impact d'un changement
- de retrouver efficacement une amélioration ou un bug
- de maintenir une qualité de code tout au long du projet

Exemple d'issues traitées


Popup create a new car -> avoid the refresh page #112


 Closed





SoCode9 opened on Mar 31


In the new carpool page, if I add a new car -> the page is reloaded -> I lost the carpool's informations
Solution : reload only the car's section

Create sub-issue 


 SoCode9 added **front improvements** **page-userSpace** on Mar 31

 SoCode9 self-assigned this on Mar 31

 SoCode9 mentioned this last week
[redo functions in the Car class #178](#)

 SoCode9 closed this as **completed** last week

check if the email already exists #150



 Closed




SoCode9 opened last month


...


No description provided.

Create sub-issue  

 SoCode9 added **back improvements** **page-adminSpace** **page-login** last month

 SoCode9 self-assigned this last month

 SoCode9 mentioned this last month

 [add error message if the email already exists in DB when i create a new user #155](#)

 SoCode9 closed this as **completed** last month