EcoRide - documentation

La gestion du projet

*Étant seul sur ce projet, j'ai opté pour **Trello** plutôt que Jira pour le suivi car il est plus adapté à une gestion visuelle, légère et efficace des tâches, sans s'encombrer d'outils qui seraient superflus dans ce contexte. Je définis moi-même les priorités, les difficultés et les délais pour chaque tâche, et toutes les cartes me sont attribuées par défaut.*

*Ce projet suit un modèle de **gestion prédictif,** car la demande est bien définie dès le départ et il n'y a pas d'interaction avec le Product Owner durant le développement, ni de livraisons partielles. J'ai choisi une **gestion en cascade** (Waterfall), qui permet de structurer les étapes du développement et de valider chaque fonctionnalité au fur et à mesure de son implémentation. Ce choix, plutôt que celui du modèle en V, est motivé par mon souhait de réaliser les tests en fin de développement.*

*Concernant l'implémentation des fonctionnalités, j'opte pour un mode de **développement incrémental** en déclinant les user stories définies dans le cahier des charges du client les unes après les autres.*

Ces choix sont matérialisés dans Trello de la manière suivante *:*

*1. **Kanban traditionnel** - je suis parti d'un modèle classique avec trois colonnes : "À faire", "En cours" et "Terminé". J'ai ajouté une limite de deux cartes simultanées dans la colonne "En cours" (renommée "Work in Progress") pour éviter la dispersion et m'assurer que les actions engagées soient finalisées avant d'en commencer de nouvelles.*

*2. **Éléments de Scrum** - pour rythmer le travail, j'ai divisé la colonne "À faire" en deux :*

- une colonne "Backlog" qui recense toutes les fonctionnalités à développer classées par priorité. Cette colonne est revue à chaque fin de sprint pour ajuster les priorités et les délais en fonction de l'avancement.

- une colonne "Sprint" qui contient la fonctionnalité ou le groupe de fonctionnalités que je souhaite réaliser. Chaque fonctionnalité est déclinée en tâches unitaires qui sont réunies dans une checklist avec des temps de réalisation estimés.

*3. **Gestion du gitflow** : pour suivre l'intégration continue du code, j'ai également divisé la colonne "Terminé" en deux :*

- une colonne "Done" pour les tâches ou user stories finalisées et validées. Si la tâche n'est pas complétement terminée à la fin du sprint, elle retourne dans le "Backlog".

- une colonne "Merged" qui contient toutes les fonctionnalités fusionnées dans la branche principale du dépôt git (géré selon les principes de Gitflow).

Concrètement, le projet est divisé en six grandes étapes :

*1. **L'analyse du besoin** qui donne une vision globale des attentes et permet de définir le stack technique qui sera utilisé en production (et par conséquent, l'environnement de développement à mettre en place). Lors de cette étape :*

1. une documentation détaillant les choix de technologies est rédigée,

2. ainsi qu'une autre concernant la configuration de l'environnement de développement.

*2. **La modélisation des spécifications fonctionnelles et techniques** (réalisée avec draw.io). Cette étape complète la documentation avec :*

1. un diagramme des cas d'utilisation,

2. MCD, MLD, MPD et diagrammes de classes,

3. un premier diagramme de séquence.

*3. **Le maquettage** réalisé avec Figma (documents ajoutés à la documentation.) comprenant :*

1. la charte graphique,

2. les wireframes et mockups (mobile et desktop) des pages :

- d'accueil,

- de vue des covoiturages,

- et de vue détaillée d'un covoiturage.

4. ***L'initialisation du projet*** *qui inclut :*

   1. *l'initialisation de git et la création d'un repository GitHub,*
   2. *l'installation et la configuration des technologies choisies,*
   3. *la création des entités et des documents,*
   4. *la création des bases de données,*
   5. *la rédaction du README.md.*

5. ***Le développement des fonctionnalités*** *des user stories divisées en 7 sprints :*

   1. *Le menu, la page d'accueil et les mentions légales (US 1 et 2)*
   2. *L'inscription, la connexion, et la vue espace utilisateur (US 7 et 8)*
   3. *L'ajout, la recherche, les filtres et les vues des covoiturages (US 3, 5 et 9)*
   4. *La participation, le démarrage, l'arrêt et l'historique des covoiturages (US 6, 10 et 11)*
   5. *Le dépôt et la validation des avis, l'espace employé (US 12)*
   6. *La gestion de compte, la vue d'indicateurs et l'espace "administrateur" (US13)*
   7. *Le filtrage des résultats de recherches de covoiturages (US 4)*

   + La création de diagramme(s) de séquences pour les process qui le nécessitent.

6. ***Le déploiement en production****, la rédaction d'une documentation détaillant le processus, et la rédaction de la présente documentation.*

Les user stories ont été traitées dans l'ordre suivant :

1. *US 1 – Page d'accueil*
2. *US 2 – Menu de l'application*
3. *US 7 – Création de compte*
4. *US 8 – Espace utilisateur*
5. *US 9 – Saisir un voyage*
6. *US 3 – Vue des covoiturages*
7. *US 5 – Vue détaillée d'un covoiturage*
8. *US 6 - Participer à un covoiturage*
9. *US 10 – Historique des trajets*
10. *US 11 – Démarrer, arrêter un trajet*

*Pour le rendu de l'ECF, une **release** a été faite et **mergée** dans la branche **main**, puis poussée en **production**.*

À l'heure de la rédaction de ces lignes, 3 user stories restent à implémenter :

1. *US 12 - Espace employé*
2. *US 13 - Espace administrateur*
3. *US 4 - Filtres des covoiturages*