

Mediscreen



Directives de sprint pour les développeurs à distance

Bienvenue dans l'équipe de développeurs à distance de Foot Forward Development Consultancy ! Ce document vise à vous familiariser avec notre processus de sprint interne et est destiné aux développeurs travaillant à distance.

Présence à la réunion et communication

Les conditions de participation aux réunions sont définies par votre responsable direct. Pour toutes les réunions auxquelles vous ne pouvez pas assister, vous recevrez un résumé, ainsi qu'une liste d'activités à réaliser.

Veuillez noter que les réunions trimestrielles à l'échelle de l'entreprise sont obligatoires, quel que soit le fuseau horaire.

Processus de sprint

Notre moyen de communication principal est le tableau Kanban. Il répertorie les stories sur lesquelles vous travaillez actuellement, ainsi que les fonctionnalités à compléter dans les prochains sprints. Lorsque vous commencez à travailler sur une story, déplacez la carte dans la colonne « En cours ». Vous devez ensuite cocher les tâches à mesure que vous les terminez. Lorsque toutes les tâches sont terminées, déplacez la story dans la colonne « À Tester ». Lorsque le responsable du produit est satisfait, la story est déplacée dans la colonne « Terminé ».

Vous travaillerez en étroite collaboration avec le responsable du produit, qui comprend et sait ce que le client souhaite. Il a le dernier mot pour approuver une story.

Il y a aussi une colonne de questions sur le tableau Kanban. Consultez-les pour connaître les problèmes éventuels, afin de pouvoir les aborder lorsque vous rencontrerez le responsable du produit.

À la fin de chaque sprint, il y aura une réunion rétrospective. Si vous ne pouvez pas y participer, veuillez remplir le modèle ci-dessous et le passer en revue avec le responsable du produit. Bonne chance et rendez-vous au premier sprint !

Modèle rétrospectif

Ce qui s'est bien passé :

- _ Le développement des microservices patient et angular front-end ;
- _ Le déploiement sous docker.

Ce qui aurait pu aller mieux :

- _ J'aurais pu commencer par développer une ui angular plutôt qu'une ui thymeleaf;
- _ J'ai mis du temps avant de savoir comment les données des patients pouvaient être initialisées sous docker puis sauvegardées même après arrêt du container.

Ce que j'aimerais faire différemment :

- _ Ajouter un bouton delete pour supprimer les données d'un patient ;
- _ Créer un microservice utilisateur pour pouvoir créer différents comptes utilisateur