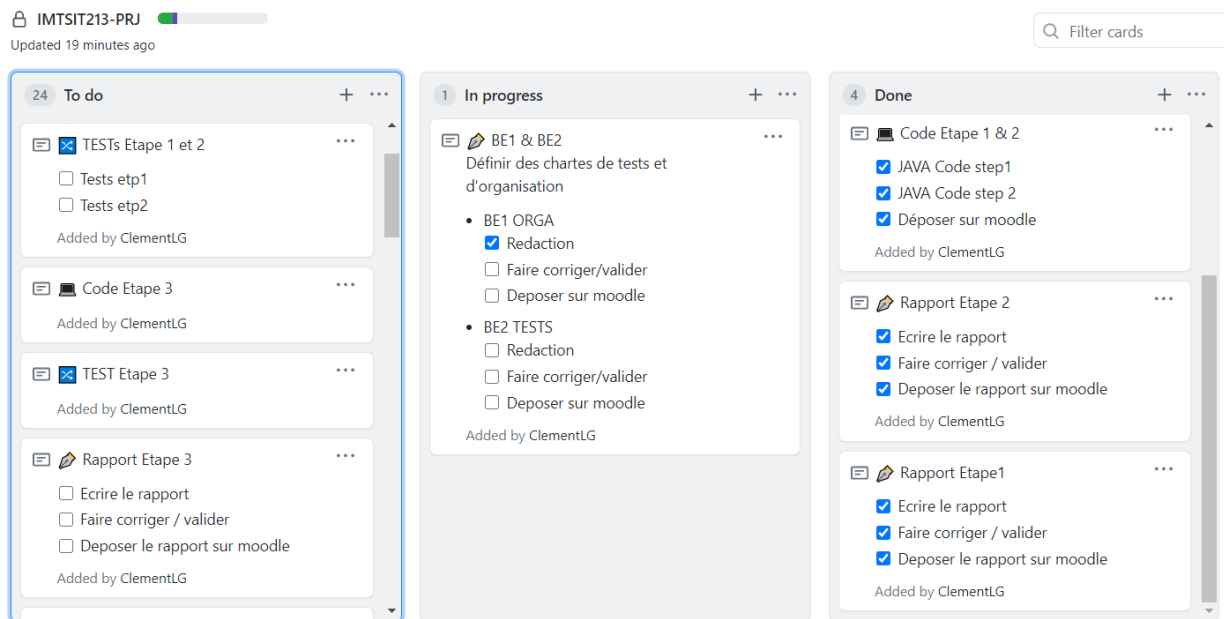


FRAIGNAC Guillaume

Organisation du travail d'équipe

Pour respecter les délais nous avons choisi d'utiliser un tableau de bord « Kanban ». Ce tableau utilise des « post-it » pour visualiser de manière très simple l'évolution du travail au cours du temps. C'est un outil qui est disponible sur GitHub et lié au répertoire.

Voici un exemple de notre Kanban :



Pour garantir la qualité de nos livrables, nous allons nous focaliser sur les points suivants :

La forme : nos livrables devront être cohérents en mise en page (utilisation de la même police, même page de garde, etc...).

Le contenu : Il doit être assez précis pour un lecteur expert du domaine, tout en restant accessible pour un lecteur qui ne l'est pas.

Les tests : c'est une partie très importante. En effet, ils nous permettent de vérifier le fonctionnement de notre code. Ainsi les tests doivent être assez exhaustifs pour couvrir l'ensemble des fonctionnalités.

Pour assurer la qualité de notre documentation nous avons mis en place des procédures de validation. Avant de rendre un livrable, nous effectuons des relectures croisées. Ainsi nous pouvons nous assurer d'une certaine qualité de documentation. Nous accordons également une importance particulière au respect des règles de bonne pratique du génie logiciel.

Pour synchroniser le travail, nous allons utiliser l'outil de versioning Git avec un répertoire hébergé sur la plateforme GitHub. Nous notons plusieurs avantages à cet outil :

- Nous pouvons créer plusieurs branches indépendantes et ainsi travailler chacun dans un environnement qui nous est dédié.
- Grâce à la fonction « reverse » de GIT, nous pouvons rattraper des erreurs commises en revenant à des versions du programme antérieures.
- Il est possible de réaliser des fusions (merge) avancées de nos fichiers entre chaque branche.

Une fois qu'une version du code est fonctionnelle et validée par l'équipe, nous pouvons « merger » notre code dans la branche master. C'est la branche qui sert de dépôt pour une version aboutit. Elle servira de point de départ à l'équipe entre chaque étape.

Afin que notre dépôt GitHub reste ordonné, nous avons imposé certaines règles :

- Des noms explicites pour chaque commit
- Faire des commit réguliers
- Le code doit au minimum correctement compiler avant chaque commit
- Utiliser des tags (add/refractor/remove) pour informer des modifications dans les commit

Avant de rendre un livrable, nous utiliserons des Frameworks de test (JUnit et Emma) afin de vérifier si le code est en mesure de passer une batterie de tests.



En termes de moyen de communication, nous utilisons Discord. C'est un outil qui facilite grandement les échanges. Nous pouvons faire des points vocaux réguliers pour échanger sur nos problèmes, questions, besoins au sujet du projet. Les échanges de fichier et l'affichage de code dans les salons textuels permettent d'améliorer notre productivité et l'efficacité de nos discussions.

Conclusion

Ce BE nous a permis d'échanger entre nous avec pour objectif d'améliorer notre efficacité dans notre travail et dans la façon de l'organiser. A partir de la prochaine séance nous mettrons à l'épreuve l'ensemble des volontés citées dans ce document. Ces règles sont là pour que l'on puisse tous être sur la même longueur d'onde et ainsi avancer plus rapidement.