Test technique Dougs - Analyse production

1. Analyse fonctionnelle

   Après lecture de l'énoncé, j'ai essayé de représenter le problème en prenant différents exemples sur papier. Ces exemples m'ont permit de dégager un ensemble de cas d'échecs du processus de validation. Certains de ces cas étaient déjà explicités dans la présentation du contexte (ex: mouvements manquants, mouvements dupliqués). D'autres cas, n'étaient pas explicités dans le document, mais semblaient toutefois importants aux yeux du comptable utilisant le logiciel (ex: manque de soldes bancaires, mouvements en dehors de la plage temporelle définie par les soldes bancaires).

2. Méthode de développement

   Afin de valider l'ensemble des cas découverts précédemment, il m'a semblé judicieux de mettre en place une démarche de TDD, en commençant par modéliser les différents éléments du processus (solde bancaire, mouvement bancaire, message d'alerte) et en mettant en place un ensemble de tests unitaires couvrant les différents cas. Afin de vérifier le fonctionnement cohérent de l'ensemble, j'ai mis en place une interface swagger qui m'a permis de documenter de tester de bout en bout l'api web.

3. Design technique

   Concernant le design technique, j'ai séparé le code en 3 couches. Une regroupant la logique métier. Une regroupant le code liée à la mise en place de l'api web, et enfin la dernière permettant de faire communiquer les deux autres. L'idée étant de mettre en place les différentes macros couches d'une clean architecture (domain/application/infra) sans aller trop loin (pas nécessaire pour la taille du test). Concernant, le design de l'api, je n'ai pas souhaité reprendre le schéma d'url proposé. Il m'a semblé plus judicieux de considérer une ressource de synchronisation et son processus de validation plutôt que la validation d'un ensemble de mouvements. Pour la gestion des causes d'échecs « reasons », je suis parti sur l'idée que l'api serait essentiellement consommée par des applications mobiles ou PWA. J'ai donc opté pour la mise en place d'un système de code et métadonnées, qui permettront potentiellement de gérer plus finement l'affichage des messages/ alertes sur les applications clientes (ex: interpolations avec stylisation, internationalisation).