

Remarques:

-Générales:

- Version ancienne de Flutter et ses dépendances
- Duplication des widgets
- Code explicitement implémenté manquant plus d'encapsulation
- Mix de la partie logique avec la partie ui et conservation des commentaires
- Mix du français et d'anglais causant une ambiguïté dans la signification
- Inclusion des anciennes assets dans le projet
- Traduction implémenté explicitement sans cubit
- Pas d'utilisation de thème

- UI:

- Appel d'api dans l'état initial des widget tel que l'historique,
- Pas de restriction d'utilisation lors de la vérification du token.
- Recherche nécessitant pas un appel d'api si on la list dès le début on doit rechercher localement par type, par date ...
- Duplication dans les pages des facturier
- Confusion du logique avec le UI dans la page register

- Logique

- Toutes les fonctions de communication avec l'api sont confondues dans un seul fichier.
- Injection du repository et la class Network dans chaque cubit ce qui en résulte une duplication non nécessaire
- Confusion de tous les cubits dans un seul dossier

Ma proposition:

- Mettre à jour la version flutter.
- Avoir des conventions par exemple au lieu de mettre toutes les communication
- Utiliser le thème inclus dans flutter qui a des noms globales presque conformes au noms de couleur que donne le designer
- Suppression complète des commentaires inutiles
- Avoir une classe pour les fonctionnalités similaires (par exemple une classe NetworkAuth qui doit inclure les fonctions concernant l'authentification avec les fonction login, logout ..etc).
- Si un Widget est spécifique à une page, on crée un dossier et on associe le Widget.
- Créer des widgets réutilisables qui sont conformes à celles dans le global design system pour la fondation des widgets.
- Utiliser un nom significatif en anglais et en camelCase pour les fonctions concernant l'appel des apis par exemple "getSomething" pour récupérer les données "createSomething ou "postSomething" pour Poster les donnée
- Faire un appel à l'api au début de l'ouverture de l'application pour s'assurer que nous faisons une seule requête pour initialiser l'historique, les notifications etc.. au lieu de faire une appel à chaque entré de la page en plus de réduction locale du balance lors du succès de la transaction et laisser l'utilisateur si nécessaire de faire une réinitialisation manuelle
- Mettre les cubits reliés dans un dossier séparé et le par exemple "SomeCubit" sous un dossier nommé "Logic".
- Je vois pas l'utilité du routeur vaut mieux avoir une navigation explicit pour notre cas
- Se débarrasser du repository

J'accompagnerai grâce à Dieu une architecture des dossiers et du démarche qui pourra mieu expliquer ma vision lors du réunion.

Finalement, celle-ci était ma propre proposition et j'espère qu'elle facilitera le travail de mes collègues et nous mènera vers un avancement meilleur.

Merci,
Chemsdine.

