Université de Lille

Thitouane Helle

Technologie Mobile

Présentation

Dans le cadre du cours de **Technologie Mobile**, j'ai développé une **application mobile** en **Flutter**.
Cette application communique avec l'**API Hearthstone** et stocke ses données (favoris) dans une base de données **Firebase**.

Quand on démarre l'application, on arrive sur la page d'accueil, on y trouve une liste de cartes Hearthstone interactives, lorsque l'on clique dessus ça ouvre une page du détail de la carte, avec une barre d'outils qui regroupe un bouton vers le menu, un bouton vers la recherche de carte et un menu de choix du set de carte que l'on veut afficher dans la liste de carte.

Dans le détail d'une carte, nous avons des informations sur la carte et la possibilité de la mettre en favorie grâce à un bouton sur l'image de la carte, la mise en favoris déclenche une vibration et une notification, une fois mis en favoris, le bouton changera et deviendra un bouton de suppression de la carte dans les favoris, ce qui déclenche aussi une vibration et une notification, la barre d'outil nous propose de revenir en arrière.

Dans le menu nous avons le choix entre le retour à l'accueil, la recherche d'une carte, la liste des favoris, le mode sombre et quitter l'application.

Dans la page des favoris nous avons la liste intéractive des cartes mises en favoris, la barre d'outil nous propose de revenir en arrière.

Le bouton mode sombre change le thème de l'application en sombre.

Le bouton quitter nous fait quitter l'application.

Fonctionnalités

Choix des packages

bloc:

Le package BLoC était imposé, j'ai donc appris à l'utiliser dans mes interactions avec Firebase et mon api, dans mon code les interactions avec les données sont séparés des Widget comme le veut l'architecture BLoC.

firebase:
J'ai choisi d'utiliser le package Firebase pour utiliser la base de données Firestore afin de stocker les cartes mises en favoris. J'ai préféré FireStore à un stockage local pour pouvoir retrouver les données sauvegardées à chaque ouverture de l'application et parce que c'est pratique à utiliser avec l'architecture BLoC.

auto_route:
Dans le but d'avoir des routes propres et des redirections simples, j'ai utilisé le package auto_route qui m'a fait gagner plus de temps que je ne le pensais.

vibration:
Afin de pouvoir utiliser facilement les vibrations du téléphone, j'ai utilisé le package vibration.

adaptive_theme:
Pour faire un mode sombre facilement et qui marche sur toute l'application, j'ai utilisé le package adaptative_theme.

Choix de l'architecture
BLoC:
Comme dis plus tôt, BLoC était imposé.
J'ai utilisé l'architecture BLoC pour séparer le traitement de données et les widgets, ainsi j'envoyer et récupérer mes données à l'aide d'état d'objet qui changeaient en fonction de l'avancement du traitement de la donnée.

Problèmes rencontrés
BLoC:
L'apprentissage de l'architecture BLoC a été très compliqué, c'est une architecture à la fois très simple à utiliser mais très compliquée à assimiler.

Firebase:
L'apprentissage de Firestore n'a pas été très compliqué mais utiliser Firestore dans l'architecture BLoC n'a pas été une évidence, cela m'a pris un peu de temps aussi.

Solutions apportés
Contraintes mobiles:
Pour répondre aux contraintes mobiles, j'ai utilisé des OrientationBuilder pour ajouter ou enlever des colonnes aux GridView quand le téléphone changeais de mode portrait/paysage.

Capteurs / Feedback:

Afin de pouvoir donner des retours à l'utilisateur lorsqu'il met en favoris une carte (fonctionnalité principale de l'application), on déclenche une notifications et une vibration à chaque ajout/retrait de donnée dans les favoris

Interaction homme machine:

Dans l'objectif d'avoir une application claire, j'ai décidé d'implémenter une barre de navigation à l'aide du menu burger dans la barre d'outils qui regroupe toutes les fonctionnalités importantes.
Pour faciliter la navigation, il y a une barre de navigation en bas de la barre d'outils pour changer les données de la gridview en fonction du set sélectionner dans la barre de navigation.

Favoris avec repository firebase:

Le système de favoris est une base de donnée Firebase, on y fournit/enlève les données dans le détail des carte à l'aide du bouton favori et on visionne les données dans la page favoris.