

Rapport de Thomas VELU sur l'application Mobistory

VERSION 1.0

SOMMAIRE

1. Introduction

1. Objectif
2. Présentation de l'application

2. Ma participation

1. UI
2. Navigation
3. Quizz
4. L'affichage du détail d'un événement

3. Après la soutenance

1. L'affichage des événements n'est pas optimisé
2. L'affichage d'un événement est trop lent

4. Retour sur ce projet

Introduction

Objectif

Cette application a été développée dans le cadre d'un cours d'interface graphique par M. Michel CHILOWICZ. Le but de cette application est d'apprendre à des étudiants à l'ESPE en deuxième année d'ingénierie en informatique l'utilisation de Jetpack Compose sous Android.

Ce rapport vise à expliquer ma participation dans le projet Mobistory avec Marius DONNÉ.

Présentation de l'application

L'application "Mobistory" permet à l'utilisateur d'en apprendre plus sur l'histoire du monde grâce à des événements sélectionnés, ou à des quizz sur ses connaissances en histoire.

Ma participation

Les sections suivantes comprendront ma participation dans le projet ainsi que des difficultés rencontrées lors du projet.

UI

Ma plus grande contribution était le développement de l'interface utilisateur et des différentes pages.
Bien qu'il y ait eu plusieurs versions des autres composantes de l'application, il n'y a eu qu'une seule idée de design de l'application ! En effet, le choix s'est porté sur une page principale avec un tiroir qui permet à l'utilisateur de s'orienter vers d'autres pages.

La plus grande difficulté sur l'interface utilisateur était de la rendre agréable à regarder et facile d'utilisation.
Pour cela, il y a eu l'idée d'utiliser les composables vus en cours comme les Card, Box ou les LazyColumn avec des Modifier.

Navigation

La navigation concerne le fait qu'un utilisateur puisse revenir sur une page ou se faire rediriger vers une autre page.
La navigation a été développée dans l'optique d'avoir des routes comme sur un site web, donc simple pour comprendre.
Cependant, en programmation, rien est jamais simple !

Dans notre projet, il y a eu 3 versions différentes de la navigation :

- Première version : sans utiliser la librairie navigation d'android, c'était ma pire idée...

Dans cette version, il y avait un variable "by remember" qui sauvegardait le composable actuelle.

Cela relevait plusieurs problèmes, l'utilisateur ne pouvait pas revenir en arrière et le code devenait illisible ;

- Deuxième version : en utilisant la librairie navigation d'android, c'était un peu mieux !

Les routes étaient définies statiquement et redirigeaient correctement les utilisateurs.

Le seul problème était pour l'affichage du détail d'un événement, en effet, les détails ont un identifiant et donc peuvent changer en fonction de l'événement ;

- Troisième version : en utilisant la librairie navigation d'android avec des routes "dynamiques".

Avec la librairie, j'ai vu que nous pouvions donner, en tant

qu'arguments, des identifiants, ce qui a permis une meilleur navigation pour l'utilisateur.

Quizz

Le quizz a été une fonctionnalité plutôt complexe à mettre en place. En effet, dès le début avec Marius DONNÉ, nous avons deux idées distinctes du quizz :

- Mon idée : faire un quizz "infini" avec un événement et l'utilisateur doit choisir une bonne réponse sur les quatre proposées ;
- L'idée de Marius DONNÉ : remettre en place l'ordre chronologique de plusieurs événements ;

Après discussion, mon idée a été retenu et mise en place dans l'application.

Pour mettre en place la logique, le plus simple était de prendre tous les événements, prendre sa date, choisir d'autres dates auprès d'autres événements et stocker dans une data class.

Puis de dresser une liste "shuffled" dans le jeu, pour qu'il y ait un ordre différent de questions à chaque jeu.

L'affichage du détail d'un événement

Cette affichage a été une difficulté énorme pour moi.

En effet, appeler les différents éléments (titre, description, image) sans compromettre la performance de l'application.

Par exemple, il y a eu des bugs comme des "=" en plus.

À noter que les "=" définissent un titre de section.

Pour régler ça, nous avons fait un Regex sur les "=" afin de créer une map pour lier le titre et le contenu.

Avant la soutenance, lors du click de l'utilisateur sur un événement, les téléchargements étaient effectués.

Avec une bonne connexion, ça ne prenait pas trop longtemps, mais pendant la soutenance, cela prenait 30~60 secondes pour ouvrir un événement.

Après la soutenance

Plusieurs remarques ont été effectuées comme :

- L'affichage des événements n'est pas optimisé, la liste des événements est recalculée à chaque recherche/filtre ;
- L'affichage d'un événement est trop lent, télécharger directement après le click d'un utilisateur n'est pas conseillé ;

L'affichage des événements n'est pas optimisé

Dans la version de notre application avant la soutenance, la liste était recalculée à chaque fois.

C'est-à-dire qu'à chaque rechargement du composable, une requête SQL était effectuée pour récupérer la liste des événements.

De plus, la recherche textuelle n'était pas optimisée non plus.

Pour régler ces problèmes, nous avons mis en place des `remember` sur la liste des événements et une liste filtrée des événements.

La liste des événements effectue une recherche SQL qu'une fois grâce au `remember`, seulement la liste filtrée sera changée lors d'une application d'un filtre ou d'une recherche.

L'affichage d'un événement est trop lent

Dans la version de notre application avant la soutenance, le téléchargement des données se faisait dans le thread courant, ce qui provoquait un ralentissement et une perte de performance.

Dans la version actuelle, une coroutine est lancée pour communiquer avec l'API de Wikidata afin d'afficher que l'essentiel (ce qui est dans notre base de données et en cache) avant de recharger la page quand les données ont fini d'être téléchargées.

Cela permet à l'utilisateur ne possédant pas une excellente connexion d'au moins avoir des éléments (titre, date, événements liés).

Retour sur ce projet

Globalement, le projet est excellent, j'ai travaillé avec un binôme excellent et très efficace.

Le projet était assez sympathique, cependant, les fonctionnalités demandées étaient peut-être de trop pour un délai si court et donc, nous avons pas pu le rajouter (exemple : le multijoueur en bluetooth).

Les retours du professeur lors de la soutenance nous ont aidé pour mieux travailler et cerner les problèmes majeurs de l'application.

Je tiens à remercier Marius DONNÉ pour son travail exemplaire et à M. Michel CHILOWICZ pour ses cours et son aide lors du projet.