LE BELLIER Georges
SEGALA Nino

pour jeudi 18 juin 2020

Programmation Mobile et Réalité Augmentée
-
Séquence 2

Introduction

Le but de cette deuxième séquence était d'aller plus loin que la première fois, de compléter notre première application, notamment en accédant à internet (il faut donc manipuler plusieurs threads) et en se connectant à une API où l'on peut alors récupérer des données (comme les liste d'items) au lieu d'utiliser des fichiers JSON (ce qu'on utilisait dans notre première application).
Il faut pouvoir mettre à jour l'adresse url de l'API, se connecter à internet et s'identifier auprès de l'API, puis récupérer la liste des items associés à l'utilisateur qui s'est connecté.
M.BOUKADIR, contrairement a ce qu'il avait annoncé n'a malheureusement pas déposé de correction de l'application de la séquence 1, ce qui nous aurait permis de repartir d'une application qui fonctionne bien.
Nous avons donc dû repartir de l'application que nous avions codé pour le première séquence, mais celle-ci ne fonctionnait pas complètement, il y avait encore des bugs et cela nous a limité lors de notre travail pour cette deuxième version.

Analyse

Notre première observation est que nous sommes plus à l'aise pour développer que lors de la séquence 1. Nous avons en effet plus d'expérience, de part les cours suivis, les séances de questions/réponses, ainsi que par notre travail personnel dans le projet de PMR.
Nous avons commencé par importer toutes les librairies dont nous avions besoin. Nous avons donc importé une librairie afin de gérer les coroutines ainsi que la librairie Retrofit afin de pouvoir faire des requêtes HTTP.
Nous avons également ajouter les permissions nécessaires tel que l'accès à internet, puisqu'il faut maintenant depuis plusieurs années demander la permission (à l'utilisateur directement dans certains cas, mais pas pour l'accès à internet, car c'est considéré comme une permission de base) afin de pouvoir les fonctionnalités associées.
Nous avons eu de nombreuses erreurs de compilation et qui étaient difficiles à comprendre. Cela nous a beaucoup ralenti.
Le traitement des requêtes vers l'API est géré avec les coroutines, en effet on a besoin de faire des traitements asynchrones, il ne faut pas que l'attente de la réponse à nos requêtes bloque tout le reste de l'application.

Conclusion

Partir d'une application de départ qui fonctionne bien nous aurait bien aidé pour traiter ce sujet. Cependant, nous avons tout de même réussi, en partant de l'application que nous avions codée, à implémenter les principales fonctionnalités demandées lors de cette deuxième séquence.
Nous avons compris l'utilisation des coroutines pour traiter les requêtes vers l'API, ainsi que leurs limites. Nous avons travaillé concrètement sur une API REST et nous avons donc expérimenté l'identification ainsi que les requêtes vers cette API se trouvant sur un serveur distant.

Perspectives

Dans un premier temps nous pourrions finir toutes les fonctionnalités demandées et que nous n'avons pas pu mettre en place. Il faudrait ensuite améliorer l'expérience utilisateur qui est très négligée ici. L'application n'est pas très belle, elle paraît assez vide, il n'y a pas assez de fonctionnalités (on pourrait vouloir modifier les noms des listes ou des items, les déplacer afin de les réorganiser, ajout de couleurs ou d'étiquettes sur les listes/items, etc). L'application, malgré son fonctionnement (elle réalise ce qu'on lui demande) ne plairait pas à la majorité des utilisateurs.
L'application est également très dépendante d'internet, ce qui pour une application de todolist est assez surprenant pour un utilisateur. Il s'attend sûrement pouvoir accéder à sa todolist même sans avoir de connexion. On pourrait alors sauvegarder les données sur le téléphone, ainsi lorsque la connexion est interrompue, l'utilisateur peut continuer à consulter ses listes. On peut même imaginer qu'il puisse modifier ses todolist en étant hors-connexions et que toutes les modifications soient envoyées à l'API lorsqu'on a à nouveau accès à internet.

Bibliographie

https://kotlinlang.org/docs/reference/
https://developer.android.com/docs