Faure Maxime & Ulysse Gaétan

Compte Rendu du TEA 1

Introduction

Dans ce TP, le but était de créer une application Android en Kotlin pour gérer des listes de tâches par utilisateur. L'application est composée de plusieurs activités (comme l'accueil, les listes, les réglages…) qui permettent à l'utilisateur de créer et voir ses listes. On a utilisé des outils pour sauvegarder les données avec des fichiers JSON grâce à la bibliothèque GSON. On a aussi appris à garder les données même après avoir quitté l'application, grâce aux préférences partagées.

Analyse du travail réalisé

Architecture générale

L'architecture de l'application repose sur 4 activités principales :

- **MainActivity** : écran d'accueil où l'utilisateur entre son pseudo. Ce pseudo est sauvegardé automatiquement et réaffiché à chaque ouverture.
- **ChoixListActivity** : permet d'afficher les listes associées à l'utilisateur, avec options de création ou de sélection.
- **ShowListActivity** : affiche les éléments d'une liste sélectionnée et permet leur gestion (ajout, suppression, édition).
- **SettingsActivity** : interface de préférences personnalisée avec `SettingsFragment`.

Les données utilisateur sont centralisées dans une structure manipulée via `ProfileToDo`, `ListeToDo`, et `ItemToDo`.

Fonctionnalités implémentées

- Saisie et sauvegarde du pseudo via SharedPreferences dans `MainActivity`.
- Ajout de liste et d'éléments dans les listes (`ChoixListActivity & ShowListActivity`).
- Persistance des données en local via `DataPersistence`, qui utilise la sérialisation JSON avec la bibliothèque GSON.

- Interface de configuration des settings via une activité de préférences basée sur `PreferenceFragmentCompat`.

Améliorations apportées

- **Structure de données bien modulaire :**
    - `ProfileToDo` contient plusieurs `ListeToDo`.
    - Chaque `ListeToDo` contient une liste d'`ItemToDo`.
- Utilisation centralisée de **`DataPersistence`** pour gérer lecture/écriture des profils.
- Code proprement organisé avec séparation des responsabilités (sérialisation, affichage, logique utilisateur).

Difficultés rencontrées

- ActionBar : nous avions commencé par modéliser la barre grise en haut de l'application par un LinearLayout horizontal afin de pouvoir mettre à son extrémité une image des trois petits points qui, lors de son clic, aurait redirigé directement vers la SettingsActivity. Nous avons ensuite appris l'existence de l'ActionBar à activer dans le fichier …
- Gestion de la persistance multi-utilisateur : il a fallu concevoir un système où chaque pseudo correspond à un profil utilisateur, sérialisé indépendamment.
- Navigation entre les activités en maintenant la bonne référence au pseudo et aux listes sélectionnées.
- Gestion des identifiants pour les listes et leurs items lors de l'ajout et lorsqu'on coche les items des listes.

Conclusion:

Ce projet nous a permis de mieux comprendre comment fonctionne une application Android. On a appris à créer plusieurs écrans qui communiquent entre eux, à afficher des listes dynamiques, et surtout à sauvegarder les données de façon durable avec des fichiers JSON. Grâce à ce TP, on sait maintenant gérer des profils utilisateurs, enregistrer leurs listes de tâches, et rendre tout ça accessible à chaque ouverture de l'application. C'était une

bonne occasion de mettre en pratique ce qu’on a vu en cours, et de mieux organiser notre code pour qu’il soit clair et facile à faire évoluer.