Rapport n°1

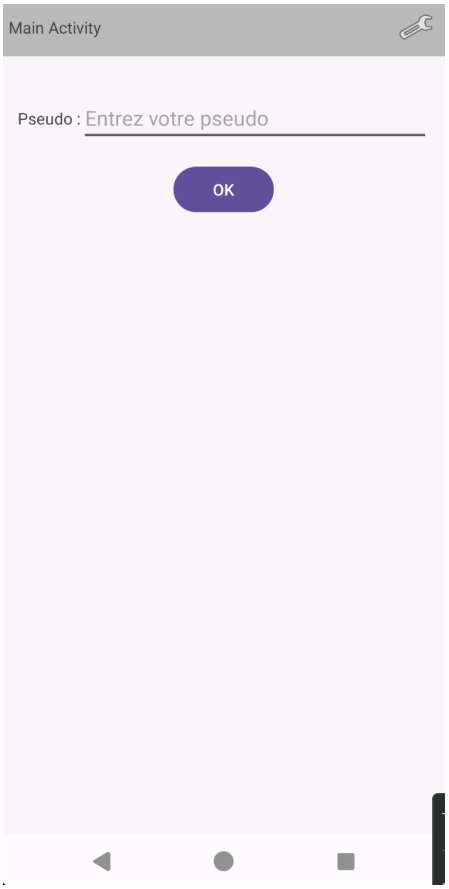
I. Introduction

Le but de ce TEA était de commencer à développer une application Android en kotlin, permettant la gestion de listes de tâches personnalisées. Cette première séquence a porté sur la mise en place de l'interface principal, de la persistance du pseudo utilisateur ainsi que de la navigation entre les différentes activités.

II. Analyse

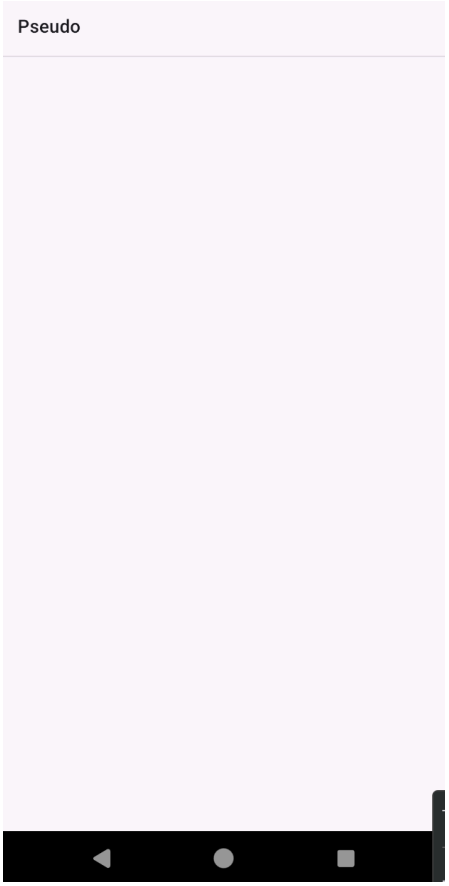
Il y avait quatre activités à réaliser pour cette application :

1. *MainActivity* :
   La main activity est celle qui s'affiche au lancement de l'application. Elle permet à l'utilisateur d'entrer son pseudo et d'accéder à ses données dans les prochaines activités. Cette activité redirige vers ChoixListeActivity en enregistrant le pseudonyme de l'utilisateur dans les préférences de l'application.

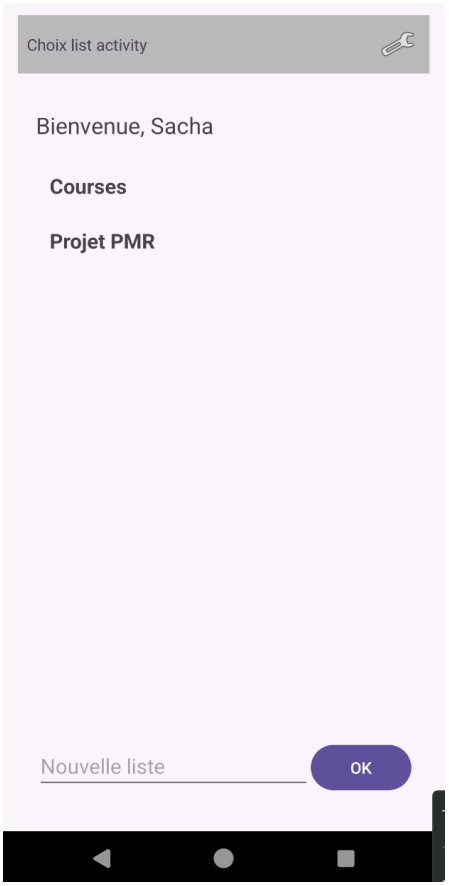


2. *SettingsActivity* :
   Cette activité permet de consulter et de modifier les préférences de l'application. Pour le moment, elle ne permet que de modifier le pseudo de l'utilisateur, mais on

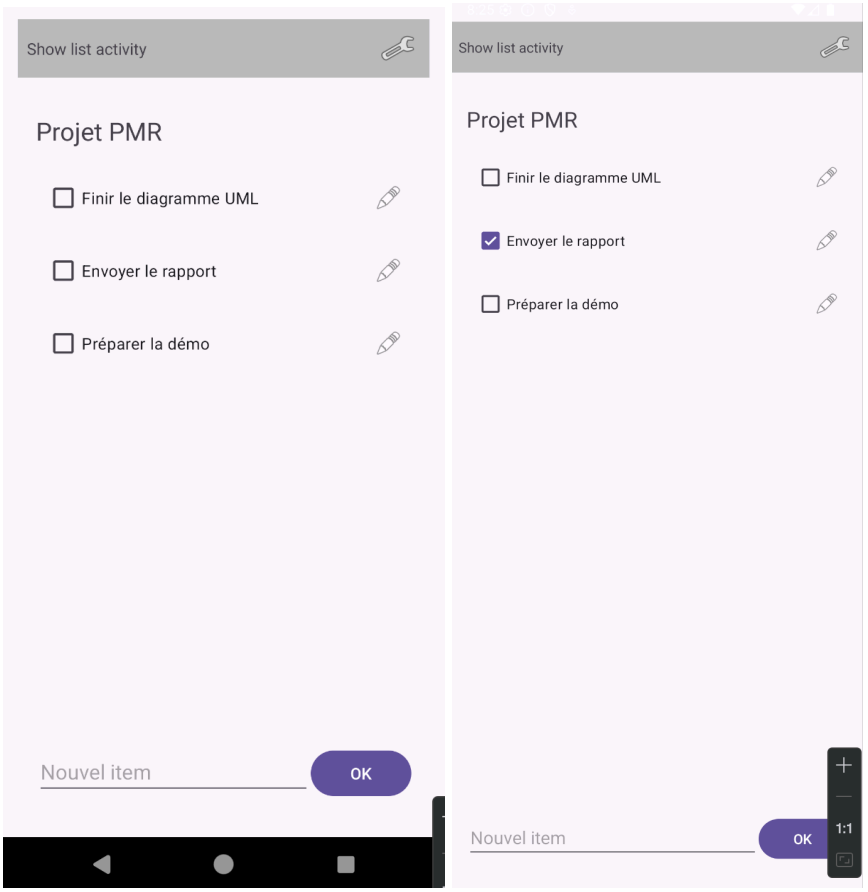
peut imaginer d'autres options pour une application plus fournie en fonctionnalités. Elle hérite de la classe PreferenceActivity et utilise le fichier R.xml.preferences pour définir les options qu'elle doit afficher.



3. *ChoixListActivity* :
   Cette activité permet de consulter les to do lists de l'utilisateur connecté, en supprimer ainsi que d'en créer de nouvelles. Il est également possible de supprimer les todo listes par clic long. Son fonctionnement est expliqué plus bas dans le rapport.



4. *ShowListActivity* :
   Cette activité permet d'accéder aux items de la to do list sélectionnée par l'utilisateur et d'en ajouter de nouvelles. Son fonctionnement est expliqué plus bas dans le rapport.



Les activités MainActivity, ChoixListActivity et ShowListActivity héritent d'une classe TemplateActivity qui contient la mise en forme et les fonctionnalités présentes sur ces trois pages, c'est-à-dire la barre du haut qui permet de se localiser dans l'application et d'accéder aux paramètres. Elle est reliée à au layout "template_activity" et permet de charger le layout des autres activités avec une méthode "loadView".

La gestion des profils se fait avec une classe DataManager qui permet de charger et de sauvegarder un profil d'utilisateur, avec ses to do lists.

L'application est aussi composé des data class suivantes :

- ProfilListeToDo qui définit la structure de données d'un profil.
- ListeToDo qui définit la structure de données d'une liste ainsi que des méthodes qui permettent de rechercher un élément ou d'en ajouter.
- ItemToDo qui définit la structure de données d'un item de to do list.

Les activités ChoixListActivity et ShowListActivity utilisent une RecyclerView afin de maximiser les performances de l'application et de minimiser sa consommation en batterie et en mémoire. Ces recycler view utilisent les adaptateurs ListAdapter et ItemAdapter respectivement pour les deux pages.

III. Conclusion

L'application permet désormais de gérer plusieurs activités coordonnées, avec une interface homogène grâce à la classe TemplateActivity. Les données des utilisateurs peuvent être conservées grâce aux préférences partagées et structurées via les classes ItemToDo, ListeToDo et ProfilListeToDo. L'affichage des listes utilise un RecyclerView relié à un Adapter, ce qui rend l'interface dynamique. Enfin, la sérialisation avec GSON assure la persistance des données entre les sessions.

IV. Perspectives

Il reste cependant quelques voies d'amélioration comme l'ajout de fonctionnalités telles que l'historique des pseudonymes utilisés, la suppression d'items (pour l'instant uniquement suppression de todo listes), ou une liste de paramètres plus étoffée.

Cependant, ce premier travail nous donne une bonne base pour les prochains TEA qui nous permettront d'implémenter une connexion réseau dans l'application ainsi que la gestion d'une base de données.