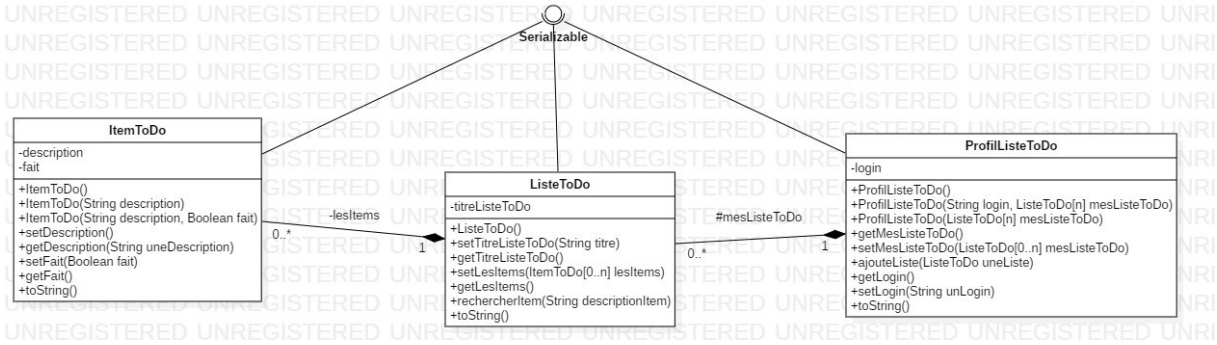
COMPTE RENDU SEQUENCE 1

Introduction :

L'objectif de cette séquence est de créer une application To Do List, qui permet à un utilisateur de créer des listes et d'ajouter des items (des tâches) à ses listes. Un fois une tâche effectuée, on doit avoir la possibilité de la repérer (avec une checkbox par exemple).
Chaque utilisateur doit pouvoir accéder à ses propres listes en entrant son identifiant. Il doit aussi pouvoir accéder à ses préférences. Pour réaliser cette application, nous allons nous appuyer sur le diagramme de classes suivant.



Présentation des classes :

ItemToDo : Cette classe permet la création de tâches, définies éventuellement par une description et un statut (fait ou non fait).
ListeToDo : Cette classe permet la création de listes. La méthode rechercheItem permet notamment d'accéder à la liste des items qui composent chaque liste.
ProfilListeToDo : Cette classe permet de définir les profils des utilisateurs ainsi que d'accéder aux listes qu'ils ont créées.

Présentation des activités :

Notre application se définit par 4 activités principales. La première, MainActivity, permet à l'utilisateur d'entrer son identifiant pour accéder à ses listes. Cette activité possède aussi une barre d'action qui permet à l'utilisateur d'accéder à ses préférences.

La seconde, SettingsActivity, est lancée depuis le bouton préférences du menu déroulant de la MainActivity. Elle permet notamment de changer le nom d'utilisateur affiché par défaut dans l'activité principale.
La troisième, ChoixListActivity, présente toutes les listes de l'utilisateur dont on a entré le nom dans la MainActivity. Elle est lancée après click sur le bouton OK dans MainActivity.
La dernière, ShowListActivity, affiche la liste des items d'une liste après avoir clické dessus dans ChoixListActivity.

Création des interfaces graphiques

MainActivity : Cette interface est constituée d'une barre d'outils, qui contient le bouton pour accéder à ses préférences, d'un TextView qui affiche « Pseudo », d'un PlainText qui permet à l'utilisateur d'entrer son pseudo, ainsi que d'un bouton « OK » qui permet de le valider ».

Aperçu de l'activité principale

ChoixListActivity : Elle est composée d'un Recycler View, qui permet d'afficher toutes les listes de l'utilisateur, d'un PlainText et d'un bouton « OK » qui permettent d'ajouter une nouvelle liste. On a ici décidé d'utiliser un Recycler View plutôt qu'un ListView, afin de pouvoir gérer un grand nombre de liste sans avoir à toutes les charger en même temps. En effet, avec un Recycler View, on a besoin de charger que les listes qui vont être affichées à l'écran, plus deux ou trois listes supplémentaires.

Aperçu ChoixListActivity

ShowListActivity : Elle est composée, de même que l'activité précédente, d'un Recycler View, permettant l'affichage de tous les items d'une liste, ainsi que d'un Plain Text et d'un bouton « OK » qui permettent d'ajouter des items à la liste sélectionnée. En revanche, dans cette activité, on place devant chaque item une checkbox qui permet à l'utilisateur de visualiser les tâches réalisées et celles qui restent à faire.

Retour sur le travail réalisé :

Au niveau chronologique, nous avons commencé par mettre en place toutes les interfaces graphiques des différentes activités, à part celle de la SettingsActivity, qui demandait plus de connaissances. Ayant l'habitude d'utiliser les ConstraintLayout plutôt que les Linear, nous avons opté pour cette solution dans la majorité de nos affichages, car celle-ci n'a pas de réelle influence sur la performance. L'affichage se voulait volontairement très simple, très épuré : on recherche surtout l'ergonomie, une prise en main simple. C'est avant tout un outil, on l'utilise pour l'aspect fonctionnel. L'aspect visuel serait évidemment à approfondir avec plus de temps.

La mise en place des classes n'a posé aucun problème. En suivant l'UML et en rajoutant une méthode ou deux, on arrive très vite au résultat.

Réadapter l'implémentation du RecyclerView vu en cours a été rapidement réalisé. On a choisi de dupliquer une grosse partie du code pour le RecyclerView de ChoixListActivity et celui de ShowListActivity, par soucis de simplicité. Sans doute aurions-nous pu optimiser ceci. Un léger problème s'est posé pour la gestion du clic sur les checkbox dans la ShowListActivity.

La persistance des données a quant à elle posé problème : comment mémoriser les actions réalisées sur les items à l'intérieure d'une liste ? Mémoriser les listes d'un utilisateur était facile, retenir les items d'une liste également. Mais, par exemple, enregistrer un nouvel item dans la liste 2 de l'utilisateur X n'était pas simple. Nous avons donc transmis aux activités ChoixListActivity et ShowListActivity la sérialization du profil courant, ce qui a permis de résoudre ce problème.
En dernier lieu, nous nous sommes renseignés sur différents sites/forums de développeurs à propos de l'utilisation de la PreferenceActivity. La mémorisation du pseudo par défaut a été réalisée. Mais pour ce qui est de l'historique des pseudos, nous n'avons pas trouvé le moyen d'écrire dans les fichiers de ressources depuis les activités.

Conclusion :

En conclusion, nous sommes très satisfaits du résultat, même si un problème principal persiste.
En effet, lorsqu'on ajoute des items à l'intérieur d'une liste, et qu'on effectue des actions dans la ShowListActivity, si on utilise le bouton de retour du téléphone, il faut revenir jusqu'à la sélection du profil si on veut voir les modifications faites à l'intérieur de la Liste en question. Nous pensons qu'il s'agit sans doute d'un problème de gestion de la vie des activités et des méthodes onRestart et onResume.
Ce travail étant à tous les deux notre premier projet Android, il a permis de bien fixer les bases et de pouvoir ajouter par la suite des éléments plus complexes.