Ministère de l'Enseignement Supérieur et de la Recherche et de l'Innovation
Université de Lille
École Centrale de Lille

Programmation mobile et réalité augmentée

Sujet de rapport : **Développement de l'application android ToDoList**

Rédigé Par : **Imen Kerkeni**
Wissem Chabchoub

Année universitaire : **2018 - 2019**

Table des matières

Contexte

N'oubliez plus jamais une tâche importante, Organisez votre vie puis profitez-en ! !" La vie peut parfois sembler écrasante, mais ça n'a pas à être le cas. Avec "ToDo", on vous garantie la tranquillité d'esprit et la sauvegarde de vos données. Notre application android "ToDo" qui permet aux utilisateurs de lister et sauvegarder avec simplicité leurs "ListToDo" .

La description détaillée de fonctionnalité de l'application se trouve dans les paragraphes suivantes.

Mots-clés : List, Activités, Sérialisation, Gson/Json, RecycleView

Chapitre 1

Analyse

1.1 Activités

L'application se base sur trois activités principales :

- ◇ MainActivity : dans laquelle l'utilisateur saisit son login.
- ◇ ChoixListActivity : dans laquelle on visualise une liste des ListToDo de l'utilisateur en question.
- ◇ ShowListActivity : dans laquelle on visualise les itemToDos de la liste sélectionnée.

1.1.1 Main Activity

L'activité principale permet à l'utilisateur de saisir son login dans un champ AutoComplete. Ce dernier est initialisé avec le dernier login saisie enregistré dans les préférences.

Le bouton "OK" permet de passer l'activité ChoixListActivity qu'on lui passe le login. Le login est aussi enregistré dans les préférences.

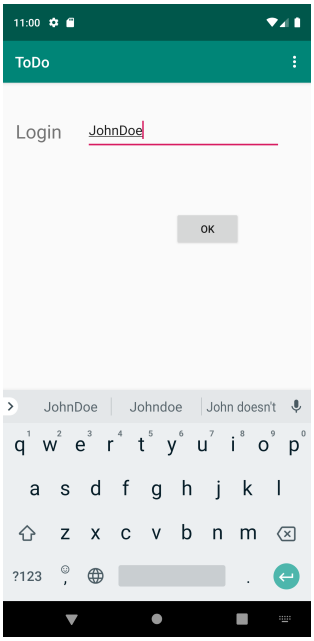


FIGURE *1.1* — Main Activity

1.1.2 ChoixListActivity

Dans cette activité, on affiche les ListeToDos de l'utilisateur en utilisant un RecyclerView. Le bouton add permet d'ajouter une nouvelle liste dont le nom est saisi dans le EditText. Le glissement d'un item permet de le supprimer. Un snack bar s'affiche dans ce cas et propose de remettre la liste supprimée. Le clique sur une liste permet de passer l'activité ShowListActivity en passant le id de la liste sélectionnée.

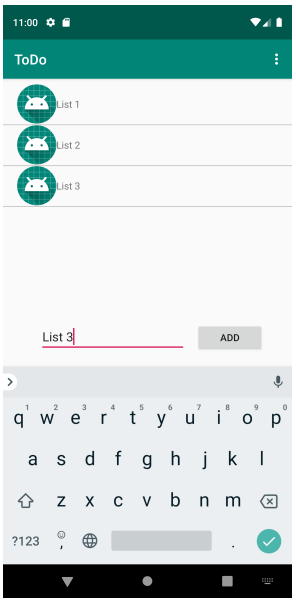


FIGURE *1.2* — ChoixListActivity

1.1.3 ShowListActivity

Dans cette activité, on affiche les ItemsToDos de la ListeToDo passé dans le intent. Les actions de glissement et le bouton Add permettent de faire les mêmes manipulations que l'activité précédente. Les ItemsToDos sont équipés avec un CheckBox.

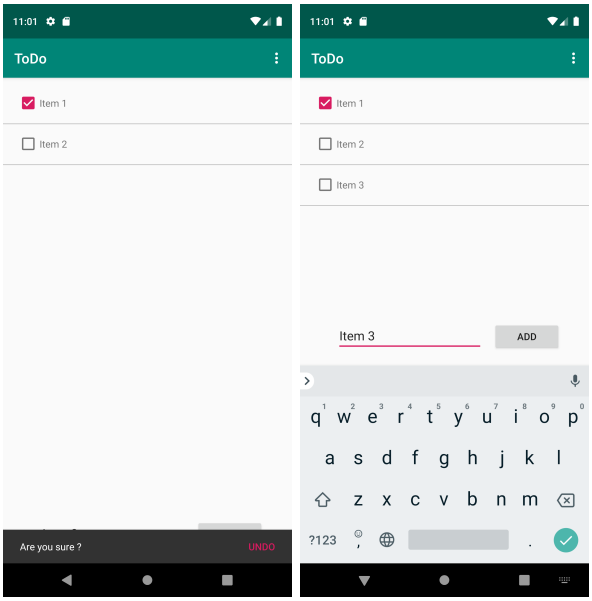


Figure 1.3 — ShowListActivity

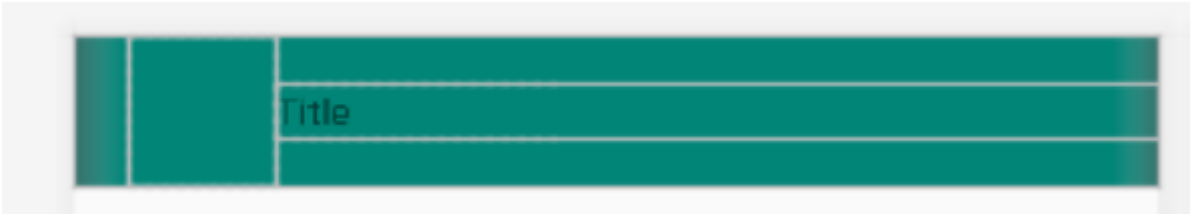
1.2 RecyclerView

Le recyclerView permet d'afficher des items en optimisant l'utilisation des ressources disponibles. Il se base sur un ItemAdapter (classe Java) et un design des items enregistré sous format xml.

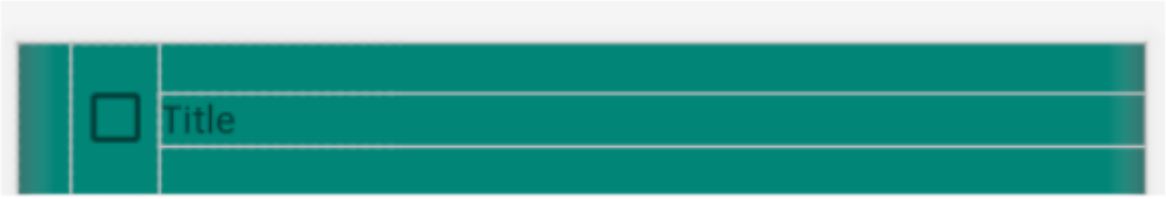
1.2.1 item_list.xml et item_item.xml

Ce fichier xml implémente le design des items du recyclerView.

i) item_list : Il concerne des items à afficher dans la liste des ListeToDos.



ii) item_item : Il concerne des items à afficher dans la liste des ItemTodos.



1.2.2 ItemAdapter.java :

Dans ce projet, on a créé deux ItemAdapters en fonction des fichiers item.xml. Leur contenu est le même à l'exception du fichier xml qu'ils utilisent.

L'item adapter contient cinq attributs :

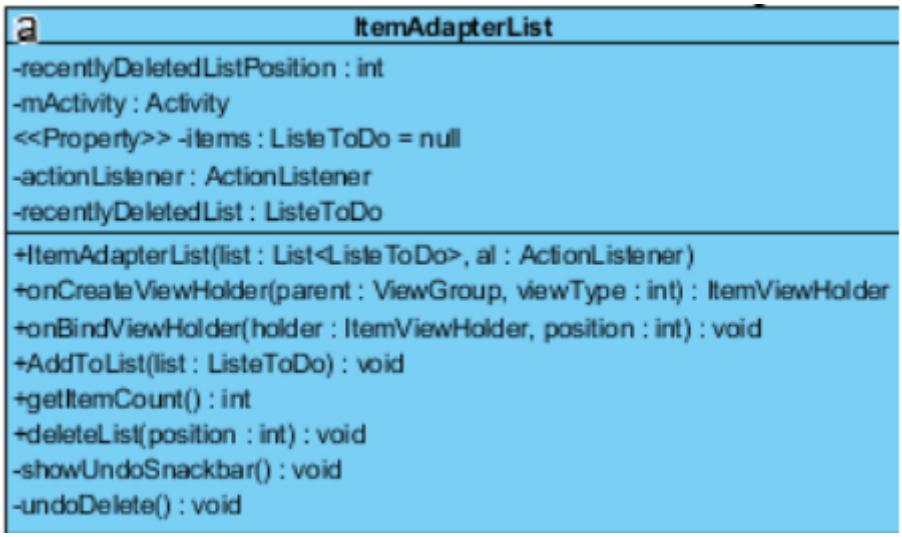
1. La liste des items à afficher
2. Une interface ActionListener qui contient des méthodes à implémenter par l'activité qui contient le RecylerView.
3. Dernier item supprimé
4. Position du dernier item supprimé
5. L'activité parente.

Le constructeur de l'adaptateur permet d'initialiser la liste des items et le ActionListener.

Dans la méthode bind, on attribue l'id de l'item au TextView ou au CheckBox pour identifier l'item en cas de clique ou de modification.

La méthode deleteListe permet de supprimer l'élément glissé et enregistrer cet élément et sa position. Si l'utilisateur choisit d'annuler la suppression, la méthode undoDelete, remet l'élément dans la liste des items dans sa position initiale.

Les méthodes deleteListe et undoDelete et le clique sur un item appellent les fonctions de l'interface ActionListener qui enregistrent les modifications dans les préférences



1.3 Persistance des données

On a choisi de structurer les ensembles des profils dans un “HashMap” afin de faciliter la distinction et la sauvegarde des ces données dans un fichier Json.

Chaque nouveau profil créé est sauvegardé dans le fichier Json “profils” situé dans les préférences de l’application. Cette opération se fait en utilisant la librairie Gson.

```
settings = PreferenceManager.getDefaultSharedPreferences(this);

//Recovering the profiles
profilsJson = settings.getString("profils","");

//Deserialization
Gson gson = new Gson();
Type type = new
TypeToken<HashMap<String,ProfilToDoList>>(){}.getType();
profils = gson.fromJson(profilsJson,type);
```

Un profil est composé d’une liste des “ListToDo” que chacune admet un identifiant et un titre. L’ensemble des listes déjà créés par l’utilisateur est sauvegardé dans le profil qui porte son login et chaque nouvelle création initialise la sauvegarde. Cette opération correspond à la sérialisation des données dans le fichier Json(Transformations des objets créés en une chaîne de caractères). En cliquant sur le nom de la liste crée, l’utilisateur est amené à sa liste des tâches à faire en utilisant la désérialisation des données déjà sauvegardées.

Plusieurs manipulations peuvent être faites :

- ◇ Création d’une nouvelle tâche
- ◇ Suppression d’une tâche

Création d’une nouvelle t’une tâche (faite ou pas faite)

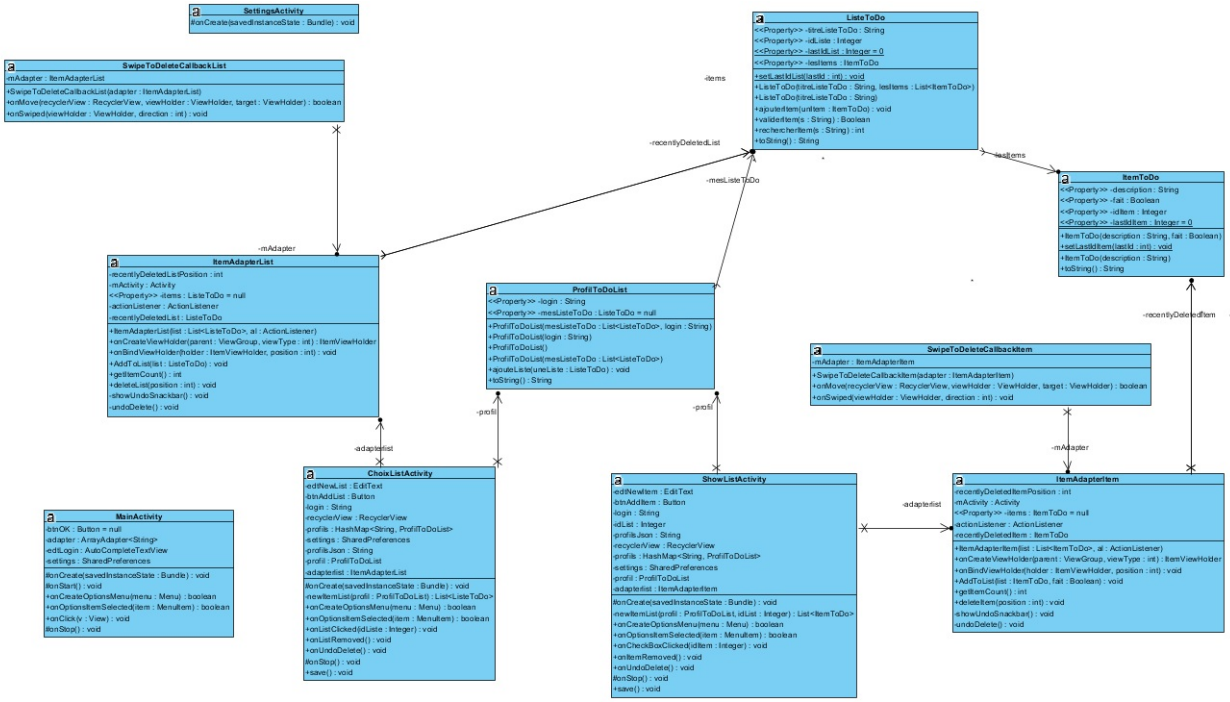
Toutes ces manipulations engendrent une mise à jour de la sauvegarde.

```
public void save(){
   Gson gson=new
GsonBuilder().enableComplexMapKeySerialization().create();
   profils.remove(login);
   profils.put(login,profil);
   profilsJson = gson.toJson(profils);
   SharedPreferences.Editor editor = settings.edit();
   editor.putString("profils",profilsJson);
   editor.commit();

}
```

1.4 Diagramme UML

Le diagramme UML complet est dans ce lien : Lien



1.5 Perspectives

L'une des perspectives possibles dans ce projet est de mettre en place une base de données Firebase. Firebase permet aussi de gérer l'authentification des utilisateurs d'une manière sécurisée.

On peut penser aussi à créer des listes "ToDo" multi utilisateurs avec un système de répartition des tâches entre les membres du groupe.

Conclusion

Ce projet était une vraie occasion pour améliorer nos connaissances en développement mobile et de découvrir plusieurs fonctionnalités et nouvelles méthodes de travail. Durant l'étape de développement de notre application on a rencontré des différentes difficultés (bugs, blocage, connaissance..) mais on a pu dépasser ces situations grâce à des connaissances et des méthodes qu'on a pu voir durant les séminaires et des recherches et des lectures bibliographiques sur internet.

Bibliographie

[1] https://medium.com/@zackcosborn/step-by-step-recyclerview-swipe-to-delete-and-

[2] https://developer.android.com/reference/android/widget/AutoCompleteTextView.html?fbclid
MV1Usva0KqKKjPrl7qeZ7SFUeheiA8

[3] https://openclassrooms.com/fr/courses/4568576-recuperez-et-affichez-des-donnees-distantes/4893781-implementez-votre-premiere-recyclerview

[4] https://developer.android.com/reference/android/preference/PreferenceActivity

[5] https://docs.microsoft.com/fr-fr/dotnet/framework/wcf/feature-details/how-to-serialize-and-deserialize-json-data