

TP4 – Développement Mobile

Persistante des données avec SQLite

Objectif

Dans ce TP, vous allez étendre le projet du TP3 pour créer une application de gestion de tâches ("ToDo List").

Ce TP vous permettra de pratiquer la persistante des données à l'aide de SQLite et SharedPreferences.

1. Objectifs pédagogiques

- Crée et manipuler une base de données SQLite via [SQLiteOpenHelper](#).
 - Gérer des préférences utilisateur avec SharedPreferences.
 - Utiliser un RecyclerView pour afficher et interagir avec des données persistantes.
 - Structurer le code Android autour de classes de données et d'adaptateurs.
-

2. Description du projet

L'application reprend la structure du TP3 :

- La première Activity reste celle d'authentification.
- Après la connexion, une deuxième Activity (appelée [ToDoActivity](#)) permet de gérer une liste de tâches ([TodoTask](#)).

Chaque utilisateur pourra ajouter, afficher, marquer comme terminée ou supprimer des tâches.

3. Structure de la deuxième Activity

L'écran est composé de deux zones principales :

1. Formulaire d'ajout de tâche (en haut de l'écran) :
 - [EditText](#) pour saisir la description de la tâche
 - [Button "Ajouter"](#) pour insérer la tâche dans la base

2. RecyclerView (occupant le reste de l'écran) :

- Affiche la liste des tâches depuis la base de données

Chaque tâche ([TodoTask](#)) contient :

- **id** : entier non signé, clé primaire auto-incrémentée
 - **isDone** : booléen (**0** = non terminée, **1** = terminée)
 - **description** : texte (contenu de la tâche)
-

4. Base de données SQLite

Créer une classe [ToDoDatabaseHelper](#) qui hérite de [SQLiteOpenHelper](#).

Structure de la table

```
CREATE TABLE TodoTask (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    isDone INTEGER DEFAULT 0,
    description TEXT NOT NULL
);
```

Méthodes à implémenter

- [insertTask\(String description\)](#) : insère une nouvelle tâche.
 - [getAllTasks\(\)](#) : retourne la liste complète des tâches.
 - [updateTaskStatus\(int id, boolean isDone\)](#) : met à jour le statut d'une tâche.
 - [deleteTask\(int id\)](#) : supprime une tâche.
-

5. SharedPreferences

Utiliser [SharedPreferences](#) pour mémoriser le nom de l'utilisateur connecté.
Ce nom sera affiché dans la barre supérieure de [ToDoActivity](#).

Méthodes utiles :

- [getSharedPreferences\("user_prefs", MODE_PRIVATE\)](#)
- [edit\(\) / putString\(\) / apply\(\)](#)
- [getString\("username", ""\)](#)

💡 Exemple :

```
SharedPreferences prefs = getSharedPreferences("user_prefs", MODE_PRIVATE);
String username = prefs.getString("username", "Utilisateur");
textViewUser.setText("Bienvenue, " + username + " !");
```

6. RecyclerView et Adapter

Créer une classe **TodoAdapter** pour gérer l'affichage des tâches dans un RecyclerView.

Éléments d'un item

Chaque élément doit contenir :

- Un TextView pour la description
- Une CheckBox pour indiquer si la tâche est terminée
- Un bouton (ou icône poubelle) pour supprimer la tâche

Fonctionnement attendu

- Lorsqu'une tâche est cochée, appeler **updateTaskStatus(id, true)** et rafraîchir le RecyclerView.
- Lorsqu'une tâche est supprimée, appeler **deleteTask(id)** puis mettre à jour la liste.

7. Exemple d'architecture minimale

```
com.emsi.mobile.todoapp/
  database/
    └── ToDoDatabaseHelper.java
  model/
    └── TodoTask.java
  adapter/
    └── TodoAdapter.java
  ui/
    ├── MainActivity.java
    └── ToDoActivity.java
  res/layout/
    ├── activity_todo.xml
    ├── item_todo.xml
    └── activity_main.xml
```

8. Livrables attendus

- **Code complet des classes :**
 - `ToDoDatabaseHelper`,
 - `TodoAdapter`,
 - `ToDoActivity`.
 - **Captures d'écran de l'application :**
 - **Affichage de la liste complète.**
 - **Ajout d'une tâche.**
 - **Marquage d'une tâche comme terminée.**
 - **Suppression d'une tâche.**
 - **Preuve de l'utilisation de `SharedPreferences` (affichage du nom utilisateur).**
-

9. Durée estimée

Environ 2 heures.

10. Bonus (facultatif)

- Ajouter une boîte de dialogue de confirmation avant suppression.
 - Permettre le filtrage des tâches (toutes / terminées / en cours).
 - Ajouter un Splash Screen affichant le nom de l'utilisateur depuis les `SharedPreferences`.
-

Rappel pédagogique

- Fermez toujours vos Cursors après lecture.
- Ne fermez pas la base après chaque requête : `SQLiteOpenHelper` gère la connexion.
- Testez vos méthodes de base de données dans le `Logcat` avant de les connecter à l'interface.