Compte-Rendu du TP3 Todo List API
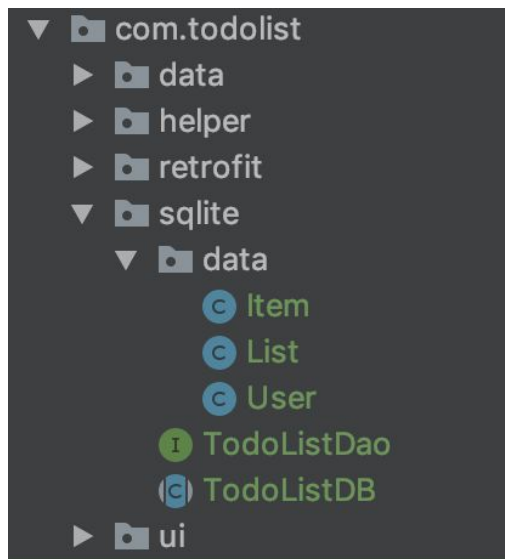
Introduction:

Les travaux déjà réalisés:

1. Créer la connexion à la base de données
2. Définir la structure de stockage local

Analyses:

La structure des codes:

Les 3 classes qui définissent la structure de la base de données:

User :

```java
package com.todolist.sqlite.data;

import androidx.annotation.NonNull;
import androidx.room.Entity;
import androidx.room.Index;
import androidx.room.PrimaryKey;

@Entity(indices = {@Index(value = {"id"})},
        tableName = "users")
public class User {
    @PrimaryKey
    @NonNull
    public int id;

    public String pseudo;

    public String password;

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getPseudo() { return pseudo; }

    public void setPseudo(String pseudo) { this.pseudo = pseudo; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }
}
```

List :

```java
package com.todolist.sqlite.data;

import androidx.annotation.NonNull;
import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;

@Entity(tableName = "list")
public class List {
    @PrimaryKey
    @NonNull
    public String id;
    public String label;
    @ColumnInfo(name = "user_id")
    public String userId;

    public String getId() { return id; }

    public void setId(String id) { this.id = id; }

    public String getLabel() { return label; }

    public void setLabel(String label) { this.label = label; }

    public String getUserId(){ return userId;}

    public void setUserId(String userId){this.userId = userId;}
}
```

Item :

```java
package com.todolist.sqlite.data;

import androidx.annotation.NonNull;
import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;

@Entity(tableName = "item")
public class Item {
    @PrimaryKey
    @NonNull
    public String id;

    public String label;

    public String url;

    public String checked;

    @ColumnInfo(name = "list_id")
    public String listId;

    public String getId() { return id; }

    public void setId(String id) {
        this.id = id;
    }

    public String getLabel() { return label; }

    public void setLabel(String label) { this.label = label; }

    public Object getUrl() { return url; }

    public void setUrl() { this.url = url; }

    public String getChecked() { return checked; }

    public void setChecked(String checked) { this.checked = checked; }

    public String getListId() { return listId; }

    public void setListId(String listId) { this.listId = listId; }
}
```

TodoListDao : l'interaction avec la base de données :

```java
@Dao
public interface TodoListDao {

    // Part List
    //----------------------------------------------------------

    @Query("SELECT * FROM list")
    java.util.List<List> getLists();

    @Query("SELECT * FROM list WHERE user_id LIKE:userId")
    java.util.List<List> findListByUserId(String userId);

    @Query("SELECT * FROM list WHERE label LIKE:labelClick")
    List findListByUserLabel(String labelClick);

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    void saveList(java.util.List<List> listsLoad);

    @Insert
    void add(List listAdd);

    @Delete
    int deleteList(List listdb);

    @Query("DELETE FROM list")
    int deleteAllList();
```

```java
    // Part Item
    //----------------------------------------------------------------

    @Query("SELECT * FROM item")
    java.util.List<Item> getAllItem();

    @Query("SELECT * FROM item WHERE list_id LIKE:listId")
    java.util.List<Item> findItemBylistId(String listId);

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    void saveItem(java.util.List<Item> item);

    @Update
    void updateItem(Item item);

    @Query("DELETE FROM item")
    int deleteAllItem();

    @Delete
    void deleteItem(Item item);
```

```java
    // Part User
    //----------------------------------------------------------------

    @Query("SELECT * FROM users")
    java.util.List<User> getAllUsers();

    @Query("SELECT * FROM users WHERE id IN(:userIds)")
    java.util.List<User> loadAllUsersByIds(int[] userIds);

    @Query("SELECT * FROM users WHERE pseudo LIKE:pseudo")
    java.util.List<User> findUserByPseudo(String pseudo);

    @Insert
    void setUser(User user);
}
```

Programmation Mobile et Réalité Augmentée
FAN Shiyu
TANG Qing

TodoListDB :

```java
package com.todolist.sqlite;


import android.content.Context;

import androidx.room.Database;
import androidx.room.Room;
import androidx.room.RoomDatabase;

import com.todolist.sqlite.data.Item;
import com.todolist.sqlite.data.List;
import com.todolist.sqlite.data.User;

@Database(entities = {User.class, List.class, Item.class}, version = 1)
public abstract class TodoListDB extends RoomDatabase{
    public abstract TodoListDao todoListDao();

    private static TodoListDB db;

    public static TodoListDB getDatabase(final Context context){
        if (db == null) {
            synchronized(TodoListDB.class){
                if(db == null){
                    db = Room.databaseBuilder(context.getApplicationContext(),TodoListDB.class,
                            name: "todolistdatabase.db").build();
                }
            }
        }
        return db;
    }
}
```