

TP 06 : Persistance des données - Bases de données SQLite -

Exercice 1.

Le but de l'exercice est de créer une application permettant d'utiliser une base de données embarquée dans le smartphone.

La base de données contient des livres. Un livre est défini par son id (un nombre qui s'incrémente automatiquement), un numéro ISBN et un titre. L'interface est dans un premier temps assez élémentaire. Elle permet de rajouter un livre dans la base ou encore de supprimer un livre (dont le numéro ISBN est saisi par l'utilisateur).

1. Concevoir les différentes interfaces permettant de répondre aux besoins décrit ci-dessus.
2. Créer une classe Livre avec les constructeurs et les accesseurs nécessaires
3. Créer une classe MaBaseSQLite qui hérite de la classe SQLiteOpenHelper. Voici le code (à compléter) de la classe :

```
package fr.enseirb.mydbsqlite;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

/**
 * Created by zemhari on 14/11/16.
 */
public class MaBaseSQLite extends SQLiteOpenHelper {

    private final static String CREATE_TABLE = "...";

    public MaBaseSQLite(Context context, String name, SQLiteDatabase.CursorFactory factory, int version){
        super(context,name,factory,version);
    }

    public void onCreate(SQLiteDatabase db){
        //On crée la base en exécutant la requête de création
        // ...
    }

    public void onUpgrade(SQLiteDatabase db,int oldVersion, int newVersion){
        //On peut fait ce qu'on veut ici mais on va juste supprimer la table et la recréer
        //...
    }
}
```

4. Créer une classe BDLivres. Il s'agit de la classe où nous allons définir réellement notre base. Voici le code à

Programmation d'applications mobiles

Travaux dirigés

compléter :

```
package fr.enseirb.mydbsqlite;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

/**
 * Created by zemmari on 14/11/16.
 */
public class BDLivres {
    private final static int VERSION_BDD = 1;
    private SQLiteDatabase bdd;
    private MaBaseSQLite maBaseSQLite;

    public BDLivres(Context context){
        //On crée la BDD et sa table :
        maBaseSQLite = new MaBaseSQLite(context,"livres",null,VERSION_BDD);
    }

    public void open(){
        //On ouvre la BDD en écriture
        //..
    }

    public void close(){
        //On ferme la BDD
        //...
    }

    public SQLiteDatabase getBdd(){
        return bdd;
    }

    public long insertLivre(Livre livre){
        //Création d'un ContentValues (fonctionne comme une HashMap)
        ContentValues values = new ContentValues();
        values.put("isbn",livre.getIsbn());
        values.put("titre",livre.getTitre());
        //on insère l'objet dans la BDD via le ContentValues
        return bdd.insert("livres",null,values);
    }

    public Livre getLivreByTitre(String titre){
        //Récupère dans un Cursor les valeur correspondant à un livre contenu dans la BDD
        Cursor cursor = bdd.query(...);
        return cursorToLivre(cursor);
    }

    public Livre cursorToLivre(Cursor cursor){
        if (cursor.getCount() == 0)
            return null;
        //...
    }
}
```

Programmation d'applications mobiles

Travaux dirigés

Passons à présent à l'UI. Vous pouvez concevoir une interface semblable à celle de la figure 1. La partie supérieure permettra de saisir les informations sur le livre à insérer et la partie inférieure servira à la recherche d'un livre par mot dans le titre. Quand l'utilisateur clique sur CHERCHER, les informations seront utilisées pour chercher le livre et le résultat sera passé par un intent à une activité (Figure 2) pour afficher les informations sur le livre trouvé.

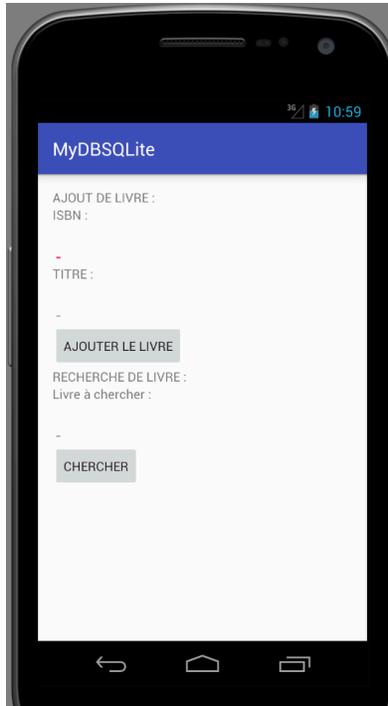


Figure 1.

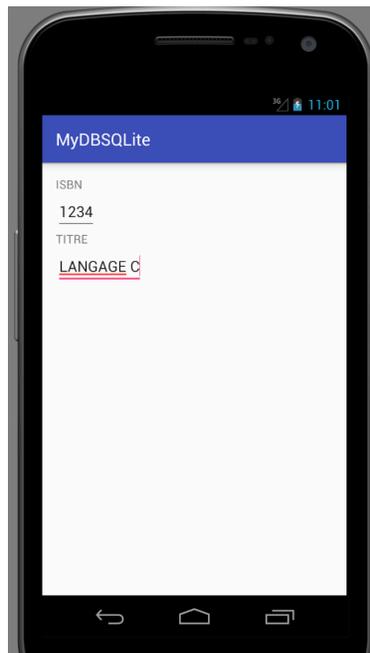


Figure 2.