Application **MyLifeCycleSQL**

L'objectif de ce TP est de découvrir les fonctions de l'API Android permettant de sauvegarder les données utilisateurs. Le point de départ sera l'application décrite Fig. 1 permettant à un utilisateur de saisir une chaîne de caractères dans une zone d'édition de texte. Le bouton « SAVE » devra sauvegarder la donnée saisie dans une base de données et le bouton « SHOW » devra afficher le contenu entier de la base de données (cf. Fig. 2).

Fig. 1

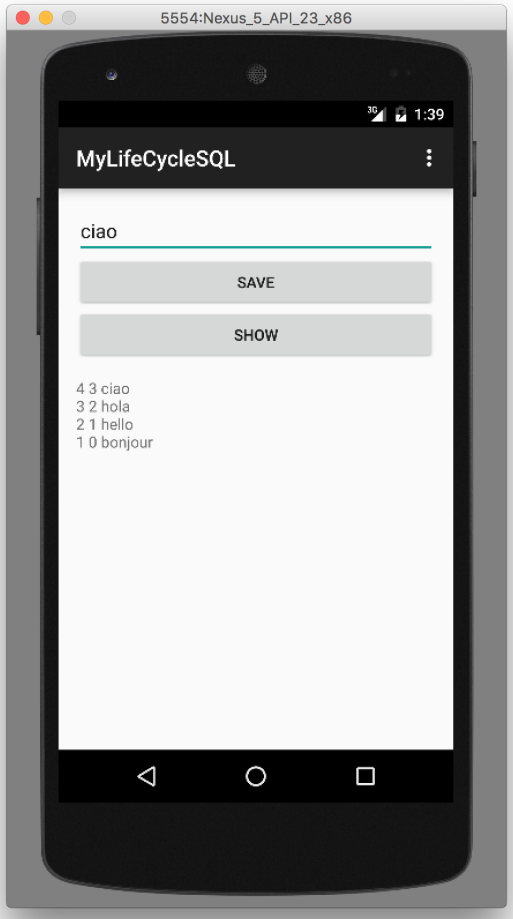

Fig. 2

Pour cela, on s'inspirera de https://developer.android.com/training/data-storage/sqlite.html, **en y appliquant quelques modifications**.

La base de données que vous allez définir sera extrêmement simple. Son nom sera **MyDataBase.db**. Elle ne sera composée que d'une seule table nommée **matable**. Cette table contiendra deux colonnes :

- **numero** de type **INTEGER**,
- **texte** de type **TEXT**.

Définissez en premier lieu une classe dite « contrat », nommée **MyDatabaseContract**, qui servira de récipient pour stocker toutes les constantes utiles à la gestion de votre base, ainsi que la structure de votre base. A la différence de ce qui est expliqué sur le site developer.android.com, vous

stockerez en variable globale de cette classe (**public static final**) :

- le nom de votre base de données : DATABASE_NAME
- son numéro de version : DATABASE_VERSION

Puis vous définirez pour la table **matable** (et plus généralement pour chaque table de la base de données associée à votre classe « contrat ») , une sous-classe abstraite (**public static abstract class**), nommée **MaTable** qui implémentera l'interface `BaseColumns`. En spécifiant que cette classe implémente `BaseColumns`, vous pourrez rajouter lors de la construction de la chaîne de caractères associée à la création de la table (cf. ci après les constantes à implanter), une colonne nommée **_ID** à votre table qui sera automatiquement gérée comme une clé primaire par les méthodes Java intervenant dans la gestion des bases de données. De plus, il est recommandé par le site de développement d'Android d'utiliser ce champ lorsque l'on manipule des bases de données, notamment lorsqu'elles sont intégrées dans un fournisseur de contenu (Content Provider).

La classe abstraite associée à la table **matable** implémentera les constantes suivantes :

- TABLE_NAME : le nom de vote table,
- COLUMN_NAME_ENTRY_INT : le nom de la première colonne,
- COLUMN_NAME_TITLE : le nom de la deuxième colonne,
- SQL_CREATE_ENTRIES : la chaîne de caractères contenant la requête SQL permettant de créer votre table,
- SQL_DELETE_ENTRIES : la chaîne de caractères contenant la requête SQL permettant de détruire votre table.

Vous devez à présent créer une classe nommée **MyDBHelper** qui se chargera de créer votre base de données. Pour cela , votre classe devra être une extension de la classe **SQLiteOpenHelper** (**public class MyDBHelper extends SQLiteOpenHelper**).

Ainsi pour accéder à votre base dans votre application principale, il vous suffira de créer un objet de type `MyDBHelper`.

```
MyDBHelper mDbHelper;
mDbHelper = new MyDBHelper(this);
```

Inspirez-vous des sections **Put Information into a Database** et **Read Information from a Database** sur le site **developer.android.com** pour insérer ou extraire des informations depuis votre base. Vous allez notamment devoir manipuler un objet de type **Cursor** pour parcourir le résultat d'une requête SQL. A l'aide des deux méthodes **moveToFirst()**, **moveToNext()**, et d'une simple boucle, vous pourrez facilement afficher les résultats d'une requête SQL à partir de cet objet.

Pour le bon déroulement de votre application, vous devrez respecter la démarche suivante :

- Au lancement de l'application, il faut récupérer la valeur du dernier numéro inséré dans la base. Si la base est vide, il faut initialiser ce numéro à 0.
- Lorsque l'utilisateur clique sur « SAVE », le contenu de la zone de texte et le numéro courant doivent être insérés dans la base. Le numéro est alors incrémenté pour la prochaine insertion.
- Lorsque l'utilisateur clique sur « SHOW », la base de données est parcourue dans l'ordre décroissant des numéros et les informations suivantes sont affichées :
  - valeur du champ _ID,
  - valeur du numéro,
  - valeur du texte.