CS493

Regis University Senior Capstone

Week Six

Intructor: Mr. Allan Rossi

Author: Mr. Jason N Mansfield

The Object Based Study Application

August 10, 2013

Contents

August 10, 2013

1 The need for a Second Database

I have reached a point in this project where I feel far more comfortable with the Android API. I have been able to grasp a few of the basic concepts related to the SQLite database, Fragments, Adapters, Cursors and other interesting elements. After I finally achieved adding and removing data to the database while updating the ListView (*ListView* 2013), I suddenly realized I would need a second database to expand an additional branch of content. Therefore, I now have two SQLite databases which are now functional. I am still trying to decide how to approach the final view which is to offer the user a look at their stored information so to speak. As it stands now the user creates a Object and associated description. This information is stored in the first database created (*Saving Data in SQL Databases* 2013):

```
public class DatabaseH extends SQLiteOpenHelper{

private SQLiteDatabase sqLiteDatabase;

private static final int DATABASE_VER = 1;
private static final String DATABASE_NAME = "learning";
//table columns

public static final class KEYS implements BaseColumns{
private KEYS() {}
public static final String TABLE_NAME = "objects";
public static final String KEY_ID = "_id";
public static final String KEY_OBNAME = "objectName";
public static final String KEY_CONTENT = "content";
public DatabaseH(Context context){
super(context, DATABASE_NAME, null, DATABASE_VER);
}
//Creating Tables
@Override
public void onCreate(SQLiteDatabase db){
db.execSQL("DROP TABLE IF EXISTS " + KEYS.TABLE_NAME);
String CREATE_OBJECT_TABLE = "CREATE TABLE " + KEYS.TABLE_NAME + "("
+ KEYS.KEY_ID + " INTEGER PRIMARY KEY," + KEYS.KEY_OBNAME + " TEXT,"
+ KEYS.KEY_CONTENT+ " TEXT" + ")";
db.execSQL(CREATE_OBJECT_TABLE);
}
}
```

Similarly, the newer database is designed to hold an id, title of sorts and then related content:

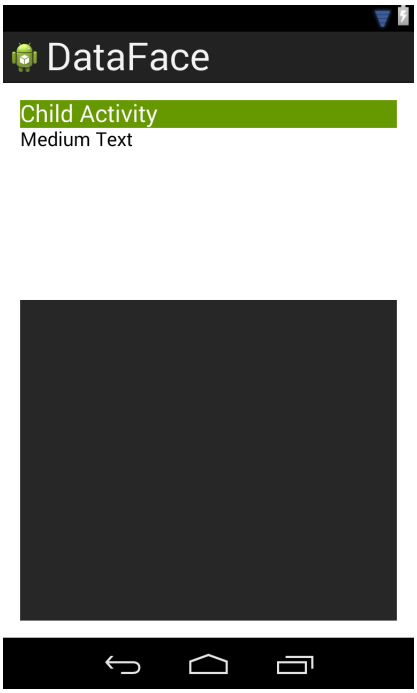
```
public class DatabaseTwo extends SQLiteOpenHelper
{
private SQLiteDatabase sqLiteDatabase;

private static final int DATABASE_VER = 1;
private static final String DATABASE_NAME = "griddatabase";
//table columns

public static final class CUBES implements BaseColumns {
private CUBES() {}
public static final String TABLE_NAME = "gridtable";
public static final String KEY_ID = "_id";
public static final String KEY_OBNAME = "cubename";
public static final String KEY_CONTENT = "cube";
}
public DatabaseTwo(Context context){
super(context, DATABASE_NAME, null, DATABASE_VER);
}
//Creating Tables
@Override
public void onCreate(SQLiteDatabase db){
db.execSQL("DROP TABLE IF EXISTS " + CUBES.TABLE_NAME);
String CREATE_OBJECT_TABLE = "CREATE TABLE " + CUBES.TABLE_NAME + "("
+ CUBES.KEY_ID + " INTEGER PRIMARY KEY," + CUBES.KEY_OBNAME + " TEXT,"
+ CUBES.KEY_CONTENT+ " TEXT" + ")";
db.execSQL(CREATE_OBJECT_TABLE);
}
```

2 Some exploration of UI possibilities

Having accomplished a ListView and arriving closer to the End Game, for this particular project, the need for aesthetics is becoming a issue. The current application is not entirely hideous, but it is not pleasing to the eye by any stretch of the imagination. I attempted to create a GridView (*GridView* 2013) at first but it appeared to make the current layout more cluttered then anything. Had I taken a different approach earlier it may have been useful? The final look of the application will unlikely be memorable but at this point, with the amount of time I have remaining, I believe I will shoot for functional and clean. The following image shows a child of the object with its title and description. On top is a Fragment which may or may not be used in the final product to contain a second ListView.



3 An attempt to use a fragment

I have had more success this week with using Fragments (*Building a Dynamic UI with Fragments* 2013) and I feel its time to try and implement one for the final View. This will allow me to expand this application in the future as I can simply swap out Fragments as opposed to completely re-writing an entire Activity. Fragments can make matters more complex, but in many ways they appear to add enough scalability that they are a more than fair trade for running an additional Activity:

```
<FrameLayout android:name="com.beta.dataface.ObjectFragment"
android:id="@+id/fragment_container"
android:layout_alignParentBottom="true"
android:layout_width="match_parent"
android:layout_height="300dp"
android:background="#D8000000"
tools:layout="@layout/child_activity"/>
```

My only regret is that I do not have more time to expand this particular application to show for my senior capstone. Although, my intentions are to continue working on this particular application until its fully functional. My goal is to have the basic needs met by the middle of next week, such as data entry for the user to both databases, and an understandable TextView for them to later review this data. The deletion feature and first ListView are already accomplished.

Bibliography

Building a Dynamic UI with Fragments (2013). Android Open Source Project. URL: http://developer.android.com/training/basics/fragments/index.html.

GridView (2013). Android Open Source Project. URL: http://developer.android.com/reference/android/widget/GridView.html.

ListView (2013). Android Open Source Project. URL: http://developer.android.com/guide/topics/ui/layout/listview.html#Example.

Saving Data in SQL Databases (2013). Android Open Source Project. URL: http://developer.android.com/training/basics/data-storage/databases.html.