

现代操作系统应用开发实验报告

学号：15331242

班级：周四早上班

姓名：明友芬

实验名称：SQLite Database & File Management

一．参考资料

- (1) 老师上课的 PPT；
- (2) <http://www.cnblogs.com/yanxiaodi/p/4941312.html>
- (3) <https://docs.microsoft.com/en-us/windows/uwp/files/>
- (4) <http://www.cnblogs.com/h82258652/p/4802076.html>
- (5) <https://github.com/Windows-Readiness/WinDevCamp>

二．实验步骤

- (1) 安装需的应用包或者软件：安装 *SQLite for Universal Windows Platform*；添加 *SQLite* 引导；搜索 *SQLitePCL* 并安装；下载并安装可视化工具；
- (2) 在 *MainPage* 的 *xaml* 文件里面添加文本框可 *search* 按钮，调整到合适的位置；
- (3) 在 *Todo* 项目下新建一个文件夹 *Services*，添加 *DbContext.cs* 文件到 *Services* 文件夹中，在 *DbContext.cs* 里面实现数据库的创建，及对数据库的增删改查的操作；
- (4) 在 *TodoItem* 里面根据在 *DbContext.cs* 里面所做的内容更改 *TodoItem* 里面的代码，在加上一些由于更新而改变的处理方法；
- (5) 在 *ViewModels* 里面实现对于 *TodoListItem* 的更改的同时对数据库内容同时更改的内容的实现；
- (6) 在主函数里面加入点击 *search* 按钮之后所对应的操作，并且在 *MainPage* 的初始化函数里面增加数据库创建的代码内容，相对应的增改操作对于数据库内容的同步实现等；
- (7) 在 *NewPage* 里面对应数据库的内容更改增删的函数；

三．实验结果截图

- (1) 向 *Todo* 表增加一个 *item*，打开可视化工具可以看见在可视化工具里面刷新之后也增加了一个字段，最开始的状态截图如图 1 所示：

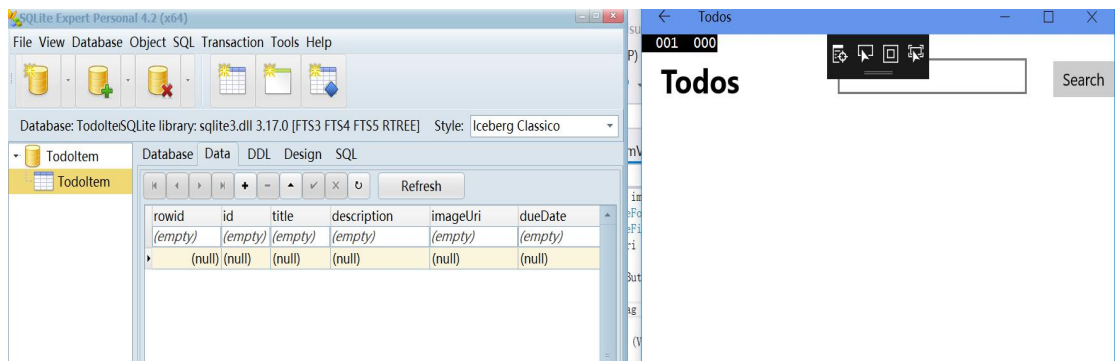


图 1

向 Todo 增加一个 item 之后再数据库中刷新之后也对应的增加了一个字段，截图如图 2 所示：

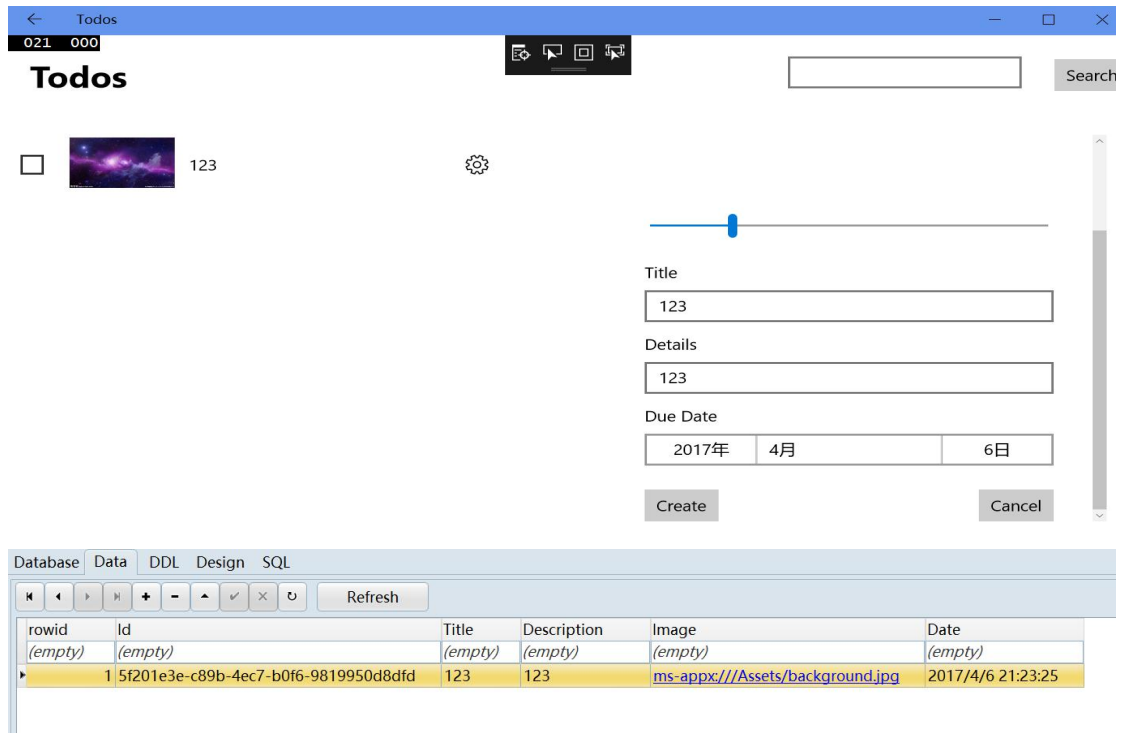
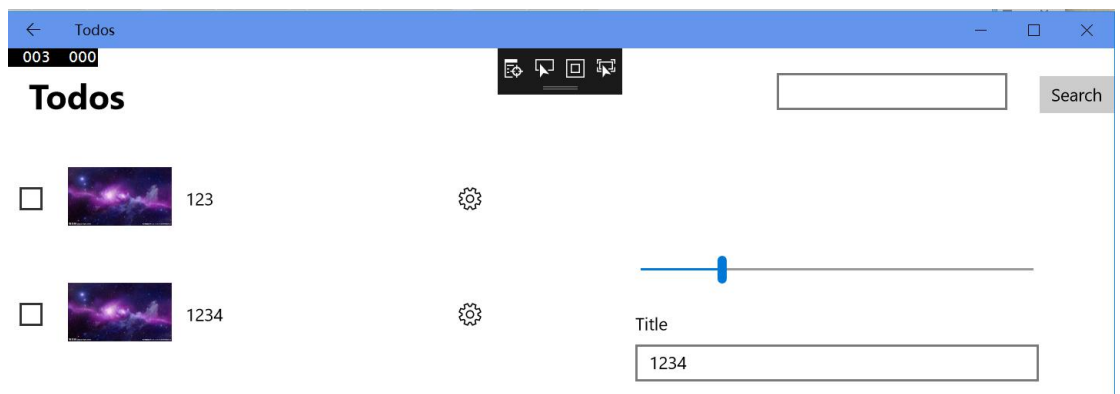


图 2

(2) 实现 item 的删除工作，同时在数据库中的内容也被删除掉，在删除之前的内容如图 3 所示：



rowid	Id	Title	Description	Image	Date
(empty)	(empty)	(empty)	(empty)	(empty)	(empty)
1	5f201e3e-c89b-4ec7-b0f6-9819950d8dfd	123	123	ms-appx:///Assets/background.jpg	2017/4/6 21:23:25
2	3bb5d8f5-3416-4958-85bf-379c6a6ed016	1234	123	ms-appx:///Assets/background.jpg	2017/4/6 21:23:25



图 3

当我们删除第一个 item 123 之后，在数据库中同时也会删除掉 123Title 为 123 的这个字段，如图 4 所示：

rowid	Id	Title	Description	Image	Date
(empty)	(empty)	(empty)	(empty)	(empty)	(empty)
2	3bb5d8f5-3416-4958-85bf-379c6a6ed016	1234	123	ms-appx:///Assets/background.jpg	2017/4/6 21:23:25

006 000

Todos

☐  1234
 






图 4

(3) 将 item 的标题更改为 12345，并且 details 更改为 12 之后，可以发现在数据库中的内容也随着改变了实验截图如图 5 所示：

rowid	Id	Title	Description	Image	Date
(empty)	(empty)	(empty)	(empty)	(empty)	(empty)
2	3bb5d8f5-3416-4958-85bf-379c6a6ed016	1234	123	ms-appx:///Assets/background.jpg	2017/4/6 21:23:25

006 000

Todos

☐  1234
 




图 5

(4) 在左上角的文本框内输入标题查找需要的 item 可以得到查询结果，截图如图 6 所示：

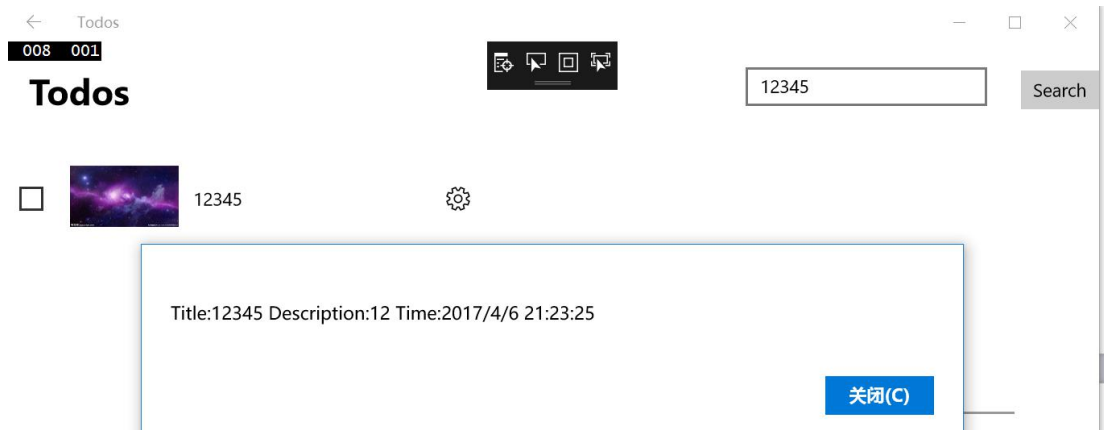


图 6

如果没有查找到对应的内容的话就输出的内容为 “No Such Item!” 截图如图 7 所示：

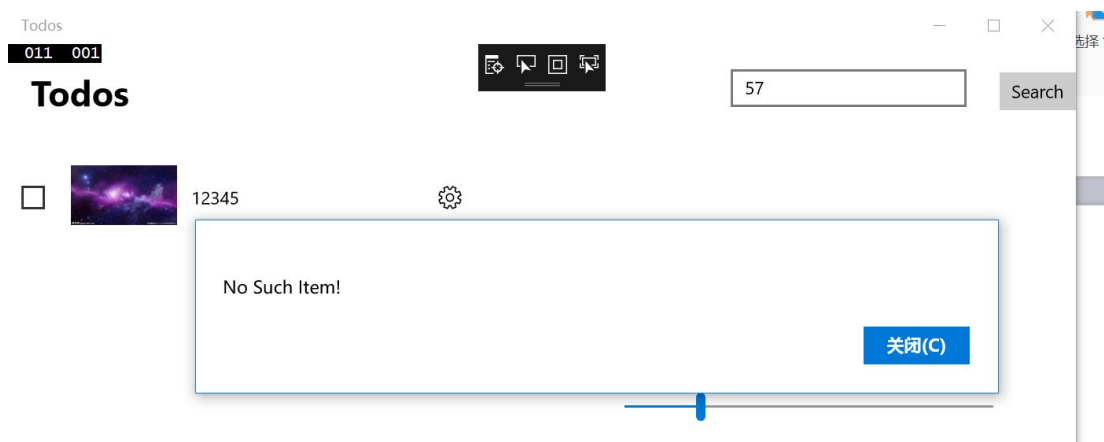
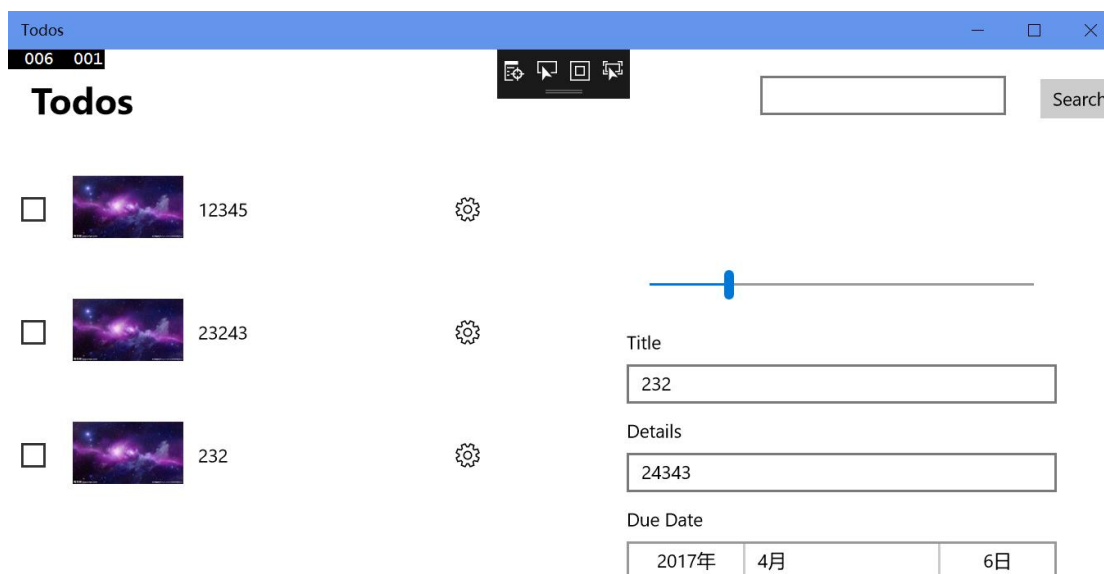


图 7

- (5) 数据库保存以及应用状态的恢复，在我们想 Todos 添加了一些 item 之后，关闭程序，关闭第三方可视化工具，再次打开程序会发现在我们的屏幕上显示的是我们关闭之前在数据库中保存的内容，关闭程序之前 Todo 的列表和数据库内容的截图如图 8 所示：



Database Data DDL Design SQL						
Refresh						
rowid	Id	Title	Description	Image	Date	
(empty)	(empty)	(empty)	(empty)	(empty)	(empty)	
2	3bb5d8f5-3416-4958-85bf-379c6a6ed016	12345	12	ms-appx:///Assets/background.jpg	2017/4/6 21:23:25	
3	6162ecec-c534-4fcd-84c2-255a3e48356c	23243	243	ms-appx:///Assets/background.jpg	2017/4/6 21:38:15	
4	6dc6fd64-2475-4740-a208-c8f5a8f8a807	232	24343	ms-appx:///Assets/background.jpg	2017/4/6 21:38:15	

图 8

关闭之后再次打开的内容如图 9 所示：

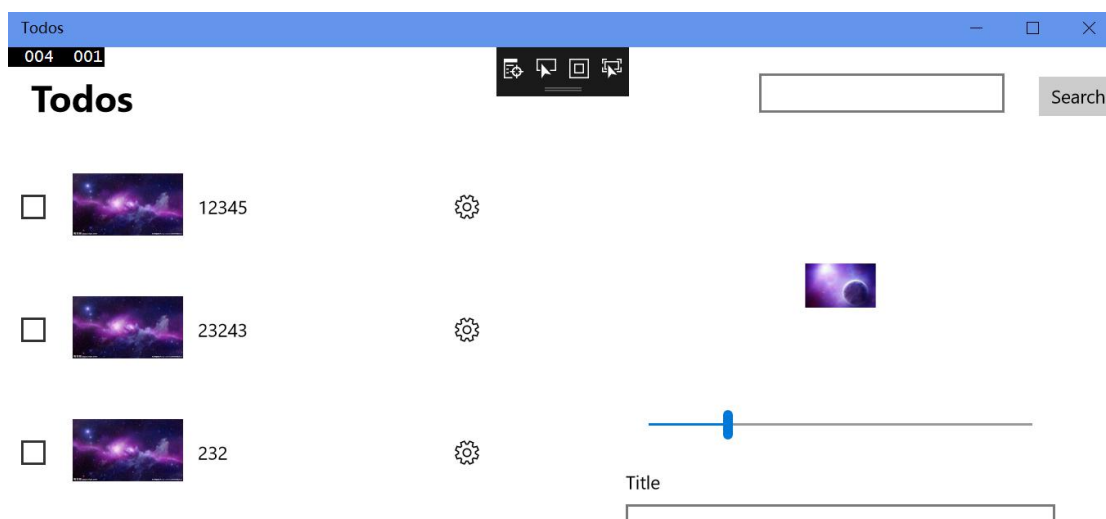


图 9

- (6) 第三方工具的使用，在上述的实验截图中已经多次使用了第三方工具来查看数据在数据库中的变化情况，以及利用这个工具来恢复程序状态的情况。
- (7) 图片的存取，每个表项的图片的存取状态截图如下图 10 所示：

图片的缺省保存状态为：

```
public static string defaultImagePath = "ms-appx:///Assets/background.jpg";
```

Image
(empty)
ms-appx:///Assets/background.jpg
ms-appx:///Assets/background.jpg
ms-appx:///Assets/background.jpg

图 10

四 . 实验过程遇到的问题

本次实验过程中遇到的问题不难，但是困扰了好一段时间，就是在写完程序之后，程序运行不了，并且报错内容大概就是无法在程序中找到我创建的数据库 *Sqlite*，最开始的时候一直认为是数据库的创建没有创建好，导致后面调用的时候找不到，但是后来发现并不是这样，经过一番检查才知道犯了一个十分低级的错误在 *MainPage* 的构造函数里面没有调用创建数据库的语句导致后面的调用失败的；其次就是在每一次更改 *item* 的值的时候，对应的数据库的值也要改变，

但是在做的过程中很容易在修改的内容里面忘记修改，导致在 *todos* 里面修改了，但是在 数据库里面没有实现对应的更改。

五 . 思考与总结

- (1) LocalFolder 的作用：获取本地应用程序数据存储区中的根文件夹，属性值是 StorageFolder，使用 "ms-appdata:///local/" 协议，可以访问本地应用程序数据存储区的文件。
- (2) StringBuilder 的作用：String 对象是不可改变的。每次使用 System.String 类中的方法之一时，都要在内存中创建一个新的字符串对象，这就需要为该新对象分配新的空间。在需要对字符串执行重复修改的情况下，与创建新的 String 对象相关的系统开销可能会非常昂贵。如果要修改字符串而不创建新的对象，则可以使用 System.Text.StringBuilder 类。例如，当在一个循环中将许多字符串连接在一起时，使用 StringBuilder 类可以提升性能。
- (3) 本次实验学会了将 c#代码与数据库内容联系起来，虽然实现的过程中遇到了不少问题，但是也确实学到了很多东西。