

现代操作系统应用开发实验报告

学号：153311117

班级：教务 2 班

姓名：黄楠绚

实验名称：homework6

一．参考资料

请在这里列出对本实验有帮助你所参考的资料或者网站。

<https://docs.microsoft.com/zh-cn/windows/uwp/files/file-access-permissions>

二．实验步骤

1.由于需要用到数据库，因此专门创建一个类进行数据库各种操作：

```
class SqlData
{
    private static string databaseName = "TodoItem.db";
    public SqlData()
    {
        var conn = getSqliteConn();
        using (var statement = conn.Prepare(SqlStatement.createDatabase))
        {
            statement.Step();
        }
    }
}
```

为了方便 debug 和理解代码，写了一个类放 sql 语句：

```
public static class SqlStatement
{
    public static string createDatabase = @"CREATE TABLE IF NOT EXISTS
        TodoItem (id          STRING PRIMARY KEY NOT NULL,
                  title       VARCHAR(140) NOT NULL,
                  description  TEXT,
                  imageUrl     VARCHAR(300),
                  dueDate      VARCHAR(140));";

    public static string insertItem = @"INSERT INTO TodoItem (id, title, description, imageUrl, dueDate)
        VALUES (?, ?, ?, ?, ?);";

    public static string selectAllItems = @"SELECT id, title, description, imageUrl, dueDate FROM TodoItem";

    public static string updateItem = @"UPDATE TodoItem set title=?, description=?, imageUrl=?, dueDate=? WHERE id=?";

    public static string deleteItem = @"DELETE FROM TodoItem WHERE id=?";
}
```

运行程序时，创建并读取数据库中表格的数据，获得数据库中的 items：

```

public static ObservableCollection<Models.TODOItem> getAllToDoItem()
{
    ObservableCollection<Models.TODOItem> todoItemList = new ObservableCollection<Models.TODOItem>();
    var conn = getSqliteConn();
    var statement = conn.Prepare(SqlStatement.selectAllItems);
    while (statement.Step() == SQLiteResult.ROW)
    {
        todoItemList.Add(new Models.TODOItem((string)statement[0], (string)statement[1], (string)statement[2],
                                                (string)statement[3], (string)statement[4]));
    }
    return todoItemList;
}

```

创建新的 item 时，对数据库表格的数据进行 Insert:

```

public static bool insertItem(string id, string title, string description, string imageUri, string dueDate)
{
    var conn = getSqliteConn();
    try
    {
        using (var statement = conn.Prepare(SqlStatement.insertItem))
        {
            statement.Bind(1, id);
            statement.Bind(2, title);
            statement.Bind(3, description);
            statement.Bind(4, imageUri);
            statement.Bind(5, dueDate);
            statement.Step();
        }
    }
    catch (Exception)
    {
        return false;
    }
    return true;
}

```

更新 item 时:

```

public static void updateItemById(string id, string title, string description, string imageUri, string dueDate)
{
    var conn = getSqliteConn();
    using (var statement = conn.Prepare(SqlStatement.updateItem))
    {
        statement.Bind(1, title);
        statement.Bind(2, description);
        statement.Bind(3, imageUri);
        statement.Bind(4, dueDate);
        statement.Bind(5, id);
        statement.Step();
    }
}

```

删除 item 时:

```

public static void deleteItem(string id)
{
    var conn = getSqliteConn();
    using (var statement = conn.Prepare(SqlStatement.deleteItem))
    {
        statement.Bind(1, id);
        statement.Step();
    }
}

```

2.关于表项图片的保存:

如果是图片是默认选择的背景图片:

```

if (imageUriString == defalutImagePath) // 如果是默认背景图片
{
    this.bitmapImage = new BitmapImage(this.imageUri);
}

```

如果是用户自行选择的图片:

在用文件选取器选择文件后，用户选择的文件复制一份到 LocalFolder，再从 localFolder 读取：

```
if (imageFile != null)
{
    string imageName = imageFile.Name;
    StorageFolder localFolder = ApplicationData.Current.LocalFolder;
    StorageFile newImageFile = await imageFile.CopyAsync(localFolder, imageName, NameCollisionOption.ReplaceExisting);
    imageUri = new Uri(newImageFile.Path);
}
```

保存 item 的图片进数据库的方式是保存它的 uri，在程序重新启动时重新获得 item 的图片的方式是：

```
// 用户自行选择的图片，从路径获得图片
private void stringPathToBitmapImage(string path)
{
    BitmapImage image = new BitmapImage(new Uri(path));
    this.bitMapImage = image;
}
```

因为和上面的默认背景图片一样都是放在应用自己的文件中，因此不需要考虑文件权限的问题

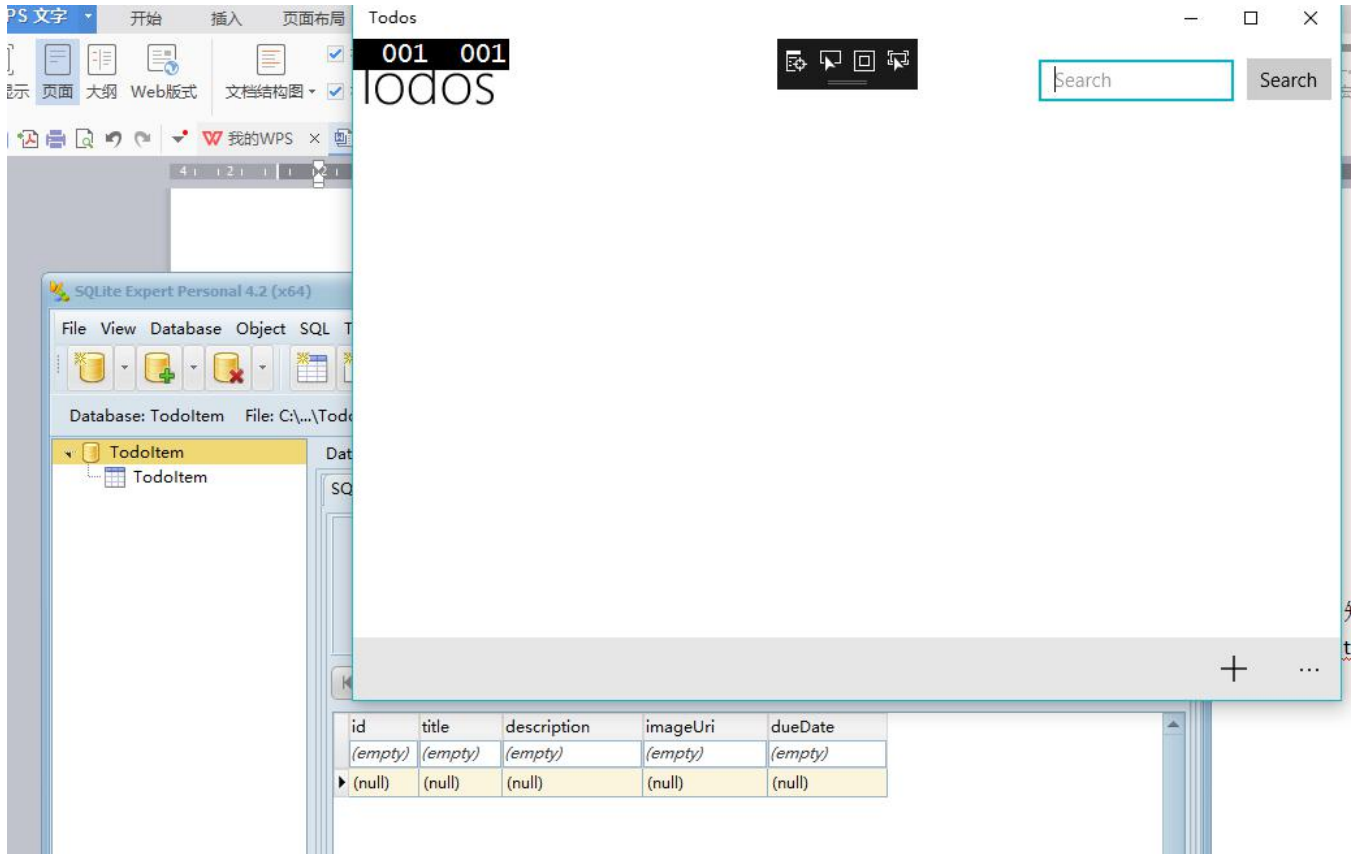
3.搜索 item:

```
private async void SearchButton_Click(object sender, RoutedEventArgs e)
{
    StringBuilder sb = new StringBuilder();
    foreach (var item in this.ViewModel.AllItems)
    {
        if (item.title.Contains(SearchText.Text))
        {
            sb.AppendLine("Title: " + item.title + "Description: " + item.description + "Time: " + item.dueDate.ToString());
        }
    }
    var i = await new MessageDialog(sb.ToString()).ShowAsync();
}
```

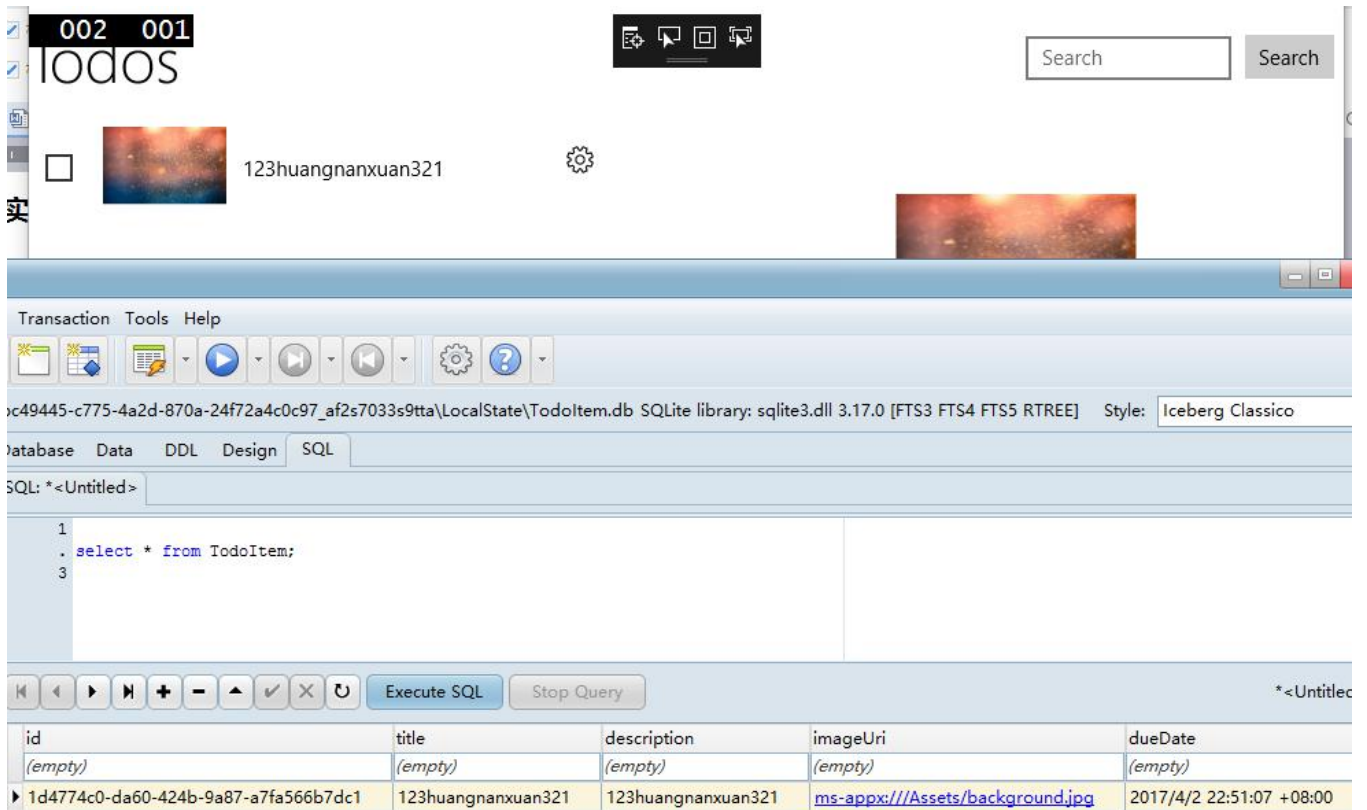
三．实验结果截图

请在这里把实验所得的运行结果截图。

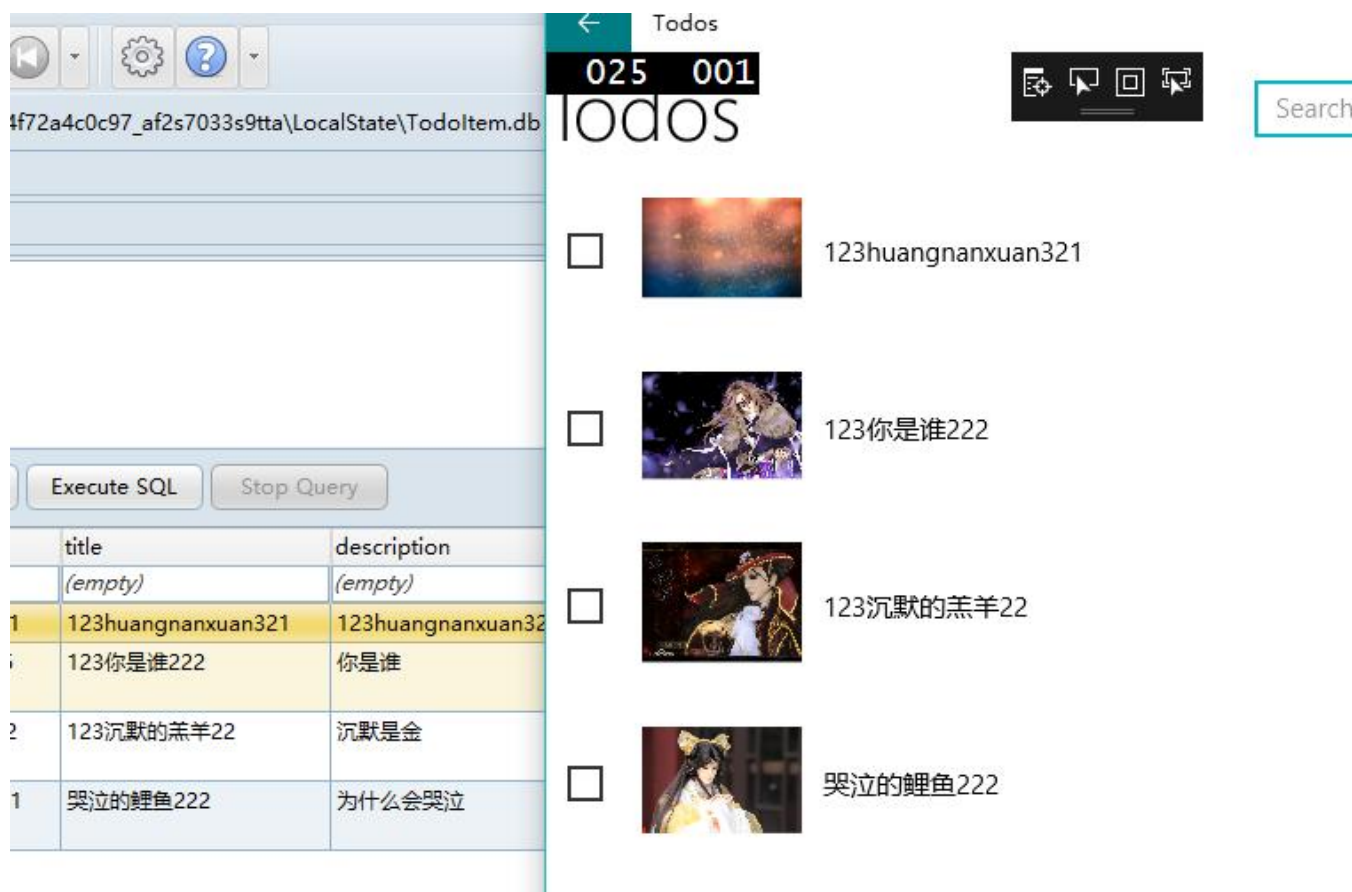
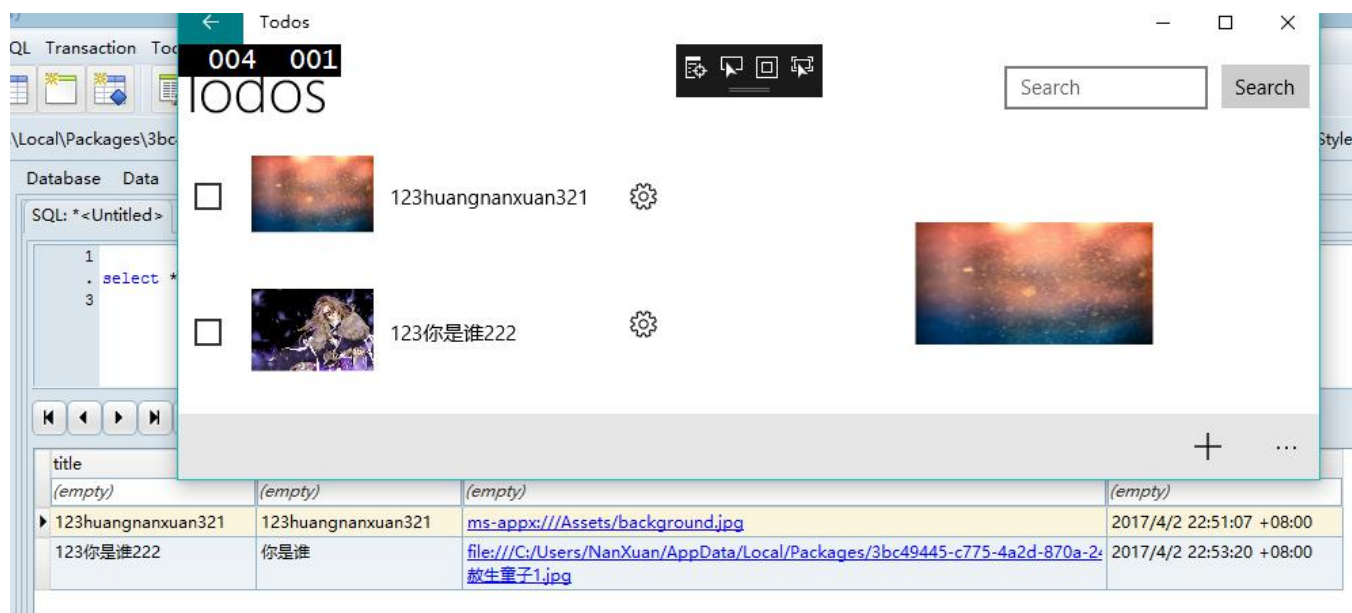
1. 使用了第三方工具查看数据库内容，刚启动程序时，默认空白，数据库也为空：



2. 在 MainPage 创建 item，数据库也相应的增加了一条记录：




3. 在 NewPage 新建 item:



4. 在 MainPage 中 Update 第 3 个 item:


003 012

iodos

☐



123huangnaxuan321

⚙️

☐



123你是谁222

⚙️

☐


123我不是沉默的羔羊233

⚙️

☐


哭泣的鲤鱼222

⚙️

Execute SQL

Stop Query

title	description	imageUri
(empty)	(empty)	(empty)
123huangnaxuan321	123huangnaxuan321	ms-a
123你是谁222	你是谁	file:///C:/Users/NanXuan/A
123我不是沉默的羔羊233	沉默是金，文静值几个钱	file:///C:/Users/NanXuan/A
哭泣的鲤鱼222	为什么会哭泣	file:///C:/Users/NanXuan/A

5. 在 NewPage 中 update 第一个 item:

001 002

iodos

☐


123弄无暇321

⚙️

☐


123你是谁222

⚙️

☐


123我不是沉默的羔羊233

⚙️

Execute SQL

Stop Query

title	description	imageUri
(empty)	(empty)	(empty)
123弄无暇321	123h快雪时晴321	file:///C:/Users/NanXuan/A
123你是谁222	你是谁	file:///C:/Users/NanXuan/A
123我不是沉默的羔羊233	沉默是金，文静值几个钱	file:///C:/Users/NanXuan/A
哭泣的鲤鱼222	为什么会哭泣	file:///C:/Users/NanXuan/A

6. 删除第 2 个 item:

001 002

iodos

☐


123弄无暇321

⚙️

☐


123我不是沉默的羔羊233

⚙️

☐


哭泣的鲤鱼222

⚙️

Execute SQL

Stop Query

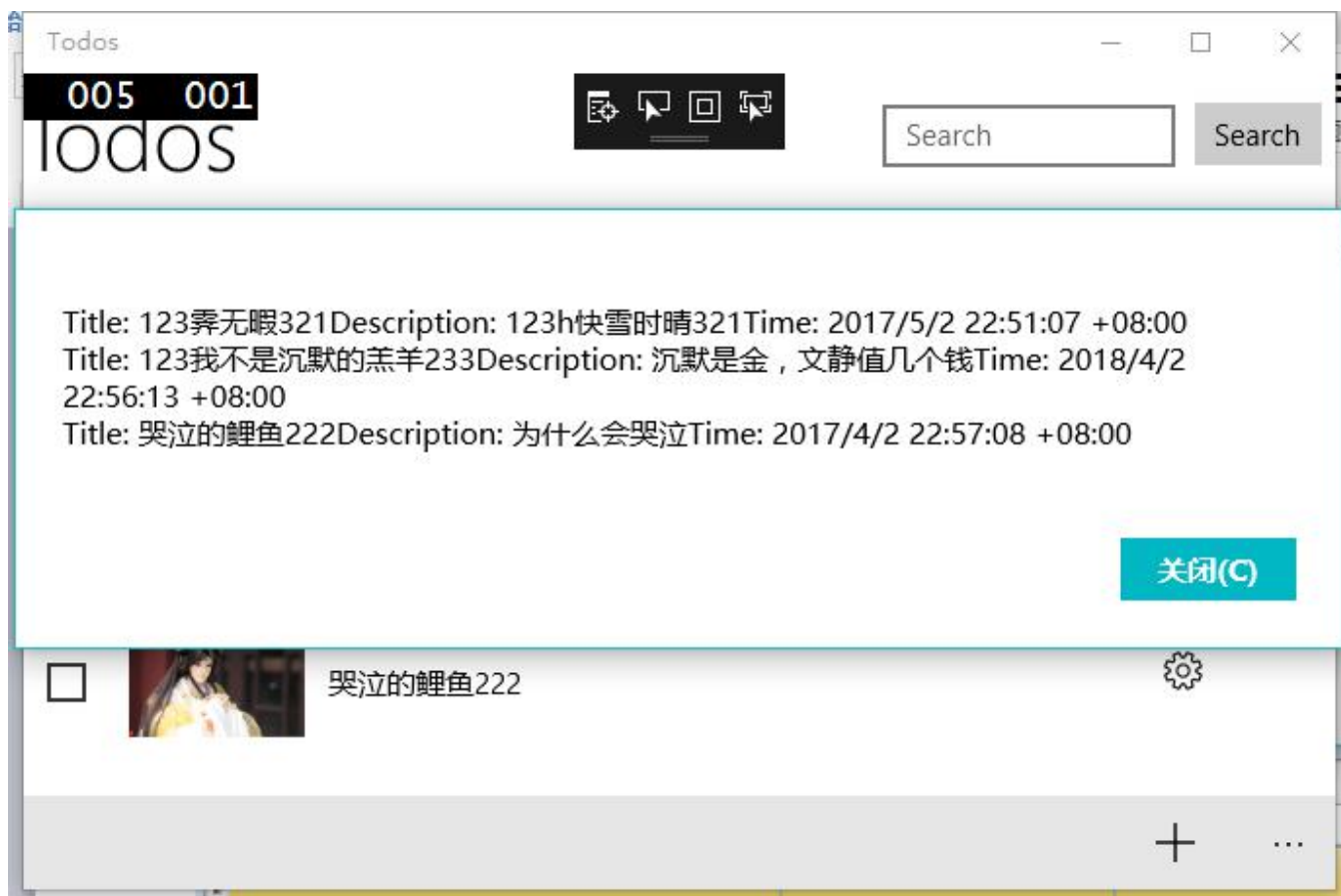
title	description	imageUri
(empty)	(empty)	(empty)
123弄无暇321	123h快雪时晴321	file:///C:/Users/NanXuan/A
123我不是沉默的羔羊233	沉默是金，文静值几个钱	file:///C:/Users/NanXuan/A
哭泣的鲤鱼222	为什么会哭泣	file:///C:/Users/NanXuan/A

7. 关掉程序，再重新启动，数据库数据完整不变，程序启动后也能恢复数据：

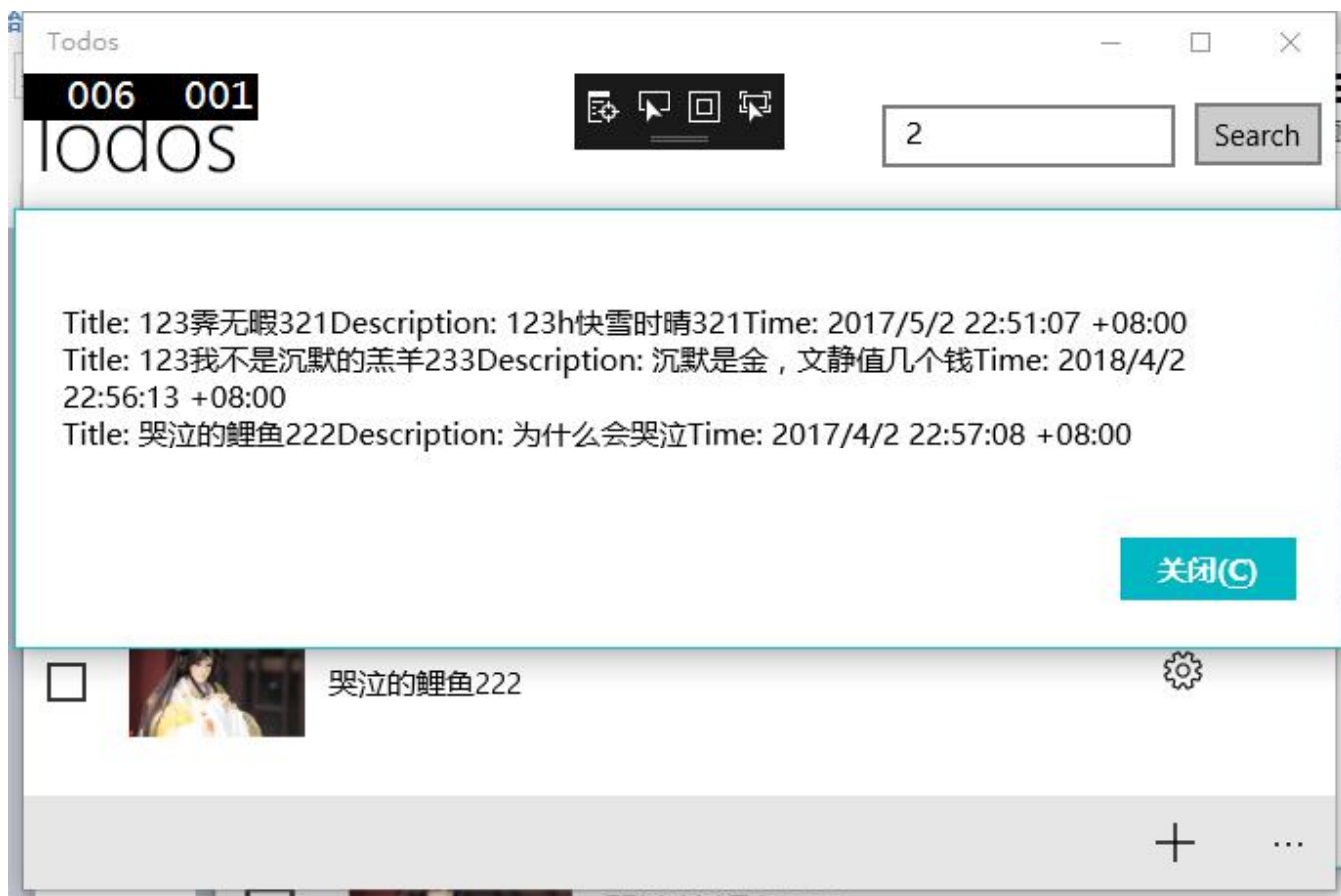


8. 搜索关键字:

空字符串匹配全部:



输入"2":



输入“3”：

Todos

006 001

iodos

3

Search

Title: 123弄无暇321Description: 123h快雪时晴321Time: 2017/5/2 22:51:07 +08:00

Title: 123我不是沉默的羔羊233Description: 沉默是金，文静值几个钱Time: 2018/4/2 22:56:13 +08:00

关闭(C)

☐



哭泣的鲤鱼222



+

...

输入汉字“哭”：

Todos

003 001

iodos

哭

Search

Title: 哭泣的鲤鱼222Description: 为什么会哭泣Time: 2017/4/2 22:57:08 +08:00

关闭(C)

☐



哭泣的鲤鱼222



+

...

输入“123 雾无暇 321”，完全匹配：



输入“啊”，不匹配任何 item：



四．实验过程遇到的问题

请在这里写下你在实验过程中遇到的问题以及解决方案。

1. 在获取表格中的数据 :select * from TodoItem 时 ,只能得到表格的第一行数据 ,一开始不知道怎么做 ,搜索无果后 , 自己自行做一些实验才发现原来在 select 时 statement.Step()一次只能获取一条数据因此把代码写成循环 :

```
var statement = conn.Prepare(@"select * from TodoItem");
while (statement.Step() == SQLiteResult.ROW)
{
    todoItemList.Add(new Models.TODOItem((string)statement[0], (string)statement[1], (string)statement[2],
                                           (string)statement[3], (string)statement[4]));
}
```

就成功获得了我想要的数据。

2. 关于文件访问权限,一开始直接不知道原来的方式会出现文件权限问题。后来问题出现了后知道 ta 为什么要把微软官网上关于文件读取的文档链接放在 ppt 上,知道问题看了一下文档了解了 LocalData 后就解决问题啦

五 . 思考与总结

请在这里写下你本次试验的心得体会以及所思所想。

LocalFolder RoamingFolder 等存储位置的作用

LocalFolder: 本地文件夹,每个应用都有一个文件夹,一般程序只能访问自己文件夹,对自己的文件夹中的文件具有读写的权限,也可以存放本地数据库

RoamingFolder: 漫游文件夹,如果用户在一台设备上安装应用后又在另一台设备上进行了安装,第一台设备上设置的所有首选项将自动应用到第二台设备上的应用, RoamingFolder 此时所存放的就是漫游数据

TemporaryFolder: 临时文件夹,转储临时文件的地方,目的在于临时调用与写入

StringBuilder 的作用

Stringbuilder 相对与 String 不可变类型,是可变的字符串;当需要对字符串进行大量变更修改等操作的时候, StringBuilder 比起 String 有助于提高效能。

心得体会以及所思所想

这次作业让我可以更进一步理解手机里面 app 的文件,原来 app 是无法访问用户的文件,让我理解了为什么在网易云音乐软件中更换歌单封面,网易的 app 文件就会多出一张图,原来是需要把用户图片 copy 过来后才能使用

1. 实验报告提交格式为 pdf。
2. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。