

现代操作系统应用开发实验报告

学号： 14970011

班级： 2015 级教务 2 班

姓名： 宋思亭

实验名称： homework6

一 . 参考资料

Windows UWP Namespaces : <https://docs.microsoft.com/en-us/uwp/api/>

StringBuilder Class :

<https://docs.microsoft.com/en-us/dotnet/core/api/system.text.stringbuilder>

课件、Demo

二 . 实验步骤

1. 阅读作业需求和课件 PPT
2. 在App.xaml.cs中创建数据库Todos.db , 包括ID , date , imgname , title , details , finish项 , 并在项目初始化时调用

```
public static SQLiteConnection conn;
private void LoadDatabase()
{
    // Get a reference to the SQLite database
    conn = new SQLiteConnection("Todos.db", SQLiteOpen.READWRITE);
    string sql = @"CREATE TABLE IF NOT EXISTS [Todo] (
        [ID] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
        [date] VARCHAR(30),
        [imgname] VARCHAR(140),
        [title] VARCHAR(140),
        [details] VARCHAR(140),
        [finish] VARCHAR(10)
    );";
    using (var statement = conn.Prepare(sql)) { statement.Step(); }
}
```

3. 在MainPage中添加搜索框，cs文件中添加搜索事件

```
<AutoSuggestBox x:Name="search" HorizontalAlignment="Right" Width="200"
PlaceholderText="Search....." IsSuggestionListOpen="False"
QuerySubmitted="search_QuerySubmitted" QueryIcon="Find" Margin="0,48,10,0" />

private async void search_QuerySubmitted(AutoSuggestBox sender,
AutoSuggestBoxQuerySubmittedEventArgs args)
{
    var text = args.QueryText.Trim();
    if (text == "")
        return;
    string alert = "";
    try
    {
        var sql = "SELECT date, title, details FROM Todo WHERE date LIKE ? OR title
LIKE ? OR details LIKE ?";
        using (var statement = conn.Prepare(sql))
        {
            statement.Bind(1, "%" + text + "%");
            statement.Bind(2, "%" + text + "%");
            statement.Bind(3, "%" + text + "%");
            while (SQLiteResult.ROW == statement.Step())
            {
                var date = statement[0].ToString();
                date = date.Substring(0, date.IndexOf(' '));
                string title = statement[1] as string;
                string details = statement[2] as string;
                alert += "Title: " + title + ";\nDetails: " + details + ";\nDue Date:
" + statement[0].ToString() + "\n\n";
            }
            if (alert == "")
                alert = "No result!\n";
            await new MessageDialog(alert).ShowAsync();
        }
    }
    catch (Exception err)
    {
        Debug.WriteLine(err.ToString());
    }
}
```

4. 点击完成事件，修改finish项内容，取消完成事件同理

```
try
```

```

{
    var dc = (sender as FrameworkElement).DataContext;
    var listitem = ToDoListView.ContainerFromItem(dc) as ListViewItem;
    var item = listitem.Content as TodoItem;
    string sql = @"UPDATE Todo SET finish = ? WHERE ID = ?";
    using (var res = conn.Prepare(sql))
    {
        res.Bind(1, "true");
        res.Bind(2, item.ID);
        res.Step();
    }
}
catch (Exception err)
{
    Debug.WriteLine(err.Message);
}

```

5. 创建事件

```

try
{
    string sql = @"INSERT INTO Todo (date, imgname, title, details,
finish) VALUES (?, ?, ?, ?, ?)";
    using (var res = conn.Prepare(sql))
    {
        res.Bind(1, date.Date.Date.ToString());
        res.Bind(2, Common.selectName);
        res.Bind(3, title.Text.Trim());
        res.Bind(4, details.Text.Trim());
        res.Bind(5, "false");
        res.Step();
        this.ViewModel.AddItem(conn.LastInsertRowId(),
date.Date.DateTime, Common.selectName, title.Text, details.Text, false);
        Common.selectName = "";
        CancelButton_Click(null, null);
        ViewModel.SelectedItem = null;
        await new MessageDialog("Create successfully!").ShowAsync();
    }
}
catch (Exception err)
{
    Debug.WriteLine(err.Message);
}

```

6. 修改事件

```

        try
        {
            string sql = @"UPDATE Todo SET date = ?, imgname = ?, title = ?,
details = ? WHERE ID = ?";
            using (var res = conn.Prepare(sql))
            {
                res.Bind(1, date.Date.Date.ToString());
                res.Bind(2, Common.selectName);
                res.Bind(3, title.Text.Trim());
                res.Bind(4, details.Text.Trim());
                res.Bind(5, ViewModel.SelectedItem.ID);
                res.Step();
                ViewModel.SelectedItem.UpdateItem(date.Date.DateTime,
Common.selectName, title.Text.Trim(), details.Text.Trim());
                await new MessageDialog("Update successfully!").ShowAsync();
            }
        }
        catch (Exception err)
        {
            Debug.WriteLine(err.Message);
        }
    }
}

```

7. 删除事件

```

        try
        {
            string sql = @"DELETE FROM Todo WHERE ID = ?";
            using (var res = conn.Prepare(sql))
            {
                res.Bind(1, ViewModel.SelectedItem.ID);
                res.Step();
                ViewModel.getItems.Remove(ViewModel.SelectedItem);
                ViewModel.SelectedItem = null;
                Common.selectName = "";
            }
        }
        catch (Exception err)
        {
            Debug.WriteLine(err.Message);
        }
    }
}

```

8. 调试项目

三 . 实验结果截图

1. 增加事件项（数据库及列表中最后一项为新增）

Form fields:

- Title: 334
- Details: 444
- Due Date: 2017年 4月 1日









Buttons: Create, Cancel

rowid	ID	date	imgname	title	details	finish
(empty)	(empty)	(empty)	(empty)	(empty)	(empty)	(empty)
1	1	2017/3/30 0:00:00		233	222	true
6	6	2017/3/30 0:00:00		22	33	false
8	8	2017/3/30 0:00:00	cake.JPG	123	456	false
9	9	2017/4/1 0:00:00	background.JPG	334	444	false

Todos

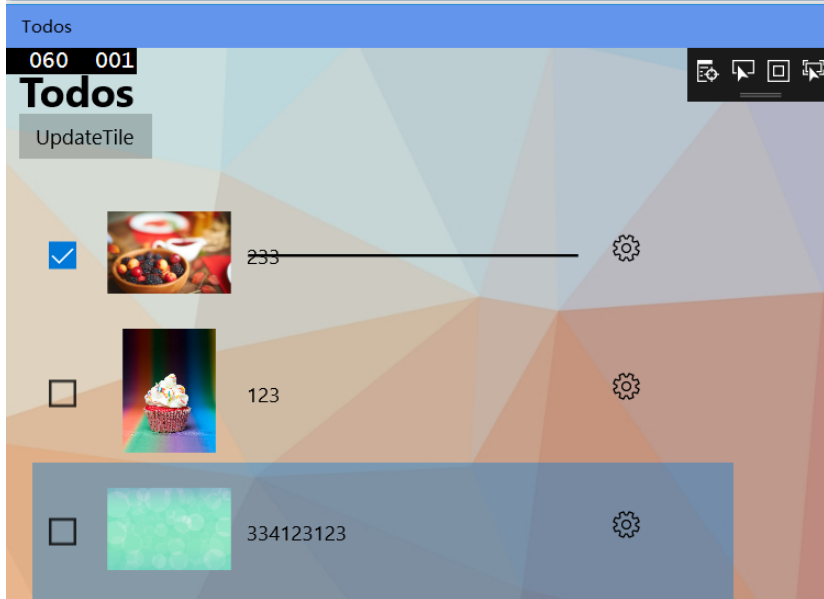
011 000

UpdateTile

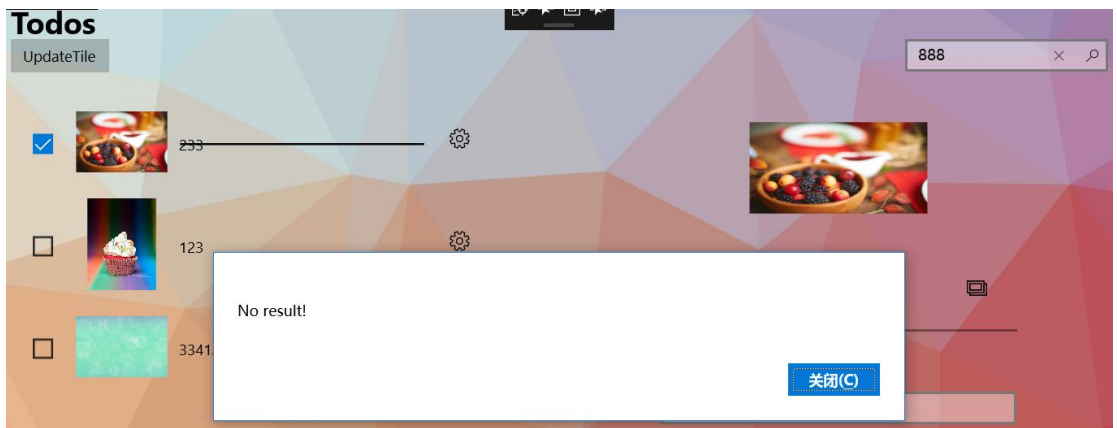
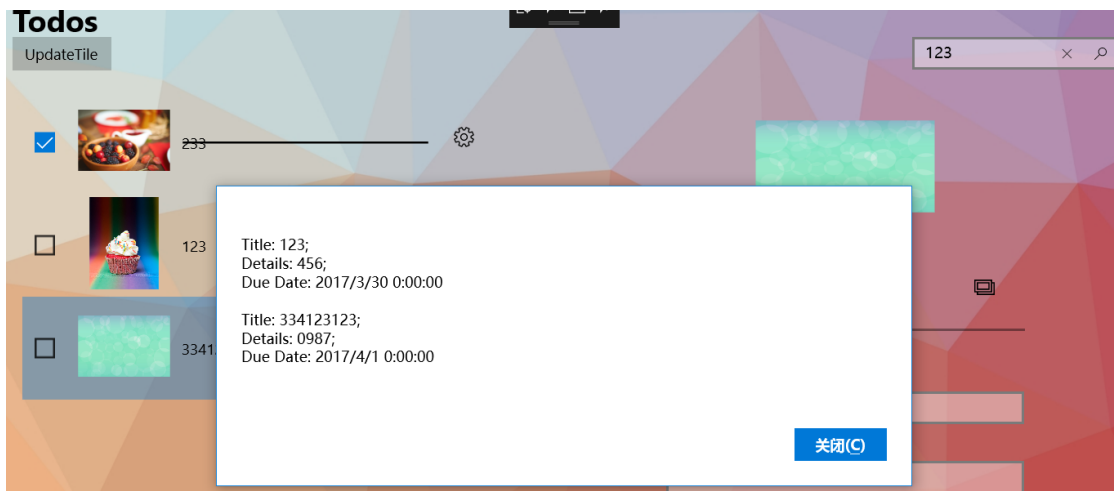
- ☒  233 
- ☐  22 
- ☐  123 
- ☐  334 

2. 删除数据项（上图第二项为删除项）

rowid	ID	date	imgname	title	details	finish
(empty)	(empty)	(empty)	(empty)	(empty)	(empty)	(empty)
1	1	2017/3/30 0:00:00		233	222	true
8	8	2017/3/30 0:00:00	cake.JPG	123	456	false
9	9	2017/4/1 0:00:00	background.JPG	334123123	0987	false



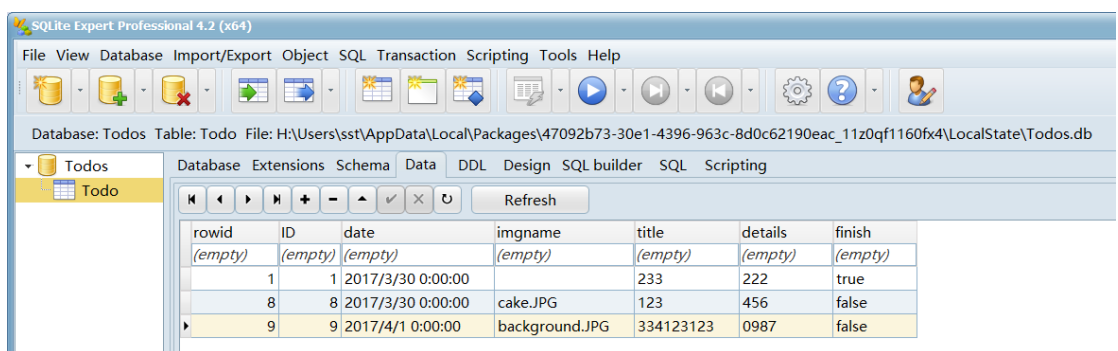
4. 查询数据（符合 123 的有两项，888 没有搜索结果）



5. 关闭窗口并重新打开，数据保持不变（实现图片存取）



6. 采用 SQLite Export 查看数据内容



四 . 实验过程遇到的问题

1. 修改事件时，如果图片没有改动，将不会保存原图片，修改设置后解决

五 . 思考与总结

1. 搜索字符串"%"意为查询所有数据，可以用这个方法读所有的数据。
2. RoamingFolder 意味着文件储存在 OneDrive 中，在其他联网的设备上也可以访问里面的数据，但每个应用程序的数据有 100K 的大小限制。
3. LocalFolder 意味着文件存储在本地，没有存储大小的限制，并且在应用程序更新时可以保留
4. StringBuilder 用于字符串匹配，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