

# 现代操作系统应用开发实验报告

学号： 15331418

班级： 晚上班

姓名： 郑柏川 实验名称： Database & File Management

## 一. 参考资料

参考的 ppt

## 二. 实验步骤

### a) 对 viewmodel 的设计

```
class TodoItemViewModel
{
    private NewObservableCollection<Models.TodoItem> allItems = new NewObservableCollection<Models.TodoItem>();
    public NewObservableCollection<Models.TodoItem> AllItems { get { return this.allItems; } }

    private Models.TodoItem selectedItem = default(Models.TodoItem);
    public Models.TodoItem SelectedItem { get { return selectedItem; } set { this.selectedItem = value; } }

    private SQLiteConnection db;

    public TodoItemViewModel() { ... }

    public async void init() { ... }
    public async Task
    AddTodoItem(string title, string description, DateTimeOffset date, Uri imageUrl) { ... }

    public void RemoveTodoItem(long id) { ... }

    public async Task
    UpdateTodoItem(long id, string title, string description, DateTimeOffset date, Uri imageUrl, bool completed) { ... }
    // 查
    public string FindTodoItem(string query) { ... }

    private bool ItemMatch(TodoItem item, string query) { ... }
}
```

细节，以 init 为例子：

```

class TodoItemViewModel
{
    private NewObservableCollection<Models.TodoItem> allItems = new NewObservableCollection<Models.TodoItem>();
    public NewObservableCollection<Models.TodoItem> AllItems { get { return this.allItems; } }

    private Models.TodoItem selectedItem = default(Models.TodoItem);
    public Models.TodoItem SelectedItem { get { return selectedItem; } set { this.selectedItem = value; } }

    private SQLiteConnection db;

    public TodoItemViewModel()
    {
        public async void init()
        {
            public async Task
            AddTodoItem(string title, string description, DateTimeOffset date, Uri imageUri)
            {
                public void RemoveTodoItem(long id)
                {
                    public async Task
                    UpdateTodoItem(long id, string title, string description, DateTimeOffset date, Uri imageUri, bool completed)
                    {
                        // 查
                        public string FindTodoItem(string query)
                        {
                            private bool ItemMatch(TodoItem item, string query)
                        }
                    }
                }
            }
        }
    }
}

```

b) Mainpage 对增删改查 采取在 itemlist 上的增删改查并实现数据库的同步：

( 以一个函数为例子 )：

```

//检查正确
//如果选中了
if (ViewModel.SelectedItem != null)
{
    Uri uri = (image_grid.Source as BitmapImage).UriSource;
    await ViewModel.UpdateTodoItem(ViewModel.SelectedItem.getid(), title_grid.Text, description_grid.Text, uri);
    ViewModel.SelectedItem = null;
    title_grid.Text = "";
    description_grid.Text = "";
    date_grid.Date = DateTime.Now;
    image_grid.Source = new BitmapImage(new Uri("ms-appx:///Assets/background.jpg"));
    UpdateButton.Content = "Creat";
    DeleteButton.Content = "Cancel";
}
else
{
    Uri uri = (image_grid.Source as BitmapImage).UriSource;
    await ViewModel.AddTodoItem(title_grid.Text, description_grid.Text, date_grid.Date, uri);
    title_grid.Text = "";
    description_grid.Text = "";
    date_grid.Date = DateTime.Now;
    image_grid.Source = new BitmapImage(new Uri("ms-appx:///Assets/background.jpg"));
}
}

```

c) 搜索框的实现

```

<TextBox x:Name="SearchContent" Grid.Column="0" Height="30"/>
<Button x:Name="SearchButton" Grid.Column="1" Height="30" Content="Search" Click="SearchButton_Click" Margin="10,0,20,0"

private void SearchButton_Click(object sender, RoutedEventArgs e)
{
    string query = SearchContent.Text;
    if (query == "")
    {
        var message = new MessageDialog("请输入查询内容", "Search Error").ShowAsync();
        return;
    }

    string result = ViewModel.FindTodoItem(query);
    if (result == "") result = "未找到";
    var message2 = new MessageDialog(result, "Search Result").ShowAsync();
}

```

三 . 实验结果截图



1491403482602	1491403482602	test1	test1	20170404	N	<a href="ms-appdata:///local/Images/1491403482602.png">ms-appdata:///local/Images/1491403482602.png</a>
1491405792128	1491405792128	2	2	20170405	N	<a href="ms-appdata:///local/Images/1491405792128.png">ms-appdata:///local/Images/1491405792128.png</a>

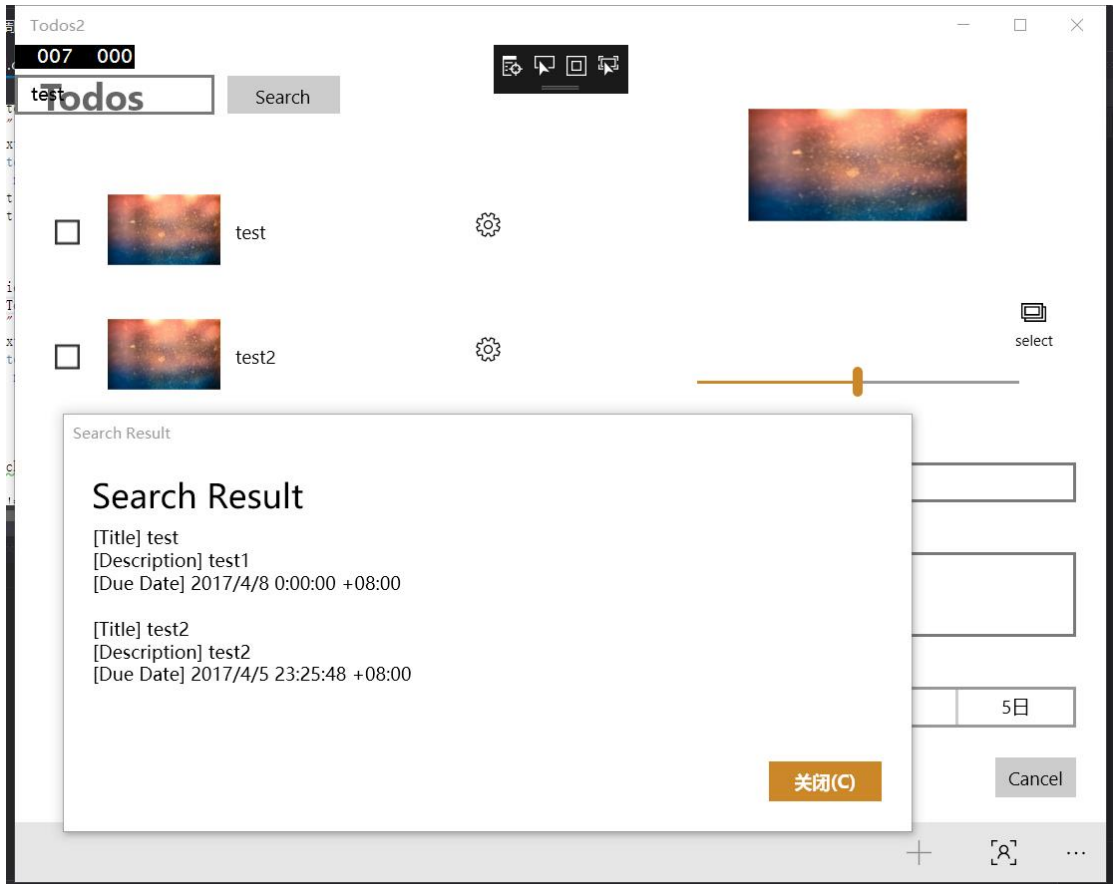
删:

1491403482602	1491403482602	test1	test1	20170404	N	<a href="ms-appdata:///local/Images/1491403482602.png">ms-appdata:///local/Images/1491403482602.png</a>
---------------	---------------	-------	-------	----------	---	---

改 (title):

1491403482602	1491403482602	test	test1	20170408	N	<a href="ms-appdata:///local/Images/1491403482602.png">ms-appdata:///local/Images/1491403482602.png</a>
---------------	---------------	------	-------	----------	---	---

查:



#### 四 . 实验过程遇到的问题

异步问题 :

C#的代码不像之前写的 js ,是可以直接等到异步的 ,一开始代码老是跑不起来 ,

后来在调用异步函数前加上 await 即可

图片问题 :

首先是 uri 与 source 的互转 , 查阅相关文档发现可以直接转化

其次是存取 , 利用 `StorageFile.GetFileFromApplicationUriAsync` 方法

## 五．思考与总结

多次接触数据库，这次也没什么变化，主要是前端在数据变化时要添加数据库的更新