

现代操作系统应用开发实验报告

学号： 14331035

班级： 周二班

姓名： 陈伟宸

实验名称： Todos

一 . 参考资料

<http://edi.wang/post/2016/2/1/windows-10-uwp-back-button-tricks>

<https://social.msdn.microsoft.com/Forums/windowsapps/en-US/ad753362-0d52-42b3-8917-b2897033ba49/getnavigationstate-doesnt-support-serialization-of-a-parameter-type-which-was-passed-to?forum=winappswithcsharp>

<http://www.newtonsoft.com/support>

二 . 实验步骤

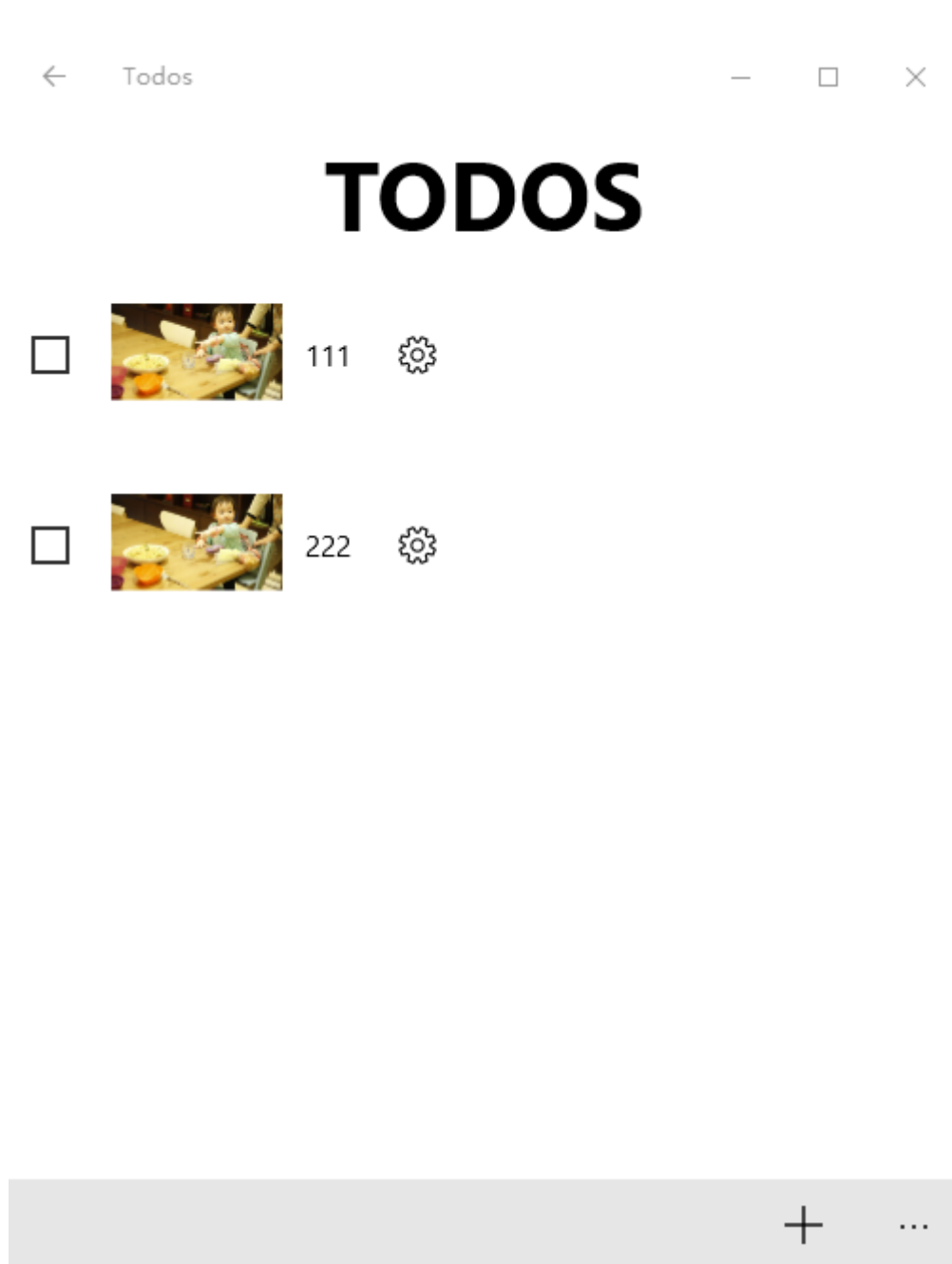
1. 修改 App.xaml.cs , 使程序在挂起并关闭之后重新运行能够回到挂起前的页面
2. 修改 TodoItemViewModels.cs , 添加用于保存数据和读取数据的函数
3. 在 NewPage.xaml.cs 中调用保存数据和读取数据的函数 , 保存挂起时的数据 , 并在程序重新运行后重新加载数据
4. 在 App.xaml.cs 中调用保存所有数据的读取所有数据的函数 , 使程序关闭后 , 再次运行时之前的数据存在。

三．实验结果截图

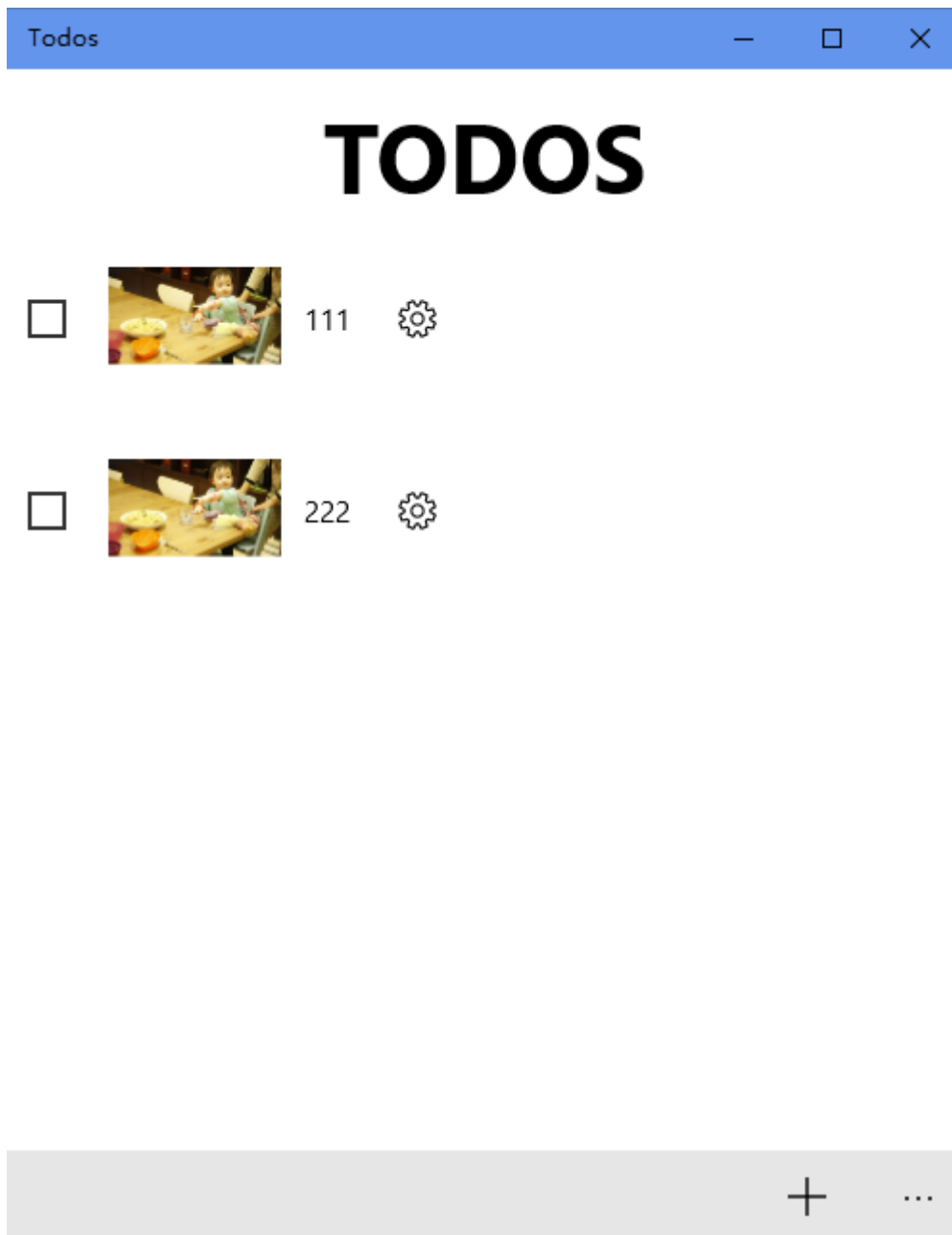
1. 运行程序，此时并未添加任何任务



2. 添加两个任务



3. 关闭程序，再次运行，之前的添加的任务仍然存在



4. 点击第一个任务，修改标题为 123

Edit Todo Item



select

Title

123



Details

111

Due Date

2016年

3月

18日



...

5. 将程序挂起并关闭，再次运行，

←


Todos


—

□

×

Edit Todo Item





select

Title

123

Details


111

Due Date

2016年

3月

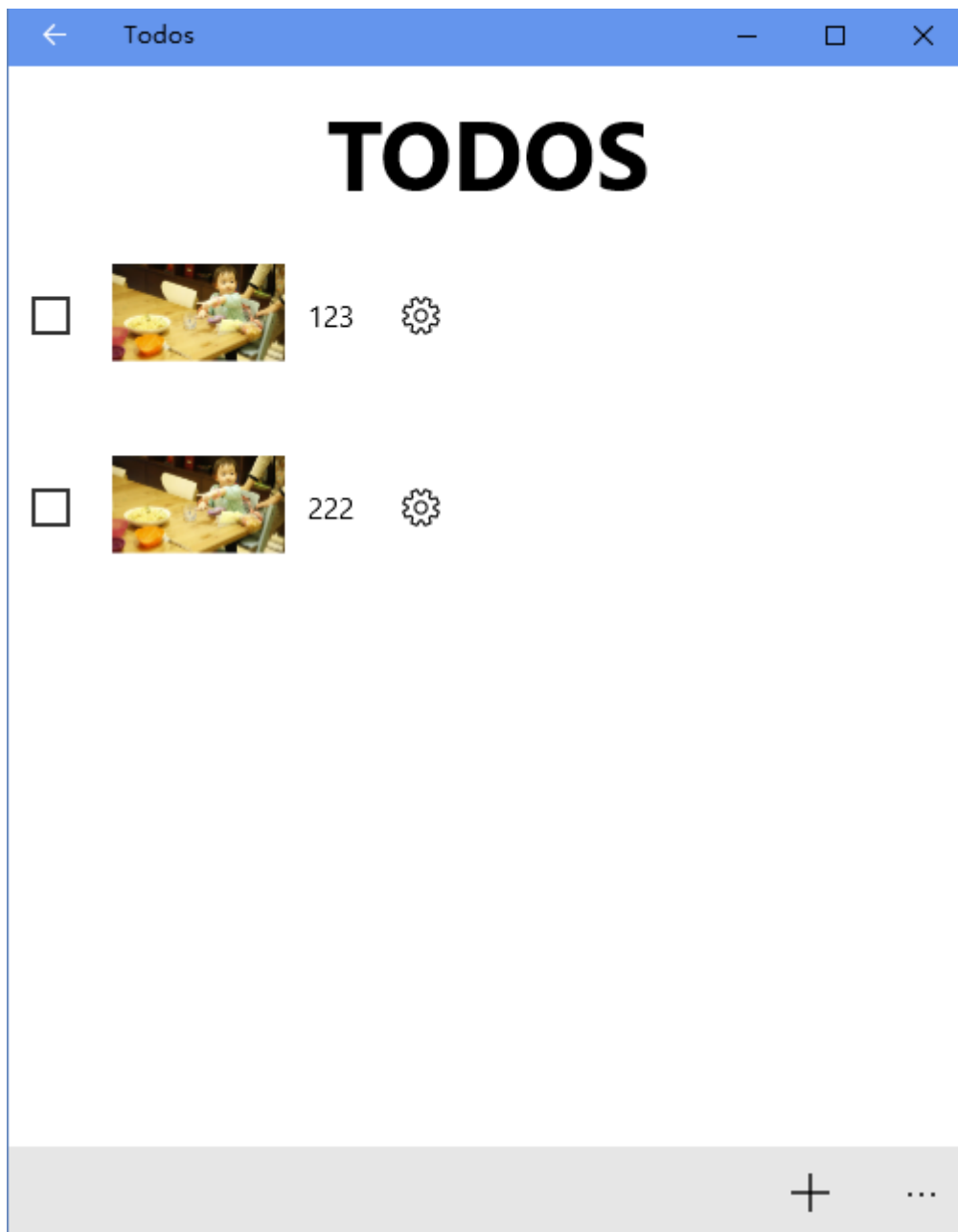
18日



...

修改的数据依然存在

6. 点击 Update ,



第一个任务的标题已更改为 123

7. 点击第二个任务，修改标题为 123，

←


Todos


—

□

×

Edit Todo Item




select

Title

123

×

Details


222

Due Date

2016年

3月

18日

 ...

8. 将程序挂起并关闭，并重新运行，

←

Todos

—

□

×

Edit Todo Item




select

Title

123

Details

222

Due Date

2016年

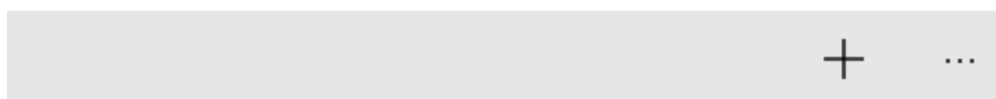
3月

18日

 ...

9. 点击 Cancel 按钮 ,

TODOS



第二个任务并未改变

10. 点击第一个任务，点击删除按钮，

TODOS

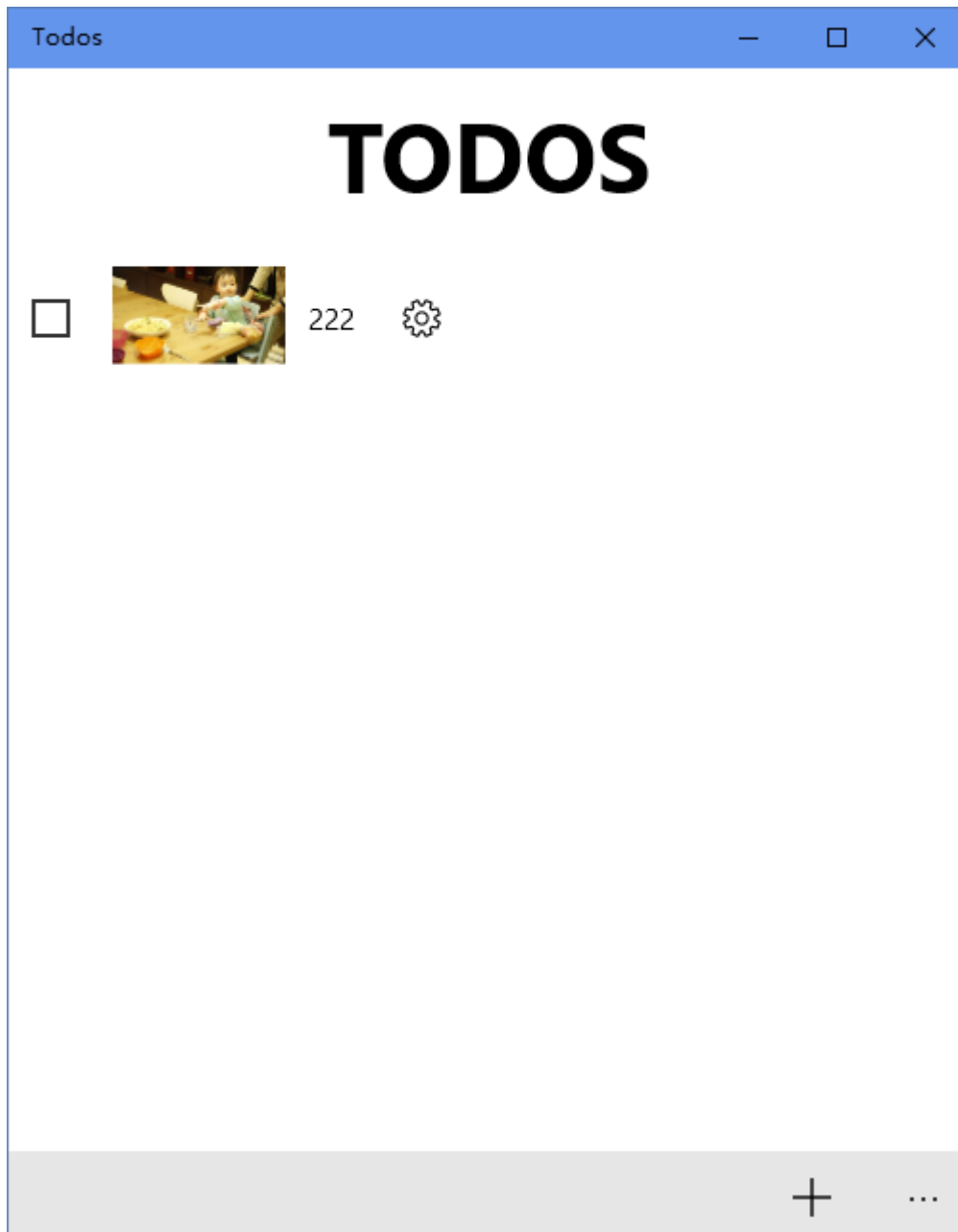


222



任务成功删除

11. 关闭程序并重新运行



数据保存正常

四 . 实验过程遇到的问题

1. 如果使用实例化的 ViewModel , 在每次页面跳转时需要传递 ViewModel , 此时调用 SetNavigationState 函数会报错 , 错误原因是该函数要求页面跳转的时候只能传递基本类型。实验中 , 曾经采用将 ViewModel 序列化后再传递的方法 ,

但是带来的新的问题是,将 ViewModel 反序列化时会调用 TodoItemViewModel 的构造函数,会导致 TodoItem 重复多个的情况。解决此问题的方法是:将 TodoItemViewModel 改为静态类,使用单例模式,这样就省去了传递 ViewModel 的步骤,自然就可以调用 SetNavigationState 函数了。

2. 由于反序列化会调用构造函数,之前的 TodoItem 的 id 将会跟上次运行时不同,导致不能使用 GUID 来定位 TodoItem。此问题暂未解决,程序中采用临时的解决方案:使用 title 来标识各 TodoItem。

3. 在 NewPage 页面挂起时,数据应该保存在一个新的 TodoItem 中,这样在重新运行程序点击 Cancel 按钮才能将该 TodoItem 恢复回原来的状态。

思考与总结

反序列化会调用构造函数,可能会导致一些很难定位的问题,在调试程序时应仔细观察各变量的变化,必要时采用单步步入的方法观察程序的执行过程。

作业要求:

1. 命名要求: 学号_姓名_作业编号,例如 14330000_林 XX_hw1。
2. 实验报告提交格式为 pdf。
3. 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。