中山大学 SUN YAT-SEN UNIVERSITY 现代操作系统期中项目实验报告

一、 项目分工

学号	名字	角色	班级	职责	贡献
14970011	宋思亭	组长	早上班	负责实现 Todos、Detail 页，整理实验报告	35%
15331220	刘沅昊	组员	早上班	负责实现 MainPage 页	25%
15331288	唐玄昭	组员	晚上班	负责测试，总结实验报告，录制视频	20%
15331067	鄧子漫	组员	晚上班	负责连接 MainPage 页与 Todos、Detail 页	20%

二、开发环境

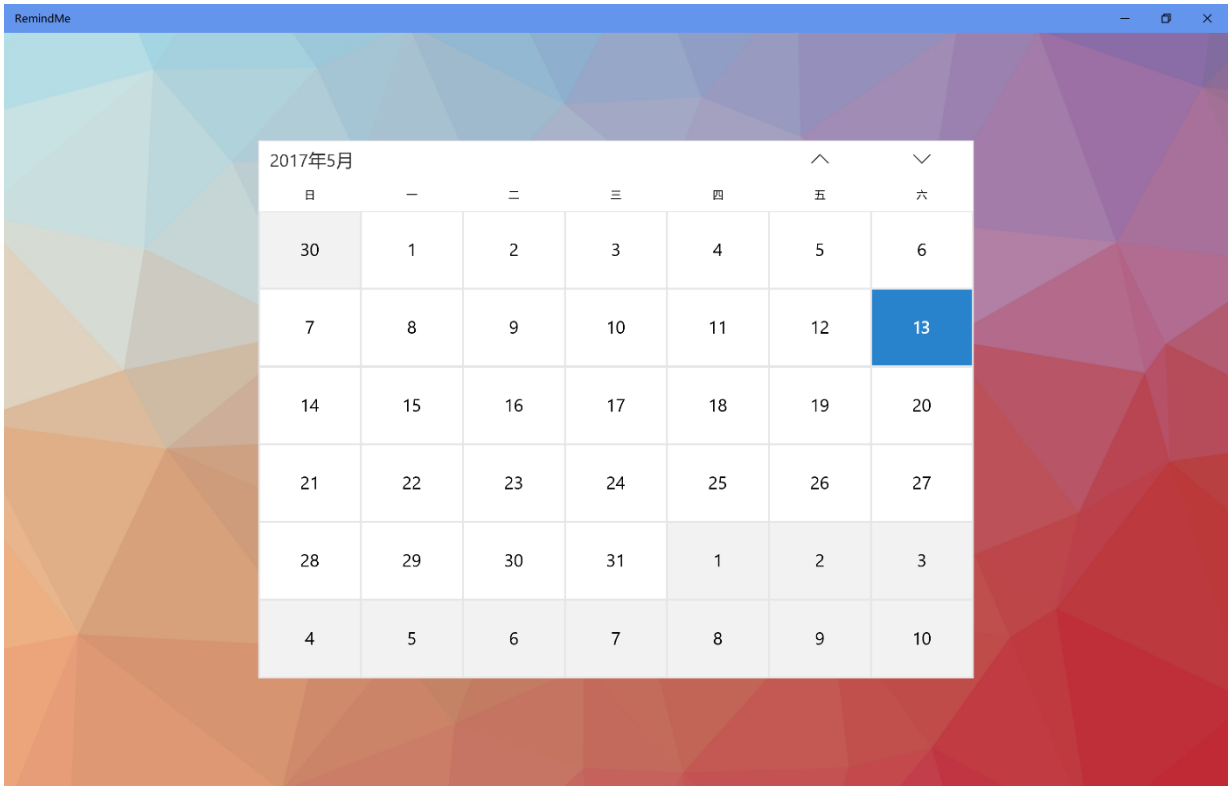
操作系统：windows 10
语言：C#、XAML
开发工具：Visual Studio

三、 项目阐述

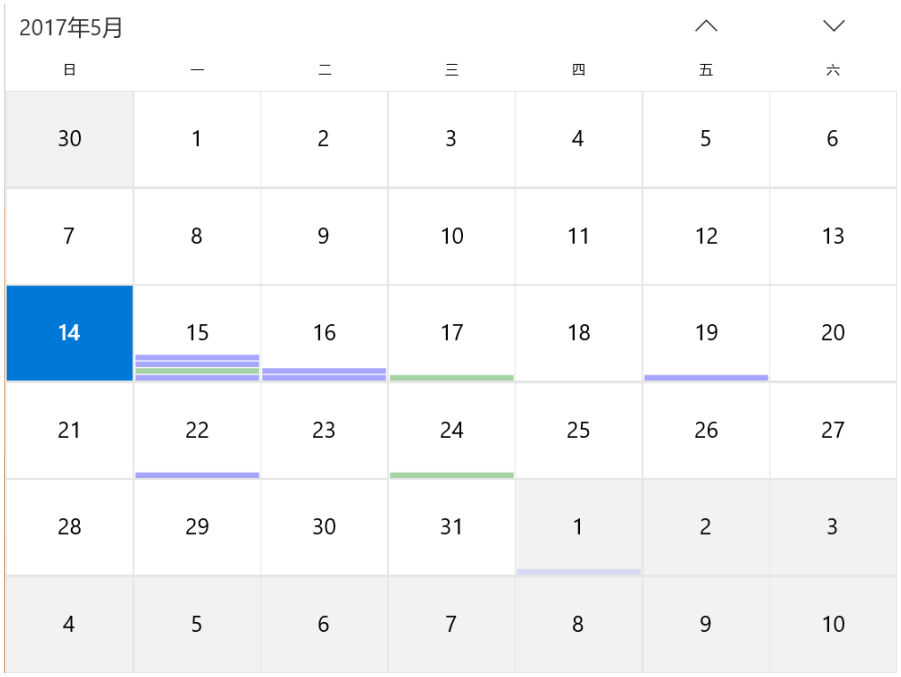
名称：RemindMe
简介：今日事今日毕。你不记得的事，我都帮你记着，还会用闹钟提醒你哦！
功能：整合备忘、提醒、整理事件的功能，方便快捷地解决事务繁多、不易整理，容易错过重要事件的现实问题。
亮点：提供直观显示每天已完成/待完成事务数量的功能，对于设置的事务有闹钟提醒功能。

四、 项目展示

打开之后首先看到的是一个很清爽的日历界面：

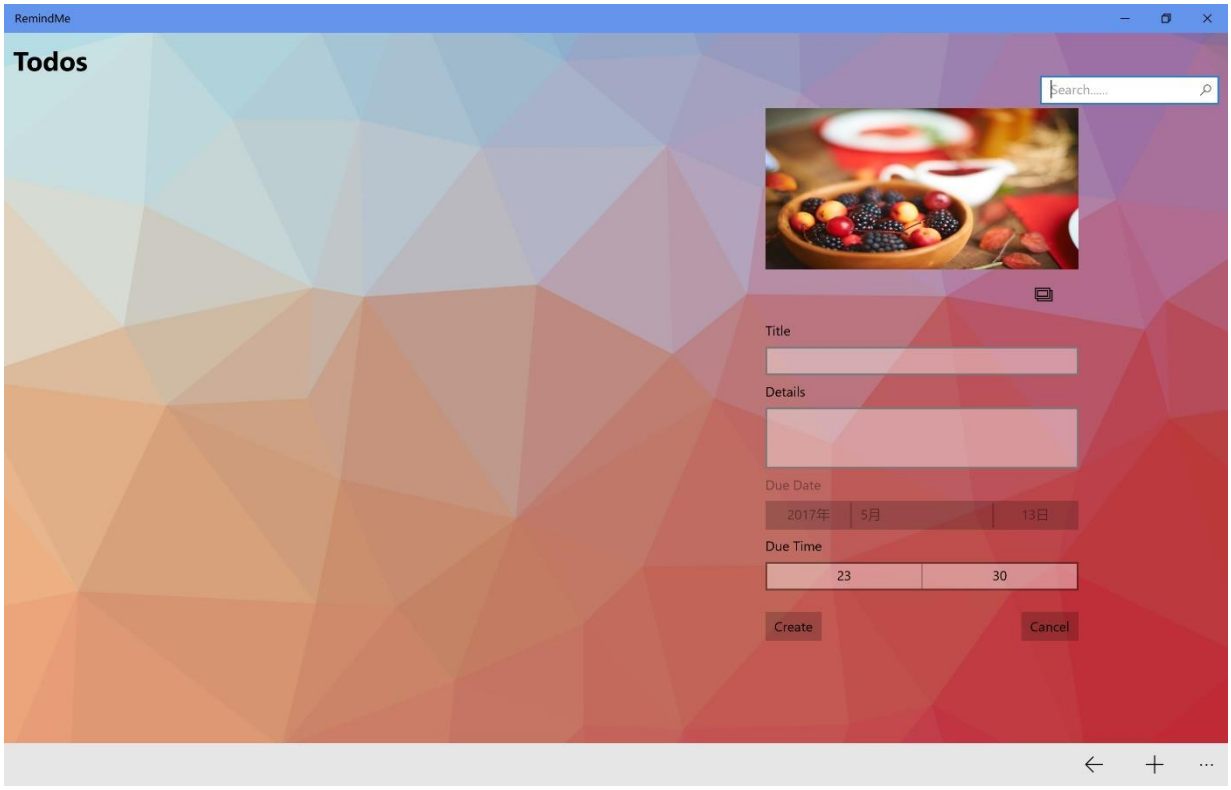


各个日期下的事件数量由小横条表示，绿色横条表示已完成事件，紫色表示待完成事件

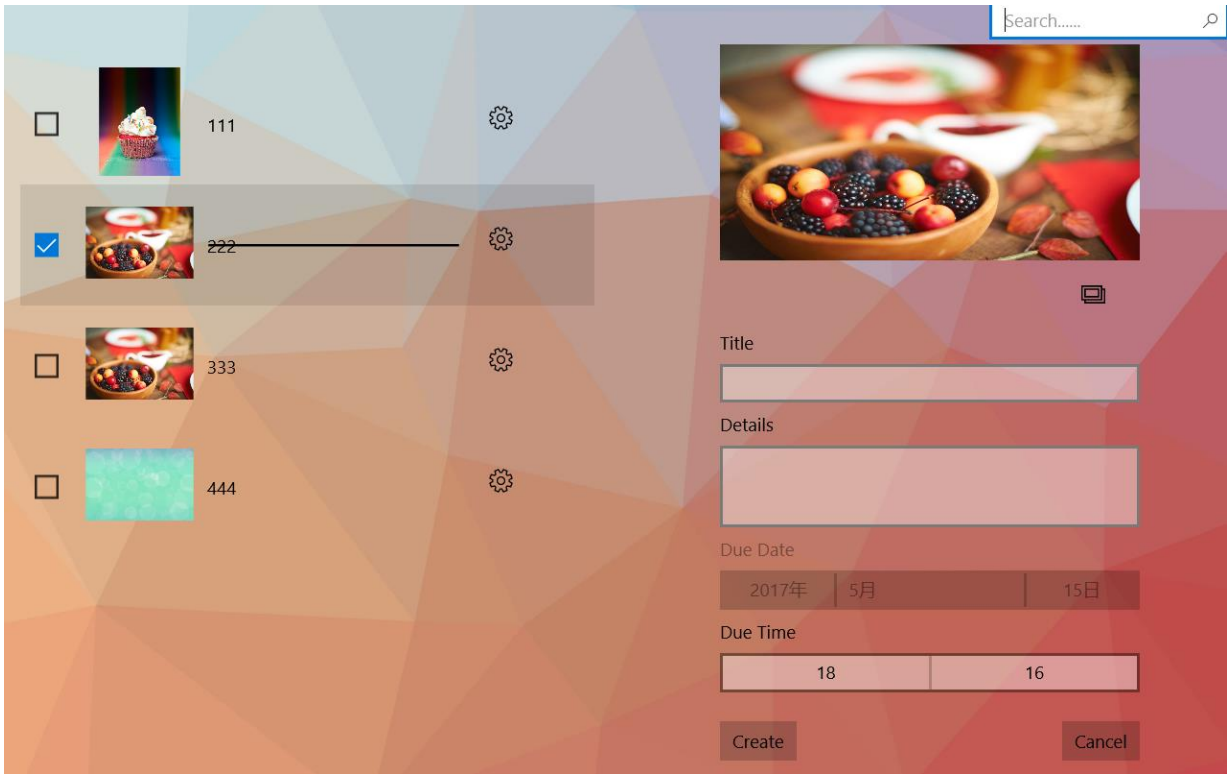


在任意一天上点击，便会跳转到所选日期的日程安排列表：

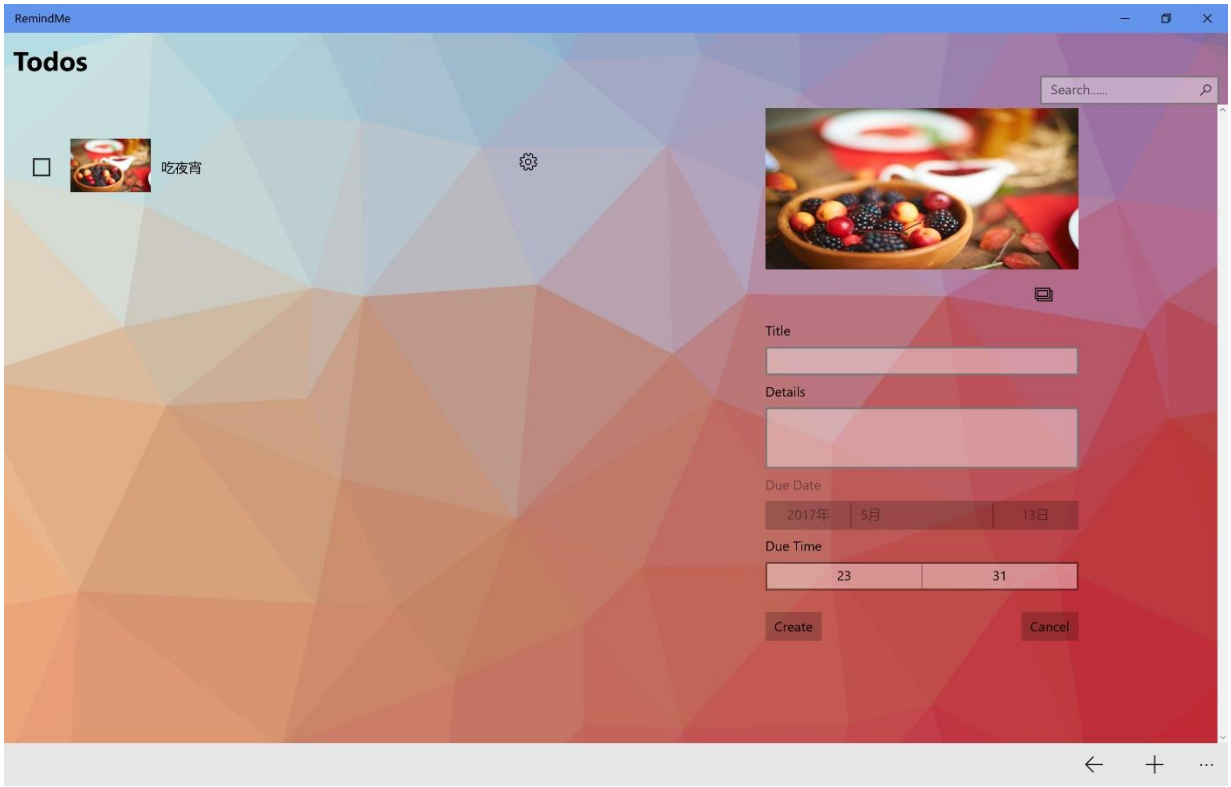
5 月 13 日



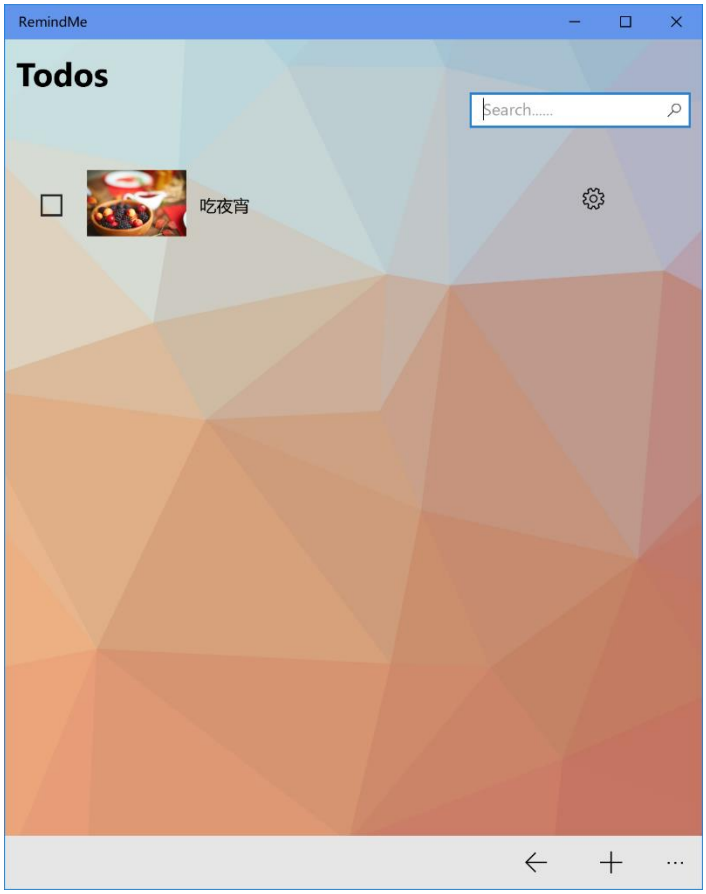
5 月 15 日



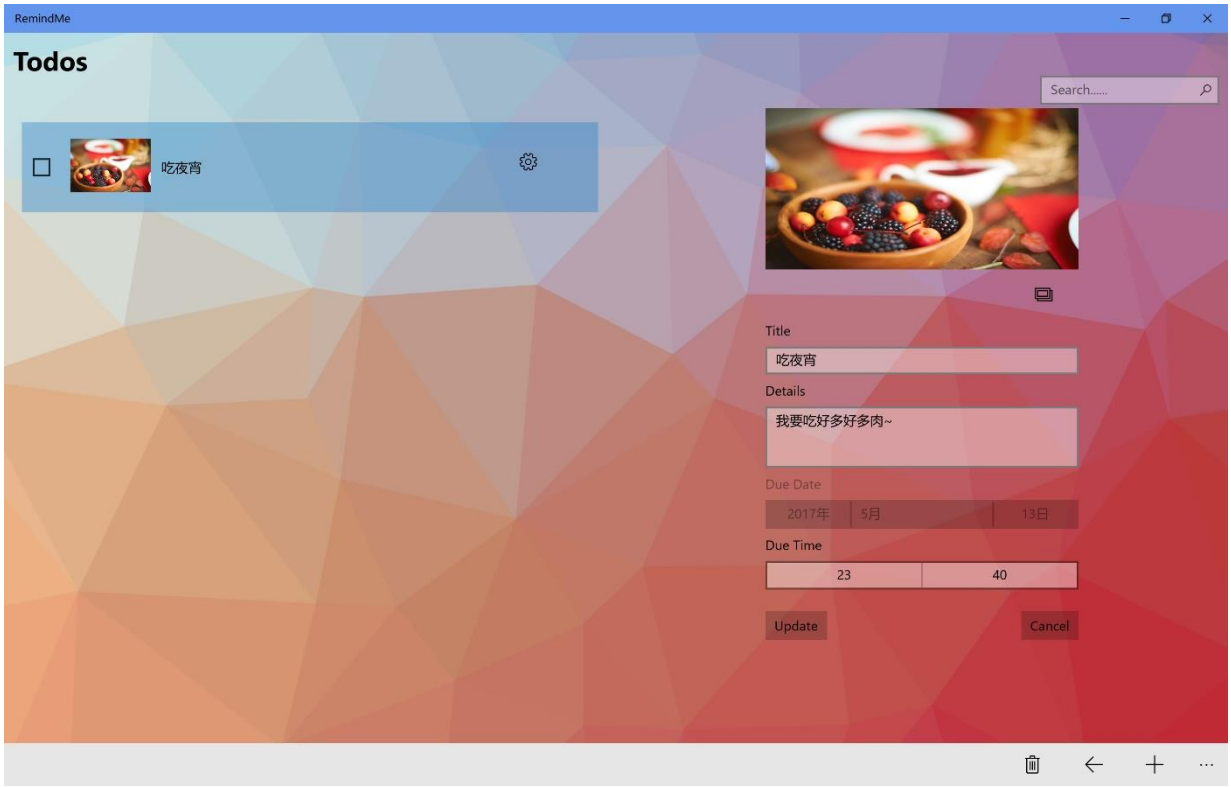
界面十分简洁，包含创建任务的查询，新任务的添加以及删除等操作
在添加了一个任务之后，可以在左边看到该任务的信息



在将屏幕缩小之后，会根据屏幕的宽度自动适配不同的显示项目：

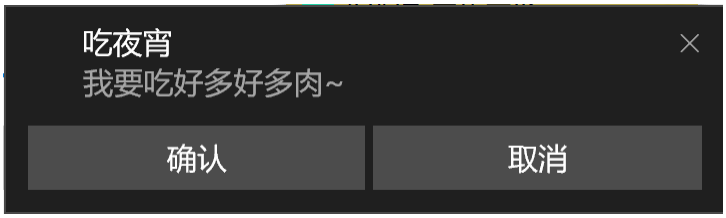


点击一个项目之后可以在右边的界面上修改信息并保存

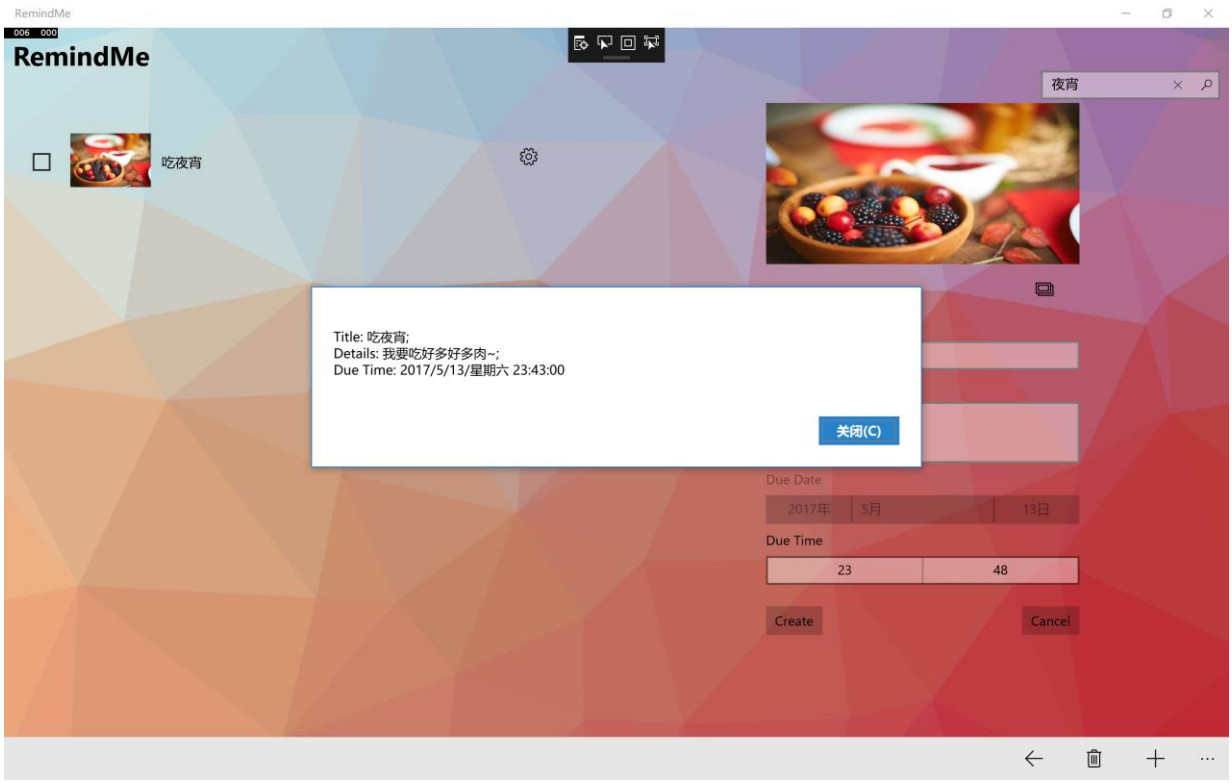


你也可以用你喜欢的图片，这里不再赘述

而在新建项目之后，系统会在指定的时间在桌面右下角/手机事件栏出现闹钟提醒：



如果你觉得项目太多找起来麻烦，你可以进行查找



系统将会根据输入的关键词进行查找，如果不存在的话将会弹出提示告知不存在

实现过程：

1. Adaptive UI

窄页面下不现实右半部份内容

```
<VisualStateManager.VisualStateGroups>
    <VisualStateGroup x:Name="VisualStateGroup">
        <VisualState x:Name="VisualStateMin0">
            <VisualState.Setters>
                <Setter Target="InlineToDoItemViewGrid.(UIElement.Visibility)"
                <Setter Target="delete.Visibility" Value="Collapsed" />
                <Setter Target="ToDoListView.(Grid.ColumnSpan)" Value="2" />
            </VisualState.Setters>
            <VisualState.StateTriggers>
                <AdaptiveTrigger MinWindowWidth="1" />
            </VisualState.StateTriggers>
        </VisualState>
        <VisualState x:Name="VisualStateMin800">
            <VisualState.StateTriggers>
                <AdaptiveTrigger MinWindowWidth="800" />
            </VisualState.StateTriggers>
        </VisualState>
    </VisualStateGroup>
</VisualStateManager.VisualStateGroups>
```

宽度小于 600 时不显示图片

```
<VisualStateManager.VisualStateGroups>
    <VisualStateGroup>
        <VisualState x:Name="narrow">
            <VisualState.StateTriggers>
                <AdaptiveTrigger MinWindowWidth="0" />
            </VisualState.StateTriggers>
            <VisualState.Setters>
                <Setter Target="image.Visibility" Value="Collapsed" />
            </VisualState.Setters>
        </VisualState>
        <VisualState x:Name="VisualStateMin600">
            <VisualState.StateTriggers>
                <AdaptiveTrigger MinWindowWidth="600" />
            </VisualState.StateTriggers>
            <VisualState.Setters>
                <Setter Target="image.Visibility" Value="Visible" />
            </VisualState.Setters>
        </VisualState>
    </VisualStateGroup>
</VisualStateManager.VisualStateGroups>
```

2. Database

数据库增删改查均有使用（以查找为例）

```
var sql = "SELECT date, time, title, details FROM Todo WHERE date LIKE ? OR time 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 + "%");
    statement.Bind(4, "%" + text + "%");
    while (SQLiteResult.ROW == statement.Step())
    {
        var date = statement[0].ToString();
        date = date.Substring(0, date.IndexOf(' '));
        var time = statement[1].ToString();
        string title = statement[2] as string;
        string details = statement[3] as string;
        alert += "Title: " + title + ";\nDetails: " + details + ";\nDue Time: " + date + " " + time + "\n";
    }
    if (alert == "")
        alert = "No result!\n";
    await new MessageDialog(alert).ShowAsync();
}
```

3. App to app communication

   可使用 QQ，email 等工具分享

```
var dc = (sender as FrameworkElement).DataContext;
var item = (ToDoListView.ContainerFromItem(dc) as ListViewItem).Content as TodoItem;
sharetitle = item.title;
sharedetail = item.details;
shareimgname = item.imgname;
var date = item.date;
sharedate = "\nDue date: " + date.Year + '-' + date.Month + '-' + date.Day + " " + time.Time;
if (shareimgname == "")
{
    shareimg = await Package.Current.InstalledLocation.GetFileAsync("Assets\\fruit.jpg");
}
else
{
    shareimg = await ApplicationData.Current.LocalFolder.GetFileAsync(shareimgname);
}
DataTransferManager.ShowShareUI();
```

4. File management

   图片文件保存相对路径地址

```
public async void setImg()
{
    if (imgname == "")
    {
        this.img = new BitmapImage(new Uri("ms-appx:///Assets/fruit.jpg"));
    }
    else
    {
        var file = await ApplicationData.Current.LocalFolder.GetFileAsync(imgname);
        IRandomAccessStream fileStream = await file.OpenAsync(FileAccessMode.Read);
        BitmapImage bitmapImage = new BitmapImage();
        await bitmapImage.SetSourceAsync(fileStream);
        this.img = bitmapImage;
    }
}
```

5. Live tiles

   采用动态磁贴，保存最近创建或更新的 5 个事件

```
TodoItemViewModel tileViewModel = new TodoItemViewModel();
var AllItems = tileViewModel.getItems;
var updater = TileUpdateManager.CreateTileUpdaterForApplication();
updater.Clear();
for (var j = AllItems.Count - 1; j >= 0 && j > AllItems.Count - 6; j--)
{
    var n = AllItems[j];
    XmlDocument tile = new XmlDocument();
    tile.LoadXml(File.ReadAllText("Tile.xml"));
    XmlNodeList tileText = tile.GetElementsByTagName("text");
    for (int i = 0; i < tileText.Count; i++)
    {
        ((XmlElement)tileText[i]).InnerText = n.title;
        i++;
        ((XmlElement)tileText[i]).InnerText = n.details;
    }
    TileNotification notification = new TileNotification(tile);
    updater.Update(notification);
}
```

五、 项目难点及解决方案

1. 日历的密度条的设置用的是 CalendarViewDayItemChanging 事件，在加载每一个 CalendarViewDayItem 的时候添加密度条，然后能根据任务有没有完成变颜色
2. 日历没有 click 事件，修改了 SelectedDatesChanged 事件，因为默认选中的日期是当天，把这个取消了
3. 增加了 DatePicker 与 TimePicker 复选框，DatePicker 获得的时间带有时分秒 00:00:00，需要另行转换格式
4. 加载日历、事件的增删改查都直接在数据库层面执行，效率较低，暂时没有更好的替代方案

六、 项目总结

在项目开发中确实遇到了许多问题，比如上面提到的，又比如 UWP 接口太少，不方便进行操作等等一系列问题，然而我们也认识到除非是真正是语言缺陷导致的问题，其他的问题一定是可解的，关键在于能不能想到解决的方法以及能不能去实现。在我们看来，决定一个软件质量的，不仅仅是它的界面，更重要的是带来的交互效果以及设计者的想法等因素。因此，我们在之前的作业的简陋的模板的基础上，对界面进行设计，对内容进行充实，我们以日历的效果来呈现，为的就是希望在原有日历的基础功能上进行扩展，以便提供更好的用户体验。

除此之外，我想说说开发过程中的感想，一个团队的协作是必不可少的，而在这基础上，怎么样去更加有效地去进行统筹规划，是组长最应该关注的事情，对于任务的分配，流程的控制，以让小组内每个人都去尽力完成自己的任务，不会有任务的重复和遗漏，以此来达到最高的效率。我们使用 gitlab 来对我们的项目进行管理，保证每个人都能很方便地去实时查看修改代码，从而使得效率更加高效，其次，我们在任务开始前，对整个任务进行分工，依据能力和个人擅长的点，去安排各自的任务，最后再整合在一起，从而完成整个项目。

综上所述，我们的项目在原有的基础上进行新功能的添加，同时也在团队管理上下了功夫，所以我们能有这么一个好的项目呈现出来，我们将会继续改进，从而在最后期末的大作业中达到更好的效果。