



一. 项目分工

学号	名字	角色	班级	职责	贡献
16340314	朱多煜	组长	周五班	填写计划书、逻辑功能部分代码实现、进行程序测试、撰写实验报告	33.3%
16340280	余漫霖	组员	周五班	搭建程序框架、改进UI界面、改进部分功能、添加统计功能，debug	33.3%
16340273	杨芮	组员	周五班	进行主体填充、逻辑功能部分代码实现、改进UI, debug, 修改实验报告	33.3%
...	

二. 开发环境

Windows10, visual studio2017

三. 项目阐述

项目名称：Kount记账

项目简介：一个简单的记账程序，专为个人或小型企业设计。只需要简单几步操作，即可快速添加账目，记录自己的收支日期、类型、明细等信息。助你养成良好的消费习惯，不在每个月末省吃俭用，天天吃土。

- 项目功能：
1. 用户可以根据自己的需要对账本进行增、删、改的操作。
 2. 在账本中，根据自己的收支情况对账目进行增、删、改的操作。
 3. 可以随时将自己的账目分享给其他人
 4. 支持账目搜索，帮你快速找到想要的账目
 5. 实现数据统计，让你的收入和支出一目了然
 6. 动态磁贴，让你不用打开程序便可以了解账目信息
 7. 外加提醒功能，防止冲动消费
 8. 播放，选择背景音乐，调节音量。

项目亮点：

1. 超支提醒功能——在创建账本时，我们可以选择每个创建一个每月有限额的账本。当我们给这个账本添加账目时，如果支出超额则会收到相应的提醒，提醒你本月指出已经超出了自己的额度。

2. 图表统计功能——对于一个账本，我们既可以对其支出和收入做一个数据的统计，并切建立一个饼状图来直观的表现用户在各类型上的支出比例。除此之外，我们还可以建立每月的收支对比图，可以直观的看出自己是不是支大于收。

3. 自定义的亚克力渲染方案，来实现背景的亚克力效果。如果用户当前焦点在该应用上，应用背景会呈现亚克力的半透明效果，若用户焦点离开了该应用，则应用的背景会显现为



自定义的浅黄色。这个参考了微软提供的XAML Controls Gallery。

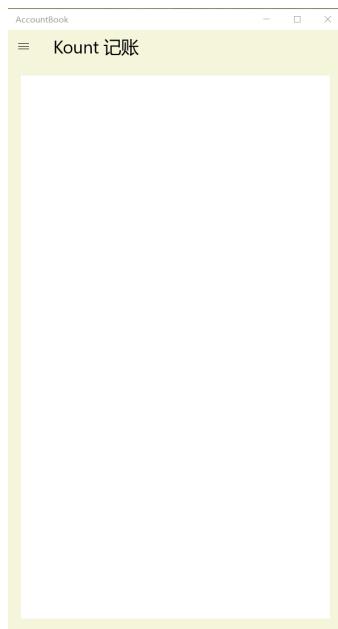
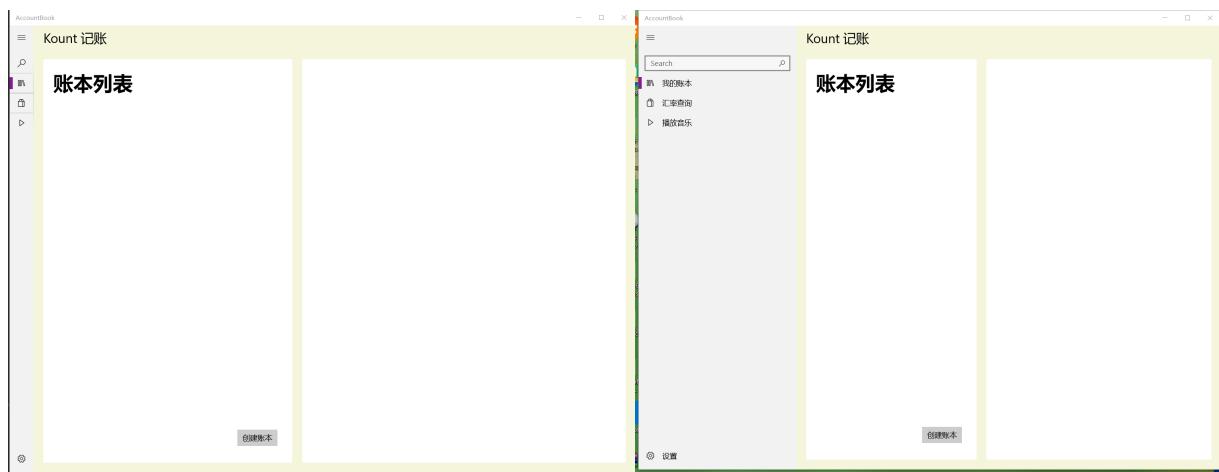
四.项目展示

项目功能演示视屏：

[https://www.teambition.com/project/5b10cf63d17ca100181f097b/works/
5b10cf63d17ca100181f097c/work/5b11675a5174c90018a892f2](https://www.teambition.com/project/5b10cf63d17ca100181f097b/works/5b10cf63d17ca100181f097c/work/5b11675a5174c90018a892f2)

以下是部分关键功能的截图

首先是进入页面时的UI，采用自适应模式





创建一个账本，有两种可选择的类型——有上限和无上限

账本信息

账本名称

 ×

每月预算

无上限 有上限

取消

创建

账本信息

账本名称

 ×

每月预算

无上限 有上限

1500

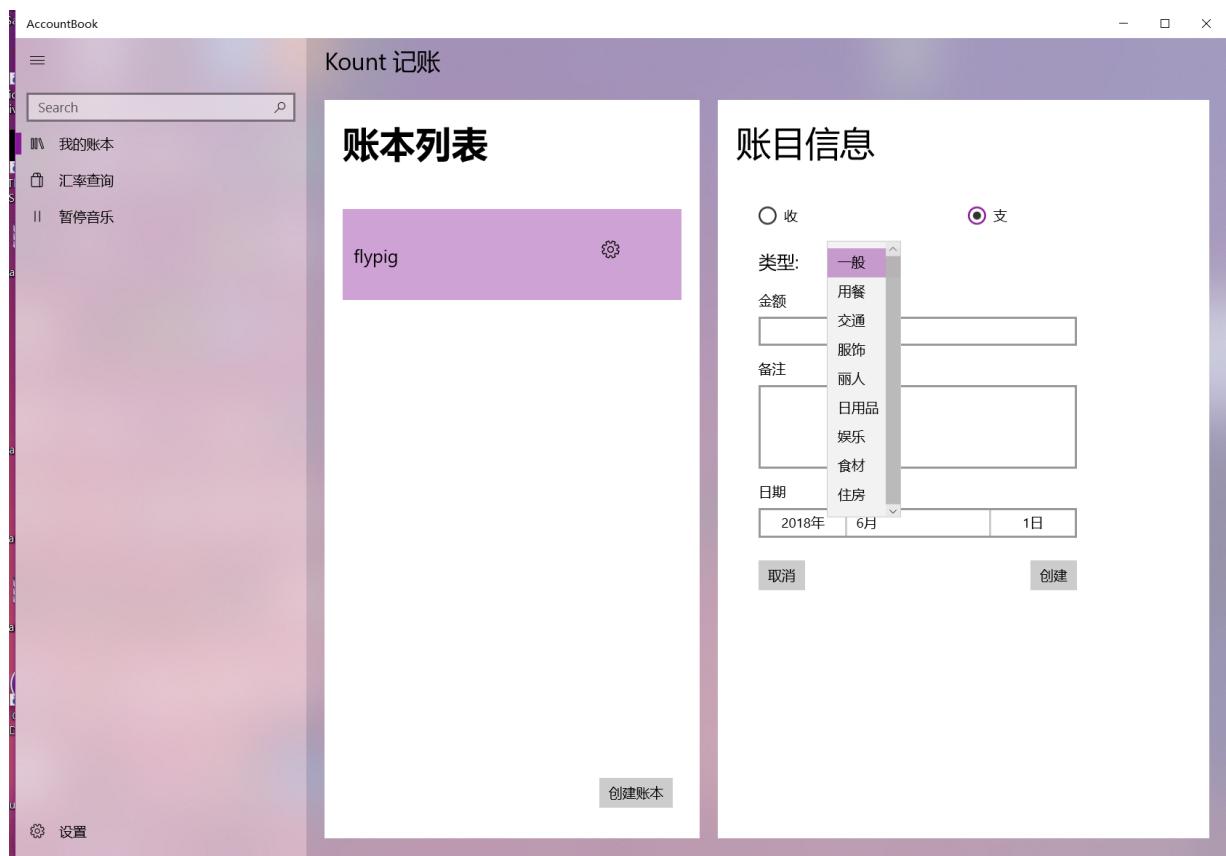
取消

创建

我们选择创建一个有上限的类型——便于后面测试提醒功能，并在该账本中添加账目。

在创建账目时我们对于支出两种方式，会有不同的类型选项





随意创建一条账目，效果如下

账目信息

收 支

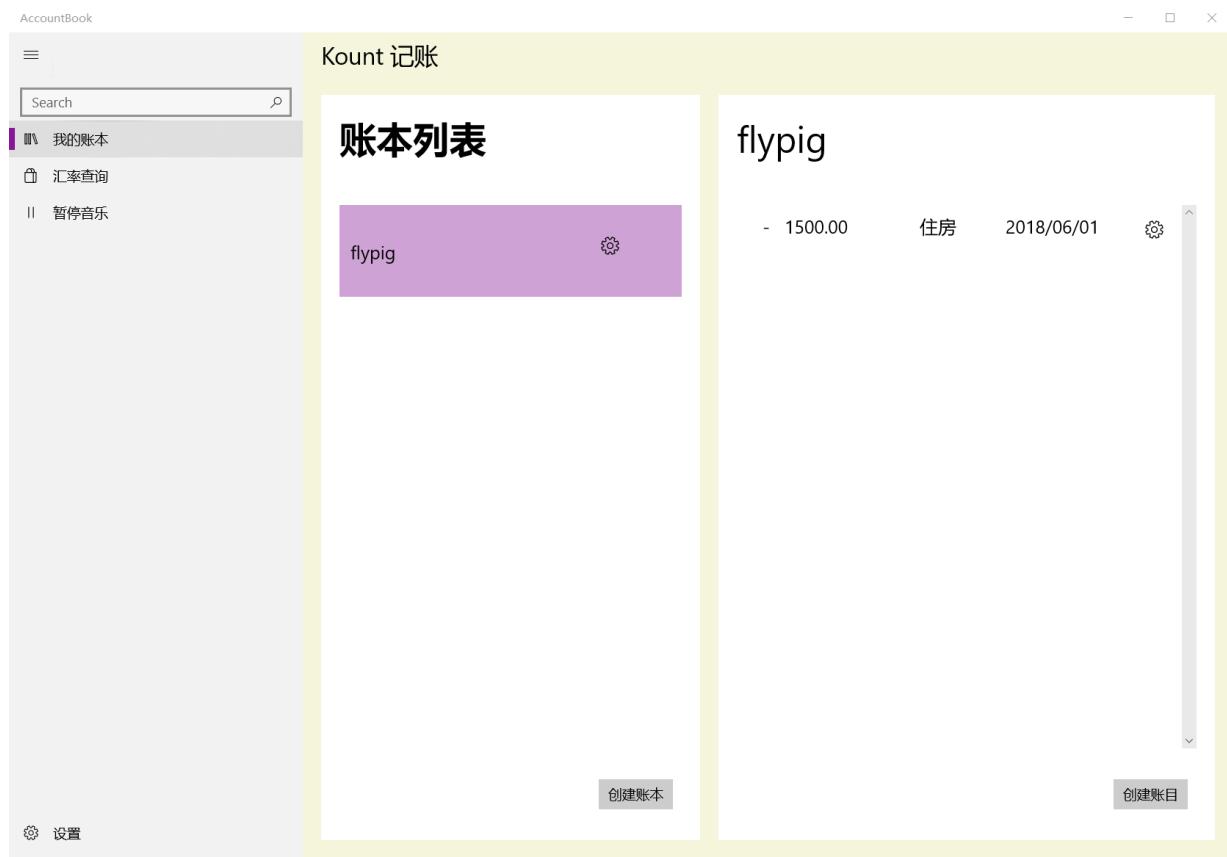
类型: 住房

金额: 1500

备注: 房租

日期: 2018年 6月 1日

取消



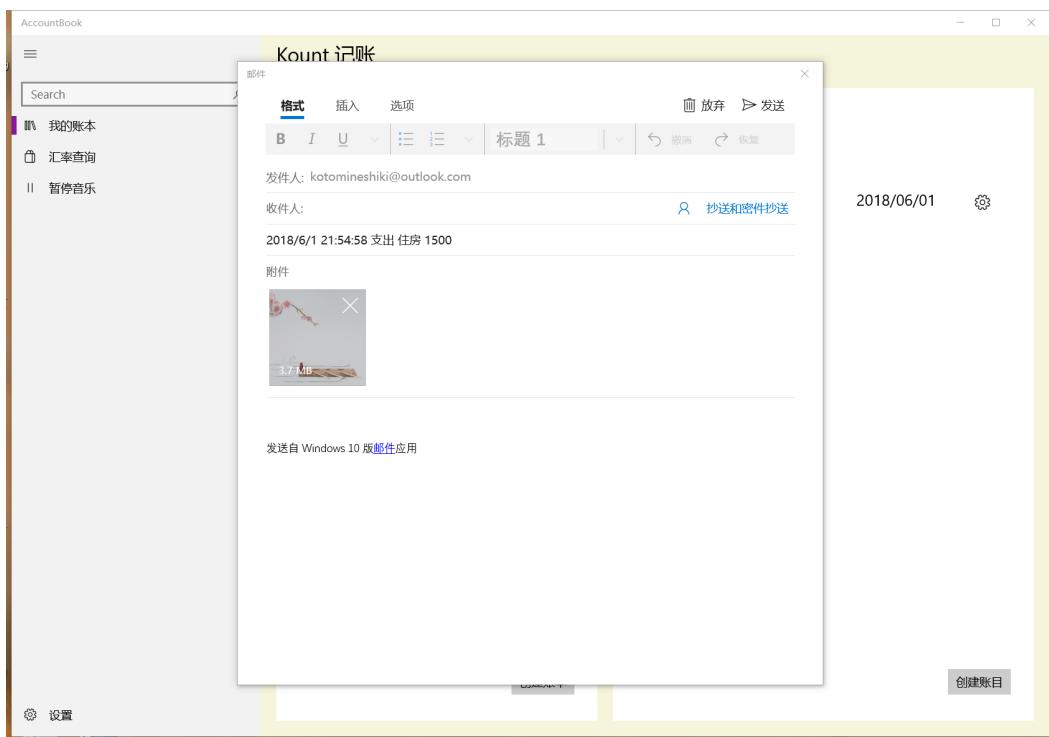
可以注意到，这是我已经用完了我当时创建账户时所定义的上限。如果我要继续创建一个支出的账目，就会出现如下的提醒



除此之外，当我们我们会根据现在所选的账本，展现动态磁贴的效果



之后我们对于每一条账目可以进行分享，界面显示如下

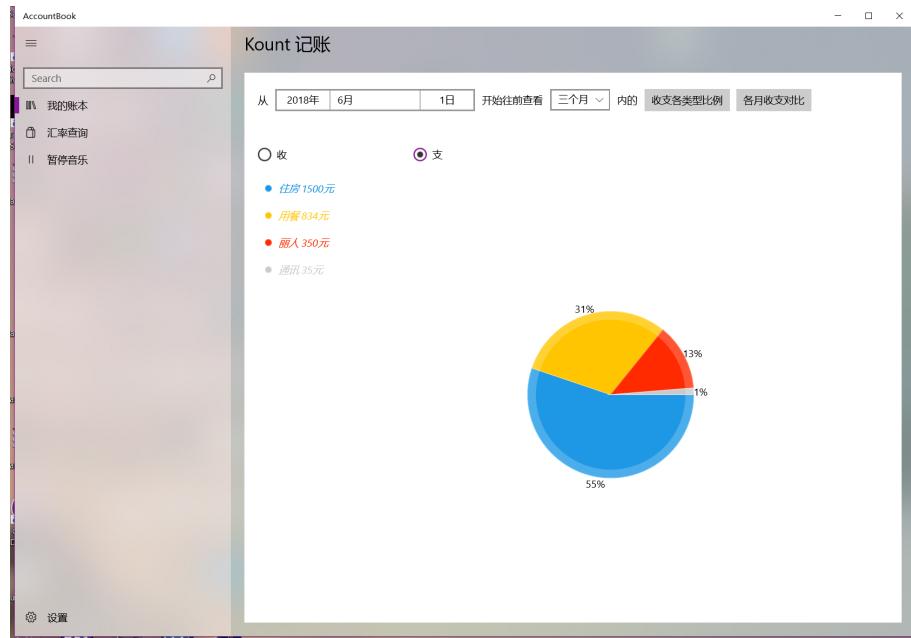


多添加几条账目后，点击账目设置按钮，进入数据统计的界面

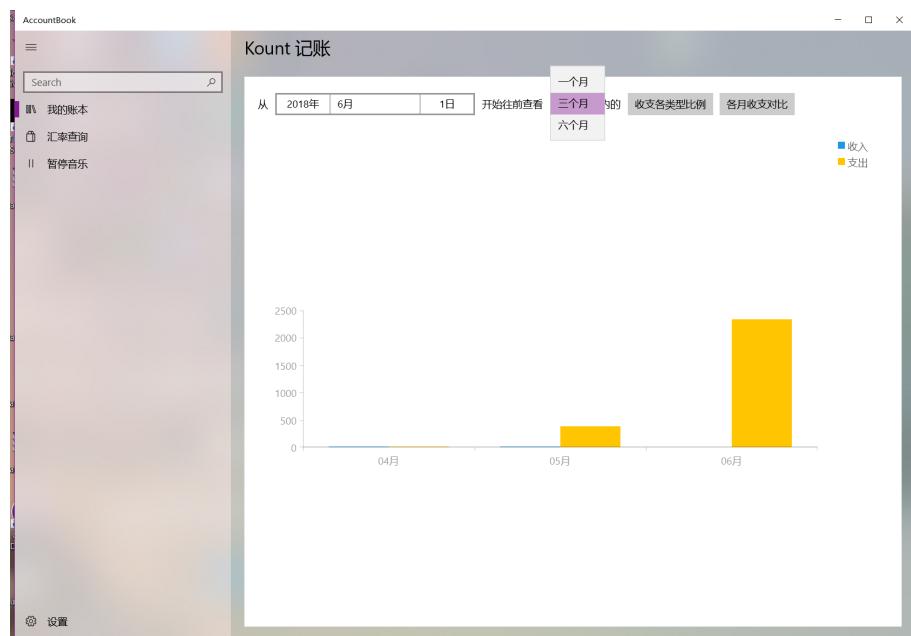




第一种数据统计的形式是收支各类型的比例，可以限定查看之前一个月、三个月和半年的收支情况



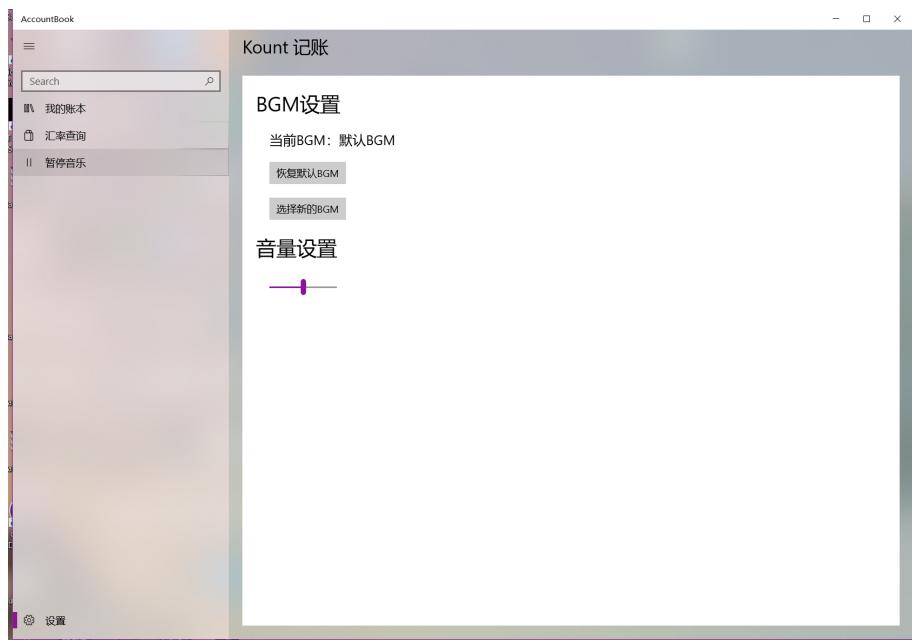
第二种数据统计类型是各月之间的收支对比，同样可以选择一个月内、三个月内和半年内的数据



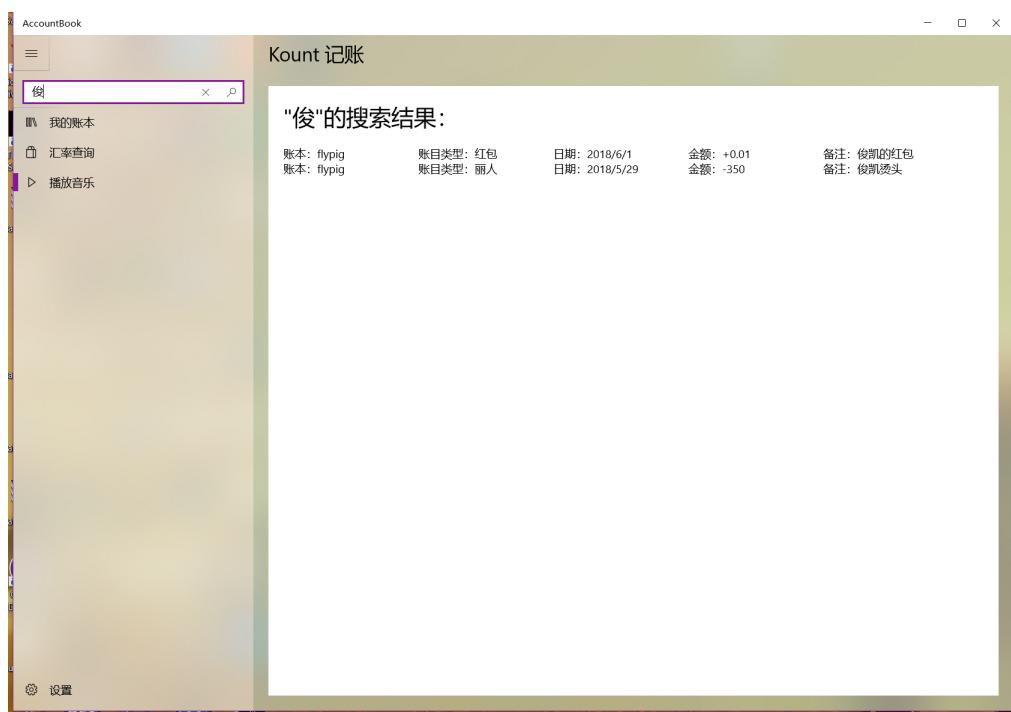
除了我的账目的这些基本功能外我们还实现了汇率查询的功能，可以查询任意两种货币之间的汇率。汇率查询下面的按键可以控制我们背景音乐的播放和暂停。



还有左下角的设置功能，设置功能可以切换当前背景音乐，或调整当前背景音乐的音量。



最后就是搜索功能，输入关键字，我们可以获得包含此关键字的所有信息



五. 项目难点及解决方案

1. 第一阶段——这一阶段主要是讨论出app的大致框架。

(1) 这一阶段的难点在于要实现一个从无到有的过程，不像是完成一个代码编程题目，会有一个很具体或者很正确的答案。这一阶段需要因为不确定的东西比较多，所我们在讨论上花费了比较多的时间，在代码实现上花的时间倒反而不是特别多。

2. 第二阶段——这一阶段主要是在上一阶段实现的框架中填充代码，实现基本功能，完成了基本的viewmodel、model的定义，以及每个页面需要显示的内容。基本上完成了记账app的多账本功能、每个账本包含不同账目功能以及利用数据库保存数据的功能。实现了创建/更新账本页面和账目信息显示页面，并且进行了数据绑定。

(1) 这一阶段最大的难点就在于实现两层的数据绑定。因为之前我们在作业中实现的绑定只有一层，而在我们的项目中不仅需要对账本与账目进行绑定，还需要对账目与其信息进行绑定。这也是这一阶段花费我们时间和精力最多的一部分。因为之前没有做过多层次的绑定，再加上网络上UWP的相关资料不多，所以我们比较难找到可以用的代码讲解或教程。最后我们采用的实现方式是——在账本中定义了list来保存账本中存在的账目。利用ViewModel来判断是否选择了一个账本，选择账本后又是否选择一个账目。在选择了一个账本之后，账单界面显示该账本中的所有账单，并且按照时间的降序排序。

(2) 在创建账单的时候，我们运用了新的ComboBox和RadioButton控件来选择账目的类型和账目是支出还是收入。但由于收入的类型与支出的类型并不相同，所以我们又需要对这两个控件在页面中的显示进行绑定。

(3) 由于我们可以储存多个账本和多个账目，所以在实现数据库储存上也出现了一些结构上的



问题。首先我们的想法是将账本和账目组成一条记录存储在数据库中，但一开始我们把账本的id作为主键，导致数据库只能存储同一个账本的一条账目记录。后来是使用账目的id作为主键，但利用这种方式的话，由于每创建一条账目，它们的id都是不同的。这就会导致我们没有办法在读取数据库的时候，高效的区分不同的账目信息。后来我们决定利用一个类型为空的记录作为一个账本的开始标志。这样当我们的数据库在读取文件时，每遇到一个类型为空的记录，就可以知道，下一条记录将会属于一个新的账本。

3. 第三阶段——这一阶段主要是优化了之前简陋的界面显示UI和搜索UI、实现了搜索功能、分享功能、汇率转化功能、动态磁贴功能、媒体播放功能、支出超额提醒功能以及媒体播放的功能。

(1) 在这一阶段中，最难也最花时间的部分莫过于统计视图的实现，可以说是我自己给自己挖的坑了。因为UWP本身不具备这样的控件，所以我们需要引用外界的工具来实现这个功能。一开始，我们选择尝试使用DevExpress来实现，但是无论是它提供的演示的微软商店应用DX Controls Demo，还是官方文档，讲解都很不清楚，很难读懂它的意思，导致花了很久时间都没成功配置，更不用说应用到我们的UWP中。不过后来，请教大佬后，大佬建议，改用Telerik UI for UWP。这个插件一下子就解决了我们的很多困扰，首先，它只需要引用一个NuGet包就可以了，比配置方法不明的DevExpress要更加轻量级、更加简单、也更加容易；其次，虽然它提供的演示应用UI for UWP Demos也有令人不解的地方，但是它的官方文档docs.telerik.com清晰齐全，给出了不少实用的例子，再和演示应用结合一起来看，就可以比较顺利地实现较为美观的饼状图和柱形图。

(2) 除了统计图的实现，我们在UI的改造和支出超额提醒上也遇到了比较大的困难。首先是各功能怎样比较合适的插入到我们的界面中，才会使得用户使用比较舒服，界面比较好看。其次是支出超额是在怎样的情况下进行提醒，如果每次打开都进行提醒的话，会不会影响到用户的使用体验。这些问题虽然都比较小，但是尝试过几种情况之后，发现对于项目的效果还是会有很大的影响。但因为我们都没有学习过比较完整的设计模式，所以只能以主页面的简洁为主，尽可能使我们的项目用起来更便利，更符合我们的使用逻辑。

六.项目总结

1. 这可以说是我们第一次完整的项目开发经历，从刚开始的确定方向，到构建一个大致的框架，再到填充框架，实现基本逻辑，再到最后的添加功能，美化UI。我们发现一个项目的开发过程远不是我们刚开始以为的那样，可以简单的分为几个大的块，然后分别去实现，再组合起来就好。不同的模块之间会有不同的联系，而这些小的联系也只有我们需要去实现时，才会发现。例如，我们原本打算将UI实现和功能实现分别分给两个人去完成，然后实现简单的组合在一起就好。但是当我们真的实现起来就会发现，很多功能的完成是基于UI的，而且显示不同的功能，对UI也会产生不同的影响。最终我们只能选择，除了部分不影响UI的功能外，我们为每个会对UI影响的功能重新创建一个界面，也就是最后只能将一个个功能分开来实现，而不能将UI和相应功能分开。



中山大學现代操作系统期中项目实验报告

2. 在这些学习的过程中，我们越发感受到官方文档和开源的重要性。不仅是Telerik UI Demos和docs.telerik.com，XAML Controls Gallery、Visualizer等用于演示的微软应用也给我提供了一些方便。虽然全英文不如中文环境那样舒适，但是只要静下心去阅读代码，一般是可以弄懂的。

(DevExpress虽有代码，但未给出明确的配置和引用类库的方法。) 尤其是对那些前人撰写与之相关的文章较少的工具，想要运用它们，研读官方文档是很有必要的。我们以后若为他人提供代码，将心比心，也需要提供清晰的注解。