

# 华中科技大学

## 《软件工程》项目报告

题目： 记账微信小程序

课程名称： 软件工程

专业班级： CS1903

组 名： 日常记账小程序

同组成员： 学号：                     

姓名：                     

学号：                     

姓名：                     

学号：                     

姓名：                     

学号：                     

姓名：                     

指导教师：                     

报告日期：

## 任 务 书

### 一 总体要求

1. 综合运用软件工程的思想，协同完成一个软件项目的开发，掌软件工程相关的技术和方法；
2. 组成小组进行选题，通过调研完成项目的需求分析，并详细说明小组成员的分工、项目的时间管理等方面。
3. 根据需求分析进行总体设计、详细设计、编码与测试等。

### 二 基本内容

根据给出的题目任选一题，自行组队，设计与开发中软件过程必须包括：

1. **问题概述、需求分析：**正确使用相关工具和方法说明所开发软件的问题定义和需求分析，比如 NABCD 模型，Microsoft Visio，StarUML 等工具 (20%)；
2. **原型系统设计、概要设计、详细设计：**主要说明所开发软件的架构、数据结构及主要算法设计，比如墨刀等工具 (35%)；
3. **编码与测试：**编码规范，运用码云等平台进行版本管理，设计测试计划和测试用例 (30%)；
4. **功能创新：**与众不同、特别吸引用户的创新 (10%)；
5. **用户反馈：**包括用户的使用记录，照片，视频等 (5%)。

## 目 录

任 务 书 .....	I
<b>1 问题定义 .....</b>	<b>1</b>
1.1 项目背景与意义 .....	1
1.1.1 需求 Need .....	1
1.1.2 方法 Approach .....	1
1.1.3 好处 Benefit .....	2
1.1.4 竞争 Competitions .....	2
1.1.5 推广 Delivery .....	2
1.2 项目基本目标 .....	3
1.3 可行性分析 .....	4
1.4 人员管理和项目进度管理 .....	5
1.4.1 人员分工 .....	5
1.4.2 项目进度管理 .....	6
<b>2 需求分析 .....</b>	<b>8</b>
2.1 需求分析概述 .....	8
2.2 UML 相关需求分析图 .....	9
2.2.1 系统用例图 .....	9
2.2.2 记账系统 E-R 图 .....	10
2.2.3 记账系统数据流图 .....	11
2.2.4 用例交互图 .....	12
2.3 原型系统设计 .....	13
<b>3 概要设计和详细设计 .....</b>	<b>16</b>
3.1 系统结构 .....	16
3.1.1 展示模块 .....	16
3.1.2 记录模块 .....	17
3.1.3 回顾模块 .....	17
3.1.4 图表模块 .....	18
3.2 对象模型设计 .....	18

3.2.1 页面类.....	18
3.2.2 数据类.....	19
3.2.3 数据库类.....	19
3.3 关键数据结构定义.....	20
3.4 关键算法设计.....	20
3.4.1 展示模块.....	20
3.4.2 记录模块.....	21
3.4.3 回顾模块.....	22
3.5 数据管理说明.....	23
3.5.1 云端资源.....	23
3.5.2 本地缓存.....	24
3.5.3 数据读取使用.....	24
<b>4 实现与测试.....</b>	<b>25</b>
4.1 实现环境与代码管理.....	25
4.2 关键函数说明.....	25
4.2.1 onLoad 界面.....	26
4.2.2 index 界面.....	26
4.2.3 chart 界面.....	27
4.2.4 record 界面.....	27
4.2.5 review 界面.....	28
4.3 测试计划和测试用例.....	28
4.4 结果分析.....	32
<b>5 总结.....</b>	<b>33</b>
5.1 用户反馈.....	33
5.2 全文总结.....	34
<b>6 体会.....</b>	<b>36</b>

# 1 问题定义

## 1.1 项目背景与意义

在本次软件工程项目开展前，本小组成员运用头脑风暴对项目进行选题，最终我们一致通过了在微信小程序平台上制作一款记账程序的方案。我们认为，勤俭节约是中华民族最重要的传统美德之一，而在日常生活中约束消费的一大途径就是记账。在过去，每个家庭往往都会有一本账本来记录所有的收支状况，但在生活节奏急剧加快的现代社会，人们很难再抽出时间用纸账本来记账、规划。因此，我们认为一个简单、便捷的微信记账小程序必定能够满足广大的用户需求、获得广阔的应用市场。

### 1.1.1 需求 Need

勤俭节约是中华民族的传统美德，也是深刻在我们内心的生活方式，中国人对于收入、支出、积蓄等财富流动是非常敏感的。例如在日常生活中，人们往往需要对自己的开销及收入进行记录，统计收入积蓄，收集商品价格信息，并根据积蓄和需要制定合适的未来消费计划。

但传统的纸质记录方式在使用时受到诸多限制，如需要纸笔工具，需要完好保存方式等。此外，如今日常节奏加快，传统的记账方式无法满足人们对快速、便捷记账模式的需求。因此人们需要一种便捷、可携带、少工具的移动记账方式，同时也需要能够回顾总结过往开销以及规划未来消费计划的功能。

### 1.1.2 方法 Approach

为保证用户使用的便捷，无需下载安装，以及产品的推广方便，本小组成员在商议后决定借助微信开发者工具，以微信小程序的形式实现移动记账功能。微信小程序的界面、功能开发语言使用 HTML 以及 Javascript，图片、UI 组件等资源使用来源网络或是开发平台提供的免费数据资源。记账小程序预计最终会实现如下功能：

- 支持不同用户同时登录程序，获取对应的记账数据。
- 能够自定义支出和收入类型，如饮食、水电费、游戏充值等。

- 能够方便快捷的完成对实时消费的记录。
- 提供整合和规划功能，方便对过往消费和未来消费的记录。

此外，在微信记账小程序完成并发布后，我们也会积极地收集用户使用后的反馈，并根据用户的意见即使更新、修正小程序的功能。

### 1.1.3 好处 Benefit

本小组所开发的软件建立在微信平台提供的便捷服务之上，用户可以使用微信直接扫码使用小程序，无需下载与安装，只需通过小程序管理页面保存我们的软件就能够随时随地记录自己的日常收支。

在小程序的功能上，我们同样考虑了用户的需求以及使用习惯，将小程序的使用方法一再简化、改进，使得任何年龄段的用户都能够轻松地使用。例如，用户可以自由地设定消费记录的类型，添加备注信息以方便日后回顾，同时用户也可以通过统计图表更加直观地看到自己一段时间内的收支状况。

### 1.1.4 竞争 Competitions

对于我们所开发的记账小程序，最大的竞争对手之一是传统的纸质记录。尽管诞生时间非常久远，在多数家庭或是商店中，纸质账本仍占据着收支记录工作的首位。但对比我们的小程序，纸质账本记录需要的记录工具繁多且不宜携带，往往只能作总体性的记录，缺乏实时性。

除了传统的纸质记账外，在软件领域，我们的记账小程序也有着非常多的竞品，例如一些移动端记账 app 或是其他的记账小程序。经过本小组调研后发现，这些竞品出于商业化的目的，往往会内置大量的广告，扰乱用户记账的注意力，这是我们的程序可以轻松避免的缺点。

对比后可知，我们的记账小程序可以规避上述二者的缺点，打造更加舒适的账本记录整合环境，使记账不再成为一种麻烦事。

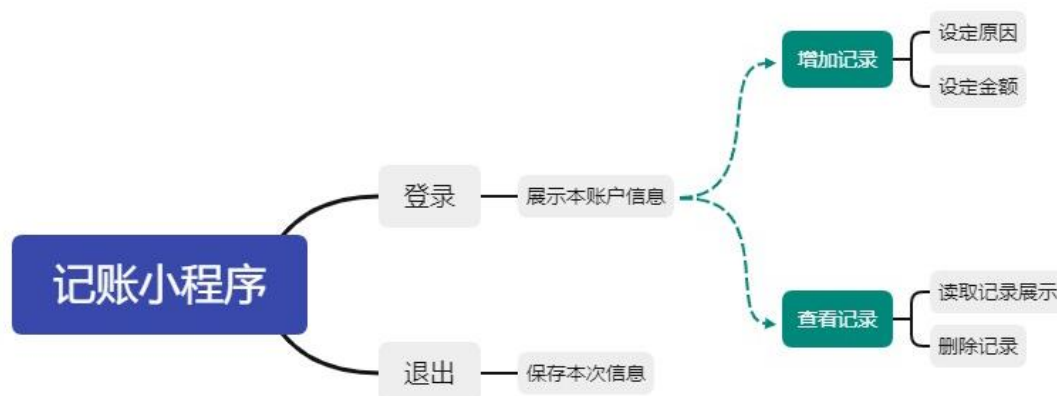
### 1.1.5 推广 Delivery

我们可以借助知名的社交平台进行营销推广，如微信公众号运营，朋友圈推广等；通过线上线下社交活动推广，例如在学院年级群、线下路演等场景中让同

学们尝试并帮助推广；借助微信小程序平台推广，让微信记账小程序推送至各个用户手中体验；向正在使用小程序的用户征集反馈意见，并发送一定的福利，产生扩大效应，吸引更多用户加入。

## 1.2 项目基本目标

本次项目的基本目标是在微信小程序平台上制作一个日常用的记账程序。针对这一目标，我们小组研究了多款移动端记账 app 或是小程序，分析出了我们需要实现的基本功能。同时，我们还研究了传统纸质记账方式以及其他记账程序的不足之处，以此来在我们的项目开发中避免这些缺点，让用户获得最佳的体验。最后，我们将研究成果绘制成图 1-1 中的思维导图。



根据如上的思维导图，我们可以基本确定项目目标的系统设计，将项目基本目标划分为下述若干个小目标：

1. 需要实现用户登录的功能，以区分在同一设备上的不同用户，同时为每一位用户单独保存账户上的记账数据信息。为此，我们可以使用微信账号作为用户账号，并借助微信云开发数据库或是本地缓存保存用户数据。
2. 需要实现用户退出的功能，在用户退出时，需要自动记录用户数据的更改，并保存在云数据库或是本地缓存。
3. 需要实现增加记录的功能，在用户记账时，可以自由设定记账金额以及原因等信息，以免查看时忘记具体原因。用户完成记账后，系统应根据本次记账数据修改该用户的数据，并在退出时保存。

4. 需要实现查看记录的功能，当用户想要回顾过往记录时，应当从云数据库或是本地缓存中读取当前用户的数据，并按记录时间逐条显示。用户在查看记录时，可以对认为错误的记录进行删改。

5. 需要实现图表展示的功能，在用户登录后，可以读取用户过往记录，自动生成统计、分析图表，以使用户可以更加清晰、直观地查看。

### 1.3 可行性分析

完成项目基本目标地分析后，本小组又针对各个目标地要求、小组成员各自的能力以及项目开发平台的便利等等因素进行考量。最终，我们从各个方面确定，当前的计划方案是可行的。

从技术难度方面对项目目标进行分析，本次项目基于微信开发者工具进行小程序的开发工作，应当是比较容易完成的。对于图形界面的开发，微信开发者工具提供了一套修改封装后的 HTML 语言——WXML，并提供了可视化的图形界面开发功能以及方便开发者使用的各类组件。此外，对于图形页面的图片等多媒体资源，我们可以采用来源于网络的免费资源。对于系统功能的开发，微信小程序所使用的语言是 JavaScript，这是一种轻量级、解释型的语言，负责为页面功能的实现提供脚本支持。尽管小组成员都未曾学过 JavaScript，但好在其易于上手。对于产品发布后的后端支持以及维护工作，微信开发者工具提供云开发服务，开发者可以使用微信提供的云服务器资源存取用户数据。

从市场需求方面对项目目标进行分析，我们认为当前记账小程序有着较为广阔的市场应用前景。从用户需求上来看，目前仍有绝大多数的人保留有记录消费、收入账单的习惯，而这些用户正需要我们所开发的这一项产品。从市场竞争上来看，虽然纸质账本仍占有大量的用户市场，且在移动端的市场上，早已有无数的记账程序在争夺着所剩不多的目标用户，但我们认为我们的产品可以在市场上站住脚跟。在产品内容上，我们会对其他的记账程序取其精华、去其糟粕，以争夺目标用户；在产品推广上，我们可以借助社交平台以及校内宣传打下用户基础，再进一步地向更广大的用户群体迈进。

从开发成本方面对项目目标进行分析，本次项目工作量较小，完成所需成本并不高。在时间成本上，本小组成员可以在课余以及完成其他课业要求的基础上，



利用空余时间完成项目。在经济成本上，本次项目不需要专门的开发设备或是开发地点，同时在数据存储资源上可以使用微信平台提供的小额的免费的云开发服务，不会产生过多的额外经济成本。

## 1.4 人员管理和项目进度管理

在确立了项目基本目标，完成了项目可行性分析后，本小组便开始制定项目进度计划，并根据计划内预计的任务负担合理地为每一位小组成员分配各个阶段的任务，并制定完成期限。

### 1.4.1 人员分工

本小组共计有 4 名成员。鉴于本次项目参与人员较少，工程量较小，我们决定采用民主制组织模式进行交流沟通，各自可以根据自身的能力以及对项目的期望，自由地选择想要负责的任务，最终的小组具体分工如下：

#### 1.

在本次项目中担任组长，以及作为小组中负责编程能力最强的成员，负责整个系统的全盘实现，以及指导、协助其他成员完成各个功能模块的具体实现。在项目开发的初期，负责完成原型系统的设计，以及后续实际实现的系统的具体结构设计。在项目开发的后期，负责对与其他共同开发的各个功能模块进行代码测试，并对源代码进行修正以及优化。在结课报告的撰写中，负责说明系统结构设计、关键数据结构以及对象设计、关键算法设计等内容。

#### 2.

在本次项目中主要负责统筹、记录项目开发中的种种细节，并负责为开发完成且已发布的产品进行维护。在项目开发过程中，负责完成登录模块设计，UI 设计等小程序前端的页面设计。在项目开发完成并发布后，负责进行产品的宣传推广，并收集用户反馈，最后根据用户意见与 一起对小程序进行版本更新。在结课报告的撰写中，负责用户需求分析、需求分析模型（UML）说明、原型系统设计展示、用户反馈以及项目总结等内容。

#### 3.

在本次项目中主要负责项目测试、图形化页面设计等任务，以保证项目最终

产品的可行性以及对于用户的吸引力。在项目开发阶段，负责完成展示模块、图表模块的设计工作，以此让用户可以最便捷、最直观地浏览过往收支统计情况。在项目开发接近尾声时，负责对整个系统功能进行测试，模拟新用户初次接触该记账小程序时可能遇到的问题，并协同 对有所欠缺的功能进行改进，力求给用户最好的体验。在结课报告的撰写中，负责系统功能实现说明、关键函数解析、系统功能测试以及分析等内容。

#### 4.

在本次项目中主要负责前期的需求分析与计划安排，以及关键功能模块的实现，以确保项目的进度管理和记账功能的完善。在项目开发的初期，负责对用户需求进行分析，并最终得到系统初步的结构设计，以此来制定进度计划、分配各个阶段的任务。在项目开发时，负责完成记录模块与回顾模块设计，以保证产品的核心功能——记账能够正常运行。在结课报告的撰写中，负责项目背景及目标、项目可行性分析以及项目分工和进度计划等内容。

### 1.4.2 项目进度管理

在进行初步的需求分析后，我们得到如图 1-2 所示的初步的系统结构设计图。该设计图将各个需要实现的产品功能划分成 4 个大模块与若干个子模块，根据各个功能模块预估的工作量，我们可以完成初步的进度计划与分工。

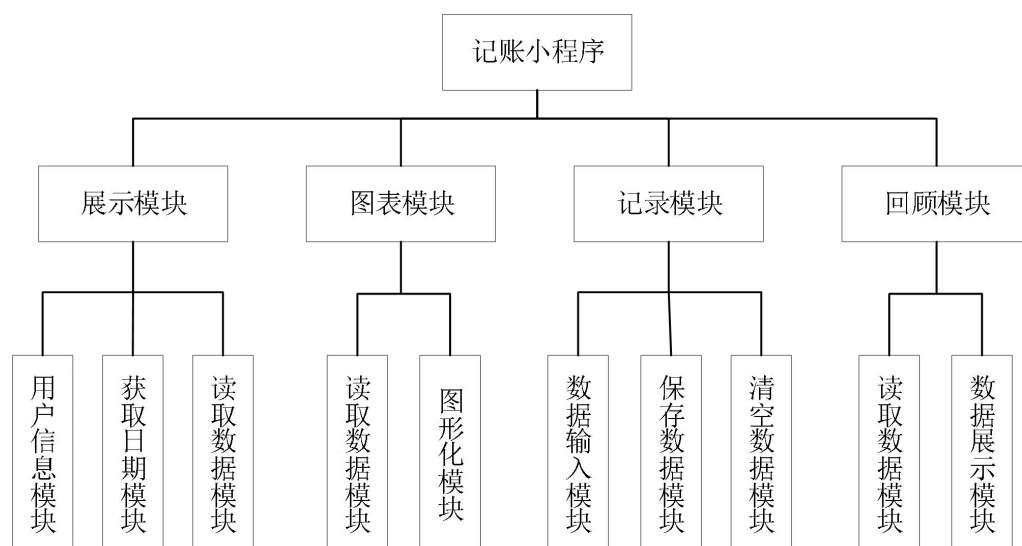


图 1-2 记账小程序系统结构图

下表 1-1 是根据上述系统结构图所分的功能模块以及项目开发过程的其他任务，得到的预计的项目完成时间的计划表。考虑到项目进行期间的课业压力以及

其他的实验任务，我们对每一项任务的计划时间进行了延长。

1. 设计阶段，分为原型设计、需求规格说明等，由所有组员共同完成。
2. 开发阶段，分为三部分的功能模块设计，由队长负责指导、协助，其他三位成员各完成一部分。
3. 测试阶段，分为系统测试和代码测试。
4. 发布阶段，分为产品发布后的用户反馈收集与处理。

表 1-1 项目进度计划表

任务	开始时间	计划天数	详细说明
需求分析	10.30	2	绘制需求分析模型（UML）
系统原型设计	10.30	5	利用墨刀完成可互动的系统原型
登录模块设计	11.8	10	实现小程序的用户账户功能
UI 设计	11.18	10	实现小程序与用户的交互功能
记录模块设计	11.8	10	实现小程序的账单记录功能
回顾模块设计	11.18	10	实现小程序的账单查看功能
展示模块设计	11.12	8	实现小程序的用户信息实时展示功能
图表模块设计	11.20	8	实现小程序的用户数据统计图展示功能
代码测试	11.29	3	对各项功能模块的代码进行 debug
系统测试	12.2	3	对系统整体进行试用并改进
用户反馈收集	12.5	5	通过社交平台收集用户的意见
用户反馈处理	12.6	5	针对用户意见对程序进行版本更新

## 2 需求分析

### 2.1 需求分析概述

在缺乏合适的记账工具的环境背景下，我们基于不同用户群体和场景做出了以下合理需求分析：

场景 1：大学生等年轻人是高消费群体，在日常生活中会有琐碎开销，需要一个方便的记账界面进行快速记账，需要高效的记账模式。且由于记账频率的问题，需要常常清理记账缓存。

场景 2：低消费群体或节俭主义者群体，往往开销频率较低，需要一个安全且可靠的记账系统，如记账本。在互联网的驱动下，这类群体往往能够使用网络应用，故同时也需要一个网络端的记账系统，来实现账单的安全且可靠的记录记录。

场景 3：上班族群体是对收支开销较为在意的群体，需要记录自己的收入与支出。并由于生存压力，需要合理规划每一笔开销，需要实时掌握自己的收支比，并做出现实中的花销调整。且由于收支的频繁性，需要一个可视化收入与支出的方式，快捷获取信息。

对于以上场景和用户需求分析，建立如表 2-1 所示的非功能性需求分析列表。

表 2-1 记账程序系统的非功能性需求

类别	非功能性需求项	需求描述
性能	EC-Req-Performance-01	界面打开时间和数据展示时间小于 0.5s
可靠性	EC-Req-Reliability-01	系统在较长一段时间内能够保存用户数据
	EC-Req-Reliability-02	系统在发生故障时，不应丢失数据
易用性	EC-Req-EasyUse-01	对于新用户，能够通过界面 UI 找到记账入口
	EC-Req-EasyUse-02	收入与支出需要做出记录区分
	EC-Req-EasyUse-03	提供账单可视化界面，快速查询开支
安全性	EC-Req-Safety-Authorization-01	所有用户需要通过微信授权登录

根据用例场景中的需求分析，可以得到以下功能：

1、用户授权：根据用户微信信息进行授权操作。

- 2、主界面展示：展示用户记录的账单信息，提供其他功能入口按钮。
- 3、记账：新增一笔记录，可填写时间、备注、价格、收支类型，提供保存和清空操作。
- 4、回顾：查看所有记录，并能对记录进行删除。
- 5、图表：将收支数据可视化，转化为图表，直观展示。
- 6、清除记录：清除用户的所有记账信息。

由以上功能需求，得到如表 2-2 所示的需求优先级列表。

表 2-2 记账程序系统的需求优先级列表

序号	用例名称	核心/外围	优先级
1	用户授权登录	外围	中
2	主界面展示	外围	低
3	记账功能	核心	高
4	账单回顾	核心	高
5	图表展示	外围	中
6	清除记录	外围	低

## 2.2 UML 相关需求分析图

### 2.2.1 系统用例图

使用用例图对用户的需求和操作做出合理分析，进一步明确目标，如图 2-1 所示。由上述的非功能性需求和需求优先级的分析，清晰用户的分级操作。

首先，为用户提供微信授权功能与主页面展示功能，该功能为关键功能。通过主界面提供的功能信息，用户以点击的交互方式进入相关功能。以主界面为根页面，扩展 4 种功能：记账、回顾、图表、清除记录。其中记账与回顾为必须功能，图表为关键功能，而清除记录为辅助功能。

同时各功能下包含多项小功能，组成该功能的实现。记账中包含填写日期、备注、价格和选择收支的功能，以及保存与清空的功能，提高易用性。回顾中包

含显示与删除记录的功能。

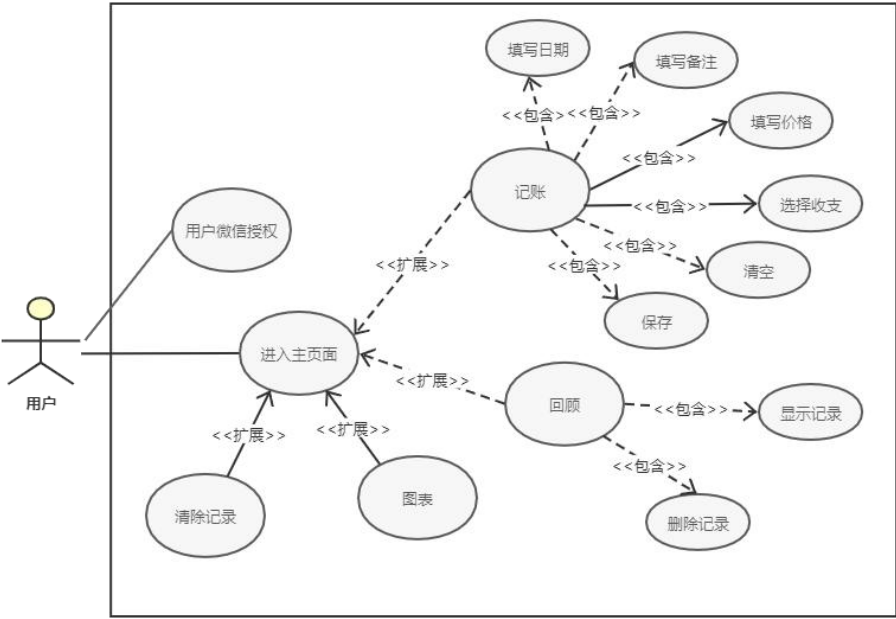


图 2-1 记账程序系统用例图

2.2.2 记账系统 E-R 图

根据上述系统用例图分析，我们将对其进行具体的实体-联系分析，从而获取所需的实体与属性，以及实体间的联系关系。记账系统 E-R 图如图 2-2 所示。

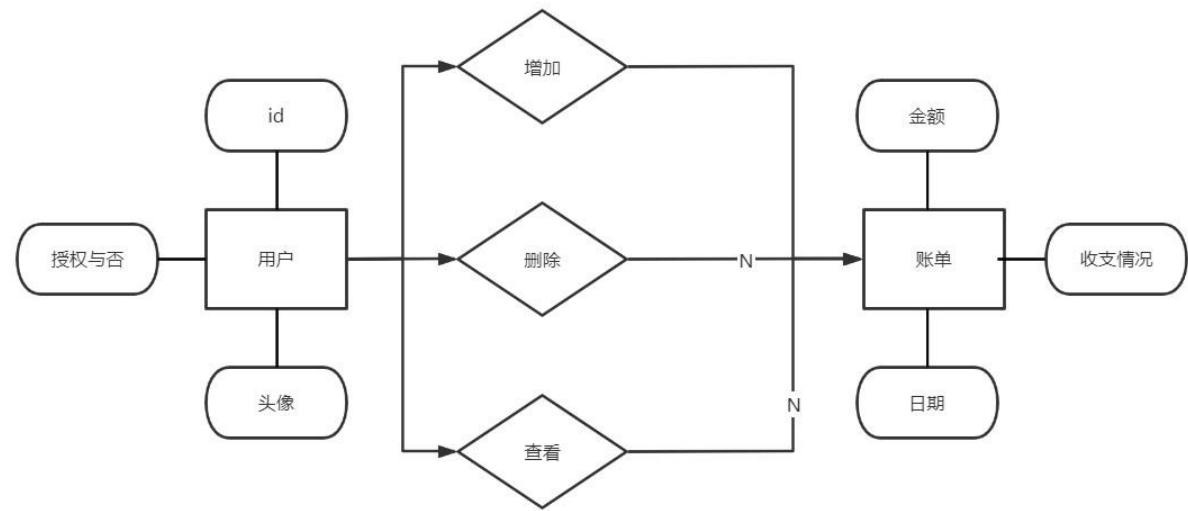


图 2-2 记账系统的 E-R 图

该系统围绕用户和账单两类实体，用户作为系统外实体，而账单则是该系统

中的抽象实体。对于系统外实体——用户而言，包含获取授权的有关信息属性，如授权确认信息、用户 ID 以及用户头像，用来作为主界面中的用户显示模块的信息准备，便于用户得知当前系统状态，以及授权信息是否正确状态。而账单信息则包含金额、日期、收支情况以及备注等属性，用与各功能的协同。基于现实生活中的记账功能实现，我们提取出了 3 个用户与账单间的联系：增加收支、删除收支以及查看收支。并将相关联系包含到各项功能的实现中。

### 2.2.3 记账系统数据流图

数据在系统内的逻辑流向和变化主要体现在授权阶段和账单信息操作阶段。绘制如图 2-3 所示的数据流图，分析系统内的数据流向。

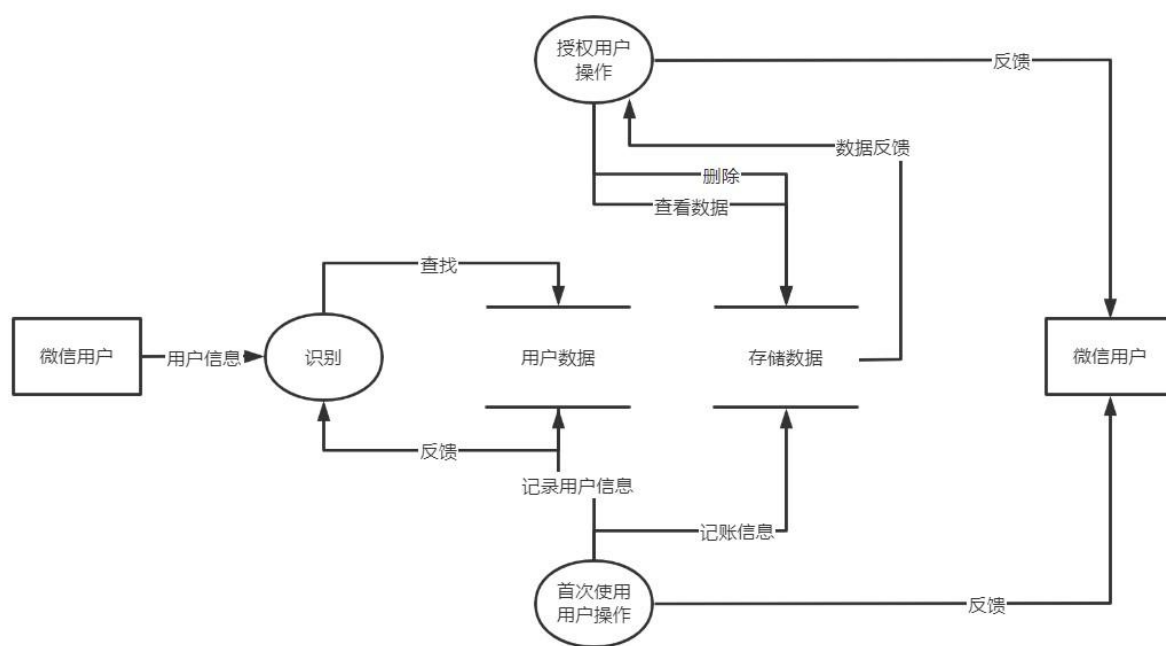


图 2-3 记账系统的数据流图

当用户进行授权时，我们对用户信息进行识别，在用户数据栏中查找有关用户数据。对于新用户而言，则需在用户数据栏新建区域，记录用户信息。

当用户进行账单信息的有关操作时，对存储的账单信息进行查看数据与删除数据的相关操作，随后相关的存储数据信息流向当前页面，对用户进行反馈。另外，对于新用户而言，需要进行记账信息的初始化过程，从而更替为老用户。

2.2.4 用例交互图

根据相关用例分析，可以清晰的展示各功能间的层级关系。通过用户的一次记账用例，绘制如图 2-4 所示的用例交互图。

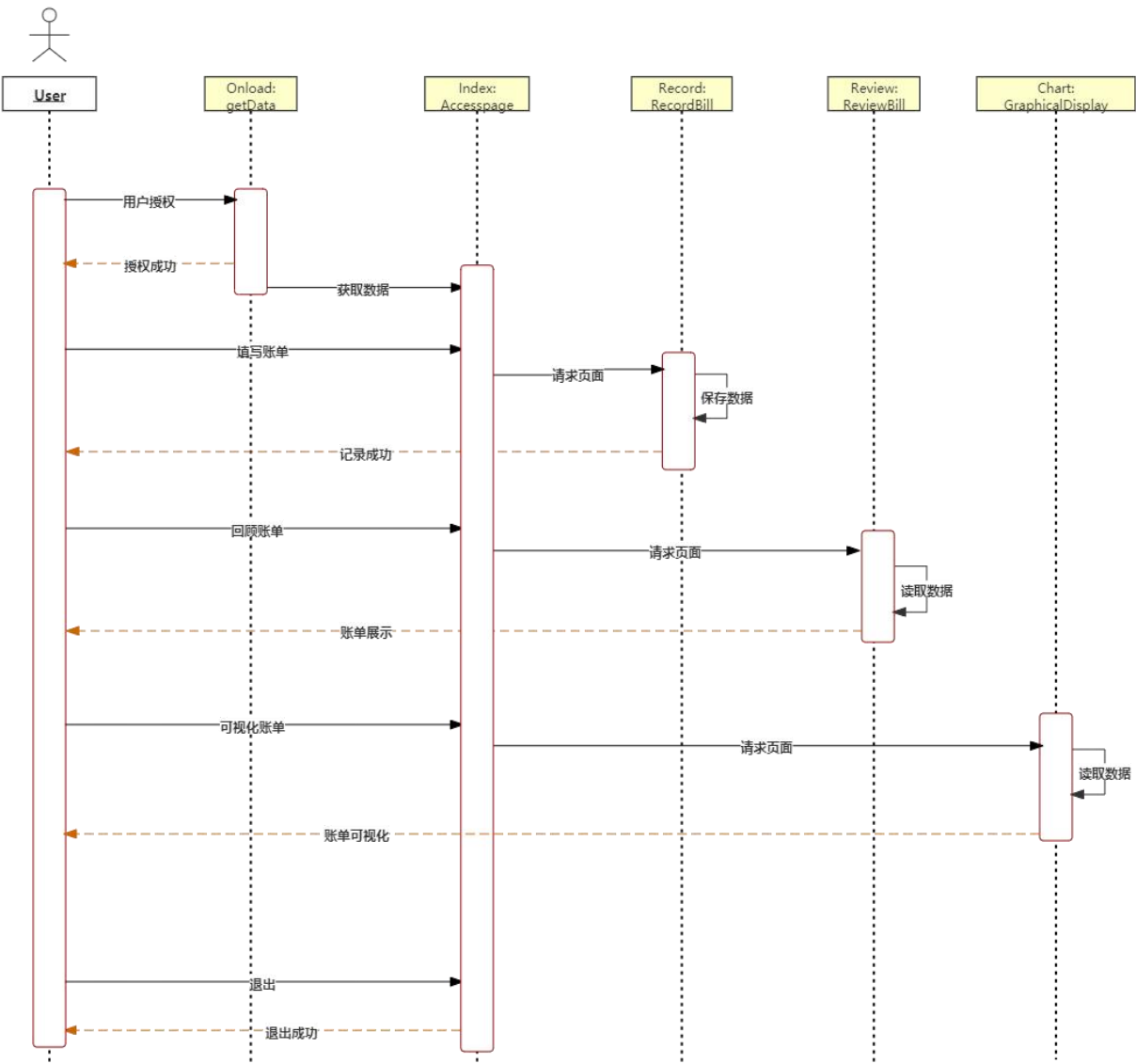


图 2-4 记账系统的用例交互图

对上图进行如下详细说明。

用户进行授权后，反馈授权成功，并跳转到主页面显示模块，在授权完成后，同样将数据逻辑性的传递给主页面使用。用户进行填写账单的点击操作后，从主页请求相关页面，即跳转到相关页面，用户进行填写账单的相关操作。当用户填写并保存后，进行数据存储，提示保存成功。用户返回主页面。相应的，当用户需要回顾记录账单时，跳转到对应页面，系统读取数据信息用于账单显示。用户



返回主页面。当用户需要可视化图表账单时，系统读取数据信息用于可视化账单。反馈可视化界面。最后用户退出系统，系统关闭，通过页面关闭反馈给用户。一次用户记账的交互过程结束。

## 2.3 原型系统设计

运用墨刀工具设计原型系统，准确说明记账系统的主要功能和用户交互界面。<https://modao.cc/app/f32f5ceb20eace5cf00aa99cee21d9df521d1998> 为记账原型系统的墨刀网页，可直接体验交互。由于原型系统为初步实现的内测模型，故在后续设计中，仅保留其中的关键性设计，对相关页面会做出调整。

### (1) 登录授权界面

在用户点击微信小程序或通过二维码扫描进入小程序时，显示登录界面。页面中心显示开始使用按钮，当用户点击后，显示获取用户授权信息的弹窗，如图 2-5 所示。

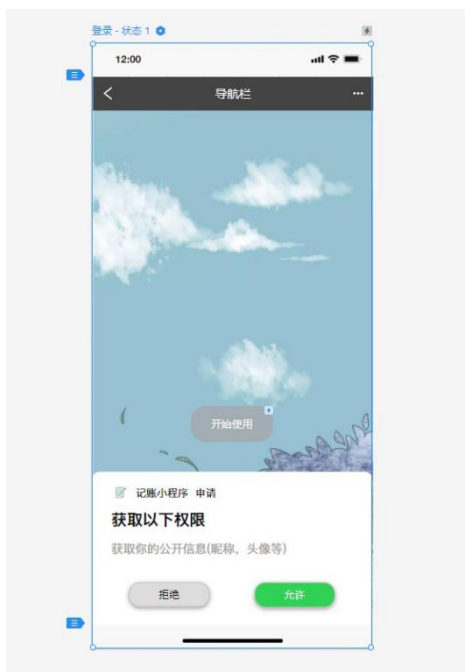


图 2-5 原型系统登录授权界面

### (2) 展示主页界面

当用户允许授权后，进入展示主页界面。在主页中，可以查看今日收入、今日支出、本周收入、本周支出，以及最近的收支情况。在最近收支情况中，显示

最近一笔的账单信息，包含收支类型、金额以及备注等。并提供记录和回顾入口按钮，指引进入相关界面。如图 2-6 所示。



图 2-6 原型系统主界面

(3) 记录界面

当在主页中点击记录按钮时，跳转到记录页面中，如图 2-7 所示。



图 2-7 原型系统记录界面

在记录界面中显示可记录选项，如时间、收支类型、金额和备注。提供记录和返回按钮。当填写完毕后点击记录会保存相关数据，并返回主页面。

#### （4）回顾界面

在主页面内进行回顾按钮点击时，显示收入与支出类型与金额，并标注有关记录备注。下分规划栏提供备忘录机制，可以对未来收支进行规划，或进行其他收支信息或收支事件的补充记录等，为用户提供更加自由的体验。整体页面如图 2-8 所示。



图 2-8 原型系统回顾界面

## 3 概要设计和详细设计

### 3.1 系统结构

整个微信小程序的项目，可以分解为多个模块，依靠模块自身功能的实现以及模块之间的协同合作达到预期的效果。对系统结构进行分析，可以将它分解为展示模块，记录模块，回顾模块和图表模块。

其中展示模块负责提供登录接口，读取用户信息，并将用户存储过的信息展示于主界面。记录模块则是记账小程序的核心，它提供记录收支信息，云端存储，更新本地数据等功能。回顾模块会将记录过的收支信息进行列表展示，同时提供信息删除功能。图标模块则给出信息的图形化展示，使得数据更加直观。

各个主要模块下，还有具体实现某一功能的小模块，系统整体结构如图 3-1 所示：



图 3-1 系统整体结构图设计

#### 3.1.1 展示模块

展示模块主要由三个模块构成：用户信息模块，获取日期模块以及读取数据模块。各个模块的功能划分如下：

（1）用户信息模块：该模块用于获取用户属性。使用全局变量 `app.globalData.used` 指示当前登录用户属性。如果用户使用过小程序，则保存有相应的应用数据；否则，则为他新增一个记录。同时，模块还会获取用户头像，用于主界面展示。

(2) 获取日期模块：用于获取用户打开界面的日期信息，使用全局变量 `app.globalData.currentData` 存储信息，为后续统计所有日期收支以及数据存储提供支持。

(3) 读取数据模块：用于获取记账信息，对于使用过微信小程序的用户，使用 `wx.getStorage` 将 `res.data` 的数据提取到 `app.globalData.dataArray` 存储。即读取过去所有收支信息，在主界面展示，并用于回顾界面和图表界面。

### 3.1.2 记录模块

记录模块提供记录收支信息功能。其下包含包括输入数据模块，保存数据模块与清除数据模块：

(1) 输入数据模块：提供输入窗口和选择窗口。用户通过虚拟键盘输入数据或者通过选择器选取数据输入。

(2) 数据保存模块：此模块是记录模块核心功能实现的重要组成部分。它会将用户输入的日期，金额，收支信息，以及备注等信息组合，随后将信息存储于本地缓存以及云端存储。同时根据收支情况，更新全局的收支信息。

(3) 清除数据模块：构造 `clear` 函数，将数据清空，置数据序列为 0。为用户下一次输入做准备。

### 3.1.3 回顾模块

展示模块展示记录收支信息，包括记录日期，收支金额，收支类型和备注。同时提供数据删除功能，能够对每一条数据进行删除。它包括读取数据模块和数据展示模块：

(1) 读取数据模块：此模块会从保存的用户数据中读取已经保存的用户信息，将每一条收支情况组合成列表的形式。同时，它提供删除数据的功能，删除数据后会同步存储。

(2) 数据展示模块：将读取并组合成列表的数据进行展示，为每一条数据提供删除按钮，底层使用读取数据模块提供的删除数据接口。

### 3.1.4 图表模块

图表模块使用图形化方式展示当前收支。具有读取数据模块和图形化模块

(1) 读取数据模块：读取存储的数据，并对按日期对所有数据进行合并，得到最终的展示数据集。

(2) 图形化模块：将得到的展示数据集展示成收入情况柱状图以及支出情况树状图。

## 3.2 对象模型设计

本次微信小程序的编写主要使用 JavaScript 语言，故在设计过程中使用了面向对象的方法，主要包含页面类、数据类以及数据库类。

### 3.2.1 页面类

此类及其子类共有的方法为 `onLoad()`，`onReady()`，`onShow()`，`onHide()`。分别实现页面的初始化，准备，展示以及隐藏。除此之外，各个子类还会实现自己的方法。

(1) `onLoad` 继承页面类，主要属性用户授权标记 `app.globalData.used`，主要实现 `login()` 方法，完成用户登录功能，获取用户属性。

(2) `index` 继承页面类，主要属性为用户头像，总收入，总支出，今日收入，今日支出等。主要实现 `remove()` 方法，能够将本地存储的所有数据进行清除，但不会清除云端数据库数据，重写 `onShow()` 方法，使其展示读取得到的数据。

(3) `record` 继承页面类，主要实现 `post()` 方法，能够将用户输入的数据保存到本地以及云端数据库。

(4) `chart` 和 `review` 继承页面类，主要属性均为记录的数据数组，均是重写 `onShow()` 方法。但前者完成图像化展示，后者完成数据读取以及列表化展示与删除。

3.2.2 数据类

此类是所有输入数据的父类，包含日期，金额，备注，收支等属性，主要实现增加数据 `add()`方法，读取数据 `read()`方法以及删除数据 `remove()`方法。

3.2.3 数据库类

主要属性为今日日期，总收入，总支出，今日收入，今日支出，数据数组以及用户授权标记。会在程序开始时初始化实例，存储所有产生的数据，同时同步本地以及云端。主要方法为增加数据 `add()`方法，读取数据 `read()`方法，删除数据 `remove()`方法以及清空所有数据 `clear()`方法。

整体的系统类图如图 3-2 所示：

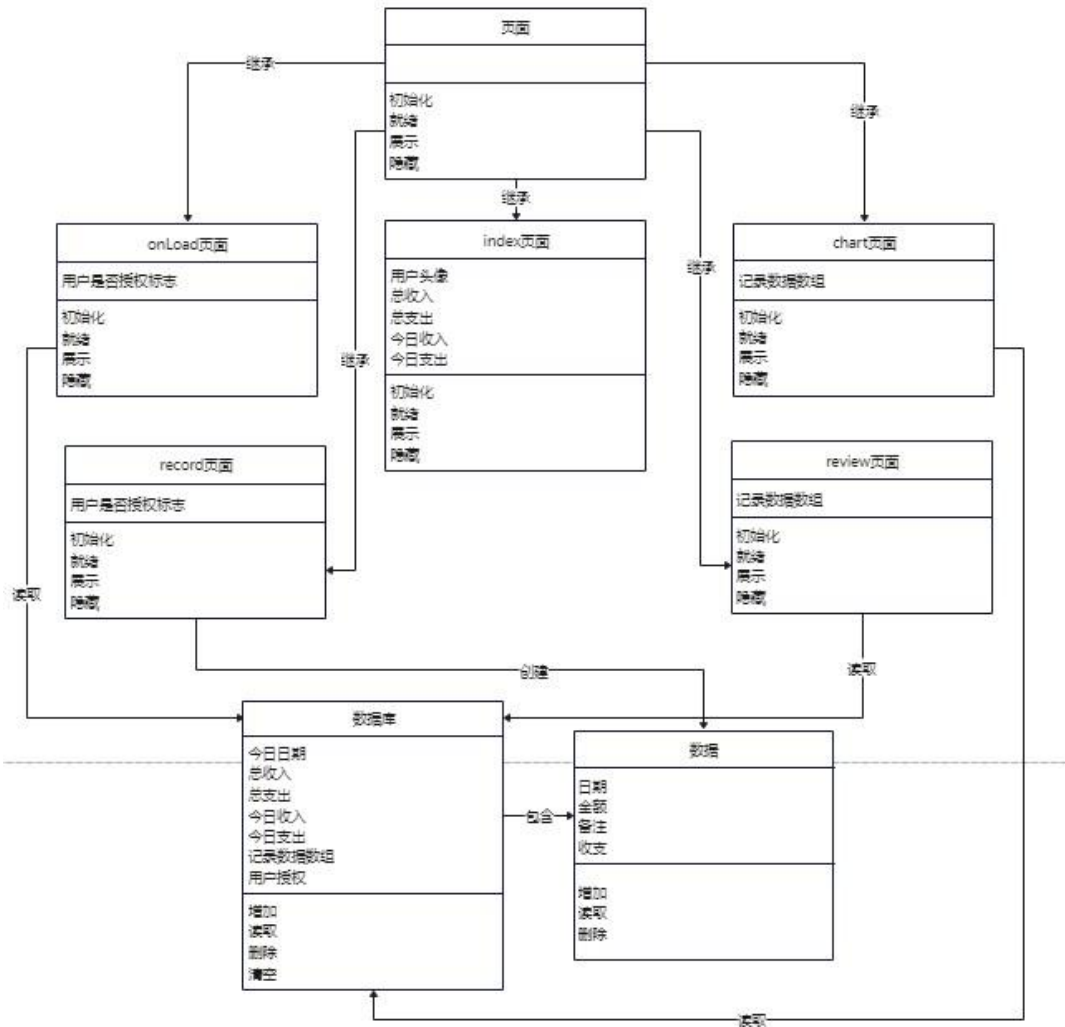


图 3-2 系统类图

### 3.3 关键数据结构定义

微信小程序实现记账功能，产生的数据主要为单次记录，账单，图形化数据，存储数据。主要的数据结构定义如下：

（1）单个记录：对于一笔账单记录，应当包含日期 `inputData`，备注 `inputBeizhu`，收支情况 `inputIndex` 以及金额 `inputPrice`。这四个属性构成一条基础的记录。

（2）账单：所有基础记录的汇总形成的列表构成总的账单，用于收支展示以及方便对某条数据的修改。

（3）图形化数据：对账单进行分类与合并，对于收支情况相同的记录划分为一类。随后，在各自的类中，将具有相同日期的记录进行合并。最终得到图形化数据。

（4）存储数据：在本地缓存以及云端存储中保存的数据，应当由用户唯一标识 `id`，今日日期，该用户账单记录，单日出支，单日出入，总出支，总收入组成。

### 3.4 关键算法设计

微信记账小程序主要由四个模块构成，下面分别分析展示模块，记录模块，回顾模块以及图表模块的主要算法设计：

#### 3.4.1 展示模块

本模块主要实现提供登录接口，读取用户信息，并将用户存储过的信息展示于主界面。其中关键为实现登录功能。

在用户打开微信小程序时，首先使用 `wx.login()` 方法，如果成功则代表此用户曾经授权，通过云端数据库中唯一用户表示 `openId` 获取用户信息，随后跳转到主界面；如果失败，则代表此用户并未授权，则为他打开授权窗口，并在授权后保存信息，同时跳转到主界面；登录过程流程图如图 3-3 所示：



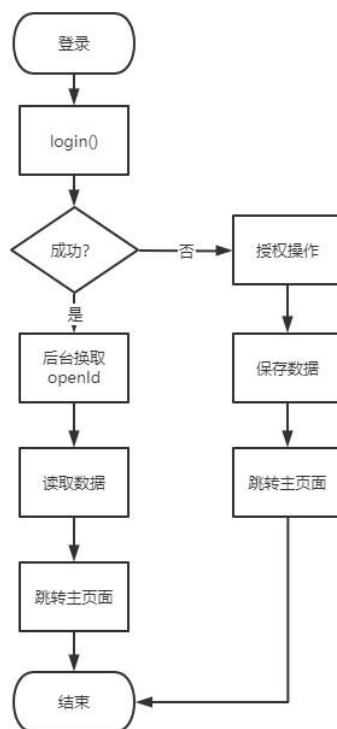


图 3-3 登录过程流程图

### 3.4.2 记录模块

主要提供记录收支信息功能。其核心为保存用户输入的数据。同时存储于本地以及云端。记录模块中的保存功能由 `post()` 实现。

首先对于输入的数据日期，金额，备注，收支情况进行处理。将日期选择得到的年，月，日合并，得到完成得的日期记录。同时，对日期进行判断，如果和今日日期相同，则今日收支数据做相应变化，否则总收支数据进行改变；对输入的金额进行转换保存，备注和收支情况保持用户输入结果。

将所有输入数据处理后，得到一条基本记录，将此记录并入账单列表中，用于后续处理；

对数据进行处理后，需要对数据进行存储。使用 `wx.setStorage(key,data)` 将数据 `data` 存储到标记为 `key` 的本地缓存中。同时，使用唯一标识用户 `openId` 构造数据，更新云端数据库 `wx.cloud.database` 中的数据；每一次数据存储后，更新当前程序中数据，保持同步。

`post()` 方法的主要流程如图 3-4 所示：

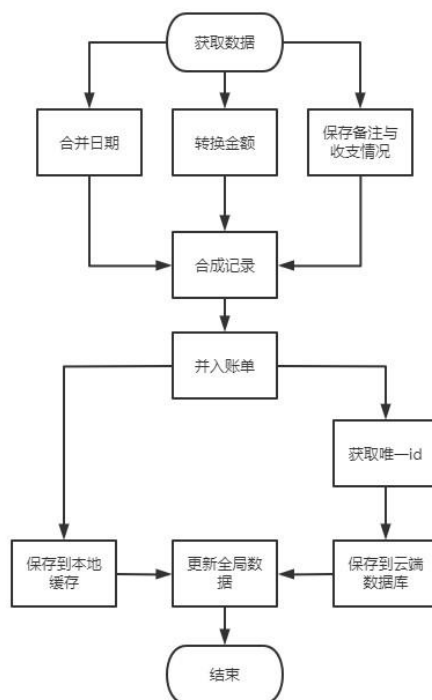


图 3-4 post()主要流程图

### 3.4.3 回顾模块

展示模块展示记录收支信息，包括记录日期，收支金额，收支类型和备注，同时提供数据删除功能。删除功能主要由 `remove()` 方法实现。

删除一个记录需要在账单中对其进行定位，由于每条记录由日期、金额、备注以及收支情况共同确定，故需要遍历整个账单列表，寻找到与选择删除记录相同的记录。

将此记录从本地缓存中删除，同时更新云端数据，删除此记录。随后对删除后的数据进行处理，即改变相应的今日收支以及总收支。保存所有改变的数据，更新全局数据，主要流程如图 3-5 所示：

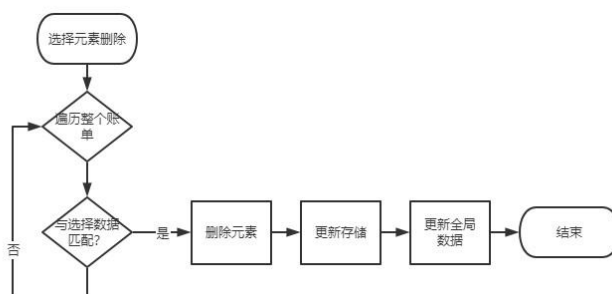


图 3-5 remove()算法流程

## 3.5 数据管理说明

微信记账小程序借助微信开发者平台实现，整体的数据被分为两个部分：本地缓存以及云端资源。云端资源主要存储，使用到的微信云开发函数以及云端数据库。本地缓存存储所有账单记录。

### 3.5.1 云端资源

云端资源中主要包含云函数，在记账小程序中，主要使用到微信用户登录 `login()` 函数以及获取用户标识 `getOpenId()` 函数。在进入小程序时调用 `login()` 函数以及 `getOpenid()` 函数，到后台换取 `openid`。

同时，在小程序的云端资源中，建立了用户数据存储的云数据库。云数据库的存储结构包括：数据记录 `id`，用户唯一标识符 `openid`，总收入 `all_in`，总支出 `all_out`，今日收入 `day_in`，今日支出 `day_out`，以及账单数组 `dataArray`。

`dataArray` 中存储着该用户的所有记录信息，每一条记录信息包含日期 `inputData`，备注 `inputBeizhu`，收支情况 `inputIndex` 以及金额 `inputPrice`。

整体的云端数据库结构如图 3-6 所示：

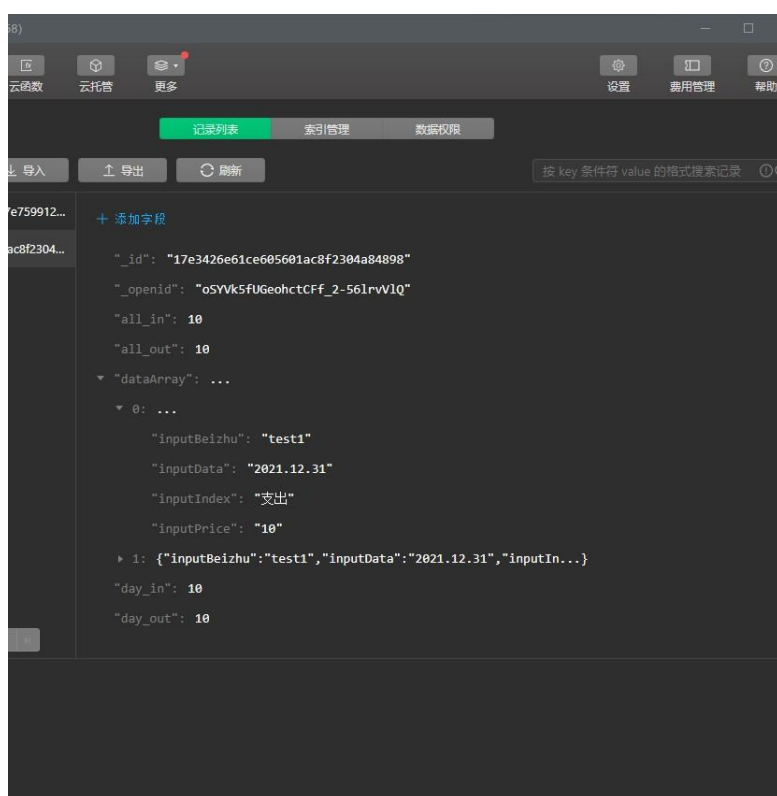


图 3-6 云端数据库存储

### 3.5.2 本地缓存

本地缓存主要存储用户近期的记录信息，本地缓存的大小为 10M，且有清除功能，会更新为近期的记录。本地缓存中数据的存储形式与云端数据库的存储基本相同，但是由于保存在本机，不需要与其他用户进行区分，故不设置标识符 `openid`，而是保存一个用户是否有数据的标记 `used`。

存储结构包括：用户是否有数据的标记 `used`，今日日期 `currentData`，总收入 `all_in`，总支出 `all_out`，今日收入 `day_in`，今日支出 `day_out`，以及账单数组 `dataArray`。

`dataArray` 中存储着该用户的所有记录信息，每一条记录信息包含日期 `inputData`，备注 `inputBeizhu`，收支情况 `inputIndex` 以及金额 `inputPrice`。

### 3.5.3 数据读取使用

对于微信小程序数据的读取，优先读取本地缓存，由于本地缓存的读取速度更快，能给用户带来更好的体验。在进入微信小程序时，会从云数据库获取 `openid`，获取成功则从本地缓存读取总收入 `all_in`，总支出 `all_out`，今日收入 `day_in`，今日支出 `day_out`，在主界面进行总体展示。获取失败表面他是新用户，会在授权后，于云端数据库保存 `openid`。

再回顾和图表界面，读取本地缓存账单数组 `dataArray`，获取所有数据，进行展示。

在存储时，需要同步更新云端数据以及本地数据。在本地缓存中选择替换旧数据的方法，而在云数据库中选择添加的方法。故可分别产生近期数据与全部数据，方便用户的使用。说明整个系统所涉及到的数据用何种方式进行存储和访问。

## 4 实现与测试

### 4.1 实现环境与代码管理

#### 1. 软件实现环境

本项目基于微信小程序开发工具实现，编程语言为 javascript, wxml 与 wxss。由于微信小程序的特点，开发者的软硬件环境对软件的开发不构成影响，因此不对软硬件环境进行说明。

#### 2. 代码管理

本项目的仓库地址为 <https://gitee.com/luyunlong001122/software>。

代码签入记录如图 4-1 所示，总提交次数为 59 次，其中组员 提交 31 次，组员 提交 20 次，组员 提交 7 次，组员 提交 1 次。组员 主要负责系统总体设计、系统实现与代码测试；组员 主要负责登录模块设计、UI 设计和反馈收集处理；组员 主要负责目录模块、图表模块的设计和系统测试；组员 主要负责记录与回顾模块设计和进度计划。



图4-1 码云平台代码签入记录

### 4.2 关键函数说明

微信小程序与网页的展示类似，每个网页均可视为独立的模块。本项目的小程序主要有 5 个不同页面，即 5 个模块组成。其中，onload 页面实现用户信息的获取和用户授权确认；index 页面实现统计信息的展示和其他功能的跳转；record 页面实现记账的主要功能，即记录账目的支出/收入相关信息；review 页面实现

账目的依次展示和删除条目功能；chart 界面实现统计和图形化展示的功能。

#### 4.2.1 onLoad 界面

功能：生成登录页面，提供授权登录按钮，获取用户登录信息，能根据用户点击情况实现界面的跳转。

实现：进入登录页面，当用户点击允许授权按钮后，获取用户信息，使用微信 `wx.navigateTo` 函数跳转到 `index` 界面。

首先获取当地时间信息，赋值到全局变量 `app.globalData.currentData` 中；

使用函数 `wx.login`，发送 `res.code` 到后台换取 `openId`, `sessionKey`, `unionId`；

`wx.getSetting` 函数用来判断用户状态信息。若 `app.globalData.used` 为 1，则表明当前用户为老用户，可将本地数据导入；

`wx.getStorage` 函数用来给数组 `dataArray` 赋值，用于获取用户的账单信息；

`wx.getUserProfile` 函数用于获取用户基本信息，如头像等；

当用户点击允许授权后，打印到控制台，将头像信息赋值给全局变量 `app.globalData.headPic`；

跳转到 `index` 界面。

#### 4.2.2 index 界面

功能：生成主页面，显示统计信息和用户头像，并能够根据用户的点击情况实现界面的跳转。

实现：在登陆界面完成授权后，使用微信 `wx.navigateTo` 函数跳转到 `index` 界面。

若为新用户，则全局变量置零，展示所有新数据

若为已授权用户，则使用 `wx.getStorage` 函数从本地缓存中根据 `key` 值读取数据，存储至全局变量中，将全局变量的数据进行展示。对应关系如表 4-1 所示：

表 4-1 全局变量关系表

全局变量	对应展示位
app.globalData.headPic	用户头像
app.globalData.day_in	今日收入
app.globalData.day_out	今日支出
app.globalData.all_in	总收入
app.globalData.all_out	总支出

同时绑定图片与三个界面的跳转函数，使得其他页面得以正常工作。在每一次从其他页面返回当前页时，若数据发生更改，刷新界面以展示最新的数据。

### 4.2.3 chart 界面

功能：读取记账模块保存的数据，统计每个日期的总收入和总支出，并据此生成以日期为横轴、总金额为纵轴的柱状图。

实现：读取本地存储的 `app.globalData.dataArray`，其中包含着所有保存的账单记录。遍历所有记录，统计所有的日期，生成不重复的收入日期字典 `dataIncome` 和支出日期字典 `dataExpend`。

遍历 `app.globalData.dataArray`，读取记录中所有收入项，对应日期的数据更新为原数据加上收入金额，最终得到每各日期对应的总收入数组 `income`；

遍历 `app.globalData.dataArray`，读取记录中所有收入项，对应日期的数据更新为原数据加上收入金额，最终得到每各日期对应的总支出数组 `expend`；

根据字典 `dataIncome` 和收入数组 `income` 绘制收入柱状图，根据字典 `dataExpend` 和支出数组 `expend` 绘制支出柱状图。

在每一次进入当前页时，若数据发生更改，刷新界面以展示最新的数据。

### 4.2.4 record 界面

功能：获取用户输入的时间、备注、价格、支出/收入类型等记账信息，生成账单条目记录在本地缓存中。

实现：用户点击主界面的“记账”图片后，使用 `wx.navigateTo` 函数跳转到 `record` 界面。

用户记帐时，调用函数在 `record.data` 中保存用户此次输入的日期、备注、价格以及选择的收支类型

`record.data.array` 中保存了可供用户选择的收支类型：“收入”、“支出”。当用户点击“当前选择：”时，会弹出界面让用户进行选择，同时修改 `record.data.index`，`index` 为 0 时表示支出，为 1 时表示收入。

当用户输入完毕，点击“保存”时，会根据 `record.data` 生成一条账单记录保存在 `app.globalData.dataArray` 中。

保存完毕后，判断用户选择的收支类型，以及用户输入的日期是否与当前系统时间日期相符合。根据判断结果，系统将 `record.data` 中的价格信息强制转换为数值，对 `app.globalData` 内的对应全局变量进行累加。

当用户点击“清空”时，清空输入行，并将 `record.data` 还原为默认值，使用户可以重新输入数据，再次记账。

#### 4.2.5 review 界面

功能：读取用户目前为止记录的账单数据，根据记录时间顺序条目式地显示在界面中，用户可以浏览历史记账记录，或是删除不正确的记录。

实现：用户点击主界面的“账单”图片后，使用 `wx.navigateTo` 函数跳转到 `review` 界面。

进入 `review` 界面时，使用 `wx:for` 循环体逐条读取 `app.globalData.dataArray` 中保存的历史账单记录，加载到页面中。

当用户点击某条记录的“删除”按钮时，获取触发删除事件的条目保存在 `res.currentTarget.dataset`，并将该条目从页面中清除。

遍历 `app.globalData.dataArray`，得到与 `res.currentTarget.dataset` 相符合的记录，使用 `splice` 函数将该记录删除。同时，判断该条目的日期和收支类型，对对应的 `app.globalData` 中的全局变量进行累减。

。

### 4.3 测试计划和测试用例

测试策略根据测试活动中是否执行软件区分，可以分为静态测试和动态测试；



按测试用例的设计思路区分，可分为白盒测试与黑盒测试；按测试时是否自动来区分，可分为手工测试与自动测试。其中，白盒测试的常见方法包括：逻辑覆盖法、路径测试法。黑盒测试的内容包括 Alpha/Beta 测试、菜单、帮助测试、发行测试、回归测试、效能测试、负载/压力测试。黑盒测试法是根据被测程序功能来进行测试，也称为功能测试，有 4 种常用技术：等价分类法、边界值分析法、错误猜测法、因果图法。测试计划分为对流程的测试和对功能的测试。

由于微信小程序的特点，我们难以对各个函数进行逐一测试，因此我们采用黑盒测试的方法，去测试每个页面是否能够正确实现预期的功能。

由于此小程序由五个模块构成，因此对这五个模块均进行一定的测试。

onload 模块主要功能是从微信获取用户授权信息，并当用户确认授权时跳转至 index 页面。因此，测试用例只需要包括两种情况，即用户点击确认按钮和用户点击拒绝按钮的情形即可。测试样例与测试结果如表 4-2 所示。

表 4-2 onload 页面测试表

测试样例	预期结果	实际结果
用户点击“授权登录按钮”	弹出微信授权提示框	
用户在微信授权提示框中点击“允许”按钮	跳转至 index 页面	正确跳转至 index 页面。
用户在微信授权提示框中点击“拒绝”按钮	提示框关闭，返回 onload 页面	提示框正确关闭，继续展示 onload 页面。



index 模块的主要功能是展示用户所记载内容的统计信息以及实现页面的跳转。因此测试内容包含两个方面，一是能否正确显示统计信息，而是能否正确实现页面的跳转。测试样例与测试结果如表 4-3 所示。

表 4-3 index 页面测试

测试样例	预期结果	实际结果
用户尚未记载任何信息	首页展示信息均为 0	
2021.12.30 支出 35 2021.12.30 收入 30 2021.12.31 支出 25 2021.12.31 收入 18	今日收入 18 今日支出 25 总收入 48 总支出 60	
用户点击记账按钮	跳转至 record 页面	跳转正确
用户点击账单按钮	跳转至 review 页面	跳转正确
用户点击图表按钮	跳转至 chart 页面	跳转正确
用户点击清除按钮	清除所有记载信息	清除正确

record 模块的主要功能是根据用户填写的相关内容生成账目信息并加以保存。同时设置有复位按钮来清除用户在输入框内的输入内容，并将收入/支出标签重置为“支出”。则测试内容主要包括测试能否正确记录收入/支出信息以及复位按钮能否正确实现预期功能。测试样例与测试结果如表 4-4 所示。

表 4-4 record 页面测试

测试样例	预期结果	实际结果
2021.12.20 支出 35 2021.12.20 收入 30	保存正确	保存正确
	用户输入框被清除，标签重置为“支出”	
用户点击清除按钮		清除正常

review 模块的主要功能是逐项展示所记载的内容，和提供删除选项用于删除一条记载内容，删除后在存储中将相关条目清除，因此 index 页面内的统计信息

也会随之改变。测试内容主要包括能否正确展示记载过的内容以及删除功能是否正常。测试样例与测试结果如表 4-5 所示。

表 4-5 review 页面



测试样例	预期结果	实际结果
2021.12.30 支出 35 2021.12.30 收入 30 2021.12.31 支出 25 2021.12.31 收入 18	正确展示所记载的四条内容	
删除第一条内容	删除正常，且 index 内信息更新，“总支出”变为 25	

chart 模块的主要功能是将记载的内容进行统计并以柱状图的形式展示。只需测试能否正确展示即可。测试样例与测试结果如表 4-6 所示。

表 4-6 chart 页面

测试样例	预期结果	实际结果
2021.12.30 支出 35 2021.12.30 收入 30 2021.12.31 支出 25 2021.12.31 收入 18	能够正确展示统计内容	
添加一条信息： 2021.12.31 收入 12	相同日期的信息能够正确合并，2021.12.31 的收入变为 30	

## 4.4 结果分析

由上述各测试结果来看，小程序的各个页面都能够正确实现预期的功能，可以正确实现记录和删除，同时各个页面均能随操作进行更新，各页面设置的功能按钮能够正常发挥作用。小程序实现了设计目标，即能够实现简单的记账功能并提供图形化展示界面，同时小程序的实现方式保证了系统的易用性和系统的轻量化。

## 5 总结

### 5.1 用户反馈

当第一代记账小程序发布后，向周围的好友推广了我们的小程序。这些用户在一时间的使用后，为我们反馈了一些使用截图，如图 5-1 所示。



图 5-1 用户使用截图

另外，我们也收到了来自这些用户的反馈，我们选取了两条用户反馈信息。如图 5-2 和图 5-3 所示。

记账程序使用小程序的方式，使用起来比较方便，界面简洁易用，图形化的展示一目了然。但是账单回顾界面缺少排序功能，希望能够改进。

图 5-2 用户反馈 1

作为一款日常记账软件，界面很好看，简洁明了，很可爱。不过在进行填写日期操作时，过于繁琐，希望选择类似可以选择的日期的那种，再接再厉！！

图 5-3 用户反馈 2

在整合用户反馈后，我们进行了规整，我们的程序主要有以下优点：

1. 没有广告，界面简洁。

2. 一键清空功能能够让过多的记录清除，避免了逐步删除的繁琐。
3. 图形化界面很直观。

同时我们的程序也有些许不足，主要有以下几点不足：

1. 填写日期时希望提供选项。
2. 希望能有修改功能。
3. 记账模式单一，希望有更多社交功能。

根据以上用户的反馈，我们对日期选择功能进行了优化，并对界面进行了优化调整。并考虑在之后的版本上线社交功能。

## 5.2 全文总结

### 总结

在软件工程项目实现的过程中，小组四位成员讨论分配了任务，通过完成各自的任务，我们合作完成了微信记账小程序。

作为小组的一名成员，我主要负责项目系统设计，系统完整实现，以及代码测试等工作，同时负责小部分部分文档编写，我在本次项目的具体工作如下：

- （1）完成了对于项目背景和可行性的调查和分析，编写了 NABCD 模型初稿，交由其余同学补充。
- （2）完成原型系统的架构，初步完成了原始的界面设计和实现。
- （3）完成系统结构的设计，编写代码实现微信小程序的基本框架。
- （4）与其他同学合作，具体实现每个模块。

### 总结

作为记账小程序小组的 UI 设计者和界面交互设计者。主要负责页面 UI 设计和页面跳转逻辑关系设计，打造一个便捷快速的记账交互模块。并且负责了部分的登录界面有关设计。同时，作为该应用的需求分析者，询问了各个群体对记账方式的看法和需求，并且提取了记账小程序的需求方式。最后，进行用户反馈收集和整合，反馈给团队，进行后续的版本迭代。具体工作如下：

- （1）完成了对原型系统中的不足进行调整。

- (2) 完成了用户记账需求收集和整体需求分析。
- (3) 完成了各页面的跳转逻辑设计。
- (4) 完成了各个页面的 UI 设计和交互设计。
- (5) 完成了用户反馈信息的收集和整合。
- (6) 进行了登录授权界面的设计与部分实现。
- (7) 进行了程序迭代页面的调整。

### 总结

在本次项目开发中，我主要负责项目前期的需求分析以及进度计划安排，项目中期的记录模块与回顾模块设计等工作，在项目文档的记录中我负责了进度计划以及系统实现的记录回顾部分，具体工作如下

(1) 完成了对于项目基本目标以及项目可行性的分析，制定了立项阶段最终的项目方案。

(2) 对项目开发、测试以及维护的工作任务进行了合适的分工，并制定了项目进度计划以确保项目的按时完成。

(3) 完成了记账小程序的账单记录功能以及回顾功能等核心模块的实现，为小程序的互动界面提供了脚本支持。

(4) 完成了项目文档中进度计划的安排以及系统实现中部分功能的原理。

(5) 完成了课程报告中的项目问题定义一节。

### 总结

在本次项目开发中，我主要负责项目的目录模块与图表模块的开发，以及编码完毕后的系统测试工作。在文档的记录中我负责了实现与测试文档的编写。具体工作如下：

- (1) 实现数据可视化功能，即利用保存的数据绘制柱形图。
- (2) 对编写的小程序进行功能测试，发现存在的问题和进行错误反馈。
- (3) 从用户的角度体验程序，提出改进建议。
- (4) 完成了项目文档中实现与测试部分内容的编写。
- (5) 完成了课程报告中实现与测试一节。

## 6 体会

，分工：系统设计，系统完整实现，代码测试。

体会：

在本次软件工程课程以及任务的完成过程中，我第一次参与了软件的设计，实现，测试等流程，最终通过合作完成了一个完整的微信小程序项目。在这一过程中，我遇到了许多困难，也有了许多收获。

在项目开始时，我们很快按照老师的要求完成选题，但由于大家第一次进行软件的创造，缺乏实现任务所需要的技能和知识，整个任务的一直停滞不前。我们小组通过商讨，决定一步步完成我们的任务。在开始时，虽然进度落后，但是大家还是在努力学习相关的知识，从基础的 JavaScript，到 git 提交代码，微信小程序开发知识等等。也正是这一段时间的学习，让我们有能力能够逐步实现我们的软件。

在项目的实际开发阶段，我的任务是负责系统的设计，以及整体框架的实现。在我完成具体的框架后，需要与其他同学合作，实现每一个具体的模块。在大家都是初次使用 JavaScript 以及微信开发者工具的情况下，项目实现的进度较为缓慢。随着每个人学习的知识渐渐丰富，我们的进度也越来越快，直至最终按时完成了任务。

软件的实现让大家很开心，但随之而来的调试，暴露出了许许多多的问题，甚至发现了初期设计的错误，不得不推倒重来。所幸，由于之前的经验，再次编写程序不再那么生疏，很快完成了软件的基本功能实现。

完成任务后，我们发布了微信小程序，得到了一些反馈，同时，也在不久获得了助教的建议。在讨论过后，我们继续完善了原先的微信小程序，使得它的功能更加完善，界面也更加讨喜，增加了特色。

在各个阶段我都遇到了一些难题，为了解决它们，我学习了许多新的知识。现在看来，整个项目带给我很多收获。我学会了使用 JavaScript，以及使用微信开发者工具，学习了软件工程中相关的知识。最重要的是，我认识到了团队合作的力量，它能使我完成个人所不能完成的任务。

最后，感谢一起努力的成员们，感谢大家共同完成了这样一个项目，也感谢老师和助教给我们的帮助。



，分工：登录模块设计，UI 设计，反馈收集与处理。

体会：

“我们的小程序终于发布啦！”这是最终结果带给我无比大的喜悦。在这 2 个月的时间里，我们从组队到建立模型，再到逐步完成整个系统的搭建，再到接收反馈，不断调整界面，不断修改 bug。这一步步走来十分的艰辛，而且中间还时不时出现考试打断预期的规划。不过值得的是，最终我们发布了记账微信小程序，这带给了我巨大的鼓舞。在收到朋友使用的反馈时，他们都投来了好奇的目光，都啧啧称奇，让我觉得我们的方向选的很正确，记账的空缺确实很大。这也让我下定决心用心继续维护后续版本，毕竟做出来的第一个可以称作能够实用的东西真的很具有纪念意义。

艰辛路途的第一重困难就是，由于我之前只写过小规模的程序，因此对这么大的工程有些恐惧，但由于组长的代码能力很强，这让我十分安心，并且还借助这次机会，学习和熟悉微信小程序的开发，熟悉了微信开发集成平台中 wxml 和 wxss 的编写。试着完成了登录界面的设计和有关 js 代码的实现。

第二重困难则是，界面设计也是一门技术活。就我全权负责的需求分析和界面 UI 交互设计，我体会颇深。一个小程序别人喜不喜欢用，关键在用的舒不舒心。若一个小程序都是广告，并且界面没有明显的指引，是很难吸引用户的。故在界面设计上花了不少功夫，通过对原型系统的修改，调整了不少功能。如在回顾界面的规划板块，在询问了很多用户群体后，发现并没有太大的需求。除此之外，主页面对用户而言也需要更为简便的展示，因此将主页面中的最近收支情况进行了删除。在考虑一部分人对可视化信息的需求和一键清空的需求，我们增加了两个功能，表格和清除功能，来补充用户需求。

通过这些设计，我也真正体会到了界面设计师、前端工程师的不易，他们常常需要花费很多时间来处理一些细小的设计。比如按钮在页面中的位置，由于机型的不同，常常需要设计一个适合大部分主流机型了界面，因此，需要对背景进行许多细微的调整，避免导致较差的界面效果。这一环节十分考验对美学的把握，我作为队内唯一一位做过海报设计的同学，担起了这一重任。

另外，还遇到了画图的难题。在绘制各种 UML 图时，需要花费大量时间学习图的具体表现形式，还要根据用户需求进行联系和表现。当系统只是在一个初级阶段时，该过程是十分考验归纳和思考能力的。不过在画完那些图后，也确实对我们的记账系统有了体系化的认识，明白了我们需要哪些数据，需要完成哪些功能，功能间的调用关系等。这对之后的系统设计与代码实现有非常大的作用。这让我明白了一点，大工程下需要使用图来分析系统的组合和实现，各种 UML 图在

大项目中是不可或缺的。

最后，我也想感谢我的 3 位队友。我们一起在这个项目中一步步前行，一个大软件工程项目如果要做好，单靠个人的能力是不可能完成的。并且团队内的配合十分重要，我为认识了这样默契十足的队友感到开心。

，分工：记录模块与回顾模块设计、进度计划。

体会：

本次软件工程项目是我第一次实际参与到的软件开发工程，在各个阶段都遇到了许多的困难，但最终也得到了许多收获。由于是初次参与软件开发，我在各个方面都极其缺乏经验，也因此碰到了许多的挫折，有些问题可能在结束之后看来显得有些可笑，但这些困难恰恰铸就了我的成长。

在项目的初期，我就因为没有经验而重重受挫。首先，由于我与其他 3 位成员不住在同一寝室，彼此之间严重缺乏沟通。由于本次项目是需要所有成员通力协作来完成的，这让无法第一时间参与到其他 3 人的交流中的我显得十分被动。好在小组成员之间都较为友善，通过 qq 群加强了彼此的交流，并进行了明确的任务分工，以免单人开发时摸不清方向。其次，由于过去的我从来都是单人独立完成程序代码的编写，从未使用过分布式版本控制系统 Git，更加不了解如何实现各个成员负责的功能模块之间的对接和版本管理。因此，在分配任务后，我对于如何进行自己的开发任务一头雾水。好在小组成员对如何使用 Gitee 码云平台进行代码托管做了指导，并且对于如何使用 JavaScript 对 HTML 进行脚本支持进行了详细的说明。

在实际开发阶段，由于是初次制作微信小程序，以及初次使用 JavaScript 语言编写程序，我对于如何开发实现自己负责的功能模块非常的迷茫。好在根据其他小组成员的指点，我找到微信官方所提供的小程序开发文档和指南。通过微信官方文档，我认识到微信开发者工具中内置了一系列分装好的功能模组，例如用户账户获取、本地数据存储等。利用官方提供的组件，我顺利完成了记录模块与回顾模块的可互动页面设计。但在功能脚本开发的方面，由于从未使用 JavaScript 这类解释型的语言，我对于它的语法以及使用方法十分不能理解。因为这个原因，我在编写 js 脚本时，出现了各种各样的 bug，大多数甚至是低级的语法错误。好在小组中有成员对如何使用 JavaScript 较为熟悉，在有他们的指导、

帮助 debug 的情况下，我总算是顺利地在规定时间内完成了自己地开发任务。

最后在项目文档的记录阶段，由于我从未接触过 UML 这类大型项目开发需要用到的模型，一直都是独自一人完成程序就算大功告成，因此在完成自己负责的文档记录阶段也遇到了不少困难。我负责的是进度计划的编排以及部分的需求分析，在此之前，我从未接触过甘特图这类分工计划安排图，因为不需要。因此在分工的安排与时限的设置上极其缺乏经验，既要考虑到各个成员的能力以及意愿，又要关注项目完成的期限以及各个阶段需要的时间，让我一时间难以抉择。好在经过小组成员间的沟通交流，我们通过协商最终达成了一个尽量满足所有要求的分工计划方案。

尽管在项目的各个阶段都遇到了不少困难，通过克服这些困难，我也获得了许多新的收获。首先，我重新认识到了团结一致、互帮互助的团队对于项目开发的重要性，认识到了个人的力量是有限的，而团队间的协调分配可以帮助每个成员扬长避短；其次，我学会了新的程序开发语言以及程序协作开发的方法，学习到了非常多的撰写文档的技巧与工具，认识到了文档记录对于协调团队合作、回顾项目历程的重要性；最后，我认识到了自己的许多不足之处，不仅是程序开发能力上的知识、技巧欠缺，还有团队沟通合作上的没有主见、不善交流等缺点，这将会激励我向着更好的自己发展。

总而言之，在本次项目的过程中我收获良多，这要归功于我的组员们对我的支持与鼓励，尤其是 同学在程序开发上的指导帮助。同时，我还要感谢老师对于我们应当如何进行软件工程项目开发的教导，让我们拥有这次机会体验实际的软件开发，提升自我。

，分工：目录模块与图表模块设计，系统测试。

体会：

图表模块的设计是小程序设计中较为复杂的一部分。但得益于 github 上已有的开源内容，我们可以直接利用现有的源代码来方便地实现，使我体会到开源项目对一些软件开发提供的巨大助益，激励我参与到开源项目的构建当中。

系统测试是我负责内容的主要部分。在系统测试过程中，我采用黑盒测试的方法，利用开发工具的真机测试功能进行测试，通过模拟正常用户的使用来测试

系统实现的正确性，也借此发现和帮助解决了不少问题，例如日期格式不统一的问题（后来改进为用户选择日期而不是直接填写）、可视化页面在灭屏、亮屏后数值会翻倍的问提等。此次测试过程使我从开发者的角度认识到软件测试的重要性。设计良好的测试用例，不仅能够发现软件代码中的问题，更有助于发现软件错误的原因和修正方法，因此系统测试在软件开发过程中是必不可少的。

本次开发是我第一次参与基于 git 的多人协同开发，虽然学习使用 git 花费了一定的时间，但 git 的使用极大的方便了对开发的管理，方便我们实现代码的合并和版本控制。另外，模块化的编程也为我们的开发提供了巨大的便利。我们不需要关心其他人负责的模块的实现方式，而只需相关内容进行调用即可，让我感受到模块化编程的重要作用。

微信小程序的主要开发语言为 JavaScript，WXML 与 WXSS。JavaScript 是我之前从未接触过的编程语言，WXML 和 WXSS 虽然与大一接触过的 HTML、CSS 类似，但那时我使用的是 dreamware 软件进行操作，而未深入到代码层面进行编辑，因此对我来说也较为陌生。好在微信小程序官方提供了详细的文档供我们阅读，网络上也有大量相关教程，让开发得以顺利进行下去。

在课堂上，我们接触了用例图、UML 图、甘特图等一系列工具，这些工具的使用让我们在编码工作开始前就对程序的结构和设计思路有了一定的认识，让我们在正式编码时不用花费过多精力在考虑代码组织结构和调用关系等问题上，大大提高了工作效率。使我认识到完善的项目分析和规划在大型项目开发上的重要作用。