

**《软件工程》项目报告**

**题目： 想记就记——面向大学生的记账软件**

**课程名称： 软件工程**

**专业班级： CS2001**

**同组成员： 刘玺语 U202015299**

**吕嘉乐 U202015318**

**毛俊棋 U202015310**

**张恒铭 U202015322**

**指导教师： 范晔斌**

**报告日期： 2022年11月30日**

**计算机科学与技术学院**

**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**

[1 问题定义 1](#_Toc120655710)

[1.1 项目背景与意义 1](#_Toc120655711)

[1.2 项目基本目标 2](#_Toc120655717)

[1.3 可行性分析 2](#_Toc120655718)

[1.4 人员管理和项目进度管理 2](#_Toc120655719)

[2 需求分析 3](#_Toc120655722)

[2.1 E-R图、数据流图 3](#_Toc120655723)

[2.2 用例图 4](#_Toc120655724)

[2.3 原型系统设计 4](#_Toc120655725)

[3 概要设计和详细设计 8](#_Toc120655729)

[3.1 系统结构及类图 8](#_Toc120655730)

[3.2 关键数据结构定义 12](#_Toc120655731)

[4 实现与测试 14](#_Toc120655735)

[4.1 实现环境与代码管理 14](#_Toc120655736)

[4.2 关键函数实现 14](#_Toc120655737)

[4.3 测试计划和测试用例 15](#_Toc120655738)

[4.4 结果分析 21](#_Toc120655741)

[5 总结 22](#_Toc120655742)

[5.1 用户反馈 22](#_Toc120655743)

[5.2 全文总结 22](#_Toc120655744)

[6 体会 23](#_Toc120655745)

[6.1 刘玺语 23](#_Toc120655746)

[6.2 吕嘉乐 23](#_Toc120655747)

[6.3 毛俊棋 23](#_Toc120655748)

[6.4 张恒铭 24](#_Toc120655749)

# 问题定义

## 项目背景与意义

我们小组在第一次小组讨论时选择开发一款针对大学生的轻量化记账软件，以下为该项目的相关分析。

### 用户需求

在数字化支付逐渐代替传统支付方式的大背景下，人们对财务的管理和支配意识越来越迟钝，这种现象在以大学生为主的年轻人群体中尤为明显，而传统的纸质账本的记录由于有诸多不便，而作为一款手机记账软件，不仅解决纸质账本记录的繁琐麻烦等问题，同时加强了大学生的理财意识。

### 解决方法

询问并分析大学生对记账软件的基本要求，简化记账流程，减少不必要的界面，只保留必要功能，同时能分析用户的收支记录并生成可视化图表，让用户更容易了解到自己的消费习惯。

### 产品优势

针对大学生开发，界面简洁美观，去除不必要的话题、新闻等界面，拥有一个更透明省心的体验；交互逻辑直观自然，节约记录时间，避免用户因花费时间记账而分散注意力；有助于让用户更了解自己的消费习惯，从而进行控制。

### 市场竞争

市面上已有许多记账软件，但大部分的使用人群都是已经工作的社会人士，而忽视了人口占比同样庞大的大学青年人群；界面过于花哨，集成了许多其他类型软件才有的功能，极大程度地浪费软件空间，体验并不算好；记账时需要设置的条目复杂繁琐，降低用户的记账积极性。

在这样的市场环境下，一款针对大学生的记账软件，保持简单直观的记账功能的前提下又没有过多的冗余功能，相信一定有不小的竞争力。

### 产品推广

先从本学院开始宣传，逐渐扩展到全校，同时广泛收集反馈意见，完善产品功能。

## 项目基本目标

考虑到使用场景和便携性，我们小组决定用手机APP的方式实现本课题，通过Github仓库对我们的项目文件进行版本管理，初期目标是实现通常记账软件的基本功能，分析记账流程并进行简化，最后对界面进行适当的美化。

## 可行性分析

通过上述的分析可知我们的课题在市场方面有一定的需求空间，而在实现方面，通过安卓设计软件结合之前学习过的java编程知识，虽然有一定的学习成本，但还是有可行性的。

## 人员管理和项目进度管理

### 人员管理

刘玺语：后端数据库操作实现

吕嘉乐：前后端接口交互实现

张恒铭：前端页面设计&测试

毛俊棋：文档撰写&原型设计

### 项目进度管理

项目进度如图 1.1所示。

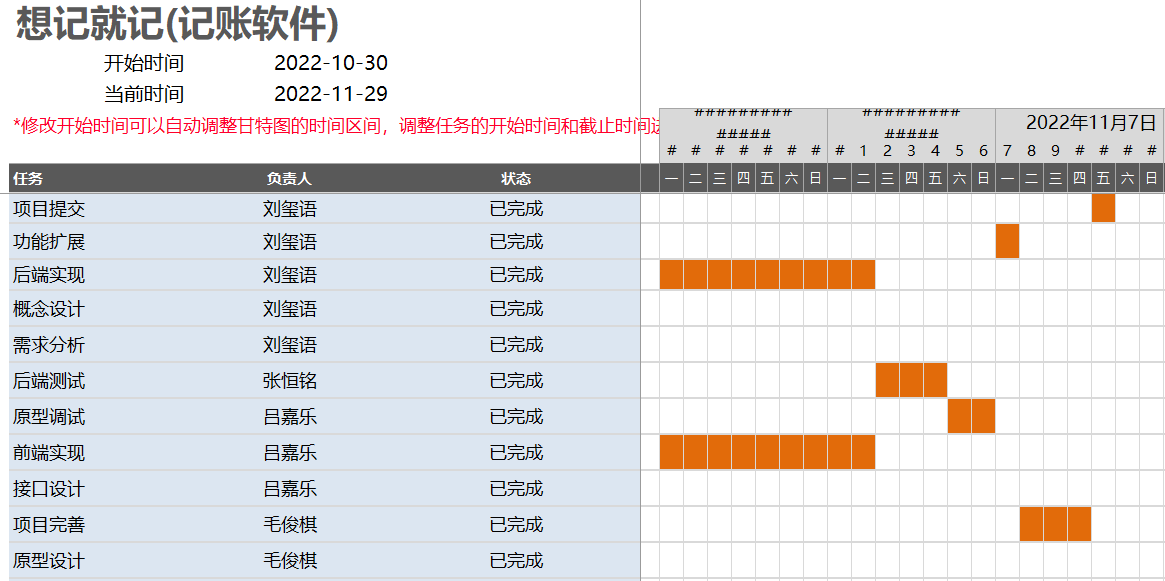


图 .1 项目进度甘特图

# 需求分析

## E-R图、数据流图

本课题的E-R图如图 2.1所示，数据流图如图 2.2所示。

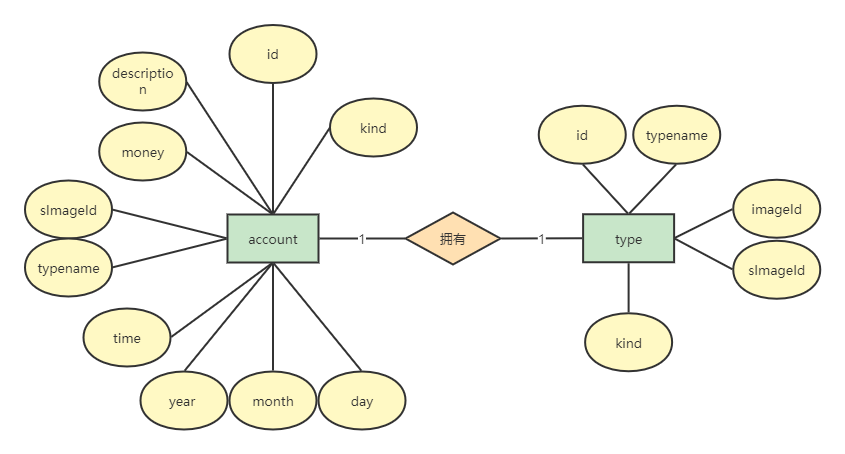


图 .1 系统E-R图

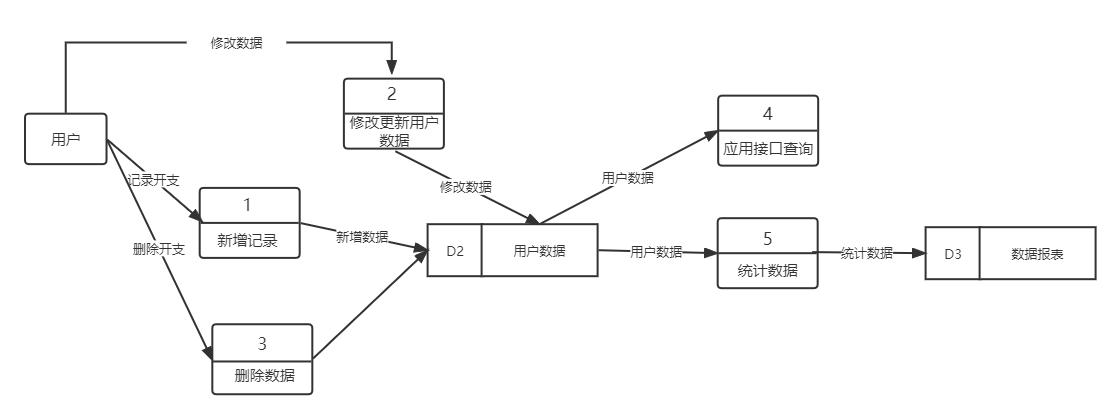


图 .2 系统数据流图

详细的说明在[概要设计和实现说明](#_概要设计和详细设计)中给出。

## 用例图

本课题的用例图如图 2.3所示。



图 .3 系统用例图

## 原型系统设计

### 设计工具

原型基于墨刀企业版设计。

### 设计思路

由前面确定的产品定位，按Android界面为基础设计各个页面，其中包括集成所有功能的主页面，显示收支记录的统计页面，修改或新建收支记录的记录页面，搜索收支记录的搜索页面等。

### 设计成果

原型系统效果详见网页：<https://modao.cc/app/pYmUQHPtriph59XDcPdj>

主页面如图 2.4所示。顶部的标题栏提供菜单入口（左侧）和搜索入口（右侧），下方显示的是本月的收支概览，其中的“图表分析”按钮可跳转到统计页面，再往下就是当天的收支记录，以列表的方式依次堆叠，底部的“+”按钮提供添加收支记录的入口。

菜单页如图 2.5所示。由于功能较少，采用从左侧插入的方式展示，其中初步设计三个入口，分别对应关于页、统计页和设置页。

|  |  |
| --- | --- |
| 图 .4 主页面 | 图 2.5 菜单页 |

搜索页如图 2.6所示。顶部分别是返回主页面的返回按钮，键入关键词的搜索框和搜索按钮，搜索结果将同主页面一样以列表的形式呈现。

统计页如图 2.7所示。顶部左侧是返回主页面的返回按钮，右侧是更改统计日期范围的设置按钮，下方通过切换“支出”和“收入”选项分别展示设置日期范围内的收支表。

|  |  |
| --- | --- |
| 图 .6 搜索页 | 图 2.7 统计页 |

记录页如图 2.8所示。顶部左侧为取消添加/修改操作的取消按钮，中间为修改该记录为收入还是支出的选项卡，下方表格为各个收支的分类选项，其中收入和支出有所区别，再下方则是输入该记录备注信息和金额的输入框，底部则是软件内自带的数字键盘，专用于输入金额，备注信息则使用系统软键盘。

设置页如图 2.9所示。顶部还是具备返回主页面的返回按钮，中间暂时未设计更具体的功能，只有一个清空用户记录的按钮，其他功能会在后续维护中跟进。

关于页与设置页基本相同。区别在于只有显示信息和返回按钮，供用户查看该软件的相关信息。

|  |  |
| --- | --- |
| 图 .8 记录页 | 图 2.9 设置页 |

# 概要设计和详细设计

## 系统结构及类图

系统主要由两大部分组成负责前端部分页面展示的Activity类和负责数据存储的Database类。

界面类的UML图如图 3.1所示。所有界面均继承于Android原生的Activity基类AppCompatActivity。子类通过在onCreate 生命周期中调用setContentView方法指定该界面的布局文件.xml。通过findViewById 方法绑定xml文件中设定好的控件。

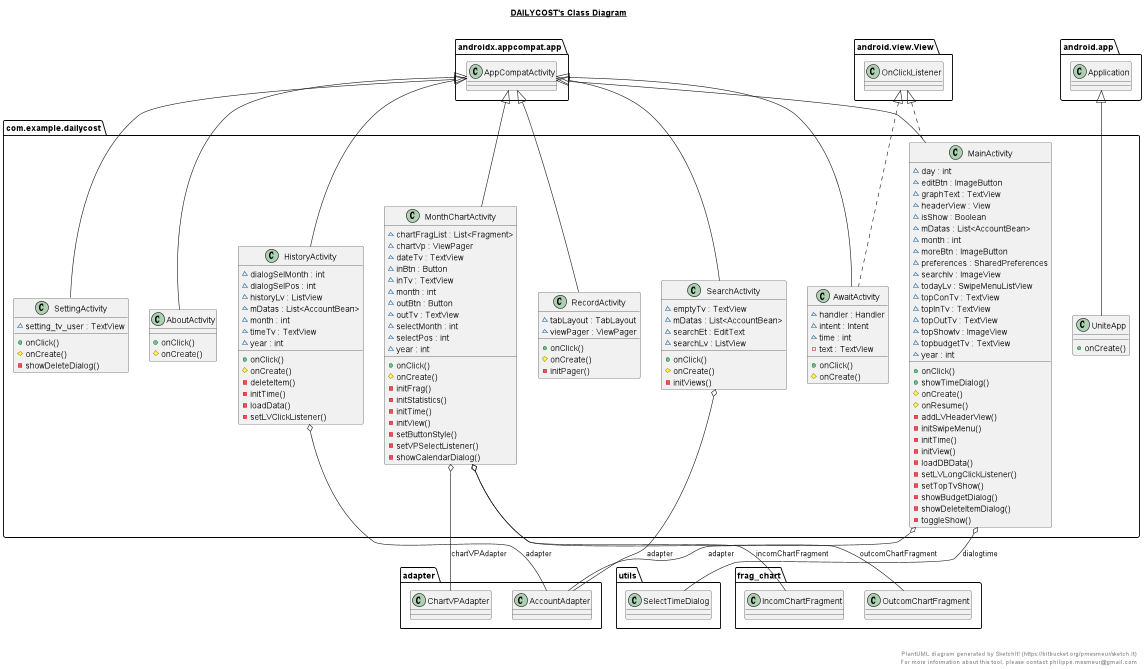


图 .1 界面类的UML图

这里以MainActivity为例，该界面主要由SwipeMenuListView 列表控件和多个按钮实现。主要的数据成员为时间（默认值为时间）和当前页面保存数据的列表mDatas。重要的方法主要是，数据载入，界面初始化，设置监听。

数据载入loadDBdata是通过DB类中静态方法getAccountListOneDayFrom-Accounttb得到当天的数据并存入mDatas中，再通知SwipeMenuListView的Adapter刷新界面。同时重写Activity类中的onResume生命周期函数，调用loadDBdata使得每次回到主页面都将重新载入数据。

界面初始化initView 主要是绑定控件和加入头控件到SwipeMenuListView中。

设置监听，主要是通过设置回调函数的方法setonclicklistener,同时将MainActivity实现onclicklistener接口，将MainActivity作为setonclicklistener的参数传给控件如添加按钮。在MainActivity. Onclicklistener 中根据被点击的控件id确定点击事件。另一种监听方法是针对SwipeMenuListView列表的动作。通过内部匿名类给SwipeMenuListView指定响应函数，这里主要实现两个方法，对于一个记录左滑显示两个按钮，分别是修改与删除。

对于其他界面，基本只实现了onClick方法和onCreate方法。其具体过程也与MainActivity类似。

Fragment类的UML图如图 3.2所示。在Android程序中一般使用ViewPager和Fragment的组合实现导航栏和页面的组合，当点击MainActivity中的添加按钮时，界面会跳转到RecordActivity中，该界面持ViewPager和Fragment 两个数据成员，ViewPager实现界面切换，Fragment实现了大部分的界面展示和数据交互功能。故在此详细介绍Fragment界面类。

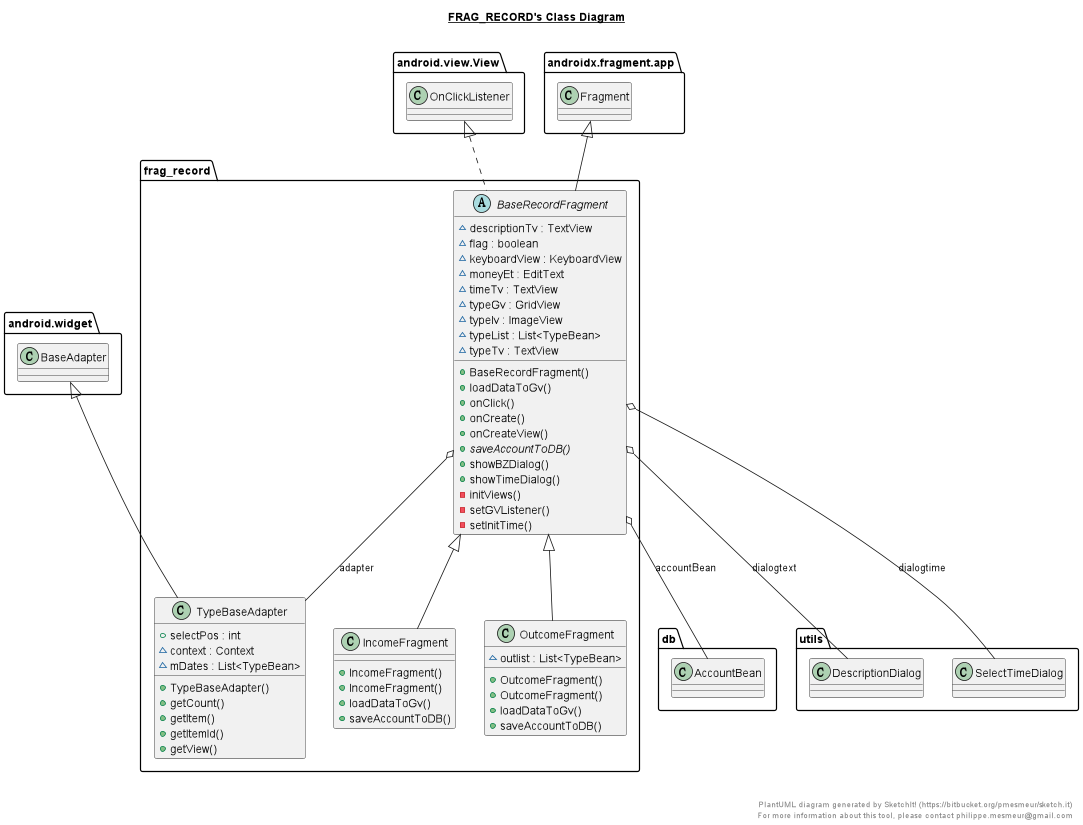


图 .2 Fragment界面类的UML图

基类BaseRecordFragment 和两个子类IncomeFragment和OutcomeFragment实现。基类实现界面展示、设置监听。子类实时根据记录属于支出或收入的类型实现从数据交互功能。

BaseRecordFragment中自定义了键盘控制用户输入主要包括金额、备注、时间。同时通过GridView展示记录的具体的支出收入类型（如交通、食物），并设置了监听，具体的类型载入GridView则交由子类完成。提交函数saveAccountToDB也交由子类完成。

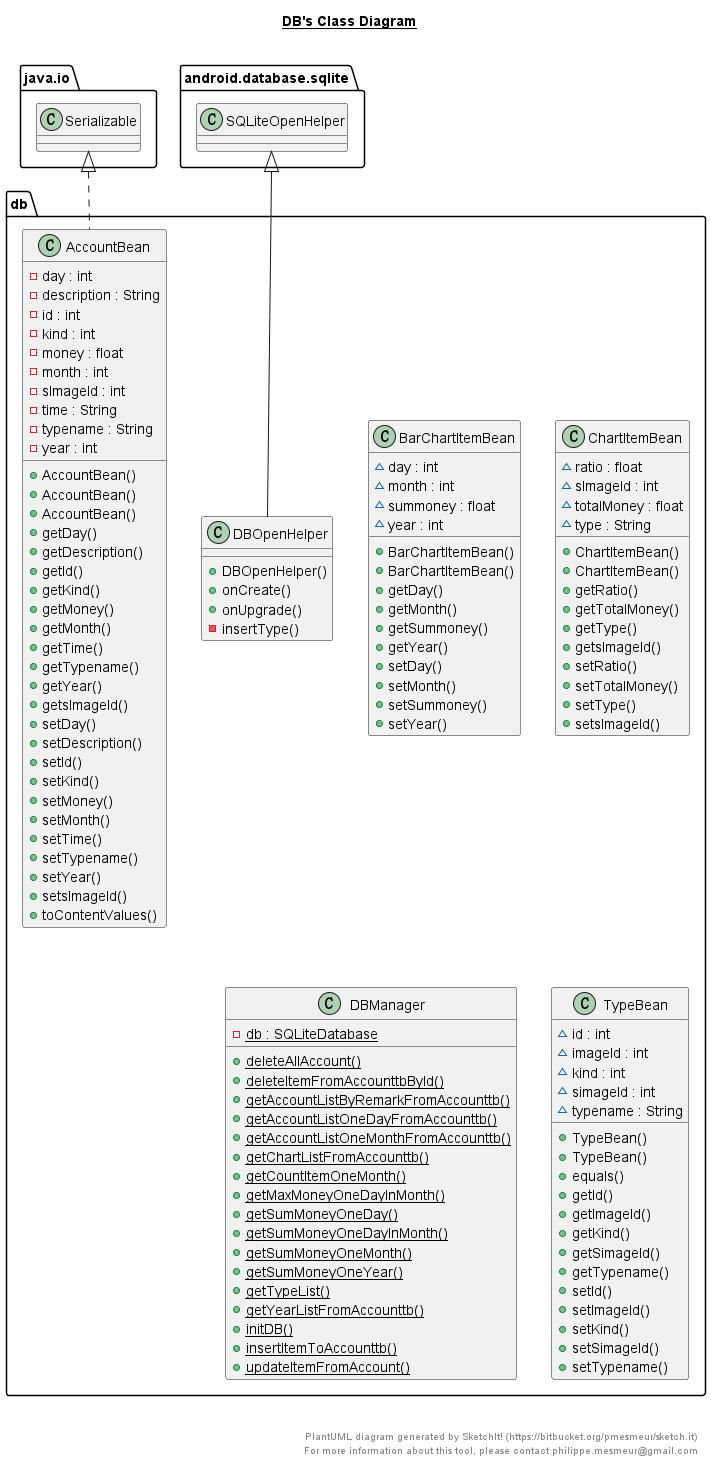


图 .3 DB类的UML图

## 关键数据结构定义

### 账本记录

AccountBean: //每个AccountBean对应数据库中的一条记录

Int id //自增主码

String typename //当前记录的类型名

int sImageId //当前记录类型对应的图片素材id

String description //当前记录的备注

float money //当前记录金额

String time //时间戳

int year,month,day //年月日，便于查询

int kind //当前记录种类(收入/支出)

public AccountBean(int id, String typename, int sImageId, String description,

float money, String time, int year, int month, int day, int kind):

包含全体数据成员的构造函数

public ContentValues toContentValues():

将对象信息打包成ContentValues用于写入到数据库(insert和update)

### 图片素材

TypeBean: //每个TypeBean对应素材库中(选中/未选中)的两个素材

public TypeBean(int id, String typename, int imageId, int simageId, int kind):

包含全体数据成员的构造函数

**3.3关键算法设计**

### DBManager接口说明

List<TypeBean> getTypeList(int kind):

获取图标资源

void insertItemToAccounttb(AccountBean bean):

插入记录

List<AccountBean> getAccountListOneDayFromAccounttb(int year,int month,int day):

查询某一天的记录集合

List<AccountBean> getAccountListOneMonthFromAccounttb(int year,int month):

查询某一月的记录集合

List<AccountBean> getAccountListByRemarkFromAccounttb(String description):

根据“’%“+description+”%‘”匹配,查询符合的记录集合

float getSumMoneyOneDay(int year,int month,int day,int kind):

统计某天收入/支出总额

float getSumMoneyOneMonth(int year,int month,int kind):

统计某月收入/支出总额

float getSumMoneyOneYear(int year,int kind):

统计某年收入/支出总额

float getMaxMoneyOneDayInMonth(int year,int month,int kind):

查询某个月中消费最高的一天的收入/支出总额

int getCountItemOneMonth(int year,int month,int kind):

查询某个月的消费指出总额

List<Integer> getYearListFromAccounttb():

查询当前有记录存在的年份集合

int deleteItemFromAccounttbById(int id):

根据id删除某条记录

void deleteAllAccount():

清空表中全体记录

List<ChartItemBean> getChartListFromAccounttb(int year,int month,int kind):

用于分类统计

List<BarChartItemBean> getSumMoneyOneDayInMonth(int year,int month,int kind):返回柱状图信息列表

void updateItemFromAccount(AccountBean bean):

根据传入参数的id,将表中原数据其他列更新

# 实现与测试

## 实现环境与代码管理

使用语言：Java

编译器：Android Studio

代码管理平台：GitHub

## 关键函数实现

getTypeList(int kind): select \* from typetb where kind = ?

nsertItemToAccounttb(AccountBean bean): db.insert("accounttb",null,values);

getAccountListOneDayFromAccounttb(int year,int month,int day):

select \* from accounttb where year=? and month=? and day=? order by id desc

getAccountListOneMonthFromAccounttb(int year,int month):

select \* from accounttb where year=? and month=? order by id desc

getAccountListByRemarkFromAccounttb(String description):

select \* from accounttb where description like '%description%'

getSumMoneyOneDay(int year,int month,int day,int kind):

select sum(money) from accounttb where year=? and month=? and day=?

getSumMoneyOneMonth(int year,int month,int kind):

select sum(money) from accounttb where year=? and month=? and kind=?

getSumMoneyOneYear(int year,int kind):

select sum(money) from accounttb where year=? and kind=?

getMaxMoneyOneDayInMonth(int year,int month,int kind):

getCountItemOneMonth(int year,int month,int kind):

select count(money) from accounttb where year=? and month=? and kind=?

getYearListFromAccounttb():select distinct(year) from accounttb order by year asc

deleteItemFromAccounttbById(int id): db.delete("accounttb", "id=?", {id })

deleteAllAccount():delete from accounttb

getChartListFromAccounttb(int year,int month,int kind):

select typename,sImageId,sum(money)as total from accounttb where year=? and month=? and kind=? group by typename order by total desc

getSumMoneyOneDayInMonth(int year,int month,int kind):

select day,sum(money) from accounttb where year=? and month=? and kind=? group by day

updateItemFromAccount(AccountBean bean):

db.update("accounttb",values,"id=?", {bean.getId()})

## 测试计划和测试用例

### 测试计划

通过对增、删、改、查各功能分别作多次数据处理来检测软件的功能完善性和用户使用体验。

### 测试用例

1. 主界面展示

主界面（如图 4.1所示）显示每月收支以及当日详细收支记录，详细收支记录显示收支种类以及添加时间。

1. 添加记录

通过主界面下方“+”进入，页面如图 4.2所示，可以选择收支种类以及添加备注（备注可用于查找记录）及其添加时间，下方数字键盘用于添加金额。

|  |  |
| --- | --- |
| Screenshot_20221128_160929_com.example.dailycost  图 .1 主界面 | 添加页面  图 4.2 添加记录界面 |

1. 搜索记录
   1. 备注查询

通过主界面右上方搜索按钮进入，其界面如图 4.3所示，可通过搜索备注关键字查询特定记录，记录显示和主界面相同，展示收支明细。

* 1. 日期查询

通过主界面中间的日期进入，其界面如图 4.4所示，可将主界面详细记录显示切换为选定日期详细条目。

|  |  |
| --- | --- |
| 搜索  图 .3 按备注查询记录 | Screenshot_20221128_161102_com.example.dailycost  图 4.4 按日期查询记录 |

1. 修改与删除

通过左划具体记录进入修改与删除选项，如图 4.5所示。选择修改，进入修改页面，其界面与图 4.2相同；选择删除，弹出提示框询问是否删除，如图 4.6所示。

|  |  |
| --- | --- |
| Screenshot_20221128_155323_com.example.dailycost  图 .5 显示修改和删除选项 | Screenshot_20221128_155332_com.example.dailycost  图 4.6 删除记录的提示框 |

1. 查询每月收支占比及趋势

通过主界面“查看图表分析”进入，其界面如图 4.7所示，展示本月总收支、收支趋势、收支占比，可通过右上角日历按钮切换查看其它月份。

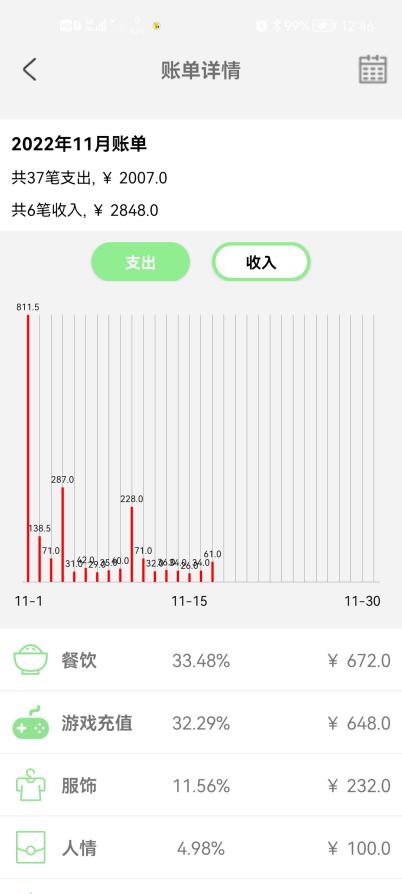


图 .7 图表分析展示

1. 查询每月账单记录

通过主界面左上角菜单进入（如图 4.8所示），选择账单记录进行查看，其界面如图 4.9所示，显示本月账单记录，可通过右上角日历选择其它月份。

|  |  |
| --- | --- |
| Screenshot_20221128_162649_com.example.dailycost  图 .8 菜单界面 | Screenshot_20221128_162715_com.example.dailycost  图 4.9 账单记录 |

## 结果分析

上述测试结果表明该产品已实现记账软件的所有基本功能，界面交互和数据处理方面符合预期，在具体实现与构思的结合后基本实现前期对产品的构思，且测试过程中未发生软件异常退出等重大程序漏洞。

# 总结

## 用户反馈

软件界面美观简洁。作为一款记账软件，操作简单，能很好的展示每天的收支情况，其图表分析能很好的展示收支变化和占比情况，有助于规划消费习惯。

## 全文总结

经过本次软件工程实践课程，在项目管理、工程实践、版本管理、技术框架几个方面有比较深刻的体会，下面谈谈我们在实践中遇到的一些问题。

从技术框架来说，本次我们小组选择Android平台进行开发，首先是考虑到了安卓手机在国内市场的普及率，更多的用户能为我们带来更广的市场。但是因为先前没有使用Android框架开发的经验，所以在项目初期和整合阶段遇到过一些问题，如前端交互部分需要花费大量时间阅读文档，sdk版本不能统一等。

从版本管理而言，起初我们小组是采用Github共享文档、项目文件等。但到了实践中期，遇到突发的bug等情况，为了尽快解决问题，有时直接将文件打包通过U盘、qq等方式直接传输给对应负责的同学，最后再一台主机上手动整合，由于连带gradle项目的配置文件，也造成了上文提到的sdk版本问题。

工程实践角度，由于项目的规模相较于以往课程的控制台程序规模大得多，这就需要我们在实现时采用一定的设计模式，保证程序可运行的前提下，使得代码尽量逻辑条理清晰、可读性强，也便于后续的扩展和维护。我们也梳理了项目的结构形成文档，便于开发者理解不同模块之间的调用关系。

项目管理层面而言，由于是小组合作完成一次项目，协商过程中多少会产生一些分歧，如立项早期的头脑风暴环节。但是经过仔细讨论、列出清晰的优缺点，小组成员很快达成一致，每位成员都能很好的完成自己分内的任务，主动形成文档助于对接的组员理解，团队氛围十分融洽。值得提高的是对于项目实现时间安排的规划，因为是第一次参与这样的开发流程，再加之其他课程中期任务的影响，好在运动会的延后为我们提供充足时间最后完善我们的作品。

最后感谢小组的每位成员为本次项目做出的贡献，也谢谢范老师验收时对我们项目提出的宝贵意见，这次实践收获良多。

# 体会

以下为各个组员在本次课题设计中的心得体会。

## 刘玺语

## 吕嘉乐

本次软件工程我主要有两点体会。

首先是关于项目结构设计的问题，由于是第一次接触安卓app的开发，在前后端分离上完成的不是很理想。往往一个activity中既承担了UI展示的功能又要完成数据处理，很多的功能都直接交由activity实现，项目的耦合度很高，在debug和添加新功能时比较困难，在共同开发的过程中理解代码逻辑也比较困难。后来有了解到现在实际的安卓开发就会采用mvp或mvvm架构，比较好的解决了这个问题。

其次是共同开发中，小组成员之间的交流共同中，一般而言每个人负责自己部分的代码逻辑，提供接口让其他人访问，这种模式在遇到bug时，就比较麻烦。由于不清楚对方的代码实现，只能排除自己代码中的问题。问题定位起来比较麻烦，修改倒是比较简单。在本次项目中，我和我的小组成员都学习了很多，之前写代码都是一个人写，首次共同开发遇到了很多问题，大部分也在我们的共同努力下得到了解决，我们也收获良多。

## 毛俊棋

本次课设中，我主要负责课题初期的方案设计和后期的文档总结，负责的方面更侧重项目的准备和整理。

在此期间，我遇到了许多曾经自己完成其他课题时不曾遇到的问题。如何统合组员间的意见，同时让项目不过于天马行空以至于难以实现，贯穿课题的整个设计阶段，这让我充分了解到小组的协作工作是多么重要。其次在作原型设计时由于对软件使用的熟练度不高，即便组员提出了一些看似容易实现的需求，但在设计过程中总让人不尽人意，最后也耗费不少时间在交流设计的过程中，不过因此也最终得到一份差强人意的设计方案。最后则是在本次文档的撰写过程中，由于软件实现过程中并没有跟进文档的内容，导致最后在软件实现的章节对具体的函数或数据结构都要重新阅读项目代码并整理，在这期间浪费了不必要的时间。

## 张恒铭

在本次软件工程课设中，我们小组通过分析多款记账软件，设计并实现了属于我们自己的记账软件——想记就记。我在小组分工中担任部分的页面实现以及对软件的功能跟踪测试。

通过这近两个月的小组合作下，我对于团队合作开发有了更深刻的认识。在过去的单人开发中，进度是不会冲突，但在多人开发中，因为每一个人的开发进度不同，所以沟通汇总每个人的进度就格外重要。这次我们的开发进度本来应该通过线上同步，但因为过去独立开发的习惯选择每次都是组员中各自发送已完成的部分，造成了组员间沟通的不及时。这次小组合作后，相信以后能更好地进行团队开发任务。同时，我作为测试人员，测试软件的实际功能实现，因此我需要想出可能存在问题，这也帮助我对测试用例的设计有了更多体会。

总而言之，通过这次小组合作后，我学到了很多有助于团队合作的习惯。