**总目录**

1. **附录A 软件需求分析报告文档**
2. **附录B 软件概要设计报告文档**
3. **附录C 软件详细设计报告文档**
4. **附录D 测试用例报告文档**

附录A 软件需求分析报告文档

1.引言………………………………………………………………………………………………2

[1.1 编写目的 2](#_Toc127799063)

[1.2 项目风险 2](#_Toc127799064)

[1.3 文档约定 2](#_Toc127799065)

[1.4 预期读者和阅读建议 2](#_Toc127799066)

[1.5 产品范围 2](#_Toc127799067)

[1.6 参考文献 3](#_Toc127799068)

[2. 综合描述 3](#_Toc127799069)

[2.1 产品的功能 3](#_Toc127799070)

[2.2 用户类和特性 3](#_Toc127799072)

[2.3 运行环境 4](#_Toc127799073)

[3. 外部接口需求 4](#_Toc127799076)

[3.1 用户界面 4](#_Toc127799077)

[3.2 硬件接口 16](#_Toc127799078)

[3.3 软件接口 17](#_Toc127799079)

[4系统用例设计 16](#_系统用例设计)

[5. 系统功能需求 19](#_Toc127799081)

[5.1 用例描述 1](#_Toc127799082)9

[5.2 用例业务流程描述 2](#_Toc127799083)1

[6. 可拓展功能 2](#_Toc127799085)2

# 引言

此需求分析报告主要介绍产品——成电微记的基本用户界面和功能等基本信息，帮助用户建立起对产品的初步、大致的认知。在阅读时最好结合APP的操作一起阅读。在阅读本文档之后，用户应该可以基本了解产品的使用。

## 编写目的

此报告目的在于介绍成电微记APP的基本情况，包括不限于功能介绍、基本系统架构等。

## 项目风险

* 任务提出者：需要在本阶段向开发者明确说明软件需求，并与开发者讨论需求的可行性、开发用户界面模型明确业务需求；
* 软件开发者：需在本阶段理解任务提出者的需求；
* 产品使用者：在本阶段向项目组描述产品需求。

## 文档约定

* 正文风格：宋体、五号
* 一级标题：宋体、四号
* 二级标题：宋体、小四

## 预期读者和阅读建议

* 用户：用户优先阅读原型图的解释，熟悉软件界面。其次阅读用例以及系统功能需求，了解软件的基本功能及使用；
* 开发人员：优先阅读原型图解释以及用例描述，明确软件的功能需求，并参考接口需求以及参考文献了解软件的开发技术；
* 测试人员：阅读产品的用例描述和功能需求描述，为后续测试活动做铺垫；
* 文档编写入员：仔细阅读，不断补充完善产品需求报告。

## 产品范围

本产品致力于为青少年尤其是学生提供多种生活服务。产品所具有的日记、记账、待办功能可以为用户的生活带来极大便利。通过日记功能，用户可以记录自己日常的所思所感，通过记账功能可以对自己的日常支出了然于心，提高自己的理财和规划能力，通过待办功能可以条理地安排日常的计划，避免任务拖沓，提高工作和学习效率。

## 参考文献

* + [情境感知服务](https://developer.huawei.com/consumer/cn/hms/huawei-awarenesskit)
  + [机器学习服务](https://developer.huawei.com/consumer/cn/hms/huawei-mlkit)
  + [图像服务](https://developer.huawei.com/consumer/cn/hms/huawei-imagekit/)
  + [音频服务](https://developer.huawei.com/consumer/cn/hms/huawei-audiokit/)

# 综合描述

本产品是普通生活类app，运行于华为手机可正常使用全部功能，使用华为HMS为用户提供生活便利：如写日记、利用TODO模块计划日常任务以及使用记账功能规划日常开销。

## 产品的功能

本产品主要由三大功能块构成：日记、TODO和账单。

日记模块为用户提供发表日记的功能，用户可以上传多种类型的日记。

TODO模块为用户提供规划事务的功能，用户可以使用该模块可以为自己的工作学习做规划。

账单模块为用户提供记账功能，用户可以统计自己的收入和支出，依次为依据规划自己的日常开销。

每个模块的功能组成如下：

图2-1 产品功能表

|  |  |  |
| --- | --- | --- |
| 日记 | TODO | 账单 |
| 上传文字 | 浏览待办事务 | 浏览已有的收入支出 |
| 上传音频 | 添加待办事务 | 添加收入 |
| 上传图片、视频 | 编辑已有待办事务 | 添加支出 |
|  | 查看待办事务的数据分析 | 计划预算 |

## 用户类和特性

目标用户①：第一类用户心思细腻，喜欢记录分享自己的日常生活，使用成电微记的日记功能可以方便记录日常生活的点滴。

目标用户②：第二类用户有较强理财观念，希望规划日常生活的开支，使用成电微记的账单功能可以满足需求，并且可以设置预算来避免花销过多。

目标用户③：第三类用户有较强自律性和计划性，使用成电微记的待办功能可以合理安排自己的日常计划，使得学习和工作都井井有条。

## 运行环境

鉴于成电微记使用了HMS服务，因此成电微记必须运行于华为手机才可正常使用全部内容，否则某些功能无法正常使用。

# 外部接口需求

成电微记使用了多种HMS服务：

* + [情境感知服务](https://developer.huawei.com/consumer/cn/hms/huawei-awarenesskit)：成电微记中使用该服务主要用于获取天气情况。
  + [机器学习服务](https://developer.huawei.com/consumer/cn/hms/huawei-mlkit)：成电微记中使用该服务主要用于实时进行语音识别。
  + [图像服务](https://developer.huawei.com/consumer/cn/hms/huawei-imagekit/)：成电微记中使用该服务主要用于图片的相关处理。
  + [音频服务](https://developer.huawei.com/consumer/cn/hms/huawei-audiokit/)：成电微记中使用该服务主要用于音频的处理。

## 用户界面

此节描述需求分析阶段的原型图，对各个界面进行阐述，介绍各个功能以及按钮作用，与实际开发成品可能略有出入。

### **3.1.1日记首页**



图3-1日记首页原型图

* 日期及天气
  + 本部分使用华为HMS api 实现，获取当前天气及星期信息
* 日记展示
  + 日记抬头
    - 头像（圆圈），日期，星期，天气，地点
  + 日记内容
    - 支持文字，音频，视频，图片
  + 日记删除
    - 点击右侧更多按钮，弹出删除按钮，实现日记删除。
* 日记上传
  + 见“日记上传”

### **3.1.2日记上传**



图3-2 日记上传原型图

* 入口：日记首页-上传日记
* 文字上传框
  + 单击编辑输入
* 照片及视频上传
  + 可实现选择手机文件上传，或者拍照，录像
  + 单击叉号删除照片或视频
* 音频上传
  + 可实现选择手机文件上传，或者录音
  + 单击叉号删除音频
* 发表
  + 点击右上角发表
* 取消
  + 单击取消，放弃发表流程，退回首页

### **3.1.3相册**

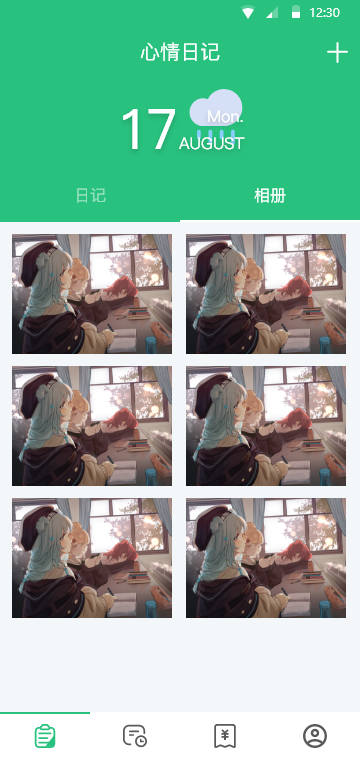


图3-3 相册原型图

* 入口：相册
* 在本部分实现查看所有发表在日记的照片

### **3.1.4TODO首页**



图3-4 TODO首页原型图

* 入口：待办
* 显示所有待办事项
  + 未完成待办事项在前，按照添加待办事项的时间顺序排列
  + 每一个待办事项完成后排在最后
* 完成待办事项
  + 点击“完成”按钮，完成待办事项
* 编辑待办事项
  + 点击待办事件，即可编辑待办事件
* 添加待办事件
  + 点击 + ，添加待办事件
* 查看数据分析
  + 点击右上角饼图，即可查看待办事项完成情况的数据分析
* 待办通知
  + 当待办的开始时间到达时，进行系统通知：”待办：xxxxx，可以开始啦“

### **3.1.5添加待办**



图3-5 添加待办原型图

* 入口：待办首页-添加按钮
* 输入待办名称。名称限制字数
* 选择开始时间。
  + 精确到分钟
* 关闭添加待办
  + 可直接点击 x ，关闭添加待办框
  + 点击“取消”，关闭待办框
  + 关闭后不保存该待办
* 添加待办。
  + 待办名称和待办事件均选择完成以后，点击“确认”，完成待办事项添加。

### **3.1.6编辑待办事项**



图3-6 编辑待办原型图

* 入口：待办首页—具体待办事项
* 编辑待办事项
  + 在待办首页点击待办框以后，显示该页面
  + 显示待办名称，开始时间
  + 可更改待办名称，开始时间
* 确认修改
  + 编辑完成以后，点击“确认”，保存本次修改
* 取消修改
  + 点击关闭按钮，不保存本次修改
* 删除
  + 点击“删除”，删除该待办

### **3.1.7完成度分析**

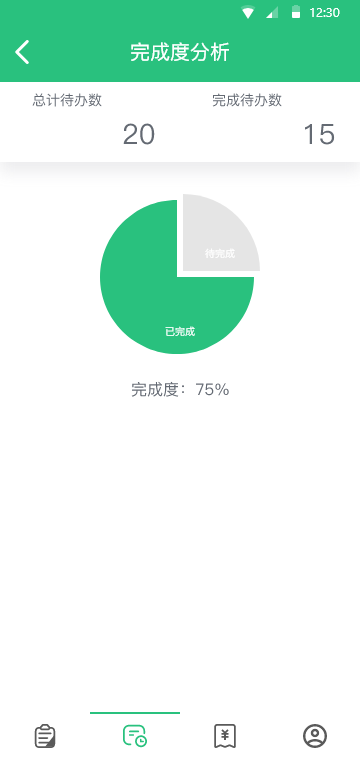


图3-7 完成度分析原型图

* 入口：待办首页-完成度分析按钮
* 显示总待办数，完成待办数
  + 总待办数为已添加的所有待办
  + 已删除待办不包含其中
* 显示完成度统计图
* 显示完成率

#### 3.1.8账单首页



图3-8 账单首页原型图

* 入口：账单
* 显示本月支出，本月收入
  + 本月支出，已添加本月的所有支出之和
  + 本月收入，已添加本月的所有收入之和
* 以每一天为分隔，显示记账清单
  + 每条记账包括记账名称，备注，金额
  + 支出金额显示负号，收入金额显示正号
* 添加记账
  + 点击右下角 + ，添加记账

#### 3.1.9添加

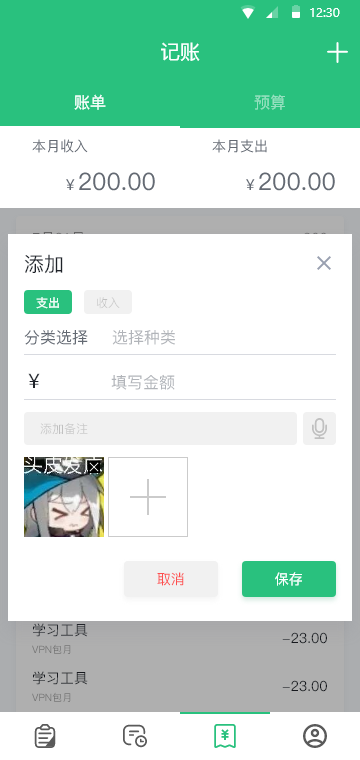


图3-9 账单添加原型图

* 入口：账单首页-添加
* 选择“支出“，添加支出
* 输入支出金额
* 选择支出分类
  + 饮食：食堂三餐，聚餐，水果零食，外卖，外出吃饭
  + 衣服饰品：衣服裤子，化妆饰品，鞋帽包包
  + 通讯交通：话费，网费，邮费，公共交通，打车租车，自行车
  + 学习进修：书籍软件，学杂费，文印费，学习工具
  + 休闲娱乐：电影，旅游，健身，其他
  + 其他杂项
* 添加备注（限制备注字数）
  + 文字添加
  + 语音添加
* 添加图片（限制图片数量最多三张）

### **3.1.10预算**



图3-10 预算首页原型图



图3-11 编辑预算原型图

* 入口：账单首页-预算
* 点击上方切换到”预算界面“
* 显示本月预算总支出
  + 总支出为所有预算支出之和
* 显示各类预算金额
  + 预算分类名称
  + 预算金额
  + 余额。余额为预算金额 - 该分类已支出金额
  + 显示所有分类。
    - 饮食
    - 衣服饰品
    - 通讯交通
    - 学习进修
    - 休闲娱乐
    - 其他杂项
* 编辑预算：
  + 点击某条预算后，即可对该分类的支出分类进行编辑
  + 未编辑支出金额的分类，显示0.00
* 超值提醒
  + 当某一类预算第一次超支时，进行系统通知”xx类预算已超支“。该分类后续超支不再提醒
  + 总预算超支时，进行系统通知：”本月总预算已超支“。

### 3.1.11我的



图3-12 “我的”原型图

* 入口：我的
* 更改头像
  + 流程及使用能力同“日记-上传照片”

## 硬件接口

以下为成电微记中使用到的硬件相关服务：

* + - * 调用摄像头
      * 读取sd卡
      * 话筒录音
      * 读取手机状态
      * 网络访问

## 软件接口

* 数据库：Room（2.3.0）

# 系统用例设计

这些用例是系统功能的体现，在系统功能需求描述中将详细的描述这些用例。

* 1. **日记**

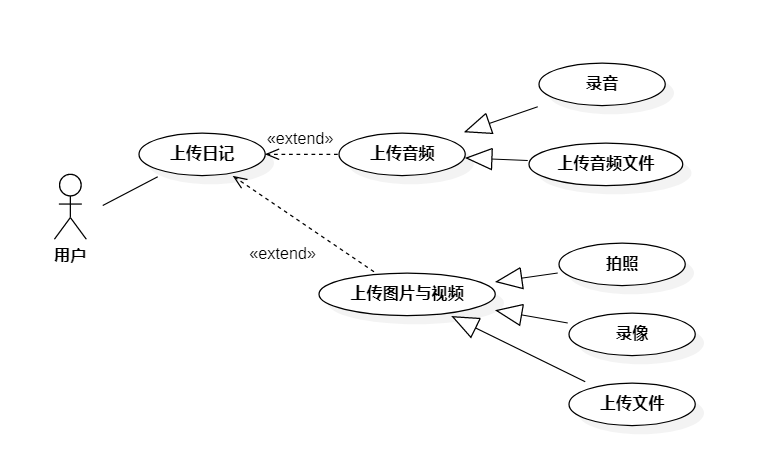


图4-1 日记模块用例图

用户使用日记功能，可以浏览发布的日记、发布新的日记。发布的新日记可以为多种格式：文字、音频、图像和视频。在发布媒体文件时可以选择本地媒体或即时录入。

1. **TODO**

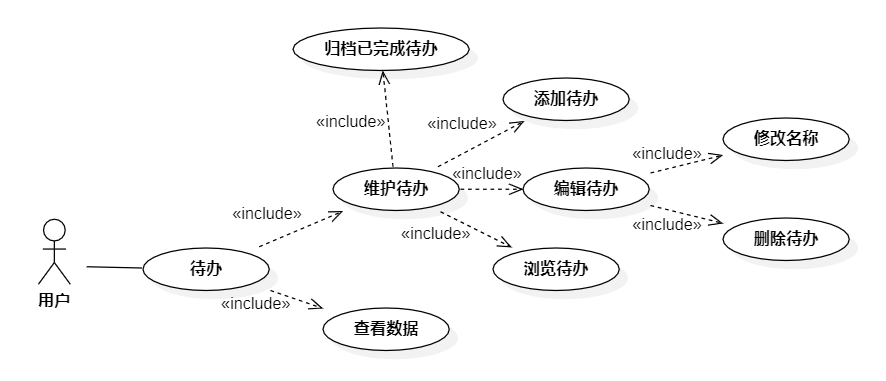


图4-2 TODO模块用例图

用户使用待办功能可以浏览已发布待办事项、编辑已有待办事项、添加新的待办事项、对完成待办进行归档操作、查看数据分析等操作。

1. **记账**

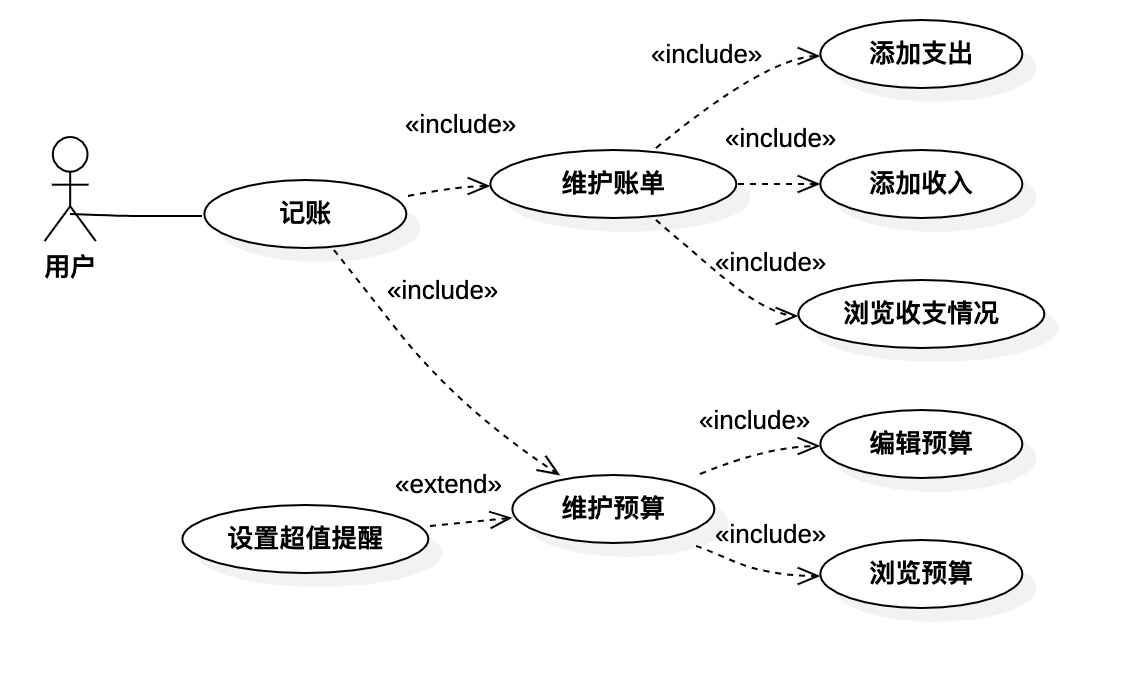


图4-3 记账模块用例图

用户在记账模块可以查看实时的收入与支出，并对收入和支出进行添加。另外用户也可以使用预算功能对接下来的生活支出进行预算规划，在支出超出计划预算时系统将进行超值提醒。

1. **我的**

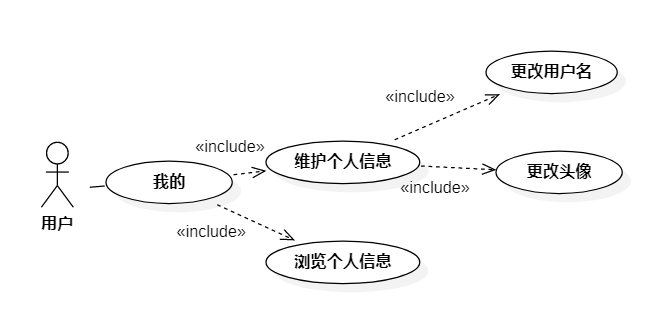


图4-4 我的模块用例图

在“我的”界面，用户可以浏览用户信息以及更改头像。

**4.5系统**

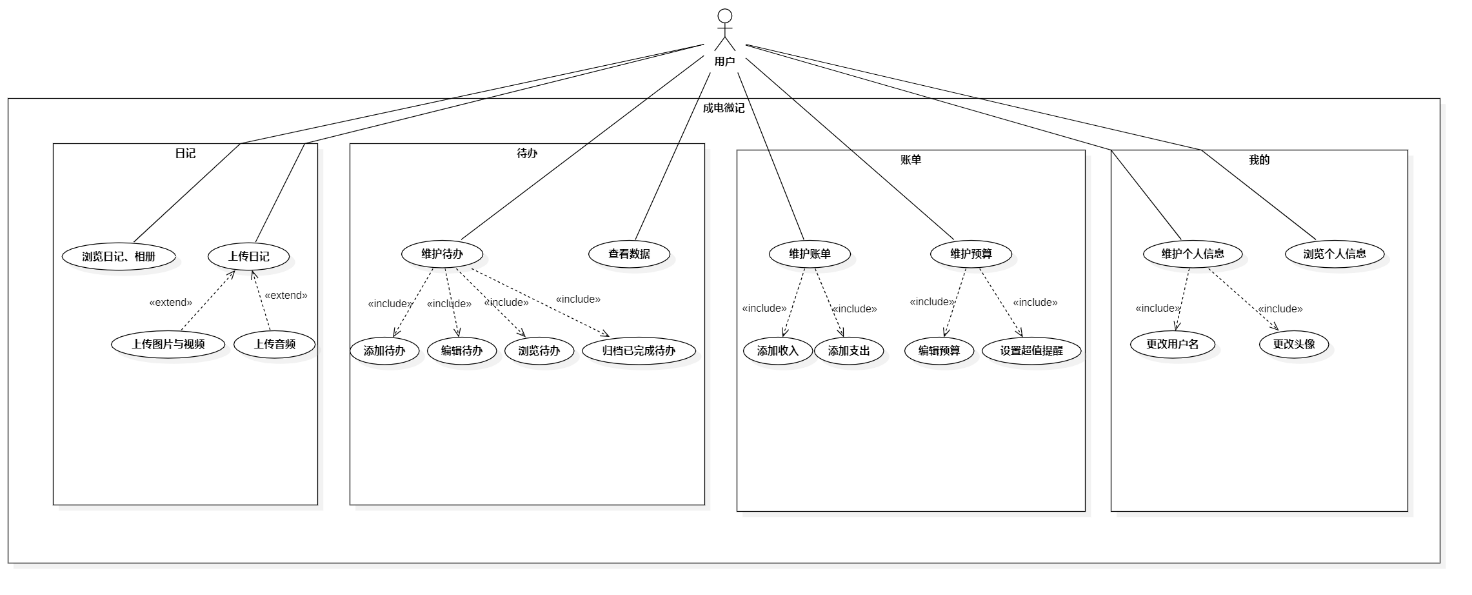


图4-5 系统用例图

# 系统功能需求

## 用例描述

此节对各个子系统用例进行详细的用例描述。

表5-1日记用例描述表

|  |  |
| --- | --- |
| 用例名称 | 日记 |
| 简要说明 | 用户在日记模块可以上传日记，同时也可以上传媒体文件，包括图片、视频和音频，上传的日记可以在“日记”一栏进行浏览，在“相册”一栏可以浏览上传过的图片及视频。 |
| 前置条件 | 已获取读取本地文件和调用录音和录像设备的权限。 |
| 基本事件流 | 1.点击加号按钮出现写日记界面  2.在输入栏中输入自己想写的文字  3.点击下方的添加媒体文件按钮  4.选择要上传的媒体文件类型  5.选择要上传的内容，进行确认  6.点击右上方按钮进行日记发布 |
| 其他事件流 | 1.点击已发布日记进入详情页  2.点击右上方按钮删除该日记 |
| 后置条件 | 日记一栏显示已发布的日记，相册显示已发布的图片和视频 |

表5-2 记账用例描述表

|  |  |
| --- | --- |
| 用例名称 | 记账 |
| 简要说明 | 用户在记账模块可以记录自己的日常收入和支出，达到计划开销的目的，也可使用预算功能对未来支出进行规划。 |
| 基本事件流 | 1.在记账界面点击右上方按钮进入添加界面  2.选择添加类型（支出or收入）  3.选择分类  4.填写金额  5.添加备注（可选），可以手动输入也可以实时语音识别  6.点击保存发布信息 |
| 其他事件流1 | 1.进入预算界面  2.选择预算分类  3.找到希望计划的预算分类  4.在开支预算一栏直接修改预算金额 |
| 其他事件流2 | 1.进入记账界面  2.点击已发布收入和支出信息  3.根据个人需要进行修改  4.点击保存即可修改完成 |
| 后置条件 | 记账一栏显示已发布的收入和支出，并显示当月支出和收入，预算界面也显示各个分类的预算以及当月的总预算。 |

表5-3 我的用例描述表

|  |  |
| --- | --- |
| 用例名称 | 我的 |
| 简要说明 | 用户在“我的”模块可以修改自己的昵称以及头像 |
| 基本事件流 | 1.进入“我的”界面  2.点击“修改头像”  3.选择心仪的图片即可修改成果 |
| 其他事件流1 | 1.进入“我的”界面  2.点击个人信息中的用户名  3.进行用户名的修改 |
| 后置条件 | 显示修改后的用户名和头像 |

表5-4 待办用例描述表

|  |  |
| --- | --- |
| 用例名称 | 待办 |
| 简要说明 | 用户在待办模块可以对自己接下来的任务进行合理规划，并且可以查看过往的计划完成情况 |
| 基本事件流 | 1.进入待办界面  2.点击上方按钮进行任务添加  3.输入任务名称及时间  4.点击确定完成待办的添加 |
| 其他事件流1 | 1.进入待办界面  2.点击要修改的待办任务  3.根据需要修改名称或时间  4.点击确定进行保存 |
| 其他事件流2 | 1.进入待办界面  2.点击数据分析按钮  3. 浏览待办完成情况 |
| 后置条件 | 显示已发布的待办，并且数据分析页面统计完成情况 |

## 用例业务流程描述

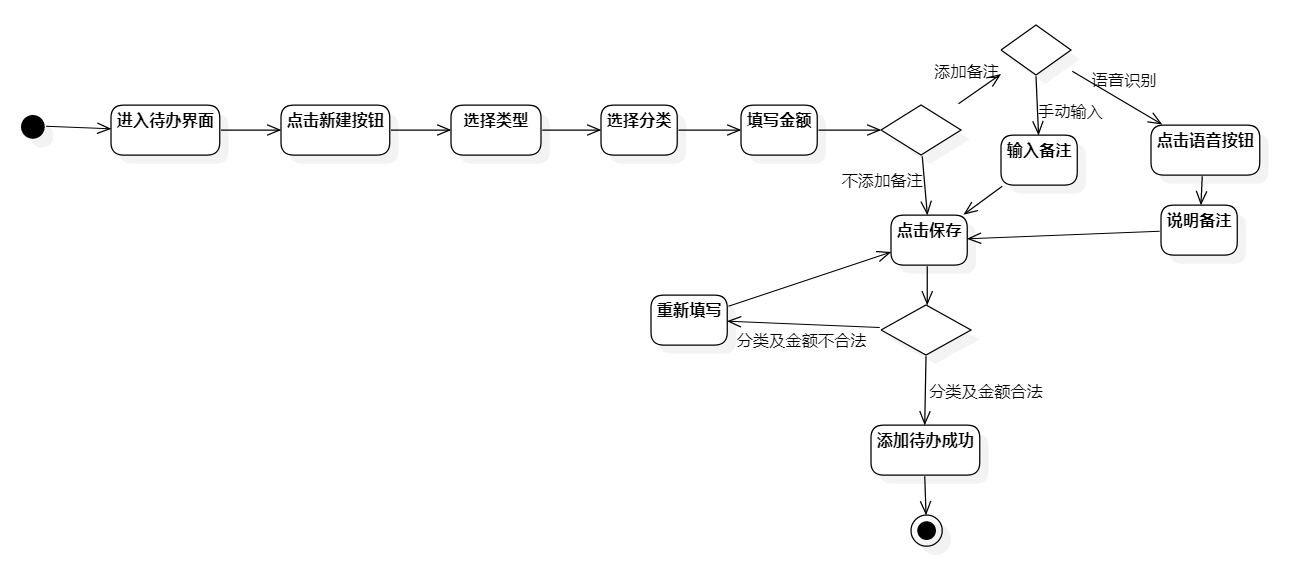
以“添加新的支出”该活动为例，活动图如下：

图5-2 “添加支出”活动图

# 可拓展功能

1.考虑日记的私密性，可以对日记内容进行加密处理。

2.日记模块添加草稿箱功能，用于保存尚未完成的日记。

# 附录B 软件概要设计报告文档

1.引言……………………………………………………………………………………………22

[1.1 预期读者和阅读建议 23](#_Toc127799099)

[2. 设计概述 23](#_Toc127799101)

[2.1 设计原则和设计要求 23](#_Toc127799103)

[3. 系统逻辑设计 24](#_Toc127799104)

[3.1 系统组织设计 24](#_Toc127799105)

[3.2 系统结构设计 24](#_Toc127799106)

[3.3 系统类设计 2](#_Toc127799109)5

[3.4 关键处理过程 2](#_Toc127799109)8

[4. 系统出错处理设计 28](#_Toc127799113)

[4.1 系统出错处理表 28](#_Toc127799114)

[4.2 维护处理过程表 28](#_Toc127799115)

[5. 数据库设计 29](#_Toc127799119)

# 引言

本概要设计报告将对软件的设计进行概括描述，将介绍产品的系统组成、内部类设计、出错处理等方面，使用户对软件的基本设计有大致了解。

## 预期读者和阅读建议

* 开发人员；阅读产品的系统设计，了解产品的架构。
* 测试人员；根据软件的设计介绍思考测试用例的设计。
* 文档编写人员；根据实际软件不断调整报告的编写。

# 设计概述

本节描述现有开发条件和需要实现的目标，说明进行概要设计时应该遵循的设计原则和必须采用的设计方法。

## 设计原则和设计要求

描述对本软件系统进行概要设计的原则，考虑以下几方面的内容：

* 命名规则；小驼峰命名法
* 模块独立性原则：根据功能划分为相对独立的多个组件。
* 开闭原则：应该对外延具有开放性，对修改具有封闭性。

# 系统逻辑设计

## 系统组织设计

系统组织设计通过系统组织表描述本系统由哪些子系统(模块)组成，这些子系统与业务职能之间的关系，如下：

表3-1 系统组织设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号 | 英文名称 | 中文名称 | 业务职能 |
| 1 | JournalFragment | 日记 | 负责日记的编写功能 |
| 2 | TodoFragment | 待办 | 负责待办模块的功能 |
| 3 | AccountFragment | 账单 | 负责账单相关的功能 |
| 4 | InfoFragment | 个人信息 | 负责个人信息相关的功能 |

## 系统结构设计

本节将对系统特性作较为详细的描述

系统用例模块映射表说明每个子模块实现那些用例或者用例的一部分。

表3-2-1 日记模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号：1  子系统英文名称：JournalFragment  子系统中文名称：日记 | | | |
| 用例编号 | 系统用例  英文名称 | 系统用例  中文名称 | 操作功能 |
| 1 | JournalEditActivity | 写日记 | 用户进入编写日记界面写日记并发布 |
| 2 | JourFragment | 浏览日记 | 浏览已发布的日记 |
| 3 | JournalItemDetailActivity | 日记详情 | 具体查看某个已发布的日记详情 |
| 4 | AlbumFragment | 相册 | 查看已分类的各个相册 |
| 5 | AlbumPictureFragment | 相册详情 | 点击查看具体某个分类的相册内容 |

表3-2-2 待办模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号：2  子系统英文名称：TodoFragment  子系统中文名称：待办 | | | |
| 用例编号 | 系统用例  英文名称 | 系统用例  中文名称 | 操作功能 |
| 1 | TodoFragment | 查看待办 | 显示待办情况 |
| 2 | AddDialogFragment | 添加待办 | 添加新的待办任务 |
| 3 | UpdateDialogFragment | 编辑待办 | 对已有待办进行各个编辑操作 |

表3-2-3 账单模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号：3  子系统英文名称：AccountFragment  子系统中文名称：账单 | | | |
| 用例编号 | 系统用例  英文名称 | 系统用例  中文名称 | 操作功能 |
| 1 | AccoFragment | 浏览账单 | 显示各个分类以及本月的收入和支出情况 |
| 2 | AccoAddFragment | 编辑账单 | 关于账单的操作功能：添加账单、修改账单、删除账单 |
| 3 | BudgetFragment | 预算 | 显示各项分类的预算等基本功能 |

表3-2-4 我的模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号：4  子系统英文名称：InfoFragment  子系统中文名称：个人信息 | | | |
| 用例编号 | 系统用例  英文名称 | 系统用例  中文名称 | 操作功能 |
| 1 | InfoFragment | 查看编辑个人信息 | 显示个人信息，并可以对用户名及头像进行更改 |

## 设计类结构

### **3.3.1基本类设计**

下图为成电微记基本架构设计示意图，意在描述各个功能模块的逻辑组成结构。下图主要显示各个功能相关的Fragment（Activity）的跳转、组成关系以及基本的与数据增删查改有关类的设计。

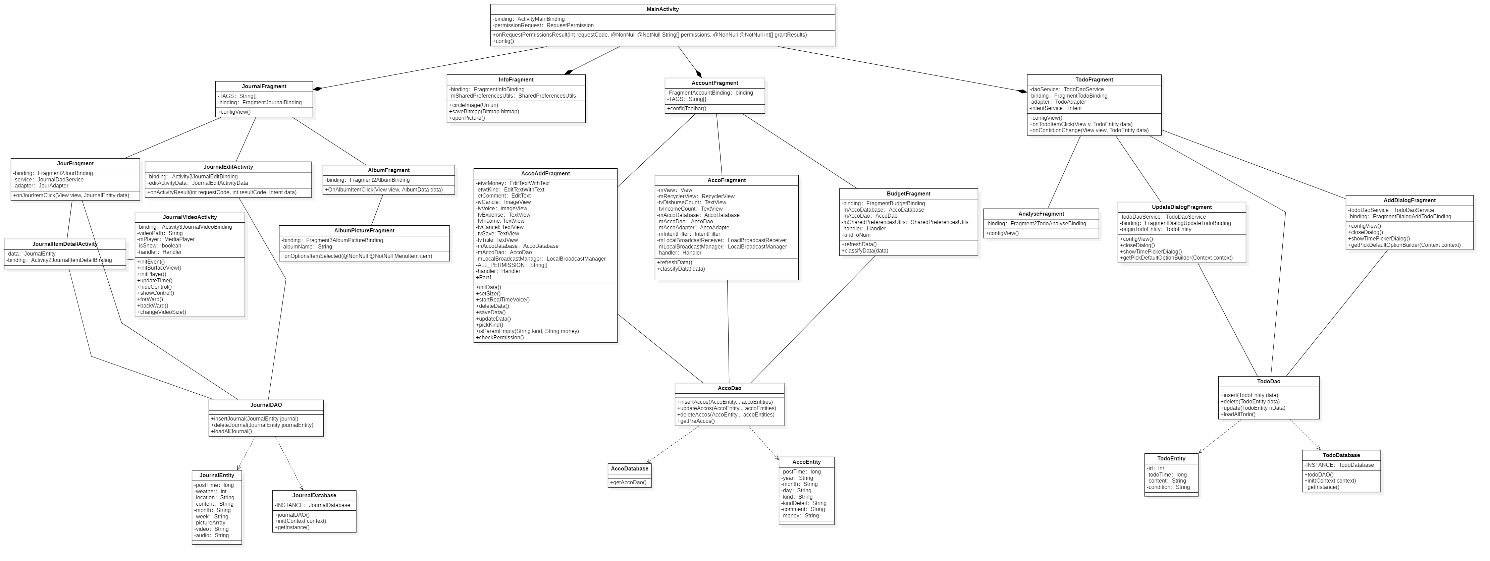
采用MVC三层架构,下图中,从上到下依次为:所有activity以及fragment都是可与用户产生交互的边界类; JournalDAO、AccoDao和TodoDao为实现数据存储的控制类;entity以及database为存储实体数据的实体类。 

图3-3-1 基本类设计图

### 3.3.2 HMS相关类

下图为设计HMS服务的相关类结构示意图

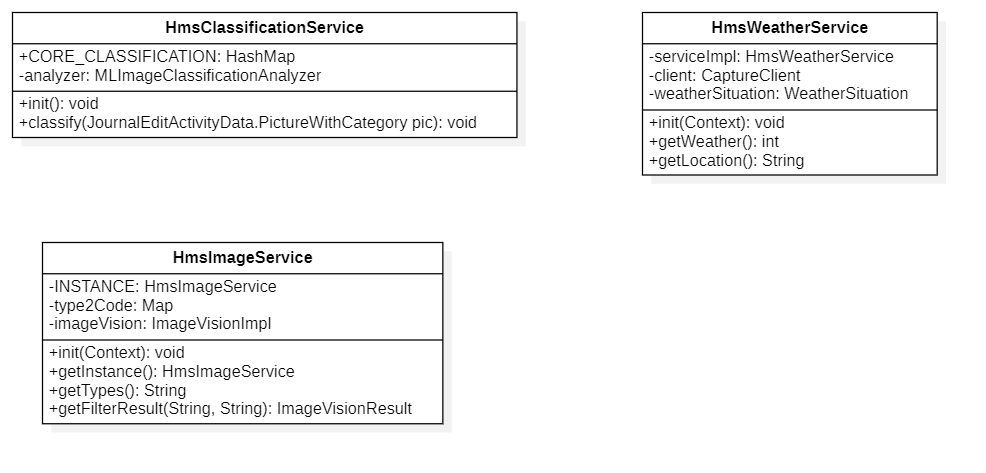


图3-3-2 HMS服务相关类图

### 3.3.3 数据库相关类

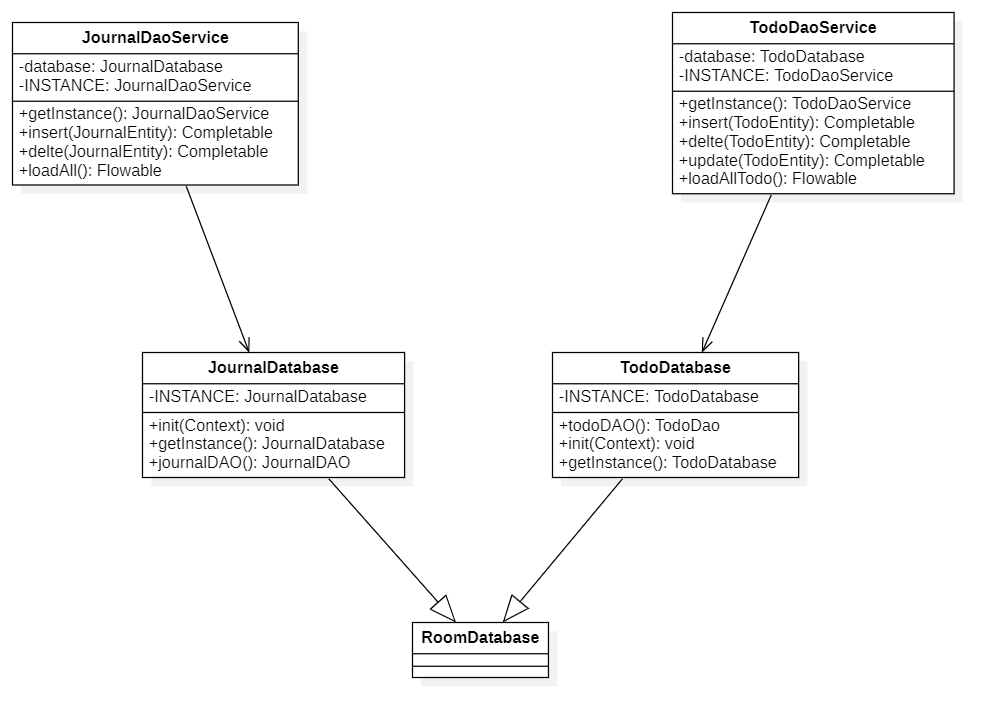


图3-3-3 数据库相关类图

## 关键处理过程或者算法的设计

以“用户添加新的支出”为例，顺序图示例如下：

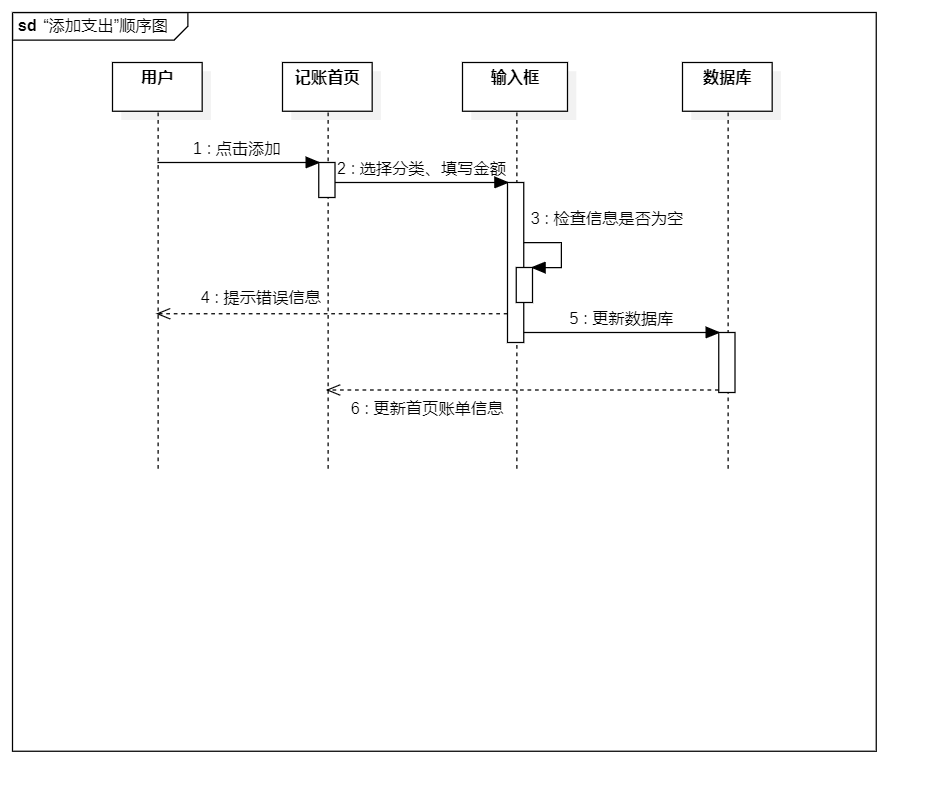


图3-4 添加支出顺序图

# 系统出错处理设计

## 系统出错处理表

本表给出有关出错处理的产生原因、提示信息、以及建议处理方法。

表4-1 系统出错处理表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号：3  子系统英文名称：AccountFragment  子系统中文名称：账单 | | | |
| 错误编号 | 错误名称 | 错误原因 | 处理方式 |
| 1 | 输入不合法 | 用户未选择分类或未填写金额 | 向用户显示错误信息，提示正确输入。 |

## 维护处理过程表

系统出错时，将调用维护处理过程对错误进行处理，有关维护处理过程的各项内容由维护处理过程表进行描述。

表4-2 维护处理过程表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 子系统编号：3  子系统英文名称：AccountFragment  子系统中文名称：账单 | | | | | | |
| 错误编号 | 处理过程 | 处理过程 | 处理功能 | 入口参数 | 出口参数 |
| 英文名称 | 中文名称 |
| 1 | isParameterEmpty | 参数是否为空 | 检查“种类”和“金额”是否为空 | String kind；String money | 布尔值 |

# 数据库设计

见附录D

附录C 软件详细设计报告文档

[1. 引言 34](#_Toc88679123)

[1.1 编写目的 34](#_Toc88679124)

[1.2 项目风险 34](#_Toc88679125)

[1.3 预期读者和阅读建议 34](#_Toc88679126)

[1.4 参考资料 34](#_Toc88679127)

[2. 支撑环境 35](#_Toc88679128)

[2.1 数据库管理系统 35](#_Toc88679129)

[2.2 开发工具、中间件以及数据库接口 35](#_Toc88679130)

[2.3 网络环境 35](#_Toc88679131)

[3. 子系统详细设计（例如：Room数据库处理待办事项） 35](#_Toc88679132)

# 引言

本详细设计报告主要帮助预期读者对象，熟悉和了解这份文档所编写的目的、存在风险及产品范围。

## 编写目的

本详细设计报告基于《成电微记需求分析报告》，《成电微记概要设计报告》，《成电微记数据库设计说明书》编写。本报告在概要设计基础上编址，将为下一阶段程序编码工作提供依据，作为项目验收的重要内容和后期维护的必备资料。

## 项目风险

本软件开发项目的全部风险承担者及风险承担职责如下：

* 任务提出者：需要对整个项目范围，项目目标及项目全国过程掌控负责
* 项目管理者：需要对整个项目进度及项目实现负责
* 软件开发者：需要对项目是实现负责
* 测试人员 ：需要对项目各个功能环节，进行黑白盒测试，对测试用例，测试结果负责

## 预期读者和阅读建议

本软件系统详细设计报告所针对的可能的读者包括：

* 开发人员；
* 项目经理；
* 测试人员；
* 文档编写人员；

## 参考资料

本详细设计报告参考文献及资料：

* 用户界面风格；
* 开发本项目时所要用到的标难；
* 使用实例文档；
* 本软件系统详细设计报告中所引用的文件、资料；
* 相关软件系统详细设计报告；

# 支撑环境

## 数据库管理系统

描述数据库管理系统、以及安装配置情况，需要描述的内容可能包括：

* 产品名称以及发行厂商

数据库：Room 发行厂商：Google

* 版本号

2.3.0

* 语言或代码集

Java

* 安装位置

APP内

## 开发工具、中间件以及数据库接口

描述所选用的工具软件和中间件的名称、版本号，以及开发工具与数据库或者中间件接口的情况。如果使用了多种开发工具、辅助开发工具、第三方软件部件、多种中间件、多种接口、等答应该逐项分别描述，并且说明每一项的适用范围。需要描述的内容可能包括：

* 产品名称以及发行厂商

开发工具：Android Studio 发行厂商：Google

* 版本号

4.2.2

* 语言或代码集

Java / Kotlin

* 数据库接口名称

Room

* 数据库接口方式

通过封装实体类Entity、数据访问对象Data Access Object、RoomDatabase实现类来使用数据库的功能

## 网络环境

通畅的网络环境

# 子系统详细设计（例如：Room数据库处理待办事项）

软件部件表：

表2-1 软件部件表

|  |  |
| --- | --- |
| 部件名称 | Room数据库处理待办事项 |
| 部件调用者 | TodoFragment： 从数据库获取数据；  AddDialogFragment： 往数据库中添加数据；UpdateDialogFragment：更新数据库中的数据。 |
| 部件被调用者 | TodoDaoService |
| 部件入口参数 | getInstance():无参方法，使用单例模式封装，返回唯一对象；  insert/delete/update(TodoEntity):参数均为一个数据表对象，作用分别是插入数据、根据主键删除数据、根据主键更新数据  loadAllTodo():无参方法，从数据库中加载所有的待办事项数据 |
| 部件标识形式 | Todo相关功能 |
| 运行环境 | Android 10及以上 |

流程描述：

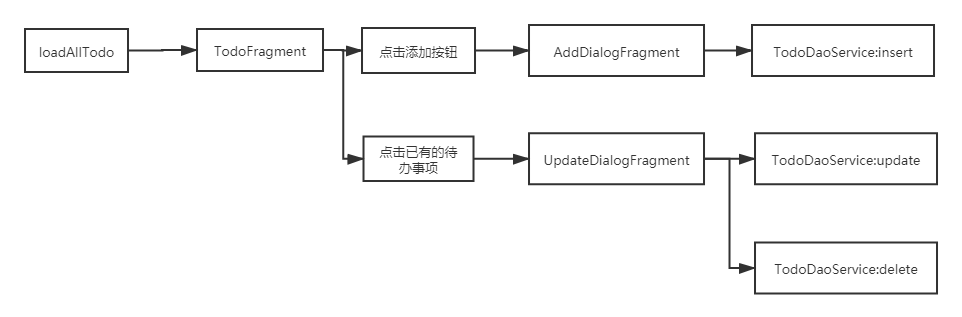


图2-1 Room数据库处理待办事项函数调用关系图

各函数作用：

* loadAllTodo：使用TodoDaoService提供的loadAllTodo()方法来完成从数据库获取数据的功能；
* TodoFragment：TodoFragment为待办事项功能的主页，该页面负责加载所有待办事项，并展示
* AddDialogFragment：点击页面添加按钮可以弹出添加新的待办事项的弹窗AddDialogFragment
* （TodoDaoService：insert）：AddDialogFragment弹窗通过调用TodoDaoService的insert(TodoEntity)方法来实现往数据库插入新的待办事项的功能
* UpdateDialogFragment：点击每一条待办事项可以弹出修改待办事项的弹窗UpdateDialogFragment(已完成的待办事项只支持删除，并不支持修改)
* (TodoDaoService：updata):该弹窗通过调用TodoDaoService对象的update(TodoEntity)实现更新
* (TodoDaoService：delete):该弹窗通过调用TodoDaoService对象的delete(TodoEntity)实现删除。

附录D 测试用例报告文档

[0. 文档介绍 4](#_Toc75634798)2

[0.1 文档目的 4](#_Toc75634799)2

[0.2 文档范围 4](#_Toc75634800)2

[0.3 读者对象 4](#_Toc75634801)2

[0.4 术语与缩写解释 4](#_Toc75634803)2

[1. 接口－路径测试用例 4](#_Toc75634804)2

[1.1 被测试对象（单元）的介绍 4](#_Toc75634805)2

[1.2 测试范围与目的 4](#_Toc75634806)5

[1.3 测试环境与测试辅助工具的描述 4](#_Toc75634807)5

[1.4 测试驱动程序的设计 4](#_Toc75634808)5

[1.5 接口测试用例 4](#_Toc75634809)5

[1.6 路径测试的检查表 47](#_Toc75634810)

[2. 功能测试用例 48](#_Toc75634811)

[2.1 被测试对象的介绍 48](#_Toc75634812)

[2.2 测试范围与目的 48](#_Toc75634813)

[2.3 测试环境与测试辅助工具的描述 49](#_Toc75634814)

[2.4 测试驱动程序的设计 4](#_Toc75634815)9

[2.5 功能测试用例 49](#_Toc75634816)

[2.6 测试步骤及预期结果 49](#_Toc75634816)

[3. 图形用户界面测试用例 4](#_Toc75634829)9

[3.1 被测试对象的介绍 5](#_Toc75634830)0

[3.2 测试范围与目的 5](#_Toc75634831)1

[3.3 测试环境与测试辅助工具的描述 5](#_Toc75634832)1

**0. 文档介绍**

**0.1 文档目的**

本软件测试用例报告主要帮助预期读者对象，熟悉和了解这份文档所编写的目的、存在风险及产品范围。

**0.2 文档范围**

本文档包含测试范围为：接口路径测试，功能测试，健壮性测试，性能测试，图形用户界面测试。

**0.3 读者对象**

产品经理，开发人员，测试人员等相关人员

**0.4 术语与缩写解释**

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| 黑盒测试 | 在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试， |
| 白盒测试 | 清楚盒子内部的东西以及里面是如何运作的情况下进行程序测试 |
| 等价类划分 | 将程序所有可能的输入数据（有效的和无效的）划分成若干个等价类。 |

**1. 接口－路径测试用例**

**1.1 被测试对象（单元）的介绍**

测试用于判断点击事件后是否需要隐藏光标软键盘的函数。该函数首先判断是否是点击事件，若为点击事件则获取当前焦点所在，若当前焦点处于EditText控件内，则隐藏光标和键盘，同时保存内容。

函数具体代码如下：代码1-1

|  |
| --- |
| /\*\*  \* 获取点击事件  \*/  @Override  public boolean dispatchTouchEvent(MotionEvent ev) {  if (ev.getAction() == MotionEvent.ACTION\_DOWN) {  View view = getCurrentFocus();  //如果点击在非EditView位置，隐藏光标和键盘，同时保存修改后的内容  if (isHideInput(view, ev)) {  HideSoftInput(view.getWindowToken());  view.clearFocus();  sharedPreferencesUtils.putString(view.getId() + "",((EditText)view).getText().toString());  }  }  return super.dispatchTouchEvent(ev);  }  /\*\*  \* 判定是否需要隐藏  \*/  private boolean isHideInput(View v, MotionEvent ev) {  if (v != null && (v instanceof EditText)) {  int[] l = {0, 0};  v.getLocationInWindow(l);  int left = l[0], top = l[1], bottom = top + v.getHeight(), right = left + v.getWidth();  if (ev.getX() > left && ev.getX() < right && ev.getY() > top && ev.getY() < bottom) {  return false;  } else {  return true; }  }  return false;  } |

 函数流程图如下：

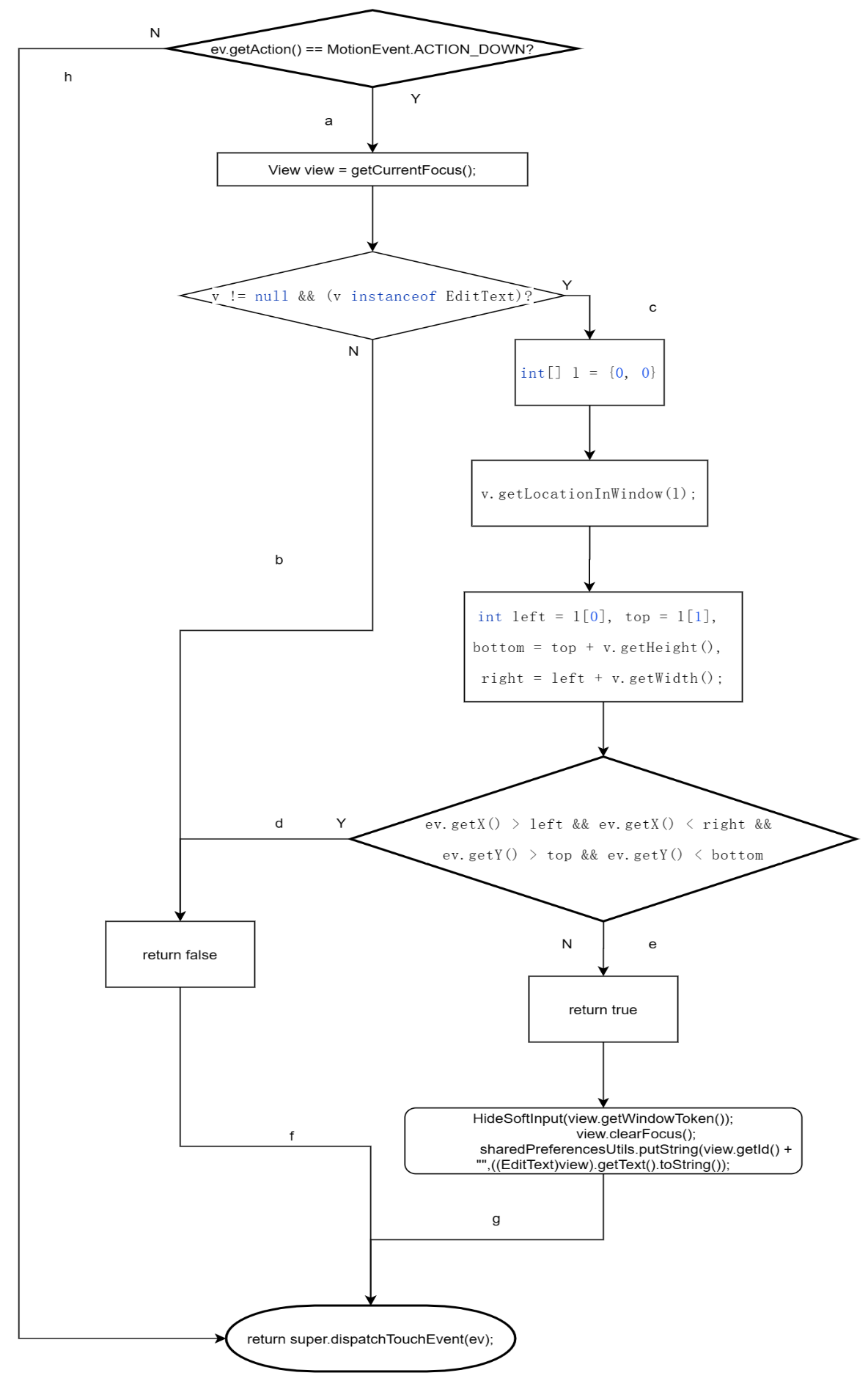


图1-1 该函数流程图

**1.2 测试范围与目的**

 测试该函数能否正确执行

**1.3 测试环境与测试辅助工具的描述**

 测试环境：安卓操作系统

**1.4 测试方法**

 白盒测试：语句覆盖，分支覆盖，条件覆盖，路径覆盖

**1.5 接口测试用例**

 1.5.1语句覆盖

分析：

设计足够多的测试用例，使得程序中每条语句至少被执行一次，因此设计用例时，需要注意将所有的语句都覆盖到，当已有的测试路径无法覆盖所有语句时，需要再增加测试路径，使其能够覆盖所有的语句。

测试用例如下：

表1-1 语句覆盖测试用例

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输入项 | | | | | | 测试路径 | 测试结果 |
| ev.getAction() | v | Left | Right | Top | Bottom |  |  |
| MotionEvent.ACTION\_DOMN | v=null | \ | \ | \ | \ | a,b,d,f | 不作反应 |
| MotionEvent.ACTION\_DOMN | v！=null  &&v instanceof EditText | <ev.getX( ) | >ev.getX( ) | <ev.getY( ) | >ev.getY( ) | c,e,f | 隐藏软键盘光标，并保存内容 |

 1.5.2分支覆盖

分析：

在遇到判定结构时，设计的测试用例需要使程序中每个判定的"真"和"假"至少执行一次。图中共有三个判断结构，从第一个判断分支开始，设置测试用例来将每一个判断结构真假都执行一次，知道每一个判断分支都处理完成。

当ev.getAction()为MotionEvent.ACTION\_DOMN时，第一个判断结构为真，经过路径a，v=null，第二个判断分支的结果为假，经过路径b，f。

当ev.getAction()为MotionEvent.ACTION\_MOVE时，第一个判断结构为假，经过路径h。由此第一个判断分支真假都经历一次。

接下来处理第二个判断分支，第二个判断结果为假的路径在第一个测试用例中已经走过一次，因此仅需要设置条件v！=null&&v instanceof EditText 使得该判断分支为真，经过路径c，再设置条件使得第三个判断分支结果为假，经过路径e，g。

最后处理第三个判断分支，之前第三个判断分支结果为假已经执行过一次，因此只需要设置条件使得第三个判断分支结果为真再执行一次即可完成分支覆盖的测试。

测试用例如下：

表1-2 分支覆盖测试用例

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输入项 | | | | | | 测试路径 | 测试结果 |
| ev.getAction() | v | Left | Right | Top | Bottom |  |  |
| MotionEvent.ACTION\_DOMN | v=null | \ | \ | \ | \ | a,b,f | 不作反应 |
| MotionEvent.ACTION\_DOMN | v！=null  &&v instanceof EditText | <ev.getX( ) | >ev.getX( ) | <ev.getY( ) | >ev.getY( ) | a,c,e,g | 隐藏软键盘光标，并保存内容 |
| MotionEvent.ACTION\_MOVE | \ | \ | \ | \ | \ | h | 不作反应 |
| MotionEvent.ACTION\_DOMN | v！=null  &&v instanceof EditText | <ev.getX( ) | <ev.getX( ) | <ev.getY( ) | >ev.getY( ) | a,c,d,f | 隐藏软键盘光标，并保存内容 |

1.5.3 条件覆盖

分析：

选择足够多的用例，使得程序中每个判定的每个条件取得各种可能的结果。从第一个判断分支开始，将一个分支内的所有条件的各种结果都取完后，再设置用例来覆盖下一个分支的条件的所有可能。

第一个判断分支仅有一个条件，将该条件真假各取一次即可。先取ev.getAction()为MotionEvent.ACTION\_DOMN，则第一个判断内条件为真，再取v=null，使得第二个判断分支内的第一个条件为假，经过路径a，b，f。第二个判断分支的条件真已经取过，此次取ev.getAction()为MotionEvent.ACTION\_MOVE，则第一个判断内条件为假，经过路径h。由此第一个判断分支内的条件的可能的结果已经取完。

第二个判断分支有两个条件，在上一条路径测试中已经将第二个判断分支的第一个条件取为真，则还需设置用例使得该条件结果为真，即v！=null&&v instanceof EditText ，第二个条件为真，接下来是第三个判断分支，将left，tight，top，bottom全部取为真，则经过路径a，c，e，g。

第二个判断分支的第二个条件还有结果为假时没有测试，因此在第一个判断分支结果为真时，取v！=null&& v 不为EditText 控件，使得第二个判断分支的第二个条件为假，经过路径a，b，f。

第一个第二个判断分支内的每个条件的可能结果都取过后，接下来完成第三个判断分支内条件的测试。在之前的测试中，第三个判断分支内的四个条件结果为真都已经测试过，因此此次取值使得四个条件结果都为假，经过路径a，c，d，f。

测试用例如下：

表1-3 条件覆盖测试用例

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输入项 | | | | | | 测试路径 | 测试结果 |
| ev.getAction() | v | Left | Right | Top | Bottom |  |  |
| MotionEvent.ACTION\_DOMN | v=null | \ | \ | \ | \ | a,b,f | 不作反应 |
| MotionEvent.ACTION\_MOVE | \ | \ | \ | \ | \ | h | 不作反应 |
| MotionEvent.ACTION\_DOMN | v！=null  &&v instanceof EditText | <ev.getX( ) | >ev.getX( ) | <ev.getY( ) | >ev.getY( ) | a,c,e,g | 隐藏软键盘光标，并保存内容 |
| MotionEvent.ACTION\_DOMN | v！=null  && v 不为EditText 控件 | \ | \ | \ | \ | a,b,f | 隐藏软键盘光标，并保存内容 |
| MotionEvent.ACTION\_DOMN | v！=null  &&v instanceof EditText | >ev.getX( ) | <ev.getX( ) | >ev.getY( ) | <ev.getY( ) | a,c,df | 隐藏软键盘光标，并保存内容 |

1.5.4 路径覆盖

分析：选择足够多的测试用例，使程序中每条路径至少执行一次

测试用例如下：

从ev.getAction()为MotionEvent.ACTION\_DOMN开始，v=null，经过路径a，b，f。

取ev.getAction()为MotionEvent.ACTION\_MOVE，经过路径h。

此时还有路径c，d，e，g没有走过，因此取ev.getAction()为MotionEvent.ACTION\_DOMN，v！=null

&&v instanceof EditText ，并使第三个判断分支结果为真，经过路径a，c，e，g。

此时还有路径d，f没有测过。取ev.getAction()为MotionEvent.ACTION\_DOMN，v！=null&&v instanceof EditText ，第三个判断分支结果为假，经过路径a，c，d，f。

至此所有路径都经过过至少一次。

表1-4 语句覆盖测试用例

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输入项 | | | | | | 测试路径 | 测试结果 |
| ev.getAction() | v | Left | Right | Top | Bottom |  |  |
| MotionEvent.ACTION\_DOMN | v=null | \ | \ | \ | \ | a,b,f | 不作反应 |
| MotionEvent.ACTION\_MOVE | \ | \ | \ | \ | \ | h | 不作反应 |
| MotionEvent.ACTION\_DOMN | v！=null  &&v instanceof EditText | <ev.getX( ) | >ev.getX( ) | <ev.getY( ) | >ev.getY( ) | a,c,e,g | 隐藏软键盘光标，并保存内容 |
| MotionEvent.ACTION\_DOMN | v！=null  &&v instanceof EditText | >ev.getX( ) | <ev.getX( ) | >ev.getY( ) | <ev.getY( ) | a,c,df | 隐藏软键盘光标，并保存内容 |

**1.6 路径测试的检查表**

表1-5 路径测试检查表

|  |  |
| --- | --- |
| **检查项** | **结论** |
| 数据类型问题  （１）变量的数据类型有错误吗？  （２）存在不同数据类型的赋值吗？  （３）存在不同数据类型的比较吗？ | 无 |
| 变量值问题  （１）变量的初始化或缺省值有错误吗？  （２）变量发生上溢或下溢吗？  （３）变量的精度不够吗？ | 无 |
| 逻辑判断问题  （１）由于精度原因导致比较无效吗？  （２）表达式中的优先级有误吗？  （３）逻辑判断结果颠倒吗？ | 无 |
| 循环问题  （１）循环终止条件不正确吗？  （２）无法正常终止（死循环）吗？  （３）错误地修改循环变量吗？  （４）存在误差累积吗？ | 无 |
| 内存问题  （１）内存没有被正确地初始化却被使用吗？  （２）内存被释放后却继续被使用吗？  （３）内存泄漏吗？  （４）内存越界吗？  （５）出现野指针吗？ | 无 |
| 文件I/O问题  （１）对不存在的或者错误的文件进行操作吗？  （２）文件以不正确的方式打开吗？  （３）文件结束判断不正确吗？  （４）没有正确地关闭文件吗？ | 无 |
| 错误处理问题  （１）忘记进行错误处理吗？  （２）错误处理程序块一直没有机会被运行？  （３）错误处理程序块本身就有毛病吗？如报告的错误与实际错误不一致，处理方式不正确等等。  （４）错误处理程序块是“马后炮”吗？如在被它被调用之前软件已经出错。 | 无 |

**2. 功能测试用例**

本部分采用黑盒测试

**2.1 被测试对象的介绍**

 账单部分支出金额输入

**2.2 测试范围与目的**

 测试范围：账单部分支出金额输入

测试目的：测试金额输入功能是否正常

**2.3 测试环境与测试方法**

 测试环境：安卓操作系统

 测试方法：等价类划分

**2.4 用例描述及前置条件**

|  |  |
| --- | --- |
| 用例描述 | 输入支出金额 |
| 前置条件 | 点击账单--支出，点击金额输入框 |

**2.5 功能测试用例**

表2-1 功能测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 输入条件 |  | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 第一个字符为”+” | 小数点（1个）+两位以内数字 | “+.00”~”+.99” | 1 | 第一个字符为”+“或者小数点或者数字，且小数点后有两位以上的数字 | 7 |
| 0+数字 | 0加任意长度数字 | 2 | 有一个以上的小数点 | 8 |
| 0+小数点（1个）+两位以内数字 | “0.00~0.99” | 3 | 字母 | 9 |
| 第一个字符为”.”(小数点) | 小数点（1个）后两位以内数字 | “.00”~”.99” | 4 | 非字母，数字，”+””.”的字符 | 10 |
| 开头为一个及以上的0 | 加数字 | 0开头任意长度数字 | 5 |  |  |
| 加数字，小数点（1个）及小数点后两位以内数字 | 0开头任意长度数字+“.”+两位数字 | 6 |  |  |

**2.6 测试步骤及预期结果**

表2-2 测试步骤及预期结果

|  |  |  |
| --- | --- | --- |
| 步骤描述 | 输入数据 | 预期结果 |
| 1.键入第一个字符 | + | 金额显示为”0.00” |
|  | 0 | 金额显示为”0.00” |
|  | “.”(小数点) | 金额显示为”0.00” |
|  | 非零数字 | 显示输入数字 |
|  | 除数字，运算符”+”，小数点以外的字符 | 提示请输入正确金额 |
| 2.键入剩下的字符 | 数字或小数点（仅1个），且仅为小数点后两位 | 显示输入内容 |
|  | 数字或小数点（仅1个），，且小数点后超过两位 | 显示输入内容，小数点两位以后的内容不显示 |
|  | 字符串中存在多个小数点 | 提示请输入正确金额 |
|  | 字符串中存在小数点，数字以外的字符 | 提示请输入正确金额 |

1. **图形用户界面测试用例**

**3.1 被测试对象的介绍**

测试界面的显示内容与控件。

**3.2 测试范围**

  测试范围：所有界面的所有控件

**3.3 用户界面测试的检查表**

表3-1 用户界面测试检查表

|  |  |
| --- | --- |
| 检查项 | 评价 |
| 窗口切换、移动、改变大小时正常吗？ | 正常 |
| 各种界面元素的文字正确吗？（如标题、提示等） | 正确 |
| 各种界面元素的状态正确吗？（如有效、无效、选中等状态） | 正确 |
| 各种界面元素支持键盘操作吗？ | 支持 |
| 各种界面元素支持触摸操作吗？ | 正常 |
| 对话框中的缺省焦点正确吗？ | 正常 |
| 数据项能正确回显吗？ | 正常 |
| 对于常用的功能，用户能否不必阅读手册就能使用？ | 正常 |
| 执行有风险的操作时，有“确认”、“放弃”等提示吗？ | 有 |
| 操作顺序合理吗？ | 合理 |
| 有联机帮助吗？ | 无 |
| 各种界面元素的布局合理吗？美观吗？ | 合理，美观 |
| 各种界面元素的颜色协调吗？ | 协调 |
| 各种界面元素的形状美观吗？ | 美观 |
| 字体美观吗？ | 美观 |
| 图标直观吗？ | 直观 |