记账APP设计文档

**1 概述**

**1.1 目的**

开发一款可以具有本地存储、数据自动备份的记账app。这款app可以完成流水记账，消费可视化分析等功能，并通过这些功能帮助app使用者合理规划消费。

**1.2 背景**

现在的用户需要一款软件来记录每日的收支情况，以统计每日的花销。

**1.3 术语定义及说明**

流水图：描述收支的流水

用户表：存放用户数据

账单表：存放用户记录的账单详细信息

标签表：存放系统提供的类型以及用户自定义的标签信息

预算表：存放用户设置的年预算、月预算以及各标签类别预算

**1.4 项目概述**

该记账APP通过实现系统账户登录功能、收支记账功能、统计功能、图表分析功能来实现记录日常生活中的账单支出和收入，和月底的账单整理功能

**1.5 参考资料**

《软件工程-实践者的研究方法》，机械工业出版社

《软件工程实例教程》，清华大学出版社

**2 设计概述**

**2.1 需求**

2.1.1 系统账户登录

用户首次进入APP时，应用需要提供一些初始化设置。同时应具备文本密码登录、文本密码登录以及密码错误提醒功能。

2.1.2 收支记录、记账

用户可以为自己的账本建立不同的账户，同时可以在记账时选择对应的账户。记账时，用户也可以选择账单对应的资金流动类型，并进行一级、二级记账类型的选择及定义。在用户记账时，APP会自动填写记账时的即时系统时间作为默认值，用户也可以自行修改时间。

2.1.3 数据统计

在统计界面，用户可以看到账单的所有账户的当前资产信息，也能分展示该账户的资产变化情况。用户也可以在某一个给定的时间范围内查看整个账单的所有账单信息

2.1.4 图表分析

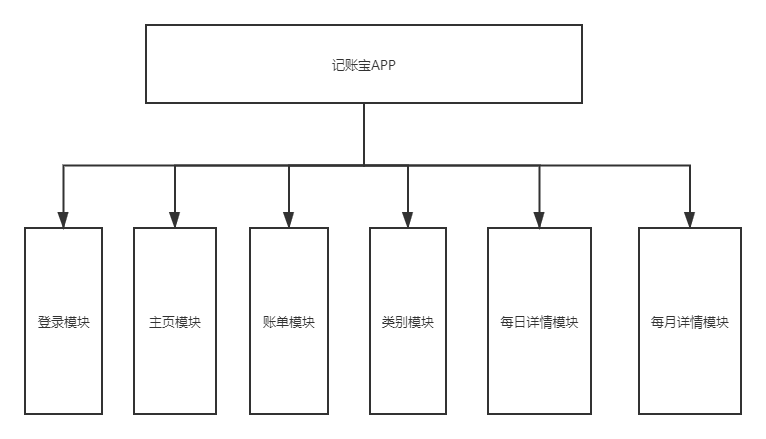
用户可以按照单条账单的一级的分类或者按照指定成员分类，来查看给定范围内的资产变化图表。用户也能选定一个时间范围，展示出对应时间内的统计数据图表。

**2.2 总体结构**

登陆->记账->统计->图表分析

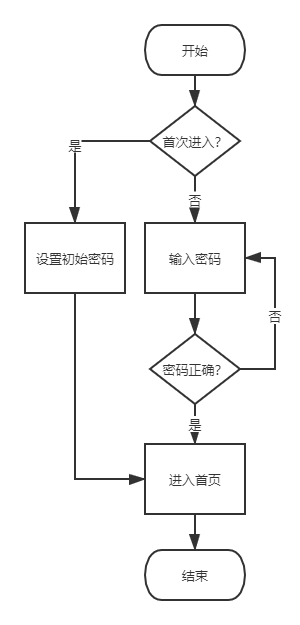
**3 功能模块划分**

**3.1 模块划分**



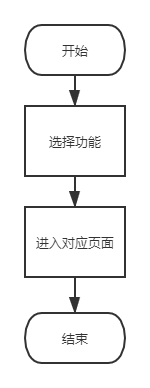
**3.2 登录模块**

功能：确认用户身份，作为启动钥匙进入记账系统



**3.3主页模块**

功能：作为目录页面作为连接其他页面和功能的纽带



接口按钮定义：

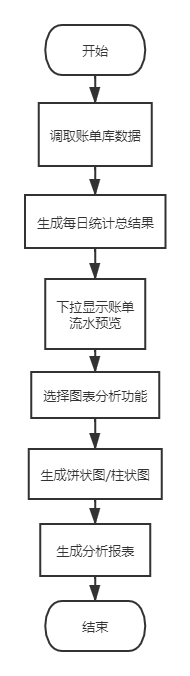
“记一笔”：进入记账页面（即账单模块），填写对应内容完成一笔记账。

“总览”：进入账目统计总览页面，下转每日详情和每月详情模块。

图表分析接口：转入图表分析模块，生成饼状图/柱状图形成账单分析。

**3.4 每日详情模块（或称：流水模块）**

功能：显示用户本日的收入和支出的统计概况和详细账单



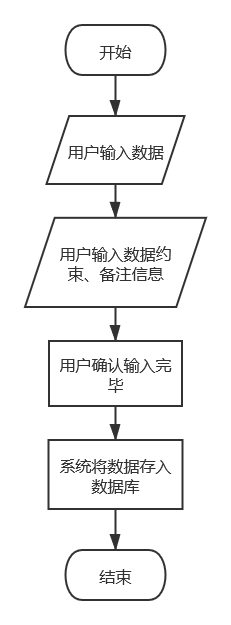
接口按钮：

下拉栏：账单流水预览，点击进入单笔详情

图表分析：将统计结果转化生成图表，生成分析

**3.5 账单模块**

功能：允许用户向账本添加一笔账单

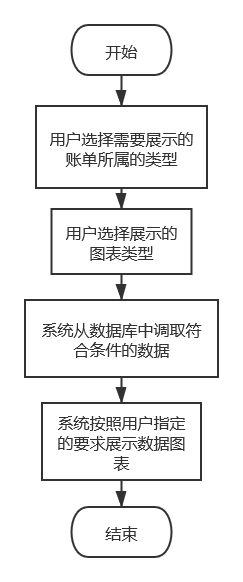


接口按钮：

保存：点击后，系统会进行存储操作，将用户输入的账单存入数据库

**3.6 类别模块（或称：图表展示模块）**

功能：按照选定的一级/二级类型，展示对应类型下的账单统计数据

****

接口按钮：

类型名：用户通过点击类型来进入该类型账单的详细统计界面

图表类型：选择不同的图表类型，则系统会展示不同形式的图表（如饼图、折线图等）

**3.7 每月详情模块（或称：流水模块）**

功能：显示用户按月统计的收入和支出的统计概况和详细账单

流程大体类似3.4，仅仅是时间跨度不同，此处略去。

**4 数据库系统设计**

**4.1 数据库种类及特点**

本次数据库调用Mysql数据库进行实现。该数据库为关系型数据库，关系型数据库是指采用了关系模型来组织数据的数据库，而关系模型是由二维表及其联系组成的数据组织。该数据库的特点有：

①易于维护，都是使用表结构，格式一致。

②使用方便，SQL语句通用，可用于复杂查询。

③复杂操作，支持SQL，可用于一个表以及多个表之间非常复杂的查询。

**4.2数据库逻辑结构**

1.MySQL逻辑架构整体分为三层，最上层为客户端层，并非MySQL所独有，诸如：连接处理、授权认证、安全等功能均在这一层处理。

2.MySQL大多数核心服务均在中间这一层，包括查询解析、分析、优化、缓存、内置函数(比如：时间、数学、加密等函数)。所有的跨存储引擎的功能也在这一层实现：存储过程、触发器、视图等。

3.最下层为存储引擎，其负责MySQL中的数据存储和提取。和Linux下的文件系统类似，每种存储引擎都有其优势和劣势。中间的服务层通过API与存储引擎通信，这些API接口屏蔽了不同存储引擎间的差异。不同引擎只会简单的响应上层服务器的请求，而不会相互通信

**4.3访问数据方式**

本项目运用的Mysql通过JDBC接口实现后端Java语言对Mysql数据库的使用，其中通过SQL语句作为媒介，实现对数据的增、删、查、改。

**5 信息编码设计**

**5.1 代码结构设计**

本次项目后端使用Java语言进行编写，前段使用xml语言进行前段页面的编写。并且要求以高内聚，低耦合的原则进行编码，以实现程序的简洁。

**5.2 命名规范要求**

本次项目代码的命名要求贴近该变量所代表的实际含义进行命名，使得代码简洁明了，便于阅读。