

目录

前 言	4
1 可行性分析	5
1.1 背景介绍	5
1.2 经济性与实用性分析	5
1.3 技术可行性分析	5
1.4 结论	6
2 需求分析	7
2.1 功能需求	7
2.2 用例分析	7
2.2.1 个人中心用例分析	8
2.2.2 用户用例分析	10
2.3 环境需求	14
2.3.1 软件环境要求	14
2.3.2 开发环境要求	15
3 系统设计	16
3.1 系统目标	16
3.2 系统功能结构	16
3.3 数据库设计	19
3.3.1 数据库需求分析	19
3.3.2 实体关系图（E-R 图）	20
3.3.3 数据表设计	20
4 详细设计	23
4.1 注册	23
4.1.1 界面设计	23
4.1.2 设计思想	23
4.1.3 关键代码	24

4.2 登录	24
4.2.1 界面设计	25
4.2.2 设计思想	25
4.2.3 关键代码	26
4.3 主界面	26
4.3.1 界面设计	26
4.3.2 设计思想	27
4.3.3 关键代码	27
4.4 备忘录管理	28
4.4.1 界面设计	28
4.4.2 设计思想	29
4.4.3 关键代码	29
4.5 文件管理	30
4.5.1 界面设计	30
4.5.2 设计思想	30
4.5.3 关键代码	31
4.6 日程安排管理	32
4.6.1 界面设计	32
4.6.2 设计思想	33
4.6.3 关键代码	33
4.7 记账本管理	34
4.7.1 界面设计	34
4.7.2 设计思想	34
4.7.3 关键代码	35
4.8 个人中心	35
4.8.1 界面设计	36
4.8.2 设计思想	36
4.8.3 关键代码	36
5 软件测试	37

5.1 备忘录管理测试.....	37
5.1.1 测试用例.....	37
5.1.2 界面测试.....	38
5.2 文件管理测试	40
5.2.1 测试用例.....	40
5.2.2 界面测试.....	40
5.3 日程安排测试	42
5.3.1 测试用例.....	42
5.3.2 界面测试.....	42
5.4 记账本测试.....	44
5.4.1 测试用例.....	44
5.4.2 界面测试.....	45
5.5 个人中心测试.....	46
5.5.1 测试用例.....	46
5.5.2 界面测试.....	47
5.6 测试结论	47
6 结论	48

前 言

以前我们选择将生活中的种种事情记录在纸张上，用笔写下心中的安排与计划。可这样却有一个很大的问题，纸张很容易被遗失，这样使得我们往往找不到记录的重要信息。而将其记录在本子上固然是要安全些，可是大部分人不会将本子随身携带，这造成了当我们需要查看这些记录，却没有携带在身边。将信息而记录在手机却解决了这些问题。随着当经互联网的高速发展，手机已成为人们身上必不可少的随身物品。无论去哪，我们都带着它，手机也成为了我们存储和查找信息的首选。

现在的手机上备忘录，日程安排等 APP 层出不穷，但大部分都只是单独的 APP。而我设计的这个生活助手 APP 则是将备忘录、文件管理、日程安排和记账功能^[1]结合起来，为用户提供一个更集中，更方便简洁的生活记录 APP。

在 Android 安全措施中，默认的情况下，应用程序是没权限对用户、其他程序以及操作系统进行操作的。这包括对用户的隐私数据进行读/写，例如：手机内存、手机号码等，网络访问以及对设备的活动进行保持等。因此，应用程序进程属于安全的黑盒子，不会对其他应用程序造成干扰，除了对其进行声明权限，而生活助手通过 Android 开发管理系统数据，使用户可以安全，便捷的使用系统。

在下面，我将对我的这个生活助手 APP 进行分析、设计和实现。

1 可行性分析

1.1 背景介绍

本系统主要是利用 Android 应用程序设计实现对程序开发，利用 Java 编程语言+Linux 内核+ SQLite 数据库开发的设计模式，实现各个记录的插入、索引和分类。

1.2 经济性与实用性分析

现在的手机上备忘录，日程安排等 APP 层出不穷，但大部分都只是单独的 APP，而我设计的这个生活助手 APP 则是将这些包括记账功能结合起来，为用户提供一个更集中，更方便简洁的生活记录 APP,我将基于 Android 平台对应用程序的开发、研究展开简单的分析,以期提升人们的智能设备操作体验。

Android 手机正成为全球使用量最多的电子产品。不管是幼儿还是老人也开始习惯使用应用程序，如果可能的话，他们希望在移动设备上找到所有可用的东西。相较传统的纸上记录方法，手机记录更加简单方便，省去额外携带和寻找的功夫。

1.3 技术可行性分析

本系统是个小型的 APP，开发难度适中。通过对诸多开发工具以及指导老师的建议，最终确定采用 Android studio 作为开发工具，通过真机和模拟器来测试运行效果，SQLite 作为关系数据库，以 Android 程序应用开发技术为基础进行的开发设计。

本文设计了基于 Android 的生活助手 APP,从层次上可以分为数据层、服务层和交互层。使用 SQLite 作为数据库，是无服务器的，SQLite 不需要配置，这意味着不需要安装或管理，是非常小的轻量级数据库。Android 系统是一款真正的开放的系统，开发方便，开发人员只需要写几行代码就可以将功能强大的组件添加到自己的程序中。Android Studio 是谷歌推出的一个 Android 集成开发工具，基于 IntelliJ IDEA。相比其他软件，Android 稳定速度快，拥有功能强大的 UI 编辑器，完善的插

件管理和支持多种代码管理工具使其使用起来更加方便快捷，整合了 Gradle 构建工具。

综上所述，本系统在技术上是能够实现的、是完全可行的。

1.4 结论

我决定采用 Java 编程语言+Linux 内核+ SQLite 数据库在 Android studio 平台进行系统的开发，这种方案可以充分实现用户的需求，亦不需要再额外增添硬件设备，产生相关经济支出，在经济和技术上具备开发可行性。

2 需求分析

2.1 功能需求

随着时代的发展，手机已成为我们身边必不可少的物品，大部分人都会随身携带手机，生活助手能够让我们更方便的记录和规划自己的生活,也能提醒我们开始工作和，避免信息的遗漏和地点的限制。

经过调研、分析和设计，本系统主要具有以下业务功能：

（1）注册登录功能：未注册的用户可通过注册功能成为注册用户。注册后用户可以登录系统进行操作。

（2）备忘录功能： 备忘录功能包括便签的添加、修改和删除，用户还可以通过关键字和备忘录便签类型查询相关便签。

（3）文件管理功能：可查看手机内存中的 ppt、txt 和图片等文件以及文件夹，对文件进行删除操作。还可以新建文件夹和关键字搜索。

（4）日程安排：安排每日任务，对每个完成任务进行打卡，通过关键字查询相关安排。

（5）记账本：添加收支记录，统计每月账单，显示每月消费趋势。

（6）个人中心：可以对用户数据进行操作。

2.2 用例分析

通过对系统业务功能的分析，接下来通过用例模型来设计需求，使开发功能达成共识。通过分析，可以获得用例与参与者之间的关系，可以绘制出以下用例模型。

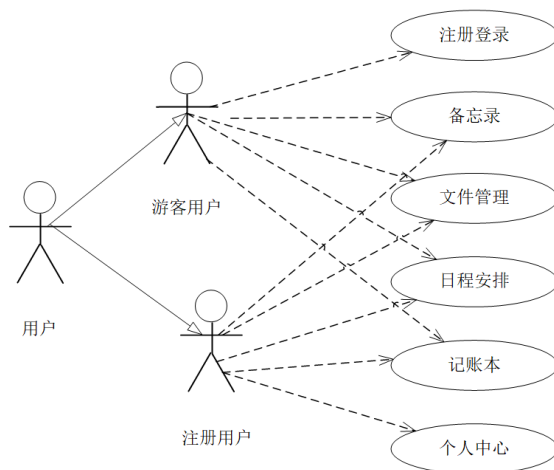


图 2-1 系统总用例图

2.2.1 个人中心用例分析

在用户通过登录注册后方可使用个人中心功能，个人中心功能主要是对用户账号的操作。具体如图 2-2 所示。

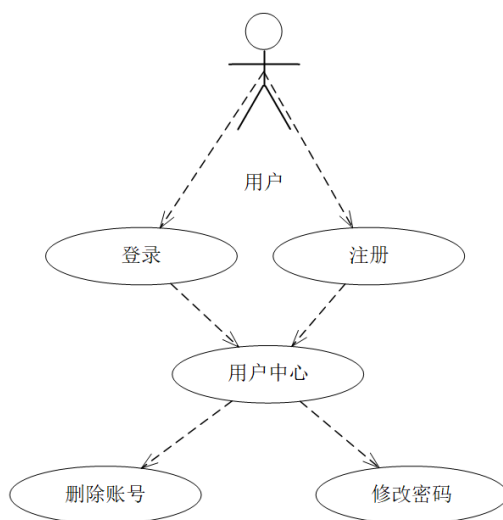


图 2-2 个人中心用例图

（1）登录用例，用户登录功能主要是通过比对输入框内容和数据库中的数据判断用户是否能够成功登录，具体如表 2-1 所示。

表 2-1 用户登录用例描述

用例名称	用户登录
功能描述	本用例主要用于用户登录
输入	用户账号，用户密码
处理	系统比对信息，出现不符合，则报错；否则审核通过

输出	输出登录成功
前提条件	用户存在
事件流	基流 用户想要登录系统时，用例启动 （1）该用户提出所要执行的操作 如果用户成功登录，则执行分支流 S-1:登录成功 如果用户登录失败，则执行分支流 S-2:重新登录 分支流 S-1:登录成功 进入系统主界面 S-2:重新登录 用户操作完后，用例结束
后置条件	用例成功后，用户接入系统

（2）用户注册用例主要涉及用户注册功能，在当前设备不存在用户时方可进行用户注册，具体如表 2-2 所示：

表 2-2 用户注册用例描述

用例名称	用户注册
功能描述	本用例主要用于用户注册
输入	用户账号，用户密码
处理	检查用户信息是否符合要求
输出	用户注册成功
前提条件	当前设备无注册用户
事件流	基流 用户要求注册使用系统时，用例启动 （1）系统跳转到用户注册页面，用户根据提示输入用户名和密码 （2）用户输入用户名和密码，并通过验证码验证 （3）系统检查信息是否有效（E-1），并将信息存入数据库 替代流 E-1：如果用户输入的信息无效，系统则提示用户输入错误，用户重新输入信息，或取消注册，用例结束
后置条件	用例成功后，添加新用户

2.2.2 用户用例分析

在本系统中，用户能够使用备忘录、文件管理、日程安排和记账本功能，具体如图 2-3 所示：

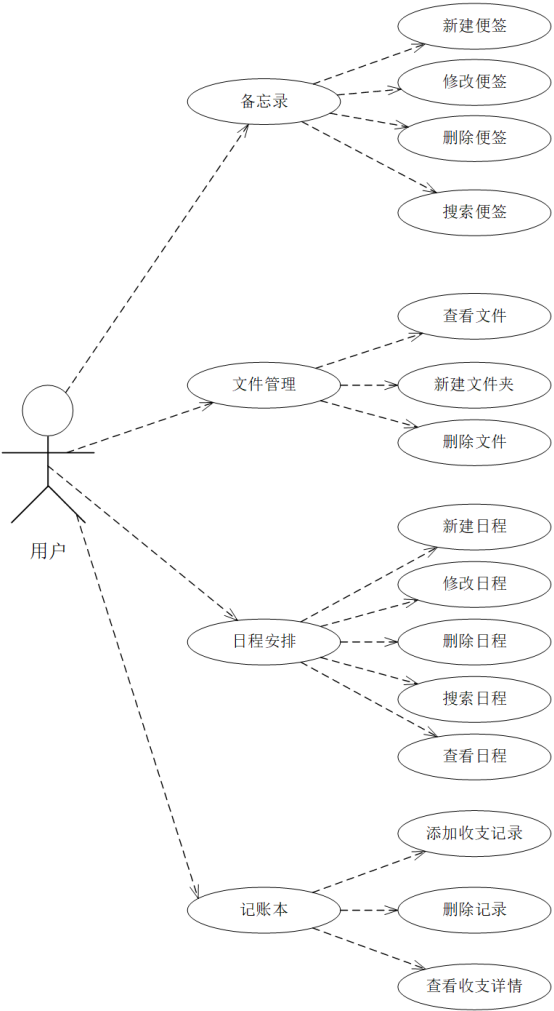


图 2-3 用户用例图

（1）备忘录是人们日常工作很好的辅助工作。备忘录用例主要涉及新建便签、修改便签信息、删除便签、根据关键字搜索便签、根据便签类型搜索便签等功能，具体如表 2-3 所示。

表 2-3 备忘录用例描述

用例名称	备忘录
功能描述	本用例主要用于管理备忘录中便签
输入	便签标题、内容、类型
处理	判断输入框是否为空

输出	数据保存成功
前提条件	无
事件流	<p>基流</p> <p>点击进入备忘录，用例启动</p> <p>（1）系统列出所有的便签</p> <p>（2）该用户提出所要执行的操作</p> <p>如果该用户需要新建普通便签，则执行分支流 S-1:新建普通便签</p> <p>如果该用户需要新建提醒便签，则执行分支流 S-2:新建提醒便签</p> <p>如果该用户需要修改便签，则执行分支流 S-3:修改便签</p> <p>如果该用户需要删除便签，则执行分支流 S-4:删除便签</p> <p>如果该用户需要查找便签，则执行分支流 S-5:查找便签</p> <p>如果该用户需要改变便签样式，则执行分支流 S-6:改变便签样式</p> <p>分支流</p> <p>S-1:新建普通便签</p> <p>点击按钮，进入新建普通便签界面，添加并保存新的便签信息</p> <p>S-2:新建提醒便签</p> <p>点击按钮，进入新建提醒便签界面，设置提醒时间，添加并保存新的提醒便签信息</p> <p>S-3:修改便签</p> <p>点击需要修改的便签，进入便签信息详情界面，修改并保存便签信息</p> <p>S-4:删除便签</p> <p>长按便签，弹出选择删除对话框，选择确定后便签信息被删除并更新数据库</p> <p>S-5:查找便签</p> <p>点击类型下拉列表查找不同类型的便签。点击搜索按钮，输入关键字进行查询。</p> <p>S-6:改变便签样式</p> <p>用户点击改变样式按钮，可将便签展示位列表模式或宫格模式。</p> <p>用户操作完后，用例结束</p>
后置条件	用例成功后，便签信息被添加、修改或删除

（2）文件管理管理用例包含查看、搜索、删除文件和添加新文件夹功能，具体如 2-4 所示：

表 2-4 文件管理用例描述

用例名称	文件管理
功能描述	本用例主要用于查看根目录下的各种文件
输入	无
处理	判断软件是否有查看内存卡权限，有则打开手机内存，无则打开失败
输出	打开目录成功
前提条件	用户登陆成功
事件流	<p>基流</p> <p>用户查看文件时，用例启动</p> <p>（1）系统列出目录下所有文件</p> <p>（2）该用户提出所要执行的操作</p> <p>如果用户需要查看文件夹内容，则执行分支流 S-2:查看文件夹内容</p> <p>如果用户需要删除文件，则执行分支流 S-3:删除文件</p> <p>如果用户需要添加新的文件夹，则执行分支流 S-4:新建文件夹</p> <p>如果用户需要搜索文件，则执行分支流 S-4:搜索文件</p> <p>分支流</p> <p>S-2:查看文件夹内容</p> <p>点击进入文件夹，查看文件夹内容</p> <p>S-3:删除文件</p> <p>长按文件，文件被删除并更新列表</p> <p>S-4:新建文件夹</p> <p>点击新建按钮，新建文件夹</p> <p>S-5:搜索</p> <p>通过关键字搜索文件</p> <p>用户操作完后，用例结束</p>
后置条件	用例成功后，文件被打开、删除或新建

（3）日程安排用例包含添加、修改、删除，搜索安排事件，查看日程安排记录等功能，具体如表 2-5 所示：

表 2-5 日程安排用例描述

用例名称	日程安排
功能描述	本用例主要用于管理日程安排
输入	安排事件标题、内容、时间，是否完成

处理	判断输入框是否为空
输出	数据保存成功
前提条件	无
事件流	<p>基流</p> <p>点击进入日程安排，用例启动</p> <p>（3）系统列出当月日历，点击日期显示当日安排</p> <p>（4）该用户提出所要执行的操作</p> <p>如果该用户需要新建安排，则执行分支流 S-1:新建安排</p> <p>如果该用户需要修改安排，则执行分支流 S-2:修改安排</p> <p>如果该用户需要删除安排，则执行分支流 S-3:删除安排</p> <p>如果该用户需要查找安排，则执行分支流 S-4:查找安排</p> <p>如果该用户需要查看全部安排，则执行分支流 S-5:查看全部安排</p> <p>分支流</p> <p>S-1:新建安排</p> <p>进入新建安排界面，添加并保存新的安排信息，数据库插入新的数据</p> <p>S-2:修改安排</p> <p>进入安排修改界面，修改并保存安排信息，数据库数据被修改</p> <p>S-3:删除安排</p> <p>用户长按想要删除的事件，弹出选择删除对话框，选择确定，安排信息被删除并更新数据库</p> <p>S-4:查找安排</p> <p>用户点击搜索按钮，输入关键字查找相关安排。</p> <p>S-5: 查看全部安排</p> <p>用户点击进入事件记录按钮，可将查看全部安排及安排完成情况。</p> <p>用户操作完后，用例结束</p>
后置条件	用例成功后，安排信息被添加、修改或删除

（4）记账本用例包含新增和删除收支信息，查看收支记录和每月消费趋势等功能，具体如表 2-6 所示：

表 2-6 记账本用例描述

用例名称	记账本
功能描述	本用例主要用于管理记账本
输入	账单类型，金额

处理	判断输入内容
输出	数据保存成功
前提条件	无
事件流	<p>基流</p> <p>用户进入记账本时，用例启动</p> <p>（3）系统列出当日所有账单信息</p> <p>（4）用户提出所要执行的操作</p> <p>如果系统管理员需要新建账单信息，则执行分支流 S-2:新建账单信息</p> <p>如果系统管理员需要删除账单信息，则执行分支流 S-3:删除账单信息</p> <p>如果系统管理员需要隐藏账单信息，则执行分支流 S-4:隐藏账单信息</p> <p>如果系统管理员需要查看账单记录，则执行分支流 S-5:查看账单记录</p> <p>如果系统管理员需要查看账单图表，则执行分支流 S-6:查看账单图表</p> <p>分支流</p> <p>S-2: 新建账单信息</p> <p>进入添加账单界面，输入并保存账单信息</p> <p>S-3:删除账单信息</p> <p>长按账单，弹出选择删除对话框，选择确定账单信息被删除并更新数据库</p> <p>S-4: 隐藏账单信息</p> <p>点击隐藏按钮，隐藏部分账单信息</p> <p>S-5: 查看账单记录</p> <p>进入账单记录界面，查看每月账单</p> <p>S-6: 查看账单图表</p> <p>进入账单图表界面，可查看每月消费和收入趋势，以及不同类型收支百分比。</p> <p>用户操作完后，用例结束</p>
后置条件	用例成功后，账单信息被修改或删除

2.3 环境需求

2.3.1 软件环境要求

运行环境：系统能运行于安卓系统的手机中。

软件配置：Android studio。

2.3.2 开发环境要求

本系统基于 Android 应用程序开发技术，SQLite 数据库以及 JAVA 语言工具。

开发工具：Android studio

测试工具：Android 8.0.0 华为手机

开发语言：Android

数据库：SQLite 数据库

图形化工具：SQLiteSpy

字符编码：UTF-8

3 系统设计

3.1 系统目标

基于 Android 的生活助手 APP 是指一个集合了备忘录、文件管理、记账本和日程安排等帮助规划记录生活功能的综合 APP。

本系统主要是完成各个功能内存储数据的各种调用与交互界面的设计。

3.2 系统功能结构

系统分为五个模块：备忘录、记账本、日程安排、文件管理和个人中心。

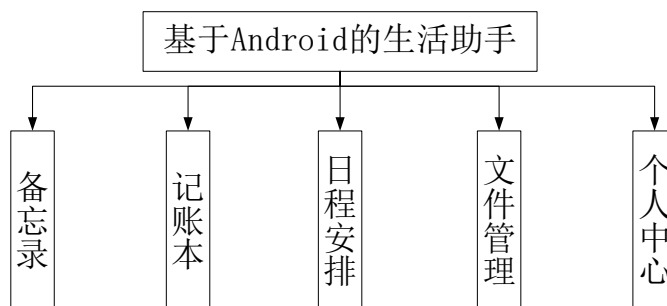


图 3-1 模块层次图

● 备忘录

备忘录模块主要是为用户提供一个记录生活中各种琐碎或重要事情的功能，比如日记，家人朋友的生日和学习笔记等，用户还可以设置定时提醒便签，提醒用户注意事项。该功能还提供关键字查找和类型查找两种搜索方式，让用户能够更方便快捷的寻找相关便签。

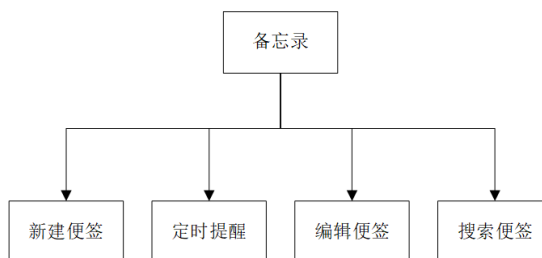


图 3-2 备忘录层次图

● 文件管理

文件管理模块主要是为用户提供手机内存文件展示的功能，如果用户想要查看手机内的某个文件，可以通过这个功能进行查看，也可以通过关键字搜索。

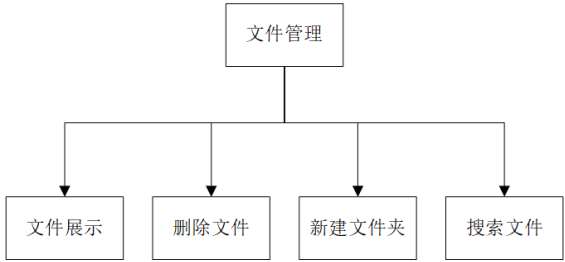


图 3-3 文件管理层次图

● 记账本

记账本模块主要实现的是记录收支条目，用户可在此记录收入和支出，可以清楚的看到每日和每月收支，也可以通过图表观察到每月的收支趋势。

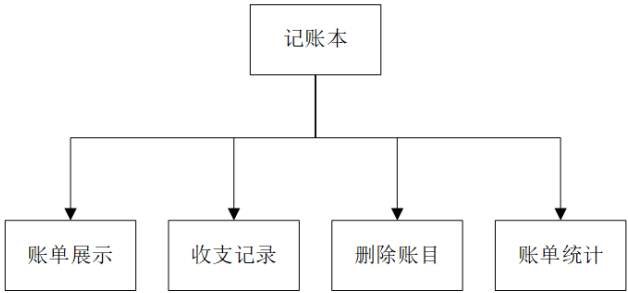


图 3-4 记账本层次图

● 日程安排

日程安排模块主要是对每日行程的规划，用户可以在新建安排时设定时间，通过点击相应日期查看当日安排。

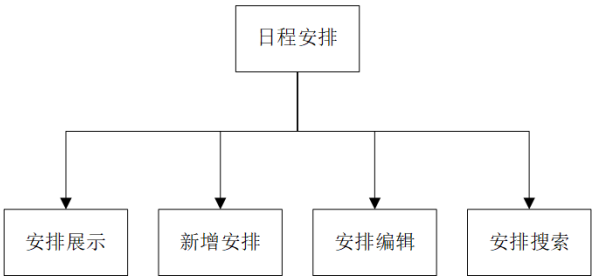


图 3-5 日程安排层次图

● 个人中心

个人中心模块。用户可以删除账号，修改密码，跳转到注册界面。

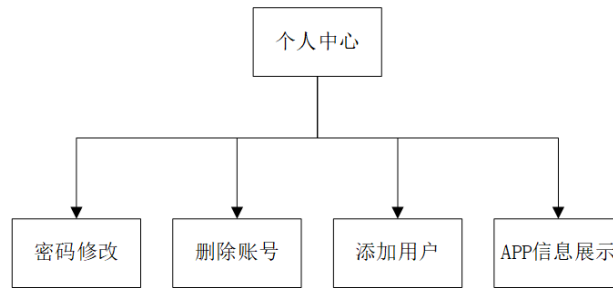


图 3-6 个人中心层次图

根据上述分析，可得出生活助手 APP 的系统功能结构图和系统流程图。

系统功能结构图显示了 5 个模块，包含备忘录、日程管理、文件管理、记账本功能以及用户账号进行操作，具体设计如图 3-7 所示。

系统流程图显示对不同功能的操作流程，首先判断用户是否注册，非注册用户无法进入系统，进入系统后用户可以对系统进行操作具体流程如图 3-8 所示

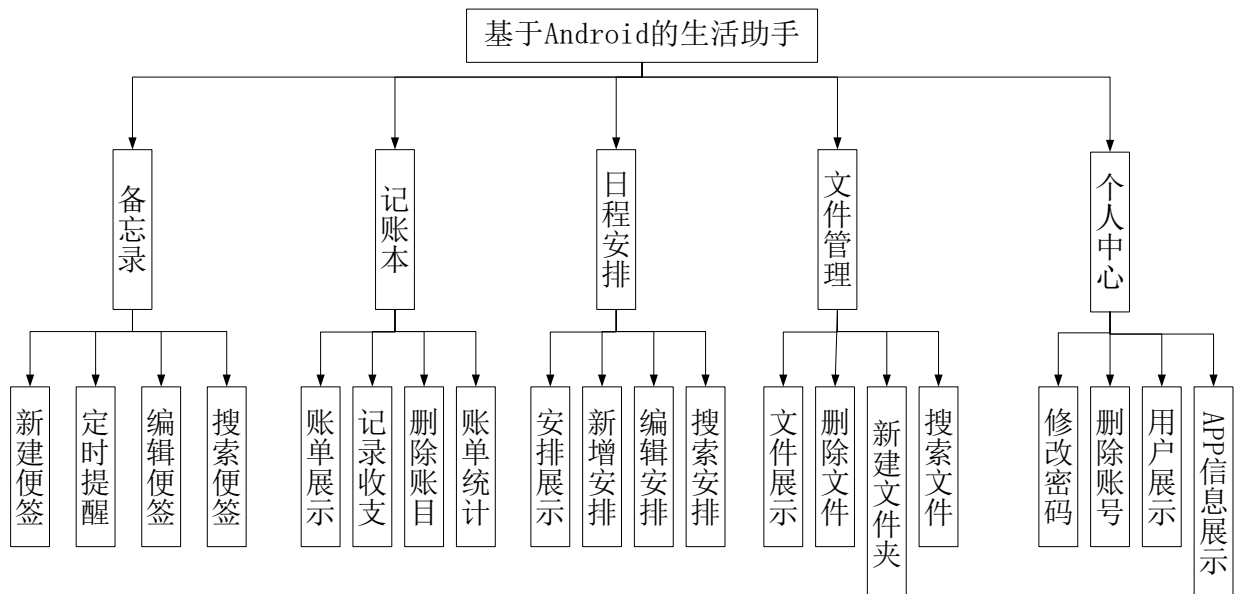


图 3-7 系统功能结构图

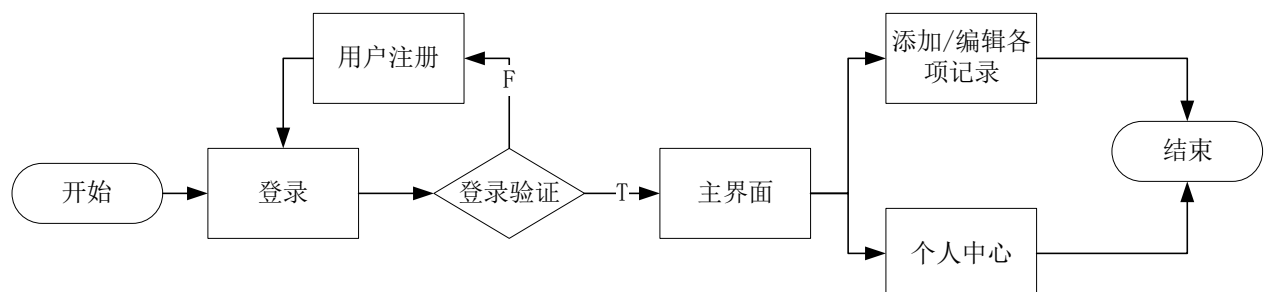


图 3-8 系统流程图

3.3 数据库设计

3.3.1 数据库需求分析

根据分析，实体有用户、记账本收支信息、记账本图表、备忘录信息、日程安排。根据系统的需求，数据库中需要创建以下数据表。

(1) 用户信息表：用于存放用户的用户名和密码。

用户（用户名，密码）

```
create table usertb(name varchar(20),password varchar(20))
```

(2) 记账本信息表：用于存放收支记录的类型、图片和价格等信息。

记账本（记账本 ID，类型，被选中类型图片，备注，价格，保存时间字符串，账单年月日，收支类型（收入---1 支出---0））

```
"create table accounttb(id integer primary key autoincrement,typename
varchar(10),smageld integer,mark varchar(80),money float," +
"time varchar(60),year integer,month integer,day integer,kind
integer)";
```

(3) 记账本类型表：用于存放记账本收支图片。

记账本类型（类型 id，类型名称，未选中图片 id，被选中图片 id，收支类型：）
（收入---1 支出---0））

```
"create table typetb(id integer primary key autoincrement,typename
varchar(10),imageld integer,smageld integer,kind integer)";
```

(4) 备忘录信息表：用于记录用户添加便签的标题、内容和类型等。

备忘录信息（便签 ID,便签标题，便签内容，最后编辑时间分类，便签类型（0--普通标签；1--提醒便签），提醒时间）

```
"create table memotb(id integer primary key autoincrement,title
varchar(20),content varchar(200),now_time varchar(60)," +
"tag integer default 1,date_time varchar(20),kind integer)"
```

(5) 日程安排信息表：用于记录用户每日日程安排信息。

日程安排（日程安排 ID，标题，内容，保存时间，保存年月日，是否完成：
1---完成，0---未完成）

```
"create table scheduletb(id integer primary key autoincrement,title
varchar(20),content varchar(200),"
" time varchar(60),year integer,month integer,day integer,kind
integer default 0);
```

(6) 系统配置信息表：用于粗放系统相关配置信息。

3.3.2 实体关系图（E-R 图）

E-R 图提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。通过需求分析阶段的分析，设计生活助手系统数据库模型，使用 Visio 建立概念模型，在 E-R 图中可以发现了我有四个数据表，分别为用户数据表、备忘录数据表、日程管理数据表和记账本数据表，具体描述如图 3-9。

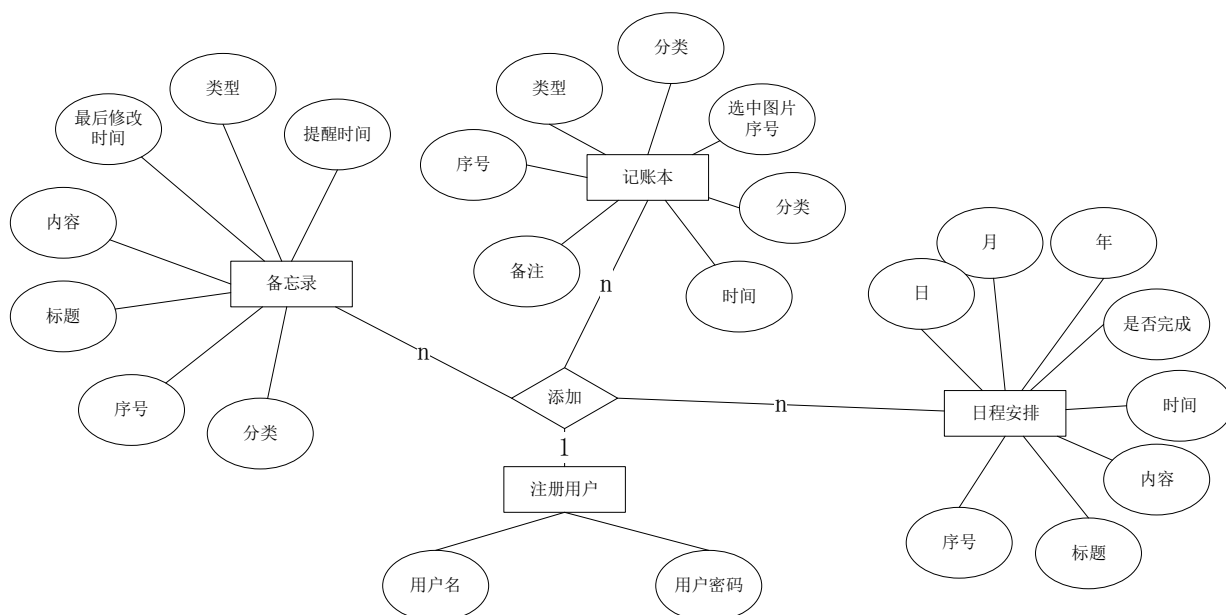


图 3-9 E-R 图

3.3.3 数据表设计

(1) 用户信息表（usertb）

用户信息表主要用于存储用户的信息，表 user 的结构如表所示。

表 3-1 用户注册信息表

字段名	字段类型	长度	主/外键	备注
name	varchar	20		用户名

password	varchar	20		用户密码
----------	---------	----	--	------

(2) 记账本信息表 (accounttb)

记账本信息表主要用于存储日程账目的信息，表 accounttb 的结构如表所示。

表 3-2 记账本信息表

字段名	字段类型	长度	主/外键	备注
id	int	11	PK	序号
typename	varchar	10		类型
slmageId	int	11		被选中类型图片
mark	varchar	80		备注
money	float	32		价格
time	varchar	60		时间字符串
year	int	11		年
month	int	11		月
day	int	11		日
kind	int	11		类型 收入---1 支出---0

(3) 记账本类型表 (news_info)

记账本表主要用于存储记账本类型的图片的信息，表 typetb 的结构如表 3-3 所示。

表 3-3 记账本类型表

字段名	字段类型	长度	主/外键	备注
id	int	11	PK	序号
type_name	varchar	10		类型名称
imageId	int	11		未被选中图片 id
simageId	int	11		被选中图片 id
kind	int	11		类型：收入-1 支出-0

(4) 备忘录信息表 (memotb)

备忘录信息表主要用于存储日程记录的便签信息，表 memotb 的结构如表所示。

表 3-4 备忘录信息表

字段名	字段类型	长度	主/外键	备注
id	int	11	PK	序号
title	varchar	20		标题
content	varchar	200		内容
now_time	varchar	20		便签最后编辑时间
tag	int	11		分类
date_time	varchar	20		便签提醒时间
kind	int	11		便签类型：0--普通标签；1--提醒便签

(5) 日程安排信息表 (scheduletb)

日程安排信息表主要用于存储日常安排信息，表 scheduletb 的结构如表所示。

表 3-5 日程安排信息表

字段名	字段类型	长度	主/外键	备注
id	int	11	PK	序号
title	varchar	20		标题
content	varchar	200		内容
time	varchar	20		保存时间
year	int	11		年
month	int	11		月
day	int	11		日
kind	int	11		是否完成：1---完成，0---未完成

4 详细设计

4.1 注册

用户在首次进入系统时可以通过这个模块申请账号。

4.1.1 界面设计



图 4-1 注册界面设计图

4.1.2 设计思想

用户注册主要是当用户提出注册请求的时候，系统跳转到注册界面提示用户填写相关注册信息，用户填写信息后提交注册，系统检查是否信息是否正确，是则显示注册成功，并将用户信息保存到数据库，否则提示错误信息。

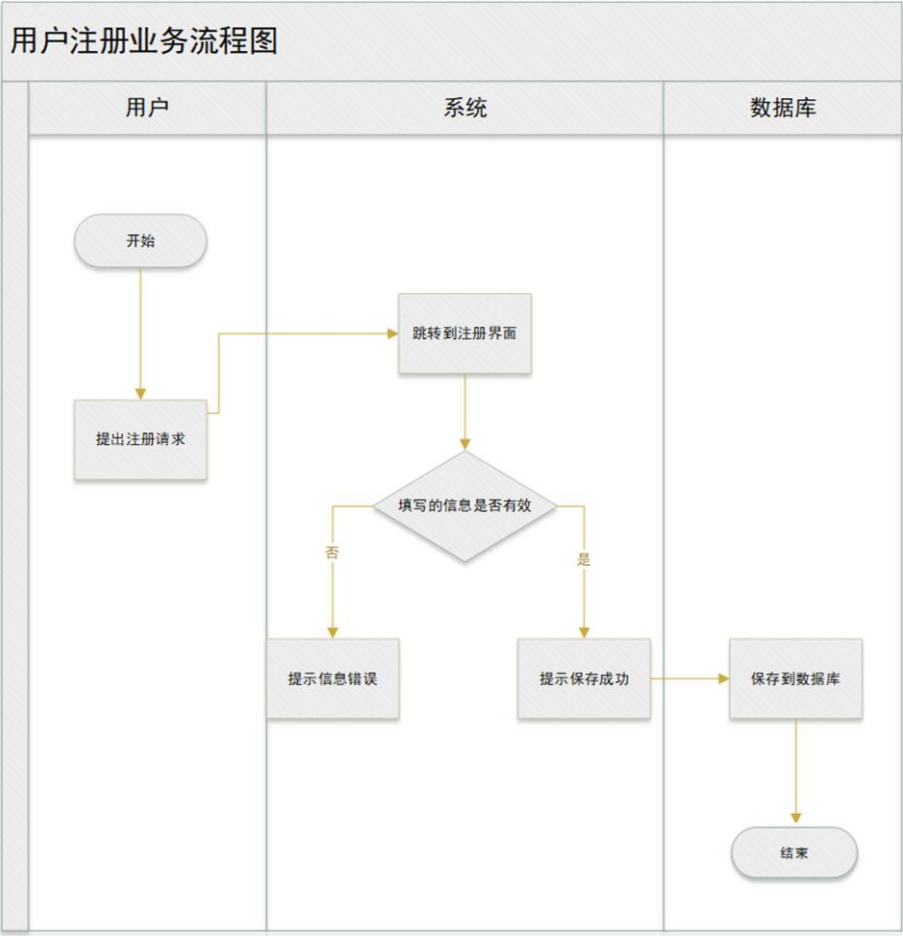


图 4-2 用户注册业务流程图

4.1.3 关键代码

注册功能部分代码：

```
//将用户名和密码加入到数据库中
UserDao.add(username, password);
Intent intent1 = new Intent(this, MainActivity.class);
startActivity(intent1);
finish();
Toast.makeText(this, "验证通过，注册成功",
Toast.LENGTH_SHORT).show();
```

4.2 登录

用户可通过登录模块进入系统或进入注册模块。

4.2.1 界面设计



图 4-3 登录界面设计图

4.2.2 设计思想

用户登录主要是当用户提出登录请求时，系统提示用户填写账号、密码。系统通过比对数据库数据来判断用户输入信息是否正确，正确则进入用户主页面，否则系统提示验证失败，用户重新输入正确信息即可成功登录，用户亦可通过登录模块进入注册模块。

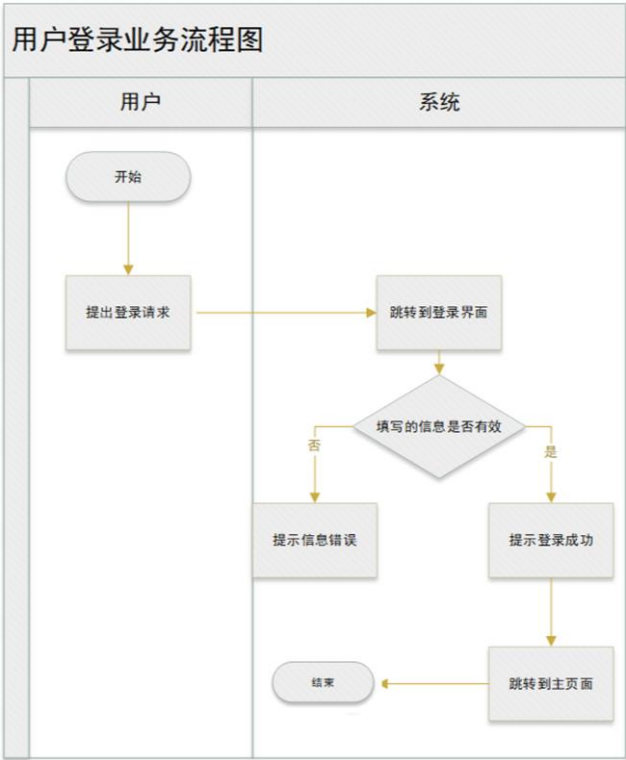


图 4-4 用户登录业务流程图

4.2.3 关键代码

//用户登录功能部分代码：

```
//如果数据库为空
if (UserDao.getCount() == 0 || UserDao.find().getName().isEmpty() || UserDao.find().getPassword().isEmpty())
{
    Toast.makeText(this, "请先注册账号", Toast.LENGTH_SHORT).show();
} else {
    if (!TextUtils.isEmpty(name) || !TextUtils.isEmpty(password)) {
        if (name.equals(UserDao.find().getName()) && password.equals(UserDao.find().getPassword()))
        { //比较
            Toast.makeText(this, "登录成功", Toast.LENGTH_SHORT).show();
            //跳转到主界面
            Intent intent = new Intent(this, MainActivity.class);
            startActivity(intent);
            finish(); //销毁此 Activity
        } else {
            Toast.makeText(this, "用户名或密码不正确，请重新输入", Toast.LENGTH_SHORT).show();
        }
    } else {
        Toast.makeText(this, "请输入你的用户名或密码", Toast.LENGTH_SHORT).show();
    }
}
```

4.3 主界面

本模块是连接各个功能入口的一个功能模块。

4.3.1 界面设计



图 4-5 主界面设计图

4.3.2 设计思想

主页面显示本 APP 各个功能的链接，使各个功能更加有条理有规划。

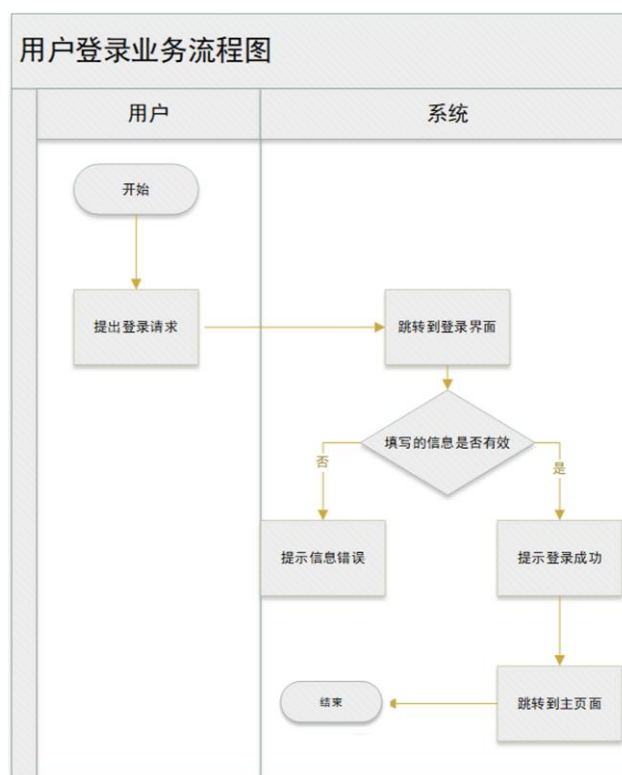


图 4-6 主界面业务流程图

4.3.3 关键代码

//主界面功能部分代码:

```
gview.setOnItemClickListener(new AdapterView.OnItemClickListener() { // 为 GridView 设置项单击事件
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        Intent intent = null; // 创建 Intent 对象
        switch (i) {
            case 0:
                Intent intent1 = new Intent(MainActivity.this, Memo.class);
                startActivity(intent1); // 打开 Memo
                break;
            case 1:
                Intent intent2 = new Intent(MainActivity.this, FileManager.class);
                startActivity(intent2); // 打开 File
                break;
            case 2:
                Intent intent3 = new Intent(MainActivity.this, Schedule.class);
                startActivity(intent3); // 打开 Schedule
                break;
            case 3:
                // ... (code continues)
            
```

```
        Intent intent0 = new Intent(MainActivity.this, Account.class);//
        startActivity(intent0);// 打开 Account
        break;
    case 4:
        Intent intent4 = new Intent(MainActivity.this, User.class);
        startActivity(intent4);
        break;
    case 5:
        Intent intent5 = new Intent(MainActivity.this, Login.class);
        startActivity(intent5);
        finish();
        break;
    }
});
```

4.4 备忘录管理

本模块是提供给用户记录和设置提醒相关事件的一个功能模块。

4.4.1 界面设计



图 4-7 备忘录界面设计图

4.4.2 设计思想

备忘录业务流程是描述当用户向系统提出备忘录管理请求时，系统转到备忘录页面，根据用户提出的请求，执行删除、添加或修改备忘录便签操作，并将结果保存到数据库。

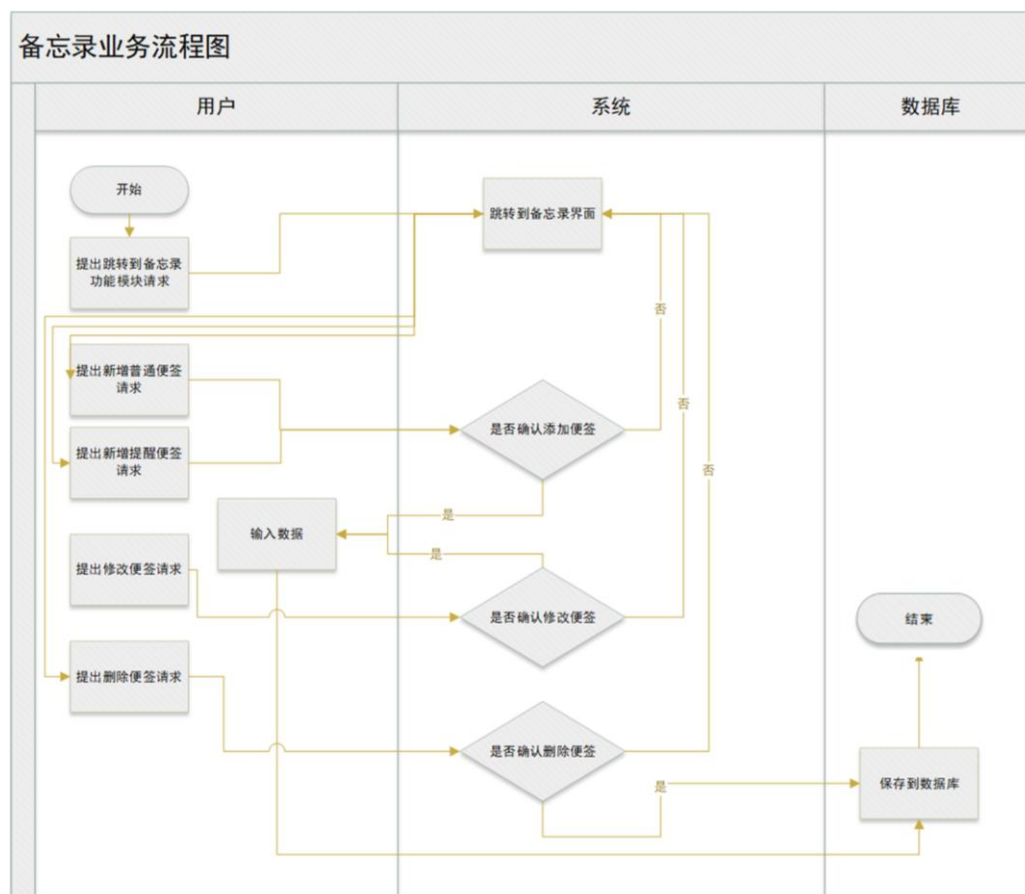


图 4-8 备忘录业务流程图

4.4.3 关键代码

//备忘录部分代码：

```
if (!TextUtils.isEmpty(content) || !TextUtils.isEmpty(title)) {
    if (memoId != null && MemoDao.find(Integer.parseInt(memoId)).getContent() != null) {
        tb_memo.setId(Integer.parseInt(memoId));
        tb_memo.setTitle(title);
        tb_memo.setContent(content);
        tb_memo.setNow_time(now_time);
        tb_memo.setTag(tag);
        tb_memo.setDate_time(now_time);
        MemoDao.updateMemo(tb_memo);
    }
```

```
// 弹出信息提示
Toast.makeText(MemoRecord.this, " 数 据 修 改 成 功 ! ",
Toast.LENGTH_SHORT).show();
}else {
MemoDao.insertItemToMemotb(new Tb_Memo(0,title,content,now_time,tag,now_time,0));
// 弹出信息提示
Toast.makeText(MemoRecord.this, " 数 据 保 存 成 功 ! ",
Toast.LENGTH_SHORT).show();
}
```

4.5 文件管理

本模块提供给用户查看手机内存根目录下的包括 word、PPT 和 Excel 等各种文件。

4.5.1 界面设计

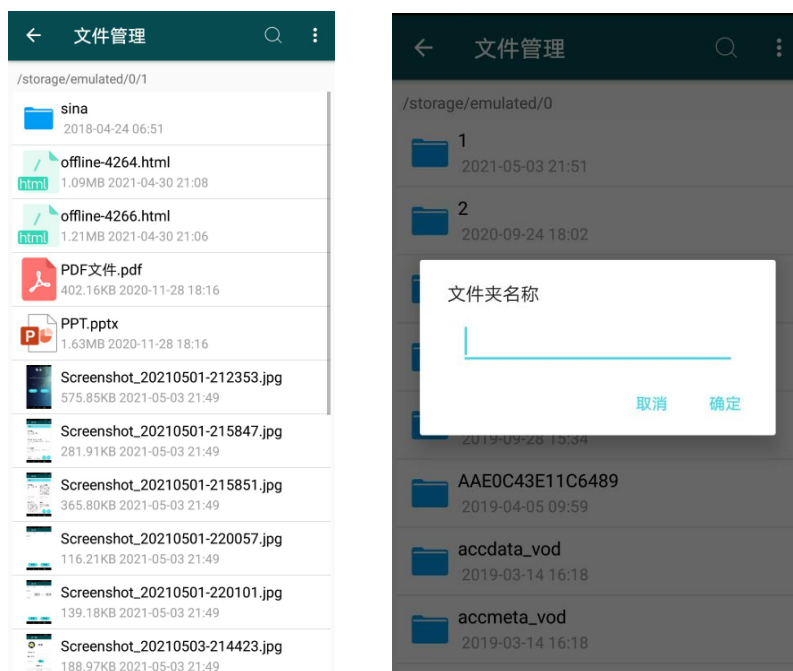


图 4-9 文件管理界面设计图

4.5.2 设计思想

文件管理业务流程是描述当用户向系统提出文件管理请求时，系统转到文件管理页面，根据用户提出的请求，执行删除、搜索或添加文件夹操作。

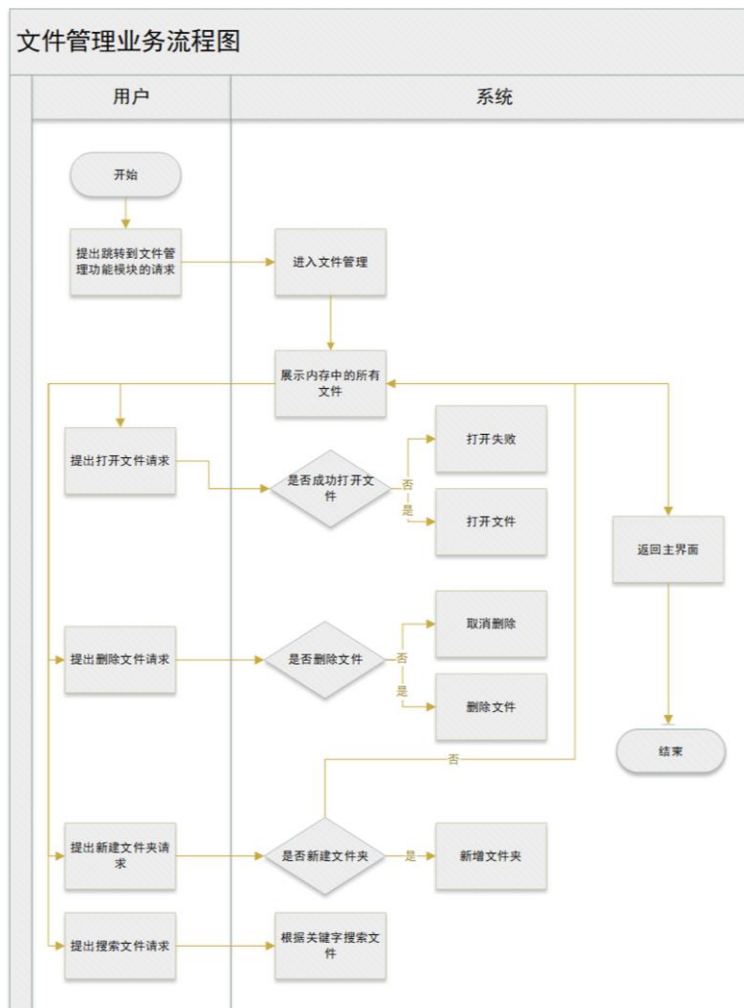


图 4-10 文件管理业务流程图

4.5.3 关键代码

```

if (pfile.exists()) { // 判断该路径是否存在
    files = pfile.listFiles(); // 该文件下的所有文件和文件夹列表
}
if (files != null && files.length > 0) { // 非空验证
    for (File file : files) { // foreach 循环遍历
        FileInfo item = new FileInfo();
        if (file.isHidden()) {
            continue; // 跳过隐藏文件(.xxx)
        }
        if (file.isDirectory() && file.canRead()) { // 是否是文件夹和是否可读
            file.isHidden(); // 是否是隐藏文件
            Log.d("HH", "isDirectory()"); // log 输出测试
            // 获取文件夹目录结构
            item.icon = R.drawable.folder; // 图标
            item.size = ""; // 大小
            item.type = FileManager.T_DIR;
        }
    }
}
  
```

```

else if(file.isFile()){// 文件
    Log.d("HH", "isFile()");//log 输出测试
    String ext = getFileEXT(file.getName());
    // 文件的图标
    item.icon=getDrawableIcon(ext);// 根据扩展名获取图标
    // 文件的大小
    String size = getSize(file.length());
    item.size=size;
    item.type = FileManager.T_FILE;
    if (checkEndsInArray(ext, new String[]{"png","gif","jpg","bmp"})) {
        item.type = FileManager.T_PIC;
    }
    if (checkEndsInArray(ext, new String[]{"mp4","3gp","mpeg","mov","flv"})) {
        // video
        item.type = FileManager.T_VIDEO;
    }
} else{// 其它
    Log.d("HH", "mul_file");//log 输出测试
    item.icon=R.drawable.mul_file;
}
item.name=file.getName();// 名称
item.path=file.getPath();// 路径
// 最后修改时间
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm");
Date date = new Date(file.lastModified());
String time = sdf.format(date);
item.time=time;
list.add(item);
}

```

4.6 日程安排管理

本模块提供给用户添加每日日程安排。

4.6.1 界面设计

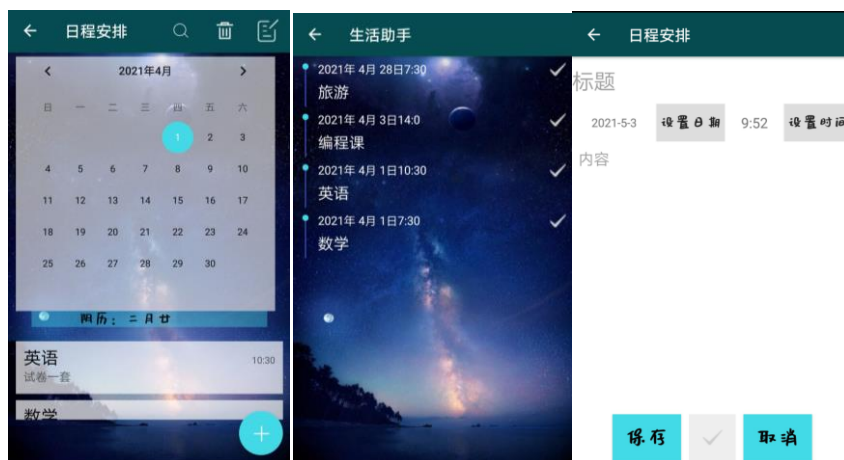


图 4-11 日程安排界面设计图

4.6.2 设计思想

日程安排业务流程是描述当用户向系统提出日程安排管理请求时，系统转到日程安排界面，根据用户提出的请求，执行删除、添加或修改安排操作，并将结果保存到数据库。

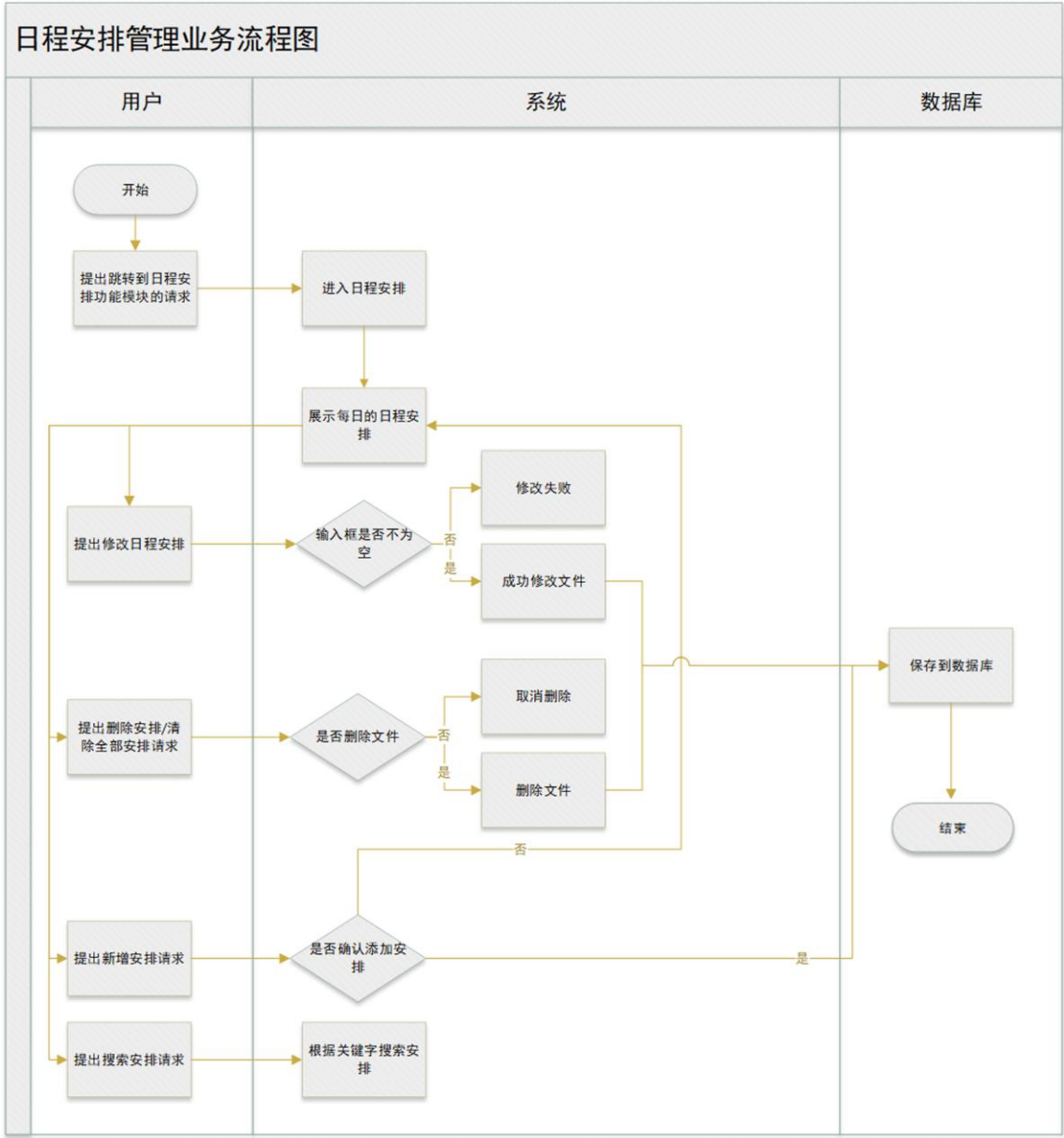


图 4-12 日程安排管理业务流程图

4.6.3 关键代码

//日程安排功能部分代码：

```
//根据点击的日期显示记录
private void queryData(int year,int month ,int day) {
    List<Tb_Schedule> list = ScheduleDao.getOneDayFromScheduletb(year,month,day);
    scheduleData.clear();
    scheduleData.addAll(list);
    adapter.notifyDataSetChanged();
}
```

4.7 记账本管理

本模块是提供给用户记录每日账单的一个功能模块。

4.7.1 界面设计



图 4-13 记账本管理界面设计图

4.7.2 设计思想

记账本管理业务流程是描述当用户向系统提出记账本管理请求时，系统转到记账本管理界面，根据用户提出的请求，执行删除、添加收支记录、隐藏部分信息或查看全部记录时，并将结果保存到数据库。

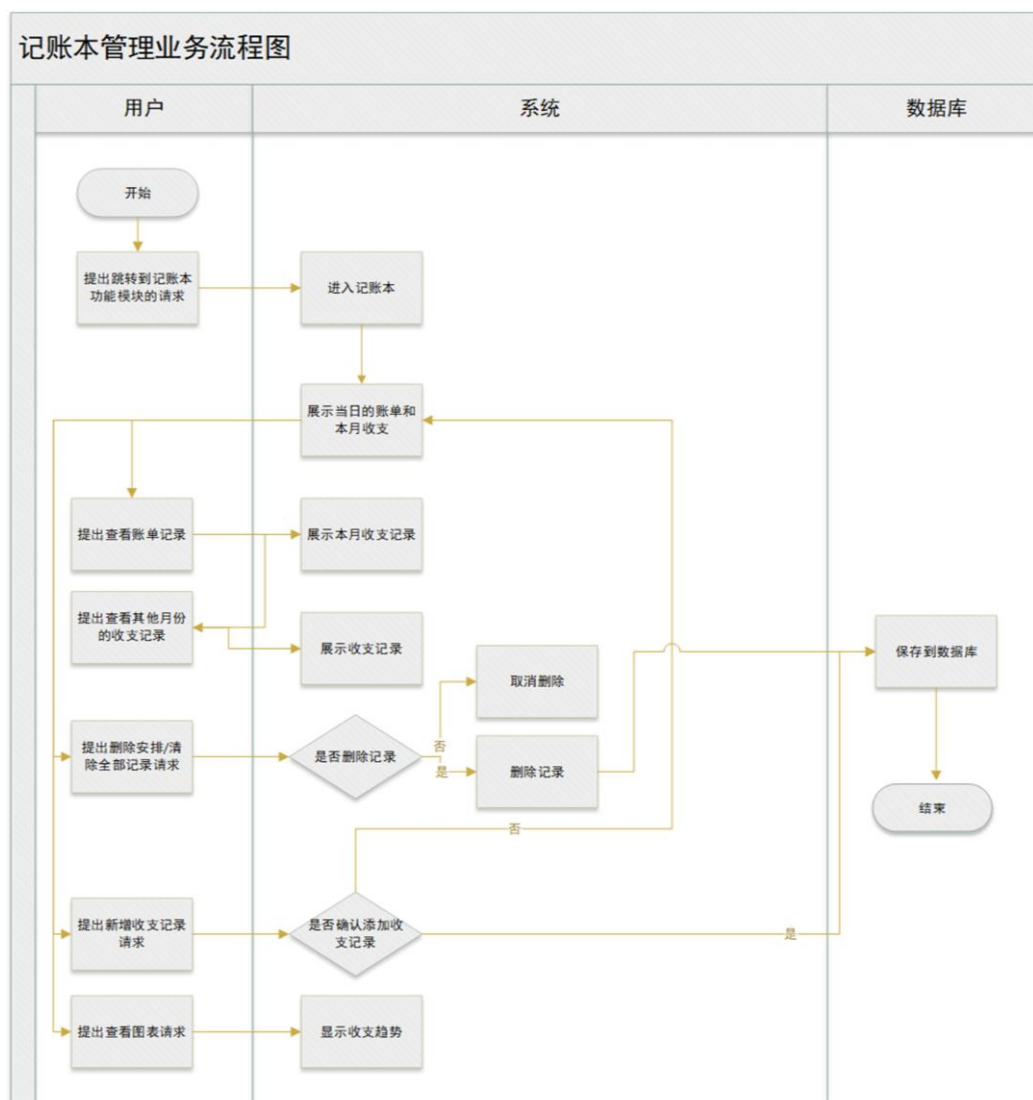


图 4-14 记账本管理业务流程图

4.7.3 关键代码

//记账本功能部分代码：

```

super.loadDataToGV();
//获取数据库当中的数据源
List<Tb_ActType> inlist = AccountDao.getTypeList(1);
typeList.addAll(inlist);
adapter.notifyDataSetChanged();
typeTv.setText("其他");
typeIv.setImageResource(R.mipmap.in_qt_fs);
  
```

4.8 个人中心

本模块主要展示用户和 APP 信息。

4.8.1 界面设计



图 4-15 个人中心界面设计

4.8.2 设计思想

个人中心业务流程是描述当用户向系统提出查看个人中心请求时，系统转到个人中心界面，展示用户和 app 的信息，以及删除账号和修改用户密码的功能模块。

4.8.3 关键代码

```
String password = user_pwd.getText().toString().trim();
switch (view.getId()) {
    case R.id.user_update: //修改按钮
        if(UserDao.getCount() != 0 || !TextUtils.isEmpty(password)){
            UserDao.updata(password);
            Toast.makeText(User.this,"修改成功!",Toast.LENGTH_SHORT).show();
        }else {
            Toast.makeText(User.this,"请输入密码!",Toast.LENGTH_SHORT).show();
        }
        break;
    .....
}
```

5 软件测试

软件测试是一种实际输出与预期输出之间的审核或者比较过程。在测试的过程中，需要尽可能的发现并改正软件的错误，它可以提高软件的可靠性。在本设计中主要采用黑盒测试技术来实现系统的集成测试和系统测试。

5.1 备忘录管理测试

5.1.1 测试用例

表 5-1 备忘录管理测试用例

功能描述	新建、修改、删除和搜索便签、设置便签提醒		
用例目的	用户能成功管理备忘录便签信息		
前提条件	用户登入系统		
输入/动作	输出/响应	实际情况	
进入“备忘录”界面，点击新建便签按钮选择新建便签	成功保存新的便签信息到数据库，且在“备忘录”界面显示新增加的便签	数据插入执行成功，界面刷新并显示新的便签信息	
进入“备忘录”界面，点击想要修改的便签。	成功保存修改后的便签信息到数据库，且在“备忘录”界面显示修改后的便签。	数据修改成功，数据库更新，界面刷新并显示新的便签信息	
进入“备忘录”界面，长按便签，点击清空按钮。	弹出删除窗口，选择删除后成功执行删除数据库操作，且在“备忘录”界面减少被删的便签信息	数据库更新成功，界面减少被删除的便签信息	
进入“备忘录”界面，点击搜索按钮或分类下拉列表	根据关键字或便签类型查找相关便签	在“备忘录”界面显示相关便签	
设置提醒便签	在设置好的时间进行提醒	数据插入执行成功，界面刷新并显示新的便签信息	

5.1.2 界面测试

● 新建便签

- (1) 进入备忘录界面，点击添加按钮，编辑便签并保存如图 5-1。
- (2) 数据插入执行成功，界面显示新纪录如图 5-2。



图5-1 新建便签功能测试截图



图5-2 新建便签功能测试截图

测试结果：在备忘录界面显示新记录，便签保存成功，测试通过。

● 修改便签

- (1) 修改后便签的内容，如图 5-3。
- (2) 点击保存后，便签内容被成功更改，如图 5-4。



图5-3 修改便签功能测试截图



图5-4 修改便签功能测试截图

测试结果：在备忘录界面显示新内容，修改保存成功，测试通过。

● 删除便签

- (1) 删除和清空便签，如图 5-5。
- (2) 点击确定后，便签被删除，在备忘录界面显示新内容，如图 5-6。



图5-5 删除便签功能测试截图

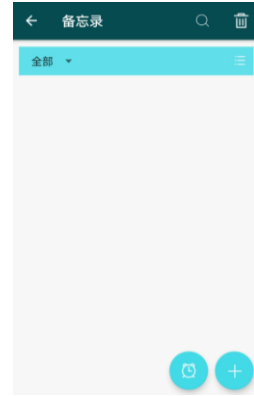


图5-6 删除便签功能测试截图

测试结果：在备忘录界面显示新内容，删除成功，测试通过。

● 搜索便签

(1) 根据关键字或便签分类搜索相关的便签，如图 5-7 所示。



图5-7 搜索便签功能测试截图

测试结果：在备忘录界面显示搜索内容，测试通过。

● 便签提醒

(1) 设置提醒便签，如图 5-8。

(2) 点击保存后，在备忘录界面显示新内容，并在标题栏进行提醒，如图 5-9。



图5-8 设置提醒功能测试截图

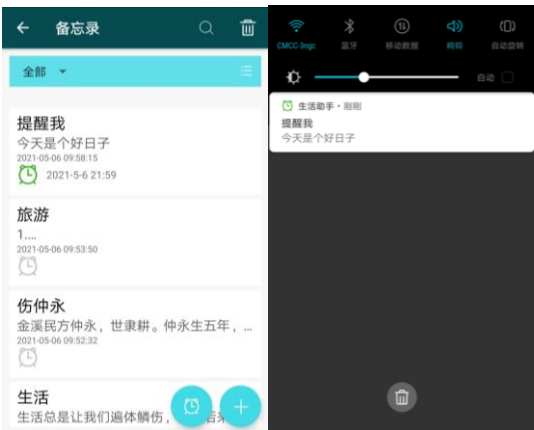


图5-9 设置提醒功能测试截图

测试结果：成功在设置时间发出通知栏提醒，提醒便签保存成功，测试通过。

5.2 文件管理测试

5.2.1 测试用例

表 5-2 文件管理测试用例

功能描述	文件管理	
用例目的	用户能查看内存中的文件、删除文件、搜索文件、新建文件夹	
前提条件	用户登入系统	
输入/动作	输出/响应	实际情况
删除文件	长按文件对文件进行删除操作	删除成功，刷新界面，减少被删除的便签信息
搜索文件	根据关键字查找便签	在界面显示相关文件
新建文件夹	点击新建文件夹选项	在界面显示新增加的文件夹

5.2.2 界面测试

● 删除文件

（1）当需要删除手机中的文件时，可以长按对应文件，弹出删除窗口对文件进行操作，如图5-10所示。

（2）删除成功后数据库更新，界面刷新，同时在界面减少对应的删除的文件，如图5-11。



图5-10 删除文件功能测试截图



图5-11 删除文件功能测试截图

测试结果：成功删除对应的文件，测试通过。

● 搜索文件

(1) 根据关键字或便签分类搜索便签。

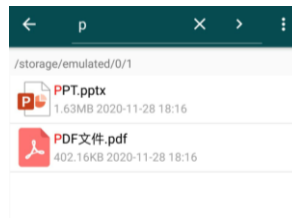


图5-12 搜索文件测试截图

测试结果：在界面显示搜索内容，测试通过。

● 新建文件夹

(1) 点击新建文件夹选项。



图5-13 新建文件夹功能测试截图

测试结果：在界面显示新建内容。

5.3 日程安排测试

5.3.1 测试用例

表 5-3 日程安排测试用例

功能描述	新建、修改、删除和搜索日程，以及查看日程记录和日程完成情况		
用例目的	用户能成功管理日程安排信息		
前提条件	用户登入系统		
输入/动作		输出/响应	实际情况
进入“日程安排”界面，点击新建按钮选择创建新的日程安排。		成功在数据库中插入新的日程信息，且在“日程安排”界面显示新增加的日程。	数据插入执行成功，界面刷新并显示新的便签信息。
进入“日程安排”界面，点击想要修改的日程安排。		成功保存修改后的日程信息到数据库，且在界面显示修改后的日程。	数据更新成功，界面刷新并显示新的日程信息。
进入“日程安排”界面，长按想要删除的日程或点击清空按钮。		成功执行删除数据库操作，且在界面减少被删的日程信息。	数据库更新成功，界面减少被删除的日程信息。
进入“日程安排”界面，点击搜索按钮。		根据关键字查找日程。	在“日程安排”界面显示相关日程。

5.3.2 界面测试

- 新建日程

（1）进入“日程安排”界面，点击按钮创建新的日程安排，编辑日程并保存，如图 5-14。

（2）数据插入执行成功，界面显示新日程，如图 5-15。



图5-14 新建日程功能测试截图



图5-15 新建日程功能测试截图

测试结果：在界面显示新记录，日程保存成功，测试通过。

● 修改日程

(1) 修改日程信息，如图 5-16。

(2) 点击保存后，日程内容被成功更改，如图 5-17。



图5-16 修改日程功能测试截图



图5-17 修改日程功能测试截图

测试结果：在界面显示新内容，修改保存成功，测试通过。

● 删除日程

(1) 删除和清空日程，如图 5-18。

(2) 点击确定后，日程被删除，在界面显示减少后的内容，如图 5-19。

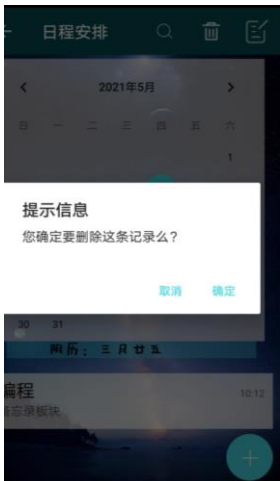


图5-18 删除日程功能测试截图



图5-19 删除日程功能测试截图

测试结果：在界面显示刷新后内容，删除成功，测试通过。

● 搜索日程

(1) 根据关键字搜索日程



图5-20 搜索日程功能测试截图

测试结果：在界面显示搜索内容，测试通过。

5.4 记账本测试

5.4.1 测试用例

表 5-4 记账本测试用例

功能描述	新建和删除收支记录、查看收支趋势和每月收支记录		
用例目的	记录日程收支，并显示收支趋势，使用户更好的规划消费		
前提条件	用户登入系统		
输入/动作	输出/响应	实际情况	

进入“记账本”界面，点击新建按钮创建新的收支记录消息	成功插入新的收支信息到数据库，且在“记账本”界面显示新增加的记录	数据插入执行成功，数据库更新，并显示新的记录在界面
进入“记账本”界面，长按想要删除的记录。	成功执行删除数据库操作，且在界面减少被删的记录。	数据库更新成功，界面减少被删除的收支信息
进入“记账本”界面，点击图表按钮	进入图表界面，查看每月消费趋势	图表界面显示相关内容

5.4.2 界面测试

● 新建收支记录

- (1) 进入“记账本”界面，点击新建按钮选择新建收支记录，如图 5-21。
- (2) 数据插入执行成功，界面显示新记录，如图 5-22 所示。



图5-21 新建收支记录功能测试截图



图5-22 新建收支记录功能测试截图

测试结果：在界面显示新记录，保存成功，测试通过。

● 删除收支记录

- (1) 长按记录对记录进行删除操作，如图 5-23。
- (2) 点击确定后，记录被删除，刷新数据库。同时在界面减少删除的内容，如图 5-24。



图 5-23 删除收支记录功能测试截图



图 5-24 删除收支记录日程功能测试截图

测试结果：在界面显示刷新后内容，删除成功，测试通过。

● 查看收支趋势

(1) 点击图表按钮

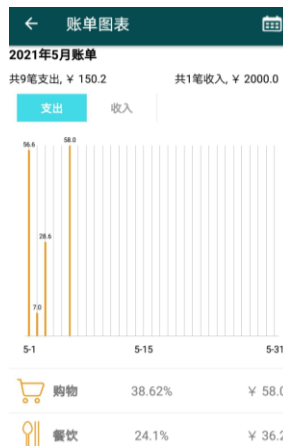


图5-25 查看收支趋势功能测试截图

测试结果：在界面显示相关月份图表信息和不同类型收支百分比，测试通过。

5.5 个人中心测试

5.5.1 测试用例

表 5-5 个人中心测试用例

功能描述	查看用户个人信息、修改用户密码、删除账号		
用例目的	用户对自己账号进行管理		
前提条件	用户登入系统		
输入/动作	输出/响应	实际情况	

输入新的用户密码，点击确认修改	用户密码改变	数据库更新成功，用户密码被改变
点击删除账号按钮	删除所有数据	数据库更新成功

5.5.2 界面测试

(1) 打开个人中心界面



图5-26 个人中心功能测试截图

测试结果：成功修改用户密码和删除账号，测试通过。

5.6 测试结论

共对5个功能模块，16个用例进行了测试。

通过以上测试，预期功能已基本实现，且界面跳转无问题，完成了需求分析要求中的各个功能。

6 结论

为本次设计实现了基于 Android 的生活助手 APP，该设计使用了 SQLite 数据库技术、Java 语言工具，这个 APP 具有以下功能：

1. 账户管理

管理用户的账户信息。接受用户账户信息的提交。

2. 记账功能

记账功能主要由新增收支、收支统计和统计图表组成。技术实现中主要使用了 Fragment 方法将新增收支模块化，使其可同时存在在一个 UI 界面，MPAndroidChart 统计图表技术。满足了用户日常记账的需求，还能使用户能更直观地观察自己的消费趋势及情况。

3. 备忘录功能

备忘录分为普通便签和提醒便签，运用了 AlarmManager 和 BroadcastReceiver 完成通知栏信息的定时发送，通过 Filter ()异步过滤器来完成关键字查找功能。该功能使用户脱离了传统的纸质记录的方法，手机记录使用户无论是在生活、学习还是工作上都能够随时随地的记录和查找，其中添加的提醒功能更是为用户日常生活有了一个更好的保障。

4. 日程管理

安排每日任务，通过 CalendarView OnDateChangeListener 点击日历某个时间查看当日的安排。对每个完成任务进行打卡、查找与统计。日期划分使用户的安排更加明了，通过日历表使用户能够更方便的记录和查找。

5. 文件查看

利用 WRITE_EXTERNAL_STORAGE 获取 sd 卡写的权限，可查看手机内中根目录下的 word、txt、excel 等文件和文件夹，并对其进行编辑。

该 APP 操作简单,界面直观明了，为用户节省了大量的学习时间，且可以满足大部分人的基础需求。