

河北工业大学城市学院

# 毕业设计说明书

作者：李宪鹏 学号：177765

系：计算机科学与软件工程系

专业：软件工程

题目：基于 Android 的物品管理 App 的  
设计与实现

指导者：

评阅者：

2021 年 5 月 17 日

# 毕业设计（论文）中文摘要

## 基于 Android 的物品管理 App 的设计与实现

### 摘要：

随着城市的发展与繁荣，越来越多的青年人离开家乡，选择在大都市进行工作和生活，使得青年人对物品管理有较大需求。本系统选择使用 Android 平台，文中介绍了开发的背景情况和选题的意义，从各个方面详细说明了选择 Android 系统的独特优势。

本文论述了物品管理 App 的设计构思与实现过程，对系统进行了可行性分析、需求分析，对于整个物品管理 App 的数据库中的表设计和功能模块做了细致的分析，详细地说明各个功能模块的基本逻辑和功能。该系统实现了管理住所、房间、家具和物品等信息、在物品处于过期和临期状态时进行提示、将物品打包到容器、导出到 Excel 表格、应用锁定等功能。

**关键词：** Android App 物品管理

# 毕业设计（论文）外文摘要

**Title**     Design and Implementation of Android Based Goods  
Management App

## **Abstract**

With the development and prosperity of cities, more and more young people leave their hometowns and choose to work and live in metropolises, which makes young people have a greater demand for goods management. This system chooses to use the Android platform. The article introduces the background of the development and the meaning of the topic, and explains in detail the unique advantages of choosing the Android system from all aspects. This article discusses the design concept and implementation process of the goods management App, carries out a feasibility analysis and demand analysis of the system, makes a detailed analysis of the table design and functional modules in the database of the entire goods management App, and explains each function module in detail. The basic logic and functions. The system implements functions such as managing information about residence, room, furniture, and goods, prompting when goods are in an expired or impending state, packing goods into containers, exporting to Excel tables, and application locking.

**Keywords:**    Android   App   Goods management

## 目 次

1 引言	1
1.1 Microsoft Visio Professional 简介	1
1.2 Sketch 简介	1
1.3 Android 简介	2
1.4 Java 简介	2
1.5 Android Studio 简介	2
1.6 SQLite 简介	2
2 可行性研究	2
2.1 经济可行性	3
2.2 技术可行性	3
3 需求分析	3
3.1 总体目标	3
3.2 具体目标	3
3.3 系统数据模型	4
4 概要设计	8
4.1 体系结构设计	8
4.2 接口设计	9
4.3 代码设计	10
4.4 数据库设计	10
4.5 运行设计	13
4.6 安全保密设计	13
4.7 维护设计	14
5 详细设计	14
5.1 详细设计概述	14
5.2 界面设计	14
5.3 用户模块定义	14
5.4 概览模块定义	16
5.5 空间模块定义	17
5.6 打包模块定义	18
5.7 设置模块定义	19
5.8 适配器实现	22

5.9 服务实现.....	25
5.10 界面实现.....	26
6 测试.....	29
6.1 测试原则及测试方法概述.....	29
6.2 测试项目测试用例.....	30
6.3 软件测试结论.....	33
结论.....	35
参考文献.....	36
致谢.....	37

## 1 引言

随着社会的发展和人们对技术认知的不断提高，移动设备的普及程度非常高，随着时间的推移，其性能也每年都有大幅提升，还涌现出许多其专用的操作系统。经过市场的发展和选择，目前主流的移动设备上的操作系统是 Android 和 iOS<sup>[1]</sup>。我国经济发展迅速，人民收入逐步增加，消费也不断提高，物品浪费的现象时有发生。而且城市发展迅速，越来越多的大学生毕业后选择在大城市进行工作和生活，他们中的绝大多数都会选择租赁房屋，这就避免不了经常搬家的问题。如何利用手机的便携性来帮助用户管理物品，并减少物品的浪费，成了一个问题。考虑 iOS、Android 的用户数量，本系统选择在用户数量更多的 Android 系统上运行。本文将探讨该基于 Android 的物品管理 App 如何实现，使得用户可以随时随地地管理物品、在物品过期和临期时给予提示。

本文研究的目的在于掌握开发 Android 应用的技术。在开发的过程中，主要通过 Google 公司的 Android SDK，在 Android Studio 中完成该款 Android 物品管理 App 的设计。通过开发一款 Android 的物品管理 App，解决目前个人物品管理 App 所存在的弊端。与此同时，在开发的过程中，进一步强化和巩固自己所学的 Android 理论，掌握和理解 Android 各个组件的使用，掌握项目的开发流程，积累自己的 Android 项目经验，熟练 Android 各类接口的使用，提高自己解决问题的本领，了解有关 Android 开发的新知识。

### 1.1 Microsoft Visio Professional 简介

Visio 是一款专业的绘制图表的办公软件，可以创建流程图、E-R 图、数据流图等多种图表<sup>[2]</sup>。Visio 中附带多种模板，绘制图表简便快捷<sup>[3]</sup>。

### 1.2 Sketch 简介

Sketch 是由荷兰公司 Sketch BV 开发 macOS 专用的矢量图形编辑器，主要用于网站、移动应用程序的界面和交互设计<sup>[4]</sup>。

Sketch 软件的特点是容易理解，入门门槛很低。对于绝大多数的数字产品设计，

Sketch 都能替代，如 Adobe Photoshop 等<sup>[5]</sup>。

### 1.3 Android 简介

Android 开发四大组件分别是：用于表现功能的 Activity；运行没有界面的后台服务的 Service；用于接收 Broadcast 的 Broadcast Receiver；支持数据跨应用的存储和读取、可以和数据库结合起来的 Content Provider。

Android 具有开放、硬件丰富、开发方便等平台优势。

### 1.4 Java 简介

Java 是一种应用面很广的计算机编程语言，特点是跨平台、面向对象等等。

Java 编程语言的风格特别接近 C++。Java 平台由 Java 虚拟机（Java Virtual Machine）和 Java 应用编程接口构成<sup>[6]</sup>。目前常用的 Java 平台基于 Java 1.8，最新版本为 Java 16。

### 1.5 Android Studio 简介

Android Studio 是一个集成开发环境，主要是用于开发 Android 平台程序的，是 Google 官方推荐的工具<sup>[7]</sup>。Android Studio 基于 JetBrains IntelliJ IDEA，为 Android 开发特殊定制，可以运行在 Windows、macOS 和 Linux 平台上<sup>[8]</sup>。Android Studio 的构建工具是 Gradle，不仅提供 Android 开发工具，还提供了 Android Virtual Device 用于开发和调试，使用者不需要拥有 Android 手机，只要新建 Android Virtual Device 就能进行 App 的预览。

### 1.6 SQLite 简介

SQLite 不仅支持标准的 SQL 语法，还遵循了数据库的 ACID 事物<sup>[9]</sup>。它使用的是动态的、弱类型的 SQL 语法<sup>[10]</sup>。它在运行时所需要的资源特别低，性能也不错，而且支持多种主流的操作系统，所以被很多设备所搭载。

## 2 可行性研究

在开发工作进行之前，首先分析本应用的经济可行性和技术可行性。为后续的工作，如需求分析等等打下基础。

## 2.1 经济可行性

目前的主流手机搭载的系统，主要有 Android、iOS 两种。对比二者可以发现，Android 比较自由灵活，而 iOS 则要封闭许多。考虑到搭载二者设备的规模，Android 的装机量比 iOS 的装机量多至少二倍。从设备成本的考虑，Android 设备的软硬件平台都比 iOS 廉价许多。站在经济角度，无论从实现的难度还是成本来看，本应用使用 Android 平台具备很大优势，因此经济上是可行的。

## 2.2 技术可行性

考虑到 Java 和 Kotlin 的流行程度，本系统选择使用 Java 进行编写，使用 Android Studio 作为 IDE。使用 Android 虚拟机（Pixel 4 XL）和 Android 智能手机（MI 8）进行调试和测试。Java 具有跨平台的特性，是一种多平台通用的语言，虽然各个平台的操作系统不一样，但对 Java 程序来说，实现功能的原理基本相同。本系统的开发使用到了 SQLite 数据库和 Android 四大组件的技术。这两项技术在当前十分稳定，并且在 Android 领域使用较为广泛，因此在技术上是可行的。

综上所述，该系统是可行的。

# 3 需求分析

## 3.1 总体目标

本文设计的程序目的是为用户提供一个管理生活中个人物品的平台，通过对过期和临期物品的提醒，来避免因物品过期而导致的浪费，为用户提供生活的便利。这款 App 的特点有信息实时性、适用面广、操作简单便捷、易维护等。该软件为同类 App 提供了一些新的思路 and 想法，不仅可以记录物品的基本信息，在物品即将过期和过期的时候进行提醒，还可以将物品进行打包。本款 App 经过多次修改和验证，程序在功能、界面美观程度和性能上都能满足用户基本的需求。

## 3.2 具体目标

根据需求分析的结果，该个人物品管理 App 的功能主要分为用户类、管理类和设置类，具体功能如下。

- （1）提供登录系统、注册账号、注销账号的功能；



(2) 提供切换住所、查看过期和临期物品列表、查询和修改当前住所账号、创建物品、搜索物品的功能；



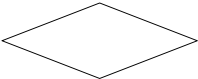

(3) 提供房间-家具-物品的层级显示、容器-物品的层级显示、修改和删除物品的功能；

(4) 提供修改用户头像、修改物品被列为临期状态的条件、住所管理、房间管理、家具管理、容器管理、类别管理、设置应用密码、输入密码超过 5 次锁定应用、将所有物品详情存入 Excel 文件、对应用进行反馈和评分的功能。

### 3.3 系统数据模型

E-R 方法的全称是 Entity-Relationship Approach, 中文翻译是“实体-联系方法”<sup>[11]</sup>。它是一种描述真实生活中的结构模型的方法, 是表示概念模型的一种方式<sup>[12]</sup>。通过对本 App 进行需求分析, 本系统中使用的数据库所需的实体分别为用户、住所、账号(住所中的水电气等账号)、房间、家具、容器、种类、物品。

表 3-1 E-R 图符号约定表

符号含义	符号	说明
实体		表示实体集
属性		表示属性
联系		表示联系集
线段		将实体连接到属性或联系集

用户属性包含用户 ID、用户名、密码、性别、头像、提醒时间, 其 E-R 图见图 3-1。

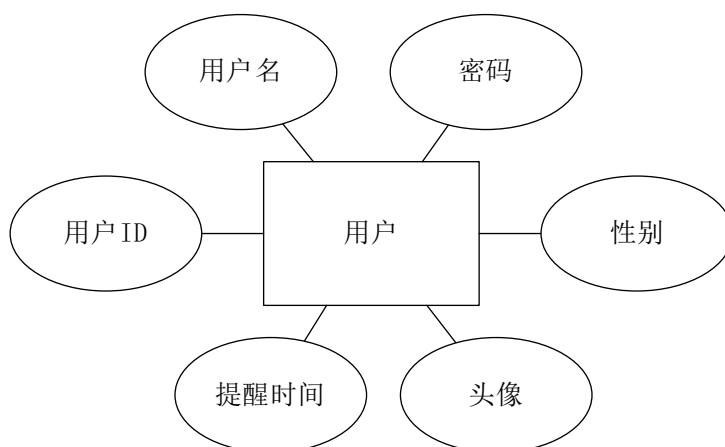


图 3-1 用户实体 E-R 图

住所属性包含住所 ID、住所名、用户 ID，其 E-R 图见图 3-2。

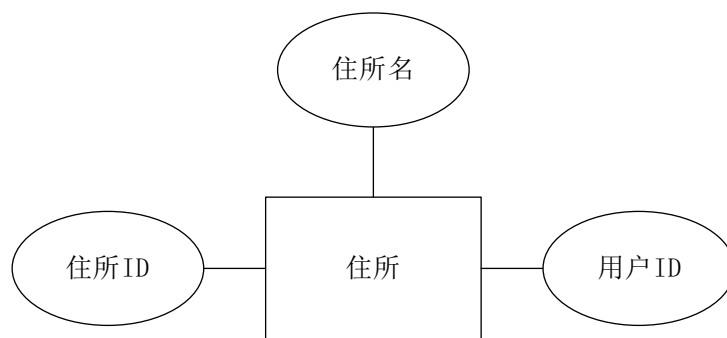


图 3-2 住所实体 E-R 图

账号指的是水电气的账号，其属性包含账号 ID、账号名、账号值、用户 ID，其 E-R 图见图 3-3。

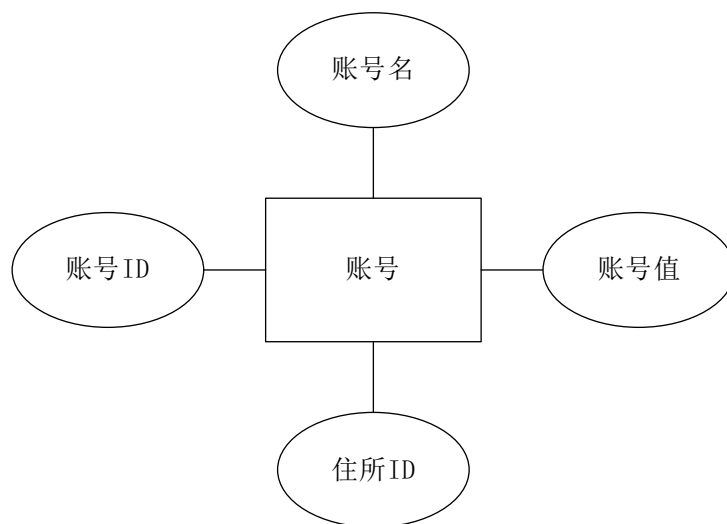


图 3-3 账号实体 E-R 图

房间属性包含房间 ID、房间名、用户 ID、住所 ID，其 E-R 图见图 3-4。

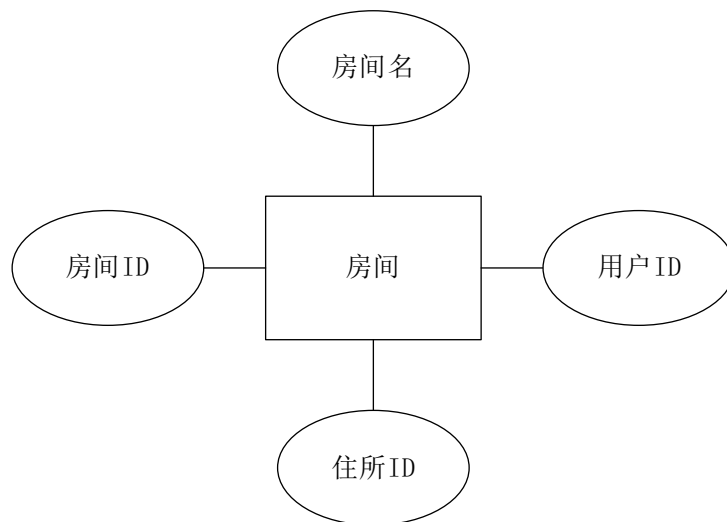


图 3-4 房间实体 E-R 图

家具属性包含家具 ID、家具名、用户 ID、住所 ID、房间 ID，其 E-R 图见图 3-5。

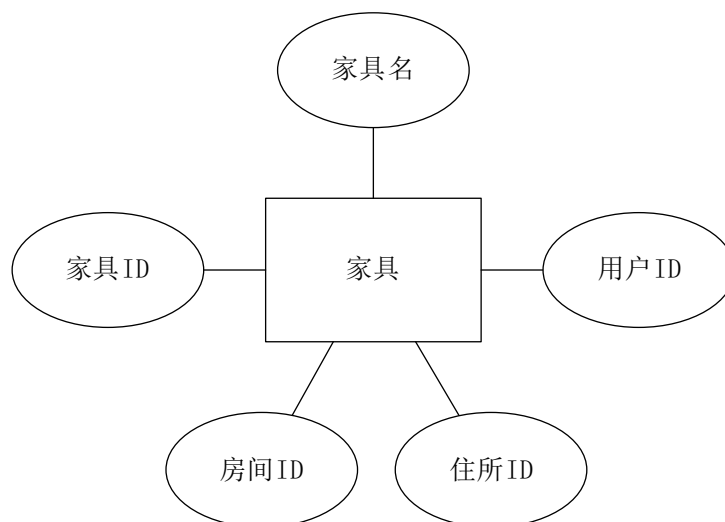


图 3-5 家具实体 E-R 图

容器属性包含容器 ID、容器名、用户 ID，其 E-R 图见图 3-6。

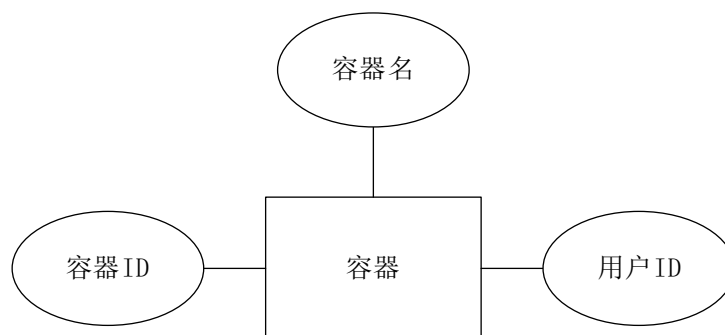


图 3-6 容器实体 E-R 图

种类属性包含种类 ID、种类名、用户 ID，其 E-R 图见图 3-7。

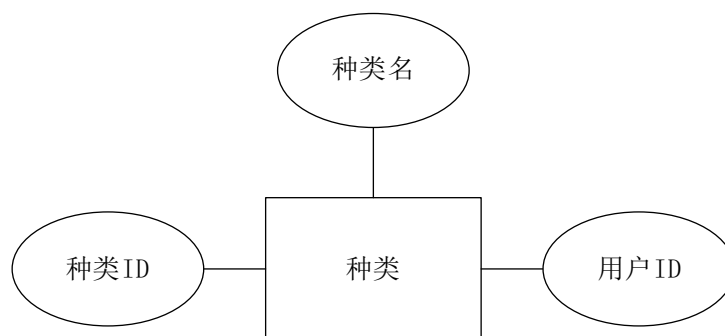


图 3-7 种类实体 E-R 图

物品属性包含物品 ID、物品名、物品数量、物品照片 1、物品照片 2、物品照片 3、物品购买日期、物品生产日期、物品保质期、物品保质期种类、物品过期天数、物品是否过期、物品是否临期、物品是否打包、物品备注、用户 ID、住所 ID、房间 ID、家具 ID、容器 ID、种类 ID，其 E-R 图见图 3-8。

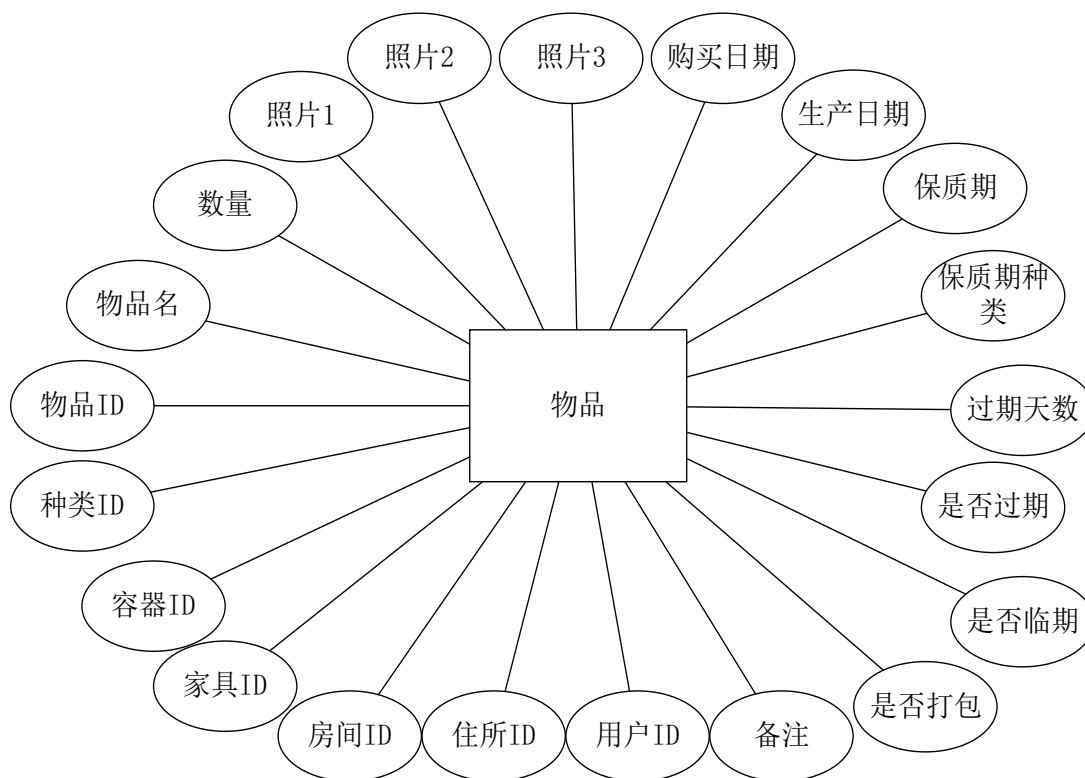


图 3-8 物品实体 E-R 图

系统 E-R 图见图 3-9，在该系统中，1 个用户对应多个住所，1 个住所对应多个房间，1 个住所对应多个水电气等账号，1 个房间对应多个家具，1 个家具对应多个物品，1 个容器对应多个物品，1 个物品对应 1 个物品种类。

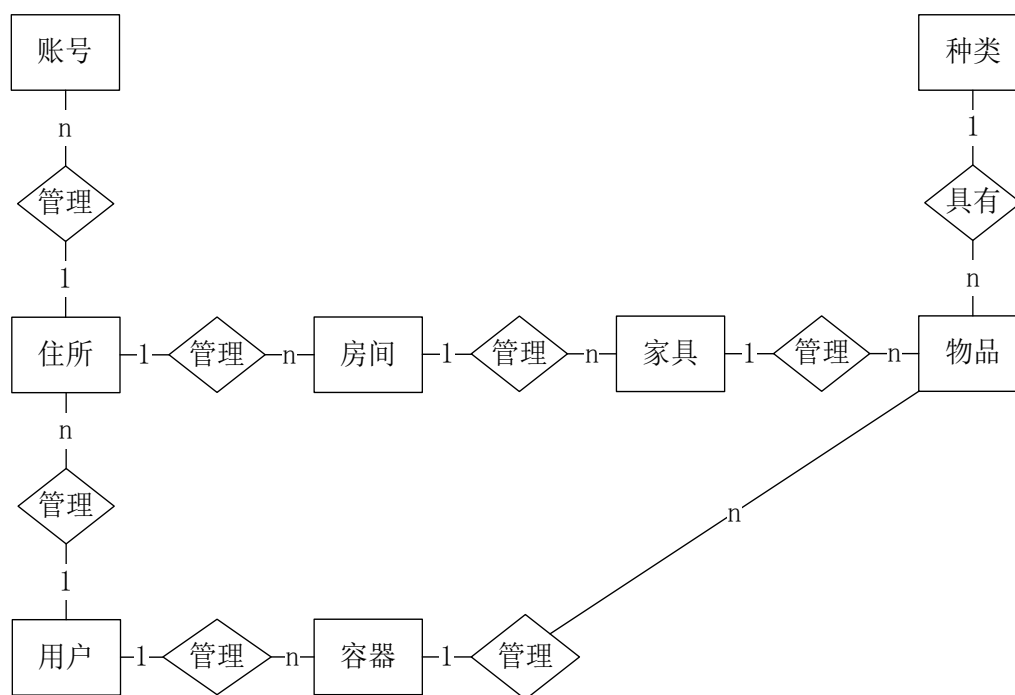


图 3-9 系统 E-R 图

## 4 概要设计

概要设计也叫总体设计,要考量全局问题,目的是从总体上设计 App 处理的方案。

### 4.1 体系结构设计

针对前面的需求分析中的总体目标和具体目标进行分析,可以得出本系统的功能图,如下图所示。

本系统的功能分为有用户、概览、空间、打包、设置五类,见图 4-1。

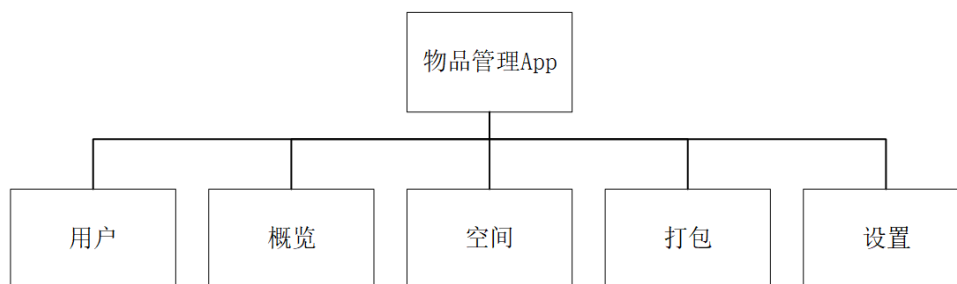


图 4-1 系统总功能图

用户类需要实现系统的登录和账号注册,见图 4-2。

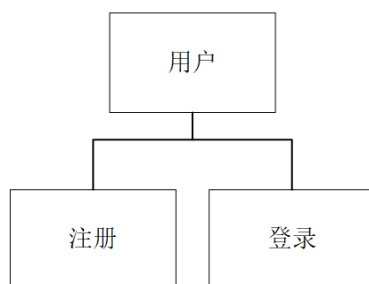


图 4-2 系统用户模块功能图

概览类需要实现切换住所、搜索物品、查询临期和过期物品、当前住所账号的显示和修改、添加物品等功能,见图 4-3。

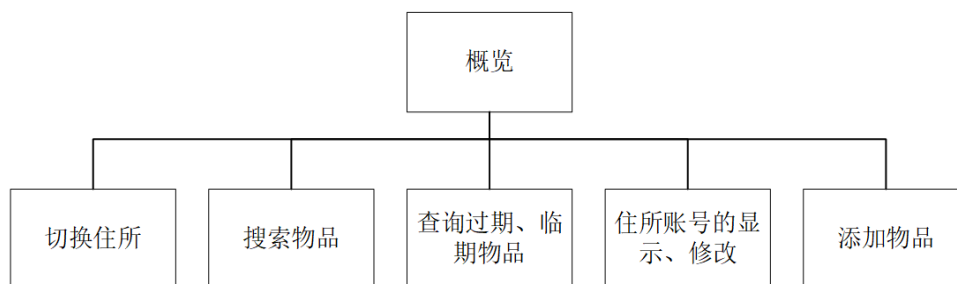


图 4-3 系统概览模块功能图

空间类需要实现切换住所、搜索物品、显示当前住所下所有房间 Tab、显示此房间下所有家具列表、显示此家具下所有物品列表、查看、修改和删除物品等功能,见图 4-4。

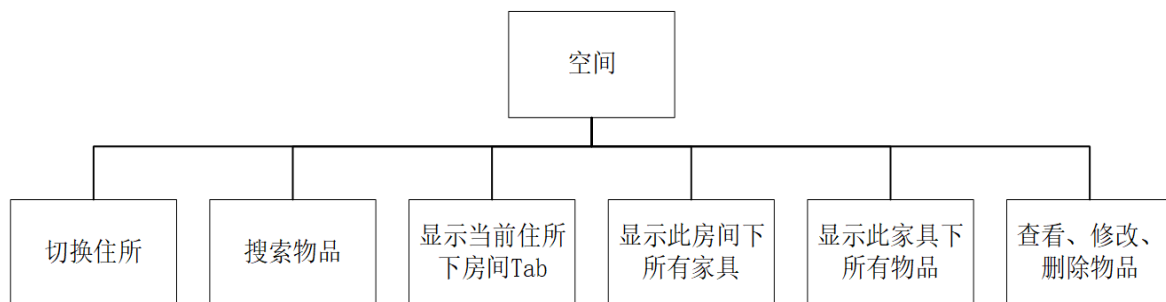


图 4-4 系统空间模块功能图

打包类需要实现切换住所、搜索物品、显示当前用户下所有容器 Tab、显示此容器下所有物品列表、查看、修改和删除物品等功能，见图 4-5。

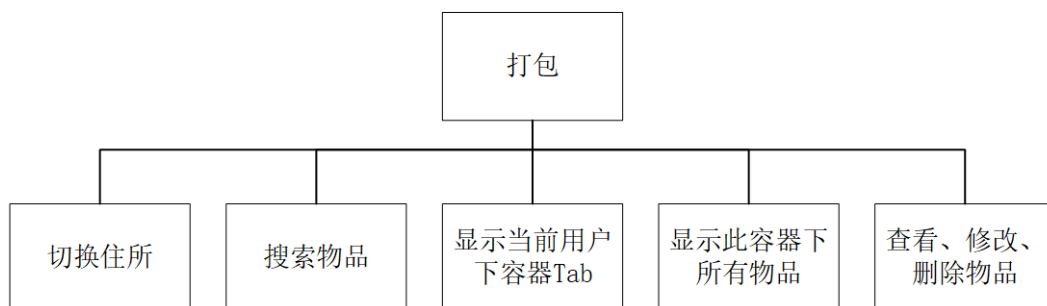


图 4-5 系统打包模块功能图

设置类需要实现修改用户信息、修改判定物品临期时间、管理住所、房间、家具、容器和种类、应用密码的开启、关闭、修改和验证、导出到 Excel 表格、联系开发者、应用评分、关于应用、用户注销等功能，见图 4-6。

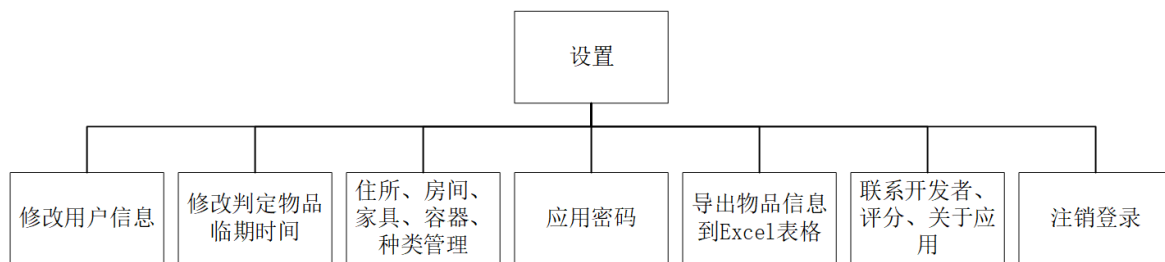


图 4-6 系统设置模块功能图

## 4.2 接口设计

### (1) 系统界面

在进行 App 界面设计时，要从图片、文字、图标、留白等方面考虑。力求设计简约、美观大方，做到界面不花哨，用户在使用功能的时候能很快找到要操作之处，而不是让用户一头雾水。可以参考和利用网络上无版权的原型图来对界面进行搭建。

本 App 的界面包括欢迎界面和主页面，而主页面通过导航栏分成 4 部分，分别是概览、空间、打包和设置四个界面。

## （2）软件接口

物品管理 App 的数据接口是通过使用 Content Provider 来操作 Android 手机自带的 SQLite。使用 Content Provide 对 SQLite 进行绑定,然后使用 Content Resolver,通过对每个表使用 Uri 来区分要操作哪个表。Content Observer 与 DB 中的 Trigger 相似,通过使用 Content Observer 来处理特定 Uri 所引起数据库中所存储数据的改变,来进行对应操作。

## 4.3 代码设计

在总体设计中,最重要的部分是系统代码的设计。

系统代码设计得合理与否对系统的功能、维护等方面会产生很大影响,由此,在进行 App 代码设计的时候应该做到以下要求,从而减少问题的出现。

（1）做到代码复用,避免编写重复的代码,可以将常用的方法封装成工具类;

（2）充分理解需求分析中各个功能的具体要求,厘清各个模块之间的关系,避免做无用功和出现逻辑错误;

（3）对方法、变量或者其他涉及到逻辑的地方添加详细的备注,方便日后进行代码修改时能一眼看出该段代码的功能;

（4）不要写长篇大论、一眼望不到底的代码,可以根据不同功能创建相应方法,在需要的时候进行调用,这样不仅方便阅读代码,而且在后期能快速找到要修改之处;

（5）注意变量的初始化,避免出现不必要的问题。

## 4.4 数据库设计

Android 在系统中自带了 SQLite 数据库,还为 SQLite 数据库提供了许多 API,开发者只需要调用这些 API 即可实现对数据库的访问和修改,数据库的操作可以和 Content Provider 进行结合使用。

本系统将数据主要保存到 SQLite 数据库中,程序通过使用 Content Provider 来操作数据库,通过对数据库进行查找来获得信息,使用 Cursor 游标来进行查找,将查找结果用 Adapter、ViewHolder 和显示在 RecyclerView 上,从而展示在手机界面上。

### 4.4.1 数据库物理结构设计

SQLite 的存储类有 NULL、INTEGER、REAL、TEXT、BLOB,使用 SQLite 时,不需要指定长度,因为它本身会根据内容进行动态匹配。

(1) 用户表, 如表 4-1 所示, 记录用户的用户名、密码、性别、头像、提醒时间信息, 通过 userID 来对用户个体进行区分。

表 4-1 用户数据表

字段名	数据类型	其他	描述
userID	INTEGER	PRIMARY KEY AUTOINCREMENT	用户 ID
username	TEXT	NOT NULL	用户名
password	TEXT	NOT NULL	密码
gender	TEXT	NOT NULL	性别
profilePhoto	MEDIUMBLOB	NOT NULL	头像
reminderTime	INTEGER	NOT NULL	提醒时间

(2) 住所表, 如表 4-2 所示, 记录住所的住所名信息, 通过 houseID 来对住所个体进行区分, userID 用来绑定用户, 记录该住所是属于哪个用户的。

表 4-2 住所数据表

字段名	数据类型	其他	描述
houseID	INTEGER	PRIMARY KEY AUTOINCREMENT	住所 ID
houseName	TEXT	NOT NULL	住所名
userID	INTEGER	FOREIGN KEY	用户 ID

(3) 账号表, 如表 4-3 所示, 记录账号的账号名、账号值信息。

表 4-3 账号数据表

字段名	数据类型	其他	描述
accountID	INTEGER	PRIMARY KEY AUTOINCREMENT	账号 ID
accountName	TEXT	NOT NULL	账号名
accountValue	TEXT	NOT NULL	账号值
houseID	INTEGER	FOREIGN KEY	住所 ID

(4) 房间表, 如表 4-4 所示, 记录房间的房间名信息, 通过 roomID 来对房间个体进行区分, userID 用来绑定用户, 记录该房间是属于哪个用户的。houseID 用来绑定住所, 记录该房间是属于哪个住所的。

表 4-4 房间数据表

字段名	数据类型	其他	描述
roomID	INTEGER	PRIMARY KEY AUTOINCREMENT	房间 ID
roomName	TEXT	NOT NULL	房间名
userID	INTEGER	FOREIGN KEY	用户 ID
houseID	INTEGER	FOREIGN KEY	住所 ID

(5) 家具表, 如表 4-5 所示, 记录家具的家具名信息, 通过 furnitureID 来对家具个体进行区分, userID 用来绑定用户, 记录该家具是属于哪个用户的。houseID 用来绑定住所, 记录该家具是属于哪个住所的。roomID 用来绑定房间, 记录该家具是属于哪个房间的。



表 4-5 家具数据表

字段名	数据类型	其他	描述
furnitureID	INTEGER	PRIMARY KEY AUTOINCREMENT	家具 ID
furnitureName	TEXT	NOT NULL	家具名
userID	INTEGER	FOREIGN KEY	用户 ID
houseID	INTEGER	FOREIGN KEY	住所 ID
roomID	INTEGER	FOREIGN KEY	房间 ID

(6) 容器表, 如表 4-6 所示, 记录容器的容器名信息, 通过 containerID 来对容器个体进行区分, userID 用来绑定用户, 记录该容器是属于哪个用户的。

表 4-6 容器数据表

字段名	数据类型	其他	描述
containerID	INTEGER	PRIMARY KEY AUTOINCREMENT	容器 ID
containerName	TEXT	NOT NULL	容器名
userID	INTEGER	FOREIGN KEY	用户 ID

(7) 种类表, 如表 4-7 所示, 记录种类的种类名信息, 通过 categoryID 来对种类个体进行区分, userID 用来绑定用户, 记录该种类是属于哪个用户的。

表 4-7 种类数据表

字段名	数据类型	其他	描述
categoryID	INTEGER	PRIMARY KEY AUTOINCREMENT	种类 ID
categoryName	TEXT	NOT NULL	种类名
userID	INTEGER	FOREIGN KEY	用户 ID

(8) 物品表, 如表 4-8 所示, 记录物品的名称、数量、照片 1、照片 2、照片 3、购买日期、生产日期、保质期、保质期种类、过期天数、是否处于过期状态、是否处于临期状态、是否处于打包状态、备注信息, 通过 goodsID 来对物品个体进行区分, userID 用来绑定用户, 记录该物品是属于哪个用户的。houseID 用来绑定住所, 记录该物品是属于哪个住所的。roomID 用来绑定房间, 记录该物品是属于哪个房间的。furnitureID 用来绑定家具, 记录该物品是属于哪个家具的。containerID 用来绑定容器, 记录该物品是属于哪个容器的。categoryID 用来绑定种类, 记录该物品是属于哪个种类的。

表 4-8 物品数据表

字段名	数据类型	其他	描述
goodsID	INTEGER	PRIMARY KEY AUTOINCREMENT	物品 ID
goodsName	TEXT	NOT NULL	物品名
INTEGER	INTEGER		数量
goodsPhoto1	MEDIUMBLOB		照片 1
goodsPhoto2	MEDIUMBLOB		照片 2
goodsPhoto3	MEDIUMBLOB		照片 3
buyTime	TEXT		购买日期

续表 4-8 物品数据表

字段名	数据类型	其他	描述
manufactureDate	TEXT		生产日期
qualityGuaranteePeriod	TEXT		保质期
qualityGuaranteePeriodType	INTEGER	NOT NULL	保质期种类
overtime	TEXT		过期天数
isOvertime	BOOLEAN	NOT NULL	是否过期
isCloseOvertime	BOOLEAN	NOT NULL	是否临期
packed	BOOLEAN	NOT NULL	是否打包
remark	TEXT		备注
userID	INTEGER	FOREIGN KEY	用户 ID
houseID	INTEGER	FOREIGN KEY	住所 ID
roomID	INTEGER	FOREIGN KEY	房间 ID
furnitureID	INTEGER	FOREIGN KEY	家具 ID
containerID	INTEGER	FOREIGN KEY	容器 ID
categoryID	INTEGER	FOREIGN KEY	种类 ID

## 4.5 运行设计

### 4.5.1 环境

#### (1) 系统开发环境

系统的开发环境：Android Studio；

系统的开发语言：Java；

系统的数据库：SQLite；

开发环境的运行平台：macOS Big Sur。

#### (2) 客户端

运行平台：Android 8 及更高版本；

分辨率：最佳效果 2248\*1080 像素。

### 4.5.2 运行控制

通过考量本 App 的需求分析，可知该款物品管理 App 主要为用户提供所保存物品的基本信息，对用户进行过期和临期物品的提醒服务。

根据上述基本功能，可将系统划分为 5 个模块：用户、概览、空间、打包、设置。

由于本系统划分了模块，可以针对模块来进行开发，后期维护工作也较为方便。

## 4.6 安全保密设计

### 4.6.1 数据安全性

本物品管理 App 的主要功能是根据条件进行数据查询和将查询结果进行展示，物

品信息涉及个人隐私，用户可以设置应用密码，退出后下次打开 App 时会需要输入密码，5 次机会用完后将应用将被锁定。

#### 4.6.2 登录用户的安全性

打开本 App 后，首先显示欢迎界面，该界面上有应用的图标和 Slogan，此时对密码进行判断。如果存在密码，则打开输入密码界面，如果剩余机会为 0，则界面提示需要到手机设置中清除本 App 所有数据才能使用。如果没有密码或密码输入正确，才可进入本 App。用户在登录和输入正确密码之前，物品管理 App 并不涉及到物品信息的显示，保护了用户的隐私不被侵犯。

### 4.7 维护设计

本 App 系统所注重的后期维护主要为功能更新。在功能更新之前，对用户反馈进行分析，了解用户的需求，目前 App 有哪些缺点，竞品 App 有哪些优点而本 App 没有。

把这些信息进行汇总讨论，分析哪些是可行的，定期反馈给开发人员，对功能上线进行排期，从而满足用户的需求，以出色的体验吸引新用户，同时使老用户不流失。

## 5 详细设计

### 5.1 详细设计概述

详细设计可以看作是对总体设计的精细化，对前面所分析出的模块完成算法的详细设计，完成 App 界面的布局等等。

### 5.2 界面设计

主界面的设计要简洁，通过底部导航栏和 4 个 Fragment 来组成主界面，4 个 Fragment 分别为概览、空间、打包、设置 4 个模块。

一个合格的系统界面少不了图片的显示，如果界面上需要显示特定图片，本 App 选择添加一些可缩放矢量图形 SVG。通过对 App 主题的设置使得整体效果更符合 Google Material Design 的审美。Material Design 是由 Google 推出的设计语言，旨在统一 Android 程序界面设计风格<sup>[13]</sup>。

### 5.3 用户模块定义

#### (1) 模块定义

模块名称：用户。

模块功能：本模块中主要进行用户登录、注册，登录成功后跳转到主界面。

## (2) 输入项目

用户名；密码；头像；性别。

## (3) 输出项目

当 App 验证登录或注册信息无误的时候，跳转到 App 主界面。

主要输出项目有：用户登录成功；用户登录失败。

## (4) 程序逻辑

本模块涉及到用户登录、注册及应用密码的使用，见图 5-1。

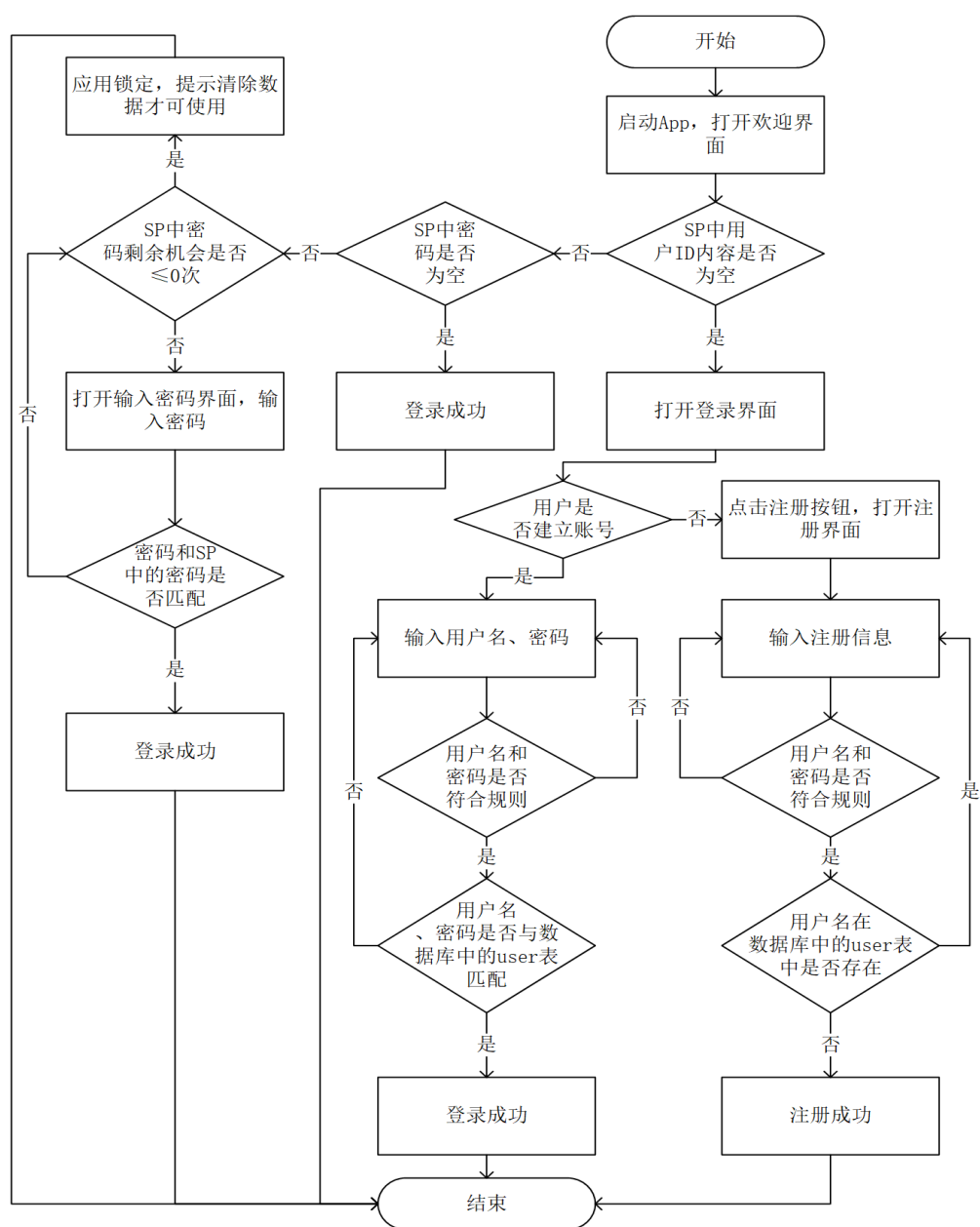


图 5-1 用户模块流程图

## 5.4 概览模块定义

### (1) 模块定义

模块名称：概览。

模块功能：本模块中主要功能有：显示过期临期物品列表及物品的管理；切换住所；搜索物品；查看和修改账号；新建物品。

### (2) 输入项目

要创建物品的信息；要修改的账号信息；搜索信息；要切换的住所信息；要修改和删除的物品信息。

### (3) 输出项目

主要输出项目有：物品列表；账号信息。

### (4) 程序逻辑

#### 1) 过期临期物品管理

过期临期物品管理涉及到物品的条件查找、修改和删除，见图 5-2。

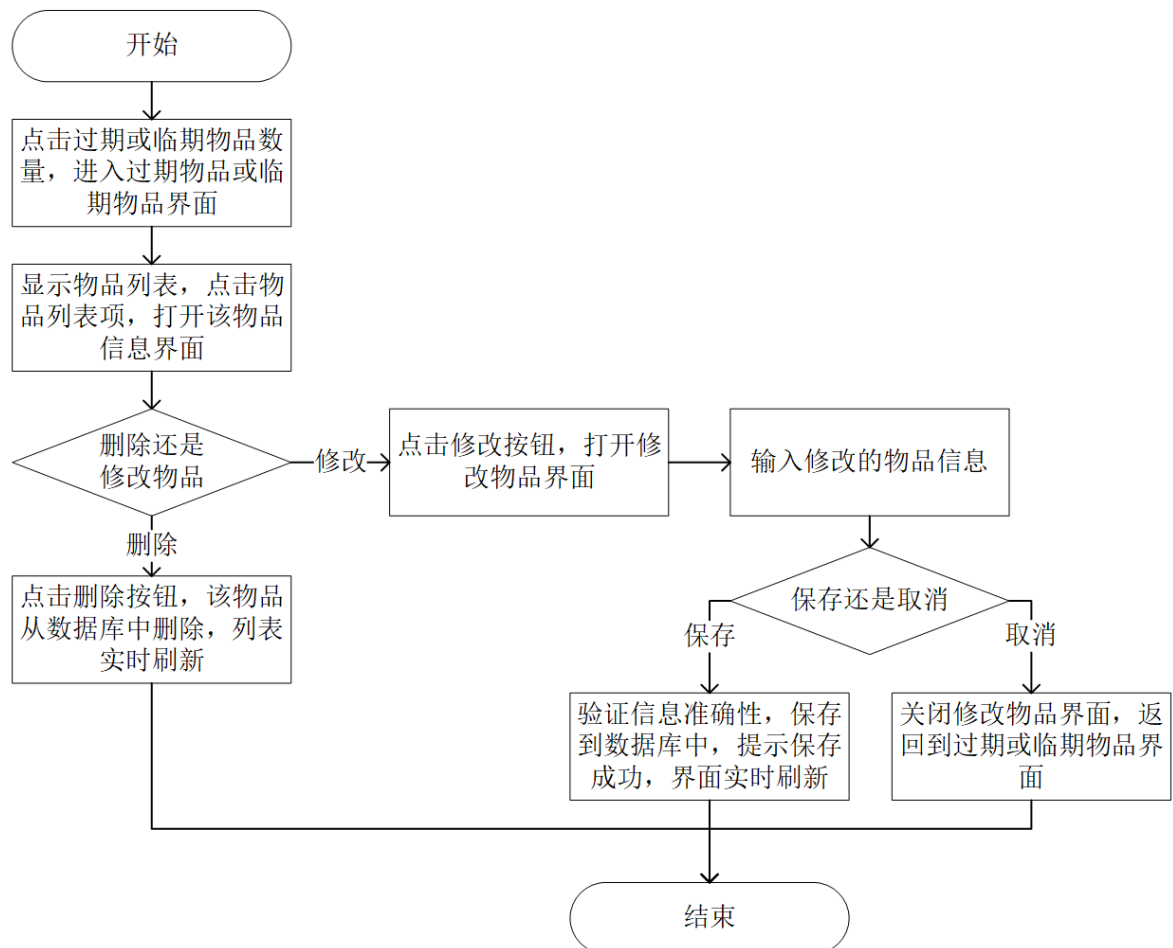


图 5-2 过期临期物品管理流程图

## 2) 切换住所:

点击当前住所名称,弹出该用户所有住所列表,点击任意项即可切换到该住所,同时界面中的信息刷新为该住所的有关信息。

## 3) 搜索物品

打开搜索界面,输入条件信息,选择在当前住所还是所有住所中搜索,点击确定按钮即可显示符合条件的物品列表。点击物品项可打开查看物品信息的界面,可选择删除和修改物品,删除或修改物品后,将结果保存到数据库中,并且查询结果列表实时刷新。

## 4) 查看和修改账号

界面中显示当前住所的水电气等账号信息,长按可以复制账号到剪切板,点击账号右边的修改按钮,打开修改账号对话框,输入账号信息,点击保存按钮,即可将其存入数据库并刷新界面,点击取消按钮,关闭对话框。

## 5) 新建物品

打开新建物品界面,输入物品信息并保存,系统会校验信息的准确性,如果准确,则保存到数据库中,所有物品列表和临期过期物品的数量也会刷新,如果不准确,提示错误,让用户修改。

# 5.5 空间模块定义

## (1) 模块定义

模块名称: 空间。

模块功能: 本模块中主要功能有: 切换住所; 显示放置在家具中的物品及物品的查看修改和删除。

## (2) 输入项目

要切换的住所信息; 选择住所、房间、家具; 要修改和删除的物品信息。

## (3) 输出项目

主要输出项目有: 房间列表; 家具列表; 物品列表; 物品修改或删除是否成功的提示。

## (4) 程序逻辑

### 1) 切换住所

点击当前住所名称,弹出该用户所有住所列表,点击任意项即可切换到该住所,

同时界面中的信息刷新为该住所的有关信息。

## 2) 家具中物品管理

家具中物品管理涉及到房间、家具、物品的条件查找、修改和删除，见图 5-3。

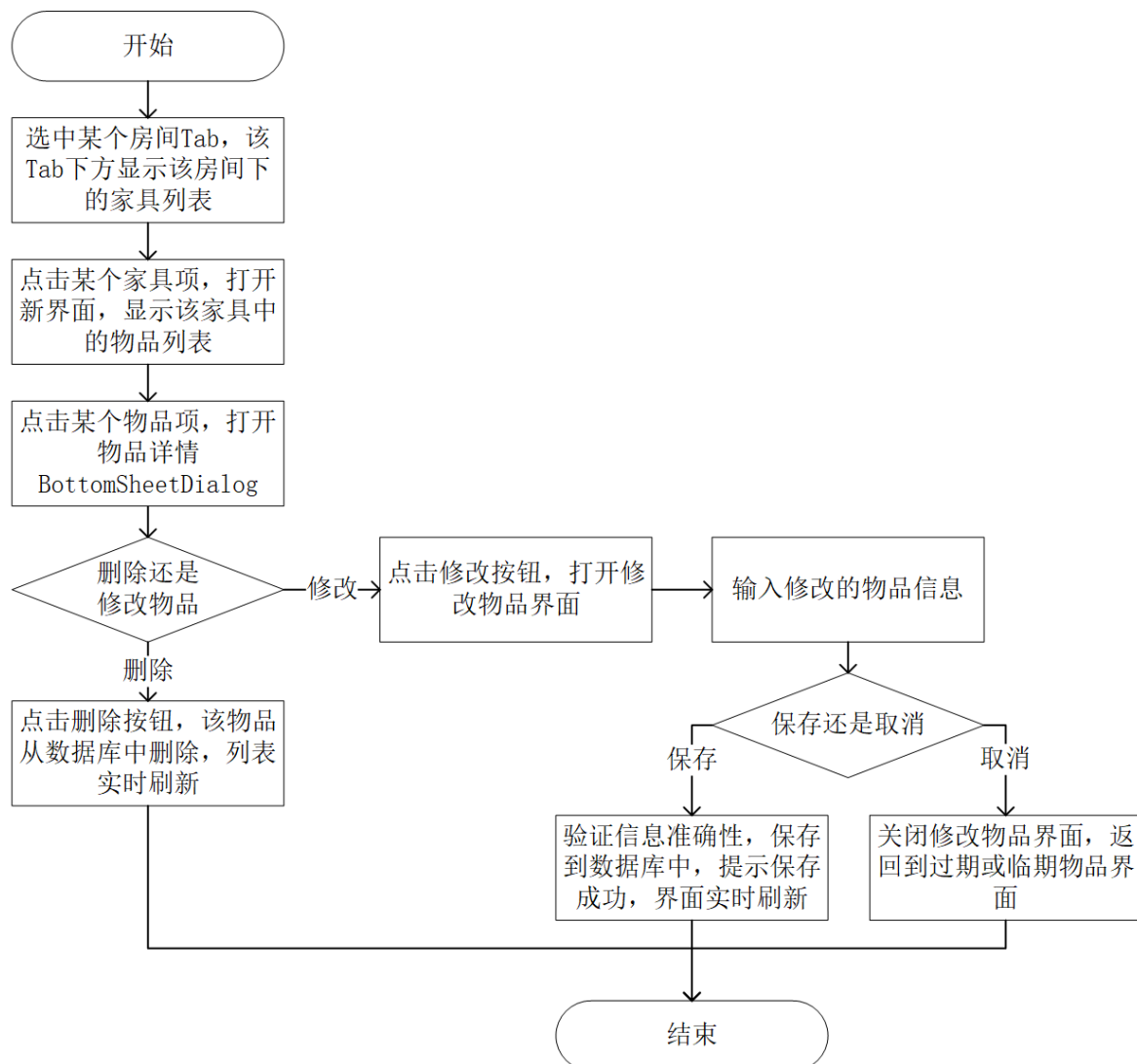


图 5-3 家具中物品管理流程图

## 5.6 打包模块定义

### (1) 模块定义

模块名称：打包。

模块功能：本模块中主要功能有：切换住所；显示处于容器里面的物品及物品的查看修改和删除。

### (2) 输入项目

要切换的住所信息；选择容器；要修改和删除的物品信息。

### (3) 输出项目

主要输出项目有：容器列表；物品列表；物品修改或删除是否成功。

#### (4) 程序逻辑

##### 1) 切换住所

点击当前住所名称，弹出该用户所有住所列表，点击任意项即可切换到该住所，同时界面中的信息刷新为该住所的有关信息。

##### 2) 容器中物品管理

容器中物品涉及到容器、物品的条件查找、修改和删除，见图 5-4。

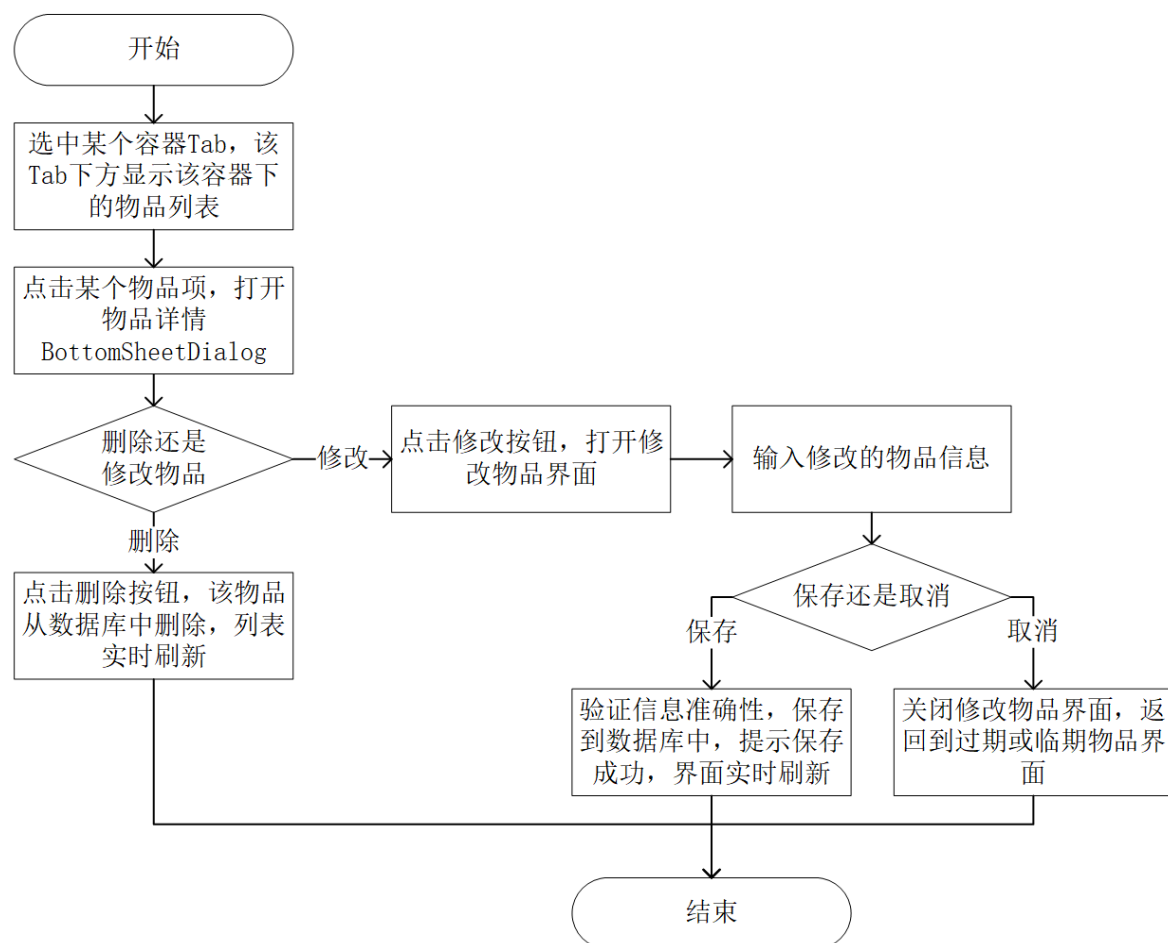


图 5-4 查看修改和删除容器中物品流程图

## 5.7 设置模块定义

### (1) 模块定义

模块名称：设置。

模块功能：本模块中主要功能有：修改用户头像；修改物品临期的判断天数；住所、房间、家具、容器、种类管理；应用密码管理；导出物品信息到 Excel 表格；联系开发者；评分。



## (2) 输入项目

用户头像；物品临期的判断天数；要创建和修改的住所、房间、家具、容器、种类的信息；应用密码。

## (3) 输出项目

主要输出项目有：住所、房间、家具、容器、种类的列表；各功能执行的结果。

## (4) 程序逻辑

### 1) 修改用户头像

修改头像涉及到权限申请、读取相册和相机、裁剪照片。

从相机中拍照需要获取相机权限，从相册中选择照片需要获取读取和写入存储权限。通过 Intent 调用系统相册的裁剪功能，设置默认裁剪比例为 1:1，该功能的流程图见图 5-5。

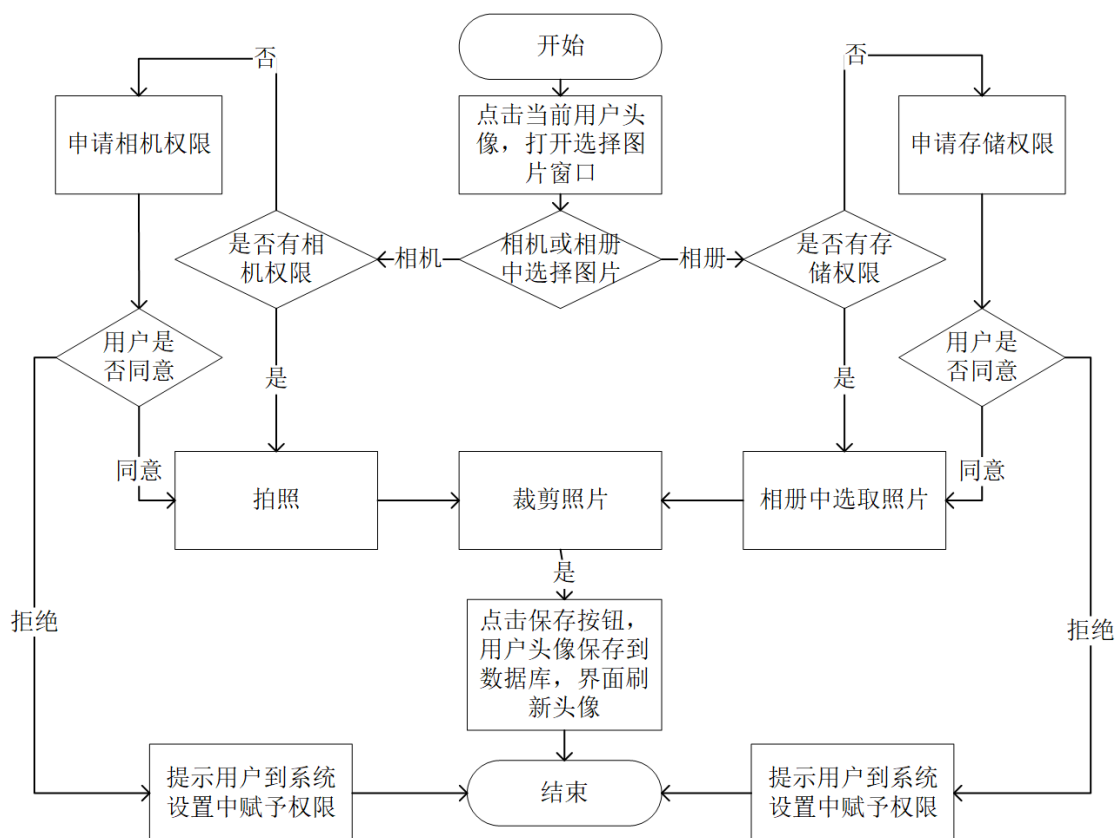


图 5-5 修改用户头像流程图

### 2) 修改物品临期的判断天数

点击设置项，打开修改物品临期的判断天数的 Dialog，输入天数并保存，即可保存到数据库中，点击取消按钮，关闭 Dialog。

### 3) 住所、房间、家具、容器、种类管理

查看：点击对应设置项，打开的界面中显示其列表。

修改、删除、添加：点击列表项修改，左划列表项删除，点击按钮添加，逻辑如图 5-6 所示。

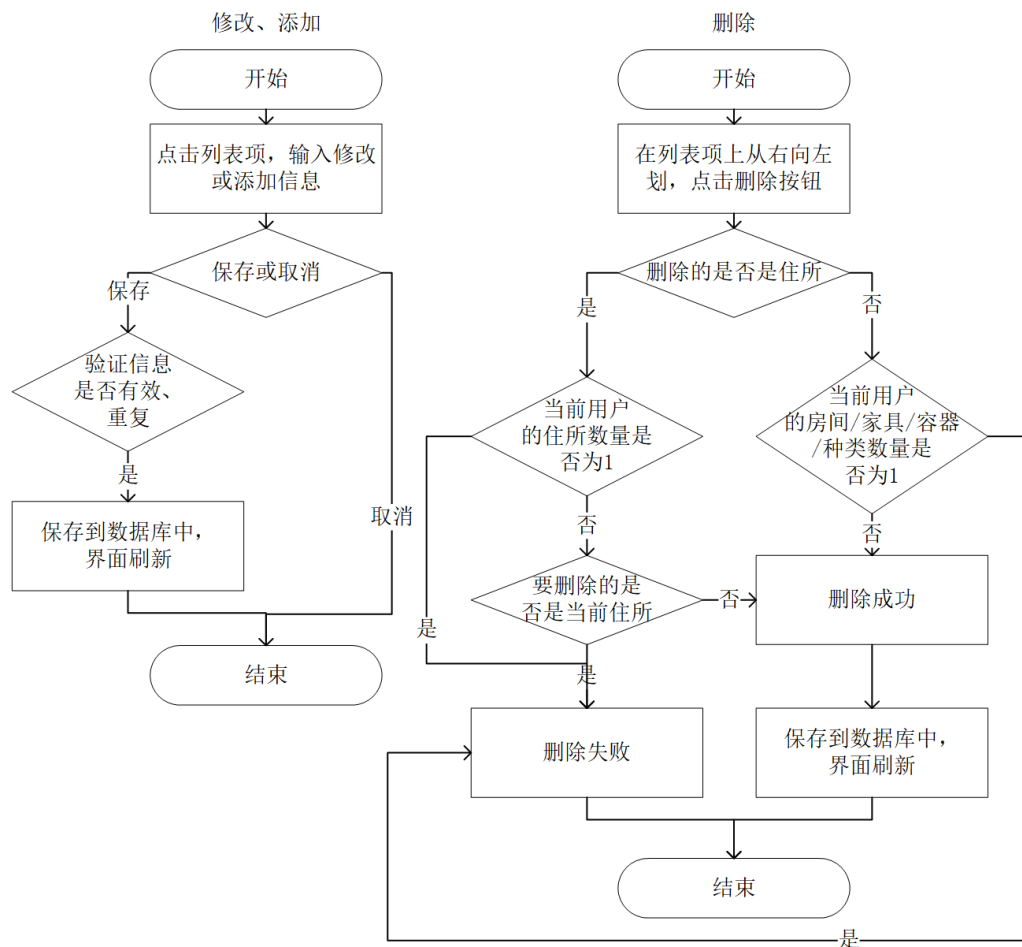


图 5-6 修改删除添加流程图

#### 4) 导出物品信息到 Excel 表格

在用户设备的根目录里创建 Excel 文档，将当前用户去除图片的所有物品信息导出到该文档中，该文档保存到设备根目录下，文件名称为导出物品表+系统当前时间戳。

#### 5) 联系开发者

打开设备上的邮箱 App，自动按照指定模板创建新邮件，待用户完善内容即可发送给开发者。若没有安装邮箱 App，则进行提示；

#### 6) 评分

打开设备自带的应用商店，并跳转到本应用的详情，若没有安装应用商店，则进行提示。

#### 7) 应用密码管理

应用密码管理涉及到对 SP 数据的读取和写入，通过判断错误次数来锁定应用，

见图 5-7。

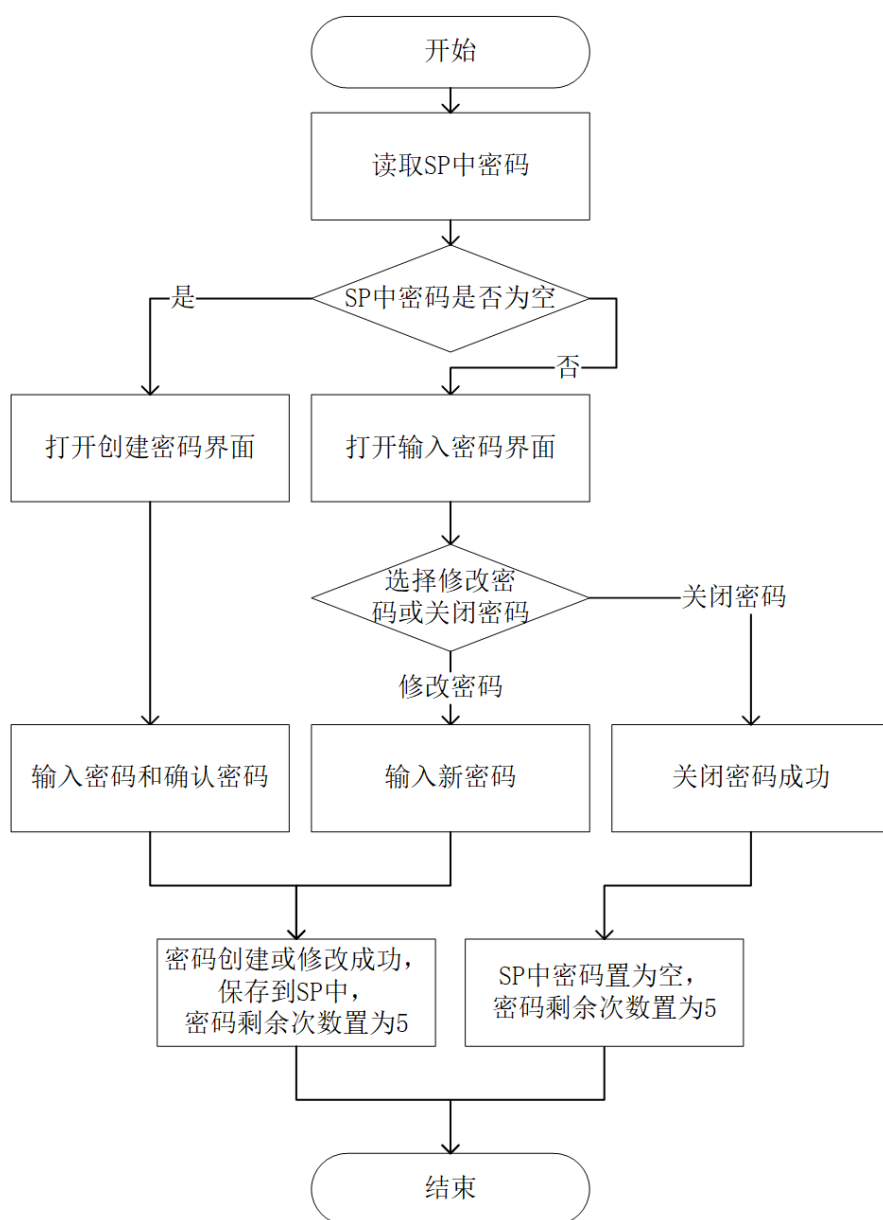


图 5-7 应用密码管理流程图

## 5.8 适配器实现

由于本 App 需要显示住所、房间、家具、物品等列表，还需要对列表项进行增删改的操作。这些功能不仅需要读写数据库，还要求在界面上实时刷新。因此，在各个列表的显示上用到了 RecyclerView，此控件需要用 Adapter 进行初始化。由于这些列表内容不同，故不能进行复用，下面以打包模块中的物品列表进行举例说明。

### (1) 列表项样式

列表项样式见 5-1，从上到下依次是照片、名称、状态，其中状态有三种，绿色、橙色、红色分别代表未过期且未临期、已临期、已过期。

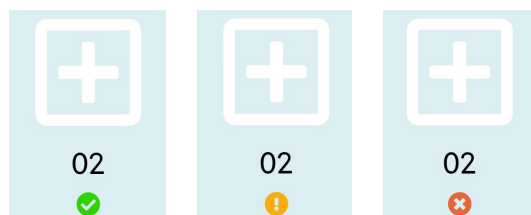


图 5-8 列表项样式图

## (2) 列表项显示内容代码

//ViewHolder 用于承载每一个列表项的视图

```
public class MyViewHolder extends RecyclerView.ViewHolder {
    public TextView goodsName; //物品名称
    public ImageView goodsPhoto; //物品照片
    public ImageView overdue; //物品状态
    public ViewGroup layout_content; //列表项布局
    public MyViewHolder(View itemView) { //根据 ID 获取列表项布局中的控件
        super(itemView);
        goodsName = itemView.findViewById(R.id.goodsName);
        goodsPhoto = itemView.findViewById(R.id.goodsPhoto);
        overdue = itemView.findViewById(R.id.overdue);
        layout_content = itemView.findViewById(R.id.layout_content);
    }
}
```

## (3) 此列表中的物品实体定义代码

//GoodsEntity 用于定义从 SQLite 中查询出的物品信息实体

```
public static class GoodsEntity {
    int goodsID; //物品 ID
    String goodsName; //物品名称
    byte[] goodsPhoto; //物品照片
    int overdue; //物品状态
}
```

## (4) 列表适配器代码

```
public class PackGoodsAdapter extends RecyclerView.Adapter<PackGoodsAdapter.
MyViewHolder> {
    SharedPreferences sharedPreferences; //定义 SP 数据，用于从读取当前用户 ID
    int userID, containerID; //定义用户 ID、容器 ID
    Activity activity; //定义活动，用于实现点击物品打开 BottomSheetDialog
    PackContent packContent; // Tab 项的 Fragment，用于调用其中的重设适配器方法
    private Context mContext; //定义上下文
    private List<GoodsEntity> listGoods = new ArrayList<>(); //定义物品实体集合
    public PackGoodsAdapter(Context mContext, String id, PackContent packContent,
        Activity activity) { //适配器构造方法，用于初始化上述定义的变量
        this.mContext = mContext;
        this.packContent = packContent;
        this.activity = activity;
    }
}
```

```

sharedPreferences = this.mContext.getSharedPreferences("here", Context.
    MODE_PRIVATE);
userID = sharedPreferences.getInt("userID", -1); //从 SP 中获取用户 ID
// 定义游标, 从 SQLite 中查询当前用户的所有容器, 获取其 ID
Cursor containerCursor = queryContainer(this.mContext, new String[]{"containerID"}, "userID=?", new String[]{String.valueOf(userID)});
containerCursor.moveToPosition(Integer.parseInt(id));
containerID = containerCursor.getInt(0);
// 定义游标, 从数据库中查询当前用户和当前容器中状态为打包的物品
Cursor goodsCursor = queryGoods(this.mContext, new String[]{"goodsID",
    "goodsName", "goodsPhoto1", "isOvertime", "isCloseOvertime",
    "userID=? AND containerID=? AND packed=?", new String[]{String.
    valueOf(userID), String.valueOf(containerID), String.valueOf(1)});
// 定义 5 个数组用于存放查询的物品信息
int[] temp1 = new int[goodsCursor.getCount()];
String[] temp2 = new String[goodsCursor.getCount()];
byte[][] temp3 = new byte[goodsCursor.getCount()][];
short[] temp4 = new short[goodsCursor.getCount()];
short[] temp5 = new short[goodsCursor.getCount()];
int overdue; //用于存放物品状态图片的资源 ID
//遍历物品查询结果, 对物品状态进行判断, 之后保存到物品集合中
for (int i = 0; goodsCursor.moveToNext(); i++) {
    temp1[i] = goodsCursor.getInt(0);
    temp2[i] = goodsCursor.getString(1);
    temp3[i] = goodsCursor.getBlob(2);
    temp4[i] = goodsCursor.getShort(3);
    temp5[i] = goodsCursor.getShort(4);
    if (temp4[i] == 1) {
        overdue = mContext.getResources().getIdentifier("ic_overdue",
            "drawable", mContext.getPackageName()); }
    else if (temp5[i] == 1) {
        overdue = mContext.getResources().getIdentifier("ic_close_
            overdue", "drawable", mContext.getPackageName()); }
    else {
        overdue = mContext.getResources().getIdentifier("ic_check",
            "drawable", mContext.getPackageName()); }
    GoodsEntity goodsEntity = new GoodsEntity(temp1[i], temp2[i], temp3[i],
        overdue);
    listGoods.add(goodsEntity);
}
}

//当需要新的 ViewHolder 来显示列表项时, 会调用此方法去创建
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(mContext).inflate(R.layout.item_goods,

```

```

        parent, false); //获取列表中每行的布局文件
        return new MyViewHolder(view);
    }
    //将物品实体信息绑定在 ViewHolder 上, 设置列表项显示内容
    public void onBindViewHolder(MyViewHolder holder, int position){
        holder.goodsName.setText(listGoods.get(position).goodsName); //设置名称
        //将物品照片信息 (byte[]) 转换为 Bitmap 图片
        Bitmap bitmap = byteArrayToBitmap(listGoods.get(position).goodsPhoto);
        holder.goodsPhoto.setImageBitmap(bitmap); //设置物品照片
        //设置物品状态图片
        holder.overdue.setImageResource(listGoods.get(position).overdue);
        holder.layout_content.setOnClickListener(v -> { //设置列表项的点击监听
            //定义和显示物品详情界面
            PackGoodsDialog packGoodsDialog = new PackGoodsDialog();
            packGoodsDialog.showBottomSheetDialog(mContext, activity, listGoods.
                get(position).goodsID, this, holder, position);
        });
    }
    public void delete(int position) { //物品详情中, 点击删除按钮时调用
        deleteGoods(mContext, "goodsID=?", new String[] { String.valueOf(
            listGoodsEntity.get(position).goodsID)});
        packContent.reSetAdapter();
    }
}

```

## 5.9 服务实现

由于本 App 保存了物品的生产日期、保质期、物品过期临期状态等数据, 并且这些数据保存到了数据库中, 故需要每天至少对物品的过期临期状态进行一次更新。该物品管理 App 会在用户启动应用时进行一次更新, 在启动后不杀死进程的情况下每 6 小时进行一次更新。更新时, 如果存在过期或临期物品, 需要发送通知提醒用户, 用户点击通知即可打开本 App。

上述功能需要监听系统事件和后台更新数据库, 并且更新数据库和发送通知时不需要显示用户界面, 故选用 Android 的 BroadcastReceiver 组件和 Service 组件进行实现。

### (1) 定义 BroadcastReceiver 和 Service

```

//定义开机监听
<receiver android:name=".service.BootReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED">
    </action>

```

```

</intent-filter>
</receiver>
//定义定时监听
<receiver android:name=".AlarmReceiver" />
//定义服务
<service android:name=".LongRunningService"
          android:enabled="true"
          android:exported="true" />

```

## (2) 服务实现逻辑

本服务对物品表中的特定物品进行查询，根据系统当前时间来更新物品状态。如果有临期或过期物品存在则发送通知，见图 5-9。

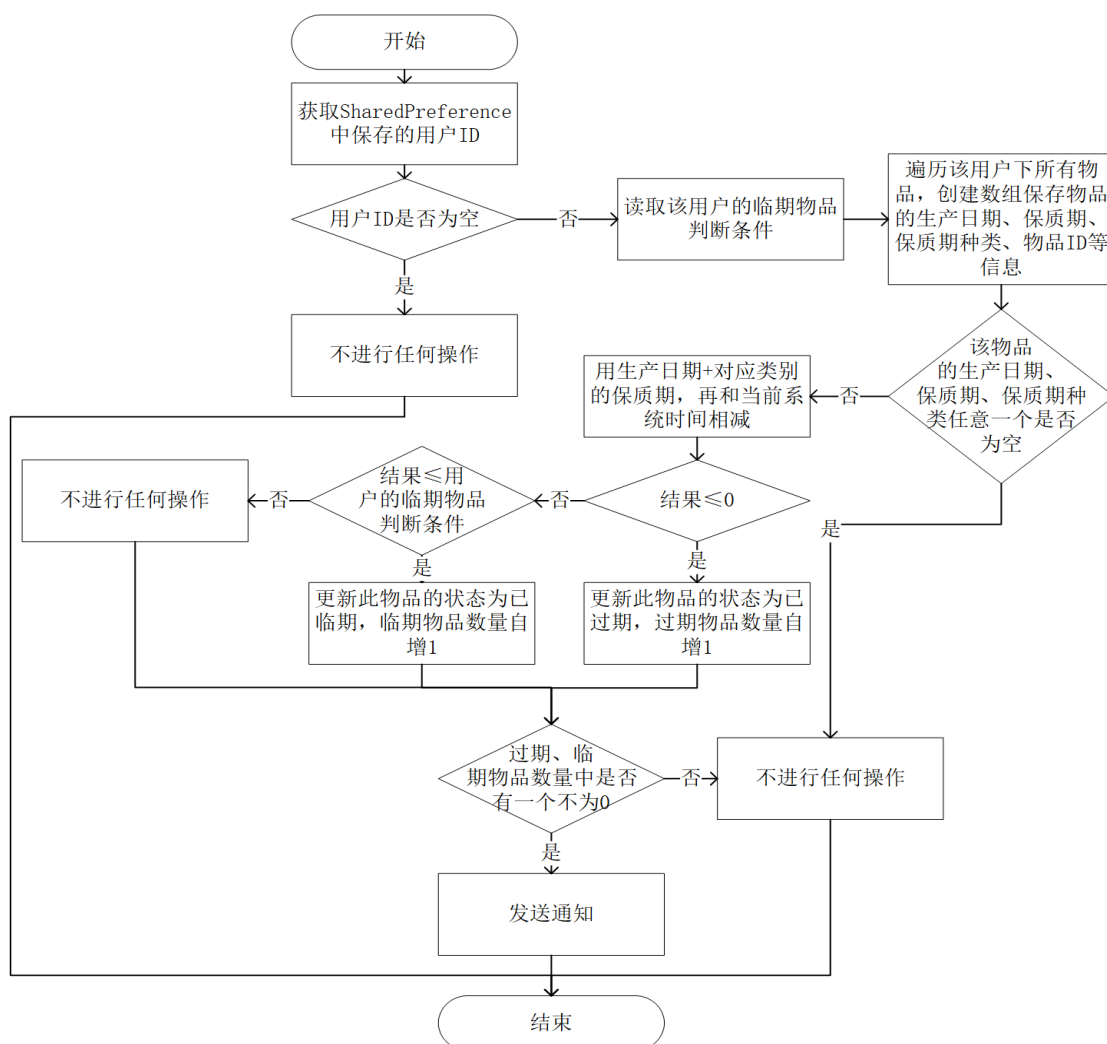


图 5-9 服务流程图

## 5.10 界面实现

(1) 用户模块：登录活动上有用户账号和密码 EditText、Login Button、跳转到注册 Button；登录活动上有用户账号和密码 EditText、性别 Radio Button、注册 Button、跳转到登录 Button。若信息输入错误，界面上会以红色字体进行提示。



图 5-10 注册、登录注册界面图

(2) 概览模块：最上面从左到右依次是切换住所 PopupWindow、标题、搜索，下面是物品临期过期数量，然后是当前住所的账号信息，右下角是添加物品按钮。



图 5-11 概览界面图

(3) 空间模块：最上面从左到右依次是切换住所、标题，下面是当前住所下的房间 Tab，然后是当前选中房间 Tab 下的家具列表，点击家具打开该家具下的物品列



表，点击对应物品可以查看详情。

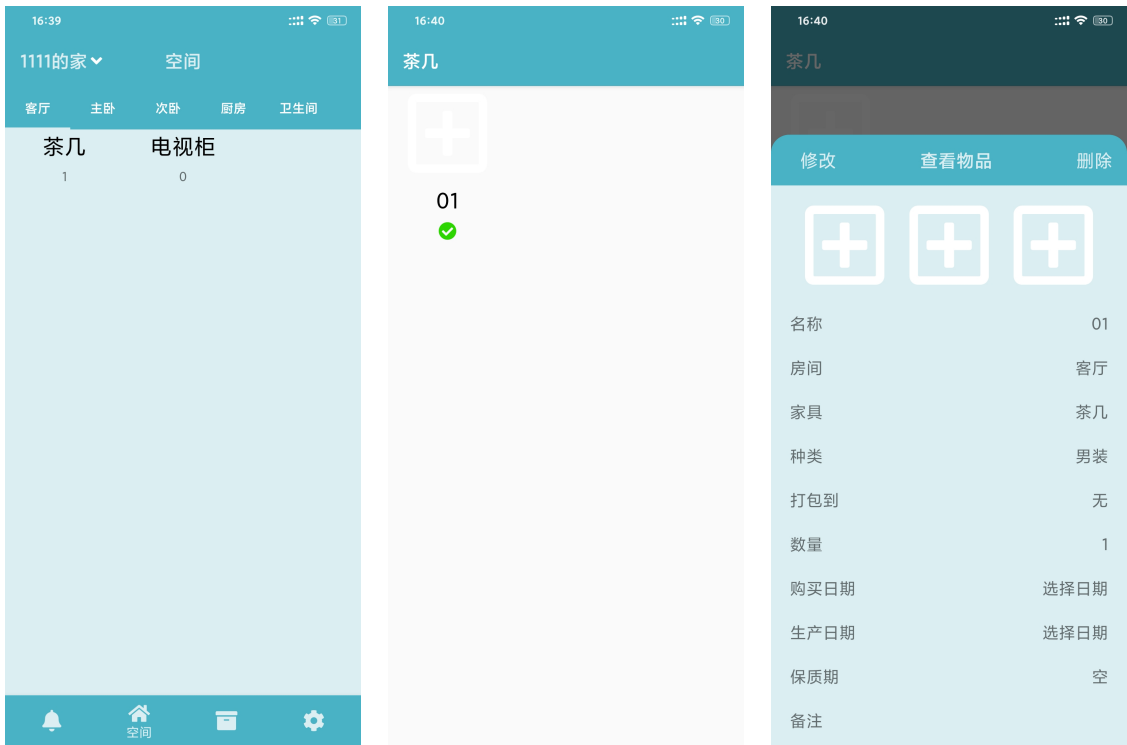


图 5-12 空间界面图

(4) 打包-主界面：最上面从左到右依次是切换住所 PopupWindow、标题 TextView，下面是当前用户下的容器 Tab，Tab 下方是当前选中容器 Tab 下的物品列表，点击对应物品可以查看详情。



图 5-13 打包界面图

(5) 设置模块：纵向布局依次是用户信息、通用组、安全组、其他组、注销按

钮。通用设置组包括过期物品提前通知、住所、房间、家具、容器、种类；安全设置组包括密码、导出为表格；其他设置组包括联系开发者、评分、关于。



图 5-14 设置界面图

## 6 测试

### 6.1 测试原则及测试方法概述

首先对待测模块开始单元测试，如果全部通过，再整合模块和设计信息，开始集成测试。通过后，得到已经集成的软件，再和软件需求开始确认测试。通过后，得到已经确认的软件，再和其他元素开始系统测试。通过系统测试后，得到可交付的软件。

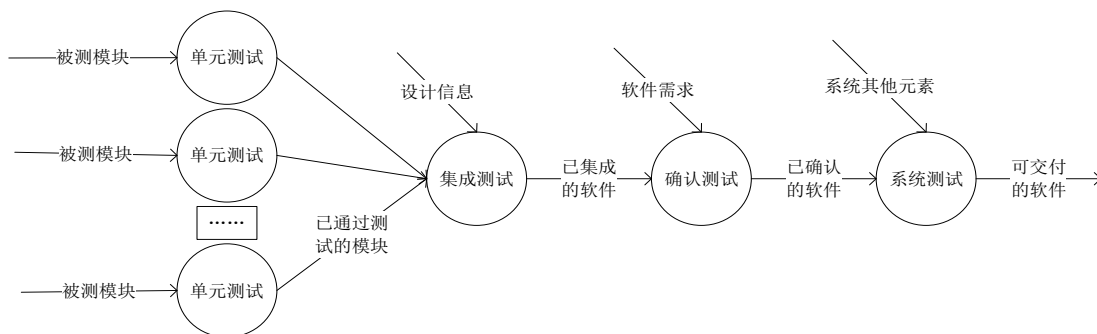


图 6-1 软件系统的测试过程图

### 6.1.1 测试原则

(1) 软件测试不仅要尽早测试，也要不断地进行测试<sup>[14]</sup>。这样能把错误在早期解决完毕，后期出错的数量会大大减少，最大程度地减少成本，提高软件的健壮性和质量；

(2) 测试结果说明不了软件没有任何瑕疵，证明不了软件是完备的、毫无错误的<sup>[15]</sup>；

(3) 在进行测试之前要根据实际情况来选择用例，用例由输入数据和预期输出效果构成；

(4) 在测试中使用不合法输入是为了发现问题，使用合法输入是为了验证模块的功能能否正常工作。

### 6.1.2 测试方法

本 App 主要进行黑盒测试，着重检验 App 的功能，在测试过程中力求寻找以下种类的差错。

- (1) 功能错误或遗漏功能；
- (2) 界面显示错误；
- (3) 软件逻辑错误；
- (4) 性能错误；
- (5) 初始化和终止错误。

## 6.2 测试项目测试用例

对用户模块进行测试，表 6-1 为测试用例说明及测试结果。

表 6-1 用户模块测试用例

功能	测试用例描述	操作过程及数据	预期结果	实际结果
注册	输入不满足规则的账号、不满足规则的密码、与密码不同的确认密码	1; 1; null	注册失败	同预期
	输入满足规则的账号、满足规则的密码、与密码相同的确认密码	1111 ; 111111 ; 111111	注册成功	同预期
登录	输入满足规则但不存在的账号和密码	2222; 222222	登录失败	同预期
	输入不满足规则的账号和密码	22; 2222	登录失败	同预期
	输入满足规则并且存在的账号和密码	1111; 111111	登录成功	同预期

对概览模块进行测试，表 6-2 为测试用例说明及测试结果。

表 6-2 概览模块测试用例

功能	测试用例描述	操作过程及数据	预期结果	实际结果
显示过期临期物品列表及物品的查看修改和删除	显示过期物品列表	点击过期物品数量	打开过期物品界面，显示列表	同预期
	显示临期物品列表	点击临期物品数量	打开临期物品界面，显示列表	同预期
	删除物品	在物品列表中点击物品项，点击删除按钮	物品在 SQLite 中删除，物品列表刷新	同预期
	修改物品	在物品列表中点击物品项，点击修改按钮，输入要修改的信息	物品在 SQLite 中修改，物品列表刷新	同预期
切换住所	点击要切换的目标住所	点击 2222	打开欢迎界面，跳转到显示 2222 住所相应信息的主界面	同预期
搜索物品	输入搜索信息和选项	点击搜索按钮，输入 1，选择搜索所有住所	显示所有住所中物品名称包含 1 的物品	同预期
复制账号	长按账号	长按账号	账号复制到粘贴板上	同预期
修改账号	点击编辑按钮，输入新账号	点击编辑按钮，输入 10000	账号保存到 SQLite 中，界面上的账号同时刷新成新账号	同预期
新建物品	点击新建物品按钮，输入物品信息	点击新建物品按钮，选择从相机拍摄照片，物品名为台灯，房间、家具、类别为各个选项的第二个，打包开关打开，容器为选项的第二个，购买日期、生产日期为 2021 年 4 月 1 日，保质期为 20 天，备注为空，点击保存按钮	物品成功添加到 SQLite 中；点击过期物品列表，会显示该物品	同预期

对空间模块进行测试，表 6-3 为测试用例说明及测试结果。

表 6-3 空间模块测试用例

功能	用例描述	操作过程及数据	预期结果	实际结果
切换住所	点击要切换的目标住所	点击 2222	打开初始界面，1 秒后进入主程序，布局信息已经切换为 2222 住所的信息	同预期
显示放置在家具中的物品	显示当前住所的所有房间 Tab	点击导航栏中的“空间”按钮。	打开“空间”界面，显示当前住所中所有房间 Tab	同预期
	房间 Tab 下显示该房间中的家具列表	点击“卧室”Tab	TabLayout 下方显示“卧室”中的所有家具和家具中的物品数量	同预期
	点击某个家具，显示该家具中的物品列表	点击“衣柜”	打开新界面，显示“衣柜”中所有的物品列表	同预期

续表 6-3 空间模块测试用例

功能	用例描述	操作过程及数据	预期结果	实际结果
物品的查看 修改和删除	删除物品	在物品列表中点击物品项，点击删除按钮	物品在 SQLite 中删除，物品列表刷新	同预期
	修改物品	在物品列表中点击物品项，点击修改按钮，输入要修改的信息	物品在 SQLite 中修改，物品列表刷新	同预期

对打包模块进行测试，表 6-4 为测试用例说明及测试结果。

表 6-4 打包模块测试用例

功能	用例描述	操作过程及数据	预期结果	实际结果
切换住所	点击要切换的目标住所	点击 2222	打开初始界面，1 秒钟后进入主程序，布局信息已经切换为 2222 住所的信息	同预期
显示放置在容器中的物品及物品的查看修改和删除	显示当前用户的所有容器 Tab	点击导航栏中的“打包”按钮。	打开“打包”界面，显示当前用户的所有容器 Tab	同预期
	容器 Tab 下显示该容器中的物品列表	点击“行李箱”Tab	TabLayout 下方显示“行李箱”中的所有物品列表	同预期
	删除物品	在物品列表中点击物品项，点击删除按钮	物品在 SQLite 中删除，物品列表刷新	同预期
	修改物品	在物品列表中点击物品项，点击修改按钮，输入要修改的信息	物品在 SQLite 中修改，物品列表刷新	同预期

对设置模块进行测试，表 6-5 为测试用例说明及测试结果。

表 6-5 设置模块测试用例

功能	用例描述	操作过程及数据	预期结果	实际结果
修改用户头像信息	选择头像	点击用户头像，点击相机按钮，点击拍照按钮，点击确定按钮，点击保存按钮	新头像保存到 SQLite 中，同时界面刷新用户头像	同预期
修改物品临期的判断天数	输入修改的信息	点击该设置项，打开对话框，输入要设置的天数，点击保存按钮	新信息保存到 SQLite 中，同时刷新物品表中的物品临期状态	同预期
住所、房间、家具、容器、种类管理	添加	点击添加按钮，打开对话框，输入新信息，点击保存按钮	SQLite 中保存该项，同时刷新列表	同预期
	删除	在列表中找到要删除的项，在项上从右向左划，点击删除按钮	SQLite 中删除该项，同时刷新列表	同预期
	修改	点击要修改的项，打开对话框，输入新信息，点击保存按钮	SQLite 中修改该项，同时刷新列表	同预期
应用密码管理	创建密码	在没有设置密码的状态下点击“密码”设置，打开新建密码活动，输入两次 1111	创建成功，退出应用后重进需要输入密码	同预期

续表 6-5 设置模块测试用例

功能	用例描述	操作过程及数据	预期结果	实际结果
应用密码管理	创建密码	在没有设置密码的状态下点击“密码”设置, 打开新建密码活动, 输入两次 1111	创建成功, 退出应用后重进需要输入密码	同预期
	修改密码	在有密码的状态下点击“密码”设置, 输入原密码 1111, 选择修改密码, 输入新密码 2222	修改成功	同预期
	关闭密码	在有密码的状态下点击“密码”设置, 输入原密码 1111, 选择关闭密码	关闭成功	同预期
导出物品信息到 Excel 表格	点击该设置项	选择“导出物品信息到 Excel”设置项, 打开提示 Dialog, 点击确定按钮	物品信息成功导出到设备根目录下, 以“导出物品表”+当前时间戳.xls 命名	同预期
联系开发者	点击设置项	点击“联系开发者”设置项	打开系统邮件, 自动根据模板创建新邮件	同预期
评分	点击设置项	点击“联系开发者”设置项	打开系统应用商店中本应用的界面	同预期

### 6.3 软件测试结论

根据上述测试流程和实际结果, 本 App 的测试结论如表 6-6 所示。

表 6-6 软件系统测试结论

功能名称	所属模块	预期结果	测试结果
登录、注册	用户模块	登录、注册正常, 程序初始化正常, 界面显示正常	同预期
物品搜索	概览模块	搜索结果准确, 显示效果正常, 界面切换正常	同预期
账号管理	概览模块	显示、复制、编辑账号正常, 显示效果正常	同预期
切换住所	概览模块、空间模块、打包模块	切换住所成功, 显示效果正常, 界面切换正常	同预期
物品增删改查	概览模块、空间模块、打包模块	操作成功, 界面实时同步刷新, 功能正常	同预期
住所、房间、家具、容器、列表管理	设置模块	功能正常	同预期
密码	设置模块	功能正常	同预期

#### 6.3.1 软件能力

依据本物品管理 App 测试的结果, 本 App 在 Android 8 以及更新的 Android 平台上可以正常使用。本 App 可以实现管理物品信息、在物品处于过期和临期状态时进行

提示等功能。

### 6.3.2 缺陷和限制

考虑到开发者自身的掌握情况和对 App 使用场景的分析, App 使用了 SQLite。由于本系统使用了 Material Design 的主题, 各个界面风格较为统一, 界面简约又不失美观, 但个性化的风格较少, 部分界面还有待添加自定义过渡动画, 界面过渡效果还有较大程度的提高。还应该进一步优化内存和功能。

## 结 论

在本论文中主要讨论了日常生活中物品浪费的问题，并且简单的阐述了解决问题的思路，并在此问题的基础上开发了基于 Android 平台的物品管理 App。本文细致的介绍了物品管理 App 的实现所需的软件、语言和技术，详细说明了系统的具体需求、模块层次设计和各个模块的实现过程。

该系统实现了管理住所、房间、家具和物品等信息、在物品处于过期和临期状态时进行提示、将物品打包到容器、导出物品信息到 Excel 文件、应用锁定、每日定时后台更新数据库等功能。在功能实现上，使用了 Service、Broadcast Receiver、Content Provider、Fragment、SQLite 数据库、RecyclerView 及其对应的适配器。虽然本系统基本上实现了预定的计划和目标，但经过用户使用后的反馈，无论是从功能上还是从界面上，还有较大空间进行改进。因此，在本系统已实现的基础上，还有以下几方面的提升方向。

- (1) 创建应用的服务器，使应用能实现云备份和云恢复的功能；
- (2) 增加从表格中导入物品信息的功能；
- (3) 可以使用新技术、框架和语言对系统进行升级，如 Flutter、Jetpack Compose、Kotlin；
- (4) 将本 App 在各大安应用商店上架；
- (5) 开发其他平台的版本。



## 参 考 文 献

- 1 欧阳燊.Android Studio 开发实战：从零基础到 App 上线. 北京：清华大学出版社，2018
- 2 David J Parker.Mastering Data Visualization with Microsoft Visio Professional 2016.Birmingham: Packt Publishing, 2016
- 4 Don Norman.The Design of Everyday Things.New York: Basic Books, 2013
- 3 陈晓历. 新印象——中文版 Sketch 图标与 UI 界面设计实例教程. 北京：人民邮电出版社，2020
- 5 Tidwell Jenifer.Designing Interfaces: Patterns for Effective Interaction Design.Sebastopol: O'Reilly Media, 2020
- 6 柳伟卫. Java 核心编程. 北京：清华大学出版社，2020
- 7 安辉.Android App 开发从入门到精通框. 北京：清华大学出版社，2018
- 8 欧阳燊.Android Studio 开发实战：从零基础到 App 上线（第 2 版）. 北京：清华大学出版社，2018
- 9 何红辉.Android 开发进阶：从小工到专家. 北京：人民邮电出版社，2016
- 10 郭霖. 第一行代码——Android（第 3 版）. 北京：人民邮电出版社，2020
- 11 张锡英，李林辉，边继龙. 数据库系统原理（第 2 版）. 哈尔滨：哈尔滨工业大学出版社，2016
- 12 Nenad Jukic, Susan Vrbsky, Svetlozar Nestorov.Database Systems: Introduction to Databases and Data Warehouses.Burlington: Prospect Press, 2016
- 13 李刚. 疯狂 Android 讲义. 北京：清华大学出版社，2019
- 14 杨怀洲. 软件测试技术. 北京：电子工业出版社，2017
- 15 郑人杰，马素霞. 软件工程概论（第 3 版）. 北京：机械工业出版社，2020

## 致 谢

行文至此，落笔为终。转眼间已经到了毕业设计作品的最后一步，大学四年的学习生涯即将结束，始于 2017 年秋，终于 2021 年夏，目之所及，皆是回忆。

有师如斯，庆幸之至，感谢我的论文指导老师韩老师，亦师亦友，本作品的顺利完成，离不开老师的耐心指导。韩老师认真的悉心教导和博学的专业知识，令我十分欣赏。在此也要感谢我本科四年所有教育我的老师们，是你们教会了我专业技能和人生的智慧。祝愿老师们工作顺利、平安幸福。

感谢家人对我无微不至的照顾。父母对我的养育之恩，只有通过自身的不断努力，成为他们的骄傲才能略作报答。

毕业设计对我来说，既是一件挑战，也是一份收获，我对 Android 技术的了解和认识加深了许多，为我将来走向社会进行工作打下了一定程度的基础，为我的大学生涯画下一个句号。

送君千里，终有一别。感谢所有的相遇，山高路远，我们更高处相见！