G08小组

“城院跳蚤”微信小程序

总体设计报告



摘要

本总体设计文档旨在为二手交易小程序的开发提供全面的架构规划和技术指导，明确系统的整体结构、功能模块划分、数据存储方式以及系统交互流程等关键方面。其主要读者包括项目开发团队成员，如软件工程师、测试人员、产品经理等，以便他们能够依据此文档进行后续的详细设计、编码实现、测试以及维护工作。

目录

[1 引言 2](#_Toc182159978)

[1.1 编写目的 2](#_Toc182159979)

[1.2 背景 2](#_Toc182159980)

[1.3 定义 2](#_Toc182159981)

[1.4 参考资料 3](#_Toc182159982)

[2 总体设计 3](#_Toc182159983)

[2.1 需求概述 3](#_Toc182159984)

[2.2 运行环境 4](#_Toc182159985)

[2.2.1 硬件环境 4](#_Toc182159986)

[2.2.2 软件环境 4](#_Toc182159987)

[2.3 基本设计概念和处理流程 5](#_Toc182159988)

[2.3.1 系统架构 5](#_Toc182159989)

[2.3.2 处理流程 6](#_Toc182159990)

[2.4 结构设计 7](#_Toc182159991)

[2.4.1 功能模块结构 7](#_Toc182159992)

[2.4.2 模块关系 8](#_Toc182159993)

[2.3基本设计概念和处理流程 8](#_Toc182159994)

[2.6人工处理过程 9](#_Toc182159995)

[2.7尚未解决的问题 9](#_Toc182159996)

[3接口设计 9](#_Toc182159997)

[3.1用户接口 9](#_Toc182159998)

[3.2外部接口 9](#_Toc182159999)

[3.3内部接口 9](#_Toc182160000)

[4运行设计 10](#_Toc182160001)

[4.1运行模块组合 10](#_Toc182160002)

[4.2运行控制 10](#_Toc182160003)

[4.3运行时间 10](#_Toc182160004)

[5系统数据结构设计 10](#_Toc182160005)

[5.1逻辑结构设计要点 10](#_Toc182160006)

[5.2物理结构设计要点 11](#_Toc182160007)

[6.出错处理和恢复 11](#_Toc182160008)

[6.1可能出现的错误及处理方式： 11](#_Toc182160009)

# 1 引言

## 1.1 编写目的

本总体设计文档旨在为二手交易小程序的开发提供全面的架构规划和技术指导，明确系统的整体结构、功能模块划分、数据存储方式以及系统交互流程等关键方面。其主要读者包括项目开发团队成员，如软件工程师、测试人员、产品经理等，以便他们能够依据此文档进行后续的详细设计、编码实现、测试以及维护工作。

## 1.2 背景

随着人们消费观念的转变和环保意识的增强，二手交易市场日益繁荣。本二手交易小程序应运而生，旨在为用户提供一个便捷、高效、安全的二手物品交易平台。该小程序由G08小组负责开发，预期将满足广大用户对于闲置物品再利用和获取性价比高的二手商品的需求。

## 1.3 定义

**二手物品**：曾经被使用过，目前处于闲置状态且仍具有一定使用价值，可供在平台上进行交易转让的物品。

**用户**：在小程序中注册并参与二手物品交易活动的个人，包括出售物品的卖家和购买物品的买家。

**订单**：买家与卖家之间达成交易意向后生成的记录，包含交易物品信息、价格、买卖双方信息、交易状态等详细内容。

**微信小程序：**小程序的一种，是一种不需要下载安装即可使用的应用，它实现了应用“触手可及”的梦想，用户扫一扫或搜一下即可打开应用。

## 1.4 参考资料

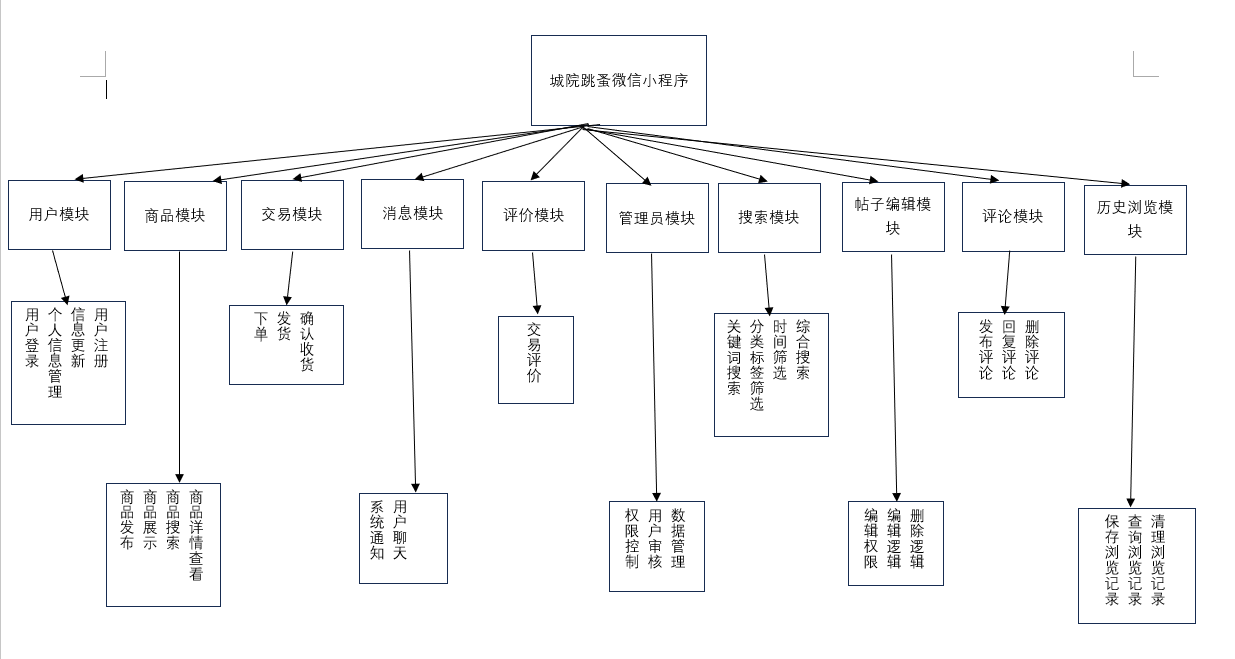
《软件工程 - 实践者的研究方法》（Roger Pressman 著），用于指导软件开发过程中的规范和方法。

微信小程序官方开发文档，提供小程序开发的技术框架、接口规范等重要参考信息。

相关数据库管理系统（如 MySQL、MongoDB 等）的技术文档，辅助设计数据存储结构和操作方式。

# 2 总体设计

## 2.1 需求概述

通过对目标用户群体的调研和分析，确定二手交易小程序应具备以下核心功能：  


**用户注册与登录**：支持多种注册方式，方便用户快速创建账号并登录系统，确保用户身份的唯一性和安全性。

**物品发布与管理**：卖家能够详细发布二手物品信息，包括物品名称、类别、品牌、成色、价格、描述和图片等，并可对已发布物品进行编辑、下架等操作。

**物品浏览与搜索**：用户可通过多种方式浏览和查找感兴趣的二手物品，如分类浏览、关键词搜索、热门推荐等，快速定位目标物品。

**交易沟通**：提供买卖双方直接沟通的渠道，方便双方就物品细节、价格、交易方式等进行协商，促进交易顺利达成。

**订单处理**：支持买家下单购买物品，卖家确认，双方完成交易后显示订单等一系列流程操作，同时提供订单状态查询和跟踪功能。

**个人中心**：展示用户个人信息、交易记录、收藏物品等，方便用户管理自己的交易活动和关注感兴趣的物品。

**信用评价体系**：建立完善的信用评价机制，激励用户诚信交易，提高平台交易质量和可信度。

## 2.2 运行环境

### 2.2.1 硬件环境

**服务器端：**

处理器：Intel Xeon E5 - 2620 v4 及以上，具备多核心和高频率，以应对大量用户请求和数据处理。

内存：32GB 及以上，确保系统在高负载下能够高效运行，缓存常用数据。

硬盘：1TB 及以上的企业级硬盘，用于存储用户数据、物品信息、交易记录等大量数据，采用 RAID 技术保障数据可靠性。

网络带宽：100Mbps 及以上的独享带宽，保证数据传输的快速和稳定，满足用户频繁的数据交互需求。

**用户端：**

智能手机或平板电脑，具备常见的移动设备配置，如处理器性能良好（如骁龙 600 系列及以上、麒麟 700 系列及以上），内存 1GB 及以上，屏幕分辨率能够清晰显示小程序界面，支持触摸操作，具备网络连接功能（Wi - Fi 或移动数据网络）。

### 2.2.2 软件环境

**服务器端：**

操作系统：Linux（如 Ubuntu Server 18.04 LTS），具有高度稳定性和安全性，适合部署后端服务。

数据库管理系统：MySQL 8.0 或 MongoDB 4.0，用于存储和管理结构化和非结构化数据，满足不同类型数据的存储需求。

后端框架：Spring Boot（Java）或 Django（Python），提供高效的后端开发框架，简化开发流程，提高系统性能和可扩展性。

应用服务器：Tomcat 9.0（与 Spring Boot 配合）或 uWSGI（与 Django 配合），负责处理 HTTP 请求，将动态页面内容返回给用户端。

消息队列：Redis，用于处理异步任务（如消息推送、订单状态更新通知等），提高系统响应速度和并发处理能力。

**用户端：**

微信客户端：7.0 及以上版本，作为小程序的运行容器，提供基础的运行环境和接口支持。

微信小程序基础库：2.0 及以上版本，确保小程序能够使用最新的功能和特性，保证良好的用户体验。

## 2.3 基本设计概念和处理流程

### 2.3.1 系统架构

采用分层架构模式，分为用户层、表现层、业务逻辑层、数据访问层和数据层。

**用户层**：用户通过微信客户端访问小程序，与系统进行交互。

**表现层**：负责展示小程序的用户界面，接收用户输入并将处理结果呈现给用户，采用微信小程序原生技术（WXML、WXSS、JavaScript）实现。

**业务逻辑层**：处理系统的核心业务逻辑，如用户注册登录、物品发布与管理、交易流程处理、信用评价计算等，通过后端编程语言（Java 或 Python）实现。

**数据访问层：**提供对数据层的访问接口，负责与数据库进行交互，执行数据的增删改查操作，采用相应的数据库访问技术（如 JDBC、PyMongo）。

**数据层：**存储系统的所有数据，包括用户数据、物品数据、订单数据、评价数据等，由数据库管理系统（MySQL 或 MongoDB）进行管理。

### 2.3.2 处理流程

**用户注册登录流程：**用户在小程序端输入注册信息（如手机号、密码、验证码等），点击注册按钮后，数据通过网络传输到服务器端。服务器端对输入数据进行合法性验证，若验证通过，则将用户信息存储到数据库中，并返回注册成功信息给小程序端，同时生成用户登录凭证。用户登录时，输入账号密码，服务器端验证凭证，验证通过后允许用户登录系统，获取用户相关信息并展示在个人中心页面。

**物品发布流程：**卖家登录后，在发布物品页面填写物品详细信息，上传图片，点击发布按钮。小程序端将物品数据上传至服务器端，业务逻辑层对数据进行处理，包括图片存储路径处理、数据格式校验等。校验通过后，数据访问层将物品信息插入到数据库中的物品信息表，发布成功后返回成功提示给卖家。

**物品浏览与搜索流程：**用户在小程序首页或搜索页面进行操作，选择分类浏览或输入关键词搜索。表现层将用户请求发送至业务逻辑层，业务逻辑层根据请求从数据访问层获取相应的物品数据（从数据库中查询符合条件的物品记录），并将数据返回给表现层进行展示。在浏览物品列表时，用户可点击物品卡片查看详情，业务逻辑层根据物品 ID 从数据库获取详细信息并返回给表现层展示在详情页。

**交易沟通流程：**买家在物品详情页留下评论后点击“我想要”按钮。此时，卖家可以收到通知，并可以选择是否交易，系统建立买卖双方的聊天通道，进入私聊界面，双方发送的消息通过消息队列（Redis）进行中转，实时推送至对方小程序端，实现即时通信。聊天记录存储在数据库中，以便用户查看历史聊天内容。

**订单处理流程：**买家与卖家协商一致后，买家在物品详情页点击 “立即购买” 按钮，填写交易时间、交易地址等信息，提交订单。小程序端将订单数据发送至服务器端，业务逻辑层进行订单处理，包括生成订单号等操作。处理完成后，数据访问层将订单信息插入订单表，并更新相关数据（如物品状态、用户交易记录等）。卖家在个人中心查看订单，确认交货信息，双方线下交易后更新订单状态为 “已发货”，买家收到货物后，在订单详情页点击 “确认收货”，订单状态更新为 “已完成”。

## 2.4 结构设计

### 2.4.1 功能模块结构

**用户管理模块**：负责用户注册、登录、信息修改、密码找回等功能，确保用户账号安全和信息准确。

物品管理模块：实现物品发布、编辑、下架、查看等操作，对物品信息进行全面管理，包括图片处理和数据存储。

**搜索与推荐模块**：提供物品搜索功能（基于关键词、类别等）和个性化推荐（根据用户浏览历史、收藏记录等），帮助用户快速找到感兴趣的物品。

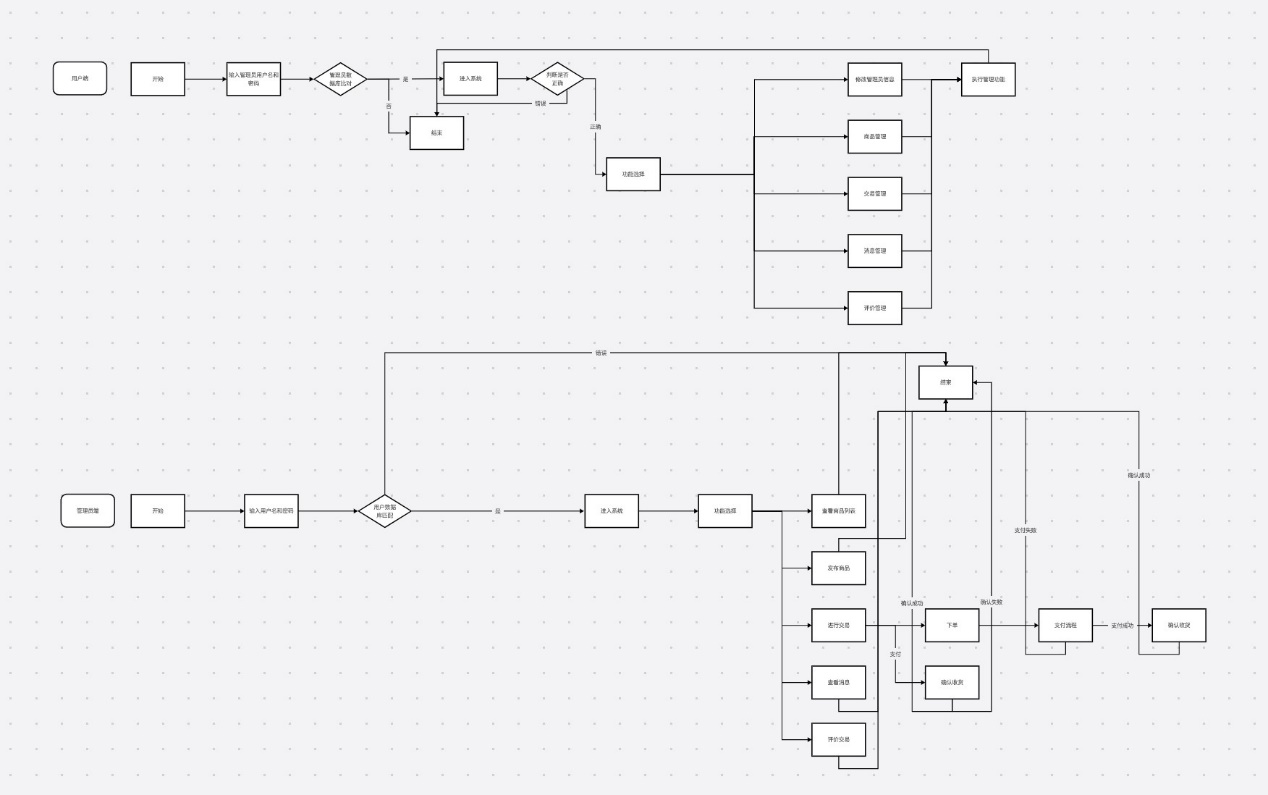
**沟通模块**：支持买卖双方聊天，实时消息推送，保证交易沟通顺畅，包括评论记录管理和消息提醒功能。

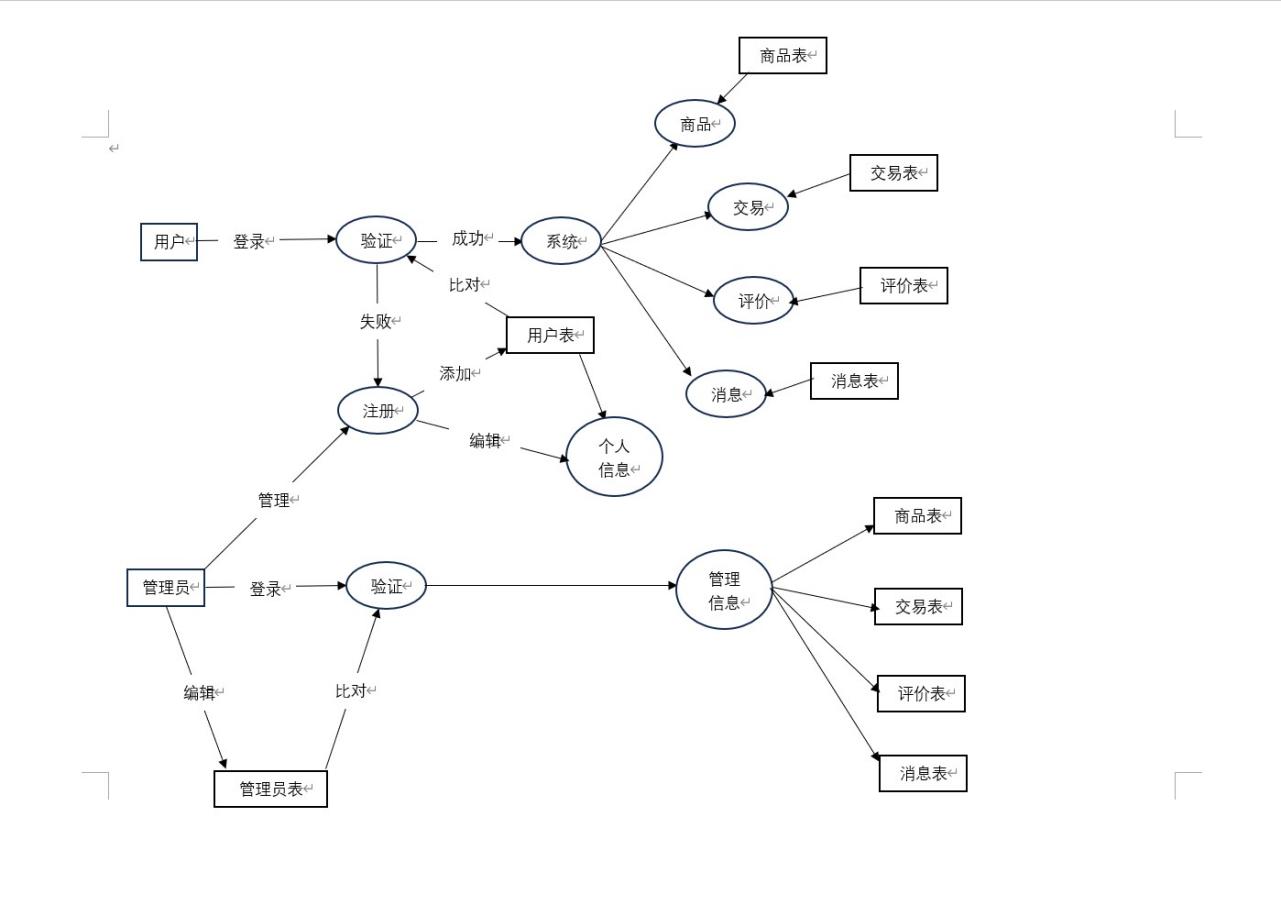
个人中心模块：展示用户个人信息、交易记录、收藏物品、信用评价等，方便用户管理自身交易活动和查看个人相关信息。

### 2.4.2 模块关系

用户管理模块为整个系统提供用户身份验证和信息管理基础，其他模块在用户登录后根据用户权限和操作需求调用相关功能。物品管理模块与交易管理模块紧密相关，物品发布后可在交易流程中被购买和处理。搜索与推荐模块依赖物品管理模块的数据，为用户提供物品查找服务。沟通模块在交易过程中被频繁使用，与交易管理模块协同工作，促进交易双方沟通。个人中心模块整合了用户在系统中的各项信息和操作记录，与其他模块都有一定的数据交互。信用评价模块在交易完成后由交易管理模块触发，评价结果影响用户在系统中的信誉度，进而影响其他模块的操作（如搜索排名、交易权限等）。

## 2.3基本设计概念和处理流程

2.4结构



# 3接口设计

## 3.1用户接口

我们将向用户提供简单的筛选推荐命令，开放模型接口。

## 3.2外部接口

标签系统与数据库mysql之间的接口，商家信息与美团等开放接口

## 3.3内部接口

内部拥有推荐系统与使用pytorch训练的模型之间的接口，

# 4运行设计

## 4.1运行模块组合

共分为三个模块，

1.用户模块，存储注册用户的个人信息，如头像、账号密码收藏等。

2.推荐模块，为用户提供近期帖子推送

3.标签模块，为帖子打上对应标签

登录时用到用户模块，首页推荐帖子用到推荐模块与标签模块，点进帖子查看详情与评价用到标签模块，个人信息及收藏用到用户模块与标签模块

## 4.2运行控制

**用户登录**：输入账号密码即可登录，点击注册输入对应信 息即可注册用户

**首页论坛**：展示别人帖子，上拉刷新

**商品页**：浏览加盟校园商家

**个人中心**：方便用户对自身交易活动进行管理。

## 4.3运行时间

响应时间：在一般网络环境下，操作响应时间通常不超过 1 秒，确保用户操作流畅，如点击按钮、切换页面等操作能快速得到反馈。

# 5系统数据结构设计

## 5.1逻辑结构设计要点

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 字段说明 |
| 用户信息表 | |  |
| user\_id | 字符串 | c |
| username | 字符串 | 用户名 |
| password | 字符串 | 用户密码 |
| 发帖信息表 | |  |
| post\_id | 字符串 | 帖子唯一标识 ID |
| user\_id | 字符串 | 发帖用户 ID |
| title | 字符串 | 帖子标题 |
| content | 文本 | 帖子内容 |
| post\_time | 时间 | 发帖时间 |
| 回贴信息表 | |  |
| reply\_id | 字符串 | 回帖唯一标识 ID |
| user\_id | 字符串 | 用户 ID |
| content | 字符串 | 回帖内容 |
| reply\_date | 日期 | 回帖日期 |
| 收藏信息表 | |  |
| fid | 整型 | 收藏id |
| user\_id | 字符串 | 用户id |
| post\_id | 字符串 | 帖子id |
| 历史信息表 | |  |
| num | 整型 | 唯一标识每条历史记录 |
| date | 日期 | 操作发生的日期 |
| user\_id | 用户编号 | 用户唯一标识 ID |

## 5.2物理结构设计要点

本系统内所使用的每个数据结构中的每个数据项都储存在SQL数据库中

# 6.出错处理和恢复

## 6.1可能出现的错误及处理方式：

**网络连接问题**：若网络不稳定或中断，小程序会提示 “网络连接异常，请检查网络设置”，用户可检查手机网络连接（如 Wi-Fi 是否正常、移动数据是否开启）后点击重试。

输入错误：如价格格式错误（输入非数字或小数点位置不正确）、必填项未填写等，系统会在相应位置显示红色错误提示，用户需根据提示修改输入内容后重新提交。

**服务器繁忙**：在高并发情况下，服务器可能繁忙导致操作响应缓慢或失败，小程序会提示 “服务器繁忙，请稍后重试”，用户可稍等片刻后再次操作。

**恢复措施**：对于因网络问题导致的操作中断，网络恢复后用户可重新进行操作，若遇到服务器故障等严重问题，平台会及时进行修复，并在修复后通知用户，用户无需特殊操作，等待恢复正常后可继续使用小程序。