
青 岛 科 技 大 学

本 科 毕 业 设 计

题 目 基于微信小程序的校园图书交易平台

设计与实现

指导教师 庄国涛

辅导教师 庄国涛

学生姓名 郎吉豪

学生学号 1806030208

数理 学院 数学与应用数学 专业 应数 182 班

2022 年 4 月 25 日

基于微信小程序的校园图书交易平台

设计与实现

摘 要

随着互联网的进步和人们节约环保消费理念的普及，市面上在线闲置物品交易平台蓬勃发展，但高校二手图书这一特定交易场景下仍存在不平衡不充分的问题。本设计以青岛科技大学为例，基于微信小程序平台，使用云开发、Python 等技术，致力建立一个可落地的校园二手书交易平台，以期优化校园二手图书流通领域有效供给，激发消费潜力的释放，响应国家“双碳目标”。

关键词：二手图书；小程序；Python；云开发；Node.js

THE DESIGN AND IMPLEMENTATION OF THE CAMPUS BOOK TRADING PLATFORM BASED ON WECHAT MINIPROGRAM

ABSTRACT

With the progress of the Internet and the popularization of people's consumption concept of saving environmental protection, online idle items trading platform in the market is booming, but there are still imbalance and insufficient problems in the particular trading scenario of second-hand books in colleges and universities. This design takes Qingdao University of Science and Technology as an example, based on wechat miniprogram platform, using Cloud-base development, Python and other technologies, is committed to the establishment of a practical campus second-hand book trading platform, in order to optimize the effective supply of second-hand books on campus circulation field, stimulate the release of consumption potential, in response to the national "carbon peaking and carbon neutrality goals".

KEY WORDS: used books, miniprogram, Python, cloud-base development, Node.js

目 录

前言	1
1 绪论	2
1.1 高校二手书市场特点	2
1.2 同类产品分析	2
1.3 相关技术	2
1.3.1 小程序	2
1.3.2 云开发	3
1.3.3 云函数	3
1.3.4 云数据库	3
1.3.5 Python	3
1.3.6 Go-cqhttp	3
2 需求分析与可行性分析	4
2.1 功能需求分析	4
2.1.1 用户需求分析	4
2.1.2 运维人员需求分析	4
2.2 可行性分析	4
2.2.1 技术可行性	4
2.2.2 效益可行性	5
2.2.3 运营可行性	5
3 系统设计与开发	6
3.1 系统设计目标	6
3.2 总体架构设计	6
3.3 模块功能设计	7
3.3.1 登录、认证与信息维护模块	7
3.3.2 发布与管理模块	8
3.3.3 浏览与搜索模块	10
3.3.4 信息详情页模块	12
3.3.5 辅助交易模块	13
3.3.6 外源信息整合与管理模块	14
3.3.7 后台管理模块	14
3.4 数据库设计	14
3.4.1 概念结构设计	14
3.4.2 各表数据项说明	15
3.5 开发工具与开发环境	16
总结	17
参考文献	18
附录	19

致谢	20
----------	----

前言

随着互联网的进步和人们节约环保消费理念的普及，各类在线二手交易平台蓬勃发展，人们可以便捷地出售自己的闲置物品，也可以从世界各地的用户手中买到合适的二手商品。但此类平台大而全的同时降低了信息分发和物流运输的效率，限制了高校学生校园内二手商品交易这种限制地域场景下的交易效率。加之受新冠疫情影响，传统在线二手交易场景下的个人卖家寄快递消毒不规范、包裹跨地区流动、校园易成为管控区等弊病也日益凸显。同时，校园二手交易场景还有用户画像单一、商品种类固定的特点，图书是其中最高频的交易商品，虽然二手商品是传统的非标准商品，但图书品类相对保存良好、参数固定、潜在卖家和买家画像固定，天然具备可自然流转、复用价值高的属性。

然而在现实校园中，书籍重复利用率低的现象依然十分严重，曾经的知识载体论斤售卖，令人心痛。随着国家“碳达峰、碳中和”双碳目标的提出，怎样优化高校场景下图书交易体验、提升高校二手图书的循环利用率、构建绿色低碳校园显得尤为重要。

为了进一步减少物流成本及其过程中的病毒传播风险、提高校园场景下的交易效率，本项目旨在开发一款针对校园二手图书交易场景的在线平台。

1 绪论

1.1 高校二手书市场特点

由于院校学科划分、培养方案设定的高度相似，高等院校场景下的二手书需求和供应存在趋同的现象，因此图书在闲置物品交易中还具备相对保存良好、参数固定、潜在卖家和买家画像固定的特性。在理想情况下，高年级学生的专业课教材可以自然地像低年级学生流转，各类考研、考公、考证书籍也可以重复利用。因此在本校区范围内优化资源配置、提升交易效率是解决校园二手书交易难题的关键。

1.2 同类产品分析

1. 闲鱼、转转等综合性在线二手交易平台

阿里巴巴旗下的闲鱼是国内最大的在线闲置用品交易平台，背靠阿里系的交易信息和用户信息，使用者不仅可以手动上传商品，还可以直接导入自己在淘宝购买过的商品，同时可以对接芝麻信用等用户信息保证交易安全。但就校园二手书这一特定场景特定领域的交易而言，闲鱼在大而全的同时未能做到小而美，如没有针对书籍品类做出快速发布通道，没能有效协调区域内二手书供需等等，这也是一众在线综合性二手交易平台的通病。

2. 孔夫子旧书网、多抓鱼等书籍品类交易平台

孔夫子旧书网、多抓鱼等专业书籍品类交易平台，通过扫描 ISBN 码进行发布等方法优化了图书的发布流程，但仍未有效协调区域内书籍供需的问题，也没有充分满足大学生这一特定用户群体的需求。

3. ‘互助群’、‘表白墙’等校内论坛

QQ、微信是当代大学生最主要的社交平台和关系链沉淀池，得益于平台充分的流量，高校学子大部分闲置书籍交易都通过其生态下衍生出的‘表白墙’、‘互助群’等校园媒体进行。但这些途径无法对交易信息汇总整合，也缺少必要的、标准化的发布引导过程，同时也存在这覆盖面有限、交互体验差、人员构成复杂、安全性低等问题。

综合以上分析，平台应结合以上平台优点，基于微信这一天然的大学生社交平台、模仿扫码发布等使用流程，采取实名认证等建立互信的方案；同时要通过互补的方式尽力克服彼此的一些弊端。

1.3 相关技术

1.3.1 小程序

微信小程序是当下最为流行的连接用户与服务的方式，它可以在微信平台内被便捷地

获取和传播，同时具有出色的使用体验。^[1]得益于其运行在微信这一当下最主流的即时通讯平台下，小程序具有免安装、跨平台、易于社交传播的特点，我们可以很方便的调用用户信息、位置、登录、微信支付等在线交易必要的接口，高效地完成符合用户习惯的前端开发。综合以上，平台采取前端小程序后端云开发的技术方案。

1.3.2 云开发

云开发是腾讯云推出的一整套后端方案，包含云函数、云数据库、云存储和云托管，复用微信私有协议及链路，使开发者无需关注登录鉴权、服务器及运行环境搭建、数据接口、数据库安全、并发扩容等繁琐项目，可以专注于代码逻辑本身，提升开发效率。同时，按量付费的特性及相当量的免费额度可减少前期投入，降低项目整体运营成本，考虑到后续将平台推广到其他学校，云开发整体打包也可以降低部署的难度，数据库部署在自己独立的云环境上也可以保障运营者的数据主权。

1.3.3 云函数

云函数是一种基于云原生的无服务器执行环境，使开发者在无服务器的情况下运行代码，只需使用 Node.js 语言编写核心代码并设置代码运行的条件，即可。SCF 是实时文件处理和数据处理等场景下理想的计算平台。

1.3.4 云数据库

微信云数据库可以看作是简化版的 MongoDB，是一种 JSON 型数据库，相较于 MySQL 等关系型数据库，JSON 型数据库无 schema 结构，上传数据时字段随意定义和增减，甚至同一个字段的数据类型都可以不一样，特别适合字段类型多变的业务。二手商品作为非标品，其参数项并不固定，为了增加用户自由度，使用户可以根据商品和需求情况选择性的填写信息，本项目适于采用云数据库。

1.3.5 Python

Python 是近年来大热的一门编程开发语言，具有简洁易读、拓展性强等优秀的特性。Flask 是一个使用 Python 编程语言所开发的轻量 Web 网络应用框架，可用于接收处理 post、get 等多种网络请求并返回处理结果。本项目中的“消息雷达”模块正是通过基于 Python 的 Flask 网络框架所制作的 API 来接收和处理 QQ 群内交易信息，并通过 Regex 正则表达式对信息进行筛选和提炼而实现的。同时，Python 众多的爬虫相关包与成熟的机器学习框架也为项目进一步整理聚合更多交易信息及智能化分类筛选信息提供了拓展空间。

1.3.6 Go-cqhttp

Go-cqhttp 是一款无 GUI、支持消息事件回调上报的开源 QQ 机器人。本项目采用该软件进行高校二手交易群内信息的采集。

2 需求分析与可行性分析

2.1 功能需求分析

2.1.1 用户需求分析

- (1) 用户能够利用微信实现注册、登录、修改用户信息等操作；
- (2) 用户能够方便快捷地通过扫码、拍照、文字输入等方式发布图书及购买需求；
- (3) 用户可以在线对图书和需求进行浏览，以选择自己感兴趣的内容，联系发布者进行交易；
- (4) 用户可以将自己感兴趣的图书或需求添加到收藏中；
- (5) 用户可以对平台内的图书和需求进行检索，快速定位自己感兴趣的内容；
- (6) 可以对自己发布的内容进行删改查管；
- (7) 用户可以为自己发布的商品或需求生成一张图片格式的海报，便于在朋友圈等平台进行分享；
- (8) 用户可以对同一本图书的多个发布进行比价，可以查看包含某一特定标签的商品集合，可以查看某个用户发布的全部信息；
- (9) 用户可以按照自己的实际需求，选择仅浏览本校区的商品或者屏蔽掉求购信息。

2.1.2 运维人员需求分析

- (1) 运维人员可以对用户信息进行检索、修改、删除、拉黑等操作；
- (2) 可以对商品信息进行删除、修改、审核、检索等操作；
- (3) 可以对首页 banner 及跳转链接进行编辑、替换、增加等操作。

2.2 可行性分析

2.2.1 技术可行性

微信小程序平台自 2016 年 9 月推出以来便收获了众多关注，微信官方提供了详尽的开发文档和易用的开发工具，并且封装好了微信相关接口的调用方案，提供了云开发等多种开发运维支持，近七年的发展和维护使其成为一个功能强大、系统成熟的应用平台，小程序在中国拥有海量开发者，也诞生了相当多体验优秀的产品。

开发者已充分了解小程序开发的基本流程，基本熟悉 wxml、Javascript、wxss、wxs、Python、Node.js 语言的语法规则及特性，学习了云开发、数据库、正则表达式、接口规范等相关知识。参考了大量的应用范例，总体来讲本项目在技术上完全可行。

2.2.2 效益可行性

除开发人力成本外，项目运营成本域名注册及 ICP 认证开支等初始开支，供于程序运行的云资源开销（包含 CDN 流量、云函数资源使用量、云储存空间容量等）、图书 ISBN 查询 API 接口按量付费及其他用于运维的开销。总体而言成本完全可控。

盈利方面，虽然中介是二手买卖中常见的盈利模式，但我们设计的平台会免费向用户提供商品展示和交易撮合的服务。但校园图书交易有其固有的周期性：卖书的高峰期往往在学期末、毕业时等节点，此时为买方市场，平台可以低价收购，同时也为卖方提供了保底服务；而在学期初等时间节点则形成了求购高峰期的买方市场，平台可以适当提升价格出售存货获利。此外还可采用广告收入、CPS 收入等等在线平台常见的收益模式，完全可覆盖运营成本，具有商业上的可行性。^[2]

除经济效益外，平台还有相当的生态效益：图书背后的造纸、印刷、装订、包装、运输行业均拍放了大量的二氧化碳，据北京绿交所数据，1kg 的纸制品会产生约 3.5kg 的二氧化碳。通过推动二手交易来延长图书使用周期、提升综合利用率，可大大减少在这一系列过程中不必要的碳足迹。

2.2.3 运营可行性

从各类校园论坛信息统计和用户调研中看，校园图书交易仍存在着不平衡不充分的问题，有大量的持有闲置图书的同学因缺少高效便捷的信息发布渠道而无法出手，同时很多有需求的同学因为找不到安全可靠的书源而只能去高价购买新书，本程序能够较好的整合分散的交易信息，提供高效的撮合交易功能，可以有效满足市场需求。微信是当代大学生必备的 APP 之一，微信小程序即点即用的特点免除了用户下载 APP 的苦恼，微信一键登录的特性也免除了用户记忆大量账号密码的麻烦，极大降低了用户的使用成本，同时程序可以在用户的社交关系链上进行传播，在短时间内达到较好的用户增长的效果。

3 系统设计与开发

3.1 系统设计目标

根据需求分析和可行性分析，系统应达到的设计目标有：

(1) 安全性

小程序运行在微信的沙箱环境内，由行业顶尖的安全专家提供安全保障，因此主要关心程序自身逻辑和数据库权限带来的安全风险。

(2) 真实性

作为交易信息分发平台，信息的真实性与时效性及用户的可信性是促成交易的关键。

(3) 高并发

云开发的动态扩容特性既保证了在高峰期能满足大量用户同时访问的需求，又能在平稳期减少资源使用。

(4) 易用性

在 UI 设计和交互设计上参照主流成熟的交易软件，遵循用户习惯，减少学习和适应成本，同时在用户体验流程关键部分予以图示指引，保证系统易用性。

(5) 可维护

系统采用小程序原生 MINA 框架，实现了页面独立、视图层与逻辑层分离，各个层次模块之间耦合度低，同时得益于云环境共用、HTTPAPI 接口等特性，拓展开发和业务迁移方便，整体维护难度不大。

3.2 总体架构设计

微信小程序可以看作是一种特殊的 B/S 架构，其中 Browser 端是微信客户端；Server 端的功能则由云函数和云数据库承担。采用三层结构设计方案，这是当前主流的前端开发方案，即将程序划分为三层独立的层级，包括用户表示层（wxml、wxss）、业务逻辑层（Javascript、Node.js、wxs）与数据访问层（数据库操作接口、云数据库）。用户使用小程序进行相关操作时，用户表示层将用户所触发的事件传递到业务逻辑层进行相关处理，随后逻辑层会把加工处理后的数据变化反馈到用户表示层，从而触发用户表示层的数据变化或页面更新。微信小程序的开发基于 MINA 框架，MINA 框架的主体文件是位于根目录下的 App.js、App.json 和 App.wxss。^[4] App.js 文件是程序的入口文件，用来描述小程序的全局逻辑，存放全局变量。App.json 文件是全局配置文件，包含 pages、window、tabbar、component、permission 等基本属性，分别用于配置页面、窗口样式、导航栏、组件、运行时权限等。App.wxss 文件是全局样式文件，用于规定所有页面都可用的样式效果。页面文件保存在 pages 文件夹中，都由独立 wxml 文件、wxss 文件、js 文件以及 json 文件构成，可以独立的对每个单独的页面进行开发和设计。

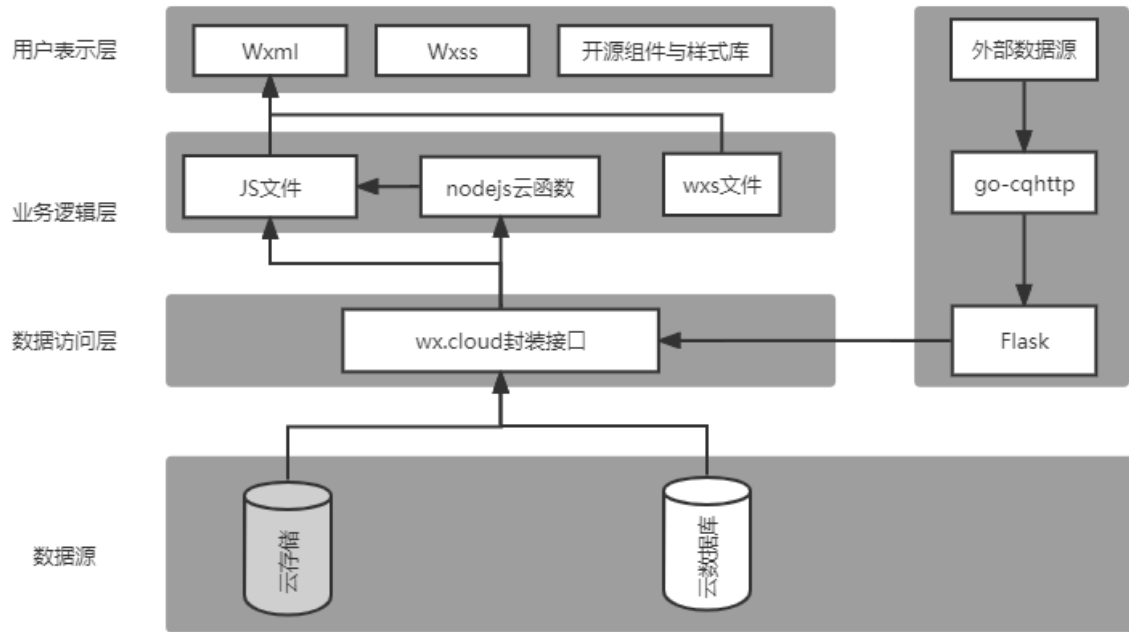


图 3-1 系统架构示意图

Fig. 3-1architecture diagram

3.3 模块功能设计

系统的整体功能模块划分如下图所示：

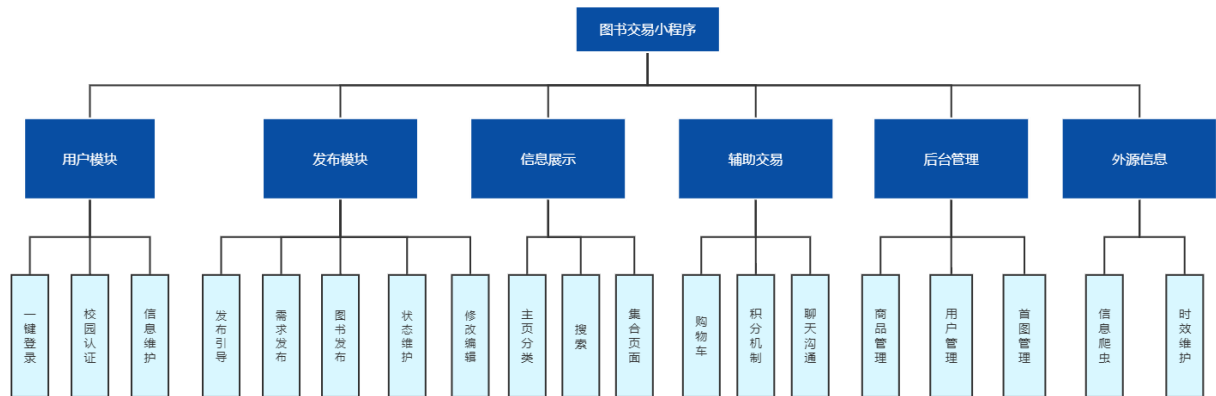


图 3-2 功能模块图

Fig. 3-2 Function modules diagram

3.3.1 登录、认证与信息维护模块

该部分主要包括登录页、认证页、个人页、信息修改页以及其他页面下登录检测模

块。

登录页用于获取用户头像、昵称等必要信息及位置、存储等保证程序功能正常运行必须的权限。随后根据用户信息在用户数据库中的有无引导完成登录、认证等操作。

实现登录主要是在登录页面加载时通过调用 `login` 云函数获取用户的 `openid` 来实现一个无感知的身份认证，但仍需用户点击登录按钮调用官方 `getUserProfile()` 接口来获取用户头像、昵称等必要的身份信息。小程序保持登录状态主要有写入缓存和写入全局变量两种方案，写入缓存方案的优点在于在其他页面需要使用用户信息时，一次读入赋值便可使用大量信息，考虑到本项目中包含用户信息较多且用户退出再登录为低频操作，本项目采用写入缓存的解决方案，即登录完成后会调用官方 `wx.setStorageSync`（字段名，字段值）方法将用户信息写入小程序的缓存，其他页面根据缓存中是否有用户信息来判定用户的登录状态，退出登录则也只需将缓存中的用户信息重新设为空值。^[3]

认证采用校园邮箱验证的方式，目前仅适配了青岛科技大学的本科学生，用户输入学号后前端会调用后台 `sendcode` 云函数并传送学号参数，云函数会利用 `smtp` 协议向该学号对应的校园邮箱发送由五位随机数字组成的验证码，并将该验证码返回前端，在前端完成对用户学生身份的校验，若用户输入的验证码与后台返回的验证码值相匹配，则进行后续操作，否则提示用户重新输入。

个人页包含背景图片、个人信息、发布信息、客服帮助等区域组成。背景根据用户校区不同匹配不同的校园风景图片，增强用户信任与身份认同；发布信息包含想买、想卖两个用户在平台发布信息的集合入口，用户可以快捷方便的管理自己所发布的商品与求购信息。

在信息修改页用户可以修改联系方式、头像、昵称、校区等信息。

其他页面下登录检测模块通过 `wx.getStorageSync()` 方法来获取对应的缓存信息来检测缓存中是否包含用户信息来判断用户此时登录状态，由此对页面进行不同的渲染，引导用户进行不同的操作逻辑，避免一些难以控制的 `bug` 发生。

3.3.2 发布与管理模块

发布模块由发布引导页、图书发布页及需求发布页三个主要页面构成。

发布引导页固定在底栏 `tabbar` 一级入口上，采用图示的方法给用户清晰易懂的操作说明。在逻辑层，引导页需要获取用户的登录状态，若用户已经登录，则应在视图层展示导航到图书和需求两个发布界面的按钮，使用户根据自己的需要进行下一步操作；否则应展示导航到登录界面的按钮，引导用户先登录再进行后续操作。

图书发布页采取类似微信朋友圈的发布方式，尊重用户的使用习惯，在用户界面上由主要信息填写区、图片上传与预览区及价格、取货方式和联系方式等相关信息补充区组成。部分非必要信息输入框处于默认隐藏的状态，降低用户心理成本。校区信息、联系方式等信息默认从缓存中的用户信息中获取，用户也可以对其进行修改。在用户填写完必要信息后可以点击标签选项，唤起一个半屏弹窗组件 `tagarea`，该组件可以根据用户

输入的发布信息中提取关键词、根据用户身份信息获取学院、专业等标签以及二手交易常见的新旧程度描述词等作为待选标签，用户可以点击这些标签添加到商品信息中，也可以自行输入添加。在用户点击发布按钮之后，系统会弹出联系方式确认的模态框，用户可根据自己的需要选择或修改自己的多种联系方式，并同时验证信息输入是否符合规范，在用户点击确认且信息符合规范后会进行上传图片、添加图书信息到数据库的操作，完成信息的发布。发布完成后跳转到成功页，成功页有三个功能按钮，用户可以查看所发布图书的详情，也可以为当前发布的商品生成一张图片海报供于分享传播，还可以初始化发布状态再次发布另一本图书。

扫码发布使用户可以通过扫描书籍封底条形码的方式获取 13 位的 ISBN 码，经验证后以该 ISBN 码为参数调用 book 云函数，云函数首先会尝试在 book 数据表中查询有无该本书的出版信息，有则直接返回图书信息，若查询结果为空，那么会向书籍查询 API 接口请求数据，随后将 API 得到的信息返回给前端，同时将信息存入 book 数据表，避免重复调用 API 接口带来的额外资费，也可以提升查询效率。前端根据云函数返回值自动完善标题与图片等信息。云数据库 book 数据表中的图书数据爬取自“中教图书”网站 (<http://jcxzwsx.com>)，有以高校教材为主、包含 ISBN 号、标题、作者、出版社、摘要等信息的 9000 余条数据，经笔者以本专业所用教材为例测试，在库率达到 80% 以上，能较好满足程序需求。书籍查询使用“极速数据”(<http://www.jisuAPI.com/API/ISBN/>)提供的 API 接口。^[5]

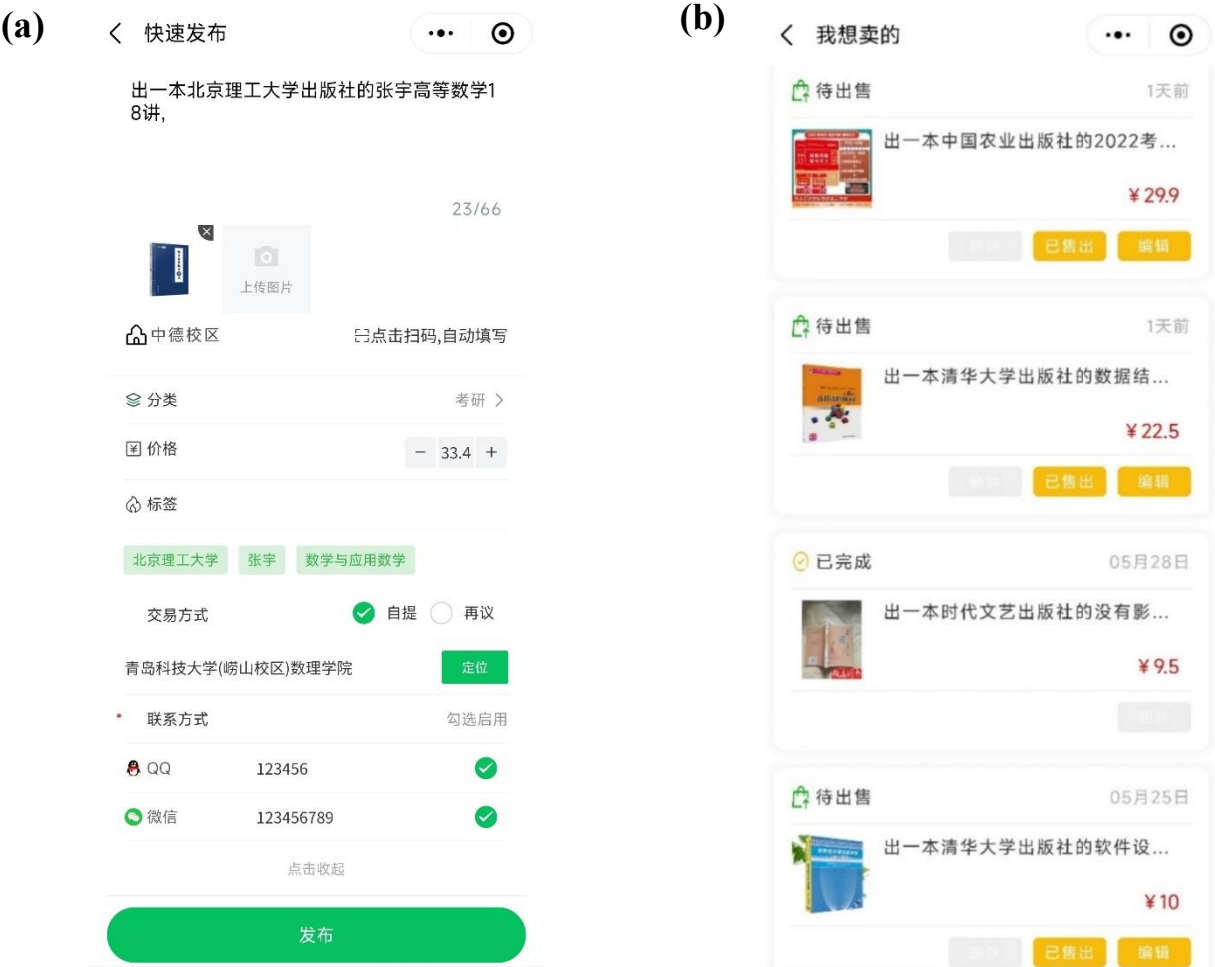


图 3-3(a) 发布页示例图(b) 信息管理页示例图

Fig. 1-1 schematic illustration for the user interface of (a) publish page and (b) management page

求购发布页在功能、操作、逻辑上同发布页基本一致，只是删除了不必要的信息项，在引导方式上进行了一定调整。

发布信息管理模块包括商品列表页、求购列表页及信息修改页三个页面组成。

商品列表页将自己所发布的信息汇集一处，并对商品的不同销售状态做出直观标注，分为在售、已售出等状态，便于集中进行管理，每个商品均包含擦亮、下架、删除等快捷操作及导往编辑页面的快捷入口。

求购列表页在功能逻辑上类似于商品列表页，在显示字段上做适合求购信息的调整。编辑页可以对自己发布的信息进行修改。

3.3.3 浏览与搜索模块

信息组织模块的目的是为了辅助用户快速的找到自己想要的商品信息，提升交易率。分为主页、搜索、集合页三个板块。三个板块基于类似的模板，均通过对数据库查询语

句及页数、排序方式的修改，调用 `querylist` 云函数，获取对应的商品信息数组，并传递到商品浏览区组件进行渲染；均有回顶部的快捷悬浮按钮以及触底刷新机制。

商品浏览区封装为一个组件，可供于主页面、搜索页、集合页等进行调用，入参为一个包含发布信息的数组，组件会渲染数组中的信息。组件采用两种可以供用户选择的布局方式：朋友圈式和瀑布流式，前者可以灵活地渲染各种不同的信息，提供丰富的信息展示；后者是目前电商类程序的主流，有更直观的视觉冲击力，在同一屏幕下可以展示更多商品信息。用户可以根据自己的浏览习惯，在组件左上角进行切换两种布局。同时提供“仅看本校区”、“不看求购”两个筛选按钮，可快速排除不符合条件的信息项，增加浏览效率。组件化的封装与复用大大减少了不必要的重复开发，便于后续维护，也使得用户拥有统一的用户体验。

主页面主要由搜索栏、**banner** 栏、分类栏、筛选区和商品浏览区组成。其中搜索栏、分类栏通过 `vant` 组件和一些 `wxss` 样式属性控制，始终悬浮在屏幕顶部，方便用户随时进行切换和跳转。搜索栏仅作为一个跳转链接区，具体搜索功能由单独的搜索页面承担；**banner** 栏用于展示广告轮播图，可由运维人员在数据库中更换轮播图 and 对应链接；分类栏使得用户可以通过点触或滑动的方式切换分类，修改对应的数据库查询语句，得到不同的商品信息，传递到浏览区进行渲染。



图 3-4(a) 主页示例图(b) 集合页示例图

Fig. 1-1 schematic illustration for the user interface of (a) index page and (b) collection page

搜索页面可由主页顶部的搜索栏标识进行跳转，可根据用户输入的关键词，在数据库中检索出符合条件的商品或需求信息；还提供了历史搜索的功能，便于用户复查商品。搜索功能通过在数据库查询语句内使用正则表达式进行关键词判断，进而从数据库中选出符合要求的信息；历史搜索功能会在每次用户每次搜索时记录关键词并写入缓存来实现，同时具有去重功能，保证同一关键词多次搜索不会重复记录，用户在搜索界面点击历史搜索词即可快速复查，同时为保护用户隐私，提供一键清除历史记录的功能。

集合页分为按照同款图书整合、按照标签整合与按照发布者整合三种状态，使用户可以按照不同的专题检索商品，便于进行货比三家、打包团购等操作。

3.3.4 信息详情页模块

详情页包含用户信息区、商品信息区、联系方式区、更多发布区、同款商品区和底部

操作栏，各区域根据商品信息的不同及用户登录状态的不同进行有选择的渲染。联系方式区提供了 QQ、微信聊天方式的一键复制及站内私聊的入口；更多发布区仅针对发布了多个商品的平台实名用户所发布的信息开放，便于浏览者打包购买其他商品；同款商品区仅针对用户发布的包含 ISBN 的商品，便于浏览者进行比价选优。底部操作栏根据浏览者是否为发布者提供编辑或加入购物车两种操作按钮。

3.3.5 辅助交易模块

包括买卖双方在线聊天、积分奖励、购物车模块、海报生成模块。

好的沟通是二手交易成功的关键，除了可以方便的交换联系方式外，系统还提供了站内聊天功能，用户可以在详情页或者购物车中快速联系上发布者，咨询自己想知道的更多信息。当用户开启聊天时，会调用 chat 云函数，生成一个包含双方昵称、头像及当前商品、需求信息等字段的聊天信息表，方便在双方的聊天列表页进行渲染。在主界面 tabbar 也提供了聊天列表的一级入口，便于用户在多个聊天窗口之间切换。

消息的实时推送与获取渲染是即时通讯工具必备的功能，本项目封装了一个用于实现该功能的组件，在组件内调用通过云数据库 watch 监听操作实现的监听器来实现新消息的自动获取与渲染，基本上符合主流 IM 工具的界面效果和操作习惯。

区别于 QQ、微信等其他 IM 工具，本项目在聊天页内不仅提供文字和图片的即时通讯，还应提供常用语快速发送功能，使用户可以更快发送二手交易语境下的常用语。此外在聊天页最上方提供一个商品卡片展示当前所咨询商品的简略信息及出售状态，并可以快速跳转到详情页，方便用户随时查看所关心的商品信息。

碳减量、碳积分是目前国内软件厂商最常用的对用户绿色行为的激励机制，二手书交易具有天然的环保属性，因此也应引入碳积分机制。当用户完成交易时，应当给予一定数量的碳积分奖励，本项目中碳积分核算模型来自于北京绿色交易所碳足迹计算器数据，按照平均每本书重 200g、每 1kg 纸制品的生产、制备、运输产生 3.5kg 二氧化碳进行粗略核算，用户每完成一次交易，可减少约 700g 碳排放，平台通过统计商品、需求数据库内为该用户所发布的且 status 字段为 2 的条目数量，便可得出用户总减碳量，用户总减碳量即为小程序用户的碳积分，可用后续其他营销活动的开展。

购物车模块是电商平台必备的功能之一，使买方可以持续关注特定商品状态，对可选的同类商品进行比较，给用户做出购买决策的时间。作为有买卖双方所发布信息的二手交易平台，其购物车设计更像一种收藏的功能，可以加入买卖双方的发布信息，便于随时建立联系。

Painter 是一款优秀的前端海报生成工具，通过 json 字段即可控制海报渲染，小程序 palette 文件夹下配置了渲染所需的模板文件并开放为一个调用接口，在发布成功页等页面引入后可直接传参调用，生成一个包含用户昵称、发布昵称、程序二维码的海报，方便用户进行分享传播。

3.3.6 外源信息整合与管理模块

作为一个交易信息平台，拥有足够庞大的信息量是吸引用户的关键，尤其在初创时期，应当使平台有一定量真实的交易信息，因此应当对其他平台已经存在的交易信息进行爬取上架，一方面扩充平台内容，另一方面也可以发展潜在用户。

本项目“信息雷达”利用 go-cqhttp 软件模拟登陆 QQ 并获取校内二手交易群内的消息，并将消息通过 post 方法上报到电脑本地搭建的 flask 服务框架上，flask 框架通过判断消息是否来自指定的群聊、是否包含特定的关键词来进行初筛，再通过正则表达式对消息进行图文分离等进一步的清洗，随后将信息上传到云数据库中，实现内容的扩充。

但对于爬取来的商品没有对应发布者进行维护，容易造成信息更新不及时、状态时效滞后的问题，因此应当增加一个定时管理模块，当商品发布一段时间以后，自动对商品进行下架处理，保证平台内信息的时效性。

时效管理通过 clear 云函数实现，该云函数在运行时会对数据库中所有的采集而来的交易信息进行判断，当信息的创建时间 creat 字段内的时间戳小于当前时间戳超过 1 天时，会将对应的 status 字段更新为 2，即为下架处理，其他用户将不会刷到该信息，减少了用户的注意力浪费。并且为该函数创建一个定时触发器 time-trigger，使云函数在每小时的指定时间进行运行，实现自动时效管理。

3.3.7 后台管理模块

云数据库和云开发为开发者提供了 CMS 内容管理平台，在云开发控制台开通内容管理，配置管理员帐户并导入用户数据库、商品数据库等表单并建立字段映射模型后便可以对内容进行直观的增删改查等常用操作，完成了后台管理所需的用户管理、商品、需求信息管理等操作，实现了平台后台管理的需求。^[6]

3.4 数据库设计

3.4.1 概念结构设计

为实现模块功能设计中的各项功能，系统主要存在商品与需求信息、购物车记录、聊天列表和标准图书等实体，一个用户可以发布多条商品与需求信息，一个用户可以加购多个商品，一个商品也可以被加入多个用户的购物车，一本书可能对应多个商品，多个商品可能对应一本书，每个聊天列表信息对应两个人，但每个人都可以创建多个聊天，各实体间的关系如图 3 所示。

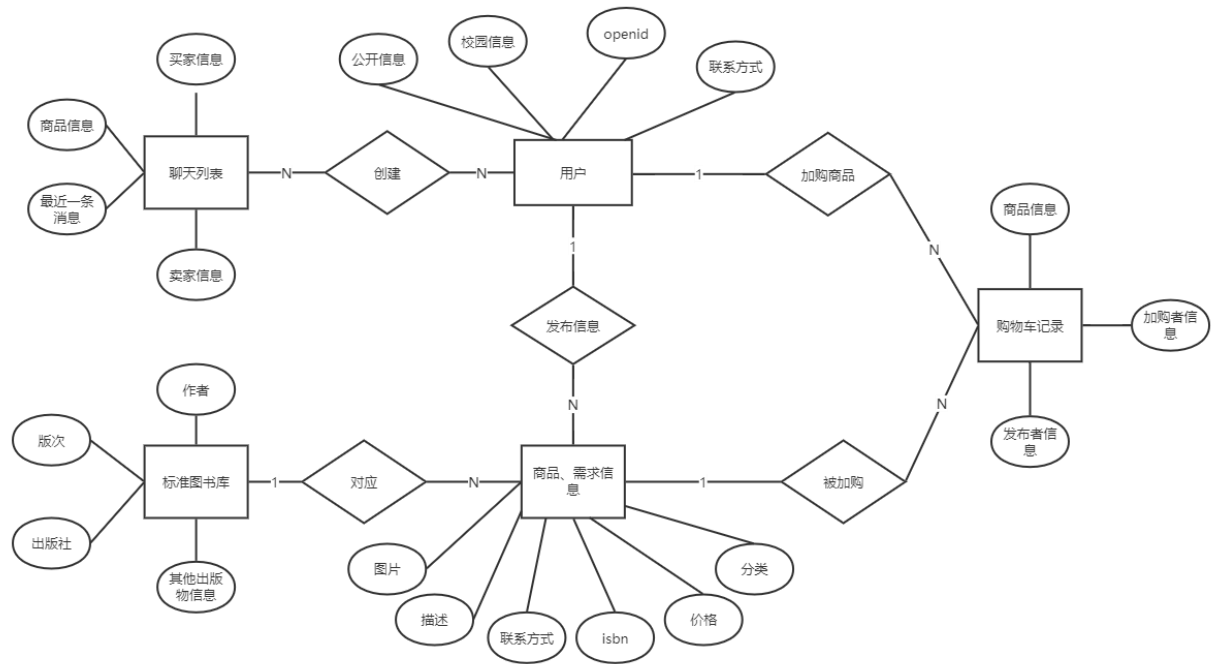


图 3-5 简化数据库 E-R 图

Fig. 3-5 Database E-R diagram

除图所示实体关系之外，还有 banner 轮播图配置信息等辅助实体，用于辅助运维人员的一些操作，与主体业务关联不大。

3.4.2 各表数据项说明

表 3-1 用户信息表

Tab. 3-1 The Table of User Info

字段名称	数据类型	说明
_openid	String	微信用户标识
nickName	String	用户昵称
avatarUrl	String	用户头像
QQ	String	用户 QQ 号码
WX	String	用户微信号码
Campus	Enum	用户校区
Uid	Int	用户的学工号
School	String	用户所在学院
Major	String	用户专业
_id	String	数据库索引

表 3-2 图书、需求信息表

Tab. 3-2 The Table of publish information

字段名称	数据类型	说明
_openid	String	发布者 openid
Type	Enum	信息类型
Isbn	Int	图书 ISBN 号
_id	String	数据库索引
status	String	信息状态
method	Enum	交易方式
Title	String	内容
piclist	Array	图片
Campus	String	所在校区
Place	String	具体位置
Avatar	String	发布者头像
Nickname	String	发布者姓名

表 3-2 购物车记录表

Tab. 3-2 The Table of cart information

字段名称	数据类型	说明
_openid	String	加购者 openid
Itemid	Int	图书 ISBN 号
_id	String	数据库索引
status	String	信息状态
Title	String	内容
piclist	Array	图片
Avatar	String	发布者头像
Nickname	String	发布者姓名
pubopenid	String	发布者 openid

表 3-1 至表 3-3 中所标明的数据字段基本已经覆盖其余表项中所出现的字段。

3.5 开发工具与开发环境

- 1.开发工具：微信开发者工具 RC1.05.2203251、Vscode1.66.2、APIPost
- 2.运行环境：微信基础库 2.22.0、python3.9.7、Node.js16.13.0
- 3.支撑软件：go-cqhttp

总结

本项目围绕提升校园二手图书交易效率这一中心需求，仿照其他的一些二手交易平台，设计并实现了基本的用户的注册与登录、商品需求的发布与管理、信息的浏览与检索及其他辅助的站内聊天、购物车等功能，遵循了主流软件的设计规范并设计了简单易懂的引导流程，使用户易于上手、体验优秀。

同时在复现并融合主流二手平台功能的基础上，平台也有相当的创新功能和理念，如高效汇集外源信息，有效解决了平台初期内容少、获客难的运营难题；标签机制，优化了商品的匹配逻辑；海报生成，便于用户的社交传播等等，这些功能创新针对性的解决前期调研中发现的痛点，有助于优化整体流程，提升交易效率。

但由于毕业设计时间有限、笔者能力尚有欠缺以及微信给予开发者的权限有限等客观原因，程序仍有一些不足之处，如无权调用微信支付接口、界面不够美观、代码有一定冗余等等，现在已经将项目在 `github` 和 `gitee` 平台开源，后续笔者自己也将持续维护这一项目，并尽量将其投入实战运营，在市场实践的检验中与老师的指导下，不断迭代优化。

参考文献

- [1] 李建文. 基于微服务的个性化购物微信小程序设计与实现[D].华中科技大学,2020.
- [2] 胡思思,马诗豪,姜美.绿色校园背景下的校园二手交易平台[J].现代商业,2021(23).
- [3] 赵壮. 推荐算法在校园二手交易平台中的研究与应用[D].武汉轻工大学,2021.
- [4] 赵明,贲祺舒,吕怡.基于微信小程序的高校校园二手物品交易平台的设计与实现[J].科技与创新,2022(07):55-58.
- [5] 廖倩倩.基于微信小程序与云开发的发票检查系统实现[J].数字通信世界,2022(05):29-30+33.
- [6] 徐志杰,许忠维,吴昊.基于微信小程序的圈层式校园墙设计与实现[J].电脑知识与技术, 2022.0633.
- [7] 宋莺,谢小玲,余文婷,周亚辉,张怡敏.智慧校园微信小程序云服务开发与构建[J].长江信息通信,2022,35(04):174-176.
- [8] 杜雨荃,王晓菊,田立勤.基于微信小程序的网上购物系统的设计与实现[J].网络安全技术与应用,2022(04):60-62.
- [9] 微信官方文档[EB/OL].<https://developers.weixin.qq.com/miniprogram/dev/framework/>.
- [10] X. Sun and W. Zhou, "Study on the Web Site of Reach-Me-Down Bargaining Based on Campus Network," 2010 International Conference on Internet Technology and Applications, 2010.
- [11] Yang, Xiaohan. Campus Second-hand Trading Application Design and Implementation. Diss. California State University, Northridge, 2018.

附录

致谢

毕设已毕，感激不尽。

首先要感谢我的导师——庄国涛老师，在毕设的选题到完成的全过程中，对本设计的方向和架构进行了总体的把握和指导，并细致耐心地对毕设可能的商业化落地提出了关键性的指导和帮助。

感谢所参阅的众多技术博客与文档使我在踩到坑里时仍有安全感；感谢文晓港的 colorUI 样式库和有赞团队的 vant 组件库，让一个审美严重拉跨的理工男减少了对样式的设计的抓狂。

感谢程序的所有内测用户给出的批评与建议。

感谢在青科的四年，以及所有帮助过我的老师和同学们。