

# 中国大学生计算机设计大赛



## 软件开发类作品文档简要要求

作品编号: 2024011248

作品名称: 校园小程序

作 者: 李金钊 张汇琼 蔡佳怡 黄东玲 马欣怡

版本编号: V2.5.3

填写日期: 2024.04.01

## 目 录

第一章 需求分析 .....	3
第二章 概要设计 .....	4
第三章 详细设计 .....	5
第四章 测试报告 .....	13
第五章 安装及使用 .....	14
第六章 项目总结 .....	14
参考文献.....	16

# 第一章 需求分析

## 1.1 背景与目的

本小程序旨在构建一个卓越的校园服务平台，为师生提供便捷性、安全性和多样性，从而实现校园生活的完美互动与畅通交流。对标作品主要是有料同学、GCC 百宝箱等一些主要面向学生群体的优秀校园助手程序。

## 1.2 主要功能

特别说明:以下功能需在个人设置上进行学生认证(可以通过上传学生证或者输入学号以及密码进行验证)，学校官方账号会分配一个专属的管理员账号,多个学校数据不共享。

### 1.2.1 信息发布

官方公告：管理员发布学校重要通知和公告，师生可以点赞、关注及评论，以便学校更好地了解学生的动态。

动态专区：师生可自由发布、点赞、评论校内动态，分享校园生活。

### 1.2.2 学生服务

二手闲置：学生可以发布二手物品信息，实现资源再利用（该平台采用学生证认证，以提高学生交易的安全性。详细信息请参阅架构设计）。

任务中心：用户可以自由发布校园活动、兼职等信息，促进勤工俭学，解决师生困难。

课程表：已绑定教务系统的学生，可通过小程序查看课表信息。

### 1.2.3 系统配套功能

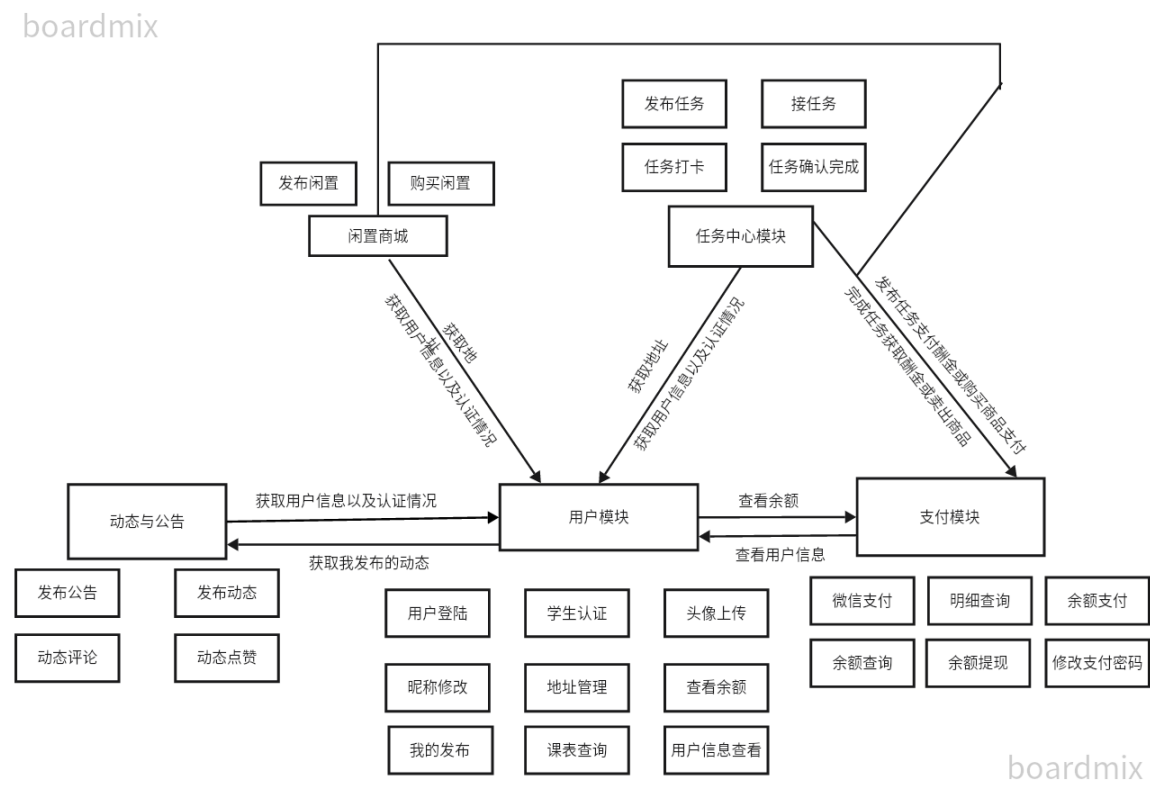
余额管理：小程序支持微信支付、收益提现、余额详情查询等。

地址管理：小程序可以保存常用地址，方便用户在任务中心或二手闲置中使用。

个人设置：用户可以设置昵称、头像、支付密码以及学生认证等。

# 第二章 概要设计

## 2.2 业务架构图



### 2.2.1 业务架构设计

注意： 由于程序架构设计过于复杂，图的空间有限，模块间的全部接口调用请看第三章的 api 设计。

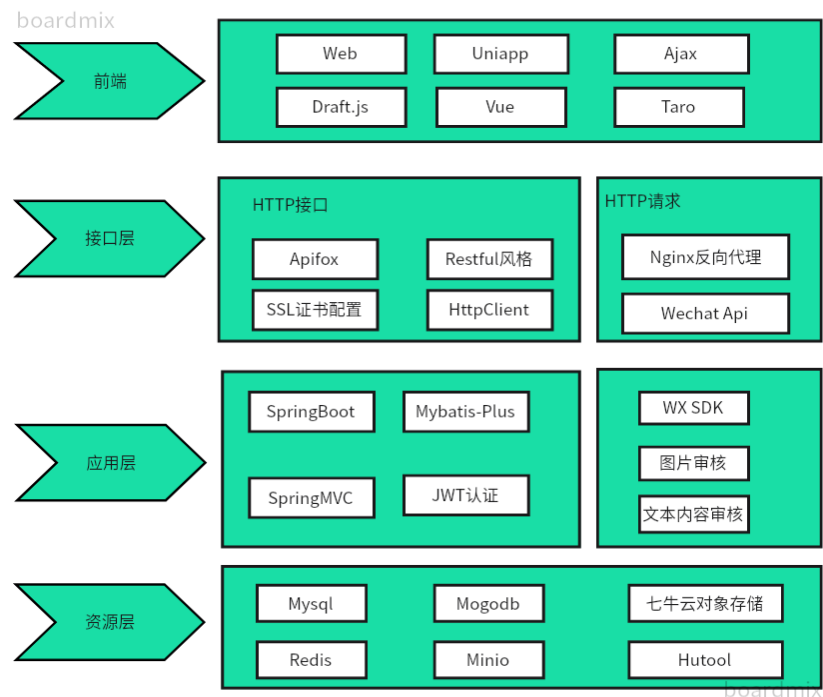
1. 动态与公告模块
2. 任务中心模块
3. 支付模块
4. 用户模块
5. 闲置商城模块

该程序业务层主要分为五大模块，部分模块调用接口较多，比如用户模块, 支付模块等，具体的接口详情在 api 文档均有记录。

# 第三章 详细设计

## 3.1 应用架构设计

应用架构图：



这个是开发该小程序主要用到的技术框架：

1. 数据库的重要数据均已进行加密，比如支付密码使用的是安全级别较高的 Bcrypt 加密，同时配置了 SSL 证书，支持 Https，安全性得到了保障；
2. 存储数据使用了主流的数据库，Mysql 作为存储关系型较强的信息，Redis 用于保存缓存信息。由于发布时间较新的动态信息、部分用户的登陆状态信息此类信息需要频繁读取，小程序的前端使用懒加载，提高了一定的性能；为了图片的响应速度得到保障，使用了七牛云的对象存储以及 CDN 加速，同时使用了 Minio 对象存储；
3. 由于本项目是基于小程序的开发，为此使用了 HttpClient 调用微信的 API 进行业务需求的书写，并且封装了一个 SDK 给项目调用，该 SDK 发布了在远程仓库；
4. 该小程序配置了后台管理系统(Web 端)，使用 Nginx 反向代理进行系统的部署，同时配置了 SSL 证书；
5. 该项目是前后端分离项目，前端使用了 Ajax 进行发送请求，后端接口文档使用 Apifox，类似于 Swagger 产品；
6. 后端基于 SpringBoot 进行项目的开发，使用基 ORM(Object Relational Mapping)的框架 Mybatis-Plus 同时使用 JWT 作为身份认证，提高了安全性以及性能(JWT 的好处是减少了服务器的压力，同时 Token 解密需要密钥)。

## 3.2 界面设计

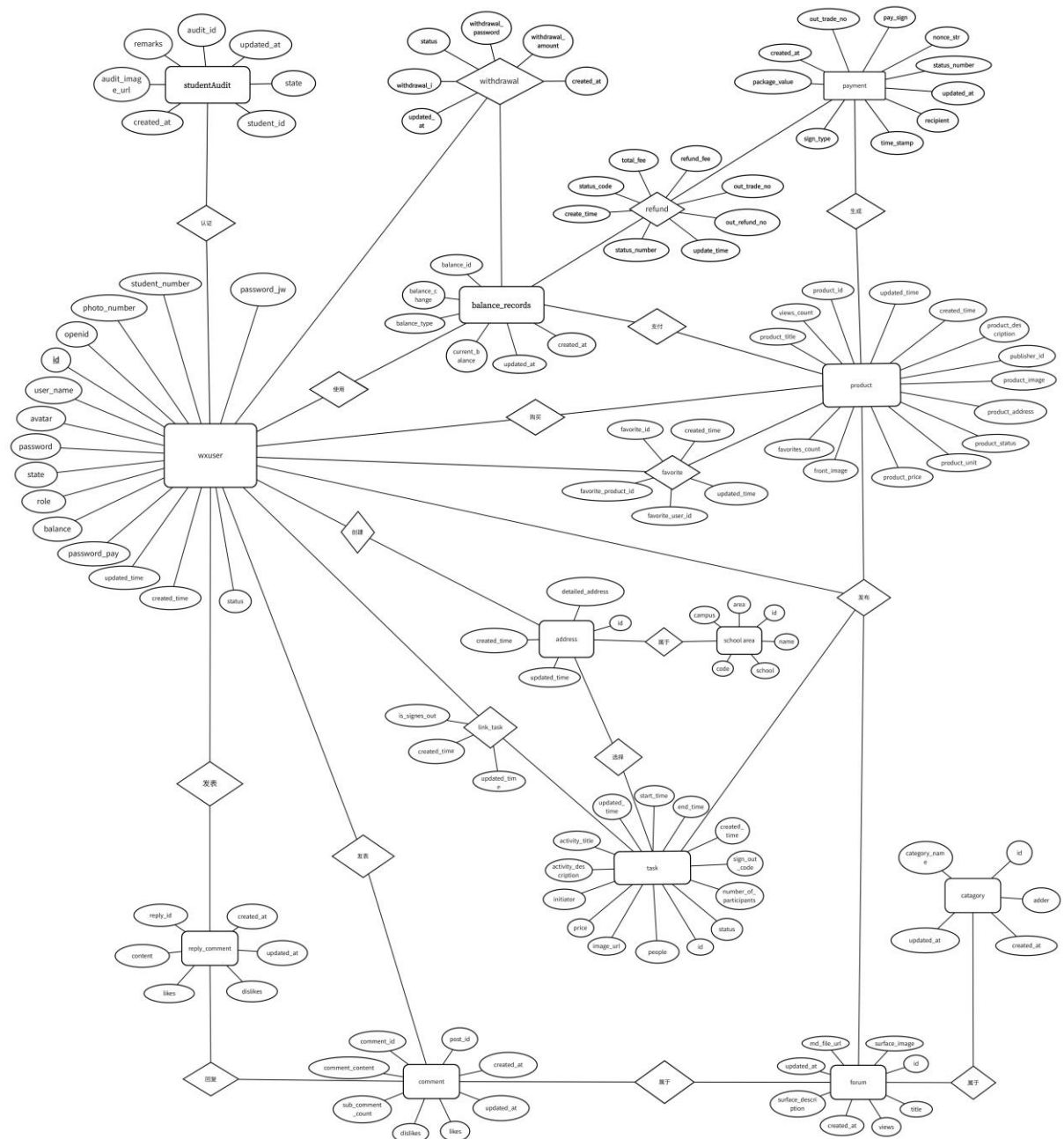


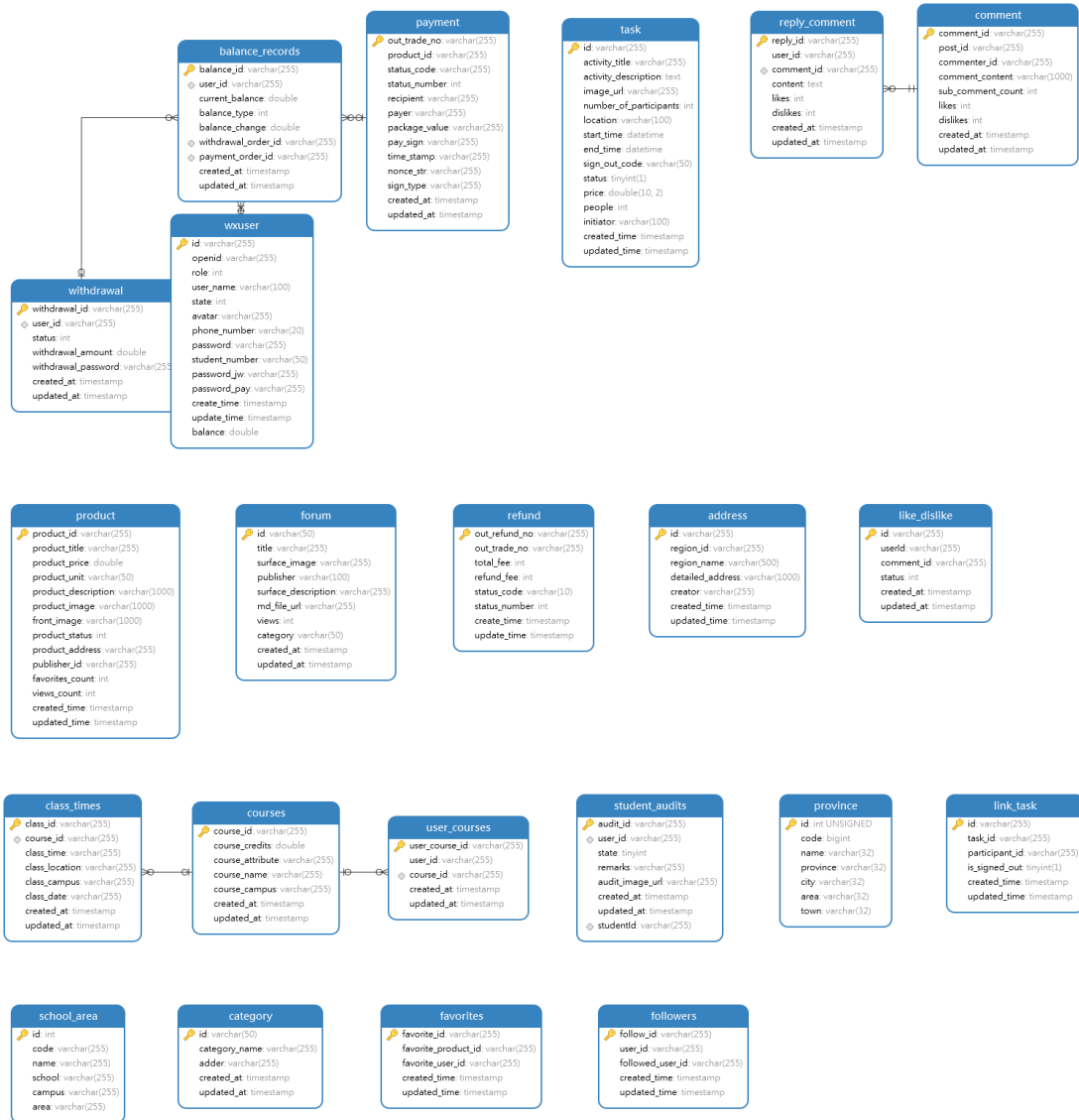




### 3.3 数据库设计







注意：

任务表的 statu: (0 开始接单（已经支付了），1 是结束（钱已经付款给接单者）2 是已经被接单了，3 是未支付，算是暂时保存了（是已经创建了订单了）5: 是没有调用微信订单，6 是申请退款的（没有被接单的时候，也就是 0 的时候），7 取消订单（已经被接单了，是接单的取消）然后设置为 0，所以 7 可能不存在，因为取消订单不需要发布者同意）//8`接单者确定送达（数字 1 是发布者按下确认送达后打款）（正常顺序是 530281）

user 表的 statu:\* 状态，正常是 0（默认（未认证）正常（0）、封禁（2）、注销（3）、已认证（1），5 是上传了证件照认证的（在 1 的基础上））

详细的请看本目录的 SqlER. pdf 以及 Sql 中文字段. pdf 文档

## 3.4 关键技术

### 3.4.1 如何保障支付的安全性

支付安全是任何支付系统的核心，主要涉及数据库安全和支付身份验证两个方面。程序采取了以下方案进行提高安全性：

1. 严格控制数据库访问权限：本程序数据库采用多用户，不同权限配置；
2. 定期进行数据库备份：服务器配置了 Shell 脚本定期备份；
3. 程序的支付密码设置为独立的，并使用 Bcrypt 进行密码的加密保存。

### 3.4.2 如何防止订单被恶意拍下拒绝付款以及卡订单发布导致其他用户拍不了该订单

1. 程序对接了微信支付的查询订单情况 API 以及微信支付回调 API，用户支付成功后，微信服务器访问本程序指定的接口，实现回调功能；
2. 用户下单时，程序设置定时器并加入多线程。设置 15 分钟后可通过微信支付订单查询用户是否成功支付；若用户提交订单但未付款，系统将自动取消订单。

### 3.4.3 任务中心如何确保接单人员以及完成任务

1. 程序采取双方确认机制：在双方确认送达后即可完成任务，完成任务后接单员即得酬金。
2. 程序采取拍照并上传，可给发布者确认完成任务。

### 3.4.4 图片渲染速度以及程序响应速度的慢

1. 程序采取 Redis 缓存保存部分“高读取，低修改”的数据，如登陆的 Token 信息、动态的前 Top 10 的文章等，可减少数据库的读取。
2. 程序采用两种对象存储混合使用，采用设计模式的策略模式，减少单一对象存储的读取压力，并对某个网络延迟较高的对象存储使用 CDN 加速。
3. 小程序前端已实现懒加载，将一部分缓存保存在用户手机上，从而减少服务器请求次数。

### 3.4.5 课程表数据保存

问题：通过 API 可获取到了课表信息，但格式为 JSON 格式，不适合保存在 Mysql 中。程序采取 ORM 映射，但 JSON 格式映射仍会映射成 String，且通过一些工具类进行 JSON 读取相关信息相当于遍历字符串；不同用户的数据对应 JSON 格式的内容不同，其内容涉及到时间、日期、课程以及地点不一致，查询当天课程等需遍历一整个 JSON 字符串，耗时较久，由于 Sql 语句的书写较为繁琐，需使用子查询，给数据库带来一定的压力。

解决方案：使用 Mogodb 而不是 Mysql

1. 由于原始数据是 JSON 格式的非关系型数据, 在关系型数据库 Mysql 下保存数据, 每个用户的数据字段不唯一, 耦合性过高; 在非关系型数据库 Mogodb 下, 不存在这个问题。
2. 使用 Mogodb 保存的格式为 JSON 格式, 读取相关的信息方便, 可根据键值对进行遍历查询, 可实现根据日期进行查询或者根据其他键值对查询。

## 3.5 接口设计

本程序制定了一套 API 设计规范:

1. API 风格尽量使用 RESTful 风格
2. 响应格式

```
interface APIResponse<T> {  
    data: T;  
    code: number;  
    msg: string;  
}
```

- ✧ data: JSON 格式的响应数据。
  - 出于可扩展性考虑, data 必须是对象不能是数组。
  - 如果请求失败, data 可以为 null。
- ✧ code: 自定义状态码, 非 [HTTP 状态码](#)。
- ✧ msg: 额外消息, 可用于请求失败时的错误提示。
- ✧ 字段命名风格: 返回的字段统一采用[驼峰命名法](#)。

3. ID 生成
4. 统一使用 uuid。
5. 时间格式: 返回的时间格式统一用 Unix 时间戳。
6. HTTP 状态码

HTTP 响应状态码用来表明特定 [HTTP](#) 请求是否成功完成。响应被归为以下五大类:

1. [信息响应](#) (100 - 199)
2. [成功响应](#) (200 - 299)
3. [重定向消息](#) (300 - 399)
4. [客户端错误响应](#) (400 - 499)
5. [服务端错误响应](#) (500 - 599)

遵循 [HTTP 状态码](#) 规范, 例如:

- 资源创建成功返回 201 状态码。
- 授权认证失败返回 401 状态码。
- 资源不存在则返回 404 状态码。

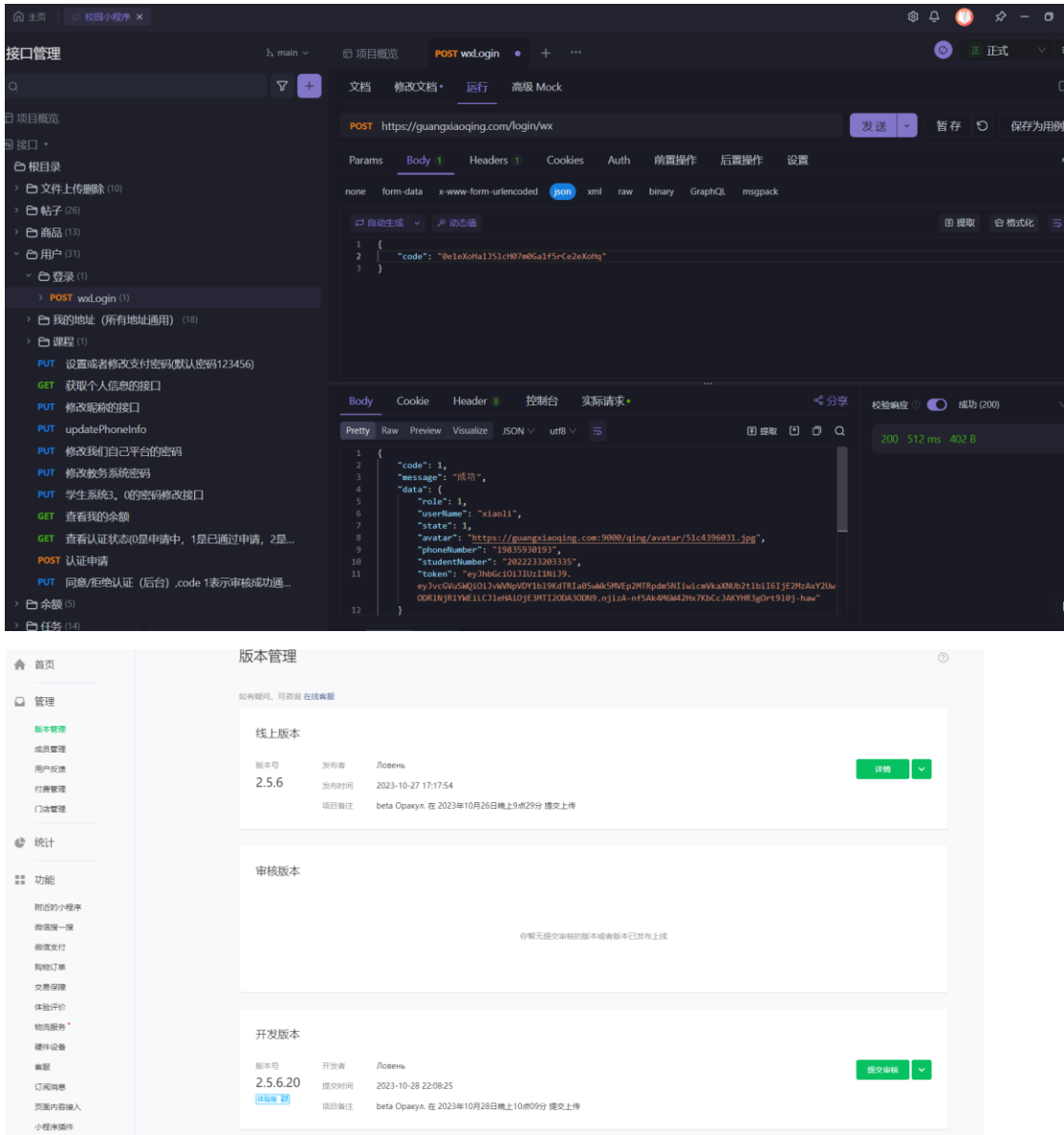
7. OpenAPI 参见 OpenAPI 规范 (中文版) <https://openapi.apifox.cn/>
8. 参数检验

API 规范文档(推荐):完整文档在本目录的 API 规范文档 PDF

在线链接(推荐):

<https://apifox.com/apidoc/shared-7a42976b-cc7d-4e4c-979f-b9fc95139fd3>

## 第四章 测试报告



1. 本程序配备 openapi 以及 json 格式的 api 文件,支持导入 Apifox 进行测试
2. 代码中每个功能都有单元测试,以及配置了全局异常处理,自定义异常等
3. 本程序 api 响应信息符合标准格式;如:4xx 代表前端传输数据异常或无权限等
4. 由于使用了 Redis 以及前端做了懒加载,所以性能得到了一定的提高
5. 程序经过开发测试,黑,白盒测试以及上线试运营测试,是一个成熟的程序

# 第五章 安装及使用

部署问题：

前端使用 uniapp 微信原生语言进行开发, 不需要安装大量的依赖, 部署上更方便。

后端使用了多环境部署, 有 local 环境, prod 环境以及 docker 环境, 并书写了 Dockerfile 进行构建镜像, 可利用 docker 进行部署, 实现了转移方便, 部署方便以及操作难度降低。

Project ▾  
GCC mini-app [qing] D:\system\desk\计算机  
  > .idea  
  > doc  
  > docker  
    Dockerfile  
    Gcc-mini-app.jar  
  > sql  
    devqing.sql

application.yaml

AddressMapper.xml

Dockerfile

application

```
1 FROM openjdk:20-jdk-alpine
2 VOLUME /tmp
3 add Gcc-mini-app.jar
4 RUN sh -c 'touch /docker_boot.jar'
5 ENTRYPOINT ["java", "-jar", "/docker_boot.jar"]
6
7 EXPOSE 443
```

RefundMapper.xml  
ReplyCommentMapper.xml  
SchoolAreaMapper.xml  
StudentAuditsMapper.xml  
TaskMapper.xml  
WithdrawalMapper.xml  
WxuserMapper.xml  
apiclient\_cert.p12  
apiclient\_key.pem  
application.yaml  
application-dev.yaml  
application-last.yaml  
application-local.yaml  
application-prod.yaml  
cuuemo.cn.pfx  
new.p12  
prod.pfx  
test  
target  
  classes  
  generated-sources  
  generated-test-sources

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

```
spring:
  profiles:
    active: prod

servlet:
  multipart:
    max-file-size: 5MB
    max-request-size: 5MB

wx:
  miniapp:
    configs:
      - appid: wxb53fb342651c6c2a
        secret: 6b9e4c65c7e531285e98e894b39afe6a
        token: xiaoligongzuoshi
        aesKey: xiaoligongzuoshi
        msgDataFormat: JSON

pay:
  appId: wxb53fb342651c6c2a
  mchId: 1652632090
  mchKey: on8rr08djaowyr9hyqfivsultaxcpjs9
  keyPath: classpath:new.p12
```

# 第六章 项目总结

本项目致力于构建一个卓越的校园服务平台, 为师生提供便捷性、安全性和多样性, 实现校园生活的完美互动与畅通交流。本项目的目标是对标有料同学、GCC 百宝箱等优秀校园助手程序, 为学生群体提供更好的校园生活服务。

在项目开发过程中，我们主要实现了以下主要功能模块：官方公告、动态专区、学生服务（二手闲置、任务中心、课程表）、系统配套功能（余额管理、地址管理、个人设置）等。同时，我们也进行了业务架构设计、应用架构设计、界面设计、数据库设计等方面的工作。

在技术实现方面，我们采用了一系列先进的技术框架和安全措施，包括数据库加密、SSL 证书配置、微信支付、Nginx 反向代理、前后端分离、JWT 身份认证等。这些技术的运用，保障了系统的安全性、性能和用户体验。

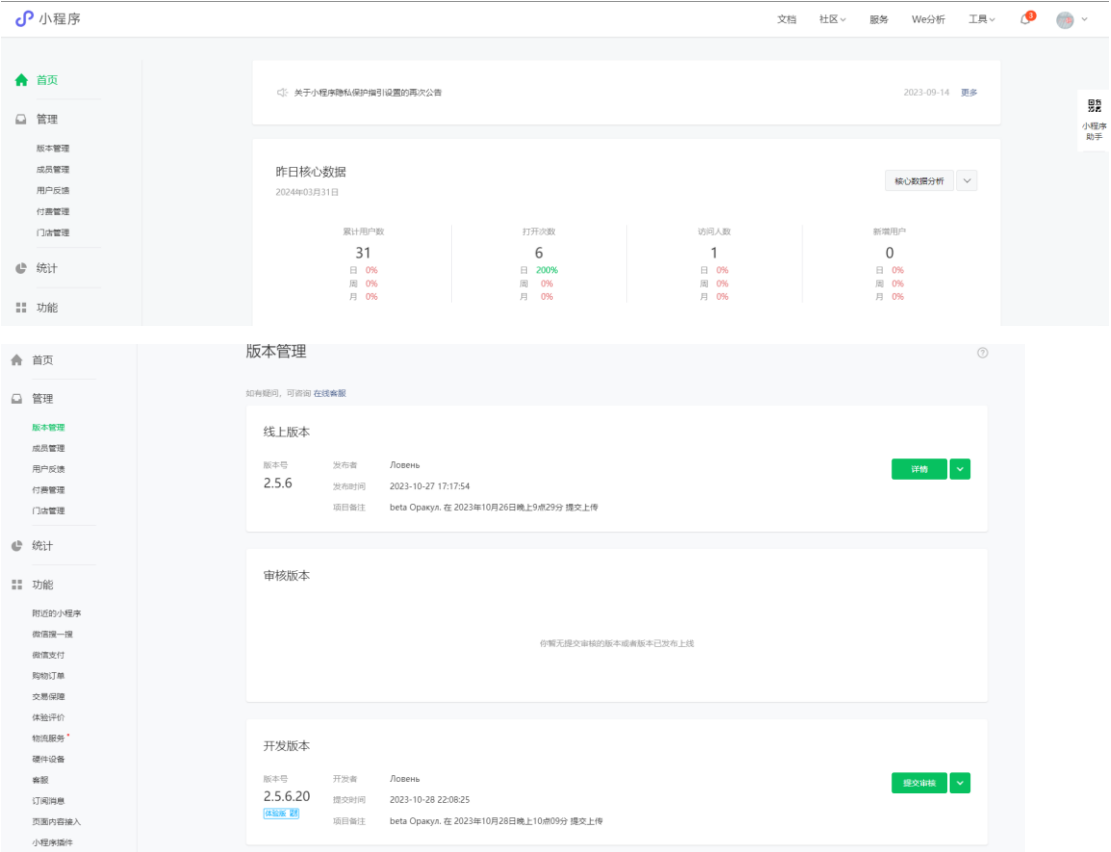
我们认识到支付安全、订单管理、任务中心的管理等问题是需要特别关注的重点。同时，我们也应该关注系统的运行速度、安全性、扩展性、部署方便性和可用性等技术指标，以保证整个系统的稳定运行和用户满意度。

在项目其他方面，我们需要不断总结经验，提高团队协作能力，不断优化产品体验，以期实现项目的长期发展和商业价值。

本项目是一个充满挑战和机遇的创新之路，我们相信通过团队的不懈努力和持续创新，一定能够打造一个卓越的校园服务平台，为广大师生提供更好的校园生活体验。

项目协调：本项目成员组开发积极性十分高, 配置十分默契, 前后端分工明确

升级演进：



程序已经发布试运营, 并获得了一定的流量, 后续可能调查下用户期望以及功能上的问题等进行迭代更新；

1. 任务完成审核采取更智能, 更方便的形式；
2. 更新 API 路由, 更符合 RESTful 风格

# 参考文献

【请按照标准参考文件格式填写】