

基于Uni-App与Spring Boot的校园代领跑腿平台设计与实现

孟 丽, 徐贞顺

北方民族大学计算机科学与工程学院, 宁夏 银川

收稿日期: 2026年5月29日; 录用日期: 2026年7月30日; 发布日期: 2026年8月7日

摘 要

本文基于uni-app和Spring Boot前后端分离架构, 设计并实现一款微信小程序形态的校园代领跑腿平台, 用以解决传统校园跑腿服务普遍存在的供需信息不对称、交易缺少资金保障、人工派单运营效率低下等痛点问题。平台面向普通用户、认证骑手、平台管理员三类角色, 落地账号注册登录、订单发布与抢单、全流程订单状态管控、个人账户钱包支付结算等核心业务; 系统依托骑手实名准入审核、多因子权重订单智能分发、交易资金存管三项核心机制优化业务闭环。论文完整开展需求调研、可行性分析、系统架构、功能模块与数据库设计, 重点对订单智能匹配算法、高并发抢单、支付安全等关键技术难点进行原理剖析与方案落地说明。经过功能全覆盖测试、多机型兼容性测试、高并发压力测试, 验证本系统运行稳定、高并发承载能力可靠, 具备良好使用体验, 在智慧校园便民服务数字化落地中具备实际应用价值。

关键词

校园跑腿, Uni-App, 前后端分离, 智能匹配, 交易安全

Design and Implementation of Campus Errand Service Platform Based on Uni-App and Spring Boot

Li Meng, Zhenshun Xu

School of Computer Science and Engineering, North Minzu University, Yinchuan Ningxia

Received: May 29, 2026; accepted: July 30, 2026; published: August 7, 2026

Abstract

Based on the separated front-end and back-end architecture of uni-app and Spring Boot, this paper

文章引用: 孟丽, 徐贞顺. 基于 Uni-App 与 Spring Boot 的校园代领跑腿平台设计与实现[J]. 软件工程与应用, 2026, 15(4): 547-559. DOI: 10.12677/sea.2026.154050

designs and implements a campus errand-running service platform in the form of a WeChat Mini Program, aiming to address prevalent pain points of traditional campus errand services, including asymmetric supply-demand information, lack of fund guarantee for transactions, and low operational efficiency caused by manual order assignment. The platform serves three types of users: regular users, verified errand runners and platform administrators, and realizes core business functions such as account registration and login, order release and order grabbing, full-process order status management, as well as personal wallet payment and settlement. Three core mechanisms are adopted to optimize the business loop, namely real-name access review for errand runners, intelligent order distribution based on multi-factor weighting, and escrow of transaction funds. This paper comprehensively conducts demand research, feasibility analysis, system architecture design, functional module design and database design. It focuses on analyzing the principles and implementing solutions for key technical difficulties including the intelligent order matching algorithm, high-concurrency order grabbing, and payment security. Full-coverage functional testing, multi-device compatibility testing and high-concurrency stress testing have been carried out. The test results verify that the system operates stably with reliable high-concurrency bearing capacity and favorable user experience, delivering practical application value for the digital construction of convenient public services in smart campuses.

Keywords

Campus Errands Service, Uni-App, Separated Front-End and Back-End, Intelligent Matching, Transaction Security

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着高校校园规模的不断扩大和学生生活需求的多样化, 校园跑腿服务逐渐成为学生日常生活中不可或缺的一部分。传统跑腿模式高度依赖 QQ 群、微信群等零散社交渠道, 面临订单分散、沟通成本高昂及资金难以追溯等瓶颈。为破解上述难题, 本文以师生实际需求为导向, 提出并构建了一个集订单实时跟踪、资金第三方担保及评价监督于一体的闭环服务生态。本研究设计并实现了一个基于 uni-app 框架的校园代领跑腿平台[1], 旨在为校园师生提供安全、高效、便捷的跑腿服务。

2. 系统分析

2.1. 需求分析

当前高校学生课程紧凑、活动密集, 对即时性、便捷性服务需求显著提升。传统跑腿模式依赖 QQ 群、微信群、朋友圈等渠道发布信息, 存在订单分散、沟通成本高、信任度不足、纠纷难以处理等问题。同时, 骑手资质无法统一审核、服务质量参差不齐、订单状态无法实时追踪、资金交易缺乏第三方保障, 导致整体服务效率偏低。为解决上述痛点, 本平台以师生实际需求为导向, 构建了集订单发布、智能接单、实时跟踪、安全支付、评价监管于一体的数字化校园跑腿服务体系。通过对校园师生的调研和分析, 平台主要满足以下需求:

1) 用户需求: 用户能够发布代拿快递、代取餐品、代买零食、代送鲜花等多种类型的订单, 查看订单状态, 管理个人信息和地址。

- 2) 骑手需求：骑手能够查看待接单订单，接单并完成配送，管理个人认证信息，查看收支明细。
- 3) 平台需求：平台能够管理用户和骑手信息，处理订单匹配，保障交易安全，发布系统通知。

2.2. 可行性分析

技术层面，uni-app 一次开发多端发布，可快速生成微信小程序，降低开发与维护成本；Spring Boot 具备高效稳定的后端支撑能力，技术成熟度高、开源生态完善，开发风险低。经济层面，平台依托云服务器轻量化部署，初期投入低，后期可通过订单佣金、广告位等方式实现良性运营，具备可持续性。社会层面，平台既能为学生提供安全可靠的生活服务，也可为在校学生提供合规兼职岗位，提升校园服务数字化水平，具备良好应用前景。

2.3. 国内外研究现状与同类产品对比

2.3.1. 国内外研究现状

1) 国外研究现状

海外校园跑腿服务发展成熟，研究多聚焦调度算法、配送管理与平台风控，产品以原生 APP 为主，极少采用小程序架构。欧洲学者 Giuffrida 分析了校园跑腿落地可行性，采用兼职配送模式，但无线上资金担保与纠纷处理机制[2]。总体来看，国外相关技术虽成熟，但架构、准入规则与国内校园管理模式不符，也为本研究选用小程序架构、实名认证、多维派单提供了思路。

2) 国内研究现状

李洋、李文超探讨了校园跑腿线上化优化思路[3]，仅停留在理论阶段。李静、常齐月开发的小程序仅具备基础接单下单功能，缺少风控模块[4]。张宏娜等人研发的抢单系统基于 Web 架构[5]，不适配移动端小程序。周小华以新闻调查形式剖析跑腿行业经营乱象，发现商业跑腿平台存在强制收取入网费、退费机制缺失、从业者资金权益无保障等监管漏洞，仅从线下行业规范角度提出整改建议，未针对校园场景设计线上跑腿服务系统，缺乏订单风控、实名核验、资金担保等数字化解决方案研究[6]。国内现有平台普遍存在认证宽松、订单匹配简单、资金无保障、并发稳定性差等问题。本文结合 Uni-app 与 Spring Boot 技术，融合现有研究优势，设计集智能派单、实名认证、资金担保于一体的解决方案，弥补现存短板。

2.3.2. 同类平台对标分析

为直观对比国内外现有校园跑腿服务产品的技术方案与运营模式差异，本文选取 5 款主流同类平台，从前段载体、派单逻辑、骑手管控、资金结算四大维度开展横向对比，具体内容如表 1 所示。

Table 1. Comparison of similar platforms table
表 1. 同类平台对标分析表

平台名称	前端载体	派单方式	骑手管控	资金结算
美团跑腿校园版	小程序 + APP	系统就近派单 + 人工调度	平台签约专职骑手	平台统一结算提现
校米跑腿	微信公众号	用户自主抢单，无智能匹配	手机号简易注册，无在校资质核验	第三方平台提现
校内快跑	微信小程序	单纯距离派单	自由注册，无需学生认证	个人微信转账结算
美国 Grubhub 校园版	原生 APP	AI 多参数派单 (距离 + 时效)	社会全职骑手，无在校限制	校园卡 + 信用卡结算
韩国 Zoomi 校园跑腿	Kakao 内嵌 H5	就近距离自动派单	简单实名认证，不限在校生	本土电子钱包结算

3. 系统设计

3.1. 系统架构

本系统选用前后端分离的开发模式，能有效减少各功能模块间的关联程度，同时提高开发速度与系统后续拓展能力。前端分为微信小程序与管理后台两部分：微信小程序面向师生用户与骑手，提供轻量化、高可用的移动端服务，支持订单快速发布与状态实时刷新；管理后台面向管理员，实现用户审核、订单监控、数据统计、系统公告等功能。后端采用 Spring Boot 构建 RESTful 接口，系统后端依照控制层、业务层、数据访问层的分层结构进行搭建，实现业务处理与数据操作的解耦，保证代码规范与可维护性。数据层采用 MySQL 存储结构化业务数据，Redis 缓存高频访问数据，减轻数据库压力，提升系统响应速度与并发承载能力。整体架构层次清晰、易于扩展，可满足后续功能迭代需求。

- 前端层负责用户界面的展示和交互，采用了双端设计：
- 1) Vue 管理后台：基于 Vue 2.6.14 框架和 Element UI 组件库构建，为管理员提供直观的系统管理界面，支持用户管理、订单管理、数据统计等功能。
 - 2) 微信小程序：基于 Uni-app 跨平台框架开发，支持在微信小程序端运行，为用户提供便捷的移动端服务，包括订单发布、状态查询、钱包管理等功能。
- 后端层是系统的核心，负责处理业务逻辑和数据处理：
- 1) Spring Boot 应用：提供 RESTful API 接口，处理前端的请求。
 - 2) 控制器层(Controller)：接收和处理 HTTP 请求，调用相应的服务进行业务处理。
 - 3) 服务层(Service)：实现核心业务逻辑，如用户认证、订单管理、支付处理等。
 - 4) 数据访问层(Mapper)：与数据库交互，执行数据的增删改查操作。
- 数据层负责数据的存储和管理：
- 1) MySQL 数据库：存储系统的结构化数据，包括用户信息、订单信息、地址信息、交易记录等。
 - 2) Redis 缓存：用于缓存热点数据，如用户会话、订单状态等，提高系统的响应速度和并发处理能力。
- 具体如图 1 所示。

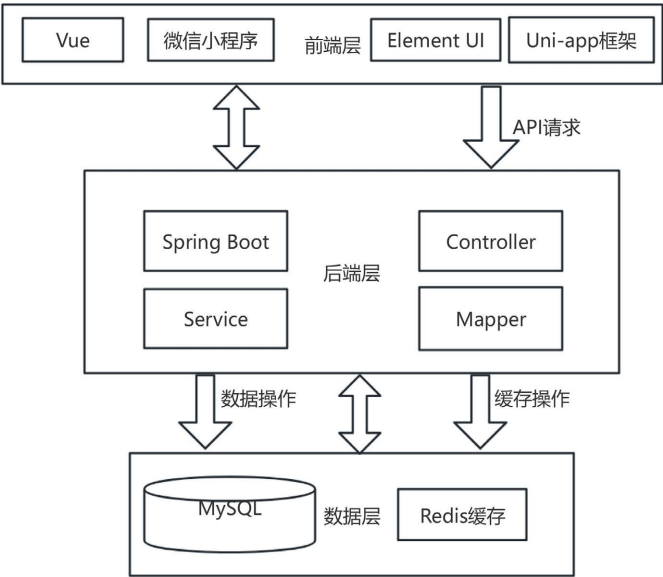


Figure 1. System architecture diagram
图 1. 系统架构图

3.2. 功能模块设计

系统围绕用户、骑手、管理员三种角色构建完整业务流程，实现订单发布、智能接单、实时追踪、安全支付、服务评价等核心功能。用户模块支持快速注册、实名验证、地址管理，确保服务对象真实可信。订单模块为核心模块，覆盖订单创建、智能推荐、接单、配送、完成、取消全流程，状态实时同步。骑手模块严格执行实名认证与学生资质审核[1] [7] [8]，保障服务安全性与可靠性。支付模块对接微信支付，支持余额充值、自动扣费、佣金结算、提现，交易记录可追溯。评价模块实现服务后评分与留言，形成监督机制。消息模块推送订单状态、系统公告、提现结果等通知，提升信息触达效率。平台主要包含以下功能模块：

- 1) 用户模块：用户注册、登录、个人信息管理、地址管理。
- 2) 订单模块：订单发布、接单、状态管理、订单详情查看。
- 3) 骑手模块：骑手身份认证和资质审核。
- 4) 支付模块：充值、提现、收支明细查看。
- 5) 评价模块：用户对骑手的评价和评分。
- 6) 消息模块：系统通知发布和查看。具体如图 2 所示。

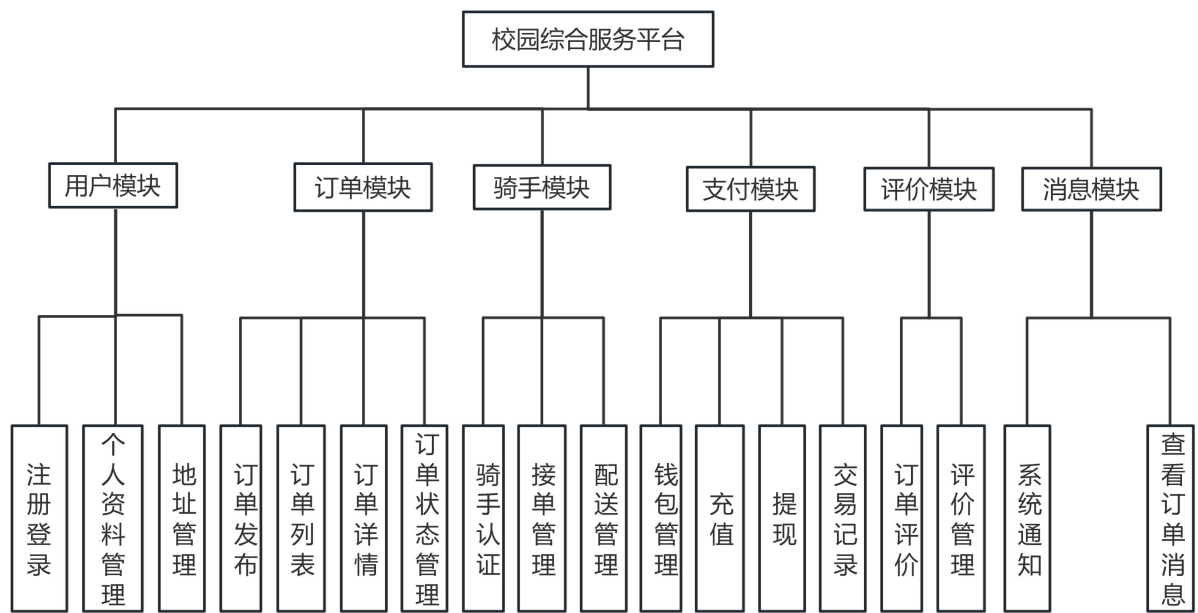


Figure 2. System function module diagram
图 2. 系统功能模块图

3.3. 数据库设计

平台数据库包含用户、地址、订单、评价、认证、钱包等数据表，用于支撑用户管理、订单交易、资金结算等业务运行。

- 1) 用户信息表用于统一存储平台所有使用者的账号基础信息、身份角色类型，以及骑手资格审核状态等关键数据。如表 2 所示。
- 2) 地址信息表用于保存用户常用收货与配送地址，支持多地址管理与默认地址标记。如表 3 所示。
- 3) 订单数据表用于记录平台所有跑腿订单的详细信息、业务状态与各环节时间节点。如表 4 所示。

Table 2. User table
表 2. 用户表

字段名	数据类型	约束说明	自动生成	备注
id	Integer	主键，自增	自动生成	用户唯一标识
username	String	用户名，长度 3~20	-	登录用户名
password	String	密码，加密存储	-	登录密码
phone	String	手机号，唯一	-	用于登录和验证
is_rider	Boolean	是否为骑手	默认值为 False	标记用户是否为骑手
rider_status	String	骑手状态	默认值为 pending	Pending: 待认证， approved: 已认证，rejected: 已拒绝
created_at	Date Time	创建时间	默认值为当前时间	记录用户注册时间
updated_at	Date Time	更新时间	默认值为当前时间，自动更新	记录信息更新时间

Table 3. Address table
表 3. 地址表

字段名	数据类型	约束说明	自动生成	备注
id	Integer	主键，自增	自动生成	地址唯一标识
user_id	Integer	外键，关联 user 表，级联删除	-	用户 ID
name	String	收货人姓名	-	收货人姓名
phone	String	收货人手机号	-	收货人联系方式
province	String	省份	-	地址省份
city	String	城市	-	地址城市
district	String	区县	-	地址区县
detail	String	详细地址	-	具体门牌号等
is_default	Boolean	是否默认地址	默认值为 False	标记默认地址
created_at	Date Time	创建时间	默认值为当前时间	记录地址创建时间
updated_at	Date Time	更新时间	默认值为当前时间，自动更新	记录地址更新时间

Table 4. Order table
表 4. 订单表

字段名	数据类型	约束说明	自动生成	备注
id	Integer	主键，自增	自动生成	订单唯一标识
user_id	Integer	外键，关联 user 表，级联删除	-	下单用户 ID
rider_id	Integer	外键，关联 user 表，可为空	-	接单骑手 ID
type	String	订单类型	-	代拿快递/代取餐品/ 代买零食/代送鲜花/其他
name	String	订单名称	-	订单简要描述
price	Decimal	订单价格	-	跑腿费用

续表

status	String	订单状态	默认值为 pending	Pending: 待接单, accepted: 待配送, delivering: 配送中, completed: 已完成, cancelled: 已取消
address_id	Integer	外键, 关联 address 表, 级联删除	-	收货地址 ID
description	Text	订单详细描述, 可为空	-	订单具体要求
created_at	DateTime	创建时间	默认值为当前时间	订单创建时间
accepted_at	DateTime	接单时间, 可为空	-	骑手接单时间
completed_at	DateTime	完成时间, 可为空	-	订单完成时间
cancelled_at	DateTime	取消时间, 可为空	-	订单取消时间

4) 服务评价表用于存储用户对骑手完成订单后的评分与文字反馈, 形成服务质量监督机制。如表 5 所示。

Table 5. Comment table
表 5. 评价表

字段名	数据类型	约束说明	自动生成	备注
id	Integer	主键, 自增	自动生成	评价唯一标识
order_id	Integer	外键, 关联 orders 表, 唯一	-	订单 ID
user_id	Integer	外键, 关联 user 表, 级联删除	-	评价用户 ID (下单用户)
rider_id	Integer	外键, 关联 user 表, 级联删除	-	被评价骑手 ID
rating	Integer	评分, 1~5 星	-	服务评分
content	Text	评价内容, 可为空	-	评价详细内容
created_at	DateTime	创建时间	默认值为当前时间	评价创建时间

5) 系统通知表用于存储平台向用户推送的公告、订单提醒、状态变更等消息内容。如表 6 所示。

Table 6. Notification table
表 6. 通知表

字段名	数据类型	约束说明	自动生成	备注
id	Integer	主键, 自增	自动生成	通知唯一标识
title	String	通知标题	-	通知标题
content	Text	通知内容	-	通知详细内容
is_active	Boolean	是否激活	默认值为 True	标记通知是否显示
created_at	DateTime	创建时间	默认值为当前时间	通知创建时间
updated_at	DateTime	更新时间	默认值为当前时间, 自动更新	通知更新时间

6) 骑手认证信息表用于管理骑手身份审核资料, 确保骑手为在校学生并完成实名验证。如表 7 所示。

Table 7. Rider qualification certification table
表 7. 骑手认证管理表

字段名	数据类型	约束说明	自动生成	备注
id	Integer	主键，自增	自动生成	认证唯一标识
user_id	Integer	外键，关联 user 表，唯一	-	用户 ID
real_name	String	真实姓名	-	骑手真实姓名
student_id	String	学号	-	骑手学号
id_card	String	身份证号	-	骑手身份证号
photo	String	证件照片 URL	-	身份证照片或学生证照片
status	String	认证状态	默认值为 pending	Pending: 待审核， approved: 已通过， rejected: 已拒绝
reason	Text	拒绝原因，可为空	-	认证拒绝原因
created_at	DateTime	创建时间	默认值为当前时间	认证申请时间
updated_at	DateTime	更新时间	默认值为当前时间，自动更新	认证状态更新时间

7) 用户钱包表用于记录账户余额、收支明细与各类交易流水，保障资金安全可追溯。如表 8 所示。

Table 8. Wallet table
表 8. 钱包表

字段名	数据类型	约束说明	自动生成	备注
id	Integer	主键，自增	自动生成	记录唯一标识
user_id	Integer	外键，关联 user 表，级联删除	-	用户 ID
type	String	记录类型	-	Recharge: 充值， withdraw: 提现， order_pay: 订单支付， order_income: 订单收入
amount	Decimal	金额	-	交易金额，正数表示收入， 负数表示支出
status	String	状态	默认值为 success	Success: 成功， pending: 处理中，failed: 失败
order_id	Integer	外键，关联 orders 表，可为空	-	关联订单 ID (如果是订单相关交易)
transaction_id	String	第三方交易 ID，可为空	-	如微信支付交易 ID
created_at	DateTime	创建时间	默认值为当前时间	记录创建时间

8) 管理员信息表用于存储平台后台管理人员账号、权限与操作信息，保障后台管理安全。如表 9 所示。

3.4. 平台核心算法与关键技术难点实现

3.4.1. 订单智能匹配与分发机制

本平台采用多维度加权评分算法实现订单智能派单，派单权重因子包含骑手在线状态、骑手与下单用户地理位置距离、骑手当前待接单负载、历史服务好差评率四项指标。当用户提交跑腿订单后，后端首先筛选

满足服务范围、在线已完成实名认证的骑手候选集合，依据预设权重公式计算每位骑手综合得分，系统优先将订单推送至综合评分最高骑手；若骑手超时未接单，则顺延推送至次优骑手。具体流程图如图 3 所示。

Table 9. Admin table
表 9. 管理员表

字段名	数据类型	约束说明	自动生成	备注
id	Integer	主键，自增	自动生成	管理员唯一标识
username	String	用户名，唯一	-	登录用户名
password	String	密码，加密存储	-	登录密码
name	String	管理员姓名	-	管理员真实姓名
role	String	角色	默认值为 admin	Admin: 超级管理员, operator: 操作员
created_at	DateTime	创建时间	默认值为当前时间	管理员创建时间
updated_at	DateTime	更新时间	默认值为当前时间，自动更新	管理员信息更新时间

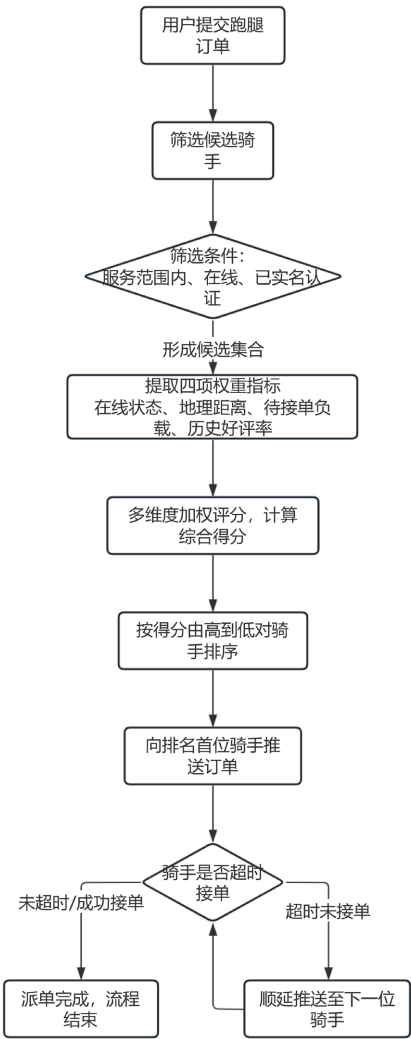


Figure 3. Algorithm flowchart
图 3. 算法流程图

算法伪代码:

```
//订单智能匹配计算逻辑
List<Rider> allOnlineAuthRider = 查询在线且完成实名骑手集合;
for (Rider rider : allOnlineAuthRider){
    double score = 距离权重*DistanceScore + 负载权重*LoadScore + 好评权重*CommentScore;
    rider.setTotalScore(score);
} //骑手按分数降序排序, 自上而下推送订单
sort(allOnlineAuthRider DESC by TotalScore);
sendOrderPush(allOnlineAuthRider);
```

3.4.2. 关键技术难点解决方案

1) 高并发抢单防超卖方案

多骑手同时争抢单个订单会出现重复接单的数据异常, 本系统采用 Redis 分布式锁 + 数据库乐观锁双保险方案: 骑手发起抢单请求时[9], 后端利用 Redis SETNX 指令抢占订单分布式锁, 抢到锁后再更新数据库订单状态; 数据库订单表增设 version 版本字段, 更新订单时通过版本号实现乐观锁校验, 避免超卖问题。同时使用 RabbitMQ 消息队列削峰, 瞬时大量抢单请求入队异步处理, 缓解服务器瞬时压力。

2) 钱包支付安全实现方案

针对平台充值、佣金提现的资金安全问题, 从三层做防护: ① 所有资金变动接口添加请求签名校验, 防止接口恶意篡改参数; ② 每一笔充值、提现、订单结算操作全量落地资金流水日志, 流水不可删除; ③ 用户密码、支付密钥采用 AES 对称加密存储, 杜绝明文泄露, 保证全链路资金可追溯。

3.4.3. 多层次安全防护体系整体设计

针对平台高并发抢单与资金交易两类核心风险, 本系统搭建多层防护体系解决技术难点, 在抢单并发控制层面采用 Redis 分布式锁搭配数据库乐观锁双重校验机制, 骑手提交抢单请求时后端通过 SETNX 指令抢占订单缓存锁过滤无效并发请求, 同时订单表增设 version 版本字段, 更新订单状态时比对版本号杜绝多骑手重复接单的超卖问题, 搭配 RabbitMQ 消息队列对瞬时海量请求异步削峰, 缓解服务器瞬时访问压力; 在钱包资金安全层面构建三层防护逻辑, 所有充值、提现、结算类资金接口增加请求签名校验抵御参数篡改攻击, 每一笔资金操作永久留存不可删除的交易流水并绑定订单与第三方支付单号实现全程溯源, 用户密码、支付密钥等敏感信息统一通过 AES 对称加密后存储, 从接口、业务日志、数据存储全链路保障平台交易资金安全。

4. 系统实现

4.1. 技术栈选择

本系统选用成熟可靠的技术组合完成开发工作, 以此保障系统运行稳定且便于后期维护与升级, 具体技术方案如表 10 所示。

Table 10. Technology stack table
表 10. 技术栈表

类别	技术/框架	版本	用途	类别
前端框架	Uni-app	3.0+	跨平台应用开发	前端框架
前端语言	Vue.js	2.0	界面开发	前端语言
后端框架	Spring Boot	2.7.x	后端服务开发	后端框架
数据库	MySQL	8.0	数据存储	数据库
接口文档	Swagger	3.0	API 文档生成	接口文档
部署环境	云服务器	CentOS 7	应用部署	部署环境

4.2. 核心功能实现

4.2.1. 用户注册登录

用户注册时, 系统会验证手机号的合法性。登录时, 系统会验证用户名和密码, 并生成 JWT 令牌用于后续的身份验证。

4.2.2. 订单发布与接单

用户发布订单时, 需要选择订单类型、填写订单详情和价格。骑手可以查看待接单订单, 并选择合适的订单进行接单。

4.2.3. 订单状态管理

订单状态包括待接单、待配送、配送中、已完成、已取消等状态, 系统会根据订单的进展自动更新订单状态。

4.2.4. 钱包与支付管理

用户可以通过微信支付进行充值, 骑手可以提现 earnings。系统会记录所有的交易记录, 用户可以查看收支明细。

5. 系统测试与运行

5.1. 测试环境

为保证测试结果准确可信, 本次测试搭建完整软硬件环境: 服务端采用 CentOS 7 云服务器[10], MySQL 8.0 关系型数据库, Redis 6 缓存, JDK 1.8 运行环境; 客户端操作系统为 Windows 10, 移动端包括 iOS 16 与 Android 13 系统; 开发与调试工具为 HBuilderX 3.8.0、微信开发者工具; 测试设备为 iPhone 13、华为 Mate 40 智能手机; 网络环境包括校园 WiFi 与移动 4G 网络; 测试方式以人工遍历测试、接口调用测试、并发模拟测试为主。

5.2. 功能测试

以系统核心业务流程为测试主线, 设计典型测试用例, 对用户模块、订单模块、骑手模块及钱包管理模块进行逐项验证。

测试环节主要覆盖以下内容: 用户账号注册与登录校验、个人信息修改、代取快递与代买商品等订单创建、骑手接单及抢单操作、配送进度实时同步、订单信息查询、账户余额查询与资金提现等。

测试结果表明: 系统各模块功能逻辑正确, 业务流程完整顺畅, 未出现崩溃、卡顿与数据异常问题, 各项功能均达到设计预期, 可满足校园跑腿实际使用需求。

5.3. 兼容性测试

由于系统基于微信小程序与 uni-app 开发, 需对不同移动平台进行兼容性测试。分别在 iOS 与 Android 系统、不同分辨率机型、不同微信基础库版本下进行界面展示与交互操作测试。

测试结果显示: 系统页面布局自适应效果良好, 按钮响应、图片加载、表单提交等操作均正常, 在主流机型上均能保持一致的交互体验, 兼容性满足要求。

5.4. 性能测试

在保证功能正常的前提下, 对系统并发处理能力与响应效率进行测试。通过模拟多用户同时发起订单查询、提交请求等操作, 检测接口响应时间与系统稳定性。

压力测试结果显示: 在 50 并发用户数(VUM)的模拟压测场景下, 系统核心接口平均响应时间(RT)均保持在 2 秒以内, 吞吐量稳定, 无明显请求阻塞或内存溢出(OOM)现象。这表明系统具备优良的高并发承载能力, 完全契合校园典型业务场景的应用需求。

5.5. 系统运行效果

目前系统已完成部署并在校园范围内投入试运行。实际运行表明, 系统运行稳定可靠, 界面简洁直观, 操作流程简便, 师生使用体验良好。

系统可高效实现校园内跑腿需求发布与接单配送闭环, 提升了校园生活服务效率, 具有一定的实用价值与推广意义, 如下图 4 所示。



Figure 4. Page module block diagram
图 4. 页面模块框图

6. 总结与展望

本文以 uni-app 为核心技术, 设计并落地实现了一款校园代领跑腿服务平台[6], 解决了传统校园跑腿服务的痛点。平台采用前后端分离架构, 支持微信小程序端运行, 实现了用户注册登录、订单发布与接单、订单状态管理等核心功能。平台通过严格的骑手认证机制确保服务质量, 高效的订单分发机制提升配送效率, 以完善的交易安全保障机制保障用户权益。

未来, 平台可以进一步优化流程, 提升订单流转效率; 加强数据分析, 为用户提供个性化服务推荐。平台的成功实现为校园生活服务数字化转型提供了参考, 具有一定的推广价值。

参考文献

[1] 罗颖. 代排队代跑腿 南宁“代经济”悄然兴起[N]. 南宁晚报, 2025-08-02(005).

-
- [2] Giuffrida, N., Pira, M.L., Fazio, M., Inturri, G. and Ignaccolo, M. (2021) On the Spatial Feasibility of Crowdshipping Services in University Communities. *Transportation Research Procedia*, **52**, 19-26.
<https://doi.org/10.1016/j.trpro.2021.01.004>
- [3] 李洋, 李文超. 校园应用场景下跑腿类应用的改进思考[J]. 数字通信世界, 2024(3): 54-56.
- [4] 李静, 常齐月. 基于微信小程序的校园互助平台设计[J]. 软件, 2023, 44(6): 104-106.
- [5] 张宏娜, 周绪川, 贾泽锋. 基于 Web 技术的校园跑腿 APP 抢单算法的设计与实现[J]. 西北民族大学学报: 自然科学版, 2017, 38(3): 30-34.
- [6] 周小华, 廖叶婷. “这些钱收得安心吗?” [N]. 长沙晚报, 2024-11-05(006).
- [7] 张雨柔, 丁永慧, 江婷. 校园“跑腿”业务经营策略研究[J]. 市场周刊, 2023, 36(5): 33-36.
- [8] 余海燕. 基于众包平台的跑腿代购即时配送优化研究[J]. 工业工程, 2022, 25(4): 100-108.
- [9] 郭丁然. 用数据帮 2 万多家中小微企业融资 河南这一做法已向全国推广[N]. 河南商报, 2021-02-25(A02).
- [10] 陈合合, 覃冬玲, 李林霞. 积分换服务——“互联网+”模式下的可行性研究[J]. 现代交际, 2019(7): 76-77.