

微信小程序应用开发赛

群精选接龙介绍

目 录

1	项目介绍	1
1.1	项目背景	1
1.1.1	数据采集	1
1.1.2	社区团购	1
1.2	项目简介	2
1.2.1	雏形	2
1.2.2	现状	2
1.2.3	展望	3
1.3	目标用户	3
1.4	应用场景	3
1.4.1	社区商家卖货	3
1.4.2	社区居民买货	4
1.4.3	供应商分销	4
1.5	痛点与解决方案	4
1.5.1	线下商家转型	4
1.5.2	商家资源对接	5
1.5.3	提供就业岗位	5
1.6	竞品分析	5
2	产品设计	6
2.1	产品功能	6
2.1.1	首页	6
2.1.2	店铺	6
2.1.3	我的	7
2.2	交互设计	8
2.2.1	创建团购接龙流程	8
2.2.2	购物流程	9
2.3	运营方案	10
2.3.1	引流策略	10
2.3.2	存留策略	11
2.3.3	转化策略	11

2.3.4	优化策略	12
3	关键技术	13
3.1	客户端	13
3.1.1	小程序端	13
3.1.2	浏览器端	13
3.2	服务端	14
3.2.1	开发框架	14
3.2.2	部署环境	14
4	系统设计	16
4.1	总体设计	16
4.1.1	架构设计	16
4.1.2	数据库设计	17
4.1.3	功能结构设计	18
4.2	功能模块设计	19
4.2.1	接龙管理模块	19
4.2.2	信息管理模块	20
4.2.3	店铺管理模块	21
4.2.4	订单管理模块	22
4.2.5	用户中心模块	23
5	系统测试	25
5.1	测试环境	25
5.2	测试流程与方法	25
5.3	测试结果	27
5.3.1	功能性测试	27
5.3.2	服务端性能测试	28
5.3.3	小程序端性能测试	29

1 项目介绍

1.1 项目背景

1.1.1 数据采集

2020 年初，自湖北省武汉市爆发的新型冠状病毒感染肺炎向全国蔓延，此次疫情已被 WHO 认定为国际关注的突发卫生公共事件。为防止疫情扩散，社会各界（院校、企业、社区、医疗机构等组织）需要切实做好人员健康情况统计工作。但是，并不是所有组织都有财力、技术迅速开发一款应用来辅助统计、提高效率，而传统的统计方式如纸张记录、Excel 软件记录等，不仅操作不便而且统计难度大，由此观之，社会亟需一款通用且能够方便快捷地记录信息并提供自动统计数据的工具。

1.1.2 社区团购

团购的概念兴起于 2016 年的长沙，简单地说，社区拼团就是微商的升级版，与微信上的 S2B2C（Supply chain platform to Business to Customer，服务于中小商家的供应链平台）分销模式类似，只不过将分销、拼团玩法与社区场景结合在一起，核心在于围绕社区建起社交圈，把一个小区的业主拉进一个群里，统一集采集配。社区团购所具有的特点是：社交属性、社群场景、低价团购、C2B 预售以及落地配。

为赢得疫情这场没有硝烟的战争，国家出台多项政策扶持中小企业并鼓励线上零售的发展。而相较于昂贵的线上平台入驻和商品推广费用，社区团购抓住微信群，结合社区熟人关系链这一新渠道的流量红利，有效降低获客成本。与线下实体店、“店”加“柜”等重运营模式相比，社区团购没有门店或利用已有社区零售店，省去了店铺的高租金、高人力和高时间成本，运营模式轻，操作难度相对较低，便于潜在团长快速入门。

受疫情影响，居民出行意愿大大降低，传统的社区零售店客流量大大减少，难以维持运营。如果商家转型线上成为社区团购的团长，由于居民有更多的时间花费在微信等互联网 App 上，疫情时期将是宣传营销的好机会。

由于全国各地社区纷纷实施封闭式管理，快递物流“最后一公里”的配送问题变得更为严重，而社区团购采用落地配送代替快递物流，主要由社区内的团长负责将商品落实到团员手里，不仅解决了封闭管理下居民生活必需品的刚需，而且降低了物流成本，交付体验也比电商更好，稳定性更优[7]。再者，每一团长的提货点或社区店均有专职人员负责商品的服务和售后，便于消费者提交退换货申请，及时便捷的售后服务有利于增加顾客黏性。

总而言之，疫情的爆发使得大多数零售企业线上线下全生态业务的供应链建设产生变革，人货场的关系被重构，社区、社群、社交的价值被凸显，社区团购业态在此次疫情中获得爆发式增长。在新模式的冲击下，传统的线下商家、个人卖家甚至供应商，如何借助红利趋势打造自己的微信社区团购，是一个值得考虑的问题。

1.2 项目简介

群精选接龙以社群卖货分销（接龙）工具为切入点，架设一个在供应商、团长与消费者间的平台，上游对接供货商与团长，下游连接消费者，为团购开启新模式。

1.2.1 雏形

项目的前身是于疫情期间开发的群接龙工具，以小程序的形式对微信传统的接龙功能进行拓展，提供更好的界面设计与统计功能。通过使用我们的工具轻松地创建填表接龙，可以方便快捷地统计成员每日健康状况、口罩购买情况等信息。小程序一经上线便广泛应用于社会各界，在短时间内吸引了数十万用户，形成平台宝贵的初始客源。

1.2.2 现状

由于居民小区进行封闭式管理，限制出行，很多用户利用填表接龙功能进行生活物资购买统计，由商家和志愿者统一送货上门。结合当下社区团购模式爆发增长的背景，为项目新增了团购接龙的功能，通过在小程序中上架货品，一键转发群聊便可以开始卖货。同时还有拼团、秒杀等热门营销模式可供选择。销售进程实时更新，通过微信在线收款，另有一键导出接龙数据，快速配货发送等丰富功能。本平台目前的运营数据如下所示：

表1-1 运营数据

累计用户个数	236.2 万
日活跃用户个数	10.2 万
活跃团长个数	2.8 万
铺主个数	5650
店铺个数	7253
月交易额	1584 万

1.2.3 展望

现阶段，为了打造一个一站式的销售平台，正招募各大供应商以及线下商家、个人卖家等潜在团长入驻。彼时，将提供团长、供货商资源对接服务，并通过定期开展免费的社群管理培训，帮助团长建立社群、运营社群以及快速出单。

1.3 目标用户

群精选接龙主要服务于线下社区商家、社区居民及大供应商，可以满足上述人群的下列需求：

- ① 希望转型线上销售的社区商家，可将微信群转化为自己的私域流量，成为团长，并借助平台向团员推荐商品完成卖货流程；
- ② 喜欢从可靠的渠道以低廉的价格购买商品的用户，可以通过群精选接龙平台，从周边熟悉的商户处购买商品；
- ③ 拥有大量货源的供应商，可以联系平台运营者构筑供应链，再借助平台团长资源对商品进行分销。

1.4 应用场景

1.4.1 社区商家卖货

上午 10 点，B 的便利店进了一批商品，是新鲜的酸奶、山竹等。由于乳制品和水果保鲜期只有几天，店主 B 想尽快将商品销售出去。但是受新冠疫情影响且小区采取封闭式管理，很多居民外出意愿不高，B 的店铺门可罗雀，新到的商品无人问津。这时，B 想起了群精选接龙小程序，他立刻打开手机，在平台的店

铺中新建购物接龙，并分享到由小区居民组成的微信群。看到消息的部分居民表示感兴趣，纷纷下单后即由店铺专人送货上门，商品很快就售罄了。

1.4.2 社区居民买货

白领 C 在写字楼上班。下午 1 点，正值休息期间，C 想看看今天有什么实惠的东西可以买回家，于是打开微信，在居民群中看到小区附近便利店的店主 B 正用群精选接龙卖山竹和酸奶，日期新鲜、价格合理，已有几十人参与了接龙购买，C 也接龙各买了一份，并备注了送达时间。下班回到家后，水果和酸奶按时送达。

1.4.3 供应商分销

供应商 S 掌握大量商品资源，苦于分销的线下实体店关门或者客流量少等问题而有大量商品滞销，亟需开辟新的卖货渠道。S 与群精选接龙平台进行联系，建立合作关系。

S 为平台提供商品来源，平台则通过直播带货的方式向活跃的团长推荐商品，再由团长向各自的团员推销商品，从而帮助 S 分销商品。在给消费者带来物美价廉的商品的同时，也提升了 S 的销量。

1.5 痛点与解决方案

随着电商发展成熟，线下实体店客流量越发减少，小商家纷纷谋求增加或转型线上模式的机会。然而传统电商平台准入门槛高，如果想增加客流量还需要支付平台高额的引流费用，导致部分商家苦苦找不到加入电商行业的门路。另外小商家小本经营，往往没有特别多的供货源，也很少为了时下热门爆款去寻求新的供应商，从而错失许多盈利的良机。还有许多全职宝妈或白领希望可以在家赚些外快，但目前许多在家就能赚钱的线上兼职不是骗局就是时薪太低。针对这些社会痛点难点，下面将分小节介绍本项目提供的几个解决方案。

1.5.1 线下商家转型

通过群精选接龙平台，线下商家可以没有门槛的轻松开启线上业务，把线下卖货场景转移到线上，同时依托微信的社交属性形成自身的私域流量，免去平台的引流费用。相比微信提供的传统文字群接龙要求消费者打字下单、商家手工统计等不

便利性，团长可以利用本平台进行创建接龙、发布商品、处理订单、在线收款以及查看自动统计信息等，而用户则可以通过小程序浏览商品，一键选购，加入接龙。

1.5.2 商家资源对接

从平台建立至今，群精选接龙已积累了数十万用户基础，有超过两万的活跃团长留存。目前，平台运营者正积极寻找合作的供应商构建供应链，计划未来通过直播的形式向团长介绍时下热门商品，为团长提供货源的同时帮助供应商、大商家分销商品。

1.5.3 提供就业岗位

平台现正面向社区中的全职宝妈及白领等群体招募签约团长，由平台进行专业培训，帮助团长快速建立社群并有技巧地运营社群。为团长提供专业服务的同时，也为社区提供了新的就业岗位。

1.6 竞品分析

从目标用户角度出发，本产品的用户群与淘宝、京东等传统电商平台高度相似，都是为商家、供应商提供一个销售商品的平台，顾客可以通过平台选购商品完成交易。与传统电商平台中商家与顾客的松散连接不同，群精选接龙平台融入社交属性，商家通过微信群和线下交流维系与顾客间的关系，构筑自身的私域流量来代替平台引流，能够增加顾客黏性。

从商品交付方式出发，传统的电商商品配送为 B2C 配送，“最后一公里”的配送问题是物流成本高居不下的重要原因之一。而本产品采用 S2B 配送方式，将平台库存配送给物理位置稳定的各社区团长，由团长解决最后的配送问题，用较低的团长分成代替较高的第三方或自建物流费用，能够大大减少物流成本。

从社区团购模式出发，本产品依然采用熟悉的接龙方式销售商品。传统的运营方式为商家在社区群中吆喝，用户复制对话框接龙的方式选择商品，最后由商家手动统计订单，通过红包的方式完成支付。而本平台在接龙模式的基础上添加丰富 UI 与良好交互，方便商家创建商品接龙和发布，还有自动化管理库存等功能。顾客通过浏览接龙信息，对商品有直观了解后可以选择购买并一键支付。同时平台提供自动统计和订单数据导出等功能，方便商家管理配送。

2 产品设计

2.1 产品功能

2.1.1 首页

首页提供创建个人接龙的入口，如：“团购接龙”和“填表接龙”让用户能够以登录身份创建相应接龙；“复制接龙”提供优秀案例和历史接龙，用户可以选择其一做模板快速创建接龙。以个人登录或店铺身份创建的接龙都会以时间顺序排列并在首页显示。另外，首页还提供“团长教程”指导广大入门团长如何创建社群、快速卖货。



图

2-1 群精选接龙小程序首页功能

2.1.2 店铺

“店铺”页面允许用户创建并管理店铺。店铺内还有许多管理功能，如：通过“接龙管理”以店铺身份创建和编辑接龙；通过“订单管理”统计并一键导出订单；通过“商品库”便捷管理商品和库存；通过“自提点”保存常用的自提点地址；同时还可以查看店铺粉丝和邀请合作伙伴一同管理店铺。

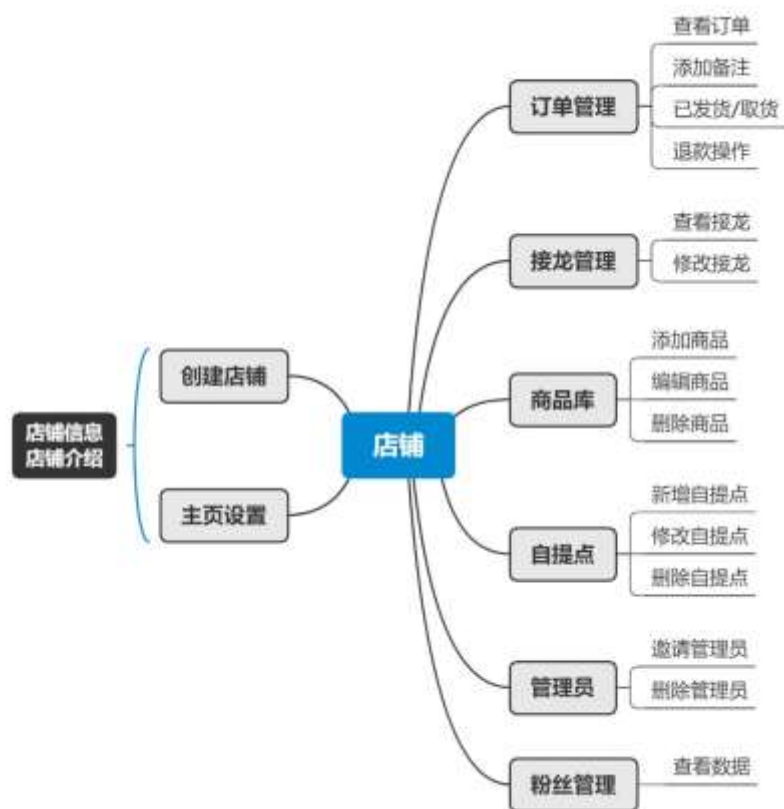


图 2-2 群精选接龙小程序“店铺”页面功能

2.1.3 我的

“我的”页面即用户中心，提供的功能有：通过“我的钱包”查看钱包余额、收支明细及提现；通过“已购买的订单”和“已卖出的订单”管理创建或接收到的订单；还可以快速查看发布或参与的接龙记录以及所关注的店铺。



图 2-3 群精选接龙小程序“我的”页面功能

2.2 交互设计

2.2.1 创建团购接龙流程

商家可以通过小程序创建团购接龙, 在接龙中添加团购商品的条目及详细介绍, 设置团购模式、参团人数及优惠金额等相关信息, 然后一键发布接龙, 分享到群聊。

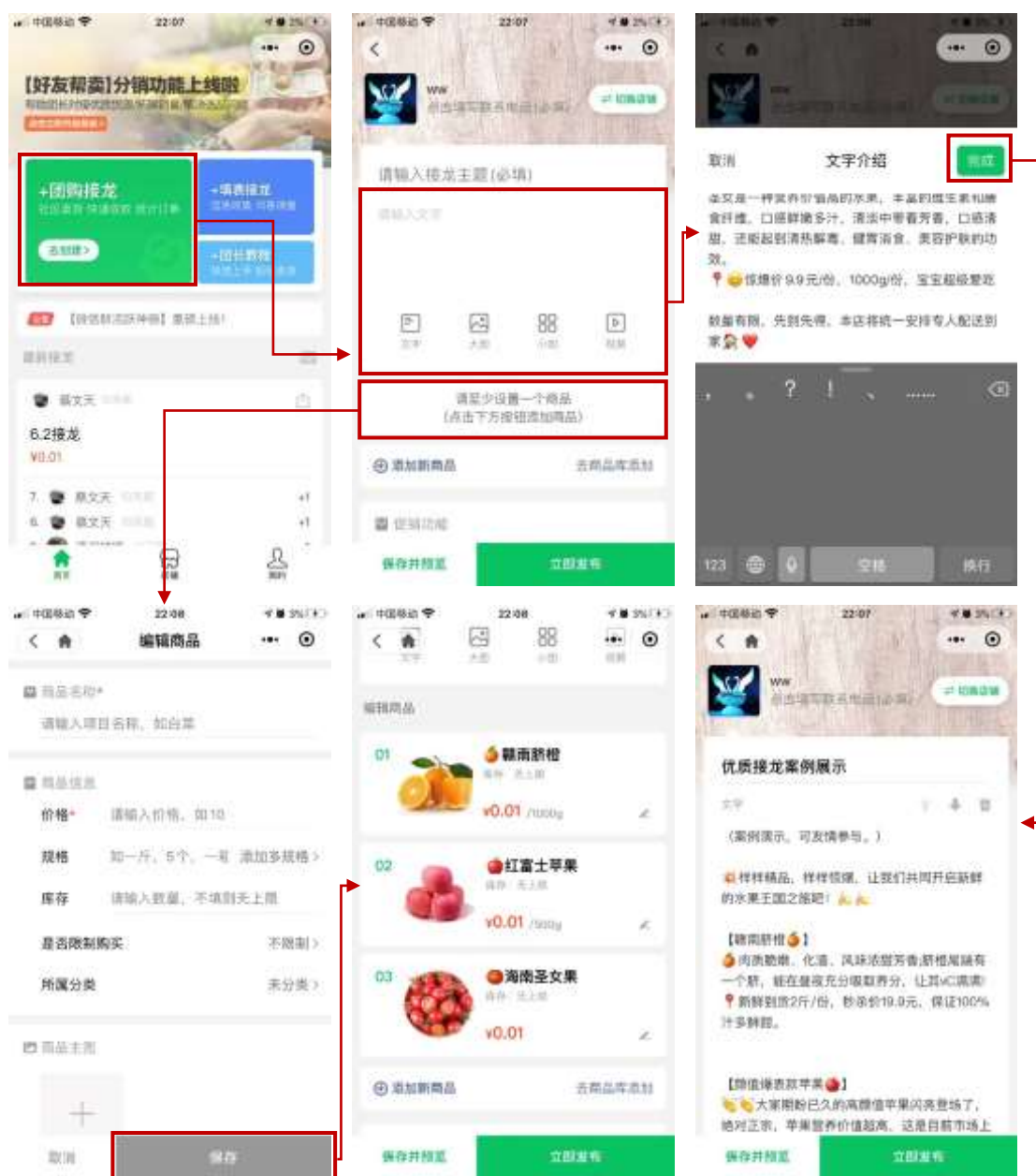




图 2-4 创建团购接龙流程

2.2.2 购物流程

在接龙发布后，顾客可以浏览商品，选择自己喜欢的商品加入购物车并完成付款；商家可以在后台查看订单信息并更新订单状态。



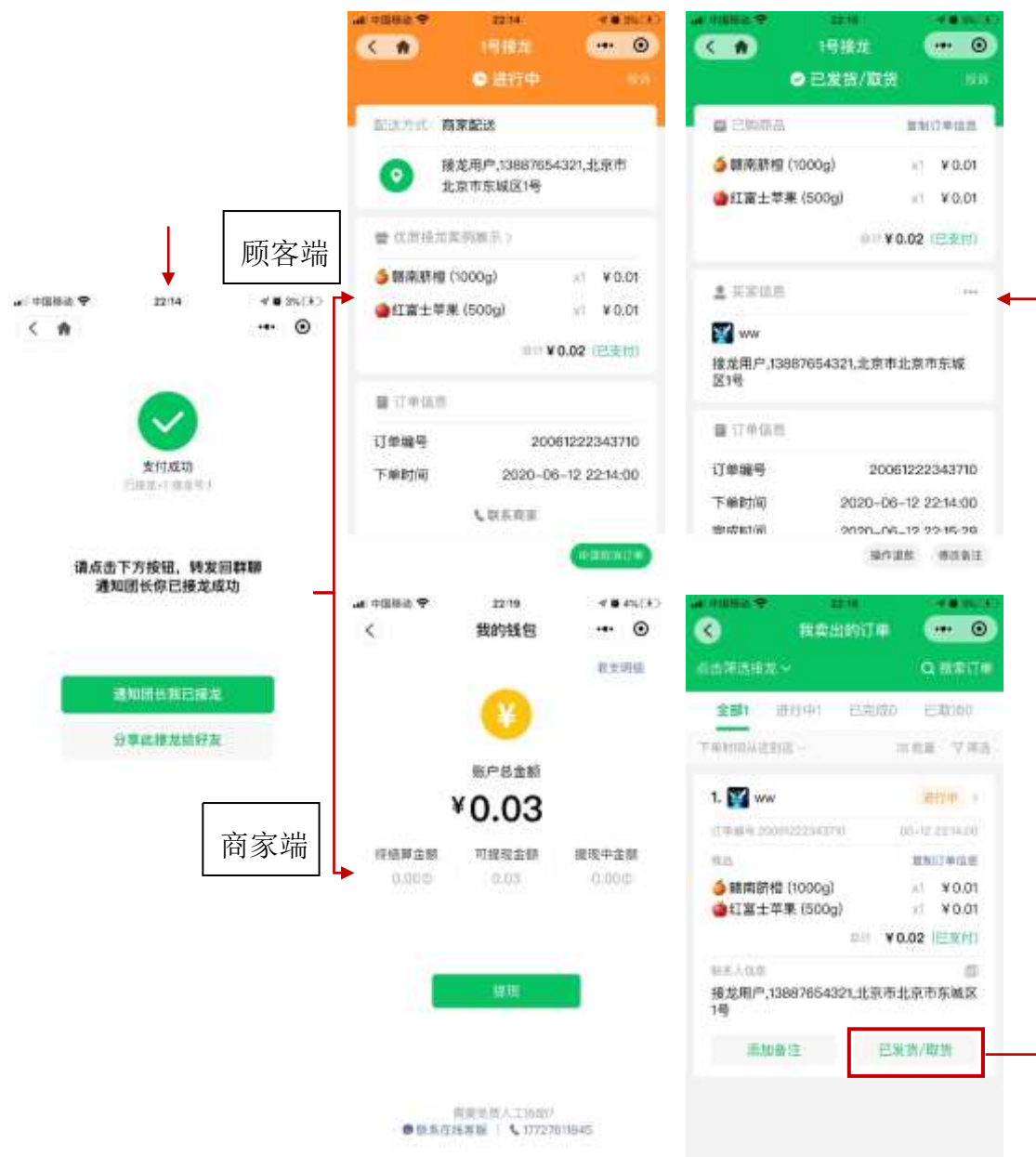


图 2-5 购物流程

2.3 运营方案

2.3.1 引流策略

在平台建立之初, 能否获取首批用户是重中之重, 为此构建了如下引流策略:

- ① 满足用户需求。针对 2020 年新冠疫情期间各行各业的数据统计需求, 群精选接龙允许用户根据应用场景自定义接龙填写表单项目, 待成员完成接

龙后能自动统计数据。由于功能设计合理，广受用户好评，在上线之初便迅速累计数十万用户；

- ② 社群分享。群接龙以分享的形式在微信群中流传，由于群里都是身边熟悉的街坊邻居，信任度较高，更容易地让用户点击与关注群精选接龙；
- ③ 公众号推广。平台拥有自己的微信公众号“群精选接龙”，由专业人士定期撰写发布团长教程，指导团长如何建立社群、科学卖货。通过公众号的推广，带领读者深入体验，让更多潜在用户接触群精选接龙。

2.3.2 存留策略

在吸引用户后，针对如何通过满足客户的需求，使用户继续访问平台更多的页面、增加在平台上的停留时间以及未来的再次使用，构建如下存留策略：

- ① 优化交互策略，提高用户体验。通过构建清晰的交互界面和更好的交互逻辑，同时提供选项，以点击的方式减少用户输入，为用户带来更加良好的体验；
- ② 优化推荐系统。通过用户画像分析用户的收藏店铺及购物喜好，再据此向团长推荐相关商品，使用户可以接触到更多的优质的合适的产品，从而提高用户对平台的满意度及黏性。

2.3.3 转化策略

用户存留不是最终目的，如何将用户转化才是关键。为了刺激用户转化，构建如下转化策略：

- ① 抓住顾客刚需。物美价廉的商品与可靠的渠道是消费者亘古不变的需求，平台提供的服务将极力满足这些需求；
- ② 提高服务质量。在适当的位置、页面给用户提供更多咨询的途径，增加平台与用户、用户与用户之间的交流，提高用户归属感；
- ③ 提供专属服务。为团长和供货商之间提供资源对接服务，以直播形式为团长带来时下热门爆款商品货源，同时为大商家配备 VIP 专属客服，全程在线服务；
- ④ 培养优质团长。为入门团长提供免费社群管理培训课程，帮助团长高效出

单。同时定期开展拼团接龙活动，助力团长提高销量。

2.3.4 优化策略

在运营一段时间后，对相关数据进行分析总结，对运营策略发现如下可调整优化的方面：

- ① 优化服务回报。分析运营数据，统计各项服务的投入产出比，找出回报大的投入方向。对于低回报的服务进行优化，如果优化后效果不好，减少成本或者放弃；
- ② 优化推广渠道。分析各个推广渠道，效果好的加大投入，效果差的减少投入或是放弃。

3 关键技术

3.1 客户端

3.1.1 小程序端

gulp.js 是一个基于流的自动化构建工具，主要用来设定程序自动处理静态资源的工作。为了能够更好的应用模块化、组件化思想，以工程化的方式管理小程序端开发，本项目采用 gulp.js 进行模块化打包和文件预处理，其中主要执行的自动化工作如下所示：

- ① Stylus 转换 WXSS。由于本项目没有调用第三方组件库，主要使用微信原生态组件，在构建定制化组件时需要编写大量复杂的样式。而相比于往往会导致许多重复性工作的原生 WXSS (CSS)，开发者更喜欢使用提供了变量、混写、嵌套、逻辑等拓展功能的 CSS 预处理器。因此，基于对开发效率和部署环境的考量，本项目采用 Stylus 进行样式编写，并通过在 gulp.js 中配置 gulp-stylus、gulp-
- ② rename 插件将 styl 文件转换成小程序端可以识别的 WXSS 文件；ES6 转换 ES5。虽然微信开发者工具提供了对 JavaScript 语言的 ES6 代码转换 ES5 代码功能，但 WXS 语言主要参考的是 ES5 标准，缺乏对 ES6 标准的支持。对于项目中使用了 ES6 语法的 WXS 代码，我们的解决方案是在 gulp.js 中配置 Babel 插件将 ES6 代码转换为 ES5 代码；
- ③ 文件压缩。官方对于程序大小的限制十分严格，要求小程序单个主包/分包大小不能超过 2M，整个小程序所有分包大小不超过 8M。因此，基于对大小限制和应用性能的考量，本项目通过在 gulp.js 中配置 gulp-uglify-es、gulp-cssso、gulp-jsonminify 等插件对文件进行压缩。

3.1.2 浏览器端

Kibana 是一个针对 Elasticsearch 的开源分析及可视化平台，用来搜索、查看交互存储在 Elasticsearch 索引中的数据。其提供了丰富多样的图表，如直方图、条形图、饼状图以及地图等，来进行高级的数据分析及展示。由于项目已实际上线

使用且目标用户体量大，对系统的运维工作不再是纸上谈兵。因此，为了更好地对系统进行监控、管理、维护等工作，本项目基于 Kibana 技术于浏览器端实现系统数据可视化。通过浏览器，运维人员可以更直观的理解系统中海量的日志、请求统计、响应时间统计以及系统健康度等数据。

3.2 服务端

3.2.1 开发框架

Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。本项目服务端采用社区生态更完善的 Java 语言进行开发，由于小程序端轻量、便捷、开发较快，为了维持两端同步推进，服务端使用不仅继承了

Spring 框架原有的优秀特性，而且更加轻量、配置简化的 Spring Boot 框架来快速开发，从而让项目能够早日上线，同时也为后期项目规模成型时搭建微服务架构打下基础。

3.2.2 部署环境

由于项目主要应用场景为社区购物，面对的商品大多是日常生活用品和生鲜食品等，具有需求频率高、购买时间集中等特点。这些特点对系统有着高并发、高性能、高可用等要求，因此，本项目拟采用微服务架构部署系统，其中主要使用的技术如下所示：

- ① Docker 是一个开源的应用容器引擎，可以让开发者将应用以及依赖包打包到一个轻量级、可移植的容器中。通过将系统中的功能模块封装成各种微服务，独立部署到不同的 Docker 容器中，降低各服务之间的耦合度，当某个服务出错停止运行时不会影响到其他服务的运行，从而提高系统的健壮性；
- ② Kubernetes 是自动化容器操作的开源平台，这些操作包括部署，调度和节点集群间扩展。使用 Kubernetes 管理系统中的容器集群，不仅可以实现当某个服务宕机时，快速的在新的容器中重新运行，提高系统的可用性，而且可以对项目具有的计算资源进行负载均衡，合理分配；

- ③ RocketMQ 是一款分布式、队列模型的消息中间件。使用 RocketMQ 构建消息队列,不仅可以通过发布-订阅的工作模式进一步降低各服务间的耦合性,而且可以通过异步处理降低系统响应时间和削平峰值流量,从而提高系统性能和高并发处理能力;
- ④ Ingress 是管理外部访问集群中服务的 API 对象,提供负载平衡、SSL 终端和基于名称的虚拟主机等功能。通过 Ingress 可以实现访问流量的负载均衡,提高系统用户体验。

4 系统设计

4.1 总体设计

4.1.1 架构设计

本系统的架构采用分层结构，总共划分为六层，其中：

- ① 数据层为系统提供数据支持，采用 MySQL 作为主要的数据库进行数据存储，以 Redis 作中间缓存提高数据读写效率，以 MongoDB 为 JSON 类数据存储提供支持，以 ElasticSearch 存储搜索日志数据；
- ② 支撑层为系统的设计开发提供支持，如使用 gulp.js 构建工具实现小程序端模块化组件化编程、服务接口采用 RESTful 设计风格、后端采用 Spring Boot 框架快速构建 Java 应用、使用 Swagger 工具自动化构建服务接口文档等；
- ③ 服务层囊括了系统运行过程中需要的一些公共基础服务，如使用 RocketMQ 构建的消息队列、Logstash 的日志收集解析、容器镜像、自动化部署、健康监控、链路追踪等服务；
- ④ 业务层包含了系统主要的业务服务，如接龙管理、订单服务、人物画像、用户服务、管理后台、消息发送等功能；
- ⑤ 网关层掌控系统流量入口，有使用 ingress 实现的流量负载均衡以及其他如权限鉴别、流量限制、协议转换、错误码统一、监控告警、服务发现等能交由网关统一执行的工作，使业务层关注于业务逻辑的实现。
- ⑥ 展示层是系统的界面展示，包含小程序端用户界面的实现、系统日志等数据的可视化以及后台管理的浏览器端界面。

具体的系统架构图如图4-1所示。

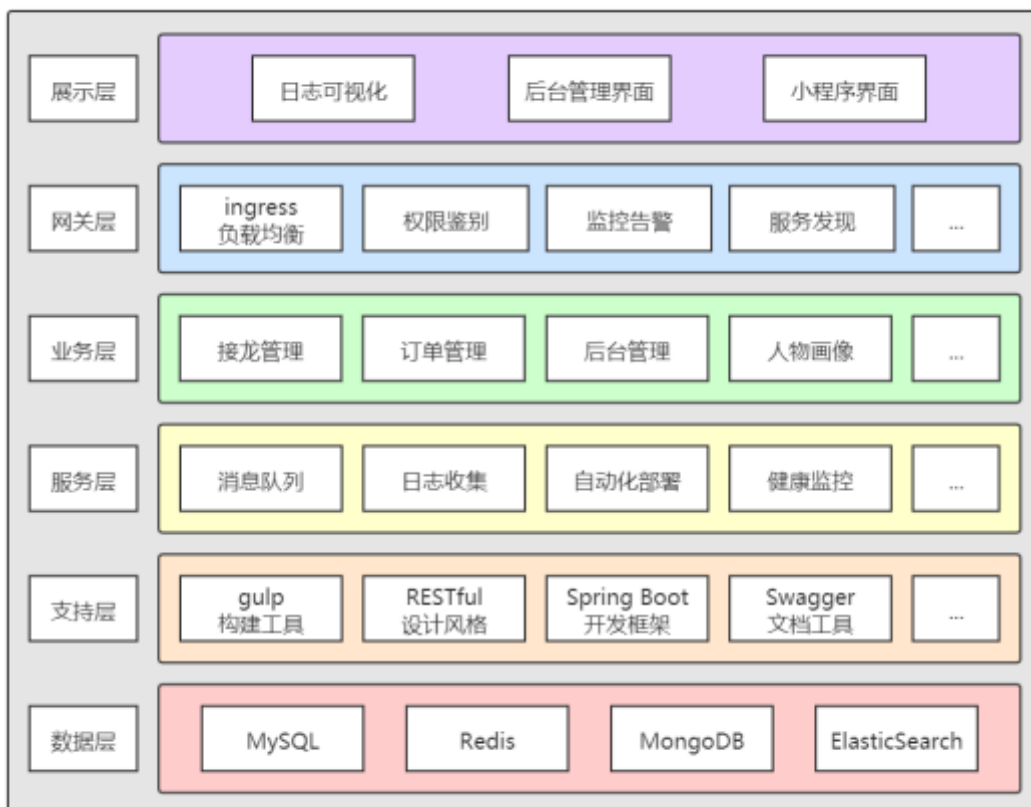


图 4-1 系统架构图

4.1.2 数据库设计

本系统早期构建的 E - R 图共具有十六个实体，分别是用户、接龙、文字描述、图片、视频、填表接龙、问题、填写类问题、选择类问题、图片类问题、团购接龙、店铺、商品、收货地址、自提点以及订单。同时该模型还描述了这十六个对象间的联系，比如用户与订单的关系是非强制性一对多的关系，一个用户可以没有或创建多个订单，但一个订单有且只能由一个用户创建。

具体的 E - R 图如下所示.

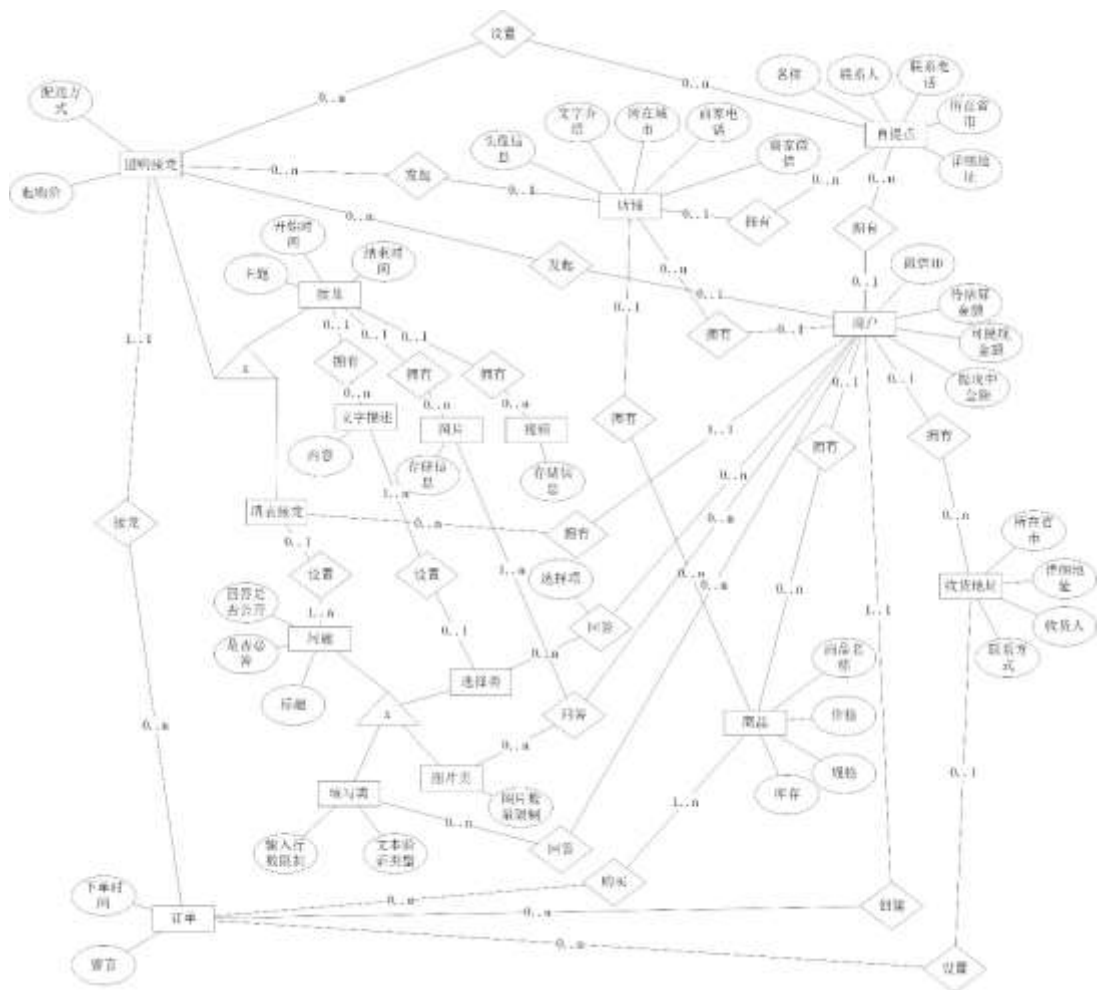


图 4-2 E-R 图

4.1.3 功能结构设计

为确保系统的可拓展性,对系统的功能结构设计应该采用模块划分的形式实现。
本产品的功能结构图如下所示:

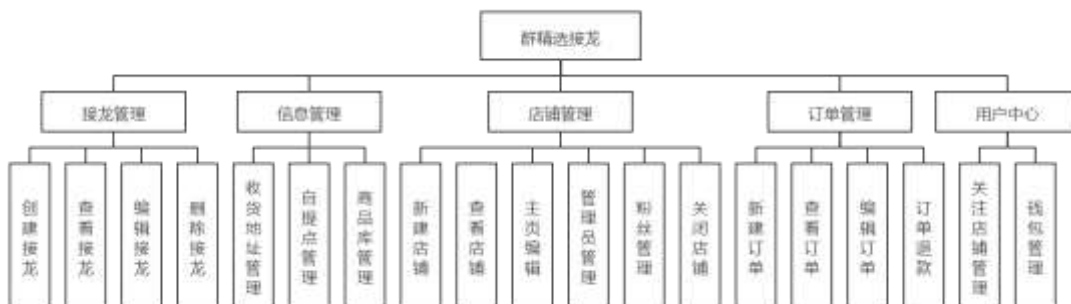


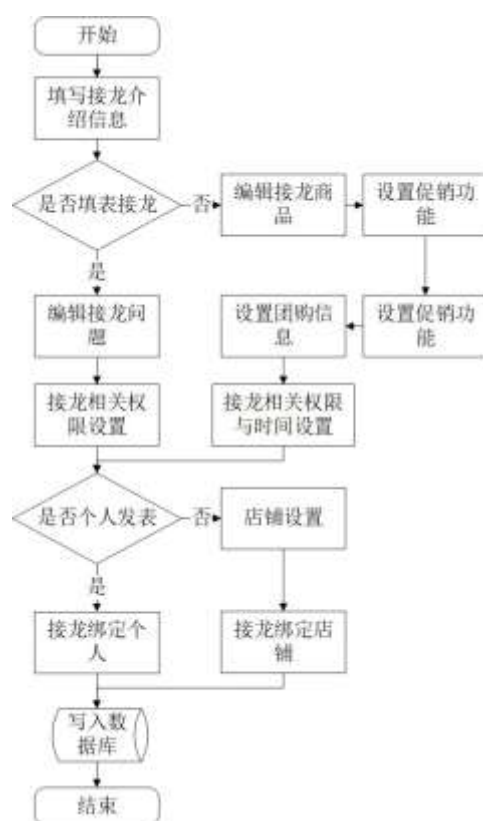
图 4-3 功能结构图

4.2 功能模块设计

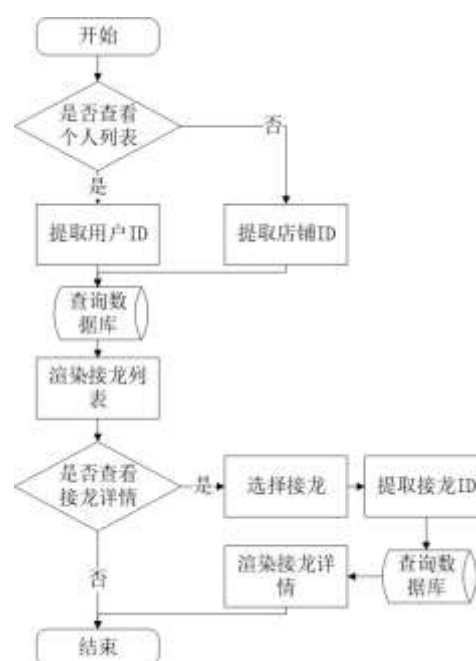
根据 4.1.3 小节的设计，将群精选接龙应用分成接龙管理、信息管理、店铺管理、订单管理以及用户中心五个模块，本节将对上述模块的设计与实现进行更深入的阐述说明。

4.2.1 接龙管理模块

接龙管理模块是接龙功能的基础，其中包含创建接龙、查看接龙、编辑接龙与删除接龙四个主要功能。接龙类型分为填表接龙和团购接龙，用户可根据需要发起接龙并分享到微信群中。群里的其他用户可以点击链接查看接龙详情并进行表单填写、购买商品等操作。发起者可以对接龙的内容或状态进行修改，同时还提供删除功能。在接龙的过程中发布者可以看到一些实时的统计数据。

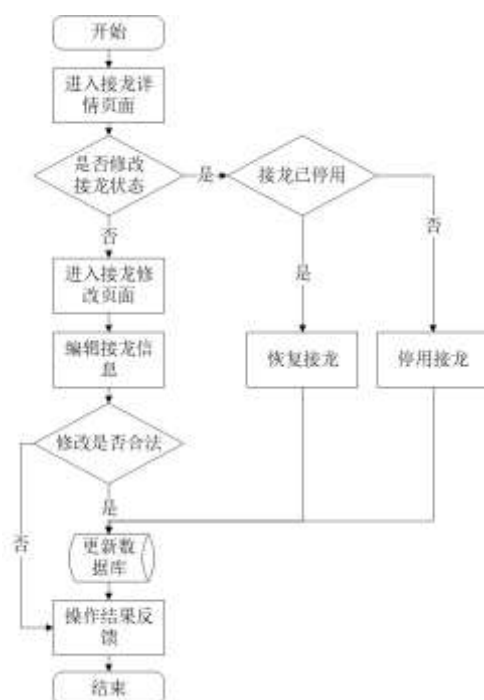


创建接龙

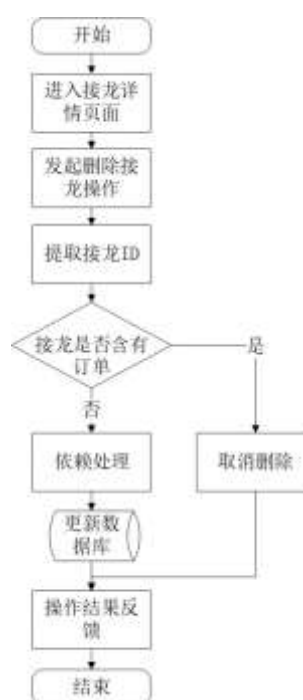


b) 查看接龙

a)



c) 编辑接龙



d) 删除接龙图

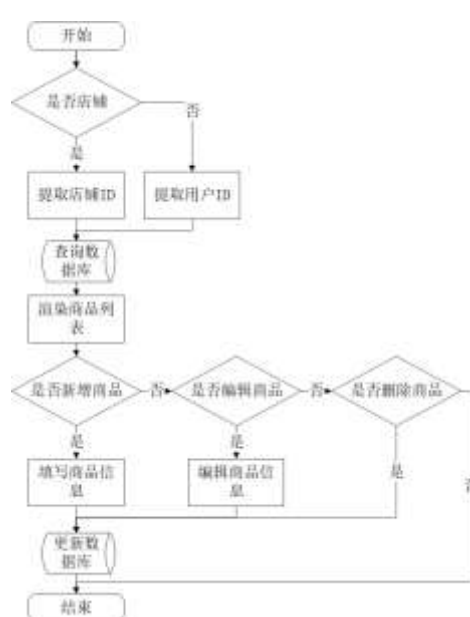
4-4 接龙管理模块流程设计

4.2.2 信息管理模块

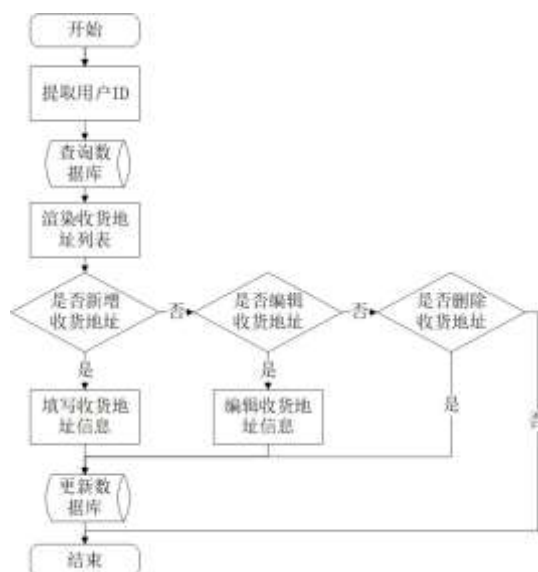
信息管理模块主要用于管理个人或店铺常用的信息。其中，个人和各店铺都拥有独立的自提点与商品库，可通过自提点管理和商品库管理功能存储常用信息。收货地址管理则让用户可以存储常用的收获地址。



a) 自提点管理



b) 商品库管理

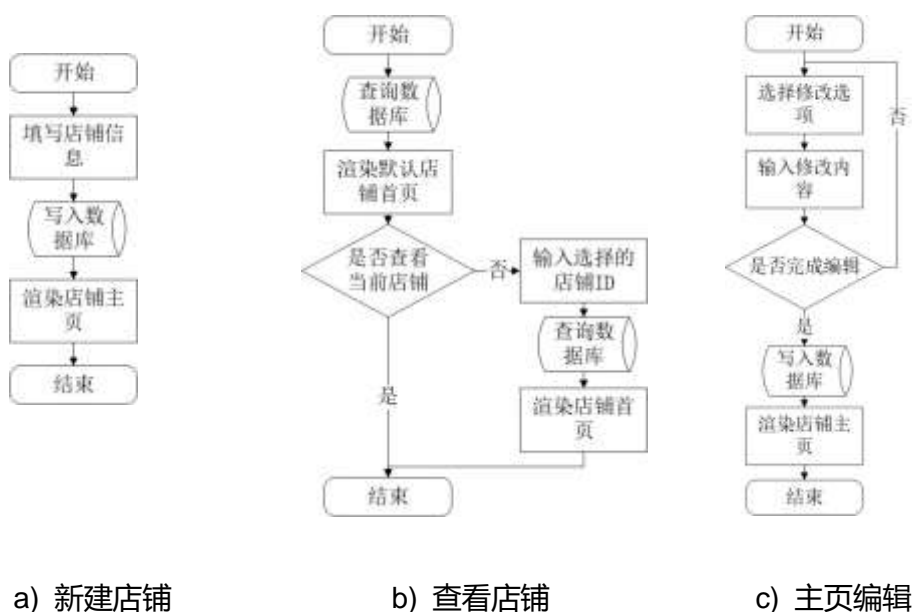


c) 收货地址管理

图 4-5 信息管理模块流程设计

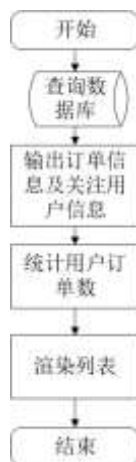
4.2.3 店铺管理模块

店铺管理模块中，可以操作的部分有创建店铺、查看店铺、粉丝管理、管理员管理及关闭店铺等，商家可以通过创建店铺进行更专业的管理。通过主页编辑模块商家可以编辑展示给用户的相关信息及介绍等内容。后台汇集的多个服务模块帮助商家进行订单管理、接龙管理、商品管理等业务。管理员模块允许店主将其他人列为管理员，共同管理店铺。粉丝管理自动统计店铺的活跃用户及粉丝，方便管理员进行管理及追踪。





d) 管理员管理



e) 粉丝管理图



f) 删除店铺

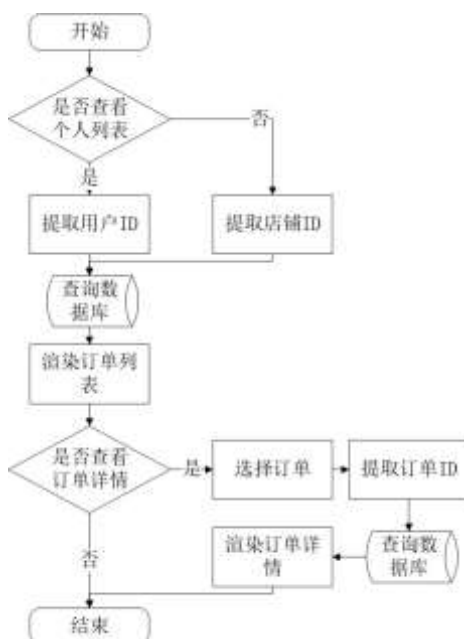
图 4-6 店铺管理模块流程设计

4.2.4 订单管理模块

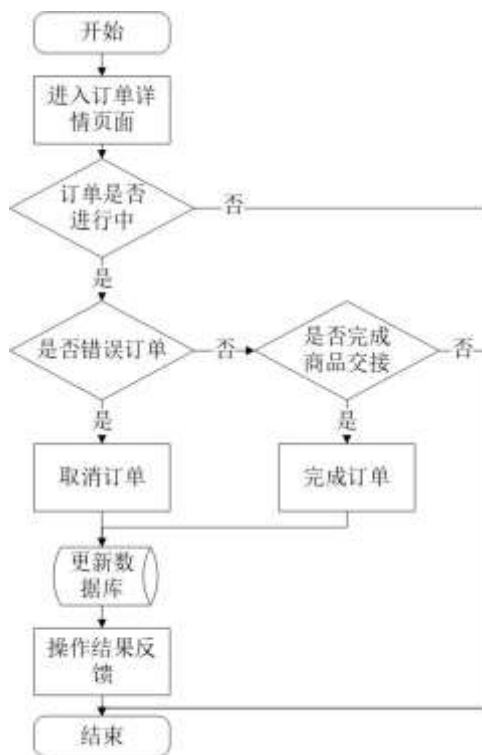
订单管理用于团购接龙产生的订单。顾客在浏览团购接龙时选择喜欢的商品并完成支付后即可创建订单。查看订单功能供消费者查看创建的订单或供团长查看个人、店铺接收到的订单。编辑订单可供发生错误购买的情况下取消订单，或是在商品交接后完成订单。当商家无法完成订单时可选择将款项退回给顾客。同时还为团长提供导出订单记录功能，以 Excel 表格形式展现主干信息，提高团长发货效率。



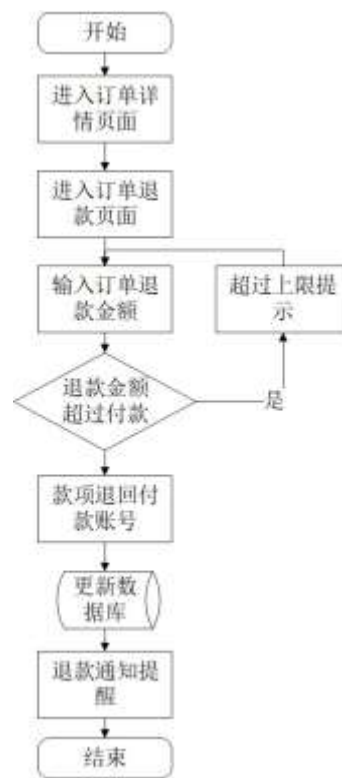
a) 创建订单



b) 查看订单



c) 编辑订单

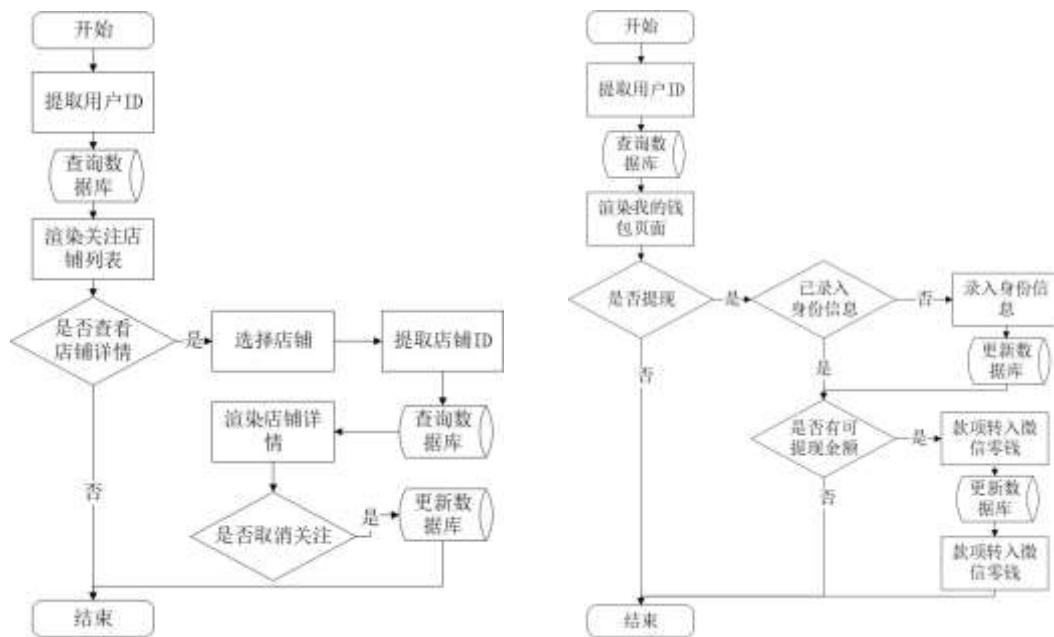


d) 订单退款图

4-7 订单管理模块流程设计

4.2.5 用户中心模块

用户中心可以查看账号相关信息。钱包管理帮助用户了解当前钱包中含有的待结算金额、可体现金额以及体现中金额，同时还提供收支明细查看。对于团长卖货的收入可以在我的钱包进行提现操作，将存放在平台上的款项转移到微信零钱中。用户在团购接龙过程中关注的店铺会展示在用户中心，方便用户追踪店铺新的接龙动态。个人页面还包含了查看接龙记录、购买订单、卖出订单等信息的入口。



a) 关注店铺管理

b) 钱包管理

图 4-8 用户中心模块流程设计

5 系统测试

5.1 测试环境

在进行测试工作前，应该确认并记录运行测试的环境，根据系统所需环境和测试工作计划配置合适的环境是能够顺利进行测试的保障，相同的环境也是再次获得相似测试结果的前提。本项目选取的测试环境中配置了多台服务器主机以及一台手机，这能在一定程度上模拟系统的真实运行情况。测试环境的具体参数如下所示。

表5-1 客户端测试环境

操作系统	iOS 13.4.1
内存	256GB
处理器	A11 芯片
主要测试工具	WeChat v7.0.12
辅助测试工具	微信开发者工具

表5-2 服务端测试环境

操作系统	CentOS 7.2
内存	8GB*6+2GB*1
处理器	双核*6+单核*1
数据库管理系统	MySQL 5.7、MongoDB

5.2 测试流程与方法

本系统测试的主要流程为：先提出测试计划，对测试计划进行核查可行性，通过审核后设计相应的测试用例，搭建测试环境，根据测试用例手动进行单元测试和集成测试，根据测试结果对系统进行完善与优化的同时将结果记录在案并整理成文档记录，方便后续查看。

系统测试工作流程图如下所示。

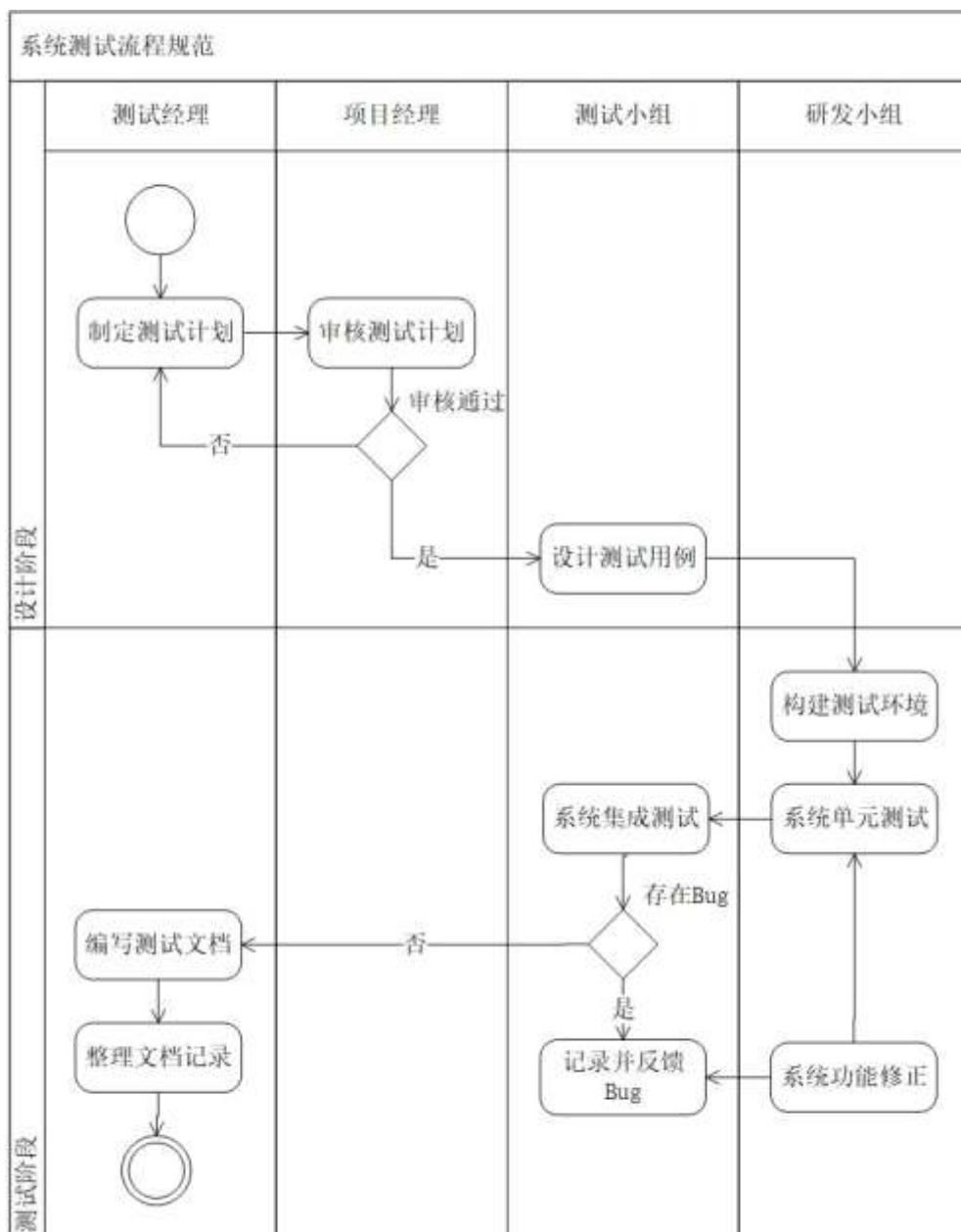


图5-1 测试流程图

系统功能测试方法一般有单元测试和集成测试。其中，单元测试是指研发人员在进行代码编写对最小可测元素的测试，一般是对单个函数或者单个模块进行测试检验。本系统在开发过程中主要使用 Junit 对后端函数进行单元测试。

当系统完成一定程度的开发，各模块间有一定的关联，对已通过单元测试的模块进行组合测试被称为集成测试。本系统在开发过程中主要使用实体机小程序端进行集成测试。通过这两种测试能够评估系统实现的正确性。

5.3 测试结果

5.3.1 功能性测试

本小节以对接龙管理模块的测试为例子阐述实际的系统功能性测试,其测试用例如下所示。

表5-3 接龙管理模块测试用例

设计目的	检验接龙管理模块各项功能是否达到预期效果			
测试指导	关于接龙管理的需求分析			
测试数据	接龙信息			
功能名称	步骤描述	前置条件	预期结果	测试状态
查看接龙	1.测试人员进首页; 2.测试人员点击接龙列表中某条接龙信息。	微信用户授权登录; 用户曾经创建接龙且没有删除	1.测试人员能看到自身创建所有的接龙; 2. 测试人员能看到接龙的详细信息。	测试通过
编辑接龙	3.测试人员在接龙详情点击“接龙设置”-“修改接龙内容”; 4.测试人员修改信息后点击“完成修改”-“立即恢复”。	微信用户授权登录; 用户曾经创建接龙且没有删除	3.当前接龙状态变为状态,页面跳转到接龙编辑页面; 4.页面跳转到接龙详情页面且相应信息已变更,同时接龙状态从暂停中恢复。	测试通过
删除接龙	5. 测试人员在接龙详情点击“接龙设置”-“删除接龙”-“确定”	微信用户授权登录; 用户曾经创建接龙且没有删除;测试的接龙没有订单关联	5.页面跳转回首页且测试的接龙已从接龙列表中消失	测试通过

创建接龙	1.测试人员在首页点击“+ 团购接龙”； 2. 测试人员填写相应信息后点击“立即发布”； 3.测试人员点击返回按钮	微信用户授权登录	1.页面跳转到创建团购接龙页面； 2.页面跳转到接龙详情页面,具体内容与填写的内容吻合； 3.首页接龙列表顶端出现刚新增的接龙。	测试通过
------	---	----------	--	------

对于系统中其他模块也编写了测试用例并进行相应的测试。通过一系列测试后，可以认为实现的系统已达到预期目标。

5.3.2 服务端性能测试

通过Kibana上可视化的统计数据图对服务端性能进行评估，具体分析如下：

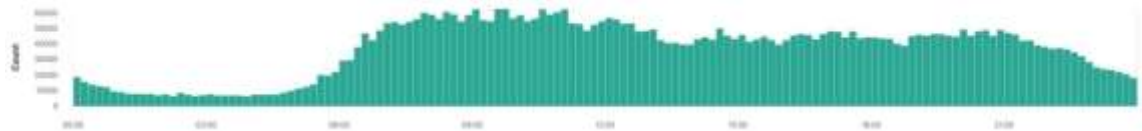


图5-2 时间-访问量统计图

从图 5-2 可以看出，横轴时间刻度为十分钟一小格，系统访问高峰时间段为早上七点至晚上九点，目前系统至少可承受接近六万的巅峰访问量，对于具有十万注册用户的现状来说，这样的性能可以满足当前需求。



图 5-3 时间-最大响应时间统计图



图 5-4 时间-平均响应时间统计图

图 5-3、图 5-4 分别统计了单位时间内的最大响应时间和平均响应时间。从

图中可以看出，最大响应时间没有超过 0.6s，平均响应时间则在 0.1s 内，算上网络传输延迟时间，对用户操作的响应时间能保持在 2~3 秒以内，能够带来良好的用户体验。

5.3.3 小程序端性能测试



a) 概况



b) 运行性能



c) 启动性能



d) 网络性能图

5-5 微信开发者工具性能测试图

图 5-5 展示了微信开发者工具自带的性能测试。从图中可以看出，群精选接龙小程序在 ios 设备上的初启动耗时 1.8s，在 android 设备上的初启动耗时 3.6s，网络请求的平均耗时为 0.5s，基本上渲染和响应时间能保持在 3s 以内，且不占用过多系统内存，符合当下基本的用户体验要求。