基于微信小程序的进存销管理系统的设计与实现

摘要

尽管市面上已经有许多成熟的进存销管理系统，但是作为小型的商家，为此专门开发一个系统是不太现实的，且还需购买相应的硬件设备。为了使这些商家也能够使用进存销系统进行商店的管理，通过微信小程序无需下载、即开即用的特点使得进存销系统在手机上运行，让每一个商家都能够无门槛使用，即是本系统要解决的主要问题。

本文从本系统开发等背景开始，以系统功能模块以及软件设计作为主要陈述内容，包含了需求分析，总体设计以及详细设计三大部分。系统后台采用nginx作为服务器，mongodb作为数据库，利用express框架作为后端应用框架。前台开发使用微信开发者工具，前端使用与微信原生系统一致的基础样式库(WeUI)使得用户体验更加统一。经过了设计、实现与测试，本系统已基本完成系统的预期功能。

关键词：进存销系统；微信小程序；mongodb；nginx

DESIGN AND IMPLEMENTATION OF MINI PROGRAM – BASED STOCKING SELLING AND STORING SYSTEM

ABSTRACT

Although there have been many mature stocking selling and storing system (SSSS), however, as a small business, it is not realistic to develop a system for this purpose, and the corresponding hardware devices should be purchased. In order to enable these merchants to use the store management system, the store management system can be run on the mobile phone through WeChat small program without downloading. It is the main problem to be solved in this system that every merchant can use it without any threshold.

This paper starts from the background of the system development and takes the system function module and software design as the main presentation content, including three parts: requirement analysis, system design and detailed design. Nginx is used as the server in the background of the system, mongodb as the database and Express framework as the back-end application framework. Front-end development uses the WeChat developer tool, and the front-end uses the basic style library (WeUI) consistent with WeChat's native system to make the user experience more unified. After design, implementation and testing, the system has basically completed the expected functions of the system.

Key words: stocking selling and storing system ; Mini Program; mongodb; nginx

目 录

[1 绪论 1](#_Toc517388477)

[1.1课题研究的背景 1](#_Toc517388478)

[1.2课题的研究意义 2](#_Toc517388479)

[1.3 国内外现状与发展趋势 2](#_Toc517388480)

[1.4 论文结构和内容 3](#_Toc517388481)

[2 相关技术简介 4](#_Toc517388482)

[2.1 Mini Program 4](#_Toc517388483)

[2.2 nginx 4](#_Toc517388484)

[2.3 node.js 5](#_Toc517388485)

[2.4 mongodb 6](#_Toc517388486)

[3 需求分析 8](#_Toc517388487)

[3.1 设计目标 8](#_Toc517388488)

[3.2 可行性分析 8](#_Toc517388489)

[3.2.1经济可行性 8](#_Toc517388490)

[3.2.2技术可行性 9](#_Toc517388491)

[3.2.3操作可行性 9](#_Toc517388492)

[3.2.4其他可行性 9](#_Toc517388493)

[3.3 性能需求 9](#_Toc517388494)

[3.4 功能需求 10](#_Toc517388495)

[3.4.1商品入货管理 10](#_Toc517388496)

[3.4.2 商品管理 10](#_Toc517388497)

[3.4.3 商品销售 10](#_Toc517388498)

[3.4.4生成报表 11](#_Toc517388499)

[3.4.5供应商管理 11](#_Toc517388500)

[3.5 需求分析总结 11](#_Toc517388501)

[4 总体设计 12](#_Toc517388502)

[4.1 概要设计 12](#_Toc517388503)

[4.2 数据库设计 13](#_Toc517388504)

[4.2.1 数据流图 13](#_Toc517388505)

[4.2.3 数据库结构设计 14](#_Toc517388506)

[4.2.3 数据库表结构设计 18](#_Toc517388507)

[5 详细设计与系统实现 23](#_Toc517388508)

[5.1 开发运行环境 23](#_Toc517388509)

[5.2 系统流程图 23](#_Toc517388510)

[5.3 商店管理功能实现 25](#_Toc517388511)

[5.4 库存管理功能实现 27](#_Toc517388512)

[5.5 商店销售功能实现 29](#_Toc517388513)

[5.6 商店报表功能实现 30](#_Toc517388514)

[6 测试 32](#_Toc517388515)

[6.1 测试目的 32](#_Toc517388516)

[6.2 测试用例 33](#_Toc517388517)

[6.2.1 扫码入货界面测试 33](#_Toc517388518)

[6.2.2 库存管理页面测试 33](#_Toc517388519)

[6.3 测试结果分析 34](#_Toc517388520)

[参考文献 35](#_Toc517388521)

[致谢 36](#_Toc517388522)

[附录A 部分源代码 37](#_Toc517388523)

[附件：开题报告](#_Toc517388524)

[附件：译文及原文影印件](#_Toc517388525)

# 绪论

这十年来，互联网以及移动支付的发展已经极大地改变了我们的日常生活。尤其是在电子商务方面，改变了以往购物信息不对称以及时空限制的难题，使得人们可以足不出户，享受全球好货。实体零售业也因此收到了很大的冲击，传统零售业的销售额的增速在过去几年出现了明显下降，其中百货超市的销售额甚至在 2015 年开始出现持续的负增长[1]。在传统零售行业整体一片哀嚎的背景下，小型商店的销售额增长速度却在过去七年稳定保持在九个点以上。

## 1.1课题研究的背景

互联网的蓬勃发展催生了许多新型的商业模式，其中影响范围最为深远的便是电子商务。在过去的三十多年里，中国零售行业的发展大致上经历了五个阶段。2000以前的百货商场，2000年开始兴起的综合购物中心模式。这两种模式直到2008年左右都是一片欣欣向荣的景象，直到金融危机波及到传统零售业和电商的迅速崛起，使得电商逐步蚕食传统零售业的份额，甚至有人说：“传统零售的的寒冬已经到来”。首先受到波及的是服装家居用品，后来是家电行业，再后来便是快消品行业，与此同时，装修建材在线上的销售额也越来越大，对线下的打击是逐个品类被击破。去掉汽车行业和加油站，线上交易已经抢走了整个行业的20%交易额。

尽管实体零售业整体的发展显现出疲软的态势，但便利店等小型商店等业态表现却很抢眼。在过去7年中，便利店行业销售增速稳定在9%及以上水平。根据商务部公布的《中国便利店景气指数报告》，便利店景气指数为72.22，远高于50.0荣枯线，总体显现出健康的发展态势。究其原因，主要有以下几点：一是便利店选址要求通常不高，可以让其更容易渗入到居民区附近；二是消费人群大都是在工作间隙或移动途中选择进店购买，只需花费零碎时间便可购买到自己所需要的物品；三是我国城市人口密度大，为其发展提供了人口基础。

虽然我们街头没隔不远就能看见便利店，但是其中连锁的便利店相比之下并不多，在中国大陆地区，连锁便利店目前仅有上海和广东的门店超过5000家，因此小卖部之类的小型商店仍然占据了主要部分，其为我们生活所提供的便利是不言而喻的。

## 1.2课题的研究意义

在信息化时代，传统的手工记账方式已经越来越不能够满足当代社会的的便捷、可靠以及对数据价值挖掘的需求。绝大多数超市或者连锁的便利店都拥有自己的进存销管理系统。而作为这个行业主力的社区小卖部这些个人商店，却不能够享受到信息化时代的便利。因此，为这些商户开发一款不需要门槛就能够使用的进存销管理系统是很有必要的。当今移动互联网发展规模不断扩大，在《第38次中国互 联网络发展状况统计报告》中显示，截至2016年6月，我国手机网民规模达6.56亿，网民中使用手机上网的人群占比由 2015年底的90.1%提升至92.5%，仅通过手机上网的网民占比 达到24.5%，网民上网设备进一步向移动端集中[2]。本系统基本上能够运行在目前几乎所有主流的手机上，使得商户们触手可及便可轻松管理自己的商店，助力其更好地发展，让其更好的服务社区居民，为居民持续提供优质、便利的生活。

## 1.3 国内外现状与发展趋势

中国便利店行业集中度很低，究其原因，主要原因有以下三个方面：仍然处于发展的初级阶段、我国区域经济发展不平衡和文化差异大。从目前连锁便利店分布状况我们可以得知，只有粤、沪地区的便利商店超过5000家，而大部分地区的总数却不超过1000。因此，不同地区不平衡的发展为便利店发展提供巨大潜力。相比于便利店企业，社区小型便利店的分布却是已经深入到千家万户，承担起了即时购物的主要载体。通过研究其他国家便利店行业的发展进程我们可以发现，其发展状况与经济发展和城市化水平密切相关，因此可以结合其他国家的发展进程与和目前我国的发展情况进行对比和分析。根据研究，在一个国家的城市区域，当人均国内生产总值达到3000美元时，城市的便利店就有可能建立并维持下去的可能性;当人均国内生产总值超过6000美元时，城市的便利店就进入了增长阶段。国内一线城市平均每3000人就可以养活一个便利店，其辐射范围通常半径仅为500米[3]。就比如上世纪末的日本一样，人均国内生产总值3000美元、7-11、罗森、全家等便利店巨头在日本市场开始崛起。随着其人均国内生产总值和城市化水平增长，日本便利店行业进入了一个长期稳定的增长阶段。中国这个行业的早期发展历程与日本类似，随着人均国内生产总值的不断提高，我国在上世纪九十年代出现了第一家便利店，到了 2005 年前后，国外大型连锁便利店开始大规模进驻中国。目前，伴随着我国城市化水平的不断提高，中国的人均国民生产总值也已经超过了日本在便利店发展初期的水平。而且，在可以预见的未来里，虽然我国经济增速继续放缓，但温和增长的势头不会改变，对中国便利店行业进一步施展拳脚也将起到很好的促进作用。

## 1.4 论文结构和内容

第一章，结合电子商务和传统零售的区别与联系，介绍传统零售的发展背景、课题研究意、以及在国内外现状与发展趋势。

第二章，介绍本次项目所运用到的主要、核心技术，以及他们各自的特点。

第三章，利用场景分析等软件工程需求分析方法[9]，进行项目各个功能模块的进行需求分析。

第四章，通过E-R图、数据流图等图表与文字来陈述本次总体设计和数据库设计。

第五章，详细叙述本系统各个功能模块具体的功能以及实现结果。

# 2 相关技术简介

本次系统开发前后端均采用JavaScript作为开发语言，后端js运行环境基于Node.js(v8.8.4)，采用Nginx1.14作为后端服务器，系统运行于微信6.6.7及以上版本。

## 2.1 Mini Program

Mini Program即是小程序，是微信在2016年9月发布的一款“轻应用”入口。早在2013年，baidu便提出了“轻应用”这个观点：它既有乃至超越原生应用的用户体验，又具有Web app的可被检索与智能分发的特点，将有效解决优质应用和服务与移动用户需求对接的问题。

微信将小程序视为一组非常轻巧的工具 - 与普通应用程序相比 - 可以触及用户，并将它们与真实世界（以及它背后的信息）连接起来。换句话说，微信正在将功能扩展到更多的操作系统，微信内的微型程序成为用户的日常应用程序。小程序的目的是希望成为一个真正的平台，使得各个行业都可以从中受益。其具有以下优点：

1. 不需要安装，用户需要使用时打开即可，不用占用系统空间；
2. 由于使用微信作为平台，因此不需要开发ios与Android两个版本，降低了开发的成本；
3. 应用响应速度接近原生应用；
4. 支持许多功能强大且调用方便的API。

## 2.2 nginx

Nginx (engine x) 是一个高性能的HTTP和反向代理服务器，也是一个IMAP/POP3/SMTP服务器[4]。Nginx是由ИгорьСысоев为俄罗斯访问量第二的Rambler.ru站点开发的，初始版本0.1.0在2004年10月4日发布。

作者将其源码以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而闻名。2011年6月1日，nginx 1.0.4发布。

Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。其特点是占用内存少，支持高并发，且在实际应用环境中，nginx的并发能力确实在同类型的服务器中性能更加优异，中国大陆使用nginx网站大型网站有：baidu、京东、新浪、网易、腾讯、淘宝等。

相对于apache服务器来说，nginx服务器具有以下优点：

1. 轻量级，同样起web 服务，比apache 占用更少的内存及资源；
2. 抗并发，nginx 处理请求是异步非阻塞的，而apache 则是阻塞型的，在高并发下nginx 能保持低资源低消耗高性能；
3. 高度模块化的设计，编写模块相对简单；
4. Nginx支持7层负载均衡；
5. Nginx支持https[10]

## 2.3 node.js

Node.js—也称为Node, 是一个服务器端的JavaScript环境(参见http://nodejs.org)。它基于谷歌的一种运行时实现—准确的来说应该叫做“V8”引擎。V8和Node主要是通过C和c++中实现的，其首要目标是性能和低内存消耗。但是，VB在浏览器中主要支持JavaScript(最明显的就是谷歌的Chrome)， Node旨在支持长时间运行的服务器进程。

与其他大多数现代程序运行环境不同，Node进程不依赖多线程来支持并发来完成业务逻辑;它基于异步I/O事件模型，将Node服务器进程看作是一个单线程的守护进程，这个守护进程能够定制化将JavaScript引擎嵌入到其中。这与其他编程语言的大多数事件服务器不同，它们以库的形式出现，Node支持语言层面的事件模型。

JavaScript非常适合这种方法，因为它支持事件回调。例如，当浏览器完全加载文档时，用户单击一个按钮，或实现Ajax请求，事件触发回调。JavaScript的功能特性使得创建可以作为事件处理程序注册的匿名函数对象变得非常容易。

## 2.4 mongodb

MongoDB是一个免费的、开源的、面向文档的跨平台数据库程序。MongoDB将其归类为一个NoSQL数据库程序，它使用具有模式的json类文档。MongoDB是由MongoDB公司开发的，在GNU Affero通用公共许可证和Apache许可的结合下发布的。

MongoDB支持字段、范围查询、正则表达式搜索。查询可以返回特定的文档字段，还包括用户定义的JavaScript函数。查询还可以配置为返回给定大小的结果的随机样本。除此之外，它还具有以下特点：

1. MongoDB文档中的字段可以用主索引和辅助索引进行索引。
2. MongoDB提供了副本集的高可用性。一个复制集由两个或多个数据副本组成。每个复制集成员可以在任何时候扮演主副本或次副本的角色。默认情况下，所有的写和读操作都在主副本上完成。次要副本使用内置复制来维护主副本的数据。
3. MongoDB使用分片水平伸缩。用户选择shard键，该键确定集合中的数据将如何分布。数据被分割成范围(基于碎片键)并分布在多个碎片上。
4. MongoDB可以在多个服务器上运行，平衡负载或重复数据，以便在出现硬件故障时保持系统运行。
5. MongoDB可以作为一个文件系统，在多台机器上具有负载平衡和数据复制特性，用于存储文件。
6. 聚合框架允许用户获取使用SQL GROUP BY子句的结果类型。聚合操作符可以串在一起形成管道—类似于Unix管道。聚合框架包括可以连接来自多个文档的文档的$lookup操作符，以及标准偏差等统计操作符。
7. MongoDB支持称为上限集合的固定大小集合。这种类型的集合保持插入顺序，一旦达到指定的大小，就像一个循环队列。

# 需求分析

## 3.1 设计目标

因为目前市面上并未找到同类的产品，因此本系统设计的目标主要参考本系统在PC端的实现。首先要满足进存销系统的所有基本功能，如进货管理，商品管理，产品销售等；其次在软件设计的时候应该充分考虑到用户的使用体验，尽可能完备地考虑到用户的需求；最后，由于本系统仍然是一个信息管理系统，本系统应该对数据的安全性做考虑。

## 3.2 可行性分析

从系统开发的经济、技术、社会、法律以及其他方面，分析系统开发的可行性，综合各个因素，整体评估系统是否能在现有的条件上得到实现，并且是否合算，从而确定整个系统的可行性。

### 3.2.1经济可行性

由于现阶段市场上同类产品并不多，且随着信息化时代技术的发展，无纸质办公已经越来越得到普及，加之数字化的数据能够更好的服务大数据时代。因此本系统不仅可以代替商家完成从纸质记录到电子记录到转化，还能够通过数据分析为商家直观的显示出商店日常到运行状况，以便其调整经营策略。从而，为小商店带来更多的销售利润。而且当下正值小便利店高速发展的阶段，对类似系统需求丰富比较大，即使不销售软件，也可以为商家提供一些增值服务。相关开发技术人员设计开发费用较低，后期维护费用较低，在经济上，此系统的开发是可行的。

### 3.2.2技术可行性

虽然微信小程序是近两年才兴起的技术，直到2017年才开始公测，但是其相关技术均已经发展了许多年。尽管没有相关项目的支撑，但是使用同类技术的其他产品市场上已经可以给到许多的参考案例。如node.js和mongodb，经过了这么多年的发展，已经在市场上占据了不小的份额，且其未来的发展趋势也非常乐观，加之其易维护的特点，因此在技术上，此系统的开发是可行的。

### 3.2.3操作可行性

系统以普通便利店店主为使用对象，作为一个以微信作为入口平台的小程序，其UI风格以及使用的方式都将与微信类似，尽量减少两者转换过程中给用户带来的不适感。所有即时操作（如打开扫码枪进行扫货）都要放在最显眼以及最容易点击到的位置。如果有必要，应该在用户首次使用的时候给予一些教程性的指引，以便用户可以完全掌握系统的使用流程和要点。

### 3.2.4其他可行性

系统为毕业设计课题，由学生个人在指导教师的指导下按照软件开发的流程，进行设计开发。属于学生的自主设计和开发行为，整个开发过程，未侵犯他人专利技术，系统内容规范健康，不涉及不良信息，符合法律要求。涉及的虚拟销售产品和测试数据健康积极，符合社会规范，符合道德可行性。

## 3.3 性能需求

由于本软件本质上仍然是一个信息管理系统，因此软件的可用性和其数据的安全性是本软件所要考虑的主要问题。其应该具有以下特点：

1. 功能完备性

本软件应该能够很好的完成基本的进存销系统所具有的功能，如货物管理、产品销售等功能。

1. 软件可用性

由于本软件会涉及到金钱交易，因此需要能够保证其可用性。过多的不正常运行不仅会影响到用户的体验，还会可能使用户蒙受经济损失。

1. 界面友好性

因为小程序仍然是运行于微信中，因此应该尽量保持与原生微信UI保持较为和谐的配色；其次在使用的过程中应该尽量保持用户的使用习惯，让用户能够快速上手；常用的功能应该让用户能够尽量减少点击操作。

## 3.4 功能需求

本系统所应该包含的必要的三大功能模块有：入货管理、商品管理和商品销售。为了提高用户体验，应该具备一些其他的为商家提供便利的功能，如供应商通讯录、打印报表等功能。

### 3.4.1商品入货管理

用户可以通过本功能模块对商品实现进货、生成订货单，查看历史订货信息等功能。并可以追溯到每一笔订货的来源，价格，日期等信息，以便日后复核。进货的方式可以手动输入，亦可选择扫描商品条形码进行入货。

### 3.4.2 商品管理

用户可以通过本功能模块对店内商品进行管理。如查看商品库存信息，修改商品价格信息等。由于商品通常较多，应该能够使得用户检索等过程较为便捷，可以辅以分级菜单或查询功能。

### 3.4.3 商品销售

此功能模块可以实现用户扫码生成订单进行销售的功能。在扫描一件商品后可以选择商品数量、继续扫描等操作直到完成客户所有商品的扫描。为了保证系统运行的可靠性，也可以选择手动输入替代扫码进货的功能。最后会生成一个订单，实现对顾客对收款。

### 3.4.4生成报表

通过对用户进货、销售的信息的分析，可以根据用户选择打印销售额报表、利润报表等信息。更进一步的可以考虑生成商品流通信息，能够让商家根据不同商品的流动速度选择每次进货的数量，其次可以根据进货信息分析某种商品在哪个进货商的平均进货价格较为便宜。

### 3.4.5供应商管理

由于商店运行会涉及到许多的供应商，因此可以在本系统内增加供应商管理功能。其功能除了保存供应商电话、地址等信息外，还能在进货单上附上供应商信息，能够让每一笔进货的来源都可以追溯到。此外，还能够将供应商信息作为数据分析的来源之一，给用户推荐更加实惠的供应来源。

## 3.5 需求分析总结

根据对现有其他平台的进存销系统的调查，除了基本的功能要能够很好地实现之外，良好的用户界面以及尽可能少的减少输入和点击操作能够大大的提高用户的使用体验，因此在整个系统的设计和实现的过程中，都要把良好的的设计理念贯穿始终。这体现于系统的各个业务逻辑步骤中，每个功能模块的视线都要遵循此原则。其次，要有很好的错误反馈还有必要的容错性，比如可撤销的操作，不可逆操作要给出醒目的确认等等。

# 总体设计

在本章节中陈述的是本系统设计的具体功能模块的层次结构以及数据库相关的设计。

## 4.1 概要设计

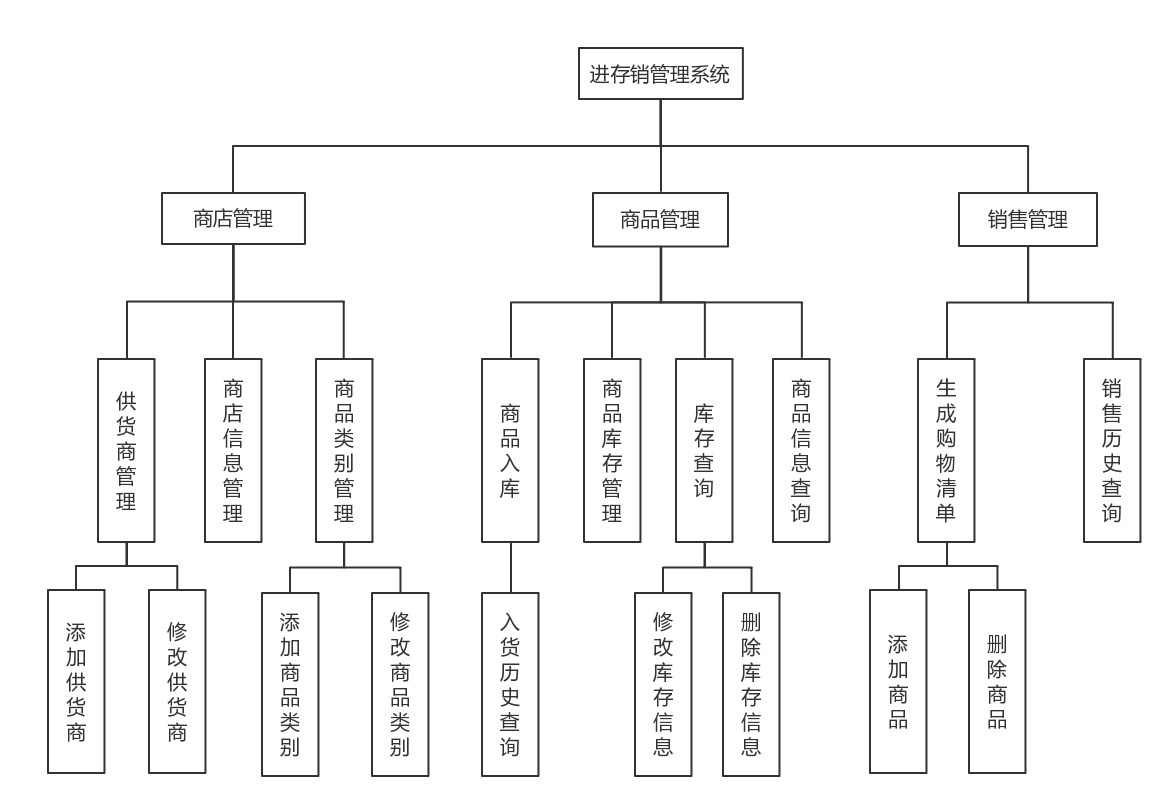
根据上一章的内容可以确立本系统的具体大模块有如下三个：商店管理、商品管理和销售管理。其功能模块图如4.1所示

图4.1进存销管理系统功能模块图

上述功能模块图中，供货商管理，商店信息管理和商品信息管理模块属于比较简单也是比较基本的功能模块，操作流程也比较简单，因此不再进行赘述。下文将对其他功能模块所涉及到的业务逻辑进行进一步解说。

## 4.2 数据库设计

### 4.2.1 数据流图

顶层数据流图如图4.3所示。网站的外部实体有商店和服务器数据库管理员。其中服务器数据管理员的主要工作是为商品数据更新查询API和维护商品数据库。

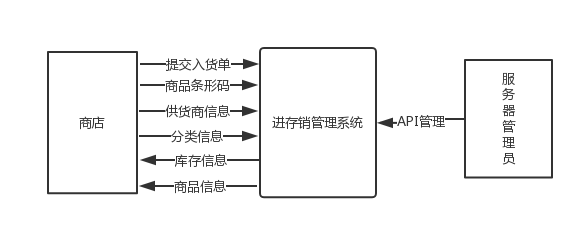


图4.3 顶层数据流图

一级数据流图如图4.4所示。其数据流的主要部分是货物数据流，通过其流动产生了本系统的大部分数据表。其中主要的模块有入货模块和商品出售模块。

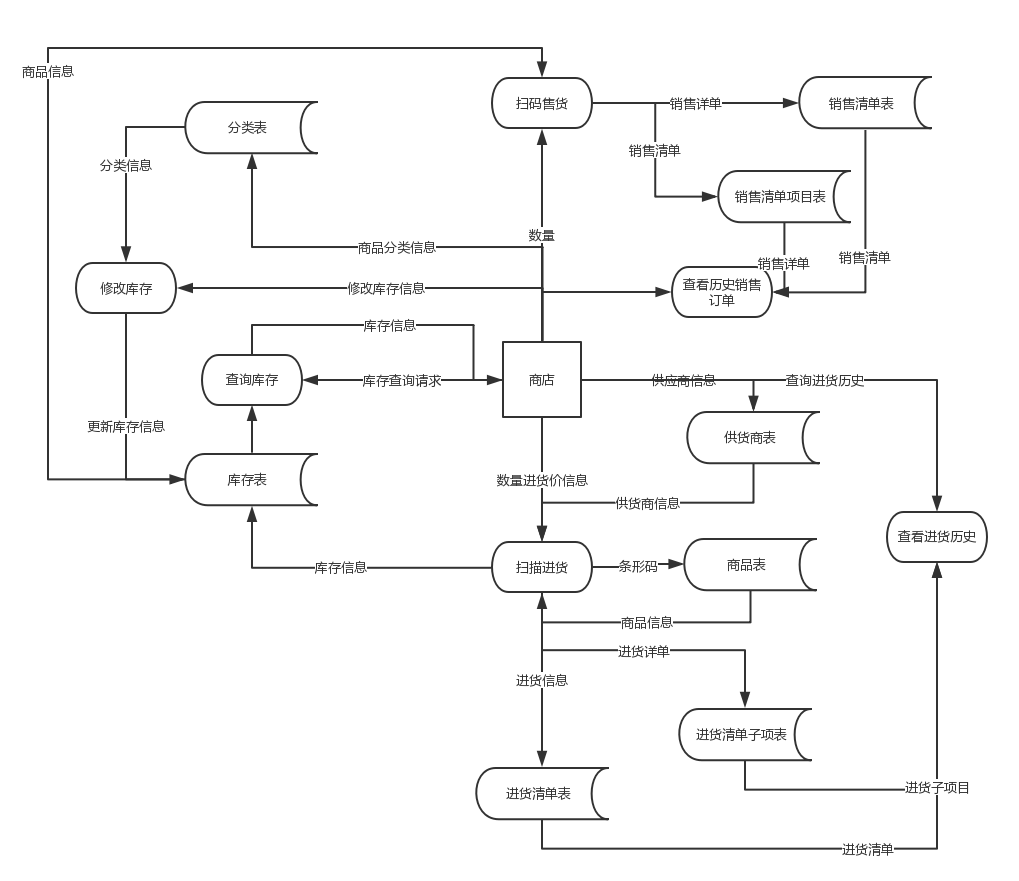


图4.4 0层数据流图

### 4.2.3 数据库结构设计

为了更加直观地表示概念模型，一般常用概念模型来进行表示。表示方法有很多，其中E-R方法是PPSChen于1976年提出的实体联系方法（Entity-Relationship approach）。该方法用E-R图来描述现实世界的概念模型[8]。

根据上一节的系统数据流图分析，我们可以对整个系统的各个实体与其之间的联系有一个打给的结构。图4.5为系统E-R图。本系统所涉及到的大部分实体都是围绕商品产生的，比如一个进货单会有多个货物、一个购物单会有多个商品。其联系都较为简单，符合日常使用逻辑，不再一一赘述。

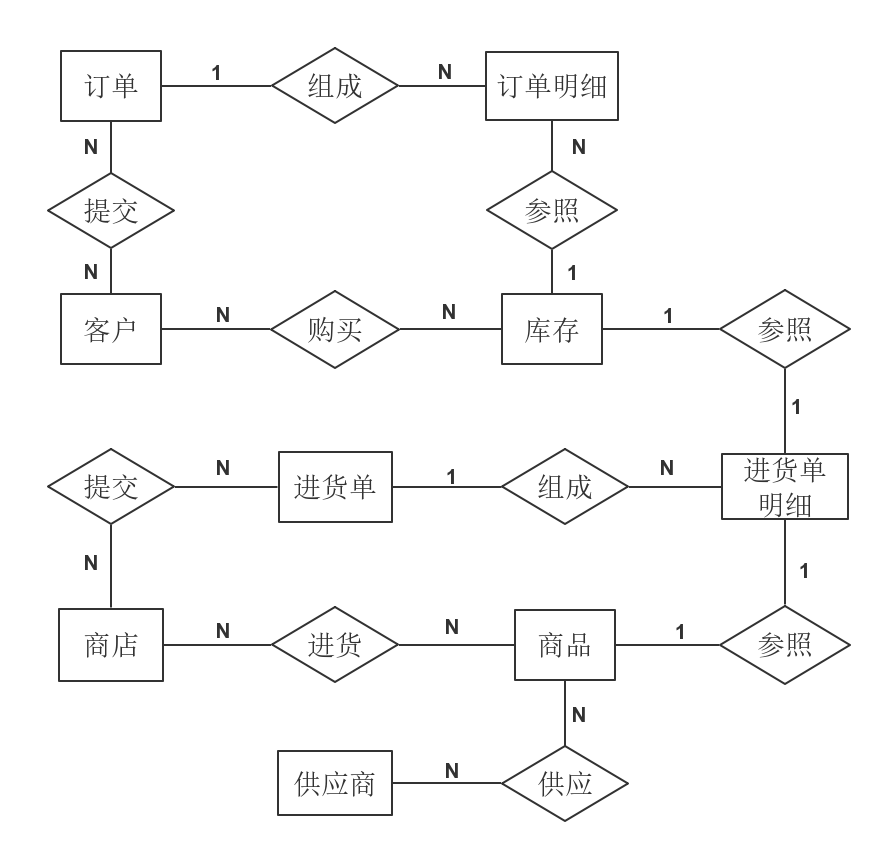


图4.5 系统E-R图

下面介绍的是每个实体的详细属性。如图4.6所示，为商品分类实体的属性。本系统为了便于用户对本商店的货物进行检索，加入了商品的分类，用户可以自己定义商品分类，在商品入货的时候为商品标记其类别。

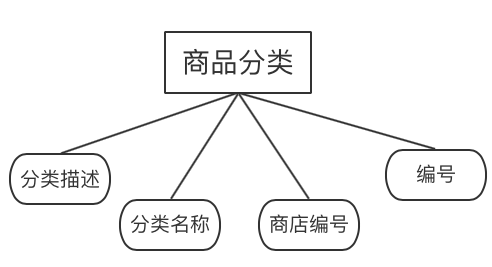


图4.6 商品分类属性图

商品实体如图4.7所示。商品使用条形码的值作为其唯一标记，而作为货物的时候，对应每个商品都会有其所属的商店的ID，以便区分不同商店的库存。其中的二级分类ID为用户自定义的商品类别。

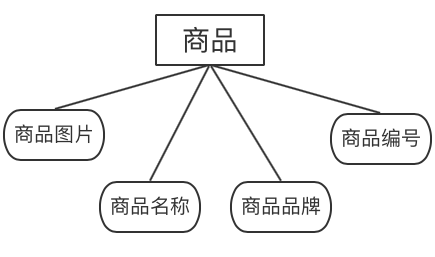


图4.7 商品分类属性

入货清单实体如图4.8所示。入货清单中包含的是每次入货的非商品相关信息，如入货时间、入货数、入货总金额以及供货商信息。

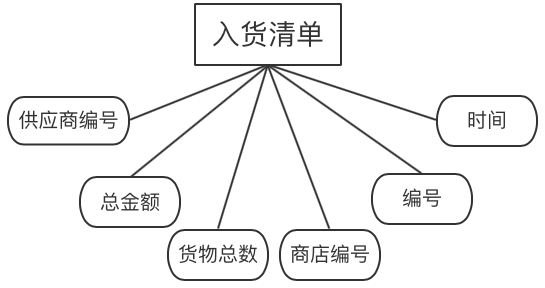


图4.8 入货清单属性

入货清单子项实体如图4.9所示。其包含的信息为入货清单中的每一件货物的信息，如商品的条形码、商品的数目和商品的进货价等信息。其入货清单ID作为与入货清单从属关系的标记。

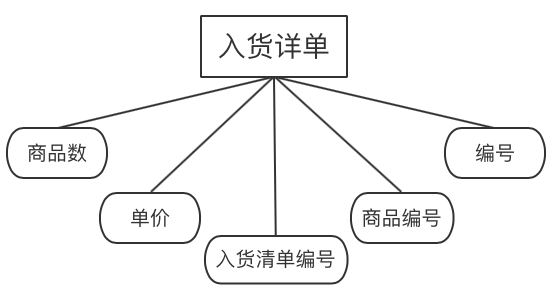


图4.9 入货详单属性

购物清单实体如图4.10所示。其包含的主要信息为生成购物单的时间、本次购物的商品总数和总金额等信息。

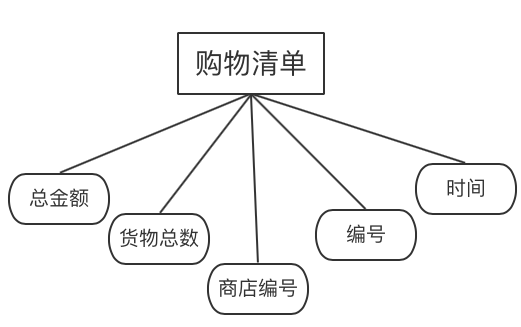


图4.10 购物清单属性

购物清单子项实体如图4.11所示。其内容为每个购物清单中每项商品的信息，包括商品的售价以及商品的数目。购物清单ID作为与其与购物清单关系的标记。

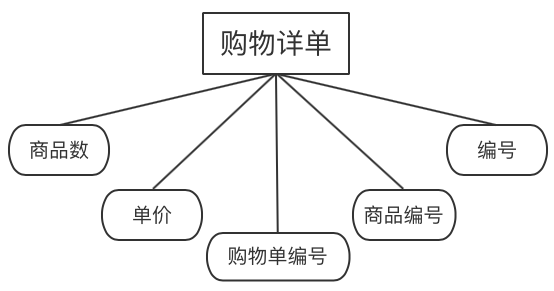


图4.11 购物详单属性

库存实体如图4.12所示。包含的属性为货物的条形码，数量和所属的商店ID。用于记录商店的货物库存信息。

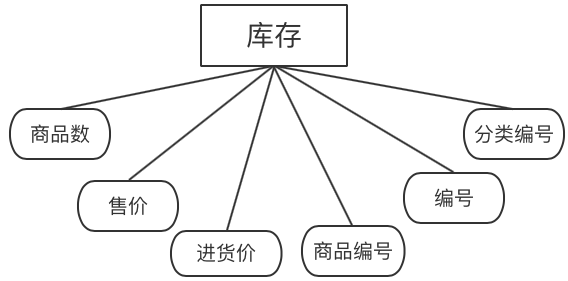


图4.14 库存属性

供应商实体如图4.13所示。包含了供货商联系方式、姓名等信息。用户可以将其作为供货商的通讯录，也可以在进货时将其附在进货单上，以便日后查阅。

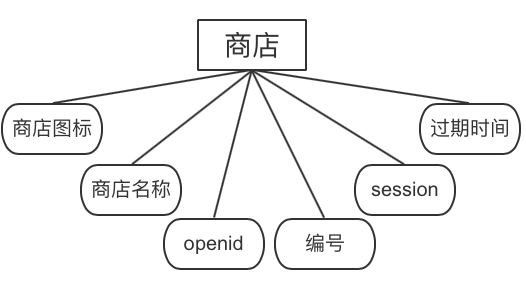


图4.13 商店属性

商店实体如图4.14所示。其包含了商店的基本信息，其主键为每个用户的微信的openid。

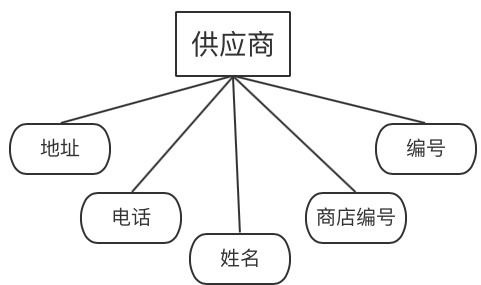


图4.14 供应商属性

### 4.2.3 数据库表结构设计

根据上文对本系统对数据流分析和数据库实体分析，可以建立数据库物理结构。本节将对数据库表结构设计进行叙述。

1. 商店表

商店是本系统的实体之一，商店表用于存储表示每个商店的编号、名称、商店图标等信息。由于小程序为了保护数据的安全性，不建议将用户的openid在网络上进行传输，因此本表中还存有用于维护用户登录态的信息如session字段和expires字段。

表4-1 商店表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| openid | String | 28 | 是 | 否 | 唯一编号 |
| name | String | 32 | 否 | 可 | 商店名称 |
| icon | String | 32 | 否 | 可 | 商店图标 |
| session | varchar | 32 | 否 | 否 | 用于维护小程序登录态 |
| expires | Date | 32 | 否 | 否 | 登录态过期时间 |

1. 货物分类表

表4-2所示为商店的货物分类表，用于存储商店的货物分类信息，如分类的名称和分类的描述信息。

表4-2 货物分类表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| \_id | ObjectId | 32 | 是 | 否 | 数据库提供的唯一编号 |
| openid | String | 32 | 否 | 否 | 分类所属商店的openid |
| name | String | 32 | 否 | 否 | 分类的名称 |
| remark | String | 32 | 否 | 可 | 分类的描述信息 |

1. 货物表

货物表信息如下表4-3所示，其包含了如商品的条形码、商品名称、商品图片链接以及商品的品牌信息。由于从条形码查询API调用返回的每条商品数据均有20个以上的商品属性，为了日后所需本系统的数据库都将其存储起来，但表4-3只列举了本系统只需要用到的字段，其他字段将不一一列举。

表4-3 货物表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| \_id | ObjectId | 32 | 是 | 否 | 数据库提供的唯一编号 |
| barcode | String | 32 | 否 | 否 | 商品的条形码 |
| name | String | 32 | 否 | 否 | 商品名称 |
| img | String | 32 | 否 | 可 | 商品图片的链接 |
| brand | String | 16 | 否 | 是 | 品牌 |

1. 库存表

库存表为所存储的信息为每个商店的库存商品信息，其表结构如表4-4所示。所包含的信息有货物的条形码、进货价、售价还有库存数目等信息。

表4-4 库存表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| \_id | ObjectId | 32 | 是 | 否 | 数据库提供的唯一编号 |
| barcode | String | 32 | 否 | 否 | 条形码 |
| openid | String | 32 | 否 | 否 | 商店的编号 |
| categoryId | String | 32 | 否 | 否 | 商品所属的分类的编号 |
| cost | Double | 16 | 否 | 否 | 进货价 |
| prize | Double | 16 | 否 | 否 | 售价 |
| number | Integer | 16 | 否 | 否 | 的库存数数 |

1. 入货清单表

入货清单表会在每一次入货产生信息，其包含的信息不包括具体的商品信息而是一些诸如进货时间、供货商、进货数目和进货总金额等信息。其表结构如表4-5所示。

表4-5 入货清单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| \_id | ObjectId | 32 | 是 | 否 | 数据库提供的唯一编号 |
| openid | String | 32 | 否 | 否 | 商店的编号 |
| supplierId | String | 32 | 否 | 否 | 供货商编号 |
| totalCost | Double | 16 | 否 | 否 | 进货总价 |
| count | Integer | 16 | 否 | 否 | 商品总数 |
| remark | String | 32 | 否 | 可 | 进货单备注 |
| time | Date | 32 | 否 | 否 | 进货时间 |

1. 入货详单表

入货详单表里面的每一条数据都是入货清单里面的一条商品信息，即每一条入货详单数据都从属于唯一一个入货清单数据。其包含的信息有货物的条形码、所属的入货清单编号、商品的数目和进货价。其表结构如表4-6所示。

表4-6 入货详单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| \_id | ObjectId | 32 | 是 | 否 | 数据库提供的唯一编号 |
| barcode | String | 32 | 否 | 否 | 商品的条形码 |
| purchaseId | String | 32 | 否 | 否 | 所属的购物清单ID |
| count | Integer | 16 | 否 | 否 | 货物数目 |
| cost | Double | 16 | 否 | 否 | 货物单价 |

1. 售货清单表

售货清单表用户存储每一次购物信息，包含了商品的总价、商品数目和购物清单备注信息。其表结构如表4-7所示。

表4-7 售货清单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| \_id | ObjectId | 32 | 是 | 否 | 数据库提供的唯一编号 |
| openid | String | 32 | 否 | 否 | 商店ID |
| totalPrize | Double | 16 | 否 | 否 | 商品总价 |
| count | Integer | 16 | 否 | 否 | 商品总数 |
| remark | String | 32 | 否 | 可 | 购物单备注信息 |

1. 售货详单表

售货详单表里面的每一条数据都是售货清单里面的一条商品信息。其包含的信息有商品的条形码、所属的购物清单编号、商品的数目和单价。其表结构如表4-8所示。

表4-8 售货详单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| \_id | ObjectId | 32 | 是 | 否 | 数据库提供的唯一编号 |
| barcode | String | 32 | 否 | 否 | 商品的条形码 |
| saleId | String | 32 | 否 | 否 | 所属的购物清单ID |
| prize | Double | 16 | 否 | 否 | 商品单价 |
| count | Integer | 16 | 否 | 否 | 商品数目 |

1. 供货商表

供货商表用于给用户提供存储供货商通讯录信息，在每一次进货的记录中亦会包含供货商信息。其包含的信息有供货商名称、供货商联系方式以及供货商地址。其表结构如图4.9所示。

表4-9 供货商表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为主键 | 可否为空 | 说明 |
| \_id | ObjectId | 32 | 是 | 否 | 数据库提供的唯一编号 |
| openid | String | 32 | 否 | 否 | 商店ID |
| name | String | 32 | 否 | 否 | 供货商名称 |
| tel | Double | 16 | 否 | 否 | 供货商联系方式 |
| address | Integer | 16 | 否 | 可 | 供货商地址 |

# 详细设计与系统实现

在本文上述章节中已经进行了本系统的需求分析和概要设计。在此基础上，对本系统进行详细设计与系统实现。

## 5.1 开发运行环境

操作系统：macOS High Sierra

开发语言：JavaScript

数据库：mongodb

开发工具：微信开发者工具，vscode，robo3T等

应用服务器：Nginx

客户端要求：微信5.3及以上版本

服务器端框架：express[11]

## 5.2 系统流程图

扫码入货功能是系统主要功能之一，其实现的功能是通过用户手机对商品的条形码进行扫描，获得商品的信息并呈现给用户，用户可以对入货的商品进行编辑如填写商品分类、入货数目。对于每一次入货，都会生成一个进货单，其中包含了本次进货的时间、商品数目、货物总价和供货商的信息，以便用户日后进行查询。

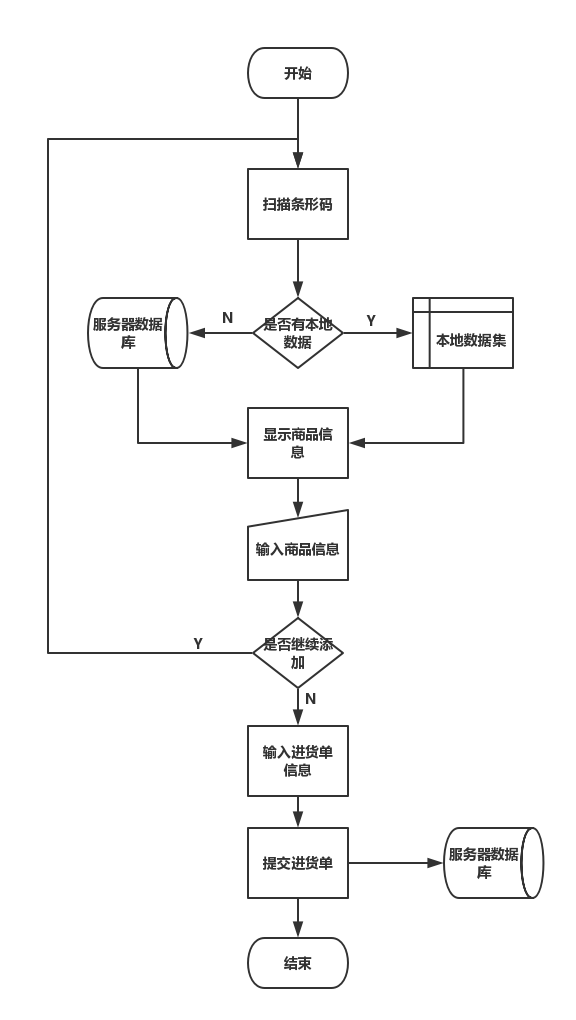


图4.2 扫码入货功能流程图

扫码售货也是系统的主要功能之一，也许产生许多数据流的模块。用户通过扫码可以获取到商品的信息：如本商品的库存数量、商品售价、商品进货价等信息。通过输入购买的商品数目和重复添加本次所需要售卖的商品，完成后提交一个购物清单。购物清单内会有本次清单的商品总数和总金额等信息，并且亦会实时修改库存信息。

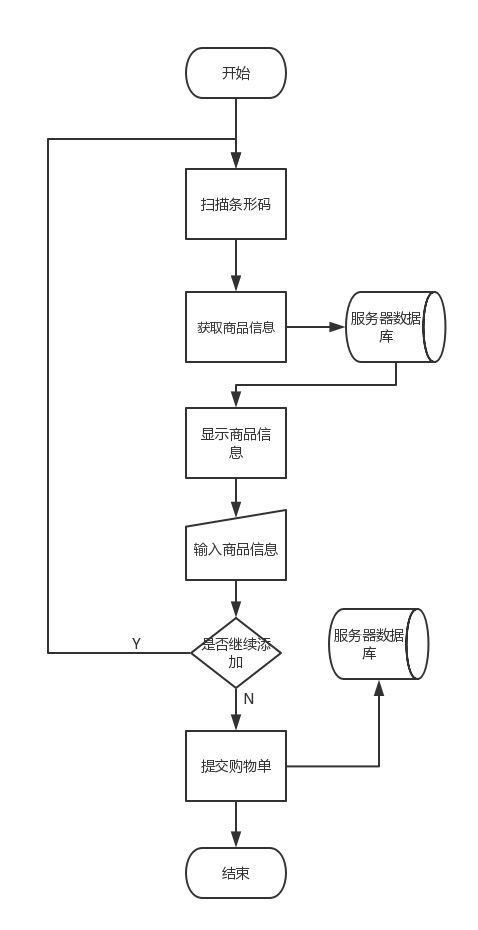


图4.3 扫码售货功能流程图

## 5.3 商店管理功能实现

商店管理功能包含了商店信息管理、商品类别管理以及供货商通讯录管理三个功能。这三个功能与商品没有直接的关系，但是其产生的信息会围绕在进货与售货功能的始终。

1. 商品类别管理

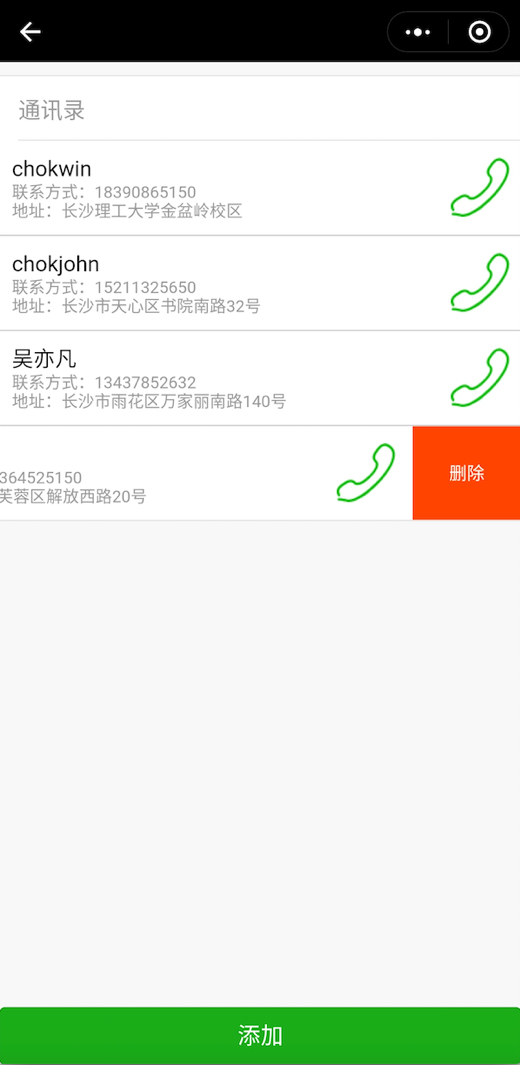
商品类别管理页面如图5.1所示，本功能的主要作用是为不同商店提供可以自定义的商品类别，这样能够更加符合不同商家的实际分类需求，便于商品的检索和管理。

图5.1 商品分类管理页面图 5.2 供货商通讯录管理

1. 供货商通讯录管理

供货商管理可以让店家保存供货商的基本信息，也可以在程序页面直接进行拨号操作。供货商信息还会附加到每一次进货单中，使得每一批货物都有源可溯。在输入电话信息的时候会对用户输入进行校验，判断是否为正确的电话号码格式。供货商管理页面如图5.2所示

## 5.4 库存管理功能实现

（此处略）

# 测试

## 6.1 测试目的

尽管在软件工程科学的方法指导下完成了系统的设计与开发，但是由于软件系统是一个关系紧密且复杂的逻辑系统，因此仅凭严格的设计以及严格的开发流程并不能够完全确保系统不会出现任何缺陷。不存在缺陷的系统是不可能存在的，但是通过软件测试，我们能够尽可能多的在软件系统投入使用前发现目前存在的缺陷并对缺陷进行修复。软件测试与维护这个生命周期是软件生命周期中最长的亦是最重要的，这不仅保证了系统的稳定性以及满足预定的需求，也能够使得软件生命得到延续。因此，通过软件测试对系统进行评估，找到其中隐藏的缺陷并对其加以修复是很有必要的。

软件测试是为了完善系统质量的技术，通过翻阅书籍，发现大多对软件测试的定义为使用人工或自动的手段来运行某个系统的过程，其目的在于检验它是否满足规定的需求，或是弄清预期结果与实际结果的区别。

## 6.2 测试用例

### 6.2.1 扫码入货界面测试

本功能为系统的主要功能，在使用本功能之前，用户必须已经完成注册且已经授权登录。具体的测试如表6-1所示。

**表6-1 扫码入货测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 前提 | 测试步骤 | 预期结果 | 实际结果 | 是否通过 |
| 用  户  已  经  登  录 | 1. 在暗光时打开扫码入货 2. 将摄像头对准待扫描条形码 | 提示“请打开闪光灯” | 提示“请打开闪光灯” | 通过 |
| 1. 打开扫码入货 2. 扫描非商品格式的条形码 | 提示“条形码格式错误” | 提示“条形码格式错误” | 通过 |
| 1. 打开扫码入货 2. 扫描商品条形码 3. 商品数目选择0 | 提示“进货数不能为0” | 提示“进货数不能为0” | 通过 |
| 1. 打开扫码入货 2. 扫描商品条形码 3. 填入正确的商品数目和进货价 4. 点击确定 | 入货清单中显示进货信息 | 入货清单中显示进货信息 | 通过 |
| 1. 将进货清单其中一条货物信息左滑 2. 点击删除 | 入货清单中删去选择的货物 | 入货清单中删去选择的货物 | 通过 |

### 6.2.2 库存管理页面测试

在库存管理中用户可以查看库存信息，并且可以库存商品进行排序与检索。本次测试主要考察的是其排序、检索结果是否准确无误。具体测试情况如表6-2所示。

**表6-2 库存管理测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 前提 | 测试步骤 | 预期结果 | 实际结果 | 是否通过 |
| 用户  已经  登录  并且  库存  信息  中存  在商  品 | 1.进入库存管理页面 | 显示默认排序的库存信息 | 显示默认排序的库存信息 | 通过 |
| 1. 进入库存管理页面 2. 选择“日用品”分类 | 显示该用户分类为“日用品”的商品 | 显示该用户分类为“日用品”的商品 | 通过 |
| 1. 进入库存管理页面 2. 选择按库存数量降序 | 按库存数量降序显示所有库存商品 | 按库存数量降序显示所有库存商品 | 通过 |
| 1. 进入库存管理页面 2. 选择“日用品”分类 3. 选择按库存数量升序 | 按库存数量升序显示分类为“日用品”的商品 | 按库存数量升序显示分类为“日用品”的商品 | 通过 |

## 6.3 测试结果分析

本系统总体上通过了主要基本功能模块的测试，但是正如前文所说的那样，仍然还是不能够保证本系统是毫无缺陷的，但是通过本次测试使得本系统原本不合理的设计得到修改。尽管如此，本系统在安全性以及稳定性上仍有欠缺之处，且本系统是一个信息管理系统，因此安全性以及稳定性仍然是衡量本系统优劣的重要指标之一，需要日后进一步完善。

# 参考文献

1. 阿里研究院，便利店的下一站[EB/OL]. http://www.aliresearch.com/blog/article/detail/id/21419.html
2. 刘红卫. 微信小程序应用探析[J]. 无线互联科技, 2016(23):11-12.
3. 王珊珊. 我国社区便利店经营现状及发展对策研究[J]. 科技致富向导, 2012(15):116-116.
4. Reese W. Nginx: the high-performance web server and reverse proxy[M]. Belltown Media, 2008.
5. Tilkov S, Vinoski S. Node.js: Using JavaScript to Build High-Performance Network Programs[J]. IEEE Internet Computing, 2010, 14(6):80-83.
6. 霍多罗夫. MongoDB权威指南[M]. 人民邮电出版社, 2014.
7. Crockford D. JSON: The fat - free alternative to XML[J]. Proc of Xml, 2006.
8. 王珊, 萨师煊. 数据库系统概论.第5版[M]. 高等教育出版社, 2014.
9. 软件需求工程 康雁 科学出版社，2011
10. 魏兴国. HTTP和HTTPS协议安全性分析[J]. 程序员, 2007(7):53-55.
11. express 官网http://www.expressjs.com.cn/

# 致谢

我首先要感谢我的论文指导老师邓广慧。邓老师对我论文给出了许多指导性的意见，在论文撰写过程中及时对我遇到的困难和疑惑给予悉心指点，提出了许多有益的改善性意见，同时，还要感谢我的同学还有室友们，在四年的大学生活互相学习，互相帮忙，共同度过了一段完美难忘的时光。

此外，还要感谢朋友以及同学们在论文编写中带给的大力支持和帮忙，给我带来极大的启发。也要感谢参考文献中的作者们，透过他们的研究文章，使我对研究课题有了很好的出发点。

最后，谢谢论文评阅老师们的辛苦工作。衷心感谢我的家人、朋友，以及同学们，真是在他们的鼓励和支持下我才得以顺利完成此论文。

# 附录A 部分源代码