##### 摘要

在今日，互联网行业迅速发展，老百姓对于互联网和智能手机所带来的便利十分认同，随着生活节奏的加快，以及店铺成本过高导致的店铺面积狭小等问题，传统方式上的点餐效率相对低下，人们更乐于扫码点餐，无需排队等候，节省了消费者和餐厅的大量时间成本。微信点餐小程序对餐厅的意义在于，高效点餐：更好的展示主推菜品，有利于提高消费；在餐厅繁忙时，客人自助点餐可有效节约人力，消费者体验提高。

本系统使用Thinkphp编写后台管理系统以及供前端使用的API，后台管理系统是前后端分离的结构，后台系统的前端使用了Vue-cli，前端采用微信小程序开发，前端与后台是通过Json数据实现交互，从而在小程序实现多商家点餐小程序的各种功能。消费者的主要功能有：选择喜欢的或距离较近的店铺，并可以收藏店铺，在店铺查看菜品及其菜品详情，加入购物车，下单，查看订单，并对完成的订单进行菜品评论等。商家的主要功能有：申请开通店铺，修改店铺信息和头像，添加菜品，修改菜品信息和折扣，设置菜品分类，审核菜品评论，查看店铺的订单、更改订单状态等。超级管理员除了拥有普通的商家功能外，还拥有审核店铺申请、设置首页轮播图等管理等功能。

关键字

多商家；微信小程序；点餐；Thinkphp；Json；vue-cli

##### ABSTRACT

Today, the Internet industry is developing rapidly. The common people agree with the convenience brought by the Internet and smartphones. With the acceleration of the pace of life and the small size of shops caused by high store costs, the traditional way of ordering is relatively efficient. People are more willing to scan codes to order meals without waiting in line, saving consumers and restaurants a lot of time and cost. The significance of the WeChat ordering applet for restaurants lies in efficient ordering: better display of the main dishes, which is conducive to increasing consumption; when the restaurant is busy, self-ordering by guests can effectively save manpower and improve the consumer experience.

This system uses Thinkphp to write the back-end management system and the API for the front-end. The back-end management system is a structure separated from the front and the back. The front-end of the back-end system uses Vue-cli, the front-end uses WeChat applet development, and the front-end and back-end are implemented through Json data. Interaction, so as to realize the various functions of the multi-merchant ordering applet in the applet. The main functions of consumers are: choose the store they like or are close to, and can collect the store, view the dishes and their details in the store, add to the shopping cart, place an order, view the order, and comment on the completed order, etc. . The main functions of merchants are: apply for opening a store, modify store information and avatars, add dishes, modify dish information and discounts, set dish categories, review dish reviews, view shop orders, change order status, etc. In addition to ordinary merchant functions, super administrators also have management functions such as reviewing store applications and setting home page carousel pictures.

KEY WORDS

Multi-merchant; WeChat applet; ordering; Thinkphp; Json; vue-cli

##### 目录

[第1章 绪论 5](#_Toc64646926)

[1.1 研究背景及意义 5](#_Toc64646927)

[1.2 论文结构 5](#_Toc64646928)

[第2章 需求分析 6](#_Toc64646929)

[2.1 可行性分析 6](#_Toc64646930)

[2.1.1 技术可行性 6](#_Toc64646931)

[2.1.2 操作可行性 6](#_Toc64646932)

[2.1.3 经济可行性 6](#_Toc64646933)

[2.2 多商家点餐小程序系统的需求分析 6](#_Toc64646934)

[2.2.1 小程序端用户需求 7](#_Toc64646935)

[2.2.2 商家端用户需求分析 7](#_Toc64646936)

[2.2.3 超级管理员端需求分析 7](#_Toc64646937)

[第3章 系统的相关开发技术及工具 8](#_Toc64646938)

##### 前言

现如今，随着智能手机的普及和互联网大潮的发展，大多数人的生活都离不开智能手机。市场上不断涌现出各种各样的移动APP，而在2017年1月9日，微信小程序杀入战场，凭借着微信所持有的巨量用户和便利，微信小程序不用安装，搜一搜或扫一扫即可，触手可及，用完即走，利于使用等特点，使得微信小程序的发展前途一片光明。而传统的点餐流程要经过：顾客取号、纸质菜单点菜、服务员核实、服务员手动下单等，在节假日客流高峰时会消耗大量的人力，如果没处理好，会给顾客带来不好的体验。

多商家点餐小程序建设的目的在于：帮助商家和顾客节省时间，顾客直接扫码点餐，从点餐流程来看，全程不需要排队点餐，省去了客人排队、与服务员交流的时间。帮助商家节省人力成本，从点餐环节来看，客人点餐不需要服务员，服务员只需要上餐，节省了开店的人工成本。

# 绪论

## 研究背景及意义

随着互联网的高速发展，移动互联网和手机支付在我国的普及，改变了许多人的生活方式，电子商务搭上了互联网的快车，也发展的如日中天，换来的是人们快节奏的生活方式，而如何为人们的生活带来便利更好的服务，便是我们现今所要思考的。

多商家点餐小程序系统目的是有效地提高餐饮业的工作效率以及规范服务体系，提高整体服务质量和管理水平，并为规模化经营提供坚实的技术基础。它是一款融合传统菜谱与无线点菜信息化于一体的电子点菜设备，是餐饮业的一项重大革新。简单来说，微信点餐小程序系统就是通过小程序所实现的一种可视化餐饮业点菜工具。微信点餐系统对于餐饮公司的意义在于，高效点菜：方便点菜员与顾客沟通，更好展示主推菜品，有效提高单桌消费额；餐厅繁忙时，方便顾客自助点菜，有效节约人工，提高服务质量。

## 论文结构

本文的论文主要分为七个章节来完成：

1. 分析开发背景及意义。
2. 对多商家点餐小程序系统进行需求分析和可行性分析。
3. 简明扼要的描述在多商家点餐小程序系统开发过程中利用的相关技术。
4. 对多商家点餐小程序的设计目标、功能和数据库进行介绍。
5. 对多商家点餐小程序系统的各个功能实现进行简要阐述。
6. 阐述本人对多商家点餐小程序系统实现的个人总结和对该系统的未来展望。

# 需求分析

## 可行性分析

### 技术可行性

多商家点餐小程序系统是由微信开发者工具和JetBrains PhpStorm、JetBrains WebStorm等工具开发，以微信小程序作为前端，前端与后台接口通过json数据进行交互，后台使用ThinkPHP搭建；管理系统的前端使用vue搭建，前后端分离结构，管理系统后端也是同微信小程序的后台。如今ThinkPHP越来越成熟，ThinkPHP5.0对API友好，支持mysql、sqlite等多个数据库，ThinkPHP对于不同模块并无太高要求，一个应用多个模块,并且多个模块可配置不同。同时，ThinkPHP与微信小程序搭配时，开发较为简单。数据库采用稳定可靠的Mysql数据库。因此在技术上是可行的。

### 操作可行性

对于微信小程序，用户只需懂得打开微信扫码或使用微信小程序搜索即可打开使用，不需像普通app一般安装，直接依附微信存在，使用完毕之后直接推出，方便简单。

### 经济可行性

ThinkPHP、Vue是开源框架，使用其开发项目无需支付费用，仅仅只需购买服务器和相应运维人员的费用即可。因为是开源，所以提高了系统的复用性和扩展性。

综上所述，多商家点餐小程序系统的实现基本具备开发价值。

## 多商家点餐小程序系统的需求分析

本系统意在为商户搭建一个平台，微信小程序用户可在平台上进行浏览选择店铺，并在其店铺进行点餐和下单。该系统的服务区域并不是狭小的某个店铺，而是多商家的模式，丰富了微信用户的选择多样性，而多商家的模式意味着开发成本分担大，且复用性高，减轻商家的开店成本。

### 小程序端用户需求

多商家点餐小程序通过与后台的json数据交互，获取店铺信息、用户的位置信息、菜品信息，用于用户即时获取自己位置附近的店铺资料，并对应店铺的菜品信息，从而支持用户点菜，下单，并餐后对菜品进行评价，用户还可收藏店铺，以便于下次光临时，能更快的进入。

点餐小程序实现的功能：浏览并选择店铺、浏览菜品、加入购物车、添加菜品到购物车、下单和餐后评价功能、收藏店铺的实现；但由于支付接口需要商家认证，现阶段无法真正实现。

### 商家端用户需求分析

商家通过申请开通店铺，并可以添加并修改自己店铺的菜品，包括菜品名称、描述、价格和折扣等；还需添加和修改菜品分类，以便于实现点餐小程序的菜品分类导航功能；因为微信用户用完餐后可对菜品进行评论，为防止菜品出现违规言论，需要有评论审核，能删除不合法的言论；具备一般常用的店铺信息修改，如修改店铺位置、店铺名称、店铺头像、店铺状态等；具有订单管理的功能，如查看订单详情，修改订单状态等。

商家端管理后台主要功能点：添加菜品、修改菜品信息，删除菜品，添加菜品分类，删除菜品分类，修改菜品分类，审核菜品评论，修改店铺信息，查看店铺订单和修改订单状态。

### 超级管理员端需求分析

超级管理员除了拥有商家端的功能之外还兼具审核店铺开通申请，以及首页轮播图的添加和修改等功能。

# 系统的相关开发技术及工具

## 微信小程序技术

[微信](https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1/3905974)小程序，[小程序](https://baike.baidu.com/item/%E5%B0%8F%E7%A8%8B%E5%BA%8F)的一种，英文名Wechat Mini Program，是一种不需要下载安装即可使用的[应用](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8)，它实现了应用“触手可及”的梦想，用户扫一扫或搜一下即可打开应用。2016年1月11日微信小程序的概念被张小龙正式提出，于2017年1月9日正式上线。

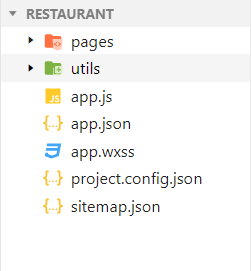
微信小程序的文件结构包含一个描述整体程序的app和多个描述各自页面的page，如下图3-1所示。一个小程序的主题由三个文件组成，创建项目时形成，在项目根目录下，分别是app.js（用于设置小程序的逻辑结构），app.json（用于小程序的全局配置），app.wxss（用于设置小程序的全局样式）；而每个单独的小程序page由四个文件组成，分别是：js文件、wxml文件、wxss文件、json文件，js文件负责页面的逻辑交互，wxml文件负责页面结构，wxss文件是页面样式表，json文件负责页面配置。

图3-1微信小程序项目原始文件目录结构

## Vue.js技术

Vue.js是一个小型、高性能、能组件化的JavaScript MVVM库，是一套用于构建用户界面的渐进式框架。Vue以数据分析驱动及组件化的思维创立的，采用自底向上增量开发的设计。MVVM由Model、View、ViewModel三部分构成，在MVVM架构下，View和Model之间并无直接联系，而是通过ViewModel和Model进行双向数据绑定来进行交互，从而View和Model之间的同步完全自动，开发者只需关注业务逻辑。Vue.js支持丰富的插件，如vue-router路由组件，可构建组件之间的路由，使用axios与后台接口进行数据交互，支持element UI等。

## PHP技术

PHP原始为Personal Home Page的缩写，现已正式更名为Hypertext Preprocesso，中文名称为超文本预处理器。是一种通用开源脚本语言，PHP是在服务器端执行的脚本语言，主要适用于Web开发领域，随着移动应用的兴起，PHP也可用于开发API接口。PHP是一种非常容易学习和使用的一门语言，语法特点类似于C语言，但没有C语言复杂的地址操作，还加上了面向对象的概念，与mysql数据库衔接完美，多年的深耕带来丰富的函数库，各种插件层出不穷，还有强大的跨平台特性使得开发者对其青睐不已。

## 开发工具

### 微信开发者工具

微信开发者工具是腾讯免费提供给微信系列开发者的，开发者可以在其上完成页面开发调试，代码补全，以及小程序预览和发布功能，使开发变的更加直观方便。

### JetBrains PhpStorm

PhpStorm是经由JetBrains公司开发的一种商业的PHP集成式开发工具程序，也是国内广大PHP程序员最为常用的开发软件之一。凭借着其多年发展经验以及专业出色的技术优势收获了无数用户的欢迎和热爱！PhpStorm的首次出现即覆盖了PHP开发智能IDE的长期空白，不仅提供了必不可少的工具，如自动化转换，深层代码分析，联机错误检查和快速修复等多种功能模式，可随时帮助用户进行编码进行调整，运行单元测试和可视化调试操作，且提供了更专业动态错误预防，最佳自动完成和代码重构，零配置调试以及扩展的HTML，CSS和JavaScript编辑器.，从而在支持最新的PHP趋势下更好的保证代码。

### JetBrains WebStorm

**WebStorm**是一款优秀的Web前端编辑器，WebStorm在html5和JavaScript 方面也很出色。WebStorm可以说是“Web前端开发神器”、“强大的HTML5编辑器”和“最智能的JavaScript IDE”，软件支持强大的导航功能，软件是一款非常适合编程人员来使用的Web前端编辑器。

# 多商家点餐小程序系统设计

## 系统功能设计

### 小程序端功能设计

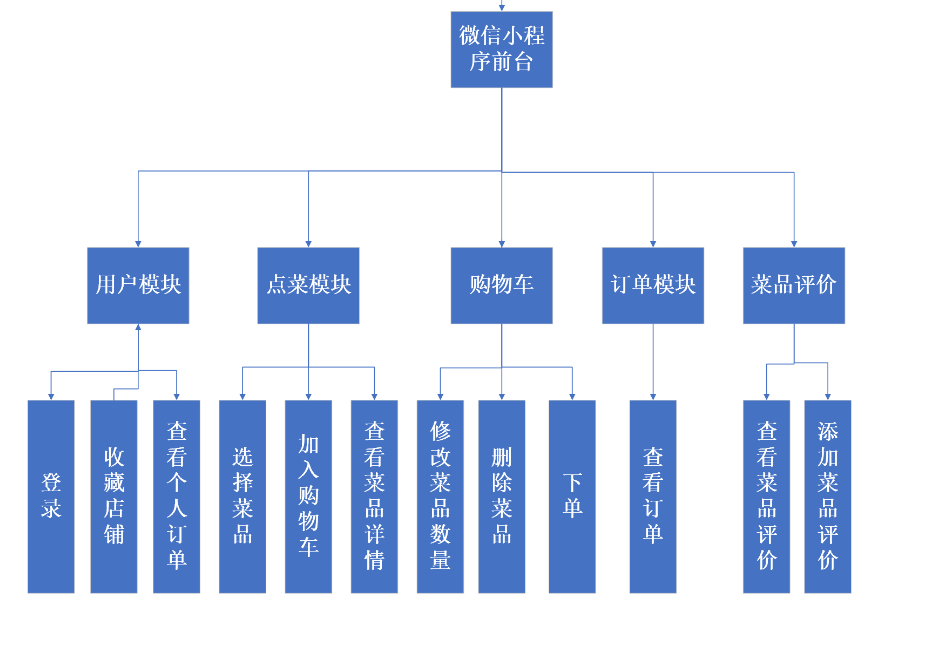
根据第二章的需求分析，小程序所需的功能有：登录、选择店铺、查看菜品、查看菜品评论，添加菜品到购物车、修改购物车菜品、下单、评论菜品、查看历史订单、收藏店铺。功能结构图见4-1。

图4-1 多商家点餐小程序端功能结构图

### 后台管理端功能设计

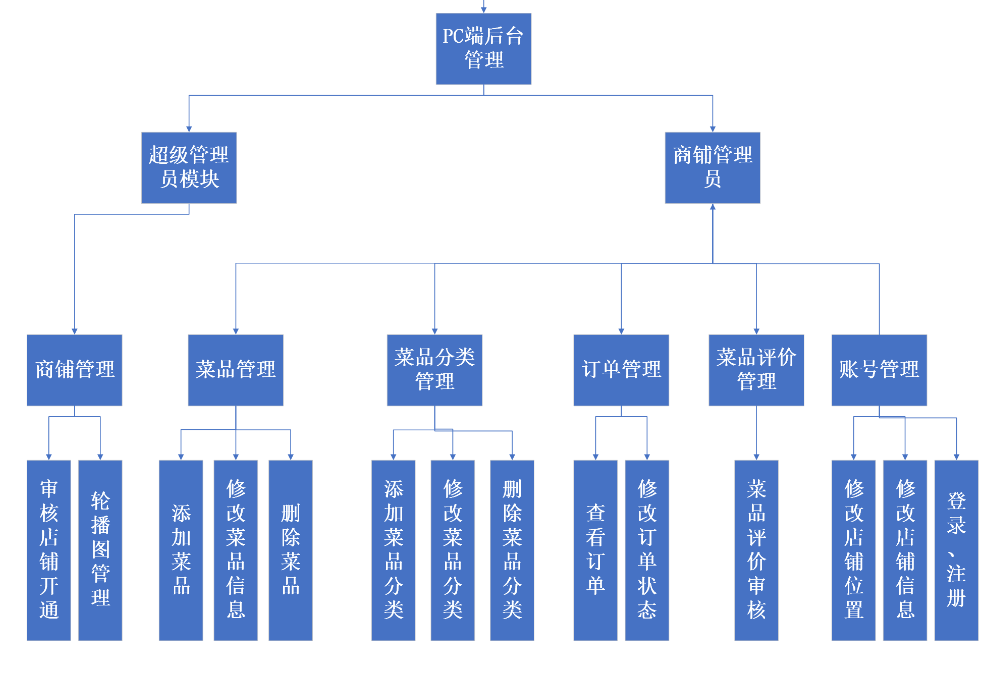
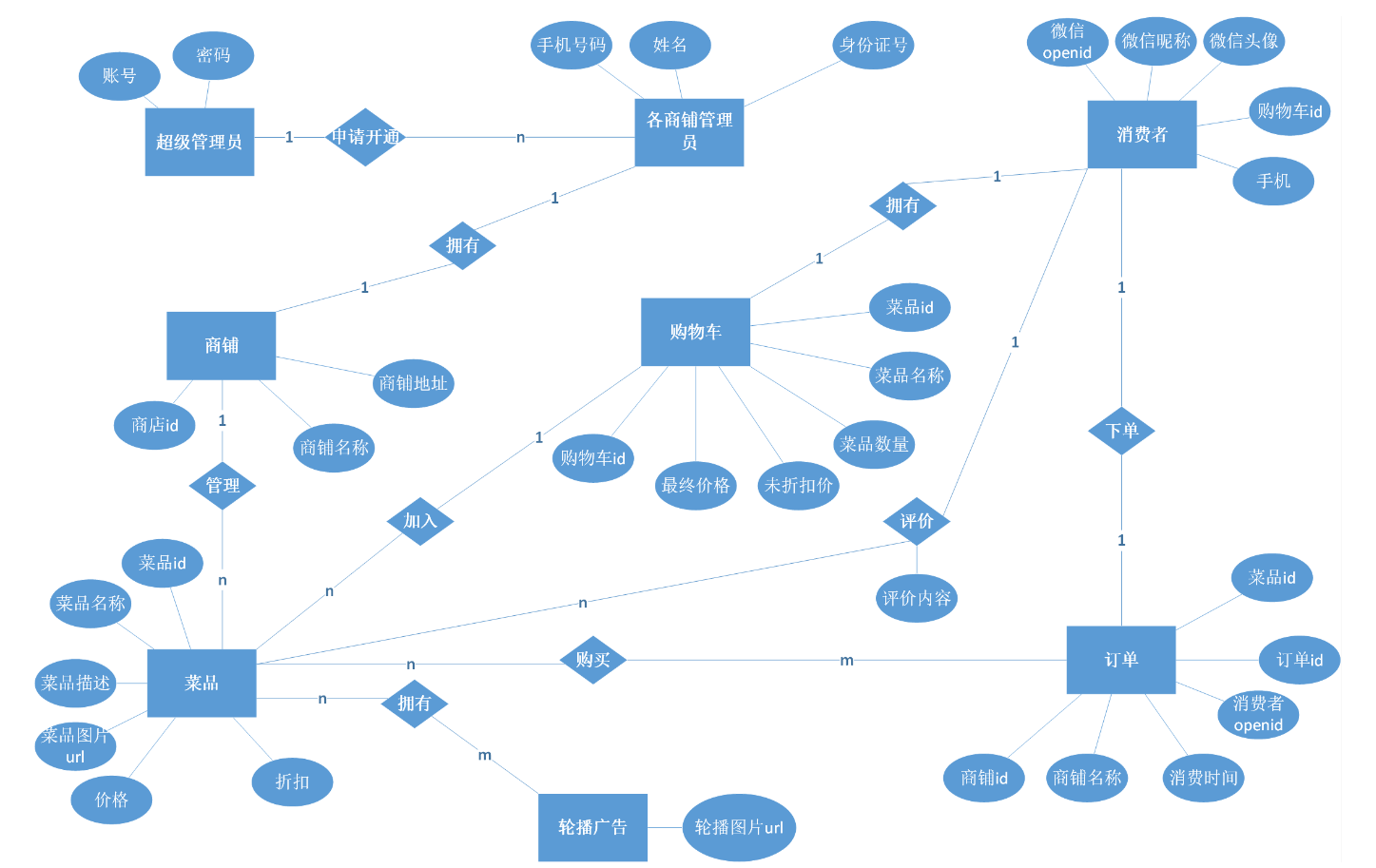
 根据第二章的需求分析，后台管理员端所需的功能有：注册、登录，申请开通店铺、修改店铺信息、添加菜品、修改菜品信息、添加菜品分类、修改菜品分类、查看订单详情、修改订单状态、审核菜品评论；除了以上的店铺管理员功能，超级管理员还具备审核店铺开通申请、添加轮播和修改轮播等功能。如下图后台管理端功能结构4-2所示。

图4-2 后台管理功能设计

## 系统数据库设计

由于MySql具备速度快、轻便、低成本等特点，使用其作为本系统的数据库是合理的选择。通过以上的功能架构图的设计，为满足系统需求的实体关系从而制作出数据库实体E-R图。

图4-3 E-R图

### 数据库表结构设计

多商家点餐小程序系统的数据库设计十分精细，因此实体之间关联极多，需要优化表的结构。

表 4-1 表汇总

|  |  |  |
| --- | --- | --- |
| 表序号 | 表名 | 表描述 |
| 表4-2 | user | 微信用户表 |
| 表4-3 | storeadmin | 后台管理员表 |
| 表4-4 | store\_apply | 店铺申请表 |
| 表4-5 | store | 店铺信息表 |
| 表4-6 | order\_vice | 订单附表 |
| 表4-7 | order\_main | 订单主表 |
| 表4-8 | meal\_number | 店铺取餐号记录表 |
| 表4-9 | lunbo | 轮播图表 |
| 表4-10 | food | 菜品表 |
| 表4-11 | favorite | 店铺收藏表 |
| 表4-12 | comment | 菜品评论表 |
| 表4-13 | category | 菜品分类表 |

（1）微信用户表，表名称为user，用于存储微信小程序用户的信息，详细设计见表4-2。

表4-2 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| openid | varchar(255) | Y | 微信openid，主键 |
| wxname | varchar(255) | N | 微信昵称 |
| head | varchar(255) | N | 用户头像url |
| phone | varchar(255) | N | 手机号码 |

（2）后台管理员表，表名称为storeadmin，用于存储管理员的信息，详细设计见表4-3。

表4-3 后台管理员表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| storeId | varchar(255) | N | 商店id |
| username | varchar(255) | N | 账号 |
| pwd | varchar(255) | N | 密码 |
| phone | varchar(255) | N | 手机号码 |
| id\_card | varchar(255) | N | 身份证号 |
| limited | int(255) | N | 权限 |

（3）店铺申请表，表名称为store\_apply，用于记录商铺的申请表，以及申请状态以便超级管理员审核，详细设计见表4-4。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| applyId | int(11) | Y | 操作id，主键 |
| storeId | varchar(255) | N | 商店id |
| storename | varchar(255) | N | 商店名称 |
| state | varchar(255) | N | 审核状态 |
| nickname | varchar(255) | N | 真实姓名 |
| phone | varchar(255) | N | 手机号码 |
| city | varchar(255) | N | 所在城市 |
| city\_j | float(255,0) | N | 店铺所在经度 |
| city\_w | float(255,0) | N | 店铺所在维度 |
| id\_card | varchar(255) | N | 身份证号码 |
| location | varchar(255) | N | xx-xx店 |
| username | varchar(255) | N | 申请人账号，外键 |

（4）店铺信息表，表名称为store，存储店铺的基本信息，详细设计见表4-5。

表 4-5 店铺信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| storeId | varchar(100) | Y | 商店id |
| storename | varchar(255) | N | 商店名称 |
| city | varchar(255) | N | 所在城市 |
| location | varchar(255) | N | 地理位置 |
| city\_j | decimal(65,5) | N | 店铺所在经度 |
| city\_w | decimal(65,5) | N | 店铺所在维度 |
| store\_img | varchar(255) | N | 店铺头像url |
| store\_status | tinyint(255) | N | 店铺状态（1为开店，0为闭业） |

（5）订单附属表，表名称为order\_vice，依赖于订单主表，主要是存储订单的菜品详情，如菜品名称、数量、价格等信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| order\_main | varchar(255) | N | 订单主表id，外键连接订单主表 |
| foodId | varchar(255) | N | 菜品id |
| foodName | varchar(255) | N | 菜品名称 |
| foodNumber | int(11) | N | 菜品数量 |
| price | decimal(10,2) | N | 价格 |
| picture | varchar(255) | N | 菜品图片url |
| discount | decimal(5,2) | N | 菜品折扣 |
| comment | tinyint(255) | N | 该单是否评价（0否1是） |

（6）订单主表，表名称为order\_mian，下位表是order\_vice，主要存储微信用户的订单信息，如订单id，商铺id，微信用户id，购买总价，备注，总价等，详细见表4-5。

表 4-5 订单主表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| orderId | varchar(255) | Y | 订单id，主键 |
| sellerId | varchar(255) | N | 卖家id |
| buyerId | varchar(255) | N | 买家id |
| time | datetime | N | 订单创建时间 |
| status | varchar(255) | N | 订单状态 |
| total | decimal(10,2) | N | 总金额 |
| No | int(11) | N | 取餐号 |
| comment | varchar(255) | N | 备注 |

（7）店铺取餐号记录表，表名称为meal\_number，用于存储各个店铺的取餐号，记录每个商铺每一天的取餐号码，详细设计见表4-6。

表4-6 店铺取餐号记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| storeId | varchar(255) | N | 商店id，外键 |
| no | int(11) | N | 取餐号 |
| date | varchar(255) | N | 日期 |

（8）轮播图表，表名称为lunbo，用于记录微信小程序端主页的轮播url，详细设计如下表4-7。

表4-7 轮播图表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| pid | int(11) | Y | 图片id，主键 |
| pic | varchar(255) | N | 图片地址url |
| title | varchar(255) | N | 图片描述 |
| state | varchar(255) | N | 状态（0为下线，1为上线） |

（9）菜品表，表名称为food，存储每个店铺的菜品信息，如菜品的名称、描述、价格、图片的url、折扣、所属店铺、所属分类等。详细设计见下表4-8。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| foodId | varchar(100) | Y | 菜品id，主键 |
| foodName | varchar(255) | N | 菜品名称 |
| price | int(10) | N | 菜品价格 |
| describe | varchar(255) | N | 描述图片url |
| discount | decimal(5,2) | N | 折扣 |
| cid | int(255) | N | 菜品分类 |
| imgurl | varchar(255) | N | 菜品图片url |
| storeId | varchar(255) | N | 所属商店id |
| sales | int(255) | N | 销量 |
| state | varchar(255) | N | 菜品状态 |

（10）店铺收藏表，表名称为favorite，存储微信用户所收藏的店铺，方便用户下次光临喜欢的店铺，详细设计见表4-9。

表 4-9 店铺收藏表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| openid | varchar(255) | N | 微信用户的openid，外键 |
| storeid | varchar(255) | N | 商店id |

（11）菜品评价表，表名称为comment，存储了每个菜品的评价、评价人、所属订单、评价状态，详细设计见表4-10。

表 4-10 菜品评价表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| foodId | varchar(255) | N | 评价食物id |
| comment | varchar(255) | N | 评论内容 |
| comment\_time | datetime | N | 评论时间 |
| openid | varchar(255) | N | 评论者id |
| orderId | varchar(255) | N | 来自的订单id |
| id | int(11) | Y | 评论id，主键 |
| storeId | varchar(255) | N | 商店id |
| status | tinyint(255) | N | 评价状态（1为显示，2为隐藏） |

（12）菜品分类表，表名称为category，存储了每个店铺的菜品分类，详细设计见表4-11。

表 4-11 菜品分类表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 备注 |
| cid | int(11) | Y | 分类id，主键 |
| category | varchar(255) | N | 分类名 |
| storeId | varchar(255) | N | 所属商店id |

# 系统实现