|  |
| --- |
| **基于ThinkPHP框架的点餐系统** |

|  |
| --- |
| Ordering System Based On The ThinkPHP Framework |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 杨超 | 学号 | | 201311671227 | |
| 所在学院 | 信息学院 | | 班级 | | 信管1131 |
| 所在专业 | 信息管理与信息系统 | | | | |
| 申请学位 | 工学学士学位 | | | | |
| 指导教师 | 涂超 | | 职称 | | 副教授 |
| 副指导教师 |  | | 职称 | |  |
| 答辩时间 | 2017年6月3日 | | | | |

[设计总说明 I](#_Toc7644)

**[introduction](#_Toc513)** [2](#_Toc513)

[1 绪论 3](#_Toc20650)

[1.1 为什么要将饮餐业作为研究对象？ 3](#_Toc13334)

[1.2 开发点餐系统的目的在哪里？ 4](#_Toc6541)

**[2](#_Toc37)** [技术支持和需求分析 4](#_Toc37)

[2.1 需要哪些技术支持？ 4](#_Toc29912)

[2.2 任务和开发环境 5](#_Toc5375)

[2.3 功能分析 5](#_Toc4788)

[2.3.1后台总代理 5](#_Toc13379)

[2.3.2前台点餐平台 6](#_Toc8190)

[2.4 实体的描述 6](#_Toc19015)

**[3](#_Toc9969)** [系统设计 7](#_Toc9969)

[3.1 系统内容结构 7](#_Toc21133)

[3.2 数据库结构设计 8](#_Toc20059)

[3.2.1前台点餐平台的数据库设计 8](#_Toc21693)

[3.2.2后台总代理的数据库设计 10](#_Toc13094)

[作为总代理，我们需要添加其他的管理员，因此我们要设计一张管理员表；我们还生成注册码给品牌商，因此我们要设计注册码表和品牌商表；之后我们还有设备管理模块，首先需要一张设备表，然后我们需要对设备进行查询，查询按照品牌商、商铺、地区的顺序查找，因此需要商铺表和地区表，在找到某个商铺下的所有设备之后，我们还要查看每个设备的续费状态，因此还需要一张续费表。 10](#_Toc26223)

[3.3 接口的设计 12](#_Toc14203)

[3.3.1管理员的接口 12](#_Toc16463)

[3.3.2设备的接口 13](#_Toc15320)

[3.3.3品牌商的接口 14](#_Toc10596)

[3.3.4注册码的接口 14](#_Toc6586)

[4 系统详细设计和实现 15](#_Toc20860)

[4.1 前台功能模块的实现 15](#_Toc13528)

[4.1.1广告 16](#_Toc15373)

[4.1.2就餐方式 17](#_Toc24324)

[4.1.3菜谱分类 17](#_Toc9996)

[4.1.4每道菜的详情 18](#_Toc27682)

[4.1.5购物车 18](#_Toc30084)

[4.1.6支付方式 19](#_Toc32478)

[4.2 总代理功能模块的实现 20](#_Toc9214)

[4.2.1总代理登录 20](#_Toc7173)

[4.2.2品牌商管理 21](#_Toc15598)

[4.2.2设备管理 23](#_Toc16810)

[4.2.3管理员管理 25](#_Toc13762)

[4.2.4注册码管理 25](#_Toc26756)

[4.3 系统实现 26](#_Toc8527)

[4.3.1数据库的实现 26](#_Toc18726)

[4.3.2部分模块的实现 30](#_Toc19112)

**[5](#_Toc19453)** [系统测试 30](#_Toc19453)

[5.1 测试用例 31](#_Toc9123)

[5.2 测试结果 31](#_Toc567)

[6 总结 32](#_Toc3225)

[鸣 谢 33](#_Toc14088)

[参考文献 34](#_Toc26410)

# 设计总说明

"民以食为天"是中国一句家喻户晓的古话,最早出自《汉书·郦食其传》,云:"王者以民为天,民以食为天"[[[1]](#endnote-0)]。在当今时代，餐饮业的发展越来越快，传统的人工点餐方式的弊端越来越明显，在互联网+的推动下，开发出一款智能的电子点餐系统就尤为重要了。

在分析了当前饮餐行业发展的现状，再结合餐厅具体的工作流程，我们设计并开发了一个可以替代传统手工完成的新型智能方便的点餐系统。

本点餐系统分为两个模块，及前端和后台。前端我们选择性JavaScript作为开发语言，并且使用了 JQuery库 和 Bootstrap（一种CSS框架），开发工具选择的是 WebStrom；后台采用PHP语言，基于ThinkPHP框架，以PHPstorm为开发工具，MySQL为数据库进行开发。

为了方便后期的管理，本系统通过角色将系统划分为一个前台点餐平台子模块和三个后台管理子模块，它们分别是总代理模块、代理模块和商铺模块，它们三个是有上下级关系，即总代理生成代理，代理生成店铺，店铺在对菜谱进行有效的管理。

由于本系统比较复杂，我一个人能力有限，无法驾驭，因此只是参与了总代理模块和点餐平台的开发（本论文也就这个两个模块进行讨论），而其他的两个模块则交给了另一位小伙伴开发了。

本系统以打破传统餐厅的手工点餐模式为出发点，努力实现简化点餐流程，节省人力成本，提高劳动效率为目标，使得点餐、下单、支付、用餐一体化，满足客户的需求，提高服务质量，推动餐饮业发展。

关键词：点餐系统；ThinkPHP；智能化

# **introduction**

"Hunger breeds discontentment" is Chinese a word from the earliest known to every family saying, "Han Li Shiqi biography", said: "the king to the people for days, hunger breeds discontentment". In today's era, the development of the catering industry more and more quickly, the traditional way of ordering the artificial defects are more and more obvious, in the promotion of Internet plus under the development of an intelligent electronic ordering system is particularly important.In the analysis of the current development of the restaurant industry, combined with the specific work flow of the restaurant, we designed and developed a new intelligent and convenient meal ordering system which can replace the traditional manual.

This meal system is divided into two modules, and the front and back. The front-end language is JavaScript, the use of the framework is JQuery and Bootstrap, development tools is WebStrom; the background of the PHP language, based on the ThinkPHP framework, using PHPstorm as development tool, MySQL as database development.

For the convenience of management, the system through the role of the system is divided into a front ordering module and three background management module, which is the general agent module, agent module and shops modules, three of them are superior and subordinate, namely general agent generating agent, agent into the shop, shop effectively management of recipes.

This system in order to break the traditional mode of manual meal restaurant as the starting point, try to simplify the ordering process, save manpower cost, improve labor efficiency as the goal, the ordering, orders, payment, meal integration, to meet customer demand, improve service quality, promote the development of the catering industry.

**Keywords:** Ordering System; ThinkPHP; Intellectualization

基于ThinkPHP框架的点餐系统

信息管理与信息系统，201311671227，杨超

指导教师：涂超

毕业设计说明书

# 绪论

## 为什么要将饮餐业作为研究对象？

在当今的这个时代，计算机技术和软件技术在信息化的发展中占主导地位，它覆盖了很多领域。随着现代传媒和互联网的普及及应用，网络通信、电子办公、网上缴费、网络一体化的进程加速加速再加速。

最近几年，物物相连的物联网也开始逐渐进入了人们的视野，改变和提高了人们的生活方式，在数字化和智能化的这股浪潮的冲击下，网上点餐系统这种新型的点餐方式便运营而生了。这种终端的点餐系统，记录了商铺的菜谱种类，销售情况和好评度，让消费者一目了然，不但节省了点餐的时间，而且商家可以通过系统的数据推出最受消费者喜爱的菜肴，极大的减少了消费者用餐的时间和降低了商家的经营成本。

在实习期间，每天工作很辛苦，最大的感触是生活节奏加快，时间根本不够。比如，中餐和午休时间只有仅仅的1.5小时，也就是说午饭时间越短，午休时间就越充足，然而，大多数餐馆都还是沿用传统的点餐模式，选好座位，手动写单，开始消费，收到支付，整个流程下来，不仅耗费了时间，而且无形间就耗费了不少的宝贵时间，比如：从下单到接单的这一段时间差。并且，我们在点餐的时候，只能凭借一本菜谱，上面只有寥寥数笔的单价、名称和图片，我们根本不知道其他消费者的评价，有时候可能就点了自己不喜欢吃的菜也是很正常的。

而作为商家，到了饭点的时候，店铺里往往是爆满，这时候人手往往不够。从接待顾客坐下，到点餐，再到做饭，到用餐付款，这期间会有许许多多的其他顾客，大家的需求是不一致，但是大家都是希望能够更及时地填饱自己的肚子。这时候对商家从下单到出菜的时间要求就非常短了，而且这期间可能会出错，把甲客户点的菜上个乙客户了，这样会让客户感到服务不到位，回头客就大大减少，影响店铺的盈利。

因此，设计出一个智能的点餐系统是当前饮食产业的必然产物。

本系统就是面向消费者和商家的点餐系统。此系统是一种比较智能化的管理系统，它不仅能够满足消费点餐便捷的需求，也可以满足商家能够及时处理订单的需求。它能够实现点餐的基本功能，一方面商家可以对菜谱的信息进行录入、查询、修改和删除，另一方面，消费者可以对菜谱进行翻阅，可以通过评论对菜进行友好的选择。

## 开发点餐系统的目的在哪里？

在我们开发这个系统之前，我们要周全考虑到系统所有的使用者，要充分满足他们的需求。在我们的点餐系统中，商家可以将自己的产品放入到系统中进行统一的管理，比如新商品的录入和查询，订单的查询和展示，营业额的统计和管理等等，顾客可以通过浏览网页的形式快速地查看商家的所有商品，而且对于每到菜都可以进行仔细的查看。如果想要吃某一种菜，还可以进行查询，以最快的速度找到自己喜欢的菜肴。

在系统设计过程中，为了避免查询数据和统计数据等困难，需要代码规范化，软件统一化，确保软件的可读性、可维护性和实用性；删除重复的环节，实现管理的科学化；界面要简洁实用，尽量满足需求。本次课程设计可以加深我们对MYSQL数据库、PHP语言、thinkphp框架以及前端技术的理解和使用程度，增强发现问题、分析问题和解决问题的能力。本次课程设计，使我们能够主动查询相关开发语言的相关资料，更好扩展了我们的知识点，从而达到大学生自我发现问题、分析问题和解决问题的目的。

# 技术支持和需求分析

## 需要哪些技术支持？

本次开发用的技术支持为前台的HTML、CSS、JavaScript、JQuery、Bootstrap，后台用到的技术是PHP、ThinkPHP、Mysql。

HTML(HyperText Mark-up Language)是目前网络上应用最为广泛的语言,也是构成网页文档的主要语言[[[2]](#endnote-1)]。

CSS全称层叠样式表，它可以用来修饰HTML或XML样式。它不仅可以静态地修饰网页，还可以配合动态语言（如：Javascript）来动态地修饰网页。

目前常用的网页布局设计主要有表格技术和DIV+CSS技术两种,其中使用表格技术进行网页布局对于网页设计人员来说比较容易理解并通过代码实现,而DIV+CSS技术作为一种新的网页设计技术,实现的功能将更强,灵活性也更高[[[3]](#endnote-2)]。

JavaScript也是前端开发工程师必须掌握的三种技能之一：描述网页内容的HTML、描述网页样式的CSS以及描述网页行为的JavaScript[[[4]](#endnote-3)]。

JQuery是一个专门用来动态改变Web页面文档的JavaScript库[[[5]](#endnote-4)]。它可以用少量地代码代替大量的JavaScript代码，而且在兼容性上得到很大的提升，一些简单的语法糖可以是编程变得更加简单灵活，是一个不错的框架。

Bootstrap是一个CSS的框架，它的内部封装了大量的美观的组件，我们使用它，不仅可以节省我们的开发时间，而且可以使得界面变得根据简洁大方。

PHP是一种用于创建动态WEB页面的服务端脚本语言。其语法结合了c语言、java语言和Perl语言的优点和长处[[[6]](#endnote-5)]，利用我们的学习和掌握，使用范围十分之广，尤其是在web领域有很好的适用性。相比于CGI或者Perl，能够更加快速地执行动态网页。因为PHP将程序直接嵌入到HTML中，执行效率远远高于生成HTML标记的CGI，并且PHP通过编译代码，达到加密和优化的目的，使代码运行更快[[[7]](#endnote-6)]。

ThinkPHP是国人开发出来的具有简单易懂、模块齐全的轻量级的PHP框架。

Mysql是一个关系型数据库，用来存储一些表数据信息，它与PHP可谓是“黄金搭档”。

## 任务和开发环境

本人的主要任务是要在规定的时间内完成总代理模块的开发和点餐平台的开发，使得总代理可以添加修改删除查询管理员、设备、注册和品牌商，而顾客可以在点餐机上点餐、管理购物车、下单和支付。

开发环境如表2.1所示：

表2-1 开发环境表

|  |  |
| --- | --- |
| 集成开发环境 | phpStrom |
| 网页开发环境 | WebStrom |
| 测试环境 | 新版浏览器 |
| 版本控制工具 | Git |
| 数据库 | mysql |
| 数据库管理工具 | Navicat |
| 服务器 | apache |
| 操作系统 | windows |

## 功能分析

### 2.3.1后台总代理

通过总代理登录界面进入总代理后台，总代理用户可以对设备、注册码、管理员和品牌商进行操作，比如说录入数据，修改数据，删除数据和模糊查询数据。

我通过对需求的分析，感觉总代理模块需要实现下面几个功能，如表2-2所示：

表2-2 后台总代理功能表

|  |  |
| --- | --- |
| 功能 | 具体描述 |
| 登录界面 | 用户输入总代理账号、总代理密码和系统随机生成的验证码登录到总代理后台页面。 |
| 品牌商管理 | 用户只有在通过登录后才能操作。用户可以添加、查看、修改、删除和搜索品牌商。新增品牌商可以对选择级别，填写名称、账号、密码、公司名称、联系人、手机和备注。 |
| 注册码管理 | 用户只有在通过登录后才能操作。用户可以批量删除注册码；也可以一次性添加多个注册码给指定的品牌商；还可以查看注册码属于哪个品牌商和使用的状态；可以将注册码重新分配给另一个品牌商。 |
| 管理员管理 | 用户只有在通过登录后才能操作。用户可以添加、查看、修改和删除管理员。 |
| 设备管理 | 用户只有在通过登录后才能操作。用户可以通过三级菜单的形式一层层地查找对应的代理商下的商铺下的区域的所有设备信息，设备信息包括名称、MAC地址、注册日期、到期时间、状态和设备的详细地址，还可以对设备进行编辑、续费、删除和查询；用户还可以直接对设备进行模块搜索。 |

### 2.3.2前台点餐平台

顾客进入前台点餐平台之前，首先会看到广告界面，这些广告是通过后台的设置来完成的；然后系统会让顾客选择就餐方式，方式包括打包和餐厅内就餐两种；之后进入点餐界面，先通过类别筛选，然后再选择单个菜，单个菜还可以选择它的配菜；用户选中的菜会加入到购物车，用户可以对购物车进行管理；当点好餐之后，用户就可以下单了；最后跳转到支付页面，它应该支持三个支付方式，微信支付、支付宝支付和现金支付。

通过上面的分析，我认为前台点餐品台需要实现下面几个功能，如表2-3所示：

表2-3 前台点餐平台功能表

|  |  |
| --- | --- |
| 功能 | 具体描述 |
| 激活设备 | 商家激活设备需要将设备的MAC地址和注册码绑定起来，然后续费激活。如果已经激活并且没有过期则不需要此步骤。 |
| 点餐 | 用户选择了就餐方法，进入到点餐界面后，可以通过菜品的类别查看有哪些菜，然后单击某道菜可以查看这道菜的详细的信息，如图片、名称、类别、单价、描述，还可以进行点餐选择属性，如数量、饮料等，如果喜欢这道菜就把它添加到购物车；当点餐完毕后可以通过购物车查看自己点了哪些菜，如果您不满意还可以删除，确认之后就可以下订单支付了。 |

## 实体的描述

通过上面的分析，我们不难得出前端的实体是设备，后台的实体是总代理管理员，下面对这些实体进行分析：

1. 设备：设备ID、设备码、设备状态、设备名称、激活码、设备开始时间、设备结束时间和纸张情况。
2. 总代理：主键ID、管理员账号、管理员密码、管理员电话和管理员备注。

# 系统设计

## 系统内容结构

本系统主要是为了点餐者可以在餐厅中快速地点餐，而餐厅的经营者能够更快地为消费者提供服务的电子点餐系统。

我负责的是总代理和点餐模块。接下来请看图3.1、3.2、3.3的功能模块图。



图3.1 点餐系统总的功能模块图

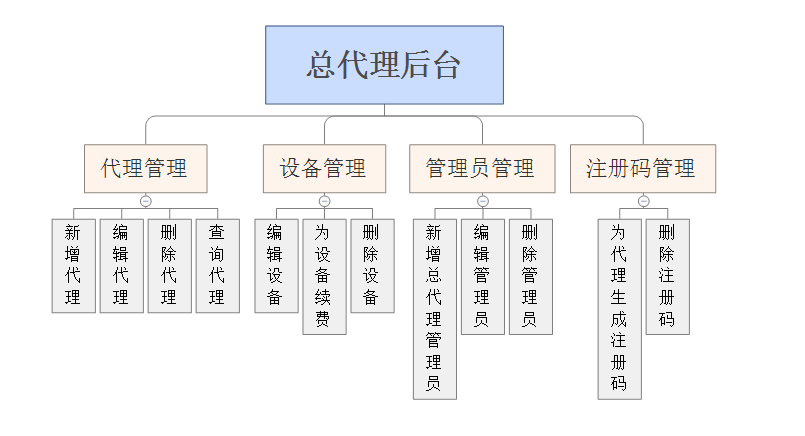


图3.2 总代理的功能模块图

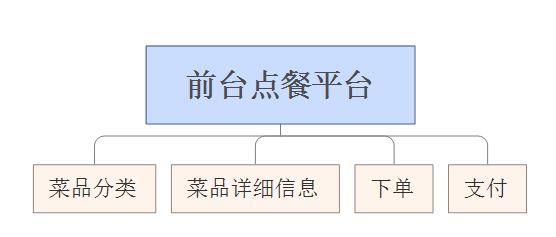


图3.3 前台点餐平台的功能模块图

## 数据库结构设计

数据库设计是指对于一个给定的应用环境，构造（设计）优化的数据库逻辑模式和物理结构，并据此建立数据库及其应用系统，使之能够有效地存储和管理数据，满足各种用户的应用需求，包括信息管理要求和数据操作要求[[[8]](#endnote-7)]。

### 3.2.1前台点餐平台的数据库设计

在点餐平台中，菜品的类别、菜品、菜品的属性这些信息在商铺的后台里就已经设计好了（另一位小伙伴负责），我这里直接获取数据了，就不需要重新设计了。

然而，在顾客点餐下单的过程中，还是会产生以下三个表的，它们分别为订单表、菜品订单表和菜品属性订单表。

接下来我们来详细看看这几张表：

1. order（订单数据表），主要用来存储订单数据。如表3-1所示：

表3-1 订单数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| order\_id | int | 11 | 是 | 订单id |
| order\_sn | varchar | 100 | 否 | 订单号 |
| order\_type | tinyint | 4 | 否 | 订单类型 |
| pay\_type | tinyint | 4 | 否 | 支付类型 |
| total\_amount | decimal | 10 | 否 | 订单总价 |
| add\_time | int | 11 | 否 | 下单时间 |
| pay\_time | int | 11 | 否 | 支付时间 |
| order\_status | tinyint | 4 | 否 | 订单状态 |
| table\_num | char | 4 | 否 | 餐桌号 |

1. order\_food（菜品订单数据表），主要用来存储订单中的菜品数据。如表3-2所示：

表3-2 菜品订单数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| order\_food\_id | int | 11 | 是 | 菜品订单id |
| order\_id | int | 11 | 否 | 关联订单id |
| food\_id | int | 11 | 否 | 菜品id |
| food\_num | int | 11 | 否 | 菜品份数 |
| food\_price2 | decimal | 10 | 否 | 菜品加属性单价 |
| food\_name | varchar | 100 | 否 | 菜品名称 |
| restaurant\_id | int | 11 | 否 | 店铺id |
| table\_num | char | 4 | 否 | 餐桌号 |

1. order\_food\_attribute（菜品属性订单数据表），主主要用来存储订单中的菜品的属性数据。如表3-3所示：

表3-3 菜品属性订单数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| id | int | 11 | 是 | 属性订单id |
| order\_food\_id | int | 11 | 否 | 关联订单id |
| food\_attribute \_name | varchar | 50 | 否 | 订单属性名 |
| food\_attribute\_price | decimal | 10 | 否 | 属性价格 |
| print\_id | int | 11 | 否 | 打印机id |
| count\_type | smallint | 4 | 否 | 是否列入统计 |

### 3.2.2后台总代理的数据库设计

### 作为总代理，我们需要添加其他的管理员，因此我们要设计一张管理员表；我们还生成注册码给品牌商，因此我们要设计注册码表和品牌商表；之后我们还有设备管理模块，首先需要一张设备表，然后我们需要对设备进行查询，查询按照品牌商、商铺、地区的顺序查找，因此需要商铺表和地区表，在找到某个商铺下的所有设备之后，我们还要查看每个设备的续费状态，因此还需要一张续费表。

接下来我们来详细看看这几张表：

1. manager（总代理管理员表），主要用来存储总代理后台的管理员的详细信息。如表4-1所示：

表3-4 总代理管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| manager\_id | int | 11 | 是 | 主键ID |
| manager\_account | varchar | 80 | 否 | 管理员账号 |
| manager\_password | char | 32 | 否 | 管理员密码 |
| manager\_phone | varchar | 11 | 否 | 管理员电话 |
| manager\_ps | varchar | 255 | 否 | 管理员备注 |

1. business（代理信息表），主要用来保存代理的信息。如表4-2所示：

表3-5 代理信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| business\_id | int | 11 | 是 | 代理id |
| business\_name | varchar | 50 | 否 | 代理名称 |
| business\_account | varchar | 100 | 否 | 代理账号 |
| business\_password | varchar | 80 | 否 | 代理密码 |
| corporate\_name | varchar | 50 | 否 | 公司名字 |
| business\_phone | varchar | 11 | 否 | 手机号 |
| business\_contact | varchar | 50 | 否 | 联系人 |
| business\_ps | varchar | 255 | 否 | 备注 |
| business\_grade | smallint | 4 | 否 | 代理等级 |

1. restaurant（店铺表），用来保存店铺的相关信息。如表4-3所示：

表3-6 店铺表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| restaurant\_id | int | 11 | 是 | 店铺id |
| restaurant\_name | varchar | 50 | 否 | 店铺名 |
| telephone1 | varchar | 11 | 否 | 店铺电话1 |
| telephone2 | varchar | 11 | 否 | 店铺电话2 |
| address | varchar | 255 | 否 | 店铺地址 |
| logo | varchar | 100 | 否 | 店铺图标 |
| advertise\_time | int | 11 | 否 | 店铺广告时间 |
| tplcolor\_id | int | 11 | 否 | 模板颜色 |
| business\_id | int | 11 | 否 | 关联代理id |
| city1 | int | 11 | 否 | 省 |
| city2 | int | 11 | 否 | 市 |
| city3 | int | 11 | 否 | 区 |

1. code（激活码表），保存设备激活码的详细信息。如表3-7所示：

表4-4 激活码表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| code\_id | int | 11 | 是 | 主键id |
| code | char | 16 | 否 | 激活码 |
| code\_status | tinyint | 4 | 否 | 状态 |
| code\_timestamp | int | 11 | 否 | 有效时间 |
| rest\_timestamp\_ | int | 11 | 否 | 剩余时间 |
| business\_id | int | 11 | 否 | 代理id |
| restaurant\_id | int | 11 | 否 | 店铺id |
| last\_time | int | 11 | 否 | 最后一次的登录时间 |

1. device（设备表），用来保存设备的相关信息。如表3-8所示：

表4-5设备表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| device\_id | int | 11 | 是 | 主键id |
| device\_code | varchar | 50 | 否 | 设备码 |
| device\_status | tinyint | 4 | 否 | 设备状态 |
| device\_name | varchar | 50 | 否 | 设备名称 |
| code\_id | int | 11 | 否 | 激活码id |
| start\_time | int | 11 | 否 | 设备开始时间 |
| end\_time | int | 11 | 否 | 设备结束时间 |

1. region（地区表），用来保存地区的相关信息。如表3-9所示：

表3-9地区表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| id | int | 11 | 是 | 主键id |
| name | varchar | 50 | 否 | 地区名称 |
| parent\_id | tinyint | 4 | 否 | 改地区的父级 |
| level | tinyint | 4 | 否 | 地区的等级 |

1. renew（续费表），用来保存设备续费情况的。如表3-10所示：

表3-10续费表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| id | int | 11 | 是 | 主键id |
| renew\_time1 | tinyint | 4 | 否 | 续费1年 |
| renew\_time2 | tinyint | 4 | 否 | 续费3年 |
| renew\_time3 | tinyint | 4 | 否 | 续费5年 |

## 接口的设计

在前台点餐页面中，我只是负责数据的交互，接口另一个小伙伴在商铺管理模块已经写好了，这里就不累述了。

这里详细介绍后台总代理模块的接口设计。

### 3.3.1管理员的接口

总代理后台系统只有登录才能够进入，因此在登录的时候要进行管理员登录接口checklogin；在登录之后我们需要查看管理员列表的接口admin，任意修改某个管理员接口edit\_admin，删除某个管理员接口del\_admin，添加新管理员接口add\_admin。详细信息如图3.4所示：

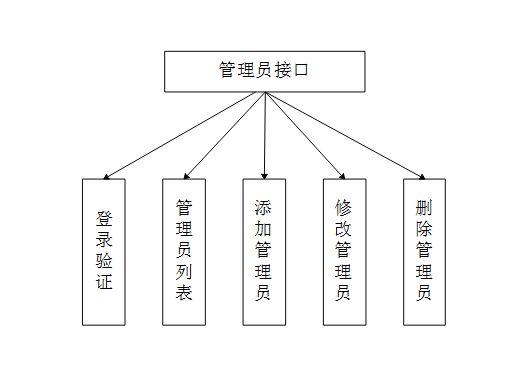


图3-4 总代理管理员接口图

### 3.3.2设备的接口

设备管理模块，我们需要所有设备进行查看的接口device，我们需要模糊查询设备showdevicebykey，我们需要按照日期查询设备searchDevicebyTime，我们需要点击具体的品牌商->店铺->区域对设备进行查询showInfobykey，我们需要编辑某个设备的信息update\_device以及我们需要删除某个设备del\_device。详细信息如图3.5所示：

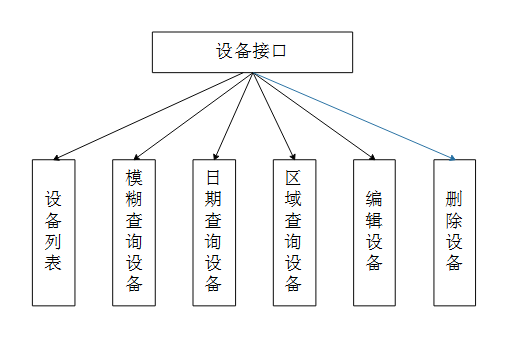


图3-5 总代理设备接口图

### 3.3.3品牌商的接口

在登录之后我们需要查看品牌商列表的接口agent，并且我们要获取每个品牌商的等级getManagerRank，我们需要按照条件来查找品牌商，我们任意修改某个品牌商接口edit\_\_business，删除某个品牌商接口del\_business，添加新品牌商接口add\_\_business。详细信息如图3.6所示：

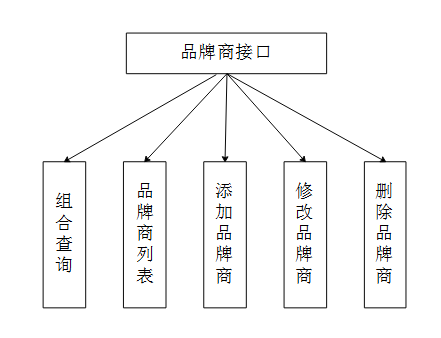


图3-6 总代理品牌商接口图

### 3.3.4注册码的接口

在登录之后我们需要查看注册码列表的接口code，删除某个注册码接口deleteCode，批量删除多个注册码接口batch\_delete，添加注册码接口create\_code，改变某个注册码的所属商家changeCodeBusiness。详细信息如图3.7所示：

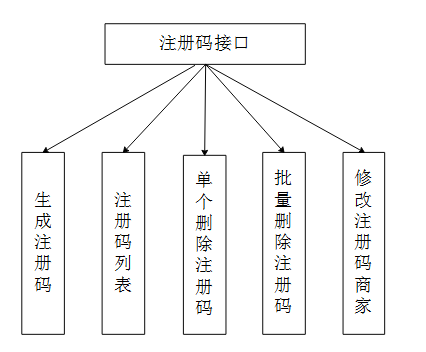


图3-7 总代理注册码接口图

# 系统详细设计和实现

## 前台功能模块的实现

在实现前台各个模块之前，我们首先看看前台的大致流程图，如图4-1所示：

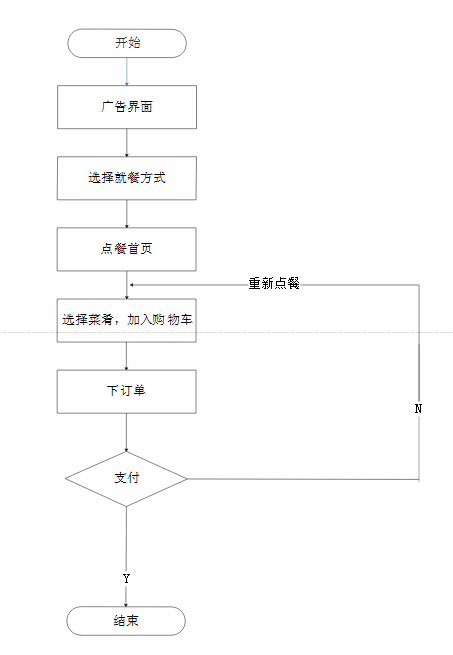


图4-1 前端点餐流程图界面

### 4.1.1广告

在正式点餐之前，我们首先会看到广告界面。广告是网站是相当重要的一部分，一方面它可以用来宣传自己餐厅的一些重要信息，另一方面它也可以带来可观的经济收入，如果其他商家需要借助您的点餐平台打广告也是很不错的！广告界面如图4-2所示：



图4-2 广告界面

### 4.1.2就餐方式

由于工作关系或者是时间关系，有一些顾客并不想在餐厅里享用美食，因此本系统退出了店内就餐和打包带走两种方式。如图4-3所示：



图4-3 选择就餐方式界面

### 4.1.3菜谱分类

餐厅为了尽可能地照顾到每位顾客的口味亦或者是为了吸引顾客，会推出很多美食产品。那么问题来了，如果菜的数量过多会让顾客目接不暇，不知道点哪些菜好，因此本系统在点餐首页推出了菜谱分类的效果，这样将菜进行分类，让顾客们能够一目了然。如图4-4所示：



图4-4 菜谱分类界面

### 4.1.4每道菜的详情

在按照分类查找到喜欢的菜的时候，我们需要查看每道菜的详细心情，包括它的价格、描述、数量、属性以及模样，如果满意的话，就可以把它放入您的购物车中。如图4-5所示：



图4-5 菜的详情界面

### 4.1.5购物车

购物车是本系统必不可的一部分，它其实就相当于你去超市买东西手中的篮子。你选择需要的东西，然后将它们放入到篮子中，以方便下单付钱的时候使用。本系统的购物车和上面的篮子大同小异，您将喜欢的菜肴放入到购物车中以便统一管理，如果你喜欢便可以保留，如果您不喜欢也可以剔除，最后可以下订单了。如图4-6所示：



图4-6 购物车界面

### 4.1.6支付方式

当下订单之后，我们就要开始支付了。在如今电子现金使用如此频繁的情况下，本系统为了顾客着想也推出了多种支付方式，比如支付宝转账、微信扫描或者是传统的现金支付都可以，在完成支付之后，您就可以凭借小票在餐桌前等候美食的到来了。如图4-7所示：



图4-7 支付页面

## 总代理功能模块的实现

### 4.2.1总代理登录

任何一个系统，在进入管理页面之前，都是需要进行登录来身份验证，这样能够保证系统的安全性，本系统也不例外。在总代理登录界面，我们需要对总代理的账号、总代理的密码和验证码进行验证，只有登录成功，才能跳转到后台。

首先先看一下后台登录的流程图，如图4-8所示：

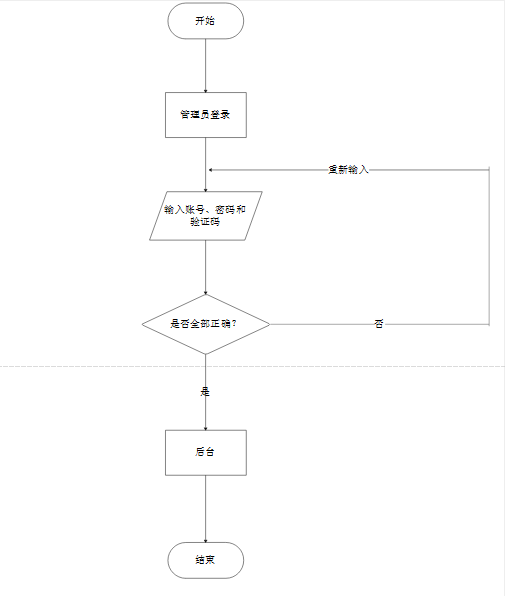


图4-8 总代理登录流程图

然后我们在看看总代理登录界面的设计，如图4-9所示：



图4-9 总代理登录页面

### 4.2.2品牌商管理

作为总代理，它必须得有自己的品牌商，也就是代理商。不然的话，总代理的事情很多，会忙不过来的，因为总代理要负责的区域太大了，因此需要代理商来分担压力。

进入品牌商界面，我们会看见一个分页表格，这个表格中记录了当前登录的代理商下面的所有品牌商，我们可以查看具体的某一品牌商的信息，如名称、账号、等级、公司名称、联系人等等，如图4-10所示：



图4-10 品牌商列表页面

在查看了每个品牌商的信息之后，我们可以单独地对某个品牌商的信息进行修改，如图4-11所示：



图4-11 修改某个品牌商页面

当然，如果您不需要这个品牌商，也可以把它删除掉，但是要确认一下，以防止您误点到“删除”，如图4-12所示：

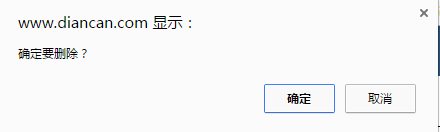


图4-12 删除某个品牌商页面

同时，如果您想查看某个品牌商的信息，本系统也支持搜索功能，比如您输入”一”进行品牌商和账号的组合搜索，结果如图4-13所示：



图4-13 查询某个品牌商页面

当然，有删除就有添加，这里您可以随意地添加品牌商，如图4-14所示：



图4-14 添加品牌商页面

### 4.2.2设备管理

设备管理也是后台总代理中一个重要的环节。这里要罗列出所有商家的设备，但是由于设备太多，我们也是按照前面菜谱的方式将设备分成多级进行管理，即品牌商、商铺、地区的顺序，最后点击某个地区之后就可以看到设备列表了，如我们操作点击了“广州方雅电子有限公司 -> 方派餐饮系统 -> 番禺区”，如图4-15所示：



图4-15 搜索设备页面

在查看了某个设备的信息之后，同样的，我们可以对其进行编辑、删除、续费和按照时间查询的功能，这里只截图编辑和续费功能（删除和查询和上面大同小异），如图4-16、图4-17所示：

在编辑页面，我们可以就设备的民称、结束日期、详细地址、状态进行修改保存。



图4-16 设备编辑页面

如果某个设备需要续费，我们可以对其进行续费，时间是1、3、5年三种。



图4-17 设备续费页面

### 4.2.3管理员管理

管理员管理的主要功能是删除管理员功能、查看管理员功能、修改管理员功能和新增管理员功能。如图4-18所示：

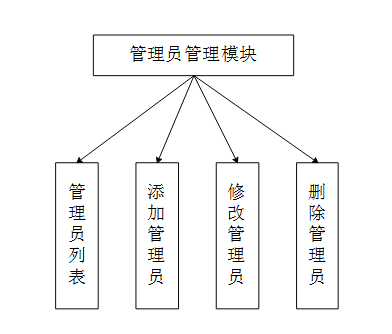


图4-18 管理员功能图

添加总代理管理员：如果总代理管理员想要添加一个新的管理员来帮助管理后台，那么总代理管理员要通过后台的管理员管理模块添加一个新的管理员，然后新的管理员通过总代理登录界面登录到后台才能够使用系统的相应的功能。

编辑总代理管理员：对于之前添加录入的管理员，可以录入的信息有误或者录入的信息不完整，这时候需要对信息进行再次编辑，比如手机号、备注、名称等等。

删除总代理管理员：对于一些因为某些事情不能够担任管理员身份的人，比如辞职的人，我们要删除它的管理员身份。

### 4.2.4注册码管理

注册码管理主要功能有：分页查看注册码列表、修改某个注册码的所属品牌商、单条和批量删除注册码和系统生成1个或10个注册码给某个品牌商，如图4-19所示：

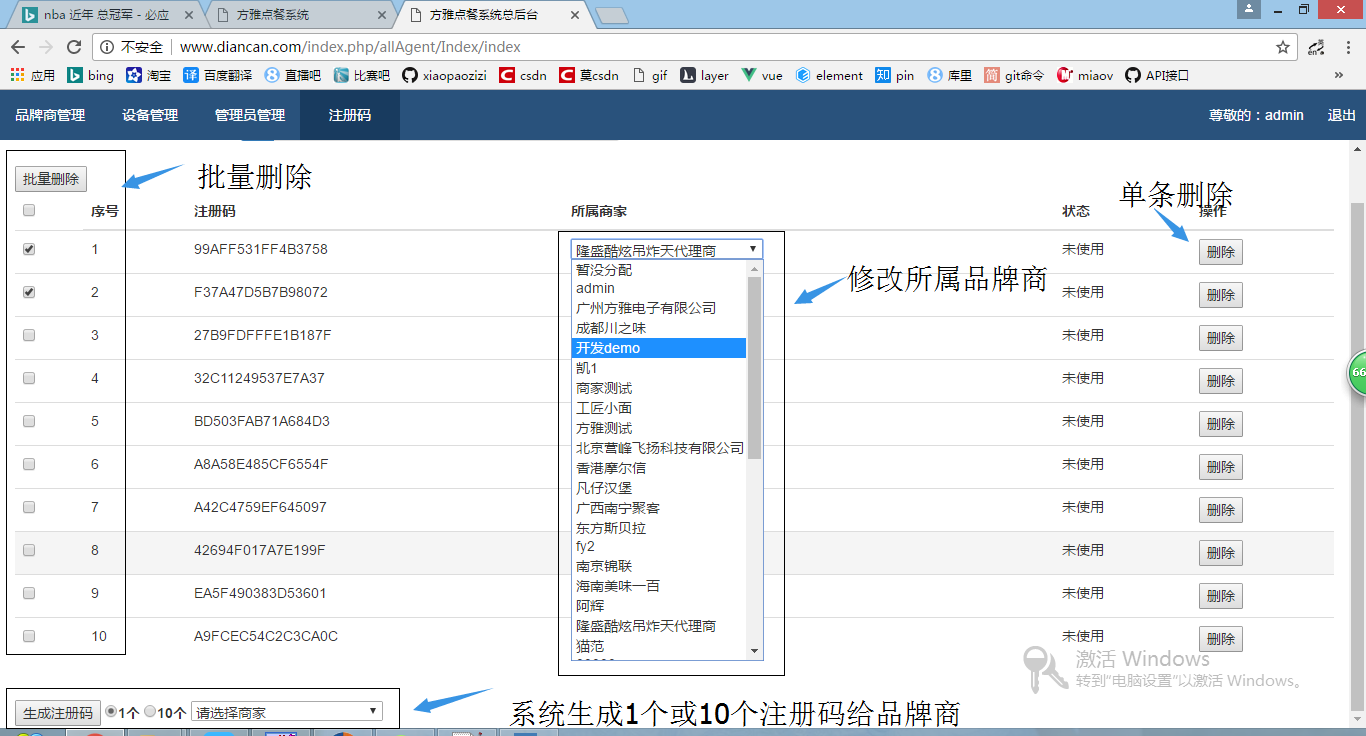


图4-19 注册码操作界面

## 系统实现

### 4.3.1数据库的实现

本系统是面向中小型餐厅的点餐系统，所以同时访问它的数量也不会多，因此我们选择了MYSQL数据库，如果后期访问量增大，我们还可以更换为其他的数据库。在前面我们已经规划好了大致的表结构，这么我们就简单地描述一下怎么实现他们。

1. 订单表的实现。如图4-20所示：



图4-20 订单表DDL

1. 菜品订单表的实现。如图4-21所示：

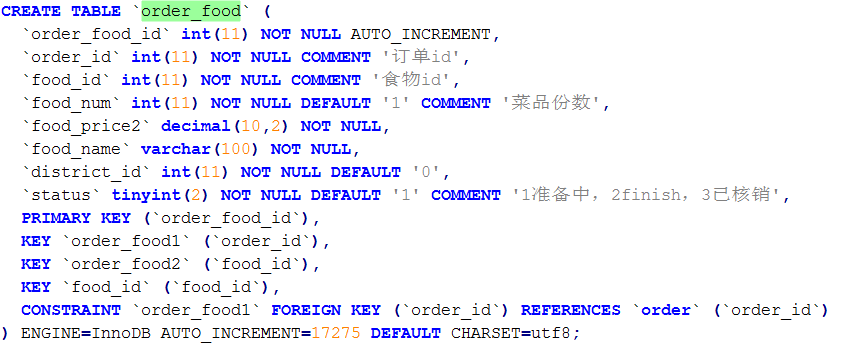


图4-21 菜品订单表DDL

1. 菜品属性订单数据表的实现。如图4-22所示：

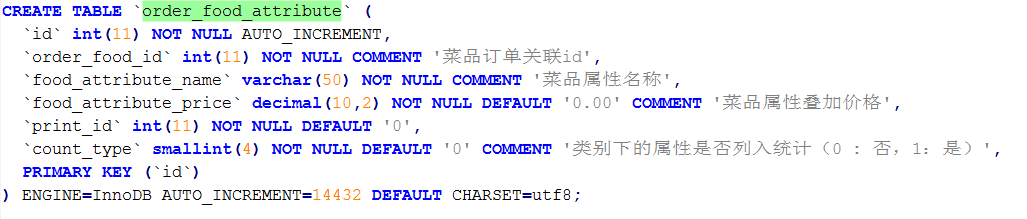


图4-22 菜品属性订单表DDL

1. 总代理管理员表的实现。如图4-23所示：

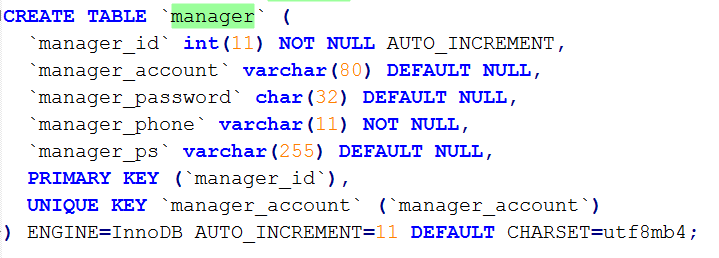


图4-23 总代理管理员表DDL

1. 代理信息表的实现。如图4-24所示：

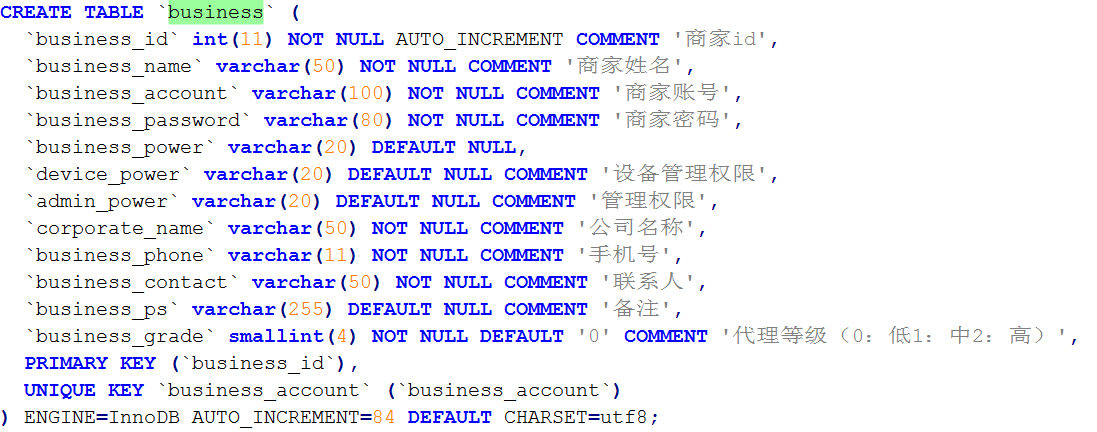


图4-24 代理信息表DDL

1. 店铺表的实现。如图4-25所示：



图4-25 商铺表DDL

1. 激活码表的实现。如图4-26所示：

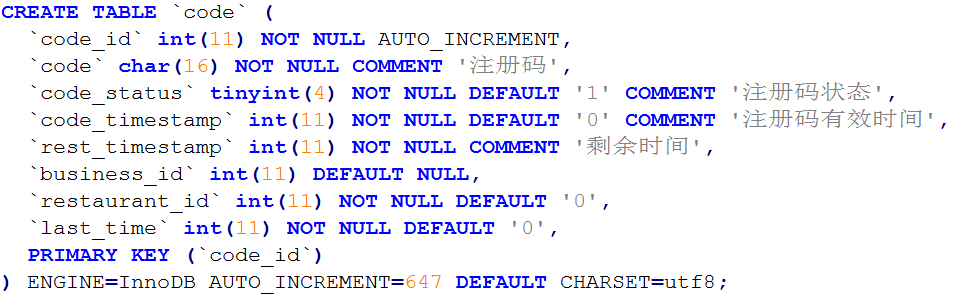


图4-27 激活码表DDL

1. 设备表的实现。如图4-27所示：

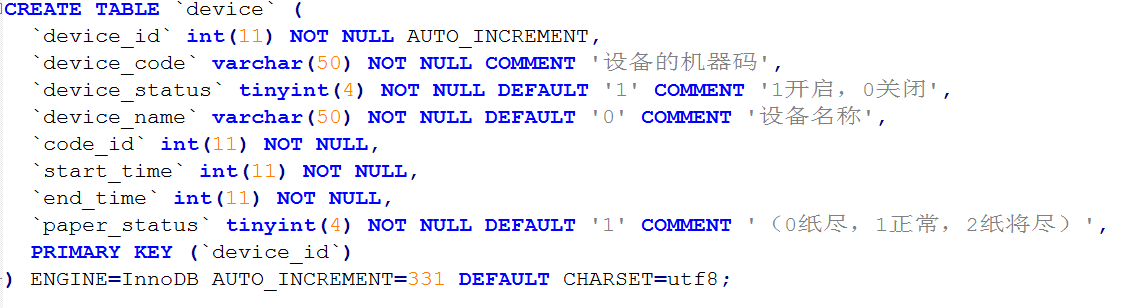


图4-27 设备表DDL

1. 地区表的实现。如图4-28所示：

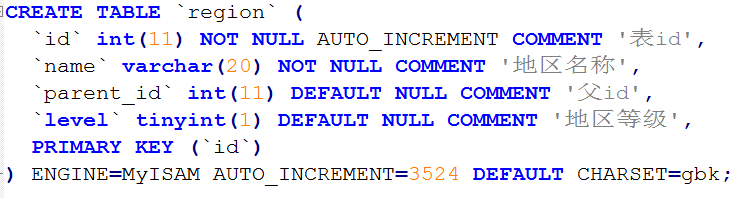


图4-28 地区表DDL

1. 续费表的实现。如图4-29所示：

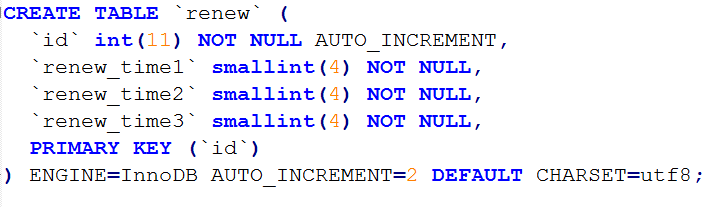


图4-29 续费表DDL

### 4.3.2部分模块的实现

由于系统代码比较多，这这里不可能一一罗列，因此我只是挑选了总代理登录的模块进行讲解。

在登录之前，我们要对总代理用户的账号、总代理用户的密码和验证码进行验证。

public function checklogin(){

if(!empty($\_GET)){

$verify = new \Think\Verify();

$Vresult = $verify->check($\_GET['code']);

if($Vresult){

$manager = D('manager');

$condition['manager\_account'] = I('get.username');

$result = $manager->where($condition)->find();

if($result){

if($result['manager\_password'] == I('get.pwd')){

session('manager\_account',$result['manager\_account']);

session('manager\_id',$result['manager\_id']);

$msg['msg'] = "登录成功!";

$msg['code'] = 0;

}else{

$msg['msg'] = "密码有误!";

$msg['code'] = 1;

}

}else{

$msg['msg'] = "用户不存在!";

$msg['code'] = 2;

}

}else{

$msg['msg'] = "验证码有错误!";

$msg['code'] = 3;

}

}else{

$msg['msg'] = "输入错误!";

$msg['code'] = 4;

}

exit(json\_encode($msg));

}

# 系统测试

亦称“测试”、“调试”。用测试数据对开发的计算机信息系统进行正确性验证的过程。主要目的是检测系统的各种要素(如程序、设备及数据库等)能否协调有效地发挥功能[[[9]](#endnote-8)]。

## 测试用例

软件测试是很有必要的，在现实中影响影响因素有很多，比如：测试所用的方法、使用的技术、测试人员的不稳定性以及软件自身开发时的所携带的问题。但是这些都是不可避免的，这就凸显出测试的重要性了。

如果我们有一个参照的测试用例，每个测试人员都按照上面的步骤一个个地测试，就可以保证测试的质量，将人为的测试误差最小化。

本点餐系统功能测试用例表如表5-1所示：

表5-1 测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试模块及功能名 | 测试内容 | 期望结果 |
| 1 | 总代理模块 | 能够实现登入登出；能够实现设备、注册码的增删改查；能够为品牌商分配注册码； | 可以实现 |
| 2 | 前台点餐模块 | 能够很好地展示出该商铺的所有菜品；能够点餐、下单和支付。 | 可以实现 |

## 测试结果

测试结果包括界面及功能测试和代码测试两部分，本系统对每个功能点做了详细的测试，测试结果正常[[[10]](#endnote-9)]。

一般情况下，测试结果主要包括两个部分，即功能的测试和代码的测试。

本系统的测试结果表如表5-2所示：

表5-2 系统功能测试结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 测试模块 | 功能点 | 质量保证标准 | 问题属性 | 测试结果 |
| F1 | 总代理模块 | 登录系统 | 正确性 | 故障 | 正常 |
| 品牌商管理 | 正确性 | 缺陷 | 正常 |
| 设备管理 | 正确性 | 故障 | 正常 |
| 管理员管理 | 正确性 | 故障 | 正常 |
| 注册码管理 | 正确性 | 故障 | 正常 |
| 退出系统 | 正确性 | 故障 | 正常 |
| F2 | 点餐模块 | 广告模块 | 正确性 | 故障 | 正常 |
| 选择就餐方式 | 正确性 | 故障 | 正常 |
| 菜品展示 | 正确性 | 故障 | 正常 |
| 购物车 | 正确性 | 故障 | 正常 |
| 下订单 | 正确性 | 故障 | 正常 |
| 支付模块 | 正确性 | 故障 | 正常 |

# 总结

本篇论文研究的是点餐系统，分为了四个子系统，分别是总代理系统、代理系统、商铺系统和点餐系统。它们之间的关系是上下级的关系，及总代理管辖代理，代理关系商铺，商铺展示菜品。

总代理生成代理，为代理分配注册码；代码生成店铺，为店铺分配注册码给设备；商铺将设备安装在餐馆中供给顾客点餐。无论是哪个子模块系统，都涉及到数据的增删改查。

最开始对这种代理模式的点餐系统不了解，导致在设计界面中遇到了很多麻烦，浪费了许多时间。但是这也算是一种开发经验的积累，下次遇到类似的问题应该可以好一些。

本系统在部署上还是存在一些问题的。比如：如何将前台模块部署到商铺的设备上，打印机的连接情况，支付宝和微信支付的申请问题，代码的不规范和代码重复量大的问题，还有一些没有考虑到的问题等等。因此本系统还有很大的进步空间。

在本次系统开发的过程中，我学习到很多，代码编程是一方面，其中业务的了解和团队的沟通交流也是至关重要的。

# 鸣 谢

在这次的毕设开发中，我要感谢我的小伙伴，我对需求不懂的地方，他都会给我细心讲解。在编码的过程中，遇到不能解决的问题都向他请教，都得到了及时地解决，没有耽误项目的进度。

通过这次毕设还让我明白了一些简单而朴实的道理。只有合作才能将事情做得更好，自己一个人的时间和精力是很有限的，要想把项目做好，还要依靠团队中所有人，只有所有人齐心协力，才可以发挥“1+1>2”的效果；做什么事情都要先规划好，要先学会思考，思考的差不多了，在动手开发，这样会事半功倍。比如在编码的过程中，我需要做一个搜索的功能，我会先想好它需要哪些功能，然后用草图和文件在笔记本上写出来，最后用编码一步步地实现；坚持就是胜利，一切都是“纸老虎”，不到最后绝不放弃。在开发这个系统的时候，我曾经有过气馁，甚至还想过放弃，但是我到底还是坚持下来了，最终顺利完成了系统的开发；

最后再一次感谢所有在这次毕业设计中曾经帮助过我的老师、朋友还有同学们，没有他们的支持我不可能完成这次毕业设计。老师帮我选择一个好的课堂，同学们为我提供了大量的帮助，比如查阅资料，遇到不懂的问题他们都会耐心地为我解答。还有，我要感谢被我引用使用的论文和文献。

# 

# 参考文献

1. [] 叶建农.叶建农:赋予“民以食为天”以新的诠释.前进论坛,2009-04-15.

   [↑](#endnote-ref-0)
2. [] 刘华星,杨庚.HTML5——下一代Web开发标准研究.计算机技术与发展,2011-08-10. [↑](#endnote-ref-1)
3. [] 漆昊晟,欧阳群.DIV+CSS网页布局技术初探.科技广场,2009-07-01. [↑](#endnote-ref-2)
4. [] David Flanagan.JavaScript: The Definitive Guide: Activate Your Web Pages, 6th edition. O'Reilly,2011-05-10. [↑](#endnote-ref-3)
5. [] Ryan Benedetti.Head First jQuery.O'Reilly深入浅出系列,2013. [↑](#endnote-ref-4)
6. [] 梁存良.基于PHP技术的高校学生管理系统设计与实现.河南科技学院学报(自然科学版),2005. [↑](#endnote-ref-5)
7. [] 王晓敏,邝孔武.信息系统分析与设计.清华大学出版社,2013. [↑](#endnote-ref-6)
8. [] 王珊,萨师煊.书籍款系统概论（第4版）.高等教育出版社,2013. [↑](#endnote-ref-7)
9. [] 余秉坚.中国会计百科全书 Chinese Accounting Encyclop Aedia.辽宁人民出版社,1999-03. [↑](#endnote-ref-8)
10. [] 王旭.黑龙江省道路运输证管理系统的设计与实现.互联网论文库,2010. [↑](#endnote-ref-9)