|  |
| --- |
| **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 | | | | |

目 录

[设计总说明 II](#_Toc135064045)

**[introduction](#_Toc135064046)** [II](#_Toc135064046)

[1 绪论 1](#_Toc452804152)

[1.1 为什么要将饮餐业作为研究对象？ 2](#_Toc452804156)

[1.2 开发点餐系统的目的在哪里？ 2](#_Toc452804157)

[2 技术支持和需求分析 4](#_Toc452804159)

[2.1 需要哪些技术支持？ 6](#_Toc452804168)

[2.1.1 HTML 6](#_Toc452804169)

[2.1.2 CSS 6](#_Toc452804170)

[2.1.3 JavaScript 6](#_Toc452804170)

[2.1.4 JQuery 6](#_Toc452804170)

[2.1.5 Bootstrap 6](#_Toc452804170)

[2.1.6 PHP 6](#_Toc452804170)

[2.1.7 ThinkPHP 6](#_Toc452804170)

[2.1.8 Mysql 6](#_Toc452804170)

[2.3 任务和环境部署 4](#_Toc452804160)

[2.4 功能分析 4](#_Toc452804160)

[2.4.1 后台总代理 6](#_Toc452804169)

[2.4.2 前台点餐平台 6](#_Toc452804170)

[2.5 数据的大致描述 4](#_Toc452804160)

[2.5.1 实体的描述 6](#_Toc452804169)

[2.5.2 数据库的描述 6](#_Toc452804170)

[3 系统设计 8](#_Toc452804176)

[3.1 系统内容结构 8](#_Toc452804178)

[3.2 数据库结构设计 9](#_Toc452804180)

[3.2.1 数据库逻辑结构设计 9](#_Toc452804181)

[3.2.2 数据库设计 11](#_Toc452804183)

[3.3 接口的设计 17](#_Toc452804191)

[4.3.1 管理员的接口 9](#_Toc452804181)

[4.3.1 设备的接口 9](#_Toc452804181)

[4.3.1 注册码的接口 9](#_Toc452804181)

[4.3.1 品牌商的接口 9](#_Toc452804181)

[4 系统详细设计和实现 14](#_Toc452804184)

[4.1 前台功能模块的实现 14](#_Toc452804185)

[4.1.1 广告 6](#_Toc452804169)

[4.1.2 就餐方式 6](#_Toc452804170)

[2.1.1 菜谱 6](#_Toc452804169)

[2.1.1 每道菜 6](#_Toc452804169)

[2.1.1 购物车 6](#_Toc452804169)

[2.1.1 支付方式 6](#_Toc452804169)

[4.2 总代理功能模块的实现 15](#_Toc452804186)

[2.1.1 品牌商管理 6](#_Toc452804169)

[2.1.1 设备管理 6](#_Toc452804169)

[2.1.1 管理员管理 6](#_Toc452804169)

[2.1.1 注册码管理 6](#_Toc452804169)

[4.3 系统的实现 15](#_Toc452804186)

[2.1.1 数据库的实现 6](#_Toc452804169)

[2.1.1 核心模块的实现 6](#_Toc452804169)

[5测试与结论 22](#_Toc452804196)

[5.1 测试 23](#_Toc452804198)

[5.2 适用范围 23](#_Toc452804199)

[6 总结 30](#_Toc452804201)

[鸣 谢 33](#_Toc452804206)

[参考文献 33](#_Toc452804206)

# 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]。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、大方、友好、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]。

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]。

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]。

JQuery是一个专门用来动态改变Web页面文档的JavaScript库[5]。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，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[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

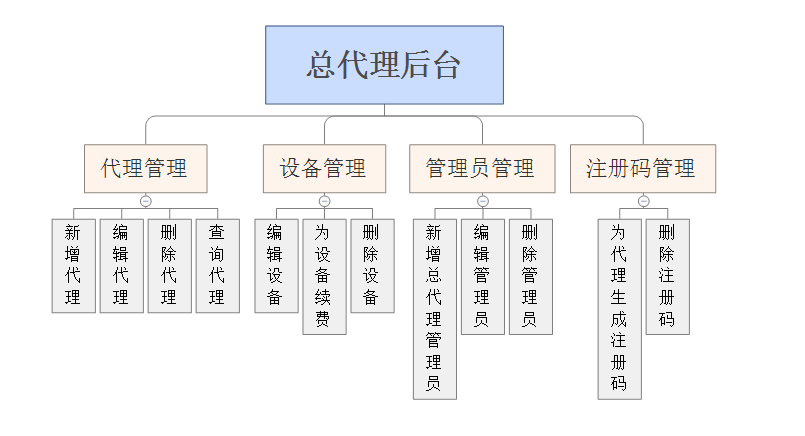


图3.2 总代理的功能模块图

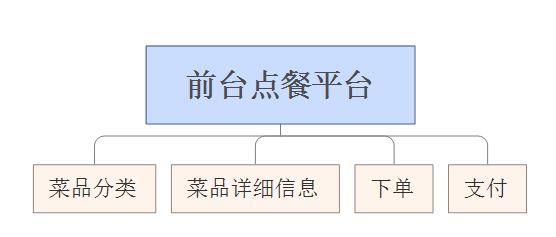


图3.3 前台点餐平台的功能模块图

## 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]。

### 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：

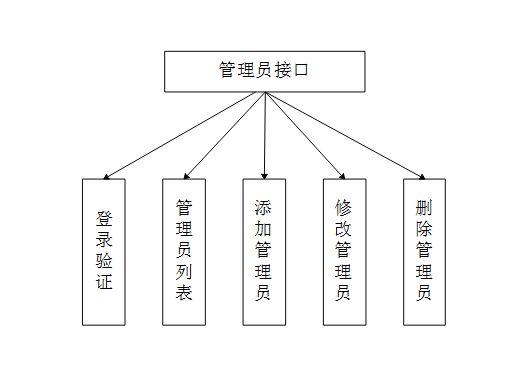


图3-4 总代理管理员接口图

### 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：

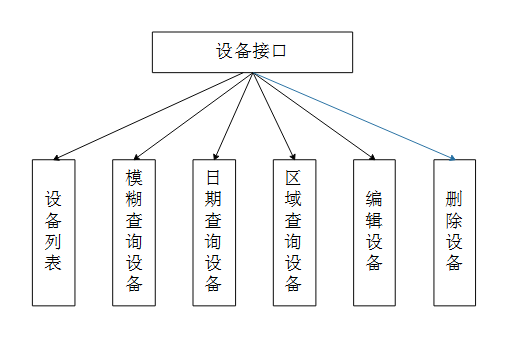


图3-5 总代理设备接口图

### 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：

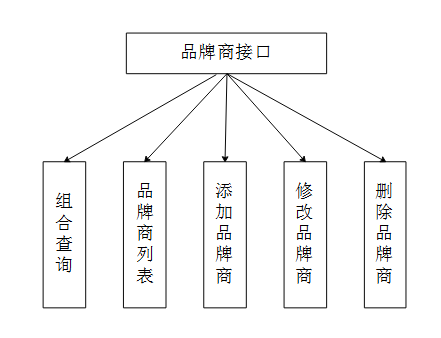


图3-6 总代理品牌商接口图

### 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：

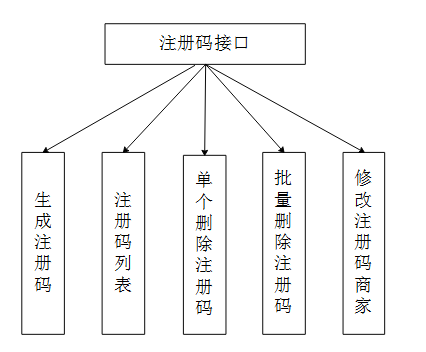


图3-7 总代理注册码接口图

# 系统总体需求分析

系统分析方法是指把要解决的问题作为一个系统，对系统要素进行综合分析，找出解决问题的可行方案的咨询方法。

按照一般的系统设计来说，一个系统通常有增删改查四个部分组成，这样有利于管理人员进行快速有效的操作。此外，对于复杂的系统来说，应该有模糊和组合搜索，这样可以提高查找的速度，节省时间。而且界面也要简洁大方，这样用户使用起来就可以和舒服了。

本次点餐系统我负责的主要是总代码模块和点餐模块，下面就简单介绍一下他们有哪些部分组成：

总代理模块主要面向的人群是总代理，包含了品牌商管理、设备管理、管理员管理和注册码。

品牌商管理，包括品牌商的名称、账号、等级、公司名称、联系人、手机号和备注。

设备管理：通过三级菜单的形式，一层层地向下查找，代理商->店铺名称->区域->某个具体的设备。设备的信息包括了设备名称、设备的MAC地址、注册时间、到期时间、设备状态和详细地址。

管理员管理：包括管理员的账号、密码、级别、品牌商的权限、设备权限、管理员权限、手机号和备注。

注册码管理：包括注册码、所属商家和状态。

点餐模块主要面向的人群是顾客。顾客通过餐厅的设备（点餐机）进入了点餐平台，然后选择点餐方式（店内用餐、打包带走），之后就进入了点餐首页进行订餐、下单和支付环节，完成点餐。

# 系统设计

# 系统设计

系统设计是根据需求规格为系统制定技术实现方案，即提出系统的物理模型[7]。

本系统采用ThinkPHP框架和Mysql数据库对网站进行后台的整体开发，为了使得系统能够安全性、适用性、易维护性，我们决定使用静态页面和动态页面两种方式混合开发。

本系统在设计时应该满足以下几个目标：

1.界面设计要简洁大方，让人看上去思路很清晰，没有冗余。

2.基本功能要完善，内容要充实，顾客可以很流畅地下单，商家可以很快的接单。

3.可扩展性强，可以随意添加总代理、代理、商铺和菜品。

4.安全性要得到保证，每个用户都要有自己的账号的密码。

5.要分权限设置，每个人的权限不同所能做的工作也将不一致。

6.代码的可读性和可维护性要得到保证，尽量优化代码。

## 系统体系结构

本次点餐系统采用的是C/S架构，每个商铺会有多台设备，改设备的MAC地址和代理给的注册码结合才能使用，而顾客可以直接在商铺进行点餐。

本系统共有四类用户：

总代理用户：这类用户可以对管理员进行操作；可以对品牌商（代理商）进行操作；可以对设备进行统一的管理；还可以为每个品牌商（代理商）分配注册码。

代理用户：这类用户可以对自己的子级商铺进行任意的增删改查；可以将总代理给自己的设备和注册码任意分配给指定的商铺；也可以统计自己的所有商铺的营业情况。

商铺用户：这类用户可以完善自己的商家信息；可以管理自己的设备；可以任意操作自己餐馆的菜品；可以通过系统查询自己商铺的营业情况；可以管理自己的打印机设备，用来打印小票；还可以对自己的设备进行模板设置；

顾客：虽然是顾客点餐，但是它也是商铺用户的前台显示。顾客来到商铺，在商铺的设备上进行点餐、下单和付款。

## 功能模块设计

层次模块结构是把一套大的系统划分为若干小的子系统，然后再将子系统划分为若干个子模块。

本系统一共分为是个子系统，分别为总代理后台、代理后台、店铺后台和点餐平台。

我负责的是总代理和点餐模块。接下来请看图4.1、4.2、4.3的功能模块图。



图4-1 点餐系统总的功能模块图

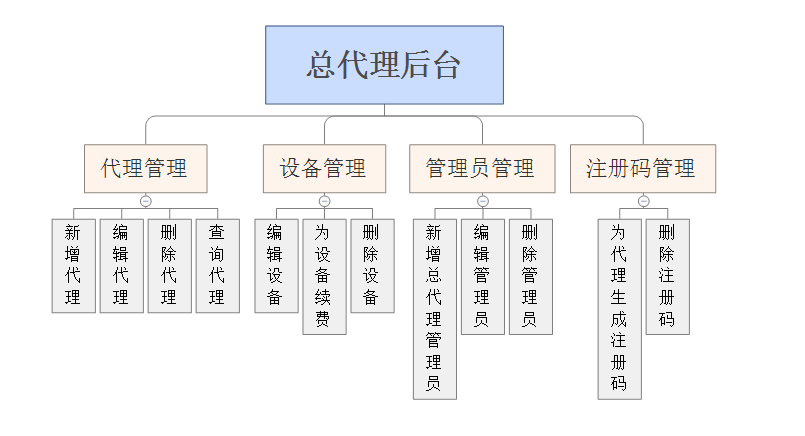


图4-2 总代理的功能模块图

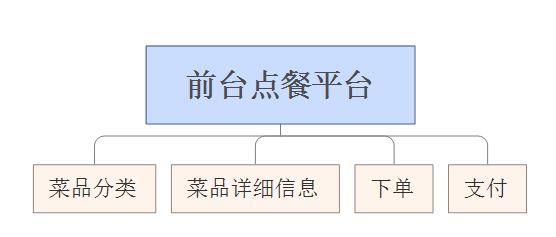


图4-3 前台点餐平台的功能模块图

### 总代理模块的功能设计

总代理模块可以说是整个系统的规划，它相当于是超级管理员，它对设备、管理员、品牌商和注册码有最高权限，它可以为品牌商添加设备和注册码，使得品牌商有自己的设备和注册码，然后品牌商可以将设备和注册码给自己下的商铺，商铺的设备的机器码和注册码匹配才可以运行。

下面会详细说明一下这些功能模块的功能：

1.管理员管理模块:为了方便整个系统的管理，超级管理员，即总代理可以生成多个管理员，这些管理员分为超级、高级、中级和低级管理员，这些管理员对品牌商和设备都有一定的权限。

2.品牌商管理模块：品牌商相当于代理，总代理管理员可以将自己的设备和注册码可以分配给品牌商（代理），品牌商（代理）就可以将设备和注册码分配给商铺，这样就可以减小总代理的压力，代理对总代理负责就可以了。

3.设备管理模块：所谓的设备管理，就是管理所以注册了改系统注册码的所有机器的MAC地址，只有MAC地址、注册码和有效期三者同时存在才能使得设备的系统正常运行。

4.注册码管理模块：在这里，总代理负责生成若干个注册码给品牌商（代理），代理有了这些注册码才能把它们给他们的下级商铺的设备，设备的MAC地址和注册码结合才能使用。

### 前台点餐平台模块的功能设计

前台模板模块主要是分界面来展示的：广告界面、就餐方式界面、点餐界面、支付界面。

## 数据库设计

### 数据库的详细设计

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]。

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4-1 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4-2 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4-3 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. restaurant\_manager（店铺管理员信息表），主要用来保存店铺管理员的信息。如表4-4所示：

表4-4 店铺管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| id | int | 11 | 是 | 主键id |
| manager\_name | varchar | 50 | 否 | 管理员名称 |
| login\_account | varchar | 100 | 否 | 登陆账号 |
| password | char | 32 | 否 | 密码 |
| business\_id | int | 11 | 否 | 代理id |
| restaurant\_id | int | 11 | 否 | 店铺id |
| manager\_phone | varchar | 11 | 否 | 电话 |
| manager\_ps | varchar | 255 | 否 | 备注 |

1. code（激活码表），保存设备激活码的详细信息。如表4-5所示：

表4-5 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4-6所示：

表4-3 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 |
| paper\_status | tinyint | 4 | 否 | 纸张情况 |

1. food\_category（菜品分类信息表），用来保存菜品分类的相关信息。如表4-7所示：

表4-7 菜品分类信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| food\_category\_id | int | 11 | 是 | 主键id |
| food\_category\_name | varchar | 50 | 否 | 分类名称 |
| restaurant\_id | int | 11 | 否 | 店铺id |
| sort | int | 11 | 否 | 排序 |

1. food（菜品信息表），主要用来保存菜品的数据信息。如表4-8所示：

表4-8 菜品信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| food\_id | int | 11 | 是 | 主键id |
| food\_name | varchar | 50 | 否 | 菜品名称 |
| food\_img | varchar | 255 | 否 | 菜品图片 |
| food\_price | decimal | 10 | 否 | 菜品价格 |
| star\_level | tinyint | 4 | 否 | 不同的星级 |
| hot\_level | tinyint | 4 | 否 | 辣的程度 |
| is\_prom | smallint | 1 | 否 | 是否开启时价 |
| foods\_num\_day | int | 11 | 否 | 每天供应份数 |
| food\_desc | varchar | 255 | 否 | 菜品描述 |
| restaurant\_id | int | 11 | 否 | 店铺id |
| is\_sale | tinyint | 4 | 否 | 是否上架 |
| print\_id | int | 11 | 否 | 打印机id |
| sort | int | 11 | 否 | 排序 |

1. food\_category\_relative（菜品与菜品分类关联表），用来保存店铺的相关信息主要用来保存菜品与菜品分类的关联的数据信息。如表4-9所示：

表4-9 菜品与菜品分类关联表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| id | int | 11 | 是 | 主键id |
| food\_id | int | 11 | 否 | 菜品id |
| food\_category\_id | int | 11 | 否 | 菜品分类id |

1. attribute\_type（属性类别表），主要用来保存菜品属性的类别。如表4-10所示：

表4-10 属性类别表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| attribute\_type\_id | int | 11 | 是 | 属性类别id |
| type\_name | varchar | 100 | 否 | 类别名称 |
| print\_id | int | 11 | 否 | 打印机id |
| restaurant\_id | int | 11 | 否 | 店铺id |
| select\_type | smallint | 4 | 否 | 属性单、多选 |
| food\_id | int | 11 | 否 | 所属菜品 |
| count\_type | smallint | 4 | 否 | 属性是否列入统计 |

1. food\_attribute（菜品属性表），要用来保存菜品的属性。如表4-11所示：

表4-11 菜品属性表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| food\_attribute\_id | int | 11 | 是 | 属性id |
| attribute\_name | varchar | 50 | 否 | 属性名 |
| attribute\_price | decimal | 10 | 否 | 属性价格 |
| attribute\_type\_id | int | 11 | 否 | 属性类别id |

1. order（订单数据表），主要用来存储订单数据。如表4-12所示：

表4-12 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4-13所示：

表4-13 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4-14所示：

表4-14 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 |

# 系统实现

本系统一个是有4个模块组成，总代理模块、代理模块、商铺后台模块和前台点餐模块。由于我是和另一个小伙伴一起写的，本人主要负责总代理模块和前台点餐模块，这样我就对这两部分的实现做一个详细的说明。

## 前台功能模块的实现

前台功能模块分为注册机器码模块、广告模块、就餐方式模块、菜单模块、购物车模块、下单模块和支付模块。接下来我来看看下面这几个部分：

1.广告界面：用于放一些广告图片；

2.选择就餐方式界面：用户能够在这里选择自己喜欢的就餐方式，打包带走还是在餐厅里吃；

3.自助点餐界面：用户可以按照类别选择自己喜欢的菜肴，同时可以选择每道菜的属性（配菜）；

4.购物车：用户选中了的菜会自动添加到购物车中，然后进行下单；

5.支付界面：用户可以选择支付宝、微信或者现金支付。

广告界面运行页面如图5-1所示：



图5-1 广告界面

选择就餐方式页面如图5-2所示：



图5-2 选择就餐方式界面

自助点餐界面页面如图5-3所示：



图5-3 自助点餐界面页

支付页面如图5-4所示：



图5-4 支付页面

## 总代理功能模块的实现

总代理子系统能包括管理员身份验证、添加品牌商模块、设备管理模块、添加管理员模块和注册码管理模块等。

总代理管理员登录时需要输入相应的信息（在本系统中，需要输入总代理的账号、总代理的密码和系统自动生成的随机验证码），如果任何一项不正确都会有错误提示信息，如果正确则可以登录到后台总代理管理模块首页。

管理员登录页面如图5-5所示：



图5-5 登录页面

管理员登录的流程图如图5-6所示：

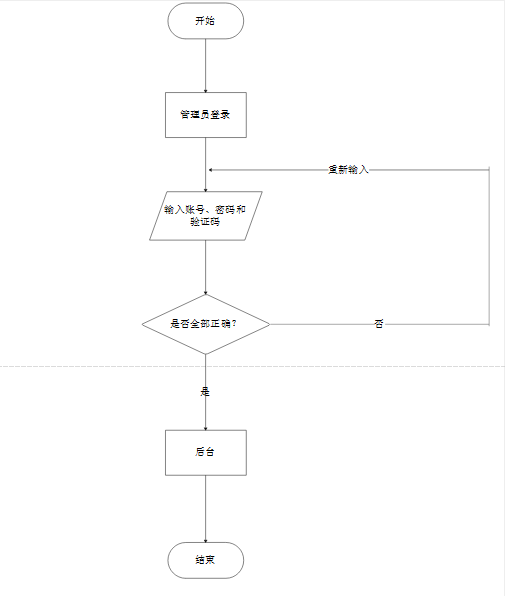


图5-6 管理员登录流程图

品牌商管理功能，用户登入后台总代理页面后，点击“品牌商管理”之后，会出现管理品牌商的页面。在这里我们可以对品牌商进行录入、查看、修改、删除和查询，详细如图5-7所示：



图5-6 品牌商管理界面

后台管理系统会对所有的设备进行统一化管理，在页面的左边是设备的代理，点击任意代理可以看到改代理下的商铺所在区域下的所有设备的运行情况。 详细如图5-7所示：

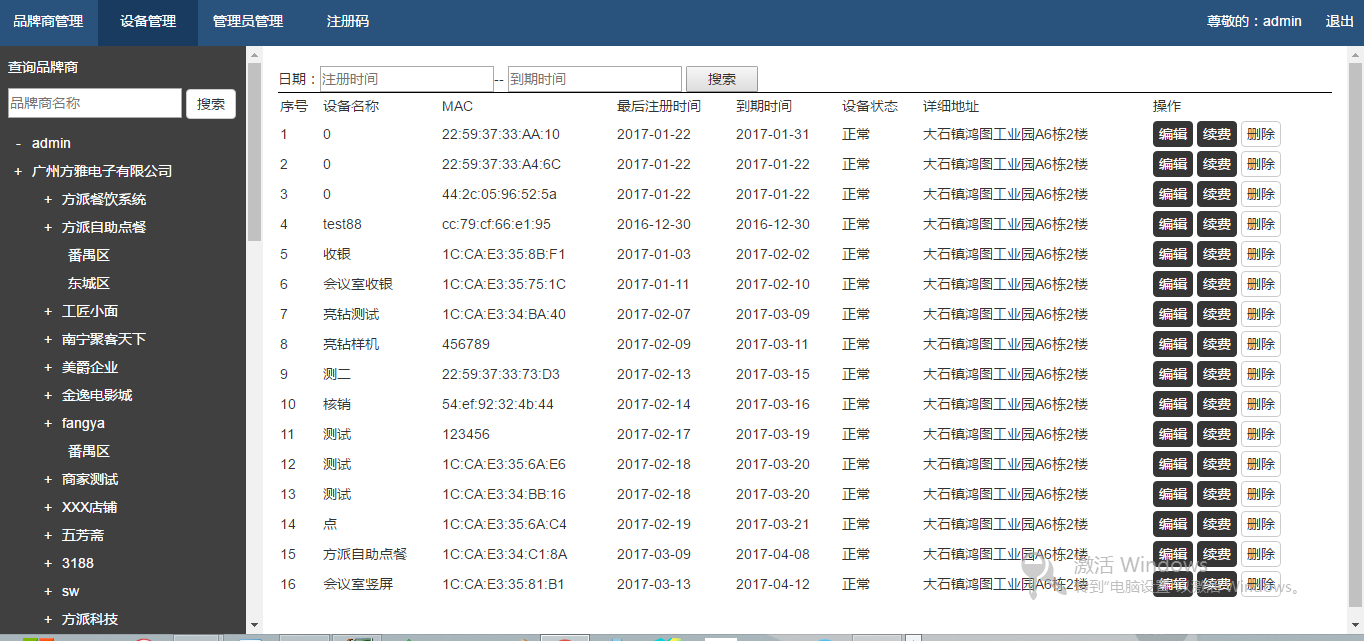


图5-7 设备管理界面

管理员模块，可以添加多个不同级别的管理员，这些管理员对管理设备和注册码的权限不一致。如图5-8所示：

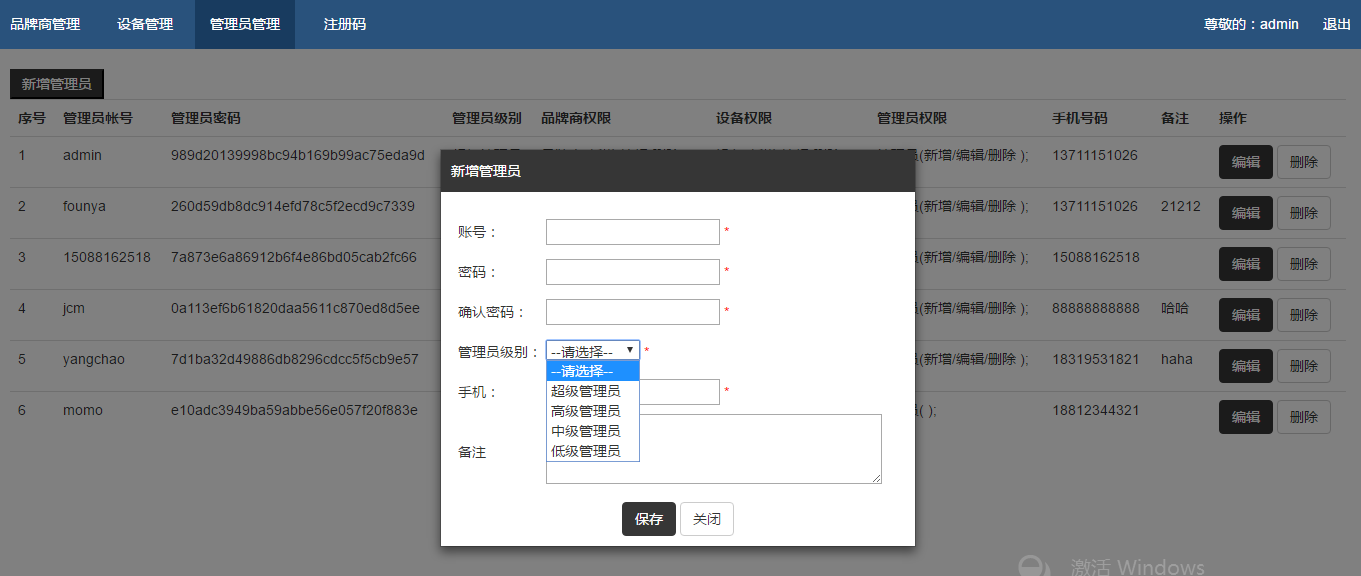


图5-8 管理员管理界面

注册码管理，所有的设备只有将自己机器的MAC地址和注册码绑定在有效期内才能正常地使用点餐系统。在这里，管理员可以随意生个1个或者10个注册码给相应的代理，然后代理再把注册码分配给自己所管辖的商铺的设备。详情如图5-9所示：

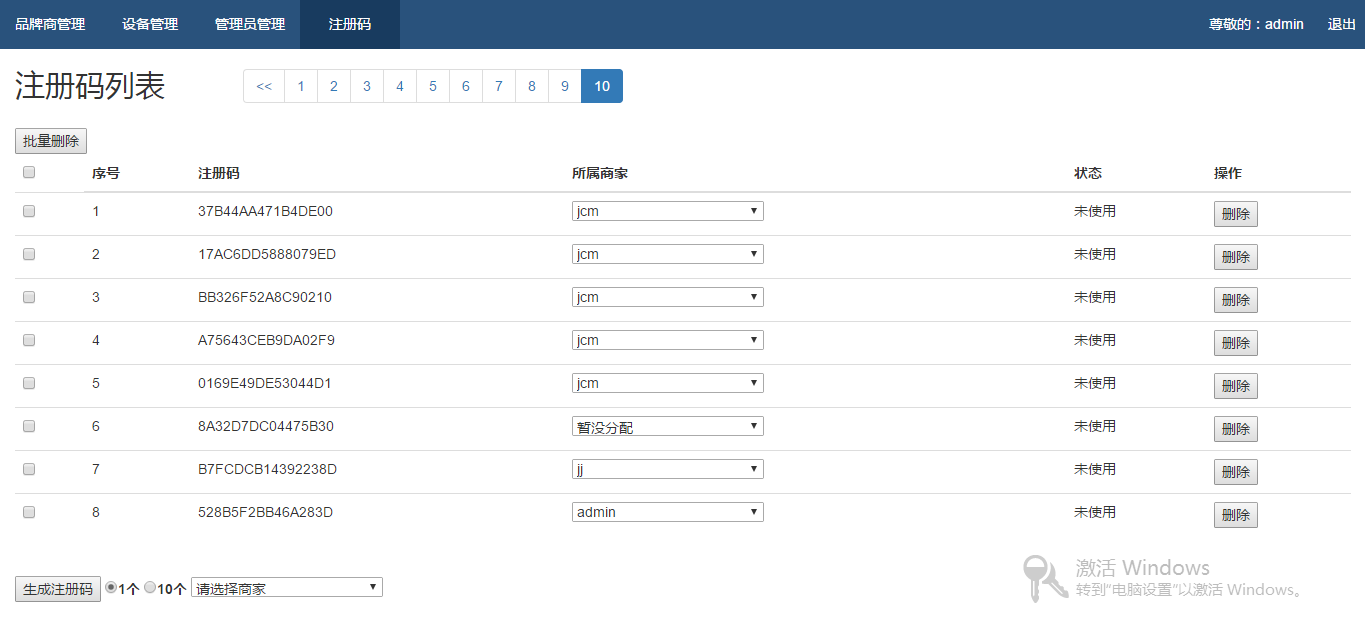


图5-9 注册码管理界面

# 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]。

## 系统整体测试步骤

软件测试一般有三个方便，单元测试、集成测试和确认测试。

1. 单元测试

单元测试就是将被测试的模块化为为最小化，然后与其他的单元程序进行隔离来测试。

明确单元测试的内容,是保证单元测试质量的基本要求[10]。

2．集成测试

集成测试就是将所有的小模块拼装起来形成一个系统，然后再对这个系统进行集成测试。

3．确认测试

在经过单元测试和集成测试后，我们需要验证是否与用户的需要是否一致，这就是确认测试。

## 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6-1所示：

表6-1 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。测试结果表如表6-2所示：

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6-2所示：

表6-2 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道理。一个人的能力毕竟是很有限的，只有合作才能将事情做得更好；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；坚持就是胜利，一切都是“纸老虎”，不到最后绝不放弃；

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，以及在设计中被我引用或参考的论文和著作，表示由衷的感谢。

# 

# 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 .

[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.

[3] 漆昊晟,欧阳群.DIV+CSS网页布局技术初探.科技广场,2009-07-01.

[4] David Flanagan.JavaScript: The Definitive Guide: Activate Your Web Pages, 6th edition. O'Reilly,2011-05-10.

[5] Ryan Benedetti.Head First jQuery.O'Reilly深入浅出系列,2013.

[6] 葛焱东,罗衍清.基于PHP的网页动态生成技术. 黑龙江科技信息,2000-11-15.

[7] 王晓敏,邝孔武.信息系统分析与设计.清华大学出版社,2013.

[8] 王珊,萨师煊.书籍款系统概论（第4版）.高等教育出版社,2013.

[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.

[10] 蔡高亮.软件单元测试.信息技术与标准化,2008-02-10.

附录一：部分前端代码

order.html:

<include file="Public/head"/>  
<body class="order-body">  
<input type="hidden" name="isOpenNum" id="isOpenNum" value="{$Think.session.isOpenNum}">  
<input type="hidden" name="tpl" value="{$tpl}" id="tpl"/>  
 <header class="home-header">  
 <a href="\_\_CONTROLLER\_\_/index" class="order-return">  
 <img src="\_\_PUBLIC\_\_/images/lt.png">  
 <span>重新点餐</span>  
 </a>  
 <div class="text-center order-head">海涵自助点餐系统</div>  
 </header>  
 <div class="order">  
 <!-- 左边分类 start -->  
 <div class="order-left">  
 <ul class="sorts-list">  
 <volist name = "info" id = "v">  
 <li>  
 <a href="" data-toggle="tab" id="category\_type" onclick="showtypefood({$v.food\_category\_id})">  
 <span>{$v.food\_category\_name}</span>  
 </a>  
 </li>  
 </volist>  
 </ul>  
 </div> <!-- 左边分类 end -->  
 <!-- 中间菜品列表 start -->  
 <div class="order-content">  
 <div class="container-fluid">  
 <div class="tab-content" id="food\_info">  
 <div class="tab-pane fade in active" id="recommend">  
 <div class="row">  
 <volist name = "info1" id = "v1">  
 <div class="col-sm-6 col-md-4">  
 <a href="javascript:void(0)" class="food-item" data-toggle="modal"

data-target="#foodModal" data-food\_id="{$v1.food\_id}" onclick="findfoodinfo(this)">  
 <div class="food-box">  
 <div class="food-box-img">  
 <img src="/{$v1.food\_img}">  
 </div>  
 <div class="clearfix">  
 <div class="pull-left">  
 <div class="star">  
 <if condition = "$v1.star\_level eq 1">  
 <span></span>  
 </if>  
 <if condition = "$v1.star\_level eq 2">  
 <span>★★</span>  
 </if>  
 <if condition = "$v1.star\_level eq 3">  
 <span>★★★</span>  
 </if>  
 <if condition = "$v1.star\_level eq 4">  
 <span>★★★★</span>  
 </if>  
 <if condition = "$v1.star\_level eq 5">  
 <span>★★★★★</span>  
 </if>  
 </div>  
 </div>  
 <div class="pull-right text-right">  
 <div class="cayenne">  
 <if condition = "$v1.hot\_level eq 0">  
  
 </if>  
 <if condition = "$v1.hot\_level eq 1">  
 <!-- <img src="\_\_PUBLIC\_\_/images/cayenne.png"> -->  
 </if>  
 <if condition = "$v1.hot\_level eq 2">  
 <img src="\_\_PUBLIC\_\_/images/cayenne.png">  
 <img src="\_\_PUBLIC\_\_/images/cayenne.png">  
 </if>  
 <if condition = "$v1.hot\_level eq 3">  
 <img src="\_\_PUBLIC\_\_/images/cayenne.png">  
 <img src="\_\_PUBLIC\_\_/images/cayenne.png">  
 <img src="\_\_PUBLIC\_\_/images/cayenne.png">  
 </if>  
 </div>  
 </div>  
 </div>  
 </div>  
 <div class="food-details">  
 <div class="food-name">{$v1.food\_name}</div>  
 <div class="food-price">&yen;{$v1.food\_price}</div>  
 </div>  
 </a>  
 </div>  
 </volist>  
 </div>  
 </div>  
 </div>  
 </div>  
 </div><!-- 菜品列表 end -->  
  
 <!-- 右边已选菜品 start -->  
 <div class="order-right">  
 <div class="container-fluid">  
 <!-- 菜品选择列表 start -->  
 <div class="food-select" id="foodlist">  
 </div><!-- 菜品选择列表 end -->  
 </div>  
 <div class="total" >  
 <p>合计：&yen;<span id="Total">0.00</span>元</p>  
 <button class="common-btn place-order-btn" onclick="PlaceOrder2()">  
 <span>立即下单</span>  
 <img src="\_\_PUBLIC\_\_/images/gt.png">  
 </button>  
 </div>  
 </div><!-- 右边已选菜品 end -->  
 </div>  
  
 <!-- 菜品Modal -->  
 <div class="modal fade" id="foodModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">  
 <div class="food-modal-dialog" id="modelfood">  
 </div>  
 </div>  
 <!-- 餐桌Modal -->  
 <div class="modal" id="tableModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">  
 <div class="modal-dialog">  
 <div class="tableModal-return">  
 <button class="btn-none" data-dismiss="modal">  
 <span>&lt;</span>  
 <span>返回点餐</span>  
 </button>  
 </div>  
 <div class="number text-center">  
 <div class="number-tips">1.领取点餐机旁边的餐牌号</div>  
 <div class="number-tips">2.输入餐牌号，按确认</div>  
 <div class="number-input">  
 <input type="text" id="numtext" data-order\_type="{$Think.get.order\_type}" value="" readonly="readonly">  
   
 <button id="btn" onclick="placeor()">确认</button>  
 </div>  
 <div class="number-table clearfix">  
 <div class="pull-left">  
 <table>  
 <tr>  
 <td>  
 <button class="num-btn">1</button>  
 </td>  
 <td>  
 <button class="num-btn">2</button>  
 </td>  
 <td>  
 <button class="num-btn">3</button>  
 </td>  
 </tr>  
 <tr>  
 <td>  
 <button class="num-btn">4</button>  
 </td>  
 <td>  
 <button class="num-btn">5</button>  
 </td>  
 <td>  
 <button class="num-btn">6</button>  
 </td>  
 </tr>  
 <tr>  
 <td>  
 <button class="num-btn">7</button>  
 </td>  
 <td>  
 <button class="num-btn">8</button>  
 </td>  
 <td>  
 <button class="num-btn">9</button>  
 </td>  
 </tr>  
 </table>  
 </div>  
 <div class="pull-left number-table-right">  
 <table>  
 <tr>  
 <td>  
 <button class="num-btn">0</button>  
 </td>  
 </tr>  
 <tr>  
 <td>  
 <button id="del-num">  
 <p>删</p>除  
 </button>  
 </td>  
 </tr>  
 </table>  
 </div>  
 </div>   
 </div>  
 </div>  
 </div>  
</body>  
<script src="\_\_PUBLIC\_\_/js/Home/order.js"></script>  
</html>

附录二：部分后台代码

IndexController.class.php

<?php  
namespace Home\Controller;  
use Think\Controller;  
class IndexController extends Controller  
{

public function order(){  
 if ($this->is\_security){  
 $this->is\_security = false;  
 $tableNum = I("get.tablenum");  
 $orderType = I("get.order\_type");   
 if($tableNum){  
 session("tableNum",$tableNum);  
 }  
 if($orderType){  
 session("orderType",$orderType);  
 }  
 //判断选择餐牌号的页面是否开启  
 $restaurant\_process\_model = D("restaurant\_process");  
 $condition["process\_id"] = 4; //级别大于当前流程页   
 $condition["restaurant\_id"] = session("restaurant\_id");  
 $isOpenNum = $restaurant\_process\_model->where($condition)->field("process\_status")->find()['process\_status'];  
 session("isOpenNum",$isOpenNum);  
  
//-------------------------------------------菜品分类信息---------------------------------------  
 $food\_category = D('food\_category');  
 $category\_time = D('category\_time');  
 $condition['restaurant\_id'] = session('restaurant\_id');  
 $condition['is\_timing'] = 0; # 是否定时  
 $arr = $food\_category->where($condition)->order('sort asc')->select();   
   
 $where['restaurant\_id'] = session('restaurant\_id');  
 $where['is\_timing'] = 1;  
 $food\_categoryIdList = $food\_category->where($where)->field('food\_category\_id')->select(); if($food\_categoryIdList){   
 $food\_categoryNewIdList = array();

foreach($food\_categoryIdList as $foodvv){  
 $food\_categoryNewIdList[] = $foodvv['food\_category\_id'];  
 }  
 //第一种时间段的定时查询   
 $current\_time = time();  
 $t\_condition['time1'] = array("lt",$current\_time);  
 $t\_condition['time2'] = array("gt",$current\_time);// time1<$current\_time<time2  
 $t\_condition['category\_id'] = array("in",$food\_categoryNewIdList);  
 # 条件是当前时间要在开始时间和结束时间之内 并且分类要开启了定时 获取它的分类ID  
 $category\_ids = $category\_time->where($t\_condition)->distinct("category\_id")->field("category\_id")->select();  
 if($category\_ids){ //存在时间段定时记录(当前店铺)  
 $category\_id\_list = array();  
 foreach ($category\_ids as $k => $v) { # 将分类ID再遍历出来，放到另外一个数组里面  
 $index = "cid" . $v['category\_id'];  
 $category\_id\_list[$index] = $v['category\_id'];  
 }  
 }  
 //第二种星期段的定时查询   
 $current\_week = date("w");  
 $ftg\_condition['timing\_day'] = array("like", "%" . $current\_week . "%");  
 $ftg\_condition['food\_category\_id'] = array("in",$food\_categoryNewIdList);  
 $category\_timing\_model = D("food\_category\_timing");  
 $category\_ids2 = $category\_timing\_model->where($ftg\_condition)->distinct("food\_category\_id")->field("food\_category\_id,start\_time,end\_time")->select();   
 $category\_id\_list2 = array();  
 if($category\_ids2){ //存在星期段定时记录(当前店铺)  
 foreach ($category\_ids2 as $kk => $vv) {  
 $start\_time = strtotime($vv['start\_time']);  
 $end\_time = strtotime($vv['end\_time']);  
 if($start\_time < $current\_time && $end\_time > $current\_time){  
 $index = "cid" . $vv["food\_category\_id"];  
 $category\_id\_list2[$index] = $vv["food\_category\_id"];  
 }  
 }  
 }  
  
 //两种定时情况结果合并  
 if($category\_id\_list == null){  
 $categoryIdsList = $category\_id\_list2;  
 }else if($category\_id\_list2 == null){  
 $categoryIdsList = $category\_id\_list;  
 }else{  
 $categoryIdsList = array\_merge($category\_id\_list, $category\_id\_list2);  
 }  
 $lastCategoryIdsList = array();  
 foreach ($categoryIdsList as $vvv) {  
 $lastCategoryIdsList[] = $vvv;  
 }  
 if($lastCategoryIdsList){   
 $l\_condition['food\_category\_id'] = array("in", $lastCategoryIdsList);  
 $arr2 = $food\_category->where($l\_condition)->order('sort asc')->select();  
 $arr = array\_merge($arr, $arr2);   
 }  
 }  
 $this->assign("info", $arr);   
  
//--------------------------------------------菜品信息-----------------------------------------   
 $food = D('food');  
 $food\_category\_relative = D('food\_category\_relative');   
 if($arr){   
 $foodIdArr = array();   
 foreach($arr as $vinfo){  
 $where1['food\_category\_id'] = $vinfo['food\_category\_id'];  
 $foodIdList = $food\_category\_relative->where($where1)->field('food\_id')->select(); #　在食物与食物分类表中根据分类ＩＤ查询食物ＩＤ  
 foreach($foodIdList as $fil){  
 // 先判断关于该食物ID的订单在今天内所对应的份数是否已经超过额定的份数  
 $start=mktime(0,0,0,date("m"),date("d"),date("Y")); //当天开启时间  
 $end=mktime(0,0,0,date('m'),date('d')+1,date('Y'))-1; //当天结束时间  
  
 $Model = M(); // 实例化一个model对象 没有对应任何数据表  
 $num = $Model->query(" select t1.food\_num as num from order\_food t1 inner join  
 `order` t2 on t1.order\_id = t2.order\_id and t1.food\_id = $fil[food\_id] and t2.order\_status in ('3','11','12')  
 and t2.pay\_time between $start and $end");  
  
 if($num){  
 $sum = 0;  
 foreach($num as $n){  
 $sum += $n['num'];  
 }  
 // 查询出该food\_id对应多少限额  
 $fit\_num = D("food")->where(array("food\_id"=>$fil['food\_id']))->getField("foods\_num\_day");  
 if($sum < $fit\_num){  
 $foodIdArr[] = $fil['food\_id'];   
 }  
 }else{  
 $foodIdArr[] = $fil['food\_id'];   
 }  
 }  
 }  
   
 if($foodIdArr){   
 $f\_condition['is\_sale'] = 1; //1:上架  
 $f\_condition['food\_id'] = array("in",$foodIdArr);  
  
 $arr1 = $food->where($f\_condition)->order('sort asc')->select();  
  
 $prom = D('prom');   
 foreach($arr1 as $k1=>$v1){  
 if($v1['is\_prom'] == 1){   
 $where2['prom\_id'] = $v1['food\_id'];   
 $when\_time = time();  
 $where2['prom\_start\_time'] = array("lt",$when\_time);   
 $where2['prom\_end\_time'] = array("gt",$when\_time);  
 $prom\_price = $prom->where($where2)->field('prom\_price')->find()['prom\_price'];  
 if($prom\_price){   
 $food\_price = $prom\_price;   
 }else{  
 $food\_price = $v1['food\_price'];   
 }  
 }else{  
 $food\_price = $v1['food\_price'];  
 }  
 $arr1[$k1]['food\_price'] = $food\_price;  
 }  
 }  
 }  
 $this->assign("info1", $arr1);  
 $this->assign("tpl",change\_telcolor());  
 $this->display("order");  
  
 }else{  
 $this->overdue();  
 exit;  
 }  
}

}