

黑龙江大学

“软件工程专业综合实践”总结报告

论文题目:	餐饮管理系统			
学 院:	软件学院			
年 级:	2015 级			
专 业:	软件工程			
指导教师:	丁大勇老师			
参与人员	(1)	(2)	(3)	(4)
学号:	20154471	20156658	20156661	20155394
姓名:	毕浩宇	朱银松	舒永吉	国家兴

2018 年 6 月 29 日

摘要

本项目的开发是为了为商户和用户提供一个订餐平台,用于想要提供外卖服务的餐馆以及想要提供送餐服务个别公司内部餐厅提供一个订餐平台。本产品为用户提供了订餐、查看菜品、查看商家、订单追踪、差评率等信息。为商户提供了接受订单、活动管理、收益查看等功能集于一体的餐饮管理系统。对于普通商户提供了前后台的点餐服务,为集团商户提供了供销用户管理。市面上类似产品较多且参差不齐,本产品的集合了市面上大部分产品的优势(包括语音提示、点餐等)并完善了已有优势,增加了集团商户的供销用户管理模块,为类似于学校食堂提供了一个更好的平台。

关键词

关键词 1: 餐饮管理; 关键词 2: 食堂外卖; 关键词 3: 订单管理

目录

摘要	I
第 1 章 绪论	1
1.1 课题研究的目的是和意义	1
1.2 课题研究的背景	1
1.3 系统开发环境简介	2
1.4 论文的工作内容及论文的结构	4
第二章 需求分析	5
2.1 事件表建立。	5
2.2 建立系统用例模型	7
2.3 对象模型建立	33
第三章 系统设计	34
3.1 软件结构图	34
3.2 过程设计	36
3.3 数据库设计	45
第四章 系统的实现	49
4.1 数据库的相关实现	49
4.2 登录注册模块	50
4.3 商户管理模块	58
4.4 用户模块	67
4.5 本章小结	81
第五章 系统测试	82
5.1 测试概要	82
5.2 测试执行情况	82
5.3 测试用例	83
5.5 测试结果与结论	112
总结	115
参考文献	116
致谢	116

第 1 章 绪论

1.1 课题研究的目的和意义

本项目的开发是为了为商户和用户提供一个订餐平台,用于普通想要提供外卖服务的餐馆以及想要提供送餐服务个别公司内部餐厅提供一个订餐平台。

本产品为用户提供了订餐、查看菜品、查看商家、订单追踪、差评率等信息。为商户提供了接受订单、活动管理、收益查看、原料管理。对于普通商户提供了前后台的点餐服务,为集团商户提供了供销用户管理。

1.2 课题研究的背景

我国餐饮业的餐饮信息管理系统的开发,规模大小各不相同,开发方法、模式、平台也不尽相同。有的是购买的系统,有的是自己开发,有的还进行了多次开发,因为开发人员和方法的原因,系统维护和升级都比较困难。信息共享性差,数据冗余大,维护困难。以前的餐饮管理信息系统一般采用面向过程、面向数据的开发方法或者采用传统的面向对象的开发方法,开发的系统一般是基于单机或局域网,系统的结构是单机或 C/S 结构,开发使用的软件或平台多为简单的关系数据库管理系统。不断增长的复杂性、多样性和相互关联性是当今管理信息系统的共同特征。传统的面向功能开发方法,已逐渐不适应越来越复杂和多变的需求,其成果不能得到很好的重用。而面向对象技术能够有效地控制复杂性、适应多变性,具有灵活性、低风险性、可重用性,从而能提高系统开发的效率。

随着 Internet 技术的迅速发展, B/S 体系结构技术和分布式计算技术的逐渐成熟。开发基于新的硬件基础和软件开发方法的餐饮管理信息系统成为可能。UML 是对面向对象系统建模的标准语言。从面向功能的开发方法转向使用 UML 的面向对象开发方法是信息系统开发的趋势。RUP 是基于 UML 的新一代面向对象开发方法,和以往的面向对象方法相比,其迭代和增量开发思想更适合软件系统的开发。中小型饭店管理系统本人主要完成销售管理子系统的分析、设计及实现

1.3 系统开发环境简介

本系统采用 B/S 结构，运行平台为 Windows XP 以上，系统开发平台为 MyEclipse 2017，系统开发语言为 JSP，系统后台数据库为 MySQL，分辨率最佳效果为 1366×768。

1.3.1 B/S 结构的数据库访问模式

在 B/S 体系结构系统中，用户通过浏览器向分布在网络上的许多服务器发出请求，服务器对浏览器的请求进行处理，将用户所需信息返回到浏览器。B/S 结构简化了客户机的工作，客户机上只需配置少量的客户端软件。服务器将担负更多的工作，对数据库的访问和应用程序的执行将在服务器上完成。浏览器发出请求，而其余如数据请求、加工、结果返回以及动态网页生成等工作全部由 Web Server 完成。实际上 B/S 体系结构是把二层 C/S 结构的事务处理逻辑模块从客户机的任务中分离出来，由 Web 服务器单独组成一层来负担其任务，这样客户机的压力减轻了，把负荷分配给了 Web 服务器。B/S 三层体系结构如图 1-1 所示。



图 1-1 B/S 三层架构示意图

这种结构不仅把客户机从沉重的负担和不断对其提高的性能的要求中解放出来，也把技术维护人员从繁重的维护升级工作中解脱出来。由于客户机把事务处理逻辑部分分给了功能服务器，使客户机一下子“苗条”了许多，不再负责处理复杂计算和数据访问等关键事务，只负责显示部分，所以维护人员不再为程序的维护工作奔波于每个客户机之间，而把主要精力放在功能服务器上程序的更新工作。这种三层结构在层与层之间相互独立，任何一层的改变不会影响其它层的功能。

1.3.2 MyEclipse 简介

MyEclipse 企业级工作平台（MyEclipse Enterprise Workbench，简称 MyEclipse）是对 Eclipse IDE 的扩展，利用它我们可以在数据库和 JavaEE 的开发、发布，以及应用程序服务器的整合方面极大的提高工作效率。它是功能丰富的 JavaEE 集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持 HTML, CSS, Javascript, MySQL。

1.3.3 JSP 动态网络技术与 Tomcat 简介

JSP（JavaServer Pages）是 Sun 公司推出的一种动态网页技术。JSP 技术是以 Java 语言作为脚本语言的，熟悉 JAVA 语言的人可以很快上手。

JSP 本身虽然也是脚本语言，但是却和 PHP、ASP 有着本质的区别。PHP 和 ASP 都是由语言引擎解释执行程序代码，而 JSP 代码却被编译成 Servlet 并由 Java 虚拟机执行，这种编译操作仅在对 JSP 页面的第一次请求时发生。因此普遍认为 JSP 的执行效率比 PHP 和 ASP 都高。

1.4 论文的工作内容及论文的结构

1.4.1 论文的工作内容

1. 项目开发计划书
2. 需求规格说明书
3. 系统设计说明书
4. 用户页面
5. 商户页面
6. 服务器程序
7. 软件测试

1.4.2 论文的结构

1. 绪论
2. 需求分析
3. 系统设计
4. 系统实现
5. 系统测试

第二章 需求分析

本章将对餐饮管理系统进行需求分析，并确定出餐饮管理系统的基本数据需求、功能需求和性能需求，为以后系统的设计与实现提供依据。主要建立系统的事件表、用例图、用例描述、对象关系模型和顺序图等。

2.1 事件表建立。

2.1.1 识别系统相关者

采用参与者词汇表进行描述，如表 2.1 所示。

表 2.1 参与者词汇表

序号	词汇	同义词	描述
1	用户	使用者、供给用户	使用平台购买菜品的人
2	商户	使用者	使用平台提供菜品的人
3	集团商户	商户、使用者	可以为某一团体内的人提供菜品的商户
4	普通商户	商户、使用者	可以使用前台点餐系统的商户
5	后台管理员		响应创建折扣和订阅程序

2.1.2 建立事件表

如表 2.2 所示。

表 2.2 顾客支持系统事件表

事件	触发器	来源	动作	响应	目的地
下订单	新订单	用户	生成订单并提交给商家	实时连接、订单确认、订单细节、交易处理	商家
订单管理	查看订单	用户	查看订单状态	实时的订单状态	商家
注册	新用户	使用者	更新用户列表	更新后用户列表	系统
登录	已注册的使用者	使用者	查询用户名密码	验证用户身份	系统
管理账户信息	管理请求	使用者	提交修改的信息	修改账户信息	系统
原料管理	管理请求	商户	更新原料信息	更新后的数据库	系统
管理菜品	管理请求	商户	更新菜品信息	更新后的数据库	系统
管理用户	管理请求	商户	更新用户信息	更新后的数据库	系统
用户取消订单	订单修改请求	客户	修改订单	修改确认、事务处理、订单修改细节	商家
生成收益表	商家请求	商家	生成收益表	收益表	商家

2.2 建立系统用例模型

2.2.1 用例图

本系统的用例模型如图 2.1 所示。

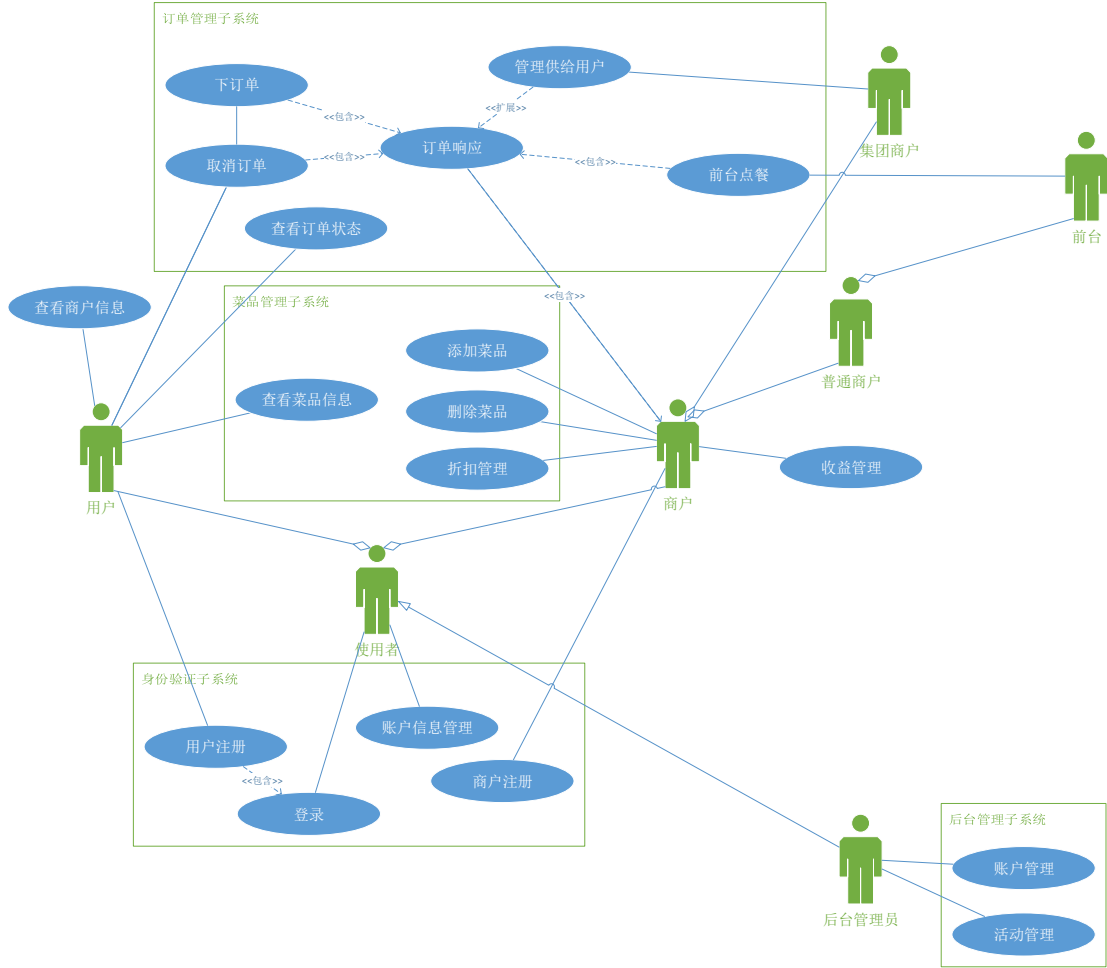


图 2.1 系统用例模型图

2.2.1.1 订餐管理子系统用例描述

订餐管理子系统用例图如图 2.2 所示：

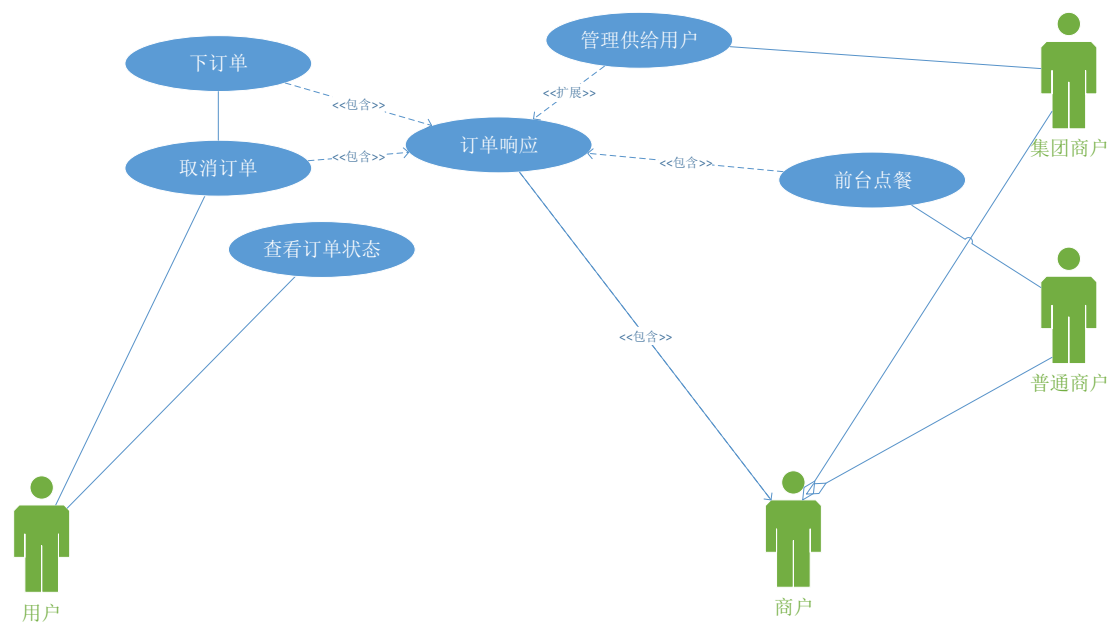


图 2-2 订餐子系统用例模型图

2.2.1.1.1 下订单用例描述

下订单用例描述如表 2.3 所示：

表 2.3 下订单用例描述表

用例名	下订单	优先权	
用例 ID	CY1.1	用例类型	业务用例
主要业务参与者	用户		
描述	用户购买需要的菜品下次		
前置条件	用户已登录		
触发条件	用户将菜品加入购物车		
基本流程	<ol style="list-style-type: none">1) 用户将菜品加入购物车。2) 用户提交订单。3) 生成默认订单信息。4) 用户修改订单信息（配送地址、备注）。5) 用户确定订单。6) 检查配送地址是否为空。7) 检查订单电话是否为空。8) 检查订单电话是否已验证9) 用户支付钱款。10) 系统提交至数据库并向商户推送订单信息，用例结束。		

续表:

表 2.3 下订单用例描述表 (续表)

替代流程	<p>1-a: 用户添加菜品前有不同商户的菜品在购物篮中, 则清空购物篮再添加。</p> <p>2-a: 用户未提交订单, 直接退出, 保留至用户购物车。</p> <p>3-a: 若默认订单信息为空, 则自动转跳指填写订单信息页面。</p> <p>6-a: 若地址为空, 则提示用户添加地址。</p> <p>7-a: 若电话为空, 则提示用户添加订单电话</p> <p>8-a: 若订单电话未验证, 则为用户验证</p> <p>9-a: 若用户为付款, 则进入未完成订单列表, 等待五分钟用户付款</p> <p>9-a-a: 若五分钟后用户未付款, 则自动取消订单</p> <p>10-a: 若商家拒绝接单, 则给用户弹出商家拒绝接单的信息, 并退还钱款, 用例结束。</p>
后置条件	用户订单提交完成
实现约束和说明	用户已登录
待解决问题	无

2.2.1.1.2 取消订单用例描述

取消订单用例描述如表 2.4 所示：

表 2.4 取消订单用例描述表

用例名	取消订单	优先权	
用例 ID	CY1.2	用例类型	业务用例
主要业务参与者	用户		
描述	用户取消已下的订单		
前置条件	用户已登录且已有未送到的订单		
触发条件	用户发出取消订单的指令		
基本流程	1) 用户点击取消订单按钮，用例开始。 2) 弹出取消订单页面，等待用户填写。 3) 用户点击确认取消。 4) 等待商家回应 5) 退换用户钱款 6) 订单取消成功，弹出取消成功页面，用例结束		
替代流程	4-a: 若商家拒绝取消，则弹出商家拒接页面，用例结束		
后置条件	订单取消完成		
实现约束和说明	用户已登录、用户有未送达订单、商家给予反馈信息		
待解决问题	无		

2.2.1.1.3 查看订单状态用例描述

查看订餐用例描述如表 2.5 所示

表 2.5 查看订单用例描述表

用例名	查看订单状态	优先权	
用例 ID	CY 1.3	用例类型	业务用例
主要业务参与者	用户		
描述	用户查看已下单的订单状态。		
前置条件	用户已登录且有已下单的订单。		
触发条件	用户点击订单		
基本流程	1) 用户点击订单 2) 弹出订单状态 3) 等待用户点击确认收货 4) 弹出评价页面 5) 等待用户评价 6) 更新数据库信息，用例结束		
替代流程	4-a: 若用户未评价，则不更新数据库，用例结束		
后置条件	订单完成		
实现约束和说明	用户已登录且有未收货的订单		
待解决问题	无		

2.2.1.1.4 订单响应用例描述

订单响应用例描述如表 2.6 所示：

表 2.6 订单响应用例描述表

用例名	订单响应	优先权	
用例 ID	CY 1.4	用例类型	业务用例
主要业务参与者	商户		
描述	商家响应订单信息。		
前置条件	商家在线且处于营业期。		
触发条件	用户下订单		
基本流程	1) 商家收到订单信息，用例开始。 2) 等待商家接受订单。 3) 将订单响应信息返回给用户。 4) 将订单信息传递给原料管理模块 5) 用例结束。		
替代流程	2-a: 若商家拒接订单，返回拒接信息，用例结束		
后置条件	商家已响应信息		
实现约束和说明	商家已在线		
待解决问题	无		

2.2.1.1.5 前台点餐用例描述

前台点餐用例描述如表 2.7 所示

表 2.7 前台点餐用例描述表

用例名	前台点餐	优先权	
用例 ID	CY 1.5	用例类型	业务用例
主要业务参与者	普通商户		
描述	商户的前台点餐交由后台处理订单		
前置条件	商家已登录		
触发条件	普通商户（前台）开始点餐		
基本流程	1) 普通商户（前台）开始点餐，用例开始。 2) 点餐完成，提交订单，弹出所需的金额。 3) 等待普通商户（前台）确认金额。 4) 将订单发送至订单响应模块。 5) 用例结束。		
替代流程	3-a: 商户点击取消，返回基本流程 1 4-a: 若订单被退回，则重复发送		
后置条件	客户点餐完成		
实现约束和说明	普通商户需要同时登陆两个子账号		
待解决问题	同一个账户内的信息传递的		

2.2.1.1.6 管理供给用户用例描述

管理供给用户用例描述如表 2.8 所示：

表 2.8 管理供给用户用例描述表

用例名	管理供给用户	优先权	
用例 ID	CY 1.7	用例类型	业务用例
主要业务参与者	集团商户		
描述	对于集团商户提供可供销售的用户管理		
前置条件	集团商户已登录		
触发条件	集团商户点击供给用户管理		
基本流程	1) 集团商户点击供给用户管理，用例开始 2) 集团商户点击上传用户列表信息。 3) 解析 Excel 表格信息并更新数据库 4) 提示商户更新完成，用例结束		
替代流程	3-a: Excel 文件不符合格式，则提示用户修改文件格式，返回基本流程 2) 3-b: 更新数据库失败则三次		
后置条件	集团商户已更新供给用户		
实现约束和说明			
待解决问题			

2.2.1.2 菜品管理子系统

菜品管理子系统用例模型图如图 2.3 所示

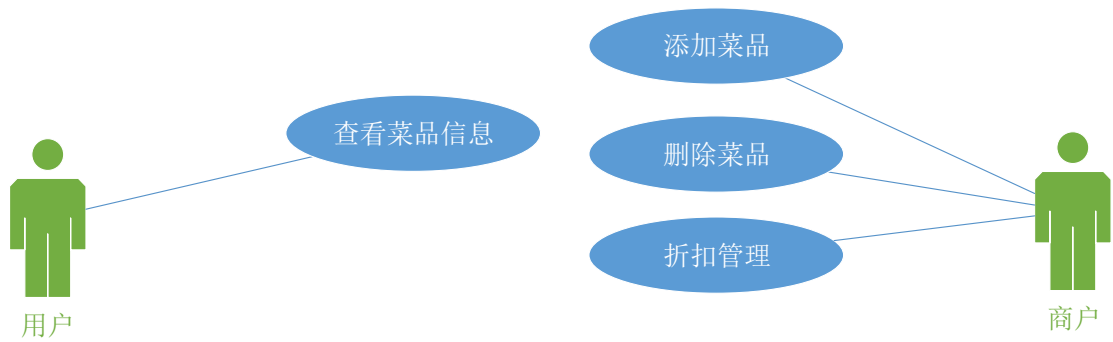


图 2.3 菜品管理用例模型图

2.2.1.2.1 查看菜品信息用例描述

查看菜品信息用例描述如表 2.9 所示：

表 2.9 查看菜品信息用例描述表

用例名	查看菜品信息	优先权	
用例 ID	CY 2.1	用例类型	业务用例
主要业务参与者	用户		
描述	用户查看希望点餐的菜品信息		
前置条件	用户已登录		
触发条件	用户点击某一菜品		
基本流程	1) 用户点击某一菜品，用例开始 2) 读取菜品信息以及差评率 3) 显示菜品信息 4) 用例结束		
替代流程	2-a: 若商品未有人购买，则不显示差评率		
后置条件	用户已查看菜品信息		
实现约束和说明	用户已登录、商户已有菜品		
待解决问题	无		

2.2.1.2.2 添加菜品用例描述

添加菜品用例描述如表 2.10 所示

表 2.10 添加菜品信息用例描述表

用例名	添加菜品	优先权	
用例 ID	CY 2.2	用例类型	业务用例
主要业务参与者	商户		
描述	商户添加新的菜品		
前置条件	商户已登录		
触发条件	商户点击添加菜品		
基本流程	1) 商户点击添加菜品，用例开始 2) 等待商家添加菜品信息 3) 检查是否已添加图片 4) 检查是否已添加菜品名 5) 检查是否已价格信息 6) 更新数据库信息 7) 提示添加成功，用例结束		
替代流程	3-a: 若未添加图片，则提示用户添加图片 4-a: 若未添加图片，则提示用户添加菜品名 5-a: 若未添加价格，则提示用户添加价格 5-b: 若单品价格高于 100，则提示用户价格过高请重新填写 6-a: 若更新数据库失败，则重试		
后置条件	商户已添加菜品		
实现约束和说明	商户已登录		
待解决问题	无		

2.2.1.2.3 删除菜品用例描述

删除菜品用例描述如表 2.11 所示：

表 2.11 删除菜品信息用例描述表

用例名	删除菜品	优先权	
用例 ID	CY 2.3	用例类型	业务用例
主要业务参与者	商户		
描述	商户下架菜品		
前置条件	商户已登录且已上架菜品		
触发条件	商家点击删除按钮		
基本流程	1) 商家点击删除按钮用例开始 2) 弹出确认提示框，等待商家确认 3) 更新数据库信息 4) 弹出删除成功提示框 5) 用例结束		
替代流程	2-a: 商家点击取消，用例结束		
后置条件	商家已下架某菜品		
实现约束和说明	商家已登录且有某种菜品		
待解决问题	无		

2.2.1.2.4 折扣管理

折扣管理用例描述如表 2.12 所示：

表 2.12 折扣管理信息用例描述表

用例名	折扣管理	优先权	
用例 ID	CY 2.4	用例类型	业务用例
主要业务参与者	商户		
描述	商户对店铺内的菜品进行折扣		
前置条件	商户已登录且有菜品		
触发条件	商户点击折扣管理		
基本流程	1) 商户点击折扣管理，用例开始 2) 商家输入折扣后金额或折扣金额以及开始结束时间 3) 检查折扣后金额是否溢出 4) 检查结束时间是否晚于开始时间 5) 检查开始时间是否在于当前时间 6) 更新数据库 7) 用例结束		
替代流程	3-a: 若金额溢出，则提示商家商家金额溢出，返回基本流程 1) 4-a: 若结束时间晚于开始时间，则提示商户时间输入异常，返回基本流程 2) 5-a: 若开始时间晚于当前时间，则提示用户开始时间早于当前时间，返回基本流程 2)		
后置条件	商家已开始某一折扣		
实现约束和说明	商家有菜品		
待解决问题	无		

2.2.1.3 身份验证子系统

身份验证子系统如图 2.4 所示：

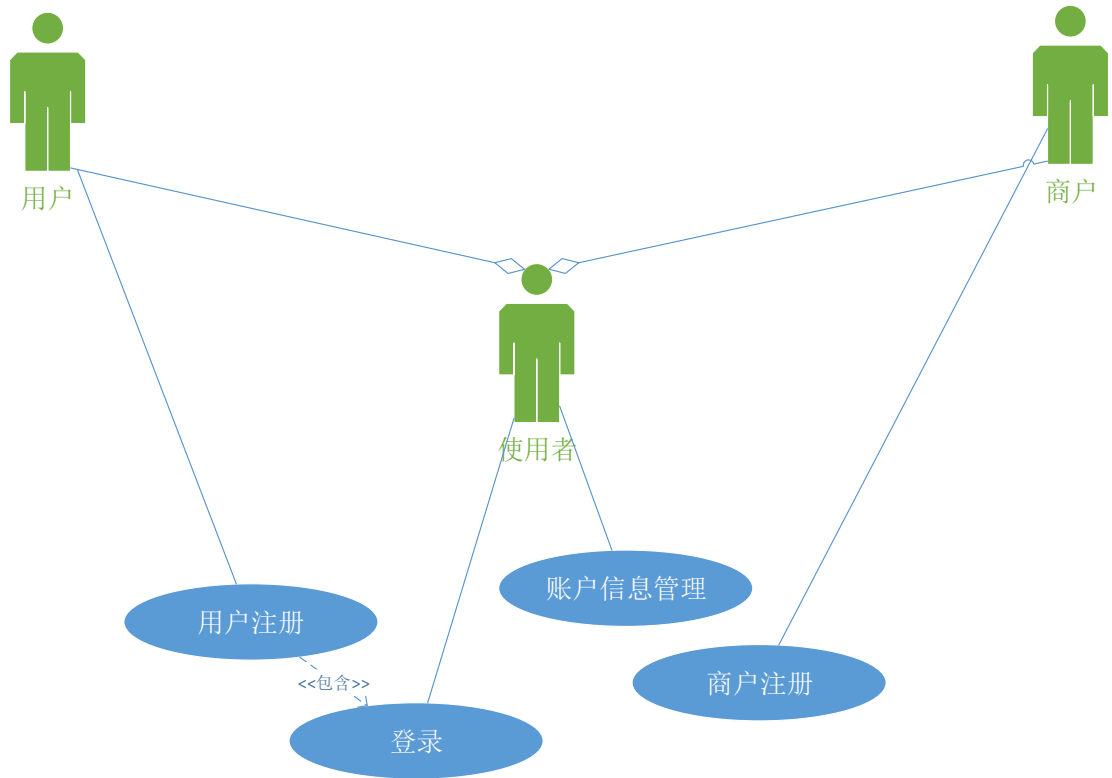


图 2.4 身份验证用例模型图

2.2.1.3.1 用户注册用例描述

用户注册用例描述如表 2.13 所示

表 2.13 用户注册用例描述表

用例名	用户注册	优先权	
用例 ID	CY 3.1	用例类型	业务用例
主要业务参与者	用户		
描述	注册为新用户		
前置条件	用户使用本系统		
触发条件	用户点击注册		
基本流程	<ol style="list-style-type: none">1) 用户点击注册，用例开始2) 弹出注册页面，等待用户填写注册信息3) 等待用户点击提交。4) 检查两次密码输入是否相同5) 检查是否用户名是否符合规范6) 检查是否密码填写是否符合规范7) 检查是否手机号填写是否符合规范8) 向手机发送验证码，等待用户输入9) 验证验证码是否正确10) 自动为用户分配用户 ID11) 更新用户数据库，用例结束		

续表：

表 2.13 用户注册用例描述表（续表）

替代流程	4-a: 若两次不相同, 则提示用户两次密码不一致, 返回基本流程 3) 5-a: 若用户名不符合规范, 则提示用户用户名不符合规范, 返回基本流程 3) 6-a: 若密码不符合规范, 则提示用户密码不符合规范, 返回基本流程 3) 7-a: 若用户名手机号不规范, 则提示用户手机号不符合规范, 返回基本流程 3) 9-a: 若验证码不正确则提示错误, 等待用户再次输入, 返回基本流程 9) 9-b: 若用户为收到验证码, 则返回基本流程 8) 10-a: 若 ID 重复则重新生成
后置条件	用户已注册
实现约束和说明	系统运行正常
待解决问题	无

2.2.1.3.2 登录用例描述

登录用例描述如表 2.14 所示：

表 2.14 登录用例描述表

用例名	登录	优先权	
用例 ID	CY 3.2	用例类型	业务用例
主要业务参与者	使用者		
描述	使用者登录账户		
前置条件	用户使用系统		
触发条件	用户点击登录		
基本流程	1) 用户点击登录，用例开始 2) 弹出登录页面，等待用户输入登录信息 3) 验证用户名是否存在 4) 验证密码是否正确 5) 登录成功，用例结束		
替代流程	3-a: 若用户名不存在，则提示用户用户名不存在，请检查用户名，返回基本流程 2) 4-a: 若密码错误，则提示用户密码错误，请检查密码，返回基本流程 2)		
后置条件	用户已登录		
实现约束和说明	用户已注册		
待解决问题	无		

2.2.1.3.3 账户信息管理

账户信息管理如表 2.14 所示：

表 2.14 账户管理用例描述表

用例名	账户管理	优先权	
用例 ID	CY 3.3	用例类型	业务用例
主要业务参与者	使用者		
描述	使用者修改账户密码		
前置条件	用户已登录		
触发条件	用户点击账户管理		
基本流程	1) 用户点击账户管理，用例开始 2) 弹出用户账户管理页面，等待用户更改 3) 检查密码是否符合规范 4) 检查用户名是否符合规范 5) 检查手机号是否符合规范 6) 更新数据库，用例结束		
替代流程	3-a: 若密码不符合规范，则提示用户密码不符合规范，返回基本流程 3) 4-a: 若用户名不符合规范，则提示用户用户名不符合规范，返回基本流程 3) 5-a: 若用户名手机号不规范，则提示用户手机号不符合规范，返回基本流程 3)		
后置条件	账户信息已更改		
实现约束和说明	用户已注册且登录		
待解决问题	无		

2.2.1.3.4 商户注册子系统

商户注册子系统如表 2.15 所示：

表 2.15 商户注册用例描述表

用例名	商户注册	优先权	
用例 ID	CY 3.4	用例类型	业务用例
主要业务参与者	商户		
描述	商户注册成为平台商户		
前置条件	商户使用该系统		
触发条件	商户发起注册事件		
基本流程	<ol style="list-style-type: none">1) 商户发起注册事件，用例开始2) 弹出注册窗口，等待商户填写3) 检查商铺名称是否符合规范4) 检查商铺工商注册名称是否符合规范5) 检查商铺 Logo 是否符合规范6) 检查商铺简介是否符合规范7) 检查商铺地址是否符合规范8) 查询统一纳税人识别号是否正确9) 查询卫生合格证编号是否正确10) 检查法人身份证信息是否正确11) 更新数据库信息，用例结束		

续表：

表 2.15 商户注册用例描述表（续表）

替代流程	<p>3-a: 若商铺名称不符合规范，则提示商户检查商铺名称</p> <p>4-a: 若商铺工商注册名称不符合规范，则提示商户工商注册名称</p> <p>5-a: 若商铺 Logo 不符合规范，则提示商户检查商铺 Logo</p> <p>6-a: 若商铺简介不符合规范，则提示商户检查商铺简介</p> <p>7-a: 若商铺地址不符合规范，则提示商户检查商铺地址</p> <p>8-a: 若统一纳税人识别号不符合规范，则提示商户检查统一纳税人识别号</p> <p>9-a: 若卫生合格证编号不符合规范，则提示商户检查卫生合格证编号</p> <p>10-a: 若法人身份证信息不符合规范，则提示商户检查商法人身份证信息</p>
后置条件	商户已注册
实现约束和说明	系统运行正常
待解决问题	商户合法性认证

2.2.1.4 后台管理子系统

后台管理子系统用例模型如图 2.5 所示：

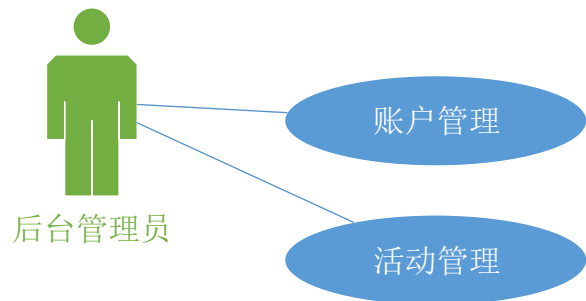


图 2.5 后台管理用例模型图

2.2.1.4.1 账户管理用例描述

账户管理用例描述如表 2.16 所示：

表 2.16 账户管理用例描述表

用例名	账户管理	优先权	
用例 ID	CY 4.1	用例类型	业务用例
主要业务参与者	后台管理员		
描述	后台管理员查看账户信息		
前置条件	管理员已登录		
触发条件	管理员点击账户管理		
基本流程	1) 管理员点击账户管理，用例开始 2) 读取数据库，显示账户信息 3) 用例结束		
替代流程	2-a: 若数据库无账户信息，则提示无用户，用例结束		
后置条件	后台管理员已查看账户信息		
实现约束和说明	后台管理员已登录		
待解决问题	无		

2.2.1.4.2 活动管理用例描述

活动管理用例描述如表 2.17 所示：

表 2.17 活动管理用例描述表

用例名	活动管理	优先权	
用例 ID	CY 4.2	用例类型	业务用例
主要业务参与者	后台管理员		
描述	后台管理员对折扣进行管理		
前置条件	后台管理员已登录		
触发条件	管理员点击活动管理		
基本流程	1) 管理员点击活动管理，用例开始 2) 弹出活动管理页面，等待后台管理员输入 3) 检查金额是否会溢出 4) 检查结束时间是否晚于开始时间 5) 检查开始时间是否在于当前时间 6) 更新数据库，用例结束		
替代流程	3-a: 若金额溢出，则提示金额溢出，返回基本流程 2) 4-a: 若结束时间晚于开始时间，则提示商户时间输入异常，返回基本流程 2) 5-a: 若开始时间晚于当前时间，则提示用户开始时间早于当前时间，返回基本流程 2)		
后置条件	后台管理员已完成活动设置		
实现约束和说明	后台管理员已登录		
待解决问题	无		

2.2.1.5 其他

其他用例模型图如图 2.6 所示：

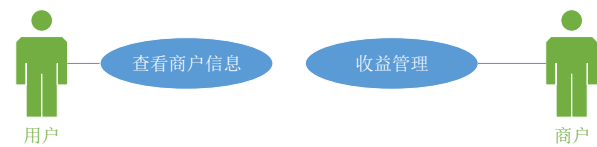


图 2.6 其他用例模型图

2.2.1.5.1 查看商户信息用例描述

查看商户信息用例描述如表 2.18 所示：

表 2.18 查看商户信息用例描述表

用例名	查看商户信息	优先权	
用例 ID	CY 5.1	用例类型	业务用例
主要业务参与者	用户		
描述	用户查看商户的信息		
前置条件	用户已登录		
触发条件	用户点击某一商家的链接		
基本流程	1) 用户点击某一店家的链接，用例开始 2) 读取数据库信息（商家信息及商家的菜品列表） 3) 显示商家信息 4) 用例结束		
替代流程	2-a: 若改商家无菜品，则只读取商家信息，转跳至基本流程 3)		
后置条件	商家信息已展示		
实现约束和说明	用户已登录且平台有商家		
待解决问题	无		

2.2.1.5.2 收益管理用例描述

收益管理用例描述如表 2.19 所示：

表 2.19 收益管理用例描述表

用例名	收益管理	优先权	
用例 ID	CY 5.2	用例类型	业务用例
主要业务参与者	商家		
描述	商家查看收益		
前置条件	商家已登录		
触发条件	商家点击查看收益		
基本流程	1) 商家点击查看收益，用例开始 2) 读取商家的收益、用户量 3) 显示商家的收益及用户量 4) 用例结束		
替代流程	2-a: 若商家没有收益，则提示商家未有销售，用例结束		
后置条件	商家已查看收益		
实现约束和说明	商家已运营超过一天		
待解决问题	无		

2.2.2 下订单用例顺序图描述

下订单用例的系统顺序图如图 2.7 所示：

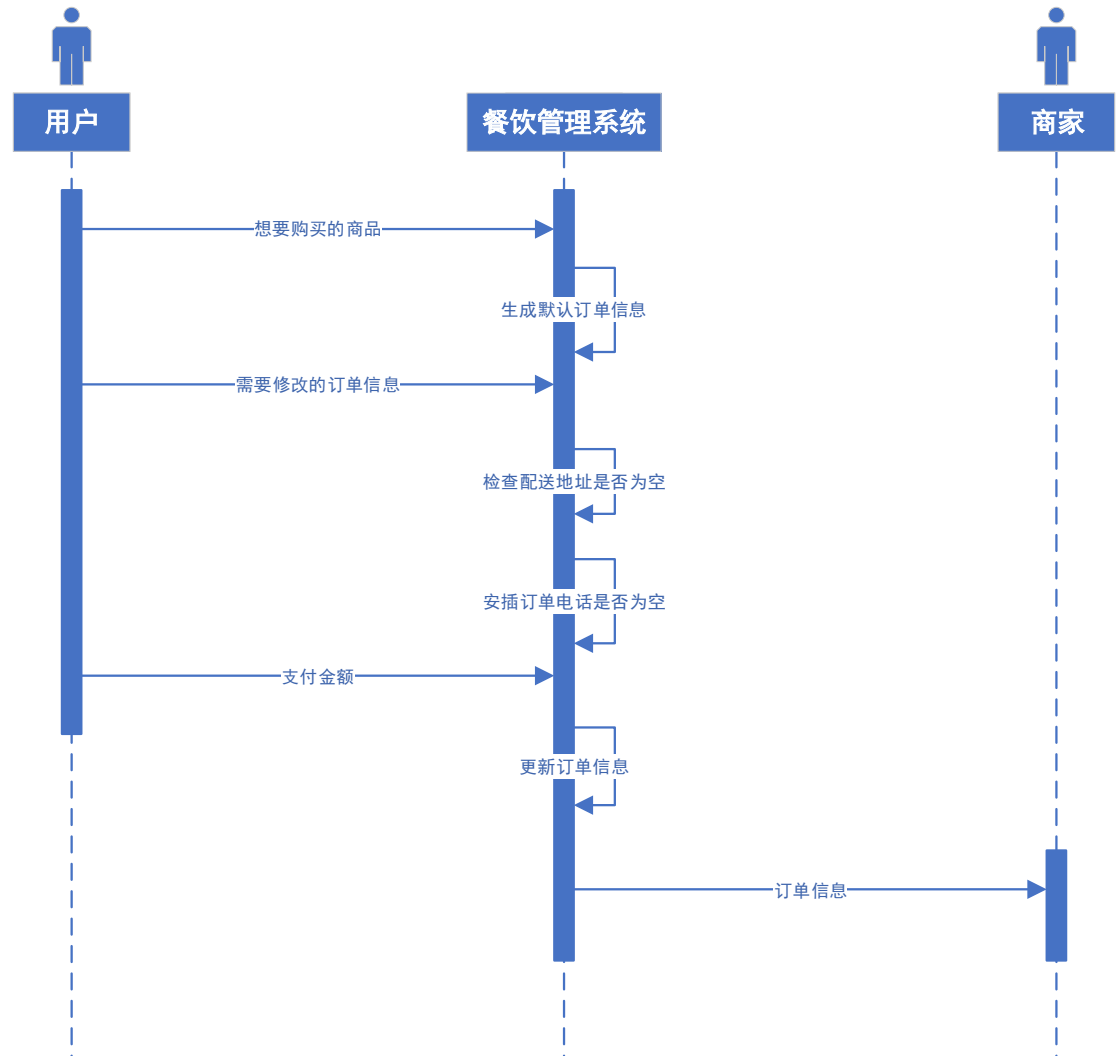


图 2.7 下订单用例顺序图

2.2.3 下订单用例活动描述

下订单用例活动图如图 2.8 所示：

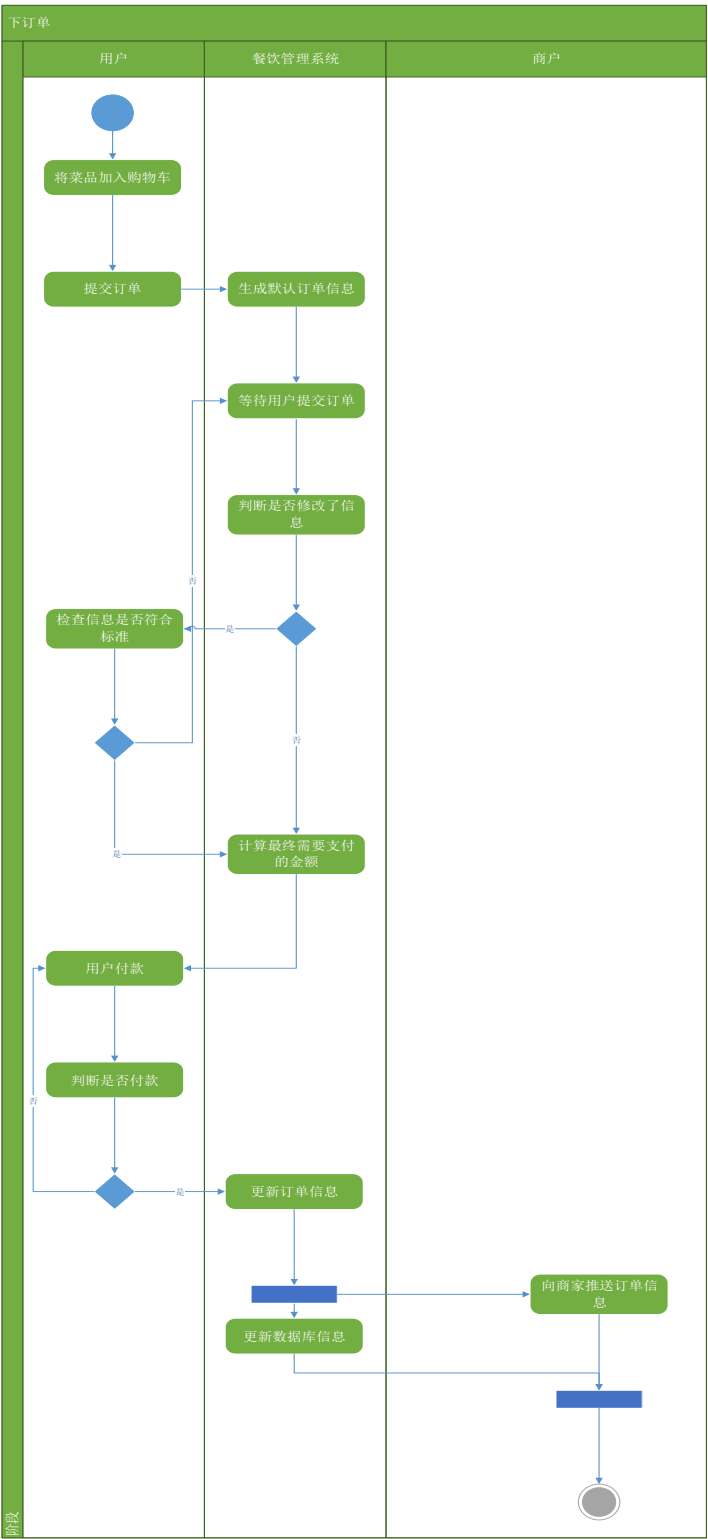


图 2.8 下订单用例活动图

2.3 对象模型建立

餐饮管理系统系统类图如图 2.9 所示。



图 2.9 餐饮管理系统类图

第三章 系统设计

3.1 软件结构图

3.1.1 拓扑结构

系统拓扑图如图 3.1 所示。

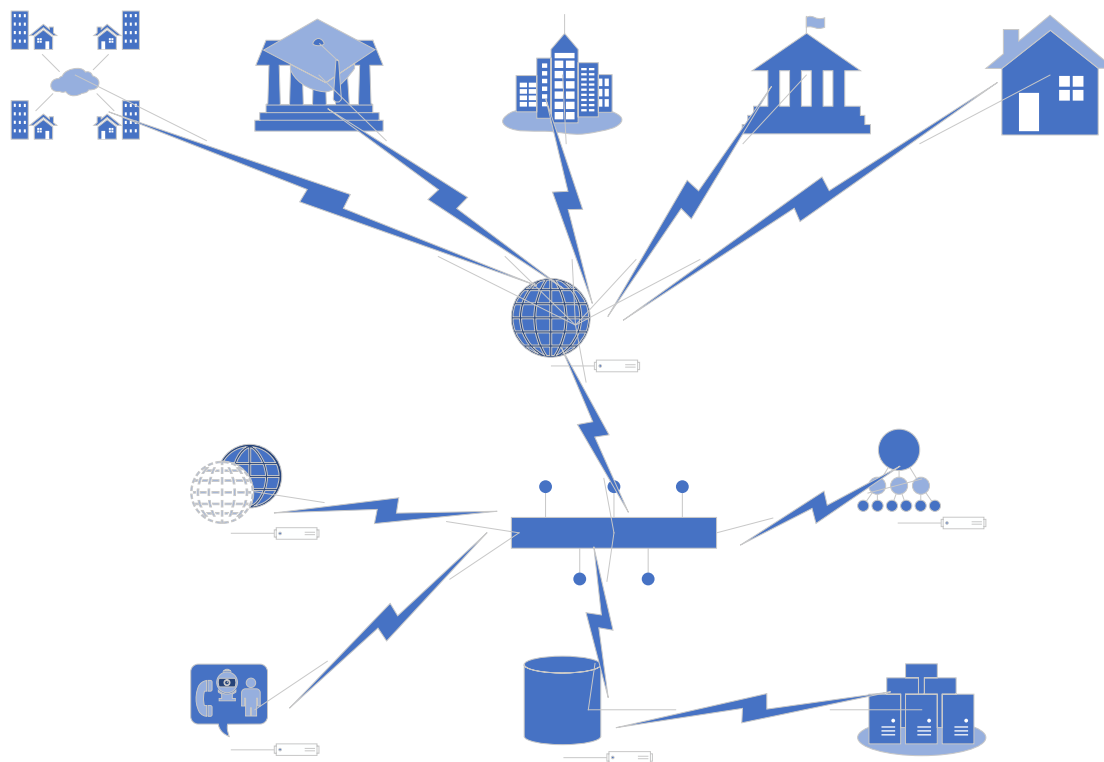


图 3.1 系统拓扑图

3.1.2 软件结构

软件结构图如图 3.2 所示：

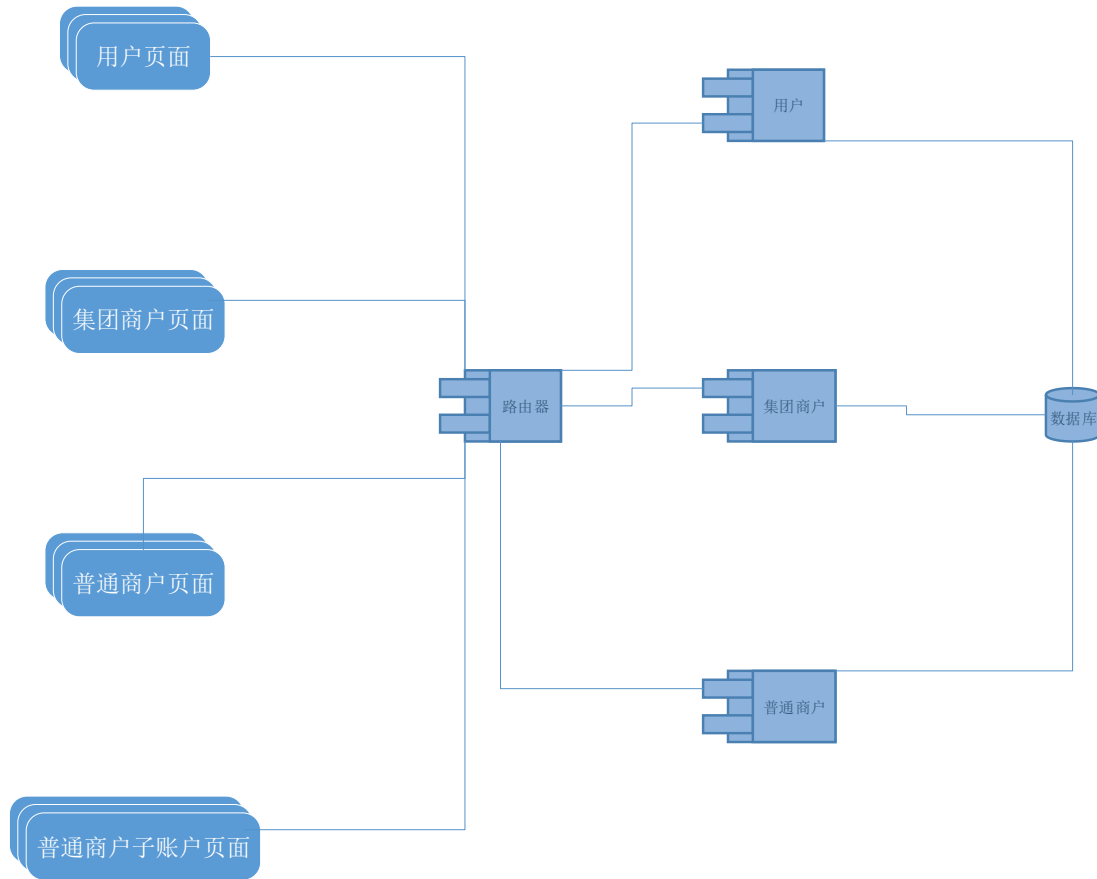


图 3.2 软件结构图

3.2 过程设计

3.2.1 逻辑结构分析

逻辑结构分析如图

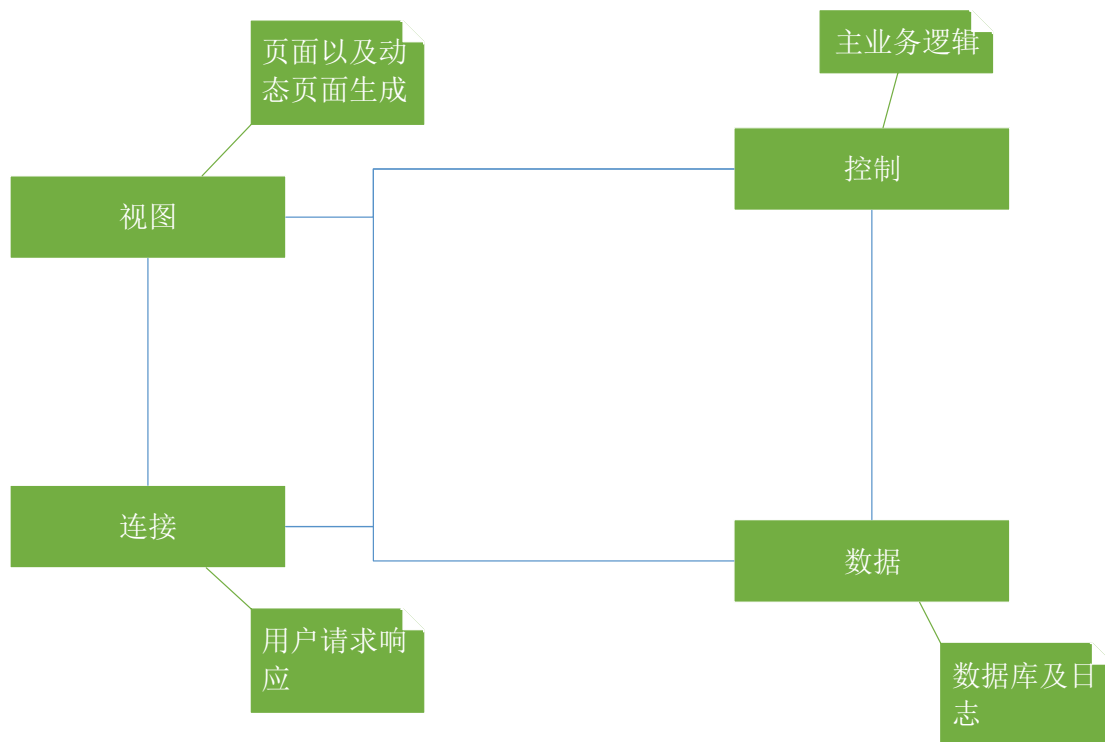


图 3.3 逻辑结构图

3.2.2 过程分析

3.2.2.1 用例分析

用例分析图如图 3.4 所示。

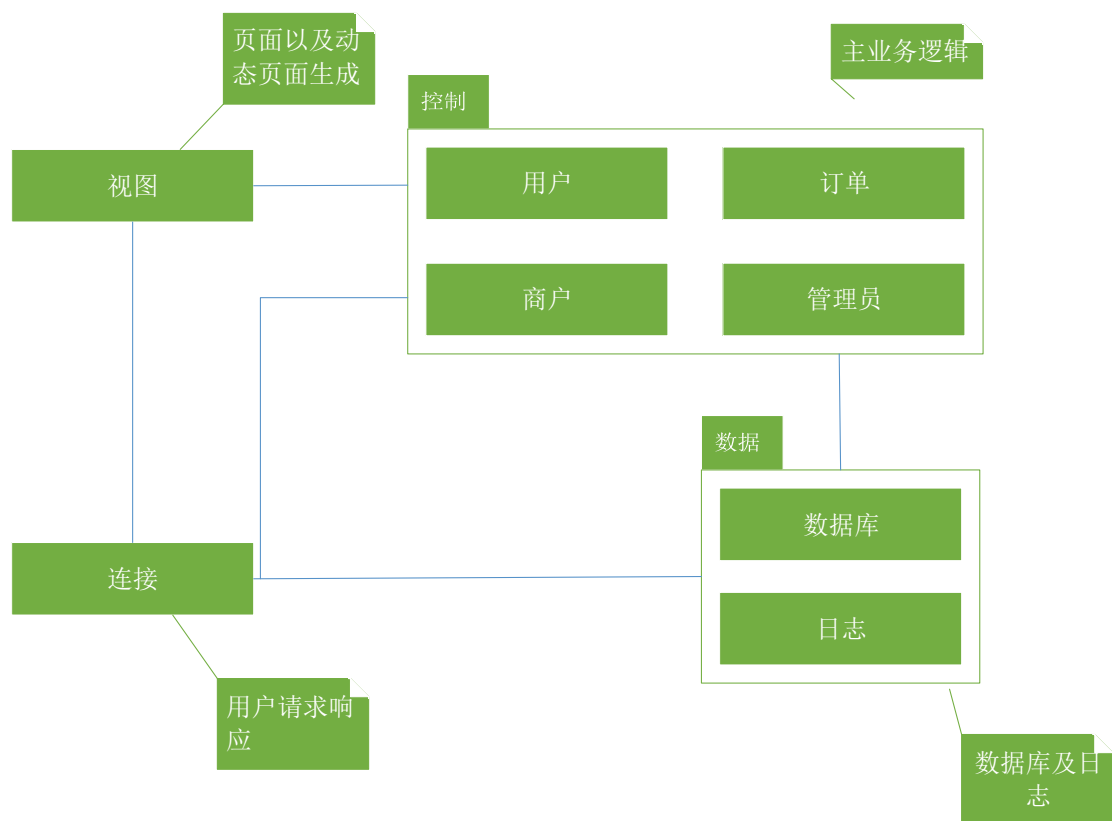


图 3.4 用例分析图

3.2.3 概念类分析

概念类图如图 3.5 所示。

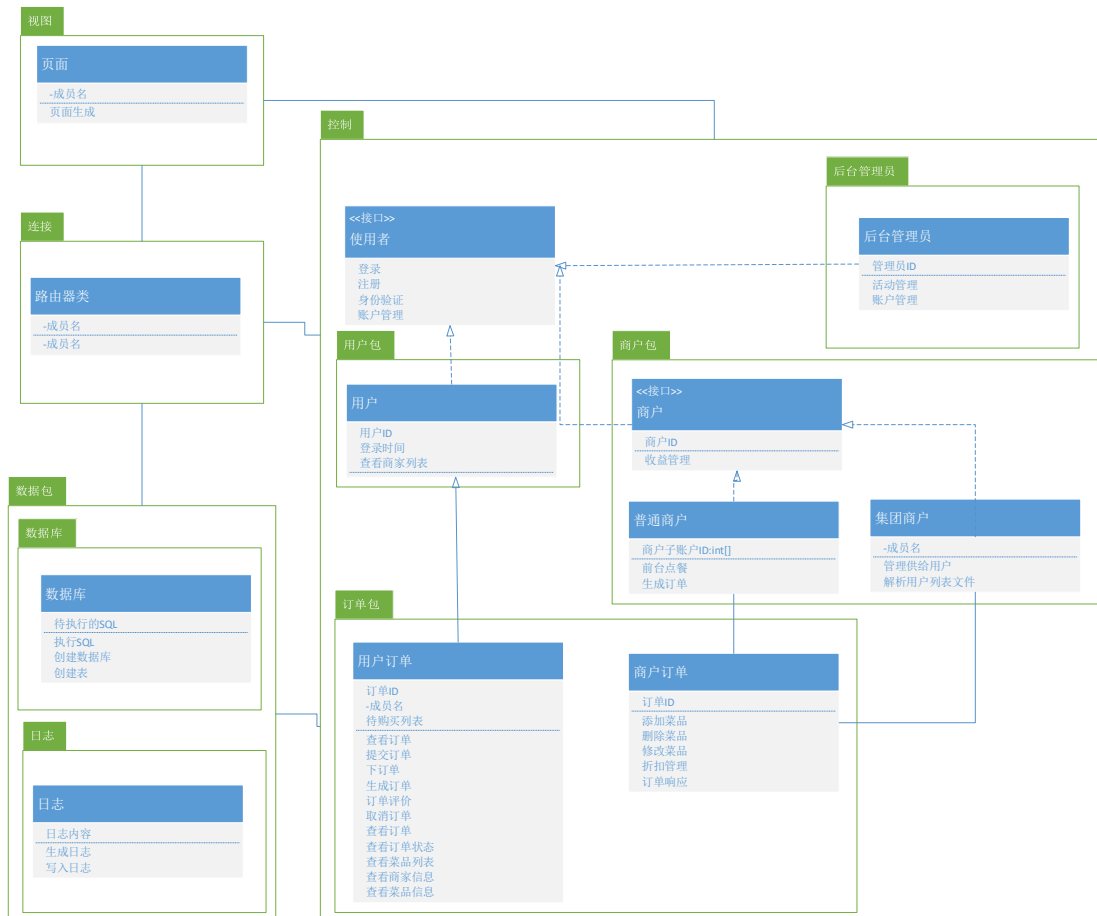


图 3.5 概念类图

3.2.3.1 概念类目录

概念类字典目录如表 3.1 所示

表 3.1 概念类字典目录

概念类名	类型	说明	条目编号
页面	边界类	系统页面	Y 1
路由器	控制类	转发用户请求	K 1
数据库	实体类	对数据库进行增删改查的操作	S 1.1
日志	实体类	对系统行为进行记录	S 1.2
用户	实体类	用户相关的基础操作	S 2.1.1
普通商户	实体类	普通商户相关的特有操作	S 2.2.1
集团商户	实体类	集团商户相关的特有操作	S 2.2.2
用户订单	实体类	用户对订单操作的主体类	S 2.1.2
商户订单	实体类	商户对订单操作的主体类	S 2.2.2
后台管理员	实体类	后台管理员的相关操作类	S 2.3

3.2.3.2 概念类条目

3.2.3.1 页面

表 3.2 页面概念类条目

编号	Y 1
概念类名	页面
职责	系统页面
属性	边界类
说明	系统页面
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 ◆ 创建/删除： ◆ 更新： ◆ 读取：

3.2.3.2 路由器

表 3.3 路由器概念类条目

编号	K 1
概念类名	路由器
职责	处理用户请求
属性	控制类
说明	处理用户的订单、登录、注册、管理等请求
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 ◆ 创建/删除： ◆ 更新： ◆ 读取：

3.2.3.3 数据库

表 3.4 数据库概念类条目

编号	S 1.1
概念类名	数据库
职责	对数据库进行增删改查等操作
属性	实体类
说明	对数据库操作
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 ◆ 创建/删除： ◆ 更新： ◆ 读取：

3.2.3.4 日志

表 3.5 日志概念类条目

编号	S 1.2
概念类名	日志
职责	记录系统行为
属性	实体类
说明	对用户请求及数据库操作进行记录
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 <div style="margin-left: 100px;"> <ul style="list-style-type: none"> ◆ 创建/删除： ◆ 更新： ◆ 读取： </div>

3.2.3.5 用户

表 3.6 用户概念类条目

编号	S 2.1.1
概念类名	用户
职责	用户相关操作
属性	实体类
说明	对用户进行的登录、注册、管理等执行相关操作
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 <div style="margin-left: 100px;"> <ul style="list-style-type: none"> ◆ 创建/删除： ◆ 更新： ◆ 读取： </div>

3.2.3.6 普通商户

表 3.7 普通商户概念类条目

编号	S 2.2.1
概念类名	普通商户
职责	执行对普通商户的相关操作
属性	实体类
说明	对订单、点餐的操作
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 <li style="padding-left: 100px;">◆ 创建/删除： <li style="padding-left: 100px;">◆ 更新： <li style="padding-left: 100px;">◆ 读取：

3.2.3.7 集团商户

表 3.8 集团商户概念类条目

编号	S 2.2.2
概念类名	集团商户
职责	对集团商户的相关操作
属性	实体类
说明	对集团商户的供给用户管理、订单管理
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 <li style="padding-left: 100px;">◆ 创建/删除： <li style="padding-left: 100px;">◆ 更新： <li style="padding-left: 100px;">◆ 读取：

3.2.3.8 用户订单

表 3.9 用户订单概念类条目

编号	S 2.1.2
概念类名	用户订单
职责	对用户订单的相关操作
属性	实体类
说明	对用户订单的添加、取消、查看等操作
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 ◆ 创建/删除： ◆ 更新： ◆ 读取：

3.2.3.9 商户订单

表 3.10 商户订单概念类条目

编号	S 2.2.2
概念类名	商户订单
职责	对商户订单的相关操作
属性	实体类
说明	对商户订单的响应、查看等操作
特殊需求	<ul style="list-style-type: none"> ● 范围： ● 容量： ● 更新品率 ◆ 创建/删除： ◆ 更新： ◆ 读取：

3.2.3.10 后台管理员

表 3.11 后台管理员概念类条目

编号	S 2.3
概念类名	后台管理员
职责	后台管理员的相关操作
属性	实体类
说明	对用户列表以及优惠管理的操作
特殊需求	<ul style="list-style-type: none">● 范围：● 容量：● 更新品率<ul style="list-style-type: none">◆ 创建/删除：◆ 更新：◆ 读取：

3.3 数据库设计

3.3.1 概念结构设计

基于键的数据模型图如图 3.6 所示。

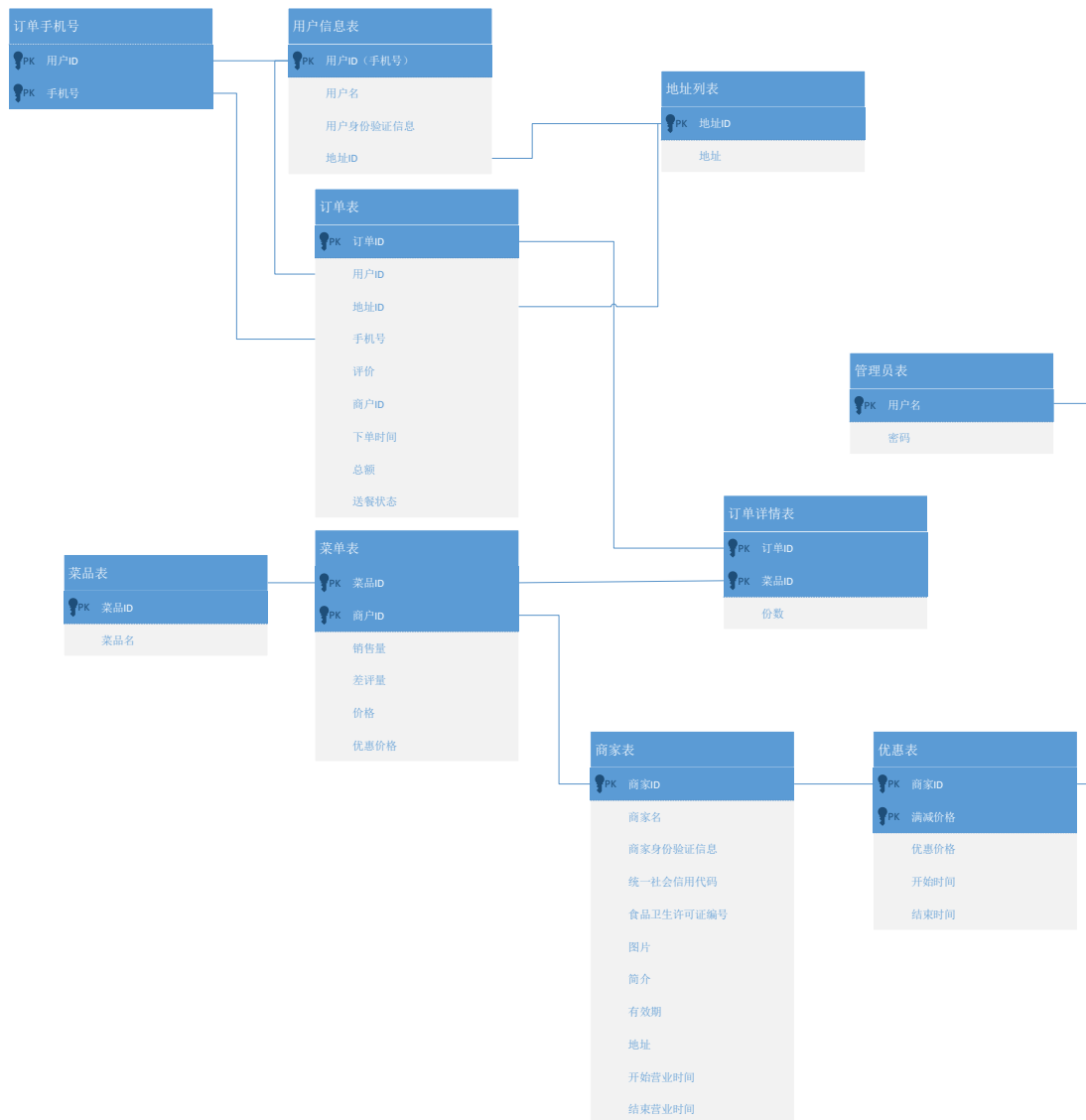


图 3.6 基于键的数据模型

3.3.2 逻辑结构设计

UML 数据库模型如图 3.7 所示。

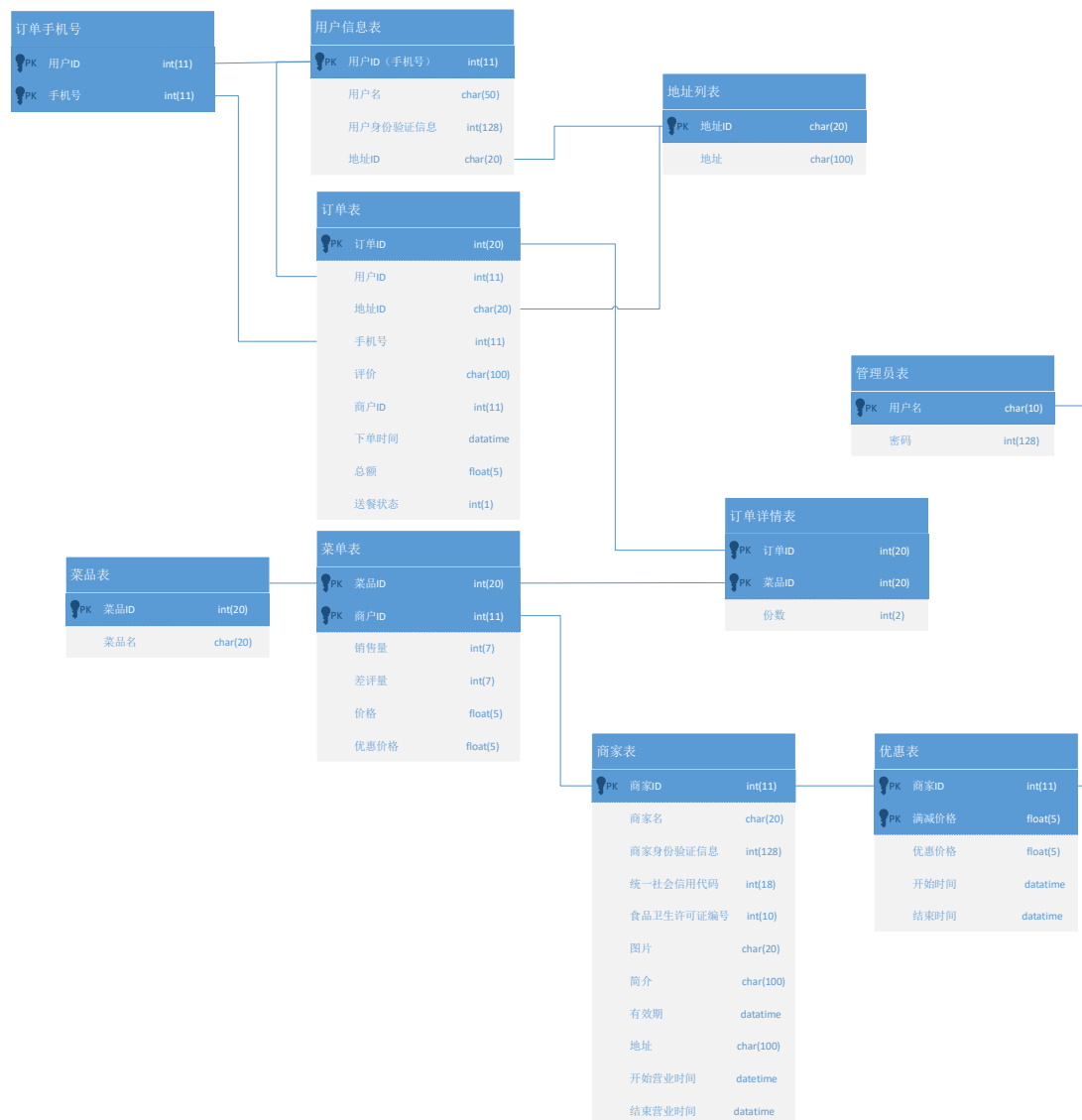


图 3.7 UML 数据库模型

3.3.3 物理结构设计

3.3.3.1 创建数据库

```
create database OrderDinnerSystemDatabase
```

3.3.3.2 创建表:

```
create table addressList (addressId char(20), address char(100), primary  
key(addressId))
```

```
create table manager (userName char(20), password long, primary  
key(userName))
```

```
create table orderInfo (orderId int(12), caiping int, fenshu int, primary  
key(orderID, caiping))
```

```
create table youHui (shangJiaID int, manJianJiaGe float, youHuiJiaGe  
float, kaiShiShiJian datetime, jieShuShiJian datetime, primary key(shangJiaID,  
manJianJiaGe))
```

```
create table shangJiaList (shangJiaID int, shangPingMing char(20),  
password int, tongYiSheHuiXingYongDaiMa int, shiPingWeiShengXuKeBianHao  
int, tuPian char(20), jianJie char(20), youXiaoQi datetime, address char(10),  
kaiShiYeShiJian time, jieShuYingYeShiJian time, primary key(shangJiaID))
```

```
create table orderPhone (userID int, phone char(11), primary key(userID,  
phone))
```

```
create table userInfoList (userID int, idCheck int, addressID char(20),  
primary key(userID))
```

```
create table orderList (orderId int, userID int, addressID char(20), phone  
char(11), shanghuID int, orderTime datetime, zongE float, songcanFlag int,  
primary key(orderID))
```

```
create table caipingList (caipingID int, caipingName char(20), primary
```

```
key(caipingID))
```

```
create table caidanList (caipingID int, shanghuID int, saleNumber int,  
chapingNumber int, jiage float, youhujiage float, primary key(caipingID,  
shanghuID) )
```

```
create table yongliaoList (caipingID int, yuliaoID int, youliang float,  
primary key(caipingID, yuliaoID))
```

```
create table yuanliao (yuanliaoID int, yuanliaoName char(20), primary  
key(yuanliaoID))
```

3.3.3.3 插入数据

```
insert into addressList values('001', 'heida')  
insert into manager values('001', 13712345678)  
insert into orderInfo values('001', 1, 5)  
insert into youHui values('001', 20.0, 5.0, 20180601121301, 20180608)  
insert into shangJiaList values('001','guobaorou', 123456, 100, 001,  
'a.png', 'wu', 20180608, '黑大食堂', 080000, 183000)  
insert into orderPhone values(1271234567, 1234567891)  
insert into userInfoList values(1371234567, 12357895, 'heida')  
insert into orderList values(001, 1371234567, '001', 1371234567, 001,  
20180601122003, 15, 0)  
insert into caipingList values(001, 'guobaorou')  
insert into caidanList values(001, 001, 25, 3, 34, 5.0)  
insert into yongliaoList values(001, 001, 2.0)  
insert into yuanliao values(001, 'rouduan')
```

第四章 系统的实现

4.1 数据库的相关实现

4.1.1 数据库的连接

任何系统的开发都会用到数据库，因此实现 Java 与数据库的链接是系统的实现过程中相当重要的一步。本系统使用 MySQL，采用 jdbc:odebc 方式连接数据库，关键代码如下：

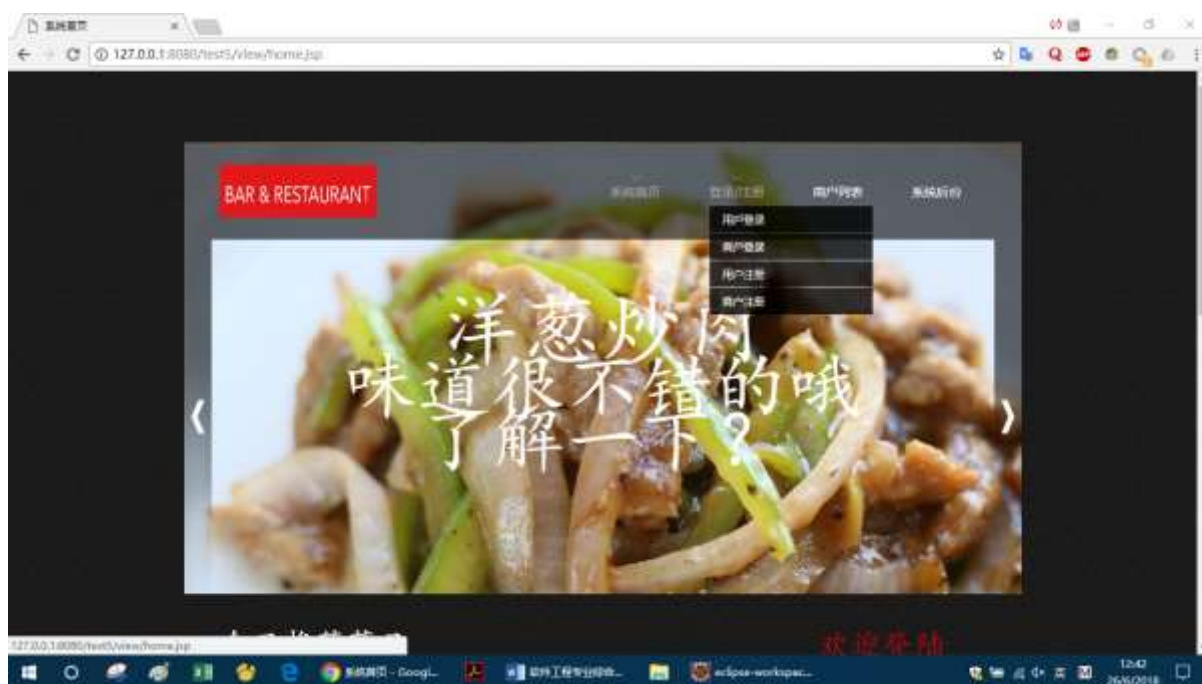
```
package mybeanDB;
import java.util.*;
import java.sql.*;
public class ConnectionPool {
    private Vector<Connection> pool;
    private int poolSize = 10;
    private static ConnectionPool instance = null;
    public static ConnectionPool getInstance() {
        if (instance == null) {
            instance = new ConnectionPool();
        }
        return instance;
    }
    private ConnectionPool() {
        pool = new Vector<Connection>(poolSize);
        Connection conn = null;
        for (int i = 0; i < poolSize; i++) {
            try {
                Class.forName("com.mysql.jdbc.Driver");
                conn =
                DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/cateringmanagement",
                "root", "12345678");
                pool.add(conn);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
    public Connection getConnection() {
        if (pool.size() > 0) {
            Connection conn = pool.get(0);
            pool.remove(conn);
            return conn;
        } else {
            return null;
        }
    }
    public void release(Connection conn) {
        pool.add(conn);
    }
}
```

4.1.2 数据库的操作

对数据库的操作分为静态操作和动态操作。静态操作即查询操作，通过 `executeQuery(String sql)` 语句实现；动态操作通过 `executeUpdate` 实现。

4.2 登录注册模块

4.2.1 系统主页面

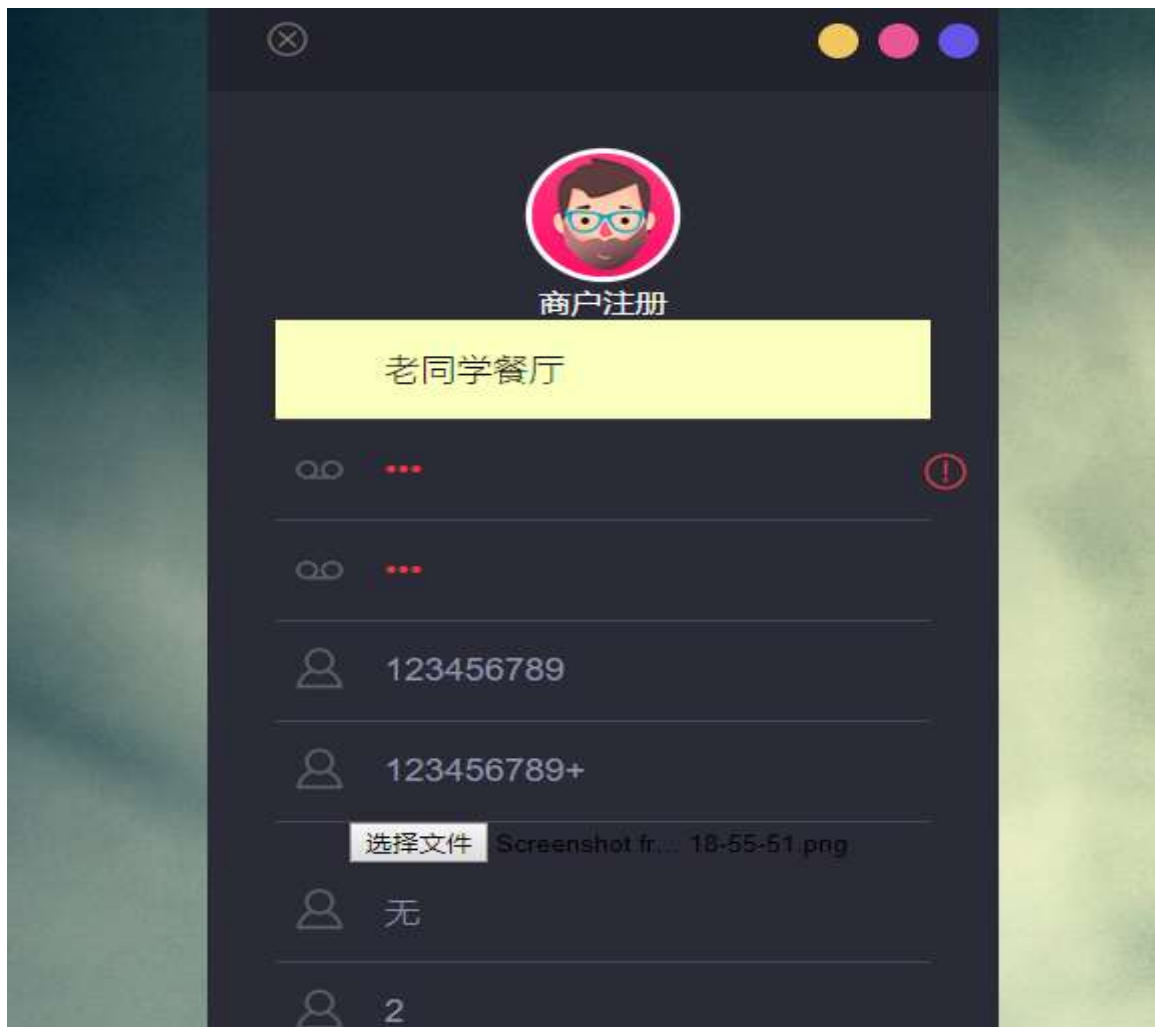


主页面主要实现了用户登录、用户注册、商户登录、商户注册等功能，运用 `css`、`html`、`JavaScript` 构造页面，运用 `jsp` 实现以上 4 个功能

下拉列表：

```
<li class="<%=loginhide%>"><a href="">登录/注册</a>
<ul>
<li><a href="index-custLogin.jsp">用户登录</a></li>
.....
<li><a href="shanghuRegister.jsp">商户注册</a></li>
</ul></li>
```

4.2.2 商户注册页面

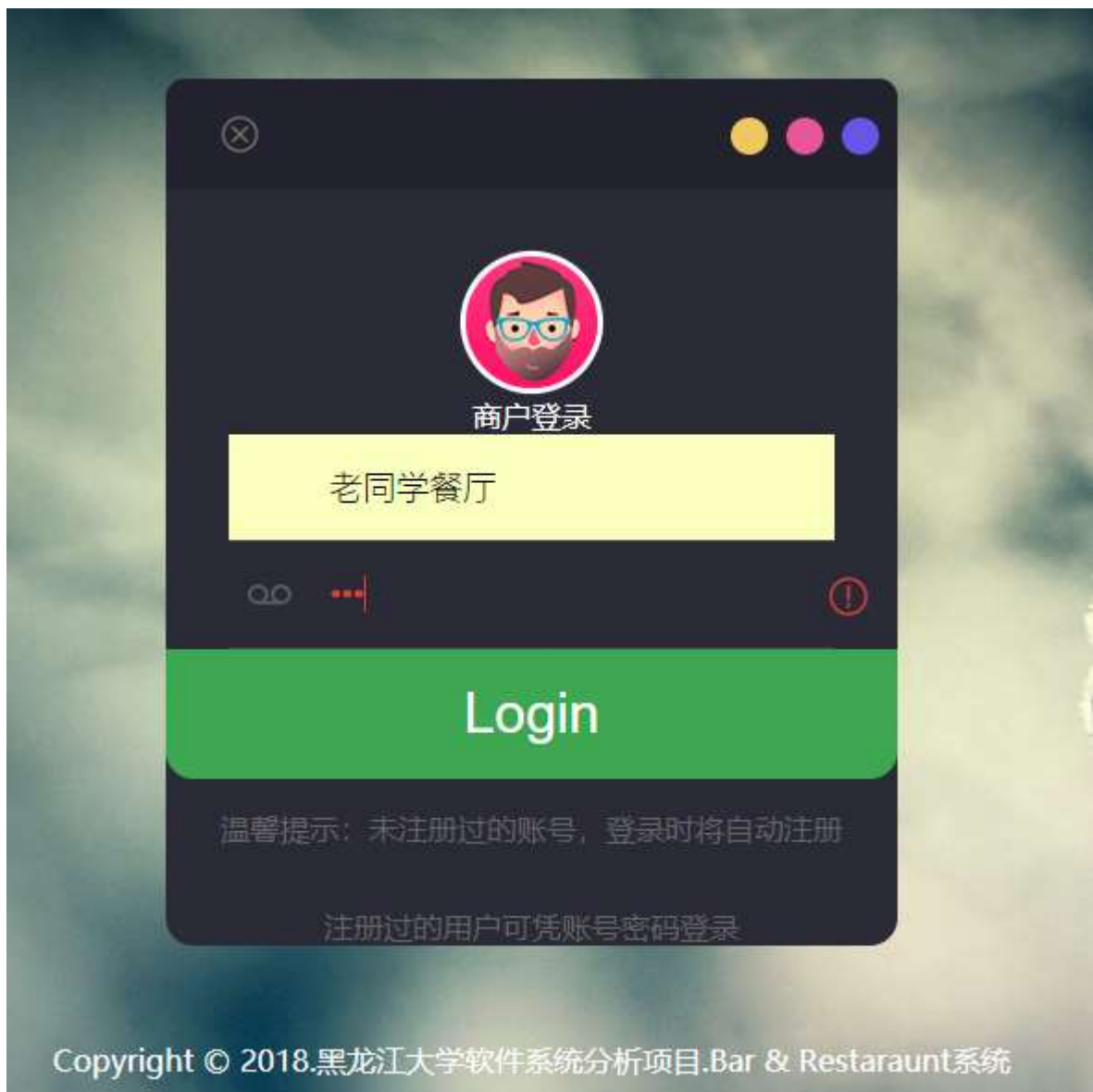


主要代码:

```
//插入数据库
Connection conn = ConnectionPool.getInstance().getConnection();
PreparedStatement restm = conn.
prepareStatement("insert into
shangjialist(shangJiaID,.....,jieShuYingYeShiJian) values(?,.....,?)")
RandomNumber randomID=new RandomNumber();
.....
restm.setString(1,randShangJiaID );
.....
restm.setString(11, endtime);
```

```
//执行  
restm.execute();  
//释放连接  
ConnectionPool.getInstance().release(conn);
```

4.2.3 商户登录页面



主要代码:

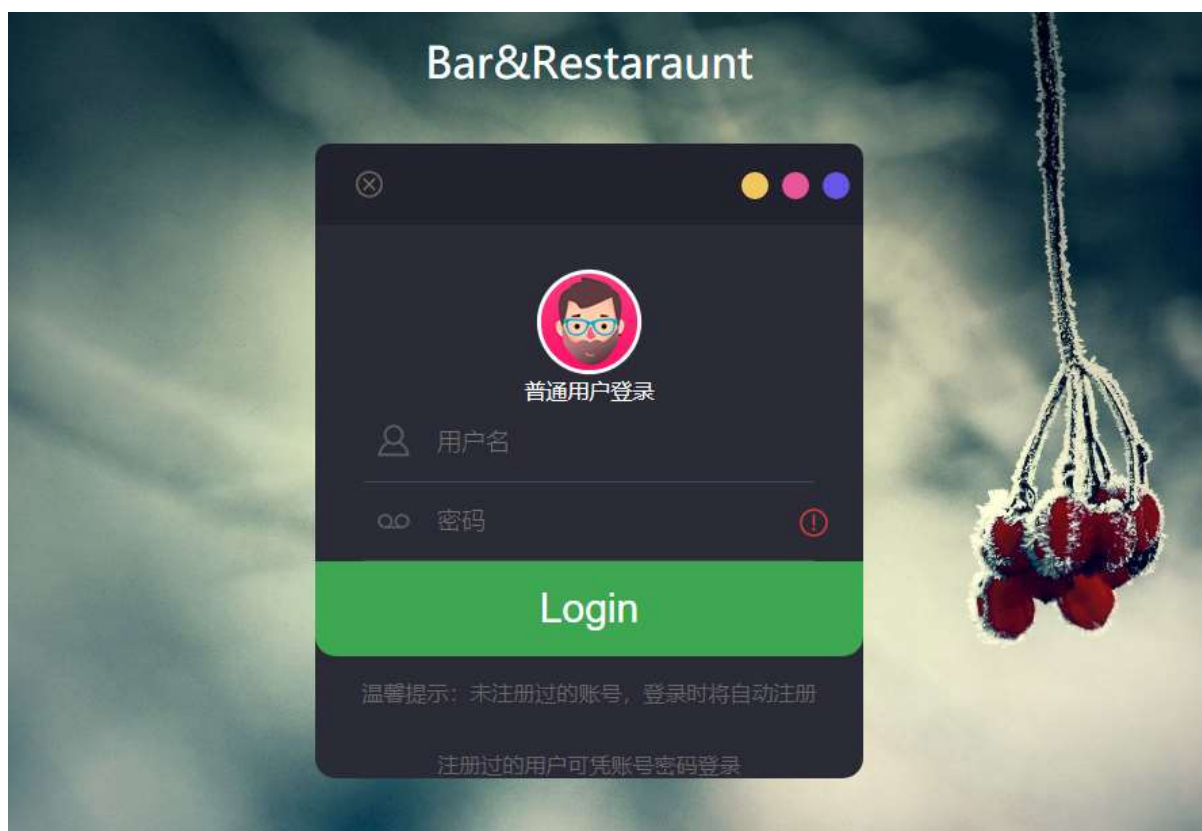
```
while (rs1.next()) {  
    if (rs1.getString("shangjiaName").equals(uname1)) {  
        if (!rs1.getString("idCheck").equals(password1)) {
```

```

        ConnectionPool.getInstance().release(conn);
        request.setAttribute("error", "密码错误!");
        return;
    }
}

```

4.2.4 用户登录页面



该页面实现用户登录功能，用 session 保存用户登录信息。当用户输入用户名和密码时，与数据库比对，若用户名密码正确，则登录到系统主页面，根据用户输入的不同情况在“温馨提示”出给出不同的提示。该功能用 jsp、html、css、MySQL 实现

<p class="warning">温馨提示：未注册过的账号，登录时将自动注册</p>

```

<%   String error = "";
    if(request.getAttribute("error") != null) {
        error =(String)request.getAttribute("error");
        out.print("<font color=' red' >" + error + "</font>");
    }

```


%>

根据输入的不同情况，error 为不同值，用来提示用户。这些情况分别包括“请输入密码”、“请输入用户名”、“密码错误”、“用户不存在，请注册后登录！”，和数据库比对给出不同提示

核心代码：

```
if (uname1.equals("") && password1.equals("")) {
    request.setAttribute("error", "请输入用户名和密码");

    request.getRequestDispatcher("index-custLogin.jsp").forward(request,
response);

    return;
.....
//普通登录
.....

ResultSet rs = stm.executeQuery("select * from userinfolist");
RandomNumber randomID=new RandomNumber();
while (rs.next()) {
    //用户名存在
    if (rs.getString("uname").equals(uname1)) {
        //密码错误
        if (!rs.getString("idCheck").equals(password1)) {
            .....

            request.getRequestDispatcher("index-custLogin.jsp").forward(request,
response);

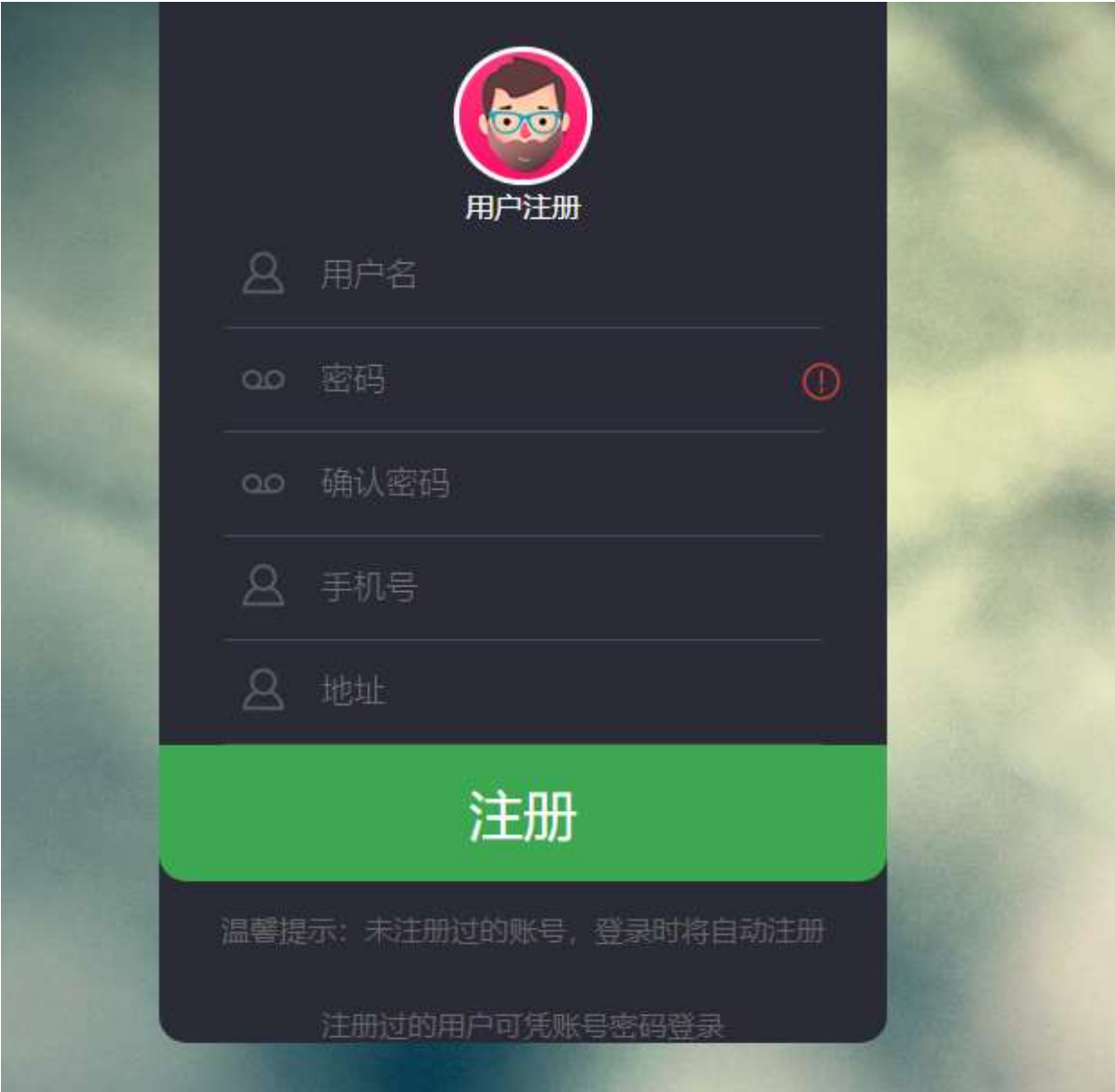
            return;

        } else {//正常登录
            .....
```

```
        ConnectionPool.getInstance().release(conn);
        response.sendRedirect("home.jsp");
        return;
    .....
    request.setAttribute("error", "用户不存在, 请注册后登录!");
    .....
    request.getRequestDispatcher("index-custLogin.jsp").forward(request,
response);
```

4.2.5 用户注册页面

商户和用户分别填写相关信息，根据填写情况给出不同提示，根据用户输入的不同情况在“温馨提示”处给出不同的提示。该功能用 jsp、html、css、MySQL 实现、讲表单提交的处理页面，然后存入数据库。

A user registration form with a dark blue background and a green '注册' (Register) button. The form includes fields for '用户名' (Username), '密码' (Password), '确认密码' (Confirm Password), '手机号' (Mobile Number), and '地址' (Address). A red warning icon is next to the password field. Below the form, there are two lines of text: '温馨提示：未注册过的账号，登录时将自动注册' and '注册过的用户可凭账号密码登录'.

主要代码：

```
request.setCharacterEncoding("UTF-8");  
//普通注册  
String unname = (String) request.getParameter("uname");  
.....
```

```
        if (uname.equals("")) {
            request.setAttribute("error", "请输入用户名");

request.getRequestDispatcher("register.jsp").forward(request, response);

            return;
        }
        .....

        ResultSet rs = stm.executeQuery("select * from userinfolist");
        RandomNumber randomID=new RandomNumber();
        while (rs.next()) {
            if (rs.getString("userID").equals(uname)) {
                .....

                request.setAttribute("error", "用户存在");

request.getRequestDispatcher("register.jsp").forward(request, response);

                return;
            }
        }

        PreparedStatement restm = conn.prepareStatement("insert into
userinfolist(userID,uname,idCheck,addressID) values(?, ?, ?, ?)");
        .....

        //生成随机 ID
        .....

        //添加进用户信息表
        .....

        //订单手机
        restm1.setString(1, randUserID);
        restm1.setString(2, phone);

        //添加进地址表
```

```
.....  
  
//执行  
execute();  
  
.....
```

4.3 商户管理模块

4.3.1 商户查看商户信息

商家信息	
* 商家ID	a75fa91
* 商家名	老同学餐厅
* 简介	老同学餐厅
* 营业时间	20150601 — 20190601
* 地址	学府41街
* 统一社会信用代码	123456789
* 食品卫生许可编号	123456789
* 有效期	2

4.3.2 商户查看菜品列表

商户查看本店菜品信息，并且可以修改、删除每个菜品



主要代码：

```
Connection conn = ConnectionPool.getInstance().getConnection();
//读取菜单列表
Statement stm_caidanlist = conn.createStatement();
ResultSet rs_caidanlist = stm_caidanlist.executeQuery("select * from
caidanlist");
//读取菜品列表
Statement stm_caipinglist = conn.createStatement();
ResultSet rs_caipinglist = stm_caipinglist.executeQuery("select * from
caipinglist");
.....
ConnectionPool.getInstance().release(conn);
```

4.3.3 商户添加菜品

商户可以为该店添加新的菜品（已有的菜品不能再次添加）

添加菜品

* 菜品名 地三鲜

* 价格 13 元

优惠价格 0 元

添加

主要代码：

```
ResultSet caipinSet=stm_tmp.executeQuery("select * from caipinglist");  
//添加菜品  
String caipingID="";  
boolean flag=true;//默认菜单列表中不存在该菜品  
while(caipinSet.next()) {  
    //菜品列表中存在该菜品  
    if( caipingName.equals(caipinSet.getString("caipingName")) ) {  
        caipingID=caipinSet.getString("caipingID");  
        break;  
    }  
}  
if(flag) {//不存在该菜品  
    boolean b_flag=false;  
    .....
```

```

//添加菜单
//判断菜单列表中是否存在该菜品
Statement stm=conn.createStatement();
ResultSet rs=stm.executeQuery("select * from caidanlist");
boolean isCunzai=true;
while(rs.next()) {
    .....
    request.setAttribute("error","添加成功");
}
ConnectionPool.getInstance().release(conn);

```

4.3.4 商户查看新订单

商户可以查看用户刚下或为处理的订单

商户后台管理		新订单						
商家管理		用户名	订餐电话	送货地址	备注	消费	下单时间	操作/状态
菜品管理		朱根松	15545099633	黑大A区2号		31.0	2018-06-26 21:57:19	确认 取消 查看详情
查看菜单列表		国家兴	13720151234	黑大C区1号		45.0	2018-06-26 22:00:26	确认 取消 查看详情
添加菜品		舒永吉	15545099633	黑大A区1号		43.0	2018-06-26 21:58:53	确认 取消 查看详情
订单管理		国家兴	13720151234	黑大C区1号		44.0	2018-06-26 22:00:08	确认 取消 查看详情
历史订单		华浩宇	13720151234	黑大C区2号		32.0	2018-06-26 21:59:30	确认 取消 查看详情
新订单		舒永吉	15545099633	黑大A区1号		31.0	2018-06-26 21:58:35	确认 取消 查看详情
收益管理		华浩宇	13720151234	黑大C区2号		32.0	2018-06-26 21:59:41	确认 取消 查看详情
查看收益		朱根松	15545099633	黑大A区2号		30.0	2018-06-26 21:55:43	确认 取消 查看详情
折扣管理		朱根松	15545099633	黑大A区2号		30.0	2018-06-26 21:56:33	确认 取消 查看详情
本店折扣								

主要代码:

```

ResultSet rs_order = stm_order.executeQuery("select * from orderlist");
while(rs_order.next()) {

```



```
.....  
}
```

```
ConnectionPool.getInstance().release(conn);
```

4.3.5 查看历史订单

用户可以查看历史订单

商户后台管理

商家信息

菜品管理

查看菜单列表

添加菜品

订单管理

历史订单

新订单

收益管理

查看收益

折扣管理

本店折扣

历史订单

用户名	订餐电话	送货地址	备注	消费	下单时间	操作/状态	
朱振松	15545899633	黑大A区2号		31.0	2018-06-26 21:57:19	商户确认订单	查看详情
国家兴	13720151234	黑大C区1号		45.0	2018-06-26 22:00:26	商户确认订单	查看详情
舒永吉	15545899633	黑大A区1号		43.0	2018-06-26 21:58:53	商户确认订单	查看详情
毕浩宇	13720151234	黑大C区2号		32.0	2018-06-26 21:59:30	商户确认订单	查看详情

4.3.6 查看订单详情

商户可查看每个订单的订单详情



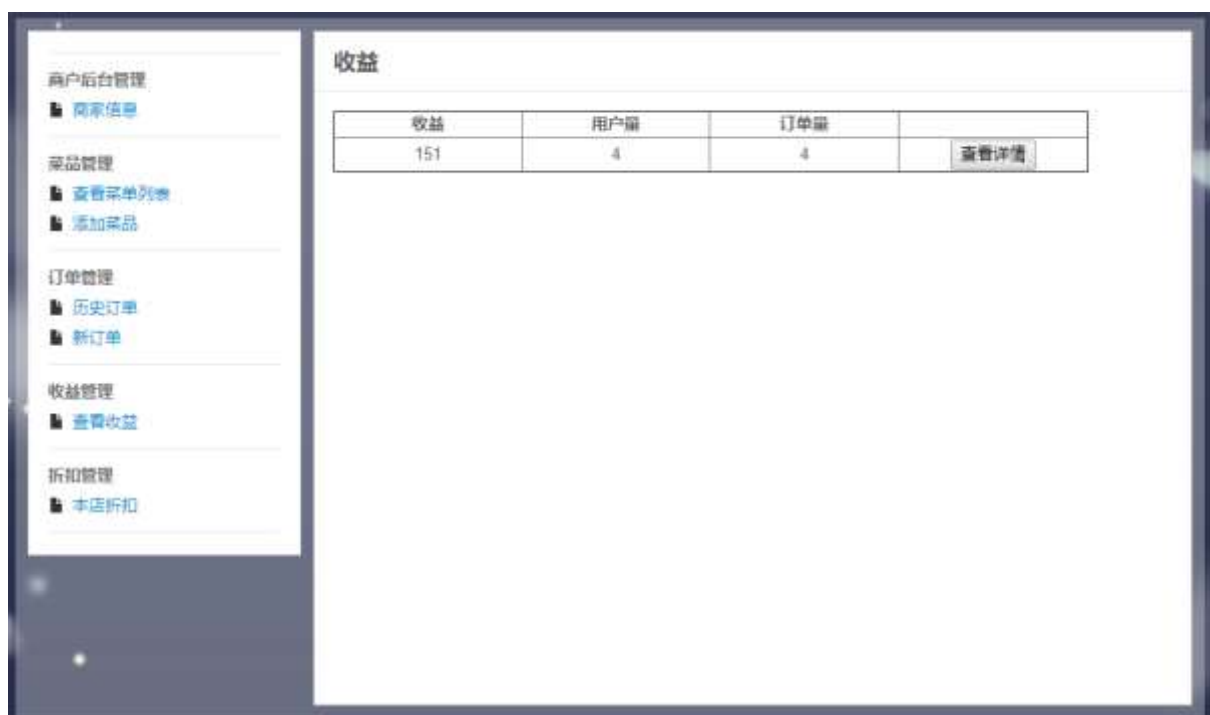
主要代码:

```
//创建连接
//读取订单表
PreparedStatement rstm_order = conn.prepareStatement("select * from
orderlist where orderID=?");
.....
if (rs_order.next()) {
    phone = rs_order.getString("phone");//订餐电话
    .....
    String userID = rs_order.getString("userID");//用户 ID
    PreparedStatement rstm_user = conn.prepareStatement("select * from
userinfolist where userID=?");
    .....
    if(rs_user.next()) {
```

```
        username=rs_user.getString("uname");//下单用户名
    }
    rstm_user.close();
//start_读取地址表
.....
//start_读取订单详情表
.....
//start_读取订单详情表
.....
```

4.3.7 商户查看收益

商户可以查看全部收益，并且可以查看收益详情（用户在该商店的消费）



4.3.8 查看收益详情

用户名	订单量	消费
朱碧松	1	31
国家兴	1	45
舒永吉	1	43
毕浩宇	1	32

主要代码：

```
//读取订单列表

ResultSet rs = stm.executeQuery("select * from orderlist");

//读取数据库
while(rs.next()) {

    String id = rs.getString("shangJiaID");

    boolean flag =

Integer.parseInt(rs.getString("songcanFlag"))==1?true:false;

    //判定商户 ID 是否相同 * 并且成功送出的商品
    if(id.equals(shangJiaID) && flag) {

        sum += rs.getFloat("zongE");//总收益

        dingdanCount++;//总订单

        String userid=rs.getString("userID");

        int index = yonghu_list.indexOf(userid);

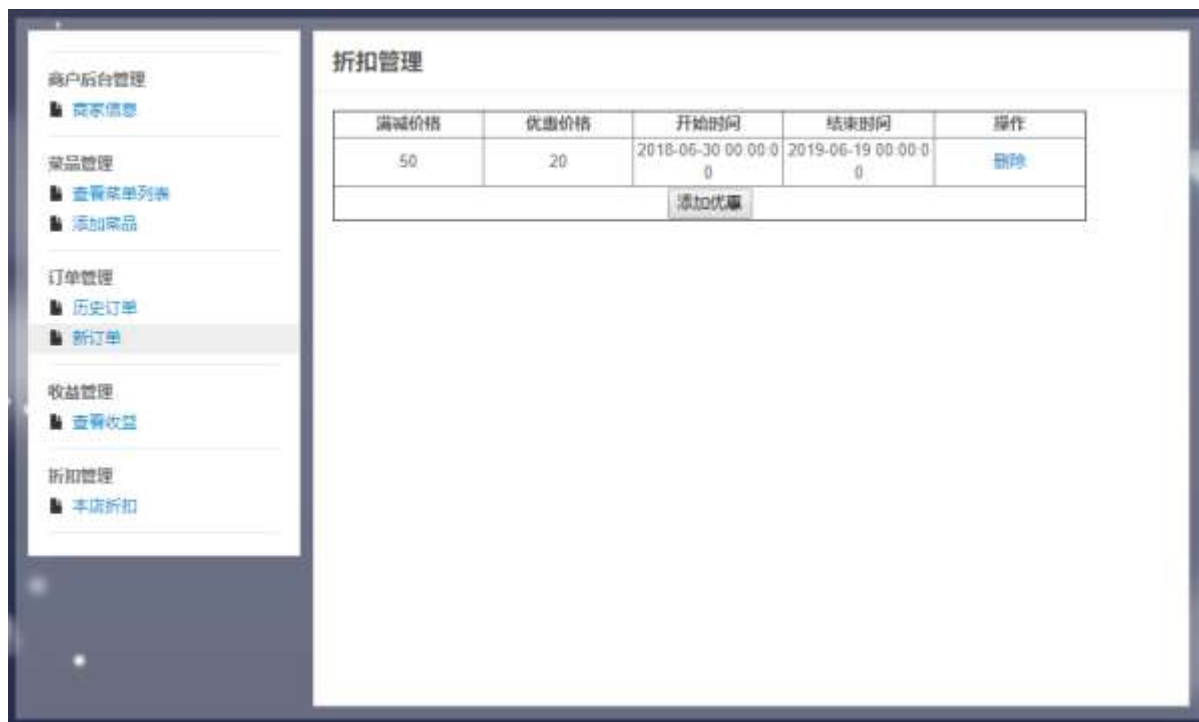
        if(index == -1) {//新用户

            yonghuCount++;//总用户
```

```
        //添加用户
        //添加个人订单
        //添加个人消费
    } else {
        //添加个人订单
        //添加个人消费
    }
}
}
//读取用户名
stm = conn.createStatement();
rs =stm.executeQuery("select * from userinfolist");
while(rs.next()) {
    int index = yonghu_list.indexOf(rs.getString("userID"));
    if(index > -1) {
        yonghu_list.set(index, rs.getString("uname"));
    }
}
stm.close();
ConnectionPool.getInstance().release(conn);
```

4.3.9 商户查看和添加折扣（优惠）

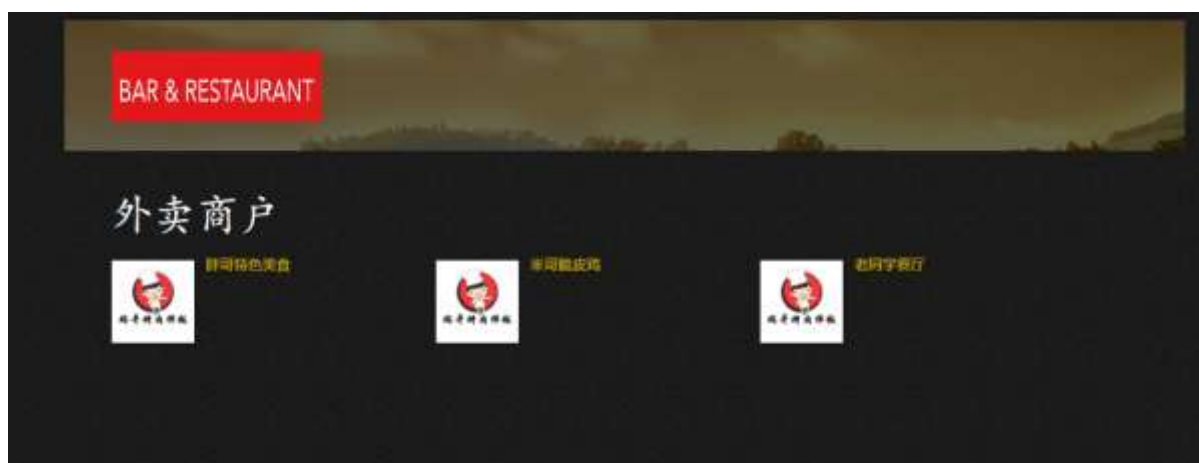
商户可以查看已添加的折扣信息，并且可以删除已有的折扣、添加新的折扣。



4.4 用户模块

4.4.1 用户下订单

4.4.1.1 用户点击商户列表



用户点击某一商户，此处以胖哥特色美食为例

营养套餐

占位图片

青椒肉丝
家常菜
¥14

0 份

占位图片

肉段茄子
家常菜
¥12

0 份

占位图片

鱼香肉丝
家常菜
¥12

0 份

占位图片

锅包肉
家常菜
¥15

0 份

提交订单

用户选购商品，点击提交

送餐地址: 黑大
手机号: 13545899633
编辑

确认订单信息

菜品	单价(元)	数量	小计(元)
 肉段茄子	12	7	84.0
商家留言:		满15.0减10.0: -10.0	
			店铺合计: ¥ 74.0

付款

点击付款:



核心代码:

根据数据库数据动态显示商户:

.....

```
while (shanghulist.next()) {
    shangJiaID = shanghulist.getString("shangJiaID");
    shanghuName = shanghulist.getString("shangjiaName");
    if (i % 3 == 0) {
        %>
        <tr>
        <div class="wrapper indent-bot">
        <%
        }
        %>
        <td>
```

.....

```
href="../caipinlist1/caipinglist.jsp?shangJiaID=<%=shangJiaID %>&shanghuName=<%=shanghuName %>" ><%=shanghuName%></a>
```

.....

if (i + 1 % 3 == 0)//换一行显示

每个商户的菜单列表，动态显示核心代码:

.....

```
shangjiaName = (String)
```



```

request.getParameter("shanghuName");

        shangJiaID = (String) request.getParameter("shangJiaID");

        .....

        ResultSet rscaipinglist =
stmcaipinglist.executeQuery("select * from caipinglist");

        ResultSet rscaidanlist = stmcaidanlist.executeQuery("select *
from caidanlist");

        .....

        while (rscaidanlist.next()) {
if(shangJiaID.equals((String)rscaidanlist.getString("shangJiaID")))
{
        jiage=(String)rscaidanlist.getString("jiage");
        .....
        while(rscaipinglist.next()) {
                .....
                if(caipingID.equals(caipingID1)) {
                        flag=1;
caipingName=(String)rscaipinglist.getString("caipingName");

                                                                %>

                .....

                <h4><%=caipingName%></h4>

                .....

                <b>¥<%=jiage%></b>

                .....

                <input name=<%=inputName %> type="text" value='0'>份</input>
                <i class="price"><%=jiage%></i>
订单页面显示地址核心代码:

ResultSet rsluserinfolist = stmt1.executeQuery("select * from

```

```
userinfolist");
.....
while (rsluserinfolist.next()) {
    if (rsluserinfolist.getString("userId").equals(userId)) {
        while (rs2orderphone.next()) {
            if (rs2orderphone.getString("userId").equals(userId)) {
                phone = rs2orderphone.getString("phone");
                flag1 = 1;
                break;
            }
        }
        while (rs3addresslist.next()) {
            if
(rsluserinfolist.getString("addressId").equals(rs3addresslist.getString("ad
dressId"))) {
                address = rs3addresslist.getString("address");
                .....
            }
        }
    }
}
```

4.4.1.2 显示订单详情，并且计算最佳优惠

思想：判断订单表里菜品数量，若唯一，再判断优惠价格是否为零，若不为零则减去商家对应最佳优惠，若不唯一，则直接由总额减去商家对应最佳优惠。

4.4.1.3 订单处理，将结算成功的订单存入数据库：

```
.....

PreparedStatement stmm = conn.prepareStatement("insert into
orderinfo values( ?,?,? )");

PreparedStatement stmm2 = conn.prepareStatement("insert into
orderlist values( ?,?,?,?,? ,?,? ,?,? )");

stmm2.setString(1, orderID);

.....

stmm2.setString(8, "0");

String liuyan = (String) request.getParameter("liuyan");

stmm2.setString(9, liuyan);

stmm2.execute();

.....
```

4.4.2 用户个人中心

个人中心	
* 用户ID	20156664
* 手机号	15545899633
* 地址	黑大

4.4.2.1 读取数据库该用户信息:

```
.....

if (rsluserinfolist.getString("userId").equals(userId)) {
while (rs2orderphone.next()) {
if (rs2orderphone.getString("userId").equals(userId)) {
phone=rs2orderphone.getString("phone");
flag1=1;
break;
}
}
while (rs3addresslist.next()) {
if
(rsluserinfolist.getString("addressId").equals(rs3addresslist.getString("ad
dressId"))) {
address=rs3addresslist.getString("address");
flag2=1;
break;
}
}
}
.....
```

4.4.3 我的订单

个人中心	我的订单
定单中心	
我的订单	
账户相关	
我的地址	

商户名	订餐电话	送货地址	备注	消费	下单时间	状态	
胖哥特色美食	15545899633	黑大		74.0	2018-06-26 13:56:21	等待送达	查看详情

4.4.3.1 读取我的订单

```

.....

String shangjiaID = rs_order.getString("shangJiaID");
while(rs_shangjia.next()) {
    if(shangjiaID.equals(rs_shangjia.getString("shangJiaID"))) {
        //out.print("<td>" + rs_user.getString("uname") + "</td>"); //用户表中的用户名
        //out.print("<td>" + rs_order.getString("phone") + "</td>"); //订单表中的订餐电话

        shangjianame.add(rs_shangjia.getString("shangjiaName"));
        phone.add(rs_order.getString("phone"));
        break;
    }
}

//获取配送地址

Statement stm_address =
conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);

ResultSet rs_address = stm_address.executeQuery("select * from
addresslist");

String addressId = rs_order.getString("addressId");
while(rs_address.next()) {
    if(addressId.equals(rs_address.getString("addressId"))) {
        %>
        <!-- 订单处理 -->
        <script>
        function queren(bt) { //确认订单
        document.orderform.action="orderChuli.jsp?orderId="+bt.value+"&chuli=1;
        document.orderform.submit();
    }
}

```

```

}

function quxiao(bt) { //取消订单
.....

}

function detail(bt) { //查看详情
.....

}

.....

out.print("<table width='100%' border=1><tr><th>商户名</th><th>订餐电话</th><th>送货地址</th><th>备注</th><th>消费</th><th>下单时间</th><th>状态</th><th></th></tr>");

int size = shangjianame.size();
if(size == 0) {
out.print("<tr align='center'>");
out.print("<td colspan=8>无订单</td>");
out.print("</tr>");
}
else { //状态 0 位新订单, 1 为确认订单, 1 为商家取消订单。-1 为用户退单
for(int i=0; i<size; i++) {
out.print("<tr align='center'>");
out.print("<td>" + shangjianame.get(i) + "</td>");
.....

out.print("<td>" + orderTime.get(i) + "</td>");
if(flag.get(i) == 0) {
out.print("<td><font color='red'>等待送达</font></td>");
} else if(flag.get(i) == -1) {
out.print("<td>已取消订单</td>");
} else {
out.print("<td>已完成订单</td>");

```

```

    }

    out.print("<td><a
href=\"member_order_detail.jsp?orderId="+orderId.get(i)+"\">查看详情
.....

```

4.4.3.2 订单详情



核心代码:

```

<%
String userId=(String)session.getAttribute("userId");
if(userId == null) {
response.sendRedirect("../view/home.jsp");
return;
}
//创建连接
//读取订单表

PreparedStatement rstm_order = conn.prepareStatement("select * from
orderlist where orderId=?");

rstm_order.setString(1, orderId);
ResultSet rs_order = rstm_order.executeQuery();
if (rs_order.next()) {
//订餐电话
//下单时间

```

```
//消费总额
//留言
//start_读取商户表
//下单用户名
}

//start_读取地址表
//地址 ID

PreparedStatement rstm_address = conn.prepareStatement("select * from
addresslist where addressId=?");

rstm_address.setString(1, addressID);

ResultSet rs_address = rstm_address.executeQuery();

if(rs_address.next()) {
//送餐地址
}

rstm_address.close();

//end
}

//start_读取订单详情表
while(rs_orderifo.next()) {
//菜品 ID
//读取菜品表
while(rs.next()) {
if(caiID.equals(rs.getString("caipingID"))) {
//菜品名
//份数
.....

//start_读取订单详情表

PreparedStatement rstm_caidan = conn.prepareStatement("select * from
caidanlist where shangJiaID=?");
```



```
rstm_caidan.setString(1, shangJiaID);
ResultSet rs_caidan = rstm_caidan.executeQuery();
while(rs_caidan.next()) {
String caiID = rs_caidan.getString("caipingID");
int index = caipinID.indexOf(caiID);
if(index != -1) {
Jiage.add(rs_caidan.getInt("jiage"));
}
}
rstm_caidan.close();
<!-- 订单处理 -->
<script>
function queren(bt) { //确认订单
document.orderform.action="orderChuli.jsp?orderId="+bt.value+"&chuli=1"
;document.orderform.submit();
}
function quxiao(bt) { //取消订单
document.orderform.action="orderChuli.jsp?orderId="+bt.value+"&chuli=2"
;
document.orderform.submit();
}
function detail(bt) { //查看详情
document.orderform.action="orderChuli.jsp?orderId="+bt.value+"&chuli=2"
;
document.orderform.submit();
}
.....
<td class="coll"><span class="item-name">商家:</span>
<%=shangjiaName %></td>
```

```

        <td class="col2"><span class="item-name">下单时间:</span><span
class="om_time"
.....
int size=caipinName.size();
for(int i=0; i<size; i++) {
out.print("<tr>");
out.print("<td class=\"col-name\"> <span>"+caipinName.get(i)+"</span>
</td>");
.....
out.print("<td class=\"col-sub-total\">
¥"+Jiage.get(i)*fenshu.get(i)+"</td> ");
out.print("</tr>");
    
```

4.4.4 用户地址

4.4.4.1 显示用户地址核心代码:

```

.....

for="address_address"><span class="required-mark">*</span>详细地址
</label>

<input type="text" name="address" id="address_address"
    
```

```
value=<%=address %> />
<p>（请填写真实详细的送餐地址）</p>
<label for="address_phone"><span class="required-mark">*</span>手机号码
</label>

<input type="text" name="phone" id="address_phone"
value=<%=phone%> />
<p>（请保证外卖员能打通该号码）</p>
<div class="ui-form-footer">
<button type="submit" class="btn btn-yellow">保存</button>
<%
String updateAddress= (String)session.getAttribute("updateAddress");
if(updateAddress==null)updateAddress="";
out.print(updateAddress);
```

4.4.5 用户修改密码



个人中心

我的订单

我的地址

修改密码

修改密码

* 原密码

(请输入现在正在使用的密码)

* 新密码

(请输入新密码)

* 重复新密码

(请再输入一次新密码)

提交更改

根据用户填写情况，系统给出不同提示，输入正确后，存入数据库

4.4.5.1 修改密码核心代码:

```
String old_password = (String) request.getParameter("old_password");
.....

ResultSet rs = stm.executeQuery("select * from userinfolist");
if (old_password.equals("")) {
    session.setAttribute("xiugaizhuangtai", "请输入旧密码!");
    ConnectionPool.getInstance().release(conn);
    request.getRequestDispatcher("member_pwd.jsp").forward(request,
    response);
    return;
}
..... else {
    session.setAttribute("xiugaizhuangtai", "修改成功!");
    .....
```

%>

4.5 本章小结

根据本章，实现了系统的功能，使得系统能正确运行，整个代码逻辑严谨，运行符合预期，满足要求

第五章 系统测试

5.1 测试概要

5.1.1 测试环境

测试环境表如表 5.1 所示：

表 5.1 测试环境配置表

环境	应用服务器	数据库服务器	客户端
硬件配置	Windows CPU: CPU 奔腾III以上 内存: 512M 以上 硬盘: 500G 或以上	MySQL	Windows xp Windows7/8/10 CPU: CPU 奔腾III以上 内存: 512M 以上 硬盘: 100G 或以上
软件配置	Window 2000	Window 2000	Window 2000
网络配置	10M LAN	10M LAN	10M LAN

5.1.2 测试计划

测试计划如表5.2所示：

表 5.2 测试计划表

版本/时间	计划开始时间	实际开始时间	计划结束时间	实际结束时间	加班	增加资源
登录模块	12-3	12-3	12-5	12-5	否	否
注册模块	12-6	12-6	12-8	12-8	否	否
商户模块	12-9	12-9	12-10	12-10	否	否
用户模块	12-11	12-13	12-13	12-13	否	否

5.2 测试执行情况

测试主要由舒永吉、朱银松完成、采用黑盒测试的方法，测试系统并记录结果。由毕浩宇负责整理汇总。

5.3 测试用例

5.3.1 功能性

功能性测试表如表 5.3 所示：

表 5.3 功能性测试表

需求 /功能	测试类型	是否通过	备注
注册	功能测试	Y	符合功能要求
登录	功能测试	Y	符合功能要求
查看商家信息	功能测试	Y	符合功能要求
增加菜品	功能测试	Y	符合功能要求
删除菜品	功能测试	Y	符合功能要求
查看菜品	功能测试	Y	符合功能要求
修改菜品	功能测试	Y	符合功能要求
商户查看订单	功能测试	Y	符合功能要求
查看收益	功能测试	Y	符合功能要求
添加优惠	功能测试	Y	符合功能要求
删除优惠	功能测试	Y	符合功能要求
查看商户列表	功能测试	Y	符合功能要求
查看商户菜品列表	功能测试	Y	符合功能要求
用户下订单	功能测试	Y	符合功能要求
查看用户个人中心	功能测试	Y	符合功能要求
编辑用户地址	功能测试	Y	符合功能要求
用户修改密码	功能测试	Y	符合功能要求
确认订单	功能测试	Y	符合功能要求
取消订单	功能测试	Y	符合功能要求
用户查看退单	功能测试		不通过
用户订单列表	功能测试	Y	不通过

5.3.2 覆盖分析

本次测试对系统需求的覆盖情况为 ☒ 需求覆盖率= Y (P) 项/需求项总数 × 100%=100%

5.4 测试用例

5.4.1 功能测试

5.4.1.1 注册模块（用户）

功能测试模块（用户）用例表如表 3-1 所示：

表 3-1 注册模块（用户）测试用例表

测试编号	A001				
模块名称	注册模块（用户）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	用户填写用户名、密码、确认密码、手机号、地址，点击注册，注册成功直接登录主页				
用例	1、用户名：aaa、密码：123、确认密码：123、手机号：15545979753、地址：黑大 2、用户名：（没有）、密码：123、确认密码：123、手机号：15545979753、地址：黑大 3、用户名：aaa、密码：（没有）、确认密码：123、手机号：15545979753、地址：黑大 4、用户名：aaa、密码：123、确认密码：（没有）、手机号：15545979753、地址：黑大 5、用户名：aaa、密码：123、确认密码：123、手机号：（没有）、地址：黑大 6、用户名：aaa、密码：123、确认密码：123、手机号：15545979753、地址：（没有）				

续表：

表 3-1 注册模块测试用例表

预期情况	1	注册成功，修改数据库，并直接登录主页面
	2	注册失败，不能进入主页面，重新填写
	3	注册失败，不能进入主页面，重新填写
	4	注册失败，不能进入主页面，重新填写
	5	注册失败，不能进入主页面，重新填写
	6	注册失败，不能进入主页面，重新填写
实际结果	1、注册成功，修改数据库，并直接登录主页面 2、注册失败，不能进入主页面，重新填写 3、注册失败，不能进入主页面，重新填写 4、注册失败，不能进入主页面，重新填写 5、注册失败，不能进入主页面，重新填写 6、注册失败，不能进入主页面，重新填写	
结论	1、测试通过 2、测试通过 3、测试通过 4、测试通过 5、测试通过 6、测试通过	

5.4.1.2 注册模块（商户）

功能测试模块（商户）用例表如表 3-2 所示：

表 3-2 注册模块（商户）测试用例表

测试编号	A002				
模块名称	注册模块（商户）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户填写商户名、商户密码、确认密码、统一社会信用代码、 食品卫生许可编号、选择店铺 logo 图片、商户简介、有效期、 地址、开始营业时间、结束营业时间、点击 login 注册并登陆 到商户主页面。				

续表:

表 3-2 注册模块(商户)测试用例表

用例	<p>1、 商户名: 龙凤餐厅、商户密码: 123、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片: 选择图片、商户简介: 好好吃、有效期: 20300225、地址: 学府四道街、开始营业时间:8: 00、结束营业时间 22:00</p> <p>2、 商户名: (没有)、商户密码: 123、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片: 选择图片、商户简介: 好好吃、有效期: 20300225、地址: 学府四道街、开始营业时间:8: 00、结束营业时间 22:00</p> <p>3、 商户名: 龙凤餐厅、商户密码:(没有)、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片: 选择图片、商户简介: 好好吃、有效期: 20300225、地址: 学府四道街、开始营业时间:8: 00、结束营业时间 22:00</p> <p>4、 商户名: 龙凤餐厅、商户密码: 123、确认密码: (没有)、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片: 选择图片、商户简介: 好好吃、有效期: 20300225、地址: 学府四道街、开始营业时间:8: 00、结束营业时间 22:00</p> <p>5、 商户名: 龙凤餐厅、商户密码: 123、确认密码:123、统一社会信用代码: (没有)、食品卫生许可编号:000、选择店铺 logo 图片: 选择图片、商户简介: 好好吃、有效期: 20300225、地址: 学府四道街、开始营业时间:8: 00、结束营业时间 22:00</p> <p>6、 商户名: 龙凤餐厅、商户密码: 123、确认密码:123、统一社会信用代码:000、食品卫生许可编号: (没有)、选择店铺 logo 图片: 选择图片、商户简介: 好好吃、有效期: 20300225、地址: 学府四道街、开始营业时间:8: 00、结束营业时间 22:00</p> <p>7、 商户名: 龙凤餐厅、商户密码: 123、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片: 不选择图片、商户简介: 好好吃、有效期: 20300225、地址: 学府四道街、开始营业时间:8: 00、结束营业时间 22:00</p> <p>8、 商户名: 龙凤餐厅、商户密码: 123、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片: 选择图片、商户简介: (没有)、有效期: 20300225、地址: 学府四道街、开始营业时间:8: 00、结束营业时间 22:00</p>
----	---

续表：

表 3-2 注册模块（商户）测试用例表

用例	<p>9、 商户名：龙凤餐厅、商户密码：123、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片：选择图片、商户简介：好好吃、有效期：（没有）、地址：学府四道街、开始营业时间:8：00、结束营业时间 22:00</p> <p>10、 商户名：龙凤餐厅、商户密码：123、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片：选择图片、商户简介：好好吃、有效期:、地址（没有）、开始营业时间:8：00、结束营业时间 22:00</p> <p>11、 商户名：龙凤餐厅、商户密码：123、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片：选择图片、商户简介：好好吃、有效期：20300225、地址、开始营业时间：（没有）、结束营业时间 22:00</p> <p>12、 商户名：龙凤餐厅、商户密码：123、确认密码:123、统一社会信用代码:000、食品卫生许可编号:000、选择店铺 logo 图片：选择图片、商户简介：好好吃、有效期：20300225、地址、开始营业时间:8：00、结束营业时间：（没有）</p>	
预期情况	1	注册并登陆到商户主页面
	2	注册失败，重新填写
	3	注册失败，重新填写
		注册失败，重新填写
	5	注册失败，重新填写
	6	注册失败，重新填写
	7	注册失败，重新填写
	8	注册失败，重新填写
	9	注册失败，重新填写
	10	注册失败，重新填写
	11	注册失败，重新填写
	12	注册失败，重新填写

续表：

表 3-2 注册模块（商户）测试用例表

实际结果	1、注册并登陆到商户主页面 2、注册失败，重新填写 3、注册失败，重新填写 4、注册失败，重新填写 5、注册失败，重新填写 6、注册失败，重新填写 7、注册失败，重新填写 8、注册失败，重新填写 9、注册失败，重新填写 10、注册失败，重新填写 11、注册失败，重新填写 12、注册失败，重新填写
结论	1、测试通过 2、测试通过 3、测试通过 4、测试通过 5、测试通过 6、测试通过 7、测试通过 8、测试通过 9、测试通过 10、测试通过 11、测试通过 12、测试通过

5.4.1.3 登录模块（用户）

登录模块（用户）用例表如表 3-3 所示：

表 3-3 登录模块（用户）测试用例表

测试编号	A003				
模块名称	登录模块（用户）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	填写用户名和用户密码，校验是否登录成功				
用例	1、用户名：aaa 密码：123 2、用户名：bbb 密码：123 3、用户名：aaa 密码：321 4、用户名：（没有）密码：123 5、用户名：aaa 密码：（没有）				
预期情况	1	登录成功，进入主页面			
	2	用户名错误，不能进入主页面			
	3	密码错误，不能进入主页面			
	4	请填写用户名			
	5	请填写密码			
实际结果	1、 用户登录成功 2、 用户登录失败，重新填写 3、 用户登录失败，重新填写 4、 用户登录失败，重新填写 5、 用户登录失败，重新填写				
结论	1、 测试通过 2、 测试通过 3、 测试通过 4、 测试通过 5、 测试通过				

5.4.1.4 登录模块（商户）

登录模块（商户）用例表如表 3-4 所示：

表 3-4 登录模块（商户）测试用例

测试编号	A004				
模块名称	登录模块（商户）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [✓] 正式发布				
定义	填写商户名和、商户密码，校验是否登录成功				
用例	1、商户名：龙凤餐厅 密码：123 2、商户名：龙龙餐厅 密码：123 3、商户名：龙凤餐厅 密码：321 4、商户名：（没有） 密码：123 5、商户名：龙凤餐厅 密码：（没有）				
预期情况	1	登录成功，进入主页面			
	2	用户名错误，不能进入主页面			
	3	密码错误，不能进入主页面			
	4	请填写商户名			
	5	请填写密码			
实际结果	1、 登录成功，进入主页面 2、 用户名错误，不能进入主页面 3、 密码错误，不能进入主页面 4、 请填写商户名 5、 请填写密码				
结论	1、 测试通过 2、 测试通过 3、 测试通过 4、 测试通过 5、 测试通过				

5.4.1.5 商户模块（查看商家信息）

商户模块（查看商家信息）用例表如表3-5所示：

表 3-5 商户模块（查看商家信息）测试用例表

测试编号	A005				
模块名称	商户模块（查看商家信息）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户点击商户信息、显示商户具体信息				
用例	1 用户点击商户信息 数据库未被破坏 2 用户点击商户信息 数据库被破坏				
预期情况	1	显示商户信息			
	2	程序崩溃			
实际结果	1、 显示商户信息 2、 程序崩溃				
结论	1、 测试通过 2、 测试通过				

5.4.1.6 商户模块（添加菜品）

商户模块（添加菜品）用例表如表3-6所示：

表 3-6 商户模块（添加菜品）测试用例表

测试编号	A006				
模块名称	商户模块（添加菜品）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户填写菜品名、菜品价格、优惠价格				
用例	1 菜品名：麻辣烫、菜品价格：45、优惠价格：20 2 菜品名：（没有）、菜品价格：45、优惠价格：20 3 菜品名：麻辣烫、菜品价格：（没有）、优惠价格：20				
预期情况	1	添加成功			
	2	请输入菜品名			
	3	请输入价格			
实际结果	1、添加成功 2、请输入菜品名 3、请输入价格				
结论	1、测试通过 2、测试通过 3、测试通过				

5.4.1.7 商户模块（删除菜品）

商户模块（删除菜品）用例表如表3-7所示：

表 3-7 商户模块（删除菜品）测试用例表

测试编号	A007				
模块名称	商户模块（删除菜品）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户点击删除，数据未被破坏				
用例	1、点击删除，数据未被破坏，删除成功 2、点击删除，数据未被破坏，删除失败				
预期情况	1	删除成功			
	2	删除失败			
实际结果	1、删除成功 2、删除失败				
结论	1、测试通过 2、测试通过				

5.4.1.8 商户模块（查看菜品）

商户模块（查看菜品）用例表如表3-8所示：

表 3-8 商户模块（查看菜品）测试用例表

测试编号	A008				
模块名称	商户模块（查看菜品）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户点击查看菜品列表，数据未被破坏				
用例	1、点击查看菜品列表，数据未被破坏，显示成功 2、点击查看菜品列表，数据未被破坏，显示失败				
预期情况	1	显示成功			
	2	显示成败			
实际结果	1、显示成功 2、显示失败				
结论	1、测试通过 2、测试通过				

5.4.1.9 商户模块（修改菜品）

商户模块（修改菜品）用例表如表3-9所示：

表 3-9 商户模块（修改菜品）测试用例表

测试编号	A009				
模块名称	商户模块（修改菜品）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户点击修改菜品，将重新填写菜品名或菜品价格或优惠价格				
用例	1、将菜品改为肉段茄子，其余不变 2、将价格改为 20，其余不变 3、将优惠价格改为 2，其余不变 4、将菜品信息全置为空				
预期情况	1	修改成功			
	2	修改成功			
	3	修改成功			
	4	显示“请输入菜品名”			
实际结果	1、修改成功 2、修改成功 3、修改成功 4、显示“请输入菜品名”				
结论	1、测试通过 2、测试通过 3、测试通过 4、测试通过				

5.4.1.10 商户模块（商户查新订单或历史订单）

商户查新订单或历史订单用例表如表3-10所示：

表 3-10 商户查新订单或历史订单测试用例表

测试编号	A010				
模块名称	商户查新订单或历史订单				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户点击查看订单或历史订单并显示				
用例	1、点击查看新订单或历史订单，数据未被破坏，显示成功 2、点击查看新订单或历史订单，数据被破坏， 显示失败				
预期情况	1	查看成功			
	2	显示出错页面			
实际结果	1、查看成功 2、显示出错页面				
结论	1、测试通过 2、测试通过				

5.4.1.11 商户模块（查看收益）

商户查看收益用例表如表3-11所示：

表 3-11 商户查看收益测试用例表

测试编号	A011				
模块名称	商户查看收益				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户点击查看收益				
用例	1、点击查看收益，数据未被破坏，显示成功 2、点击查看收益，数据被破坏，显示失败				
预期情况	1	查看成功			
	2	查看成功			
实际结果	1、查看成功 2、查看成功				
结论	1、测试通过 2、测试通过				

5.4.1.12 商户模块（添加优惠）

商户添加优惠模块用例表如表3-12所示：

表 3-1 2 商户添加优惠模块测试用例表

测试编号	A012				
模块名称	商户添加优惠				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户点击填写 满减价格、优惠价格、开始时间、结束时间				
用例	1、满减价格：50、优惠价格:20、开始时间：大于当前时间、结束时间：大于开始时间 2、满减价格：（没有）、优惠价格:20、开始时间：大于当前时间、结束时间：大于开始时间 3、满减价格：50、优惠价格：（没有）、开始时间：大于当前时间、结束时间：大于开始时间 4、满减价格：50、优惠价格：（没有）、开始时间：小于当前时间、结束时间：小于开始时间				
预期情况	1	添加优惠成功			
	2	添加优惠失败			
	3	添加优惠失败			
	4	添加优惠失败			
实际结果	1、添加优惠成功 2、添加优惠失败 3、添加优惠失败 4、添加优惠失败				
结论	1、测试通过 2、测试通过 3、测试通过 4、测试通过				

5.4.1.13 商户模块（删除优惠）

商户删除优惠模块用例表如表3-13所示：

表 3-13 商户删除优惠模块测试用例表

测试编号	A013				
模块名称	商户删除优惠				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户点击删除优惠				
用例	1、点击删除优惠，数据未被破坏，删除成功 2、点击删除优惠，数据被破坏，删除失败				
预期情况	1	删除成功			
	2	删除失败			
实际结果	1、删除成功 2、删除失败				
结论	1、测试通过 2、测试通过				

5.4.1.14 用户模块（查看商户列表）

查看商户列表用例表如表3-14所示：

表 3-14 查看商户列表测试用例表

测试编号	A014				
模块名称	查看商户列表				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	用户点击查看商户列表				
用例	1、点击商户列表，数据未被破坏，显示成功 2、点击商户列表，数据被破坏，显示失败				
预期情况	1	显示成功			
	2	显示失败			
实际结果	1、显示成功 2、显示失败				
结论	1、测试通过 2、测试通过				

5.4.1.15 用户模块（查看商户菜品列表）

查看商户菜品列表用例表如表3-15所示：

表 3-15 查看商户菜品列表测试用例表

测试编号	A015				
模块名称	查看商户菜品列表				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	用户点击商户名				
用例	1、点击商户名，数据未被破坏，显示成功 2、点击商户名，数据被破坏，显示失败				
预期情况	1	显示成功			
	2	显示失败			
实际结果	1、显示成功 2、显示失败				
结论	1、测试通过 2、测试通过				

5.4.1.16 用户模块（用户下订单）

用户下订单用例表如表3-16所示：

表 3-16 用户下订单测试用例表

测试编号	A016				
模块名称	用户下订单				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	用户点击商户列表，点击商户名，显示该商户的菜品列表，添加份数 、点击提交				
用例	1、用户点击商户列表，点击商户名，显示该商户的菜品列表，添加份数 、点击提交，点击结算，数据未被破坏，下订单成功 2、用户点击商户列表，点击商户名，显示该商户的菜品列表，添加份数 、点击提交，点击结算，数据被破坏，程序崩溃				
预期情况	1	下订单成功			
	2	下订单失败			
实际结果	1、、下订单成功 2、下订单失败				
结论	1、测试通过 2、测试通过				

5.4.1.17 用户模块（查看用户个人中心）

查看用户个人中心用例表如表3-17所示：

表 3-17 查看用户个人中心测试用例表

测试编号	A017				
模块名称	查看用户个人中心				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	用户点击个人中心				
用例	1、点击个人中心，数据未被破坏，显示成功 2、点击个人中心，数据被破坏，显示失败				
预期情况	1	显示成功			
	2	显示失败			
实际结果	1、显示成功 2、显示失败				
结论	1、测试通过 2、测试通过				

3.1.18 用户模块（编辑用户地址）

编辑用户地址用例表如表3-18所示：

表 3-18 编辑用户地址测试用例表

测试编号	A018				
模块名称	编辑用户地址				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	用户点击我的地址，填写详细地址、手机号码				
用例	1、填写详细地址：a3-117、手机号码:15545966666 2、填写详细地址：（没有）、手机号码:15545966666 3、填写详细地址：a3-117、手机号码：（没有）				
预期情况	1	编辑成功，提示“修改成功”			
	2	编辑失败，提示“请输入地址”			
	3	编辑失败，提示“请输入手机号码”			
实际结果	1、编辑成功，提示“修改成功” 2、编辑失败，提示“请输入地址” 3、编辑失败，提示“请输入手机号码”				
结论	1、测试通过 2、测试通过 3、测试通过				

5.4.1.19 用户模块（用户修改密码）

用户修改密码用例表如表3-19所示：

表 3-19 用户修改密码测试用例表

测试编号	A019				
模块名称	用户修改密码				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	填写原密码、新密码、重复密码（用户原密码为 123）				
用例	1、原密码正确 123、新密码 321、重复密码 321 2、原密码正确 123、新密码（没有）、重复密码 321 3、原密码正确 123、新密码 321、重复密码（没有） 4、原密码正确 123、新密码 321、重复密码 456 5、原密码不正确 456、新密码 321、重复密码（没有） 6、原密码不正确 456、新密码（没有）、重复密码 321 7、原密码不正确 456、新密码 321、重复密码（没有） 8、原密码不正确 456、新密码 321、重复密码 456				
预期情况	1	修改成功			
	2	请填写新密码			
	3	请填写重复密码			
	4	原密码和新密码不匹配			
	5	原密码不正确			
	6	原密码不正确			
	7	原密码不正确			
	8	原密码不正确			

续表：

表 3-19 用户修改密码测试用例表

实际结果	1、修改成功 2、请填写新密码 3、请填写重复密码 4、原密码和新密码不匹配 5、原密码不正确 6、原密码不正确 7、原密码不正确 8、原密码不正确
结论	1、测试通过 2、测试通过 3、测试通过 4、测试通过 5、测试通过 6、测试通过 7、测试通过 8、测试通过

5.4.1.20 商户模块（确认订单）

确认订单用例表如表3-20所示：

表 3-20 确认订单测试用例表

测试编号	A020				
模块名称	确认订单				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户查看新订单，若有新订单，并且商户接收接单，则点击确认订单，更改数据库中的订单状态				
用例	1、点击确认订单，数据未被破坏，接单成功 2、点击确认，数据被破坏，接单失败				
预期情况	1	接单成功			
	2	接单失败			
实际结果	1、接单成功 2、接单失败				
结论	1、测试通过 2、测试通过				

5.4.1.21 商户模块（取消订单）

查看订单详情用例表如表3-21所示：

表 3-22 取消订单（商户）测试用例表

测试编号	A021				
模块名称	取消订单（商户）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	商户查看新订单，若有新订单，并且商户拒接接单，则点击取消订单，更改数据库中的订单状态				
用例	1、点击取消，数据未被破坏，取消成功 2、点击取消，数据被破坏，取消失败				
预期情况	1	取消成功，并在历史订单中显示			
	2	取消失败			
实际结果	1、取消成功 2、取消失败				
结论	1、测试通过 2、测试通过				

3.1.22 用户模块（查看订单）

查看订单用例表如表 3-22 所示：

表 3-22 查看订单（用户）测试用例表

测试编号	A022				
模块名称	查看订单（用户）				
建立日期		建立人员		修改日期	
状态	[] 草稿 [] 正在修改 [√] 正式发布				
定义	用户点击“我的订单”，查看所有订单情况				
用例	1、点击我的订单，数据未被破坏，显示成功 2、点击我的订单，数据被破坏，显示失败				
预期情况	1	显示成功，查看最近订单情况			
	2	显示失败			
实际结果	1、显示成功 2、显示失败				
结论	1、测试通过 2、测试通过				

5.5 测试结果与结论

- 1、测试充分，安全性不足，
- 2、风险控制措施采取压力测试
- 3、测试目标基本完成
- 4、测试基本通过

5.5.1 bug 性质及模块分布图

bug性质及模块分布图如图4-1所示：

图4-1 bug类型及数量

注册 登录	Bug性质				
	A 类	B 类	C 类	D 类	总数
查看商家信息	0	0	0	0	0
增加菜品	0	0	0	0	0
删除菜品	0	0	0	0	0
查看菜品	0	0	0	0	0
修改菜品	0	0	0	0	0
商户查看订单	0	0	0	0	0
查看收益	0	0	0	0	0
添加优惠	0	0	0	0	0
删除优惠	0	0	0	0	0
查看商户列表	0	0	0	0	0
查看商户菜品列表	0	0	0	0	0
用户下单	0	0	0	0	0
查看用户个人中心	0	0	0	0	0
编辑用户地址	0	0	0	0	0
用户修改密码	0	0	0	0	0
确认订单	0	0	0	0	0
取消订单	0	0	0	0	0
用户查看退单	0	0	0	0	0
确认订单	0	0	0	0	0

软件测试合格须符合以下标准：

表 4-2 测试合格标准

A 类错误	B 类错误	C 类错误	D 类错误
无	无	<1%	<5%

5.5.2 功能

基本功能完成。

5.5.3 易用性

界面友好，便于使用，极大地提高了用户订餐便利、商户管理便利。

5.5.4 可靠性

系统可靠，能准确的实现用户和商户的需求。

5.5.5 兼容性

能兼容装有 java 运行环境的所有系统。

5.5.6 安全性

安全性较为可靠。

5.5.7 改进建议

优化代码、提高运行效率。

总结

完成了餐饮管理系统的需求分析、系统设计、编码、测试、打包发布等所有工作。基本完成了系统的所有功能。系统可以正确独立运行。

参考文献

- [1] “‘软件工程专业综合实践’教学大纲”. [C]. 黑龙江大学软件学院. 2017, 5, 2.
- [2] “(1) 项目开发计划”. [C]. 黑龙江大学软件学院. 2016, 4, 22.
- [3] “(2) 需求规格说明书”. [C]. 黑龙江大学软件学院. 2017, 3, 17.
- [4] “(3) 系统分析与设计报告”. [C]. 黑龙江大学软件学院. 2016, 5, 9.
- [5] “(4) 系统测试报告”. [C]. 黑龙江大学软件学院. 2016, 4, 22.
- [6] “(5) “软件工程专业综合实践”总结报告”. [C]. 黑龙江大学软件学院. 2017, 3, 17.
- [7] ““软件工程专业综合实践”教学大纲”. [C]. 黑龙江大学软件学院. 2017, 3, 17.

致谢

最感谢的是我们项目的导师丁大勇老师，作为我们的导师，总是耐心的为我们讲解项目的重点内容，积极指导我们。其次我要感谢是我们项目组的所有成员，每个人都那么认真负责，利用自己宝贵的时去完成自己的部分。