

开 发 计 划

(版本：2.0)

软件名称：智能订餐系统

开发小组：胡周珏，崔令祎

2020 年 6 月 28 日

目录

第一章 选题背景.....	3
第二章 市场调研.....	3
2.1 市场需求背景	3
2.2 市场前景分析	4
第三章 软件产品.....	5
3.1 概述功能	5
3.2 产品	5
3.3 项目开发环境	6
3.4 提交日期	6
3.5 验收标准	6
第四章 可行性分析.....	6
4.1 经济可行性	6
4.2 技术可行性	6
4.3 资源可行性	7
4.4 使用可行性	7
4.5 法律可行性	7
第五章 开发计划.....	7
5.1 开发环境	7
5.2 项目工作分解	8
5.3 项目进度安排	9
5.4 项目预算	10
5.5 技术分析	10
5.6 关键问题	11
参考文献.....	11

第一章 选题背景

我们小组选择了企业微信报餐订餐系统这一题。在给出的题目中我们比较倾向于开发移动端加上后台管理的完整系统。在学习了将近三年的程序设计后，我们也想检验自己的编程水平，想看看自学和利用网络资料的能力是否达到标准，是否能设计出符合市场需求的软件。而在所给的题目中，企业微信报餐订餐这一系统比较接近我们的日常生活，也是我们常常会用到的系统。相比其他系统，我们对它的需求比较清晰，对它的框架也比较了解，所以选择了它。这个软件的开发需要我们去学习许多之前没学过或仅仅接触过的东西，是困难也是挑战。

第二章 市场调研

2.1 市场需求背景

随着互联网的发展，智能手机的大范围普及，移动互联网走进了我们生活的方方面面，其易携带性使得人们更习惯于通过移动设备浏览互联网，人们的生活，办公和学习都可以在移动设备上进行。在这个背景下，越来越多的传统行业开始将自身与自动化智能化管理和移动互联网结合起来。

餐饮业的工作方式也发生着巨大的变化。在传统的餐饮业中，生产服务全是手工的。在用餐高峰期，客人往往需要排队点餐，由服务员记录下菜单后再交给主厨，点餐效率低下。由于长时间的等待，用户往往会产生负面情绪，影响就餐体验。对于工作人员，在人流量大时工作压力大，往往会手忙脚乱容易出错。这样的点餐方式，造成了很大的人力物力以及时间的浪费。再这样的背景下，餐饮行业也积极向互联网方向改革来寻求这些问题的解决方法。在一份点餐系统调查报告中，对于用户最希望实现的点餐系统功能这一问题下，有 82.3% 的用户选择了预定点餐，7.9% 的用户选择了菜品评价。可以见得预定点餐是点餐系统需要实现的主要功能。现在也有越来越多的餐厅开发了自己的手机点餐系统，比如海底捞，仓桥家等，不同餐厅的点餐系统可能在细节和有些功能上有所不同，但无一例外，它们都具有预约点餐功能。客户可以通过手机客户端完成餐厅菜品的图文浏览、在线下单以及在线结账。这种方式极大的提高了餐厅的信息化水平，客户

可以省去排队和等餐的时间，方便客户规划时间，实现到店立即就餐；还可以线上浏览菜单，无需到店就可知道有没有自己想要的菜品，增强了信息流动。而餐厅也可以减少服务员数量，降低成本。

在当下有的点餐系统中我们也可以发现，几乎所有的点餐软件都是基于微信等平台，很少有餐厅会选择单独开发 app，在我们调查中只有海底捞这种大型知名连锁餐饮品牌才具有。一方面，餐饮店太多，用户不可能每到一家店就下载一个应用，再一方面，餐饮店应用的使用频率并不高，用户没必要下载一个应用消耗内存。相较之下，微信具有非常大的用户群体，几乎每个手机上都有微信，客户使用微信小程序点餐，方便简洁，无需下载，不占内存，十分符合市场需求。

2.2 市场前景分析

企业微信订餐报餐系统要完成的是食堂订餐报餐的功能。来食堂就餐的客人利用小程序可以查看电子菜单，进行点餐、买单，还能对就餐进行评价。食堂则通过后台发布、修改菜单，接收客户的订单，统计营业数据进行分析。微信点餐系统是架起消费者与商家之间的桥梁，给餐厅消费者和管理人员都将带来全新的体验，其具有以下优势。

(1) 为餐厅节约大量的人力成本。传统点餐模式需要服务员记录客人所点菜单，人工生成订单，这样必然需要大量的人力资源。而点餐报餐系统不仅可以省下这部分的人力资源，还能减少因为人工原因造成的错误，提高商家的服务质量。

(2) 方便推广。客户端的开发是基于微信小程序的，微信小程序是一种不需要下载安装即可使用的应用，它基于微信平台，实现了应用“触手可及”，用户只需要扫一扫二维码或者搜索一下即可打开应用，不占手机内存，方便使用。而且当商家更新菜单或上新新菜时，也可以通过微信向客户推送。

(3) 系统收集分析餐厅各种数据，并不断优化。客户可以对点过的菜品进行评价，系统收集用户的评价反馈给商家，便于商家与客户之间的交流，利于商家对菜品口味，服务质量进行改进。同时系统对餐厅营业额，销售数据进行统计，能更加方便的进行财务管理。通过数据分析也可以得到用户最喜欢的菜，日销量最高的菜，对每天的材料采购提供一定的指导作用。

(4) 节约客户时间，提高点餐效率。客户可以通过点餐报餐系统提前预约点餐，缩短了用餐时间，方便快捷。

(5) 利于客户信息管理。传统的点餐模式中，新老客户的信息维护全靠人工记录，容易造成信息的丢失。而点餐报餐系统采用电子会员卡积分的形式，积分可以兑换优惠券或菜品，减少客户流失。

餐饮行业并不是一个小的市场，每个餐厅品牌想要信息化都需要自己的点餐小程序，点餐报餐系统具有较大的市场。而不同餐厅的点餐报餐系统大同小异，整体的功能和框架并不需要修改，只用根据商家的需求进行适当的调整，工作量并不大。有较好的经济前景。

第三章 软件产品

3.1 概述功能

本项目主要完成网上订餐系统，主要包括两部分内容：(1)用户订餐操作，包括浏览菜单、查看菜单信息等子模块；(2)系统后台管理：包括用户管理、菜单管理、订单管理、留言管理等子模块。

产品目标是实现食堂订餐系统的流畅性和方便顾客订餐用餐(包括网上菜品更新、网上订餐服务)，使服务效率提高，减少食物浪费现象。

3.2 产品

- 程序

程序名称：基于微信小程序的订餐报餐系统

使用语言：HTML, JavaScript, Python, SQL

- 文件

项目开发计划书；系统需求说明书；设计说明书；测试文档；软件开发日志；

系统维护手册；详细技术资料；用户手册；

- 服务

服务项目	开始日期	服务的期限
培训安装	2020年7月1日	3年

维护	2020 年 7 月 1 日	5 年
使用帮助	2020 年 7 月 1 日	5 年

3.3 项目开发环境

硬件环境：安装 Windows10 系统的 PC 机

软件环境：MySQL，Myeclipse，IntelliJ IDEA，微信小程序开发者工具

3.4 提交日期

2020 年 6 月 28 日

3.5 验收标准

客户与项目组首先模拟适用，再投入餐厅部分实验，中间共同发现问题、分析解决方法、完善系统产品，逐步应用于食堂订餐。在系统应用进一步成熟的过程中继续改进系统，或者将系统升级，使其具有更加强大的功能。

第四章 可行性分析

4.1 经济可行性

本系统设计需要多台 PC 和移动端，需要一台独立服务器，还需租借腾讯云服务器（学生优惠 10 元/月），需要进行项目开发、系统测试，后期系统维护。不同餐厅的点餐报餐系统大同小异，整体的功能和框架并不需要修改，只用根据商家的需求进行适当的调整。系统开发只有第一次需要投入大量人力物力，之后的点餐报餐系统开发，只需稍加修改，工作量小，效益大。

4.2 技术可行性

本系统是基于 JavaEE 研发的，在 Windows 操作系统下，使用 MySql 作为后端数据层支持，Tomcat 服务器，用 JavaScript 编写客户端微信小程序，利用 SpringBoot 搭建 web 项目构建系统后台。对此相关技术支持很多并且已经很成熟，在网络上也有很多学习资料，小组成员技术不足之处可在网络上自行学习，

所以在技术可行性方面没有任何问题。

4.3 资源可行性

软件资源: Windows 10

硬件资源: CPU 2 核 2.70GHz 及以上, 内存 8.00G 及以上, 100M 网卡, 硬盘 500G 及以上, 带液晶显示屏。

4.4 使用可行性

该产品使用方便简易, 软件设计的提供给后台操作人员的接口仅仅涉及到简单的文件的导入导出, 数据添加、修改、删除等操作, 因此仅仅需要操作人员熟悉简单电脑操作即可, 不需要专门进行培训。

客户端的使用会涉及到各种类型的人群, 微信小程序简单易上手, 凭借其简洁明了的 UI 和快捷的操作性, 并不要求用户对其特别熟悉, 因此可以做到使用方法简单易懂, 操作方法尽量浅显明了, 使用户能够在短时间内借助说明快速上手。为了提高系统的实用性, 要求具有较强的可靠性和较大的吞吐量。

4.5 法律可行性

该产品没有侵权或者抄袭等违法行为, 故可行。

第五章 开发计划

5.1 开发环境

本文所开发设计的点餐系统采用 B/S 结构, Web 服务器和数据库服务器都是部署在同一主机上。

操作系统	Windows
CPUC	2 核 2.5HZ 以上
内存	8G 以上
硬盘	500G 以上

WEB 服务器	Tomcat
数据库系统	Mysql

5.2 项目工作分解

小组在每次的作业中都需要共同讨论，具体编写文档工作分配给各位同学，我们初步完成了项目开发中需要完成的各项工作的分配，让大家对每一部分以及整体思路都有详细的了解。

负责人	阶段	工作任务
崔令祎	起 始 阶 段 (Inception)	<ul style="list-style-type: none"> • 需求分析; • 完成项目调研; • 进行可行性分析;
	细 化 阶 段 (Elaboration)	<ul style="list-style-type: none"> • 对层次功能进行分析，拟定程序结构; • 把握系统每部分状态；完成面向对象设计;
	构 建 阶 段 (Construction)	<ul style="list-style-type: none"> • 小程序模块详细设计
	测试维护阶段	<ul style="list-style-type: none"> • 软件审核 • 文档排版，打印、分发工作 • 用户培训工作
胡周珏	起 始 阶 段 (Inception)	<ul style="list-style-type: none"> • 分析调查结果，拟定实现方案 • 编制项目开发计划;
	细 化 阶 段 (Elaboration)	<ul style="list-style-type: none"> • 完成结构化设计; • 设计软件界面; • 对整体数据对象进行分析；设计数据库;
	构 建 阶 段 (Construction)	<ul style="list-style-type: none"> • 后台管理系统详细设计
	测试维护阶段	<ul style="list-style-type: none"> • 软件评定 • 文件的编制 • 软件安装工作

5.3 项目进度安排

里程碑	主要工作	应完成的文档	计划起始时间	计划终止时间
系统需求	项目调研	选题报告	2020.3.1	2020.3.5
	论证项目可行性		2020.3.5	2020.3.10
	指定项目初步计划		2020.3.10	2020.3.15
需求分析	确定系统运行环境	需求说明书	2020.3.5	2020.3.6
	确定系统功能及性能要求		2020.3.6	2020.3.12
	确定项目开发计划		2020.3.13	2020.3.18
概要设计	建立系统总体结构，划分功能模块	中期报告-1	2020.3.21	2020.3.23
	定义各功能模块接口		2020.3.21	2020.3.23
	数据库设计		2020.3.24	2020.3.25
	制定组装测试计划		2020.3.25	2020.3.26
详细设计	设计各模块具体实现算法	设计说明书	2020.3.26	2020.3.29
	确定模块间详细接口		2020.3.30	2020.3.31
	制定模块测试方案		2020.4.1	2020.4.2
实现	编写程序源代码	中期报告-2	2020.4.2	2020.4.12
	进行模块测试和调试		2020.4.12	2020.4.13
	编写用户手册		2020.4.13	2020.4.15
	对实现过程和已完成的文档进行评审		2020.4.15	2020.4.16
集成测试	执行集成测试计划	测试文档	2020.4.17	2020.4.20
验收测试	测试整个软件系统		2020.4.21	2020.4.22
	试用用户手册	用户手册	2020.4.22	2020.4.24
	编写开发总结	课程论文	2020.4.24	2020.4.30
维护	为纠正错误，完善应用而进行修改			
	对修改进行配置管理			
	编写故障报告和修改报告			
	修订用户手册			

5.4 项目预算

事项	预算
项目调查, 购买资料 (3月1日)	100 元
租用域名、云服务器 (3月3日)	200 元
后期宣传 (7月1日)	3000 元

5.5 技术分析

1) SpringBoot

SpringBoot 框架简化了 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，默认配置了很多框架的使用方式，从而使开发人员不再需要定义样板化的配置。我们平时搭建一个 Spring Web 项目的时候需要配置 web.xml，加载 Spring 和 Springmvc，配置数据库连接、Spring 事务、日志文件等等，配置完成后还需要部署 Tomacat 调试。但是如果使用 SpringBoot，我们仅仅需要非常少的几个配置就可以迅速方便的搭建起来一套 web 项目，而且我们不用关心框架之间的兼容性，适用版本等各种问题，适合初学者，所以我们选择了 SpringBoot。

2) FreeMarker

FreeMarker 是一个用 Java 语言编写的模板引擎，它基于模板来生成文本输出。而在 Java 领域，表现层技术主要有三种：Jsp、FreeMarker 和 Velocity。我们选择了 FreeMarker 有以下原因。首先，我们对 web 开发并不熟悉，而 FreeMarker 支持 jsp 标签，宏定义比 jsp tag 方便，同时内置了大量常用功能，比如 html 过滤，日期金额格式化等等。功能代码十分简洁，上手快，使用非常方便。其次，我们是小组合作，需要分工明确。jsp 页面前后端的代码写到了一起，耦合度很高，前台的需要熟悉后台环境，需要去调试；后台的要熟悉界面设计技术。交替性的工作需要花费一定的学习成本，效率低下。而使用 FreeMarker，前后端完全分离，小组成员分头工作，互不影响，提高效率。

3) 数据库

在系统中，数据库相关的环节很多，需要了解他的性能和一些基本的操作常

识。在之前的数据库课程中，我们掌握了 Mysql 的使用，较为熟悉，故选择 Mysql。

4) Web 服务器

Web 服务器主要是针对于配置和部署，对目录的配置，调试；对配置文件属性的修改；对访问权限和并发性的控制；Java 类的部署等。SpringBoot 默认使用的是 Tomcat 作为内嵌的服务器，故选择 Tomcat。

5.6 关键问题

- **产品大小：**实践经验表明项目风险和产品的大小成正比。公认产品大小度量单位是以功能点计。我们的订餐系统需要尽可能满足商户及顾客的各种需求。
- **团队人员经验：**我们的团队面临着经验不足的风险。
- **客户因素：**客户需求经常变化，商家或者顾客不了解的该订餐系统的使用方法，并且商家和顾客之间会因为口味等因素发生沟通问题等。
- **应对方案：**
 - (1)尽可能提供全套的订餐服务以满足商家和顾客的需求。
 - (2)团队成员尽可能熟悉研发工具，使其变得更加得心应手，更好的为该订餐系统服务。
 - (3)多与客户进行沟通，了解他们的想法，并尽可能将想法实现。

参考文献

[1]CSDN. 当开发人员遇上非功能性需求[EB/OL].

<https://baijiahao.baidu.com/s?id=1624165373473108545&wfr=spider&for=pc> 百度百科-软件开发, <https://baike.baidu.com/item/软件开发>, 2019-01-31.

[2]概览 | 微信开放文档[EB/OL].

<https://developers.weixin.qq.com/miniprogram/dev/devtools/devtools.html>

[3]餐饮点餐小程序市场优势在哪里[EB/OL].

<http://www.shunjian8.com/xcxdt/7214.html>, 2018-09-07.

[4]软件开发可行性分析报告[EB/OL].

<https://wenku.baidu.com/view/e3c605af590216fc700abb68a98271fe910eaf8c.html?qq-pf-to=pcqq.c2c>, 2019-10-06.

《软件综合实习》

业
务
需
求

(版本：2.0)

软件名称：智能食堂订餐系统

开发小组：胡周珏，崔令祎

2020 年 6 月 28 日

目录

第一章 概述.....	3
第二章 现有业务.....	3
第三章 业务建议.....	4
3.1 概述	4
3.2 功能需求	4
3.3 非功能需求.....	4
第四章 业务场景.....	5
参考文献.....	9

第一章 概述

1) 编写目的

本文档的编写目的是对智能食堂订餐系统的软件需求进行描述和规约，为后续的系统分析、设计和实现工作奠定基础。文档将详细地定义系统的功能和非功能需求，获取场景和用例。本文档也用于和客户进行沟通，明确客户需求的细节。

2) 适用范围

本文档适用的软件为：智能食堂订餐系统

与该软件相关的特性、子系统、模型等均符合本文档中的内容。

3) 定义

本文档中涉及的术语定义在项目词汇表中给出。

4) 参考资料

- 《面向对象软件工程——使用 UML、模式与 Java》(第 3 版)，清华大学出版社，2011.
- IEEE Recommended Practice for Software Requirements Specifications, IEEE Std830-1998

5) 概述

本文档包括简介、现有业务和建议的业务三部分。目前系统部分对当前智能食堂订餐系统进行分析，指出其不足并给出本系统开发的必要性；建议部分列举系统的功能需求，从不同方面规定非功能需求。该部分还描述了系统的各种场景并归纳为系统的用例，如登录、检索、下单等。

第二章 现有业务

当下有的点餐系统中我们也可以发现，几乎所有的点餐软件都是基于微信等平台，很少有餐厅会选择单独开发 app，在我们调查中只有海底捞这种大型知名连锁餐饮品牌才具有。一方面，餐饮店太多，用户不可能每到一家店就下载一个应用，再一方面，餐饮店应用的使用频率并不高，用户没必要下载一个应用消耗内存。

• 弊端和局限性

这种订餐方式存在如下弊端和局限性：

- a) 下载不方便
- b) 注册过程麻烦

- c) 多个商家有不同的平台

第三章 业务建议

3.1 概述

功能需求主要有用户登录、浏览查找、订餐、评价等，非功能需求包括高可靠性、操作便捷、响应快速和其他性能，系统模型包括具体的参与者和用例，参与者有用户和商家，用例包括发布、查看信息、点餐等。

3.2 功能需求

1) 用户需求

用户登录后可以管理个人信息查看订单；当用户在进入该商家的订餐页面后，用户可以收藏、联系卖家、购买、确认交易或取消交易。用户填写订单时需要提供姓名、联系方式、交易时间、交易地点，确认菜品数量和价格等信息，该系统支持在线支付，也可以确认交易时间和地点后线下交易，交易完成后由买家确认交易；用户可以浏览菜品浏览、查看和搜索；当买方用户在交易结束后未确认交易，卖方用户可以联系管理员寻求帮助。

2) 菜品需求

菜品信息主要包括名称、类型、价格、上架时间、数量、图片和简介等，其中价格需要在合理范围内。不得发布违反法律法规的商品，用户可随时联系管理员举报。

3.3 非功能需求

1) 可用性

年轻用户对于此类 App 都很熟悉。本系统较为简单及界面友好，同时类似产品目前广泛使用，因此可以认为用户不需要培训即可使用本产品的全部功能。

2) 可靠性

本系统上线后如无维护等情况全天运营。普通用户能对菜品只可查看，不可更改。商家管理人员可对任意数据项进行读写操作。

3) 性能

- 响应时间需求。本系统对用户请求合适的平均响应时间在 0.5s 左右，最长响应时间不应该超过 2s。

- 吞吐量需求。考虑实际情况，每秒处理的请求在 1000 条以下。因此本系统吞吐量不大，并发数要求不高。
- 容量需求。对于本系统的适用范围，菜品数为 10000，在线用户为 50。
- 资源需求。初期预测本系统负载不大，预计不会超 20 万条记录，内存占用不超过 1GB。预计磁盘占用不超过 10GB。对于网络请求，假定页面平均请求 100Kb，每秒请求为 100 次，因此预计需要大约 10 M bps 的带宽。

4) 接口

- 用户界面

本系统用户界面为手机端的小程序。界面开发时采用美观简洁设计。

- 硬件接口

本系统为 C/S 架构，需要硬件为能够运行浏览器的设备。

- 软件接口

本系统使用 MySQL Server、小程序 API。

5) 法律及版权声明

本系统使用个人编写及开源代码构成，使用开源部分遵守开源代码所采用的协议，非开源部分版权归本开发组所有。本系统（包括但不限于软件、使用等方面）最终解释权归本开发组所有。

6) 适用的标准

计算机软件开发规范 GB 8566—88;

计算机软件单元测试指南 GB/T 15532—95;

软件维护指南 GB/T 14079—93;

计算机软件可靠性和可维护性管理 GB/T 14394—93

第四章 业务场景

场景名称	登录
参与者	用户 A
事件流	1. A 在页面上单击“登录”按钮，进入登录页面。 2. A 在登录界面填写一份包含用户名和登录密码的表单并提交。

	3. 系统收到 A 的登录请求，进行验证后，A 登录成功
--	------------------------------

场景名称	修改个人资料
参与者	用户 A
事件流	<p>1. A 在页面上单击“资料管理”按钮，进入个人资料页面查看用户名、手机号等内容。</p> <p>2. A 在个人资料页面单击“修改个人资料”按钮，进入个人资料修改页面</p> <p>3. A 在个人资料修改页面编辑修改自己的用户名、手机号等内容，修改完成后点击“确认修改”按钮。</p> <p>4. 系统收到 A 的修改请求，完成修改，返回个人资料页面。</p> <p>5. A 在个人资料页面单击“修改登录密码”按钮，进入登录密码修改页面。</p> <p>6. A 在登录密码修改页面输入新的登录密码，并再次输入以确认，填写完成后点“确认修改”按钮。</p> <p>7. 系统收到 A 的修改请求，完成修改，返回个人资料页面。</p>

场景名称	查看订单
参与者	用户 A
事件流	<p>1. A 单击“个人商品信息”按钮，进入个人商品信息查看页面。</p> <p>2. A 在个人商品信息查看页面单击“我的订单”按钮，进入个人订单查看页面，可查看所有自己的订单。</p> <p>3. 系统收到请求，跳转到个人订单查看页面，展示 A 已购订单的商品名称、图片</p>

场景名称	发布菜单
参与者	商家 A
事件流	<p>1. A 在页面上单击“发布”按钮，进入商品发布页面。</p> <p>2. A 在商品发布页面填写名称、描述、价格，选择商品类别，并上传照片，完成后单击“确认发布”按钮。</p> <p>3. 系统管理员收到 A 的发布请求，检查发布信息是否有效。</p>

	4. 通过 A 的发布请求。 5.通知 A 发布请求已通过，并将该菜单加入列表中
--	---

场景名称	订单管理
参与者	商家 A
事件流	1. A 单击“商家信息”按钮，进入信息查看页面。 2. 在信息查看页面单击“我的菜单” 按钮，进入已发布菜单查看，查看所有自己发布的菜品。 3. 在信息查看页面单击“订单”按钮，进入订单查看用例，查看所有的订单。

场景名称	查看收藏
参与者	用户 A
事件流	1. A 单击“我的”按钮，进入个人信息查看页面。 2. A 在个人信息查看页面单击“我的收藏” 按钮，进入个人已收藏商品查看页面，可查看所有自己收藏的菜品。 3. 系统收到请求，跳转到个人收藏查看页面，展示 A 收藏商品的商品名称、图片和价格。

场景名称	查看消息
参与者	用户 A
事件流	1. A 单击“消息管理”按钮，进入消息管理页面。 2. A 在消息管理页面中查看历史消息的列表。 3. A 单击其中的一条消息，进入对话。

场景名称	在线对话
参与者	用户 A, 商家 B
事件流	1. A 进入和 B 对话页面，发送消息。 2. 系统将对话消息发送给 B，B 接收消息进入对话页面。

	<p>3.B 和 A 在对话页面中进行交流。</p> <p>4. 两人通过关闭对话页面，退出对话。</p>
--	---

场景名称	浏览菜单
参与者	用户 A
事件流	<p>1. 开启应用后，可通过上划、下划浏览菜单上的菜品。</p> <p>2. 菜品按分类类别顺序排列，A 可以单击某类别按钮查看该类别菜品。</p> <p>3. A 通过上下划动浏览该类别菜品，找到自己感兴趣的菜品时单击该对应区域进入详情页面中，可通过单击“返回”按钮返回前一个浏览页面。</p>

场景名称	搜索菜品
参与者	用户 A
事件流	<p>1. 单击在浏览界面上方的搜索框，输入自己想搜索的商品的关键词后单击“搜索”按钮。</p> <p>2. 系统通过关键词在所有商品中检索并返回与关键词符合度高的商品，在结果页面中显示。</p> <p>3. 如果没有结果返回，可修改关键词后重复 1</p> <p>4. 浏览结果页面单击想找的商品对应区域进入该商品的页面中。</p>

场景名称	查看菜品
参与者	用户 A
事件流	<p>1. A 进入某一菜品的页面，浏览详细信息，包括名称、类型、价格、供应时间、剩余数量、图片和简介等。</p> <p>2. A 单击“返回”按钮返回前一个浏览页面。</p>

场景名称	下单
参与者	用户 A, 商家 B
事件流	<p>1. A 查看一件商品的页面，单击页面中的“购买”按钮。</p> <p>2. 系统跳转到购买页面。</p>

	<p>3. A 填写页面订单中的姓名、联系方式、取餐时间、地点，确认菜品数量和价格等信息，单击“确认”按钮提交。然后等待 B 的确认。</p> <p>4. 系统将购买信息发送给 B。B 查看订单的信息并确认交易。</p> <p>5. 系统将 B 的确认消息发送给 A，下单成功。</p>
--	---

场景名称	订单完成
参与者	用户 A, 商家 B
事件流	<p>1. A 和 B 在线下完成了交易。</p> <p>2. A 进入订单页面，通过单击确认按钮交易结束。</p> <p>3. B 收到消息，进行确认，交易结束</p>

参考文献

- [1]百度文库.餐厅点餐概要设计说明书[EB/OL].<https://wenku.baidu.com/view/a582ec7a00768e9951e79b89680203d8ce2f6ac9.html>,2019.
- [2] qq_36501609.微信点餐需求分析[EB/OL].https://blog.csdn.net/qq_36501609/article/details/89929225,2019-05-07.
- [3] 编程小石头.点餐系统的开发[EB/OL].https://blog.csdn.net/qiushi_1990/article/details/97749686,2020-02-07.
- [4] 重庆冰炫科技有限公司.冰点云订餐食堂报餐点餐[EB/OL].<http://www.cmerp.com/>,2017.
- [5] 嘻唰唰.手机微信订餐报餐系统[EB/OL].<https://www.xeeyee.cn/>,2017.

系统需求说明书

(版本：2.0)

软件名称：智能食堂订餐系统

开发小组：胡周珏，崔令祎

2020 年 6 月 28 日

目录

1 引言	3
1.1 选题背景	3
1.2 市场调研	3
2 软件需求分析	5
2.1 功能需求与非功能需求	5
2.2 软件需求建模	6
2.3 面向对象设计	9
2.4 初步概要设计	11
3 系统模型	11
3.1 用例模型	11
3.2 用户界面	15
参考文献.....	19

1 引言

1.1 选题背景

我们小组选择了企业微信报餐订餐系统这一题。在给出的题目中我们比较倾向于开发移动端加上后台管理的完整系统。在学习了将近三年的程序设计后，我们也想检验自己的编程水平，想看看自学和利用网络资料的能力是否达到标准，是否能设计出符合市场需求的软件。而在所给的题目中，企业微信报餐订餐这一系统比较接近我们的日常生活，也是我们常常会用到的系统。相比其他系统，我们对它的需求比较清晰，对它的框架也比较了解，所以选择了它。这个软件的开发需要我们去学习许多之前没学过或仅仅接触过的东西，是困难也是挑战。

1.2 市场调研

1.2.1 市场需求背景

随着互联网的发展，智能手机的大范围普及，移动互联网走进了我们生活的方方面面，其易携带性使得人们更习惯于通过移动设备浏览互联网，人们的生活，办公和学习都可以在移动设备上进行。在这个背景下，越来越多的传统行业开始将自身与自动化智能化管理和移动互联网结合起来。

餐饮业的工作方式也发生着巨大的变化。在传统的餐饮业中，生产服务全是手工的。在用餐高峰期，客人往往需要排队点餐，由服务员记录下菜单后再交给主厨，点餐效率低下。由于长时间的等待，用户往往会产生负面情绪，影响就餐体验。对于工作人员，在人流量大时工作压力大，往往会手忙脚乱容易出错。这样的点餐方式，造成了很大的人力物力以及时间的浪费。再这样的背景下，餐饮行业也积极向互联网方向改革来寻求这些问题的解决方法。在一份点餐系统调查报告中，对于用户最希望实现的点餐系统功能这一问题下，有 82.3% 的用户选择了预定点餐，7.9% 的用户选择了菜品评价。可以见得预定点餐是点餐系统需要实现的主要功能。现在也有越来越多的餐厅开发了自己的手机点餐系统，比如海底捞，仓桥家等，不同餐厅的点餐系统可能在细节和有些功能上有所不同，但无一

例外，它们都具有预约点餐功能。客户可以通过手机客户端完成餐厅菜品的图文浏览、在线下单以及在线结账。这种方式极大的提高了餐厅的信息化水平，客户可以省去排队和等餐的时间，方便客户规划时间，实现到店立即就餐；还可以线上浏览菜单，无需到店就可知道有没有自己想要的菜品，增强了信息流动。而餐厅也可以减少服务员数量，降低成本。

在当下有的点餐系统中我们也可以发现，几乎所有的点餐软件都是基于微信等平台，很少有餐厅会选择单独开发 app，在我们调查中只有海底捞这种大型知名连锁餐饮品牌才具有。一方面，餐饮店太多，用户不可能每到一家店就下载一个应用，再一方面，餐饮店应用的使用频率并不高，用户没必要下载一个应用消耗内存。相较之下，微信具有非常大的用户群体，几乎每个手机上都有微信，客户使用微信小程序点餐，方便简洁，无需下载，不占内存，十分符合市场需求。

1.2.2 市场前景分析

企业微信订餐报餐系统要完成的是食堂订餐报餐的功能。来食堂就餐的客人利用小程序可以查看电子菜单，进行点餐、买单，还能对就餐进行评价。食堂则通过后台发布、修改菜单，接收客户的订单，统计营业数据进行分析。微信点餐系统是架起消费者与商家之间的桥梁，给餐厅消费者和管理人员都将带来全新的体验，其具有以下优势。

(1) 为餐厅节约大量的人力成本。传统点餐模式需要服务员记录客人所点菜单，人工生成订单，这样必然需要大量的人力资源。而点餐报餐系统不仅可以省下这部分的人力资源，还能减少因为人工原因造成的错误，提高商家的服务质量。

(2) 方便推广。客户端的开发是基于微信小程序的，微信小程序是一种不需要下载安装即可使用的应用，它基于微信平台，实现了应用“触手可及”，用户只需要扫一扫二维码或者搜索一下即可打开应用，不占手机内存，方便使用。而且当商家更新菜单或上新新菜时，也可以通过微信向客户推送。

(3) 系统收集分析餐厅各种数据，并不断优化。客户可以对点过的菜品进行评价，系统收集用户的评价反馈给商家，便于商家与客户之间的交流，利于商家对菜品口味，服务质量进行改进。同时系统对餐厅营业额，销售数据进行统计，

能更加方便的进行财务管理。通过数据分析也可以得到用户最喜欢的菜，日销量最高的菜，对每天的材料采购提供一定的指导作用。

(4) 节约客户时间，提高点餐效率。客户可以通过点餐报餐系统提前预约点餐，缩短了用餐时间，方便快捷。

(5) 利于客户信息管理。传统的点餐模式中，新老客户的信息维护全靠人工记录，容易造成信息的丢失。而点餐报餐系统采用电子会员卡积分的形式，积分可以兑换优惠券或菜品，减少客户流失。

餐饮行业并不是一个小的市场，每个餐厅品牌想要信息化都需要自己的点餐小程序，点餐报餐系统具有较大的市场。而不同餐厅的点餐报餐系统大同小异，整体的功能和框架并不需要修改，只用根据商家的需求进行适当的调整，工作量并不大。有较好的经济前景。

2 软件需求分析

2.1 功能需求与非功能需求

功能性需求 (特性)	<ol style="list-style-type: none"> 1. 菜名，价格必须存储在用户和其他人访问的数据库中。 2. 用户通过微信账号自动登录 3. 点餐时有查询功能，方便用户搜索自己喜欢的菜品，点击该菜品图片显示该菜品的详细信息 4. 下单后未付款时可以取消订单。付款后订单生成，订单信息系统自动存储 5. 用户可以查阅以往订过的餐品信息 6. 餐品评价中顾客可以对这次服务质量，留下自己相对餐厅说的话，完成用户与餐厅之间的交互 7. 用户下单时可以使用抵用券获得优惠 8. 菜品管理中可以对菜品信息进行添加、修改和查询操作 9. 订单管理中可以通过未确认、已确认、已下单三种形式进行管理、也可查看所有订单信息； 10. 用户生成订单后，系统向后台发送通知，商家选择是否成交；
非功能性需求	<ol style="list-style-type: none"> 1. 这个系统在任何时候都必须是可用的。一周中只允许有 2 分钟宕机时间。 2. 所有提交数据一般在 3 秒内响应，及时更新每个时间段已

	<p>下单数量</p> <p>3. 对于出错情况应该及时显示出错信息，比如“此时间段订单数量已达上限”</p> <p>4. 这个系统应该设计成可以处理任何一个餐饮商家的运营需要。</p> <p>5. 界面简洁，操作简单，菜品、通知等信息的录入工作简便</p> <p>6. 客户预定信息的处理（包括录入、删除）“即时发布、即时见效”</p> <p>7. Web 开发中服务器代码对输入的参数进行验证</p> <p>8. 包括数据备份、恢复、日志管理、垃圾数据清除等基本功能</p> <p>9. 需要具备可持续性和可维护性。</p> <p>10. 本应用可以根据现实的变化以及消费者或者卖家的需求及时改变，适应性灵活性好。</p>
--	---

2.2 软件需求建模

1) 功能分析

针对食堂排队浪费时间等问题，我们提出制作一个软件，提供两个方面的服务：面向学生或公司职工，用户通过微信小程序登录，打开软件即可根据需要选择自己想要的饭菜，饭菜做好后卖家在系统确认即可发送短信给用户到食堂取餐；食堂在后台可以上传菜品信息，提供预约、点餐等服务。

行为模型如下——Figure 1-用户端状态图、Figure 2-商家端状态图

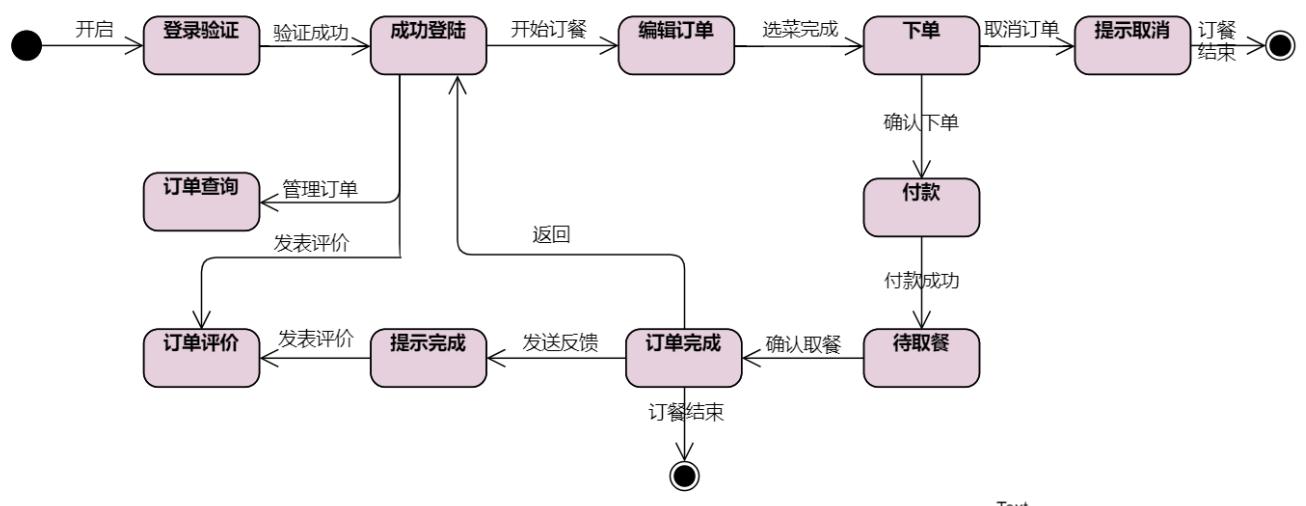


Figure 1-用户端状态图

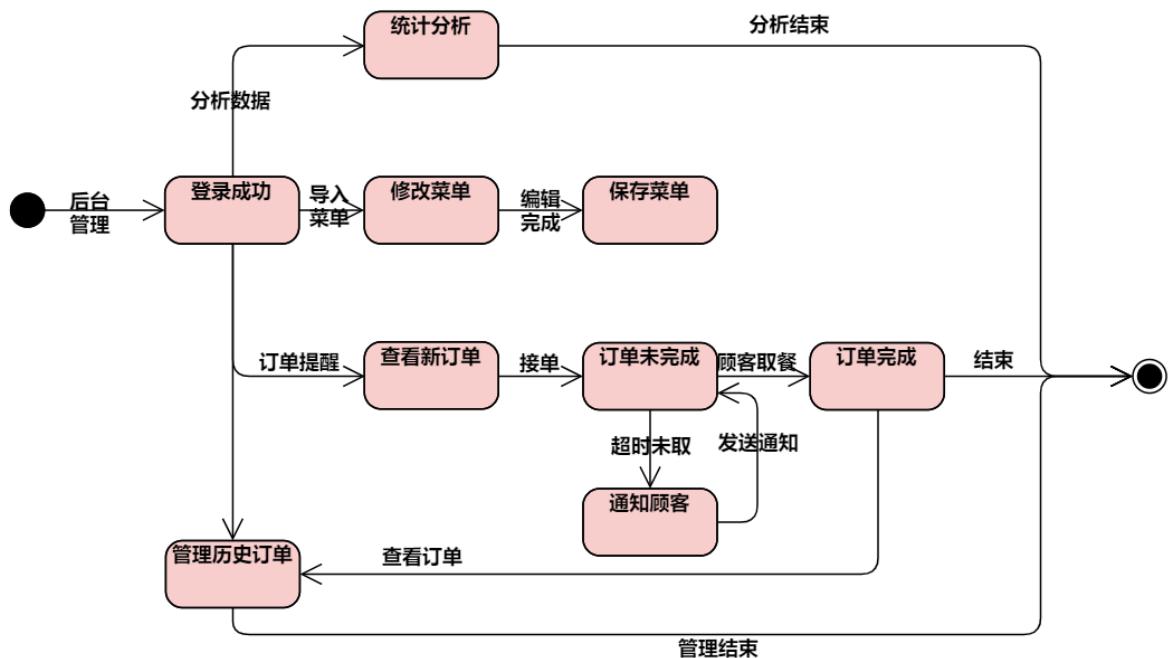


Figure 2-商家端状态图

2) 数据流

研究数据如何流动，由数据流、加工以及文件、端点等成份构成的数据流图 DFD 如下：

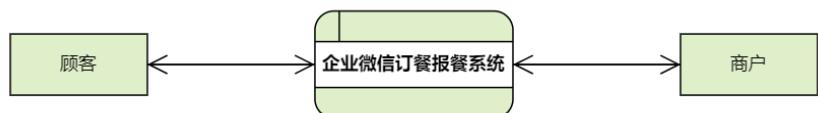


Figure 3 顶层图

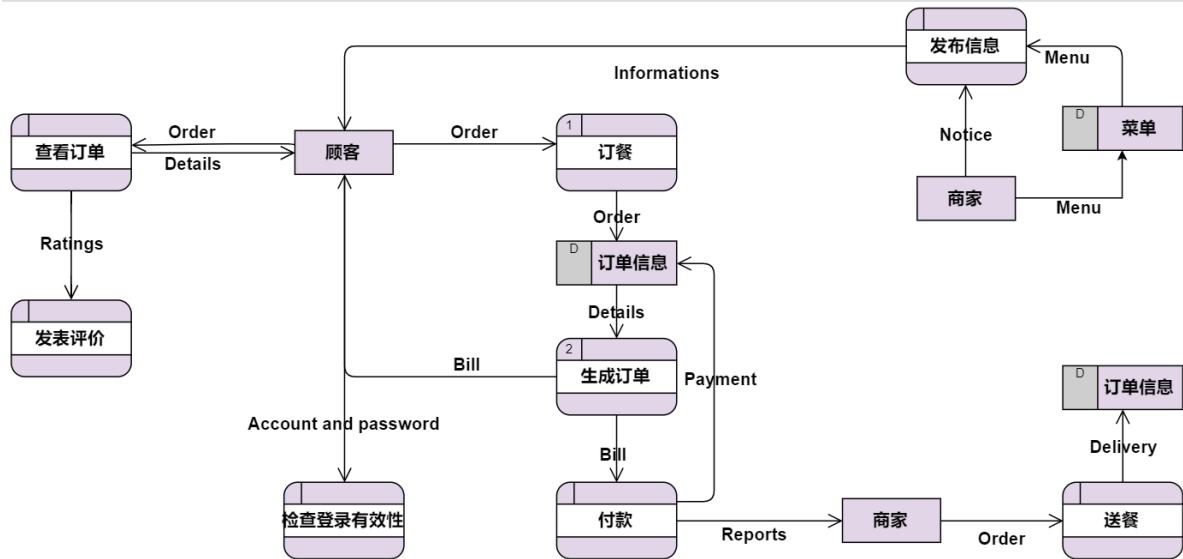


Figure 4 细化数据流图

3) 信息建模

核心概念是实体和关系，主要工具是语义数据模型 Figure 5-ER 图。要找出现实世界的对象，然后用属性来描述对象。

实体

- 顾客（顾客号，用户名，手机号，密码，地址）
- 商家（商家编号，名称，密码，菜单编号）
- 菜单（菜单编号，商家编号，菜名，价格，会员价，基本描述，创建时间）
- 订单（订单号，商家号，顾客号，送餐地址，联系电话，菜单列表，金额，订单生成时间，订单状态）
- 反馈（信息号，顾客号，订单号，时间，内容）

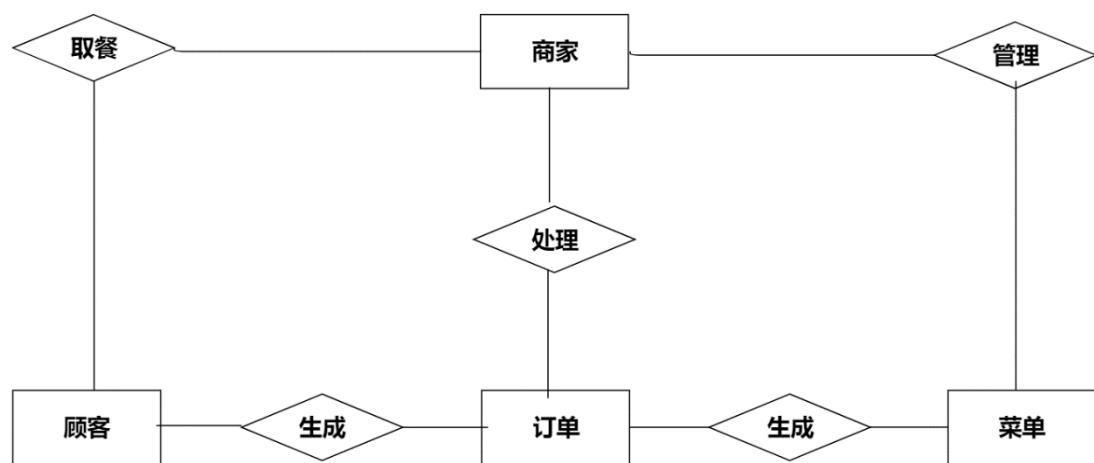


Figure 5-ER 图

2.3 面向对象设计

a) 序列图

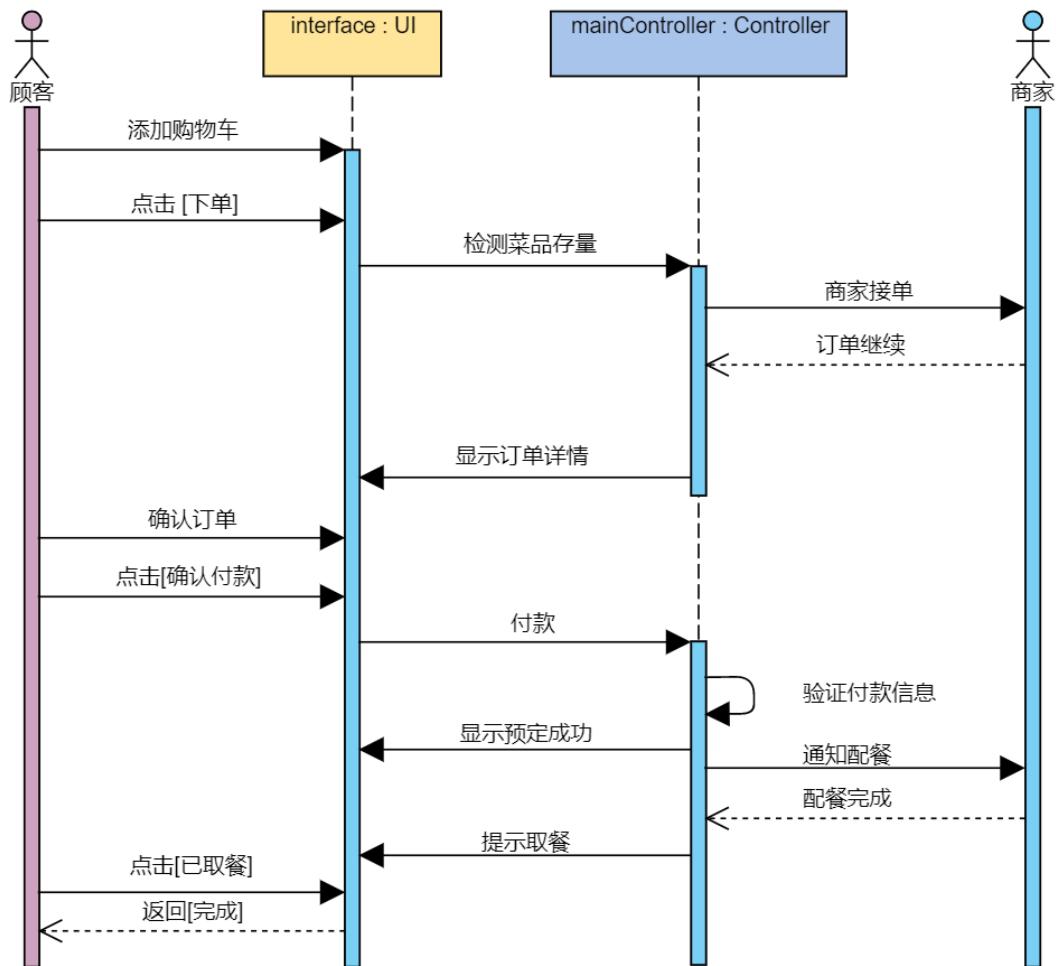


Figure 6-订餐模块序列图

b) 活动图

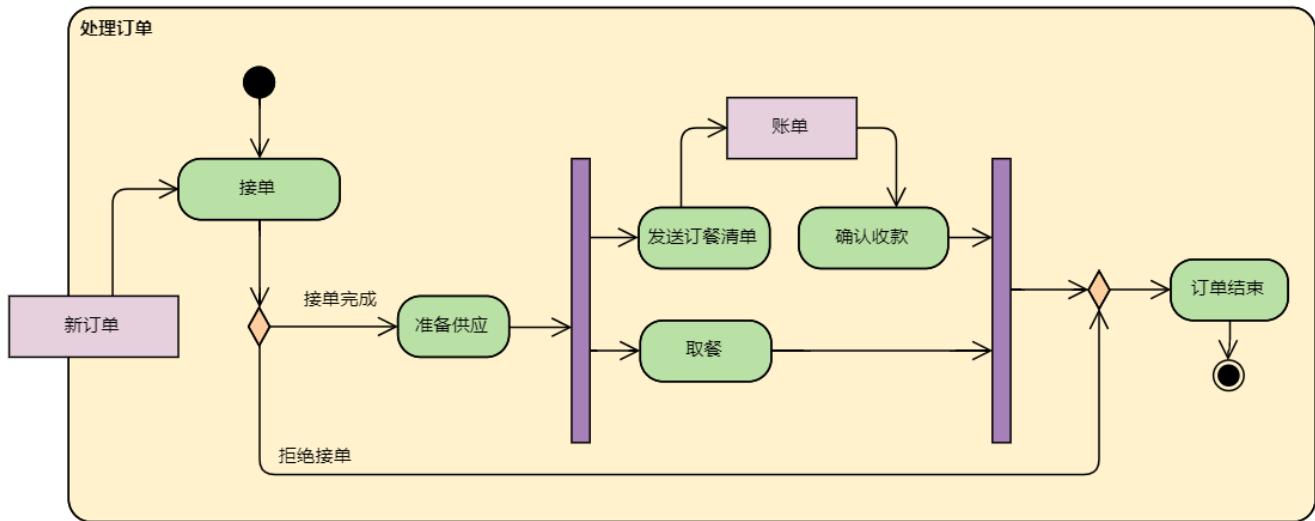


Figure 7-订餐模块活动图

c) 类视图

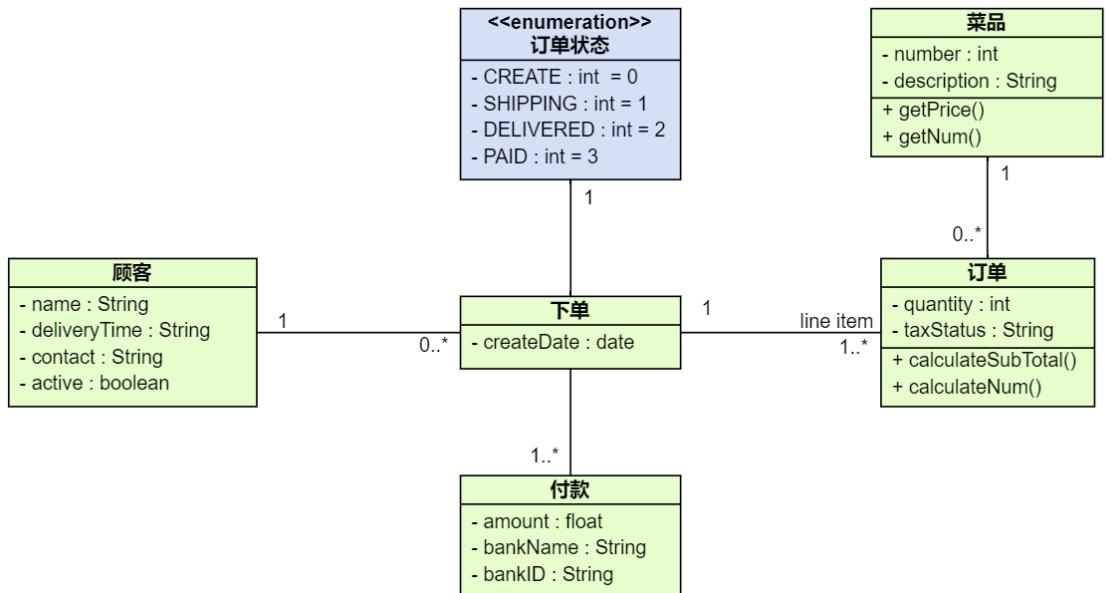


Figure 8-点餐系统类图

2.4 初步概要设计

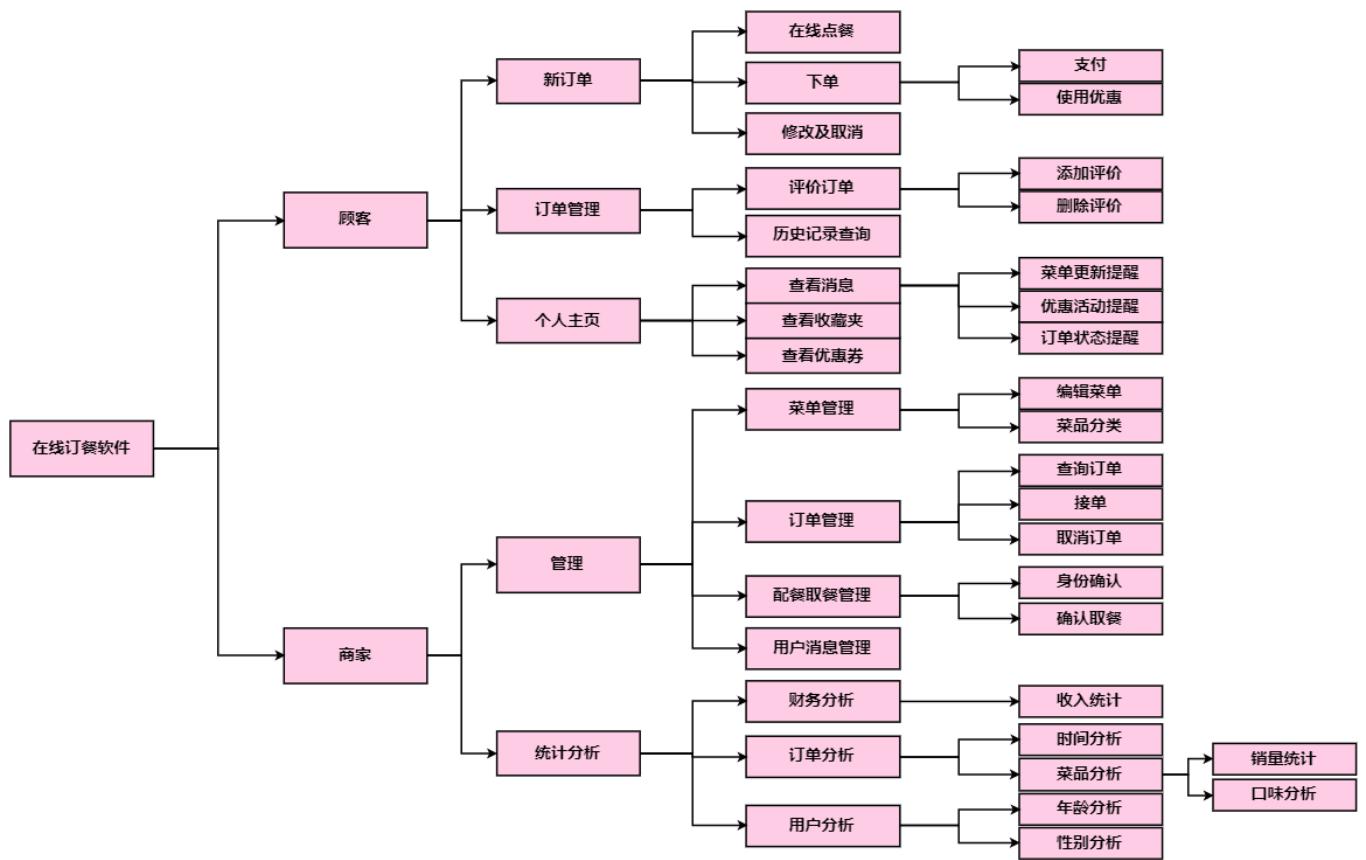


Figure 9-模块图

3 系统模型

3.1 用例模型

Table 1 参与者信息表

名称	解释
用户	登录的客户，可以使用信息管理、浏览搜索、点餐等所有系统提供的用户功能
商家	进行后台管理菜单和订单等操作

Table 2 用例信息表

编号	用例名称	用例级别	描述
01	Login	子功能	用户登录，系统获取用户信息
02	ManagePersona!Information	用户目标	修改用户的用户名、手机等

03	PublishItem	用户目标	发布菜品并填写简要描述
04	CheckDialogs	用户目标	查看消息管理页面
05	SearchItem	子功能	搜索
06	FavorItem	子功能	收藏
07	PurchaseItem	用户目标	下单, 卖家确认
08	CloseDeal	子功能	确认取餐, 交易结束
09	CheckOrder	子功能	查看我的订单

- 用例详细信息表如下

用例名称	login
范围	系统用例
级别	子功能
主要参与者	用户
前置条件	用户进入个人页面
后置条件	进入登录状态
主流程	<ol style="list-style-type: none"> 用户单击页面上的“登录”按钮。 系统收到用户的请求,跳转到登录页面,展示登录表单供填写。 用户填写登录表单的用户名和登录密码,填写完成后单击“登录”按钮。 系统判别表单信息是否有效。 登陆成功
扩展流程	无
发生频率	随时

用例名称	PublishItem
范围	系统用例
级别	用户目标
主要参与者	商家
前置条件	已登录
后置条件	更新菜单
主流程	<ol style="list-style-type: none"> 商家在页面上单击“发布”按钮。 系统收到请求, 跳转到菜单发布页面。 用户在菜单发布页面填写名称、描述、价格, 选择类别, 并可以上传照片, 完成后单击“确认发布”按钮。 系统收到发布请求, 检查发布信息是否有效

扩展流程	可能菜单信息无效，要求完善后重新提交
发生频率	随时

用例名称	SearchItems
范围	系统用例
级别	子功能
主要参与者	用户
前置条件	搜索框
后置条件	用户进入某一菜品的页面或返回空页面
主流程	<ol style="list-style-type: none"> 1. 用户单击搜索框，输入自己想找的商品的关键词后单击“搜索”按钮 2. 系统通过关键词在所有商品中检索并返回与关键词符合度高的商品，在结果页面中显示。 3. 如果没有结果返回，修改关键词后重复步骤 1。 4. 用户浏览结果页面单击想找的商品对应区域进入该商品的页面中。
扩展流程	无
发生频率	随时

用例名称	Purchaseltem
范围	系统用例
级别	用户目标
主要参与者	用户，商家
前置条件	用户进入菜单页面
后置条件	更新菜单状态；反馈结果给用户
主流程	<ol style="list-style-type: none"> 1. 顾客浏览商品或者搜索商品进入查看菜单页面，单击页面中的“购买”按钮。 2. 系统接收顾客用户的请求，跳转到购买页面。 3. 用户编辑页面订单中的姓名、联系方式、取餐时间地点，确认餐品数量和价格等信息，单击“确认”按钮提交，等待商家的处理。 4. 系统将购买信息发送给商家。商家查看订单的信息并确认交易。 5. 系统将商家的确认消息发送给用户，订餐成功
扩展流程	<ol style="list-style-type: none"> 1. 商家取消订单。第 4 步，卖家查看订单选择取消交易，调用 CancelOrder 2. 顾客取消订单。第 4 步，买家查看订单选择取消交易，调用 CancelOrder 3. 未登录。 在第 1 步，用户尚未登录： (1) 跳转登录页面，进入登录用例

	(2) 登录成功，返回主流程。
发生频率	随时，可能短期较集中

用例名称	CloseDeal
范围	系统用例
级别	用户目标
主要参与者	用户，商家
前置条件	取餐完成
后置条件	更新菜单信息，用户发布评价
主流程	<ol style="list-style-type: none"> 1. 顾客进入订单页面，单击确认订单结束。 2. 系统跳转到评价页面，顾客进行评价和打分并提交。 3. 系统将订单消息和评价发送给商家。 4. 商家进入订单页面，查看评价并确认交易结束。
扩展流程	<p>用户未确认交易结束： 在线下取餐结束后，顾客未及时确认交易结束。</p> <p>(1) 商家进入订单页面，发送提醒给用户。 (2) 系统将商家的提醒消息发送给用户。 (3) 用户接收消息，进入主流程第 1 步</p>
发生频率	随时，短期较集中发生

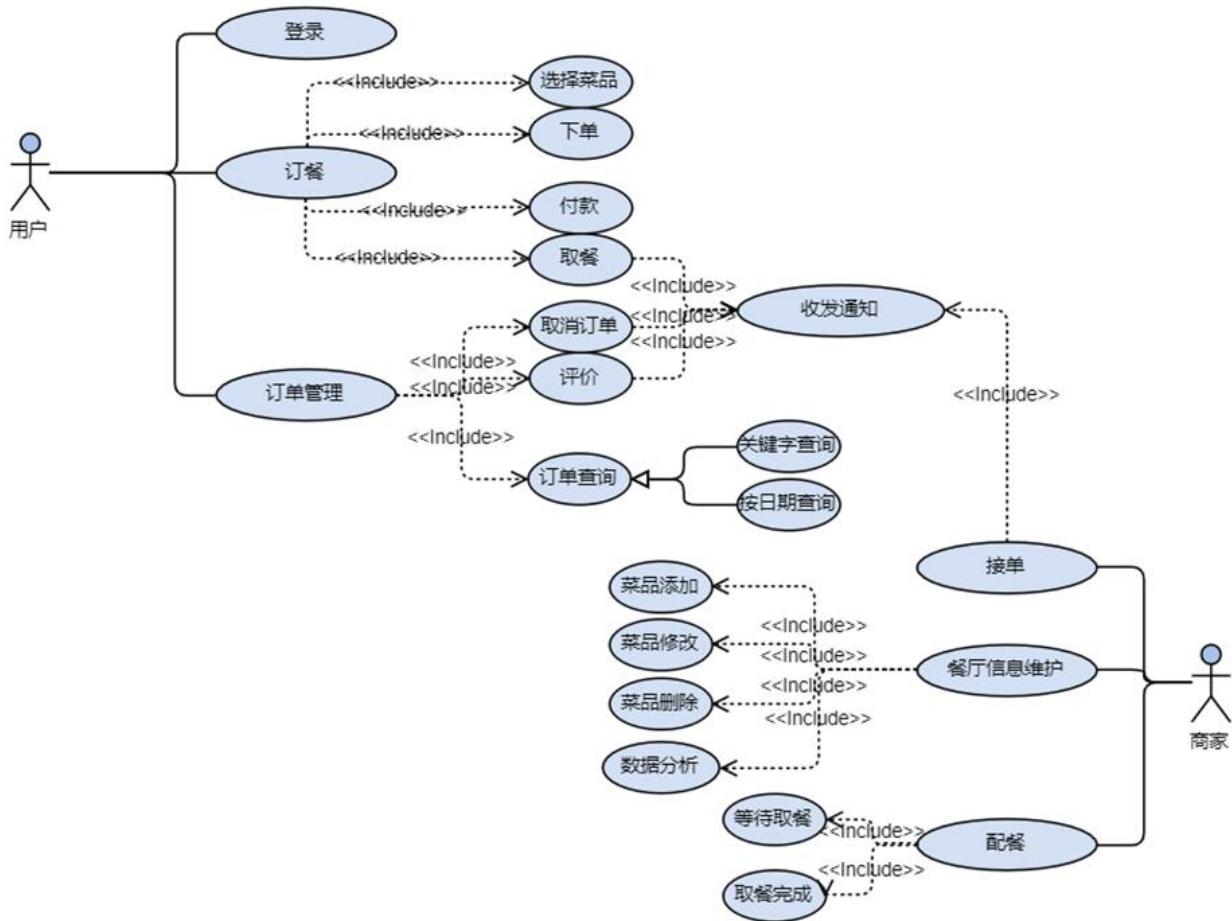


Figure 10 用例图

3.2 用户界面

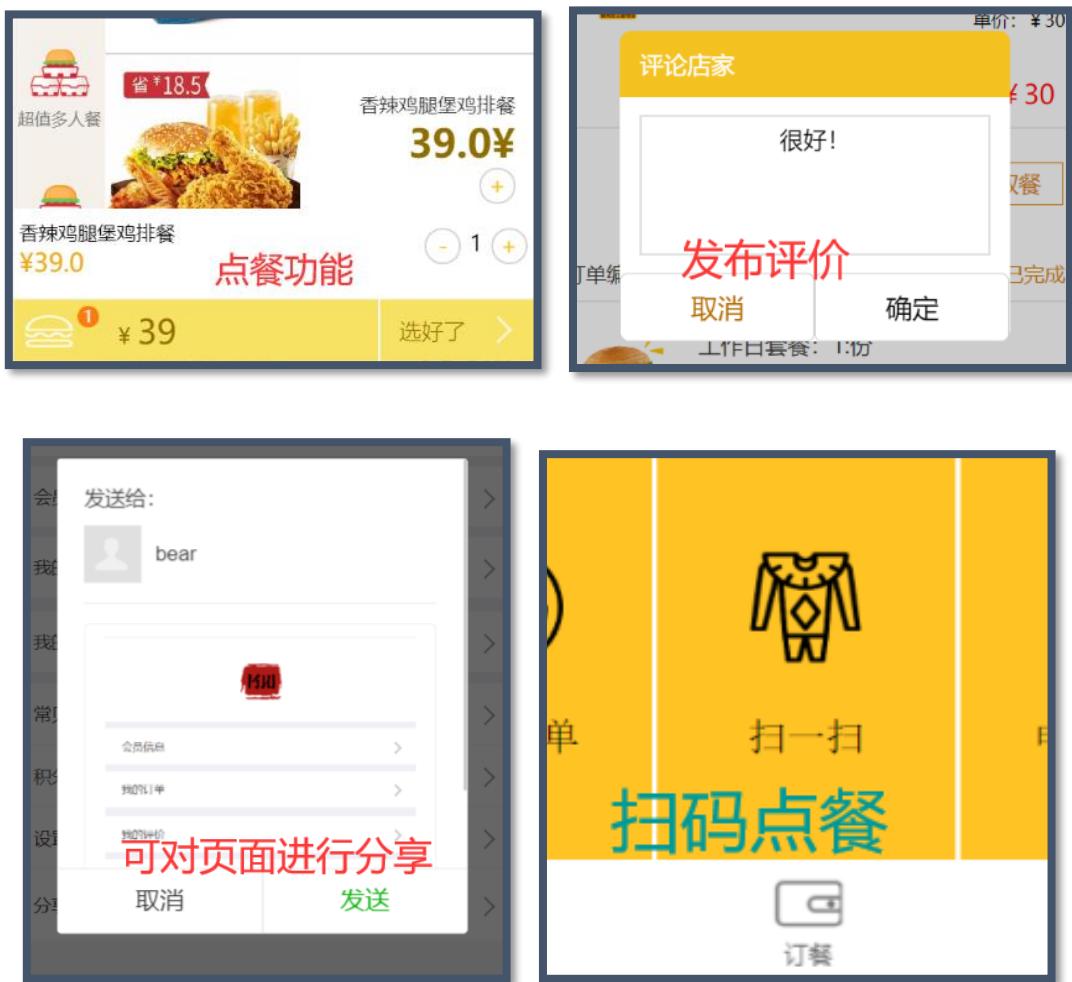
根据系统功能，用户界面应包含主页、登录界面、搜索结果显示界面、个人页、菜品页、用户对话界面和分类页等。具体页面设计如下：

- (1) 主页：在本页显示本 App/网页名称、制作者、显示登录和注册选项以及搜索选项，可链接至登录和注册界面以及搜索界面。
- (2) 登录页面：本页面用于登录，在本页提示用户输入 账户密码，能链接至注册页面。
- (3) 注册页面：在本页提示用户输入新的账户密码、邮箱及验证码等，可返回主页及登录界面。
- (4) 搜索结果显示页面：在本页顺序排列符合的搜索结果，对于每个结果显示图片、价格、卖家信息等，同时对用户提供收藏选项并记录用户信息，可返回主页。

- (5) 个人资料页面：用户在此页对自身账户进行设置，如修改头像、密码、基本页面设置等，能返回搜索页面或链接至个人商品页。
- (6) 历史记录页面：用户在此页能查看已收藏的菜品、购买记录，能返回搜索页面或链接至个人资料页。
- (7) 用户对话页面：用于用户间的沟通，可返回搜索结果页面，简略显示双方聊天记录，便于用户与卖家进行售后沟通。
- (8) 消息记录页面：用于显示用户的聊天记录，能进入对话页面或返回至上个页面。
- (9) 分类浏览页面：本页面按类别分类便于用户能快速准确地浏览并找到目标商品。
- (10) 点餐页面：显示某一菜品的价格、图片等信息，可进行购买、收藏等操作，可返回搜索页面、主页。

用户界面设计图如下：







参考文献

[1]概览 | 微信开放文档[EB/OL].

<https://developers.weixin.qq.com/miniprogram/dev/devtools/devtools.html>

[2] qq_36501609. 微信点餐需求分析[EB/OL].

https://blog.csdn.net/qq_36501609/article/details/89929225, 2019-05-07.

[3] 编程小石头. 点餐系统的开发[EB/OL].

https://blog.csdn.net/qiushi_1990/article/details/97749686, 2020-02-07.

[4] 重庆冰炫科技有限公司. 冰点云订餐食堂报餐点餐[EB/OL].

<http://www.cmerp.com/>, 2017.

[5] 嘻唰唰. 手机微信订餐报餐系统[EB/OL].<https://www.xeeyee.cn/>, 2017.

《软件综合实习》

系
统
分
析

(版本：2.0)

软件名称：智能食堂订餐系统

开发小组：胡周珏，崔令祎

2020 年 6 月 28 日

第一章 对象模型

1) 对象模型分为实体类、边界类和控制类。

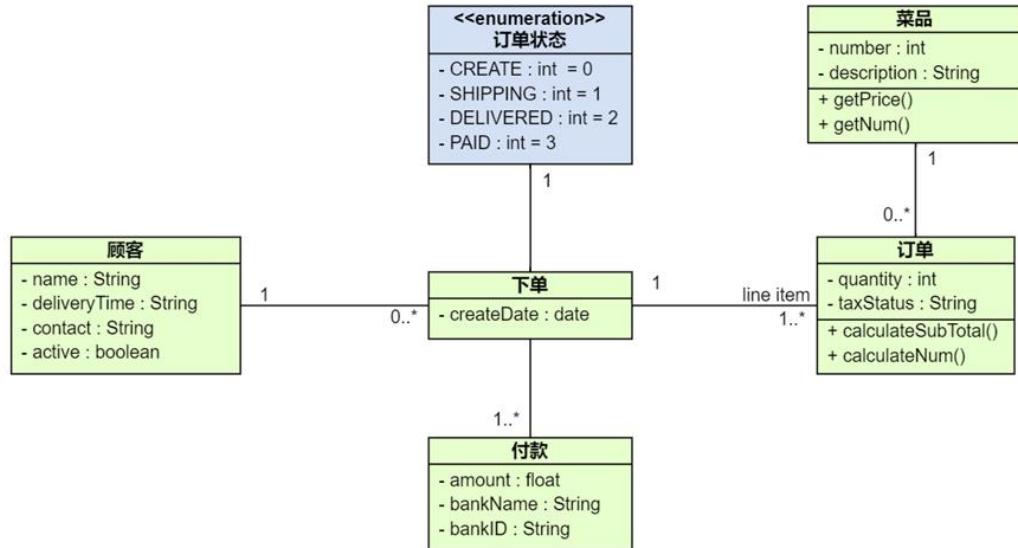
实体类名称	属性	关联类	定义
User	userID: int username: String password: String phonenumber: String e-mail: String	Order Dialog	用户信息类，保存用户名等信息
Item	itemID: int itemname: String itemdescription: String itemprice: double img_src: String add_time: date status: String	Order	菜品类，保存名称、简要描述等信息
Order	order_id: int ordertime: datetime place: String status: String	User Item	订单类，保存交易的各项信息
Dialog	content: String	User	对话类保存对话的消息记录

边界类名称	定义
LoginPage	登录页面
ItemListPage	点餐页面
HomePage	主页
PersonalInfoPage	用户信息页面
PersonalOrderPage	订单列表页面
CommentPage	评价页面
DialogPage	客服对话页面

控制类名称	定义
ItemControl	控制发布、搜索、浏览和收藏过程
PurchaseControl	订餐控制
UserInfoControl	控制用户注册、登录、查看并修改个人信息和查看个人订单的过程

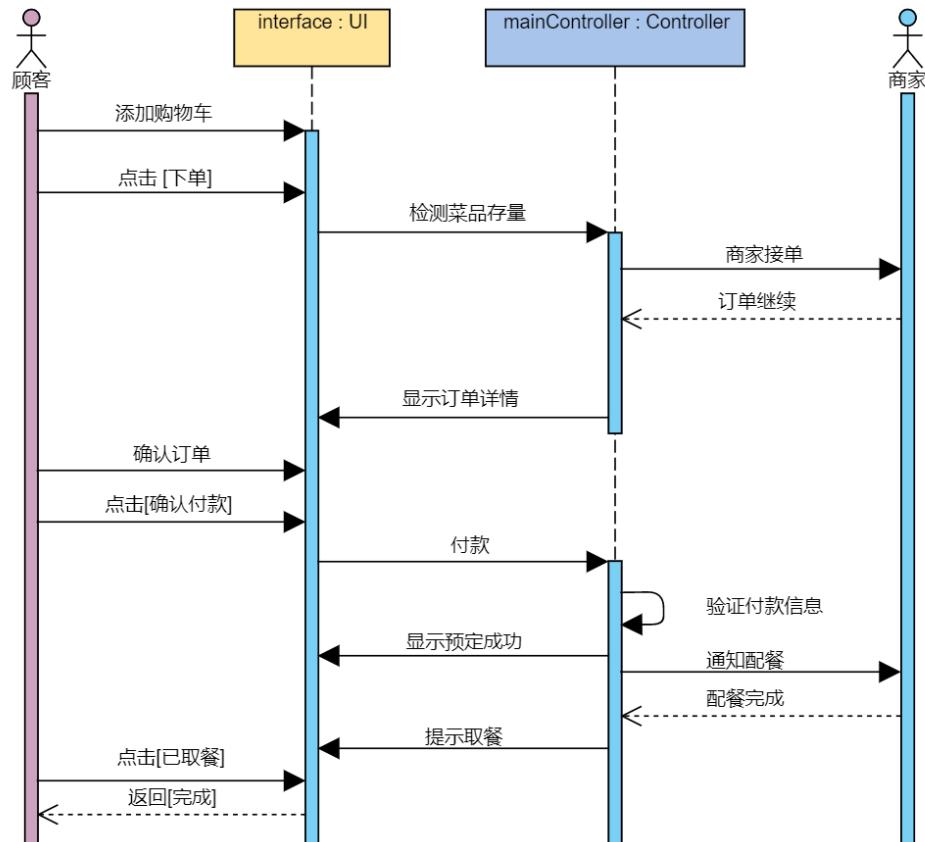
DialogControl	控制用户进行对话的过程
---------------	-------------

2) 类图



第二章 动态模型

1) 系统顺序图



2) 操作契约

操作	submitUserInfo (username: Username, password: Password, e-mail: Email, phonenum: Phonenum, userID: UserID)
交叉引用	Register
前置条件	用户进入注册界面
后置条件	创建 User 的实例 u; u. userID 赋值; u. userName 赋值为 username; u. password 赋值为 password; u. phonenum 赋值为 phonenum; u. e-mail 赋值为 e-mail。

操作	submitItemInfo(itemname: Itemname, itemprice: Itemprice, photo: Photo, itemID: ItemID, itemdescription: Itemdescription, itemtype: Itemtype)
交叉引用	PublishItem
前置条件	用户已登录
后置条件	创建了 Item 的实例 i; 创建 itemid, 为 i. itemID 赋值; i. itemName 赋值为 itemname; i. itemDescription 赋值为 itemdescription; i. itemPrice 赋值为 itemprice; i. add_time 赋值为当前时间; i. status 赋值为 onsale; 保存图片, 将图片地址赋给 i. img_src; i. itemType 关联 ItemType。

操作	submitOrder (buyerID: UserID, dealData: Date, dealTime: Time, dealPlace: String, price: Price, sellerID: UserID, itemID: ItemID)
交叉引用	PurchaseItem

前置条件	查看某一商品
后置条件	<p>归建了 Order 的实例 o;</p> <p>基于 buyerID 和 sellerID 的匹配，将 o 与 buyer 和 seller 关联；</p> <ul style="list-style-type: none"> o. buyer 赋值为 buyerID; o. item 赋值为 itemID; o. seller 赋值为 sellerID; o. date 赋值为 dealDate; o. time 赋值为 dealTime; o. place 赋值为 dealPlace; o. price 赋值为 price; o. status 赋值为 created。

操作	confirmOrder(o:Order,i:item)
交叉引用	Purchaseltem
前置条件	顾客已提交订单
后置条件	<ul style="list-style-type: none"> o. status 赋值为 confirmed; i. status 赋值为 bought; <p>基于 buyerID, 将 Order 与 OrderList 关联</p>

3) 状态图

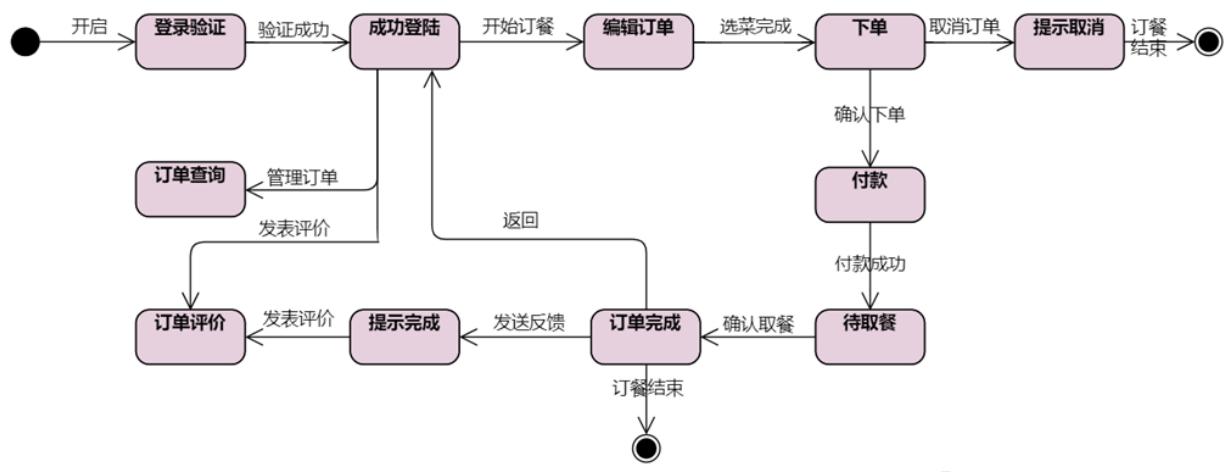


Figure 1 用户端状态图

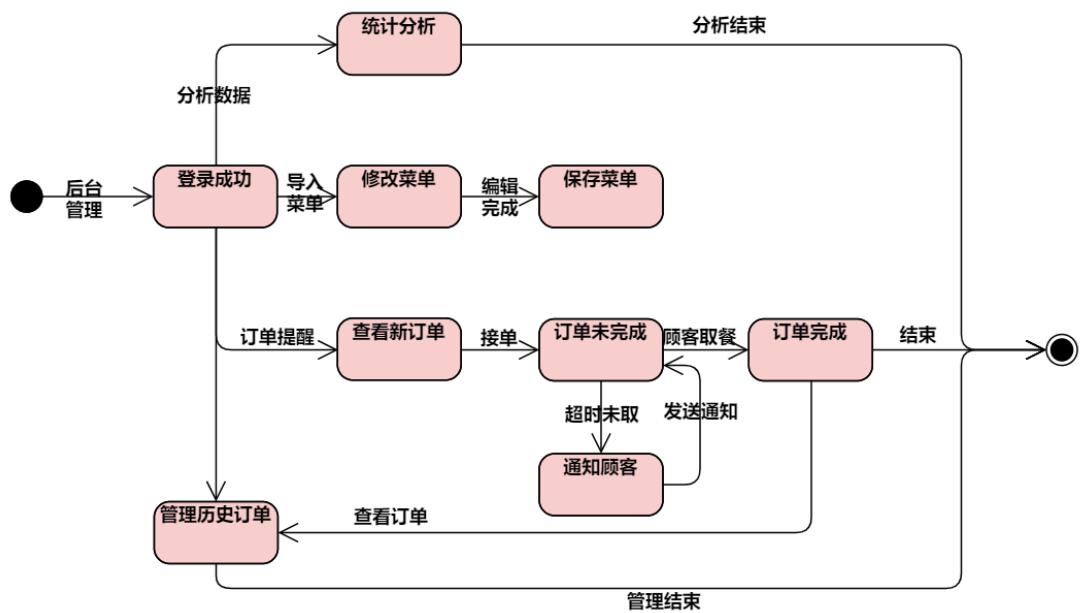


Figure 2 商家端状态图

《软件综合实习》

系统设计说明书

(版本：2.0)

软件名称：智能食堂订餐系统

开发小组：胡周珏，崔令祎

2020 年 6 月 28 日

摘要

针对就餐高峰期人员堵塞，传统人工点餐就餐处理流程效率低下等问题，设计实现了点餐系统，具有点餐客户端和点餐后台管理系统。

本文主要介绍了点餐系统的软件架构设计、软件设计模型、概要设计，详细设计以及技术难点，设计出了系统的大概原型，规划了后期开发计划。

概要设计对系统的总体框架、流程进行介绍，说明了系统中的接口设计和数据结构设计。详细设计则介绍了各个模块的功能，流程，用到的数据和测试计划。用户订餐有 APP 订餐、微信订餐、Web 端订餐三种方式，考虑到时间资源、人力资源有限，技术难点实现和技术原型选取微信订餐进行设计。

关键词：点餐； java； 小程序； 推荐算法

目录

摘要.....	2
1 软件架构设计	4
1.1 引言.....	4
1.2 目前体系架构	4
1.3 设计目标	5
1.4 数据管理设计	5
1.5 其他设计	6
2 软件设计模型	6
2.1 逻辑视图	6
3 概要设计	13
3.1 引言	13
3.2 总体设计	13
3.3 接口设计	19
3.4 运行设计	20
3.5 系统数据结构设计	21
3.6 系统出错处理	22
4 详细设计	23
4.1 小程序系统设计	23
4.2 商家管理系统设计	25
总结.....	30
参考文献.....	30

1 软件架构设计

1.1 引言

1) 编写目的

本软件架构文档的编写目的是对智能食堂订餐系统软件的系统结构进行描述和定义，在之前的需求规约、系统分析的基础上，利用各种模型，详细展示系统的结构，为后续的软件实现工作奠定基础。 文档将定义系统架构的设计目标，描述系统的结构、子系统定义、软硬件部署、数据管理、软件控制、边界条件等内容。本文档用于 开发团队明确系统的架构和设计，并以之为依据进行开发工作。

2) 适用范围

本文档适用于的软件：智能食堂订餐系统。

与该软件相关的特性、子系统、模型等均符合本文档中的内容。

3) 定义

本文档中涉及的术语定义在《项目词汇表》中给出。

4) 概述

本文档包括引言、目前系统架构、设计目标、软件的系统架构四部分。

目前系统部分对当前食堂订餐软件进行分析并指出其不足。系统架构设计目标部分结合软件需求，列举出系统设计的目标。软件系统架构给出系统的架构和子系统的分解，并以文字表述和模型图相结合的方式展示系统的数据管理等设计。

本文件的各部分内容联系紧密，互为补充和对照，共同呈现本系统的软件架构。

1.2 目前体系架构

目前存在的系统是一个线下的原始人工系统，即由商贩或者店家通过较低的价格收购旧书或者生活用品，自行整理分类后运至学校门口进行小范围短时间销售的这一流程所构成的系统。

新的系统是手机端的 App，快捷简便且便于用户上手。

1.3 设计目标

- (1)高可用性：本软件作为一个交易平台，如果在交易发生时发生系统不可用的情形，将会带来不可知的后果。因此系统需要保证较高的可用性。
- (2)安全性：本系统注册为实名制，数据库内保留有比较重要的个人信息，因此必须保证系统的安全性。
- (3)高性能：本系统运行过程中实时响应，并且具有较大的流量，因此要求有较高的性能。
- (4)可扩展性：本系统在初期规模较小时，不需要较为高规格的硬件。但应当考虑到在用户数增加后对系统的扩展

1.4 数据管理设计

选择关系数据库存储信息。

商家属性表

关系名称		shop	关系别名	商店信息			
属性名	别名	类型	长度	值域	唯一	可空	备注
sno	商家编号	INT		Y	N		
sname	商家名称	VARCHAR	50		Y	N	
saddress	商家地址	VARCHAR	50		Y	Y	

菜品关系属性表

关系名称		goods	关系别名	商品信息			
属性名	别名	类型	长度	值域	唯一	可空	备注
gno	菜品号	INT		Y	N		
gname	菜品名	VARCHAR	50		N	N	
gscale	规格	VARCHAR	50		N	Y	
gprice	价格	FLOAT			Y	Y	

顾客关系属性表

关系名称		employee	关系别名	员工信息			
属性名	别名	类型	长度	值域	唯一	可空	备注
eno	用户号	INT		Y	N		
ename	姓名	VARCHAR	50		N	N	
esex	性别	VARCHAR	2		N	Y	

销售关系属性表

关系名称		sale		关系别名		销售信息	
属性名	别名	类型	长度	值域	唯一	可空	备注
gno	菜品号	INT		Y	N		
Sno	商家号	INT		Y	N		
mount	数量	INT		N	Y		

订单关系属性表

关系名称		hire		关系别名		雇佣信息	
属性名	别名	类型	长度	值域	唯一	可空	备注
sno	商家号	INT		Y	N		
eno	用户号	INT		Y	N		
cost	价格	FLOAT		N	Y		

1.5 其他设计

1) 访问控制和安全设计

登录用户在使用他们的权限之前需要先提供用户名和登录密码，经检查是匹配的才通过认证。

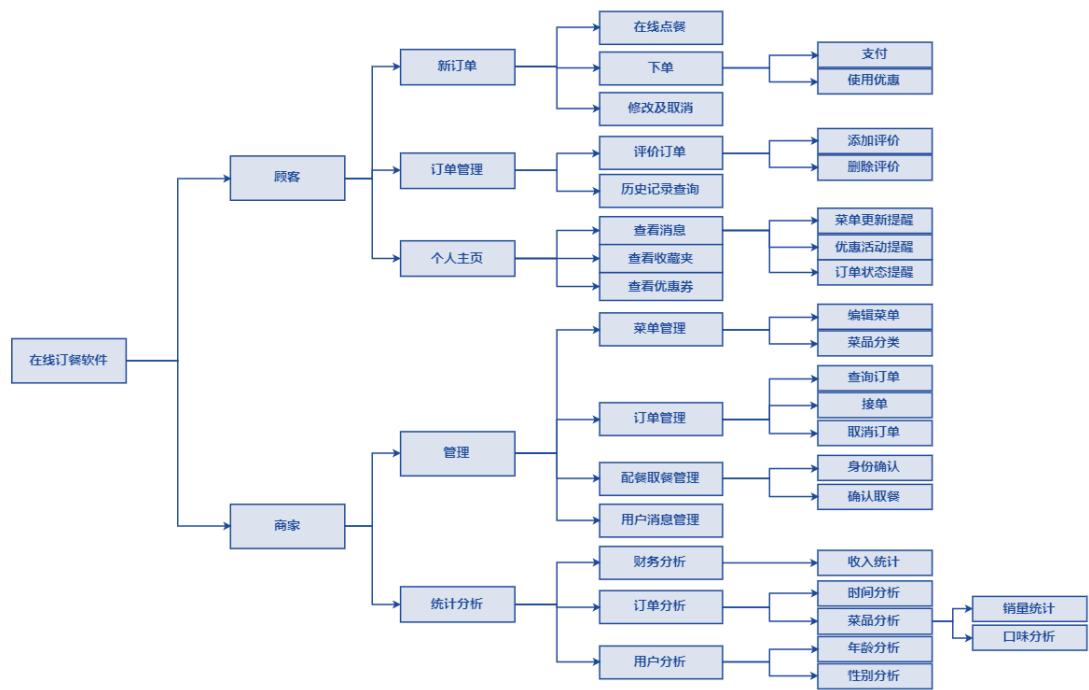
2) 可靠性设计。

为了提高系统的可靠性，我们考虑采取冗余与备份的方案。设置备用服务器在主服务器发生故障时将系统运行切换至备用服务器上；数据库冗余并备份，防止数据的丢失与意外更改。

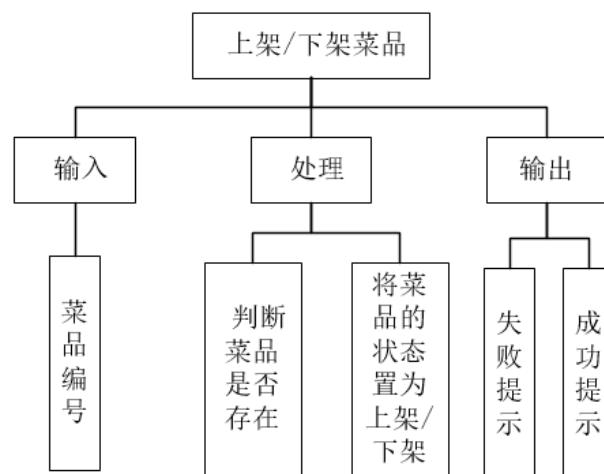
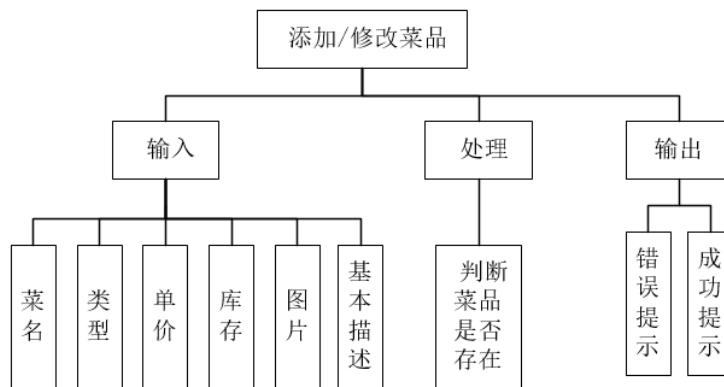
2 软件设计模型

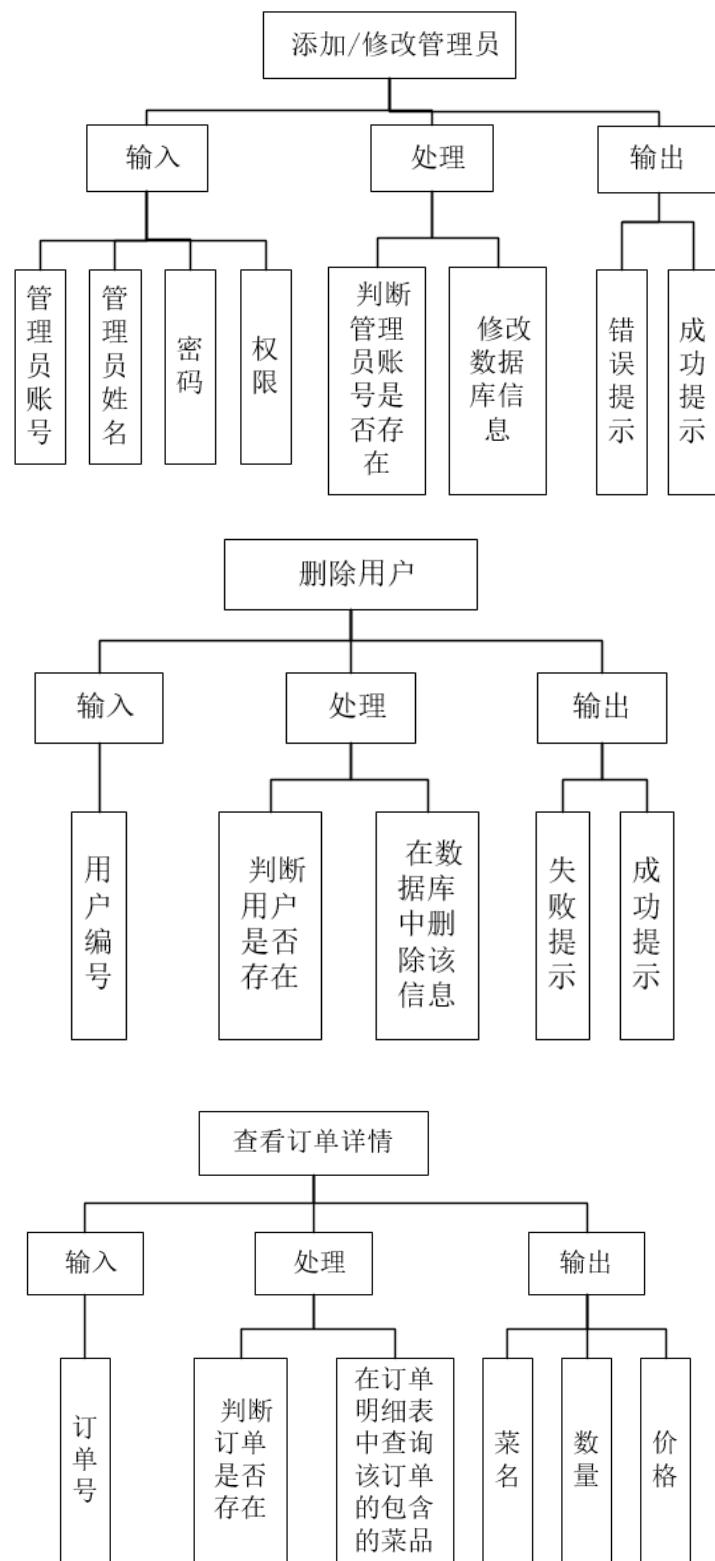
2.1 逻辑视图

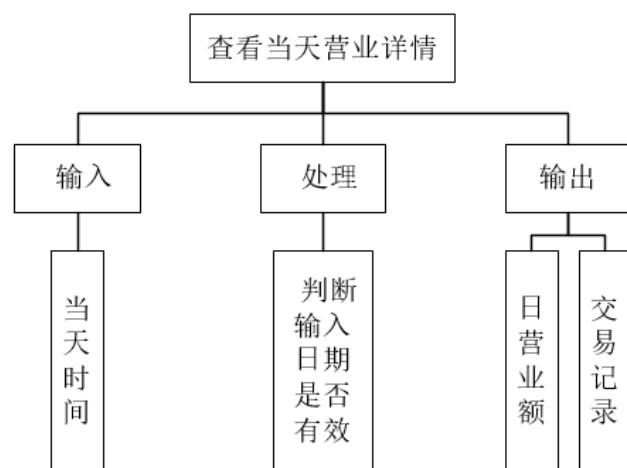
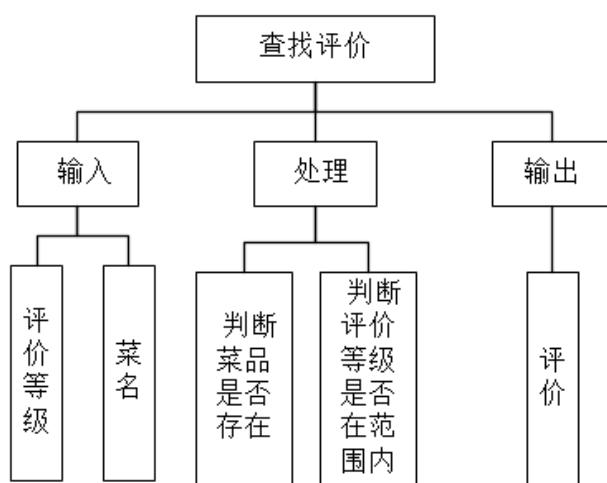
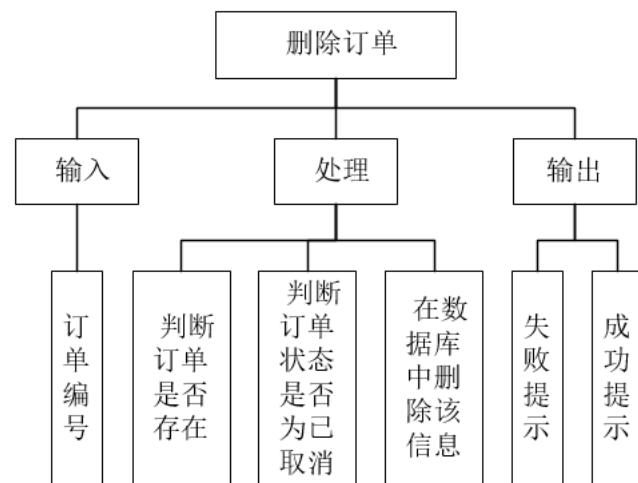
1) 系统架构

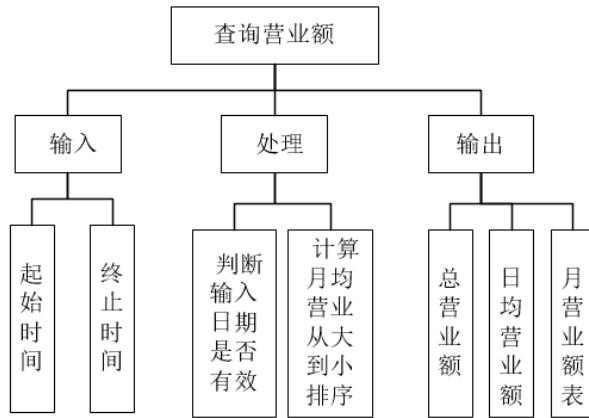


子系统的架构如下：









2) UseCase 实现

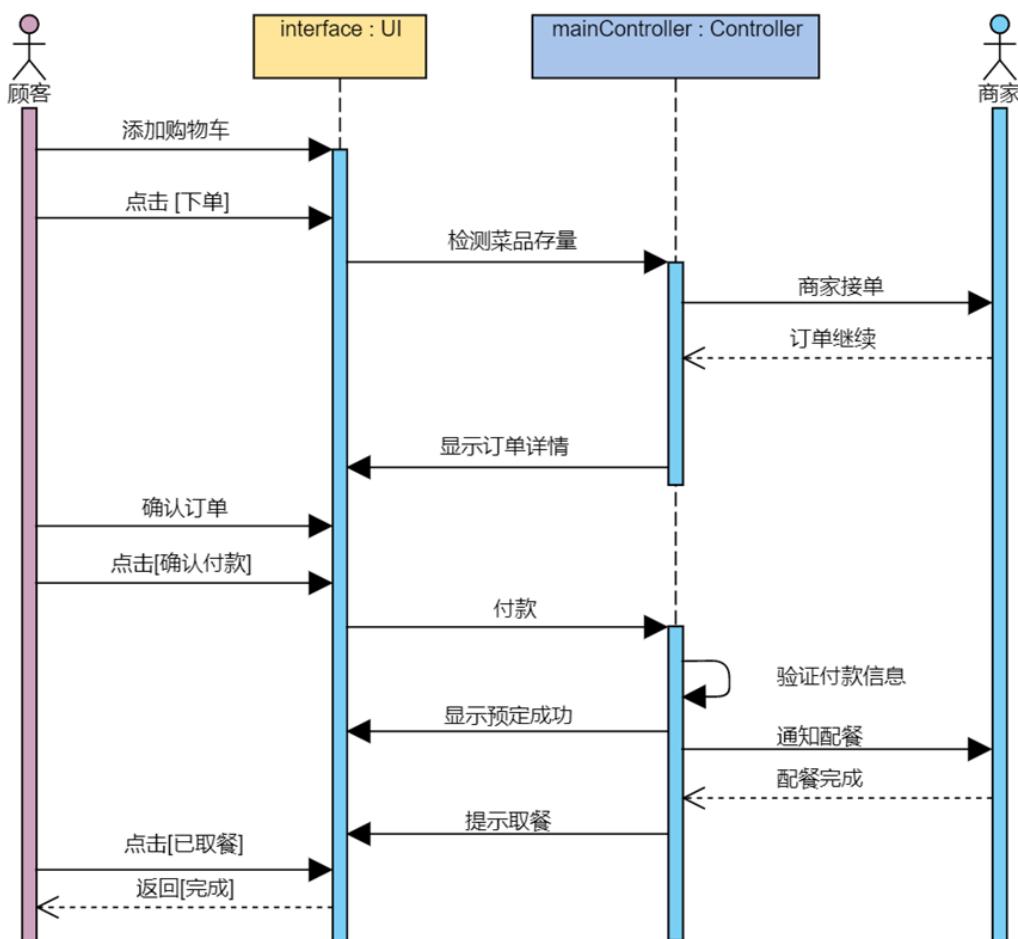


Figure 1 总用例图

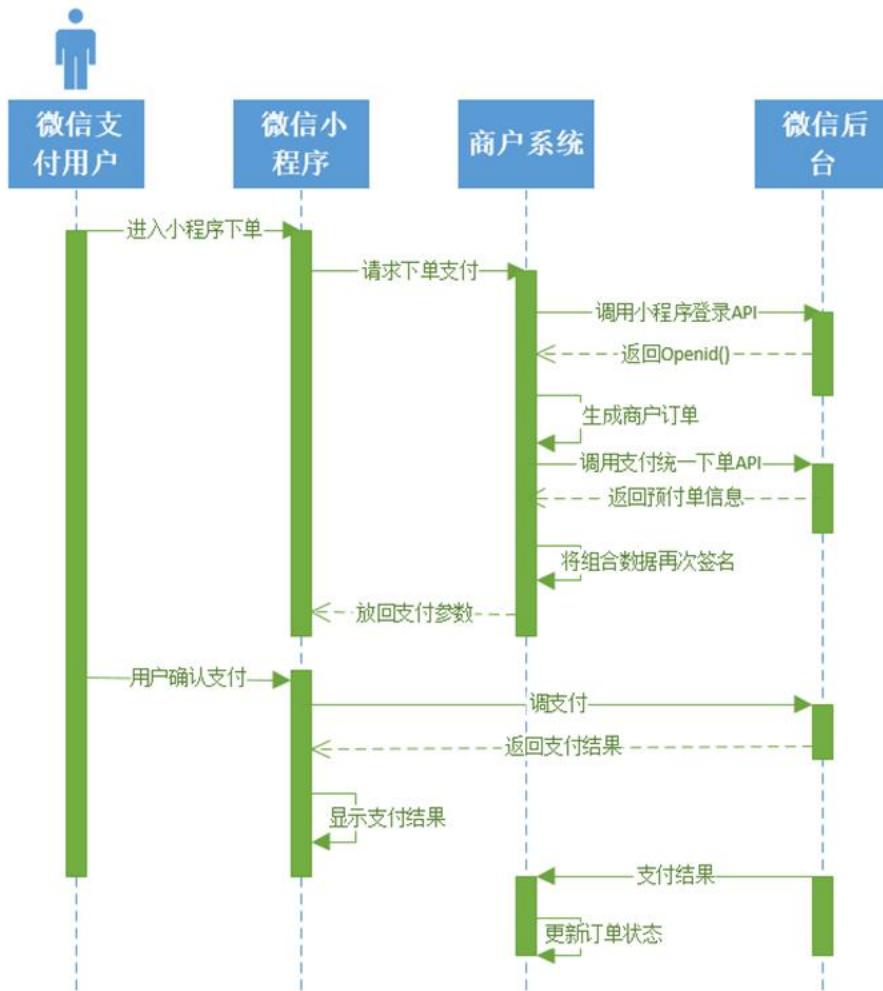


Figure 2 用户端用例图

3) 设计类图

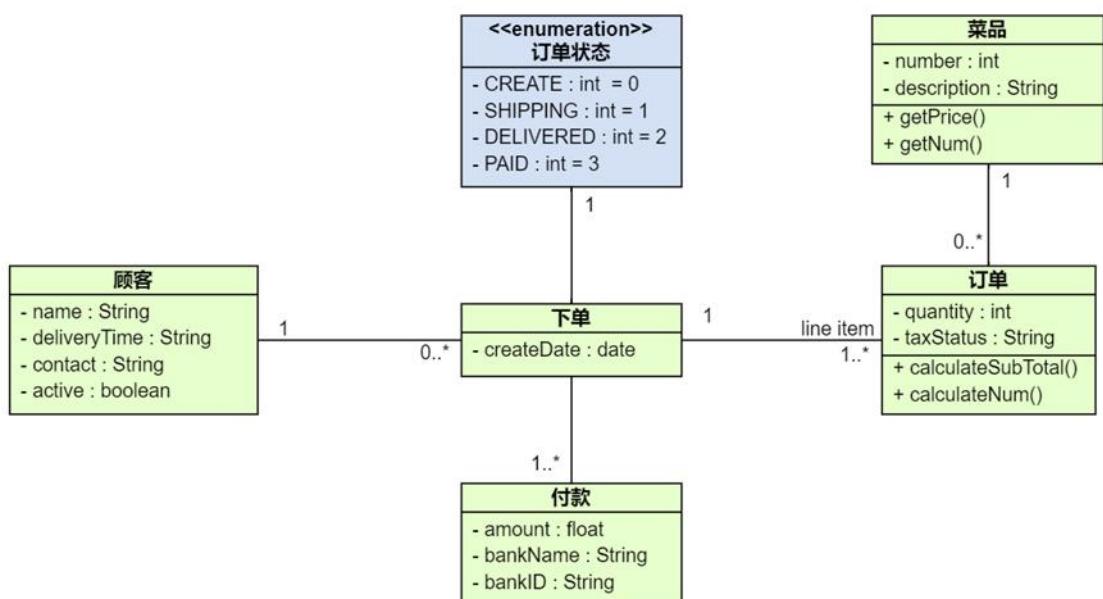


Figure 3 用户子系统类图

4) 其他图

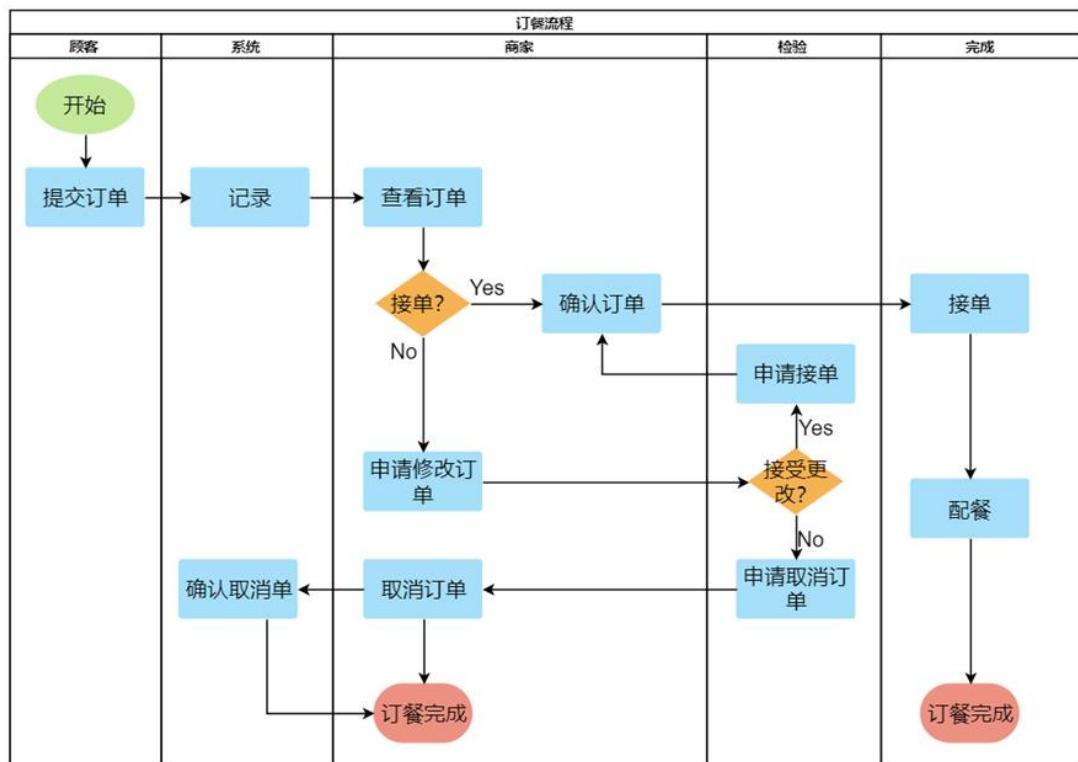


Figure 4 总流程

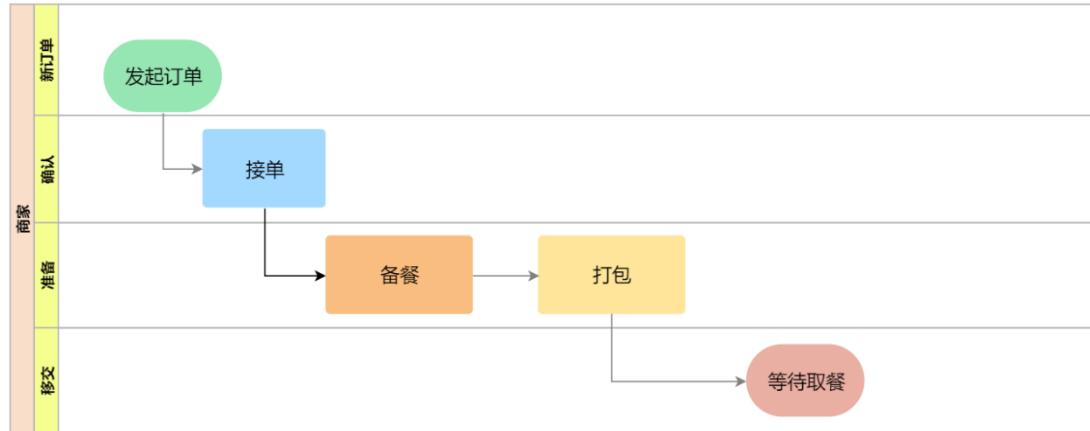


Figure 5 商家接单模块流程图

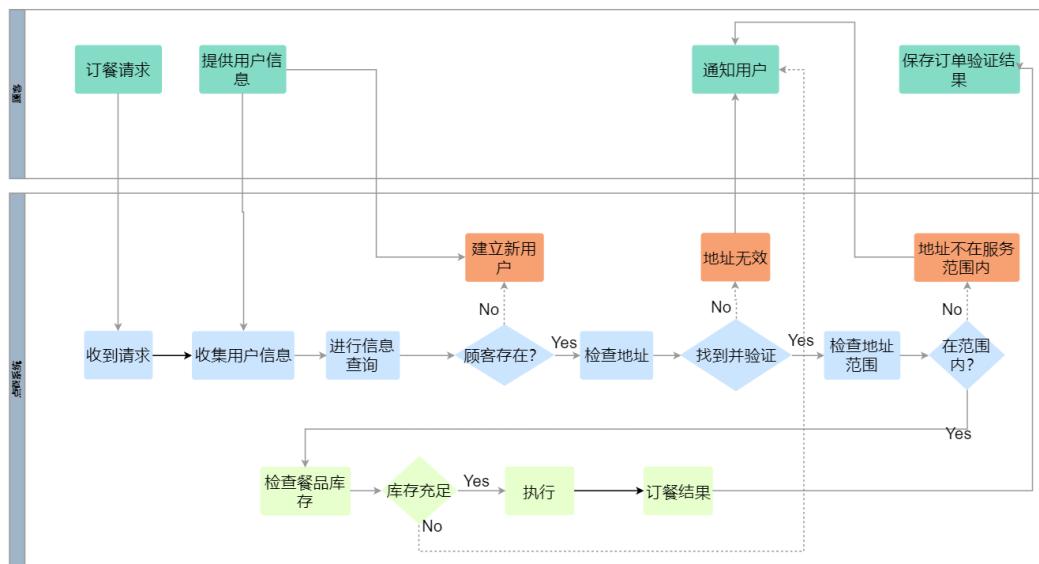


Figure 6 后台处理流程图

3 概要设计

3.1 引言

在线订餐系统提供的主要功能：1)普通用户能浏览系统、给系统留言并能注册成会员 2)会员能网上订餐、对商品进行评价 3)管理员能对商品进行增删改查管理 4)管理员能对各种商品进行销售统计 5)管理员能管理普通会员 6)管理员能对会员的订单进行管理 7)智能推荐功能。

3.2 总体设计

1) 需求规定

- 系统的输入项目

管理员信息填写

菜品信息填写

会员信息填写

- 系统的输出项目

订单信息

菜谱信息

营业额统计

用户评价信息

积分信息

- 系统处理功能

数据的录入与修改：管理员信息录入、菜品信息录入、会员信息修改

常规查询：订单信息查询、菜单信息查询、点菜单详情查询

报表生成：点菜信息表、菜单信息表、账单信息表、会员信息表、管理员信息表、

财务报表

一般分析：非授权用户不得进入程序系统，无修改权的用户不得修改档案。

2) 运行环境

硬件环境：各类 android 手机，IOS 手机，平板

软件环境：android2.2 及以上版本，IOS7.0 及以上版本

3) 基本设计概念和处理流程

本系统设计的目标是实现一个功能较为强大的网络订餐系统，该系统能够实现用户注册、商品浏览、商品选购、用户订单查询等各种功能；除此之外，在系统的后台，管理员能够自主处理用户订单，并实现对商品信息的集中管理。从需求分析中得知，可以将功能分为如下功能模块：

- 用户管理：注册食堂商家维护人员和订餐会员帐号
- 菜品管理：商家随时更新自己所经营的菜品种，以供订餐客户进行查询
- 订单管理：订餐客户进系统，查看自己要订的菜品，下订单。可查询和修改订单
- 配餐管理：该功能是食堂管理员对订餐客户的订单进行审核确认，没问题直接确认
- 通知管理：供商家发布公告消息，这样登录用户就可以在线第一时间看到公告信息，改变传统黑板报信息传递模式提高效率
- 评价管理：订餐客户订餐后可对用餐体验进行评分，这样一方面是维护消费者的权益，另一方面是为了督促食堂进而将强了餐厅之间的服务竞争同时提高交易的信誉度
- 统计报表分析：包括对所有的查询信息都能以 excle 形式将数据导出设计。

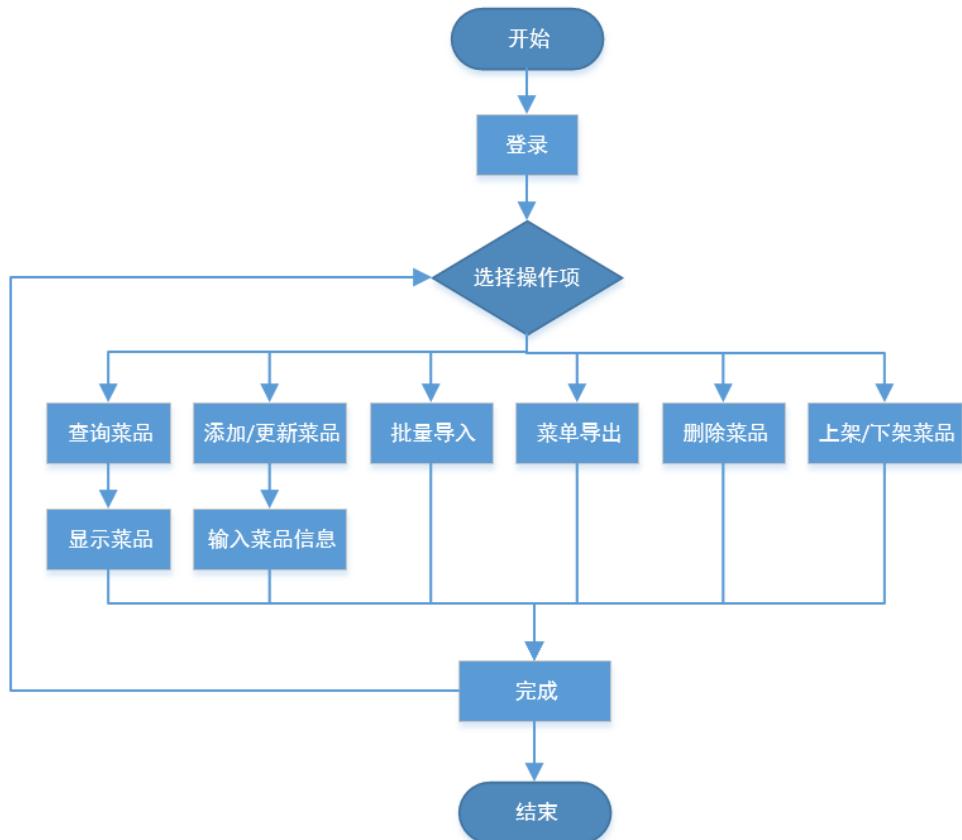


Figure 7 菜品管理流程图

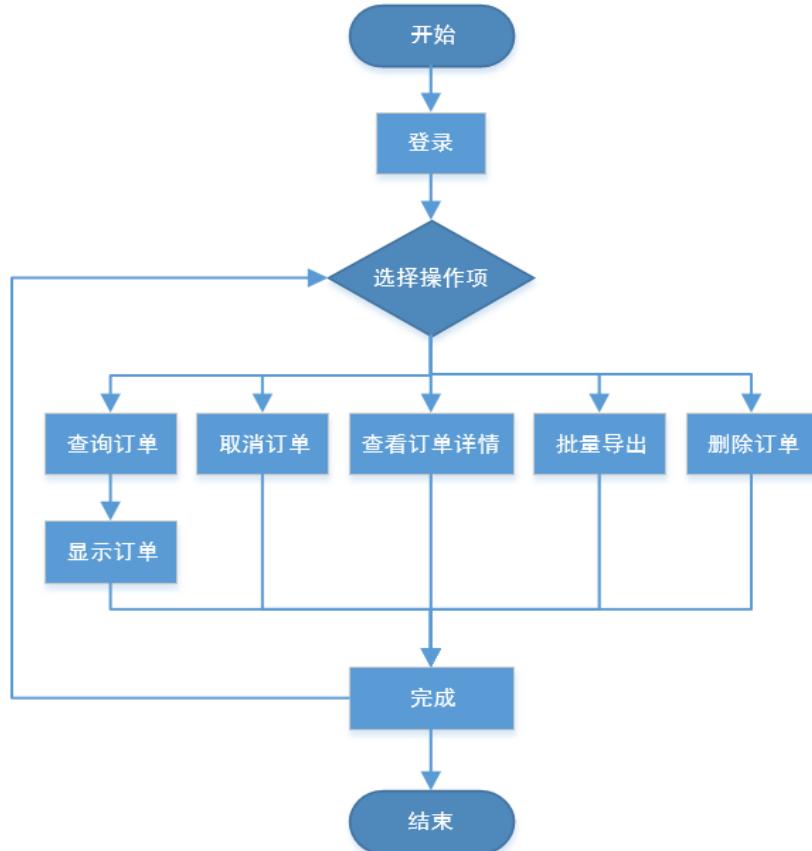


Figure 8 订单管理流程图

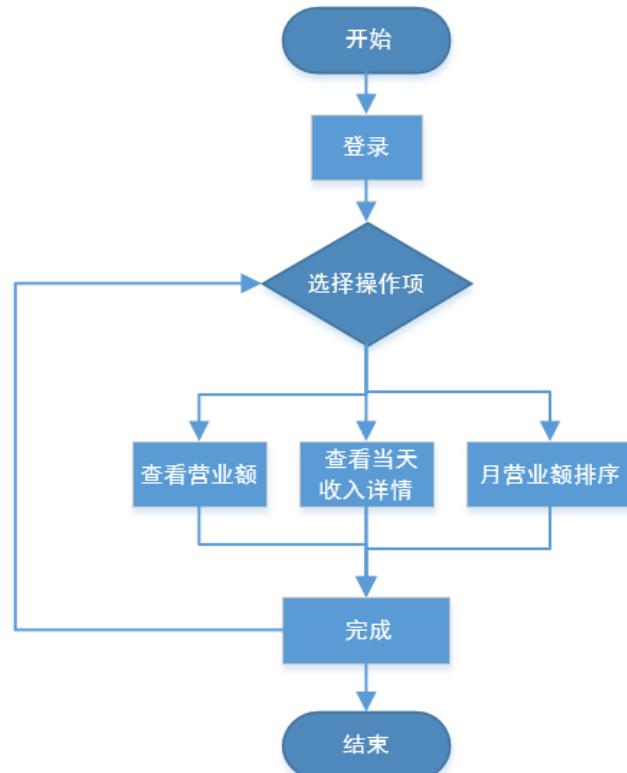


Figure 9 财务分析流程图

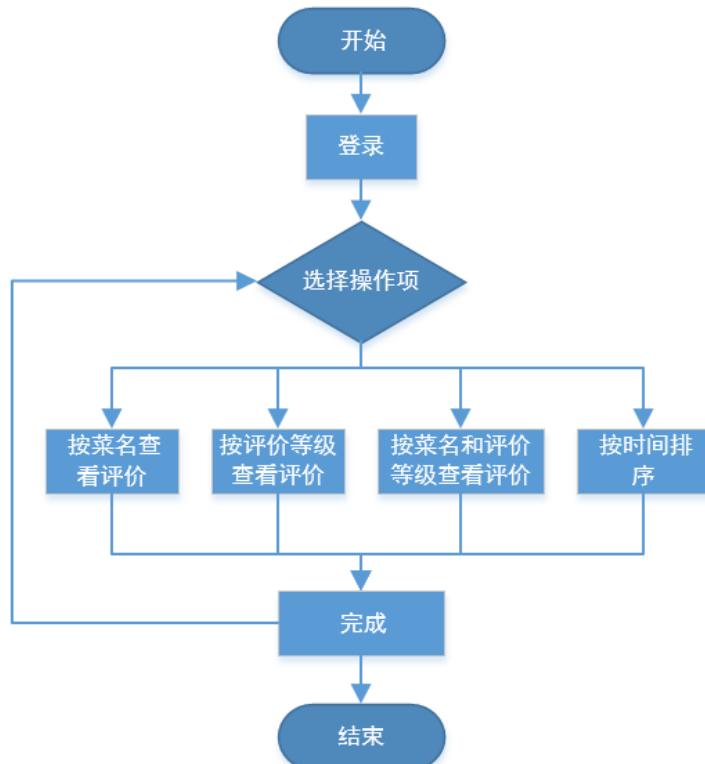


Figure 10 评价管理流程图

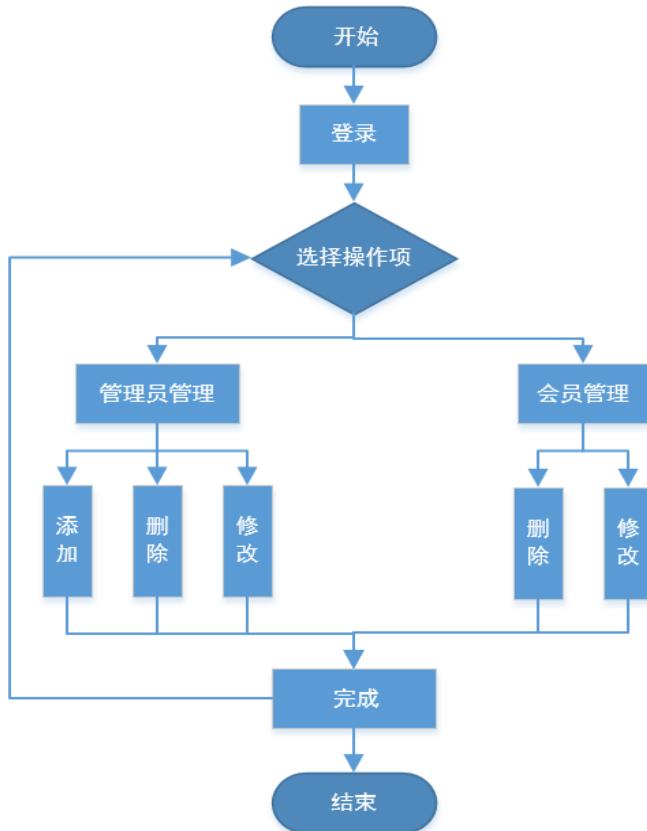


Figure 11 用户管理流程图

4) 结构

- 系统逻辑结构

就 Web 站点的逻辑结构而言，大致分为三个层次：浏览器—Web 服务器—数据库服务器。三种层次相互独立、相互依赖、密不可分。具体来说，用户端可以借助于 URL 与 Web 服务器建立网络连接，并就 Web 服务器上的信息进行浏览、下载等操作；应用逻辑层则主要通过 JDBC 和数据库服务器进行数据交换。

为确保该系统的安全性及网络访问性能，可根据用户类型将应用逻辑细分为两个层次：一层是针对未经注册的用户，这部分用户只有查看部分页面及功能的权限；另一层则主要针对已经注册过的会员和系统管理员，这部分用户除了具有查看部分页面及功能的权限之外，还能够进行后端数据库操作。

5) 功能需求与程序的关系

后台管理系统							
	验证登录程序	添加信息程序	修改信息程序	删除信息程序	查询信息程序	打印信息程序	排序程序
菜单管理	√	√	√	√	√	√	
用户管理	√	√	√	√	√	√	
订单管理	√			√	√	√	√
评价管理	√				√		√
财务分析	√				√	√	√

客户订餐系统					
	菜单模块	数据统计模块	查询模块	数据库操作模块	增加删除模块
浏览功能	√			√	
点餐功能	√			√	√
查询功能		√	√		

6) 人工处理过程

- 需要人工的从键盘或者手机屏幕进行相应的操作，录入数据
- 客户下单后，需要厨房将菜制作好，服务员为顾客上菜
- 后台需要人工管理

7) 尚未解决的问题

在智能订餐部分，需要运用大数据、机器学习中的协同过滤推荐等算法。app 和小程序不能运行推荐算法，推荐算法应该在后端服务器上实现，用 java, python 或者 nodejs 实现，然后提供 restful api 给 app 和小程序调用，app 和小程序只做推荐的界面。

一类是协同过滤，协同过滤是一大类方法，下面有很多具体的不同算法，有基于

深度神经网路的，有传统的，`python` 资源很多，`nodejs` 则比较少。另一类是基于内容过滤，也有很多具体的方法，如基于知识图谱的推荐算法等，`python` 资源丰富。

3.3 接口设计

1) 用户接口

用户通过界面接口实现参数的输入，进入相应的界面后输入提示的信息即可产生相应的任务。

对话框	提供的命令
主页	订餐，智能推荐，一键下单，系统操作
点餐	商家筛选，商家选择，确认，退出
查询	类别选择，语音搜索，拍照搜索，文字搜索，查询，退出
菜单操作	口味选择，价格选择，数量选择，添加，删除，提交，退出
订单操作	下单，付款，取消订单，退出
数据分析	统计条件单选，输出设置，关键字编辑，确认，返回
系统设置	菜单系统操作，数据统计，订单管理，更改口令，返回

2) 外部接口

软件接口：MySQL 数据库、浏览器、服务器

硬件接口：鼠标、键盘

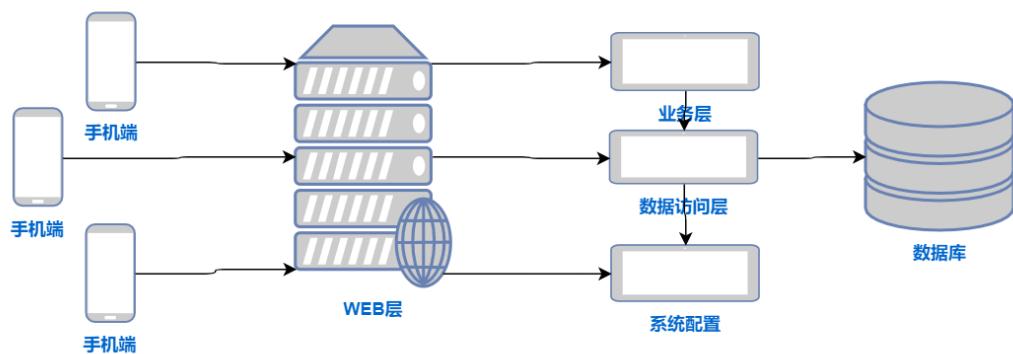


Figure 8 模型结构

3) 内部接口

各模块间的数据联系主要通过访问数据库接口。对公用函数、用户对象和窗口提供调用接口说明。接口传递的信息将是以数据结构封装的数据，以参数传递或返回值的形式在各模块间传输。具体参数的结构将在详细设计中说明。

顶层模块	二级模块	接口
数据	底层数据操作	取记录, 执行 SQL 语句
	用户操作	对应数据库表操作
界面	用户界面	生成任务
	缓冲区	生成批处理

3.4 运行设计

1) 运行模块组合

- 登陆模块：程序启动后,进入登陆模块,工作人员输入帐号密码点击登陆系统调用密码校验模块,校验无误后进入菜单主模块。
- 菜单主模块：用户选择桌号和菜肴,单击对话框按钮事件触发主模块调用各下层模块,生成对应的该顾客的菜单,存入菜单数据库,屏幕上显示出顾客最终的点菜菜单。
- 修改模块：如顾客需要修改菜单则根据需要单击添加、删除按钮,系统自动进入修改模块进行菜单修改。返回菜单主模块。
- 结账模块：顾客单击结账按钮,调用结账子模块,弹出对应的子对话框显示该顾客的菜单信息及消费总金额。

2) 运行控制

外界操作符合要求就可根据用户的输入调用各子模块,若超出规定范围则会出现对话框提示用户信息错误。

3) 运行时间

各操作的时间响应的上限应控制在 1s 内。

3.5 系统数据结构设计

1) 菜单:	菜品编号(foodId)	整型
	菜品名(foodName)	字符串
	菜品类型编号(typeId)	整型
	单价(foodPrice)	浮点型
	库存(foodStock)	整型
	菜品图片(foodIcon)	字符串
	基本描述(foodDescrib)	字符串
	创建时间(createTime)	Date 类型
	状态(foodStatus)	整型
2) 菜品类型:	类型编号(typeId)	整型
	类型名称(typeName)	字符串
3) 管理员:	管理员账号(adminId)	整型
	管理员姓名(adminName)	字符串
	密码(password)	字符串
	权限(adminType)	整型
	创建时间(createTime)	Date 类型
4) 用户:	用户编号(userId)	整型
	用户名(userName)	字符串
	用户电话(phone)	字符串
	积分(score)	整型
	会员等级(level)	整型
5) 订单:	订单编号(orderId)	整型
	用户名(userName)	字符串
	用户电话(phone)	字符串
	总价(orderAmount)	浮点型
	订单状态(orderstatus)	整型
	下单时间(createTime)	Date 类型

	付款时间(payTime)	Date 类型
6) 订单明细:	编号(detailId)	整型
	订单编号(orderId)	整型
	菜品编号(foodId)	整型
	菜品名 (foodName)	字符串
	数量(quantity)	整型
	价格(foodPrice)	浮点型
7) 评价:	评价编号(commentId)	整型
	评价等级(commentLevel)	整型
	评价内容(content)	字符串
	用户编号(userId)	整型
	用户名(username)	字符串

3.6 系统出错处理

1) 出错信息

错误	系统输出信息的形式	含意	处理方法
下单出错	弹出提示框	商品不存在	重新选择
输出错误	弹出提示框	输出不规范	检查并重试
查询错误	查询结果界面显示	信息不存在	发送提示
登录错误	登录界面显示错误信息	用户不存在; 密码或用户名错误	重新输入
支付错误	跳转到提示页面	余额不足或密码错误	退出支付
网络连接错误	弹出提示框	网络故障	尝试重新连接
数据库连接错误	弹出提示框	数据库设置错误	检查数据库
其他	未知错误	用户操作不当、系统出错	程序中止

2) 补救措施

- 由于数据在数据库中已经有备份，在系统出错、原始系统数据丢失后可以依靠数据库的恢复功能，并且依靠日志文件使系统再启动，确保用户数据不会丢失或遭到破坏。但这么做可能占用更多数据存储空间。
- 利用恢复再启动技术使软件从故障点恢复执行或从头开始重新运行。死机重启该软件后，可以恢复到前一个故障点。

3) 系统维护设计

- 数据库的维护

SQL Server 2008 及以上版本提供数据库的备份和恢复的功能，可以方便的实现数据库的维护管理。

- 软件功能方面的维护

由于采用模块化的设计方法，每个模块(窗口)间独立性较高，便于软件的维护，对于个别功能的修改只需修改一个窗口。

4 详细设计

4.1 小程序系统设计

1) 程序描述

- 目的意义

为用户提供便捷的订餐服务

- 程序特点

非常驻内存；是子程序；是不可重入程序；有覆盖要求；是并发处理。

2) 功能

客户端子系统包括用户登录、个人信息维护、餐品信息浏览、维护购物车、生成订单等功能。

3) 性能

- 用户进行某一项操作的响应时间应在一定可接受的范围内，太慢会影响使用，给用户造成不便。
- 当新增一个预约或其它操作时应能在主页及时显示，不能有太多的延时。

- 采用观察者模式能及时更新信息，随着日期的改变所有的信息也应该跟着的变化。
- 由于要读写数据库，因此对电脑的配置也有一定的要求。

4) 输入项

名称	标识	数据类型	数据来源	说明
用户输入	Input	string	用户接口	输入文字
菜单关键字	Keyword	string	用户接口	菜单内容
订单号	No	String	系统生成	每个订单唯一对应
点餐信息	Food	Order	用户接口	点餐结构化数据
要删除的信息	Delete	Int	用户接口	删除内容编号
退出命令	Quit	Boolean	用户接口	是否退出
用户设置	Setting	Boolean	用户接口	设置项状态
数据统计命令	analysis	Boolean	用户接口	命令为 true 则进行统计

5) 输出项

名称	标识	数据类型	输出形式
用户界面	Text	String	图形界面
搜索结果	SearchRes	结构化数据	列表
订单信息	Order	结构化数据	列表
点餐结果	Reservation	Boolean	订餐是否成功
删除结果	Deleted	Boolean	删除是否成功
相对对话框	Dialog	String	文字信息
统计结果	Static	结构化数据	列表

6) 接口

- 软件开发 API
小程序逻辑层提供丰富的微信原生 API，可以方便的调用，进行获取用户信息，本地存储，支付功能等功能的设计。
- 参数赋值和调用方式
参数由交互界面赋值，在小程序视图层由函数进行调用，进行模块间传值。
- 关联的数据结构
结构化存储的菜单信息文件、系统配置文件、临时存放的数据。

7) 注释设计

- 加在模块首部的注释：作者、版本信息
- 加在各分枝点处的注释：类作用及相关信息
- 对各变量：功能、范围、缺省条件等
- 对使用的逻辑所加的注释：注明构造函数的作用；注明方法参数、返回值、异常

4.2 商家管理系统设计

1) 菜单管理模块

- 程序描述

该模块程序主要实现商家在后台添加、删除、修改菜单，控制菜品的上架和下架，对菜单实现批量处理。

- 功能

- a) 添加或修改菜品
- b) 上架或下架菜品

- 性能

运行时间响应<1s.

- 输入和输出项

输入：菜名，菜品类型，单价，库存，菜品图片，菜品基本描述

输出：菜单

- 限制条件

导入菜单时将 excel 的数据导入到数据库，导出菜单时也导出到 excel。

- 测试计划

测试模块	测试功能	测试指标
菜单管理	添加菜品	能够将新的菜品信息录入到菜单中并显示出来
	删除菜品	删除掉已有的菜品
	上架/下架菜品	改变菜品的状态，上架的菜品客户才能加入购物车，下架菜品并不删除菜品的信息，只是客户无法选取

	导入菜单	将 excel 中的菜单导入数据库
	导出菜单	将数据库中的菜单导入到 excel 并保存到指定文件夹
	修改菜品	修改已有菜品的信息，并显示

2) 用户管理模块

- 程序描述

该程序主要对会员和管理员进行管理。对会员进行信息查看。添加、修改、删除管理员，给管理员不同的权限。

- 功能

- a) 添加和修改用户

对于管理员信息可以进行添加或修改，会员信息在用户端由用户输入注册，后台可以修改用户信息。管理员具有两种权限，一种是普通管理员，一种是超级管理员，超级管理员可以对用户信息进行管理而普通管理员不行。

- b) 删除用户：在后台可对管理员或者会员信息进行删除

- 输入输出项

输入：管理员账号、姓名、密码、权限；会员姓名、电话、积分、会员等级

输出：执行成功和失败提示。

- 限制条件

当登录后台的账户是超级管理员账户时才可以进行用户管理，普通管理员账户没有这个权限。

- 测试计划

测试模块	测试功能	测试指标
用户管理	添加管理员	能够将新的管理员信息录入到数据库，并且新的管理员账户可以成功登陆后台进行权限内的操作
	修改用户信息	修改已有用户的信息并显示
	删除用户	删除已有的用户
	管理员权限	测试普通管理员和超级管理员的权限是否正确，修改管理员权限后权限是否发生变化

3) 订单管理模块

- 程序描述

该程序主要对订单进行管理，可以实现查看订单、取消订单、删除订单和导出订单等操作。为方便管理员查看订单，将订单信息可视化，按照下单的时间将订单在表格中显示，新订单排在表首。在收到新订单后管理员可以查看订单详情来获取订单中所点的菜和需求。管理员可以对订单进行取消和删除，订单信息也可以从数据库批量导入到 excel 中。

- 功能

- a) 查看订单详情

管理员点击查看订单详情，跳转到订单详情界面，显示订单包含的菜品、数量及其价格。

- b) 取消订单

点击取消按钮，若该订单存在，改变订单在数据库的状态为已取消。取消后的订单为无用订单，可进行删除。

- c) 删除订单

对数据库中该订单的信息进行删除，并且只能删除已经取消了的无用订单。

- d) 导出订单：将订单导出到 excel 中。

- 输入输出项

输入：订单编号

输出：订单详情（菜品，数目，价格），操作成功失败提示。

- 流程逻辑

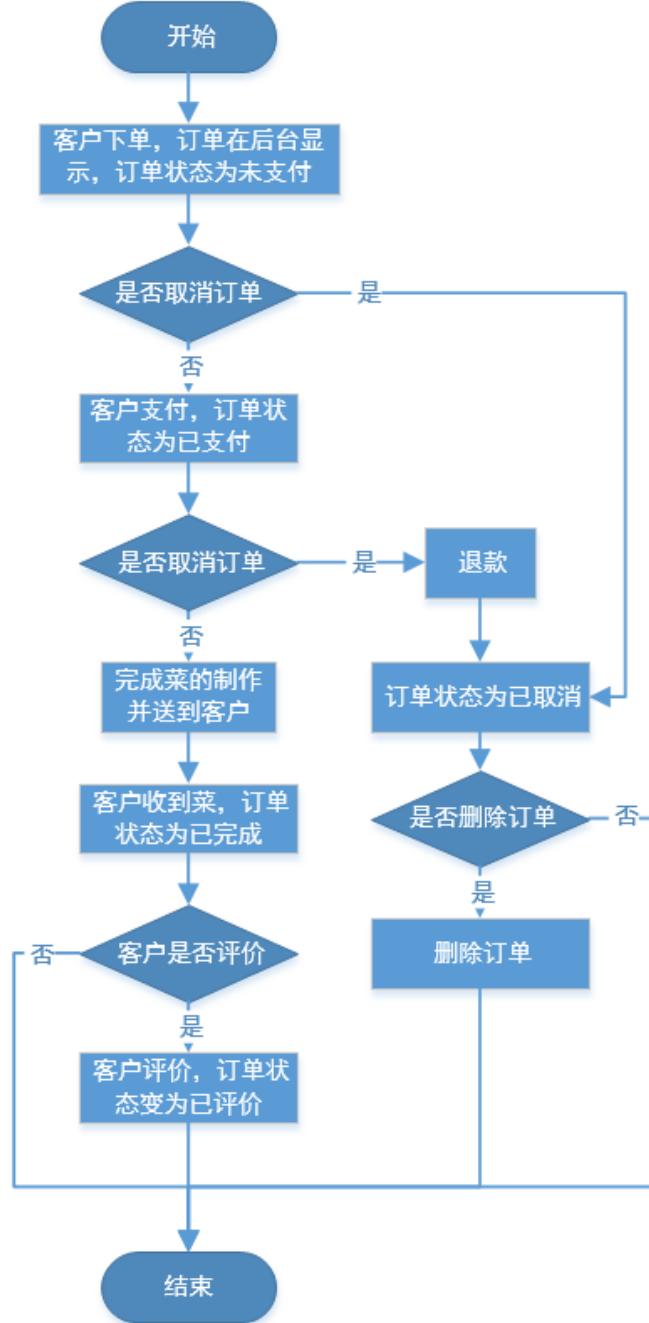


Figure 11 订单流程图

- 测试计划

测试模块	测试功能	测试指标
订单管理	查看订单详情	能够查看订单具体点的哪些菜品，每道菜的数量和价格
	删除订单	删除已有的已取消订单
	导出订单	将数据库中的订单导入到 excel 并保存到指定文件夹

	取消订单	取消订单，订单只有在为完成的状态下才能取消，若取消前未支付则直接取消，取消前是已支付则退款后取消
	订单状态	测试订单的状态变化是否正确，各个阶段的订单状态是否符合要求

4) 评价管理模块

- 程序描述

该程序主要实现评价管理，为了方便管理员查看客户评价，将评价可视化，生成评价表，按评价时间的先后排列。同时提供按菜品查找和按评分等级查找两种查找客户评价的方式。可以让管理员方便的找到菜品的评价以及客户对餐厅的赞美和批评。

- 功能

- a) 查找评价

查询评价信息时评价等级和菜名至少要输入其中一个作为查询条件，也可以同时输入两个进行查找。

- b) 按时间排序：将评价按照评价时间的先后顺序排序。

- 输入输出项

输入：评价等级（1-5 星），菜名

输出：评价

- 测试计划

测试模块	测试功能	测试指标
评价管理	查看评价	能够用菜名查询、评价等级查询、菜名和评价等级查询三种方式对评价进行查询，并输出查询的评价表。默认时显示所有评价按时间的排列的评价表。
	按时间排序	评价表可以按时间的先后顺序排序

5) 财务统计模块

- 程序描述

该程序主要实现财务统计，统计当天收入，月平均收入和年度月营业额。可以将月收入按金额大小排序，可以查看当天收入的详情、查询某个时间段的总营业额和日平均营业额。

- 功能

a) 查看当天的营业详情

显示当天的实时营业额和交易记录，交易记录包含订单号，用户电话，用户名和交易金额。

b) 查询营业额

根据时间查询某段时间的营业情况。月营业额表包含每个月的总营业额和日均营业额。点击月月营业额详情可得到该月每天的营业额。

- 输入输出项

输入：起始时间，终止时间。

输出：总营业额，日均营业额，月均营业额表，单日营业，单日交易记录表（订单号，用户电话，用户名和交易金额）

- 测试计划

测试模块	测试功能	测试指标
财务统计	按金额排序	月营业额表能够按金额大小排序
	查询营业额	输入一个事件段，能够输出该时间段的总营业，日均营业额，月营业额表。月营业额表点击详情可显示该月的日营业额表。
	查看当天营业情况	能够显示当天的营业额和交易记录

总结

至此，在确保需求合理的情况下，中期任务顺利完成，完成了系统软件；后面会尽量做到优化完善，希望加入独特的功能或亮点，过于复杂的问题可以简化处理。

关于文档的说明：此版本中期报告在第一版基础上进行了增改，改动之处较大的主要有“后期工作计划”部分、“原型设计”部分，对这两处进行了重新编写。另外新加入了“总结”章节。其余均为排版上的调整。

参考文献

[1]微信开放文档. App (Object object) [EB/OL]. <https://developers.weixin.qq.com/miniprogram/dev/reference/api/App.html>, 2020.

[2]wechat-miniprogram. GitHub wechat-miniprogram/recycle-view[EB/OL].

<https://github.com/wechat-miniprogram/recycle-view>, 2020.

[3] u011376686 的博客. 协同过滤算法商品推荐 [EB/OL]. <https://blog.csdn.net/u011376686/article/details/54287594/>, 2017.

[4] 微信支付开发文档. <https://pay.weixin.qq.com/wiki/doc/api/index.html>

[5] tenpay 使用文档 [EB/OL]. <https://www.npmjs.com/package/tenpay>, 2020.

[6] qq_36501609. 微信点餐需求分析 [EB/OL].

https://blog.csdn.net/qq_36501609/article/details/89929225, 2019-05-07.

[7] 编程小石头. 点餐系统的开发 [EB/OL].

https://blog.csdn.net/giushi_1990/article/details/97749686, 2020-02-07.

[8] 重庆冰炫科技有限公司. 冰点云订餐食堂报餐点餐 [EB/OL].

<http://www.cmerp.com/>, 2017.

[9] 嘻唰唰. 手机微信订餐报餐系统 [EB/OL]. <https://www.xeeyee.cn/>, 2017.

《软件综合实习》

编
码
实
现

(版本：1.0)

软件名称：智能食堂订餐系统

开发小组：胡周珏，崔令袆

2020 年 4 月 10 日

目录

1 开发计划.....	2
1.1 参与人员与承担的任务.....	2
1.2 开发前原始计划.....	2
2 功能描述.....	4
3 设计描述.....	4
3.1 组件说明.....	4
3.2 数据库关系表.....	9

1. 开发计划

1.1 参与人员与承担的任务

负责人	阶段	工作任务
崔令祎	起始阶段 (Inception)	<ul style="list-style-type: none"> • 需求分析； • 完成项目调研； • 进行可行性分析；
	细化阶段 (Elaboration)	<ul style="list-style-type: none"> • 对层次功能进行分析，拟定程序结构； • 把握系统每部分状态；完成面向对象设计；
	构建阶段 (Construction)	<ul style="list-style-type: none"> • 小程序模块详细设计
	测试维护阶段	<ul style="list-style-type: none"> • 软件审核 • 文档排版，打印、分发工作 • 用户培训工作
胡周珏	起始阶段 (Inception)	<ul style="list-style-type: none"> • 分析调查结果，拟定实施方案 • 编制项目开发计划；
	细化阶段 (Elaboration)	<ul style="list-style-type: none"> • 完成结构化设计； • 设计软件界面； • 对整体数据对象进行分析；设计数据库；
	构建阶段 (Construction)	<ul style="list-style-type: none"> • 后台管理系统详细设计
	测试维护阶段	<ul style="list-style-type: none"> • 软件评定 • 文件的编制 • 软件安装工作 • 用户手册编写

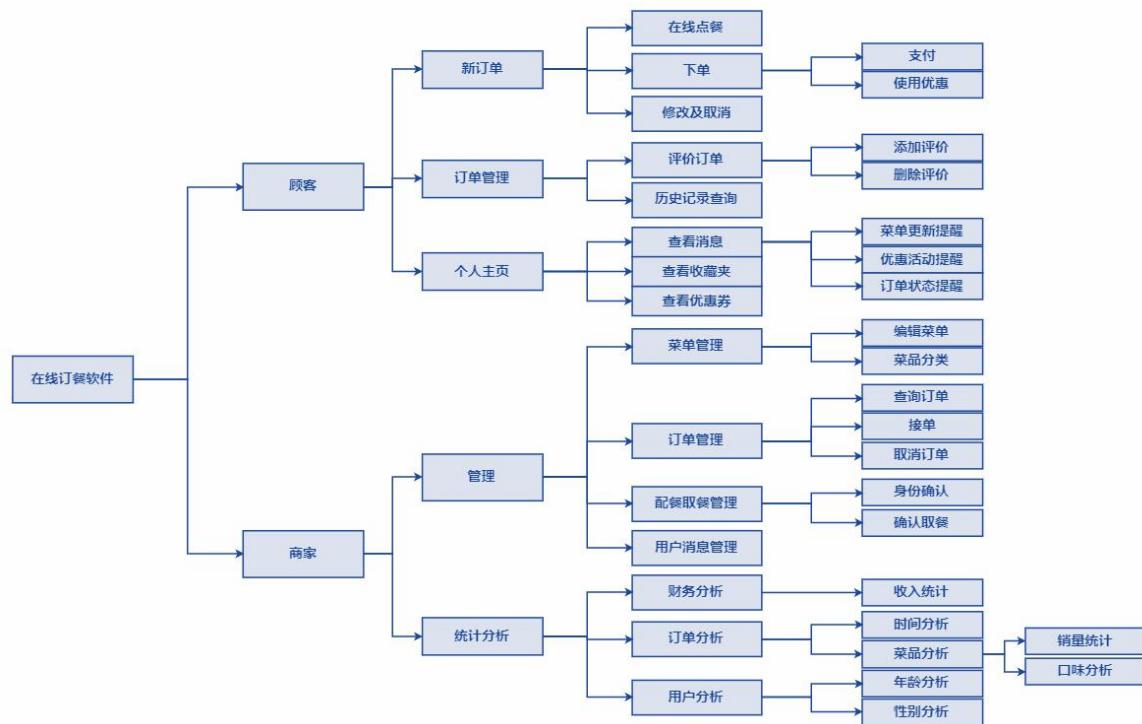
Table 1 项目任务分工表

1.2 开发前原始计划

里程碑	主要工作	应完成的文档	计划起始时间	计划终止时间
系统需求	项目调研	选题报告	2020. 3. 1	2020. 3. 5
	论证项目可行性		2020. 3. 5	2020. 3. 10
	指定项目初步计划		2020. 3. 10	2020. 3. 15
需求分析	确定系统运行环境	需求说明书	2020. 3. 5	2020. 3. 6
	确定系统功能及性能要求		2020. 3. 6	2020. 3. 12
	确定项目开发计划		2020. 3. 13	2020. 3. 18
概要设计	建立系统总体结构，划分功能模块	中期报告-1	2020. 3. 21	2020. 3. 23
	定义各功能模块接口		2020. 3. 21	2020. 3. 23
	数据库设计		2020. 3. 24	2020. 3. 25
详细设计	制定组装测试计划		2020. 3. 25	2020. 3. 26
	设计各模块具体实现算法	设计说明书	2020. 3. 26	2020. 3. 29
	确定模块间详细接口		2020. 3. 30	2020. 3. 31
实现	制定模块测试方案		2020. 4. 1	2020. 4. 2
	编写程序源代码	中期报告-2	2020. 4. 2	2020. 4. 12
	进行模块测试和调试		2020. 4. 12	2020. 4. 13
集成测试	编写用户手册		2020. 4. 13	2020. 4. 15
	对实现过程和已完成的文档进行评审		2020. 4. 15	2020. 4. 16
验收测试	执行集成测试计划	测试文档	2020. 4. 17	2020. 4. 20
	测试整个软件系统		2020. 4. 21	2020. 4. 22
	试用用户手册	用户手册	2020. 4. 22	2020. 4. 24
维护	编写开发总结	课程论文	2020. 4. 24	2020. 4. 30
	为纠正错误，完善应用而进行修改			
	对修改进行配置管理			
	编写故障报告和修改报告			
	修订用户手册			

Table 2 项目开发计划表

2. 功能描述



3. 设计描述

3.1 组件说明

以下列出了每一个组件的主要属性和方法。

(1) Comment.java

属性：

```

private int commentId;
private String openid;
private String name;
private String avatarUrl;
private String content;
private Date createTime;
  
```

(2) AdminInfo.java

属性：

```

private Integer adminId
private String phone
private String username
  
```

```
private String password  
private Integer adminType  
private Date createTime  
private Date updateTime
```

(3) Food.java

属性：

```
private Integer foodId;  
private String foodName;  
private BigDecimal foodPrice;  
private Integer foodStock;  
private String foodDesc;  
private String foodIcon;  
private Integer foodStatus = FoodStatusEnum.UP.getCode();  
private Integer leimuType;  
private Integer adminId;  
private Date createTime;  
private Date updateTime;
```

(4) Leimu.java

属性：

```
private Integer leimuId;  
private Integer adminId;  
private String leimuName;  
private Integer leimuType;  
private Date createTime;  
private Date updateTime;
```

(5) UserInfo.java

属性：

```
private int id;  
private String username;  
private String phone;
```

```
private String openid;  
private String zhuohao;  
private String renshu;  
private Date createTime;  
private Date updateTime;
```

(6) OrderDetail.java

属性：

```
private Integer detailId;  
private Integer orderId;  
private int foodId;  
private String foodName;  
private BigDecimal foodPrice;  
private Integer foodQuantity;  
private String foodIcon;
```

(7) OrderRoot.java

属性：

```
private Integer orderId;  
private String buyerName;  
private String buyerPhone;  
private String buyerOpenid;  
private BigDecimal orderAmount;  
private Integer orderStatus = OrderStatusEnum.NEW.getCode();  
private Integer payStatus = 0;  
private Integer cuidan = 0;  
private Date createTime;  
private Date updateTime;
```

(8) AdminCommentController.java

属性：

```
private CommentRepository repository;
```

方法：

```
public String list(@RequestParam(value = "page", defaultValue = "1") Integer page,  
@RequestParam(value = "size", defaultValue = "7") Integer size,ModelMap map)  
  
public String remove(@RequestParam(value = "commentId", required = false) Integer  
commentId,ModelMap map)
```

(9) AdminController.java

属性：

```
private AdminRepository repository;
```

方法：

```
public String list(HttpServletRequest request, ModelMap map)  
  
public String index(@RequestParam(value = "adminId", required = false) Integer adminId,ModelMap  
map)  
  
public String save(@Valid AdminForm form,BindingResult bindingResult, ModelMap map)  
  
public String remove(@RequestParam(value = "adminId", required = false) Integer adminId, ModelMap  
map)
```

(10) AdminCustomerController.java

属性：

```
private UserRepository repository;
```

方法：

```
public String list(@RequestParam(value = "page", defaultValue = "1") Integer page,  
@RequestParam(value = "size", defaultValue = "10") Integer size,ModelMap map)  
  
public String remove(@RequestParam(value = "id", required = false) Integer id,  
ModelMap map)
```

(11) AdminFoodController.java

属性：

```
private FoodRepository foodRepository;  
  
private LeiMuRepository leiMuRepository;
```

方法：

```
public String list(@RequestParam(value = "page", defaultValue = "1") Integer  
page,@RequestParam(value = "size", defaultValue = "6") Integer size,ModelMap map)
```

```

public String remove(@RequestParam(value = "foodId", required = false) Integer foodId, ModelMap map)

public String index(@RequestParam(value = "foodId", required = false) Integer foodId,
ModelMap map)

public String onSale(@RequestParam("foodId") int foodId, ModelMap map)

public String offSale(@RequestParam("foodId") int foodId, ModelMap map)

public String save(@Valid FoodReq form, BindingResult bindingResult, ModelMap map)

public String export(HttpServletRequest response, ModelMap map)

public String excel(ModelMap map)

public String uploadExcel(@RequestParam("file") MultipartFile file, ModelMap map)

```

(12) AdminLeimuController.java

属性：

private LeiMuRepository repository;

方法：

```

public String list(ModelMap map)

public String index(@RequestParam(value = "leimuId", required = false) Integer leimuId, ModelMap
map)

public String remove(@RequestParam(value = "leimuId", required = false) Integer leimuId, ModelMap
map)

public String save(@Valid LeimuReq form, BindingResult bindingResult, ModelMap map)

public String excel(ModelMap map)

public String uploadExcel(@RequestParam("file") MultipartFile file, ModelMap map)

```

(13) AdminOrderController.java

属性：

private WxOrderUtils wxOrder;

方法：

```

public String list(@RequestParam(value = "page", defaultValue = "1") Integer
page, @RequestParam(value = "size", defaultValue = "20") Integer size, ModelMap map)

public String remove(@RequestParam(value = "orderId", required = false) Integer orderId, ModelMap
map)

public String cancel(@RequestParam("orderId") int orderId, ModelMap map)

```

```
public String detail(@RequestParam("orderId") int orderId, ModelMap map)  
public String finished(@RequestParam("orderId") int orderId, ModelMap map)  
public String export(HttpServletRequest response, ModelMap map)
```

(14) LoginAdminController.java

属性：

Private AdminRepository repository;

方法：

```
public String loginAdmin(@RequestParam("phoneOrname") String phoneOrname,  
@RequestParam("password") String password, HttpServletRequest response)  
public String logout(HttpServletRequest request, HttpServletResponse response,  
ModelMap map)
```

3.2 数据库关系表

详情见系统设计文档

《软件综合实习》

软
件
测
试

(版本：2.0)

软件名称：智能食堂订餐系统

开发小组：胡周珏，崔令祎

2020 年 6 月 25 日

目录

一、软件测试计划.....	2
1. 测试策略.....	2
2. 测试范围和测试方法.....	2
2.1 测试的子系统.....	2
2.2 测试需求.....	2
3. 测试用例.....	4
4. 测试环境.....	9
4.1 硬件环境.....	9
4.2 软件环境.....	9
4.3 通信环境.....	10
二、软件测试总结.....	10
1. 测试进度情况.....	10
2. 测试问题解决.....	11
3. 综合评价.....	12
3.1 软件能力.....	12
3.2 缺陷和限制.....	12

一、 软件测试计划

1. 测试策略

(1) 进入准则

测试开始条件即进入准则：系统已经提交，软件发布，测试计划完成，测试用例设计完毕，测试环境搭建完毕，测试人员到位。

(2) 暂停准则

软件运行时出现崩溃或发现问题、漏洞

(3) 退出准则

系统功能全部测试，缺陷严重程度为 PI 的数量为 0；测试用例执行率=100%；提交测试文档。

测试需求	参加人员	预期结果
功能测试	胡周珏	确保用户所选定的功能正常实现
性能测试	崔令祎	测试在不同的工作量条件下的性能行为响应时间
安全性测试	胡周珏	对数据功能的访问正常，核实只能访问其所属用户类型被授予的功能
界面测试	崔令祎	窗口的对象和特征符合标准，进行的浏览包括窗口之间的切换，以及各种访问方法的使用
兼容测试	崔令祎	在不同的操作系统上能够运行

Table 1 整体测试策略表

2. 测试范围和测试方法

2.1 测试的子系统

点餐小程序和商家管理系统。

2.2 测试需求

用例	功能	编号	测试需求	测试方法
登录	验证账号和密	TR-01	1. 检测用户账户是否存在	手工测试

	码后进入系统		2. 检测用户的密码和账户是否匹配 3. 所有信息输入不能为空	
管理用户信息	添加/修改用户信息	TR-02	1. 输入的账户符合格式且不可重复 2. 输入的密码不小于 6 位数 3. 所有信息输入不能为空 4. 信息有效情况下，用户的信息能修改或添加成功	手工测试
	删除用户信息	TR-03	删除已有的用户	手工测试
管理菜品类目	添加/修改菜品类目	TR-04	1. 输入的符合格式且不可重复 2. 所有信息输入不能为空 3. 批量导入成功 4. 信息有效情况下，菜品类目的信息能修改或添加成功	手工测试
	删除菜品类目	TR-05	在该菜品类目下没有菜品时，成功删除	手工测试
管理菜谱	添加/修改菜品类目	TR-06	1. 输入的符合格式 2. 所有信息输入不能为空 3. 批量导入成功 4. 信息有效情况下，菜品的信息能修改或添加成功	手工测试
	删除菜品	TR-07	删除掉已有的菜品	手工测试
	导出菜谱	TR-08	将数据库中的菜单导入到 excel 并保存到指定文件夹	手工测试
	菜品上下架	TR-09	1. 能够正确的对菜品进行上架或下架操作，并相应改变商品状态 2. 上架的菜品客户才能加入购物车，下架菜品并不删除菜品的信息	手工测试

			息，只是客户无法选取	
管理订单	删除订单	TR-10	删除已有的已取消订单	手工测试
	导出订单	TR-11	将数据库中的订单导入到 excel 并保存到指定文件夹	手工测试
	取消订单	TR-12	取消订单，订单只有在未完成的状态下才能取消，若取消前未支付则直接取消，取消前是已支付则退款后取消	手工测试
查看订单信息	直接显示订单信息	TR-13	1. 能够正确的显示订单信息 2. 能够查看订单详情	手工测试
查看评价信息	直接显示评价信息	TR-14	能够正确的显示评价信息	手工测试
查看菜品类目信息	直接显示菜品类目信息	TR-15	能够正确的显示菜品类目信息	手工测试
查看菜谱	直接显示菜谱信息	TR-16	能够正确的显示菜谱信息	手工测试
查看用户信息	直接显示用户信息	TR-17	能够正确的显示用户信息	手工测试
管理员权限	不同管理员有不同访问权限	TR-18	1. 测试普通管理员和超级管理员的权限是否正确 2. 修改管理员权限后权限是否发生变化	手工测试

Table 2 功能需求表

3. 测试用例

需求项	测试需求编号	测试用例编号	测试用例	
			名称	检测登录账户是否存在
登录	TR-01	TR-01-01	测试对象	登录界面
			输入	账号

管理用户	TR-02	TR-01	输出	无
			步骤	提交给数据库之后查询是否存在并返回结果
			名称	检测登录密码是否正确
			测试对象	登录界面
			输入	密码
			输出	无（成功），账户或密码错误（失败）
			步骤	提交给数据库之后查询是否和用户名匹配并返回结果
			名称	检测空输入
			测试对象	登录界面
			输入	空
			输出	弹出窗口显示：请输入手机号/密码
			步骤	输入框检测输入是否为空
		TR-02-01	名称	检测手机号格式是否正确
			测试对象	管理员界面
			输入	手机号 111222
			输出	手机需要 11 位数字
			步骤	在输入时判断输入框内的字符是否满足 11 位
		TR-02-02	名称	检测手机号是否已存在
			测试对象	管理员界面
			输入	手机号
			输出	无（不存在）；该账户已存在，请重新输入（存在）
			步骤	提交给数据库之后查询是否存在并返回结果
		TR-02-03	名称	检测空输入
			测试对象	管理员界面
			输入	空
			输出	弹出窗口显示：输入不能为空
			步骤	输入框检测输入是否为空
		TR-02-04	名称	检测密码字符数
			测试对象	管理员界面
			输入	密码 123
			输出	密码不能少于 6 个字符
			步骤	在输入时判断输入框内的字符是否小于 6 个并弹出窗口
		TR-02-05	名称	信息有效情况下，用户的信息能修改或添加成功
			测试对象	管理员界面

			输入	有效信息
			输出	添加/修改成功
TR-03	TR-03-01	名称	检测能否删除已有用户	
		测试对象	管理员界面	
		输入	无	
		输出	删除成功	
		步骤	选择已有用户，点击删除按钮	
管理菜品类目	TR-04-01	名称	检测类目 type 输入格式	
		测试对象	菜品类目管理界面	
		输入	aa	
		输出	只能输入整型数据	
		步骤	输入框检测输入是否为整型	
	TR-04-02	名称	检测类目 Type 是否已存在	
		测试对象	菜品类目管理界面	
		输入	手机号	
		输出	无（不存在）；已存在，请重新输入（存在）	
		步骤	提交给数据库之后查询是否存在并返回结果	
	TR-04	名称	检测空输入	
		测试对象	菜品类目管理界面	
		输入	空	
		输出	弹出窗口显示：输入不能为空	
		步骤	输入框检测输入是否为空	
	TR-04-04	名称	检测批量导入	
		测试对象	菜品类目管理界面	
		输入	选取一个 excel 文件，且文件格式符合要求	
		输出	无（成功）；导入错误（失败）	
		步骤	检测文件是否符合要求，符合导入数据	
	TR-04-05	名称	检测信息有效情况下，菜品类目的信息能修改或添加成功	
		测试对象	菜品类目管理界面	
		输入	有效信息	
		输出	添加/修改成功	
		名称	检测能否成功删除已有菜品类目	
	TP-05	测试对象	菜品类目管理界面	
		输入	无	
		输出	无（成功）；该菜品类目下还有有菜品（失败）	

			步骤	在数据库中检测，该菜品类目下是否有菜品，若有，无法删除，若无，成功删除。
管理菜谱	TR-06	TR-06-01	名称	检测库存输入格式
			测试对象	菜谱管理界面
			输入	aa
			输出	只能输入整型数据
			步骤	输入框检测输入是否为整型
	TR-06-02		名称	检测单价输入格式
			测试对象	菜谱管理界面
			输入	aa
			输出	只能输入整型数据
	TR-06	TR-06-03	步骤	输入框检测输入是否为整型
			名称	检测空输入
			测试对象	菜谱管理界面
			输入	空
			输出	弹出窗口显示：输入不能为空
	TR-06-04		步骤	输入框检测输入是否为空
			名称	检测批量导入
			测试对象	菜谱管理界面
			输入	选取一个 excel 文件，且文件格式符合要求
			输出	无（成功）；导入错误（失败）
	TR-06-05		步骤	检测文件是否符合要求，符合导入数据
			名称	检测信息有效情况下，菜品的信息能修改或添加成功
			测试对象	菜谱管理界面
			输入	有效信息
			输出	添加/修改成功
	TR-07	TR-07-01	名称	检测能否成功删除已有菜品
			测试对象	菜谱管理界面
			输入	无
			输出	删除成功
			步骤	选择已有菜品，点击删除按钮
	TR-08	TR-08-01	名称	检测导出菜谱
			测试对象	菜谱管理界面
			输入	无
			输出	菜单导出.xls
			步骤	点击菜单导出按钮
	TR-09	TR-09-01	名称	检测菜品下架

管理订单			测试对象	菜谱管理界面
			输入	无
			输出	菜品下架，且状态变为已下架，小程序端不显示该菜品
			步骤	选取销售中的菜品，点击下架按钮
			名称	检测菜品上架
			测试对象	菜谱管理界面
			输入	无
			输出	菜品上架，且状态变为销售中，小程序端显示该菜品
			步骤	选取下架中的菜品，点击上架按钮
			名称	检测能否成功删除已取消的订单
查看	TR-10	TR-10-01	测试对象	订单管理界面
			输入	无
			输出	删除成功
			步骤	选择已取消订单，点击删除按钮
			名称	检测导出订单
	TR-11	TR-11-01	测试对象	订单管理界面
			输入	无
			输出	订单导出.xls
			步骤	点击订单导出按钮
			名称	检测能否成功取消订单
	TR-12	TR-12-01	测试对象	订单管理界面
			输入	无
			输出	成功取消（订单未完成）；订单状态不正确（订单已完成）
			步骤	选择订单，点击取消按钮
			名称	检测是否正确显示订单列表
	TR-13	TR-13-01	测试对象	订单管理界面
			输入	点击订单管理
			输出	显示订单列表
			步骤	单击订单管理，显示订单列表
			名称	检测是否正确显示订单信息
	TR-13	TR-13-02	测试对象	订单管理界面
			输入	点击订单详情
			输出	显示订单详细信息
			步骤	单击订单详情，显示订单详细信息
			名称	检测是否正确显示评价列表
	TR-14	TR-14-01	测试对象	评价管理界面
			输入	点击评价管理

			输出	显示评价信息列表
			步骤	单击评价管理，显示评价信息列表
TR-15	TR-15-01		名称	检测是否正确显示菜品类目列表
			测试对象	菜品类目管理界面
			输入	点击菜品类目管理
			输出	显示菜品类目信息列表
			步骤	单击菜品类目管理，显示菜品类目信息列表
TR-16	TR-16-01		名称	检测是否正确显示菜品列表
			测试对象	菜谱管理界面
			输入	点击菜谱管理
			输出	显示菜品信息列表
			步骤	单击菜谱管理，显示菜品信息列表
TR-17	TR-17-01		名称	检测是否正确显示用户信息列表
			测试对象	用户管理界面
			输入	点击用户管理
			输出	显示用户信息列表
			步骤	单击用户管理，显示用户信息列表
权限	TR-18	TR-18-01	名称	检测普通管理员访问权限
			测试对象	管理员界面
			输入	普通管理员登录
			输出	不能访问管理员界面
	TR-18	TR-18-02	名称	检测超级管理员访问权限
			测试对象	管理员界面
			输入	超级管理员登录
			输出	可以访问管理员界面对管理员信息进行修改

Table 3 测试用例表

4. 测试环境

4.1 硬件环境

服务器：一台 8GB 内存 500GB 硬盘的笔记本

客户端：具有 6GB RAM、128GB ROM 数据网络接入的 Andriod 智能手机。

4.2 软件环境

服务器端：

操作系统：Windows 10

开发软件：MySQL, IntelliJ IDEA

客户端：

Android 9

4.2 通信环境要求

服务器：10Mbps 以上的广域网接入，包括但不限于有线网络、WIFI。

客户端：移动通信网络。

二、软件测试总结

1. 测试进度情况

测试需求标识号	测试用例标识号	测试用例状态	测试结果
TR-01	TR-01-01	已测试	已通过
	TR-01-02	已测试	已通过
	TR-01-03	已测试	已通过
TR-02	TR-02-01	已测试	未通过
	TR-02-02	已测试	未通过
	TR-02-03	已测试	未通过
	TR-02-04	已测试	未通过
	TR-02-05	已测试	已通过
TR-03	TR-03-01	已测试	已通过
TR-04	TR-04-01	已测试	已通过
	TR-04-02	已测试	已通过
	TR-04-03	已测试	已通过
	TR-04-04	已测试	已通过
	TR-04-05	已测试	已通过

TR-05	TR-05-01	已测试	未通过
TR-06	TR-06-01	已测试	已通过
	TR-06-02	已测试	已通过
	TR-06-03	已测试	已通过
	TR-06-04	已测试	已通过
	TR-06-05	已测试	已通过
TR-07	TR-07-01	已测试	已通过
TR-08	TR-08-01	已测试	已通过
TR-09	TR-09-01	已测试	已通过
	TR-09-02	已测试	已通过
TR-10	TR-10-01	已测试	已通过
TR-11	TR-11-01	已测试	已通过
TR-12	TR-12-01	已测试	已通过
TR-13	TR-13-01	已测试	已通过
	TR-13-02	已测试	已通过
TR-14	TR-14-01	已测试	已通过
TR-15	TR-15-01	已测试	已通过
TR-16	TR-16-01	已测试	已通过
TR-17	TR-17-01	已测试	已通过
TR-18	TR-18-01	已测试	已通过
	TR-18-02	已测试	已通过

Table 4 测试用例执行结果表

2. 测试问题解决

测试需求标识号	测试用例标识号	错误或问题描述	错误或问题状态
TR-02	TR-02-01	手机号格式错误未检测出	已解决

TR-02	TR-02-02	账户重复检测未检测出	已解决
TR-02	TR-02-03	添加管理员时空输入未检测出	已解决
TR-02	TR-02-04	密码输入不满 6 位未检测出	已解决
TR-05	TR-05-01	菜品类目下还有菜品时未阻止删除	待解决

Table 5 测试问题解决表

3. 综合评价

3.1 软件能力

软件基本能够完成设计时的各项功能需求，各个用例也基本得到实现。

3.2 缺陷和限制

- (1) 由于未进行压力测试，系统的稳定性尚不可知，面对较高的并发请求时系统的稳定性存在一定的不稳定因素。
- (2) 系统的界面有些简陋，不够美观。
- (3) 没有在不同的操作系统和机器上进行测试，兼容性方面也可能存在问题。

《软件综合实习》

软
件
交
付

(版本：2.0)

软件名称：智能食堂订餐系统

开发小组：胡周珏，崔令袆

2020 年 6 月 26 日

目录

1 交付清单.....	2
2 用户手册.....	2
2.1 引言.....	3
2.2 运行环境.....	3
2.3 运行说明.....	3

1 交付清单

阶段	文件名	文件类型
计划、需求分析阶段	选题报告	pdf
设计阶段	中期报告（1）	pdf
	项目进度计划表	pdf
开发阶段	工作日志	pdf
	中期报告（2）	pdf
	软件演示	MP4
	开发计划	pdf
	业务需求	pdf
	系统需求	pdf
测试、重构阶段	系统分析	pdf
	系统设计	pdf
	编码实现	pdf
	软件测试	pdf
	软件交付	pdf
	软件演示视频	MP4
	汇报材料	PPT
总结、交付阶段	课程论文	pdf
	软件开发文档	pdf
	软件包	zip
	实习总结	pdf

Table 1-1 文档清单

2 用户手册

2.1 引言

用户手册的编写目的是对智能食堂订餐软件的使用和运行进行说明。手册中介绍了软件的功能，规定了运行环境，详细地说明了使用过程和运行步骤。能够辅助用户顺利使用软件。

本用户手册包括运行环境、运行说明两部分。运行环境部分规定了软件的使用环境。运行说明介绍了系统的运行步骤。本文件的各部分内容联系紧密，详尽地描述了软件的情况和使用。各部分互为补充和对照，共同为用户的使用进行指导。

2.2 运行环境

2.2.1 硬件环境

服务器：一台 8GB 内存 500GB 硬盘的笔记本

客户端：具有 6GB RAM、128GB ROM 数据网络接入的 Andriod 智能手机。

2.2.2 软件环境

(1) 服务器端：

操作系统：Windows 10

开发软件：MySQL，IntelliJ IDEA

(2) 客户端：Android 8/9

2.2.3 通信环境要求

服务器：10Mbps 以上的广域网接入，包括但不限于有线网络、WIFI。

客户端：移动通信网络、WIFI。

2.3 运行说明

2.3.1 点餐小程序

(1) 登录：打开小程序，允许授权

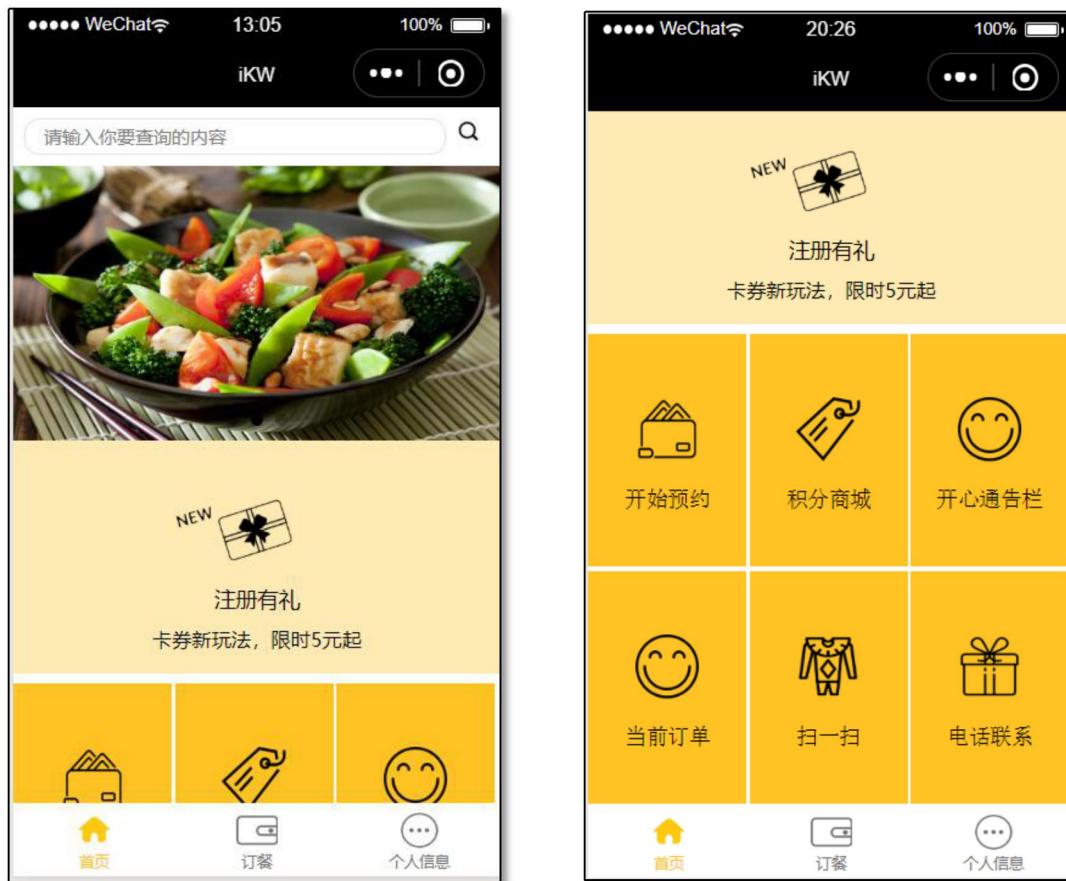


Figure 4-1 小程序首页

(2) 注册：点击个人信息，填写会员信息



Figure 4-2 个人主页



Figure 4-3 修改个人信息

- (2) 浏览菜单：点击订餐，出现菜单主页，单击菜品类别，进入相应的菜系。
- (3) 点餐：将要点的菜加入购物车，选好后，点击“选好了”，进入结算。



Figure 4-3 点餐界面

(4) 我的订单：可以查看、待上餐、已取消、待评价、已完成的订单。



Figure 4-4 待上餐订单



Figure 4-5 已完成订单

(4) 发布评价：客户在用餐后可对餐厅做出评价



Figure 4-6 发布评价

(5) 查看评价：在个人中心，“我的评价”里可以查看到餐厅的所有评价和个人评价。



Figure 4-7 查看所有评价



Figure 4-7 查看我的评价

2.3.2 商家管理系统

(1) 登录：输入账户和密码

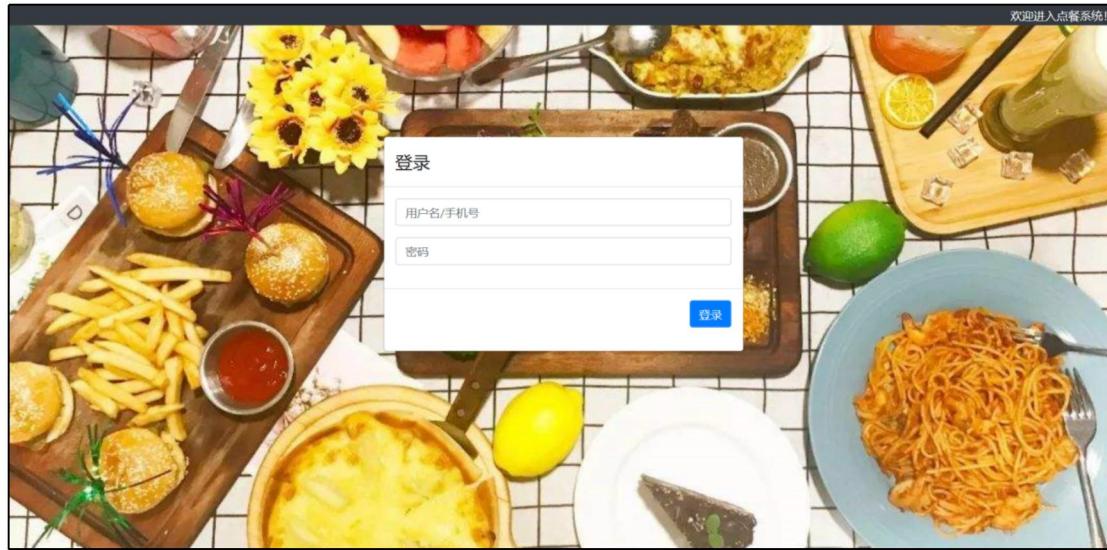


Figure 4-8 登录界面

(2) 查看菜单

管理员		菜谱管理										
菜品管理	菜品类目管理		添加菜品			批量导入			菜单导出			
	菜谱管理											
	菜品编号	菜品名	单价	库存	菜品描述	类目	图片	创建时间	菜品状态	操作		
	72	白米饭	2	326	东北大米，粒粒饱满。	1		2020-4-30 7:12:51	已下架	修改	上架	删除
	73	皮蛋瘦肉粥	5	200	材料：大米，皮蛋，猪肉；质地黏稠，口感顺滑、清淡可口,开胃消食,可促进伤口愈合,是病后体虚的食疗食谱之一。	1		2020-4-30 7:12:51	销售中	修改	下架	删除
	74	西红柿炒鸡蛋	18	149	材料：西红柿、鸡蛋；营养搭配合理，鲜艳、爽口、开胃。	2		2020-4-30 7:12:51	销售中	修改	下架	删除
	75	青椒炒肉丝	30	100	材料：青椒、肉食；口味香辣，菜品色香味俱全。	2		2020-4-30 7:12:51	销售中	修改	下架	删除
	76	海带排骨汤	28	173	材料：海带、排骨；此汤菜肉烂脱骨，海带滑烂，汤鲜味美，海带含有丰富的钙，可防人体缺钙，还有降血压的功效。	3		2020-4-30 7:12:51	销售中	修改	下架	删除
	77	铁板鱿鱼	45	98	材料：鱿鱼、红椒；酱香可口，鱿鱼爽滑弹牙。	5		2020-4-30 7:12:52	销售中	修改	下架	删除

Figure 4-9 菜单列表

(3) 增加/修改菜品

The screenshot shows a '添加单个菜品' (Add Single Dish) form. On the left is a dark sidebar menu with options: 管理员 (Administrator), 菜品类目管理 (Category Management), 菜谱管理 (Menu Management), 用户管理 (User Management), 订单管理 (Order Management), 展示图管理 (Display Management), 评价管理 (Review Management), 财务分析 (Financial Analysis), and 退出登陆 (Logout). The main area has a title '添加单个菜品'. It contains several input fields: '菜品名' (Dish Name), '菜品价格' (Dish Price), '菜品库存' (Dish Inventory), '食物描述' (Food Description), '所属类目' (Category), which has a dropdown menu showing '主食' (Main Course); '图片' (Image), which is currently empty; and a blue '提交' (Submit) button.

Figure 4-10 添加或修改菜品

(4) 查看管理员信息：登录的账户为超级管理员时可以查看管理员信息，并进行增删查改，但若登录账户为普通管理员，则无权查看管理员信息。

The screenshot shows a '管理员管理' (Administrator Management) page. The left sidebar is identical to Figure 4-10. The main area has a title '管理员管理' and a '添加管理员' (Add Administrator) button. Below is a table with administrator data:

ID	名字	密码	手机号	类型	创建时间	修改时间	操作
1	ll	123456	123456	管理员	2020-5-30 8:33:24	2020-5-30 8:33:53	修改 删除
89	张三	1	111111	管理员	2020-5-30 8:43:01	2020-5-30 8:43:01	修改 删除
91	2	22	22	普通员工	2020-5-30 8:46:55	2020-5-30 8:47:15	修改 删除

Figure 4-11 查看管理员信息 (超级管理员)



Figure 4-12 查看管理员信息（普通管理员）

(5) 添加/修改管理员信息

Figure 4-13 添加或修改管理员信息

(6) 查看菜品类目

The screenshot shows a software interface titled 'Food Category Management'. On the left is a vertical navigation bar with the following items: 'Administrator', 'Food Category Management' (which is highlighted in yellow), 'Cuisine Management', 'User Management', 'Order Management', 'Display Management', 'Evaluation Management', 'Financial Analysis', and 'Logout'. The main content area has a title 'Food Category Management' and two buttons: 'Add Category' and 'Batch Import'. Below is a table with columns: Category ID, Name, Creation Time, Modification Time, and Operations (Edit, Delete). The table contains five rows of data:

类目编号	名字	创建时间	修改时间	操作
1	主食	2020-4-30 7:12:34	2020-4-30 7:12:34	修改 删除
2	炒菜	2020-4-30 7:12:34	2020-4-30 7:12:34	修改 删除
3	汤品	2020-4-30 7:12:34	2020-4-30 7:12:34	修改 删除
4	砂锅	2020-4-30 7:12:34	2020-4-30 7:12:34	修改 删除
5	铁板	2020-5-30 12:52:42	2020-5-30 12:52:42	修改 删除

Figure 4-14 Food Category List

(7) 添加/修改菜品类目

The screenshot shows a software interface titled 'Add Food Category'. On the left is a vertical navigation bar with the same items as Figure 4-14. The main content area has a title 'Add Food Category' and two input fields: 'Category Name' and 'Category Type'. A blue 'Submit' button is located below the fields.

Figure 4-15 Add or Modify Category List

(8) 批量导入菜品类目：选择文件，导入数据



Figure 4-16 批量导入菜品类目

(9) 查看用户信息

用户管理					
用户编号	用户名	用户电话	积分	会员等级	
1	小崔	135589657741	500	L1	
47	小胡	13886241779	500	L1	
上一页 1 下一页					

Figure 4-17 用户列表

(10) 查看订单信息

餐厅订单管理							
欢迎进入点餐系统!							
导出订单到Excel							
订单id	用户名	手机号	订单金额	订单状态	创建时间	操作	
95	小胡	13886241779	96	已完结	2020-5-30 12:59:47	详情	取消
83	小胡	13886241779	88	已评价	2020-4-30 13:56:40	详情	取消
1	小胡	13886524178	22	已取消	2020-4-30 7:10:41	详情	删除

上一页 1 下一页

Figure 4-18 订单列表

(11) 查看订单详细信息：选取订单点击详情，获取订单详细信息

菜品订单详情页						
欢迎进入点餐系统!						
导出订单到Excel						
订单id	商品图片	商品名称	价格	数量	总额	
95		西红柿炒鸡蛋	18	1	18	
95		铁板鱿鱼	45	1	45	
95		海带排骨汤	28	1	28	
95		皮蛋瘦肉粥	5	1	5	

返回

Figure 4-18 订单详细信息列表

(12) 查看评价信息

评价管理						
评价编号	用户头像	用户名	评价内容	评价时间	操作	
1		小崔	服务很好，上菜速度很快，就是青椒炒肉丝肉少了点，全是青椒。	2020-4-27 7:30:22	删除	
2		小张	味道很好，老板还送了一个荷包蛋，开心~	2020-4-23 7:31:46	删除	
3		不知道取什么名字	差评差评差评差评差评，铁板鱿鱼全是辣椒洋葱，这么辣怎么吃啊，鱿鱼也很少emmmmmm	2020-4-26 7:33:08	删除	
100		小胡	味道还可以，铁板鱿鱼最好吃，下次还点！	2020-5-30 13:00:38	删除	

Figure 4-19 餐厅评价信息列表