



四川大学计算机学院·软件学院
College of Computer Science &
College of Software Engineering
Sichuan University

No. 24 South Section 1, Yihuan Road, Chengdu, Sichuan, China 610065

Tel: (28) 85403330 <http://cs.scu.edu.cn> <http://sw.scu.edu.cn>

软件设计说明书

饱了么智能订餐系统

学院	软件学院
组号	第四小组
项目	饱了么智能订餐系统
组长	李美瑾
组员	黄晨钊，姬茗瑞，樊杨，刘雨果

目录

1.	引言	3
1.1	编写目的.....	3
1.2	背景.....	3
1.3	定义.....	3
1.4	参考资料.....	4
2.	总体设计	5
2.1	系统总架构.....	5
2.2	系统性能.....	6
2.3	输入输出要求.....	6
3.	功能设计	8
3.1	餐厅管理员系统功能设计	8
3.2	服务员系统功能设计	13
3.3	后厨系统功能设计	17
4.	界面设计	19
4.1	界面设计概述.....	19
4.2	餐厅管理员界面设计	21
4.3	服务员系统界面设计	27
4.4	后厨系统界面设计	32
5.	数据库设计	34
5.1	数据关系设计	34
5.2	数据表详细设计	35
6.	系统防错设计	38
6.1	AI 识别接口精度设计	38
6.2	登录设计	39
6.3	表单提交设计	39
6.4	出错处理设计	39
7.	部署设计	40
8.	接口设计	41

1. 引言

1.1 编写目的

本文档描述软件产品系统设计说明书的目的是：

1. 明确智能订餐系统的数据结构和系统结构，界定系统实现功能的范围；
2. 定义软件总体要求，作为用户和软件开发人员之间相互了解的基础；
3. 说明系统整体的概要设计。为下一步的详细设计做准备，也为之后系统的编码实现提供支持；
4. 提供性能要求、初步设计和用户影响的信息，作为软件人员进行软件结构设计和编码的基础；
5. 作为软件总体测试的依据。

本说明书的主要使用人员为：项目主管、项目开发人员、系统维护人员、系统管理员和编写详细设计人员等。

1.2 背景

智能订餐系统是一套完整的针对各类型餐厅经营管理流程信息化的软件产品，从顾客进入餐厅订餐到用餐结束再到最后的结账买单，需要一套全面的、高效的信息化管理软件来辅助，以对餐厅菜品、员工、订餐流程、菜品烹饪、传菜、结账等过程进行信息化管理，能够及时向不同角色的员工发布实时的公告信息，并能检索查看餐厅的历史经营数据。

该系统的提出和开发者为软件项目管理课程第四小组全体成员，用户面向智能餐饮业从业人员。该系统基于 Web 和微信小程序原生系统，需要无线局域网环境支持。

1.3 定义

缩写、术语符号	解释
OrderSystem	本系统英文名，适用于餐厅运营过程中顾客到店后服务员远程点菜、传菜、结算等服务的信息化管理
Tomcat	系统所使用的应用服务器（Application Server）
MySQL	系统所使用的数据库管理系统（DBMS）
SQL	Structured Query Language（结构化查询语言），一种用于访问、查询数据库的语言

系统	若未特别指出，统指智能订餐系统
数据中心	以各类数据为核心，依托成熟的存储、数据库、GIS、网络等技术，按照统一标准，建立的具有信息管理、分析、查询、统计及服务的一体化数据管理体系。
数据管理	利用数据库、数据仓库、元数据和网络等技术，建立分布式、集中式或集中加分布式数据管理系统，开展数据接收、组织存储、运行维护、更新、共享交换等工作，实现对数据资源的有效组织和应用。
数据维护	在制定维护方案基础上，对数据和数据库进行的日常维护与监控、备份与恢复、应急处理和监督管理等，从而保护数据的安全性和可移植性。
用户	智能订餐系统的使用者
内聚性	单个程序模块所执行的诸任务在功能上的互相关联的程序。
耦合	在软件模块之间相互依赖的方式和程度。类型包括公共环境耦合、内容耦合、控制耦合、数据耦合、混合耦合和相依耦合。
并发性	在同一时间间隔内多个活动的出现。并发能通过多个线程的交错或同时执行来获得。
一致性	在文档或系统或组件的各部分之间，一致、标准化、无矛盾的程度。
约束	一种语义条件或限制。在 UML 中，某些约束是预定义的，另一些则是用户定义的。约束时 UML 中三种可扩展性机制之一。
崩溃	计算机系统或部件的突然的和完全地失效。
主程序员	主程序员组的领导者、一高级程序员，他的职责包括产生赋予组的软件的关键部分，协调组的活动，瓶身其他组成员的工作且对正在开发的软件有全面的技术了解。
本项目	本文档中本项目均指智能订餐系统，即我们小组在项目管理课程中开发的项目。

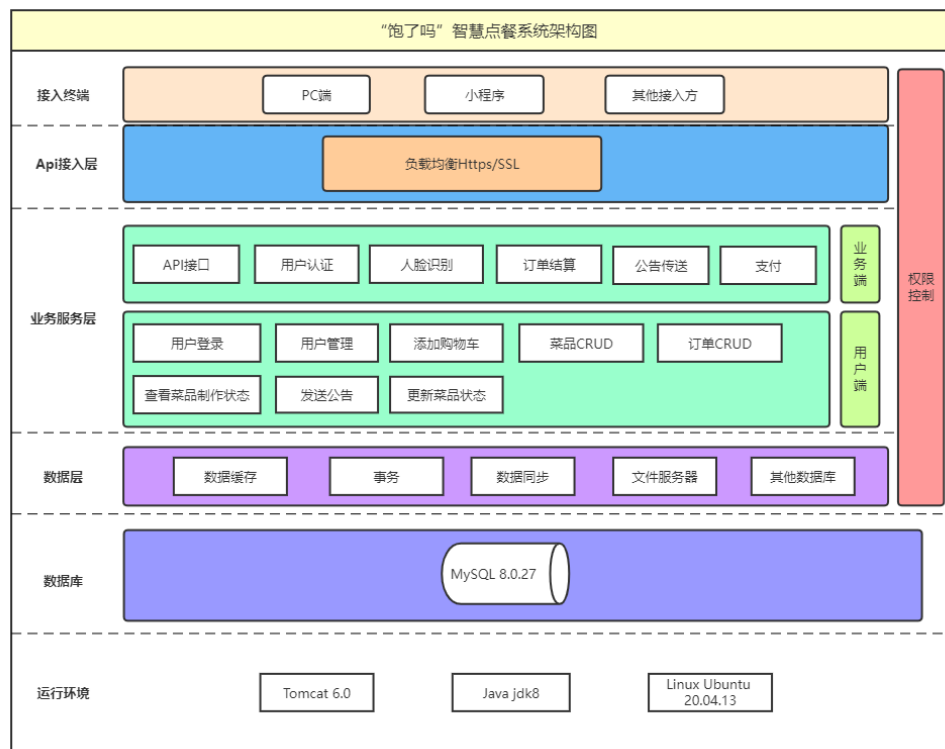
1.4 参考资料

序号	文档名称	文档编号	版本	发布日期
1	《计算机信息系统安全保护等级划分准则》		GB 17859	
2	《智能订餐系统需求规格说明书》		V2.0	2022
3	《Element UI 官方文档》			
4	《SpringCloud 官方文档》			
5	《软件工程过程》		GB/T 11457	2016
6	《计算机软件开发规范》		GB8566-88	
7	《信息技术与软件生存周期过程》		GB/T 8566	2007

2. 总体设计

2.1 系统总架构

系统架构设计图如下：



如图 1 所示，“饱了吗”智慧点餐系统的架构包含接入中断、API 接入层、业务服务层、数据层、数据库，其运行环境为 Tomcat 6.0、Java jdk8、Linux Ubuntu 20.04.13，并且在跨架构层级上做了权限管理，以保证数据的安全。

1. 接入终端

由于本系统需要满足用户移动端使用的需求，需要为网页端和微信小程序端设置数据访问接口，并预留其他设备的访问接口，以保证系统的可拓展性。

2. API 接入层

API 接入层为接入终端提供数据交互的接口，接入终端向系统发送 http/https 请求。网页端使用 Ajax 请求，微信小程序使用 wx.request 请求。

3. 业务服务层

这一层提供系统的主要业务和服务逻辑，向下与数据层进行交互，向上与 API 接入层进行数据交换。业务端多由系统自发产生，而用户端由用户行为进行

触发。

4. 数据层

数据层提供相应的数据服务，向下操纵数据库，向上与业务服务层交互数据。数据层主要提供数据缓存、事务、数据同步、文件服务器等服务。

5. 数据库

数据库主要储存数据，并向上提供数据。

2.2 系统性能

实时性：系统需要能够实时更新数据（订单状态、菜品状态等），数据同步时长不超过 0.1 秒。

并发性：满足较高并发的安全访问，并发数至少满足 100，对于更高数量的并发访问，能够采取队列等待方式进行高密度访问的缓冲，同时向用户界面显示相应提示信息。

可靠性：该系统在数据和文件传输方面需要能够准确无误的进行传输，尽可能少量的丢包现象。

安全性：系统需要设置路由拦截，对于管理员登录应该使用人脸验证做双重安全保证，避免管理员密码被盗后修改重要信息；同时数据库储存的菜品信息、订单信息、员工信息需要做安全性保证，避免被黑客入侵。

2.3 输入输出要求

本系统的输入输出需求如下表所示：

编号	功能名称	功能描述	输入	处理
G1	用户登录	提供用户登录验证功能，并对异常登录情况进行反馈	用户名、密码	正确则进入系统，错误弹出提示信息
G2	修改个人信息	为用户提供修改用户信息的接口	个人信息（用户名、密码、性别等）	完成填写后将数据在数据库中进行更新
A1	员工管理	为餐厅管理员提供员工管理接口，其中包括添加员工、注销员工和查看员工销	添加员工操作/注销员工操作	更新员工信息/返回员工销售榜详情

		售榜等功能		
A2	菜品管理	为餐厅管理员提供菜品管理接口，其中包括添加菜品、删除菜品和查看菜品详情等功能	添加菜品操作/删除菜品操作/相应订单的订单 ID	更新菜品信息/返回菜品详情
A3	订单管理	为餐厅管理员提供订单管理接口，其中包括查看历史订单和查看当前订单功能	查看订单操作	返回相应订单信息
A4	发送公告	为餐厅管理员提供发送公告接口，其中包括给后厨人员发送公告、给服务员发送公告等功能	公告内容+发送人 ID	将公告发送给相应用户，并对该用户进行消息提醒
W1	点单	为服务员提供点单接口，其中包括扫码选择桌号、选择菜品、展示购物车和确认订单等功能	桌面上的二维码/菜品 ID/点击展示购物车/确认订单	展示当前该桌的点单信息/将该菜品加入购物车/返回购物车中的菜品信息/将订单更新至数据库
W2	传菜	服务员可以通过该接口查看待传菜列表	点击查看传菜列表	返回待传菜列表信息
W3	查看公告	服务员可以通过该接口查看公告，其中包括查看管理员公告和查看后厨信息	点击查看公告	返回公告列表
W4	结算订单	为服务员提供结算订单接口，其中包括查看结算订单列表和选定订单结算	点击查看结算订单列表/订单 ID	返回结算订单列表/对选定订单进行结算
C1	烹饪菜品	为后厨人员提供查看待烹饪菜品列表、修改菜品状态等接口	点击查看待烹饪菜品列表/修改菜品信息	返回待烹饪菜品列表/更新相应菜品的数据
C2	向服务员	后厨人员可以通过该接口	消息内容+服	将消息发送至相应服务

	发送消息	向服务员发送消息	务员 ID	员，并进行提醒
--	------	----------	-------	---------

3. 功能设计

3.1 餐厅管理员系统功能设计

3.1.1 模块描述

餐厅管理员系统是整个系统中不可或缺的部分，起到了管理整个餐厅的作用，本模块直接关系到了餐厅的整体运作。本模块具体分为三个子部分，分别是人员管理、菜品管理、订单管理和公告系统。人员管理主要涉及餐厅管理员对于餐厅员工的信息管理，包括查看，编辑员工的相关属性等。菜品管理主要提供展示和编辑、添加功能，管理员可以对于菜品详细情况进行修改或添加。订单管理包括历史订单的查看和正在进行中的订单的维护以及订单数据分析。公告系统是指管理员可以发布公告，其他餐厅工作人员可以查看公告。餐厅管理员系统通过对餐厅人员、菜品和订单的管理与维护，支持餐厅的数字化运行，管理员可以通过服务系统进行餐厅的运营和维护，简单便捷，易用高效。

3.1.2 功能

3.1.2.1 员工管理

(1) 员工信息列表展示

描述：使用列表形式展示所有员工的基本信息。

说明：该员工列表包括分页、搜索、概要展示和详细展示。

(2) 编辑员工信息

描述：管理员可编辑员工信息。

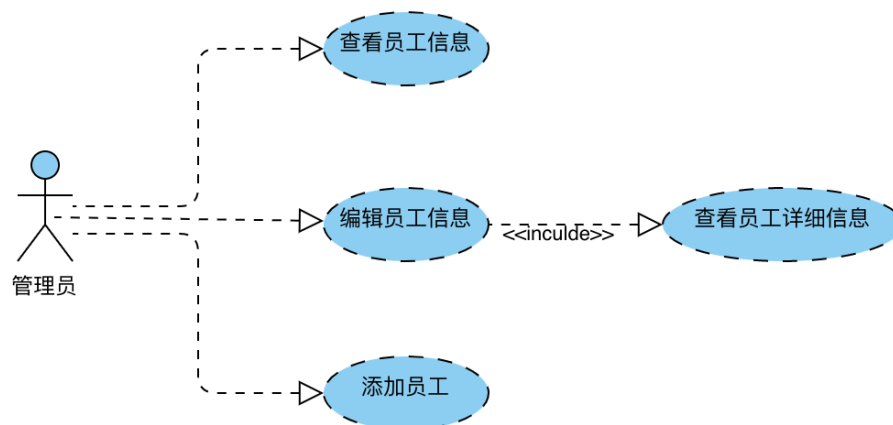
说明：管理员查看单一用户的详细信息，包括员工头像、编号、姓名与职位、工资、奖金与电话号码。但是只能编辑员工的职位、工资与奖金。

(3) 添加员工

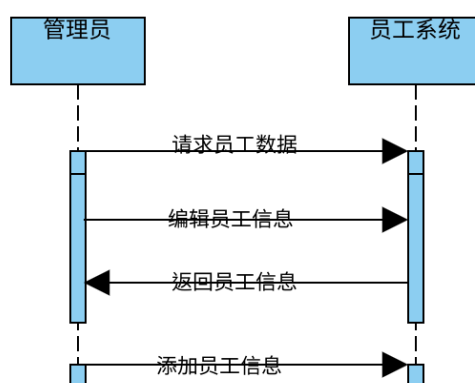
描述：增加新员工，并为该员工指明相应的信息。

说明：管理员可以增加新员工。进入页面后选择员工身份、生成默认密码、填写工资与奖金。添加完成后系统给出新员工的账号信息。

员工管理用例图如下：



员工管理时序图如下：



3.1.2.2 菜品管理

(1) 菜品信息展示

描述：以列表形式展示菜品信息。

说明：以分页的形式显示菜品列表，并且支持搜索功能。页面拥有上一页、下一页、首页、末页功能。界面中每个菜品应有单独的展示区域，卡片中展示菜品概要信息，将单页数据列表显示在界面上，页面中显示分页导航链接。搜索区域支持菜品编号与菜品姓名搜索，输入合适的编号后，页面展示相应菜品，如果没有此菜品则显示零个菜品。

(2) 修改菜品信息

描述：管理员可以修改菜品的名称、分组和价格

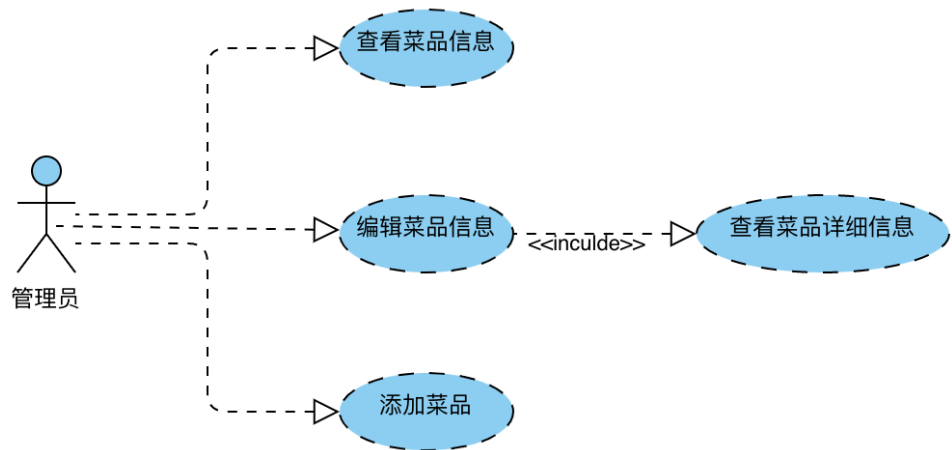
说明：修改菜品信息时，菜品编号、图像不可更改，菜品的名称、分组、价格为必填信息。

(3) 添加菜品

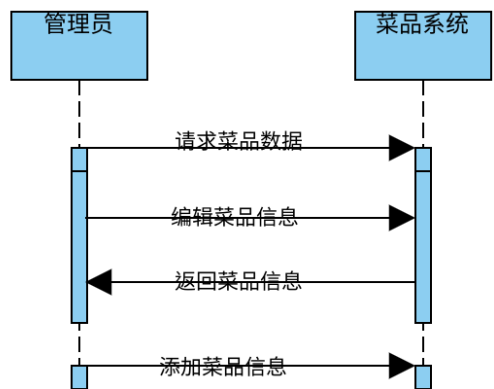
描述：添加新菜品。

说明：进入页面后选择菜品分组、详细信息默认为空、填写菜品名称与价格。
添加完成后系统提示添加新菜品成功。

菜品管理用例图如下：



菜品管理时序图如下：



3.1.2.3 订单管理

(1) 展示正在进行的订单

描述：以列表形式展示正在进行中的订单，支持搜索和分页功能。

说明：以分页的形式显示订单列表，并且支持搜索功能。页面拥有上一页、下一页、首页、末页功能。界面中每个订单应有单独的展示区域，卡片中展示订单概要信息，将单页数据列表显示在界面上，页面中显示分页导航链接。搜索区域支持订单编号与订单桌号搜索，输入合适的编号后，页面展示相应订单，如果没有此订单则显示零个订单。

(2) 订单操作

描述：对订单进行免单和删除操作。

说明：管理员可以查看特定订单的详细情况，如订单编号、订单桌号、订单内容、提交订单的服务员姓名与账号、订单状态、订单数额。管理员在此订单详情页可以进行对订单的免单与删除操作，在提示信息得到确认后完成相应的操作。否则继续显示订单详情页面。

(3) 查看历史订单

描述：以列表形式分页展示所有订单

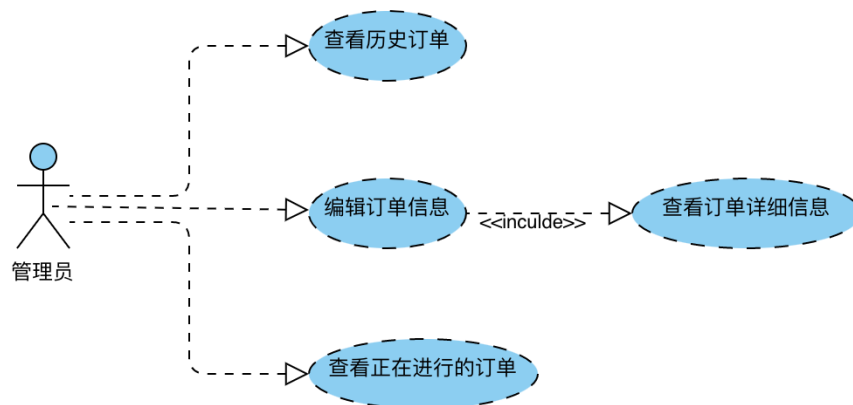
说明：管理员可以查看本店的历史经营数据。将默认所有的订单数据分页展示在页面中，餐厅管理员也可以进行搜索以获取相应的历史订单数据。同时页面中对完成订单的数据做出必要的统计，如各菜品的销售量与服务员的完成订单数量。

(4) 订单数据分析

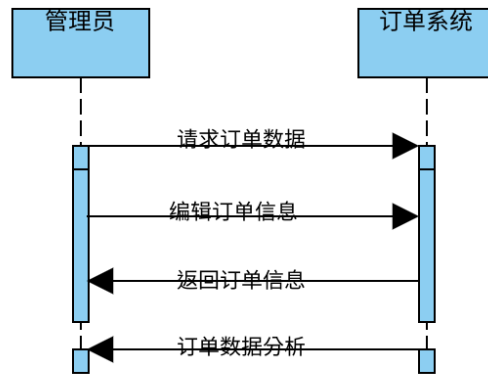
描述：管理员可以查看餐厅的订单数据，并且使用图表等形式直观展示。

说明：将本店服务员销售排行榜以及菜品排行榜用数据可视化的方式展示在web 页面当中。

订单系统用例图如下：



订单系统时序图如下



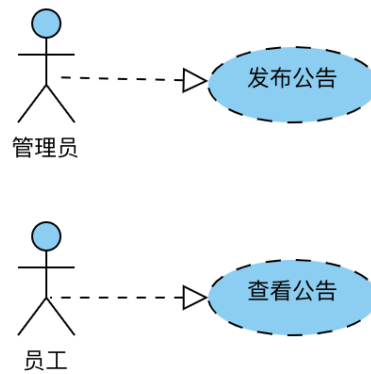
3.1.2.4 公告系统

(1) 发布公告

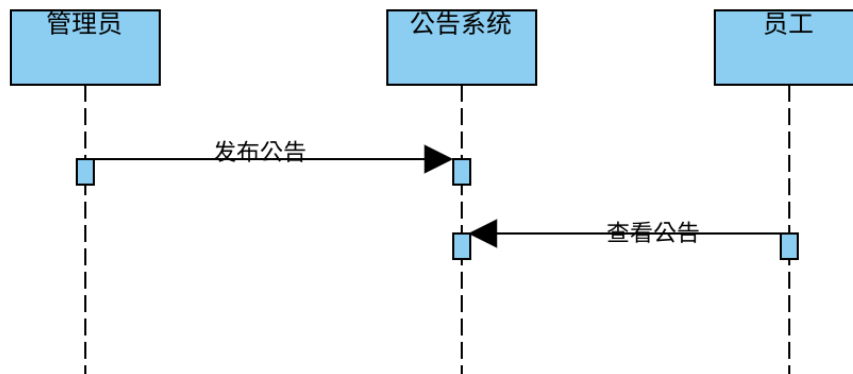
描述：管理员可以发布公告给员工。

说明：管理员可以编辑公告，编辑完成后发布公告，系统中除了管理员之外均可收到此公告。公告内容必填，如果点击发布时没有内容则给出提示；公告的标题默认为“标题”。

公告系统用例图如下：



公告系统时序图如下：

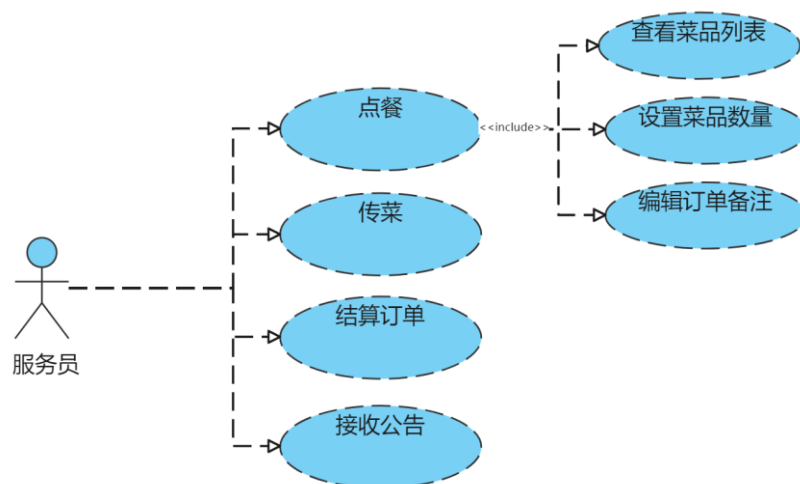


3.2 服务员系统功能设计

3.2.1 模块描述

服务员功能板块是智能订餐系统中涉及较多业务逻辑处理的板块。服务员负责为顾客点单、传菜、结算订单。服务员扫描桌面二维码后自动进入点餐界面，下单每个菜品的数量，编写备注后即可自动生成订单提交；服务员可以实时接收传菜消息，后厨完成菜品后该传菜消息由系统自动发送；但顾客就餐完成后，服务员负责为顾客结算订单；同时，服务员能够接收餐厅管理员发布的近期公告。本小节将通过 UML 图和接口描述，说明服务员功能的具体设计和实现细节。

服务员功能模块的用例图如下：

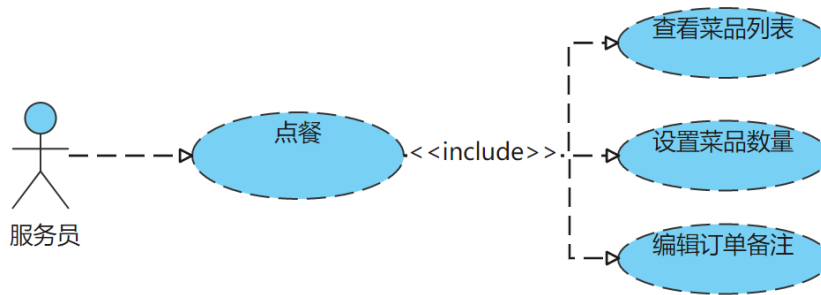


3.2.2 点餐系统

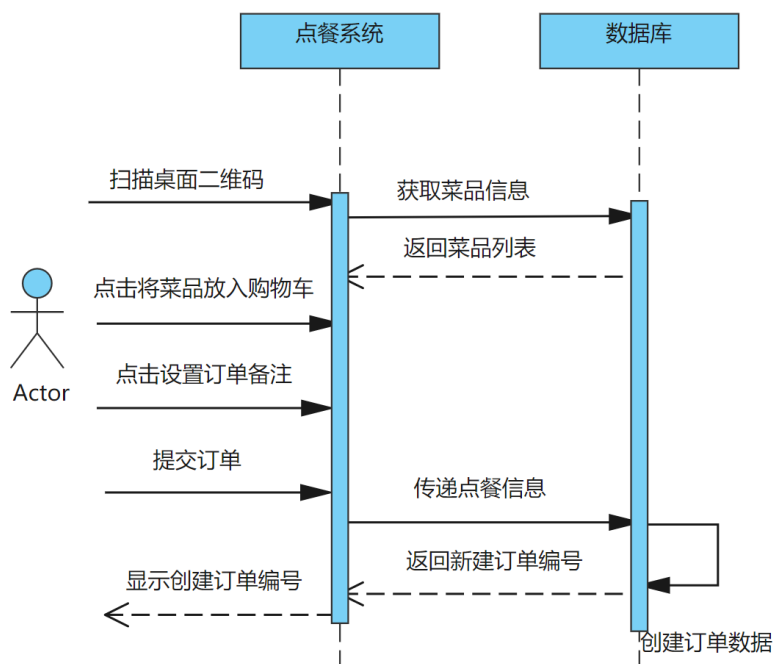
描述：服务员为顾客点单。

说明：服务员可以通过扫描桌面二维码进入点单界面，也可以点击进入点餐界面并手动设置订单桌号。点餐时页面将分页进行菜品展示，服务员点击菜品数量的加号和减号进行菜品下单操作。服务员完成菜品点餐后，可以为订单添加口味备注等信息，点餐提交后自动传送给后厨系统。

服务员点餐系统用例图如下：



服务员点餐时序图如下：



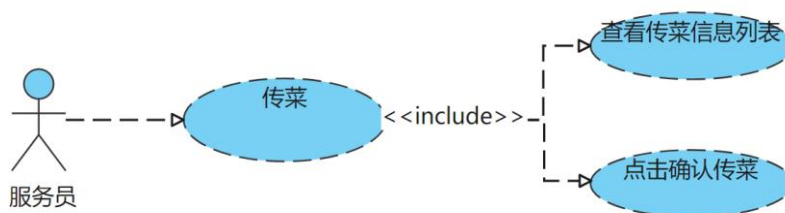
3.2.3 传菜系统

描述：服务员到厨房中端菜传至相应桌号位置。

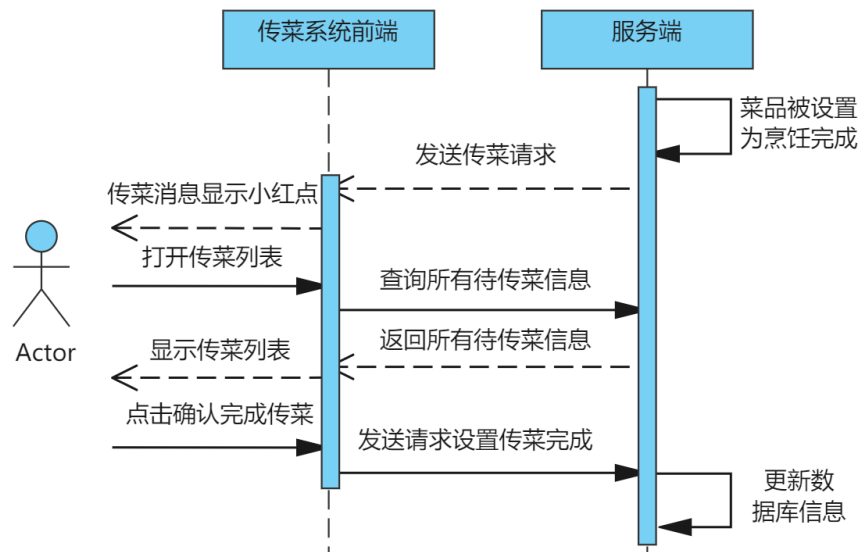
说明：服务员为顾客点单后，系统将该订单分配给该服务员负责传菜、结算。

当该订单下的某一道菜品烹饪完成后，系统自动向服务员发送传菜消息提醒服务员传菜。

服务员传菜系统用例图如下：



服务员传菜系统时序图如下：

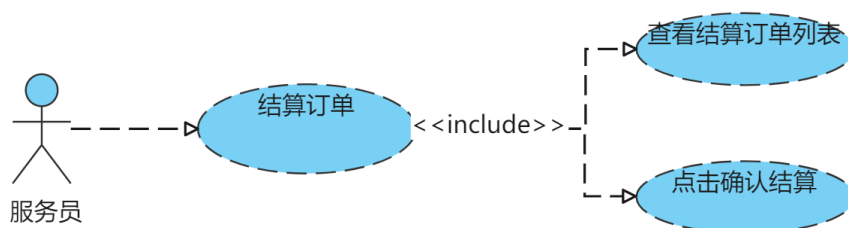


3.2.4 结算订单系统

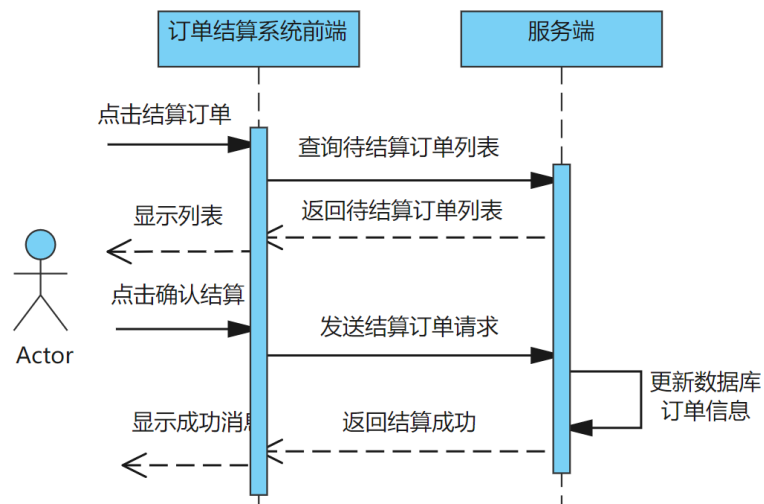
描述：服务员为顾客结算订单。

说明：服务员为顾客点单后，系统将该订单分配给该服务员负责传菜、结算。当该订单的每一道菜品都烹饪完毕且完成传菜后，该订单即可支付，服务员即可等待用户支付。支付完成后服务员点击已结算订单，则该笔订单视为已完成。

服务员订单结算系统用例图如下：



服务员订单阶段系统时序图如下：

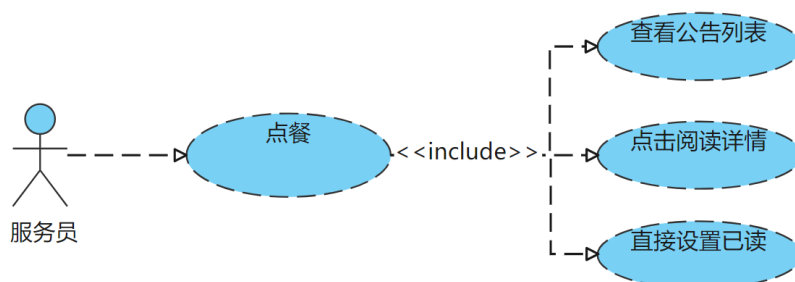


3.2.5 公告系统

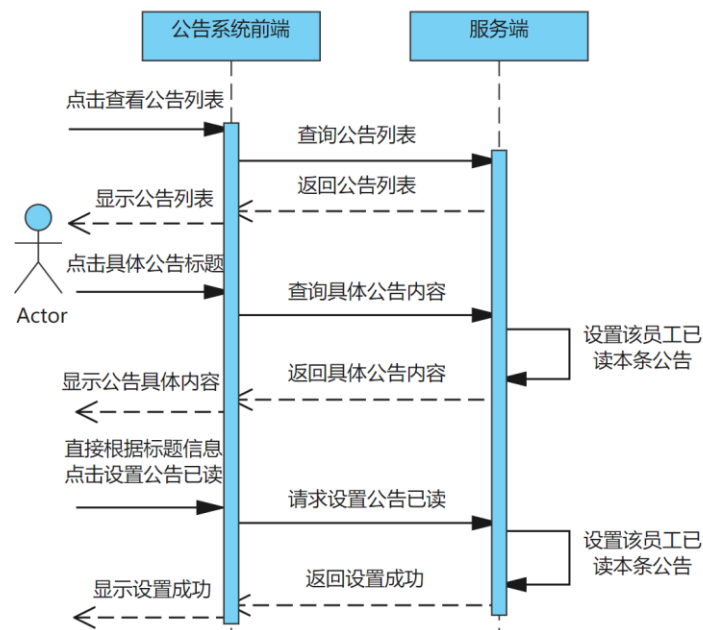
描述：服务员可以接收管理员发布的公告信息。

说明：管理员在系统中发布公告后，服务员即可接收查看。服务员查看信息后该公告自动标记为已读；服务员也可以查看公告详情，直接在公告列表中通过阅读标题将该信息设置为已读。

服务员公告系统用例图如下：



服务员公告系统时序图如下：



3.3 后厨系统功能设计

3.2.1 模块描述

后厨是美食的生产地，为后厨赋能相当于从源头提升餐饮服务质量，因此，后厨系统模块起着至关重要的作用。本模块具体分为两个子部分，分别是后厨菜品烹饪确认功能和后厨查看公告功能。后厨菜品烹饪确认功能主要是针对厨师这一群体，我们拟采用语音识别等创新型功能提升厨师工作效率，使得整体烹饪进展可视化，方便管理。后厨查看公告功能同样服务于管理员、厨师、服务员多个群体，将由管理员发布公告，厨师、服务员查看相应公告，然后进行相关工作，这项功能方便了餐厅前后台的沟通，使得沟通可以追溯。

3.2.2 功能模块

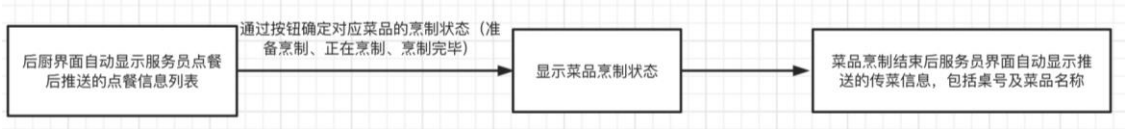
3.2.2.1 后厨菜品烹饪确认功能

后厨界面自动获取服务员推送的点餐信息列表，通过按钮或语音识别或开放接口功能确定对应菜品的烹制状态（准备烹制、正在烹制、烹制完毕）。

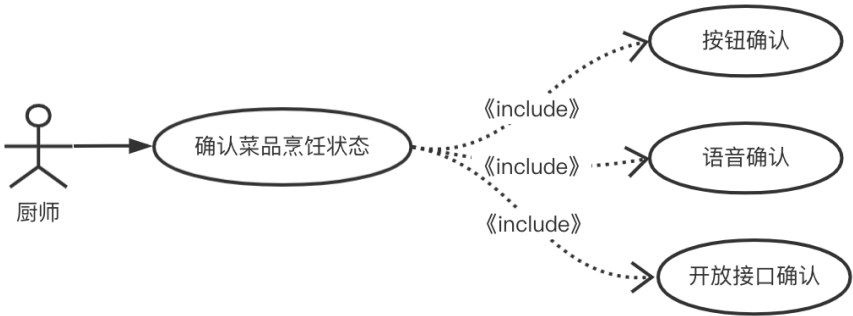
后厨界面自动显示服务员点餐后推送的点餐信息列表，列表中包括菜品名称、菜品对应桌号、菜品数量，列表后显示菜品烹制状态按钮，可以通过按钮改变并显示菜品烹制状态。

处理：利用语音关键字识别改变烹制状态、烹制结束后向服务员界面推送菜品烹制成功消息

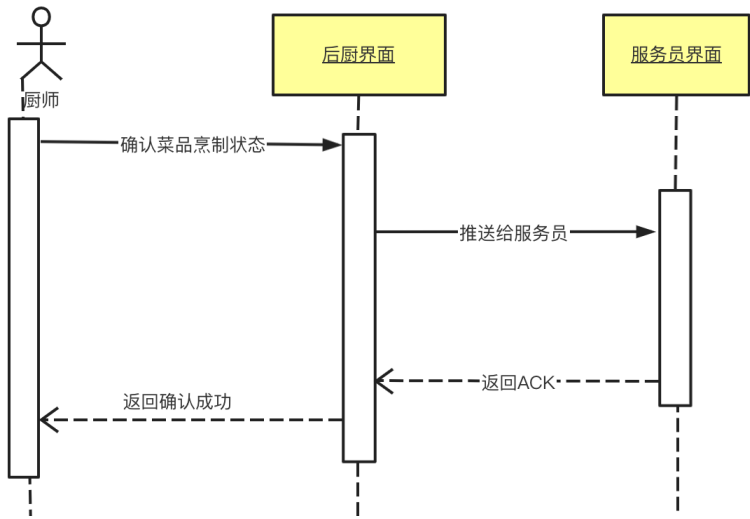
输出：确定菜品烹制状态按钮颜色：等待烹制：绿色，正在烹制：红色，烹制完毕：菜品信息及按钮从界面中消失；菜品烹制结束后服务员界面自动显示推送的传菜信息，包括桌号及菜品名称。



后厨菜品烹饪确认用例图如下：



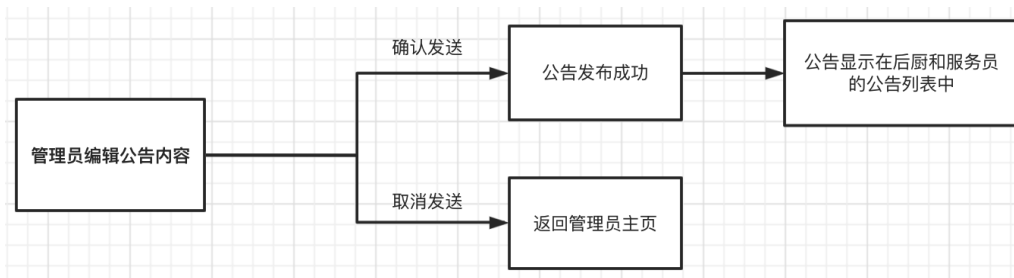
后厨菜品烹饪确认时序图如下：



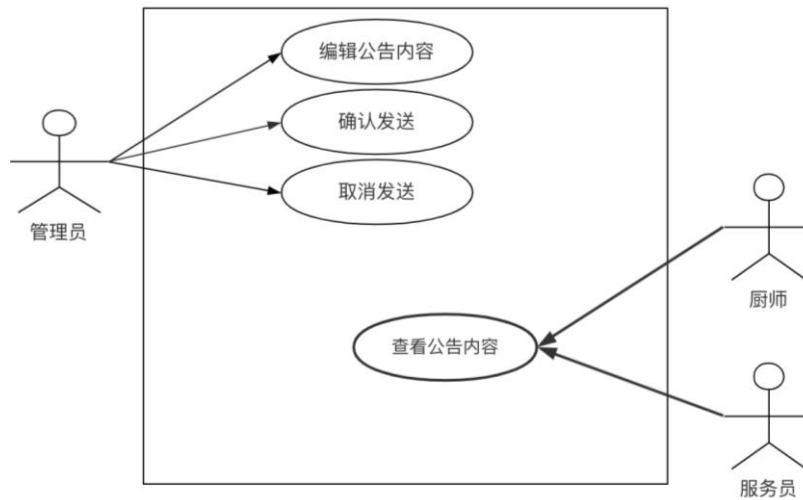
2.5.5 后厨查看公告功能

后厨人员和点餐服务员界面自动显示管理员发布的实时公告。由管理员编辑公告内容，编辑完成后可以确认发送或者取消发送，发送成功后，相应公告会出现在后厨和服务员的公告列表，后厨、服务员可以根据公告内容进行相关工作，使得整个沟通流程更方便、更智能化。

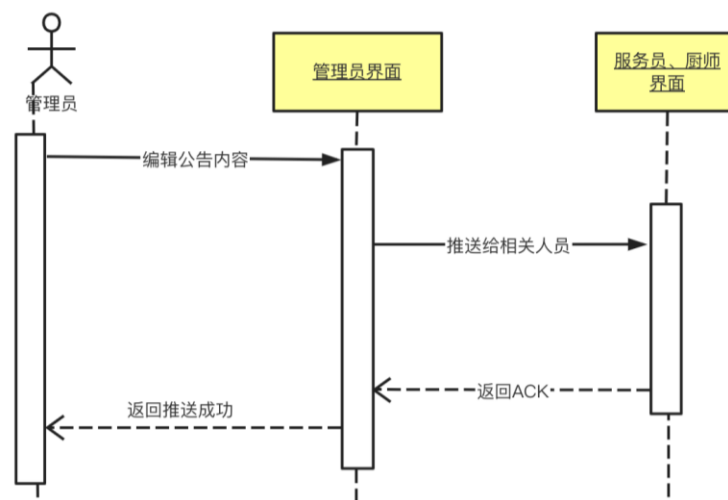
- 1) 输入：管理员界面输入的公告
- 2) 输出：将实时公告显示在公告界面上



公告功能用例图：



公告功能时序图：



4. 界面设计

4.1 界面设计概述

用户界面，也称 UI（User Interface），是人与程序互动的重要部分。因此，一个好的界面至关重要，界面是否友好，是否易用都是我们需要考虑的范畴。一

一个好的 UI 设计往往能第一时间吸引用户的眼球，激发用户的兴趣。

设计原则：

- 易用性

一个好的界面需要易用，因为程序做出来就是让用户使用的，如果设计出来的界面不能很好的供用户使用，不利于用户使用那会在第一时间失去用户。因此我们需要掌握好界面的易用性。另外，我们的软件起始意图是为了高校教育使用，因此易用性显得至关重要。

- 美观与协调性

一个好的界面需要具备大小适合美学观点，感觉舒适和协调，能够吸引用户的眼球等特点，这有这样才能留住用户。

- 设置向导

如果用户使用了一个功能后，不知道如何做下一个，他们就会放弃。所以需要最好的方式来引导用户如何使用软件。

- 合理性

屏幕对角线相交的位置是用户直视的地方，正上方四分之一处为易吸引用户注意力的位置，因此我们在设置界面时也需要注意界面的合理性。

- 借鉴其他好的软件

多了解同类软件的界面，并加以分析与了解，直到能够区别好的用户界面与差的用户界面。但是还需要在模仿别人的界面的同时有所创新。

- 独特性

如果一味的遵循业界的界面标准，则会丧失自己的个性，因此在符合以上规范的情况下有自己独特的标准这样才能很好的吸引用户。

- 容错性

好的界面需要有一定的容错性，尽量使每个动作都很容易可逆这样才能够让用户更多的去探索应用程序，尽量能够在一定程度上可以警告用户那些潜在的负面影响。

本系统采取以下设计理念：

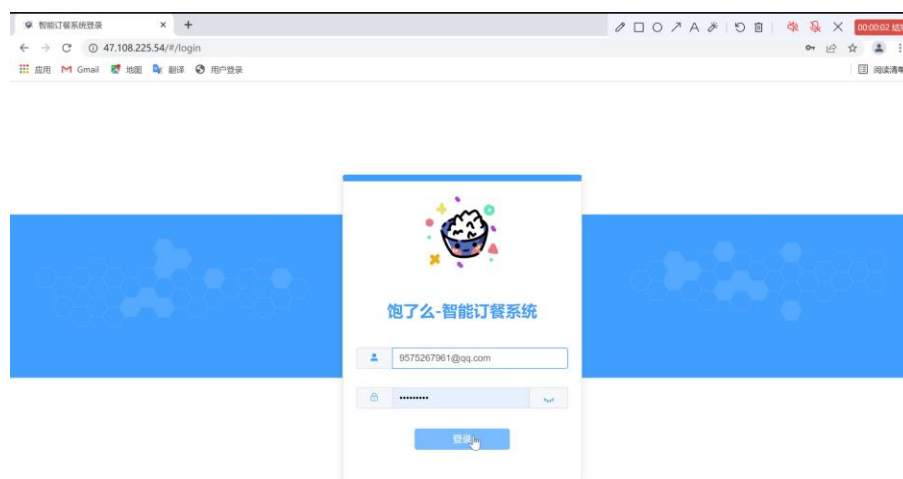
- 简洁且美观
- 易于用户操作

- 界面合理易用
- 少量简洁的变化
- 具有独特性

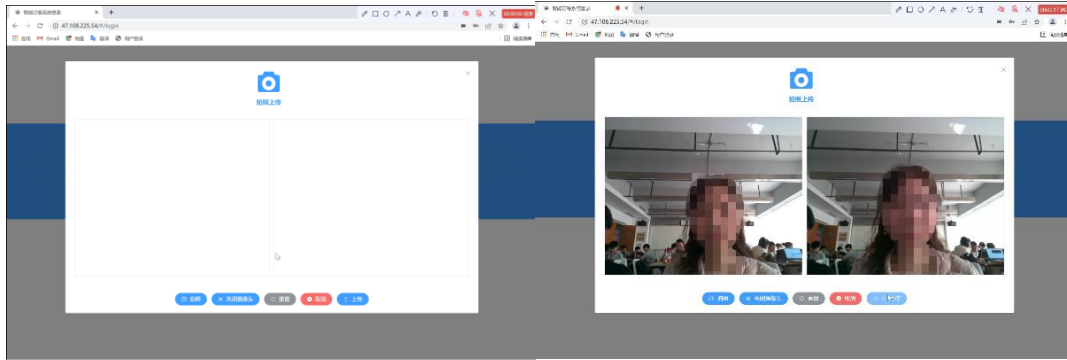
4.2 餐厅管理员界面设计

4.2.1 登录界面设计

在登录界面，用户使用两个输入框进行用户名和密码的输入，并且两个文本框前有相应的图标暗示其功能。在密码框后提供显示密码/隐藏密码的按钮，便于用户在输入密码时确认自己的密码输入是否正确。在用户输入完成后，点击下方“登录”按钮，即可进行用户人脸识别验证。在登录卡片中使用图片和蓝色的字体显示订餐系统的 logo 和名称。名称字体的颜色和卡片顶部以及背景条带的颜色一致，使得整个界面元素简洁但联系紧密。



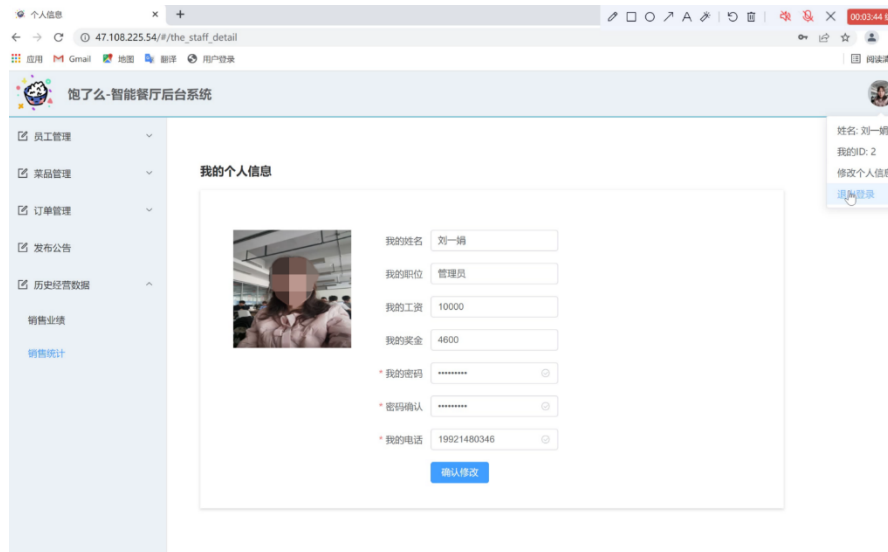
用户验证功能我们使用 Modal 弹出框的形式进行人脸验证。弹出框的正上方提示验证方式为“拍照上传”。在中部有两个矩形框，左侧显示摄像头的实时画面，右侧显示点击“拍照”按钮后拍摄下的照片。在底部有“拍照”、“关闭摄像头”、“重置”、“取消”和“上传”五个按钮，并都有相应的图标进行提示。在用户完成拍摄后，点击提交按钮，系统将对用户输入的信息进行验证。



4.2.2 个人信息界面

个人信息界面和本部分以下的界面一样，在顶部左侧显示系统名称和 logo，右侧显示登录用户头像，并且点击头像显示相关信息和退出登录按钮；在左下侧显示导航栏，右下侧显示相应页面。这样设计使得所有界面保持高度的一致性，并且较为明显的导航栏能够便于用户进行页面跳转。

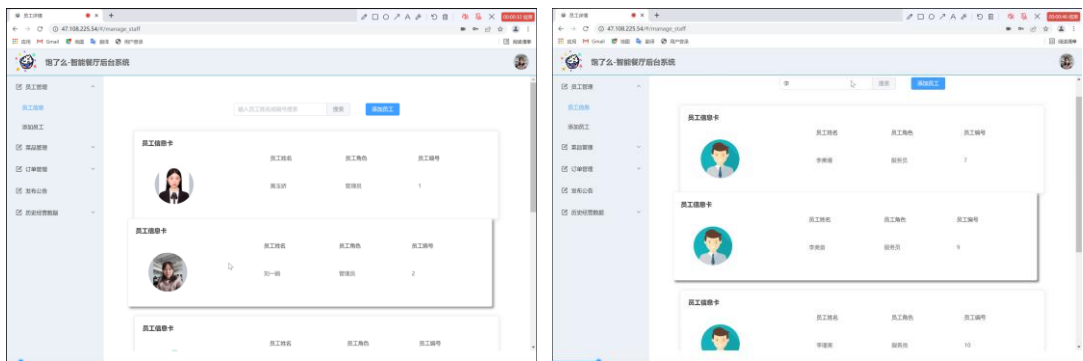
个人信息界面在左侧显示用户的照片，在右侧显示相应的信息。用户能够通过点击“确认修改”按钮对个人信息进行修改和提交。



4.2.3 员工信息界面

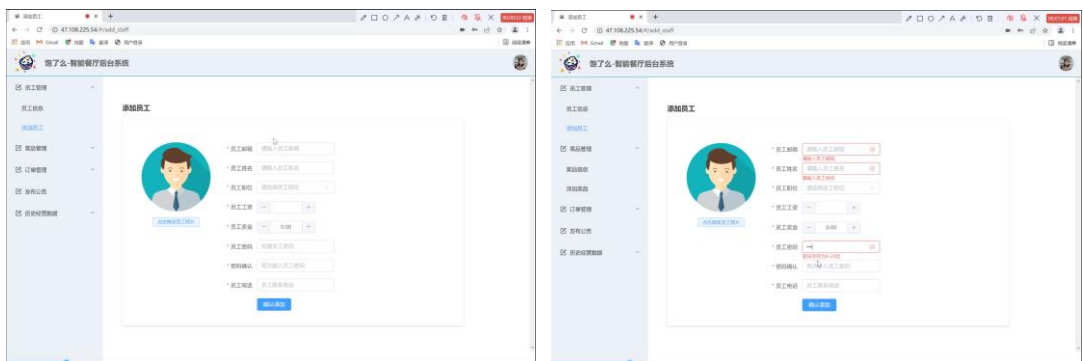
员工信息界面中，支持按照员工信息名和变化进行搜索，使用卡片的方式对员工信息进行显示，包括员工照片、员工姓名、员工角色和员工编号。对于鼠标指向的卡片，能够进行浮动显示进行强调。并且通过分页方式对信息进行分页显示，以便于用户的查看。点击“添加员工”按钮，能够跳转到添加员工界面。虽然与左侧导航栏的添加员工按钮重复，但在本页面中再次进行设置，使得页面跳转更加符合用户的心理预期。卡片是可点击的，跳转到员工信息详情页，支持对

员工信息的查看和修改。



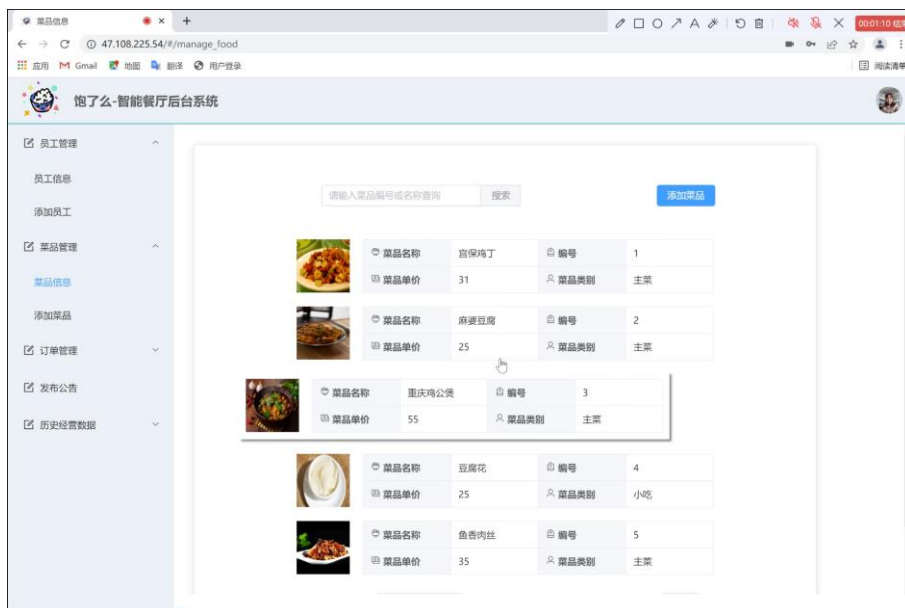
4.2.4 添加员工界面

在添加员工界面中，我们进行了表单的设计，使用文本框、数字框、密码框和下拉选择框对各种信息进行输入。对必须填写的信息用红色的“*”号进行标注。同时对于不符合要求的信息进行提醒。左侧可以修改员工的照片，在不上传照片的情况下，系统提供了默认的头像，便于管理员快速进行员工信息的添加。

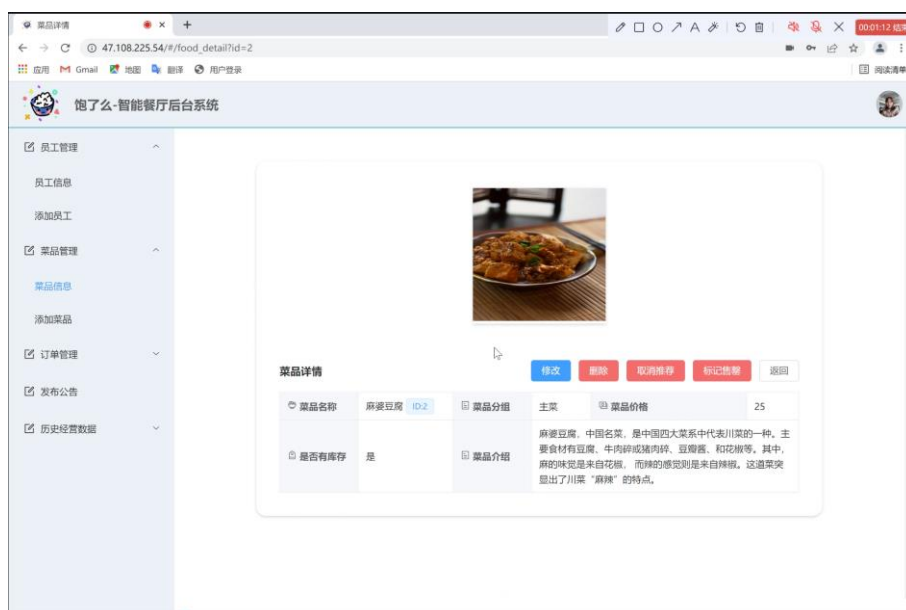


4.2.5 菜品信息界面

在菜品信息界面中，使用卡片形式对各个菜品的信息进行显示，当鼠标处于卡片上时，卡片进行浮动显示。在页面上方提供搜索框，可以对菜品进行搜索。与员工页面同理，我们在菜品信息页面也添加了“添加菜品”的导航按钮，同时对菜品信息进行了分页显示。而每一张卡片都是可点击的，导航到每一道菜品的详细信息和信息修改页面。



在菜品详情页中，显示菜品的信息和库存情况，并在上方显示“修改”、“删除”、“取消推荐”、“标记售罄”和“返回”按钮，分别对应相关功能。



4.2.6 正在进行的订单界面

在此页面，我们通过表格的形式对订单信息进行显示，其中包括“订单编号”、“订单桌号”、“订单总额”、“服务员姓名”和操作按钮等信息和操作方式。并且通过分页的方式对信息进行显示，支持表格内信息进行搜索。在进行相应操作时，我们通过弹窗的形式提醒用户该操作的后果。

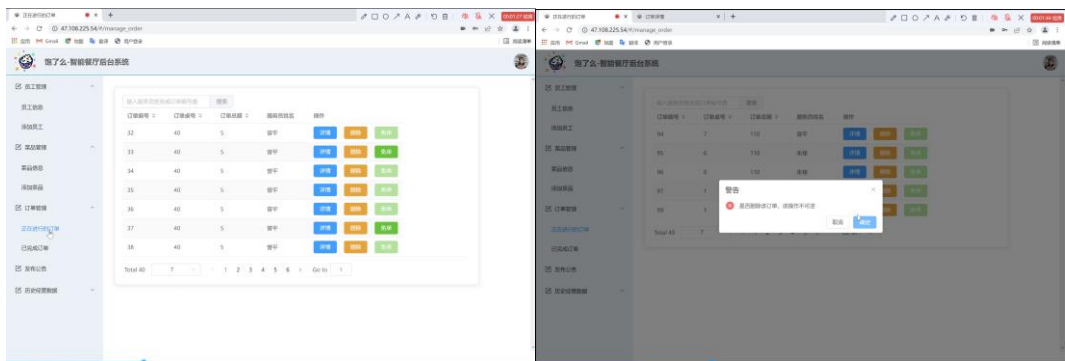
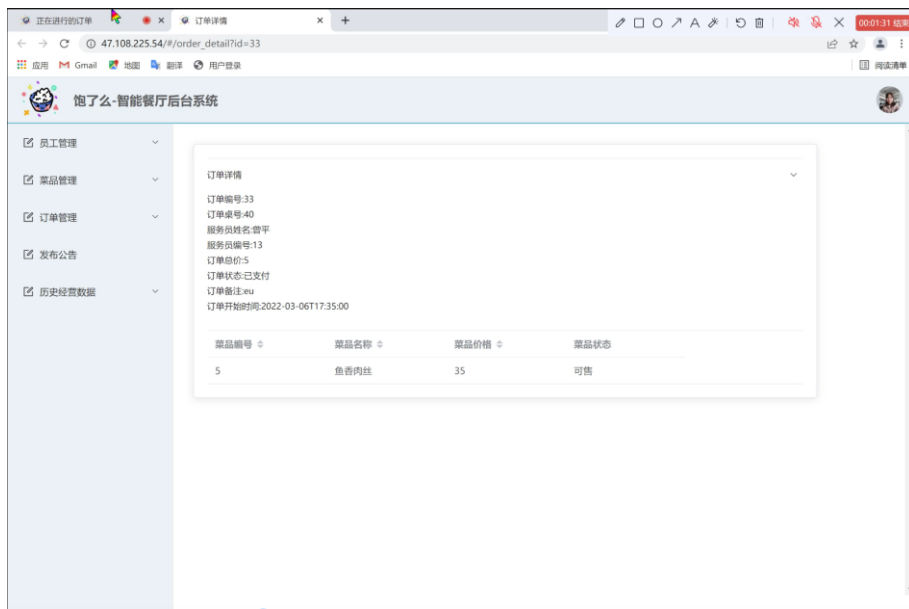


图 1 正在进行的订单界面

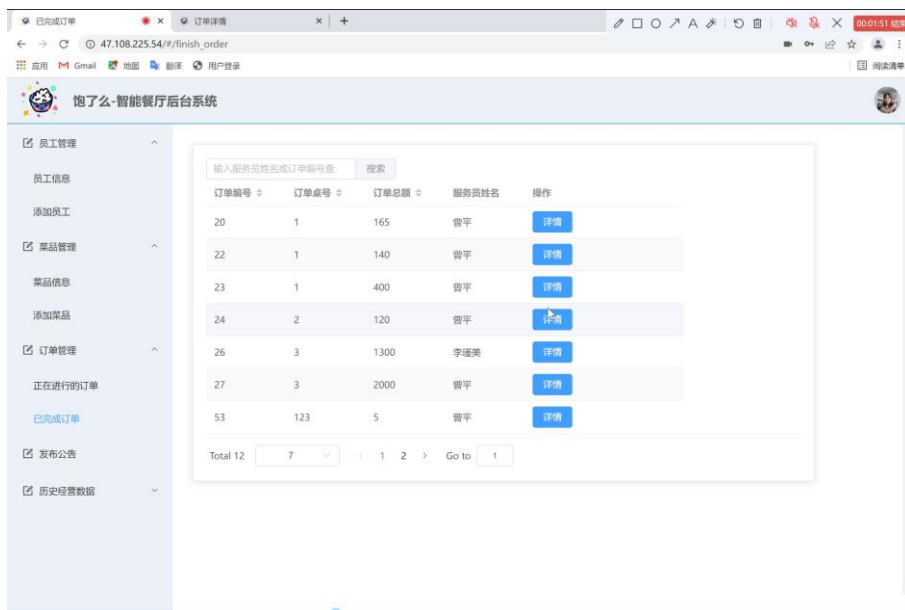
4.2.7 订单详情界面

在“正在进行的订单”界面的表格中点击“详情”按钮，就可以跳转到相应菜品的详情界面。在界面上方显示订单的编号、桌号、服务员信息、总价、支付状态等订单基础信息。并在下方以表格的形式显示订单的菜品信息。



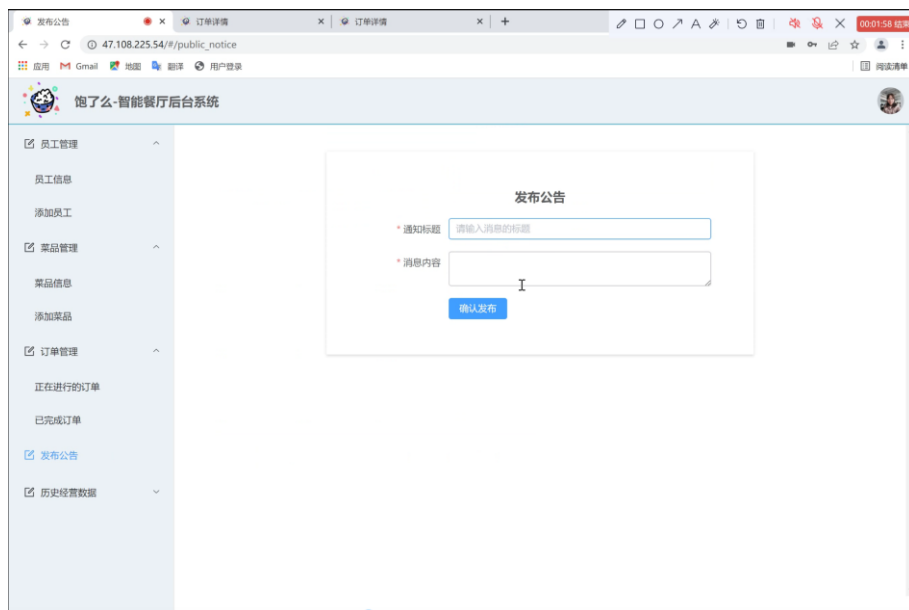
4.2.8 已完成订单界面

与“正在进行的订单”界面一样，我们通过表格分页的形式对订单信息进行展示，并支持表格内信息搜索。点击“详情”按钮，同样能跳转到“订单详情界面”。



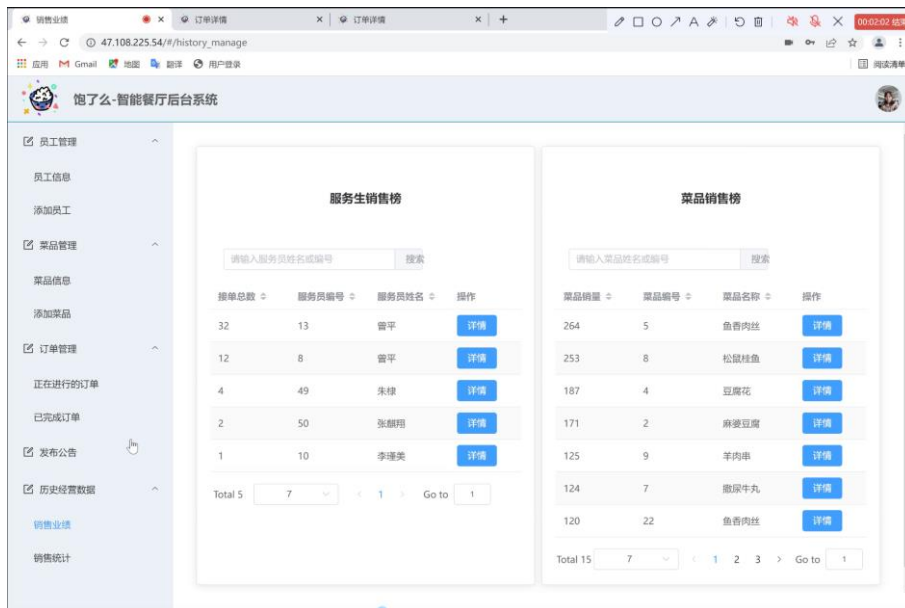
4.2.9 发布公告界面

在发布公告界面中，提供文本框，便于用户进行通知标题和消息内容的输入，并用红色“*”号对必须填写的信息进行提示。在下方有“确认发布”按钮，点击后进行公告的发布。



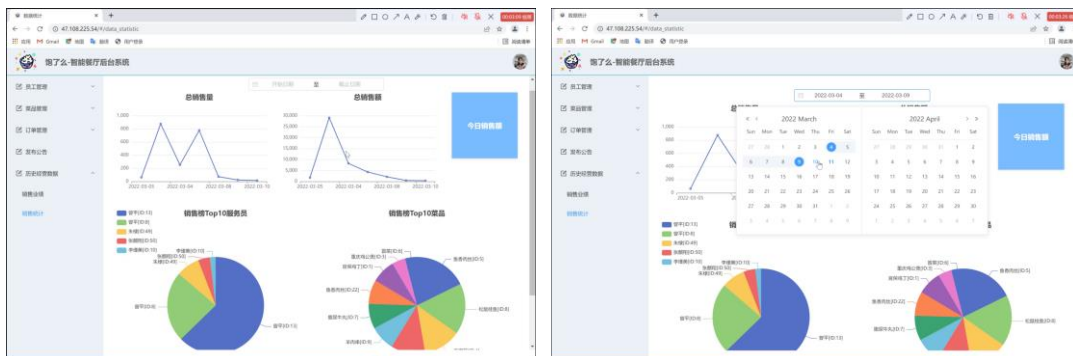
4.2.10 销售业绩界面

与“正在进行的订单”界面类似，我们通过表格分页的模式对服务生销售榜和菜品销售榜进行显示，支持表格内信息搜索。



4.2.11 销售统计界面

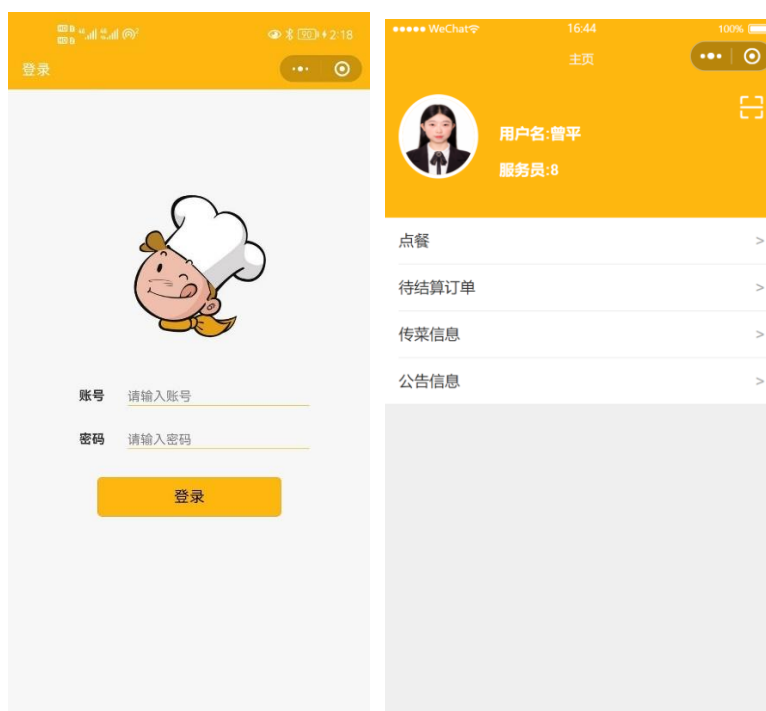
在该界面中，我们使用 echarts 对相关数据信息进行直观的显示，包括总销售量与时间的折线图、总销售额与时间的折线图、服务员销售额饼图、菜品销售额饼图。并且能够设置销售统计的起始时间和终止时间。



4.3 服务员系统界面设计

4.3.1 主界面

为了帮助服务员快速进行功能菜单选择，服务员系统主界面采用板块化设计显示功能列表。主页显示服务员基本个人信息，右上角有扫码点餐功能。主页功能板块中包含服务员的常用重要功能：点餐、结算订单、查看传菜信息、查看后厨信息、查看公告。



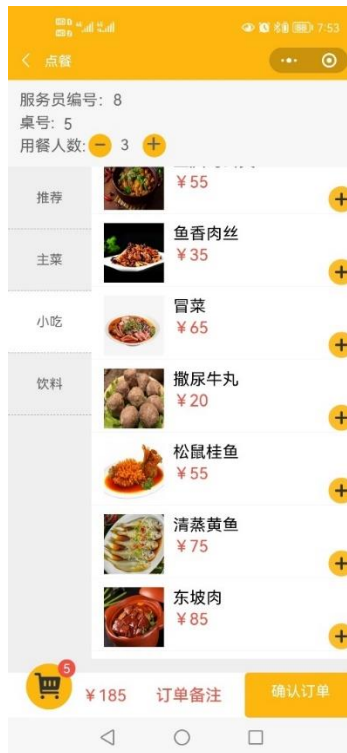
4.3.2 点餐界面

点餐界面顶部显示当前基本信息：服务员编号、桌号、用餐人数。服务员编号系统自动填充不可更改，桌号通过扫码自动获得，服务员可以点击用餐人数的加号和减号按钮设置用餐人数。

点餐界面右侧有菜品类别导航栏，能够帮助服务员快速寻找不同类别的菜品。默认显示导航栏中的推荐菜品，还可以点击跳转到主菜、小吃、饮料等其他类别。

点餐界面主要部分显示菜品列表，服务员可以通过点击菜品旁边的加号将菜品添加到购物车，也可以点击减号移除菜品。当页面显示菜品售罄时，不能对菜品进行任何操作。

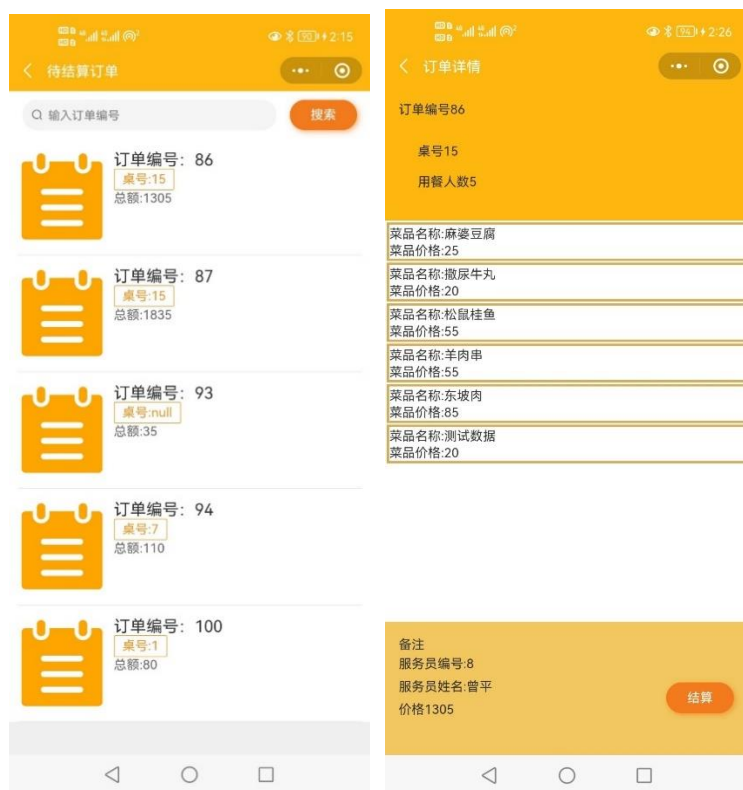
点餐界面底部是购物车图标，点击后可以查看购物车中全部菜品及相应点餐数量。点击旁边的“订单备注”按钮可以根据顾客为该订单添加相应备注；点击“确认点餐”按钮后系统自动提交订单，将订单推送至后厨等待烹饪。



4.3.3 待结算订单界面

待结算订单界面显示该服务员负责的所有待结算订单，顶部设置有搜索框，服务员可以根据订单编号进行搜索。订单列表中显示订单的基本信息：订单编号、桌号、总额，便于服务员快速查看和结算订单。

点击某个结算订单后跳转到详情界面，可以查看订单相关菜品、用餐人数等信息，点击底部的结算按钮并确定后完成该笔订单的结算。



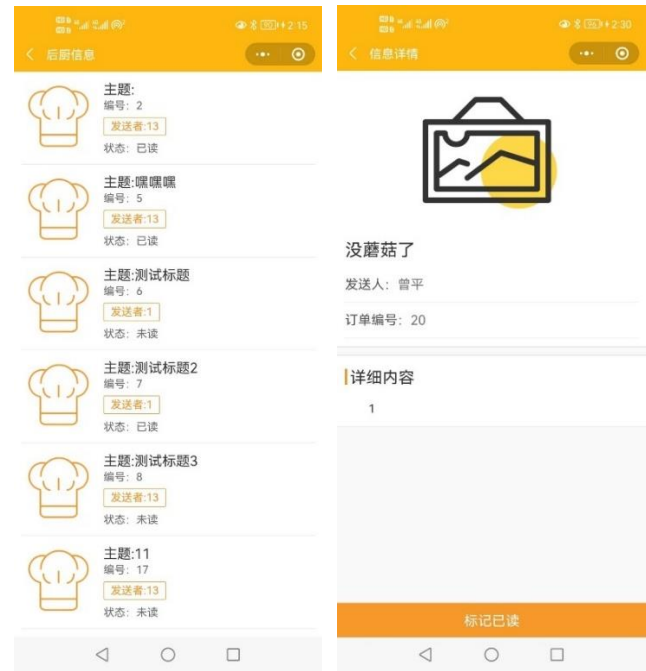
4.3.4 传菜信息界面

传菜信息界面显示该服务员负责的订单中所有待传菜列表，包括菜品图片、菜品编号、菜品名称，以及该菜品所属的订单编号信息。服务员点击某个菜品项的“传菜”按钮并确认后，该菜品视为已完成传菜。



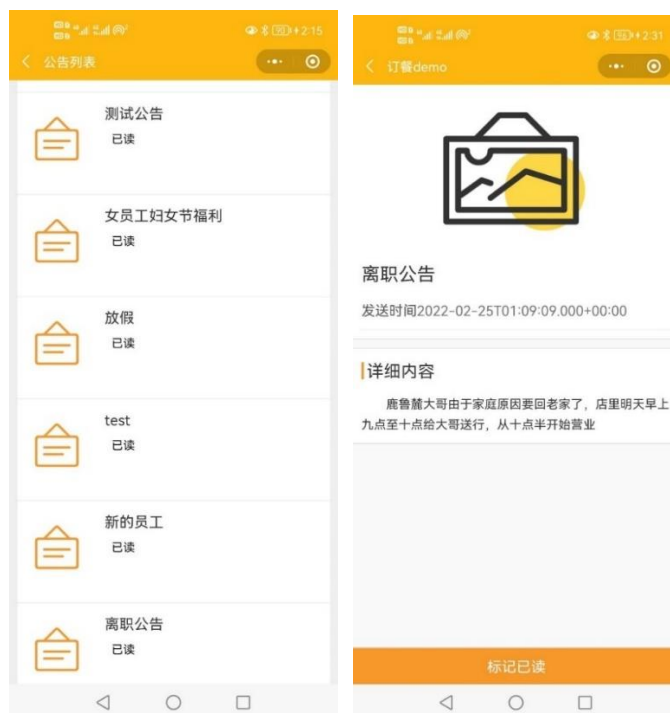
4.3.5 后厨信息界面

后厨信息界面显示所有后厨发送给服务员的信息，信息项旁边有厨师帽图片，可以在列表中看到发送消息标题、发送人的基本信息。点击某项具体后厨信息后跳转到消息具体界面，可以查看消息的具体内容，可以点击底部“确认已读”按钮标记该消息已读。



4.3.6 公告信息界面

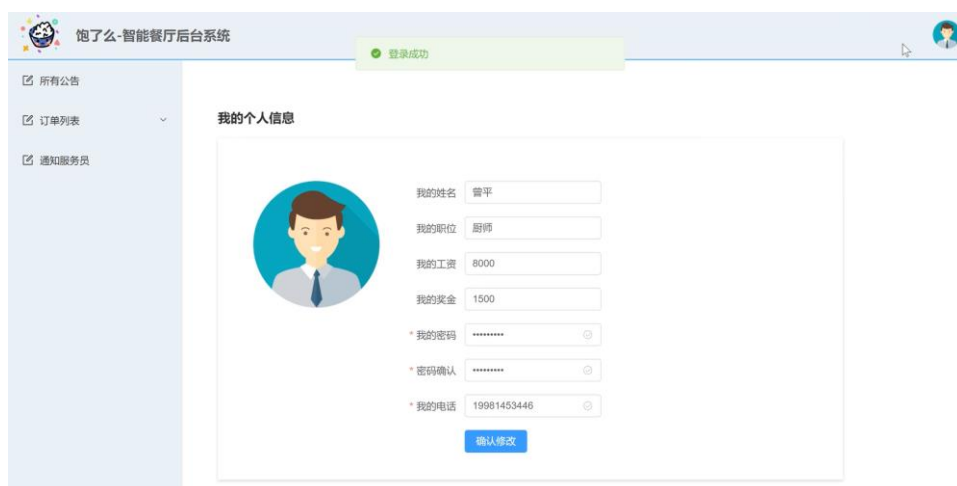
公告信息界面显示管理员发送的所有公告，信息项旁边有公告图片，可以在列表中查看发送公告的标题。点击某项具体公告信息后跳转到信息详情界面，可以查看公告的具体内容，可以点击底部“确认已读”按钮标记该公告已读。



4.4 后厨系统界面设计

4.4.1 后厨个人信息界面

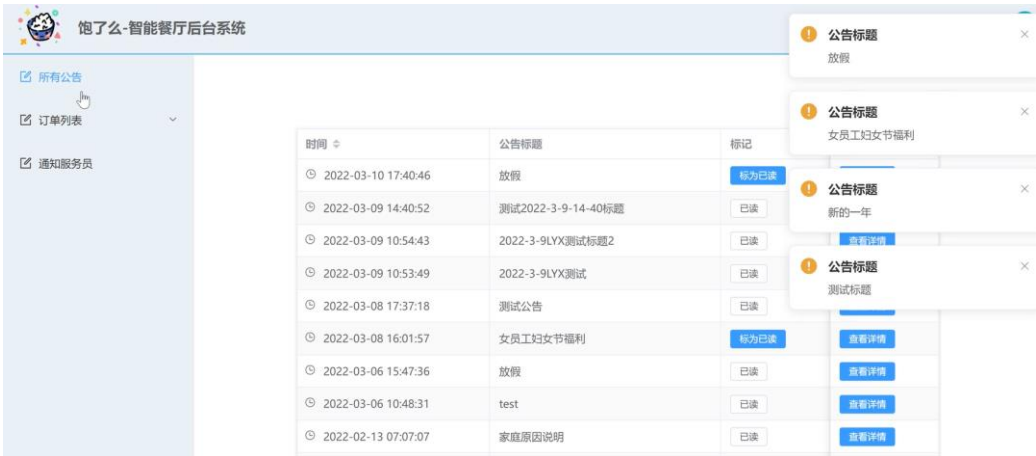
后厨登录后默认显示后厨的个人信息，包括姓名、职位、奖金、密码、电话。后厨可以在个人信息界面修改密码和电话，修改完成后点击确认修改即可提交修改内容。



4.4.2 后厨公告界面

后厨公告界面根据时间显示所有管理员发布的公告列表，包含公告标题、发布时间、已读未读状态的基本信息，近期末读公告会自动弹出提醒。点击具体公告项查看详情按钮后可以查看公告的具体内容，查看后系统自动设置为已读。也

可以不查看公告内容，仅通过点击标题旁的“标为已读”的蓝色按钮，将公告标记为已读。



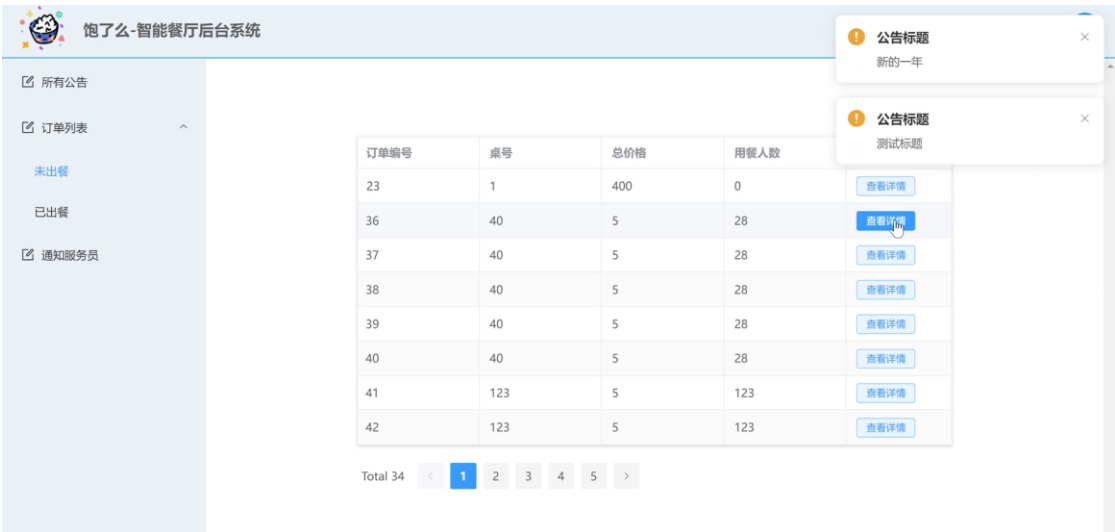
4.4.3 后厨订单列表界面

1) 未出餐订单界面

未出餐订单界面显示所有未出餐订单项，包括订单的编号、桌号、总价格、用餐人数的基本信息。为了便于后厨专注于炒菜，后厨不需要点击屏幕，只需要说出“查看 xx 编号订单详情”，就可以自动打开某个订单的详情。当然也可以通过点击查看详情按钮实现界面跳转。

2) 已出餐订单界面

已出餐订单界面显示所有已出餐订单项，包括订单的编号、桌号、总价格、用餐人数的基本信息。同样地，后厨只需要说出“查看 xx 编号订单详情”，就可以自动打开某个订单的详情。当然也可以通过点击查看详情按钮实现界面跳转。



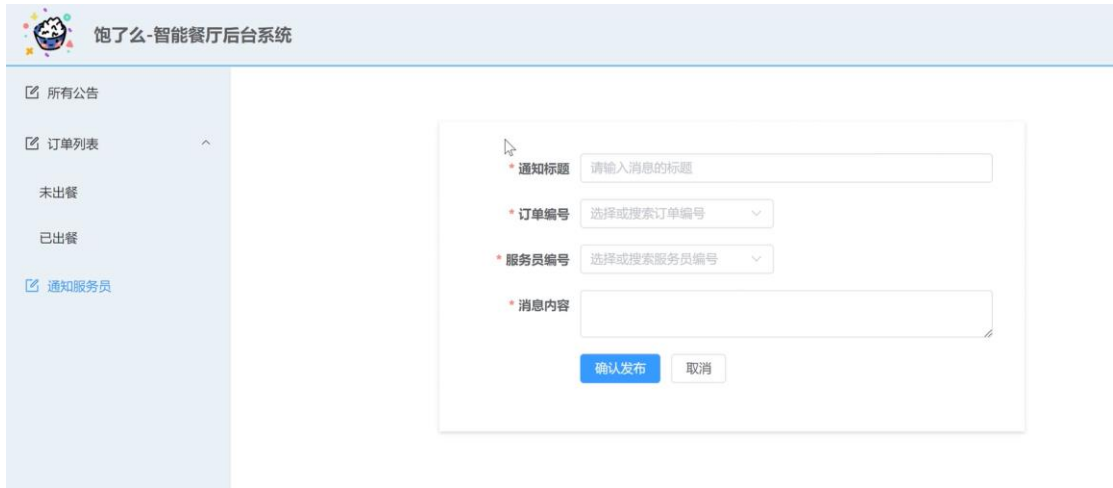
3) 订单界面详情

未出餐订单界面详情中，显示该订单所有的菜品信息，并可以更改菜品状态。当厨师开始烹饪时，需要说出“开始烹饪 xx 菜品”（xx 为菜品名称），系统将自动使用关键字语音识别技术将该菜品的状态由“准备烹饪”修改为“正在烹饪”；当厨师烹饪完成时，需要说出“烹饪完成 xx 菜品”（xx 为菜品名称），系统将自动使用关键字语音识别技术将该菜品的状态由“正在”修改为“烹饪完成”；当然，当厨师方便时，也可以通过点击界面的蓝色按钮也可以完成相应的状态修改。



4.4.4 通知服务员界面

在后厨通知服务员界面呈现消息表单，后厨需要填写消息标题，并指定相关订单编号或者服务员编号任意一项，系统会自动匹配另一项（相应订单的服务员或订单信息）。编辑消息内容后点击“确认发布”即可完成向服务员的消息发送。

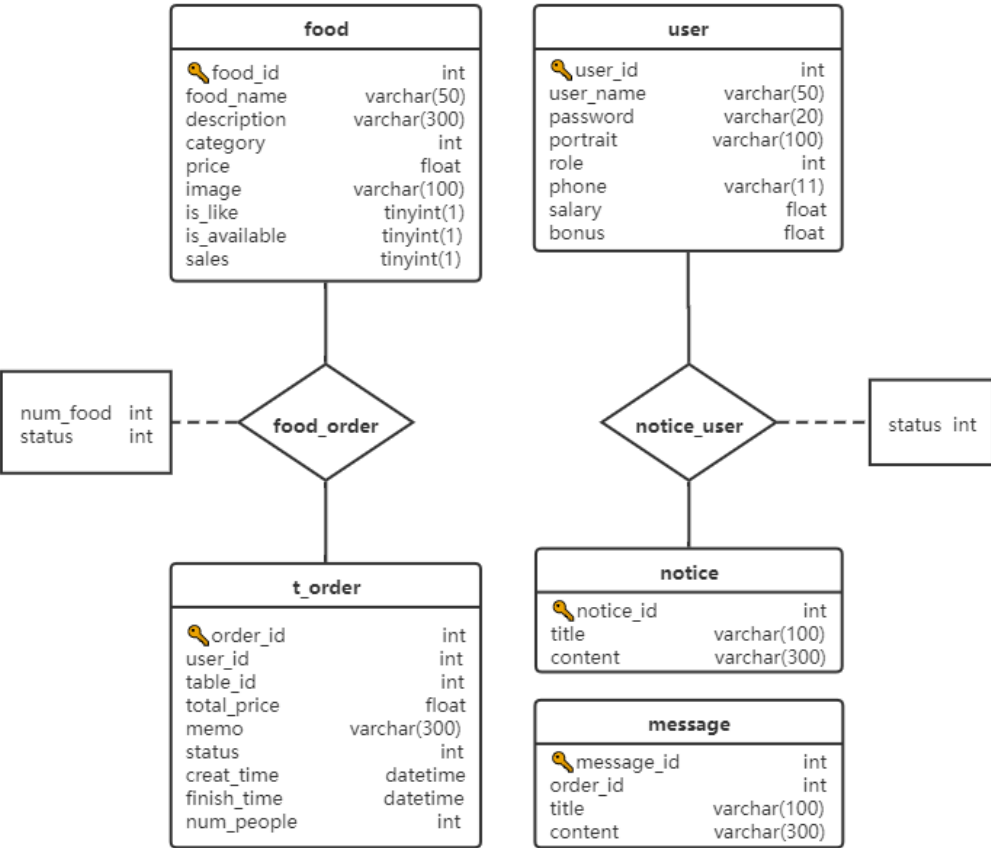


5. 数据库设计

5.1 数据关系设计

本项目数据库整体结构如下 ER 图所示。实体表主要包括菜品表、用户表、

订单表、通知表。其中，订单和菜品有联系表，用于记录某笔订单中点了哪些菜品；通知和用户表也具有联系表，用于记录用户是否已读该通知。



5.2 数据表详细设计

用户表（user）

用户表用于存储管理员、服务员、厨师的基本信息，由于本系统的用户即餐厅的工作人员，因此将三者统称为用户。

字段名	描述	类型
user_id	登录账号【主键】	int
password	登录密码	string
username	用户名	string
portrait	头像图片的地址	string
role	1-管理员，2-服务员，3-厨师	int
phone	电话号码	string

salary	工资	int
bonus	奖金	int

菜品表 (food)

菜品表用于存储主菜、小吃、饮料的基本信息，目前还没有添加套餐的选项。

字段名	描述	类型
food_id	菜品唯一标识符【主键】	int
food_name	菜品名称	string
description	菜品简介描述	string
category	1-主菜, 2-小吃, 3-饮料	int
price	价格	float
image	菜品图片	string
is_like	是否推荐	bool
is_available	是否可提供	bool

订单 (t_order)

订单表用于存储消费者的每一笔订单信息。

字段名	描述	类型
order_id	订单唯一标识符【主键】	int
user_id	提交该订单的服务员 ID【外键】	string
table_id	桌号	string
total_price	订单总价格	float
memo	备注	string
status	订单状态 (0-未支付, 1-排队, 2-做好未支付, 3-已支付, 4-买单)	int
create_time	创建时间	datetime
finish_time	结餐时间	datetime
num_people	用餐人数	int

订单与菜品关联表（food_order）

本表用于将订单和菜品关联起来，帮助厨师和服务员查看每一道餐品的烹饪情况以及每一笔订单的完成情况。

字段名	描述	类型
order_id	order 的 ID 【外键】 【主键】	int
food_id	food 的 ID 【外键】 【主键】	string
num_food	点单数量	int
status	烹饪状态（0-还没炒，1-正在炒，2-炒好了）	int
delivery	传菜状态（0-没传，1-传了）	int

公告（notice）

本表用于存储系统中管理员发布的公告内容。

字段名	描述	类型
notice_id	唯一标识符 【主键】	int
title	公告标题	string
content	公告内容	string
create_time	创建日期	datetime

关联表（notice_user）

本表用于记录每一个公告的阅读状态和发布信息。

字段名	描述	类型
notice_id	公告 ID 【外键】 【主键】	int
user_id	用户 ID 【外键】 【主键】	int
status	0-未读，1-已读	int

消息（message）

本表用于存储本系统中用户发送的信息以及相关的维护信息。

字段名	描述	类型
message_id	主键【主键】	int
order_id	订单 ID【外键】	int
title	标题	string
content	消息	string
create_time	创建日期	datetime
send_user	发送者 ID	int
receive_user	接收者 ID	int
status	状态（0-未读，1-已读）	int

6. 系统防错设计

6.1 AI 识别接口精度设计

在项目中运用到了人脸识别等 AI 识别接口，但识别就面临一个识别错误的问题，其中识别错误的情况有：正确的用户识别不通过、错误的用户识别通过。

在项目中，我们通过调用百度智能云 API 完成人脸识别，接口将返回一个匹配度（0~100%）。我们设定一个阈值，如果返回匹配度超过该阈值则识别通过，否则识别不通过。正确的用户识别不通过的原因是阈值被设置得过高，然而模型识别结果不是完全准确的，导致正确用户的匹配度未能达到阈值，不通过本次人脸识别校验；错误的用户识别不通过的原因是阈值被设置得过低，同时因为模型不完全准确的原因，导致错误用户的匹配度达到该阈值，通过了本次人脸识别校验。

项目小组通过测试发现，如果将阈值设置为 86% 以下，虽然正确用户不通过识别的比例只有 0.01%，错误用户通过识别的比例会增大到 18%；如果将阈值设置为 93% 以上，虽然错误用户通过的比例只有 0.02%，但正确用户不通过的比例会增大到 16.8%。

所以，为了避免项目在人脸识别环节出现错误情况，在本次项目中识别的阈

值被设置为 89%，此时正确用户不通过、错误用户通过的比例都较低，在 0.09% 左右。

6.2 登录设计

为了防止用户未登录就通过 url 访问系统功能，项目对各个功能模块都添加了路由拦截，只有完成该身份验证的用户登录系统后才能访问相应模块的后端接口。

用户登录时会出现不存在用户名、密码和用户名不匹配这两种错误情况，对于每一种错误情况，后端都会返回相应的错误码 code 和错误原因提示信息 msg。

6.3 表单提交设计

为了防止像后端提交格式不正确的表单数据，前端对每个表单数据都做了出错性校验。比如，必填的字段不能为空，邮箱必须符合格式，电话号码必须符合格式，校验密码必须和第一次输入的密码相同等。

表单验证可以放在客户端和服务端，客户端验证实际是直接在已下载到本地的页面中调用脚本来进行验证，它不但能检查用户输入的无效或者错误数据，还能检查用户遗漏的必选项；而服务器端的验证则是将页面提交到服务器处理，服务器上的另一个包含表单的页面先执行对表单的验证，然后再返回相应结果到客户端，它的缺点是每一次验证都要经过服务器，不但消耗时间长，而且会大大增加服务器的负担。在客户端进行验证，能把服务器端的任务分给多个客户端去完成，从而减轻服务器端的压力，让服务器专门做其他更重要的事情。

6.4 出错处理设计

a.采用后备技术，周期性地把磁盘信息记录到磁带上就是对于磁盘媒体的一种后备技术，当系统数据丢失时启用的副本的建立和启动的技术；

b.采用降效技术，使用另一个效率稍低的系统或方法来求得所需结果的某些部分，例如一个自动系统的降效技术可以是手工操作和数据的人工记录；

c.采用恢复及再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行。

7. 部署设计

1. 部署内容

该系统 web 前端使用 nodejs 部署，java 后端和通过 springboot 打包部署，微信小程序前端部署在本地调试。其中，web 前端和 java 后端均部署在阿里云服务器上，以实现餐厅的服务员、后厨人员及餐厅管理员相关功能的使用。

2. 部署范围

面向使用该系统的餐厅服务员、后厨人员及餐厅管理者

3. 部署环境要求

1) 硬件环境：服务器一台，建议配置如下

产品名称	基本配置说明
应用和数据库服务器	Intel® i3@2.0GHz ，2GB 内存，Apache Tomcat，MySQL 8.0.27，256GB 硬盘数量 1 个

2) 软件环境

分类	名称	版本
操作系统	Linux Ubuntu	20.04.13
数据库平台	MySQL	8.0.27
应用平台	Apache Tomcat	6.0
客户端软件	Google Chrome	8.0
	微信	8.0.21
编译程序	Java	1.5.0

3) 网络环境：广域网或餐厅内局域网

4. 部署依赖

- 1) Linux Ubuntu 20.04.13
- 2) MySQL 8.0.27 及系统相关数据库
- 3) Apache Tomcat 6.0
- 4) Java 1.5.0
- 5) Google Chrome 8.0，微信 8.0.21 及应用系统

8. 接口设计

```
title: 点餐系统 v1.0.0
language_tabs:
  - shell: Shell
  - http: HTTP
  - javascript: JavaScript
  - ruby: Ruby
  - python: Python
  - php: PHP
  - java: Java
  - go: Go
toc_footers: []
includes: []
search: true
code_clipboard: true
highlight_theme: darkula
headingLevel: 2
generator: "@tarslib/widdershins v4.0.4"
```

点餐系统

v1.0.0

点餐

GET 进入用户信息修改

GET /userInfo

用户请求进入信息修改界面

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» userData	User	true	none	none
»» userId	integer	true	none	员工编号与登录账号
»» password	string	true	none	none
»» username	string	true	none	无
»» role	integer	true	none	1-管理员, 2-服务员, 3-厨师
»» portrait	string	true	none	图片链接
»» phone	string	true	none	none
»» salary	number	true	none	none
»» bonus	number	true	none	none

点餐/管理员

POST 发布公告

POST /admin/notice

管理员点击确认 可以发布公告

Body 请求参数

title: test
content: hello world

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» title	body	string	true	公告标题

名称	位置	类型	必选	说明
» content	body	string	true	公告内容

返回示例
成功

```
{
  "code": 1,
  "msg": "success"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

点餐/管理员/人员管理

GET 根据员工身份筛选

GET /user/select_by_role

Body 请求参数

```
role: "2"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» role	body	string	true	1-管理员，2-服务员，3-厨师

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[User]	true	none	所有服务员的数组，如果该数组 size 为 0 则没查到任何
»» userId	integer	true	none	员工编号与登录账号
»» password	string	true	none	none
»» username	string	true	none	无
»» role	integer	true	none	1-管理员，2-服务员，3-厨师
»» portrait	string	true	none	图片链接
»» phone	string	true	none	none
»» salary	number	true	none	none
»» bonus	number	true	none	none

GET 根据员工姓名或 ID 进行模糊查询

GET /user/select_by_name_id

如果提交的查询调价为空，返回所有数据

Body 请求参数

[query](#): Mar

--

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» query	body	string	true	姓名或 ID

返回示例

成功

```
{
  "code": 12,
  "data": [
    {
      "userId": 500000198501048500,
      "username": "龙超",
      "role": 1,
      "portrait": "http://dummyimage.com/728x90",
      "password": "adipisicing",
      "phone": "18152271833",
      "salary": 74,
      "bonus": 64
    }
  ]
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[object]	true	none	none

名称	类型	必选	约束	说明
»» userId	integer	false	none	none
»» username	string	false	none	none
»» role	integer	false	none	none
»» portrait	string	false	none	none
»» password	string	false	none	none
»» phone	string	false	none	none
»» salary	integer	false	none	none
»» bonus	integer	false	none	none

POST 删除人员

POST /user/delete

Body 请求参数

id: "21"

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	true	被删除人的 ID

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	是否操作成功
» msg	string	true	none	错误具体信息

POST 修改人员

POST /user/edit

如果某字段为空，默认不修改该字段

Body 请求参数

```
userId: "37"
username: Mary
bonus: "1245.5"
salary: "66.7"
phone: "1232313213"
portrait: http:yuyu.jpg
password: "23232323"
role: "3"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» userId	body	string	true	none
» username	body	string	false	none
» bonus	body	string	false	none
» salary	body	string	false	none
» phone	body	string	false	none
» portrait	body	string	false	none

名称	位置	类型	必选	说明
» password	body	string	false	none
» role	body	string	false	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否修改成功

GET 根据 ID 显示人员信息

GET /user/select_by_id

主要用于点击后跳转到该人员的详细界面

Body 请求参数

id: "37"

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	false	需要查询人的 ID

返回示例
成功

```
{
  "code": 31,
  "data": {
    "userId": 150000200505316770,
    "username": "孔芳",
    "role": 1,
    "portrait": "http://dummyimage.com/234x60",
    "password": "fugiat consectetur",
    "phone": "19827721914",
    "salary": 18,
    "bonus": 55
  }
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	object	true	none	none
»» userId	integer	true	none	none
»» username	string	true	none	none
»» role	integer	true	none	none
»» portrait	string	true	none	none
»» password	string	true	none	none
»» phone	string	true	none	none
»» salary	integer	true	none	none

名称	类型	必选	约束	说明
»» bonus	integer	true	none	none

GET 查询服务员的销售榜

GET /user/get_top3_waiter

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[UserSales]	true	none	none
»» userId	integer	true	none	none
»» username	string	true	none	none
»» role	integer	true	none	1-管理员, 2-服务员, 3-厨师
»» portrait	string	true	none	none
»» password	string	true	none	none
»» phone	string	true	none	none
»» salary	number	true	none	none
»» bonus	number	true	none	none
»» sales	integer	true	none	提交的订单数量

GET 显示所有人员

GET /user/list

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» data	[User]	true	none	所有服务员的数组，如果该数组 size 为 0 则没查到任何
»» userId	integer	true	none	员工编号与登录账号
»» password	string	true	none	none
»» username	string	true	none	无
»» role	integer	true	none	1-管理员，2-服务员，3-厨师
»» portrait	string	true	none	图片链接
»» phone	string	true	none	none
»» salary	number	true	none	none
»» bonus	number	true	none	none
» code	integer	true	none	是否查询成功

POST 增加人员

POST /user/add

注意：ID 数据库自动添加，前端不用发送

Body 请求参数

username: wew
bonus: "1245.5"
salary: "66.7"
phone: "1232313213"
portrait: yuyu.jpg
password: "23232323"
role: "2"
email: 123@qq.com

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» username	body	string	true	none
» bonus	body	string	true	none
» salary	body	string	true	none
» phone	body	string	true	none
» portrait	body	string	true	none
» password	body	string	true	none
» role	body	string	true	none
» email	body	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否操作成功
» msg	string	true	none	信息
» data	User	true	none	none
»» userId	integer	true	none	员工编号与登录账号
»» password	string	true	none	none
»» username	string	true	none	无
»» role	integer	true	none	1-管理员, 2-服务员, 3-厨师
»» portrait	string	true	none	图片链接
»» phone	string	true	none	none
»» salary	number	true	none	none
»» bonus	number	true	none	none

点餐/管理员/菜品管理

POST 设置菜品可出售

POST /food/set_available

Body 请求参数

```
id: "2"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none

名称	位置	类型	必选	说明
» id	body	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否设置成功

POST 推荐菜品

POST /food/set_like

Body 请求参数

```
id: "4"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	true	推荐菜品 id

返回示例

返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

POST 删除菜品

POST /food/delete

Body 请求参数

```
id: "50"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	false	删除菜品 ID

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否删除成功

POST 添加菜品

POST /food/add

注意：ID 数据库自动添加，前端不用发送

Body 请求参数

```
foodName: 卷心菜
price: "54"
description: labore ex ea nulla
isLike: "1"
category: "2"
image: http://dummyimage.com/234x60
isAvailable: "0"
sales: "5"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» foodName	body	string	true	none
» price	body	string	true	none
» description	body	string	true	none
» isLike	body	string	true	none
» category	body	string	true	none
» image	body	string	true	none
» isAvailable	body	string	true	none
» sales	body	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否添加成功
» data	Food	true	none	none
»» foodName	string	true	none	菜品名称
»» price	number	true	none	菜品价格
»» description	string	true	none	菜品简介
»» foodId	integer	true	none	菜品 id
»» isLike	integer	true	none	0/1 是否推荐
»» category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
»» image	string	true	none	菜品图片 url 链接
»» isAvailable	integer	true	none	0/1 是否可售

POST 设置菜品售空

POST /food/set_not_available

Body 请求参数

```
id: "2"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none

名称	位置	类型	必选	说明
» id	body	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否修改成功

POST 修改菜品

POST /food/edit

只修改不为空的值

Body 请求参数

```

foodId: "53"
foodName: 牛肉饼
price: "54"
description: 原汁原味
isLike: "1"
category: "2"
image: http://dummyimage.com/234x60
isAvailable: "1"
sales: "37"

```

请求参数

名称	位置	类型	必选	说明
----	----	----	----	----

名称	位置	类型	必选	说明
body	body	object	false	none
» foodId	body	string	true	none
» foodName	body	string	false	none
» price	body	string	false	none
» description	body	string	false	none
» isLike	body	string	false	none
» category	body	string	false	none
» image	body	string	false	none
» isAvailable	body	string	false	none
» sales	body	string	false	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否操作成功

点餐/管理员/订单管理

GET 根据订单 id 展示某订单的详细信息

GET /order/all_detail_info

返回结果包括订单点的菜品、点单服务员、订单基本信息

Body 请求参数

id: "1"

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	object	true	none	none
»» waiter	User	true	none	none
»»» userId	integer	true	none	员工编号与登录账号
»»» password	string	true	none	none
»»» username	string	true	none	无
»»» role	integer	true	none	1-管理员, 2-服务员, 3-厨师
»»» portrait	string	true	none	图片链接
»»» phone	string	true	none	none
»»» salary	number	true	none	none
»»» bonus	number	true	none	none

名称	类型	必选	约束	说明
»» foodList	[Food]	true	none	点菜信息
»»» foodName	string	true	none	菜品名称
»»» price	number	true	none	菜品价格
»»» description	string	true	none	菜品简介
»»» foodId	integer	true	none	菜品 id
»»» isLike	integer	true	none	0/1 是否推荐
»»» category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
»»» image	string	true	none	菜品图片 url 链接
»»» isAvailable	integer	true	none	0/1 是否可售
»» order	TOrder	true	none	none
»»» orderId	integer	true	none	订单编号，每个订单唯一
»»» tableId	integer	true	none	桌号
»»» userId	integer	true	none	服务员 ID
»»» memo	string	true	none	订单备注
»»» totalPrice	number	true	none	订单总价
»»» status	integer	true	none	订单状态（0-未支付，1-已支付）
»»» createTime	string	true	none	点餐时间
»»» finishTime	string	true	none	结餐时间
»»» numPeople	integer	true	none	用餐人数

GET 根据 ID 显示订单详情

GET /order/get_by_id

Body 请求参数

```
id: "3"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	true	none

返回示例

成功

```
{
  "code": 1,
  "msg": "success",
  "data": {
    "orderId": 4,
    "userId": 1,
    "tableId": 1,
    "totalPrice": 100,
    "memo": "加辣",
    "status": 3,
    "createTime": "2022-03-02T17:44:28",
    "finishTime": "2022-03-04T10:47:02",
    "numPeople": 4,
    "username": "刘一璇 1"
  }
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none

名称	类型	必选	约束	说明
»» orderId	integer	true	none	none
»» userId	integer	true	none	none
»» tableId	integer	true	none	none
»» totalPrice	number	true	none	none
»» memo	string	true	none	none
»» status	integer	true	none	none
»» createTime	string	true	none	none
»» finishTime	string	true	none	none
»» numPeople	integer	true	none	none
»» username	string	true	none	none

GET 根据桌号或编号搜索订单

GET /order/select_by_table_or_id

如果提交的查询调价为空，返回所有数据

Body 请求参数

```
id: "3"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	false	桌号或订单号

返回示例

返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[TOrder]	true	none	none
»» orderId	integer	true	none	订单编号，每个订单唯一
»» tableId	integer	true	none	桌号
»» userId	integer	true	none	服务员 ID
»» memo	string	true	none	订单备注
»» totalPrice	number	true	none	订单总价
»» status	integer	true	none	订单状态（0-未支付，1-已支付）
»» createTime	string	true	none	点餐时间
»» finishTime	string	true	none	结餐时间
»» numPeople	integer	true	none	用餐人数

GET 根据订单创建日期查询订单

GET /order/select_by_date

Body 请求参数

date: 2022-02-25

请求参数

名称	位置	类型	必选	说明
----	----	----	----	----

名称	位置	类型	必选	说明
body	body	object	false	none
» date	body	string	true	日期

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[TOrder]	true	none	none
»» orderId	integer	true	none	订单编号，每个订单唯一
»» tableId	integer	true	none	桌号
»» userId	integer	true	none	服务员 ID
»» memo	string	true	none	订单备注
»» totalPrice	number	true	none	订单总价
»» status	integer	true	none	订单状态（0-未支付，1-已支付）
»» createTime	string	true	none	点餐时间
»» finishTime	string	true	none	结餐时间
»» numPeople	integer	true	none	用餐人数

POST 删除订单

POST /order/delete

Body 请求参数

```
id: "2"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否删除成功
» msg	string	true	none	具体信息

POST 订单直接免单

POST /order/free

Body 请求参数

```
id: "2"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	是否面单成功
» msg	string	true	none	返回消息内容

点餐/管理员/订单管理/显示订单

GET 正在进行

GET /order/todoList

显示正在进行的订单

返回示例

成功

```
{
  "code": 1,
  "data": [
    {
      "orderId": 46,
```

```

    "tableId": 38,
    "userId": 18,
    "totalPrice": 51,
    "status": 65,
    "memo": "incididunt sint deserunt",
    "createTime": "2010-01-27 19:24:51",
    "finishTime": "2001-02-05 23:05:53",
    "numPeople": 56
  }
]
}

```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[object]	true	none	none
»» orderId	integer	false	none	none
»» tableId	integer	false	none	none
»» userId	integer	false	none	none
»» totalPrice	integer	false	none	none
»» status	integer	false	none	none
»» memo	string	false	none	none
»» createTime	string	false	none	none

名称	类型	必选	约束	说明
»» finishTime	string	false	none	none
»» numPeople	integer	false	none	none

GET 已完成

GET /order/finishList

显示已完成的订单信息

返回示例

成功

```
{
  "code": 1,
  "data": [
    {
      "orderId": 46,
      "tableId": 38,
      "userId": 18,
      "totalPrice": 51,
      "status": 65,
      "memo": "incididunt sint deserunt",
      "createTime": "2010-01-27 19:24:51",
      "finishTime": "2001-02-05 23:05:53",
      "numPeople": 56
    }
  ]
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[object]	true	none	none
»» orderId	integer	false	none	none
»» tableId	integer	false	none	none
»» userId	integer	false	none	none
»» totalPrice	integer	false	none	none
»» status	integer	false	none	none
»» memo	string	false	none	none
»» createTime	string	false	none	none
»» finishTime	string	false	none	none
»» numPeople	integer	false	none	none

GET 显示所有订单信息

GET /order/list

返回示例

成功

```
{
  "code": 1,
  "data": [
    {
      "orderId": 46,
      "tableId": 38,
      "userId": 18,
      "totalPrice": 51,
      "status": 65,
      "memo": "incididunt sint deserunt",
      "createTime": "2010-01-27 19:24:51",
      "finishTime": "2001-02-05 23:05:53",
      "numPeople": 56
    }
  ]
}
```

```
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[object]	true	none	none
»» orderId	integer	false	none	none
»» tableId	integer	false	none	none
»» userId	integer	false	none	none
»» totalPrice	number	false	none	none
»» status	integer	false	none	none
»» memo	string	false	none	none
»» createTime	string	false	none	none
»» finishTime	string	false	none	none
»» numPeople	integer	false	none	none
»» username	string	true	none	none

点餐/管理员/菜品查询

GET 显示所有菜品信息

GET /food/list

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[Food]	true	none	none
»» foodName	string	true	none	菜品名称
»» price	number	true	none	菜品价格
»» description	string	true	none	菜品简介
»» foodId	integer	true	none	菜品 id
»» isLike	integer	true	none	0/1 是否推荐
»» category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
»» image	string	true	none	菜品图片 url 链接
»» isAvailable	integer	true	none	0/1 是否可售

GET 根据菜品编号或名称查找菜品

GET /food/select_by_name_id

模糊查询。如果提交结果为空，则返回所有数据

Body 请求参数

query: 菜

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» query	body	string	true	菜品编号或名称

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	[Food]	true	none	none
»» foodName	string	true	none	菜品名称
»» price	number	true	none	菜品价格
»» description	string	true	none	菜品简介
»» foodId	integer	true	none	菜品 id
»» isLike	integer	true	none	0/1 是否推荐
»» category	integer	true	none	菜品类别: 1-主菜, 2-小吃, 3-饮料
»» image	string	true	none	菜品图片 url 链接
»» isAvailable	integer	true	none	0/1 是否可售

GET 根据价格区间查找菜品

GET /food/select_by_price

Body 请求参数

min: "40"
max: "60"

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» min	body	string	true	最低价格（min 与 max 不可同时为 null）
» max	body	string	true	最高价格（min 与 max 不可同时为 null）

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	该价格区间是否存在菜品
» data	[Food]	true	none	符合条件的食物
»» foodName	string	true	none	菜品名称
»» price	number	true	none	菜品价格
»» description	string	true	none	菜品简介
»» foodId	integer	true	none	菜品 id
»» isLike	integer	true	none	0/1 是否推荐

名称	类型	必选	约束	说明
»» category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
»» image	string	true	none	菜品图片 url 链接
»» isAvailable	integer	true	none	0/1 是否可售

GET 根据类别查询菜品

GET /food/select_by_category

Body 请求参数

```
category: "2"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» category	body	string	true	1-主菜，2-小吃，3-饮料

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	查询是否成功

名称	类型	必选	约束	说明
» data	[Food]	true	none	符合条件的菜品数组
»» foodName	string	true	none	菜品名称
»» price	number	true	none	菜品价格
»» description	string	true	none	菜品简介
»» foodId	integer	true	none	菜品 id
»» isLike	integer	true	none	0/1 是否推荐
»» category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
»» image	string	true	none	菜品图片 url 链接
»» isAvailable	integer	true	none	0/1 是否可售

GET 根据 id 显示菜品

GET /food/select_by_id

Body 请求参数

```
id: "3"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» data	Food	true	none	none
»» foodName	string	true	none	菜品名称
»» price	number	true	none	菜品价格
»» description	string	true	none	菜品简介
»» foodId	integer	true	none	菜品 id
»» isLike	integer	true	none	0/1 是否推荐
»» category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
»» image	string	true	none	菜品图片 url 链接
»» isAvailable	integer	true	none	0/1 是否可售

GET 查询菜品的销售榜

GET /food/get_top3_food

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none

名称	类型	必选	约束	说明
» data	[FoosSales]	true	none	none
»» foodName	string	true	none	菜品名称
»» price	number	true	none	菜品价格
»» description	string	true	none	菜品描述
»» foodId	integer	true	none	菜品 ID
»» isLike	integer	true	none	0/1 是否推荐
»» category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
»» image	string	true	none	菜品图片连接
»» isAvailable	integer	true	none	0/1 是否可售
»» sales	integer	true	none	菜品销量

点餐/服务员/公告

POST 公告设为已读

POST /waiter/notice/status

Body 请求参数

```
notice_id: "2"
user_id: "8"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» notice_id	body	string	true	none

名称	位置	类型	必选	说明
» user_id	body	string	true	none

返回示例
成功

```
{
  "code": 1,
  "msg": "成功"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none

GET 实时公告

GET /waiter/notice

服务员实时公告页面

Body 请求参数

```
user_id: "3"
```

请求参数

名称	位置	类型	必选	说明
----	----	----	----	----

名称	位置	类型	必选	说明
body	body	object	false	none
» user_id	body	string	true	none

返回示例

成功

```
{
  "code": 1,
  "msg": "成功",
  "data": {
    "userId": 123,
    "userName": "蒋丽",
    "role": 2,
    "portrait": "http://dummyimage.com/120x90",
    "password": 123456,
    "notice": [
      {
        "notice_id": 1,
        "title": "test1",
        "status": 0,
        "content": "test1 content"
      },
      {
        "notice_id": 2,
        "title": "test2",
        "status": 1,
        "content": "test2 content"
      }
    ]
  }
}
```

```
{
  "code": -1,
  "msg": "失败",
  "data": {
    "user": {
      "user_id": 83,
      "user_name": "马明",
      "role": 84,
      "portrait": "voluptate",
      "password": 20
    }
  }
}
```



```
    },
    "notice": [
      {
        "title": "自日往角难们",
        "status": 84
      },
      {
        "title": "政委程受入对劳",
        "status": 2
      },
      {
        "title": "群器从速先好",
        "status": 75
      }
    ]
  }
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» userId	integer	true	none	none
»» userName	string	true	none	none
»» role	integer	true	none	none
»» portrait	string	true	none	none
»» password	string	true	none	none

名称	类型	必选	约束	说明
»» notice	[object]	true	none	none
»»» noticeId	integer	true	none	none
»»» title	string	true	none	none
»»» status	integer	true	none	none
»»» content	string	true	none	none
»»» createTime	string	true	none	none

POST 公告详情页

POST /waiter/notice_details

显示公告详情

Body 请求参数

```
notice_id: "1"
user_id: "8"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» notice_id	body	string	true	none
» user_id	body	string	true	none

返回示例

成功

```
{
  "code": 1,
  "msg": "成功",
  "data": {
    "noticeId": 25,
    "title": "test",
    "content": "hello world",
```

```

    "createTime": "2022-2-12-14-31",
    "status": 1
  }
}
{
  "code": -1,
  "msg": "失败",
  "data": {
    "notice_id": 70,
    "title": "要温除目",
    "content": "in aliqua id sed elit",
    "create_time": "1991-03-07 03:29:31",
    "status": 33
  }
}

```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» noticeId	integer	true	none	none
»» title	string	true	none	none
»» content	string	true	none	none
»» createTime	string	true	none	none
»» status	integer	true	none	none

点餐/服务员/后厨消息

POST 后厨消息设为已读

POST /waiter/message/status

Body 请求参数

```
message_id: "1"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» message_id	body	string	true	none

返回示例

成功

```
{
  "code": 1,
  "msg": "成功"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none

GET 后厨消息列表

GET /waiter/message

服务员实时公告页面

Body 请求参数

```
receive_user: "2"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» receive_user	body	string	true	none

返回示例
成功

```
{
  "code": 1,
  "msg": "成功",
  "data": {
    "userId": 123,
    "userName": "蒋丽",
    "role": 2,
    "portrait": "http://dummyimage.com/120x90",
    "password": "123456",
    "messages": [
      {
        "messageId": 1,
        "orderId": 1,
        "title": "test2",
        "content": "hello",
        "createTime": "2022-12-20 12:12:12",
        "sendUser": 2,
        "receiveUser": 3,
        "status": 0,
        "senderName": "shan"
      }
    ]
  }
}
```

```

    ]
  }
}

```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» userId	integer	true	none	服务员的 user_id
»» userName	string	true	none	none
»» role	integer	true	none	none
»» portrait	string	true	none	none
»» password	string	true	none	none
»» messages	[object]	true	none	none
»»» messageId	integer	false	none	none
»»» orderId	integer	false	none	none
»»» title	string	false	none	none
»»» content	string	false	none	none
»»» createTime	string	false	none	none
»»» sendUser	integer	false	none	none
»»» receiveUser	integer	false	none	none
»»» status	integer	false	none	none

名称	类型	必选	约束	说明
»»» senderName	string	false	none	发这条消息的后厨的 username

GET 后厨消息详情

GET /waiter/message_details

显示后厨发送的消息

Body 请求参数

```
message_id: "1"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» message_id	body	string	true	none

返回示例

成功

```
{
  "code": 1,
  "msg": "成功",
  "data": {
    "messageId": 1,
    "orderId": 1,
    "title": "test2",
    "content": "hello",
    "createTime": "2022-12-20 12:12:12",
    "sendUser": 2,
    "receiveUser": 3,
    "status": 0,
    "senderName": "zhan"
  }
}
```

```
{
  "code": -1,
  "msg": "失败",
  "data": {
    "order_id": 83,
    "title": "酸造料感该光",
    "content": "proident sed Lorem",
    "create_time": "2006-06-09 14:18:16",
    "status": 78
  }
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» messageId	integer	true	none	none
»» orderId	integer	true	none	none
»» title	string	true	none	none
»» content	string	true	none	none
»» createTime	string	true	none	none
»» sendUser	integer	true	none	none
»» receiveUser	integer	true	none	none

名称	类型	必选	约束	说明
»» status	integer	true	none	none
»» senderName	string	true	none	none

点餐/服务员/点菜

POST 确认点单

POST /waiter/take_order

点击确认点单

Body 请求参数

```
{
  "user_id": 94,
  "table_id": 40,
  "num_people": 28,
  "memo": "eu",
  "total_price": 5,
  "orderViews": [
    {
      "foodId": 5,
      "numFood": 58
    }
  ]
}
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» userId	body	integer	true	none
» tableId	body	integer	true	none
» numPeople	body	integer	true	none

名称	位置	类型	必选	说明
» memo	body	string	true	none
» totalPrice	body	number	true	none
» orderViews	body	[object]	true	none
»» foodId	body	integer	true	none
»» numFood	body	integer	true	none

返回示例
成功

```
{
  "code": 1,
  "msg": "成功"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none

GET 菜品列表

GET /waiter/food

服务员菜品列表

请求参数

名称	位置	类型	必选	说明
query	query	string	false	none

返回示例

成功

```
{
  "code": 46,
  "msg": "dolor",
  "data": [
    {
      "description": "quis tempor in Ut nostrud",
      "category": 20,
      "isLike": true,
      "foodId": 38,
      "price": 23,
      "sales": 17,
      "image": "http://dummyimage.com/400x400",
      "isAvailable": true,
      "name": "活节里何和真"
    },
    {
      "isAvailable": true,
      "isLike": false,
      "image": "http://dummyimage.com/400x400",
      "description": "in dolor tempor enim",
      "name": "工到将离",
      "category": 78,
      "foodId": 34,
      "sales": 95,
      "price": 37
    }
  ]
}

{
  "code": -1,
  "msg": "失败",
  "data": {
    "food": {
      "name": "少部王名好图",
      "price": 7,
      "description": "voluptate",
      "food_id": 66,
```

```

        "is_like": true,
        "category": 23,
        "image": "http://dummyimage.com/336x280",
        "is_available": true,
        "sales": 1
    }
}
}

```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	[object]	true	none	none
»» name	string	false	none	none
»» price	number	false	none	none
»» description	string	false	none	none
»» foodId	integer	false	none	none
»» isLike	boolean	false	none	none
»» category	integer	false	none	none
»» image	string	false	none	none
»» isAvailable	boolean	false	none	none
»» sales	integer	false	none	none

GET 菜品详情页面

GET /waiter/food_details

服务员菜品详情

请求参数

名称	位置	类型	必选	说明
food_id	query	string	true	none

返回示例

成功

```
{
  "code": 1,
  "msg": "成功",
  "data": {
    "foodName": "重庆小面",
    "price": 40,
    "description": "aliqua",
    "foodId": 94,
    "isLike": 1,
    "category": 1,
    "image": "http://dummyimage.com/180x150",
    "isAvailable": 1,
    "sales": 85
  }
}

{
  "code": -1,
  "msg": "失败",
  "data": {
    "food": {
      "name": "务术形增",
      "price": 23,
      "description": "dolor consectetur reprehenderit id",
      "food_id": 18,
      "is_like": false,
      "category": 63,
      "image": "http://dummyimage.com/400x400",
```

```
        "is_available": true,
        "sales": 85
    }
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» foodName	string	true	none	none
»» price	number	true	none	none
»» description	string	true	none	none
»» foodId	integer	true	none	none
»» isLike	integer	true	none	none
»» category	integer	true	none	none
»» image	string	true	none	none
»» isAvailable	integer	true	none	none
»» sales	integer	true	none	none

点餐/服务员/订单结算

GET 待结算订单列表

GET /waiter/to_pay

显示待结算订单

Body 请求参数

```
user_id: "8"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» user_id	body	string	true	none

返回示例

成功

```
{
  "code": 1,
  "msg": "成功",
  "data": [
    {
      "orderId": 3,
      "userId": 2,
      "tableId": 39,
      "totalPrice": 1,
      "memo": "sit",
      "createTime": "2220-12-12 12:12:12",
      "finishTime": "2220-12-12 12:12:12",
      "numPeople": 97
    }
  ]
}

{
  "code": -1,
  "msg": "失败",
  "data": {
    "order": {
```

```

    "order_id": 51,
    "table_id": 30,
    "user_id": 17,
    "total_price": 82,
    "status": 4,
    "memo": "fugiat adipisicing nulla dolor elit",
    "create_time": "1982-11-21 12:41:09",
    "finish_time": "2010-04-02 14:45:22",
    "num_people": 13
  }
}
}

```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	[object]	true	none	none
»» orderId	integer	false	none	none
»» userId	integer	false	none	none
»» tableId	integer	false	none	none
»» totalPrice	integer	false	none	none
»» memo	string	false	none	none
»» createTime	string	false	none	none
»» finishTime	string	false	none	none
»» numPeople	integer	false	none	none

POST 确认结算

POST /waiter/to_pay/pay

点击结算按钮 改变订单状态为已结算

Body 请求参数

```
order_id: "1"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» order_id	body	string	true	none

返回示例
成功

```
{
  "code": 1,
  "msg": "成功"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none

POST 订单结算详情页面

POST /waiter/order_details

订单结算详情

Body 请求参数

```
order_id: "1"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» order_id	body	string	true	none

返回示例

成功

```
{
  "code": 1,
  "msg": "成功",
  "data": {
    "torder": {
      "orderId": 49,
      "userID": 49,
      "tableId": 90,
      "totalPrice": 96,
      "memo": "dolor eu sunt",
      "status": 1,
      "createTime": "2013-10-27 10:38:00",
      "finishTime": "2020-10-09 08:31:51",
      "numPeople": 14
    },
    "foodList": [
      {
        "description": "culpa sint magna velit veniam",
        "isLike": 25,
```

```

        "category": 67,
        "price": 24,
        "foodName": "离存之该程心",
        "image": "123123.123.123",
        "numFood": 12
    }
]
}
}
{
    "code": -1,
    "msg": "失败",
    "data": {
        "order_id": 36,
        "table_id": 98,
        "create_time": "1999-07-08 02:13:47",
        "finish_time": "2012-10-16 18:42:21",
        "num_people": 26,
        "user": {
            "user_id": 66,
            "user_name": "赵秀兰"
        },
        "food": [
            {
                "description": "sunt quis irure",
                "food_name": "也回利等比装单",
                "food_id": 63,
                "category": 43
            }
        ]
    }
}
}

```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» torder	object	true	none	none
»»» orderId	integer	true	none	none
»»» userId	integer	true	none	none
»»» tableId	integer	true	none	none
»»» totalPrice	integer	true	none	none
»»» memo	string	true	none	none
»»» status	integer	true	none	none
»»» createTime	string	true	none	none
»»» finishTime	string	true	none	none
»»» numPeople	integer	true	none	none
»» foodList	[object]	true	none	none
»»» description	string	false	none	none
»»» isLike	integer	false	none	none
»»» category	integer	false	none	none
»»» price	integer	false	none	none
»»» foodName	string	false	none	none
»»» image	string	false	none	none
»»» numFood	integer	false	none	none

点餐/服务员/传菜消息

GET 传菜推送消息列表

GET /waiter/take_food

显示传菜消息

Body 请求参数

```
{}
```

请求参数

名称	位置	类型	必选	说明
user_id	query	string	true	none
body	body	object	false	none

返回示例
成功

```
{
  "code": 1,
  "msg": "成功",
  "data": [
    {
      "tableId": 1,
      "orderId": 2,
      "foodId": 3,
      "foodName": "豆腐",
      "image": "www.123.com",
      "status": 2
    }
  ]
}

{
  "code": -1,
  "msg": "失败",
  "data": {
    "user": {
      "user_id": 92,
      "user_name": "冯洋",
      "role": 13,
      "portrait": "consequat ut sunt amet Excepteur",
      "password": 79
    }
  }
}
```

```
    },
    "ordered_food": {
      "name": "比各三县社",
      "food_id": 47,
      "category": 18,
      "image": "http://dummyimage.com/400x400",
      "table_id": 37
    }
  }
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	[object]	true	none	none
»» tableId	integer	false	none	none
»» orderId	integer	false	none	none
»» foodId	integer	false	none	none
»» foodName	string	false	none	none
»» image	string	false	none	none
»» status	integer	false	none	none

GET 确认传菜

GET /waiter/take_food/update

将传菜消息转为已传菜

Body 请求参数

order_id: "2"
food_id: "2"

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» order_id	body	string	true	none
» food_id	body	string	true	none

返回示例

成功

{
"code": 1,
"msg": "成功"
}

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

点餐/后厨/订单

POST 订单菜品烹饪状态

POST /back-kitchen/detail/status

改变未出餐订单菜品烹饪状态。

Body 请求参数

```
order_id: "1"
food_id: "1"
status: 0,1,2
table_id: "1"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» order_id	body	string	true	订单编号
» food_id	body	string	true	订单中某个菜品编号
» status	body	string	true	菜品烹饪状态（0-准备烹饪，1-正在烹饪，3-烹制完成）
» table_id	body	string	true	桌号

返回示例
成功

```
{
  "code": 1,
  "msg": "成功",
  "data": null
}

{
  "code": -1,
  "msg": "失败",
  "data": null
}
```

返回结果

状态码	状态码含义	说明	数据类型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	
» code	integer	true	none	no
» msg	string	true	none	no
» data	null	true	none	no

GET 订单列表

GET /back-kitchen/order_list

显示订单列表

返回示例

成功

```
{
  "code": 1,
  "msg": "成功",
  "data": {
    "user": {
      "ID": 123,
      "name": "name",
      "imgUrl": "url"
    },
    "orders": {
      "todo": [
        "order_id1",
        "order_id2"
      ],
      "complete": [
        "order_id1",
        "order_id2"
      ]
    }
  }
}

{
  "code": -1,
```

```
"msg": "失败"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» user	object	true	none	none
»»» ID	integer	true	none	none
»»» name	string	true	none	none
»»» imgUrl	string	true	none	none
»» orders	object	true	none	none
»»» todo	[string]	true	none	none
»»» complete	[string]	true	none	none

GET 获取所有订单号

GET /back-kitchen/orderIds

获取所有订单号。

返回示例

成功

```
{
```

```
"code": 1,
"msg": "成功",
"data": {
  "orderId": [
    1,
    2,
    3,
    4
  ]
}
{
  "code": -1,
  "msg": "失败",
  "data": null
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» orderId	[integer]	true	none	none

GET 订单详情

GET /back-kitchen/detail

订单详情页面

Body 请求参数

```
order_id: "1"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» order_id	body	string	true	订单编号

返回示例
成功

```
{
  "code": 1,
  "msg": "成功",
  "data": {
    "order_id": "abc",
    "table_id": 1,
    "waiter_name": "name",
    "waiter_id": 1,
    "order_note": "note",
    "food": [
      {
        "foodId": 1,
        "name": "name",
        "number": 1,
        "status": 1
      }
    ]
  }
}

{
  "code": -1,
  "msg": "失败"
}
```

返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» orderId	integer	true	none	none
»» tableId	integer	true	none	none
»» username	string	true	none	none
»» userId	integer	true	none	none
»» memo	string	true	none	none
»» food	[object]	true	none	none
»»» foodId	integer	false	none	none
»»» name	string	false	none	none
»»» number	integer	false	none	none
»»» status	integer	false	none	none

点餐/后厨/消息

POST 消息编辑

POST /back-kitchen/notify

消息编辑页面

Body 请求参数

```
title: string
order_id: string
content: string
create_time: string
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» title	body	string	true	消息标题
» order_id	body	string	true	订单编号
» content	body	string	true	订单内容
» create_time	body	string	true	创建时间

返回示例
成功

```
{
  "code": 1,
  "msg": "成功"
}

{
  "code": -1,
  "msg": "失败"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none

名称	类型	必选	约束	说明
» msg	string	true	none	none

POST 后厨通知服务员

POST /back-kitchen/message

后厨通知服务员消息。

Body 请求参数

title: string

content: string

order_id: string

create_time: string

send_user: "13"

receive_user: "8"

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» title	body	string	true	通知标题
» content	body	string	true	通知内容
» order_id	body	string	true	订单号
» create_time	body	string	true	通知创建时间
» send_user	body	string	true	发送通知后厨 id
» receive_user	body	string	true	接收通知服务员 id

返回示例

成功

```
{
  "code": 1,
  "msg": "成功",
  "data": null
}
```

```
}

{
  "code": -1,
  "mgs": "失败",
  "data": null
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

点餐/后厨/公告

GET 实时公告

GET /back-kitchen/notice

实时公告页面

Body 请求参数

```
user_id: string
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» user_id	body	string	true	none

返回示例
成功

```
{
```



```
"code": 1,
"msg": "成功",
"data": {
  "notices": [
    {
      "content": "123",
      "createTime": "2022-2-2 12:00:00",
      "createTimeStr": "123",
      "noticeId": 1,
      "status": 0,
      "title": "123"
    }
  ]
}
{
  "code": -1,
  "msg": "失败"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» notices	[object]	true	none	none

名称	类型	必选	约束	说明
»»» content	string	false	none	none
»»» createTime	string	false	none	none
»»» createTimeStr	string	false	none	none
»»» noticeId	integer	false	none	none
»»» status	integer	false	none	none
»»» title	string	false	none	none

POST 后厨公告消息标记为已读

POST /back-kitchen/notice/read

后厨公告消息标记为已读。

Body 请求参数

```
user_id: "13"
notice_id: "1"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» user_id	body	string	true	后厨用户 id
» notice_id	body	string	true	公告 id

返回示例
成功

```
{
  "code": 1,
```

```
"msg": "成功",
"data": null
}

{
  "code": -1,
  "msg": "失败",
  "data": null
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	null	true	none	none

点餐/后厨/出餐

POST 烹饪状态

POST /back-kitchen/food_status

设置烹饪状态的按钮

Body 请求参数

```
status: "1"
order_id: string
food_id: string
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» status	body	string	true	当前订单特定菜品烹饪状态
» order_id	body	string	true	none
» food_id	body	string	true	none

返回示例
成功

```
{
  "code": 1,
  "msg": "成功"
}

{
  "code": -1,
  "msg": "失败"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none

GET 未出餐

GET /back-kitchen/unfinished

显示未出餐订单列表

返回示例

成功

```
{
  "code": 9,
  "msg": "reprehenderit eu Lorem",
  "data": [
    {
      "orderId": 41,
      "tableId": 64,
      "userId": 3,
      "totalPrice": 32,
      "status": 25,
      "memo": "in",
      "createTime": "1978-12-07 12:04:11",
      "finishTime": "2001-07-24 06:18:22",
      "numPeople": 73
    }
  ]
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	[TOrder]	true	none	none
»» orderId	integer	true	none	订单编号，每个订单唯一
»» tableId	integer	true	none	桌号

名称	类型	必选	约束	说明
»» userId	integer	true	none	服务员 ID
»» memo	string	true	none	订单备注
»» totalPrice	number	true	none	订单总价
»» status	integer	true	none	订单状态（0-未支付，1-已支付）
»» createTime	string	true	none	点餐时间
»» finishTime	string	true	none	结餐时间
»» numPeople	integer	true	none	用餐人数

GET 已出餐

GET /back-kitchen/finished

显示已出餐订单列表

返回示例

成功

```
{
  "code": -1,
  "msg": "失败"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none

名称	类型	必选	约束	说明
» msg	string	true	none	none
» data	[TOrder]	true	none	none
»» orderId	integer	true	none	订单编号，每个订单唯一
»» tableId	integer	true	none	桌号
»» userId	integer	true	none	服务员 ID
»» memo	string	true	none	订单备注
»» totalPrice	number	true	none	订单总价
»» status	integer	true	none	订单状态（0-未支付，1-已支付）
»» createTime	string	true	none	点餐时间
»» finishTime	string	true	none	结餐时间
»» numPeople	integer	true	none	用餐人数

登录

POST 用户登录界面请求

POST /user/index

所有用户登录接口，其中包括餐厅管理员、餐厅服务员以及餐厅后厨。(1) 判断 code，为-1 表示登录失败，此时 msg == "1"表示用户不存在，msg == "2"表示密码错误 (2) 判断 code，为 1 表示登录成功，此时 data.role == 1, 2, 3 代表：1-管理员，2-服务员，3-厨师

Body 请求参数

```
email: 1446775342@qq.com
password: "123"
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» email	body	string	true	用户邮箱
» password	body	string	true	用户密码

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	-1 失败（原因见 msg），1 成功
» data	User	true	none	none
»» userId	integer	true	none	员工编号与登录账号
»» password	string	true	none	none
»» username	string	true	none	无
»» role	integer	true	none	1-管理员，2-服务员，3-厨师
»» portrait	string	true	none	图片链接
»» phone	string	true	none	none
»» salary	number	true	none	none
»» bonus	number	true	none	none
» msg	string	true	none	"1"-用户名不存在，"2"-密码不存在，"success"-成功

GET 显示已登录员工的信息

GET /user/user_info

Body 请求参数

id: "1"

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» id	body	string	true	管理员自己的 ID

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» data	User	true	none	none
»» userId	integer	true	none	员工编号与登录账号
»» password	string	true	none	none
»» username	string	true	none	无
»» role	integer	true	none	1-管理员，2-服务员，3-厨师
»» portrait	string	true	none	图片链接
»» phone	string	true	none	none

名称	类型	必选	约束	说明
»» salary	number	true	none	none
»» bonus	number	true	none	none
» code	integer	true	none	none

数据可视化

GET 销售榜前 10 菜品

GET /showTopFood

销售榜前 10 菜品信息

返回示例

成功

```
{
  "code": 1,
  "msg": "success",
  "data": [
    {
      "foodName": "鱼香肉丝",
      "foodId": 5,
      "sales": 1388,
      "percent": 0.8218
    },
    {
      "foodName": "鱼香肉丝",
      "foodId": 22,
      "sales": 120,
      "percent": 0.071
    },
    {
      "foodName": "麻婆豆腐",
      "foodId": 2,
      "sales": 49,
      "percent": 0.029
    },
    {

```

"foodName": "豆腐花",
"foodId": 4,
"sales": 25,
"percent": 0.0148
},
{
"foodName": "松鼠桂鱼",
"foodId": 8,
"sales": 21,
"percent": 0.0124
},
{
"foodName": "羊肉串",
"foodId": 9,
"sales": 14,
"percent": 0.0083
},
{
"foodName": "测试数据",
"foodId": 20,
"sales": 14,
"percent": 0.0083
},
{
"foodName": "BB 面",
"foodId": 31,
"sales": 13,
"percent": 0.0077
},
{
"foodName": "可口可乐",
"foodId": 33,
"sales": 13,
"percent": 0.0077
},
{
"foodName": "东坡肉",
"foodId": 11,
"sales": 9,
"percent": 0.0053
},
{
"foodName": "撒尿牛丸",
"foodId": 7,

```
    "sales": 8,
    "percent": 0.0047
  },
  {
    "foodName": "重庆鸡公煲",
    "foodId": 3,
    "sales": 6,
    "percent": 0.0036
  },
  {
    "foodName": "小丸子",
    "foodId": 32,
    "sales": 5,
    "percent": 0.003
  },
  {
    "foodName": "宫保鸡丁",
    "foodId": 1,
    "sales": 3,
    "percent": 0.0018
  },
  {
    "foodName": "冒菜",
    "foodId": 6,
    "sales": 1,
    "percent": 0.0006
  }
]
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none

名称	类型	必选	约束	说明
» msg	string	true	none	none
» data	[object]	true	none	none
»» foodName	string	false	none	none
»» foodId	integer	false	none	none
»» sales	integer	false	none	none
»» percent	number	false	none	none

POST 近 M 天销售额变化

POST /sale_change

展示第 start 天到第 end 天销售额随时间变化

Body 请求参数

```
start: 2022-03-08
end: 2022-03-11
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» start	body	string	true	none
» end	body	string	true	none

返回示例
成功

```
{
  "code": 1,
  "msg": "ok",
  "data": [
    {
```

```
    "time": "03-06",
    "sales": 8
  },
  {
    "time": "03-07",
    "sales": 8
  },
  {
    "time": "03-08",
    "sales": 8
  },
  {
    "time": "03-09",
    "sales": 8
  },
  {
    "time": "03-10",
    "sales": 8
  },
  {
    "time": "03-11",
    "sales": 8
  },
  {
    "time": "03-12",
    "sales": 8
  }
]
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none

名称	类型	必选	约束	说明
» msg	string	true	none	none
» data	[object]	true	none	none
»» time	string	false	none	none
»» salePrice	integer	false	none	none

GET 销售榜前 10 服务生

GET /showTopWaiter

返回示例

成功

```
{
  "code": 1,
  "msg": "success",
  "data": [
    {
      "userName": "曾平",
      "userId": 13,
      "sales": 32,
      "percent": 0.6531
    },
    {
      "userName": "曾平",
      "userId": 8,
      "sales": 12,
      "percent": 0.2449
    },
    {
      "userName": "朱棣",
      "userId": 49,
      "sales": 4,
      "percent": 0.0816
    },
    {
      "userName": "李瑾美",
      "userId": 10,
      "sales": 1,
```

"percent": 0.0204
}
]
}

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» data	[object]	true	none	none
»» username	string	false	none	none
»» userId	integer	false	none	none
»» sales	integer	false	none	销售额
»» percent	number	false	none	所占比例

POST 近 M 天销量变化

POST /sale_num_change

展示第 start 天到第 end 天销售额随时间变化

Body 请求参数

start: 2022-03-09
end: 2022-03-10

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» start	body	string	true	none
» end	body	string	true	none

返回示例
成功

```
{
  "code": 1,
  "msg": "success",
  "data": [
    {
      "time": "2022-03-09",
      "sales": null,
      "nums": "23"
    },
    {
      "time": "2022-03-10",
      "sales": null,
      "nums": "7"
    }
  ]
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none

名称	类型	必选	约束	说明
» data	[object]	true	none	none
»» time	string	true	none	none
»» sales	integer	true	none	none

GET 今日总销售额

GET /today_sales

返回示例

成功

```
{
  "code": 51,
  "msg": "sint quis ipsum est",
  "total": 72
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 **200**

名称	类型	必选	约束	说明
» code	integer	true	none	none
» msg	string	true	none	none
» total	integer	true	none	none

数据模型

Food


```
{
  "foodName": "string",
  "price": 0,
  "description": "string",
  "foodId": 0,
  "isLike": 0,
  "category": 0,
  "image": "string",
  "isAvailable": 0
}
```

属性

名称	类型	必选	约束	说明
foodName	string	true	none	菜品名称
price	number	true	none	菜品价格
description	string	true	none	菜品简介
foodId	integer	true	none	菜品 id
isLike	integer	true	none	0/1 是否推荐
category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
image	string	true	none	菜品图片 url 链接
isAvailable	integer	true	none	0/1 是否可售

FoosSales

名称	类型	必选	约束	说明
foodName	string	true	none	菜品名称
price	number	true	none	菜品价格
description	string	true	none	菜品描述
foodId	integer	true	none	菜品 ID

```
{
  "foodName": "string",
  "price": 0,
  "description": "string",
  "foodId": 0,
  "isLike": 0,
  "category": 0,
  "image": "string",
  "isAvailable": 0,
  "sales": 0
}
```

属性

名称	类型	必选	约束	说明
isLike	integer	true	none	0/1 是否推荐
category	integer	true	none	菜品类别：1-主菜，2-小吃，3-饮料
image	string	true	none	菜品图片连接
isAvailable	integer	true	none	0/1 是否可售
sales	integer	true	none	菜品销量

TOrder


```
{
  "orderId": 0,
  "tableId": 0,
  "userId": 0,
  "memo": "string",
  "totalPrice": 0,
  "status": 0,
  "createTime": "string",
  "finishTime": "string",
  "numPeople": 0
}
```

属性

名称	类型	必选	约束	说明
orderId	integer	true	none	订单编号，每个订单唯一
tableId	integer	true	none	桌号
userId	integer	true	none	服务员 ID
memo	string	true	none	订单备注

名称	类型	必选	约束	说明
totalPrice	number	true	none	订单总价
status	integer	true	none	订单状态（0-未支付，1-已支付）
createTime	string	true	none	点餐时间
finishTime	string	true	none	结餐时间
numPeople	integer	true	none	用餐人数

UserSales


```
{
  "userId": 0,
  "username": "string",
  "role": 0,
  "portrait": "string",
  "password": "string",
  "phone": "string",
  "salary": 0,
  "bonus": 0,
  "sales": 0
}
```

属性

名称	类型	必选	约束	说明
userId	integer	true	none	none
username	string	true	none	none
role	integer	true	none	1-管理员，2-服务员，3-厨师
portrait	string	true	none	none
password	string	true	none	none

名称	类型	必选	约束	说明
phone	string	true	none	none
salary	number	true	none	none
bonus	number	true	none	none
sales	integer	true	none	提交的订单数量

User


```
{
  "userId": 0,
  "password": "string",
  "username": "string",
  "role": 0,
  "portrait": "string",
  "phone": "string",
  "salary": 0,
  "bonus": 0
}
```

属性

名称	类型	必选	约束	说明
userId	integer	true	none	员工编号与登录账号
password	string	true	none	none
username	string	true	none	无

名称	类型	必选	约束	说明
role	integer	true	none	1-管理员，2-服务员，3-厨师
portrait	string	true	none	图片链接
phone	string	true	none	none
salary	number	true	none	none
bonus	number	true	none	none