**平时作业**

**宣奔1230696180**

选题2

基于Python实现点菜系统

要求：

对系统进行功能需求分析

设计合理的[数据结构](https://zhida.zhihu.com/search?content_id=247776201&content_type=Article&match_order=1&q=%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84&zhida_source=entity" \t "https://zhuanlan.zhihu.com/p/_blank)和[系统框架](https://zhida.zhihu.com/search?content_id=247776201&content_type=Article&match_order=1&q=%E7%B3%BB%E7%BB%9F%E6%A1%86%E6%9E%B6&zhida_source=entity" \t "https://zhuanlan.zhihu.com/p/_blank)

界面设计美观、清楚、合理

编程简练，程序功能齐全，能正确运行

具有一定的创新性

说明书、流程图要清楚

课题完成后必须按要求提交课程设计报告

任务：

建立一个[字典](https://zhida.zhihu.com/search?content_id=247776201&content_type=Article&match_order=1&q=%E5%AD%97%E5%85%B8&zhida_source=entity" \t "https://zhuanlan.zhihu.com/p/_blank)，存储菜名，菜价。用户输入菜名，显示菜价。最后显示顾客的点菜单，菜的单价，及总价，输入折扣率，显示实付的钱数。

3. 课程设计说明书

1.主要数据和主要[函数](https://zhida.zhihu.com/search?content_id=247776201&content_type=Article&match_order=1&q=%E5%87%BD%E6%95%B0&zhida_source=entity" \t "https://zhuanlan.zhihu.com/p/_blank)介绍：

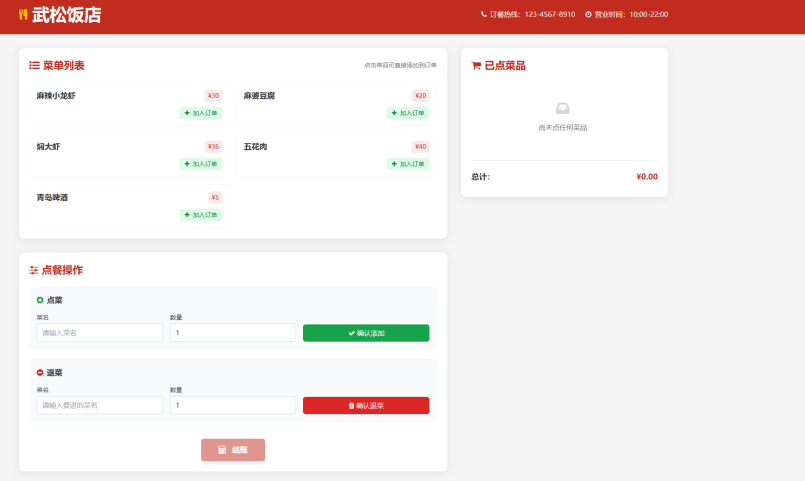
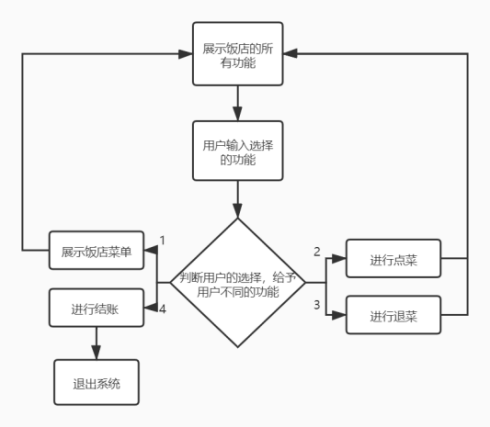
字典menu存储了饭店中的所有菜的Mingcheng，及饭菜所对应的单价

字典ordef\_list存储了用户点菜的Mingcheng，及其数量

函数order\_dishes 用户进行点菜，需要用户进行输入菜的Mingcheng和数量

函数back\_dishes用户进行退菜，需要用户进行输入菜的Mingcheng和数量

函数sum\_dishes用户进行结账，显示用户的最终所点的饭菜的单价、数量、总结和总的消费，并根据用户输入的折扣率来计算用户实际支付的金额。



点菜系统课程设计综合报告

摘要

本课程设计聚焦于饭店点菜系统的开发，通过功能需求分析、数据结构与系统框架设计，运用Python语言实现了包含查看菜单、点菜、退菜及结账等核心功能的完整系统。系统采用字典作为核心数据结构，通过函数拆分与流程优化确保代码结构清晰、逻辑严谨。本报告详细阐述了系统开发过程，涵盖功能需求、数据结构、代码实现、函数说明及运行逻辑等关键环节，并针对课程设计报告的编写提出系统性建议，为同类课程设计提供参考范例。

关键词

点菜系统；Python；数据结构；函数拆分；课程设计

一、功能需求分析

1.1 功能展示模块

系统需具备清晰直观的功能菜单展示能力，能够引导用户通过数字选择实现查看菜单、点菜、退菜及结账等核心操作。功能菜单应采用循环展示机制，确保用户完成单次操作后可返回主菜单继续选择，直至主动选择结账功能退出系统。

1.2 菜单查看模块

系统应支持用户实时查看饭店当前提供的所有菜品及其对应价格信息。菜单展示需采用结构化格式，确保菜品名称与价格信息清晰可辨，为后续点菜操作提供数据支撑。

1.3 点菜功能模块

点菜功能需实现以下核心要求：

菜品存在性校验：确保用户输入菜品名称存在于系统菜单中

数量合法性校验：要求输入数量为正整数，并支持重复点菜时的数量累加

数据动态更新：实时更新用户点菜记录，为退菜和结账功能提供准确数据

1.4 退菜功能模块

退菜功能需满足以下业务逻辑：

菜品存在性校验：确认用户欲退菜品存在于当前订单中

数量合理性校验：确保退菜数量不超过已点数量且为正整数

数据动态调整：根据退菜数量实时更新订单记录，当某菜品数量归零时自动移除该菜品记录

1.5 结账功能模块

结账功能应实现以下完整流程：

订单明细展示：按结构化格式显示菜品名称、单价、数量及小计金额

总价计算：汇总所有菜品小计金额生成订单总价

折扣处理：支持用户输入折扣率（0-1区间），计算实付金额并处理异常输入

系统退出：完成结账后终止程序运行

二、数据结构与系统框架设计

2.1 核心数据结构设计

系统采用双字典结构实现数据管理：

菜单字典（menu）：以菜品名称为键（str类型），对应价格（int类型）为值，存储饭店基础菜品数据

订单字典（order\_list）：以菜品名称为键（str类型），对应数量（int类型）为值，动态记录用户点菜情况

2.2 系统框架设计

系统采用模块化设计思想，通过主控函数（main）实现功能调度，各功能模块通过函数封装实现独立操作。程序运行流程遵循"展示菜单→用户选择→功能执行→结果反馈→循环等待"的闭环逻辑，确保系统操作的连贯性和完整性。

三、Python代码实现

3.1 基础数据结构初始化

python

# 菜品信息字典

menu = {

"麻辣小龙虾": 30,

"麻婆豆腐": 20,

"焖大虾": 36,

"五花肉": 40,

"青岛啤酒": 5

}

# 订单信息字典

order\_list = {}

3.2 核心功能函数实现

3.2.1 菜单展示函数

python

def show\_menu():

print("\n本饭店的菜单如下：")

for idx, (dish, price) in enumerate(menu.items(), 1):

print(f"菜名：{dish}\t价格：{price}")

print("-" \* 30)

3.2.2 点菜处理函数

python

def order\_dishes():

dish\_name = input("请输入你要点菜的名称：")

if dish\_name in menu:

try:

quantity = int(input("请输入你要点的数量："))

if quantity <= 0:

raise ValueError

order\_list[dish\_name] = order\_list.get(dish\_name, 0) + quantity

print(f"小二，我要 {dish\_name}{quantity} 份！")

except ValueError:

print("输入错误：数量必须为正整数！")

else:

print("错误提示：菜单中没有这道菜！")

3.2.3 退菜处理函数

python

def back\_dishes():

dish\_name = input("请输入你要退菜的名称：")

if dish\_name in order\_list:

try:

quantity = int(input("请输入你要退的数量："))

if quantity <= 0 or quantity > order\_list[dish\_name]:

raise ValueError

order\_list[dish\_name] -= quantity

if order\_list[dish\_name] == 0:

del order\_list[dish\_name]

print(f"已为您退掉 {dish\_name}{quantity} 份")

except ValueError:

print("输入错误：退菜数量必须为正整数且不超过已点数量！")

else:

print("错误提示：您未点这道菜！")

3.2.4 结账处理函数

python

def sum\_dishes():

if not order\_list:

print("错误提示：您还没有点菜！")

return

print("\n您的点菜单：")

print("菜名\t\t单价\t数量\t小计")

total = sum(menu[dish] \* quantity for dish, quantity in order\_list.items())

for dish, quantity in order\_list.items():

print(f"{dish}\t\t{menu[dish]}\t{quantity}\t{menu[dish]\*quantity}")

print(f"总价：{total} 元")

try:

discount = float(input("请输入折扣率（如0.8表示8折）："))

if 0 < discount <= 1:

actual\_pay = total \* discount

print(f"实付金额：{actual\_pay:.2f} 元")

else:

raise ValueError

except ValueError:

print("折扣率无效，按原价结算")

print(f"实付金额：{total:.2f} 元")

print("感谢您的惠顾，欢迎下次光临！")

exit()

3.3 主控函数实现

python

def main():

print("-----欢迎来到武松饭店-----")

while True:

print("\n本饭店的功能如下：")

print("1. 查看菜单")

print("2. 进行点菜")

print("3. 进行退菜")

print("4. 进行结账")

choice = input("客官的选择是：")

operations = {

'1': show\_menu,

'2': order\_dishes,

'3': back\_dishes,

'4': sum\_dishes

}

if choice in operations:

operations[choice]()

else:

print("无效选择，请重新输入！")

if \_\_name\_\_ == "\_\_main\_\_":

main()

四、代码实现说明

4.1 函数拆分策略

系统采用"单一职责原则"进行函数设计：

show\_menu()：专注菜单展示功能

order\_dishes()：处理点菜业务逻辑

back\_dishes()：管理退菜操作流程

sum\_dishes()：完成结账全流程

main()：负责系统流程控制与功能调度

4.2 流程优化措施

采用字典存储功能映射关系，消除多重if-else判断，使用字典的get()方法实现数量累加的简洁表达，通过异常处理机制统一管理输入错误，运用生成器表达式优化总价计算性能。

4.3 输入校验机制

系统在关键操作节点设置多重校验：

点菜环节：校验菜品存在性、数量正整数性

退菜环节：校验菜品存在性、数量合理性（>0且≤已点数量）

结账环节：校验折扣率有效性（0<rate≤1）

五、课程设计说明书补充

5.1 主要数据结构说明

数据结构 类型 用途 示例数据

menu dict 存储菜品信息 {"麻辣小龙虾":30}

order\_list dict 记录订单信息 {"麻婆豆腐":2}

5.2 核心函数说明

函数名 输入参数 返回值 功能描述

show\_menu 无 无 展示菜单信息

order\_dishes 无 无 处理点菜请求

back\_dishes 无 无 处理退菜请求

sum\_dishes 无 无 完成结账流程

main 无 无 系统主控流程

5.3 系统运行逻辑

初始化菜单数据和订单数据，进入主循环展示功能菜单，根据用户选择调用对应功能函数，功能执行后返回主菜单（结账功能除外），结账成功后终止程序运行。

六总结与展望

本课程设计成功实现了饭店点菜系统的核心功能，通过模块化设计和严谨的输入校验机制确保了系统的健壮性。系统采用Python字典数据结构实现高效的数据管理，通过函数拆分使代码结构清晰易维护。未来工作可考虑以下扩展方向：

增加数据库支持实现数据持久化

开发图形用户界面提升用户体验

添加用户管理系统支持多用户操作

实现网络功能支持远程点餐

本设计为后续开发奠定了坚实基础，相关设计思路和实现技术可为同类信息系统开发提供参考。

附录：完整源代码

menu = {

"麻辣小龙虾": 30,

"麻婆豆腐": 20,

"焖大虾": 36,

"五花肉": 40,

"青岛啤酒": 5

}

order\_list = {}

def show\_menu():

print("\n本饭店的菜单如下：")

for idx, (dish, price) in enumerate(menu.items(), 1):

print(f"菜名：{dish}\t价格：{price}")

print("-" \* 30)

def order\_dishes():

dish\_name = input("请输入你要点菜的名称：")

if dish\_name in menu:

try:

quantity = int(input("请输入你要点的数量："))

if quantity <= 0:

print("数量要为正整数，请重新操作！")

return

if dish\_name in order\_list:

order\_list[dish\_name] += quantity

else:

order\_list[dish\_name] = quantity

print(f"小二，我要{dish\_name}{quantity}份！")

except ValueError:

print("数量无效，请输入整数！")

else:

print("菜单中没有这道菜，请重新选择！")

def back\_dishes():

dish\_name = input("请输入要退菜的名称：")

if dish\_name in order\_list:

try:

quantity = int(input("请输入要退的数量："))

if quantity <= 0:

print("退菜数量要为正整数，请重新操作！")

return

if quantity > order\_list[dish\_name]:

print("退菜数量不能超过已点数量，请重新操作！")

return

order\_list[dish\_name] -= quantity

if order\_list[dish\_name] == 0:

del order\_list[dish\_name]

print(f"已为您退掉 {dish\_name}{quantity} 份")

except ValueError:

print("输入的数量无效，请输入整数！")

else:

print("没点这道菜无法退菜，请重新选择！")

def sum\_dishes():

if not order\_list:

print("还没有点菜，无法结账！")

return

print("\n您的点菜单：")

print("菜名\t\t单价\t数量\t小计")

total = 0

for dish, quantity in order\_list.items():

price = menu[dish]

subtotal = price \* quantity

total += subtotal

print(f"{dish}\t\t{price}\t{quantity}\t{subtotal}")

print(f"总价：{total} 元")

try:

discount\_rate = float(input("请输入折扣率（如 0.8 表示 8 折，1.0 表示无折扣）："))

if 0 < discount\_rate <= 1:

actual\_pay = total \* discount\_rate

print(f"实付金额：{actual\_pay:.2f} 元")

else:

print("折扣率无效，按无折扣计算！")

print(f"实付金额：{total:.2f} 元")

except ValueError:

print("输入无效，按无折扣计算！")

print(f"实付金额：{total:.2f} 元")

print("感谢您的惠顾，欢迎下次光临！")

exit()

def main():

print("-----欢迎来到武松饭店-----")

while True:

print("\n本饭店的功能如下：")

print("1. 查看菜单")

print("2. 进行点菜")

print("3. 进行退菜")

print("4. 进行结账")

choice = input("客官的选择是：")

if choice == "1":

show\_menu()

elif choice == "2":

order\_dishes()

elif choice == "3":

back\_dishes()

elif choice == "4":

sum\_dishes()

else:

print("无效的选择，请重新输入！")

if \_\_name\_\_ == "\_\_main\_\_":

main()

完整界面代码

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>武松饭店 - 点餐系统</title>

<script src="https://cdn.tailwindcss.com"></script>

<link href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-awesome.min.css" rel="stylesheet">

<!-- 配置Tailwind自定义颜色和字体 -->

<script>

tailwind.config = {

theme: {

extend: {

colors: {

primary: '#C12C1F', // 中国红 - 主色调

secondary: '#FFB800', // 金黄色 - 辅助色

neutral: '#F5F5F5', // 浅灰色 - 背景色

dark: '#333333', // 深灰色 - 文字色

},

fontFamily: {

sans: ['Inter', 'system-ui', 'sans-serif'],

},

}

}

}

</script>

<style type="text/tailwindcss">

@layer utilities {

.content-auto {

content-visibility: auto;

}

.card-shadow {

box-shadow: 0 4px 20px rgba(0, 0, 0, 0.08);

}

.dish-card-hover {

transition: all 0.3s ease;

}

.dish-card-hover:hover {

transform: translateY(-5px);

box-shadow: 0 8px 30px rgba(0, 0, 0, 0.12);

}

}

</style>

</head>

<body class="bg-neutral min-h-screen text-dark">

<!-- 顶部导航栏 -->

<header class="bg-primary text-white shadow-md">

<div class="container mx-auto px-4 py-4 flex justify-between items-center">

<div class="flex items-center space-x-3">

<i class="fa fa-cutlery text-2xl text-secondary"></i>

<h1 class="text-[clamp(1.5rem,3vw,2.5rem)] font-bold">武松饭店</h1>

</div>

<div class="hidden md:flex items-center space-x-6">

<div class="flex items-center">

<i class="fa fa-phone mr-2"></i>

<span>订餐热线：123-4567-8910</span>

</div>

<div class="flex items-center">

<i class="fa fa-clock-o mr-2"></i>

<span>营业时间：10:00-22:00</span>

</div>

</div>

</div>

</header>

<!-- 主内容区 -->

<main class="container mx-auto px-4 py-8">

<div class="grid grid-cols-1 lg:grid-cols-3 gap-8">

<!-- 左侧菜单区域 -->

<section class="lg:col-span-2">

<div class="bg-white rounded-xl p-6 card-shadow mb-8">

<div class="flex justify-between items-center mb-6">

<h2 class="text-2xl font-bold text-primary flex items-center">

<i class="fa fa-list-ul mr-2"></i> 菜单列表

</h2>

<div class="text-sm text-gray-500">

点击菜品可直接添加到订单

</div>

</div>

<div class="grid grid-cols-1 md:grid-cols-2 gap-4" id="menu-container">

<!-- 菜单项将通过JS动态生成 -->

</div>

</div>

<!-- 操作区域 -->

<div class="bg-white rounded-xl p-6 card-shadow">

<h2 class="text-2xl font-bold text-primary flex items-center mb-6">

<i class="fa fa-sliders mr-2"></i> 点餐操作

</h2>

<div class="space-y-6">

<!-- 点菜表单 -->

<div class="p-4 border border-gray-100 rounded-lg bg-gray-50">

<h3 class="text-lg font-semibold mb-4 flex items-center">

<i class="fa fa-plus-circle text-green-600 mr-2"></i> 点菜

</h3>

<div class="grid grid-cols-1 md:grid-cols-3 gap-4">

<div>

<label for="dish-name" class="block text-sm font-medium text-gray-700 mb-1">菜名</label>

<input type="text" id="dish-name" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-primary/50" placeholder="请输入菜名">

</div>

<div>

<label for="dish-quantity" class="block text-sm font-medium text-gray-700 mb-1">数量</label>

<input type="number" id="dish-quantity" min="1" value="1" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-primary/50">

</div>

<div class="flex items-end">

<button id="add-dish" class="w-full bg-green-600 hover:bg-green-700 text-white font-medium py-2 px-4 rounded-md transition duration-200 flex items-center justify-center">

<i class="fa fa-check mr-1"></i> 确认添加

</button>

</div>

</div>

</div>

<!-- 退菜表单 -->

<div class="p-4 border border-gray-100 rounded-lg bg-gray-50">

<h3 class="text-lg font-semibold mb-4 flex items-center">

<i class="fa fa-minus-circle text-red-600 mr-2"></i> 退菜

</h3>

<div class="grid grid-cols-1 md:grid-cols-3 gap-4">

<div>

<label for="remove-dish-name" class="block text-sm font-medium text-gray-700 mb-1">菜名</label>

<input type="text" id="remove-dish-name" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-primary/50" placeholder="请输入要退的菜名">

</div>

<div>

<label for="remove-dish-quantity" class="block text-sm font-medium text-gray-700 mb-1">数量</label>

<input type="number" id="remove-dish-quantity" min="1" value="1" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-primary/50">

</div>

<div class="flex items-end">

<button id="remove-dish" class="w-full bg-red-600 hover:bg-red-700 text-white font-medium py-2 px-4 rounded-md transition duration-200 flex items-center justify-center">

<i class="fa fa-trash mr-1"></i> 确认退菜

</button>

</div>

</div>

</div>

<!-- 结账按钮 -->

<div class="text-center pt-4">

<button id="checkout-btn" class="bg-primary hover:bg-primary/90 text-white font-bold py-3 px-10 rounded-md transition duration-200 text-lg shadow-lg hover:shadow-xl transform hover:-translate-y-0.5">

<i class="fa fa-calculator mr-2"></i> 结账

</button>

</div>

</div>

</div>

</section>

<!-- 右侧订单区域 -->

<section>

<div class="bg-white rounded-xl p-6 card-shadow sticky top-4">

<h2 class="text-2xl font-bold text-primary flex items-center mb-6">

<i class="fa fa-shopping-cart mr-2"></i> 已点菜品

</h2>

<div id="order-list" class="space-y-3 max-h-[400px] overflow-y-auto pr-2">

<!-- 订单内容将通过JS动态生成 -->

<div class="text-center text-gray-500 py-10">

<i class="fa fa-inbox text-4xl mb-3 opacity-30"></i>

<p>尚未点任何菜品</p>

</div>

</div>

<div class="border-t border-gray-200 mt-6 pt-6">

<div class="flex justify-between text-lg font-semibold mb-2">

<span>总计：</span>

<span id="total-price" class="text-primary">¥0.00</span>

</div>

<div id="discount-section" class="hidden">

<div class="mt-4 mb-3">

<label for="discount-rate" class="block text-sm font-medium text-gray-700 mb-1">折扣率</label>

<input type="number" id="discount-rate" min="0.01" max="1" step="0.01" value="1.00" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-primary/50">

</div>

<div class="flex justify-between text-lg font-semibold text-green-600">

<span>折扣后应付：</span>

<span id="discounted-price">¥0.00</span>

</div>

</div>

</div>

</div>

</section>

</div>

</main>

<!-- 结账弹窗 -->

<div id="checkout-modal" class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 hidden">

<div class="bg-white rounded-xl p-8 max-w-md w-full mx-4 transform transition-all scale-95 opacity-0" id="modal-content">

<div class="text-center mb-6">

<div class="inline-flex items-center justify-center w-16 h-16 rounded-full bg-primary/10 text-primary mb-4">

<i class="fa fa-receipt text-3xl"></i>

</div>

<h3 class="text-2xl font-bold text-primary">账单详情</h3>

</div>

<div id="bill-details" class="space-y-2 mb-6 max-h-[300px] overflow-y-auto pr-2">

<!-- 账单详情将通过JS动态生成 -->

</div>

<div class="border-t border-gray-200 pt-4 mb-6">

<div class="flex justify-between font-semibold mb-1">

<span>原价总计：</span>

<span id="bill-total">¥0.00</span>

</div>

<div class="flex justify-between font-semibold text-green-600">

<span>实付金额：</span>

<span id="bill-payment">¥0.00</span>

</div>

</div>

<div class="text-center">

<button id="confirm-payment" class="bg-primary hover:bg-primary/90 text-white font-bold py-3 px-10 rounded-md transition duration-200 w-full">

确认支付

</button>

<p class="text-gray-500 text-sm mt-4">感谢光临，欢迎下次再来！</p>

</div>

</div>

</div>

<!-- 通知提示 -->

<div id="notification" class="fixed bottom-6 right-6 bg-dark text-white px-6 py-3 rounded-lg shadow-lg transform translate-y-20 opacity-0 transition-all duration-300 flex items-center z-50">

<i id="notification-icon" class="fa fa-info-circle mr-2"></i>

<span id="notification-message"></span>

</div>

<script>

// 菜单数据

const menu = {

"麻辣小龙虾": 30,

"麻婆豆腐": 20,

"焖大虾": 36,

"五花肉": 40,

"青岛啤酒": 5

};

// 已点菜品列表

let orderList = {};

// DOM元素

const menuContainer = document.getElementById('menu-container');

const orderListElement = document.getElementById('order-list');

const totalPriceElement = document.getElementById('total-price');

const discountedPriceElement = document.getElementById('discounted-price');

const discountRateElement = document.getElementById('discount-rate');

const discountSection = document.getElementById('discount-section');

const checkoutBtn = document.getElementById('checkout-btn');

const checkoutModal = document.getElementById('checkout-modal');

const modalContent = document.getElementById('modal-content');

const billDetails = document.getElementById('bill-details');

const billTotal = document.getElementById('bill-total');

const billPayment = document.getElementById('bill-payment');

const confirmPaymentBtn = document.getElementById('confirm-payment');

const addDishBtn = document.getElementById('add-dish');

const removeDishBtn = document.getElementById('remove-dish');

const notification = document.getElementById('notification');

const notificationIcon = document.getElementById('notification-icon');

const notificationMessage = document.getElementById('notification-message');

// 初始化菜单

function initMenu() {

menuContainer.innerHTML = '';

for (const [dish, price] of Object.entries(menu)) {

const dishCard = document.createElement('div');

dishCard.className = 'border border-gray-100 rounded-lg p-4 dish-card-hover cursor-pointer';

dishCard.innerHTML = `

<div class="flex justify-between items-start mb-3">

<h3 class="font-semibold text-lg">${dish}</h3>

<span class="bg-primary/10 text-primary px-2 py-1 rounded-md text-sm font-medium">¥${price}</span>

</div>

<div class="text-right">

<button class="add-to-order text-sm bg-green-100 hover:bg-green-200 text-green-700 px-3 py-1 rounded transition duration-200" data-dish="${dish}">

<i class="fa fa-plus mr-1"></i> 加入订单

</button>

</div>

`;

menuContainer.appendChild(dishCard);

}

// 为"加入订单"按钮添加事件监听

document.querySelectorAll('.add-to-order').forEach(button => {

button.addEventListener('click', (e) => {

e.stopPropagation();

const dish = e.currentTarget.getAttribute('data-dish');

addDish(dish, 1);

});

});

// 点击菜品卡片也可以添加菜品

document.querySelectorAll('.dish-card-hover').forEach(card => {

card.addEventListener('click', () => {

const dish = card.querySelector('.add-to-order').getAttribute('data-dish');

addDish(dish, 1);

});

});

}

// 更新订单列表显示

function updateOrderList() {

if (Object.keys(orderList).length === 0) {

orderListElement.innerHTML = `

<div class="text-center text-gray-500 py-10">

<i class="fa fa-inbox text-4xl mb-3 opacity-30"></i>

<p>尚未点任何菜品</p>

</div>

`;

discountSection.classList.add('hidden');

checkoutBtn.disabled = true;

checkoutBtn.classList.add('opacity-50', 'cursor-not-allowed');

} else {

orderListElement.innerHTML = '';

for (const [dish, quantity] of Object.entries(orderList)) {

const price = menu[dish];

const subtotal = price \* quantity;

const orderItem = document.createElement('div');

orderItem.className = 'flex justify-between items-center p-3 border-b border-gray-100 last:border-0';

orderItem.innerHTML = `

<div>

<h4 class="font-medium">${dish}</h4>

<p class="text-sm text-gray-500">单价：¥${price}</p>

</div>

<div class="flex items-center space-x-4">

<div class="flex items-center border rounded-md">

<button class="decrease-qty px-2 py-1 text-gray-600 hover:bg-gray-100" data-dish="${dish}">

<i class="fa fa-minus text-xs"></i>

</button>

<span class="px-2">${quantity}</span>

<button class="increase-qty px-2 py-1 text-gray-600 hover:bg-gray-100" data-dish="${dish}">

<i class="fa fa-plus text-xs"></i>

</button>

</div>

<span class="font-semibold">¥${subtotal.toFixed(2)}</span>

<button class="remove-item text-red-500 hover:text-red-700" data-dish="${dish}">

<i class="fa fa-trash-o"></i>

</button>

</div>

`;

orderListElement.appendChild(orderItem);

}

// 添加数量调整事件监听

document.querySelectorAll('.decrease-qty').forEach(button => {

button.addEventListener('click', (e) => {

const dish = e.currentTarget.getAttribute('data-dish');

removeDish(dish, 1);

});

});

document.querySelectorAll('.increase-qty').forEach(button => {

button.addEventListener('click', (e) => {

const dish = e.currentTarget.getAttribute('data-dish');

addDish(dish, 1);

});

});

document.querySelectorAll('.remove-item').forEach(button => {

button.addEventListener('click', (e) => {

const dish = e.currentTarget.getAttribute('data-dish');

removeDish(dish, orderList[dish]);

});

});

discountSection.classList.remove('hidden');

checkoutBtn.disabled = false;

checkoutBtn.classList.remove('opacity-50', 'cursor-not-allowed');

}

updateTotalPrice();

}

// 更新总价

function updateTotalPrice() {

let total = 0;

for (const [dish, quantity] of Object.entries(orderList)) {

total += menu[dish] \* quantity;

}

totalPriceElement.textContent = `¥${total.toFixed(2)}`;

// 计算折扣后价格

const discount = parseFloat(discountRateElement.value) || 1;

const discountedPrice = total \* discount;

discountedPriceElement.textContent = `¥${discountedPrice.toFixed(2)}`;

// 更新账单信息

billTotal.textContent = `¥${total.toFixed(2)}`;

billPayment.textContent = `¥${discountedPrice.toFixed(2)}`;

}

// 添加菜品

function addDish(dish, quantity) {

if (!menu.hasOwnProperty(dish)) {

showNotification('错误', `没有${dish}这道菜哦！`, 'error');

return;

}

if (quantity <= 0 || isNaN(quantity)) {

showNotification('错误', '数量必须是正整数！', 'error');

return;

}

if (orderList.hasOwnProperty(dish)) {

orderList[dish] += quantity;

} else {

orderList[dish] = quantity;

}

updateOrderList();

showNotification('成功', `已添加 ${dish} ${quantity} 份`, 'success');

}

// 移除菜品

function removeDish(dish, quantity) {

if (!orderList.hasOwnProperty(dish)) {

showNotification('错误', `没有点${dish}这道菜，无法退哦！`, 'error');

return;

}

if (quantity <= 0 || isNaN(quantity)) {

showNotification('错误', '数量必须是正整数', 'error');

return;

}

if (quantity > orderList[dish]) {

showNotification('错误', `最多只能退${orderList[dish]}份！`, 'error');

return;

}

orderList[dish] -= quantity;

if (orderList[dish] <= 0) {

delete orderList[dish];

}

updateOrderList();

showNotification('成功', `已退 ${dish} ${quantity} 份`, 'success');

}

// 显示结账弹窗

function showCheckoutModal() {

if (Object.keys(orderList).length === 0) {

showNotification('提示', '还没点菜呢，不用结账！', 'info');

return;

}

// 生成账单详情

billDetails.innerHTML = '';

let total = 0;

for (const [dish, quantity] of Object.entries(orderList)) {

const price = menu[dish];

const subtotal = price \* quantity;

total += subtotal;

const billItem = document.createElement('div');

billItem.className = 'flex justify-between py-1';

billItem.innerHTML = `

<span>${dish} × ${quantity}</span>

<span>¥${subtotal.toFixed(2)}</span>

`;

billDetails.appendChild(billItem);

}

const discount = parseFloat(discountRateElement.value) || 1;

const payment = total \* discount;

billTotal.textContent = `¥${total.toFixed(2)}`;

billPayment.textContent = `¥${payment.toFixed(2)}`;

// 显示弹窗并添加动画

checkoutModal.classList.remove('hidden');

setTimeout(() => {

modalContent.classList.remove('scale-95', 'opacity-0');

modalContent.classList.add('scale-100', 'opacity-100');

}, 10);

}

// 隐藏结账弹窗

function hideCheckoutModal() {

modalContent.classList.remove('scale-100', 'opacity-100');

modalContent.classList.add('scale-95', 'opacity-0');

setTimeout(() => {

checkoutModal.classList.add('hidden');

}, 300);

}

// 显示通知

function showNotification(title, message, type = 'info') {

notificationMessage.textContent = message;

// 设置图标和背景色

if (type === 'success') {

notificationIcon.className = 'fa fa-check-circle mr-2';

notification.classList.remove('bg-red-600', 'bg-blue-600');

notification.classList.add('bg-green-600');

} else if (type === 'error') {

notificationIcon.className = 'fa fa-exclamation-circle mr-2';

notification.classList.remove('bg-green-600', 'bg-blue-600');

notification.classList.add('bg-red-600');

} else {

notificationIcon.className = 'fa fa-info-circle mr-2';

notification.classList.remove('bg-green-600', 'bg-red-600');

notification.classList.add('bg-blue-600');

}

// 显示通知

notification.classList.remove('translate-y-20', 'opacity-0');

notification.classList.add('translate-y-0', 'opacity-100');

// 3秒后隐藏通知

setTimeout(() => {

notification.classList.remove('translate-y-0', 'opacity-100');

notification.classList.add('translate-y-20', 'opacity-0');

}, 3000);

}

// 事件监听

addDishBtn.addEventListener('click', () => {

const dish = document.getElementById('dish-name').value.trim();

const quantity = parseInt(document.getElementById('dish-quantity').value) || 0;

addDish(dish, quantity);

// 清空输入框

document.getElementById('dish-name').value = '';

document.getElementById('dish-quantity').value = '1';

});

removeDishBtn.addEventListener('click', () => {

const dish = document.getElementById('remove-dish-name').value.trim();

const quantity = parseInt(document.getElementById('remove-dish-quantity').value) || 0;

removeDish(dish, quantity);

// 清空输入框

document.getElementById('remove-dish-name').value = '';

document.getElementById('remove-dish-quantity').value = '1';

});

discountRateElement.addEventListener('input', updateTotalPrice);

checkoutBtn.addEventListener('click', showCheckoutModal);

confirmPaymentBtn.addEventListener('click', () => {

hideCheckoutModal();

orderList = {};

updateOrderList();

showNotification('成功', '支付成功！感谢您的光临！', 'success');

});

// 点击模态框外部关闭

checkoutModal.addEventListener('click', (e) => {

if (e.target === checkoutModal) {

hideCheckoutModal();

}

});

// 初始化

initMenu();

updateOrderList();

</script>

</body>

</html>