**重庆交通大学信息科学与工程学院**

**课程设计报告**

|  |  |
| --- | --- |
| 题 目 |  |
| 课程名称 | 计算思维综合实践I |
| 专业班级 | 计算机类24级 班 |
| 学 号 |  |
| 姓 名 |  |
| 指导教师 |  |

2025年1月

项目评分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 考核环节 | | 分值 | 评分标准与等级 | 成绩 |
| 项目  答辩  30% | 数据模型20% | 100 | A、数据模型设计合理，对数据模型语义及关键算法设计能清晰阐述；  B、数据模型设计合理，对数据模型语义及关键算法设计能比较清晰阐述；  C、数据模型设计基本合理，能比较清楚阐述数据模型语义及关键算法的设计思路；  D、数据模型设计存在小瑕疵，数据语义基本清楚，能对算法进行说明；  E、未完成数据模型设计或不能说明思路，关键算法未完成，或抄袭 |  |
| 系统功能40% | 100 | A、程序演示完全正确，能正确回答老师问题  B、按要求完成90%及以上功能，能正确回答老师问题  C、按要求完成80%及以上功能，基本能回答老师问题  D、按要求完成70%及以上功能，能回答老师多数问题  E、存在30%以上功能未完成或抄袭 |  |
| 工具使用 30% | 100 | A、对所用开发工具优缺点有清晰的认识，能熟练使用开发工具完成项目设计和编码  B、对所用开发工具优缺点有比较清晰的认识，能较熟练使用开发工具完成项目设计和编码  C、能熟练使用开发工具完成项目设计和编码  D、能较熟练使用开发工具完成项目设计和编码  E、对所选用开发工具掌握不熟悉，无法在开发环境中完成项目开发 |  |
| 团队合作10% | 100 | 1. 答辩时能清楚表达自己的意见，并能自觉维护团队和谐； 2. 答辩时能清楚表达自己的意见，能在潜意识中维护团队和谐；   C、答辩时能够表达自己的意见，在维护团队和谐方面表现一般；  C、答辩时能够在一定程度上表达自己的意见，或维护团队和谐意识淡漠；  E、答辩时不能表达自己的意见，或破坏团队和谐。 |  |
| 项目  报告  20% | 模型算法描述 40% | 100 | A、构建数据模型合理，算法描述清晰  B、构建数据模较合理，算法描述较清晰  C、构建数据模型基本合理，算法描述基本清晰  D、构建的数据模型在报告中描述不完整，但源代码中基本合理，算法描述基本清晰，但有缺项  E、未完成相关任务或抄袭 |  |
| 团队合作10% | 100 | 1. 项目分工合理，能高质量完成自己应该完成部分，能将自己完成的部分集成进整个系统 2. 项目分工合理，能较好地完成自己所完成的部分，能将自己完成的部分集成进整个系统 3. 项目分工基本合理，基本能完成自己应完成的任务，能将自己完成的部分集成进整个系统 4. 项目分工基本合理，基本能完成自己应完成的任务，但无法将自己完成的部分集成进整个系统 5. 未完成自己应完成部分。 |  |
| 报告质量 50% | 100 | A、报告规范，严格按照软件工程规范书写，分工明确，个人设计与实现阐述清楚  B、报告规范，能按照软件工程规范书写，分工明确，个人设计与实现阐述清楚，个别条目书写不完全符合要求，阐述基本清楚  C、报告规范，能按照软件工程规范书写，分工明确，个人设计与实现阐述基本清楚，存在20%以内条目书写不完全符合要求  D、报告规范，能按照软件工程规范书写，分工明确，个人设计与实现的阐述一般，存在30%以内条目书写不完全符合要求  E、报告不规范或存在30%以上条目书写不完全符合要求或抄袭 |  |
| 教师 | |  | | |

**餐厅点餐结算系统设计**

**设**

**计**

**说**

**明**

**书**

**2025年1月**

目 录

[第一章 系统分析 1](#_Toc10916_WPSOffice_Level1)

[1.1 系统概述 1](#_Toc19790_WPSOffice_Level2)

[1.2 系统使用角色及用户分析 1](#_Toc20440_WPSOffice_Level2)

[1.3 系统业务流程分析 2](#_Toc2058_WPSOffice_Level2)

[1.4 项目团队成员分工 2](#_Toc19193_WPSOffice_Level2)

[1.5 系统业务功能及数据分析 3](#_Toc32457_WPSOffice_Level2)

[1.5.2 收银员功能需求 4](#_Toc19193_WPSOffice_Level3)

[1.5.3 厨师功能需求 4](#_Toc32457_WPSOffice_Level3)

[1.5.4 服务员功能需求 5](#_Toc28657_WPSOffice_Level3)

[1.5.5 老板功能需求 5](#_Toc12133_WPSOffice_Level3)

[1.5.6 顾客需求 6](#_Toc12133_WPSOffice_Level3)

[第二章 系统设计 7](#_Toc19790_WPSOffice_Level1)

[2.1 系统功能设计 7](#_Toc28657_WPSOffice_Level2)

[2.1.1 系统设置子系统 7](#_Toc20017_WPSOffice_Level3)

[2.1.2 招生信息子系统 9](#_Toc22657_WPSOffice_Level3)

[2.1.3 学生信息管理子系统 9](#_Toc19177_WPSOffice_Level3)

[2.1.4 学生信息查询子系统 9](#_Toc12122_WPSOffice_Level3)

[2.2 数据结构设计 9](#_Toc12133_WPSOffice_Level2)

[2.2.1 整体数据模型 9](#_Toc16384_WPSOffice_Level3)

[2.2.2 数据结构设计 10](#_Toc16721_WPSOffice_Level3)

[第三章 系统实现及测试 12](#_Toc20440_WPSOffice_Level1)

[3.1 系统实现 12](#_Toc20017_WPSOffice_Level2)

[3.2 测试 12](#_Toc22657_WPSOffice_Level2)

[第四章 总结 14](#_Toc2058_WPSOffice_Level1)

第一章 系统分析

* 1. **系统概述**

餐厅点餐结算系统是一个集点餐、订单处理、厨房管理、结算支付和数据分析于一体的综合性管理系统。旨在提高餐厅的运营效率、服务质量和顾客满意度，同时为餐厅管理者提供决策支持。

通过初步了解，该餐厅点餐结算系统包含：

1、本系统用户包括消费者、收银员、厨师、服务员、餐厅老板、系统管理员等。

2、消费者现场点餐并分配餐位，并得到小票，可以预订餐位。

3、收银员负责对小票进行结算并收费，当消费金额不足 500 元时需要收取餐位费，结算时，只能对“已出菜”菜品进行结算，收银员还需要对当天的结算进行统计，当天结束后，还需要统计当天“待做”菜品和“已做”菜品。

4、厨师可以按序查看每个小票并做菜，当做完一个菜时需要更改该菜的状态，菜的初始状态均为“待做”，已出菜的菜品应改为“已做”，厨师也可以查看当天已完成的所有菜品。

5、服务员可以查看小票并将已进行服务，也可以根据客户的要求增加菜肴或加点酒水，增加的菜肴或加点的酒水由服务员录入该客户的小票中，已端到餐桌的菜其状态需要改为“已出菜”。

6、餐厅老板可以对所有信息进行查看，以及进行各种统计查询。

7、系统管理员对系统基本数据进行管理，包括餐位、菜肴、酒水等基础数据。

* 1. 系统使用角色及用户分析

一、系统使用角色划分

根据用户的基本业务要求分析，整个《餐厅点餐结算系统》的最终用户可以划分为如下几个组也就是不同的角色：

**1**、消费者

功能需求:现场点餐、预订餐位、获取小票。

操作流程:进入餐厅后，选择餐位进行现场点餐，系统分配餐位并生成小票。消费者可以提前预订餐位，以便在特定时间到店用餐。

**2**、收银员

功能需求:小票结算收费、统计当天结算、统计当天菜品状态。

操作流程:当消费者用餐完毕后，收银员对小票进行结算。如果消费金额不足 500 元，收取餐位费。结算时只能对 “已出菜” 菜品进行结算。当天结束后，统计当天 “待做” 菜品和 “已做” 菜品，以便进行库存管理和业绩核算。

**3**、厨师

功能需求:查看小票、服务客户、增加菜肴或酒水、更改菜品状态。

操作流程:服务员可以查看小票，了解客户的订单情况，为客户提供服务。根据客户的要求增加菜肴或加点酒水，并将其录入该客户的小票中。当菜品端到餐桌后，将其状态改为 “已出菜”。

**4**、服务员

功能需求：查看小票、服务客户、增加菜肴或酒水、更改菜品状态。 操作流程：服务员可以查看小票，了解客户的订单情况，为客户提供服

务。根据客户的要求增加菜肴或加点酒水，并将其录入该客户的小票中。当菜品端到餐桌后，将其状态改为 “已出菜”。

**5**、餐厅老板

功能需求：查看所有信息、进行统计查询。

操作流程：餐厅老板可以随时查看系统中的所有信息，包括订单、菜品状态、员工工作情况等。通过进行各种统计查询，了解餐厅的经营状况，以便制定决策和调整策略。

**6**、系统管理员

功能需求：管理系统基本数据。

操作流程：系统管理员负责对餐位、菜肴、酒水等基础数据进行管理，包括添加、修改、删除和查询等操作。确保系统中的基础数据准确无误，为餐厅的正常运营提供支持。

二、系统使用用户分析

角色是一类用户的概括，每种角色都可以有多个用户，每个用户都具有所属角色的系统功能。每个用户均是唯一的，都具有用户名和密码。每个独立的用户在具有所属角色的功能权限的基础上，还可以根据用户的具体需要配置有别于角色的特殊功能权限。

# 系统业务流程分析

整个餐饮运营业务围绕着顾客用餐流程展开，各岗位工作人员根据岗位职能的需要对顾客用餐相关信息进行处理，其核心操作是实现对顾客用餐信息数据的处理与流转操作，即完成点餐数据的采集、菜品制作进度的更新和用餐服务记录的管理操作，但不同岗位操作的数据内容有所差别。总体业务流程如图 1.1。

整个业务流程以学生信息为数据中心，对数据操作分为四个阶段：

* + 1. 点餐流程：消费者进店，服务员引导就座或消费者线上预订餐位后到店。顾客点餐，可选择菜品、规格及特殊要求，系统生成订单并分配餐位（现场点餐）或确认预订信息（线上预订），同时打印小票给顾客。
    2. 做菜流程：厨师按序查看小票准备食材并做菜，完成后更改菜品状态为 “已做”，系统通知服务员上菜。
    3. 服务流程：服务员查看小票提供服务，根据顾客要求增加菜品或酒水并录入系统，上菜后将菜品状态改为 “已出菜”。
    4. 结算流程：顾客用餐完毕，收银员对 “已出菜” 菜品结算，消费不足 500元收餐位费，支持多种支付方式，支付成功打印小票，日结时统计相关数据。
    5. 管理流程：餐厅老板查看运营数据，系统管理员管理基础数据及系统维护。

# 项目团队成员分工

（明确写出每个团队负责的内容和功能模块，后面的内容根据分工要求，每个项目组成员自己写自己的那一部分。同时，每一组除了需要提交每个成员自己的设计报告（纸质打印版）之外，每一组需要提交一份项目的整体设计报告（电子版））

刘治序：管理员,餐厅老板

傅嘉豪：厨师,服务员

潘俊杰：消费者,收银员

|  |
| --- |
|  |

图1.1 总体业务流程

## 1.5 系统业务功能及数据分析

根据整个系统的业务流程分析用户对整个系统的功能需求、数据需求及对数据处理的过程。

通过对业务流程的分析，整个系统包含5种类型的用户，不同类型的用户使用的系统功能不同，系统根据用户类型可以配置不同的功能。不同类型用户使用的功能见图1.2。

**图示

描述已自动生成**图1.2 角色功能需求

### 1.5.1 系统管理员功能需求

**1、系统设置与维护**

* **功能描述**

负责系统基础及日常维护。

* **数据描述**

系统参数、服务器信息、数据库备份路径。。

**2、餐位信息设置**

* **功能描述**

全面管理餐位，包括增删改查、状态设置与布局规划。

* **数据描述**

餐位编号、位置、类型、状态。

**3、菜肴信息设置**

* **功能描述**

维护菜肴各类信息，支持增删改操作。。

* 数据描述

菜名、价格、分类、库存、图片、食材等。

**4、用户权限设置**

* **功能描述**

创建与管理用户账号及权限，按需分配与调整。

* **数据描述**

用户名、密码、角色、权限列表。

### 1.5.2 收银员功能需求

**1、结账收银**

* **功能描述**

处理顾客结账，多种支付方式，提供清晰小票，更新订单状态。

* **数据描述：**

订单信息、支付方式、金额、结账时间。

**2、结算统计**

* **功能描述**

按时间段统计收银数据，统计菜品数量。

* **数据描述：**

统计时段、收入金额、菜品数量。

### 1.5.3 厨师功能需求

**1、订单查看与菜品制作**

* **功能描述**

依序查看订单，按要求制作菜品。

* **数据描述：**

订单编号、菜品信息。

**2、菜品状态更新**

* **功能描述**

时更新菜品状态，查看制作进度。

* **数据描述：**

菜品状态、更新时间、订单信息。

### 1.5.4 服务员功能需求

**1、点餐服务**

* **功能描述**

引导顾客点餐，记录需求，提交订单。

* **数据描述：**

桌号、人数、点餐时间、菜品信息。

**2、上菜服务**

* **功能描述**

及时上菜，提供相关服务。

* **数据描述：**

上菜时间、菜品名称、桌号。

### 1.5.5 餐厅老板需求

**1、数据总览与分析**

* **功能描述**

实时获取多类数据，展示分析，支持对比。

* **数据描述：**

数据类型、时间范围、展示形式。

**2、经营决策支持**

* **功能描述**

依数据分析决策，调整经营策略。

* **数据描述：**

决策类型、依据、时间、实施记录

### 1.5.6 顾客需求

**1、个人信息管理**

* **功能描述**

注册登录后管理个人资料，查看消费记录。

* **数据描述：**

姓名、账号、密码、消费记录。

**2、点餐操作**

* **功能描述**

通过多种方式（现场、线上）浏览菜品酒水信息，自由选择并提交订单，可备注特殊要求。

* **数据描述：**

点餐时间、所选菜品酒水、数量、特殊要求、联系方式。

# 第二章 系统设计

以餐厅运营管理需求为导向，将其转化为系统需实现的功能，构建出餐厅点餐结算系统功能架构。

## 2.1 系统功能设计

将用户功能需求转换为计算机软件系统需要实现的功能，整个《餐厅点餐结算系统》功能架构如图2.1。

 图2.1 系统总体功能架构

### 2.1.1 管理员子系统

**1、角色设置及权限配置**

* **功能描述**

完成角色的录入、修改，并设置角色的操作权限，即该角色能够使用的系统功能。

本功能主要分为：角色录入、角色修改、角色权限设置。

* **操作数据描述**

每个角色必须唯一，采用角色代码作为唯一标识，可操作的数据包含角色代码、角色名称、角色权限。角色权限用一个字符串表达，每个权限（即每个功能）用一个数字表达，多个权限之间用“，”分割。

**2、操作员设置及配置**

* **功能描述**

录入、修改操作员信息并分配角色，含操作员录入、修改、角色分配。保证操作员唯一，录入姓名等信息，通过角色分配确定权限，适应人员变动与权限调整。

* **操作数据描述**

工号唯一标识操作员，含姓名、加密密码、角色代码。如工号 “003” 操作员 “李四”，密码 “\*\*\*”，角色 “02”，登录后拥有收银员权限，可进行相关操作。

### 2.1.2 消费者子系统

**1、个人信息管理**

* **功能描述**

消费者注册时录入姓名、联系方式、等信息。

* **操作数据描述**

姓名为字符串，联系方式合规格式，密码加密存储，注册时间记录创建账号时刻。如姓名 “张三”，电话 “12312345678’’。

**2、菜品展示与分类管理**

* **功能描述**

分类展示菜品，自定义分类，上传菜品图片与详细信息，支持搜索推荐。帮助顾客选菜，突出菜品特色，提升点餐效率与体验，吸引顾客点餐。

* **操作数据描述**

菜品分类代码唯一，菜品 ID 唯一，含名称、价格、描述等。如菜品 “宫保鸡丁” ID “001”，价格 “28.00”，分量等。

**3、点餐操作与订单生成**

* **功能描述**

提供多种点餐方式，生成含详细信息的订单，自动计算总价，推送厨房并向顾客确认。

* **操作数据描述**

订单编号唯一，含点餐时间、顾客与餐位信息、菜品详情、总价、是否预约。如订单 “0001”，2023 - 09 - 01 12:00 点餐，顾客姓名 “王五”，餐位 “05”，菜品 “麻婆豆腐 2 份” 等，总价 “56.00”，“已预约”。

### 2.1.3 厨师子系统

**1、订单接收与处理**

* **功能描述**

实时接收订单，按序展示详情，支持标记分类筛选。助厨师安排任务，提高出餐效率，确保订单处理有序，满足顾客需求。

* **操作数据描述**

记录接收时间、订单编号、桌号与菜品信息，标记分类以代码表示。如 2024 - 12 - 05 12:10 接收订单 “0002”，桌号 “06”，菜品 “鱼香肉丝 1 份”以及顾客的备注等。

**2、菜品制作与进度更新**

* **功能描述**

厨师标记菜品制作进度，系统实时更新同步显示，完成后通知上菜并记录时间，便于服务员安排上菜。

* **操作数据描述**

进度以代码表示，记录完成时间、操作记录。如菜品 “宫保鸡丁” 进度 “正在烹饪”，2024 - 12 - 05 12:20 完成，操作记录含厨师信息与操作时间。

### 2.1.4 收银员子系统

**1、结账收银功能**

* **功能描述**

计算消费金额，支持多种支付方式，打印小票更新订单状态。

* **操作数据描述**

金额精确计算，支付方式代码表示，记录结算时间等。如消费 “80.00”，支付方式 “微信支付”，2024 - 12 - 05 13:00 结算，小票含订单 “0001” 详细消费信息

### 2.1.5 服务员子系统

**1、上菜服务功能**

* **功能描述**

处理顾客在用餐过程中的加菜、退菜、菜品问题反馈等需求。加菜时及时录入系统，退菜按规定流程处理并记录原因，对菜品问题反馈及时传达厨房并跟进解决，确保顾客问题得到妥善处理。

* **操作数据描述**

上菜时间精确记录，关联订单与菜品信息。如 2024 - 12 - 05 13:00 为订单 “0005” 上 “糖醋排骨”，同时记录顾客添加酒水 “青岛啤酒 2 瓶” 等服务需求操作。

**2、餐桌管理功能**

* **功能描述**

管理餐桌状态，包括空闲、占用、预订等。顾客用餐完毕及时清理餐桌，更新状态为空闲，为新顾客安排就座。

* **操作数据描述**

餐桌状态以代码表示，记录使用时长、清理时间等。如餐桌 “08” 状态 “占用”， 2024 - 12 - 05 13:00 清理后更新为 “空闲”，用于统计分析。

## 2.2 数据结构设计

### 2.2.1 整体数据模型

本系统涉及的数据信息主要包含销售数据、库存数据、员工绩效数据等。这些数据不是完全隔离的，二是相互之间有联系的。在数据结构中必须表达出这些数据之间的联系，以实现数据的共享，减少数据的冗余，保证数据的一致性。整个系统的数据模型如图2.3。

|  |
| --- |
|  |

图2.3 总体数据模型

该模型中，每个表格都是一个实体，每个实体均有相应的属性。消费者和结算支付之间是一对多的联系，也就是说，通过结算支付中的结算时间将结算支付和消费者联系起来，表达此时间的结算一定是特定的消费者。

### 2.2.2 数据结构设计

在C语言中采用结构体（struct）表达各个实体。将实体名转换为结构体名，实体的属性转为结构体的属性。

如将实体“角色”转换为结构体为：

struct Role {

int roleCode; // 角色代码（唯一）

string roleName; // 角色名称

string rolePermissions; // 角色权限

};

**操作员：**

struct Operator {

int operatorId; // 操作员工号(唯一)

string operatorName; // 操作员姓名

string password; // 操作员密码

int roleCode; // 角色代码

};

**菜品：**

struct Dish {

int dishId; // 菜品ID（唯一）

string dishName; // 菜品名称

double price; // 菜品价格

string description; // 菜品描述（分量）

bool dishStatus; //菜品状态

};

**订单：**

struct Order {

int orderId; // 订单编号（唯一）

string orderTime; // 点餐时间

int customerId; // 顾客ID

int tableNumber; // 桌号

bool isReserved; // 是否为预订订单

Dish dishes[100]; // 订单包含的菜品信息

double totalPrice; // 订单总价

};

**顾客：**

struct Customer {

int customerId; // 顾客ID(唯一)

string customerName; // 顾客姓名

string contactNumber; // 顾客联系电话

};

**餐桌：**

struct Table {

int tableNumber; // 桌号(唯一)

string tableStatus; // 餐桌状态，如空闲、占用、预订等

};

**支付方式：**

struct PaymentMethod {

int paymentMethodId; // 支付方式ID

string paymentMethodName; // 支付方式名称,如现金、微信支付、支付宝支付等

};

**结算：**

struct Settlement {

int settlementId; // 结算ID

int orderId; //订单表

string settlementTime; // 结算时间

int paymentMethodId; // 支付方式ID

double amount; // 结算金额

};

厨师：

struct Chef {

int chefId; // 厨师ID（唯一）

string chefName; // 厨师姓名

};

# 第三章 系统实现及测试

## 3.1 系统实现

本部分选取自己的核心模块，描述其实现的具体细节，包含流程图、源代码和部分界面。其它代码放在附件之中。

### 3.1.1 角色设置及权限配置

本功能模块主要完成角色的录入、修改，并设置角色的操作权限，即该角色能够使用的系统功能。流程图如图所示。



图3.1 角色设置及权限配置流程

首先根据上层菜单的功能选择“角色设置及权限配置”，展示本功能的下级功能菜单，根据选择的功能完成相应流程处理，实现角色设置及功能配置。其中保存是确认输入的信息是正确的信息，并将信息保存到内存和文件中。

完成本功能的主要代码如下：

……

当输入\*\*\*\*时，截图如下：

……

## 3.2 测试

软件测试是在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。是一种实际输出与预期输出之间的审核或者比较过程。一般分为单元测试、集成测试、确认测试、系统测试等。我们主要完成单元测试和集成测试。

单元测试是对软件组成单元进行测试，其目的是检验软件基本组成单位的正确性，测试的对象是软件设计的最小单位：模块。

集成测试也称联合测试，将程序模块采用适当的集成策略组装起来，对系统的接口及集成后的功能进行正确性检测的测试工作。其主要目的是检查软件单位之间的接口是否正确，集成测试的对象是已经经过单元测试的模块。

每种测试必须设计测试用例进行测试。测试用例的写法如表3.1。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 项目/软件 | 学生基本信息管理系统 | | 程序版本 | | V1.0 | | |
| 功能名称 | 角色修改 | | | | | | |
| 测试目的： | 测试角色信息是否能正常修改。 | | | | | | |
| 预置条件 | 选择角色修改功能 | | | | | | |
| 异常情况 | 修改数据不能保存 | | | | | | |
| 用例编号 | 相关用例 | 目的 | 操作步骤 | 输入数据 | 期望结果 | 执行结果 | 缺陷报告号 |
| 001 |  | 测试正确修改性 | 1根据角色名查询到需要修改的角色  2 修改角色  3 保存 | 新的角色名称 | 能正常修改 | 成功 |  |
| 002 |  | 测试重复的角色名 | 1 查找一个已有的角色名  2 查询另一个角色  3 将角色名修改为1中的角色名 | 已有的一个角色名 | 提示信息，不能修改为已有的角色名 | 错误 | 角色名称没有检查重复 |
| …… | …… | …… | …… | …… | …… | …… | …… |

# 第四章 总结

（说明：不但要对本次设计所做工作及收获进行总结，还要对所设计系统的不足进行总结）