**武汉纺织大学**

**《C语言程序设计》课程设计报告**

**食堂菜品信息管理系统**

**成 绩：**

**学 号： 2426010216**

**姓 名： 龙和怡**

**班 级： 数学类12402班**

**指导教师： 吴宛萍**

**报告日期： 2025年6月12日**

**1 题目与要求**

**题目：食堂菜品信息管理系统**

**1.1 问题提出**

食堂菜品种类繁多，食材用料对于学生来说并不透明，且口感参差不齐。本人计划编写一个食堂菜品信息管理系统，主要用来管理食堂菜品的基本信息及学生品尝后的反馈。

**1.2 功能要求**

1. 建立食堂每个菜品基本信息包括：窗口名称、菜品名称、食材用料、备菜时间、剩菜处理、学生反馈；
2. 管理员可以对菜品信息进行修改；

3、根据每位同学对菜品的评分反馈计算平均值；

4、排序：按平均分排序；

5、根据菜品名称或食材关键词查询评分；

6、输出：平均评分和菜品信息。

**2 功能设计**

**2.1 算法设计**

本系统需要实现的功能要求：

1. 利用switch语句设计如图1所示的主菜单：

学生主菜单（1～4）

1―― 查 询

2―― 输 出

3―― 添加反馈

4―― 退 出

管理员主菜单（1～6）

1―― 输入信息

2―― 修改信息

2―― 求平均分

3―― 排 序

4―― 查 询

5―― 输 出

6―― 退 出

图1 菜品信息管理员管理系统主菜单 图2 菜品信息用户系统主菜单

身份检验菜单

请输入选项编号（1～3）

1―― 选择身份

2―― 检验身份

3―― 执行相应身份的权利

排序菜单

请输入选项编号（1）

1―― 按平均分排序

…

图3 用户身份检验菜单 图4 排序子菜单

**2.2 模块图**

主菜单系统-全局调控

修改菜品文件信息

修改菜品文件信息

用户菜单

管理员菜单

计算平均分

修改删除菜品文件信息

添加菜品文件信息

菜品信息查询

添加用户反馈

排序

图5 菜品信息管理系统模块图

开始

**2.3 部分模块流程图**

比如身份检验模块：

选择身份

N

管理员密码

Y

该用户不是管理员

开启管理员权限

结束

**3 程序代码设计**

* **主控模块**

1. 函数原型：

int main() {

int choice;

loadData();

while(1) {

printf("\n请选择用户类型:\n");

printf("1. 普通用户\n2. 管理员登录\n0. 退出\n");

scanf("%d", &choice);

getchar();

switch(choice) {

case 1: showUserMenu(); break;

case 2: if(authenticateAdmin()) showAdminMenu(); break;

case 0: saveData(); exit(0);

default: printf("无效输入\n");

}

}

return 0;

}

2）功 能：系统入口，控制用户类型选择和程序退出；

协调各功能模块调用。

设计过程：初始化数据加载——>循环显示主菜单——>根据选择调用对应功能。

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 作用 |
| choice | int | 存储用户菜单选择 |
| dishes | Dish[MAX\_DISHES] | 存储所有菜品数据 |
| dishCount | int | 当前菜品数量 |
| isAdmin | bool | 管理员权限标志 |

* **普通用户**
* **数据存储模块**

1. 函数原型：

void saveData() {

FILE\* file = fopen(DATA\_FILE, "wb");

fwrite(&dishCount, sizeof(int), 1, file);

for(int i=0; i<dishCount; i++) {

fwrite(&dishes[i], sizeof(Dish), 1, file);

}

fclose(file);

}

1. 功 能：二进制文件的读写操作；

数据永久存储[2]。

设计过程：检查文件存在性——>二进制读写结构体数组——>包含头尾校验。

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 作用 |
| file | FILE\* | 文件指针 |
| count | int | 临时存储菜品信息 |

* **菜品管理模块**

1）函数原型：

void addDish() {

if(!isAdmin) return;

Dish newDish = {0};

printf("窗口名称: ");

fgets(newDish.window, 50, stdin);

// ...其他字段输入

dishes[dishCount++] = newDish;

saveData();

}

2）功 能：菜品的增删改查；

数据的合法性校验。

设计过程：输入数据验证——>数据操作管理——>实时保存数据[1]。

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 作用 |
| newDish | Dish | 临时存储新菜品 |
| dishName | char[50] | 存储查询的菜品名 |
| dishindex | int | 菜品数组索引 |

* **反馈管理模块**

1. 函数原型：

void addFeedback() {

Feedback fb;

do {

printf("评分(1-5): ");

scanf("%d", &fb.score);

} while(fb.score<1 || fb.score>5);

// ...添加反馈

// 实时计算平均分

float sum = 0;

for(int i=0; i<dish.feedbackCount; i++) {

sum += dish.feedbacks[i].score;

}

dish.avgScore = sum / dish.feedbackCount;

}

1. 功 能：添加学生评价；

实时评分计算[1]。

设计过程：菜品存在性检查——>评分范围校验——>动态计算平均分[1]

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 作用 |
| newFeedback | Feedback | 新反馈数据 |
| sum | float | 评分总和 |

* **权限控制模块[2]**

1）函数原型：

bool authenticateAdmin() {

char pwd[50];

printf("管理员密码: ");

disableEcho();

fgets(pwd, 50, stdin);

enableEcho();

return strcmp(pwd, ADMIN\_PASSWORD)==0;

}

1. 功 能：管理员身份验证；

终端回显控制。

设计过程：跨品台终端控制——>密码不会显处理——>字符串比对验证

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 作用 |
| password | Char[50] | 存储输入的密码 |
| term | struct termios | Linux终端设置 |

* **数据结构定义模块**

1）函数原型：

菜品结构体：

typedef struct {

char window[50]; // 窗口名称

char name[50]; // 菜品名称

char ingredients[200]; // 食材

char prepTime[20]; // 备餐时间

Feedback feedbacks[20];// 反馈数组

int feedbackCount; // 反馈数量

float avgScore; // 平均分

} Dish;

反馈结构体：

typedef struct {

char window[50]; // 窗口名称

char name[50]; // 菜品名称

char ingredients[200]; // 食材

char prepTime[20]; // 备餐时间

Feedback feedbacks[20];// 反馈数组

int feedbackCount; // 反馈数量

float avgScore; // 平均分

} Dish;

**3 总结**

1. 程序调试情况良好；
2. 本人在程序设计中感想：一整个系统的搭建是每一个小的模块构成的，他们彼此并不孤立。在书写代码时要保证代码的一致性，否则容易报错。在搭建系统时首先要构思系统框架确保功能的完善性。
3. 遇到的大问题：
4. 在管理员菜单模块，以我的知识储备无法实现终端密码不回显，通过询问AI，查询资料，得以实现。
5. 在最初搭建平台的时候，意识到我自己编码时使用的OS系统会导致代码有使用局限性，无法适配检查作业时用的windows系统。问AI解决了这个问题。
6. 在普通用户菜单中，输入新的反馈后，计算平均分这一步，我最开始设计的是，将平均分归零后再计算，却导致了计算结果为0，计算有误，于是更改成实时计算。
7. 在最初编写代码时，总是面临后续函数不能被读取又无法通过其他形式表出，因为所学知识实在有限，于是重新了解其他头函数，重新框定代码范围。

**4 结束语**

本次编码实属不易，由衷感谢deepseek给予的实时帮助、专业的回答及细心的讲解。他不知疲倦的倾情付出，我才能够顺利写出代码，完成系统检测！

**参考文献**

[1] 数据分析与算法结构- C语言描述（Mark Allen Weiss）

[2] 跨平台软件开发实践（清华大学出版社）