重庆师范大学

C++程序设计课程设计

、

项目名称： 菜单管理系统设计与实现

专业年级： 2023 计算机科学与技术

学生姓名： 谭祎（2023051603132）

穆江梅（2023051603125）

黄子幸（2023051603159）

蒋巧瑞（2023051603152）

蒋远余（2023051603128）

指导教师： 李晓虹

目录

0.小组分工

姓名	分工
谭祎	1、前端：将按钮和页面设置的部分函数写好放到 Mywidget.h 中； 完成前端基本代码架构和绘制方法、页面跳转基础代码，并给成员讲解； 2、后端：完成后端链表查找算法的编写，实现顾客端数据和管理员端数据的交互。 3、前后端交互：将管理员端的各项功能与链表连接起来。 4、文档：整体订正文档，格式优化。 5、打包安装包。
穆江梅	1、前端：完成欢迎界面和退出界面的绘制，绘制系统安装图标； 2、后端：完成后端链表增加、删除算法的编写，去食堂收集菜品数据。 3、前后端交互：将顾客端的各项功能与链表连接起来。 4、文档：完成《项目说明书》第 3 章。
蒋远粂	1、前端：完成登录界面的绘制，完成各界面背景图的设计； 2、后端：完成后端链表排序算法的编写。 3、前后端交互：将管理员端的各项功能与链表连接起来。 4、文档：完成《项目说明书》第 1、2 章。
黄子幸	1、前端：完成管理员界面的绘制； 2、后端：完成菜单结构体的定义和后端链表的定义，解决文件输入输出。 3、前后端交互：将管理员端的各项功能与链表连接起来。 4、文档：完成《项目说明书》第 4 章 4.1.1 和 4.2。
蒋巧瑞	1、前端：完成顾客界面的绘制，找好背景音乐并插入文件； 2、后端：定义头文件 Statement.h 中的函数声明，完成后端链表修改算法的编写。 3、前后端交互：将顾客端的各项功能与链表连接起来。 4、文档：完成《项目说明书》第 4 章 4.1.2。

1.系统分析

1.1 系统介绍

本系统旨在实现一个菜单管理系统，该系统包括菜单系统顾客端的信息录入、存储、查询、修改、删除，菜单系统顾客端的查看菜单、点菜等基本功能。此外，

系统还提供了统计分析功能，能够根据顾客消费情况分析不同类型顾客的消费习惯，并根据菜品的销售情况为相关厨师给予一定的奖励。在菜品信息方面，我们结合了重师食堂进行实地考察。系统采用用户登录界面认证，确保不同用户只能访问其权限范围内的功能。

1.2 系统功能需求

1.2.1 主界面

用户登录认证：我们的系统面向两个用户顾客和管理员，所以我们的主界面有两个不同的入口顾客端和管理端，通过账号和密码的输入进行认证和登录。

1.2.2 顾客端

查看菜单：用户通过此功能查看菜单信息。

点餐：用户通过输入菜品的名字，购买数量等菜品相关信息进行点餐。

消费情况查询：用户通过此功能可查询自己的点餐日期、菜品信息，以供用户了解自己的消费情况。

退出系统：以便用户快捷地退出系统。

1.2.3 管理端

新增/删减菜品：管理员通过此功能实现对菜单中菜品的新增和删减。

修改菜品信息：管理员通过此功能对菜品的信息进行修改。

查询菜品：以便管理员对菜品进行查询。

销量榜单查询：对菜品销量进行排序，以供管理员查询，方便对相关优秀厨师进行奖励。

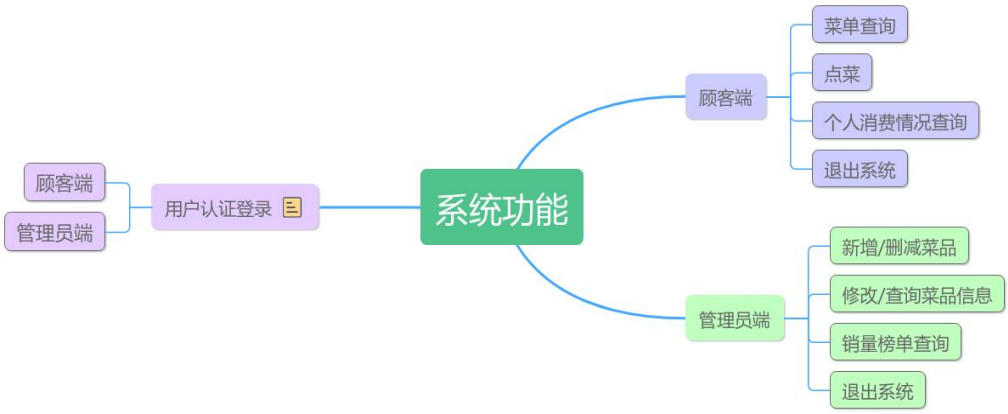

图 1-1 系统功能图

2.系统设计

2.1 系统总体设计

本系统采用了分层结构设计，包括数据层、逻辑层和表示层。数据层负责菜品信息的存储和管理，逻辑层负责处理业务逻辑，表示层使用 EasyX 库实现用户界面。

2.1.1 数据层设计

数据层采用了链表和 txt 文件来存储菜品信息。链表结构用于在内存中管理菜品信息，而 txt 文件用于将菜品信息持久化到磁盘中，以防止数据丢失。

2.1.2 逻辑层设计

逻辑层负责处理业务逻辑，包括菜品信息的录入、修改、删除、查询等操作。逻辑层与数据层进行交互，确保数据的一致性和完整性。

2.1.3 表示层设计

表示层使用 EasyX 库实现用户界面，包括用户登录认证，查询信息等。界面设计简洁明了，操作流畅，符合用户习惯。

2.2 模块设计

数据管理模块：负责菜品信息的存储和管理。我们采用了链表和 txt 的数据存储方式来实现。

用户界面模块：使用 EasyX 库实现用户友好的界面。

业务逻辑模块：处理菜品信息的录入、修改、删除、查询等业务逻辑。在顾客端和管理端的实现上，设计了两者的交互，通过数据层和逻辑层的交互实现。

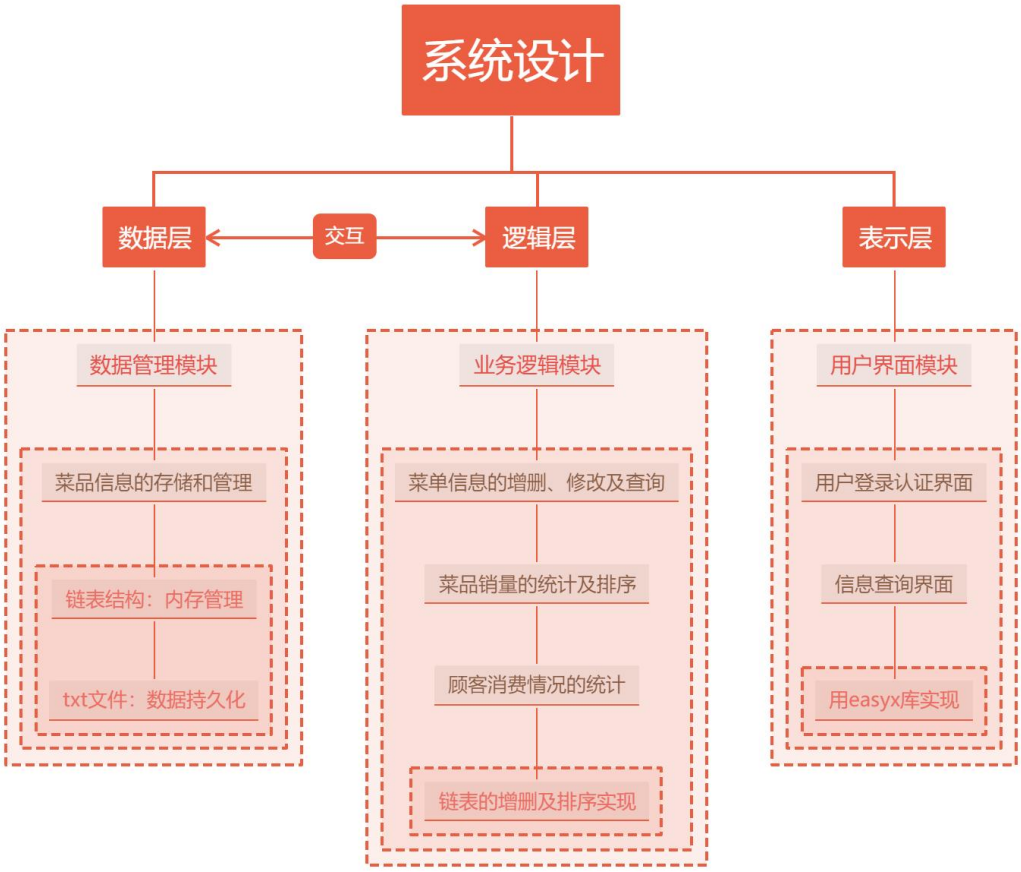

图 2-1 系统设计架构图

3.系统实现

3.1 程序代码

3.1.1 头文件

3.1.1.1 Mywidget.h

```
1  #ifndef MYWIDGET_H
2  #define MYWIDGET_H
```

```
3
4  #include <graphics.h> // 图形库
5  #include <string> // 处理字符串
6  #include <vector> // 处理向量（动态数组）
7  #include <map> // 处理映射（关联数组）
8  #include <tchar.h> // 处理字符和字符串
9  #include <conio.h> // 控制台输入输出
10 #include <cstdio> // 标准输入输出
11 #include <cstring> // 字符串处理函数
12 #include <mmsystem.h> // 多媒体系统函数
13 #include <windows.h> //  Windows API
14 #include <iostream> // 标准输入输出流
15 #include <fstream> // 文件流，
16 #include <locale> // 本地化库
17 #include <cstring> // 字符串处理
18 #include <sstream> // 字符串流
19 #pragma comment(lib, "Winmm.lib") // 链接多媒体库
20 using namespace std;
21
22
23 // 设置窗口位置
24 void setWindowPosition(HWND hwnd) // 定义一个函数，用于设置窗口位置
25 {
26     int screenWidth = GetSystemMetrics(SM_CXSCREEN); // 获取屏幕的宽度
27     int screenHeight = GetSystemMetrics(SM_CYSCREEN); // 获取屏幕的高度
28
29     RECT rect; // 定义一个 RECT 结构体，用于存储窗口的尺寸和位置
30     GetWindowRect(hwnd, &rect); // 获取窗口的当前尺寸和位置，将结果存储在 rect 结
构体中
31     int windowWidth = rect.right - rect.left; // 计算窗口的宽度
32     int windowHeight = rect.bottom - rect.top; // 计算窗口的高度
33
34     int x = (screenWidth - windowWidth) / 2; // 计算窗口左上角的 X 坐标，使其位
于屏幕中央
35     int y = (screenHeight - windowHeight) / 2; // 计算窗口左上角的 Y 坐标，使其
位于屏幕中央
36     MoveWindow(hwnd, x, y, rect.right - rect.left, rect.bottom - rect.top,
TRUE); // 使用 MoveWindow 函数将窗口移动到目标位置
37 }
38
39 // 定义一个按钮结构体，包含按钮的属性
40 struct Button {
41     int x, y, width, height, textSize; // 按钮的坐标、宽度、高度和文字大小
42     const TCHAR* text; // 按钮显示的文本
```

```
43     COLORREF textColor; // 按钮文字的颜色
44     COLORREF borderColor; // 按钮边框的颜色
45     COLORREF fillColor; // 按钮填充的颜色
46     COLORREF hoverColor; // 按钮悬停时的颜色
47     bool hovered; // 按钮是否悬停的状态
48 };
49
50 std::map<const TCHAR*, Button> buttonsMap; // 使用映射存储按钮
51
52 // 初始化按钮的属性
53 void initButton(Button& button, int x, int y, int width, int height, const
TCHAR* text, int textSize, COLORREF textColor, COLORREF borderColor, COLORREF
fillColor, COLORREF hoverColor) {
54     button.x = x; // 设置按钮的 X 坐标
55     button.y = y; // 设置按钮的 Y 坐标
56     button.width = width; // 设置按钮的宽度
57     button.height = height; // 设置按钮的高度
58     button.text = text; // 设置按钮的文本
59     button.textSize = textSize; // 设置按钮文本的大小
60     button.textColor = textColor; // 设置按钮文本的颜色
61     button.borderColor = borderColor; // 设置按钮边框的颜色
62     button.fillColor = fillColor; // 设置按钮填充的颜色
63     button.hoverColor = hoverColor; // 设置按钮悬停时的颜色
64     button.hovered = false; // 初始化按钮的悬停状态为假
65 }
66
67 // 绘制按钮
68 void drawButton(Button& button)
69 {
70     COLORREF currentFillColor = button.fillColor; // 设置当前填充颜色为按钮的填
充颜色
71     if (button.hovered) { // 如果按钮处于悬停状态
72         currentFillColor = button.hoverColor; // 设置当前填充颜色为悬停颜色
73     }
74
75     setlinecolor(button.borderColor); // 设置按钮边框的颜色
76     setfillcolor(currentFillColor); // 设置按钮填充的颜色
77     solidrectangle(button.x, button.y, button.x + button.width, button.y +
button.height); // 绘制实心矩形按钮
78
79     setbkmode(TRANSPARENT); // 设置背景模式为透明
80     settextcolor(button.textColor); // 设置文本颜色为按钮文本颜色
81     settextstyle(button.textSize, 0, _T("思源黑体")); // 设置文本样式和大小
82     int textWidth = textwidth(button.text); // 获取文本的宽度
```

```
    int textHeight = textheight(button.text); // 获取文本的高度
    int textX = button.x + (button.width - textWidth) / 2; // 计算文本的 X 坐标，使其在按钮中居中
    int textY = button.y + (button.height - textHeight) / 2; // 计算文本的 Y 坐标，使其在按钮中居中
    outtextxy(textX, textY, button.text); // 在指定位置绘制文本
}

// 检查鼠标是否悬停在按钮上
bool isMouseHover(Button& button)
{
    POINT mousePos; // 定义一个 POINT 结构体用于存储鼠标位置
    GetCursorPos(&mousePos); // 获取鼠标当前位置
    ScreenToClient(GetForegroundWindow(), &mousePos); // 将鼠标位置转换为窗口客户端坐标
    int mouseX = mousePos.x; // 获取鼠标的 X 坐标
    int mouseY = mousePos.y; // 获取鼠标的 Y 坐标
    // 检查鼠标是否在按钮范围内
    return (mouseX > button.x && mouseX < button.x + button.width && mouseY > button.y && mouseY < button.y + button.height);
}

// 检查按钮是否被点击
bool isButtonClicked(Button& button) {
    return isMouseHover(button) && (GetAsyncKeyState(VK_LBUTTON) & 0x8000); // 如果鼠标悬停在按钮上且左键按下，返回真
}

// 创建一个按钮并添加到映射中
void createButton(const TCHAR* name, int x, int y, int width, int height, const TCHAR* text, int textSize, COLORREF textColor, COLORREF borderColor, COLORREF fillColor, COLORREF hoverColor) {
    Button button; // 定义一个按钮对象
    initButton(button, x, y, width, height, text, textSize, textColor, borderColor, fillColor, hoverColor); // 初始化按钮属性
    buttonsMap[name] = button; // 将按钮添加到映射中
}

// 绘制所有按钮
void drawButtons(const IMAGE& background) {
    putimage(0, 0, &background); // 将背景图片绘制到窗口

    for (auto& pair : buttonsMap) { // 遍历映射中的每个按钮
        Button& button = pair.second; // 获取当前按钮
```

```
119       button.hovered = isMouseHover(button); // 检查按钮是否悬停
120       drawButton(button); // 绘制按钮
121    }
122
123    FlushBatchDraw(); // 刷新批量绘制的内容
124}
125
126#endif
```

3.1.1.2 Statement.h

```
1  #ifndef STATEMENT_H
2  #define STATEMENT_H
3  #define OK 1
4  #define ERROR 0
5
6  typedef int Status;
7  typedef int ElemType;
8
9  struct Dish
10 {
11     int id;                  // 编号
12     char name[200];            // 名称
13     double price;             // 价格
14     int sales;                // 销量
15     char canteen[200];          // 菜品所属食堂
16     char chefName[200];         // 制作菜品的厨师名称
17     bool chefIsExcellent;       // 这个厨师是否是优秀厨师
18 };
19
20 //链表定义
21 typedef struct LNode
22 {
23     Dish data;
24     struct LNode* next;
25 } LNode, * LinkList;
26
27 LinkList L;
28 LinkList L1;
29
30 bool IsDishNameDuplicate(LinkList L, const char* name);
31
32 //欢迎界面
33 void WelcomeInterface();
```

```
34
35 //登录界面
36 void LoginInterface();
37
38 /*顾客端*/
39 void Customermainface();//主界面
40 void printdish();
41 void Order(IMAGE& img);//点餐界面
42 void Consumeface();//查看消费情况界面
43
44 /*管理员*/
45 void Adminmainface();//主界面
46 void printmenu();
47 void Adddish();
48 void Deletedish();
49 void Changedish();
50 void Checkdish();
51 void Sortdish();
52
53 //退出页面
54 void Thanksface();
55
56 //系统后端（用链表存储）
57 Status InitList(LinkList& L);
58 Status ListInsert(LinkList& L, Dish dish);
59 Status ListDelete(LinkList& L, const char* name);
60 LNode* LocateElem(LinkList L, const char* name);
61 void ListSort(LinkList& L);
62 Status ListModify(LinkList& L, const char* name, Dish newdata);
63 Status ReadFromFile(LinkList& L, const char* filename);
64 Status WriteToFile(LinkList& L, const char* filename);
65
66 #endif
```

3.1.2 源文件

3.1.2.1 主函数

```
1  int main()
2  {
3      InitList(L);
4      InitList(L1);
5      ReadFromFile(L, "file\\dishes.txt");
```

```
6       ReadFromFile(L1, "file\\bill.txt");
7
8       // 循环播放音乐
9       mciSendString(_T("open \"music\\music.mp3\" type mpegvideo alias music"),
NULL, 0, NULL);
10      mciSendString(_T("play music repeat"), NULL, 0, NULL);
11
12      WelcomeInterface();
13      LoginInterface();
14      cout << "系统演示完毕！";
15
16      mciSendString(_T("close music"), NULL, 0, NULL);
17
18      return 0;
19  }
```

3.1.2.2 欢迎界面

```
1   void WelcomeInterface()
2   {
3       IMAGE img;
4       initgraph(800, 450);//欢迎窗口大小
5       loadimage(&img, _T("photo\\welcome.png"), 800, 450);//设置图片宽、高，填
满窗口
6       putimage(0, 0, &img);//图片显示的位置
7       HWND hwnd;// 设置窗口句柄变量保存窗口信息
8       hwnd = GetHWnd();// 获取窗口句柄
9       setWindowPosition(hwnd); //设置窗口位置
10      SetWindowText(hwnd, _T("欢迎使用我们的系统！")); //设置窗口标题
11      Sleep(3000);     //程序暂停 3 秒
12      closegraph();   //关闭图形页面
13  }
```

3.1.2.3 登陆界面

```
1   void LoginInterface()
2   {
3       // 初始化窗口大小
4       initgraph(1280, 720);
5       IMAGE img;
6
7       // 加载图片并设置宽高填满窗口
8       loadimage(&img, _T("photo\\login.png"), 1280, 720);
9       putimage(0, 0, &img); // 显示图片的位置
```

```
10
11      // 获取窗口句柄
12      HWND hwnd = GetHWnd();
13      setWindowPosition(hwnd);//设置窗口位置
14      SetWindowText(hwnd, _T("知“食”分子——重庆师范大学食堂菜单管理系统"));//设置窗口标题
15
16      // 创建按钮
17      createButton(_T("customerlogin"), 640, 100, 420, 210, _T("顾客登录"), 100, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
18      createButton(_T("adminlogin"), 640, 410, 420, 210, _T("管理员登录"), 100, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
19
20      BeginBatchDraw();//开始批量绘制
21
22      while (1) {
23          cleardevice();//清除设备
24          drawButtons(img);//绘制按钮
25          EndBatchDraw();//结束批量绘制
26
27          MOUSEMSG msg = GetMouseMsg();
28          msg = GetMouseMsg();// 获取鼠标消息
29          if (isButtonClicked(buttonsMap[_T("customerlogin")]) == 1)
30          {
31              //顾客登录
32
33              setbkcolor(WHITE);//设置背景颜色为白色
34
35              char Customeraccount[] = "Customer";//顾客账号
36              char Customerpassword[] = "123456";//顾客密码
37
38              // 显示输入框
39              char inputAccont[100];
40              char inputPassword[100];
41              //显示输入框并获取输入的账号
42              InputBox(inputAccont, 20, "账号验证", "请输入账号（默认为Customer）:", "你只能输入默认账号啦，哈哈", 0, 0, false);
43
44              // 检查账号
45              if (strcmp(inputAccont, Customeraccount) == 0)//若输入账号与顾客账号相同
46              {
47                  cleardevice();//清除设备上的内容
48                  drawButtons(img);//重新绘制按钮
```

```
                BeginBatchDraw();//开始批量绘制
                EndBatchDraw();//结束批量绘制
                MessageBox(hwnd, "存在该账号！", "账号验证", MB_OK | MB_ICONINFORMATION);//弹出提示框
                //显示输入框并获取输入的密码
                InputBox(inputPassword, 20, "密码验证", "请输入账号（默认为123456）：", "输入默认账号啦也就只能输入默认密码啦，哈哈", 0, 0, false);

                // 检查密码
                if (strcmp(inputPassword, Customerpassword) == 0)//若输入密码与顾客密码相同
                {
                    MessageBox(hwnd, "密码正确！", "密码验证", MB_OK | MB_ICONINFORMATION);
                    // 删除按钮，关闭窗口，跳转到主页面
                    auto d1 = buttonsMap.find("customerlogin"); //查找顾客登录按钮
                    auto d2 = buttonsMap.find("adminlogin"); //查找管理员登录按钮
                    if (d1 != buttonsMap.end() && d2 != buttonsMap.end()) //如果找到这两个按钮
                    {
                        buttonsMap.erase(d1);//删除顾客登录按钮
                        buttonsMap.erase(d2);//删除管理员登录按钮
                    }
                    //清空设备
                    cleardevice();
                    //关闭图形界面
                    closegraph();
                    //跳转到主页面
                    Customermainface();
                }
                else
                {
                    MessageBox(hwnd, "密码错误！", "密码验证", MB_OK | MB_ICONERROR);//弹出提示框
                    continue;
                }
                break;
            }
            else
            {
                MessageBox(hwnd, "不存在该账号，请重新输入", "账号验证", MB_OK | MB_ICONERROR);//弹出提示框
```

```
84                 break;
85             }
86         }
87         else if (isButtonClicked(buttonsMap[_T("adminlogin")]) == 1)
88         {
89             //管理员登录
90 
91             setbkcolor(WHITE);//设置背景颜色为白色
92 
93             //顾客的默认账号和密码
94             char Customeraccount[] = "Admin";
95             char Customerpassword[] = "123456";
96             // 显示输入框用于输入账号
97             char inputAccont[100];
98             char inputPassword[100];
99             InputBox(inputAccont, 20, "账号验证", "请输入账号（默认为Admin）：
", "你只能输入默认账号啦，哈哈", 0, 0, false);
100 
101             // 检查输入的账号是否与顾客的默认账号匹配
102             if (strcmp(inputAccont, Customeraccount) == 0)
103             {
104                 cleardevice();//清除设备上的内容
105                 drawButtons(img);//重新绘制按钮
106                 BeginBatchDraw();//开始批量绘制
107                 EndBatchDraw();//结束批量绘制
108                 MessageBox(hwnd, "存在该账号！", "账号验证", MB_OK |
MB_ICONINFORMATION);//显示消息框，告知用户账号存在
109                 InputBox(inputPassword, 20, "密码验证", "请输入账号（默认为
123456）：", "输入默认账号啦也就只能输入默认密码啦，哈哈", 0, 0, false);
110 
111                 // 检查输入的密码是否与顾客的默认密码匹配
112                 if (strcmp(inputPassword, Customerpassword) == 0)
113                 {
114                     MessageBox(hwnd, "密码正确！", "密码验证", MB_OK |
MB_ICONINFORMATION);
115 
116                     // 删除按钮，关闭窗口，跳转到主页面
117                     auto d1 = buttonsMap.find("customerlogin");
118                     auto d2 = buttonsMap.find("adminlogin");
119                     if (d1 != buttonsMap.end() && d2 != buttonsMap.end())
120                     {
121                         buttonsMap.erase(d1);
122                         buttonsMap.erase(d2);
123                     }
```

```
                    //清空设备
                    cleardevice();
                    //关闭图形界面
                    closegraph();
                    //跳转到管理员界面
                    Adminmainface();
                }
                else
                {
                    MessageBox(hwnd, "密码错误！", "密码验证", MB_OK | MB_ICONERROR);//显示消息框，告知用户密码错误
                    continue;
                }
                break;
            }
            else
            {
                MessageBox(hwnd, "不存在该账号，请重新输入", "账号验证", MB_OK | MB_ICONERROR);//显示消息框，告知用户账号不存在
                continue;
            }
        }
    }

    FlushBatchDraw();//刷新批量绘制的内容（确保所有绘制操作都显示在界面上）
    Sleep(30);//暂停 30 毫秒
}
```

3.1.2.4 系统顾客端

```
void Customermainface()
{
    initgraph(1280, 720);
    cleardevice();
    IMAGE img;

    loadimage(&img, _T("photo\\customer.png"), 1280, 720);
    putimage(0, 0, &img);

    HWND hwnd = GetHWnd();
    setWindowPosition(hwnd);
    SetWindowText(hwnd, _T("知“食”分子——重庆师范大学食堂菜单管理系统(顾客端)"));

```

```
    createButton(_T("ordermenu"), 530, 300, 210, 80, _T("查看菜单"), 50, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
    createButton(_T("order"), 530, 405, 210, 80, _T("点餐"), 50, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
    createButton(_T("consume"), 530, 510, 210, 80, _T("消费情况"), 50, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
    createButton(_T("exit"), 530, 615, 210, 80, _T("退出程序"), 50, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));

    BeginBatchDraw();
    while (1) {
        ExMessage msg = getmessage();
        cleardevice();
        drawButtons(img);
        EndBatchDraw();

        if (isButtonClicked(buttonsMap[_T("ordermenu")])) {
            cleardevice();
            printdish();
            break;
        }
        else if (isButtonClicked(buttonsMap[_T("order")])) {
            Order(img);
            cleardevice();
            continue;
        }
        else if (isButtonClicked(buttonsMap[_T("consume")])) {
            cleardevice();
            Consumeface();
            break;
        }
        else if (isButtonClicked(buttonsMap[_T("exit")])) {
            cleardevice();
            Thanksface();
            break;
        }
    }
}
void printdish()
{
    initgraph(1280, 720);
    cleardevice();
    IMAGE img;

```

```
54      loadimage(&img, _T("photo\\window.png"), 1280, 720);
55      putimage(0, 0, &img);
56
57      HWND hwnd1 = GetHWnd();
58      setWindowPosition(hwnd1);
59      SetWindowText(hwnd1, _T("知“食”分子——重庆师范大学食堂菜单管理系统(顾客
端)"));
60
61      setbkmode(OPAQUE);
62      setbkcolor(BLACK);
63      settextcolor(WHITE);
64      settextstyle(50, 0, _T("思源黑体"));
65      outtextxy(40, 100, _T("编号"));
66      outtextxy(170, 100, _T("菜名"));
67      outtextxy(300, 100, _T("价格"));
68      outtextxy(470, 100, _T("销量"));
69      outtextxy(600, 100, _T("所属食堂"));
70      outtextxy(830, 100, _T("厨师"));
71      outtextxy(960, 100, _T("是否为“优秀厨师”"));
72
73      // 绘制菜单内容
74      LinkList p = L->next;
75      int y = 150;
76      while (p) {
77          setbkmode(TRANSPARENT);
78          settextcolor(BLACK);
79          settextstyle(30, 0, _T("思源黑体"));
80
81          TCHAR buffer[50];
82          _stprintf_s(buffer, _T("%d"), p->data.id);
83          outtextxy(40, y, buffer);
84          outtextxy(170, y, p->data.name);
85          _stprintf_s(buffer, _T("%.2f"), p->data.price);
86          outtextxy(300, y, buffer);
87          _stprintf_s(buffer, _T("%d"), p->data.sales);
88          outtextxy(470, y, buffer);
89          outtextxy(600, y, p->data.canteen);
90          outtextxy(830, y, p->data.chefName);
91          outtextxy(960, y, p->data.chefIsExcellent ? _T("Yes") : _T("No"));
92          y += 20;
93          p = p->next;
94      }
95
96      setfillcolor(WHITE);
```

```
97      fillrectangle(1060, 620, 1260, 700);
98
99      // 绘制按钮文本
100     setbkmode(TRANSPARENT);
101     settextcolor(BLACK);
102     settextstyle(50, 0, _T("思源黑体"));
103     outtextxy(1125, 635, _T("返回"));
104
105     while (1) {
106         ExMessage m2 = getmessage();
107
108         if (m2.x > 1060 && m2.x < 1260 && m2.y > 620 && m2.y < 700)
109         {
110             setlinecolor(RED);
111             rectangle(1060, 620, 1260, 700);
112         }
113         else {
114             setlinecolor(BLACK);
115             rectangle(1060, 620, 1260, 700);
116         }
117         if (m2.message == WM_LBUTTONDOWN) {
118             cleardevice();
119             Customermainface();
120             break;
121         }
122     }
123}
124//点餐功能
125void Order(IMAGE& img)
126{
127     // 新增消费记录
128     Dish newDish;
129     char inputBuffer[50];
130
131     // 输入名称并检查重复
132     while (true)
133     {
134         InputBox(newDish.name, 50, "请输入所需菜品的名称");
135         if (!IsDishNameDuplicate(L, newDish.name))
136         {
137             HWND hwndInput1 = GetHWnd();
138             MessageBox(hwndInput1, "该菜品不存在！请重新输入", "提示", MB_OK);
139         }
140         else
```

```
141         {
142             break;
143         }
144     }
145 
146     // 输入编号
147     InputBox(inputBuffer, 10, "请输入购买菜品的日期（案例：240601）");
148     newDish.id = atoi(inputBuffer);
149 
150     // 输入价格
151     InputBox(inputBuffer, 10, "请输入所需菜品的价格");
152     newDish.price = atof(inputBuffer);
153 
154     InputBox(inputBuffer, 10, "请输入购买菜品的份数");
155     newDish.sales = atof(inputBuffer);
156     //保证管理员端销量数据更新
157     LNode* p = L->next;
158     while (p)
159     {
160         if (strcmp(p->data.name, newDish.name) == 0)
161         {
162             p->data.sales = p->data.sales + newDish.sales;
163             break;
164         }
165         p = p->next;
166     }
167 
168     // 输入食堂
169     InputBox(newDish.canteen, 50, "请输入新增菜品所属食堂");
170 
171     // 输入厨师名称
172     InputBox(newDish.chefName, 50, "请输入制作菜品的厨师名称");
173 
174     // 输入菜品是否好吃
175     InputBox(inputBuffer, 10, "请问菜品好吃吗（1 表示是，0 表示否）");
176     newDish.chefIsExcellent = (atoi(inputBuffer) == 1);
177 
178     // 插入链表
179     ListInsert(L1, newDish);
180     WriteToFile(L1, "file\\bill.txt");
181     WriteToFile(L, "file\\dishes.txt");
182 
183     HWND hwnd = GetHWnd();
184     MessageBox(hwnd, "点菜成功！", "提示", MB_OK);
```

```
185
186    // 清屏并重新绘制之前的内容
187    cleardevice();
188    putimage(0, 0, &img);
189
190}
191
192//消费情况界面
193void Consumeface()
194{
195    initgraph(1280, 720);
196    cleardevice();
197    IMAGE img;
198
199    loadimage(&img, _T("photo\\window.png"), 1280, 720);
200    putimage(0, 0, &img);
201
202    HWND hwnd1 = GetHWnd();
203    setWindowPosition(hwnd1);
204    SetWindowText(hwnd1, _T("知“食”分子——重庆师范大学食堂菜单管理系统(顾客
端)"));
205
206    setbkmode(OPAQUE);
207    setbkcolor(BLACK);
208    settextcolor(WHITE);
209    settextstyle(50, 0, _T("思源黑体"));
210    outtextxy(40, 100, _T("日期"));
211    outtextxy(170, 100, _T("菜名"));
212    outtextxy(300, 100, _T("价格"));
213    outtextxy(470, 100, _T("购买份数"));
214    outtextxy(600, 100, _T("所属食堂"));
215    outtextxy(830, 100, _T("厨师"));
216    outtextxy(960, 100, _T("是否好吃"));
217
218    // 绘制菜单内容
219    LinkList p = L1->next;
220    int y = 150;
221    while (p) {
222        setbkmode(TRANSPARENT);
223        settextcolor(BLACK);
224        settextstyle(30, 0, _T("思源黑体"));
225
226        TCHAR buffer[50];
227        _stprintf_s(buffer, _T("%d"), p->data.id);
```

```
228         outtextxy(40, y, buffer);
229         outtextxy(170, y, p->data.name);
230         _stprintf_s(buffer, _T("%.2f"), p->data.price);
231         outtextxy(300, y, buffer);
232         _stprintf_s(buffer, _T("%d"), p->data.sales);
233         outtextxy(470, y, buffer);
234         outtextxy(600, y, p->data.canteen);
235         outtextxy(830, y, p->data.chefName);
236         outtextxy(960, y, p->data.chefIsExcellent ? _T("Yes") : _T("No"));
237         y += 20;
238         p = p->next;
239     }
240 
241     setfillcolor(WHITE);
242     fillrectangle(1060, 620, 1260, 700);
243 
244     // 绘制按钮文本
245     setbkmode(TRANSPARENT);
246     settextcolor(BLACK);
247     settextstyle(50, 0, _T("思源黑体"));
248     outtextxy(1125, 635, _T("返回"));
249 
250     while (1) {
251         ExMessage m2 = getmessage();
252 
253         if (m2.x > 1060 && m2.x < 1260 && m2.y > 620 && m2.y < 700)
254         {
255             setlinecolor(RED);
256             rectangle(1060, 620, 1260, 700);
257         }
258         else {
259             setlinecolor(BLACK);
260             rectangle(1060, 620, 1260, 700);
261         }
262         if (m2.message == WM_LBUTTONDOWN)
263         {
264             cleardevice();
265             Customermainface();
266             break;
267         }
268     }
269}
```

3.1.2.5 系统管理员端

```
1  //主界面
2  void Adminmainface()
3  {
4      /*cleardevice();
5      Sleep(10);
6      closegraph();*/
7
8      // 初始化窗口大小
9      initgraph(1280, 720);
10     cleardevice();
11     IMAGE img;
12
13     // 加载图片并设置宽高填满窗口
14     loadimage(&img, _T("photo\\admin.png"), 1280, 720);
15     putimage(0, 0, &img); // 显示图片的位置
16
17     // 获取窗口句柄
18     HWND hwnd = GetHWnd();
19     setWindowPosition(hwnd);
20     SetWindowText(hwnd, _T("知“食”分子——重庆师范大学食堂菜单管理系统(管理员端)"));//设置窗口标题
21
22     // 创建按钮
23     createButton(_T("add"), 530, 300, 210, 60, _T("新增菜品"), 40, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
24     createButton(_T("delete"), 530, 370, 210, 60, _T("删除菜品"), 40, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
25     createButton(_T("change"), 530, 440, 210, 60, _T("信息修改"), 40, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
26     createButton(_T("check"), 530, 510, 210, 60, _T("信息查询"), 40, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
27     createButton(_T("sort"), 530, 580, 210, 60, _T("销量榜单"), 40, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
28     createButton(_T("exit"), 530, 650, 210, 60, _T("退出程序"), 40, WHITE, BLACK, RGB(72, 116, 203), RGB(0, 32, 96));
29
30     BeginBatchDraw();//批量绘制
31
32     while (1) {
33         cleardevice();
34         drawButtons(img);
35
36         MOUSEMSG msg = GetMouseMsg();
```

```
        msg = GetMouseMsg();// 获取鼠标消息
        if (isButtonClicked(buttonsMap[_T("add")]) == 1) {//如果新增菜品按钮被点击
            cleardevice();//清空设备
            drawButtons(img);//绘制按钮
            BeginBatchDraw();//开始批量绘制
            EndBatchDraw();//结束批量绘制
            Adddish();//调用新增菜品函数
        }
        else if (isButtonClicked(buttonsMap[_T("delete")]) == 1)//如果删除菜品按钮被点击
        {
            cleardevice();
            drawButtons(img);
            BeginBatchDraw();
            EndBatchDraw();
            Deletedish();//调用删除菜品函数
        }
        else if (isButtonClicked(buttonsMap[_T("change")]) == 1)//如果信息修改按钮被点击
        {
            cleardevice();
            drawButtons(img);
            BeginBatchDraw();
            EndBatchDraw();
            Changedish();//调用信息修改函数
        }
        else if (isButtonClicked(buttonsMap[_T("check")]) == 1)//如果信息查询按钮被点击
        {
            cleardevice();
            drawButtons(img);
            BeginBatchDraw();
            EndBatchDraw();
            Checkdish();//调用信息查询函数

        }
        else if (isButtonClicked(buttonsMap[_T("sort")]) == 1)//如果销量榜单按钮被点击
        {
            cleardevice();
            drawButtons(img);
            BeginBatchDraw();
            EndBatchDraw();
```

```
76              Sortdish();//调用销量榜单函数
77          }
78          else if (isButtonClicked(buttonsMap[_T("exit")]) == 1)//如果退出程序
按钮被点击
79          {
80              cleardevice();
81              drawButtons(img);
82              BeginBatchDraw();
83              EndBatchDraw();
84              Thanksface();//调用退出程序函数
85          }
86      }
87
88      FlushBatchDraw();
89      Sleep(30);
90  }
91  //打印菜单
92  void printmenu()
93  {
94      // 初始化窗口大小
95      initgraph(1280, 720);
96      cleardevice();
97      IMAGE img;
98
99      // 加载图片并设置宽高填满窗口
100     loadimage(&img, _T("photo\\window.png"), 1280, 720);
101     putimage(0, 0, &img); // 显示图片的位置
102
103     // 获取窗口句柄
104     HWND hwnd = GetHWnd();
105     setWindowPosition(hwnd);//设置窗口位置
106     SetWindowText(hwnd, _T("知“食”分子——重庆师范大学食堂菜单管理系统(管理员
端)"));//设置窗口标题
107
108     setbkmode(OPAQUE); // 设置字体背景模式为不透明
109     setbkcolor(BLACK); // 设置背景颜色为黑色
110     settextcolor(WHITE); // 设置字体颜色为白色
111     settextstyle(50, 0, "思源黑体");//设置字体样式和大小
112     outtextxy(40, 100, "编号");//打印标号标题
113     outtextxy(170, 100, "菜名");//打印菜名标题
114     outtextxy(300, 100, "价格");//打印价格标题
115     outtextxy(470, 100, "销量");//打印销量标题
116     outtextxy(600, 100, "所属食堂");//打印所属食堂标题
117     outtextxy(830, 100, "厨师");//打印厨师标题
```

```
118    outtextxy(960, 100, "是否为“优秀厨师”");//打印是否为“优秀厨师”标题
119
120    LinkList p = L->next;//获取链表中的第一个菜品的指针
121    int y = 150; // 初始 y 坐标，用于在屏幕上垂直打印菜品信息
122    while (p) //遍历链表
123    {
124        setbkmode(TRANSPARENT);
125        settextcolor(BLACK); // 设置字体颜色为黑色
126        settextstyle(30, 0, "思源黑体"); //设置字体样式
127
128        char buffer[50];
129        // 将整型和浮点型转换为字符串，并调用 outtextxy 函数在置顶位置输出
130        sprintf_s(buffer, sizeof(buffer), "%d", p->data.id);
131        outtextxy(40, y, buffer);
132        outtextxy(170, y, p->data.name);
133        sprintf_s(buffer, sizeof(buffer), "%.2f", p->data.price);
134        outtextxy(300, y, buffer);
135        sprintf_s(buffer, sizeof(buffer), "%d", p->data.sales);
136        outtextxy(470, y, buffer);
137        outtextxy(600, y, p->data.canteen);
138        outtextxy(830, y, p->data.chefName);
139        outtextxy(960, y, p->data.chefIsExcellent ? "Yes" : "No");
140        y += 20; //增加 y 坐标，换行显示
141        p = p->next;
142    }
143    setfillcolor(WHITE); //设置填充颜色为白色
144    fillrectangle(300, 380, 380, 410); //绘制一个矩形框
145    while (1) {
146        ExMessage m2 = getmessage();
147
148        if (m2.x > 300 && m2.x < 380 && m2.y>380 && m2.y < 410) { //判断用
户点击位置是否在矩形框内
149            setlinecolor(RED);  //在则设置线框颜色为红色
150            rectangle(300, 380, 380, 410);
151        }
152        else {
153            setlinecolor(WHITE);  //不在则设置线框颜色为白色
154            rectangle(300, 380, 380, 410);
155        }
156        if (m2.message == WM_LBUTTONDOWN) { //若用户按下鼠标左键
157            Adminmainface(); //进入管理人员主界面
158        }
159    }
160}
```

```
161//新增菜品功能
162bool IsDishNameDuplicate(LinkList L, const char* name)
163{
164    // 检查是否存在该菜品（有返回 1，无返回 0）
165    LinkList p = L->next;
166    while (p) {
167        //判断是否有与输入名称相同的菜品
168        if (strcmp(p->data.name, name) == 0)
169        {
170            return true;
171        }
172        p = p->next;
173    }
174    return false;
175}
176
177void Adddish()
178{
179    // 新增菜品
180    Dish newDish;
181    char inputBuffer[50];
182    // 输入名称并检查重复
183    while (true)
184    {
185        InputBox(newDish.name, 50, "请输入新增菜品的名称");//弹出输入框让用户输
入菜品名称
186        if (IsDishNameDuplicate(L, newDish.name)) //检查链表中是否已有相同名称
的菜品
187        {
188            HWND hwndInput1 = GetHWnd();//获得当前窗口句柄
189            MessageBox(hwndInput1, "该菜品已经存在，请重新输入", "提示",
MB_OK);//弹出消息框
190        }
191        else
192        {
193            break;//如果菜品名不存在，跳出循环
194        }
195    }
196
197    // 输入编号
198    InputBox(inputBuffer, 10, "请输入新增菜品的编号");//弹出输入框让用户输入菜品
编号
199    newDish.id = atoi(inputBuffer);//将输入的字符串转换为整型编号
200
```

```
201
202     // 输入价格
203     InputBox(inputBuffer, 10, "请输入新增菜品的价格");//弹出输入框让用户输入菜品
价格
204     newDish.price = atof(inputBuffer);//将输入的字符串转换为浮点型价格
205
206     // 默认销量为 0
207     newDish.sales = 0;
208
209     // 输入食堂
210     InputBox(newDish.canteen, 50, "请输入新增菜品所属食堂");//弹出输入框让用户输
入菜品所属食堂
211
212     // 输入厨师姓名
213     InputBox(newDish.chefName, 50, "请输入制作菜品的厨师名称");//弹出输入框让用
户输入制作菜品的厨师姓名
214
215     // 输入厨师是否优秀
216     InputBox(inputBuffer, 10, "请输入厨师是否优秀（1 表示是，0 表示否）");//弹出输
入框让用户输入厨师是否优秀
217     newDish.chefIsExcellent = (atoi(inputBuffer) == 1);//将输入的字符串转换为
布尔值
218
219     // 将新菜插入链表
220     ListInsert(L, newDish);//将新菜品插入链表中
221     WriteToFile(L, "file\\dishes.txt");  //将链表写入文件
222     printmenu();  //打印菜单
223
224}
225//删除菜品
226void Deletedish()
227{
228     char input[50];
229     InputBox(input, 50, "请输入要删除的菜品名称");//弹出输入框让用户输入要删除的菜
品名称
230     if (IsDishNameDuplicate(L, input) == 0) //判断菜品是否存在
231     {
232         HWND hwndInput1 = GetHWnd();//获取当前窗口句柄
233         MessageBox(hwndInput1, "该菜品不存在！请重新输入", "提示", MB_OK);//弹
出消息框提示菜品不存在
234     }
235     ListDelete(L, input); //从链表中删除指定菜品
236     WriteToFile(L, "file\\dishes.txt"); //将链表写入文件
237     HWND hwnd = GetHWnd();//获取当前窗口句柄
```

```
238     MessageBox(hwnd, "删除成功！", "提示", MB_OK);//弹出消息框提示删除成功
239
240}
241//修改菜品
242void Changedish() {
243     char input[50];
244     InputBox(input, 50, "请输入要修改的菜品名称");//弹出消息框提示要修改的菜品名称
245     if (!IsDishNameDuplicate(L, input)) { //判断菜品是否存在
246         HWND hwndInput1 = GetHWnd();//获取当前窗口句柄
247         MessageBox(hwndInput1, "该菜品不存在！请重新输入", "提示", MB_OK);//弹出消息框提示菜品不存在
248     }
249     else {
250         Dish newDish;
251         InputBox(newDish.name, 50, "请输入新的菜品名称");//弹出输入框让用户输入新的菜品名称
252         char buffer[50];
253
254         InputBox(buffer, 50, "请输入新的菜品编号");//弹出输入框让用户输入新的菜品编号
255         newDish.id = atoi(buffer);//将输入的字符串转换为整型编号
256
257         InputBox(buffer, 50, "请输入新的菜品价格");//弹出输入框让用户输入菜品价格
258         newDish.price = atof(buffer);//将输入的字符串转换为浮点型价格
259
260         InputBox(buffer, 50, "请输入新的菜品销量");//弹出输入框让用户输入菜品销量
261         newDish.sales = atoi(buffer);//将输入的字符串转换为整型销量
262
263         InputBox(newDish.canteen, 50, "请输入新的菜品所属食堂");//弹出输入框让用户输入菜品所属食堂
264         InputBox(newDish.chefName, 50, "请输入新的制作菜品的厨师名称");//弹出输入框让用户输入制作菜品的厨师姓名
265
266         InputBox(buffer, 50, "请输入新的厨师是否为优秀厨师 (0 或 1)");//弹出输入框让用户输入厨师是否优秀
267         newDish.chefIsExcellent = atoi(buffer);//将输入的字符串转换为布尔值，1 表示优秀，0 表示不优秀
268
269         if (ListModify(L, input, newDish) == OK) {//尝试修改链表中的菜品信息
270             HWND hwndInput2 = GetHWnd();//获取当前窗口句柄
271             MessageBox(hwndInput2, "菜品修改成功！", "提示", MB_OK);//弹出消息框提示修改成功
272         }
273         else {
```

```
274            HWND hwndInput2 = GetHWnd();//获取当前窗口句柄
275            MessageBox(hwndInput2, "菜品修改失败！", "提示", MB_OK);//弹出消息
框提示修改失败
276        }
277    }
278}
279//查询菜品
280void Checkdish()
281{
282    char input[50];
283    InputBox(input, 50, "请输入要查询的菜品名称");//弹出输入框让用户输入要查询的菜
品名称
284    LNode* result = LocateElem(L, input);  //在链表中查找指定菜品
285    std::ostringstream oss;//创建一个字符串流用于构建消息框内容
286    if (result)
287    {
288        //将菜品编号、菜品名称、菜品价格、菜品销量等信息写入字符串流
289        oss << "编号: " << result->data.id << "\n";
290        oss << "名称: " << result->data.name << "\n";
291        oss << "价格: " << result->data.price << "\n";
292        oss << "销量: " << result->data.sales << "\n";
293        oss << "食堂: " << result->data.canteen << "\n";
294        oss << "厨师: " << result->data.chefName << "\n";
295        oss << "优秀厨师: " << (result->data.chefIsExcellent ? "是" : "否") <<
"\n";
296        MessageBox(GetHWnd(), oss.str().c_str(), "查询结果", MB_OK);//弹出消
息框显示查询结果
297    }
298    else
299    {
300        MessageBox(GetHWnd(), "未找到该菜品", "查询结果", MB_OK);//弹出消息框提
示未找到该菜品
301    }
302}
303
304//销量榜单界面
305void Sortdish()
306{
307    ListSort(L); //对链表进行排序
308    WriteToFile(L, "file\\dishes.txt"); //将链表写入文件
309    printmenu();   //打印菜单
310
311}
```

3.1.2.6 退出界面

```
1  void Thanksface()
2  {
3      IMAGE img;
4      initgraph(800, 450);//退出窗口大小
5      loadimage(&img, _T("photo\\byebye.png"), 800, 450);//设置图片宽、高，填满
窗口
6      putimage(0, 0, &img);//图片显示的位置
7      HWND hwnd;// 设置窗口句柄变量保存窗口信息
8      hwnd = GetHWnd();// 获取窗口句柄
9      setWindowPosition(hwnd);//设置窗口位置
10     SetWindowText(hwnd, _T("感谢使用我们的系统！"));//设置窗口标题
11     Sleep(6000); //程序暂停 6 秒
12     closegraph(); //关闭图形界面
13 }
```

3.1.2.7 系统后端（用链表存储数据）

```
1  // 初始化
2  Status InitList(LinkList& L)
3  {
4      L = new LNode; //创建新节点作为链表头
5      L->next = NULL;//设置头节点的下一个指针为空
6      return OK;
7  }
8
9
10 // 在链表最后插入新节点
11 Status ListInsert(LinkList& L, Dish dish)
12 {
13     LinkList p = L;//p 指针指向头节点
14     while (p->next)   //遍历找到尾结点
15     {
16         p = p->next;
17     }
18     LinkList s = new LNode; //创建新节点 s
19     s->data = dish; //将菜品信息赋值给新节点 s 的数据域
20     s->next = NULL;
21     p->next = s;  //结点 p 向后插入新节点 s
22     return OK;
23 }
24
25 // 删除链表中 name 属性等于指定值的节点
26 Status ListDelete(LinkList& L, const char* name)
```

```
{
    LinkList p = L;
    //遍历链表，判断该节点的下一个结点值是否是需要删除的节点
    while (p->next && strcmp(p->next->data.name, name) != 0)
    {
        p = p->next;
    }
    if (!p->next) return ERROR;
    LinkList q = p->next;   //找到需要删除的节点
    p->next = q->next;   //将需要删除的节点从链表中移除
    delete q;     //释放节点内存
    return OK;
}

// 查找指定 name 的节点
LNode* LocateElem(LinkList L, const char* name)
{
    LinkList p = L->next;
    while (p && strcmp(p->data.name, name) != 0)  //遍历并判断链表节点是否存在该值
    {
        p = p->next;
    }
    return p; //返回找到的节点
}

// 插入排序（根据 sales 从大到小排序）
void ListSort(LinkList& L)
{
    if (!L || !L->next || !L->next->next)
    {
        return; // 如果链表为空或只有一个元素，无需排序
    }
    LNode* sorted = NULL; // 有序链表

    while (L->next != NULL)
    {
        // 从原链表中取下第一个节点
        LNode* p = L->next;
        L->next = p->next;//移除原链表中的第一个节点

        // 插入到有序链表中
        if (sorted == NULL || sorted->data.sales < p->data.sales)
        {
```

```
70              // 插入到有序链表的开头
71              p->next = sorted;
72              sorted = p;
73          }
74          else {
75              // 在有序链表中找到插入的位置
76              LNode* q = sorted;
77              while (q->next != NULL && q->next->data.sales >= p->data.sales)
78              {
79                  q = q->next;
80              }
81              p->next = q->next;
82              q->next = p;
83          }
84      }
85      L->next = sorted; //将排序后的链表重新连接原链表上
86  }
87
88  // 修改（链表中指定 name 的节点数据）
89  Status ListModify(LinkList& L, const char* name, Dish newdata)
90  {
91      LinkList p = L->next;
92      while (p != NULL)   //遍历链表
93      {
94          if (strcmp(p->data.name, name) == 0)   //判断链表节点是否存在该值
95          {
96              p->data = newdata;        //存在就修改该值为 newdata
97              return OK;
98          }
99          p = p->next;
100     }
101     return ERROR;
102 }
103
104 // 从文件读取链表数据
105 Status ReadFromFile(LinkList& L, const char* filename)
106 {
107     ifstream inFile(filename); //打开文件
108     if (!inFile)
109     {
110         cerr << "文件打开失败！" << endl;
111         return ERROR;
112     }
113     //读取文件中的每个数据，并将其存储在链表中
```

```
114    Dish dish;
115    while (inFile >> dish.id >> dish.name >> dish.price >> dish.sales >>
dish.canteen >> dish.chefName >> dish.chefIsExcellent)
116    {
117        ListInsert(L, dish); //将读取到的数据插入到链表中
118    }
119    inFile.close(); //关闭文件
120    return OK;
121}
122
123// 将链表数据写入文件
124Status WriteToFile(LinkList& L, const char* filename)
125{
126    ofstream outFile(filename); //打开文件
127    if (!outFile) {
128        cout << "文件打开失败！" << endl;
129        return ERROR;
130    }
131    //遍历链表并将数据写入文件
132    LinkList p = L->next;
133    while (p) {
134        outFile << p->data.id << " " << p->data.name << " " << p->data.price
<< " " << p->data.sales << " "
135            << p->data.canteen << " " << p->data.chefName << " " <<
p->data.chefIsExcellent << endl;//将每个节点的数据写入文件，每个字段之间用空格分隔
136        p = p->next;
137    }
138    outFile.close();   //关闭文件，完成写入
139    return OK;
140}
```

3.2 运行界面

3.2.1 欢迎

图 3-1 系统欢迎界面

3.2.2 登录

图 3-2 登录界面

3.2.3 顾客端

图 3-3 顾客端主界面

图 3-4 查看菜单功能

图 3-5 点菜功能

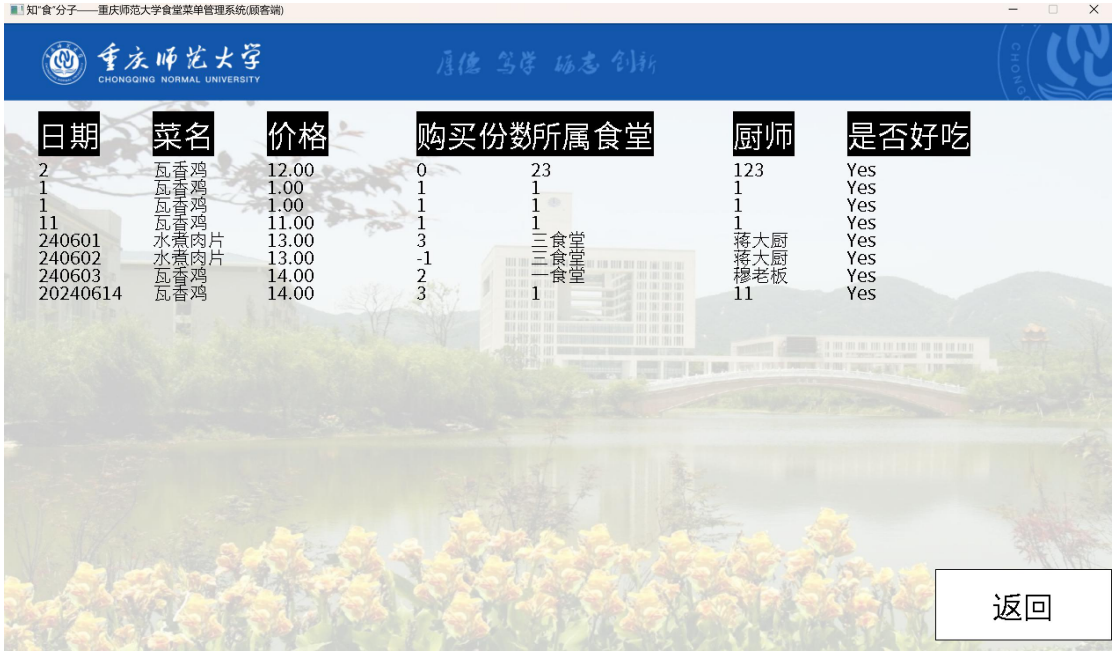

图 3-6 查看消费情况功能

3.2.4 管理员端

图 3-7 管理员端主界面

图 3-8 新增菜品功能

图 3-9 删除菜品功能

图 3-10 信息修改功能

图 3-11 信息查询功能

图 3-12 销量榜单界面

3.2.5 退出程序界面

图 3-13 退出程序界面

4.系统评价

4.1 系统功能评价

4.1.1 顾客端

（一）菜单查询功能

① 优点：清晰的菜单展示能帮助顾客快速找到他们喜欢的菜品。菜单具有价格、销量、厨师、所属食堂以及厨师是否为优秀厨师等的信息，这样的有助于顾客更快地做出选择。搜索功能，可以让顾客直接搜索到想要的菜品。

② 改进建议：如果顾客能看到菜品的图片，配合详细的描述，就能更好地选择。

（二）点菜功能

① 优点：顾客能够直接在系统中选择和确认订单，不需要人工消耗。

② 改进建议：根据顾客的点菜历史和偏好，系统可以推荐他们可能会喜欢的菜品。提供特定的组合套餐，给予折扣，吸引顾客选择。

（三）个人消费情况查询功能

① 优点：顾客可以随时查看自己的消费情况，非常透明、清晰，避免误会。保存顾客的历史消费记录，方便顾客对比和分析。显示每道菜的顾客评价，可以帮助顾客参考过去的就餐体验。

② 改进建议：可以提供顾客在用餐前后对自己的消费预算。在查询消费情况时，如果有可用的优惠券或会员折扣，可以及时提醒顾客。

4.1.2 管理端

（一）新增/删减菜品功能

1、增加菜单

① 优点：

实现简洁明了，提示信息准确清晰。用户只需依次输入菜名、编号、价格、销量、等信息即可完成菜品添加。

② 改进建议：

可以增加对输入的验证，如价格是否合理（正数且在合理范围内）。 可以增加异常处理，提示用户重新输入合法的菜名和价格。

2、删除菜单

① 优点：只需输入菜名（保证菜名正确，其余信息不影响）即可删除对应菜品，使用简单方便。提示信息清晰，操作结果反馈及时。

② 改进建议：使用菜名查找效率较低，尤其是菜单项多时，可以考虑使用菜品编号来提高效率。删除之前可以增加二次确认，避免误操作。记录删除操作日志，便于后续追踪和审计。

（二）修改/查询菜单信息功能

1、修改菜单信息

① 优点：用户可以通过输入菜名、价格、销量等直接修改菜品信息，操作简便。修改成功与否有明确提示，便于用户确认。

② 改进建议： 可以使用菜品 ID 替代菜名查找，提高查找效率和准确性。

可以提供图形化界面，增强用户修改菜品信息的体验感。

2、查询菜单信息

① 优点： 一次性显示所有菜品的信息，信息展示直观明了。直接点击按钮即可查询，简单易用。

② 改进建议：当菜单项多时，可以分页显示避免信息量过大。可以增加按价格范围、类别等条件查询功能，提升查询灵活性。

（三）销量榜单查询功能

① 优点： 清晰展示热门菜品排行榜，帮助用户了解受欢迎的菜肴。销量排序准确，且在每次修改/新增/删除后可以自动重新排序。

② 改进建议：可以增加图表（如柱状图、饼图）展示销量数据，直观且美观。

（四）退出系统

优点：一键退出系统，快捷方便。退出时有友好的确认提示，防止误操作。

4.2 心得体会

我们小组完成了一个具有实际应用价值的菜单管理系统。从最初的系统设计到最终功能的实现，我们每个成员都付出了许多努力，收获了非常多的宝贵经验和知识。以下是我们在这个项目中的一些心得体会：

（一）遇到的技术挑战与解决方案

① 数据存储与管理：在系统开发过程中，我们综合运用了链表来实现对菜单和订单数据的高效管理。这种数据结构不仅提高了系统的性能，还让我们更透彻地理解了数据结构在实际应用中的作用。

② 安全性保证：系统采用了角色登录界面认证，以确保不同用户只能访问其权限范围内的功能。通过设计和实现权限管理机制，我们有效地保障了系统的安全性，防止了未授权的访问和操作。

③ 团队合作与学习收获

④ 分工与协作：在项目开始时，我们进行了详细的分工，每个人都负责不同模块的开发。在项目进行过程中，我们保持了高效的沟通和协作，及时分享解决问题的经验和心得。这种良好的团队协作使得项目得以顺利推进。

⑤ 综合能力提升：通过这个项目，我们不仅巩固了C++方面的知识，还提升了项目管理和团队合作的能力。每个成员在实践中都积累了许多宝贵的经验，这对未来的学习和工作都具有重要意义。

⑥ 反思与未来展望

⑦ 面对挑战：在开发过程中，我们遇到了一些技术难题，例如如何从文件中读取信息、将信息存储在链表中以及等。通过团队的共同努力，这些问题得到了很好的解决。这些挑战不仅磨练了我们的技术能力，也培养了我们解决问题的能力和应变能力。

⑧ 持续改进：虽然项目已经完成，但我们也认识到还有很多可以改进的地方。例如，在菜单功能上可以更加智能和多样化。这些都是我们未来可以进一步提升的方向。

（二）总结

通过小组项目，我们深刻地体会到了理论与实践相结合的重要性。这个菜单管理系统不仅让我们学到了很多专业技术知识，还培养了我们的团队协作精神和项目管理能力。在这个过程中，每个成员都得到了成长和提升。未来，我们将继续努力，运用所学知识，参与更多的项目，不断挑战自我，提升能力。