餐厅点餐系统面向对象设计书

课程名称 软件工程

姓 名

组 别 1组

专业年级

## 1引言

### 1.1编写目的

为使商家降低人工服务成本，提高服务效率，实现智能化餐厅，提高消费者的用餐质量，我开发小组在多方面考察、分析、研究现有餐厅点菜管理系统的基础之上，以提高消费者满意程度及商家服务水平和市场竞争力为目标，致力于开发出一套可视化程度高、功能齐全、集分析管理与一体的餐厅点餐系统，极具有市场价值。

本文主要介绍了“餐厅点餐系统”的整个设计开发过程。该主要系统功能包括以下几个模块:系统管理模块（包括用户管理、权限管理、服务端登录、系统备份、系统还原）、基础数据设置管理、顾客管理模块（包括会员注册与登录、点菜、催单、上菜、统计、会员积分兑换优惠、结账、评价等）、统计管理、后厨管理模块（包括接收订单、提醒服务员上菜）、采购管理等模块。该总体设计说明书体现了清晰的餐厅点餐系统管理流程，完成了餐厅日常管理要求，是一个典型的信息管理系统。该系统大大地简化了操作流程，提高了餐厅的工作效率。

编写这份总体设计说明书主要是为了使小组成员充分认识和掌握本系统的情况，以便确定本系统的后继工作，本说明书主要提交给用户和小组成员参考，以便最终实现用户的要求，给用户一份满意的答卷。

### 1.2背景

随着我国餐饮业的飞速发展，餐厅数量激增，传统的经营模式不仅人工成本高，还效率低下。为了提高餐厅的竞争力，提高服务效率，传统手工点菜方式逐渐被淘汰，各种餐厅点餐系统应运而生，呈现多元化发展。

本系统的名称：餐厅点餐系统

项目与其他软件，系统的关系：本项目采用客户机/服务器原理，客户端的程序是建立在Windows 系统上以PowerBuilder以及MySql数据库等为开发软件的应用程序。

### 1.3定义

|  |  |
| --- | --- |
| 角色 | 需求描述 |
| 顾客 | 根据系统操作提示自主进行点菜、加菜、退菜操作，系统要能够根据顾客的选择进行自动结账计算并显示。 |
| 服务员 | 根据系统操作提示对已经上桌的菜进行标记，查看账单进行收账。 |
| 管理员 | 查看营业情况，各菜品反映情况，顾客反馈意见等，对数据库初始数据的设置，系统维护与升级。对数据进行分析，并根据顾客点菜情况适当修改菜单。权限设置，数据备份。。 |
| 厨师 | 查看点餐情况，处理订单，查看顾客对菜品的评价。 |
| 采购部门 | 根据各菜品的受欢迎度采购食材。 |
| 数据流图 | 数据流图描绘系统Z的逻辑模型，图中没有任何具体的物理元素，只是描绘信息在系统中流动和处理的情况。 |
| 系统的流程图 | 系统流程图是描绘物理系统的传统工具。它的基本思想是用图形符号以黑盒子的形式描绘系统里的每个部件（程序，文件，数据库，表格，人工过程等）。 |

\*注：以上信息将以数据的形式在网络之间传递，如没有特殊要求，将不再产生实体打印单据。

### 1.4参考资料

《实用软件工程》，郑人杰等著，清华大学出版社；

《软件工程》第二版，李代平等著，清华大学出版社；

《软件工程》第六版，Roger S．Pressman著，机械工业出版社；

《软件工程课程实验指导书》，山东交通学院计算机科学与工程系；

《数据库系统概论》（第四版）,萨师煊 著, 高等教育出版社；

《软件工程导论》（第五版）,张海藩 著，清华大学出版社。

## 2总体设计

### 2.1需求规定

本系统在顾客点菜时主要输入顾客的会员号（非会员可注册或直接点餐），餐桌号信息，然后将电子账单反馈给顾客，顾客确认无误后在线支付或在前台支付领取纸质账单。

本系统在顾客点餐时查询菜单主要输入查询的相关信息，包括菜品编号、菜名、菜品种类等信息。

本文主要介绍了“餐厅点餐系统”的整个设计开发过程。该主要系统功能包括以下几个部分:

（1）系统管理模块：主要包括用户管理、权限管理、服务端登录、系统维护、系统还原、系统备份；

（2）基础数据设置管理模块：菜品的种类、价格、口味、主要材料、图片设置等；

（3）顾客管理模块：包括会员登录、点菜、催单、上菜、统计、会员积分兑换优惠、结账、评价等；

（4）统计管理模块：点菜次数、账单统计、选菜总价统计；

（5）采购管理模块：缺料统计、处理到料单、菜品统计；

（6）帮助模块：对用户在使用系统是遇到的问题提供相应的帮助。

（7）后厨管理模块：包括接收订单、提醒服务员上菜；

（8）餐桌管理模块**：**包括排队，选座，餐桌状态等

（9）采购管理模块**：**包括库存，缺料，处理到料单等

本系统对性能的规定：

（1）系统的更新处理时间应该在可接受的范围内；

（2）系统的数据查询时间应该在可接受的范围内；

（3）系统的数据统计时间应该在可接受的范围内。

（4）操作方式上应该能够满足鼠标和键盘任意切换的需要；

（5）能够支持Windows95、Windows98、Windows2000、Windows Me、Windows XP运行环境。

（6）留有与其他系统的接口。

### 2.2运行环境

（1）硬件环境：运行本软件要求处理器在奔腾Ⅲ以上，内存在256MB以上的计算机。

（2）软件环境：本系统支持的操作系统包括：Microsoft Windows 2006及以上系统

本系统支持的数据库为Microsoft Office access；

本软件的开发工具为PowerBuilder程序设计语言。

### 2.3基本设计概念和处理流程

本系统的基本设计概念和处理流程如下：

（1）系统启动时，激活身份验证模块。客户端包括游客和会员身份，服务端包括管理者和服务员身份。

（2）通过身份验证后，对系统进行初始化，判断用户身份，导入到不同的处理页面。

（3）系统进入消息循环状态，通过事件驱动机制激活各功能模块，并执行指定模块。

（4）各项功能执行完毕后，重新进入消息循环状态。

（5）由退出系统事件激活关闭系统模块，退出系统。

### 2.4系统结构图

完整的系统结构图：

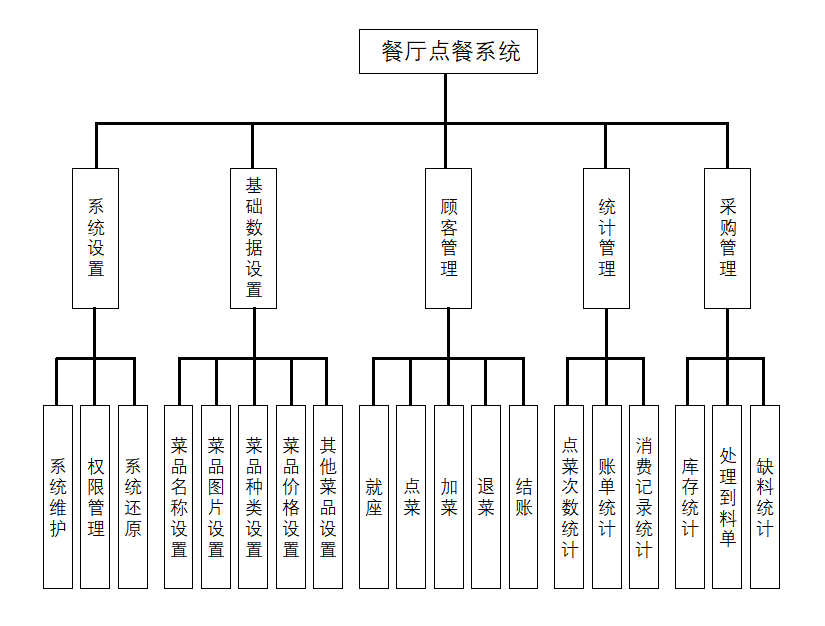


图2.4.1 系统结构图

### 2.5功能需求与程序的关系

（1）用户身份验证必须调用“验证用户”模块程序进行身份验证；

（2）顾客点菜时必须调用“采购管理”中的“缺料记录”信息；

（3）服务员在传菜时需要调用“顾客管理”中的“点菜”信息；

（4）顾客结账时进行查询要调用“顾客管理”中的顾客“点菜”及“上菜”信息。

### 2.6人工处理过程

（1）新增菜品时需要对相应的菜谱信息进行录入。

（2）对于已经过时不用的菜品需要管理人员对菜单数据进行相应的删除。

（3）其他方面数据的处理系统将会自动进行。

（4）用户可以修改会员。

（5）如果当天营业前菜品存量较昨日发生变化，对菜品存量进行修改。

## 3设计问题域子系统

### 3.1调整需求

为方便用户操作与管理，系统分为客户端和服务端（包括前台服务端、后厨服务端），用户包括顾客、管理员、前台服务员、后厨管理员。客户端用户为顾客；前台服务端用户为管理员、前台服务员；后厨服务端为后厨管理员。需求分析中的经理与管理员功能进行合并。

### 3.2设计类

根据需求的调整，本系统的类主要包括顾客、管理员、后厨管理员、服务员、采购员、菜品、菜单、账单、营业额、订单。

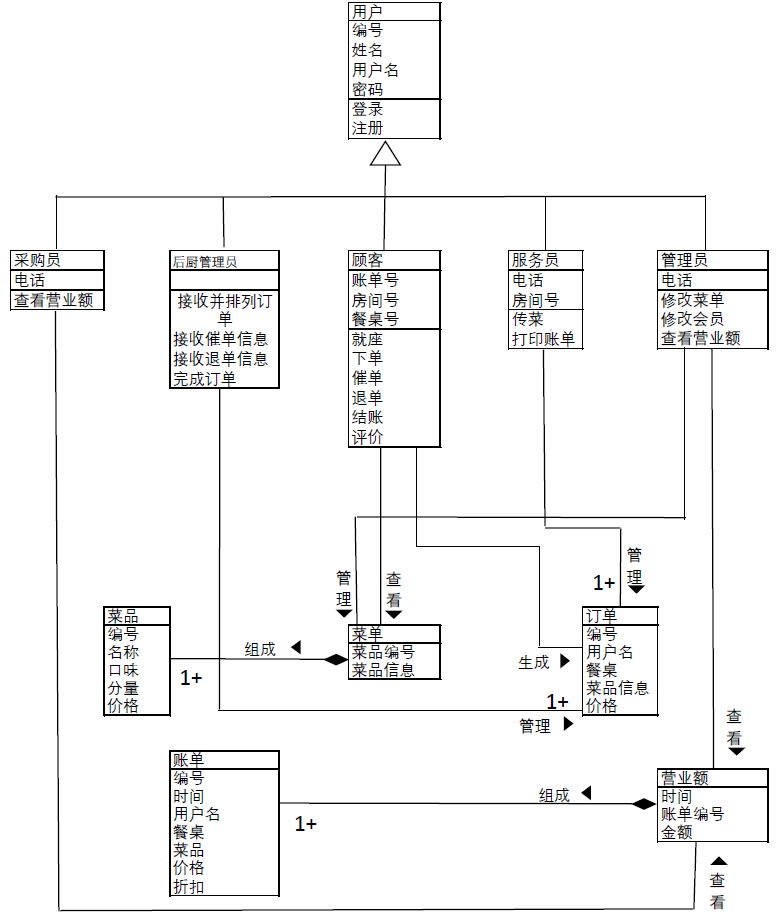


图3.2.1 类图

注:菜品、菜单、账单、订单、营业额均为实用类，只封装类的属性。

### 3.3建立问题域模型

图3.3.1问题域模型

前台计算机

顾客

后厨计算机

餐桌触屏机

服务员

后厨管理员

管理员

数据库

订单

菜单

营业额

账单

用户

事务

事务

事务

电话线

电话线

顾客

界面

采购员

菜单

订单

订单

## 4设计人机交互子系统

### 4.1分类用户

本系统的用户按职能分为:顾客、前台服务员、厨师、管理员

### 4.2描述用户

**1、顾客：**

目的：使用系统点餐、催单、退餐、结账、评价，会员在点餐前登录账户进行积分，非会员可注册成为新会员。

特点：年龄不限，教育水平不限。

成功的关键因素：整体操作界面要简洁易懂、易于操作；结账信息要准确无误；催单退餐要迅速并且反馈及时；菜品信息及时更新。

熟练程度：不限。

表4.2.1 任务脚本1

|  |
| --- |
| 主脚本： |
| ·顾客信息及时链接到数据库  ·点餐信息传输到前台，经前台传达到后厨，再由前台反馈给顾客下单成功的信号  ·催单时顾客的催单信息直接传至后厨  ·结账时，顾客使用的客户端调用前台的账单信息，由顾客确认后反馈给前台，顾客在线支付或现金支付后，客户端反馈给前台服务端打印账单  ·顾客使用客户端进行评价，评价信息储存在服务端数据库中  ·客户端中的菜品信息或属性要随时更新 |

**2、前台服务员**

目的：管理顾客的用餐情况，协调顾客与后厨，打印账单。

特点：年龄：18岁以上；教育水平：初中以上；

成功的关键因素：数据库更新迅速；操作指令简洁易懂；各项操作指令不能冲突，需要人工确认的项目不能出错；前台服务端系统响应迅速。

熟练程度：专家。

表4.2.2 任务脚本2

|  |
| --- |
| 主脚本： |
| ·前台服务端及时接收客户端的点餐信息，并即时传达到后厨的服务端分支；  ·在顾客进行催单操作时，前台即时传达指令至后厨；后厨制作菜品完毕后，前台接收到信息后指派服务员上菜；  ·在顾客结账调动账单时前台收到提醒。  ·前台可根据顾客意愿选择是否打印纸质账单和发票。 |

1. **后厨管理员**

目的：接收订单、制作菜品、确认菜品完成。

特点：年龄不限；教育水平不限

成功的关键因素：后厨服务端应即时传达和接收菜品信息，厨师制作完成菜品应即时确认完成；按时间优先规则对菜品进行排序；。

熟练程度：不限

表4.2.3 任务脚本3

|  |
| --- |
| 主脚本： |
| ·后厨客户端接收到订单后发出提示音提醒后厨管理员，并反馈给顾客订单已接收的信号。  ·后厨客户端接收到催单信号时发出提示音。  ·后厨客户端接收到退餐信息时应及时更新餐品制作列表。  ·后厨管理员完成菜品后点击“已完成”按钮，后厨客户端将信息反馈至前台客户端，提醒服务员上菜。 |

1. **管理员**

目的：管理餐厅所有数据库、查看营业情况、修改菜单、修改菜品库存信息、修改会员信息，管理采购系统。

特点：年龄不限；教育水平：高中以上；

成功的关键因素：数据库存储数据和系统统计数据功能不能出错。

熟练程度：专家

表4.2.4 任务脚本4

|  |
| --- |
| 主脚本： |
| ·管理员登录客户端，账号密码正确后进入后台查询今日营业额。  ·管理员修改菜单明细。  ·管理员修改会员信息。  ·管理员不能修改账单数据库。  ·管理员可以管理菜品库存信息。 |

### 4.3设计命令层

**1、准则**

界面务必简洁明了；客户便于找到所需信息；反馈要迅速；重要信息要放在

显眼位置；界面元素要排列有序；不同元素要便于区分；信息要做到多样化。

**2、建立命令层**

（1）客户端

客户端登录界面包括：会员登录按钮、用户名窗口、密码窗口、图标、登录和注册按钮、游客进入按钮。

客户端点餐界面包括：菜单、“我的订单”按钮、“下单”按钮、“结账”按钮。

菜品属性界面包括：菜品图片、口味（加辣、微辣、不要辣）按钮、备注框、份数（大、中、小）按钮、确定选择按钮。

我的订单界面包括；已点菜品名称、份数、口味、价格，催单按钮、退餐按钮。

结账界面包括：已点餐品列表及总价、出错反馈按钮、确认支付按钮。三种结账方式：

1.会员在余额中扣除。按钮：余额扣除（会员）提示信息：查看我的余额

2.二维码扫描支付。图片：支付宝收款码和微信收款码。

3.前台结账。按钮：前台结账。

用餐完成界面：打印纸质账单和发票按钮，评价按钮，退出系统按钮。

（2）前台服务端

登录界面包括：管理员通道：用户名、密码、管理员登录按钮；服务员通道；

管理员操作界面：总营业额查询、今日营业额查询按钮，查看菜单按钮、修改菜单按钮、会员信息管理按钮。

服务员操作界面：各桌桌号按钮；每桌详情界面按钮：查看菜单；打印纸质账单；

（3）后厨服务端

订单列表界面：每份订单每个菜品后有“完成”按钮。

**3、细化命令层**

对各个命令层进行排序并对命令层中的操作分块。

（1）客户端

**Step1：**

顾客在登录界面选择不同通道进入点餐界面。

**Step2：**

点餐界面中菜单、订单、结账在同一命令层，

顾客点击菜单任一菜品可查看菜品属性。

顾客点击订单按钮可进入订单界面查看已点订单并进行催单、退餐操作。

顾客点击结账按钮可进入结账界面查看账单信息，出现错误时可点击出错反馈按钮向前台反馈信息，确认无误后点击支付按钮进行支付操作。

**Step3:**

顾客进入餐单属性界面、订单界面均可退回点餐界面。若确认支付后则不可退回。

**Step4：**

顾客进行结账操作后进入用餐完成界面，选择进行是否打印账单和发票，并对此次用餐进行评价，离桌前点击退出系统按钮，退出系统按钮也会在结账后60秒内自动生效。

顾客点餐界面可退回登录界面。

（2）前台服务端

**Step1：**

使用者在登录界面选择进入不同界面，进入服务员界面直接点击服务员登录按钮；进入管理员界面需要输入用户名与密码并点击登录按钮。

**Step2：**

进入服务员操作界面后显示餐厅桌位情况，点击已经就餐的座位可以进入桌位详情信息。

进入管理员操作界面可选择营业日期查看阶段营业额，界面自动显示今日营业额和菜单信息，点击会员信息按钮进入会员信息修改界面。

**Step3：**

服务员进入每桌详情界面后可查看每桌用餐详情，每桌详情包括桌号，已点菜品，菜品状态（口味、分量、价格，是否开始烹饪，是否已经上桌，等待时间是否超时），是否会员，会员折扣，是否需要手动打印账单和发票。

管理员在会员信息界面可对会员信息进行修改，点击确认按钮保存操作。

**Step4：**

服务员操作界面可退回登录界面。

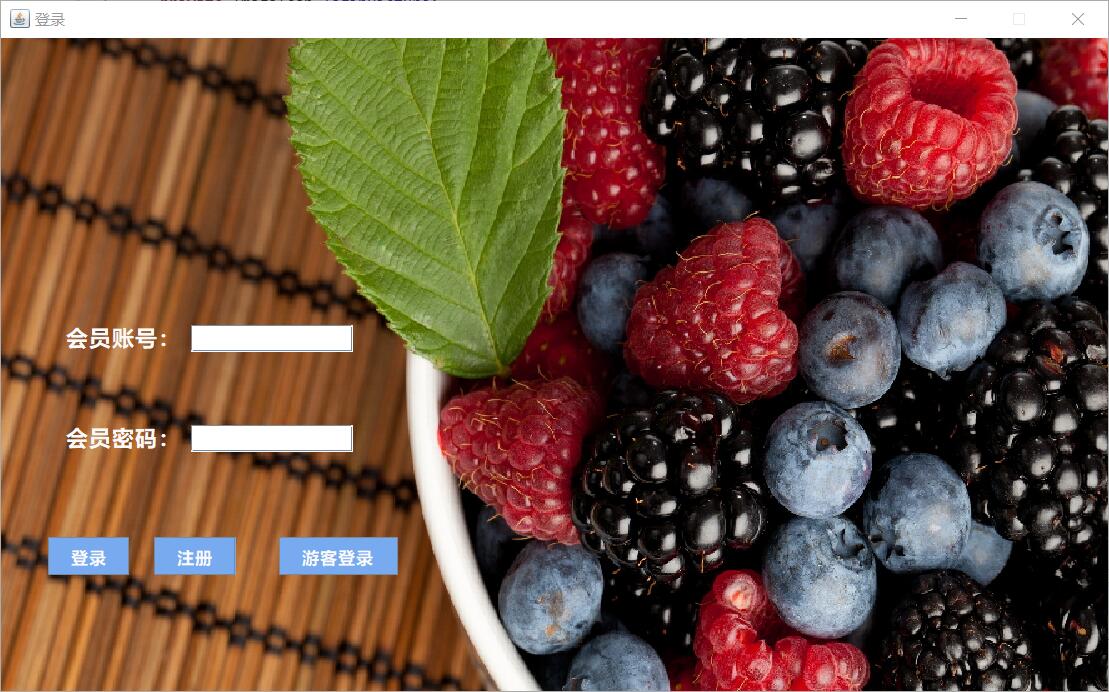
管理员操作界面可退回登录界面。

（3）后厨客户端

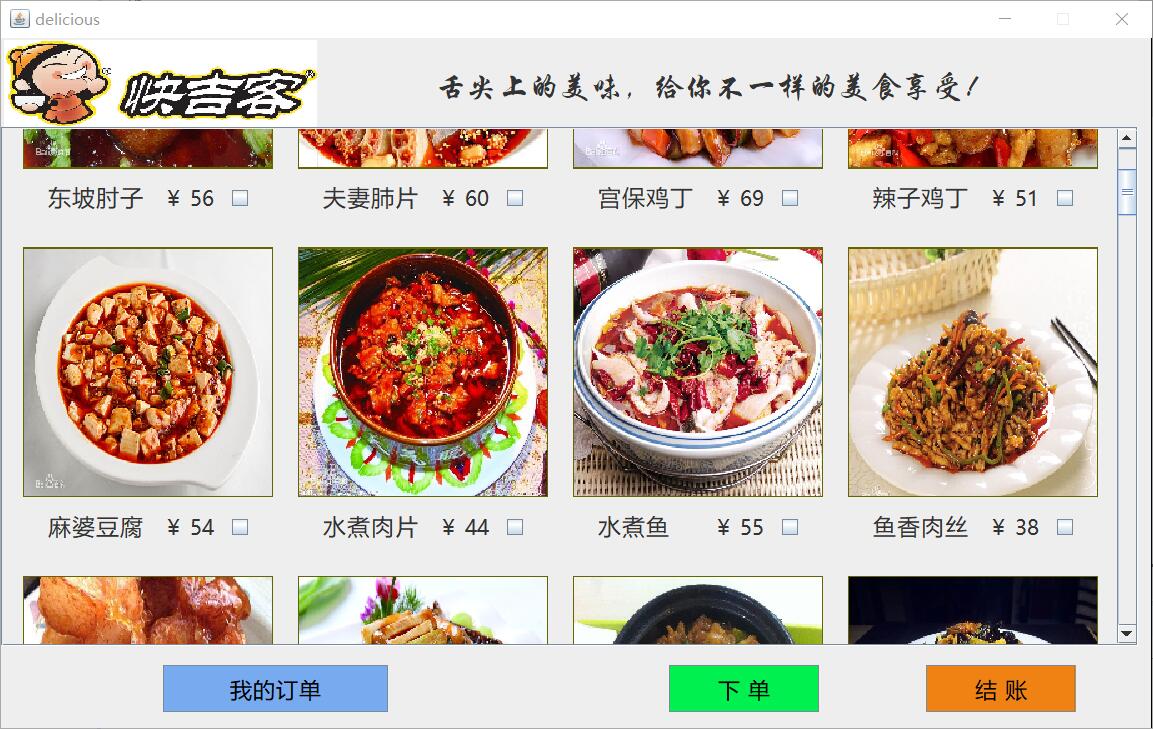
后厨订单列表界面命令层只有一层，命令包括：完成菜品，命令直接传输给前台，由前台传输给客户端。

### 4.4设计用户界面

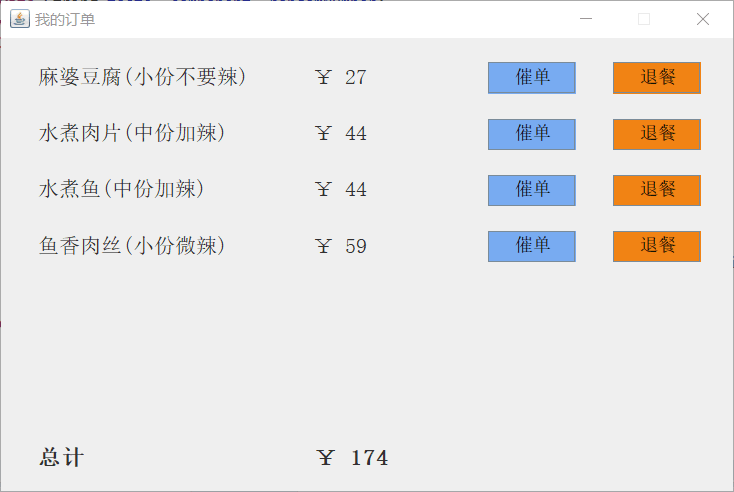
**（1）客户端登录界面**

****

**（2）客户端点餐界面**

****

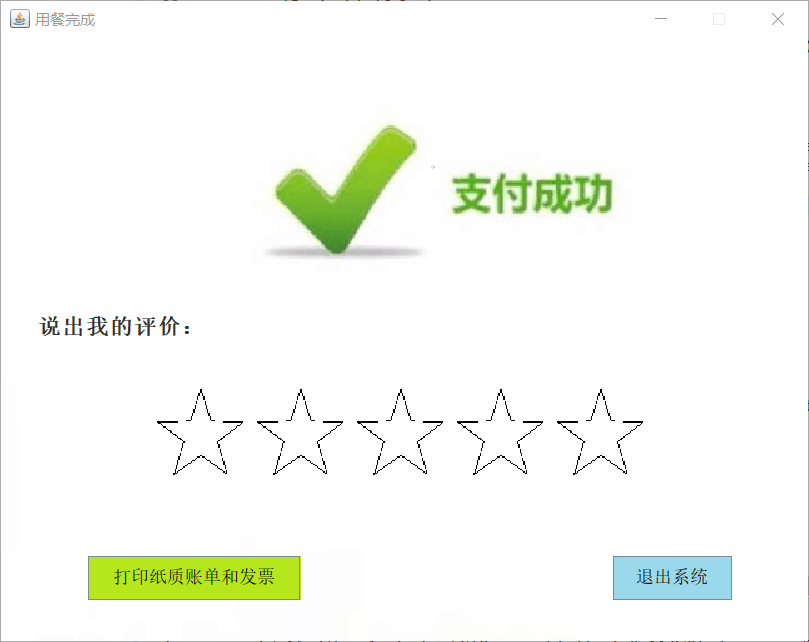
**（3）客户端我的订单界面**

****

**（4）客户端结账界面**

****

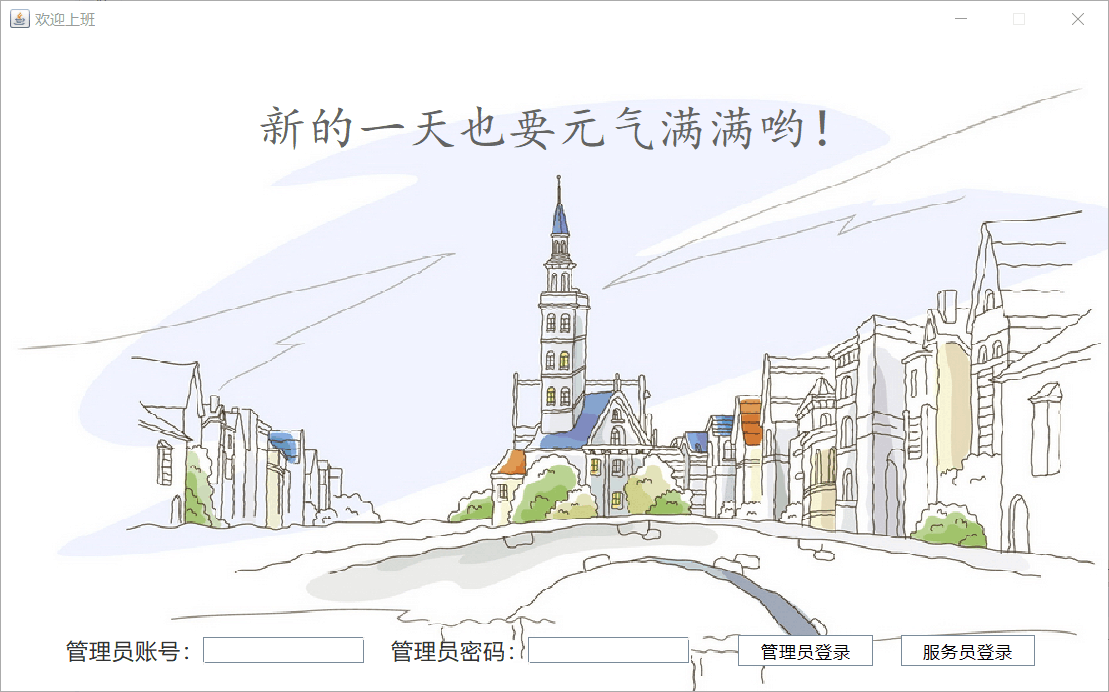
**（5）客户端用餐完成（未评价）界面**

****

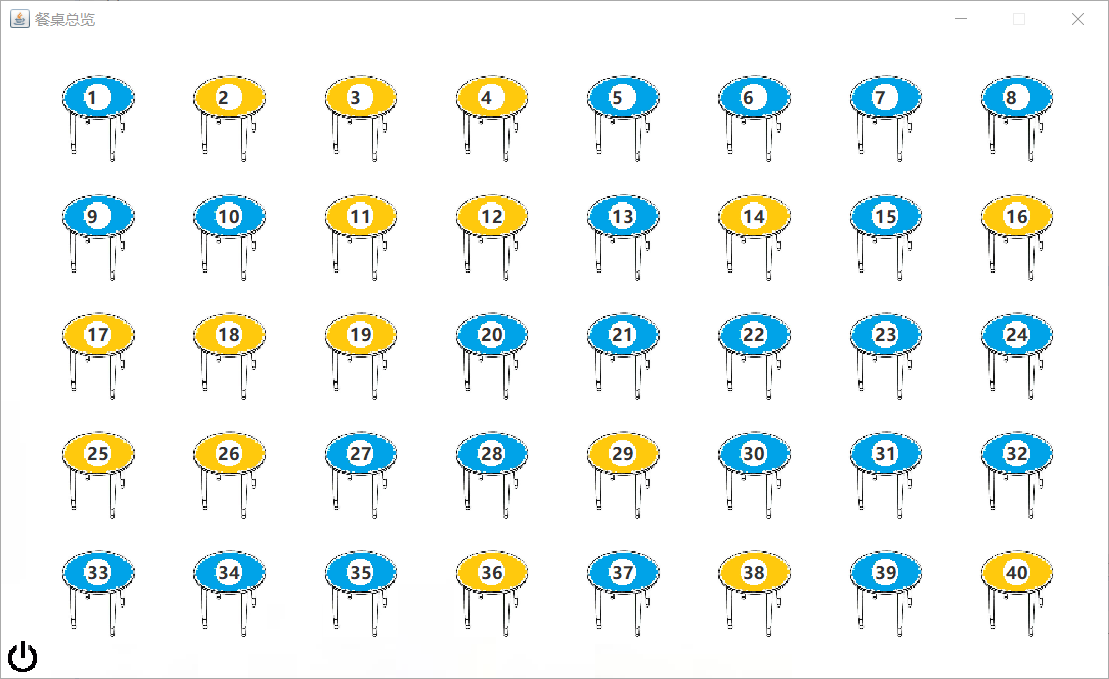
**（6）客户端用餐完成（评价成功）界面**

****

**（7）前台服务端登录界面**

****

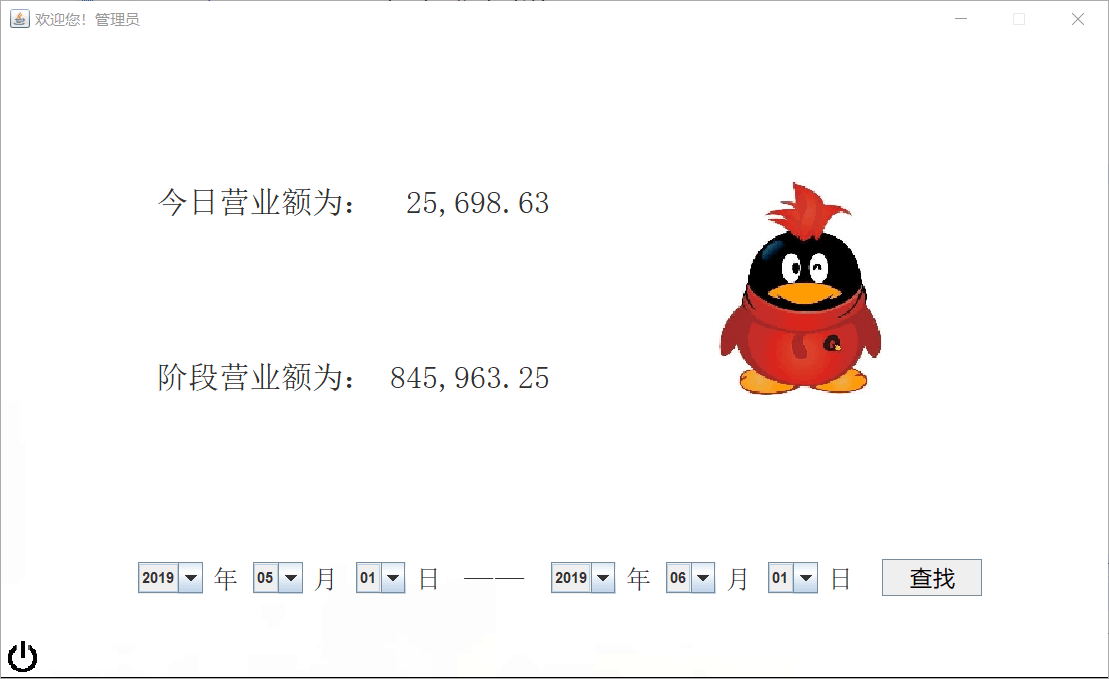
1. **前台服务端餐桌总体界面**



**（9）前台服务端餐桌详情界面**



**（10）前台服务端管理员操作界面**



## 5设计任务管理子系统

该系统为餐厅点餐系统，系统分为三部分：顾客客户端、前台服务端、后厨服务端。

每部分服务器端口的主要服务为：

1. 顾客客户端：顾客登录就座、下单、催单、退单、结账、评价。
2. 前台服务端：接收选座信息、接收顾客登录信息、传输订单至后厨、向客户端反馈账单、打印纸质账单、传输催单信息至后厨、上菜。
3. 后厨服务端：接收订单并排列订单、接收催单信息、接收退单信息、向前台传输菜品完成信息。

因为已经将并行任务分解为各个端口的任务，所以不考虑任务的优先级

### 5.1确定事件驱动型任务

（1）前台服务端中传输催单信息任务为事件驱动型任务，当前台服务端接收到来自客户端的催单信号时，该任务被唤醒，向后厨客户端发出催单信息，并且向客户反馈信息。

（2）前台服务端中向客户端反馈账单任务为事件驱动任务，当前台服务端接收来自客户端的结账信息时，该任务被唤醒，向客户端传输账单。

（3）前台服务端中打印纸质账单任务为事件驱动型任务，当客户端发出打印账单信号，该任务被唤醒，前台服务端打印账单。

（4）前台服务端中上菜任务为事件驱动型任务，当后厨服务端传达菜品完成信息，该任务被唤醒，前台发出提示音提醒服务员上菜。

### 5.2确定时钟驱动型任务

顾客客户端中退单任务为时钟驱动型任务，

### 5.3确定关键任务

顾客下单、反馈账单信息、打印账单、支付、接收和排列账单、上菜

硬件环境：运行本软件要求处理器在奔腾Ⅲ以上，内存在256MB以上的计算机。

### 5.4定义各个任务

（1）客户端任务

|  |
| --- |
| **任务名：**登录就座 |
| **描述：**事件驱动型任务，当顾客选好座位并且填写登录信息（或选择游客进入），该任务执行 |
| **操作**：该任务是将顾客信息与存入会员数据库，当顾客登录时与数据库中信息对比，判断顾客是否能通过会员通道进入点餐系统，并且在登录时向前台传达选座信息。 |
| **通信：**该任务传达前台服务端与客户端之间的通信。 |

|  |
| --- |
| **任务名：**下单 |
| **描述：**事件驱动型任务，当顾客点餐完毕点击下单按钮后该任务执行。 |
| **操作**：该任务将顾客点菜单导入临时数据库，并且传递给前台该座号的订单。 |
| **通信：**该任务完成客户端向服务端的通信 |

|  |
| --- |
| **任务名：**催单 |
| **描述：**事件驱动型任务，顾客下订单完成后，在等待期间可以点击催单按钮催单。 |
| **操作**：该任务将顾客的催单信号传递给前台，然后接受前台服务端反馈的已催单信号。 |
| **通信：**完成客户端与服务端的通信 |

|  |
| --- |
| **任务名：**退单 |
| **描述：**时间驱动型任务，顾客下订单满30分钟后未上菜，顾客可选择退单 |
| **操作**：该任务在下单时间满30分钟后才能执行，当顾客点击退单按钮，该任务执行，传递退单信息至前台，并接受前台的反馈信息 |
| **通信：**完成客户端和服务端之间的通信 |

|  |
| --- |
| **任务名：**结账 |
| **描述：**事件驱动型任务，顾客下单用餐完毕后点击结账确认订单支付该任务执行。 |
| **操作**：顾客点击结账，该任务调用消费记录数据库中顾客的消费账单，顾客确认支付后该任务根据金额生成二维码供顾客支付（或顾客点击现金支付，该任务传递信息至服务台），顾客支付完毕后，该任务使客户端系统自动跳转评价界面。 |
| **通信：**完成客户端和服务端的通信，调用服务端的数据，传向客户端，并且反馈客户端的信息至服务端 |

|  |
| --- |
| **任务名：**评价 |
| **描述：**事件驱动型任务，顾客支付完成后系统自动跳转评价界面，该任务执行 |
| **操作**：顾客选择和填写评价后，该任务将顾客的评价导入评价数据库中，并且反馈评价成功字样。 |
| **通信：**该任务将客户端的信息传递到服务端数据库，并且反馈信息至客户端。 |

（2）前台服务端任务

|  |
| --- |
| **任务名：**接收选座信息 |
| **描述：**事件驱动型任务，客户端的登录和就座任务执行后，该任务执行。 |
| **操作**：前台服务端接收到顾客的就座信息，该任务执行使前台对应座位号显示顾客就座和登录信息，并且发出提示音。 |
| **通信：**该任务接收客户端的信息并显示在服务端。 |

|  |
| --- |
| **任务名：**接收顾客登录信息 |
| **描述：**事件驱动型任务，客户端的登录和就座任务执行后，该任务执行。 |
| **操作**：前台服务端接收到顾客登录信息，将信息存储在临时数据库中，并显示在餐桌信息界面。 |
| **通信：**该任务接收客户端的信息并存储在数据库，显示在服务端。 |

|  |
| --- |
| **任务名：**传输订单至后厨 |
| **描述：**事件驱动型任务，客户端的下单任务执行后，该任务立即执行。 |
| **操作**：前台接到顾客的订单后，将订单显示在餐桌信息界面，并且传输给后厨客户端。 |
| **通信：**传递请客户端信息至后厨服务端。 |

|  |
| --- |
| **任务名：**向客户端反馈账单 |
| **描述：**事件驱动型任务，客户端的结账任务执行时，该任务执行。 |
| **操作**：顾客点击结账后，前台收到结账信息，该任务调用临时数据库中的账单并且计算总价反馈给客户端。 |
| **通信：**从前台服务端传递数据至客户端。 |

|  |
| --- |
| **任务名：**打印纸质账单 |
| **描述：**事件驱动型任务，顾客在客户端选择打印账单后执行。 |
| **操作**：顾客在客户端选择打印账单，前台接到信息后打印账单。 |
| **通信：**从客户端传递信息至前台。 |

|  |
| --- |
| **任务名：**传输催单信息至后厨 |
| **描述：**事件驱动型任务，前台客户端接收到顾客的催单信号后，该任务立即执行。 |
| **操作**：前台客户端接收到顾客的催单信息后立即传输催单信息至后厨客户端。 |
| **通信：**从前台客户端传递信息至后厨客户端。 |

|  |
| --- |
| **任务名：**上菜 |
| **描述：**事件驱动型任务，接收到后厨传递的订单完成信号后任务执行。 |
| **操作**：前台客户端接收到后厨的订单完成信号后，该任务执行显示对应订单需要上菜，发出提醒。 |
| **通信：**接收后厨服务端的信号，在前台服务端发出提示。 |

|  |
| --- |
| **任务名：**接收订单并排列订单 |
| **描述：**事件驱动型任务，后厨接收到订单后该任务立即执行。 |
| * **操作**：后厨接收到订单后，将订单按照时间先后排列显示在后厨显示屏上。 |
| * **通信：**该任务接收前台客户端的数据。 |

|  |
| --- |
| **任务名：**接收催单信息 |
| **描述：**事件驱动型任务，前台传递催单信息后该任务执行 |
| **操作**：后厨接收到催单信息后发出提示音。 |
| **通信：**该任务接收前台客户端的信息，并且转化信息在后厨服务端显示出来。 |

|  |
| --- |
| **任务名：**接收退单信息 |
| **描述：**事件驱动型任务，前台传递退单信息后该任务执行。 |
| **操作**：前台传递退单信息后，后厨接收退单信息，将该订单从订单列表删除。 |
| **通信：**该任务接收前台客户端传向后厨客户端的信息，并且管理后厨客户端的数据。 |

|  |
| --- |
| **任务名：**向前台传输菜品完成信息 |
| **描述：**事件驱动型任务，后厨管理员点击菜品完成，该任务执行。 |
| **操作**：后厨管理员确认菜品完成后，该任务向前台传递上菜信息。 |
| **通信：**该任务传输后厨服务端的信息至前台服务端。 |

## 6设计数据管理子系统

### 6.1数据库设计

**菜品数据库**：存放餐厅所有菜品的编号、名称、类型、单价等；

**餐桌数据库**：存放餐厅所有房间以及房间内餐桌的状态等；

**临时数据库**: 存放当前顾客以及所点菜品的详细记录;

**消费记录数据库**:存放顾客的消费信息；

**用户数据库**：存放会员消费信息记录。

### 6.2数据库逻辑设计

**用户信息表：**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 是否可为空 |
| 用户名 | Y\_name | string(18) | N |
| 密码 | Y\_ID | string(18) | N |

**菜品信息表：**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 是否可为空 |
| 名称 | c\_name | char(12) | N |
| 价格 | c\_price | string | N |
| 口味 | taste | char | N |
| 份量 | weight | char | N |
| 图片路径 | image | string | N |
| 数量 | number | string | N |

**餐桌信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 是否可为空 |
| 桌号 | table | string | N |
| 总价 | pay | string | Y |
| 评价 | evaluate | char | Y |

**临时信息表（订单）**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 是否可为空 |
| 桌号 | table | string | N |
| 所点菜品 | Dc\_name | char | N |
| 订单总价 | D\_pay | string | N |

**消费记录信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 是否可为空 |
| 会员名 | H\_name | string | Y |
| 桌号 | table | string | N |
| 账单菜品 | Z\_name | char | N |
| 账单总价 | Z\_pay | string | N |
| 日期 | day | char | N |
| 查询 | query | varchar | Y |

## 7设计类中的服务

综合考虑需求分析中的对象模型、动态模型、功能模型，确定该系统中的类的服务有：

工作人员登录管理系统：工作人员输入用户名和密码登录管理系统，进行后台操作。

选定座位：顾客选定空闲座位入座，同时将座位状态从空闲改成忙碌。

点菜：顾客读取菜品信息，将选定的菜品写入订单信息。订单信息从顾客终端被传递到前台服务器。

处理菜品：前台服务器接收到顾客发出的订单信息后，继续将订单信息传递到后厨端。后厨管理员接收到订单信息，开始烹饪菜品，同时将菜品状态更改为“烹饪中”。烹饪完成后将菜品状态更改为“已完成”，同时通知前台服务器，管理员接收到完成信息后派遣服务员进行配送。

结账：顾客选择结账后，计算消费金额，生成账单信息。支付成功后，将账单导入营业额，将该座位信息从忙碌改成空闲。

### 7.1设计实现服务的方法

**7.1.1设计实现服务的算法**

上文的类图中用户类包括顾客、服务员、后厨管理员、管理员、采购员，在设计实现服务的算法时，我们以流程图来实现各个用户类中的服务。

（1）用户的登录、注册。

开始

输入账号密码

判断是否存在账号

Y

注册

判断密码是否正确

进入系统

N

N

结束

N

图7.1.1.1 用户登录流程图

开始

输入账号

判断是否已经存在

输入密码

再次输入密码

两次输入是否一致

注册成功

显示“账号已存在！”

报错

Y

N

Y

N

结束

图7.1.1.2 用户注册流程图

（2）顾客就座、下单、催单、退单、结账、评价。

开始

显示空闲位置

顾客选择空闲座位

将该空闲座位记为忙碌

结束

开始

显示菜单

顾客选择菜品口味分量

提交订单

订单传递到前台

结束

是否追加订单

N

Y

图7.1.1.3 顾客就座流程图 图7.1.1.4 顾客下单流程图

开始

顾客催单

传催单信息至后厨

显示“已接受催单信息”

后厨反馈信息

结束

开始

显示菜品状态

退单成功

结束

订单是否完成？

退单失败

N

Y

客户退单

下单是否超过30分钟

Y

N

图7.1.1.5 顾客催单流程图 图7.1.1.6 顾客退单流程图

开始

显示订单

选择支付方式

支付成功，打印账单

生成账单

结束

是否无误

Y

反馈错误

N

修正错误

开始

显示“支付成功”

是否评价

提交评价

退出系统

结束

Y

N

图7.1.1.7 顾客结账流程图 图7.1.1.8 顾客评价流程图

（3）管理员修改菜单、修改会员信息、查看营业额。

注:采购员查看营业额的服务流程与管理员相同。

开始

显示当前菜单信息

结束

验证通过？

更新菜单数据库

Y

N

更改菜单信息

开始

查询会员信息

修改会员信息

结束

信息是否存在？

显示“该会员不存在”

Y

N

是否继续查询

Y

N

图7.1.1.9 管理员修改菜单流程图 图7.1.1.10 管理员修改会员信息流程图

开始

输入日期

是否存在该天营业额

显示“无营业额”

显示营业额详情额

N

Y

是否继续查询

Y

终止

N

图7.1.1.11管理员（采购员）查询营业额流程图

（4）后厨管理员接受并排列订单、接收催单信息、接收退单信息、完成订单。

开始

接受订单

减少相应菜品库存数量

将订单排入列表

将订单标记为未完成

完成

开始

接受催单信息

响铃提醒

反馈“已催单”

完成

图7.1.1.12 后厨接收并排列订单流程图 图7.1.1.13 接收催单信息流程图

开始

接收退单信息

是否正在制作

反馈“正在制作”

删除订单

增加相应菜品库存

反馈“退单成功”

结束

Y

退单失败

N

开始

订单完成

通知前台上菜

将该订单标记为已完成

结束

图7.1.1.14 后厨接收退单信息流程图 图7.1.1.15 后厨完成订单流程图

1. 服务员传菜、打印账单。

开始

接收订单完成信息

传菜

订单状态改为“已上菜”

结束

结束

开始

接收打印账单信息

导出账单

顾客确认是否无误

接收支付信息

打印账单

反馈错误

N

Y

图7.1.1.16 服务员传菜流程图 图7.1.1.17 服务员打印账单流程图

**7.1.2选择数据结构**

数据结构

用户：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

String name[MAXSIZE];

String userName[MAXSIZE];

String password[MAXSIZE];

int length=MAXSIZE;

} User;

采购员：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

String name[MAXSIZE];

String userName[MAXSIZE];

String password[MAXSIZE];

int telephone[MAXSIZE];

int length=MAXSIZE;

} Buyer;

后厨管理员：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

String name[MAXSIZE];

String userName[MAXSIZE];

String password[MAXSIZE];

int length=MAXSIZE;

} KitchenAdministrator;

顾客：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

String name[MAXSIZE];

String userName[MAXSIZE];

String password[MAXSIZE];

double accountNumber[MAXSIZE];

int roomNumber[MAXSIZE];

int tableNumber[MAXSIZE];

int length=MAXSIZE;

} Customer;

服务员：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

Stringname[MAXSIZE];

String userName[MAXSIZE];

String password[MAXSIZE];

int telephone[MAXSIZE];

int roomNumber[MAXSIZE];

int length=MAXSIZE;

} Waiter;

管理员：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

Stringname[MAXSIZE];

String userName[MAXSIZE];

String password[MAXSIZE];

int telephone[MAXSIZE];

int length=MAXSIZE;

} Administrator;

菜品：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

String name[MAXSIZE];

String taste[MAXSIZE];

int tstock[MAXSIZE];

float price[MAXSIZE];

int length=MAXSIZE;

} Dish;

订单：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

int userName[MAXSIZE];

int tableNumber[MAXSIZE];

String dishInformation [MAXSIZE];

float price[MAXSIZE];

int length=MAXSIZE;

} order;

账单：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

String date[MAXSIZE];

int userNumber[MAXSIZE];

int tableNumber[MAXSIZE];

Dish dish[MAXSIZE];

float price[MAXSIZE];

float discount[MAXSIZE];

int length=MAXSIZE;

} bill;

菜单：

const int MAXSIZE;

typedef struct{

int number[MAXSIZE];

String dishInformation [MAXSIZE];

int length=MAXSIZE;

} menu;

营业额：

const int MAXSIZE;

typedef struct{

String date[MAXSIZE];

int acountNumber[MAXSIZE];

float sum[MAXSIZE];

int length=MAXSIZE;

} turnover;

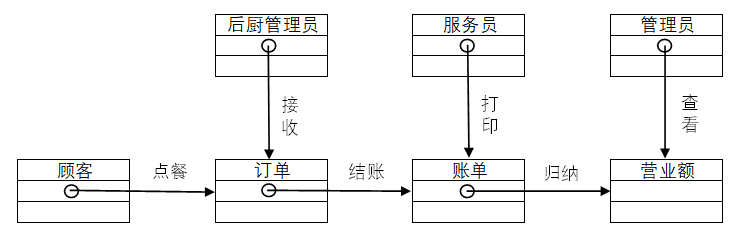
**7.1.3定义内部类和内部操作**

退单、加单：顾客在订单中选定的菜品由于某种原因在菜品烹饪前选择退餐，或者客户在结账前选择增加一些菜品，由此产生的临时信息存放到该类中，类中定义有订单编号，菜品编号、名称、口味、分量。操作有：传递该信息至前台服务端，通知后厨端，同时服务端修改客户订单信息。

时钟：客户就坐之后的一系列行为所花费的时间集合，类中定义有：就座时间、结账时间、菜品等待时间、客户就餐时间。操作有：及时刷新菜品等待时间，当超过预定数额后提醒前台服务器，给与客户一定补偿，结账时将其他定义导入消费记录，留待数据统计。

## 8设计关联

### 8.1实现单向关联



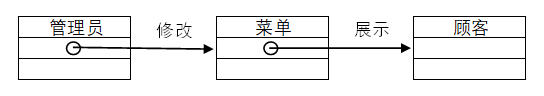


图8.1.1 单向关联图

### 8.2实现双向关联

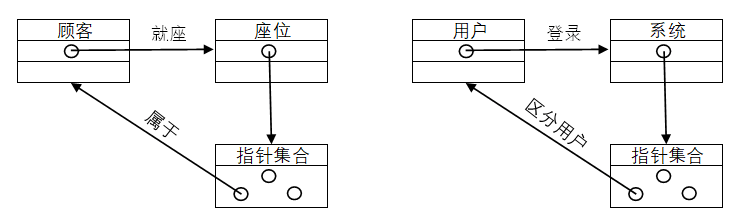
****

图8.2.1 双向关联图

### 8.3链属性的实现

顾客——座位：占有

顾客——订单：菜品

顾客——账单：菜品，总价，支付方式

订单——菜单：部分菜品

## 9设计优化

### 9.1确立优先级

账单正确性>点餐响应速度>订单信息传输速度>数据库存取速度

### 9.2提高效率的方法

**9.2.1建立菜品索引**

为客户提供详细菜单的同时建立查找方法，包括模糊查找和索引查找。通过菜系、口味建立二级索引，提高客户点餐效率。

**9.2.2减少数据库访问次数**

将顾客点餐信息存入临时信息类中，最终结账时统一存入数据库，避免中途多次调用数据库增加运行时间。

**9.2.3多次检验正确性**

在顾客点餐、后厨烹饪、菜品配送、结账时多次计算账单，确保账单正确性。

### 9.3调整继承关系

对餐厅系统的使用者类进行一定的分类继承：

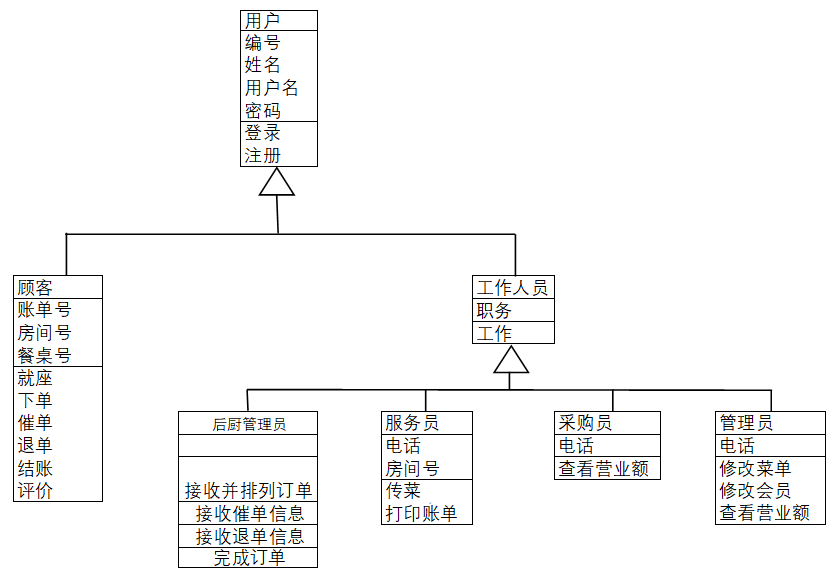


图9.3.1 类的继承关系