长 沙 学 院

软件工程基础实训Ⅱ 实训

|  |  |
| --- | --- |
| **题目** | **基于C/S架构的餐饮管理系统** |
| **学院** | **计算机科学与工程** |
| **专业(班级)** | **软件工程（21软件01）** |
| **姓名** | **任昊轩** |
| **学号** | **B20210304119** |
| **指导教师** | **潘怡、杨刚** |
| **起止日期** | **2023.06.05～2022.06.16** |

实训任务书

**课程名称：软件工程基础实训II**

**实训题目：基于C/S架构的餐饮管理系统**

**已知技术参数和设计要求：**

1. 问题描述（功能要求）：

本次案例将完成一个基于C/S架构的餐饮管理系统。

（1）.进行餐饮管理系统的功能设计。

（2）.设计餐饮管理系统的数据库。

（3）.利用JAVA的GUI类设计开发友好的人机接口。

（4）.利用JAVA语言设计监听用户操作和JDBC进行数据库的处理。

餐饮管理系统主要有6大模块：

（1）.员工管理：对员工实现增删改查。

（2）.客户管理：对客户实现增删改查。

（3）.餐台管理：对餐台实现增删改查。

（4）.菜品管理：对菜品分类、菜品实现增删改查。

（5）.点菜管理：服务员对某客户、某一空闲餐台实行开台，同时实现点菜，将餐台号与所点的菜品对应起来，分别显示出来，并记录开台时间。

（6）.结账管理：收银员对某一餐台通过统计显示消费的菜品清单统计出消费金额，通过手动输入实收金额进行找零的计算，并显示，完成结账的操作，并记录成统计数据。

基础实训II有2周时间40学时，学生基本必做任务：独立完成基于C/S架构的餐饮管理系统。

选做扩展：菜品推荐、销售统计。

**2. 运行环境要求：**

（1）客户端：

windows操作系统，JDK1.6，Eclipse。

（2）服务器：

MySQL5.6或 SQL Server 2008 数据库服务器。

3. 技术要求：

1） 掌握软件工程的需求分析和系统设计方法。

2) 掌握数据库的分析与设计，完成餐饮管理系统数据库的设计。

3) 掌握JAVA的GUI设计方法，完成登录、主界面、菜品分类管理界面、菜品管理界面等人机接口。

4） 掌握JAVA面向对象程序设计方法，设计系统所需的各种类。

5) 掌握JDBC方法，对数据库进行操作，完成主程序和菜品分类、菜品管理等功能。

**实训工作量：**

40课时

**工作计划：**

1.班级

20软件1-4

2.课时及教室安排

见实训课程工作计划表。

实训报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 任昊轩 | 学号 | B20210304119 | 班级 | 21软件1班 |
| 专业 | 软件工程 | | 指导教师姓名 | 杨刚、潘怡 | |
| 项目名称 | 基于C/S架构的餐饮管理系统 | | | | |
| **实训过程**：  需求分析：为了解决现代餐饮行业管理中出现的销售信息、管理信息的问题，开发一个基于C/S架构的餐饮管理系统，对餐饮行业进行现代化、便捷化、科学化的管理，实现对员工、客户、菜品以及餐台信息的增删改查；服务员对某客户、某一空闲餐台实行开台，同时实现点菜，将餐台号与所点的菜品对应起来，分别显示出来，并记录开台时间；收银员对某一餐台通过统计显示消费的菜品清单统计出消费金额，通过手动输入实收金额进行找零的计算，并显示，完成结账的操作。  系统设计：系统功能包括员工管理模块、客户管理模块、餐台管理模块、菜品管理模块、客户点菜模块、以及结账管理模块。  编码实现：利用Java的GUI设计人机交互界面；利用Java的JDBC所提供的方法对数据库进行操纵。  系统测试：测试系统的完备性、实用性和可靠性。  交付实施：经测试成功后对系统进行维护和运营。 | | | | | |
| **实训成果**：  实现了餐饮管理系统的基本功能，界面美观、人机交互友好 | | | | | |
| **实训总结**：  心得体会：通过这两周的实训，我对Java的GUI编程和JDBC操作数据库有了更深刻的认识，在编写代码时学会了合理利用数据结构和算法去解决实际遇到的问题，学会了如何设计以及美化界面，实现界面良好的人机交互功能，在遇到疑难问题时，先独立思考，其次查阅网络资料、和同学进行讨论，最终得到自己满意的结果。  存在的问题：数据库安全性有待加强，存在恶意攻击的风险。对实务的并发未作处理。  改进方向：  一、加强数据库的安全防护措施：  1、使用安全的密码策略  2、使用安全的帐号策略  3、加强数据库日志的记录  4、管理扩展存储过程  5、使用协议加密  6、不要让人随便探测到你的TCP/IP端口  7、修改TCP/IP使用的端口  8、拒绝来自1434端口的探测  9、对网络连接进行IP限制  10、安装数据库审计系统  二、引入多线程对事务进行并发处理 | | | | | |

基于C/S架构的餐饮管理系统

设计说明书

作者：任昊轩(B20210304119)

计算机科学与工程学院

2023年 06 月 15日

**摘要**

本课程设计的主要内容为设计一个基于C/S架构的餐饮管理系统。

本课程设计基于Java GUI编程，数据库编程，Java JDBC编程。

本系统包含客户管理、员工管理、菜品管理、餐台管理、点菜管理和结账管理六大基本模块。客户管理实现对客户信息的增删改查；员工管理实现对员工信息的增删改查；菜品管理实现对菜品信息的增删改查；餐台管理实现对餐台信息的增删改查；点菜管理实现服务员对客户的人数进行开台，并进行点菜和退菜，实现购物车的功能；结账管理实现对订单的查询和修改功能。

**关键词：**C/S架构，餐饮管理系统，Java GUI编程，Java JDBC

目录

[1 引言 1](#_Toc17615)

[1.1 编写目的 1](#_Toc29175)

[1.2 参考资料 1](#_Toc14714)

[2 需求规约 2](#_Toc6703)

[2.1 功能需求 2](#_Toc9670)

[2.2 界面需求 8](#_Toc16271)

[2.3 数据需求 8](#_Toc11128)

[3 系统设计 9](#_Toc12066)

[3.1 运行环境 9](#_Toc13407)

[3.2 系统静态结构设计 9](#_Toc166)

[3.3 人机接口设计 10](#_Toc8649)

[3.4 XXX模块设计说明 12](#_Toc7852)

[3.5 XXX模块设计说明 12](#_Toc28817)

[4 数据库设计 13](#_Toc9449)

[4.1 数据库环境说明 13](#_Toc16056)

[4.2 数据库的命名规则 13](#_Toc28151)

[4.3 逻辑结构设计 13](#_Toc18085)

[4.4 物理结构设计 14](#_Toc13322)

[5 测试用例设计 16](#_Toc17790)

[附录 18](#_Toc1130)

[附录1 程序运行结果 18](#_Toc11581)

[附录2 程序源代码 19](#_Toc949)

# 1 引言

## 1.1 编写目的

本文档是“餐饮管理系统”的软件设计说明书。编写目的是：明确软件需求、概要设计、详细设计和功能测试用例，为软件后续开发和维护提供指导。本文档的读者对象是需求分析人员、系统设计人员、系统开发人员、系统测试人员。

## 1.2 参考资料

[1] 徐转运, 张扬, 王淼. Java高级程序设计[M]. 北京:清华大学出版社, 2014.

# 2 需求规约

## 2.1 功能需求

本次案例将完成一个基于C/S架构的餐饮管理系统。

（1）.进行餐饮管理系统的功能设计。

（2）.设计餐饮管理系统的数据库。

（3）.利用JAVA的GUI类设计开发友好的人机接口。

（4）.利用JAVA语言设计监听用户操作和JDBC进行数据库的处理。

餐饮管理系统主要有6大模块：

（1）.员工管理：对员工实现增删改查。

（2）.客户管理：对客户实现增删改查。

（3）.餐台管理：对餐台实现增删改查。

（4）.菜品管理：对菜品分类、菜品实现增删改查。

（5）.点菜管理：服务员对某客户、某一空闲餐台实行开台，同时实现点菜，将餐台号与所点的菜品对应起来，分别显示出来，并记录开台时间。

（6）.结账管理：收银员对某一餐台通过统计显示消费的菜品清单统计出消费金额，通过手动输入实收金额进行找零的计算，并显示，完成结账的操作，并记录成统计数据。

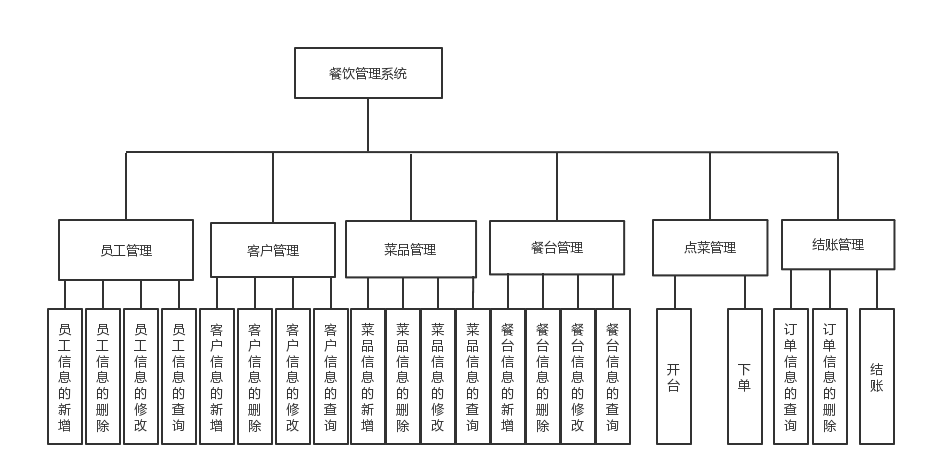


图2.1餐饮管理系统功能模块图

2.1.1 用户登录用例说明

用例名称：用户登录。

用例描述：餐厅管理员登录餐厅管理系统。

前置条件：管理员输入账号和密码。

后置条件：管理员登录成功。

活动步骤：

1. 管理员输入账号和密码
2. 点击登录按钮，管理员登录成功。

异常处理：

1. 用户账号或密码为空，用例终止。
2. 用户输入账号不存在或密码不正确，用例终止。
3. 用户输入账号或密码不正确超过次数，程序终止。

2.2.2 员工管理用例说明

用例名称：员工管理。

用例描述：餐厅管理员对员工信息进行增删改查。

前置条件：餐厅管理员对员工信息进行筛选。

后置条件：餐厅管理员对员工信息进行一系列操作。

活动步骤：

查询操作：

1. 餐厅管理员在文本框内输入员工信息，包括员工id，员工姓名，若未输入员工信息，系统将查询到所有员工。
2. 餐厅管理员查询的信息将在表格中显示。

新增操作：

1. 餐厅管理员点击新增员工按钮，弹出界面。
2. 餐厅管理员在新的界面里填写员工信息。
3. 员工信息添加成功后立即更新查询后的表格。

修改操作：

1. 餐厅管理员点击表格信息，弹出修改的界面。
2. 餐厅管理员在新的界面内的输入框中填写要修改的信息。
3. 员工信息修改成功后立即更新查询后的表格。

删除操作：

1. 餐厅管理员点击表格中该员工所属的一行。
2. 餐厅管理员点击删除按钮。
3. 员工信息删除后立即更新查询后表格。

异常处理：

新增操作：若输入员工信息有误，系统将提示输入信息有误。

修改操作：若输入要修改的信息有误，系统将提示修改错误，将不会对表格信息进行更新。

2.2.3 客户管理用例

用例名称：客户管理。

用例描述：餐厅管理员对客户信息进行增删改查。

前置条件：餐厅管理员对客户信息进行筛选。

后置条件：餐厅管理员对客户信息进行一系列操作。

活动步骤：

查询操作：

1. 餐厅管理员在文本框内输入客户信息，包括客户id，客户姓名，若未输入客户信息，系统将查询到所有客户。
2. 餐厅管理员查询的信息将在表格中显示

新增操作：

1. 餐厅管理员点击新增客户按钮，弹出界面。
2. 餐厅管理员在新的界面里填写客户信息。
3. 客户信息添加成功后立即更新查询后的表格。

修改操作：

1. 餐厅管理员点击表格信息，弹出修改的界面。
2. 餐厅管理员在新的界面内的输入框中填写要修改的信息。
3. 客户信息修改成功后立即更新查询后的表格。

删除操作：

1. 餐厅管理员点击表格中该客户的id。
2. 餐厅管理员点击删除按钮。
3. 客户信息删除后立即更新查询后表格。

异常处理：

新增操作：若输入客户信息有误，系统将提示信息有误。

修改操作：若输入要修改的信息有误，系统将提示修改错误，将不会对表格信息进行更新。

2.2.4 菜品管理用例

用例名称：菜品管理。

用例描述：餐厅管理员对菜品信息进行增删改查。

前置条件：餐厅管理员对菜品信息进行筛选。

后置条件：餐厅管理员对菜品信息进行一系列操作。

活动步骤：

查询操作：

1. 餐厅管理员在文本框内输入菜品信息，包括菜品id，菜名，若未输入菜品信息，系统将查询到所有菜品。
2. 餐厅管理员查询的信息将在表格中显示

新增操作：

1. 餐厅管理员点击新增菜品按钮，弹出界面。
2. 餐厅管理员在新的界面里填写菜品信息。
3. 菜品信息添加成功后立即更新查询后的表格。

修改操作：

1. 餐厅管理员点击表格信息，弹出修改的界面。
2. 餐厅管理员在新的界面内的输入框中填写要修改的信息。
3. 客户信息修改成功后立即更新查询后的表格。

删除操作：

1. 餐厅管理员点击表格中该菜品的id。
2. 餐厅管理员点击删除按钮。
3. 菜品信息删除后立即更新查询后表格。

异常处理：

新增操作：若输入菜品信息有误，系统将提示具体信息有误。

修改操作：若输入要修改的信息有误，系统将提示修改错误，将不会对表格信息进行更新。

2.2.5 餐台管理用例

用例名称：餐台管理。

用例描述：餐厅管理员对餐台信息进行增删改查。

前置条件：餐厅管理员对餐台信息进行筛选。

后置条件：餐厅管理员对餐台信息进行一系列操作。

活动步骤：

查询操作：

1. 餐厅管理员在文本框内输入餐台号，若未输入餐台号，系统将查询到所有餐台。
2. 餐厅管理员查询的信息将在表格中显示

新增操作：

1. 餐厅管理员点击新增餐台按钮，弹出界面。
2. 餐厅管理员在新的界面里填写餐台的座位数。
3. 菜品信息添加成功后立即更新查询后的表格。

修改操作：

1. 餐厅管理员点击表格信息，弹出修改的界面。
2. 餐厅管理员在新的界面内的输入框中填写要修改的信息。
3. 客户信息修改成功后立即更新查询后的表格。

删除操作：

1. 餐厅管理员点击表格中该餐台的餐台号。
2. 餐厅管理员点击删除按钮。
3. 餐台信息删除后立即更新查询后表格。

异常处理：

修改操作：若输入要修改的信息有误，系统将提示修改错误，将不会对表格信息进行更新。

2.2.6 点菜管理用例

用例名称：点菜管理。

用例描述：餐厅管理员对客户进行开台，点菜，下单操作。

前置条件；餐厅管理员输入客户的客户id和人数，对客户进行开台操作。

后置条件：餐厅管理员为客户完成点单，下单操作。

活动步骤：

1. 餐厅管理员输入客户id以及人数，对客户进行开台操作。
2. 系统将根据人数为该用户分配合适的餐台。
3. 餐厅管理员根据菜品信息来查询菜品，提供给客户去选菜。
4. 选菜完成后点击加入购物车后再点击下单，菜品添加到购物车中。
5. 确认无误后点击保存，生成新的订单。

异常处理：

1. 餐厅管理员输入错误的客户id或着非法的人数时，系统会提示具体的输入错误，并且不会对客户进行开台操作。
2. 餐厅管理员输入的人数大于现有空闲餐台的所能容纳的最大人数时，系统将提示找不到合适的餐台。

2.2.7 结账管理用例

用例名称：结账管理。

用例描述：餐厅管理员对订单进行查询，删除，结账的操作。

前置条件：餐厅管理员输入订单号或客户id，对订单进行模糊查询。

后置条件：餐厅管理员对所选订单进行结账处理。

活动步骤：

1. 餐厅管理员输入订单号或客户id，对订单进行查询。
2. 系统将结果显示在表格中。
3. 餐厅管理员选定某一订单，弹出新的界面。
4. 在新的界面中显示订单的应付金额与折后价格以及实付金额。
5. 餐厅管理员在实付金额处填写客户支付的金额，点击支付，系统将对该客户进行找零操作，同时更新查询后的表格。

异常处理：餐厅管理员输入金额小于折后价格，将提醒支付失败。

## 2.2 界面需求

1. 界面友好，操作简单。

2. 界面美观，用户体验感良好。

## 2.3 数据需求

表2.1 用户实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| name | char | 255 | 否 | 是 | 用户名 |
| Passwd | char | 255 | 是 | 否 | 密码 |

表2.2客户实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| customerId | int | 11 | 0 | 0 | 0 |
| name | varchar | 20 | 0 | 0 | 0 |
| sex | varchar | 2 | 0 | -1 | 0 |
| company | varchar | 20 | 0 | -1 | 0 |
| tel | varchar | 11 | 0 | -1 | 0 |
| cardID | varchar | 11 | 0 | 0 | 0 |

表2.3餐台实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| deskID | char | 23 | 否 | 是 | 餐台号 |
| status | char | 10 | 是 | 否 | 状态 |
| seating | int | 0 | 是 | 否 | 座位数 |

表2.4菜品实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| dish\_ID | char | 23 | 否 | 是 | 菜品号 |
| dish\_name | char | 10 | 是 | 否 | 菜名 |
| categoryID | int | 0 | 是 | 否 | 菜品分类编号 |
| code | char | 10 | 是 | 否 | 菜品代码 |
| unit | char | 10 | 是 | 否 | 单位 |
| price | char | 10 | 是 | 否 | 价格 |
| State | char | 4 | 是 | 否 | 状态 |

表2.5员工实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| employee\_ID | char | 23 | 否 | 是 | 员工号 |
| employee\_name | char | 10 | 否 | 否 | 姓名 |
| sex | char | 1 | 是 | 否 | 性别 |
| brith | char | 11 | 是 | 否 | 生日 |
| IDcard | char | 20 | 是 | 否 | 身份证号码 |
| nativep | char | 32 | 是 | 否 | 家庭住址 |
| tel | char | 20 | 是 | 否 | 电话号码 |
| position | char | 10 | 否 | 否 | 职位 |
| freeze | char | 4 | 否 | 否 | 是否在职 |

表2.6订单实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| orderNo | char | 20 | 否 | 是 | 订单序号 |
| orderId | char | 23 | 否 | 是 | 订单编号 |
| deskID | int | 11 | 否 | 否 | 就餐时间 |
| createtime | datetime | 32 | 否 | 否 | 创建时间 |
| money | double | 10 | 否 | 否 | 总金额 |
| customerId | int | 11 | 否 | 否 | 客户编号 |
| status | Char | 4 | 否 | 否 | 状态 |
| Numbers | Int | 11 | 否 | 否 | 人数 |

# 3 系统设计

## 3.1 运行环境

操作系统：Windows 11

数据库：MySQL 8.0.33

## 3.2 系统静态结构设计

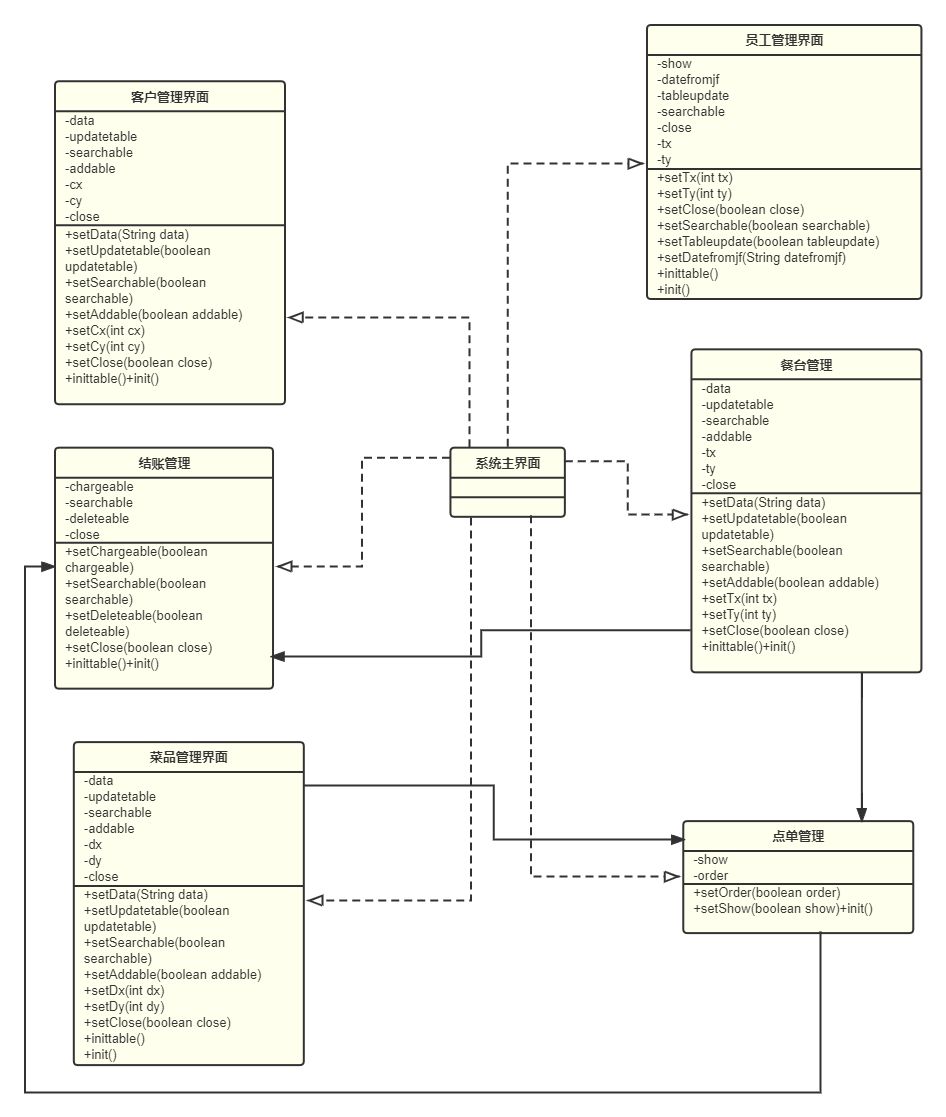


图3.1餐饮管理系统总体类图

系统静态结构说明

餐厅管理员通过主界面可以跳转至不同的界面，在员工管理界面，管理员可以实现对员工信息的增删改查，在客户管理界面，管理员可以实现对客户信息的增删改查，在菜品管理界面，管理员可以实现对菜品信息的增删改查，在餐台界面，管理员可以实现对餐台信息的增删改查，在点菜管理界面，管理员可以为顾客提供点菜服务，点菜完成后生成新的订单，在结账管理界面，管理员可以对订单进行查询和结账操作。

## 3.3 人机接口设计

3.3.1 设计应用类

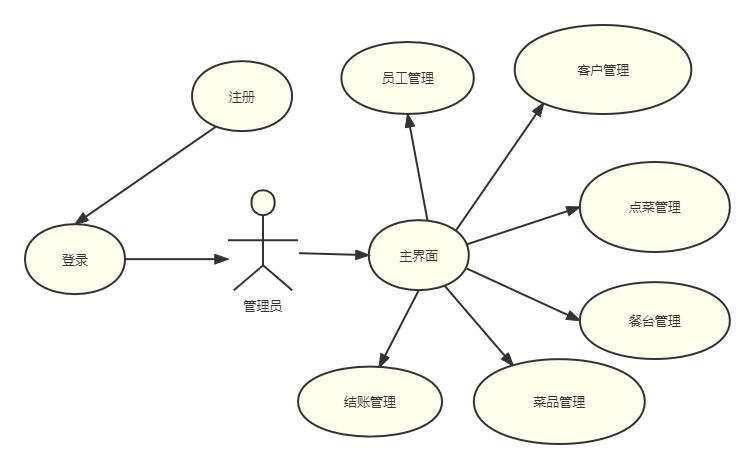


图3.2餐饮管理系统设计类图

3.3.2 用户接口1（登录界面）设计

**界面主要控件：**

JTextField，输入框，接收账号和密码。

JButton，按钮，触发登录注册事件。

**实现功能：**

1. 餐厅管理员输入账号密码，点击登录按钮跳转餐厅管理界面；账号设置为10位数字，密码长度为8到16位的数字或字母，字母区分大小写。
2. 餐厅管理员点击注册按钮跳转注册界面。



图3.3用户登录界面布局设计

3.3.3 用户接口2（注册界面）设计

**界面主要控件：**

JTextField，输入框，接收账号和密码。

JButton，按钮，触发点击事件

**实现功能：**

1. 餐厅管理员输入账号密码，点击注册按钮完成注册；账号设置为10位数字，密码长度位8到16位的数字或字母，字母区分大小写。



图3.4用户注册界面布局设计

3.3.4 用户接口3（餐饮管理系统主界面）设计

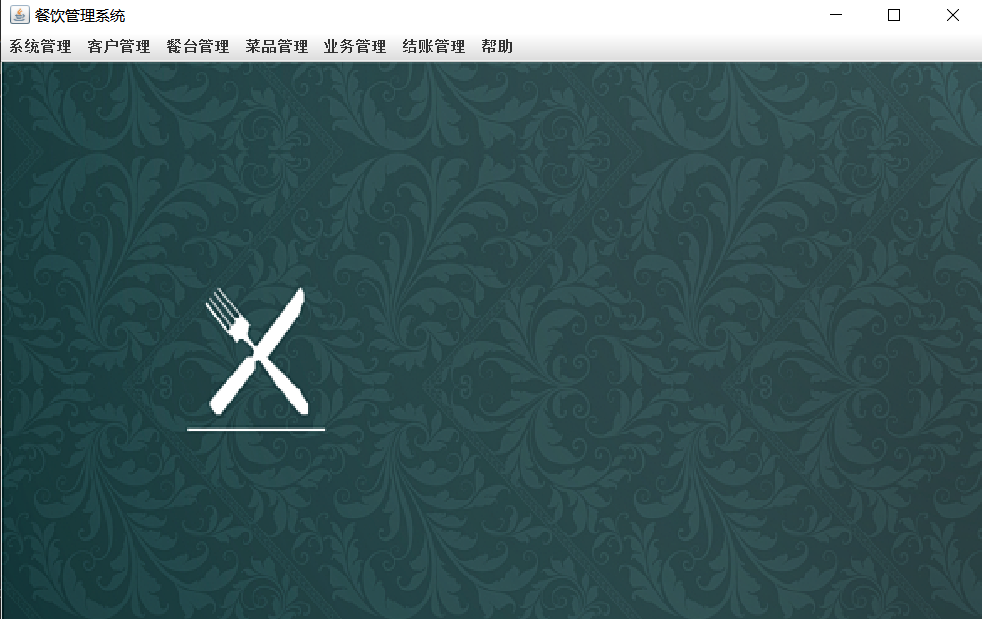


图3.5餐饮管理系统主界面

**界面主要控件：**

JButton，按钮，触发点击事件。

**实现功能：**

1. 点击员工管理，跳转至员工管理界面。
2. 点击客户管理，跳转至客户管理界面。
3. 点击餐台管理，跳转至餐台管理界面。
4. 点击菜品管理，跳转至菜品管理界面。
5. 点击结账管理，跳转至结账管理界面。
6. 点击登出按钮，退出餐饮管理系统。

3.3.5 用户接口4（员工管理界面）设计

**界面主要控件：**

JButton，按钮，触发点击事件。

JTextField，输入框，接收员工id和员工姓名。

**实现功能：**

1. 点击查询按钮，将对员工进行查询。
2. 点击新增按钮，将对进行员工的增添。
3. 点击表格中的某个员工后再点击删除员工，将对该员工进行删除。
4. 点击表格，将对员工信息进行修改。



图3.6员工管理界面

## 3.3.6 用户接口5（客户管理界面）设计

**界面主要控件：**

JButton，按钮，触发点击事件。

JTextField，输入框，接收客户id和客户姓名。

**实现功能：**

1. 点击查询按钮，将对客户进行查询。
2. 点击新增按钮，将对进行客户的增添。
3. 点击表格中的某个客户后再点击删除客户，将对该客户进行删除。
4. 点击表格，将对客户信息进行修改。

图3.7客户管理界面

3.3.7 用户接口6（餐台管理界面）设计



图3.8餐台管理界面

**界面主要控件：**

JButton，按钮，触发点击事件。

JTextField，输入框，接收餐台号。

**实现功能：**

1. 点击查询按钮，将对餐台进行查询。
2. 点击新增按钮，将对进行餐台的增添。
3. 点击表格中的某个餐台后再点击删除餐台，将对该餐台进行删除。
4. 点击表格，将对餐台信息进行修改。

3.3.8 用户接口7（菜品管理界面）设计



## 图3.9菜品管理界面

**界面主要控件：**

JButton，按钮，触发点击事件。

JTextField，输入框，接收菜品id和菜品名。

**实现功能：**

1. 点击查询按钮，将对菜品进行查询。
2. 点击新增按钮，将对进行菜品的增添。
3. 点击表格中的某个菜品后再点击删除菜品，将对该菜品进行删除。
4. 点击表格，将对菜品信息进行修改。

3.3.9 用户接口8（点菜管理界面）设计

**界面主要控件：**

界面1：

JButton，按钮，触发点击事件。

JTextField，输入框，接收客户id和就餐人数。

界面2：

JButton，按钮，触发点击事件。

JTextField，输入框，接收菜名。

JComboBox，下拉列表，对菜品信息进行筛选。

**实现功能：**

界面1：

1. 输入客户编号和就餐人数，点击开台，系统根据人数进行分配合适的餐台。
2. 点击开始点菜跳转至界面2

界面2：

1. 输入菜名和选择菜品类型，点击搜索，系统筛选出符合要求的菜品。
2. 点击清空选择，清空JPanel中的所有组件，并且清空文本框，将下拉列表复位
3. 点击开始点菜，选择菜品，点击下单，菜品将添加到订单中



图3.10点菜管理界面（1）



图3.10点菜管理界面（2）

3.3.10 用户接口9（结账管理界面）设计

**界面主要控件：**

JButton，按钮，触发点击事件。

JTextField，输入框，接收订单号和客户id。

**实现功能：**

1. 点击查询按钮，将对订单进行查询。
2. 点击表格中的某个餐台后再点击删除订单，将对该订单进行删除。
3. 点击订单，将对订单进行结账操作。



## 图3.11结账管理界面

## 3.4 注册模块设计说明

（1）功能：该模块是餐饮系统对管理员进行注册操作。

（2）测试要点：

1. 输入的账号或密码为空将提示账号或密码为空。

2. 输入的账号密码错误则提示账号密码有误。

（3）程序流程图如图3.12所示：

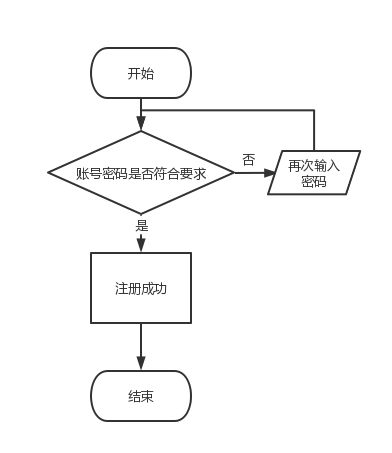


图3.12注册模块流程图

## 3.5 登录模块设计说明

（1）功能：该模块主要是管理员对餐饮管理系统进行登录操作。

（2）测试要点：

1. 输入的账号或密码为空将提示账号或密码为空。

2. 输入的账号密码错误则提示账号密码有误。

（3）程序流程图如图3.13所示：

## wps

图3.13登录模块程序流程图

## 3.6 员工管理模块设计说明

（1）功能：实现管理员对员工的增删改查。

1. 新增员工，程序流程图如图3.14所示：

测试要点：

1. 点击新增员工按钮，弹出新增员工窗口。
2. 输入员工的信息，若员工信息无误且完整，则提示新增成功。
3. 若信息不完整，且信息有误，系统会给出提醒。

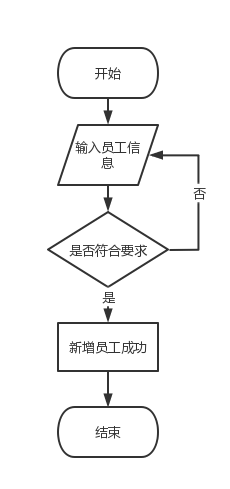


图3.14新增员工信息流程图

1. 删除员工，程序流程图如图3.15所示：



图3.15删除员工流程图

测试要点：

1. 点击表格中员工的信息，点击删除，系统提示删除成功。
2. 修改员工，程序流程图如图3.16所示：

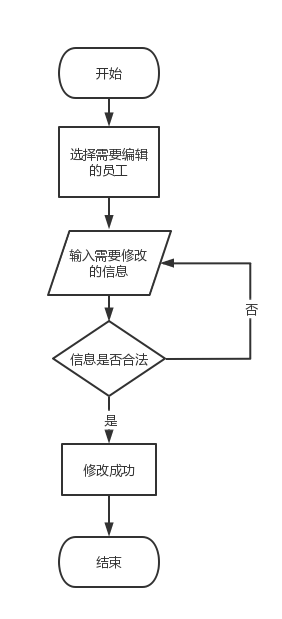


图3.16修改员工信息流程图

测试要点：

1. 点击表格中员工的信息，弹出新的界面。
2. 在新的界面中填写要修改的信息，若输入的信息合法则提示修改成功
3. 若输入的信息不合法，则提示信息不合法。
4. 查询员工，程序流程图如图3.17所示：

测试要点：

1. 在文本框中输入员工的姓名和id，点击查询按钮。
2. 查询信息将在表格中显示。

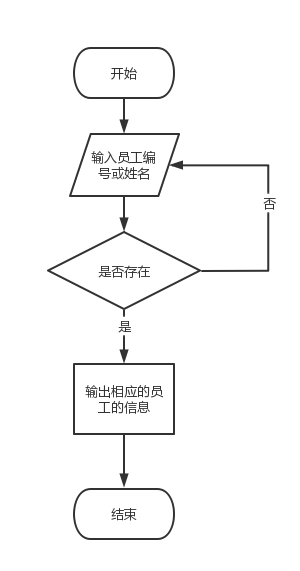


图3.17查询员工信息流程图

## 3.7 客户管理模块设计说明

（1）功能：实现管理员对客户的增删改查。

1. 新增客户，程序流程图如图3.18所示：

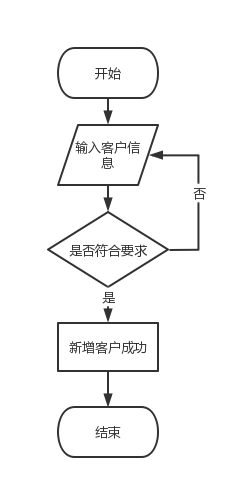


图3.18新增客户流程图

测试要点：

1. 点击新增客户按钮，弹出新增客户窗口。
2. 输入客户的信息，若客户信息无误且完整，则提示新增成功。
3. 若信息不完整，且信息有误，系统会给出提醒。
4. 删除客户，程序流程图如图3.19所示：



图3.19删除客户流程图

测试要点：

1. 点击表格中客户的信息，点击删除，系统提示删除成功。
2. 修改客户，程序流程图如图3.20所示：

测试要点：

1. 点击表格中客户的信息，弹出新的界面。
2. 在新的界面中填写要修改的信息，若输入的信息合法则提示修改成功。
3. 若输入的信息不合法，则提示信息不合法。
4. 查询客户，程序流程图如图3.21所示：

测试要点：

1. 在文本框中输入客户的姓名和id，点击查询按钮。
2. 查询信息将在表格中显示。

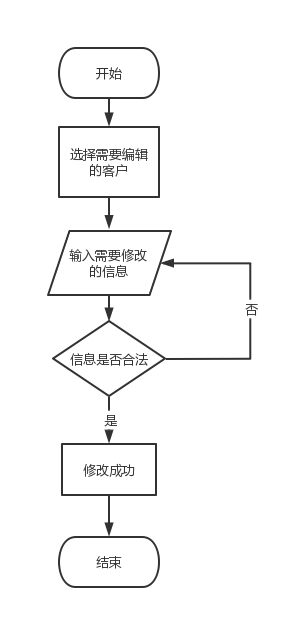
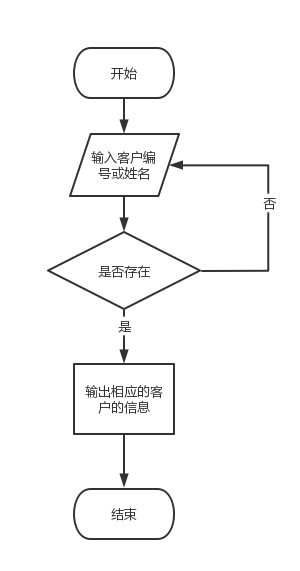
 

图3.20修改客户流程图  示：

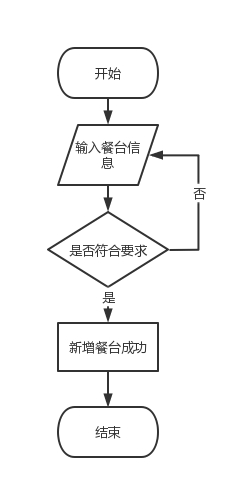


图3.22新增餐台流程图

测试要点：

1. 点击新增餐台按钮，弹出新增餐台界面。
2. 输入餐台的座位数，若输入为数字且合法，则新增餐台成功，否则提示新增失败。
3. 删除餐台，程序流程图如图3.23所示：



图3.23删除餐台流程图

测试要点：

1. 点击餐台信息，点击删除按钮，系统提示删除成功
2. 修改餐台，程序流程图如图3.24所示：

测试要点：

1. 点击表格中餐台的信息，弹出新的界面。
2. 在新的界面中填写要修改的信息，若输入的信息合法则提示修改成功。
3. 若输入的信息不合法，则提示信息不合法。
4. 查询餐台，程序流程图如图3.25所示：

测试要点：

1. 在文本框中输入餐台号，点击查询按钮。
2. 查询信息将在表格中显示。

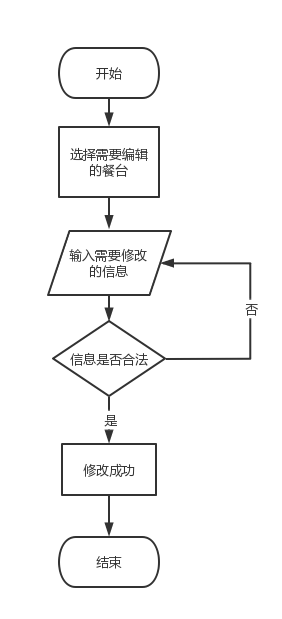
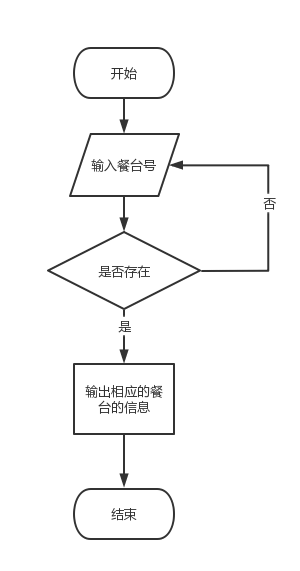
 

图3.24修改餐台流程图 图3.25查询餐台流程图

## 3.9 菜品管理模块设计说明

（1）功能：实现管理员对菜品信息的增删改查

1. 新增菜品，程序流程图如图3.26所示：

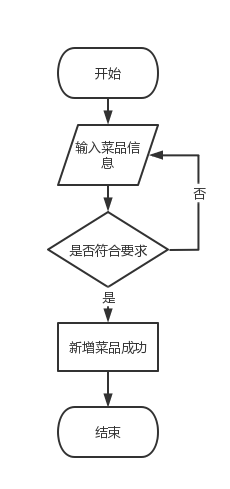


图3.26新增菜品程序流程图

测试要点：

1. 点击新增菜品按钮，弹出新增菜品窗口。
2. 输入菜品的信息，若菜品信息无误且完整，则提示新增成功。
3. 若信息不完整，且信息有误，系统会给出提醒。
4. 删除菜品，程序流程图如图3.27所示：



图3.27删除菜品流程图

测试要点：

1. 点击菜品信息，点击删除按钮，系统提示删除成功。
2. 修改菜品，程序流程图如图3.28所示：

测试要点：

1. 点击表格中餐台的信息，弹出新的界面。
2. 在新的界面中填写要修改的信息，若输入的信息合法则提示修改成功。
3. 若输入的信息不合法，则提示信息不合法。
4. 查询菜品，程序流程图如图3.29所示：

测试要点：

1. 在文本框中输入菜品名或菜品id，点击查询按钮。
2. 查询信息将在表格中显示。

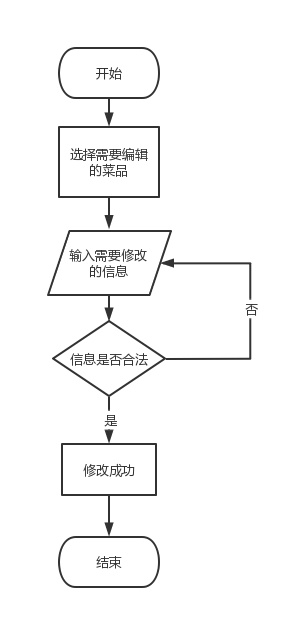
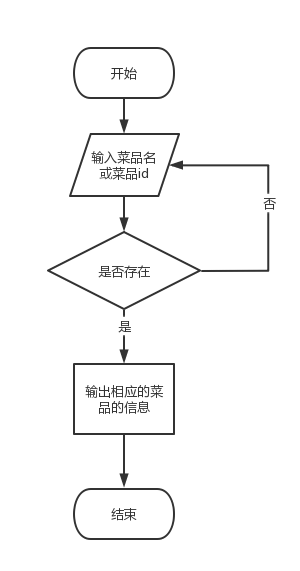
 

图3.28删除菜品程序流程图 图3.29查询菜品程序流程图

## 3.10 点单管理模块设计说明

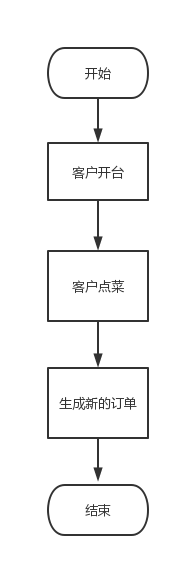
（1）功能：实现管理员对客户进行开台，点单操作。

（2）测试要点：

1. 输入客户的客户号以及人数，为客户开台。

2. 客户选菜，生成新的订单。

（3）程序流程图如图3.30所示：



3.30点菜管理程序流程图

## 3.11 结账管理模块设计说明

（1）功能：实现管理员对订单进行结账管理。

（2）测试要点：

1. 在输入框中输入订单编号或客户编号，查询的订单将在表格中显示。

2. 点击订单，跳转至结账界面，将会根据客户的身份对客户进行优惠。

3. 在新的界面输入金额，找零

（3）程序流程图如图3.31所示：

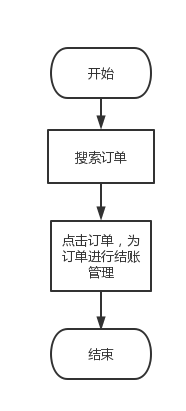


图3.31订单管理流程图

# 4 数据库设计

## 4.1 数据库环境说明

数据库版本：MySQL 8.0.28

运行环境：Windows 11

## 4.2 数据库的命名规则

数据库名：canteen

数据表命名规范：

（1）采用26个英文字母（区分大小写）和0-9的自然数（经常不需要）加上下划线’\_’组成，命名简洁明确，多个单词用下划线’\_’分隔。

（2）全部小写命名，禁止出现大写。

（3）禁止使用数据库关键字，如：name,time, datetime,password等。

（4）表名称不应该取得太长（一般不超过三个英文单词）。

（5）表的名称一般使用名词或者动宾短语。

（6）用单数形式表示名称。

（7）表必须填写描述信息（使用SQL语句建表时）。

字段命名规范：

（1）采用26个英文字母（区分大小写）和0-9个自然数（经常不需要）加下划线’\_’组成，命名简洁明确，多个单词用下划线’\_’分隔。

（2）全部小写命名，禁止出现大写。

（3）字段必须填写描述信息。

（4）禁止使用数据库关键字。

（5）字段名称一般采用名词或动宾短语。

（6）采用字段的名称必须是易于理解，一般不超过三个英文单词。

（7）在命名表的列时，不要重复表的名称。

（8）不要在列的名称中包含数据类型。

（9）字段命名使用完整命名，禁止缩写。

## 4.3 逻辑结构设计

关系模型（主键用下划线标注，双下划线表示既是主键又是外键）

管理员（用户名，密码）

客户（客户编号，客户姓名，性别，出生日期，联系方式，身份）

菜品（菜品编号，菜品名，菜品价格，菜品类别，菜品口味，烹饪方式）

员工（员工编号，员工姓名，员工性别，出生日期，联系方式，就职日期）

餐台（餐台编号，餐台状态，餐台人数）

餐台消费记录（订单编号，餐台编号）

客户消费记录（订单编号，消费时间，客户编号）

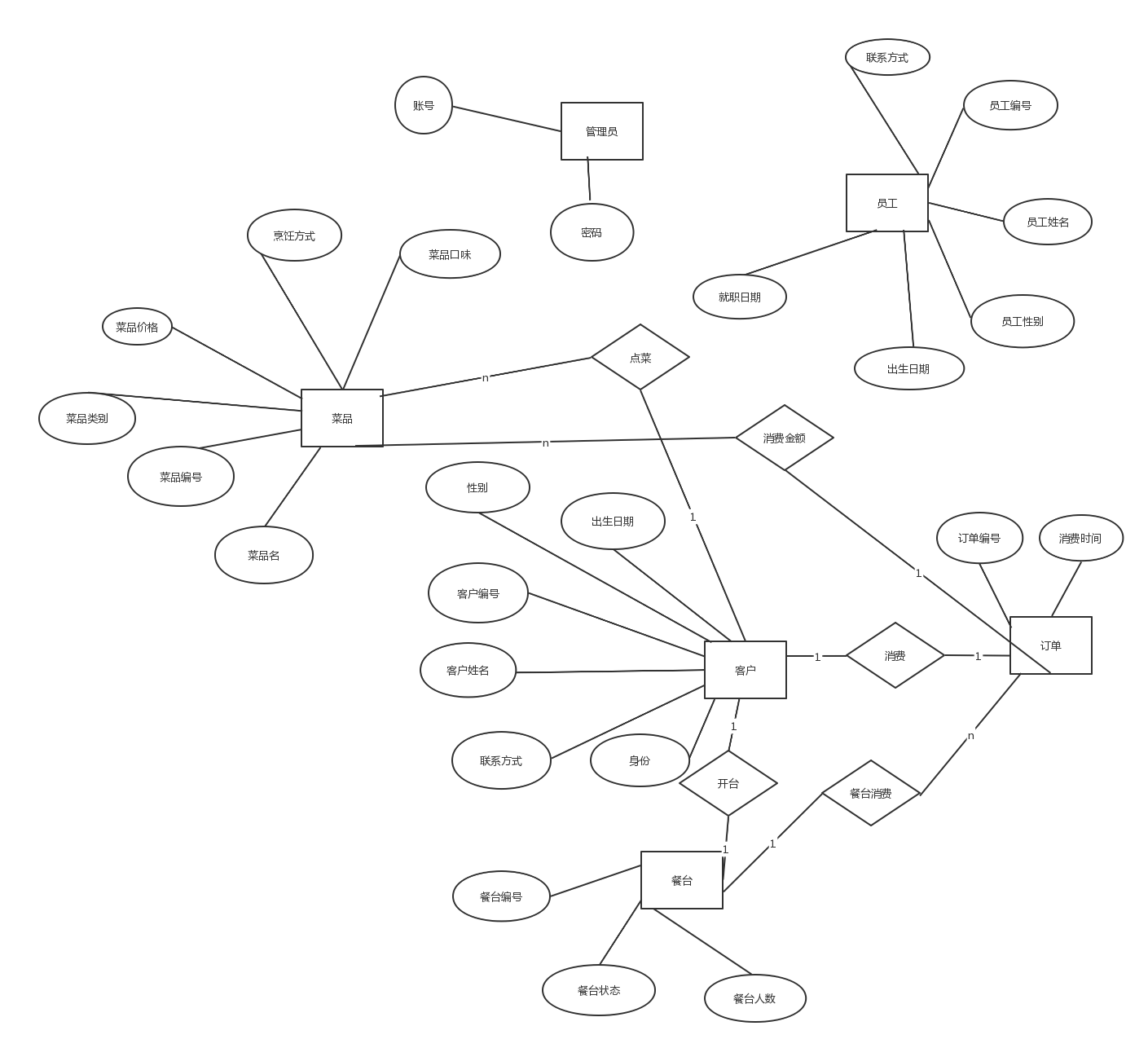
点单（订单编号，菜品编号，数量）

订单（订单编号，餐台编号，客户编号，开台时间，消费金额，订单状态，人数）

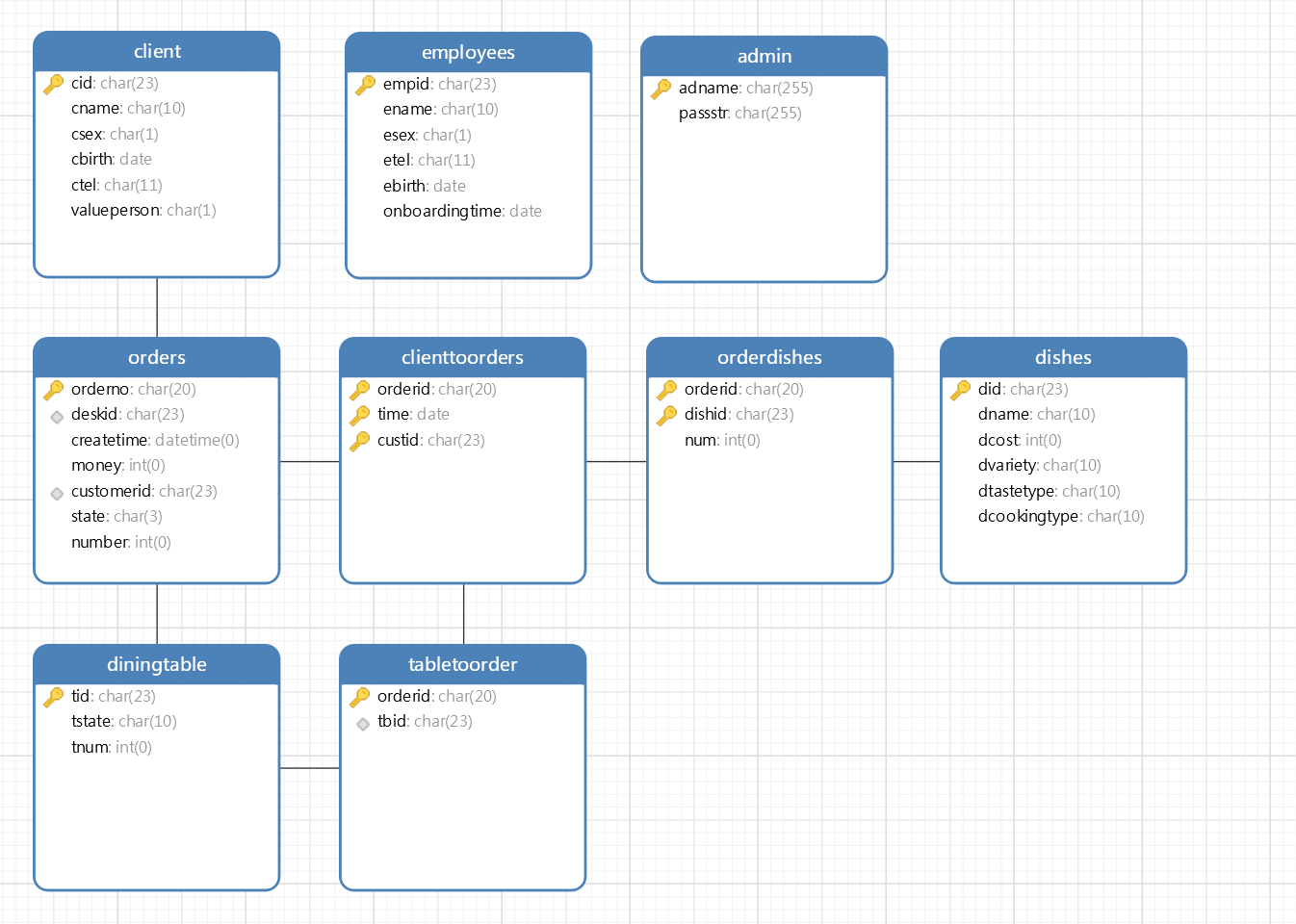
范式分析

所列关系均为第三范式，因为它消除了非主属性对主属性的传递依赖

E-R图



逻辑模型图



## 4.4 物理结构设计

### 4.4.1 数据库表

表4.1 表汇总

|  |  |
| --- | --- |
| 表 名 | 功 能 说 明 |
| admin | 存储管理员账号密码 |
| client | 存储客户信息 |
| diningtable | 存储餐台信息 |
| orders | 存储订单信息 |
| employees | 存储员工信息 |
| clienttoorders | 存储客户订单信息 |
| tabletoorder | 存储餐台订单信息 |
| orderdishes | 存储菜品订单信息 |
| dishes | 存储菜品信息 |

表4.2 管理员信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | admin | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | adname | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | adname | char | N | Y |  | 无 | 10位数字 |
| 2 | passstr | char | Y | N |  | 无 | 8-16位数字或字母 |
| sql脚本 | | CREATE TABLE `admin` (  `adname` char(255) NOT NULL COMMENT '用户名',  `passstr` char(255) DEFAULT NULL COMMENT '密码',  PRIMARY KEY (`adname`),  CONSTRAINT `a1` CHECK ((regexp\_like(`adname`,\_utf8mb4'^[0-9]+$') and (length(`adname`) = 10))),  CONSTRAINT `a2` CHECK ((regexp\_like(`passstr`,\_utf8mb4'^[0-9a-zA-Z]+$') and (length(`passstr`) >= 8) and (length(`passstr`) <= 16)))  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

表4.3 客户信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | client | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | cid | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | cid | char | N | Y |  | 无 | 无 |
| 2 | cname | char | Y | N |  | 无 | 无 |
| 3 | csex | char | Y | N |  | 无 | 无 |
| 4 | cbirth | date | Y | N |  | 无 | 无 |
| 5 | ctel | char | Y | N |  | 无 | 无 |
| 6 | valueperson | char | Y | N |  | 无 | 是、否中的一个 |
| sql脚本 | | CREATE TABLE `client` (  `cid` char(23) NOT NULL COMMENT '客户号',  `cname` char(10) DEFAULT NULL COMMENT '姓名',  `csex` char(1) DEFAULT NULL COMMENT '性别',  `cbirth` date DEFAULT NULL COMMENT '出生日期',  `ctel` char(11) DEFAULT NULL COMMENT '电话号码',  `valueperson` char(1) DEFAULT NULL COMMENT '会员',  PRIMARY KEY (`cid`),  CONSTRAINT `c1` CHECK ((`csex` in (\_utf8mb4'男',\_utf8mb4'女'))),  CONSTRAINT `c2` CHECK ((`valueperson` in (\_utf8mb4'是',\_utf8mb4'否'))),  CONSTRAINT `c3` CHECK (((length(`ctel`) = 11) and regexp\_like(`ctel`,\_utf8mb4'^[0-9]+$')))  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

表4.4 员工信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | employees | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | empid | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | empid | char | N | Y |  | 无 | 无 |
| 2 | ename | char | Y | N |  | 无 | 无 |
| 3 | esex | char | Y | N |  | 无 | 无 |
| 4 | etel | char | Y | N |  | 无 | 无 |
| 5 | ebirth | date | Y | N |  | 无 | 无 |
| 6 | onboardingtime | date | Y | N |  | 无 | 无 |
| sql脚本 | | CREATE TABLE `employees` (  `empid` char(23) NOT NULL COMMENT '员工号',  `ename` char(10) DEFAULT NULL COMMENT '姓名',  `esex` char(1) DEFAULT NULL COMMENT '性别',  `etel` char(11) DEFAULT NULL COMMENT '电话号码',  `ebirth` date DEFAULT NULL COMMENT '出生日期',  `onboardingtime` date DEFAULT NULL COMMENT '入职日期',  PRIMARY KEY (`empid`),  CONSTRAINT `e1` CHECK ((`esex` in (\_utf8mb4'男',\_utf8mb4'女'))),  CONSTRAINT `e2` CHECK ((regexp\_like(`etel`,\_utf8mb4'^[0-9]+$') and (length(`etel`) = 11)))  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

表4.5 菜品信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | dishes | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | did | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | did | char | N | Y |  | 无 | 无 |
| 2 | dname | char | Y | N |  | 无 | 无 |
| 3 | dcost | int | Y | N |  | 无 | 无 |
| 4 | dvariety | char | Y | N |  | 无 | 无 |
| 5 | dtastetype | char | Y | N |  | 无 | 无 |
| 6 | dcookingtype | char | Y | N |  | 无 | 无 |
| sql脚本 | | CREATE TABLE `dishes` (  `did` char(23) NOT NULL COMMENT '菜品号',  `dname` char(10) DEFAULT NULL COMMENT '菜名',  `dcost` int DEFAULT NULL COMMENT '价格',  `dvariety` char(10) DEFAULT NULL COMMENT '品种分类',  `dtastetype` char(10) DEFAULT NULL COMMENT '口味分类',  `dcookingtype` char(10) DEFAULT NULL COMMENT '烹饪方法',  PRIMARY KEY (`did`),  CONSTRAINT `d1` CHECK ((`dvariety` in (\_utf8mb4'牛肉类',\_utf8mb4'猪肉类',\_utf8mb4'家禽类',\_utf8mb4'海鲜类',\_utf8mb4'蔬菜类'))),  CONSTRAINT `d2` CHECK ((`dtastetype` in (\_utf8mb4'咸鲜味',\_utf8mb4'香甜味',\_utf8mb4'酸甜辣味',\_utf8mb4'酸甜味',\_utf8mb4'酱香味',\_utf8mb4'麻辣味',\_utf8mb4'香辣味'))),  CONSTRAINT `d3` CHECK ((`dcookingtype` in (\_utf8mb4'拌',\_utf8mb4'腌',\_utf8mb4'卤',\_utf8mb4'炒',\_utf8mb4'熘',\_utf8mb4'烧',\_utf8mb4'焖',\_utf8mb4'蒸',\_utf8mb4'烤',\_utf8mb4'煎',\_utf8mb4'炸',\_utf8mb4'炖',\_utf8mb4'煮',\_utf8mb4'煲',\_utf8mb4'烩')))  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

表4.6 餐台信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | diningtable | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | tid | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | cid | char | N | Y |  | 无 | 无 |
| 2 | tstate | char | Y | N |  | 无 | 无 |
| 3 | tnum | int | Y | N |  | 无 | 无 |
| sql脚本 | | CREATE TABLE `diningtable` (  `tid` char(23) NOT NULL COMMENT '餐台号',  `tstate` char(10) DEFAULT NULL COMMENT '状态',  `tnum` int DEFAULT NULL COMMENT '座位数',  PRIMARY KEY (`tid`),  CONSTRAINT `diningtable\_chk\_1` CHECK ((`tstate` in (\_utf8mb4'空闲',\_utf8mb4'占用')))  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

表4.7 订单信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | orders | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | orderno | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | orderno | char | N | Y |  | 无 | 外键 |
| 2 | deskid | char | Y | N |  | 无 | 外键 |
| 3 | createtime | datetime | Y | N |  | 无 | 无 |
| 4 | money | int | Y | N |  | 无 | 无 |
| 5 | customerid | char | Y | N |  | 无 | 外键 |
| 6 | state | char | Y | N |  | 无 | 无 |
| 7 | number | int | Y | N |  | 无 | 无 |
| sql脚本 | | CREATE TABLE `orders` (  `orderno` char(20) NOT NULL COMMENT '订单编号',  `deskid` char(23) DEFAULT NULL COMMENT '餐台号',  `createtime` datetime DEFAULT NULL COMMENT '就餐时间',  `money` int DEFAULT NULL COMMENT '消费金额',  `customerid` char(23) DEFAULT NULL COMMENT '客户编号',  `state` char(3) DEFAULT NULL COMMENT '状态',  `number` int DEFAULT NULL COMMENT '就餐人数',  PRIMARY KEY (`orderno`),  KEY `deskid` (`deskid`),  KEY `customerid` (`customerid`),  CONSTRAINT `orders\_ibfk\_1` FOREIGN KEY (`deskid`) REFERENCES `diningtable` (`tid`),  CONSTRAINT `orders\_ibfk\_2` FOREIGN KEY (`customerid`) REFERENCES `client` (`cid`),  CONSTRAINT `orders\_ibfk\_3` FOREIGN KEY (`orderno`) REFERENCES `clienttoorders` (`orderid`),  CONSTRAINT `o1` CHECK ((`state` in (\_utf8mb4'已支付',\_utf8mb4'未支付')))  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

表4.8 客户订单信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | clientoorders | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | orderid,time,custid | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | orderid | char | N | Y |  | 无 | 外键 |
| 2 | time | date | N | N |  | 无 | 外键 |
| 3 | custid | char | N | Y |  | 无 | 外键 |
| sql脚本 | | CREATE TABLE `clienttoorders` (  `orderid` char(20) NOT NULL COMMENT '订单号',  `time` date NOT NULL DEFAULT (curdate()) COMMENT '订单生成日期',  `custid` char(23) NOT NULL COMMENT '客户编号',  PRIMARY KEY (`orderid`,`time`,`custid`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

表4.9 餐台订单信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | tabletoorder | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | orderid | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | orderid | char | N | Y |  | 无 | 无 |
| 2 | tid | char | Y | Y |  | 无 | 无 |
| sql脚本 | | CREATE TABLE `tabletoorder` (  `orderid` char(20) NOT NULL COMMENT '订单编号',  `tbid` char(23) DEFAULT NULL COMMENT '餐台号',  PRIMARY KEY (`orderid`),  KEY `tbid` (`tbid`),  CONSTRAINT `tabletoorder\_ibfk\_1` FOREIGN KEY (`tbid`) REFERENCES `diningtable` (`tid`),  CONSTRAINT `tabletoorder\_ibfk\_2` FOREIGN KEY (`orderid`) REFERENCES `clienttoorders` (`orderid`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

表4.10 菜品订单信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | orderdishes | | | | | |
| 数据库用户 | | 数据库管理员 | | | | | |
| 主 键 | | orderid | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | 无 | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | orderid | char | N | Y |  | 无 | 外键 |
| 2 | dishid | char | N | Y |  | 无 | 外键 |
| 3 | num | char | N | N |  | 无 | 无 |
| sql脚本 | | CREATE TABLE `orderdishes` (  `orderid` char(20) NOT NULL COMMENT '订单编号',  `dishid` char(23) NOT NULL COMMENT '菜品号',  `num` int DEFAULT NULL COMMENT '数量',  PRIMARY KEY (`orderid`,`dishid`),  KEY `dishid` (`dishid`),  CONSTRAINT `orderdishes\_ibfk\_1` FOREIGN KEY (`dishid`) REFERENCES `dishes` (`did`),  CONSTRAINT `orderdishes\_ibfk\_2` FOREIGN KEY (`orderid`) REFERENCES `clienttoorders` (`orderid`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; | | | | | |
| 备注 | |  | | | | | |

### 4.4.3存储过程、函数及触发器的设计

存储过程：

CREATE DEFINER=`root`@`localhost` PROCEDURE `addclient`(nam char(10),sex char(1),tel char(11),birth date,pvalue char(1),out state int)

begin

declare str char(23);

if length(tel)=0 or length(nam)=0

then set state=1;

elseif sex in('男','女') and length(tel)=11 and pvalue in('是','否')

then set state=2;

set str=concat(date\_format(curdate(),'%y%m%d'),round('100'+rand()\*899));

insert into client

values(str,nam,sex,birth,tel,pvalue);

elseif sex not in('男','女') and length(tel)=11 and pvalue in('是','否')

then set state=3;

elseif sex in('男','女') and length(tel)=11 and pvalue not in('是','否')

then set state=4;

else set state=5;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `adddishes`(nam char(10),much char(30),v char(10),taste char(10),cooking char(10),out state int)

begin

declare str char(23);

if v in("牛肉类","猪肉类","家禽类","海鲜类","蔬菜类") and taste in("咸鲜味","香甜味","酸甜辣味","酸甜味","酱香味","麻辣味","香辣味") and cooking in("拌","腌","卤","炒","熘","烧","焖","蒸","烤","煎","炸","炖","煮","煲","烩") and much regexp '^[0-9]+$' then set state=1;

set str=concat('D',date\_format(curdate(),'%y%m%d'),round('100'+rand()\*899));

insert into dishes values(str,nam,much,v,taste,cooking);

elseif much not regexp '^[0-9]+$' then set state=2;

elseif much regexp '^[0-9]+$' and v not in ("牛肉类","猪肉类","家禽类","海鲜类","蔬菜类") then

set state=3;

elseif much regexp '^[0-9]+$' and taste not in("咸鲜味","香甜味","酸甜辣味","酸甜味","酱香味","麻辣味","香辣味")

then set state=4;

elseif much regexp '^[0-9]+$' and cooking not in("拌","腌","卤","炒","熘","烧","焖","蒸","烤","煎","炸","炖","煮","煲","烩") then set state=5;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `addemployee`(nam char(10),sex char(1),tel char(11),birth date,out state int)

begin

declare str char(23);

if length(tel)=0 or length(nam)=0

then set state=1;

elseif sex in('男','女') and length(tel)=11

then set state=2;

set str=concat(date\_format(curdate(),'%y%m%d'),round('100'+rand()\*899));

insert into employees

values(str,nam,sex,tel,birth,curdate());

elseif sex not in('男','女') and length(tel)=11

then set state=3;

else set state=4;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `addtable`(tnum char(10),out state int)

begin

declare str char(23);

if tnum regexp '^[0-9]+$' then

set str=concat('T',date\_format(curdate(),'%y%m%d'),round('100'+rand()\*899));

insert into diningtable values(str,'空闲',tnum);

set state=1;

else set state=2;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `enroll`(ad char(255),pasword char(255),out state int)

begin

if ad regexp '^[0-9]+$' and length(ad)=10

and pasword regexp '^[0-9a-zA-Z]+$' and length(pasword)>=8 and length(pasword)<=16

then set state=1;

insert into admin values(ad,pasword);

elseif length(ad)=0 or length(pasword)=0 then set state=2;

elseif ad not regexp '^[0-9]+$' or length(ad)!=10 then

set state=3;

else set state=4;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `login`(ad char(255),pasword char(255),out state int)

begin

if length(ad)=0 or length(pasword)=0 then set state=1;

elseif exists(select \*from admin where adname=ad and passstr=pasword)

then set state=2;

else set state=3;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `modclient`(id char(23),inde int,indata char(30),out state int)

begin

if inde=1 then update client set cname=indata where cid=id;

set state=1;

elseif inde=2 and indata in('男','女') then

update client set csex=indata where cid=id;

set state=1;

elseif inde=2 and indata not in('男','女') then

set state=2;

elseif inde=4 and length(indata)=11 and indata regexp '^[0-9]+$' then

update client set ctel=indata where cid=id;

set state=1;

elseif inde=4 then

set state=4;

elseif inde=3 and indata regexp '^[0-9-]+$' and indata like '\_\_\_\_-\_\_-\_\_'

or indata like '\_\_\_\_-\_-\_' or indata like '\_\_\_\_-\_\_-\_' or indata like '\_\_\_\_-\_-\_\_' then

if convert(left(indata,4),unsigned integer)>convert(year(curdate()),unsigned integer) then set state=3;

elseif indata like '\_\_\_\_-\_\_-\_\_' then

if left(indata,4)=year(curdate()) and right(left(indata,7),2)=month(curdate())

and convert(right(indata,2),unsigned integer)<=convert(day(curdate()),unsigned integer) then

update client set cbirth=indata where cid=id;

set state=1;

elseif left(indata,4)!=year(curdate()) and convert(right(left(indata,7),2),unsigned integer)<=12 then

if convert(right(indata,2),unsigned integer)<=convert(day(last\_day(concat(left(indata,4),'-',right(left(indata,7),2),'-1'))),unsigned integer) then

update client set cbirth=indata where cid=id;

set state=1;

else set state=3;

end if;

else set state=3;

end if;

elseif indata like '\_\_\_\_-\_-\_' then

if left(indata,4)=year(curdate()) and right(left(indata,6),1)=month(curdate())

and convert(right(indata,1),unsigned integer)<=convert(day(curdate()),unsigned integer) then

update client set cbirth=indata where cid=id;

set state=1;

elseif left(indata,4)!=year(curdate()) and convert(left(indata,4),unsigned integer)<convert(year(curdate()),unsigned integer) then update client set cbirth=indata where cid=id;

set state=1;

else set state=3;

end if;

elseif indata like '\_\_\_\_-\_\_-\_' then

if left(indata,4)=year(curdate()) and right(left(indata,7),2)=month(curdate())

and convert(right(indata,1),unsigned integer)<=convert(day(curdate()),unsigned integer) then

update client set cbirth=indata where cid=id;

set state=1;

elseif left(indata,4)!=year(curdate()) and convert(right(left(indata,7),2),unsigned integer)<=12 then

update client set cbirth=indata where cid=id;

set state=1;

else set state=3;

end if;

else -- '\_\_\_\_-\_-\_\_'

if left(indata,4)=year(curdate()) and right(left(indata,6),1)=month(curdate())

and convert(right(indata,2),unsigned integer)<=convert(day(curdate()),unsigned integer) then

update client set cbirth=indata where cid=id;

set state=1;

elseif left(indata,4)!=year(curdate()) and

convert(right(indata,2),unsigned integer)<=convert(day(last\_day(concat(left(indata,4),'-',right(left(indata,6),1),'-1'))),unsigned integer) then

update client set cbirth=indata where cid=id;

set state=1;

else set state=3;

end if;

end if;

else set state=3;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `moddishes`(id char(23),inde int,indata char(30),out state int)

begin

if inde=1 then update dishes set dname=indata where did=id;

set state=1;

elseif inde=2 and indata regexp '^[0-9]+$' then update dishes set dcost=indata where did=id;

set state=1;

elseif inde=2 then set state=2;

elseif inde=3 and indata in("牛肉类","猪肉类","家禽类","海鲜类","蔬菜类") then update dishes set dvariety=indata where did=id;set state=1;

elseif inde=3 then set state=3;

elseif inde=4 and indata in("咸鲜味","香甜味","酸甜辣味","酸甜味","酱香味","麻辣味","香辣味") then update dishes set dtastetype=indata where did=id;set state=1;

elseif inde=4 then set state=4;

elseif inde=5 and indata in("拌","腌","卤","炒","熘","烧","焖","蒸","烤","煎","炸","炖","煮","煲","烩") then update dishes set dcookingtype=indata where did=id;set state=1;

else set state=5;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `modemployee`(id char(23),inde int,indata char(30),out state int)

begin

if inde=1 then update employees set ename=indata where empid=id;

set state=1;

elseif inde=2 and indata in('男','女') then

update employees set esex=indata where empid=id;

set state=1;

elseif inde=2 and indata not in('男','女') then

set state=2;

elseif inde=3 and length(indata)=11 and indata regexp '^[0-9]+$' then

update employees set etel=indata where empid=id;

set state=1;

elseif inde=3 then

set state=3;

elseif inde=4 and indata regexp '^[0-9-]+$' and indata like '\_\_\_\_-\_\_-\_\_'

or indata like '\_\_\_\_-\_-\_' or indata like '\_\_\_\_-\_\_-\_' or indata like '\_\_\_\_-\_-\_\_' then

if convert(left(indata,4),unsigned integer)>convert(year(curdate()),unsigned integer) then set state=4;

elseif indata like '\_\_\_\_-\_\_-\_\_' then

if left(indata,4)=year(curdate()) and right(left(indata,7),2)=month(curdate())

and convert(right(indata,2),unsigned integer)<=convert(day(curdate()),unsigned integer) then

update employees set ebirth=indata where empid=id;

set state=1;

elseif left(indata,4)!=year(curdate()) and convert(right(left(indata,7),2),unsigned integer)<=12 then

if convert(right(indata,2),unsigned integer)<=convert(day(last\_day(concat(left(indata,4),'-',right(left(indata,7),2),'-1'))),unsigned integer) then

update employees set ebirth=indata where empid=id;

set state=1;

else set state=4;

end if;

else set state=4;

end if;

elseif indata like '\_\_\_\_-\_-\_' then

if left(indata,4)=year(curdate()) and right(left(indata,6),1)=month(curdate())

and convert(right(indata,1),unsigned integer)<=convert(day(curdate()),unsigned integer) then

update employees set ebirth=indata where empid=id;

set state=1;

elseif left(indata,4)!=year(curdate()) and convert(left(indata,4),unsigned integer)<convert(year(curdate()),unsigned integer) then update employees set ebirth=indata where empid=id;

set state=1;

else set state=4;

end if;

elseif indata like '\_\_\_\_-\_\_-\_' then

if left(indata,4)=year(curdate()) and right(left(indata,7),2)=month(curdate())

and convert(right(indata,1),unsigned integer)<=convert(day(curdate()),unsigned integer) then

update employees set ebirth=indata where empid=id;

set state=1;

elseif left(indata,4)!=year(curdate()) and convert(right(left(indata,7),2),unsigned integer)<=12 then

update employees set ebirth=indata where empid=id;

set state=1;

else set state=4;

end if;

else -- '\_\_\_\_-\_-\_\_'

if left(indata,4)=year(curdate()) and right(left(indata,6),1)=month(curdate())

and convert(right(indata,2),unsigned integer)<=convert(day(curdate()),unsigned integer) then

update employees set ebirth=indata where empid=id;

set state=1;

elseif left(indata,4)!=year(curdate()) and

convert(right(indata,2),unsigned integer)<=convert(day(last\_day(concat(left(indata,4),'-',right(left(indata,6),1),'-1'))),unsigned integer) then

update employees set ebirth=indata where empid=id;

set state=1;

else set state=4;

end if;

end if;

else set state=4;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `modtable`(id char(23),inde int,indata char(30),out state int)

begin

if indata regexp '^[0-9]+$' then update diningtable set tnum=indata where tid=id;

set state=1;

else set state=2;

end if;

End

CREATE DEFINER=`root`@`localhost` PROCEDURE `opentable`(cusd char(23),num char(10),out state int,out tableid char(23),out stri char(23))

begin

declare str char(23);

if num not regexp '^[0-9]+$' or length(num)=0 then set state=2;

elseif exists(select \*from clienttoorders where custid=cusd)

and (curdate())=(select time from clienttoorders where custid=cusd) then set state=5;

elseif exists(select \*from client where cid=cusd) then

if exists(select\* from diningtable where tnum>=num and tstate='空闲' order by tnum asc limit 1) then set state=1;

set str=concat('Z',date\_format(curdate(),'%y%m%d'),round('100'+rand()\*899));

insert into clienttoorders(orderid,custid) values(str,cusd);

set tableid=(select tid from diningtable where tnum>=num order by tnum asc limit 1);

update diningtable set tstate='占用' where tid=tableid;

insert into tabletoorder values(str,tableid);

set stri=str;

insert into orders(orderno,deskid,createtime,customerid,state,number) values(str,tableid,now(),cusd,'未支付',num);

else set state=3;

end if;

else set state=4;

end if;

End

4.5 安全性设计

4.5.1 防止用户直接操作数据库的方法

（1）用户标识和鉴定

（2）存取控制

（3）数据加密

### 4.5.2 用户帐号密码的加密方法

1. 防止用户直接操作数据库的方法。用户只能用帐号登录到应用软件，通过应用软

件访问数据库，而没有其他途径操作数据库。

2. 用户帐号密码的加密方法。对用户帐号的密码进行加密处理，确保在任何地方都

不会出现密码的明文。

# 5 测试用例设计

表5.1 登录-注册测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 登录 | yl-001 | 输入‘用户名’和‘密码’ | 2942990081，1237236182 | 登录成功 | 通过 |
| yl-002 | 2942990081，1234567 | 登录失败 | 通过 |
| yl-003 | 29429900811，1237236182 | 登录失败 | 通过 |
| yl-004 | 2942990081,空 | 登录失败 | 通过 |
| yl-005 | 空,1237236182 | 登录失败 | 通过 |
| yl-006 | 2942990081,一个空格 | 登录失败 | 通过 |
| yl-007 | 一个空格，1237236182 | 登录失败 | 通过 |
| yl-008 | 2942990081，密码长度超过32 | 登录失败 | 通过 |
| yl-009 | 用户名超过36，1237236182 | 登录失败 | 通过 |
| 注册 | yl-010 | 用户名，密码 | 8888888888,23216323 | 创建成功 | 通过 |
| y1-011 | 用户名，密码 | 8888888888,1 | 创建失败 | 通过 |

表5.2 员工管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 增加员工 | y2-001 | 点击添加按  钮，输入‘姓  名’，‘性别’，‘出生日期’，‘电话号码’，‘就职日期’ | Anish，女，15273751972,  2002-08-03,2022-06-18 | 新增成功 | 通过 |
| 删除员工 | y2-002 | 点击删除按钮 | Anish | 删除成功 | 通过 |
| 修改员工 | y2-003 | 点击表格 | Anish->A | 修改成功 | 通过 |
| 查询员工 | y2-004 | 输入员工编  号或姓名，点击查询  按钮 | 220618545 | 查询成功 | 通过 |

表5.3 客户管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 增加客户 | y3-001 | 点击添加按  钮，输入‘姓  名’，‘性别’，‘出生日期’，‘电话号码’，‘会员’ | 啦啦啦，女，2022-06-22，12345678901，是 | 新增成功 | 通过 |
| 删除客户 | y3-002 | 点击删除按钮 | 啦啦啦 | 删除成功 | 通过 |
| 修改客户 | y3-003 | 点击表格 | 女->男 | 修改成功 | 通过 |
| 查询客户 | y3-004 | 输入员工编  号或姓名，点击查询  按钮 | 220622787 | 查询成功 | 通过 |

表5.4 菜品管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 增加菜品 | y4-001 | 点击添加按  钮，输入‘菜品名’，‘价格’，‘品种分类’，‘口味分类’，‘烹饪方式’ | 2313，50，家禽类，酱香味，烧 | 新增成功 | 通过 |
| 删除菜品 | y4-002 | 点击删除按钮 | 2313 | 删除成功 | 通过 |
| 修改菜品 | y4-003 | 点击表格 | 50->40 | 修改成功 | 通过 |
| 查询菜品 | y4-004 | 输入菜品编  号或菜名，点击查询  按钮 | D220623595 | 查询成功 | 通过 |

表5.5 餐台管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 增加餐台 | y5-001 | 点击添加按  钮，输入‘座位数’ | 10 | 新增成功 | 通过 |
| 删除餐台 | y5-002 | 点击删除按钮 | T220623148 | 删除成功 | 通过 |
| 修改餐台 | y5-003 | 点击表格 | 10->20 | 修改成功 | 通过 |
| 查询餐台 | y5-004 | 输入菜品编  号或菜名，点击查询  按钮 | T220623148 | 查询成功 | 通过 |

表5.6 点菜管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 开台 | y6-001 | 输入客户id和人数 | 220622545，1 | 开台成功 | 通过 |
| 生成订单 | y6-002 | 点菜后点击保存 | 无 | 操作成功 | 通过 |

表5.7 结账管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 删除订单 | y7-001 | 点击订单后点击删除按钮 | Z220626767 | 删除成功 | 通过 |
| 结账操作 | Y7-002 | 点击订单后输入金额 | 500 | 支付成功 | 通过 |

# 附录

## 附录1 程序运行结果



图1 登录界面

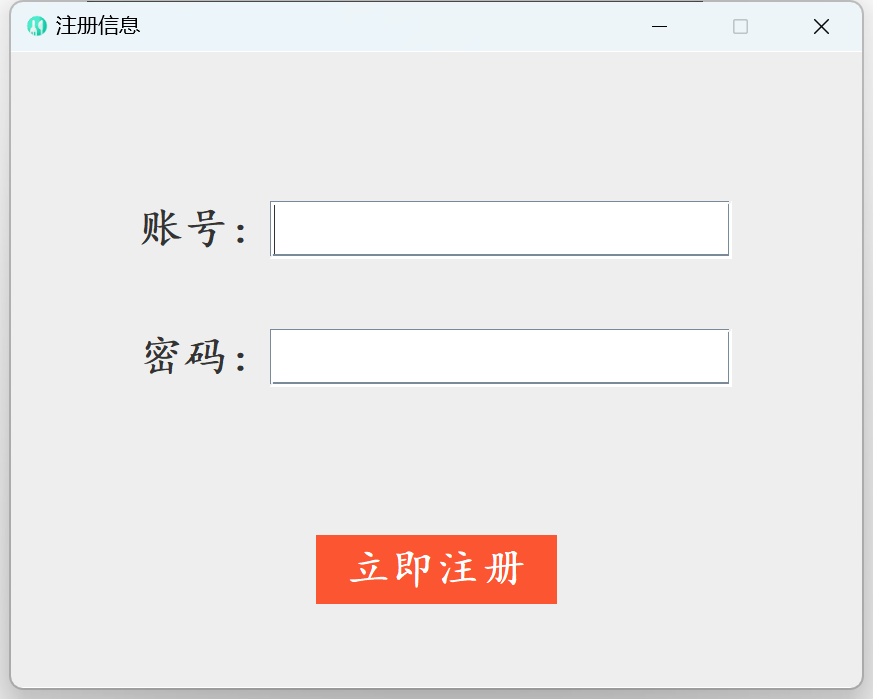


图2 注册界面

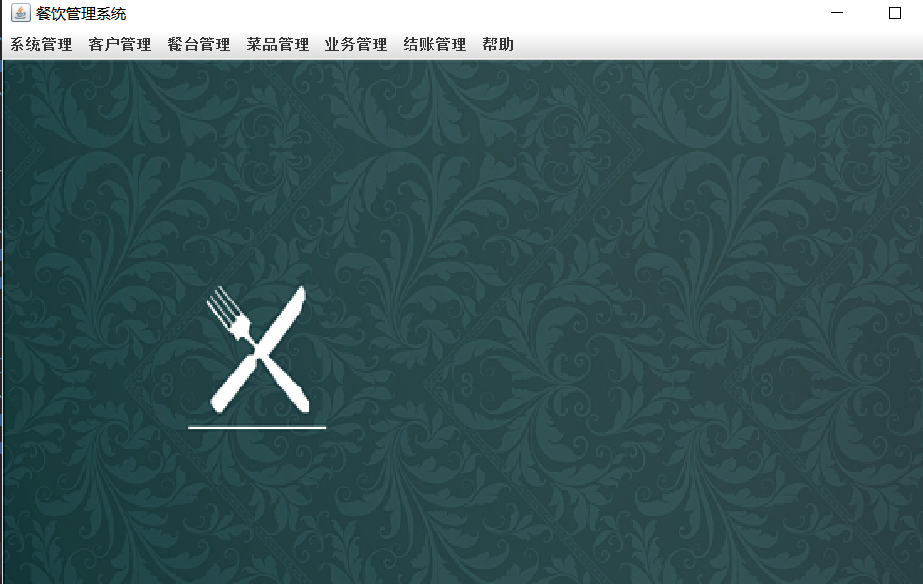


图3 系统主界面



图4 员工管理界面



图5 客户管理界面



图6 餐台管理界面



图7 菜品管理界面



图8 点菜界面1



图9 点菜管理界面2



图10点菜管理界面3



图12 结账管理界面

附录2 程序源代码   
public class CategoryManner {  
    JFrame jf=new JFrame(" 菜 品 分 类 管 理 界 面 ");  
    JPanel controlPanel;  
    //定义表格的数据模型  
    public static DefaultTableModel model;  
    //定义一个表格  
    public static MyTable table;  
    //定义一个滚动面板，用来放置表格  
    JScrollPane scrollPane;  
    JButton back=new JButton("返回主菜单");  
    JButton del=new JButton("删除");  
    JButton cri=new JButton("添加");  
    JButton update=new JButton("修改");  
    JButton LookFor =new JButton("查询");  
    JButton fh =new JButton("返回");  
    public CategoryManner() throws Exception{  
        //注册驱动  
        DatabaseUtil databaseUtil = new DatabaseUtil();  
        JDBC001 jdbc001 = databaseUtil.getJdbc001();  
        Connection coon = jdbc001.getsConnection();  
        Statement statement = jdbc001.getsStament();  
        LookForCategory lookForCategory =new LookForCategory();  
        UpdateCategory updateCategory = new UpdateCategory();  
        Nullchar nullchar = new Nullchar();  
        jf.setContentPane(new JPanel(){  
            public void paintComponent(Graphics g) {  
                super.paintComponent(g);  
  
                //下面这行是为了背景图片可以跟随窗口自行调整大小，可以自己设置成固定大小  
                Image background;  
                try {  
                    background = ImageIO.read(new File("D:\\picture\\a02f3cc968159c646ddf697cbdcfd83d(1).jpeg"));  
                } catch (IOException e) {  
                    throw new RuntimeException(e);  
                }  
                g.drawImage(background, 0, 0, this.getWidth(), this.getHeight(), this);  
            }  
        });  
        jf.setLayout(null);  
        jf.setBounds(300,300,800,600);  
        jf.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);  
        jf.setLocationRelativeTo(null);//窗口居中  
  
        JLabel name = new JLabel("请输入菜品类别：");  
        name.setFont(new Font("黑体", Font.PLAIN, 15));  
        name.setBounds(150,40,130,30);  
        JTextField tf\_1 = new JTextField();  
        tf\_1.setBounds(285,40,120,30);  
        //表头  
        String[] head={"菜品分类编号","菜品类别","描述"};  
        //初始化数据  
        Object[][] datas  = new String[][] {  
        };  
        //初始化表格数据模型  
        model = new DefaultTableModel(datas, head);  
        //用表格模型初始化表格  
        table = new MyTable(model);  
        //初始化滚动面板  
        scrollPane = new JScrollPane(table);  
  
        //初始化控制面板，添加相关按钮  
  
        back.setBounds(20,40,120,30);  
        scrollPane.setBounds(20,80,620,380);  
        cri.setBounds(150,480,80,30);  
        LookFor.setBounds(420,40,80,30);  
        fh.setBounds(520,40,80,30);  
        del.setBounds(240,480,80,30);  
        update.setBounds(330,480,80,30);  
        //设置面板布局，上面控制按钮，中间放表格列表  
  
        //设置表格中的对齐方式  
        DefaultTableCellRenderer renderer = new DefaultTableCellRenderer();  
        renderer.setHorizontalAlignment(SwingConstants.CENTER);  
        //通过列名获取表格的列对象，并将上面设好对齐方式应用到指定列上  
        table.getColumn("菜品分类编号").setCellRenderer(renderer);  
        table.getColumn("菜品类别").setCellRenderer(renderer);  
        table.getColumn("描述").setCellRenderer(renderer);  
  
        //设置表格宽度，以及拖动的最大和最小宽度  
        DefaultTableColumnModel dcm = (DefaultTableColumnModel)table.getColumnModel();  
        dcm.getColumn(0).setPreferredWidth(60);  
        dcm.getColumn(0).setMinWidth(45);  
        dcm.getColumn(0).setMaxWidth(75);  
        dcm.getColumn(0).setPreferredWidth(150);  
        //设置行高  
        //table.setRowHeight(25);  
        try {  
            String strSql = "select \* from category" ;  
            ResultSet rs= statement.executeQuery(strSql);  
            while (rs.next()) {  
                model.addRow(new Object[]{rs.getString("categoryID"),rs.getString("cname"),rs.getString("describ")});  
            }  
  
        }  
        catch (Exception ex) {  
            throw new RuntimeException(ex);  
        }  
        //允许点击JTable的表头进行升降排序  
        table.setAutoCreateRowSorter(true);  
        /tBounds(100,100,600,400);  
  
        //居中显示窗口  
        jf.setLocationRelativeTo(null);  
        int row = table.getSelectedRow();  
        back.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
  
                jf.dispose();//关闭此页面  
                new MenuGui();  
  
            }  
        });  
        cri.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                try {  
                    new AddCategory();  
                } catch (Exception exception) {  
                    exception.printStackTrace();  
                }  
            }  
        });  
        del.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                int row = table.getSelectedRow();  
                if (row != -1) {  
                    String id = (String) table.getValueAt(row,0);  
                    int Id = Integer.parseInt(id);  
                    System.out.println("删除菜品分类编号 : "+id);  
                    try {  
                        PreparedStatement ps = coon.prepareStatement("delete from category where categoryID=?");  
                        ps.setInt(1, Id);  
                        int rs = ps.executeUpdate();  
                        model.removeRow(row);  
                    } catch (Exception throwables) {  
                        throwables.printStackTrace();  
                    }  
                } else {  
                    JOptionPane.showMessageDialog(null,  
                            "没有选中行或者已为空表", "提示",  
                            JOptionPane.PLAIN\_MESSAGE);  
                }  
            }  
        });  
        update.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                int row = table.getSelectedRow();  
                if(row!=-1){  
                    try {  
                        new ModifyCategory();  
                    } catch (Exception exception) {  
                        exception.printStackTrace();  
                    }  
                }  
                else{  
                    JOptionPane.showMessageDialog(null,  
                            "请选择需要修改的菜品类别", "提示",  
                            JOptionPane.PLAIN\_MESSAGE);  
                }  
            }  
        });  
        LookFor.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                if (nullchar.isBlankSpace(tf\_1.getText())) {  
                    JOptionPane.showMessageDialog(null,  
                            "输入信息有误", "提示",  
                            JOptionPane.PLAIN\_MESSAGE);  
                }  
                else {  
                    String name  = tf\_1.getText();  
                    try {  
                        lookForCategory.LookFor(model,name);  
                    } catch (Exception exception) {  
                        exception.printStackTrace();  
                    }  
                }  
            }  
        });  
        fh.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                try {  
                    updateCategory.updata(model);  
                } catch (Exception exception) {  
                    exception.printStackTrace();  
                }  
            }  
        });  
        jf.add(scrollPane);  
        jf.add(name);  
        jf.add(tf\_1);  
        jf.add(LookFor);  
        jf.add(fh);  
        jf.add(back);  
        jf.add(cri);  
        jf.add(del);  
        jf.add(update);  
        jf.setVisible(true);  
  
  
  
    }  
    public static DefaultTableModel getModel(){  
        return model;  
    }  
    public static MyTable getTable(){  
        return table;  
    }  
}

public class CheckoutManner {

DatabaseUtil databaseUtil=new DatabaseUtil();

JPanel controlPanel;

//定义表格的数据模型

public static DefaultTableModel model;

public static MyTable table;

public static String YIN;

public static int deskID;

//定义一个表格

//定义一个滚动面板，用来放置表格

JScrollPane scrollPane;

JButton back = new JButton("返回主菜单");

JButton xg = new JButton("结账");

JButton LookFor = new JButton("查询");

JButton fh = new JButton("返回");

JDBC001 jdbc001 = databaseUtil.getJdbc001();

Connection coon = jdbc001.getsConnection();

Statement statement = jdbc001.getsStament();

LookForOrder lookForOrder = new LookForOrder();

Nullchar nullchar = new Nullchar();

UpdateOrder updateOrder = new UpdateOrder();

public CheckoutManner() throws Exception {

//注册驱动

JFrame jf = new JFrame("账单信息管理");

jf.setContentPane(new JPanel() {

public void paintComponent(Graphics g) {

super.paintComponent(g);

//下面这行是为了背景图片可以跟随窗口自行调整大小，可以自己设置成固定大小

Image background;

try {

background = ImageIO.read(new File("D:\\picture\\a02f3cc968159c646ddf697cbdcfd83d(1).jpeg"));

} catch (IOException e) {

throw new RuntimeException(e);

}

g.drawImage(background, 0, 0, this.getWidth(), this.getHeight(), this);

}

});

jf.setLayout(null);

jf.setBounds(300, 300, 800, 600);

jf.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

jf.setLocationRelativeTo(null);//窗口居中

//表头

String[] head = {"订单编号", "餐台编号", "创建日期","总金额", "客户编号", "状态", "就餐人数"};

//初始化数据

Object[][] datas = new String[][]{

};

//初始化表格数据模型

model = new DefaultTableModel(datas, head);

//用表格模型初始化表格

table = new MyTable(model);

//初始化滚动面板

scrollPane = new JScrollPane(table);

//初始化控制面板，添加相关按钮

back.setBounds(20, 40, 120, 30);

scrollPane.setBounds(20, 80, 700, 380);

xg.setBounds(330, 480, 80, 30);

LookFor.setBounds(420,40,80,30);

fh.setBounds(520,40,80,30);

JLabel name = new JLabel("请输入订单关键字");

name.setFont(new Font("黑体", Font.PLAIN, 15));

name.setBounds(150,40,130,30);

JTextField tf\_1 = new JTextField();

tf\_1.setBounds(285,40,120,30);

//设置面板布局，上面控制按钮，中间放表格列表

//设置表格中的对齐方式

DefaultTableCellRenderer renderer = new DefaultTableCellRenderer();

renderer.setHorizontalAlignment(SwingConstants.CENTER);

//通过列名获取表格的列对象，并将上面设好对齐方式应用到指定列上

table.getColumn("订单编号").setCellRenderer(renderer);

table.getColumn("餐台编号").setCellRenderer(renderer);

table.getColumn("创建日期").setCellRenderer(renderer);

table.getColumn("总金额").setCellRenderer(renderer);

table.getColumn( "客户编号").setCellRenderer(renderer);

table.getColumn("状态").setCellRenderer(renderer);

table.getColumn( "就餐人数").setCellRenderer(renderer);

//设置表格宽度，以及拖动的最大和最小宽度

DefaultTableColumnModel dcm = (DefaultTableColumnModel) table.getColumnModel();

dcm.getColumn(0).setPreferredWidth(60);

dcm.getColumn(0).setMinWidth(35);

dcm.getColumn(0).setMaxWidth(180);

dcm.getColumn(2).setPreferredWidth(100);

dcm.getColumn(0).setPreferredWidth(150);

//设置行高

//table.setRowHeight(25);

try {

String strSql = "select \* from order\_infor";

ResultSet rs = statement.executeQuery(strSql);

while (rs.next()) {

model.addRow(new Object[]{rs.getString("orderId"), rs.getInt("deskID"),

rs.getString("createtime"), rs.getString("money"),rs.getInt("customerId"),

rs.getString("status"), rs.getInt("Numbers")});

}

} catch (Exception ex) {

throw new RuntimeException(ex);

}

//允许点击JTable的表头进行升降排序

table.setAutoCreateRowSorter(true);

//居中显示窗口

jf.setLocationRelativeTo(null);

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

jf.dispose();//关闭此页面

new MenuGui();

}

});

xg.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int row = table.getSelectedRow();

if(row!=-1){

if(((String)table.getValueAt(row,5)).equals("未支付")) {

try {

YIN = (String) table.getValueAt(row, 3);

deskID = (int) table.getValueAt(row, 1);

new CheckOut();

} catch (Exception exception) {

exception.printStackTrace();

}

}

else{

JOptionPane.showMessageDialog(null,

"该账单已支付！！", "提示",

JOptionPane.PLAIN\_MESSAGE);

}

}

}

});

LookFor.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (!nullchar.isBlankSpace( tf\_1.getText())) {

try {

lookForOrder.LookFor(model, tf\_1.getText());

} catch (Exception exception) {

exception.printStackTrace();

}

}

}

});

fh.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

updateOrder.updata(model);

} catch (Exception exception) {

exception.printStackTrace();

}

}

});

jf.add(scrollPane);

jf.add(back);

jf.add(LookFor);

jf.add(fh);

jf.add(name);

jf.add(tf\_1);

jf.add(xg);

jf.setVisible(true);

}

public static DefaultTableModel getModel(){

return model;

}

public static MyTable getTable(){

return table;

}

public static String getYIN(){

return YIN;

}

public static int getDeskID(){

return deskID;

}

public static void main(String[] args) throws Exception {

new CheckoutManner();

}

}

public class CustomerMAnner {

DatabaseUtil databaseUtil=new DatabaseUtil();

JPanel controlPanel;

//定义表格的数据模型

public static DefaultTableModel model;

public static MyTable table;

//定义一个表格

//定义一个滚动面板，用来放置表格

JScrollPane scrollPane;

JButton back = new JButton("返回主菜单");

JButton del = new JButton("删除");

JButton cri = new JButton("添加");

JButton xg = new JButton("修改");

JButton LookFor = new JButton("查询");

JButton fh = new JButton("返回");

JDBC001 jdbc001 = databaseUtil.getJdbc001();

Connection coon = jdbc001.getsConnection();

Statement statement = jdbc001.getsStament();

LookForCustomer lookForCustomer = new LookForCustomer();

Nullchar nullchar = new Nullchar();

UpdateCustomer updateCustomer = new UpdateCustomer();

public CustomerMAnner() throws Exception {

//注册驱动

JFrame jf = new JFrame("客户信息管理");

jf.setContentPane(new JPanel() {

public void paintComponent(Graphics g) {

super.paintComponent(g);

//下面这行是为了背景图片可以跟随窗口自行调整大小，可以自己设置成固定大小

Image background;

try {

background = ImageIO.read(new File("D:\\picture\\a02f3cc968159c646ddf697cbdcfd83d(1).jpeg"));

} catch (IOException e) {

throw new RuntimeException(e);

}

g.drawImage(background, 0, 0, this.getWidth(), this.getHeight(), this);

}

});

jf.setLayout(null);

jf.setBounds(300, 300, 800, 600);

jf.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

jf.setLocationRelativeTo(null);//窗口居中

//表头

String[] head = {"客户编号","客户名称","性别","单位","电话号","贵宾卡号"};

//初始化数据

Object[][] datas = new String[][]{

};

//初始化表格数据模型

model = new DefaultTableModel(datas, head);

//用表格模型初始化表格

table = new MyTable(model);

//初始化滚动面板

scrollPane = new JScrollPane(table);

//初始化控制面板，添加相关按钮

back.setBounds(20, 40, 120, 30);

scrollPane.setBounds(20, 80, 680, 380);

cri.setBounds(150, 480, 80, 30);

del.setBounds(240, 480, 80, 30);

xg.setBounds(330, 480, 80, 30);

LookFor.setBounds(420,40,80,30);

fh.setBounds(520,40,80,30);

JLabel name = new JLabel("请输入客户关键字");

name.setFont(new Font("黑体", Font.PLAIN, 15));

name.setBounds(150,40,135,30);

JTextField tf\_1 = new JTextField();

tf\_1.setBounds(285,40,120,30);

//设置面板布局，上面控制按钮，中间放表格列表

//设置表格中的对齐方式

DefaultTableCellRenderer renderer = new DefaultTableCellRenderer();

renderer.setHorizontalAlignment(SwingConstants.CENTER);

//通过列名获取表格的列对象，并将上面设好对齐方式应用到指定列上

table.getColumn("客户编号").setCellRenderer(renderer);

table.getColumn("客户名称").setCellRenderer(renderer);

table.getColumn("性别").setCellRenderer(renderer);

table.getColumn("单位").setCellRenderer(renderer);

table.getColumn("电话号").setCellRenderer(renderer);

table.getColumn("贵宾卡号").setCellRenderer(renderer);

//设置表格宽度，以及拖动的最大和最小宽度

DefaultTableColumnModel dcm = (DefaultTableColumnModel) table.getColumnModel();

dcm.getColumn(0).setPreferredWidth(60);

dcm.getColumn(0).setMinWidth(45);

dcm.getColumn(0).setMaxWidth(75);

//设置行高

//table.setRowHeight(25);

try {

String strSql = "select \* from customer";

ResultSet rs = statement.executeQuery(strSql);

while (rs.next()) {

model.addRow(new Object[]{rs.getInt("customerId"), rs.getString("name"),

rs.getString("sex"), rs.getString("company"),rs.getString("tel"),

rs.getString("cardID")});

}

} catch (Exception ex) {

throw new RuntimeException(ex);

}

//允许点击JTable的表头进行升降排序

table.setAutoCreateRowSorter(true);

//居中显示窗口

jf.setLocationRelativeTo(null);

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

jf.dispose();//关闭此页面

new MenuGui();

}

});

del.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int row = table.getSelectedRow();

if (row != -1) {

int id = (int) table.getValueAt(row, 0);

try {

PreparedStatement ps = coon.prepareStatement("delete from customer where customerId=?");

ps.setInt(1, id);

int rs = ps.executeUpdate();

model.removeRow(row);

} catch (Exception throwables) {

throwables.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(null,

"没有选中行或者已为空表", "提示",

JOptionPane.PLAIN\_MESSAGE);

}

}

});

cri.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

new AddCustomer();

} catch (Exception exception) {

exception.printStackTrace();

}

}

});

xg.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int row = table.getSelectedRow();

if(row!=-1){

try {

new ModifyCustomer();

} catch (Exception exception) {

exception.printStackTrace();

}

}

else{

JOptionPane.showMessageDialog(null,

"请选择需要修改的客户", "提示",

JOptionPane.PLAIN\_MESSAGE);

}

}

});

LookFor.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (nullchar.isBlankSpace(tf\_1.getText())) {

JOptionPane.showMessageDialog(null,

"输入信息有误", "提示",

JOptionPane.PLAIN\_MESSAGE);

}

else {

String name = tf\_1.getText();

try {

lookForCustomer.LookFor(model,name);

} catch (Exception exception) {

exception.printStackTrace();

}

}

}

});

fh.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

updateCustomer.updata(model);

} catch (Exception exception) {

exception.printStackTrace();

}

}

});

jf.add(scrollPane);

jf.add(back);

jf.add(LookFor);

jf.add(fh);

jf.add(name);

jf.add(tf\_1);

jf.add(cri);

jf.add(del);

jf.add(xg);

jf.setVisible(true);

}

public static DefaultTableModel getModel(){

return model;

}

public static MyTable getTable(){

return table;

}

public static void main(String[] args) throws Exception {

new CustomerMAnner();

}

}

public class employeeManner {  
    DatabaseUtil databaseUtil=new DatabaseUtil();  
    JPanel controlPanel;  
    //定义表格的数据模型  
    public static DefaultTableModel model;  
    public static MyTable table;  
    //定义一个表格  
    //定义一个滚动面板，用来放置表格  
    JScrollPane scrollPane;  
    JButton back = new JButton("返回主菜单");  
    JButton del = new JButton("删除");  
    JButton cri = new JButton("添加");  
    JButton xg = new JButton("修改");  
    JButton LookFor = new JButton("查询");  
    JButton fh = new JButton("返回");  
    JDBC001 jdbc001 = databaseUtil.getJdbc001();  
    Connection coon = jdbc001.getsConnection();  
    Statement statement = jdbc001.getsStament();  
    LookForEmployee lookForEmployee= new LookForEmployee();  
    Nullchar nullchar = new Nullchar();  
    UpdateEmployee updateEmployee = new UpdateEmployee();  
    public employeeManner() throws Exception {  
        //注册驱动  
        JFrame jf = new JFrame("员工信息管理");  
        jf.setContentPane(new JPanel() {  
            public void paintComponent(Graphics g) {  
                super.paintComponent(g);  
  
                //下面这行是为了背景图片可以跟随窗口自行调整大小，可以自己设置成固定大小  
                Image background;  
                try {  
                    background = ImageIO.read(new File("D:\\picture\\a02f3cc968159c646ddf697cbdcfd83d(1).jpeg"));  
                } catch (IOException e) {  
                    throw new RuntimeException(e);  
                }  
                g.drawImage(background, 0, 0, this.getWidth(), this.getHeight(), this);  
            }  
        });  
        jf.setLayout(null);  
        jf.setBounds(300, 300, 800, 600);  
        jf.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);  
        jf.setLocationRelativeTo(null);//窗口居中  
  
        //表头  
        String[] head = {"员工编号", "员工名字", "性别","出生日期", "身份证号", "家庭住址", "电话号码", "职位","是否在职"};  
        //初始化数据  
        Object[][] datas = new String[][]{  
        };  
        //初始化表格数据模型  
        model = new DefaultTableModel(datas, head);  
        //用表格模型初始化表格  
        table = new MyTable(model);  
        //初始化滚动面板  
        scrollPane = new JScrollPane(table);  
  
        //初始化控制面板，添加相关按钮  
        back.setBounds(20, 40, 120, 30);  
        scrollPane.setBounds(20, 80, 720, 380);  
        cri.setBounds(150, 480, 80, 30);  
        del.setBounds(240, 480, 80, 30);  
        xg.setBounds(330, 480, 80, 30);  
        LookFor.setBounds(420,40,80,30);  
        fh.setBounds(520,40,80,30);  
        JLabel name = new JLabel("输入员工关键字：");  
        name.setFont(new Font("黑体", Font.PLAIN, 15));  
        name.setBounds(150,40,130,30);  
        JTextField tf\_1 = new JTextField();  
        tf\_1.setBounds(285,40,120,30);  
        //设置面板布局，上面控制按钮，中间放表格列表  
  
        //设置表格中的对齐方式  
        DefaultTableCellRenderer renderer = new DefaultTableCellRenderer();  
        renderer.setHorizontalAlignment(SwingConstants.CENTER);  
        //通过列名获取表格的列对象，并将上面设好对齐方式应用到指定列上  
        table.getColumn("员工编号").setCellRenderer(renderer);  
        table.getColumn("员工名字").setCellRenderer(renderer);  
        table.getColumn("性别").setCellRenderer(renderer);  
        table.getColumn("出生日期").setCellRenderer(renderer);  
        table.getColumn("身份证号").setCellRenderer(renderer);  
        table.getColumn("家庭住址").setCellRenderer(renderer);  
        table.getColumn("电话号码").setCellRenderer(renderer);  
        table.getColumn("职位").setCellRenderer(renderer);  
        table.getColumn("是否在职").setCellRenderer(renderer);  
        //设置表格宽度，以及拖动的最大和最小宽度  
        DefaultTableColumnModel dcm = (DefaultTableColumnModel) table.getColumnModel();  
        dcm.getColumn(0).setPreferredWidth(60);  
        dcm.getColumn(0).setMinWidth(45);  
        dcm.getColumn(0).setMaxWidth(75);  
        dcm.getColumn(4).setPreferredWidth(120);  
        //设置行高  
        //table.setRowHeight(25);  
        try {  
            String strSql = "select \* from employee";  
            ResultSet rs = statement.executeQuery(strSql);  
            while (rs.next()) {  
                model.addRow(new Object[]{rs.getString("employee\_ID"), rs.getString("employee\_name"),  
                        rs.getString("sex"), rs.getString("brith"),rs.getString("IDcard"),  
                        rs.getString("nativep"), rs.getString("tel"), rs.getString("position"),  
                        rs.getString("freeze")});  
            }  
  
        } catch (Exception ex) {  
            throw new RuntimeException(ex);  
        }  
        //允许点击JTable的表头进行升降排序  
        table.setAutoCreateRowSorter(true);  
        /tBounds(100,100,600,400);  
        //居中显示窗口  
        jf.setLocationRelativeTo(null);  
  
        back.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                jf.dispose();//关闭此页面  
                new MenuGui();  
            }  
        });  
        del.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                int row = table.getSelectedRow();  
                if (row != -1) {  
                    String id = (String) table.getValueAt(row, 0);  
                    int Id = Integer.parseInt(id);  
                    System.out.println("删除员工编号 : "+id);  
                    try {  
                        PreparedStatement ps = coon.prepareStatement("delete from employee where employee\_ID=?");  
                        ps.setInt(1, Id);  
                        int rs = ps.executeUpdate();  
                        model.removeRow(row);  
                    } catch (Exception throwables) {  
                        throwables.printStackTrace();  
                    }  
                } else {  
                    JOptionPane.showMessageDialog(null,  
                            "没有选中行或者已为空表", "提示",  
                            JOptionPane.PLAIN\_MESSAGE);  
                }  
            }  
        });  
        cri.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                try {  
                    new AddEmployee();  
                } catch (Exception exception) {  
                    exception.printStackTrace();  
                }  
            }  
        });  
        xg.addActionListener(new ActionListener() {  
  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                int row = table.getSelectedRow();  
                if(row!=-1){  
                    try {  
                        new ModifyEmployee();  
                    } catch (Exception exception) {  
                        exception.printStackTrace();  
                    }  
                }  
                else{  
                    JOptionPane.showMessageDialog(null,  
                            "请选择需要修改的员工", "提示",  
                            JOptionPane.PLAIN\_MESSAGE);  
                }  
            }  
        });  
        LookFor.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                if (nullchar.isBlankSpace(tf\_1.getText())) {  
                    JOptionPane.showMessageDialog(null,  
                            "输入信息有误", "提示",  
                            JOptionPane.PLAIN\_MESSAGE);  
                }  
                else {  
                    String name  = tf\_1.getText();  
                    try {  
                        lookForEmployee.LookFor(model,name);  
                    } catch (Exception exception) {  
                        exception.printStackTrace();  
                    }  
                }  
            }  
        });  
        fh.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                try {  
                    updateEmployee.updata(model);  
                } catch (Exception exception) {  
                    exception.printStackTrace();  
                }  
            }  
        });  
        jf.add(scrollPane);  
        jf.add(back);  
        jf.add(LookFor);  
        jf.add(fh);  
        jf.add(name);  
        jf.add(tf\_1);  
        jf.add(cri);  
        jf.add(del);  
        jf.add(xg);  
        jf.setVisible(true);  
  
  
    }  
    public static DefaultTableModel getModel(){  
        return model;  
    }  
    public static MyTable getTable(){  
        return table;  
    }  
}