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** | **郑婷玉** |
| **学号** | **B20200902113** |
| **指导教师** | **杨刚 潘怡** |
| **起止日期** | **2023.06.05～2023.06.16** |

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，Idea或Eclipse。

（2）服务器：

MySQL5.6或 SQL Server 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

21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 | 郑婷玉 | 学号 | B20200902113 | 班级 | 21软件01 | |
| 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**：（说明实训各阶段的工作过程，包括需求分析、系统设计、编码实现、系统测试、交付实施等阶段的主要工作。）  需求分析阶段：根据实训任务书的问题思考需求。  系统设计阶段：主要完成对数据库的设计。  编码实现阶段：GUI 编程设计页面，处理数据，完成 GUI 页面和数据库数据交互。  系统测试阶段：使用测试数据进行常规测试和边缘测试，试错，修改漏洞。 | | | | | | |
| **实训成果**：  参见基于C/S架构的餐饮管理系统设计说明书。 | | | | | |
| **实训总结**：（包括心得体会、存在的问题和改进方向。）  在这个课程设计中，我通过实际操作加深了对Java和数据库的理解，并进一步认识到Java和数据库在现实生活中的广泛应用。这次课程设计的目的是巩固我们对数据库和Java基本原理和基础理论的理解，掌握数据库应用系统的设计开发的基本方法，以提高我们的综合运用所学知识的能力。为了使我们的数据库应用系统开发设计更加规范、有序、正确、高效，我们采用了工程化阶段开发设计过程与方法，这包括需求分析、结构设计、逻辑结构设计、物理结构设计、数据库实施，以及数据库系统运行与维护。我们以此为基础设计了一个餐饮管理系统。该系统要求我们不仅灵活掌握数据库知识和技术，还需掌握某种客户端开发工具或语言。通过我的设计，系统的功能基本符合任务要求，能够完成餐饮管理的各个模块，包括员工管理、开台、管理、菜谱设置和点菜服务等。如果在课程设计中遇到了问题，我可以与同学相互交流讨论或上网寻找资料解决问题。总之，本次课程设计帮助我进一步掌握了软件设计的方法，加强了对数据库软件和IDEA的使用，同时学习到了很多理论知识和实践技能。尽管在技术和时间上存在一些不足，比如用户界面不够美观，出错处理不够完善等问题，这些仍需今后进一步的改善。总的来说，通过本次面向对象的课程设计，我对Java高级特性和数据库操作有了更深入的了解，受益匪浅。 | | | | | |

基于C/S架构的餐饮管理系统

设计说明书

作者：郑婷玉（B20200902113）

计算机科学与工程学院

2023年 06 月 15日

**摘要**

**随着我国市场经济的不断发展和国民生活水平的不断提高，越来越多的人前往高档消费场所，如餐厅等。传统的手工点菜方式存在难以计算、查找、更改、容易出错和效率低等缺点，已逐渐被高档消费场所的服务管理平台淘汰。因此，各种类型和形式的餐厅点菜管理系统应运而生，并呈现出多样化的发展趋势。为了更好地满足广大消费者的多元化消费需求和不同消费水平的要求，提高餐厅的服务管理质量和工作人员的工作效率，我们经过多方面的考察、分析和研究现有的酒店点菜管理系统，并致力于开发一套可视化程度高、功能全面、集分析管理于一体的点餐管理系统，旨在提高消费者的满意度、商家的服务水平和市场竞争力，具有极高的市场价值。**

**关键词：**java ，C/S架构，餐饮管理系统

目录

[已知技术参数和设计要求： 2](#_Toc137828875)

[1. 问题描述（功能要求）： 2](#_Toc137828876)

[**2. 运行环境要求：** 2](#_Toc137828877)

[3. 技术要求： 2](#_Toc137828878)

[实训工作量： 3](#_Toc137828879)

[工作计划： 3](#_Toc137828880)

[1 引言 1](#_Toc137828881)

[1.1 编写目的 1](#_Toc137828882)

[1.2 参考资料 1](#_Toc137828883)

[2 需求规约 2](#_Toc137828884)

[2.1 功能需求 2](#_Toc137828885)

[2.1.2 员工信息管理用例说明 3](#_Toc137828886)

[2.1.3 客户信息管理用例说明 4](#_Toc137828887)

[2.1.4 餐台信息管理用例说明 4](#_Toc137828888)

[2.1.5 菜品分类信息管理用例说明 5](#_Toc137828889)

[2.1.6 菜品信息管理用例说明 6](#_Toc137828890)

[2.1.7 点单信息管理用例说明 7](#_Toc137828891)

[2.2 界面需求 7](#_Toc137828892)

[2.3 数据需求 8](#_Toc137828893)

[3 系统设计 13](#_Toc137828894)

[3.1 运行环境 13](#_Toc137828895)

[3.2 系统静态结构设计 13](#_Toc137828896)

[3.3 人机接口设计 14](#_Toc137828897)

[3.4 登录模块设计说明 16](#_Toc137828898)

[3.5员工管理模块设计说明 17](#_Toc137828899)

[3.6 点菜模块设计说明 19](#_Toc137828900)

[4 数据库设计 21](#_Toc137828901)

[4.1 数据库环境说明 21](#_Toc137828902)

[4.2 数据库的命名规则 21](#_Toc137828903)

[4.3 逻辑结构设计 22](#_Toc137828904)

[4.4 物理结构设计 24](#_Toc137828905)

[4.5 安全性设计 31](#_Toc137828906)

[5 测试用例设计 33](#_Toc137828907)

[表5.2餐台管理测试用例 33](#_Toc137828908)

[附录 35](#_Toc137828909)

[附录1 程序运行结果 35](#_Toc137828910)

[附录2 程序源代码 38](#_Toc137828911)

# 1 引言

## 1.1 编写目的

本文档是“基于C/S架构的餐饮管理系统”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] 戴远泉，程宁，胡文杰. Java高级程序设计实战教程[M]. 北京: 人民邮电出版社，2022．

# 2 需求规约

## 2.1 功能需求

系统管理员通过该部分功能完成餐厅点菜管理系统中基础数据的设置工作。

主要工作包括:

（1）.员工管理：对员工实现增删改查。

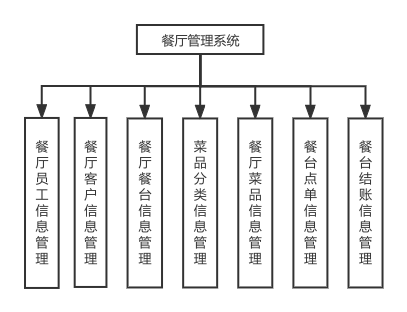
（2）.客户管理：对客户实现增删改查。

（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。



（**图名： 仿宋，小四。**

**图内文字：宋体，五号。）**

2.1.1 用户登录用例说明

用例名称：用户登录。

用例描述：用户可从登录窗口选择身份进行登录，进入主管理页面。如果数据库内没有对应的用户信息，则无法登录。

前置条件：用户请求登录。

后置条件：用户登录成功，进入操作页面。

活动步骤：

1

用户输入账号和密码。

2

系统根据身份检查用户的账号和密码是否输入正确。

3

用户进入操作页面。

扩展点：

1. 无。

异常处理：

1如果账号密码不匹配，则“提示账号或者密码错误”，无法登录。

2.1.2 员工信息管理用例说明

用例名称：员工信息管理

用例描述：增删改查员工信息

前置条件：用户成功登录并打开员工管理界面操作。

后置条件：数据库成功更新（查询）数据。

活动步骤：

修改员工信息:

1. 用户打开修改员工界面，表格显示符合条件的员工数据。
2. 双击某一行数据，并将数据显示在输入框中
3. 填写修改的信息
4. 点击确认按钮提示修改成功。

删除员工信息：

1. 用户打开删除员工界面，表格显示符合条件的员工数据。
2. 双击某一行数据，并将数据显示在输入框中
3. 点击确认按钮系统提示删除成功。

新增员工信息：

1. 填入新增员工的信息。
2. 点击新增按钮提交信息。
3. 系统提示新增成功。

异常处理：

1. 修改的信息不符合设定条件，提示修改失败。
2. 未选择修改信息项点击修改，提示“修改失败”。

2.1.3 客户信息管理用例说明

用例名称：客户信息管理

用例描述：增删改查客户信息

前置条件：用户成功登录并打开客户管理界面操作。

后置条件：数据库成功更新（查询）数据。

活动步骤：

修改客户信息:

1. 用户在主界面点击客户管理，进入客户管理界面
2. 双击某项需要修改的客户信息
3. 填写修改的信息
4. 点击修改按钮

删除客户信息：

1. 用户在主界面点击客户管理，进入客户管理界面
2. 双击某项需要删除的客户信息
3. 填写删除的信息
4. 点击删除按钮

新增客户信息：

1. 用户在主界面点击客户管理，进入客户管理界面
2. 填写新增的信息
3. 点击新增按钮

异常处理：

1. 修改的信息不符合设定条件，提示修改失败。
2. 未选择修改信息项点击修改，提示“修改失败”。

2.1.4 餐台信息管理用例说明

用例名称：餐台信息管理

用例描述：增删改查餐台信息

前置条件：管理员成功登录并打开餐台管理界面操作。

后置条件：数据库成功更新（查询）数据。

活动步骤：

修改餐台信息:

1用户在主界面点击餐台管理，进入餐台管理界面

2双击某项需要修改的客户信息

3填写修改的信息

4点击修改按钮

删除餐台信息：

1用户在主界面点击餐台管理，进入餐台管理界面

2双击某项需要删除的客户信息

3填写删除的信息

4点击删除按钮

新增餐台信息：

1用户在主界面点击餐台管理，进入餐台管理界面

2填写新增的信息

3点击新增按钮

异常处理：

1修改的信息不符合设定条件，提示修改失败。

2未选择修改信息项点击修改，提示“修改失败”。

2.1.5 菜品分类信息管理用例说明

用例名称：菜品分类信息管理

用例描述：增删改查菜品分类信息

前置条件：管理员成功登录并打开菜品分类管理界面操作。

后置条件：数据库成功更新（查询）数据。

活动步骤：

修改菜品分类信息:

1. 管理员点击搜索按钮，表格显示符合条件的菜品分类数据。
2. 双击某项需要修改的菜品分类信息所在行填写修改信息。
3. 点击确认按钮提示是否确认修改。
4. 系统提示提交信息是否正确。
5. 点击确认按钮提示修改成功。

删除菜品分类信息：

1. 管理员点击搜索按钮，表格显示符合条件的菜品分类数据。
2. 双击某项需要删除的菜品分类信息所在行。
3. 点击确认按钮提示是否确认删除。
4. 点击确认按钮系统提示删除成功。

新增菜品分类信息：

1. 填入新增菜品分类的信息。
2. 点击新增按钮提交信息。
3. 系统提示新增成功。

异常处理：

1. 修改的信息不符合设定条件，提示修改失败。
2. 未选择删除信息项点击删除，提示“请先选中行”。
3. 未选择修改信息项点击修改，提示“修改失败”。

2.1.6 菜品信息管理用例说明

用例名称：菜品信息管理

用例描述：增删改查菜品信息

前置条件：管理员成功登录并打开菜品管理界面操作。

后置条件：数据库成功更新（查询）数据。

活动步骤：

修改菜品信息:

1. 管理员点击搜索按钮，表格显示符合条件的菜品数据。
2. 双击某项需要修改的菜品信息所在行填写修改信息。
3. 点击确认按钮提示是否确认修改。
4. 系统提示提交信息是否正确。
5. 点击确认按钮提示修改成功。

删除菜品信息：

1. 管理员点击搜索按钮，表格显示符合条件的菜品数据。
2. 双击某项需要删除的菜品信息所在行。
3. 点击确认按钮提示是否确认删除。
4. 点击确认按钮系统提示删除成功。

新增菜品信息：

1. 填入新增菜品分类的信息。
2. 点击新增按钮提交信息。
3. 系统提示新增成功。

异常处理：

1. 修改的信息不符合设定条件，提示修改失败。
2. 未选择删除信息项点击删除，提示“请先选中行”。
3. 未选择修改信息项点击修改，提示“修改失败”。

2.1.7 点单信息管理用例说明

用例名称：点单信息管理

用例描述：客户下单

前置条件：管理员成功登录并打开点单管理界面根据用户提示操作。

后置条件：数据库成功更新（查询）订单数据。

活动步骤：

1. 填写下单工号，选择点餐桌号，点击开始点餐
2. 选择相应的类别，将其添加到购物车中
3. 输入顾客手机号和实付金额，并按下结账按钮

异常处理：

1. 未选择删除信息项点击删除，提示“请先选中行”。
2. 加入购物车时分数小于等于0时提示“请正确填写下单数量”。

## 2.2 界面需求

用户界面是人与计算机之间的媒介。用户通过用户界面来与计算机进行信息交换。因此，用户界面的质量，直接关系到应用系统的性能能否充分发挥，能否使用户准确、高效、轻松、愉快地工作。所以软件的友好性、易用性对于软件系统至关重要。

1、简易性

界面的简洁是要让用户便于使用、便于理解、并能减少用户发生错误选择的可能性。本系统在按钮设计上尽可能的使用户操作便捷、轻松。

2、一致性

本系统界面的结构清晰且所用的术语要保持一致，风格与内容相一致，界面的色调字体也尽量保持一致。在字体设计上使用了楷体、宋体等字体，字体大小根据界面进行适当调整。界面布局使用了绝对布局，使界面始终处于居中位置，让用户视觉上感觉到界面友好。

3、规范性

通常一个用户界面的元素包括界面主颜色、字体颜色、字体大小、界面布局、界面交互方式、界面功能分布、界面输入输出模式。本系统界面设计符合大众习惯，适用于用户操作。

## 2.3 数据需求

表2.1 用户实体

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 唯一标识 | 说明 |
| id | int | 32 | 是 | 编号 |
| 用户名 | varchar | 20 | 否 | 登录用户名 |
| 密码 | varchar | 20 | 否 | 登录密码 |

表2.2 员工实体

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 唯一标识 | 说明 |
| id | int | 32 | 是 | 编号 |
| 姓名 | varchar | 20 | 否 | 员工姓名 |
| 性别 | enum |  | 否 | 员工性别 |
| 身份证号 | varchar | 18 | 是 | 员工身份证号 |
| 电话号码 | varchar | 11 | 否 | 员工联系电话 |
| 状态 | enum |  | 否 | 员工状态 |

表2.3 客户实体

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 唯一标识 | 说明 |
| id | int | 32 | 是 | 编号 |
| 姓名 | varchar | 20 | 否 | 客户姓名 |
| 性别 | enum |  | 否 | 客户性别 |
| 电话号码 | varchar | 11 | 否 | 客户联系电话 |
| 优惠折扣 | double |  | 否 | 客户优惠折扣 |

表2.4 餐桌实体

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 唯一标识 | 说明 |
| id | int | 32 | 是 | 编号 |
| 代码 | varchar | 8 | 是 | 餐桌代码 |
| 座位数 | int | 32 | 否 | 餐桌座位数 |
| 状态 | enum |  | 否 | 餐桌状态 |

表2.5 菜品分类实体

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 唯一标识 | 说明 |
| id | int | 32 | 是 | 编号 |
| 分类名称 | varchar | 20 | 否 | 菜品分类名称 |
| 菜品描述 | varchar | 255 | 否 | 菜品分类描述 |

表2.6 菜品实体

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 唯一标识 | 说明 |
| id | int | 32 | 是 | 编号 |
| 菜品名 | varchar | 20 | 否 | 菜品名称 |
| 分类编号 | int | 32 | 否 | 菜品分类编号 |
| 价格 | double |  | 否 | 菜品单价 |
| 描述 | varchar | 255 | 否 | 菜品描述 |
| 状态 | enum |  | 否 | 菜品状态 |
| 销量 | int | 32 | 否 | 菜品销量 |

表2.7 订单实体

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 唯一标识 | 说明 |
| orderNo | varchar | 20 | 是 | 订单编号 |
| 餐桌编号 | int | 32 | 否 | 餐桌编号 |
| 创建时间 | datetime |  | 否 | 订单创建时间 |
| 价格 | double |  | 否 | 订单总金额 |
| 客户id | int | 32 | 否 | 订单对应的客户id |
| 状态 | enum |  | 否 | 订单状态 |
| 就餐人数 | int | 32 | 否 | 订单对应就餐人数 |

表2.8订单明细实体

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 唯一标识 | 说明 |
| id | int | 32 | 是 | 编号 |
| 订单编号 | varchar | 20 | 否 | 所属订单编号 |
| 菜品编号 | int | 32 | 否 | 菜品id |
| 数量 | int | 32 | 否 | 该菜品下单数量 |

# 3 系统设计

采用C/S架构，视图层负责界面的展示和鼠标响应函数的实现，DAO层用于功能函数的实现，实体层负责增删改时，作为修改对象传入，方便增删改功能。

## 运行环境

1. 硬件环境的约束:

CPU: INTEL 及兼容1. 8GHZ以上

内存: 512M以上

硬盘空间:500M以上的硬盘空间

1. 软件环境的约束:

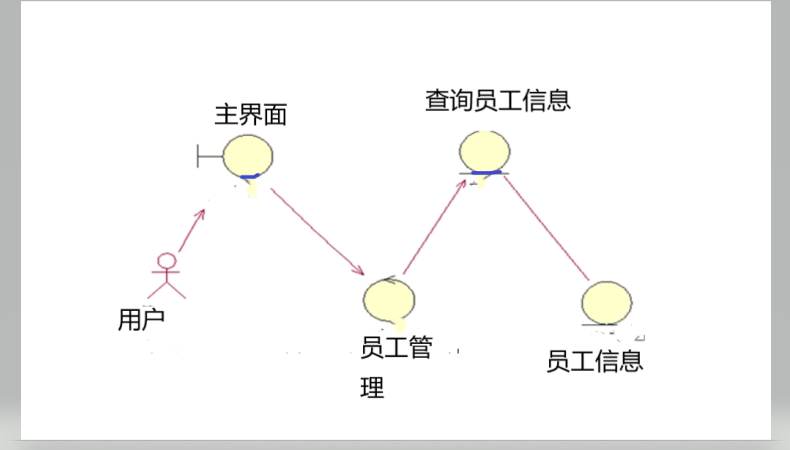
操作系统：Windows XP/Windows 7-11

## 系统静态结构设计

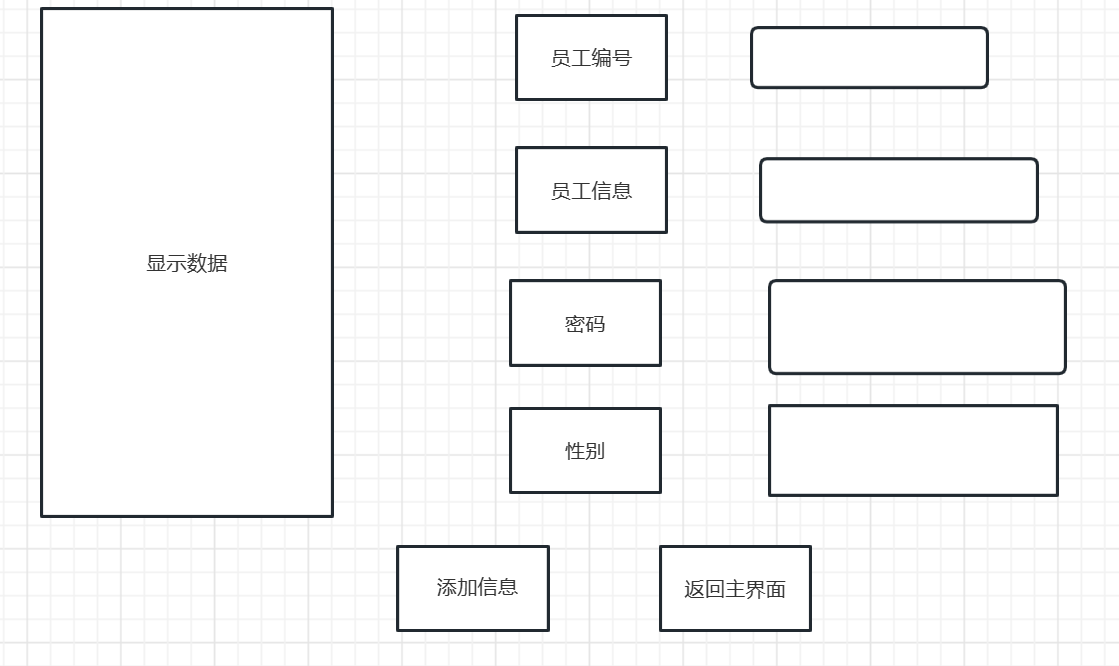
DAO层：accountDAO、customerDAO、cateDAO、dishDAO、employeeDAO、itemDAO、orderDAO、tableDAO  
  
entity层：Cate、Customer、DishEmployee、ordering、Orderitem、Table、User  
  
Frame层：account、customer、test、chart、dish、dishcate、order、addem、delem、findallem、selectem、udem、table、LoginPanel、MainFrame

## 3.3 人机接口设计

3.3.1 设计应用类

图3.3 员工信息查询

3.3.2 用户接口1（具体标识符）设计

图3.2 新增员工信息界面布局设计

3.3.3 用户接口2（具体标识符）设计

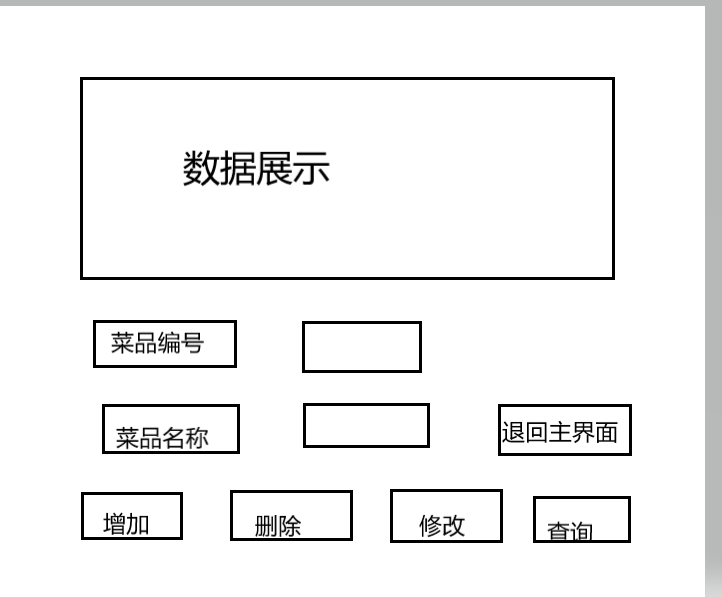


图3.2 菜品管理界面布局设计

## 3.4 登录模块设计说明

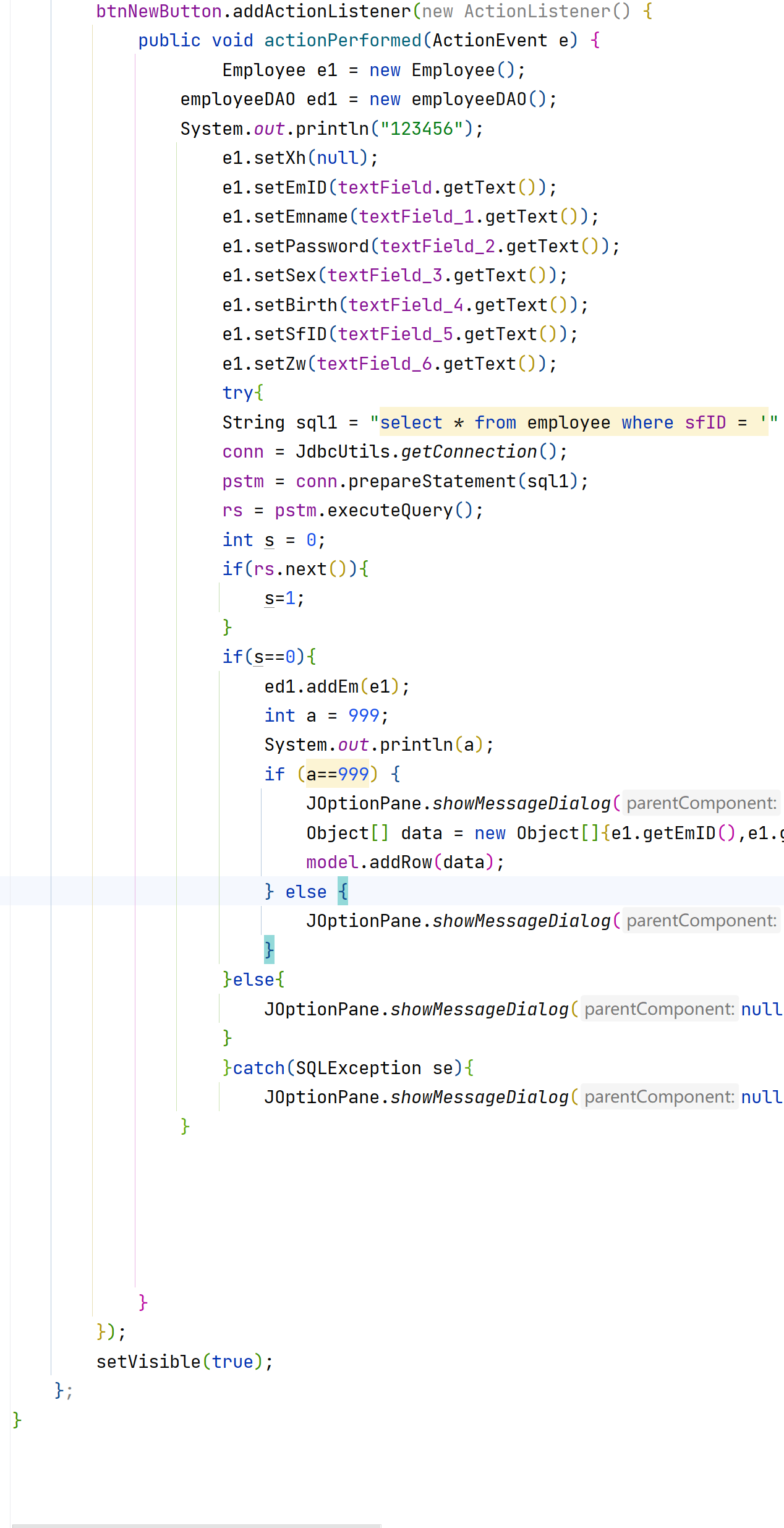
当系统启动时，将执行LoginPanel.java 中的main方法，实例化一个界面用于获得用户输入的账号密码，用户点击登录按钮后将信息传递给监听函数，监听函数内对user表执行查询功能，用于验证用户输入的登录信息是否正确，并给出相应的消息提醒。



## 3.5员工管理模块设计说明

员工关系模块分为5个界面分别是：查询所有员工的信息，增加员工信息，删除员工信息、修改员工信息以及查询特定id或者姓名等员工的信息。

点击相应员工管理模块会弹出相应的界面，用户填写相应数据后点击按钮会执行界面类的鼠标响应函数，会对员工表进行增删改查操作并将相应的数据展现在界面中。



## 3.6 点菜模块设计说明

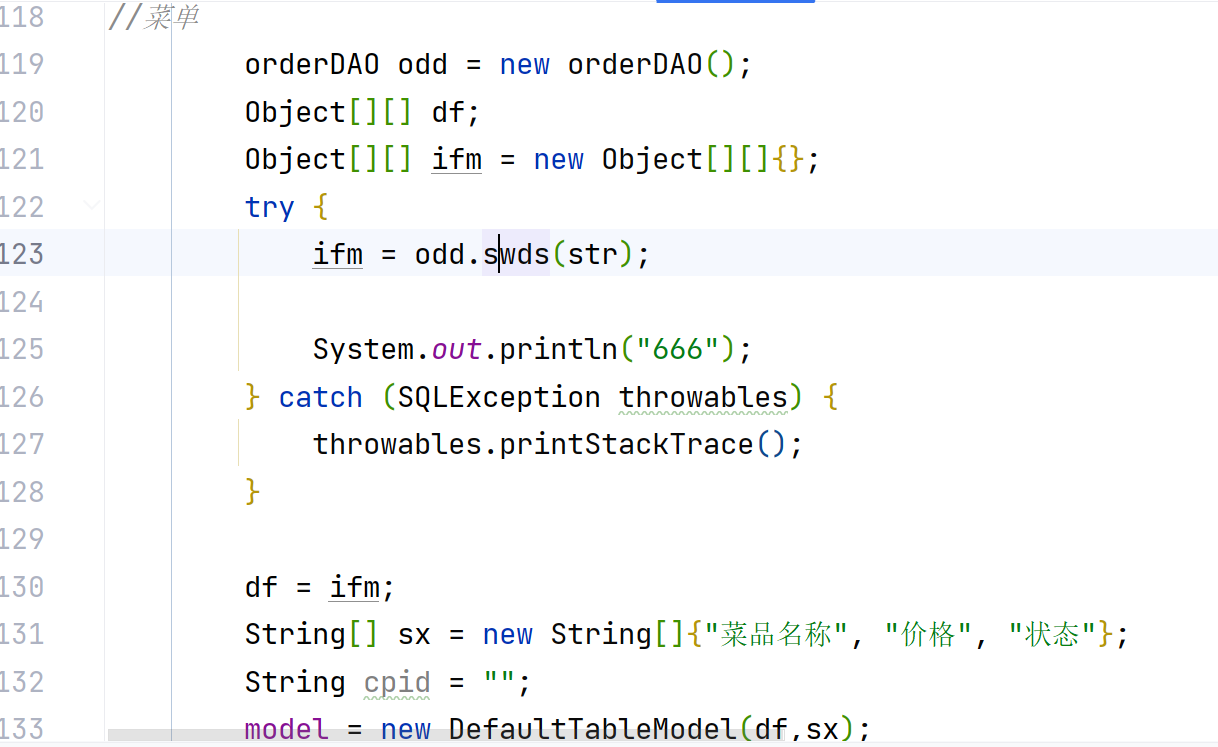
界面层：在主界面点击点菜管理进入点菜模块，选择点餐桌号以及下单工号并点击开始点餐进入点餐。

选择菜品种类，添加或者删除菜品到购物车，输入客户电话和实付金额，点击结账，弹出账单界面。

DAO层：实现对dish表的查询操作并封装成函数返回二维数组给界面层展现数据。

实体层：用做增删改查的参数，方便增删改查。

界面层菜单显示：



DAO层：查询功能的实现，并将查询结果以二维数组的形式返回。

D

# 4 数据库设计

## 4.1 数据库环境说明

本餐饮管理系统采用Java语言进行开发，JDK采用17，开发工具使用IDEA，数据库使用 MySQL5.6。

## 4.2 数据库的命名规则

一、数据库命名规范  
       可以采用26个英文字母 (区分大小写) 和0-9的自然数 (一般不需要) 加上下划线 ‘\_’ 组成，命名简介明确 (Student\_Union)，多个单词用下划线 ‘\_’ 分隔，一个项目一个数据库，多个项目慎用同一个数据库  
  
 二、表命名规范  
 1.采用26字母和0-9的自然数加上下互相 ‘\_’ 组成，命名简洁明确，多个单词用下划线 ‘\_’ 隔开  
 2.全部小写命名，尽量避免出现大写  
 3.禁止使用关键字，如：name，time，select 等等  
 4.表名称不要取得太长  
 5.表的名称一般使用名词或者动宾短语  
 6.也要注意单词形式，列如：使用 user ，而不是 users  
 7.表必须填写描述信息  
  
 三、字段命名规范  
 1.采用26字母和0-9的自然数加上下互相 ‘\_’ 组成，命名简洁明确，多个单词用下划线 ‘\_’ 隔开  
 2.全部小写命名，尽量避免出现大写  
 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.采用字段的名字必须是易于理解，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（如：在 user 表中，出现 user\_name 字段）  
 8.字段命名使用完整名称  
  
 四、SQL语言编码规范  
 1.所有关键字最好大写，如：DELETE、UPDATE、INSERT等等  
 2.多有函数及其参数中除用户变量以外的部分大写  
 3.在定义变量时用到的数据类型小写

## 4.3 逻辑结构设计

E-R图

关系模型

红色为主键，蓝色为外键

管理员（编号，用户名，密码）

员工（编号，姓名， 性别，身份证号， 状态，手机号）

客户（编号， 姓名， 性别， 手机号）

菜品分类（编号，名称， 描述）

菜品（编号，名称， 价格，描述，状态，销量）

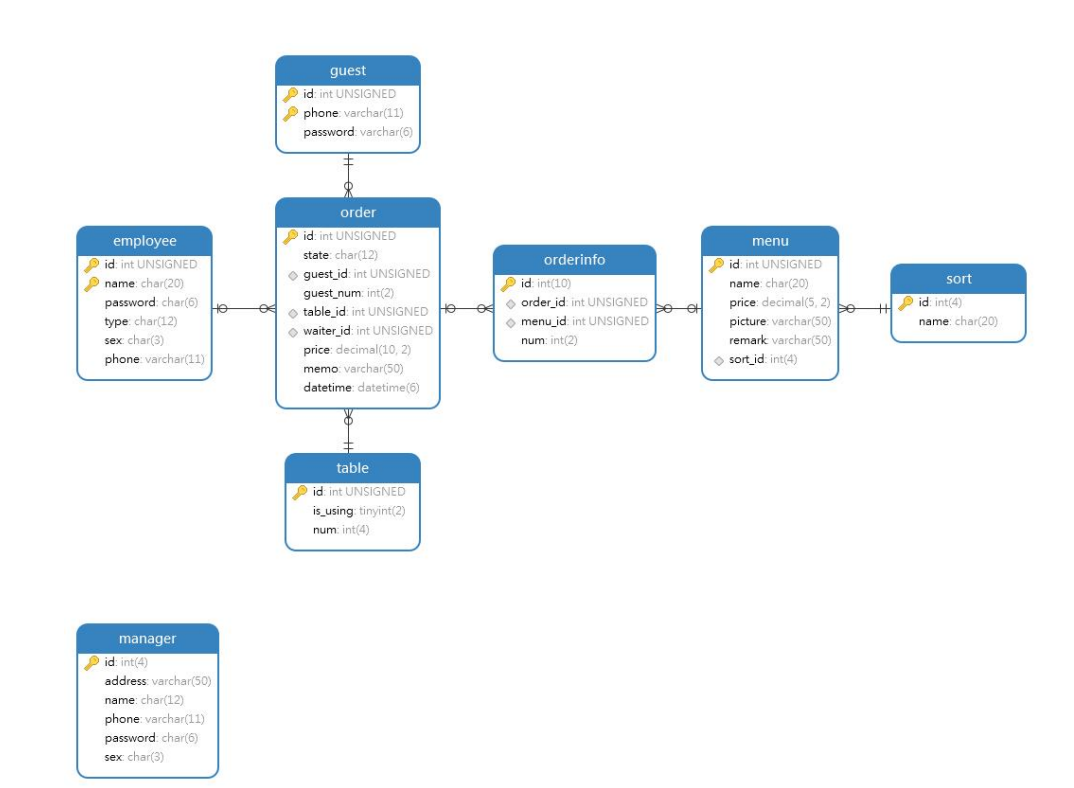
订单（订单编号， 创建时间，就餐人数，消费金额，客户id，订单状态，餐台编号）

订单明细（编号，订单编号，菜品编号，数量）

范式分析

由关系模型可得不含部分依赖和传递依赖，满足3NF。

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 |
| user | 管理管理员账号密码信息 |
| employee | 存储员工信息 |
| customer | 存储客户信息 |
| category | 存储菜品分类信息 |
| Dish | 存储菜品信息 |
| orderding | 存储订单信息 |
| orderItem | 存储订单明细信息 |
| desk | 存储餐台信息 |

表4.2 user表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | user | | | | | |
| 主 键 | | id | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | | id | | | | | |
| 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 | id | int(11) | N | Y |  |  | 唯一约束 |
| 2 | username | varchar(20) | N | Y |  |  | 唯一约束 |
| 3 | password | varchar(20) | N | N |  |  |  |
| sql脚本 | | CREATE TABLE user (  id int AUTO\_INCREMENT COMMENT '序号',  username varchar(20) NOT NULL UNIQUE COMMENT '用户名',  password varchar(20) NOT NULL COMMENT '密码',  PRIMARY KEY (id) ); | | | | | |
| 备注 | |  | | | | | |

表4.3 employee表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | employee | | | | | |
| 主 键 | | id | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | | id | | | | | |
| 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 | id | int(11) | N | Y |  |  | 唯一约束 |
| 2 | name | varchar(20) | N | N |  |  | 唯一约束 |
| 3 | sex | enum | N | N |  |  | (‘男’, ‘女’) |
| 4 | identityID | varchar(18) | N | Y |  |  |  |
| 5 | tel | varchar(11) | Y | Y |  |  |  |
| sql脚本 | | CREATE TABLE employee (  id int AUTO\_INCREMENT COMMENT '序号',  name varchar(20) NOT NULL COMMENT '姓名',  sex Enum ('男', '女') COMMENT '性别',  identityID varchar(18) NOT NULL COMMENT '身份证号',  tel varchar(11) COMMENT '电话号码',  status Enum ('在职', '离职') COMMENT '状态',  PRIMARY KEY (id) ); | | | | | |
| 备注 | |  | | | | | |

表4.4 customer表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | customer | | | | | |
| 主 键 | | id | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | | id | | | | | |
| 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 | id | int(11) | N | Y |  |  | 唯一约束 |
| 2 | name | varchar(20) | N | N |  |  | 唯一约束 |
| 3 | sex | enum | N | N |  |  | (‘男’, ‘女’) |
| 4 | tel | varchar(11) | Y | Y |  |  |  |
| 5 | base | double | N | N |  |  |  |
| sql脚本 | | CREATE TABLE customer (  id int AUTO\_INCREMENT COMMENT '序号',  name varchar(20) NOT NULL COMMENT '姓名',  sex varchar(2) COMMENT '性别',  tel varchar(11) COMMENT '电话号码',  base double COMMENT '折扣率',  PRIMARY KEY (id) ); | | | | | |
| 备注 | |  | | | | | |

表4.5 category表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | category | | | | | |
| 主 键 | | id | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | | id | | | | | |
| 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 | id | int(11) | N | Y |  |  | 唯一约束 |
| 2 | name | varchar(20) | Y | N |  |  |  |
| 3 | describe | varchar(255) | Y | N |  |  |  |
| sql脚本 | | CREATE TABLE category (  id int AUTO\_INCREMENT COMMENT '序号',  name varchar(20) COMMENT '菜分类名',  `describe` varchar(255) COMMENT '描述',  PRIMARY KEY (id) ); | | | | | |
| 备注 | |  | | | | | |

表4.6 desk表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | desk | | | | | |
| 主 键 | | id | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | | id | | | | | |
| 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 | id | int(11) | N | Y |  |  | 唯一约束 |
| 2 | seating | int | N | N |  |  |  |
| 3 | status | enum | N | N |  |  | Enum ('忙碌', '空闲', '预订') |
| sql脚本 | | CREATE TABLE desk (  id int AUTO\_INCREMENT COMMENT '序号',  seating int NOT NULL COMMENT '座位数',  status Enum ('忙碌', '空闲', '预订') COMMENT '状态：忙碌，空闲, 预订',  PRIMARY KEY (id) ); | | | | | |
| 备注 | |  | | | | | |

表4.7 order表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | order | | | | | |
| 主 键 | | orderNo | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | | orderNo | | | | | |
| 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 | varchar(20) | N | Y |  |  |  |
| 2 | deskId | int | N | N |  |  |  |
| 3 | createTime | datetime | N | N |  |  |  |
| 4 | money | double | N | N |  |  |  |
| 5 | customerId | int | N | N |  |  |  |
| 6 | status | enum | N | N |  |  | Enum ('已支付', '未支付') |
| 7 | number | int | N | N |  |  |  |
| sql脚本 | | CREATE TABLE `order` (  orderNo varchar(20) COMMENT '订单编号（当期日期yyyymmdd+8位随机数）',  deskId int NOT NULL COMMENT '餐台号',  createTime datetime NOT NULL COMMENT '就餐日期时间',  money double NOT NULL COMMENT '金额',  customerId int NOT NULL COMMENT '客户id',  status Enum ('已支付', '未支付') COMMENT '状态:已支付，未支付',  number int NOT NULL COMMENT '就餐人数',  PRIMARY KEY (orderNo) ); | | | | | |
| 备注 | |  | | | | | |

表4.8 orderItem表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | orderItem | | | | | |
| 主 键 | | id | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | | id | | | | | |
| 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 | id | int(11) | N | Y |  |  | 唯一约束 |
| 2 | orderId | int | N | N |  |  |  |
| 3 | foodId | enum | N | N |  |  |  |
| 4 | cnt | int | N | N |  |  |  |
| sql脚本 | | CREATE TABLE orderItem (  id int AUTO\_INCREMENT COMMENT '序号',  orderId varchar(20) NOT NULL COMMENT '订单编号',  foodId int NOT NULL COMMENT '菜品编号',  cnt int NOT NULL COMMENT '菜品数量',  PRIMARY KEY (id) ); | | | | | |

表4.9 food表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | food | | | | | |
| 主 键 | | id | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | | id | | | | | |
| 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 | id | int(11) | N | Y |  |  | 唯一约束 |
| 2 | name | varchar(20) | N | N |  |  |  |
| 3 | categoryId | int | Y | N |  |  |  |
| 4 | price | double | N | N |  |  |  |
| 5 | describe | varchar(255) | Y | N |  |  |  |
| 6 | status | enum | N | N |  |  | Enum ('已下架', '上架中') |
| 7 | sum | double | N | N |  |  |  |
| 8 | filePath | varchar(255) | Y | N |  |  |  |
| sql脚本 | | CREATE TABLE food (  id int AUTO\_INCREMENT COMMENT '菜品序号',  name varchar(20) NOT NULL COMMENT '菜名',  categoryId int COMMENT '菜品类别编号',  price double NOT NULL COMMENT '价格',  `describe` varchar(255) COMMENT '描述',  status Enum ('已下架', '上架中') COMMENT '状态：已下架，上架中',  sum int NOT NULL DEFAULT 0 COMMENT '菜品销量',  filePath varchar(255) COMMENT '菜品路径',  PRIMARY KEY (id) ); | | | | | |
| 备注 | |  | | | | | |

## 4.5 安全性设计

对于数据库的安全性设计，我们可以从以下几个方面入手：

区分不同的访问者：为了保证数据库的安全性，我们需要区分不同的访问者，并为他们分配不同的权限。对于一些敏感数据和操作，只有授权的用户才能够访问和执行。同时，我们还需定期审查和更新授权策略，以及删除已失效的账号。

区分不同的访问类型：数据库中通常包含读取、修改和删除等多种访问类型。我们要根据不同的访问类型制定不同的访问策略，并限制对敏感数据的修改和删除。同时，还需要记录每个访问者的访问日志，以便出现故障时进行排查和恢复。

区分不同的数据对象：在数据库中，不同的数据对象具有不同的敏感程度。我们需要将不同的数据对象分为不同的安全等级，并采取不同的安全控制措施来保护它们。例如，对于高等级的数据对象，我们需要采取更加严格的措施，如加密存储、禁止外部访问等。

通过以上的措施，我们能够有效地提高数据库的安全性，从而保护软件系统中的敏感数据和操作。同时，在设计之初我们也应该考虑到这些问题，避免在后期进行大规模的修改和维护。

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

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。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。

4.5.1 角色与权限

表4.3 角色与权限

|  |  |  |
| --- | --- | --- |
| 角色 | 可以访问的表与列 | 操作权限 |
| 用户 | Employee | 增删改查 |
| Customer | 增删改查 |
| Desk | 增删改查 |
|  | Dish | 增删改查 |
| Ordering | 增删改查 |
| Orderitem | 增删改查 |

# 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’ | test，test | 登录成功 | 通过 |
| yl-002 | test，1234567 | 登录失败 | 通过 |
| yl-003 | test1，test | 登录失败 | 通过 |
| yl-004 | test,空 | 登录失败 | 通过 |
| yl-005 | 空,test | 登录失败 | 通过 |
| yl-006 | test,一个空格 | 登录失败 | 通过 |
| yl-007 | 一个空格，test | 登录失败 | 通过 |
| yl-008 | test，密码长度超过32 | 登录失败 | 通过 |

### 表5.2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 |
| 餐台查询 | yl-001 | 餐台号 | 1 | 查询成功 | 通过 |
| yl-002 | Null | 查询失败 | 通过 |
| 餐台修改 | yl-004 | 餐台号，状态座位数 | 1，1，6 | 修改成功 | 通过 |
| yl-005 | 1，0,12 | 修改成功 | 通过 |
| yl-003 | Null,1,6 | 修改失败 | 通过 |
| 餐台删除 | yl-006 | 餐台号 | 1 | 删除成功 | 通过 |
| yl-007 | null | 删除失败 | 通过 |
| 餐台添加 | yl-008 | 状态，座位数 | 1，6 | 添加成功 | 通过 |
| yl-009 | null,6 | 添加成功 | 通过 |
| yl-010 | 0，null | 添加失败 | 通过 |

#### 表5.3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 |
| 员工查询 | yl-001 | 员工号 | 1 | 查询成功 | 通过 |
| yl-002 | Null | 查询失败 | 通过 |
| 员工修改 | yl-003 | 姓名， 密码，职位，性别， 电话号  码 | 小明，111111，服务员，男，  11111111111 | 修改成功 | 通过 |
| yl-004 | 小明，null，服务员，男，  11111111111 | 修改失败 | 通过 |
| yl-005 | 小明，111111，null，男，  11111111111 | 修改失败 | 通过 |
| yl-006 | 小明，111111，服务员，null，  11111111111 | 修改失败 | 通过 |
| yl-006 | 小明，111111，服务员，男， null | 修改失败 | 通过 |
| yl-007 | Null，111111，服务员，男，  11111111111 | 修改失败 | 通过 |
| 员工删除 | yl-008 | 员工号 | 1 | 删除成功 | 通过 |
| yl-009 | null | 删除失败 | 通过 |
| 员工添加 | yl-010 | 姓名，密码，职位，性别，  电话号码 | 小明，111111，服务员，男，  11111111111 | 添加成功 | 通过 |
|  |  |  |  |  |

# 附录

## 附录1 程序运行结果

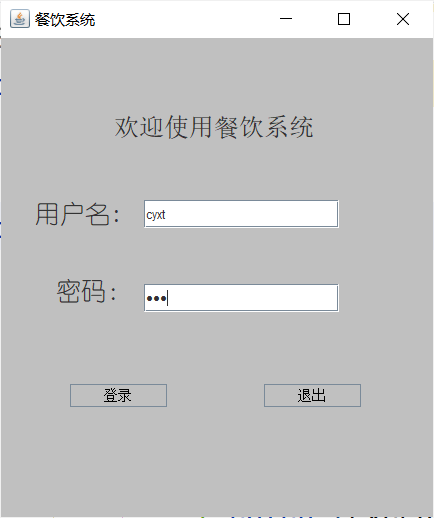


图1登录界面



图二主界面



图三点单和结账界面

图四餐台管理界面



## 附录2 程序源代码

**主界面代码**

package Frame;

import Frame.customer.customer;

import Frame.dish.chart;

import Frame.dish.dish;

import Frame.dish.dishcate;

import Frame.order.order;

import Frame.staff.\*;

import Frame.table.table;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.text.DateFormat;

import java.util.Date;

public class MainFrame extends JFrame {

public MainFrame(){

\*//\*\*主界面框架

\\* setLayout(new FlowLayout());

setTitle("餐饮系统主页");

setBounds(400,90,1000,800);

setDefaultCloseOperation(\*EXIT\_ON\_CLOSE\*);

setBackground(Color.\*orange\*);

getContentPane().setBackground(Color.\*orange\*);

JMenuBar mbar = new JMenuBar();

this.setJMenuBar(mbar);

JMenu m1 = new JMenu("员工管理");

JMenu m2 = new JMenu("餐台管理");

m2.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

dispose();

new table();

}

});

JMenu m3 = new JMenu("菜品管理");

JMenuItem m31 = new JMenuItem("菜品管理");

m31.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

dispose();

new dish();

}

});

m31.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

JMenuItem m32 = new JMenuItem("菜品分类管理");

m32.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

m32.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

dispose();

new dishcate();

}

});

JMenuItem m33 = new JMenuItem("菜品销售统计");

m33.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

m33.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

dispose();

new chart();

}

});

m3.add(m31);

m3.add(m32);

m3.add(m33);

JMenu m4 = new JMenu("点菜管理");

m4.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

dispose();

new order();

}

});

JMenu m6 = new JMenu("客户管理");

m6.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

dispose();

new customer();

}

});

JMenu m7 = new JMenu("帮助");

JMenuItem m71 = new JMenuItem("退出登录");

m7.add(m71);

m71.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

dispose();

}

});

m1.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

m2.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

m3.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

m4.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

m6.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

m7.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

mbar.add(m1);

mbar.add(m2);

mbar.add(m3);

mbar.add(m4);

mbar.add(m6);

mbar.add(m7);

\*//\*\*下拉菜单\*\*(\*\*员工管理\*\*)

\\* JMenuItem m11 = new JMenuItem("查看所有员工信息");

m11.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

m1.add(m11);

m11.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

new findallem();

dispose();

}

});

​ JMenuItem m12 = new JMenuItem("增加员工信息");

​ m12.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

​ m1.add(m12);

​ m12.addActionListener(new ActionListener() {

​ public void actionPerformed(ActionEvent e) {

​ dispose();

​ new addem();

​ }

​ });

​ JMenuItem m13 = new JMenuItem("删除员工信息");

​ m13.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

​ m1.add(m13);

​ m13.addActionListener(new ActionListener() {

​ public void actionPerformed(ActionEvent e) {

​ dispose();

​ new delem();

​ }

​ });

​ JMenuItem m14 = new JMenuItem("更改员工信息");

​ m14.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

​ m1.add(m14);

​ m14.addActionListener(new ActionListener() {

​ public void actionPerformed(ActionEvent e) {

​ dispose();

​ new udem();

​ }

​ });

​ JMenuItem m15 = new JMenuItem("查询员工信息");

​ m15.setFont(new Font("Microsoft YaHei UI", Font.\*PLAIN\*, 18));

​ m1.add(m15);

​ m15.addActionListener(new ActionListener() {

​ public void actionPerformed(ActionEvent e) {

​ dispose();

​ new selectem();

​ }

​ });

​ \*//\*\*问候语

\\* JLabel wh = new JLabel("欢迎回来！" +

​ "天气热多喝水！");

​ wh.setFont(new Font("宋体",1,30));

​ wh.setForeground(Color.\*red\*);

​ wh.setBounds(100,100,400,500);

​ JLabel bs = new JLabel("现在是北京时间：");

​ bs.setFont(new Font("宋体",1,30));

​ bs.setForeground(Color.\*BLUE\*);

​ add(wh,BorderLayout.\*CENTER\*);

​ add(bs,BorderLayout.\*CENTER\*);

​

​ setVisible(true);

​ }

​

​ public static void main(String[] args) {

​ new MainFrame();

​ }

}

**数据库连接代码**

package util;

import java.io.InputStream;

import java.sql.\*;

import java.util.Properties;

\*//\*\*连接工具类

\*public class JdbcUtils {

\*//\*\*获取连接

\\* public static Connection getConnection() {

Connection conn = null;

try {

Class.\*forName\*("com.mysql.cj.jdbc.Driver");

String url = "jdbc:mysql://localhost:3306/cyxt?&useSSL=false&serverTimezone=UTC";

String username = "root";

String password = "123456";

conn = DriverManager.\*getConnection\*(url,username,password);

} catch (Exception e) {

e.printStackTrace();

}

return conn;

}

​ \*//\*\*关闭连接

\\* public static void closeResource(Connection conn, PreparedStatement st)throws SQLException{

​ st.close();

​ conn.close();

​ }

​ public static void closeResource(Connection conn, ResultSet rs,PreparedStatement st)throws SQLException{

​ st.close();

​ rs.close();

​ conn.close();

​ }

​

}

**点菜和结账界面代码**

**package Frame.order;**

**import DAO.dish.cateDAO;**

**import DAO.dish.dishDAO;**

**import DAO.ordering.itemDAO;**

**import DAO.ordering.orderDAO;**

**import DAO.table.tableDAO;**

**import Frame.MainFrame;**

**import Frame.account.account;**

**import entity.Ordering;**

**import entity.Orderitem;**

**import util.CheckUtils;**

**import java.awt.BorderLayout;**

**import java.awt.EventQueue;**

**import javax.swing.\*;**

**import javax.swing.border.EmptyBorder;**

**import java.awt.Color;**

**import java.awt.Dimension;**

**import java.awt.FlowLayout;**

**import java.awt.SystemColor;**

**import java.awt.GridBagLayout;**

**import java.awt.GridBagConstraints;**

**import javax.swing.border.LineBorder;**

**import javax.swing.table.DefaultTableModel;**

**import java.awt.event.ActionListener;**

**import java.awt.event.ActionEvent;**

**import java.awt.Font;**

**import java.awt.Component;**

**import java.awt.event.MouseAdapter;**

**import java.awt.event.MouseEvent;**

**import java.math.BigDecimal;**

**import java.sql.SQLException;**

**public class order extends JFrame {**

**​ private JPanel contentPane;**

**​ private JTable table;**

**​ private JPanel contentPane\_1;**

**​ private JTable table1;**

**​ private String[] cpfl;**

**​ private String[] cth;**

**​ private orderDAO ordd;**

**​ private cateDAO ctd;**

**​ private orderDAO odd;**

**​ private tableDAO tbd;**

**​ private DefaultTableModel model;**

**​ private DefaultTableModel model1;**

**​ private JComboBox comboBox;**

**​ private JComboBox comboBox\_1;**

**​ private String cbb ;**

**​ private JTextField txtgh;**

**​ private JTextField txttel;**

**​ private JTextField textField;**

**​ private int selectedRow;**

**​ private int selectedRow1;**

**​ private String cp;**

**​ private String cp1;**

**​ private itemDAO itd;**

**​ private Orderitem ordit;**

**​ private dishDAO dsd;**

**​ private JLabel lbmn;**

**​ private Ordering ordering;**

**​ private orderDAO odd1;**

**​ public String ddh;**

**​ public account account;**

**\*/\*\***

**\\* \*\* Launch the application.**

**\\* \*\*/**

**\\* public static void main(String[] args) {**

**EventQueue.\*invokeLater\*(new Runnable() {**

**public void run() {**

**try {**

**order frame = new order();**

**frame.setVisible(true);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**});**

**}**

**​ \*/\*\***

**\\* \*\* Create the frame.**

**\\* \*\*/**

**\\* public order() {**

**​ setTitle("\u70B9\u83DC\u754C\u9762");**

**​ setDefaultCloseOperation(JFrame.\*EXIT\_ON\_CLOSE\*);**

**​ setBounds(450, 100, 948, 684);**

**​ contentPane = new JPanel();**

**​ contentPane.setBackground(SystemColor.\*info\*);**

**​ setContentPane(contentPane);**

**​ contentPane.setLayout(null);**

**​ \*//\*\*菜品选择框**

**\\* JScrollPane scrollPane = new JScrollPane(table);**

**​ scrollPane.setBounds(23, 55, 566, 263);**

**​ contentPane.add(scrollPane);**

**​ \*//\*\*下拉种类选项**

**\\* ctd = new cateDAO();**

**​ int n = 0;**

**​ try {**

**​ cpfl = ctd.showfl();**

**​ n = cpfl.length+1;**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ String str = cpfl[0];**

**​ System.\*out\*.println(str);**

**\*//\*\*菜单**

**\\* orderDAO odd = new orderDAO();**

**Object[][] df;**

**Object[][] ifm = new Object[][]{};**

**try {**

**ifm = odd.swds(str);**

**​ System.\*out\*.println("666");**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ df = ifm;**

**​ String[] sx = new String[]{"菜品名称", "价格", "状态"};**

**​ String cpid = "";**

**​ model = new DefaultTableModel(df,sx);**

**​ table = new JTable(model);**

**​ table.setFont(new Font("华光综艺\_CNKI", Font.\*PLAIN\*, 18));**

**​ table.setRowHeight(25);**

**​ comboBox\_1 = new JComboBox();**

**​ for(String i: cpfl){**

**​ comboBox\_1.addItem(i);**

**​ }**

**​ comboBox\_1.setBounds(144, 22, 141, 23);**

**​ contentPane.add(comboBox\_1);**

**​ \*//\*\*下拉框监听 重新加载表格**

**\\* comboBox\_1.addActionListener(new ActionListener() {**

**​ public void actionPerformed(ActionEvent e) {**

**​ String strr = comboBox\_1.getEditor().getItem().toString();**

**​ retable(strr);**

**​ }**

**​ });**

**​ scrollPane.setViewportView(table);**

**​ table.setAutoResizeMode(JTable.\*AUTO\_RESIZE\_ALL\_COLUMNS\*);**

**​ \*//\*\*表格监听 获取选中行的菜品名称 用于加入购物车**

**\\* table.addMouseListener(new MouseAdapter() {**

**​ @Override**

**​ public void mouseClicked(MouseEvent e) {**

**​ selectedRow = table.getSelectedRow();**

**​ String cp = (String) table.getValueAt(selectedRow,0);**

**​ System.\*out\*.println(cp);**

**​ }**

**​ });**

**\*//\*\*桌号下拉显示**

**\\* tbd = new tableDAO();**

**itd = new itemDAO();**

**comboBox = new JComboBox();**

**try {**

**cth = tbd.showfl();**

**} catch (SQLException throwables) {**

**throwables.printStackTrace();**

**}**

**for(String i: cth){**

**comboBox.addItem(i);**

**}**

**comboBox.setBounds(620, 75, 117, 36);**

**contentPane.add(comboBox);**

**comboBox.addActionListener(new ActionListener() {**

**\*//\*\*桌号下拉框监听\* \*:\*\*获取信息 重新加载购物车**

**\\* public void actionPerformed(ActionEvent e) {**

**String strr = comboBox.getEditor().getItem().toString();**

**try {**

**retablee(strr);**

**lbmn.setText(itd.cm(strr));**

**} catch (SQLException throwables) {**

**throwables.printStackTrace();**

**}**

**}**

**});**

**\*//\*\*购物车**

**\\* JScrollPane scrollPane\_1 = new JScrollPane(table1);**

**scrollPane\_1.setBounds(23, 350, 566, 287);**

**contentPane.add(scrollPane\_1);**

**itd = new itemDAO();**

**Object[][] df1;**

**Object[][] ifm1 = new Object[][]{};**

**try {**

**\*//\* \*从数据库获取购物车信息**

**\\* ifm1 = itd.swgwc(comboBox.getEditor().getItem().toString());**

**} catch (SQLException throwables) {**

**throwables.printStackTrace();**

**}**

**​ df1 = ifm1;**

**​ String[] sx1 = new String[]{"菜品名称","份数"};**

**​ String cpid1 = "";**

**​ model1 = new DefaultTableModel(df1,sx1);**

**​ table1 = new JTable(model1);**

**​ table1.setFont(new Font("华光综艺\_CNKI", Font.\*PLAIN\*, 18));**

**​ table1.setRowHeight(25);**

**​ \*//\*\*表格监听 获取选中行的菜品名称 用于删除**

**\\* table1.addMouseListener(new MouseAdapter() {**

**​ @Override**

**​ public void mouseClicked(MouseEvent e) {**

**​ selectedRow1 = table1.getSelectedRow();**

**​ String cp1 = (String) table1.getValueAt(selectedRow1,0);**

**​ System.\*out\*.println(cp1);**

**​ }**

**​ });**

**​ scrollPane\_1.setViewportView(table1);**

**​ table1.setAutoResizeMode(JTable.\*AUTO\_RESIZE\_ALL\_COLUMNS\*);**

**​ table1.setModel(new DefaultTableModel(new Object[][] {},**

**​ new String[] {"菜品名称", "份数"}));**

**​ JLabel lblNewLabel = new JLabel("\u70B9\u9910\u684C\u53F7\uFF1A");**

**​ lblNewLabel.setFont(new Font("宋体", Font.\*PLAIN\*, 22));**

**​ lblNewLabel.setBounds(620, 31, 124, 34);**

**​ contentPane.add(lblNewLabel);**

**​ \*//\*\*开台按钮**

**\\* JButton btnNewButton = new JButton("\u5F00\u59CB\u70B9\u9910");**

**​ btnNewButton.addActionListener(new ActionListener() {**

**​ public void actionPerformed(ActionEvent e) {**

**​ ordd = new orderDAO();**

**​ String th = (String) comboBox.getSelectedItem();**

**​ try {**

**​ ordd.kaitai(th);**

**​ JOptionPane.\*showMessageDialog\*(null, "开台成功！请开始点餐！");**

**​ } catch (SQLException throwables) {**

**​ JOptionPane.\*showMessageDialog\*(null, "开台失败！此餐台已在用餐！");**

**​ }**

**​ }**

**​ });**

**​ btnNewButton.setFont(new Font("华光综艺\_CNKI", Font.\*PLAIN\*, 18));**

**​ btnNewButton.setBounds(707, 136, 117, 36);**

**​ contentPane.add(btnNewButton);**

**​ \*//add dish**

**\\* JButton btnNewButton\_1 = new JButton("\u6DFB\u52A0\u8BE5\u83DC\u54C1");**

**​ btnNewButton\_1.addActionListener(new ActionListener() {**

**​ public void actionPerformed(ActionEvent e) {**

**​ itd = new itemDAO();**

**​ dsd = new dishDAO();**

**​ ordit = new Orderitem();**

**​ \*//\*\*获取桌号**

**\\* String th = (String) comboBox.getSelectedItem();**

**​ \*//\*\*往订单详情实体设定内容**

**\\* ordit.setDesk(th);**

**​ ordit.setPay(1);**

**​ try {**

**​ \*//**

**\\* System.\*out\*.println((String) table.getValueAt(selectedRow,0)+"222");**

**​ \*//\* \*获取菜品\*\*id**

**\\* ordit.setDishId(dsd.getid((String) table.getValueAt(selectedRow,0)));**

**​ System.\*out\*.println((String) table.getValueAt(selectedRow,0)+"111");**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ ordit.setAmount(1);**

**​ \*//\*\*向购物车添加**

**\\* try {**

**​ itd.diancai(ordit);**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ \*//\* \*刷新\*\*table1**

**\\* try {**

**​ retablee(th);**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ }**

**​ });**

**​ btnNewButton\_1.setFont(new Font("华光综艺\_CNKI", Font.\*PLAIN\*, 18));**

**​ btnNewButton\_1.setBounds(693, 212, 141, 36);**

**​ contentPane.add(btnNewButton\_1);**

**​ \*//del dish**

**\\* JButton btnNewButton\_1\_1 = new JButton("\u5220\u9664\u8BE5\u83DC\u54C1");**

**​ btnNewButton\_1\_1.addActionListener(new ActionListener() {**

**​ public void actionPerformed(ActionEvent e) {**

**​ itd = new itemDAO();**

**​ dsd = new dishDAO();**

**​ ordit = new Orderitem();**

**​ \*//\*\*获取桌号**

**\\* String th = (String) comboBox.getSelectedItem();**

**​ \*//\*\*给实体设定号桌号和菜号来定位删除**

**\\* ordit.setDesk(th);**

**​ try {**

**​ System.\*out\*.println((String) table1.getValueAt(selectedRow1,0)+"222");**

**​ ordit.setDishId(dsd.getid((String) table1.getValueAt(selectedRow1,0)));**

**​ System.\*out\*.println((String) table1.getValueAt(selectedRow1,0)+"111");**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ \*//\*\*从购物车中删除**

**\\* try {**

**​ itd.shancai(ordit);**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ new account(ddh);**

**​ \*//\* \*刷新\*\*table1**

**\\* try {**

**​ retablee(th);**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ }**

**​ });**

**​ btnNewButton\_1\_1.setFont(new Font("华光综艺\_CNKI", Font.\*PLAIN\*, 18));**

**​ btnNewButton\_1\_1.setBounds(693, 280, 141, 36);**

**​ contentPane.add(btnNewButton\_1\_1);**

**​ JLabel lblNewLabel\_1 = new JLabel("\u83DC\u54C1\u7C7B\u522B\uFF1A");**

**​ lblNewLabel\_1.setFont(new Font("宋体", Font.\*PLAIN\*, 22));**

**​ lblNewLabel\_1.setBounds(23, 11, 124, 34);**

**​ contentPane.add(lblNewLabel\_1);**

**​ \*//\*\*退出界面**

**\\* JButton btnNewButton\_1\_2\_1 = new JButton("\u9000\u51FA\u672C\u754C\u9762");**

**​ btnNewButton\_1\_2\_1.addActionListener(new ActionListener() {**

**​ public void actionPerformed(ActionEvent e) {**

**​ dispose();**

**​ new MainFrame();**

**​ }**

**​ });**

**​ btnNewButton\_1\_2\_1.setFont(new Font("华光综艺\_CNKI", Font.\*PLAIN\*, 18));**

**​ btnNewButton\_1\_2\_1.setBounds(665, 583, 199, 36);**

**​ contentPane.add(btnNewButton\_1\_2\_1);**

**​ JLabel lblNewLabel\_2 = new JLabel("\u4E0B\u5355\u5DE5\u53F7\uFF1A");**

**​ lblNewLabel\_2.setFont(new Font("宋体", Font.\*PLAIN\*, 22));**

**​ lblNewLabel\_2.setBounds(750, 31, 124, 34);**

**​ contentPane.add(lblNewLabel\_2);**

**\*//Gonghao**

**\\* txtgh = new JTextField();**

**​ txtgh.setBounds(745, 77, 145, 33);**

**​ contentPane.add(txtgh);**

**​ txtgh.setColumns(10);**

**\*//pay**

**\\* JButton btnNewButton\_1\_1\_1 = new JButton("结账");**

**btnNewButton\_1\_1\_1.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**itd = new itemDAO();**

**ordering = new Ordering();**

**odd1 = new orderDAO();**

**tbd = new tableDAO();**

**ordit = new Orderitem();**

**ordd = new orderDAO();**

**boolean flag = false;**

**​ \*//\* \*获取桌号**

**\\* String th = (String) comboBox.getSelectedItem();**

**​ \*//\* \*判断顾客是否为\*\*VIP\*\*，提供打折**

**\\* try {**

**​ flag = itd.vip(txttel.getText());**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ if(flag == true){**

**​ ordering.setVip(2);**

**​ } else {**

**​ ordering.setVip(1);**

**​ }**

**​ \*//\* \*生成订单信息**

**\\* ordering.setOrderNo(CheckUtils.\*generateOrderNumber\*());**

**​ ddh = ordering.getOrderNo();**

**​ \*//\* \*获取桌号\*\*ID**

**\\* try {**

**​ ordering.setDeskId(tbd.getdkid(th));**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ \*//\* \*获取当前时间**

**\\* ordering.setCreatetime(CheckUtils.\*generateTimestamp\*());**

**​ \*//\* \*设置订单金额，考虑\*\*VIP\*\*折扣**

**\\* try {**

**​ if(ordering.getVip()==1){**

**​ ordering.setMoney(Double.\*parseDouble\*(itd.cm(th)));**

**​ } else {**

**​ BigDecimal m = new BigDecimal(0.95);**

**​ BigDecimal vm = new BigDecimal(String.\*valueOf\*(m.multiply(BigDecimal.\*valueOf\*(Double.\*parseDouble\*(itd.cm(th))))));**

**​ ordering.setMoney(CheckUtils.\*round\*(vm,2).doubleValue());**

**​ }**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ \*//\* \*设置订单的其他信息**

**\\* ordering.setCustomertel(txttel.getText());**

**​ ordering.setStatus("已支付");**

**​ \*//\* \*判断实付金额是否填写**

**\\* if(textField.getText().equals("")){**

**​ JOptionPane.\*showMessageDialog\*(null,"请填入实付金额！");**

**​ } else {**

**​ ordering.setPaidin(Double.\*parseDouble\*(textField.getText()));**

**​ }**

**​ \*//\* \*计算找零**

**\\* BigDecimal mon = new BigDecimal(ordering.getMoney());**

**​ BigDecimal paid = new BigDecimal(ordering.getPaidin());**

**​ BigDecimal cha = new BigDecimal(String.\*valueOf\*(paid.subtract(mon)));**

**​ ordering.setChange(CheckUtils.\*round\*(cha,2).doubleValue());**

**​ \*//\* \*判断服务员工号是否填写**

**\\* if(txtgh.getText().equals("")){**

**​ JOptionPane.\*showMessageDialog\*(null,"请填入下单工号！");**

**​ } else {**

**​ ordering.setEmid(Integer.\*parseInt\*(txtgh.getText())); \*//\*\*收集服务员工号用于插入历史订单**

**\\* }**

**​ \*//\* \*往数据库里插入订单信息**

**\\* try {**

**​ ordd.dn(ordering);**

**​ } catch (SQLException throwables) {**

**​ JOptionPane.\*showMessageDialog\*(null,"请输入正确的下单工号！");**

**​ }**

**​ \*//\* \*更新订单详情表中的订单编号**

**\\* try {**

**​ odd1.cg(ordering);**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ \*//\* \*将餐台改回空闲状态**

**\\* try {**

**​ tbd.trnfree(ordering.getDeskId());**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ \*//\* \*刷新购物车面板**

**\\* try {**

**​ retablee(tbd.getdkn(ordering.getDeskId()));**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ \*//\* \*应付金额更新**

**\\* lbmn.setText("");**

**​ }**

**​ });**

**​ btnNewButton\_1\_1\_1.setFont(new Font("华光综艺\_CNKI", Font.\*PLAIN\*, 18));**

**​ btnNewButton\_1\_1\_1.setBounds(665, 343, 199, 36);**

**​ contentPane.add(btnNewButton\_1\_1\_1);**

**​ JLabel lblNewLabel\_1\_1 = new JLabel("\u8D2D\u7269\u8F66\uFF1A");**

**​ lblNewLabel\_1\_1.setFont(new Font("宋体", Font.\*PLAIN\*, 22));**

**​ lblNewLabel\_1\_1.setBounds(23, 317, 124, 34);**

**​ contentPane.add(lblNewLabel\_1\_1);**

**​ JLabel lblNewLabel\_3 = new JLabel("\u5E94\u4ED8\uFF1A");**

**​ lblNewLabel\_3.setFont(new Font("宋体", Font.\*PLAIN\*, 24));**

**​ lblNewLabel\_3.setBounds(620, 467, 104, 47);**

**​ contentPane.add(lblNewLabel\_3);**

**\*//\*\*实付**

**\\* textField = new JTextField();**

**​ textField.setBounds(693, 533, 171, 33);**

**​ contentPane.add(textField);**

**​ textField.setColumns(10);**

**​ JLabel lblNewLabel\_3\_1 = new JLabel("\u5B9E\u4ED8\uFF1A");**

**​ lblNewLabel\_3\_1.setFont(new Font("宋体", Font.\*PLAIN\*, 24));**

**​ lblNewLabel\_3\_1.setBounds(620, 524, 104, 47);**

**​ contentPane.add(lblNewLabel\_3\_1);**

**\*//\*\*应付金额**

**\\* lbmn = new JLabel("");**

**​ lbmn.setFont(new Font("宋体", Font.\*PLAIN\*, 25));**

**​ lbmn.setBounds(707, 479, 127, 23);**

**​ contentPane.add(lbmn);**

**​ JLabel lblNewLabel\_3\_2 = new JLabel("\u987E\u5BA2\u624B\u673A\u53F7\uFF1A");**

**​ lblNewLabel\_3\_2.setFont(new Font("Dialog", Font.\*PLAIN\*, 24));**

**​ lblNewLabel\_3\_2.setBounds(599, 410, 188, 47);**

**​ contentPane.add(lblNewLabel\_3\_2);**

**\*//\*\*手机号**

**\\* txttel = new JTextField();**

**​ txttel.setColumns(10);**

**​ txttel.setBounds(737, 410, 171, 45);**

**​ contentPane.add(txttel);**

**​ setVisible(true);**

**​ }**

**​ \*//\*\*刷新购物车**

**\\* public void retable (String str){**

**​ orderDAO odd = new orderDAO();**

**​ Object[][] df;**

**​ Object[][] ifm = new Object[][]{};**

**​ try {**

**​ ifm = odd.swds(str);**

**​ System.\*out\*.println("666");**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ df = ifm;**

**​ String[] sx = new String[]{"菜品名称", "价格", "状态"};**

**​ DefaultTableModel model1 = new DefaultTableModel(df,sx);**

**​ table.setModel(model1);**

**​ }**

**​ \*//\*\*刷新表格**

**\\* public void retablee (String str) throws SQLException{**

**​ itemDAO itd = new itemDAO();**

**​ Object[][] df;**

**​ Object[][] ifm = new Object[][]{};**

**​ try {**

**​ ifm = itd.swgwc(str);**

**​ System.\*out\*.println("999");**

**​ } catch (SQLException throwables) {**

**​ throwables.printStackTrace();**

**​ }**

**​ df = ifm;**

**​ String[] sx = new String[]{"菜品名称", "份数"};**

**​ DefaultTableModel model1 = new DefaultTableModel(df,sx);**

**​ table1.setModel(model1);**

**​ lbmn.setText(itd.cm(str));**

**​ }**

**}**

点单DAO类代码

package DAO.ordering;

import DAO.table.tableDAO;

import entity.Ordering;

import entity.Orderitem;

import util.JdbcUtils;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class orderDAO {

private Connection conn = null;

private PreparedStatement stmt = null;

private ResultSet rs =null;

private PreparedStatement stmt1 = null;

private PreparedStatement stmt2 = null;

private ResultSet rs1 =null;

private tableDAO tbd;

//展示菜品

public Object[][] swds(String cate) throws SQLException {

conn = JdbcUtils.getConnection();

String sql = "select \* from dish where categoryname = ? and status ='在售'";

stmt = conn.prepareStatement(sql);

stmt.setString(1,cate);

rs = stmt.executeQuery();

int count = 0;

while(rs.next()){

count++;

}

rs = stmt.executeQuery();

Object [][] info = new Object[count][3];

count = 0;

while (rs.next()) {

info[count][0] = rs.getString("name");

info[count][1] = rs.getDouble("price");;

info[count][2] = rs.getString("status");;

count++;

}

JdbcUtils.closeResource(conn,rs,stmt);

return info;

}

//开始点餐（开台并选中点餐台号）

public void kaitai(String th) throws SQLException{

String str = new String(th);

conn = JdbcUtils.getConnection();

String sql = "update desk set desk.status ='用餐中' where desk.no = '"+th+"' and desk.status = '空闲'";

stmt = conn.prepareStatement(sql);

int n = stmt.executeUpdate();

System.out.println(n);

System.out.println(sql);

System.out.println("开始点餐！");

}

// 结账并向历史订单表插入新的记录!

public void dn(Ordering ord) throws SQLException{

conn = JdbcUtils.getConnection();

String sql = "INSERT INTO ordering VALUES (0,?,?,?,?,?,?,?,?,?,?)";

stmt = conn.prepareStatement(sql);

stmt.setString(1,ord.getOrderNo());

stmt.setInt(2,ord.getDeskId());

stmt.setString(3,ord.getCreatetime());

stmt.setDouble(4,ord.getMoney());

stmt.setString(5,ord.getCustomertel());

stmt.setString(6,ord.getStatus());

stmt.setInt(7,ord.getEmid());

stmt.setInt(8,ord.getVip());

stmt.setDouble(9,ord.getPaidin());

stmt.setDouble(10,ord.getChange());

stmt.executeUpdate();

System.out.println(124);

}

//结账 将item表里的菜品都改成pay=2（已支付）

public void cg(Ordering or) throws SQLException{

tbd = new tableDAO();

conn = JdbcUtils.getConnection();

String sql = "UPDATE orderitem SET ordering =? WHERE desk=? AND pay = 1";

String sql1 = "UPDATE orderitem SET pay=2 WHERE ordering =?";

stmt = conn.prepareStatement(sql);

stmt1 = conn.prepareStatement(sql1);

stmt.setString(1,or.getOrderNo());

stmt.setString(2,tbd.getdkn(or.getDeskId()));

stmt1.setString(1,or.getOrderNo());

stmt.executeUpdate();

stmt1.executeUpdate();

}

}