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** | **岳大俊** |
| **学号** | **B20210304122** |
| **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 | 岳大俊 | 学号 | B20210304122 | 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**：  需求分析：根据餐饮行业的特点和该企业的实际情况，本餐饮管理系统以餐饮业务为基础，突出管理，从专业角度出发，提供科学有效的管理模式。能够针对中餐因其菜品的多样化和特色化的服务提供标准化的管理。能够提供符合餐饮企业自身要求的较科学的标准化、流程化管理，解决餐饮行业专业人才欠缺的问题。能够依据订餐、点菜、结账等环节的繁重化、复杂化问题，实现强化管理、降低成本、堵漏节流等效益。能够针对企业的经营现状做出科学的分析，使得企业对市场的应变能力得到提高。  系统设计：系统管理具有员工管理、客户管理、餐台管理、菜品管理、点菜管理、结账管理6大模块，设置留个菜单分别表示各个板块，各个板块具有对各信息进行增删改查功能，且对某客户、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。对某一餐台通过统计显示消费的菜品清单并统计出消费金额，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  编码实现：JAVA的GUI类设计开发友好的人机接口，JAVA语言设计监听用户操作和JDBC进行数据库的处理。  系统测试：用eclipse运行JAVA代码，测试数据，测试系统的功能性和实用性。  交付实施：测试成功，对系统进行维护，并开始投入使用。 | | | | | |
| **实训成果**：  通过JAVA代码编辑实现完成了一个基础的餐饮管理系统，实现了餐饮服务的基本功能。了解如何使界面美观，如何连接数据库，对数据库数据进行增删改查等操作。 | | | | | |
| **实训总结**：（包括心得体会、存在的问题和改进方向。）  心得体会：通过此次实训，加深了对JAVA的GUI编程的了解、界面设计，了解了如何对界面进行美化、添加各种部件，加深了对监听器的使用。对JDBC连接数据库操作印象加深。在系统设置期间，要结合实际问题，增加功能，建立各表之间的联系。遇到问题，根据错误提示，认真思考解决问题，主动查询相关资料，虚心请教同学，将问题解决。  存在的问题：界面设计较于简洁，实现方法低级，数据库连接安全性不高，  改进方向：   1. 增加更多功能：添加新的模块和功能来扩展系统。 2. 完善用户体验：提升系统的易用性和效率、改进界面设计、提供更好的反馈、简化用户操作。 3. 改进性能：针对系统运行的速度、响应时间等方面进行优化，使得系统更加高效。 4. 增加安全措施：加强系统安全性，包括数据传输加密、账号权限管理、审计日记等安全功能。 5. 优化数据库设计：调整数据库结构，合理设计索引和查询语句等，提高系统的查询和处理效率。 6. 移植到云端：将系统迁移到云端平台上，具备更好的可扩展性和稳定性，同时降低硬件成本。根据具体情况需要，结合实际，选择合适的改进方向。 | | | | | |

基于C/S架构的餐饮管理系统

设计说明书

作者：岳大俊（B20210304122）

计算机科学与工程学院

2023年 06 月 15日

**摘要**

本课程设计的是餐饮管理系统，系统具有员工管理、客户管理、菜品管理、餐台管理、点菜管理、结账管理六大功能。设计使用JAVAGUI编程对界面进行美化布局设计，根据C/S构架对各板块的布局进行开发，通过JDBC连接数据库，并进行数据库编程，完成对员工信息的增删改查、对客户信息的增删改查、对于菜品信息的增删改查、对餐台信息的增删改查，点菜时，自动选择正售菜品和空餐台，并对点菜表进行增删改查，对结账列表增删改查、对结账信息的增删改查。

**关键词：**java ，C/S架构，餐饮管理系统 ，JDBC。

目录

[已知技术参数和设计要求： 2](#_Toc138181168)

[1. 问题描述（功能要求）： 2](#_Toc138181169)

[**2. 运行环境要求：** 2](#_Toc138181170)

[3. 技术要求： 2](#_Toc138181171)

[实训工作量： 3](#_Toc138181172)

[工作计划： 3](#_Toc138181173)

[1 引言 1](#_Toc138181174)

[1.1 编写目的 1](#_Toc138181175)

[1.2 参考资料 1](#_Toc138181176)

[2 需求规约 2](#_Toc138181177)

[2.1 功能需求 2](#_Toc138181178)

[2.2 界面需求 6](#_Toc138181179)

[2.3 数据需求 7](#_Toc138181180)

[3 系统设计 9](#_Toc138181181)

[3.1 运行环境 9](#_Toc138181182)

[3.2 系统静态结构设计 9](#_Toc138181183)

[3.3 人机接口设计 10](#_Toc138181184)

[3.4 登录模块设计说明 19](#_Toc138181185)

[3.5 主界面模块设计说明 20](#_Toc138181186)

[4 数据库设计 21](#_Toc138181187)

[4.1 数据库环境说明 21](#_Toc138181188)

[4.2 数据库的命名规则 21](#_Toc138181189)

[4.3 逻辑结构设计 21](#_Toc138181190)

[4.4 物理结构设计 23](#_Toc138181191)

[4.5 安全性设计 30](#_Toc138181192)

[5 测试用例设计 31](#_Toc138181193)

[附录 35](#_Toc138181194)

[附录1 程序运行结果 35](#_Toc138181195)

[附录2 程序源代码 38](#_Toc138181196)

# 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] 陈强.精通Java开发技术 . 清华大学出版社.2014

[2] 黄俊.Java程序设计与应用开发.第二版.机械工业出版社.2014

[3] 王爱国.Java面向对象程序设计. 机械工业出版社.2014

[4] 向昌成.Java程序设计项目化教程. 清华大学大学出版社.2013

# 2 需求规约

## 2.1 功能需求

要求完成一个C/S架构的餐饮管理系统

1. 对餐饮服务系统进行功能设计
2. 设计餐饮管理系统数据库
3. 利用JDBC对数据库进行连接
4. 利用JAVA语言设计管理员操作、修改数据库数据

餐饮管理系统6大板块

1. 员工管理：对员工信息进行增删改查
2. 客户管理：对客户信息进行增删改查
3. 餐台管理：对餐台信息进行增删改查
4. 菜品管理：对菜品信息进行增删改查
5. 点菜管理，服务员对顾客进行选空餐台，点正售菜品，同时对点菜信息进行增删改查
6. 结账管理：对顾客消费菜品进行结账，对点菜信息进行查询。结账完成将所选忙餐台状态变为空，并将结账信息全部改为已付款。

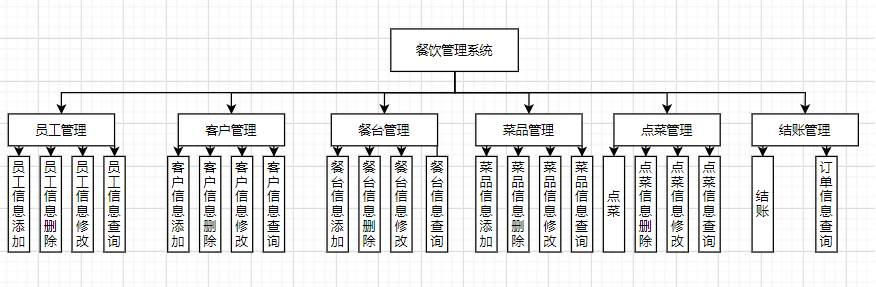


图2.1 餐饮管理系统功能模块图

2.1.1 管理员登录用例说明

用例名称：管理员登录

用例描述：餐饮服务行业的管理员登录餐饮管理系统

餐饮管理员对员工信息、菜品信息、客户信息、餐台信息进行增删改查，可以根据客户需求进行点餐，并进行结账。

前置条件：管理员舒服账号密码

后置条件：账号密码匹配成功，管理员成功登录系统

活动步骤：

1. 管理员输入账号密码。
2. 系统读取管理员的账号密码的信息。

扩展点：

1. 管理员可通过数据库更改密码

异常处理：

1. 如果帐号与密码不匹配，用例停止
2. 账号密码为空，用户终止

2.1.2 员工信息管理用例说明

用例名称：员工管理

用例描述：管理员对入职员工信息进行登记，对员工信息进行增删改查

前置条件：系统入职、升职、离职请求。

后置条件：系统将员工信息录入员工管理信息表。

活动步骤：

1.打开员工管理，显示所有员工信息。

2.员工入职，进行信息登记，添加员工的id、编号、姓名、性别、出生日期、身份证号、籍贯、家庭住址、学历、婚姻状态、职业信息。系统录取员工的信息。

3.员工离职，用鼠标左键确定选定的员工信息，系统删除员工的所有信息。

4.员工升职或信息错误，选中该员工信息，系统将此员工所有信息提出，进行修改。

5.可以根据员工id、编号、姓名、性别、籍贯、家庭住址、婚姻状态、职位进行查询，并满足条件的员工信息。

扩展点：

1.员工可以升职。

异常处理：

1. 查询时，如果员工信息不存在，则查询失败，用例终止。
2. 添加时，若员工有一项信息为空或已存在该员工信息时，则添加失败，用例停止。

2.1.3 客户信息管理用例说明

用例名称：客户管理

用例描述：管理员对客户信息进行登记，对客户信息进行增删改查

前置条件：客户办理会员。

后置条件：系统将客户信息录入客户管理信息表。

活动步骤：

1.打开客户管理，显示所有客户信息。

2.客户办理会员，进行信息登记，添加客户的id、姓名、性别、单位、电话、贵宾卡号。系统录取客户的信息。

3.客户退理会员，用鼠标左键确定选定的客户信息，系统删除客户的所有信息。

4.客户信息错误，选中该客户信息，系统将客户所有信息提出，进行修改。

5.查询客户信息，可以根据客户id、姓名、性别、单位、电话、贵宾卡号进行查询，显示满足条件的客户信息。

异常处理：

1. 查询时，如果客户信息不存在，则查询失败，用例终止。
2. 添加时，若客户有一项信息为空或已存在该客户信息时，则添加失败，用例停止。

异常处理：

1. 添加信息时，如果餐台信息为空，则用例终止。
2. 查询时，若客户餐台错误，则查询失败，用例停止。

2.1.4 餐台信息管理用例说明

用例名称：餐台管理

用例描述：管理员对餐台信息进行登记，对餐台信息进行增删改查

前置条件：新餐台信息加入。

后置条件：系统录入餐台管理信息表。

活动步骤：

1.打开餐台管理，显示所有餐台信息。

2.新增餐台信息，进行信息登记，添加餐台的id、编号、座位数、状态、默认开台时间。

3.删除餐台信息，鼠标左键选中餐台信息，系统将该餐台所有信息行删除。

4.修改餐台信息，餐台开台或信息错误，选中该餐台信息，系统将此餐台所有信息提出，进行修改。

5.查询餐台信息，可以根据餐台id、编号、座位数、状态进行查询，显示满足条件的餐台信息。

异常处理：

1.查询时，如果餐台信息不存在，则查询失败，用例终止。

2.添加时，若餐台有一项信息为空或已存在该餐台信息时，则添加失败，用例停止。

2.1.5 菜品信息管理用例说明

用例名称：菜品管理

用例描述：管理员对菜品信息进行登记，对菜品信息进行增删改查

前置条件：新菜品信息加入。

后置条件：系统录入菜品管理信息表。

活动步骤：

1.打开菜品管理，显示所有菜品信息。

2.新增菜品信息，进行信息登记，添加菜品的id、分类、菜品名、单位、单价、状态、折扣。

3.删除菜品信息，鼠标左键选中菜品信息，系统将该菜品所有信息行删除。

4.修改菜品信息，菜品打折或信息错误，选中该菜品信息，系统将此菜品所有信息提出，进行修改。

5.可以根据菜品id、分类、菜品名、状态进行查询，显示满足条件的菜品信息。

异常处理：

1. 查询时，如果菜品信息不存在，则查询失败，用例终止。
2. 添加时，若菜品有一项信息为空或已存在该菜品信息时，则添加失败，用例停止。

2.1.6 点菜管理用例说明

用例名称：点菜管理

用例描述：顾客点菜，服务员进行选餐台，点菜进行登记，对客户点菜菜品信息进行增删改查

前置条件：顾客光临。

后置条件：系统录入顾客餐台号以及所点菜品。

活动步骤：

1.顾客光临，服务员服务顾客，确定客户姓名，餐台和就餐人数。

2.服务员给顾客菜单进行选菜，添加菜品编号、菜品名、数量、就餐人数。

3.删除点菜信息列表，鼠标左键选中点菜特定列表，系统将该点菜信息行删除。

4.修改点菜信息错误，选中该点菜信息，系统将此点菜菜品信息提出，进行修改。

5.可以根据点菜的菜品编号、菜品名称进行查询，显示满足条件的点菜信息。

扩展点：

1. 点菜时，会列出状态处于空状态餐台，将选中餐台的状态换成忙
2. 点菜时，会列出状态处于“1”（可售卖）的菜品，供顾客选择。
3. 点菜时，会将菜品编号，顾客名，菜品名，菜品数量添加入结账表格里，并与结账表同步进行添加，删除操作。

异常处理：

1. 查询时，如果点菜信息不存在，则查询失败，用例终止。
2. 添加时，若点菜信息有一项信息为空或已存在该菜品信息时，则添加失败，用例停止。
3. 扩展点：

2.1.7 结账管理用例说明

用例名称：结账管理

用例描述：顾客结账，前台打开结账管理对客户结账菜品信息进行增删改查。

前置条件：顾客结账。

后置条件：系统录入顾客餐台号以及所点菜品。

活动步骤：

1.顾客到前台结账或者找服务员结账，打开结账管理。

2.结账表展示明细。

3.可以根据结账表的顾客姓名、菜品编号、菜品名称进行查询，显示满足条件的结账信息。

4.系统自动算出该顾客未结账的总金额，再填上优惠金额，算出找零。

7.最后结账。

扩展点：

1.结账后，会将结账表里的状态从“未结账”改为“已结账”。

2.结账后，会将餐台表里的状态从“忙”改为“空”。

扩展点：

1. 确定结账后，将该点菜表里的

异常处理：

1.查询时，如果结账信息不存在，则查询失败，用例终止。

## 2.2 界面需求

* 界面美观，部件规整
* 操作简单，使用方便
* 数据显示清晰，高效运转

## 2.3 数据需求

表2.1 管理员实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 不是null | 键 | 注释 |
| Id | int | 10 | √ | 主键 | 主码，非空 |
| Username | varchar | 20 | √ |  | 用户名 |
| password | varchar | 20 | √ |  | 密码 |

**表2.2员工实体**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 不是null | 键 | 注释 |
| Id | int | 10 | √ | 主键 | 主码，非空 |
| Emp\_id | varchar | 4 | √ |  | 员工编号 |
| Emp\_name | varchar | 20 | √ |  | 员工姓名 |
| sex | varchar | 2 | √ |  | 性别 |
| Birthday | date |  | √ |  | 出生日期 |
| ID\_card | varchar | 18 | √ |  | 身份证号 |
| Native\_place | varchar | 20 | √ |  | 籍贯 |
| Home\_address | varchar | 40 | √ |  | 家庭地址 |
| Education | varchar | 10 | √ |  | 教育 |
| marriage | varchar | 2 | √ |  | 婚否 |
| position | varchar | 10 | √ |  | 职业 |

**表2.3客户实体**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 不是null | 键 | 注释 |
| Id | int | 10 | √ | 主键 | 主码，非空 |
| Name | varchar | 20 | √ |  | 客户姓名 |
| Sex | varchar | 4 | √ |  | 性别 |
| Company | varchar | 20 | √ |  | 单位 |
| Tel | varchar | 11 | √ |  | 电话号码 |
| carID | varchar | 10 | √ |  | 贵宾卡号 |

**表2.4餐台实体**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 不是null | 键 | 注释 |
| Id | int | 11 | √ | 主键 | 主码，非空 |
| Table\_id | varchar | 8 | √ |  | 餐台编号 |
| Table\_num | int | 11 | √ |  | 座位数 |
| Status | varchar | 8 | √ |  | 状态 |
| Open\_time | time |  | √ |  | 开台时间 |

**表2.5菜品实体**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 不是null | 键 | 注释 |
| Id | int | 11 | √ | 主键 | 主码，非空 |
| Categorize | varchar | 10 | √ |  | 菜品分类 |
| Dish\_name | varchar | 50 | √ |  | 菜品名 |
| Unit | varchar | 10 | √ |  | 单位 |
| price | decimal | 10 | √ |  | 单价 |
| status | varchar | 8 | √ |  | 状态 |
| discount | int | 11 | √ |  | 折扣 |

**表2.6点菜管理**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 不是null | 键 | 注释 |
| Id | int | 11 | √ | 主键 | 主码，非空 |
| Name | varchar | 11 | √ |  | 客户名 |
| Table\_id | varchar | 8 | √ |  | 餐台编号 |
| Food\_id | varchar | 11 | √ |  | 菜品编号 |
| Food\_name | varchar | 10 | √ |  | 菜品名称 |
| Number | varchar | 4 | √ |  | 数据 |
| Meal\_number | varchar | 4 | √ |  | 就餐人数 |

**表2.7结账管理**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 不是null | 键 | 注释 |
| Id | int | 11 | √ | 主键 | 主码，非空 |
| Food\_id | varchar | 11 | √ |  | 菜品编号 |
| Name | varchar | 10 | √ |  | 客户姓名 |
| Food\_name | varchar | 20 | √ |  | 菜品名 |
| Number | varchar | 11 | √ |  | 数量 |
| Price | varchar | 8 | √ |  | 价格 |
| status | varchar | 4 | √ |  | 状态 |

# 3 系统设计

## 3.1 运行环境

操作系统：Windows 11

数据库：Navicat Mysql 16.0.11

## 3.2 系统静态结构设计

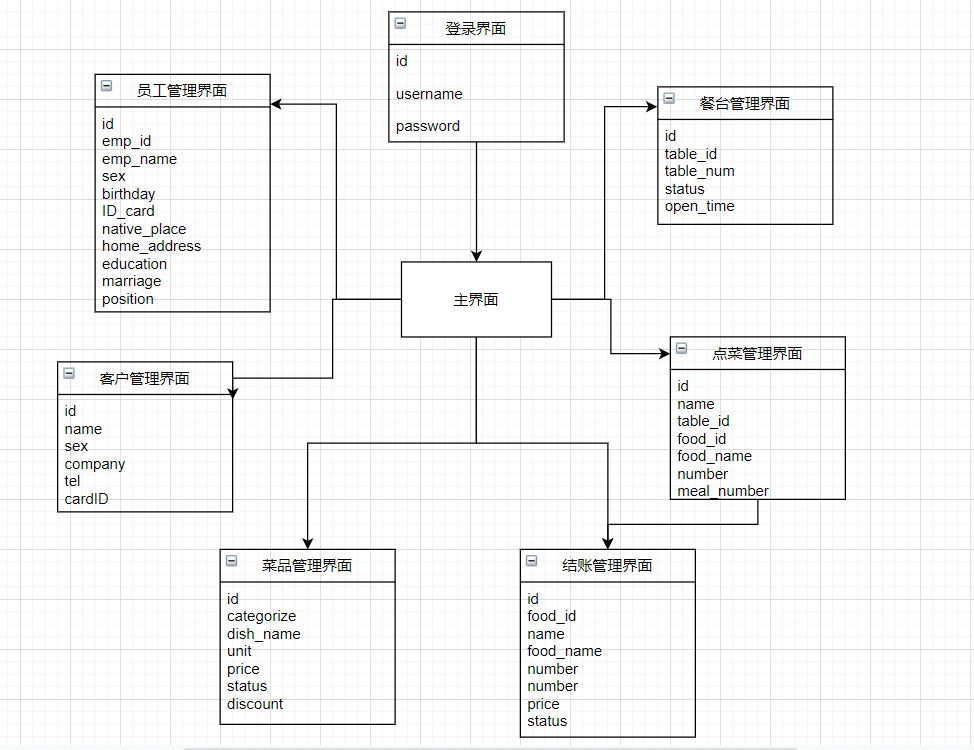


图3.1餐饮管理系统总体类图

系统静态结构说明

先设置登录界面，输入用户名和密码点击登录按钮与数据库的数据进行匹配，匹配成功成功登录，进入主界面，主界面有员工管理、客户管理、餐台管理、菜品管理、点菜管理、结账管理六大菜单栏。分别打开，对应各自信息的增删改查功能，其中点菜管理可对结账管理的信息进行间接操作。

## 3.3 人机接口设计

3.3.1 设计应用类

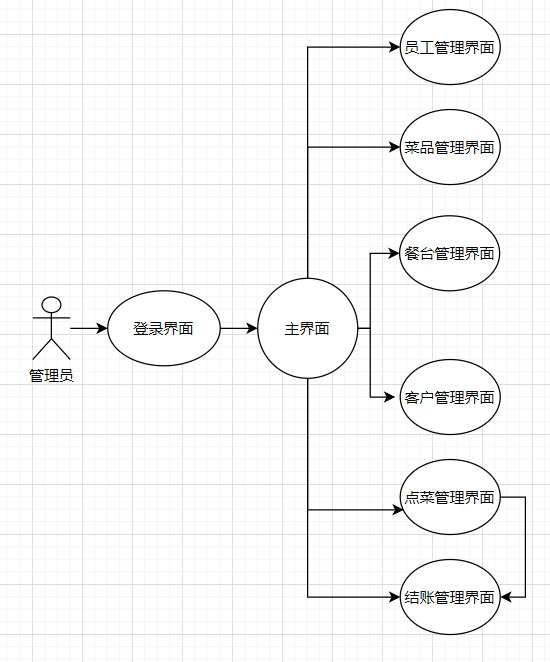


图3.2 餐饮管理的设计类图

3.3.2 用户接口1（登录界面）设计

**主要部件：**

JTextField:文本框，输入框

JButton：按钮，点击触发监听器事件

JLabel：文本，显示标签

Image：设置界面Logo

JComboBox:设置下拉框

ImageIcon：设置界面背景图片

**实现功能：**

1.管理员输入账号密码，匹配数据库，账号密码匹配成功则成功登录。

图3.3 登录界面设计图

3.3.3 用户接口2（主界面）设计

**主要部件：**

JTextField:文本框，输入框

JButton：按钮，点击触发监听器事件

JLabel：文本，显示标签

Image：设置界面Logo

JComboBox:设置下拉框

ImageIcon：设置界面背景图片

JMenuBar：创建菜单栏对象

JMenu：创建菜单

**实现功能：**

1. 点击员工管理，打开员工管理界面
2. 点击餐台管理，打开餐台管理界面
3. 点击客户管理，打开客户管理界面
4. 点击菜品管理，打开菜品管理界面
5. 点击点菜管理，打开点菜管理界面
6. 点击结账管理。打开结账管理界面

图3.4 主界面设计图

3.3.4 用户接口3（员工管理界面）设计

**主要部件：**

JTextField:文本框，输入框

JButton：按钮，点击触发监听器事件

JLabel：文本，显示标签

Image：设置界面Logo

JComboBox:设置下拉框

ImageIcon：设置界面背景图片

JMenuBar：创建菜单栏对象

JMenu：创建菜单

**实现功能：**

1. 点击添加，进行员工信息添加
2. 选中删除行，点击删除，删除该行信息
3. 选中修改行，点击修改，修改该行信息
4. 在文本框里输出员工的id,编号，姓名，籍贯，家庭住址，学历，婚否，职业任意信息，点击查找，在数据库里进行匹配，查询复合条件的所有信息
5. 点击刷新，刷新员工管理界面

图3.5 员工管理界面设计图

3.3.5 用户接口4（客户管理界面）设计

**主要部件：**

JTextField:文本框，输入框

JButton：按钮，点击触发监听器事件

JLabel：文本，显示标签

Image：设置界面Logo

JComboBox:设置下拉框

ImageIcon：设置界面背景图片

JMenuBar：创建菜单栏对象

JMenu：创建菜单

**实现功能：**

1. 点击添加，进行客户信息添加
2. 选中删除行，点击删除，删除该行信息
3. 选中修改行，点击修改，修改该行信息
4. 在文本框里输出客户的id,用户名，性别，单位，电话，贵宾卡号其中的任一信息，点击查找，在数据库里进行匹配，查询复合条件的所有信息
5. 点击刷新，刷新客户管理界面

图3.6 客户管理界面设计图

3.3.6 用户接口5（餐台管理界面）设计

**主要部件：**

JTextField:文本框，输入框

JButton：按钮，点击触发监听器事件

JLabel：文本，显示标签

Image：设置界面Logo

JComboBox:设置下拉框

ImageIcon：设置界面背景图片

JMenuBar：创建菜单栏对象

JMenu：创建菜单

**实现功能：**

1. 点击添加，进行餐台信息添加
2. 选中删除行，点击删除，删除该行信息
3. 选中修改行，点击修改，修改该行信息
4. 在文本框里输出餐台的id,编号，座位数，状态其中的任一信息，点击查找，在数据库里进行匹配，查询复合条件的所有信息
5. 点击刷新，刷新餐台管理界面



图3.7 餐台管理界面设计图

3.3.7 用户接口6（菜品管理界面）设计

**主要部件：**

JTextField:文本框，输入框

JButton：按钮，点击触发监听器事件

JLabel：文本，显示标签

Image：设置界面Logo

JComboBox:设置下拉框

ImageIcon：设置界面背景图片

JMenuBar：创建菜单栏对象

JMenu：创建菜单

**实现功能：**

1. 点击添加，进行菜品信息添加
2. 选中删除行，点击删除，删除该行信息
3. 选中修改行，点击修改，修改该行信息
4. 在文本框里输出菜品的id,分类，菜品名，单位，单价，状态，折扣其中的任一信息，点击查找，在数据库里进行匹配，查询复合条件的所有信息
5. 点击刷新，刷新菜品管理界面

图3.8 菜品管理界面设计图

3.3.8 用户接口7（点菜管理界面）设计

**主要部件：**

JTextField:文本框，输入框

JButton：按钮，点击触发监听器事件

JLabel：文本，显示标签

Image：设置界面Logo

JComboBox:设置下拉框

ImageIcon：设置界面背景图片

JMenuBar：创建菜单栏对象

JMenu：创建菜单

**实现功能：**

1. 点击添加，进行订单信息添加
2. 选中删除行，点击删除，删除该行信息
3. 选中修改行，点击修改，修改该行信息
4. 在文本框里输出订单的客户名,餐台号，就餐人数信息，点击查找，在数据库里进行匹配，查询复合条件的所有信息
5. 点击刷新，刷新点菜管理界面
6. 点击结账，进入结账管理界面

图3.9 菜品管理界面设计图

3.3.9 用户接口8（结账管理界面）设计

**主要部件：**

JTextField:文本框，输入框

JButton：按钮，点击触发监听器事件

JLabel：文本，显示标签

Image：设置界面Logo

JComboBox:设置下拉框

ImageIcon：设置界面背景图片

JMenuBar：创建菜单栏对象

JMenu：创建菜单

**实现功能：**

1. 输入客户姓名，点击查询，显示该顾客所有订单。
2. 系统自动算出，该顾客未支付的金额，前天输入实收金额和优惠金额，点击找零算出应找金额。
3. 点击结账，结账完成，将餐台状态变为空，订单菜品状态变为已支付。



图3.10 结账管理界面设计图

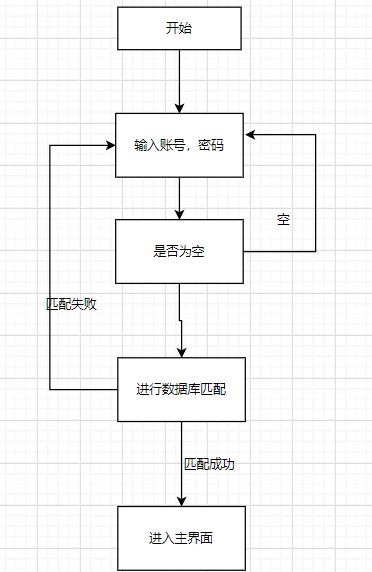
## 3.4 登录模块设计说明

（1）功能：该模块是管理员登录餐饮管理系统的操作

（2）测试要点：

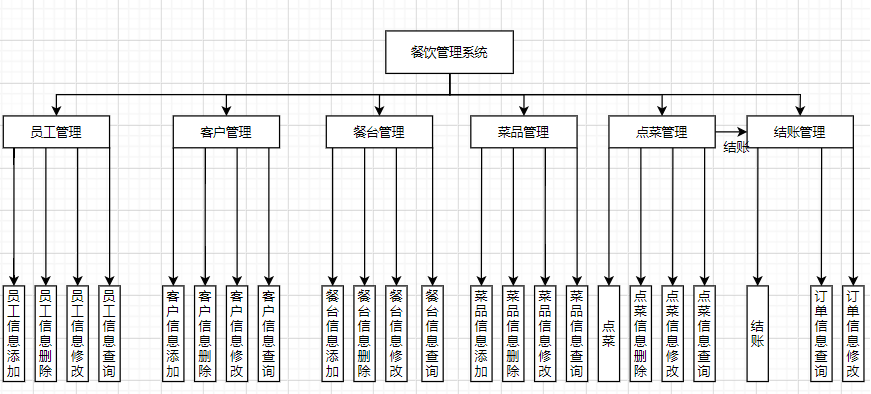
1.输入账号密码不能为空

2.输入的账号密码要匹配。



## 3.5 主界面模块设计说明

（1）功能：进入餐饮管理系统，打开员工管理界面，客户管理界面，餐台管理界面，菜品管理界面，点菜管理界面，结账管理界面。



# 4 数据库设计

## 4.1 数据库环境说明

操作系统：Windows 11

数据库：Navicat Mysql 16.0.11

## 4.2 数据库的命名规则

数据库名：餐饮管理

数据表命名规范：

（1）采用26个英文字母，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。。

（4）表名称不应该取得太长。

（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

## 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（id，用户名，密码）

员工（id,员工编号，员工姓名，性别，出生日期，身份证号，籍贯，家庭地址，学历，婚否，职业）

客户（id,用户名。性别，单位。电话，贵宾卡号）

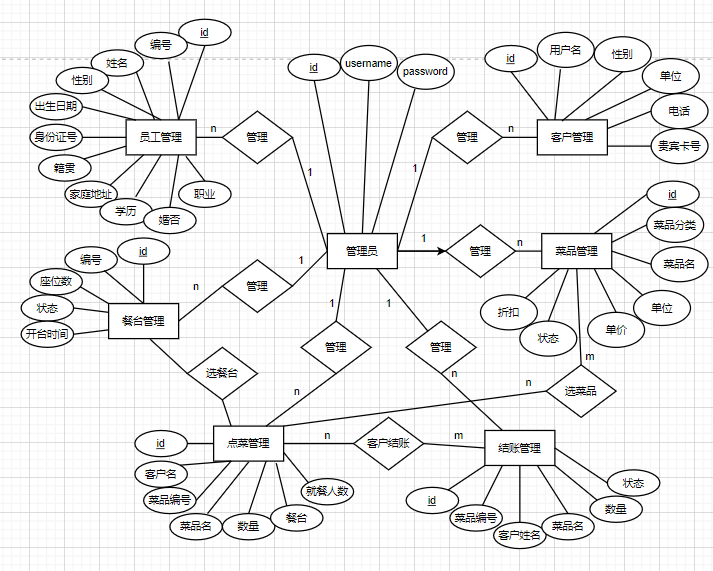
菜品（id,菜品分类，菜品名，单位，单价，状态，折扣）

餐台（id,餐台编号，座位数，状态，开台时间）

点单（id,客户名，菜品编号，菜品名，数量，餐台，就餐人数）

结账（id,菜品编号，客户姓名，菜品名，数量，价格，状态）

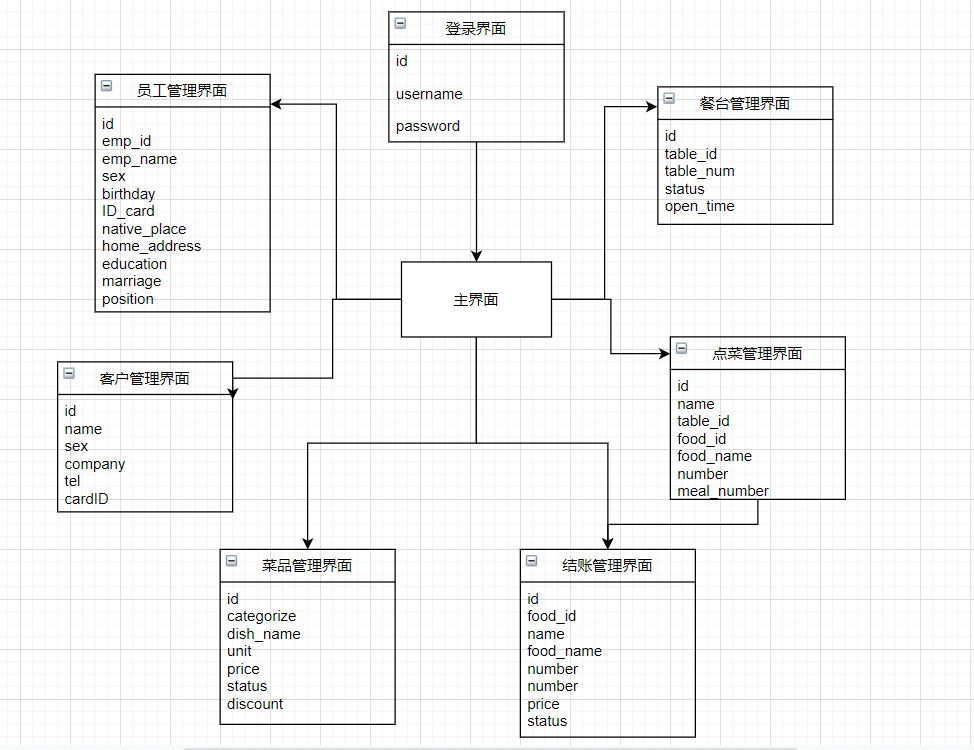
E-R图



范式分析

第三范式，消除了非主属性对主属性的传递依赖

逻辑模型图



## 4.4 物理结构设计

提示：建立系统程序员视图，包括：数据在内存中的安排，包括对索引区、缓冲区的设计；所使用的外存设备及外存空间的组织，包括索引区、数据块的组织与划分；访问数据的方式方法。下列必选项必须说明：

4.4.1 数据库表

表4.1 表汇总

|  |  |
| --- | --- |
| 表 名 | 功 能 说 明 |
| administrator | 存储管理员账号密码 |
| employee | 存储员工信息 |
| client | 存储客户信息 |
| table | 存储餐台信息 |
| dish | 存储菜品信息 |
| ordering | 存储点菜信息 |
| price | 存储结账信息 |

表4.2 管理员表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | administrator | | | | | |
| 数据库用户 | | root | | | | | |
| 主 键 | | id | | | | | |
| 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 | Id | Int | N | Y |  |  | 主码，最大长度10 |
| 2 | username | varchar | N | Y |  |  | 最大长度20 |
| 3 | password | varchar | N | N |  |  | 最大长度20 |
| sql脚本 | | Creat table administrator（  id int(10) primary key,not null,unique,  username varchar(20) not null ,unique,  password varchar(20) not null  ）; | | | | | |
| 备注 | |  | | | | | |

表4.3 员工表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | employee | | | | | |
| 数据库用户 | | root | | | | | |
| 主 键 | | id | | | | | |
| 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 | id | int | N | Y |  |  | 主码，最大长度10 |
| 2 | emp\_id | varchar | N | Y |  |  | 最大长度4 |
| 3 | emp\_name | varchar | N | N |  |  | 最大长度20 |
| 4 | Sex | varchar | N | N |  |  | 最大长度为2 |
| 5 | birthday | date | N | N |  |  |  |
| 6 | ID\_card | varchar | N | Y |  |  | 最大长度为18 |
| 7 | native\_place | varchar | N | N |  |  | 最大长度为20 |
| 8 | home\_adress | varchar | N | N |  |  | 最大长度为40 |
| 9 | education | varchar | N | N |  |  | 最大长度为10 |
| 10 | marriage | varchar | N | N |  |  | 最大长度为2 |
| 11 | position | varchar | N | N |  |  | 最大长度为10 |
| sql脚本 | | Creat table employee（  id int(10) primary key,not null,unique,  emp\_id int(4) not null ,unique,  emp\_id varchar(20) not null,  sex varchar(2) not null,  birthday date not null,  ID\_card varchar(18) not null,  Native\_adress varchar(20) not null,  Home\_address varchar(40) not null,  education varchar(20) not null,  marriage varchar(2) not null,  position varchar(10) not null,  ）; | | | | | |
| 备注 | |  | | | | | |

表4.3 客户表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | client | | | | | |
| 数据库用户 | | root | | | | | |
| 主 键 | | id | | | | | |
| 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 | id | int | N | Y |  |  | 主码，最大长度10 |
| 2 | name | varchar | N | N |  |  | 最大长度20 |
| 3 | Sex | varchar | N | N |  |  | 最大长度2 |
| 4 | company | varchar | N | N |  |  | 最大长度为20 |
| 5 | tel | date | N | N |  |  | 最大长度为11 |
| 6 | cardID | varchar | N | Y |  |  | 最大长度为10 |
| sql脚本 | | Creat table client（  id int(10) primary key,not null,unique,  name varchar(20) not null,  sex varchar(2) not null,  company varchar(20) not null,  tel varchar(11) not null,  cardID varchar(10) not null,  ）; | | | | | |
| 备注 | |  | | | | | |

表4.4 餐台表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | table | | | | | |
| 数据库用户 | | root | | | | | |
| 主 键 | | id | | | | | |
| 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 | id | int | N | Y |  |  | 主码，最大长度10 |
| 2 | Table\_id | varchar | N | Y |  |  | 最大长度8 |
| 3 | Table\_num | Int | N | N |  |  | 最大长度11 |
| 4 | Status | Varchar | N | N |  |  | 最大长度为8 |
| 5 | Open\_time | Time | N | N |  | 00:00:00 |  |
| sql脚本 | | Creat table table（  id int(10) primary key,not null,unique,  table\_id varchar(4) not null ,unique,  table\_num int(11) not null,  status varchar(8) not null,  open\_time time not null,  constraint fk\_table\_id foreign key(table\_id)references ordering(table\_id)  ）; | | | | | |
| 备注 | |  | | | | | |

表4.5 菜品表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | dish | | | | | |
| 数据库用户 | | root | | | | | |
| 主 键 | | id | | | | | |
| 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 | id | int | N | Y |  |  | 主码，最大长度10 |
| 2 | categorize | varchar | N | Y |  |  | 最大长度为10 |
| 3 | dish\_name | varchar | N | N |  |  | 最大长度为50 |
| 4 | unit | varchar | N | N |  |  | 最大长度为10 |
| 5 | price | varchar | N | N |  |  | 最大长度为10 |
| 6 | status | varchar | N | Y |  | 空 | 最大长度为8 |
| 7 | discount | int | N | N |  |  | 最大长度为11 |
| sql脚本 | | Creat table dish（  id int(10) primary key,not null,unique,  categorize varchar(10) not null ,unique,  dish\_name varchar(50) not null,  unit varchar(10) not null,  price decimal(10) not null,  status varchar(8) not null,  discount int(11) not null  ）; | | | | | |
| 备注 | |  | | | | | |

表4.6 点菜表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | ordering | | | | | |
| 数据库用户 | | root | | | | | |
| 主 键 | | id | | | | | |
| 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 | id | int | N | Y |  |  | 主码，最大长度10 |
| 2 | name | varchar | N | Y |  |  | 最大长度20 |
| 3 | table\_id | varchar | N | N |  |  | 最大长度8 |
| 4 | food\_id | varchar | N | N |  |  | 最大长度为11 |
| 5 | food\_name | varchar | N | N |  |  | 最大长度为10 |
| 6 | number | varchar | N | Y |  |  | 最大长度为4 |
| 7 | meal\_number | varchar | N | N |  |  | 最大长度为4 |
| sql脚本 | | Creat table dish（  id int(10) primary key,not null,unique,  name varchar(20) not null ,unique,  table\_id varchar(8) not null,  food\_id varchar(11) not null,  food\_name varchar(10) not null,  number varchar(4) not null,  meal\_number varchar(4) not null,  constraint fk\_table\_id foreign key(table\_id)references table(table\_id),  constraint fk\_food\_name foreign key(food\_name)references dish(dish\_name)  ）; | | | | | |
| 备注 | |  | | | | | |

表4.7 结账表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | | price | | | | | |
| 数据库用户 | | root | | | | | |
| 主 键 | | id | | | | | |
| 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 | id | int | N | Y |  |  | 主码，最大长度10 |
| 2 | food\_id | varchar | N | N |  |  | 最大长度为11 |
| 3 | name | varchar | N | Y |  |  | 最大长度20 |
| 4 | food\_name | varchar | N | N |  |  | 最大长度为10 |
| 5 | number | varchar | N | Y |  |  | 最大长度为4 |
| 6 | price | varchar | N | N |  |  | 最大长度8 |
| 7 | status | varchar | N | N |  | 未支付 | 最大长度为4 |
| sql脚本 | | Creat table dish（  id int(10) primary key,not null,unique,  food\_id varchar(11) not null,  name varchar(20) not null ,unique,  food\_name varchar(10) not null,  number varchar(4) not null,  price varchar(8) not null,  status varchar(10)  constraint fk\_food\_id foreign key(food\_id)references ordering(food\_id),  constraint fk\_name foreign key(name)references ordering(name),  constraint fk\_food\_name foreign key(food\_name)references ordering(food\_name),  constraint fk\_number foreign key(number)references ordering(number),  constraint fk\_price foreign key(price)references dish(price)  ）; | | | | | |
| 备注 | |  | | | | | |

## 4.5 安全性设计

管理员登录所用的账号密码必须为数据库里的数据，连接数据库需要数据库的用户名和密码

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

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。（3）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

（1）仅有管理员拥有权限。

表4.3 角色与权限

|  |  |  |
| --- | --- | --- |
| 角色 | 可以访问的表与列 | 操作权限 |
| USER1 | 员工管理 |  |
| 客户管理 |  |
| 餐台管理 |  |
| 菜品管理 |  |
| 点菜管理 |  |
| 结账管理 |  |
| 餐台管理 |  |
|
|

# 5 测试用例设计

表5.1 登录测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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’ | root1，123456 | 登录成功 | 通过 |
| yl-002 | root1，1234567 | 登录失败 | 通过 |
| yl-003 | root2，123456 | 登录失败 | 通过 |
| yl-004 | root2,空 | 登录失败 | 通过 |
| yl-005 | 空,1234567 | 登录失败 | 通过 |
| yl-006 | root2,一个空格 | 登录失败 | 通过 |
| yl-007 | 一个空格，123456 | 登录失败 | 通过 |
| yl-008 | root，密码长度超过32 | 登录失败 | 通过 |
| yl-009 | 用户名超过36，123456 | 登录失败 | 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id,编号,姓名，性别，出生日期，籍贯，家庭地址，学历，婚否，职位 | 1，101，江大河，男，2003-02-18，37068120020218111，山东省，烟台,本科，未婚，服务员 | 新增成功 | 通过 |
| 删除员工 | y2-002 | 点击删除按钮 | 1 | 删除成功 | 通过 |
| 修改员工 | y2-003 | 点击表格 | 101->102 | 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 | 江大河 | 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id‘姓  名’，‘性别’，‘单位，‘电话‘会员卡号’ | 1001.江大圣，男，学生，1773545665，7758522 | 新增成功 | 通过 |
| 删除客户 | y3-002 | 点击删除按钮 | 1001 | 删除成功 | 通过 |
| 修改客户 | y3-003 | 点击表格 | 男->女 | 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 | 7758522 | 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id,分类，菜品名，单位，单价，状态，折扣 | 10021，西餐，凉拌唐三藏，1，10000，1，1 | 新增成功 | 通过 |
| 删除菜品 | y4-002 | 点击删除按钮 | 10021 | 删除成功 | 通过 |
| 修改菜品 | y4-003 | 点击表格 | 西餐->中餐 | 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 | 10021 | 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 | 101 | 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 | 输入客户名和人数 | 张三，11 | 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 |
| 查询订单 | y7-001 | 输入额客户名后点击查询按钮 | 张三 | 查询成功 | 通过 |
| 结账操作 | Y7-002 | 点击结账按钮 | 100 | 结账成功 | 通过 |

# 附录

## 附录1 程序运行结果



图1 登录界面

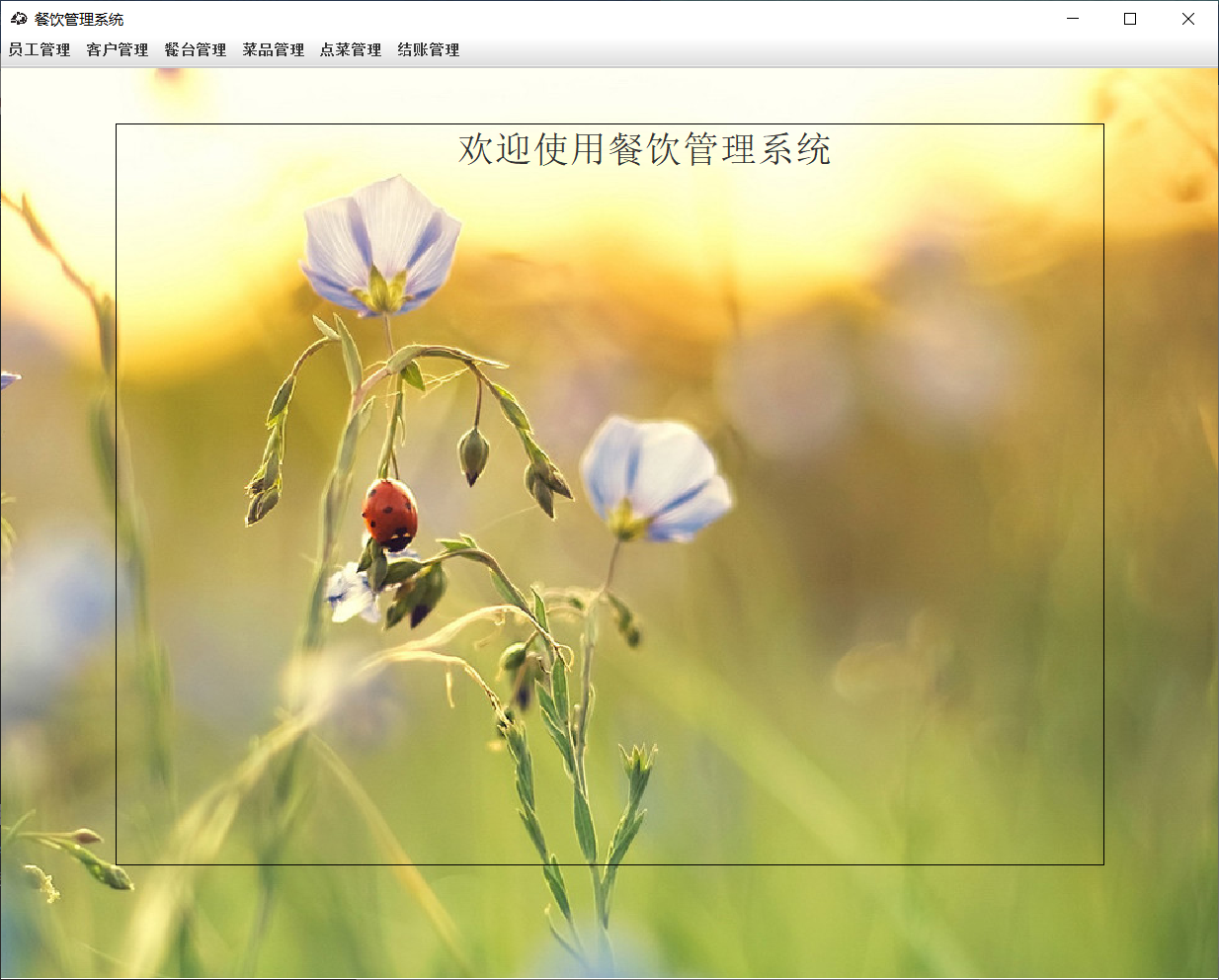


图2,主界面

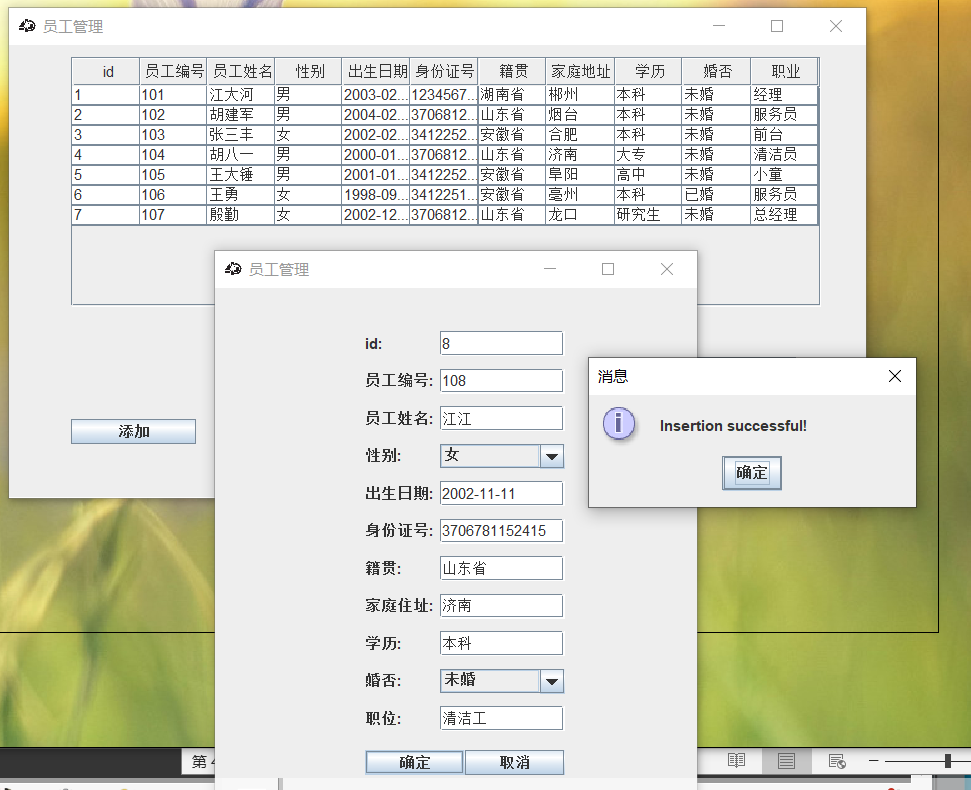


图3员工管理界面

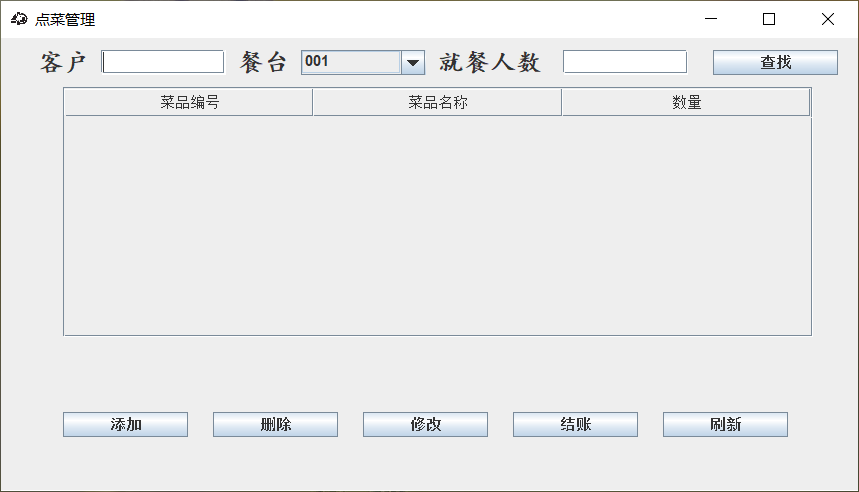
图4，客户管理界面

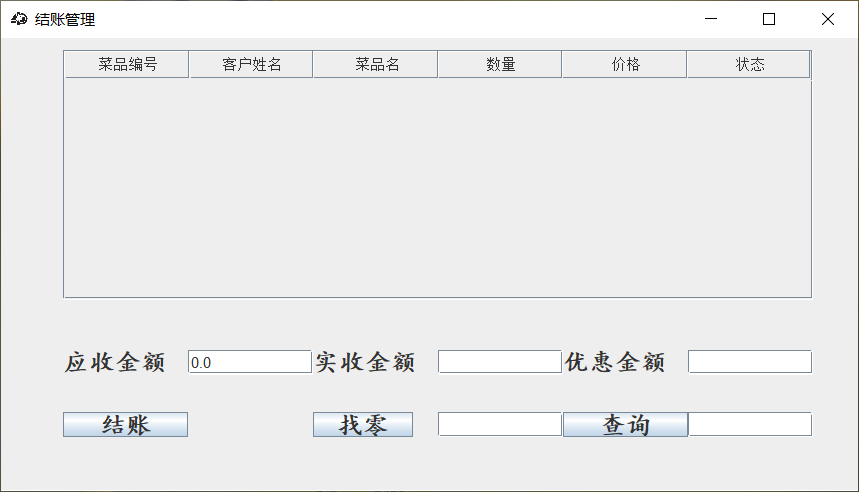


图5，餐台管理界面



图6，菜品管理界面

图7，点菜管理界面

图8，结账管理界面

## 附录2 程序源代码

登录界面：

package pack;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import javax.swing.\*;

public class dljm extends JFrame{

dljm(){

setTitle("登录");

setSize(400,300);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

JLabel l0=new JLabel("欢 迎 登 陆 餐 饮 系 统");

l0.setBounds(40,30,400,30);//设置定位

l0.setFont(new Font("黑体",Font.***BOLD***,25));//设置字体

//l0.setBackground(Color.WHITE);//面板背景色

//l0.setBorder(BorderFactory.createLineBorder(Color.BLACK));//设置边框

add(l0,BorderLayout.***NORTH***);

JLabel username=new JLabel("用户名");

username.setBounds(100,80,80,20);

username.setFont(new Font("楷体", Font.***BOLD***, 20));

add(username);

JTextField tx1=new JTextField();

tx1.setBounds(170,80,120,20);

tx1.repaint();//重绘文本框防止文本框消失

add(tx1);

JLabel password=new JLabel("密 码");

password.setBounds(100,100,100,50);

password.setFont(new Font("楷体", Font.***BOLD***, 20));

add(password);

JPasswordField tx2=new JPasswordField();//加密文本框

tx2.setBounds(170,115,120,20);

tx2.repaint();

add(tx2);

JButton bt1=new JButton("登录");

bt1.setFont(new Font("楷体", Font.***BOLD***, 15));

//bt1.setBackground(Color.LIGHT\_GRAY);//设置背景颜色

//bt1.setForeground(Color.black);//设置字体颜色

bt1.setBorder(BorderFactory.*createRaisedBevelBorder*());//将边框增加立体感，createLoweredBevelBorder方法创建一个凹陷效果的边框。

bt1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent arg0) {

// **TODO** Auto-generated method stub

String username = tx1.getText();//获取输入的账号密码

String password = new String(tx2.getPassword());//getText()方法安全性不高，可以写成getPassword该形式;

try {

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

Connection con=DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/餐饮管理","root","2272343628");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(

"select \* from entity\_1\_administrator\_information where username='" + username +

"' and password='" + password + "'");

if (rs.next()) {

JOptionPane.*showMessageDialog*(bt1, "登录成功！");

new zjm();

dispose();}//关闭界面

else {

JOptionPane.*showMessageDialog*(bt1, "用户名或密码错误！");

con.close();}

} catch(Exception e) {

System.***out***.println(e);

}

}

});

bt1.setBounds(100,200,80,20);

add(bt1);

JButton bt2=new JButton("取消");

bt2.setFont(new Font("楷体", Font.***BOLD***, 15));

bt2.setBorder(BorderFactory.*createRaisedBevelBorder*());

bt2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent arg0) {

// **TODO** Auto-generated method stub

dispose();

}

});

bt2.setBounds(210,200,80,20);

add(bt2);

ImageIcon bj1 = new ImageIcon("C:\\peixunworkspace\\picture\\22.jpg");

JLabel bgLl = new JLabel(bj1);

bgLl.setBounds(0, 0, getWidth(), getHeight());

bgLl.setHorizontalAlignment(JLabel.***CENTER***);

bgLl.setVerticalAlignment(JLabel.***CENTER***);

bgLl.setOpaque(true);

getLayeredPane().add(bgLl, new ~~Integer~~(Integer.***MIN\_VALUE***));

addComponentListener(new ComponentAdapter() {

public void componentResized(ComponentEvent e) {

bgLl.setBounds(0, 0, getWidth(), getHeight());

}

});

add(bgLl);

setLocationRelativeTo(null);//设置窗口弹出位置为中间

setVisible(true);

}

}

class test1{

public static void main(String[] args) {

new dljm();

}

}

主界面：

package pack;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class zjm extends JFrame{

zjm(){

setTitle("餐饮管理系统");

setSize(1000,800);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

JLabel l1=new JLabel("欢迎使用餐饮管理系统");

l1.setBounds(370,-10,500,150);

l1.setFont(new Font("宋体",Font.***PLAIN***,30));

add(l1);

JMenuBar menuBar = new JMenuBar();//创建菜单栏对象

setJMenuBar(menuBar);

JMenu menu0 =new JMenu("员工管理");

menuBar.add(menu0);

JMenuItem menuItemy = new JMenuItem("打开员工管理");

menuItemy.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

new employee();

}});

menu0.add(menuItemy);

JMenu menu1 =new JMenu("客户管理");

JMenuItem menuItem1= new JMenuItem("打开客户管理");

menuItem1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new Client();

}

});

menuBar.add(menu1);

menu1.add(menuItem1);

JMenu menu2 = new JMenu("餐台管理");

JMenuItem menuItem2 = new JMenuItem("打开餐台管理");

menuItem2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new Tables();

}});

menuBar.add(menu2);

menu2.add(menuItem2);

JMenu menu3 = new JMenu("菜品管理");

JMenuItem menuItem3 = new JMenuItem("打开菜品管理");

menuItem3.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new cadish();

}});

menuBar.add(menu3);

menu3.add(menuItem3);

JMenu menu4 = new JMenu("点菜管理");

menuBar.add(menu4);

JMenuItem menuItem4 = new JMenuItem("打开点菜管理");

menuItem4.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new order();

}

});

menu4.add(menuItem4);

JMenu menu5 = new JMenu("结账管理");

JMenuItem menItem6=new JMenuItem("打开结账管理");

menItem6.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new jiezhang(null);

}

});

menuBar.add(menu5);

menu5.add(menItem6);

ImageIcon bj1 = new ImageIcon("C:\\peixunworkspace\\picture\\123.jpg");

JLabel bgLl = new JLabel(bj1);

bgLl.setBounds(0, 0, getWidth(), getHeight());

getLayeredPane().add(bgLl, new ~~Integer~~(Integer.***MIN\_VALUE***));

addComponentListener(new ComponentAdapter() {

public void componentResized(ComponentEvent e) {

bgLl.setBounds(0, 0, getWidth(), getHeight());

}

});

add(bgLl);

setLocationRelativeTo(null);

setVisible(true);

}

public void paint(Graphics g) {//在界面中绘入边框

super.paint(g);

g.setColor(Color.***BLACK***);

g.drawRect(100, 100, 800, 600);

}

}

class zjm1{

public static void main(String[] args) {

new zjm();

}

}

员工管理界面：

package pack;

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.sql.\*;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class employee extends JFrame{

private JTable table;

private DefaultTableModel tableModel;

employee(){

setTitle("员工管理");

setSize(700,400);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

// Initialize the table model.

tableModel = new DefaultTableModel();

// Add columns to the table model.

tableModel.addColumn("id");

tableModel.addColumn("员工编号");

tableModel.addColumn("员工姓名");

tableModel.addColumn("性别");

tableModel.addColumn("出生日期");

tableModel.addColumn("身份证号");

tableModel.addColumn("籍贯");

tableModel.addColumn("家庭地址");

tableModel.addColumn("学历");

tableModel.addColumn("婚否");

tableModel.addColumn("职业");

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

String sql = "SELECT \* FROM employee\_information";

PreparedStatement statement = conn.prepareStatement(sql);

ResultSet resultSet = statement.executeQuery();

while (resultSet.next()) {

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = resultSet.getString(1);

row[1] = resultSet.getString(2);

row[2] = resultSet.getString(3);

row[3] = resultSet.getString(4);

row[4] = resultSet.getString(5);

row[5] = resultSet.getString(6);

row[6] = resultSet.getString(7);

row[7] = resultSet.getString(8);

row[8] = resultSet.getString(9);

row[9] = resultSet.getString(10);

row[10] = resultSet.getString(11);

tableModel.addRow(row);

}

} catch (SQLException ex) {

ex.printStackTrace();

}

table = new JTable(tableModel);

JScrollPane scrollPane = new JScrollPane(table);//添加滚动边框

scrollPane.setBounds(50,10,600,200);

scrollPane.repaint();

add(scrollPane);

JButton bt1=new JButton("添加");

bt1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new empadd();

}});

bt1.setBounds(50,300,100,20);

add(bt1);

//获得鼠标左键选中行

table.addMouseListener(new MouseAdapter() {

*@Override*

public void mousePressed(MouseEvent e) {

if (e.getButton() == MouseEvent.***BUTTON1***) { // Check if left mouse button was clicked.

int col = table.columnAtPoint(e.getPoint());

int row = table.rowAtPoint(e.getPoint());

if (row >= 0 && col >= 0) { // Check if row and column indices are valid.

}

}

}

});

JButton bt2=new JButton("删除");

bt2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int row = table.getSelectedRow();

if (row >= 0) {

deleteRow(row);

}

}

});

bt2.setBounds(170,300,100,20);

add(bt2);

JButton bt3=new JButton("修改");

bt3.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int selectedRow = table.getSelectedRow();

if (selectedRow >= 0) {

updateRow(selectedRow);

}

}

});

bt3.setBounds(290,300,100,20);

add(bt3);

JTextField searchField =new JTextField();

searchField.setBounds(530,300,100,20);

add(searchField);

JButton bt4=new JButton("查找");

bt4.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

if (e.getSource() == bt4) {

try {

// 连接到数据库并准备查询语句

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8", "root", "2272343628");

String searchTerm = searchField.getText();

String query = "SELECT \* FROM employee\_information where id = ? OR emp\_id = ? OR emp\_name like ? OR sex like ? OR native\_place like ? OR home\_address like ? OR education like ? OR marriage like ? OR position like ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1,searchTerm );

stmt.setString(2,searchTerm );

stmt.setString(3,searchTerm );

stmt.setString(4,searchTerm );

stmt.setString(5,searchTerm );

stmt.setString(6,searchTerm );

stmt.setString(7,searchTerm );

stmt.setString(8,searchTerm );

stmt.setString(9,searchTerm );

// 执行查询，并将结果显示在控制台上

ResultSet rs = stmt.executeQuery();

tableModel.getDataVector().removeAllElements();

tableModel.fireTableDataChanged();

table.removeAll();

while (rs.next()) {

//先清空表格

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = rs.getString(1);

row[1] = rs.getString(2);

row[2] = rs.getString(3);

row[3] = rs.getString(4);

row[4] = rs.getString(5);

row[5] = rs.getString(6);

row[6] = rs.getString(7);

row[7] = rs.getString(8);

row[8] = rs.getString(9);

row[9] = rs.getString(10);

row[10] = rs.getString(11);

tableModel.addRow(row);

}

// 关闭资源

rs.close();

stmt.close();

conn.close();

searchField.setText("");

} catch (SQLException ex) {

ex.printStackTrace();

}

}}

});

bt4.setBounds(410,300,100,20);

add(bt4);

JButton bt5 = new JButton("刷新");

bt5.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

new employee();

}

});

bt5.setBounds(530,250,100,20);

add(bt5);

setLocationRelativeTo(null);

setVisible(true);

}

private void deleteRow(int rowIndex) {

try {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理";

String username = "root";

String password = "2272343628";

Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password);

// Delete the row with the given ID from the 'mytable' table.

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object id = tableModel.getValueAt(rowIndex, idColumnIndex); // Get the ID of the row to delete.

String sql = "DELETE FROM employee\_information WHERE id = ?";

PreparedStatement statement = conn.prepareStatement(sql);

statement.setObject(1, id); // Set the value of the primary key parameter in the query.

statement.executeUpdate();

// Remove the row from the table model.

tableModel.removeRow(rowIndex);

conn.close(); // Close the connection.

JOptionPane.*showMessageDialog*(table, "delete successful!");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void updateRow(final int rowIndex) {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

// Create dialog to edit row data.

Object[] oldRowData = new Object[tableModel.getColumnCount()];

for (int i = 0; i < oldRowData.length; i++) {

oldRowData[i] = tableModel.getValueAt(rowIndex, i);

}

Object[] newRowData = showEditRowDialog(oldRowData);

if (newRowData != null) { // Check if user clicked the "OK" button.

// generate and execute UPDATE statement

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object id = newRowData[idColumnIndex];

StringBuilder sqlBuilder = new StringBuilder("UPDATE employee\_information SET ");

sqlBuilder.append("emp\_id = ?, ");

sqlBuilder.append("emp\_name = ?, ");

sqlBuilder.append("sex = ?, ");

sqlBuilder.append("birthday = ?, ");

sqlBuilder.append("ID\_card= ? ,");

sqlBuilder.append("native\_place = ?, ");

sqlBuilder.append("home\_address = ?, ");

sqlBuilder.append("education = ?, ");

sqlBuilder.append("marriage = ?, ");

sqlBuilder.append("position = ?");

sqlBuilder.append("WHERE id = ?");

String sql = sqlBuilder.toString();

PreparedStatement statement = conn.prepareStatement(sql);

for (int i = 1; i < tableModel.getColumnCount(); i++) {

statement.setObject(i, newRowData[i]);

}

statement.setObject(tableModel.getColumnCount(), id);

statement.executeUpdate();

// Update the table model with the new row data.

for (int i = 0; i < newRowData.length; i++) {

tableModel.setValueAt(newRowData[i], rowIndex, i);

}

JOptionPane.*showMessageDialog*(table, "update successful!");

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private Object[] showEditRowDialog(Object[] oldRowData) {

// Create a dialog with input fields filled with the old data.

JComponent[] components = new JComponent[oldRowData.length];

for (int i = 0; i < oldRowData.length; i++) {

components[i] = new JTextField(oldRowData[i].toString());

}

int result = JOptionPane.*showConfirmDialog*(null, components,

"修改", JOptionPane.***OK\_CANCEL\_OPTION***);

if (result == JOptionPane.***OK\_OPTION***) {

// Retrieve the modified values from the input fields.

Object[] newRowData = new Object[oldRowData.length];

for (int i = 0; i < newRowData.length; i++) {

String text = ((JTextField) components[i]).getText();

newRowData[i] = text;

}

return newRowData;

} else {

return null;

}

}

}

class emp{

public static void main(String[] args) {

SwingUtilities.*invokeLater*(new Runnable() {

*@Override*

public void run() {

new employee();

}

});

}

}package pack;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.Window;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.\*;

public class empadd extends JFrame{

String text1 ;

String text2 ;

String text3;

String text4;

String text5 ;

String text6;

String text7;

String text8;

String text9;

String text10;

String text11;

empadd(){

setTitle("员工管理");

setSize(400,500);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

JLabel c1=new JLabel("id:");

c1.setBounds(120,20,105,50);

add(c1);

JTextField tx1=new JTextField();

tx1.setBounds(180,35,100,20);

add(tx1);

JLabel c2=new JLabel("员工编号:");

c2.setBounds(120,50,105,50);

add(c2);

JTextField tx2=new JTextField();

tx2.setBounds(180,65,100,20);

add(tx2);

JLabel c3=new JLabel("员工姓名:");

c3.setBounds(120,80,105,50);

add(c3);

JTextField tx3=new JTextField();

tx3.setBounds(180,95,100,20);

add(tx3);

JLabel c4=new JLabel("性别:");

c4.setBounds(120,110,105,50);

add(c4);

JComboBox tx4 = new JComboBox();//添加下拉框

tx4.addItem(" ");//给下拉框添加内容

tx4.addItem("男");

tx4.addItem("女");

tx4.setBounds(180,125,100,20);

add(tx4);

JLabel c5=new JLabel("出生日期:");

c5.setBounds(120,140,105,50);

add(c5);

JTextField tx5=new JTextField();

tx5.setBounds(180,155,100,20);

add(tx5);

JLabel c6=new JLabel("身份证号:");

c6.setBounds(120,170,105,50);

add(c6);

JTextField tx6=new JTextField();

tx6.setBounds(180,185,100,20);

add(tx6);

JLabel c7=new JLabel("籍贯:");

c7.setBounds(120,200,105,50);

add(c7);

JTextField tx7=new JTextField();

tx7.setBounds(180,215,100,20);

add(tx7);

JLabel c8=new JLabel("家庭住址:");

c8.setBounds(120,230,105,50);

add(c8);

JTextField tx8=new JTextField();

tx8.setBounds(180,245,100,20);

add(tx8);

JLabel c9=new JLabel("学历:");

c9.setBounds(120,260,105,50);

add(c9);

JTextField tx9=new JTextField();

tx9.setBounds(180,275,100,20);

add(tx9);

JLabel c10=new JLabel("婚否:");

c10.setBounds(120,290,105,50);

add(c10);

JComboBox tx10 = new JComboBox();//添加下拉框

tx10.addItem(" ");//给下拉框添加内容

tx10.addItem("已婚");

tx10.addItem("未婚");

tx10.setBounds(180,305,100,20);

add(tx10);

JLabel c11=new JLabel("职位:");

c11.setBounds(120,320,105,50);

add(c11);

JTextField tx11=new JTextField();

tx11.setBounds(180,335,100,20);

add(tx11);

JButton b1=new JButton("确定");

b1.setBounds(120,370,80,20);

b1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

try {

text1 = tx1.getText();

text2 = tx2.getText();

text3 = tx3.getText();

text4 = (String) tx4.getSelectedItem();

text5 = tx5.getText();

text6 = tx6.getText();

text7 = tx7.getText();

text8 = tx8.getText();

text9 = tx9.getText();

text10 = (String) tx10.getSelectedItem();

text11 = tx11.getText();

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

Connection con=DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8"

,"root","2272343628");

Statement stmt = con.createStatement();

String sql = "INSERT INTO employee\_information VALUES (?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, text1);

pstmt.setString(2, text2);

pstmt.setString(3, text3);

pstmt.setString(4, text4);

pstmt.setString(5, text5);

pstmt.setString(6, text6);

pstmt.setString(7, text7);

pstmt.setString(8, text8);

pstmt.setString(9, text9);

pstmt.setString(10, text10);

pstmt.setString(11, text11);

pstmt.executeUpdate();

// 输出成功消息

JOptionPane.*showMessageDialog*(b1, "Insertion successful!");

if(e.getSource() == b1) {

Window[] windows = Window.*getWindows*();

for(Window window : windows) {

if(window instanceof employee) {

window.dispose();

}

}

}

employee e1 =new employee();

e1.repaint();

dispose();

new empadd();

con.close();

} catch (NumberFormatException ex) {

// 处理用户输入无效数据的情况

JOptionPane.*showMessageDialog*(b1, "Invalid age value!", "Error", JOptionPane.***ERROR\_MESSAGE***);

} catch (SQLException ex) {

// 处理数据库插入错误

JOptionPane.*showMessageDialog*(b1, "SQL Exception: " + ex.getMessage(), "Error", JOptionPane.***ERROR\_MESSAGE***);

}

catch (ClassNotFoundException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

});

add(b1);

JButton b2=new JButton("取消");

b2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

}

});

b2.setBounds(200,370,80,20);

add(b2);

setLocationRelativeTo(null);

setVisible(true);

}

}

class empa{

public static void main(String[] args) {

new empadd();

}

客户管理界面：

package pack;

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.sql.\*;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class Client extends JFrame{

private JTable table;

private DefaultTableModel tableModel;

Client(){

setTitle("客户管理");

setSize(700,400);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

// Initialize the table model.

tableModel = new DefaultTableModel();

// Add columns to the table model.

tableModel.addColumn("id");

tableModel.addColumn("用户名");

tableModel.addColumn("性别");

tableModel.addColumn("单位");

tableModel.addColumn("电话");

tableModel.addColumn("贵宾卡号");

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

String sql = "SELECT \* FROM client\_information";

PreparedStatement statement = conn.prepareStatement(sql);

ResultSet resultSet = statement.executeQuery();

while (resultSet.next()) {

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = resultSet.getString(1);

row[1] = resultSet.getString(2);

row[2] = resultSet.getString(3);

row[3] = resultSet.getString(4);

row[4] = resultSet.getString(5);

row[5] = resultSet.getString(6);

tableModel.addRow(row);

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

table = new JTable(tableModel);

JScrollPane scrollPane = new JScrollPane(table);//添加滚动边框

scrollPane.setBounds(50,10,600,200);

scrollPane.repaint();

add(scrollPane);

JButton bt1=new JButton("添加");

bt1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new Cliadd();

}});

bt1.setBounds(50,300,100,20);

add(bt1);

//获得鼠标左键选中行

table.addMouseListener(new MouseAdapter() {

*@Override*

public void mousePressed(MouseEvent e) {

if (e.getButton() == MouseEvent.***BUTTON1***) { // Check if left mouse button was clicked.

int col = table.columnAtPoint(e.getPoint());

int row = table.rowAtPoint(e.getPoint());

if (row >= 0 && col >= 0) { // Check if row and column indices are valid.

}

}

}

});

JButton bt2=new JButton("删除");

bt2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int row = table.getSelectedRow();

if (row >= 0) {

deleteRow(row);

}

}

});

bt2.setBounds(170,300,100,20);

add(bt2);

JButton bt3=new JButton("修改");

bt3.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int selectedRow = table.getSelectedRow();

if (selectedRow >= 0) {

updateRow(selectedRow);

}

}

});

bt3.setBounds(290,300,100,20);

add(bt3);

JTextField searchField =new JTextField();

searchField.setBounds(530,300,100,20);

add(searchField);

JButton bt4=new JButton("查找");

bt4.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

if (e.getSource() == bt4) {

try {

// 连接到数据库并准备查询语句

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8", "root", "2272343628");

String searchTerm = searchField.getText();

String query = "SELECT \* FROM client\_information WHERE id =? or name like ? or sex like ? or company like ? or tel like ? or cardID like ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1,searchTerm );

stmt.setString(2,searchTerm );

stmt.setString(3,searchTerm );

stmt.setString(4,searchTerm );

stmt.setString(5,searchTerm );

stmt.setString(6,searchTerm );

// 执行查询，并将结果显示在控制台上

ResultSet rs = stmt.executeQuery();

tableModel.getDataVector().removeAllElements();

tableModel.fireTableDataChanged();

table.removeAll();

while (rs.next()) {

//先清空表格

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = rs.getString(1);

row[1] = rs.getString(2);

row[2] = rs.getString(3);

row[3] = rs.getString(4);

row[4] = rs.getString(5);

row[5] = rs.getString(6);

tableModel.addRow(row);

}

// 关闭资源

rs.close();

stmt.close();

conn.close();

searchField.setText("");

} catch (SQLException ex) {

ex.printStackTrace();

}

}}

});

bt4.setBounds(410,300,100,20);

add(bt4);

JButton bt5 = new JButton("刷新");

bt5.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

new Client();

}

});

bt5.setBounds(530,250,100,20);

add(bt5);

setLocationRelativeTo(null);

setVisible(true);

}

private void deleteRow(int rowIndex) {

try {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理";

String username = "root";

String password = "2272343628";

Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password);

// Delete the row with the given ID from the 'mytable' table.

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object id = tableModel.getValueAt(rowIndex, idColumnIndex); // Get the ID of the row to delete.

String sql = "DELETE FROM client\_information WHERE id = ?";

PreparedStatement statement = conn.prepareStatement(sql);

statement.setObject(1, id); // Set the value of the primary key parameter in the query.

statement.executeUpdate();

// Remove the row from the table model.

tableModel.removeRow(rowIndex);

conn.close(); // Close the connection.

JOptionPane.*showMessageDialog*(table, "delete successful!");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void updateRow(final int rowIndex) {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

// Create dialog to edit row data.

Object[] oldRowData = new Object[tableModel.getColumnCount()];

for (int i = 0; i < oldRowData.length; i++) {

oldRowData[i] = tableModel.getValueAt(rowIndex, i);

}

Object[] newRowData = showEditRowDialog(oldRowData);

if (newRowData != null) { // Check if user clicked the "OK" button.

// generate and execute UPDATE statement

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object id = newRowData[idColumnIndex];

StringBuilder sqlBuilder = new StringBuilder("UPDATE client\_information SET ");

sqlBuilder.append("name = ?, ");

sqlBuilder.append("sex = ?, ");

sqlBuilder.append("company = ?, ");

sqlBuilder.append("tel = ?, ");

sqlBuilder.append("cardID = ?");

sqlBuilder.append("WHERE id = ?");

String sql = sqlBuilder.toString();

PreparedStatement statement = conn.prepareStatement(sql);

for (int i = 1; i < tableModel.getColumnCount(); i++) {

statement.setObject(i, newRowData[i]);

}

statement.setObject(tableModel.getColumnCount(), id);

statement.executeUpdate();

for (int i = 0; i < newRowData.length; i++) {

tableModel.setValueAt(newRowData[i], rowIndex, i);

}

JOptionPane.*showMessageDialog*(table, "update successful!");

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private Object[] showEditRowDialog(Object[] oldRowData) {

// Create a dialog with input fields filled with the old data.

JComponent[] components = new JComponent[oldRowData.length];

for (int i = 0; i < oldRowData.length; i++) {

components[i] = new JTextField(oldRowData[i].toString());

}

int result = JOptionPane.*showConfirmDialog*(null, components,

"修改", JOptionPane.***OK\_CANCEL\_OPTION***);

if (result == JOptionPane.***OK\_OPTION***) {

// Retrieve the modified values from the input fields.

Object[] newRowData = new Object[oldRowData.length];

for (int i = 0; i < newRowData.length; i++) {

String text = ((JTextField) components[i]).getText();

newRowData[i] = text;

}

return newRowData;

} else {

return null;

}

}

}

class Cli{

public static void main(String[] args) {

SwingUtilities.*invokeLater*(new Runnable() {

*@Override*

public void run() {

new Client();

}

});

}

}

package pack;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.Window;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.\*;

public class Cliadd extends JFrame{

String text1;

String text2 ;

String text3;

String text4;

String text5;

String text6;

Cliadd(){

setTitle("客户管理");

setSize(400,410);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

JLabel c1=new JLabel("id:");

c1.setBounds(120,-30,105,150);

add(c1);

JTextField tx1=new JTextField(1);

tx1.setBounds(180,35,100,20);

add(tx1);

JLabel c2=new JLabel("用户名:");

c2.setBounds(120,0,105,150);

add(c2);

JTextField tx2=new JTextField();

tx2.setBounds(180,65,100,20);

add(tx2);

JLabel c3=new JLabel("性别:");

c3.setBounds(120,30,105,150);

add(c3);

JTextField tx3=new JTextField();

tx3.setBounds(180,95,100,20);

add(tx3);

JLabel c4=new JLabel("单位:");

c4.setBounds(120,60,105,150);

add(c4);

JTextField tx4=new JTextField();

tx4.setBounds(180,125,100,20);

add(tx4);

JLabel c5=new JLabel("电话:");

c5.setBounds(120,90,105,150);

add(c5);

JTextField tx5=new JTextField();

tx5.setBounds(180,155,100,20);

add(tx5);

JLabel c6=new JLabel("贵宾卡号:");

c6.setBounds(120,120,105,150);

add(c6);

JTextField tx6=new JTextField();

tx6.setBounds(180,185,100,20);

add(tx6);

JButton b1=new JButton("确定");

b1.setBounds(120,280,80,20);

b1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

try {

text1 = tx1.getText();

text2 = tx2.getText();

text3 = tx3.getText();

text4 = tx4.getText();

text5 = tx5.getText();

text6 = tx6.getText();

//

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

Connection con=DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8"

,"root","2272343628");

Statement stmt = con.createStatement();

String sql = "INSERT INTO client\_information VALUES (?,?,?,?,?,?)";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, text1);

pstmt.setString(2, text2);

pstmt.setString(3, text3);

pstmt.setString(4, text4);

pstmt.setString(5, text5);

pstmt.setString(6, text6);

pstmt.executeUpdate();

// 输出成功消息

JOptionPane.*showMessageDialog*(b1, "Insertion successful!");

if(e.getSource() == b1) {

Window[] windows = Window.*getWindows*();//获取所有活动窗口，并检查每个窗口是否为MyFrame1。

for(Window window : windows) {

if(window instanceof Client) {

window.dispose();

}

}

}

Client e1 =new Client();

e1.repaint();

dispose();

new Cliadd();

con.close();

} catch (NumberFormatException ex) {

// 处理用户输入无效数据的情况

JOptionPane.*showMessageDialog*(b1, "Invalid age value!", "Error", JOptionPane.***ERROR\_MESSAGE***);

} catch (SQLException ex) {

// 处理数据库插入错误

JOptionPane.*showMessageDialog*(b1, "SQL Exception: " + ex.getMessage(), "Error", JOptionPane.***ERROR\_MESSAGE***);

}

catch (ClassNotFoundException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

});

add(b1);

JButton b2=new JButton("取消");

b2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

}

});

b2.setBounds(200,280,80,20);

add(b2);

setLocationRelativeTo(null);

setVisible(true);

}

}

class Cla{

public static void main(String[] args) {

new Tabadd();

}

}

餐台管理：

package pack;

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.sql.\*;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class Tables extends JFrame{

private JTable table;

private DefaultTableModel tableModel;

Tables(){

setTitle("餐台管理");

setSize(700,400);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

// Initialize the table model.

tableModel = new DefaultTableModel();

// Add columns to the table model.

tableModel.addColumn("id");

tableModel.addColumn("餐台编号");

tableModel.addColumn("座位数");

tableModel.addColumn("状态");

tableModel.addColumn("开台时间");

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

String sql = "SELECT \* FROM table\_information";

PreparedStatement statement = conn.prepareStatement(sql);

ResultSet resultSet = statement.executeQuery();

while (resultSet.next()) {

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = resultSet.getString(1);

row[1] = resultSet.getString(2);

row[2] = resultSet.getString(3);

row[3] = resultSet.getString(4);

row[4] = resultSet.getString(5);

tableModel.addRow(row);

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

table = new JTable(tableModel);

JScrollPane scrollPane = new JScrollPane(table);//添加滚动边框

scrollPane.setBounds(50,10,600,200);

scrollPane.repaint();

add(scrollPane);

JButton bt1=new JButton("添加");

bt1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new Tabadd();

}});

bt1.setBounds(50,300,100,20);

add(bt1);

//获得鼠标左键选中行

table.addMouseListener(new MouseAdapter() {

*@Override*

public void mousePressed(MouseEvent e) {

if (e.getButton() == MouseEvent.***BUTTON1***) { // Check if left mouse button was clicked.

int col = table.columnAtPoint(e.getPoint());

int row = table.rowAtPoint(e.getPoint());

if (row >= 0 && col >= 0) { // Check if row and column indices are valid.

}

}

}

});

JButton bt2=new JButton("删除");

bt2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int row = table.getSelectedRow();

if (row >= 0) {

deleteRow(row);

}

}

});

bt2.setBounds(170,300,100,20);

add(bt2);

JButton bt3=new JButton("修改");

bt3.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int selectedRow = table.getSelectedRow();

if (selectedRow >= 0) {

updateRow(selectedRow);

}

}

});

bt3.setBounds(290,300,100,20);

add(bt3);

JTextField searchField =new JTextField();

searchField.setBounds(530,300,100,20);

add(searchField);

JButton bt4=new JButton("查找");

bt4.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

if (e.getSource() == bt4) {

try {

// 连接到数据库并准备查询语句

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8", "root", "2272343628");

String searchTerm = searchField.getText();

String query = "SELECT \* FROM table\_information WHERE id =? or table\_id = ? or table\_num like ? or status like ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1,searchTerm );

stmt.setString(2,searchTerm );

stmt.setString(3,searchTerm );

stmt.setString(4,searchTerm );

// 执行查询，并将结果显示在控制台上

ResultSet rs = stmt.executeQuery();

tableModel.getDataVector().removeAllElements();

tableModel.fireTableDataChanged();

table.removeAll();

while (rs.next()) {

//先清空表格

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = rs.getString(1);

row[1] = rs.getString(2);

row[2] = rs.getString(3);

row[3] = rs.getString(4);

row[4] = rs.getString(5);

tableModel.addRow(row);

}

// 关闭资源

rs.close();

stmt.close();

conn.close();

searchField.setText("");

} catch (SQLException ex) {

ex.printStackTrace();

}

}}

});

bt4.setBounds(410,300,100,20);

add(bt4);

JButton bt5 = new JButton("刷新");

bt5.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

new Tables();

}

});

bt5.setBounds(530,250,100,20);

add(bt5);

setLocationRelativeTo(null);

setVisible(true);

}

private void deleteRow(int rowIndex) {

try {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理";

String username = "root";

String password = "2272343628";

Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password);

// Delete the row with the given ID from the 'mytable' table.

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object id = tableModel.getValueAt(rowIndex, idColumnIndex); // Get the ID of the row to delete.

String sql = "DELETE FROM table\_information WHERE id = ?";

PreparedStatement statement = conn.prepareStatement(sql);

statement.setObject(1, id); // Set the value of the primary key parameter in the query.

statement.executeUpdate();

// Remove the row from the table model.

tableModel.removeRow(rowIndex);

conn.close(); // Close the connection.

JOptionPane.*showMessageDialog*(table, "delete successful!");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void updateRow(final int rowIndex) {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

// Create dialog to edit row data.

Object[] oldRowData = new Object[tableModel.getColumnCount()];

for (int i = 0; i < oldRowData.length; i++) {

oldRowData[i] = tableModel.getValueAt(rowIndex, i);

}

Object[] newRowData = showEditRowDialog(oldRowData);

if (newRowData != null) { // Check if user clicked the "OK" button.

// generate and execute UPDATE statement

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object id = newRowData[idColumnIndex];

StringBuilder sqlBuilder = new StringBuilder("UPDATE table\_information SET ");

sqlBuilder.append("table\_id = ?, ");

sqlBuilder.append("table\_num = ?, ");

sqlBuilder.append("status = ?,");

sqlBuilder.append("open\_time = ?");

sqlBuilder.append("WHERE id = ?");

String sql = sqlBuilder.toString();

PreparedStatement statement = conn.prepareStatement(sql);

for (int i = 1; i < tableModel.getColumnCount(); i++) {

statement.setObject(i, newRowData[i]);

}

statement.setObject(tableModel.getColumnCount(), id);

statement.executeUpdate();

for (int i = 0; i < newRowData.length; i++) {

tableModel.setValueAt(newRowData[i], rowIndex, i);

}

JOptionPane.*showMessageDialog*(table, "update successful!");

conn.close();

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private Object[] showEditRowDialog(Object[] oldRowData) {

// Create a dialog with input fields filled with the old data.

JComponent[] components = new JComponent[oldRowData.length];

for (int i = 0; i < oldRowData.length; i++) {

components[i] = new JTextField(oldRowData[i].toString());

}

int result = JOptionPane.*showConfirmDialog*(null, components,

"修改", JOptionPane.***OK\_CANCEL\_OPTION***);

if (result == JOptionPane.***OK\_OPTION***) {

// Retrieve the modified values from the input fields.

Object[] newRowData = new Object[oldRowData.length];

for (int i = 0; i < newRowData.length; i++) {

String text = ((JTextField) components[i]).getText();

newRowData[i] = text;

}

return newRowData;

} else {

return null;

}

}

}

class Tab{

public static void main(String[] args) {

SwingUtilities.*invokeLater*(new Runnable() {

*@Override*

public void run() {

new Tables();

}

});

}

}package pack;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.Window;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.\*;

public class Tabadd extends JFrame{

String text1;

String text2 ;

String text3;

String text4;

String text5;

Tabadd(){

setTitle("餐台管理");

setSize(400,410);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

JLabel c1=new JLabel("id:");

c1.setBounds(120,-30,105,150);

add(c1);

JTextField tx1=new JTextField(1);

tx1.setBounds(180,35,100,20);

add(tx1);

JLabel c2=new JLabel("餐台编号:");

c2.setBounds(120,0,105,150);

add(c2);

JTextField tx2=new JTextField();

tx2.setBounds(180,65,100,20);

add(tx2);

JLabel c3=new JLabel("座位数:");

c3.setBounds(120,30,105,150);

add(c3);

JTextField tx3=new JTextField();

tx3.setBounds(180,95,100,20);

add(tx3);

JLabel c4=new JLabel("状态:");

c4.setBounds(120,60,105,150);

add(c4);

JTextField tx4=new JTextField();

tx4.setBounds(180,125,100,20);

add(tx4);

JLabel c5=new JLabel("开台时间:");

c5.setBounds(120,90,105,150);

add(c5);

JTextField tx5=new JTextField();

tx5.setBounds(180,155,100,20);

add(tx5);

JButton b1=new JButton("确定");

b1.setBounds(120,280,80,20);

b1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

try {

text1 = tx1.getText();

text2 = tx2.getText();

text3 = tx3.getText();

text4 = tx4.getText();

text5 = tx5.getText();

//

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

Connection con=DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8"

,"root","2272343628");

Statement stmt = con.createStatement();

String sql = "INSERT INTO table\_information VALUES (?,?,?,?,?)";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, text1);

pstmt.setString(2, text2);

pstmt.setString(3, text3);

pstmt.setString(4, text4);

pstmt.setString(5, text5);

pstmt.executeUpdate();

// 输出成功消息

JOptionPane.*showMessageDialog*(b1, "Insertion successful!");

if(e.getSource() == b1) {

Window[] windows = Window.*getWindows*();//获取所有活动窗口，并检查每个窗口是否为MyFrame1。

for(Window window : windows) {

if(window instanceof Tables) {

window.dispose();

}

}

}

Tables e1 =new Tables();

e1.repaint();

dispose();

new Tabadd();

con.close();

} catch (NumberFormatException ex) {

// 处理用户输入无效数据的情况

JOptionPane.*showMessageDialog*(b1, "Invalid age value!", "Error", JOptionPane.***ERROR\_MESSAGE***);

} catch (SQLException ex) {

// 处理数据库插入错误

JOptionPane.*showMessageDialog*(b1, "SQL Exception: " + ex.getMessage(), "Error", JOptionPane.***ERROR\_MESSAGE***);

}

catch (ClassNotFoundException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

});

add(b1);

JButton b2=new JButton("取消");

b2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

}

});

b2.setBounds(200,280,80,20);

add(b2);

setLocationRelativeTo(null);

setVisible(true);

}

}

class Taba{

public static void main(String[] args) {

new Tabadd();

}

}菜品管理界面：

package pack;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ComponentAdapter;

import java.awt.event.ComponentEvent;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.\*;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class cadish extends JFrame{

private JTable table;

private DefaultTableModel tableModel;

cadish(){

setTitle("菜品管理");

setSize(700,400);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

// Initialize the table model.

tableModel = new DefaultTableModel();

// Add columns to the table model.

tableModel.addColumn("id");

tableModel.addColumn("菜品分类");

tableModel.addColumn("菜品名");

tableModel.addColumn("单位");

tableModel.addColumn("单价");

tableModel.addColumn("状态");

tableModel.addColumn("折扣");

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

String sql = "SELECT \* FROM dish\_information";

PreparedStatement statement = conn.prepareStatement(sql);

ResultSet resultSet = statement.executeQuery();

while (resultSet.next()) {

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = resultSet.getString(1);

row[1] = resultSet.getString(2);

row[2] = resultSet.getString(3);

row[3] = resultSet.getString(4);

row[4] = resultSet.getString(5);

row[5] = resultSet.getString(6);

row[6] = resultSet.getString(7);

tableModel.addRow(row);

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

table = new JTable(tableModel);

JScrollPane scrollPane = new JScrollPane(table);//添加滚动边框

scrollPane.setBounds(50,10,600,200);

scrollPane.repaint();

add(scrollPane);

JButton bt1=new JButton("添加");

bt1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

new dish();

table.invalidate();

}});

bt1.setBounds(50,300,100,20);

add(bt1);

//获得鼠标左键选中行

table.addMouseListener(new MouseAdapter() {

*@Override*

public void mousePressed(MouseEvent e) {

if (e.getButton() == MouseEvent.***BUTTON1***) { // Check if left mouse button was clicked.

int col = table.columnAtPoint(e.getPoint());

int row = table.rowAtPoint(e.getPoint());

if (row >= 0 && col >= 0) { // Check if row and column indices are valid.

}

}

}

});

JButton bt2=new JButton("删除");

bt2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int row = table.getSelectedRow();

if (row >= 0) {

deleteRow(row);

}

}

});

bt2.setBounds(170,300,100,20);

add(bt2);

JButton bt3=new JButton("修改");

bt3.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int selectedRow = table.getSelectedRow();

if (selectedRow >= 0) {

updateRow(selectedRow);

}

}

});

bt3.setBounds(290,300,100,20);

add(bt3);

JTextField searchField =new JTextField();

searchField.setBounds(530,300,100,20);

add(searchField);

JButton bt4=new JButton("查找");

bt4.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

if (e.getSource() == bt4) {

try {

// 连接到数据库并准备查询语句

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8", "root", "2272343628");

String searchTerm = searchField.getText();

String query = "SELECT \* FROM dish\_information WHERE id =? OR dish\_name like ? OR categorize like ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1,searchTerm );

stmt.setString(2,searchTerm );

stmt.setString(3,searchTerm );

// 执行查询，并将结果显示在控制台上

ResultSet rs = stmt.executeQuery();

//先清空表格

tableModel.getDataVector().removeAllElements();

tableModel.fireTableDataChanged();

table.removeAll();

while (rs.next()) {

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = rs.getString(1);

row[1] = rs.getString(2);

row[2] = rs.getString(3);

row[3] = rs.getString(4);

row[4] = rs.getString(5);

row[5] = rs.getString(6);

row[6] = rs.getString(7);

tableModel.addRow(row);

}

// 关闭资源

rs.close();

stmt.close();

conn.close();

searchField.setText("");

} catch (SQLException ex) {

ex.printStackTrace();

}

}}

});

bt4.setBounds(410,300,100,20);

add(bt4);

JButton bt5 = new JButton("刷新");

bt5.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

new cadish();

}

});

bt5.setBounds(530,250,100,20);

add(bt5);

setLocationRelativeTo(null);

setVisible(true);

}

private void deleteRow(int rowIndex) {

try {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理";

String username = "root";

String password = "2272343628";

Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password);

// Delete the row with the given ID from the 'mytable' table.

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object id = tableModel.getValueAt(rowIndex, idColumnIndex); // Get the ID of the row to delete.

String sql = "DELETE FROM dish\_information WHERE id = ?";

PreparedStatement statement = conn.prepareStatement(sql);

statement.setObject(1, id); // Set the value of the primary key parameter in the query.

statement.executeUpdate();

// Remove the row from the table model.

tableModel.removeRow(rowIndex);

conn.close(); // Close the connection.

JOptionPane.*showMessageDialog*(table, "delete successful!");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void updateRow(final int rowIndex) {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

// Create dialog to edit row data.

Object[] oldRowData = new Object[tableModel.getColumnCount()];

for (int i = 0; i < oldRowData.length; i++) {

oldRowData[i] = tableModel.getValueAt(rowIndex, i);

}

Object[] newRowData = showEditRowDialog(oldRowData);

if (newRowData != null) { // Check if user clicked the "OK" button.

// generate and execute UPDATE statement

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object id = newRowData[idColumnIndex];

StringBuilder sqlBuilder = new StringBuilder("UPDATE dish\_information SET ");

sqlBuilder.append("categorize = ?, ");

sqlBuilder.append("dish\_name = ?, ");

sqlBuilder.append("unit = ?, ");

sqlBuilder.append("price = ?, ");

sqlBuilder.append("status = ?, ");

sqlBuilder.append("discount = ? ");

sqlBuilder.append("WHERE id = ?");

String sql = sqlBuilder.toString();

PreparedStatement statement = conn.prepareStatement(sql);

for (int i = 1; i < tableModel.getColumnCount(); i++) {

statement.setObject(i, newRowData[i]);

}

statement.setObject(tableModel.getColumnCount(), id);

statement.executeUpdate();

// Update the table model with the new row data.

for (int i = 0; i < newRowData.length; i++) {

tableModel.setValueAt(newRowData[i], rowIndex, i);

}

JOptionPane.*showMessageDialog*(table, "update successful!");

conn.close();

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private Object[] showEditRowDialog(Object[] oldRowData) {

// Create a dialog with input fields filled with the old data.

JComponent[] components = new JComponent[oldRowData.length];

for (int i = 0; i < oldRowData.length; i++) {

components[i] = new JTextField(oldRowData[i].toString());

}

int result = JOptionPane.*showConfirmDialog*(null, components,

"修改", JOptionPane.***OK\_CANCEL\_OPTION***);

if (result == JOptionPane.***OK\_OPTION***) {

// Retrieve the modified values from the input fields.

Object[] newRowData = new Object[oldRowData.length];

for (int i = 0; i < newRowData.length; i++) {

String text = ((JTextField) components[i]).getText();

newRowData[i] = text;

}

return newRowData;

} else {

return null;

}

}

}

class cad{

public static void main(String[] args) {

SwingUtilities.*invokeLater*(new Runnable() {

*@Override*

public void run() {

new cadish();

}

});

}

}package pack;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.Window;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.\*;

public class dish extends JFrame{

String text1;

int text\_1;

String text2 ;

String text3;

String text4;

String text5 ;

String text6;

String text7;

int text\_7 ;

dish(){

setTitle("菜品管理");

setSize(400,410);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

JLabel c1=new JLabel("id:");

c1.setBounds(120,-30,105,150);

add(c1);

JTextField tx1=new JTextField(1);

tx1.setBounds(180,35,100,20);

add(tx1);

JLabel c3=new JLabel("菜品名:");

c3.setBounds(120,30,105,150);

add(c3);

JLabel c2=new JLabel("菜品类别:");

c2.setBounds(120,0,105,150);

add(c2);

JComboBox comboBox = new JComboBox();//添加下拉框

comboBox.addItem(" ");//给下拉框添加内容

comboBox.addItem("西餐");

comboBox.addItem("湘菜");

comboBox.addItem("粤菜");

comboBox.addItem("鲁菜");

comboBox.setBounds(180,65,100,20);

add(comboBox);

JTextField tx3=new JTextField();

tx3.setBounds(180,95,100,20);

add(tx3);

JLabel c4=new JLabel("单位:");

c4.setBounds(120,60,105,150);

add(c4);

JTextField tx4=new JTextField();

tx4.setBounds(180,125,100,20);

add(tx4);

JLabel c5=new JLabel("单价:");

c5.setBounds(120,90,105,150);

add(c5);

JTextField tx5=new JTextField();

tx5.setBounds(180,155,100,20);

add(tx5);

JLabel c6=new JLabel("状态:");

c6.setBounds(120,120,105,150);

add(c6);

JTextField tx6=new JTextField();

tx6.setBounds(180,185,100,20);

add(tx6);

JLabel c7=new JLabel("折扣:");

c7.setBounds(120,150,105,150);

add(c7);

JTextField tx7=new JTextField();

tx7.setBounds(180,215,100,20);

add(tx7);

JButton b1=new JButton("确定");

b1.setBounds(120,280,80,20);

b1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

try {

text1 = tx1.getText();

text\_1 =Integer.*parseInt*(text1);

text2 = (String) comboBox.getSelectedItem();

text3 = tx3.getText();

text4 = tx4.getText();

text5 = tx5.getText();

text6 = tx6.getText();

text7 = tx7.getText();

text\_7 =Integer.*parseInt*(text7);

//

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

Connection con=DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8"

,"root","2272343628");

Statement stmt = con.createStatement();

String sql = "INSERT INTO dish\_information VALUES (?,?,?,?,?,?,?)";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setInt(1, text\_1);

pstmt.setString(2, text2);

pstmt.setString(3, text3);

pstmt.setString(4, text4);

pstmt.setString(5, text5);

pstmt.setString(6, text6);

pstmt.setInt(7, text\_7);

pstmt.executeUpdate();

// 输出成功消息

JOptionPane.*showMessageDialog*(b1, "Insertion successful!");

if(e.getSource() == b1) {

Window[] windows = Window.*getWindows*();//获取所有活动窗口，并检查每个窗口是否为MyFrame1。

for(Window window : windows) {

if(window instanceof cadish) {

window.dispose();

}

}

}

cadish e1 =new cadish();

e1.repaint();

dispose();

new dish();

con.close();

} catch (NumberFormatException ex) {

// 处理用户输入无效数据的情况

JOptionPane.*showMessageDialog*(b1, "Invalid age value!", "Error", JOptionPane.***ERROR\_MESSAGE***);

} catch (SQLException ex) {

// 处理数据库插入错误

JOptionPane.*showMessageDialog*(b1, "SQL Exception: " + ex.getMessage(), "Error", JOptionPane.***ERROR\_MESSAGE***);

}

catch (ClassNotFoundException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

});

add(b1);

JButton b2=new JButton("取消");

b2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

}

});

b2.setBounds(200,280,80,20);

add(b2);

setLocationRelativeTo(null);

setVisible(true);

}

}

class dish2{

public static void main(String[] args) {

new dish();

}

}点菜管理界面：

package pack;

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.sql.\*;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class order extends JFrame{

String text1 ;

String text2 ;

String text3 ;

String text4 ;

String text5 ;

public String getText1() {

return text1;

}

public void setText1(String text1) {

this.text1 = text1;

}

public String getText2() {

return text2;

}

public void setText2(String text2) {

this.text2 = text2;

}

public String getText3() {

return text3;

}

public void setText3(String text3) {

this.text3 = text3;

}

public String getText4() {

return text4;

}

public void setText4(String text4) {

this.text4 = text4;

}

public String getText5() {

return text5;

}

public void setText5(String text5) {

this.text5 = text5;

}

private JTable table;

private DefaultTableModel tableModel;

order(){

setTitle("点菜管理");

setSize(700,400);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

JLabel c1 = new JLabel("客户");

c1.setBounds(30,10,50,20);

c1.setFont(new Font("楷体",Font.***BOLD***,20));

add(c1);

JTextField txt1=new JTextField();

txt1.setBounds(80,10,100,20);

add(txt1);

JLabel c2 = new JLabel("餐台");

c2.setBounds(190,10,50,20);

c2.setFont(new Font("楷体",Font.***BOLD***,20));

add(c2);

JComboBox<String> comboBox2 =new JComboBox();

// 创建数据库连接

try{

String url = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

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

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select table\_id from table\_information ");

while (rs.next()) {

comboBox2.addItem(rs.getString("table\_id"));

}

comboBox2.setBounds(240,10,100,20);

// String sql1 ="select table\_id from ordering\_information where name like ?" ;

// PreparedStatement stmt1 = conn.prepareStatement(sql1);

// stmt1.setString(1,text2);

// ResultSet rs1 = stmt1.executeQuery();

// comboBox2.addItem(rs1.getString("table\_id"));

add(comboBox2);

conn.close();

}catch (SQLException ex) {

ex.printStackTrace();

}

JLabel c3 = new JLabel("就餐人数");

c3.setBounds(350,10,100,20);

c3.setFont(new Font("楷体",Font.***BOLD***,20));

add(c3);

JTextField txt3=new JTextField();

txt3.setBounds(450,10,100,20);

add(txt3);

JButton bt0=new JButton("查找");

bt0.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

text1 = txt1.getText();

text2 = (String) comboBox2.getSelectedItem();

text3 = txt3.getText();

// **TODO** Auto-generated method stub

if (e.getSource() == bt0) {

try {

// 连接到数据库并准备查询语句

Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8", "root", "2272343628");

String query = "SELECT food\_id,food\_name,number FROM ordering\_information where name like ? and table\_id like ? and meal\_number like ? ";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1,text1);

stmt.setString(2,text2);

stmt.setString(3,text3);

// 执行查询，并将结果显示在控制台上

ResultSet rs = stmt.executeQuery();

tableModel.getDataVector().removeAllElements();

tableModel.fireTableDataChanged();

table.removeAll();

while (rs.next()) {

//先清空表格

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = rs.getString(1);

row[1] = rs.getString(2);

row[2] = rs.getString(3);

tableModel.addRow(row);

}

// 关闭资源

rs.close();

stmt.close();

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

}}

});

bt0.setBounds(570,10,100,20);

add(bt0);

// Initialize the table model.

tableModel = new DefaultTableModel();

// Add columns to the table model.

tableModel.addColumn("菜品编号");

tableModel.addColumn("菜品名称");

tableModel.addColumn("数量");

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

String sql = "SELECT food\_id,food\_name,number FROM ordering\_information where name = ? ";

PreparedStatement statement = conn.prepareStatement(sql);

statement.setString(1, text1);

ResultSet resultSet = statement.executeQuery();

while (resultSet.next()) {

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = resultSet.getString(1);

row[1] = resultSet.getString(2);

row[2] = resultSet.getString(3);

tableModel.addRow(row);

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

table = new JTable(tableModel);

JScrollPane scrollPane = new JScrollPane(table);//添加滚动边框

scrollPane.setBounds(50,40,600,200);

scrollPane.repaint();

add(scrollPane);

//获得鼠标左键选中行

table.addMouseListener(new MouseAdapter() {

*@Override*

public void mousePressed(MouseEvent e) {

if (e.getButton() == MouseEvent.***BUTTON1***) { // Check if left mouse button was clicked.

int col = table.columnAtPoint(e.getPoint());

int row = table.rowAtPoint(e.getPoint());

if (row >= 0 && col >= 0) { // Check if row and column indices are valid.

}

}

}

});

JButton bt1=new JButton("添加");

bt1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

text1 = txt1.getText();

text3 = txt3.getText();

ordadd or1 = new ordadd(text1);

or1.tx2.setText(text1);

or1.tx6.setText(text3);

}});

bt1.setBounds(50,300,100,20);

add(bt1);

JButton bt2=new JButton("删除");

bt2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int row = table.getSelectedRow();

if (row >= 0) {

deleteRow(row);

}

}

});

bt2.setBounds(170,300,100,20);

add(bt2);

JButton bt3=new JButton("修改");

bt3.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

int selectedRow = table.getSelectedRow();

if (selectedRow >= 0) {

updateRow(selectedRow);

}

}

});

bt3.setBounds(290,300,100,20);

add(bt3);

JButton bt5 = new JButton("刷新");

bt5.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

new order();

}

});

bt5.setBounds(530,300,100,20);

add(bt5);

JButton bt6 = new JButton("结账");

bt6.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

text1 = txt1.getText();

jiezhang jz1 = new jiezhang(text1);

}

});

bt6.setBounds(410,300,100,20);

add(bt6);

setLocationRelativeTo(null);

setVisible(true);

}

private void deleteRow(int rowIndex) {

try {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password);

// Delete the row with the given ID from the 'mytable' table.

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object food\_id = tableModel.getValueAt(rowIndex, idColumnIndex); // Get the ID of the row to delete.

String sql = "DELETE FROM ordering\_information WHERE food\_id = ? ";

PreparedStatement statement = conn.prepareStatement(sql);

statement.setObject(1, food\_id); // Set the value of the primary key parameter in the query.

statement.executeUpdate();

String sql1 = "DELETE FROM price\_information WHERE food\_id like ? ";

PreparedStatement statement1 = conn.prepareStatement(sql1);

statement1.setObject(1, food\_id); // Set the value of the primary key parameter in the query.

statement1.executeUpdate();

// Remove the row from the table model.

tableModel.removeRow(rowIndex);

conn.close(); // Close the connection.

JOptionPane.*showMessageDialog*(table, "delete successful!");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void updateRow(final int rowIndex) {

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

// Create dialog to edit row data.

Object[] oldRowData = new Object[tableModel.getColumnCount()];

for (int i = 0; i < oldRowData.length; i++) {

oldRowData[i] = tableModel.getValueAt(rowIndex, i);

}

Object[] newRowData = showEditRowDialog(oldRowData);

if (newRowData != null) { // Check if user clicked the "OK" button.

// generate and execute UPDATE statement

int idColumnIndex = 0; // Set this to the index of the primary key column.

Object food\_id = newRowData[idColumnIndex];

StringBuilder sqlBuilder = new StringBuilder("UPDATE ordering\_information SET ");

sqlBuilder.append("food\_name = ?, ");

sqlBuilder.append("number = ?");

sqlBuilder.append("where food\_id = ?");

String sql = sqlBuilder.toString();

PreparedStatement statement = conn.prepareStatement(sql);

for (int i = 1; i < tableModel.getColumnCount(); i++) {

statement.setObject(i, newRowData[i]);

}

statement.setObject(tableModel.getColumnCount(), food\_id);

statement.executeUpdate();

// Update the table model with the new row data.

for (int i = 0; i < newRowData.length; i++) {

tableModel.setValueAt(newRowData[i], rowIndex, i);

}

String sql1 ="UPDATE price\_information ,ordering\_information SET price\_information.food\_name = ordering\_information.food\_name,price\_information.number =ordering\_information.number where ordering\_information.food\_id = price\_information.food\_id";

PreparedStatement statement1 = conn.prepareStatement(sql1);

statement1.executeUpdate();

JOptionPane.*showMessageDialog*(table, "update successful!");

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private Object[] showEditRowDialog(Object[] oldRowData) {

// Create a dialog with input fields filled with the old data.

JComponent[] components = new JComponent[oldRowData.length];

for (int i = 0; i < oldRowData.length; i++) {

components[i] = new JTextField(oldRowData[i].toString());

}

int result = JOptionPane.*showConfirmDialog*(null, components,

"修改", JOptionPane.***OK\_CANCEL\_OPTION***);

if (result == JOptionPane.***OK\_OPTION***) {

// Retrieve the modified values from the input fields.

Object[] newRowData = new Object[oldRowData.length];

for (int i = 0; i < newRowData.length; i++) {

String text = ((JTextField) components[i]).getText();

newRowData[i] = text;

}

return newRowData;

} else {

return null;

}

}

}

class ord{

public static void main(String[] args) {

SwingUtilities.*invokeLater*(new Runnable() {

*@Override*

public void run() {

new order();

}

});

}

}package pack;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.Window;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.\*;

import com.mysql.jdbc.DatabaseMetaData;

public class ordadd extends JFrame{

//获取当前时间

java.util.Date now = new java.util.Date();

//将当前时间转换为java.sql.Time类型

java.sql.Time timeNow = new java.sql.Time(now.getTime());

String text1 ;

String text2 ;

String text3 ;

String text4 ;

String text5 ;

String text6 ;

String text7 ;

JTextField tx2;

JTextField tx6;

ordadd(String s){

setTitle("点菜管理");

setSize(400,400);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

// JLabel c1=new JLabel("id:");

// c1.setBounds(120,20,105,50);

// add(c1);

// JTextField tx1=new JTextField();

// tx1.setBounds(180,35,100,20);

// add(tx1);

JLabel c2=new JLabel("客户:");

c2.setBounds(120,50,105,50);

add(c2);

tx2=new JTextField(text2);

tx2.setBounds(180,65,100,20);

add(tx2);

JLabel c3=new JLabel("餐台编号:");

c3.setBounds(120,80,105,50);

add(c3);

JComboBox<String> comboBox3 =new JComboBox();

// 创建数据库连接

try{

String url = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

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

String sql1 ="select table\_id from ordering\_information where name like ? group by table\_id" ;

PreparedStatement stmt1 = conn.prepareStatement(sql1);

stmt1.setString(1,s);

ResultSet rs1 = stmt1.executeQuery();

while (rs1.next()) {

comboBox3.addItem(rs1.getString("table\_id"));

}

String sql ="select table\_id from table\_information where status like '空' " ;

PreparedStatement stmt = conn.prepareStatement(sql);

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

comboBox3.addItem(rs.getString("table\_id"));

}

comboBox3.setBounds(180,95,100,20);

add(comboBox3);

conn.close();

}catch (SQLException ex) {

ex.printStackTrace();

}

JLabel c35 =new JLabel("菜品编号");

c35.setBounds(120,110,105,50);

add(c35);

JTextField tx35=new JTextField();

tx35.setBounds(180,125,100,20);

add(tx35);

JLabel c4=new JLabel("菜品名:");

c4.setBounds(120,140,105,50);

add(c4);

JComboBox<String> comboBox4 =new JComboBox();

// 创建数据库连接

try{

String url = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

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

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select dish\_name from dish\_information where status = 1");

while (rs.next()) {

comboBox4.addItem(rs.getString("dish\_name"));}

comboBox4.setBounds(180,155,100,20);

add(comboBox4);

conn.close();

}catch (SQLException ex) {

ex.printStackTrace();

}

JLabel c5=new JLabel("数量:");

c5.setBounds(120,170,105,50);

add(c5);

JTextField tx5=new JTextField();

tx5.setBounds(180,185,100,20);

add(tx5);

JLabel c6=new JLabel("就餐人数:");

c6.setBounds(120,200,105,50);

add(c6);

tx6=new JTextField(text6);

tx6.setBounds(180,215,100,20);

add(tx6);

JButton b1=new JButton("确定");

b1.setBounds(120,260,80,20);

b1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

try {

// text1 = tx1.getText();

text2 = tx2.getText();

text3 = (String) comboBox3.getSelectedItem();

text4 = tx35.getText();

text5 = (String) comboBox4.getSelectedItem();

text6 = tx5.getText();

text7 = tx6.getText();

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

Connection con=DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8"

,"root","2272343628");

//Statement stmt = con.createStatement();

String sql = "Insert into ordering\_information values(null,?,?,?,?,?,?)";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, text2);//姓名

pstmt.setString(2, text3);//桌号

pstmt.setString(3, text4);//菜品编号

pstmt.setString(4, text5);//菜品名

pstmt.setString(5, text6);//数量

pstmt.setString(6, text7);//就餐人数

pstmt.executeUpdate();

String sql1 = "Insert into price\_information values(null,?,?,?,?,(select price from dish\_information where dish\_name like ?),'未支付')";

PreparedStatement pstmt1 = con.prepareStatement(sql1);

pstmt1.setString(1, text4);

pstmt1.setString(2, text2);

pstmt1.setString(3, text5);

pstmt1.setString(4, text6);

pstmt1.setString(5, text5);

pstmt1.executeUpdate();

String sql2 = "UPDATE table\_information set status = '忙' , open\_time = ? where status like '空' and table\_information.table\_id =(select ordering\_information.table\_id from ordering\_information where name like ? group by ordering\_information.table\_id)";

PreparedStatement pstmt2 = con.prepareStatement(sql2);

pstmt2.setTime(1, timeNow);

pstmt2.setString(2, text2);

pstmt2.executeUpdate();

// 输出成功消息

JOptionPane.*showMessageDialog*(b1, "successful!");

dispose();

new ordadd(s);

con.close();

} catch (NumberFormatException ex) {

// 处理用户输入无效数据的情况

JOptionPane.*showMessageDialog*(b1, "Invalid age value!", "Error", JOptionPane.***ERROR\_MESSAGE***);

} catch (SQLException ex) {

// 处理数据库插入错误

JOptionPane.*showMessageDialog*(b1, "SQL Exception: " + ex.getMessage(), "Error", JOptionPane.***ERROR\_MESSAGE***);

}

catch (ClassNotFoundException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

});

add(b1);

JButton b2=new JButton("取消");

b2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

}

});

b2.setBounds(200,260,80,20);

add(b2);

setLocationRelativeTo(null);

setVisible(true);

}

}

class orda{

public static void main(String[] args) {

new ordadd(null);

}

}结账管理界面：

package pack;

import java.awt.\*;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

import java.util.\*;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class jiezhang extends JFrame{

String text1;

Double text2;

Double text3;

Double text33;

Double text4;

Double Totalprice ;

String text5;

JTextField jt3;

private JTable table;

private DefaultTableModel tableModel;

jiezhang(String s){

text1 = s;

setTitle("结账管理");

setSize(700,400);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

// Initialize the table model.

tableModel = new DefaultTableModel();

// Add columns to the table model.

tableModel.addColumn("菜品编号");

tableModel.addColumn("客户姓名");

tableModel.addColumn("菜品名");

tableModel.addColumn("数量");

tableModel.addColumn("价格");

tableModel.addColumn("状态");

// Get connection to the database.

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

String sql = "SELECT food\_id,name,food\_name,number,price,status FROM price\_information where name like ? ";

PreparedStatement statement = conn.prepareStatement(sql);

statement.setString(1, text1);

ResultSet resultSet = statement.executeQuery();

while (resultSet.next()) {

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = resultSet.getString(1);

row[1] = resultSet.getString(2);

row[2] = resultSet.getString(3);

row[3] = resultSet.getString(4);

row[4] = resultSet.getString(5);

row[5] = resultSet.getString(6);

tableModel.addRow(row);

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

table = new JTable(tableModel);

JScrollPane scrollPane = new JScrollPane(table);//添加滚动边框

scrollPane.setBounds(50,10,600,200);

scrollPane.repaint();

add(scrollPane);

// JButton b1 = new JButton("返回点菜管理");

// b1.addActionListener(new ActionListener() {

//

// @Override

// public void actionPerformed(ActionEvent e) {

// // **TODO** Auto-generated method stub

//

// dispose();

// }

//

// });

// b1.setBounds(500,300,150,20);

// add(b1);

JLabel l1 = new JLabel("应收金额");

l1.setBounds(50,250,100,20);

l1.setFont(new Font("楷体",Font.***BOLD***,20));

add(l1);

JTextField j1=new JTextField();

try{

Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password);

String sql1 = "select sum(price\*number) as total\_price from price\_information where name like ? and status like '未支付'";

PreparedStatement stmt = conn.prepareStatement(sql1);

stmt.setString(1, text1);

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

Totalprice = rs.getDouble("total\_price");

j1.setText(String.*valueOf*(Totalprice));

}

conn.close();

}catch (SQLException ex) {

ex.printStackTrace();

}

JLabel l2 = new JLabel("实收金额");

l2.setBounds(250,250,100,20);

l2.setFont(new Font("楷体",Font.***BOLD***,20));

add(l2);

JTextField j2=new JTextField();

j2.setBounds(350,250,100,20);

add(j2);

JLabel ll2 = new JLabel("优惠金额");

ll2.setBounds(450,250,100,20);

ll2.setFont(new Font("楷体",Font.***BOLD***,20));

add(ll2);

JTextField jj2=new JTextField();

jj2.setBounds(550,250,100,20);

add(jj2);

JButton l3 = new JButton("找零");

jt3=new JTextField();

jt3.setBounds(350,300,100,20);

add(jt3);

l3.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

text2 = Double.*parseDouble*(j1.getText());

text3 = Double.*parseDouble*(j2.getText());

text33 = Double.*parseDouble*(jj2.getText());

text4 = text3-text2+text33 ;

jt3.setText(String.*valueOf*(text4));

}

});

l3.setFont(new Font("楷体",Font.***BOLD***,20));

l3.setBounds(250,300,80,20);

add(l3);

JButton l5 = new JButton("查询");

JTextField j4=new JTextField();

j4.setBounds(550,300,100,20);

add(j4);

l5.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

text5 = j4.getText();

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

String sql = "SELECT food\_id,name,food\_name,number,price,status FROM price\_information where name like ? ";

PreparedStatement statement = conn.prepareStatement(sql);

statement.setString(1, text5);

ResultSet resultSet = statement.executeQuery();

tableModel.getDataVector().removeAllElements();

tableModel.fireTableDataChanged();

while (resultSet.next()) {

Object[] row = new Object[tableModel.getColumnCount()];

row[0] = resultSet.getString(1);

row[1] = resultSet.getString(2);

row[2] = resultSet.getString(3);

row[3] = resultSet.getString(4);

row[4] = resultSet.getString(5);

row[5] = resultSet.getString(6);

tableModel.addRow(row);

}

String sql11 = "select sum(price\*number) as total\_price from price\_information where name like ? and status like '未支付'";

PreparedStatement stmt11 = conn.prepareStatement(sql11);

stmt11.setString(1, text5);

ResultSet rs11 = stmt11.executeQuery();

j1.setText("");

while (rs11.next()) {

Totalprice = rs11.getDouble("total\_price");

j1.setText(String.*valueOf*(Totalprice));

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

}

});

j1.setBounds(150,250,100,20);

add(j1);

l5.setBounds(450,300,100,20);

l5.setFont(new Font("楷体",Font.***BOLD***,20));

add(l5);

JButton l4 = new JButton("结账");

l4.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

text1 = s ;

jiez jz11 = new jiez(text1,text5);

}

});

l4.setBounds(50,300,100,20);

l4.setFont(new Font("楷体",Font.***BOLD***,20));

add(l4);

setLocationRelativeTo(null);

setVisible(true);

}

}

package pack;

import java.awt.Font;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.Window;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.\*;

public class jiez extends JFrame{

String text1;

String text2;

JLabel j1;

JButton b1;

JButton b2;

jiez(String s,String t){

text1=s;

text2=t;

setTitle("餐饮管理系统");

setSize(400,300);

setLayout(null);

Image icon = Toolkit.*getDefaultToolkit*().getImage("C:\\peixunworkspace\\picture\\44.jpeg"); // 创建图标

setIconImage(icon);// 设置窗口图标（Logo）

j1 =new JLabel("是否确认结账");

j1.setFont(new Font("楷体",Font.***BOLD***,30));

j1.setBounds(100,80,200,30);

add(j1);

b1 = new JButton("确定");

b1.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

String jdbcUrl = "jdbc:mysql://localhost:3306/餐饮管理?useSSL=false&useUnicode=yes&characterEncoding=UTF-8";

String username = "root";

String password = "2272343628";

try (Connection conn = DriverManager.*getConnection*(jdbcUrl, username, password)) {

String sql = "UPDATE price\_information set status = '已支付' where status like '未支付'";

PreparedStatement stmt = conn.prepareStatement(sql);

stmt.execute();

String sql1 = "UPDATE table\_information set status = '空' ,open\_time =00 where status like '忙' and table\_information.table\_id =(select ordering\_information.table\_id from ordering\_information where name like ? group by ordering\_information.table\_id)";

PreparedStatement stmt1 = conn.prepareStatement(sql1);

stmt1.setString(1,text1);

stmt1.execute();

String sql11 = "UPDATE table\_information set status = '空' ,open\_time =00 where status like '忙' and table\_information.table\_id =(select ordering\_information.table\_id from ordering\_information where name like ? group by ordering\_information.table\_id)";

PreparedStatement stmt11 = conn.prepareStatement(sql11);

stmt11.setString(1,text2);

stmt11.execute();

// //删除所有点菜记录

// String sql2 = "delete from ordering\_information where name like ?)";

// PreparedStatement stmt2 = conn.prepareStatement(sql2);

// stmt2.setString(1,text1);

// stmt2.execute();

// //清空所有结账记录

// String sql3 = "delete from price\_information)";

// PreparedStatement stmt3 = conn.prepareStatement(sql3);

// stmt3.setString(1,text1);

// stmt3.execute();

JOptionPane.*showMessageDialog*(b1,"结账成功!");

if(e.getSource() == b1) {

Window[] windows = Window.*getWindows*();

for(Window window : windows) {

if(window instanceof jiezhang) {

window.dispose();

window.repaint();

}

}

}

dispose();

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

}

});

b1.setBounds(50,200,100,20);

add(b1);

b2 = new JButton("取消");

b2.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

dispose();

}

});

b2.setBounds(250,200,100,20);

add(b2);

setLocationRelativeTo(null);

setVisible(true);

}

}

class jie{

public static void main(String[] args) {

new jiez(null,null);

}

}