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** | **李宗鑫** |
| **学号** | **B20210304130** |
| **指导教师** | **杨刚** |
| **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 | 李宗鑫 | 学号 | B20210304130 | 班级 | 21软件01班 |
| 专业 | 软件工程 | | 指导教师姓名 | 杨刚 | |
| 项目名称 | 基于C/S架构的餐饮管理系统 | | | | |
| **实训过程**：（说明实训各阶段的工作过程，包括需求分析、系统设计、编码实现、系统测试、交付实施等阶段的主要工作。）  需求分析：  （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。  系统设计：  餐饮管理系统主要有8大模块：  （1）.员工管理界面：对员工实现增删改查。  （2）.客户管理界面：对客户实现增删改查。  （3）.餐台管理界面：对餐台实现增删改查。  （4）.菜品分类管理界面：对菜品分类实现增删改查。  （5）.菜品管理界面：对菜品分类、菜品实现增删改查。  （6）.点菜管理界面：同时实现点菜，所点菜品的增删改查，将餐台号与所点的菜品对应起来。  （7）.开台管理界面：服务员对某客户、某一空闲餐台实行开台，以及对餐台信息的修改查询。  （8）.结账管理界面：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。  编码实现：  package package2;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.swing.JFrame;  import javax.swing.JLabel;  import javax.swing.JMenu;  import javax.swing.JMenuBar;  import javax.swing.JMenuItem;  import javax.swing.JSplitPane;  import javax.swing.JTree;  import javax.swing.event.TreeSelectionEvent;  import javax.swing.event.TreeSelectionListener;  import javax.swing.tree.DefaultMutableTreeNode;  public class ManagerInterface {  JFrame jf = new JFrame("餐饮管理系统");    final int WIDTH = 1000;  final int HEIGHT = 600;  //组装视图  public void init() {  //给窗口设置属性  jf.setBounds((ScreenUtils.getScreenWidth()-WIDTH)/2,(ScreenUtils.getScreenHeight()-HEIGHT)/2,WIDTH,HEIGHT);  jf.setResizable(false);//不能重新设置大小    //设置菜单栏  JMenuBar jmb = new JMenuBar();  JMenu jMenu = new JMenu("系统管理");  JMenu jMenu1 = new JMenu("帮助");  JMenuItem m1 = new JMenuItem("切换账号");  JMenuItem m2 = new JMenuItem("退出程序");  m1.addActionListener(new ActionListener() {    public void actionPerformed(ActionEvent e) {  new LoginFrame().initFrame();  jf.dispose();  }    });    m2.addActionListener(new ActionListener() {    public void actionPerformed(ActionEvent e) {  System.exit(0);  }    });  jMenu.add(m1);  jMenu.add(m2);  jmb.add(jMenu);  jmb.add(jMenu1);    jf.setJMenuBar(jmb);    //设置分割面板  JSplitPane sp = new JSplitPane();    //支持连续布局  sp.setContinuousLayout(true);  sp.setDividerLocation(150);//分割条从左往右150个像素  sp.setDividerSize(7);//分割条的宽度为7个像素    //设置左侧内容  DefaultMutableTreeNode root = new DefaultMutableTreeNode("管理目录");    DefaultMutableTreeNode employee = new DefaultMutableTreeNode("员工管理界面");  DefaultMutableTreeNode customer = new DefaultMutableTreeNode("客户管理界面");  DefaultMutableTreeNode desk = new DefaultMutableTreeNode("餐台管理界面");  DefaultMutableTreeNode category = new DefaultMutableTreeNode("菜品分类管理界面");  DefaultMutableTreeNode dish = new DefaultMutableTreeNode("菜品管理界面");  DefaultMutableTreeNode orderdish = new DefaultMutableTreeNode("点菜管理界面");  DefaultMutableTreeNode result = new DefaultMutableTreeNode("开台界面");  DefaultMutableTreeNode change = new DefaultMutableTreeNode("结账界面");    root.add(employee);    root.add(customer);  root.add(desk);  root.add(category);  root.add(dish);  root.add(orderdish);  root.add(result);  root.add(change);    JTree tree = new JTree(root);    //设置当前tree默认选中点菜管理界面  tree.setSelectionRow(5);  tree.addTreeSelectionListener(new TreeSelectionListener(){  //当条目选中变化后，这个方法会执行  public void valueChanged(TreeSelectionEvent e) {  //得到当前选中的结点对象  Object lastPathComponent = e.getNewLeadSelectionPath().getLastPathComponent();    if(employee.equals(lastPathComponent)) {  sp.setRightComponent(new EmployeeManageComponent(jf));  sp.setDividerLocation(150);  }else if(customer.equals(lastPathComponent)) {  sp.setRightComponent(new CustomerManageComponent(jf));  sp.setDividerLocation(150);  }if(desk.equals(lastPathComponent)) {  sp.setRightComponent(new DeskManageComponent(jf));  sp.setDividerLocation(150);  }if(category.equals(lastPathComponent)) {  sp.setRightComponent(new CategoryManageComponent(jf));  sp.setDividerLocation(150);  }if(dish.equals(lastPathComponent)) {  sp.setRightComponent(new DishManageComponent(jf));  sp.setDividerLocation(150);  }if(orderdish.equals(lastPathComponent)) {  sp.setRightComponent(new OrderitemManageComponent(jf));  sp.setDividerLocation(150);  }if(result.equals(lastPathComponent)) {  sp.setRightComponent(new OrderManageComponent(jf));  sp.setDividerLocation(150);  }if(change.equals(lastPathComponent)) {  sp.setRightComponent(new ChangeManageComponent(jf));  sp.setDividerLocation(150);  }    }    });    sp.setRightComponent(new DishManageComponent(jf));    //放置到左边  sp.setLeftComponent(tree);  jf.add(sp);  jf.setVisible(true);    }      public static void main(String[] args) {  new ManagerInterface().init();  }      }  系统测试：      实际交付： | | | | | |
| **实训成果**： | | | | | |
| **实训总结**：（包括心得体会、存在的问题和改进方向。）  经过这次实训，我了解到自身对JDBC编程的不熟悉，对数据掌握的不熟练以及图形界面方面的不足，在之后的学习中，我将逐渐完善自己对各方面的学习，熟练自己的操作，增强自己对java的学习。 | | | | | |

基于C/S架构的餐饮管理系统

设计说明书

作者：B20210304130

计算机科学与工程学院

2023年 06 月 15日

**摘要**

该餐饮系统基于C/S架构，使用java语言开发，。系统具备以下功能设计：

（1）.员工管理界面：对员工实现增删改查。

（2）.客户管理界面：对客户实现增删改查。

（3）.餐台管理界面：对餐台实现增删改查。

（4）.菜品分类管理界面：对菜品分类实现增删改查。

（5）.菜品管理界面：对菜品分类、菜品实现增删改查。

（6）.点菜管理界面：同时实现点菜，所点菜品的增删改查，将餐台号与所点的菜品对应起来。

（7）.开台管理界面：服务员对某客户、某一空闲餐台实行开台，以及对餐台信息的修改查询。

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

系统的数据库设计需要涵盖用户信息、菜品信息、订单信息、员工信息等相关数据，使用Java的图形界面设计开发，提供直观友好的人机交互界面，包括菜单展示、订单处理、用户登录等功能，通过Java语言设计监听用户操作，并利用JDBC处理数据库相关操作，实现数据的存取和处理。

**关键词：**java ，C/S架构，餐饮管理系统

目录

[1 引言 1](#_Toc11882504)

[1.1 编写目的 1](#_Toc11882505)

[1.2 参考资料 1](#_Toc11882506)

[2 需求规约 2](#_Toc11882507)

[2.1 功能需求 2](#_Toc11882508)

[2.2 界面需求 3](#_Toc11882509)

[2.3 数据需求 3](#_Toc11882510)

[3 系统设计 4](#_Toc11882511)

[3.1 运行环境 4](#_Toc11882512)

[3.2 系统静态结构设计 4](#_Toc11882513)

[3.3 人机接口设计 5](#_Toc11882514)

[3.4 XXX模块设计说明 7](#_Toc11882515)

[3.5 XXX模块设计说明 7](#_Toc11882516)

[4 数据库设计 8](#_Toc11882517)

[4.1 数据库环境说明 8](#_Toc11882518)

[4.2 数据库的命名规则 8](#_Toc11882519)

[4.3 逻辑结构设计 8](#_Toc11882520)

[4.4 物理结构设计 9](#_Toc11882521)

[4.5 安全性设计 10](#_Toc11882522)

[5 测试用例设计 12](#_Toc11882523)

[附录 14](#_Toc11882524)

[附录1 程序运行结果 14](#_Toc11882525)

[附录2 程序源代码 15](#_Toc11882526)

# 1 引言

## 1.1 编写目的

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。

# 2 需求规约

## 2.1 功能需求

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。

2.1.1 用餐登记用例说明

用例名称：餐饮管理

用例描述：用户根据用户要求对开台点餐结算操作

前置条件：顾客请求开台

后置条件：客户完成结算

活动步骤：

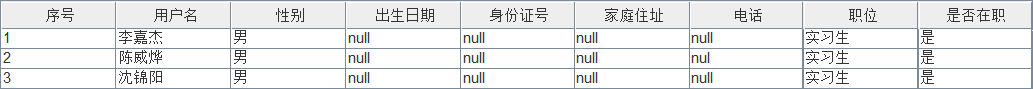
1. 客户请求开台。
2. 用户进行点单。
3. 添加订单明细记录。
4. 客户结账。

## 2.2 界面需求

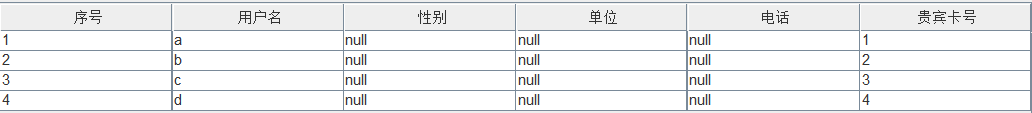
利用JAVA的GUI类设计开发友好的人机接口，设计监听用户操作和JDBC进行数据库的处理。

## 2.3 数据需求

员工实体：



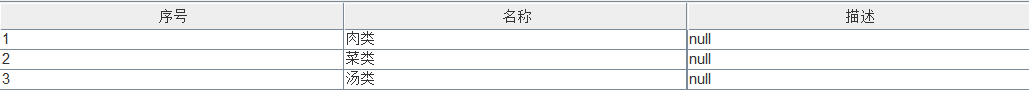
客户实体：



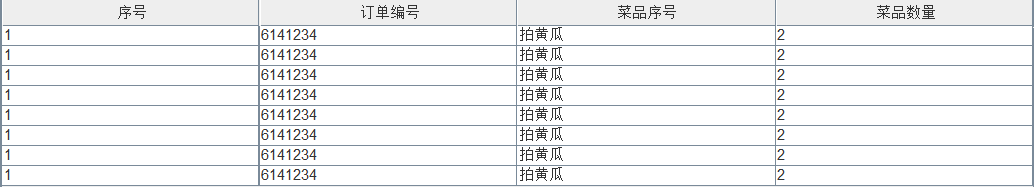
餐台实体：



菜品分类实体：



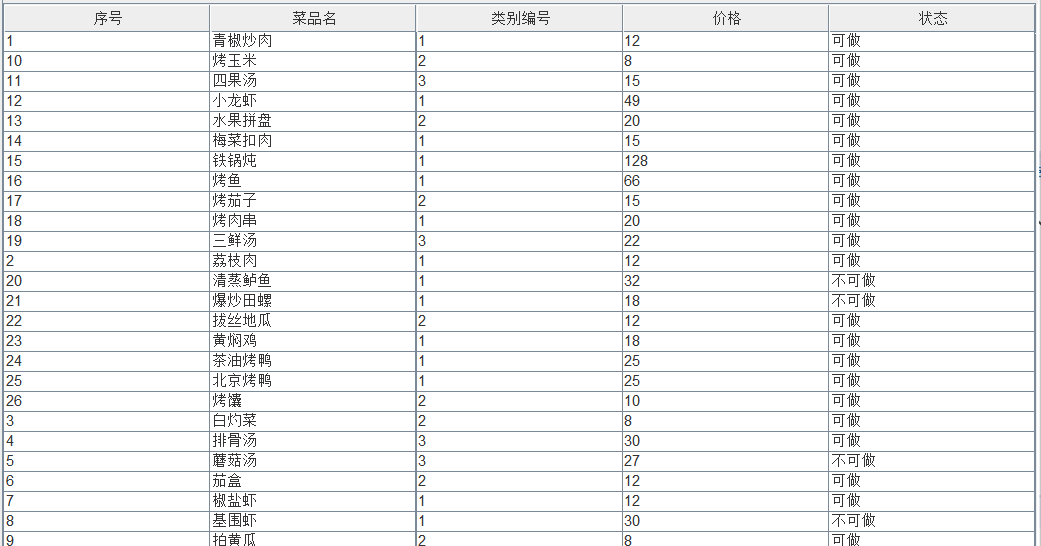
订单明细实体：



订单实体：



菜品实体：



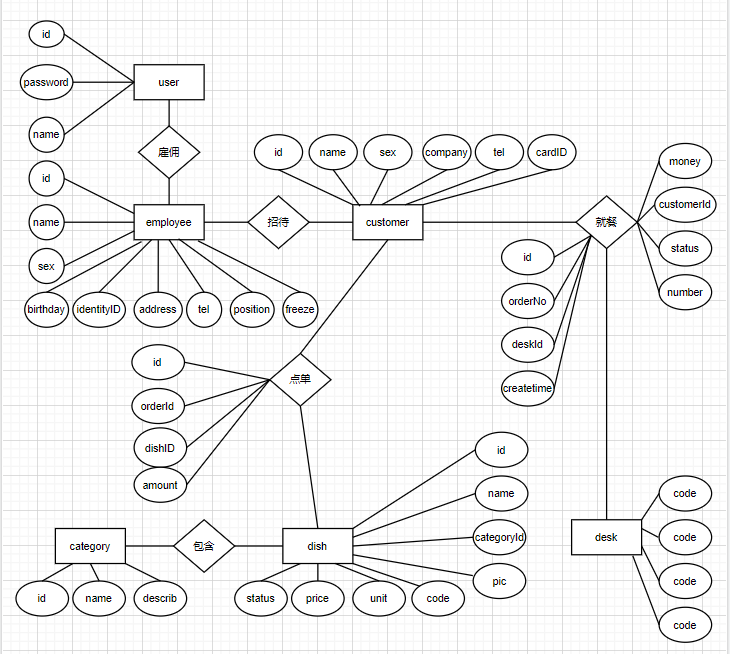
# 3 系统设计

LoginFrame实现登录界面，ManagerInterface实现主界面，主界面设置当前tree默认选中点菜界面，当条目选中变化后，显示对应管理界面，切换账号和退出程序可通过菜单栏操作。

## 3.1 运行环境

编程通过eclipse实现，数据库通过MySQL workbench 8.0 CE实现

## 3.2 系统静态结构设计



系统静态结构说明

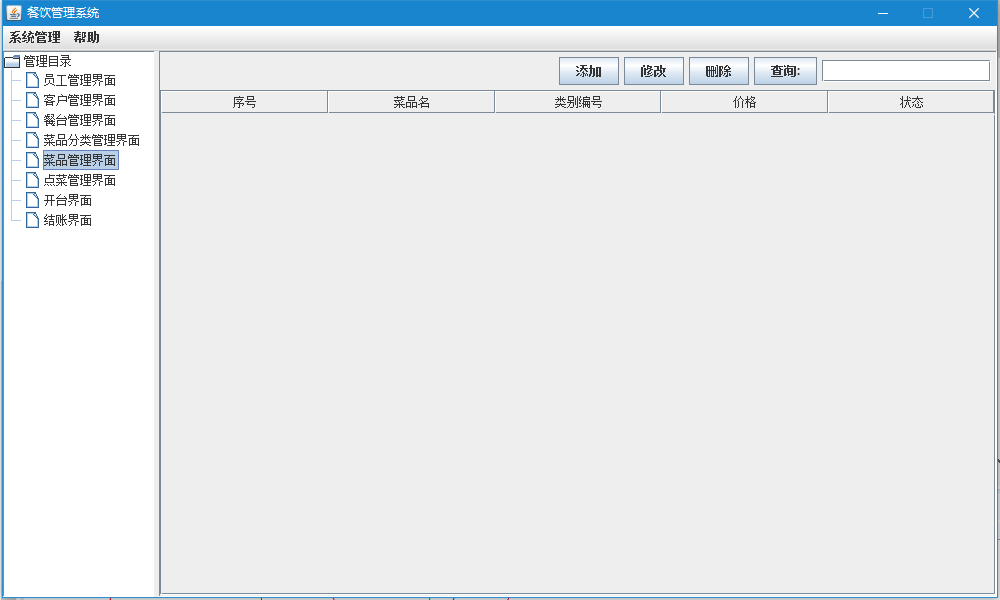
用户登陆系统以后，员工招待顾客，通过顾客的要求进行点菜，传入系统的订单明细为(id,orderId,dishId,amount)形式，点菜完成后，订单再次传入数据库，以方便完成结账操作生成的订单为（id,orderNo,deskId,createtime,money,customerId,,status,number）形式。

## 3.3 人机接口设计

3.3.1用户接口1（具体标识符）设计



3.3.2 用户接口2（具体标识符）设计



# 4 数据库设计

## 4.1 数据库环境说明

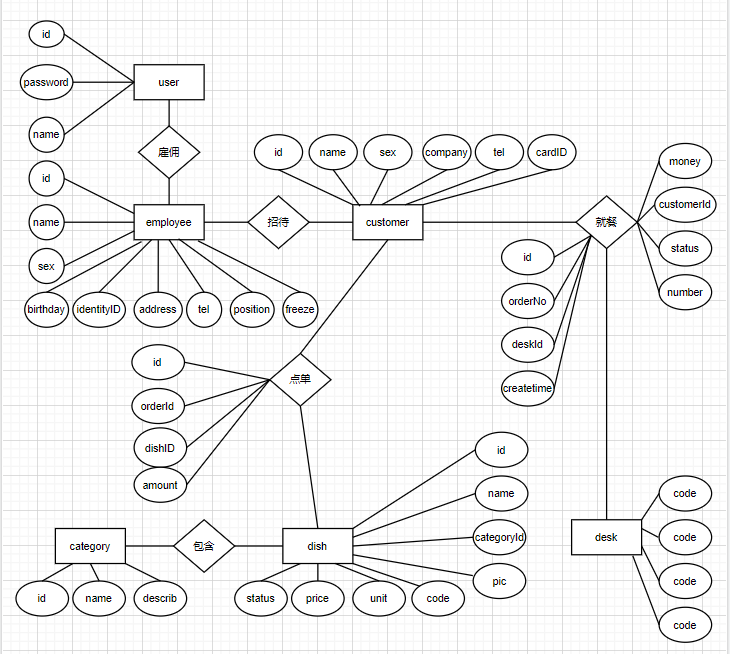
数据库工具为：MySQL Workbench 8.0 CE

## 4.2 数据库的命名规则

## 数据库命名以英文翻译为命名

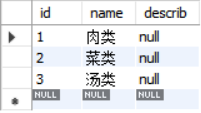
## 4.3 逻辑结构设计

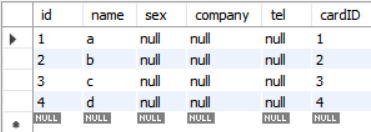
4.3.1 E-R图

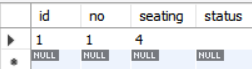


## 4.4 物理结构设计

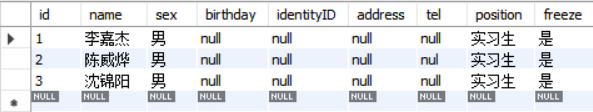
4.4.1 数据库表

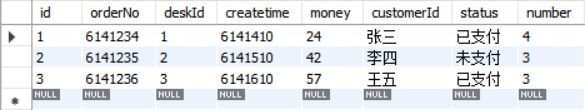


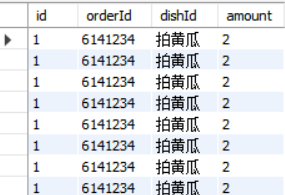


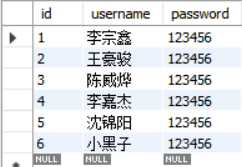












## 4.5 安全性设计

4.5.1 防止用户直接操作数据库的方法

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

//获取输入框中的字符串

String nameText = namejtf.getText();

String pawText = pawjtf.getText();

User user = **new** User(nameText,pawText);

//访问登录接口

UserDaoImpl userdao = **new** UserDaoImpl();

//比较用户名 密码 设置初始账户admin，密码123456

**if**(userdao.select(user)){

**new** ManagerInterface().init();

}

**else** {

//创建一个新窗体弹出

JFrame jf = **new** JFrame();

jf.setTitle("登录相应!!");

jf.setSize(100,100);

jf.setDefaultCloseOperation(JFrame.***DISPOSE\_ON\_CLOSE***);

//创建一个新窗体

java.awt.FlowLayout f1 = **new** java.awt.FlowLayout();

jf.setLayout(f1);

JLabel jla = **new** JLabel("用户名或密码错误!!");

jf.add(jla);

jf.setVisible(**true**);

jf.setLocationRelativeTo(**null**);//居中显示

}

}

4.5.2 用户帐号密码的加密方法

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

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

//获取用户录入的信息

String id = iField.getText();

String name = uField.getText();

String paw = pField.getText();

User user = **new** User(id,name,paw);

UserRigisterImpl userrigister = **new** UserRigisterImpl();

userrigister.select(user);

jf.dispose();

}

# });

# 附录：程序源代码

package package2;

import java.awt.Dimension;

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import javax.imageio.ImageIO;

import javax.swing.Box;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class LoginFrame {

JFrame jf = new JFrame("欢迎进入餐饮管理系统");

final int WIDTH=500;

final int HEIGHT=300;

//组装视图

public void initFrame(){//创建一个窗体的对象

//设置窗体的相关属性

jf.setBounds((ScreenUtils.getScreenWidth()-WIDTH)/2,(ScreenUtils.getScreenHeight()-HEIGHT)/2,WIDTH,HEIGHT);

jf.setResizable(false);//不能重新设置大小

//组装登录相关的元素

Box vBox = Box.createVerticalBox();

//组装用户名

Box uBox = Box.createHorizontalBox();

JLabel uLabel = new JLabel("用户名:");

JTextField uField = new JTextField(15);

uBox.add(uLabel);

uBox.add(Box.createHorizontalStrut(20));

uBox.add(uField);

//组装密码

Box pBox = Box.createHorizontalBox();

JLabel pLabel = new JLabel("密 码:");

JTextField pField = new JTextField(15);

pBox.add(pLabel);

pBox.add(Box.createHorizontalStrut(20));

pBox.add(pField);

//组装按钮

Box btnBox = Box.createHorizontalBox();

JButton loginBtn = new JButton("登录");

JButton rigistBtn = new JButton("注册");

btnBox.add(loginBtn);

btnBox.add(Box.createHorizontalStrut(100));

btnBox.add(rigistBtn);

vBox.add(Box.createVerticalStrut(40));

vBox.add(uBox);

vBox.add(Box.createVerticalStrut(20));

vBox.add(pBox);

vBox.add(Box.createVerticalStrut(40));

vBox.add(btnBox);

jf.add(vBox);

jf.setVisible(true);//可视化

//登录按钮添加监听器

ButtonAction btnaction = new ButtonAction();

loginBtn.addActionListener(btnaction);

btnaction.count=100;

btnaction.namejtf = uField;

btnaction.pawjtf = pField;

rigistBtn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//跳转到注册界面

new RigisterInterface().init();

//当前界面消失

jf.dispose();

}

});

}

//客户端程序的入口

public static void main(String[] args) {

LoginFrame loginframe = new LoginFrame();

loginframe.initFrame();

}

}

package package2;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JSplitPane;

import javax.swing.JTree;

import javax.swing.event.TreeSelectionEvent;

import javax.swing.event.TreeSelectionListener;

import javax.swing.tree.DefaultMutableTreeNode;

public class ManagerInterface {

JFrame jf = new JFrame("餐饮管理系统");

final int WIDTH = 1000;

final int HEIGHT = 600;

//组装视图

public void init() {

//给窗口设置属性

jf.setBounds((ScreenUtils.getScreenWidth()-WIDTH)/2,(ScreenUtils.getScreenHeight()-HEIGHT)/2,WIDTH,HEIGHT);

jf.setResizable(false);//不能重新设置大小

//设置菜单栏

JMenuBar jmb = new JMenuBar();

JMenu jMenu = new JMenu("系统管理");

JMenu jMenu1 = new JMenu("帮助");

JMenuItem m1 = new JMenuItem("切换账号");

JMenuItem m2 = new JMenuItem("退出程序");

m1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

new LoginFrame().initFrame();

jf.dispose();

}

});

m2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

jMenu.add(m1);

jMenu.add(m2);

jmb.add(jMenu);

jmb.add(jMenu1);

jf.setJMenuBar(jmb);

//设置分割面板

JSplitPane sp = new JSplitPane();

//支持连续布局

sp.setContinuousLayout(true);

sp.setDividerLocation(150);//分割条从左往右150个像素

sp.setDividerSize(7);//分割条的宽度为7个像素

//设置左侧内容

DefaultMutableTreeNode root = new DefaultMutableTreeNode("管理目录");

DefaultMutableTreeNode employee = new DefaultMutableTreeNode("员工管理界面");

DefaultMutableTreeNode customer = new DefaultMutableTreeNode("客户管理界面");

DefaultMutableTreeNode desk = new DefaultMutableTreeNode("餐台管理界面");

DefaultMutableTreeNode category = new DefaultMutableTreeNode("菜品分类管理界面");

DefaultMutableTreeNode dish = new DefaultMutableTreeNode("菜品管理界面");

DefaultMutableTreeNode orderdish = new DefaultMutableTreeNode("点菜管理界面");

DefaultMutableTreeNode result = new DefaultMutableTreeNode("开台界面");

DefaultMutableTreeNode change = new DefaultMutableTreeNode("结账界面");

root.add(employee);

root.add(customer);

root.add(desk);

root.add(category);

root.add(dish);

root.add(orderdish);

root.add(result);

root.add(change);

JTree tree = new JTree(root);

//设置当前tree默认选中点菜管理界面

tree.setSelectionRow(5);

tree.addTreeSelectionListener(new TreeSelectionListener(){

//当条目选中变化后，这个方法会执行

public void valueChanged(TreeSelectionEvent e) {

//得到当前选中的结点对象

Object lastPathComponent = e.getNewLeadSelectionPath().getLastPathComponent();

if(employee.equals(lastPathComponent)) {

sp.setRightComponent(new EmployeeManageComponent(jf));

sp.setDividerLocation(150);

}else if(customer.equals(lastPathComponent)) {

sp.setRightComponent(new CustomerManageComponent(jf));

sp.setDividerLocation(150);

}if(desk.equals(lastPathComponent)) {

sp.setRightComponent(new DeskManageComponent(jf));

sp.setDividerLocation(150);

}if(category.equals(lastPathComponent)) {

sp.setRightComponent(new CategoryManageComponent(jf));

sp.setDividerLocation(150);

}if(dish.equals(lastPathComponent)) {

sp.setRightComponent(new DishManageComponent(jf));

sp.setDividerLocation(150);

}if(orderdish.equals(lastPathComponent)) {

sp.setRightComponent(new OrderitemManageComponent(jf));

sp.setDividerLocation(150);

}if(result.equals(lastPathComponent)) {

sp.setRightComponent(new OrderManageComponent(jf));

sp.setDividerLocation(150);

}if(change.equals(lastPathComponent)) {

sp.setRightComponent(new ChangeManageComponent(jf));

sp.setDividerLocation(150);

}

}

});

sp.setRightComponent(new DishManageComponent(jf));

//放置到左边

sp.setLeftComponent(tree);

jf.add(sp);

jf.setVisible(true);

}

public static void main(String[] args) {

new ManagerInterface().init();

}

}