****

**课程设计报告**

**课程设计题目：**《企业考勤系统》

**学生姓名**：朱广成

**专 业：计算机科学与技术**

**班 级**：2020702

**学 号：2020210883**

**指导教师：张军**

**课程名 ：Java程序设计**

**2022年 4月 29日**

目录

1. 课程设计时间：

2022/4/18~2022/4/29

二、课程设计地点：

在线

三、课程设计目标：

由数据库和Java的连接，设计一个企业考勤的完整系统，该系统要完全具有实验设计要求的所有功能，并且满足美观布局要求，要熟练的掌握java和数据库的连接操作，重新回忆数据库的操作语句，以及图形界面的运用，熟悉了解各个组件的功能，监听器的左右，布局的设计。

1. 课程设计内容：

1）设计企业考勤数据库，通过仔细调研，分析出具体的实体和属性；

（2）创建和设计相应的表；根据系统功能需求设计相应的视图、存储过程。根据数据的完整性需求设计相应的触发器；

（3）系统分为三种用户，员工用户、部门管理员用户、管理员用户，不同用户的操作功能不同，因此登录后显示的界面应该不同；

（4）管理员用户登陆系统后，主要对员工用户和部门管理员信息进行增加、删除、修改；员工信息包括员工编号（5位数字）、员工名（中文姓名或英文姓名）、登陆密码（8位以上）、部门、性别（从下拉列表选择“男”或“女”）、联系电话（11或12位数字组成）等信息；部门管理员信息包括部门管理员员工编号（5位数字）、部门管理员姓名（中文姓名或英文姓名）、登陆密码（8位以上）、性别（从下拉列表选择“男”或“女”）、部门（只能是中文）、联系电话（11或12位数字组成）；管理员也需要对部门信息进行增加、删除、修改，部门信息包括部门编号（3位数字）、部门名称（只能是中文）、备注；

（5）员工登陆到系统后，可以进行考勤登记，考勤信息包括考勤日期时间（考勤日期时间可获得系统时间和日期，不需员工输入）、考勤类别（考勤类别分为上班、下班、请假），如果考勤类别为请假，需要输入请假理由；员工也可按日、月、年、指定时间段查看自己的考勤信息。

（6）部门管理员用户登陆系统后，可以增加、删除、修改员工的考勤信息；也可根据指定时间段查看本部门员工的考勤信息；也可按照月、年、指定时间段统计本部门员工的各类考勤数据。

（7）窗体上要求设计菜单，以实现各功能；工具栏的设计可选；可以考虑信息的导入导出，例如可以导出部门的考勤信息，导出文件格式为xls

五、课程设计过程（要求论述详实）

1. 需求分析

系统的功能需求

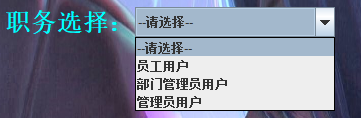
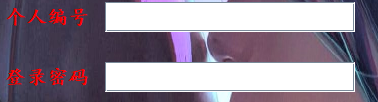
设计一个企业考勤系统，系统可以由登录账号和登录密码及个人的选择进入到不同的身份系统，要求实现的是在管理员系统可以实行按对员工用户和部门管理员信息的增删改，部门管理员可以增删改部门员工的考勤，以及查询考勤，员工主要是登录打卡考勤以及查看自己的考勤信息，最后就是要对信息有一定的要求。

1. 系统设计

系统整体逻辑结构设计:

整个系统首先是设计登陆界面,这是系统的基础,由此选择登录身份,和输入个人账号和登录密码,点击登录后若账号密码正确则可进入到对应的职务界面中,还在登录界面设计了一个忘记密码,这也是基于实际情况设计的一个额外功能.

登陆界面

管理员

部门管理员

普通员工

界面功能:

1. 增删改员工信息 1.增删改员工考勤信息 1.进行打卡操作
2. 增删改部门信息 2.根据指定时间查看考 2.根据时间查看个人考勤
3. 增加额外的一些功能 勤 3.查看个人信息
4. 返回主界面 4.返回主界面 4.返回主界面

数据库支持

普通员工数据表:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 员工编号 | 登陆密码 | 联系电话 | 员工姓名 | 性别 | 部门 |
| varchar(50) | varchar(50) | varchar(50) | varchar(50) | varchar(50) | varchar(50) |

部门管理员数据表:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 部门管理员编号 | 部门管理员姓名 | 部门管理员密码 | 部门管理员性别 | 部门 | 部门管理员电话 |
| varchar(50) | varchar(50) | varchar(50) | varchar(50) | varchar(50) | varchar(50) |

考勤信息数据表:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 考勤状态 | 考勤时间 | 请假理由 | 编号 | 部门 |
| varchar(50) | datetime | varchar(50) | varchar(50) | varchar(50) |

部门信息数据表

|  |  |  |
| --- | --- | --- |
| 部门编号 | 部门名称 | 备注 |
| varchar(50) | varchar(50) | varchar(50) |

数据库设计（按数据库设计的规范化流程来写）:

首先根据题目设计了普通员工数据表:

其中规定了员工编号为主键,编号,姓名,密码列值不能为空.

员工编号→登录密码,员工姓名,联系电话,性别,部门

联系电话→登陆密码 (找回密码的基础)

部门→外码→部门管理员

对于部门管理员数据表:

其中规定部门管理员编号为主键,部门管理员登录密码,姓名,电话列值不为空

部门管理员编号→部门管理员姓名,部门管理员密码,部门管理员电话,性别,部门

部门→外码→部门信息.名称

考勤信息表:

规定编号为主键,考勤状态不为空,考勤时间设置默认值自动获取当前系统时间

编号→考勤状态,考勤时间,部门

部门→外码→部门管理员

部门信息表:

规定部门编号为主键,部门名称列值不能为空.

部门编号→部门名称

安全性设计：

对于登陆密码设置为\*展示，无法直接看到登录密码，也就是使JPasswordField方法，具体代码：



登录时登录密码：



对于数据库中的访问，设计了存储过程，触发器的使用，这样即使对方获取了sql语句也无法直接对数据进行破坏，对各个表设置主键，规定范式。

对每个操作中文本框的内容都设置了正则表达式的约束，当不满足要求时无法进行操作。

3. 详细实现

每个模块的详细实现（具体的技术实现细节、解决的关键问题、具体的算法、对错误问题的解决方法等，可以粘贴关键代码，完整代码附在报告最后）

1. 登陆界面:

由jframe创建界面,里面添加了下拉列表框组件,由此选择需要登录的职务界面,然后是添加了登录账号和登陆密码以及对应的文本框,对文本框设置对应的监听器也是关键,由获取的文本框内容决定进入的登录界面.

错误的问题就是因为在其他类中需要调用此文本框的获取值,开始在类中方法新建了String值后面无法引用,所以要用静态变量获取文本框内容.static String[] *s1* = new String[3];

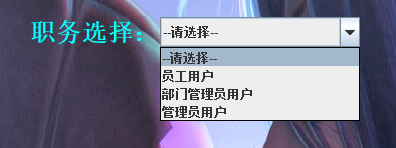
jTextField.addKeyListener(new KeyListener() {  
 @Override  
 public void keyTyped(KeyEvent e) {  
 }  
 @Override  
 public void keyPressed(KeyEvent e) {  
 }  
 @Override  
 public void keyReleased(KeyEvent e) {  
 *s1*[0]=jTextField.getText();  
 }  
});



设置职务选择下拉列表框:

JLabel jLabel4=new JLabel("职务选择：");  
JComboBox cmb=new JComboBox(); *//创建JComboBox*cmb.addItem("--请选择--"); *//向下拉列表中添加一项*cmb.addItem("员工用户");  
cmb.addItem("部门管理员用户");  
cmb.addItem("管理员用户");

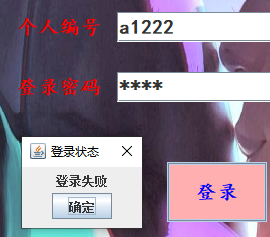
cmb.addItemListener(new ItemListener() {  
 @Override  
 public void itemStateChanged(ItemEvent e) {  
 String zhiwu=cmb.getSelectedItem().toString();  
 if(zhiwu.equals("--请选择--")){  
 field3.setText("");  
 }else  
 {  
 field3.setText(zhiwu);  
 *s2*=cmb.getSelectedItem().toString();  
 }  
 }  
});



对输入的账号和密码规定正则表达式,当满足要求时才可以进入职务的页面:

*s1*[0].matches("1\\d{4}")&&*s1*[1].length()>=8

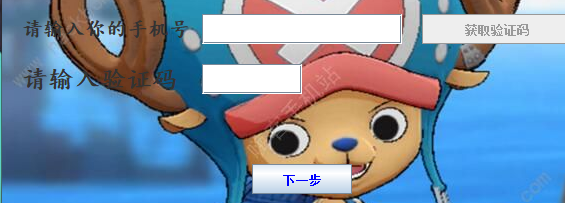
不满足时会弹出对话框提示登录失败：



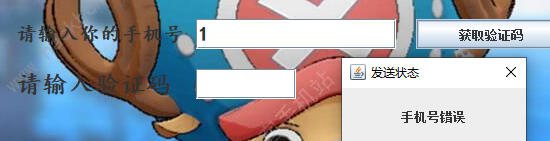
1. 找回密码界面:  
   根据实际情况设置的这个功能,员工可以根据自己的手机号,从数据库中获取个人密码.这里考虑到安全性,我增添了获取验证码的功能,这里我解决的主要问题就是相对获取验证码这一按钮绑定监听器,又想实现实时监控,当手机号码文本框内容为空,按钮无法点击,当不为空可以点击获取验证码,但手机号码不对也会报错,都满足时才会弹出对话框:验证码已发送.最后从网上查阅资料使用了getDocument().addDocumentListener(new DocumentListener()

完整代码:

jTextField1.getDocument().addDocumentListener(new DocumentListener() {  
 @Override  
 public void insertUpdate(DocumentEvent e) {  
 button3.setEnabled(true);  
 }  
 @Override  
 public void removeUpdate(DocumentEvent e) {  
 if(jTextField1.getText().length()==0)  
 {  
 button3.setEnabled(false);  
 }  
 }  
 @Override  
 public void changedUpdate(DocumentEvent e) {  
 }  
});  
jTextField1.addKeyListener(new KeyListener() {  
 @Override  
 public void keyTyped(KeyEvent e) {  
 }  
 @Override  
 public void keyPressed(KeyEvent e) {  
 }  
 @Override  
 public void keyReleased(KeyEvent e) {  
 *s1*[0]=jTextField1.getText();  
 }  
});



当输入一个数字，获取验证码按钮即可点击，但是会提示手机号错误：



3.管理员界面:

数据库界面主要是要连接数据库实现查询，增删改的功能，连接数据库的语句可以通用。

try {  
 Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");  
 System.*out*.println("数据库驱动加载成功");  
 } catch (ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
 String connectionUrl =  
 "jdbc:sqlserver://127.0.0.1:1433;"  
 + "databaseName=企业考勤系统;"  
 + "user=sa;"  
 + "password=123456;";  
 try {  
 *con* = DriverManager.*getConnection*(connectionUrl);  
 System.*out*.println("数据库连接成功");  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return *con*;  
}

整个界面的最大问题就是使用sql语句实现数据库的操作，对显示所有员工信息时，不知道如何显示，最后从网上查阅资料使用了Jtable方法，通过Resultset从数据库中不断读取数据然后用Object将获取的数据转换成合适的JTable的数据格式。展示格式如下：



以及在后面修改数据库中数据时，最开始直接使用sql语句修改总是不行，后来使用String语句然后对语句进行sql操作，就是对字符串的拼接有了很大的进步。如删除操作时：

String sql ="delete from dbo.普通员工 where 员工编号=";  
s.append(sql);  
s.append(c1.*s2*[0]);  
Statement st=*con*.createStatement();  
 st.executeUpdate(s.toString());

在管理员界面使用了很多按钮，通过点击按钮实现该按钮的功能，所以要对所有按钮绑定监听器。

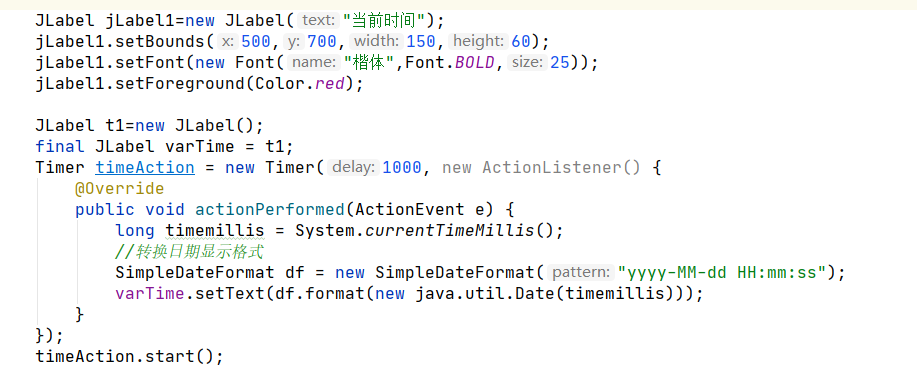
ActionListener actionListener = new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getActionCommand().equals("续茶")) {  
 JOptionPane.*showMessageDialog*(btn1, "您叫的茶正在路上");  
 } else if (e.getActionCommand().equals("呼叫秘书")) {  
 JOptionPane.*showMessageDialog*(btn2, "秘书正在赶来");  
 } else if (e.getActionCommand().equals("下工")) {  
 JOptionPane.*showMessageDialog*(btn3, "已通知各部门下工");  
 } else if (e.getActionCommand().equals("递烟")) {  
 JOptionPane.*showMessageDialog*(btn4, "小弟正来递烟");  
 } else if (e.getActionCommand().equals("员工基本信息")) {  
 Test t1=new Test();}  
 else if(e.getActionCommand().equals("添加员工信息")){  
 new enroll();  
 }  
 else if(e.getActionCommand().equals("删除员工信息")){  
 new ck();  
 }  
 else if(e.getActionCommand().equals("修改员工信息")){  
 new xiugai();  
 }  
 }  
 };

对按钮里的title进行比较操作，当比较的字符串和要实现的功能相同，则会对该按钮进行操作，然后对所有按钮绑定一个监听器：addActionListener(actionListener)。

对底层我设置了一个自动获取当前时间，显示在界面上。时间还会实时变化。这个功能在普通员工和部门管理员都有。



实现代码如下：



为了可以直接返回登陆界面，我设置了一个返回登录界面按钮，点击时，会关闭当前管理员界面，重新打开一个登录的界面。

代码的实现如下：

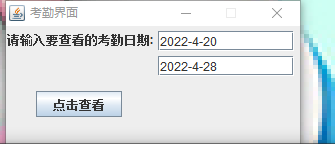


部门管理员界面：

该界面主要是实现增删改员工考勤信息，和通过指定时间段查看部门员工管理员信息。

因为是部门管理员，所以我根据实际情况是仅仅对个人部门进行操作。对于考勤信息的增删改，我是通过设计按钮点击实现，比如当点击增加考勤信息的按钮时会跳转到一个增添考勤的界面，该界面，管理员需要的在员工编号文本框，和考勤状态文本框中输入内容，通过getText()获取文本框内容，然后再通过连接数据库，将该增添信息存入到员工考勤表中。

在查看考勤时，开始时我将时间设置的是varchar类型，后来发现查看考勤时在文本框直接输入日期的格式时会报错，通过了解发现文本框传入的是字符串类型，无法直接解析日期格式，于是我将考勤时间一列的类型设置为datetime类型，然后再次输入时成功解决。按照如下格式即可查看指定的考勤信息。



对考勤信息的展示同管理员的查看员工信息一样，通过JTable展示信息，设置了滚轮，可以通过滑动显示未显示出的信息。具体代码：

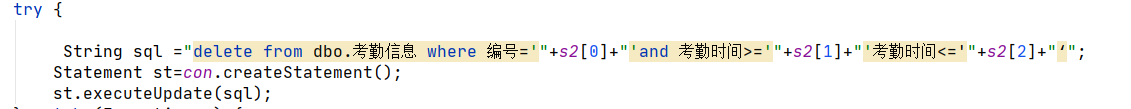
String sql="select *\** from dbo.考勤信息 where 部门='"+getkq.*getDep1*(JFrameBackground.*s1*[0])+"'"+" and 考勤时间>='"+考勤.*s2*[0]+"'"+" and 考勤时间<='"+考勤.*s2*[1]+"'";  
PreparedStatement pstm=conn.prepareStatement(sql);  
ResultSet rs =pstm.executeQuery();  
int count=0;  
while (rs.next()){  
 count++;  
}  
rs=pstm.executeQuery();  
Object[][] info=new Object[count][5];  
count=0;  
while (rs.next()){  
 info[count][0]=rs.getString("考勤状态");  
 info[count][1]=rs.getString("考勤时间");  
 info[count][2]=rs.getString("请假理由");  
 info[count][3]=rs.getString("编号");  
 info[count][4]=rs.getString("部门");  
 count++;  
}  
String[] title={"考勤状态","考勤时间","请假理由","编号","部门"};  
this.tabDemo=new JTable(info,title);  
this.jth = this.tabDemo.getTableHeader();  
this.scpDemo.getViewport().add(tabDemo);

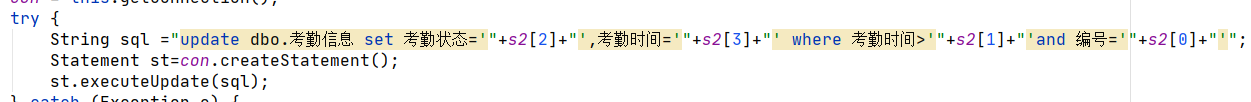
考勤信息展示如下：



删除考勤和修改考勤都是类似的，需要输入员工编号和一个时间段，对该时间段的该员工的考勤进行删除和修改考勤状态操作。

代码如下：





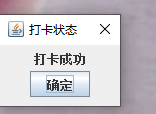
普通员工界面：

普通员工界面的功能较少，只有进行考勤信息，和查看个人考勤信息，在这里我最开始设置的打卡操作，跳转的界面需要输入员工编号，后来考虑实际情况应该是直接进行打卡就可以直接存入到数据，因此员工编号应该从登录界面的登陆账号文本框直接获取到的，于是打卡的上下班，直接点击打卡后会直接跳转打卡成功，而请假操作，会需要输入请假信息，我对考勤信息表，直接设置了考勤状态，考勤时间，请假理由，当打卡操作为上下班时不需要输入请假理由，请假理由默认为空值，当请假时，会将请假理由的文本框内容存入到表中的请假理由列中，打卡的时间是数据库表中自动获取的系统时间。

打卡上下班的sql语句：

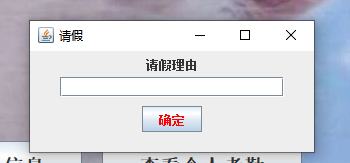
s5[0]=get.*getDep*(JFrameBackground.*s1*[0]);//通过get类中的getDep方法是获取到该员工的部门

*sql*=*con*.prepareStatement("insert into dbo.考勤信息(考勤状态,编号,部门) values ('上班',?,?)");  
 *sql*.setString(1,JFrameBackground.*s1*[0]);//从登录界面登陆账号文本框获取到的员工编号。  
 *sql*.setString(2, s5[0]);//该员工的部门  
 *sql*.executeUpdate();



请假的代码：

s5[0]=get.*getDep*(JFrameBackground.*s1*[0]);//获取的员工部门  
*sql*=*con*.prepareStatement("insert into dbo.考勤信息(考勤状态,请假理由,编号,部门) values ('请假',?,?,?)");  
*sql*.setString(1,qj[0]);//当请假时弹出的请假界面里请假理由的文本框内容  
*sql*.setString(2,JFrameBackground.*s1*[0]);  
*sql*.setString(3, s5[0]);//员工部门  
*sql*.executeUpdate();

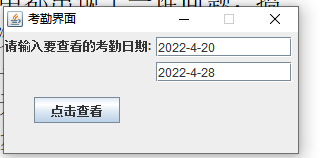


点击查看个人考勤时的按钮会跳转到查看个人考勤查看的界面，该界面我同部门管理员的查看考勤一样，设置了两个文本框，需要输入查看日期的起始时间和终止时间，然后使用sql语句查询数据库中考勤信息该时间段该员工的考勤信息。

具体的查询代码如下：

String sql="select *\** from dbo.考勤信息 where 编号='"+JFrameBackground.*s1*[0]+"'"+" and 考勤时间>='"+个人考勤.*s2*[0]+"'"+" and 考勤时间<='"+个人考勤.*s2*[1]+"'";  
  
PreparedStatement pstm=conn.prepareStatement(sql);  
ResultSet rs =pstm.executeQuery();

需要输入的格式如下：



1. 课程设计总结：

本次课程设计对我的帮助很大，这是这么久以来第一次做如此大规模的系统设置，它综合了java，数据库，需要用到Java的图形界面的各种东西，在实验开始时自己一头雾水，无从下手，甚至在连接数据库操作中都出现了一堆问题，搞了好久的时间，最后在室友的帮助下解决掉了问题，这次课设的展开是以小组开始的，我在小组中的合作中了解到了如何与他人分工，与他们探讨，与他人共同进步，在老师的指导下，我们每天井井有序的开展着每天的任务，每天在写代码时都会遇到各种各样的问题，在他人的帮助下我才一步步解决，这正是小组合作的重要性，在实验过程中，我对图形界面中各种组件的使用越来越熟悉，又重新会想起上学期数据库的语句，后期的我已经可以独自解决大部门问题，在整个课设中我还对布局的美观的重要性有了更深的认识。

参考文献：

《Java课程设计实战教程》 ---人民邮电出版社

《Java从入门到精通》 ---清华大学出版社