

工资管理系统

实验报告

学号 E41514015 专业 15 信息安全 姓名 陶蕊
实验日期 2018.6.13 教师签字 成绩



目录

1	概述.....	3
1.1	背景.....	3
1.2	应用领域与使用对象.....	3
1.3	参考资料.....	3
2	系统综述.....	4
2.1	系统功能简介.....	4
3	功能.....	4
3.1	登陆.....	4
3.2	创建账号.....	5
3.3	忘记密码与密码重置.....	7
3.4	员工模式.....	12
3.4.1	员工模式主页.....	12
3.4.2	员工个人信息查询.....	13
3.4.3	员工工资信息查询.....	13
3.5	领导模式.....	12
3.5.1	领导模式主页.....	12
3.5.2	员工工资信息查询.....	12
3.5.3	员工工资信息修改.....	12
3.5.4	删除员工.....	12
4	安全模块.....	21
4.1	登陆验证码.....	21
4.2	数字签名认证.....	19
4.3	密码加密.....	23
4.4	访问权限控制.....	24
5	附录（部分源代码）.....	25

一、概述

1.1 背景

财务是每一个单位都会涉及的问题，管理软件雨后春笋般崛起的今天，企业不仅需要通过软件有效地对其实施管理，还想操作能够尽可能简单，软件易于上手。通过查阅相关资料，对市场需求进行分析，对即成工资管理软件的研究和测试，我利用自学的部分 java web 知识，把这次期末作业当成是对自己的编程能力的挑战。除 java 课所学知识，我外加了 jsp、servlet、http、ui 美化等部分制作了一个简易的工资管理系统，所用数据库为 SQL server2014，JDK 版本为 1.8.0。

1.2 应用领域与使用对象

应用领域为所有企业和单位，使用对象为企业和单位中的领导人与员工。

1.3 参考资料

参考 easyui 网站提供的界面设计模板。

网址	简介
http://www.jeasyui.com/index.php	

二、系统综述

2.1 系统功能简介

工资管理管理系统分为领导和员工两种登陆模式，员工主要是对信息的查看，包括个人基本信息和工资信息（对数据库的“查”）。领导主要是对员工的管理，包括对员工的工资信息进行查询（对数据库的“查”），删除员工（对数据库的“删”），修改员工工资信息（对数据库的“改”）。在登陆主页有密码忘记选项，回答三个基本核对信息可以实现对密码的重置（对数据库的“改”）；另外登陆主页还有账号注册选项（对数据库的“增”），新来的员工可以通过此链接注册。

由于本专业为信息安全，故此系统加入安全方面的因素，如验证码、数字签名认证、密码哈希算法加密、访问限权等。

此学生信息管理系统可以有效地帮助企业 and 单位对员工信息实施存储和管理，方便员工和领导的使用。

三、功能

3.1 登陆

支持两种登陆模式，分别为员工模式和领导模式，不同模式所拥有的权限不同



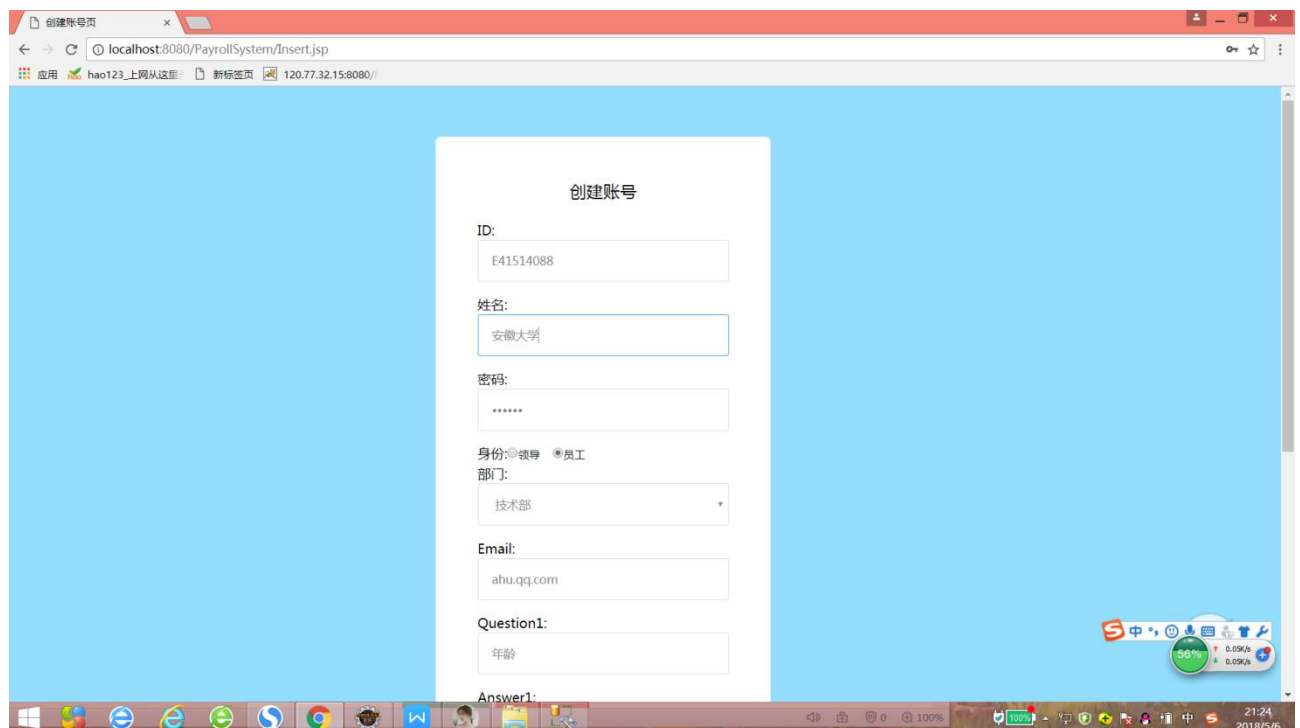
3.2 创建账号

点击登录页面下方的创建账号：

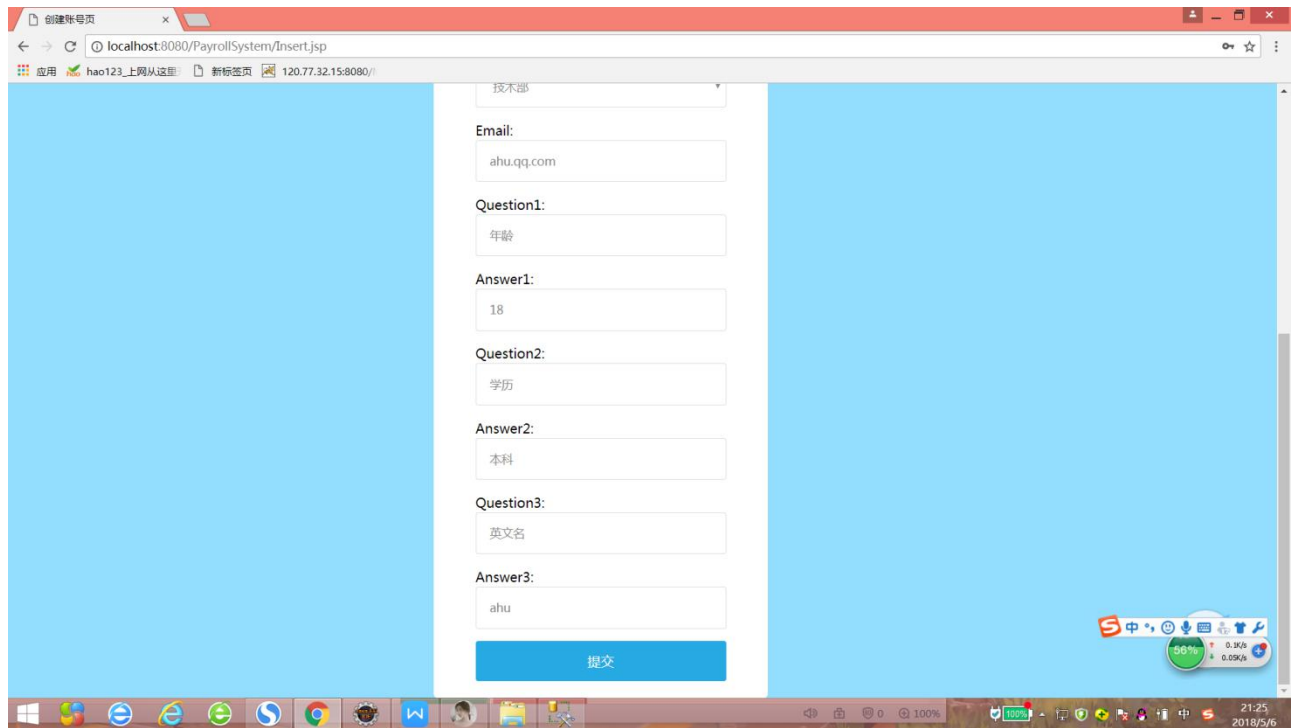


The image shows a login page for a '工资管理系统' (Payroll System). At the top is a user icon. Below it, the title '工资管理系统' is centered. There are input fields for 'ID' and '密码' (Password). Below these are radio buttons for '领导' (Leader) and '员工' (Employee). A '验证码' (Captcha) field shows '2009'. A large blue '登录' (Login) button is at the bottom. Below the login button are three links: '找回密码' (Forgot Password), '创建账号' (Create Account, highlighted with a red box), and '数字签名' (Digital Signature).

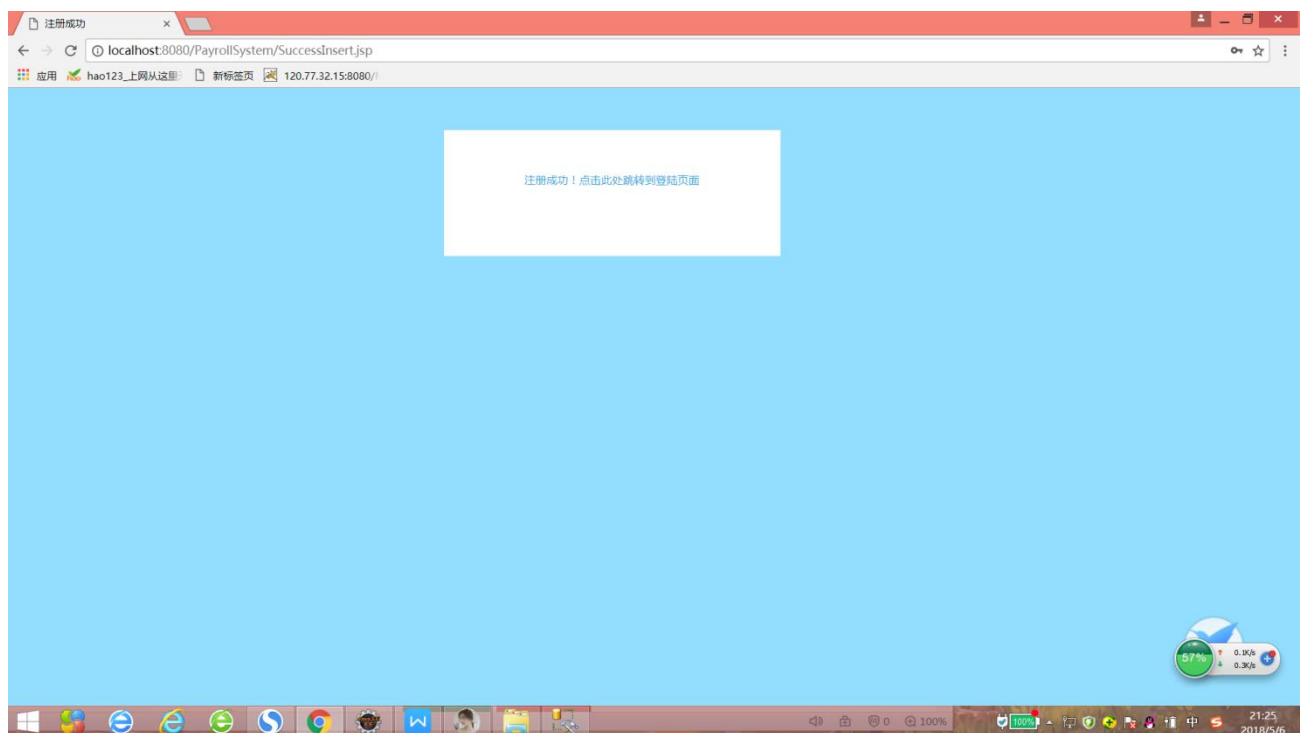
进入创建账号界面，输入相关信息：



The image is a screenshot of a web browser showing the '创建账号' (Create Account) page. The browser's address bar shows 'localhost:8080/PayrollSystem/Insert.jsp'. The page has a light blue background. The '创建账号' form is centered and contains the following fields: 'ID:' with value 'E41514088', '姓名:' (Name) with value '安徽大学', '密码:' (Password) with masked characters '*****', '身份:' (Role) with radio buttons for '领导' and '员工' (selected), '部门:' (Department) with a dropdown menu showing '技术部', 'Email:' with value 'ahu.qq.com', 'Question1:' (Security Question) with value '年龄', and 'Answer1:' (Security Answer) which is currently empty. The Windows taskbar at the bottom shows the date and time as '21:24 2018/5/6'.



点击提交，显示注册成功，点击链接即可跳转到登陆页面：



注册此账号前的数据库：

ID	STPasswo...	STRank	STName	Email	Departme...	Question1	Answer1	Question2	Answer2	Question3	Answer3
E4151401...	123456	领导	陶蕊	10215899...	行政部	你的本科大...	安徽大学	你的大学入...	15	你的本科专...	信息安全
E4151401...	123456	员工	王惠茹	NULL	技术部	NULL	NULL	NULL	NULL	NULL	NULL
E4151401...	123456	员工	胡若霄	NULL	人力资源部	NULL	NULL	NULL	NULL	NULL	NULL
E4151401...	123456	员工	曹杰	NULL	财务部	NULL	NULL	NULL	NULL	NULL	NULL
E4151402...	123456	员工	吴亦凡	NULL	运维部	NULL	NULL	NULL	NULL	NULL	NULL
E4151402...	123456	员工	吴彦祖	NULL	营销部	NULL	NULL	NULL	NULL	NULL	NULL
E4151402...	123456	员工	吴磊	NULL	测试部	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

注册此账号后的数据库：

ID	STPasswo...	STRank	STName	Email	Departme...	Question1	Answer1	Question2	Answer2	Question3	Answer3
E4151401...	123456	领导	陶蕊	10215899...	行政部	你的本科大...	安徽大学	你的大学入...	15	你的本科专...	信息安全
E4151401...	123456	员工	王惠茹	NULL	技术部	NULL	NULL	NULL	NULL	NULL	NULL
E4151401...	123456	员工	胡若霄	NULL	人力资源部	NULL	NULL	NULL	NULL	NULL	NULL
E4151401...	123456	员工	曹杰	NULL	财务部	NULL	NULL	NULL	NULL	NULL	NULL
E4151402...	123456	员工	吴亦凡	NULL	运维部	NULL	NULL	NULL	NULL	NULL	NULL
E4151402...	123456	员工	吴彦祖	NULL	营销部	NULL	NULL	NULL	NULL	NULL	NULL
E4151402...	123456	员工	吴磊	NULL	测试部	NULL	NULL	NULL	NULL	NULL	NULL
E4151408...	123456	员工	安徽大学	ahu.qq.co...	技术部	年龄	18	学历	本科	英文名	ahu
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.3 忘记密码与密码重置

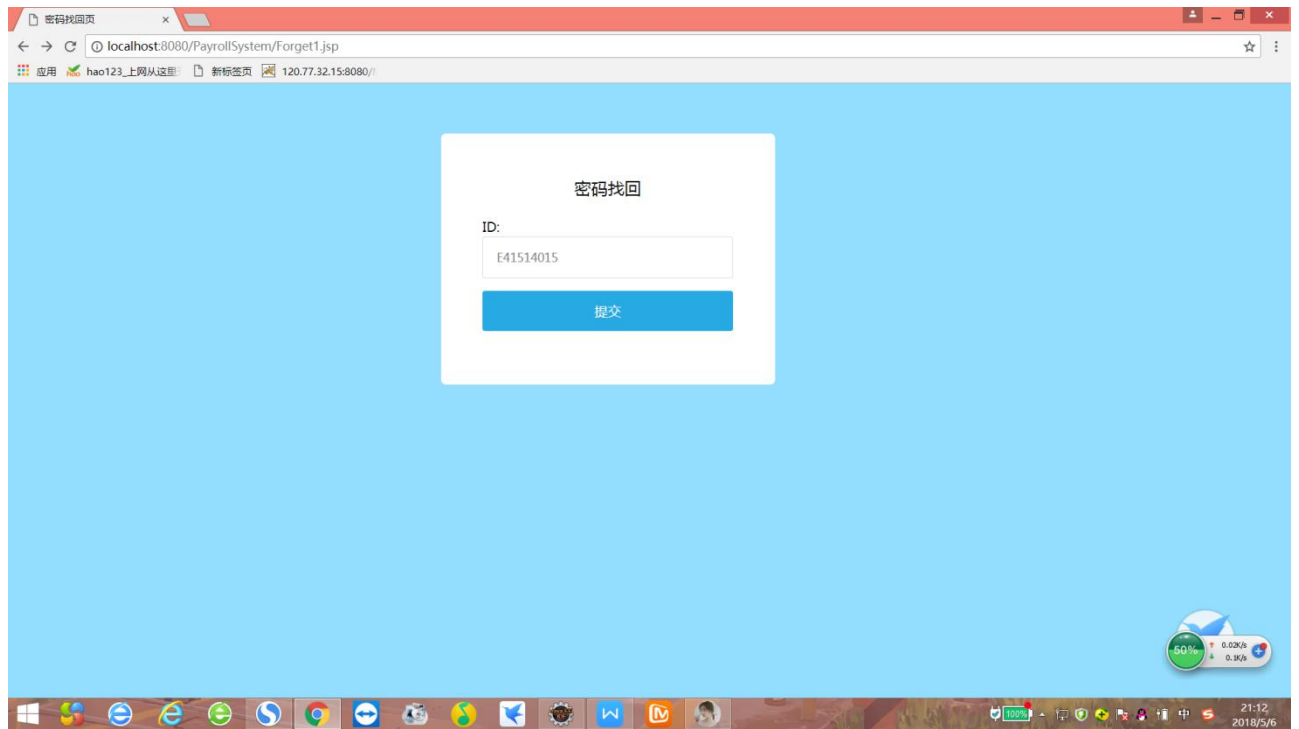
点击登录页面左下角忘记密码：



工资管理系统

☒领导 ☐员工

验证码：



输入要您的 ID:

A close-up view of the password recovery form. The form is centered on a light blue background. It features a white box with the title '密码找回' (Password Recovery) at the top. Below the title, there is a label 'ID:' followed by a text input field that contains the text 'E41514015'. At the bottom of the form is a blue button with the text '提交' (Submit).

回答三个基本问题：



密码找回

ID:

E41514015

问题一：你的本科大学是？

安徽大学

问题二：你的大学入学年级是？

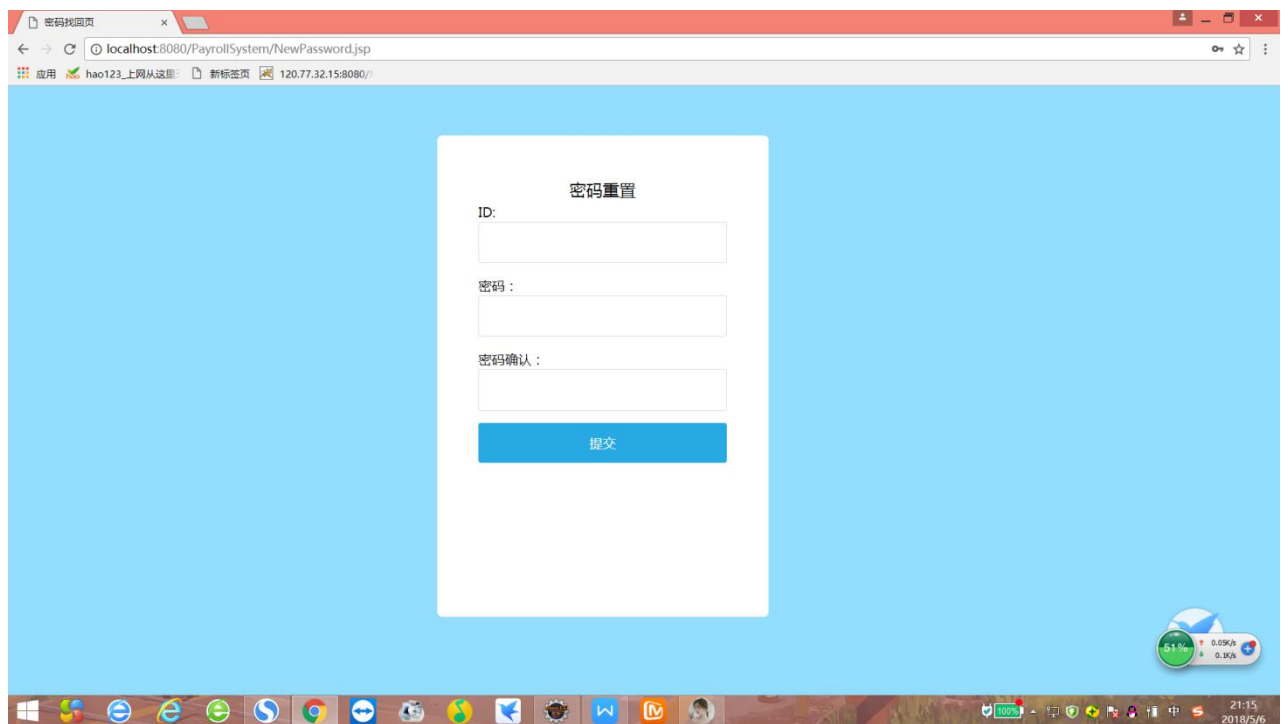
15

问题三：你的本科专业是？

信息安全

提交

核对成功，进入密码重置页面：



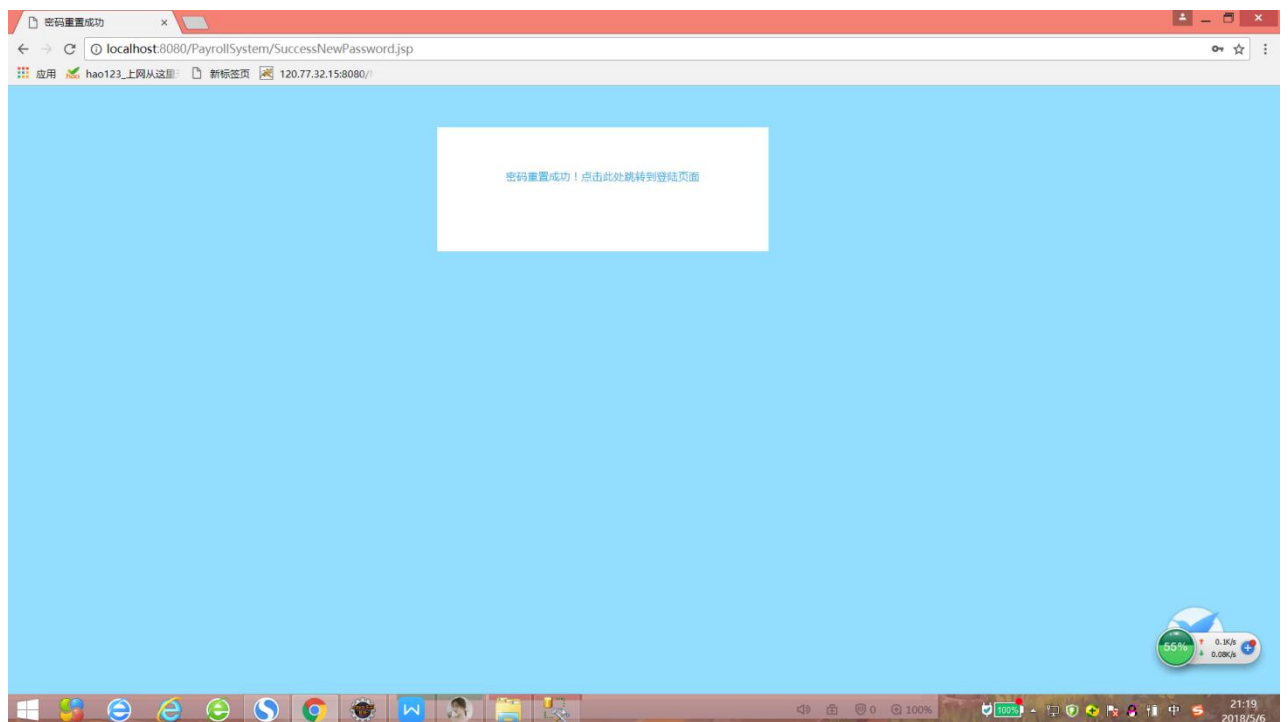
密码重置

ID:

密码:

密码确认:

提交



密码修改成功，点击此链接可跳转到登陆页面

数据库中原本的密码为 123456:

象资源管理器

LENOVO-PC\SQLEXPRESS (SQL)

数据库

系统数据库

PayrollSystem

数据库关系图

表

系统表

FileTables

dbo.Salary

dbo.Sheet1\$

dbo.Staff

视图

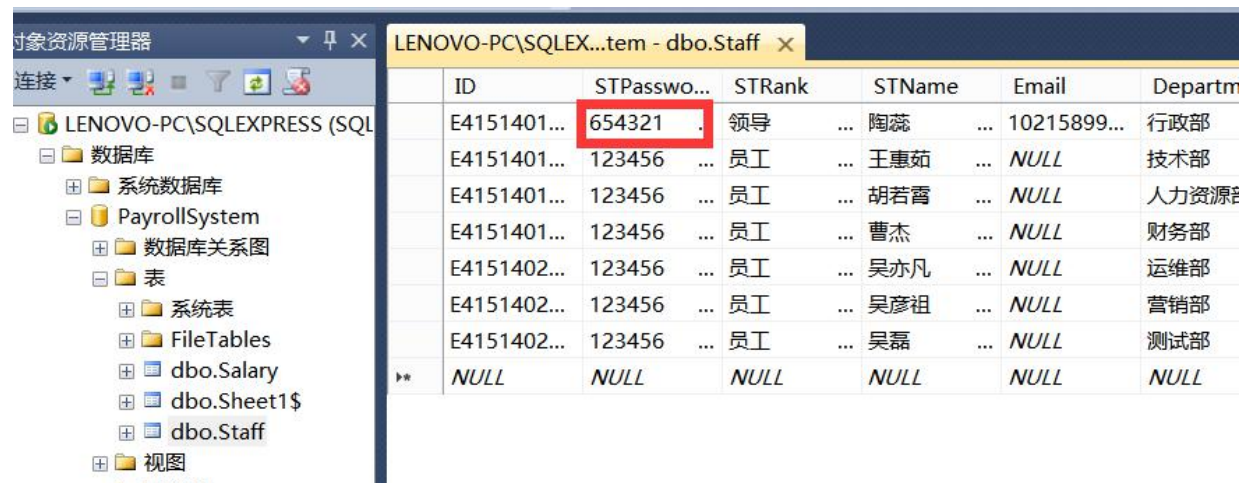
同义词

可编程性

LENOVO-PC\SQLEXPRESS...tem - dbo.Staff

ID	STPasswo...	STRank	STName	Email	Depart
E4151401...	123456	领导	陶蕊	10215899...	行政部
E4151401...	123456	员工	王惠茹	NULL	技术部
E4151401...	123456	员工	胡若霄	NULL	人力资源部
E4151401...	123456	员工	曹杰	NULL	财务部
E4151402...	123456	员工	吴亦凡	NULL	运维部
E4151402...	123456	员工	吴彦祖	NULL	营销部
E4151402...	123456	员工	吴磊	NULL	测试部
NULL	NULL	NULL	NULL	NULL	NULL

修改后的密码为 654321:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the '对象资源管理器' (Object Explorer) pane displays the database structure for 'LENOVO-PC\SQLEXPRESS (SQL Server)'. The '数据库' (Databases) folder is expanded, showing '系统数据库' (System Databases) and 'PayrollSystem'. Under 'PayrollSystem', the '数据库关系图' (Database Diagrams) folder is expanded, showing '表' (Tables). The 'dbo.Staff' table is selected. The main pane displays the 'dbo.Staff' table data. The table has columns: ID, STPasswo..., STRank, STName, Email, and Departm. The first row shows ID 'E4151401...', STPasswo... '654321' (highlighted with a red box), STRank '领导', STName '陶蕊', Email '10215899...', and Departm '行政部'. The second row shows ID 'E4151401...', STPasswo... '123456', STRank '员工', STName '王惠茹', Email 'NULL', and Departm '技术部'. The third row shows ID 'E4151401...', STPasswo... '123456', STRank '员工', STName '胡若霄', Email 'NULL', and Departm '人力资源部'. The fourth row shows ID 'E4151401...', STPasswo... '123456', STRank '员工', STName '曹杰', Email 'NULL', and Departm '财务部'. The fifth row shows ID 'E4151402...', STPasswo... '123456', STRank '员工', STName '吴亦凡', Email 'NULL', and Departm '运维部'. The sixth row shows ID 'E4151402...', STPasswo... '123456', STRank '员工', STName '吴彦祖', Email 'NULL', and Departm '营销部'. The seventh row shows ID 'E4151402...', STPasswo... '123456', STRank '员工', STName '吴磊', Email 'NULL', and Departm '测试部'. The eighth row shows ID 'NULL', STPasswo... 'NULL', STRank 'NULL', STName 'NULL', Email 'NULL', and Departm 'NULL'.

ID	STPasswo...	STRank	STName	Email	Departm
E4151401...	654321	领导	陶蕊	10215899...	行政部
E4151401...	123456	员工	王惠茹	NULL	技术部
E4151401...	123456	员工	胡若霄	NULL	人力资源部
E4151401...	123456	员工	曹杰	NULL	财务部
E4151402...	123456	员工	吴亦凡	NULL	运维部
E4151402...	123456	员工	吴彦祖	NULL	营销部
E4151402...	123456	员工	吴磊	NULL	测试部
NULL	NULL	NULL	NULL	NULL	NULL

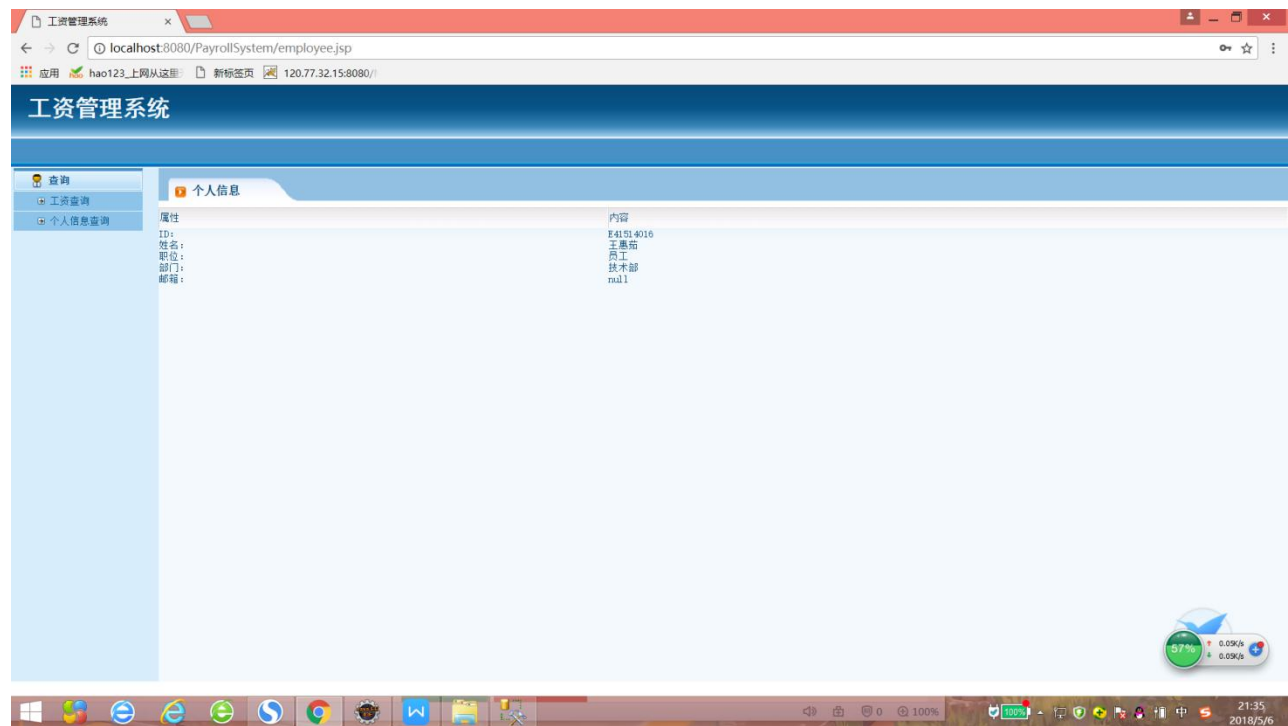
3.3 员工模式

员工模式的权限包括个人信息的查询、个人工资的查询。

3.3.1 员工模式主页总览



3.3.2 员工个人基本信息查询



3.3.3 员工个人工资信息查询

每一个员工工资信息中都存入了一年的记录:

工资管理系统

查询

- 工资查询
- 个人信息查询

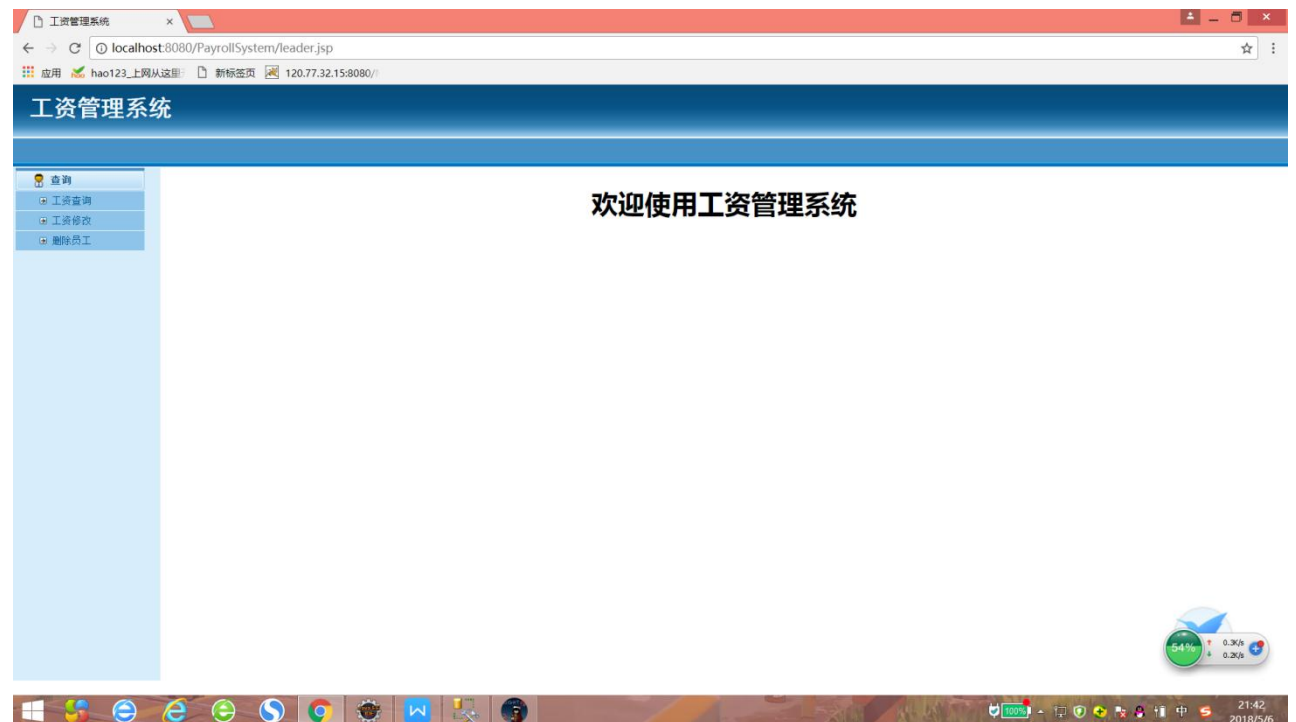
工资

ID	姓名	部门	时间	应付工资	公积金	社保	税收	加班费	实发工资	年终奖	年工资
E41514016	王惠茹	技术部	2017-1	10000.0	800.0	100.0	1500.0	2000.0	9600.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-10	10000.0	750.0	100.0	1750.0	1500.0	9375.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-11	10000.0	600.0	100.0	750.0	1500.0	9500.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-12	10000.0	600.0	100.0	750.0	1500.0	9500.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-2	10000.0	800.0	175.0	1500.0	1500.0	9100.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-3	10000.0	800.0	175.0	1500.0	1000.0	8600.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-4	10000.0	800.0	175.0	1500.0	1000.0	8600.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-5	10000.0	750.0	175.0	1750.0	2000.0	9875.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-6	10000.0	750.0	175.0	1750.0	2000.0	9875.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-7	10000.0	750.0	175.0	1750.0	2000.0	9875.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-8	10000.0	750.0	150.0	1750.0	2000.0	9875.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-9	10000.0	750.0	150.0	1750.0	1500.0	9375.0	5000.0	118150.0

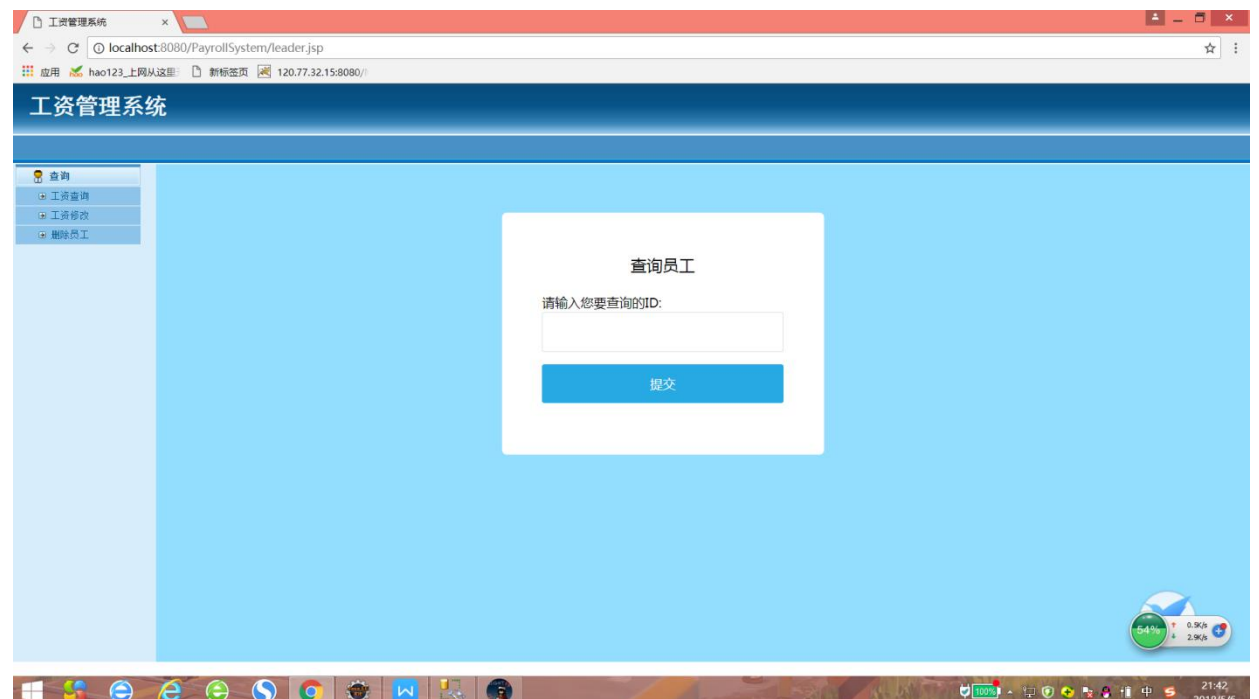
3.3 领导模式

领导模式的权限包括对员工的工资信息进行查询，删除员工，修改员工工资信息。

3.3.1 领导模式主页总览



3.3.2 员工工资查询



输入要查询的员工 ID:

查询员工

请输入您要查询的ID:

E41514016

提交

显示查询结果:

工资管理系统

localhost:8080/PayrollSystem/leader.jsp

应用hao123上网从这里新标签页120.77.32.15:8080/

工资管理系统

查询

工资查询

工资修改

删除员工

工资

ID	姓名	部门	时间	应付工资	公积金	社保	税收	加班费	实发工资	年终奖	年工资
E41514016	王惠茹	技术部	2017-1	10000.0	800.0	100.0	1500.0	2000.0	9600.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-10	10000.0	750.0	100.0	1750.0	1500.0	9375.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-11	10000.0	600.0	100.0	750.0	1500.0	9550.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-12	10000.0	600.0	100.0	750.0	1500.0	9550.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-2	10000.0	800.0	175.0	1500.0	1500.0	9100.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-3	10000.0	800.0	175.0	1500.0	1000.0	8900.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-4	10000.0	800.0	175.0	1500.0	1000.0	8900.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-5	10000.0	750.0	175.0	1750.0	2000.0	9875.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-6	10000.0	750.0	175.0	1750.0	2000.0	9875.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-7	10000.0	750.0	175.0	1750.0	2000.0	9875.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-8	10000.0	750.0	150.0	1750.0	2000.0	9875.0	5000.0	118150.0
E41514016	王惠茹	技术部	2017-9	10000.0	750.0	150.0	1750.0	1500.0	9375.0	5000.0	118150.0

55%

5.7K/s

5.8K/s

21:44

2018/5/6

若输入不存在的 ID:



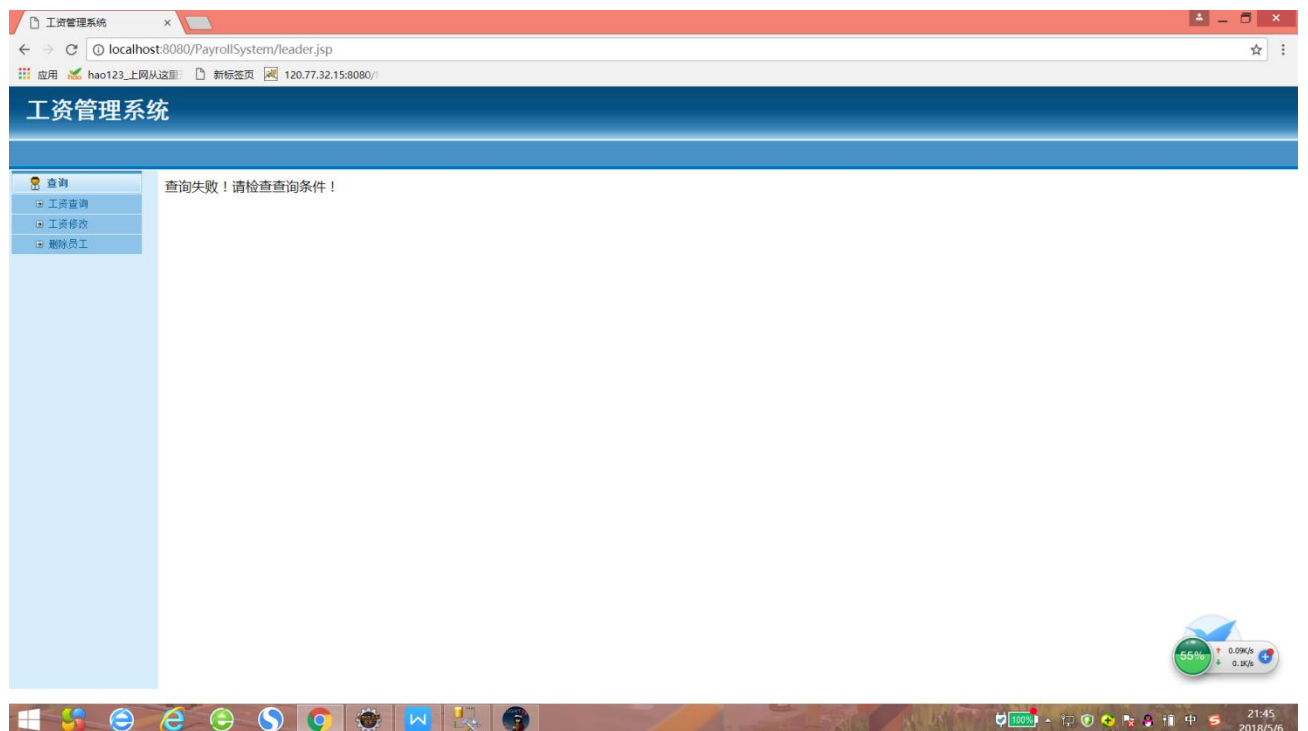
查询员工

请输入您要查询的ID:

E41514082

提交

提示查询失败!

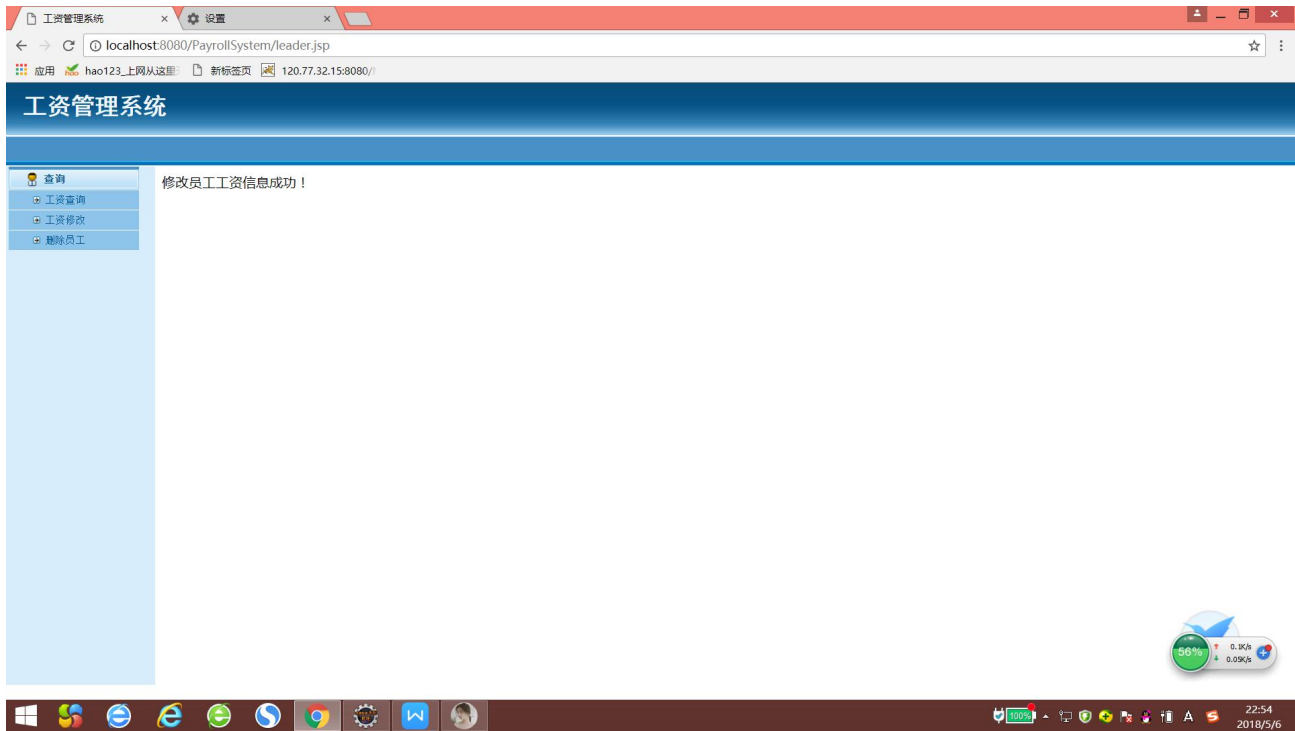


3.3.3 员工工资修改

输入要修改的员工 ID 及相关信息:



提示修改成功：



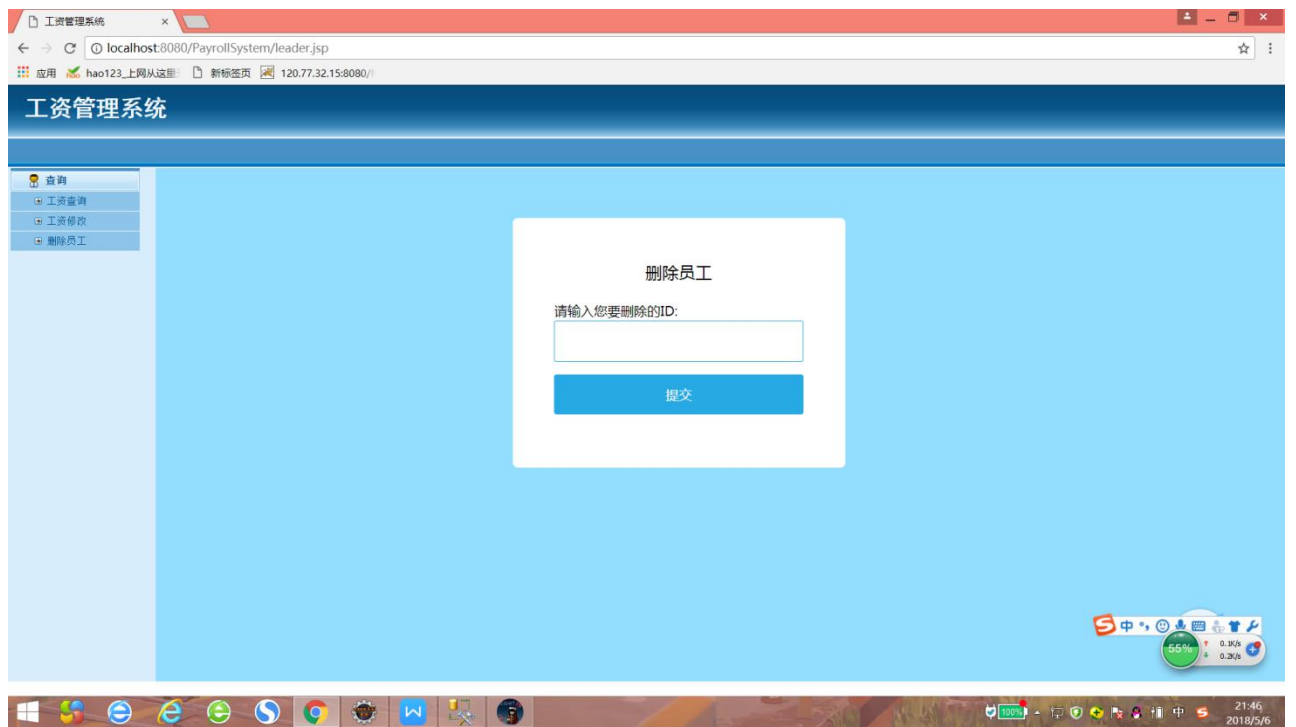
修改前的数据库：

LENOVO-PC\SQLEX...em - dbo.Salary													
	ID	STName	Departme...	Time1	Initial	Accumula...	Social	Tax	Overtime	Final	Yearbonus	Annual	
✓	E4151401...	陶蕊	...	2017-1	...	20000	1600	200	3000	4000	19200	50000	276300
	E4151401...	陶蕊	...	2017-10	...	20000	1500	200	2500	3000	18750	50000	276300
	E4151401...	陶蕊	...	2017-11	...	20000	1200	200	2500	3000	19000	50000	276300
	E4151401...	陶蕊	...	2017-12	...	20000	1200	200	2500	3000	19000	50000	276300
	E4151401...	陶蕊	...	2017-2	...	20000	1600	250	3000	3000	18200	50000	276300
	E4151401...	陶蕊	...	2017-3	...	20000	1600	250	3000	2000	17200	50000	276300
	E4151401...	陶蕊	...	2017-4	...	20000	1600	250	3000	2000	17200	50000	276300
	E4151401...	陶蕊	...	2017-5	...	20000	1500	250	2500	4000	19750	50000	276300
	E4151401...	陶蕊	...	2017-6	...	20000	1500	250	2500	4000	19750	50000	276300

修改后的数据库：

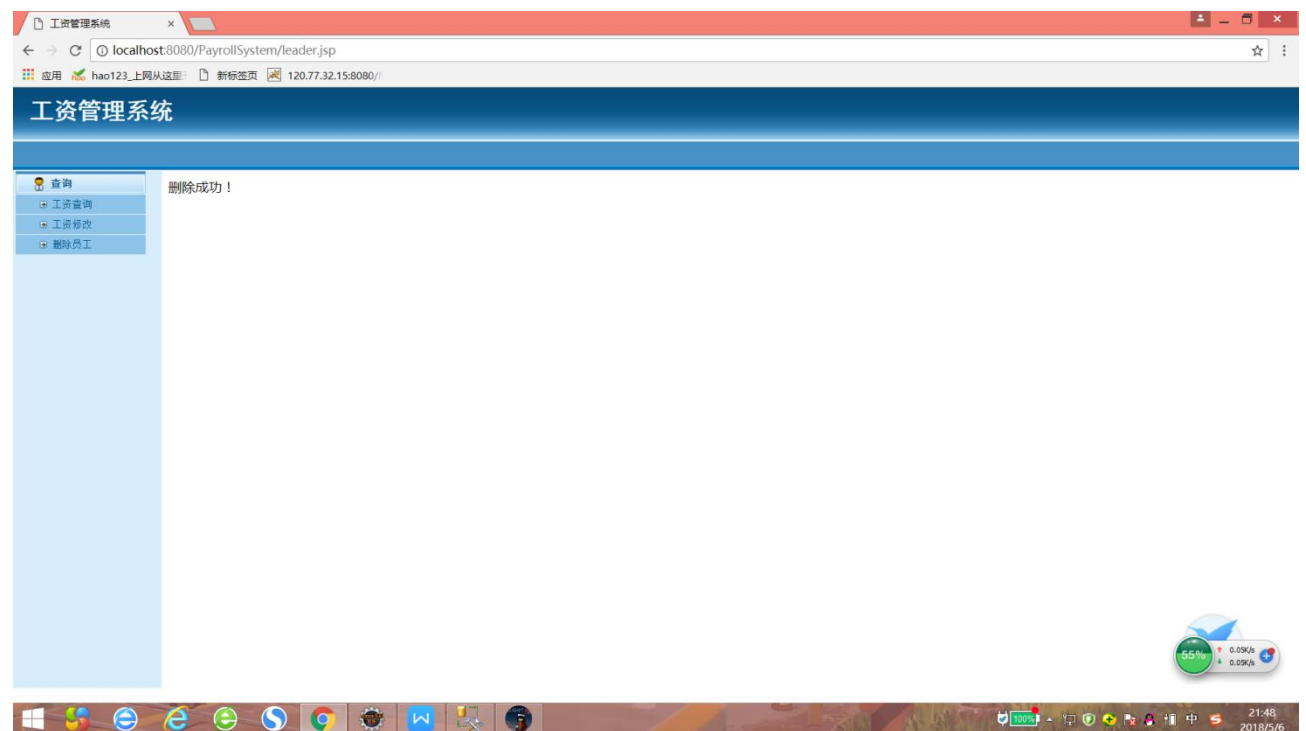
LENOVO-PC\SQLEX...em - dbo.Salary													
	ID	STName	Departme...	Time1	Initial	Accumula...	Social	Tax	Overtime	Final	Yearbonus	Annual	
	E4151401...	陶蕊	...	2017-1	...	20000	1600	200	3000	4000	19200	50000	1
	E4151401...	陶蕊	...	2017-10	...	20000	1500	200	2500	3000	18750	50000	276300
▶	E4151401...	陶蕊	...	2017-11	...	20000	1200	200	2500	3000	19000	50000	276300
	E4151401...	陶蕊	...	2017-12	...	20000	1200	200	2500	3000	19000	50000	276300
	E4151401...	陶蕊	...	2017-2	...	20000	1600	250	3000	3000	18200	50000	276300
	E4151401...	陶蕊	...	2017-3	...	20000	1600	250	3000	2000	17200	50000	276300
	E4151401...	陶蕊	...	2017-4	...	20000	1600	250	3000	2000	17200	50000	276300
	E4151401...	陶蕊	...	2017-5	...	20000	1500	250	2500	4000	19750	50000	276300
	E4151401...	陶蕊	...	2017-6	...	20000	1500	250	2500	4000	19750	50000	276300

3.3.4 删除员工



输入要删除的员工 ID:

提示删除成功！



删除前的数据库：

象资源管理器

连接

LENOVO-PC\SQLEXPRESS (SQL)

数据库

系统数据库

PayrollSystem

数据库关系图

表

系统表

FileTables

dbo.Salary

dbo.Sheet1\$

dbo.Staff

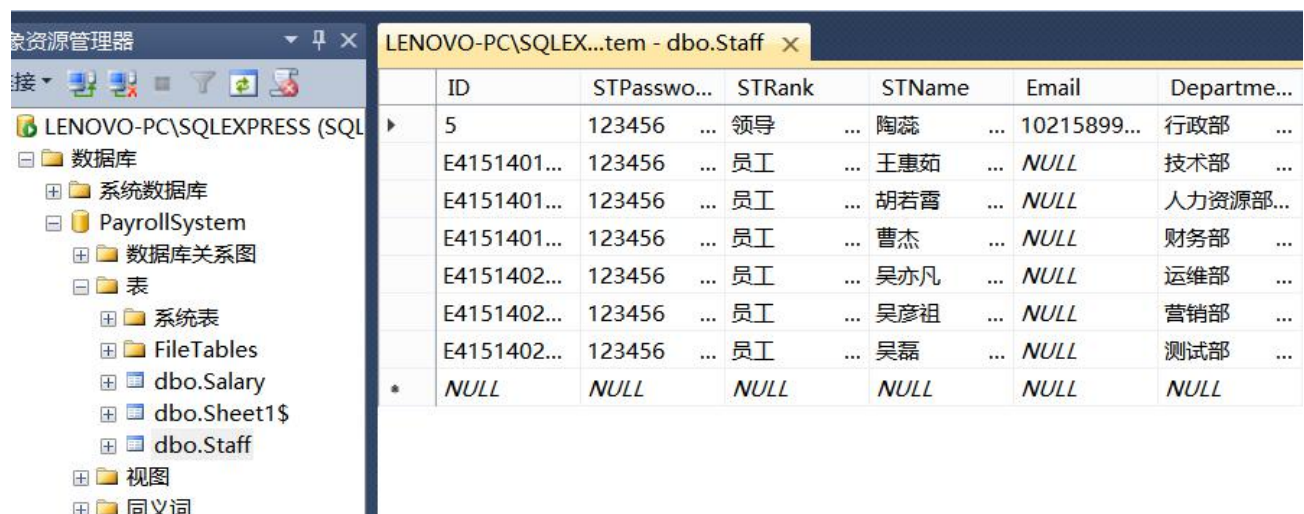
视图

同义词

LENOVO-PC\SQLEXPRESS...tem - dbo.Staff

	ID	STPasswo...	STRank	STName	Email	Departme...	Question
	E4151401...	123456	领导	陶蕊	10215899...	行政部	你的本科...
	E4151401...	123456	员工	王惠茹	NULL	技术部	NULL
	E4151401...	123456	员工	胡若霄	NULL	人力资源部	NULL
	E4151401...	123456	员工	曹杰	NULL	财务部	NULL
	E4151402...	123456	员工	吴亦凡	NULL	运维部	NULL
	E4151402...	123456	员工	吴彦祖	NULL	营销部	NULL
	E4151402...	123456	员工	吴磊	NULL	测试部	NULL
	E4151408...	123456	员工	安徽大学	ahu.qq.co...	技术部	年龄
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

删除后的数据库:



ID	STPasswo...	STRank	STName	Email	Departme...
5	123456	领导	陶淼	10215899...	行政部
E4151401...	123456	员工	王惠茹	NULL	技术部
E4151401...	123456	员工	胡若霄	NULL	人力资源部
E4151401...	123456	员工	曹杰	NULL	财务部
E4151402...	123456	员工	吴亦凡	NULL	运维部
E4151402...	123456	员工	吴彦祖	NULL	营销部
E4151402...	123456	员工	吴磊	NULL	测试部
NULL	NULL	NULL	NULL	NULL	NULL

4 安全模块

结合本专业（信息安全）知识，做了四大安全模块：

（一）、登录验证码

验证码：

（二）、数字签名认证

点击登录页面左下角数字签名：



工资管理系统

ID

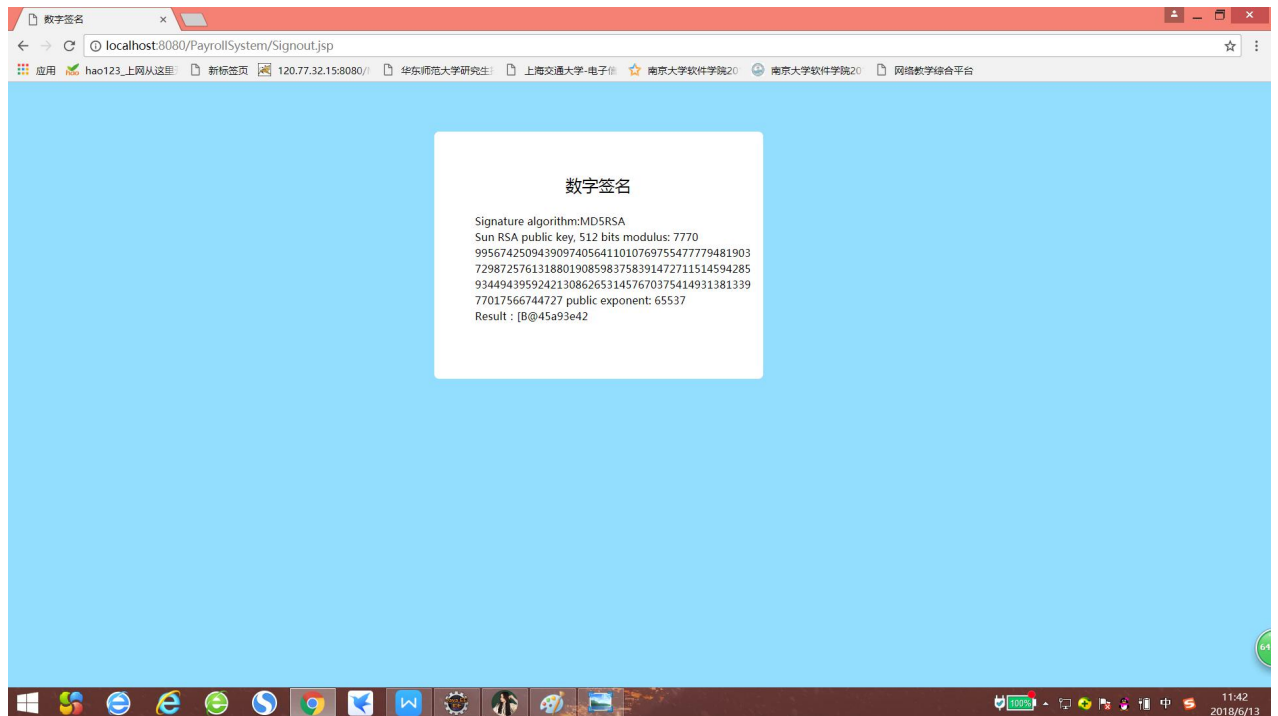
密码

☐ 领导 ☐ 员工

验证码：

登录

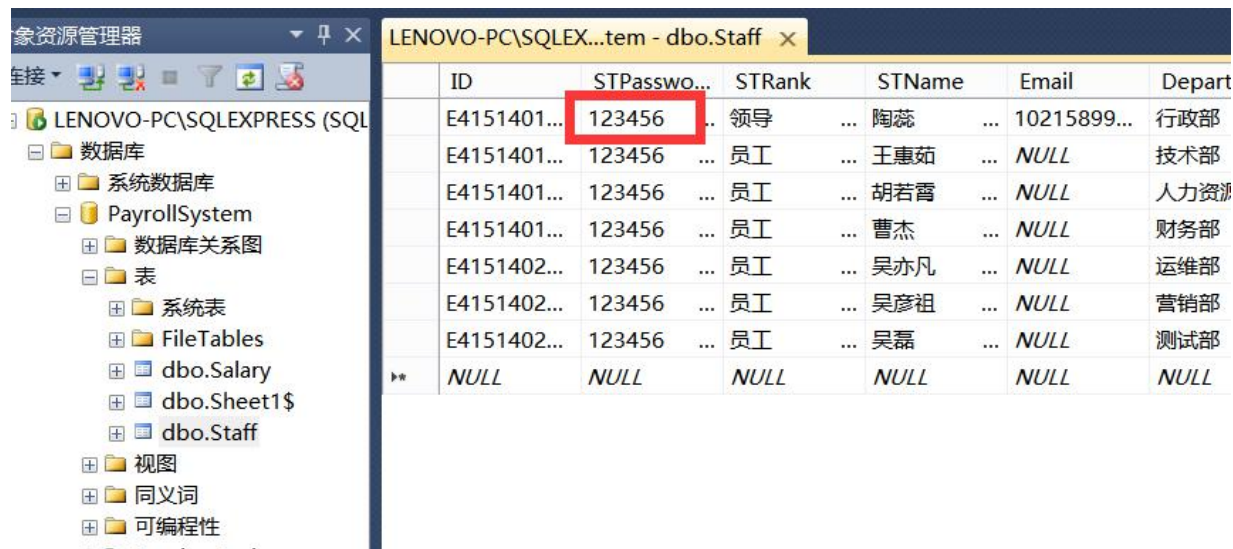
[找回密码](#) [创建账号](#) [数字签名](#)



（三）、密码加密

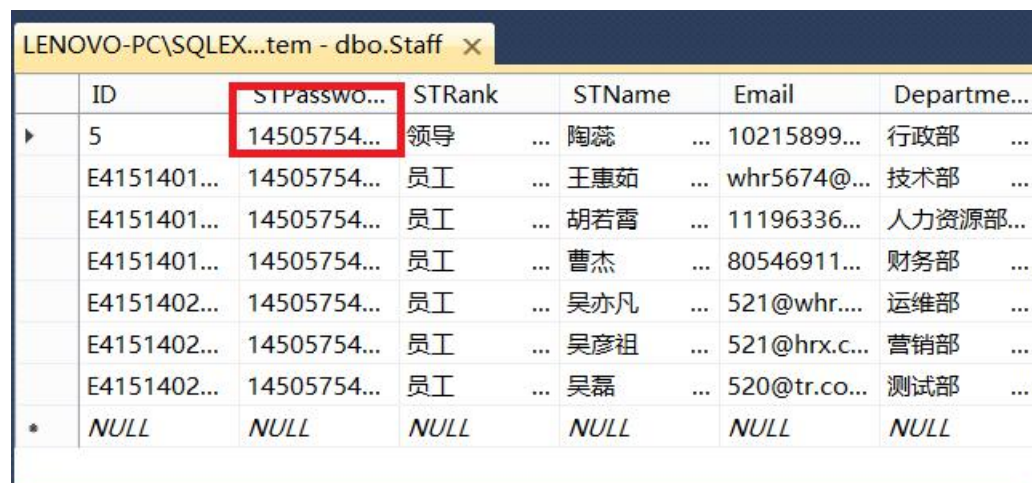
采用哈希加密将密码加密后存储在数据库中

密码加密前的数据库：



ID	STPassword	STRank	STName	Email	Depart
E4151401...	123456	领导	陶蕊	10215899...	行政部
E4151401...	123456	员工	王惠茹	NULL	技术部
E4151401...	123456	员工	胡若霄	NULL	人力资源部
E4151401...	123456	员工	曹杰	NULL	财务部
E4151402...	123456	员工	吴亦凡	NULL	运维部
E4151402...	123456	员工	吴彦祖	NULL	营销部
E4151402...	123456	员工	吴磊	NULL	测试部
NULL	NULL	NULL	NULL	NULL	NULL

密码加密后的数据库：



ID	STPassword	STRank	STName	Email	Departme...
5	14505754...	领导	陶蕊	10215899...	行政部
E4151401...	14505754...	员工	王惠茹	whr5674@...	技术部
E4151401...	14505754...	员工	胡若霄	11196336...	人力资源部
E4151401...	14505754...	员工	曹杰	80546911...	财务部
E4151402...	14505754...	员工	吴亦凡	521@whr....	运维部
E4151402...	14505754...	员工	吴彦祖	521@hrx.c...	营销部
E4151402...	14505754...	员工	吴磊	520@tr.co...	测试部
NULL	NULL	NULL	NULL	NULL	NULL

（四）、访问权限控制

登陆分为员工模式与领导模式，不同模式权限不同，很好的控制了访问权限。



5 附录：部分源代码

此处对 JavaBean、Servlet、JSP 各选取一部分进行展示：

JavaBean 选取 Staff.java 以及 StaffDAO.java：

Staff.java:

```
package beans;

public class Staff {
    private String ID;
    private String STPassword;
    private String STRank;
    private String STName;
    private String Email;
    private String Department;
    private String Question1;
    private String Answer1;
    private String Question2;
    private String Answer2;
    private String Question3;
    private String Answer3;
    public String getQuestion3() {
        return Question3;
    }
    public void setQuestion3(String question3) {
        this.Question3 = question3;
    }
    public String getAnswer3() {
        return Answer3;
    }
    public void setAnswer3(String answer3) {
        this.Answer3 = answer3;
    }
    public String getAnswer2() {
        return Answer2;
    }
    public void setAnswer2(String answer2) {
        this.Answer2 = answer2;
    }
    public String getQuestion2() {
        return Question2;
    }
    public void setQuestion2(String question2) {
        this.Question2 = question2;
    }
}
```

```

public String getAnswer1() {
    return Answer1;
}
public void setAnswer1(String answer1) {
    this.Answer1 = answer1;
}
public String getID(){
    return ID;
}
public void setID(String ID)
{
    this.ID=ID;
}
public String getQuestion1() {
    return Question1;
}
public void setQuestion1(String question1) {
    this.Question1 = question1;
}
public String getSTPassword() {
    return STPassword;
}
public void setSTPassword(String STPassword)
{
    this.STPassword=STPassword;
}
public String getSTRank(){
    return STRank;
}
public void setSTRank(String STRank)
{
    this.STRank=STRank;
}
public String getSTName(){
    return STName;
}
public void setSTName(String STName)
{
    this.STName=STName;
}
public String getEmail(){
    return Email;
}
public void setEmail(String Email)

```

```

    {
        this.Email=Email;
    }
    public String getDepartment(){
        return Department;
    }
    public void setDepartment(String Department)
    {
        this.Department=Department;
    }
}

```

StaffDAO.java:

```

package beans;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

public class StaffDAO {
/**
 * @return
 * @throws SQLException
 * @throws InstantiationException
 * @throws IllegalAccessException
 * @throws ClassNotFoundException
 */
    public static Connection getConnection()throws
SQLException,InstantiationException,IllegalAccessException,ClassNotFo
undException{

        Connection conn=null;
        Statement stat = null;
        Class.forName(DBConst.CLASSNAME).newInstance();
        String url = DBConst.URL;
        String user = DBConst.NAME;
        String password = DBConst.PASSWORD;
        conn=DriverManager.getConnection(url, user, password);

```

```

        return conn;
    }
/**
 * @param sql
 * @return
 * @throws Exception
 */
public ArrayList select(String sql)throws Exception{
    ArrayList<Staff> result=new ArrayList<Staff>();
    Connection conn=null;
    Statement stmt=null;
    ResultSet rs=null;
    try {
        conn =getConnection();
        stmt=conn.createStatement();
        rs=stmt.executeQuery(sql);
        while(rs.next()){
            Staff S=new Staff();
            S.setID(rs.getString("ID"));
            S.setSTPassword(rs.getString("STPassword"));
            S.setSTRank(rs.getString("STRank"));
            S.setSTName(rs.getString("STName"));
            S.setEmail(rs.getString("Email"));
            S.setDepartment(rs.getString("Department"));
            S.setQuestion1(rs.getString("Question1"));
            S.setQuestion2(rs.getString("Question2"));
            S.setQuestion3(rs.getString("Question3"));
            S.setAnswer1(rs.getString("Answer1"));
            S.setAnswer2(rs.getString("Answer2"));
            S.setAnswer3(rs.getString("Answer3"));
            result.add(S);
        }
    }catch(SQLException sqle)
    {
        throw new SQLException("select data
exception:"+sqle.getMessage());
    }catch(Exception e)
    {
        throw new Exception("Symtem a exception:"+e.getMessage());
    }finally{
        try{
            if(rs!=null) {
                rs.close();
            }
        }
    }
}

```

```

        }catch(Exception e){
            throw new Exception("statement close
exception:"+e.getMessage());
        }
        try {
            if(conn!=null) {
                conn.close();
            }
        }catch(Exception e) {
            throw new Exception("connection close
exception:"+e.getMessage());
        }
    }
    return result;
}

```

```

public void updatePassword(Staff S) throws Exception {
    Connection conn=null;
    PreparedStatement ps=null;
    String sql="update Staff set STPassword='"+S.getSTPassword()+"' where
ID='"+S.getID()+"'";
    System.out.println("sql"+sql);
    try {
        conn=getConnection();
        ps=conn.prepareStatement(sql);
        ps.executeUpdate();
    }catch (SQLException sqle) {
        throw new Exception("update exception:"+sqle.getMessage());
    }finally {
        try {
            if(ps!=null) {
                ps.close();
            }
        }catch(Exception e) {
            throw new Exception("ps close exception:"+e.getMessage());
        }
    }
    try {
        if(conn!=null) {
            conn.close();
        }
    }catch(Exception e) {
        throw new Exception("connection close exception:"+e.getMessage());
    }
}

```

```

    }
}

public static void insert(Staff staff) throws Exception {
    Connection conn=null;
    PreparedStatement ps=null;
    String sql="insert into
Staff(ID,STPassword,STRank,STName,Email,Department,Question1,Answer1,
Question2,Answer2,Question3,Answer3)
values('"+staff.getID()+"','"+staff.getSTPassword()+"','"+staff.getST
Rank()+"','"+staff.getSTName()+"','"+staff.getEmail()+"','"+staff.get
Department()+"','"+staff.getQuestion1()+"','"+staff.getAnswer1()+"','
"+staff.getQuestion2()+"','"+staff.getAnswer2()+"','"+staff.getQuesti
on3()+"','"+staff.getAnswer3()+"')";
    try {
        conn=getConnection();
        ps=conn.prepareStatement(sql);
        ps.executeUpdate();
    }catch(SQLException sqle) {
        throw new Exception("insert data
exception:"+sqle.getMessage());
    }finally {
        try {
            if(ps!=null)
            {
                ps.close();
            }
        }catch(Exception e) {
            throw new Exception("ps close exception:"+e.getMessage());
        }
    }
    try {
        if(conn!=null)
        {
            conn.close();
        }
    }catch(Exception e) {
        throw new Exception("connection close
exception:"+e.getMessage());
    }
}

public void delete(Staff staff)throws Exception{

```

```

Connection conn=null;
PreparedStatement ps=null;
try{
    conn=getConnection();
    String sql="delete from Staff where ID='"+staff.getID()+"'";
    ps=conn.prepareStatement(sql);
    int count=ps.executeUpdate();
    System.out.println("删除了"+count+"条记录");
}catch (SQLException sqle){
    throw new Exception("delete data exception:
"+sqle.getMessage());
}finally{
    try{
        if(ps!=null) {
            ps.close();
        }
    }catch(Exception e){
        throw new Exception("ps close exception:
"+e.getMessage());
    }
}
try{
    if(conn!=null) {
        conn.close();
    }
}catch(Exception e) {
    throw new Exception("connection close exception:
"+e.getMessage());
}
}
}

```

Servlet 选取 login.java (登陆的 servlet 后台处理) 和 LInquire.java (员工工资信息查询的后台处理) :

login.java:

```

package servlets;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;

```

```

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import beans.*;

public class Login extends HttpServlet{
    public void doGet(HttpServletRequest request,HttpServletResponse
    response)throws ServletException,IOException{
        doPost(request,response);
    }
    public void doPost(HttpServletRequest request,HttpServletResponse
    response)throws ServletException,IOException{
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        //String ip=request.getRemoteAddr();
        Staff S=new Staff();
        StaffDAO SDAO=new StaffDAO();
        Salary Sa=new Salary();
        SalaryDAO SalaryDAO=new SalaryDAO();

        String ID;
        String STPassword;
        String STRank;
        String STName;
        String SQL;
        String Email;
        String Department;
        String Time1;
        float Initial;
        float Accumulation;
        float Social;
        float Tax ;
        float Overtime ;
        float Final;
        float Yearbonus;
        float Annual;

        HttpSession session=request.getSession(true);
        String incode = (String)request.getParameter("code");
        String rightcode = (String)session.getAttribute("rCode");
        String password=request.getParameter("password");

        if(incode.equals(rightcode)){
            ;

```



```

    }else{
        response.sendRedirect("FailLogin.jsp");
    }
ArrayList<Staff> result=new ArrayList<Staff>();
ArrayList<Salary> result1=new ArrayList<Salary>();

ID=request.getParameter("ID");
STPassword=request.getParameter("STPassword");
int c=STPassword.hashCode();//哈希函数加密密码
STPassword="" +c;
STRank=request.getParameter("STRank");

if(STRank.equals("1"))
{
    STRank="领导";
}
else if(STRank.equals("2"))
{
    STRank="员工";
}
try {
    SQL="select * from Staff where ID='"+ID+"' AND
    STPassword='"+STPassword+"' AND STRank='"+STRank+"'";
    result=SDAO.select(SQL);
    S=result.get(0);
    STName=S.getSTName();
    Email=S.getEmail();
    Department=S.getDepartment();

    session.setAttribute("STName",STName);
    session.setAttribute("ID",ID);
    session.setAttribute("Email",Email);
    session.setAttribute("Department",Department);
    session.setAttribute("STRank",STRank);
    SQL="select * from Salary where ID='"+ID+"'";
    result1=SalaryDAO.select(SQL);
    int i=0;
    int j=result1.size();
    while(j!=0)
    {
        Sa=result1.get(i);
        Time1=Sa.getTime1();
        Initial=Sa.getInitial();
        Accumulation=Sa.getAccumulation();//公积金
    }
}

```

```

        Social=Sa.getSocial();//社会保障
        Tax=Sa.getTax();//个人所得税
        Overtime=Sa.getOvertime();
        Final=Sa.getFinal();
        Yearbonus=Sa.getYearbonus();
        Annual=Sa.getAnnual();
        session.setAttribute("STName"+i+"",STName);
        session.setAttribute("ID"+i+"",ID);
        session.setAttribute("Email"+i+"",Email);
        session.setAttribute("Department"+i+"",Department);
        session.setAttribute("Time1"+i+"",Time1);
        session.setAttribute("Initial"+i+"",Initial);
        session.setAttribute("Accumulation"+i+"",Accumulation);
        session.setAttribute("Social"+i+"",Social);
        session.setAttribute("Tax"+i+"",Tax);
        session.setAttribute("Overtime"+i+"",Overtime);
        session.setAttribute("Final"+i+"",Final);
        session.setAttribute("Yearbonus"+i+"",Yearbonus);
        session.setAttribute("Annual"+i+"",Annual);

        i++;
        j--;
    }
    session.setAttribute("i",i-1);
    if(result.size()!=0)
    {
        if(STRank.equals("领导"))
        {
            response.sendRedirect("leader.jsp");

        }
        else if(STRank.equals("员工"))
        {
            response.sendRedirect("employee.jsp");
        }
    }
    else
    {
        response.sendRedirect("FailLogin.jsp");
    }
} catch (Exception e) {
    e.printStackTrace();
}
out.flush();

```

```

out.close();
}
}

LInquire.java:
package servlets;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.servlet.ServletException;

import beans.SalaryDAO;
import beans.Salary;

public class LInquire extends HttpServlet{
    public void doGet(HttpServletRequest request,HttpServletResponse
response)throws ServletException,IOException{
        doPost(request,response);
    }
    public void doPost(HttpServletRequest request,HttpServletResponse
response)throws ServletException,IOException{
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html;charset=utf-8"); //设置输出编码格
式
        PrintWriter out=response.getWriter();

        Salary S=new Salary();
        SalaryDAO STUserDAO=new SalaryDAO();
        String ID;
        ID=request.getParameter("ID");
        String SQL;
        ArrayList<Salary> result=new ArrayList<Salary>();

        try {
            SQL="select * from Salary where ID='"+ID+"'";
            result=STUserDAO.select(SQL);
            if(result.size()==0)
            {
                response.sendRedirect("FailInquire.jsp");
            }
        }
    }
}

```

```

else{
    HttpSession session=request.getSession(true);
    session.setAttribute("i",result.size());
    for(int i=0;i<result.size();i++)
    {
        S=result.get(i);
        session.setAttribute("ID"+i+"",S.getID());
        session.setAttribute("STName"+i+"",S.getSTName());
        session.setAttribute("Time1"+i+"",S.getTime1());
        session.setAttribute("Initial"+i+"",S.getInitial());

        session.setAttribute("Department"+i+"",S.getDepartment());

        session.setAttribute("Accumulation"+i+"",S.getAccumulation());
        session.setAttribute("Social"+i+"",S.getSocial());
        session.setAttribute("Tax"+i+"",S.getTax());
        session.setAttribute("Overtime"+i+"",S.getOvertime());
        session.setAttribute("Final"+i+"",S.getFinal());

        session.setAttribute("Yearbonus"+i+"",S.getYearbonus());
        session.setAttribute("Annual"+i+"",S.getAnnual());
    }
    response.sendRedirect("RespLinquire.jsp");
}
}catch(Exception e) {
    e.printStackTrace();
}

out.flush();
out.close();
}
}

```

JSP 页面选用 `index.jsp` (登录页面) 和 `EmployeeSalaryIn.jsp` (员工工资信息展示页面):

Index.jsp:

```

<%@ page language="java" import="java.util.*,beans.*"
contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>

<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServ
erPort()+path+"/";

```

%>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Cache-Control" content="no-cache">
<meta http-equiv="Expires" content="0">
<base href="<%=basePath%>">

    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <!--
    <link rel="stylesheet" type="text/css" href="styles.css">
    -->
<title>登录页面</title>
<link href="style/css/login.css" rel="stylesheet" type="text/css" />
<script type="text/javascript"
src="style/js/canvas-particle.js"></script>
</head>

<body>
<div class="login_box">
    <div class="login_l_img"></div>
    <div class="login">
        <div class="login_logo"></div>
        <div class="login_name">
            <p>工资管理系统</p>
        </div>
        <form id="Login" name="Login" method="post" onsubmit=""
action="Login">
            <input name="ID" type="text" value="ID"
onfocus="this.value=''" onblur="if(this.value==''){this.value='ID'}">
            <span id="password_text"
onclick="this.style.display='none';document.getElementById('password')
```



```

        </script>

</body>
</html>

EmployeeSalaryIn.jsp:
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <!-- 包含公共的 JSP 代码片段 -->

<title>工资查询</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="style/js/jquery.js"></script>
<script type="text/javascript" src="style/js/page_common.js"></script>
<link href="style/css/common_style_blue.css" rel="stylesheet"
type="text/css">
<link rel="stylesheet" type="text/css" href="style/css/index_1.css" />

</head>
<body>
<!-- 页面标题 -->
<div id="TitleArea">
    <div id="TitleArea_Head"></div>
    <div id="TitleArea_Title">
        <div id="TitleArea_Title_Content">
             工资
        </div>
    </div>
    <div id="TitleArea_End"></div>
</div>

<!-- 主内容区域（数据列表或表单显示） -->
<div id="MainArea">
<table class="MainArea_Content" align="center" cellpadding="0"
cellspacing="0">
    <!-- 表头 -->
    <thead>
        <tr align="left" valign="middle" id="TableTitle">

```

```

        <td>ID</td>
        <td>姓名</td>
        <td>部门</td>
        <td>时间</td>
        <td>应付工资</td>
        <td>公积金</td>
        <td>社保</td>
        <td>税收</td>
        <td>加班费</td>
        <td>实发工资</td>
        <td>年终奖</td>
        <td>年工资</td>
    </tr>
</thead>
<tbody>
<%
        int
j=Integer.parseInt(session.getAttribute("i").toString());
        for(int i=0;i<=j;i++)
        {
%>
<tr>
        <td><%=session.getAttribute("ID"+i+"")%></td>
        <td><%=session.getAttribute("STName"+i+"")%></td>
        <td><%=session.getAttribute("Department"+i+"")%></td>
        <td><%=session.getAttribute("Time1"+i+"")%></td>
        <td><%=session.getAttribute("Initial"+i+"")%></td>
        <td><%=session.getAttribute("Accumulation"+i+"")%></td>
        <td><%=session.getAttribute("Social"+i+"")%></td>
        <td><%=session.getAttribute("Tax"+i+"")%></td>
        <td><%=session.getAttribute("Overtime"+i+"")%></td>
        <td><%=session.getAttribute("Final"+i+"")%></td>
        <td><%=session.getAttribute("Yearbonus"+i+"")%></td>
        <td><%=session.getAttribute("Annual"+i+"")%></td>
</tr>
<%
        }
%>
</tbody>
</div>
</body>
</html>

```