**目录**

[1.系统开发平台 1](#_Toc91186110)

[2.数据库规划 1](#_Toc91186111)

[2.1 任务陈述 1](#_Toc91186112)

[2.2 任务目标 1](#_Toc91186113)

[3.系统定义 2](#_Toc91186114)

[3.1 系统边界 2](#_Toc91186115)

[3.2 用户视图 2](#_Toc91186116)

[4.需求分析 3](#_Toc91186117)

[4.1 用户需求说明 3](#_Toc91186118)

[4.1.1 数据需求 3](#_Toc91186119)

[4.1.2 事务需求 4](#_Toc91186120)

[4.2 系统需求说明 6](#_Toc91186121)

[4.2.1软件环境 6](#_Toc91186122)

[4.2.2硬件环境 6](#_Toc91186123)

[4.2.3初始数据库大小 6](#_Toc91186124)

[4.2.4数据库增长速度 6](#_Toc91186125)

[4.2.5记录查找的类型和平均数量 6](#_Toc91186126)

[4.2.6性能 7](#_Toc91186127)

[4.2.7安全性 7](#_Toc91186128)

[5.数据库逻辑设计 7](#_Toc91186129)

[5.1 E-R图 7](#_Toc91186130)

[5.2 数据字典 12](#_Toc91186131)

[5.2.1实体和联系属性 12](#_Toc91186132)

[5.2.2属性描述 13](#_Toc91186133)

[5.3 关系表 16](#_Toc91186134)

[6.数据库物理设计 17](#_Toc91186135)

[6.1 索引 17](#_Toc91186136)

[6.2 视图 18](#_Toc91186137)

[6.3 安全机制 18](#_Toc91186138)

[6.3.1系统安全 18](#_Toc91186139)

[6.3.2数据安全 19](#_Toc91186140)

[7.应用程序设计 19](#_Toc91186141)

[7.1 功能模块 19](#_Toc91186142)

[7.1.1计算功能 19](#_Toc91186143)

[7.1.2存储功能 19](#_Toc91186144)

[7.1.3输入及输出功能 19](#_Toc91186145)

[7.1.4交互功能 19](#_Toc91186146)

[7.2 界面设计 20](#_Toc91186147)

[7.2.1管理员登录界面 20](#_Toc91186148)

[7.2.2管理员用户界面 20](#_Toc91186149)

[7.2.3员工登录界面 30](#_Toc91186150)

[31](#_Toc91186151)

[7.2.4员工用户界面 31](#_Toc91186152)

[7.3 重要代码 34](#_Toc91186153)

[7.3.1触发器 34](#_Toc91186154)

[7.3.2 多表联接 36](#_Toc91186155)

[8.测试和运行 37](#_Toc91186156)

[9.总结 38](#_Toc91186157)

[参考文献 38](#_Toc91186158)

# 1.系统开发平台

**题目：**员工人事档案管理系统

**数据库：**MySQL

**应用开发工具：**Eclipse

**操作系统：**Windows 10

# 2.数据库规划

## 2.1 任务陈述

结合公司实际的职工、制度，经过实际的需求分析建立一个员工人事档案管理系统, 主要实现对员工的职工档案资料进行有效的录入、删除、修改和查询，根据职工编号对符合条件的员工档案资料进行修改以及根据职工编号对符合条件的员工档案资料进行删除，根据职工编号查询符合条件的员工档案资料等功能。

## 2.2 任务目标

1. 登录功能

实现系统管理员、普通用户的登录

1. 普通用户相应功能

查询自己的个人信息

1. 管理员用户相应功能

添加，删除，修改，查询系统的各个模块

1. 查询功能

系统管理员浏览信息，查看某个员工的信息等。

1. 调动子系统

修改员工基本表中相应信息，录入调动信息。

1. 离退休处理子系统

修改员工基本表中相应信息，录入离退休信息。

1. 系统维护

界面友好，美观，操作方便。

# 3.系统定义

## 3.1 系统边界

员工人事档案管理系统

**员工**

**员工人事档案管理系统**

人事调度、奖惩记录等

查询

查询

记录

**员工信息模块**

查询

公司员工数

**人事档案管理模块**

查询

**管理员**

审核&管理

更新&查询更新

更新

提交员工人事档案表更新

更新

属于

更新

图1. 系统边界

## 3.2 用户视图

表1 各类用户功能需求

|  |  |
| --- | --- |
| 用户视图 | 功能需求 |
| 员工基本信息 | 列出员工的个人基本信息  列出员工的工作信息  输出该员工的员工状态 |
| 部门信息 | 输出该员工所在部门编号  输出该员工所在部门名称  输出该员工所在部门负责人  输出该员工所在上级部门编号 |
| 用户信息 | 输入用户的用户名  输入用户的密码  输入用户的用户类型 |
| 调动信息 | 列出员工的序号  列出员工的员工编号  输入员工的调动日期  输入员工的调入部门  输入员工的调出部门  输入员工的调动原因  更新调动信息表 |
| 系统管理员 | 输入用户的用户名  输入用户的密码  输入用户的用户类型  列出员工的序号  列出员工的员工编号  输入员工的调动日期  输入员工的调入部门  输入员工的调出部门  输入员工的调动原因  更新用户信息表 |
| 普通用户 | 输入用户的用户名  输入用户的密码  输入用户的用户类型  列出员工的基本信息  列出员工的部门信息  输出该员工的员工状态 |

# 4.需求分析

## 4.1 用户需求说明

### 4.1.1 数据需求

**(1)普通用户视图:**

1. 员工基本信息

工作证号，姓名，性别，民族，生日，政治面貌，文化程度，婚姻情况，身份证号码，手机号码，邮箱，籍贯地区邮政编码，户口所在地地区邮政编码

1. 员工工作信息

工作证号，入职日期，员工状态（1-在职，2-兼职，3-试用，4-离职，5-返聘，6-退休），岗位职务编号

1. 员工（登录）信息

旧密码，新密码

1. 调动信息

序号，工作证号，调动原因，调动日期，调入部门编号，调出部门编号

1. 奖惩信息

序号，工作证号，奖惩时间，奖惩原因，奖惩金额

1. 员工薪资明细

工作证号，基本工资，奖励金额，惩罚金额，工资薪资所得

工作证号唯一，工作信息由工作证号唯一确定。普通用户（即员工）只可以对员工基本信息、员工登录信息进行修改。

**（2）管理用户视图:**

1. 员工基本信息

工作证号，姓名，性别，民族，生日，政治面貌，文化程度，婚姻情况，身份证号码，手机号码，邮箱，籍贯地区邮政编码，户口所在地地区邮政编码

1. 员工工作信息

工作证号，入职日期，员工状态（1-在职，2-兼职，3-试用，4-离职，5-返聘，6-退休），岗位职务编号

1. 员工（登录）信息

工作证号，密码

1. 部门信息

部门编号，部门名称，部门负责人员工编号，上级部门编号

1. 调动信息

序号，工作证号，调动原因，调动日期，调入部门编号，调出部门编号

1. 奖惩信息

序号，工作证号，奖惩时间，奖惩原因，奖惩金额

1. 离职信息

工作证号，离职原因，离职时间

1. 员工薪资明细

工作证号，基本工资，奖励金额，惩罚金额，工资薪资所得

1. 管理员登录

管理员工号，密码

管理员工号唯一；员工管理者可对员工基本信息、员工工作信息、员工登录信息、部门信息、调动信息、奖惩信息、离职信息、员工薪资明细、管理员登录信息进行修改和查询。

### 4.1.2 事务需求

**1) 普通用户(员工)视图:**

数据查询:列出该普通用户的详细的个人基本信息

列出该普通用户的奖惩信息

列出该普通用户的详细工作信息

列出该普通用户的岗位调动信息

列出该普通用户的薪资明细

数据更新:更新该普通用户的个人基本信息

更新该普通用户的登录信息

数据输出:输出该用户的详细个人基本信息

输出该用户的登录信息

**(2)管理用户视图:**

数据查询:列出普通用户的详细个人基本信息

列出普通用户的详细工作信息

列出普通用户的登录信息

列出各部门的基本信息

列出普通用户的奖惩记录

列出所有部门的离职信息

列出员工调动信息

列出员工的薪资信息

数据录入:录入员工登陆信息

录入管理员登陆信息

录入部门信息

录入员工个人基本信息

录入员工工作信息

录入员工工资信息

录入员工奖惩信息

录入员工离职信息

录入员工调动信息

数据更新:更新员工信息

更新部门信息

更新部门信息

更新员工工资信息

更新员工离职信息

更新员工调动信息

数据删除:删除部门信息

删除员工基本信息

删除员工登录信息

删除员工工作信息

删除员工调动信息

删除员工离职信息

删除员工奖惩信息

数据输出:输出员工基本信息

输出员工工作信息

输出员工登录信息

输出部门信息

输出员工奖惩信息

输出员工工资信息

输出员工离职信息

## 4.2 系统需求说明

**4.2.1软件环境**

软件环境需要支持Window 10操作系统，及SQL Server 2008数据库

**4.2.2硬件环境**

AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz, 16GB内存,2512G SSD。

**4.2.3初始数据库大小**

大约有在职员工10名，兼职员工5名，试用员工5名，离职员工3名，返聘员工3名，退休员工3名，管理用户3名；

包含3个部门。

**4.2.4数据库增长速度**

员工为每一年添加10名左右，同时删除5名左右；其他信息增长速度不定。

**4.2.5记录查找的类型和平均数量**

查找员工基本信息的情况………大约每天20次

查找部门信息的情况………大约每天10次

查找用户信息的情况………大约每天30次

**4.2.6性能**

单个记录查询时间少于1秒，高峰期少于5秒

多个记录查询时间少于5秒，高峰期少于10秒

更新/保存记录时间少于1秒，高峰期少于5秒

**4.2.7安全性**

每个用户在进入系统时都必须有口令保护；

每个用户分配特定的用户视图所应有的访问权限；

不同用户根据其事先定义的角色不同进入不同的用户视图界面。

# 5.数据库逻辑设计

## 5.1 E-R图

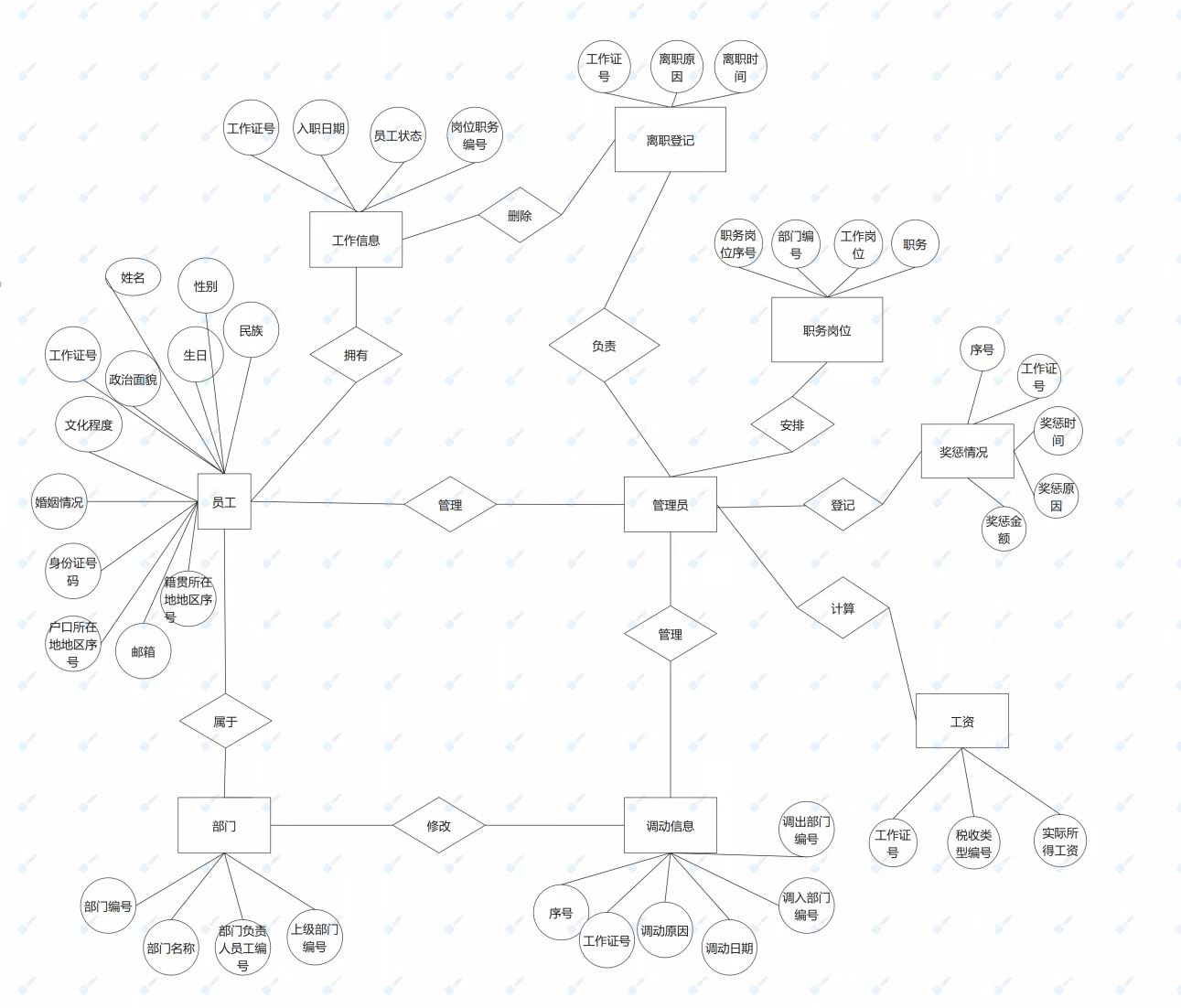


图2 系统E-R图（林齐）

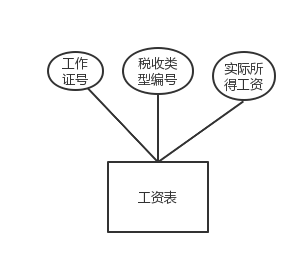


图3 工资表E-R图（林齐）

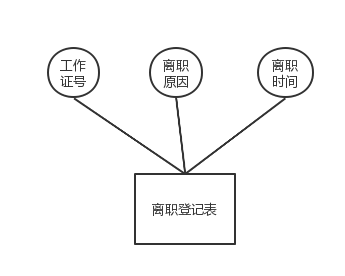


图4 离职登记表E-R图（林齐）

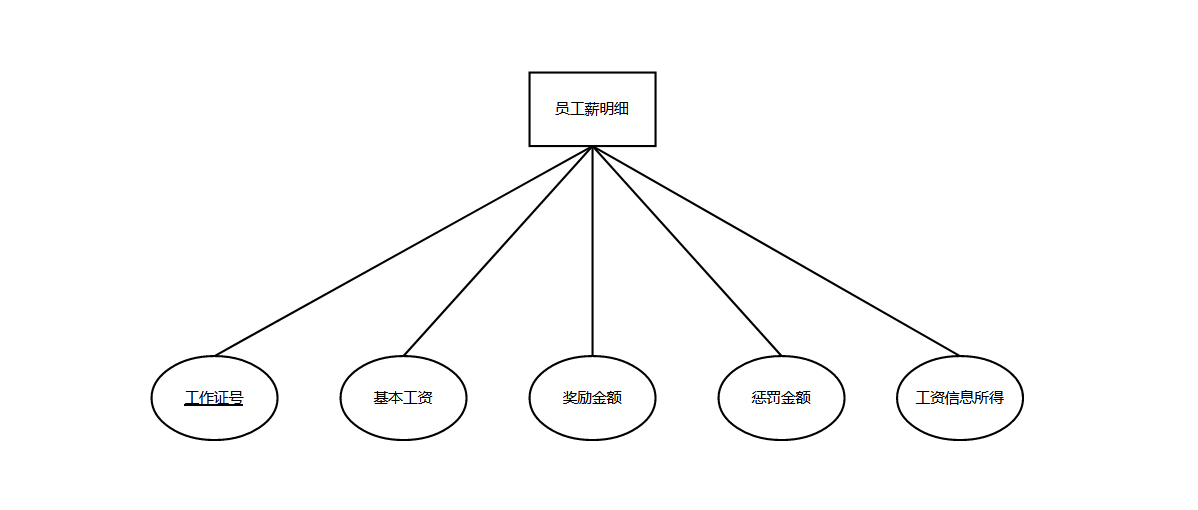


图5 员工薪资明细表E-R图（王佳沁）

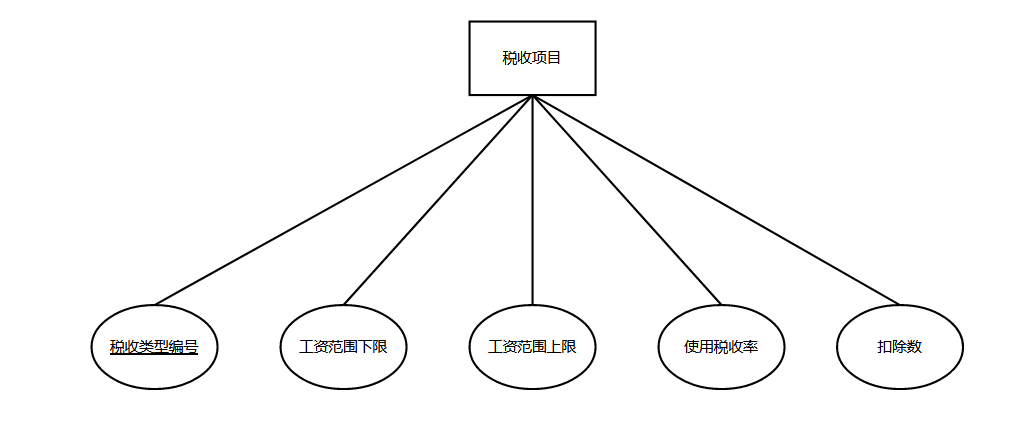


图6 税收项目表E-R图（王佳沁）

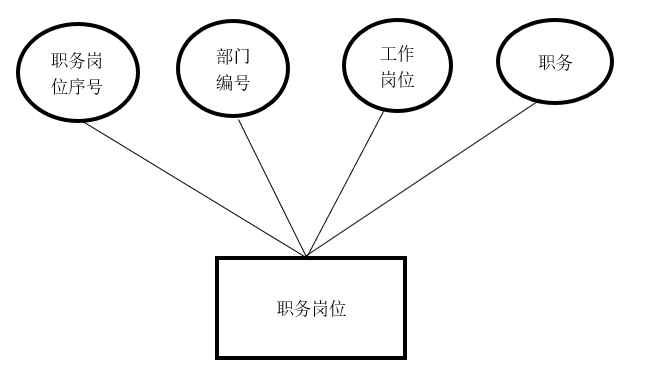
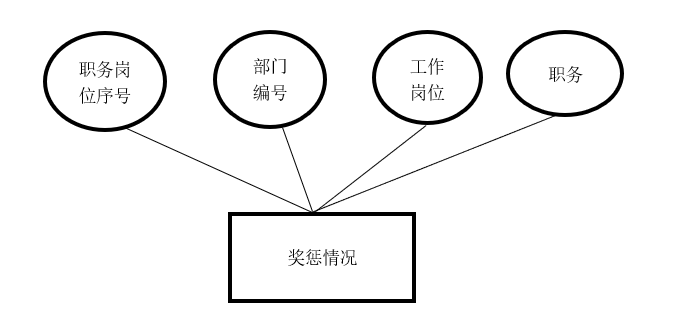


图7 职务岗位表E-R图（刘思琦）

图8 奖惩情况表E-R图（刘思琦）

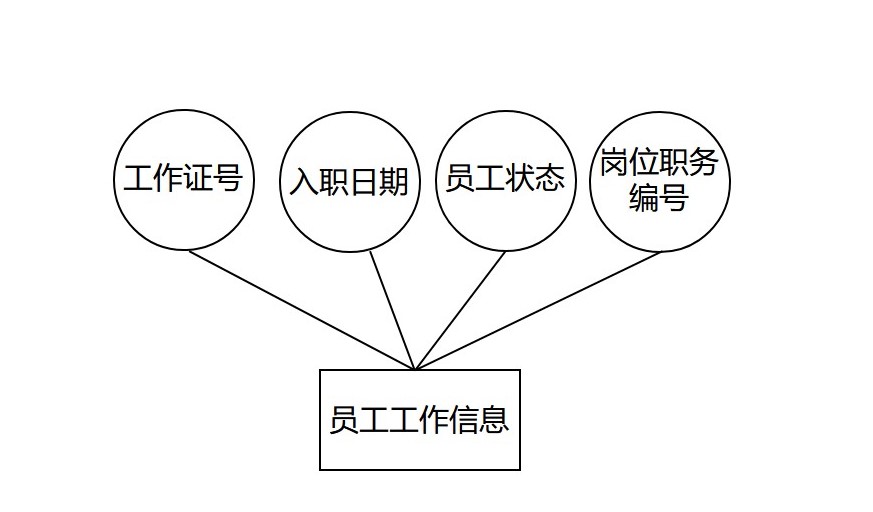


图9 员工工作信息表E-R图（王祺）

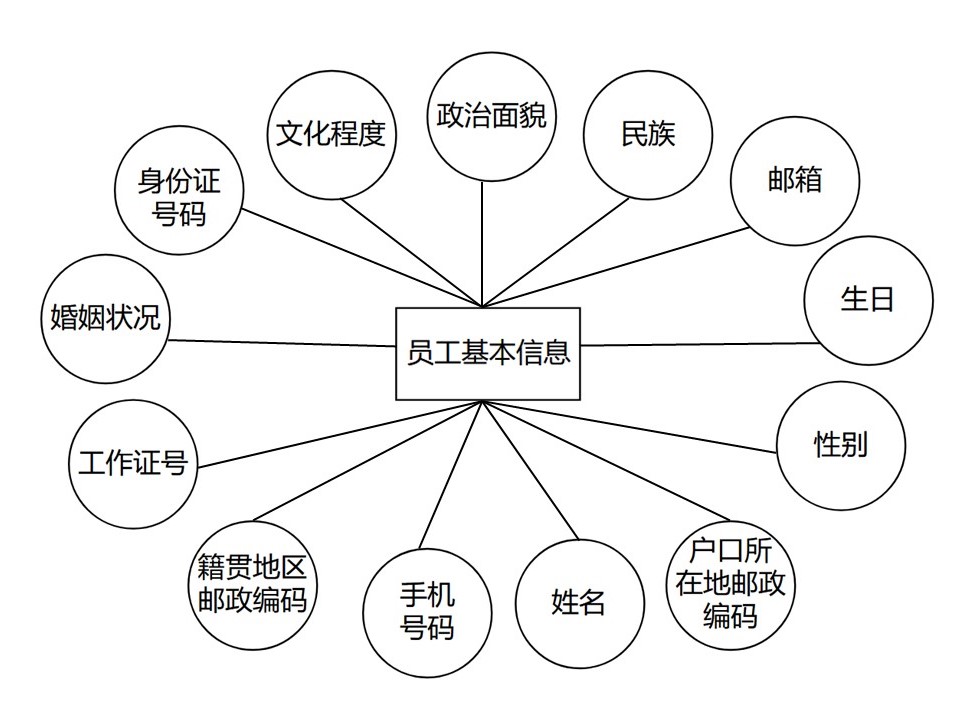


图10 员工基本信息表E-R图（王祺）

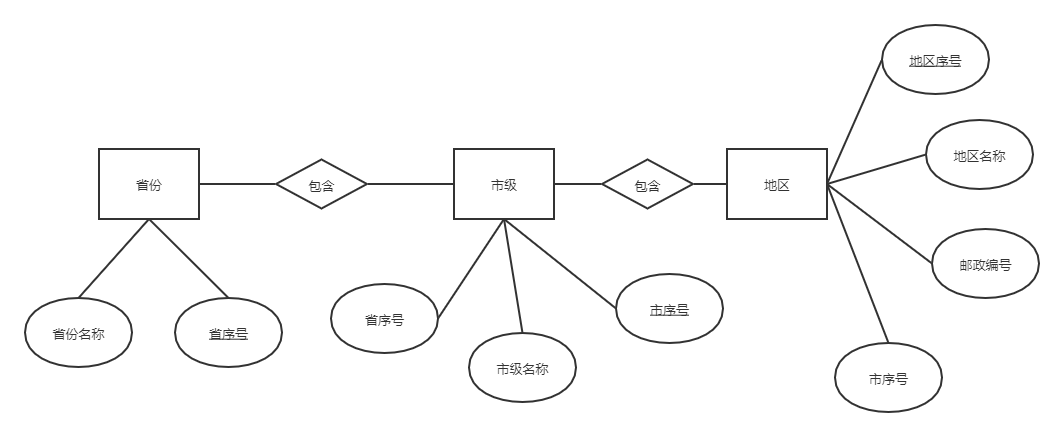


图11 地方表E-R图（洪佳滢）

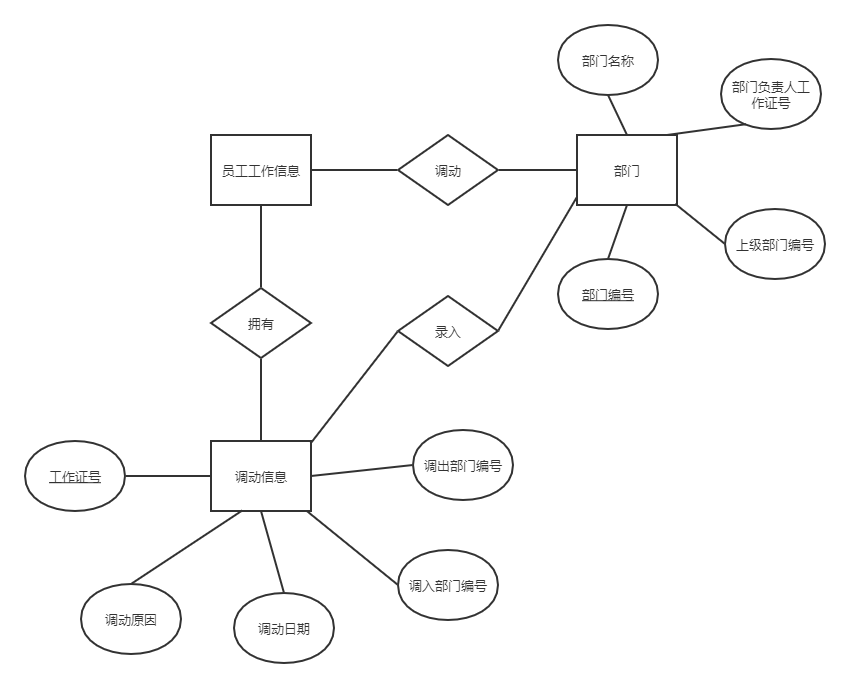


图12 调动信息表E-R图（洪佳滢）

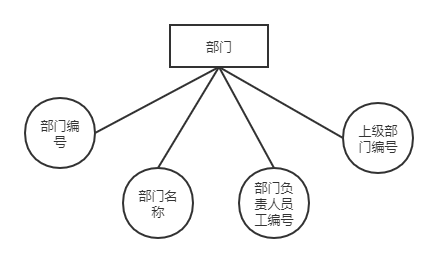


图13 部门表E-R图（刘纯瑩）

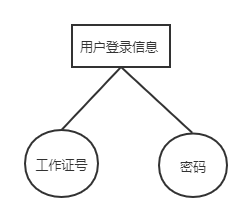


图14 部门表E-R图（刘纯瑩）

## 5.2 数据字典

### 5.2.1实体和联系属性

5.2.1 实体和属性的联系

普通用户user： 用户编号UNo，用户姓名UName，用户性别USex，用户照片UPhoto，用户民族UNation，用户生日UBirthday，用户政治面貌UStatus，用户文化程度UEdu，用户婚姻情况 UMarriage，用户籍贯UBorn，用户身份证号UId，用户手机号码UPhone，用户邮箱UMail，用户户口所在地UResidence，用户入职日期UEntry，用户工作岗位UPosition，用户职务UDuty，用户部门编号DNo；

系统管理员Employee：员工调动日期ETransferDate，员工调入部门EEnterDepartment，员工调出部门EExitDepartment；

部门department： 部门编号DNo，部门名称DName，部门负责人DPrincipal；

信息登记表sheet： 用户名UserName，密码Password，用户类型UserType；

### 5.2.2属性描述

表2 关系属性说明

|  |  |  |  |
| --- | --- | --- | --- |
| 实体 | 属性 | 类型及大小 | 空 |
| 员工基本信息表  information | 工作证号emp\_id | int | 否 |
| 姓名emp\_name | varchar(225) | 否 |
| 性别gender | char(1) | 否 |
| 生日emp\_bir | date | 否 |
| 手机号码phone\_num | varchar(225) | 否 |
| 身份证号码id\_card | varchar(225) | 否 |
| 政治面貌political\_status | varchar(50) | 否 |
| 婚姻情况 marital\_status` | char(1) | 否 |
| 籍贯地区邮政编码native\_place\_code | int | 否 |
| 文化程度culture\_level | varchar(50) | 否 |
| 邮箱email | varchar(225) | 否 |
| 户口所在地地区邮政编码registered\_residence\_code | int | 否 |
| 民族minority | varchar(50) | 否 |
| 部门表  department | 部门编号dept\_id | int | 否 |
| 部门名称dept\_name | varchar(225) | 否 |
| 部门负责人员工编号main\_emp\_id | int | 否 |
| 上级部门编号sup\_dept\_id | int | 否 |
| 调动信息表  transfer | 序号trans\_id | int | 否 |
| 工作证号trans\_emp\_id | int | 否 |
| 调动原因trans\_reason | text | 否 |
| 调动日期trans\_date | date | 否 |
| 调入部门编号in\_dept\_id | Int | 否 |
| 调出部门编号out\_dept\_id | int | 否 |
| 员工工作信息表  work\_info | 工作证号work\_emp\_id | int | 否 |
| 入职日期work\_date | date | 否 |
| 员工状态emp\_status | int | 否 |
| 岗位职务编号work\_status\_id | int | 否 |
| 员工（登录）信息表  emp\_login | 工作证号emp\_id | int | 否 |
| 密码emp\_passwords | varchar(225) | 否 |
| 省份表  province | 省序号province\_id | int | 否 |
| 省份名称province\_name | varchar(64) | 否 |
| 市级表  city | 市序号city\_id | int | 否 |
| 市级名称city\_name | varchar(64) | 否 |
| 省序号provinceId | int | 否 |
| 地区表  area | 地区序号area\_id | int | 否 |
| 地区名称area\_name | varchar(64) | 否 |
| 市序号city\_id | int | 否 |
| 邮政编码postal\_code | char(6) |  |
| 职务岗位表  station | 职务岗位序号work\_station\_id | int | 否 |
| 部门编号dept\_id | int | 否 |
| 工作岗位work\_status | varchar(225) | 否 |
| 职务duty | varchar(225) | 否 |
| 奖惩表  reward | 序号reward\_id | int | 否 |
| 奖惩时间reward\_date | date | 否 |
| 工作证号reward\_emp\_id | int | 否 |
| 奖惩金额reward\_mon | double | 否 |
| 奖惩原因reward\_reason | text | 否 |
| 离职登记表  dimission | 工作证号dim\_emp\_id | int | 否 |
| 离职原因dim\_reason | text | 否 |
| 离职时间dim\_date | date | 否 |
| 工资表  payroll | 工作证号pay\_emp\_id | int | 否 |
| 税收类型编号tax\_type\_id | int | 否 |
| 实际所得工资act\_salary | double | 否 |
| 员工薪资明细表emp\_salary | 工作证号sal\_emp\_id | int | 否 |
| 基本工资basic\_salary | double | 否 |
| 奖励金额reward\_sal | double |  |
| 惩罚金额punish\_sal | double |  |
| 工资薪资所得emp\_salary | double | 否 |
| 税收项目表  sal\_tax | 税收类型编号tax\_id | int | 否 |
| 工资范围下限smallest\_sal | double | 否 |
| 工资范围上限highest\_sal | double |  |
| 使用税收率tax\_rate | double | 否 |
| 扣除数deduction\_sal | double | 否 |
| 管理员登录表  admin\_login | 管理员工号admin\_id | int | 否 |
| 密码admin\_passwords | varchar(225) | 否 |

## 5.3 关系表

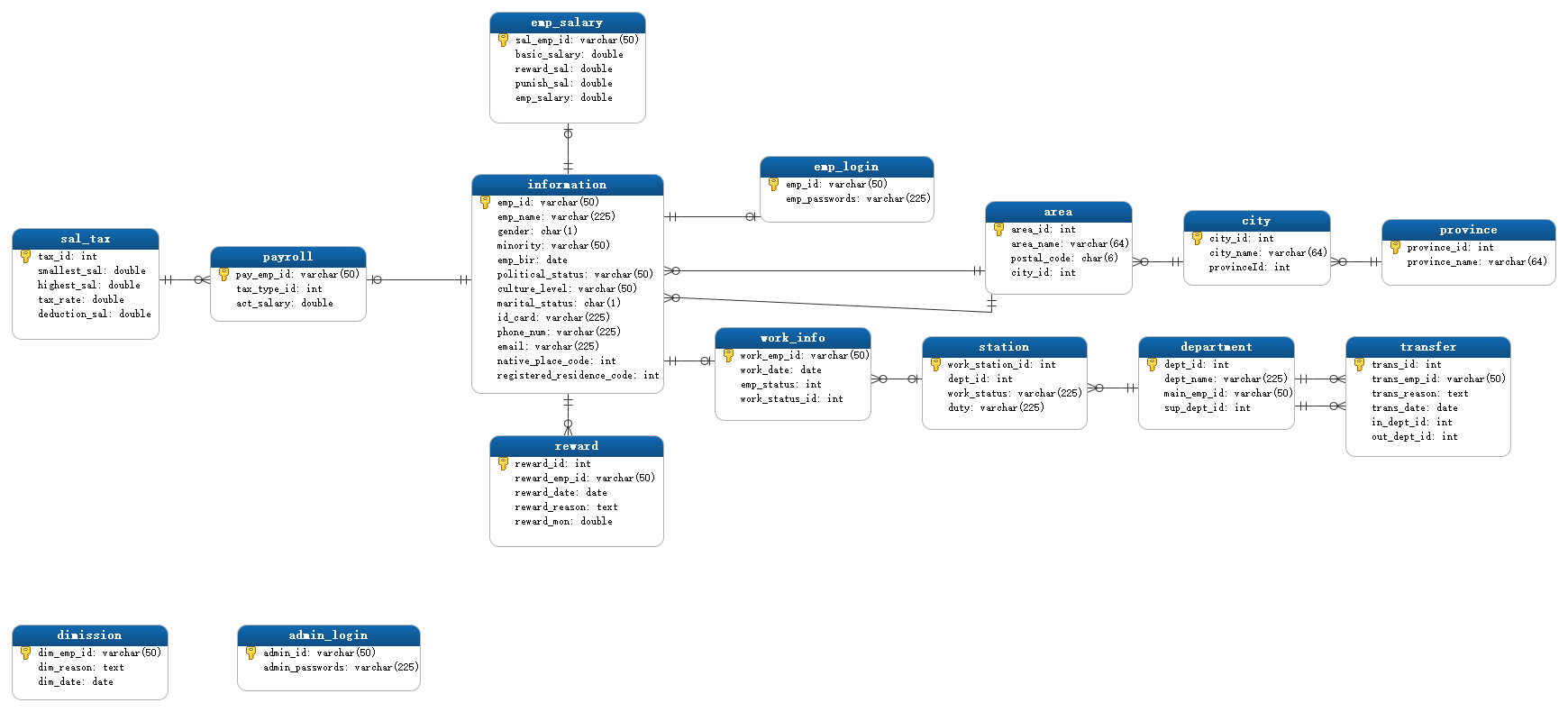


图13 主、外键约束

# 6.数据库物理设计

## 6.1 索引

数据库中索引如表3所示：

表3 索引列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ObjectName** | **IndexName** | **ID** | **AverageFreeBytes** | **LogicalFragementation** |
| admin\_login | admin\_id | 1 | 7964.0 | 0.0 |
| area | area\_id | 2 | 7690.0 | 0.0 |
| city | city\_id | 2 | 7830.0 | 0.0 |
| department | dept\_id | 3 | 7935.0 | 0.0 |
| dimission | dim\_emp\_id | 1 | 7706.0 | 100.0 |
| emp\_login | emp\_id | 1 | 7970.0 | 0.0 |
| emp\_salary | sal\_emp\_id | 1 | 4712.0 | 100.0 |
| information | **emp\_id** | 3 | 7719.0 | 100.0 |
| payroll | pay\_emp\_id | 2 | 7844.0 | 100.0 |
| province | province\_id | 1 | 8021.0 | 100.0 |
| reward | reward\_id | 2 | 7823.0 | 0.0 |
| sal\_tax | tax\_id | 1 | 7459.0 | 0.0 |
| station | work\_station\_id | 2 | 7732.0 | 0.0 |
| transfer | trans\_id | 3 | 4903.0 | 100.0 |
| work\_info | work\_emp\_id | 2 | 8001.0 | 0.0 |

在全部索引中除每张表的主键索引外还包括另外两个单独建立的索引：

（1）information 中的emp\_id索引：员工的工作证号对应员工每一个人，在查询员工基本信息，相关工作信息以及人员调动等方面都占据着很大的作用，所以建立了此索引；

（2）admin\_login 中的admin\_id索引：管理员起着管理员工信息的作用，虽然每一个员工都有自己的工作证号，但大多数情况下只是修改个人信息，对于人员调动，离职等相关信息都需要管理员来对相关信息进行管理操作，所以建立了此索引。

## 6.2 视图

本数据库中主要建立了以下、视图：

用于企业管理的员工管理数据库，包含员工信息表information、部门信息表department、员工薪水情况表emp\_salary。

1.对员工管理数据库创建视图Emp\_view1,包含所有男员工工作证号、姓名、工作年限和学历，字段名用中文表示。

create view Emp\_view1 as select emp\_id, emp\_name,gender,emp\_bir,phone\_num from information where 性别='男' with check option;

1. 创建创建视图Emp\_view 2,包含工作证号、姓名、所在部门名称和收入

Create or replace view Emp\_view2 as select information.emp\_id,emp\_name,department.dept\_name,payroll.art\_salary

from information,department,payroll

where information.emp\_id=payroll.emp\_id and information.dept\_id=department.dept\_id;

建立以上视图主要是因为：

（1）基于以上视图实现复杂查询，以避免书写频繁执行连接操作；

（2）以上视图所包含的表的数据关系到许多用户，视图可以隐藏基本表；可以禁止所有用户访问数据库表，而要求员工等用户只能通过视图操作数据；

（3）在一定程度上保护高级语言应用程序不受某些数据库结构修改的影响。

## 6.3 安全机制

### 6.3.1系统安全

为系统管理员和普通用户用户在数据库中存储登录口令，登录时必须经验证一致才可进入系统；

由于管理员用户有频繁的查询更新修改等操作且该类用户数量有限，所以为每一个管理员用户在数据库中建立单独的登录名和用户名，同时，管理员拥有普通用户的权限以及自己的专有权限，在管理员登录时，其密码由SQLSever实现验证，由于SQLSever的登录密码是经加密保存的，所以具有较高的安全性。

### 6.3.2数据安全

在该系统中，任何有关删除或添加数据的操作都需要二次确认才可以在数据库中真正执行。通过高级语言应用程序的限制，不同的用户只能查询修改其有权访问的数据。同时，提供员工基本信息的备份功能，防止数据丢失。

# 7.应用程序设计

## 7.1 功能模块

### 7.1.1计算功能

（1）计算员工当月应纳税额。

（2）统计指定部门的员工数。

### 7.1.2存储功能

存储员工的具体信息及员工与部门、部门与职务岗位的关系，并能方便快速地实现对以上信息的删除增加等操作。

### 7.1.3输入及输出功能

输入的限定信息等数据经处理后，可以在显示器上按事先定义的格式显示，特别是查看员工工作信息时可以跟踪员工当前的工作状态(1-在职，2-兼职，3-试用，4-离职，5-返聘，6-退休)。另外还能通过指定设备存储员工薪资等报表，以便打印或长期保存。

### 7.1.4交互功能

本系统具有交互界面，会在用户进行输入等操作时给予必要的提示，或者在操作发生错误时给予必要的提示和帮助。在查找员工的籍贯和户口所在地时可以通过邮政编码自动确认。

## 7.2 界面设计

### 7.2.1管理员登录界面



图15 管理员登录界面

### 7.2.2管理员用户界面



图16 管理员用户界面

(1)基本信息管理页面

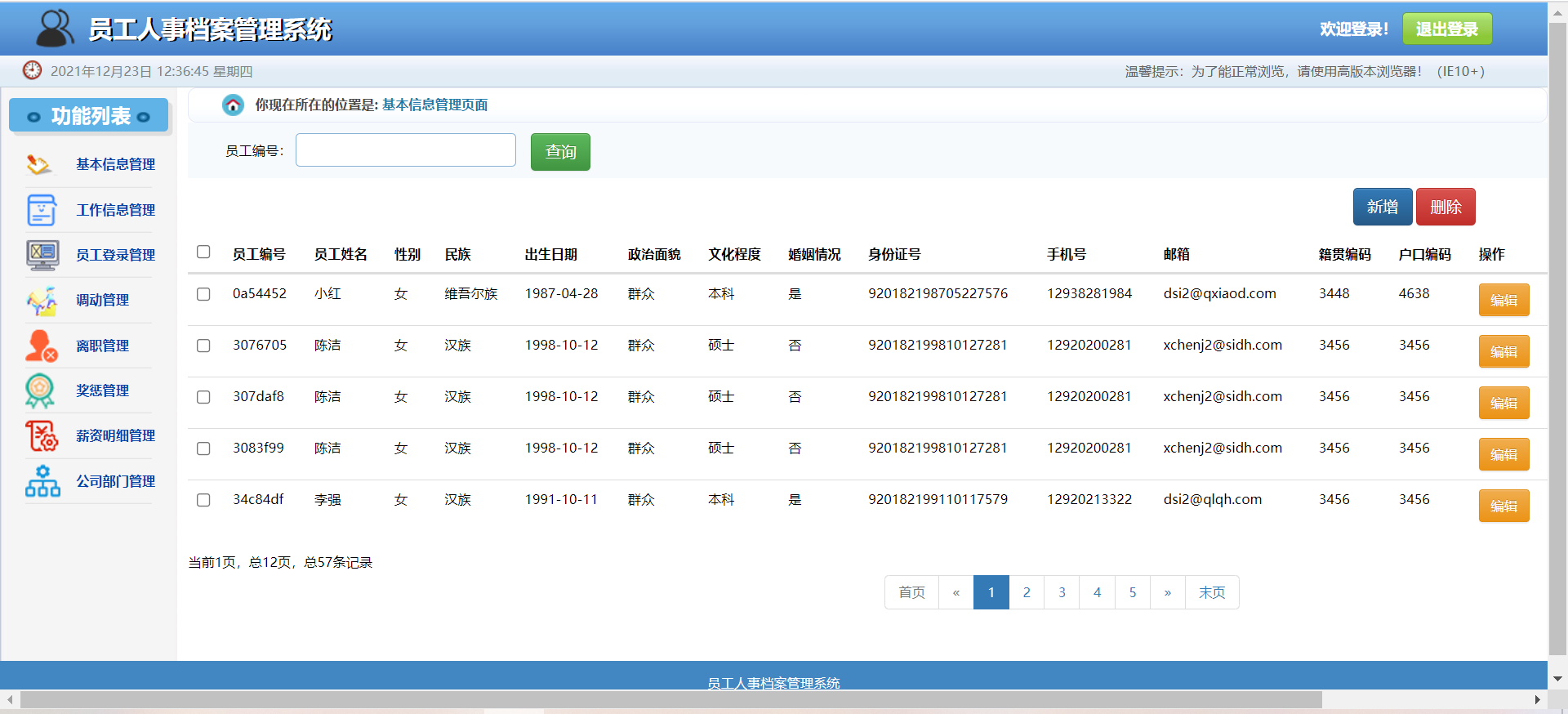


图17 本信息管理页面



图18查询员工

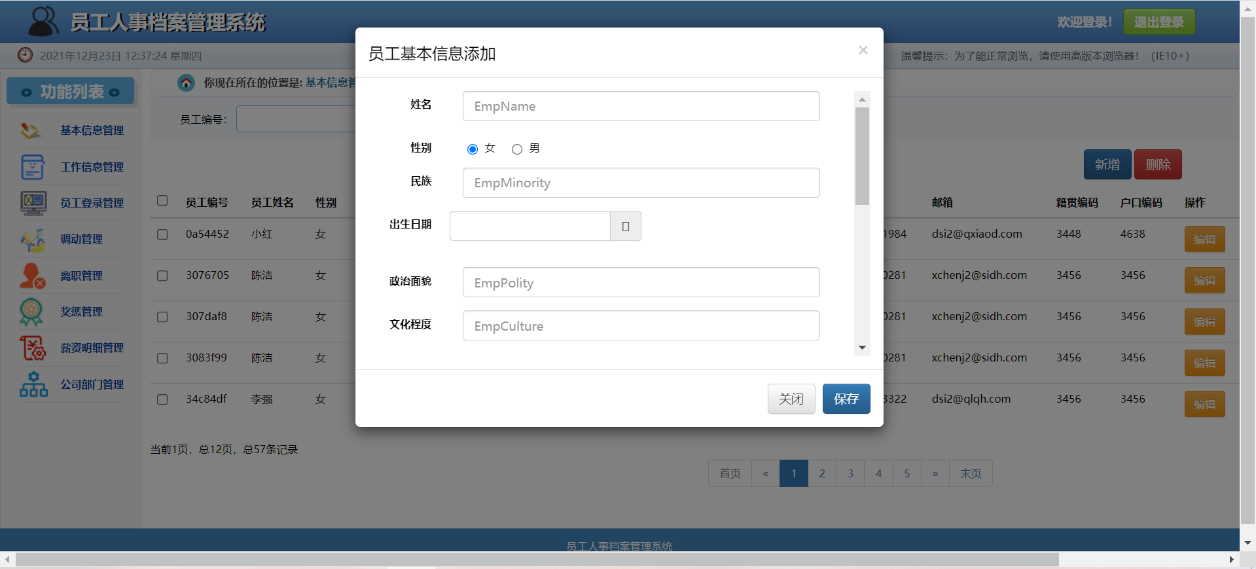


图19 添加员工基本信息

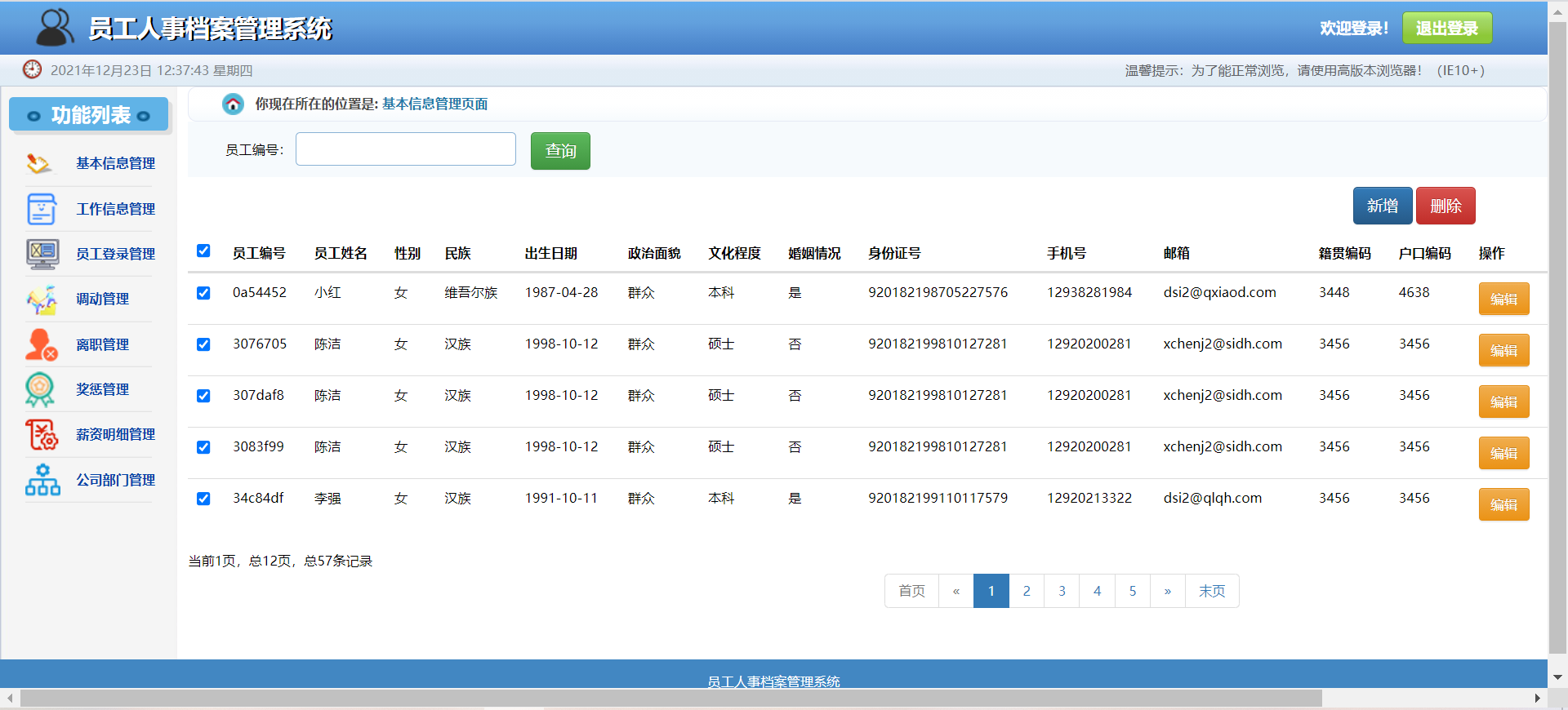


图20 员工基本信息添加成功

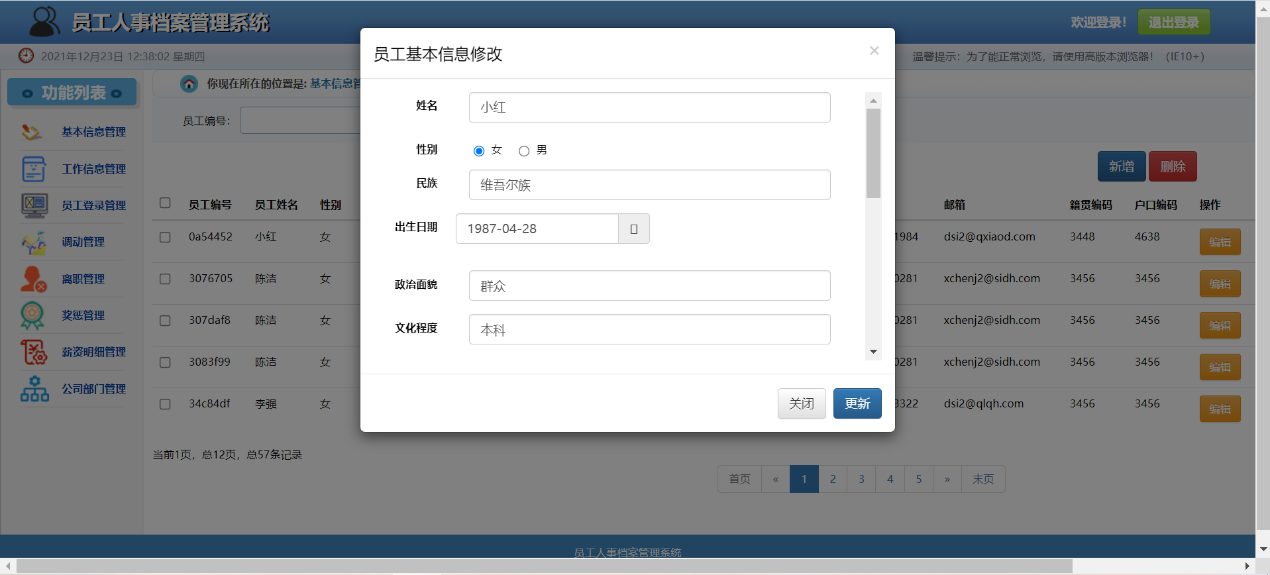


图21 修改员工基本信息



图22 员工基本信息修改成功

（3）管理员工工作信息



图23 查询员工工作信息



图24 新增员工工作信息

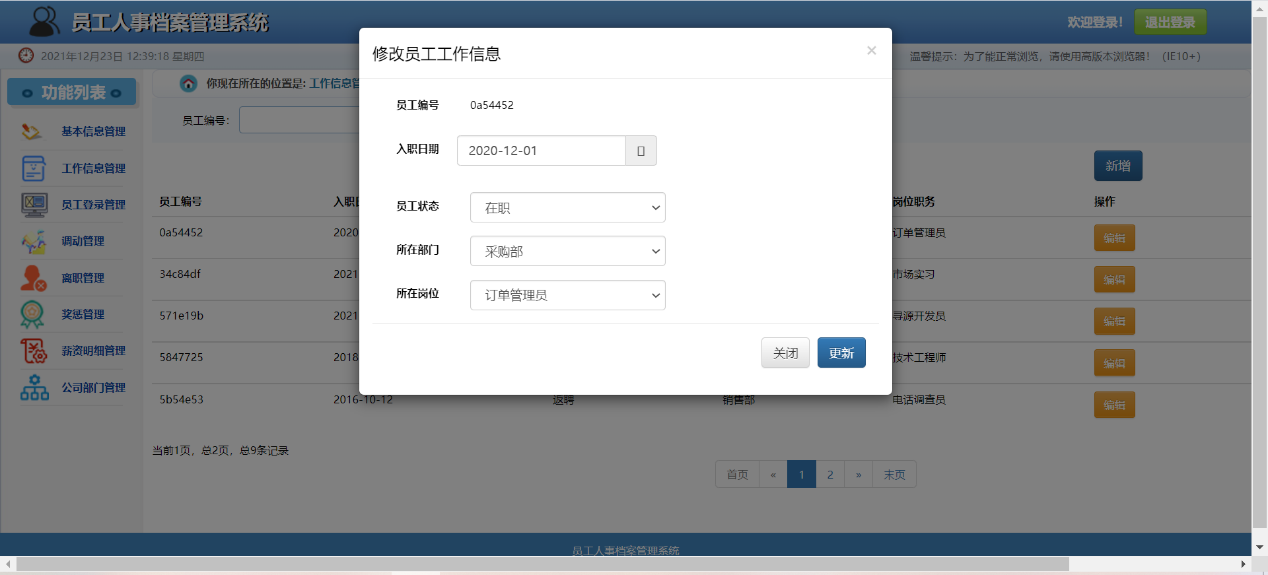


图25 修改员工工作信息

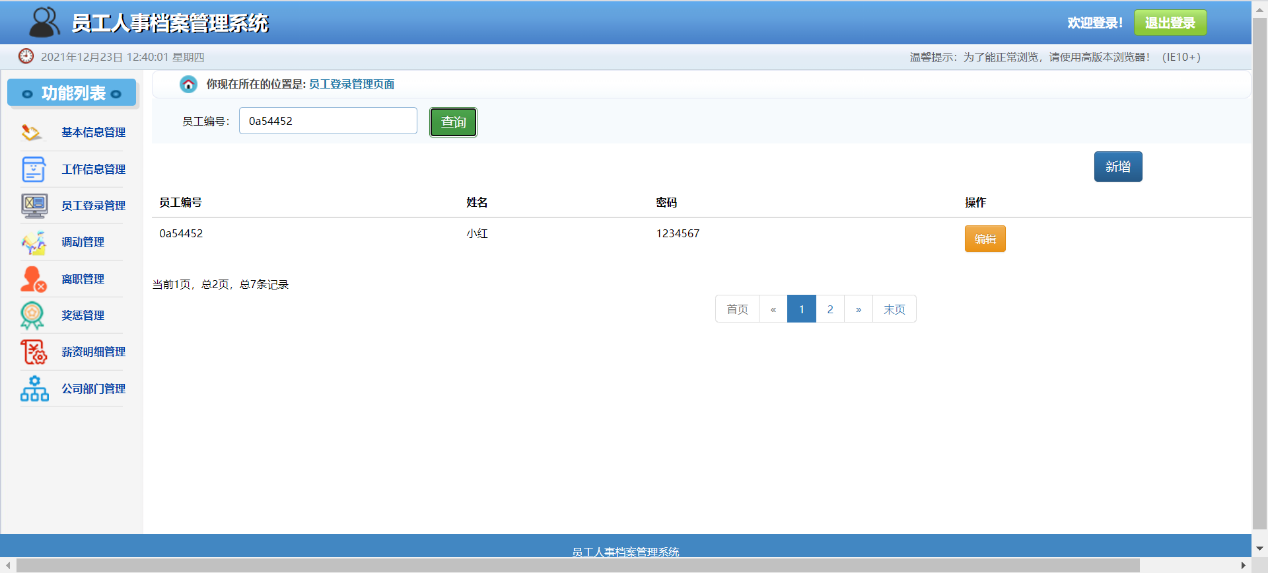


图26 员工基本信息修改成功

（4）管理员工登录信息



图27 新增员工登录信息



图28 修改员工登录信息

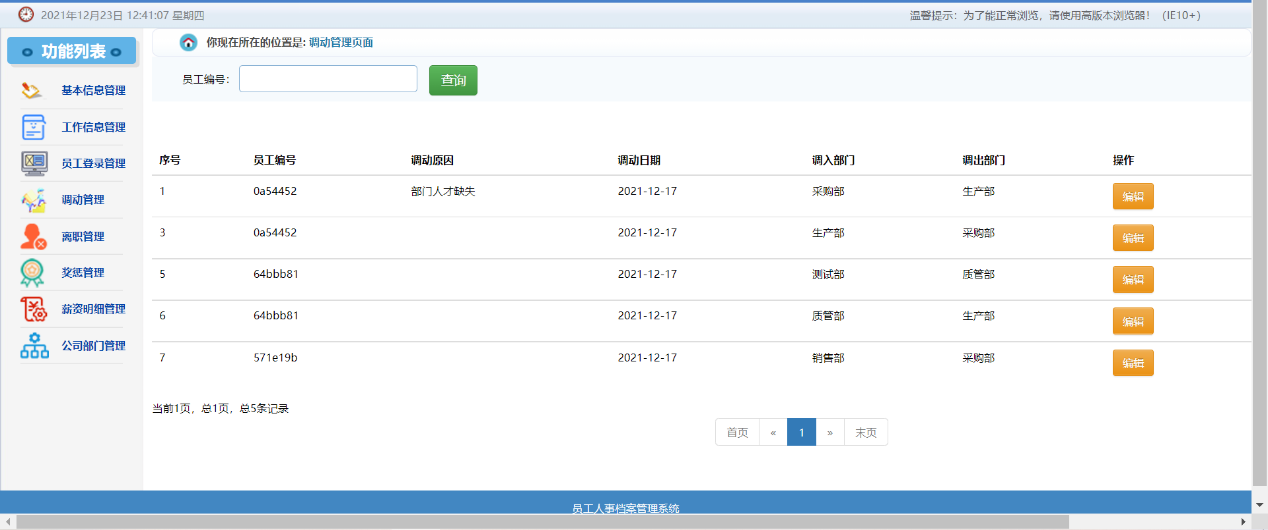
（5）调动管理界面

图29 调动管理界面



图30 查询调动信息



图31 修改调动信息

（6）离职管理界面

图32 离职管理界面

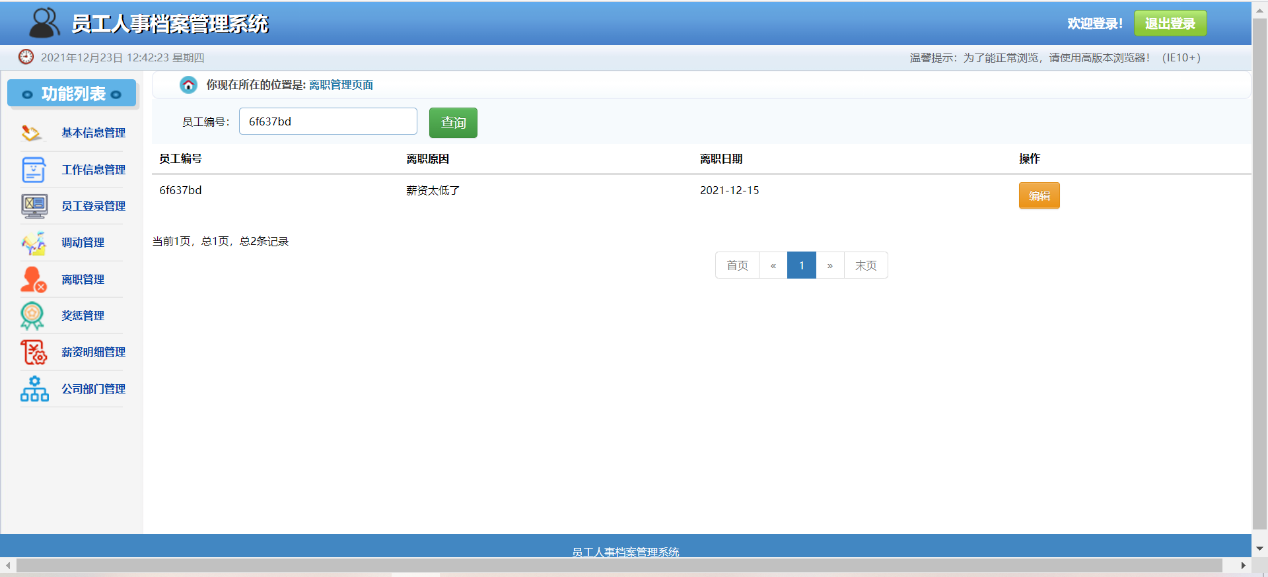


图33 查询调动信息



图34 修改调动信息

（7）奖惩管理界面

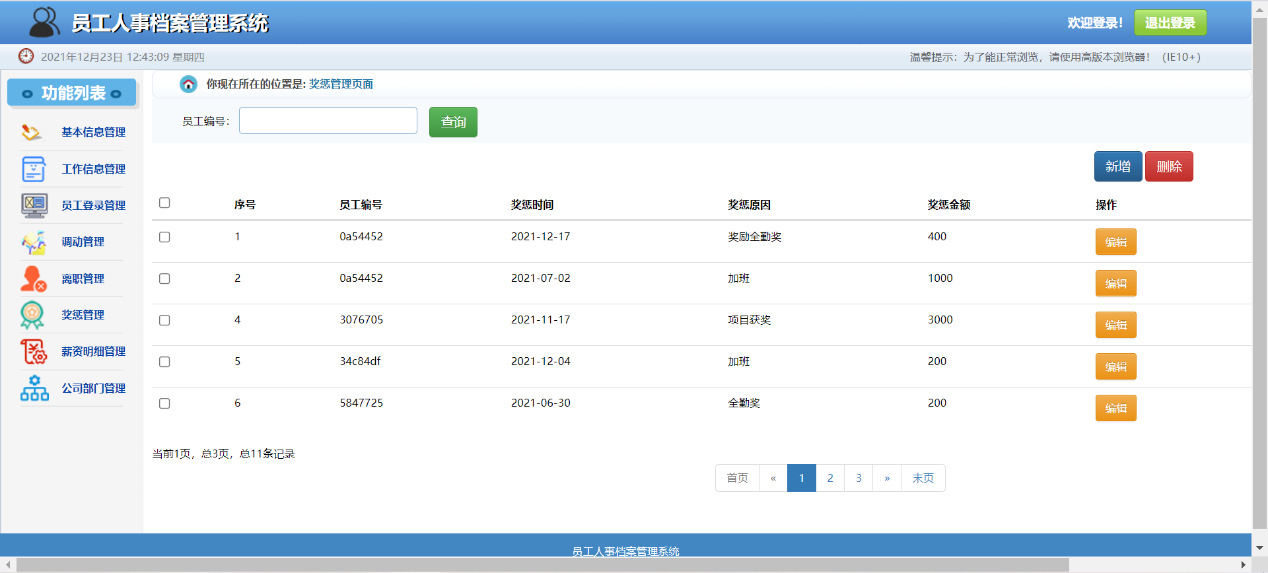


图35奖惩管理界面

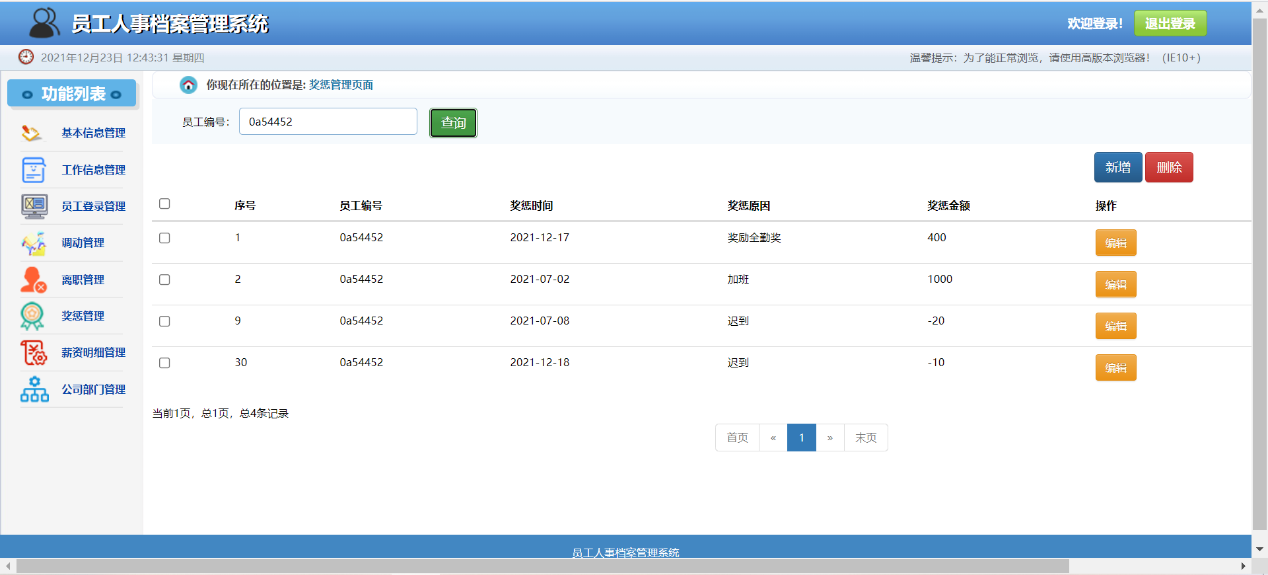


图36 查询奖惩信息



图37 新增奖惩信息



图38 修改奖惩信息

（8）薪资明细管理界面

图39薪资明细管理界面



图40 查询薪资明细



图41 修改员工总工资信息

（8）公司部门管理界面



图42 公司部门管理界面

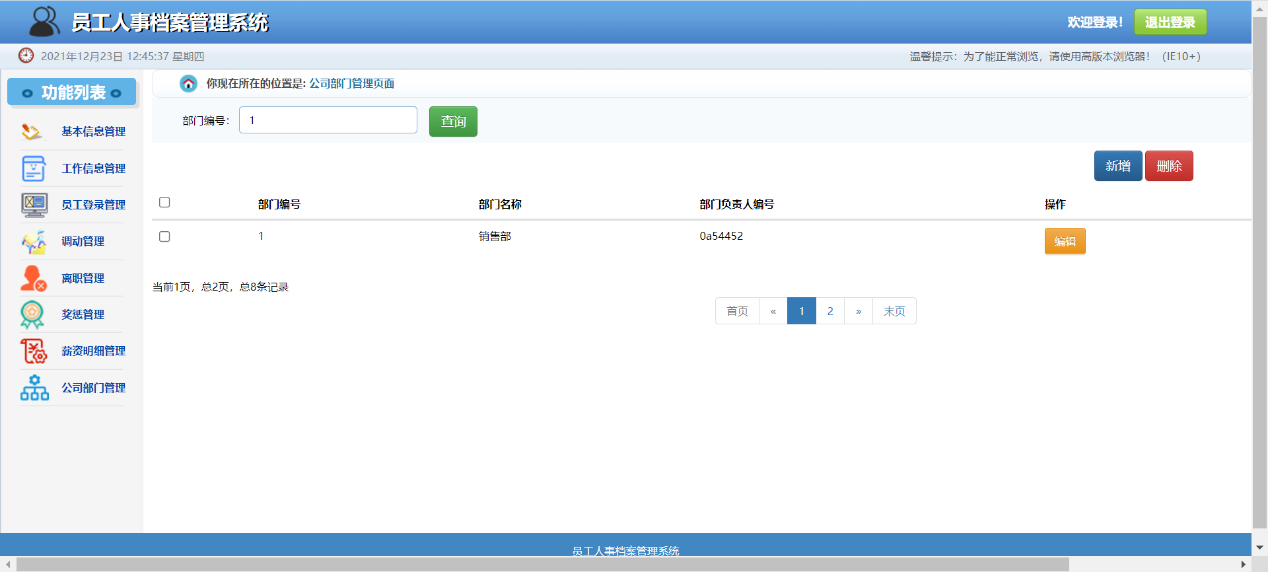


图43查询部门信息

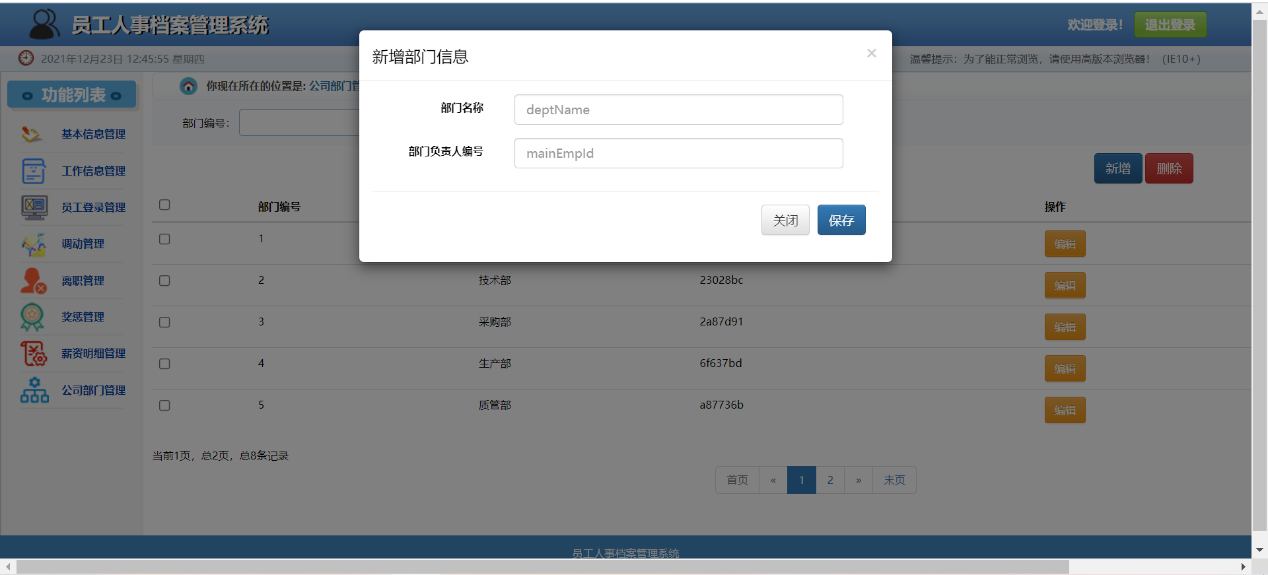


图44 新增部门信息

图45 修改部门信息

### 7.2.3员工登录界面

### 

图46 员工登录界面

### 7.2.4员工用户界面

图47 员工用户界面

(1)员工基本信息界面



图48 员工基本信息



图49 修改员工基本信息

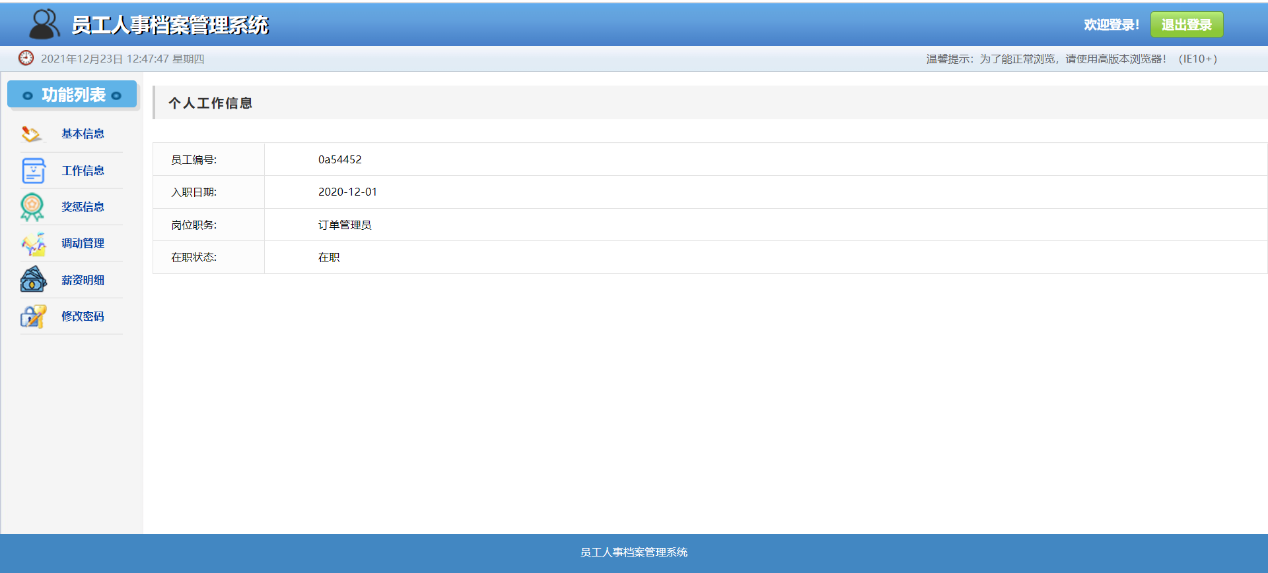


图50 员工信息修改成功

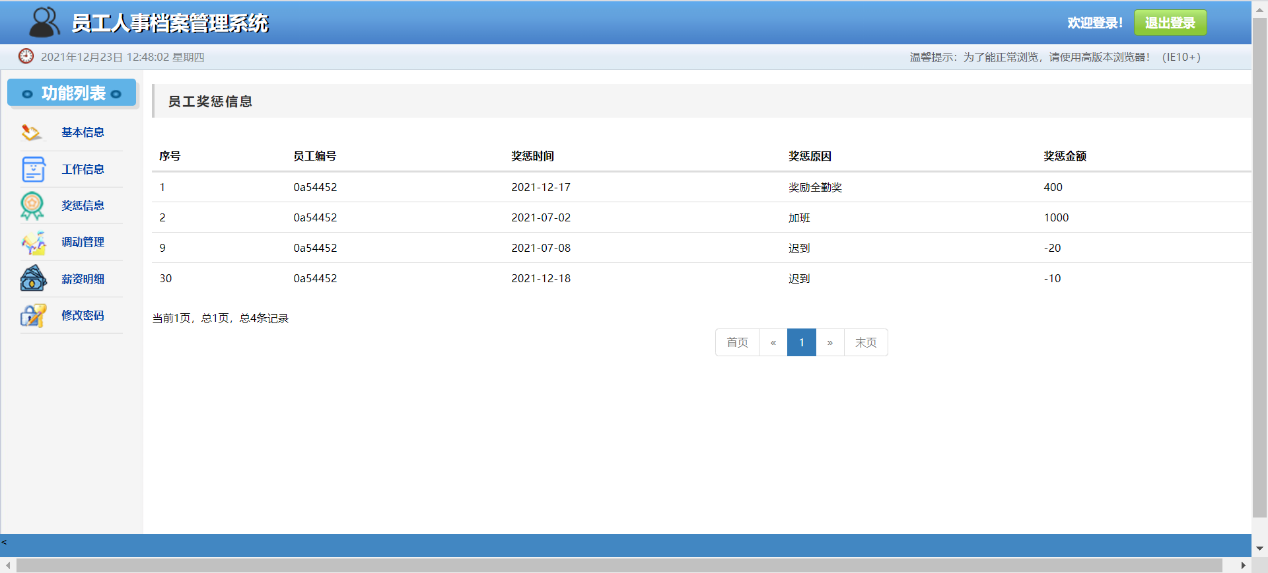
(2)员工奖惩情况界面 

图51 员工奖惩情况界面

(3)员工岗位调动信息界面

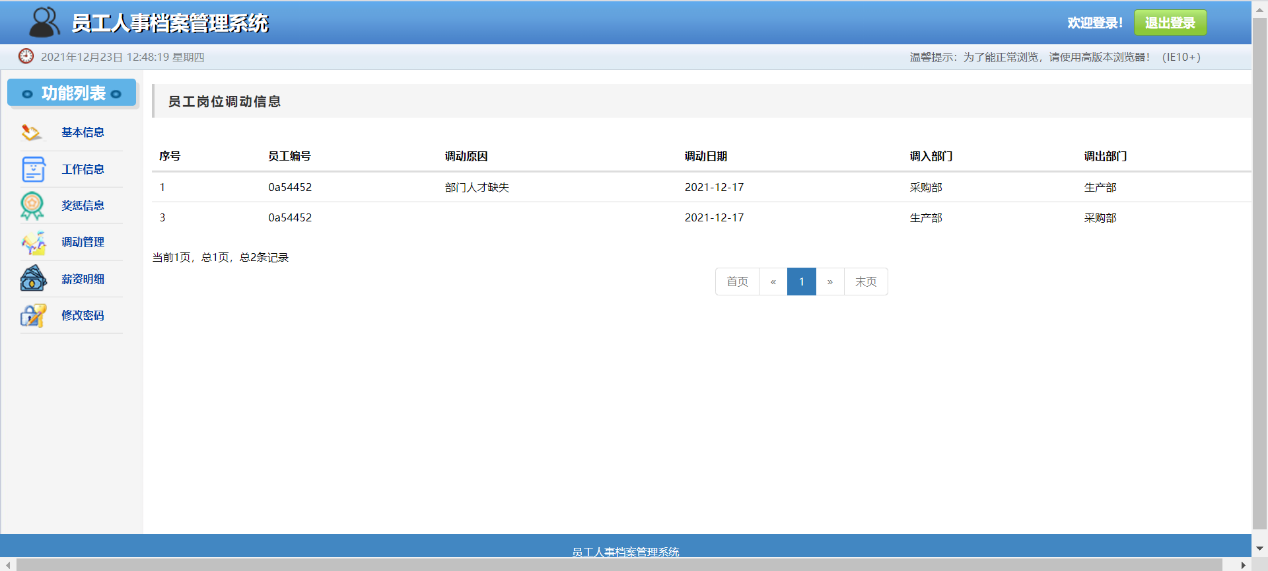


图52 员工岗位调动信息界面

(4)员工薪资明细界面

图53员工薪资明细界面

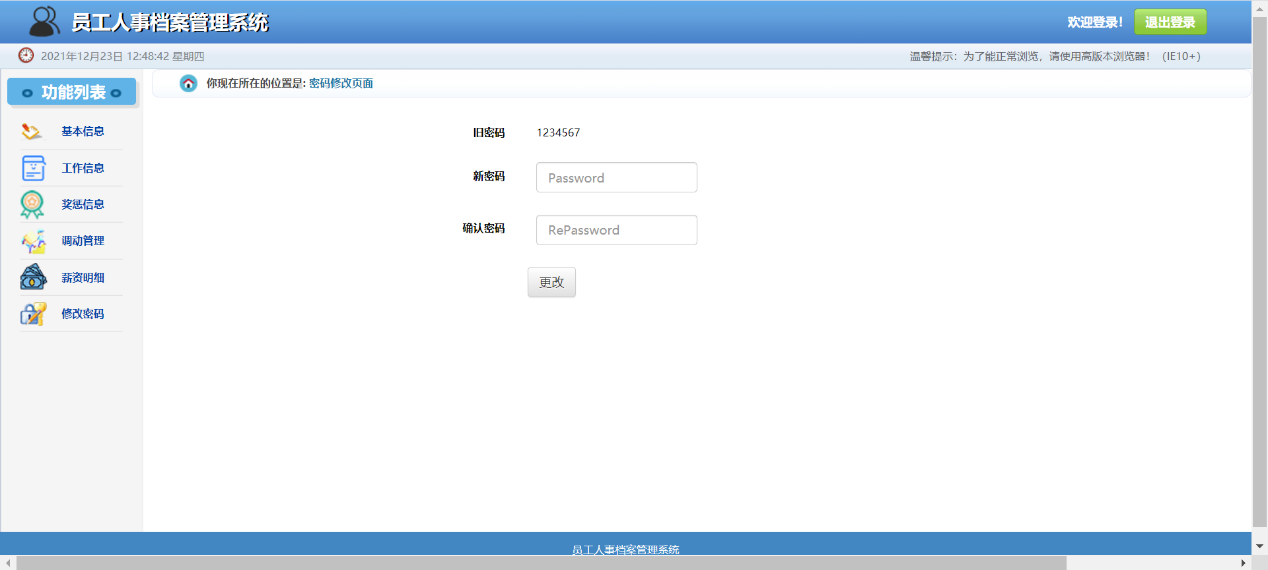
(5)密码修改界面

图54 密码修改界面

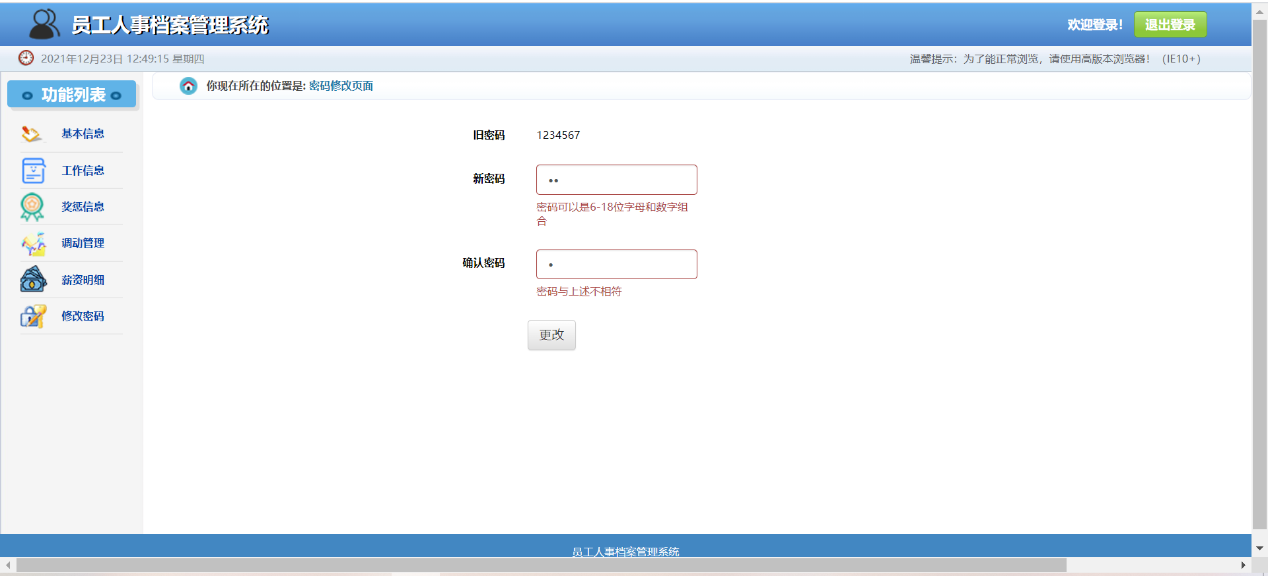


图55 密码校验

## 7.3 重要代码

### 7.3.1触发器

ID号对于用户是很重要的属性，是唯一的标识。通过调用UUID，随机生成ID号，Replace ID号里的“-”，并取前六位。以下是ID生成的触发器：

管理员：SET new.admin\_id=substring(REPLACE(UUID(),'-',''),1,7);

员工：Set new.emp\_id=substring(replace(uuid(),'-',''),1,7);

**reward表**

**insert\_salary触发器** （reward表插入新信息，若emp\_salary中已有该信息，更新emp\_salary表相应的字段；否则新增该信息）

SET if exists(select \* from emp\_salary where sal\_emp\_id=new.reward\_emp\_id) then

if (new.reward\_mon>0) then

update emp\_salary set reward\_sal=reward\_sal+new.reward\_mon,

emp\_salary=emp\_salary+new.reward\_mon where sal\_emp\_id=new.reward\_emp\_id;

else

UPDATE emp\_salary SET punish\_sal=punish\_sal+new.reward\_mon,

emp\_salary=emp\_salary+new.reward\_mon WHERE sal\_emp\_id=new.reward\_emp\_id;

end if;

else

IF (new.reward\_mon>0) THEN

insert into emp\_salary (sal\_emp\_id,reward\_sal,emp\_salary) values(new.reward\_emp\_id,reward\_sal+new.reward\_mon,emp\_salary+new.reward\_mon);

ELSE

INSERT INTO emp\_salary (sal\_emp\_id,punish\_sal,emp\_salary) VALUES(new.reward\_emp\_id,punish\_sal+new.reward\_mon,emp\_salary+new.reward\_mon);

END IF;

end if;

**update\_salary触发器**（reward表中的奖惩金额更新，更新emp\_salary表）

IF (new.reward\_mon>0) THEN

UPDATE emp\_salary SET reward\_sal=reward\_sal-old.reward\_mon+new.reward\_mon,

emp\_salary=emp\_salary-old.reward\_mon+new.reward\_mon WHERE sal\_emp\_id=new.reward\_emp\_id;

ELSE

UPDATE emp\_salary SET punish\_sal=punish\_sal-old.reward\_mon+new.reward\_mon,

emp\_salary=emp\_salary-old.reward\_mon+new.reward\_mon WHERE sal\_emp\_id=new.reward\_emp\_id;

END IF;

**delete\_salary触发器**（删去reward表中的信息，更新emp\_salary表）

IF (old.reward\_mon>0) THEN

UPDATE emp\_salary SET reward\_sal=reward\_sal-old.reward\_mon,

emp\_salary=emp\_salary-old.reward\_mon WHERE sal\_emp\_id=old.reward\_emp\_id;

ELSE

UPDATE emp\_salary SET punish\_sal=punish\_sal-old.reward\_mon,

emp\_salary=emp\_salary-old.reward\_mon WHERE sal\_emp\_id=old.reward\_emp\_id;

END IF;

**work\_info表**

**insert\_dimission触发器**（当work\_info表中员工状态为离职，在dimission表内添加员工离职信息）

if (new.emp\_status=4) then

insert into dimission (dim\_emp\_id,dim\_date) values(new.work\_emp\_id,now());

end if;

**update\_dimission触发器**（work\_info表中员工状态更改不为离职时，从dimission表中删除该信息）

IF (new.emp\_status=4) THEN

INSERT INTO dimission (dim\_emp\_id,dim\_date) VALUES(new.work\_emp\_id,NOW());

else

DELETE FROM dimission WHERE dim\_emp\_id=old.work\_emp\_id;

END IF;

**update\_transfer触发器**（work\_info表中岗位所在部门发生变化时，在transfer表中新增调动信息）

if (old.work\_status\_id != new.work\_status\_id) then

insert into transfer (trans\_emp\_id,trans\_date,in\_dept\_id,out\_dept\_id) values(old.work\_emp\_id,

now(),(select dept\_id from station where work\_station\_id=old.work\_status\_id),

(SELECT dept\_id FROM station WHERE work\_station\_id=new.work\_status\_id));

end if;

**emp\_salary表**

**update\_final\_salary触发器**（当emp\_salary表中的基本金额更新时，总薪资发生改变）

if (old.basic\_salary is null) then

set new.emp\_salary=old.emp\_salary+new.basic\_salary;

else

SET new.emp\_salary=old.emp\_salary-old.basic\_salary+new.basic\_salary;

end if;

### 7.3.2 多表联接

将员工基本信息表与地区表、省份表、市级表相连接，籍贯所在地编号和户口所在地编号为地区表的外键，省份表编号为市级表的外键，市级表编号为地区表的外键。

SELECT emp\_id, emp\_name, gender, minority, emp\_bir, political\_status, culture\_level,

marital\_status,

id\_card, phone\_num, email, native\_place\_code,

area\_id,area\_name,postal\_code,area.city\_id,

city.city\_id,city\_name,provinceId,

province\_id,province\_name

FROM information

LEFT JOIN `area` ON native\_place\_code=area\_id

LEFT JOIN city ON city.city\_id=area.city\_id

LEFT JOIN province ON province\_id=provinceId;

将职务岗位表与部门表相连接，部门编号为部门表的外键。

SELECT work\_station\_id, station.dept\_id, work\_status, duty,department.dept\_id,main\_emp\_id,sup\_dept\_id

FROM station

LEFT JOIN department ON department.dept\_id = station.dept\_id;

将员工登录信息表与员工基本信息表通过工作证号连接

SELECT emp\_login.emp\_id, emp\_passwords,information.emp\_id,emp\_name

FROM `emp\_login`

LEFT JOIN information ON `emp\_login`.emp\_id = information.emp\_id;

将调动信息表与部门表相连接，调入部门编号、调出部门编号为部门表的外键。

SELECT trans\_id, trans\_emp\_id, trans\_date, in\_dept\_id, out\_dept\_id,d\_in.dept\_id,d\_in.dept\_name,d\_in.main\_emp\_id

,d\_out.dept\_id,d\_out.dept\_name,d\_out.main\_emp\_id

FROM transfer

LEFT JOIN department d\_in ON in\_dept\_id = d\_in.dept\_id

LEFT JOIN department d\_out ON out\_dept\_id = d\_out.dept\_id;

# 8.测试和运行

根据软件生命周期，员工人事档案管理系统测试被分为了单元测试，集成测试和系统测试三个阶段：

1. 单元测试：该阶段主要是对程序所需SQL语句进行测试。在编码阶段完成后就着手进行的单元测试主要是用来保证当前代码的正确性,更重要的是用来保证代码修复、改进和重构之后的正确性。
2. 集成测试：在单元测试的基础上，按照设计要求，将本系统的各个部件进行联合测试，以决定它们能否在一起共同工作。本阶段的测试工作包括：测试不同的用户身份是否可以正确获取对应权限；各种报表的存储位置和格式是否可以按要求实现，薪资金额是否准确以及是否对应等内容。
3. 系统测试：该阶段邀请了小组外的几名同学，分别以员工和管理员的身份使用系统。测试了查询信息，修改信息、删除信息等操作，测试尽可能多的情况。并根据他们的使用感受增添一些使用提示。

大量数据测试是对录入员工的基本信息进行，188位员工的基本信息可以在3秒内录入，系统的性能基本上满足需求。

# 9.总结

这次课程设计我们做的是员工人事档案管理系统， 主要是服务于员工与系统管理员，实现管理员对员工人事档案进行信息化管理，降低人事档案管理的成本，提高工作效率。系统中采用的技术主要有：Mybatis数据库框架技术和MVC三层架构以及SSM、SpringMvc、SpingBoot、BootStrap框架，同时利用Idea集成开发环境。

在整个数据库课程设计中，小组队员全员参与，分工明确，各司其职。在共同协商下，高效地确定出整体设计框架。但过程中仍有不足之处，小组成员不断修改、完善数据库表，争取做到自己最满意的程度。事实证明，只靠一人是不能顺利完成一套功能完整的系统的，必须充分利用团队的力量。

开发一套系统，最重要的是细心，并不是一定要做到面面俱到，但也要充分考虑到客户的需求和现实意义，不管什么系统，只用运用到实际应用中，才具有先现实意义。所以在准备工作中要正确分析社会需求了解现实应用，画出流程图，把大体框架做好，然后再逐一细化。我们不可能做到面面俱到，但一定要做到步步扎实，我们作为程序编程人员，要保持清醒的头脑，以现实为依据，让自己的每一行代码都能实现自己的意义。

通过这次课程设计，收获的不仅仅是课程上的知识得到实际应用，还有编程的基本习惯和开发系统时应注意的流程。

# 参考文献

1. 宋金玉，陈萍，陈刚.数据库原理与应用.2版.北京：清华大学出版社，2014
2. 路游，于与宗.数据库系统课程设计.北京：清华大学出版社，2009
3. ForFuture Code. <https://blog.csdn.net/ganquanzhong/article/details/80610526.CSDN,2019>
4. GitHub. <https://github.com/netbuffer/ssmbootstrap_table.GitHub,2021>
5. 梵空. https://download.csdn.net/download/qq\_34988391/10713377.CSDN,2018