成都东软学院

综合实训报告

|  |  |
| --- | --- |
| **实训题目**： | 企业人事管理系统的设计与实现 |
| **指导教师：** | 范畅 |
| **学 院** | 计算机与软件学院 |
| **系 别：** | 软件工程系 |
| **专业班级：** | 软件工程21303 |
| **学 号：** | 21010230338 |
| **学生姓名：** | 陈浩 |
| **实 训 期：** | 2022-2023学年第1学期 |

**2022年11月**

# 第1章 项目概述

## 1.1 项目背景

随着计算机技术的迅猛发展，通过计算机技术实现企业人事管理是大势所趋。相对于企业而言，运用计算机支持企业高效快速地完成企业人事管理的日常事务管理，是加快企业人事管理科学化、高效化、规范化、自动化的基本前提，也是提高企业综合实力和市场竞争力的有效途径。

目前，在国内外各大型企业事业单位中，人事管理系统使用是非常广泛的。同时，许多开发人员开发了许多基于不同行业的企业人事管理系统。有一个单机输入版本，也有C/S和B/S模式的，而不同的企业有不同的人事管理制度，这就决定了不同的企业需要不同的系统。而且企业人数越多就会产生越大的信息量，传统的方法就越被人诟病；开发一个适合大多数中小企业的人事管理系统，使其具有优良的性能并能快速投入各企业使用，是加快企业人才管理工作自动化、信息资源化建设进程必要途径。开发适合大多中小企业的管理信息系统，是促进企业管理信息化高速发展，为企业的发展提供便利的课题。

## 1.2 项目目的和意义

放眼当下，现代化信息技术的发展，已经让我们的生活和工作方式发生了翻天覆地的变化，它是在传统意义形式上和人事管理的内涵上的根本改变。在过去，一支笔、记录手册和办公桌就是所有办公所需用品了。如今电脑，扫描仪和打印机已经基本取代旧的办公用品。今天，人事管理是完全启用计算机和计算机辅助快速，准确地完成许多复杂的任务。这些都大大缩短了管理时间，人员管理的内容已被改造成一个更广泛的系统性值。我认为，在现代化的进程中，引入现代化管理的思想，建立一套可面向企业单位人事的信息管理系统，也是十分必要的。

随着网络技术和数据库功能的技术提升，基于JavaWeb等各种框架的管理信息系统正式成为主流。同时，Internet的快速发展与普及，国内许多大企业都怕已经有属于自己的企业人事管理系统，为企业的快速运营提供便捷。但对于中小型企业来说，在不需要大量数据库服务器的前提下，开发出一个功能强大，操作简单的人事管理系统。

# 第2章 可行性分析

## 2.1 经济可行性

从该系统的开发和为公司带来的收益上来讲，该系统的开发花费不了太多的费用。公司需要有自己的服务器和局域网用来部署系统，开发方面的工具：JAVA可以用idea、数据库可以用Mysql，后期系统投入使用后又可以节省人力、财力的消耗，能有效地提高员工信息更新效率，节约资源，所以为长远考虑，此系统在经济方面会节约出不少费用。

## 2.2 技术可行性

本系统的开发使用了JAVA语言，其具备多个优秀的特性，如：移植性强，面向对象等特性，在开发软件方面：具有开发高效、安全可靠的特点，数据库使用平时较为常见的关系型数据库Mysql，这些都是学习过程中经常使用的，所以本系统具有技术可行性。

## 2.3 操作可行性

该系统使用简单方便，具有清晰友好的操作界面，对系统的使用人员要求并不高，实际上公司每个人都可以快速上手使用，需要使用的时候只需要花费少量的时间按照页面文字功能描述点击即可操作，所以系统在操作上是完全可行的。

# 第3章 开发技术与环境

## 3.1 系统技术介绍

系统为前后端分离开发，后端采用SSM框架进行开发，并使用Shiro作为系统的安全框架，前端采用jsp+html+JavaScript+jQuery为前端页面开发技术，并使用Layui作为前端UI框架，SSM（Spring+SpringMVC+MyBatis）框架集由Spring、MyBatis两个开源框架整合而成（SpringMVC是Spring中的部分内容），常作为数据源较简单的web项目的框架。

3.1.1 Spring

Spring是一个开源框架，Spring是于2003年兴起的一个轻量级的Java开发框架，由Rod Johnson在其著作Expert One-On-One J2EE Development and Design中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。 简单来说，Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。Spring的优点：（1）通过Spring的IOC特性，将对象之间的依赖关系交给了Spring控制，方便解耦，简化了开发。（2）通过Spring的AOP特性，很容易实现事务，日志，权限的控制。（3）提供了对其他优秀开源框架的集成支持。（4）低侵入式。

3.1.2 SpringMVC

Spring MVC属于Spring Framework的后续产品，已经融合在Spring Web Flow里面，它原生支持的Spring特性，让开发变得非常简单规范。Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。SpringMVC的优点：（1）springMVC是使用了MVC设计思想的轻量级web框架，对web层进行解耦，是的我们开发更简洁。（2）与Spring无缝衔接。（3）灵活的数据验证，格式化，数据绑定机制。

3.1.3 MyBatis

MyBatis本是apache的一个开源项目iBatis, 2010年这个项目由apache software foundation 迁移到了google code，并且改名为MyBatis 。MyBatis是一个基于Java的持久层框架。iBATIS提供的持久层框架包括SQL Maps和Data Access Objects（DAO）MyBatis消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis使用简单的XML或注解用于配置和原始映射，将接口和Java的POJO（Plain Old Java Objects，普通的 Java对象）映射成数据库中的记录。可以这么理解，MyBatis是一个用来帮你管理数据增删改查的框架。优点：（1）sql写在xml文件中，便于统一管理和优化，解除sql和程序代码的耦合。（2）提供映射标签，支持对象和和数据库orm字段关系的映射，支持对象关系映射标签，支持对象关系的组建。（3）提供xml标签，支持编写动态sql。

3.1.4 Shiro

Apache Shiro是一个强大且易用的Java安全框架，执行身份验证、授权、密码和会话管理。使用Shiro的易于理解的API，可以快速、轻松地获得任何应用程序，从最小的移动应用程序到最大的网络和企业应用程序。其包括三个核心组件：Subject，SecurityManager 和 Realms

1、Subject：

即“当前操作用户”。但是，在Shiro中，Subject这一概念并不仅仅指人，也可以是第三方进程、后台帐户（Daemon Account）或其他类似事物。它仅仅意味着“当前跟软件交互的东西”。Subject代表了当前用户的安全操作，SecurityManager则管理所有用户的安全操作。

2、SecurityManager：

它是Shiro框架的核心，典型的Facade模式，Shiro通过SecurityManager来管理内部组件实例，并通过它来提供安全管理的各种服务。

3、Realm：

Realm充当了Shiro与应用安全数据间的“桥梁”或者“连接器”。也就是说，当对用户执行认证（登录）和授权（访问控制）验证时，Shiro会从应用配置的Realm中查找用户及其权限信息。从这个意义上讲，Realm实质上是一个安全相关的DAO：它封装了数据源的连接细节，并在需要时将相关数据提供给Shiro。当配置Shiro时，你必须至少指定一个Realm，用于认证和（或）授权。配置多个Realm是可以的，但是至少需要一个。

Shiro内置了可以连接大量安全数据源（又名目录）的Realm，如LDAP、关系数据库（JDBC）、类似INI的文本配置资源以及属性文件等。如果缺省的Realm不能满足需求，还可以插入代表自定义数据源的自己的Realm实现。

3.1.5 JavaScript

JavaScript（简称“JS”） 是一种具有函数优先的轻量级，解释型或即时编译型的编程语言。虽然它是作为开发Web页面的脚本语言而出名的，但是它也被用到了很多非浏览器环境中，JavaScript 基于原型编程、多范式的动态脚本语言，并且支持面向对象、命令式和声明式（如函数式编程）风格。

3.1.6 jQuery

jQuery 是一个快速，小型且功能丰富的 JavaScript 库。借助易于使用的 API（可在多种浏览器中使用），它使 HTML 文档的遍历和操纵，事件处理，动画和 Ajax 等事情变得更加简单。兼具多功能性和可扩展性，jQuery 改变了数百万人编写 JavaScript 的方式。

jQuery 的核心特性可以总结为：具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的 css 选择器，并且可对 CSS 选择器进行扩展；拥有便捷的插件扩展机制和丰富的插件。jQuery 兼容各种主流浏览器，如 IE 6.0+、FF 1.5+、Safari 2.0+、Opera 9.0 + 等。

3.1.7 Ajax

AJAX就是“Asynchronous JavaScript and XML”，指的是一套结合了JavaScript和xml技术的客户端网页开发技术。Ajax用于开发出用户体验更好，交互性更强的网页技术。使用JavaScript向服务器发送请求并处理这个请求时，不会影响用户刘览器的核心对象XMLHttpRequest（非IE对象）传输数据。通过这个对象，您的JavaScript可在页面整体无刷新的情况与服务器端交换数据，即就是产生局部刷新的效果。Ajax在浏览器（客户端）与Web服务器（服务端）之间使用的是异步数据传输（HTTP请求)这样就可以在页面加载完成后重新发送少量需要更新的数据，而不是重新发送整个页面到客户端，提高了页面加载速度，也有更好的用户体验。

3.1.8 HTML

HTML英文全称翻译过来就是超文本标记（标签）语言，标准通用标记语言下的一个应用。它是一门静态网页制作技术，里面可以很多种标记（标签）来制作网页，最常见的标签有：table标签、input标签、form标签、a标签（超链接）、img标签等。但这些只能做出网页最基本的样式出来，一般的网页都有动态效果，可以通过引入js插件，或者写js代码，可以实现页面的动态效果，还可以引入css样式代码，用来对页面的布局、字体、颜色、背景和其它效果进行修改，让页面更好看。

3.1.9 JSP

JSP（全称JavaServer Pages）是由Sun Microsystems公司主导创建的一种动态网页技术标准。JSP部署于网络服务器上，可以响应客户端发送的请求，并根据请求内容动态地生成HTML、XML或其他格式文档的Web网页，然后返回给请求者。JSP技术以Java语言作为脚本语言，为用户的HTTP请求提供服务，并能与服务器上的其它Java程序共同处理复杂的业务需求。

## 3.2 系统开发平台及运行环境

硬件资源：CPU i7-8750H 2.20GHz 16GB内存

软件资源：1. 开发平台操作系统：Win7/Win10/Win11操作系统

数据库：MySql

2．开发工具：idea

3. java环境：jdk-1.8

运行环境：tomcat服务器，谷歌浏览器

# 第4章 需求分析

## 4.1 系统功能模块概述和分析

项目采用SSM框架技术来实现人事管理系统，其中web层（表示层）：该层主要包括SpringMVC中的controller类和JSP页面。Service层（业务逻辑层）：service接口和实现类。Dao层（数据访问层）：DAO接口和mybatis映射文件组成。项目基本的两个功能包含用户登陆和用户退出，默认的管理员用户名为173,密码是111222，该项目的主体功能包含员工管理、请假管理、加班管理、出差管理、统计管理、公告管理、部门管理，角色管理八个功能模块。其中员工管理有添加、编辑和查询员工信息功能，请假管理有请假申请和审批功能，加班管理包括加班申请和加班审批功能，出差管理包括出差申请和审批功能，统计有部门人数图表统计，公共管理有添加和编辑功能，部门管理有部门查询和添加部门两个功能，角色管理有查询和添加角色两个功能，每个模块中的查询界面都包含修改、删除和条件筛选功能。每个以表格形式呈现的数据都可以下载和打印。

一般来说，我们需要完成以下的要求才能获得公众的认可。这使得我们的系统开发更有价值。

（1）数据库设计的合理性：数据冗余应尽可能控制在最小范围内，以免重复数据浪费太多存储空间。同时，也减轻了服务器的操作负担，提高了计算机的整体运行速度。

（2）界面整体设计的友好性：用户界面的友好性大多数情况下作为用户判断系统好坏的第一主观感受。友好的用户界面往往能在第一时间让用户心情愉快，从而进一步提升用户的使用感受，因此界面的友好性是整个系统及其重要的组成部分。

（3）主要功能与特殊功能兼备：本系统明确要求除了能够实现信息的录入，删除，查询等基础的功能之外，还要求能够按照用户的需求进行相对应的操作。

## 4.2 系统功能模块设计

4.2.1 系统总架构

图3-1是系统设计的总体框架图，通过图示的形式，描述了系统各部分的功能模块结构关系。图中的每一个模块都能够完全独立地实现其具体的功能。

图4-1 人事管理系统总体框架图

4.2.2 系统功能总体设计结构

为了实现本系统目标，将系统分为员工管理、请假管理、加班管理、出差管理、统计管理、公告管理、部门管理、角色管理8个核心功能管理模块，每个模块都实现了具体增、删、改、查功能，根据系统的整体架构分析出每个管理模块应具备的功能，如图4-2所示是整个系统的总体设计结构图

图4-2 系统功能总体设计结构图

4.2.3 系统权限设计结构

下图4-3、4-4、4-5分别是管理员、部门经理和员工登录系统后的权限图，如下所示：

图4-3 管理员权限结构图

图4-4 部门经理权限结构图

图4-5 员工权限结构图

## 4.3 数据库设计

4.3.1 数据库概念结构设计

根据系统的需要，数据项和数据结构设计出满足用户需求的一系列实体，并设计出它们相互之间的关系，为后边的逻辑结构设计打下基础。这些实体包括各种各样的具体信息，通过它们相互之间的作用形成数据的流动。

该系统依据上面的设计规划出的实体及其属性有，如下：

1. 员工信息{id工号，姓名，性别，学历，工资，密码，入职时间，出生日期，手机号，邮箱}
2. 公告{公告id，标题，发布时间，谁可以看，内容，发布者，发布者id}
3. 出勤{签到时间，最迟签到时间，签退时间}
4. 部门{部门id，部门名，部门地址，创建时间}
5. 员工部门{员工部门id，员工表id，部门表id}
6. 员工角色{员工角色id，员工id，角色id}
7. 员工请假信息{请假id，员工id，开始时间，结束时间，请假理由，填表时间，审批状态，请假天数}
8. 员工加班信息{加班id，员工id，加班天数，开始时间，结束时间，理由，填表时间，状态}
9. 员工头像{员工id，最近更改日期，图片名，图片路径}
10. 员工出勤{考勤id，员工号id，签到时间，签退时间，考勤当天日期，备注（状态）}
11. 角色名称{角色id，角色名称，角色描述}
12. 出差信息{出差id，员工id，开始时间，结束时间，理由填表时间，状态，天数，出差地址}

4.3.2 数据库表设计

相关的概念结构设计完成以后，就能够将上面的相关的概念结构转换为数据库支持的实际数据模型，也就是数据的逻辑结构，建立相应的数据库表，下面是数据库中实体表的详细设计。

本系统包括公告表出勤表，部门表，员工信息表，员工部门表，员工角色表，员工请假信息表，员工加班信息表，员工头像表，员工出勤表，角色名称表，出差信息表，如表3-1所示：

表3-1 数据库实体表

|  |  |  |
| --- | --- | --- |
| **中文表名** | **英文表名** | **备注** |
| 公告表 | announcement | 公告信息 |
| 出勤表 | attendancesetting | 员工出勤信息 |
| 部门表 | depart | 部门信息 |
| 员工信息表 | employee | 员工信息 |
| 员工部门表 | employee\_depart | 员工部门信息 |
| 员工角色表 | employee\_role | 员工角色信息 |
| 员工请假信息表 | leavetime | 请假信息 |
| 员工加班信息表 | overtime | 加班信息 |
| 员工头像表 | picture | 员工头像信息 |
| 员工出勤表 | punchclock | 员工出勤信息 |
| 角色名称表 | role | 角色名称 |
| 出差信息表 | travelschedule | 出差信息 |

各表详细设计如下：

表3-2 announcement公告表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | int | Not null | 公告id（主键） |
| title | varchar | Not null | 标题 |
| releaseTime | date | null | 发布时间 |
| roleToSee | varchar | null | 谁可以看 |
| content | text | null | 内容 |
| author | varchar | null | 发布者 |
| eid | int | null | 发布者id |

表3-3 attendancesetting考勤时间设置表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| starttime | time | Not null | 签到时间（主键） |
| lateststarttime | time | Not null | 最迟签到时间 |
| endtime | time | null | 签退时间 |

表3-4 depart部门表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| dept\_id | int | Not null | 部门id（主键） |
| dept\_name | varchar | Not null | 部门名 |
| dept\_address | varchar | null | 部门地址 |
| established | date | Not null | 创建时间 |

表3-5 employee员工信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | bigint | Not null | id(工号）（主键） |
| name | varchar | Not null | 姓名 |
| sex | varchar | null | 性别 |
| edu | varchar | null | 学历 |
| wage | double | null | 工资 |
| password | varchar | null | 密码 |
| hiredate | date | null | 入职时间 |
| birthday | date | null | 出生日期 |
| phone | varchar | null | 手机号 |
| email | varchar | null | 邮箱 |

表3-6 employee\_depart员工部门表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | int | Not null | 员工部门id（主键） |
| eid | int | Not null | 员工表id |
| did | int | Not null | 部门表id |

表3-7 employee\_role员工角色表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | int | Not null | 员工角色id（主键） |
| eid | int | null | 员工id |
| rid | int | null | 角色id |

表3-8 leavetime员工请假信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | int | Not null | 请假id（主键） |
| eid | int | Not null | 员工id |
| starttime | date | null | 开始时间 |
| endtime | date | null | 结束时间 |
| reason | varchar | null | 请假理由 |
| filldate | date | null | 填表时间 |
| status | varchar | null | 审批状态 |
| days | int | null | 请假天数 |

表3-9 overtime员工加班信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | int | Not null | 加班id（主键） |
| eid | int | Not null | 员工id |
| days | int | null | 加班天数 |
| starttime | date | null | 开始时间 |
| endtime | date | null | 结束时间 |
| reason | varchar | null | 理由 |
| filldate | date | null | 填表时间 |
| status | varchar | null | 状态 |

表3-10 picture员工头像表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| eid | int | Not null | 员工id（主键） |
| date | timestamp | null | 最近更改日期 |
| name | varchar | null | 图片名 |
| path | varchar | null | 图片路径 |

表3-11 punchclock员工出勤表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | int | Not null | 考勤id（主键） |
| employeeid | int | Not null | 员工号id |
| punch\_inTime | datetime | null | 签到时间 |
| punch\_outTime | datetime | null | 签退时间 |
| attendanceTime | date | null | 考勤当天日期 |
| ps | varchar | null | 备注 |

表3-12 role角色名称表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | bigint | Not null | 角色id（主键） |
| name | varchar | null | 角色名称 |
| desc\_ | varchar | null | 角色描述 |

表3-13 travelschedule出差信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **数据类型** | **是否为空** | **含义** |
| id | int | Not null | 出差id（主键） |
| eid | int | Not null | 员工id |
| starttime | date | null | 开始时间 |
| endtime | date | null | 结束时间 |
| reason | varchar | null | 理由 |
| filldate | date | null | 填表时间 |
| status | varchar | null | 状态 |
| days | int | null | 天数 |
| address | varchar | null | 出差地址 |

# 第5章 设计与实现

系统界面是以网页作为主要的交互方式，由HTML+JSP组合而成。是用户直观操作的对象，用户通过点击各种请求按钮将请求发送到控制层组件中，作为MVC中表现层，将数据以网页的形式显示给用户，控制层主要负责控制转发，数据处理。

5.1 关键代码

通过自定义设置Shiro中的Realm，用于认证和授权用户权限。用户角色权限验证，通过用户id获取角色权限，登录控制，通过账号id获取登录密码

@Override

protected AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principalCollection) {

//验证通过

int id = (int) principalCollection.getPrimaryPrincipal();

//通过id获取角色和权限

Set<String> roles = roleService.listRoles(id);

//Set<String> permissions = permissionService.listPermissions(id);

SimpleAuthorizationInfo simpleAuthorizationInfo = new SimpleAuthorizationInfo();

//simpleAuthorizationInfo.setStringPermissions(permissions);

simpleAuthorizationInfo.setRoles(roles);

System.out.println(roles);

return simpleAuthorizationInfo;

}

@Override

protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken token)

throws AuthenticationException {

//获取账号id和密码

UsernamePasswordToken t = (UsernamePasswordToken)token;

String sid = token.getPrincipal().toString();

// int id = Integer.valueOf(sid);

int id = Integer.parseInt(sid);

String password = new String(t.getPassword());

char[] password1 = t.getPassword();

String pwd = String.valueOf(password1);

//获取数据库中的密码

String passwordInDB = employeeService.getPassword(id);

if (!passwordInDB.equals(password) || passwordInDB == null) {

throw new AuthenticationException();

}

return new SimpleAuthenticationInfo(id,password,getName());

}

通过用户的登录信息去Shiro中查找用户的权限，从Shiro的Subject中获取当前操作用户的权限进行判断，跳转相应的界面并将当前操作用户的信息传入session中存储起来。如果用户信息登录信息错误则返回登录页面并提示。

/\*用户登录\*/

@Controller

public class LoginContreller {

@RequestMapping(value="login", method = RequestMethod.POST)

public String login(Model model, String id, String password) {

Subject subject = SecurityUtils.getSubject();

System.out.println(subject.getSession());;

UsernamePasswordToken token = new UsernamePasswordToken(id, password);

try {

subject.login(token);

Session session = subject.getSession();

session.setAttribute("subject", subject);

if (subject.hasRole("admin")) {

System.out.println(subject.hasRole("admin"));

return "adminIndex";

} else if (subject.hasRole("employee")) {

System.out.println("employeeIndex");

return "employeeIndex";

} else {

System.out.println("managerIndex");

return "managerIndex";

}

} catch (Exception e) {

model.addAttribute("error", "工号id或密码错误!");

return "login";

}

}

}

在配置文件中配置Shiro的过滤工厂类及权限配置所有的请求(除去配置的静态资源请求或请求地址为anon的请求)都要通过登录验证,如果未登录则跳到/login

<!-- 配置shiro的过滤器工厂类，id- shiroFilter要和我们在web.xml中配置的过滤器一致 -->

<bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean">

<!-- 调用我们配置的权限管理器 -->

<property name="securityManager" ref="securityManager" />

<!-- 配置我们的登录请求地址 -->

<property name="loginUrl" value="/login" />

<!-- 如果您请求的资源不再您的权限范围，则跳转到/403请求地址 -->

<property name="unauthorizedUrl" value="/unauthorized" />

<!-- 退出 -->

<property name="filters">

<util:map>

<entry key="logout" value-ref="logoutFilter" />

</util:map>

</property>

<!-- 权限配置 -->

<property name="filterChainDefinitions">

<value>

<!-- anon表示此地址不需要任何权限即可访问 -->

/login=anon

/resources/\*\*=anon

/doLogout=logout

<!--所有的请求(除去配置的静态资源请求或请求地址为anon的请求)都要通过登录验证,如果未登录则跳到/login -->

/\*\* = authc

</value>

</property>

</bean>

5.2 登录设计

整个系统共用一个登录页面，通过权限分配不同首页面，通过员工的工号（也就是id）和密码实现登录，不支持修改密码，修改密码按钮只是为了提醒用户当忘记密码时提示用户联系管理员修改密码。



图5-1 登录页面

管理员登录后的首页面，页面左侧和头部分为功能选择区域，中间为考勤签到和签退功能，系统管理员为该系统最大权限管理者，有该系统的所有权限，能对考勤的签到和签退时间进行修改以及部门和角色管理。

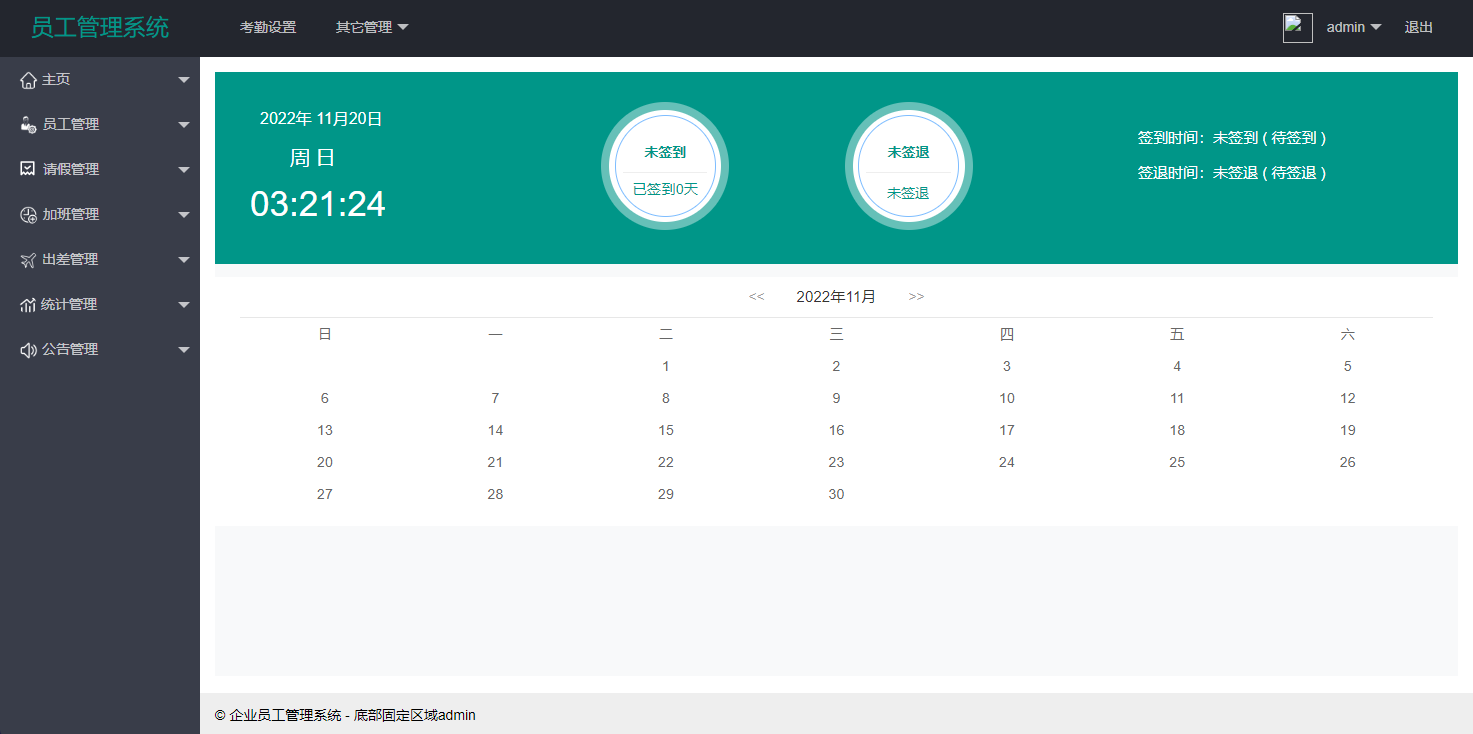


图5-2 管理员首页

管理员查看所有员工的基本信息，其中的功能包括选择查询，员工信息的修改、删除和增加，能对数据进行导出、打印和筛选列。

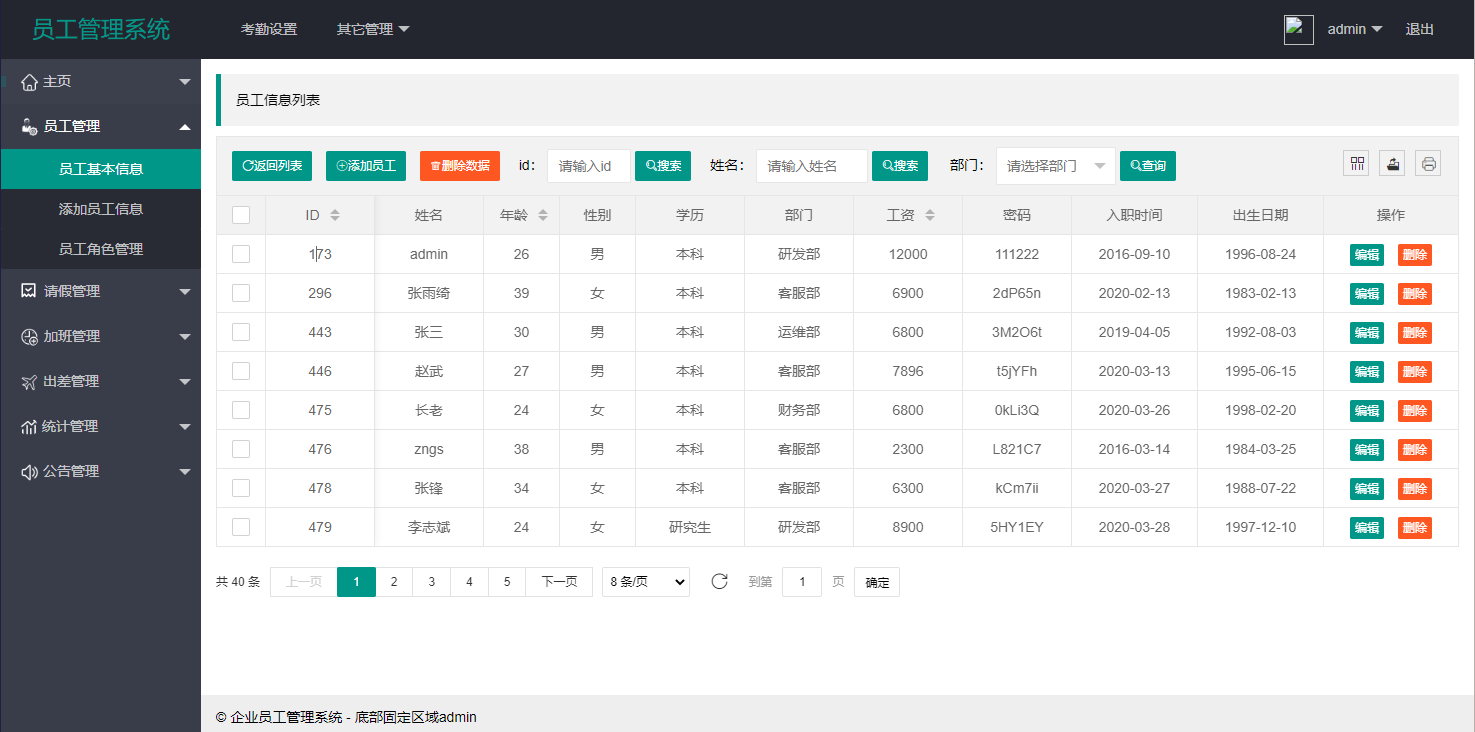


图5-3 （管理员）员工基本信息界面

添加员工信息，其密码为随机生成的，点击生成密码即可



图5-4 （管理员）添加员工信息界面

员工角色管理界面，可通过条件查询相应的员工角色信息，包括删除和修改员工角色信息。

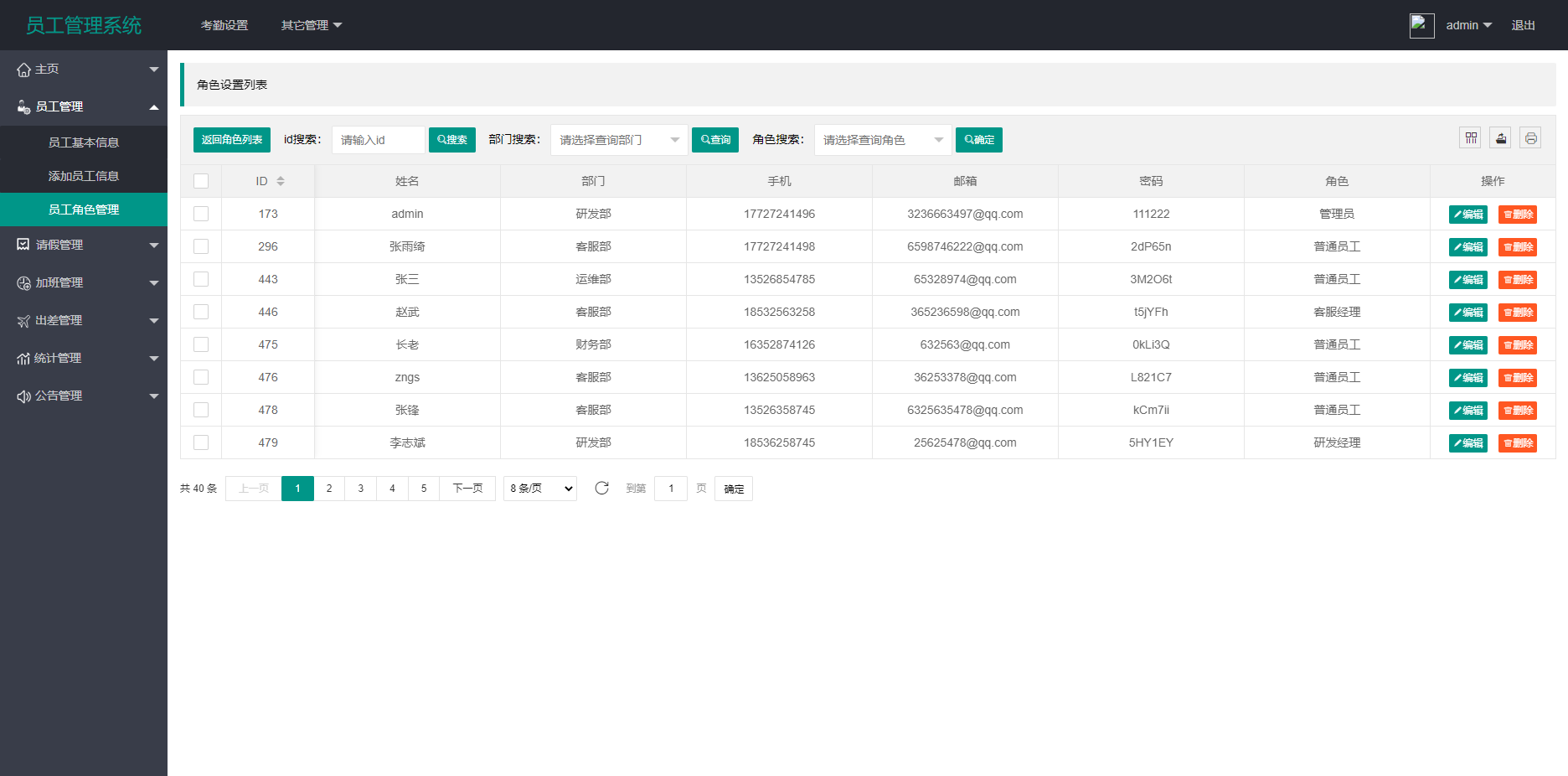


图5-5 （管理员）员工角色管理界面

请假审批界面，所有角色同用一个界面，自动获取当前登录的用户信息所属部门填入表中，不能进行修改，并获取当前的时间日期。



图5-5 请假申请界面

请假记录查看，只能查看当前登录用户的请假记录信息和审核状态。

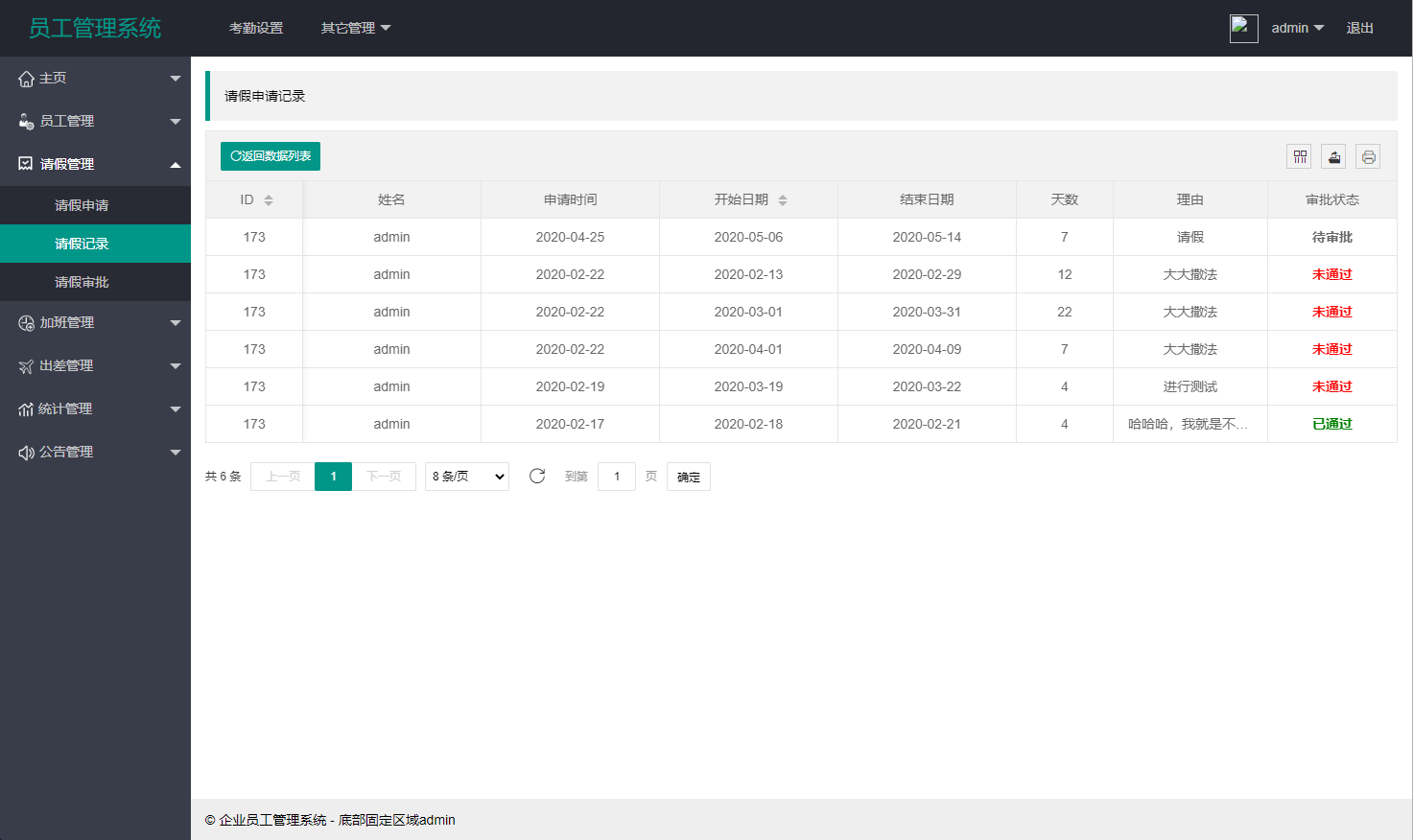


图5-6 请假记录界面

请假审批，管理员可以查看到所有的请假信息包括对员工的请假进行审批。能够对数据进行打印和导出。

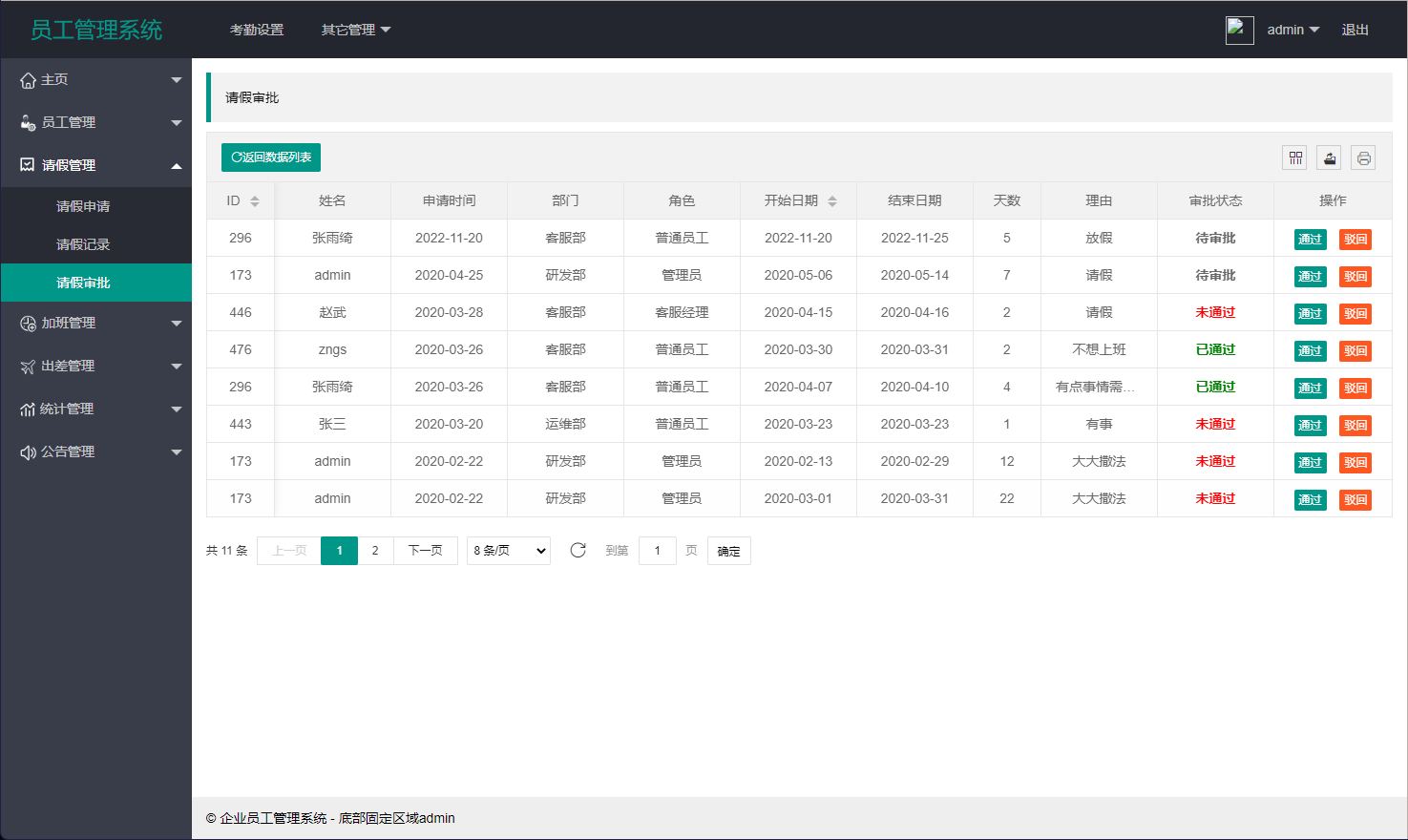


图5-7 （管理员）请假审批界面

加班申请，获取当前登录用户的信息和部门自动填入表格，并且不能进行修改。



图5-8 加班申请界面

加班记录，查看当前登录用户的加班信息，和审核状态，表格模板和请假审批的表格模板一致。能对数据进行导出和打印。

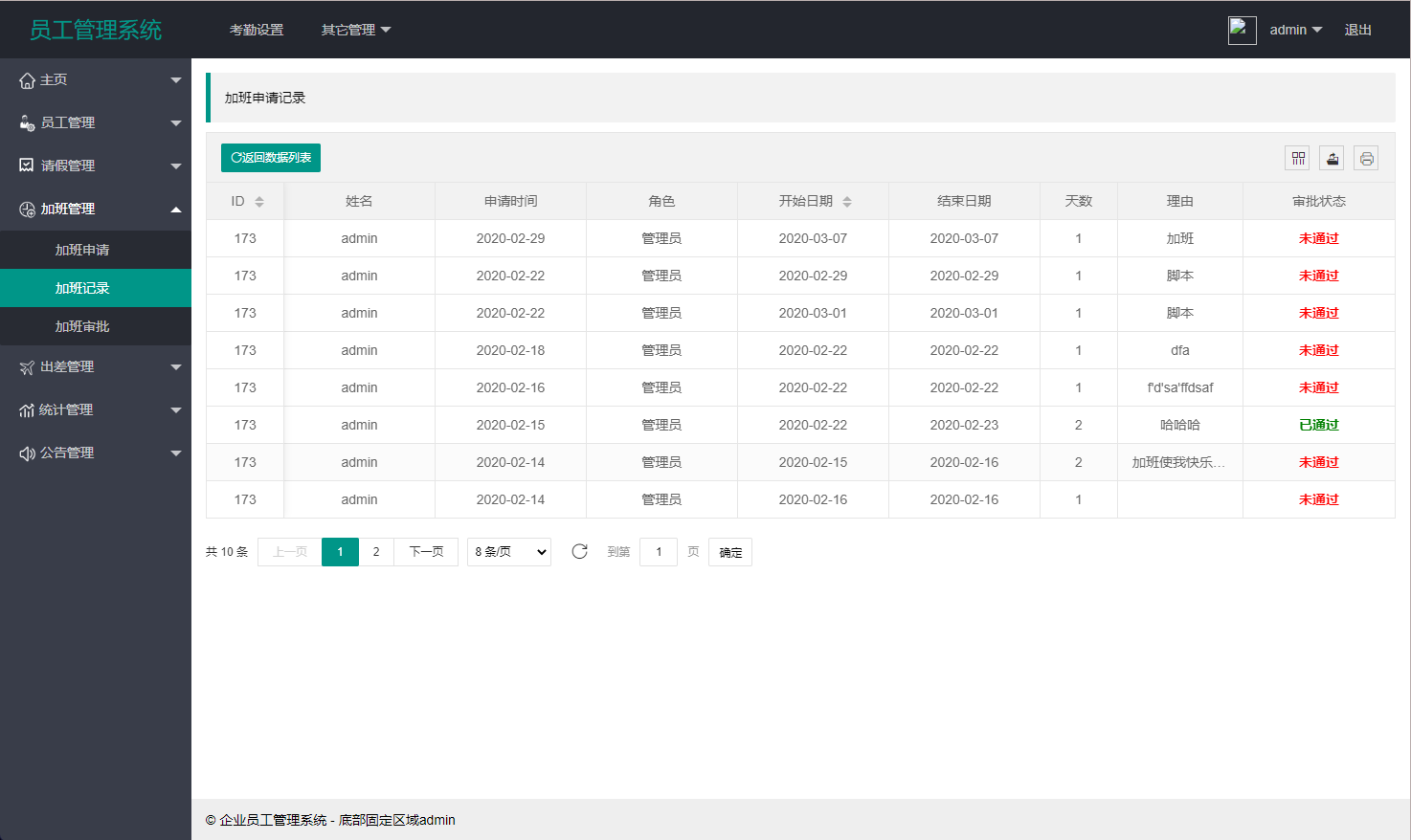


图5-9 加班记录界面

加班审批，对员工所有的加班申请进行审批，表格模板和请假审批表格一致，可以对数据导出和打印。

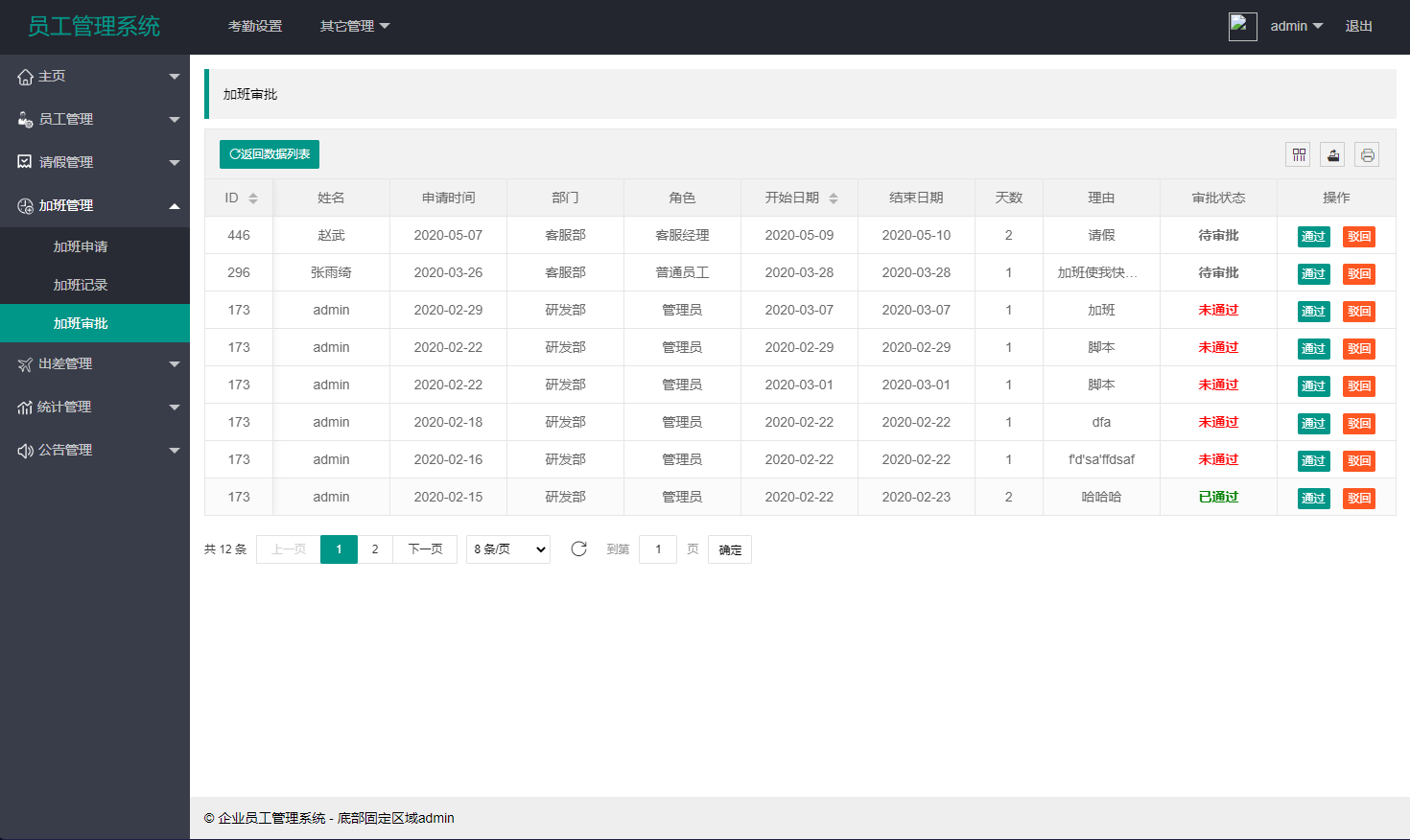


图5-10 加班审批界面

出差申请界面，自动获取当前登录用户信息和部门进行填入，不支持修改。

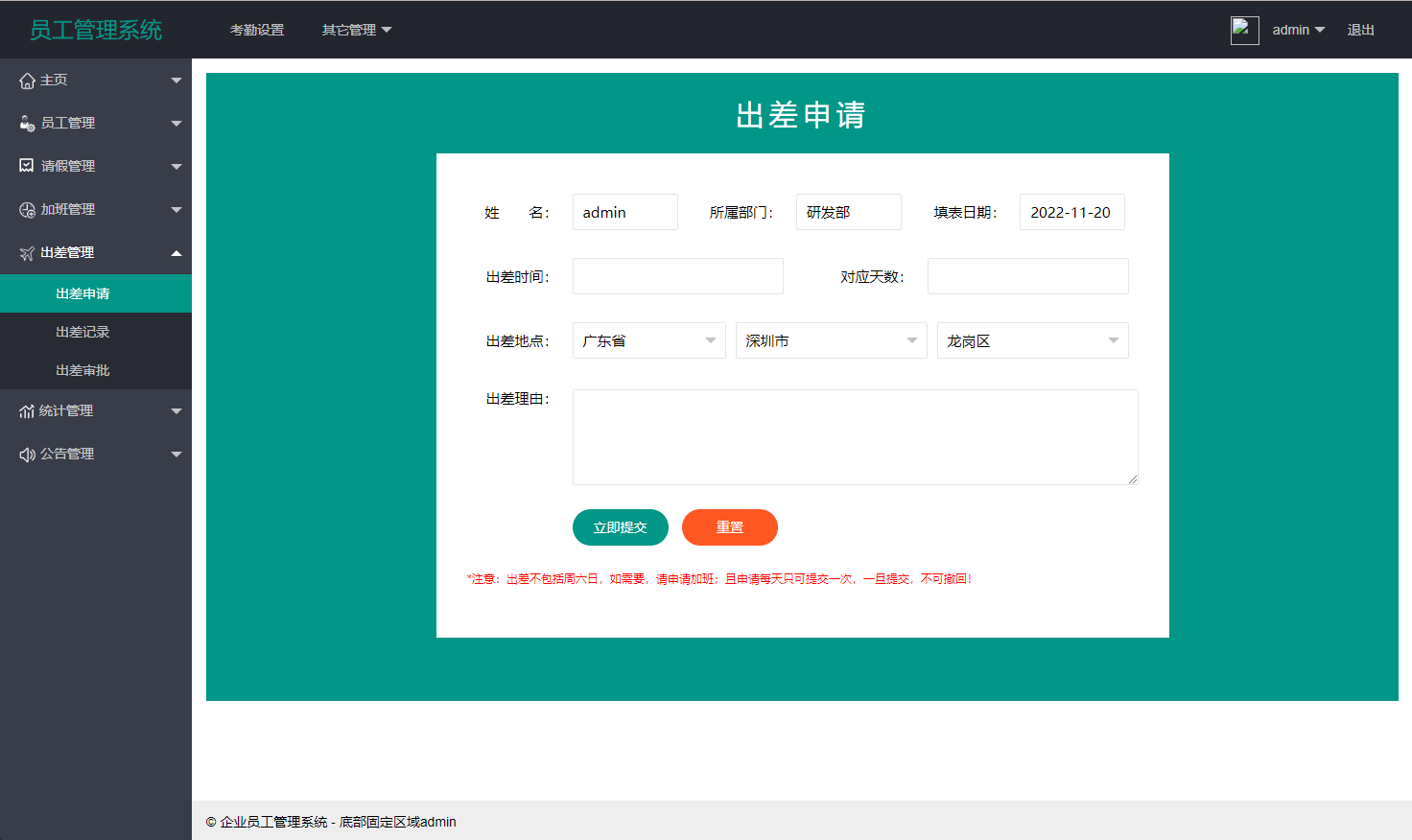


图5-11 出差申请界面

部门人数统计界面，统计部门人数以直方图和饼状图的形式展示出来，更加直观。

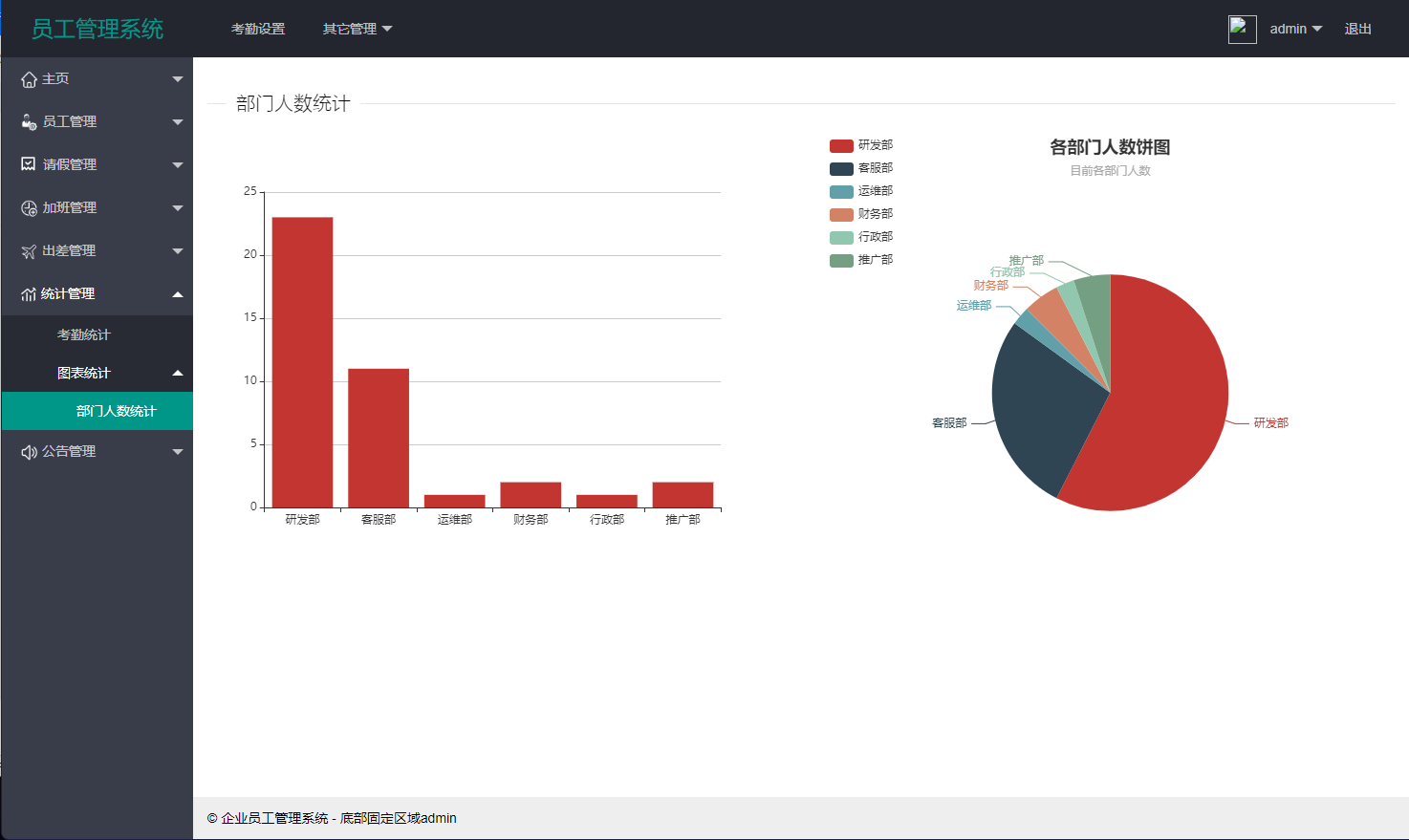


图5-12 （管理员）部门人数统计界面

管理员可对公告进行管理，修改、添加和删除，支持导出和打印，其中公告可以设置为在不同的角色权限下是否可查看。

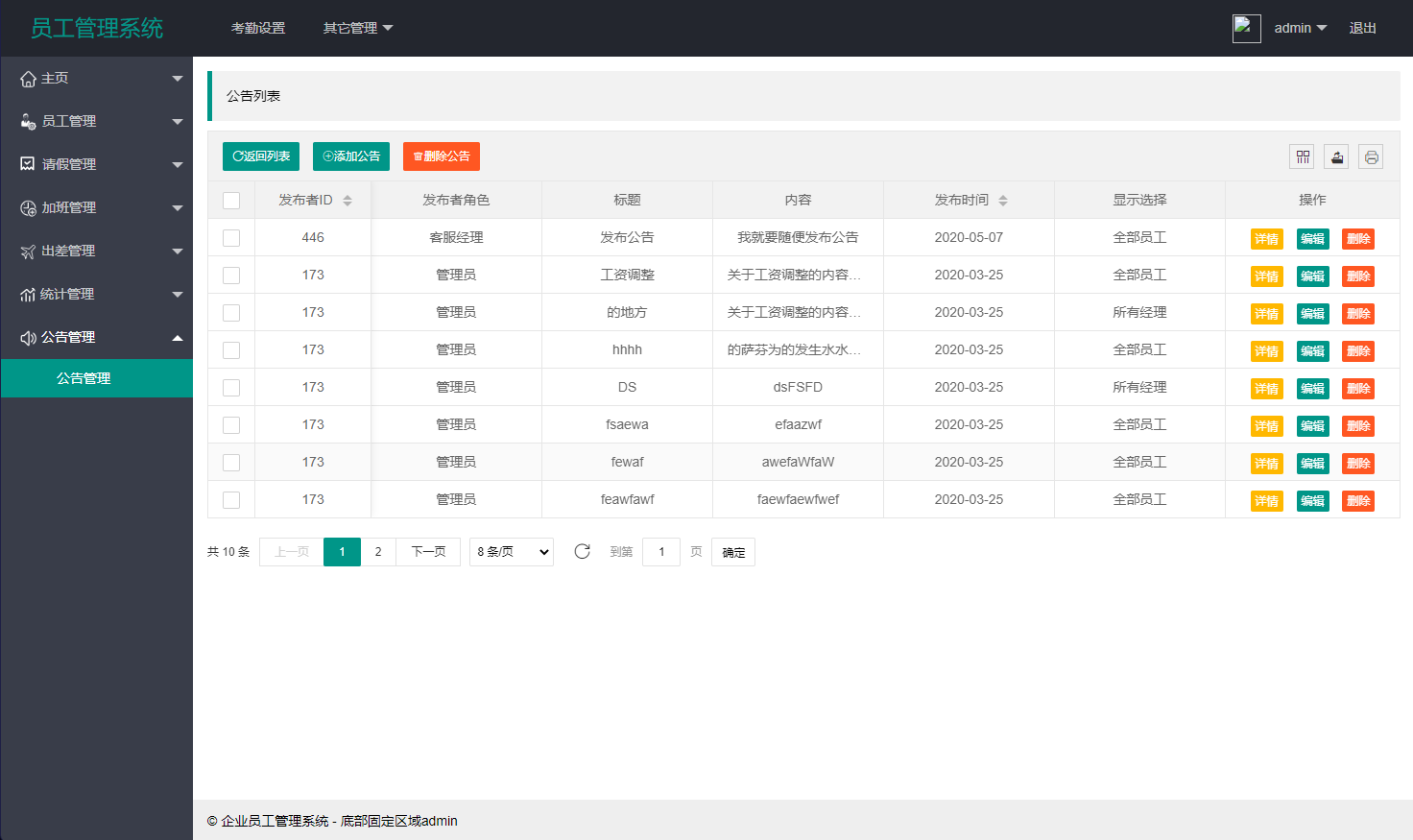


图5-13 公告管理界面



图5-14 添加公告

部门管理界面，管理员可以对部门进行增加、修改和删除操作。

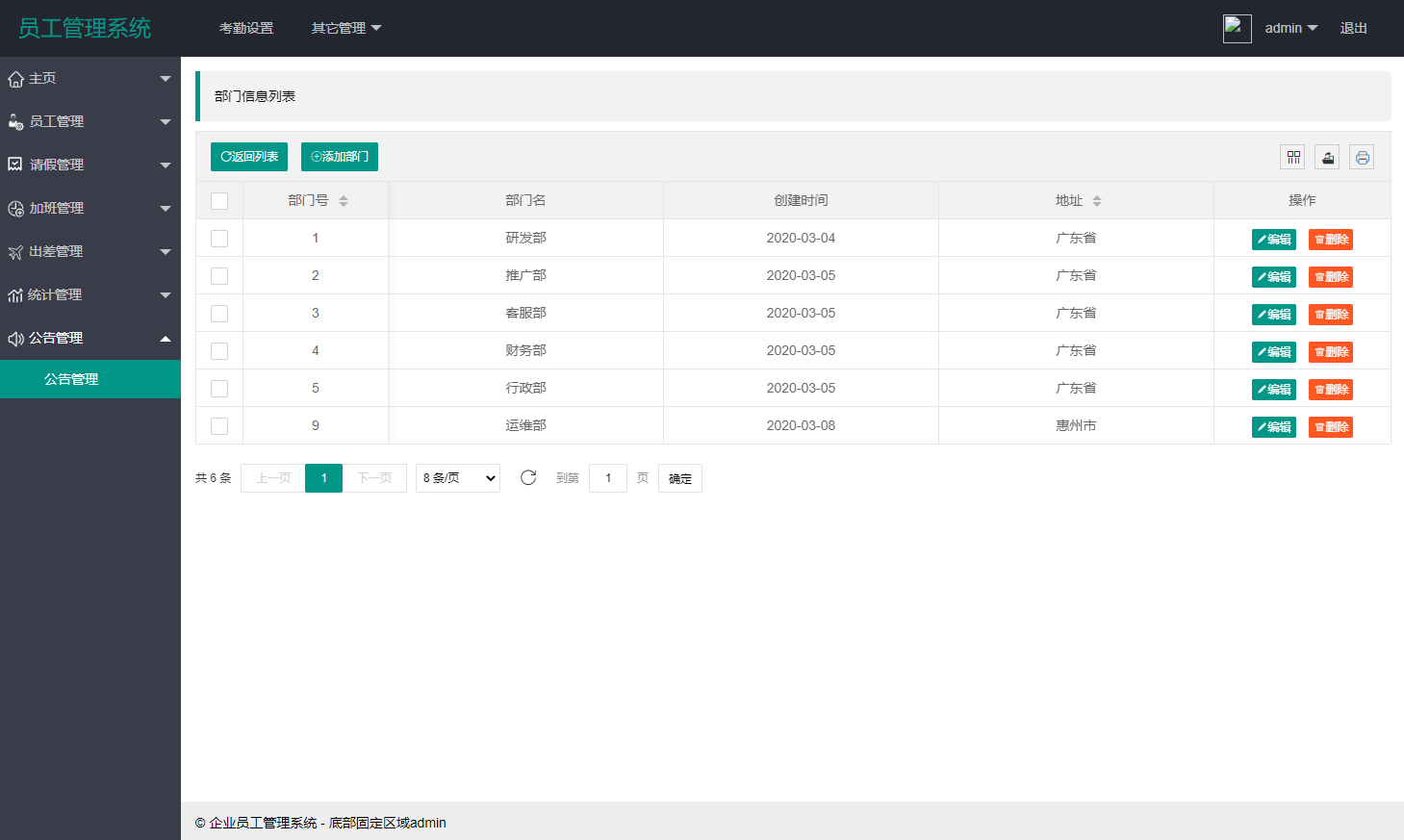


图5-15 （管理员）部门管理界面

管理员可以对系统的角色进行添加、修改和删除等操作。

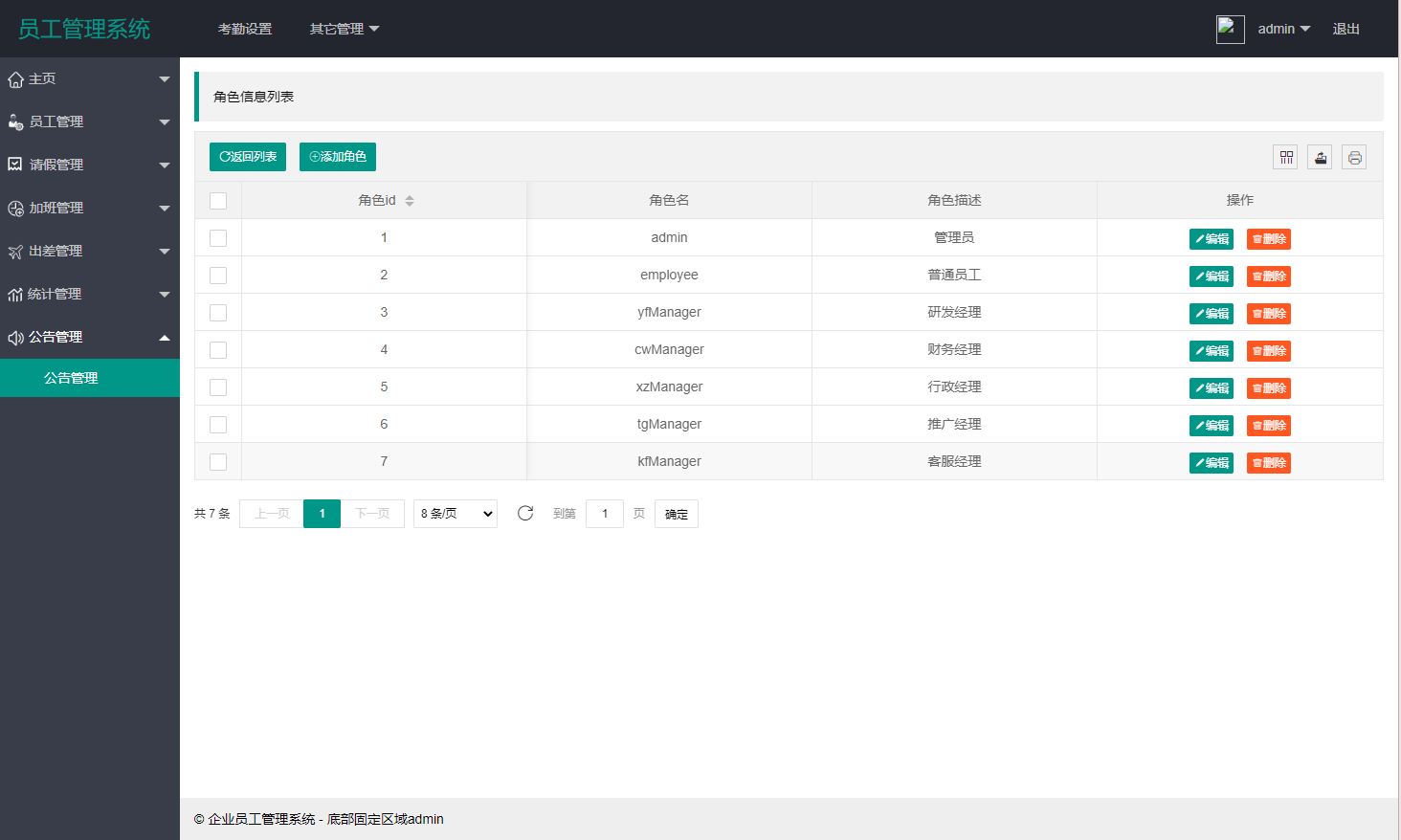


图5-16 （管理员）角色管理界面

普通员工首页，其左侧功能栏包括个人信息查看和修改，考勤记录查看，请假、加班和出差申请和记录查看。其头部还包含一个功能查看发布的公告。

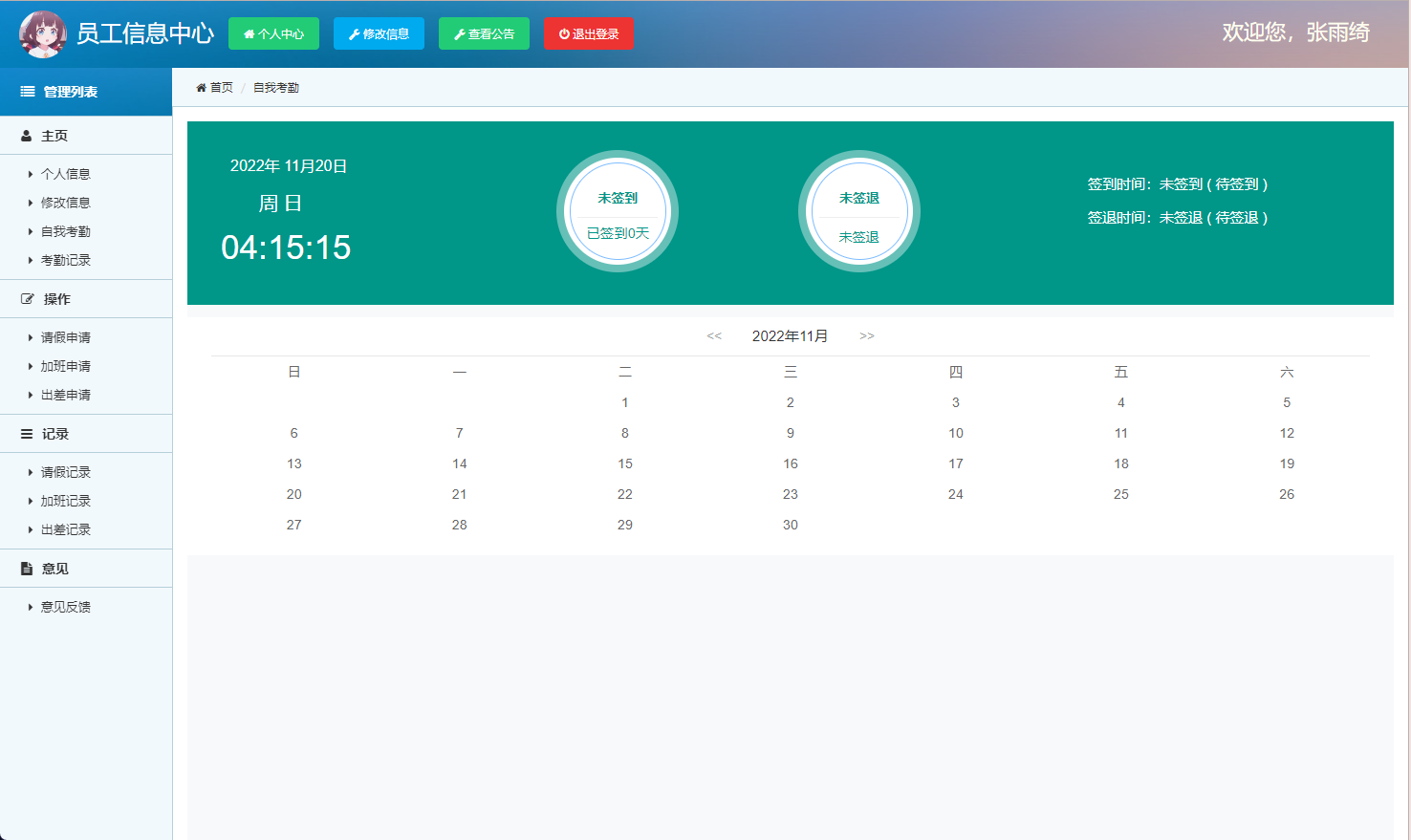


图5-17 普通员工首页

经理首页，和普通员工首页一样，但显示的功能比普通员工多。包括能对当前经理部门下的员工信息进行添加、修改和删除操作，还能对部门人员的请假、加班、出差申请进行审批，发布和删除公告。

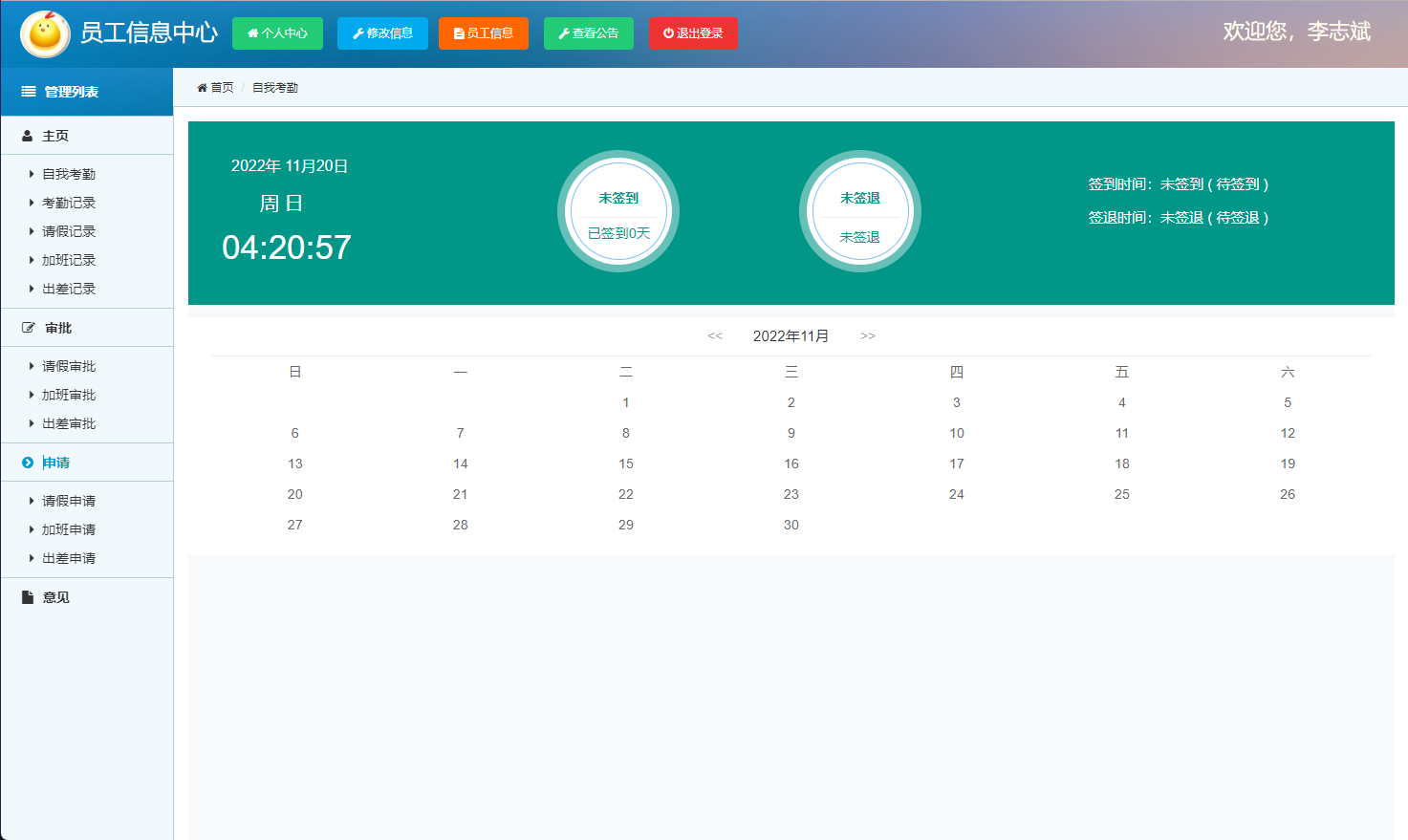


图5-18 经理首页

该系统中所有以表格展示的数据都支持导出和打印。

# 第6章 结论

在这段做实训项目的时间里，把所学的专业知识在实际操作中运用的滋

味是“苦涩”又“甘甜”的。

“苦涩”在于发现自己的专业知识匮乏，发现了自己的不足之处。当遇到困

难的时候，手足无措的时候，进度就降了下来，求助于老师同学，才得以继续下

去。这暴露了自己的依赖性，缺乏独立思考的能力。因为自己的水平有限，一些

好的设想，好的功能最终却没能把它实现。而“甘甜”在于自己收获了不少的东

西，在遇到问题解决问题后，我学会了细心总结，查阅资料，试图寻找别的可能

性，促使了自己的进步，每当自己完成了一小部分的时候，都会雀跃无比，为将

来累计经验，使得自己得能力进一步的提升。

对于即将要走向工作岗位的我，需要更快的从学生转变为工作者，所以在以

后的日子里要不断的努力提升自己，制定好未来的计划，在“苦涩”中磨砺自

己，在“甘甜”中收获和提升自己，继续奋发前进，实现自己心中所追求的目

标。

本系统有不完善的地方，欢迎各位师长和同行提出宝贵的意见。

|  |  |  |  |
| --- | --- | --- | --- |
| 指导教师  评阅意见 |  | | |
| 报告成绩 |  | | |
| 指导教师签字 |  | 评 阅 日 期 | 年 月 日 |