

# 青岛理工大学

## 毕业设计（论文）

题目： 基于 SSM 的运维管理系统的设计与实现

院 别： 信息与控制工程学院

专业班级： 软件工程 164 班

学生姓名： 侯建伟

学 号： 201607344

指导教师： 李兰 史大鹏

2020 年 月 日

## 摘 要

运维管理系统的主要作用是实现企业运维的高效化、智能化的管理。通过使用运维管理系统对企业进行高效管理，可有效地降低企业的人事管理成本，增强企业的核心竞争力。

通过对国内外不同类型的运维管理系统进行横向对比，充分了解不同运维管理系统在功能上的优缺点，并通过查阅运维管理相关资料，确定了本系统的三大模块，分别是非注册用户模块、普通员工模块与管理员模块。三个模块主要实现了企业信息管理，人事管理，招聘管理等相关功能。

本项目整体采用了 SSM 框架进行开发，并使用 B/S 架构作为系统的架构模式，数据库采用 MySQL 进行搭建。前端页面设计主要使用了 bootstrap 与 Layui。同时使用 Apache 公司旗下的软件 ECharts 实现数据可视化。除此之外，本项目还使用到了 Ajax 与 session 技术来优化页面数据交互。

关键词：运维管理系统， B/S 架构， SSM 框架， MySQL 数据库，数据可视化。

## ABSTRACT

The main function of operation and maintenance management system is to realize the efficient and intelligent management of enterprise operation and maintenance. By using operation and maintenance management system for efficient management of enterprises, it can effectively reduce the cost of personnel management and enhance the core competitiveness of enterprises.

Through the horizontal comparison of different types of operation and maintenance management systems at home and abroad, fully understand the advantages and disadvantages of different operation and maintenance management systems in terms of functions, and through consulting the relevant information of operation and maintenance management, determine three main modules of this system, which are the unregistered user module, the ordinary staff module and the administrator module.

The whole project uses SSM framework for development, and uses B / S structure as the structure mode of the system, and using MySQL for database construction. The front page design mainly uses bootstrap and Layui. And I also use the software Echarts that designed by Apache Company to realize data visualization. In addition, this project also uses Ajax and session technology to optimize page data interaction.

**KEY WORDS:** Operation and maintenance management system, B / S structure, SSM framework, MySQL database, Data visualization.

# 目录

|                         |    |
|-------------------------|----|
| 第 1 章 绪 论.....          | 1  |
| 1.1 研究背景及意义.....        | 1  |
| 1.2 运维管理系统国内外研究现状.....  | 2  |
| 1.2.1 国外发展现状.....       | 2  |
| 1.2.2 国内发展现状.....       | 2  |
| 1.3 研究内容.....           | 3  |
| 1.4 论文结构.....           | 3  |
| 第 2 章 系统相关概念和技术.....    | 5  |
| 2.1 B/S 架构.....         | 5  |
| 2.2 JSP 技术.....         | 5  |
| 2.3 SSM 框架.....         | 6  |
| 2.3.1 Spring 框架.....    | 6  |
| 2.3.2 SpringMVC 模式..... | 7  |
| 2.3.3 MyBatis 框架.....   | 7  |
| 2.4 MySQL 数据库.....      | 8  |
| 2.5 AJAX 技术.....        | 9  |
| 第 3 章 系统设计.....         | 10 |
| 3.1 需求分析.....           | 10 |
| 3.1.1 系统需求分析.....       | 10 |
| 3.1.2 员工端需求分析.....      | 11 |
| 3.1.3 管理员端需求分析.....     | 12 |
| 3.2 系统功能分析.....         | 13 |
| 3.2.1 未注册用户角度分析.....    | 14 |
| 3.2.2 员工角度分析.....       | 14 |
| 3.2.3 管理员角度分析.....      | 16 |
| 3.3 数据库设计.....          | 17 |
| 3.3.1 数据库概念结构设计.....    | 17 |
| 3.3.2 数据库表结构设计.....     | 18 |
| 第 4 章 系统详细设计和实现.....    | 21 |
| 4.1 用户登录与新用户注册模块.....   | 21 |
| 4.1.1 注册功能实现.....       | 21 |
| 4.1.2 登录功能实现.....       | 23 |
| 4.1.3 忘记密码功能实现.....     | 25 |
| 4.2 用户管理模块.....         | 28 |
| 4.3 员工管理与企业信息管理模块.....  | 31 |
| 4.4 公告管理模块.....         | 34 |
| 4.5 下载中心模块.....         | 36 |
| 4.5.1 文件的上传.....        | 36 |
| 4.5.2 文件下载.....         | 37 |
| 4.6 培训管理模块.....         | 39 |
| 4.7 考勤管理.....           | 40 |
| 4.8 数据可视化模块.....        | 43 |

|                       |    |
|-----------------------|----|
| 第 5 章 系统测试.....       | 45 |
| 5.1 测试类型.....         | 45 |
| 5.2 测试用例.....         | 45 |
| 5.2.1 登录测试用例.....     | 45 |
| 5.2.2 注册测试用例.....     | 46 |
| 5.2.3 员工信息管理测试用例..... | 46 |
| 5.2.4 下载资料管理测试用例..... | 47 |
| 5.2.5 招聘管理测试用例.....   | 48 |
| 第 6 章 总结与展望.....      | 50 |
| 6.1 总结.....           | 50 |
| 6.2 展望.....           | 50 |
| 致谢.....               | 51 |
| 参考文献.....             | 52 |

# 第1章 绪论

随着当下网络的不断普及和新技术的不断推出，越来越多的企业使用信息化的运维管理系统对企业进行规范化，信息化的管理。运维管理系统作为企业运行不可或缺的一部分，它所扮演的是一个为企业内部员工和管理者提供信息资源和内部服务的重要角色，并面向外部人员提供信息。

## 1.1 研究背景及意义

运维管理工作最重要的就是对企业内各种资源进行管理，为企业管理者和员工提供应有的技术支持和帮助。人类的各种活动都是从最初的劳动开始，随着科技的不断进步，人类的活动开始慢慢借助科技的手段加以实现。运维管理系统也不例外，过去的运维管理的实现是基于人事部门的手工记录和整理。人事部门的工作人员将新员工资料逐一记录，然后将信息整合，最后保存到企业档案中。就这样，才完成了录入一个新员工基本资料的操作。而随着企业雇佣的人数增加，人事部门的工作压力也就逐渐增大，过程也就越发繁琐，且一旦出错，将对企业造成直接的经济损失，更甚会给企业带来不必要的麻烦。小型企业的运维管理通过人力部门的手动管理可以应对，但是随着企业规模的扩张，久而久之，手动管理必将被淘汰。对于大型企业来说，引进一套先进的运维管理系统进行人事管理是十分必要的。

运维管理系统已经成为现今大部分企业一个比较普遍的应用，信息化技术已经深入到企业中的各个职能部门<sup>[1]</sup>。不论是员工信息的管理，或是企业岗位的调整，都不再是纯手工的记录。从企业管理者的角度来看，实现企业运维管理的现代化、信息化是与时俱进的，适应时代变化的一种表现。从企业人事管理者的角度来看，先进的运维管理系统能提高管理者的办事速度，提高工作效率，更重要的是，实现了无纸化管理，降低了企业生产经营成本。从员工的角度来看，先进的运维管理系统可以让员工更加快速的融入集体中，及时了解企业最新的动向，提高了员工的积极性，进而为企业创造更大的价值。

虽然很大一部分企业已经基本实现了系统化的运维管理，但是对于一些小型企业和使用落后运维管理系统的企业来说，他们的管理方法或系统已很难满足当下的需求。比如说，某些基于旧平台开发的运维管理系统因界面设计不够友好，操作不流畅，业务逻辑不清晰等原因，无法胜任日常的企业运维管理功能。又或是系统的安全性不到位，后期维护因为底层语言版本过旧所导致的新功能的部署

极为困难，同样也会被淘汰。因此结合实际使用过程中可能会出现不足之处，需要设计出一套详实完善的运维管理系统来提高工作效率，使得企业的运维管理更加高效，便捷。针对以上情况，开发出一套让企业全部员工借助网络和现有设备使用运维管理系统的工作刻不容缓，这可以为广大企业管理者和员工提供极大地便利。所以说，运维管理系统的设计是一项有着重大意义的工作，项目后期前景也十分乐观。

## 1.2 运维管理系统国内外研究现状

随着科学技术的不断进步与个人电脑在企业中的普及，运维管理也加速进入了智能化、网络化的发展潮流。由国外主导并研发的运维管理系统在 20 年代末就已开始广泛运用，运维管理系统由于其高效的管理能力，方便快捷的信息共享机制，广泛的得到全球关注，并不断地为社会发展，企业管理等领域贡献属于自己的力量。

### 1.2.1 国外发展现状

因为国外计算机与网络在 20 世纪末广泛应用于银行、证券、保险、医疗和政府等重要职能部门，所以国外的运维管理系统起步相对较早，随着系统和技术的不断升级，国外的运维管理系统已经形成了一套成熟的，相对完善的体系。到现在，很大一部分企业和大型金融机构仍然使用国外的运维管理软件<sup>[2]</sup>。

且这些公司在早期就已实现了运维管理系统的商品化服务，许多知名企业也加入了运维管理系统的开发与设计之中，比如惠普公司在 2014 年推出的 HP Operation Management Service 以及卓豪公司旗下的 ManageEngine。这些公司研发的产品使用者多，且覆盖面广，在全球范围内有着数以万计的忠实用户。

### 1.2.2 国内发展现状

相比于国外，我国由于历史条件以及计算机和计算机网络普及时间较晚等原因，导致我国运维管理系统的开发与使用落后于国外，适用范围也相对小众<sup>[2]</sup>。在运维管理系统刚被引入时，大部分企业不看好运维管理系统，且很大一部分企业仍旧坚持使用传统的基于人工管理的运维模式。在 2000 年前后，我国大型企业起头使用非我国自主研发的运维管理系统对企业进行运维管理。近几年来，国家加大对计算机软硬件开发企业的政策扶持，各大软件公司加快了运维管理系统的开发进度，很大程度上加快了我国运维管理系统国产化的发展进程。

从第一个运维管理系统在国内出现，到现在已经有基于不同门类，适用于不

同环境的运维管理系统<sup>[3]</sup>。截至目前，国内企业大部分采用的运维管理系统是神州信息运维管理系统、阿里云 DevOps 运维管理系统等。

综上所述，国外的运维管理系统相对于国内来说起步更早，且他们在早期就已实现了运维管理系统的商品化服务，并成立了许多知名的系统软件公司。他们在产品需求、技术开发、销售服务和系统维护等方面拥有不俗的实力，经验也相对完善，具有一定的稳定性。虽然国内的运维管理系统的开始研发的时间较晚，但是通过近些年的不断努力，我国的运维管理系统已经拥有了一定的规模，达到了较高的技术水平。但是在功能设计和技术支持等方面，我国自主设计研发的运维管理系统与国外的系统相比，还是存着有一定的差距。我国自主研发和设计的运维管理系统在许多方面仍有较大的进步空间。

### 1.3 研究内容

本论文以企业运维管理系统工作的各个特点，以 B/S 架构和 MVC 设计模式为导向，设计并实现了基于 SSM 框架的运维管理系统。本文将企业日常运维管理工作视为研究对象，通过进行背景研究，从系统的需求确立、软件规格设计、代码编写，项目整体开发实现，具体的讲解了系统中每个模块的作用，论述了系统主要部分的工作原理，力求做到系统界面设计友好，功能具体，符合实际需求。下面将具体介绍本论文的主要研究目标。

(1) 了解企业运维管理系统的相关知识、研究与其相关的技术理论，并结合国内外发展的现状，综合分析得出目前国内外运维管理系统中存在的种种问题，开发和设计出一套功能完善的，界面友好的运维管理系统。

(2) 综合对比现有的运维管理系统，根据软件开发过程，进行需求分析。

(3) 通过完成本次毕业设计，能够独立进行需求分析，并根据需求分析进行系统设计，了解软件开发的过程，熟练掌握数据库的逻辑设计以及表结构设计，并能独立完成的各个功能模块的代码实现。

### 1.4 论文结构

本论文为基于目前运维管理系统的现状，设计一套符合当下企业制度的运维管理系统，本论文的整体组织结构如下：

第一章 绪论：主要对如下两部分进行阐述：首先对运维管理系统的研究背景进行了调研。介绍了运维管理系统在国内外的历史与使用情况。第二点，本章对论文整体结构进行了说明。



第二章 系统相关概念和技术介绍：介绍运维管理系统中使用到的 C/S 架构、SSM 框架、JSP 技术、MySQL 数据库与 AJAX 技术等。

第三章 系统设计：根据企业运维管理的实际需求，结合了多种需求分析方法，包括了需求分析、系统用例分析、绘制用例图等手段明确了运维管理系统的需求。最后进行了数据库设计的详细介绍。

第四章 系统详细设计：主要介绍了运维管理系统中各个子模块的实现过程，用流程图对系统功能进行解析，并节选重要代码进行具体解释说明，最后添加了程序运行的图片。

第五章 系统测试：本章节主要介绍了以下两点内容：第一，结合软件测试相关书籍介绍进行系统测试的重要性。第二，使用两种主流测试技术-黑盒测试与白盒测试对运维管理系统进行综合测试，并在测试完成后整理测试用例。

第六章 总结与展望：对本论文内容进行总结，同时提出了后期需要进一步完善之处。

## 第 2 章 系统相关概念和技术

本章主要对系统所需要的相关概念和技术进行了介绍，包括了 B/S 架构、JSP 技术，SSM 框架，MySQL 数据库，AJAX 技术。

### 2.1 B/S 架构

B/S (Browser-Server, 浏览器-服务器) 架构，是 web 开发中两种基本架构之一<sup>[4]</sup>。相对于 C/S (Client-Server, 客户机-服务器) 架构来说，其主要优点是：B/S 架构的维护和升级的方便程度远远高于 C/S 架构。

B/S 架构，主要分为三层，其内容分别是表现层、逻辑层与数据访问层。其中表现层的功能是完成用户和后台交互以及查询出最终结果输出的功能。逻辑层的主要功能是利用服务器来处理客户端发送的各种不同的应用数据的功能。数据层的主要功能是接受到来自客户端的请求之后，进行数据的操作，大多为数据的运算。

对运维管理系统来说，管理人员若采用 C/S 架构进行实施和部署项目，对于小型办公企业或许是可行的。但是对于大型企业来讲，系统维护人员可能需要在几百台甚至上千台不同种类，甚至基于不同平台的客户端进行调试，工作人员的效率是相当低的。

B/S 架构的管理软件只需要安装在服务器端，且在 B/S 架构下，管理人员可以实现热部署，即在服务器端进行代码更新就可实现代码的部署，这有效的降低了开发成本，提高了开发效率。

因为 C/S 架构需要在专用网络下才能运行，且必须要经过多重检验，可以不借助外来的安全系统。相对来说，其后期的代码维护难度较大。B/S 架构可以部署在广域网上，因为其安全性相对差，需要借助专门的安全系统。但是现在的安全系统技术已经非常成熟，也不必担心系统出现较大的安全隐患。

结合以上内容，C/S 框架安全性相对较高，有着一定的专用性，比较适合银行，国家机关等系统需要保持一定稳定性和保密要求较高的企业。对于大型企业来说，因为员工对与计算机技术的了解程度参差不齐，所以 B/S 架构的系统受欢迎程度明显是更高的<sup>[4]</sup>。

### 2.2 JSP 技术

JSP (Java Server Pages) 是由甲骨文公司旗下的 Sun Microsystems 公司主导并创建的一种动态网页技术标准。

部署在服务器上的 JSP 可以响应客户端的请求，并且可以根据请求的内容动态生成 web 页面，并返回给请求发送者。JSP 技术的开发以 Java 语言为编写基础，可以保证软件在各种不同架构下运行和使用<sup>[5]</sup>。此外，JSP 技术的代码编译工作可以一次性完成，且后期的维护和更新相对容易，这也是 JSP 开发技术得到广泛应用的原因。

### 2.3 SSM 框架

SSM 框架，即 Spring、SpringMVC、MyBatis 三大框架技术的简称。2.3.1 到 2.3.3 这三部分将介绍这三大技术的特点。

#### 2.3.1 Spring 框架

Spring Framework 是继承 Web 层、业务层、DAO 层和持久性层的组件的开放源容器框架。

Spring 框架的七大模块以及其常见组件如图 2.3.1 所示。

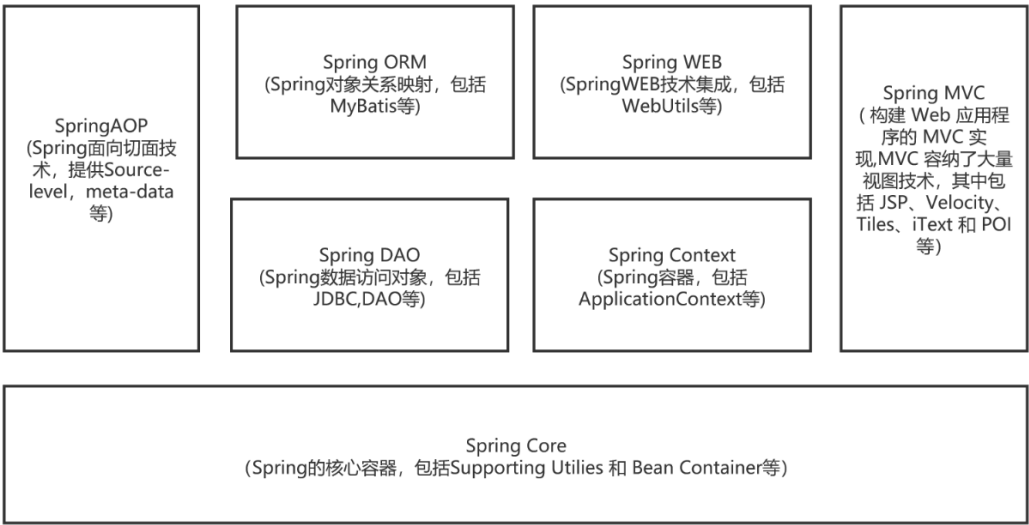


图 2.3.1 Spring 框架的七大模块以及其常见组件

以上 Spring 所有的模块都可以独立的进行使用，也可以与上图中提到的其他组件共同使用，以达到所需求的目的，下面将对 Spring 中核心的两个模块，Spring Core（核心容器）与 Spring AOP（面向切面编程）进行说明。

Spring Core 是 Spring 框架核心的内容，同时也是基础的内容。Bean Factory 作为 Spring Core 的核心组件之一，体现了工厂模式的最大特点和优势。Bean Factory 利用 IoC（控制反转，Inversion of Control）极大地降低了软件的耦合度，降低了代码开发工作量<sup>[6]</sup>。

Spring AOP 机制可以将软件开发者在业务流程中的共享功能独立地提取出来，生成独立的功能代码。在业务流程执行的过程之中，Spring 框架会根据业务流程不同要求，自动将独立的功能代码切入到业务流程的正确位置。因为 Spring AOP 在 Spring 中使用的是声明性的事务管理，仅需要在配置文件中进行关键代码的配置，就可以实现事务管理的功能<sup>[6]</sup>。

### 2.3.2 SpringMVC 模式

SpringMVC 的本质是一个 servlet。其中的核心是 DispatcherServlet。它是以请求为驱动的，系统接收请求会将请求发送到控制器，然后通过相对应的模型对象使用分派器来展示请求结果的视图。

SpringMVC 的工作原理如图 2.3.2 所示。

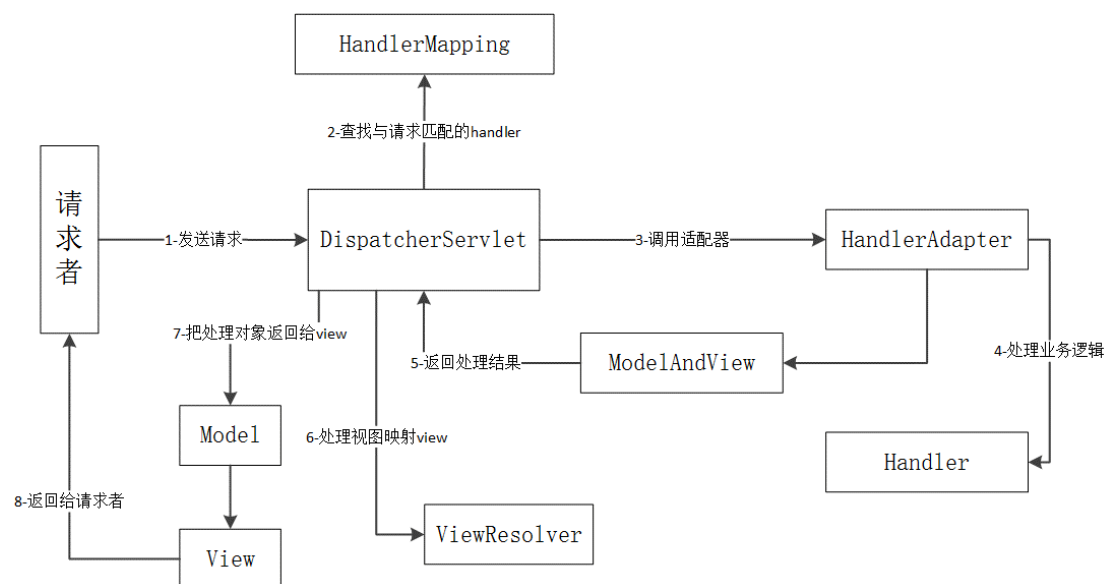


图 2.3.2 SpringMVC 的工作原理

通过上图，我们可以了解 SpringMVC 的工作原理，相比于其他的设计模式来说，SpringMVC 的代码是封装的，具有低耦合性的特点。同时在 MVC 模式中，各司其职。如果一旦某个层的功能需求发生了一定的变化，需要进行代码的编辑，仅仅需要更改相应层中的代码即可，其他区域的代码不会受到影响。

对于开发人员来说，MVC 模式实现了按层将系统分开，前端开发人员可以进行开发视图层中的 JSP 页面，后台开发人员可以专门开发业务层与控制层。

### 2.3.3 MyBatis 框架

MyBatis 是 Apache 软件基金会旗下的第一个开源项目 iBatis。iBatis 框架在 2010 年项目迁移到 google code 之后更名为 MyBatis。MyBatis 是当前非常流行

的 ORM（Object Relational Mapping）框架，其功能强大，实现也相对简单，优雅<sup>[7]</sup>。

下图介绍了 MyBatis 的工作流程。

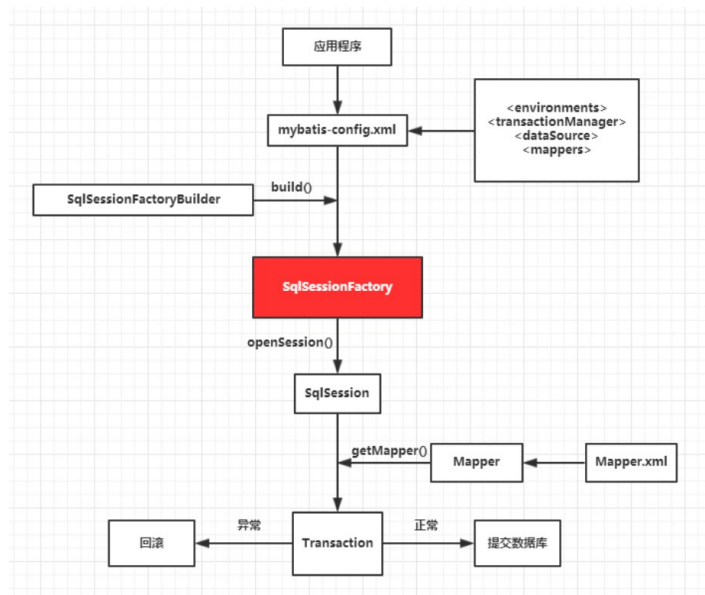


图 2.3.3 MyBatis 的工作流程

由此很容易看出，MyBatis 框架同 JDBC 相比，可以有效减少代码量。且作为一款持久化的框架，其结构小巧简单。最重要的是，MyBatis 支持编写动态 SQL 语句。

## 2.4 MySQL 数据库

MySQL 的起源最早可以追溯到 1979 年，其前身最初是基于 BASIC 平台的报表工具。经过 40 余年的发展，软件先后被 Sun 公司, Oracle 公司收购。目前经过多次关键性的技术更新，软件已更新到 8.0 版本，是世界上使用最广泛的几种数据库之一。

目前在数据库市场上，MySQL 数据库并不是唯一被广泛应用的，该数据库的竞品还是有很多的。然而，MySQL 是目前较为常用的数据库。与其他数据库进行不同方面的综合比较，MySQL 数据库具有以下优点：

首先，MySQL 服务器性能相对其他的服务器来说，性能较高，稳定性较强，故出现宕机等异常情况的概率较低。其次，MySQL 数据库作为开源的软件，没有版权的限制和限制，自主性强，使用成本相对于 Oracle 数据库或者其他竞品来说更低。

其次，MySQL 数据库软件的体积较小，安装与维护仅需根据文档说明就可

以解决大部分问题<sup>[8]</sup>。

最重要的是，MySQL 数据库支持多种不同的操作系统。可支持多种不同的开发语言。

## 2.5 AJAX 技术

AJAX 技术，即 Asynchronous JavaScript And XML 的简称。Ajax 是一种广泛应用于 web 端的技术，而并非单独的一种编程语言。在本系统中，Ajax 是通过 JavaScript 在 web 浏览器和 web 服务器之间用来发送或接收数据。

使用 AJAX 技术可以实现不重新加载整个网页就可快速更新部分网页中部分与后台相关数据的技术，AJAX 技术通过在后台与 Web 服务器交换数据，使得用户不必要每次提交请求时刷新整个页面，直接起到了快速响应的作用。通过使用 AJAX 技术，可以使 web 应用功能更加强大，达到了更轻巧迅速，更简便应用的特点。

## 第 3 章 系统设计

作为运维管理系统而言，其主要的面向对象是企业内的员工与管理者，运维管理系统涉及到企业内的许多部门。在正式进行设计开发系统之前，对企业运维管理进行分析和综合研判是极为重要的。开发本系统的目的是实现运维管理的信息化，作用是提高人事部门的工作效率并提高公司信息化程度。本章节将主要介绍本系统的需求分析、系统功能分析与数据库的设计<sup>[13]</sup>。

### 3.1 需求分析

本系统是基于 SSM 框架的运维管理系统，通过此系统，企业内员工可以通过浏览器访问系统来进行个人信息的浏览、修改功能以及日常打卡，查看考勤状况，也可以查看企业对员工的培训计划，并可以查看企业内全部岗位以及公告栏和下载文件功能。管理人员可以查看，修改或编辑员工个人信息，包括基本信息，个人合同和薪酬，也可以针对企业内员工进行培训，并上传资料，并且添加了招聘功能，提供一个面向非注册员工的招聘通道，管理员可查看非注册员工投递的资料，并予以审核。

#### 3.1.1 系统需求分析

运维管理系统的成功与否取决于使用者体验的好坏，对于大部分系统来说，要做界面友好，响应速度有保障，权限分明以及可视化是极为重要的。本小节将对运维管理系统进行需求分析，并通过需求分析，做到如下几点。

（1）系统操作界面友好，操作逻辑清晰，网页加载速度快。

作为运维管理系统，其适用对象主要是企业内办公人员以及企业内的全部员工，友好的界面有助于更多方便快捷的找到所需使用的功能或查找所需要的文件。只有当员工和管理员快速的上手使用本系统，才会有更好的体验，并可降低培训使用新系统的成本。且网页的加载速度也十分重要，根据研究发现，当网页在 5 秒后还未加载或在刷新后仍无法显示后，人们会主动关掉网页<sup>[9]</sup>。所以，本系统对网页速度访问的要求为：在用户数量达到一个较大值后仍可对网站内容进行访问，并在不卡顿的情况下完成具体的操作。

（2）权限分明。

运维管理系统的使用者主要分为两个部分：员工和企业管理者，其中员工面对的主要是企业内的普通员工和投递简历的非注册员工，企业管理者主要面向的是人事部门人员以及企业的管理者。其中，为了保证员工信息的真实性，只允许

在管理员端添加员工基础信息，且对基本信息的修改做出了限制，如：身份证，用户名等。为了防止员工恶意修改企业内部资料，管理员可随时删除用户发布的文档以及上传的文件。管理员可以对员工基本信息、企业内职位信息、企业内部岗位的信息，公告和下载中心、员工考勤和请假状况等进行修改。

### （3）可视化。

作为运维管理系统，有着优秀的可视化界面是极为重要的，在员工或管理者登入系统后，就可以快速的了解全部功能，方便用户或管理者执行所需的任务。其次本系统设有多级导航栏，导航栏的布局位置合理，功能清晰，方便用户或管理者及时查找到所需要的功能，有助于更加高效的进行后续的操作。最后在导航栏添加了数据可视化模块，使用数据可视化工具 ECharts 实现员工基本信息可视化查看。

#### 3.1.2 员工端需求分析

运维管理系统的总目标有如下两点：实现有效的运维管理、实现企业内部信息与数据的共享。企业内员工使用运维管理系统员工端可以满足其获取企业内部信息的要求。

经过对企业内员工和管理者的调查以及参考国内外其他平台的运维管理系统发现，运维管理系统大多是按照如下的逻辑进行运作的：员工在注册后，才可进入系统，进入系统后，首先员工进行上班打卡，员工可以对个人信息进行管理，并可查看企业内部部门、职位、管理员上传的文档、个人考勤状态，个人合同与薪酬以及培训管理，并实现数据可视化。

运维管理系统员工端的需求如图 3.1.2 所示。



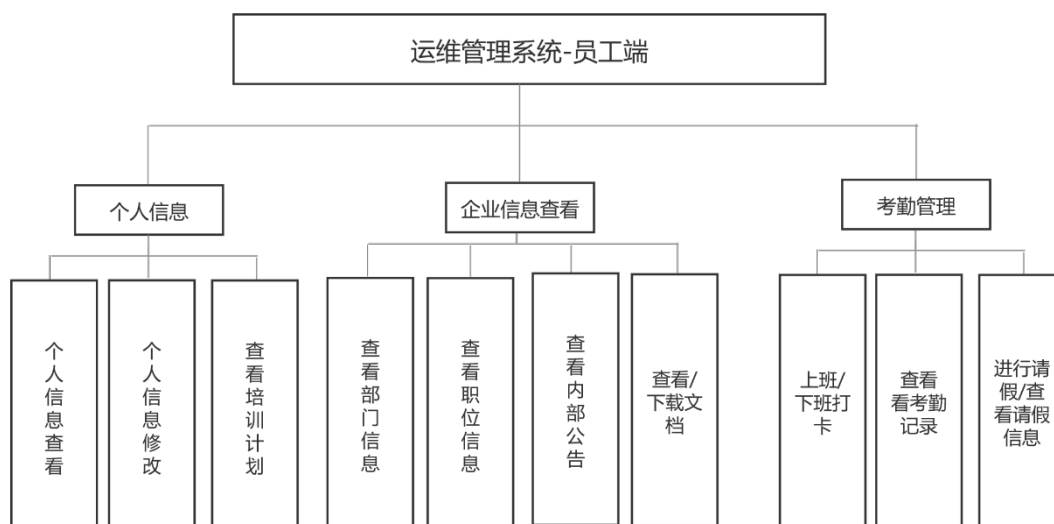


图 3.1.2 运维管理系统员工端需求图

由图 3.1.2 可知，运维管理系统员工端的内容主要包括个人信息、企业信息查看、考勤管理几个模块。其中个人信息包括查看个人信息、对个人信息的编辑（姓名，身份证号等关键信息更改权限受限）以及查看培训计划。企业信息查看包括查看管理员更新的部门信息与职位信息、查看由管理员发布的内部公告、查看或下载管理员上传的文档。

### 3.1.3 管理员端需求分析

运维管理系统除了面向企业内部员工之外，还应能对企业内部的各种信息进行综合管理，所以运维管理系统的管理员端是必不可少的一部分。运维管理系统的管理员端由一下几大部分所构成，它们分别是：用户管理（全部用户，包括其他管理员）、企业信息管理、考勤管理、培训管理与招聘管理。

运维管理系统管理员端的需求如图 3.1.3 所示。

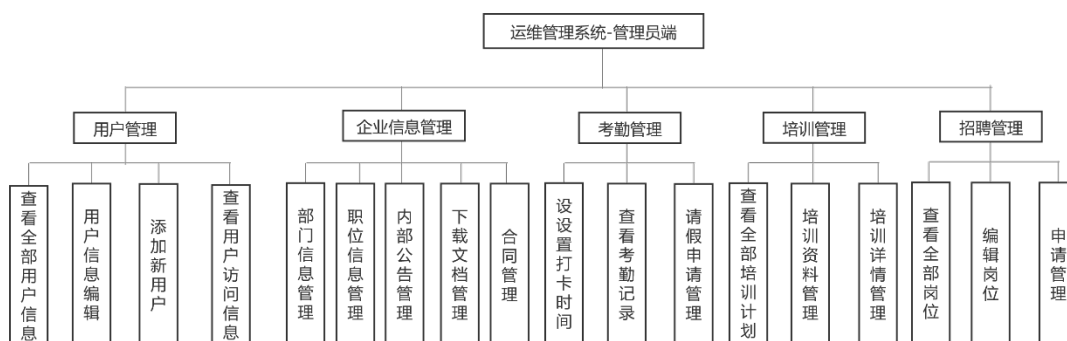


图 3.1.3 运维管理系统管理员端需求图

由图 3.1.3 运维管理系统管理员端需求图可以看出，运维管理系统管理员端

的功能相较于员工端来说添加了许多，主要是可以对企业内部信息进行编辑和修改，并添加了招聘管理的功能，达到了管理企业内部信息的作用。本运维管理系统后台主要包括以下几个大模块，分别是用户管理、企业信息管理、考勤管理、培训管理以及招聘管理。

(1) 用户管理是对已完成注册用户的基本信息进行管理。管理员登入系统后，可在右侧导航栏中点击用户管理部分。用户管理部分可查看用户的访问信息，并可对用户信息进行查看与修改。

(2) 企业信息管理包括五大部分，分别是：部门信息管理、职位信息管理、内部公告管理、下载文档管理与合同管理。管理员在登入系统后，在导航栏对应位置可选择上述介绍的功能操作，进行信息的查询与修改。

(3) 考勤管理包括修改打卡时间、查看考勤记录与请假申请管理功能。打卡时间修改后，员工端上班打卡与下班打卡的考勤时间点也随之变化，上班迟到与下班早退的具体判断由后台控制器进行判断。查看考勤记录是根据数据库中打卡时间与天数进行综合判断得出的，管理员可直接通过后台页面进行编辑。请假管理功能实现了管理员可查看由员工发出的请假申请。管理员在收到员工请假申请后，需进行审批。

(4) 培训管理的主要功能为查看当前已发布的培训计划、制定新的培训计划并管理培训资料、对培训详情进行管理等功能。培训管理主要针对注册员工和管理员，管理员可上传新的培训资料或管理当前数据库中已有的培训资料，并可添加员工培训记录，以及对员工培训进行量化考核。

(5) 公告管理功能是方便员工和管理员查看公告的工具，内容由管理员发布，管理员在登录后可新建公告或删除已有公告以及更改原有公告的内容。

(6) 招聘管理主要面向的对象是未注册员工与管理员，未注册员工可在首页顶部导航栏中的招聘页面中查看岗位信息，并选择投递简历。管理员可对岗位信息进行管理，包括修改岗位，招聘人数和投递通道开启与否等。管理员也可查看全部投递的信息，并对投递的信息进行判断，决定是否通过该员工的职位申请。

总而言之，运维管理系统需要满足企业中的普通员工和管理者的需求，并且需要做到快速响应。这样才能完成一个功能完备，界面友好的运维管理系统。

### 3.2 系统功能分析

针对使用本系统的不同用户，进行系统功能的分析，有助于为后期软件设计

提供良好的框架，节约开发时间，提高开发效率<sup>[10]</sup>。本模块的作用是通过分析使用本系统的不同用户得出具体的分析结论。

### 3.2.1 未注册用户角度分析

为保证系统内部资料的安全，未注册的用户可使用的功能有限，仅可查由管理员发布的招聘信息并投递简历。未注册员工可投递本公司的多个职位，由管理人员进行审核批准。

本系统中未注册普通用户用例图如图 3.2.1 所示。

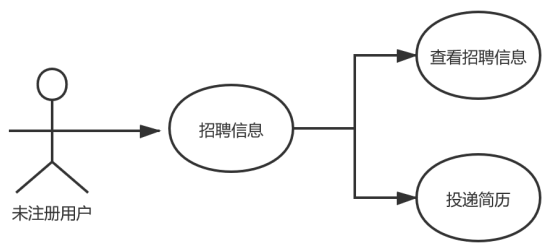


图 3.2.1 未注册用户用例图

### 3.2.2 员工角度分析

运维管理系统主要面向的对象是企业内的员工，相对于未注册员工，权限较高，可以查看企业内部信息并下载一些文档。

本系统中员工的用例图如图 3.2.2 所示。

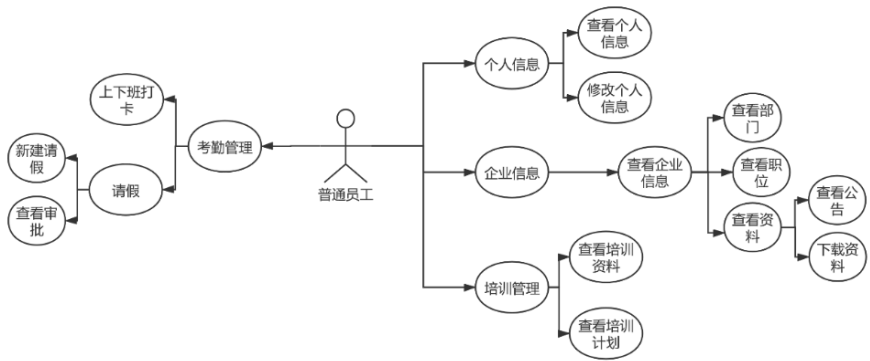


图 3.2.2 员工用例图

#### (1) 考勤管理

员工在登录页面输入正确的登录信息并完成验证码验证后，系统通过调用登录相关控制器代码实现自动判断用户信息。信息验证通过后，进入运维管理系统员工端。

在主页-我的桌面中，在页面上部可找到上下班考勤打卡按钮。员工需在管理

员设置的时间段内执行上班打卡与下班发卡的。操作。若上下班打卡时间点出现异常，即出现了上班迟到或下班早退的情况，员工需在考勤管理中添加当日考勤异常的原因。

员工请假需要在左侧导航栏个人考勤中点击我要请假添加请假信息，需要按要求填写开始日期、结束日期、请假类型以及内容。提交完成后，由管理员进行审核，可在我的请假中找到对应的请假审批结果。

## （2）个人信息

员工登录后，可在左侧导航栏中用户管理查看个人信息，并可修改本人的登录名，用户名以及密码。由于员工权限有限，员工个人信息的更新由管理员进行编辑。

## （3）企业信息

员工登录后，通过左侧导航栏点击部门管理、职位管理、公告管理与下载中心可查看由管理员发布的企业内部的信息。

员工仅可查看这些信息，并可下载来自下载中心由管理员上传的文档。

## （4）培训管理

员工登录后，在左侧导航栏中点击培训管理，可查看培训计划、我的培训以及培训资料。

员工可查看由管理员制定的培训计划、已完成的培训的各项基本信息，查看全部的培训资料，并可实现网页内播放视频以及下载全部培训资料。

### 3.2.3 管理员角度分析

运维管理系统中的管理员拥有最高权限，在验证登录信息正确后，可对本系统中全部的数据进行管理与编辑，管理员的用例图如图 3.2.3 所示。

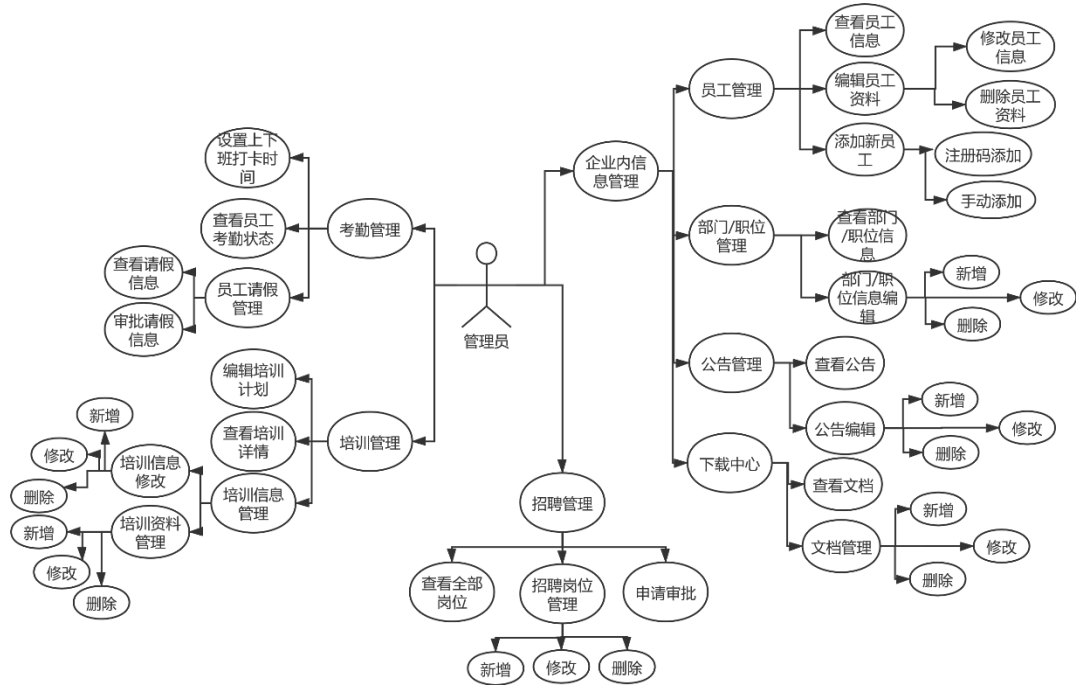


图 3.2.3 管理员用例图

#### (1) 管理员考勤管理

管理员进入系统后，点击左侧考勤管理，可设置员工上下班考勤打卡时间点，查看并编辑全部员工考勤记录，以及查看员工提交的请假申请。除此之外还可编辑员工考勤记录，包括上班、下班天数、迟到次数和请假次数。

管理员可查看员工提交的请假申请，并可编辑员工请假天数和审批状态。

#### (2) 管理员培训管理

管理员进入系统后，点击左侧导航栏的培训管理，进入培训管理页面。管理员可从培训管理导航栏子菜单选择下述的功能：培训计划、培训详情、添加培训信息与培训资料管理功能。

培训计划变更需修改后台 JSP 页面，前端显示无法进行更新操作。培训详情可查看由管理员添加的、员工已完成的培训计划。

培训信息管理包括了添加培训信息与培训资料管理，添加培训信息需提供姓名、培训内容、开始与结束日期、完成情况与成绩评定才可提交。修改培训信息需在培训详情中进行修改。培训资料管理可添加新的培训资料，上传的资料面向

全部登录用户，即登录之后便可进行查看或下载。管理员还具有编辑已上传的培训资料和上传新培训资料的功能。

#### （1） 管理员企业内信息管理

管理员作为企业内部信息的管理者，主要负责管理企业中员工信息、部门信息、公告以及可供下载的资料进行管理。

对企业内信息查看在导航栏在员工管理、职位/部门管理、公告管理和下载中心中的查看功能便可实现。

对信息的修改只需要在导航栏找到对应功能，对信息的修改包括了员工信息的增删改、职位与部门的增删改、公告的增删改与下载文件的增删改，仅需根据页面提示进行编辑即可，过程与培训管理类似，在此不做赘述。

#### （2） 管理员招聘管理

管理员另一个重要作用是负责整理上传人事部门提交的新招聘信息，并及时更新企业内的招聘信息与招聘的具体数据，以保证招聘信息的有效性与准确性。

管理员还需审核提交未注册员工投递的简历，筛选出合格简历，及时响应应聘者反馈信息，保证应聘者信息在短时间内得到有效处理。

管理员招聘管理仅需点击左侧导航栏的招聘管理，对应功能有岗位公布与申请概况。岗位公布主要负责上传新的岗位，并及时更新招聘信息，申请概况可查看非注册应聘者投递的简历或下载其上传的附件简历，并予以审核。

### 3.3 数据库设计

数据库作为运维管理系统不可或缺的一部分，主要起到了存储系统数据的作用，数据库结构设计的好坏直接会直接影响到整个运维管理系统的运行效率。所以，设计一个结构合理，逻辑关系清晰的数据库有助于提高系统处理 SQL 语句查询的执行效率。同时也有效地保证了数据的一致性与完整性。

#### 3.3.1 数据库概念结构设计

数据库概念结构设计采用 E-R 图。E-R 图，是由 P. P. S. Chen 于 1976 年提出的理论。其主要功能是描述实体、属性和实体之间的联系等。实体之间的联系有以下三种类型：1：1（一对一联系）、1：M（一对多联系）、M:N 联系（多对多联系）。

本系统基础数据主要包括员工个人信息、管理员个人信息、岗位信息、部门信息、公告、下载文档、考勤记录、培训信息、可视化数据等与员工和管理员相

关的信息实体。下图将展示运维管理系统的 E-R 图。

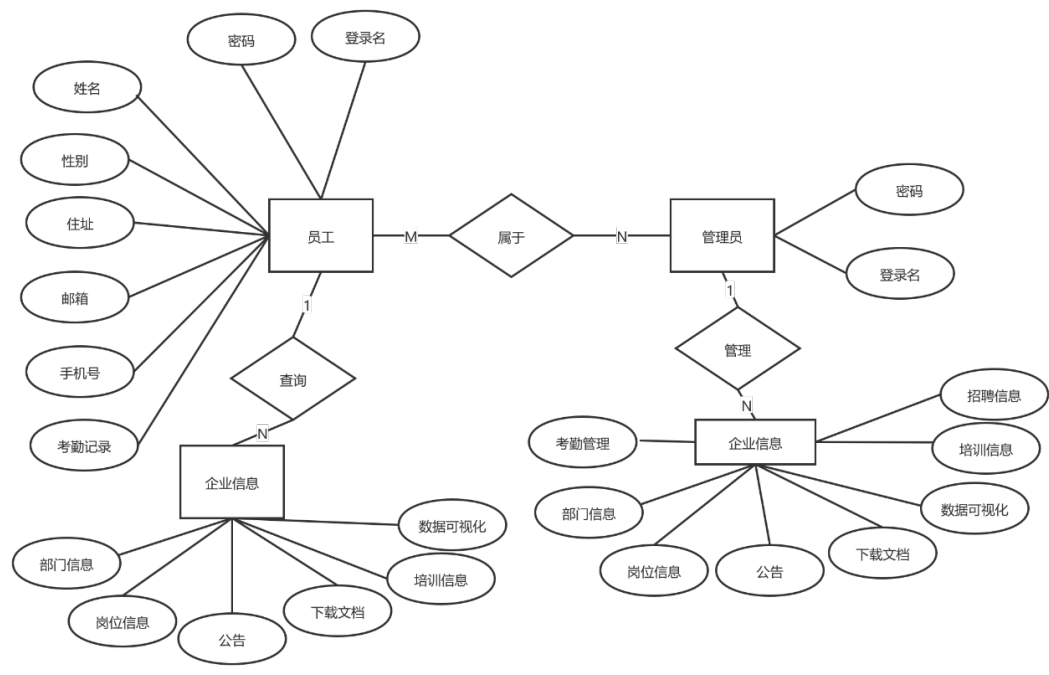


图 3.3.1 运维管理系统 E-R 图

3.3.2 数据库表结构设计

表 3.3.2.1 用户信息表 (user\_inf)

| 字段名        | 字段含义                                 | 字段类型         | 备注                   |
|------------|--------------------------------------|--------------|----------------------|
| ID         | 用户 id                                | int (11)     | PK, NOT NULL, UNIQUE |
| loginname  | 登录名                                  | varchar(20)  | NOT NULL, UNIQUE     |
| password   | 密码                                   | varchar(255) | NOT NULL, UNIQUE     |
| email      | 邮箱                                   | varchar(255) | NOT NULL             |
| status_id  | 员工状态                                 | int (11)     | NOT NULL             |
| createdate | 创建日期                                 | TIMESTAMP    | TIMESTAMP NOT NULL   |
| username   | 用户名                                  | varchar(20)  | NOT NULL             |
| emp_id     | 职工表外键<br>employee_inf                | int (11)     | FK, NOT NULL         |
| che_id     | 考勤表外键<br>checkwork_inf               | int (11)     | FK, NOT NULL         |
| con_id     | 签约表外键<br>confidentialitycontract_inf | int (11)     | FK, NOT NULL         |
| sal_id     | 工资表外键<br>salary_inf                  | int (11)     | FK, NOT NULL         |

表 3.3.2.2 职工信息表 (employee\_inf)

| 字段名          | 字段含义                 | 字段类型        | 备注                   |
|--------------|----------------------|-------------|----------------------|
| ID           | 用户 id                | int (11)    | PK, NOT NULL, UNIQUE |
| dept_id      | 部门 ID<br>外键 dept_inf | int (11)    | FK, NOT NULL         |
| job_id       | 职位 ID<br>外键 job_inf  | int (11)    | FK, NOT NULL         |
| name         | 邮箱                   | varchar(20) | NOT NULL             |
| card_id      | 员工状态                 | varchar(20) | NOT NULL             |
| address      | 住址                   | varchar(50) | NOT NULL             |
| username     | 用户名                  | varchar(20) | NOT NULL             |
| sex_id       | 性别                   | int (11)    | NOT NULL             |
| education_id | 教育程度                 | int (11)    | NOT NULL             |
| createdate   | 创建日期                 | TIMESTAMP   | TIMESTAMP NOT NULL   |
| user_id      | 用户 ID<br>外键 user_inf | int (11)    | FK, NOT NULL, UNIQUE |

表 3.3.2.3 访问信息表 (uservisit\_inf)

| 字段名          | 字段含义     | 字段类型         | 备注                                   |
|--------------|----------|--------------|--------------------------------------|
| ID           | 访问顺序 id  | int (11)     | PK, NOT NULL, UNIQUE, auto_increment |
| login_time   | 登入时间     | varchar(255) | NOT NULL                             |
| exit_time    | 退出时间     | varchar(255) | NOT NULL                             |
| visit_ip     | 访问 IP 地址 | varchar(255) | NOT NULL                             |
| user_address | 用户所在地    | varchar(255) | NOT NULL                             |
| browser      | 所用浏览器    | varchar(255) | NOT NULL                             |
| system       | 操作系统版本   | varchar(255) | NOT NULL                             |
| version      | 软件版本     | varchar(255) | NOT NULL                             |
| loginname    | 登录名      | varchar(255) | NOT NULL                             |

表 3.3.2.4 培训信息表 (train\_inf)

| 字段名     | 字段含义                  | 字段类型     | 备注                                   |
|---------|-----------------------|----------|--------------------------------------|
| ID      | 培训顺序 ID               | int (11) | PK, NOT NULL, UNIQUE, auto_increment |
| emp_id  | 职工表外键<br>employee_inf | int (11) | FK, NOT NULL                         |
| dept_id | 部门 ID<br>外键 dept_inf  | int (11) | FK, NOT NULL                         |
| job_id  | 职位 ID, 外键 job_inf     | int (11) | FK, NOT NULL                         |



| 续表 3.3.2.4  |        |              |                       |
|-------------|--------|--------------|-----------------------|
| startdate   | 开始培训日  | varchar(255) | NOT NULL              |
| enddate     | 结束培训日  | varchar(255) | NOT NULL              |
| totallength | 总培训时间  | int (11)     | NOT NULL              |
| completion  | 完成度    | int (11)     | NOT NULL              |
| grade       | 成绩     | int (11)     | NOT NULL              |
| traintime   | 培训添加时间 | TIMESTAMP    | TIMESTAMP<br>NOT NULL |

表 3.3.2.5 公告信息表 (notice\_inf)

| 字段名        | 字段含义                 | 字段类型        | 备注                                   |
|------------|----------------------|-------------|--------------------------------------|
| ID         | 公告顺序 id              | int (11)    | PK, NOT NULL, UNIQUE, auto_increment |
| title      | 标题                   | varchar(50) | NOT NULL                             |
| content    | 公告内容                 | text        | NOT NULL                             |
| createdate | 创建日期                 | TIMESTAMP   | TIMESTAMP<br>NOT NULL                |
| user_id    | 用户 ID<br>外键 user_inf | int (11)    | FK, NOT NULL, UNIQUE                 |

表 3.3.2.6 文档下载信息表 (document\_inf)

| 字段名        | 字段含义                 | 字段类型         | 备注                                   |
|------------|----------------------|--------------|--------------------------------------|
| ID         | 下载文档顺序 id            | int (11)     | PK, NOT NULL, UNIQUE, auto_increment |
| title      | 标题                   | varchar(50)  | NOT NULL                             |
| filename   | 文档名                  | varchar(300) | NOT NULL                             |
| remark     | 备注                   | varchar(300) | NOT NULL                             |
| createdate | 创建日期                 | TIMESTAMP    | TIMESTAMP<br>NOT NULL                |
| user_id    | 用户 ID<br>外键 user_inf | int (11)     | FK, NOT NULL, UNIQUE                 |

表 3.3.2.7 公告信息表 (notice\_inf)

| 字段名       | 字段含义    | 字段类型         | 备注                                   |
|-----------|---------|--------------|--------------------------------------|
| ID        | 岗位顺序 ID | int (11)     | PK, NOT NULL, UNIQUE, auto_increment |
| jobtype   | 职位信息    | int (11)     | NOT NULL                             |
| content   | 职位介绍    | varchar(255) | NOT NULL                             |
| peoplenum | 招聘人数    | int (11)     | NOT NULL                             |
| Status    | 招聘状态    | int (11)     | NOT NULL                             |
| enddate   | 结束日期    | varchar(255) | NOT NULL                             |

## 第4章 系统详细设计和实现

本章的主要内容是对系统之中的各个功能模块进行详细设计与代码实现，主要对用户登录与新用户注册、考勤管理、员工管理、公告管理、下载中心、数据可视化中的所使用到的技术与重点实现方法进行了描述，并添加了对应各个不同模块的流程图与关键代码的描述。

### 4.1 用户登录与新用户注册模块

用户登录模块主要面向的使用者为已注册的员工与管理员。新用户注册模块的面向对象为未注册的用户，用户账号可由管理员在后台自行注册。为防止用户使用自动化脚本进行恶意注册，本系统注册采用注册码加管理员直接注册的方式进行新用户注册。验证码由管理员申请获得，每天有次数限制。若未注册用户想自行进行员工身份的注册，则必须获得管理员提供的注册码，验证码注册后失效。

#### 4.1.1 注册功能实现

未注册用户点击登录框左下方的“没有账号？立即注册”，系统通过视图解析器跳转到对应注册码填写界面。未注册用户需获取有管理员提供的注册码（管理员以管理员身份登入系统后，在顶部导航栏处点击获取注册码，获取的注册码可供未注册用户使用），同时管理员生成的注册码会保存在数据库中的 `code_inf` 中，以供对照。

注册码是由 `controller` 层中 `RegistRandomCode`、`RegistController` 与 `RegistCheckController` 类完成的，其中 `RegistRandomCode` 类功能为生成一个 9 位数的随机字符串，`RegistController` 为控制器模块实现，主要负责注册码验证功能，`RegistCheckController` 主要负责实现注册信息验证，并使用 `Ajax` 进行前端校验。

下为 `Ajax` 前端校验重要代码。

```
var registCode=document.getElementById("registCode").value;
//使用 Ajax 先行进行前端校验，是封装在 jQuery 中的，触发 onclick 事件后，执行本校验函数功能。
```

下为 `RegistCheckController` 的主要业务代码：

```
private AhualyService ahualyservice;//接口实现类对象
@RequestMapping ("/toCreateCode")//管理员点击，通过本请求实现生成验证码

    Public    Map    <String, String>    toCreateCode    (HttpServletRequest
```

response, HttpSession session)//验证码生成

@RequestMapping("/toregistCode")//将提交注册码上传至 toregistCode  
拦截器判断是否正确

下为 RegistCheckController 中主要业务代码:

@RequestMapping(value="checkRegistCode", method = RequestMethod. POST)  
//实现获取 ajax 封装的数据, 并进行数据比对

@RequestMapping(value="check\_Register\_loginname", method = RequestMethod. POST)//进行用户名, 登录名数据比对验证, 其他验证如:  
邮箱、密码等原理类似, 不再赘述。

具体的注册功能逻辑流程图如图 4. 1. 1 (1) 所示。

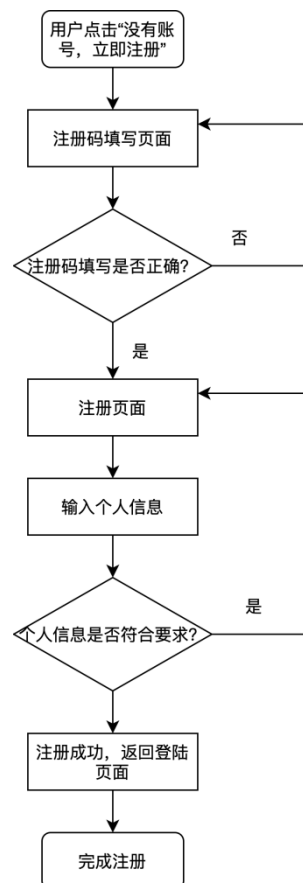


图 4. 1. 1 (1) 注册功能逻辑流程图

注册部分对应界面展示如下所示。



图 4.1.1 (2) 验证码验证



图 4.1.1 (3) 验证码验证-2



图 4.1.1 (4) 验证码验证-3



图 4.1.1 (6) 员工注册实现-1

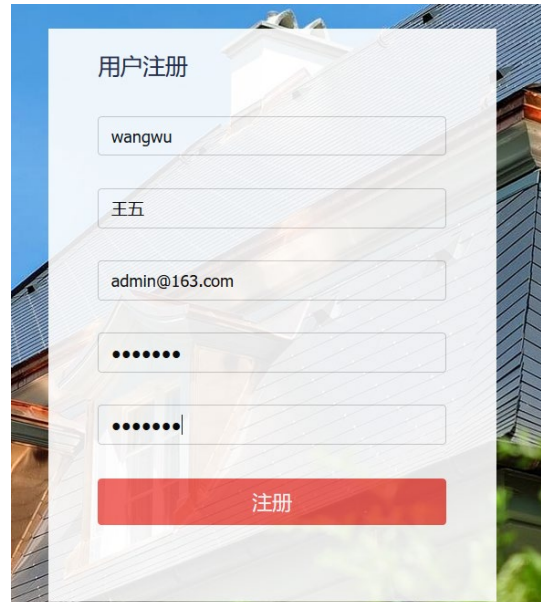


图 4.1.1 (7) 员工注册实现

## 4.1.2 登录功能实现

已注册的用户登录需要输入正确的登录信息，包括用户名和密码。为防止简单的脚本注入攻击，特别添加验证码功能，只有用户名密码验证码全部正确才可

登入系统。登录功能的实现主要基于登陆拦截器与验证码生成器进行实现的。同时在登录验证时使用了 session 技术，如果为获取到成功登入系统的 session，则默认无法访问，且登录成功后，会把从 HTTP 请求头中获取的数据（浏览器信息，操作系统，访问时间）获取并单独存放在数据库中 user\_visit 表中。

登录功能主要使用到的类为 controller 层中的 LoginCheckController 与 VerificationCodeController 实现的。其中 LoginCheckController 主要负责登录信息验证，与前端 Ajax 结合使用，VerificationCodeController 为验证码生成功能，负责生成验证码。

具体登录功能逻辑流程图如下图所示。

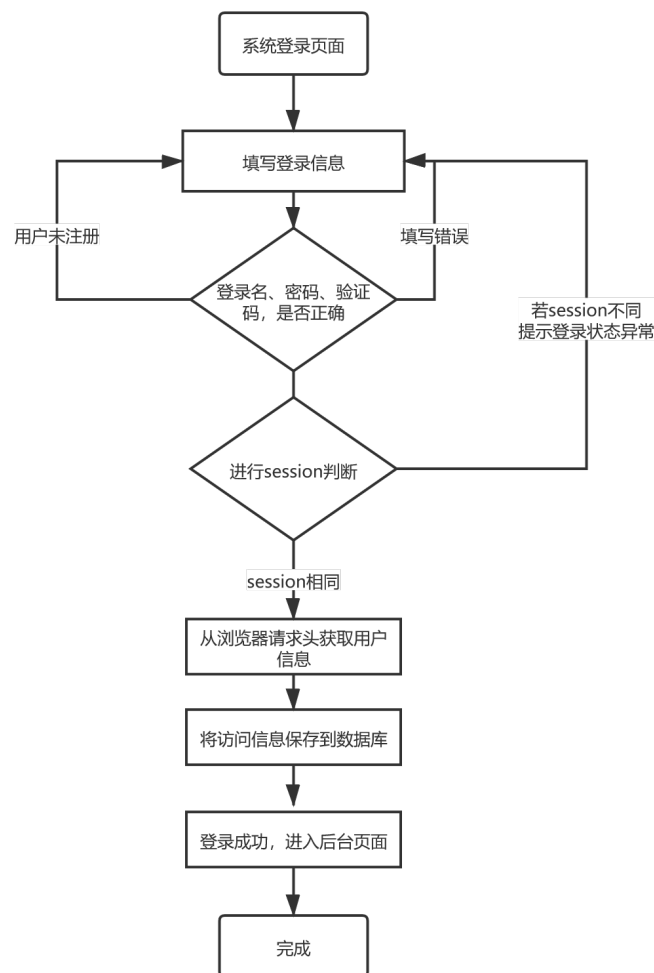


图 4.1.2 (1) 登录功能逻辑流程图

用户登录部分对应界面展示如下所示。



图 4. 1. 2(2) 登录验证功能



图 4. 1. 2(3) 用户登录成功后台主界面



图 4. 1. 2(3) 用户登录成功后台主界面

### 4. 1. 3 忘记密码功能实现

忘记密码功能主要面向的使用对象是用户，管理员修改密码需要借用其他管理员账户登入后进行修改。因密码采用 MD5 加盐加密算法进行加密，从数据库

端查看到的密码每次都不是相同的，故无法通过数据库直接获取密码。

忘记密码功能主要使用到的类为 controller 层下的 FindPasswordCheckController 类与 FindPasswordController 类，其中 FindPasswordCheckController 主要负责找回密码前的验证，验证用户名以及登录名是否匹配，同时使用了 utils.common 下的 ShiroMD5Privacy 类进行加盐加密，保证了系统的安全性。

下为 FindPasswordCheckController 类的主要业务代码：

```
@RequestMapping("checkLoginName")//密码找回前的用户名验证。
```

```
String loginname = request.getParameter("loginname");//用户名比对
```

```
@RequestMapping("checkUsername")//用户名与密码匹配验证
```

```
User user = ahualyservice.findUserByLogin(loginname);//获取登录名。
```

```
String username2=user.getUsername();//获取用户名，后续代码为进行判断。
```

下为 FindPasswordController 类中的主要业务代码：

```
if(!user_input_verifyCode.toLowerCase().equals(session.getAttribute("code")))//验证码验证。
```

```
ahualyservice.toUpdatePassword(user.getLoginname(), ShiroMD5Privacy.privacy(user.getLoginname(), user.getPassword()));//调用加盐加密算法，将更新后密码加密。
```

下为 ShiroMD5Privacy 类中加密算法实现的主要代码：

```
String algorithmName = "MD5";//确定加密类型为 MD5。
```

```
Object source = password;//加密对象为密码。
```

```
Object salt = ByteSource.Util.bytes(loginname);//定义加盐数据类型
```

```
int hashIterations = 100; //加密次数，此处为加密 100 次，可修改为任意次数。
```

```
Object privacy = new SimpleHash(algorithmName, source, salt, hashIterations);//执行加密过程，拼接得出最终加盐加密的密码
```

忘记密码具体业务逻辑流程如下图所示。

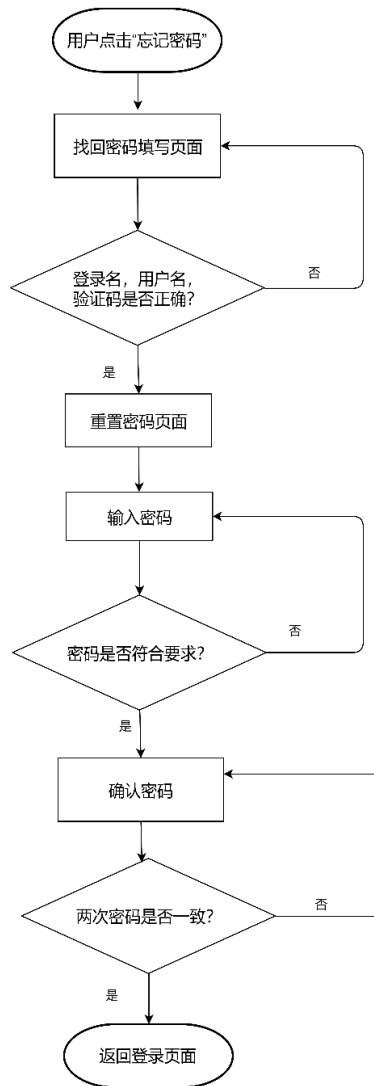


图 4.1.3 (1) 忘记密码业务逻辑流程图

忘记密码部分对应界面展示如下所示。



图 4.1.3 (2) 找回密码实现-1



图 4.1.3 (3) 找回密码实现-2





图 4.1.3 (3) 找回密码实现-3



图 4.1.3 (4) 重置密码

## 4.2 用户管理模块

用户管理模块主要适用对象为员工与管理员，具体需求详见第三章需求分析对应的用户管理部分。用户管理模块主要可查看登陆过的用户信息与编辑员工基本资料。

用户管理模块主要使用到的类有 controller 层的 UserController 类与 UserLoginController 类，其中 UserController 主要负责实现员工信息的增删改查以及管理员直接添加新用户功能，UserLoginController 类提供了包括 IP 地址（因本地访问，默认 IP 地址均为 0.0.0.0），登录设备，访问时间等，并将上述信息存至数据库，并可实现编辑操作。

下为 UserController 类中关键代码。

```
@RequestMapping(value="/user/list", method=RequestMethod.GET) // 查询所有用户显示用户列表
```

```
PageModel pageModel = new PageModel(); // 分页功能应用
```

```
if(pageIndex != null) {...} // 进行分页判断，省略具体判断流程
```

```
@RequestMapping(value="/user/edit", method=RequestMethod.POST) // 编辑信息操作
```

```
public String toedit(Integer status_id, Integer pageIndex, Model model, Integer id, @ModelAttribute User user) {...}
```

```
@RequestMapping(value="/user/add", method=RequestMethod.POST) // 管理员实现用户注册，具体进行登录名重复判断邮箱信息检测，session 验证，加盐加密算法应用详见 UserController 类中具体代码。除此之外，添加用户实现自动生成
```

外键 static\_id, 以作为关联用户的外键。

下为 UserLoginController 类中关键代码。

```
String pwd = ShiroMD5Privacy.privacy(loginname, password); //获取用户提交加密后的登录信息
```

```
user = ahualyservice.login(loginname, pwd);
```

```
if(user != null) {..} //进行密码判断, 以及获取响应头信息, 包括设备类型等内容。具体详见 UserLoginController 类中代码。
```

```
String urlInfo = "http://www.ip138.com/ips138.asp?ip="+ipAddress; //借助此网站查询 IP 地址, 因默认使用本地服务器, 此代码默认返回 0.0.0.0.
```

下图为用户管理模块的流程图(图 4.2.1)与管理员添加新用户(图 4.2.2)的流程图。

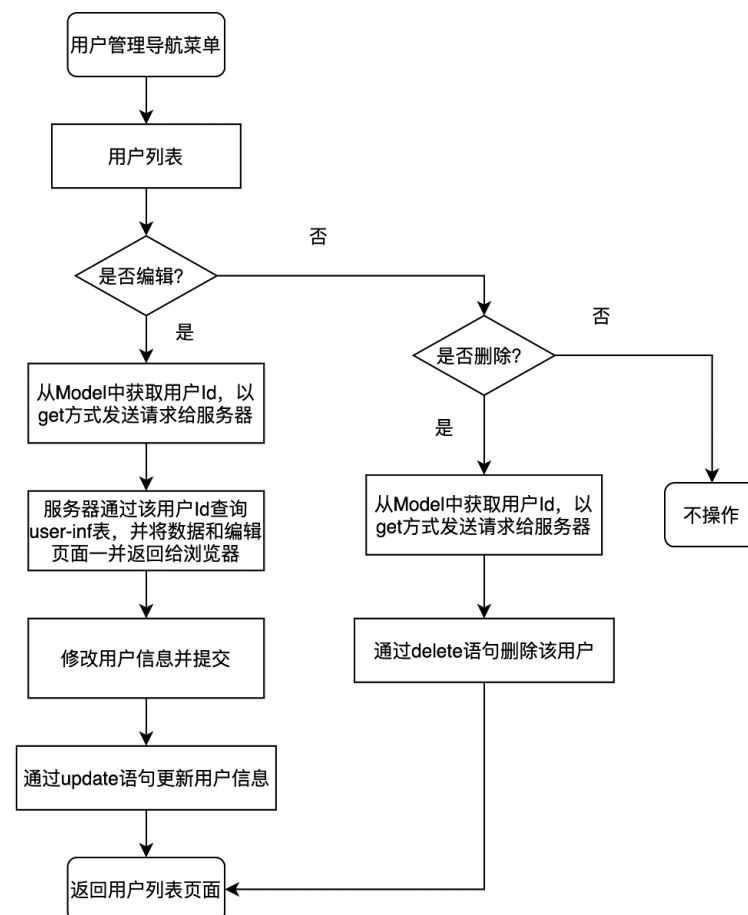


图 4.2.1 用户管理模块流程图

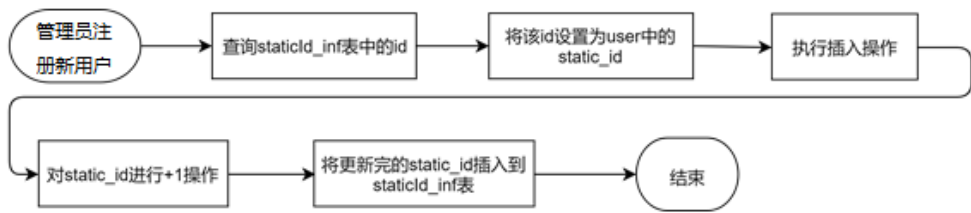


图 4.2.2 管理员添加新用户流程图

员工管理部分对应界面展示如下所示。

首页 / 访问用户信息

请输入查找内容

| ID | 登录用户       | 登录时间                | 退出时间                | 访客IP            | 用户地址 | 用户设备                        | 浏览器     | 操作系统    | 版本   | 操作 |
|----|------------|---------------------|---------------------|-----------------|------|-----------------------------|---------|---------|------|----|
| 83 | admin      | 2020-05-07 21:26:36 | 2020-05-07 21:29:01 |                 |      | Windows NT 10.0; Win64; x64 | Firefox | Windows | 76.0 | 管理 |
| 84 | admin      | 2020-05-07 21:31:58 | 2020-05-07 21:36:29 |                 |      | Windows NT 10.0; Win64; x64 | Firefox | Windows | 76.0 | 管理 |
| 85 | admin      | 2020-05-08 09:10:24 | 2020-05-08 09:13:42 | 0:1             |      | Windows NT 10.0; Win64; x64 | Firefox | Windows | 76.0 | 管理 |
| 86 | houjianwei | 2020-05-08 09:43:50 | 2020-05-08 09:53:53 | 0:0:0:0:0:0:0:1 |      | Windows NT 10.0; Win64; x64 | Firefox | Windows | 76.0 | 管理 |

图 4.2.3（1）管理员查看网站访问信息

我的桌面 添加用户 ×

\*登录名

\*用户名

\*邮箱

\*密码

\*确认密码

提交

图 4.2.3（2）管理员添加员工信息

请输入查找内容

| 登录名        | 用户名 | 状态  | 注册日期                | 操作 |
|------------|-----|-----|---------------------|----|
| manager    | 总经理 | 已审核 | 2020-04-11 13:52:15 | 编辑 |
| houjianwei | 侯建伟 | 已审核 | 2020-05-09 16:19:53 | 编辑 |
| zhangsan   | 张三  | 已审核 | 2020-05-10 10:17:59 | 编辑 |
| houdawei   | 侯大伟 | 未审核 | 2020-05-22 15:25:57 | 编辑 |

[上一页](#)
[下一页](#)
[跳转到](#)  [确定](#)

总共4条记录, 当前显示1-4条记录。

图 4.2.3（3）管理员审核员工信息

### 4.3 员工管理与企业信息管理模块

员工管理模块与企业信息管理模块(包括职位管理与部门管理),所使用的技术与原理大致相同主要是添加新的员工信息与企业信息,编辑已存在的员工信息与企业信息,删除以上信息的功能。其中,添加员工信息与企业信息是着重实现的部分,编辑与删除仅需操作对应数据库语句即可。

下面将重点介绍添加信息的功能,仅介绍添加员工,因其还涉及到一个外键问题,职位管理与部门管理不涉及此问题。首先需要手动管理用户信息,即实现用户 ID 与员工姓名绑定,当且仅当用户存在且违背关联,才可添加员工信息,因为在数据库建表时,已设置外键关联,无法做二次修改。其中主要用的类有 controller 层中的 EmployeeController 与 dao 层的 EmployeeDynaSqlProvider 和 EmployeeDao。

下为 EmployeeController 中的重要代码。

```
@RequestMapping (value="/ employee/list", method=RequestMethod. GET)//  
展示员工信息。
```

```
List<Employee> job_list =  
ahualyservice. get_EmployeeList(employee, pageModel) ;//将获取的员工信息存  
储到链表中, 后续过程为循环遍历输出, 代码详见 EmployeeController。
```

```
@RequestMapping (value="/ employee/ toassociation"... //添加用户之前, 要  
进行关联校验
```

```
public ModelAndView toassociation(String loginname, ModelAndView  
mv, HttpSession session, Model model) {
```

```
User user = ahualyservice. findUserByLogin(loginname) ;
```

```
Employee employee=null;
```

```
if(user!=null&&employee==null) {//进行关联校验主要代码, 关联成功的必要  
条件是获取到全局 ID, 这个 ID 指向员工信息, 并将此 ID 存入 mv 中, 前端页  
面直接从 mv 中获取全局 ID, 代码见前端页面 jsp-employee-association. jsp。
```

```
List<Dept> dept_list = ahualyservice. findAllDept() ;
```

```
model. addAttribute("job_list", job_list) ;
```

```
@RequestMapping (value="/ employee/ add"... //进行员工信息添加
```

```
if(id!=null) {
```

```
this.genericAssociation(job_id, dept_id, sex_id, education_id, employee);
ahualyservice.update_EmployeeInfo(employee); }else{...} //进行员工信息判
断代码，具体添加过程详见代码。
```

下为EmployeeDynaSqlProvider 主要代码。

```
SELECT("*");
FROM(EMPLOYEEETABLE);

if(employee.getDept() != null && employee.getDept().getId() != null &&
employee.getDept().getId() != 0) //进行条件判断，查看部门 id 是否存在。{
    WHERE(" DEPT_ID = #{employee.dept.id} ");
    if(params.get("pageModel") != null) {
        sql += " limit #{pageModel.firstLimitParam} , #{pageModel.pageSize}
"; //进行动态查询，为分页动态查询内容。
```

public String count(Map<String, Object> params) //动态查询，原理同分页查询类似，目的为动态查询总数量。

```
return new SQL() {
    {SELECT("count(*)"); FROM(EMPLOYEEETABLE); ...} //具体详细代码
见 EmployeeDynaSqlProvider
```

下为 EmployeeDao 主要代码：

```
@Select("select * from "+EMPLOYEEETABLE+" ") //选择查询的数据库表
@Results({ //进行查询
    @Result(column="DEPT_ID", property="dept", //通过员工表的ID来查询 user
表中的信息。
```

(select ="com. dao. DeptDao. get\_Info", fetchType=FetchType. EAGER)), //此  
处使用了一种加载方式-急加载，急加载指的是加载一个实体，若定义了急加载，  
则定义该属性的对象会立即从数据库加载。

```
@Select("select * from "+EMPLOYEEETABLE+" where name like binary
CONCAT(' %', #{content}, ' %') " + "or user_id =(select emp_id from
"+USERTABLE+" where loginname =#{content})") //进行员工注册名的查询。
```

```
@Results({
    @Result(column="user_id", property="user",
```

one=@One(select="com. dao. UserDao. get \_StaticInfoLoginname", fetchType=FetchType. EAGER)),//将查询到的结果传递到页面上, 并以急加载的方式展示。

其余部分代码原理类似, 数据库操作代码可见 dao 层对应文件, 在此不做重复说明。

@Select("select SUBSTRING(address, 1, 2) from "+EMPLOYEEETABLE+" ")

List<String> get\_addressList();//此处为添加数据可视化获取员工省份信息的部分代码, 与数据可视化对应, 具体内容见详细设计中的数据可视化部分。

下图为员工管理逻辑流程图。

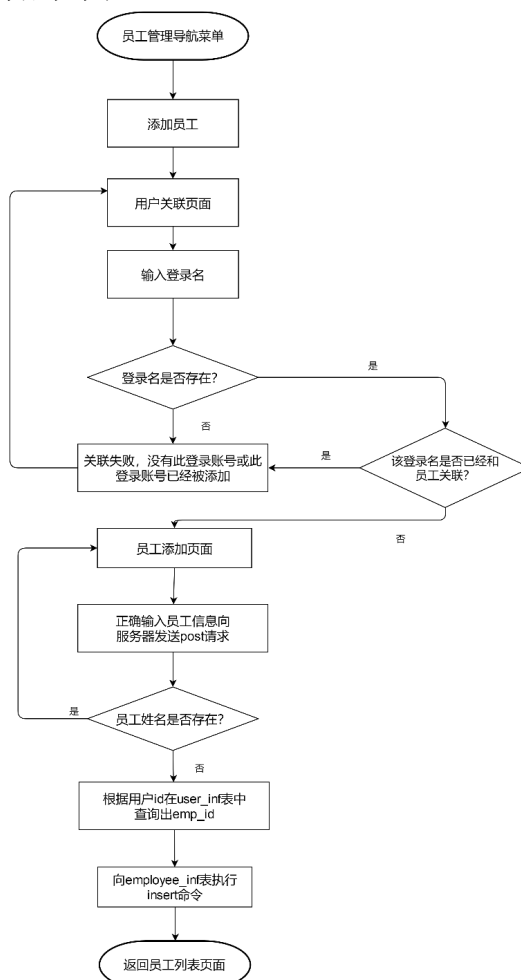


图 4.5 (1) 员工管理逻辑流程图

员工管理与企业信息管理的对应展示界面如下图所示。



图 4.5（2）员工信息列表展示



图 4.5（3）企业职位信息列表展示



图 4.5（3）企业职位信息列表展示

#### 4.4 公告管理模块

公告管理模块主要面向的使用者为已登录用户与管理员，其主要功能为管理员发布公告，本模块主要功能点在于用户可随时了解管理员发布的信息，以及了解信息的发布者是谁。

本功能业务主要使用到的类为 controller 层下的 NoticeController 与 dao 层的 NoticeDynaSqlProvider 和 NoticeDao。其实现原理与员工管理模块类似。

公告管理模块的业务逻辑流程图如下图所示。

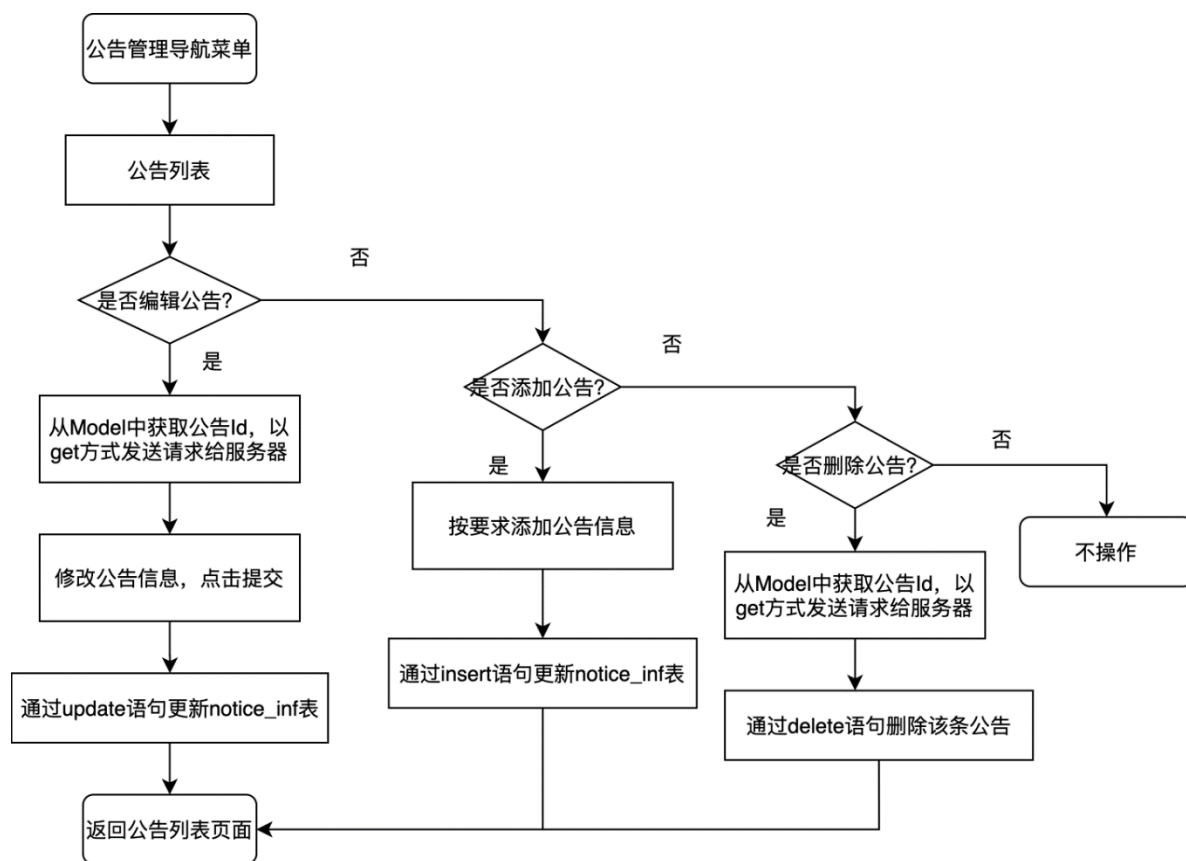


图 4.4 (1) 公告管理模块的业务逻辑流程图

公告管理对应展示界面如下图所示。

| ID | 标题       | 内容                       | 发布日期                | 发布用户  | 操作  |
|----|----------|--------------------------|---------------------|-------|-----|
| 1  | 你好       | 明天就是你自己的生日了              | 2020-04-22 15:06:36 | 通用管理员 | ✎ 查 |
| 2  | 下午开会     | 技术部下午3:00开会,请各部门准时参加会议   | 2020-04-08 15:13:24 | 总经理   | ✎ 查 |
| 3  | 五一放假     | 五一放假安排时间调整及时通知           | 2020-04-27 19:06:23 | 通用管理员 | ✎ 查 |
| 8  | 这是个新公告   | 这是一个新公告啊!                | 2020-05-10 17:13:00 | 通用管理员 | ✎ 查 |
| 9  | 最后测试     | 本次测试为最后一测试, 出现问题请联系开发人员! | 2020-05-20 11:12:16 | 通用管理员 | ✎ 查 |
| 10 | 开发人员联系方式 | 侯建伟15653248960, 微信同号     | 2020-05-20 11:13:49 | 通用管理员 | ✎ 查 |

图 4.4 (2) 查看全部公告

\*标题

\*内容

图 4.4 (3) 添加新公告



## 4.5 下载中心模块

下载中心模块主要面向管理员与已注册用户。其中管理员可上传文档，支持除文件夹外的各种格式文件的上传和下载。员工可下载管理员上传的全部资料。

### 4.5.1 文件的上传

上传文档本质也是属于表单提交，如果需要传输的视频，图片，word 文档等二进制文件还需要使用表单当中的 `enctype` 属性，此属性的属性值 `multipart/form-data` 会告诉浏览器我们提交的请求是不是 `form` 请求，而是一个 `mutipart` 请求。首先需在 `springmvc-config` 中添加配置一个 `Mutipart Resolver`，用来告诉 `DispatchServlet` 如何进行解析 `mutipart` 请求。当以上步骤进行完后，`controller` 才可正式进行文件处理操作，具体业务流程见图 4.5。

下为主要代码解释：

```
<bean id="multipartResolver" >
    <property name="maxUploadSize"> //上传文件大小限制，默认单位为
    字节
        <value>20480000</value>
    </property> //内存中的最大值
    <property name="maxInMemorySize" value="4096" />
    //请求的编码格式需要和 JSP 页面中的编码格式保持一致，否则无法读取上传的
    表单内容。此处设置为 UTF-8。
    <property name="defaultEncoding">
        <value>UTF-8</value>
    </property>
</bean>
```

上传文档对应展示界面如下图所示。

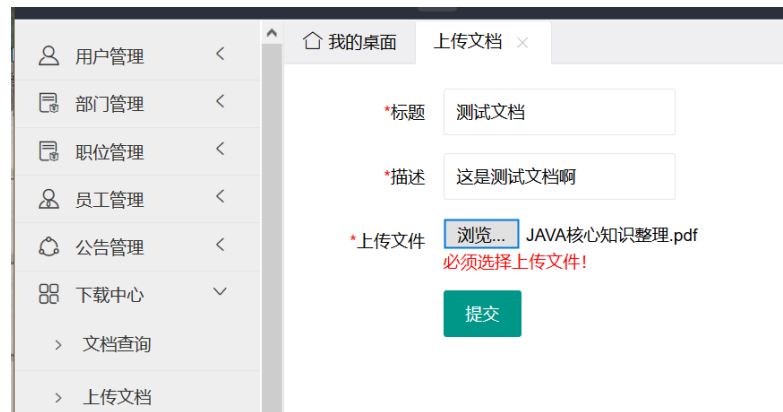


图 4.5.1 上传文档

## 4.5.2 文件下载

用户或管理员点击所下载文档右侧的下载按钮，便可实现下载文件功能。当使用者点击下载后，浏览器获取到将下载文件的 ID，之后发送下载请求。文件下载主要使用到的类为 controller 层的 DocumentController 类，其工作原理是：首先通过文件 ID 与数据库中文件名 ID 进行比对，比对相同后，获取文件所在的路径，找到文件后将文件放入文件输入流对象，使用 HTTP 响应头完成下载功能。

DocumentController 类主要业务代码如下。

```
@RequestMapping(value="/document/add")//上传新文档
```

```
String path = session.getServletContext().getRealPath("/WEB-INF/files");//获取真实路径
```

```
String downloadFielName = new String(fileName.getBytes("UTF-8"), "iso-8859-1");//防止中文乱码
```

```
headers.setContentDispositionFormData("attachment", downloadFielName);//以 attachment 方法通知浏览器进行下载。
```

```
headers.setContentType(MediaType.APPLICATION_OCTET_STREAM);//使用二进制数据流下载。
```

下图为下载文件对应的展示页面。

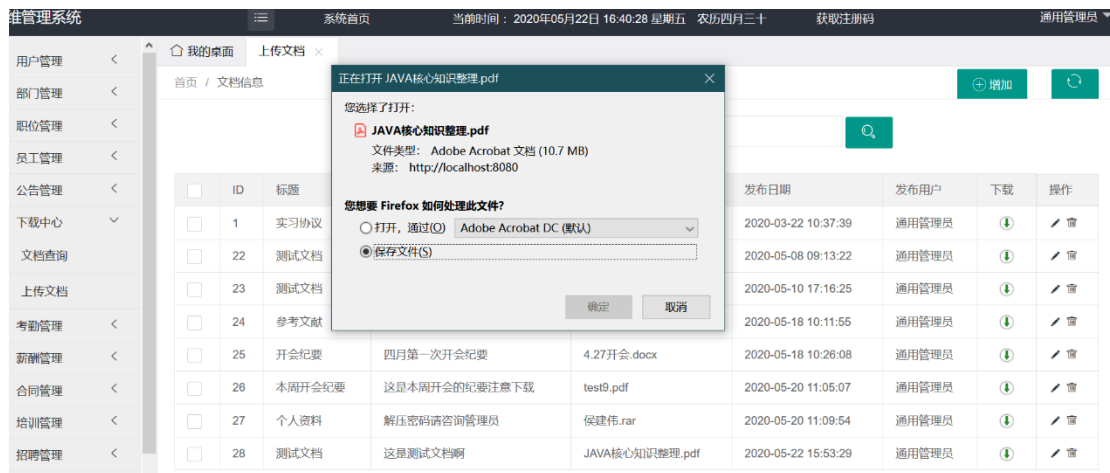


图 4.5.2 (1) 下载文件功能

上传与下载的业务逻辑流程如图 4.5.2 (2) 所示。

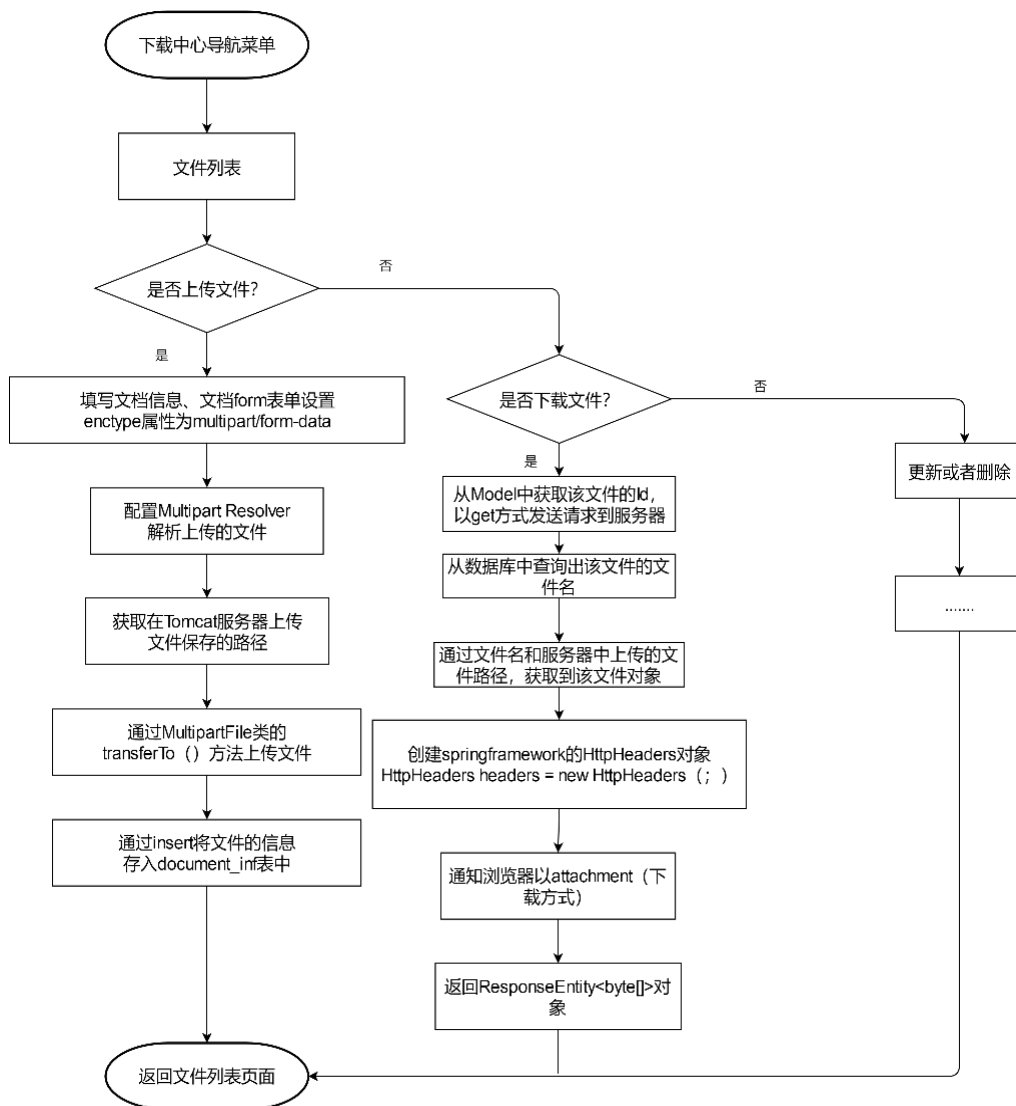


图 4.5.2 (2) 上传与下载的业务逻辑流程图

## 4.6 培训管理模块

培训管理主要功能为管理员添加与修改培训信息，并实现培训资料的下载与在线播放。添加与修改培训信息与员工信息管理部分原理基本相同，下载功能与文档的传输原理基本相同，以上两部分不做具体介绍，仅做功能演示。本模块将重点介绍视频的在线播放功能。

在线播放的使用者包括管理员与普通用户，用户点击可播放按钮后，系统自动新页面，页面内嵌播放器会播放当前视频。其大致的工作原理为：用户点击按钮后，前端会将获得的视频名称作为参数传递至 controller 层中的 TrainDataController，TrainDataController 会对视频名称做相应的字符串处理，将字符串存入 model 中，视频播放通过 EL 表达式拼接处完整视频路径。视频的在线播放借助了 ckplayer 播放器组件。

以下为 TrainDataController 中的部分重要代码。

```
model.addAttribute("filename", filename); //将视频名称存入 model 中。
```

```
var video=['${ctx}/upload/${requestScope.filename}-> video/ mp4']; //
```

从前端进行视频文件名的获取。

```
CKObject.embedHTML5 (' a1', ' ckplayer _a1', 600 , 400 , video ,  
flashvars, support); //ckplayer 插件使用 embedHTML5 () 方法进行视频播放
```

[11]。

视频播放的业务逻辑流程图如下图所示。

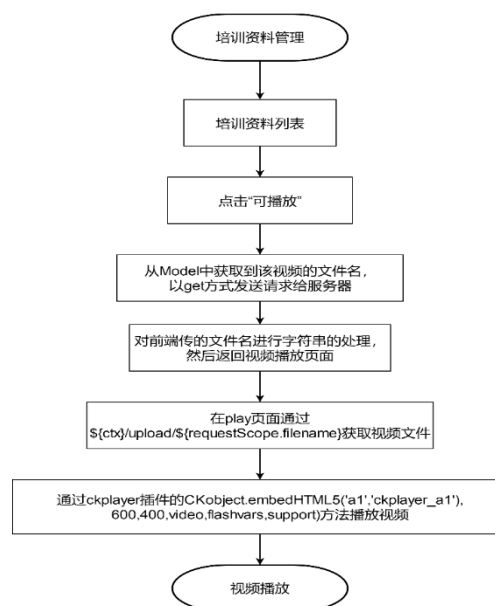


图 4.6（1）视频播放的业务逻辑流程图

培训管理主要功能展示如下所示。



图 4.6（2）查看培训详情



图 4.6（3）查看培训资料

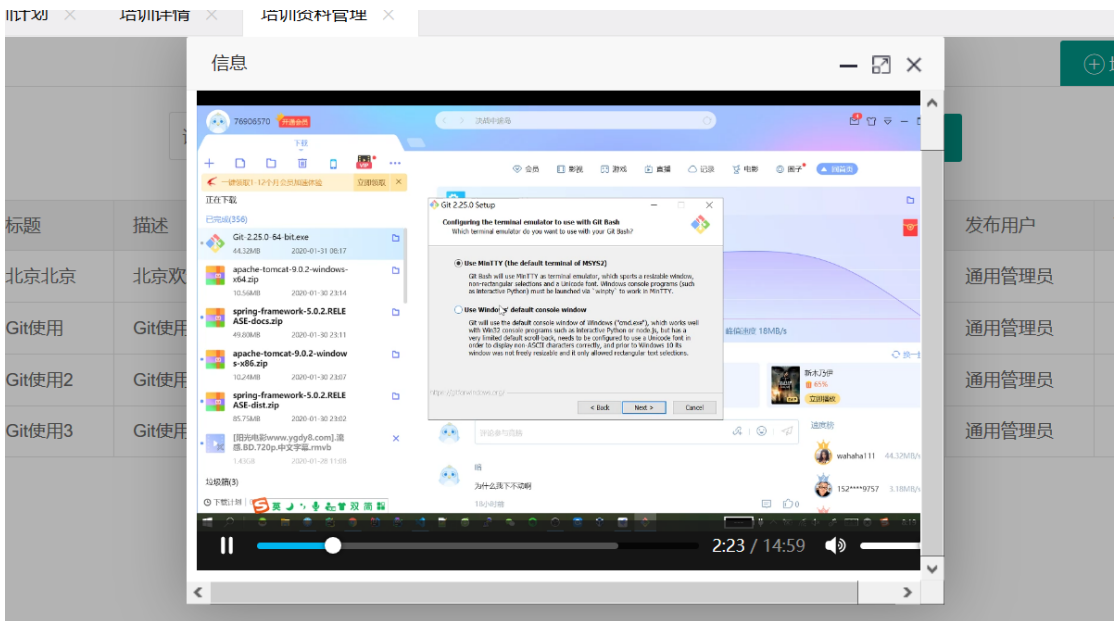


图 4.6（4）视频在线播放

## 4.7 考勤管理

考勤管理主要包括员工上下班打卡，员工请假管理与管理员的考勤统计和请假审批，考勤统计与管理员请假审批实现相对简单，所涉及的只是基础的增删改查，和员工管理模块功能类似。本模块主要介绍员工上下班打卡功能。

管理员进行打卡时间确定以及员工打卡时间判断主要使用的是

PunchClockController 类，以下为 PunchClockController 中的关键代码。

```
PunchTime punchTime = ahualyService.getPt(); // 员工获取管理员的打卡时间
```

```
Calendar cal = Calendar.getInstance(); // 获取系统时间，后续代码主要工作为切分时间，进行打卡时间判断。
```

下班打卡时间的判定与上班打卡判定的原理是相同的，在此不再赘述。

```
@RequestMapping(value = "/checkwork/punchlist") // 通过用户 ID 查询打卡记录
```

```
@RequestMapping(value = "/checkwork/statistics") // 获取用户统计数据信息  
(统计天数以及迟到或早退的次数)
```

下图为员工上下班打卡业务与员工请假逻辑流程图。

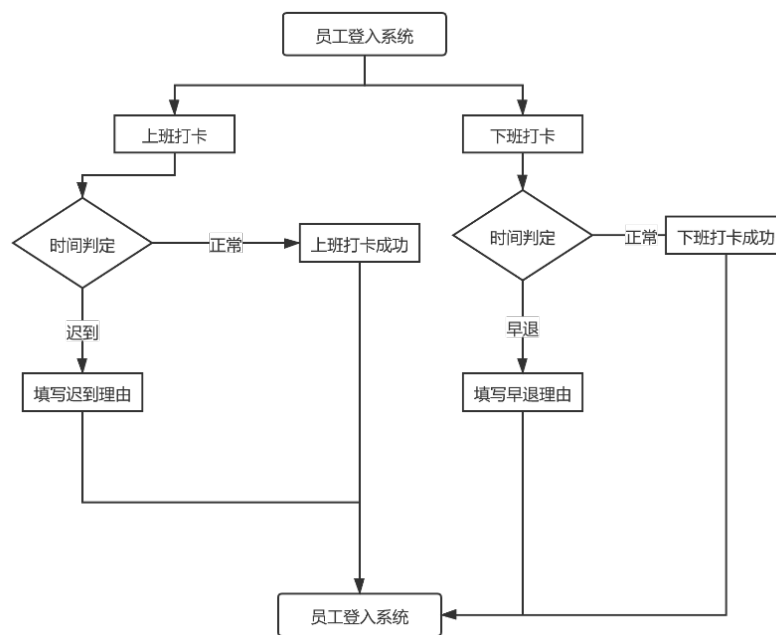


图 4.7 (1) 员工上下班打卡业务逻辑流程图

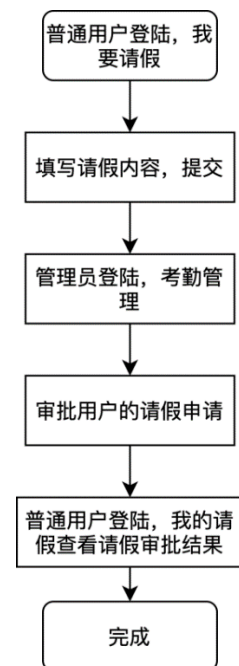


图 4.7 (2) 请假业务逻辑图

考勤管理主要功能展示如下所示。

部门管理

职位管理

员工管理

公告管理

下载中心

考勤管理

打卡时间

考勤查询

请假申请

薪酬管理

合同管理

我的桌面

打卡时间

考勤查询

\*上班时间

09:00

\*下班时间

18:00

提交

图 4.7（3）管理员规定签到时间

| <input type="checkbox"/> | 姓名  | 职位  | 部门  | 开始日期       | 结束日期       | 请假天数 | 请假类型 | 请假内容 | 审批状态 | 请假时间       | 操作 |
|--------------------------|-----|-----|-----|------------|------------|------|------|------|------|------------|----|
| <input type="checkbox"/> | 喵喵喵 | 主管  | 市场部 | 2020-05-07 | 2020-05-16 | 9    | 病假   | 感冒   | 批准   | 2020-05-08 | 编辑 |
| <input type="checkbox"/> | 侯建伟 | 架构师 | 技术部 | 2020-04-11 | 2020-04-13 | 2    | 病假   | 感冒了  | 未批准  | 2020-04-11 | 编辑 |
| <input type="checkbox"/> | 侯建伟 | 架构师 | 技术部 | 2020-05-15 | 2020-05-18 | 3    | 事假   | 回老家  | 未批准  | 2020-05-12 | 编辑 |

[上一页](#)
[下一页](#)
[跳转到](#)  [确定](#)  
 总共3条记录, 当前显示1-3条记录。

图 4.7（4）管理员查看员工请假申请

我的桌面

打卡时间

考勤查询

首页 / 考勤管理

增加

刷新

请输入查找内容

搜索

| <input type="checkbox"/> | 姓名  | 职位  | 部门  | 上班天数 | 请假天数 | 迟到天数 | 建档日期                | 操作 |
|--------------------------|-----|-----|-----|------|------|------|---------------------|----|
| <input type="checkbox"/> | 侯建伟 | 架构师 | 技术部 | 23   | 0    | 0    | 2020-05-12 16:05:02 | 编辑 |
| <input type="checkbox"/> | 李四  | 职员  | 财务部 | 15   | 8    | 1    | 2020-05-17 16:34:24 | 编辑 |

[上一页](#)
[下一页](#)
[跳转到](#)  [确定](#)  
 总共2条记录, 当前显示1-2条记录。

图 4.7（5）管理员查看员工考勤记录

我的桌面

欢迎 侯建伟 登录系统! 当前时间:2020年05月22日 18:56:58 星期五

上班打卡

下班打卡

系统通知

暂时没有通知!

系统信息

暂时没有通知!

迟到, 请到个人考勤填写迟到原因

确定

欢迎 侯建伟 登录系统! 当前时间:2020年05月22日 18:58:14 星期五

上班已打卡

下班打卡

系统通知

暂时没有通知!

系统信息

暂时没有通知!

早退, 请到个人考勤填写早退原因

确定

图 4.7（6）员工上下班打卡

## 4.8 数据可视化模块

数据可视化模块主要包含以下三部分：第一部分是查看企业内员工男女比例、第二部分是查看各个职位的员工个数、第三部分是查看员工的户籍所在地的地理位置分布图。这三部分使用 ECharts 插件进行设计，同时使用了非本地存储，减少了系统本身的大小<sup>[12]</sup>。

数据可视化的实现原理大致如下：当用户点击数据报表时，前端页面向后台发送三个 Ajax 请求，服务器按照请求的格式分别从员工表获取性别与所在地，从部门表获取职位人数。查询到所需数据后，借助 service 层的 AhualyService 中的方法返回给前端页面所需的格式，前端页面负责解析数据，并将数据与 ECharts 进行绑定，实现了动态图变化效果<sup>[12]</sup>。

下面对员工所在地分布图设计的主要代码进行展示。

```
List<String> list = employeeDao.get_addressList(); //从数据库获取省份
```

```
Map<String,Integer> map = new HashMap<>(); //循环遍历统计个数
```

下为前端页面员工所在地分布图设计的主要代码。

```
$.ajax({ //使用 ajax 加载数据    method:'post',    url:'${ctx}/echartsData3',
```

数据可视化模块业务流程逻辑图如下图所示。

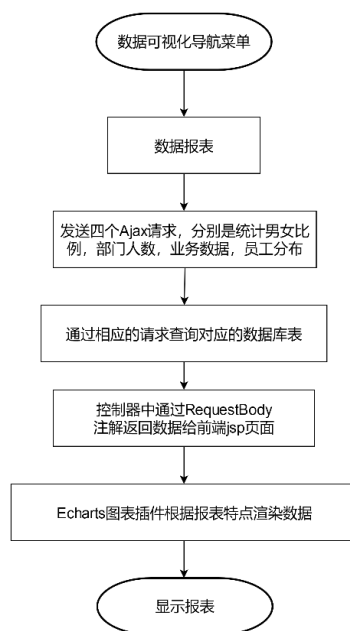




图 4.8（1）数据可视化模块业务流程逻辑图

下图为数据可视化功能展示。

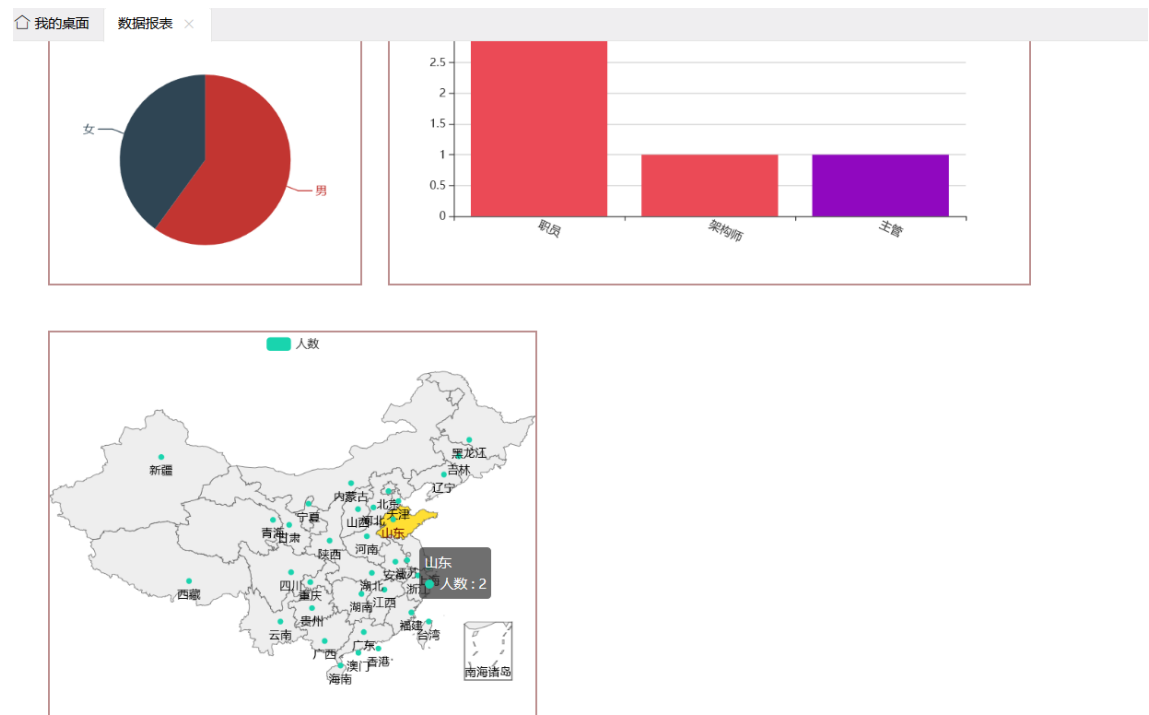


图 4.8（2）数据可视化模块功能展示

## 第 5 章 系统测试

在系统的实际开发中，遇到各种各样的问题是在所难免的，进行系统测试的目的是为了找出系统中潜在的错误，找出错误发生对应的代码块并进行解决。

### 5.1 测试类型

系统测试在软件开发中拥有极其重要的作用，如果系统出现问题，轻则导致系统运行速度降低以及处理速度减慢，重则导致系统数据异常或系统崩溃，导致不可挽回的损失。根据调查，测试的工作量应占据系统开发的总工作量的 35% 左右<sup>[15]</sup>。

一般来说，测试的主要类型有动态测试、静态测试、黑盒测试与白盒测试<sup>[13]</sup>。本系统测试主要采用的测试方法有以下两种：黑盒测试与白盒测试。黑盒测试相当于将软件放入一个不透明的容器中，测试人员需输入数据，观察输出数据是否与预期结果一致。若输出结果一致，则通过测试；若输出结果不一致，即使输出结果差异较小，也说明系统内部设计出现了问题，需要尽快解决。白盒测试相当于在一个透明的容器中进行测试，需要根据系统中的源代码，不使用用户的图形界面，对系统内的工作过程进行测试。

黑盒测试使用相对简单，即使不了解内部代码具体实现，也可很快上手进行测试。而且可以从用户的角度出发，以用户的视角来使用软件，以得出用户可能会遇到的问题。最重要的是，黑盒测试更容易实现自动化测试，可大幅节约测试时间<sup>[14]</sup>。

白盒测试可以增大测试开发代码的覆盖率覆盖率，相对于黑盒测试来说，覆盖面更广，可以考虑到一些系统界面上无法考虑到的问题。通过白盒测试，可以发现代码中未解决的问题<sup>[14]</sup>。

### 5.2 测试用例

测试用例简而言之，主要包括测试输入、执行条件、预期结果和实际结果，其作用是检查软件是否满足规定的需求。

本小节将对注册、登录、员工信息管理、下载资料管理、招聘管理几个模块进行测试，主要使用黑盒测试技术进行测试。

#### 5.2.1 登录测试用例

下表主要对登录进行测试，根据实际结果可以得知，测试结果均符合预期，证明登录模块功能正常，能够满足使用者进行登录的需求。

表 5.2.1 登录测试用例

| 编号 | 测试步骤                 | 预期结果                | 实际结果                 |
|----|----------------------|---------------------|----------------------|
| 1  | 在登录界面输入未注册的用户名       | 提示：非法用户，请重新输入       | 与预期结果相符，功能正常         |
| 2  | 输入正确用户名，输入错误密码       | 提示：密码错误，并清空已输入信息    | 与预期结果相符，功能正常         |
| 3  | 输入正确用户名，正确密码，验证码输入错误 | 提示：验证码输入错误，并清空已输入信息 | 与预期结果相符，功能正常         |
| 4  | 同时登陆两个已登录账户          | 将第一次登陆账户强制退出        | 第一次登录未强制退出，但无法进行任何操作 |
| 5  | 按正常步骤登入系统            | 根据账户信息，进入员工页面或管理员页面 | 与预期结果相符，功能正常         |

### 5.2.2 注册测试用例

下表主要对注册功能进行测试，根据实际结果可以得知，测试结果均符合预期，证明注册模块功能正常，能够满足新用户进行注册的需求。

表 5.2.2 注册测试用例

| 编号 | 测试步骤                | 预期结果              | 实际结果         |
|----|---------------------|-------------------|--------------|
| 1  | 注册码页面填写错误注册码        | 提示：注册码错误，请重新输入注册码 | 与预期结果相符，功能正常 |
| 2  | 输入已使用的注册码           | 提示：注册码错误，请重新输入注册码 | 与预期结果相符，功能正常 |
| 3  | 输入正确的注册码            | 进入注册个人信息填写界面      | 与预期结果相符，功能正常 |
| 4  | 登录名与已有登录名相同         | 提示：登录名已被注册，请重新输入。 | 与预期结果相符，功能正常 |
| 5  | 正则表达式测试：邮箱地址与用户名非法  | 提示：您的邮箱/用户名非法     | 与预期结果相符，功能正常 |
| 6  | 用户按正确格式输入信息完成注册     | 完成注册，返回登录页        | 与预期结果相符，功能正常 |
| 6  | 管理员端直接添加用户，输入已存在登录名 | 提示：登录名已被注册，请重新输入  | 与预期结果相符，功能正常 |
| 7  | 管理员端获取三次以上验证码       | 提示：本日机会已用尽，请明日再试  | 与预期结果相符，功能正常 |
|    | 管理员按正确格式输入信息完成注册    | 完成注册，返回后台管理主界面    | 与预期结果相符，功能正常 |

### 5.2.3 员工信息管理测试用例

下表主要对员工信息管理功能进行测试，根据实际结果可以得知，测试结果均符合预期，证明注册模块功能正常，能够满足新用户进行注册的需求。

表 5.2.3 员工信息管理测试用例

| 编号 | 功能 | 测试步骤  | 预期结果                         | 实际结果          |
|----|----|---|------------------------------|---------------|
| 1  | 添加 | 1. 点击员工管理, 查看全部员工信息。<br>2. 点击上侧增加, 验证用户关联, 并输入正确用户名<br>3. 用户关联验证后, 进入添加页面 | 成功进入添加员工信息界面                 | 与预期结果相符, 功能正常 |
| 2  | 添加 | 1. 点击员工管理, 查看全部员工信息<br>2. 点击上侧增加, 验证用户关联, 输入错误用户名<br>3. 用户关联验证后, 进入添加页面   | 提示: 关联失败, 没有此登录账号或此登录账号已经被添加 | 与预期结果相符, 功能正常 |
| 3  | 添加 | 1. 进入添加员工信息界面<br>2. 不填写必填项  | 提示: 必填项不可为空                  | 与预期结果相符, 功能正常 |
| 4  | 添加 | 1. 进入添加员工信息界面<br>2. 验证手机, 身份证号码, 联系地址输入错误                                 | 提示: 请输入正确信息                  | 与预期结果相符, 功能正常 |
| 5  | 添加 | 全部按正确步骤执行   | 完成添加员工信息                     | 与预期结果相符, 功能正常 |
| 6  | 编辑 | 点击员工列表左侧操作按钮, 进行编辑  | 跳转到员工信息编辑页                   | 与预期结果相符, 功能正常 |
| 7  | 编辑 | 1. 进入编辑员工信息界面<br>2. 不填写必填项  | 提示: 必填项不可为空                  | 与预期结果相符, 功能正常 |
| 8  | 编辑 | 1. 进入编辑员工信息界面<br>2. 验证手机, 身份证号码, 联系地址输入错误                                 | 提示: 请输入正确信息                  | 与预期结果相符, 功能正常 |
| 9  | 编辑 | 全部按正确步骤执行   | 完成编辑员工信息                     | 与预期结果相符, 功能正常 |
| 10 | 删除 | 点击操作中的删除按钮<br>选择取消  | 用户信息不会被删除                    | 与预期结果相符, 功能正常 |
| 11 | 删除 | 点击操作中的删除按钮<br>选择确定  | 用户信息被删除                      | 与预期结果相符, 功能正常 |

#### 5.2.4 下载资料管理测试用例

下表主要对下载资料管理功能进行测试, 根据实际结果可以得知, 测试结果均符合预期, 证明下载资料模块功能正常, 能够满足员工和管理员下载资料的需求。

表 5.2.4 下载资料管理测试用例

| 编号 | 功能   | 测试步骤  | 预期结果               | 实际结果          |
|----|------|---|--------------------|---------------|
| 1  | 查询   | 1. 点击导航栏下载中心-查询文档   | 可查看全部上传文档          | 与预期结果相符, 功能正常 |
| 2  | 上传文档 | 1. 点击导航栏下载中心-上传文档<br>2. 输入标题, 描述, 并选择上传文件                             | 成功上传, 返回文档查询页面     | 与预期结果相符, 功能正常 |
| 3  | 上传文档 | 1. 点击导航栏下载中心-上传文档<br>2. 不输入标题, 描述, 或不选择上传文件                           | 提示必填项不能为空          | 与预期结果相符, 功能正常 |
| 4  | 编辑文档 | 1. 点击导航栏下载中心-查询文档<br>2. 点击操作-编辑进行编辑<br>3. 进入编辑页, 输入新标题, 新描述, 并选择上传新文件 | 成功上传, 返回文档查询页面     | 与预期结果相符, 功能正常 |
| 5  | 删除文档 | 1. 点击导航栏下载中心-查询文档<br>2. 点击操作-删除执行删除操作<br>3. 点击确定, 可删除文档, 点击取消, 文档保留   | 成功执行操作, 可选择删除或保留文档 | 与预期结果相符, 功能正常 |
| 6  | 下载文档 | 点击操作中的下载按钮<br>选择确定  | 成功下载文件到指定路径        | 与预期结果相符, 功能正常 |

### 5.2.5 招聘管理测试用例

下表主要对招聘管理功能进行测试, 根据实际结果可以得知, 测试结果均符合预期, 证明招聘管理模块功能正常, 能够满足员工和管理员下载资料的需求。

表 5.2.5 招聘管理测试用例

| 编号 | 功能     | 测试步骤   | 预期结果                                  | 实际结果          |
|----|--------|--|---------------------------------------|---------------|
| 1  | 查看招聘信息 | 1. 从登录页面点击招聘信息<br>2. 点击投递简历, 查看全部招聘信息                      | 可查看全部招聘信息, 包括岗位信息, 人数, 状态, 截止日期, 投递等。 | 与预期结果相符, 功能正常 |
| 2  | 投递简历   | 1. 从登录页面点击招聘信息<br>2. 点击投递简历, 查看全部招聘信息<br>3. 点击立即投递, 投递简历信息 | 成功投递简历, 并提示投递成功, 点击返回可继续投递            | 与预期结果相符, 功能正常 |

续表 5.2.5

|   |         |  |                        |               |
|---|---------|--|------------------------|---------------|
| 3 | 投递简历    | 1. 从登录页面点击招聘信息<br>2. 点击投递简历, 查看全部招聘信息<br>3. 点击立即投递, 进行投递<br>4. 未填写一项或多项个人信息的填写, 点击完成 | 提示必填项不能为空              | 与预期结果相符, 功能正常 |
| 4 | 管理员查看简历 | 1. 点击招聘管理-申请概况   | 可查看全部投递简历              | 与预期结果相符, 功能正常 |
| 5 | 管理员审核简历 | 1. 点击招聘管理-申请概况<br>2. 点击操作-马上审核<br>3. 点击确定, 可选择已审核或未审核                                | 成功执行操作, 成功审核投递者会收到提示   | 与预期结果相符, 功能正常 |
| 6 | 管理员岗位添加 | 1. 点击招聘管理-岗位公布<br>2. 点击上方增加<br>3. 按要求输入必填项   | 成功执行操作, 并返回岗位公布页面      | 与预期结果相符, 功能正常 |
| 7 | 管理员岗位添加 | 1. 点击招聘管理-岗位公布<br>2. 点击上方增加<br>3. 未按要求输入必填项  | 提示必填项不能为空              | 与预期结果相符, 功能正常 |
| 8 | 管理员岗位删除 | 1. 点击招聘管理-岗位公布<br>2. 点击操作-删除<br>3. 选择确定删除与取消   | 成功执行操作, 可选择删除或保留招聘岗位信息 | 与预期结果相符, 功能正常 |
| 9 | 下载简历    | 选择想要下载的员工简历, 点击下载按钮  | 成功下载文件到指定路径            | 与预期结果相符, 功能正常 |

## 第 6 章 总结与展望

### 6.1 总结

本论文是基于 SSM 框架的运维管理系统，主要功能是实现企业内信息的高效管理。本系统主要面向企业内部人员，包括员工与管理员，以方便企业内部共享信息，进行运维管理。

本论文主要完成了以下的几项工作。

(1) 通过上网查阅相关文献，进行调研，了解运维管理系统的背景，以及运维管理系统在国内外的的发展状况，并确定了论文的研究内容与研究方向。

(2) 综合在大学期间所学习的相关知识，进行技术选型，确定使用 MySQL 数据库，SSM 框架等技术进行系统开发。

(3) 利用软件开发与数据库概念结构设计的相关知识，进行了系统设计与数据库设计，确立了系统的规格与系统的功能，并按照需求建立了数据库以及数据库相应的表与数据之间的关系。

(4) 按照论文第三章系统设计的内容，进行了代码的详细设计。实现了系统设计中的全部功能。并添加了验证码注册，视频在线播放，数据可视化的功能。

(5) 使用黑盒与白盒测试技术，完成了运维管理系统的综合测试。通过进行黑盒测试与白盒测试，解决了一些系统逻辑错误与代码错误。并总结整理出测试用例。

### 6.2 展望

由于开发本系统的时间有限，有许多功能还没有实现最优化，在以后的工作中，还需要对以下几方面进行进一步的优化。

(1) 系统的功能会随时间的变化而不断更新，可能会添加新的功能与新的数据库表，这可能会导致与已有数据相冲突，导致系统出现问题。所以，在今后的系统设计中，需要进行基于当前系统的更具体，更详细的系统设计。

(2) 部分外键约束没有添加，导致用户和管理员的薪酬管理，用户和管理员的合同管理之间需要手动添加约束，还需要进一步的完善与研究。

(3) 软件的性能还需要进一步的提升，需要通过优化数据库查询语句以加快响应速度，考虑到网络情况，考虑把 ECharts 数据可视化模块导入本地库。

## 致谢

毕业设计在这部分也就将告一段落了，在即将告别青岛理工大学之际，首先感谢我的母校，在大学四年的时光里，每一天我都过得很开心快乐。在这里，我不仅学习到了有关计算机的专业知识和相关技能，而且还学到了许多在社会上无法学到的东西。

感谢我的同学们，在你们的帮助下，我解决了许多系统设计上的小问题。有大家这样优秀的同学陪我度过大学四年的幸福时光，可以说是人生无憾了。

感谢我的家人，感谢你们这些年来对我的培养和生活上的支持。今年因为情况特殊，无法在校完成毕业设计。感谢我的家人为我提供了一个良好的学习环境，保证了我按时高效的完成毕业设计的相关任务。

感谢我的指导老师—李兰老师。虽然大学期间这是第一次正式的参加您的课程，但是通过与您探讨与学习，我学习到了许多从课本上无法学习的知识。若是没有老师您的帮助，我很难独立高效的完成毕业设计的各项工作。李兰老师那热心助人、教学严谨、求真务实的态度，都深深地影响着我和毕设小组中的每位成员，李兰老师是我们大家学习和借鉴的榜样！在此我由衷的向李兰老师表达我最诚挚的谢意和最衷心的祝福，在此希望李兰老师身体健康，事业顺利，桃李满天下！

最后感谢各位论文评审老师百忙之中对我的论文进行评审，感谢各位评审老师的辛苦付出！

侯建伟

2020年6月



## 参考文献

- [1]褚宗饶. 医院信息平台运维管理系统需求分析与设计[J]. 医学信息学杂志, 2015(9):42-46.
- [2]万涵. 智能化运维管理平台的规划设计探讨[J]. 通讯世界, 2019(6):131-133.
- [3]李甘旭. 青岛移动 WLAN 网络建设项目管理系统的设计与实现[D]. 山东:山东大学, 2015.
- [4]武强. 基于 B/S 三层架构下的基层连队管理信息系统的设计与实现[D]. 吉林:长春工业大学, 2018.
- [5]马建红、李学相、韩颖、王瑞娟、张晗. JSP 应用与开发技术(第3版)[M]. 北京:清华大学出版社 2018(07).
- [6]陈恒、楼偶俊、张立杰. JavaEE 框架整合开发入门到实践[M]. 北京:清华大学出版社 2018(02).
- [7]K. Siva Prasad Reddy. Java Persistence with MyBatis 3[J]. 2013.
- [8]丁允超. MySQL 网络数据库设计与开发(第2版)[M]. 北京:电子工业出版社. 2018(09)
- [9]张伟红. 网页打开速度快慢的决定因素[J]. 计算机与网络, 2014(10):46.
- [10]厉小军. 软件开发过程及规范[M]. 北京:清华大学出版社, 2013(07).
- [11]翟晓宁, 乔杰华, 王钰博. 基于 VLC 和 ckplayer 的视频直播系统的设计及实现[J]. 数字化用户, 2017, 023(044):291.
- [12]郑幸源, 洪亲, 蔡坚勇, et al. 基于 AJAX 异步传输技术与 Echarts3 技术的动态数据绘图实现[J]. 软件导刊, 2017(3):143-145.
- [13]Cristani M , Olivieri F , Tomazzoli C , et al. Diagnostics as a Reasoning Process: From Logic Structure to Software Design[J]. Journal of Computing and Information Technology, 2019, 27(1):43-57.
- [14]Roth R R . SOFTWARE TESTING: US 2008.
- [15]朱少民. 软件测试方法和技术[M]. 清华大学出版社, 2010.