**基于SpringBoot + Vue的**

**考勤管理系统的设计与实现**

姓 名： 鲍一诺 陈蓓儿

学 号： 1241004007 1241004003

指导教师：

**2025年7月**

**摘要**

考勤是学校以及单位对学生、员工的基本要求，它可以很好的反映出学生是否准时上下课，员工是否准时上下班。但很多时候还是会采用纸质考勤表，大大的浪费了时间，而且还存在作假的嫌疑。所以一个好的考勤管理系统的设计与实现是顺应历史潮流发展的产物。

本文分析了国内外单位信息化的现状，阐述了开发单位信息化服务平台的意义，以及具体如何开发单位考勤管理系统的设计与实现平台，以此推动单位发展。本系统用了SpringBoot框架和Vue框架以及ElementUI，实现了前后端分离。SpringBoot框架使用大量的注解，大大减少了开发的工作量，使用Vue框架和ElementUI提供的许多组件，降低了前端页面开发的难度，再结合MySQL数据库，实现了系统的主要功能。

本课题的系统是面向全体学生和员工的，并不只是提供给单位管理者使用，不仅能对单位信息进行管理，最重要的是还能给普通学生和员工提供服务。本系统能使单位各方面信息的管理更加集中，更加信息化，有利于单位的管理，促进单位的发展。

**关键词**：SpringBoot技术；单位信息化；服务平台

**Abstract**

Attendance checking is a basic requirement for schools and units for students and employees. It can well reflect whether students attend classes on time and whether employees go to and from work on time. However, paper attendance sheets are still used in many cases, which is a great waste of time and also suspected of fraud. Therefore, the design and implementation of a good attendance management system is a product that conforms to the trend of historical development.

This article analyzes the current situation of informatization in units at home and abroad, expounds the significance of developing an informatization service platform for units, and specifically how to develop the design and implementation platform of the unit's attendance management system, so as to promote the development of the unit. This system uses SpringBoot framework, Vue framework and ElementUI to realize the separation of front-end and back-end. The SpringBoot framework uses a large number of annotations, which greatly reduces the development workload. The use of many components provided by Vue framework and ElementUI reduces the difficulty of front-end page development. Combined with MySQL database, the main functions of the system are realized.

The system of this subject is oriented to all students and employees, not only for unit managers. It can not only manage unit information, but more importantly, provide services for ordinary students and employees. This system can make the management of all aspects of unit information more centralized and information-based, which is conducive to the management of the unit and promotes the development of the unit.

**Keywords:** SpringBoot technology; unit informatization; service platform

目 录

[1绪论 6](#_Toc1699734231)

[1.1研究背景 6](#_Toc827166632)

[1.2研究现状 6](#_Toc1072011970)

[1.3研究目标 7](#_Toc1174667478)

[2相关技术介绍 8](#_Toc1002058752)

[2.1Java语言 8](#_Toc1160360425)

[2.2Spring Boot框架 9](#_Toc1096542908)

[2.3Vue前端技术 10](#_Toc1785108227)

[2.3.1Vue框架 10](#_Toc84655437)

[2.3.2Element UI组件 11](#_Toc153982909)

[2.4Node技术 12](#_Toc2145101295)

[2.4.1Node的优点 12](#_Toc105166072)

[2.5MySQL数据库 12](#_Toc72841828)

[3需求分析 13](#_Toc777234387)

[3.1系统功能需求分析 13](#_Toc1702567681)

[3.1.1业务需求分析 13](#_Toc510870414)

[3.1.2数据需求分析 14](#_Toc1770083652)

[3.2系统非功能性需求分析 15](#_Toc1805856186)

[3.2.1系统处理能力需求 15](#_Toc956205408)

[3.2.2可靠性需求 15](#_Toc1829733206)

[3.2.3可用性需求 15](#_Toc926704210)

[3.2.4维修性需求 15](#_Toc371796596)

[3.2.5环境适配需求 16](#_Toc403461380)

[4系统详细设计与实现 16](#_Toc453836348)

[4.1 普通员工模块主要功能实现 16](#_Toc776810109)

[4.1.1 登录 16](#_Toc1556086753)

[4.1.2 个人中心 16](#_Toc1264878700)

[4.2 工作台模块主要功能实现 17](#_Toc1789497070)

[4.2.1 客户管理 17](#_Toc484117973)

[4.2.2 会议通知管理 17](#_Toc870787824)

[4.2.3 任务审批 17](#_Toc944498731)

[4.3管理员模块主要功能实现 17](#_Toc36368556)

[4.3.1员工管理 17](#_Toc1697954456)

[4.3.2 部门管理 17](#_Toc2016510701)

[5系统测试 17](#_Toc1211036034)

[5.1 系统测试的目的 17](#_Toc552529560)

[5.2 系统测试的方法及步骤 18](#_Toc1029387479)

[5.3测试环境 18](#_Toc160095294)

[6总结与展望 18](#_Toc190154139)

[参考文献 19](#_Toc1114042916)

# **1**绪论

## **1.1**研究背景

在学校和企业管理中，考勤管理是保障教学和企业正常运营秩序、衡量学生和员工工作状态的基础环节。传统考勤方式多依赖纸质签到、人工统计，不仅耗时耗力，还存在数据易篡改、统计误差大、追溯困难等问题。随着企业规模扩大和跨区域办公模式的普及，传统方式已难以满足高效管理需求，考勤数据分散、流程脱节等问题愈发突出，严重制约了管理效率的提升。 同时，数字化转型已成为企业发展的必然趋势，单位信息化建设要求将人力资源管理等核心环节纳入系统化管理。考勤数据作为人力资源决策的重要依据，其准确性、实时性直接影响薪酬核算、绩效评估等关键流程。

在此背景下，构建一套自动化、信息化的考勤管理系统，实现打卡、审批、统计全流程线上化，成为企业降本增效、规范管理的迫切需求。 当前，SpringBoot、Vue等技术的成熟为考勤系统的开发提供了有力支撑。SpringBoot的注解机制简化了后端开发流程，Vue的组件化设计降低了前端页面开发难度，结合MySQL数据库的高效数据处理能力，可实现前后端分离的架构设计，满足系统灵活性和可扩展性要求。基于这些技术构建的考勤管理系统，既能解决传统管理模式的痛点，又能为企业信息化建设提供基础数据支持，推动管理模式向智能化、精细化转型。

## **1.2**研究现状

随着互联网技术的不断发展和普及，考勤管理在全球范围内得到了迅猛的增长。本文将从国内和国外两个方面，对考勤管理系统的研究现状进行综述。

在国内，早期技术方面多采用射频技术嵌入员工卡牌的读卡识别方式，部分高校还沿用课堂点名和抽查点名的方式。如今，一些重点院校推出在线校园考勤管理系统，部分企业引入电子考勤系统，但传统考勤方式的弊端依旧存在，如手工签到、打卡效率低、数据不准确、易造假等，电子考勤系统也面临稳定性、安全性和可扩展性方面的挑战。在系统研究领域，国内研究与国外类似，主要集中在系统设计、功能实现和性能评估等方面，致力于提高系统的各项性能和用户体验，同时也关注系统的扩展性和可定制性。不过，目前国内在多模块融合以及用户体验优化方面仍有不足，且考勤管理缺乏与人力资源管理的有效整合，导致考勤数据在应用上存在局限性。

在国外，技术应用上，国外起步较早且技术先进，部分先进企业利用人脸识别、虹膜识别、指纹识别等生物识别技术以及移动定位技术提升考勤智能化水平，像英国还将 DNA 信息、声音信息等录入考勤管理系统 ，市面上也有舒特公司的拍照考勤机、汉王科技的人脸识别考勤机等热门设备。不过，这些技术在实际应用中存在一定问题，例如指纹识别存在无法识别的情况，蓝牙考勤面临有效接收范围有限的问题，基于 MAC 地址考勤也有技术难题。在系统研究方面，国外研究方向侧重于使用更先进的技术手段实现考勤的自动化和智能化，在系统架构、功能实现和用户体验等方面进行了较多探讨，注重系统的可扩展性、可靠性、安全性和易用性，以及系统与用户之间的交互体验，并且积极探索新兴技术在考勤管理领域的融合应用。在考勤设备方面，虽然设备先进，但在高校等场景适应性差，且设备管理与维护负担重、数据采集不稳定等问题也较为突出。

因此，无论是国内还是国外，考勤管理系统都呈现出蓬勃发展的态势。随着技术的不断进步和市场的不断扩大，考勤管理系统将在未来发挥更加重要的作用。

## **1.3**研究目标

本文旨在通过对基于 SpringBoot + Vue 的考勤管理系统的设计与实现，构建一套能满足企业考勤管理需求的信息化平台。

具体而言，本文主要研究以下方面：

1. 明确系统需求边界：全面梳理企业考勤管理的业务流程，涵盖员工打卡、请假申请、审批处理、部门管理等核心场景，明确不同角色（普通员工、管理员）的功能权限，为系统设计提供精准依据。
2. 构建合理系统架构：采用前后端分离模式，结合 SpringBoot 框架的后端优势与 Vue+ElementUI 的前端特性，设计兼具稳定性与扩展性的系统架构，确保各模块低耦合、高内聚，便于后续功能迭代。
3. 实现核心功能模块：开发员工登录、个人信息管理、考勤打卡（含正常 / 迟到 / 早退状态判定）、请假申请与审批、会议通知发布等功能，同时支持管理员对员工、部门信息的维护及考勤数据统计。
4. 保障系统性能与安全：优化数据库设计（如合理设置索引、关联关系），确保系统在并发场景下响应迅速（页面加载≤2 秒，打卡操作≤1 秒）；通过权限控制、数据加密等手段，保护考勤数据的完整性与安全性。
5. 验证系统实用价值：通过功能测试、性能测试等手段，验证系统是否满足企业实际管理需求，是否能提升考勤管理效率、降低人工成本，为企业信息化建设提供可落地的解决方案。

总之，本文的研究目标是实现一个功能齐备、高性能、安全可靠的考勤管理系统，推动企业考勤管理从传统模式向信息化、规范化转型。

# **2**相关技术介绍

## **2.1**Java语言

Java是一种面向对象的编程语言，由Sun Microsystems于1995年推出，后被Oracle公司收购。它以“一次编写，到处运行”（Write Once, Run Anywhere）的跨平台特性闻名，广泛应用于企业级应用开发、移动应用开发、Web开发等领域。

Java的设计哲学强调代码的健壮性、安全性和可移植性。它摒弃了C++中的指针操作，引入了自动内存管理（垃圾回收机制），减少了内存泄漏的风险；通过字节码（Bytecode）机制实现跨平台，编译后的字节码可在任何安装了Java虚拟机（JVM）的设备上运行，无需重新编译。

Java具有丰富的类库和生态系统，标准库提供了网络通信、多线程、输入输出等基础功能，第三方库如Spring、Hibernate等进一步扩展了其应用场景。在面向对象编程方面，Java支持封装、继承、多态三大特性，通过接口（Interface）机制实现灵活的代码复用，适合构建复杂的大型应用。

在Web开发领域，Java的企业级框架如Spring、SpringBoot提供了完整的解决方案，支持依赖注入、AOP（面向切面编程）等技术，简化了企业级应用的开发流程。结合MySQL等数据库，Java能够构建高性能、高可靠性的业务系统，如考勤管理系统中员工信息的存储与处理、考勤数据的实时统计等。

Java的强类型特性和编译时类型检查，有助于在开发阶段发现潜在错误，提升代码质量。同时，其多线程支持使其能高效处理并发请求，适合应对考勤系统中员工同时打卡等场景的性能需求。

总之，Java是一种成熟、稳定的编程语言，其跨平台能力、丰富的生态和企业级支持，使其成为构建各类复杂应用的优选语言，尤其在需要高可靠性和可维护性的业务系统开发中表现突出。

## **2.2**Spring Boot框架

SpringBoot 是基于 Spring 框架的开源 Web 开发框架，由 Pivotal 团队开发，其核心设计理念是 “约定优于配置”，旨在简化 Spring 应用的初始搭建与开发流程。作为 Spring 生态的重要组成部分，它保留了 Spring 框架的核心特性，同时通过自动化配置和起步依赖等机制，大幅降低了开发者对配置文件的编写工作量，让开发人员能够更专注于业务逻辑的实现。

SpringBoot框架的核心特性体现在以下方面：

1. 自动配置机制：通过@EnableAutoConfiguration注解触发自动配置，框架会根据类路径下的依赖、配置信息等，自动生成所需的 Bean 并注入容器。例如，当检测到 MySQL 驱动依赖时，会自动配置数据源相关 Bean，无需手动编写 XML 配置。同时，支持通过@Conditional系列注解实现条件化配置，灵活应对不同场景。
2. 起步依赖（Starter）：将常用依赖打包成统一的依赖模块，开发者只需引入相应的 starter（如spring-boot-starter-web），即可自动获取该功能所需的所有依赖，无需手动管理版本兼容性。例如引入spring-boot-starter-web后，系统会自动集成 Spring MVC、Tomcat 服务器等 Web 开发必需组件。
3. 嵌入式服务器：内置 Tomcat、Jetty 等服务器，无需单独部署容器，可直接将应用打包为可执行 JAR 文件，通过java -jar命令启动，极大简化了部署流程。
4. 核心注解：@SpringBootApplication是 SpringBoot 应用的入口注解，整合了@Configuration（标记配置类）、@ComponentScan（自动扫描组件）和@EnableAutoConfiguration（启用自动配置）三大功能，简化了启动类的编写。
5. 生产级特性：内置Actuator组件，提供健康检查、信息监控等生产级功能，便于系统运维与问题排查。

## **2.3**Vue前端技术

### **2.3.1**Vue框架

Vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

组件化开发即是对某些可以进行复用的功能进行封装的标准化工作。组件一般会内含他的内部UI元素、样式和JS逻辑代码，它可以很方便的在应用的任何地方进行快速的嵌入。组件内部可以使用其他组件来构成更复杂的组件。

组件化特点：

（1）组件是对逻辑的封装，不限于图形元素

即我们可以把if做成组件，按一个倒计时做成组件，把一段动画做成组件，把数据架构做成组件，而这些并不能成为控件，这些组件在不同的系统中可以做完美的嵌入。

（2）高内聚、低耦合

前端的组件化开发，可以很大程度上降低系统各个功能的耦合性，数据相互独立，并且提高了功能内部的聚合性。这对前端工程化及降低代码的维护来说，是有很大的好处的，内部结构密封，不与全局或其他组件产生影响，特别是针对逻辑复杂的功能能够进行拆分，更好排查问题。

（3）具备单个可移植性

组件可以提供很好的提升代码的可重用性（而非可复制性），如果有其他项目需要该组件可以直接引入使用，而不是拷贝代码，拷贝资源等。

（4）极大提高开发效率

组件划分后，组件的开发不受其他业务影响，可以多个组件并行开发，加快开发进度。在多人团队里，每个人只负责自己的业务模块，他对业务功能的增删改查，都只限定在自己的这个业务模块里，不会影响其他人的业务，他代码质量的好坏也只会影响到自己的业务模块；如果有新人的加入，可以直接分配组件进行开发，而非需要熟悉整个项目，可以从一个组件的开发使新进人员比较快速熟悉项目、了解到开发规范；对测试来说，也十分方便，大部分情况下，我们只需要着重测试修改过的业务组件即可，而不用老是进行全部回归测试。

（5）功能重用的基石

业务组件类似一个个积木一样，我们可以用积木搭建出不同的房子，同理我们也可以创建多个不同的APP。我们只需要维护好每个组件，需要用到该组件的功能时，一建引用集成就可以了。

### **2.3.2**Element UI组件

Element UI 是一套基于 Vue.js 2.0 的桌面端组件库，由饿了么前端团队开发并开源。它提供了丰富的 UI 组件，能帮助开发者快速构建功能强大、风格统一的页面，在企业级应用开发中应用广泛。Element UI组件可分为基础组件，表单组件，数据展示组件、导航组件、通知组件等。

Element UI组件具有如下优势：

1. 专注桌面端体验：在交互设计、组件尺寸、响应速度等方面针对PC端深度优化，组件的间距、字体、色彩符合桌面应用视觉习惯。
2. 一致性与规范性：提供完整设计规范，从颜色、字体到组件尺寸等都有严格把控，确保项目界面风格统一。
3. 优雅与简洁：组件设计追求优雅简洁，注重提升学信息可读性和操作便捷性，减少用户认知负荷。
4. 易用性与可拓展性：拥有清晰详尽的中文文档和丰富示例代码，API设计直观；支持主题定制，可通过Sass变量覆盖进行深度定制，组件结构便于二次封装。
5. 国际化于无障碍设计：内置完善的国际化支持，注重无障碍设计，确保视障等特殊用户也能顺畅使用应用。

## **2.4**Node技术

Node.js是一种基于Chrome V8引擎的JavaScript运行时环境，可以让JavaScript代码在服务器端运行。它采用事件驱动、非阻塞I/O模型，使其在处理高并发请求时表现出色。Node.js使得开发人员能够使用JavaScript语言编写服务器端代码，实现前后端统一，简化了应用程序的开发和维护。

### **2.4.1**Node的优点

高效的I/O操作：Node.js采用非阻塞I/O模型，能够高效处理大量并发请求，适用于实时应用程序和数据密集型应用。

轻量和快速：Node.js采用事件驱动的设计，具有轻量级和快速启动的特点，适合构建高性能的网络应用。

跨平台：Node.js可在多种操作系统上运行，如Windows、Mac和Linux，具有良好的跨平台性。

强大的包管理工具：Node.js使用npm作为包管理工具，拥有丰富的开源模块和库，开发人员可以方便地复用和分享代码。

社区支持和活跃度高：Node.js拥有庞大的开发社区和活跃的贡献者，提供丰富的文档和资源，便于开发人员学习和解决问题。

Node.js广泛应用于Web开发领域，特别适用于构建实时应用程序、API服务、单页应用程序等。借助Node.js的框架如Express.js、Koa等，开发人员可以快速搭建稳健的服务器端应用。Node.js还支持与各种数据库系统的集成，如MongoDB、MySQL等，使得数据存储和处理变得更加便捷。

总的来说，Node.js是一种强大的服务器端运行时环境，具有高效、快速、跨平台等优势，为开发人员提供了丰富的工具和资源，助力他们构建高性能的网络应用。

## **2.5**MySQL数据库

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型和大型网站的开发都选择 MySQL 作为网站数据库。

# **3**需求分析

## **3.1**系统功能需求分析

### **3.1.1**业务需求分析

考勤管理系统作为企业人力资源管理的核心工具，需覆盖员工日常考勤全部流程及管理端的统筹功能，核心业务需求包括：

1. 考勤基础管理

实现员工上下班打卡的自动化记录，支持正常、迟到、早退、缺勤等状态的自动判定；关联请假流程，对已审批的请假时段自动豁免打卡要求，确保考勤数据与实际出勤情况一致

2. 申请和审批流程

员工可提交请假（事假、病假等）、固定资产使用等申请，填写申请时段、原因等关键信息；管理员或中层领导通过系统查看申请详情，执行同意或驳回操作，审批结果实时反馈给申请人，形成闭环管理。

3. 组织与人员管理

支持部门结构的创建、修改与删除，记录部门编号、名称及员工数量；对员工信息进行全生命周期管理，包括入职登记、岗位调整、信息更新及离职处理，关联员工所属部门与岗位信息。

4. 数据统计

管理员可查看员工考勤汇总数据（如迟到次数、请假天数）、部门人员分布及固定资产使用情况，生成考勤报表并导出，为薪酬核算与绩效评估提供数据支持。

总之，考勤管理系统的业务需求非常多且复杂，本文只列举了关键部分。为了实现一个好用、易用、高性能、安全可靠的考勤管理系统，还需要深入研究和了解各个业务领域的需求，以便提供更好的解决方案。

### **3.1.2**数据需求分析

数据需求分析是一份系统设计的重要环节，它主要是明确系统应该支持怎样的数据关联操作，主要涉及需求调研、数据收集与整理、数据审视与分类等方面。在本文中将围绕商城网站系统设计与实现来探讨数据需求分析的具体实现。

1. 人员与组织数据
2. 员工数据：包括工号、姓名、性别、出生日期、部门 ID、岗位 ID、联系方式、入职时间等，其中工号为唯一标识符，部门 ID 与部门表关联。
3. 部门数据：包含部门编号、名称、员工数量、岗位数量，部门编号唯一，通过员工表的部门 ID 字段建立多对一关联。
4. 考勤与申请数据
5. 考勤数据：记录员工 ID、打卡日期、上下班打卡时间及状态，员工 ID 关联员工表，打卡状态由系统根据预设时间规则（如上班时间 8:30 前）自动判定。
6. 请假数据：包括申请人工号、请假类型、起止时间、审批状态、审批人 ID 等，申请人工号关联员工表，审批状态通过任务表同步更新。
7. 辅助管理数据
8. 会议通知数据：含标题、时间、地点、参与者、发布者工号等，

发布者工号关联员工表，参与者字段记录多个员工工号。

1. 固定资产数据：记录资产名称、价格、使用人（员工工号）、审批状态等，使用人工号关联员工表，支持资产申请与审批流程追踪。

综上所述，数据需求分析在系统设计中极其重要，不光是为了明确数据关联、建立数据表和确定字段、外键等，更是为了确保系统能够正确处理数据并达到业务目标。

## **3.2**系统非功能性需求分析

### **3.2.1**系统处理能力需求

系统需要具备大规模、高并发的处理能力，在云服务部署的情况下，系统处理能力总需求为：

1. 响应速度：员工打卡、申请提交等核心操作响应时间≤1 秒，页面加载时间≤2 秒，确保高峰期（如早 8:00-9:00 打卡时段）系统流畅运行。
2. 并发能力：支持至少 1000 名员工同时在线操作，打卡、查询等请求并发处理无延迟或数据丢失 。

### **3.2.2**可靠性需求

软件可靠性是指软件产品在规定条件下和规定时间内完成规定功能的能力。 由于功能的持续能力具有概率统计特性， 可靠性也可以定义为产品在规定条件下和规定时间内完成规定功能的概率。为保证数据库检索工作的高效完成，系统软件和数据库需要有较高的可靠性：

1.后台服务系统持续性要求：7\*24小时不间断；

2.系统有效工作时间≥99%；

3.系统不间断运行30天，运行效率不会明显降低；

4.系统故障平均间隔时间≥300天。

### **3.2.3**可用性需求

软件可用性是对软件系统易学、易用、用户满意等方面的评价。系统用户设计管理人员、技术人员、社会人员等，人员素质参差不齐，需要软件具有较高的可用性：

1.系统人机交互界面友好，便于使用；

2.对于PC端业务处理软件，具有相关专业知识的用户在接收10个工作日以内的培训后，90%的用户能够独立使用。

### **3.2.4**维修性需求

软件的维修性是指软件产品在规定的条件下和规定的时间内， 按规定的程序和方法进行维修时，保持或恢复到规定状态的能力。在发生故障时，软件应具备较高的可维修性，保障普查工作顺利进行：

1.软件系统后台服务具备自动故障恢复功能，单个部署节点发生故障时，自动恢复时间<15分钟，手工恢复时间<2小时；

2.软件系统支撑各项功能的在线更新、迭代、升级和扩充，新功能开发测试完成后，更新集成时间≤5个工作日，且升级过程中，用户日常使用不受版本更新和维护的影响。

### **3.2.5**环境适配需求

软件系统环境适配包括对于硬件环境的适配和软件环境的适配。考虑到参与本系统的人员有自主可控软硬件环境的情况，软件的环境适应性包括：

1.能够在自主可控的CPU硬件设备中部署和运行，性能能够达到设计要求；

2.支持各种主流数据库，可以在主流数据库中正常运行并保持性能；

3.B/S架构软件具有良好的浏览器兼容性，支持Windows平台以及Linux操作系统下的各类主流浏览器。

# 4系统详细设计与实现

## **4.1** 普通员工模块主要功能实现

### **4.1.1** 登录

首先启动前后端项目，然后点击相应的链接可用浏览器打开，本系统为单位管理管理系统，所以需要登录之后才能打开功能页。

### **4.1.2** 个人中心

个人中心模块可以修改密码和签到打卡。用户修改密码时加上了表单验证功能，验证规则是密码不能为空且第二次输入的密码要和第一次一致。员工每天上下班需要打卡，签到状态分为正常、迟到和早退，如果没有打卡则视为缺勤，如果有请假则申请通过之后请假时间不需要打卡，打卡按钮在系统首页。员工每天上下班需要打卡，签到状态分为正常、迟到和早退，如果没有打卡则视为缺勤，如果有请假则申请通过之后请假时间不需要打卡，打卡按钮在系统首页。

## **4.2** 工作台模块主要功能实现

### **4.2.1** 客户管理

进入客户管理页面首先看到的是已经添加的客户信息，然后管理员就可对客户信息进行操作，查询出满足所有条件的客户。

### 4.2.2 会议通知管理

领导可以通过这个页面进行会议和通知的发布，选择类型会议或者通知，会出现不同的表单内容，然后填写表单信息点发布，表单的标题、时间等字段加上了数据验证，需要符合要求才会提交成功，要不然浏览器的控制台会打印出错误信息。员工可以查看公司会议和通知的详细记录，比如发布时间、发布人以及内容等等。

### **4.2.3** 任务审批

进入待办任务列表之后可以点击审批任务按钮，会有一个弹窗，弹窗里面有申请的详情，根据内容可以对该申请进行同意或者驳回，审批之后该任务会进入已办任务的列表，相应的申请也会改变其状态。

## **4.3**管理员模块主要功能实现

### **4.3.1**员工管理

工管理页面可以查看所有员工信息、新增员工、修改员工信息、删除员工和根据条件检索。

### 4.3.2 部门管理

部门管理界面可以新增部门、修改除部门人数之外的信息、可以删除人数为0的部门，还可以查看所有部门的信息。

# 5系统测试

## 5.1 系统测试的目的

为保证系统能安全稳定的运行，系统测试是不可或缺的。系统测试是为了测试出系统存在或潜在的错误然后再改正，验证系统的功能性、性能、稳定性和安全性，以确保系统在实际使用中能够正常运行，满足用户需求，并能够应对各种压力和异常情况。

## 5.2 系统测试的方法及步骤

系统的测试选择单元测试、集成测试和系统测试三种方法。

(1)单元测试：单元测试是针对某个模块来进行检验的测试，主要是检验程序内部的逻辑问题。

(2)集成测试：集成测试又叫作组装测试，集成测试当所有模块的单元测试都完成后，再按照系统的设计要求将所有模块组装完成后进行的测试。集成测试主要是检查各模块与其它模块之间的对接过程可能存在的错误。

(3)系统测试：系统测试是将经过集成测试的所有子系统组装成一个完整的系统来进行测试。它是按照系统的需求分析来选取测试用例，并在运行环境中进行测试，以检测系统是否能正常运行且满足先前的需求。

## 5.3测试环境

Java运行环境：JDK1.8

数据库：MySQL

操作系统：Windows7及以上

# 6总结与展望

本课题基于 SpringBoot + Vue 框架设计并实现了一套考勤管理系统，旨在解决传统纸质考勤效率低下、易出错、难统计等问题，推动单位考勤管理的信息化转型。系统采用前后端分离架构，后端通过 SpringBoot 框架实现业务逻辑与数据交互，前端利用 Vue 结合 ElementUI 构建用户界面，配合 MySQL 数据库完成数据存储，形成了功能完整、操作便捷的管理平台。

从功能实现来看，系统覆盖了两类用户角色的核心需求：普通员工可完成登录、签到打卡、请假申请、密码修改等操作；管理员（中层领导）具备员工管理、部门维护、请假审批、会议通知发布等权限，同时支持客户信息管理与考勤数据统计。通过自动判定打卡状态（正常 / 迟到 / 早退）、关联请假流程豁免打卡等机制，确保了考勤数据的准确性与逻辑性。

系统的主要特点包括：界面设计简洁直观，操作流程简化，降低了用户学习成本；功能模块划分清晰，涵盖考勤全流程，满足单位日常管理需求；数据库表结构设计合理，通过外键关联保证数据一致性；采用成熟框架开发，代码可维护性与扩展性较强。

然而，系统仍存在明显不足：部分功能存在冗余且细节完善度不高，存在逻辑矛盾；安全机制较为基础，对敏感数据的保护不足；整体功能偏于简单，难以应对复杂场景的考勤管理需求。

参考文献

[1]何为,迟文恒. 服务端研发知识图谱[M].机械工业出版社:信息科学与技术丛书, 201812.430.

[2]杨开振. 深入浅出Django 2.x[M].人民邮电出版社:, 201808.442.

[3]古凌岚,张婵,罗佳. 计算机系统化项目开发教程[M].人民邮电出版社:, 201802.313.

[4]肖睿,喻晓路,朱微,张超,戴立坤. Web应用设计及实战[M].人民邮电出版社:大数据开发实战系列, 201801.238.

[5]李宗花,朱林. 软件工程原理与实践[M].南京大学出版社:, 202008.252.

[6]陆惠恩,褚秋砚. 软件工程[M].人民邮电出版社:, 201712.284.

[7]肖睿,郭泰,王丁磊. SSH框架企业级应用实战[M].人民邮电出版社:大数据开发实战系列, 201801.344.

[8]付森,石亮,吴起立,刘冰. MySQL开发与实践[M].人民邮电出版社:, 201408.261.

[9]陈陆扬. Vue.js前端开发快速入门与专业应用[M].人民邮电出版社:, 201702.207.

[10]陈晓兵. Web前端开发技术[M].南京大学出版社:, 202008.252.

[11]杨洋,刘全. 软件系统分析与体系结构设计[M].南京东南大学出版社:, 201710.207.

[12]王昊,刘友华. 信息系统分析与设计[M].南京大学出版社:, 202109.323.

[13]张永奎. 数据库原理与设计[M].人民邮电出版社:, 201909.260.

[14]佟伟光,郭霏霏. 软件测试[M].人民邮电出版社:, 201501.281.

[15]郑钢,贺亚涛,尤胜涛. 大型网站服务器容量规划[M].人民邮电出版社:201608.204.

[16]姚庆涛.浅析计算机网络应用安全[J].网络安全技术与应用,2023(05):169-171.

[17]熊威.python编程基础课程类和对象的教学设计[J/OL].中国教育技术装备:1-4

[18]齐研科,杨颖,贺喜,蔡雨耕,赵家骏,王一先. 软件开发系统及方法[P]. 重庆市：2023-05-02.

[19]朱若榕.人机交互软件界面设计的重要性[J].文化产业,2023(12):153-155.

[20]刘德山,金百东. 设计模式深入研究[M].人民邮电出版社:, 201407.222.