**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于jQuery的公司人员管理系统的分析与设计 | | |
| 课 程 | 软件工程 | | |
| 学 院 | 信息工程学院 | | |
| 专 业 | 软件技术 | 年级 | 2023级 |
| 学生姓名 | 瞿萌 | 学号 | 012322100047 |

**2024年12月13号**

**绪论**

随着公司的不断发展壮大，人员管理的复杂性日益增加。为了提高管理效率、优化人力资源配置，开发一个公司人员管理系统具有重要的现实意义。本可行性研究报告将对该系统的技术可行性、经济可行性和操作可行性进行全面分析。

**1.1系统目标**

- 实现对公司员工信息的集中管理，包括个人基本信息、工作履历、培训记录等。

- 提高人事管理流程的自动化程度，如招聘、入职、离职、调动等。

- 提供数据分析功能，为公司的人力资源决策提供支持。

**1.2系统功能**

- 员工信息管理：录入、查询、修改和删除员工信息。

- 人事流程管理：处理招聘、入职、离职、调动等流程。

- 考勤管理：记录员工的考勤情况，生成考勤报表。

- 绩效管理：设定绩效指标，进行绩效评估和反馈。

- 培训管理：安排培训课程，记录员工的培训情况。

- 报表生成：生成各种人力资源报表，如员工花名册、考勤报表、绩效报表等。

**1.3硬件需求**

1. 技术架构

- 系统可采用 B/S 架构，方便用户通过浏览器访问。

- 后端可使用主流的编程语言和数据库管理系统，如Java、Python 搭配 MySQL、Oracle 等。

2. 技术成熟度

- 相关技术已经非常成熟，有大量的成功案例可供参考。

- 开发团队可以利用现有的开发框架和工具，提高开发效率。

3. 技术风险

- 数据安全是一个重要的风险点，需要采取有效的安全措施，如数据加密、访问控制等。

- 系统的性能和稳定性也需要进行充分的测试和优化。

**2 需求分析**

### **2.1系统功能需求**

本系统主要针对的是企业员工的信息管理项目而设计。在项目的确立之初，通过与企业管理人员的相互交流，发现了目前所用系统中所存在的一些问题，并对其进行研究，根据企业对员工信息管理方面的需求，做出了以下几方面的需求分析。

\*员工入职流程

1. 招聘部门确定录用人员后，将新员工基本信息（姓名、性别、联系方式、学历等）传递给人力资源部门。

2. 人力资源专员在系统中录入新员工信息，包括分配部门、职位，并设定初始薪资、考勤等相关参数。

3. 为新员工创建系统账号和初始密码，告知新员工登录信息。

\*考勤管理流程

1. 员工通过打卡设备（如指纹打卡机、手机 APP 打卡等）进行日常考勤打卡，打卡数据实时传输至人员管理系统。

2. 系统根据预设的上班时间、下班时间和排班规则，自动判断员工的考勤状态（正常、迟到、早退、旷工等）。

3. 对于请假、出差等特殊考勤情况，员工通过系统提交申请，经上级领导审批后，系统更新考勤记录。

4. 每月末，考勤数据自动汇总生成考勤报表，供人力资源部门和部门领导查看和分析。

\*薪酬管理流程

1. 根据员工的职位、工作时长（结合考勤数据）、绩效评估结果等因素，系统按照预设的薪资计算公式，计算员工的应发工资。

2. 从应发工资中扣除社保、公积金、个人所得税等项目，得出实发工资。

3. 财务部门依据系统生成的工资单进行工资发放，并将发放记录反馈至系统。

\*绩效评估流程

1. 人力资源部门根据公司战略和各部门目标，制定绩效评估指标和标准，并将其分配到各个职位。

2. 部门领导在评估周期内（如季度、年度），根据员工的工作表现，在系统中对下属员工进行绩效评分，填写评价意见。

3. 系统根据绩效评分和评价意见生成绩效评估报告，供人力资源部门分析和决策，作为员工晋升、调薪、培训等的依据。

\*员工离职流程

1. 员工提出离职申请，经上级领导和人力资源部门审批同意后，启动离职流程。

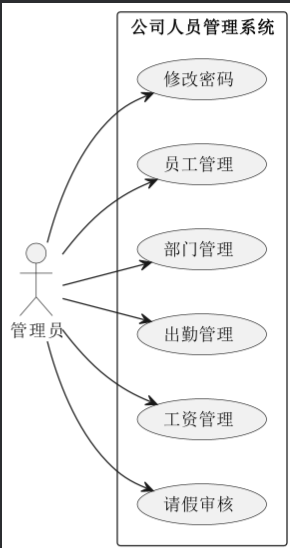
2. 系统冻结员工账号，停止考勤记录等相关功能。

3. 人力资源专员在系统中记录离职原因、离职日期等信息，并办理工资结算、社保公积金停缴等手续。

**2.2系统模型**

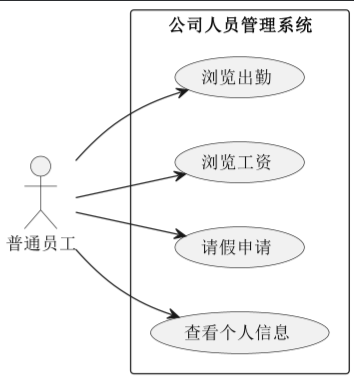
#### **2.2.2管理员用例图**

管理员是系统的管理者，其中包括六大功能模块，管理员拥有系统的最高权限，可以操作部门、员工、工资等信息，对所有的信息进行更改，适应公司的变化。所示：

****

#### **2.2.3普通员工用例图**

出于安全性的考虑，普通员工用户只有请假申请、浏览工资和浏览出勤等功能，其他的删除修改功能都没有设计。普通员工用例图如图所示：



**3 概要分析**

**3.1系统设计原则**

设计思想遵循以下几点：

\*采用B/S模式进行开发，是WEB兴起后的一种网络结构模式，其优点是后台与前台处理层次分明，大多数用户都已经习惯了传统的网页模式，不需要进行更大的改动就能够适应该系统。对员工的出勤、工资等信息进行电脑化管理，方便企业的一体化管理。

\*采用面向对象的开发与设计理念。可以进行一抽象数据类型为特点的基于对象的程序设计，他的前提是对系统的整体有一个全面的了解，通过他可以保证系统的优良，来保证开发出的系统具有良好的稳定性与实用性。

\*采用模块化设计。是将系统分为若干个功能模块，逐个实现系统的功能，有利于软件的开发、测试以及维护等功能，加强了系统的稳定性以及后期的维护过程。

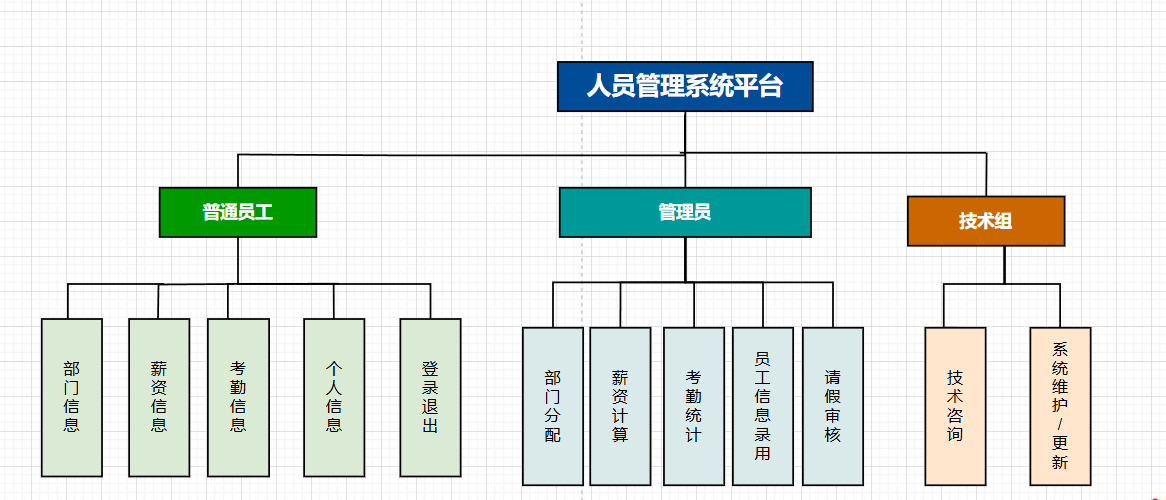
简单方便的系统界面。

\*设计简单友好的系统界面，方便用户较快的适应系统的操作。

\*速度优先原则。在开发过程中由于实际的需要，所开发出来的系统要尽量做到简洁，一方面利于测试人员的工作，最主要的是要是整个系统所占的资源达到最少，运行速度尽可能的达到最快，来实现企业快速发展的需求。

设计既要突出重点，又要细致周到。要符合企业的基本需求，在有缺陷的地方同企业进行商量并改进，使系统能够完全实现用户的需求，并尽可能的实现在后期添加功能的需求。

**3.2系统总体结构**

****

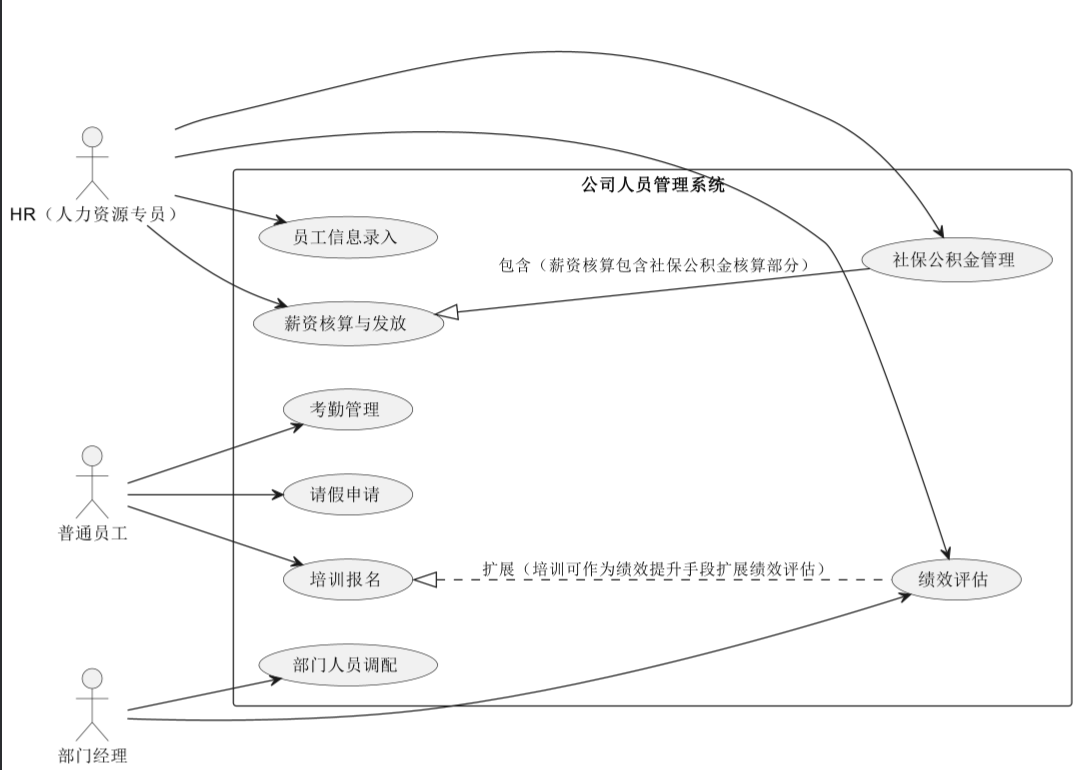
登录界面：



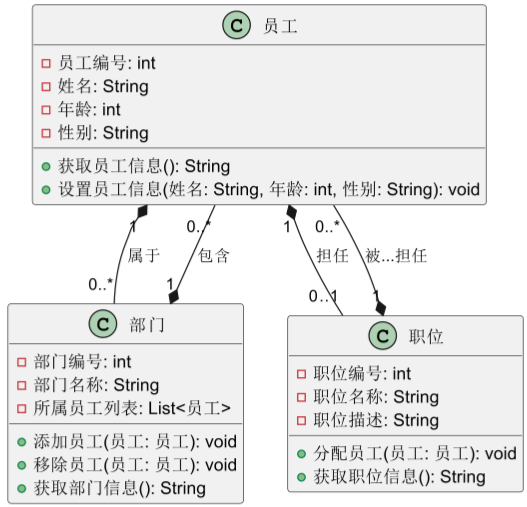
注册界面：



用例图设计：

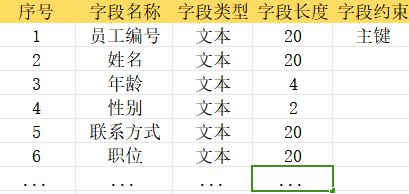
****

类图设计：

****

**3.3数据库设计**

\*员工表



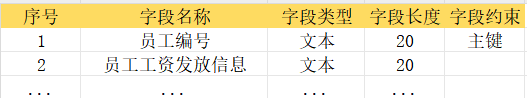
\*考勤表



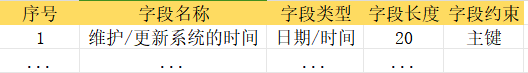
\*绩效表



\*酬薪表



\*技术管理表



\*部门表

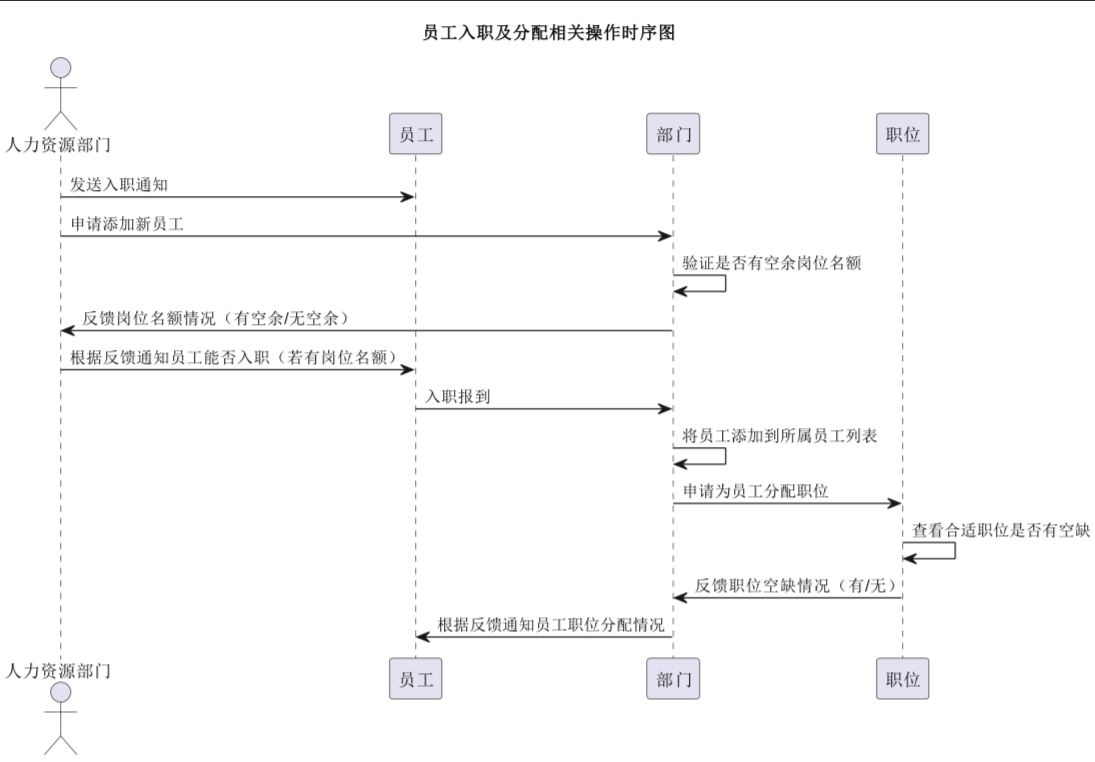


**4 用例及界面详细设计**

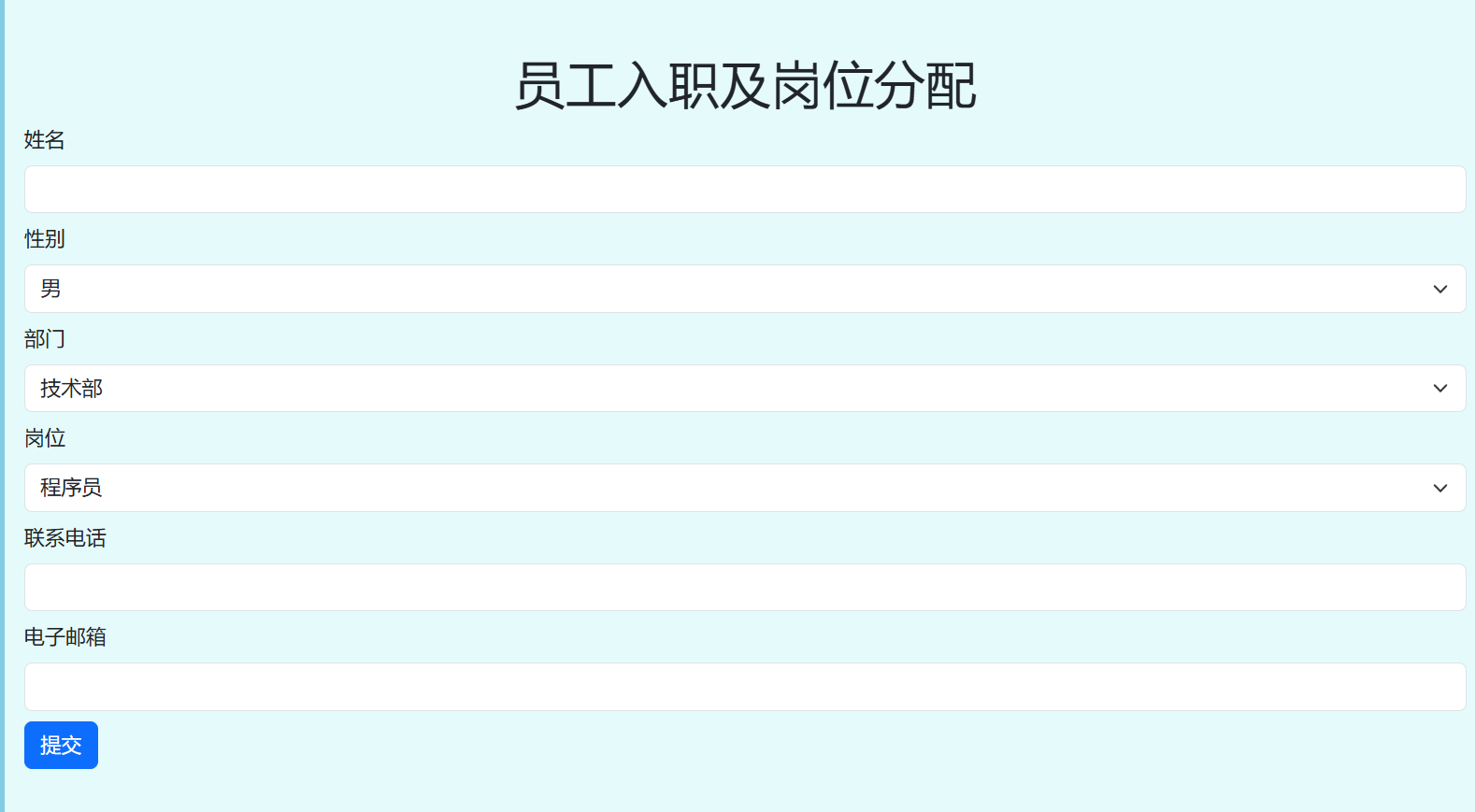
**4.1.1员工入职及分配用例规约**

| **用例名称** | **员工入职及分配** |
| --- | --- |
| 功能 | 添加用户 |
| 参与者 | 本公司人员 |
| 前置条件 | 申请添加新员工 |
| 后置条件 | 反馈岗位名额情况（有/无空余） |
| 主事件流 | 1.添加新用户 2.用户填写新密码（两次输入）  3.系统存储该员工 |
| 备选事件流 | 1.用户两次输入的密码不同   2.系统提示两次输入的密码不相同    3.用户重新填写并提交。 |

**4.2.1时序图**

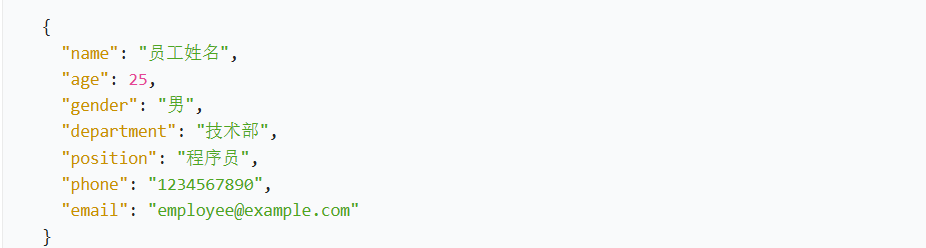
****

**4.3.1界面设计**



**4.4.1接口设计  
**（一）员工入职接口（**/api/employee/onboarding**）、****

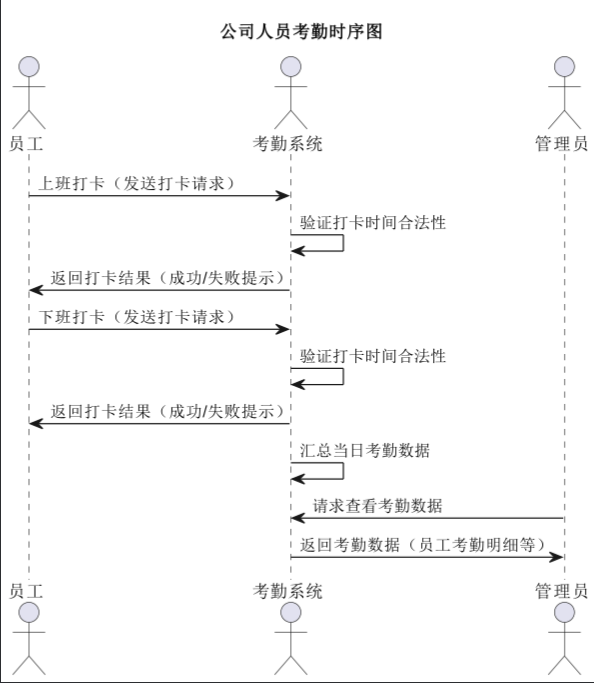
****用例：员工入职及分配。****

* **功能：添加新员工并且分配岗位。**
* **权限：普通员工。**
* **请求方法：POST**
* ****请求头****：
  + Content-Type: application/json
* ****请求体参数（JSON 格式）****：
* 
* ****响应数据格式（JSON）****：
  + 成功：{ "success": true, "message": "员工入职及岗位分配成功" }
  + 失败：{ "success": false, "message": "入职信息有误或其他错误原因" }

**4.1.2考勤用例规约**

| **用例名称** | **考勤** |
| --- | --- |
| 功能 | 记录员工上下班情况 |
| 参与者 | 本公司员工 |
| 前置条件 | 必须先登录 |
| 后置条件 |  |
| 主事件流 | 1.用户填写工号 2.用户提交 3.系统存储考勤时间 |
| 备选事件流 | 1.工号不能为空 |

**4.2.2时序图**

****

**4.3.2考勤界面**



**4.4.2接口设计**

1. ****上班打卡接口（**/api/clock-in**）****

****用例：考勤。****

* **功能：员工上下班打卡。**
* **权限：全体员工。**
* ****请求方法****：POST
* ****响应数据格式（JSON）****：
  + 成功：{ "success": true, "message": "上班打卡成功" }
  + 失败：{ "success": false, "message": "上班打卡失败，可能原因：重复打卡等" }

****（二）下班打卡接口（**/api/clock-out**）****

* ****请求方法****：POST
* ****响应数据格式（JSON）****：
  + 成功：{ "success": true, "message": "下班打卡成功" }
  + 失败：{ "success": false, "message": "下班打卡失败，可能原因：未上班打卡等" }

****（三）获取考勤记录接口（**/api/attendance-records**）****

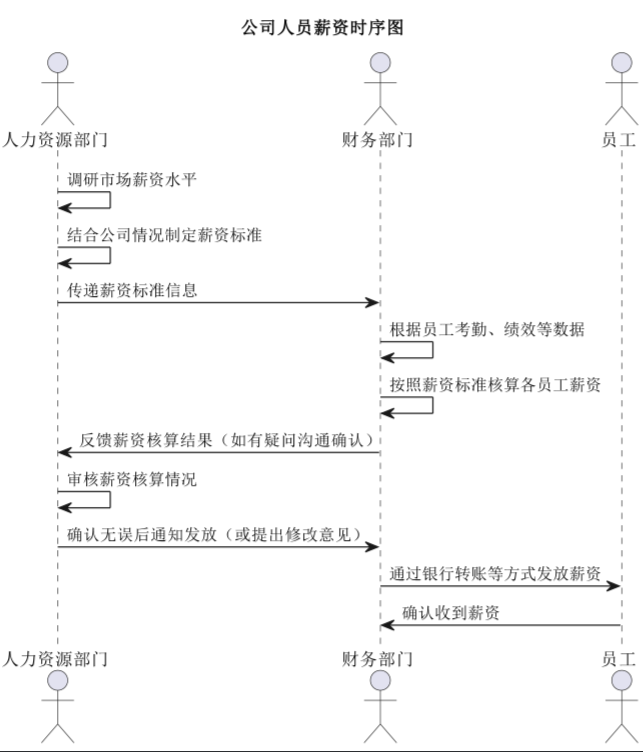
* ****请求方法****：GET
* ****响应数据格式（JSON）****：



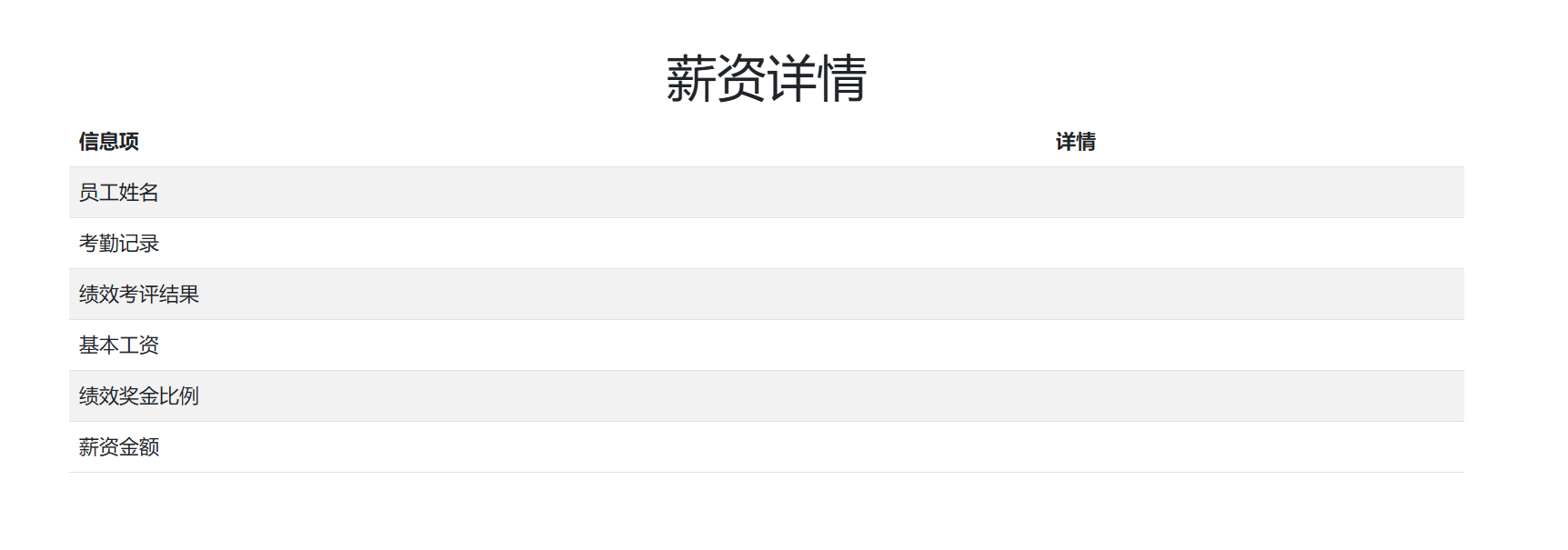
**4.1.3薪资用例规约**

| **用例名称** | **薪资** |
| --- | --- |
| 功能 | 计算该员工本月薪资 |
| 参与者 | 本公司员工 |
| 前置条件 | 必须先登录 |
| 后置条件 |  |
| 主事件流 | 1.根据市场薪资水平，结合公司情况制定薪资 2.根据员工考勤，绩效等数据核算薪资 3.确定发放途径 |
| 备选事件流 | 1.是否存在疑问 |

**4.2.3时序图**

****

**4.3.3薪资界面**

****

**4.4.3接口设计**

1. ****获取薪资记录接口（**/api/attendance**）****

****用例：薪资。****

* **功能：计算员工薪资。**
* **权限：全体员工。**
* **请求方法：GET**
* ****响应数据格式（JSON）****：{ "success": true, "attendance": [ { "date": "2024-12-01", "status": "正常" }, { "date": "2024-12-05", "status": "迟到" }] }

****（二）获取绩效考评结果接口（**/api/performance**）****

* ****请求方法****：GET
* ****响应数据格式（JSON）****：{ "success": true, "score": 80 }

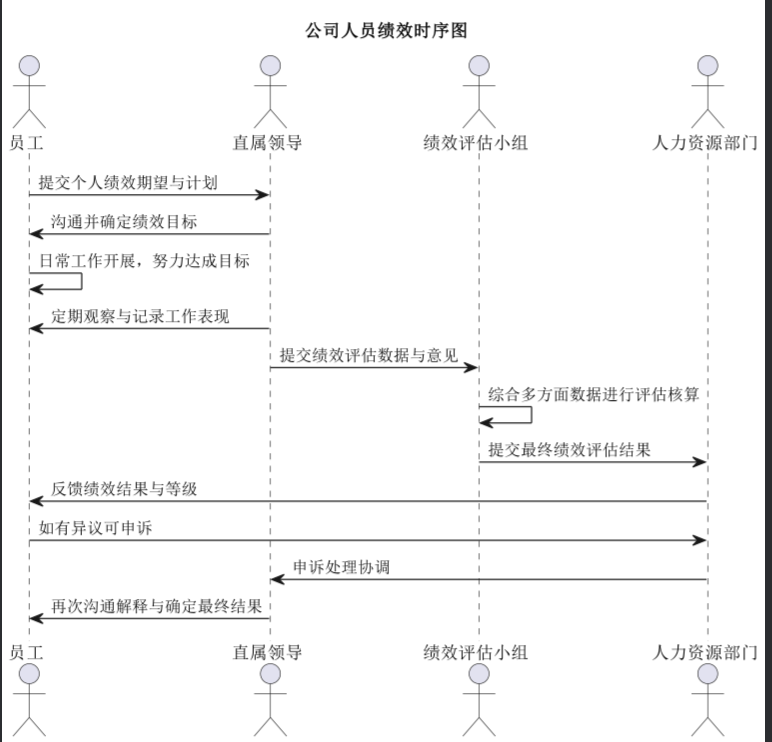
****（三）获取薪资金额接口（**/api/salary**）****

* ****请求方法****：GET
* ****响应数据格式（JSON）****：{ "success": true, "amount": 5000 }

**4.1.4绩效用例规约**

| **用例名称** | **绩效** |
| --- | --- |
| 功能 | 查看该员工本月绩效 |
| 参与者 | 本公司员工 |
| 前置条件 | 必须先登录 |
| 后置条件 |  |
| 主事件流 | 1. 计算该员工本月绩效 2.交给上级审查 3.反馈最后结果 |
| 备选事件流 | 1.是否存在疑问，申诉需求 |

**4.2.4时序图**

****

**4.3.4绩效界面**

****

**4.4.4接口设计**

1. ****获取绩效详情接口（**/api/performance-details**）****

****用例：绩效。****

* **功能：计算员工绩效。**
* **权限：全体员工。**
* **请求方法：GET**
* ****响应数据格式（JSON）****：

