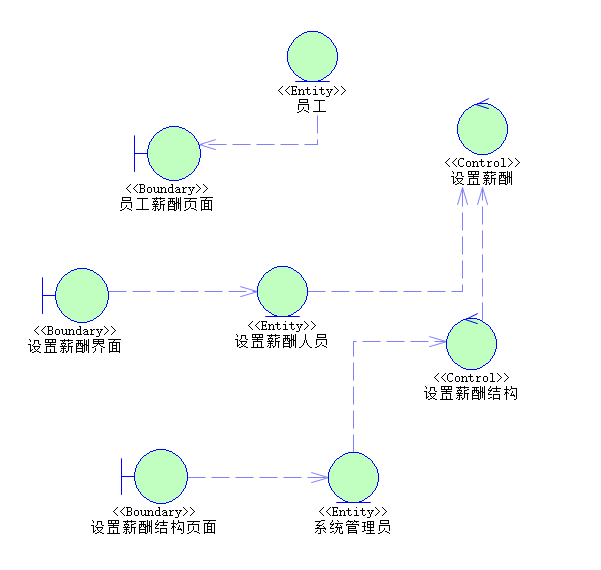
**一、概念架构**

1. 初步设计（鲁棒分析图）

企业薪酬管理系统是一款用于管理企业员工薪资、社保、公积金等信息的系统。该系统需要支持员工信息管理、薪资计算、社保公积金管理、报表生成等功能。该系统的主要目的是帮助企业管理人力资源，提高工作效率和准确性。



2. 高层分割

该系统可以分为以下几个模块：

（1）员工信息管理模块：该模块负责员工信息的录入、修改、查询等操作。

（2）薪资计算模块：该模块负责根据员工的工资、绩效等计算出员工的实际薪资。

（3）社保公积金管理模块：该模块负责管理员工的社保、公积金等信息。

（4）报表生成模块：该模块负责生成各种报表，如薪资报表、社保公积金报表等。

3. 考虑非功能需求

该系统需要具备高可用性、高性能、安全性、易用性等非功能需求。系统需要支持多用户同时访问，并且能够处理大量数据。同时，系统需要保证数据的安全性，防止数据泄露和篡改。系统还需要具备良好的用户体验，方便用户操作。

**二、细化架构**

1. 逻辑架构

a. 模块划分：

员工管理模块：负责管理员工基本信息、职位信息和薪资等级。

薪资计算模块：根据员工的基本信息、出勤记录和绩效评估，计算员工的薪资。

报表生成模块：生成各类薪酬相关的报表，如工资单、社保报告等。

b.数据流程：

员工管理模块接收员工信息的输入，并存储到数据库中。

薪资计算模块从数据库中获取员工信息、出勤记录和绩效评估数据，并进行薪 资计算。

报表生成模块从数据库中获取计算好的薪酬数据，并生成相应的报表。

c.数据模型：

数据库中包含员工表（Employee）存储员工的基本信息，薪资等级表（SalaryGrade）存储薪资等级信息，出勤记录表（Attendance）存储员工的出勤记录，绩效评估表（Performance）存储员工的绩效评估数据等。

d.组件和接口：

员工管理模块包括员工信息管理的组件，提供添加、修改和查询员工信息的接口。

薪资计算模块包括薪资计算的组件，接收员工信息和相关数据，并计算薪资。

报表生成模块包括报表生成的组件，接收计算好的薪酬数据，并生成报表。

e.安全性和权限控制：

系统实现用户身份验证，确保只有授权用户才能访问系统。

针对敏感数据和功能，设置访问权限控制，限制用户对特定数据和功能的访问权限。

f.扩展性和可维护性：

每个模块的组件应该是松耦合的，便于增加新功能或替换现有组件。

接口应该定义清晰，文档化，便于其他团队成员理解和使用。

系统应该具备可配置性，方便根据业务需求进行灵活的配置和调整。

2. 物理架构

物理架构是指在计算机系统中，描述系统硬件和网络组件的布局、连接方式和部署方式的一种表示。它关注的是系统的物理组成和互联方式，与逻辑架构相对应。

1. 服务器和存储设备：

- 员工管理模块、薪资计算模块和报表生成模块可以运行在独立的服务器上，分别部署为独立的服务。

- 数据库服务器用于存储员工信息、出勤记录、绩效评估数据和薪酬计算结果等数据。

2. 网络组件：

- 内部网络：所有服务器和数据库服务器可以连接到内部局域网（LAN），以实现内部通信和数据传输。

- 外部网络：系统可以与外部网络（如互联网）建立连接，以便员工和管理员通过外部网络访问系统。

3. 客户端设备：

- 员工可以通过桌面电脑、笔记本电脑、平板电脑或手机等设备通过外部网络访问系统的前端界面。

- 管理员可以使用专门的管理工作站或电脑通过内部网络访问系统的管理界面。

4. 安全性和冗余性：

- 网络安全设备（如防火墙、入侵检测系统）可用于保护系统免受恶意攻击。

- 数据备份和恢复策略可以通过备份服务器或其他备份媒介来确保数据的安全性和可靠性。

5. 可伸缩性和负载均衡：

- 如果系统需要支持大量用户或大量数据处理，可以考虑使用负载均衡器来平衡服务器的负载。

- 可以使用集群或分布式系统来实现系统的可伸缩性，通过水平扩展来应对不断增长的需求。

需要注意的是，物理架构的设计涉及到具体的硬件设备选择、网络拓扑和安全策略等方面，应根据实际情况进行定制和调整。在设计物理架构时，还需要考虑系统的性能要求、可用性需求、成本效益和未来的扩展计划。

3. 运行架构

1. 前端应用：

- 使用Web技术开发的前端应用可以在员工的客户端设备上运行，如桌面电脑、笔记本电脑、平板电脑或手机。

- 前端应用负责与用户交互，包括员工的登录、数据输入、界面展示等。

2. 后端服务：

- 员工管理模块、薪资计算模块和报表生成模块可以作为独立的后端服务运行。

- 后端服务负责处理业务逻辑、数据计算和数据库交互等操作。

- 后端服务可以使用常见的服务器端编程语言和框架进行开发和部署。

3. 数据库：

- 数据库用于存储员工信息、出勤记录、绩效评估数据和薪酬计算结果等数据。

- 可以选择关系型数据库（如MySQL、Oracle）或非关系型数据库（如MongoDB）作为数据存储解决方案。

4. 通信协议和接口：

- 前端应用和后端服务之间通过HTTP或HTTPS协议进行通信。

- 前端应用可以通过API接口与后端服务进行数据交换和调用后端功能。

5. 云服务或服务器：

- 可以选择将系统部署在云平台（如AWS、Azure、Google Cloud）上，利用云服务提供的计算资源和扩展能力。

- 或者可以自行搭建服务器环境，将系统部署在自己的服务器上。

6. 安全性和监控：

- 系统应该实施适当的安全措施，如数据加密、用户身份验证和访问控制，以保护系统和数据的安全。

- 可以使用日志记录和监控工具来监视系统运行状况、检测异常和进行故障排查。

4. 开发架构

该系统的开发架构采用MVC架构，将表示层、应用层、业务逻辑层、数据访问层分离，以提高系统的可维护性和可扩展性。同时，系统需要采用面向对象编程技术，以提高代码的重用性和可读性。

（1）MVC架构：该系统采用MVC架构，将表示层、应用层、业务逻辑层、数据访问层分离，以提高系统的可维护性和可扩展性。

（2）面向对象编程：系统需要采用面向对象编程技术，以提高代码的重用性和可读性。该技术使用Java语言实现。

5. 数据架构

该系统的数据架构采用关系型数据库，以支持数据的高效存储和查询。系统需要设计合理的数据表结构，并采用索引、分区等技术提高数据的查询效率。同时，系统需要采用备份、恢复等技术保证数据的安全性和可靠性。

（1）关系型数据库：该系统采用MySQL关系型数据库，以支持数据的高效存储和查询。

（2）数据表结构设计：系统需要设计合理的数据表结构，以支持系统的各种功能。

（3）索引、分区等技术：系统需要采用索引、分区等。