课题名称 软件开发实践

学 院 计算机与电子信息学院 专 业 计算机科学与技术

指导老师 黄汝维

**目录**

[课题名称 软件开发实践 1](#_Toc154786703)

**[一、 需求分析](#_Toc154786704)** [1](#_Toc154786704)

**[1.系统功能结构图](#_Toc154786705)** [1](#_Toc154786705)

**[2.](#_Toc154786706)****[用况图和扩展用况](#_Toc154786706)** [1](#_Toc154786706)

**[二、 系统分析与设计](#_Toc154786707)** [10](#_Toc154786707)

**[1.概念模型](#_Toc154786708)** [10](#_Toc154786708)

**[2.系统顺序图](#_Toc154786709)** [11](#_Toc154786709)

**[3.系统操作](#_Toc154786710)** [17](#_Toc154786710)

**[4.契约](#_Toc154786711)** [17](#_Toc154786711)

**[5.术语表](#_Toc154786712)** [19](#_Toc154786712)

**[6.协作图](#_Toc154786713)** [20](#_Toc154786713)

**[7.类图](#_Toc154786714)** [20](#_Toc154786714)

**[8.数据库设计](#_Toc154786715)** [20](#_Toc154786715)

**[三、 系统实现](#_Toc154786716)** [20](#_Toc154786716)

**[四、 课设总结（暂时不写）](#_Toc154786717)** [20](#_Toc154786717)

**[附录 类代码和主函数（需写出来）](#_Toc154786718)** [21](#_Toc154786718)

1. **需求分析**

**1.1系统功能结构图**

（1）系统功能模块

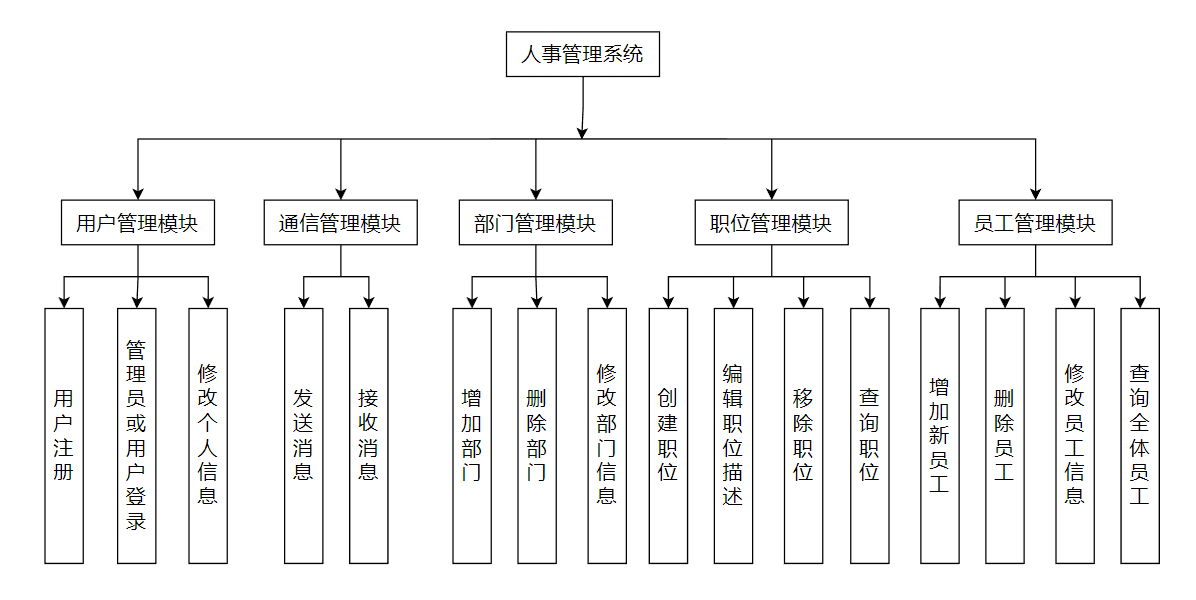


图4-1 系统功能结构图

（2）功能编号

表4-1 功能编号表

|  |  |  |
| --- | --- | --- |
| 功能模块 | 功能编号 | 功能 |
| **用户与员工管理** | FU1 | 用户注册 |
| FU2 | 用户\管理员登录 |
| FU3 | 修改个人信息 |
| FR1 | 增加新员工 |
| FR2 | 删除员工 |
| FR3 | 修改员工个人信息 |
| FR4 | 查询全体员工 |
| **部门管理** | FD1 | 增加部门 |
| FD2 | 删除部门 |
| FD3 | 修改部门信息 |
| **职位管理** | FP1 | 创建职位 |
| FP2 | 编辑职位信息描述 |
| FP3 | 移除职位 |
| FP4 | 查询职位 |
| **通信管理** | FS1 | 发送消息 |
| FS2 | 接收消息 |

## 1.2功能描述

**（1） 用户与员工管理模块**

**FU1 用户注册**

|  |  |  |
| --- | --- | --- |
| **功能名称：用户注册** | **使用单位：全体员工** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 用户工号、密码、姓名、联系方式 | 员工注册并修改初始密码 | 用户登录凭证 |

**FU2 用户\管理员登录**

|  |  |  |
| --- | --- | --- |
| **功能名称：用户\管理员登录** | **使用单位：全体员工、管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 账号和密码 | 系统验证身份信息 | 访问权限和概览 |

FU3 修改个人**信息**

|  |  |  |
| --- | --- | --- |
| **功能名称：修改个人信息** |  |  |
| 输入部分I |  | 输出部分O |
| 更新后的姓de名、联系方式、等个人信息 |  | 更新后的用户个人信息 |

**FR1 增加新员工**

|  |  |  |
| --- | --- | --- |
| **功能名称：增加新员工** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 新员工信息 | 管理员将新员工添加到系统中 | 新员工的在职记录 |

**FR2 删除员工**

|  |  |  |
| --- | --- | --- |
| **功能名称：删除员工** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 员工标识信息 | 管理员删除员工的在职记录 | 更新后的员工名册 |

**FR3 修改员工个人信息**

|  |  |  |
| --- | --- | --- |
| **功能名称：修改员工个人信息** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 员工工号、更新信息 | 管理员根据需要对员工信息进行修改 | 更新后的员工资料 |

**FR4 查询全体员工**

|  |  |  |
| --- | --- | --- |
| **功能名称：查询全体员工** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 查询请求 | 管理员查询员工记录 | 满足条件的员工列表 |

**（2） 部门管理模块**

**FD1 添加部门**

|  |  |  |
| --- | --- | --- |
| **功能名称：增加部门** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 部门信息 | 管理员创建新的部门 | 新创建的部门信息 |

**FD2 删除部门**

|  |  |  |
| --- | --- | --- |
| **功能名称：删除部门** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 部门标识信息 | 管理员删除已有的部门 | 删除后的部门列表 |

**FD3 修改部门信息**

|  |  |  |
| --- | --- | --- |
| **功能名称：修改部门信息** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 修改需求、部门信息 | 管理员对部门资料进行更新 | 更新后的部门信息 |

**（3） 职位管理模块**

**FP1 创建职位**

|  |  |  |
| --- | --- | --- |
| **功能名称：创建职位** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 职位描述信息 | 管理员创建一个新的职位 | 创建的新职位信息 |

**FP2 编辑职位信息描述**

|  |  |  |
| --- | --- | --- |
| **功能名称：编辑职位描述** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 职位信息、新的职位描述 | 管理员修改职位的详细信息 | 更新后的职位信息 |

**FP3 移除职位**

|  |  |  |
| --- | --- | --- |
| **功能名称：移除职位** | **使用单位：管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 职位标识信息 | 管理员从系统中删除职位 | 删除后的职位列表 |

**FP4 查询职位**

|  |  |  |
| --- | --- | --- |
| **功能名称：查询职位** | **使用单位：全体员工、管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 查询请求 | 查询指定或模糊匹配的职位信息 | 职位列表 |

**（4） 通信管理模块**

**FS1 发送消息**

|  |  |  |
| --- | --- | --- |
| **功能名称：发送消息** | **使用单位：全体员工、管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 编辑要发送的消息，添加附件 | 用户发送消息 | 发送消息记录 |

**FS2 接收消息**

|  |  |  |
| --- | --- | --- |
| **功能名称：接收消息** | **使用单位：全体员工、管理员** |  |
| 输入部分I | 处理描述P | 输出部分O |
| 查看消息 | 用户接收消息 | 接收消息记录 |

# 1.2 用况

## 1.2.1 系统总体用况图

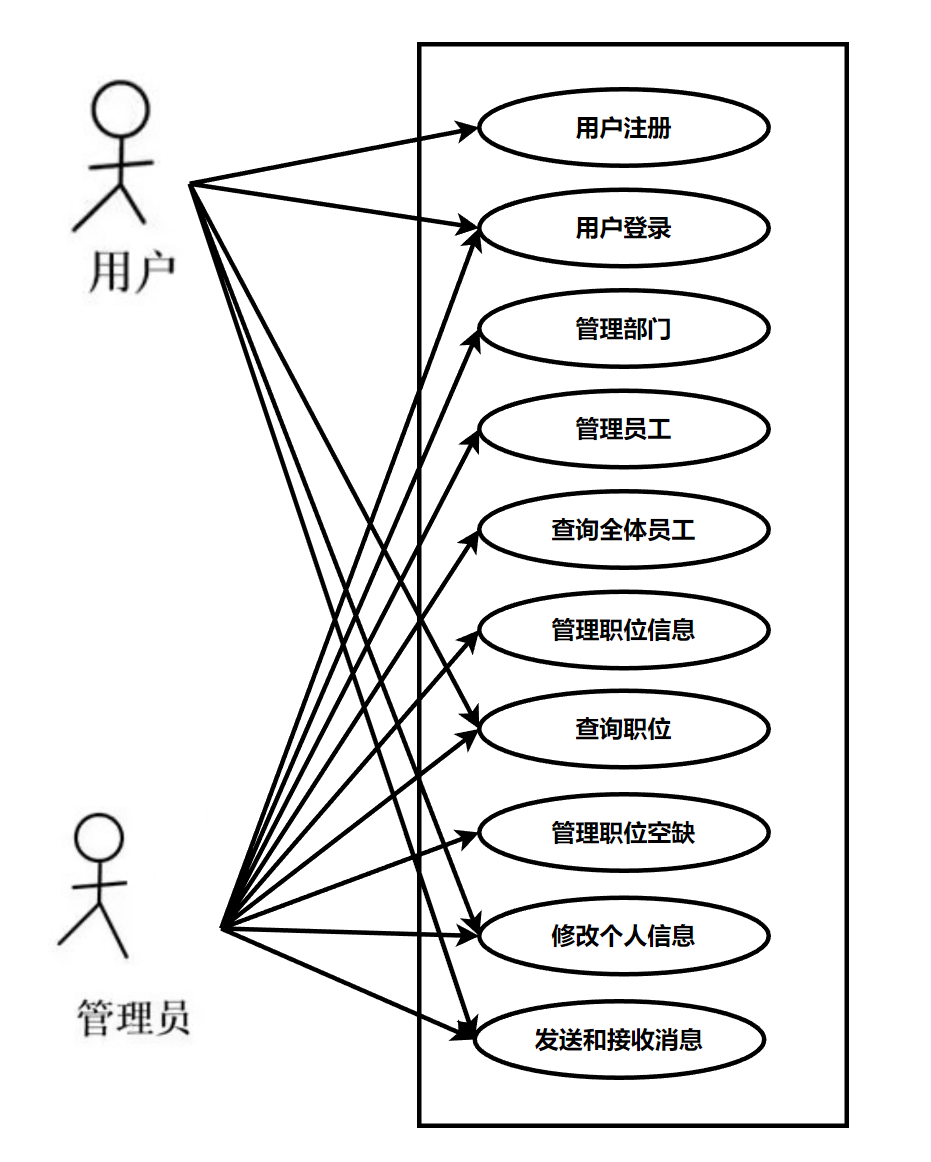


图8-1 系统总体用况图

## 1.2.2 注册

**用况：**注册

**参与者：**用户

**目的：**创建一个新的用户账户

**概述：**潜在用户提供必要的个人信息以创建一个新的账户来访问系统功能。

**类型：**主要和基本的

**交叉引用：**FU1

**典型的事件发生过程**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用例开始于潜在用户希望创建一个新账户以使用系统 | （2）系统显示注册表单页面 |
| （3）用户填写必要的注册信息，如姓名、联系方式、工号等 | （4）系统验证提供的信息，并创建新账户 |
| （5）用户完成注册并获得登录凭证 |  |

**可供选择的过程：**

如果用户未能提供所有要求的信息或某些信息无效，系统将提示错误，并要求用户更正。如果用户尝试注册的用户名已存在，系统将显示错误，并提示用户尝试不同的用户名。

## 1.2.3 登录

**用况：**登录

**参与者：**用户或管理员

**目的：**获得系统访问权限

**概述：**用户或管理员输入有效的用户名和密码，系统验证凭证后允许访问。

**交叉引用：**FU2

**典型事件发生过程：**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用户或管理员希望登录访问系统 | （2）系统提供登录页面 |
| （3）用户或管理员输入用户名和密码并提交 | （4）系统验证身份并授予权限进入主页 |
| （5）用户或管理员成功登录 |  |

**可供选择的过程：**

如身份验证失败，或是因为输入信息不完整、用户不存在或密码错误，系统提示登录失败，并允许重新尝试。

## 1.2.4 管理部门信息

**用况：增加删除修改部门**

**参与者：**管理员

**目的：** 允许管理员有效地管理公司组织结构，通过添加新的部门、删除不再存在的部门或修改现有部门的详细信息，以确保公司的部门结构数据保持最新。

**概述：**允许管理员对公司部门信息进行管理操作，包括增加新部门、删除已经废弃的部门或者修改现有部门的相关信息。这是维持公司组织结构数据有效性与准确性的一个重要过程。

**类型：**主要的和基本的

**交叉引用：FD1、FD2、FD3**

**典型的事件发生过程：**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用例开始于管理员需要增加或删除或修改部门 | （2）系统提供一个表单以输入部门的详细信息，并要求管理员填写 |
| （3）管理员填写表单，选择需要进行的操作 | （4）系统验证提交的信息，并在数据库中更新部门信息。向管理员提供操作成功或失败的反馈，以及失败的原因（如果有） |

**可供选择的过程：**

若必要信息缺失或错误，系统提示错误并要求修正。

如果尝试删除的部门中仍有员工，系统将提示该部门不能被删除，并要求移动员工到其他部门。

如果管理员尝试添加的部门名称已存在，系统将提示部门名称冲突并要求更正。

如果管理员在修改过程中未做任何更改，则系统提示无需更新。

## 1.2.5管理员工信息

用况：增加或删除或修改员工信息

参与者：管理员

许管理员在系统中有效地管理员工数据，通过删除离职员工的信息或者修改现有员工的个人详情，以确保员工数据的准确性和完整性。

概述：管理员通过系统对员工信息进行删除或更新操作。在系统验证管理员的操作权限后，允许进行删除或修改员工信息的操作，从而维护员工数据的准确性和完整性

交叉引用：**FR1、FR2、FR3**

典型事件发生过程：

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用例开始于管理员需要增加或删除或修改员工信息 | （2）系统提供员工管理页面并提供表单 |
| （3）管理员填写表单，选择需要进行的操作 | （4）系统验证提交的信息，并在数据库中更新员工信息。向管理员提供操作成功或失败的反馈，以及失败的原因（如果有） |

可供选择的过程：

如果管理员在删除操作中未能确认，系统将取消删除操作。

若管理员提交的修改信息缺失或错误，系统提示错误并要求修正。

若存在关联的业务数据限制，比如该员工是某个部门的负责人，则系统提示不能删除，并要求先更改部门负责人。

如果管理员在修改过程中未做任何更改，则系统提示无需更新。

如果系统出现错误或异常，管理员可以尝试重新执行操作或联系技术支持人员解决问题。

## 1.2.6 查询全体员工

**用况：**查询全体员工信息

**参与者：**管理员

**目的：**获取系统中所有员工的详细信息

**概述：**管理员通过系统进行查询以获取所有员工的信息，包括个人资料和职务信息。

**类型：**主要的和基本的

**交叉引用：FR4**

**典型的事件发生过程：**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）管理员希望获取所有员工的信息 |  |
| （2）管理员在系统提供的查询页面，输入查询信息，包括筛选和排序选项，设定查询条件（如部门、职务等），并提交查询请求 | （3）系统根据管理员的查询条件检索员工信息，显示满足条件的员工列表，包括详细信息 |
| （4）管理员查看并分析员工信息，成功获取所有员工信息 |  |

**可供选择的过程：**

（3）如果管理员提供的查询条件不合法或不完整，系统将提示错误并要求修正。如果没有符合查询条件的员工，系统将提示无匹配结果。

## 1.2.7 管理职位信息

**用况：**管理职位信息

**参与者：**管理员

目的：允许管理员有效地管理公司职位信息，通过添加新职位、修改现有职位或删除过时的职位，以确保公司职位数据的更新和准确性。

概述：此用况允许管理员对职位数据进行增加、修改或删除操作。这是确保组织结构与员工职位数据一致性和准确性的关键方式。

**类型：**主要的和基本的

**交叉引用：FP1、FP2、FP3**

**典型的事件发生过程：**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用例开始于管理员需要增加、修改或删除一个职位 | （2）针对增加或编辑操作，系统提供表单以输入职位的详细信息，并要求管理员填写。对于删除操作，系统询问管理员是否确认删除 |
| （）管理员填写新职位信息或更新现有职位信息，或确认删除操作 | （4）系统根据操作类型进行验证。对于增加或修改操作，如果信息验证通过，则更新数据库并显示操作成功信息。对于删除操作，系统执行删除并显示删除成功信息。 |

**可供选择的过程：**

（3）如职位描述不完整或不符合公司标准，系统会提示错误并要求修正。

如果管理员尝试添加的职位名称已存在，系统将提示职位名称冲突并要求更正。

在删除职位前，如果有员工仍属于该职位，系统将提示无法删除并建议先将员工转移到其他职位。

如果管理员在修改过程中未做任何更改，则系统提示无需更新。

对于任何操作，如果系统遇到错误或异常情况，管理员可能需要重新尝试或联系技术支持人员以解决问题。

## 1.2.8查询职位

**用况：**查询职位

**参与者：**用户、管理员

**目的：**在系统中查找符合自己需求的职位空缺。

**概述：**求职者使用系统的查询功能，查找满足其职业需求的职位空缺，并获取有关这些职位的详细信息。

**类型：**主要的和基本的

**交叉引用：FP4**

**典型的事件发生过程：**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用例开始于求职者希望找到合适的职位空缺。 |  |
| （2）用户提供搜索关键词、地点、薪水范围等搜索条件。 |  |
| （3）用户启动查询操作。 | （4）系统根据提供的搜索条件，检索并显示相关的职位空缺列表。 |
| （5）用户查看职位列表，并选择感兴趣的职位。 |  |
| （6）用户查看选定职位的详细信息。 |  |

**可供选择的过程：**

（3）如果系统无法找到符合条件的职位，系统通知求职者并提供建议。

## 1.2.9管理职位空缺

**用况：**管理职位空缺

**参与者：**管理员

**目的：**对系统中的职位空缺进行修改与删除，更新招聘信息

**概述：**管理员修改职位空缺信息并在系统中发布，以便招聘岗位更符合公司需求。

**交叉引用：**FP1、FP2

**典型事件发生过程：**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用例开始于管理员需要选择一个职位 | （2）系统提供修改界面 |
| （3）管理员修改职位空缺描述、要求和开放日期 | （4）系统修改职位空缺并将其设置为可见状态 |
| （5）管理员确认发布修改 | （6）系统确认修改成功，并在公开的招聘列表中更新信息 |

**可供选择的过程：**

（4）如果管理员决定删除此职位空缺，系统将删除相关信息，将职位空缺从招聘列表中移除。

## 1.2.10 修改个人信息

**用况：**修改个人信息

**参与者：**用户或管理员

**目的：**更新或更正用户或管理员的个人信息。

**概述：**用户或管理员登录系统，导航至个人信息页面，进行必要的更改并提交更新。

**类型：**主要的和基本的

**交叉引用：FU3**

**典型的事件发生过程：**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用例开始于用户或管理员希望更新他们在系统中的资料 |  |
| （2）用户或管理员进入个人信息页面并进行编辑 | （3）系统展示当前存储的个人信息并允许修改 |
| （4）用户或管理员更新信息并提交更改 | （5）系统验证并保存更新后的信息 |
| （6）用户或管理员的个人信息成功更新 |  |

**可供选择的过程：**

（4）如果用户或管理员提供了无效的更新，系统将显示错误并提示重新输入。

## 1.2.11 发送和接收消息

**用况：**发送和接收消息

**参与者：**用户、管理员

**目的：**促进用户间以及用户与管理员之间的通信。

**概述：**用户可以通过系统内建的消息系统发送、接收以及管理个人消息。

**类型：**主要

**交叉引用：FS1、FS2**

**典型的事件发生过程：**

|  |  |
| --- | --- |
| 参与者的动作 | 系统响应 |
| （1）用户希望发送消息给其他用户或管理员 |  |
| （2）用户在消息模块中编辑并发送消息 | （3）系统将消息传递给收件人，并保存发送记录 |
| （4）收件人接收到消息，并选择回复或忽略 | （5）系统展示收到的消息，并在用户回复时发送回复消息 |
| （6）用户发送和接收消息过程顺利完成 |  |

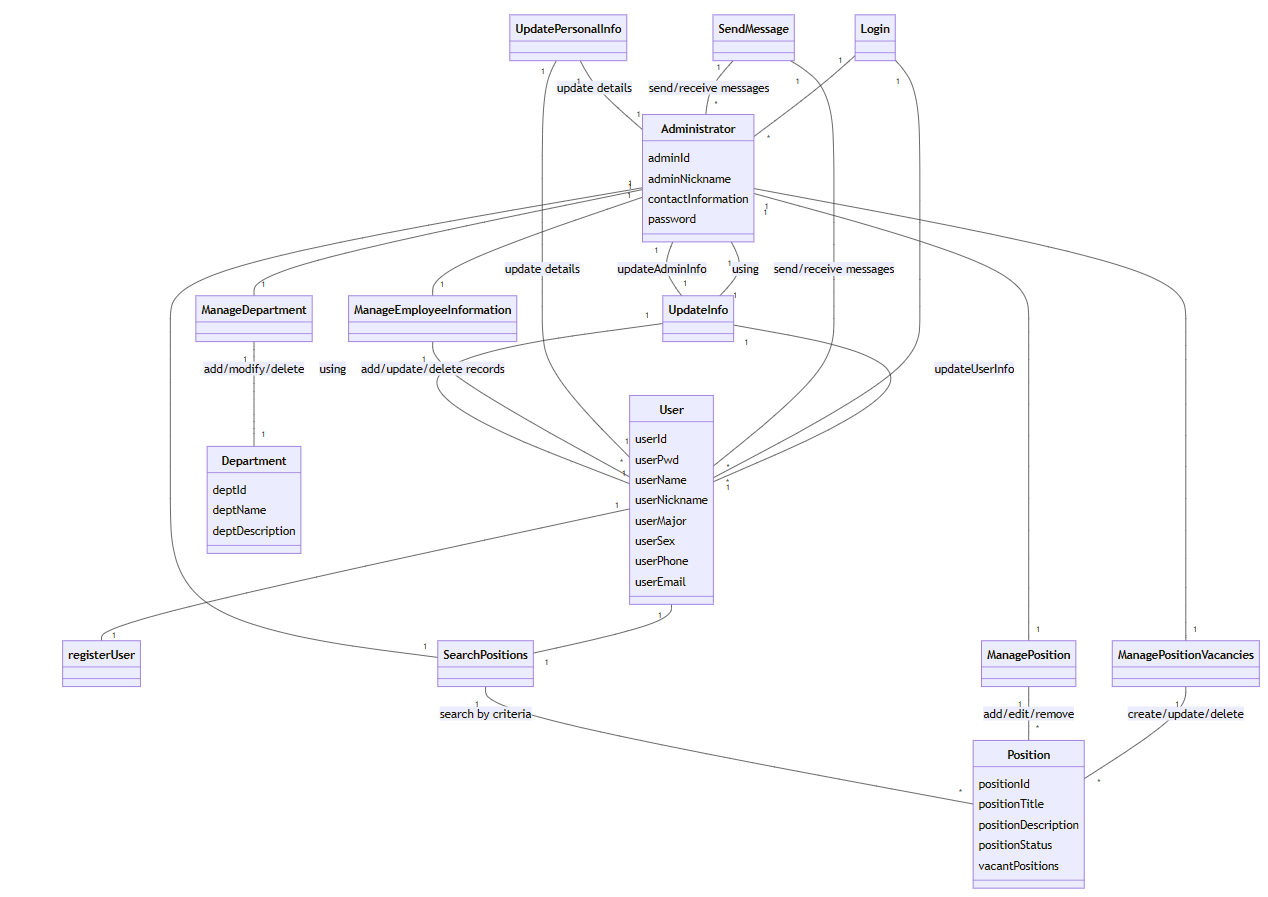
**可供选择的过程：**

（3）如果发送者或收件人不存在或收件箱已满，系统将提示错误并且不会发送该消息。

（5）如果回复消息中包含失效或不恰当的内容，系统将阻止发送并提示原因。

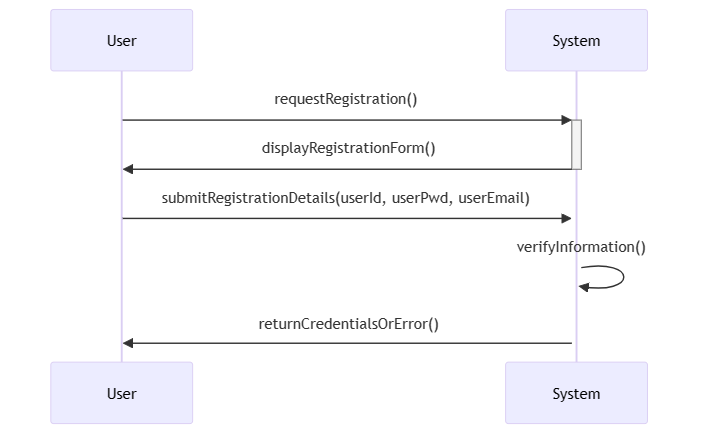
1. **系统分析与设计**

**1.概念模型**

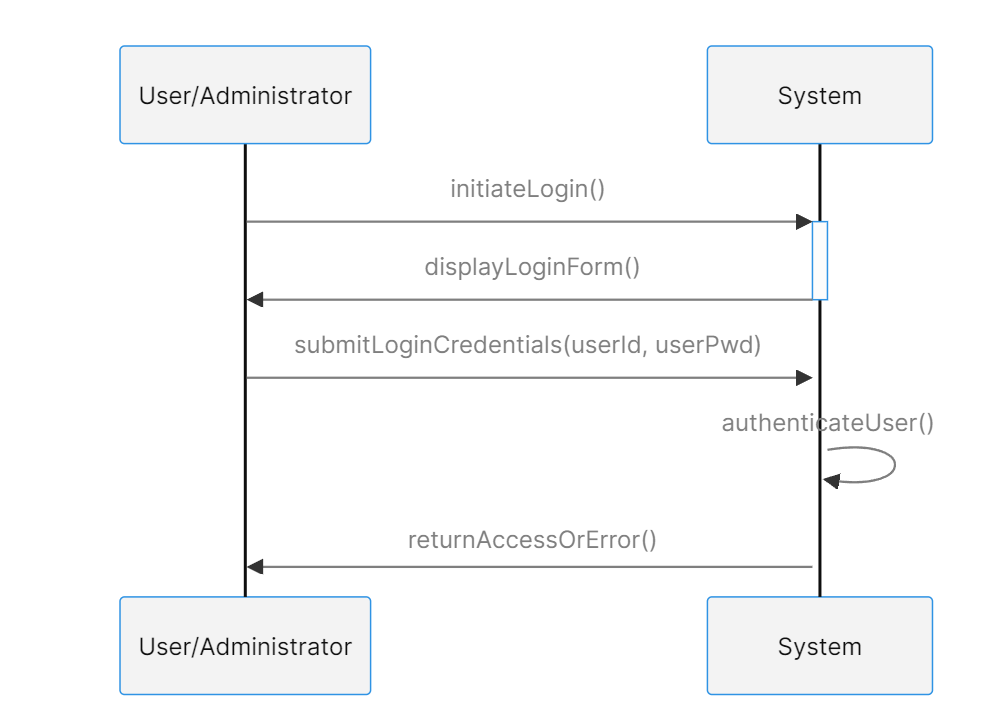


1. **系统顺序图**

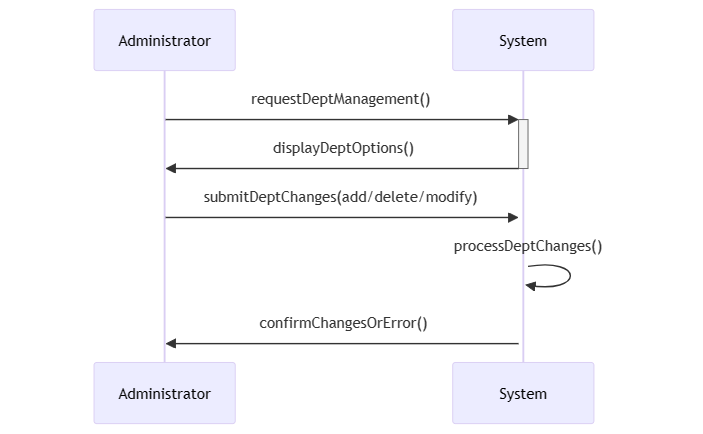
用户注册顺序图



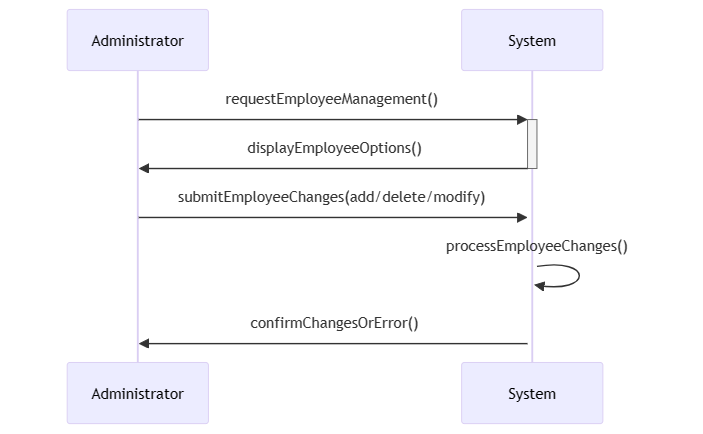
用户登录顺序图



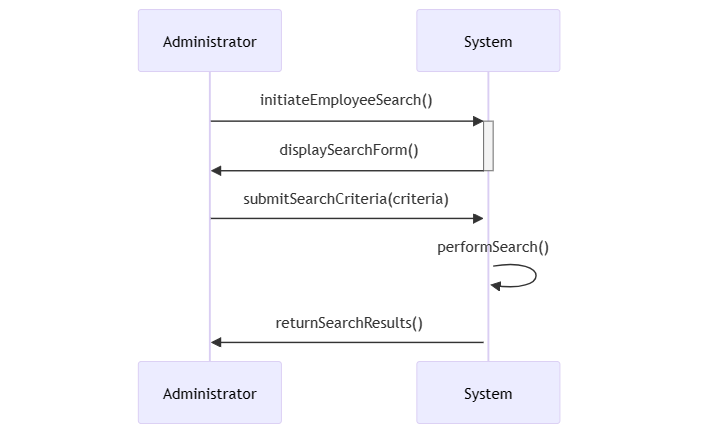
管理部门信息顺序图



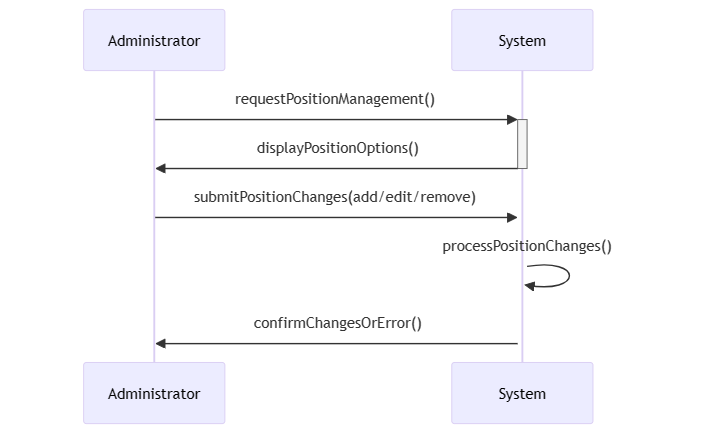
管理员工顺序图



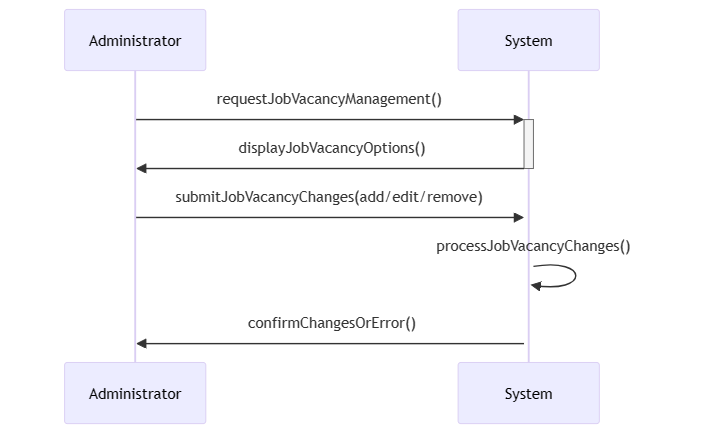
查询员工顺序图



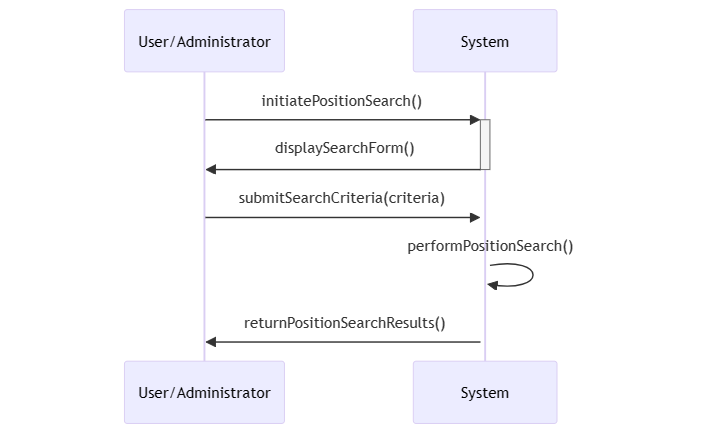
管理职位顺序图



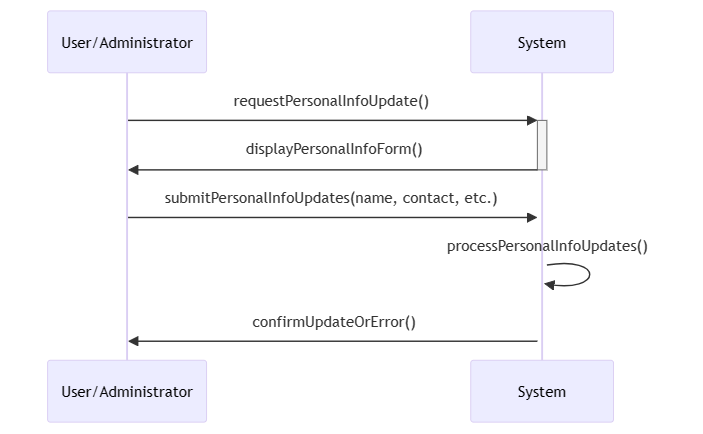
管理职位空缺顺序图



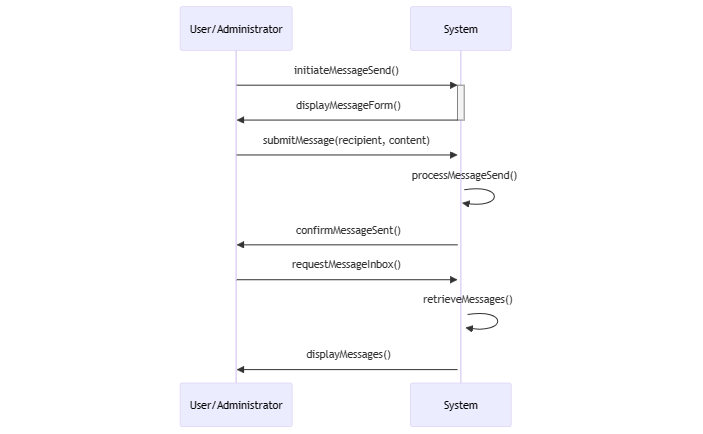
查询职位顺序图



修改个人信息顺序图

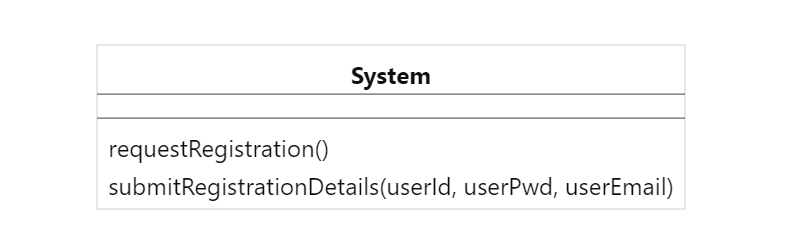


收发消息顺序图

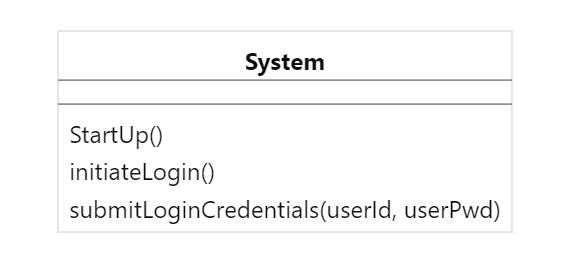


**3.系统操作**

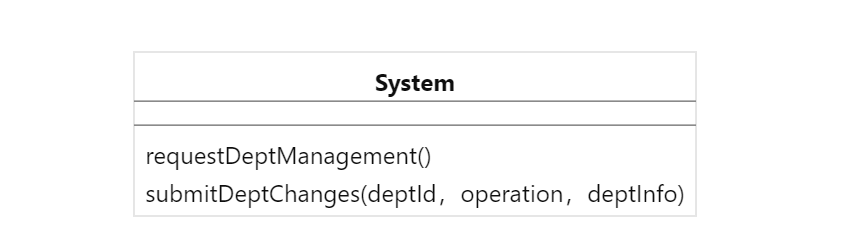
**用户注册对应系统操作**

****

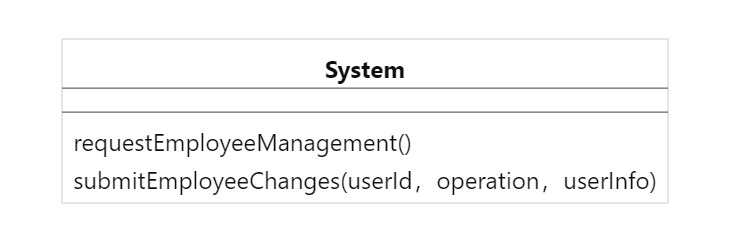
**用户登录对应系统操作**



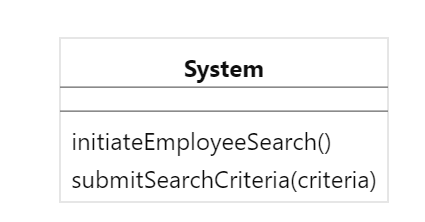
**管理部门信息对应系统操作**



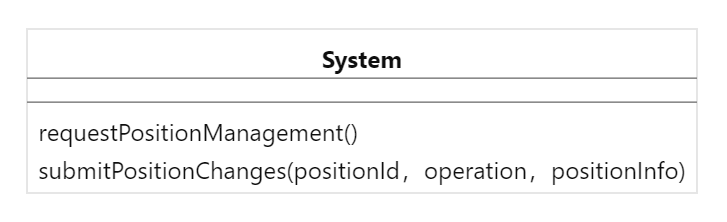
**增加或删除或修改员工信息对应系统操作**

****

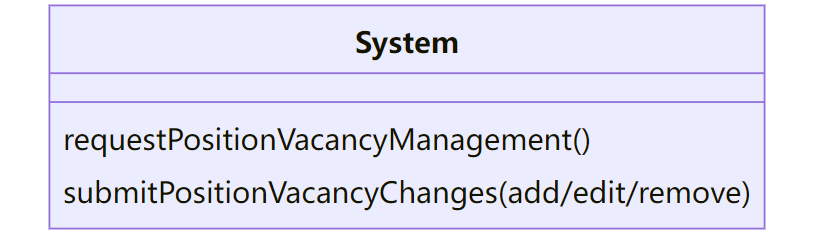
**查询全体员工对应系统操作**

****

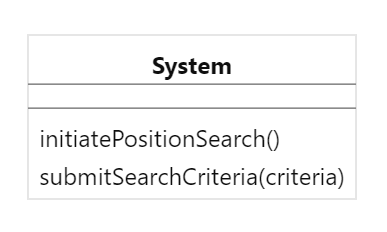
**管理职位信息对应系统操作**

****

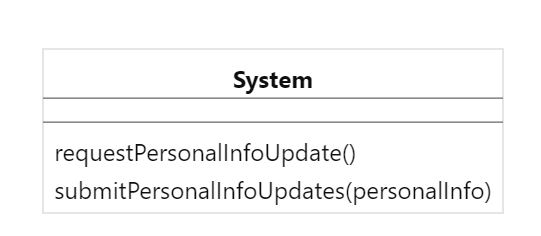
**管理职位空缺对应系统操作**

****

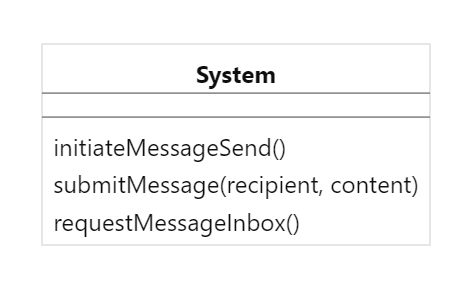
**查询职位对应系统操作**

****

**修改个人信息对应系统操作**

****

**发送和接收消息对应系统操作**



**4.契约**

### UC1: 用户注册 (User Registration)

**契约1**

**名称：**requestRegistration

**职责：**启动用户注册过程，显示注册表单。

**类型：**系统

**交叉引用：** 无

**用况：**UC1

**注释：**

**异常：**

**输出：**显示注册表单 displayRegistrationForm()

**前置条件：**用户选择进行注册。

**后置条件：**用户被呈现注册表单。

**契约2**

**名称：submitRegistrationDetails**

**职责：**职责：处理用户提交的注册表单，并创建账户。

系统 交叉引用：FU1

用况：UC1注册

注释：userId, userPwd, userEmail都不为空。

异常：如果用户ID已存在数据库中，返回提示。

输出：如果注册成功，返回登录凭证；如果失败，返回错误信息returnCredentialsOrError()。

前置条件：用户已填写注册表单。

后置条件：

1. 根据用户提供的userId, userPwd和userEmail创建一个新的用户账号保存在数据库中
2. 新建一个User实例
3. 用户被分配了登录凭证（如用户名和密码）。

### UC2: 用户登录 (User Login)

**契约3**

**契约：initiateLogin**

职责：启动登录过程，显示登录表单。

类型：系统

交叉引用：FU2

用况：UC2登录

输出：显示登录表单 displayLoginForm()

前置条件：用户选择进行登录。

后置条件：用户被呈现登录表单。

**契约4**

**契约：submitLoginCredentials**

职责：验证用户输入的登录凭证。

类型：系统

交叉引用：FU2

用况：UC2登录

注释：userId 和 userPwd都不为空。

异常：如果用户名不存在或密码错误，返回登录失败提示。

输出：如果用户名和密码验证成功，返回访问权，失败返回错误 returnAccessOrError()。

前置条件：用户已输入用户名和密码并请求登录。

后置条件：

1. 新建一个User实例
2. 用户被分配了登录凭证（如用户名和密码）。

### UC3: 管理部门信息 (Manage Department Information)

**契约5**

**契约：requestDeptManagement**

职责：为管理员提供部门管理界面，包括添加、删除和修改部门的选项。

类型：系统

交叉引用：无

用况：UC3部门管理

注释：

异常：

输出：显示部门管理选项。

前置条件：管理员选择进行部门信息的管理。

后置条件：管理员被呈现部门管理选项，供其操作选择。

**契约6**

**契约：submitDeptChanges**

职责：根据管理员的选择执行对部门的添加、删除或修改操作。

类型：系统

交叉引用：FD1

用况：UC3部门管理

注释：deptId标识特定部门，operation 表示所需执行的操作，deptInfo 包含修改 后的部门信息。

异常： 如果部门信息不合法或未找到部门ID，返回错误提示。

输出：成功执行操作后，向管理员确认改变或出错 confirmChangesOrError()。

前置条件：管理员已经指定想要执行的操作和必要信息。

后置条件：如果操作成功，系统数据库中的部门信息被相应更新。这些契约提供 了对每个操作预期行为的正式定义，使得后端开发者能够按照既定规 格来实现相应的逻辑。

### UC4: 增加或删除或修改员工信息 (Add/Delete/Modify Employee Information)

**契约7**

**名称：displayEmployeeOptions**

**职责：**为管理员提供员工信息管理界面，包括添加、删除和修改员工的选项。

**类型：**系统

**交叉引用：** FR1

**用况：**UC4

**注释：**

**异常：**

**输出：**

**前置条件：**管理员选择了增加、删除或修改员工信息的操作。

**后置条件：**管理员呈现员工信息管理选项，供其操作选择。

**契约8**

**名称：submitEmployeeChanges**

**职责：**根据管理员的选择执行对员工的添加、删除或修改操作。

**类型：**系统

**交叉引用：** FR1

**用况：**UC4

**注释：** employeeId 标识特定员工，operation 表示所需执行的操作， employeeInfo 包含修改后的员工信息。

**异常：**如果员工信息不合法或未找到员工ID，返回错误提示。

**输出：**

**前置条件：**管理员已经指定想要执行的操作和必要信息。

**后置条件：**如果操作成功，系统数据库中的员工信息被相应更新。

**契约：displayEmployeeOptions**

职责：为管理员提供员工信息管理界面，包括添加、删除和修改员工的选项。 类型：系统 交叉引用：FR1 用况：UC4管理员工信息 前置条件：管理员选择了增加、删除或修改员工信息的操作。 后置条件：管理员呈现员工信息管理选项，供其操作选择。

**契约：submitEmployeeChanges**

职责：根据管理员的选择执行对员工的添加、删除或修改操作。 类型：系统 交叉引用：FR1 用况：UC4管理员工信息 注释：employeeId 标识特定员工，operation 表示所需执行的操作，employeeInfo 包含修改后的员工信息。 异常：如果员工信息不合法或未找到员工ID，返回错误提示。 前置条件：管理员已经指定想要执行的操作和必要信息。 后置条件：如果操作成功，系统数据库中的员工信息被相应更新。

### UC5: 查询全体员工 (Query All Employees)

**系统操作：**

**契约9**

**名称：initiateEmployeeSearch**

**职责：**启动用户搜索过程，显示搜索表单。

**类型：**系统

**交叉引用：** FR4

**用况：**UC5查询全体员工信息

**注释：** 搜索条件可以包括姓名、部门、职务等。

**异常：**如果搜索条件不合法或不完整，返回错误提示。

**输出：预搜索表单**

**前置条件：**管理员已输入搜索条件。

**后置条件：**如果找到匹配结果，系统返回过滤后的员工列表。

**契约10**

**名称：submitSearchCriteria**

**职责：**根据管理员提供的搜索条件执行员工信息的搜索。

**类型：**系统

**交叉引用：** FR4

**用况：**UC5查询全体员工信息

**注释：** 搜索条件可以包括姓名、部门、职务等。

**异常：**如果搜索条件不合法或不完整，返回错误提示。

**输出：符合搜索条件的员工列表**

**前置条件：**管理员已输入搜索条件。

**后置条件：**如果找到匹配结果，系统返回过滤后的员工列表。

### UC6: 管理职位信息 (Manage Position Information)

**系统操作：**

**契约11**

**名称：displayPositionOptions**

**职责：**为管理员提供职位信息管理界面，包括添加、修改和删除职位的选项。

**类型：**系统

**交叉引用：** FP1，FP2，FP3

**用况：**UC6管理职位信息

**注释：** add表示添加，remove表示删除，modify表示修改

**异常：**

**输出：显示操作菜单**

**前置条件：**管理员选择了增加、删除或修改职位信息的操作。

**后置条件：**系统呈现职位信息管理选项，供其操作选择。

**契约12**

**名称：submitPositionChanges**

**职责：**根据管理员的选择执行对职位的添加、删除或修改操作。

**交叉引用：** FP1，FP2，FP3

**用况：**UC6管理职位信息

**注释：** positionId 标识特定职位，operation 表示所需执行的操作，positionInfo 包含修改后的职位信息。

**异常：**如果职位信息不合法或未找到职位ID，返回错误提示。

**输出：**

**前置条件：**管理员已经指定想要执行的操作和必要信息。

**后置条件：**如果操作成功，系统数据库中的职位信息被相应更新。

### UC7: 管理职位空缺 (Manage Position Vacancies)

**契约13**

**名称**：requestPositionVacancyManagement

**职责**：为管理员提供职位空缺管理界面，可能包括把空缺发布出来、修改和删除空缺信息的选项。

**类型**：系统

**交叉引用**：FP1，FP2，FP3

**用况**：UC7管理职位空缺

**前置条件**：管理员选择了管理职位空缺的操作。

**后置条件**：管理员呈现职位空缺管理选项，供其操作选择。

**契约14**

**名称**：submitPositionVacancyChanges

**职责**：根据管理员的选择执行对职位空缺的发布、删除或修改操作。

**类型**：系统

**交叉引用**：FP1，FP2，FP3

**用况**：UC7管理职位空缺

**注释**：positionId 标识特定职位空缺，operation 表示所需执行的操作，vacancyPosition 包含修改后的职位空缺信息。

**异常**：如果职位空缺信息不合法或未找到职位空缺ID，返回错误提示。

**前置条件**：管理员已经指定想要执行的操作和必要信息。

**后置条件**：如果操作成功，系统数据库中的职位空缺信息被相应更新。

UC8: 查询职位 (Search for Positions)

**契约15**

**名称**：initiatePositionSearch

**职责**：启动职位搜索过程，显示搜索表单。

**类型**：系统

**交叉引用**：FP4

**用况**：UC8查询职位

输出：显示预搜索表单

**前置条件**：用户希望查询系统中的职位。

**后置条件**：用户被呈现用于输入搜索条件的表单。

**契约16**

**名称**：submitSearchCriteria

**职责**：根据提供的搜索条件，检索符合条件的职位。

**类型**：系统

**交叉引用**：FP4

**用况**：UC8查询职位

**注释**：搜索条件可以包括职位名称、地点、薪水范围等。

**异常**：如果没有找到符合条件的职位，返回无匹配结果提示。

**前置条件**：用户已输入搜索条件并提交搜索请求。

**后置条件**：如果找到匹配结果，系统返回职位列表。

### UC9: 修改个人信息 (Modify Personal Information)

**契约17**

**名称：**requestPersonalInfoUpdate

**职责：**提供个人信息更新界面。

**类型：**系统

**交叉引用：** FA1、FU1

**用况：**UC9

前置条件：用户或管理员希望更新其个人信息。

后置条件：呈现个人信息页面，供用户或管理员进行编辑。

**契约18**

**名称：**submitPersonalInfoUpdates

**职责：**处理提交的个人信息更新请求。

**类型：**系统

**交叉引用：** FA1、FU1

**用况：**UC9

注释：name表示修改后的姓名，contact表示修改后的联系方式等

异常：如果个人信息更新包含无效的数据，返回错误提示。

前置条件：用户或管理员已编辑个人信息并提交更新请求。

后置条件：如果更新成功，系统保存更新后的个人信息。

### UC10: 发送和接收消息 (Send and Receive Messages)

**契约19**

名称：initiateMessageSend

职责：启动消息发送过程，显示消息表单。

类型：系统

交叉引用：FS1

用况：UC10发送和接收消息

输出：显示预发送消息表单

前置条件：用户希望发送消息给其他用户或管理员。

后置条件：呈现消息编辑表单，供用户编写和发送消息。

**契约20**

名称：submitMessage

职责：处理提交的消息，并将其传递给指定的收件人。

类型：系统

交叉引用：FS1

用况：UC10

注释：recipient表示收件人，content表示消息的内容

异常：如果收件人不存在或消息内容无效，返回错误提示。

前置条件：用户已编写消息并指定了收件人。

后置条件：如果消息成功发送，系统保存消息记录并通知发送者。

**契约21**

名称：requestMessageInbox

职责：提供消息接收盒，让用户查看收到的消息。

类型：系统

交叉引用：FS1

用况：UC10发送和接收消息

前置条件：用户希望查看收到的消息。

后置条件：系统检索并展示用户的消息收件盒。

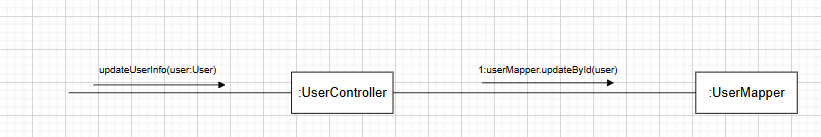
**5.术语表**

|  |  |  |
| --- | --- | --- |
| 术语 | 分类 | 注释 |
| Administrator | 类型 | 管理员 |
| Administrator.userId: String | 属性 | 管理员的用户ID |
| Administrator.adminNickname: String | 属性 | 管理员的昵称 |
| Administrator.contactInformation: String | 属性 | 管理员的联系信息 |
| Administrator.userPwd: String | 属性 | 管理员的密码 |
| User | 类型 | 用户 |
| User.userId: String | 属性 | 用户ID |
| User.userPwd: String | 属性 | 用户密码 |
| User.userName: String | 属性 | 用户名 |
| User.userNickname: String | 属性 | 用户昵称 |
| User.userMajor: String | 属性 | 用户专业 |
| User.userSex: String | 属性 | 用户性别 |
| User.userPhone: String | 属性 | 用户电话号码 |
| User.userEmail: String | 属性 | 用户电子邮件 |
| User.position: String | 属性 | 用户职位 |
| User.department: String | 属性 | 用户所在部门 |
| UserCriteria | 类型 | 用户查询条件 |
| UserCriteria.userId: String | 属性 | 用户ID |
| UserCriteria.userName: String | 属性 | 用户名 |
| UserCriteria.userAge: Int | 属性 | 用户年龄 |
| UserCriteria.userSex: String | 属性 | 用户性别 |
| UserCriteria.position: String | 属性 | 用户职位 |
| UserCriteria.department: String | 属性 | 用户所在部门 |
| UserInfo | 类型 | 用户信息 |
| UserInfo.userName: String | 属性 | 用户名 |
| UserInfo.userNickname: String | 属性 | 用户昵称 |
| UserInfo.userMajor: String | 属性 | 用户专业 |
| UserInfo.userSex: String | 属性 | 用户性别 |
| UserInfo.userPhone: String | 属性 | 用户电话号码 |
| UserInfo.userEmail: String | 属性 | 用户电子邮件 |
| UserInfo.position: String | 属性 | 用户职位 |
| UserInfo.department: String | 属性 | 用户所在部门 |
| Personnel\_Management | 类型 | 人事管理 |
| Personnel\_Management.UserController | 关联 | 人事管理-用户控制器 |
| Personnel\_Management.deptController | 关联 | 人事管理-部门控制器 |
| Personnel\_Management.positionController | 关联 | 人事管理-职位控制器 |
| Personnel\_Management.messageController | 关联 | 人事管理-消息控制器 |
| Position | 类型 | 职位 |
| Position.positionId: String | 属性 | 职位ID |
| Position.positionName: String | 属性 | 职位名称 |
| Position.status: String | 属性 | 职位状态 |
| Position.description: String | 属性 | 职位描述 |
| PositionUserController | 类型 | 职位用户控制器 |
| PositionUserController.positionMap: Map<Position> | 属性 | 职位映射表 |
| UserController | 类型 | 用户控制器 |
| UserController.userMap: Map<User> | 属性 | 用户映射表 |
| DeptController | 类型 | 部门控制器 |
| DeptController.create(deptInfo) | 方法 | 创建部门 |
| DeptController.ManagementDepartment(deptId, remove, deptInfo) | 方法 | 管理部门信息 |
| MessageController | 类型 | 消息控制器 |
| MessageController.edittedcontent: String | 属性 | 编辑后的内容 |
| Department | 类型 | 部门 |
| Department.deptId: String | 属性 | 部门ID |
| Department.deptName: String | 属性 | 部门名称 |
| Department.description: String | 属性 | 部门描述 |

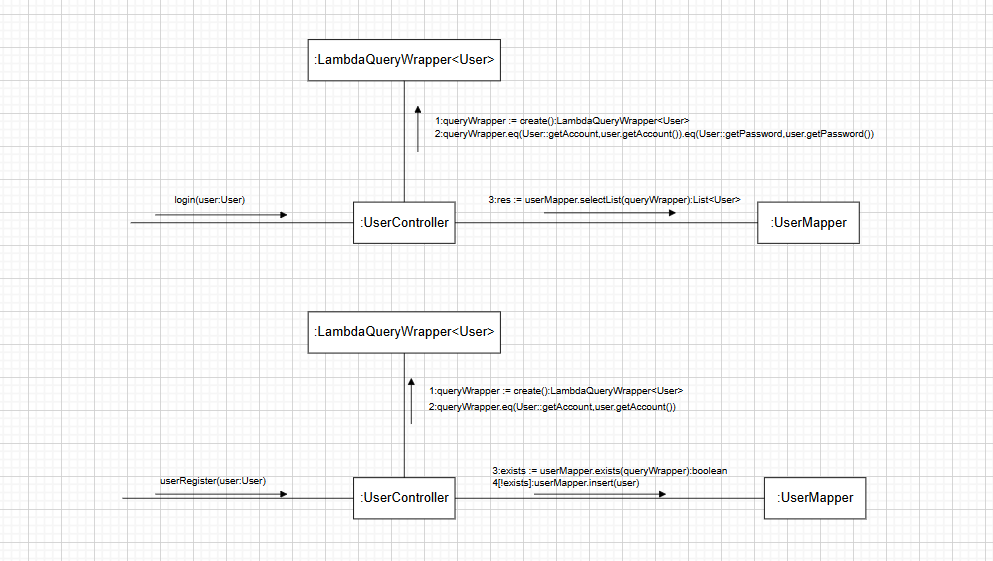
## 6.协作图

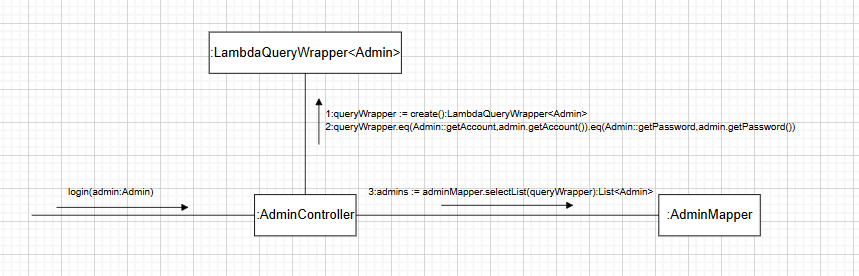
用户与管理员

用户修改个人信息：

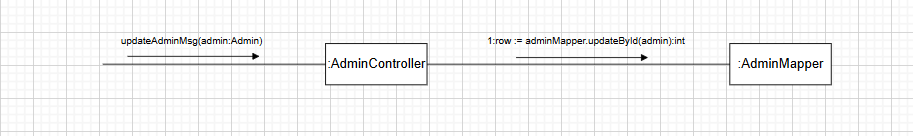


用户注册与登录：

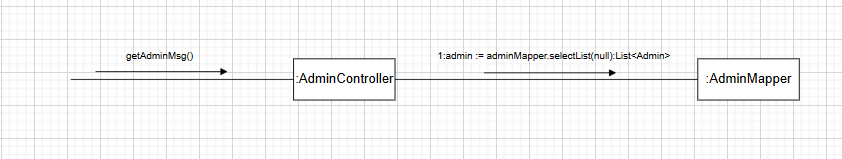


管理员登录：  


管理员更新个人信息：

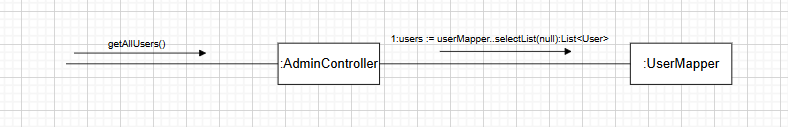


获取管理员账户信息：

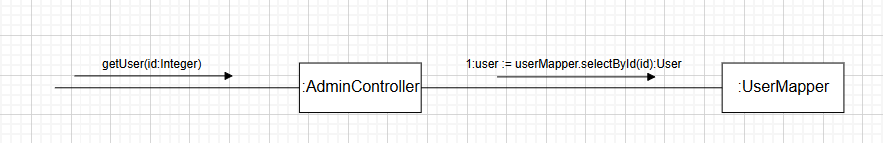


员工的增删改查

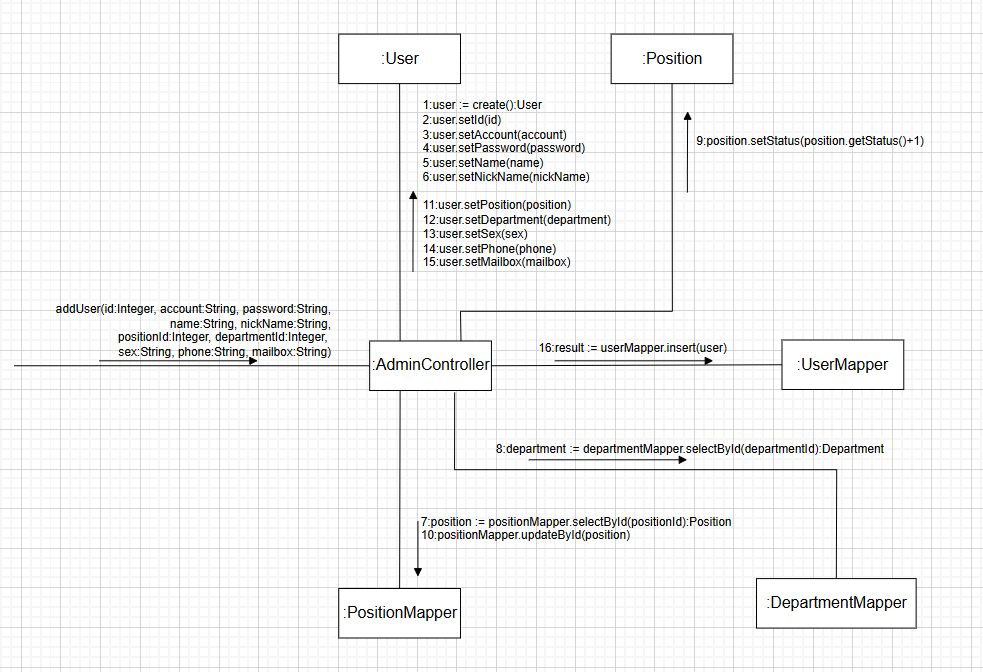
查找所有员工：



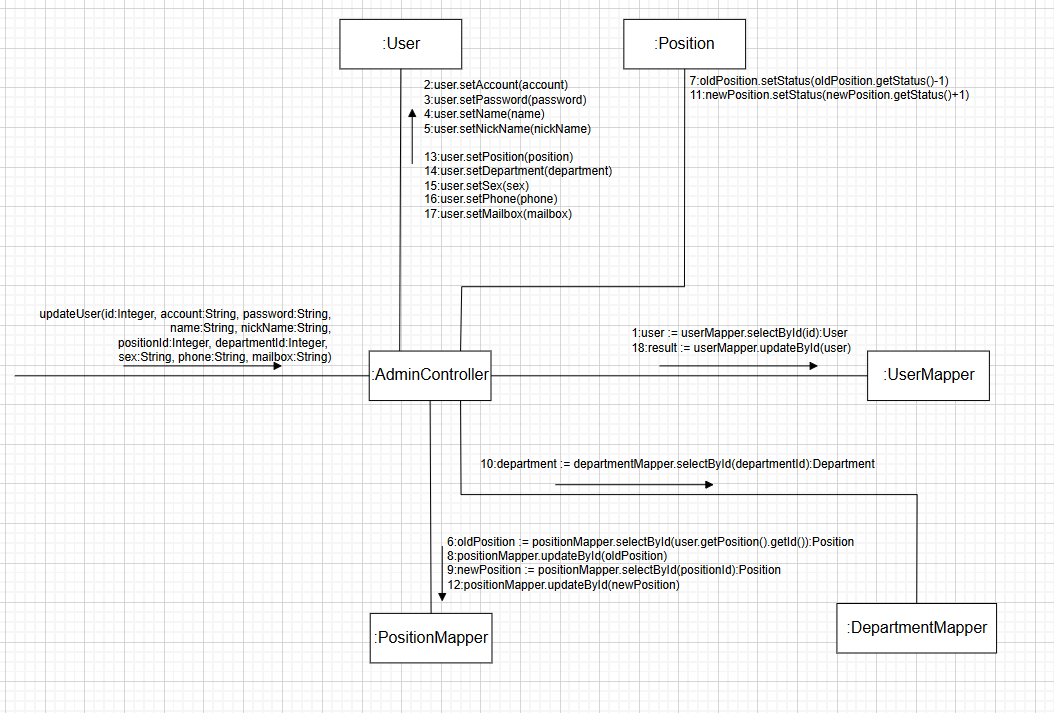
根据id查找员工：



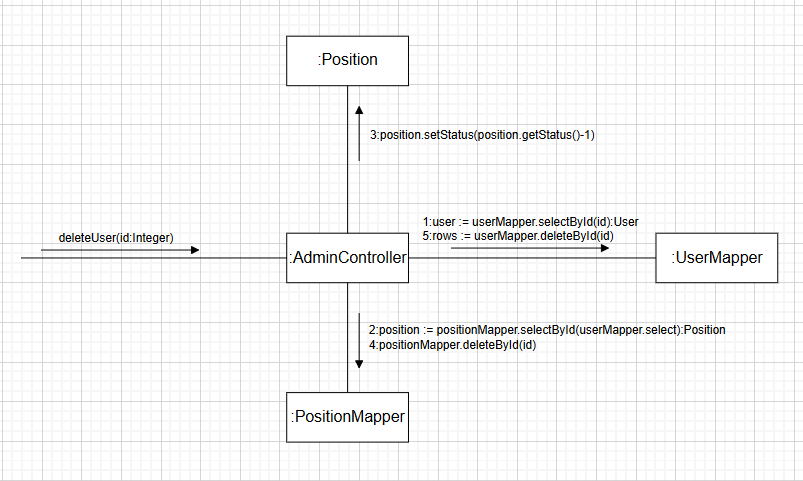
增加员工：



更新员工信息：

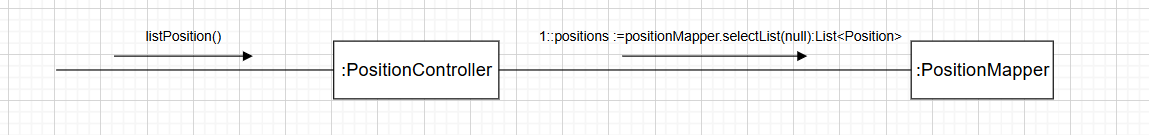


删除员工：

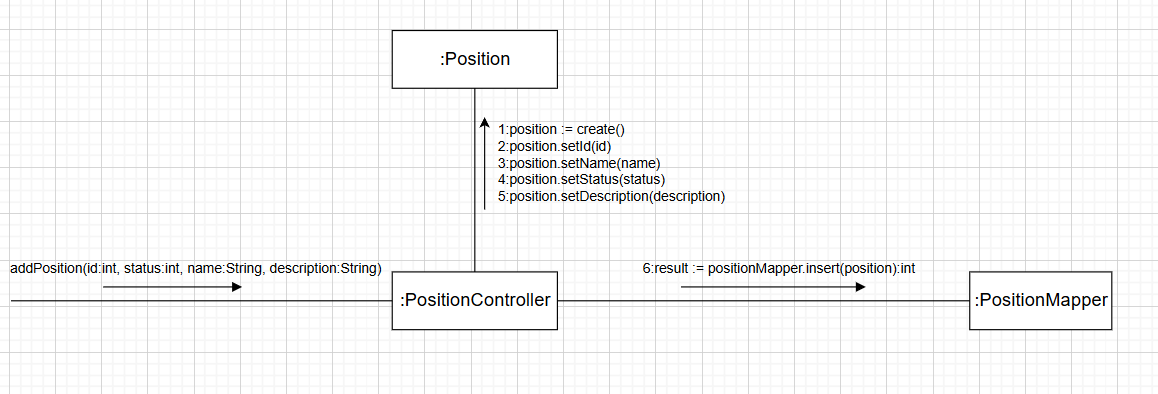


职位操作

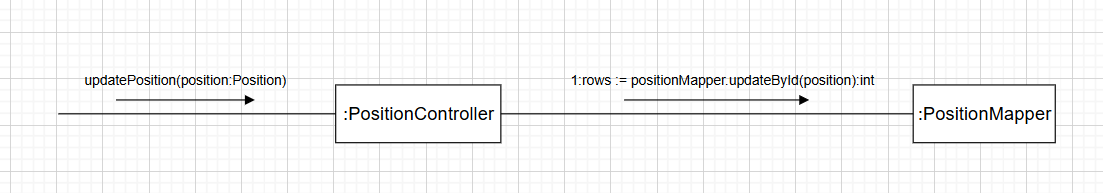
查找所有职位：



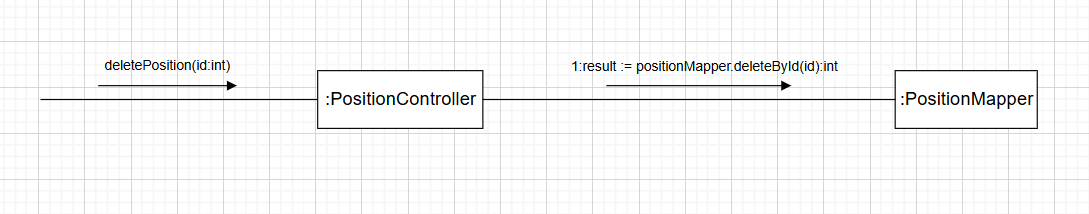
增添职位：



更新职位：

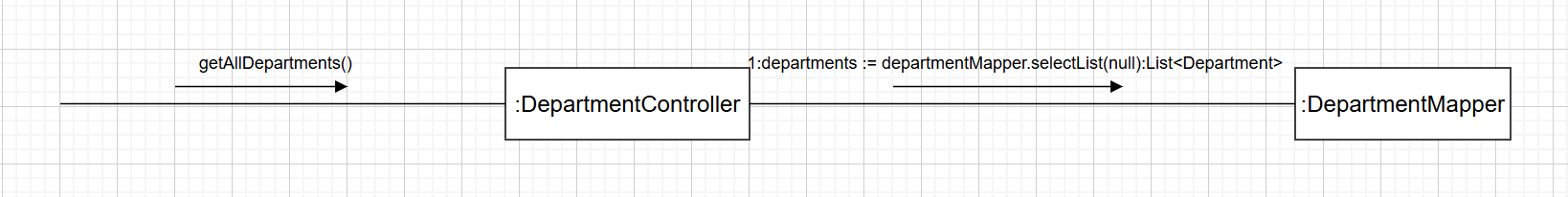


删除职位：

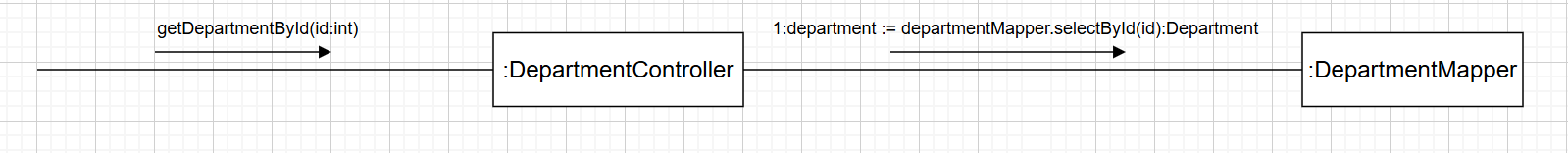


部门操作：

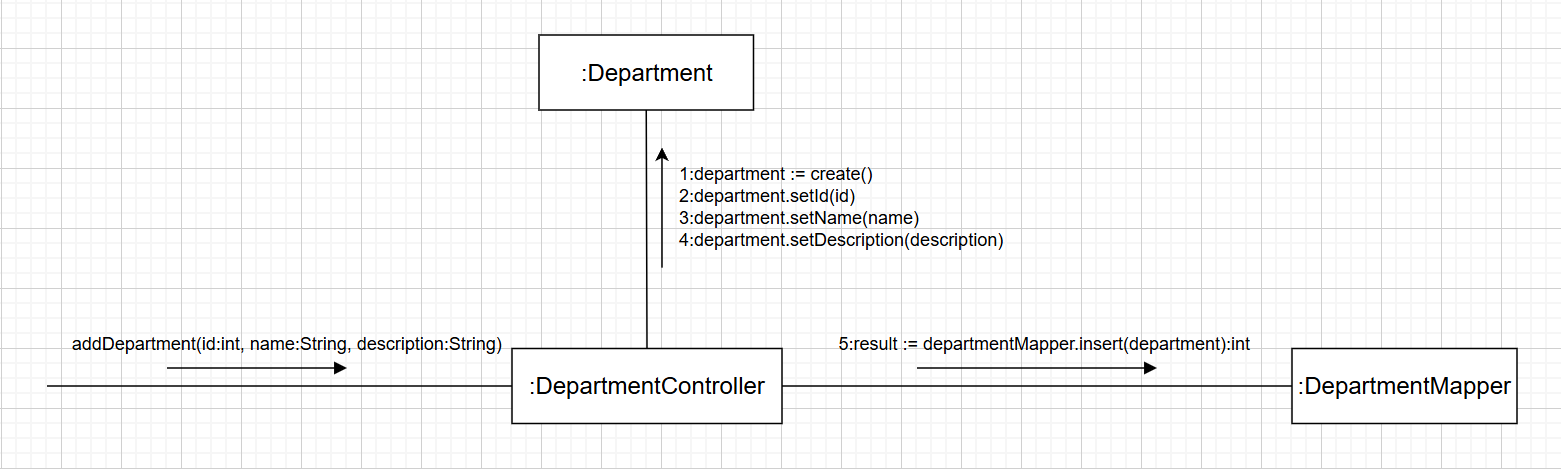
查找所有部门：



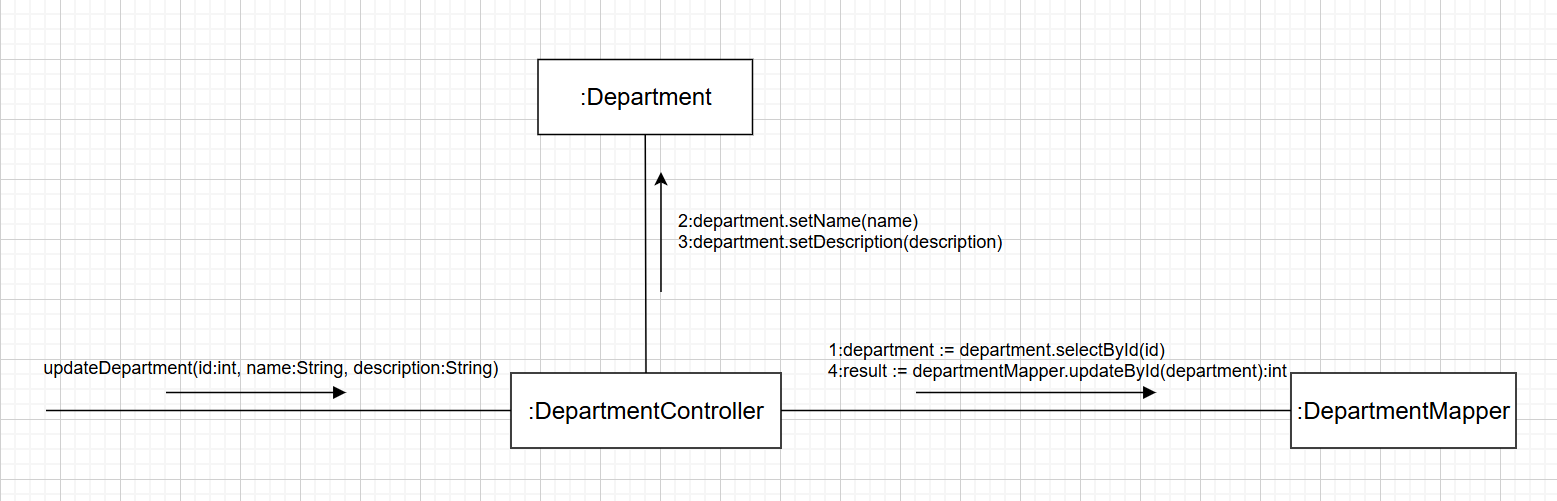
按ID查找部门：



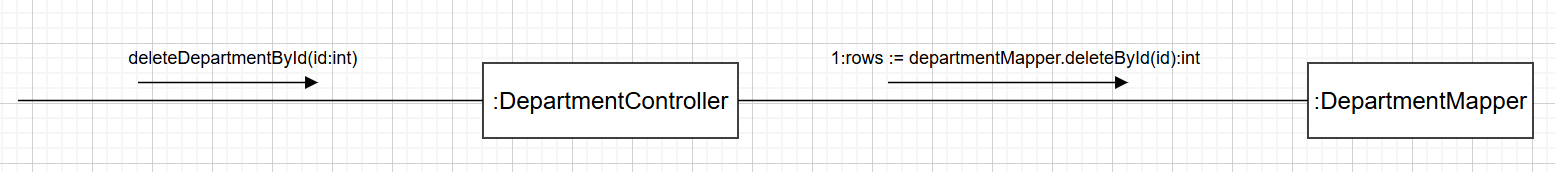
增加部门：



更新部门：

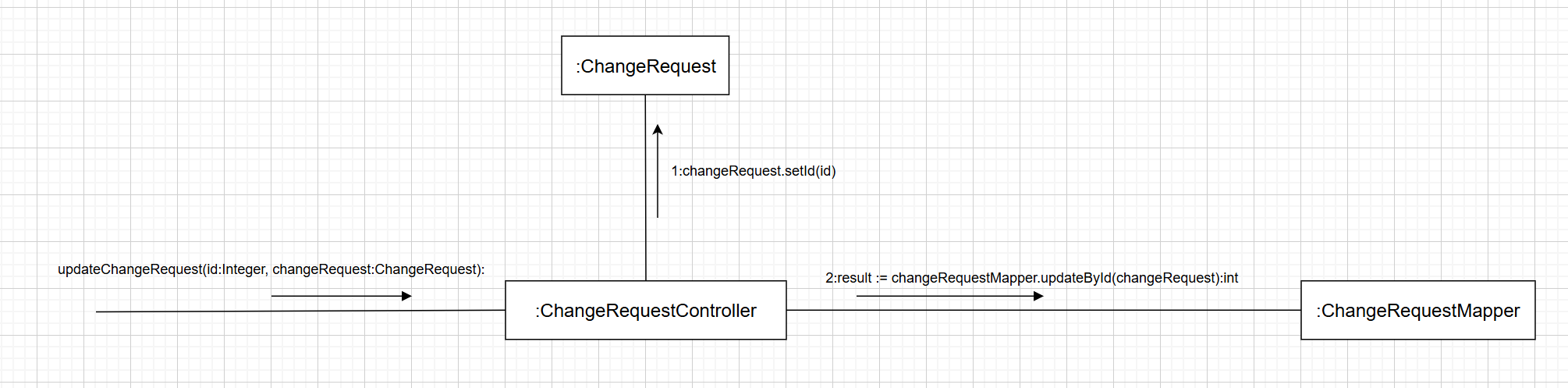


删除部门：

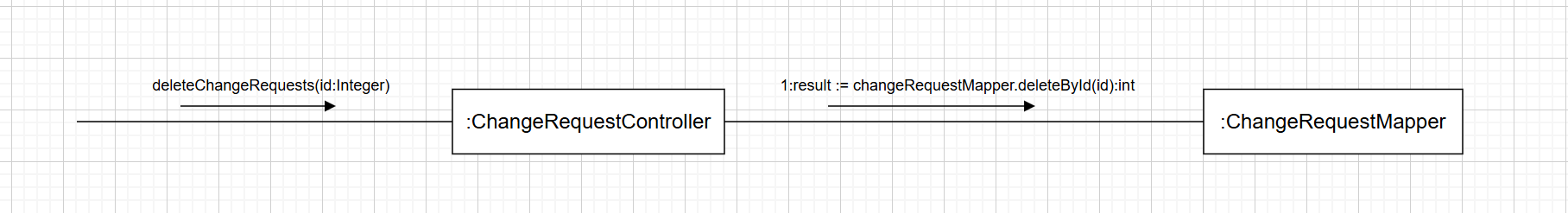


变更请求操作：

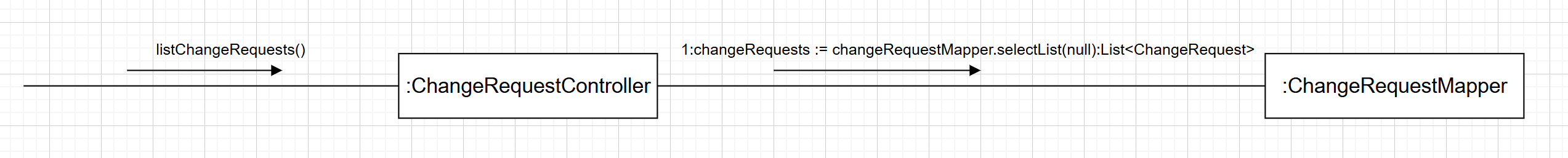
更新变更请求：



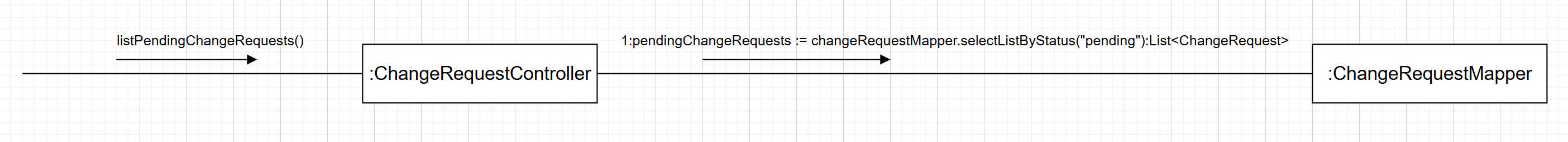
删除变更请求：



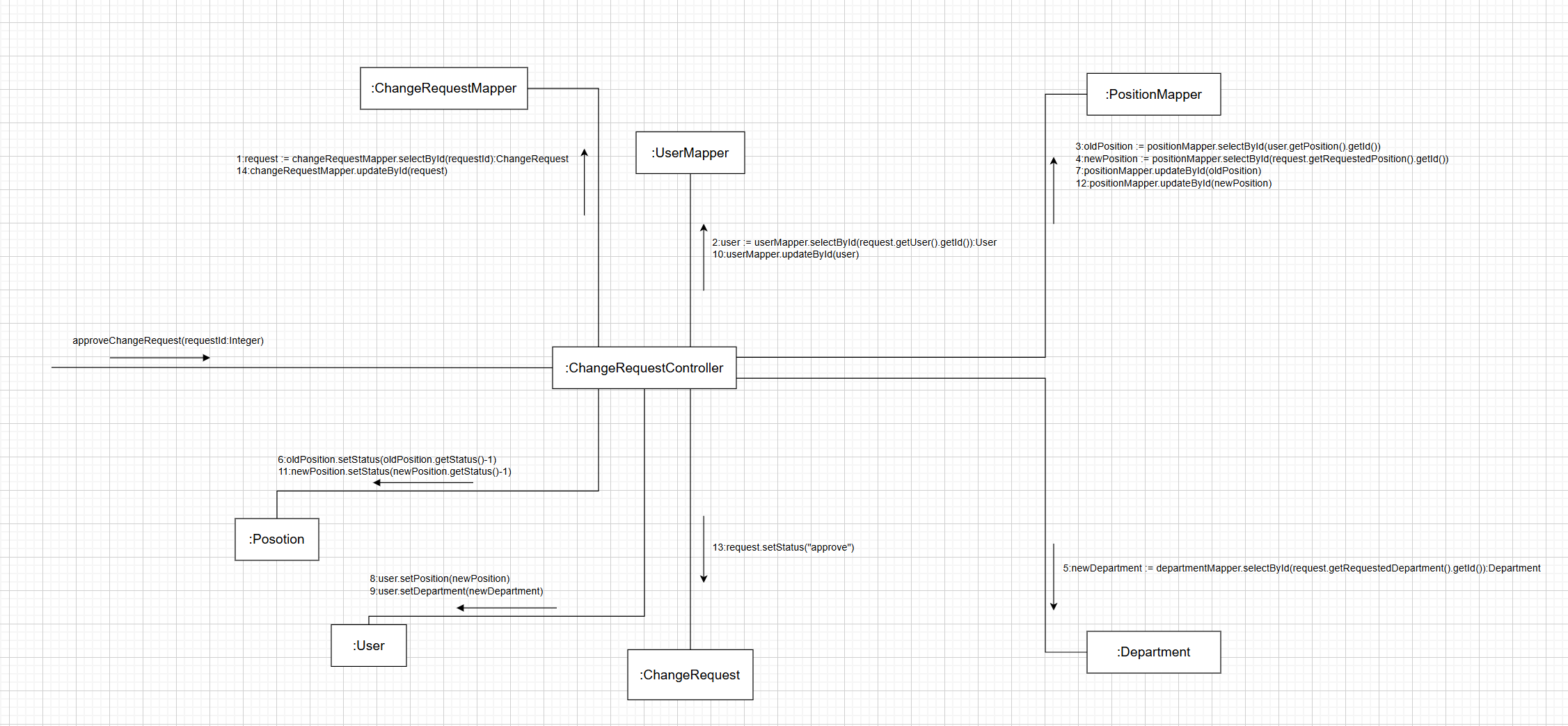
查询所有请求：



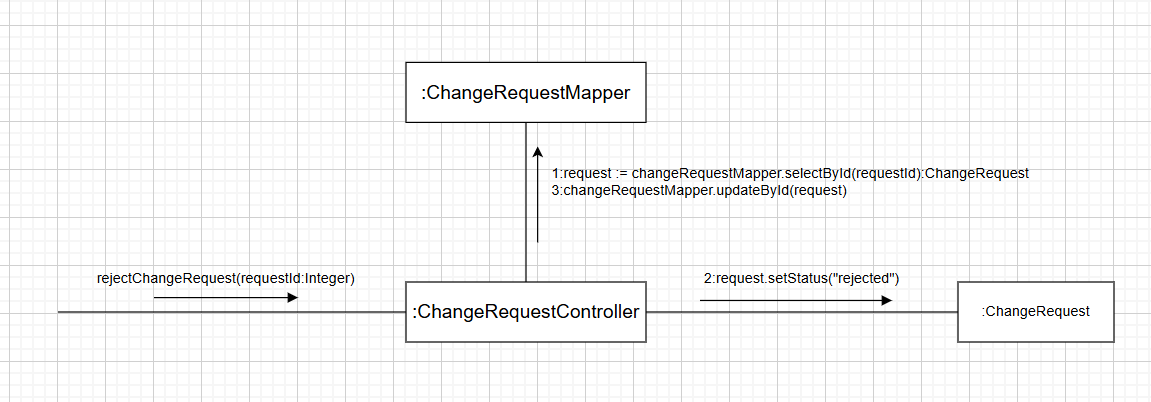
查询特定状态的请求：



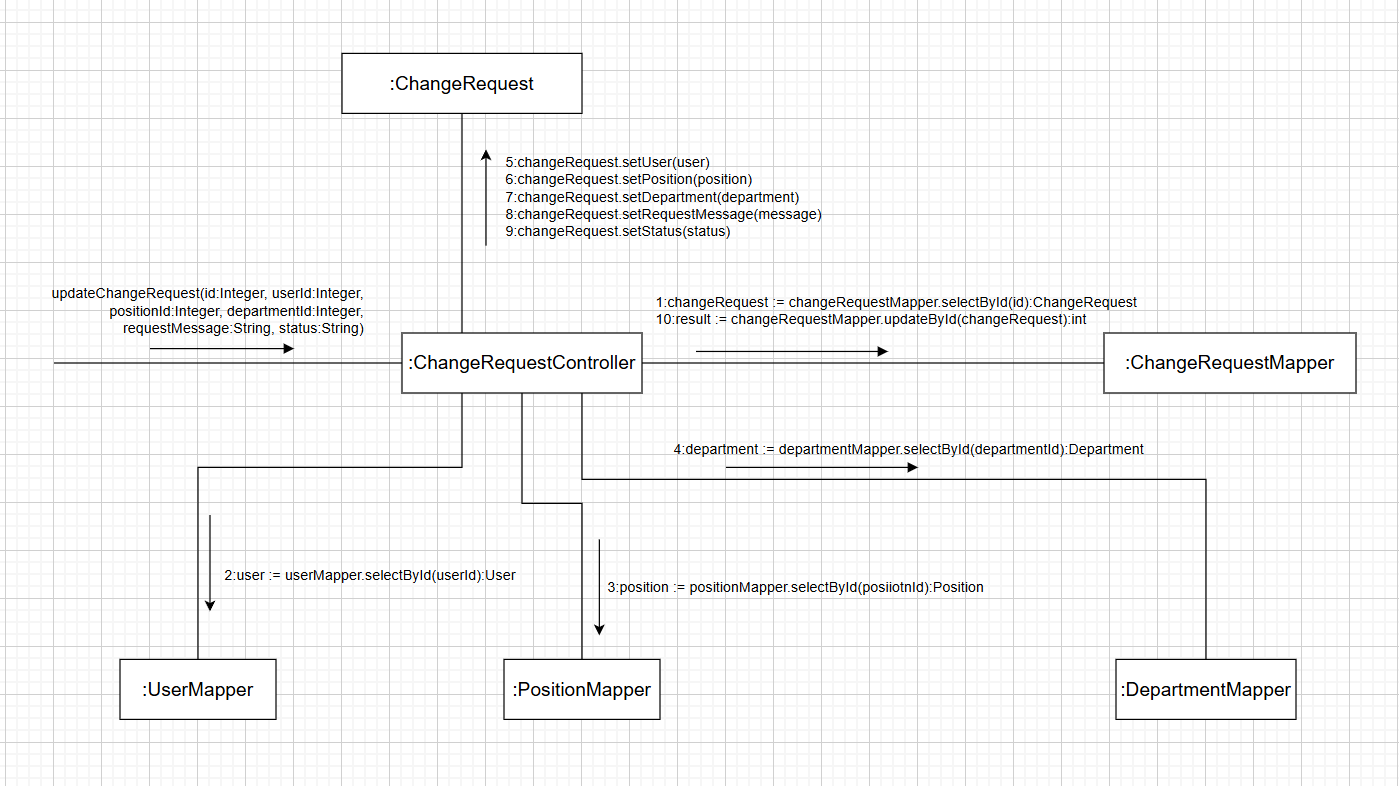
同意变更请求：



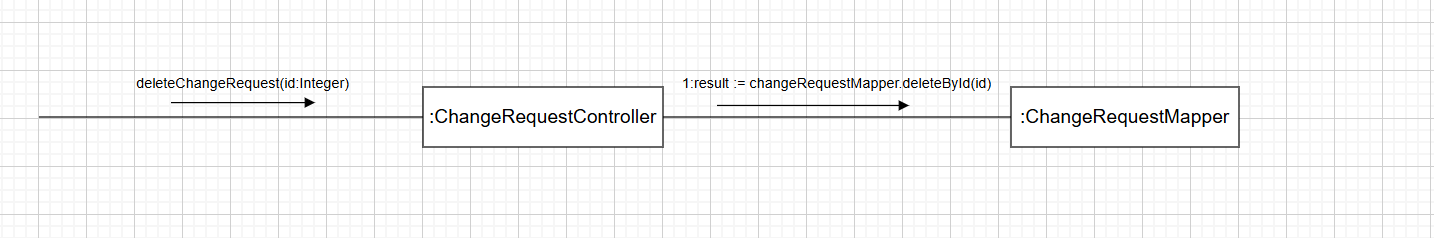
拒绝变更请求：



更新变更请求：

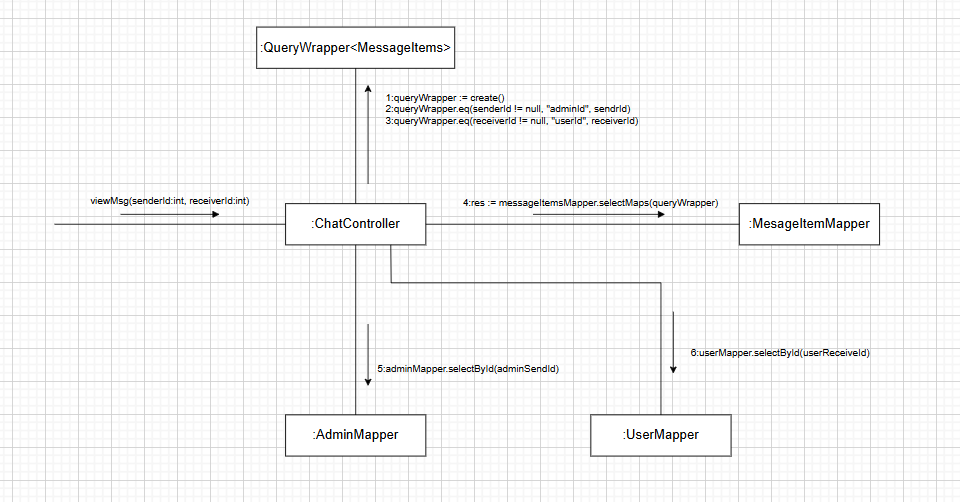


删除变更请求：

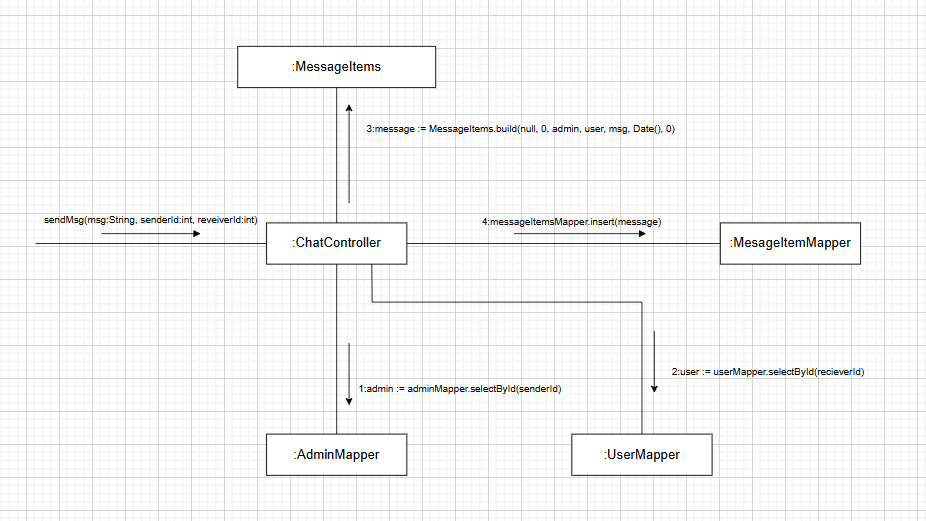


消息操作

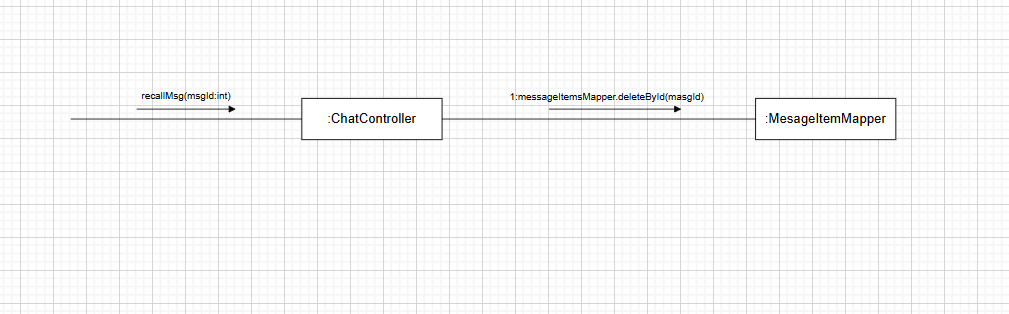
查看消息：



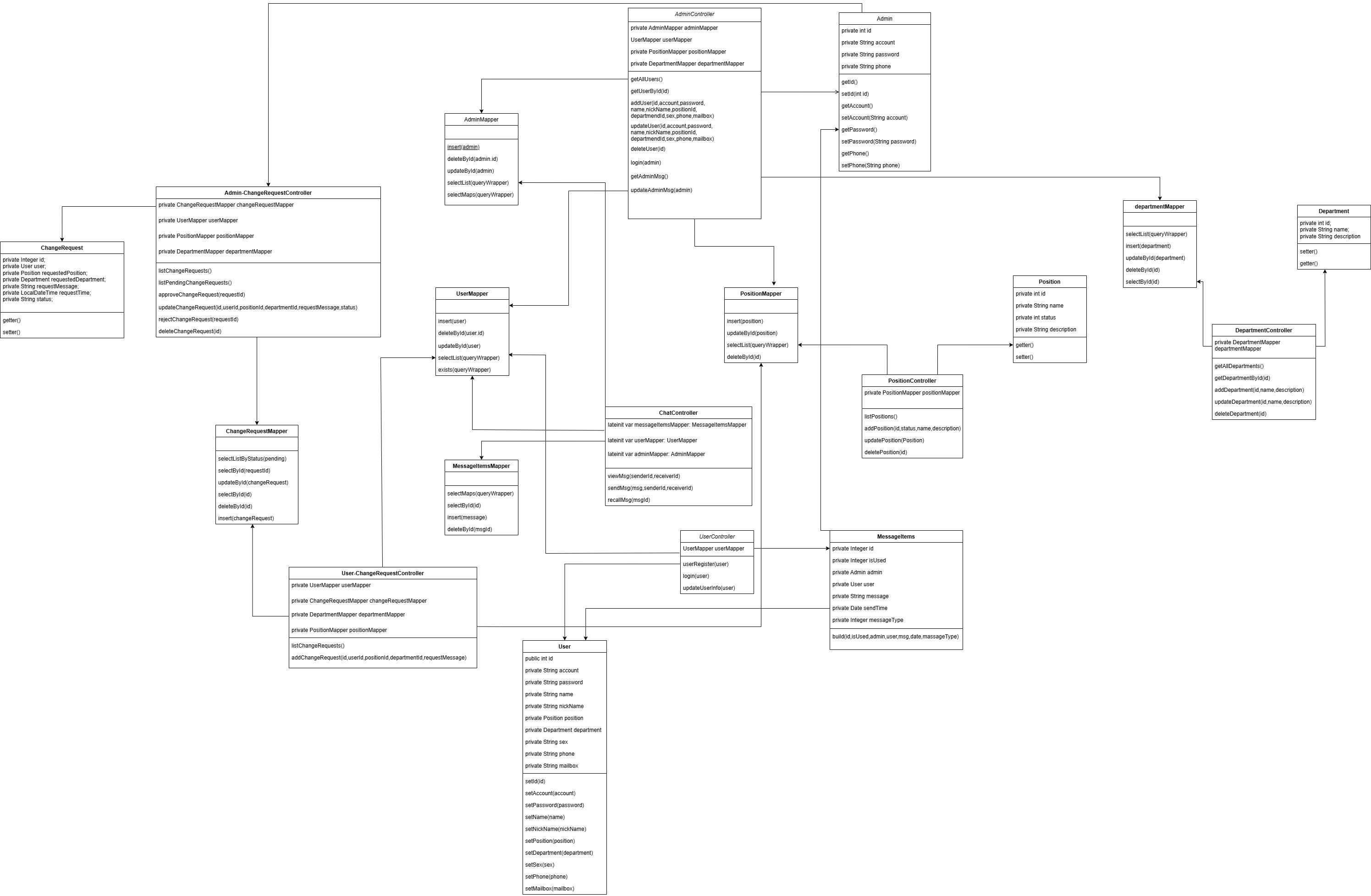
发送消息：



回复消息：



**7.类图**



**8.数据库设计**

图8-1用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段描述 | 数据类型 | 长度 | 属性 |
| 1 | userId | 用户ID | varchar | 50 | PK, 非空 |
| 2 | userPwd | 用户密码 | varchar | 250 | 非空 |
| 3 | userName | 用户名 | varchar | 100 | 非空 |
| 4 | userNickname | 用户昵称 | varchar | 100 | 非空 |
| 5 | userMajor | 用户专业 | varchar | 100 |  |
| 6 | userSex | 用户性别 | varchar | 10 |  |
| 7 | userPhone | 用户电话 | varchar | 20 |  |
| 8 | userEmail | 用户邮箱 | varchar | 100 | 非空 |
| 9 | position | 职位 | varchar | 50 |  |
| 10 | department | 所属部门 | varchar | 50 |  |

管理员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段描述 | 数据类型 | 长度 | 属性 |
| 1 | userId | 管理员ID | varchar | 50 | PK, 非空 |
| 2 | userPwd | 管理员密码 | varchar | 250 | 非空 |
| 4 | adminNickname | 管理员昵称 | varchar | 100 | 非空 |
| 4 | contactInformation | 联系方式 | varchar | 100 |  |

职位表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段描述 | 数据类型 | 长度 | 属性 |
| 1 | positionId | 职位ID | varchar | 50 | PK |
| 2 | positionName | 职位名称 | varchar | 100 | 非空 |
| 3 | status | 职位状态 | varchar | 50 |  |
| 4 | description | 职位描述 | text |  |  |

部门表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段描述 | 数据类型 | 长度 | 属性 |
| 1 | deptId | 部门ID | varchar | 50 | PK |
| 2 | deptName | 部门名称 | varchar | 100 | 非空 |
| 3 | description | 部门描述 | text |  |  |

1. **系统实现**

**（展示系统各个模块的界面，暂时不写）**

1. **课设总结（暂时不写）**

**附录 类代码和主函数**

### 管理员类（Administrator）

public class Administrator {

private String userId;

private String adminNickname;

private String contactInformation;

private String userPwd;

// 构造器, getter 和 setter 方法

// 省略其他方法

}

### 用户类（User）

public class User {

private String userId;

private String userPwd;

private String userName;

private String userNickname;

private String userMajor;

private String userSex;

private String userPhone;

private String userEmail;

private String position;

private String department;

// 构造器, getter 和 setter 方法

// 省略其他方法

}

### 用户标准类（UserCriteria）

public class UserCriteria {

private String userId;

private String userName;

private int userAge;

private String userSex;

private String position;

private String department;

// 构造器, getter 和 setter 方法

// 省略其他方法

}

### 用户信息类（UserInfo）

public class UserInfo {

private String userName;

private String userNickname;

private String userMajor;

private String userSex;

private String userPhone;

private String userEmail;

private String position;

private String department;

// 构造器, getter 和 setter 方法

// 省略其他方法

}

### 职位类（Position）

public class Position {

private String positionId;

private String positionName;

private String status;

private String description;

// 构造器, getter 和 setter 方法

// 省略其他方法

}

### 部门类（Department）

public class Department {

private String deptId;

private String deptName;

private String description;

// 构造器, getter 和 setter 方法

// 省略其他方法

}

### 主函数（Main Function）

public class Main {

public static void main(String[] args) {

// 初始化控制器实例

UserController userController = new UserController();

DeptController deptController = new DeptController();

PositionUserController positionUserController = new PositionUserController();

MessageController messageController = new MessageController();

// 模拟用户注册过程

String newUserId = "alice007";

String newUserPwd = "password123";

String newUserEmail = "alice@example.com";

userController.submitRegistrationDetails(newUserId, newUserPwd, newUserEmail);

// 模拟用户登录过程

boolean isLoginSuccessful = userController.submitLoginCredentials(newUserId, newUserPwd);

System.out.println("用户登录：" + (isLoginSuccessful ? "成功" : "失败"));

// 模拟创建一个新部门

String newDeptId = "dept001";

String newDeptName = "研发部";

String newDeptDescription = "负责产品研发";

deptController.create(new Department(newDeptId, newDeptName, newDeptDescription));

// 模拟添加一个职位

String newPositionId = "pos001";

String newPositionName = "Java开发工程师";

String newPositionDescription = "负责Java后端开发";

positionUserController.submitPositionInfoChanges(newPositionId, "ADD", new Position(newPositionId, newPositionName, "OPEN", newPositionDescription));

// 模拟发送一条消息

String recipientId = "bob123";

String messageContent = "欢迎加入公司！";

messageController.submitMessage(recipientId, messageContent);

}

}

// 假设控制器类都已正确定义如下

public class UserController {

private Map<String, User> userMap = new HashMap<>();

// ... 方法定义省略

public void submitRegistrationDetails(String userId, String userPwd, String userEmail) {

User newUser = new User(userId, userPwd, userEmail);

userMap.put(userId, newUser);

}

public boolean submitLoginCredentials(String userId, String userPwd) {

// ... 登录逻辑省略

return true; // 假设登录总是成功的

}

}

public class DeptController {

private List<Department> departments = new ArrayList<>();

public void create(Department dept) {

departments.add(dept);

}

// ... 其他方法省略

}

public class PositionUserController {

private Map<String, Position> positionMap = new HashMap<>();

public void submitPositionInfoChanges(String positionId, String operation, Position position) {

// ... 职位修改逻辑省略

}

// ... 其他方法省略

}

public class MessageController {

public void submitMessage(String recipientId, String content) {

// ... 消息发送逻辑省略

}

// ... 其他方法省略

}