**SOA期末项目说明文档（协作版）**

第6组

1853558 周长空

1951107 毛健羽

1951574 吴渭

1952377 蒋汶霖

1954083 水游龙

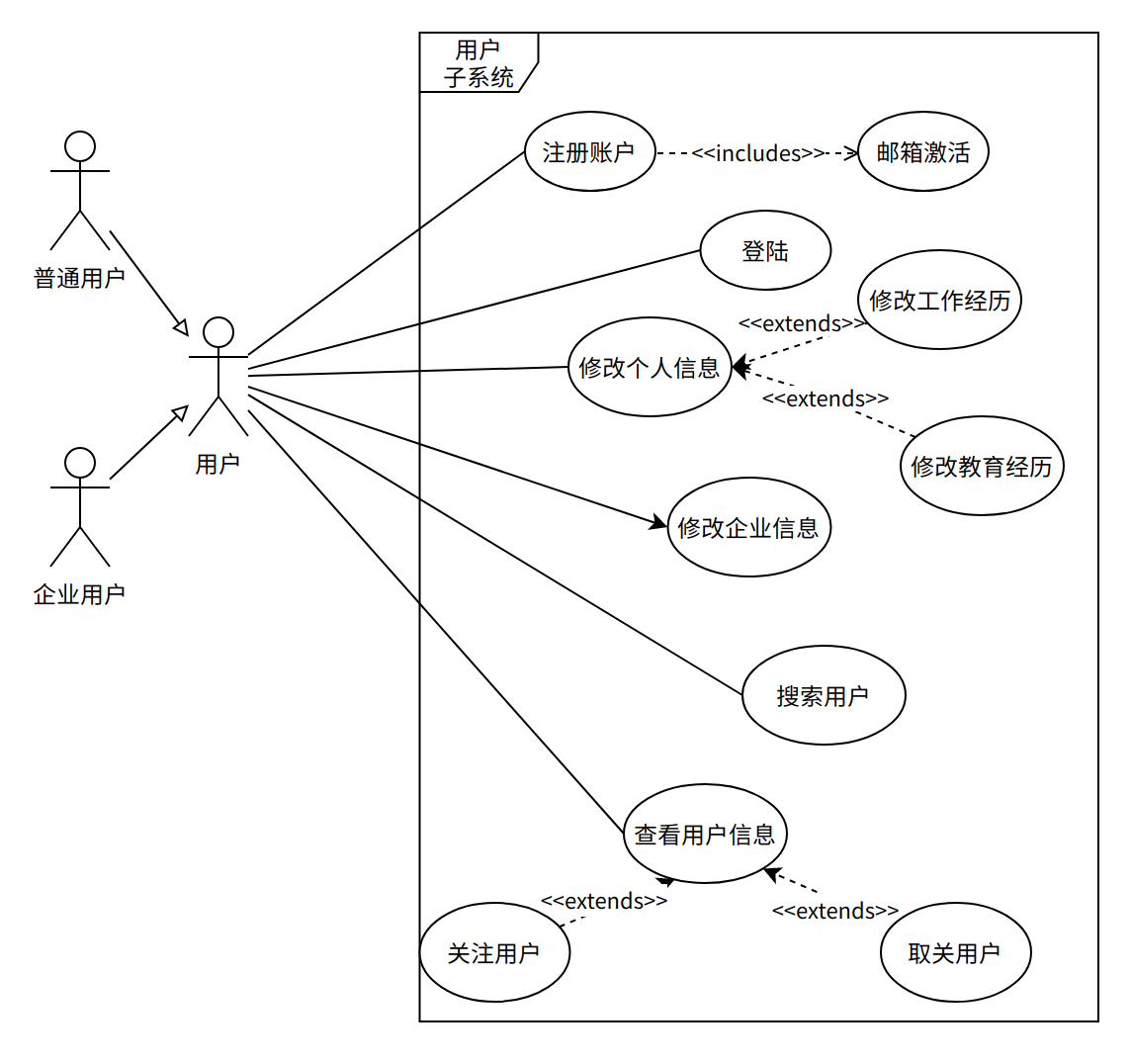
**一、项目简介**

随着近几年来国内创业热潮的兴起，许多新中小企业应运而生。招聘雇主数量稳步增长，人才需求持续旺盛。企业雇主倾向于通过一种高效的方式尽快招揽与岗位职责匹配的人才。本项目名称为职业发展社交平台,是一款专注互联网招聘求职和社交的平台。通过搭建⼀个⾯向全国的招聘信息发布平台，⽤⼈单位可以在平台上发布招聘或实习岗位；同时创建动态模块，通过订阅关注模式用户可及时获取相关推送。面向的用户范围从最初的同济大学的在校师生拓展为全国的注册求职用户。

**二、需求分析**

1. **用户子系统**

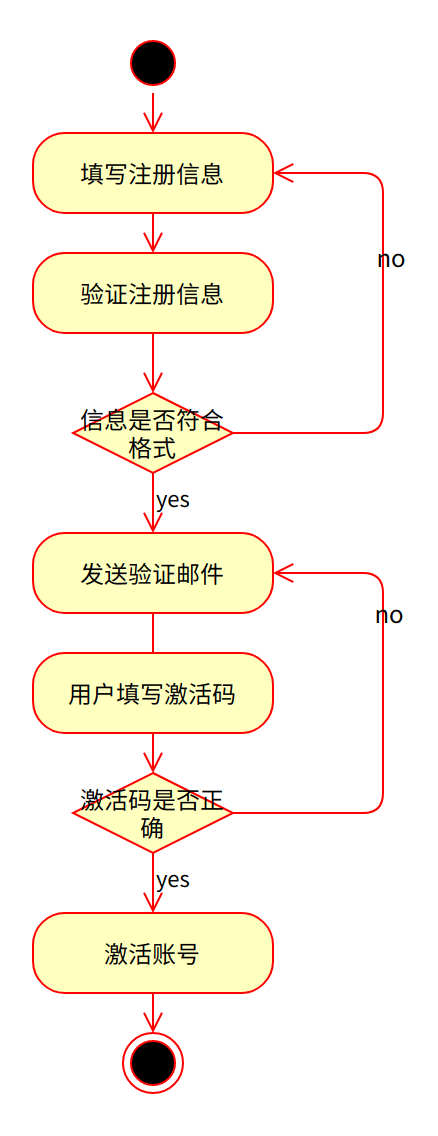
用户子系统的使用者包括普通用户和企业用户（他们共同继承于同一基类）。两者都拥有相同的注册登录功能，但就个人信息的编辑方面略有差异；两者都可以搜索并与其他用户进行交互



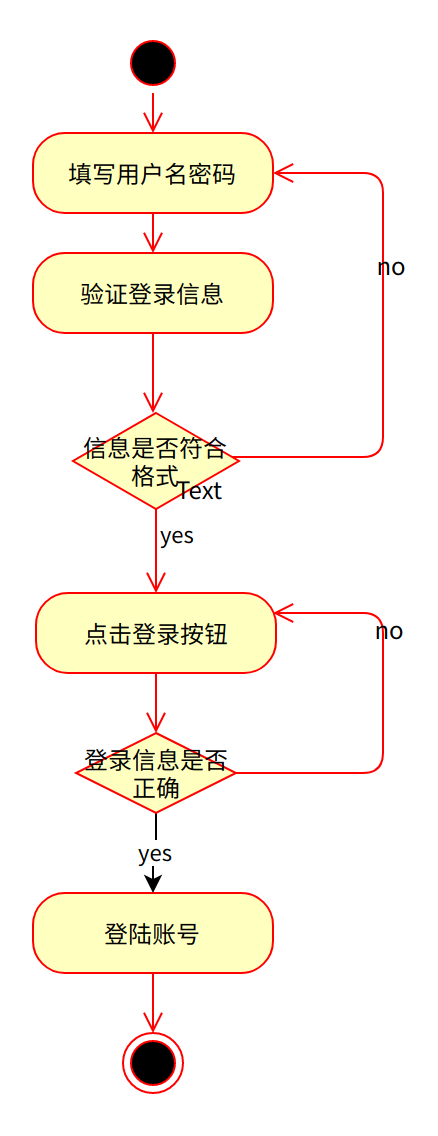
**功能建模**

**注册账户**

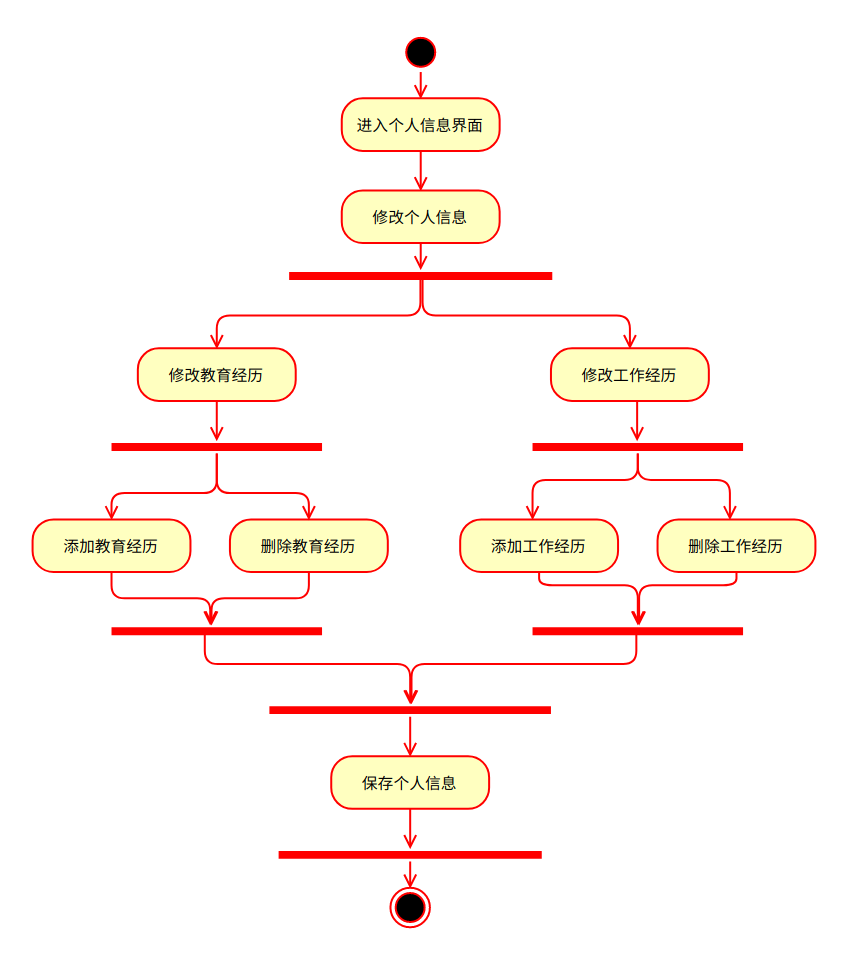
该活动图描述了用户注册账户的整体流程，从开始填写注册信息到成功激活账户结束活动，期间涉及了用户User和邮箱Email实体类，注册表单接口边界类和用户模块控制类



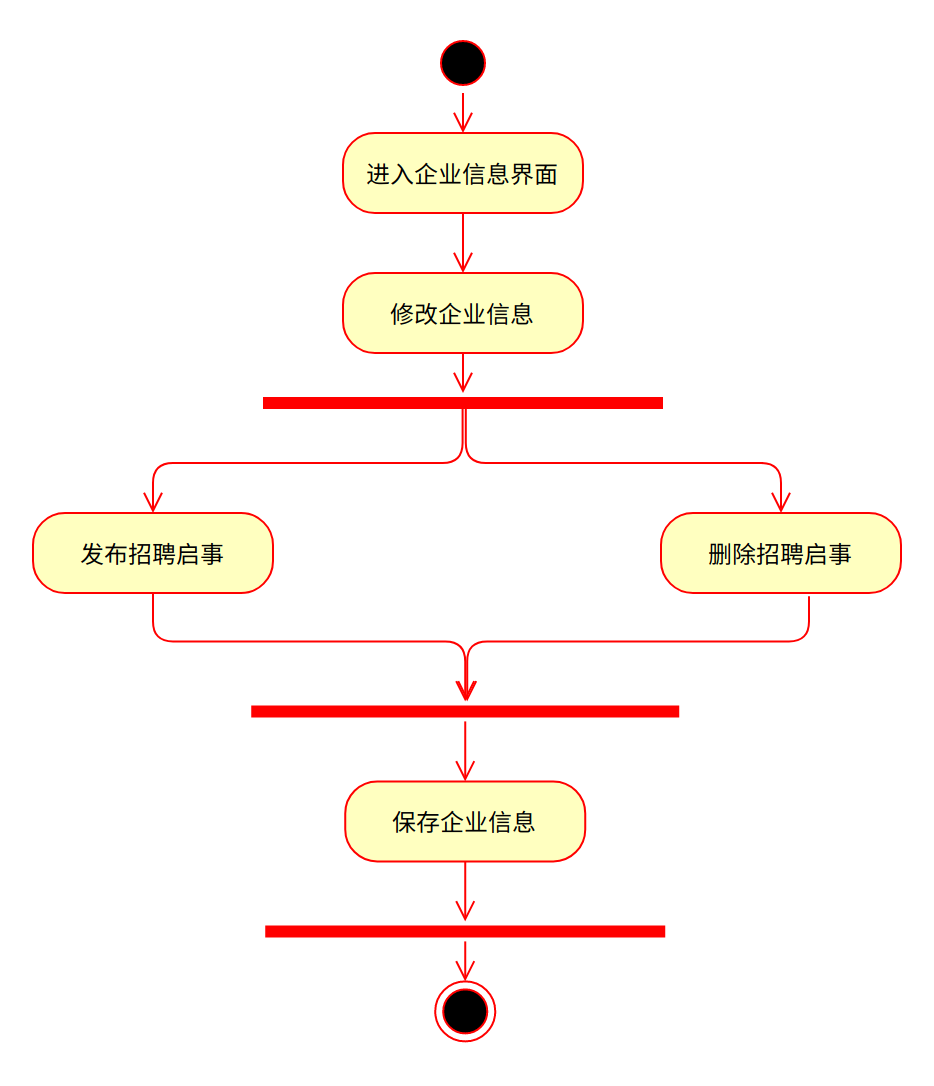
**登录账户**



**修改个人信息**



**修改企业信息**

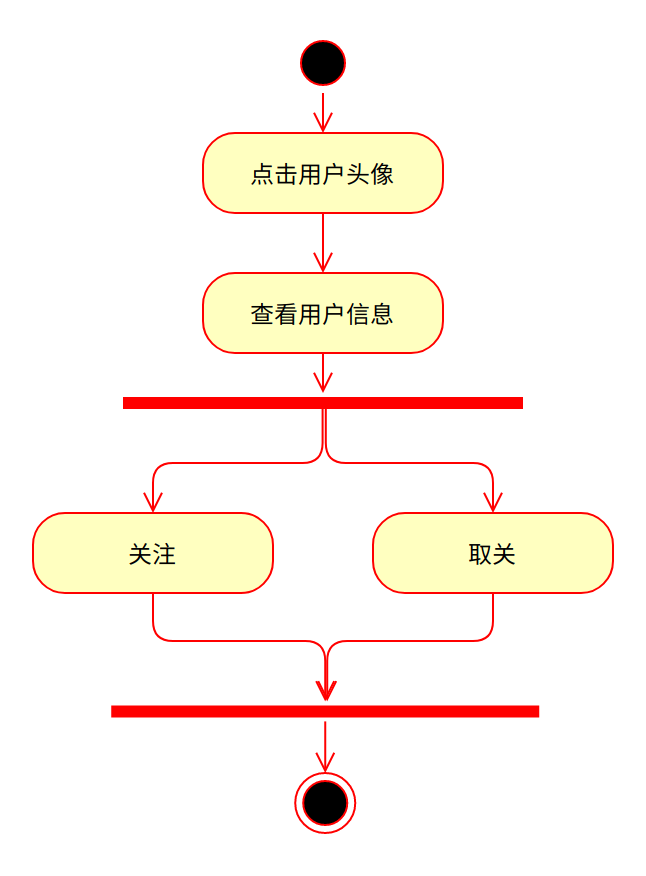


**搜索用户**

该活动图描述了用户搜索的整体流程，用户可以选择搜索的用户类型（普通用户或企业用户）、筛选条件及排序方式，在返回搜索列表，用户还可以点击用户头像，进入“查看用户信息”的扩展事件；此活动涉及用户信息实体类和用户交互的边界类

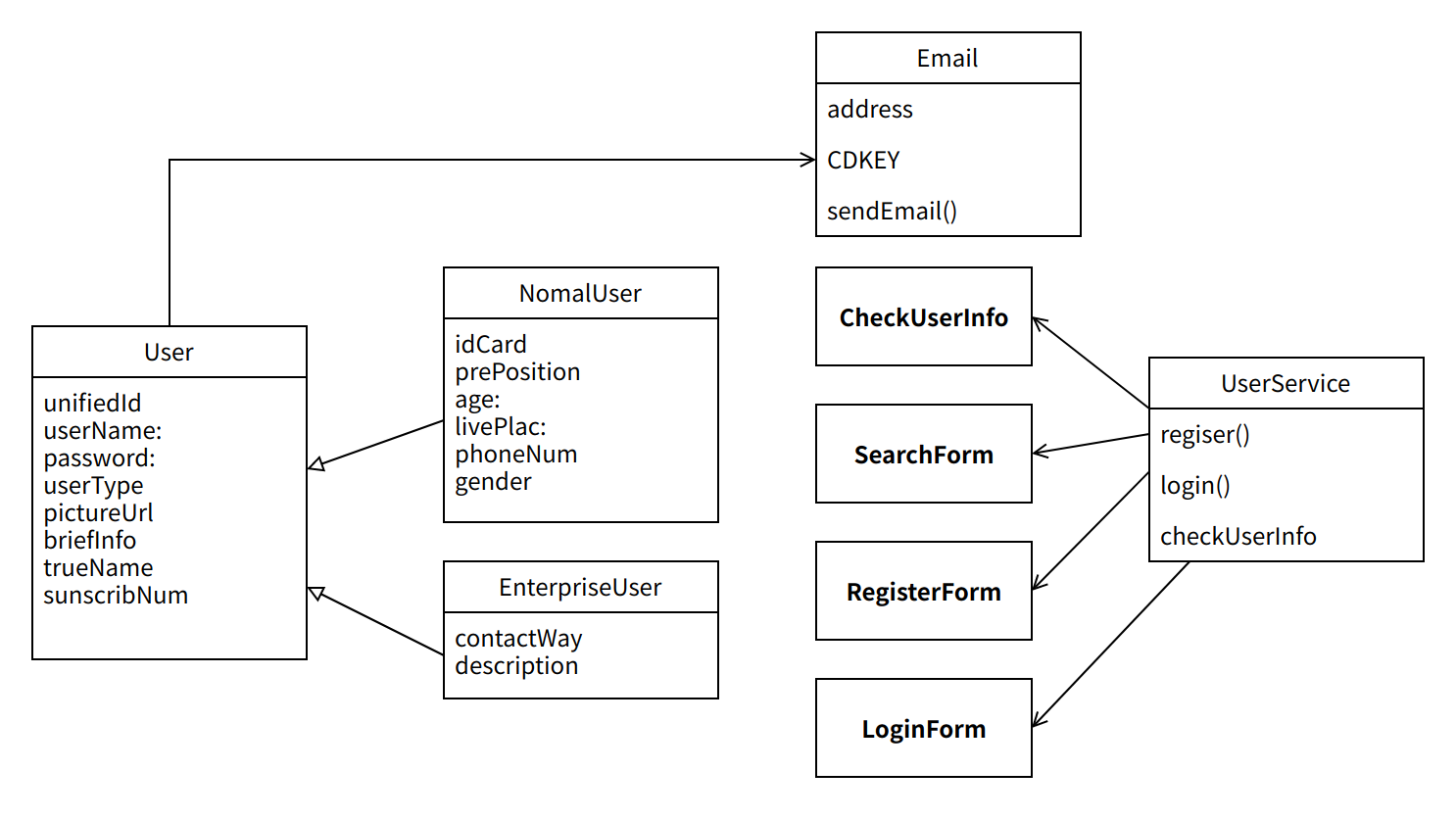


**查看用户信息**

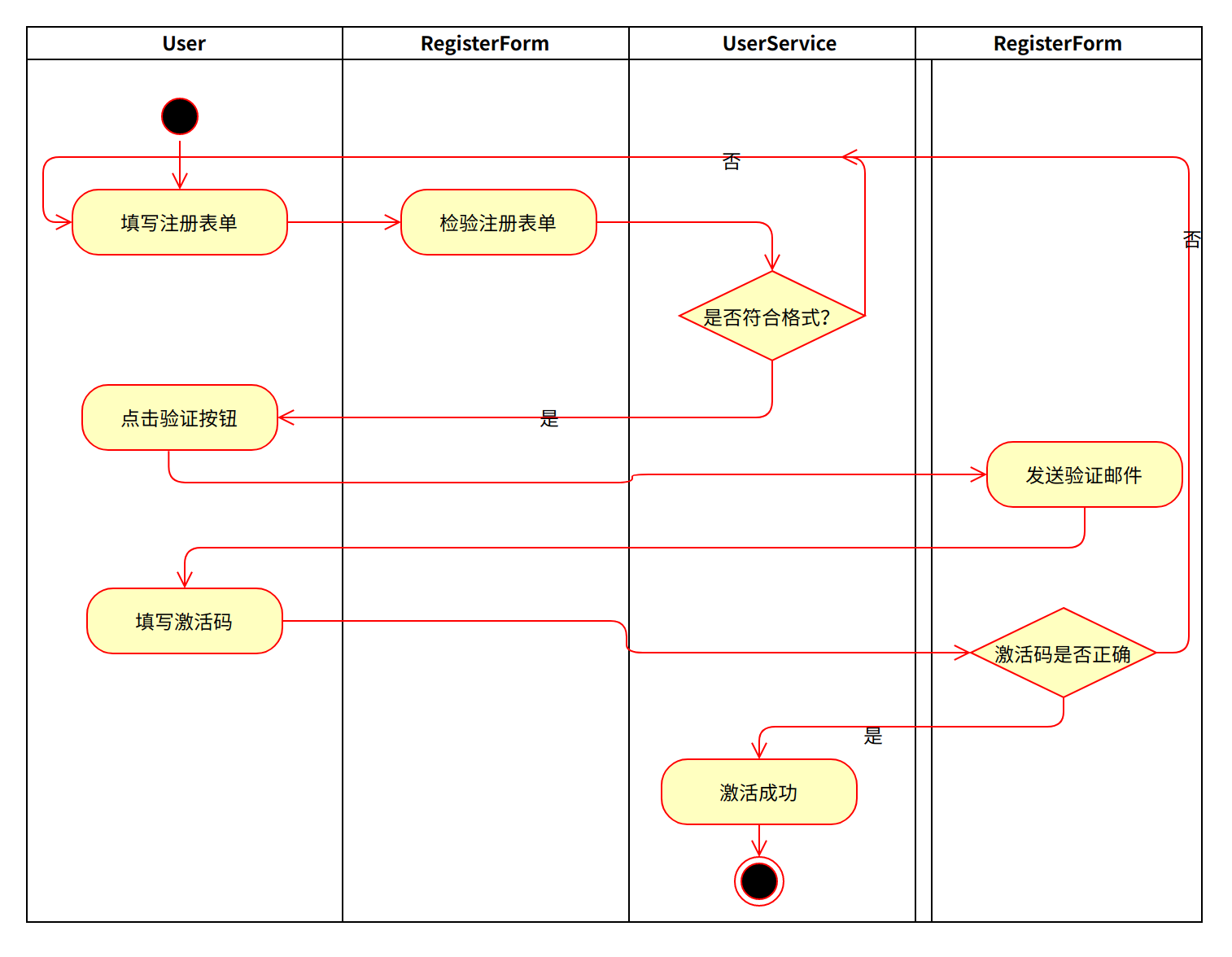


**数据建模**

在该类图中共有三种分析类：实体类、边界类和控制类。其中实体类存储具体实例的信息，包括邮箱，基础用户，普通用户和企业用户（由基础用户泛化而来）；边界类负责实体的交互并承担逻辑功能的执行任务，包括注册、登录的表单以及负责实现搜索功能的接口；控制类负责用户模块的统筹协调，编写业务逻辑并为其余分析类提供服务。



**注册账户**



**搜索用户**

无法复制加载中的内容

**修改个人信息**

无法复制加载中的内容

**修改企业信息**

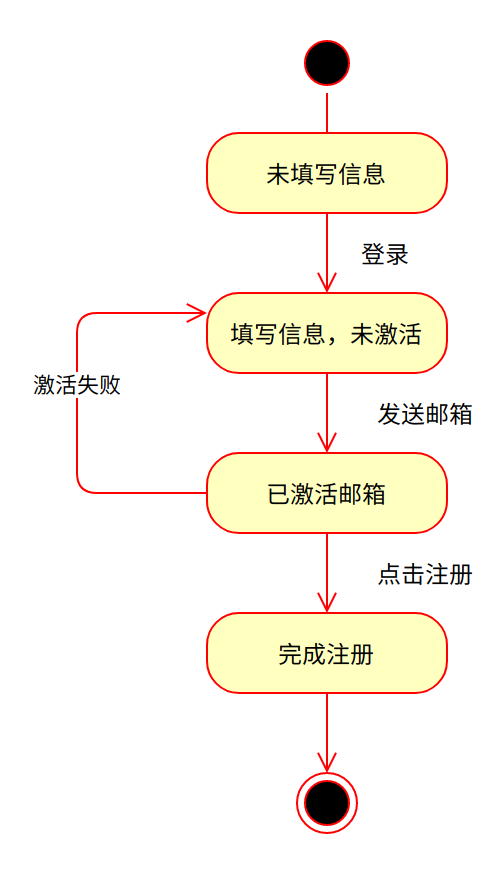
无法复制加载中的内容

**行为建模**

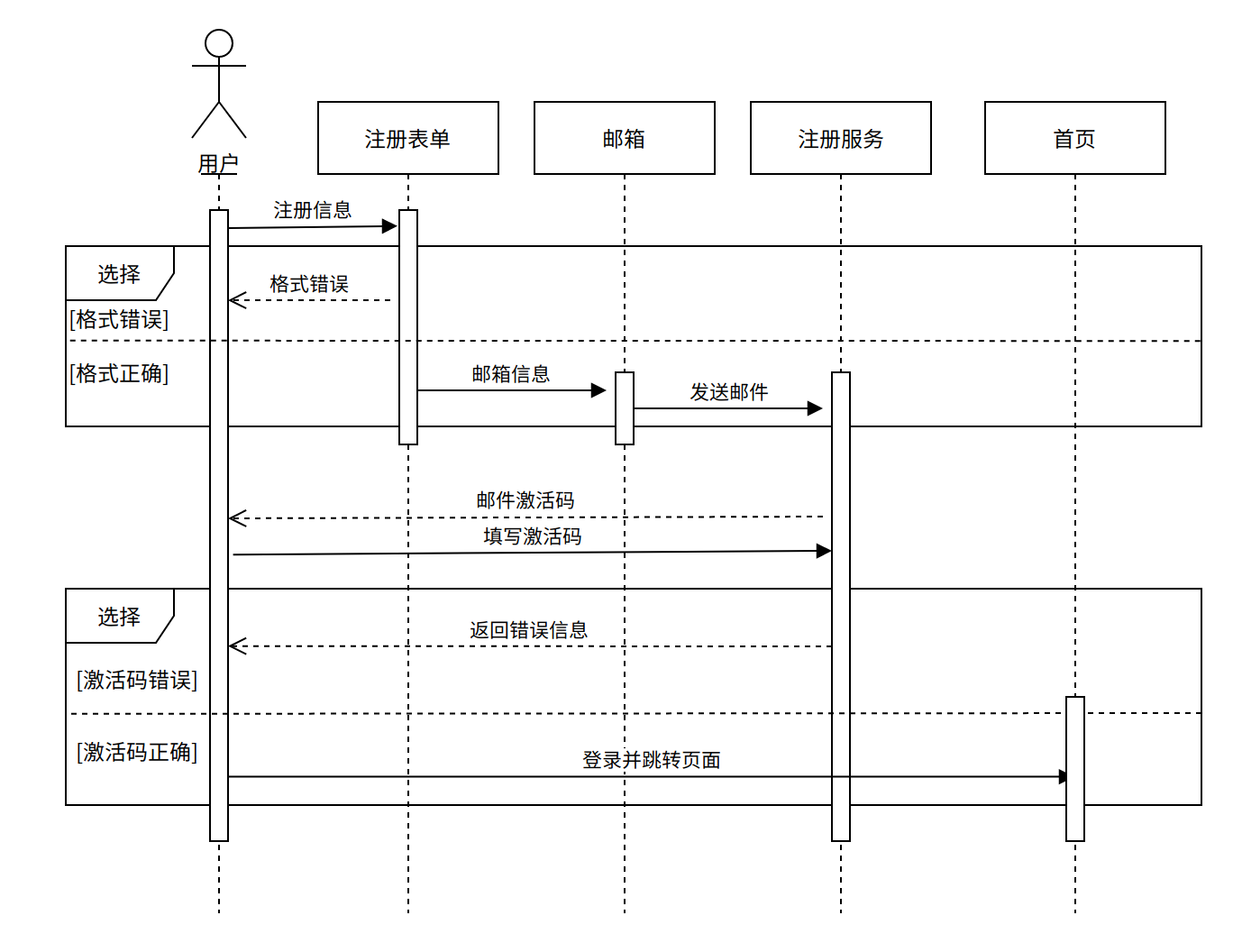
**注册账户**

**用户类（User）**

该状态图描述了在“注册”用例中，用户类的状态变化流程

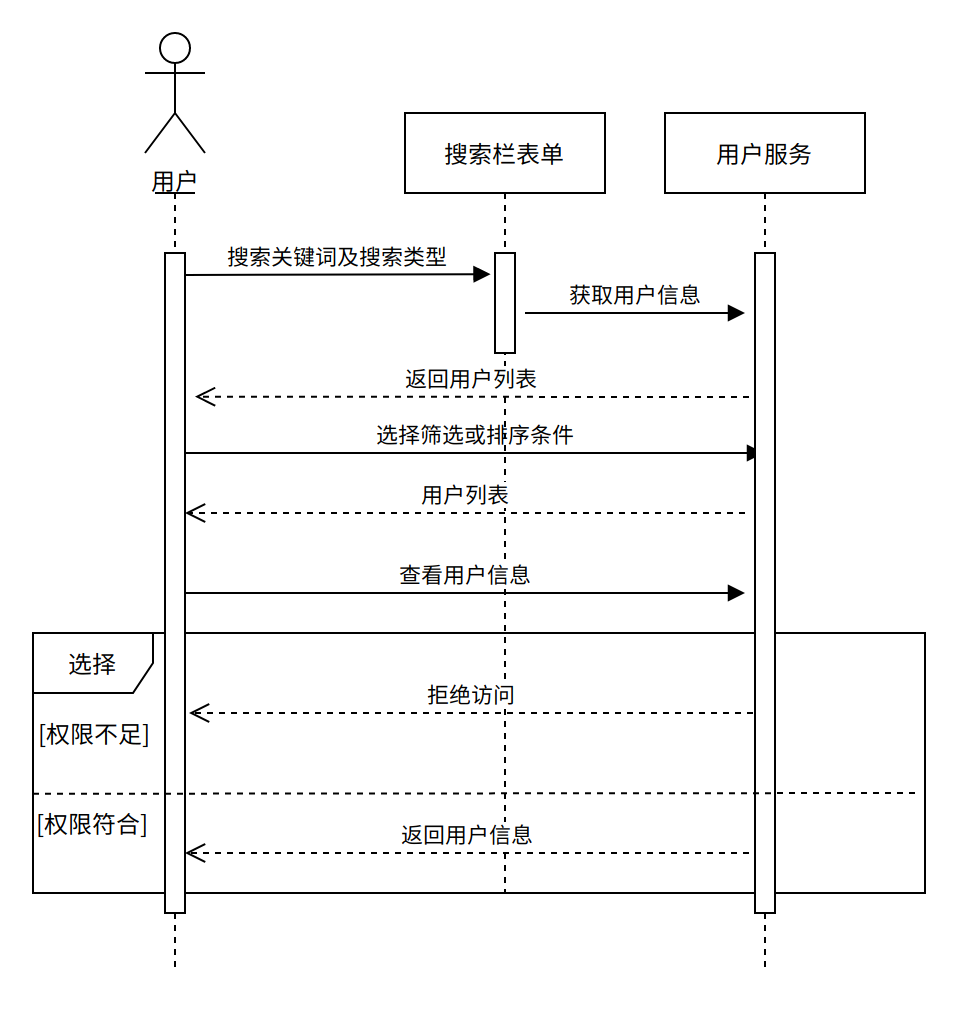


该时序图涉及用例为“注册账户”，涉及对象包含两个实体类（用户、邮箱），一个边界类（注册表单）、一个控制类（注册服务）和一个模块外部系统（首页）。该图规定了“注册账户”这一用例中各个活动的执行顺序和各对象间的信息交互。

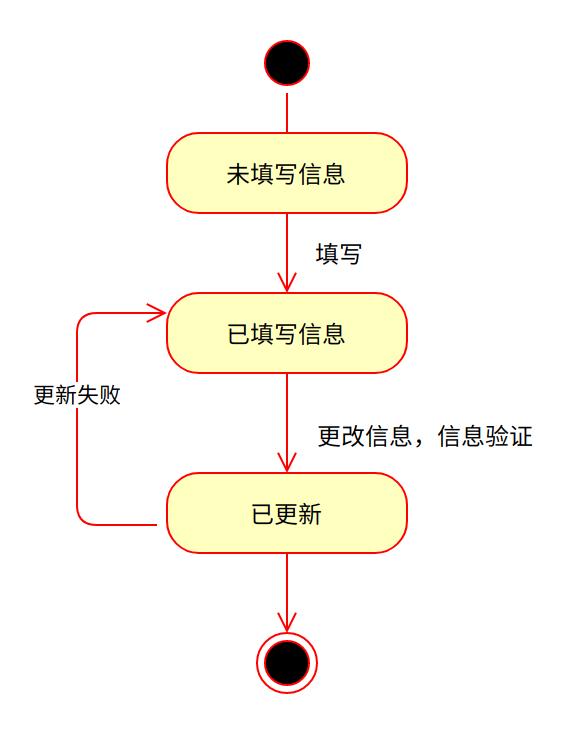


**搜索用户**

该时序图描述了“搜索用户”这一用例中的活动时间顺序及对象信息交互，涉及到了用户实体类，搜索栏表单边界类和用户服务控制类。

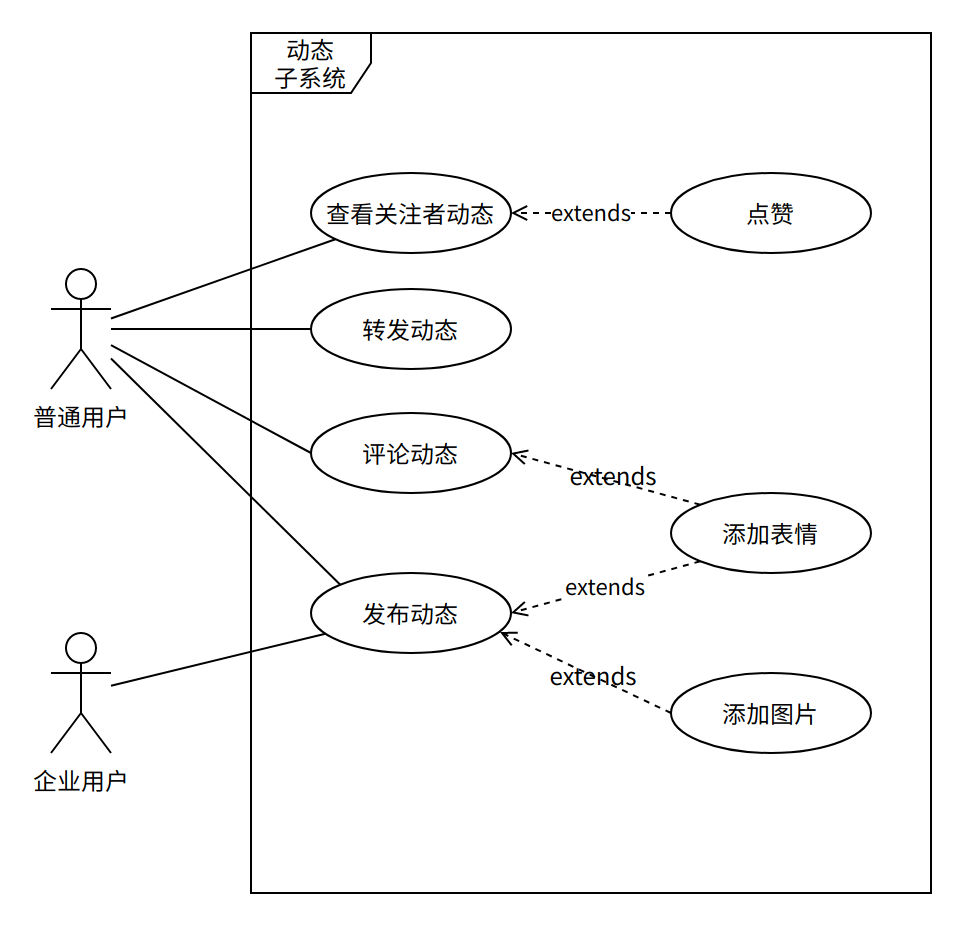


**UserInfo（用户信息类）**



1. **动态子系统**

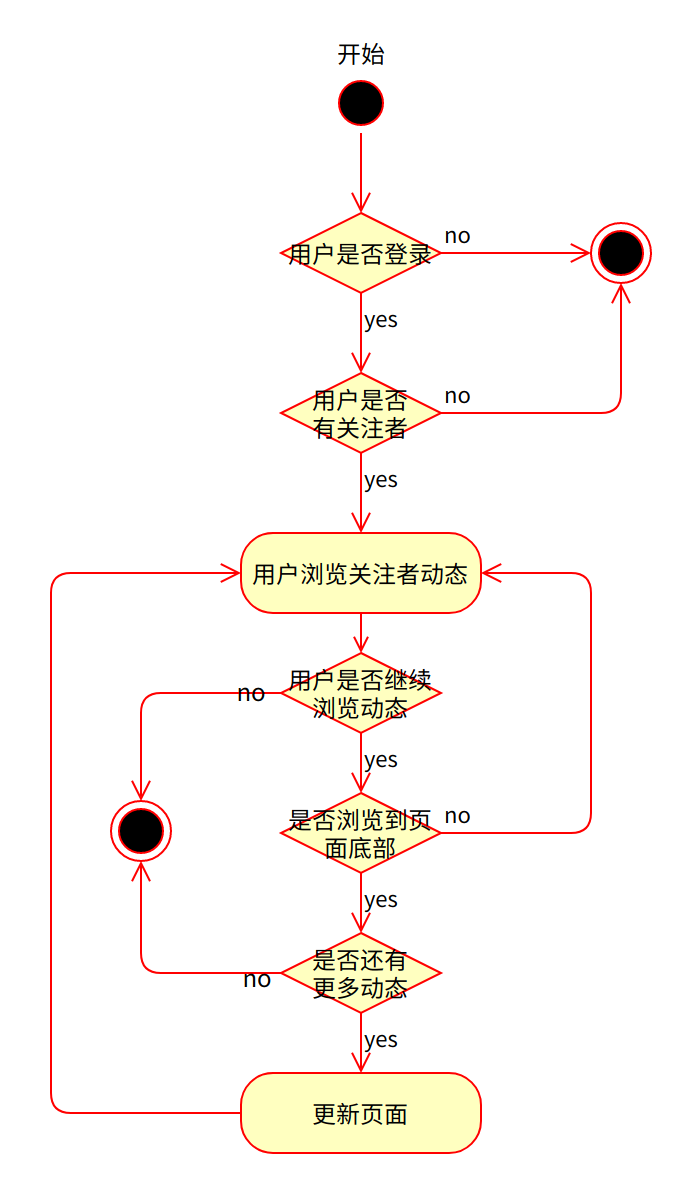
动态子系统主要使用者为普通用户与企业用户。两种用户均可发布动态，在发布时可选择添加图片或者添加表情。普通用户还可以评论动态，转发动态以及查看关注者动态。



**功能建模**

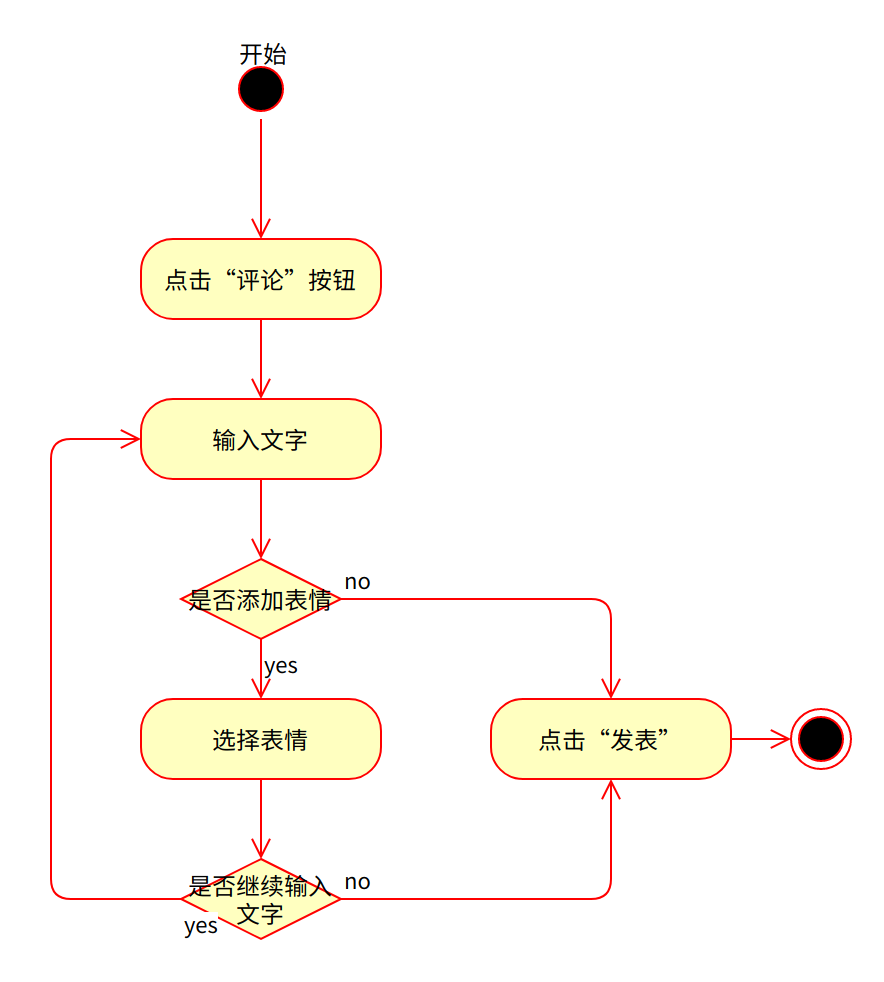
**查看关注者动态**

用户登陆后，如果此用户有关注者，系统就会在动态浏览页面推送关注者发布的动态。用户可以浏览这些动态，当浏览到页面底部时，如果还有更多动态，系统就会加载一定数量的动态，用户可以继续浏览。



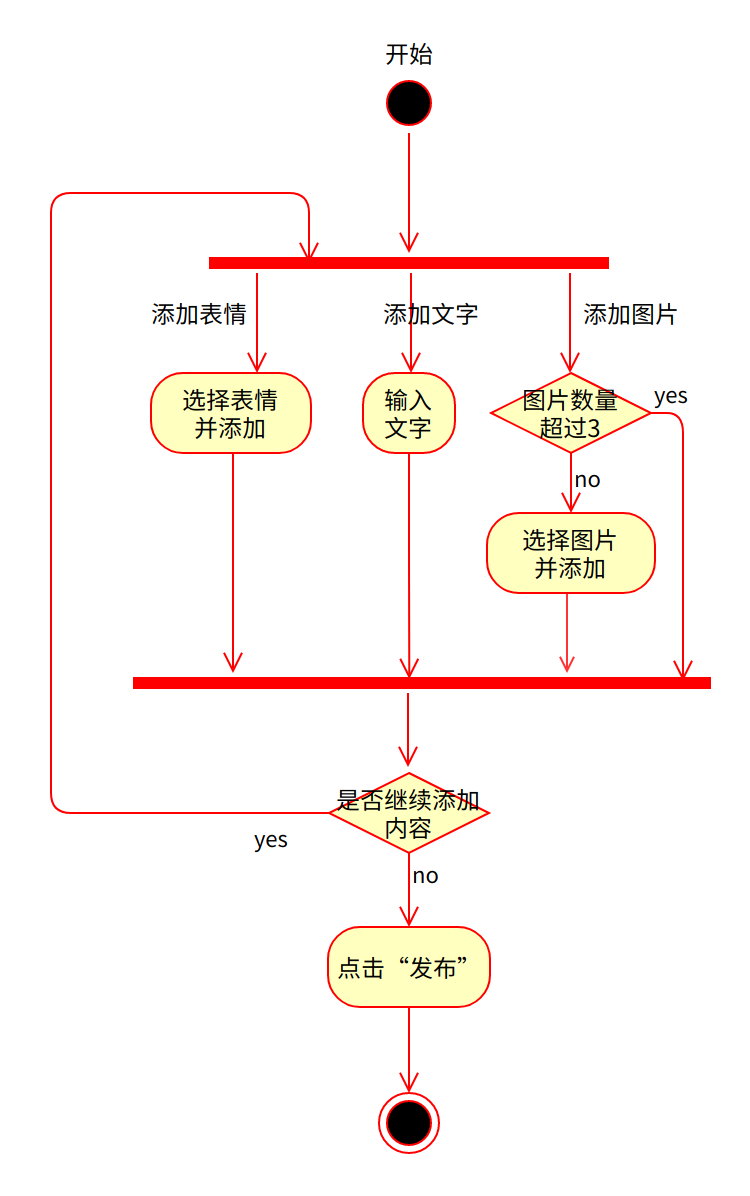
**发表评论**

用户浏览到某一条动态时，可以点击“评论”按钮展开评论。用户可以在输入框输入文字，同时也可以点击输入框下方按钮添加表情。评论输入完成后可以点击发表。



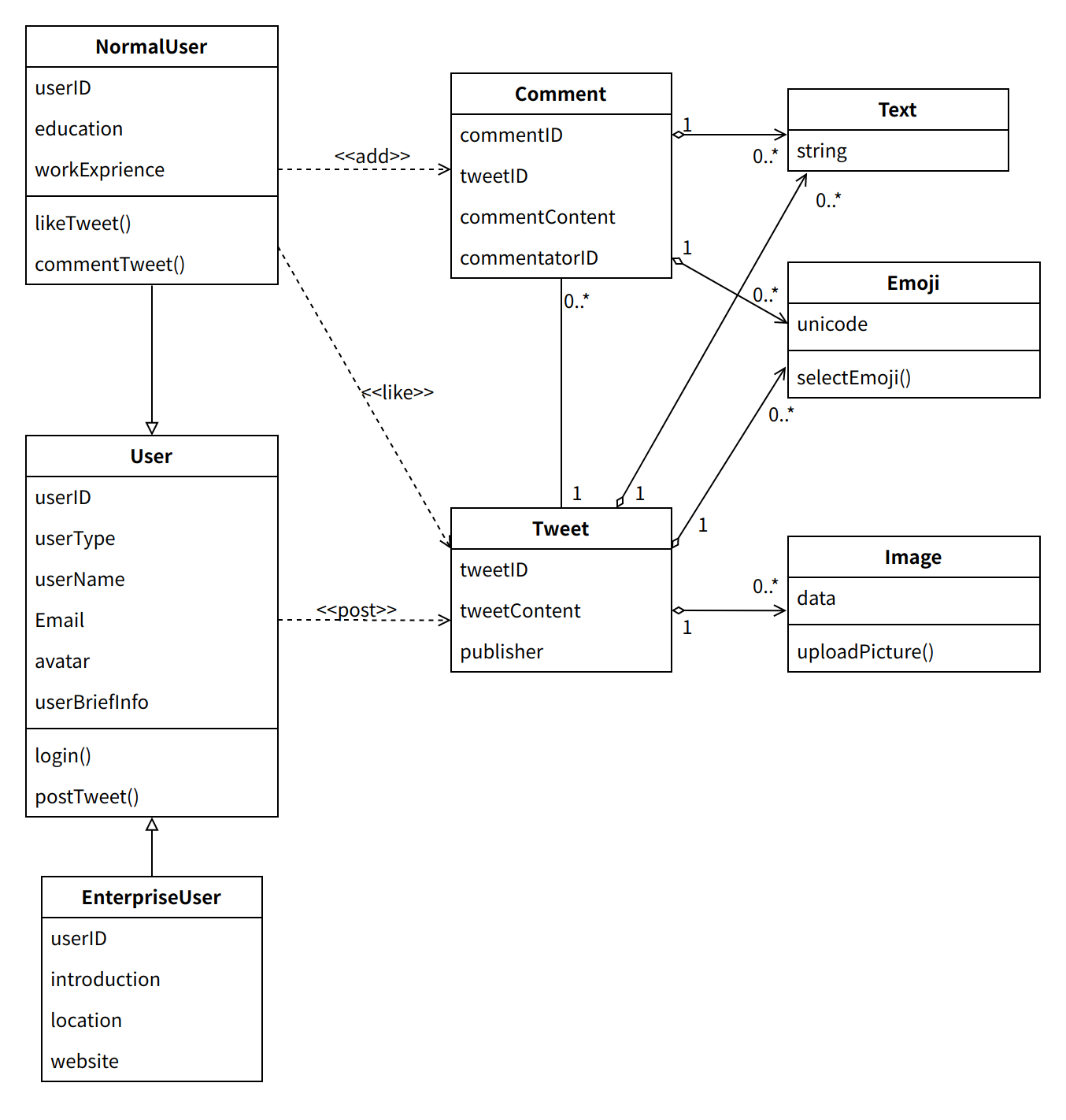
**发布动态**

用户登录后可以发布动态。用户可以分别在输入框输入文字，用表情选择器选择表情或者使用文件选择器添加图片。内容输入完成后可以点击发布。



**数据建模**

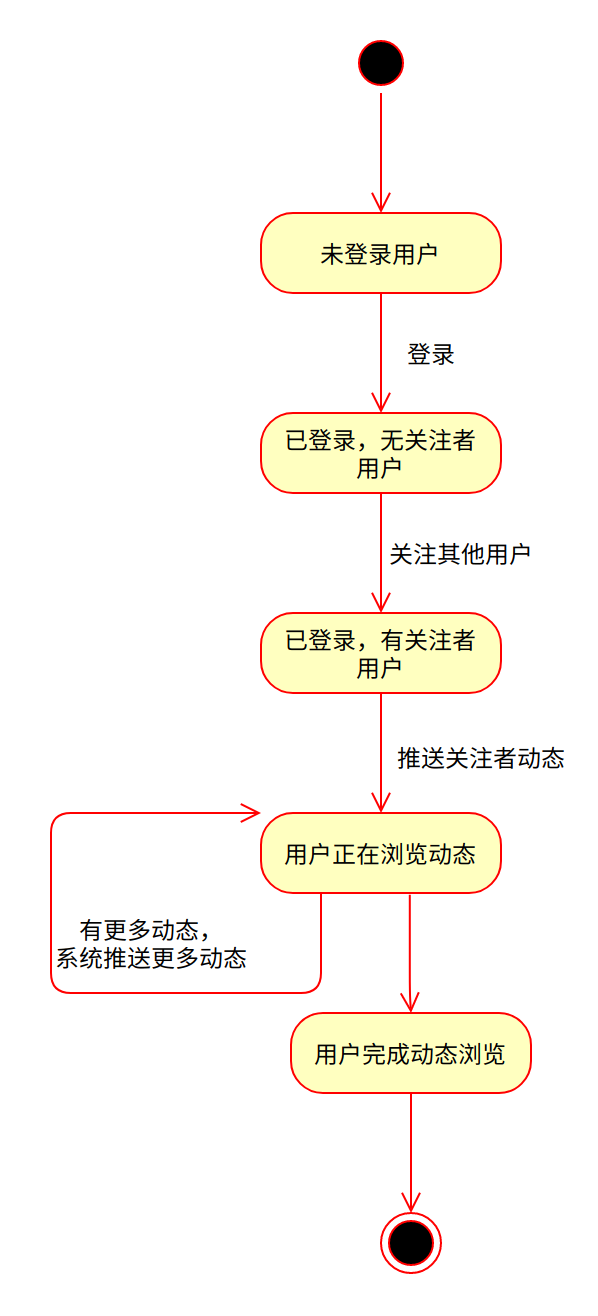
普通用户（NormalUser）与企业用户（EnterpriseUser）均继承自基类用户（User）。User可以发布动态（Tweet），普通用户还可以点赞Tweet或添加评论（Comment）。Comment与Tweet相关联，一个Tweet可以对应0或多个Comment。同时，一个Comment可以有0或多个文字（Text）与表情（Emoji），一个Tweet也可以有0或多个Text，Emoji与图片（Image）。



**行为建模**

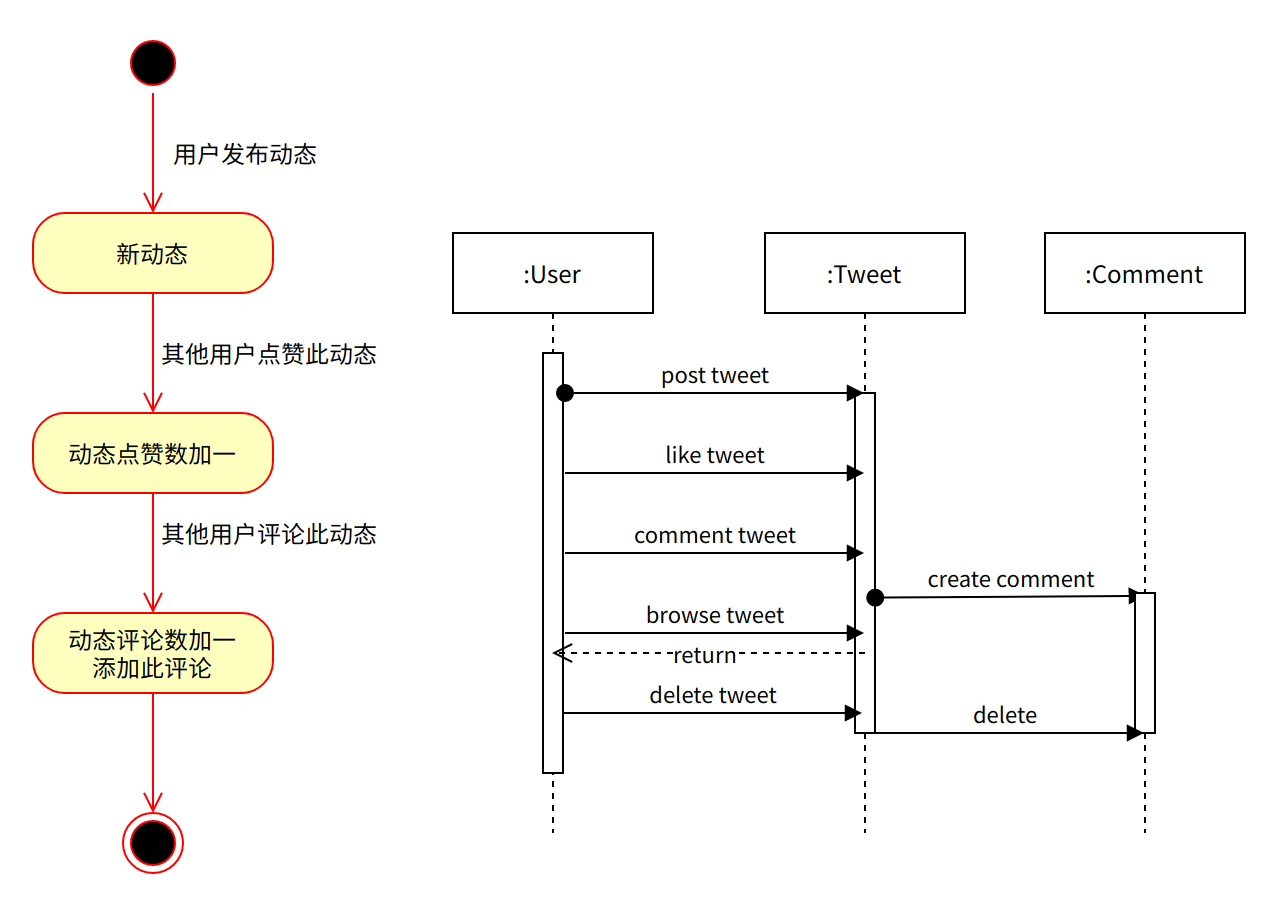
**用户类（User）**

在动态子系统中，如果一个用户登录后，没有关注者，则系统会为此状态用户推送用户关注。如果用户有关注者，则系统会为这种状态用户推送关注者动态。用户将推送动态浏览完毕时，如其关注者有更多动态，则系统会推送更多动态，用户继续浏览。否则用户完成动态浏览状态。



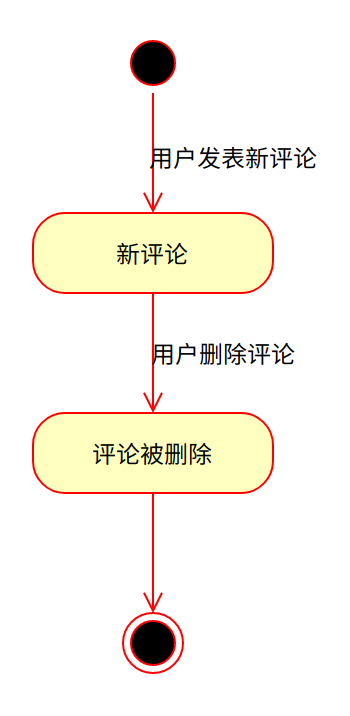
**动态类（Tweet）**

在动态子系统中，用户可以发布动态。其他用户可以点赞此动态，则动态点赞数加一。其他用户也可以添加评论，则动态评论数加一，同时生产新的评论。



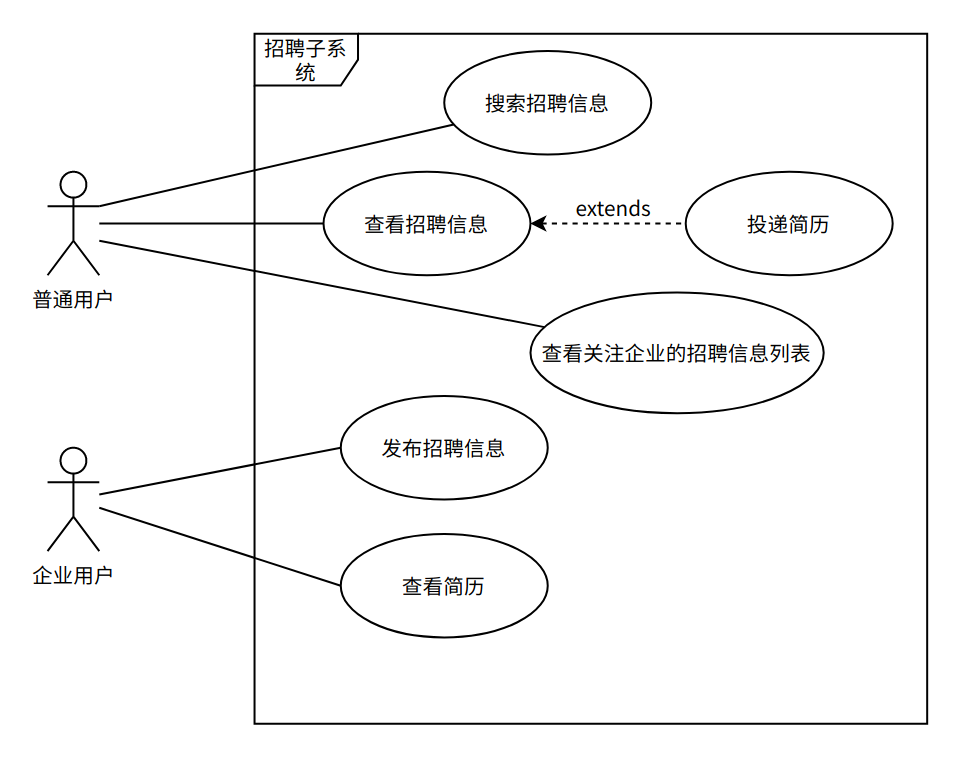
**评论类（Comment）**

用户在浏览动态时可以针对某条动态发表评论。之后也可以随时删除自己发表的评论。



1. **招聘子系统**

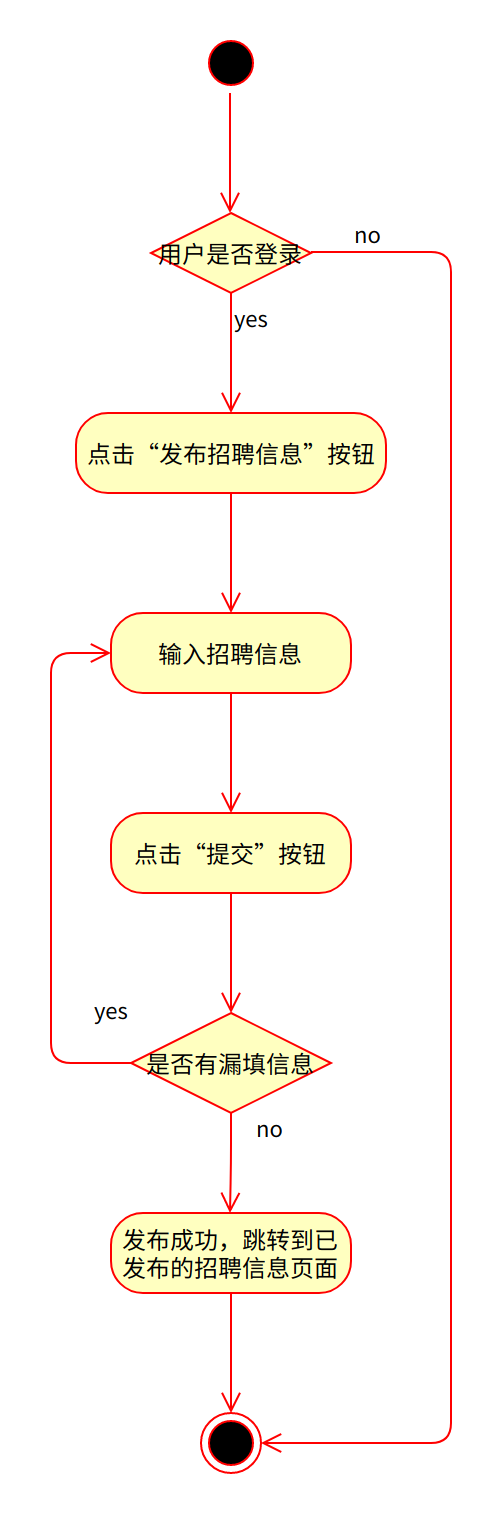
招聘子系统的使用者有普通用户和企业用户。普通用户可以查看企业用户发布的招聘信息，查看招聘信息时可以对招聘信息投递简历。企业用户可以发布招聘信息，也可以查看自己发布的招聘信息所收到的简历。



**功能建模**

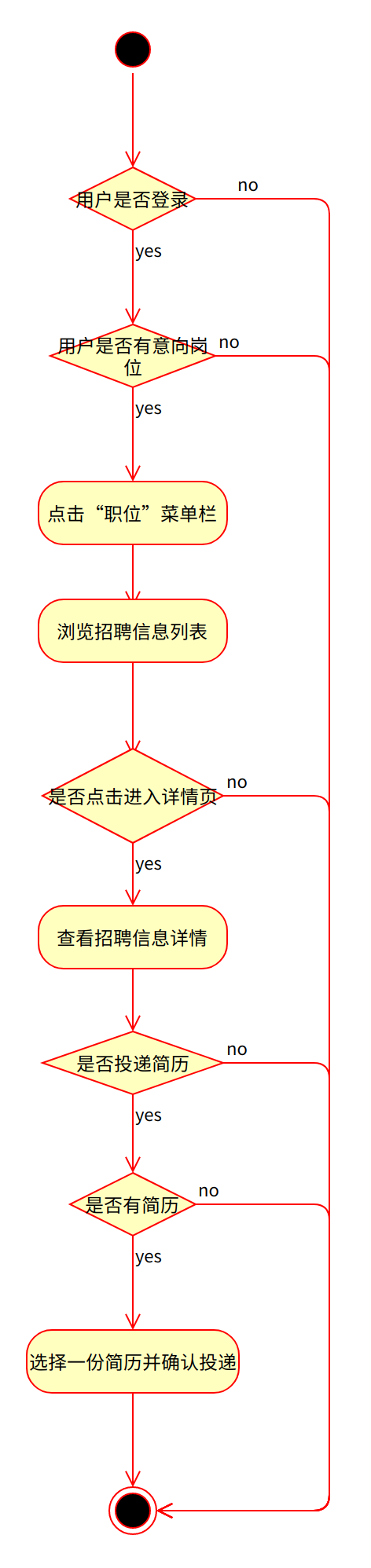
**发布招聘信息**

用户登录后，在自己的个人主页点击“发布招聘信息”按钮，并输入要发布的招聘信息，然后提交。系统会检查输入的信息是否有漏填的项目，如果有则提示用户继续输入完整信息，如果没有则招聘信息发布成功，跳转到用户已发布的招聘信息页面。



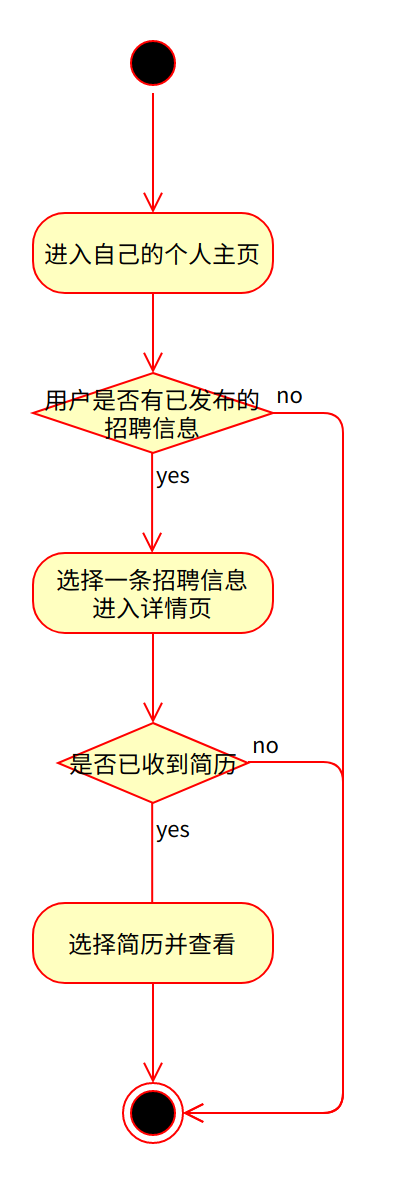
**查看招聘信息**

用户登录后点击“职位”菜单栏，浏览企业发布的招聘信息列表。浏览过程中可以点击某条招聘信息进入其详情页，在详情页中可以选择投递简历。投递简历时若没有简历则显示提示并提供跳转到个人信息页面的选项；若有简历则选择一份简历并确认投递。



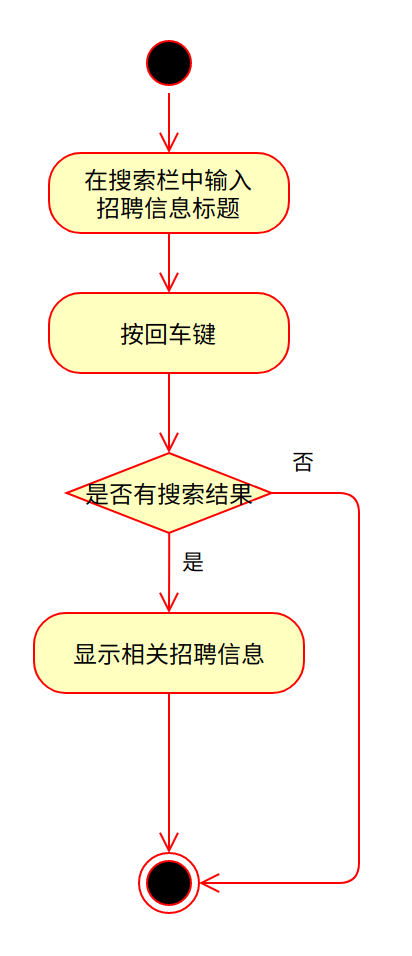
**查看简历**

企业用户进入自己的个人主页，若没有已发布的招聘信息则无法查看简历。用户点击一条招聘信息进入其详情页，然后点击“查看简历”按钮，进入该招聘信息的简历列表。如果列表不为空，则可以选择一份简历并查看。



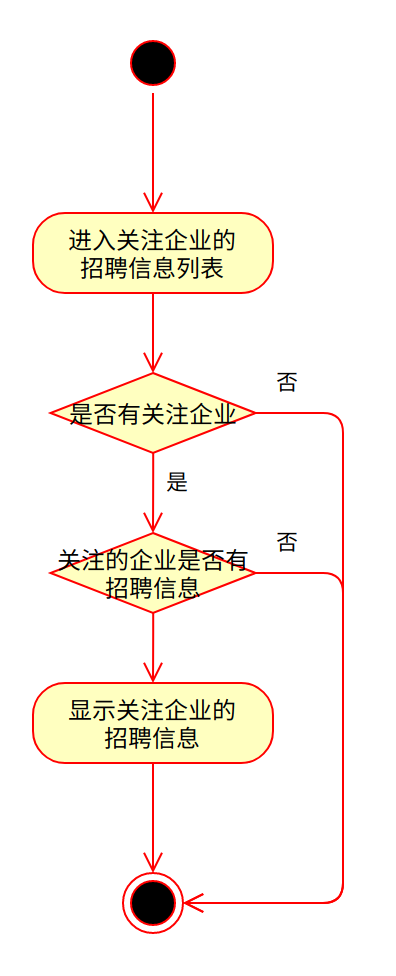
**搜索招聘信息**

用户在搜索栏中输入要搜索的招聘信息标题，并按下回车进行搜索。如果搜索到了相关招聘信息，则会显示相关招聘信息的列表。



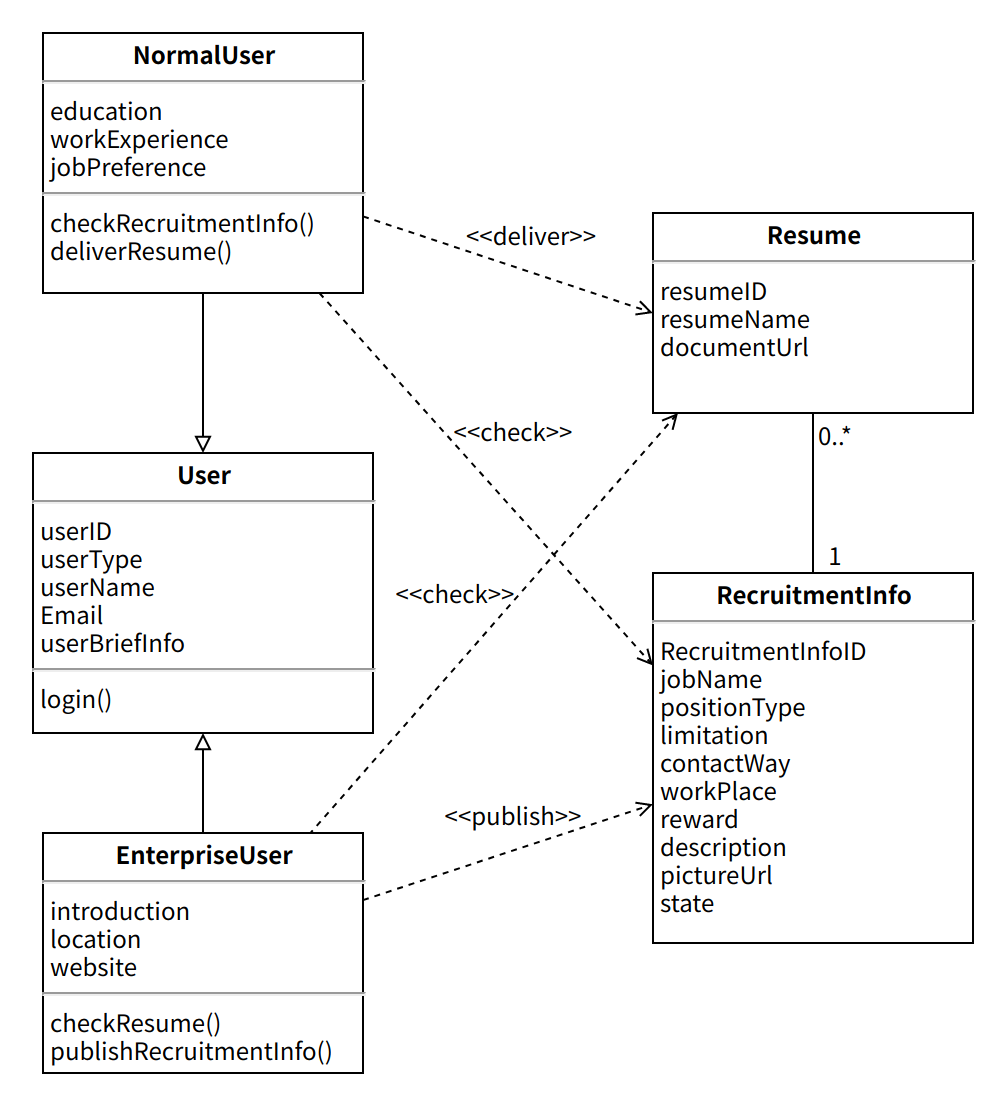
**查看关注企业的招聘信息列表**

用户关注企业后，可以查看所有关注的企业已发布的招聘信息列表。



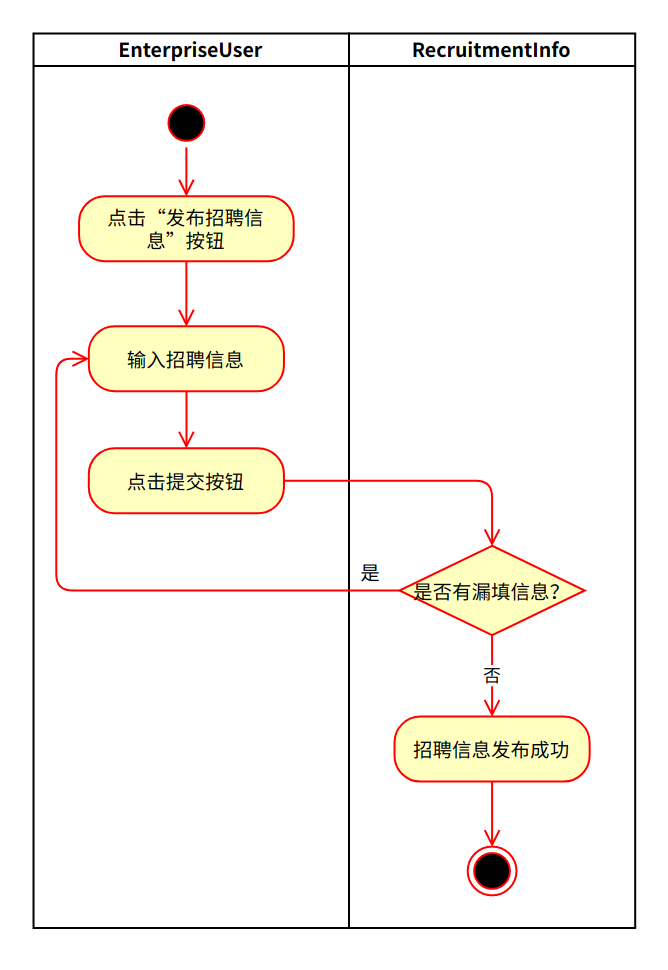
**数据建模**

招聘子系统共有五个实体类，其中普通用户类（NormalUser）和企业用户类（EnterpriseUser）继承自用户类（User）。招聘信息（RecruitmentInfo）与简历（Resume）关联，一个招聘信息（RecruitmentInfo）对应0到多个简历。普通用户可以进行查看招聘信息（checkRecruitmentIfo）和投递简历（deliverResume）两种操作。企业用户可以进行发布招聘信息（publishRecruitmentInfo）和查看简历（checkResume）两种操作。

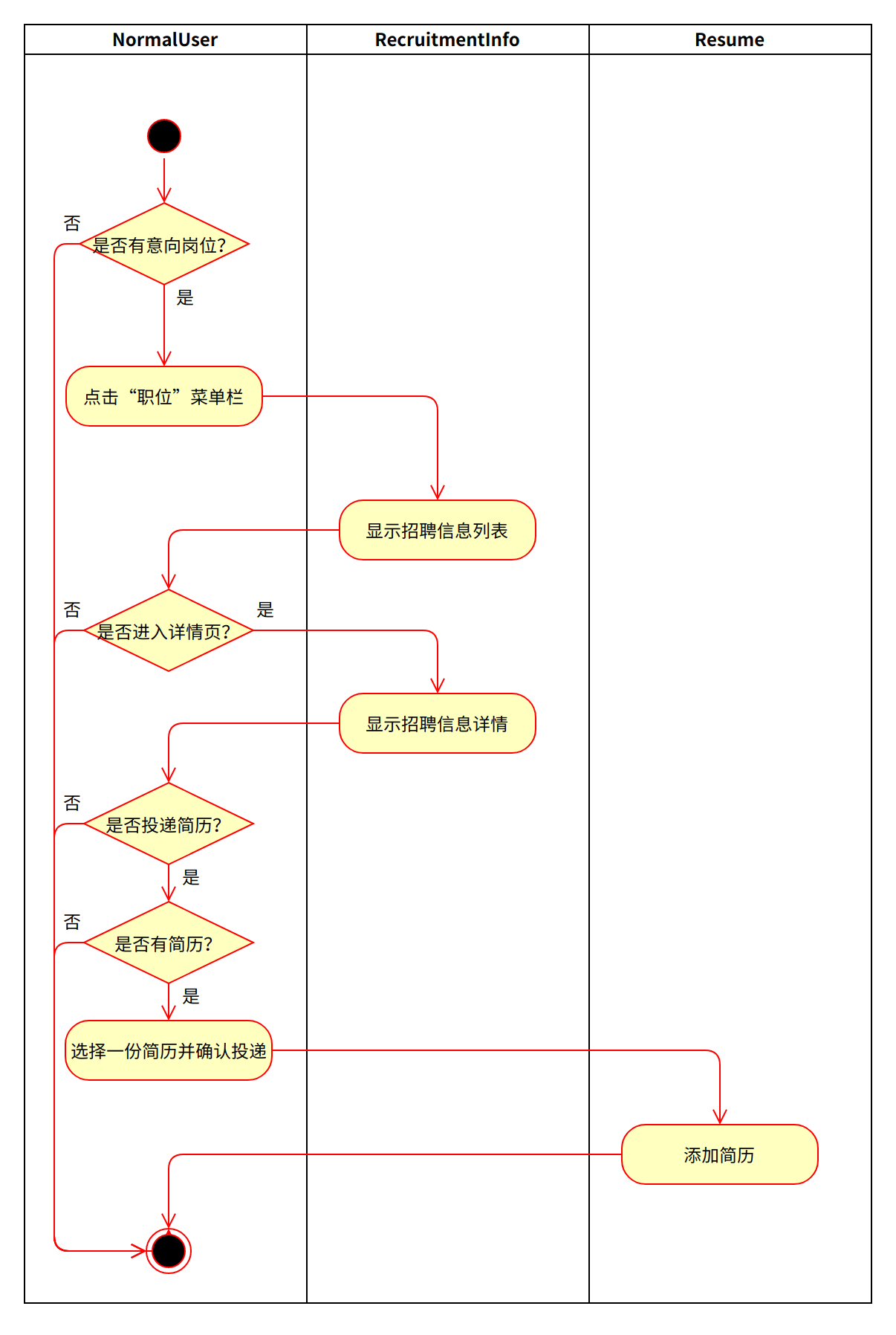


各个用例的泳道图：

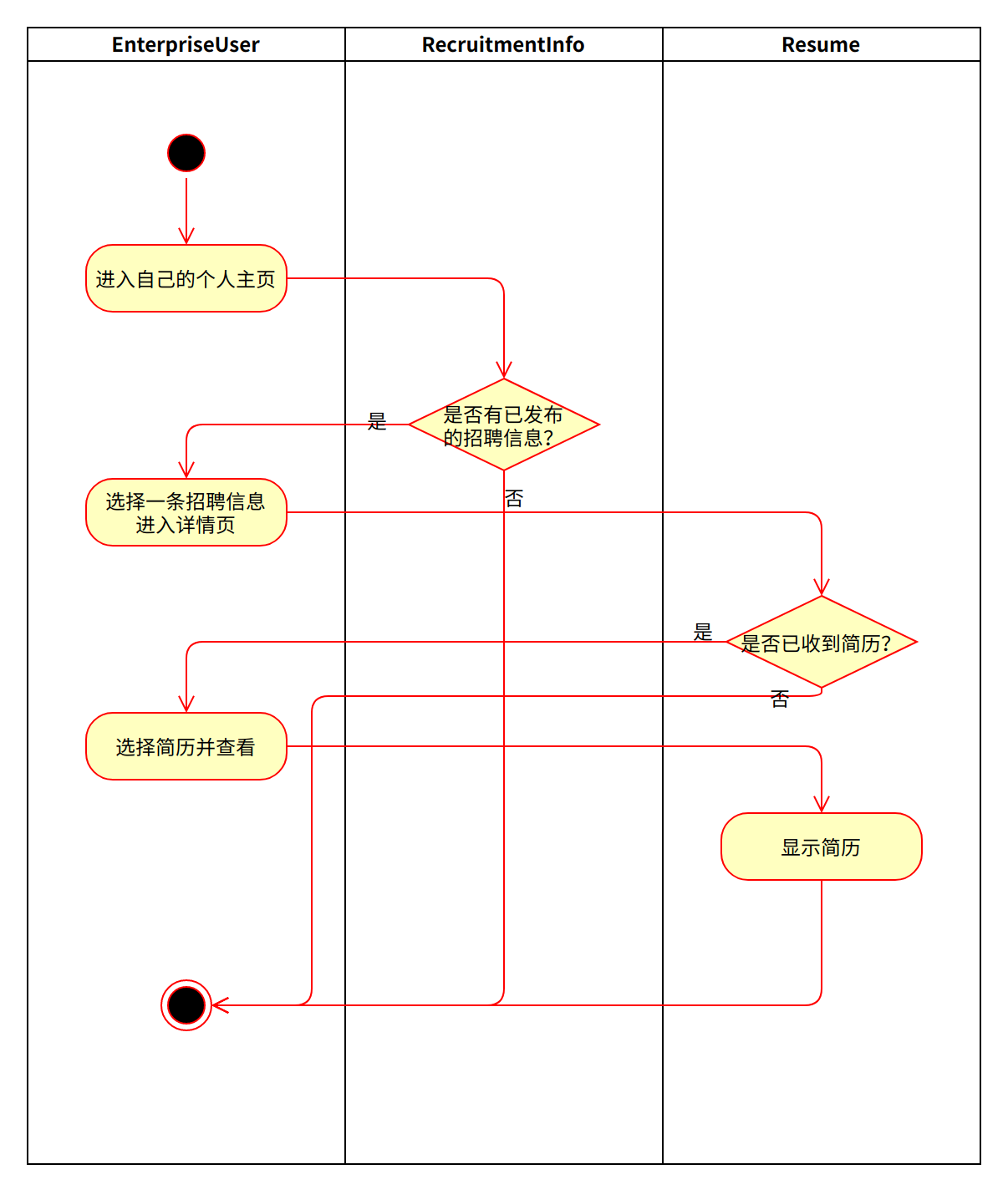
**发布招聘信息**



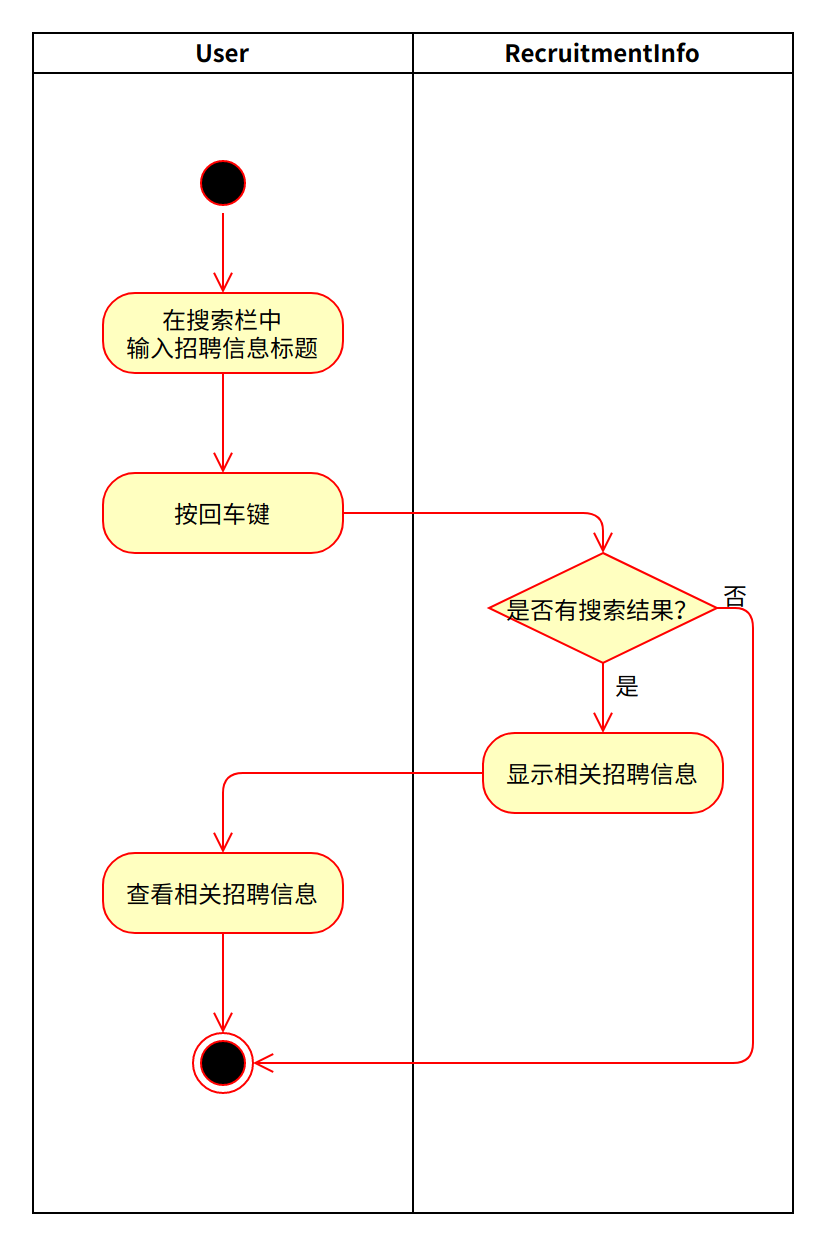
**查看招聘信息**



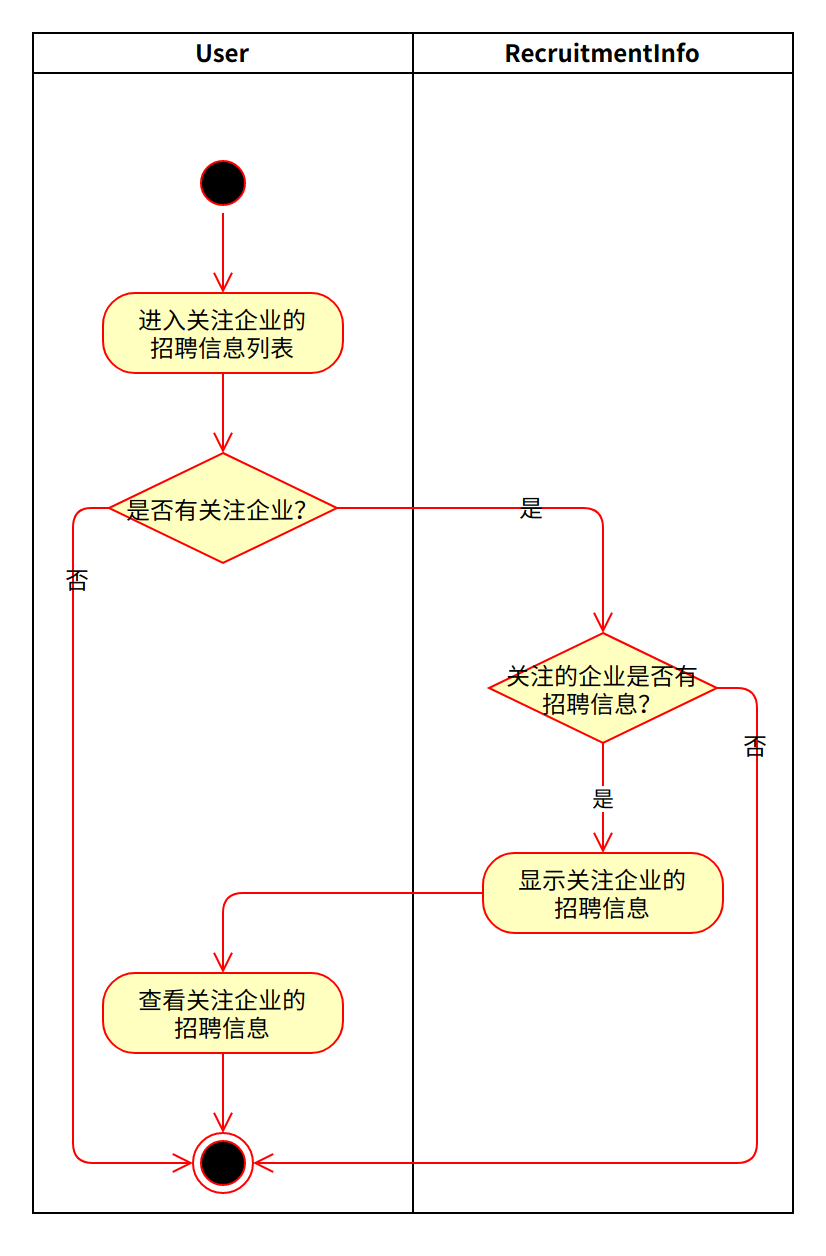
**查看简历**



**搜索招聘信息**



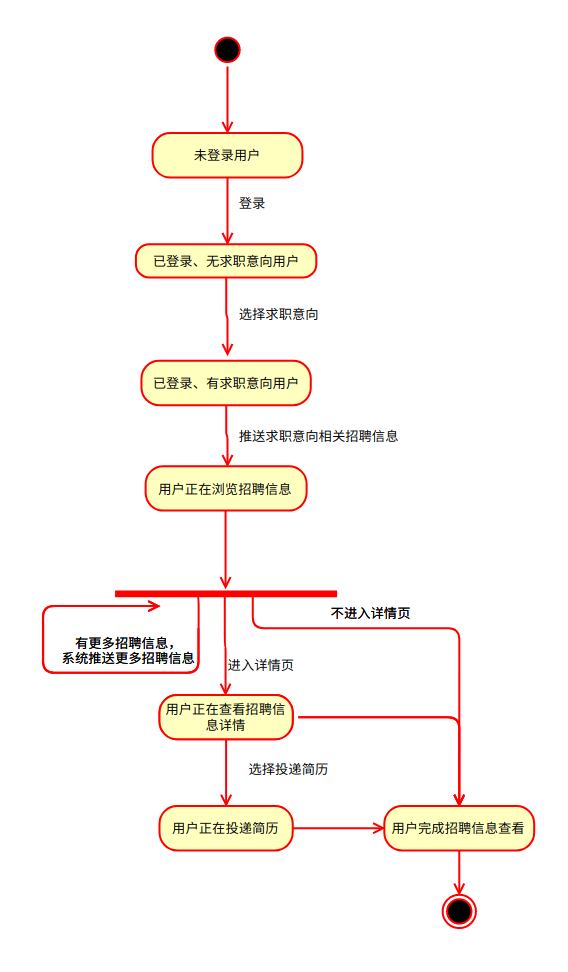
**查看关注企业的招聘信息列表**



**行为建模**

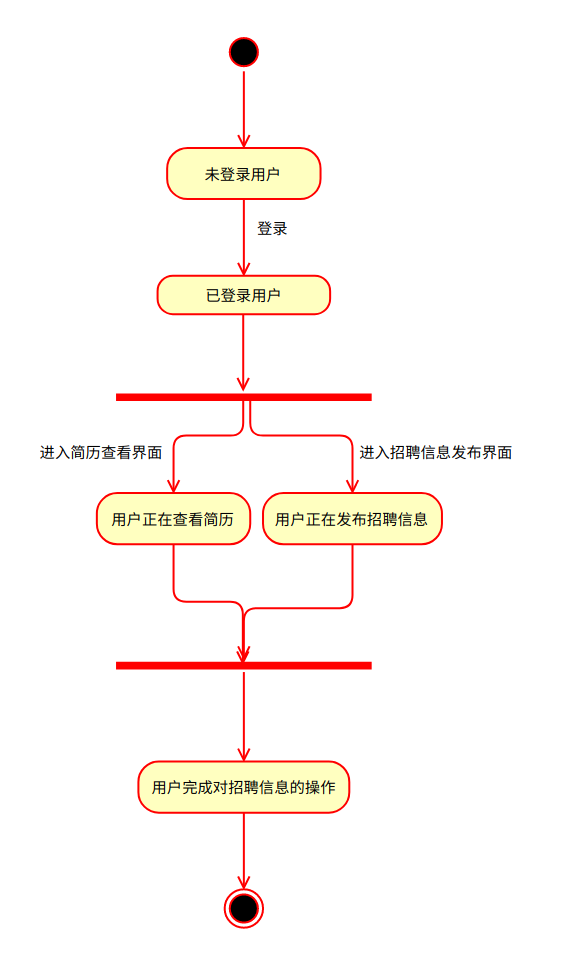
**普通用户类（NormalUser）**

用户登录之后，如果没有选择过求职意向，则无法接收到招聘信息推送。选择求职意向之后，系统会根据求职意向推送相关的招聘信息。用户将推送的招聘信息浏览完毕之后，如果有更多与其求职意向相符的招聘信息，则系统会继续推送。用户可以选择进入招聘信息详情页，进入之后还可以选择投递简历。



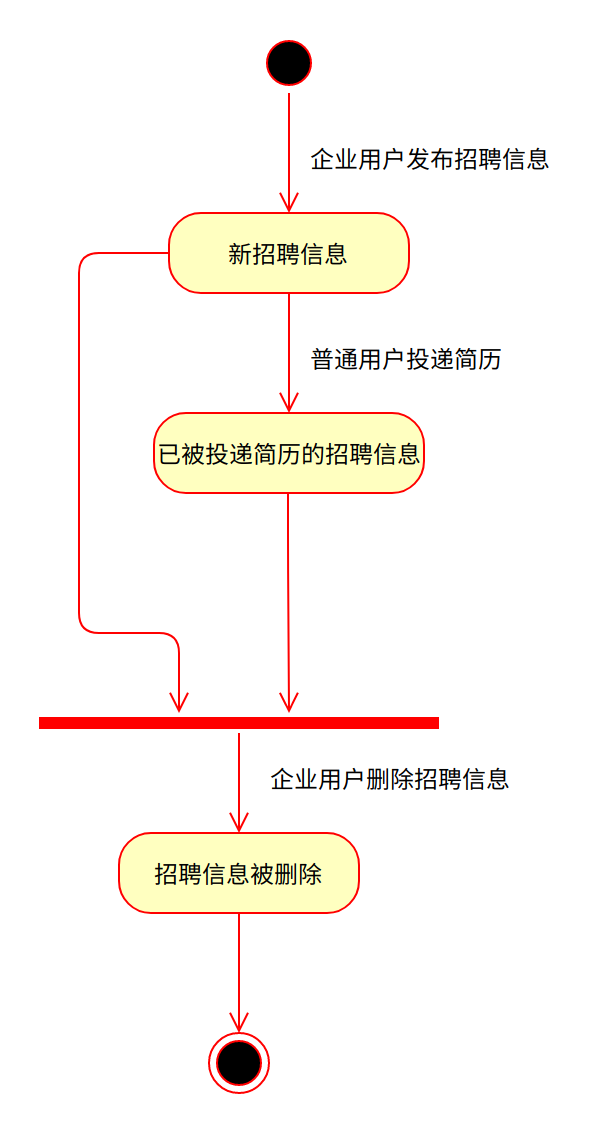
**企业用户类（EnterpriseUser）**

企业用户登录后，可以进入招聘信息发布界面进行招聘信息的发布，也可以进入简历查看收到的简历。



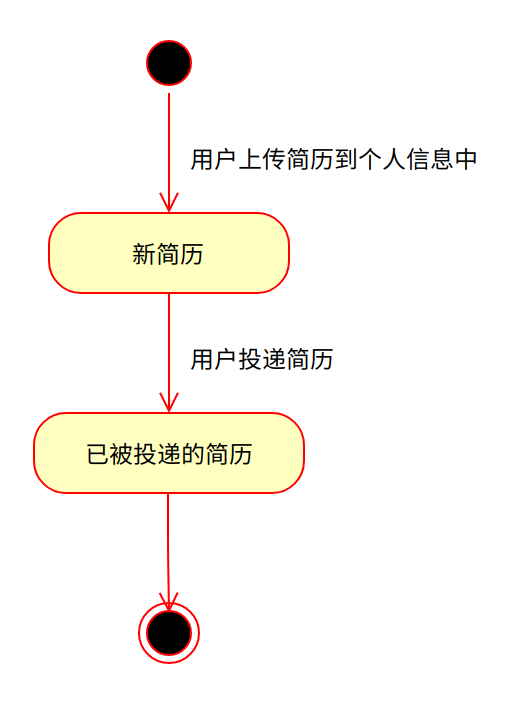
**招聘信息类（RecruitmentInfo）**

企业用户发布招聘信息后，普通用户可以对招聘信息投递简历；招聘信息发布后企业用户也可以删除已发布的招聘信息。



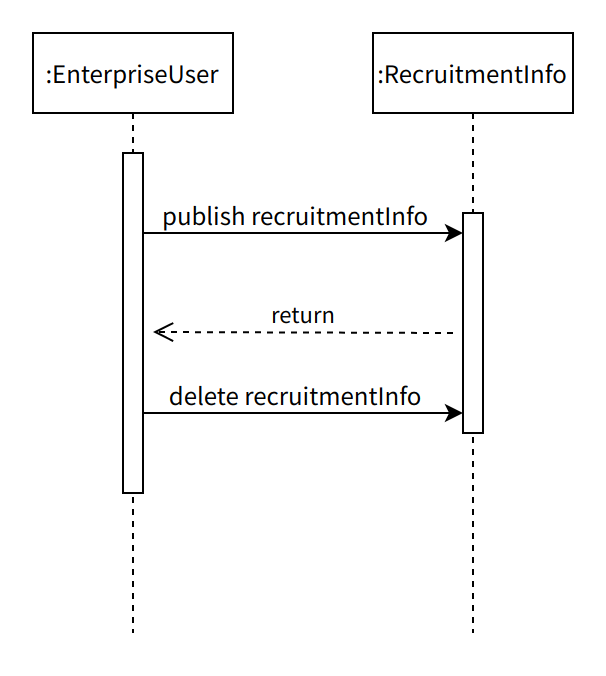
**简历类（Resume）**

用户上传简历后，简历会保存在用户的个人信息页面中。用户可以对已发布的招聘信息投递自己的简历。



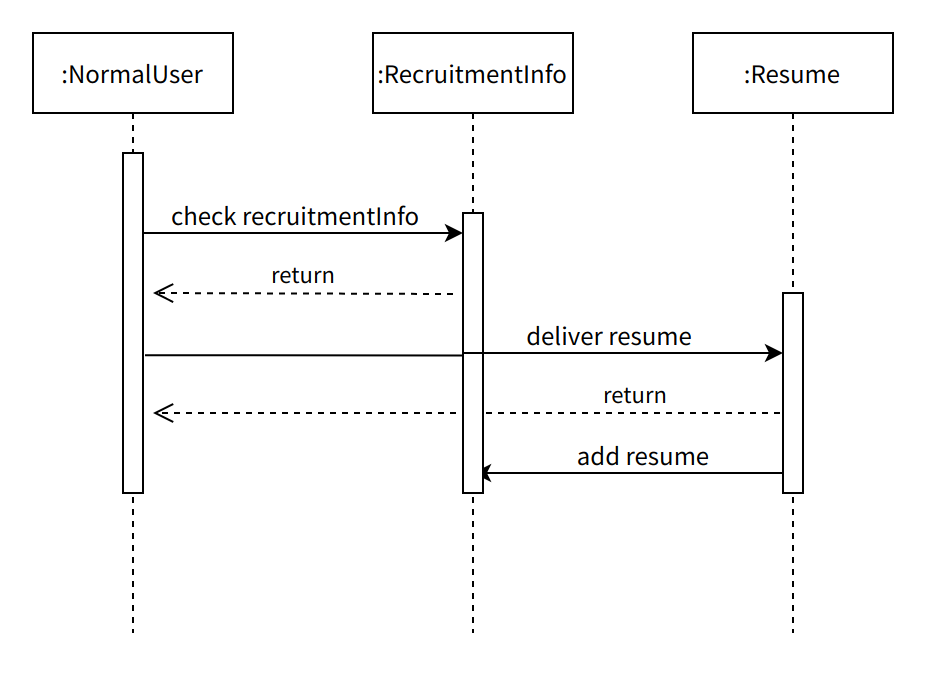
**发布招聘信息**

企业用户发布招聘信息之后，系统返回发布是否成功的信息。招聘信息被发布之后，企业可以将其删除。



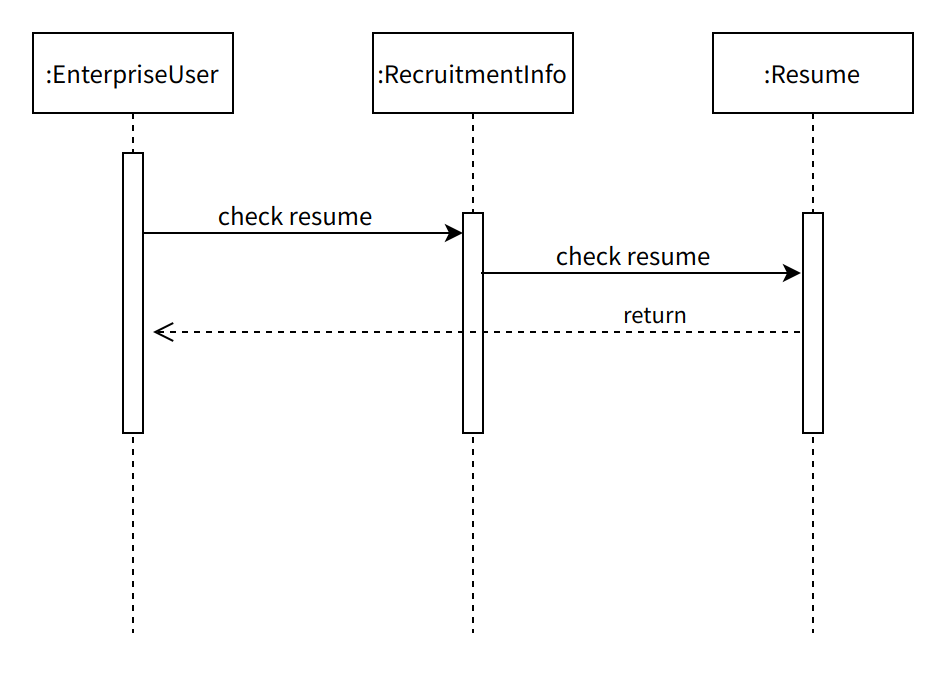
**查看招聘信息**

普通用户查看招聘信息时，系统会将招聘信息返回给用户查看。用户还可以选择向招聘信息投递简历，系统返回是否投递成功的信息。投递成功之后，简历会被添加到招聘信息的简历列表中。



**查看简历**

企业用户选择查看自己发布的某一条招聘信息的简历，系统会将对应的简历返回给用户查看。



**三、系统设计**

**3.1 数据库设计**

**chat\_record**

职责：存储用户间的聊天信息

内容：聊天内容、消息时间、聊天消息id、用户id1、用户id2

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| unified\_id1 | int | 外码 |
| unified\_id2 | int | 外码 |
| chat\_id | int | 主码 |
| contents | varchar |  |
| record\_time | datetime |  |

**comment**

职责：存储用户动态下的动态评论信息

内容：动态id、楼层数、用户id、评论内容、评论时间

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| tweet\_id | int | 外码、主码 |
| floor | int | 主码 |
| unified\_id | int | 外码 |
| record\_time | datetime |  |
| contents | varchar |  |

**edu\_experience**

职责：存储用户的教育经历

内容：用户id、教育经历id、学校名称、学位、主修专业

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| unified\_id | int | 主码 |
| num\_id | int | 主码 |
| college\_name | varchar |  |
| degree | varchar |  |
| major | varchar |  |
| start\_time | datetime |  |
| end\_time | datetime |  |

**enterprise\_info**

职责：存储企业信息

内容：用户id、联系方式、公司描述

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| unified\_id | int | 主码 |
| contact\_way | varchar |  |
| description | varchar |  |

**favorite**

职责：存储用户收藏的职位信息

内容：用户id、企业id、企业职位id

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| user\_id | int | 主码 |
| job\_name | varchar | 主码 |
| enterprise\_id | int | 主码 |

**job\_experience**

职责：存储用户的职业经历

内容：用户id、职业经历id、经历描述、公司名称、职业类型、开始时间、结束时间

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| unified\_id | int | 主码，外码 |
| num\_id | int | 主码 |
| description | varchar |  |
| position\_type | varchar |  |
| enterprise\_name | varchar |  |
| start\_time | datetime |  |
| end\_time | datetime |  |

**likes**

职责：存储用户的动态点赞记录

内容：用户id、动态id

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| unified\_id | int | 主码，外码 |
| tweet\_id | int | 主码，外码 |

**picture**

职责：存储动态的图片存放地址

内容：图片url、动态id、图片顺序id

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| picture\_url | varchar |  |
| tweet\_id | int | 主码，外码 |
| num\_id | int | 主码 |

**position**

职责：存储招聘职位信息

内容：图片url、职位状态、职位描述、薪水、工作地点、联系方式

职位类型、职位名称、发布者id、职位id

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| picture\_url | varchar |  |
| state | int |  |
| description | varchar |  |
| reward | varchar |  |
| work\_place | varchar |  |
| contact\_way | varchar |  |
| limitation | varchar |  |
| position\_type | varchar |  |
| job\_name | varchar |  |
| unified\_id | int | 外码 |
| job\_id | int | 主码 |

**resume**

职责：存储用户简历

内容：用户id、简历id、文件url、简历名称

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| resume\_name | varchar |  |
| docement\_url | varchar |  |
| resume\_id | int | 主码 |
| unified\_id | int | 主码，外码 |

**subscription**

职责：存储用户间的关注关系

内容：用户id、被关注用户id

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| unified\_id | int | 主码，外码 |
| subscription\_id | int | 主码，外码 |

**tweet**

职责：存储用户发布的动态详细信息

内容：动态id、用户id、发布时间、点赞数量、动态状态、动态内容、评论数量

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| comment\_num | int |  |
| contents | varchar |  |
| state | int |  |
| praise\_num | int |  |
| record\_time | datetime |  |
| unified\_id | int | 外码 |
| tweet\_id | int | 主码 |

**user**

职责：存储用户的详细信息

内容：用户id、用户名、密码、邮箱、用户类型、头像url、简要介绍、真实姓名

关注数、背景

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| unified\_id | int | 主码 |
| user\_name | varchar |  |
| password | varchar |  |
| email | varchar |  |
| user\_type | varchar |  |
| picture\_url | varchar |  |
| brief\_info | varchar |  |
| true\_name | varchar |  |
| subscribe\_num | int |  |
| background | varchar |  |

**user\_info**

职责：存储个人用户的特有信息

内容：用户id、id卡，职位偏好、年龄、居住地址、电话号码、性别

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| unified\_id | int |  |
| id\_card | varchar |  |
| pre\_position | varchar |  |
| age | int |  |
| live\_place | varchar |  |
| phone\_num | varchar |  |
| gender | varchar |  |

**application**

职责：存储用户已经发送的职位申请

内容：职位id、简历id、用户id、企业id、简历名称

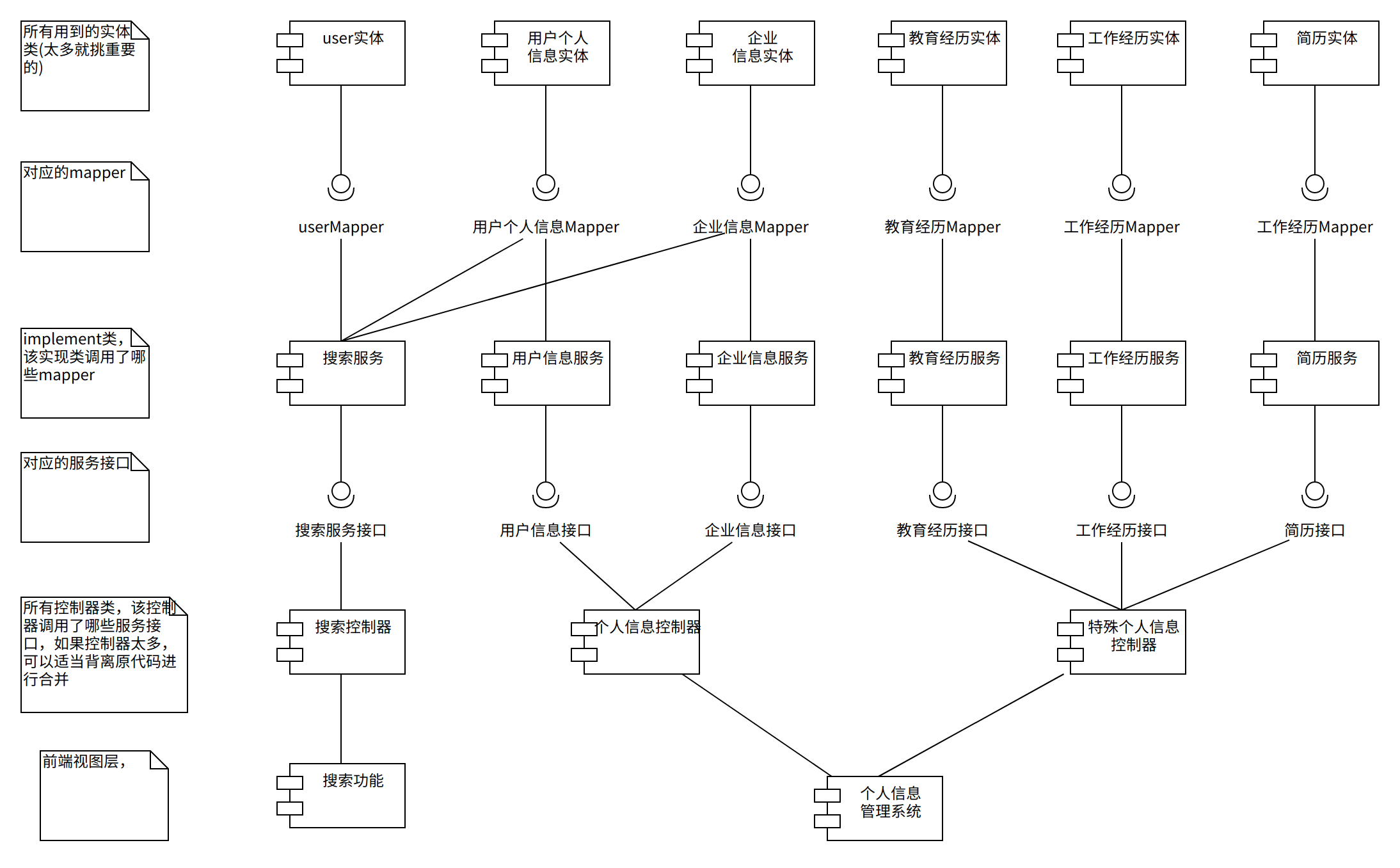
|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| job\_id | int | 主码，外码 |
| resume\_id | int | 主码，外码 |
| user\_id | int | 主码，外码 |
| enterprise\_id | int | 主码，外码 |
| resume\_name | varchar | 外码 |

**3.2 子系统设计**

**3.2.1 用户子系统**

**子系统描述**

该子系统负责管理与平台用户相关的内容，包括基本的登录注册，个人信息的获取修改等，一些比较特殊的个人信息的获取和修改，例如教育经历，工作经历，建立文件、头像等被单独提取出来，作为控制器。用户搜索各种用户的功能也被归并到该子系统中。



**接口**

|  |  |
| --- | --- |
| 接口名称：register |  |
| 描述 | 注册 |
| URL | POST /user/register |
| 参数 | userName：String  password：String  email：String  userType:String |
| 返回值 | 注册结果 |

|  |  |
| --- | --- |
| 接口名称：login |  |
| 描述 | 登录 |
| URL | POST /user/login |
| 参数 | userName：String  password：String |
| 返回值 | userType：String  unifiedId：String  pictureUrl：String  briefInfo：String  trueName：String |

|  |  |
| --- | --- |
| 接口名称：email |  |
| 描述 | 发送邮箱验证码 |
| URL | POST /user/email |
| 参数 | mail:String |
| 返回值 | 随机激活码 |

|  |  |
| --- | --- |
| 接口名称：userInfo |  |
| 描述 | 维护用户个人信息 |
| URL | POST /user/userInfo |
| 参数 | unifiedId：int  trueName：String  age：int  gender：String  phoneNum：String  livePlace：String  prePosition：String  briefInfo：String |
| 返回值 | 是否修改成功 |

|  |  |
| --- | --- |
| 接口名称：userInfo |  |
| 描述 | 获取用户个人信息 |
| URL | GET /user/userInfo |
| 参数 | uid：String  sid：String |
| 返回值 | unifiedId:int  isSubscribed：int  age：int  gender：String  phoneNum：String  prePosition：String[]  trueName:String  pictureUrl:String  briefInfo：String  email:String |

|  |  |
| --- | --- |
| 接口名称：userInfo |  |
| 描述 | 维护企业信息 |
| URL | POST /user/enterpriseInfo |
| 参数 | unifiedId：int  trueName：String  contactWay：String  description：String  briefInfo：String |
| 返回值 | 是否修改成功 |

|  |  |
| --- | --- |
| 接口名称：enterpriseInfo |  |
| 描述 | 获取企业信息 |
| URL | GET /user/enterpriseinfo |
| 参数 | uid：int  sid：int |
| 返回值 | unifiedId:int  isSubscribed：int  trueName:String  contactWay：String  description：String  pictureUrl:String  briefInfo：String  email:String |

|  |  |
| --- | --- |
| 接口名称：pic |  |
| 描述 | 上传头像 |
| URL | POST /user/pic |
| 参数 | unifiedId：int  file：File |
| 返回值 | 是否上传成功 |

|  |  |
| --- | --- |
| 接口名称：search |  |
| 描述 | 搜索企业或用户 |
| URL | GET /user/search |
| 参数 | str：String |
| 返回值 | unifiedId:int  isSubscribed：int  trueName:String  pictureUrl:String  briefInfo：String  email:String |

|  |  |
| --- | --- |
| 接口名称：edu |  |
| 描述 | 增加教育经历 |
| URL | POST /user/edu |
| 参数 | unifiedId：int  startTime：String  endTime：String  collegeName：String  degree：String  major：String |
| 返回值 | 添加是否成功 |

|  |  |
| --- | --- |
| 接口名称：edu |  |
| 描述 | 删除教育经历 |
| URL | DELETE /user/edu |
| 参数 | unifiedId：int  numId：int |
| 返回值 | 删除是否成功 |

|  |  |
| --- | --- |
| 接口名称：edu |  |
| 描述 | 工作经历 |
| URL | POST /user/job |
| 参数 | unifiedId：int  startTime：String  endTime：String  enterpriseName：String  positionType：String  description：String |
| 返回值 | 添加是否成功 |

|  |  |
| --- | --- |
| 接口名称：job |  |
| 描述 | 删除工作经历 |
| URL | DELETE /user/job |
| 参数 | unifiedId：int  numId：int |
| 返回值 | 删除是否成功 |

|  |  |
| --- | --- |
| 接口名称：resume |  |
| 描述 | 获取简历文件 |
| URL | GET /user/resume |
| 参数 | unifiedId：int |
| 返回值 | numId：int  resumeName：String  documentUrl：String |

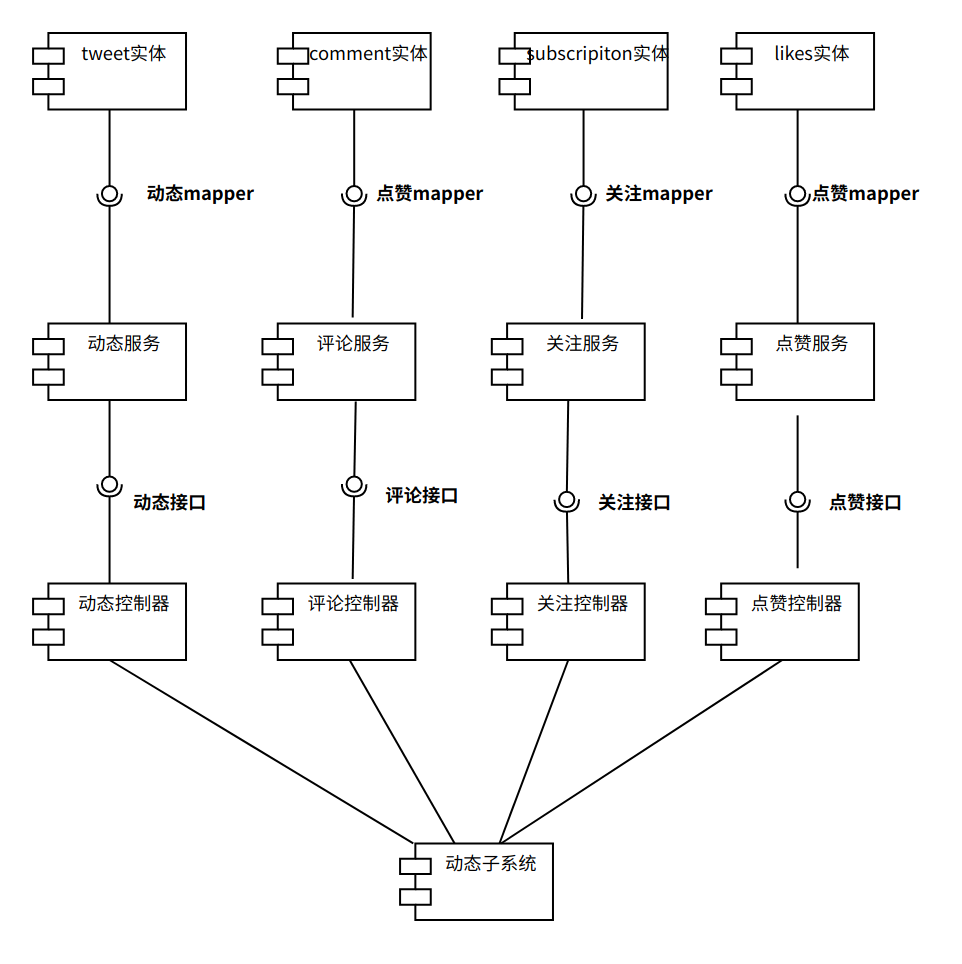
|  |  |
| --- | --- |
| 接口名称：resume |  |
| 描述 | 删除简历文件 |
| URL | DELETE /user/resume |
| 参数 | unifiedId：int  resumeId：int |
| 返回值 | 删除是否成功 |

**3.2.2 动态子系统**

**子系统描述**

动态子系统主要负责平台的动态相关服务，包括用户发布动态、评论动态、动态点赞、用户关注等，

动态子系统主要部署在8003端口。



**接口**

|  |  |
| --- | --- |
| 接口名称：putSubscription |  |
| 描述 | 关注用户 |
| URL | PUT /tweet/subscription |
| 参数 | unifiedId：int subscriptionId：int |
| 返回值 | 返回关注结果 |

|  |  |
| --- | --- |
| 接口名称：deleteSubscription |  |
| 描述 | 取消关注用户 |
| URL | DELETE /tweet/subscription |
| 参数 | unifiedId：int  subscriptionId：int |
| 返回值 | 返回取消关注结果 |

|  |  |
| --- | --- |
| 接口名称：getSubscription |  |
| 描述 | 查看是否关注用户 |
| URL | GET /tweet/subscription |
| 参数 | unifiedId：int  subscriptionId：int |
| 返回值 | 返回是否关注结果 |

|  |  |
| --- | --- |
| 接口名称：getSubscriptionList |  |
| 描述 | 获取用户关注关系列表 |
| URL | GET /tweet/subscriptionList |
| 参数 | unified：int |
| 返回值 | 返回关注关系列表 |

|  |  |
| --- | --- |
| 接口名称：getRecommend |  |
| 描述 | 获取推荐关注列表 |
| URL | GET /tweet/recommend |
| 参数 | unifiedId：int |
| 返回值 | 返回推荐用户列表 |

|  |  |
| --- | --- |
| 接口名称：putLikes |  |
| 描述 | 点赞动态 |
| URL | PUT /tweet/likes |
| 参数 | unifiedId：int  tweetId：int |
| 返回值 | 返回点赞结果 |

|  |  |
| --- | --- |
| 接口名称：deleteLikes |  |
| 描述 | 取消动态关注 |
| URL | DELETE /tweet/likes |
| 参数 | unifiedId：int  tweetId：int |
| 返回值 | 返回取消点赞结果 |

|  |  |
| --- | --- |
| 接口名称：getSelfTweet |  |
| 描述 | 查看个人动态 |
| URL | GET /tweet/self |
| 参数 | visitorId：int  intervieweeId：int  momentId：int |
| 返回值 | 返回个人动态列表 |

|  |  |
| --- | --- |
| 接口名称：getTweetList |  |
| 描述 | 获取关注者动态 |
| URL | GET /tweet/tweetList |
| 参数 | userId：int  momentId：int |
| 返回值 | 返回关注者的动态列表 |

|  |  |
| --- | --- |
| 接口名称：getComment |  |
| 描述 | 查看动态所有评论 |
| URL | GET /tweet/comment |
| 参数 | tweetId：int |
| 返回值 | 返回动态的所有评论列表 |

|  |  |
| --- | --- |
| 接口名称：postComment |  |
| 描述 | 为动态添加评论 |
| URL | POST /tweet/comment |
| 参数 | unifiedId：int  tweetId：int  content：int |
| 返回值 | 返回评论结果 |

|  |  |
| --- | --- |
| 接口名称：deleteComment |  |
| 描述 | 删除评论信息 |
| URL | DELETE /tweet/comment |
| 参数 | unifiedId：int  tweetId：int  floor：int |
| 返回值 | 返回删除评论结果 |

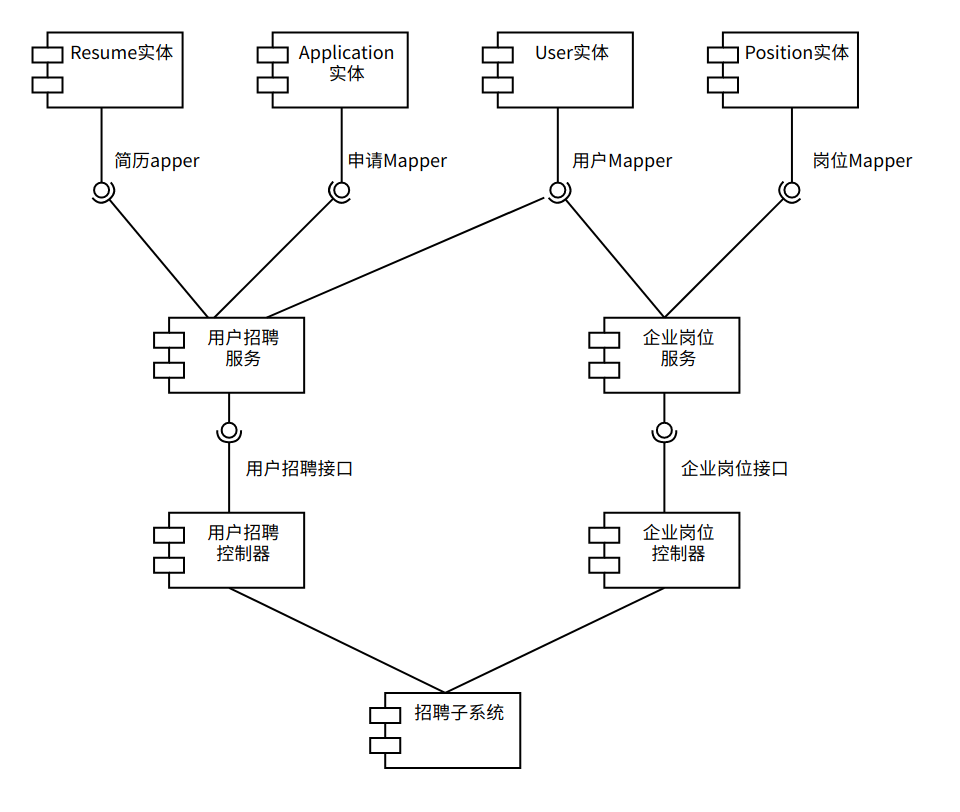
|  |  |
| --- | --- |
| 接口名称：PUTtweet |  |
| 描述 | 发布动态 |
| URL | PUT /tweet |
| 参数 | unifiedId：int  content：string  files：MutipartFiles |
| 返回值 | 返回新生成的动态Id |

|  |  |
| --- | --- |
| 接口名称：deleteTweet |  |
| 描述 | 删除动态 |
| URL | DELETE /tweet |
| 参数 | tweetId：int |
| 返回值 | 返回删除动态结果 |

**3.2.3 招聘子系统**

**子系统描述**

招聘子系统主要包括了招聘流程相关的服务，应聘方可以查看岗位信息，获取推荐岗位，申请心仪岗位；企业可以创建岗位，编辑岗位信息，查看申请者的简历等。招聘子系统部署在8005端口



**接口**

|  |  |
| --- | --- |
| 接口名称：createPosition |  |
| 描述 | 创建岗位 |
| URL | POST /recruit/enterprise/position |
| 参数 | unifiedId int  description String  reward String  workPlace String  contactWay String  limitation String  positionType String  jobName String |
| 返回值 | 返回是否创建成功 |

|  |  |
| --- | --- |
| 接口名称：deletePosition |  |
| 描述 | 企业删除岗位信息 |
| URL | DELETE /recruit/enterprise/position |
| 参数 | unifiedId int  jobId |
| 返回值 | 返回是否删除成功 |

|  |  |
| --- | --- |
| 接口名称：updatePosition |  |
| 描述 | 企业更新岗位信息 |
| URL | PUT /recruit/enterprise/position |
| 参数 | unifiedId int  jobId int  description String  reward String  workPlace String  contactWay String  limitation String  positionType String  jobName String |
| 返回值 | 返回是否修改成功 |

|  |  |
| --- | --- |
| 接口名称：getAllApplicants |  |
| 描述 | 企业查看某一岗位所有申请者信息及其简历 |
| URL | GET /recruit/applicants |
| 参数 | unifiedId int  jobId int |
| 返回值 | 返回申请用户列表 |

|  |  |
| --- | --- |
| 接口名称：getAllPositions |  |
| 描述 | 用户获取某家企业的所有岗位信息 |
| URL | GET recruit/position/all |
| 参数 | unifiedId int |
| 返回值 | 返回岗位列表 |

|  |  |
| --- | --- |
| 接口名称：getRecommendPositions |  |
| 描述 | 用户获取岗位推荐信息 |
| URL | GET recruit/position/recommend |
| 参数 | unifiedId int |
| 返回值 | 返回推荐岗位列表 |

|  |  |
| --- | --- |
| 接口名称：getPosition |  |
| 描述 | 用户获取特定岗位信息 |
| URL | GET /recruit/position/specified |
| 参数 | unifiedId int  jobId int |
| 返回值 | 返回岗位详情 |

|  |  |
| --- | --- |
| 接口名称：sendApplication |  |
| 描述 | 用户发送简历/申请岗位 |
| URL | POST recruit/application |
| 参数 | unifiedId int  enterpriseId int  jobId int  resumeId int |
| 返回值 | 返回申请发送是否成功 |

|  |  |
| --- | --- |
| 接口名称：cancleApplication |  |
| 描述 | 用户撤回发送简历/申请岗位 |
| URL | DELETE recruit/application |
| 参数 | unifiedId int  enterpriseId int  jobId int |
| 返回值 | 返回撤回是否成功 |

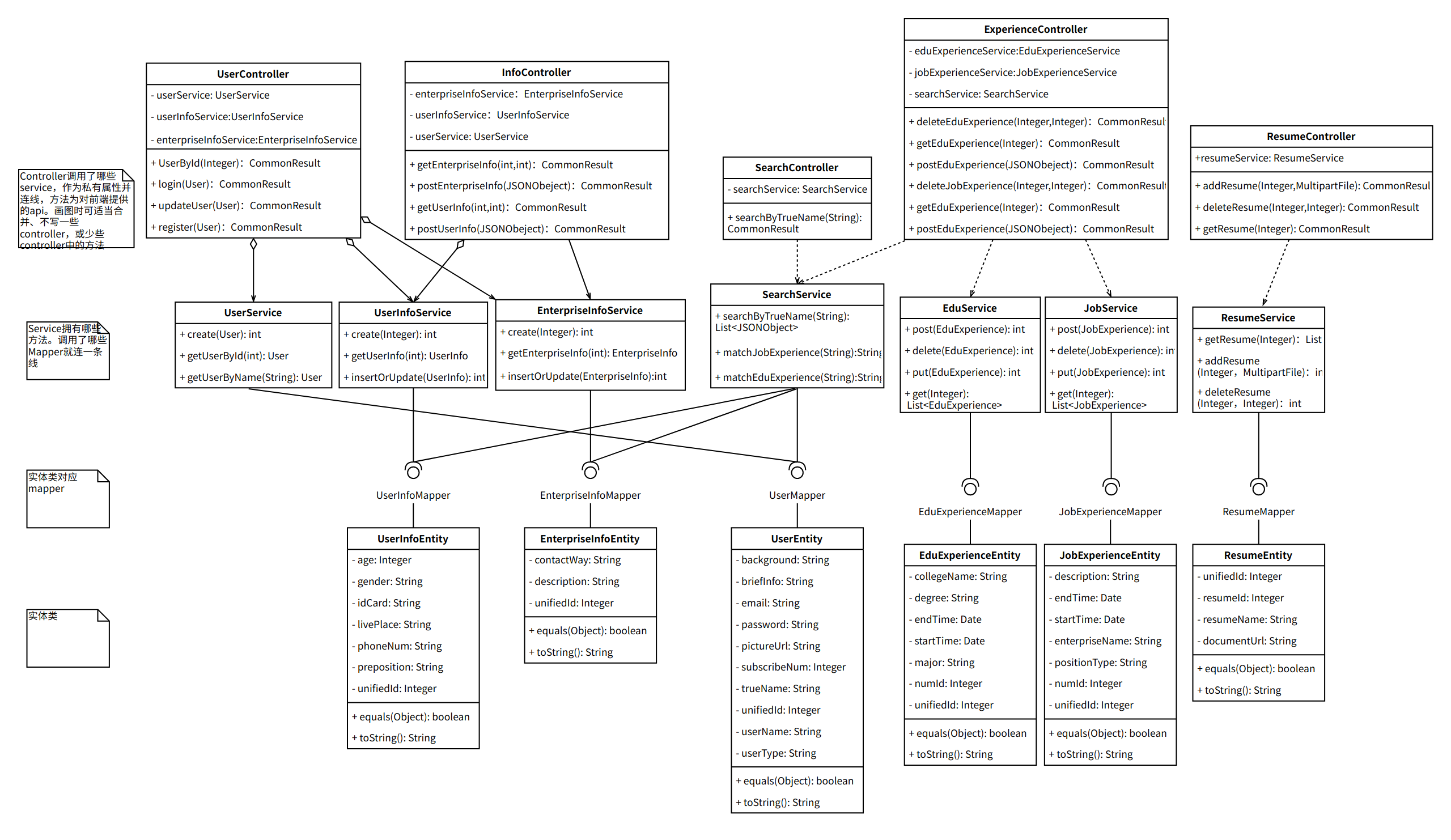
|  |  |
| --- | --- |
| 接口名称：getAppliedPositions |  |
| 描述 | 用户查看所有投递过的岗位 |
| URL | GET recruit/application |
| 参数 | userId |
| 返回值 | 返回岗位列表 |

**3.3 类设计**

**3.3.1用户子系统**

**功能**

用户子系统中底层需要和存储在数据库中的各种信息实体类打交道，通过Mapper接口，服务层来间接与数据库通信，一些服务仅需要与一个实体类通信，搜索服务需要与多个数据库实体交互。最顶层的controller层调用服务层提供的服务，类似功能的服务交由一个controller进行转发即可



**输入输出项**

|  |  |  |  |
| --- | --- | --- | --- |
| class：EduExperienceService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| postEduExperience | EduExperience | int | 添加教育经历 |
| deleteEduExperience | Integer unifiedId  Integer numId | int | 删除教育经历 |
| putEduExperience | EduExperience | int | 修改教育经历 |
| getEduExperience | Integer unifiedId | List<EduExperience> | 获取教育经历 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：EnterpriseInfoService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| create | Integer unifiedId | int | 创建企业信息 |
| getEnterpriseInfo | Int unifiedId | EnterpriseInfo | 获取企业信息 |
| insertOrUpdateEnterpriseInfo | EnterpriseInfo | int | 插入或更新企业用户信息 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：JobExperienceService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| postJobExperience | JobExperience | int | 添加工作经历 |
| deleteJobExperience | Integer unifiedId  Integer numId | int | 删除工作经历 |
| putJobExperience | JobExperience | int | 修改工作经历 |
| getJobExperience | Integer unifiedId | List<JobExperience> | 获取工作经历 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：ResumeService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| getResume | Integer unifiedId | List<Resume> | 获取简历 |
| addResume | Integer unifiedId  MultipartFile | int | 添加新简历 |
| deleteResume | Integer unifiedId  Integer ResumeId | int | 删除简历 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：HeadPictureService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| addHeadPicture | Integer unifiedId  MultipartFile file | int | 上传头像 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：SearchService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| searchByTrueName | String str | List<JSONObeject> | 按照用户真实姓名搜索 |
| matchJobExperience | String str | String | 匹配工作经历 |
| matchEduExperience | String str | String | 匹配教育经历 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：UserInfoService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| create | Integer unfiedId | int | 创建用户个人信息 |
| getUserinfo | Int unifiedId | Userinfo | 获取用户个人信息 |
| inserOrUpdateUserInfo | UserInfo | int | 更新或插入用户个人信息 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：UserService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| create | User | int | 创建用户 |
| getUserById | Int unifiedId | User | 根据统一Id获取用户 |
| getUserByName | String username | User | 根据用户名获取用户 |

**属性详细描述**

|  |  |
| --- | --- |
| class：EduexperienceServiceImplement |  |
| 简介：教育经历服务实现类 |  |
| 属性 | 描述 |
| eduExperienceMapper | 教育经历数据访问对象 |

|  |  |
| --- | --- |
| class：EnterpriseInfoServiceImplement |  |
| 简介：企业信息服务实现类 |  |
| 属性 | 描述 |
| enterpriseInfoMapper | 企业信息数据访问对象 |

|  |  |
| --- | --- |
| class：HeadPictureServiceImplement |  |
| 简介：头像服务实现类 |  |
| 属性 | 描述 |
| userMapper | 用户数据访问对象 |
| bucketName | oss存储服务名 |
| endpoint | oss存储服务端口 |

|  |  |
| --- | --- |
| class：JobExperienceServiceImplement |  |
| 简介：工作经历服务实现类 |  |
| 属性 | 描述 |
| jobExperienceMapper | 工作经历数据访问对象 |

|  |  |
| --- | --- |
| class：MailServiceImplement |  |
| 简介：邮件服务实现类 |  |
| 属性 | 描述 |
| JavaMailSender | java邮件发送对象 |

|  |  |
| --- | --- |
| class：ResumeServiceImplement |  |
| 简介：简历服务实现类 |  |
| 属性 | 描述 |
| resumeMapper | 简历数据访问对象 |
| bucketName | oss存储服务名 |
| endpoint | oss存储服务端口 |

|  |  |
| --- | --- |
| class：SearchServiceImplement |  |
| 简介：搜索服务实现类 |  |
| 属性 | 描述 |
| userMapper | 用户数据访问对象 |

|  |  |
| --- | --- |
| class：UserInfoServiceImplement |  |
| 简介：用户个人信息服务实现类 |  |
| 属性 | 描述 |
| userInfoMapper | 用户个人信息数据访问对象 |

|  |  |
| --- | --- |
| class：UserServiceImplement |  |
| 简介：用户服务实现类 |  |
| 属性 | 描述 |
| userMapper | 用户数据访问对象 |
| bucketName | oss存储服务名 |
| endpoint | oss存储服务端口 |

**方法详细描述及实现流程**

|  |  |  |
| --- | --- | --- |
| class：EduexperienceServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| postEduExperience | 添加教育经历 | 根据前端返回的教育经历信息，向数据库中插入一条新的教育经历 |
| deleteEduExperience | 删除教育经历 | 根据前端返回的统一id，删除数据库中的指定教育经历 |
| putEduExperience | 修改教育经历 | 根据前端返回的统一id和教育经历信息，更新数据库已有教育经历 |
| getEduExperience | 获取教育经历 | 根据前端返回的统一id获取该用户的所有教育经历 |

|  |  |  |
| --- | --- | --- |
| class：EnterpriseInfoServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| create | 创建企业信息 | 向数据库的EnterpriseInfo表中插入新加入的企业信息，返回统一id |
| getEnterpriseInfo | 获取企业信息 | 根据前端传入的统一id，去数据库中查找该id对应的企业信息，并返回该企业信息 |
| insertOrUpdateEnterpriseInfo | 插入或更新企业用户信息 | 根据前端传入的id，去数据库中查找该id对应的企业信息是否存在，找到后用前端传入的信息更新该企业信息 |

|  |  |  |
| --- | --- | --- |
| class：HeadPictureServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| addHeadPicture | 上传头像 | 将用户上传的建立文件通过java类MuiltPartFile存储到指定位置，并向数据库中插入对应位置的url |

|  |  |  |
| --- | --- | --- |
| class：JobExperienceServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| postJobExperience | 添加工作经历 | 根据前端返回的工作经历信息，向数据库中插入一条新的工作经历 |
| deleteJobExperience | 删除工作经历 | 根据前端返回的统一id，删除数据库中的指定工作经历 |
| putJobExperience | 修改工作经历 | 根据前端返回的统一id和工作经历信息，更新数据库已有工作经历 |
| getJobExperience | 获取工作经历 | 根据前端返回的统一id获取该用户的所有工作经历 |

|  |  |  |
| --- | --- | --- |
| class：MailServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| sendMail | 发送随机激活码邮件 | 调用邮件发送相关服务，向前端传入的邮箱地址发送一封带有随机验证码的邮件。 |

|  |  |  |
| --- | --- | --- |
| class：ResumeServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| getResume | 获取简历 | 根据前端传入的统一id，返回该用户上传的所有简历的存储的url |
| addResume | 添加新简历 | 将用户上传的建立文件通过java类MuiltPartFile存储到指定位置，并向数据库中插入对应位置的url |
| deleteResume | 删除简历 | 根据前端传入的统一id和简历id删除对应用户的指定建立文件，同时删除数据库中的记录 |

|  |  |  |
| --- | --- | --- |
| class：SearchServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| searchByTrueName | 按照用户真实姓名搜索 | 根据传入的搜索字段，利用余弦匹配算法，返回用户可能搜索的真实姓名 |
| matchJobExperience | 匹配工作经历 | 根据传入的搜索字段，利用余弦匹配算法，返回用户可能搜索的工作公司 |
| matchEduExperience | 匹配教育经历 | 根据传入的搜索字段，利用余弦匹配算法，返回用户可能搜索的高校名称 |

|  |  |  |
| --- | --- | --- |
| class：UserInfoServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| create | 创建用户个人信息 | 向数据库的UserInfo表中插入新加入的用户信息，返回统一id |
| getUserinfo | 获取用户个人信息 | 根据前端传入的统一id，去数据库中查找该id对应的用户个人信息，并返回该用户个人信息 |
| inserOrUpdateUserInfo | 更新或插入用户个人信息 | 根据前端传入的id，去数据库中查找该id对应的用户个人信息是否存在，找到后用前端传入的信息更新该用户个人信息 |

|  |  |  |
| --- | --- | --- |
| class：UserServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| create | 创建用户 | 向数据库的User表中插入新加入的用户信息，设定用户类型，返回统一id |
| getUserById | 根据统一Id获取用户 | 根据前端传入的统一id，去数据库中查找该id对应的用户，并返回该用户信息 |
| getUserByName | 根据用户名获取用户 | 根据前端传入的用户名，去数据库中查找该用户名对应的用户，并返回该用户信息 |

**限制条件及出错处理**

|  |  |
| --- | --- |
| class：EduExperienceController |  |
| 方法名 | 描述 |
| postEduExperience | 转换时间格式字符串时，使用异常捕捉，避免不符合格式的时间无法转化导致报错。判断传入参数是否包含统一认证id，否则报错 |
| deleteEduExperience | 判断传入的统一认证id和顺序id是否存在，否则返回失败 |
| putEduExperience | 转换时间格式字符串时，使用异常捕捉，避免不符合格式的时间无法转化导致报错。判断传入参数是否包含统一认证id，否则报错 |
| getEduExperience | 判断传入的统一认证i是否存在，否则返回失败 |

|  |  |
| --- | --- |
| class：EnterpriseInfoController |  |
| 方法名 | 描述 |
| getEnterpriseInfo | 判断查看者与被查看者的id是否相同 |
| updateEnterpriseInfo | 更新前需判断用户是否存在，并且传入的id不能为空 |

|  |  |
| --- | --- |
| class：HeadPictureController |  |
| 方法名 | 描述 |
| addHeadPicture | 检测是否存在上传头像的用户id，上传文件需捕获异常，防止文件上传失败 |

|  |  |
| --- | --- |
| class：JobExperienceController |  |
| 方法名 | 描述 |
| postJobExperience | 转换时间格式字符串时，使用异常捕捉，避免不符合格式的时间无法转化导致报错。判断传入参数是否包含统一认证id，否则报错 |
| deleteJobExperience | 判断传入的统一认证id和顺序id是否存在，否则返回失败 |
| putJobExperience | 转换时间格式字符串时，使用异常捕捉，避免不符合格式的时间无法转化导致报错。判断传入参数是否包含统一认证id，否则报错 |
| getJobExperience | 判断传入的统一认证i是否存在，否则返回失败 |

|  |  |
| --- | --- |
| class：ResumeController |  |
| 方法名 | 描述 |
| getResume | 判断传入参数是否包含统一认证id，否则报错 |
| addResume | 判断传入参数是否包含统一认证id。上传文件需捕获异常，防止文件上传失败 |
| deleteResume | 判断是否存在对应的统一id和简历id对应的数据库数据，没有则报删除失败 |

|  |  |
| --- | --- |
| class：SearchController |  |
| 方法名 | 描述 |
| searchTrueName | 判断关键字调用搜索服务后，搜索结果是否为空 |

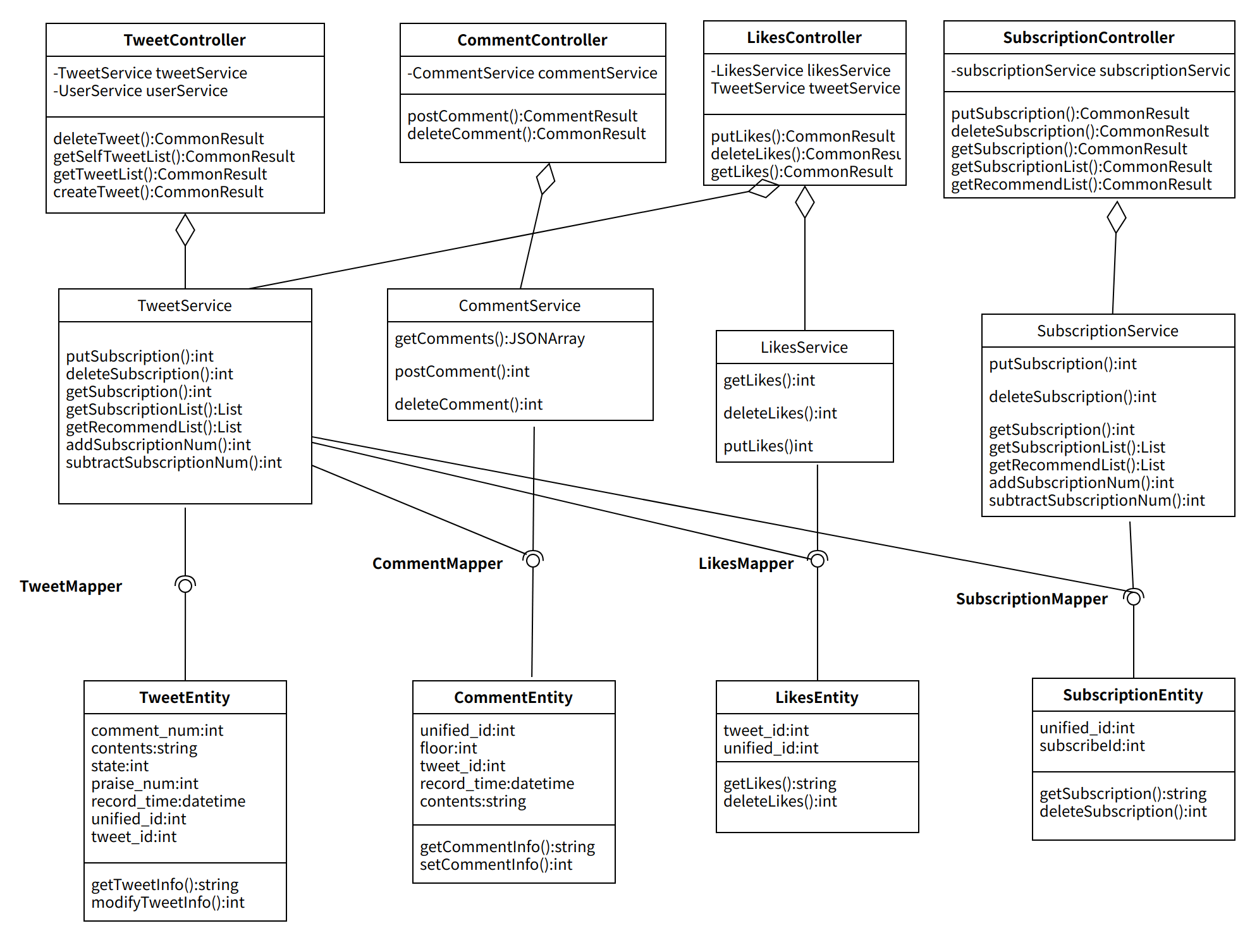
|  |  |
| --- | --- |
| class：UserController |  |
| 方法名 | 描述 |
| create | 判断用户名是否有重复，判断前端传入的用户类型是否合法 |
| getMailCaptcha | 判断同一邮箱的请求次数，多次调用触发服务熔断 |
| login | 判断用户的密码是否正确后才执行cookie、token生成操作 |
| UserById | 判断查询的返回结果是否有内容 |
| updateUser | 判断传入参数是否包含统一认证id，否则报错 |

|  |  |
| --- | --- |
| class：UserInfoController |  |
| 方法名 | 描述 |
| getUserInfo | 判断查看者与被查看者的id是否相同 |
| updateUserInfo | 更新前需判断用户是否存在，并且传入的id不能为空 |

**3.3.2 动态子系统**

**功能**

动态子系统通过各个位于顶层的Controller来处理外界请求，Controller层主要面向前端的访问请求。Controller层主要通过调用Service的各个方法来处理请求，Service层的功能是将Controller细化过的功能进行实现，通过调用不同的Mapper接口来间接操作数据库，从而实现数据的增删改查操作。



**输入输出项**

|  |  |  |  |
| --- | --- | --- | --- |
| class：CommentService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| postComment | Int tweetId | int | 输入参数为动态的id  返回值为请求结果 |
| deleteComment | Int unifiedId  Int tweetId  Int floor | int | 输入参数为动态id、用户id和楼层数  返回值为删除评论结果 |
| getComments | Int tweetId | JSONArray | 输入值为动态id  返回值为评论列表 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：LikesService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| getLikes | Int unifiedId  Int tweetId | int | 输入参数为用户id和动态id  返回值为点赞查询结果 |
| deleteLikes | Int unifiedId  Int tweetId | int | 输入参数为用户id和动态id  返回值为取消点赞结果 |
| putLikes | Int unifiedId  Int tweetId | int | 输入参数为用户id和动态id  返回值为点赞结果 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：SubscriptionService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| putSubscription | Int unifiedId  Int subscriptionId | Int | 输入参数为用户id和被关注用户id  返回值为关注结果 |
| deleteSubscription | Int unifiedId  Int subscriptionId | Int | 输入参数为用户id和被关注用户id  返回值为取消关注结果 |
| getSubscription | Int unifiedId  Int subscriptionId | int | 输入参数为用户id和被关注用户id  返回值为关注结果 |
| getSubscriptionList | Int unifiedId | List | 输入参数为用户id  返回值为关注列表 |
| getRecommendList | Int unifiedId | List | 输入参数为用户id  返回值为推荐用户列表 |
| addSubscriptionNum | Int unifiedId | int | 输入参数为用户id  返回值为请求结果 |
| subtractSubscriptionNum | Int unifiedId | int | 输入参数为用户id  返回值为请求结果 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：TweetService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| addLikesNum | Int tweetId | int | 输入参数为动态id  返回值为点赞结果 |
| subtractLikesNum | Int tweetId | int | 输入参数为动态id  返回值为取消点赞结果 |
| deleteTweet | Int tweetId | int | 输入参数为动态id  返回值为删除动态结果 |
| getSeltTweetList | Int visitorId  Int intervieweeId  Int momentId | JSONArray | 输入参数为访问者id、被访问者id、当前浏览最后动态id  返回值为个人动态列表 |
| getTweetList | Int visitorId  Int momentId | JSONArray | 输入参数为访问者id和当前浏览最后动态id  返回值为关注者的动态列表 |
| getSimpleUserInfo | Int unifiedId | JSONArray | 输入参数为用户id  返回值为用户的简要信息 |
| getTweetPicture | Int tweetId | List | 输入参数为动态id  返回值为动态的图片列表 |
| createTweet | Int unifiedId  String content  Date recordTime  List files | int | 输入参数为用户id、动态内容、发布时间、动态图片列表  返回值为动态的id |
| uploadPic | Int unifiedId  List files | int | 输入参数为动态id、动态图片  返回值为存储结果 |
| maxTweetId | none | int | 无输入参数  返回值为最大的动态id |

**属性详细描述**

|  |  |
| --- | --- |
| class：CommentServiceImpl |  |
| 简介：评论服务接口实现类 |  |
| 属性 | 描述 |
| CommentMapper  commentMapper | 评论DAO层 |
| TweetMapper  tweetMapper | 动态DAO层 |
| TweetService  tweetService | 动态相关服务类 |

|  |  |
| --- | --- |
| class：LikesServiceImpl |  |
| 简介：点赞服务接口实现类 |  |
| 属性 | 描述 |
| LikesMapper  likesMapper | 点赞DAO层 |
| TweetMapper  tweetMapper | 动态DAO层 |

|  |  |
| --- | --- |
| class：SubscriptionServiceImpl |  |
| 简介：关注服务接口实现类 |  |
| 属性 | 描述 |
| SubscriptionMapper  subscriptionMapper | 关注DAO层 |
| UserMapper  userMapper | 用户DAO层 |

|  |  |
| --- | --- |
| class：TweetServiceImpl |  |
| 简介：动态服务接口实现类 |  |
| 属性 | 描述 |
| TweetMapper  tweetMapper | 动态DAO层 |
| LikesMapper  likesMapper | 点赞DAO层 |
| UserMapper  userMapper | 用户DAO层 |
| PictureMapper  pictureMapper | 动态图片DAO层 |
| SubscriptionMapper  subscriptionMapper | 关注DAO层 |

**方法详细描述及实现流程**

|  |  |  |
| --- | --- | --- |
| class：CommentServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| postComment | 发布动态评论 | 根据前端返回的评论信息，在数据库中插入一条新的评论信息 |
| deleteComment | 删除动态评论 | 根据相应动态id、用户id和楼层数，在数据库中找到相应记录进行删除 |
| getComments | 获取动态评论 | 根据相应动态id，在数据库中查询相应id的动态的所有评论 |

|  |  |  |
| --- | --- | --- |
| class：LikesServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| getLikes | 查询点赞状态 | 根据相应用户id和动态id，在数据库中查询是否存在点赞关系 |
| deleteLikes | 取消动态点赞 | 根据相应用户id和动态id，从数据库中找到相应记录进行删除 |
| putLikes | 动态点赞 | 通过相应用户id和动态id，在数据库中新建一条点赞记录 |

|  |  |  |
| --- | --- | --- |
| class：SubscriptionServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| putSubscription | 关注用户 | 根据相应用户id和被关注者id，在数据库中插入一条新的关注记录 |
| deleteSubscription | 取消关注用户 | 根据用户id和被关注者id，在数据库中找到相应记录进行删除 |
| getSubscription | 查询是否关注某用户 | 根据用户id和被关注者id，在数据库中查找是否存在相应记录 |
| getSubscriptionList | 查询用户的所有关注者信息 | 根据相应用户id，在数据库中查找该用户的所有关注列表信息 |
| getRecommendList | 向用户推荐其他未关注的用户 | 根据相应用户id，查找该用户尚未关注的用户信息进行返回 |
| addSubscriptionNum | 增加用户被关注数 | 根据相应用户id，在数据库中寻找该用户信息，将其被关注数加一 |
| subtractSubscriptionNum | 减少用户被关注数 | 根据相应用户id，在数据库中寻找该用户信息，将其被关注数减一 |

|  |  |  |
| --- | --- | --- |
| class：TweetServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| addLikesNum | 增加动态点赞数 | 根据相应动态id，在数据库中找到相应动态信息，将其点赞数加一 |
| subtractLikesNum | 减少动态点赞数 | 根据相应动态id，在数据库中找到相应动态信息，将其点赞数减一 |
| deleteTweet | 删除动态 | 根据相应动态id，在数据库中找到该动态信息进行删除操作 |
| getSeltTweetList | 获取单独用户的所有动态 | 根据相应用户id，在数据库中寻找该用户发布的所用动态，根据momentId，返回比该编号小的十条动态 |
| getTweetList | 获取用户关注者的动态 | 根据相应用户id，在数据库中寻找该用户所有关注者发布的所用动态，根据momentId，返回比该编号小的十条动态 |
| getSimpleUserInfo | 获取用户简单个人信息 | 根据相应用户id，在数据库中查询该用户的简单用户信息 |
| getTweetPicture | 获取动态的所有图片 | 根据相应动态id，在数据库中查询该动态的所有动态图片 |
| createTweet | 发布动态 | 根据相应的动态内容，在数据库中新生成一条动态信息，并将生成的动态信息编号进行返回 |
| uploadPic | 上传动态图片至服务器 | 根据相应动态id，在数据库中新生成一条图片存储记录，并将图片存至服务器指定位置 |
| maxTweetId | 查询数据库中动态的最大编号 | 寻找当前数据库中动态信息的最大编号 |

**限制条件及出错处理**

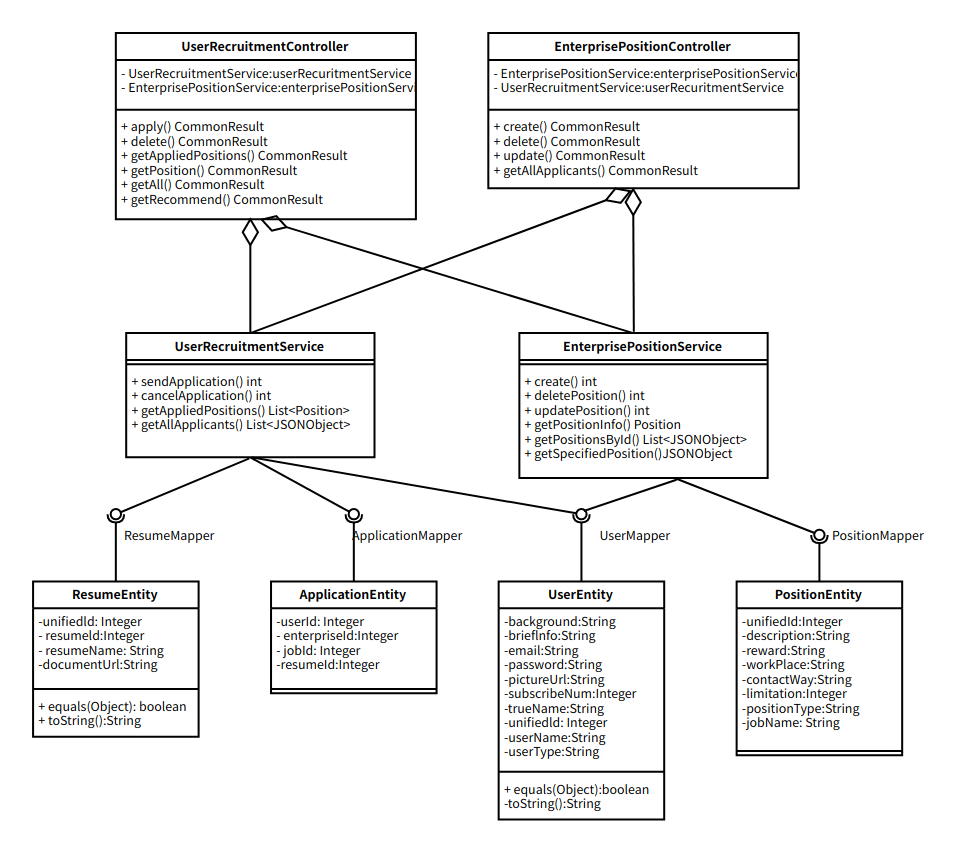
|  |  |
| --- | --- |
| class：TweetServiceImpl |  |
| 方法名 | 描述 |
| uploadPic | 在调用该方法向服务器上传图片时，若上传发生错误，则会抛出一个IOException |

|  |  |
| --- | --- |
| class：CommentServiceImpl |  |
| 方法名 | 描述 |
| postComment | 在调用该方法时，新的评论的楼层数会是原来的最大楼层数加一。若原先最大楼层数为null(没有评论)，此时该评论的楼层数将被设置为1 |

**3.3.3 招聘子系统**

**功能**

招聘子系统的视线主要包括三个层次，最上层是Controller层，这里包含两个Controller，分别负责用户招聘以及企业岗位相关的功能，是接口与实现之间的桥梁。Controller下方是Service，主要负责功能的实现，再下方是与数据库交互的DAO层，这里我们定义为若干个实体，服务如果要与数据库交互，那么需要通过不同的Mapper来实现。



**输入输出项**

|  |  |  |  |
| --- | --- | --- | --- |
| class：EnterprisePositionService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| isEnterprise | Int unifiedId | Boolean | 根据id判断某一个用户是否是企业用户 |
| isCommonUser | Int unifiedId | Boolean | 根据id判断某一个用户是否是普通用户 |
| create | Position position | int | 创建岗位 |
| deletePosition | int unifiedId, int jobId | int | 删除岗位 |
| updatePosition | Position position | int | 更新岗位 |
| getPositionInfo | int unifiedId, int jobId | Position | 获取该企业创建的岗位信息 |
| getSpecifiedPosition | int unifiedId, int jobId | JSONObject | 获取指定岗位信息+用户是否申请该岗位 |
| getPositionsById | int unifiedId,Integer momentId | List<JSONObject> | 获取企业的所有岗位 |
| getRecommendedPositionsById | int unifiedId,Integer momentId | List<JSONObject> | 根据id获取用户的推荐岗位 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：UserRecruitmentService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| sendApplication | Application application | int | 用户发送岗位申请 |
| cancelApplication | Application application | int | 用户取消岗位申请 |
| getAppliedPositions | int userId | List<Position> | 用户获取申请过的所有岗位 |
| getAllApplicants | int unifiedId, int jobId | List<JSONObject> | 企业查看岗位所有申请者信息 |

**属性详细描述**

|  |  |
| --- | --- |
| class：EnterprisePositionServiceImpl |  |
| 简介：企业岗位服务接口实现类 |  |
| 属性 | 描述 |
| PositionMapper | 岗位DAO层 |
| UserMapper | 用户DAO层 |

|  |  |
| --- | --- |
| class：UserRecruitmentServiceImpl |  |
| 简介：招聘服务接口实现类 |  |
| 属性 | 描述 |
| ApplicationMapper | 岗位申请DAO层 |
| UserMapper | 用户DAO层 |
| ResumeMapper | 简历DAO层 |

**方法详细描述及实现流程**

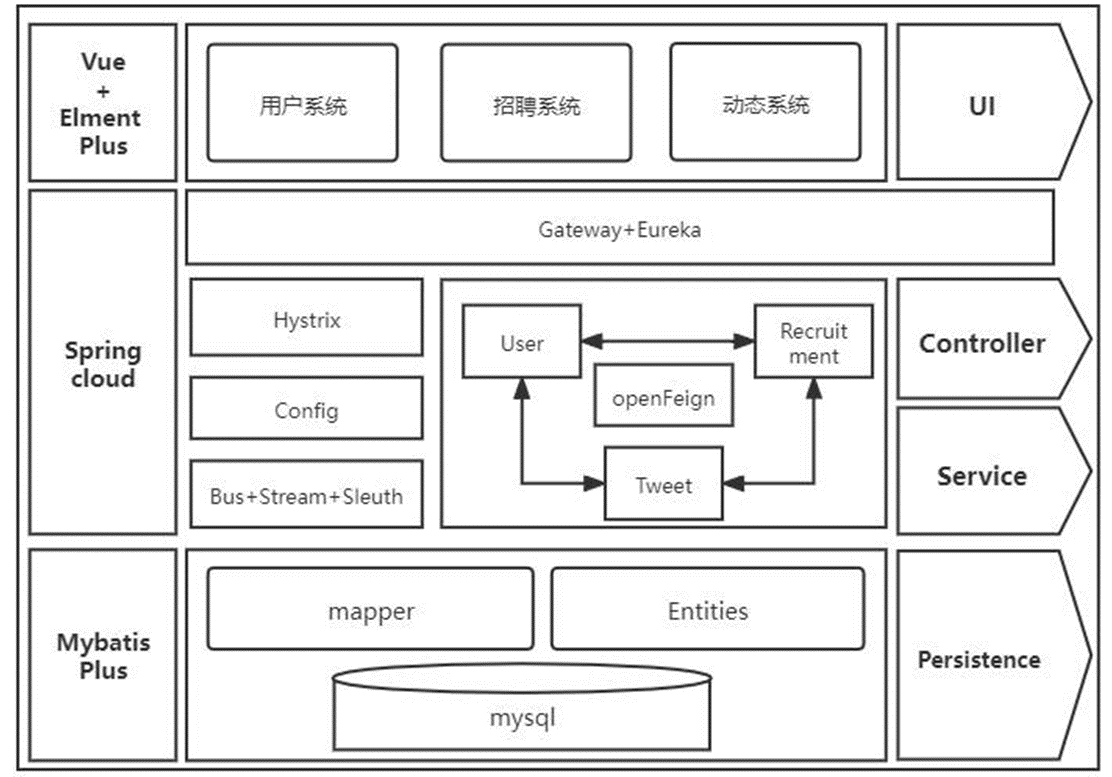
|  |  |  |
| --- | --- | --- |
| class：EnterprisePositionServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| isEnterprise | 判断是否企业用户 | 根据id查找用户，根据用户的类型判断是否是企业用户 |
| isCommonUser | 是否普通用户 | 根据id查找用户，根据用户的类型判断是否是普通用户 |
| create | 创建岗位 | 根据前端传递的岗位信息，在数据库中插入一条新的岗位数据 |
| deletePosition | 删除岗位 | 根据前端传来的企业id和岗位id，采用惰性删除的方法删除岗位。 |
| updatePosition | 更新岗位 | 根据前端传递的岗位信息，在数据库中更新这个岗位的数据 |
| getPositionInfo | 获取该企业创建的岗位信息 | 根据前端传来的企业id和岗位id，在数据库中找到相应岗位并返回 |
| getSpecifiedPosition | 获取指定岗位信息+用户是否申请该岗位 | 根据前端传来的用户id和岗位id，在数据库中找到相应岗位，同时根据用户id和岗位id判断该岗位是否已经申请 |
| getPositionsById | 获取企业的所有岗位 | 根据前端传来的企业id，匹配所有创建者为这个企业的岗位，并返回 |
| getRecommendedPositionsById | 获取推荐岗位 | 根据前端传来的用户id，首先找到该用户，获取用户的职位偏好，然后根据职位偏好查找符合的岗位并返回 |

|  |  |  |
| --- | --- | --- |
| class：UserRecruitmentServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| sendApplication | 发送岗位申请 | 根据前端传来的数据在application表中插入一条新的数据 |
| cancelApplication | 取消岗位申请 | 根据前端传来的数据在application表中移除一条特定的数据 |
| getAppliedPositions | 获取申请过的岗位 | 根据前端传递用户id，匹配application表中申请者为这个id的行，在根据行中的岗位id找到岗位的详细信息并返回 |
| getAllApplicants | 企业获取某一个岗位的所有申请者 | 根据前端传来的企业id和岗位id，去application表中查找匹配的行，随后通过行中的数据分别查找用户的详细信息以及用户简历的信息并返回。 |

**限制条件及出错处理**

|  |  |
| --- | --- |
| class：EnterprisePositionServiceImpl |  |
| 方法名 | 描述 |
| create | 根据前端传递的岗位信息，在数据库中插入一条新的岗位数据，如果插入失败则返回0 |
| getPositionInfo | 根据前端传来的企业id和岗位id，在数据库中找到相应岗位并返回，如果没找到返回null |
| getRecommendedPositionsById | 根据前端传来的用户id，首先找到该用户，获取用户的职位偏好，然后根据职位偏好查找符合的岗位并返回，但如果某一个用户是企业用户，那么他不具有职位偏好的属性，返回空数组。 |

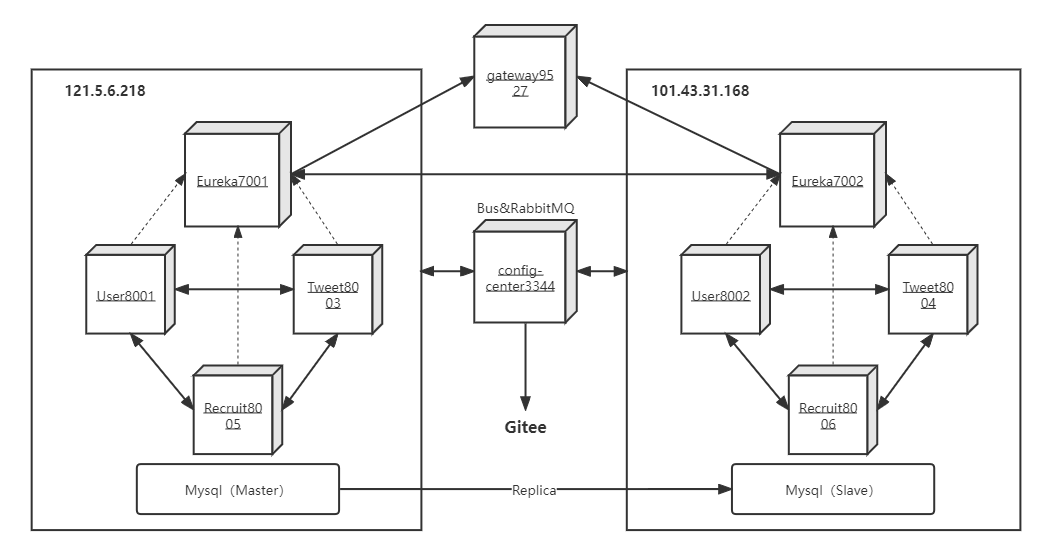
**四、项目架构与技术栈**



**横向架构**

在横向上，我们的项目架构仍分为传统的三层架构：

* 数据库持久层：Mysql数据库，采用Mybatis-plus作为ORM框架
* 服务层：按照业务逻辑分为三大模块：用户模块（User），动态模块（Tweet）和招聘模块（Recruitment）
* 表示层：采用Vue响应式框架，辅以Element组件库



**纵向架构**

在纵向上基于spring cloud搭建微服务架构：

* Eureka负责服务注册与治理
* 分布式部署，在101.43.31.168与121.5.68.218两台服务器上部署所有模块，通过两个Eureka相互注册，构成多节点高可用的服务集群
* OpenFeign实现服务间相互调用（自带负载均衡）
* Hystrix实现服务降级与服务熔断
* Config分布式配置中心，从Gitee仓库中实时爬取配置信息
* Bus消息总线和RabbitMQ消息队列，实现配置信息的动态刷新
* Gateway网关，对前端请求做token鉴权与日志记录，转发到服务模块的同时启用轮询负载均衡

**数据架构**

在两台服务器都启动了Mysql服务，采用**Mysql Replication**主从复制策略，以实现数据库层面的多节点高可用集群

* **主从复制优点**：

1. **读写分离**：

* 指定一台数据库为主数据库，负责业务逻辑的读写；其余为从数据库，负责读操作。
* 所有的写操作仅在主数据库上运行，再异步复制给所有的从数据库，实现数据同步
* 所有的从数据库均可进行读操作，以分担主数据库的线程压力

1. **双机热备**

* 一旦主数据库崩溃，一台从数据库将立刻取代其位置，以供业务顺利进展

1. **架构扩展**

* 在项目业务量增大，I/O访问频率增加时，通过多库存储，降低磁盘I/O访问的频率，提高单机I/O性能。
* **主从复制缺点**：
* 难以解决分布式主键自增的一致性问题
* 异步式的数据同步，在高压线程下容易导致主从数据库的不一致性
* **主从复制原理**：

MySQL主从复制涉及到三个线程，一个运行在主节点（log dump thread），其余两个(I/O thread, SQL thread)运行在从节点：

* 主节点 **binary log dump** 线程

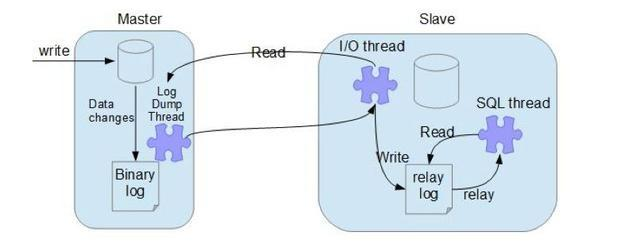
当从节点连接主节点时，主节点会创建一个log dump 线程，用于发送bin-log的内容。在读取bin-log中的操作时，此线程会对主节点上的bin-log加锁，当读取完成，甚至在发动给从节点之前，锁会被释放。

* 从节点 **I/O** 线程

当从节点上执行start slave命令之后，从节点会创建一个I/O线程用来连接主节点，请求主库中更新的bin-log。I/O线程接收到主节点binlog dump 进程发来的更新之后，保存在本地relay-log中。

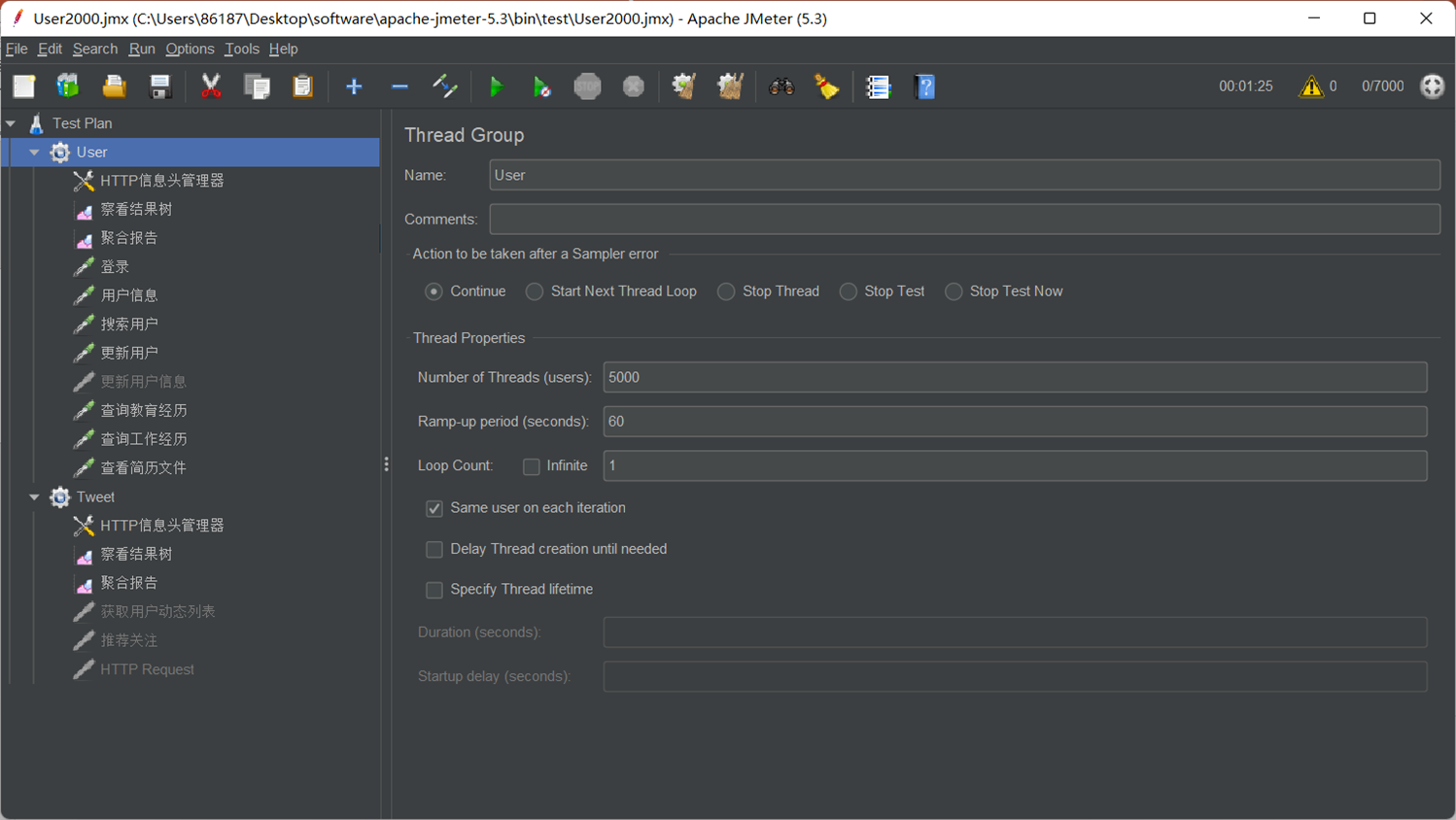
* 从节点 **SQL** 线程

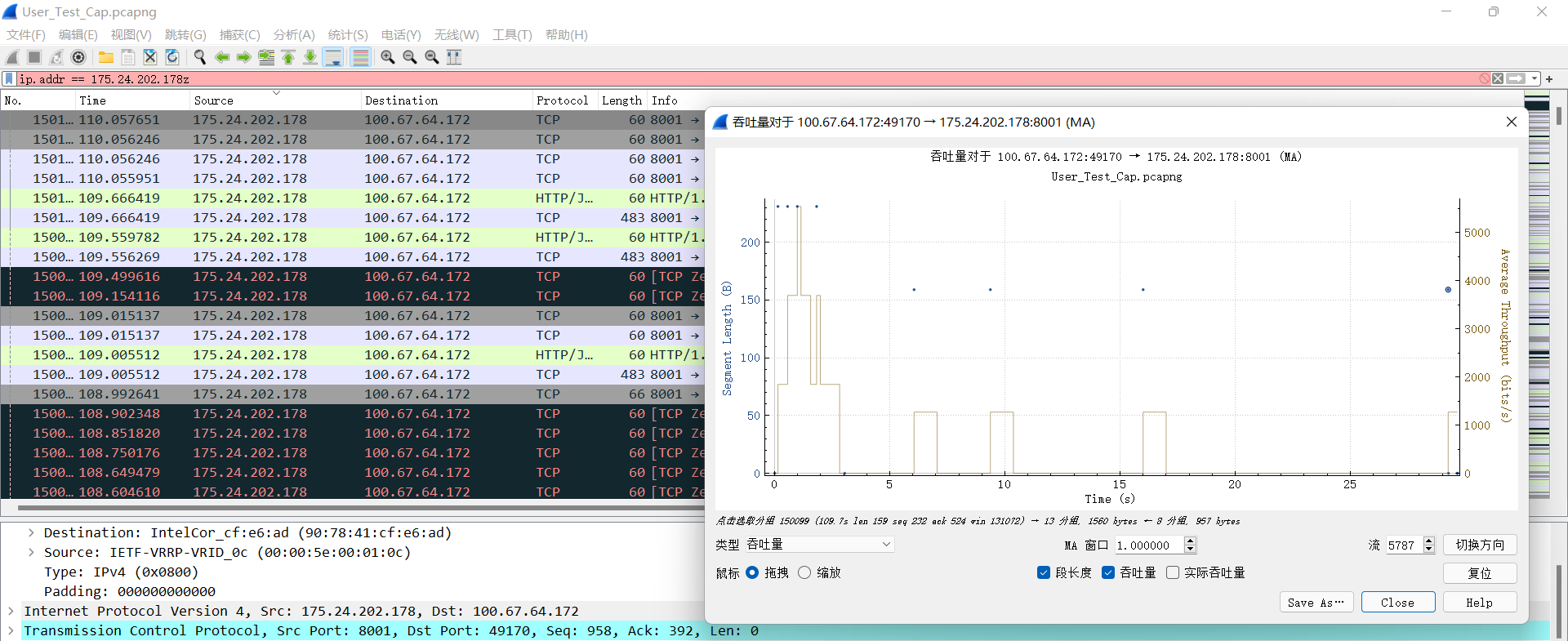
SQL线程负责读取relay log中的内容，解析成具体的操作并执行，最终保证主从数据的一致性。



**五、性能测试**

我们使用Apache Jmeter软件对我们的三个服务模块进行性能测试，使用Wireshark对跨域的Http请求进行网络抓包。分别在2000、5000到10000的线程并行度下，在60s内同时调用测试某一个模块下的所有服务接口。

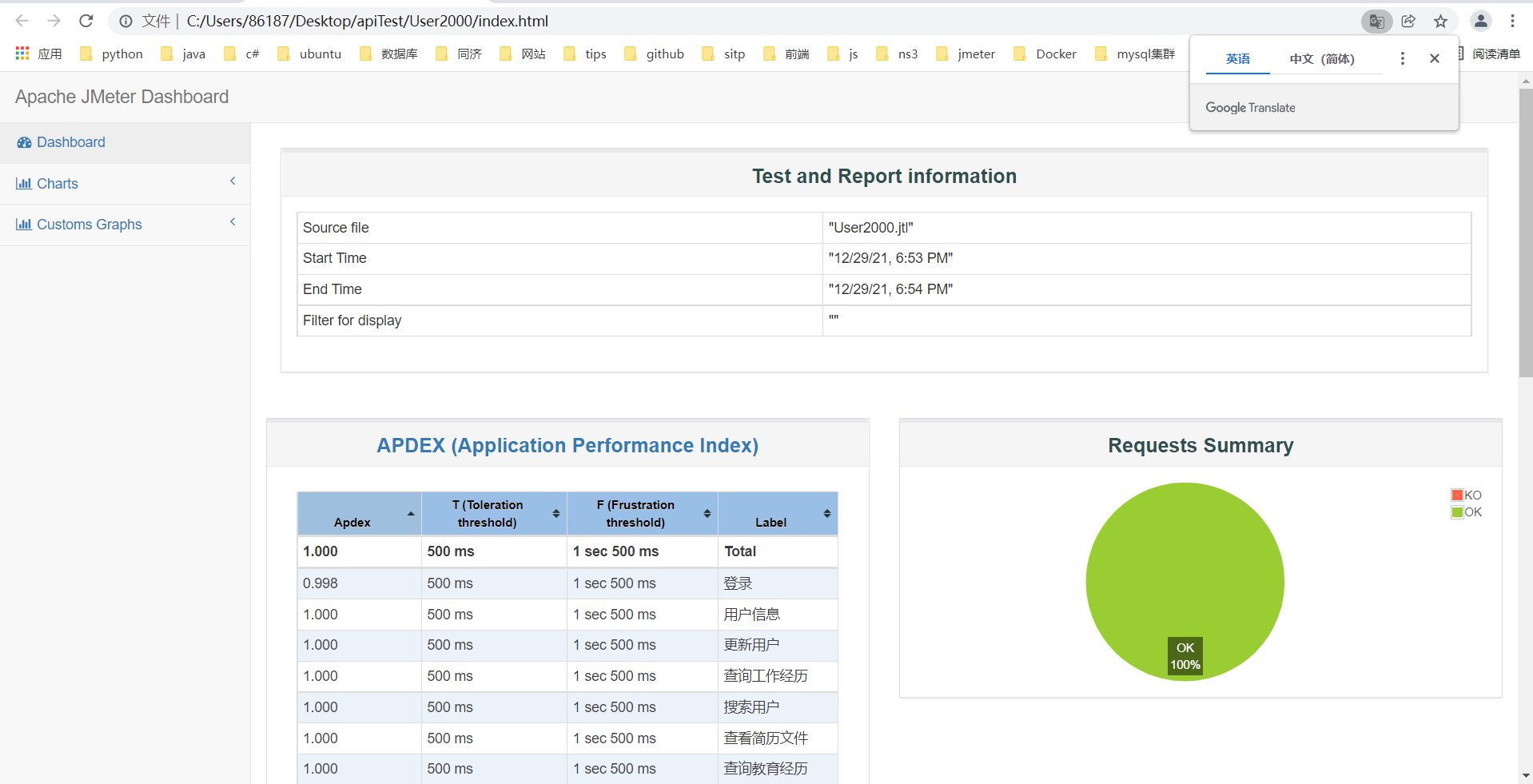




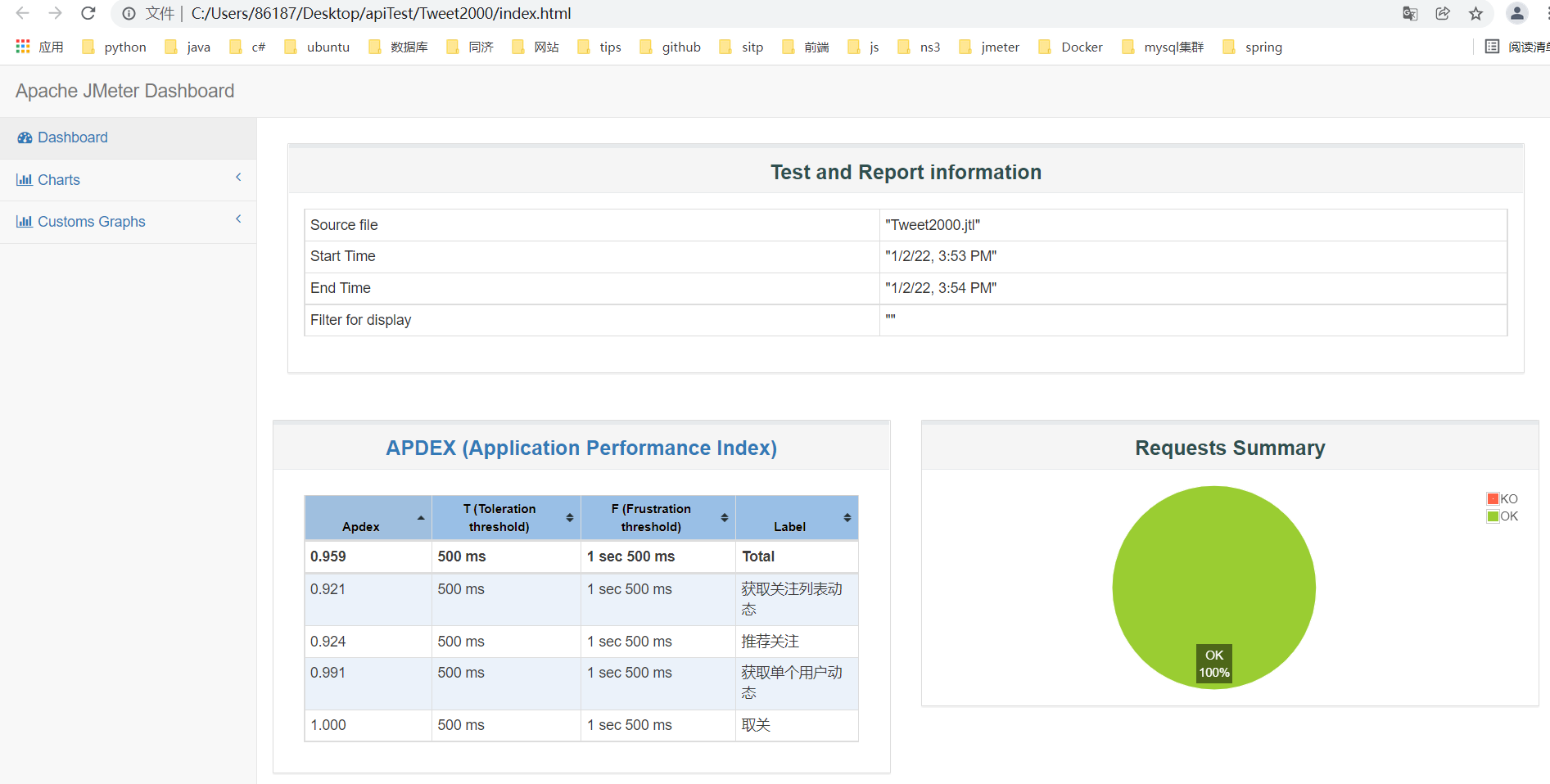
在线程并行度为2000的情况下，我们分别对三个模块的所有接口进行测试，可视化结果如下：

* Apdex: Application Performance Index 应用性能指标——越接近1表示用户满意度越高
* Toleration threshold——响应时间容忍阈值（设置为500ms）
* Frustration threshold——响应时间失败阈值（设置为1500ms）

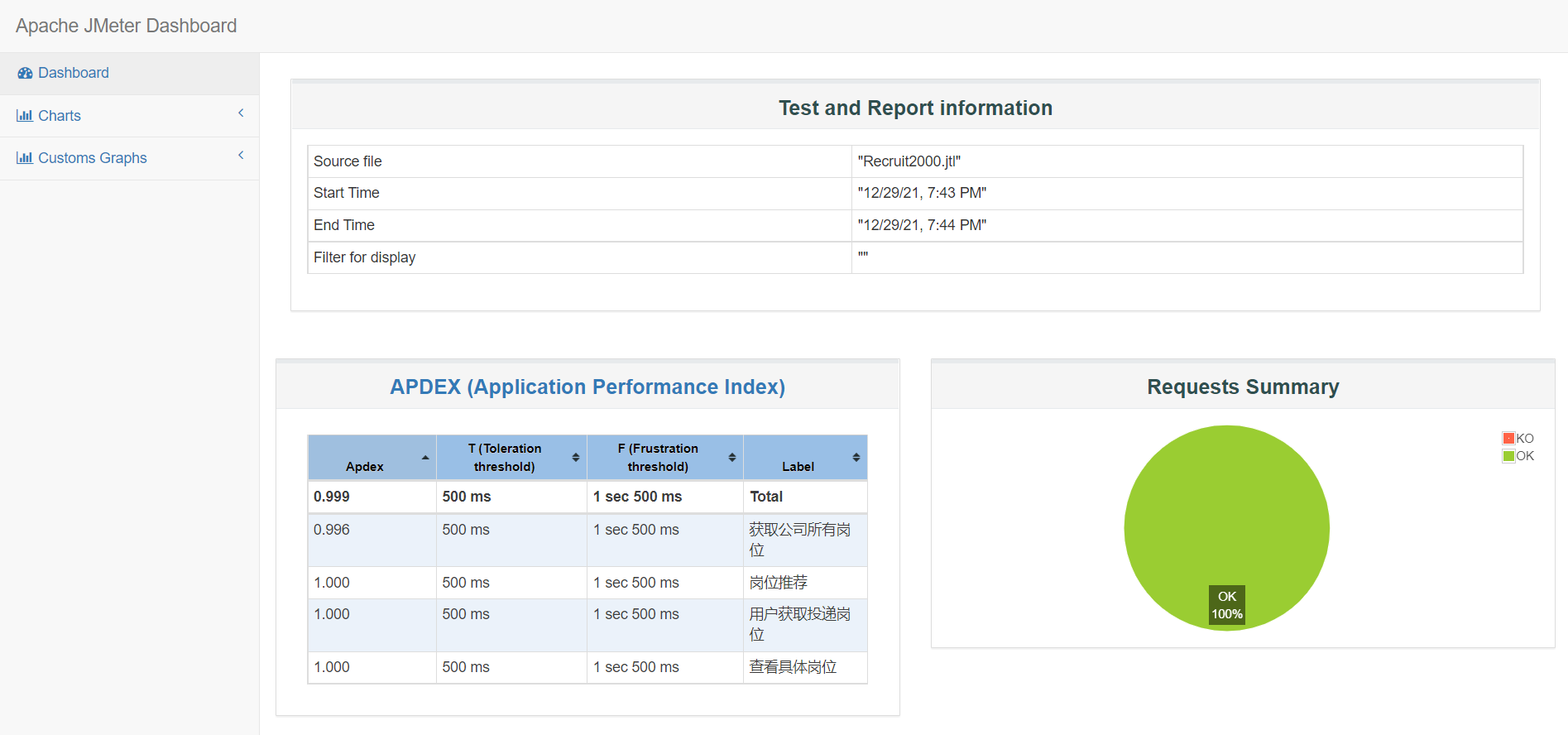
**User**



**Tweet**

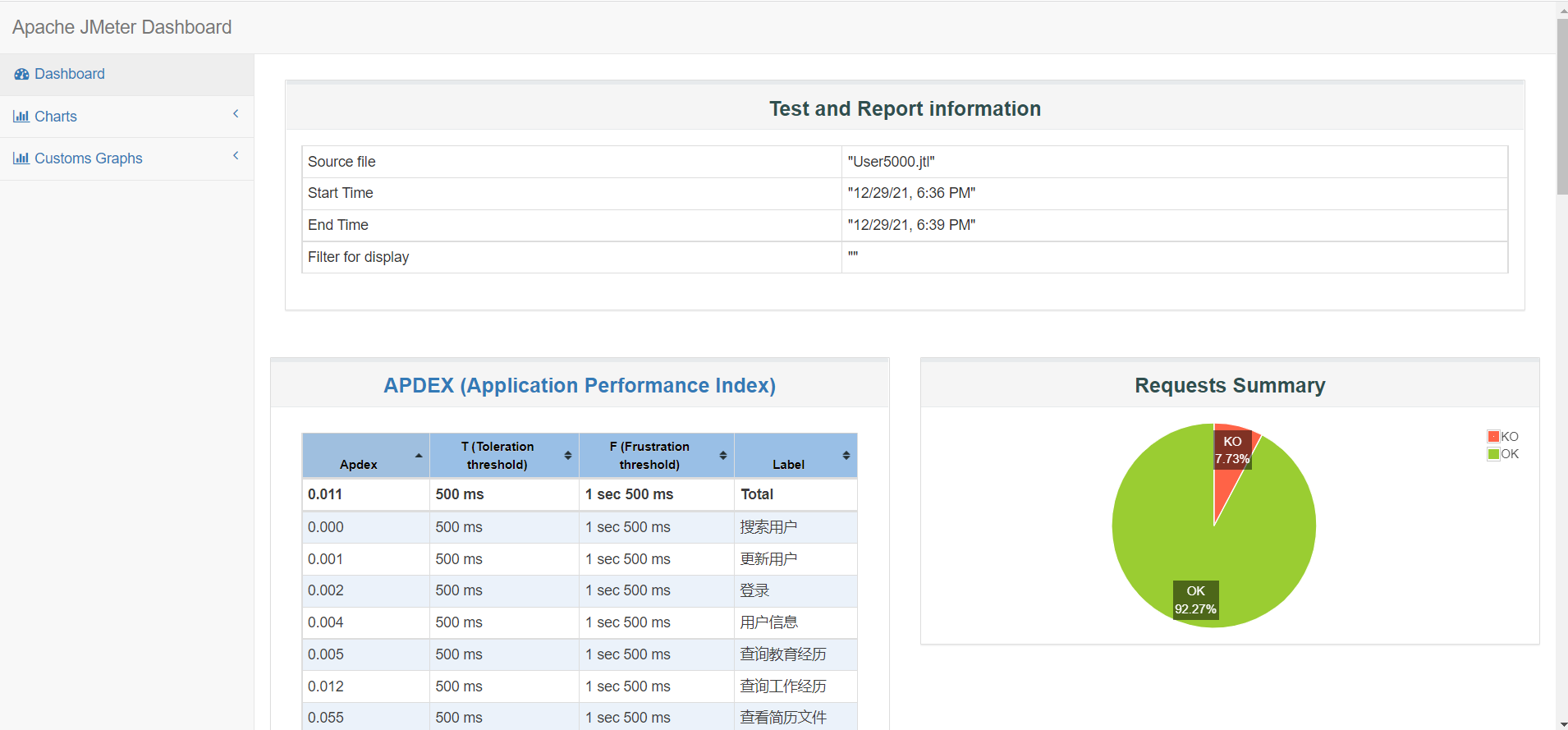


**Recruitment**

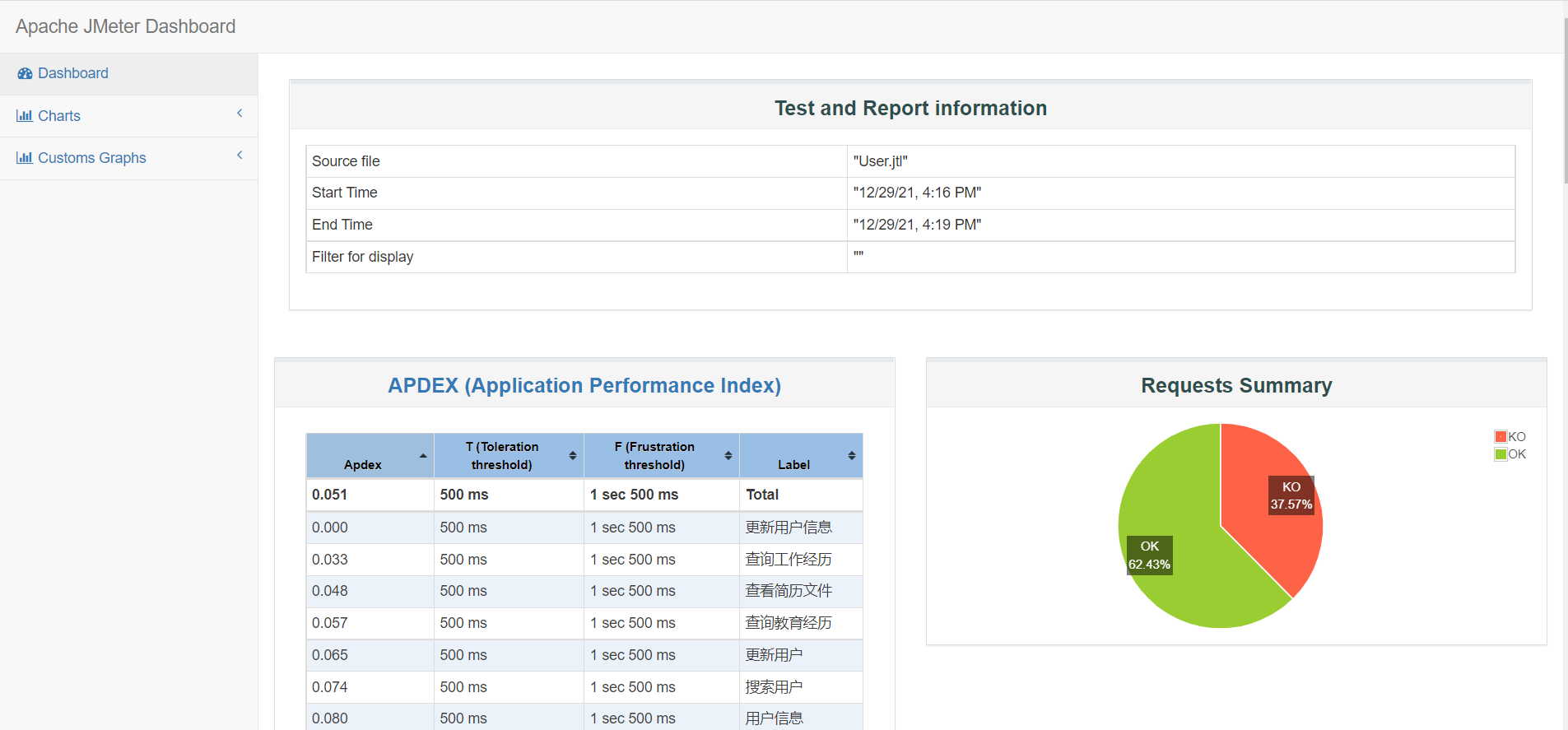


可以看到，在线程并行度2000的压力下，所有微服务模块均能保持良好的性能。而当线程并行度达到5000时，Apdex将明显下降

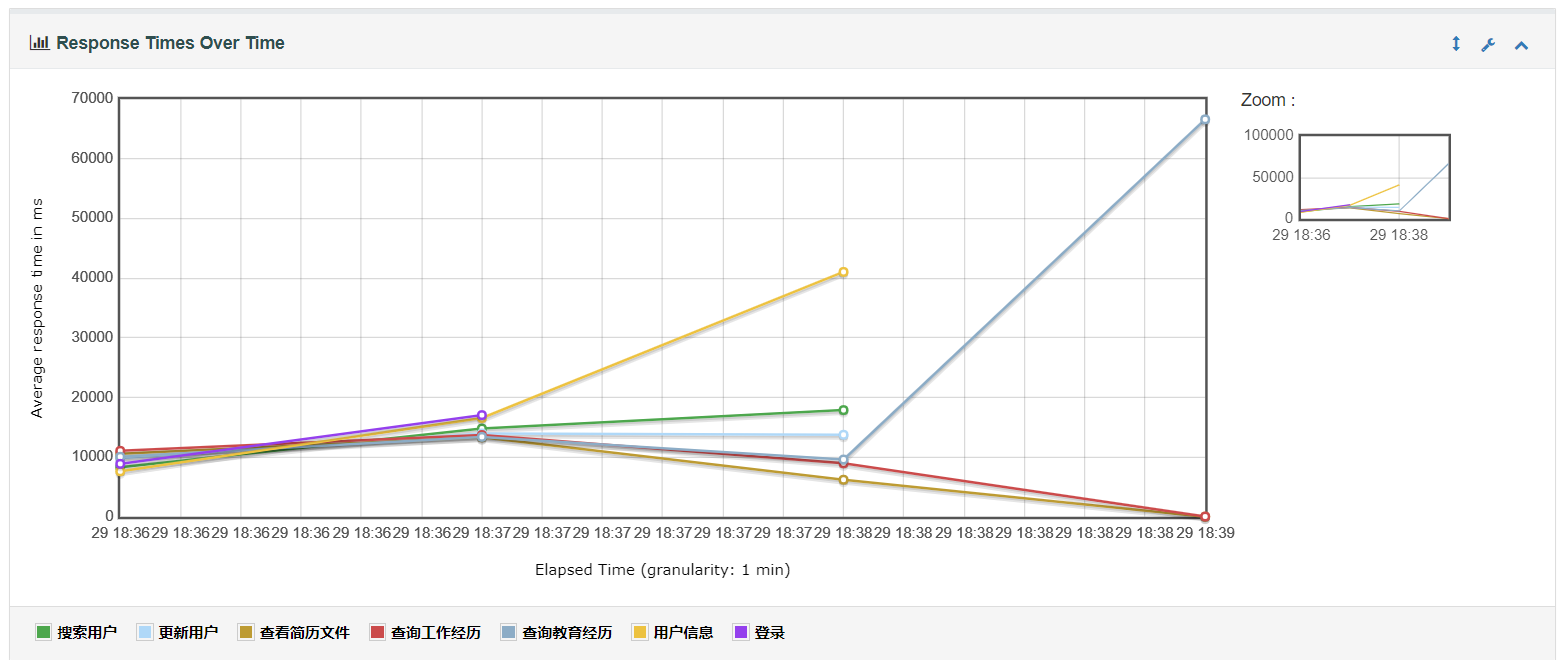
**5000**

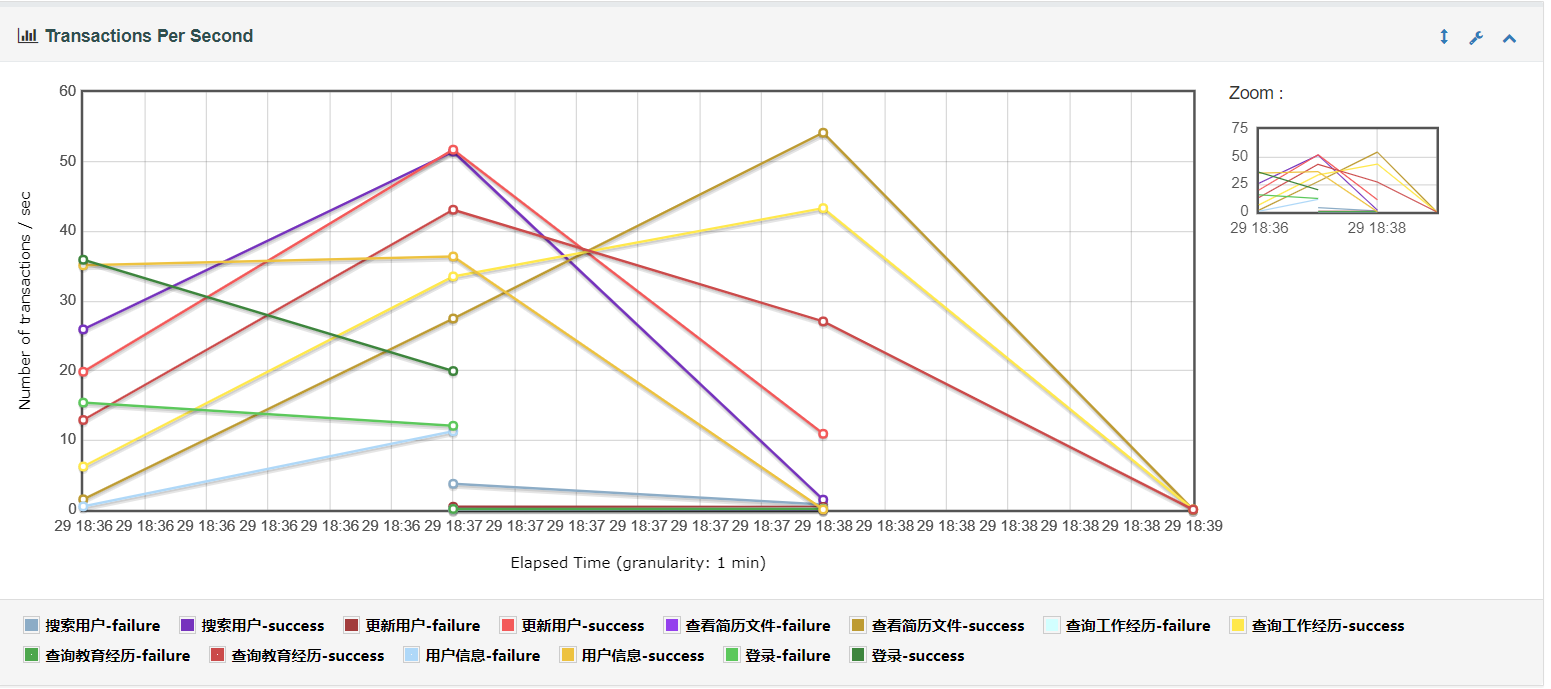


**10000**



在5000的线程并行下，所有请求的响应时间及事务吞吐量变化曲线如下：





经过测试总结，我们项目的最佳并发数约在3000左右。当线程并发度在3000以下时，系统能保持较良好的Apdex

**附录**

数据库E-R图

