# IKnow 系统设计说明书

# 团队：刚下飞机

# 目录

1 引言 ……………………………………………..………………………………………………………………………………………………1  
1.1 编写目的 ....................................................................................................................................1  
1.2 项目背景和内容概要 ................................................................................................................1  
1.3 相关资料、缩略词和术语解释 ................................................................................................1  
1.4 系统开发环境 ............................................................................................................................2  
1.5 系统运行环境 ............................................................................................................................3

2系统结构设计 ……………………………………………………………………………………………………………………………….3  
2.1 软件主体结构 ............................................................................................................................3  
2.2 主要功能数据流图 ....................................................................................................................4  
2.3 软件功能结构 ............................................................................................................................4

3软件模块设计 ……………………………………………………………………………………………………………………………….4  
 3.1 问题子系统功能模块 .................................................................................................................4  
 3.2 管理员子系统功能模块 ...........................................................................................................14  
 3.4 回答子系统功能模块 ...............................................................................................................23  
 3.5 评论子系统功能模块 ...............................................................................................................33  
 3.6 账户子系统功能模块 ...............................................................................................................39  
 3.3 回复子系统功能模块 ...............................................................................................................51

4 接口设计 …………………………………………………………………………………………………………………………………….57  
 4.1 外部接口 ...................................................................................................................................57  
 4.2 内部接口 ...................................................................................................................................59

5 系统安全和权限设计 ……......……………...………………………………………………………………………………………59  
 5.1 系统安全 ...................................................................................................................................59  
 5.2 权限设计 ...................................................................................................................................59

# 引言

**1.1 编写目的**

本系统设计说明书是关于“IKnow”的系统设计，本说明书描述了IKnow项目的系统运行环境设计，包括系统运行环境结构设计、硬件设备配置清单、软件平台配置清单；描述了系统结构设计，包括软件总体结构设计、软件技术架构设计、软件功能结构设计、软件功能模块清单；描述了功能模块设计，包括以下几个子功能模块：问题子系统功能模块、管理员子系统功能模块、回答子系统功能模块、账户子系统功能模块、评论子系统功能模块、回复子系统功能模块。

本系统设计说明书的编写目的：可以为之后的开发和部署提供参考。方便日后开发时的具体实现。

**1.2 项目背景和内容概要**

**项目名：**IKnow

**开发团队：**刚下飞机

**成员：**卓晓鑫 张庭博 朱凯文 张春翔 池政涛 郭秋中 郭福强 胡海江

**内容概要：**为了解决大学生提出学习相关问题不方便的问题，让大学生的提出的问题能够被更多人看见、回答。同时也可以帮助那些性格内向的同学可以有机会解决自己所面临的问题。并且，教师的解答或高评分的优质回答可以帮助有类似疑问的同学。该应用提供了一个可供同学与老师、同学与同学之间交流的平台。

**1.3 相关资料、缩略词和术语解释**

|  |  |
| --- | --- |
| **缩略词和术语** | **解释** |
| IDEA | IDEA 全称 IntelliJ IDEA，是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具。 |
| JavaEE | J2EE的新名称，J2EE的全称是Java 2 Platform Enterprise Edition，它是由SUN公司领导、各厂家共同制定并得到广泛认可的工业标准，或者说，它是在SUN公司领导下，多家公司参与共同制定的企业级分布式应用程序开发规范。目前，J2EE是市场上主流的企业级分布式应用平台的解决方案 |
| Spring | Spring是一个开源框架，它由Rod Johnson创建。它是为了解决企业应用开发的复杂性而创建的。 |
| Struts | 通过采用[Java Servlet](user_cancel)/[JSP](user_cancel)技术，实现了基于[Java EE](user_cancel) [Web](user_cancel)应用的Model-View-Controller（[MVC](user_cancel)）[设计模式](user_cancel)的应用框架，是MVC经典设计模式中的一个经典产品。 |
| Hibernate | Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装。 |
| Vue | Vue 是一套用于构建用户界面的渐进式JavaScript框架。 |
| Maven | Maven项目对象模型(POM)，可以通过一小段描述信息来管理项目的构建，报告和文档的项目管理工具软件。 |
| CDN | CDN的全称是Content Delivery Network，即[内容分发网络](user_cancel)。CDN是构建在现有网络基础之上的智能虚拟网络，依靠部署在各地的边缘服务器，通过中心平台的负载均衡、内容分发、调度等功能模块，使用户就近获取所需内容，降低网络拥塞，提高用户访问响应速度和命中率。 |

**1.4 系统开发环境**

操作系统：Window 10

IDE（集成开发环境）：IntelliJ IDEA

JDK：1.8

**1.5 系统运行环境**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **服务机** | 核心数（核） | | 内存（GB） | 存储空间（GB） | | 网络（Mbps） |
| 阿里云ESC | 1 | | 2 | 40 | | 1 |
| 阿里云RDS | 1 | | 1 | 20 | | 1 |
| **软件** | | **主版本号** | | | **次版本号** | | |
| Tomcat | | 9 | | | 0 | | |
| Mysql | | 5 | | | 7 | | |
| JRE | | 8 | | | 0 | | |

# 系统结构设计

**2.1 软件主体结构**

项目采用BS（浏览器/服务器）模式，主体分为三层：

**表现层（视图层）：**

表现层使用基于Vue.js的前端开发。

**行为层（应用层）：**

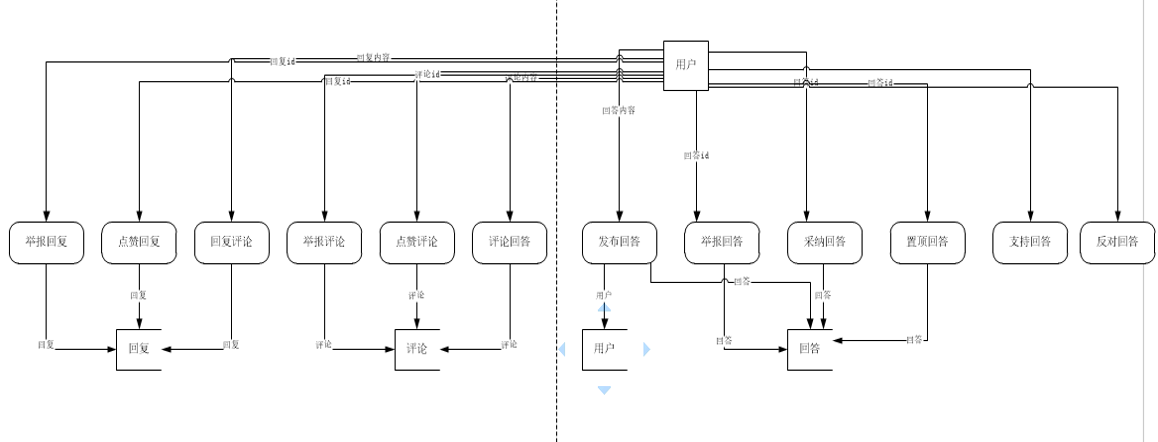
后端使用JavaEE的SSH（Spring + Struts + Hibernate）框架进行开发。

开发模式参照JavaEE分层架构，分别是DAO（数据访问对象）层、Domain（领域对象）层、Service（业务逻辑）层、Controller（控制器）层。

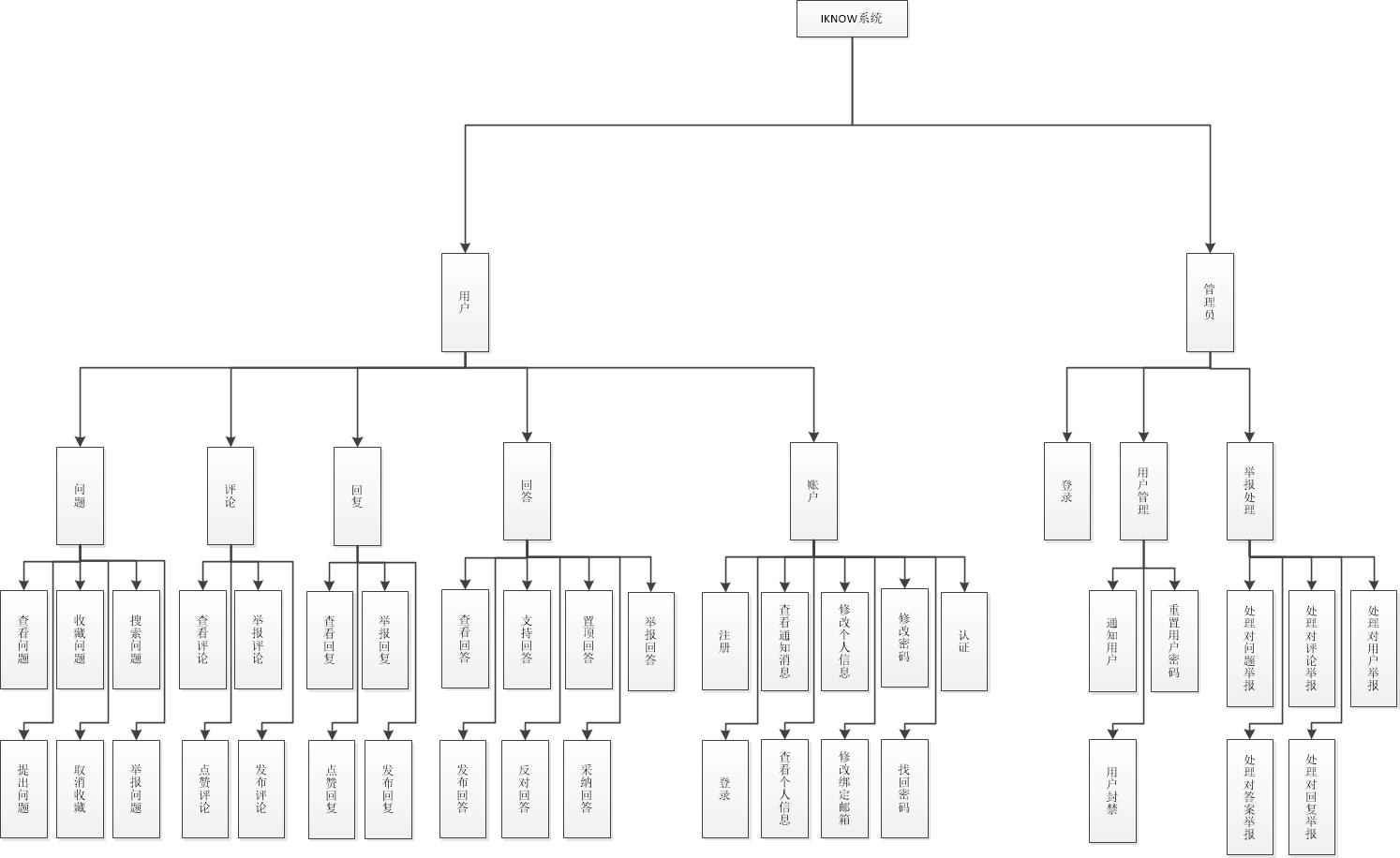
**持久层（数据层）：**

持久层使用MySql进行数据存储。

**2.2 主要功能数据流图**



**2.3 软件功能结构**



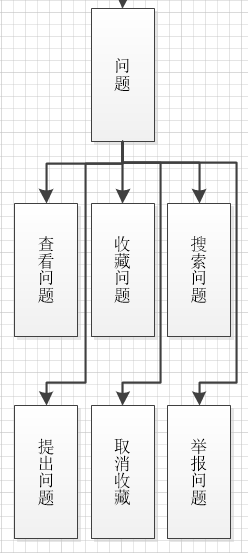
# 软件模块设计

**3.1. 问题子系统功能模块**

**3.1.1.  功能模块描述及流程**

问题的功能模块由查看问题、提出问题、搜索问题、收藏问题和举报问题等主要功能模块组成。

如图：



**3.1.2. 问题模块设计说明**

**3.1.2.1. 程序模块及功能描述**

用户在首页可以查看问题列表，提出问题、搜索问题、收藏问题与举报问题。点击问题可以进入问题的详细页面。

1. 用户在首页可以进行查看问题的功能。查看的问题分为系统推荐的热门问题与收藏的问题。
2. 用户在首页点击提出问题按钮进入提出问题界面。
3. 用户有两种方式搜索问题。一是通过选择问题分类来找到相应类别的内容。二是通过搜索框，寻找含有问题或回答中关键字的问题。
4. 用户可以在问题列表中，或者在问题的详细页面中点击收藏按钮，收藏问题。被收藏可以在用户的个人中心找到。
5. 在问题的详细信息界面或回答的具体页面中，点击举报按钮，填入举报理由即可对问题或回答进行举报。

**3.1.2.2. 性能**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 性能指标名称 | 优先级 | 性能指标要求 |
| 1 | 数据精度 | A | 数据不出错，保留较高精度 |
| 2 | 灵活性 | C | 能在不同客户端操作 |
| 3 | 响应时间 | A | 在2S内处理 |
| 4 | 适应性 | A | 能灵活适应不同系统和浏览器 |

**3.1.2.3. 输入数据项及格式**

**输入数据内容**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据格式 | 数据约束 | 输入来源 |
| a | question\_id | 问题id | Integer | 4字节 | 举报问题界面 |
| b | repost\_description | 举报描述 | String | 256字节 | 举报问题界面 |
| c | report\_reason\_id | 举报原因类型rid | Integer | 4字节 | 举报问题界面 |
| d | user\_id | 举报者id | Integer | 4字节 | 举报问题界面 |
| e | question\_text | 问题描述 | String | 256字节 | 搜索问题界面 |
| f | question\_id | 问题id | Integer | 4字节 | 查看问题界面 |
| g | question\_content | 问题内容 | String | 1024字节 | 发布问题界面 |
| h | user\_id | 用户id | Integer | 4字节 | 发布问题界面 |
| i | question\_type\_id | 问题类别id | Integer | 4字节 | 发布问题界面 |
| j | user\_id | 用户id | Integer | 4字节 | 收藏问题界面 |
| k | question\_id | 问题id | Integer | 4字节 | 收藏问题界面 |

**输入数据样式**

A．举报问题

question\_id：11

report\_des：“这种问题也敢放出来举报举报！”

B.  辅助修改密码

user\_id：1

**3.1.2.4. 输出数据项及格式**

输入数据内容

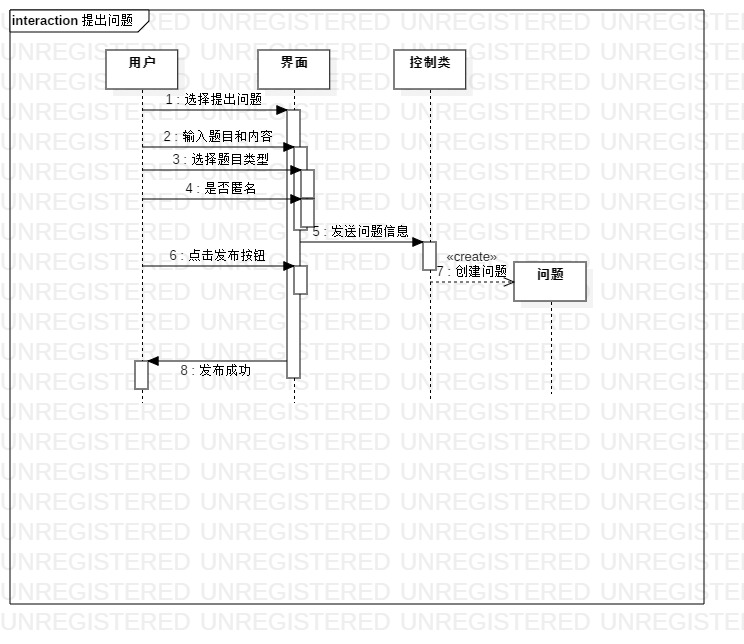
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 项标识 | 项名称 | 数据类型 | 数据约束 | 输出来源 |
| a | question\_content | 问题内容 | String | 1024字节 | 显示问题界面 |
| b | message | 回复消息 | String | 512字节 | 举报成功界面 |

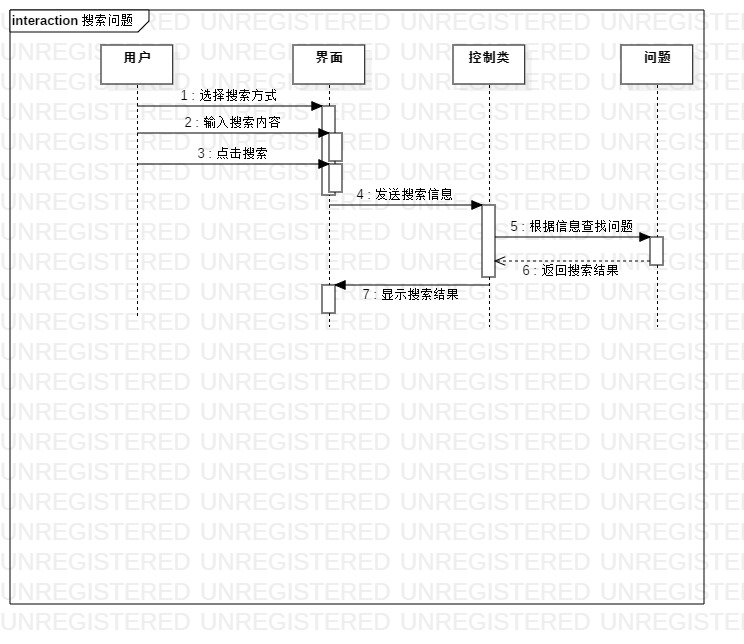
输出数据样式

question\_content：怎么画关系数据库的依赖。

message：已举报成功，等待管理员处理

**3.1.2.5. 流程处理逻辑**

提出问题：

搜索问题：

**3.1.2.6. 设计方法（算法）**

1. 查看问题：

控制类根据界面类发送来的问题id，从 问题表 中获取对应的问题对象，同时获取该问题的回答对象，将两者发送给界面类。

1. 提出问题：

控制类根据界面类发来的问题内容、发布者id、匿名标志构建问题对象，存储到问题表，然后根据发布者id构建消息对象，存储到消息表。

1. 搜索问题

控制类根据界面类发来的问题内容关键字，或者问题的类别获取相应的问题对象，并获得该问题的回答对象，将两者发布给界面类。

1. 举报问题：

控制类根据界面类发来的举报者id、问题id、举报原因类型rid、举报描述，构建举报问题记录对象，存储到举报问题记录表中。

1. 收藏问题：

控制类根据界面类发来的用户id，问题id，构建收藏问题记录，存储到收藏问题记录表

**3.1.2.7. 接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | **请求方式** | **说明** | **请求数据** | **响应数据** |
| question/viewQuestion.action | GET | 查看问题 | qid=? | 问题主页 |
| question/askQuestion.action | POST | 提出问题 | uid=?&question=?&category=?&anon=? | 新增问题主页 |
| question/searchQuestion.action | GET | 搜索问题 | message=? | 搜索结果 |
| question/reportQuestion.action | POST | 举报问题 | qid=?&uid=?&report=? | 举报成功标志 |
| question/collectQuestion.action | POST | 收藏问题 | uid=?&qid=? | 收藏成功标志 |

**3.1.2.8. 存储分配**

作为核心模块，预计存储量占比达到15%以上。

**3.1.2.9. 出错处理**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 错误编号 | 错误名称 | 错误原因 | 错误提示信息 | 处理方式 | 备注 |
| a | 404 | 获取问题信息失败 | 无法获取改问题信息！ | 跳转返回前一个页面继续查看 | 无法获取包括服务器出错和问题已删除 |
| b | 503 | 服务器错误 | 无法连接服务器！ | 提示无法连接到服务器并跳转回原界面 | 无 |

**3.1.2.10. 数据库结构**

|  |  |  |
| --- | --- | --- |
| 序号 | 表中文名称 | 表英文名称 |
| a | 用户表 | user |
| b | 评论表 | comment |
| c | 问题表 | question |
| d | 回答表 | answer |
| e | 回复表 | reply |
| f | 举报记录表 | report |

**3.1.2.11. 注释设计**

参考《阿里巴巴Java开发手册》进行注释。

**3.1.2.12. 限制条件**

无

**3.1.2.13. 测试计划**

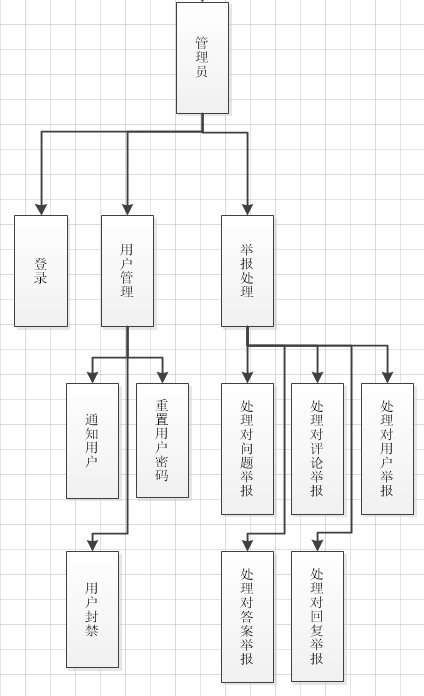
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 查看问题、提问功能及举报问题 | 初始化界面 | 操作 | 界面是否完整 | 界面各个部分是否运作情况 |
| 问题列表界面 | 在列表中浏览问题 | 按钮响应、字数检验、消息显示 | 列表中可以查看问题 |
| 查看问题界面 | 在列表中点击问题可以进入问题查看页面 | 按钮响应、字数检验、消息显示 | 在列表中点击问题可以正确跳转到查看问题界面 |
| 提问界面 | 选择发送图片 | 按钮响应、字数检验、消息显示 | 发送成功，页面显示成功信息 |
| 输入超过255字节限制的文字内容 | 按钮响应、字数检验、消息显示 | 发送失败，页面显示失败，字数超过限制提示 |
| 输入小于255字节的信息与图片 | 按钮响应、字数检验、消息显示 | 发送成功，页面显示成功信息 |
| 举报界面 | 输入空的举报内容 | 按钮响应、字数检验、消息显示 | 发送失败，页面显示失败，字数不符合规范提示 |
| 输入正确的举报内容 | 按钮响应、字数检验、消息显示 | 发送成功，页面显示成功信息 |

**3.1.2.14. 尚未解决的问题**  
无

**3.2. 管理员子系统功能模块**

**3.2.1.  功能模块描述及流程**

管理员的功能模块由登录，重置密码，封禁，举报处理等功能等主要子功能模块组成，详细参见下图：



**3.2.2. 管理员模块设计说明**

**3.2.2.1. 程序模块及功能描述**

该程序模块主要包括：管理员登录、用户管理、举报处理和通知用户等功能。

1. 管理员登录

用于验证管理员身份，只有登录成功才能使用以下三个功能。

1. 用户管理
2. 重置用户密码

用于重置用户密码，如果用户忘记密码可帮助其重置

1. 用户封禁

对于违规用户，如果该用户的问题、回答、评论、回复、个人信息有不符合规定的情况，可视严重程度进行封禁处罚，极其严重者可永久封禁。

1. 举报处理
2. 处理对问题举报

如果用户（包括匿名用户）提出的问题违反规定，可视严重程度给予相应处罚

1. 处理对答案举报

如果用户（包括匿名用户）作出的答案违反规定，可视严重程度给予相应处罚

1. 处理对评论举报

如果用户（包括匿名用户）发表的评论违反规定，可视严重程度给予相应处罚

1. 处理对回复举报

如果用户（包括匿名用户）的回复违反规定，可视严重程度给予相应处罚

1. 处理对用户举报

如果用户（包括匿名用户）存在违反规定行为，如：个人信息，问题，回答，评论，回复等，可视严重程度给予相应处罚

1. 通知用户

该功能用于通知用户有关信息，比如通知用户密码重置成功，或通知其已被封禁。举报他人成功的用户，通知其举报有效，已对违反规定用户进行处罚。被举报用户若存在违反规定行为，则通知其违反了何种规定，告知其处罚内容。

**3.2.2.2. 性能**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 性能指标名称 | 优先级 | 性能指标要求 |
| 1 | 数据精度 | A | 数据不出错，保留较高精度 |
| 2 | 灵活性 | C | 能在不同客户端操作 |
| 3 | 响应时间 | A | 能在30秒内响应 |
| 4 | 适应性 | **A** | 能灵活适应不同系统和浏览器 |

**3.2.2.3. 输入数据项及格式**

1. **输入数据内容**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据格式 | 数据约束 | 输入来源 |
| a | user\_id | 用户id | Integer | 4字节 | 点击举报按钮，辅助密码修改 |
| b | question\_id | 被举报问题id | Integer | 4字节 | 点击举报按钮 |
| c | answer\_id | 被举报回答id | Integer | 4字节 | 点击举报按钮 |
| d | reply\_id | 被举报回复id | Integer | 4字节 | 点击举报按钮 |
| e | comment\_id | 被举报评论id | Integer | 4字节 | 点击举报按钮 |
| f | repost\_description | 举报描述 | String | 256字节 | 点击举报按钮 |

表 3‑2输入数据内容列表

**（2）输入数据样式**

A．举报问题

   question\_id：11

report\_des：“这种问题也敢放出来举报举报！”

B.  辅助修改密码

    user\_id：1

**3.2.2.4. 输出数据项及格式**

1. 输出数据内容

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 项标识 | 项名称 | 数据类型 | 数据约束 | 输出来源 |
| a | user\_id | 用户id | Integer | 4字节 | 处理举报页面 |
| b | user\_name | 用户名称 | String | 128字节 | 处理举报页面 |
| c | repost\_des | 举报描述 | String | 256字节 | 处理举报页面 |
| d | repost\_time | 举报时间 | Datetime | 0字节 | 处理举报页面 |
| e | answer\_id | 被举报回答id | Integer | 4字节 | 处理举报页面 |
| f | comment\_id | 被举报评论id | Integer | 4字节 | 处理举报页面 |
| g | question\_id | 被举报问题id | Integer | 4字节 | 处理举报页面 |
| h | reply\_id | 被举报回复id | Integer | 4字节 | 处理举报页面 |

1. 输出数据样式

user\_id:1

user\_name:二蛋

repost\_des:色情信息

repost\_time:2020-04-10 00:00:00

answer\_id:1

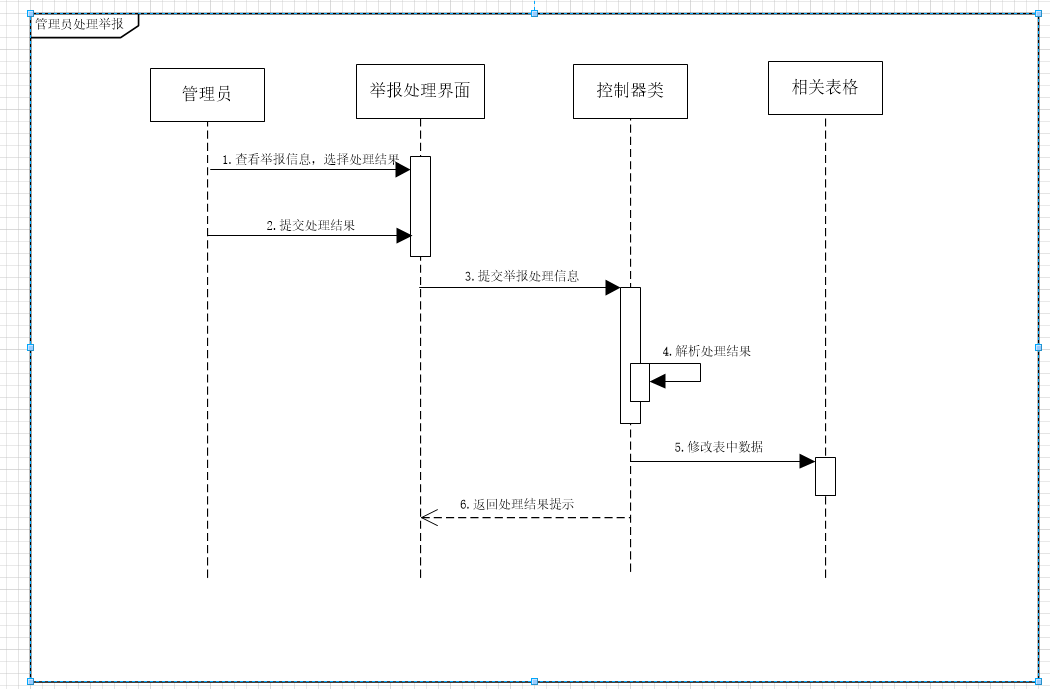
comment\_id:1

question\_id:1

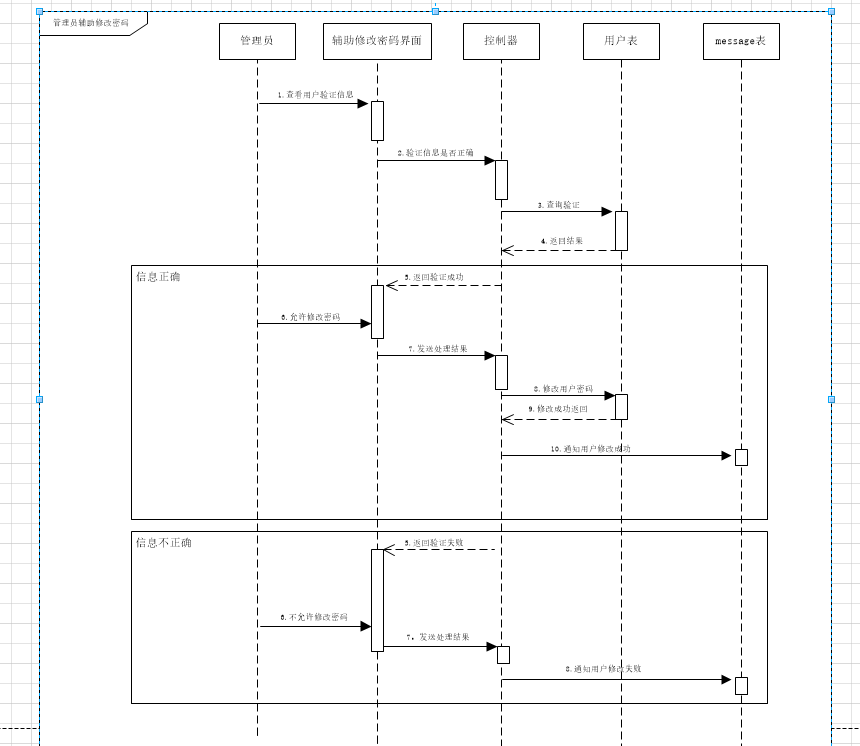
reply\_id:1

**3.2.2.5. 流程处理逻辑**

1. 管理员处理举报



1. 管理员辅助修改密码



**3.2.2.6. 设计方法（算法）**

1. 单击处理举报按钮时，控制类将举报处理结果存入数据库并根据举报用户id返回相关通知信息。
2. 单击确认辅助用户密码修改时，根据用户id更改用户密码，并返回相关通知信息。

**3.2.2.7. 接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | **请求方式** | **说明** | **请求数据** | **响应数据** |
| admin/login.action | POST | 管理员登录 | adminName=?&password=? | 后台主页 |
| admin/logout.action | POST | 管理员登出 |  | 无 |
| admin/noticeUser.action | POST | 通知用户 | uid=?$message=? | 成功标识 |
| admin/resetUserPwd.action | POST | 重置用户密码 | uid=?&pwd=? | 成功标识 |
| admin/banUser.action | POST | 封禁用户 | uid=? | 成功标识 |
| admin/handReport.action | POST | 处理举报信息 | type=?uid=?qid=?aid=?cid=? | 成功标识 |
| admin/listUser.action | GET | 列出用户（分页） | start=? | 用户列表 |
| admin/listReport.action | GET | 列出举报（分列） | start=? | 举报列表 |

**3.2.2.8. 存储分配**

作为核心模块，预计存储量占比达到5%以上。

**3.2.2.9. 出错处理**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 错误编号 | 错误名称 | 错误原因 | 错误提示信息 | 处理方式 |
| a | 404 | 举报信息未找到 | 找不到信息 | 前端显示找不到 |
| b | 503 | 服务器错误 | 无法连接服务器 | 前端显示连不上 |

**3.2.2.10. 数据库结构**

|  |  |  |
| --- | --- | --- |
| 序号 | 表中文名称 | 表英文名称 |
| a | 用户表 | user |
| b | 评论表 | comment |
| c | 问题表 | question |
| d | 回答表 | answer |
| e | 回复表 | reply |
| f | 管理员表 | admin |

**3.2.2.11. 注释设计**

参考《阿里巴巴Java开发手册》进行注释。

**3.2.2.12. 限制条件**

无

**3.2.2.13. 测试计划**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块 | 界面 | 用户行为 | 响应 | 结果 |
| 处理问题举报 | 初始化界面 |  | 是否完整 | 各个部件完整 |
| 封禁 | 点击封禁按钮 | 是否成功 | 封禁成功 |
| 违规信息 |  | 是否正确 | 显示正确 |

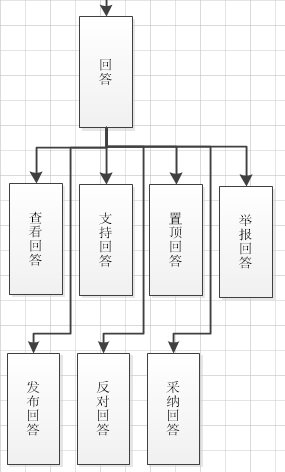
**3.2.2.14. 尚未解决的问题**

无

**3.3. 回答子系统功能模块**

**3.3.1.  功能模块描述及流程**

回答的功能模块由评论回答、发布回答、举报回答、置顶回答、采纳回答、支持回答、反对回答等主要子功能模块组成。

如图：

**3.3.2. 回答模块设计说明**

**3.3.2.1. 程序模块及功能描述**

1. 用户在问题详情界面点击 写回答 按钮，进入创建回答界面。
2. 用户在问题详情界面中点击任意回答概要，进入回答详情界面
3. 用户填写完问题概述、详情，并选择问题分类与是否匿名后，问题将记录到 问题表 ，同时显示新发布的问题的问题详情页面。
4. 用户在回答详情界面点击 举报问题 ，进入举报界面，选择举报原因并填写举报理由后，举报将记录到 举报记录表 ，同时显示系统受到举报。
5. 用户在回答的评论区中在输入框中输入评论内容，并点击发布后，该条评论将记录到 评论表 中，并创建一条给答主的消息记录插入 消息表 中，同时评论区中显示用户的评论。
6. 用户在浏览回答详情界面时，若用户为该回答所在的问题的提出者，则显示 采纳回答和置顶回答 按钮，否则不显示
7. 用户点击采纳按钮后，回答所对应的问题的采纳回答id将设为该回答id，该回答的答主所对应的 用户记录 徽章数加一，创建一条给回答者的消息记录插入 消息表 中，在问题详情界面，该回答将被置为第一个回答。
8. 用户点击置顶按钮后，回答记录设置置顶标志为True，该回答将被置于除采纳回答之外的所以回答的上方，并创建一条给回答者的消息记录插入 消息表 中。
9. 用户在回答详情界面点击赞同按钮后，将新增一条回答赞同记录到 回答赞同记录表 ，并该回答的赞同数加一，并创建一条给回答者的消息记录插入 消息表 中。
10. 用户在回答详情界面点击反对按钮后，将新增一条回答反对记录到 回答反对记录表，并创建一条给回答者的消息记录插入 消息表 中。

**3.3.2.2. 性能**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 性能指标名称 | 优先级 | 性能指标要求 |
| 1 | 数据精度 | A | 数据不出错，保留较高精度 |
| 2 | 灵活性 | C | 能在不同客户端操作 |
| 3 | 响应时间 | A | 能在3秒内响应 |
| 4 | 适应性 | **A** | 能灵活适应不同系统和浏览器 |

**3.3.2.3. 输入数据项及格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据格式 | 数据约束 | 输入源 |
| a | comment\_content | 评论内容 | String | 1000字节 | 评论回答界面 |
| b | user\_id | 评论者id | Integer | 4字节 | 评论回答界面 |
| c | answer\_id | 回答id | Integer | 4字节 | 评论回答界面 |
| d | answer\_content | 回答内容 | Integer | 10000字节 | 发布回答界面 |
| e | user\_id | 发布者id | Integer | 4字节 | 发布回答界面 |
| f | question\_id | 问题id | Integer | 4字节 | 发布回答界面 |
| g | anonymous | 匿名标志 | String | 10字节 | 发布回答界面 |
| h | answer\_id | 回答的id | Integer | 4字节 | 举报回答界面 |
| i | report\_description | 举报描述 | String | 256字节 | 举报回答界面 |
|  | report\_reson\_id | 举报原因类型rid | Integer | 4字节 | 举报回答界面 |
|  | user\_id | 举报者id | Integer | 4字节 | 举报回答界面 |
| j | answer\_id | 回答的id | Integer | 4字节 | 题主点击置顶按钮 |
| k | answer\_id | 回答的id | Integer | 4字节 | 题主点击采纳按钮 |
| l | answer\_id | 回答的id | Integer | 4字节 | 用户点击赞同按钮 |
| m | user\_id | 赞同者id | Integer | 4子节 | 用户点击赞同按钮 |
| n | answer\_id | 回答的id | Integer | 4字节 | 用户点击反对按钮 |
| o | user\_id | 反对者id | Integer | 4子节 | 用户点击反对按钮 |
| p | answer\_id | 回答的内容 | Integer | 4字节 | 用户点击查看评论 |

输入样例

comment\_content：这个回答太强了，解决了我的问题

answer\_content:这道题可以使用快速排序

answer\_id：50

user\_id：2

question\_id：33

report\_content：这个回答涉及黄暴

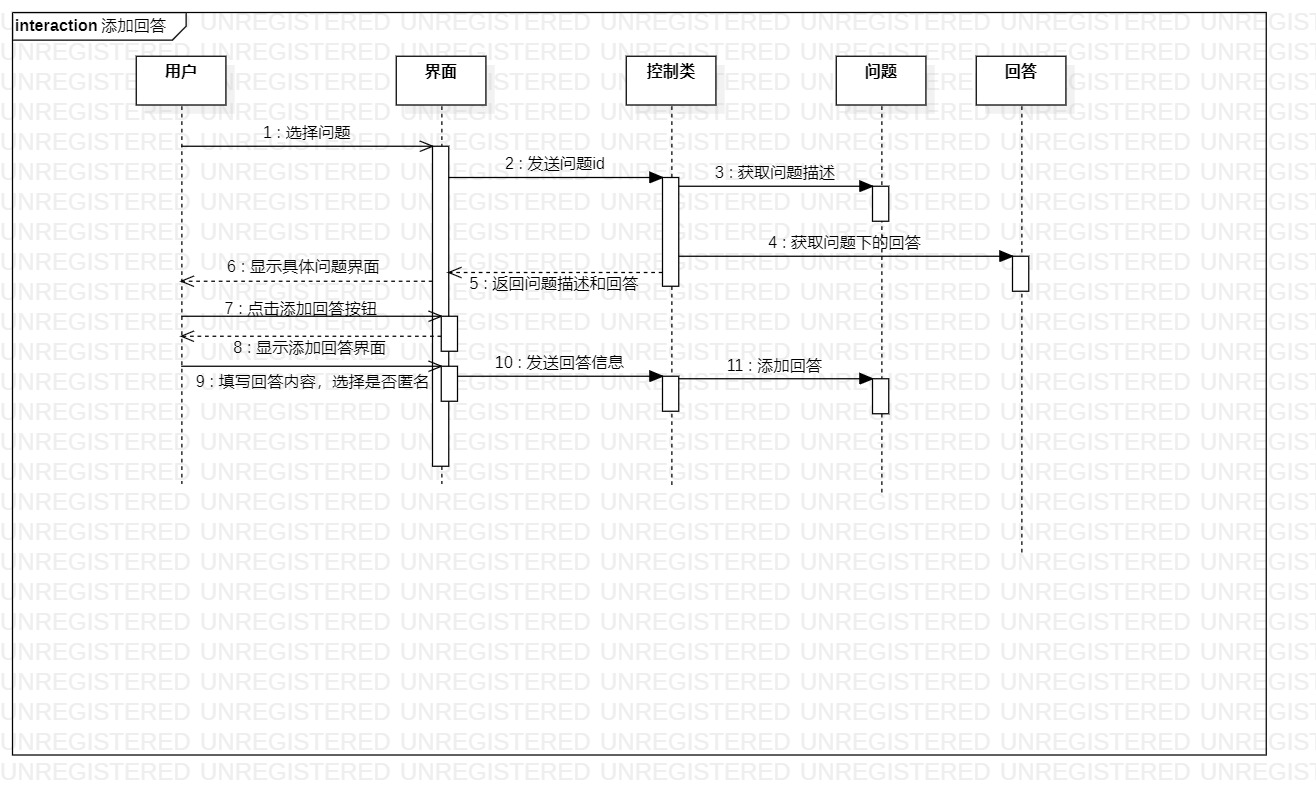
**3.3.2.4. 输出数据项及格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/元素名称 | 数据格式 | 数据约束 | 输出源 |
| a | answer\_content | 回答的内容 | String | 1000字节 | 显示回答界面 |
| b | message\_content | 举报的回复信息 | String | 512字节 | 显示回答界面 |

输出样例

answer\_content:这道题可以使用快速排序

**3.3.2.5. 流程处理逻辑**



**3.3.2.6. 设计方法（算法）**

1. 查看回答：

控制类根据界面类发送来的回答id，从 回答表 中获取对应的回答对象，同时获取该回答的评论对象，将两者发送给界面类。

1. 评论回答：

控制类根据界面类发来的评论者id、回答者id、回答id、评论内容，构建评论对象，将评论对象存储到评论表，然后根据回答者id，构建消息对象，存储到消息表。

1. 发布回答：

控制类根据界面类发来的回答内容、发布者id、问题id、匿名标志构建回答对象，存储到回答表，然后根据发布者id构建消息对象，存储到消息表。

1. 举报回答：

控制类根据界面类发来的举报者id、被举报的回答id、举报原因类型rid、举报描述，构建举报回答记录对象，存储到举报回答记录表中。

1. 采纳回答：

控制类根据界面类发来的问题id，回答id，从问题表中获取问题对象，更新问题对象的采纳回答id为回答id。然后根据回答id从回答表中获取回答对象，再根据回答者id构建消息对象，存储到消息表。

1. 置顶回答：

控制类根据界面类发来的回答id，更新回答表中对应的回答记录的置顶标记为真，然后根据回答id，从回答表中获取回答者id，构建消息对象，存储到消息表。

1. 支持回答：

控制类根据界面类发来的回答id，赞同者id，构建回答赞同记录对象，存储到回答赞同记录表，然后根据回答id，从回答表中获取回答对象，更新回答的赞同数；根据回答者id，构建消息对象，存储到消息表。

1. 反对回答：

控制类根据界面类发来的回答id，反对者id，构建回答赞同记录对象，存储到回答赞同记录表，然后根据回答id，从回答表中获取回答者id，构建消息对象，存储到消息表。

**3.3.2.7. 接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | **请求方式** | **说明** | **请求数据** | **响应数据** |
| answer/viewAnswer.action | GET | 查看回答 | aid=? | 回答主页 |
| answer/publishAnswer.action | POST | 发布回答 | qid=?&uid=?&answer=? | 新增回答主页 |
| answer/commentAnswer.action | POST | 评论回答 | aid=?&uid=?&comment=? | 评论成功标识 |
| answer/reportAnswer.action | POST | 举报回答 | aid=?&uid=?&report=? | 举报成功标识 |
| answer/acceptAnswer.action | POST | 采纳回答 | aid=?&uid=? | 采纳成功标识 |
| answer/topAnswer.action | POST | 置顶回答 | aid=?&uid=? | 置顶成功标识 |
| answer/supportAnswer.action | POST | 支持回答 | aid=?&uid=? | 支持成功标识 |
| answer/opposeAnser.action | POST | 反对回答 | aid=?&uid=? | 反对成功标识 |

**3.3.2.8. 存储分配**

作为核心模块，预计存储量占比达到20%以上。

**3.3.2.9. 出错处理**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 错误编号 | 错误名称 | 错误原因 | 错误提示信息 | 处理方式 | 备注 |
| a | 404 | 回答未找到 | 回答未找到 | 前端提示未找到 | - |
| b | 503 | 服务器错误 | 无法连接服务器 | 前端提示无法连接服务器 | - |

**3.3.2.10. 数据库结构**

|  |  |  |
| --- | --- | --- |
| 序号 | 表中文名称 | 表英文名称 |
| a | 用户表 | user |
| b | 回答表 | answer |
| c | 支持回答表 | approvalAnswer |
| d | 举报记录表 | report |
| e | 反对回答表 | opposalAnswer |
| f | 消息表 | message |

**3.3.2.11. 注释设计**

参考《阿里巴巴Java开发手册》进行注释。

**3.3.2.12. 限制条件**

无

**3.3.2.13. 测试计划**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 回答界面 | 初始化界面 |  | 界面是否完整 | 界面各个部件完整 |
| 查看回答界面 | 点击“查看详情”按钮 | 按钮响应，内容显示，界面跳转 | 查看成功，跳转到回答详情界面 |
| 发布回答界面 | 没有输入回答内容，点击发布按钮 | 按钮响应，字数检验，错误提示 | 发布失败，提示字数错误 |
| 输入回答内容，点击发布按钮 | 按钮响应，字数检验，界面跳转 | 发布成功，跳转到原来的浏览界面 |
| 回答详情界面 | 点击“支持”按钮 | 按钮响应，数据存储，错误提示 | 支持成功 |
| 点击“反对”按钮 | 按钮响应，数据存储，错误提示 | 反对成功 |
| 题主点击“采纳”按钮 | 按钮响应，页面跳转 | 采纳成功，跳转到问题界面 |
| 题主点击“置顶” | 按钮响应，页面跳转 | 置顶成功，跳转到问题界面 |
| 举报回答界面 | 没有输入举报内容，点击举报 | 按钮响应，字数检验，错误提示 | 举报失败，显示错误信息 |
| 输入举报内容，点击举报 | 按钮响应，字数检验，页面跳转 | 举报成功，，返回之前浏览的界面 |

**3.3.2.14. 尚未解决的问题**

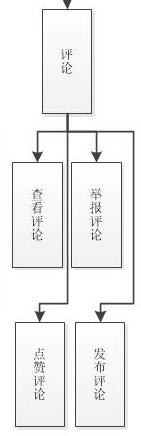
无

**3.4. 评论子系统功能模块**

**3.4.1.  功能模块描述及流程**

评论的功能模块由查看评论、点赞评论、发布评论、举报评论等主要功能模块组成。

如图：



**3.4.2. 评论模块设计说明**

**3.4.2.1. 程序模块及功能描述**

该模块分为查看评论、点赞评论、发布评论、举报评论功能

1. 查看评论

用户可以点击评论按钮，查看所有评论，每条评论包含的信息有：用户名，用户头像，点赞数等

1. 点赞评论

用户可对某个评论进行点赞，点击点赞按钮，对应评论的点赞数加1

1. 发布评论

用户可以对某个回答进行评论，评论会置于对应回答下方

1. 举报评论

如果某条评论存在违反规定的情况，用户可对该评论进行举报，进行举报界面后，可以选择相关评论的违规类型，同时需要进行详细说明才可提交举报

**3.4.2.2. 性能**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 性能指标名称 | 优先级 | 性能指标要求 |
| 1 | 数据精度 | A | 数据不出错，保留较高精度 |
| 2 | 灵活性 | C | 能在不同客户端操作 |
| 3 | 响应时间 | A | 能在3秒内响应 |
| 4 | 适应性 | **A** | 能灵活适应不同系统和浏览器 |

**3.4.2.3. 输入数据项及格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据类型 | 数据约束 | 输入源 |
| a | comment\_id | 评论id | Integer | 4字节 | 评论界面 |
| b | commnet\_Praise | 评论点赞数 | Integer | 4字节 | 评论界面 |
| c | comment\_content | 评论内容 | String | 1000字节 | 评论界面 |
| d | user\_id | 评论者id | Integer | 4字节 | 评论界面 |
| e | answer\_id | 回答id | Integer | 4字节 | 评论界面 |

输入样例

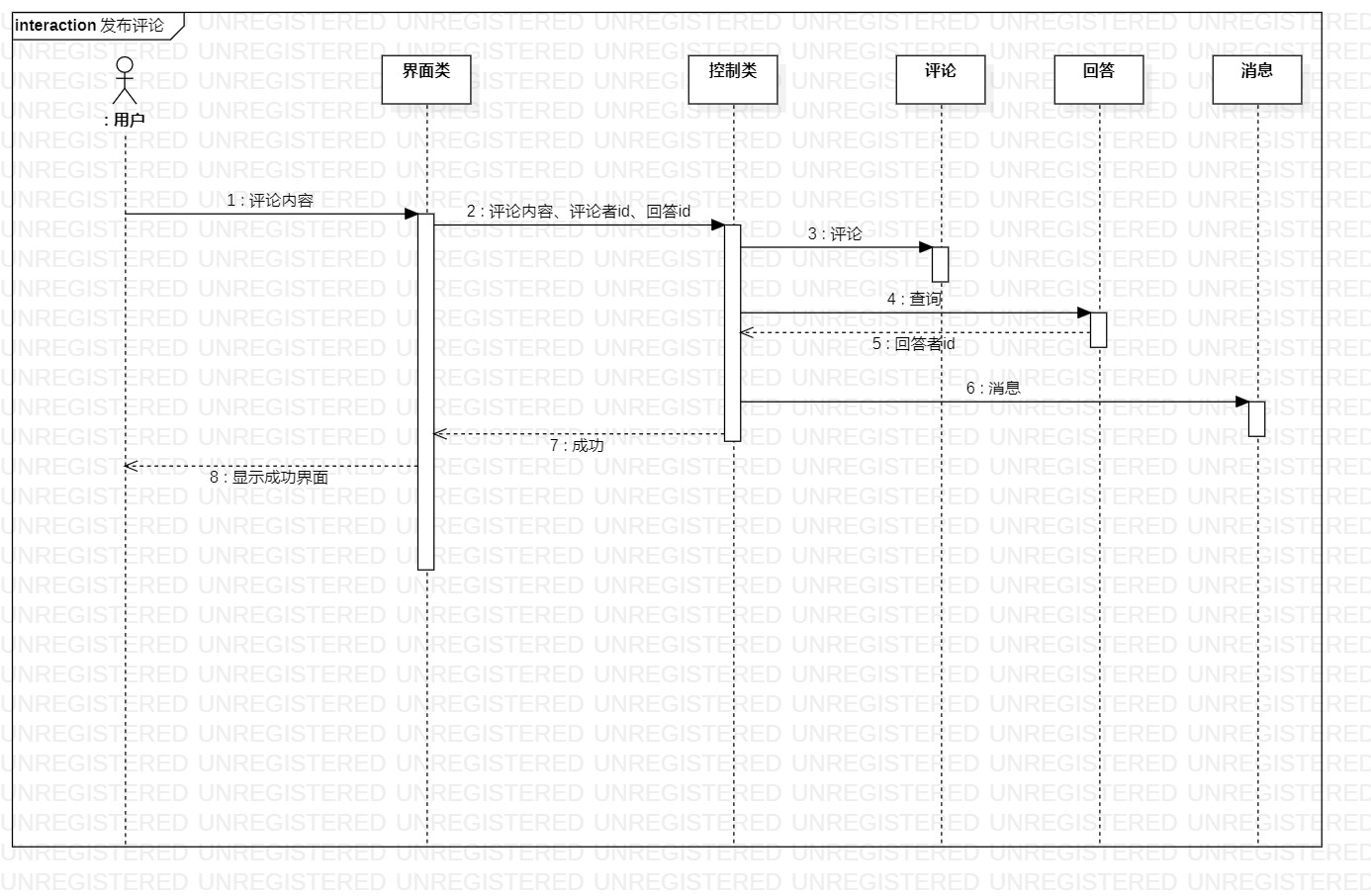
comment\_content：这个回答太强了，解决了我的问题

user\_id: 1

**3.4.2.4. 输出数据项及格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据类型 | 数据约束 | 输入源 |
| a | comment\_content | 评论内容 | String | 1000字节 | 评论界面 |
| b | message\_content | 举报后系统回复的消息内容 | String | 512字节 | 评论界面 |

**3.4.2.5. 流程处理逻辑**



**3.4.2.6. 设计方法（算法）**

1. 查看评论：

控制类根据回答id，从评论表中获取回答下的所有评论，发送给界面类

2. 点赞评论：

控制类根据评论id、点赞者id，构建评论点赞记录对象，存储到评论点赞表，同时更新评论表里该评论的点赞数，然后，获取评论者的id，构建消息对象，存储到消息表

3. 发布评论：

控制类根据评论内容、评论者id、回答id构建评论对象，存储到评论表中，同时根据回答id，查询回答者id，构建消息对象，存储到消息表中

4. 举报评论：

控制类根据举报者id、评论id、举报原因类型rid、举报描述，构建举报评论记录对象，存储到举报评论表。

**3.4.2.7. 接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | **请求方式** | **说明** | **请求数据** | **响应数据** |
| comment/viewComment.action | GET | 查看评论和回复 | aid=? | 评论回复列表 |
| comment/replyComment.action | POST | 回复评论 | cid=?&uid=?&reply=? | 回复成功标识 |
| comment/likeComment.action | POST | 点赞评论 | cid=?&uid=? | 点赞成功标识 |
| comment/reportComment.action | POST | 举报评论 | cid=?&uid=?&report=? | 举报成功标识 |

**3.4.2.8. 存储分配**

作为核心模块，预计存储量占比达到20%以上。

**3.4.2.9. 出错处理**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 错误编号 | 错误名称 | 错误原因 | 错误提示信息 | 处理方式 | 备注 |
| a | 404 | 评论未找到 | 评论未找到 | 前端提示未找到 | - |
| b | 503 | 服务器错误 | 无法连接服务器 | 前端提示无法连接服务器 | - |

**3.4.2.10. 数据库结构**

|  |  |  |
| --- | --- | --- |
| 序号 | 表中文名称 | 表英文名称 |
| a | 评论表 | comment |
| b | 评论点赞记录表 | approvalComment |
| c | 回答表 | answer |
| d | 消息表 | message |
| e | 举报记录表 | reoprt |

**3.4.2.11. 注释设计**

参考《阿里巴巴Java开发手册》进行注释。

**3.4.2.12. 限制条件**

无

**3.4.2.13. 测试计划**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块 | 界面 | 用户行为 | 响应 | 结果 |
| 评论子系统功能模块（查看评论、举报评论、点赞评论、发布评论） | 回答界面 | 点击查看评论按钮 | 是否弹出评论界面 | 界面是否完整 |
| 查看评论 | 点击举报按钮 | 是否弹出举报页面 | 界面是否完整 |
| 点击点赞按钮 | 按钮是否响应 | 点赞图标是否点亮，点赞数是否增加 |
| 点击评论按钮 | 是否响应 | 出现评论输入界面 |
| 举报评论 | （未）选择违规类型，未输入详细说明，点击举报按钮 | 是否举报成功 | 举报失败，提示：请输入详细信息 |
| （未）选择违规类型，输入详细说明，点击举报按钮 | 是否举报成功 | 举报成功 |
| 发布评论 | 输入字数超过限制，点击发布 | 按钮是否响应、字数检验 | 发布失败，提示：字数应小于100 |
| 输入字数为超过显示，点击发布 | 按钮是否响应、字数检验 | 发布成功 |

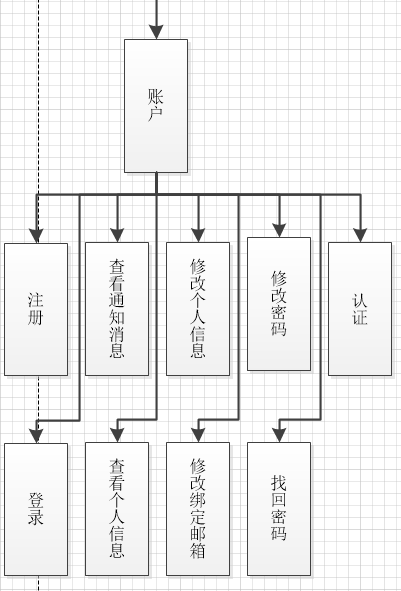
**3.4.2.14. 尚未解决的问题**

无

**3.5. 账户子系统功能模块**

**3.5.1 功能模块描述及流程**

账户的功能模块由注册、登录、查看通知消息、查看个人信息、修改个人信息、重置密码、找回密码、修改绑定邮箱等主要子功能模块组成。如下图所示：



**3.5.2 账户模块设计说明**

**3.5.2.1 程序模块及功能描述**

1. 用户在首页点击注册按钮后进入注册界面。
2. 用户填写用户名、密码、邮箱后，点击完成后，注册成功，进入主页。注册成功后将该用户的信息插入到用户表中存储。
3. 用户在首页填写邮箱、验证码或者密码后，点击登录后，登录成功，进入主页。
4. 用户在个人主页可以选择查看通知消息，查看自己的消息列表。
5. 用户在个人主页可以查看自己的个人信息，如用户名和验证邮箱等。
6. 修改个人信息
7. 修改基本信息：用户可以点击修改基本信息按钮进行进本信息的修改，在填写完要修改的信息后，点击完成，完成修改基本信息操作。
8. 修改认证信息：用户可以点击修改认证信息按钮进行认证信息的修改，在填写完需要修改的认证信息后，点击完成，完成修改认证信息操作。
9. 用户在主页点击找回密码后，进入找回密码界面，填写邮箱和验证码后，发送找回的密码邮件到用户邮箱，完成找回密码操作。
10. 用户在个人主页可以选择修改绑定邮箱，进入修改邮箱界面，填写好新邮箱及验证码，填写完成后确认，完成修改绑定邮箱操作。
11. 用户在个人信息界面选择修改密码，进入修改密码界面，填好邮箱发来的验证码和新的密码并确认后，点击修改按钮，完成修改密码操作。
12. 用户在认证界面输入学校，学工号，姓名，点击认证，完成认证操作

**3.5.2.2 性能**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 性能指标名称 | 优先级 | 性能指标要求 |
| 1 | 数据精度 | A | 数据不出错，保留较高精度 |
| 2 | 灵活性 | C | 能在不同客户端操作 |
| 3 | 响应时间 | A | 能在3秒内响应 |
| 4 | 适应性 | **A** | 能灵活适应不同系统和浏览器 |

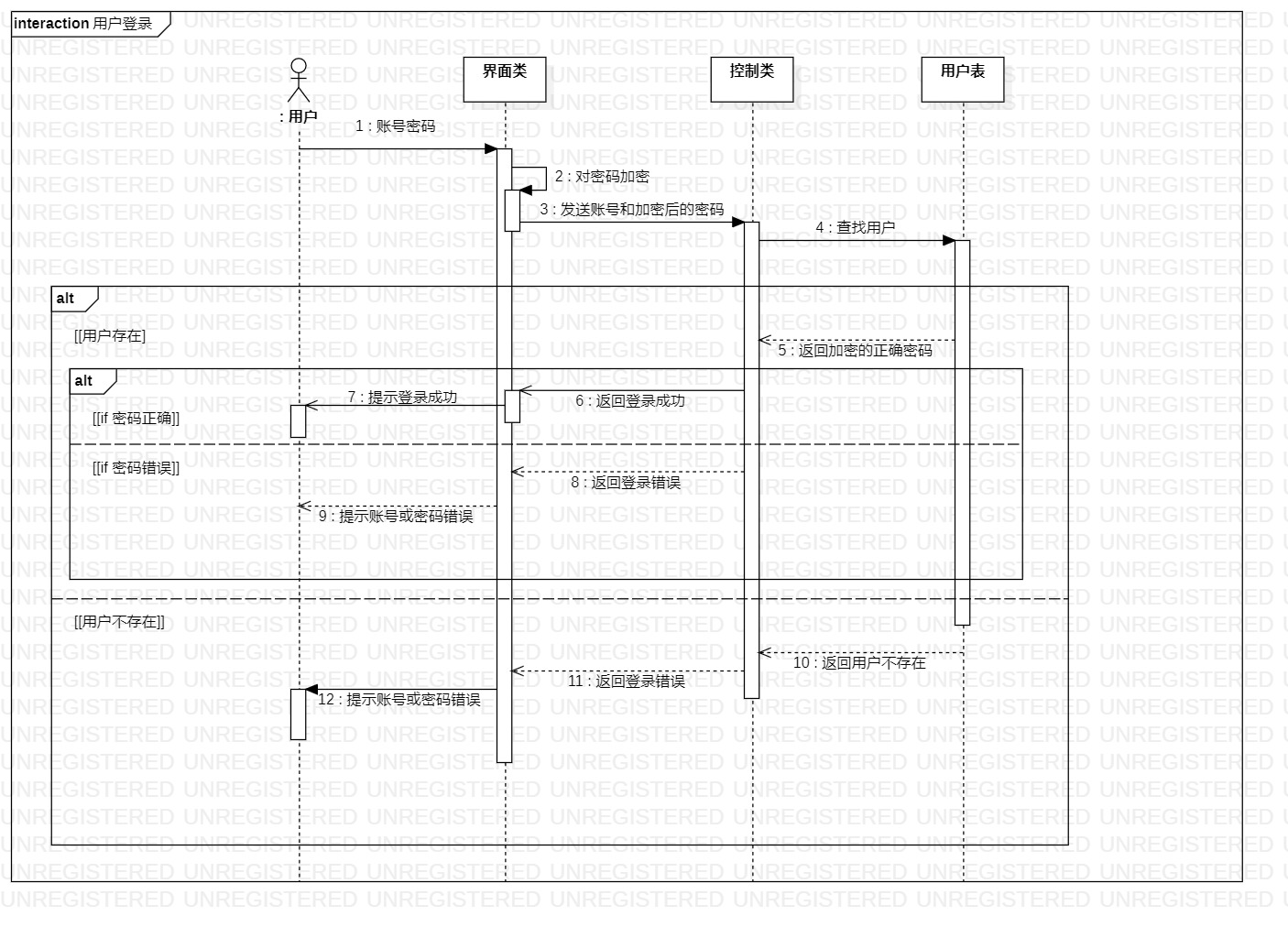
**3.5.2.3 输入数据项及格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据格式 | 数据约束 | 输入源 |
| a | user\_email | 用户邮箱 | String | 128字节 | 登录界面 |
| b | user\_passwd | 用户密码 | String | 128字节 | 登录界面 |
| c | user\_email | 用户邮箱 | String | 128字节 | 注册界面 |
| d | user\_psaawd | 用户密码 | String | 128字节 | 注册界面 |
| e | captcha | 验证码 | String | 8字节 | 注册界面 |
| f | message\_id | 通知的id | Integer | 4字节 | 通知信息概览界面 |
| g | user\_id | 用户id | Integer | 4字节 | 个人信息界面 |
| h | user\_id | 用户id | Integer | 4字节 | 修改个人信息界面 |
| i | user\_sex | 用户性别 | String | 4字节 | 修改个人信息界面 |
| j | user\_introduction | 用户个人介绍 | String | 100字节 | 修改个人信息界面 |
| k | captcha | 验证码 | String | 8字节 | 修改密码界面 |
| l | user\_passwd | 用户密码 | String | 128字节 | 修改密码界面 |
| m | captcha | 验证码 | String | 8字节 | 修改绑定邮箱界面 |
| n | user\_email | 用户邮箱 | String | 128字节 | 修改绑定邮箱界面 |
| o | user\_email | 用户邮箱 | String | 128字节 | 找回密码界面 |
| p | captcha | 验证码 | String | 8字节 | 找回密码界面 |
| q | school | 学校 | String | 64字节 | 认证界面 |
| r | num | 学工号 | String | 16字节 | 认证界面 |
| s | name | 姓名 | String | 32字节 | 认证界面 |

**3.5.2.4 输出数据项及格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据格式 | 数据约束 | 输出源 |
| a | message\_content | 通知内容 | String | 512字节 | 显示通知界面 |
| b | user\_name | 用户名 | String | 32字节 | 显示用户信息界面 |
| c | user\_sex | 性别 | String | 2字节 | 显示用户信息界面 |
| d | user\_introduction | 个人介绍 | String | 100字节 | 显示用户信息界面 |
| e | user\_email | 邮箱 | String | 128字节 | 显示用户信息界面 |
| f | user\_experience | 经验 | Integer | 4字节 | 显示用户信息界面 |

**3.5.2.5 流程处理逻辑**



**3.5.2.6 设计方法（算法）**

1. 用户登录：用户点击登录，控制类根据用户id查询用户表验证账户密码，返回验证结果给界面类，界面类根据结果进行跳转。
2. 用户注册：用户填写相关信息后点击注册，控制类将相关用户信息存入用户表
3. 用户查看通知消息：用户点击通知消息按钮，控制类根据用户id查询通知消息表，将查询结果返回给界面类。
4. 用户查看个人信息：用户点击个人信息，控制类根据用户id查询用户表，返回相关信息给界面类。
5. 用户修改账户信息：用户在个人信息界面修改个人基本信息，个人认证信息，修改密码，控制类通过用户id修改数据库的相关数据。

**3.5.2.7 接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | **请求方式** | **说明** | **请求数据** | **响应数据** |
| user/accountLogin.action | POST | 账号密码登录 | username=?&password=? | 主页 |
| user/emailLogin.action | POST | 邮箱登录 | email=?&verify=? | 主页 |
| user/logon.action | POST | 用户注册 | username=?&password=?&email=? | 登录页面 |
| user/userInfo.action | GET | 查看用户信息 | uid=? | 用户信息页面 |
| user/userNotice.action | POST | 查看用户通知 | uid=? | 用户通知页面 |
| user/modifyInfo.action | POST | 修改用户信息 | uid=?&picture=?&username=?&message=?...... | 修改成功标识 |
| user/authentication.action | POST | 认证 | uid=?&type=?&school=?&number=?& | 认证成功标识 |
| user/resetPassword.action | POST | 重置密码 | uid=?&oldPwd=?&newPwd=? | 修改成功标识 |
| user/rebindEmail.action | POST | 重新绑定邮箱 | uid=?&oldEmail=?&newEmail=?&verify=? | 绑定成功标识 |
| user/retrievePassword/action | POST | 找回密码 | uid=?&email=?&verify=? | 重置密码页面 |

**3.5.2.8 存储分配**

作为核心模块，预计存储量占比达到10%以上。

**3.5.2.9 出错处理**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 错误编号 | 错误名称 | 错误原因 | 错误提示信息 | 处理方式 |
| a | 401 | 验证信息错误 | 账号或密码错误 | 返回相关验证页面要求重新输入 |
| b | 404 | 获取信息失败 | not found | 返回前一个页面，提示继续查找 |
| c | 503 | 服务器错误 | 无法连接到服务器！ | 无 |

**3.5.2.10 数据库结构**

|  |  |  |
| --- | --- | --- |
| 序号 | 表中文名称 | 表英文名称 |
| a | 用户表 | user |
| b | 消息表 | message |

**3.5.2.11 注释设计**

参考《阿里巴巴Java开发手册》进行注释

**3.5.2.12 限制条件**

无

**3.5.2.13 测试计划**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 账户子系统模块（注册、登录、查看通知消息、查看个人信息、修改个人信息、修改绑定邮箱、重置密码、找回密码、修改认证信息） | 界面 | 用户操作 | 界面是否完整 | 界面各个部件是否完整 |
| 注册/登录 | 免密码登录、输入邮箱，点击获取验证码，输入验证码，验证码错误，点击注册/登录按钮 | 按钮相应、验证码检验、消息显示 | 注册/登录失败，提示信息：验证码错误 |
| 免密码登录、输入邮箱，点击获取验证码，输入验证码，验证码正确，点击注册/登录按钮 | 按钮相应、验证码检验、消息显示 | 注册/登录成功，进入首页 |
| 密码登录、输入账号密码，点击注册/登录按钮，账号存在且密码错误 | 按钮相应、用户名和密码检验、消息显示 | 注册/登录失败，提示信息：密码错误 |
| 密码登录、输入账号密码，点击注册/登录按钮，密码正确或账号为新账号 | 按钮相应、用户名和密码检验、消息显示 |  |
| 消息通知 | 点击通知 | 点击响应 | 响应成功，进入相关界面 |
| 个人信息 | - | 界面初始化 | 界面完整 |
| 修改个人信息 | 输入新修改的用户名，性别，自我介绍，点击确认修改，若用户名存在（除自己原先用户名） | 按钮相应、用户名检验、消息显示 | 修改失败，提示信息：用户名已存在 |
| 输入新修改的用户名，性别，自我介绍，点击确认修改，若用户名不存在 | 按钮相应、用户名检验、消息显示 | 修改成功 |
| 修改绑定邮箱 | 输入新邮箱，点击获取验证码，输入验证码，验证码错误，点击绑定 | 按钮相应、验证码检验、消息显示 | 绑定失败，提示信息：验证码错误 |
| 输入新邮箱，点击获取验证码，输入验证码，验证码正确，点击绑定 | 按钮相应、验证码检验、消息显示 | 绑定成功 |
| 重置密码 | 输入原密码、输入新密码、新密码，获取验证码并输入，若验证码错误，点击重置 | 按钮相应、验证码检验、消息显示 | 重置失败，提示信息：验证码错误 |
| 输入原密码、输入新密码、新密码，获取验证码并输入，若原密码错误，点击重置 | 按钮相应、原密码检验、消息显示 | 重置失败，提示信息：原密码错误 |
| 输入原密码、输入新密码、新密码，获取验证码并输入，若验证码两次新密码输入不同，点击重置 | 按钮相应、新密码检验、消息显示 | 重置失败，提示信息：新密码两次输入不同 |
| 输入原密码、输入新密码、新密码，获取验证码并输入，点击重置 | 按钮相应、信息检验、消息显示 | 重置成功 |
| 找回密码 | 输入邮箱，获取验证码，输入新密码和验证码，若验证码错误，点击找回 | 按钮相应、验证码检验、消息显示 | 找回失败，提示信息：验证码错误 |
| 输入邮箱，获取验证码，输入新密码和验证码，点击找回 | 按钮相应、验证码检验、消息显示 | 找回成功 |
| 认证 | 选择学校，输入姓名，学号，点击认证 | 按钮相应、信息检验、消息显示 | 认证成功 |

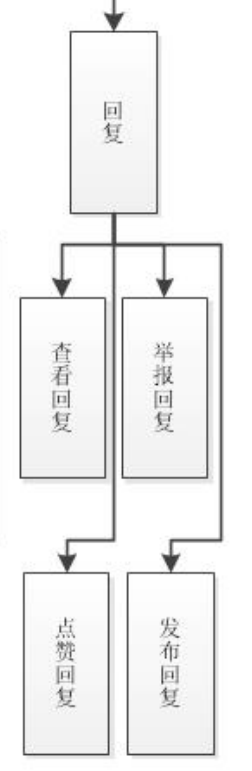
**3.5.2.14 尚未解决的问题**

无

**3.6. 回复子系统功能模块**

**3.6.1 功能模块描述及流程**

回复的功能模块由查看回复、举报回复、点赞回复、发布回复等主要子功能模块组成。如下图所示：



**3.6.2 回复模块设计说明**

**3.6.2.1 程序模块及功能描述**

1. 发布回复：

用户在评论区点击回复按钮后进入评论回复界面。用户填写回复内容后点击发布按钮，将回复存储到回复表中存储，并根据评论id获取评论的评论者id，构建消息对象存储到消息表

1. 查看回复：

a. 用户在回答详情界面点击评论按钮时，将根据回答id，获取回答下的评论，然后根据评论id获取每个评论下的前几条回复，发送给界面

b. 用户在评论区界面点击一条评论下的更多回复时，控制类根据评论id，获得评论下的所有回复，发送给界面

1. 点赞回复：

用户在评论区或评论详情对回复点击点赞按钮时，将根据更新回复表中的对应回复的点赞数，新增回复点赞记录到回复点赞记录表，同时新增消息到消息表

1. 举报回复：

用户在评论详情界面对回复点击举报按钮，将举报内容存储到举报回复表中

**3.6.2.2 性能**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 性能指标名称 | 优先级 | 性能指标要求 |
| 1 | 数据精度 | A | 数据不出错，保留较高精度 |
| 2 | 灵活性 | C | 能在不同客户端操作 |
| 3 | 响应时间 | A | 能在3秒内响应 |
| 4 | 适应性 | **A** | 能灵活适应不同系统和浏览器 |

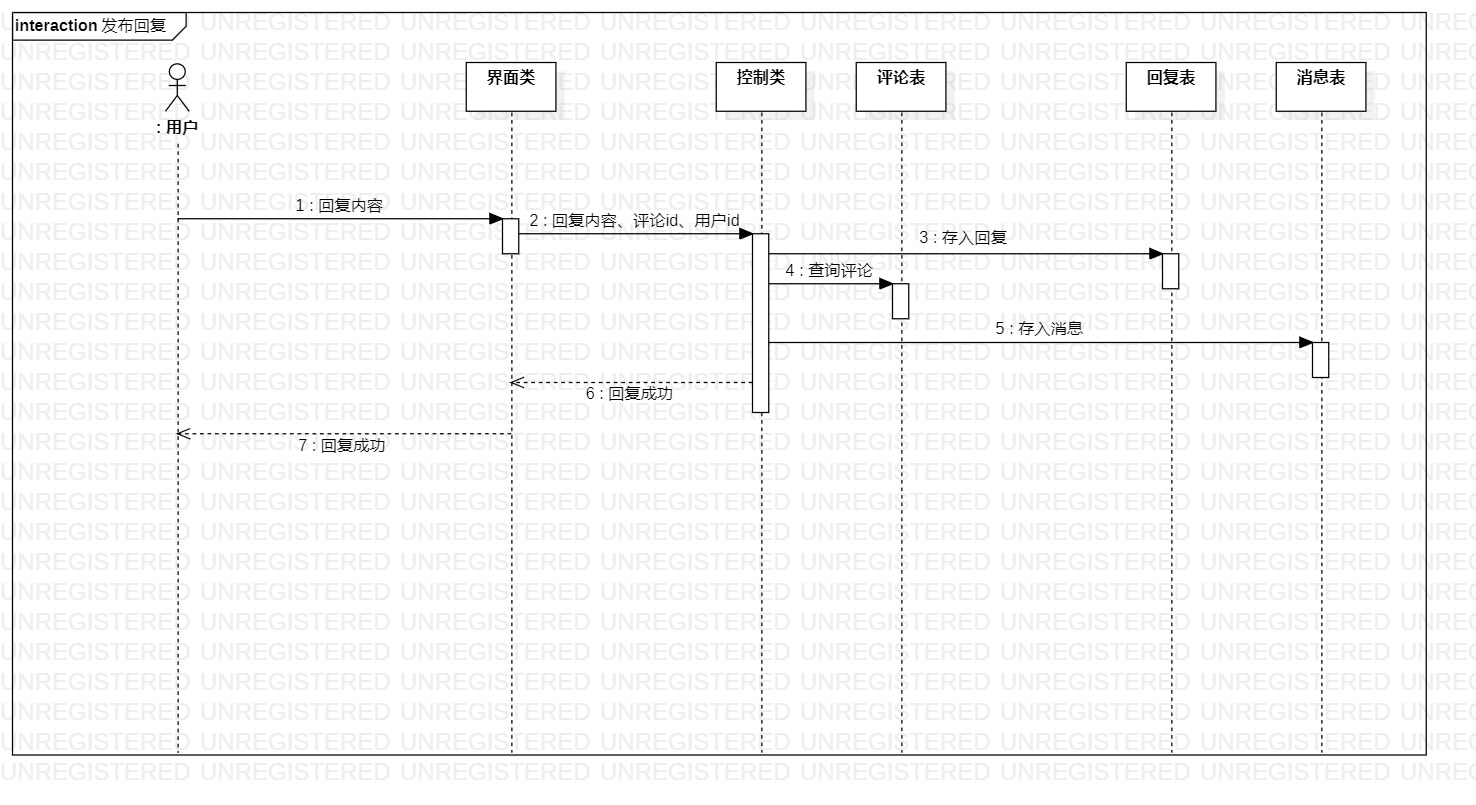
**3.6.2.3 输入数据项及格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据格式 | 数据约束 | 输入源 |
| a | reply\_content | 回复内容 | String | 1000字节 | 发布回复界面 |
| b | user\_id | 回复者id | Integer | 4字节 | 发布回复界面 |
| c | comment\_id | 评论id | Integer | 4字节 | 发布回复界面/评论区界面 |
| d | answer | 回答id | Integer | 4字节 | 回复详情界面 |
| e | reply\_id | 回复id | Integer | 4字节 | 评论区界面/评论详情界面 |
| f | user\_id | 点赞者id | Integer | 4字节 | 评论区界面/评论详情界面 |
| g | reply\_id | 被举报回复id | Integer | 4字节 | 回复举报界面 |
| h | user\_id | 举报者id | Integer | 4字节 | 回复举报界面 |
| i | report\_description | 举报描述 | String | 256字节 | 回复举报界面 |
| j | report\_reason\_id | 举报原因类别rid | Integer | 4字节 | 回复举报界面 |

**3.6.2.4 输出数据项及格式**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据元素/项标识 | 数据元素/项名称 | 数据类型 | 数据约束 | 输入源 |
| a | reply\_content | 回复内容 | String | 1000字节 | 回复界面 |
| b | message\_content | 举报后系统回复的消息内容 | String | 512字节 | 回复举报界面 |

**3.6.2.5 流程处理逻辑**



**3.6.2.6 设计方法（算法）**

1. 发布回复:

控制类根据获取的回复内容、评论id、回复者id构建回复对象，存储到回复表，再根据评论id从评论表中获取评论者用户id，构建消息对象，存储到消息表

1. 查看回复：

控制类根据获取的评论id，从回复表中获取回复数据，构建回复，发送到界面类

1. 点赞回复：

控制类根据获取的回复id、点赞者id，构建回复点赞记录对象，存储到回复点赞表，并根据回复id，从数据库中获取回复对象，更新回复点赞数，然后根据回复者id，构建消息对象，存储到消息表

1. 举报回复：

控制类根据获取的回复id、举报者id、举报原因类型rid、举报描述，构建举报回复记录对象，存储到举报回复表

**3.6.2.7 接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | **请求方式** | **说明** | **请求数据** | **响应数据** |
| reply/reply.action | POST | 回复 | rid=?&uid=?&reply=? | 回复成功标识 |
| reply/likeReply.action | POST | 点赞回复 | rid=?&uid=? | 点赞成功标识 |
| reply/reportReply.action | POST | 举报回复 | rid=?&uid=?&report=? | 举报成功标识 |

**3.6.2.8 存储分配**

作为核心模块，预计存储量占比达到20%以上。

**3.6.2.9 出错处理**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 错误编号 | 错误名称 | 错误原因 | 错误提示信息 | 处理方式 |
| a | 404 | 获取信息失败 | not found | 返回前一个页面，提示继续查找 |
| b | 503 | 服务器错误 | 无法连接到服务器！ | 无 |

**3.6.2.10 数据库结构**

|  |  |  |
| --- | --- | --- |
| 序号 | 表中文名称 | 表英文名称 |
| a | 回复表 | reply |
| b | 举报记录表 | Report |
| c | 评论表 | comment |
| d | 消息表 | message |
| e | 回复点赞记录表 | approvalReply |

**3.6.2.11 注释设计**

参考《阿里巴巴Java开发手册》进行注释

**3.6.2.12 限制条件**

无

**3.6.2.13 测试计划**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块 | 界面 | 用户行为 | 响应 | 结果 |
| 回复子系统功能模块（查看回复、举报回复、点赞回复、发布回复） | 评论界面 | 点击查看回复按钮 | 是否响应 | 界面是否完整 |
| 查看回复 | 点击举报按钮 | 是否弹出举报页面 | 界面是否完整 |
| 点击点赞按钮 | 按钮是否响应 | 点赞图标是否点亮，点赞数是否增加 |
| 点击评论按钮 | 是否响应 | 出现回复输入界面 |
| 举报回复 | （未）选择违规类型，未输入详细说明，点击举报按钮 | 是否举报成功 | 举报失败，提示：请输入详细信息 |
| （未）选择违规类型，输入详细说明，点击举报按钮 | 是否举报成功 | 举报成功 |
| 发布回复 | 输入字数超过限制，点击发布 | 按钮是否响应、字数检验 | 发布失败，提示：字数应小于100 |
| 输入字数为超过显示，点击发布 | 按钮是否响应、字数检验 | 发布成功 |

**3.6.2.14 尚未解决的问题**

无

# 接口设计

**4.1与外系统的接口**

前端使用CDN运行时加载外部CSS、JS库，如下表所示：

|  |  |  |
| --- | --- | --- |
| **第三方库** | **版本** | **URL** |
| bootstrap.min.css | 4.4.1 | https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css |
| bootstrap.min.js | 4.4.1 | https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/js/bootstrap.min.js |
| proper.min.js | 1.16.0 | https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js |
| jquery.min.js | 3.4.1 | https://lib.baomitu.com/jquery/3.4.1/jquery.min.js |
| vue.min.js | 2.6.10 | https://lib.baomitu.com/vue/2.6.10/vue.min.js |
| font-awesome | 5.11.2 | https://lib.baomitu.com/font-awesome/5.11.2/css/all.min.css |

后端使用JavaMail连接外部邮箱服务器实现邮件发送功能：

|  |  |
| --- | --- |
| 邮箱地址 |  |
| 邮箱授权码 |  |
| 所属公司 |  |

使用学校提供的接口进行身份认证：

|  |  |  |
| --- | --- | --- |
| **学校** | **远程API/数据库** | **验证信息** |
| 福州大学 |  |  |

**4.2 内部接口**

|  |  |
| --- | --- |
| **接口类型** | **描述** |
| 数据库接口 | 使用JDBC连接数据库 |
| 前后端接口 | 使用AJAX进行前后端交互 |

# 系统安全和权限设计

**5.1 系统安全**

1. 对用户密码进行hash加密。
2. 在cookie中使用密文来保存用户信息。
3. 使用阿里云ECS安全策略防止网络攻击。
4. 对输入数据进行过滤，确保输入符合我们的预期，以此防止XSS注入。
5. 使用Hibernate持久化框架防止SQL注入。
6. 多次登录失败触发验证码机制，防止暴力破解。

**5.1 权限设计**

本系统使用Struts的拦截器进行权限控制。

**后台：**

a. 没有登录无法使用任何功能。

**前台：**

1. 没有登录只能进行浏览、搜索，无法进行提问、回答、评论、点赞以及个人系统模块中的任何功能。
2. 被封禁的用户无法进行提问、回答、评论、点赞等功能。

只有题主才能对回答进行采纳、置顶。