在线聊天室项目

面向对象设计文档

李 梓瑞

2022

目录

[一. 用例精化 3](#_Toc123139096)

[1.1 提取边界类，实体类，控制类 3](#_Toc123139097)

[1.1.1 边界类 3](#_Toc123139098)

[1.1.2 实体类 4](#_Toc123139099)

[1.1.3 控制类 4](#_Toc123139100)

[1.2 构造交互图 5](#_Toc123139101)

[1.2.1 登录 5](#_Toc123139102)

[1.2.2 注册 6](#_Toc123139103)

[1.2.3 加入聊天室 6](#_Toc123139104)

[1.2.4 创建聊天室 7](#_Toc123139105)

[1.2.5 发送消息 7](#_Toc123139106)

[1.2.6 获取历史消息 8](#_Toc123139107)

[1.2.7 创建通行码 8](#_Toc123139108)

[1.2.8 发送文件 9](#_Toc123139109)

[1.2.9 封禁移出用户 9](#_Toc123139110)

[1.2.10 邀请用户 10](#_Toc123139111)

[1.2.11 获取聊天室邀请码 10](#_Toc123139112)

[1.3 精化类图 11](#_Toc123139113)

[二. 软件体系结构 12](#_Toc123139114)

[三. 构件设计 13](#_Toc123139115)

[3.1 前端-后端通讯接口 13](#_Toc123139116)

[3.1.1 通用接口 13](#_Toc123139117)

[3.1.2 用户登录 15](#_Toc123139118)

[3.1.3 用户注册 16](#_Toc123139119)

[3.1.4 用户是否存在 16](#_Toc123139120)

[3.1.5 获取用户信息 16](#_Toc123139121)

[3.1.6 修改用户自身信息 17](#_Toc123139122)

[3.1.7 校验创建聊天室通行码合法性 17](#_Toc123139123)

[3.1.8 创建聊天室 18](#_Toc123139124)

[3.1.9 获取聊天室邀请码 18](#_Toc123139125)

[3.1.10 加入聊天室 19](#_Toc123139126)

[3.1.11 邀请用户进入聊天室 19](#_Toc123139127)

[3.1.12 离开聊天室 20](#_Toc123139128)

[3.1.13 获取聊天室在线用户 20](#_Toc123139129)

[3.1.14 向聊天室发送消息 20](#_Toc123139130)

[3.1.15 获取聊天室历史消息 21](#_Toc123139131)

[3.1.16 获取用户自身信息 21](#_Toc123139132)

[3.2 在线用户维护 22](#_Toc123139133)

[3.3 断线重连 22](#_Toc123139134)

[3.4 防御性设计 22](#_Toc123139135)

[3.5 前端状态管理 23](#_Toc123139136)

[四. 用户界面设计 24](#_Toc123139137)

[4.1登录 24](#_Toc123139138)

[4.2注册 24](#_Toc123139139)

[4.3选择聊天室 25](#_Toc123139140)

[4.4聊天室主体 25](#_Toc123139141)

[4.5创建聊天室 26](#_Toc123139142)

[4.6 修改个人信息 26](#_Toc123139143)

[4.7邀请好友 27](#_Toc123139144)

[4.8聊天室邀请码 27](#_Toc123139145)

[五. 数据持久化设计 28](#_Toc123139146)

[5.1持久化技术选择 28](#_Toc123139147)

[5.2定义数据格式 28](#_Toc123139148)

[5.2.1 User表 28](#_Toc123139149)

[5.2.2 Chatroom表 28](#_Toc123139150)

[5.2.3 Message表 28](#_Toc123139151)

[5.2.4 Ban Record表 29](#_Toc123139152)

[5.2.5 Passport表 29](#_Toc123139153)

[5.3数据存取方式 29](#_Toc123139154)

# 一. 用例精化

## 1.1 提取边界类，实体类，控制类

### 1.1.1 边界类

边界类描述了用例中系统与外界环境交互的方式，在每个用例和与之关联的参与者之间都会存在边界类保证他们之间的数据流动。边界类包含界面控制，外部接口以及环境隔离三个方面。

#### 1.1.1.1 界面控制

本项目的图像用户交互界面包含的主界面以及弹出界面如表一和表二所示，主界面是显示在浏览器主体的主要界面，而弹出界面则有主界面衍生，负责完成与主界面功能相关的部分简单用例。

表格 1 主界面

|  |  |  |
| --- | --- | --- |
| **界面名称** | **界面描述** | **跳转操作** |
| 登录界面 | 用户登录账号进入系统的入口 | 1.选择注册按钮：  注册界面  2.登录成功：  聊天室选择界面 |
| 注册界面 | 用户获取账号的界面 | 1.返回：  登录界面  2.注册成功：  登录界面 |
| 聊天室选择界面 | 用户选择自身聊天室，加入聊天室等功能的界面 | 1.退出登录：  登录界面  2.选择聊天室：  聊天室界面 |
| 聊天室界面 | 用户与聊天室中其他用户进行聊天的界面 | 1.退出登录：  登录界面  2.返回主页：  聊天室选择界面 |

表格 2 弹出界面

|  |  |  |
| --- | --- | --- |
| **界面名称** | **界面描述** | **所属主界面** |
| 创建聊天室界面 | 用户使用管理员发放的通行码创建聊天室的弹窗界面 | 聊天室选择界面 |
| 修改个人信息界面 | 用户修改账号用户名，密码和头像的弹窗界面 | 聊天室选择界面或聊天室界面 |
| 邀请界面 | 用户使用用户名搜索邀请其他用户的弹窗界面 | 聊天室界面 |
| 邀请码界面 | 用户名获取当前聊天室邀请码的弹窗界面 | 聊天室界面 |
| 重复登录界面 | 用户账号在被重复登录后弹出的提示界面 | 聊天室选择界面或聊天室界面 |

#### 1.1.1.2 外部接口

**用户-系统接口：**

用户通过浏览器访问系统并采用如下设备进行输入：键盘，鼠标，触屏

同时，用户通过桌面或者移动显示屏，接收来自系统的信息

**前端-后端接口：**

前后端之间的通讯结合HTTP请求以及WebSocket（使用STOMP + SockJS的实现）完成

#### 1.1.1.3 环境隔离

本项目中不存在环境隔离

### 1.1.2 实体类

实体类表示目标软件系统中具有持久意义的信息项及其操作，此次项目中包含的实体类如下：

表格 3 实体类

|  |  |
| --- | --- |
| **实体类** | **类描述** |
| 用户 | 保存用户账号信息的类 |
| 聊天室 | 保存聊天室基本信息的类 |
| 消息 | 保存聊天室中消息信息的类 |
| 可用通行码 | 保存被管理员创建的通信码信息的类 |
| 封禁记录 | 保存用户被聊天室封禁的记录的类 |

### 1.1.3 控制类

控制类是控制用例完成功能的类，负责协调不同类的工作以满足用例需求。总体而言，本次项目中将包含如下几个主控制类，作为用例功能的入口，进行任务分解并调用各类服务完成工作。

表格 4 主控制类

|  |  |
| --- | --- |
| 控制类 | 描述 |
| 用户控制器 | 负责处理与用户账号相关的用例，如注册，登录以及账号信息修改 |
| 聊天室控制器 | 负责处理与聊天室相关的用例，如创建，加入，离开聊天室以及用户封禁 |
| 通行码管理器 | 负责处理通行码的校验以及使用 |
| 通行码生成器 | 负责通行码的生成 |
| 聊天控制器 | 负责处理聊天室中用户消息的接收与广播 |
| Token管理器 | 管理所有JWT的生成，校验以及解析 |

## 1.2 构造交互图

使用UML顺序图展现精化后的用例实现流程

### 1.2.1 登录



### 1.2.2 注册



### 1.2.3 加入聊天室



### 1.2.4 创建聊天室



### 1.2.5 发送消息



### 1.2.6 获取历史消息



### 1.2.7 创建通行码



### 1.2.8 发送文件



### 1.2.9 封禁移出用户



### 1.2.10 邀请用户



### 1.2.11 获取聊天室邀请码



## 1.3 精化类图

绿色：边界类

深蓝色：控制类

淡蓝色：实体类

灰色：接口



# 二. 软件体系结构

本次项目体系结构总体上采用了前后端分离的“前端-后端-数据层”三层BS架构，架构的分层示意图如下：



# 三. 构件设计

## 3.1 前端-后端通讯接口

### 3.1.1 通用接口

#### 3.1.1.1 HTTP应答数据格式

后端HTTP进行HTTP应答的返回结果包含如下三个字段

|  |  |
| --- | --- |
| **字段名** | **描述** |
| code | 错误码 |
| msg | 返回描述信息 |
| data | 返回数据 |

其中data字段的结构根据请求的服务不同而不同，具体格式见后文特定服务的规定

#### 3.1.1.2 WebSocket数据规定

本项目使用STOMP实现WebSocket长连接，STMOP协议中的连接，订阅地址以及消息格式如下：

##### 3.1.1.2.1 STOMP连接地址

|  |
| --- |
| $base\_url$/chat?token=$user\_token$ |

##### 3.1.1.2.2 STOMP订阅地址

订阅特定聊天室以接收消息的地址

|  |
| --- |
| $base\_url$/chat/chatroom/$chatroom\_id$ |

##### 3.1.1.2.3 STOMP消息格式

后端与前端使用STOMP进行通讯的消息格式规定如下

|  |  |
| --- | --- |
| **字段名** | **描述** |
| type | 消息类型 |
| data | 消息数据体 |

data字段的结构根据消息类型不同而不同，具体格式见后文特定服务的规定

##### 3.1.2.2.4 STOMP消息类型

|  |  |  |
| --- | --- | --- |
| **消息类型** | **描述** | **data字段格式** |
| 1 | 用户发送聊天消息 | MessageDTO见3.1.1.4 |
| 2 | 服务端广播聊天消息 | MessageDTO见3.1.1.4 |
| 3 | 账号重复登陆警告 | 无 |

#### 3.1.1.3错误码

本次项目中统一规定的错误码以及其含义如下

|  |  |
| --- | --- |
| **错误码** | **错误含义** |
| 0 | 无错误 |
| 1 | Token不存在 |
| 2 | 用户不存在 |
| 3 | 密码错误 |
| 4 | 用户重复加入聊天室 |
| 5 | 聊天室不存在 |
| 6 | 无效通行码 |
| 7 | Token过期 |
| 8 | 用户名重复 |
| 9 | 聊天室已满 |
| 10 | 用户不在聊天室中 |
| 11 | 用户被聊天室封禁 |
| -1 | 未知错误 |

#### 3.1.1.4 DTO

##### 3.1.1.4.1 UserDTO用户数据通讯接口

|  |  |
| --- | --- |
| **字段名** | **字段含义** |
| userId | 用户标识符 |
| username | 用户名 |
| password | 用户密码 |
| avatar | 用户头像 |

##### 3.1.1.4.2 ChatroomDTO聊天室数据通讯接口

|  |  |
| --- | --- |
| **字段名** | **字段含义** |
| id | 聊天室标识符 |
| title | 聊天室标题 |
| icon | 聊天室图标 |
| comment | 聊天室简介 |
| max\_user | 聊天室最大用户数 |
| admin | 聊天室管理员 |
| total\_user | 聊天室在线用户数 |
| onlines | 聊天室在线用户 |

##### 3.1.1.4.3 MessageDTO消息数据通讯接口

|  |  |
| --- | --- |
| **字段名** | **字段含义** |
| messageId | 消息标识符 |
| chatroomId | 聊天室标识符 |
| type | 消息类型 |
| content | 消息内容 |
| senderId | 发送者标识符 |
| timestamp | 发送时间 |

### 3.1.2 用户登录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/login | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| username | | 用户名 | |
| password | | 密码 | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
| uid | | 用户id | |

### 3.1.3 用户注册

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/login | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| username | | 用户名 | |
| password | | 密码 | |
| avatar | | 用户头像 | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| 无 | |  | |

### 3.1.4 用户是否存在

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/exist | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| username | | 用户名 | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| result | | 是否找到（true或false） | |
| id | | 用户名 | |
| avatar | | 用户头像 | |

### 3.1.5 获取用户信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | GET |
| **HTTP请求地址** | /api/user/userInfo | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| id | | 用户id | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| id | | 用户id | |
| username | | 用户名 | |
| avatar | | 用户头像 | |

### 3.1.6 修改用户自身信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/user/edit | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| username | | 用户名 | |
| password | | 密码 | |
| avatar | | 用户头像 | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| 无 | |  | |

### 3.1.7 校验创建聊天室通行码合法性

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chatroom/verifyPassport | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| token | | 通行码 | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| max\_user | | 最大用户数上限（通行码不合法返回错误码非0） | |

### 3.1.8 创建聊天室

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chatroom/createChatroom | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| passport | | 创建聊天室通行码 | |
| title | | 聊天室标题 | |
| comment | | 聊天室简介 | |
| max\_user | | 聊天室最大用户数 | |
| icon | | 聊天室图标 | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| 无 | |  | |

### 3.1.9 获取聊天室邀请码

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chatroom/inviteToken | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| chatroomId | | 聊天室id | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| 无 | |  | |

### 3.1.10 加入聊天室

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chatroom/join | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| token | | 邀请码 | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| 无 | |  | |

### 3.1.11 邀请用户进入聊天室

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chatroom/invite | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| userId | | 被邀请用户id | |
| chatroomId | | 聊天室id | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| 无 | |  | |

### 3.1.12 离开聊天室

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chatroom/join | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| chatroomId | | 聊天室id | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| 无 | |  | |

### 3.1.13 获取聊天室在线用户

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chatroom/onlines | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| chatroomId | | 聊天室id | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| onlines | | UserDTO数组 | |

### 3.1.14 向聊天室发送消息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chat/send | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | 数据格式同[MessageDTO](#_3.1.1.4.3_MessageDTO消息数据通讯接口) | | | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| 无 | |  | |

### 3.1.15 获取聊天室历史消息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | POST |
| **HTTP请求地址** | /api/chat/get | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| startId | | 起始消息id | |
| chatroomId | | 聊天室id | |
| cnt | | 请求数量 | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| count | | 获取到的消息数量 | |
| msg | | MessageDTO数组，获取到的消息 | |

### 3.1.16 获取用户自身信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **通讯方式** | HTTP请求 | **HTTP方法** | | GET |
| **HTTP请求地址** | /api/user/selfInfo | | | |
| **HTTP路径参数** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP请求头** | **字段名** | | **含义** | |
| token | | 用户登录凭证 | |
|  | | | | |
| **HTTP请求体** | **字段名** | | **含义** | |
| 无 | |  | |
|  | | | | |
| **HTTP应答data** | **字段名** | | **含义** | |
| id | | 用户id | |
| username | | 用户名 | |
| avatar | | 用户头像 | |
| chatrooms | | ChatroomDTO数组，用户已经加入的聊天室 | |

## 3.2 在线用户维护

在后端使用STOMP提供的Listener监听前端的STOMP连接和断开连接请求，在监听到连接时，保存连接的用户以及其session的对应关系于一个特定的数据结构中，在用户断开STOMP连接时删除对应关系的存在。

利用该方法同时能够实现重复登录的检测以及强制下线：当用户登录时如果已经存在用户和session的对应关系，则与先前的session断开连接，保存最新的连接信息。

统计聊天室在线用户时，可以对聊天室的所属用户进行过滤，只保留存在用户-session对应关系的用户（表示他们在线），进而得到某个聊天室的在线用户。

## 3.3 断线重连

前端使用STOMP与后端建议WebSocket连接并订阅特定服务，通常情况下用户需要订阅全部的自己已经加入的聊天室服务。前端应当保留聊天室与订阅的对应关系，并定时检查STOMP连接是否激活，如果STOMP连接发生断开，应该定时重试，并且每次连接重试成功时都应当重新订阅全部聊天室。同时当聊天室订阅出现错误时，也应当进行重新订阅（使用异常处理的回调函数），保证用户正常使用。

## 3.4 防御性设计

系统应尽可能考虑到可能发生的错误，并对错误进行捕获相应。同时，对于由用户引发或者对用户使用具有影响的错误，应当在视觉上给予用户充足的反馈。

此次项目内，与功能相关的错误，通过前后端共同遵守的[错误码](#_3.1.1.3错误码)规范进行定义。

对于前端：

任何的请求都可能产生错误，前端应当结合请求所对应的功能，考虑所有可能发生的错误，并当对应的错误发生时，给予用户视觉上的提示。

对于后端：

借助Spring的Controller和ExceptionHandler，通过统一的错误控制器来进行错误反馈。所有的项目定义异常应当继承统一的BubblesException接口，该接口应当定义一个获取错误码和错误信息的接口方法，不同的异常应当重写该方法返回自身对应的错误码。在统一的错误控制器中，仅需处理BubblesException，调用获取错误码的方法并组装应答消息。

## 3.5 前端状态管理

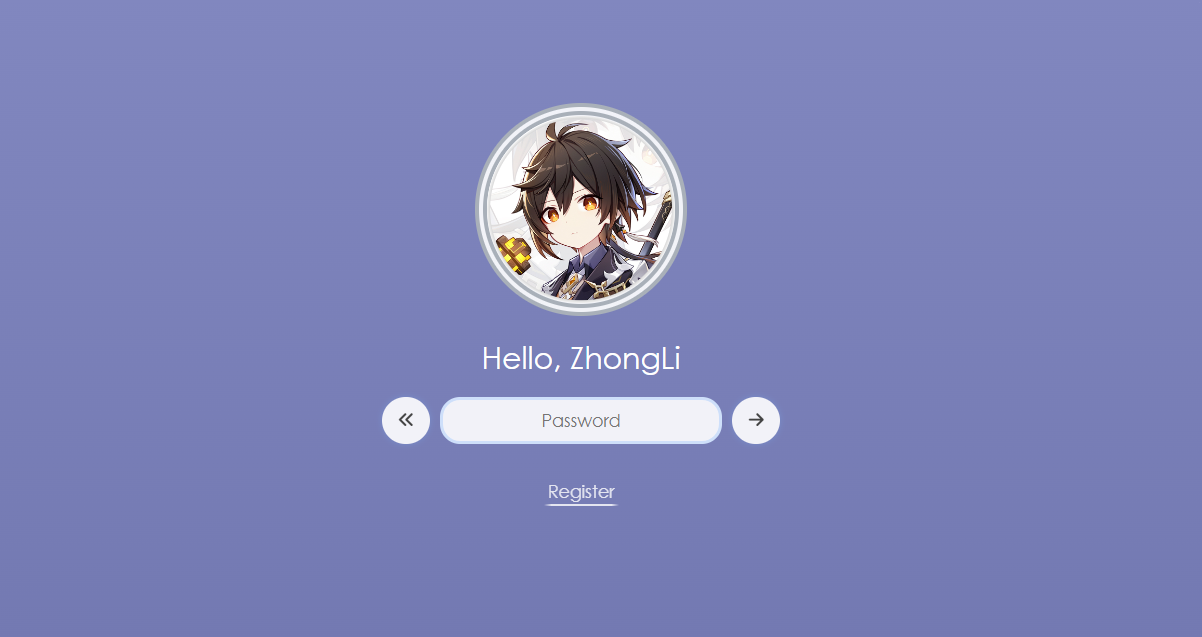
本次项目拟使用Vuex负责前端用户图形界面的全局状态管理。采用模块化的设计，将全局状态分割为以下几个状态模块：

|  |  |
| --- | --- |
| **模块** | **含义** |
| chatroom | 用户所处聊天室的全局状态 |
| localuser | 当前用户的自身状态 |
| userInfo | 管理用户信息的集合 |
| socket | 管理WebSocket连接 |
| error | 管理错误处理 |
| global | 全局常量 |
| settings | 聊天室设置 |
| tx | 事务管理 |

所有涉及多个模块协同操作的mutation和action应当位于顶层模块中，其他操作位于各个专门的模块。规定只有action中可以发起异步请求，mutation中仅负责数据变更。

# 四. 用户界面设计

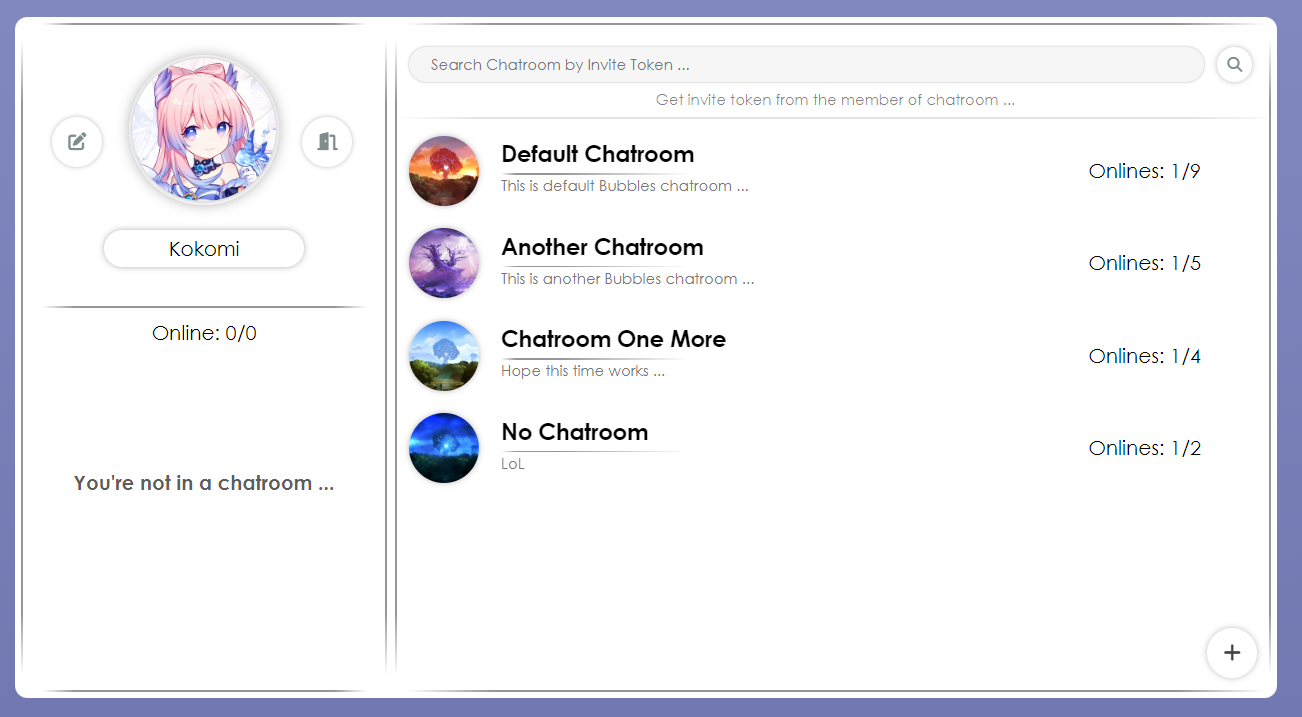
## 4.1登录



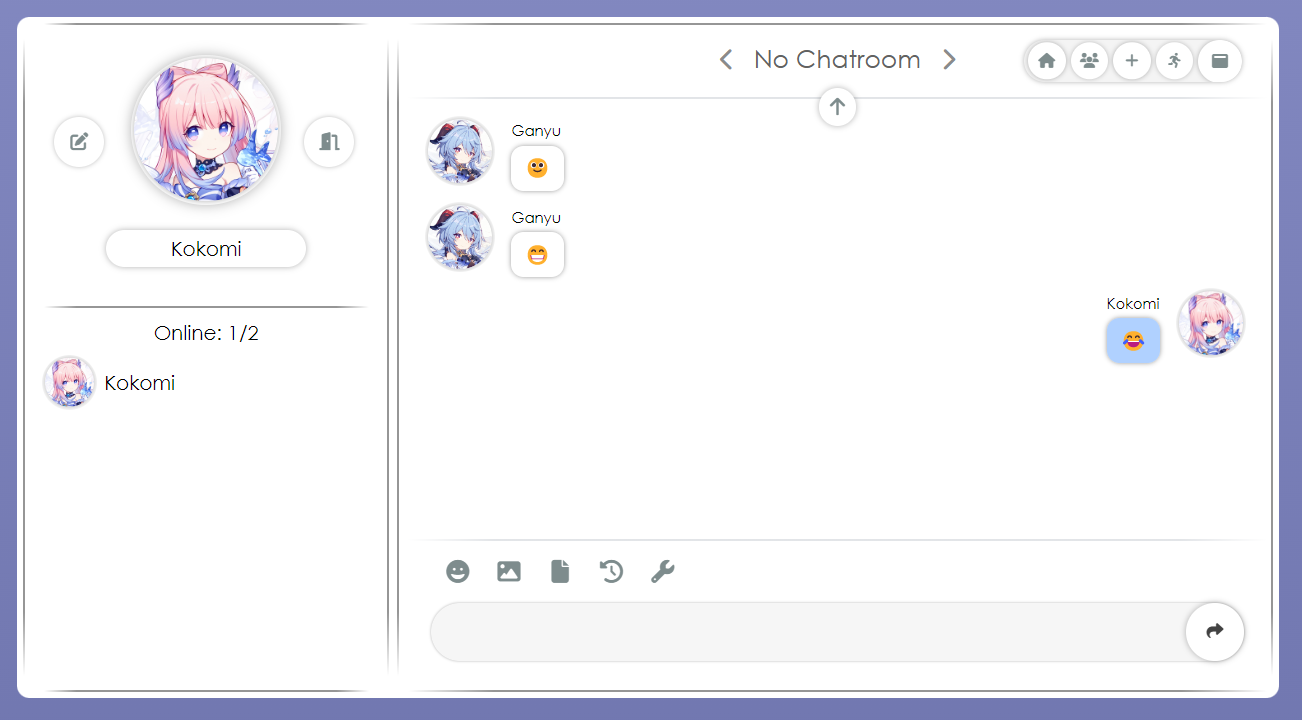
## 4.2注册



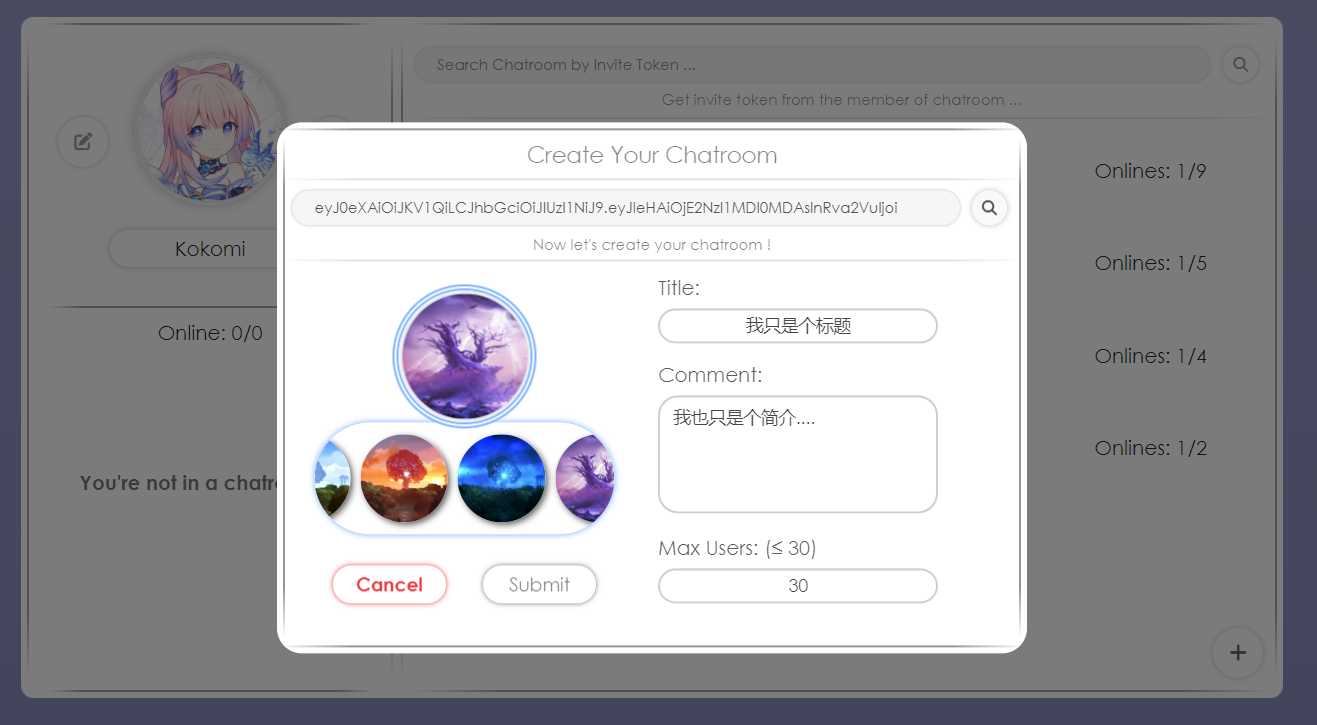
## 4.3选择聊天室



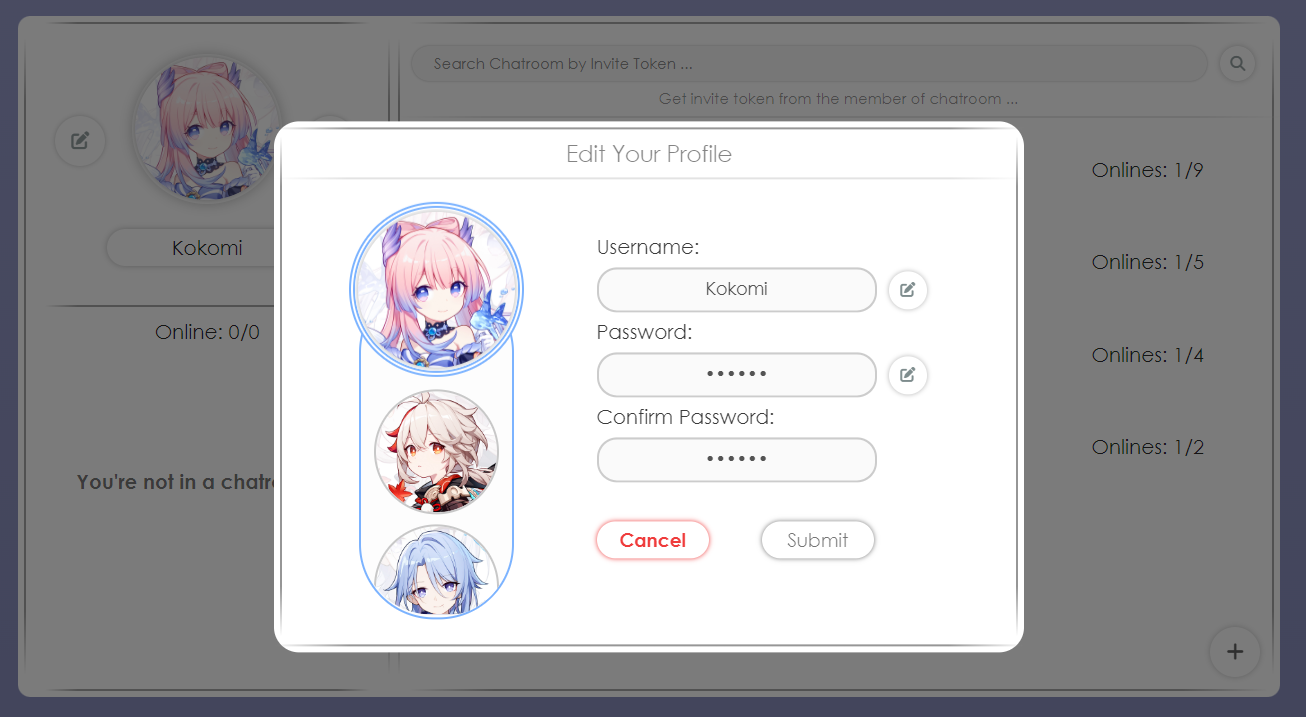
## 4.4聊天室主体



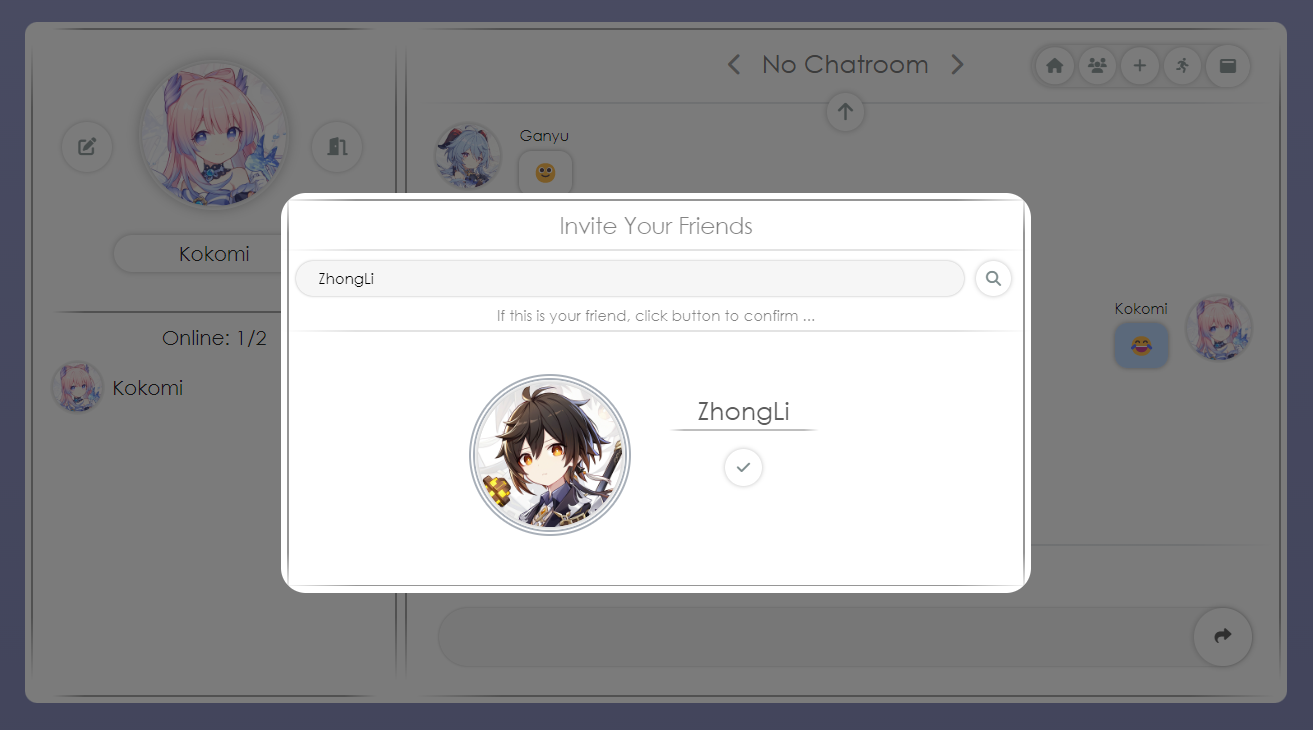
## 4.5创建聊天室



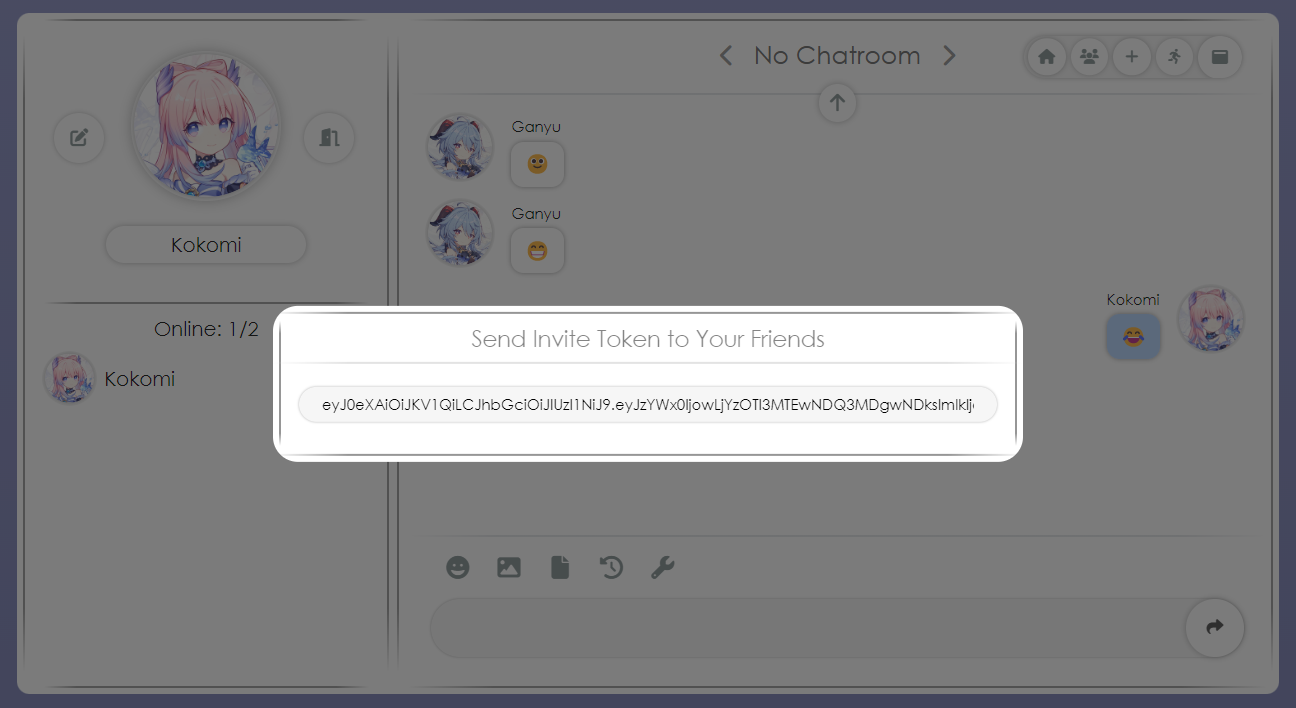
## 4.6 修改个人信息



## 4.7邀请好友



## 4.8聊天室邀请码



# 五. 数据持久化设计

## 5.1持久化技术选择

本次项目使用MySQL作为数据库服务提供平台，在此基础上利用SpringBoot JPA的支持，完成数据存取接口的实现。

## 5.2定义数据格式

本次项目包含如下的数据表，其表字段描述以及含义在下面列出

### 5.2.1 User表

存储了用户账号信息的表

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述** |
| user\_id | int | 用户唯一标识符 |
| avatar | int | 用户头像标号 |
| password | varchar (255) | 用户密码的md5码 |
| username | varchar (255) | 用户名 |

### 5.2.2 Chatroom表

存储了聊天室信息的表

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述** |
| chatroom\_id | int | 聊天室唯一标识符 |
| comment | varchar (255) | 聊天室简介 |
| title | varchar (255) | 聊天室标题 |
| admin | int | 聊天室创建者id |
| icon | int | 聊天室图表标号 |
| max\_user | int | 聊天室最大用户数 |

### 5.2.3 Message表

存储了用户聊天消息信息的表

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述** |
| message\_id | int | 消息唯一标识符 |
| chatroom\_id | int | 消息所属聊天室id |
| sender\_id | int | 消息发送者用户id |
| content | varchar (255) | 消息内容 |
| type | int | 消息类型 |
| timestamp | bigint | 发送时间戳 |

### 5.2.4 Ban Record表

存储了用户封禁记录的表

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述** |
| ban\_id | int | 封禁记录唯一标识符 |
| user\_id | int | 被封禁用户的id |
| chatroom\_id | int | 被封禁聊天室id |
| timestamp | bigint | 封禁时间 |

### 5.2.5 Passport表

存储了可用通行码的表

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述** |
| passport\_id | int | 通行码唯一标识符 |
| token | longtext | 通行码内容 |
| type | int | 通行码类型 |

## 5.3数据存取方式

对于每一张表，都会创建对应的JpaRepository数据存取接口，在接口中根据业务需求定义必要的读取，存储，修改与删除操作。上层业务层通过注入存取接口实例，并调用数据存取接口中定义的操作完成数据存取。