Chat App Design Doc

Team: Bullet(Team J)

Member:

Siyu Xie(team Lead)

He Wang (tech lead)

Huijia Zhu (doc lead)

Pengyu Zhang (dev)

Su Zhang (dev)

Suyue Zhang (dev)

Agenda

- Use case
- Design Decision
- Interface/Class Description
- GUI Introduction

Use case

- User inputs basic information for registration.
- User checks his/her profile.
- User creates a chat room.
- User applys for joinning a chat room.
- Admin approves user application.
- Admin invite a user to join his chat room.
- User sends, edit, recall, delete message to particular receiver(s).
- User reports other users.
- Admin deletes message.
- Admin bans user.
- User exits a chat room.
- User closes connection.

Design Decision

- 1. In this project, We use the following design patterns in our implementation: command, singleton.
- 2. We used websockets to communicate between the frontend and backend.
- 3. We called websocket messages sent from frontend to backend as request, websocket message sent from backend to frontend as response. The format of websocket messages is:

```
{
    "response": "...",
    "data":
    {
        .....
}
```

Interface/Class Description

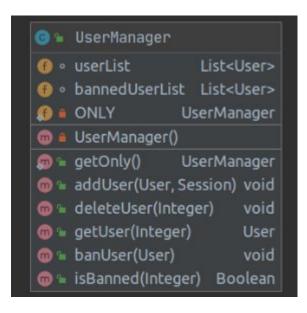
○ \(\text{ChatRoom} \)	
⑥ ≜ chatRoomName	String
⊕ ■ bannedUserList	List <user></user>
⑥ □ ChatRoom(String, Bo	olean, Integer)
@ addUser(Integer)	void
@ ← deleteUser(Integer)	void
addMsg(Message)	void
	void
@ '≡ getMsg(Integer)	Message
	Boolean
@ □ banUser(User)	void
@ ≈ isBanned(Integer)	Boolean
	Boolean
chatRoomID	Integer
@ userList	List <user></user>
nsal ist	List-Messages

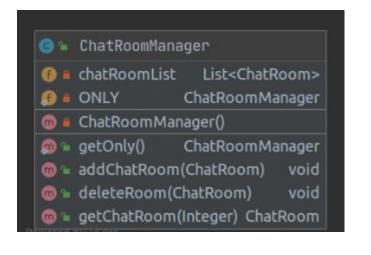
@ admin

	Triangle and	
	Message	
6 a	msgID	Integer
G a	type	String
() a	timestamp	Timestamp
(a	senderID	Integer
9 a	receiverID	Integer
(a	content	String
a	editContent(String)	void
® m	sglD	Integer
(P) ty	pe	String
O se	enderID	Integer
(P ti	mestamp	Timestamp
🕡 ге	ceiverID	Integer

⊜ w User		
⊕ username	String	
⊕ ⊆ User(String, Integer, String, String, Session)		
@ ≈ addChatRoom(ChatR	Room) void	
@ ₃ sentHate()	void	
📵 userID	Integer	
@ age	Integer	
📵 numOfHate	Integer	
interest	String	
p session	Session	
	String	
p jointChatRooms	List <chatroom></chatroom>	

Interface/Class Description

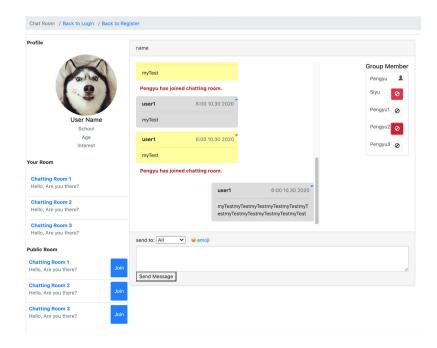




Interface/Class Description

Command Name	Description
ApproveCmd	Approve a user's request to join a chat room by the admin.
BanUserCmd	Ban a user from a chat room by the admin.
CreateRoomCmd	Create a chat room.
EditMsgCmd	Edit the content of a sent message.
InviteUserCmd	Invite a user to join a chat room by the admin.
JoinRoomCmd	Send out a user's request to join a chat room.
RecallMsgCmd	Recall a message from the chat room.
SendMsgCmd	Send a message to the chat room or a specific user.

GUI Introduction



Chat Room / Register / Login
Register
User Name
Enter your User Name
Age
Enter your Age
Interest
Enter your Interest
School
Confirm your School
Register
Login