# Project 2: GUICHAT

## Project Report

Chau Vu, Jan Rodriguez, Gabriel Frattallone

1. Revised Design
2. Conversation design
3. Protocol
4. Testing Report

# Revised Design

# Conversation design:

**Login process**

* Once users connect to the server, they type in IP address, port number, then select a username. Server checks if the username is valid and not being used
* AllUsersGUI will pop up where users can see all active users, create conversations, join conversations, and view histories

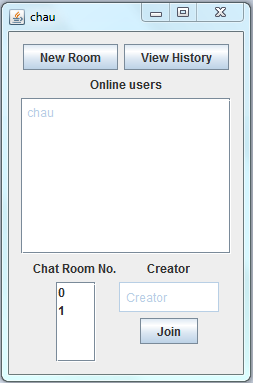
**Defining a conversation**

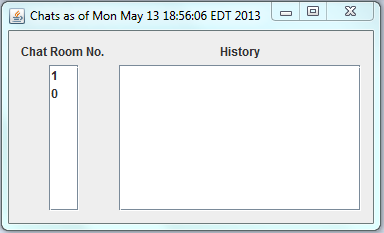
In our system, a conversation is like a chat room. Users can create chat rooms, invite people to their chat rooms. In addition, users can also choose chat rooms that s/he wants to join. Chat room is never destroyed, so users can return whenever they want. After having the AllUsersGUI shows up after logging in, users can:

*Click on “New Room” button on AllUsersGUI*: a new conversation GUI appears (in other words, a new chat room is created) where you can type in friends’ names and invite them to the conversation.

*Click on “Join” button on AllUsersGUI*: after selecting a chat room, users can see its creator’s name in a small text box. After clicking JOIN, a new conversation GUI will appear where users can normally chat, invite, leave…

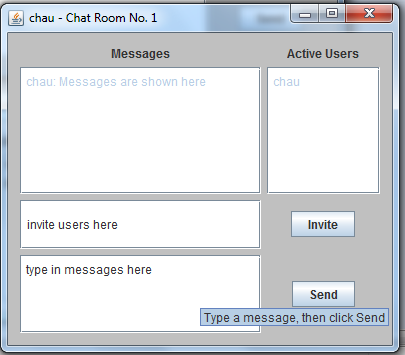
*Click on “View History” button on AllUsersGUI*: a History GUI will pop up and contain IDs of chat rooms users have joined by the time user clicks the button.





*Disconnect to server by closing AllUsersGUI:* once closing the AllUsersGUI window, user is logged off. His username can be reused. No memory of this log in is remembered.

Conversation GUI is where users send messages and invite other users.



* + When get invited, you must go to the conversation GUI that is popped up, if you don’t want to be in the conversation, just close the conversation window, and leave as if you have finished chatting.
  + The conversation GUI will show a textbox that shows messages, a textbox that inputs your messages, an invite buttons and textbox field where you can type in usernames of you friends to invite more people to the conversation. Also there is a text field that includes names of users who are currently in the conversation window.
  + Even when all users have left the chat room, they can still join the chat room later by clicking JOIN button in AllUsersGUI after selecting the right ID number of the chat room.

**Differences from previous designs**: We eliminate Private Chat Room because of some difficulties in implementing new features in our amendment. Also, chat rooms are no longer destroyed when no user is using them. Instead, users can easily return to the chat rooms by clicking JOIN in AllUsersGUI (after selecting the right ID number).

1. Protocol

**Client to Server Protocol**

OUTPUT    :== ( NEW | LEAVE | CREATOR | POST | INVITE | DISCONNECT | CONNECT )

NEW        :== "new” SPACE USERNAME SPACE CHAT\_NUM

LEAVE        :== "leave" SPACE USERNAME SPACE CHAT\_NUM

CREATOR    :== “creator” SPACE USERNAME SPACE CHAT\_NUM

POST        :== "post” SPACE USERNAME SPACE TEXT SPACE CHAT\_NUM

INVITE        :== "invite" (SPACE USERNAME  “,” )\* SPACE USERNAME SPACE CHAT\_NUM

DISCONNECT    :== "disconnect" SPACE USERNAME (SPACE CHAT\_NUM)\* SPACE CHAT\_NUM

CONNECT    :== “username” SPACE USERNAME

USERNAME    :== [a-zA-Z0-9]+

CHAT\_NUM    :== [0-9]+

SPACE           :== " "

TEXT        :== .+

**Server to CIient Protocol**

OUTPUT    :== ( NEW | POST | ADD | LEAVE | CREATOR | CREATE | CONNECT | DISCONNECT | START | ABORT )

NEW        :== "new” MEMBERS SPACE “?” SPACE HISTORY SPACE CHAT\_NUM

POST        :== "post” SPACE USERNAME SPACE TEXT SPACE CHAT\_NUM

ADD        :== “add” SPACE USERNAME SPACE CHAT\_NUM

LEAVE        :== “leave” SPACE USERNAME CHAT\_NUM

CREATOR    :== “creator” SPACE USERNAME

CREATE        :== “create” SPACE CHAT\_NUM

CONNECT    :== “connect” SPACE USERNAME

START        :== “start” MEMBERS SPACE “?” SPACE CHATS

ABORT        :== “abort”

DISCONNECT    :== "disconnect” SPACE USERNAME

CHATS        :== (SPACE CHAT\_NUM)\* SPACE CHAT\_NUM

MEMBERS    :== (SPACE USERNAME)\* SPACE USERNAME

HISTORY    :== (SPACE MESSAGE)\* SPACE MESSAGE

MESSAGE    :== USERNAME “:” SPACE TEXT

USERNAME    :== [a-zA-Z0-9]+

CHAT\_NUM    :== [0-9]+

SPACE           :== " "

TEXT        :== .+