



Chesster Chat

User Manual

An instant messaging application with an integrated chess game

Created by:

Jada Berenguer, *team manager*

Benny Lin, *presenter*

Marvis Nguyen, *recorder*

Daniel Pajulas, *recorder*

Armand Ahadi-Sarkani, *reflector*

Kevin Selda, *reflector*

Raiyan Bin Nasim, *reflector*

All affiliated project contributors are 2nd year computer engineering students at the University of California, Irvine.

Table of Contents

Glossary	3
1. Instant Messaging	4
1.1. Usage scenario	4
1.2. Goals	4
1.3 Features	5
2. Installation	5
2.1. System requirements	5
2.2. Setup and configuration	5
2.3. Uninstalling	6
3. Functions and Features	6
3.1. Detailed description of client-server communication	6
3.2. Detailed description of login and registering algorithms	6
3.3. Detailed description of chess game integration	8
Copyright	9
Error Messages	9
Index	11

Glossary

Account: Stores user's personal info.

Client: The user that initiates a message by sending a request to the server.

Command line: Where one enters a command that Linux will execute.

Contacts: A list of other users that you are friends with.

Emoji: An image that conveys a feeling or object.

Friend: A user that you are able to chat with.

Log in: In order to access your account, you have to provide your credentials (username and password) to send and receive messages.

Linux: An open-source operating system based on the UNIX architecture.

Multiplayer: More than one player.

Online status: Displays whether or not a user is online.

Password: A string of characters that allows access to an online service.

Server: The user that handles the client's request and passes that message to another user.

Sockets: A particular endpoint for sending and receiving data over an internet network.

User: One who is able to use the chat program.

Username: An identification used by the user in order to access their account.

1. Instant Messaging

1.1. Usage scenario

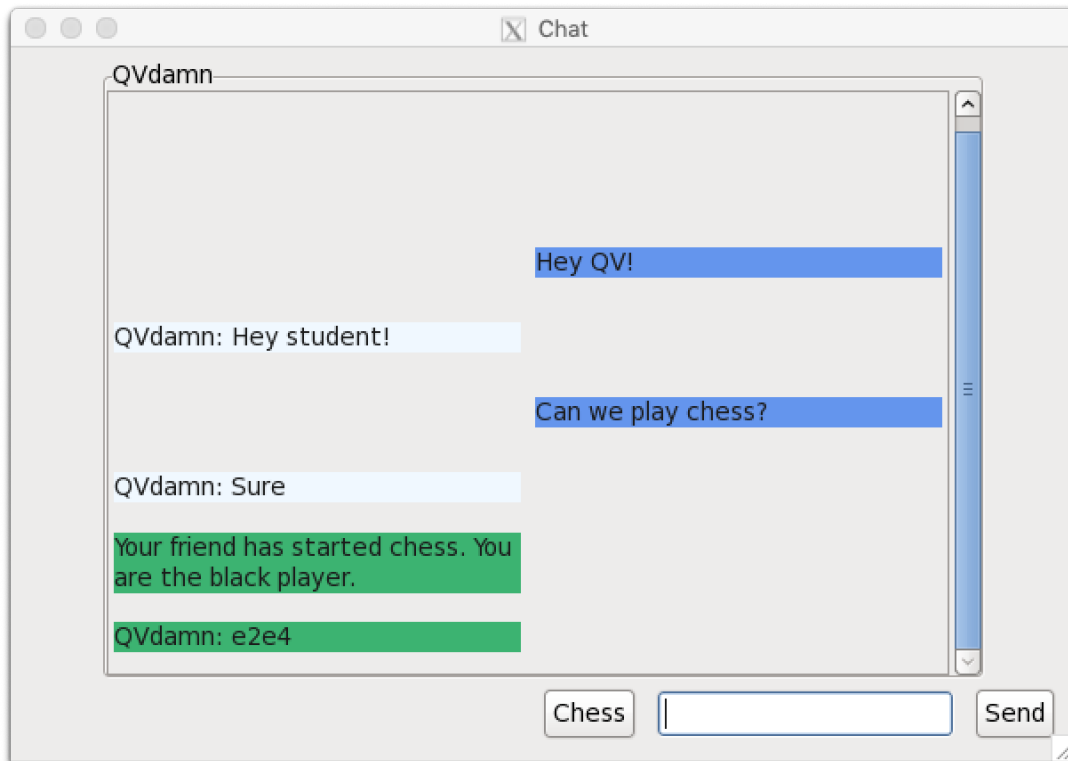


Figure 1: Main chat screen

The screenshot in *Figure 1* is an illustration of a typical chat, with the user profile at the top, a button that initiates chess, and messages flowing in from the other user on the left hand side of the screen, with messages from the current user coming in on the right hand side. A text composition window and a send button is shown on the bottom of the window.

1.2. Goals

- Extend the chess game that was created by adding a multiplayer functionality over a network.
- Create a chat application, in which a user can send messages and react to other users' messages.

- Integrate a single-window chess program that updates in real time based on the moves that players make.

1.3 Features

- Allow the user to register for a chat account and log into their chat account.
- Allow the user to add friends to their contacts and be able to chat with their friends.
- Allow the user to remove friends.
- Allow the user to accept and decline friend requests.
- Allow users to play a game of chess through the chat.

2. Installation

2.1. System requirements

- Linux environment with support for command line interface
- X11 window system installed (such as XMin on Windows or XQuartz on macOS)
- Support for GTK+ 2.0 and Pango font library (see <http://macappstore.org/pango/> to install on macOS)
- Processor with x86 architecture on server running in Linux environment
- Minimum free disk space: 1 GB
- Minimum memory: 512 MB

2.2. Setup and configuration

1. Open a window in your preferred Linux terminal program.
2. Using the command line, search through your filesystem to find the folder with the archive file. For example:
cd computer
cd downloads
3. Extract the software file from the archive file using the following command:
tar -czvf Chess_V1.0_src.tar.gz src
4. In your terminal, enter the bin folder and type:

`./GUIClient [hostname] [port number]` to launch the client program in GUI, from the same directory as the server
`./Client [hostname] [port number]` to launch the client program in the command line interface, from the same directory as the server
`./RemoteGUIClient [hostname] [port number]` to launch the client program in GUI, from any directory
`./RemoteClient [hostname] [port number]` to launch the client program in the command line interface, from any directory
`./Server [port number]` to launch the server program

2.3. Uninstalling

1. Open a window in your preferred Linux terminal program.
2. Using the command line, search through your filesystem to find the ChatFinalUser folder.
3. Delete the program folder using the following command:
`rm -rf ChatFinalUser`

3. Functions and Features

3.1. Detailed description of client-server communication

Chatting with another user:

User input: A text-based message to send to another user.

Program output: If the other user replies, a message will be shown on the current user's screen.

See *Figure 1* for a screenshot of the chat interface and more information. This interface allows the user to send text-based messages to other users. The user initiating a message is called the client, and their message is transmitted to the server securely and over the designated socket. Then, the server uses the same socket to transmit that message to another client, namely, the client that the current user is chatting with. The same process is repeated in reverse when the recipient chooses to send a reply.

3.2. Detailed description of login and registering algorithms

Login:

User input: A valid username and its corresponding correct password.

Program output: Window that leads to the chat application via the friends screen, or an error.

See *Figure 2a*. Clicking “Login” will login a user based on the credentials they have entered, if they are valid.



Figure 2a: Launch screen with login, register, and exit options

Registering:

User input: A new username and password.

Program output: Window that leads to the chat application via the friends screen, or an error.

See *Figure 2b*. Clicking “Register” will register a user based on the desired credentials they have entered. If the username a user desires matches that of another user, or if the username and/or password contain invalid characters, the program will output an error (see the Error Messages section for more details).

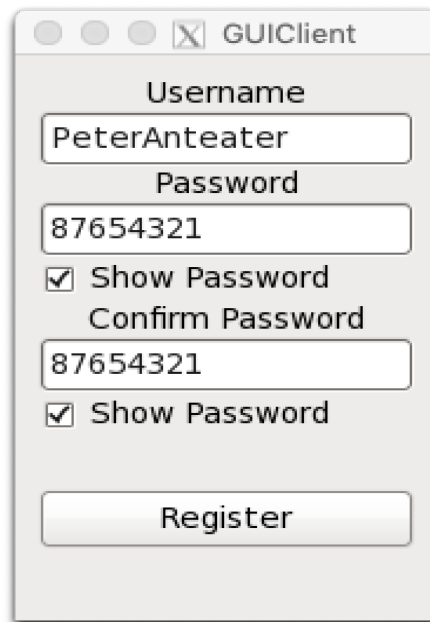


Figure 2b: Register window for user to enter the credentials they prefer to use.

Adding/Removing a Friend:

User input: User ID of the user you want to add/remove.

Program output: Sends target user a friend request or removes them from current user's friends list.

Figure 3 shows the current user's list of friends, with options to play one of them in chess or just chat them.

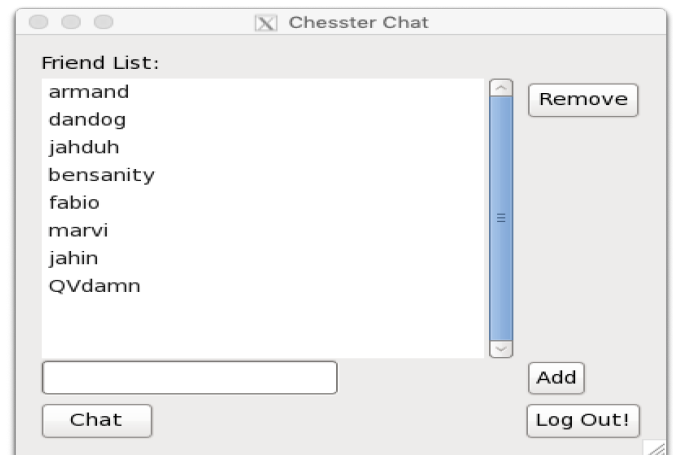


Figure 3:

Friends list

3.3. Detailed description of chess game integration

User input: Click on the chess button during a chat (see figure 1, chess button will be implemented).

Program output: Playable chess game.

Figure 4 shows the chess board setup for playing chess while chatting with another user. The chess game will open in the command line window which operates concurrently to the current chat. *Please see the Chesster User Manual (for the chess program) for explicit information regarding the chess program.*

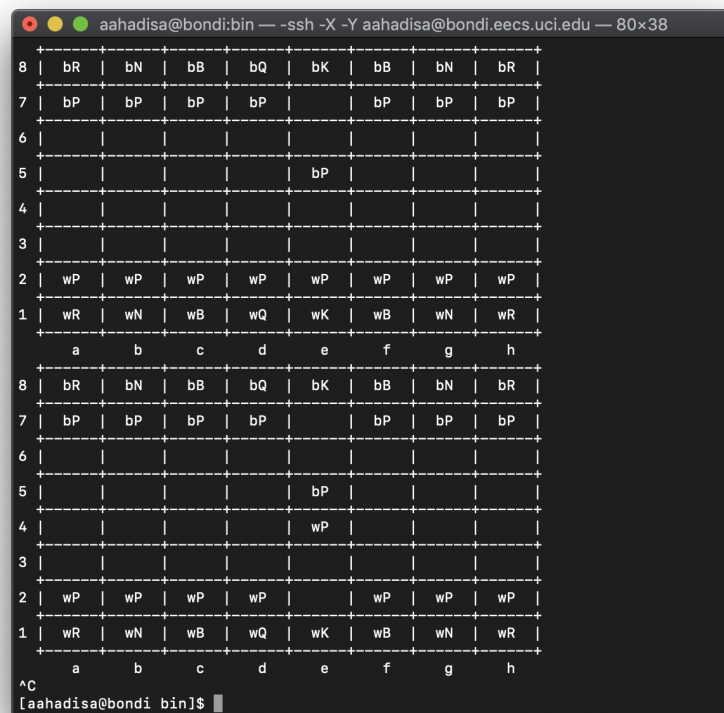


Figure 4: A chess board setup

Copyright

© **2019 Chesster**. The rights to this software and its documentation belongs to the members of this team: Jada Berenguer, Benny Lin, Armand Ahadi-Sarkani, Raiyan Nasim, Kevin Selda, Marvis Nguyen, Daniel Pajulas. All members of this team are enrolled as undergraduate computer engineering students in the Henry Samueli School of Engineering at the University of California, Irvine.

Error Messages

To see all chess-related error messages, please see the Chesster User Manual (for the chess program).

“The username you entered has already been taken. Please try again.”

You will encounter this error if you are registering for an account (see section 3.2), but your desired username matches that of an existing user. At this point, the program will prompt you to either log in as that user or create a new account with a different username (that is not taken).

“The username you entered is invalid. Please try again.”

You will encounter this error if you are logging in with a username that does not exist in the server’s database. At this point, the program will prompt you to try again.

“The password you entered is incorrect. Please try again.”

You will encounter this error if you are logging in to a valid account, but with the wrong password. At this point, the program will prompt you to try again.

“This user is already your friend, or they do not exist.”

You will encounter this error if the username of the user you are trying to add as a friend is invalid, is already your friend, or if you are adding yourself as a friend. At this point, the program will prompt you to enter a different username.

Empty field errors

You will encounter this type of error if you leave a login, registration, or other required field empty.

Connectivity errors

You will encounter this type of error if you try to connect to a server that is not active, does not exist, or could not be bound to a socket.

Index

A

Account	3, 10
---------	-------

C

Client	3, 6
Chess	4, 5, 8
Command line	3, 5, 6
Communication	6
Copyright	9

E

Emoji	3
-------	---

Error	7, 9, 10
F	
Features	5, 6
Friend	3, 5, 7, 8, 10
G	
Goals	4
I	
Instant messaging	4
L	
Linux	3, 5, 6
Log In	3, 5, 10
M	
Multiplayer	3
P	
Password	3, 7, 10
Program output	6, 7, 8
R	
Register	6, 7, 10
Remove	7, 8
S	
Setup	5
Server	3, 6, 10
Sign up	5

Sockets	3
Status	3, 5
System requirements	5

U

Usage	4
User	3, 4, 6, 8, 10
Username	3, 7, 10
User input	6, 7, 8
Uninstalling	6