**Linux chatter box design**

Server Diagram



SERVER PSEUDO CODE

# Linux server

## function Main

Get port number

Open file to write log

Open socket

Initizlie client list to save and FD\_SET arrays for multiplexing

Start multiplexing

Reset FD\_SET

Accept connection and update client list and update FD\_SET

If no more space for new user, send error message

Check user, and if multiplex signal is detected, receive message

Check message type and convert message and make sender thread

Save log using received message

If type is quit, close socket

## function open socket

Create socket and initialize socket option to possible to be reused immediately

Initialize address information

Bind socket

Listen socket

## function receive message

Receive message structure

Check type of message structure

If connect type, reset user name and send user list

Generate send message depending on the type.

## function initClientInfo

Initialize a client socket number and name on the clientList array

## function CreateMessage

Add user name from client list and fill message structure using information parameters

## function sendList

Send a list of client users

## function Write message

Check whole client list from first to last index

If client name is same with message client name, not send

If it is not empty array of clientList, send message to the client

If it is connect or quit message, send client list to update

## function writeOne

send message to only 1 client

## function display List

if the client list is not empty, display name and socket number

client diagram



client PSEUDO CODE

# clientsrc class function

## constructor

Initialize parent object for UI

## function clientstart

Create socket

Initialize socket address

Connect to the server

Send personal information

Receive user list

Create thread to receive message asynchronously

## function sendPersonalInfo

Read nickname of client user and make structure which is CONN type

## function createMessage

Fill message structure using input information

Write message

write message structure with type info

if type is quite, close socket

# non class functions

## function recvlist

Receive message as a client structure type

Display the list of client users.

## function readmsg

Keep read message

Check message type

If message type is ‘connect’ or ‘quit’ receive client list

Else display message with user name