

# Super Hyper Messenger evo EX ver.1.1.26

B02902011黃泓硯

B02902063陳俊瑋

B02902105 廖瑋中

# Register

- Client creates a socket to register account
  - !S #register
  - <username>
  - <password>
- Server returns acknowledge if succeed, repeat username if fail
  - !Y #register success
  - !N #register fail (repeat username)

time



client

register

username

password

success or fail



server

# Login

- Client creates a socket to login
  - !L #login
  - <username>
  - <password>
- Server returns acknowledge if succeed, repeat login or wrong password if fail
  - !R #repeat login
  - !u #wrong username
  - !p #wrong password

time



client

login

username

password

success or fail



server

# Private message

- Sender sends private message to server
  - !m <receiver name> <sender name> <message> #private message
- Server sends private message to receiver
  - !m <receiver name> <sender name> <message> #private message

time



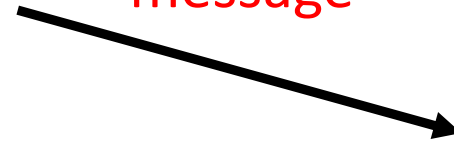
sender

message



server

message



receiver



# Broadcast message

- Sender sends broadcast message to all receivers
  - !a <sender name> <message> #broadcast message
- Server sends broadcast message to each receivers
  - !a <sender name> <message> #broadcast message



time



sender

message



server

message



receiver A

message



receiver B

message



receiver C



# Private file (Reject)

- Sender sends private file request to server
  - !f <sender name> <receiver name> <filename> #private file
- Server sends private file request to receiver
  - !f <sender name> <receiver name> <filename> #private file
- Receiver sends rejection to server
  - !r <receiver name> <sender name> #reject
- Server sends rejection to sender
  - !r <receiver name> <sender name> #reject

time



sender

request



server

request



receiver

reject



reject



# Private file (Accept)

- Sender sends private file request to server
  - !f <sender name> <receiver name> <filename> #private file
- Server sends private file request to receiver
  - !f <sender name> <receiver name> <filename> #private file
- Receiver sends acceptance to server
  - !y <receiver name> <sender name> #accept
- Server sends acceptance to sender
  - !y <receiver name> <sender name> #accept

# Private file (Accept) (cont.)

- Sender sends an inform to tell server that it wants to send a datagram
  - `!c <sender name> <receiver name>` #continue to send datagram
- Server receives the inform and prepare to receive the datagram
- Sender sends the datagram to server
  - `<datagram>`
- Server receives the datagram

# Private file (Accept) (cont.)

- Server sends an inform to receiver that it wants to send a datagram
  - `!c <sender name> <receiver name>` #continue to send datagram
- Receiver receives the inform and prepare to receive datagram
- Server sends the datagram to receiver
  - `<datagram>`
- Receiver receives the datagram

# Private file (Accept) (cont.)

- Repeat until end of file
- Sender sends end of file to server
  - !e <sender name> <receiver name> #end of file
- Server sends end of file to receiver
  - !e <sender name> <receiver name> #end of file

time



sender



server



receiver

request

request

accept

accept

datagram

datagram

OK

OK

⋮

end file

end file





# Broadcast files

- Sender sends private file to all receivers simultaneously
  - The protocol of broadcasting files is the same as the one of sending private file

time



sender

request



accept



datagram



OK



end file



server

request



accept



datagram



OK



end file



receiver A  
receiver B  
receiver C

⋮

# Historical message

- When client closes the window of the chat room, client will preserve the historical messages in a log file, which is created at local path.
- When client opens the window of the chat room again, client will open the log file and retrieve historical messages.