

# Final Project Report

## (1)Team members

資工三 B02902025 呂承洋

資工三 B02902037 宋品賢

資工三 B02902123 藍祥予

## (2)Task distribution

呂承洋: All server code & shared code & database

宋品賢: All client code & shared code & connection

藍祥予: Connection

## (3)User guide

0.How to build:

Use “make” or “./gradlew buildAll” to build both client and server jar file. Use ./gradlew buildClient and ./gradlew buildServer to build client’s and server’s jar file respectively.

The output will be put in build/libs/.

1. How to use server:

To run server, use “make runServer” to run server. The default port is 7122. To assign a port, use “make runServer PORT=XXXX”.

2. How to use client:

To run client, execute “make runClient”. There would be a login/register page for client to fill in account, password, and server IP, port. When one tries to register an account already taken by others, it’ll be rejected.

After successfully logging in, there will be 2 tables in the page. One for 1-to-1 chat rooms and the other for rooms with more capacity.

Press “Find” by the upper table to search for another user and build a chat room just for the 2 of you. The two people will be add to the room by default and the room will appear in the user’s room list as well as the other user.

Press “Build” by the lower table to build a room for more than 2 people, by default, only the builder will be in the room. When entering the room all users of the room can choose to “AddPeople” or “Leave”. The room will show up or disappear in the lower table of an user once he/she is added or chooses to leave, respectively.

Users can type message in the textbox and press “confirm” to send message. After pressing “confirm”, a box will jump out to ask for final confirmation of sending.

User can upload files by pressing the “Upload File” button. After pressing the button, a pop up window will show up and allow user to select a file to upload. After pressing confirm, the chosen file name will be shown in the box by the button. User can still type message in the textbox and send the text as well as the file at the same time.

When a file is sent to a room, owners of the room can press the “Download” button by the filename to download the file. The file will be downloaded at default download directory (usually at “~/Downloads” or “下載” in windows).

## (4)Protocol introduction

- Client-to-server

These two field will be add by ServerConnection class.

sender\_name: String, sender name

sequence\_number: String,connection sequence number, will be placed if there is no sender name (login/register).

1. registration

instruction: REGISTER

account : String

password : String

- 2.login

instruction: LOGIN

account : String

password : String

3. logout // place by ServerConnection class

instruction : LOGOUT

4. create chat room // find people

instruction: CREATE\_CHAT\_ROOM

name: String, username one wished to find

5. create meeting room // multiple

instruction: CREATE\_MEETING\_ROOM

room\_name : String, name of the room one builds

6. send message

instruction: SEND\_MESSAGE

room\_id : String

content

7. send file

instruction: SEND\_FILE

room\_id : String

content : String, filename

8. request file

instruction: REQUEST\_FILE

name : String, file token

9. add person

instruction: ADD\_PERSON

room\_id : String

name : String

10. leave room

instruction: LEAVE\_ROOM

room\_id : String

11. open room // sync historical message

instruction: OPEN\_ROOM

room\_id : String

12. File upload finish: // placed by ServerConnection

instruction: FILE\_UPLOAD\_FINISH

content: String, token

- Server-to-client

room\_type:["single", "multiple"]

type = ["file", "message"]

1. login // success

- instruction: 100  
list (List<Map>) contains: status, room\_id, room\_type, room\_name
2. login/register fail  
instruction: 110  
content
3. room list update  
instruction: 200  
list (List<Map>) contains: status, room\_id, oom\_type, room\_name
4. sync room message  
instruction: 300  
room\_id  
list (List<Map>) contains: sender\_name, type, file\_id, content, time\_stamp
5. add room message  
instruction: 310  
room\_id  
list (List<Map>) contains: sender\_name, type, file\_id, content, time\_stamp
6. add person response  
instruction: 400  
room\_id  
content
7. multiple chat room response  
instruction: 410  
content
- 8 query person response  
instruction: 420  
content
9. ack send file  
instruction : 500  
room\_id : // port for send FILE
10. ack request file  
instruction 510  
room\_id : // port for receive file  
content : String, filename

## (5)Process flow diagram

