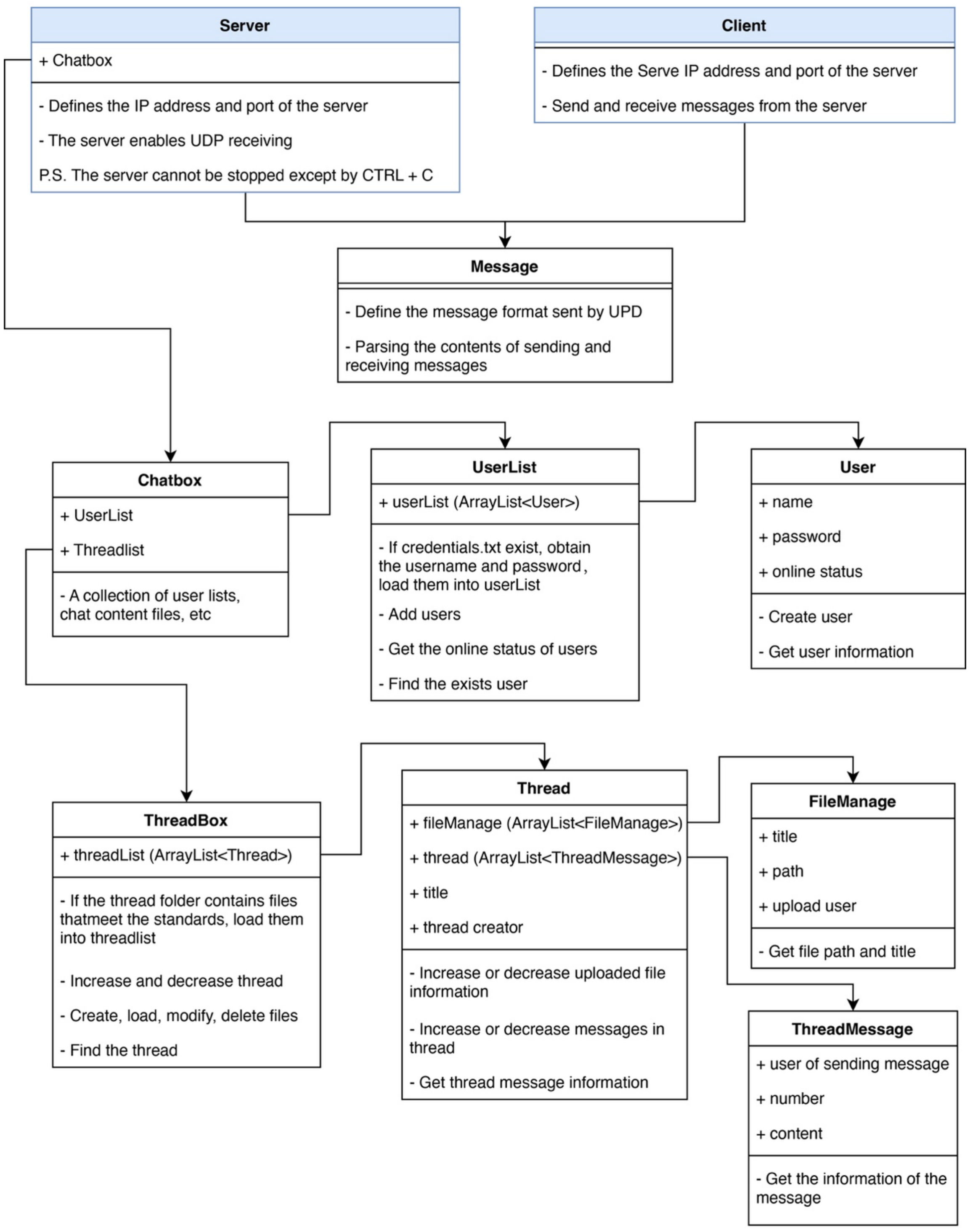
COMP3331 Assignment Report

-------Jinghan Wang z5286124

The Assignment is mainly implemented in Java, which completes the creation of a client and a server and can transmit messages and files to each other.

The following diagram shows the structure of all java files in the assignment and the main function of each java file.


Server
+ Chatbox
- Defines the IP address and port of the server
- The server enables UDP receiving
P.S. The server cannot be stopped except by CTRL + C
Client
- Defines the Serve IP address and port of the server
- Send and receive messages from the server
Message
- Define the message format sent by UPD
- Parsing the contents of sending and receiving messages
Chatbox
+ UserList
+ Threadlist
- A collection of user lists, chat content files, etc
UserList
+ userList (ArrayList<User>)
- If credentials.txt exist, obtain the username and password, load them into userList
- Add users
- Get the online status of users
- Find the exists user
User
+ name
+ password
+ online status
- Create user
- Get user information
ThreadBox
+ threadList (ArrayList<Thread>)
- If the thread folder contains files thatmeet the standards, load them into threadlist
- Increase and decrease thread
- Create, load, modify, delete files
- Find the thread
Thread
+ fileManage (ArrayList<FileManage>)
+ thread (ArrayList<ThreadMessage>)
+ title
+ thread creator
- Increase or decrease uploaded file information
- Increase or decrease messages in thread
- Get thread message information
FileManage
+ title
+ path
+ upload user
- Get file path and title
ThreadMessage
+ user of sending message
+ number
+ content
- Get the information of the message

Message:

First, the following figure shows the information structure when the server communicates with the client using UDP.

Action(number)	(Space)	Status(number)	(Space)	username	(Space)	content
1 or 2 bytes	1 byte	1 byte	1 byte		1 byte	

For First Part, it has twelve actions in this assignment

FCT(First Connect) = 0	LOG(Log in) = 1	CRT(Create Thread) = 2	MSG(Message) = 3
DLT(Delete Thread) = 4	EDT(Edit Thread) = 5	LST(List Thread) = 6	RDT(Read Thread) = 7
UPD(Upload) = 8	DWN(Download) = 9	RMV(Remove Thread) = 10	XIT(EXIT) = 11

FCT, LST, XIT content part structure: Brief function introduction;

LOG content part structure: Include Password

CRT, RDT, RMV content part struct: Include Thread Name

MSG, DLT, UPD, DWN content part struct:

Thread Name	(Space)	Extra content

EDT content part struct:

Thread Name	(Space)	Thread Message Number	(Space)	Message

For Second Part, it has four Status in this assignment

SRS(Server Send, Success) = 0	SRF(Server Send, False) = 1
CTS(Client Send, Success) = 2	NGI(Not Log In) = 3

SRS and SRF will send from Server, CTS will send from Client.

NGI will send when Server find the unauthenticated users who uses the function directly.

Client:

Client. Java is mainly the entrance of the client. Firstly, it sets the IP address and port of the server and obtains the absolute path of the current location of this file to facilitate uploading and downloading files. Then use scanner to get the input of each instruction and verify whether the input content of the command meets the requirements. If it meets the requirements, use message to obtain key information and send it to the server with DataProgramSocket. The time out of each socket is 600ms. At the same time, if the server does not respond 10 times, the program will stop.

In many tests, when the server is directly shut down and then restarted, when the client is not shut down, the client can skip username and password verification and directly upload and send messages. Therefore, if the client returns the status NGI (not log in), the program displays "the server may restart, the client closed" and then stops directly.

Server:

The server first sets the receiving IP address and port of UPD. Then, during the creation of UserList, search credentials.txt file under the same path with Server.java. If it exists, load the username and password into the UserList and set them as offline. If it does not exist, create a file to facilitate the placement of new user information.

After that, search for all files in the path that do not contain ".", set their file name to Thread title, extract from the first line and set it to Thread Creator. After that, if a message is sent by the user, add it to thread. When it is file uploading information, added it to fileManage.

After completed below, the port is opened to accept messages sent by the client. When sever get the message, it will record the Client IP address and the port. The message will be separated and completed the operation as required. After that Server will send the feedback to Client. The program cannot be stopped except Ctrl + C.

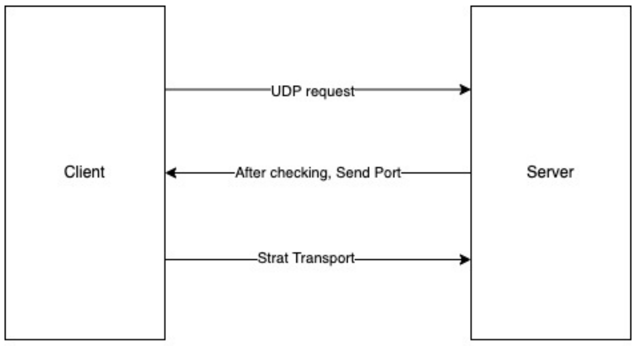
UserList, ThreadBox:

When Adding new users, new messages and new file messages, the program uses new BufferedWriter(new OutputStreamWriter(new FileOutputStream(file, true)) to append the correct message to the end of the file.

When deleting the message, the program first uses new BufferedWriter(new OutputStreamWriter(new FileOutputStream(file)) to overwrite the original file, and then uses new BufferedWriter(new OutputStreamWriter(new FileOutputStream(file, true)) to add the deleted content later.

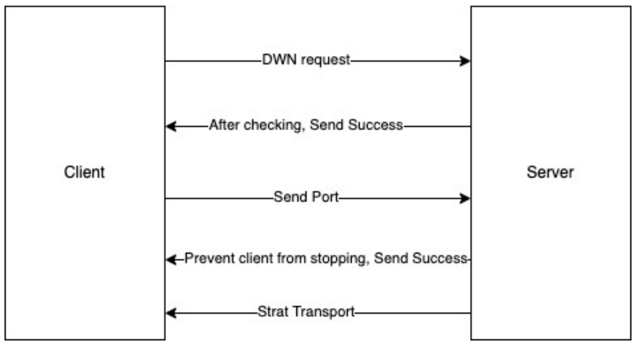
UDP:

After completing all the checks before uploading, the server creates a random port with ServerSocket to prepare for TCP transmission. The server sends this port to the client with UDP. After the client accepts it, it uses this port for transmission. First, set the byte array with 102400 bytes, then use BufferedInputStream to read the contents of the uploaded file, and then send it to the server. After the server accepts it, place the file in the folder of the corresponding thread, and add the file information to the information.txt file.



DWN:

When the check before downloading is completed, the server sends a completion message to the client. The client creates a random port and sends it to the server. To prevent the client from stopping because it does not receive a reply, the server sends a message to the client and then starts the upload operation. The file will be stored under the same path with Client.java file.


Client
DWN request
After checking, Send Success
Send Port
Prevent client from stopping, Send Success
Strat Transport
Server

Reference:

1. In Client.java and Server.java file, the function (private static String getData) converting byte [] to string, I learned some code from (private static String printData) PingServer.java (private static String printData) which provided by COMP3331 Lab02.
   Client.java 436:439
   Server.java 41:44

2. In the TCP file transmission between UPD and DWN, I learned some codes to convert the transmitted file into a byte[] that can be sent and how to convert the received byte[] into a file. https://www.cnblogs.com/debug-the-heart/p/13253886.html
   Client.java 270:277 341:357
   Server.java 251:264 304:313