# Distributed System Assignment 2 Report

Name: Pengbo Xing

ID: 1287557

Date: 2022/05/30

## 1. Problem context

This project implements a whiteboard that allows multiple users to draw simultaneously on a whiteboard by using sockets and TCP. The thread-per-connection architecture was used when dealing with the connections. There are some basic foundations of the whiteboard, users could draw on the whiteboard simultaneously, they could have a chat using the chat window, the manager who is the server could do all the above and also could save the file, open a file, clear the whiteboard and close all whiteboard. For the basic draw foundation, If the server draws something, it just sends the request like "Draw-Line-thinness-color-x1-y1-x2-y2-!" to all users (client) and lets them draw the same on their whiteboard. If the client draws something, it first sends the request to the server, the server draws on its board, then the server sends back the request to all clients and lets them draw it. There are some small pup-up windows like if the user wants to join, it will send a request like "Ask-Username" to the server, then there will be a pop-up window for the server to choose yes or no, if yes, the client is allowed to join, otherwise, the client is rejected to join.

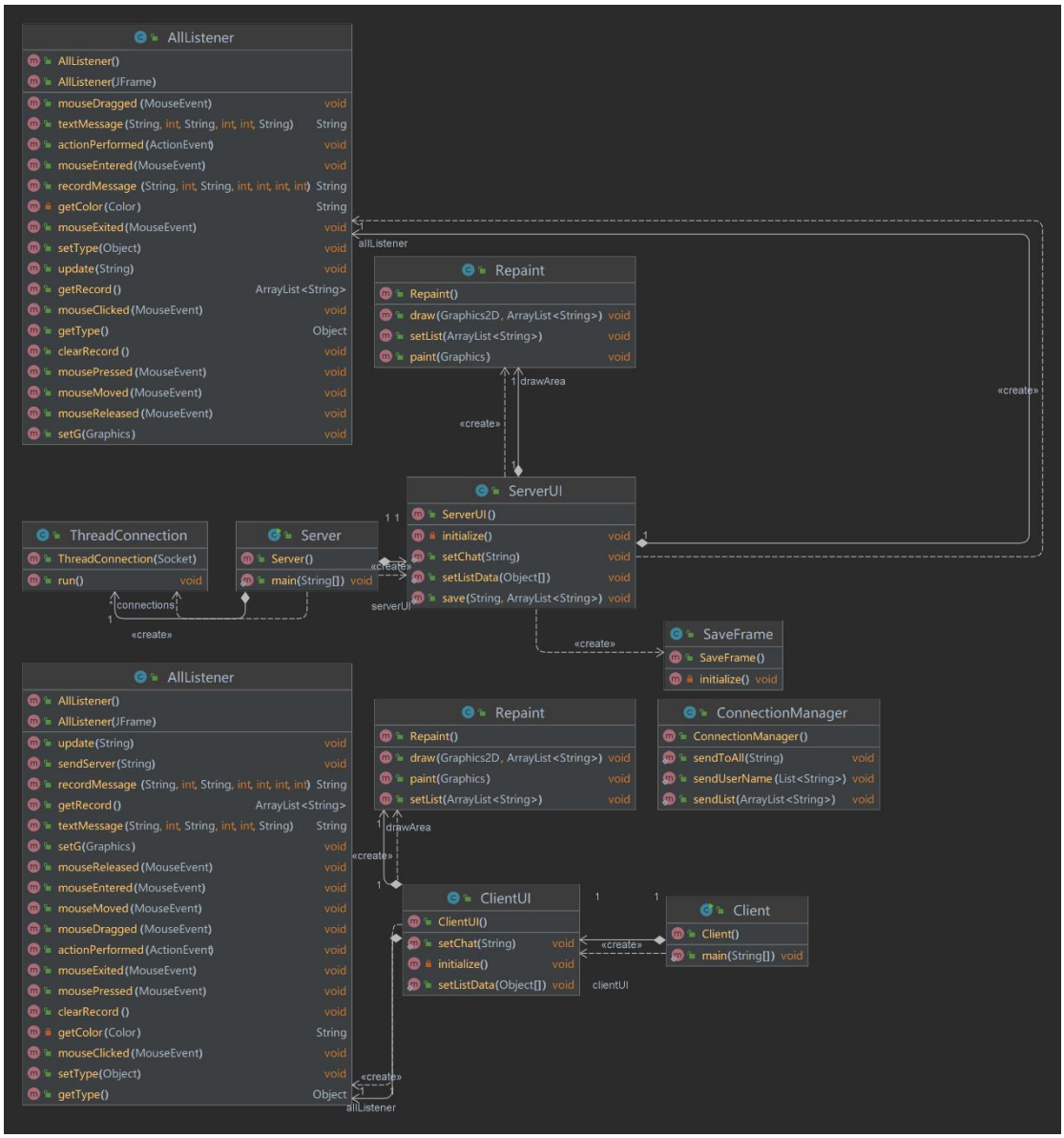
## 2. Components of the system

There are three basic components in the system, the Client, the Server, and GUI. More specifically, there are two user interfaces, one for the Client (Guest) and one for Server

(Manager). They both have the basic foundations which are drawing, and Server has more other foundations than I talked about before.

## 3. Class design and Interaction diagram

Class UML:



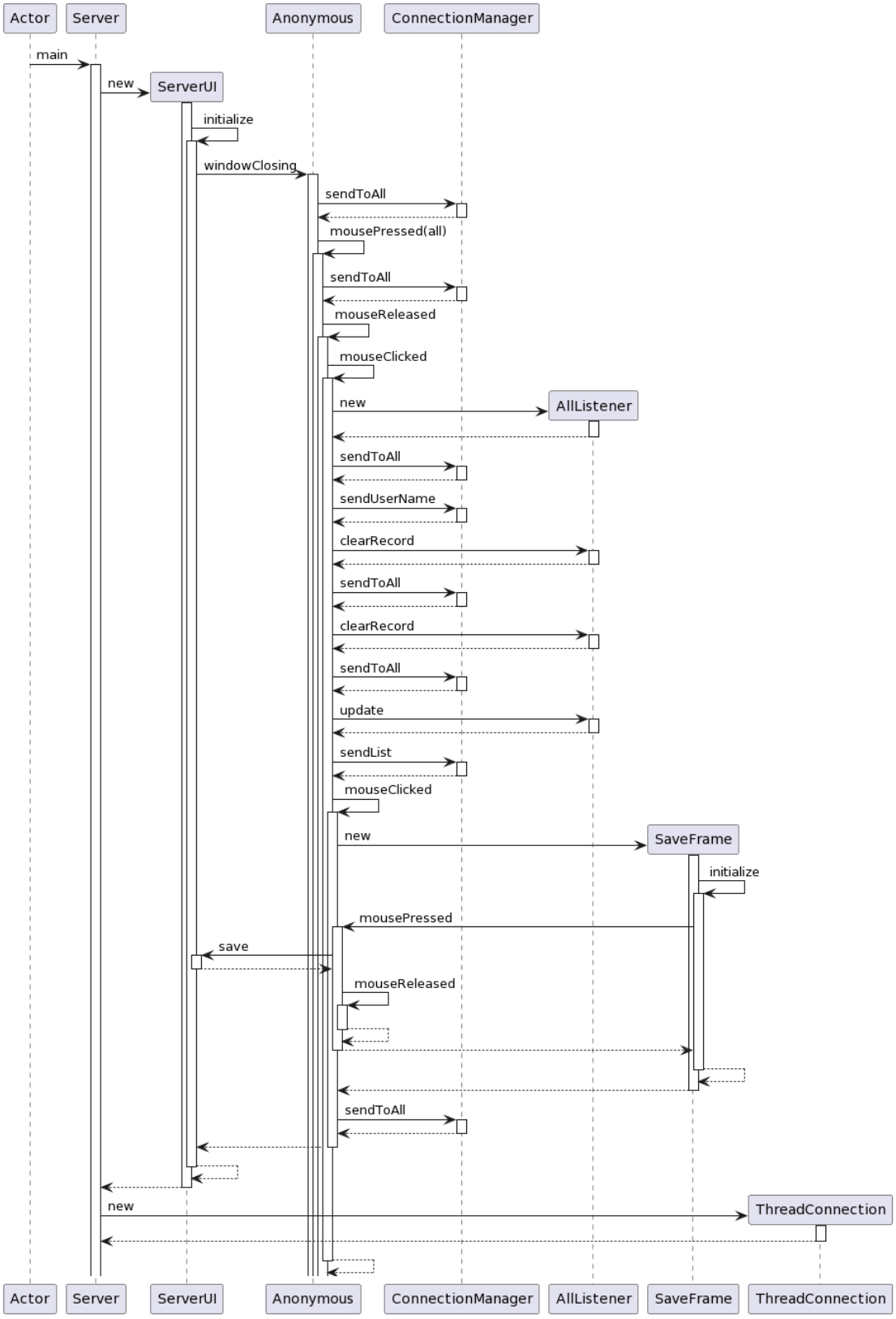

- The client and server part both have a similar class, AllListener, and repaint. AllListener is a mouse listener that records the mouse location when you press the mouse and release the button, using the location to draw a shape. Repaint allows a user to repaint a canvas based on a list of requests. It will do the operation on the

list one by one. The foundation is used when a user receives requests from others, the user’s whiteboard could show others' draws.

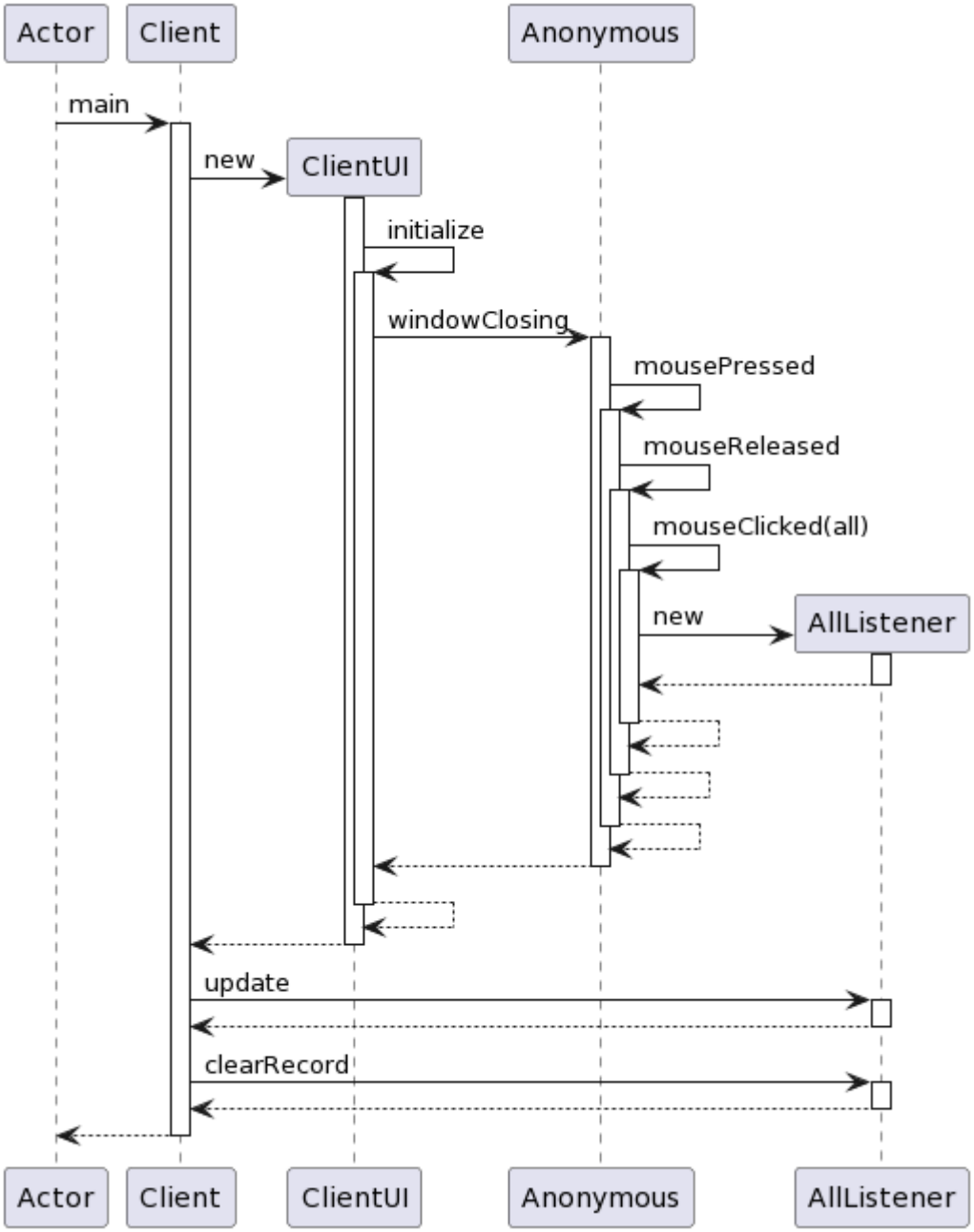
- Client class is a place to make client connections, and do the operations received from the server, like “Paint, Chat, Kick, User, New”. Paint for painting the request, Chat for displaying all chat messages, Kick for leaving the whiteboard (was kicked), User for displaying the user list, New for creating a new whiteboard.
- ConnectionManager is where the client sends the message and lists to all guests. All the methods are used synchronized to make sure the list, and string when using it to make sure its concurrency is safe.
- SaveFrame is a saving file UI for writing a filename to save.
- The server class creates the server socket and maintains on listening clients.
- ThreadConnection is created for each connection, then it operates the request sent from that user, the request like “JoinIn, Ask, Paint, Chat, Close.” JoinIn means a new user comes, paint, user list, chat list, should be updated for that user. Ask means a new one wants to join and ask for permission. Paint means to paint on the whiteboard and send it to others. Chat means it puts new chats on the chatbox and updates others. Close means it leaves the whiteboard, and the user list and connections should be updated.

Interaction diagram:

Server


Actor
Server
Anonymous
ConnectionManager
main
new
ServerUI
initialize
windowClosing
sendToAll
mousePressed(all)
sendToAll
mouseReleased
mouseClicked
new
AllListener
sendToAll
sendUserName
clearRecord
sendToAll
clearRecord
sendToAll
update
sendList
mouseClicked
new
SaveFrame
initialize
mousePressed
save
mouseReleased
sendToAll
new
ThreadConnection
Actor
Server
ServerUI
Anonymous
ConnectionManager
AllListener
SaveFrame
ThreadConnection

Client




## 4. Critical analysis and conclusions

In the ServerUI, there are menu buttons for saving the draw, creating a new draw, opening a graph, and closing the current whiteboard. For the saveAs button, I am using JFileChooser to choose the location and filename for saving the file. For the save button, just provide a filename, and it will save the file in the current location, the save foundation only supports txt files, which save all the operations like “Line-thinness-

color-x1-y1-x2-y2-!” as a string in the text file. For the open button, the Server opens that file, reads line by line, adds those operations to the draw list, then update them on the canvas, and sends the operation to other users. The new button just clears the draw list and sets a new draw panel, send to all other users. The close button just closes the frame and sends the close operation to other users.

The kick button allows the manager to select a user from the user list first, remove it from the user list, remove the connection, and the new user list to other users.

The chat button in ServerUI and ClientUI have a similar foundation, they both add a new string to the chat message, and when clicking it, update the frame list. If it is a client, send the chat message to the server, if it is a server, send the chat message to all users.

The main problem is dealing with the swing, it’s a big challenge to use at first, especially drawing the graph, it takes me a long time to draw a simple line on the panel. After understanding how it works, everything becomes easy.