COMP90015

Distributed Shared White Board

Report

Ziyang Huang
1067800

Introduction

In this assignment, a distributed shared whiteboard is implemented. The system allows multiple users to connect to the shared whiteboard and draw different shapes on the canvas. The system allows users to communicate with each other using a chat box. The system allows the manger to new, open, save, and close the application. Also, the system allows the manager to kick out a certain user.

System Architecture

The system is implemented using client-server architecture and socket programming. Manager, in this case, is the server side, and users are the client side. Information is stored on the server side and propagate to every client when needed.
Since client-server architecture is used in assignment 1, I am more familiar with it. It is easy to implement and straightforward. The system also uses thread-per-connection architecture which creates a thread for every user connection. Thread-per-connection is the most suitable here since we can handle one user's request by simply having one thread for that user.

Communication protocols & message formats

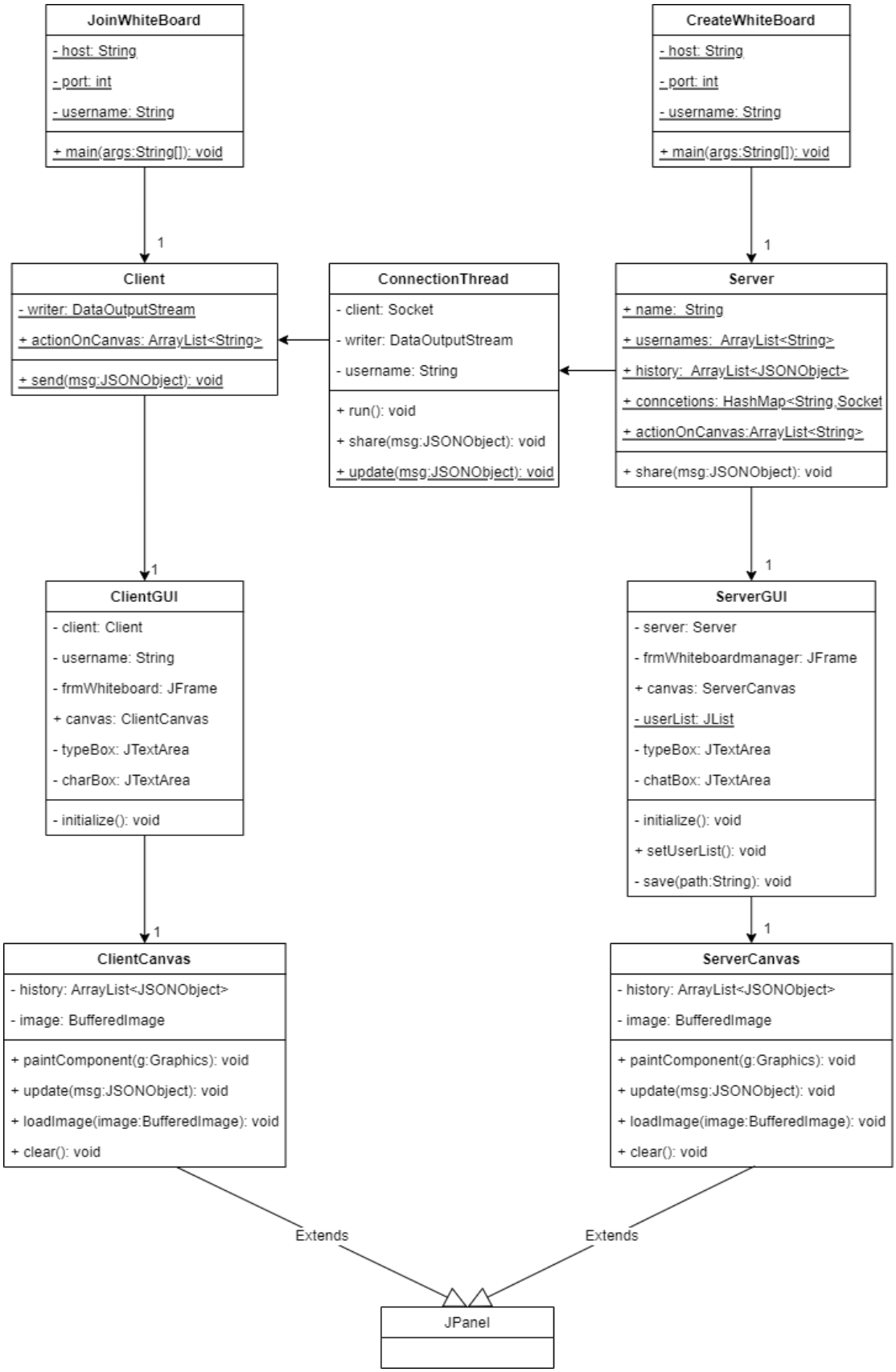
The system uses TCP connection to guarantee a reliable communication between manager and users. The message is JSON formatted. For each request, {"action": action} is specified. When manager or users get the message, they can easily identify which action to take.
Draw on canvas actions: message stores x1, y1, x2, y2 and colour

Key	value
action	Shapes
X1	X1
Y1	Y1
X2	X2
Y2	Y2
colour	Colour

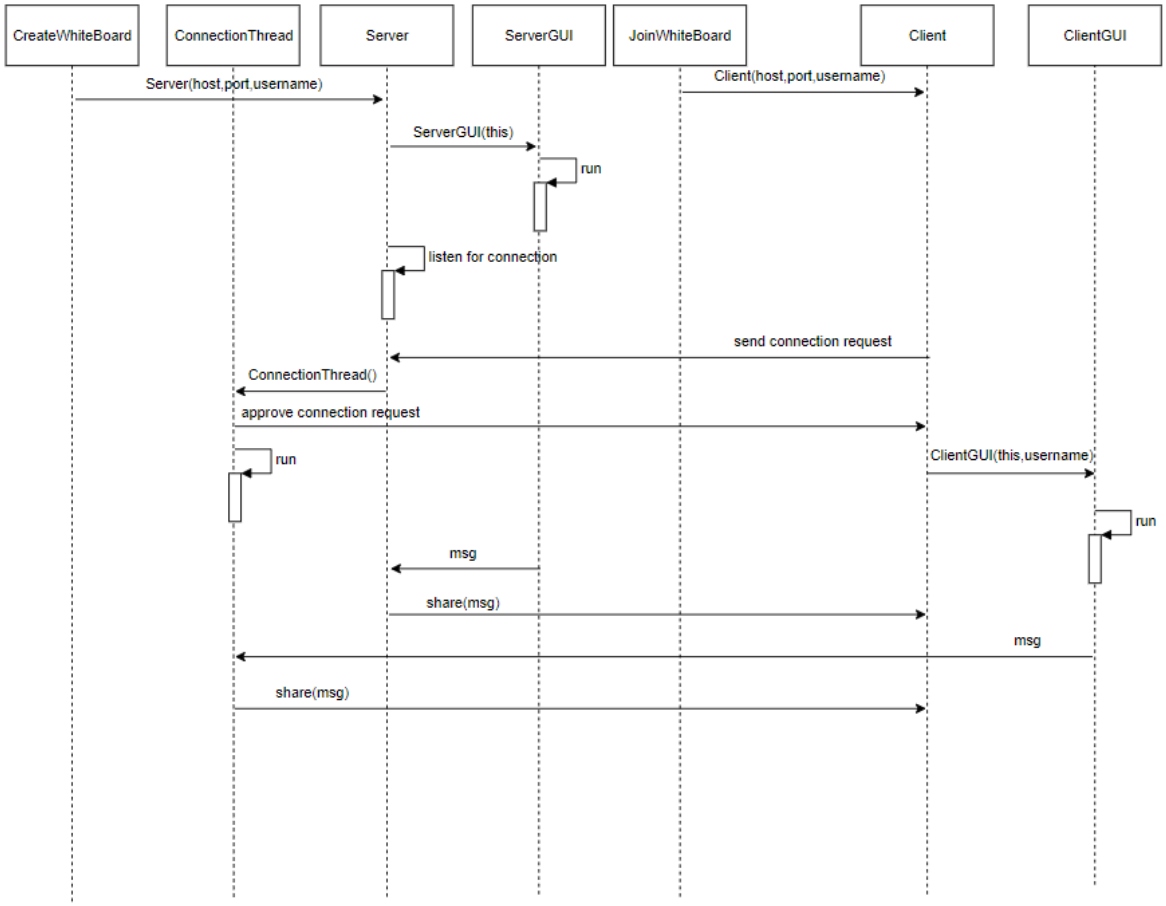
Other actions: message stores corresponding information
The reason why I choose TCP is that it is reliable. But there is still some disadvantages of using TCP. The speed of communication may not be fast enough. However, in this system, there are not so many messages being exchanged between manager and users, so TCP is suitable here.

Design diagrams

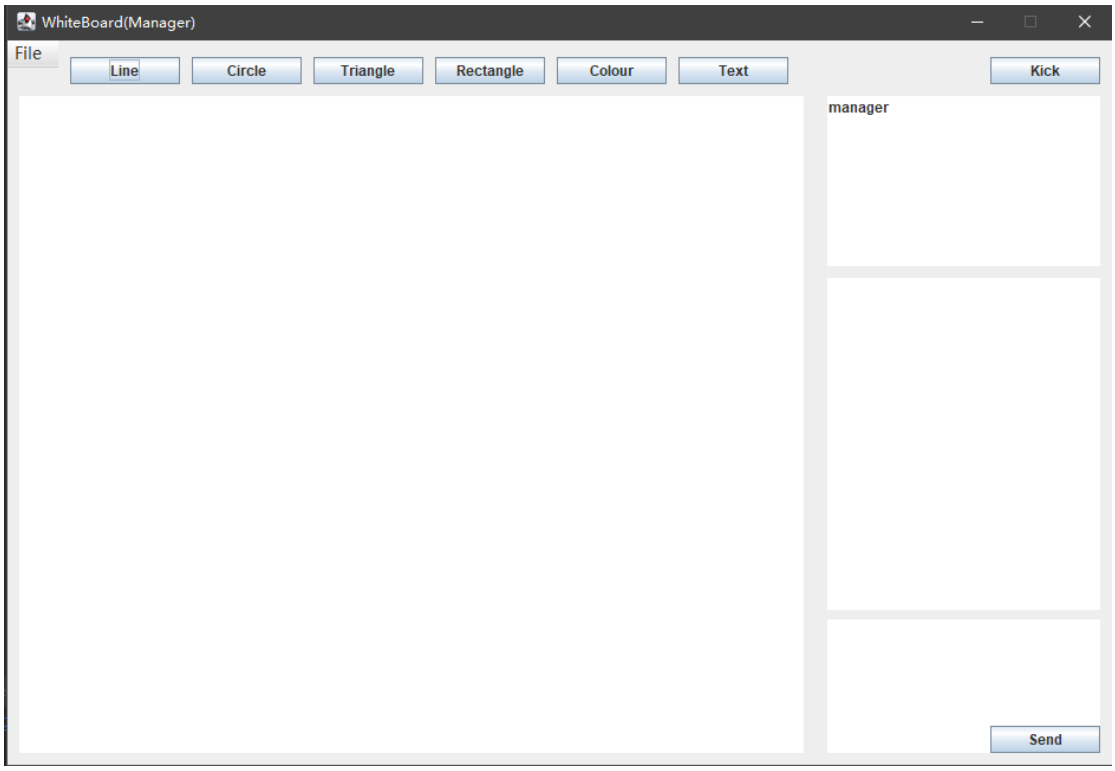
Class diagram



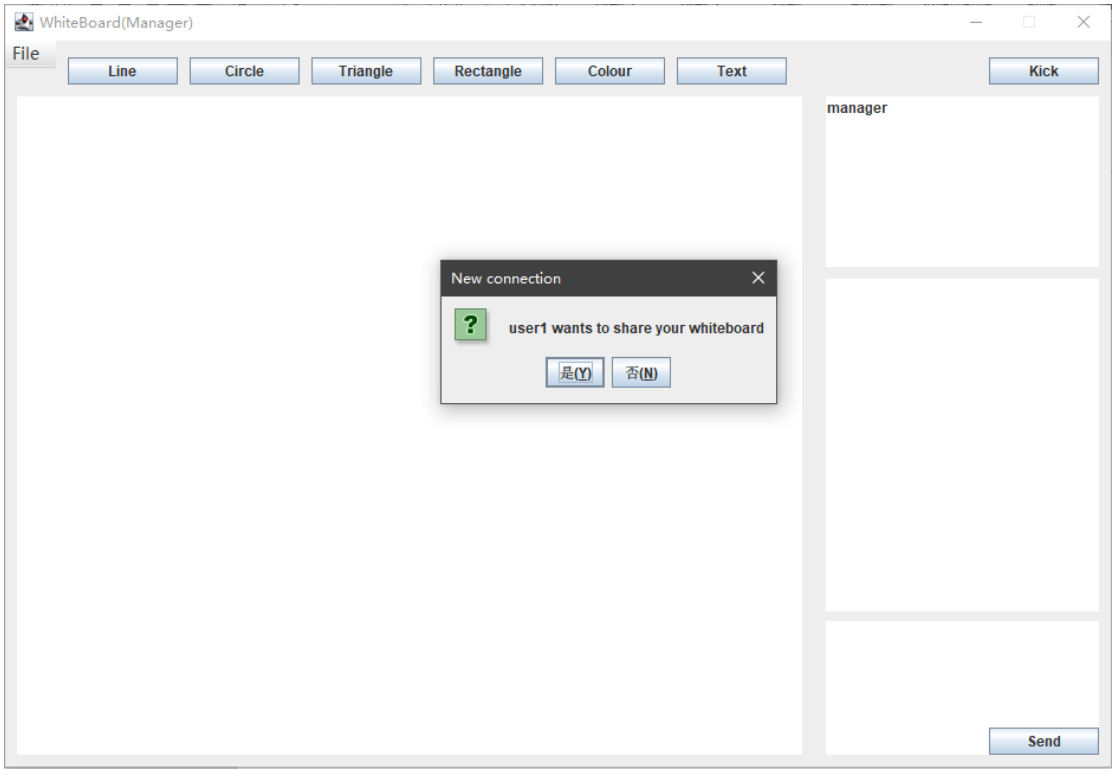
Sequence diagram



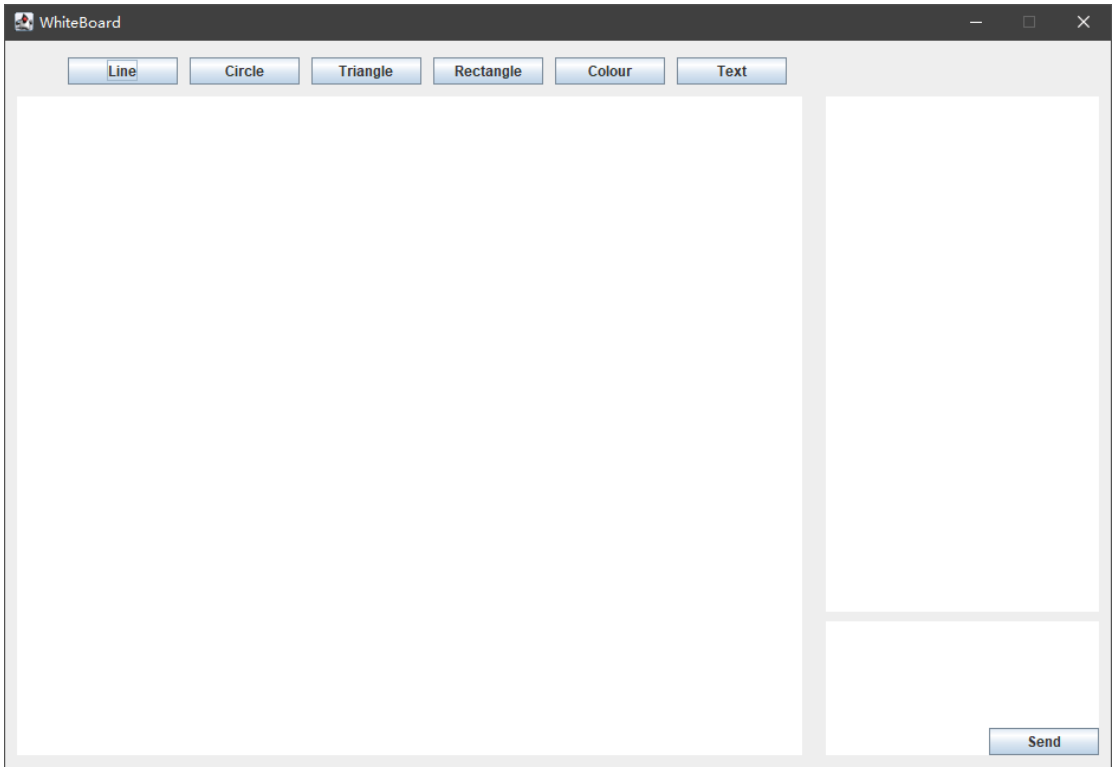
Implementation details



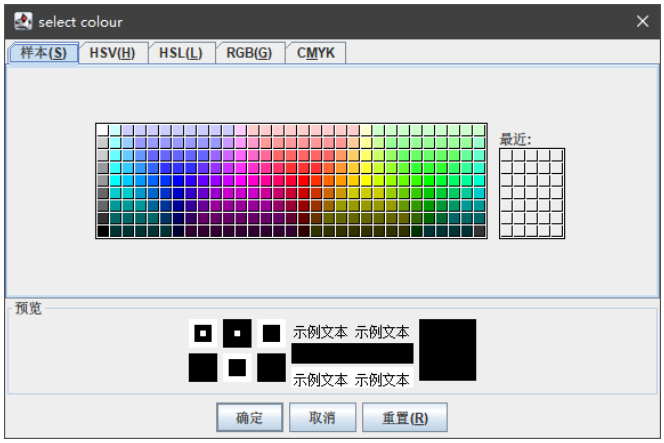
This is the server GUI. Manager can draw on the board, chat with others, have the access to a file menu, a kick button and a list of online users.



When a user wants to join the white board, a confirm dialog will pop up. And manager can choose whether to let that user join or not.



Above is the user's GUI. This will only pop up when the manager approves user's join request. As you can see, users can draw lines, circles, triangles and rectangles on the canvas.



Users can also choose their favourite colours. In this case, I used JColorChooser which provides a variety of colours and it is easy to implement. Users can also put texts on the canvas.

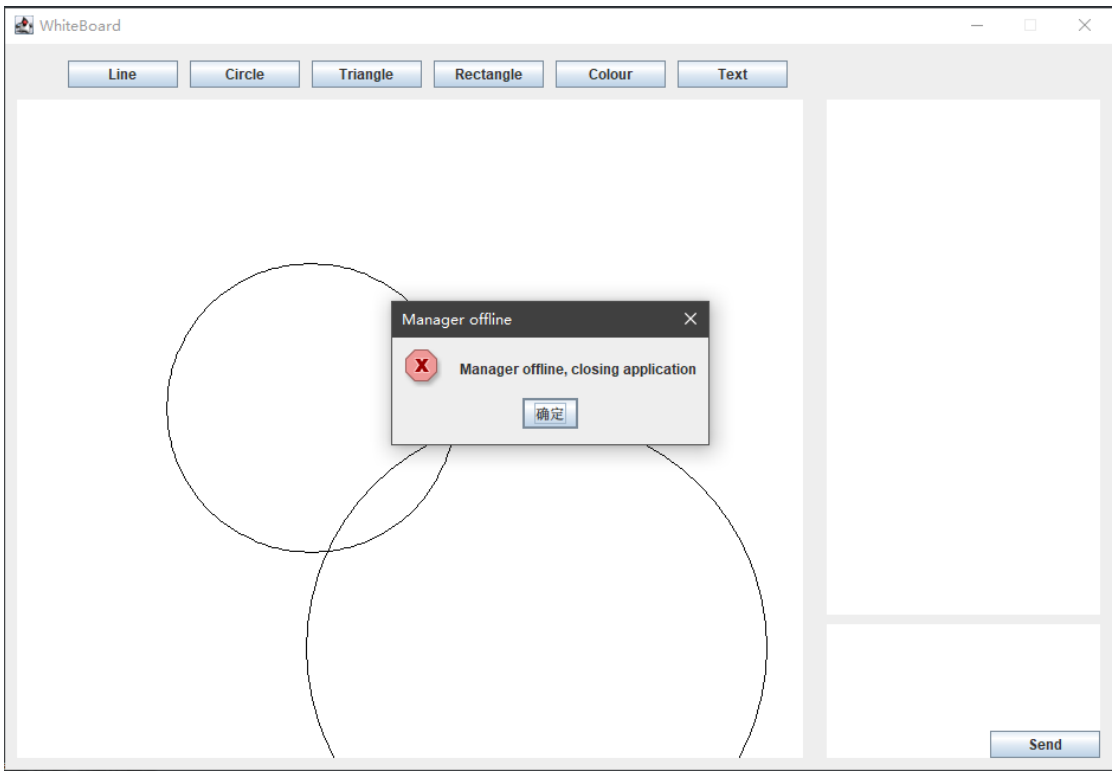
Users can use all the functionalities of the system except manager-only features.

In terms of how to let every user see the same white board, I store all the actions which draw things on the canvas into an array list. Every time a user joins the shared white board. The server will send the user all the history so that his board will be the same as everyone else the moment he joins the white board.

Each time someone makes changes to the white board, the server will share the message to every user who is connected to the server.

When the manager draws on the canvas, the server shares the message to everyone in the

network. When users draw on the canvas, they send a request to the server, then the server propagate the message to every user.



If the manager went offline for some reason. All the users who connected to the server will get a notification saying manager offline. And the application will be closed.

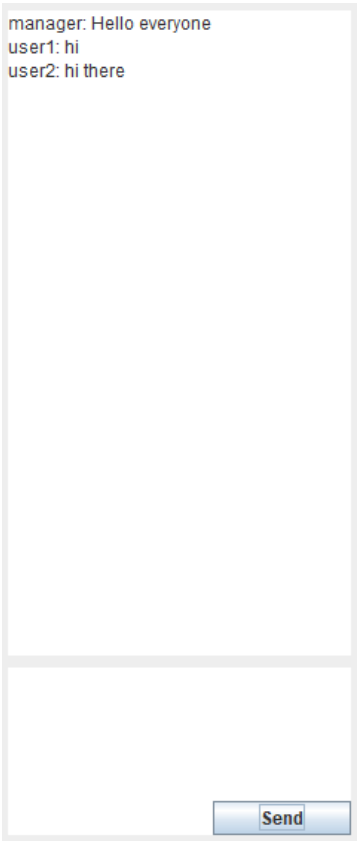
New innovations

Chat room

Users and manager can send messages to each other.

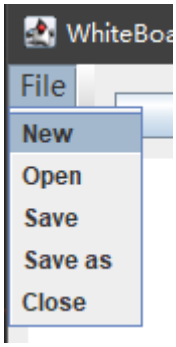
Users who joined late will not be able to see what others were talking before they join.

The chat room can help this system to be more interactive since users and manager can communicate directly using chat box.

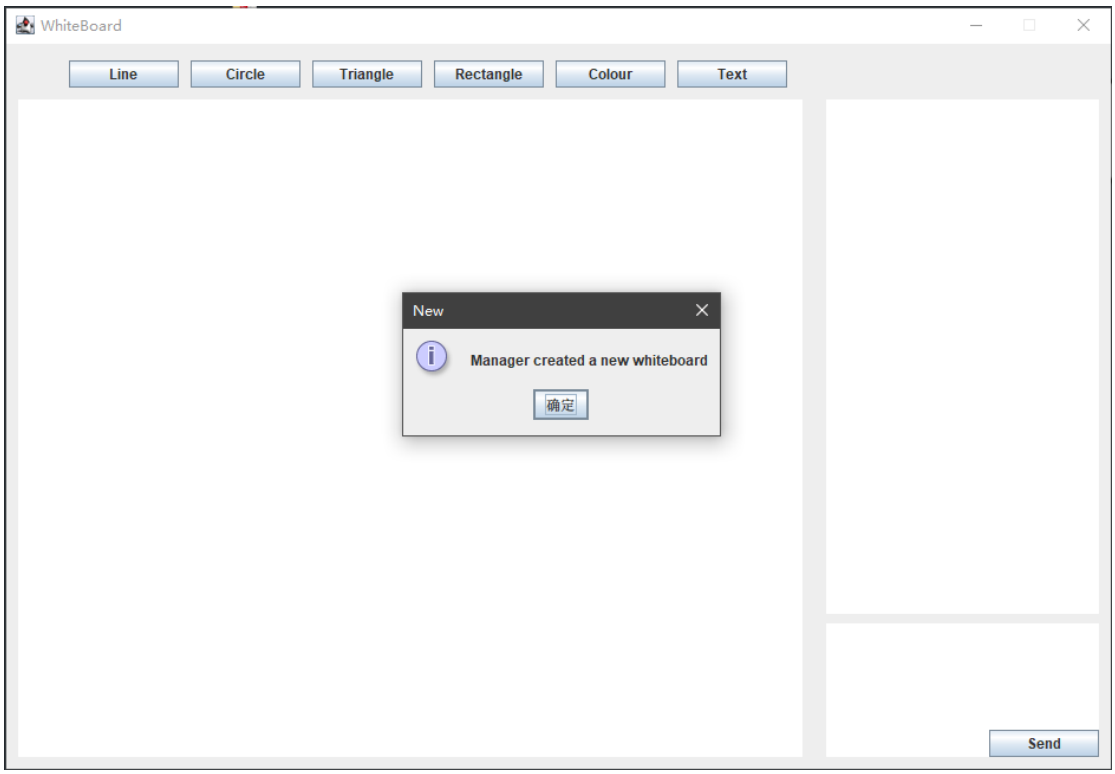


File menu

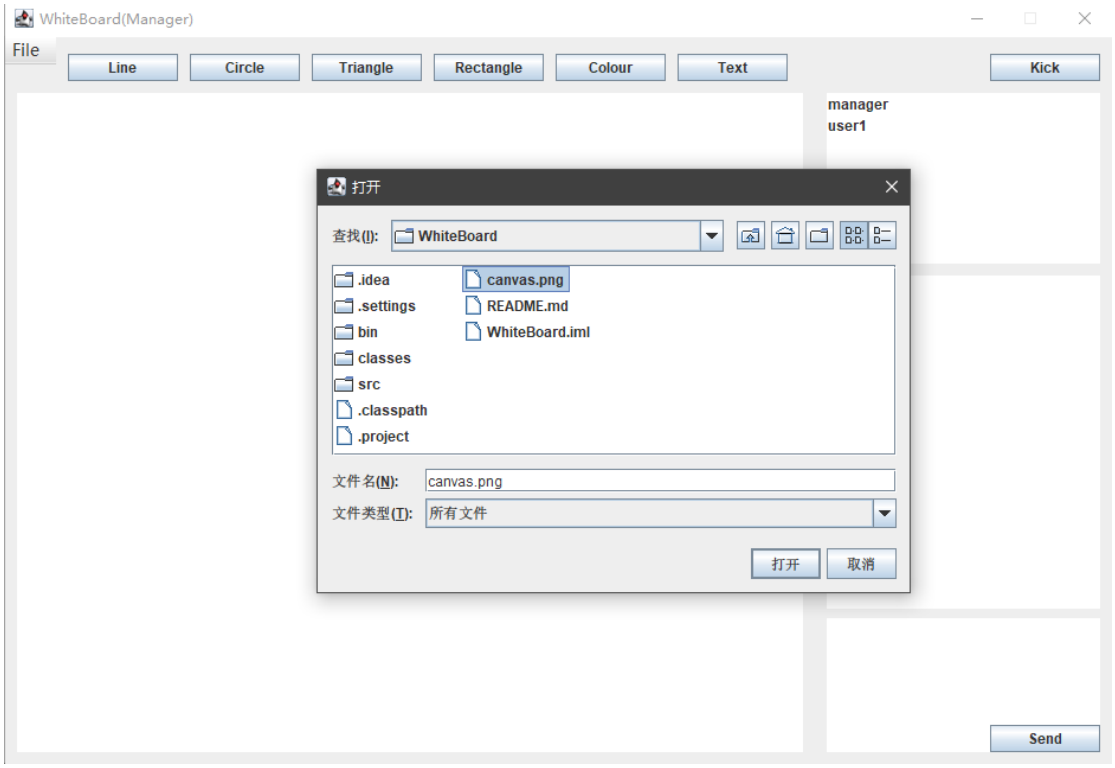
A File menu with new, open, save, saveAs and close is provided. Only manager can control this, which provides convenience to save and load pre-existing white board.



New: clear the previous whiteboard, start a new one

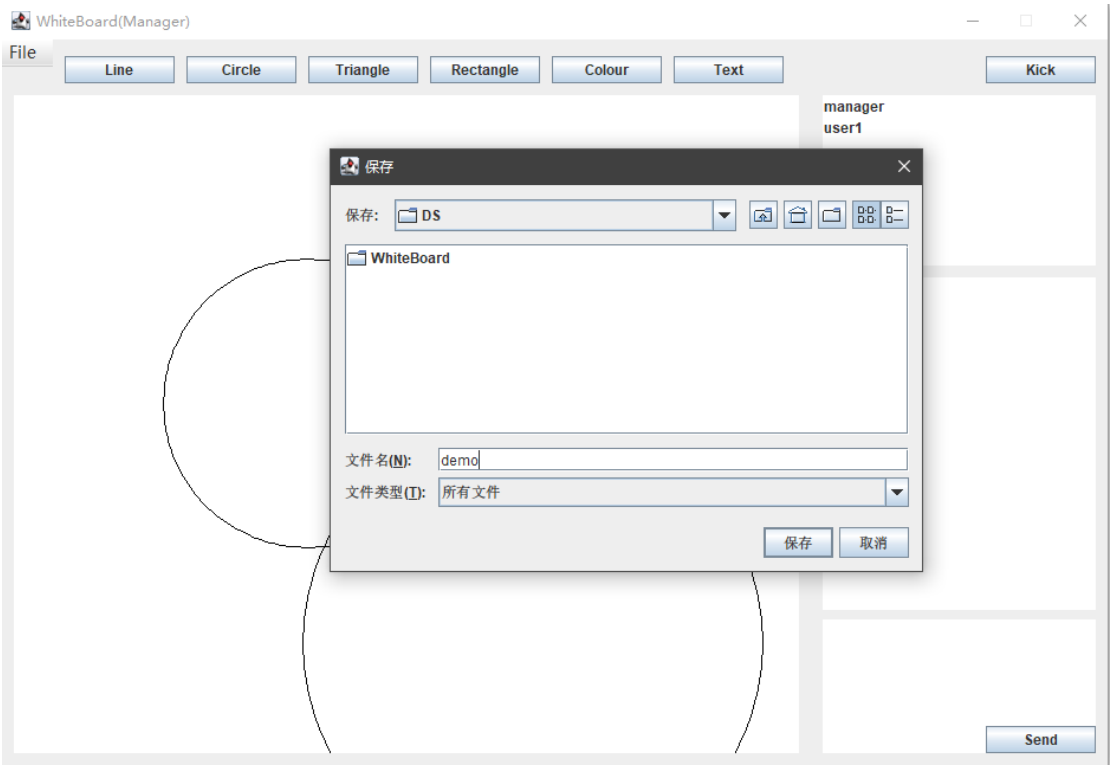


Open: Pop up a window for manager to select a png file and load the file on the whiteboard



Save: Save the current whiteboard as a png file to a default path

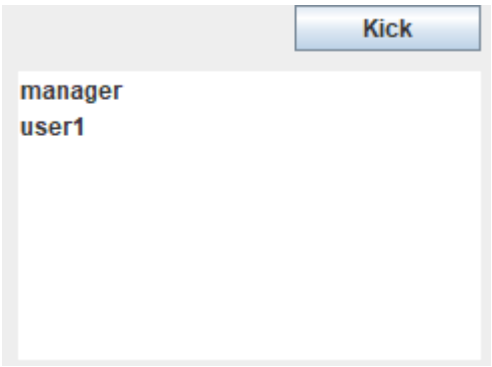
Save as: Pop up a window for manager to select a png file and store the whiteboard to that file.



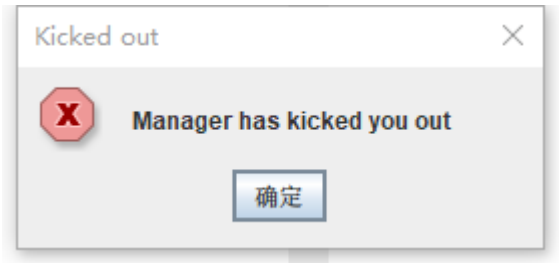
Close: Close the application

Kick

Manager can kick a certain user out.



A message will pop up on the user side telling him that Manager has kicked him out.



This helps manager to maintain the order of his whiteboard since the manager can kick people out if they did something rude or wrong.