Report of Distributed Shared White Board, COMP90051 Assignment 2, Semester 1, 2022

Weixian Fu, 1183945

29 maj 2022

1 Introduction

The problem is designing and implementing a distributed shared whiteboard that allows concurrent users(clients) to draw different shapes simultaneously on a canvas and chat online. This project is implemented by using RMI technology.

On the manager(server) side, the manager can draw shapes, chat with users, change the canvas, save the canvas, and save the canvas as a .png document. The manager can also decides whether to let the user in and kick out a certain user. A manager UI is implemented with java2D once the server is set up, and the user can connect to the server using RMI technology.

On the user(client) side, the user can draw shapes and chat with the other users and manager. Users use RMI technology to establish connections with managers and transmit and receive data. A login UI is implemented to let those users enter their user names and let the manager decide whether to let them in. A client Ui is implemented to implement the functions of drawing and chat.

As this project is implemented by using RMI technology, so its multi-threaded. Multi-clients can and will execute methods on remote objects simultaneously.

2 A brief description of the components of the system

The distributed shared whiteboard contains five components – CreateWhiteBoard, UI of manager, JoinWhiteBoard, UI of Client, Shape, StartServer, ServerService and ClientService.

CreateWhiteBoard: the implementation of manger of whiteboard, use the Class StartServer to set up the server and use the Class ManagerUI to show a whiteboard UI.

ManagerUI: the implementation of the UI of whiteboard of manager, manager can draw shapes such as line, circle, rectangle, triangle, pencil, oval, round rectangle, eraser and so on with different color. The shape are drawed by using java2D. Manager can also chat with users, change the canvas, save the canvas, and save the canvas as a .png document and kick a certain user.

StartServer: Start a server by using RMI technology.

ServerService: RMI object of server side. By using ServerServer, the manager can retrieve the user list, broadcast the chat message, kick user and retrieve the list of all shapes and so on.

JoinWhiteBoard: the implementation of user of whiteboard, it will set up an user name Ui first and after the user is allowed in, the UI of Client is set up by the Class ClientUI.

ClientUI: the implementation of the UI of whiteboard of client, manager can draw shapes such as line, circle, rectangle, triangle, pencil, oval, round rectangle, eraser and so on with different color. The shape are drawed by using java2D. Client can chat with other users and manager.

ClientServer: RMI object of client side. ClientServer will notify the information and changes of manger server.

3 Overall Class Design and an interface diagram

3.1 Overall Class Design: UML of Manager

On the Class CreateEWhiteBoard, by run the main methods, it will first set up a StartServer class and set up a ServerService Class and create a Registry. Then it set up the manager UI and register the ServerSerice.

One the Class ServerSevice, it offers 12 abstract methods and those methods is implemented in Class ServerSeviceIMine. The ServerServer can help the server to retrieve the user list, change or load the canvas, boradcast the chat message, kick user and retrieve the list of all shapes and so on.

The Class ManagerUI is the UI of manager, it can be show in 3.1.

3.2 Overall Class Design: UML of Client

On the Class JoinEWhiteBoard, by run the main methods, it will first set up a username UI. user can enter their username and click the button "Connect Now!", if the username have not been used, a notification will be send to manager UI. If the manager let this user in, the Manger UI will set up. The UI of username is shown in 3.2.

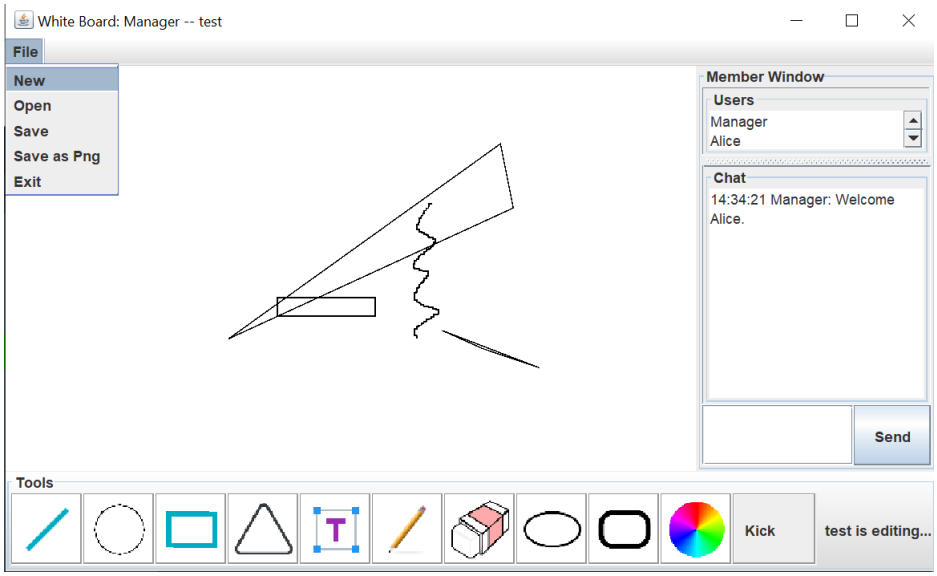
One the Class ClientSevice, it offers 10 abstract methods and those methods is implemented in Class ServerSeviceImpl. The ClientServer can help the server to notify the client the approval or kick and the leave of clients, and boradcast the chat message.

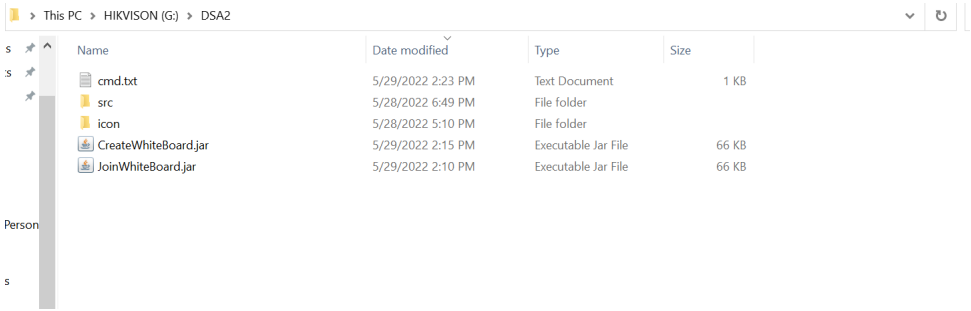
The Class ClientUI is the UI of Client, it can be show in 3.2.

3.3 Commands to Start server and Client

The CreateWhiteBoard.jar and JoinWhiteBoard.jar and the java document should be in the same directory, as the picture below shows:

Figur 1: Manager UI





Use the following commands to start the server and client by run the .jar file.

```
java -jar CreateWhiteBoard.jar localhost 5555 test
java -jar JoinWhiteBoard.jar localhost 5555 test
```

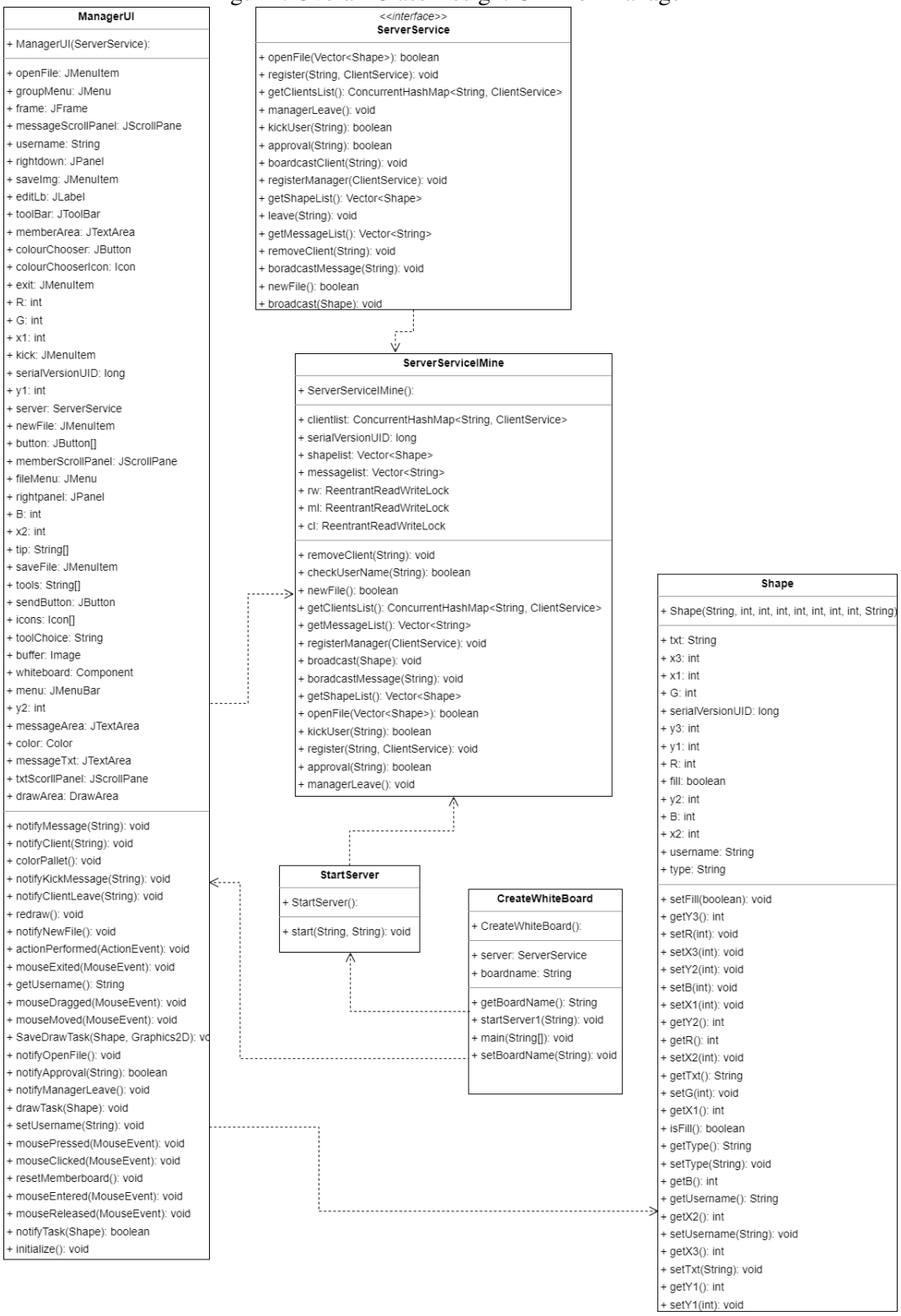
Use the following commands to complie the .java file.

```
javac src/CreateWhiteBoard.java
javac src/JoinWhiteBoard.java
```

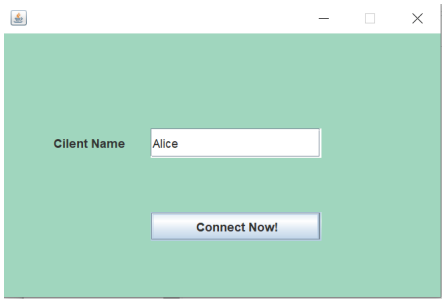
Then use the following commands to run the .java file.

```
java src.CreateWhiteBoard localhost 5555 test
java src.JoinWhiteBoard localhost 5555 test
```

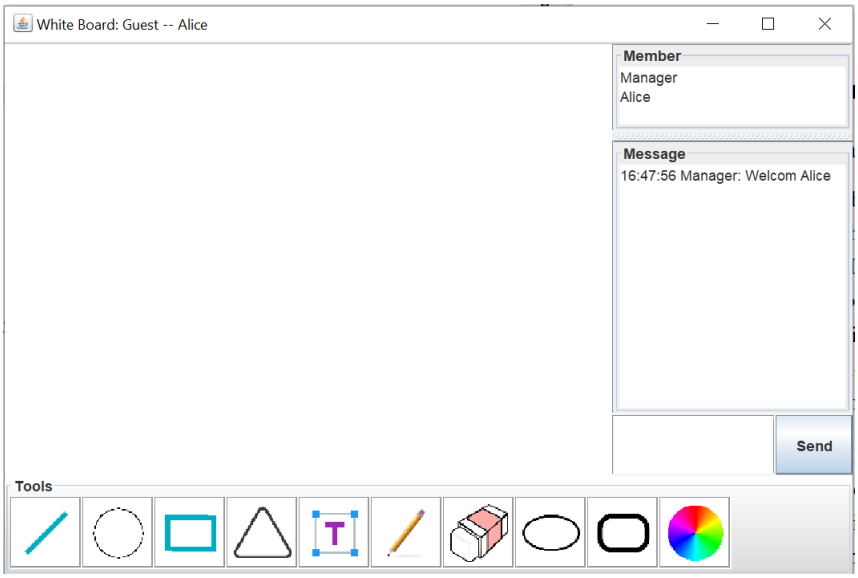
Figur 2: Overall Class Design: UML of Manager



Figur 3: Username UI



Figur 4: Client UI



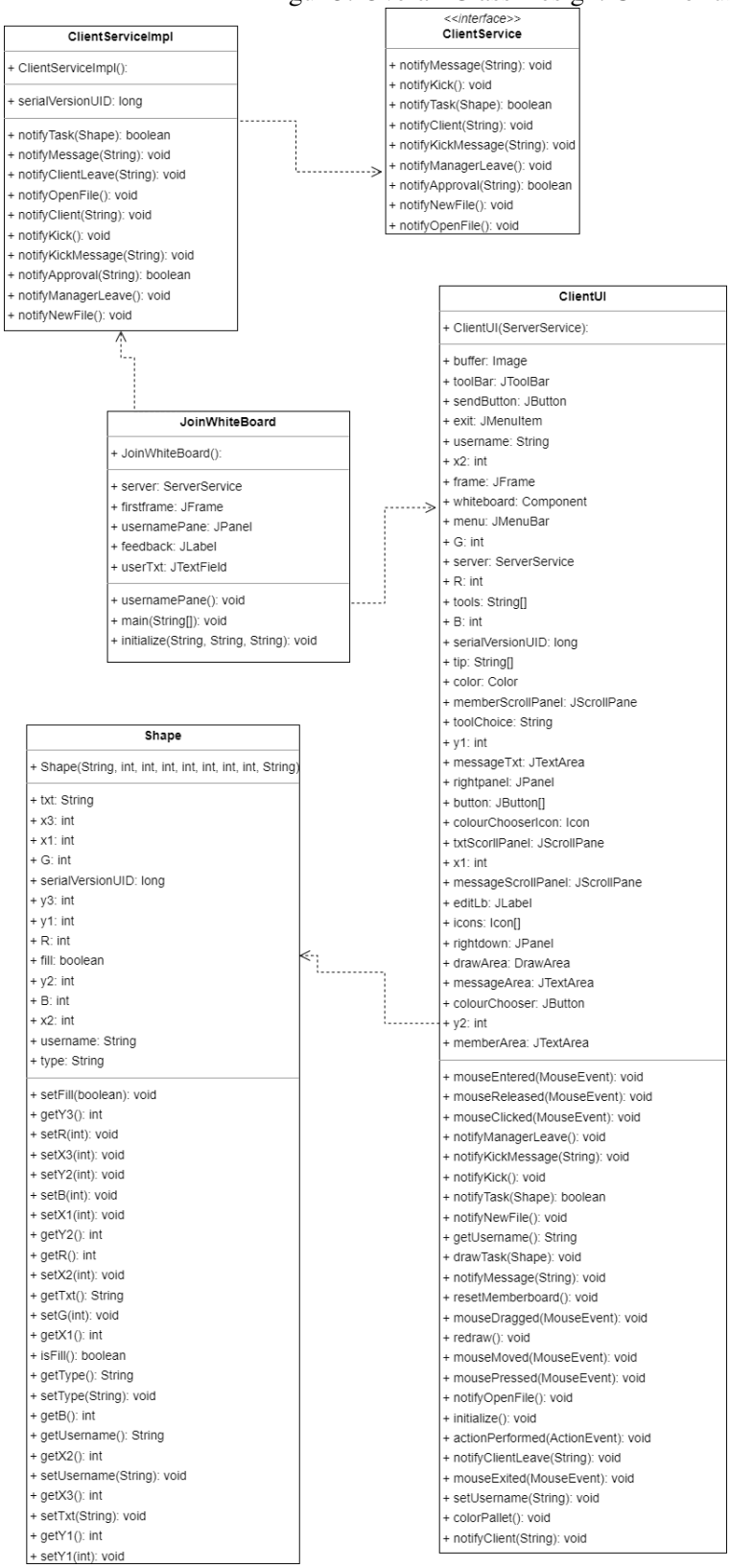
4 innovations and reflection

4.1 innovations: Advanced Features

- Chat Windows: users and manager can communicate with each other with a timestamp. The chat windows are implemented by RMI technology with the method boradcast-Message in Server and notifyMessage on the client-side. The message is in the string format in "HH:mm:ss username: messageformat. When a user sends a message, it will show in its Message area in UI, call the ServerService object's boradcastMessage method, and send the message to all users. The ServerService object will call the notifyMessage method in all clients and show the message.

- File menu: the manager can new, open, save, and save the canvas as .png file, and only the manager can control this. The new function will just clear the canvas and notify all

Figur 5: Overall Class Design: UML of user



clients by newFile method in ServerService and notifyNewFile in all ClientService objects. The open function can load the serialized list of shape objects and the save function can save the serialized list of shape objects. The save as .png function can save the canvas in .png document by ImageIO.write() function.

- Kick function: the manager can kick out a certain user by enter its name. The kick features is impletmented by kickUser method in ServerService object and notifyKick in ClientService.
- White board support line, circle, rectangle, triangle, text, pencil, oval, round rectangle and eraser, people can choose their favourite color. The draw function is implemented by java2D, and swing.JColorChooser;
- When some one is drawing, the UI will show the username of who is drawing.

4.2 message formats

- Message format: the message is send in format of string, the format is "HH:mm:ss username: message".
- Shape format: The shape object is implemented in Class shape. When the manager is trying to save or load the canvas, the shape will be saved or loaded in serialized format.
- Username format: the user name is save in a ConcurrentHashMap.

4.3 Disadvantages

- The user interface is not friendly. Manager can't directly kick by clicking the user name.
- The triangle drawing process needs to be optimized. Now the third vertex of triangle is generated and drawed randomly.
- The setup of server and clients are separated, and they can be combined into the same UI interface in the future.
- A user registration system and a user information database can be constructed.